StorageGRID를 사용합니다

StorageGRID

NetApp
April 10, 2024

목차

StorageGRID를 사용합니다

테넌트 계정을 사용합니다

테넌트 계정 사용: 개요

테넌트 계정을 사용하면 S3(Simple Storage Service) REST API 또는 Swift REST API를 사용하여 StorageGRID 시스템에 오브젝트를 저장하고 검색할 수 있습니다.

테넌트 계정이란 무엇입니까?

각 테넌트 계정에는 자체 통합 또는 로컬 그룹, 사용자, S3 버킷 또는 Swift 컨테이너 및 객체가 있습니다.

필요한 경우 테넌트 계정을 사용하여 저장된 객체를 다른 엔터티로 분리할 수 있습니다. 예를 들어, 다음과 같은 사용 사례에서 여러 테넌트 계정을 사용할 수 있습니다.

- * 기업 활용 사례: * 기업 내에서 StorageGRID 시스템을 사용하는 경우 그리드의 객체 스토리지를 조직의 여러 부서에서 분리할 수 있습니다. 예를 들어 마케팅 부서, 고객 지원 부서, 인사 부서 등의 테넌트 계정이 있을 수 있습니다.

> S3 클라이언트 프로토콜을 사용하는 경우 S3 버킷 및 버킷 정책을 사용하여 엔터프라이즈의 부서 간에 오브젝트를 분리할 수도 있습니다. 별도의 테넌트 계정을 생성할 필요가 없습니다. 를 참조하십시오 S3 클라이언트 애플리케이션 구현 지침.

- * 서비스 공급자 사용 사례: * 서비스 공급자가 StorageGRID 시스템을 사용 중인 경우, 스토리지를 임대하는 다른 엔터티로 그리드의 개체 스토리지를 분리할 수 있습니다. 예를 들어 회사 A, 회사 B, 회사 C 등에 대한 테넌트 계정이 있을 수 있습니다.

테넌트 계정을 생성하는 방법

테넌트 계정은 에 의해 생성됩니다 그리드 관리자를 사용하는 StorageGRID 그리드 관리자. 테넌트 계정을 생성할 때 그리드 관리자는 다음 정보를 지정합니다.

- 테넌트의 표시 이름(테넌트의 계정 ID가 자동으로 할당되며 변경할 수 없음)
- 테넌트 계정에서 S3 또는 Swift를 사용할지 여부를 나타냅니다.
- S3 테넌트 계정의 경우: 테넌트 계정이 플랫폼 서비스를 사용하도록 허용되는지 여부 플랫폼 서비스를 사용할 수 있는 경우 그리드 사용을 지원하도록 구성해야 합니다.
- 필요한 경우 테넌트 계정의 스토리지 할당량 — 테넌트의 객체에 사용할 수 있는 최대 GB, 테라바이트 또는 PB입니다. 테넌트의 스토리지 할당량은 물리적 크기(디스크 크기)가 아닌 논리적 양(오브젝트 크기)을 나타냅니다.
- StorageGRID 시스템에 대해 ID 페더레이션이 설정된 경우 테넌트 계정을 구성할 수 있는 루트 액세스 권한이 있는 통합 그룹입니다.
- StorageGRID 시스템에서 SSO(Single Sign-On)를 사용하지 않는 경우 테넌트 계정이 자체 ID 소스를 사용할지 또는 그리드의 ID 소스를 공유할지 여부 및 테넌트의 로컬 루트 사용자의 초기 암호를 공유할지 여부

또한, S3 테넌트 계정이 규정 요구 사항을 준수해야 하는 경우 그리드 관리자는 StorageGRID 시스템에 대해 S3 오브젝트 잠금 설정을 활성화할 수 있습니다. S3 오브젝트 잠금이 활성화된 경우 모든 S3 테넌트 계정에서 호환 버킷을 생성하고 관리할 수 있습니다.

S3 테넌트를 구성합니다

을(를) 마친 후 S3 테넌트 계정이 생성됩니다테넌트 관리자에 액세스하여 다음과 같은 작업을 수행할 수 있습니다.

- ID 통합을 설정하거나(ID 소스를 그리드와 공유하지 않는 경우) 로컬 그룹 및 사용자를 생성합니다
- S3 액세스 키 관리
- 규정 준수 버킷을 포함하여 S3 버킷 생성 및 관리
- 플랫폼 서비스 사용(활성화된 경우)
- 스토리지 사용량 모니터링

> 테넌트 관리자를 사용하여 S3 버킷을 생성 및 관리할 수 있지만 에는 가 있어야 합니다 S3 액세스 키를 사용하고 S3 REST API를 사용하여 오브젝트를 수집 및 관리합니다.

Swift 테넌트를 구성합니다

A 뒤에 Swift 테넌트 계정이 생성됩니다테넌트 관리자에 액세스하여 다음과 같은 작업을 수행할 수 있습니다.

- ID 통합을 설정하고(ID 소스를 그리드와 공유하지 않는 경우) 로컬 그룹 및 사용자를 만듭니다
- 스토리지 사용량 모니터링

> Swift 사용자는 테넌트 관리자에 액세스하려면 루트 액세스 권한이 있어야 합니다. 그러나 루트 액세스 권한에서는 사용자가 에 인증할 수 없습니다 Swift REST API 컨테이너 및 수집 개체 생성 사용자는 Swift REST API에 인증할 수 있는 Swift 관리자 권한이 있어야 합니다.

테넌트 관리자를 사용합니다

테넌트 관리자를 사용하면 StorageGRID 테넌트 계정의 모든 측면을 관리할 수 있습니다.

테넌트 관리자를 사용하여 테넌트 계정의 스토리지 사용량을 모니터링하고 ID 페더레이션을 통해 사용자를 관리하거나 로컬 그룹 및 사용자를 생성할 수 있습니다. S3 테넌트 계정의 경우 S3 키를 관리하고 S3 버킷을 관리하고 플랫폼 서비스를 구성할 수도 있습니다.

로그인 및 로그아웃 방법

테넌트 관리자에 로그인합니다

의 주소 표시줄에 테넌트에 대한 URL을 입력하여 테넌트 관리자에 액세스합니다 지원되는 웹 브라우저.

필요한 것

- 로그인 자격 증명이 있어야 합니다.
- 그리드 관리자가 제공한 대로 테넌트 관리자에 액세스하기 위한 URL이 있어야 합니다. URL은 다음 예 중 하나로 표시됩니다.

```
https://FQDN_or_Admin_Node_IP/
```

```
https://FQDN_or_Admin_Node_IP:port/
```

```
https://FQDN_or_Admin_Node_IP/?accountId=20-digit-account-id
```

```
https://FQDN_or_Admin_Node_IP:port/?accountId=20-digit-account-id
```

URL은 항상 FQDN(정규화된 도메인 이름) 또는 관리 노드에 액세스하는 데 사용되는 IP 주소를 포함하며, 포트 번호, 20자리 테넌트 계정 ID 또는 둘 다를 선택적으로 포함할 수도 있습니다.

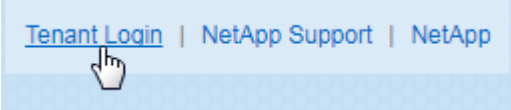
- URL에 테넌트의 20자리 계정 ID가 포함되지 않은 경우 이 계정 ID가 있어야 합니다.
- 을(를) 사용해야 합니다 지원되는 웹 브라우저.
- 웹 브라우저에서 쿠키를 활성화해야 합니다.
- 특정 액세스 권한이 있어야 합니다.

단계

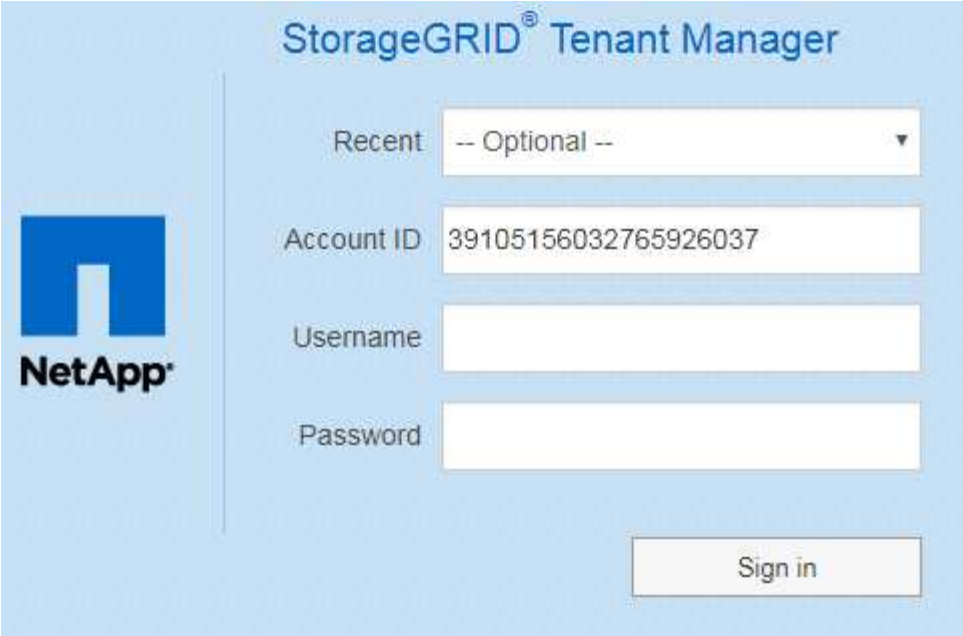
1. A를 시작합니다 지원되는 웹 브라우저.
2. 브라우저의 주소 표시줄에 Tenant Manager에 액세스하기 위한 URL을 입력합니다.
3. 보안 경고 메시지가 나타나면 브라우저의 설치 마법사를 사용하여 인증서를 설치합니다.
4. 테넌트 관리자에 로그인합니다.

   표시되는 로그인 화면은 입력한 URL과 조직에서 SSO(Single Sign-On)를 사용하고 있는지 여부에 따라 달라집니다. 다음 화면 중 하나가 표시됩니다.

   - Grid Manager 로그인 페이지 오른쪽 상단에서 * Tenant Login * 링크를 클릭합니다.

   - Tenant Manager 로그인 페이지. 아래와 같이 * Account ID * 필드가 이미 입력되어 있을 수 있습니다.



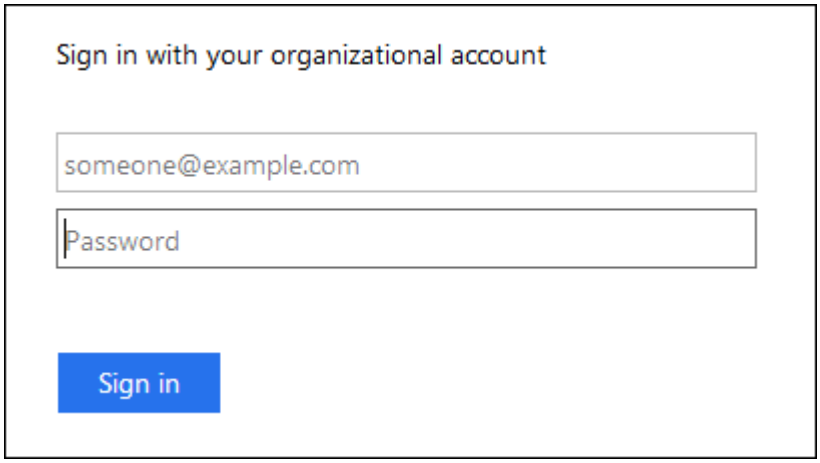
i. 테넌트의 20자리 계정 ID가 표시되지 않으면 최근 계정 목록에 테넌트 계정 이름이 나타날 경우 해당 계정 이름을 선택하거나 계정 ID를 입력합니다.

ii. 사용자 이름과 암호를 입력합니다.

iii. 로그인 * 을 클릭합니다.

Tenant Manager 대시보드가 나타납니다.

- SSO가 그리드에 활성화되어 있는 경우 조직의 SSO 페이지. 예를 들면 다음과 같습니다.



표준 SSO 자격 증명을 입력하고 * 로그인 * 을 클릭합니다.

- Tenant Manager SSO 로그인 페이지.

i. 테넌트의 20자리 계정 ID가 표시되지 않으면 최근 계정 목록에 테넌트 계정 이름이 나타날 경우 해당 계정 이름을 선택하거나 계정 ID를 입력합니다.

ii. 로그인 * 을 클릭합니다.

iii. 조직의 SSO 로그인 페이지에서 표준 SSO 자격 증명을 사용하여 로그인합니다.

Tenant Manager 대시보드가 나타납니다.

5. 다른 사람으로부터 초기 암호를 받은 경우 암호를 변경하여 계정을 보호하십시오. 사용자 이름 _ * > * 암호 변경 * 을 선택합니다.

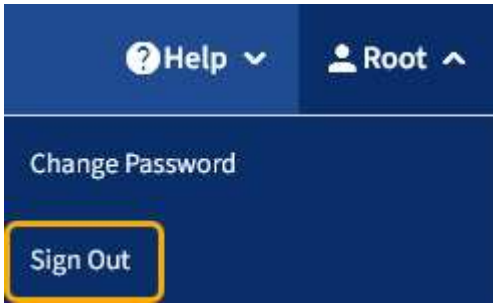
> StorageGRID 시스템에 SSO가 설정되어 있으면 테넌트 관리자에서 암호를 변경할 수 없습니다.

테넌트 관리자에서 로그아웃합니다

테넌트 관리자 작업을 마치면 로그아웃하여 권한이 없는 사용자가 StorageGRID 시스템에 액세스할 수 없도록 해야 합니다. 브라우저를 닫아도 브라우저 쿠키 설정에 따라 시스템에서 로그아웃되지 않을 수 있습니다.

단계

1. 사용자 인터페이스의 오른쪽 위 모서리에서 사용자 이름 드롭다운을 찾습니다.



2. 사용자 이름을 선택한 다음 * 로그아웃 * 을 선택합니다.

   - SSO가 사용되지 않는 경우:

     관리자 노드에서 로그아웃되었습니다. Tenant Manager 로그인 페이지가 표시됩니다.

> 두 개 이상의 관리 노드에 로그인한 경우 각 노드에서 로그아웃해야 합니다.

- SSO가 활성화된 경우:

  액세스 중인 모든 관리 노드에서 로그아웃되었습니다. StorageGRID 로그인 페이지가 표시됩니다. 방금 액세스한 테넌트 계정의 이름이 * 최근 계정 * 드롭다운에 기본값으로 나열되고 테넌트의 * 계정 ID * 가 표시됩니다.

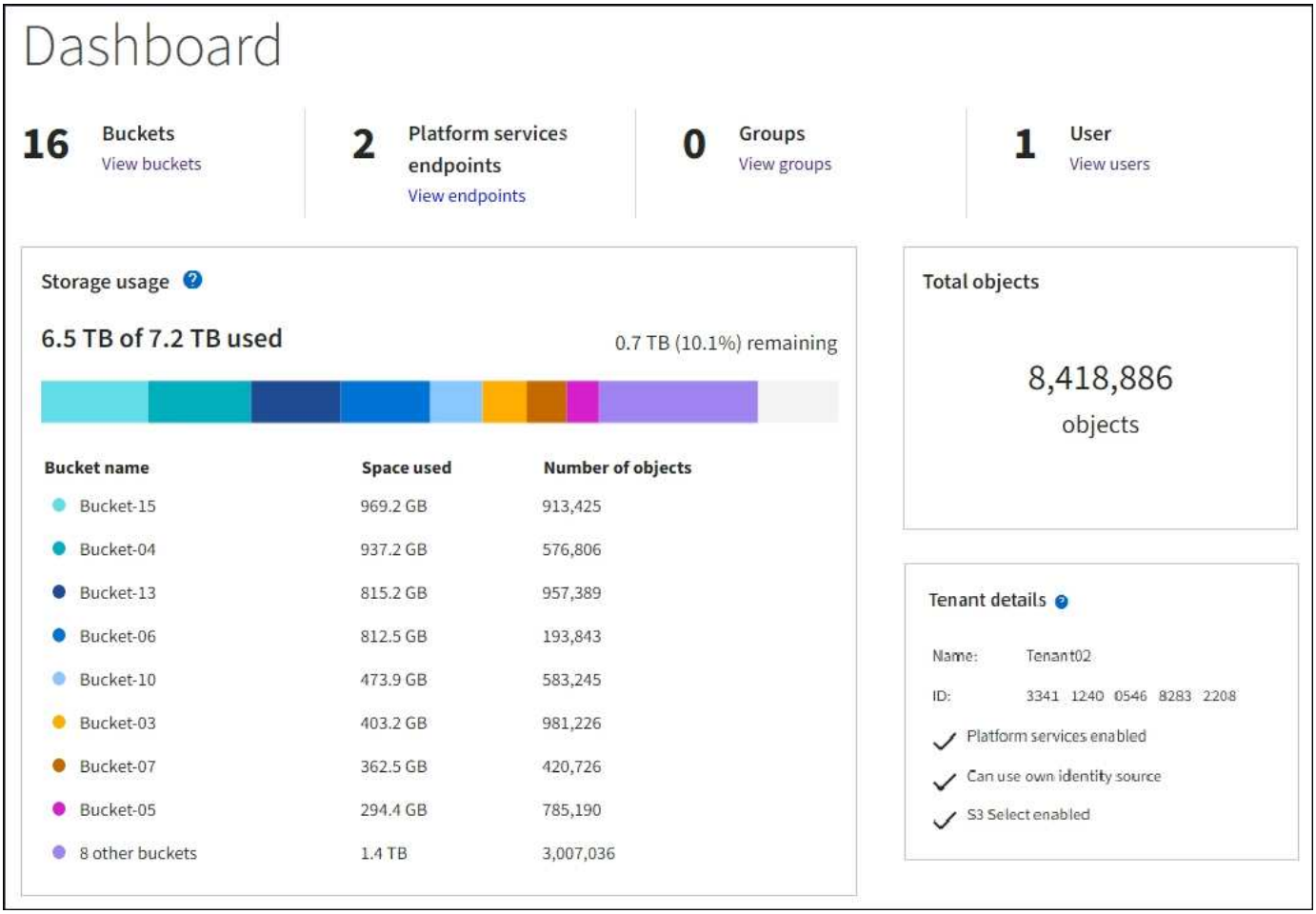
> SSO가 활성화되어 있고 Grid Manager에도 로그인한 경우, Grid Manager에서 로그아웃하여 SSO를 로그아웃해야 합니다.

테넌트 관리자 대시보드 이해

테넌트 관리자 대시보드에서는 테넌트 계정의 구성과 테넌트의 버킷(S3) 또는 컨테이너(Swift)에 있는 객체가 사용하는 공간의 양을 개괄적으로 보여 줍니다. 테넌트에 할당량이 있는 경우 대시보드에는 사용된 할당량의 양과 남아 있는 양이 표시됩니다. 테넌트 계정과 관련된 오류가 있는 경우 대시보드 에 오류가 표시됩니다.

> 사용된 공간 값은 추정값입니다. 이러한 추정치는 베스트 타이밍, 네트워크 연결 및 노드 상태의 영향을 받습니다.

객체가 업로드되면 대시보드는 다음 예제와 같습니다.



테넌트 계정 요약

대시보드 상단에는 다음 정보가 포함되어 있습니다.

- 구성된 버킷 또는 컨테이너, 그룹 및 사용자 수
- 구성된 플랫폼 서비스 엔드포인트의 수입니다

링크를 선택하여 세부 정보를 볼 수 있습니다.

대시보드의 오른쪽에는 다음과 같은 정보가 포함되어 있습니다.

- 테넌트의 총 객체 수입니다.

  S3 계정의 경우 오브젝트가 수집되지 않고 루트 액세스 권한이 있는 경우 총 오브젝트 수 대신 시작 지침이 나타납니다.

- 테넌트 계정 이름 및 ID와 테넌트가 사용할 수 있는지 여부를 포함한 테넌트 세부 정보입니다 플랫폼 서비스, 고유한 ID 소스입니다, 또는 S3 를 선택합니다 (설정된 권한만 나열됩니다).

스토리지 및 할당량 사용

Storage usage(저장소 사용) 패널에는 다음과 같은 정보가 포함되어 있습니다.

- 테넌트에 대한 객체 데이터의 양입니다.

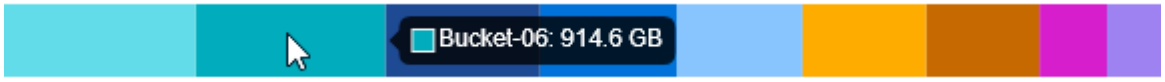
  > 이 값은 업로드된 총 오브젝트 데이터 양을 나타내며 해당 오브젝트 및 해당 메타데이터의 복사본을 저장하는 데 사용되는 공간을 나타내지 않습니다.

- 할당량이 설정된 경우 개체 데이터에 사용할 수 있는 총 공간과 남은 공간의 양과 백분율이 표시됩니다. 할당량은 섭취 가능한 오브젝트 데이터의 양을 제한합니다.

  > 할당량 활용도는 내부 추정치에 기반하며 경우에 따라 초과될 수 있습니다. 예를 들어, 테넌트가 객체를 업로드하기 시작할 때 StorageGRID는 할당량을 확인하고 테넌트가 할당량을 초과할 경우 새 베스트(ingest)를 거부합니다. 그러나 StorageGRID에서는 할당량이 초과되었는지 확인할 때 현재 업로드 크기를 고려하지 않습니다. 개체를 삭제하면 할당량 활용률이 다시 계산될 때까지 테넌트가 일시적으로 새 개체를 업로드하지 못할 수 있습니다. 할당량 사용률 계산에는 10분 이상이 소요될 수 있습니다.

- 가장 큰 버킷 또는 컨테이너의 상대적 크기를 나타내는 막대 차트.

  차트 세그먼트 위에 커서를 놓으면 해당 버킷이나 컨테이너에서 소비한 전체 공간을 볼 수 있습니다.



- 막대 도표에 대응하려면 총 오브젝트 데이터 양과 각 버킷 또는 컨테이너의 오브젝트 수를 포함하여 가장 큰 버킷 또는 컨테이너의 목록입니다.

Bucket name	Space used	Number of objects
Bucket-02	944.7 GB	7,575
Bucket-09	899.6 GB	589,677
Bucket-15	889.6 GB	623,542
Bucket-06	846.4 GB	648,619
Bucket-07	730.8 GB	808,655
Bucket-04	700.8 GB	420,493
Bucket-11	663.5 GB	993,729
Bucket-03	656.9 GB	379,329
9 other buckets	2.3 TB	5,171,588

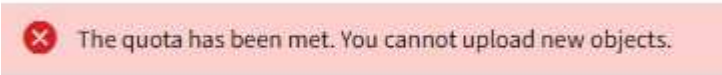
테넌트에 9개 이상의 버킷이나 컨테이너가 있는 경우 다른 모든 버킷이나 컨테이너는 목록 하단의 단일 항목으로 결합됩니다.

할당량 사용 알림을 표시합니다

그리드 관리자에서 할당량 사용 알림이 활성화된 경우 할당량이 낮거나 초과되면 다음과 같이 테넌트 관리자에 표시됩니다.

테넌트 할당량의 90% 이상이 사용된 경우 * Tenant quota usage high * 경고가 트리거됩니다. 자세한 내용은 StorageGRID 모니터링 및 문제 해결 설명서의 경고 참조를 참조하십시오.

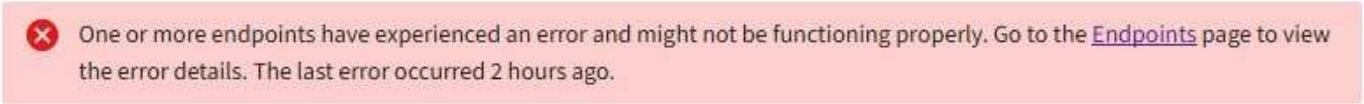
할당량을 초과하면 새 객체를 업로드할 수 없습니다.



추가 세부 정보를 보고 알림에 대한 규칙 및 알림을 관리하려면 StorageGRID 모니터링 및 문제 해결 지침을 참조하십시오.

끝점 오류

Grid Manager를 사용하여 플랫폼 서비스에 사용할 하나 이상의 엔드포인트를 구성한 경우 지난 7일 이내에 엔드포인트 오류가 발생한 경우 Tenant Manager 대시보드에 경고가 표시됩니다.



끝점 오류에 대한 세부 정보를 보려면 끝점 을 선택하여 끝점 페이지를 표시합니다.

관련 정보

플랫폼 서비스 끝점 오류 문제 해결

모니터링하고 문제를 해결합니다

테넌트 관리 **API**

테넌트 관리 **API** 이해

테넌트 관리자 사용자 인터페이스 대신 테넌트 관리 REST API를 사용하여 시스템 관리 작업을 수행할 수 있습니다. 예를 들어, API를 사용하여 작업을 자동화하거나 사용자와 같은 여러 엔터티를 더 빠르게 생성할 수 있습니다.

테넌트 관리 API:

- Swagger 오픈 소스 API 플랫폼을 사용합니다. Swagger는 개발자와 개발자가 아닌 사용자가 API와 상호 작용할 수 있는 직관적인 사용자 인터페이스를 제공합니다. Swagger 사용자 인터페이스는 각 API 작동에 대한 전체 세부 정보와 문서를 제공합니다.
- 사용 무중단 업그레이드를 지원하는 버전 관리.

테넌트 관리 API에 대한 Swagger 문서에 액세스하려면 다음을 수행합니다.

단계

1. 테넌트 관리자에 로그인합니다.
2. 테넌트 관리자 상단에서 도움말 아이콘을 선택하고 * API Documentation * 을 선택합니다.

API 작업

테넌트 관리 API는 사용 가능한 API 작업을 다음 섹션으로 구성합니다.

- * 계정 * — 스토리지 사용 정보를 가져오는 것을 포함하여 현재 테넌트 계정의 작업입니다.
- * auth * — 사용자 세션 인증을 수행하기 위한 작업.

  Tenant Management API는 Bearer Token Authentication Scheme을 지원합니다. 테넌트 로그인의 경우 인증 요청의 JSON 본문에 사용자 이름, 암호 및 accountId를 입력합니다(즉, 'POST/API/v3/authorize'). 사용자가 성공적으로 인증되면 보안 토큰이 반환됩니다. 이 토큰은 후속 API 요청 헤더("Authorization: Bearer token")에 제공되어야 합니다.

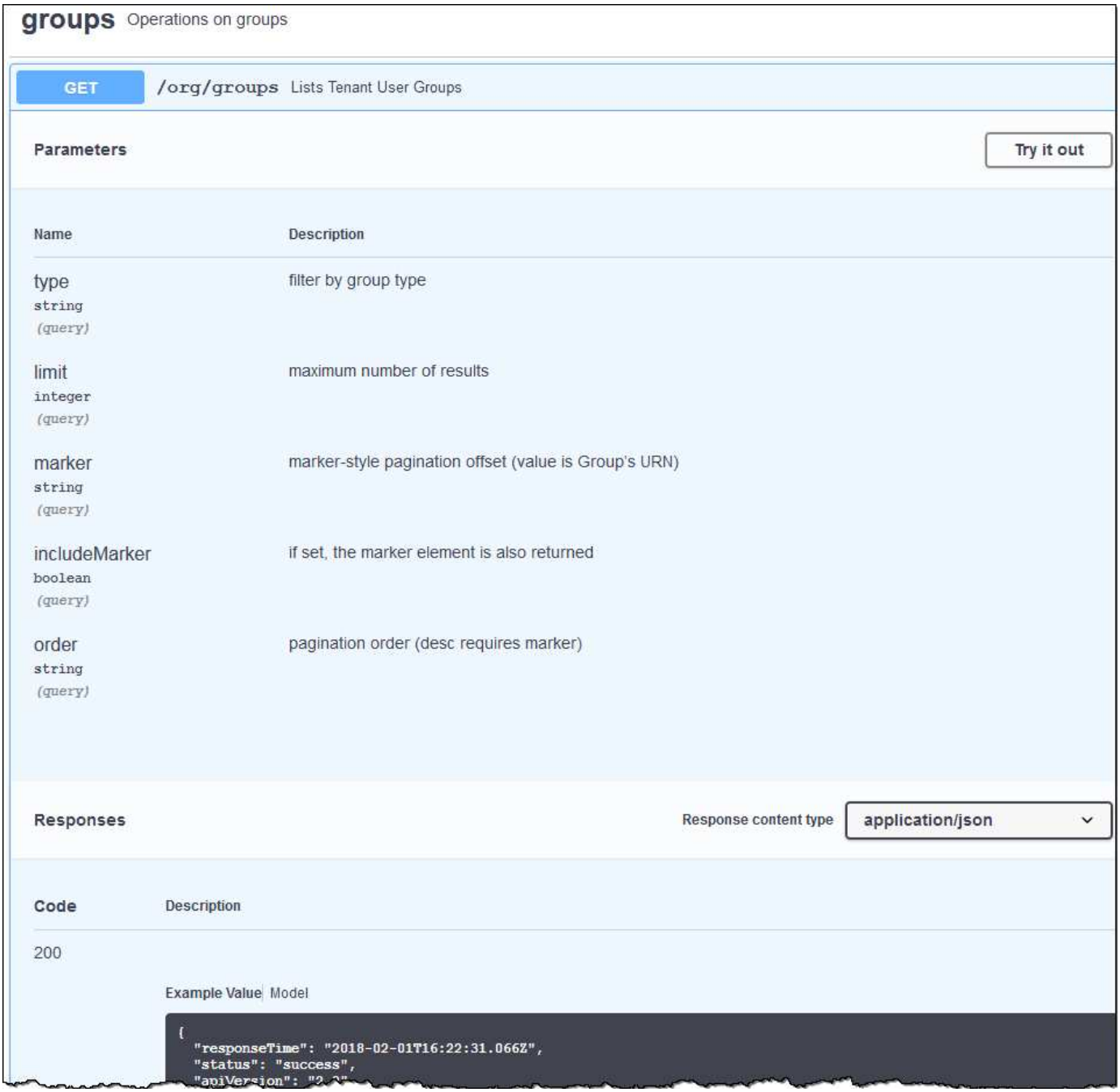
  인증 보안 개선에 대한 자세한 내용은 을 참조하십시오 사이트 간 요청 위조 방지.

  > StorageGRID 시스템에서 SSO(Single Sign-On)가 활성화된 경우 인증을 위해 다른 단계를 수행해야 합니다. 를 참조하십시오 Grid Management API 사용 지침.

- * config * — 제품 릴리스 및 테넌트 관리 API 버전과 관련된 작업. 제품 릴리스 버전과 해당 릴리스에서 지원하는 API의 주요 버전을 나열할 수 있습니다.
- * 컨테이너 * — S3 버킷 또는 Swift 컨테이너에서 다음과 같은 작업을 수행합니다.
- S3 *
  - 버킷 생성(S3 오브젝트 잠금이 활성화된 상태 및 활성화되지 않은 상태)
  - 버킷 기본 보존 수정(S3 오브젝트 잠금이 활성화된 버킷의 경우)
  - 객체에 대해 수행되는 작업에 대한 정합성 제어를 설정합니다
  - 버킷의 CORS 구성을 생성, 업데이트 및 삭제합니다
  - 객체에 대한 마지막 액세스 시간 업데이트를 설정 및 해제합니다
  - CloudMirror 복제, 알림 및 검색 통합(메타데이터 알림)을 비롯한 플랫폼 서비스에 대한 구성 설정 관리
  - 빈 버킷을 삭제합니다
- Swift *: 컨테이너에 사용되는 정합성 수준을 설정합니다
- * deactivated - features * — 비활성화된 기능을 보기 위한 작업.
- * 끝점 * — 끝점을 관리하는 작업. 엔드포인트는 S3 버킷이 StorageGRID CloudMirror 복제, 알림 또는 검색 통합에 외부 서비스를 사용할 수 있도록 합니다.
- * 그룹 * — 로컬 테넌트 그룹을 관리하고 외부 ID 소스에서 통합 테넌트 그룹을 검색하는 작업입니다.
- * identity-source * — 외부 ID 소스를 구성하고 통합 그룹 및 사용자 정보를 수동으로 동기화하는 작업.
- * 지역 * — StorageGRID 시스템에 대해 구성된 지역을 결정하는 작업.
- * S3 * — 테넌트 사용자를 위한 S3 액세스 키를 관리하는 운영
- * S3-오브젝트 잠금 * — 글로벌 S3 오브젝트 잠금 설정에서 운영, 규정 준수 지원에 사용됩니다.
- * 사용자 * — 테넌트 사용자를 보고 관리하는 작업.

작업 세부 정보

각 API 작업을 확장하면 HTTP 동작, 끝점 URL, 필수 또는 선택적 매개 변수 목록, 요청 본문(필요한 경우) 예제 및 가능한 응답을 볼 수 있습니다.



API 요청을 발행합니다

> API Docs 웹 페이지를 사용하여 수행하는 모든 API 작업은 라이브 작업입니다. 실수로 구성 데이터나 기타 데이터를 작성, 업데이트 또는 삭제하지 않도록 주의하십시오.

단계

1. 요청 세부 정보를 보려면 HTTP 작업을 선택합니다.
2. 요청에 그룹 또는 사용자 ID와 같은 추가 매개 변수가 필요한지 확인합니다. 그런 다음 이 값을 구합니다. 필요한 정보를 얻기 위해 먼저 다른 API 요청을 발급해야 할 수도 있습니다.
3. 예제 요청 본문을 수정해야 하는지 확인합니다. 이 경우 * Model * 을 선택하여 각 필드의 요구 사항을 확인할 수 있습니다.

4. 체험하기 * 를 선택합니다.
5. 필요한 매개 변수를 제공하거나 요청 본문을 필요에 따라 수정합니다.
6. Execute * 를 선택합니다.
7. 응답 코드를 검토하여 요청이 성공했는지 확인합니다.

테넌트 관리 **API** 버전 관리

테넌트 관리 API는 버전 관리를 사용하여 무중단 업그레이드를 지원합니다.

예를 들어 이 요청 URL은 API의 버전 3을 지정합니다.

```
https://hostname_or_ip_address/api/v3/authorize
```

테넌트 관리 API의 주요 버전은 이전 버전과 * _호환되지 않는 _ * 변경 사항이 있을 때 충돌합니다. 테넌트 관리 API의 부 버전은 * _ 이(가) 이전 버전과 호환된다는 변경 사항이 있을 때 충돌합니다. 호환 가능한 변경 사항에는 새 끝점 또는 새 속성 추가가 포함됩니다. 다음 예제에서는 변경 유형에 따라 API 버전을 충돌하는 방법을 보여 줍니다.

API 변경 유형입니다	이전 버전	새 버전
이전 버전과 호환 가능합니다	2.1	2.2
이전 버전과 호환되지 않습니다	2.1	3.0

StorageGRID 소프트웨어를 처음 설치하면 가장 최신 버전의 테넌트 관리 API만 활성화됩니다. 그러나 StorageGRID를 새 기능 릴리즈로 업그레이드하면 하나 이상의 StorageGRID 기능 릴리즈에 대한 이전 API 버전에 계속 액세스할 수 있습니다.

오래된 요청은 다음과 같은 방법으로 더 이상 사용되지 않는 것으로 표시됩니다.

- 응답 헤더가 "DEPRECATED:TRUE"입니다.
- JSON 응답 본문에는 "DEPRECATED"가 포함됩니다. TRUE

현재 릴리즈에서 지원되는 **API** 버전을 확인합니다

다음 API 요청을 사용하여 지원되는 API 주요 버전 목록을 반환합니다.

```
GET https://{{IP-Address}}/api/versions
{
  "responseTime": "2019-01-10T20:41:00.845Z",
  "status": "success",
  "apiVersion": "3.0",
  "data": [
    2,
    3
  ]
}
```

요청에 대한 **API** 버전을 지정합니다

PATH 파라미터('/api/v3')나 header('api-Version:3')를 이용하여 API 버전을 지정할 수 있다. 두 값을 모두 제공하면 헤더 값이 경로 값을 재정의합니다.

```
curl https://[IP-Address]/api/v3/grid/accounts

curl -H "Api-Version: 3" https://[IP-Address]/api/grid/accounts
```

사이트 간 요청 위조**(CSRF)**로부터 보호

CSRF 토큰을 사용하여 쿠키를 사용하는 인증을 강화하면 StorageGRID에 대한 CSRF(사이트 간 요청 위조) 공격으로부터 보호할 수 있습니다. Grid Manager 및 Tenant Manager는 이 보안 기능을 자동으로 활성화합니다. 다른 API 클라이언트는 로그인할 때 활성화 여부를 선택할 수 있습니다.

HTTP 양식 POST와 같이 다른 사이트에 대한 요청을 트리거할 수 있는 공격자는 로그인한 사용자의 쿠키를 사용하여 특정 요청을 만들 수 있습니다.

StorageGRID는 CSRF 토큰을 사용하여 CSRF 공격으로부터 보호합니다. 활성화된 경우 특정 쿠키의 내용은 특정 헤더 또는 특정 POST 본문 매개 변수의 내용과 일치해야 합니다.

이 기능을 활성화하려면 인증 중에 csrfToken 매개 변수를 true로 설정하십시오. 기본값은 false 입니다.

```
curl -X POST --header "Content-Type: application/json" --header "Accept:
application/json" -d "{
  \"username\": \"MyUserName\",
  \"password\": \"MyPassword\",
  \"cookie\": true,
  \"csrfToken\": true
}" "https://example.com/api/v3/authorize"
```

true이면 Grid Manager에 로그인할 때 임의의 값으로 GridCsrfToken 쿠키가 설정되고 테넌트 관리자에 로그인할 때

임의의 값으로 AccountCsrfToken 쿠키가 설정됩니다.

쿠키가 있는 경우 시스템 상태(POST, PUT, 패치, 삭제)를 수정할 수 있는 모든 요청에 다음 중 하나가 포함되어야 합니다.

- CSRF 토큰 쿠키의 값으로 설정된 헤더의 X-CSRF-Token 헤더입니다.
- 폼 인코딩된 본문을 허용하는 끝점의 경우 "csrfToken" 형식 인코딩된 요청 본문 매개 변수입니다.

CSRF 보호를 구성하려면 를 사용합니다 Grid Management API를 참조하십시오 또는 테넌트 관리 API.

> CSRF 토큰 쿠키 세트를 가진 요청은 또한 JSON 요청 본문을 CSRF 공격에 대한 추가 보호로서 기대하는 모든 요청에 대해 ""Content-Type:application/json"" 헤더를 적용합니다.

시스템 액세스를 관리합니다

ID 페더레이션을 사용합니다

ID 페더레이션을 사용하면 테넌트 그룹 및 사용자를 더 빠르게 설정할 수 있으며, 테넌트 사용자는 익숙한 자격 증명을 사용하여 테넌트 계정에 로그인할 수 있습니다.

테넌트 관리자에 대한 **ID** 페더레이션을 구성합니다

테넌트 그룹 및 사용자를 Active Directory, Azure Active Directory(Azure AD), OpenLDAP 또는 Oracle Directory Server와 같은 다른 시스템에서 관리하도록 하려면 테넌트 관리자에 대한 ID 페더레이션을 구성할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인했습니다 지원되는 웹 브라우저.
- 특정 액세스 권한이 있습니다.
- Active Directory, Azure AD, OpenLDAP 또는 Oracle Directory Server를 ID 공급자로 사용하고 있습니다.

  > 목록에 없는 LDAP v3 서비스를 사용하려면 기술 지원 부서에 문의하십시오.

- OpenLDAP를 사용하려면 OpenLDAP 서버를 구성해야 합니다. 을 참조하십시오 OpenLDAP 서버 구성 지침.
- LDAP 서버와의 통신에 TLS(Transport Layer Security)를 사용하려는 경우 ID 공급자는 TLS 1.2 또는 1.3을 사용해야 합니다. 을 참조하십시오 발신 TLS 연결에 지원되는 암호.

이 작업에 대해

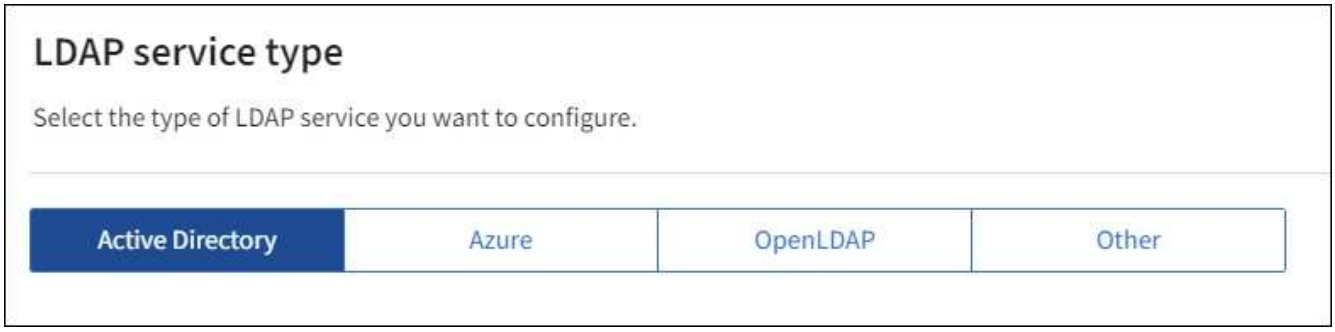
테넌트의 ID 페더레이션 서비스를 구성할 수 있는지 여부는 테넌트 계정 설정 방법에 따라 달라집니다. 테넌트가 Grid Manager용으로 구성된 ID 페더레이션 서비스를 공유할 수 있습니다. ID 페더레이션 페이지에 액세스할 때 이 메시지가 표시되면 이 테넌트에 대해 별도의 통합 ID 소스를 구성할 수 없습니다.

> This tenant account uses the LDAP server that is configured for the Grid Manager.
> Contact the grid administrator for information or to change this setting.

구성을 입력합니다

단계

1. 액세스 관리 * > * ID 페더레이션 * 을 선택합니다.
2. ID 페더레이션 사용 * 을 선택합니다.
3. LDAP 서비스 유형 섹션에서 구성할 LDAP 서비스 유형을 선택합니다.



Oracle Directory Server를 사용하는 LDAP 서버의 값을 구성하려면 * 기타 * 를 선택합니다.

4. 기타 * 를 선택한 경우 LDAP 속성 섹션의 필드를 작성합니다. 그렇지 않으면 다음 단계로 이동합니다.
   - * 사용자 고유 이름 *: LDAP 사용자의 고유 식별자가 포함된 속성의 이름입니다. 이 속성은 Active Directory의 경우 'AMAccountName', OpenLDAP의 경우 'uid'와 같습니다. Oracle Directory Server를 구성하는 경우 "uid"를 입력합니다.
   - * 사용자 UUID *: LDAP 사용자의 영구 고유 식별자가 포함된 특성의 이름입니다. 이 속성은 Active Directory의 경우 objectGUID, OpenLDAP의 경우 entryUUID와 같습니다. Oracle Directory Server를 구성하는 경우 "n스uniqueid"를 입력합니다. 지정된 속성에 대한 각 사용자의 값은 16바이트 또는 문자열 형식의 32자리 16진수 숫자여야 하며, 하이픈은 무시됩니다.
   - * 그룹 고유 이름 *: LDAP 그룹의 고유 식별자가 포함된 속성의 이름입니다. 이 속성은 Active Directory의 경우 'AMAccountName', OpenLDAP의 경우 'cn'과 같습니다. Oracle Directory Server를 구성하는 경우 cn을 입력합니다.
   - * 그룹 UUID *: LDAP 그룹의 영구 고유 식별자가 포함된 특성의 이름입니다. 이 속성은 Active Directory의 경우 objectGUID, OpenLDAP의 경우 entryUUID와 같습니다. Oracle Directory Server를 구성하는 경우 "n스uniqueid"를 입력합니다. 지정된 속성에 대한 각 그룹의 값은 16바이트 또는 문자열 형식의 32자리 16진수 숫자여야 하며, 하이픈은 무시됩니다.
5. 모든 LDAP 서비스 유형에 대해 LDAP 서버 구성 섹션에 필요한 LDAP 서버 및 네트워크 연결 정보를 입력합니다.
   - * 호스트 이름 *: LDAP 서버의 FQDN(정규화된 도메인 이름) 또는 IP 주소입니다.
   - * 포트 *: LDAP 서버에 연결하는 데 사용되는 포트입니다.

     STARTTLS의 기본 포트는 389이고 LDAPS의 기본 포트는 636입니다. 그러나 방화벽이 올바르게 구성된 경우 모든 포트를 사용할 수 있습니다.

   - * 사용자 이름 *: LDAP 서버에 연결할 사용자의 DN(고유 이름)의 전체 경로입니다.

     Active Directory의 경우 아래쪽 로그온 이름 또는 사용자 기본 이름을 지정할 수도 있습니다.

     지정된 사용자는 그룹 및 사용자를 나열하고 다음 속성에 액세스할 수 있는 권한이 있어야 합니다.

     - '사계정 이름' 또는 'uid'
     - objectGUID, entryUUID, n스uniqueid

        - 'cn'입니다
        - 'emberOf' 또는 'isMemberOf'
        - Active Directory *: objectSid, primaryGroupID, userAccountControl, userPrincipalName
        - * Azure *:'accountEnabled' 및 'userPrincipalName'
    - * 암호 *: 사용자 이름과 연결된 암호입니다.
    - * Group Base DN *: 그룹을 검색할 LDAP 하위 트리에 대한 DN(고유 이름)의 전체 경로입니다. Active Directory 예제(아래)에서 고유 이름이 기본 DN(DC=StorageGrid, DC=example, DC=com)과 관련된 모든 그룹을 통합 그룹으로 사용할 수 있습니다.

      > 그룹 고유 이름 * 값은 * 그룹 기본 DN * 내에서 고유해야 합니다.

    - * 사용자 기본 DN *: 사용자를 검색할 LDAP 하위 트리의 고유 이름(DN)의 전체 경로입니다.

      > 사용자 고유 이름 * 값은 * 사용자 기본 DN * 내에서 고유해야 합니다.

    - * 사용자 이름 형식 바인딩 * (선택 사항): 패턴을 자동으로 확인할 수 없는 경우 StorageGRID에서 기본 사용자 이름 패턴을 사용해야 합니다.

      StorageGRID가 서비스 계정에 바인딩할 수 없는 경우 사용자가 로그인할 수 있으므로 * 사용자 이름 형식 바인딩 * 을 제공하는 것이 좋습니다.

      다음 패턴 중 하나를 입력합니다.

        - * UserPrincipalName 패턴(Active Directory 및 Azure) *: '[UserName]@*example*.com'
        - * 하위 수준 로그온 이름 패턴(Active Directory 및 Azure) *: `*example*\[사용자 이름]'
        - * 고유 이름 패턴 *:'CN=[UserName],CN=Users,DC=*Example*,DC=com'

          [UserName] * 을 서면 그대로 포함합니다.

6. TLS(전송 계층 보안) 섹션에서 보안 설정을 선택합니다.
    - * STARTTLS 사용 *: STARTTLS를 사용하여 LDAP 서버와의 통신 보안을 설정합니다. 이 옵션은 Active Directory, OpenLDAP 또는 기타 에 대해 권장되지만 Azure에서는 지원되지 않습니다.
    - * LDAPS * 사용: LDAPS(LDAP over SSL) 옵션은 TLS를 사용하여 LDAP 서버에 연결합니다. Azure의 경우 이 옵션을 선택해야 합니다.
    - * TLS * 사용 안 함: StorageGRID 시스템과 LDAP 서버 간의 네트워크 트래픽은 보호되지 않습니다. 이 옵션은 Azure에서 지원되지 않습니다.

      > Active Directory 서버가 LDAP 서명을 적용하는 경우 * TLS 사용 안 함 * 옵션을 사용할 수 없습니다. STARTTLS 또는 LDAPS를 사용해야 합니다.

7. STARTTLS 또는 LDAPS를 선택한 경우 연결 보안에 사용되는 인증서를 선택합니다.
    - * 운영 체제 CA 인증서 사용 *: 운영 체제에 설치된 기본 그리드 CA 인증서를 사용하여 연결을 보호합니다.
    - * 사용자 지정 CA 인증서 사용 *: 사용자 지정 보안 인증서를 사용합니다.

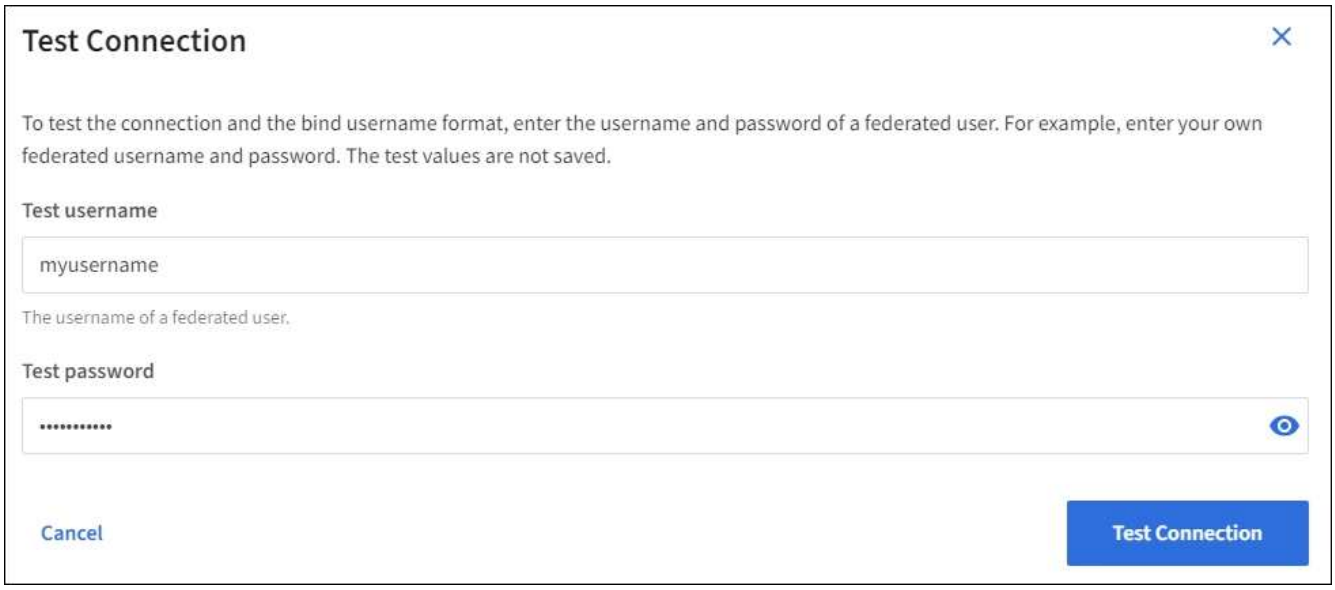
      이 설정을 선택한 경우 사용자 지정 보안 인증서를 복사하여 CA 인증서 텍스트 상자에 붙여 넣습니다.

연결을 테스트하고 구성을 저장합니다

모든 값을 입력한 후 구성을 저장하기 전에 연결을 테스트해야 합니다. StorageGRID는 LDAP 서버에 대한 연결 설정과 바인딩 사용자 이름 형식(제공한 경우)을 확인합니다.

1. Test connection * 을 선택합니다.
2. 바인딩 사용자 이름 형식을 제공하지 않은 경우:
   - 연결 설정이 유효하면 "Test connection successful(연결 테스트 성공)" 메시지가 나타납니다. Save * 를 선택하여 설정을 저장합니다.
   - 연결 설정이 잘못된 경우 ""테스트 연결을 설정할 수 없습니다"" 메시지가 나타납니다. 닫기 * 를 선택합니다. 그런 다음 문제를 해결하고 연결을 다시 테스트합니다.
3. 바인딩 사용자 이름 형식을 제공한 경우 유효한 통합 사용자의 사용자 이름과 암호를 입력합니다.

   예를 들어 사용자 이름과 암호를 입력합니다. @ 또는 / 같은 특수 문자를 사용자 이름에 포함하지 마십시오.



   - 연결 설정이 유효하면 "Test connection successful(연결 테스트 성공)" 메시지가 나타납니다. Save * 를 선택하여 설정을 저장합니다.
   - 연결 설정, 바인딩 사용자 이름 형식 또는 테스트 사용자 이름과 암호가 올바르지 않으면 오류 메시지가 나타납니다. 모든 문제를 해결하고 연결을 다시 테스트합니다.

ID 소스와 강제로 동기화합니다

StorageGRID 시스템은 ID 소스에서 페더레이션 그룹과 사용자를 정기적으로 동기화합니다. 사용자 권한을 최대한 빨리 설정하거나 제한하려는 경우 동기화를 강제로 시작할 수 있습니다.

단계

1. ID 페더레이션 페이지로 이동합니다.
2. 페이지 맨 위에서 * 서버 동기화 * 를 선택합니다.

   동기화 프로세스는 환경에 따라 다소 시간이 걸릴 수 있습니다.

ID 소스에서 페더레이션 그룹과 사용자를 동기화하는 데 문제가 있는 경우 * ID 페더레이션 동기화 실패 * 경고가 트리거됩니다.

ID 페더레이션을 비활성화합니다

그룹 및 사용자에 대한 ID 페더레이션을 일시적으로 또는 영구적으로 비활성화할 수 있습니다. ID 페더레이션을 사용하지 않도록 설정하면 StorageGRID와 ID 소스 간에 통신이 이루어지지 않습니다. 그러나 구성한 설정은 그대로 유지되므로 나중에 ID 페더레이션을 쉽게 다시 사용할 수 있습니다.

이 작업에 대해

ID 페더레이션을 사용하지 않도록 설정하기 전에 다음 사항을 확인해야 합니다.

- 페더레이션 사용자는 로그인할 수 없습니다.
- 현재 로그인한 페더레이션 사용자는 세션이 만료될 때까지 StorageGRID 시스템에 대한 액세스 권한을 유지하지만 세션이 만료된 후에는 로그인할 수 없습니다.
- StorageGRID 시스템과 ID 소스 간의 동기화는 수행되지 않으며 동기화되지 않은 계정에 대해 알림 또는 경보가 발생하지 않습니다.
- SSO(Single Sign-On)가 * Enabled * 또는 * Sandbox Mode * 로 설정된 경우 * Enable identity federation *(ID 페더레이션 사용 *) 확인란이 비활성화됩니다. ID 페더레이션을 비활성화하려면 Single Sign-On 페이지의 SSO 상태가 * 사용 안 함 * 이어야 합니다. 을 참조하십시오 SSO(Single Sign-On)를 비활성화합니다.

단계

1. ID 페더레이션 페이지로 이동합니다.
2. ID 페더레이션 사용 * 확인란의 선택을 취소합니다.

OpenLDAP 서버 구성 지침

OpenLDAP 서버를 ID 페더레이션에 사용하려면 OpenLDAP 서버에서 특정 설정을 구성해야 합니다.

ActiveDirectory 또는 Azure가 아닌 ID 소스의 경우 StorageGRID는 외부에서 비활성화된 사용자에 대한 S3 액세스를 자동으로 차단하지 않습니다. S3 액세스를 차단하려면 사용자의 S3 키를 삭제하고 모든 그룹에서 사용자를 제거합니다.

MemberOf 및 구체화 오버레이

MemberOf 및 구체화 오버레이를 활성화해야 합니다. 자세한 내용은 의 역방향 그룹 구성원 유지 관리 지침을 참조하십시오http://www.openldap.org/doc/admin24/index.html["OpenLDAP 설명서: 버전 2.4 관리자 가이드"^].

인덱싱

지정된 인덱스 키워드를 사용하여 다음 OpenLDAP 속성을 구성해야 합니다.

- olcDbIndex:objectClass eq
- "olcDbIndex:uid eq,pres,sub'
- 올크DbIndex=cn eq,pres,sub
- olcDbIndex: entryUUID eq

또한 최적의 성능을 위해 사용자 이름 도움말에 언급된 필드를 인덱싱해야 합니다.

에서 역방향 그룹 구성원 유지 관리에 대한 정보를
참조하십시오http://www.openldap.org/doc/admin24/index.html["OpenLDAP 설명서: 버전 2.4 관리자 가이드"^].

그룹을 관리합니다

S3 테넌트에 대한 그룹을 생성합니다

통합 그룹을 가져오거나 로컬 그룹을 생성하여 S3 사용자 그룹에 대한 권한을 관리할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.
- 통합 그룹을 가져오려는 경우 ID 페더레이션을 구성하고 통합 그룹이 이미 구성된 ID 소스에 있습니다.

S3에 대한 자세한 내용은 을 참조하십시오 S3을 사용합니다.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.

2. Create group * 을 선택합니다.
3. 로컬 그룹을 생성하려면 * Local group * 탭을 선택하고, 이전에 구성된 ID 소스에서 그룹을 가져오려면 * Federated group * 탭을 선택합니다.

   StorageGRID 시스템에서 SSO(Single Sign-On)를 사용하는 경우 로컬 그룹에 속한 사용자는 그룹 권한에 따라 클라이언트 애플리케이션을 사용하여 테넌트의 리소스를 관리할 수 있지만 테넌트 관리자에 로그인할 수 없습니다.

4. 그룹의 이름을 입력합니다.

- * 로컬 그룹 *: 표시 이름과 고유 이름을 모두 입력합니다. 나중에 표시 이름을 편집할 수 있습니다.
- * 페더레이션 그룹 *: 고유한 이름을 입력합니다. Active Directory의 경우 고유한 이름은 'AMAccountName' 속성과 관련된 이름입니다. OpenLDAP의 경우 고유 이름은 "uid" 특성과 관련된 이름입니다.

5. Continue * 를 선택합니다.
6. 액세스 모드를 선택합니다. 사용자가 여러 그룹에 속해 있고 모든 그룹이 읽기 전용으로 설정되어 있는 경우 사용자는 선택한 모든 설정 및 기능에 대해 읽기 전용 권한을 갖게 됩니다.
   - * 읽기-쓰기 * (기본값): 사용자는 테넌트 관리자에 로그인하여 테넌트 구성을 관리할 수 있습니다.
   - * 읽기 전용 *: 사용자는 설정 및 기능만 볼 수 있습니다. 테넌트 관리자 또는 테넌트 관리 API에서 작업을 변경하거나 수행할 수 없습니다. 로컬 읽기 전용 사용자는 자신의 암호를 변경할 수 있습니다.
7. 이 그룹에 대한 그룹 권한을 선택합니다.

   테넌트 관리 권한에 대한 정보를 참조하십시오.
8. Continue * 를 선택합니다.
9. 그룹 정책을 선택하여 이 그룹의 구성원이 가질 S3 액세스 권한을 결정합니다.
   - * S3 액세스 없음 *: 기본값. 이 그룹의 사용자는 버킷 정책을 통해 액세스가 부여되지 않는 한 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 문자열은 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 문자열은 편집할 수 없습니다.
   - * 사용자 정의 *: 그룹의 사용자에게는 텍스트 상자에 지정한 사용 권한이 부여됩니다. 언어 구문 및 예제를 비롯한 그룹 정책에 대한 자세한 내용은 S3 클라이언트 애플리케이션 구현 지침을 참조하십시오.
10. 사용자 정의 * 를 선택한 경우 그룹 정책을 입력합니다. 각 그룹 정책은 크기 제한이 5,120바이트입니다. 올바른 JSON 형식 문자열을 입력해야 합니다.

    이 예제에서 그룹 구성원은 지정된 버킷의 사용자 이름(키 접두사)과 일치하는 폴더만 나열하고 액세스할 수 있습니다. 이러한 폴더의 개인 정보를 확인할 때는 다른 그룹 정책 및 버킷 정책의 액세스 권한을 고려해야 합니다.

11. 통합 그룹을 생성하는지 또는 로컬 그룹을 생성하는지에 따라 표시되는 버튼을 선택합니다.

    - 통합 그룹: * 그룹 생성 *
    - 로컬 그룹: * 계속 *

    로컬 그룹을 만드는 경우 * Continue * 를 선택하면 4단계(사용자 추가)가 나타납니다. 이 단계는 통합 그룹에 대해서는 나타나지 않습니다.

12. 그룹에 추가할 각 사용자에 대한 확인란을 선택한 다음 * 그룹 생성 * 을 선택합니다.

    필요에 따라 사용자를 추가하지 않고 그룹을 저장할 수 있습니다. 나중에 그룹에 사용자를 추가하거나 새 사용자를 추가할 때 그룹을 선택할 수 있습니다.

13. 마침 * 을 선택합니다.

    생성한 그룹이 그룹 목록에 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

Swift 테넌트의 그룹을 생성합니다

통합 그룹을 가져오거나 로컬 그룹을 생성하여 Swift 테넌트 계정에 대한 액세스 권한을 관리할 수 있습니다. 하나 이상의 그룹에 Swift 관리자 권한이 있어야 합니다. 이 권한은 Swift 테넌트 계정의 컨테이너 및 개체를 관리하는 데 필요합니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다.

- 통합 그룹을 가져오려는 경우 ID 페더레이션을 구성하고 통합 그룹이 이미 구성된 ID 소스에 있습니다.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.

2. Create group * 을 선택합니다.
3. 로컬 그룹을 생성하려면 * Local group * 탭을 선택하고, 이전에 구성된 ID 소스에서 그룹을 가져오려면 * Federated group * 탭을 선택합니다.

   StorageGRID 시스템에서 SSO(Single Sign-On)를 사용하는 경우 로컬 그룹에 속한 사용자는 그룹 권한에 따라 클라이언트 애플리케이션을 사용하여 테넌트의 리소스를 관리할 수 있지만 테넌트 관리자에 로그인할 수 없습니다.

4. 그룹의 이름을 입력합니다.
   - * 로컬 그룹 *: 표시 이름과 고유 이름을 모두 입력합니다. 나중에 표시 이름을 편집할 수 있습니다.
   - * 페더레이션 그룹 *: 고유한 이름을 입력합니다. Active Directory의 경우 고유한 이름은 'AMAccountName' 속성과 관련된 이름입니다. OpenLDAP의 경우 고유 이름은 "uid" 특성과 관련된 이름입니다.
5. Continue * 를 선택합니다.
6. 액세스 모드를 선택합니다. 사용자가 여러 그룹에 속해 있고 모든 그룹이 읽기 전용으로 설정되어 있는 경우 사용자는 선택한 모든 설정 및 기능에 대해 읽기 전용 권한을 갖게 됩니다.
   - * 읽기-쓰기 * (기본값): 사용자는 테넌트 관리자에 로그인하여 테넌트 구성을 관리할 수 있습니다.
   - * 읽기 전용 *: 사용자는 설정 및 기능만 볼 수 있습니다. 테넌트 관리자 또는 테넌트 관리 API에서 작업을 변경하거나 수행할 수 없습니다. 로컬 읽기 전용 사용자는 자신의 암호를 변경할 수 있습니다.
7. 그룹 권한을 설정합니다.
   - 사용자가 테넌트 관리자 또는 테넌트 관리 API에 로그인해야 하는 경우 * Root Access * 확인란을 선택합니다. (기본값)
   - 사용자가 테넌트 관리자 또는 테넌트 관리 API에 액세스할 필요가 없는 경우 * Root Access * (루트 액세스 *) 확인란의 선택을 취소합니다. 예를 들어, 테넌트에 액세스할 필요가 없는 응용 프로그램의 확인란을 선택

취소합니다. 그런 다음 이러한 사용자가 컨테이너 및 개체를 관리할 수 있도록 * Swift 관리자 * 권한을 할당합니다.

8. Continue * 를 선택합니다.
9. 사용자가 Swift REST API를 사용할 수 있어야 하는 경우 * Swift administrator * 확인란을 선택합니다.

   Swift 사용자는 테넌트 관리자에 액세스하려면 루트 액세스 권한이 있어야 합니다. 그러나 루트 액세스 권한은 사용자가 Swift REST API에 인증하여 컨테이너를 생성하고 객체를 수집하는 것을 허용하지 않습니다. 사용자는 Swift REST API에 인증할 수 있는 Swift 관리자 권한이 있어야 합니다.

10. 통합 그룹을 생성하는지 또는 로컬 그룹을 생성하는지에 따라 표시되는 버튼을 선택합니다.
    - 통합 그룹: * 그룹 생성 *
    - 로컬 그룹: * 계속 *

      로컬 그룹을 만드는 경우 * Continue * 를 선택하면 4단계(사용자 추가)가 나타납니다. 이 단계는 통합 그룹에 대해서는 나타나지 않습니다.

11. 그룹에 추가할 각 사용자에 대한 확인란을 선택한 다음 * 그룹 생성 * 을 선택합니다.

    필요에 따라 사용자를 추가하지 않고 그룹을 저장할 수 있습니다. 나중에 그룹에 사용자를 추가하거나 새 사용자를 만들 때 그룹을 선택할 수 있습니다.

12. 마침 * 을 선택합니다.

    생성한 그룹이 그룹 목록에 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

관련 정보

테넌트 관리 권한

Swift를 사용합니다

테넌트 관리 권한

테넌트 그룹을 생성하기 전에 해당 그룹에 할당할 권한을 고려하십시오. 테넌트 관리 권한은 사용자가 테넌트 관리자 또는 테넌트 관리 API를 사용하여 수행할 수 있는 작업을 결정합니다. 사용자는 하나 이상의 그룹에 속할 수 있습니다. 사용자가 여러 그룹에 속한 경우 권한은 누적됩니다.

테넌트 관리자에 로그인하거나 테넌트 관리 API를 사용하려면 사용자가 하나 이상의 권한이 있는 그룹에 속해야 합니다. 로그인할 수 있는 모든 사용자는 다음 작업을 수행할 수 있습니다.

- 대시보드 보기
- 자신의 암호 변경(로컬 사용자의 경우)

모든 권한에 대해 그룹의 액세스 모드 설정은 사용자가 설정을 변경하고 작업을 수행할 수 있는지 또는 관련 설정과 기능만 볼 수 있는지 여부를 결정합니다.

> 사용자가 여러 그룹에 속해 있고 모든 그룹이 읽기 전용으로 설정되어 있는 경우 사용자는 선택한 모든 설정 및 기능에 대해 읽기 전용 권한을 갖게 됩니다.

그룹에 다음 권한을 할당할 수 있습니다. S3 테넌트와 Swift 테넌트는 다른 그룹 권한을 가집니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

권한	설명
루트 액세스	테넌트 관리자 및 테넌트 관리 API에 대한 전체 액세스를 제공합니다. • 참고: * Swift 사용자는 테넌트 계정에 로그인하려면 루트 액세스 권한이 있어야 합니다.
관리자	Swift 테넌트만 해당. 이 테넌트 계정에 대한 Swift 컨테이너 및 객체에 대한 전체 액세스를 제공합니다 • 참고: * Swift 사용자는 Swift REST API를 사용하여 모든 작업을 수행하려면 Swift 관리자 권한이 있어야 합니다.
자신의 S3 자격 증명을 관리합니다	S3 테넌트만 해당. 사용자가 자신의 S3 액세스 키를 생성하고 제거할 수 있습니다. 이 권한이 없는 사용자는 * storage(S3) * > * My S3 access keys * 메뉴 옵션을 볼 수 없습니다.
모든 버킷 관리	• S3 테넌트: 사용자가 테넌트 관리자 및 테넌트 관리 API를 사용하여 S3 버킷을 생성 및 삭제하고 S3 버킷 또는 그룹 정책에 관계없이 테넌트 계정의 모든 S3 버킷을 관리할 수 있습니다. 이 권한이 없는 사용자는 * Bucket * 메뉴 옵션을 볼 수 없습니다. • Swift 테넌트: Swift 사용자가 테넌트 관리 API를 사용하여 Swift 컨테이너의 정합성 수준을 제어할 수 있습니다. • 참고: * 테넌트 관리 API에서 Swift 그룹에만 모든 버킷 관리 권한을 할당할 수 있습니다. 테넌트 관리자를 사용하여 Swift 그룹에 이 권한을 할당할 수 없습니다.
엔드포인트 관리	S3 테넌트만 해당. 테넌트 관리자 또는 테넌트 관리 API를 사용하여 StorageGRID 플랫폼 서비스의 대상으로 사용되는 엔드포인트를 생성하거나 편집할 수 있습니다. 이 권한이 없는 사용자는 * 플랫폼 서비스 끝점 * 메뉴 옵션을 볼 수 없습니다.

관련 정보

S3을 사용합니다

Swift를 사용합니다

그룹 세부 정보를 보고 편집합니다

그룹의 세부 정보를 볼 때 그룹의 표시 이름, 사용 권한, 정책 및 그룹에 속한 사용자를 변경할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.

- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.
2. 세부 정보를 보거나 편집할 그룹의 이름을 선택합니다.

   또는 * Actions * > * View group details * 를 선택할 수 있습니다.

   그룹 세부 정보 페이지가 나타납니다. 다음 예에서는 S3 그룹 세부 정보 페이지를 보여 줍니다.

Groups > Applications

Overview

Display name:	**Applications**
Unique name:	**group/Applications**
Type:	**Local**
Access mode:	**Read-write**
Permissions:	**Root Access**
S3 Policy:	**None**
Number of users in this group:	**0**

Group permissions | S3 group policy | Users

Manage group permissions

Select an access mode for this group and select one or more permissions.

Access mode

Select whether users can change settings and perform operations or whether they can only view settings and features.

(●) Read-write ( ) Read-only

Group permissions

Select the tenant account permissions you want to assign to this group.

[x] **Root Access**

Allows users to access all Tenant Manager features. Root Access permission supersedes all other permissions.

[x] **Manage All Buckets**

Allows users to change settings of all S3 buckets (or Swift containers) in this account.

[x] **Manage Endpoints**

Allows users to configure endpoints for platform services.

[x] **Manage Your Own S3 Credentials**

Allows users to create and delete their own S3 access keys.

Save changes

3. 필요에 따라 그룹 설정을 변경합니다.

> 변경 내용을 저장하려면 각 섹션을 변경한 후 * 변경 사항 저장 * 을 선택합니다. 변경 내용이 저장되면 페이지의 오른쪽 상단에 확인 메시지가 나타납니다.

   a. 선택적으로 표시 이름 또는 편집 아이콘을 선택합니다 표시 이름을 업데이트합니다.

      그룹의 고유한 이름은 변경할 수 없습니다. 통합 그룹의 표시 이름은 편집할 수 없습니다.

   b. 필요에 따라 사용 권한을 업데이트합니다.

   c. 그룹 정책의 경우 S3 또는 Swift 테넌트를 적절하게 변경합니다.

      - S3 테넌트의 그룹을 편집하는 경우 선택적으로 다른 S3 그룹 정책을 선택합니다. 사용자 지정 S3 정책을 선택한 경우 필요에 따라 JSON 문자열을 업데이트합니다.
      - Swift 테넌트의 그룹을 편집하는 경우, 필요에 따라 * Swift 관리자 * 확인란을 선택하거나 선택 취소합니다.

        Swift 관리자 권한에 대한 자세한 내용은 Swift 테넌트에 대한 그룹 생성 지침을 참조하십시오.

   d. 필요에 따라 사용자를 추가 또는 제거합니다.

4. 변경한 각 섹션에 대해 * 변경 사항 저장 * 을 선택했는지 확인합니다.

   캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

관련 정보

S3 테넌트용 그룹을 생성합니다

Swift 테넌트용 그룹을 생성합니다

로컬 그룹에 사용자를 추가합니다

필요에 따라 로컬 그룹에 사용자를 추가할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.
2. 사용자를 추가할 로컬 그룹의 이름을 선택합니다.

   또는 * Actions * > * View group details * 를 선택할 수 있습니다.

   그룹 세부 정보 페이지가 나타납니다.

Groups > Applications

Overview

Display name:	**Applications**
Unique name:	**group/Applications**
Type:	**Local**
Access mode:	**Read-write**
Permissions:	**Root Access**
S3 Policy:	**None**
Number of users in this group:	**0**

Group permissions | S3 group policy | Users

Manage group permissions

Select an access mode for this group and select one or more permissions.

Access mode

Select whether users can change settings and perform operations or whether they can only view settings and features.

(●) Read-write ( ) Read-only

Group permissions

Select the tenant account permissions you want to assign to this group.

[x] **Root Access**

Allows users to access all Tenant Manager features. Root Access permission supersedes all other permissions.

[x] **Manage All Buckets**

Allows users to change settings of all S3 buckets (or Swift containers) in this account.

[x] **Manage Endpoints**

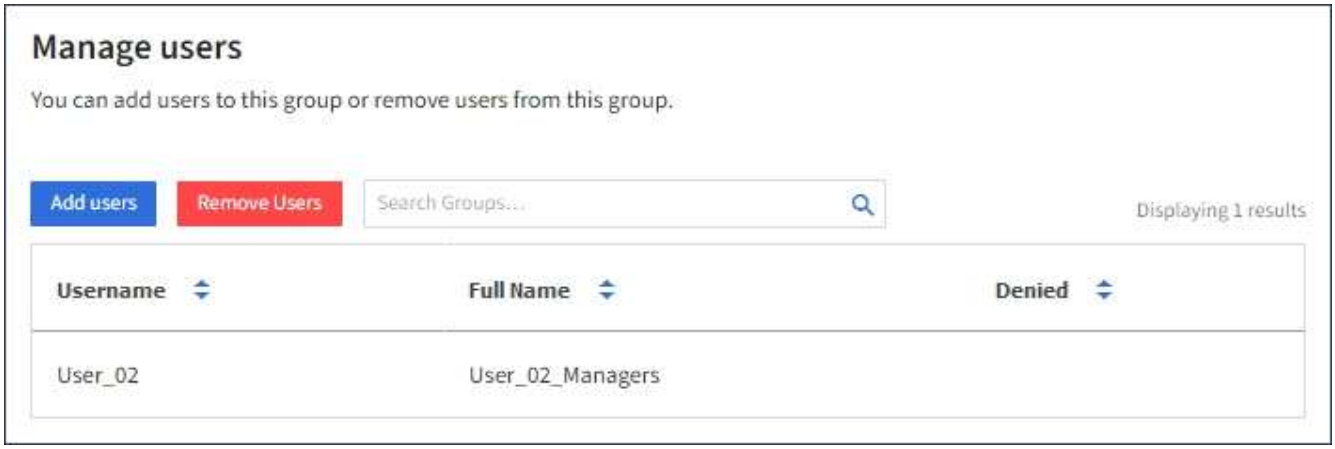
Allows users to configure endpoints for platform services.

[x] **Manage Your Own S3 Credentials**

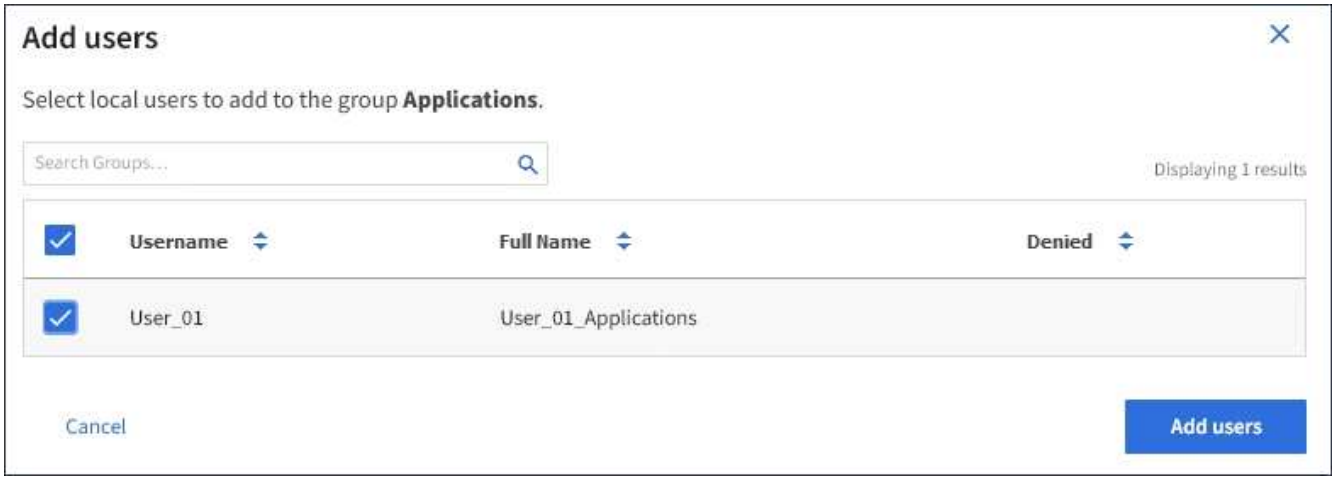
Allows users to create and delete their own S3 access keys.

Save changes

3. 사용자 * 를 선택한 다음 * 사용자 추가 * 를 선택합니다.



4. 그룹에 추가할 사용자를 선택한 다음 * 사용자 추가 * 를 선택합니다.



페이지의 오른쪽 상단에 확인 메시지가 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

그룹 이름을 편집합니다

그룹의 표시 이름을 편집할 수 있습니다. 그룹의 고유한 이름은 편집할 수 없습니다.

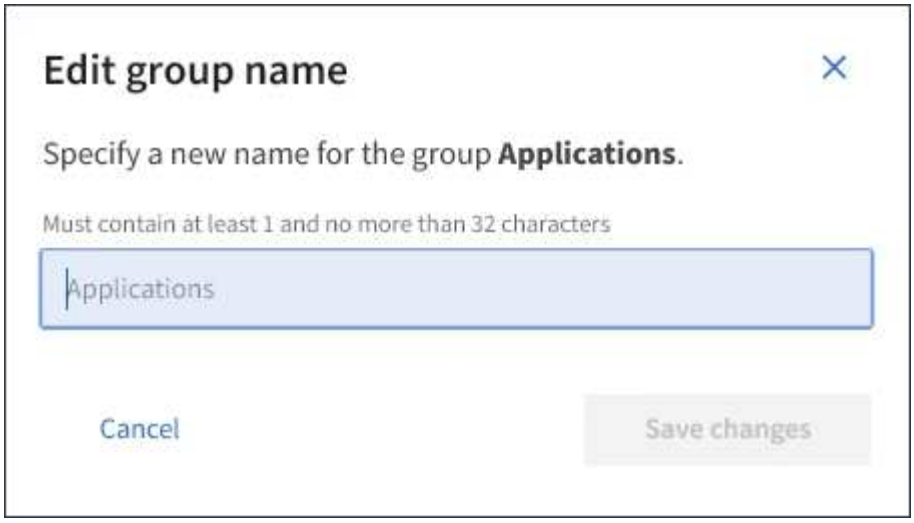
필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.
2. 표시 이름을 편집할 그룹의 확인란을 선택합니다.
3. Actions * > * Edit group name * 을 선택합니다.

   Edit group name(그룹 이름 편집) 대화 상자가 나타납니다.



4. 로컬 그룹을 편집하는 경우 필요에 따라 표시 이름을 업데이트합니다.

   그룹의 고유한 이름은 변경할 수 없습니다. 통합 그룹의 표시 이름은 편집할 수 없습니다.

5. 변경 내용 저장 * 을 선택합니다.

   페이지의 오른쪽 상단에 확인 메시지가 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

그룹이 중복되었습니다

기존 그룹을 복제하면 새 그룹을 더 빠르게 만들 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.
2. 복제할 그룹의 확인란을 선택합니다.
3. Duplicate group * 을 선택합니다. 그룹 만들기에 대한 자세한 내용은 에 대한 그룹 만들기 지침을 참조하십시오 S3 테넌트 또는 을(를) 선택합니다 Swift 테넌트입니다.
4. 로컬 그룹을 생성하려면 * Local group * 탭을 선택하고, 이전에 구성된 ID 소스에서 그룹을 가져오려면 * Federated group * 탭을 선택합니다.

   StorageGRID 시스템에서 SSO(Single Sign-On)가 활성화된 경우 로컬 그룹에 속하는 사용자는 클라이언트 애플리케이션을 사용하여 테넌트의 리소스를 관리할 수 있지만 테넌트 관리자에 로그인할 수 없습니다. 그룹 권한에 따라 다릅니다.

5. 그룹의 이름을 입력합니다.
   - * 로컬 그룹 *: 표시 이름과 고유 이름을 모두 입력합니다. 나중에 표시 이름을 편집할 수 있습니다.
   - * 페더레이션 그룹 *: 고유한 이름을 입력합니다. Active Directory의 경우 고유한 이름은 'AMAccountName' 속성과 관련된 이름입니다. OpenLDAP의 경우 고유 이름은 "uid" 특성과 관련된 이름입니다.

6. Continue * 를 선택합니다.
7. 필요에 따라 이 그룹에 대한 권한을 수정합니다.
8. Continue * 를 선택합니다.
9. 필요에 따라 S3 테넌트에 대한 그룹을 복제할 경우 * S3 정책 추가 * 라디오 버튼에서 다른 정책을 선택할 수도 있습니다. 사용자 지정 정책을 선택한 경우 필요에 따라 JSON 문자열을 업데이트합니다.
10. Create group * 을 선택합니다.

그룹을 삭제합니다

시스템에서 그룹을 삭제할 수 있습니다. 해당 그룹에만 속하는 사용자는 더 이상 테넌트 관리자에 로그인하거나 테넌트 계정을 사용할 수 없습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.

단계

1. 액세스 관리 * > * 그룹 * 을 선택합니다.

2. 삭제할 그룹의 확인란을 선택합니다.
3. Actions * > * Delete group * 을 선택합니다.

   확인 메시지가 나타납니다.

4. 확인 메시지에 표시된 그룹을 삭제하려면 * Delete group * 을 선택합니다.

   페이지의 오른쪽 상단에 확인 메시지가 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

로컬 사용자를 관리합니다

로컬 사용자를 만들고 로컬 그룹에 할당하여 사용자가 액세스할 수 있는 기능을 결정할 수 있습니다. Tenant Manager에는 ""root""라는 이름의 미리 정의된 로컬 사용자가 한 명 있습니다. 로컬 사용자를 추가 및 제거할 수는 있지만 루트 사용자는 제거할 수 없습니다.

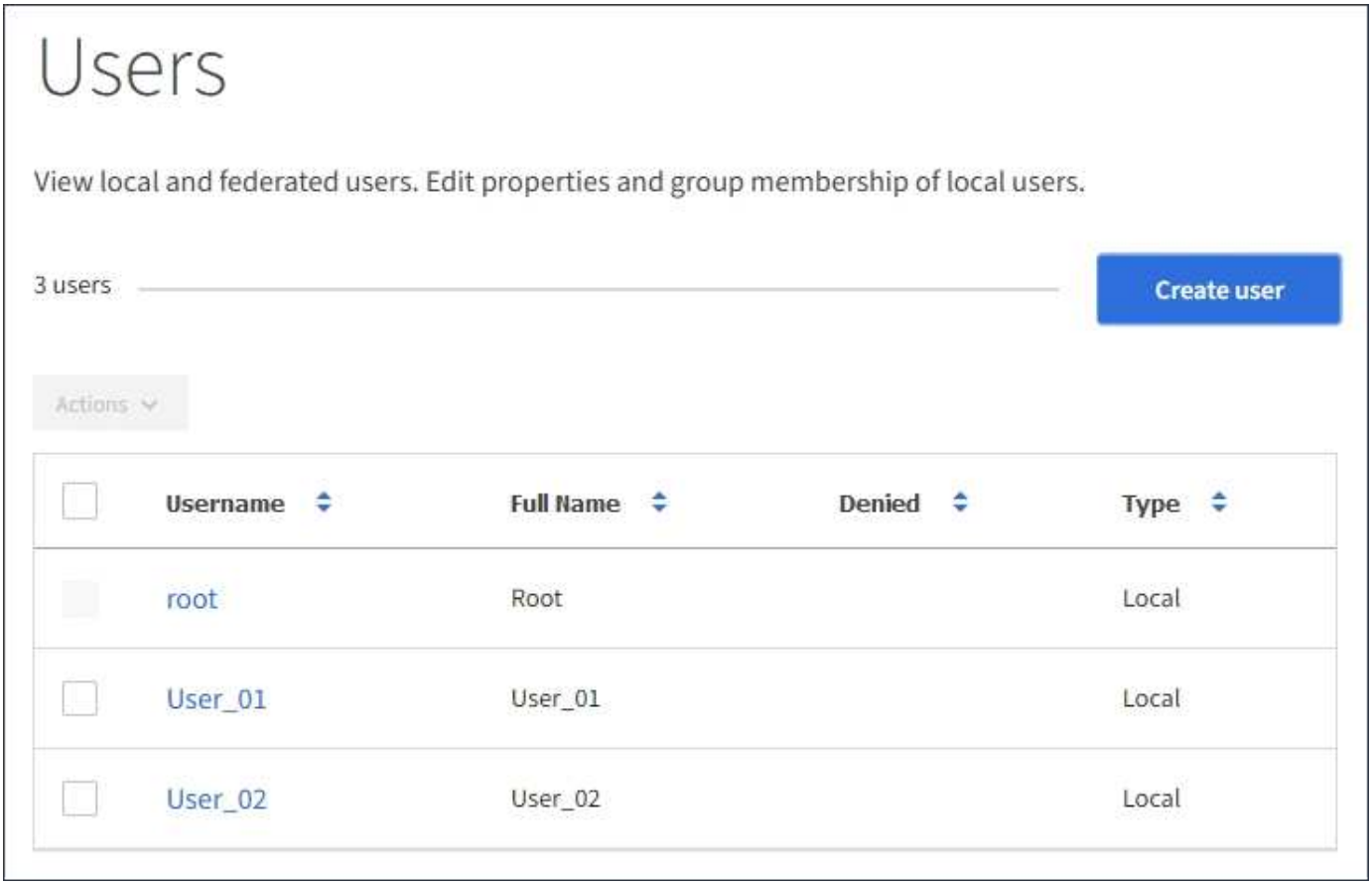
필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있는 읽기-쓰기 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.

> StorageGRID 시스템에서 SSO(Single Sign-On)가 활성화된 경우 그룹 사용 권한에 따라 S3 또는 Swift 클라이언트 애플리케이션을 사용하여 테넌트의 리소스에 액세스할 수 있지만 로컬 사용자는 테넌트 관리자 또는 테넌트 관리 API에 로그인할 수 없습니다.

사용자 페이지에 액세스합니다

액세스 관리 * > * 사용자 * 를 선택합니다.



로컬 사용자를 생성합니다

로컬 사용자를 만들고 하나 이상의 로컬 그룹에 할당하여 액세스 권한을 제어할 수 있습니다.

그룹에 속하지 않은 S3 사용자는 관리 권한이나 S3 그룹 정책이 적용되지 않습니다. 이러한 사용자는 버킷 정책을 통해 S3 버킷 액세스가 부여될 수 있습니다.

그룹에 속하지 않는 Swift 사용자는 관리 권한이나 Swift 컨테이너 액세스 권한이 없습니다.

단계

1. 사용자 생성 * 을 선택합니다.
2. 다음 필드를 작성합니다.
   - * 전체 이름 *: 이 사용자의 전체 이름(예: 사용자의 이름 및 성 또는 응용 프로그램 이름)입니다.
   - * 사용자 이름 *: 이 사용자가 로그인하는 데 사용할 이름입니다. 사용자 이름은 고유해야 하며 변경할 수 없습니다.
   - * 암호 * : 사용자가 로그인할 때 사용하는 암호입니다.
   - * 암호 확인 *: 암호 필드에 입력한 것과 동일한 암호를 입력합니다.
   - * 액세스 거부 *: * 예 * 를 선택하면 사용자가 하나 이상의 그룹에 속해 있더라도 이 사용자는 테넌트 계정에 로그인할 수 없습니다.

     예를 들어 이 기능을 사용하여 사용자의 로그인 기능을 일시적으로 중단할 수 있습니다.
3. Continue * 를 선택합니다.
4. 사용자를 하나 이상의 로컬 그룹에 할당합니다.

   그룹에 속하지 않은 사용자에게는 관리 권한이 없습니다. 권한은 누적됩니다. 사용자는 자신이 속한 모든 그룹에 대한 모든 권한을 갖게 됩니다.
5. 사용자 생성 * 을 선택합니다.

   캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

사용자 세부 정보를 편집합니다

사용자의 세부 정보를 편집할 때 사용자의 전체 이름과 암호를 변경하고, 사용자를 다른 그룹에 추가하고, 사용자가 테넌트에 액세스하지 못하도록 할 수 있습니다.

단계

1. 사용자 목록에서 세부 정보를 보거나 편집할 사용자의 이름을 선택합니다.

   또는 사용자의 확인란을 선택한 다음 * Actions * > * View user details * 를 선택합니다.
2. 필요에 따라 사용자 설정을 변경합니다.
   a. 필요에 따라 전체 이름 또는 편집 아이콘을 선택하여 사용자의 전체 이름을 변경합니다 개요 섹션.

      사용자 이름은 변경할 수 없습니다.
   b. 암호 * 탭에서 필요에 따라 사용자 암호를 변경합니다.
   c. Access * 탭에서 사용자가 로그인(* 아니요 * 선택)하거나 사용자가 필요에 따라 로그인하지 못하도록 합니다(* 예 * 선택).
   d. 그룹 * 탭에서 사용자를 그룹에 추가하거나 필요에 따라 그룹에서 사용자를 제거합니다.
   e. 각 섹션에 필요한 경우 * 변경 사항 저장 * 을 선택합니다.

캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

로컬 사용자가 중복되었습니다

로컬 사용자를 복제하면 새 사용자를 보다 빠르게 만들 수 있습니다.

단계

1. 사용자 목록에서 복제할 사용자를 선택합니다.
2. 사용자 복제 * 를 선택합니다.
3. 새 사용자에 대해 다음 필드를 수정합니다.
   - * 전체 이름 *: 이 사용자의 전체 이름(예: 사용자의 이름 및 성 또는 응용 프로그램 이름)입니다.
   - * 사용자 이름 *: 이 사용자가 로그인하는 데 사용할 이름입니다. 사용자 이름은 고유해야 하며 변경할 수 없습니다.
   - * 암호 * : 사용자가 로그인할 때 사용하는 암호입니다.
   - * 암호 확인 *: 암호 필드에 입력한 것과 동일한 암호를 입력합니다.
   - * 액세스 거부 *: * 예 * 를 선택하면 사용자가 하나 이상의 그룹에 속해 있더라도 이 사용자는 테넌트 계정에 로그인할 수 없습니다.

     예를 들어 이 기능을 사용하여 사용자의 로그인 기능을 일시적으로 중단할 수 있습니다.
4. Continue * 를 선택합니다.
5. 하나 이상의 로컬 그룹을 선택합니다.

   그룹에 속하지 않은 사용자에게는 관리 권한이 없습니다. 권한은 누적됩니다. 사용자는 자신이 속한 모든 그룹에 대한 모든 권한을 갖게 됩니다.
6. 사용자 생성 * 을 선택합니다.

   캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

로컬 사용자를 삭제합니다

StorageGRID 테넌트 계정에 더 이상 액세스할 필요가 없는 로컬 사용자를 영구적으로 삭제할 수 있습니다.

테넌트 관리자를 사용하여 로컬 사용자는 삭제할 수 있지만 페더레이션 사용자는 삭제할 수 없습니다. 통합 사용자를 삭제하려면 통합 ID 소스를 사용해야 합니다.

단계

1. 사용자 목록에서 삭제할 로컬 사용자의 확인란을 선택합니다.
2. Actions * > * Delete user * 를 선택합니다.
3. 확인 대화 상자에서 * 사용자 삭제 * 를 선택하여 시스템에서 사용자를 삭제할 것인지 확인합니다.

   캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

S3 테넌트 계정 관리

S3 액세스 키를 관리합니다

S3 테넌트 계정의 각 사용자는 StorageGRID 시스템에 오브젝트를 저장하고 검색하기 위한 액세스 키가 있어야 합니다. 액세스 키는 액세스 키 ID와 비밀 액세스 키로 구성됩니다.

이 작업에 대해

S3 액세스 키는 다음과 같이 관리할 수 있습니다.

- 자신의 S3 자격 증명 관리 * 권한이 있는 사용자는 자신의 S3 액세스 키를 생성하거나 제거할 수 있습니다.
- 루트 액세스* 권한이 있는 사용자는 S3 루트 계정 및 다른 모든 사용자의 액세스 키를 관리할 수 있습니다. 루트 액세스 키는 버킷 정책에 의해 명시적으로 비활성화되지 않는 한 테넌트의 모든 버킷과 객체에 대한 전체 액세스를 제공합니다.

StorageGRID는 서명 버전 2 및 서명 버전 4 인증을 지원합니다. 버킷 정책에 의해 명시적으로 활성화되지 않은 경우 교차 계정 액세스가 허용되지 않습니다.

자체 S3 액세스 키를 생성합니다

S3 테넌트를 사용 중이며 적절한 권한이 있는 경우 자체 S3 액세스 키를 생성할 수 있습니다. S3 테넌트 계정의 버킷 및 오브젝트에 액세스하려면 액세스 키가 있어야 합니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 자신의 S3 자격 증명 관리 권한이 있어야 합니다. 을 참조하십시오 테넌트 관리 권한.

이 작업에 대해

하나 이상의 S3 액세스 키를 생성하여 테넌트 계정의 버킷을 생성하고 관리할 수 있습니다. 새 액세스 키를 생성한 후 새 액세스 키 ID와 비밀 액세스 키로 응용 프로그램을 업데이트합니다. 보안을 위해 필요한 것보다 더 많은 키를 생성하지 말고 사용하지 않는 키를 삭제하십시오. 하나의 키만 있고 만료되려고 하는 경우 이전 키가 만료되기 전에 새 키를 만든 다음 이전 키를 삭제합니다.

각 키에는 특정 만료 시간 또는 만료 기간이 있을 수 있습니다. 만료 시간에 대한 다음 지침을 따르십시오.

- 키의 만료 시간을 설정하여 특정 기간에 대한 액세스를 제한합니다. 만료 시간을 짧게 설정하면 액세스 키 ID 및 비밀 액세스 키가 실수로 노출되었을 경우 위험을 줄일 수 있습니다. 만료된 키는 자동으로 제거됩니다.
- 환경의 보안 위험이 낮으며 정기적으로 새 키를 만들 필요가 없는 경우 키에 대한 만료 시간을 설정할 필요가 없습니다. 나중에 새 키를 만들려면 이전 키를 수동으로 삭제합니다.

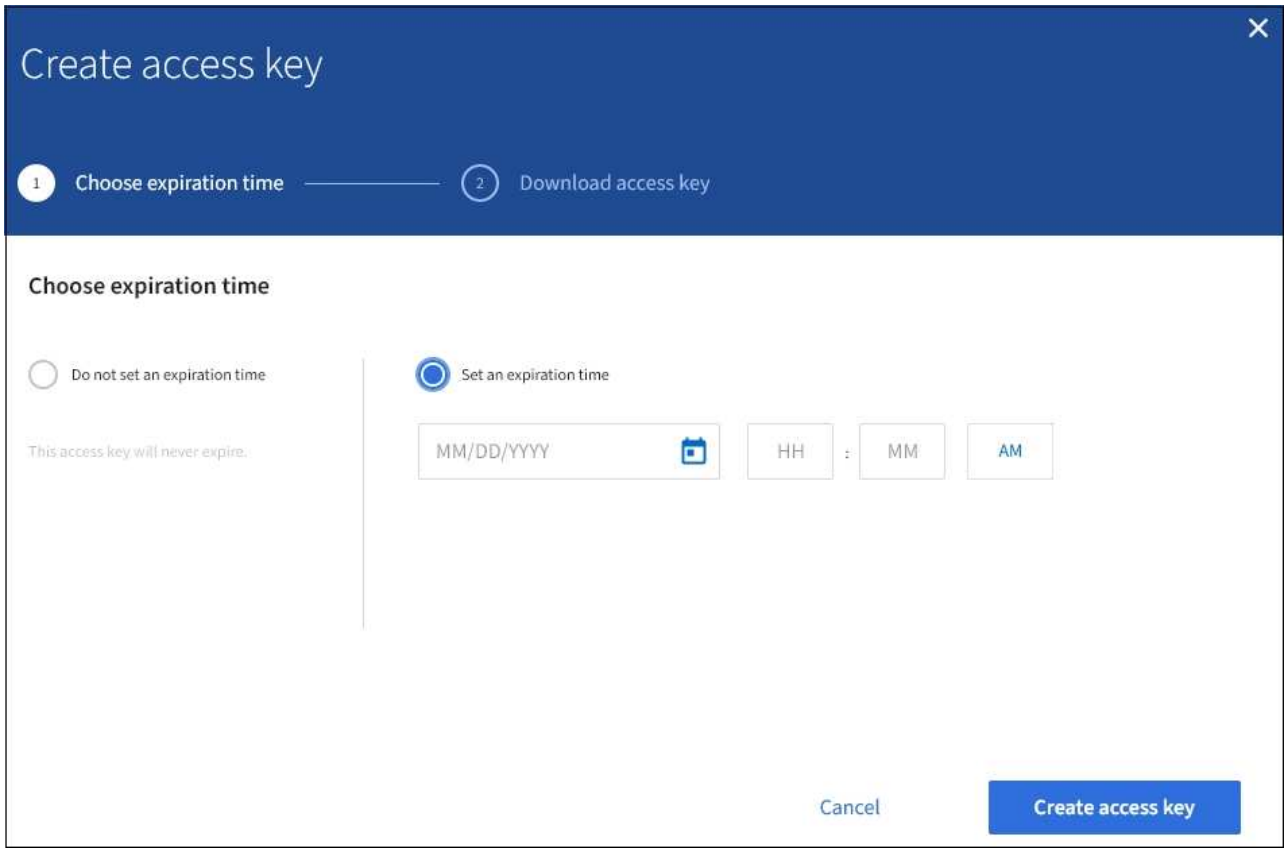
> 계정에 속한 S3 버킷 및 오브젝트는 테넌트 관리자에 계정에 대해 표시된 액세스 키 ID 및 비밀 액세스 키를 사용하여 액세스할 수 있습니다. 이러한 이유로 액세스 키를 암호로 보호해야 합니다. 액세스 키를 정기적으로 회전하고, 계정에서 사용되지 않는 키를 제거하며, 다른 사용자와 공유하지 마십시오.

단계

1. 스토리지(S3) * > * 내 액세스 키 * 를 선택합니다.

   내 액세스 키 페이지가 나타나고 기존 액세스 키가 나열됩니다.

2. Create key * 를 선택합니다.
3. 다음 중 하나를 수행합니다.
   - 만료 시간을 설정하지 않음 * 을 선택하여 만료되지 않는 키를 생성합니다. (기본값)
   - 만료 시간 설정 * 을 선택하고 만료 날짜 및 시간을 설정합니다.

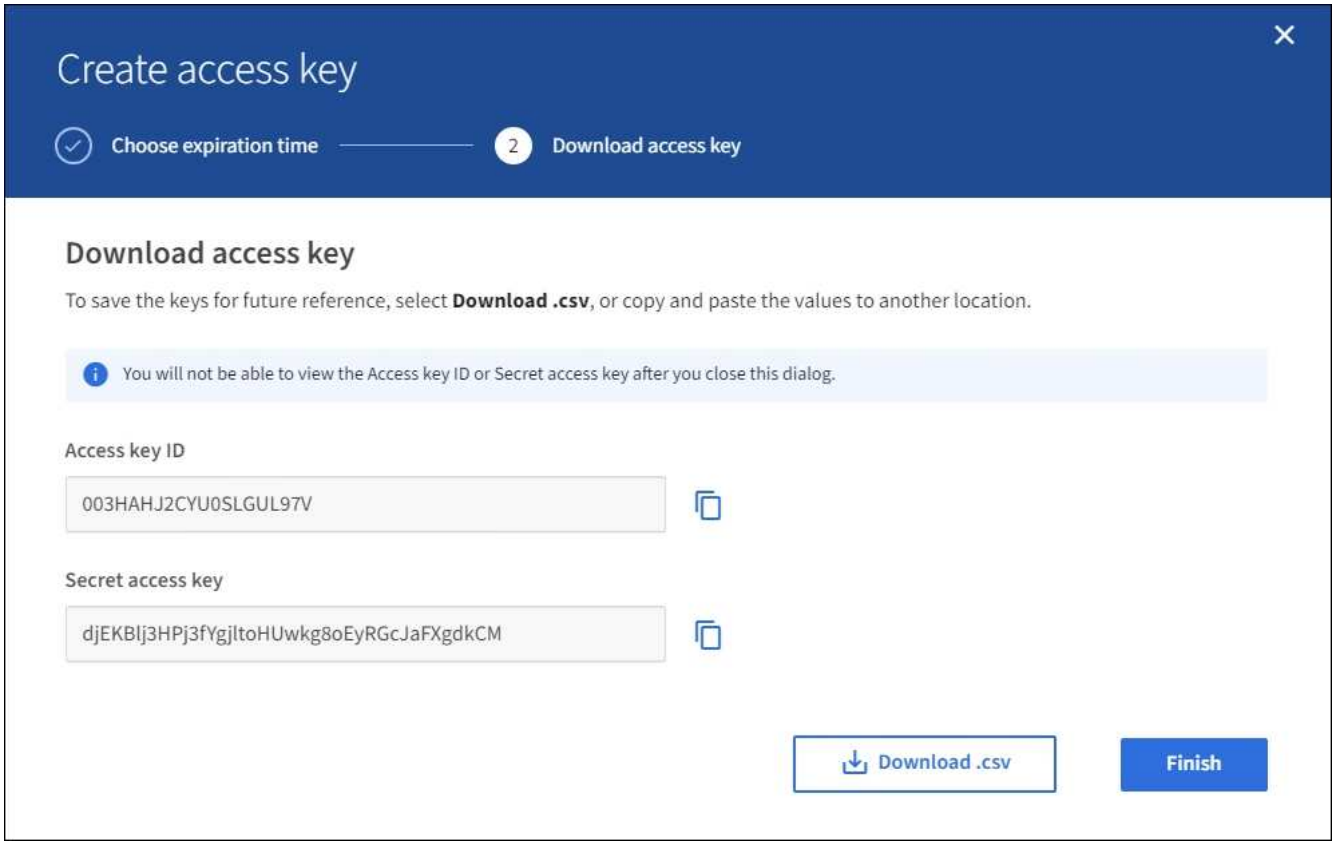


4. Create access key * 를 선택합니다.

   액세스 키 ID 및 비밀 액세스 키가 나열된 다운로드 액세스 키 대화 상자가 나타납니다.

5. 액세스 키 ID와 비밀 액세스 키를 안전한 위치에 복사하거나 * Download.csv * 를 선택하여 액세스 키 ID와 비밀 액세스 키가 포함된 스프레드시트 파일을 저장합니다.

   ⓘ 이 정보를 복사하거나 다운로드할 때까지 이 대화 상자를 닫지 마십시오. 대화 상자를 닫은 후에는 키를 복사하거나 다운로드할 수 없습니다.



6. 마침 * 을 선택합니다.

   새 키가 내 액세스 키 페이지에 나열됩니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

S3 액세스 키를 봅니다

S3 테넌트를 사용하는 경우 적절한 권한이 있으면 S3 액세스 키 목록을 볼 수 있습니다. 만료 시간을 기준으로 목록을 정렬할 수 있으므로 곧 만료되는 키를 확인할 수 있습니다. 필요에 따라 새 키를 만들거나 더 이상 사용하지 않는 키를 삭제할 수 있습니다.

필요한 것

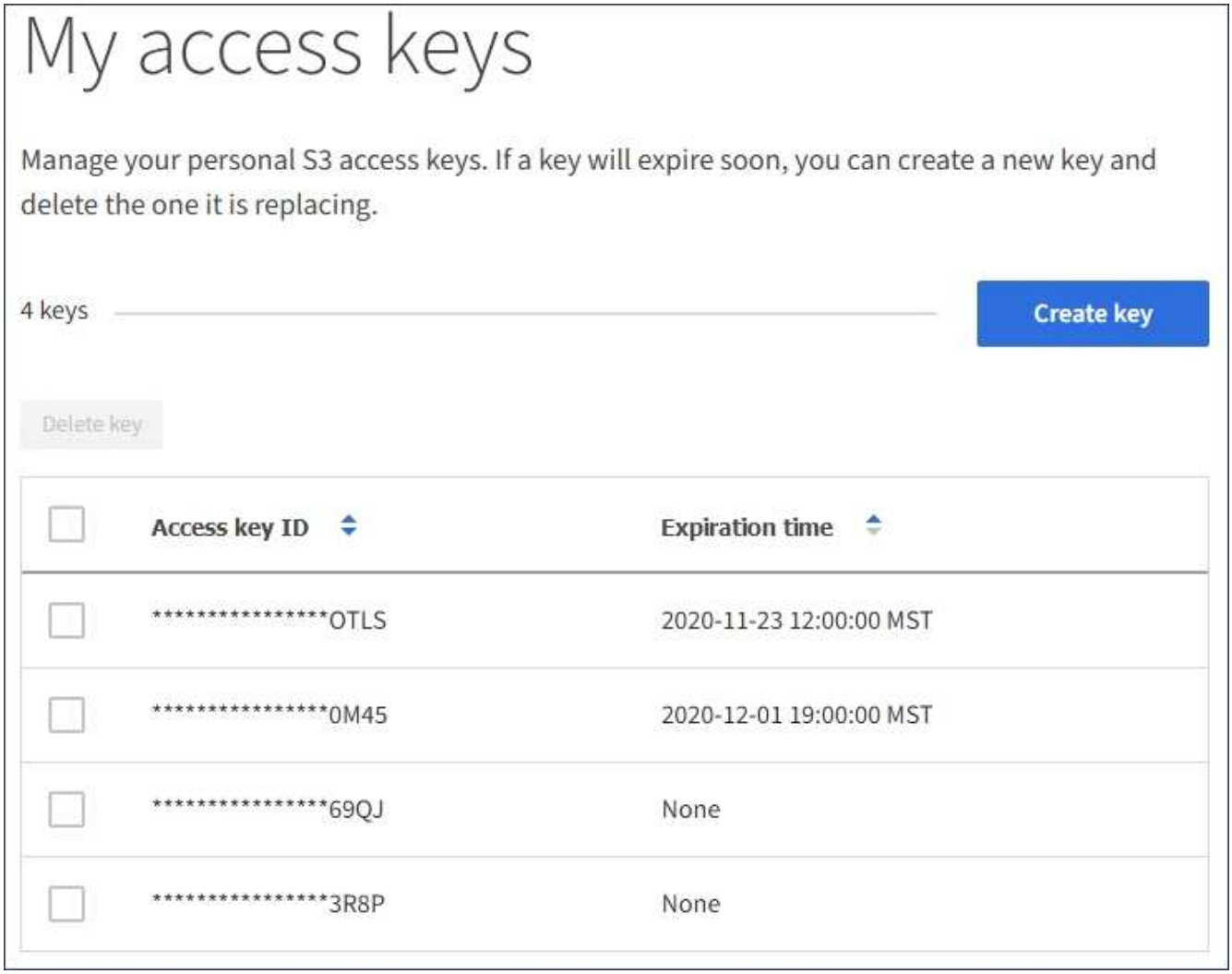
- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 자신의 S3 자격 증명 관리 권한이 있어야 합니다.

> 계정에 속한 S3 버킷 및 오브젝트는 테넌트 관리자에 계정에 대해 표시된 액세스 키 ID 및 비밀 액세스 키를 사용하여 액세스할 수 있습니다. 이러한 이유로 액세스 키를 암호로 보호해야 합니다. 액세스 키를 정기적으로 회전하고, 계정에서 사용되지 않는 키를 제거하며, 다른 사용자와 공유하지 마십시오.

단계

1. 스토리지(S3) * > * 내 액세스 키 * 를 선택합니다.

   내 액세스 키 페이지가 나타나고 기존 액세스 키가 나열됩니다.



2. 키를 * Expiration Time * 또는 * Access key ID * 로 정렬합니다.
3. 필요에 따라 새 키를 만들고 더 이상 사용하지 않는 키를 수동으로 삭제합니다.

   기존 키가 만료되기 전에 새 키를 만들면 계정의 개체에 대한 액세스 권한을 일시적으로 잃지 않고 새 키를 사용할 수 있습니다.

   만료된 키는 자동으로 제거됩니다.

관련 정보

자체 S3 액세스 키를 생성합니다

자체 S3 액세스 키를 삭제합니다

자체 **S3** 액세스 키를 삭제합니다

S3 테넌트를 사용하는 경우 적절한 권한이 있으면 자신의 S3 액세스 키를 삭제할 수 있습니다. 액세스 키가 삭제된 후에는 더 이상 테넌트 계정의 객체와 버킷에 액세스할 수 없습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.

- 자신의 S3 자격 증명 관리 권한이 있어야 합니다. 을 참조하십시오 테넌트 관리 권한.

> 계정에 속한 S3 버킷 및 오브젝트는 테넌트 관리자에 계정에 대해 표시된 액세스 키 ID 및 비밀 액세스 키를 사용하여 액세스할 수 있습니다. 이러한 이유로 액세스 키를 암호로 보호해야 합니다. 액세스 키를 정기적으로 회전하고, 계정에서 사용되지 않는 키를 제거하며, 다른 사용자와 공유하지 마십시오.

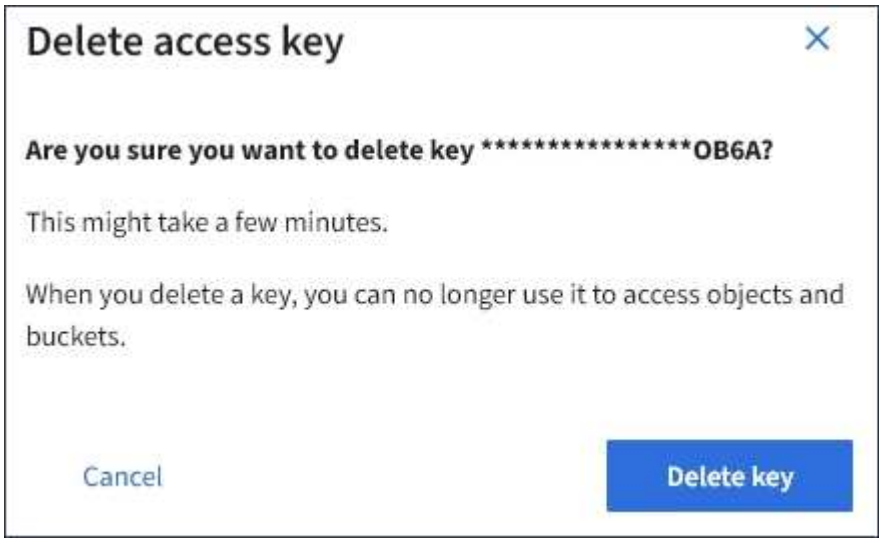
단계

1. 스토리지(S3) * > * 내 액세스 키 * 를 선택합니다.

   내 액세스 키 페이지가 나타나고 기존 액세스 키가 나열됩니다.

2. 제거할 각 액세스 키에 대한 확인란을 선택합니다.
3. Delete key * 를 선택합니다.

   확인 대화 상자가 나타납니다.



4. Delete key * 를 선택합니다.

   페이지의 오른쪽 상단에 확인 메시지가 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

다른 사용자의 **S3** 액세스 키를 생성합니다

S3 테넌트를 사용하는 경우 적절한 권한이 있으면 버킷 및 오브젝트에 액세스해야 하는 애플리케이션 같은 다른 사용자를 위한 S3 액세스 키를 생성할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있어야 합니다.

이 작업에 대해

하나 이상의 다른 사용자를 위한 S3 액세스 키를 생성하여 해당 테넌트 계정에 대한 버킷을 생성하고 관리할 수 있습니다. 새 액세스 키를 생성한 후 새 액세스 키 ID와 비밀 액세스 키로 응용 프로그램을 업데이트합니다. 보안을 위해

사용자 요구 사항보다 많은 키를 생성하지 말고 사용하지 않는 키를 삭제하십시오. 하나의 키만 있고 만료되려고 하는 경우 이전 키가 만료되기 전에 새 키를 만든 다음 이전 키를 삭제합니다.

각 키에는 특정 만료 시간 또는 만료 기간이 있을 수 있습니다. 만료 시간에 대한 다음 지침을 따르십시오.

- 키의 만료 시간을 설정하여 사용자의 액세스를 특정 기간으로 제한합니다. 만료 시간을 짧게 설정하면 액세스 키 ID 및 비밀 액세스 키가 실수로 노출될 경우 위험을 줄일 수 있습니다. 만료된 키는 자동으로 제거됩니다.
- 환경의 보안 위험이 낮으며 주기적으로 새 키를 만들 필요가 없는 경우 키의 만료 시간을 설정할 필요가 없습니다. 나중에 새 키를 만들려면 이전 키를 수동으로 삭제합니다.

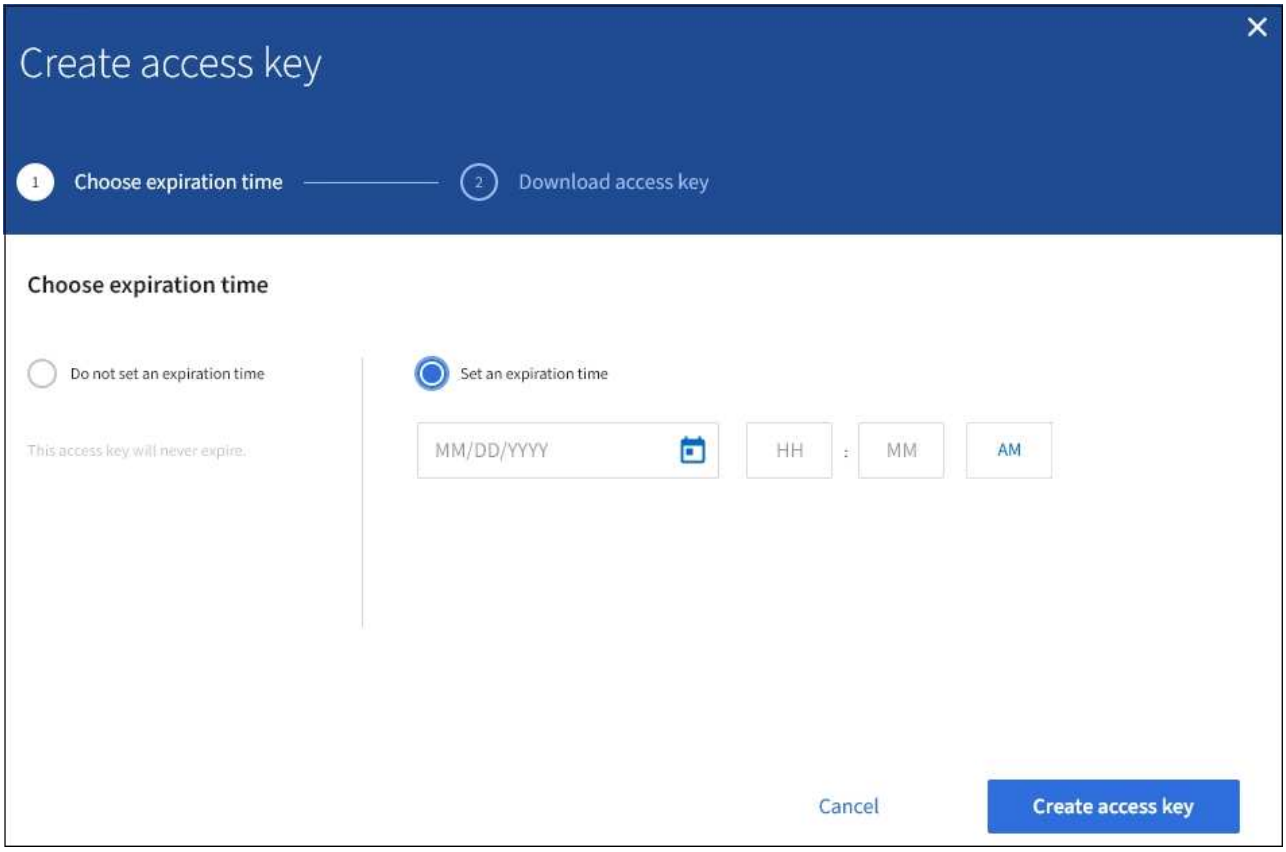
> 사용자에게 속한 S3 버킷 및 오브젝트는 테넌트 관리자에서 해당 사용자에 대해 표시된 액세스 키 ID 및 비밀 액세스 키를 사용하여 액세스할 수 있습니다. 이러한 이유로 액세스 키를 암호로 보호해야 합니다. 액세스 키를 정기적으로 회전하고, 계정에서 사용되지 않는 키를 제거하며, 다른 사용자와 공유하지 마십시오.

단계

1. 액세스 관리 * > * 사용자 * 를 선택합니다.
2. S3 액세스 키를 관리할 사용자를 선택합니다.

   사용자 세부 정보 페이지가 나타납니다.

3. 액세스 키 * 를 선택한 다음 * 키 만들기 * 를 선택합니다.
4. 다음 중 하나를 수행합니다.
   - 만료 시간을 설정하지 않음 * 을 선택하여 만료되지 않는 키를 생성합니다. (기본값)
   - 만료 시간 설정 * 을 선택하고 만료 날짜 및 시간을 설정합니다.

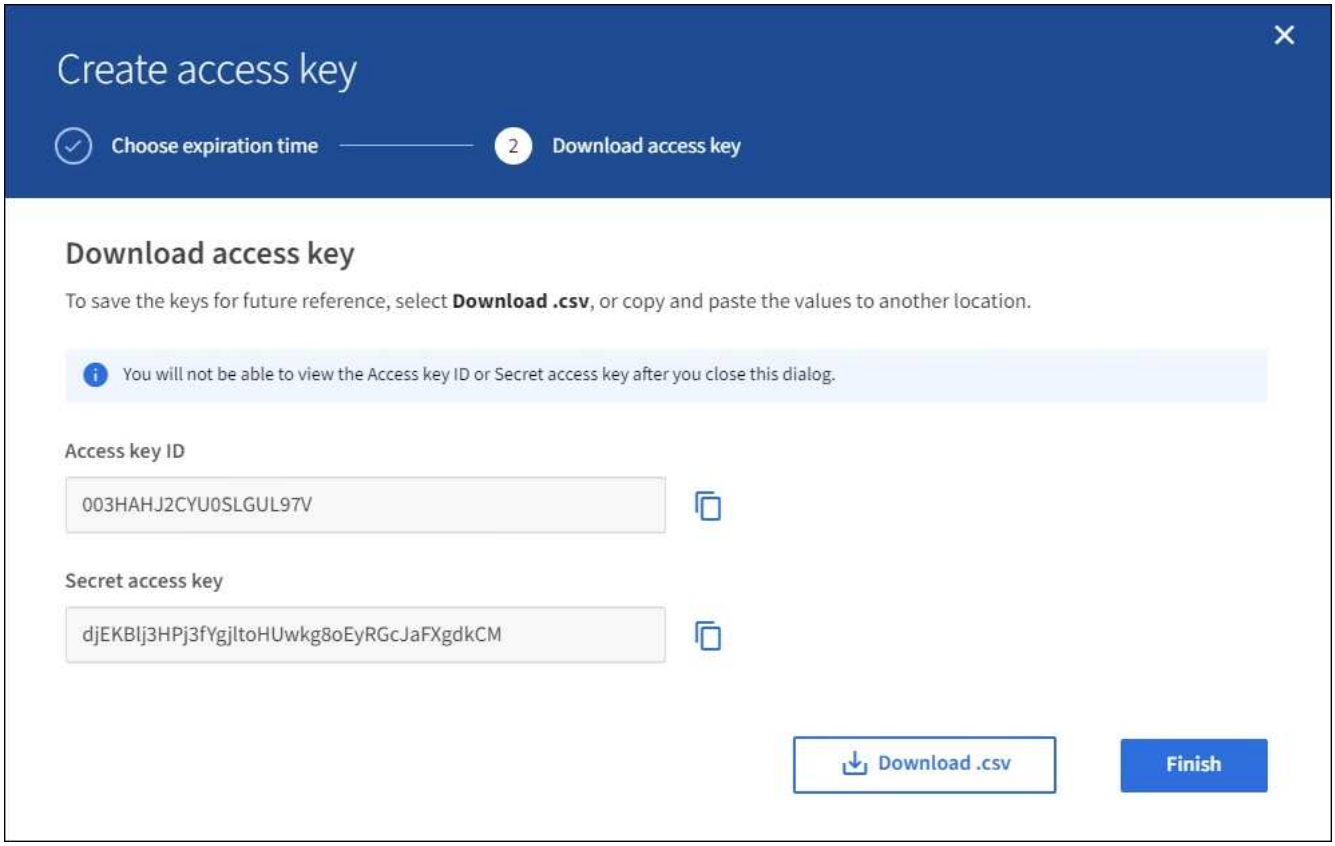


5. Create access key * 를 선택합니다.

   액세스 키 ID 및 비밀 액세스 키가 나열된 다운로드 액세스 키 대화 상자가 나타납니다.

6. 액세스 키 ID와 비밀 액세스 키를 안전한 위치에 복사하거나 * Download.csv * 를 선택하여 액세스 키 ID와 비밀 액세스 키가 포함된 스프레드시트 파일을 저장합니다.

   > 이 정보를 복사하거나 다운로드할 때까지 이 대화 상자를 닫지 마십시오. 대화 상자를 닫은 후에는 키를 복사하거나 다운로드할 수 없습니다.



7. 마침 * 을 선택합니다.

   새 키가 사용자 세부 정보 페이지의 액세스 키 탭에 나열됩니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

관련 정보

테넌트 관리 권한

다른 사용자의 **S3** 액세스 키를 봅니다

S3 테넌트를 사용하는 경우 적절한 권한이 있으면 다른 사용자의 S3 액세스 키를 볼 수 있습니다. 만료 시간을 기준으로 목록을 정렬하면 곧 만료되는 키를 확인할 수 있습니다. 필요에 따라 새 키를 생성하고 더 이상 사용하지 않는 키를 삭제할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.

- 루트 액세스 권한이 있어야 합니다.

> 사용자에게 속한 S3 버킷 및 오브젝트는 테넌트 관리자에서 해당 사용자에 대해 표시된 액세스 키 ID 및 비밀 액세스 키를 사용하여 액세스할 수 있습니다. 이러한 이유로 액세스 키를 암호로 보호해야 합니다. 액세스 키를 정기적으로 회전하고, 계정에서 사용되지 않는 키를 제거하며, 다른 사용자와 공유하지 마십시오.

단계

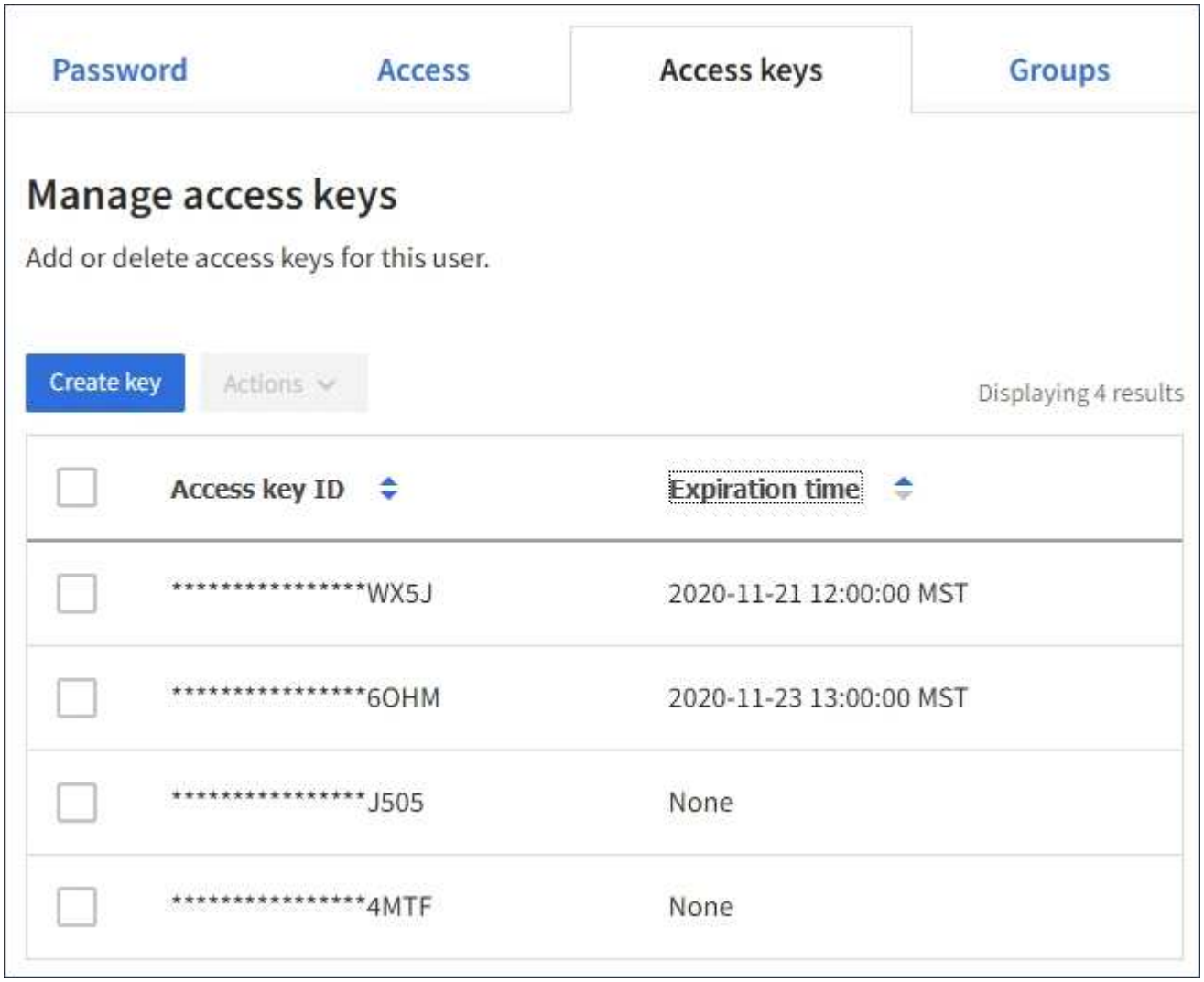
1. 액세스 관리 * > * 사용자 * 를 선택합니다.

   사용자 페이지가 나타나고 기존 사용자가 나열됩니다.

2. S3 액세스 키를 보려는 사용자를 선택합니다.

   사용자 세부 정보 페이지가 나타납니다.

3. 액세스 키 * 를 선택합니다.



4. 키를 * Expiration Time * 또는 * Access key ID * 로 정렬합니다.
5. 필요에 따라 새 키를 생성하고 에서 더 이상 사용하지 않는 키를 수동으로 삭제합니다.

기존 키가 만료되기 전에 새 키를 만들면 사용자는 계정의 개체에 대한 액세스 권한을 일시적으로 잃지 않고 새 키를 사용할 수 있습니다.

만료된 키는 자동으로 제거됩니다.

관련 정보

다른 사용자의 S3 액세스 키를 생성합니다

다른 사용자의 S3 액세스 키를 삭제합니다

다른 사용자의 **S3** 액세스 키를 삭제합니다

S3 테넌트를 사용하는 경우 적절한 권한이 있으면 다른 사용자의 S3 액세스 키를 삭제할 수 있습니다. 액세스 키가 삭제된 후에는 더 이상 테넌트 계정의 객체와 버킷에 액세스할 수 없습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 루트 액세스 권한이 있어야 합니다. 을 참조하십시오 테넌트 관리 권한.

> 사용자에게 속한 S3 버킷 및 오브젝트는 테넌트 관리자에서 해당 사용자에 대해 표시된 액세스 키 ID 및 비밀 액세스 키를 사용하여 액세스할 수 있습니다. 이러한 이유로 액세스 키를 암호로 보호해야 합니다. 액세스 키를 정기적으로 회전하고, 계정에서 사용되지 않는 키를 제거하며, 다른 사용자와 공유하지 마십시오.

단계

1. 액세스 관리 * > * 사용자 * 를 선택합니다.

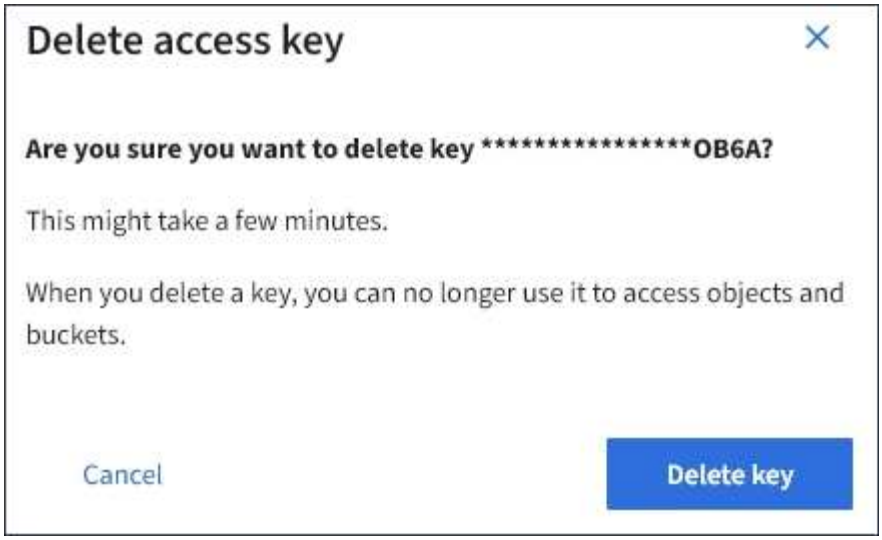
   사용자 페이지가 나타나고 기존 사용자가 나열됩니다.

2. S3 액세스 키를 관리할 사용자를 선택합니다.

   사용자 세부 정보 페이지가 나타납니다.

3. 액세스 키 * 를 선택한 다음 삭제할 각 액세스 키에 대한 확인란을 선택합니다.
4. Actions * > * Delete Selected key * 를 선택합니다.

   확인 대화 상자가 나타납니다.



5. Delete key * 를 선택합니다.

   페이지의 오른쪽 상단에 확인 메시지가 나타납니다. 캐시 때문에 변경사항을 적용하려면 15분이 소요될 수 있습니다.

S3 버킷을 관리합니다

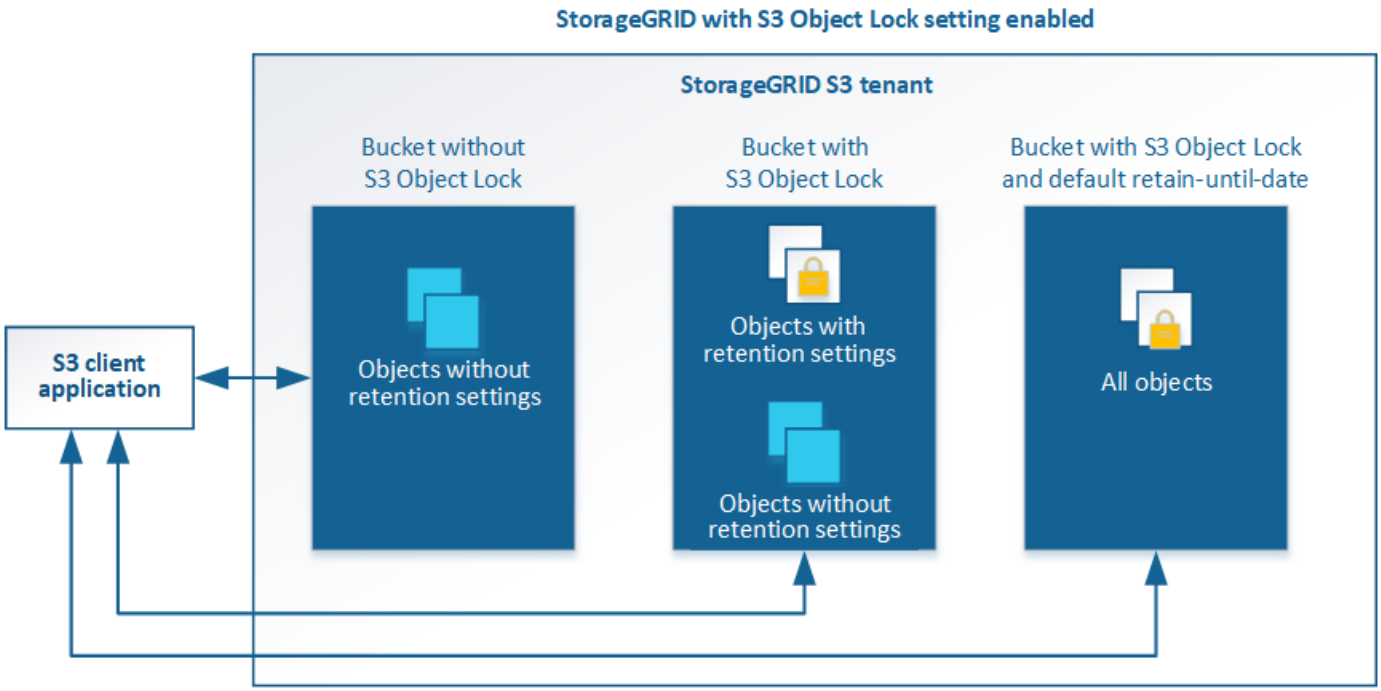
테넌트에 S3 오브젝트 잠금을 사용합니다

오브젝트가 보존 규정 요구사항을 충족해야 하는 경우 StorageGRID의 S3 오브젝트 잠금 기능을 사용할 수 있습니다.

S3 오브젝트 잠금이란 무엇입니까?

StorageGRID S3 오브젝트 잠금 기능은 Amazon S3(Amazon Simple Storage Service)의 S3 오브젝트 잠금과 동등한 오브젝트 보호 솔루션입니다.

그림에서 볼 수 있듯이 StorageGRID 시스템에 대해 글로벌 S3 오브젝트 잠금 설정을 사용하면 S3 테넌트 계정이 S3 오브젝트 잠금을 사용하거나 사용하지 않고 버킷을 생성할 수 있습니다. 버킷에 S3 오브젝트 잠금이 설정된 경우 S3 클라이언트 애플리케이션이 해당 버킷의 모든 오브젝트 버전에 대한 보존 설정을 선택적으로 지정할 수 있습니다. 오브젝트 버전에 S3 오브젝트 잠금으로 보호할 보존 설정이 지정되어 있어야 합니다.



StorageGRID S3 오브젝트 잠금 기능은 Amazon S3 규정 준수 모드에 상응하는 단일 보존 모드를 제공합니다. 기본적으로 보호된 개체 버전은 사용자가 덮어쓰거나 삭제할 수 없습니다. StorageGRID S3 오브젝트 잠금 기능은 거버넌스 모드를 지원하지 않으며, 특별한 권한이 있는 사용자가 보존 설정을 무시하거나 보호된 오브젝트를 삭제할 수 없습니다.

버킷에 S3 오브젝트 잠금이 활성화된 경우 오브젝트를 생성하거나 업데이트할 때 S3 클라이언트 애플리케이션에서 다음 오브젝트 레벨 보존 설정 중 하나 또는 모두를 선택적으로 지정할 수 있습니다.

- * Retain-until-date *: 개체 버전의 Retain-until-date가 미래인 경우 개체를 검색할 수 있지만 수정하거나 삭제할 수 없습니다. 필요에 따라 오브젝트의 보존 기간(Retain-until-date)을 늘릴 수 있지만 이 날짜는 줄일 수 없습니다.
- * 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. 법적 보류는 보존 기한 과 무관합니다.

또한 가능합니다 버킷의 기본 보존 모드 및 기본 보존 기간을 지정합니다. 고유한 보존 설정을 지정하지 않는 버킷에 추가된 각 오브젝트에 적용됩니다.

이러한 설정에 대한 자세한 내용은 을 참조하십시오 S3 오브젝트 잠금을 사용합니다.

레거시 준수 버킷을 관리합니다

S3 오브젝트 잠금 기능은 이전 StorageGRID 버전에서 사용할 수 있었던 규정 준수 기능을 대체합니다. 이전 버전의 StorageGRID를 사용하여 준수 버킷을 생성한 경우 이러한 버켓의 설정을 계속 관리할 수 있지만, 더 이상 새로운 준수 버킷을 생성할 수 없습니다. 자세한 내용은 NetApp 기술 자료 문서를 참조하십시오.

"NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

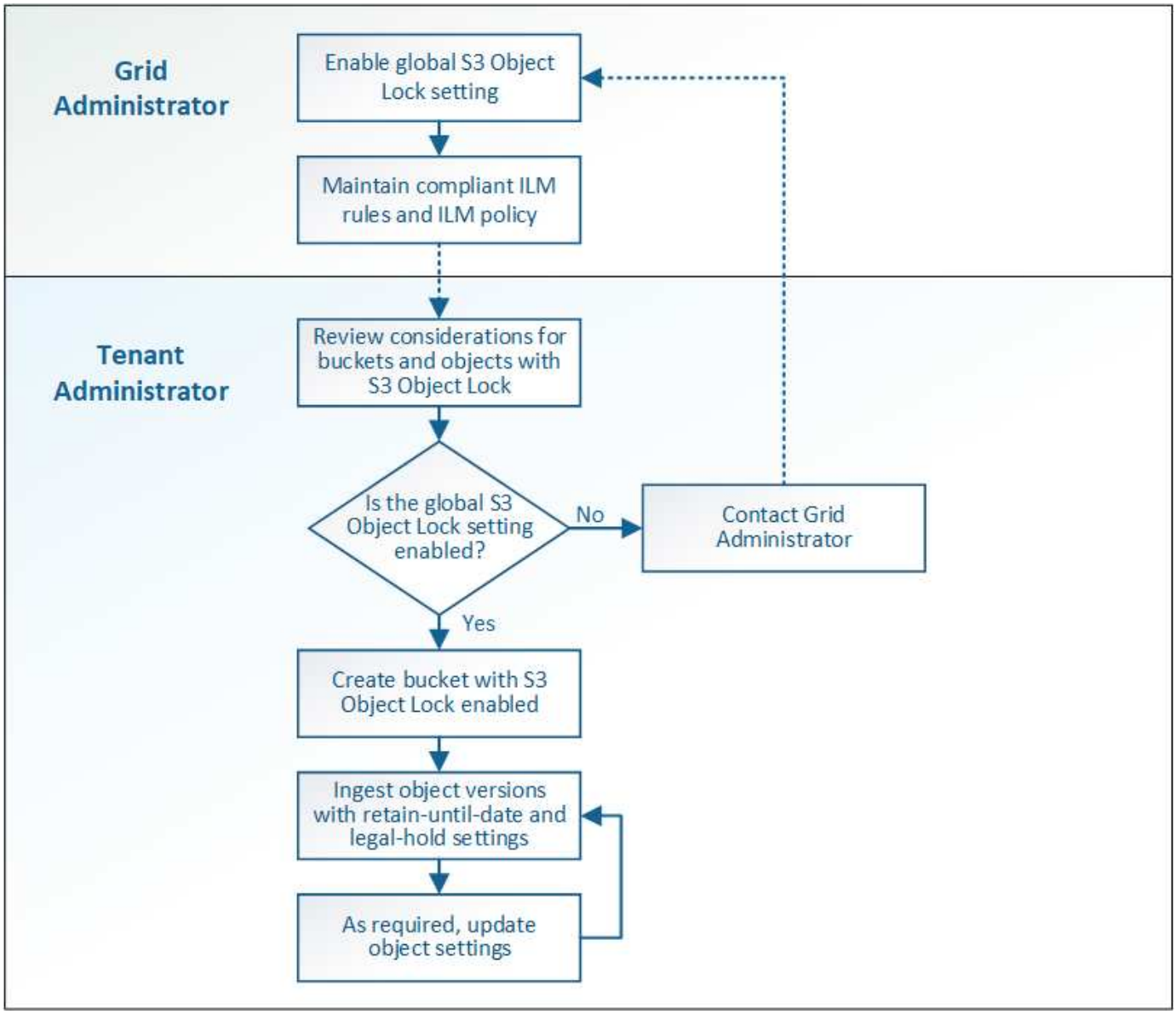
S3 오브젝트 잠금 워크플로

워크플로우 다이어그램은 StorageGRID에서 S3 오브젝트 잠금 기능을 사용하기 위한 상위 단계를 보여줍니다.

S3 오브젝트 잠금이 설정된 버킷을 생성하려면 그리드 관리자가 전체 StorageGRID 시스템에 대해 글로벌 S3

오브젝트 잠금 설정을 활성화해야 합니다. 또한 그리드 관리자는 을 확인해야 합니다 ILM(정보 수명 주기 관리) 정책 은(는) ""준수""입니다. S3 오브젝트 잠금이 활성화된 버킷의 요구 사항을 충족해야 합니다. 자세한 내용은 그리드 관리자에게 문의하거나 정보 수명 주기 관리를 사용하여 개체를 관리하는 지침을 참조하십시오.

글로벌 S3 오브젝트 잠금 설정을 활성화한 후 S3 오브젝트 잠금이 설정된 버킷을 생성할 수 있습니다. 그런 다음 S3 클라이언트 애플리케이션을 사용하여 필요에 따라 각 오브젝트 버전에 대한 보존 설정을 지정할 수 있습니다.



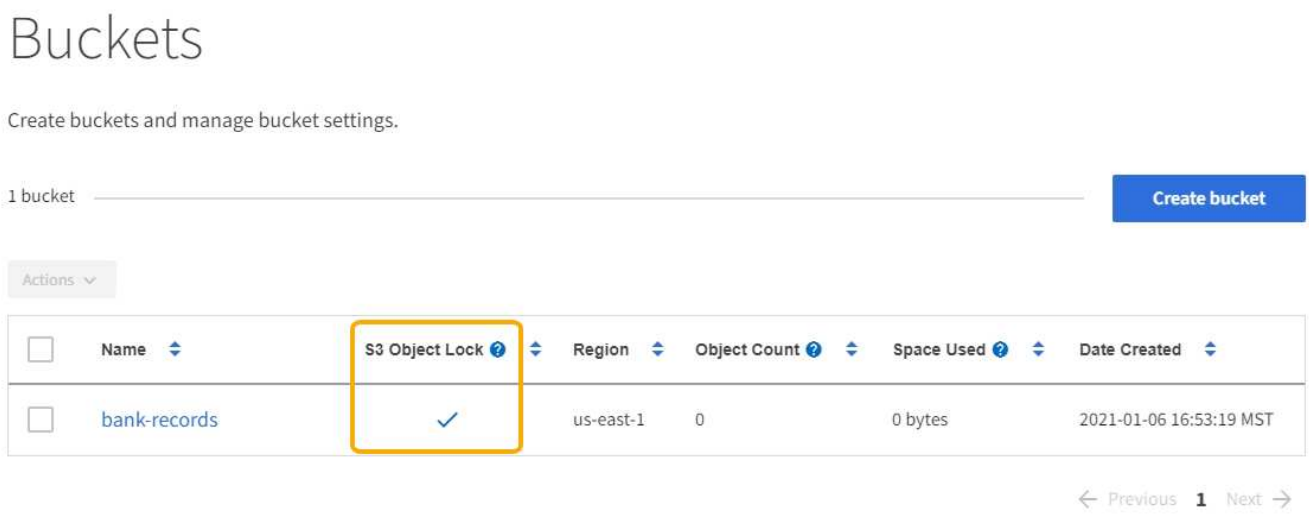
S3 오브젝트 잠금에 대한 요구사항

버킷에 대해 S3 오브젝트 잠금을 설정하기 전에 S3 오브젝트 잠금 버킷 및 오브젝트에 대한 요구사항과 S3 오브젝트 잠금이 활성화된 버킷에 포함된 오브젝트의 수명 주기를 검토하십시오.

S3 오브젝트 잠금이 설정된 버킷의 요구 사항

- StorageGRID 시스템에 대해 글로벌 S3 오브젝트 잠금 설정이 활성화된 경우 테넌트 관리자, 테넌트 관리 API 또는 S3 REST API를 사용하여 S3 오브젝트 잠금이 활성화된 버킷을 생성할 수 있습니다.

  테넌트 관리자의 이 예에서는 S3 오브젝트 잠금이 설정된 버킷을 보여 줍니다.



- S3 오브젝트 잠금을 사용하려는 경우 버킷을 생성할 때 S3 오브젝트 잠금을 활성화해야 합니다. 기존 버킷에 대해 S3 오브젝트 잠금을 활성화할 수 없습니다.
- S3 오브젝트 잠금에서 버킷 버전 관리가 필요합니다. 버킷에 대해 S3 오브젝트 잠금이 활성화된 경우 StorageGRID는 해당 버킷의 버전 관리를 자동으로 활성화합니다.
- S3 오브젝트 잠금이 설정된 버킷을 생성한 후에는 해당 버켓에 대한 S3 오브젝트 잠금을 비활성화하거나 버전 관리를 일시 중단할 수 없습니다.
- 필요에 따라 버킷의 기본 보존을 구성할 수 있습니다. 개체 버전이 업로드되면 기본 보존 기간이 개체 버전에 적용됩니다. 객체 버전 업로드 요청에 보존 모드와 보존 기간을 지정하여 버킷 기본값을 재정의할 수 있습니다.
- S3 오브젝트 라이프사이클 버킷에 대해 버킷 라이프사이클 구성이 지원됩니다.
- S3 오브젝트 잠금이 설정된 버킷에는 CloudMirror 복제가 지원되지 않습니다.

S3 오브젝트 잠금이 설정된 버킷의 오브젝트 요구사항

- 오브젝트 버전을 보호하려면 S3 클라이언트 애플리케이션이 버킷 기본 보존을 구성하거나 각 업로드 요청에서 보존 설정을 지정해야 합니다.
- 개체 버전에 대한 보존 기간을 늘릴 수 있지만 이 값을 줄일 수는 없습니다.
- 법적 조치 또는 규제 조사가 보류 중인 경우 개체 버전에 법적 증거 자료를 두어 관련 정보를 보존할 수 있습니다. 개체 버전이 법적 증거 자료 보관 중인 경우, 해당 개체가 보존 기한 에 도달한 경우에도 StorageGRID에서 해당 개체를 삭제할 수 없습니다. 법적 증거 자료 보관 기간이 해제됨과 동시에, 보존 기한이 만료된 경우 개체 버전을 삭제할 수 있습니다.
- S3 오브젝트 잠금에는 버전 관리되는 버킷을 사용해야 합니다. 보존 설정은 개별 개체 버전에 적용됩니다. 개체 버전에는 보존 기한 및 법적 보류 설정이 둘 다 있을 수 있으며, 둘 중 하나만 설정할 수도 있고 둘 다 가질 수도 없습니다. 개체에 대한 보존 기한 또는 법적 보류 설정을 지정하면 요청에 지정된 버전만 보호됩니다. 이전 버전의 개체는 잠겨 있는 상태에서 새 버전의 개체를 만들 수 있습니다.

S3 오브젝트 잠금이 설정된 버킷의 오브젝트 라이프사이클

S3 오브젝트 잠금이 설정된 버킷에 저장된 각 오브젝트는 다음 3단계를 거칩니다.

1. * 오브젝트 수집 *
    - S3 오브젝트 잠금이 설정된 버킷에 오브젝트 버전을 추가할 경우 S3 클라이언트 애플리케이션이 오브젝트에 대한 보존 설정을 선택적으로 지정할 수 있습니다(보존 기한, 법적 보류 또는 둘 다). 그런 다음

StorageGRID에서는 해당 개체의 메타데이터를 생성하며 고유한 UUID(Object Identifier)와 수집 날짜 및 시간이 포함됩니다.

- 보존 설정이 포함된 오브젝트 버전을 수집하면 해당 데이터와 S3 사용자 정의 메타데이터를 수정할 수 없습니다.
- StorageGRID는 오브젝트 메타데이터를 오브젝트 데이터와 독립적으로 저장합니다. 이 기능은 각 사이트에서 모든 오브젝트 메타데이터의 복사본을 3개 유지 관리합니다.

2. * 개체 보존 *
   - 개체의 여러 복사본이 StorageGRID에 저장됩니다. 정확한 복제본 수와 유형 및 스토리지 위치는 활성 ILM 정책의 규정 준수 규칙에 따라 결정됩니다.
3. * 개체 삭제 *
   - 보존 기한 에 도달하면 개체를 삭제할 수 있습니다.
   - 법적 증거 자료 보관 중인 개체는 삭제할 수 없습니다.

S3 버킷을 생성합니다

테넌트 관리자를 사용하여 오브젝트 데이터용 S3 버킷을 생성할 수 있습니다. 버킷을 생성할 때 버킷의 이름과 영역을 지정해야 합니다. StorageGRID 시스템에 대해 글로벌 S3 오브젝트 잠금 설정이 활성화된 경우 버킷에 대해 S3 오브젝트 잠금을 선택적으로 활성화할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인했습니다 지원되는 웹 브라우저.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 사용 권한 설정을 재정의합니다.

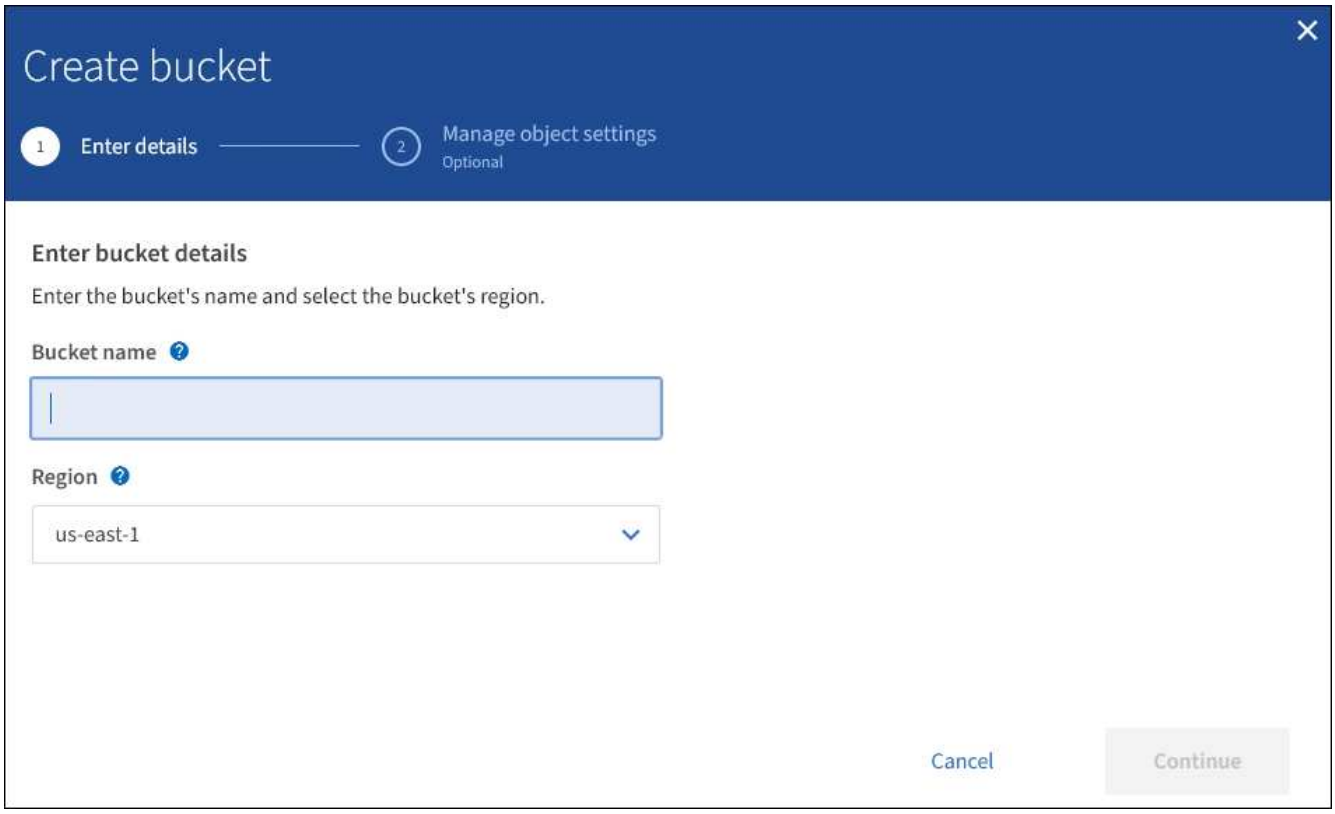
> 에서 버킷 또는 오브젝트의 S3 오브젝트 잠금 속성을 설정하거나 수정하는 권한을 부여할 수 있습니다 버킷 정책 또는 그룹 정책.

- S3 오브젝트 잠금을 사용하여 버킷을 생성하려는 경우 StorageGRID 시스템에 대해 글로벌 S3 오브젝트 잠금 설정을 활성화하고 S3 오브젝트 잠금 버킷 및 오브젝트에 대한 요구사항을 검토했습니다.

  S3 오브젝트 잠금을 사용합니다

단계

1. 스토리지(S3) * > * 버킷 * 을 선택합니다.
2. Create bucket * 을 선택합니다.



3. 버킷의 고유한 이름을 입력하십시오.

> 버킷을 생성한 후에는 버킷 이름을 변경할 수 없습니다.

버킷 이름은 다음 규칙을 준수해야 합니다.

- 각 StorageGRID 시스템에서 고유해야 합니다(테넌트 계정에서만 고유한 것은 아님).
- DNS를 준수해야 합니다.
- 3자 이상 63자 이하여야 합니다.
- 각 레이블은 소문자 또는 숫자로 시작하고 끝나야 하며 소문자, 숫자 및 하이픈만 사용할 수 있습니다.
- 가상 호스팅 스타일 요청에서 기간을 사용하지 않아야 합니다. 마침표는 서버 와일드카드 인증서 확인에 문제를 일으킬 수 있습니다.

> 자세한 내용은 를 참조하십시오 "버킷 명명 규칙에 대한 AWS(Amazon Web Services) 문서입니다".

4. 이 버킷의 영역을 선택합니다.

StorageGRID 관리자가 사용 가능한 영역을 관리합니다. 버킷 영역은 오브젝트에 적용되는 데이터 보호 정책에 영향을 미칠 수 있습니다. 기본적으로 모든 버킷은 us-east-1 영역에 생성됩니다.

> 버킷을 생성한 후에는 지역을 변경할 수 없습니다.

5. Continue * 를 선택합니다.
6. 필요한 경우 버킷에 대한 오브젝트 버전 관리를 활성화합니다.

이 버킷에 각 오브젝트의 모든 버전을 저장하려면 오브젝트 버전 관리를 활성화하십시오. 그런 다음 필요에 따라 개체의 이전 버전을 검색할 수 있습니다.

7. S3 오브젝트 잠금 섹션이 나타나면 버킷에 대해 S3 오브젝트 잠금을 선택적으로 활성화합니다.

   > 버킷을 생성한 후에는 S3 오브젝트 잠금을 설정하거나 해제할 수 없습니다.

   S3 오브젝트 잠금 섹션은 글로벌 S3 오브젝트 잠금 설정이 활성화된 경우에만 나타납니다.

   S3 클라이언트 애플리케이션이 버킷에 추가된 오브젝트에 대한 보관 종료 날짜 및 법적 보류 설정을 지정하려면 먼저 버킷에 대해 S3 오브젝트 잠금을 활성화해야 합니다.

   버킷에 대해 S3 오브젝트 잠금을 설정하면 버킷 버전 관리가 자동으로 활성화됩니다. 또한 가능합니다 버킷의 기본 보존 모드 및 기본 보존 기간을 지정합니다 고유한 보존 설정을 지정하지 않는 버킷에 수집된 각 개체에 적용됩니다.

8. Create bucket * 을 선택합니다.

   버킷이 생성되어 버킷 페이지의 테이블에 추가됩니다.

관련 정보

ILM을 사용하여 개체를 관리합니다

테넌트 관리 API 이해

S3을 사용합니다

S3 버킷 세부 정보를 봅니다

테넌트 계정에서 버킷 및 버킷 설정 목록을 볼 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.

단계

1. 스토리지(S3) * > * 버킷 * 을 선택합니다.

   Bucket 페이지가 나타나고 테넌트 계정에 대한 모든 버킷이 나열됩니다.

2. 각 버킷에 대한 정보를 검토합니다.

   필요에 따라 모든 열을 기준으로 정보를 정렬하거나 목록 앞뒤에 페이지를 표시할 수 있습니다.

   - 이름: 변경할 수 없는 버킷의 고유 이름입니다.
   - S3 오브젝트 잠금: 이 버킷에 대해 S3 오브젝트 잠금이 설정되었는지 여부.

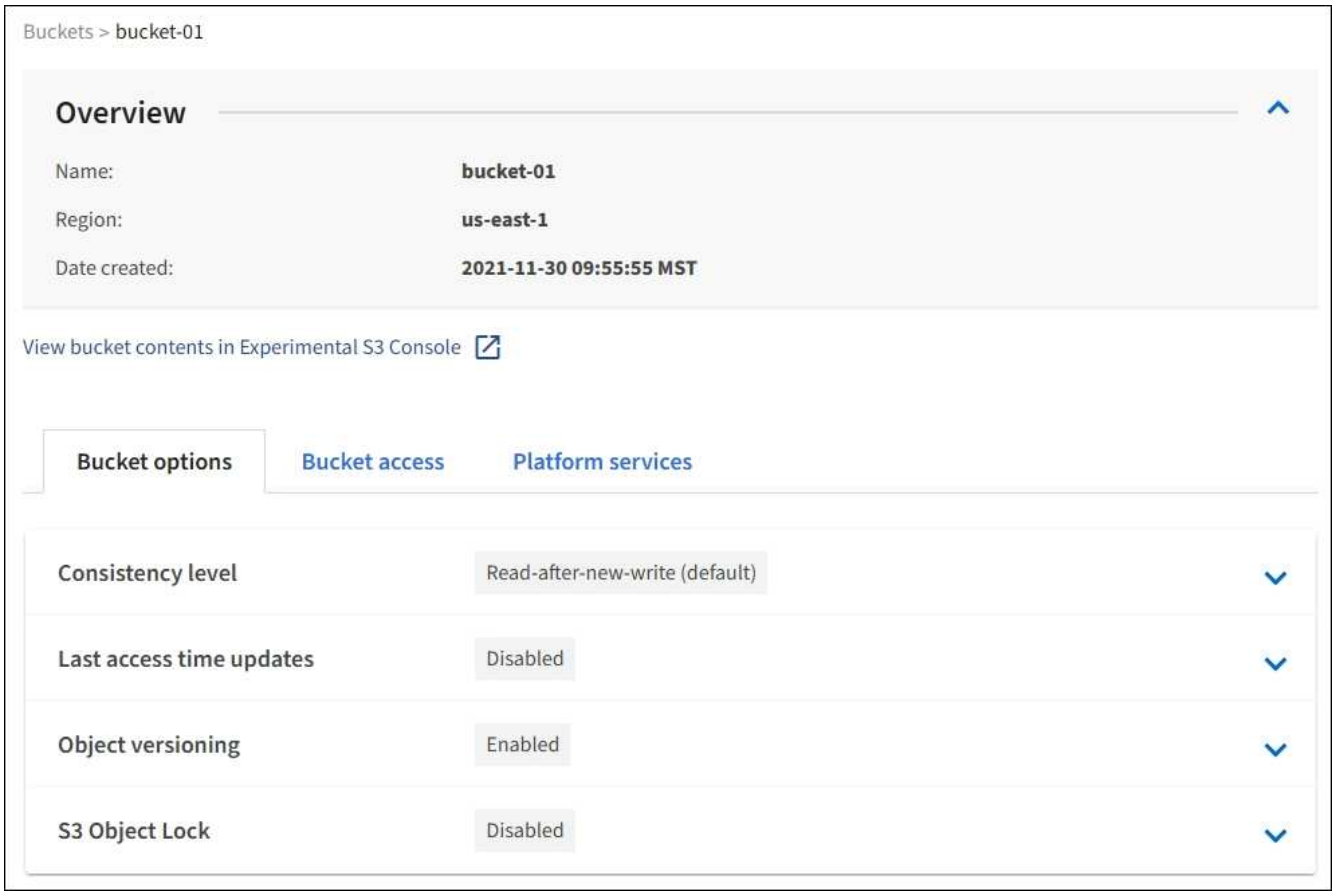
     전역 S3 오브젝트 잠금 설정이 비활성화된 경우 이 열은 표시되지 않습니다. 또한 이 열에는 레거시 준수 버킷에 대한 정보도 표시됩니다.

   - 지역: 변경할 수 없는 버킷의 영역입니다.
   - 개체 수: 이 버킷의 오브젝트 수입니다.
   - 사용된 공간: 이 버킷에 있는 모든 오브젝트의 논리적 크기입니다. 논리적 크기에는 복제 또는 삭제 코딩 복사본 또는 오브젝트 메타데이터에 필요한 실제 공간이 포함되지 않습니다.
   - 만든 날짜: 버킷을 만든 날짜 및 시간입니다.

   > 표시된 개체 수와 사용된 공간 값은 추정값입니다. 이러한 추정치는 베스트 타이밍, 네트워크 연결 및 노드 상태의 영향을 받습니다. 버킷에 버전 관리가 활성화된 경우 삭제된 개체 버전은 오브젝트 수에 포함됩니다.

3. 버킷의 설정을 보고 관리하려면 버킷 이름을 선택합니다.

   버킷 세부 정보 페이지에서는 버킷 옵션, 버킷 액세스 및 에 대한 설정을 보고 편집할 수 있습니다 플랫폼 서비스.



정합성 보장 수준을 변경합니다

S3 테넌트를 사용하는 경우 테넌트 관리자 또는 테넌트 관리 API를 사용하여 S3 버킷의 오브젝트에 대해 수행된 작업의 정합성 제어를 변경할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이러한 권한은 그룹 또는 버킷 정책의 사용 권한 설정을 재정의합니다. 을 참조하십시오 테넌트 관리 권한.

이 작업에 대해

정합성 보장 레벨은 오브젝트의 가용성과 서로 다른 스토리지 노드 및 사이트에서 이러한 오브젝트의 일관성 간의 균형을 제공합니다. 일반적으로 버킷에 대해 * Read-After-new-write * 정합성 수준을 사용해야 합니다.

새 쓰기 후 읽기 * 정합성 보장 레벨이 클라이언트 애플리케이션의 요구 사항을 충족하지 않는 경우 버킷 정합성 수준을 설정하거나 을 사용하여 정합성 보장 레벨을 변경할 수 있습니다 `Consistency-Control` 머리글. 를 클릭합니다 `Consistency-Control` 헤더는 버킷 정합성 레벨을 오버라이드합니다.

> 버킷의 정합성 수준을 변경하면 변경 후 수집된 객체만 수정된 레벨에 맞게 보장됩니다.

단계

1. 스토리지(S3) * > * 버킷 * 을 선택합니다.
2. 목록에서 버킷 이름을 선택합니다.

버킷 세부 정보 페이지가 나타납니다.

3. 버킷 옵션 * > * 정합성 보장 레벨 * 을 선택합니다.
4. 이 버킷의 오브젝트에 대해 수행된 작업의 정합성 수준을 선택합니다.
   - * 모두 *: 최고 수준의 일관성을 제공합니다. 모든 노드가 데이터를 즉시 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 객체에 대해 읽기-쓰기 후 정합성을 보장하고 객체 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 필요에 따라 만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.
5. 변경 내용 저장 * 을 선택합니다.

마지막 액세스 시간 업데이트를 사용하거나 사용하지 않도록 설정합니다

그리드 관리자가 StorageGRID 시스템에 대한 ILM(정보 수명 주기 관리) 규칙을 만들 때 오브젝트의 마지막 액세스 시간을 사용하여 해당 오브젝트를 다른 스토리지 위치로 이동할지 여부를 결정하도록 선택적으로 지정할 수 있습니다. S3 테넌트를 사용하는 경우 S3 버킷의 오브젝트에 대한 마지막 액세스 시간 업데이트를 활성화하여 이러한 규칙을 활용할 수 있습니다.

이 지침은 배치 지침에서 * Last Access Time * 옵션을 사용하는 ILM 규칙을 하나 이상 포함하는 StorageGRID 시스템에만 적용됩니다. StorageGRID 시스템에 이러한 규칙이 포함되어 있지 않으면 이 지침을 무시할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이러한 권한은 그룹 또는 버킷 정책의 사용 권한 설정을 재정의합니다. 을 참조하십시오 테넌트 관리 권한.
- 마지막 액세스 시간 * 은 ILM 규칙에 대한 * 참조 시간 * 배치 명령에 사용할 수 있는 옵션 중 하나입니다. 규칙의 참조 시간을 마지막 액세스 시간으로 설정하면 그리드 관리자는 해당 개체를 마지막으로 검색한 시기(읽기 또는 보기)에 따라 특정 저장소 위치에 개체가 배치되도록 지정할 수 있습니다.

예를 들어, 최근에 본 오브젝트를 더 빠른 스토리지에 유지하기 위해 그리드 관리자는 다음을 지정하는 ILM 규칙을 생성할 수 있습니다.

- 지난 달 동안 검색된 객체는 로컬 스토리지 노드에 남아 있어야 합니다.
- 지난 달에 검색되지 않은 객체는 오프 사이트 위치로 이동해야 합니다.

> 정보 수명 주기 관리를 사용하여 개체를 관리하는 방법에 대한 지침을 참조하십시오.

기본적으로 마지막 액세스 시간에 대한 업데이트는 사용되지 않습니다. StorageGRID 시스템에 * Last Access Time * 옵션을 사용하는 ILM 규칙이 포함되어 있고 이 옵션이 이 버킷의 오브젝트에 적용되도록 하려면 해당 규칙에 지정된 S3 버킷에 대한 마지막 액세스 시간에 대한 업데이트를 활성화해야 합니다.

> 개체가 검색될 때 마지막 액세스 시간을 업데이트하면 특히 작은 개체의 StorageGRID 성능이 저하될 수 있습니다.

StorageGRID는 객체가 검색될 때마다 다음 추가 단계를 수행해야 하므로 마지막 액세스 시간 업데이트 시 성능 영향이 발생합니다.

- 객체를 새 타임스탬프로 업데이트합니다
- ILM 대기열에 개체를 추가하여 현재 ILM 규칙 및 정책에 대해 다시 평가할 수 있습니다

이 표에는 마지막 액세스 시간이 비활성화되거나 활성화될 때 버킷의 모든 오브젝트에 적용되는 동작이 요약되어 있습니다.

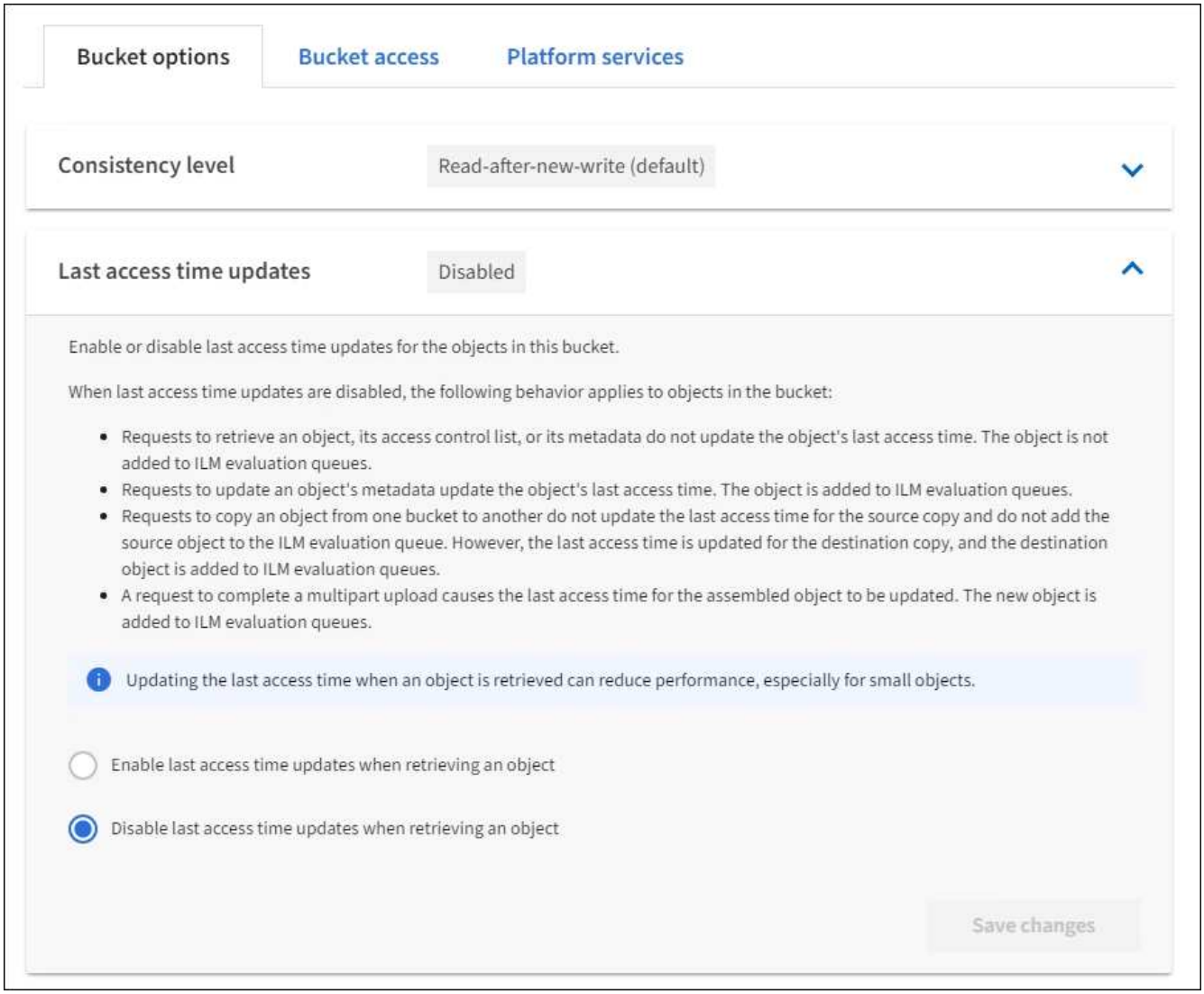
요청 유형입니다	마지막 액세스 시간이 비활성화된 경우의 동작(기본값)		마지막 액세스 시간이 설정된 경우의 동작	
	마지막 액세스 시간이 업데이트되었습니까?	ILM 평가 대기열에 객체가 추가되었습니까?	마지막 액세스 시간이 업데이트되었습니까?	ILM 평가 대기열에 객체가 추가되었습니까?
개체, 해당 액세스 제어 목록 또는 해당 메타데이터를 검색하는 요청입니다	아니요	아니요	예	예
개체의 메타데이터를 업데이트하도록 요청합니다	예	예	예	예
한 버킷에서 다른 버킷으로 오브젝트 복사 요청	• 아니요, 소스 복제본입니다 • 예, 대상 복사본에 대해 입니다	• 아니요, 소스 복제본입니다 • 예, 대상 복사본에 대해 입니다	• 예. 소스 복제본에 대해 가능합니다 • 예, 대상 복사본에 대해 입니다	• 예. 소스 복제본에 대해 가능합니다 • 예, 대상 복사본에 대해 입니다
여러 부분 업로드를 완료하도록 요청합니다	예. 조립된 개체에 대해 가능합니다	예. 조립된 개체에 대해 가능합니다	예. 조립된 개체에 대해 가능합니다	예. 조립된 개체에 대해 가능합니다

단계

1. 스토리지(S3) * > * 버킷 * 을 선택합니다.
2. 목록에서 버킷 이름을 선택합니다.

   버킷 세부 정보 페이지가 나타납니다.

3. Bucket options * > * Last access time updates * 를 선택합니다.
4. 마지막 액세스 시간 업데이트를 활성화하거나 비활성화하려면 해당 라디오 버튼을 선택합니다.



5. 변경 내용 저장 * 을 선택합니다.

관련 정보

테넌트 관리 권한

ILM을 사용하여 개체를 관리합니다

버킷의 오브젝트 버전 관리를 변경합니다

S3 테넌트를 사용하는 경우 테넌트 관리자 또는 테넌트 관리 API를 사용하여 S3 버킷의 버전 관리 상태를 변경할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인했습니다 지원되는 웹 브라우저.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 사용 권한 설정을 재정의합니다.

  테넌트 관리 권한

이 작업에 대해

버킷에 대한 오브젝트 버전 관리를 설정하거나 일시 중지할 수 있습니다. 버켓에 대한 버전 관리를 활성화한 후에는 버전이 지정되지 않은 상태로 돌아갈 수 없습니다. 그러나 버킷의 버전 관리를 일시 중단할 수 있습니다.

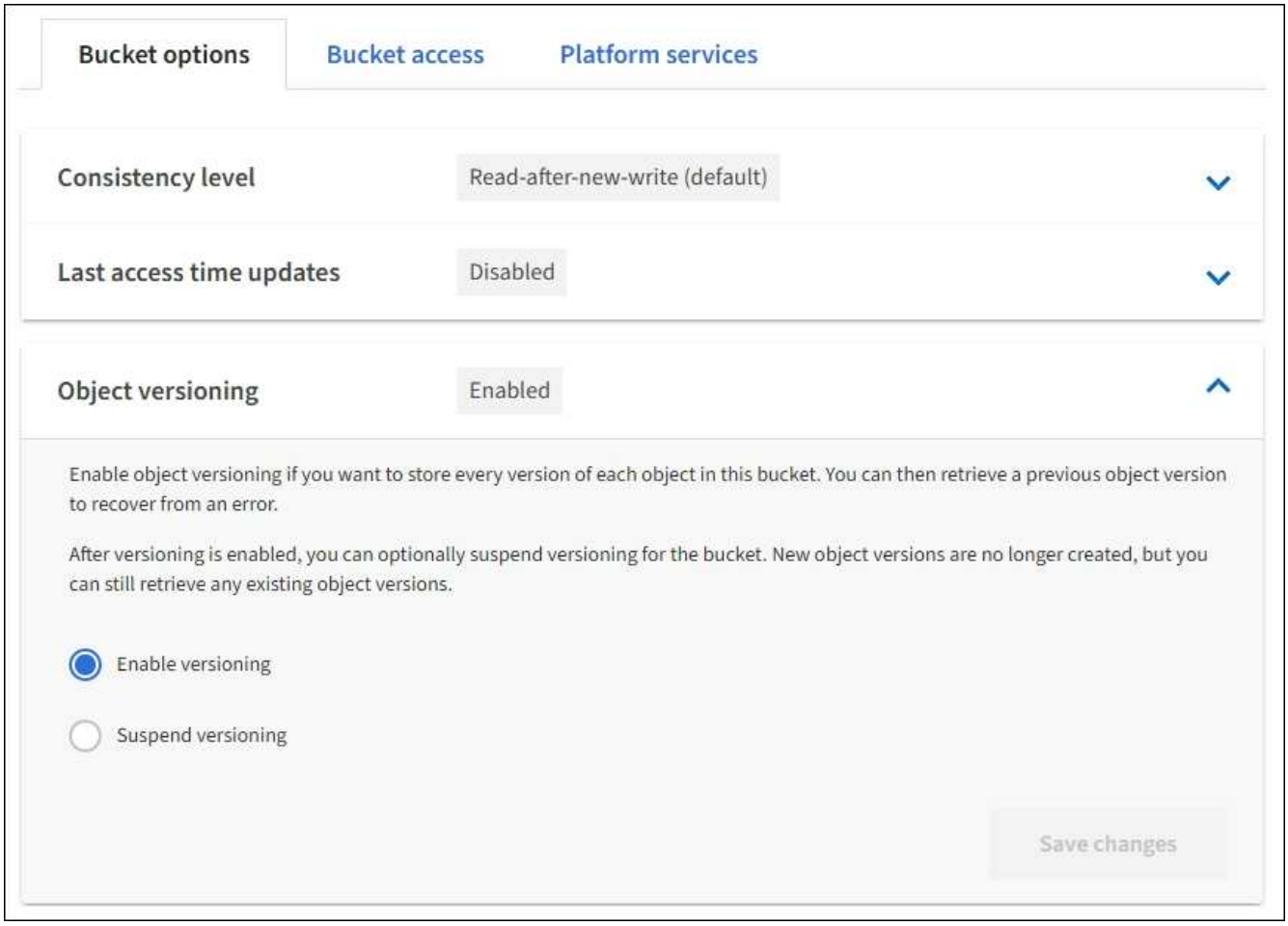
- 사용 안 함: 버전 관리가 활성화되지 않았습니다
- 사용: 버전 관리가 활성화됩니다
- 일시 중단됨: 버전 관리가 이전에 활성화되었으며 일시 중단되었습니다

S3 오브젝트 버전 관리

S3 버전 오브젝트 ILM 규칙 및 정책(예 4)

단계

1. 스토리지(S3) * > * 버킷 * 을 선택합니다.
2. 목록에서 버킷 이름을 선택합니다.
3. 버킷 옵션 * > * 오브젝트 버전 관리 * 를 선택합니다.



4. 이 버킷의 오브젝트에 대한 버전 관리 상태를 선택합니다.

> S3 오브젝트 잠금 또는 레거시 규정 준수를 활성화하면 * 오브젝트 버전 관리 * 옵션이 비활성화됩니다.

옵션을 선택합니다	설명
버전 관리를 활성화합니다	이 버킷에 각 오브젝트의 모든 버전을 저장하려면 오브젝트 버전 관리를 활성화하십시오. 그런 다음 필요에 따라 개체의 이전 버전을 검색할 수 있습니다. 버킷에 이미 있던 객체는 사용자가 수정할 때 버전이 적용됩니다.
버전 관리를 일시 중단합니다	새 개체 버전을 더 이상 만들지 않으려면 개체 버전 관리를 일시 중단합니다. 기존 개체 버전을 검색할 수 있습니다.

5. 변경 내용 저장 * 을 선택합니다.

CORS(Cross-Origin Resource Sharing) 구성

다른 도메인의 웹 애플리케이션에서 해당 버킷의 버킷 및 오브젝트에 액세스할 수 있도록 하려면 S3 버킷에 대해 CORS(Cross-Origin Resource Sharing)를 구성할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이러한 권한은 그룹 또는 버킷 정책의 사용 권한 설정을 재정의합니다.

이 작업에 대해

CORS(Cross-Origin Resource Sharing)는 한 도메인의 클라이언트 웹 애플리케이션이 다른 도메인의 리소스에 액세스할 수 있도록 하는 보안 메커니즘입니다. 예를 들어, 그래픽을 저장하기 위해 "이미지"라는 S3 버킷을 사용한다고 가정합니다. 영상물통용 CORS를 구성하여 해당 버켓의 영상을 웹사이트 http://www.example.com` 에 표시할 수 있습니다.

단계

1. 텍스트 편집기를 사용하여 CORS를 활성화하는 데 필요한 XML을 만듭니다.

   이 예에서는 S3 버킷에 대해 CORS를 활성화하는 데 사용되는 XML을 보여 줍니다. 이 XML을 사용하면 모든 도메인이 버킷에 GET 요청을 보낼 수 있지만 "http://www.example.com` 도메인에서만 POST 및 삭제 요청을 보낼 수 있습니다. 모든 요청 헤더가 허용됩니다.

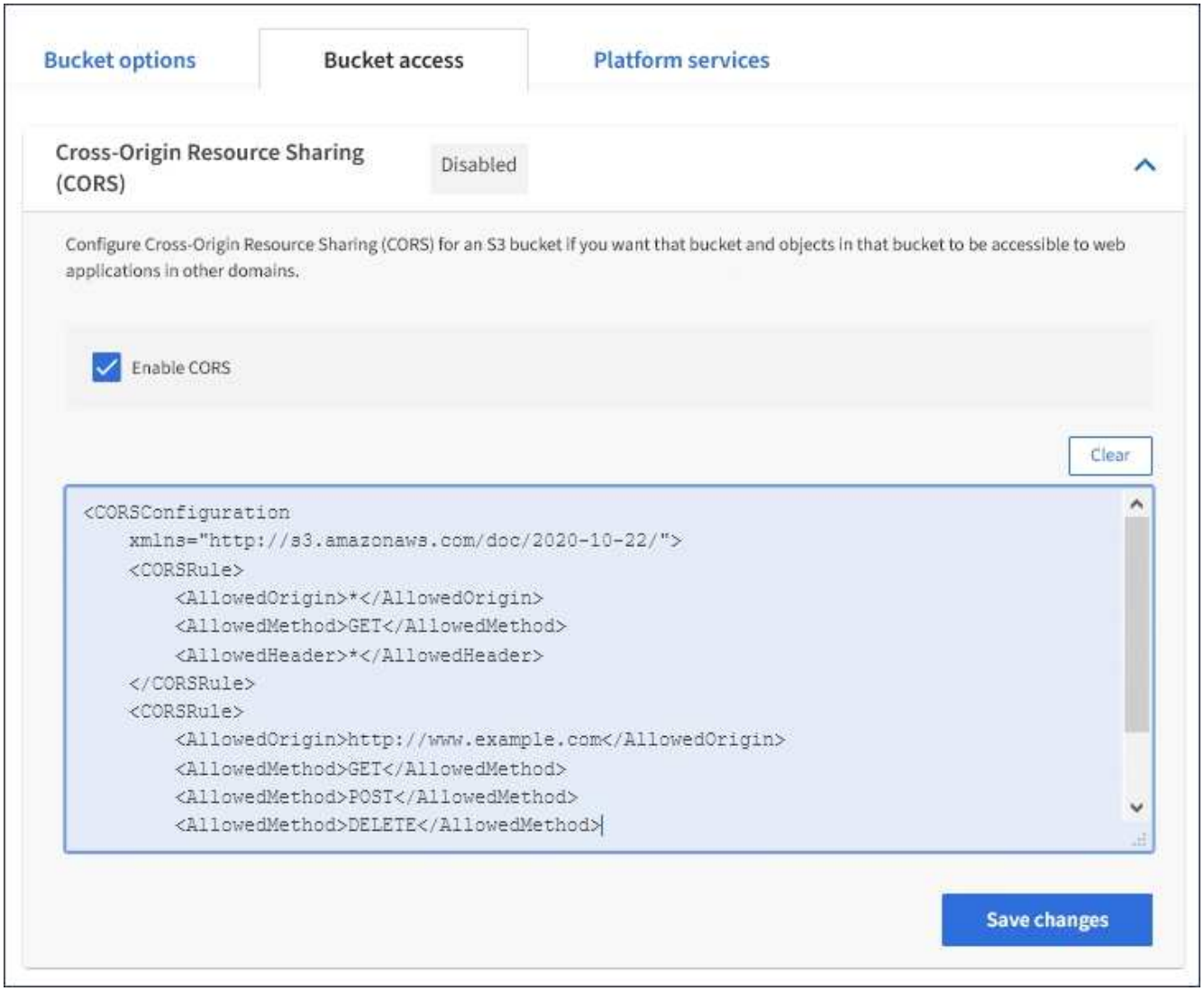
```
<CORSConfiguration
    xmlns="http://s3.amazonaws.com/doc/2020-10-22/">
    <CORSRule>
        <AllowedOrigin>*</AllowedOrigin>
        <AllowedMethod>GET</AllowedMethod>
        <AllowedHeader>*</AllowedHeader>
    </CORSRule>
    <CORSRule>
        <AllowedOrigin>http://www.example.com</AllowedOrigin>
        <AllowedMethod>GET</AllowedMethod>
        <AllowedMethod>POST</AllowedMethod>
        <AllowedMethod>DELETE</AllowedMethod>
        <AllowedHeader>*</AllowedHeader>
    </CORSRule>
</CORSConfiguration>
```

CORS 구성 XML에 대한 자세한 내용은 을 참조하십시오 "AWS(Amazon Web Services) 문서: Amazon Simple Storage Service 개발자 가이드 를 참조하십시오".

2. 테넌트 관리자에서 * 스토리지(S3) * > * 버킷 * 을 선택합니다.
3. 목록에서 버킷 이름을 선택합니다.

   버킷 세부 정보 페이지가 나타납니다.

4. Bucket access * > * Cross-Origin Resource Sharing (CORS) * 를 선택합니다.
5. CORS * 활성화 확인란을 선택합니다.
6. 텍스트 상자에 CORS 구성 XML을 붙여 넣고 * 변경 내용 저장 * 을 선택합니다.



7. 버킷의 CORS 설정을 수정하려면 텍스트 상자에서 CORS 구성 XML을 업데이트하거나 다시 시작하려면 * Clear * 를 선택하십시오. 그런 다음 * 변경 사항 저장 * 을 선택합니다.
8. 버킷에 대한 CORS를 비활성화하려면 * CORS * 활성화 확인란의 선택을 취소한 다음 * 변경 사항 저장 * 을 선택합니다.

S3 버킷을 삭제합니다

테넌트 관리자를 사용하여 비어 있는 하나 이상의 S3 버킷을 삭제할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이러한 권한은 그룹 또는 버킷 정책의 사용 권한 설정을 재정의합니다. 을 참조하십시오 테넌트 관리 권한.
- 삭제할 버킷이 비어 있습니다.

이 작업에 대해

다음 지침은 Tenant Manager를 사용하여 S3 버킷을 삭제하는 방법을 설명합니다. 를 사용하여 S3 버킷을 삭제할 수도 있습니다 테넌트 관리 API 또는 을 누릅니다 S3 REST API.

오브젝트 또는 비최신 오브젝트 버전이 포함된 S3 버킷을 삭제할 수 없습니다. S3 버전 오브젝트를 삭제하는 방법에 대한 자세한 내용은 를 참조하십시오 정보 수명 주기 관리를 사용하여 개체를 관리하기 위한 지침.

단계

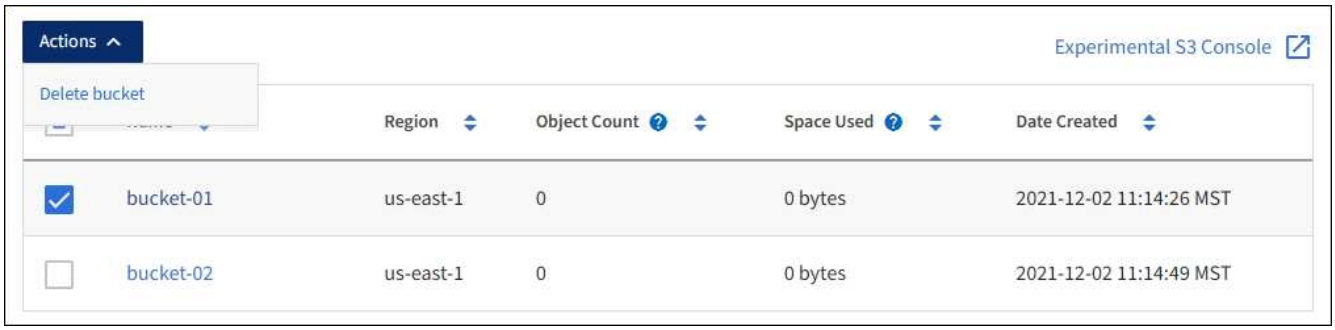
1. 스토리지(S3) * > * 버킷 * 을 선택합니다.

   Bucket 페이지가 나타나고 기존의 모든 S3 버킷을 표시합니다.

2. 삭제할 빈 버킷의 확인란을 선택합니다. 한 번에 둘 이상의 버킷을 선택할 수 있습니다.

   작업 메뉴가 활성화됩니다.

3. 작업 메뉴에서 * 버킷 삭제 * (또는 둘 이상을 선택한 경우 * 버킷 삭제 *)를 선택합니다.



4. 확인 대화 상자가 나타나면 * 예 * 를 선택하여 선택한 모든 버킷을 삭제합니다.

   StorageGRID는 각 버킷이 비어 있음을 확인한 다음 각 버킷을 삭제합니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

   버킷이 비어 있지 않으면 오류 메시지가 나타납니다. 버킷을 삭제하려면 먼저 모든 오브젝트를 삭제해야 합니다.

Experimental S3 Console을 사용합니다

S3 콘솔을 사용하여 S3 버킷의 오브젝트를 볼 수 있습니다.

S3 콘솔을 사용하여 다음을 수행할 수도 있습니다.

- 개체, 개체 버전 및 폴더를 추가하고 삭제합니다
- 개체 이름을 바꿉니다
- 버킷 및 폴더 간에 오브젝트를 이동 및 복사합니다
- 오브젝트 태그 관리
- 개체 메타데이터를 봅니다
- 객체를 다운로드합니다

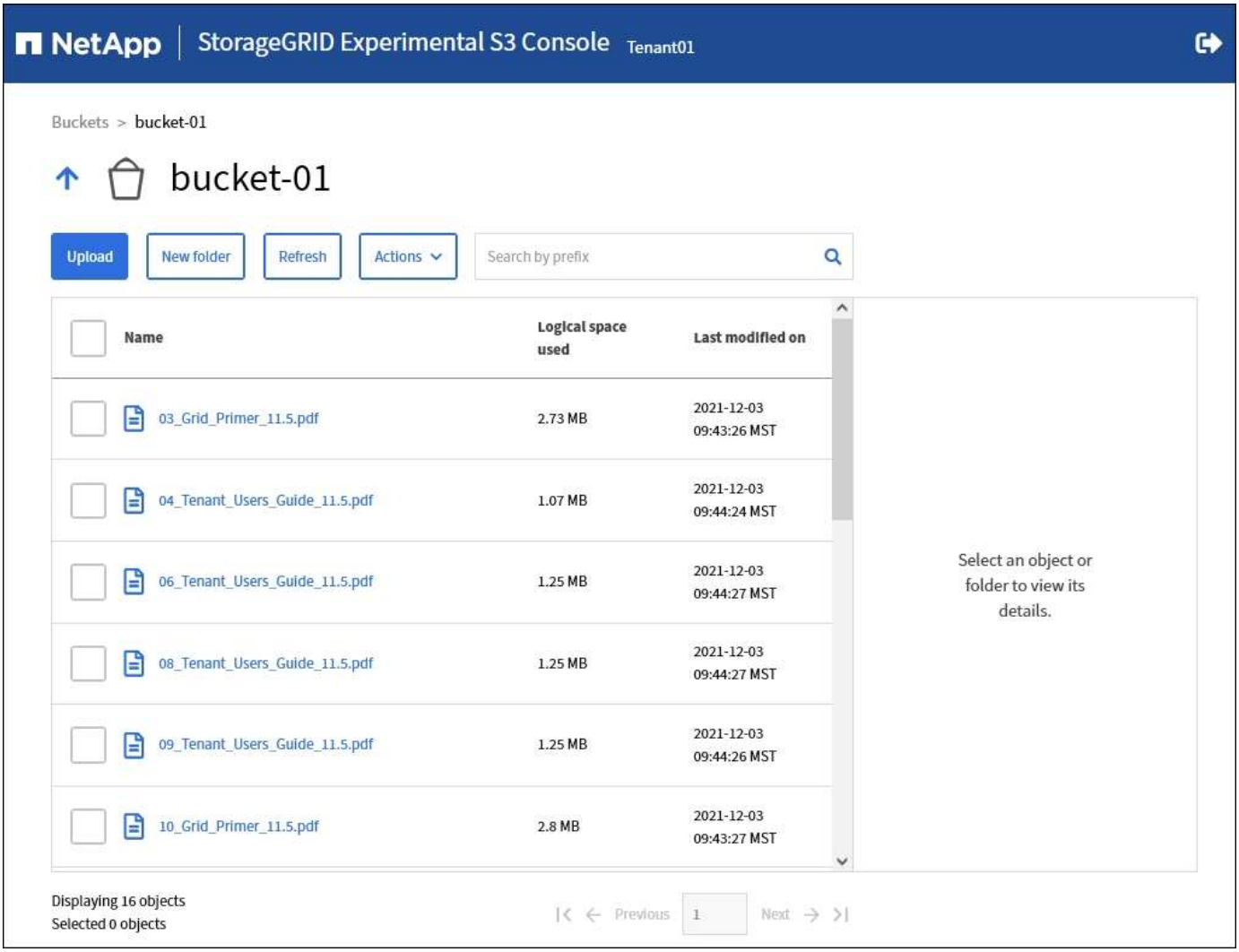
> S3 콘솔이 완전히 테스트되지 않았으며 "Experimental(실험)"으로 표시됩니다. 대량의 오브젝트 관리 또는 운영 환경에서 사용하기 위한 것이 아닙니다. 테넌트는 새로운 ILM 정책을 시뮬레이션하기 위해 개체를 업로드할 때, 수집 문제 해결 또는 개념 증명 또는 비운영 그리드를 사용하는 경우와 같이 소수의 개체에 대한 기능을 수행할 때만 S3 콘솔을 사용해야 합니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인했습니다 지원되는 웹 브라우저.
- 자신의 S3 자격 증명 관리 권한이 있습니다.
- 버킷을 만들었습니다.
- 사용자의 액세스 키 ID와 비밀 액세스 키를 알고 있습니다. 필요한 경우 이 정보가 포함된 '.csv' 파일을 사용할 수 있습니다. 를 참조하십시오 액세스 키 생성에 대한 지침.

단계

1. Bucket * 을 선택합니다.
2. 를 선택합니다 Experimental S3 Console . 버킷 세부 정보 페이지에서 이 링크에 액세스할 수도 있습니다.
3. Experimental S3 Console 로그인 페이지에서 액세스 키 ID 및 비밀 액세스 키를 필드에 붙여 넣습니다. 그렇지 않으면 * 업로드 액세스 키 * 를 선택하고 '.csv' 파일을 선택합니다.
4. 로그인 * 을 선택합니다.
5. 필요에 따라 오브젝트 관리



S3 플랫폼 서비스 관리

플랫폼 서비스란 무엇입니까?

StorageGRID 플랫폼 서비스는 하이브리드 클라우드 전략을 구현하는 데 도움이 될 수 있습니다.

테넌트 계정에 플랫폼 서비스를 사용할 수 있는 경우 모든 S3 버킷에 대해 다음 서비스를 구성할 수 있습니다.

- * CloudMirror 복제 * StorageGRID CloudMirror 복제 서비스입니다 StorageGRID 버킷에서 지정된 외부 대상으로 특정 오브젝트를 미러링하는 데 사용됩니다.

  예를 들어, CloudMirror 복제를 사용하여 특정 고객 레코드를 Amazon S3에 미러링한 다음 AWS 서비스를 활용하여 데이터에 대한 분석을 수행할 수 있습니다.

  > 소스 버킷에 S3 오브젝트 잠금이 설정된 경우 CloudMirror 복제가 지원되지 않습니다.

- * 알림 *: 버킷당 이벤트 알림 지정된 외부 SNS(Amazon Simple Notification Service ™)에 객체에 대해 수행된 특정 작업에 대한 알림을 보내는 데 사용됩니다.

  예를 들어, 버킷에 추가된 각 오브젝트에 대해 관리자에게 경고가 전송되도록 구성할 수 있습니다. 여기서 객체는

중요한 시스템 이벤트와 연결된 로그 파일을 나타냅니다.

> S3 오브젝트 잠금이 활성화된 버킷에서 이벤트 알림을 구성할 수 있지만 오브젝트의 S3 오브젝트 잠금 메타데이터(마지막 보존 날짜 및 법적 보류 상태 포함)는 알림 메시지에 포함되지 않습니다.

- * 통합 서비스 검색 * 검색 통합 서비스 외부 서비스를 사용하여 메타데이터를 검색하거나 분석할 수 있는 지정된 Elasticsearch 인덱스에 S3 개체 메타데이터를 전송하는 데 사용됩니다.

  예를 들어, S3 오브젝트 메타데이터를 원격 Elasticsearch 서비스로 전송하도록 버킷을 구성할 수 있습니다. 그런 다음 Elasticsearch를 사용하여 버킷에 대한 검색을 수행하고 객체 메타데이터에 있는 패턴에 대한 정교한 분석을 수행할 수 있습니다.

> S3 오브젝트 잠금이 활성화된 버킷에서 Elasticsearch 통합을 구성할 수 있지만 오브젝트의 S3 오브젝트 잠금 메타데이터(보존 기한 및 법적 보류 상태 포함)는 알림 메시지에 포함되지 않습니다.

플랫폼 서비스의 대상 위치는 일반적으로 StorageGRID 구축과 외부적이기 때문에 플랫폼 서비스는 데이터에 대한 외부 스토리지 리소스, 알림 서비스 및 검색 또는 분석 서비스를 사용하여 얻을 수 있는 성능과 유연성을 제공합니다.

단일 S3 버킷에 대해 모든 플랫폼 서비스 조합을 구성할 수 있습니다. 예를 들어, StorageGRID S3 버킷에서 CloudMirror 서비스 및 알림을 모두 구성하여 특정 오브젝트를 Amazon Simple Storage Service에 미러링하고 이러한 각 오브젝트에 대한 알림을 타사 모니터링 애플리케이션에 전송하여 AWS 비용을 추적할 수 있도록 할 수 있습니다.

> 그리드 관리자 또는 그리드 관리 API를 사용하여 StorageGRID 관리자가 각 테넌트 계정에 대해 플랫폼 서비스 사용을 활성화해야 합니다.

플랫폼 서비스 구성 방법

플랫폼 서비스는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성한 외부 엔드포인트와 통신합니다. 각 엔드포인트는 StorageGRID S3 버킷, Amazon 웹 서비스 버킷, SNS(Simple Notification Service) 주제 또는 로컬, AWS 또는 기타 위치에서 호스팅되는 Elasticsearch 클러스터와 같은 외부 대상을 나타냅니다.

끝점을 만든 후 버킷에 XML 구성을 추가하여 버킷에 대한 플랫폼 서비스를 활성화할 수 있습니다. XML 구성은 버킷이 작업해야 하는 오브젝트, 버킷이 취해야 하는 조치 및 버킷이 서비스에 사용해야 하는 엔드포인트를 식별합니다.

구성할 각 플랫폼 서비스에 대해 별도의 XML 구성을 추가해야 합니다. 예를 들면 다음과 같습니다.

1. 키가 '/images'로 시작하는 모든 오브젝트를 Amazon S3 버킷에 복제하려면 소스 버킷에 복제 구성을 추가해야 합니다.
2. 이러한 객체가 버킷에 저장될 때 알림을 보내려면 알림 구성을 추가해야 합니다.
3. 마지막으로 이러한 개체의 메타데이터를 인덱싱하려면 검색 통합을 구현하는 데 사용되는 메타데이터 알림 구성을 추가해야 합니다.

구성 XML의 형식은 StorageGRID 플랫폼 서비스를 구현하는 데 사용되는 S3 REST API를 통해 제어됩니다.

플랫폼 서비스	**S3 REST API**
CloudMirror 복제	• 버킷 복제를 가져옵니다 • 버킷 복제를 배치합니다

플랫폼 서비스	**S3 REST API**
알림	• 버킷 알림을 받습니다 • 버킷 통지를 보냅니다
검색 통합	• Bucket 메타데이터 알림 구성 가져오기 • Put Bucket 메타데이터 알림 구성 이러한 작업은 StorageGRID에 맞게 맞춤형으로 제공됩니다.

StorageGRID에서 이러한 API를 구축하는 방법에 대한 자세한 내용은 S3 클라이언트 애플리케이션 구현 지침을 참조하십시오.

관련 정보

플랫폼 서비스 사용에 대한 고려 사항

S3을 사용합니다

CloudMirror 복제 서비스

StorageGRID가 버킷에 추가된 지정된 오브젝트를 하나 이상의 대상 버킷에 복제하도록 하려면 S3 버킷에 대해 CloudMirror 복제를 활성화할 수 있습니다.

CloudMirror 복제는 그리드의 활성 ILM 정책과 독립적으로 작동합니다. CloudMirror 서비스는 소스 버킷에 저장된 객체를 복제하여 가능한 한 빨리 대상 버킷에 제공합니다. 오브젝트 수집의 성공 시 복제된 오브젝트 제공이 트리거됩니다.

기존 버킷에 대해 CloudMirror 복제를 설정하면 해당 버킷에 추가된 새 객체만 복제됩니다. 버킷의 기존 객체는 복제되지 않습니다. 기존 오브젝트의 복제를 강제로 수행하려면 오브젝트 복사를 수행하여 기존 오브젝트의 메타데이터를 업데이트할 수 있습니다.

> CloudMirror 복제를 사용하여 오브젝트를 AWS S3 대상으로 복사하는 경우 Amazon S3는 각 PUT 요청 헤더 내의 사용자 정의 메타데이터 크기를 2KB로 제한합니다. 객체에 2KB보다 큰 사용자 정의 메타데이터가 있는 경우 해당 객체가 복제되지 않습니다.

StorageGRID에서는 단일 버킷의 오브젝트를 여러 개의 대상 버킷으로 복제할 수 있습니다. 이렇게 하려면 복제 구성 XML에서 각 규칙의 대상을 지정합니다. 객체를 둘 이상의 버킷에 동시에 복제할 수 없습니다.

또한 버전 관리되거나 버전이 지정되지 않은 버킷에서 CloudMirror 복제를 구성하고 버전 관리되거나 버전이 지정되지 않은 버킷을 대상으로 지정할 수 있습니다. 버전 및 비버전 버킷의 모든 조합을 사용할 수 있습니다. 예를 들어 버전이 지정되지 않은 소스 버킷의 대상으로 버전 관리가 지정된 버킷을 지정하거나 그 반대로 지정할 수 있습니다. 버전이 지정되지 않은 버킷 간에 복제할 수도 있습니다.

CloudMirror 복제 서비스의 삭제 동작은 Amazon S3에서 제공하는 CRR(Cross Region Replication) 서비스의 삭제 동작과 같습니다. 소스 버킷에서 객체를 삭제해도 대상에서 복제된 객체는 삭제되지 않습니다. 소스 및 대상 버킷의 버전이 모두 지정된 경우 삭제 마커가 복제됩니다. 대상 버킷의 버전이 지정되지 않은 경우 소스 버킷에서 오브젝트를 삭제해도 삭제 마커가 대상 버킷에 복제되거나 대상 오브젝트가 삭제되지 않습니다.

객체가 대상 버킷에 복제되면 StorageGRID는 객체를 "replicas"로 표시합니다. 대상 StorageGRID 버킷은 복제본으로

표시된 객체를 다시 복제하지 않으므로 실수로 인한 복제 루프로부터 보호됩니다. 이 복제 마크는 StorageGRID 내부에 있으며 Amazon S3 버킷을 대상으로 사용할 때 AWS CRR을 활용하는 것을 방지하지 않습니다.

> 복제본을 표시하는 데 사용되는 사용자 지정 헤더는 X-NTAP-sg-replica입니다. 이 표시는 계단식 미러를 방지합니다. StorageGRID는 두 그리드 간의 양방향 CloudMirror를 지원합니다.

대상 버킷의 이벤트의 고유성과 순서는 보장되지 않습니다. 전송 성공을 보장하기 위해 수행된 작업의 결과로 소스 객체의 동일한 복제본이 두 개 이상 대상에 제공될 수 있습니다. 드물지만 둘 이상의 서로 다른 StorageGRID 사이트에서 동일한 객체가 동시에 업데이트되는 경우 대상 버킷의 작업 순서가 소스 버킷의 이벤트 순서와 일치하지 않을 수 있습니다.

CloudMirror 복제는 일반적으로 외부 S3 버킷을 대상으로 사용하도록 구성됩니다. 그러나 다른 StorageGRID 배포나 S3 호환 서비스를 사용하도록 복제를 구성할 수도 있습니다.

버킷에 대한 알림을 이해합니다

StorageGRID에서 지정된 이벤트에 대한 알림을 대상 SNS(Amazon Simple Notification Service)로 보내도록 하려면 S3 버킷에 대한 이벤트 알림을 활성화할 수 있습니다.

가능합니다 이벤트 알림을 구성합니다 알림 구성 XML을 소스 버킷과 연결합니다. 알림 구성 XML은 버킷 알림을 구성하기 위한 S3 규칙을 따르고, 엔드포인트의 URN으로 지정된 대상 SNS 항목을 따릅니다.

이벤트 알림은 알림 구성에 지정된 대로 소스 버킷에서 생성되며 대상으로 전달됩니다. 개체와 관련된 이벤트가 성공하면 해당 이벤트에 대한 알림이 생성되고 배달 대기 상태가 됩니다.

알림의 고유성과 순서는 보장되지 않습니다. 전송 성공을 보장하기 위해 수행된 작업의 결과로 하나 이상의 이벤트 알림이 대상에 전달될 수 있습니다. 그리고 납품이 비동기식이기 때문에, 특히 서로 다른 StorageGRID 사이트에서 발생하는 작업의 경우, 대상에서 알림의 시간 순서가 소스 버킷의 이벤트 순서와 일치한다고 보장할 수 없습니다. 이벤트 메시지에서 '시퀀스' 키를 사용하여 Amazon S3 문서에 설명된 대로 특정 객체에 대한 이벤트 순서를 결정할 수 있습니다.

지원되는 알림 및 메시지

StorageGRID 이벤트 알림은 Amazon S3 API를 따르며 다음과 같은 제한 사항이 적용됩니다.

- 다음 이벤트 유형에 대한 알림을 구성할 수 없습니다. 이러한 이벤트 유형은 * 지원되지 않습니다 *.
  - '3: RedundancyLostObject'를 선택합니다
  - '3:ObjectRestore:완료됨'
- StorageGRID에서 보낸 이벤트 알림은 표에 나와 있는 것처럼 일부 키를 포함하지 않고 다른 키에 대해 특정 값을 사용한다는 점을 제외하고 표준 JSON 형식을 사용합니다.

키 이름	**StorageGRID** 값
이벤트 소스	전쟁포로 S3
awsRegion	포함되지 않음
X-amz-id-2	포함되지 않음

키 이름	**StorageGRID** 값
ARN	"urn:SGWs:S3::bucket_name'

검색 통합 서비스를 이해합니다

오브젝트 메타데이터에 외부 검색 및 데이터 분석 서비스를 사용하려는 경우 S3 버킷에 대한 검색 통합을 활성화할 수 있습니다.

검색 통합 서비스는 오브젝트 또는 해당 메타데이터가 업데이트될 때마다 자동으로 그리고 비동기적으로 S3 오브젝트 메타데이터를 대상 끝점에 보내는 사용자 지정 StorageGRID 서비스입니다. 그런 다음 대상 서비스에서 제공하는 정교한 검색, 데이터 분석, 시각화 또는 머신 러닝 도구를 사용하여 오브젝트 데이터를 검색, 분석 및 분석할 수 있습니다.

버전 관리되거나 버전이 지정되지 않은 모든 버킷에 대해 검색 통합 서비스를 활성화할 수 있습니다. 검색 통합은 메타데이터 알림 구성 XML을 작업할 개체 및 개체 메타데이터에 대한 대상을 지정하는 버킷과 연결하여 구성됩니다.

알림은 버킷 이름, 오브젝트 이름 및 버전 ID(있는 경우)로 명명된 JSON 문서의 형식으로 생성됩니다. 각 메타데이터 알림에는 개체의 모든 태그 및 사용자 메타데이터 외에도 개체에 대한 표준 시스템 메타데이터 세트가 포함되어 있습니다.

> 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.

알림은 다음과 같은 경우에 생성되고 전송 대기 상태가 됩니다.

- 객체가 생성됩니다.
- 그리드의 ILM 정책 작업으로 인해 오브젝트가 삭제된 경우를 포함하여 오브젝트가 삭제됩니다.
- 오브젝트 메타데이터 또는 태그가 추가, 업데이트 또는 삭제됩니다. 메타데이터 및 태그의 전체 집합은 항상 변경된 값뿐만 아니라 업데이트 시 전송됩니다.

메타데이터 알림 구성 XML을 버킷에 추가하면 생성한 새 개체 및 데이터, 사용자 메타데이터 또는 태그를 업데이트하여 수정하는 모든 개체에 대한 알림이 전송됩니다. 그러나 버킷에 이미 있는 객체에 대해서는 알림이 전송되지 않습니다. 버킷의 모든 오브젝트에 대한 오브젝트 메타데이터가 대상으로 전송되도록 하려면 다음 중 하나를 수행해야 합니다.

- 버킷을 생성한 후 개체를 추가하기 전에 즉시 검색 통합 서비스를 구성합니다.
- 메타데이터 알림 메시지가 대상으로 전송되도록 버킷에 이미 있는 모든 객체에 대해 작업을 수행합니다.

StorageGRID 검색 통합 서비스는 Elasticsearch 클러스터를 대상으로 지원합니다. 다른 플랫폼 서비스와 마찬가지로 대상은 서비스의 구성 XML에서 URN이 사용되는 끝점에서 지정됩니다. 를 사용합니다 "NetApp 상호 운용성 매트릭스 툴" 지원되는 Elasticsearch 버전을 확인합니다.

관련 정보

검색 통합을 위한 구성 XML

메타데이터 알림에 포함된 개체 메타데이터입니다

JSON이 검색 통합 서비스에 의해 생성되었습니다

검색 통합 서비스를 구성합니다

플랫폼 서비스 사용에 대한 고려 사항

플랫폼 서비스를 구현하기 전에 이러한 서비스를 사용하기 위한 권장 사항 및 고려 사항을 검토하십시오.

S3에 대한 자세한 내용은 을 참조하십시오 S3을 사용합니다.

플랫폼 서비스 사용에 대한 고려 사항

고려 사항	세부 정보
대상 엔드포인트 모니터링	각 대상 끝점의 가용성을 모니터링해야 합니다. 대상 끝점에 대한 연결이 오랜 시간 동안 손실되고 요청의 백로그가 많은 경우 StorageGRID에 대한 추가 클라이언트 요청(예: PUT 요청)이 실패합니다. 엔드포인트에 연결할 수 있게 되면 실패한 요청을 다시 시도해야 합니다.
대상 끝점 임계치 조절	요청이 전송되는 속도가 대상 엔드포인트에서 요청을 수신할 수 있는 속도를 초과하는 경우 StorageGRID 소프트웨어는 버킷에 대한 수신 S3 요청을 스로틀할 수 있습니다. 임계치 조절은 대상 끝점으로 보내려고 기다리는 요청의 백로그가 있는 경우에만 발생합니다. 단, 들어오는 S3 요청의 실행 시간이 더 오래 걸린다는 점을 알 수 있습니다. 속도가 현저히 느린 성능을 감지하기 시작하는 경우 수집 속도를 줄이거나 용량이 더 큰 엔드포인트를 사용해야 합니다. 요청 백로그가 계속 증가하는 경우 PUT 요청과 같은 클라이언트 S3 작업이 결국 실패합니다. CloudMirror 요청은 일반적으로 검색 통합 또는 이벤트 알림 요청보다 더 많은 데이터 전송을 포함하므로 대상 엔드포인트의 성능에 영향을 받을 가능성이 더 높습니다.
주문 보증	StorageGRID은 사이트 내의 개체에 대한 작업을 주문할 수 있도록 보장합니다. 객체에 대한 모든 작업이 동일한 사이트 내에 있는 한 최종 객체 상태(복제의 경우)는 항상 StorageGRID의 상태와 동일합니다. StorageGRID는 StorageGRID 사이트 전체에서 작업이 수행되는 경우 요청을 주문하기 위해 최선의 노력을 다하고 있습니다. 예를 들어 처음에 사이트 A에 오브젝트를 작성한 다음 나중에 사이트 B에서 동일한 오브젝트를 덮어쓰는 경우 CloudMirror에서 대상 버킷에 복제한 최종 오브젝트는 새로운 오브젝트일 수 없습니다.
ILM 기반 오브젝트 삭제	AWS CRR 및 SNS 서비스의 삭제 동작과 일치하도록 StorageGRID ILM 규칙 때문에 소스 버킷의 객체가 삭제될 때 CloudMirror 및 이벤트 알림 요청이 전송되지 않습니다. 예를 들어 ILM 규칙이 14일 후에 개체를 삭제하는 경우 CloudMirror 또는 이벤트 알림 요청이 전송되지 않습니다. 반면, 검색 통합 요청은 ILM로 인해 객체가 삭제될 때 전송됩니다.

CloudMirror 복제 서비스 사용에 대한 고려 사항

고려 사항	세부 정보
복제 상태입니다	StorageGRID는 X-amz-replication-status 헤더를 지원하지 않습니다.
개체 크기	CloudMirror 복제 서비스를 통해 대상 버킷에 복제할 수 있는 개체의 최대 크기는 5TiB이며, 이는 maximum_supported_object 크기와 같습니다. • 참고 *: 단일 PUT 오브젝트 작업에 대한 maximum_recommended_size는 5GiB(5,368,709,120바이트)입니다. 5GiB보다 큰 객체가 있는 경우 대신 멀티파트 업로드를 사용합니다.
버킷 버전 관리 및 버전 ID	StorageGRID의 소스 S3 버킷에서 버전 관리가 활성화된 경우 대상 버킷의 버전 관리도 활성화해야 합니다. 버전 관리를 사용할 때는 S3 프로토콜의 제한으로 인해 대상 버킷에서 오브젝트 버전 순서가 CloudMirror 서비스에 의해 보장되지 않는 것이 가장 좋습니다. • 참고 *: StorageGRID의 소스 버킷에 대한 버전 ID는 대상 버킷의 버전 ID와 관련이 없습니다.
개체 버전에 태그 달기	CloudMirror 서비스는 S3 프로토콜의 제한으로 인해 버전 ID를 제공하는 Put Object 태그 지정 또는 Delete Object 태그 지정 요청을 복제하지 않습니다. 소스 및 대상의 버전 ID는 관련이 없으므로 특정 버전 ID에 대한 태그 업데이트를 복제할 수 없습니다. 반면 CloudMirror 서비스는 Put Object 태그 지정 요청을 복제하거나 버전 ID를 지정하지 않는 Object 태그 지정 요청을 삭제합니다. 이러한 요청은 최신 키의 태그(또는 버킷의 버전이 지정된 경우 최신 버전)를 업데이트합니다. 태그가 있는 일반 베스트(업데이트 태그 지정 안 함)도 복제됩니다.
멀티파트 업로드 및 'ETag' 값	여러 부분 업로드를 사용하여 업로드한 개체를 미러링할 때 CloudMirror 서비스는 해당 파트를 보존하지 않습니다. 따라서 대칭 복사된 오브젝트의 ETag 값은 원래 오브젝트의 ETag 값과 다릅니다.
SSE-C로 암호화된 오브젝트(고객이 제공한 키를 사용한 서버측 암호화)	CloudMirror 서비스는 SSE-C로 암호화된 객체를 지원하지 않습니다 CloudMirror 복제를 위해 소스 버킷으로 객체를 수집하려고 하고 요청에 SSE-C 요청 헤더가 포함된 경우 작업이 실패합니다.
S3 오브젝트 잠금이 활성화된 버킷	CloudMirror 복제에 대한 대상 S3 버킷에서 S3 Object Lock이 활성화된 경우 버킷 복제(Put Bucket 복제)를 구성하려고 하면 AccessDenied 오류가 발생하고 실패합니다.

플랫폼 서비스 끝점을 구성합니다

버킷에 대한 플랫폼 서비스를 구성하려면 먼저 플랫폼 서비스의 대상으로 하나 이상의 엔드포인트를 구성해야 합니다.

플랫폼 서비스에 대한 액세스는 StorageGRID 관리자가 테넌트 단위로 사용하도록 설정합니다. 플랫폼 서비스 끝점을 만들거나 사용하려면 스토리지 노드가 외부 끝점 리소스에 액세스할 수 있도록 네트워킹이 구성된 그리드에서 끝점 관리

또는 루트 액세스 권한이 있는 테넌트 사용자여야 합니다. 자세한 내용은 StorageGRID 관리자에게 문의하십시오.

플랫폼 서비스 엔드포인트란 무엇입니까?

플랫폼 서비스 끝점을 만들 때 StorageGRID가 외부 대상에 액세스하는 데 필요한 정보를 지정합니다.

예를 들어, StorageGRID 버킷에서 AWS S3 버킷으로 오브젝트를 복제하려는 경우 StorageGRID에서 AWS의 대상 버킷에 액세스하는 데 필요한 정보와 자격 증명이 포함된 플랫폼 서비스 엔드포인트를 생성할 수 있습니다.

각 플랫폼 서비스 유형에는 고유한 엔드포인트가 필요하므로 사용하려는 각 플랫폼 서비스에 대해 하나 이상의 엔드포인트를 구성해야 합니다. 플랫폼 서비스 끝점을 정의한 후 서비스를 활성화하는 데 사용되는 구성 XML에서 끝점의 URN을 대상으로 사용합니다.

둘 이상의 소스 버킷에 대해 목적지와 동일한 끝점을 사용할 수 있습니다. 예를 들어, 여러 버킷에서 검색을 수행할 수 있도록 여러 소스 버킷을 구성하여 동일한 검색 통합 엔드포인트로 오브젝트 메타데이터를 보낼 수 있습니다. 또한 소스 버킷을 구성하여 둘 이상의 엔드포인트를 대상으로 사용할 수 있습니다. 이를 통해 하나의 SNS 항목에 개체 생성 알림을 보내고 개체 삭제에 대한 알림을 두 번째 SNS 항목으로 보내는 등의 작업을 수행할 수 있습니다.

CloudMirror 복제용 엔드포인트

StorageGRID는 S3 버킷을 나타내는 복제 엔드포인트를 지원합니다. 이러한 버킷은 Amazon Web Services, 동일한 또는 원격 StorageGRID 구축 또는 다른 서비스에서 호스팅될 수 있습니다.

알림의 끝점입니다

StorageGRID는 SNS(Simple Notification Service) 엔드포인트를 지원합니다. SQS(Simple Queue Service) 또는 AWS Lambda 엔드포인트는 지원되지 않습니다.

검색 통합 서비스의 끝점입니다

StorageGRID는 Elasticsearch 클러스터를 나타내는 검색 통합 끝점을 지원합니다. 이러한 Elasticsearch 클러스터는 로컬 데이터 센터에 있거나 AWS 클라우드 또는 다른 곳에서 호스팅될 수 있습니다.

검색 통합 끝점은 특정 Elasticsearch 인덱스 및 유형을 참조합니다. StorageGRID에서 끝점을 만들기 전에 Elasticsearch에서 인덱스를 만들어야 합니다. 그렇지 않으면 끝점 생성이 실패합니다. 끝점을 만들기 전에 형식을 만들 필요가 없습니다. StorageGRID는 개체 메타데이터를 끝점으로 보낼 때 필요한 경우 형식을 만듭니다.

관련 정보

StorageGRID 관리

플랫폼 서비스 끝점에 URN을 지정합니다

플랫폼 서비스 끝점을 만들 때는 고유한 URN(리소스 이름)을 지정해야 합니다. 플랫폼 서비스에 대한 구성 XML을 만들 때 URN을 사용하여 끝점을 참조합니다. 각 끝점의 URN은 고유해야 합니다.

StorageGRID에서는 플랫폼 서비스 엔드포인트를 만들 때 이를 검증합니다. 플랫폼 서비스 끝점을 만들기 전에 끝점에 지정된 리소스가 있고 해당 리소스에 도달할 수 있는지 확인합니다.

urn 요소

플랫폼 서비스 끝점의 URN은 다음과 같이 "arn:AWS" 또는 "urn:mysite"로 시작해야 합니다.

- AWS(Amazon Web Services)에서 호스팅되는 서비스의 경우 'arn:AWS'를 사용합니다.
- 서비스가 GCP(Google Cloud Platform)에서 호스팅되는 경우 "arn:AWS"를 사용하십시오.
- 서비스가 로컬로 호스트되는 경우 urn:mysite를 사용합니다

예를 들어, StorageGRID에서 호스팅되는 CloudMirror 엔드포인트에 대해 URN을 지정하는 경우 URN은 "urn:SGWs"로 시작할 수 있습니다.

URN의 다음 요소는 다음과 같이 플랫폼 서비스의 유형을 지정합니다.

서비스	유형
CloudMirror 복제	S3
알림	SNS
검색 통합	ES

예를 들어, StorageGRID에서 호스팅되는 CloudMirror 엔드포인트에 대해 URN을 계속 지정하려면 's3'을 추가하여 urn:SGWs:s3'을 가져옵니다.

URN의 마지막 요소는 대상 URI에서 특정 대상 리소스를 식별합니다.

서비스	특정 리소스
CloudMirror 복제	버킷 이름
알림	SNS-주제-이름
검색 통합	domain-name/index-name/type-name'입니다 • 참고: * Elasticsearch 클러스터가 자동으로 인덱스를 만들도록 * 구성되지 * 인 경우 끝점을 만들기 전에 수동으로 인덱스를 만들어야 합니다.

AWS 및 **GCP**에서 호스팅되는 서비스의 여관

AWS 및 GCP 엔터티의 경우 URN은 유효한 AWS ARN입니다. 예를 들면 다음과 같습니다.

- CloudMirror 복제:

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

- 알림:

```
arn:aws:sns:region:account-id:topic-name
```

- 검색 통합:

```
arn:aws:es:region:account-id:domain/domain-name/index-name/type-name
```

> AWS 검색 통합 엔드포인트의 경우 domain-name에는 여기에 나와 있는 리터럴 문자열 domain/"이 포함되어야 합니다.

현지 호스팅 서비스를 위한 여관

클라우드 서비스 대신 로컬로 호스팅된 서비스를 사용하는 경우 URN에 필요한 요소가 세 번째 및 최종 위치에 포함되어 있는 한 유효하고 고유한 URN을 만드는 방식으로 URN을 지정할 수 있습니다. 선택 사항으로 표시된 요소를 비워 두거나 자원을 식별하고 URN을 고유하게 만드는 데 도움이 되도록 원하는 방식으로 지정할 수 있습니다. 예를 들면 다음과 같습니다.

- CloudMirror 복제:

```
urn:mysite:s3:optional:optional:bucket-name
```

  StorageGRID에서 호스팅되는 CloudMirror 엔드포인트의 경우 "urn:SGW"로 시작하는 유효한 URN을 지정할 수 있습니다.

```
urn:sgws:s3:optional:optional:bucket-name
```

- 알림:

```
urn:mysite:sns:optional:optional:sns-topic-name
```

- 검색 통합:

```
urn:mysite:es:optional:optional:domain-name/index-name/type-name
```

> 로컬에서 호스팅되는 검색 통합 끝점의 경우 끝점의 URN이 고유하면 domain-name 요소는 모든 문자열이 될 수 있습니다.

플랫폼 서비스 끝점을 만듭니다

플랫폼 서비스를 사용하려면 먼저 올바른 유형의 끝점을 하나 이상 만들어야 합니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- StorageGRID 관리자가 테넌트 계정에 대해 플랫폼 서비스를 사용하도록 설정해야 합니다.

- 끝점 관리 권한이 있는 사용자 그룹에 속해야 합니다.
- 플랫폼 서비스 끝점에서 참조하는 리소스가 생성되어야 합니다.
  - CloudMirror 복제: S3 버킷
  - 이벤트 알림: SNS 항목
  - 검색 알림: 대상 클러스터가 인덱스를 자동으로 생성하도록 구성되지 않은 경우 Elasticsearch index입니다.
- 대상 리소스에 대한 정보가 있어야 합니다.
  - URI(Uniform Resource Identifier)의 호스트 및 포트

    > StorageGRID 시스템에서 호스팅되는 버킷을 CloudMirror 복제의 엔드포인트로 사용하려면 그리드 관리자에게 문의하여 입력해야 하는 값을 확인하십시오.

  - 고유 리소스 이름(URN)

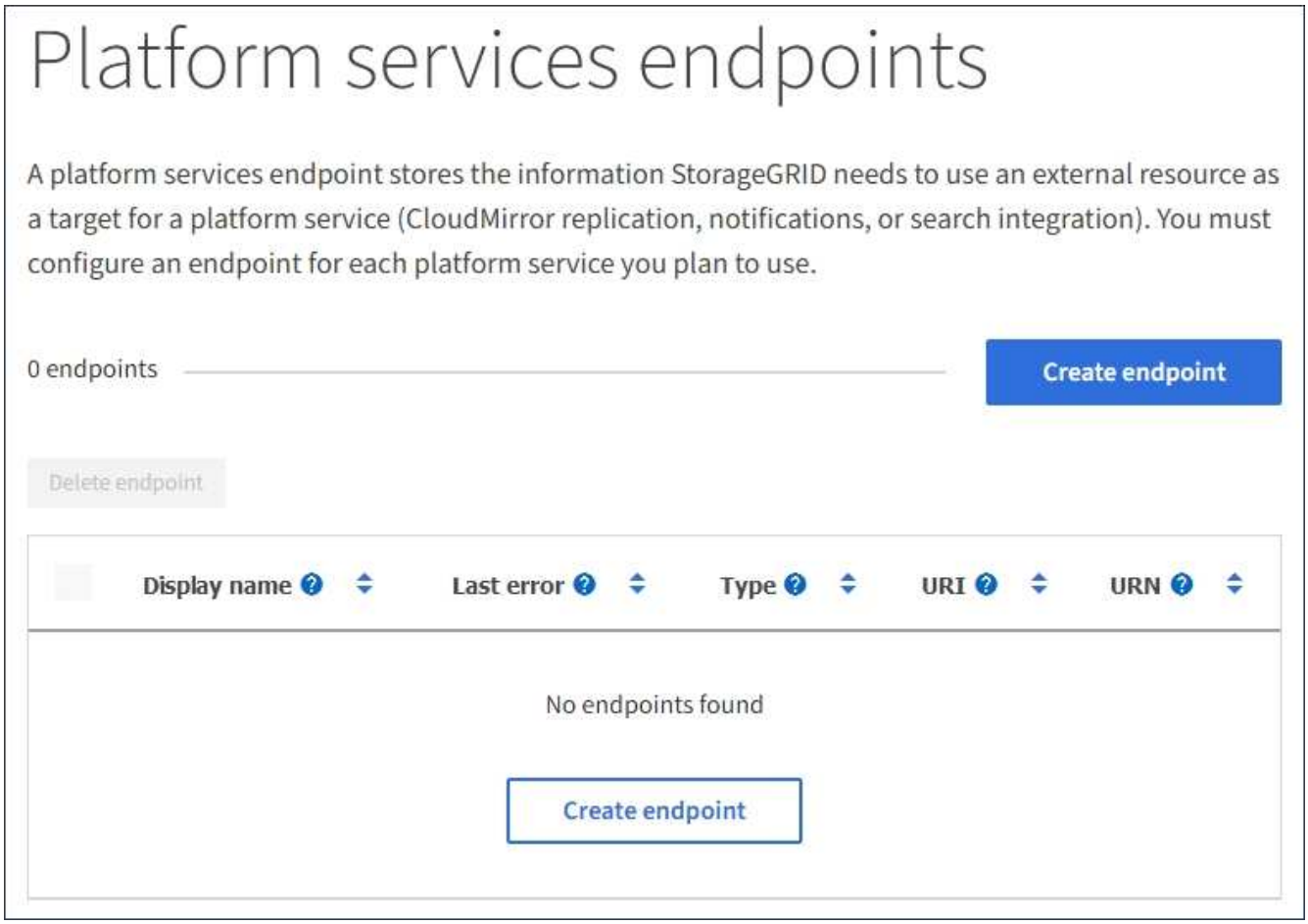
    플랫폼 서비스 끝점에 URN을 지정합니다

  - 인증 자격 증명(필요한 경우):
    - 액세스 키: 액세스 키 ID 및 비밀 액세스 키
    - 기본 HTTP: 사용자 이름 및 암호
    - CAP(C2S Access Portal): 임시 자격 증명 URL, 서버 및 클라이언트 인증서, 클라이언트 키 및 선택적 클라이언트 개인 키 암호.
  - 보안 인증서(사용자 지정 CA 인증서를 사용하는 경우)

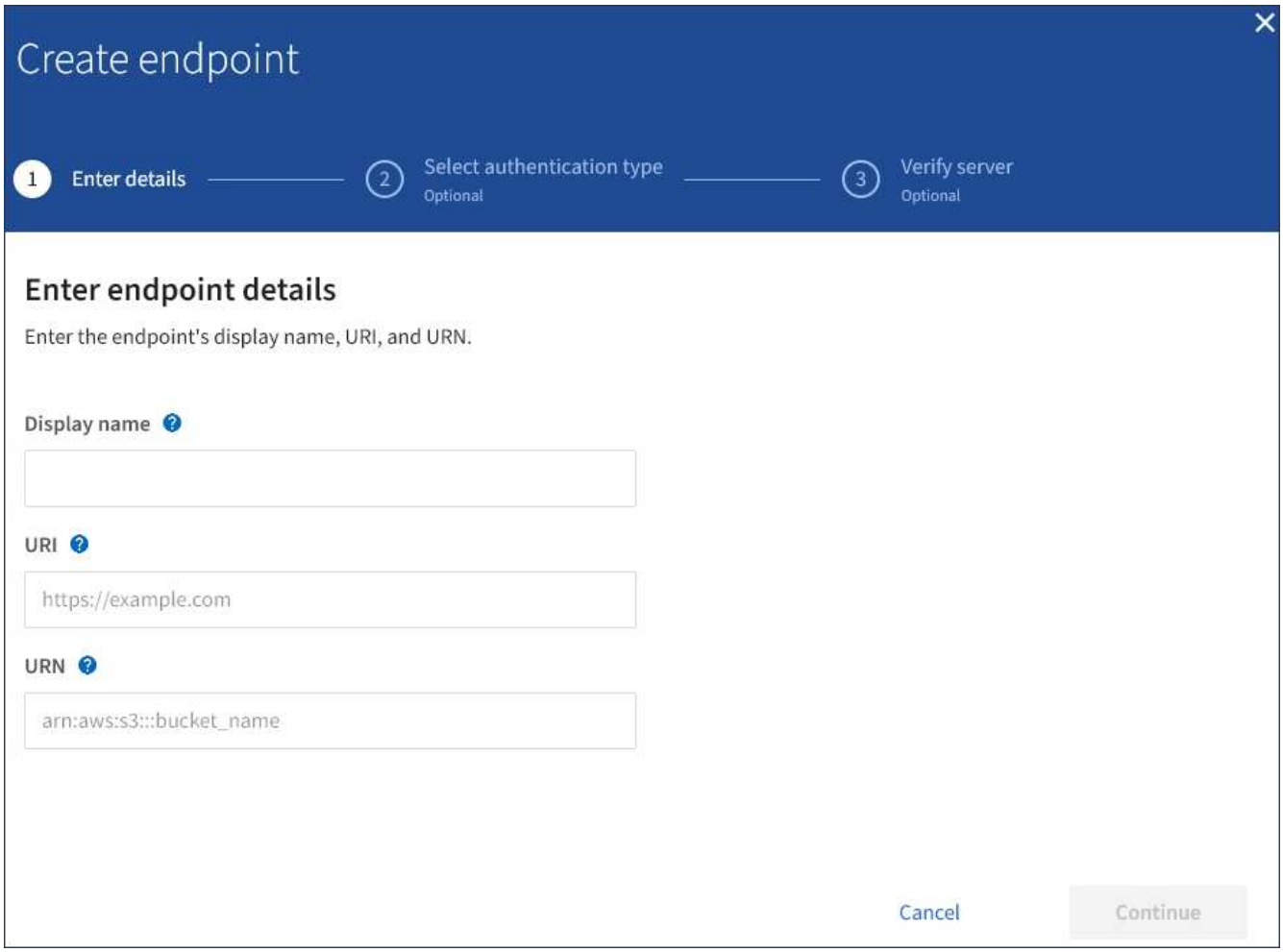
단계

1. 스토리지(S3) * > * 플랫폼 서비스 엔드포인트 * 를 선택합니다.

   플랫폼 서비스 끝점 페이지가 나타납니다.



2. 끝점 만들기 * 를 선택합니다.



3. 표시 이름을 입력하여 끝점과 그 용도를 간략하게 설명합니다.

   끝점이 지원하는 플랫폼 서비스 유형은 끝점 페이지에 나열될 때 끝점 이름 옆에 표시됩니다. 따라서 이름에 해당 정보를 포함할 필요가 없습니다.

4. URI * 필드에서 끝점의 고유 URI(Resource Identifier)를 지정합니다.

   다음 형식 중 하나를 사용합니다.

```
https://host:port
http://host:port
```

   포트를 지정하지 않으면 포트 443이 HTTPS URI에 사용되고 포트 80은 HTTP URI에 사용됩니다.

   예를 들어 StorageGRID에서 호스팅되는 버킷의 URI는 다음과 같습니다.

```
https://s3.example.com:10443
```

   이 예에서 '3.example.com'는 StorageGRID HA(High Availability) 그룹의 가상 IP(VIP)에 대한 DNS 항목을 나타내고, '10443'은 로드 밸런서 끝점에 정의된 포트를 나타냅니다.

가능하면 단일 장애 지점을 피하기 위해 로드 밸런싱 노드의 HA 그룹에 연결해야 합니다.

마찬가지로 AWS에서 호스팅되는 버킷의 URI는 다음과 같습니다.

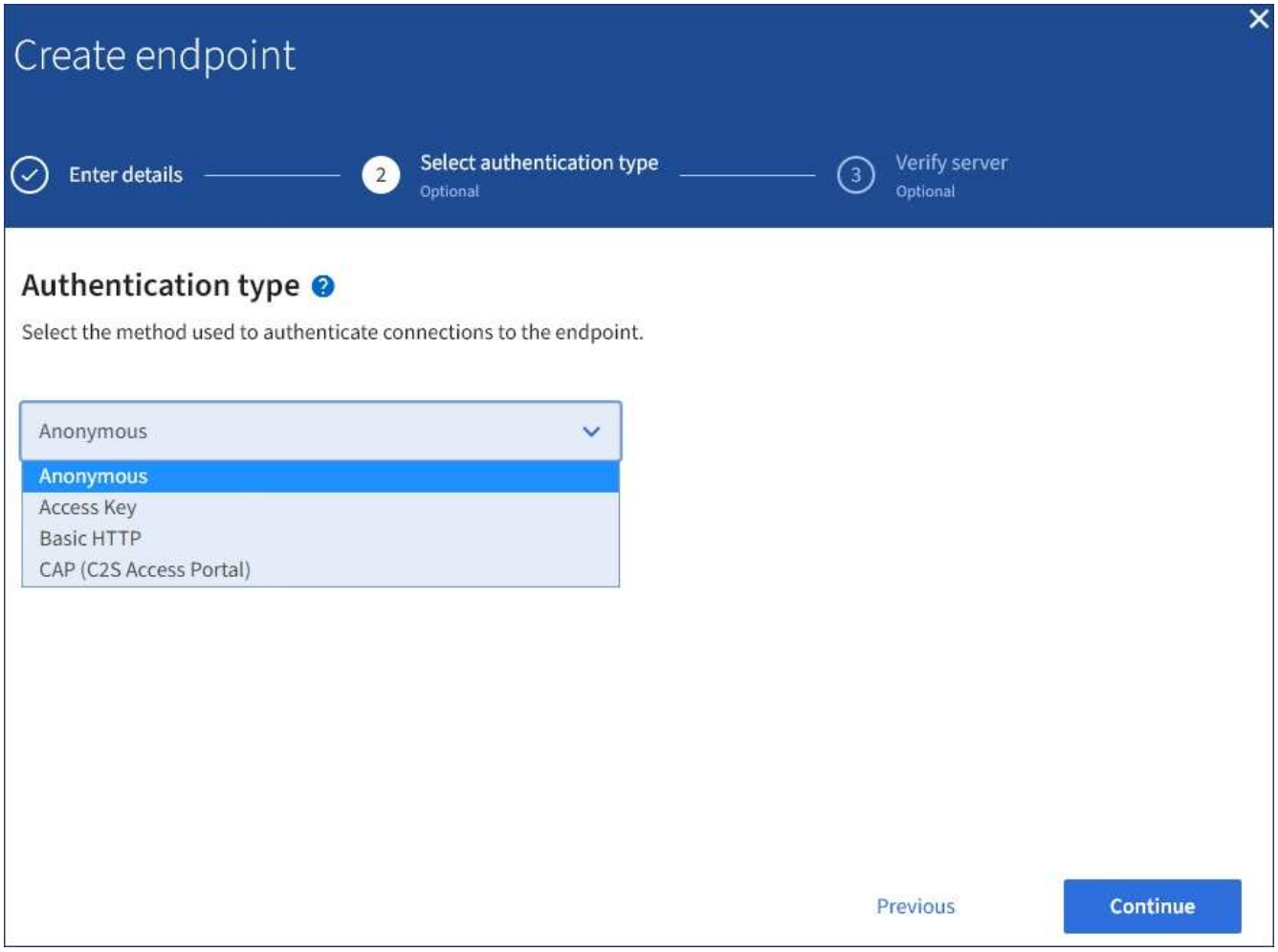
```
https://s3-aws-region.amazonaws.com
```

엔드포인트가 CloudMirror 복제 서비스에 사용되는 경우 버킷 이름을 URI에 포함하지 마십시오. 버킷 이름을 * URN * 필드에 포함시킵니다.

5. 끝점에 대한 고유 URN(리소스 이름)을 입력합니다.

   끝점이 생성된 후에는 끝점의 URN을 변경할 수 없습니다.

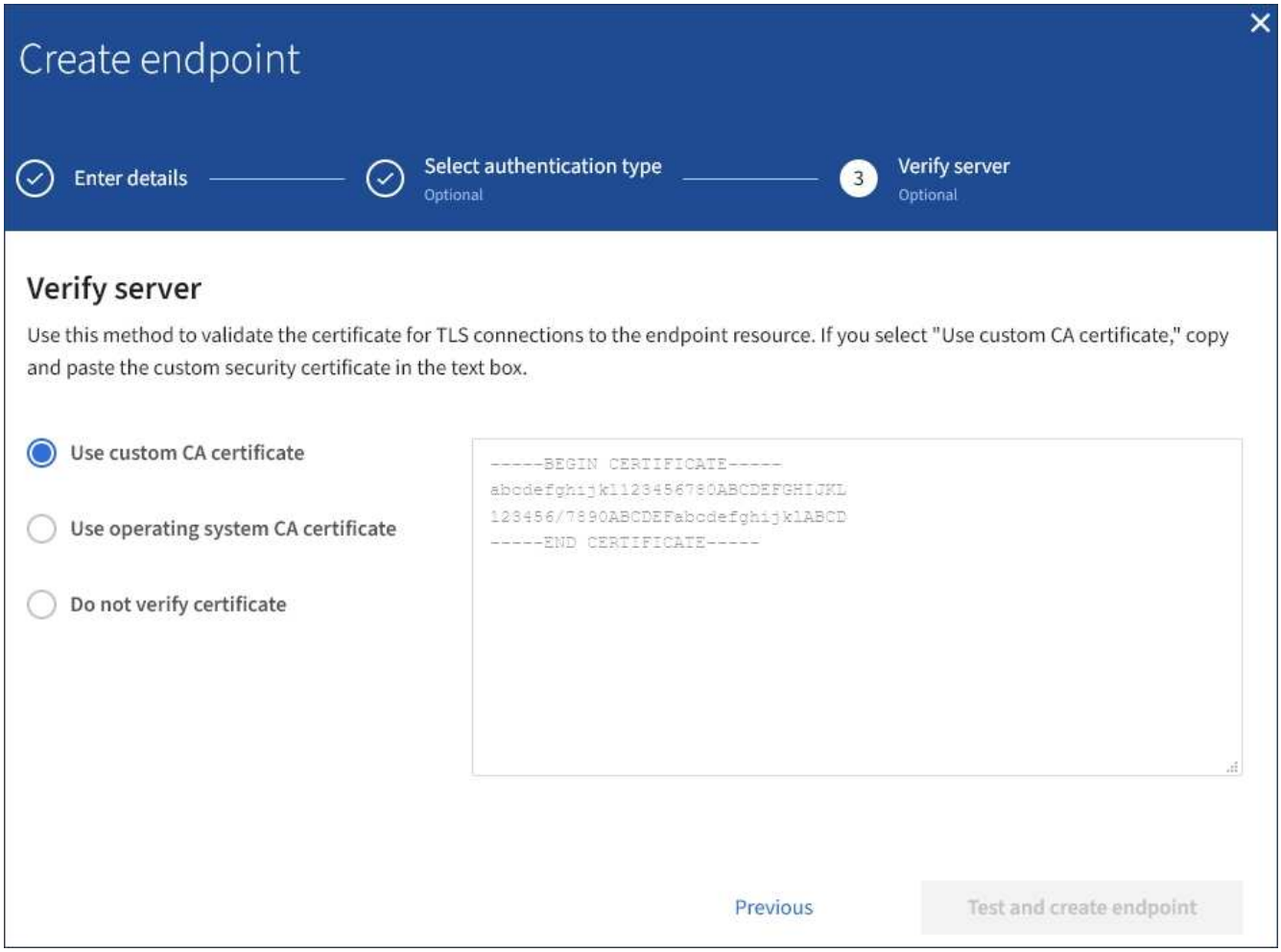
6. Continue * 를 선택합니다.
7. 인증 유형 * 의 값을 선택한 다음 필요한 자격 증명을 입력하거나 업로드합니다.



제공하는 자격 증명에 대상 리소스에 대한 쓰기 권한이 있어야 합니다.

인증 유형입니다	설명	자격 증명
익명	대상에 대한 익명 액세스를 제공합니다. 보안이 비활성화된 끝점에서만 작동합니다.	인증이 없습니다.
액세스 키	AWS 스타일 자격 증명을 사용하여 대상과의 연결을 인증합니다.	• 액세스 키 ID입니다 • 비밀 액세스 키
기본 HTTP	사용자 이름과 암호를 사용하여 대상에 대한 연결을 인증합니다.	• 사용자 이름 • 암호
CAP(C2S 액세스 포털)	인증서 및 키를 사용하여 대상에 대한 연결을 인증합니다.	• 임시 자격 증명 URL입니다 • 서버 CA 인증서(PEM 파일 업로드) • 클라이언트 인증서(PEM 파일 업로드) • 클라이언트 개인 키(PEM 파일 업로드, OpenSSL 암호화 형식 또는 암호화되지 않은 개인 키 형식) • 클라이언트 개인 키 암호 구문(선택 사항)

8. Continue * 를 선택합니다.
9. 끝점에 대한 TLS 연결을 확인하는 방법을 선택하려면 * 서버 확인 * 에 대한 라디오 버튼을 선택합니다.



인증서 확인 유형입니다	설명
사용자 지정 CA 인증서를 사용합니다	사용자 지정 보안 인증서를 사용합니다. 이 설정을 선택한 경우 사용자 지정 보안 인증서를 복사하여 * CA 인증서 * 텍스트 상자에 붙여 넣습니다.
운영 체제 CA 인증서를 사용합니다	운영 체제에 설치된 기본 그리드 CA 인증서를 사용하여 연결을 보호합니다.
인증서를 확인하지 않습니다	TLS 연결에 사용되는 인증서가 검증되지 않았습니다. 이 옵션은 안전하지 않습니다.

10. 테스트를 선택하고 끝점 * 을 작성합니다.
    - 지정된 자격 증명을 사용하여 끝점에 도달할 수 있으면 성공 메시지가 나타납니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.
    - 끝점 유효성 검사에 실패하면 오류 메시지가 나타납니다. 오류를 수정하기 위해 끝점을 수정해야 하는 경우 * 끝점 세부 정보로 돌아가기 * 를 선택하고 정보를 업데이트합니다. 그런 다음 * 테스트 를 선택하고 끝점 * 을 만듭니다.

> 테넌트 계정에 플랫폼 서비스가 활성화되어 있지 않으면 엔드포인트 생성이 실패합니다. StorageGRID 관리자에게 문의하십시오.

끝점을 구성한 후 URN을 사용하여 플랫폼 서비스를 구성할 수 있습니다.

관련 정보

플랫폼 서비스 끝점에 URN을 지정합니다

CloudMirror 복제를 구성합니다

이벤트 알림을 구성합니다

검색 통합 서비스를 구성합니다

플랫폼 서비스 끝점에 대한 연결을 테스트합니다

플랫폼 서비스에 대한 연결이 변경된 경우 끝점에 대한 연결을 테스트하여 대상 리소스가 있는지 그리고 지정한 자격 증명을 사용하여 해당 리소스에 연결할 수 있는지 확인할 수 있습니다.

필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 끝점 관리 권한이 있는 사용자 그룹에 속해야 합니다.

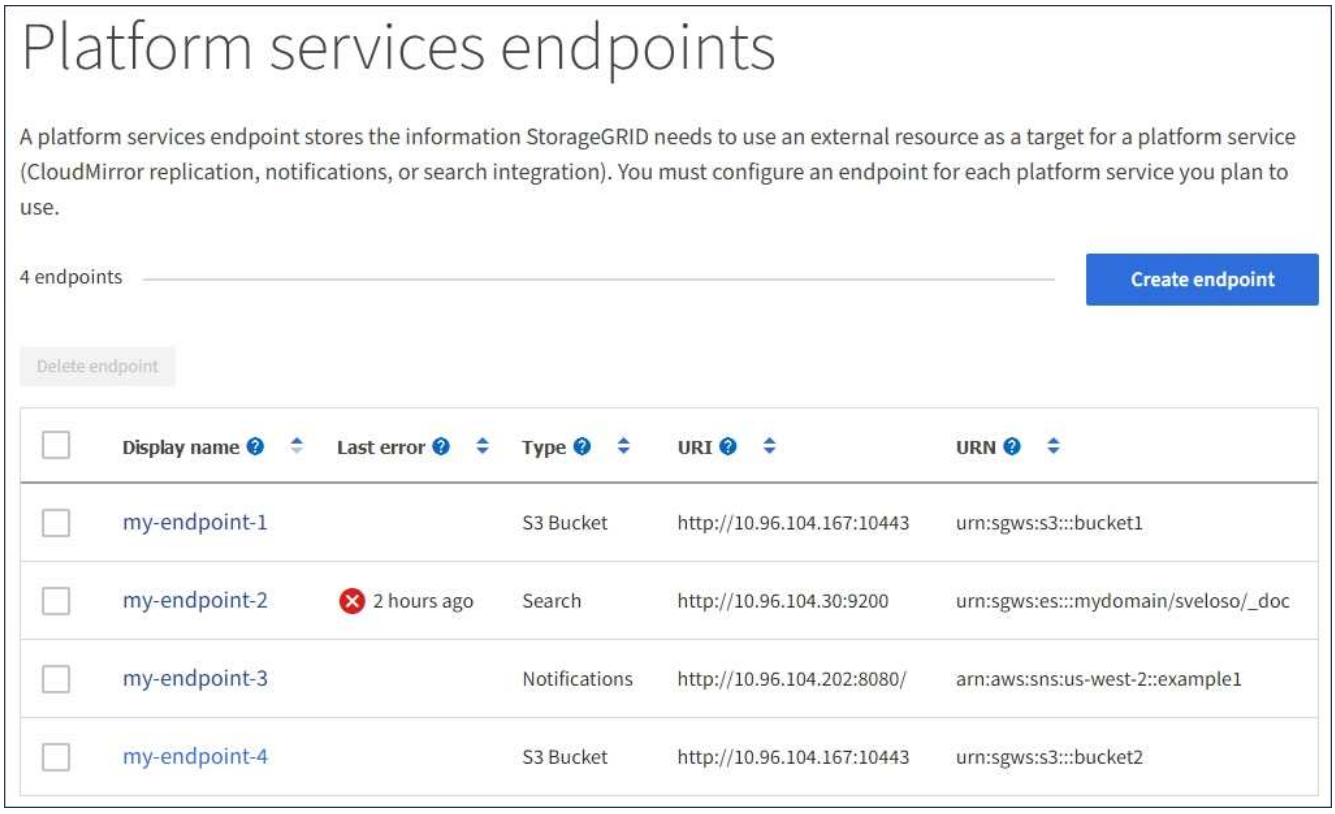
이 작업에 대해

StorageGRID는 자격 증명에 올바른 권한이 있는지 확인하지 않습니다.

단계

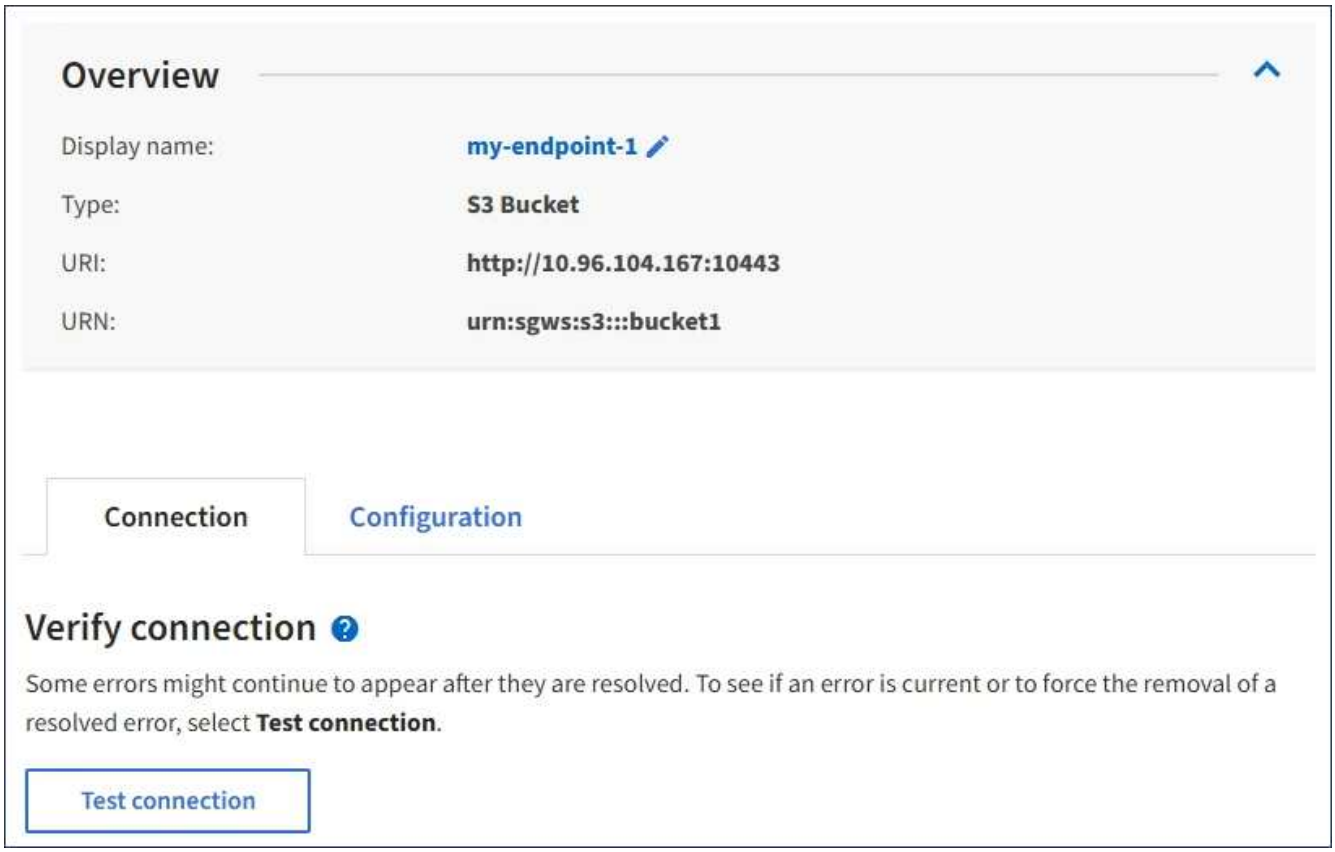
1. 스토리지(S3) * > * 플랫폼 서비스 엔드포인트 * 를 선택합니다.

   플랫폼 서비스 끝점 페이지가 나타나고 이미 구성된 플랫폼 서비스 끝점 목록이 표시됩니다.



2. 연결을 테스트할 끝점을 선택합니다.

   끝점 세부 정보 페이지가 나타납니다.



3. Test connection * 을 선택합니다.
   - 지정된 자격 증명을 사용하여 끝점에 도달할 수 있으면 성공 메시지가 나타납니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.
   - 끝점 유효성 검사에 실패하면 오류 메시지가 나타납니다. 오류를 정정하기 위해 끝점을 수정해야 하는 경우 * 구성 * 을 선택하고 정보를 업데이트합니다. 그런 다음 * 테스트 및 변경 내용 저장 * 을 선택합니다.

플랫폼 서비스 끝점을 편집합니다

플랫폼 서비스 끝점의 구성을 편집하여 이름, URI 또는 기타 세부 정보를 변경할 수 있습니다. 예를 들어 만료된 자격 증명을 업데이트하거나 대체 작동을 위한 백업 Elasticsearch 인덱스를 가리키도록 URI를 변경해야 할 수 있습니다. 플랫폼 서비스 끝점의 URN은 변경할 수 없습니다.

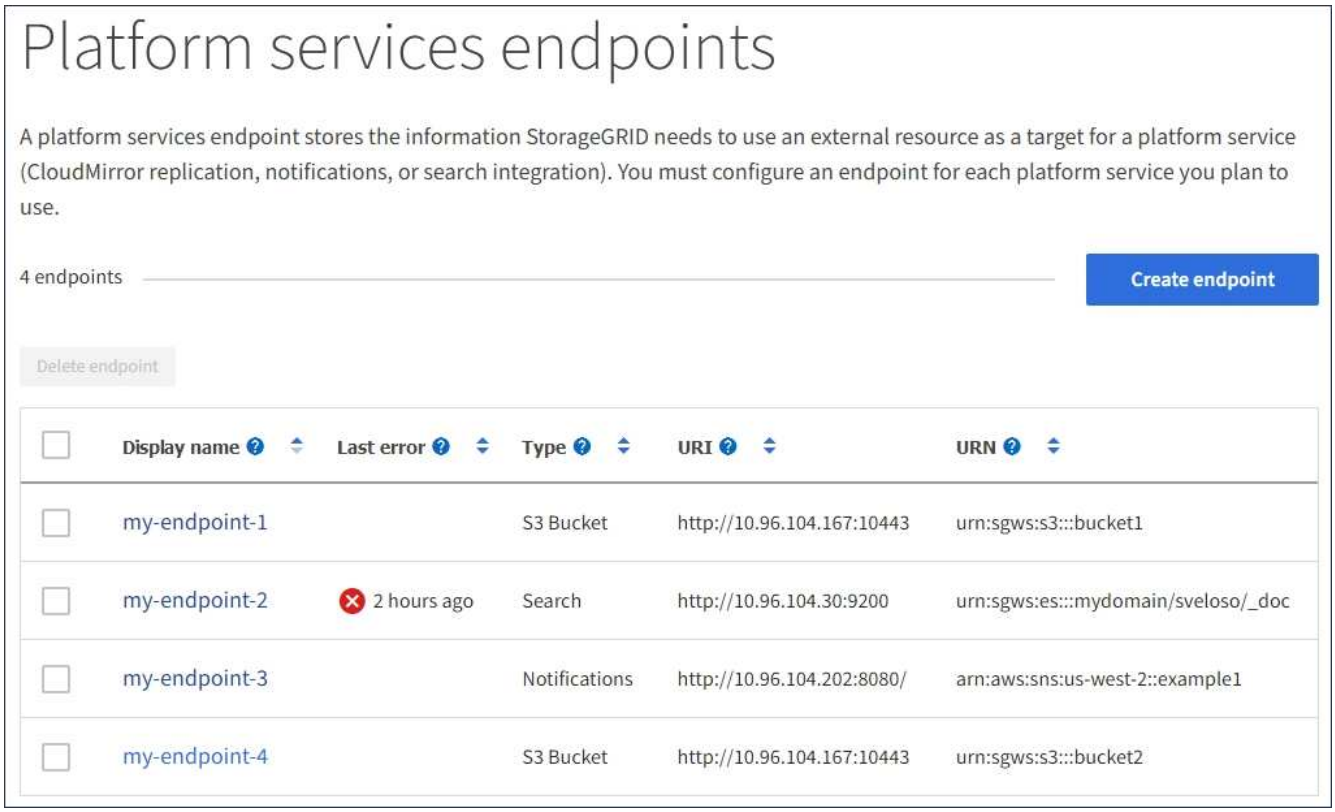
필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 끝점 관리 권한이 있는 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.

단계

1. 스토리지(S3) * > * 플랫폼 서비스 엔드포인트 * 를 선택합니다.

   플랫폼 서비스 끝점 페이지가 나타나고 이미 구성된 플랫폼 서비스 끝점 목록이 표시됩니다.



2. 편집할 끝점을 선택합니다.

   끝점 세부 정보 페이지가 나타납니다.

3. Configuration * 을 선택합니다.

Overview

Display name:	my-endpoint-3
Type:	Notifications
URI:	http://10.96.104.202:8080/
URN:	arn:aws:sns:us-west-2::example1

Connection | Configuration

Edit configuration

Endpoint details

URI

http://10.96.104.202:8080/

URN

arn:aws:sns:us-west-2::example1

Authentication type

Basic HTTP

Username

testme

Password

••••••••

Edit password

Verify server

- Use custom CA certificate
- Use operating system CA certificate
- Do not verify certificate

```
-----BEGIN CERTIFICATE-----
abcdefghijkl123456780ABCDEFGHIJKL
123456/7890ABCDEFabcdefghijklABCD
-----END CERTIFICATE-----
```

Test and save changes

4. 필요에 따라 끝점의 구성을 변경합니다.

> 끝점이 생성된 후에는 끝점의 URN을 변경할 수 없습니다.

   a. 끝점의 표시 이름을 변경하려면 편집 아이콘을 선택합니다 ✎.
   b. 필요에 따라 URI를 변경합니다.
   c. 필요에 따라 인증 유형을 변경합니다.
      - 액세스 키 인증의 경우 * S3 키 편집 * 을 선택하고 새 액세스 키 ID 및 비밀 액세스 키를 붙여 넣어 필요에 따라 키를 변경합니다. 변경 사항을 취소하려면 * S3 키 편집 되돌리기 * 를 선택합니다.
      - 기본 HTTP 인증의 경우 필요에 따라 사용자 이름을 변경합니다. 필요에 따라 * 암호 편집 * 을 선택하고 새 암호를 입력하여 암호를 변경합니다. 변경 사항을 취소해야 하는 경우 * 암호 편집 되돌리기 * 를 선택합니다.
      - CAP(C2S Access Portal) 인증의 경우 임시 자격 증명 URL 또는 선택적 클라이언트 개인 키 암호를 변경하고 필요에 따라 새 인증서 및 키 파일을 업로드합니다.

        > 클라이언트 개인 키는 OpenSSL 암호화 형식 또는 암호화되지 않은 개인 키 형식이어야 합니다.

   d. 필요에 따라 서버 확인 방법을 변경합니다.
5. Test(테스트)를 선택하고 변경 내용을 저장합니다 *.
   - 지정된 자격 증명을 사용하여 끝점에 도달할 수 있으면 성공 메시지가 나타납니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 확인합니다.
   - 끝점 유효성 검사에 실패하면 오류 메시지가 나타납니다. 끝점을 수정하여 오류를 수정한 다음 * 테스트 및 변경 내용 저장 * 을 선택합니다.

플랫폼 서비스 끝점을 삭제합니다

연결된 플랫폼 서비스를 더 이상 사용하지 않으려면 끝점을 삭제할 수 있습니다.

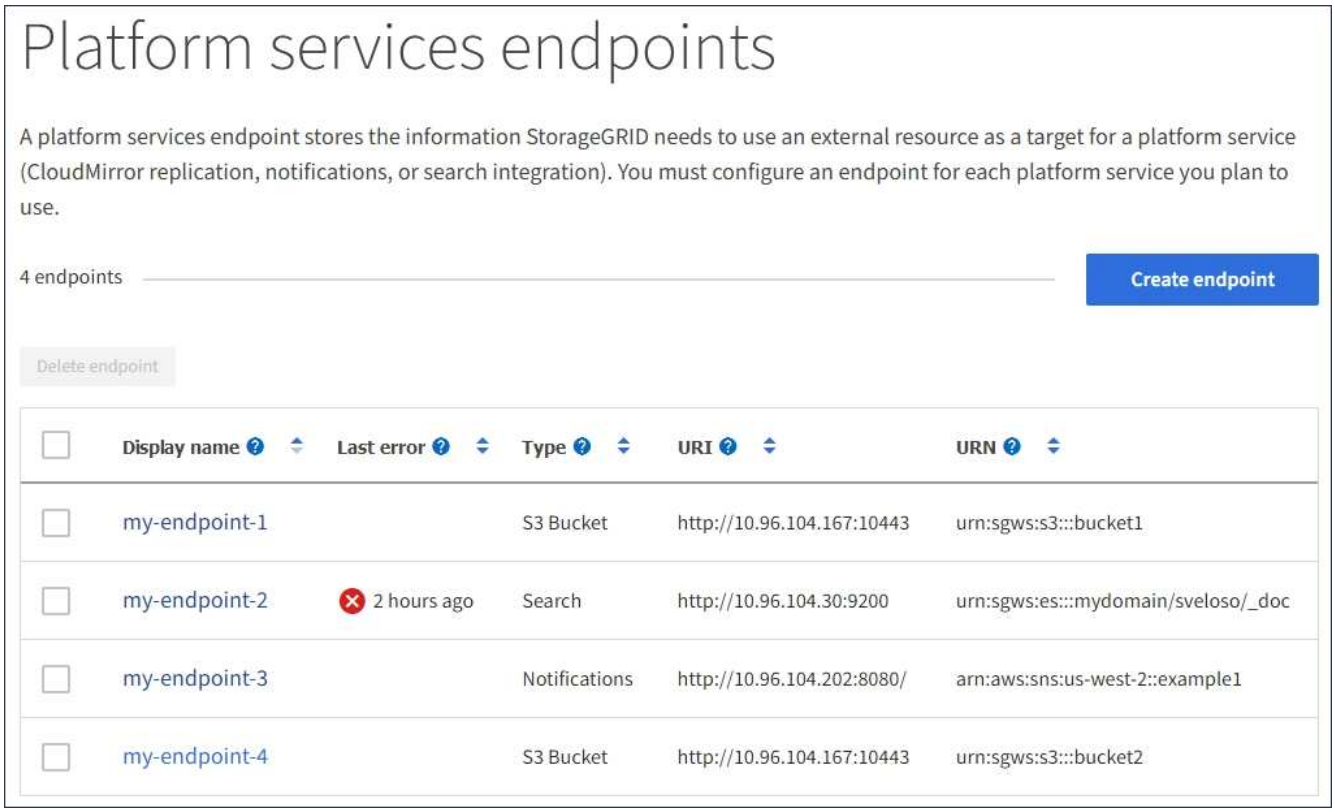
필요한 것

- 를 사용하여 테넌트 관리자에 로그인해야 합니다 지원되는 웹 브라우저.
- 끝점 관리 * 권한이 있는 사용자 그룹에 속해야 합니다. 을 참조하십시오 테넌트 관리 권한.

단계

1. 스토리지(S3) * > * 플랫폼 서비스 엔드포인트 * 를 선택합니다.

   플랫폼 서비스 끝점 페이지가 나타나고 이미 구성된 플랫폼 서비스 끝점 목록이 표시됩니다.

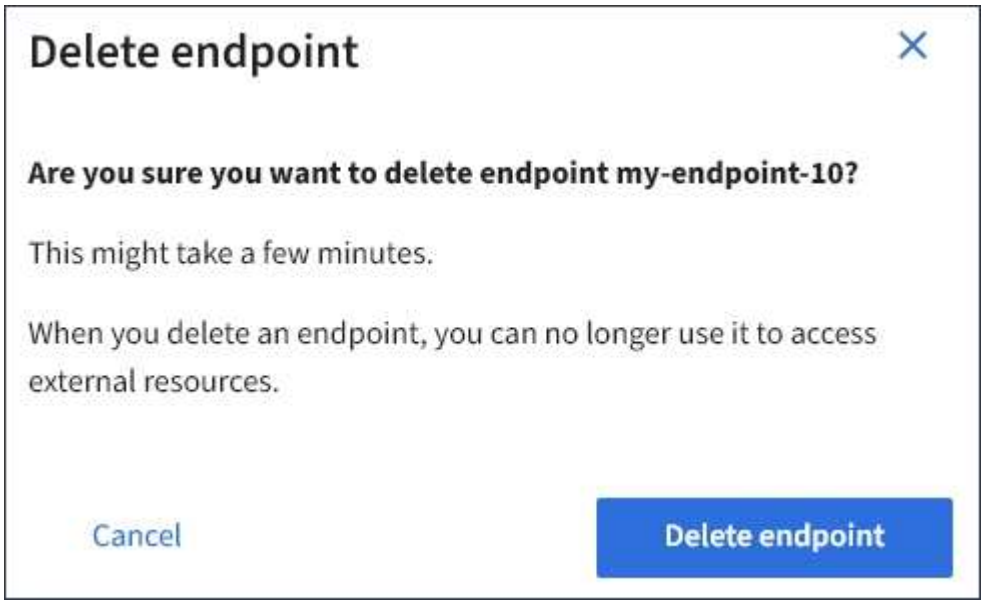


2. 삭제할 각 끝점의 확인란을 선택합니다.

> 사용 중인 플랫폼 서비스 끝점을 삭제하면 해당 끝점을 사용하는 모든 버킷에 대해 연결된 플랫폼 서비스가 비활성화됩니다. 아직 완료되지 않은 요청은 삭제됩니다. 삭제된 URN을 더 이상 참조하지 않도록 버킷 구성을 변경할 때까지 새 요청은 계속 생성됩니다. StorageGRID는 이러한 요청을 복구할 수 없는 오류로 보고합니다.

3. 작업 * > * 끝점 삭제 * 를 선택합니다.

   확인 메시지가 나타납니다.



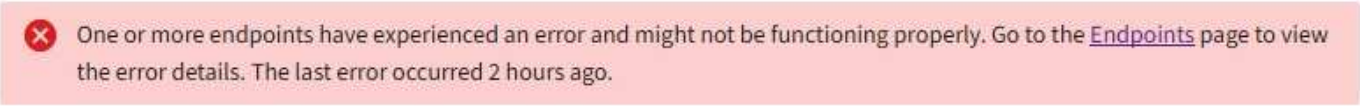
4. 끝점 삭제 * 를 선택합니다.

플랫폼 서비스 끝점 오류 문제 해결

StorageGRID가 플랫폼 서비스 끝점과 통신하려고 할 때 오류가 발생하면 대시보드에 메시지가 표시됩니다. 플랫폼 서비스 끝점 페이지에서 마지막 오류 열은 오류가 발생한 시간을 나타냅니다. 끝점의 자격 증명과 연결된 권한이 올바르지 않으면 오류가 표시되지 않습니다.

오류가 발생했는지 확인합니다

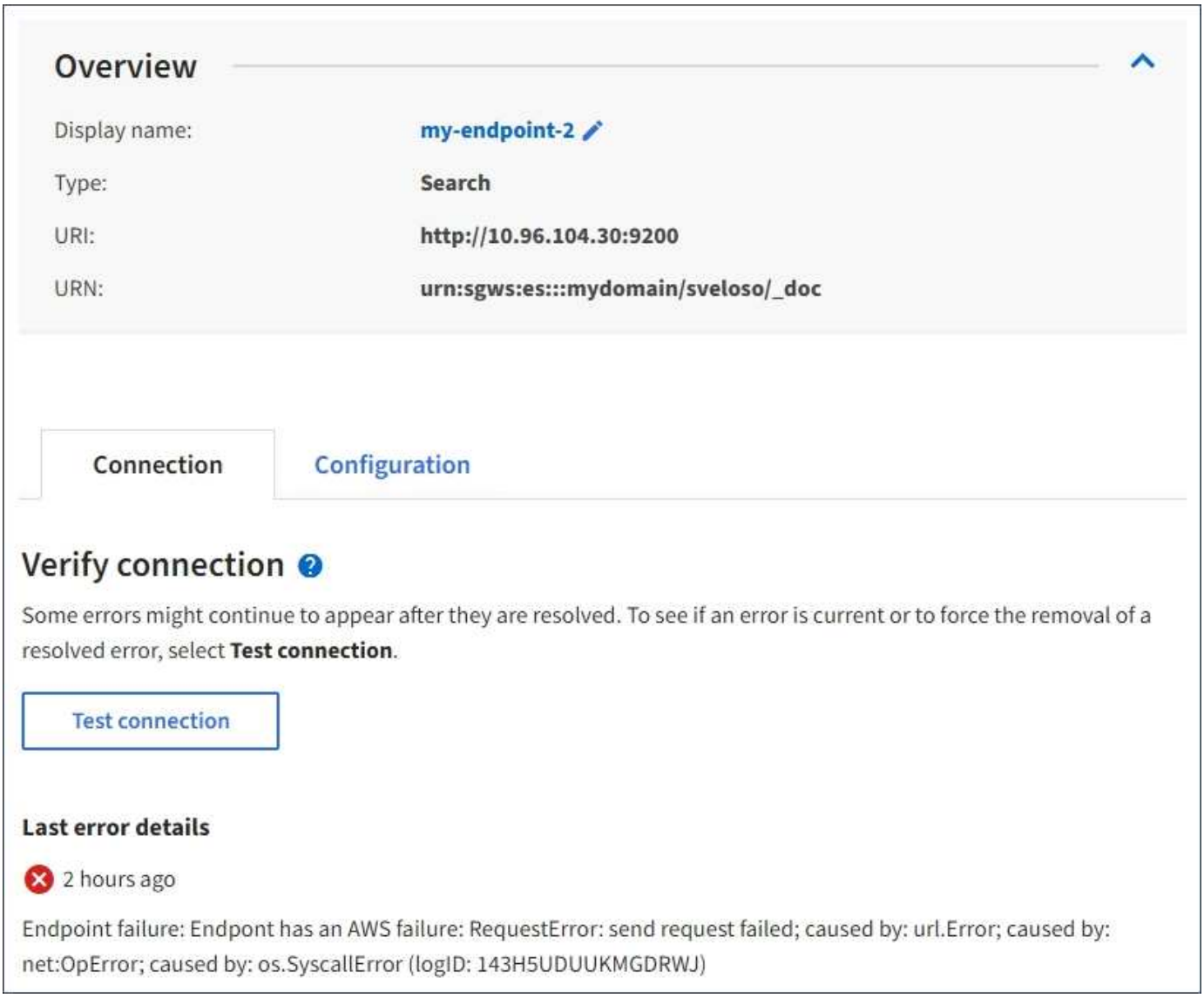
지난 7일 이내에 플랫폼 서비스 끝점 오류가 발생한 경우 테넌트 관리자 대시보드에 경고 메시지가 표시됩니다. 플랫폼 서비스 끝점 페이지로 이동하여 오류에 대한 자세한 정보를 볼 수 있습니다.



대시보드에 나타나는 동일한 오류가 플랫폼 서비스 끝점 페이지 맨 위에도 나타납니다. 자세한 오류 메시지를 보려면:

단계

1. 끝점 목록에서 오류가 있는 끝점을 선택합니다.
2. 끝점 세부 정보 페이지에서 * 연결 * 을 선택합니다. 이 탭은 끝점에 대한 가장 최근 오류만 표시하고 오류가 발생한 시간을 표시합니다. 빨간색 X 아이콘이 포함된 오류 지난 7일 이내에 발생했습니다.



오류가 여전히 최신 상태인지 확인합니다

일부 오류는 해결된 후에도 * 마지막 오류 * 열에 계속 표시될 수 있습니다. 오류가 현재 오류인지 확인하거나 테이블에서 해결된 오류를 강제로 제거하려면 다음과 같이 하십시오.

단계

1. 끝점을 선택합니다.

   끝점 세부 정보 페이지가 나타납니다.

2. 연결 * > * 연결 테스트 * 를 선택합니다.

연결 테스트 * 를 선택하면 StorageGRID가 플랫폼 서비스 끝점이 있는지, 그리고 현재 자격 증명으로 연결할 수 있는지 검증합니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.

끝점 오류를 해결합니다

끝점 세부 정보 페이지의 * 마지막 오류 * 메시지를 사용하여 오류의 원인을 확인할 수 있습니다. 일부 오류에서는 문제를 해결하기 위해 끝점을 편집해야 할 수 있습니다. 예를 들어, 올바른 액세스 권한이 없거나 액세스 키가 만료되어

StorageGRID가 대상 S3 버킷을 액세스할 수 없는 경우 클라우드미러링 오류가 발생할 수 있습니다. 이 메시지는 ""끝점 자격 증명이나 대상 액세스 업데이트 필요""이며 세부 정보는 ""AccessDenied"" 또는 ""InvalidAccessKeyId""입니다.

오류를 해결하기 위해 끝점을 편집해야 하는 경우 * 테스트 및 변경 내용 저장 * 을 선택하면 StorageGRID가 업데이트된 끝점을 검증하고 현재 자격 증명으로 연결할 수 있는지 확인합니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.

단계

1. 끝점을 선택합니다.
2. 끝점 세부 정보 페이지에서 * 구성 * 을 선택합니다.
3. 필요에 따라 끝점 설정을 편집합니다.
4. 연결 * > * 연결 테스트 * 를 선택합니다.

권한이 부족한 끝점 자격 증명

StorageGRID에서 플랫폼 서비스 끝점의 유효성을 검사할 때 끝점의 자격 증명을 사용하여 대상 리소스에 연결할 수 있는지 확인하고 기본적인 사용 권한 검사를 수행합니다. 그러나 StorageGRID는 특정 플랫폼 서비스 작업에 필요한 모든 사용 권한의 유효성을 검사하지 않습니다. 따라서 플랫폼 서비스("403 사용 금지" 등)를 사용할 때 오류가 발생하면 끝점의 자격 증명과 관련된 권한을 확인하십시오.

추가 플랫폼 서비스 문제 해결

플랫폼 서비스 문제 해결에 대한 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

StorageGRID 관리

관련 정보

플랫폼 서비스 끝점을 만듭니다

플랫폼 서비스 끝점에 대한 연결을 테스트합니다

플랫폼 서비스 끝점을 편집합니다

CloudMirror 복제를 구성합니다

를 클릭합니다 CloudMirror 복제 서비스 는 세 가지 StorageGRID 플랫폼 서비스 중 하나입니다. CloudMirror 복제를 사용하여 오브젝트를 외부 S3 버킷에 자동으로 복제할 수 있습니다.

필요한 것

- StorageGRID 관리자가 테넌트 계정에 대해 플랫폼 서비스를 사용하도록 설정해야 합니다.
- 복제 소스로 사용할 버킷을 이미 생성해야 합니다.
- CloudMirror 복제의 대상으로 사용하려는 엔드포인트가 이미 있어야 하며 URN이 있어야 합니다.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이 권한을 사용하여 테넌트 계정의 모든 S3 버킷에 대한 설정을 관리할 수 있습니다. 이러한 권한은 테넌트 관리자를 사용하여 버킷을 구성할 때 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업에 대해

CloudMirror 복제는 소스 버킷에서 엔드포인트에 지정된 대상 버킷으로 객체를 복제합니다. 버킷에 대해 CloudMirror 복제를 설정하려면 유효한 버킷 복제 구성 XML을 생성하고 적용해야 합니다. 복제 구성 XML은 각 대상에 대해 S3 버킷 엔드포인트의 URN을 사용해야 합니다.

> S3 오브젝트 잠금이 활성화된 소스 또는 대상 버킷에는 복제가 지원되지 않습니다.

버킷 복제 및 구성 방법에 대한 일반적인 정보는 교차 지역 복제(CRR)에 대한 Amazon Simple Storage Service(S3) 설명서를 참조하십시오. StorageGRID에서 S3 버킷 복제 구성 API를 구현하는 방법에 대한 자세한 내용은 를 참조하십시오 S3 클라이언트 애플리케이션 구현 지침.

객체가 포함된 버킷에서 CloudMirror 복제를 활성화하면 버킷에 추가된 새 객체가 복제되지만 버킷의 기존 객체는 복제되지 않습니다. 복제를 트리거하려면 기존 객체를 업데이트해야 합니다.

복제 구성 XML에서 스토리지 클래스를 지정하는 경우 StorageGRID는 대상 S3 끝점에 대해 작업을 수행할 때 해당 클래스를 사용합니다. 대상 끝점은 지정된 저장소 클래스도 지원해야 합니다. 대상 시스템 공급업체에서 제공하는 권장 사항을 따르십시오.

단계

1. 소스 버킷에 대한 복제 지원:

   텍스트 편집기를 사용하여 S3 복제 API에 지정된 대로 복제를 활성화하는 데 필요한 복제 구성 XML을 생성합니다. XML을 구성할 때:

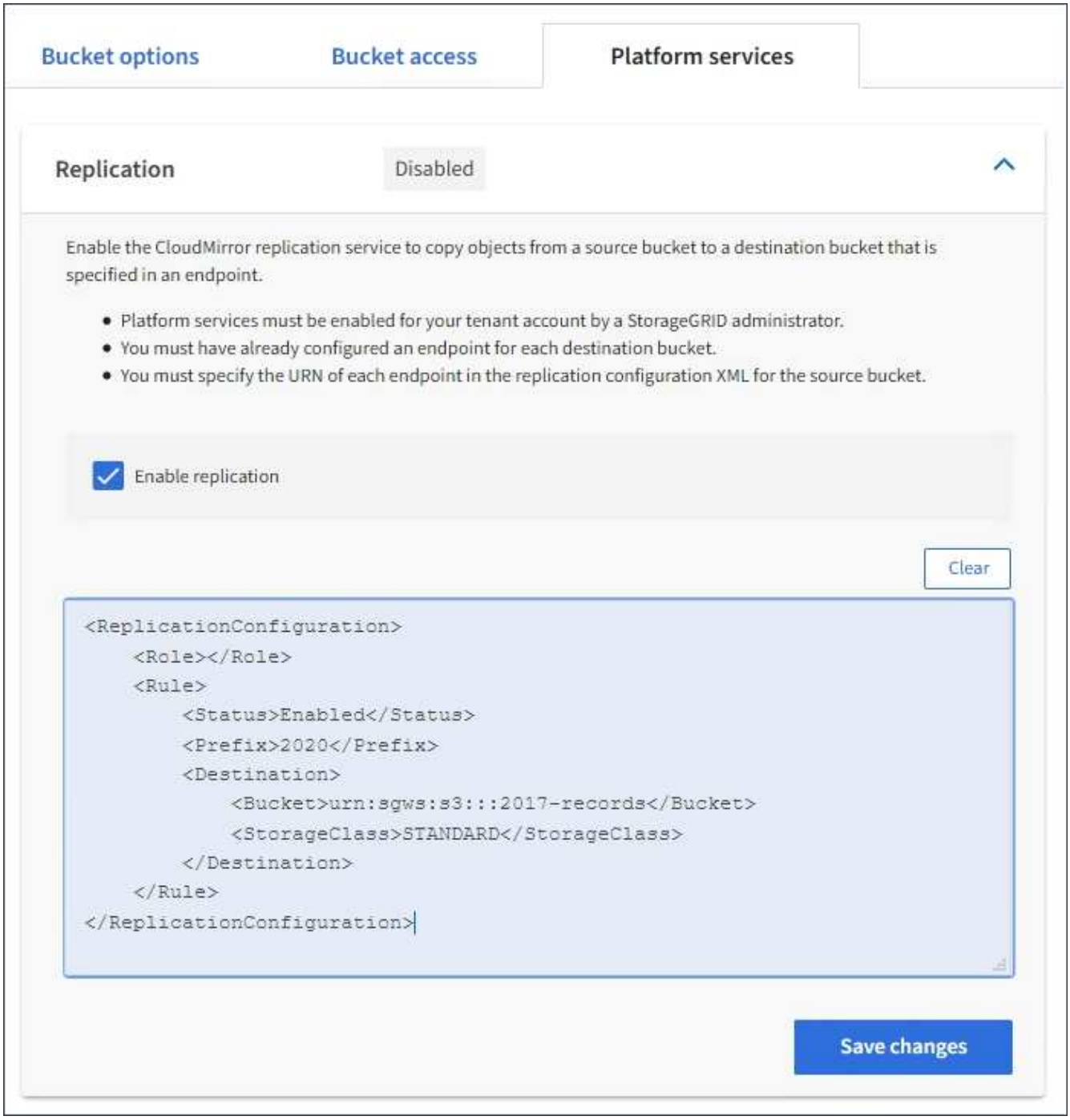
   - StorageGRID는 복제 구성의 V1만 지원합니다. 즉, StorageGRID는 규칙에 대해 'Filter' 요소의 사용을 지원하지 않으며 개체 버전 삭제에 대해서는 V1 규약을 따릅니다. 자세한 내용은 복제 구성에 대한 Amazon 설명서를 참조하십시오.
   - S3 버킷 엔드포인트의 URN을 대상으로 사용합니다.
   - 필요한 경우 "<StorageClass>" 요소를 추가하고 다음 중 하나를 지정합니다.
     - 'Standard': 기본 스토리지 클래스. 객체를 업로드할 때 스토리지 클래스를 지정하지 않으면 '표준' 스토리지 클래스가 사용됩니다.
     - S tandard_IA: (Standard - Infrequent Access) 액세스 빈도가 낮지만 필요한 경우 빠른 액세스가 필요한 데이터에 이 스토리지 클래스를 사용합니다.
     - Reduced_redundancy: standard 스토리지 클래스보다 중복성이 적은 비위험, 재현 가능한 데이터에 이 스토리지 클래스를 사용합니다.
   - 구성 XML에서 Role을 지정하면 무시됩니다. 이 값은 StorageGRID에서 사용되지 않습니다.

```
<ReplicationConfiguration>
    <Role></Role>
    <Rule>
        <Status>Enabled</Status>
        <Prefix>2020</Prefix>
        <Destination>
           <Bucket>urn:sgws:s3:::2017-records</Bucket>
           <StorageClass>STANDARD</StorageClass>
         </Destination>
     </Rule>
</ReplicationConfiguration>
```

2. 테넌트 관리자에서 * 스토리지(S3) * > * 버킷 * 을 선택합니다.
3. 소스 버킷의 이름을 선택합니다.

   버킷 세부 정보 페이지가 나타납니다.

4. 플랫폼 서비스 * > * 복제 * 를 선택합니다.
5. 복제 사용 * 확인란을 선택합니다.
6. 복제 구성 XML을 텍스트 상자에 붙여 넣고 * 변경 내용 저장 * 을 선택합니다.



> 그리드 관리자 또는 그리드 관리 API를 사용하여 StorageGRID 관리자가 각 테넌트 계정에 대해 플랫폼 서비스를 활성화해야 합니다. 구성 XML을 저장할 때 오류가 발생하면 StorageGRID 관리자에게 문의하십시오.

7. 복제가 올바르게 구성되었는지 확인합니다.
   a. 복제 구성에 지정된 대로 복제 요구 사항을 충족하는 객체를 소스 버킷에 추가합니다.

      앞서 설명한 예에서는 접두사 "2020"과 일치하는 객체가 복제됩니다.

   b. 객체가 대상 버킷에 복제되었는지 확인합니다.

      오브젝트 크기가 작은 경우 복제가 빠르게 수행됩니다.

관련 정보

S3을 사용합니다

플랫폼 서비스 끝점을 만듭니다

이벤트 알림을 구성합니다

알림 서비스는 세 가지 StorageGRID 플랫폼 서비스 중 하나입니다. 버킷에 대한 알림을 활성화하여 지정된 이벤트에 대한 정보를 AWS SNS(Simple Notification Service ™)를 지원하는 대상 서비스로 전송할 수 있습니다.

필요한 것

- StorageGRID 관리자가 테넌트 계정에 대해 플랫폼 서비스를 사용하도록 설정해야 합니다.
- 알림 소스로 사용하려면 이미 버킷을 만들어야 합니다.
- 이벤트 알림 대상으로 사용하려는 엔드포인트가 이미 있어야 하며 URN이 있어야 합니다.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이 권한을 사용하여 테넌트 계정의 모든 S3 버킷에 대한 설정을 관리할 수 있습니다. 이러한 권한은 테넌트 관리자를 사용하여 버킷을 구성할 때 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업에 대해

이벤트 알림을 구성한 후 소스 버킷의 개체에 대해 지정된 이벤트가 발생할 때마다 알림이 생성되어 대상 끝점으로 사용되는 SNS(Simple Notification Service) 항목으로 전송됩니다. 버킷에 대한 알림을 활성화하려면 유효한 알림 구성 XML을 생성하고 적용해야 합니다. 알림 구성 XML은 각 대상에 대해 이벤트 알림 끝점의 URN을 사용해야 합니다.

이벤트 알림 및 구성 방법에 대한 일반 정보는 아마존 문서를 참조하십시오. StorageGRID에서 S3 버킷 알림 구성 API를 구현하는 방법에 대한 자세한 내용은 S3 클라이언트 애플리케이션 구현 지침을 참조하십시오.

객체가 포함된 버킷에 대해 이벤트 알림을 활성화하면 알림 구성이 저장된 후 수행되는 작업에 대해서만 알림이 전송됩니다.

단계

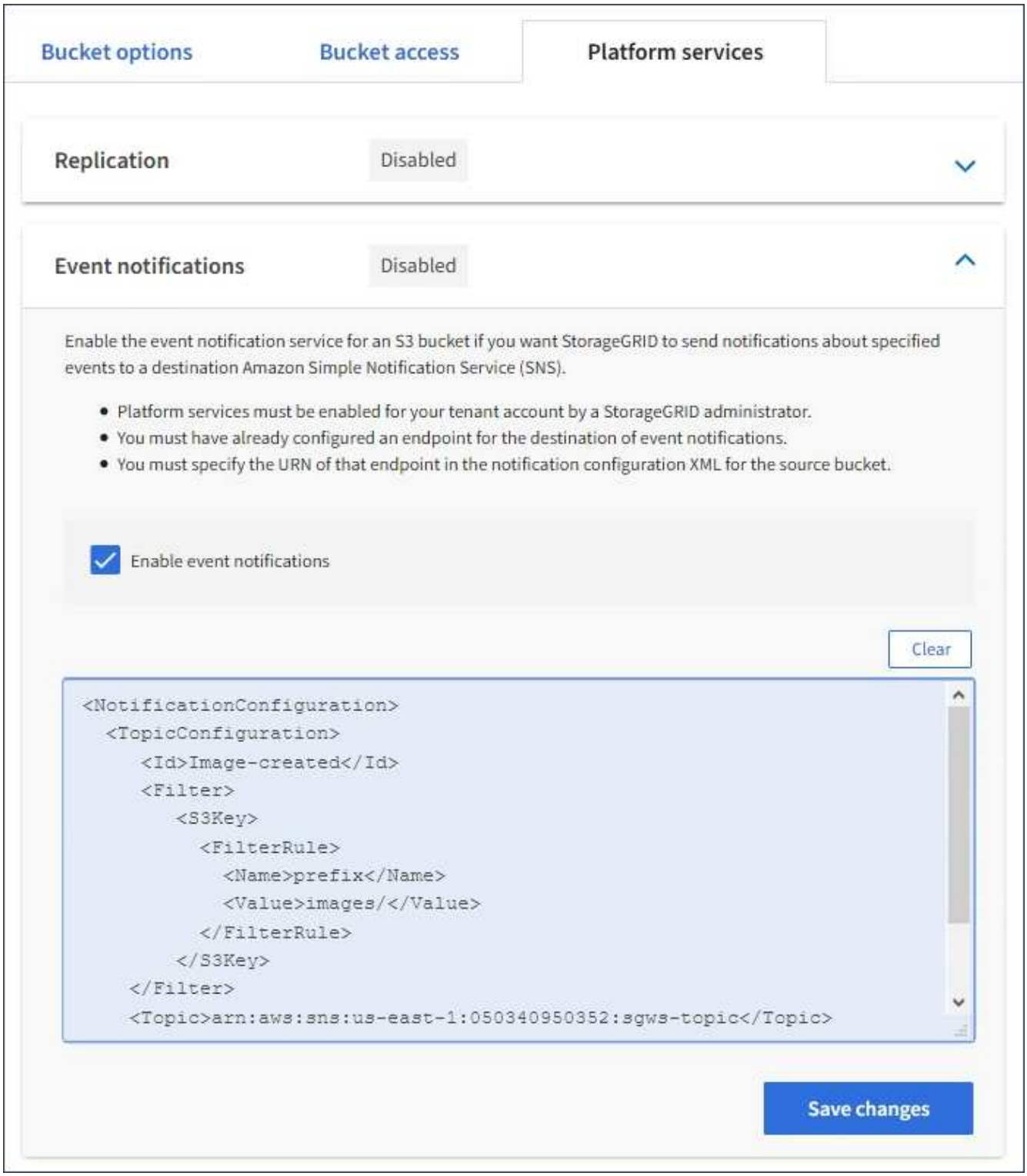
1. 소스 버킷에 대한 알림 활성화:
    - 텍스트 편집기를 사용하여 S3 알림 API에 지정된 대로 이벤트 알림을 활성화하는 데 필요한 알림 구성 XML을 생성합니다.
    - XML을 구성할 때는 이벤트 알림 끝점의 URN을 대상 항목으로 사용합니다.

```
<NotificationConfiguration>
  <TopicConfiguration>
     <Id>Image-created</Id>
     <Filter>
        <S3Key>
          <FilterRule>
            <Name>prefix</Name>
            <Value>images/</Value>
          </FilterRule>
        </S3Key>
    </Filter>
    <Topic>arn:aws:sns:us-east-1:050340950352:sgws-topic</Topic>
    <Event>s3:ObjectCreated:*</Event>
   </TopicConfiguration>
 </NotificationConfiguration>
```

2. 테넌트 관리자에서 * 스토리지(S3) * > * 버킷 * 을 선택합니다.
3. 소스 버킷의 이름을 선택합니다.

   버킷 세부 정보 페이지가 나타납니다.

4. 플랫폼 서비스 * > * 이벤트 알림 * 을 선택합니다.
5. 이벤트 알림 사용 * 확인란을 선택합니다.
6. 알림 구성 XML을 텍스트 상자에 붙여 넣고 * 변경 내용 저장 * 을 선택합니다.



> 그리드 관리자 또는 그리드 관리 API를 사용하여 StorageGRID 관리자가 각 테넌트 계정에 대해 플랫폼 서비스를 활성화해야 합니다. 구성 XML을 저장할 때 오류가 발생하면 StorageGRID 관리자에게 문의하십시오.

7. 이벤트 알림이 올바르게 구성되었는지 확인합니다.

   a. 구성 XML에 구성된 알림을 트리거하기 위한 요구 사항을 충족하는 소스 버킷의 객체에 대한 작업을 수행합니다.

이 예제에서는 "images/" 접두사로 객체를 만들 때마다 이벤트 알림이 전송됩니다.

b. 알림이 대상 SNS 항목으로 전달되었는지 확인합니다.

예를 들어 대상 주제가 AWS SNS(Simple Notification Service)에 호스팅된 경우, 알림 전송 시 이메일을 보내도록 서비스를 구성할 수 있습니다.


```
{
   "Records":[
      {
         "eventVersion":"2.0",
         "eventSource":"sgws:s3",
         "eventTime":"2017-08-08T23:52:38Z",
         "eventName":"ObjectCreated:Put",
         "userIdentity":{
            "principalId":"1111111111111111111"
         },
         "requestParameters":{
            "sourceIPAddress":"193.51.100.20"
         },
         "responseElements":{
            "x-amz-request-id":"122047343"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"Image-created",
            "bucket":{
               "name":"test1",
               "ownerIdentity":{
                  "principalId":"1111111111111111111"
               },
               "arn":"arn:sgws:s3:::test1"
            },
            "object":{
               "key":"images/cat.jpg",
               "size":0,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
               "sequencer":"14D90402421461C7"
            }
         }
      }
   ]
}
```


대상 항목에서 알림이 수신되면 StorageGRID 알림에 대한 소스 버킷을 성공적으로 구성한 것입니다.

관련 정보

버킷에 대한 알림을 이해합니다

S3을 사용합니다

플랫폼 서비스 끝점을 만듭니다

검색 통합 서비스를 사용합니다

검색 통합 서비스는 세 가지 StorageGRID 플랫폼 서비스 중 하나입니다. 오브젝트 생성, 삭제 또는 해당 메타데이터 또는 태그가 업데이트될 때마다 대상 검색 인덱스에 오브젝트 메타데이터를 전송하도록 이 서비스를 활성화할 수 있습니다.

테넌트 관리자를 사용하여 버킷에 사용자 지정 StorageGRID 구성 XML을 적용하여 검색 통합을 구성할 수 있습니다.

> 검색 통합 서비스로 인해 개체 메타데이터가 대상으로 전송되기 때문에 해당 구성 XML을 _ 메타데이터 알림 구성 xml _ 이라고 합니다. 이 구성 XML은 이벤트 알림을 설정하는 데 사용되는 _notification 구성 xml_과 다릅니다.

를 참조하십시오 S3 클라이언트 애플리케이션 구현 지침 다음 사용자 지정 StorageGRID S3 REST API 작업에 대한 자세한 내용은 다음을 참조하십시오.

- 버킷 메타데이터 알림 구성 요청을 삭제합니다
- 버킷 메타데이터 알림 구성 요청을 가져옵니다
- PUT 버킷 메타데이터 알림 구성 요청

관련 정보

검색 통합을 위한 구성 XML

메타데이터 알림에 포함된 개체 메타데이터입니다

JSON이 검색 통합 서비스에 의해 생성되었습니다

검색 통합 서비스를 구성합니다

S3을 사용합니다

검색 통합을 위한 구성 **XML**

검색 통합 서비스는 "<MetadataNotificationConfiguration>" 및 "</MetadataNotificationConfiguration>" 태그에 포함된 규칙 집합을 사용하여 구성됩니다. 각 규칙은 규칙이 적용되는 오브젝트와 StorageGRID가 해당 오브젝트의 메타데이터를 보내야 하는 대상을 지정합니다.

개체 이름의 접두어를 기준으로 개체를 필터링할 수 있습니다. 예를 들어, 접두어 '이미지'가 있는 객체에 대한 메타데이터를 한 대상으로, 접두어 '비디오'가 있는 객체에 대한 메타데이터를 다른 대상으로 전송할 수 있습니다. 중복되는 접두사가 있는 구성은 유효하지 않으며 제출될 때 거부됩니다. 예를 들어 접두사 test를 가진 개체에 대해 하나의 규칙과 접두사 test2 를 가진 개체에 대한 두 번째 규칙을 포함하는 구성은 허용되지 않습니다.

검색 통합 서비스를 위해 생성된 StorageGRID 엔드포인트의 URN을 사용하여 대상을 지정해야 합니다. 이러한 엔드포인트는 Elasticsearch 클러스터에 정의된 인덱스 및 유형을 나타냅니다.

```
<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. • 세 번째 요소는 'es'여야 합니다. • URN은 메타데이터가 저장된 인덱스 및 형식으로 domain-name/myindex/MyType 형식으로 끝나야 합니다. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 다음과 같은 형식을 취합니다. • "arn:aws: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

샘플 메타데이터 알림 구성 XML을 사용하여 고유한 XML을 구성하는 방법을 배웁니다.

모든 개체에 적용되는 메타데이터 알림 구성입니다

이 예제에서 모든 오브젝트의 오브젝트 메타데이터는 동일한 대상으로 전송됩니다.

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

두 가지 규칙을 사용하여 메타데이터 알림 구성

이 예에서는 접두사 /images와 일치하는 객체의 객체 메타데이터가 한 대상으로 전송되고 접두사 /videos와 일치하는 객체의 객체 메타데이터는 두 번째 대상으로 전송됩니다.

```
<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>
```

관련 정보

S3을 사용합니다

메타데이터 알림에 포함된 개체 메타데이터입니다

JSON이 검색 통합 서비스에 의해 생성되었습니다

검색 통합 서비스를 구성합니다

검색 통합 서비스는 개체가 생성, 삭제 또는 해당 메타데이터 또는 태그가 업데이트될 때마다 대상 검색 인덱스에 개체 메타데이터를 보냅니다.

필요한 것

- StorageGRID 관리자가 테넌트 계정에 대해 플랫폼 서비스를 사용하도록 설정해야 합니다.
- 인덱싱할 콘텐츠가 있는 S3 버킷을 이미 생성해야 합니다.
- 검색 통합 서비스의 대상으로 사용하려는 끝점이 이미 있어야 하며 URN이 있어야 합니다.
- 모든 버킷 관리 또는 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 이 권한을 사용하여 테넌트 계정의 모든 S3 버킷에 대한 설정을 관리할 수 있습니다. 이러한 권한은 테넌트 관리자를 사용하여 버킷을 구성할 때 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업에 대해

소스 버킷에 대한 검색 통합 서비스를 구성한 후 객체를 만들거나 객체의 메타데이터 또는 태그를 업데이트하면 대상 엔드포인트로 객체 메타데이터가 전송됩니다. 이미 객체가 포함된 버킷에 대해 검색 통합 서비스를 활성화하면 기존 객체에 대한 메타데이터 알림이 자동으로 전송되지 않습니다. 이러한 기존 개체를 업데이트하여 대상 검색 인덱스에 해당 메타데이터가 추가되도록 해야 합니다.

단계

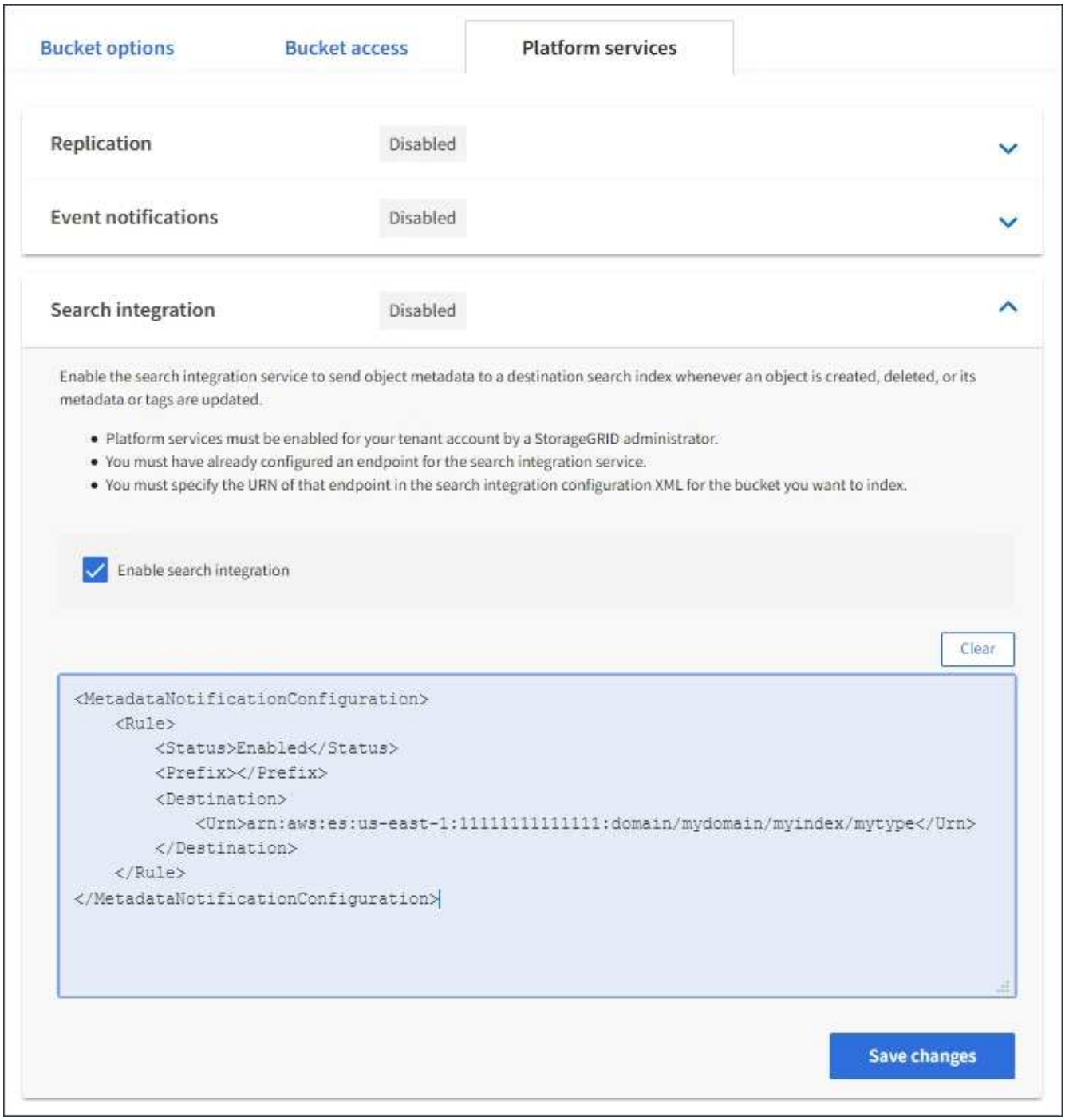
1. 텍스트 편집기를 사용하여 검색 통합을 활성화하는 데 필요한 메타데이터 알림 XML을 만듭니다.
   - 검색 통합을 위한 구성 XML에 대한 정보를 참조하십시오.
   - XML을 구성할 때는 검색 통합 끝점의 URN을 대상으로 사용합니다.

```
<MetadataNotificationConfiguration>
 <Rule>
    <Status>Enabled</Status>
    <Prefix></Prefix>
    <Destination>
       <Urn>arn:aws:es:us-east-
1:11111111111111:domain/mydomain/myindex/mytype</Urn>
       </Destination>
 </Rule>
</MetadataNotificationConfiguration>
```

2. 테넌트 관리자에서 * 스토리지(S3) * > * 버킷 * 을 선택합니다.
3. 소스 버킷의 이름을 선택합니다.

   버킷 세부 정보 페이지가 나타납니다.

4. 플랫폼 서비스 * > * 통합 검색 * 을 선택합니다
5. 검색 통합 사용 * 확인란을 선택합니다.
6. 메타데이터 알림 구성을 텍스트 상자에 붙여 넣고 * 변경 내용 저장 * 을 선택합니다.



> 그리드 관리자 또는 관리 API를 사용하여 StorageGRID 관리자가 각 테넌트 계정에 대해 플랫폼 서비스를 활성화해야 합니다. 구성 XML을 저장할 때 오류가 발생하면 StorageGRID 관리자에게 문의하십시오.

7. 검색 통합 서비스가 올바르게 구성되었는지 확인합니다.
   a. 구성 XML에 지정된 대로 메타데이터 알림을 트리거하기 위한 요구 사항을 충족하는 객체를 소스 버킷에 추가합니다.

      앞의 예제에서 버킷에 추가된 모든 오브젝트는 메타데이터 알림을 트리거합니다.

   b. 개체의 메타데이터와 태그가 포함된 JSON 문서가 끝점에 지정된 검색 인덱스에 추가되었는지 확인합니다.

작업을 마친 후

필요에 따라 다음 방법 중 하나를 사용하여 버킷에 대한 검색 통합을 비활성화할 수 있습니다.

- 스토리지(S3) * > * 버킷 * 을 선택하고 * 검색 통합 활성화 * 확인란의 선택을 취소합니다.
- S3 API를 직접 사용하는 경우 Delete Bucket 메타데이터 알림 요청을 사용합니다. S3 클라이언트 애플리케이션 구현 지침을 참조하십시오.

관련 정보

검색 통합 서비스를 이해합니다

검색 통합을 위한 구성 XML

S3을 사용합니다

플랫폼 서비스 끝점을 만듭니다

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 예에서는 'test'라는 버킷에 'gws/tagging.txt' 키가 있는 객체가 생성될 때 생성될 수 있는 JSON의 예를 보여 줍니다. 시험용 버킷은 버전 관리가 되지 않아 rionId 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 및 설명입니다
버킷 및 오브젝트 정보	버켓 이름
키 : 개체 키 이름	거장된 버킷을 가진 개체의 개체 버전
'리기온': '우동-1'과 같은 버켓 지역	시스템 메타데이터
'크기': HTTP 클라이언트에 표시되는 개체 크기(바이트)입니다	'mD5': 객체 해시
사용자 메타데이터	metadata: 객체에 대한 모든 사용자 메타데이터를 키 값 쌍으로 사용합니다 키: 값
태그	"태그": 오브젝트에 대해 정의된 모든 오브젝트 태그는 키 값 쌍으로 제공됩니다 키: 값

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.

S3을 사용합니다

S3:개요를 사용합니다

StorageGRID는 REST(Representational State Transfer) 웹 서비스 세트로 구현되는 S3(Simple Storage Service) API를 지원합니다. S3 REST API를 지원하므로 StorageGRID 시스템을 사용하는 사내 오브젝트 스토리지와 S3 웹 서비스를 위해 개발된 서비스 지향 애플리케이션을 연결할 수 있습니다. 따라서 클라이언트 애플리케이션의 현재 S3 REST API 호출 사용에 대한 변경이 최소화됩니다.

S3 REST API 지원으로 변경

S3 REST API에 대한 StorageGRID 시스템의 지원 변경사항을 알고 있어야 합니다.

놓습니다	설명
11.6	• 객체 가져오기 및 헤드 객체 요청에 'PARTNUMBER' 요청 매개 변수 사용 지원 추가 • S3 오브젝트 잠금의 버킷 레벨에서 기본 보존 모드 및 기본 보존 기간에 대한 지원이 추가되었습니다. • 객체에 대해 허용 가능한 보존 기간 범위를 설정하기 위해 '객체 잠금 잔여 보존 기간' 정책 조건 키에 대한 지원이 추가되었습니다. • 단일 PUT 오브젝트 작업의 maximum_recommended_size가 이제 5GiB(5,368,709,120바이트)입니다. 5GiB보다 큰 객체가 있는 경우 대신 멀티파트 업로드를 사용합니다. ⓘ StorageGRID 11.6에서 단일 PUT 객체 작업에 대한 Maximum_supported_size는 5TiB로 유지됩니다(5,497,558,138,880 바이트). 그러나 5GiB를 초과하는 개체를 업로드하려고 하면 * S3 PUT 오브젝트 크기가 너무 큼 * 경고가 트리거됩니다.
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. • Content-MD5 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. • '3:ObjectRestore:Post' 이벤트 유형에 대한 버킷 알림 지원이 추가되었습니다. • 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 지원 및 지원되는 TLS 암호 제품군의 업데이트된 목록이 추가되었습니다. • CLB 서비스는 더 이상 사용되지 않습니다.
11.3	• 고객이 제공한 키(SSE-C)를 사용하여 오브젝트 데이터의 서버측 암호화에 대한 지원이 추가되었습니다. • Bucket 수명주기 작업(만료 작업에만 해당) 및 "x-amz-expiration" 응답 헤더에 대한 삭제, 가져오기 및 PUT 지원이 추가되었습니다. • 수집 시 동기식 배치를 사용하는 ILM 규칙의 영향을 설명하기 위해 PUT 개체, Put Object-Copy 및 MultiPart Upload가 업데이트되었습니다. • 지원되는 TLS 암호 그룹 목록이 업데이트되었습니다. TLS 1.1 암호가 더 이상 지원되지 않습니다.

놓습니다	설명
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 버킷의 규정 준수 설정에 대한 지원이 추가되었습니다.
11.0	버킷에 대한 플랫폼 서비스(CloudMirror 복제, 알림 및 Elasticsearch 검색 통합) 구성 지원 추가 또한 버킷에 대한 객체 태그 지정 위치 제약 조건 및 사용 가능한 정합성 제어 설정에 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.

지원되는 버전

StorageGRID는 다음과 같은 S3 및 HTTP 버전을 지원합니다.

항목	버전
S3 사양	_Simple Storage Service API Reference_2006-03-01
HTTP	1.1 HTTP에 대한 자세한 내용은 HTTP/1.1(RFC 7230-35)을 참조하십시오. • 참고 *: StorageGRID는 HTTP/1.1 파이프라이닝을 지원하지 않습니다.

관련 정보

"IETF RFC 2616:HTTP/1.1(Hypertext Transfer Protocol)"

"AWS(Amazon Web Services) 문서: Amazon Simple Storage Service API Reference 를 참조하십시오"

StorageGRID 플랫폼 서비스 지원

StorageGRID 플랫폼 서비스를 사용하면 StorageGRID 테넌트 계정에서 원격 S3 버킷, SNS(Simple Notification Service) 엔드포인트 또는 Elasticsearch 클러스터와 같은 외부 서비스를 활용하여 그리드에 의해 제공되는 서비스를 확장할 수 있습니다.

다음 표에는 사용 가능한 플랫폼 서비스와 이를 구성하는 데 사용되는 S3 API가 요약되어 있습니다.

플랫폼 서비스	목적	**S3 API**를 사용하여 서비스를 구성합니다
CloudMirror 복제	소스 StorageGRID 버킷에서 구성된 원격 S3 버킷으로 오브젝트를 복제합니다.	버킷 복제를 배치합니다
알림	소스 StorageGRID 버킷의 이벤트에 대한 알림을 구성된 SNS(Simple Notification Service) 엔드포인트로 보냅니다.	버킷 통지를 보냅니다
검색 통합	StorageGRID 버킷에 저장된 객체에 대한 객체 메타데이터를 구성된 Elasticsearch 인덱스로 전송합니다.	Bucket 메타데이터 알림을 배치합니다 • 참고: * 이것은 StorageGRID 사용자 정의 S3 API입니다.

그리드 관리자는 테넌트 계정에 대해 플랫폼 서비스를 사용하도록 설정해야 사용할 수 있습니다. 그런 다음 테넌트 관리자는 테넌트 계정의 원격 서비스를 나타내는 끝점을 만들어야 합니다. 서비스를 구성하려면 이 단계가 필요합니다.

플랫폼 서비스 사용을 위한 권장 사항

플랫폼 서비스를 사용하기 전에 다음 권장 사항을 숙지해야 합니다.

- CloudMirror 복제, 알림 및 검색 통합이 필요한 S3 요청을 가진 활성 테넌트 100개 이상을 허용하지 않는 것이 좋습니다. 활성 테넌트가 100개 이상인 경우 S3 클라이언트 성능이 저하될 수 있습니다.
- StorageGRID 시스템의 S3 버켓에서 버전 관리 및 CloudMirror 복제를 모두 사용하는 경우, 대상 엔드포인트에 S3 버킷 버전 관리도 활성화할 것을 권장합니다. 이를 통해 CloudMirror 복제가 엔드포인트에 비슷한 개체 버전을 생성할 수 있습니다.
- 소스 버킷에 S3 오브젝트 잠금이 설정된 경우 CloudMirror 복제가 지원되지 않습니다.
- 대상 버킷에 레거시 규정 준수 기능이 설정된 경우 CloudMirror 복제가 실패하고 AccessDenied 오류가 표시됩니다.

관련 정보

테넌트 계정을 사용합니다

StorageGRID 관리

버킷 작업

PUT 버킷 메타데이터 알림 구성 요청

테넌트 계정 및 연결을 구성합니다

클라이언트 응용 프로그램에서 연결을 허용하도록 StorageGRID를 구성하려면 하나 이상의 테넌트 계정을 만들고 연결을 설정해야 합니다.

S3 테넌트 계정 생성 및 구성

S3 API 클라이언트가 StorageGRID에 오브젝트를 저장하고 검색할 수 있으려면 먼저 S3 테넌트 계정이 필요합니다. 각 테넌트 계정에는 고유한 계정 ID, 그룹 및 사용자, 컨테이너 및 객체가 있습니다.

S3 테넌트 계정은 StorageGRID 그리드 관리자가 그리드 관리자 또는 그리드 관리 API를 사용하여 생성합니다. S3 테넌트 계정을 생성할 때 그리드 관리자는 다음 정보를 지정합니다.

- 테넌트의 표시 이름(테넌트의 계정 ID가 자동으로 할당되며 변경할 수 없음)
- 테넌트 계정이 플랫폼 서비스를 사용하도록 허용되는지 여부 플랫폼 서비스를 사용할 수 있는 경우 그리드 사용을 지원하도록 구성해야 합니다.
- 필요한 경우 테넌트 계정의 스토리지 할당량 — 테넌트의 객체에 사용할 수 있는 최대 GB, 테라바이트 또는 PB입니다. 테넌트의 스토리지 할당량은 물리적 크기(디스크 크기)가 아닌 논리적 양(오브젝트 크기)을 나타냅니다.
- StorageGRID 시스템에 대해 ID 페더레이션이 설정된 경우 테넌트 계정을 구성할 수 있는 루트 액세스 권한이 있는 통합 그룹입니다.
- StorageGRID 시스템에서 SSO(Single Sign-On)를 사용하지 않는 경우 테넌트 계정이 자체 ID 소스를 사용할지 또는 그리드의 ID 소스를 공유할지 여부 및 테넌트의 로컬 루트 사용자의 초기 암호를 공유할지 여부

S3 테넌트 계정이 생성된 후 테넌트 사용자는 테넌트 관리자에 액세스하여 다음과 같은 작업을 수행할 수 있습니다.

- ID 소스를 그리드와 공유하지 않는 경우 ID 페더레이션을 설정하고 로컬 그룹 및 사용자를 만듭니다
- S3 액세스 키를 관리합니다
- S3 오브젝트 잠금이 설정된 버킷을 포함하여 S3 버킷을 생성하고 관리합니다
- 플랫폼 서비스 사용(활성화된 경우)
- 스토리지 사용량을 모니터링합니다

> S3 테넌트 사용자는 테넌트 관리자를 사용하여 S3 버킷을 생성 및 관리할 수 있지만, S3 액세스 키를 가지고 S3 REST API를 사용하여 오브젝트를 수집 및 관리해야 합니다.

관련 정보

StorageGRID 관리

테넌트 계정을 사용합니다

클라이언트 연결 구성 방법

그리드 관리자는 S3 클라이언트가 StorageGRID에 연결하여 데이터를 저장 및 검색하는 방법에 영향을 주는 구성을 선택합니다. 연결에 필요한 특정 정보는 선택한 구성에 따라 다릅니다.

클라이언트 응용 프로그램은 다음 중 하나를 연결하여 개체를 저장하거나 검색할 수 있습니다.

- 관리 노드 또는 게이트웨이 노드의 부하 분산 서비스 또는 선택적으로 관리 노드 또는 게이트웨이 노드의 고가용성(HA) 그룹의 가상 IP 주소입니다
- 게이트웨이 노드의 CLB 서비스 또는 게이트웨이 노드의 고가용성 그룹의 가상 IP 주소(선택 사항)입니다

> CLB 서비스는 더 이상 사용되지 않습니다. StorageGRID 11.3 릴리스 전에 구성된 클라이언트는 게이트웨이 노드에서 CLB 서비스를 계속 사용할 수 있습니다. 로드 밸런싱을 제공하기 위해 StorageGRID에 의존하는 다른 모든 클라이언트 애플리케이션은 로드 밸런서 서비스를 사용하여 연결해야 합니다.

- 외부 로드 밸런서가 있거나 없는 스토리지 노드

StorageGRID를 구성할 때 그리드 관리자는 그리드 관리자 또는 그리드 관리 API를 사용하여 다음 단계를 수행할 수 있습니다. 이 모든 단계는 선택 사항입니다.

1. 로드 밸런서 서비스의 끝점을 구성합니다.

   로드 밸런서 서비스를 사용하려면 끝점을 구성해야 합니다. 관리 노드 또는 게이트웨이 노드의 부하 분산 서비스는 들어오는 네트워크 연결을 클라이언트 애플리케이션에서 스토리지 노드로 분산합니다. 로드 밸런서 끝점을 만들 때 StorageGRID 관리자는 포트 번호, 엔드포인트가 HTTP 또는 HTTPS 연결을 수락하는지 여부, 엔드포인트를 사용할 클라이언트 유형(S3 또는 Swift) 및 HTTPS 연결에 사용할 인증서(해당하는 경우)를 지정합니다.

2. 신뢰할 수 없는 클라이언트 네트워크를 구성합니다.

   StorageGRID 관리자가 노드의 클라이언트 네트워크를 신뢰할 수 없도록 구성하는 경우 노드는 로드 밸런서 끝점으로 명시적으로 구성된 포트의 클라이언트 네트워크에서 인바운드 연결만 허용합니다.

3. 고가용성 그룹을 구성합니다.

   관리자가 HA 그룹을 생성하면 여러 관리 노드 또는 게이트웨이 노드의 네트워크 인터페이스가 액티브-백업 구성에 배치됩니다. HA 그룹의 가상 IP 주소를 사용하여 클라이언트 연결이 이루어집니다.

각 옵션에 대한 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

관련 정보

StorageGRID 관리

요약**:** 클라이언트 연결을 위한 **IP** 주소 및 포트

클라이언트 애플리케이션은 그리드 노드의 IP 주소와 해당 노드의 서비스 포트 번호를 사용하여 StorageGRID에 접속합니다. HA(고가용성) 그룹이 구성되어 있는 경우 클라이언트 애플리케이션은 HA 그룹의 가상 IP 주소를 사용하여 연결할 수 있습니다.

클라이언트 연결을 만드는 데 필요한 정보입니다

이 표에는 클라이언트가 StorageGRID에 연결할 수 있는 다양한 방법과 각 연결 유형에 사용되는 IP 주소 및 포트가 요약되어 있습니다. 자세한 내용은 StorageGRID 관리자에게 문의하거나 StorageGRID 관리 지침 에서 그리드 관리자에서 이 정보를 찾는 방법에 대한 설명을 참조하십시오.

연결 위치	클라이언트가 연결하는 서비스입니다	**IP** 주소입니다	포트
HA 그룹	로드 밸런서	HA 그룹의 가상 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
HA 그룹	CLB 참고:** CLB 서비스는 더 이상 사용되지 않습니다.	HA 그룹의 가상 IP 주소입니다	기본 S3 포트: • HTTPS: 8082 • HTTP: 8084
관리자 노드	로드 밸런서	관리 노드의 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
게이트웨이 노드	로드 밸런서	게이트웨이 노드의 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
게이트웨이 노드	CLB 참고:** CLB 서비스는 더 이상 사용되지 않습니다.	게이트웨이 노드의 IP 주소입니다 • 참고:** 기본적으로 CLB 및 LDR에 대한 HTTP 포트는 사용되지 않습니다.	기본 S3 포트: • HTTPS: 8082 • HTTP: 8084
스토리지 노드	LDR	스토리지 노드의 IP 주소입니다	기본 S3 포트: • HTTPS: 18082 • HTTP: 18084

예

S3 클라이언트를 게이트웨이 노드 HA 그룹의 로드 밸런서 끝점에 연결하려면 아래와 같이 구조화된 URL을 사용합니다.

- "https://*VIP-of-HA-group*:_LB-endpoint-port_`

예를 들어 HA 그룹의 가상 IP 주소가 192.0.2.5이고 S3 로드 밸런서 끝점의 포트 번호가 10443인 경우 S3 클라이언트는 다음 URL을 사용하여 StorageGRID에 연결할 수 있습니다.

- "https://192.0.2.5:10443`

클라이언트가 StorageGRID에 연결하는 데 사용하는 IP 주소에 대한 DNS 이름을 구성할 수 있습니다. 로컬 네트워크 관리자에게 문의하십시오.

관련 정보

StorageGRID 관리

HTTPS 또는 **HTTP** 연결을 사용하도록 결정합니다

로드 밸런서 끝점을 사용하여 클라이언트 연결을 만들 때는 해당 끝점에 지정된 프로토콜(HTTP 또는 HTTPS)을 사용하여 연결해야 합니다. 스토리지 노드 또는 게이트웨이 노드의 CLB 서비스에 대한 클라이언트 연결에 HTTP를 사용하려면 해당 사용을 설정해야 합니다.

기본적으로 클라이언트 응용 프로그램이 게이트웨이 노드의 스토리지 노드 또는 CLB 서비스에 연결할 때는 모든 연결에 암호화된 HTTPS를 사용해야 합니다. 선택적으로 Grid Manager에서 * HTTP Connection * 그리드 사용 옵션을 선택하여 보안성이 떨어지는 HTTP 연결을 활성화할 수 있습니다. 예를 들어, 클라이언트 애플리케이션은 비운영 환경에서 스토리지 노드에 대한 접속을 테스트할 때 HTTP를 사용할 수 있습니다.

> 요청은 암호화되지 않은 상태로 전송되므로 프로덕션 그리드에 대해 HTTP를 설정할 때는 주의해야 합니다.

> CLB 서비스는 더 이상 사용되지 않습니다.

HTTP 연결 사용 * 옵션을 선택한 경우 클라이언트는 HTTPS에 사용하는 것과 다른 HTTP 포트를 사용해야 합니다. StorageGRID 관리 지침을 참조하십시오.

관련 정보

StorageGRID 관리

활성, 유휴 및 동시 HTTP 연결의 이점

S3 요청에 대한 끝점 도메인 이름입니다

클라이언트 요청에 S3 도메인 이름을 사용하려면 StorageGRID 관리자가 S3 경로 스타일 및 S3 가상 호스팅 스타일 요청에서 S3 도메인 이름을 사용하는 연결을 허용하도록 시스템을 구성해야 합니다.

이 작업에 대해

S3 가상 호스팅 스타일 요청을 사용하려면 그리드 관리자가 다음 작업을 수행해야 합니다.

- 그리드 관리자를 사용하여 StorageGRID 시스템에 S3 끝점 도메인 이름을 추가합니다.
- 클라이언트가 StorageGRID에 대한 HTTPS 연결에 사용하는 인증서가 클라이언트에 필요한 모든 도메인 이름에 서명되었는지 확인합니다.

  예를 들어, 끝점이 '3.company.com` 인 경우 그리드 관리자는 HTTPS 연결에 사용되는 인증서에 '3.company.com` 끝점 및 끝점 와일드카드 주체 대체 이름(SAN):' * .s3.company.com` 가 포함되어 있는지 확인해야 합니다.

- 필요한 와일드카드 레코드를 포함하여 끝점 도메인 이름과 일치하는 DNS 레코드를 포함하도록 클라이언트에서 사용하는 DNS 서버를 구성합니다.

클라이언트가 로드 밸런서 서비스를 사용하여 연결하는 경우 그리드 관리자가 구성하는 인증서는 클라이언트가 사용하는 로드 밸런서 끝점에 대한 인증서입니다.

> 각 로드 밸런서 끝점마다 고유한 인증서가 있으며 각 끝점이 서로 다른 끝점 도메인 이름을 인식하도록 구성할 수 있습니다.

클라이언트가 스토리지 노드 또는 게이트웨이 노드의 CLB 서비스에 연결하는 경우 그리드 관리자가 구성하는 인증서는

그리드에 사용되는 단일 사용자 지정 서버 인증서입니다.

> CLB 서비스는 더 이상 사용되지 않습니다.

자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

이러한 단계를 완료한 후에는 가상 호스팅 스타일 요청(예: ' bucket.s3.company.com`)을 사용할 수 있습니다.

관련 정보

StorageGRID 관리

REST API에 대한 보안을 구성합니다

S3 REST API 구성을 테스트합니다

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

필요한 것

- 에서 AWS CLI를 다운로드하여 설치했습니다 "aws.amazon.com/cli".
- StorageGRID 시스템에서 S3 테넌트 계정을 생성했습니다.

단계

1. StorageGRID 시스템에서 생성한 계정을 사용하도록 Amazon 웹 서비스 설정을 구성합니다.
   a. 구성 모드 'AWS configure'로 진입한다
   b. 생성한 계정의 AWS 액세스 키 ID를 입력합니다.
   c. 생성한 계정의 AWS Secret Access 키를 입력합니다.
   d. 사용할 기본 영역을 입력합니다(예: us-east-1).
   e. 사용할 기본 출력 형식을 입력하거나 * Enter * 를 눌러 JSON을 선택합니다.
2. 버킷을 만듭니다.

```
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"
```

1. 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.

2. 버킷의 내용을 나열하여 객체가 업로드되었는지 확인합니다.

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

3. 개체를 삭제합니다.

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

4. 버킷을 삭제합니다.

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

StorageGRID에서 **S3 REST API**를 구현하는 방법

클라이언트 애플리케이션은 S3 REST API 호출을 사용하여 StorageGRID에 연결하여 버킷을 생성, 삭제 및 수정할 수 있을 뿐만 아니라 오브젝트를 저장 및 검색할 수 있습니다.

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 제어

일관성 제어는 애플리케이션의 요구에 따라 오브젝트의 가용성과 서로 다른 스토리지 노드 및 사이트 전체에서 오브젝트의 일관성 간의 균형을 제공합니다.

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.

오브젝트 작업을 다른 정합성 보장 레벨에서 수행하려는 경우 각 버킷 또는 각 API 작업에 대해 정합성 제어를 지정할 수 있습니다.

일관성 제어

정합성 보장 제어는 StorageGRID에서 객체를 추적하는 데 사용하는 메타데이터가 노드 간에 분산되므로 클라이언트 요청에 대한 객체의 가용성에 영향을 줍니다.

버킷 또는 API 작업에 대한 정합성 제어를 다음 값 중 하나로 설정할 수 있습니다.

- * ALL *: 모든 노드가 즉시 데이터를 수신하거나 요청이 실패합니다.
- * strong-global *: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
- * strong-site *: 사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
- * read-after-new-write *: (기본값)는 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 객체 및 객체 업데이트 모두에 대한 최종 일관성을 제공합니다. S3 버킷의 경우 필요에 따라 만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.

"새 쓰기 후"와 "사용 가능한" 일관성 제어 기능을 사용합니다

HEAD 또는 GET 연산에서 "read-after-new-write" 정합성 제어를 사용하면 StorageGRID는 다음과 같이 여러 단계로 조회를 수행합니다.

- 먼저 낮은 일관성을 사용하여 오브젝트를 찾습니다.
- 이 조회에 실패하면 강력한 글로벌 동작에 해당하는 일관성 수준에 도달할 때까지 조회가 다음 일관성 수준에서 반복됩니다.

헤드 또는 GET 연산에서 "READ-After-NEW-WRITE" 일관성 컨트롤을 사용하지만 개체가 없으면 개체 조회는 항상 강력한 글로벌 동작의 일관성 수준에 도달합니다. 이 정합성 보장 수준에서는 각 사이트에서 개체 메타데이터의 여러 복사본을 사용할 수 있어야 하므로 같은 사이트에 있는 둘 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 많이 발생할 수 있습니다.

Amazon S3와 유사한 일관성 보증이 필요하지 않으면 일관성 제어를 ""사용 가능""으로 설정하여 헤드 및 GET 작업에 대한 이러한 오류를 방지할 수 있습니다. 헤드 또는 GET 작업에서 "사용 가능한" 정합성 제어를 사용할 경우 StorageGRID는 최종 일관성만 제공합니다. 일관성 수준을 높일 때 실패한 작업을 다시 시도하지 않으므로 오브젝트 메타데이터의 여러 복사본을 사용할 필요가 없습니다.

API 작업에 대한 정합성 제어를 지정합니다

개별 API 작업의 정합성 제어를 설정하려면 작업에 대해 정합성 보장 제어가 지원되어야 하며 요청 헤더에 정합성 제어를 지정해야 합니다. 이 예제에서는 개체 가져오기 작업을 위해 일관성 컨트롤을 "문자열 사이트"로 설정합니다.

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

> 개체 넣기 작업과 개체 가져오기 작업 모두에 대해 동일한 일관성 컨트롤을 사용해야 합니다.

버킷의 정합성 제어를 지정합니다

버킷의 일관성 제어를 설정하려면 StorageGRID PUT 버킷 정합성 보장 요청 및 GET 버킷 정합성 보장 요청을 사용할 수 있습니다. 또는 테넌트 관리자 또는 테넌트 관리 API를 사용할 수 있습니다.

버킷의 정합성 제어 기능을 설정할 때는 다음 사항에 유의하십시오.

- 버킷의 일관성 제어를 설정하면 버킷의 오브젝트 또는 버킷 구성에 대해 수행된 S3 작업에 사용되는 일관성 제어가 결정됩니다. 버킷 자체의 작동에는 영향을 미치지 않습니다.
- 개별 API 작업의 정합성 제어는 버킷의 정합성 제어를 재정의합니다.
- 일반적으로 버킷은 기본 일관성 제어인 "read-after-new-write"를 사용해야 합니다. 요청이 올바르게 작동하지 않는 경우 가능한 경우 응용 프로그램 클라이언트 동작을 변경합니다. 또는 클라이언트가 각 API 요청에 대한 정합성 제어를 지정하도록 구성합니다. 버킷 레벨에서만 정합성 제어를 최후의 수단으로 설정하십시오.

일관성 제어 및 **ILM** 규칙이 상호 작용하여 데이터 보호에 영향을 미치는 방식

일관성 제어와 ILM 규칙 모두 오브젝트의 보호 방법에 영향을 미칩니다. 이러한 설정은 상호 작용할 수 있습니다.

예를 들어, 개체가 저장될 때 사용되는 일관성 컨트롤은 오브젝트 메타데이터의 초기 배치에 영향을 미치는 반면 ILM 규칙에 대해 선택된 수집 동작은 오브젝트 복사본의 초기 배치에 영향을 줍니다. StorageGRID에서는 클라이언트 요청을 이행하기 위해 오브젝트의 메타데이터와 해당 데이터에 모두 액세스해야 하므로 일관성 수준과 수집 동작에 적합한 보호 수준을 선택하면 초기 데이터 보호 수준을 높이고 시스템 응답을 더욱 정확하게 예측할 수 있습니다.

ILM 규칙에 대해 다음과 같은 수집 동작을 사용할 수 있습니다.

- * Strict * : ILM 규칙에 지정된 모든 사본은 클라이언트에 반환되기 전에 만들어야 합니다.
- * 균형 *: StorageGRID는 수집 시 ILM 규칙에 지정된 모든 복제본을 생성하려고 합니다. 그렇지 않을 경우 중간 복사본이 만들어지고 클라이언트에 성공적으로 반환됩니다. ILM 규칙에 지정된 복사본은 가능한 경우 만들어집니다.
- * 이중 커밋*: StorageGRID는 즉시 개체의 임시 복사본을 만들고 클라이언트에 성공을 반환합니다. ILM 규칙에 지정된 복사본은 가능한 경우 만들어집니다.

> ILM 규칙에 대한 수집 동작을 선택하기 전에 에서 이러한 설정에 대한 전체 설명을 읽어보십시오 ILM을 사용하여 개체를 관리합니다.

일관성 제어 및 **ILM** 규칙이 상호 작용하는 방법의 예

다음 ILM 규칙 및 다음 일관성 수준 설정이 있는 두 사이트 그리드가 있다고 가정합니다.

- * ILM 규칙 *: 로컬 사이트와 원격 사이트에 각각 하나씩, 두 개의 오브젝트 복사본을 만듭니다. Strict 수집 동작이 선택됩니다.
- * Consistency level *: "trong-global"(개체 메타데이터가 모든 사이트에 즉시 배포됩니다.)

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 규칙 및 "'strong-site' 정합성 보장 수준을 사용한 경우 객체 데이터가 원격 사이트에 복제되었지만 객체 메타데이터가 그 위치에 배포되기 전에 클라이언트에 성공 메시지가 표시될 수 있습니다. 이 경우 오브젝트 메타데이터의 보호 수준이 오브젝트 데이터의 보호 수준과 일치하지 않습니다. 수집 후 곧바로 로컬 사이트가 손실되면 오브젝트 메타데이터가 손실됩니다. 객체를 검색할 수 없습니다.

일관성 수준과 ILM 규칙 간의 상호 관계는 복잡할 수 있습니다. 도움이 필요한 경우 NetApp에 문의하십시오.

관련 정보

버킷 정합성 보장 요청 가져오기

버킷 정합성 보장 요청을 배치합니다

StorageGRID ILM 규칙이 개체를 관리하는 방법

그리드 관리자는 정보 라이프사이클 관리(ILM) 규칙을 생성하여 S3 REST API 클라이언트 애플리케이션에서 StorageGRID 시스템으로 수집된 오브젝트 데이터를 관리합니다. 그런 다음 이러한 규칙을 ILM 정책에 추가하여 시간 경과에 따라 오브젝트 데이터가 저장되는 방법 및 위치를 결정합니다.

ILM 설정은 개체의 다음 측면을 결정합니다.

- * 지역 *

  StorageGRID 시스템(스토리지 풀) 또는 클라우드 스토리지 풀 내에서 오브젝트 데이터의 위치입니다.

- * 스토리지 등급 *

  오브젝트 데이터를 저장하는 데 사용되는 스토리지의 유형(예: 플래시 또는 회전식 디스크)

- * 손실 방지 *

  복제, 삭제 코딩 또는 두 가지 유형의 복사본을 만들 수와 복사본 유형을 지정합니다.

- * 보존 *

  오브젝트의 데이터 관리 방식, 저장 위치 및 데이터 손실을 보호하는 방법에 대한 시간이 지나면서 변동합니다.

- * 수집 중 보호 *

  수집 중에 오브젝트 데이터를 보호하는 데 사용되는 방법: 동기 배치(Ingest 동작에 대한 균형 또는 엄격 옵션 사용) 또는 중간 복사본 만들기(이중 커밋 옵션 사용).

ILM 규칙을 사용하여 개체를 필터링 및 선택할 수 있습니다. S3을 사용하여 수집된 개체의 경우 ILM 규칙을 통해 다음 메타데이터를 기반으로 개체를 필터링할 수 있습니다.

- 테넌트 계정
- 버킷 이름

- 수집 시간
- 키
- 마지막 액세스 시간입니다

> 기본적으로 마지막 액세스 시간에 대한 업데이트는 모든 S3 버킷에 대해 비활성화됩니다. StorageGRID 시스템에 마지막 액세스 시간 옵션을 사용하는 ILM 규칙이 포함된 경우 해당 규칙에 지정된 S3 버킷의 마지막 액세스 시간에 대한 업데이트를 활성화해야 합니다. Tenant Manager의 [버킷 최종 액세스 시간] 요청, [S3 * > * Bucket * > * [마지막 액세스 시간 구성] * 확인란을 사용하거나 Tenant Management API를 사용하여 마지막 액세스 시간 업데이트를 활성화할 수 있습니다. 마지막 액세스 시간 업데이트를 설정할 때는 특히 작은 오브젝트가 있는 시스템에서 StorageGRID 성능이 저하될 수 있다는 점에 유의하십시오.

- 위치 제약 조건
- 개체 크기
- 사용자 메타데이터
- 개체 태그

ILM에 대한 자세한 내용은 정보 수명 주기 관리를 통해 개체 관리 지침을 참조하십시오.

관련 정보

테넌트 계정을 사용합니다

ILM을 사용하여 개체를 관리합니다

버킷 최종 액세스 시간 요청

오브젝트 버전 관리

버전 관리를 사용하면 개체의 여러 버전을 유지하여 실수에 의한 개체 삭제로부터 보호하고 이전 버전의 개체를 검색하고 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. 버킷에 대해 이 기능을 설정하려면 각 버킷에 대해 버전 관리를 명시적으로 활성화해야 합니다. 버킷의 각 오브젝트에는 StorageGRID 시스템에서 생성되는 버전 ID가 할당됩니다.

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 관리가 활성화된 버킷의 S3 오브젝트에서 버전 관리를 지원하므로 비현재 시간을 참조 시간으로 사용하는 ILM 규칙을 생성할 수 있습니다. 개체가 업데이트되면 이전 버전은 업데이트되지 않습니다. 비현재 시간 필터를 사용하면 이전 버전의 오브젝트에 대한 스토리지 영향을 줄이는 정책을 생성할 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.

S3 버전 오브젝트에 대한 ILM 정책 예제를 보려면 정보 수명 주기 관리로 오브젝트를 관리하는 지침을 참조하십시오.

관련 정보

ILM을 사용하여 개체를 관리합니다

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 존재하지 않을 것으로 예상되는 경로에 개체가 있는지 정기적으로 확인하는 경우 ""사용 가능한"" 일관성 제어를 사용해야 합니다. 예를 들어, 응용 프로그램이 해당 위치에 배치되기 전에 위치를 지정할 경우 ""사용 가능"" 정합성 제어를 사용해야 합니다.

그렇지 않으면 헤드 작업에서 개체를 찾지 못할 경우 하나 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다.

PUT Bucket 정합성 보장 요청을 사용하여 각 버킷에 대해 ""사용 가능" 정합성 제어를 설정하거나 개별 API 작업에 대한 요청 헤더에서 정합성 제어를 지정할 수 있습니다.

개체 키에 대한 권장 사항

StorageGRID 11.4 이상에서 생성된 버킷의 경우 성능 모범 사례에 맞게 오브젝트 키 이름을 제한하는 것은 더 이상 필요하지 않습니다. 예를 들어, 이제 개체 키 이름의 처음 4개 문자에 임의의 값을 사용할 수 있습니다.

StorageGRID 11.4 이전 릴리스에서 생성된 버킷의 경우 객체 키 이름에 대한 다음 권장 사항을 계속 따르십시오.

- 개체 키의 처음 네 문자로 임의 값을 사용하면 안 됩니다. 이는 이전 AWS에서 권장하는 키 접두사와 다릅니다. 대신 "이미지"와 같은 비무작위, 고유하지 않은 접두사를 사용해야 합니다.
- 이전 AWS 권장 사항에 따라 키 접두사에 임의의 고유 문자를 사용하려면 객체 키에 디렉토리 이름을 접두사로 지정해야 합니다. 즉, 다음 형식을 사용합니다.

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

  이 형식 대신:

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

저장된 오브젝트 압축 * 옵션을 선택한 경우(* 구성 * > * 시스템 * > * 그리드 옵션 *) S3 클라이언트 응용 프로그램은 바이트 범위를 지정하는 오브젝트 가져오기 작업을 수행하지 않아야 합니다. 이러한 ""범위 읽기"" 작업은 StorageGRID가 요청된 바이트에 액세스하기 위해 개체의 압축을 효과적으로 해제해야 하기 때문에 비효율적입니다. 매우 큰 개체에서 작은 바이트 범위를 요청하는 Get Object 작업은 특히 비효율적입니다. 예를 들어, 50GB의 압축 개체에서 10MB 범위를 읽는 것은 매우 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.

관련 정보

- 일관성 제어
- 버킷 정합성 보장 요청을 배치합니다
- StorageGRID 관리

S3 REST API에서 지원되는 작업 및 제한 사항

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). 요청에 "x-amz-date" 헤더를 포함하면 날짜 요청 헤더에 지정된 모든 값이 무시됩니다. AWS 서명 버전 4를 사용할 때는 날짜 헤더가 지원되지 않으므로 서명된 요청에 "x-amz-date" 헤더가 있어야 합니다.

공통 요청 헤더

StorageGRID 시스템은 에 의해 정의된 공통 요청 헤더를 지원합니다 "AWS(Amazon Web Services) 문서: Amazon Simple Storage Service API Reference 를 참조하십시오"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 X-amz-content-SHA256 헤더용으로 'unsigned-payload' 값이 제공된 것처럼 유효성 검사 없이 허용됩니다.
X-amz-security-token	구현되지 않았습니다. 'XNotImplemented'를 반환합니다.

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 HTTP '권한 부여' 헤더와 쿼리 매개 변수 사용 등 두 가지 인증 방법을 지원합니다.

HTTP 인증 헤더를 사용합니다

HTTP '권한 부여' 헤더는 버킷 정책에서 허용하는 익명 요청을 제외한 모든 S3 API 작업에서 사용됩니다. '권한 부여' 헤더에는 요청을 인증하는 데 필요한 모든 서명 정보가 들어 있습니다.

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 없습니다. 이 키를 사용하면 타사에 리소스에 대한 제한된 액세스를 제공할 수 있습니다.

서비스에 대한 작업

StorageGRID 시스템은 서비스에 대해 다음 작업을 지원합니다.

작동	구축
서비스 받기	모든 Amazon S3 REST API 동작으로 구현됩니다.
스토리지 사용량을 가져옵니다	Get Storage Usage 요청은 계정이 사용하는 총 스토리지 양과 계정과 연결된 각 버킷에 대해 알려줍니다. 이 작업은 경로에 /을 추가하고 사용자 지정 쿼리 매개 변수("?x-ntap-sg-usage")를 추가하는 서비스에 대한 작업입니다.
옵션 /	클라이언트 애플리케이션은 스토리지 노드의 S3 인증 자격 증명을 제공하지 않고 스토리지 노드의 S3 포트에 대한 "옵션/" 요청을 발행하여 스토리지 노드의 사용 가능 여부를 결정할 수 있습니다. 이 요청을 사용하여 모니터링을 수행하거나, 외부 로드 밸런서가 스토리지 노드가 다운된 시점을 식별하도록 할 수 있습니다.

관련 정보

스토리지 사용 요청 가져오기

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 지원하려면 이러한 제한을 DNS 명명 규칙으로 제한해야 합니다.

"AWS(Amazon Web Services) 문서: 버킷 제한 및 제한 사항"

S3 API 엔드포인트 도메인 이름을 구성합니다

버킷 가져오기(개체 나열) 및 버킷 버전 가져오기 작업은 StorageGRID 정합성 보장 제어를 지원합니다.

개별 버킷에 대해 마지막 액세스 시간에 대한 업데이트가 설정되었는지 여부를 확인할 수 있습니다.

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

작동	구축
버킷 삭제	모든 Amazon S3 REST API 동작으로 구현됩니다.
버킷 CORS를 삭제합니다	이 작업은 버킷에 대한 CORS 구성을 삭제합니다.
Bucket 암호화를 삭제합니다	이 작업은 버킷에서 기본 암호화를 삭제합니다. 암호화된 기존 개체는 암호화된 상태로 유지되지만 버킷에 추가된 새 개체는 암호화되지 않습니다.
버킷 수명 주기를 삭제합니다	이 작업은 버킷에서 라이프사이클 구성을 삭제합니다.

작동	구축
버킷 정책을 삭제합니다	이 작업은 버킷에 연결된 정책을 삭제합니다.
버킷 복제를 삭제합니다	이 작업은 버킷에 연결된 복제 구성을 삭제합니다.
버킷 태그 지정을 삭제합니다	이 작업은 "태그 지정" 하위 리소스를 사용하여 버킷에서 모든 태그를 제거합니다.
버킷(목록 오브젝트), 버전 1 및 버전 2를 가져옵니다	이 작업은 버킷에 있는 오브젝트의 일부 또는 전체(최대 1,000개)를 반환합니다. 객체에 대한 스토리지 클래스는 객체가 REDucted_redundancy 스토리지 클래스 옵션으로 인제스트된 경우에도 두 값 중 하나를 가질 수 있습니다. • 객체가 스토리지 노드로 구성된 스토리지 풀에 저장되었음을 나타내는 'Standard'입니다. • Glacier는 해당 객체가 Cloud Storage Pool에서 지정한 외부 버킷으로 이동되었음을 나타냅니다. 버킷에 동일한 접두사가 있는 삭제된 키의 수가 많은 경우, 응답에는 키가 포함되지 않은 몇 가지 CommonPrefixes가 포함될 수 있습니다.
버킷 ACL 가져오기	이 작업은 양수 응답 및 버킷 소유자의 ID, DisplayName 및 권한을 반환하며, 이는 소유자가 버킷에 대한 모든 액세스 권한을 가지고 있음을 나타냅니다.
버킷 CORS를 가져옵니다	이 작업은 버킷에 대한 'CORS' 구성을 반환합니다.
버킷 암호화 가져오기	이 작업은 버킷의 기본 암호화 구성을 반환합니다.
버킷 수명 주기 가져오기	이 작업은 버킷의 수명 주기 구성을 반환합니다.
버킷 위치를 가져옵니다	이 작업은 Put Bucket 요청에서 "LocationConstraint" 요소를 사용하여 설정된 영역을 반환합니다. 버켓 지역이 us-east-1인 경우 해당 지역에 대해 빈 문자열이 반환됩니다.
버킷 알림을 받습니다	이 작업은 버킷에 연결된 알림 구성을 반환합니다.
버킷 객체 버전을 가져옵니다	버킷에 대한 읽기 액세스 기능을 사용하면 하위 리소스의 이 작업에서는 버킷에 있는 모든 개체 버전의 메타데이터를 나열합니다.
버킷 정책 가져오기	이 작업은 버킷에 연결된 정책을 반환합니다.
버킷 복제를 가져옵니다	이 작업은 버킷에 연결된 복제 구성을 반환합니다.
버킷 태그 지정을 가져옵니다	이 작업은 "태그 지정" 하위 리소스를 사용하여 버킷에 대한 모든 태그를 반환합니다.

작동	구축
버킷 버전 관리 가져오기	이를 위해 장난 서브리소스를 사용해 버켓의 버전 상태를 반환한다. • *blank*: 버전 관리가 활성화된 적이 없습니다(버킷을 ""버전 없음""). • 사용: 버전 관리가 활성화됩니다 • 일시 중단됨: 버전 관리가 이전에 활성화되었으며 일시 중단되었습니다
개체 잠금 구성을 가져옵니다	이 작업은 버킷 기본 보존 모드 및 기본 보존 기간(구성된 경우)을 반환합니다. 을 참조하십시오 개체 잠금 구성을 가져옵니다 을 참조하십시오.
헤드 버킷	이 작업은 버킷이 있는지 그리고 버킷에 액세스할 권한이 있는지 여부를 결정합니다. 이 작업은 다음을 반환합니다. • X-NTAP-sg-bucket-id: UUID 형식의 버킷 UUID • X-NTAP-sg-trace-id: 연관된 요청의 고유 추적 ID.

작동	구축
버킷 을 놓습니다	이 작업은 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 us-east-1 영역에 생성되지만 요청 본문의 LocationConstraint 요청 요소를 사용하여 다른 영역을 지정할 수 있습니다. "LocationConstraint" 요소를 사용할 때는 Grid Manager 또는 Grid Management API를 사용하여 정의된 영역의 정확한 이름을 지정해야 합니다. 사용할 지역 이름을 모르는 경우 시스템 관리자에게 문의하십시오. • 참고 *: PUT 버킷 요청이 StorageGRID에 정의되지 않은 지역을 사용하는 경우 오류가 발생합니다. • 'x-amz-bucket-object-lock-enabled' 요청 헤더를 포함시켜 S3 오브젝트 잠금이 활성화된 버킷을 생성할 수 있습니다. 을 참조하십시오 S3 오브젝트 잠금을 사용합니다. 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.
버킷 CORS를 넣습니다	이 작업은 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, 그래픽을 저장하기 위해 "images"라는 S3 버킷을 사용한다고 가정합니다. 'images' bucket에 대한 CORS 설정을 통해 해당 bucket의 이미지를 웹사이트(+ http://www.example.com+`) 에 표시할 수 있습니다.
Bucket 암호화를 적용합니다	이 작업은 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. 서버쪽 암호화 설정 규칙을 지정할 때 SEAlgorithm 매개변수를 AES256으로 설정하고 KMSMasterKeyID 매개변수를 사용하지 마십시오. 객체 업로드 요청이 이미 암호화를 지정한 경우(즉, 요청에 "x-amz-server-side-encryption- *" 요청 헤더가 포함된 경우) 버킷 기본 암호화 구성이 무시됩니다.

작동	구축
버킷 수명 주기를 놓습니다	이 작업은 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) • NoncurrentVersionExpiration(NoncurrentDays) • 필터(접두사, 태그) • 상태 • ID입니다 StorageGRID는 다음 작업을 지원하지 않습니다. • AbortIncompleteMultipartUpload를 중단합니다 • ExpiredObjectDeleteMarker 를 참조하십시오 • 전환 버킷 수명 주기의 만료 작업이 ILM 배치 명령과 상호 작용하는 방법을 이해하려면 정보 수명 주기 관리를 통해 개체를 관리하기 위한 지침에서 ""ILM이 개체의 수명 내내 작동하는 방식""을 참조하십시오. • 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
버킷 통지를 보냅니다	이 작업은 요청 본문에 포함된 알림 구성 XML을 사용하여 버킷에 대한 알림을 구성합니다. 다음과 같은 구현 세부 사항에 유의해야 합니다. • StorageGRID는 SNS(Simple Notification Service) 항목을 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 없으면 400개의 잘못된 요청 오류가 InvalidArgument 코드와 함께 반환됩니다. • 다음 이벤트 유형에 대한 알림을 구성할 수 없습니다. 이러한 이벤트 유형은 * 지원되지 않습니다 *. ◦ '3: RedundancyLostObject'를 선택합니다 ◦ '3:ObjectRestore:완료됨' • StorageGRID에서 보낸 이벤트 알림은 다음 목록과 같이 일부 키를 포함하지 않고 다른 키에 대해 특정 값을 사용한다는 점을 제외하고 표준 JSON 형식을 사용합니다. • * eventSource * 를 선택합니다 전쟁포로 S3 • * awsRegion * 포함되지 않음 • x-amz-id-2 * 포함되지 않음 • * 표시 * "urn:SGWs:S3::bucket_name'
버킷 정책을 적용합니다	이 작업은 버킷에 연결된 정책을 설정합니다.

작동	구축

버킷 복제를 배치합니다	이 작업은 요청 본문에 제공된 복제 구성 XML을 사용하여 버킷에 대한 StorageGRID CloudMirror 복제를 구성합니다. CloudMirror 복제의 경우 다음과 같은 구축 세부 정보를 알고 있어야 합니다.
	
	• StorageGRID는 복제 구성의 V1만 지원합니다. 즉, StorageGRID는 규칙에 대해 'Filter' 요소의 사용을 지원하지 않으며 개체 버전 삭제에 대해서는 V1 규약을 따릅니다. 자세한 내용은 를 참조하십시오 "복제 구성에 대한 Amazon S3 문서".
	• 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.
	
	복제 구성이 성공하려면 엔드포인트가 있어야 합니다. 종점이 존재하지 않으면 400개의 불량 요청으로 실패한다. "복제 정책을 저장할 수 없습니다. 지정한 끝점 URN이 없습니다:_URN_.
	
	• 구성 XML에서 역할 을 지정할 필요는 없습니다. 이 값은 StorageGRID에서 사용되지 않으며 제출될 경우 무시됩니다.
	• 구성 XML에서 스토리지 클래스를 생략하면 StorageGRID에서는 기본적으로 '표준' 스토리지 클래스를 사용합니다.
	• 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 통보되지 않습니다.
	  ◦ 복제된 후 오브젝트 또는 버킷을 삭제하면 StorageGRID는 지역 간 복제 V1에 대한 표준 Amazon S3 삭제 동작을 따릅니다.

Bucket 태그 달기	이 작업은 "태그 지정" 하위 리소스를 사용하여 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.
버킷 버전 관리	이 구현은 재세팅된 서브리소스를 사용하여 기존 버킷의 버전 관리 상태를 설정합니다. 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 ID "null"을 받습니다.

작동	구축
개체 잠금 구성을 배치합니다	이 작업은 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 개체 잠금 구성을 배치합니다 을 참조하십시오.

관련 정보

일관성 제어

버킷 최종 액세스 시간 요청 가져오기

버킷 및 그룹 액세스 정책

S3 작업이 감사 로그에서 추적되었습니다

ILM을 사용하여 개체를 관리합니다

테넌트 계정을 사용합니다

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 개발자 가이드: 개체 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

버킷에 라이프사이클 구성을 적용하는 경우 버킷의 라이프사이클 설정은 항상 StorageGRID ILM 설정을 재정의합니다. StorageGRID는 ILM이 아닌 버킷의 만료 설정을 사용하여 특정 개체의 삭제 또는 유지 여부를 결정합니다.

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.

- 버킷 수명 주기를 삭제합니다
- 버킷 수명 주기 가져오기
- 버킷 수명 주기를 놓습니다

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 접두사 categy1과 일치하고 key2 값이 tag2 인 객체에만 적용됩니다. Expiration 매개변수는 필터와 일치하는 객체가 2020년 8월 22일 자정에 만료되도록 지정합니다.
2. 규칙 2는 접두사 'category2'/ 와 일치하는 객체에만 적용됩니다. Expiration 매개 변수는 필터와 일치하는 객체가 수집되고 100일 후에 만료되도록 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 접두사 'category3'/ 와 일치하는 객체에만 적용됩니다. Expiration 매개 변수는 일치하는 객체의 비최신 버전이 비최신 상태가 된 후 50일 후에 만료되도록 지정합니다.

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

버킷에 라이프사이클 구성을 적용합니다

문서 수정 상태 구성 파일을 작성한 후 PUT Bucket 수명주기 요청을 실행하여 이를 버킷에 적용합니다.

이 요청은 예제 파일의 수명 주기 구성을 'testbucket'이라는 이름의 버킷의 개체에 적용합니다.

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

라이프사이클 구성이 버킷에 성공적으로 적용되었는지 확인하려면 Get Bucket 수명주기 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

Put Object, Head Object 또는 Get Object 요청을 실행할 때 수명 주기 구성의 만료 규칙이 특정 개체에 적용되는지 확인할 수 있습니다. 규칙이 적용될 경우 응답에는 개체 만료 시기 및 일치하는 만료 규칙을 나타내는 Expiration 매개변수가 포함됩니다.

> 버킷 라이프사이클이 ILM을 무시하기 때문에 표시된 '만료 날짜'는 객체가 삭제될 실제 날짜입니다. 자세한 내용은 을 참조하십시오 개체 보존이 결정되는 방식.

예를 들어, 이 PUT 오브젝트 요청은 2020년 6월 22일에 발행되었으며 'testbucket' 버킷에 오브젝트를 배치했습니다.

```
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 head Object 요청은 testbucket 버킷에서 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": {}
}
```

S3 오브젝트 잠금 기본 버킷 보존을 사용합니다

버킷에 S3 오브젝트 잠금이 활성화된 경우 버킷에 추가된 각 오브젝트에 적용되는 기본 보존 모드와 기본 보존 기간을 지정할 수 있습니다.

- 버킷 생성 중에 버킷에 대해 S3 오브젝트 잠금을 설정하거나 해제할 수 있습니다.
- 버킷에 대해 S3 오브젝트 잠금이 활성화된 경우 버킷의 기본 보존을 구성할 수 있습니다.
- 기본 보존 구성은 다음을 지정합니다.
  - 기본 보존 모드: StorageGRID는 "규정 준수" 모드만 지원합니다.
  - 기본 보존 기간(일 또는 년).

개체 잠금 구성을 가져옵니다

객체 잠금 구성 가져오기 요청을 사용하면 버킷에 대해 객체 잠금이 설정되어 있는지 확인하고, 활성화된 경우 버킷에 대해 기본 보존 모드 및 보존 기간이 구성되어 있는지 확인할 수 있습니다.

새로운 오브젝트 버전이 버킷에 수집되면 "x-amz-object-lock-mode"가 지정되지 않은 경우 기본 보존 모드가 적용됩니다. x-amz-object-lock-retain-until-date를 지정하지 않으면 기본 보존 기간을 사용하여 Retain-until-date를 계산합니다.

이 작업을 완료하려면 S3:GetBucketObjectLockConfiguration 권한이 있거나 계정 루트여야 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>
```

개체 잠금 구성을 배치합니다

객체 잠금 구성 요청을 사용하면 객체 잠금이 활성화된 버킷의 기본 보존 모드 및 기본 보존 기간을 수정할 수 있습니다. 이전에 구성한 기본 보존 설정을 제거할 수도 있습니다.

새로운 오브젝트 버전이 버킷에 수집되면 "x-amz-object-lock-mode"가 지정되지 않은 경우 기본 보존 모드가 적용됩니다. x-amz-object-lock-retain-until-date를 지정하지 않으면 기본 보존 기간을 사용하여 Retain-until-date를 계산합니다.

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

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

PUT 요청에는 Content-MD5 요청 헤더를 지정해야 합니다.

요청 예

```
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>
```

버킷에 대한 사용자 지정 작업

StorageGRID 시스템은 S3 REST API에 추가되고 시스템에 고유한 맞춤형 버킷 작업을 지원합니다.

다음 표에는 StorageGRID에서 지원하는 사용자 지정 버킷 작업이 나열되어 있습니다.

작동	설명	를 참조하십시오
버킷 일관성 확보	특정 버킷에 적용되는 정합성 보장 레벨을 반환합니다.	버킷 정합성 보장 요청 가져오기
버킷 일관성을 유지합니다	특정 버킷에 적용되는 정합성 수준을 설정합니다.	버킷 정합성 보장 요청을 배치합니다
버킷 최종 액세스 시간 가져오기	특정 버킷에 대해 마지막 액세스 시간 업데이트를 사용할 수 있는지 여부를 반환합니다.	버킷 최종 액세스 시간 요청 가져오기
버킷 최종 접근 시간	특정 버킷에 대한 마지막 액세스 시간 업데이트를 활성화하거나 비활성화할 수 있습니다.	버킷 최종 액세스 시간 요청

작동	설명	를 참조하십시오
버킷 메타데이터 알림 구성을 삭제합니다	특정 버킷과 연결된 메타데이터 알림 구성 XML을 삭제합니다.	버킷 메타데이터 알림 구성 요청을 삭제합니다
Bucket 메타데이터 알림 구성 가져오기	특정 버킷과 연결된 메타데이터 알림 구성 XML을 반환합니다.	버킷 메타데이터 알림 구성 요청을 가져옵니다
Put Bucket 메타데이터 알림 구성	버킷에 대한 메타데이터 알림 서비스를 구성합니다.	PUT 버킷 메타데이터 알림 구성 요청
준수 설정이 적용된 버킷을 배치합니다	더 이상 사용되지 않으며 지원되지 않음: Compliance를 사용하는 새 버킷을 더 이상 생성할 수 없습니다.	사용되지 않음: 준수 설정이 포함된 Bucket을 넣습니다
버킷 규정 준수	더 이상 사용되지 않지만 지원됨: 기존 레거시 준수 버킷에 대해 현재 적용되는 규정 준수 설정을 반환합니다.	사용되지 않음: 버킷 준수 요청 가져오기
버킷 규정 준수	사용되지 않지만 지원됨: 기존 레거시 준수 버킷의 준수 설정을 수정할 수 있습니다.	폐기됨: 버킷 준수 요청을 넣으십시오

관련 정보

감사 로그에서 S3 작업을 추적했습니다

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 제어 는 다음과 같은 경우를 제외하고 모든 개체 작업에서 지원됩니다.
  - 객체 ACL을 가져옵니다
  - '옵션 /'
  - 개체를 법적 증거 자료 보관
  - 개체 보존
  - 개체 내용 을 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
개체 삭제	멀티팩터 인증(MFA)과 응답 헤더 X-amz-MFA는 지원되지 않습니다. 오브젝트 삭제 요청을 처리할 때 StorageGRID는 저장된 모든 위치에서 오브젝트의 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 rrionId 하위 리소스를 사용해야 합니다. 이 하위 리소스를 사용하면 버전이 영구적으로 삭제됩니다. 만약 rionId가 삭제 표식에 해당하면 응답 헤더 x-amz-delete-marker가 TRUE로 설정된다. • 버전 지원 버킷에서 rionId 하위 리소스 없이 개체를 삭제하면 삭제 표식이 생성됩니다. 삭제 마커의 rionId는 x-amz-version-id 응답 헤더를 사용하여 반환되고 x-amz-delete-marker 응답 헤더는 TRUE로 설정됩니다. • 버전 일시 중지된 버킷에서 rionId ' 하위 리소스 없이 개체를 삭제하면 기존 'null' 버전 또는 'null' 삭제 표식이 영구적으로 삭제되고 새 'null' 삭제 표식이 생성됩니다. x-amz-DELETE-MARKER 응답 헤더가 TRUE로 설정된 상태로 반환됩니다. • 참고 *: 경우에 따라 객체에 대해 여러 개의 삭제 마커가 존재할 수 있습니다.
여러 개체를 삭제합니다	멀티팩터 인증(MFA)과 응답 헤더 X-amz-MFA는 지원되지 않습니다. 동일한 요청 메시지에서 여러 객체를 삭제할 수 있습니다.
개체 태그 지정 삭제	"태그 지정" 하위 리소스를 사용하여 개체에서 모든 태그를 제거합니다. 모든 Amazon S3 REST API 동작으로 구현됩니다. • 버전 관리 * 요청에 rionId 쿼리 매개 변수가 지정되지 않은 경우 버전 지정된 버킷에 있는 개체의 최신 버전에서 모든 태그가 삭제됩니다. 객체의 현재 버전이 삭제 표식이면 x-amz-delete-marker 응답 헤더가 true로 설정된 상태로 "MethodNotAllowed" 상태가 반환됩니다.
객체 가져오기	객체 가져오기

작동	구축
객체 ACL을 가져옵니다	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.
객체 법적 증거 자료 보관	S3 오브젝트 잠금을 사용합니다
개체 보존 가져오기	S3 오브젝트 잠금을 사용합니다
객체 태그 지정 가져오기	"태그 지정" 하위 리소스를 사용하여 개체에 대한 모든 태그를 반환합니다. 모든 Amazon S3 REST API 동작으로 구현됩니다 • 버전 관리 * 요청에 rionId 쿼리 매개 변수가 지정되지 않은 경우 이 작업은 최신 버전의 개체에서 버전 관리되는 버킷으로 모든 태그를 반환합니다. 객체의 현재 버전이 삭제 표식이면 x-amz-delete-marker 응답 헤더가 true로 설정된 상태로 "MethodNotAllowed" 상태가 반환됩니다.
헤드 개체	헤드 개체
사후 개체 복원	사후 개체 복원
개체 를 넣습니다	개체 를 넣습니다
개체 - 복사 를 선택합니다	개체 - 복사 를 선택합니다
개체를 법적 증거 자료 보관	S3 오브젝트 잠금을 사용합니다
개체 보존	S3 오브젝트 잠금을 사용합니다

작동	구축
개체 태그 지정	태그 지정 하위 리소스를 사용하여 기존 개체에 태그 집합을 추가합니다. 모든 Amazon S3 REST API 동작으로 구현됩니다 • 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 * 오브젝트 태그 지정을 사용하여 개체의 태그를 업데이트하는 경우 StorageGRID에서는 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 * 요청에 rionId 쿼리 매개 변수가 지정되지 않은 경우 작업에서 버전 관리되는 버킷의 가장 최신 버전의 개체에 태그를 추가합니다. 객체의 현재 버전이 삭제 표식이면 x-amz-delete-marker 응답 헤더가 true로 설정된 상태로 "MethodNotAllowed" 상태가 반환됩니다.

관련 정보

S3 작업이 감사 로그에서 추적되었습니다

S3 오브젝트 잠금을 사용합니다

StorageGRID 시스템에서 글로벌 S3 오브젝트 잠금 설정이 활성화된 경우 S3 오브젝트 잠금이 설정된 버킷을 생성한 다음 각 버킷의 기본 보존 기간 또는 해당 버킷에 추가한 각 오브젝트 버전에 대한 특정 보관 기간 및 법적 증거 보관 설정을 지정할 수 있습니다.

S3 오브젝트 잠금을 사용하면 고정된 시간 또는 무기한으로 오브젝트를 삭제 또는 덮어쓰는 것을 방지하기 위해 오브젝트 레벨 설정을 지정할 수 있습니다.

StorageGRID S3 오브젝트 잠금 기능은 Amazon S3 규정 준수 모드에 상응하는 단일 보존 모드를 제공합니다. 기본적으로 보호된 개체 버전은 사용자가 덮어쓰거나 삭제할 수 없습니다. StorageGRID S3 오브젝트 잠금 기능은 거버넌스 모드를 지원하지 않으며, 특별한 권한이 있는 사용자가 보존 설정을 무시하거나 보호된 오브젝트를 삭제할 수 없습니다.

버킷에 대해 **S3** 오브젝트 잠금을 설정합니다

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

- 테넌트 관리자를 사용하여 버킷을 생성합니다.

  테넌트 계정을 사용합니다

- X-amz-bucket-object-lock-enabled 요청 헤더로 PUT Bucket 요청을 이용하여 bucket을 생성한다.

  버킷 작업

버킷이 생성된 후에는 S3 오브젝트 잠금을 추가하거나 비활성화할 수 없습니다. S3 오브젝트 잠금에는 버킷 버전 관리가 필요하며, 이 버전은 버킷을 생성할 때 자동으로 활성화됩니다.

S3 오브젝트 잠금이 활성화된 버킷에는 S3 오브젝트 잠금 설정이 있는 오브젝트와 없는 오브젝트의 조합이 포함될 수 있습니다. StorageGRID는 S3 오브젝트 잠금 버킷의 오브젝트에 대한 기본 보존 기간을 지원하며 오브젝트 잠금 구성 버킷 작업을 지원합니다. '3:object-lock-remaining-days' 정책 조건 키는 객체에 대해 허용되는 최소 및 최대 보존 기간을 설정합니다.

버킷에 **S3** 오브젝트 잠금이 설정되었는지 확인

S3 오브젝트 잠금이 활성화되어 있는지 확인하려면 를 사용합니다 개체 잠금 구성을 가져옵니다 요청하십시오.

S3 오브젝트 잠금 설정으로 오브젝트를 생성합니다

S3 오브젝트 잠금이 활성화된 버킷에 오브젝트 버전을 추가할 때 S3 오브젝트 잠금 설정을 지정하려면 오브젝트 넣기, 오브젝트 복사 넣기 또는 다중 파트 업로드 요청을 시작합니다. 다음 요청 헤더를 사용하십시오.

> 버킷을 생성할 때 S3 오브젝트 잠금을 활성화해야 합니다. 버킷을 생성한 후에는 S3 오브젝트 잠금을 추가하거나 비활성화할 수 없습니다.

- 'x-amz-object-lock-mode'로, 반드시 준수(대소문자 구분)되어야 합니다.

  > x-amz-object-lock-mode를 지정할 경우 x-amz-object-lock-retain-until-date를 지정해야 합니다.

- 'X-amz-object-lock-retain-until-date'
  - 유지 종료 날짜 값은 2020-08-10T21:46:00Z 형식이어야 합니다. 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.

- "x-amz-object-lock- *" 요청 헤더가 Put Object 요청에 있는 경우 Content-MD5" 요청 헤더가 필요합니다. 개체 복사 또는 멀티파트 업로드를 시작하려면 Content-MD5가 필요하지 않습니다.
- 버킷에 S3 오브젝트 잠금이 활성화되어 있지 않고 "x-amz-object-lock- *" 요청 헤더가 있는 경우 400개의 잘못된 요청(InvalidRequest) 오류가 반환됩니다.
- Put Object 요청은 AWS 동작에 맞춰 "x-amz-storage-class:reduced_redundancy"를 사용할 수 있도록 지원합니다. 하지만 오브젝트가 S3 오브젝트 잠금이 설정된 버킷으로 수집되면 StorageGRID는 항상 이중 커밋 수집을 수행합니다.
- 이후 GET 또는 HEAD 객체 버전 응답에는 헤더 "x-amz-OBJECT-LOCK-MODE", "x-amz-OBJECT-REGATE-DATE" 및 "x-amz-OBJECT-LOCK-REGAL-HOLD"가 포함되며, 구성된 경우 요청 송신자가 올바른 '3:GET *' 권한을 가지고 있는 경우 이에 해당합니다.
- 이후 개체 버전 삭제 또는 개체 버전 삭제 요청은 보존 기한 이전이거나 법적 보류가 켜져 있는 경우 실패합니다.

S3 오브젝트 잠금 설정을 업데이트합니다

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

- '개체 법적 증거 자료 보관'

  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.

- '개체 보존'을 선택합니다
  - 모드 값은 규정 준수(대/소문자 구분)여야 합니다.
  - 유지 종료 날짜 값은 2020-08-10T21:46:00Z 형식이어야 합니다. 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.

관련 정보

ILM을 사용하여 개체를 관리합니다

테넌트 계정을 사용합니다

개체 를 넣습니다

개체 - 복사 를 선택합니다

멀티파트 업로드를 시작합니다

오브젝트 버전 관리

"Amazon Simple Storage Service 사용자 가이드: S3 Object Lock 사용"

StorageGRID는 에 대해 다음 AWS 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;=.(&T
- =

- =
- 를 누릅니다
- !=
- 사이
- 인치

패턴 일치 연산자

- 좋아요
- _
- %

단일 작업자

- NULL입니다
- NULL이 아닙니다

수학 연산자

- 를 누릅니다
- -
- *
- /
- %

StorageGRID는 AWS 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-서버측-암호화'

SSE 요청 헤더는 다음 오브젝트 작업에서 지원됩니다.

- 개체 를 넣습니다
- 개체 - 복사 를 선택합니다
- 멀티파트 업로드를 시작합니다

SSE-C를 사용합니다

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

요청 헤더	설명
"x-amz-server-sidesidebencryptionsever-customer-algorithm"입니다	암호화 알고리즘을 지정합니다. 헤더 값은 AES256이어야 합니다.
"x-amz-server-sideebencryptionsever-customer-key"	개체를 암호화하거나 해독하는 데 사용할 암호화 키를 지정합니다. 키의 값은 256비트 base64로 인코딩되어야 합니다.
"X-amz-server-sidemidebencryptionsever-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.

- 객체 가져오기
- 헤드 개체
- 개체 를 넣습니다
- 개체 - 복사 를 선택합니다
- 멀티파트 업로드를 시작합니다
- 부품 업로드
- 업로드 부품 - 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.

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

관련 정보

객체 가져오기

헤드 개체

개체 를 넣습니다

개체 - 복사 를 선택합니다

멀티파트 업로드를 시작합니다

부품 업로드

업로드 부품 - 복사

"Amazon S3 개발자 가이드: 고객 제공 암호화 키(SSE-C)를 사용하여 서버측 암호화를 사용하여 데이터 보호"

객체 가져오기

S3 오브젝트 가져오기 요청을 사용하여 S3 버킷에서 오브젝트를 검색할 수 있습니다.

개체 및 다중 파트 개체를 가져옵니다

'PARTNUMBER' 요청 파라미터를 사용하면 멀티파트 또는 분할된 개체의 특정 부분을 검색할 수 있습니다. X-amz-MP-parts-count 응답 요소는 개체의 부품 수를 나타냅니다.

분할/다중 파트 개체와 비분할/비다중 파트 개체 모두에 대해 'PARTNUMBER'를 1로 설정할 수 있지만, "x-amz-MP-parts-count" 응답 요소는 분할된 또는 다중 파트 개체에 대해서만 반환됩니다.

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-서버측-암호화-고객-알고리즘: AES256 지정.
- 'x-amz-서버측-암호화-고객-키': 오브젝트의 암호화 키를 지정합니다.
- X-amz-서버측-암호화-고객-키-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 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.

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 가져오십시오. 키 이름이나 값에 인쇄할 수 없는 문자가 포함되어 있으면 "x-amz-missing-meta" 헤더가 반환되지 않습니다.

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않으며 'XNotImplemented'를 반환합니다.

- X-amz-웹사이트-리디렉션-위치

버전 관리

rrionId 서브리소스를 지정하지 않으면 가장 최근 버전의 객체가 버전 관리되는 버킷에 페치됩니다. 객체의 현재 버전이 삭제 표식이면 x-amz-DELETE-MARKER 응답 헤더가 TRUE로 설정된 상태로 "Not Found" 상태가 반환됩니다.

Get Object for Cloud Storage Pool 개체의 동작

개체가 클라우드 스토리지 풀에 저장된 경우(정보 수명 주기 관리를 통해 개체 관리 지침 참조) 오브젝트 가져오기 요청의 동작은 개체의 상태에 따라 달라집니다. 자세한 내용은 " 헤드 개체 "를 참조하십시오.

> 객체가 클라우드 스토리지 풀에 저장되고 오브젝트 복사본이 하나 이상 그리드에 존재하는 경우, 객체 가져오기 요청은 클라우드 스토리지 풀에서 데이터를 검색하기 전에 그리드에서 데이터를 검색하려고 시도합니다.

개체의 상태입니다	**Get Object**의 동작입니다
StorageGRID로 수집되었지만 아직 ILM에 의해 평가되지 않은 오브젝트 또는 기존 스토리지 풀에 저장된 오브젝트 또는 삭제 코딩 사용	200OK 개체의 복사본이 검색됩니다.
Cloud Storage Pool의 개체이지만 아직 검색할 수 없는 상태로 전환되지 않았습니다	200OK 개체의 복사본이 검색됩니다.
개체가 검색할 수 없는 상태로 전환되었습니다	403 금지됨, InvalidObjectState 개체를 검색 가능한 상태로 복원하려면 POST 개체 복원 요청을 사용합니다.

개체의 상태입니다	**Get Object**의 동작입니다
복구할 수 없는 상태에서 복원 중인 개체입니다	403 금지됨, InvalidObjectState POST 개체 복원 요청이 완료될 때까지 기다립니다.
객체가 클라우드 스토리지 풀에 완전히 복구되었습니다	200OK 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. 일부 개체의 일부가 이미 검색 불가능한 상태로 전환되었거나 개체의 일부가 아직 복원되지 않은 경우 Get Object 요청이 '200 OK'를 잘못 반환할 수도 있습니다.

다음과 같은 경우:

- Get Object 요청이 일부 데이터를 반환하지만 전송 도중에 중지될 수 있습니다.
- 이후 개체 가져오기 요청이 403 사용 금지 를 반환할 수 있습니다.

관련 정보

서버측 암호화를 사용합니다

ILM을 사용하여 개체를 관리합니다

사후 개체 복원

S3 작업이 감사 로그에서 추적되었습니다

헤드 개체

S3 헤드 오브젝트 요청을 사용하여 오브젝트 자체를 반환하지 않고 오브젝트에서 메타데이터를 검색할 수 있습니다. 객체가 클라우드 스토리지 풀에 저장된 경우 헤드 객체를 사용하여 객체의 전환 상태를 확인할 수 있습니다.

헤드 개체 및 다중 파트 개체

'PARTNUMBER' 요청 파라미터를 사용하면 멀티파트 또는 분할된 개체의 특정 부분에 대한 메타데이터를 검색할 수 있습니다. X-amz-MP-parts-count 응답 요소는 개체의 부품 수를 나타냅니다.

분할/다중 파트 개체와 비분할/비다중 파트 개체 모두에 대해 'PARTNUMBER'를 1로 설정할 수 있지만, "x-amz-MP-parts-count" 응답 요소는 분할된 또는 다중 파트 개체에 대해서만 반환됩니다.

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-서버측-암호화-고객-알고리즘: AES256 지정.

- 'x-amz-서버측-암호화-고객-키': 오브젝트의 암호화 키를 지정합니다.
- X-amz-서버측-암호화-고객-키-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 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.

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 헤드 요청은 키 이름이나 값에 인쇄할 수 없는 문자가 포함된 경우 "x-amz-missing-meta" 헤더를 반환하지 않습니다.

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않으며 'XNotImplemented'를 반환합니다.

- X-amz-웹사이트-리디렉션-위치

클라우드 스토리지 풀 객체에 대한 응답 헤더입니다

객체가 클라우드 스토리지 풀에 저장된 경우(정보 수명 주기 관리를 통해 객체 관리 지침 참조) 다음 응답 헤더가 반환됩니다.

- X-amz-STERAGE-CLASS:빙하
- X-amz-restore

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

개체의 상태입니다	헤드 객체에 대한 응답
StorageGRID로 수집되었지만 아직 ILM에 의해 평가되지 않은 오브젝트 또는 기존 스토리지 풀에 저장된 오브젝트 또는 삭제 코딩 사용	200 OK (특별한 응답 헤더는 반환되지 않음)
Cloud Storage Pool의 개체이지만 아직 검색할 수 없는 상태로 전환되지 않았습니다	200OK X-amz-STERAGE-CLASS:빙하 "x-amz-restore:ongoing-request="false", expiry-date="sat, 23 JULY 20 2030 00:00:00:00 GMT" 객체가 검색 불가능한 상태로 전환되기 전까지 만료 날짜 값은 향후 일정 원거리 시간으로 설정됩니다. 정확한 전환 시간은 StorageGRID 시스템에 의해 제어되지 않습니다.

개체의 상태입니다	헤드 객체에 대한 응답
개체가 검색할 수 없는 상태로 전환되었지만 하나 이상의 복사본이 그리드에 있습니다	200OK X-amz-STERAGE-CLASS:빙하 "x-amz-restore:ongoing-request="false", expiry-date="sat, 23 JULY 20 2030 00:00:00:00 GMT" 만료 날짜 값은 앞으로 어느 정도 먼 시간으로 설정됩니다. • 참고 *: 그리드의 복사본을 사용할 수 없는 경우(예: 스토리지 노드가 다운된 경우), 객체를 성공적으로 검색하기 전에 POST 객체 복원 요청을 발행하여 클라우드 스토리지 풀에서 복제본을 복원해야 합니다.
개체가 검색할 수 없는 상태로 전환되었으며 그리드에 복사본이 없습니다	200OK X-amz-STERAGE-CLASS:빙하
복구할 수 없는 상태에서 복원 중인 개체입니다	200OK X-amz-STERAGE-CLASS:빙하 "x-amz-restore:진행 중인-request="true"'
객체가 클라우드 스토리지 풀에 완전히 복구되었습니다	200OK X-amz-STERAGE-CLASS:빙하 "x-amz-restore:ongoing-request="false", expiry-date="sat, 23 July 20 2018 00:00:00:00 GMT" '만료 날짜'는 Cloud Storage Pool의 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. 일부 개체의 일부가 이미 검색 불가능 상태로 전환되었거나 개체의 일부가 아직 복원되지 않은 경우 헤드 객체 요청이 "x-amz-restore:ongoing-request="false""를 잘못 반환할 수 있습니다.

버전 관리

rrionId 서브리소스를 지정하지 않으면 가장 최근 버전의 객체가 버전 관리되는 버킷에 페치됩니다. 객체의 현재 버전이 삭제 표식이면 x-amz-DELETE-MARKER 응답 헤더가 TRUE로 설정된 상태로 "Not Found" 상태가 반환됩니다.

관련 정보

서버측 암호화를 사용합니다

ILM을 사용하여 개체를 관리합니다

사후 개체 복원

S3 작업이 감사 로그에서 추적되었습니다

사후 개체 복원

S3 POST 오브젝트 복원 요청을 사용하여 클라우드 스토리지 풀에 저장된 오브젝트를 복원할 수 있습니다.

지원되는 요청 유형입니다

StorageGRID는 개체 복원을 위한 POST 개체 복원 요청만 지원합니다. 복원의 선택 유형을 지원하지 않습니다. SELECT 요청은 'XNotImplemented'를 반환합니다.

버전 관리

필요한 경우 rionId를 지정하여 버전이 있는 버킷에서 개체의 특정 버전을 복원합니다. rionId를 지정하지 않으면 객체의 최신 버전이 복원됩니다

클라우드 스토리지 풀 객체에 대한 **POST** 객체 복구의 동작

개체가 클라우드 스토리지 풀에 저장된 경우(정보 수명 주기 관리를 통해 개체 관리 지침 참조) POST 개체 복원 요청은 개체의 상태에 따라 다음과 같은 동작을 수행합니다. 자세한 내용은 " 헤드 개체 "를 참조하십시오.

> 객체가 클라우드 스토리지 풀에 저장되어 있고 하나 이상의 오브젝트 복제본도 그리드에 있는 경우 POST 객체 복원 요청을 실행하여 객체를 복원할 필요가 없습니다. 대신 Get Object 요청을 사용하여 로컬 복사본을 직접 검색할 수 있습니다.

개체의 상태입니다	**POST** 개체 복원 동작
StorageGRID로 수집되었지만 ILM에서 아직 평가되지 않은 오브젝트 또는 클라우드 스토리지 풀에 없는 오브젝트	403 금지됨, InvalidObjectState
Cloud Storage Pool의 개체이지만 아직 검색할 수 없는 상태로 전환되지 않았습니다	200 OK 어떤 변경도 없다. • 참고 *: 객체가 검색 불가능한 상태로 전환되기 전에는 해당 '만료 날짜'를 변경할 수 없습니다.

개체의 상태입니다	**POST** 개체 복원 동작
개체가 검색할 수 없는 상태로 전환되었습니다	'202 수락됨'은 요청 본문에 지정된 일 동안 검색할 수 있는 객체 복사본을 클라우드 스토리지 풀에 복원합니다. 이 기간이 끝나면 객체는 복구할 수 없는 상태로 돌아갑니다. 선택적으로 "계층" 요청 요소를 사용하여 복원 작업이 완료될 때까지 걸리는 시간("신속", "표준" 또는 "대량")을 결정합니다. '계층'을 지정하지 않으면 '표준' 계층이 사용됩니다. • 주의 *: 오브젝트가 S3 Glacier Deep Archive로 전환되었거나 Cloud Storage Pool이 Azure Blob Storage를 사용하는 경우 "빠른" 계층을 사용하여 복원할 수 없습니다. 다음 오류는 403 사용 금지, InvalidTier로 반환됩니다. 검색 옵션은 이 저장소 클래스에서 지원되지 않습니다.
복구할 수 없는 상태에서 복원 중인 개체입니다	409갈등대, RestoreAlreadyInProgress
객체가 클라우드 스토리지 풀에 완전히 복구되었습니다	200OK • 참고: * 개체가 검색 가능한 상태로 복원되면, POST 개체 복원 요청을 새 값으로 다시 발행하여 '만료 날짜'를 변경할 수 있습니다. 복원 날짜는 요청 시간을 기준으로 업데이트됩니다.

관련 정보

ILM을 사용하여 개체를 관리합니다

헤드 개체

S3 작업이 감사 로그에서 추적되었습니다

개체 를 넣습니다

S3 PUT 오브젝트 요청을 사용하여 오브젝트를 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 PUT 오브젝트 작업의 maximum_recommended_size는 5GiB(5,368,709,120바이트)입니다. 5GiB보다 큰 객체가 있는 경우 대신 멀티파트 업로드를 사용합니다.

> StorageGRID 11.6에서 단일 PUT 객체 작업에 대한 Maximum_supported_size는 5TiB(5,497,558,138,880바이트)입니다. 그러나 5GiB를 초과하는 개체를 업로드하려고 하면 * S3 PUT 오브젝트 크기가 너무 큼 * 경고가 트리거됩니다.

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 Put, Put Object-Copy, Get 및 head 요청이 성공합니다.
- StorageGRID는 키 이름 또는 값의 해석된 값에 인쇄할 수 없는 문자가 포함된 경우 "x-amz-missing-meta" 헤더를 반환하지 않습니다.

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.

- 캐시-컨트롤
- 'Content-Disposition'
- 콘텐츠 인코딩

  Content-Encoding의 torageGRID에 대해 AWS-chunched를 지정하면 다음 항목을 확인할 수 없습니다.

  - StorageGRID는 청크 데이터에 대해 청크 서명을 확인하지 않습니다.
  - StorageGRID는 오브젝트에 대해 사용자가 "x-amz-decoded-content-length"를 제공하는 값을 확인하지 않습니다.
- 컨텐츠-언어

- 콘텐츠 길이
- 내용-MD5
- 콘텐츠 유형
- '만료'
- 전송 인코딩

  AWS-chunched 페이로드 서명도 사용되는 경우 청크된 전송 인코딩이 지원됩니다.

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

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

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

  ILM 규칙의 참조 시간으로 * 사용자 정의 작성 시간 * 옵션을 사용하려면 객체를 만들 때 기록하는 메타데이터의 이름으로 "creation-time"을 사용해야 합니다. 예를 들면 다음과 같습니다.

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

  창조시간 값은 1970년 1월 1일 이후 초 단위로 평가된다.

  ILM 규칙은 참조 시간에 * 사용자 정의 작성 시간 * 과 수집 동작에 대한 균형 또는 엄격 옵션을 모두 사용할 수 없습니다. ILM 규칙을 만들면 오류가 반환됩니다.

- X-amz-태깅
- 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 오브젝트 잠금을 사용합니다

- SSE 요청 헤더:
  - 'X-amz-서버측-암호화'
  - X-amz-서버측-암호화-고객-키-MD5
  - 'X-amz-서버측-암호화-고객-키'
  - 'X-amz-서버측-암호화-고객-알고리즘'

    을 참조하십시오 서버측 암호화에 대한 요청 헤더

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않습니다.

- X-amz-ACL 요청 헤더는 지원되지 않습니다.
- X-amz-website-redirect-location 요청 헤더는 지원되지 않으며 XNotImplemented를 반환합니다.

스토리지 클래스 옵션

X-amz-STERAGE-CLASS 요청 헤더는 지원된다. 'x-amz-storage-class'에 대해 제출된 가치는 수집 중에 StorageGRID가 오브젝트 데이터를 보호하는 방식에 영향을 주며, ILM에 의해 결정되는 StorageGRID 시스템에 저장된 개체의 영구 복사본의 수가 아닙니다.

수집된 개체와 일치하는 ILM 규칙이 Ingest 동작에 Strict 옵션을 사용하는 경우 "x-amz-storage-class" 헤더는 영향을 주지 않습니다.

X-amz-storage-class에 사용할 수 있는 값은 다음과 같다.

- '표준'(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(동기 배치)에 지정된 모든 오브젝트 복사본을 즉시 생성할 수 있으면 'x-amz-storage-class' 헤더가 효과가 없습니다.
- Reduced_redundancy를 선택합니다
  - * 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. Reduced_redundancy 옵션은 개체와 일치하는 ILM 규칙이 복제된 단일 복사본을 만들 때 가장 적합합니다. 이 경우 'REDED_READITORY'를 사용하면 모든 수집 작업에 대해 불필요한 오브젝트 복사본을 생성하고 삭제할 필요가 없습니다.

  다른 상황에서는 reducted_redundancy 옵션을 사용하지 않는 것이 좋습니다. REDED_READITAINEY는 수집 중에 오브젝트 데이터가 손실될 위험을 증가시킵니다. 예를 들어, ILM 평가가 발생하기 전에 실패한 스토리지 노드에 단일 복사본이 처음 저장되는 경우 데이터가 손실될 수 있습니다.
- 주의 *: 한 번에 하나의 복제 사본만 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.

reducted_redundancy를 지정하면 개체를 처음 인제스트할 때 생성되는 복제본 수만 영향을 받습니다. 활성 ILM 정책에 따라 개체를 평가할 때 개체의 복사본 수에 영향을 주지 않으며 StorageGRID 시스템에서 낮은 수준의 중복성에 데이터가 저장되지 않습니다.

- 참고 *: S3 오브젝트 잠금이 활성화된 버킷으로 오브젝트를 인스팅하는 경우, reducted_redundancy 옵션이

무시됩니다. 개체를 레거시 준수 버킷으로 인스팅하는 경우 REDED_REPREADITORIAL' 옵션은 오류를 반환합니다. 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-서버측-암호화'
- * SSE-C *: 사용자가 제공 및 관리하는 고유 키로 객체를 암호화하려면 이 헤더 세 개를 모두 사용합니다.
  - X-amz-서버측-암호화-고객-알고리즘: AES256 지정.
  - 'X-amz-서버측-암호화-고객 키': 새 오브젝트의 암호화 키를 지정합니다.
  - X-amz-서버측-암호화-고객-키-MD5: 새 개체의 암호화 키에 대해 MD5 다이제스트를 지정합니다.
- 주의: * 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.
- 참고: * 개체가 SSE 또는 SSE-C로 암호화된 경우 버킷 수준 또는 그리드 수준 암호화 설정은 무시됩니다.

버전 관리

버켓에 대한 버전 관리가 활성화된 경우 저장할 개체의 버전에 대해 고유한 rionId가 자동으로 생성됩니다. 이 인상파 ID는 X-amz-version-id 응답 헤더를 사용하여 응답에서도 반환됩니다.

버전 관리가 일시 중단된 경우 개체 버전은 null rionId로 저장되며, null 버전이 이미 있는 경우에는 덮어쓰게 됩니다.

관련 정보

ILM을 사용하여 개체를 관리합니다

버킷 작업

S3 작업이 감사 로그에서 추적되었습니다

서버측 암호화를 사용합니다

클라이언트 연결 구성 방법

개체 - 복사 를 선택합니다

S3 PUT 오브젝트 복사 요청을 사용하여 S3에 이미 저장된 오브젝트 복사본을 생성할 수 있습니다. Put Object - Copy 작업은 GET 및 PUT를 수행하는 작업과 동일합니다.

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 PUT 오브젝트 작업의 maximum_recommended_size는 5GiB(5,368,709,120바이트)입니다. 5GiB보다 큰 객체가 있는 경우 대신 멀티파트 업로드를 사용합니다.

> StorageGRID 11.6에서 단일 PUT 객체 작업에 대한 Maximum_supported_size는 5TiB(5,497,558,138,880바이트)입니다. 그러나 5GiB를 초과하는 개체를 업로드하려고 하면 * S3 PUT 오브젝트 크기가 너무 큼 * 경고가 트리거됩니다.

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 키 이름 또는 값의 해석된 값에 인쇄할 수 없는 문자가 포함된 경우 "x-amz-missing-meta" 헤더를 반환하지 않습니다.

지원되는 요청 헤더입니다

지원되는 요청 헤더는 다음과 같습니다.

- 콘텐츠 유형
- X-amz-copy 소스
- X-amz-copy-source-if-match
- X-amz-copy-source-if-none-match
- X-amz-copy-source-if-수정되지 않음-since
- X-amz-copy-source-if-modified-since
- x-amz-meta- 뒤에 사용자 정의 메타데이터를 포함하는 이름-값 쌍이 있습니다
- 'x-amz-metadata-directive': 기본값은 'copy'로, 객체와 관련 메타데이터를 복사할 수 있습니다.

  오브젝트를 복사할 때 기존 메타데이터를 덮어쓰거나 오브젝트 메타데이터를 업데이트하도록 "replace"를 지정할 수 있습니다.

- X-amz-스토리지 클래스
- 'x-amz-tagging-directive': 기본값은 'copy'로, 객체와 모든 태그를 복사할 수 있습니다.

  개체를 복사할 때 기존 태그를 덮어쓰거나 태그를 업데이트하려면 "다시 배치"를 지정할 수 있습니다.

- 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 오브젝트 잠금을 사용합니다

- SSE 요청 헤더:
  - "x-amz-copy-sourcestevids-server-sideencryptionsever-customer-algorithm"입니다
  - X-amz-copy-sources.\xserver-side-encryption-customer-key
  - "X-amz-copy-sourcestifs-server-side-encryption-customer-key-md5'
  - 'X-amz-서버측-암호화'
  - X-amz-서버측-암호화-고객-키-MD5
  - 'X-amz-서버측-암호화-고객-키'
  - 'X-amz-서버측-암호화-고객-알고리즘'

  을 참조하십시오 서버측 암호화에 대한 요청 헤더

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않습니다.

- 캐시-컨트롤
- 'Content-Disposition'
- 콘텐츠 인코딩
- 컨텐츠-언어
- '만료'
- X-amz-웹사이트-리디렉션-위치

스토리지 클래스 옵션

"x-amz-storage-class" 요청 헤더가 지원되며 일치하는 ILM 규칙에서 이중 커밋 또는 밸런스의 수집 동작을 지정하는 경우 StorageGRID에서 생성되는 개체 복사본 수에 영향을 줍니다.

- '표준'입니다

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

- Reduced_redundancy를 선택합니다

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

> S3 오브젝트 잠금이 활성화된 버킷으로 오브젝트를 인스팅하는 경우 REDED_REPREADITORY' 옵션이 무시됩니다. 개체를 레거시 준수 버킷으로 인스팅하는 경우 REDED_REPREADITORIAL' 옵션은 오류를 반환합니다. StorageGRID은 규정 준수 요구 사항이 충족될 수 있도록 항상 이중 커밋 수집을 수행합니다.

Put Object - Copy에서 **x-amz-copy-source** 사용

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

소스와 대상이 일치하고 "x-amz-metadata-directive" 헤더가 replace"로 지정된 경우 해당 요청의 메타데이터 값으로 오브젝트의 메타데이터가 업데이트됩니다. 이 경우 StorageGRID는 오브젝트를 다시 수집하지 않습니다. 여기에는 두 가지 중요한 결과가 있습니다.

- Put Object-Copy를 사용하여 기존 개체를 현재 위치에서 암호화하거나 기존 개체의 암호화를 변경할 수 없습니다. X-amz-서버측-암호화 헤더나 x-amz-서버측-암호화-고객-알고리즘 헤더를 제공하면 StorageGRID는 요청을 거부하고 XNotImplemented를 반환합니다.
- 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

서버 측 암호화를 사용하는 경우 소스 개체가 암호화되었는지 여부 및 대상 개체를 암호화할 계획인지에 따라 요청 헤더가 제공됩니다.

- 소스 객체가 SSE-C(customer-provided key)를 사용하여 암호화된 경우, 객체를 해독한 다음 복사할 수 있도록 객체 복사 요청(Put Object-Copy request)에 다음 세 개의 헤더를 포함해야 합니다.
  - X-amz-copy-sourcesources. ever-sideboretationsencryptionsever-customer-algorithm은 AES256을 지정합니다.
  - 'x-amz-copy-sourcesources.x.server-side-encryption-customer-key'는 소스 객체를 만들 때 제공한 암호화 키를 지정합니다.
  - "x-amz-copy-sourcesourcesifx-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-서버측-암호화-고객-알고리즘: AES256 지정.
  - 'X-amz-서버측-암호화-고객-키': 대상 객체에 대한 새 암호화 키를 지정합니다.
  - X-amz-서버측-암호화-고객-키-MD5: 새 암호화 키의 MD5 다이제스트를 지정합니다.
- 주의: * 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.
- SSE(StorageGRID)에서 관리되는 고유 키로 대상 객체(사본)를 암호화하려면 객체 복사 요청(Put Object-Copy request)에 이 헤더를 포함시킵니다.
  - 'X-amz-서버측-암호화'

참고: * 오브젝트의 '서버측 암호화' 값은 업데이트할 수 없습니다. 대신 X-amz-metadata-directive:replace를 사용하여 새로운 서버 측 암호화 값으로 복사본을 만듭니다.

버전 관리

소스 버킷의 버전이 있는 경우 "x-amz-copy-source" 헤더를 사용하여 객체의 최신 버전을 복사할 수 있습니다. 개체의 특정 버전을 복사하려면 rionId 하위 리소스를 사용하여 복사할 버전을 명시적으로 지정해야 합니다. 목적지 버킷의 버전 관리가 되면 생성된 버전은 'x-amz-version-id' 응답 헤더로 반환됩니다. 대상 버킷의 버전 관리가 일시 중단된 경우 x-amz-version-id는 "null" 값을 반환합니다.

관련 정보

ILM을 사용하여 개체를 관리합니다

서버측 암호화를 사용합니다

S3 작업이 감사 로그에서 추적되었습니다

개체 를 넣습니다

SelectObjectContent 를 선택합니다

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

자세한 내용은 를 참조하십시오 "SelectObjectContent에 대한 AWS 문서".

필요한 것

- 테넌트 계정에 S3 Select 권한이 있습니다.
- 쿼리할 객체에 대한 '3:GetObject' 권한이 있습니다.
- 쿼리할 객체가 CSV 형식이거나 CSV 형식 파일이 포함된 GZIP 또는 BZIP2 압축 파일입니다.
- SQL 식의 최대 길이는 256KB입니다.
- 입력 또는 결과에 있는 모든 레코드의 최대 길이는 1MiB입니다.

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>
```

SQL 쿼리의 예

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

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

질의할 파일의 처음 몇 줄인 'sub-EST2020_all.csv'는 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 예

```
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.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

이 섹션에서는 StorageGRID가 멀티파트 업로드 작업을 지원하는 방법에 대해 설명합니다.

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

- 해당 버킷에 대한 다중 파트 업로드 나열 쿼리 결과는 불완전한 결과를 반환할 수 있으므로 단일 버킷에 대한 동시 다중 파트 업로드 1,000개를 초과할 수 없습니다.
- 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 Strict 또는 Balanced 수집 동작을 사용하는 경우 ILM은 다중 파트 개체의 각 부분을 인제스트할 때 계산되고 다중 파트 업로드가 완료될 때 전체 개체에 대해 평가됩니다. 이 사항이 개체 및 파트 배치에 미치는 영향에 대해 알고 있어야 합니다.
  - S3 멀티파트 업로드가 진행 중인 동안 ILM이 변경되면 멀티파트 업로드가 완료될 때 개체의 일부 부분이 현재 ILM 요구 사항을 충족하지 못할 수 있습니다. 올바르게 배치되지 않은 모든 부품은 ILM 재평가를 위해 대기 중이며 나중에 올바른 위치로 이동됩니다.
  - 파트에 대한 ILM을 평가할 때 StorageGRID은 개체의 크기가 아닌 파트 크기를 필터링합니다. 즉, 개체의 일부를 개체의 ILM 요구 사항을 전체가 충족하지 않는 위치에 저장할 수 있습니다. 예를 들어, 규칙이 모든 오브젝트 10GB 이상이 DC1에 저장되는 반면 모든 작은 오브젝트는 DC2에 저장되는 것으로 지정하는 경우 10개 부분 멀티파트 업로드의 각 1GB 부분은 DC2에 저장됩니다. 개체에 대한 ILM을 전체적으로 평가할 때 개체의 모든 부분이 DC1로 이동합니다.
- 모든 멀티파트 업로드 작업은 StorageGRID 정합성 제어를 지원합니다.
- 필요한 경우 다중 파트 업로드와 함께 서버측 암호화를 사용할 수 있습니다. SSE(StorageGRID 관리 키가 있는 서버쪽 암호화)를 사용하려면 '다중 파트 업로드 시작' 요청에만 'x-amz-서버측-암호화' 요청 헤더가 포함됩니다. SSE-C(고객이 제공한 키와 함께 서버측 암호화)를 사용하려면 다중 파트 업로드 시작 요청 및 각 후속 업로드 파트 요청에서 동일한 세 가지 암호화 키 요청 헤더를 지정합니다.

작동	구축
다중 파트 업로드 나열	을 참조하십시오 다중 파트 업로드 나열
멀티파트 업로드를 시작합니다	을 참조하십시오 멀티파트 업로드를 시작합니다

작동	구축
부품 업로드	을 참조하십시오 부품 업로드
업로드 부품 - 복사	을 참조하십시오 업로드 부품 - 복사
멀티파트 업로드를 완료합니다	을 참조하십시오 멀티파트 업로드를 완료합니다
멀티파트 업로드를 중단합니다	모든 Amazon S3 REST API 동작으로 구현됩니다
파트 목록	모든 Amazon S3 REST API 동작으로 구현됩니다

관련 정보

- 일관성 제어
- 서버측 암호화를 사용합니다

다중 파트 업로드 나열

다중 파트 업로드 나열 작업은 버킷에 대해 진행 중인 다중 파트 업로드를 나열합니다.

지원되는 요청 매개 변수는 다음과 같습니다.

- 인코딩 형식
- 최대 업로드
- 키-마커
- 접두사
- 업로드-ID-마커

기한도자 요청 매개변수는 지원되지 않습니다.

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. 전체 다중 파트 업로드 작업이 수행되는 경우, 즉 개체가 작성되는 시점(해당되는 경우 버전)입니다.

멀티파트 업로드를 시작합니다

다중 파트 업로드 시작 작업은 개체에 대한 다중 파트 업로드를 시작하고 업로드 ID를 반환합니다.

X-amz-STERAGE-CLASS 요청 헤더는 지원된다. 'x-amz-storage-class'에 대해 제출된 가치는 수집 중에 StorageGRID가 오브젝트 데이터를 보호하는 방식에 영향을 주며, ILM에 의해 결정되는 StorageGRID 시스템에 저장된 개체의 영구 복사본의 수가 아닙니다.

수집된 개체와 일치하는 ILM 규칙이 Ingest 동작에 Strict 옵션을 사용하는 경우 "x-amz-storage-class" 헤더는 영향을

주지 않습니다.

X-amz-storage-class에 사용할 수 있는 값은 다음과 같다.

- '표준'(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(동기 배치)에 지정된 모든 오브젝트 복사본을 즉시 생성할 수 있으면 'x-amz-storage-class' 헤더가 효과가 없습니다.

- Reduced_redundancy를 선택합니다
  - * 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. Reduced_redundancy 옵션은 개체와 일치하는 ILM 규칙이 복제된 단일 복사본을 만들 때 가장 적합합니다. 이 경우 'REDED_READITORY'를 사용하면 모든 수집 작업에 대해 불필요한 오브젝트 복사본을 생성하고 삭제할 필요가 없습니다.

  다른 상황에서는 reducted_redundancy 옵션을 사용하지 않는 것이 좋습니다. REDED_READITAINEY는 수집 중에 오브젝트 데이터가 손실될 위험을 증가시킵니다. 예를 들어, ILM 평가가 발생하기 전에 실패한 스토리지 노드에 단일 복사본이 처음 저장되는 경우 데이터가 손실될 수 있습니다.

- 주의 *: 한 번에 하나의 복제 사본만 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.

reducted_redundancy를 지정하면 개체를 처음 인제스트할 때 생성되는 복제본 수만 영향을 받습니다. 활성 ILM 정책에 따라 개체를 평가할 때 개체의 복사본 수에 영향을 주지 않으며 StorageGRID 시스템에서 낮은 수준의 중복성에 데이터가 저장되지 않습니다.

- 참고 *: S3 오브젝트 잠금이 활성화된 버킷으로 오브젝트를 인스팅하는 경우, reducted_redundancy 옵션이 무시됩니다. 개체를 레거시 준수 버킷으로 인스팅하는 경우 REDED_REPREADITORIAL' 옵션은 오류를 반환합니다. StorageGRID은 규정 준수 요구 사항이 충족될 수 있도록 항상 이중 커밋 수집을 수행합니다.

지원되는 요청 헤더는 다음과 같습니다.

- 콘텐츠 유형
- x-amz-meta- 뒤에 사용자 정의 메타데이터를 포함하는 이름-값 쌍이 있습니다

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

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

ILM 규칙의 참조 시간으로 * 사용자 정의 작성 시간 * 옵션을 사용하려면 객체를 만들 때 기록하는 메타데이터의 이름으로 "creation-time"을 사용해야 합니다. 예를 들면 다음과 같습니다.

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

창조시간 값은 1970년 1월 1일 이후 초 단위로 평가된다.

> 레거시 규정 준수 기능이 설정된 버킷에 객체를 추가하는 경우 사용자 정의 메타데이터로 '생성 시간'을 추가할 수 없습니다. 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 오브젝트 잠금 사용

- SSE 요청 헤더:
  - 'X-amz-서버측-암호화'
  - X-amz-서버측-암호화-고객-키-MD5
  - 'X-amz-서버측-암호화-고객-키'
  - 'X-amz-서버측-암호화-고객-알고리즘'

    서버측 암호화에 대한 요청 헤더

> StorageGRID에서 UTF-8 문자를 처리하는 방법에 대한 자세한 내용은 Put Object 설명서를 참조하십시오.

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 객체를 암호화하려면 다중 파트 업로드 시작 요청에서 다음 헤더를 사용하십시오. 업로드 부품 요청에 이 헤더를 지정하지 마십시오.
  - 'X-amz-서버측-암호화'
- * SSE-C *: 사용자가 제공 및 관리하는 고유 키를 사용하여 개체를 암호화하려는 경우 다중 파트 업로드 시작 요청(및 각 후속 업로드 파트 요청)에서 이 헤더 세 개를 모두 사용합니다.
  - X-amz-서버측-암호화-고객-알고리즘: AES256 지정.
  - 'X-amz-서버측-암호화-고객 키': 새 오브젝트의 암호화 키를 지정합니다.

  - X-amz-서버측-암호화-고객-키-MD5: 새 개체의 암호화 키에 대해 MD5 다이제스트를 지정합니다.
- 주의: * 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않으며 'XNotImplemented'를 반환합니다

- X-amz-웹사이트-리디렉션-위치

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. MultiPart Upload 작업이 완료되면 개체가 만들어지고 버전이 적용됩니다(해당하는 경우).

관련 정보

ILM을 사용하여 개체를 관리합니다

서버측 암호화를 사용합니다

개체 를 넣습니다

부품 업로드

파트 업로드 작업은 개체에 대해 여러 부분으로 업로드되는 파트를 업로드합니다.

지원되는 요청 헤더입니다

지원되는 요청 헤더는 다음과 같습니다.

- 콘텐츠 길이
- 내용-MD5

서버측 암호화에 대한 요청 헤더

다중 파트 업로드 시작 요청에 대해 SSE-C 암호화를 지정한 경우 각 업로드 파트 요청에 다음 요청 헤더를 포함해야 합니다.

- X-amz-서버측-암호화-고객-알고리즘: AES256 지정.
- 'x-amz-서버측-암호화-고객 키': '멀티파트 업로드 시작' 요청에서 제공한 암호화 키와 동일한 암호화 키를 지정합니다.
- "X-amz-서버측-암호화-고객-키-MD5': '멀티파트 업로드 시작' 요청에서 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 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. MultiPart Upload 작업이 완료되면 개체가 만들어지고 버전이 적용됩니다(해당하는 경우).

관련 정보

서버측 암호화를 사용합니다

업로드 부품 - 복사

파트 업로드 - 복사 작업은 기존 개체의 데이터를 데이터 소스로 복사하여 개체의 일부를 업로드합니다.

Part-Copy 업로드 작업은 모든 Amazon S3 REST API 동작으로 구현됩니다.

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

지원되는 요청 헤더는 다음과 같습니다.

- X-amz-copy-source-if-match
- X-amz-copy-source-if-none-match
- X-amz-copy-source-if-수정되지 않음-since
- X-amz-copy-source-if-modified-since

서버측 암호화에 대한 요청 헤더

다중 파트 업로드 시작 요청에 대해 SSE-C 암호화를 지정한 경우 각 업로드 파트 - 복사 요청에 다음 요청 헤더를 포함해야 합니다.

- X-amz-서버측-암호화-고객-알고리즘: AES256 지정.
- 'x-amz-서버측-암호화-고객 키': '멀티파트 업로드 시작' 요청에서 제공한 암호화 키와 동일한 암호화 키를 지정합니다.
- "X-amz-서버측-암호화-고객-키-MD5': '멀티파트 업로드 시작' 요청에서 제공한 것과 동일한 MD5 다이제스트를 지정합니다.

소스 객체가 SSE-C(customer-provided key)를 사용하여 암호화된 경우, 객체가 해독되고 복사될 수 있도록 업로드 파트 - 복사 요청에 다음 세 개의 헤더를 포함해야 합니다.

- 'x-amz-copy-sourcesources. ever-sideboretationsencryptionsever-customer-algorithm': AES256 지정.
- 'x-amz-copy-sourcesources.x.server-side-encryption-customer-key': 소스 객체를 만들 때 제공한 암호화 키를 지정합니다.
- "x-amz-copy-sourcesourcesifx-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 "'서버측 암호화 사용'의 고려 사항을 검토하십시오.

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. MultiPart Upload 작업이 완료되면 개체가 만들어지고 버전이 적용됩니다(해당하는 경우).

멀티파트 업로드를 완료합니다

전체 다중 파트 업로드 작업은 이전에 업로드한 파트를 조립하여 개체의 여러 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

"x-amz-storage-class" 요청 헤더가 지원되며 일치하는 ILM 규칙에서 이중 커밋 또는 밸런스의 수집 동작을 지정하는 경우 StorageGRID에서 생성되는 개체 복사본 수에 영향을 줍니다.

- '표준'입니다

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

- Reduced_redundancy를 선택합니다

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

  > S3 오브젝트 잠금이 활성화된 버킷으로 오브젝트를 인스팅하는 경우 REDED_REPREADITORY' 옵션이 무시됩니다. 개체를 레거시 준수 버킷으로 인스팅하는 경우 REDED_REPREADITORIAL' 옵션은 오류를 반환합니다. 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.

> 반환된 "ETag" 값은 MD5 데이터 합계가 아니라 다중 파트 오브젝트에 대한 "ETag" 값의 Amazon S3 API 구현을 따릅니다.

버전 관리

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

버켓에 대한 버전 관리가 활성화된 경우 저장할 개체의 버전에 대해 고유한 rionId가 자동으로 생성됩니다. 이 인상파 ID는 X-amz-version-id 응답 헤더를 사용하여 응답에서도 반환됩니다.

버전 관리가 일시 중단된 경우 개체 버전은 null rionId로 저장되며, null 버전이 이미 있는 경우에는 덮어쓰게 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 메시지는 알림이 실패한 마지막 객체에 대해 "버킷 이름 오브젝트 키에 대한 알림을 게시하지 못했습니다"라고 표시됩니다. (이 메시지를 보려면 * nodes * > **Storage Node* * > * Events * 를 선택합니다. 테이블 상단의 마지막 이벤트 보기) 이벤트 메시지는 '/var/local/log/bycast-err.log'에도 나열됩니다.

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

관련 정보

ILM을 사용하여 개체를 관리합니다

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
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

이름	**HTTP** 상태입니다
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
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

이름	**HTTP** 상태입니다
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
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

이름	설명	HTTP 상태입니다
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 S3 REST API 작업

StorageGRID 시스템별 S3 REST API에 작업이 추가됩니다.

- 버킷 정합성 보장 요청 가져오기

  Get Bucket 정합성 보장 요청을 사용하면 특정 버킷에 적용되는 정합성 보장 수준을 확인할 수 있습니다.

- 버킷 정합성 보장 요청을 배치합니다

  PUT 버킷 정합성 보장 요청을 사용하면 버킷에서 수행된 작업에 적용할 정합성 수준을 지정할 수 있습니다.

- 버킷 최종 액세스 시간 요청 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.

- 버킷 최종 액세스 시간 요청

  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.

- 버킷 메타데이터 알림 구성 요청을 삭제합니다

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

- 버킷 메타데이터 알림 구성 요청을 가져옵니다

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

- PUT 버킷 메타데이터 알림 구성 요청

  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.

- 스토리지 사용 요청 가져오기

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

- 레거시 규정 준수에 대한 더 이상 사용되지 않는 버킷 요청

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

버킷 정합성 보장 요청 가져오기

Get Bucket 정합성 보장 요청을 사용하면 특정 버킷에 적용되는 정합성 보장 수준을 확인할 수 있습니다.

기본 정합성 보장 컨트롤은 새로 생성된 객체에 대해 읽기/쓰기 작업을 보장하도록 설정됩니다.

이 작업을 완료하려면 S3:GetBucketConsistency 권한이 있거나 계정 루트가 됩니다.

요청 예

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

응답

응답 XML에서 "<Consistency>"는 다음 값 중 하나를 반환합니다.

일관성 제어	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.
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 Amazon S3 일관성 보장과 가장 비슷합니다. • 참고: * 응용 프로그램이 존재하지 않는 개체에 대한 헤드 요청을 사용하는 경우 하나 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다. 이러한 오류를 방지하려면 Amazon S3와 유사한 일관성 보장이 필요하지 않는 한 일관성 제어를 ""사용 가능""으로 설정합니다.

일관성 제어	설명
사용 가능(헤드 작업의 최종 일관성)	"새 쓰기 후 다시 쓰기" 정합성 수준과 동일하게 동작하지만 헤드 작업에 대한 최종 정합성 보장만 제공합니다. 스토리지 노드를 사용할 수 없는 경우 "새 쓰기 후"보다 헤드 작업에 더 높은 가용성을 제공합니다. 헤드 작업에 대한 Amazon S3 정합성 보장과 다릅니다.

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 제어

버킷 정합성 보장 요청을 배치합니다

PUT 버킷 정합성 보장 요청을 사용하면 버킷에서 수행된 작업에 적용할 정합성 수준을 지정할 수 있습니다.

기본 정합성 보장 컨트롤은 새로 생성된 객체에 대해 읽기/쓰기 작업을 보장하도록 설정됩니다.

이 작업을 완료하려면 S3:PutBucketConsistency 권한이 있거나 계정 루트가 됩니다.

요청하십시오

"x-ntap-sg-consistency" 매개 변수는 다음 값 중 하나를 포함해야 합니다.

일관성 제어	설명
모두	모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다.
강함 - 글로벌	모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 정합성을 보장합니다.

일관성 제어	설명
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 Amazon S3 일관성 보장과 가장 비슷합니다. • 참고: * 응용 프로그램이 존재하지 않는 개체에 대한 헤드 요청을 사용하는 경우 하나 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다. 이러한 오류를 방지하려면 Amazon S3와 유사한 일관성 보장이 필요하지 않는 한 일관성 제어를 ""사용 가능""으로 설정합니다.
사용 가능(헤드 작업의 최종 일관성)	"새 쓰기 후 다시 쓰기" 정합성 수준과 동일하게 동작하지만 헤드 작업에 대한 최종 정합성 보장만 제공합니다. 스토리지 노드를 사용할 수 없는 경우 "새 쓰기 후"보다 헤드 작업에 더 높은 가용성을 제공합니다. 헤드 작업에 대한 Amazon S3 정합성 보장과 다릅니다.

- 참고: * 일반적으로 "새 쓰기 후" 정합성 보장 제어 값을 사용해야 합니다. 요청이 올바르게 작동하지 않는 경우 가능한 경우 응용 프로그램 클라이언트 동작을 변경합니다. 또는 클라이언트가 각 API 요청에 대한 정합성 제어를 지정하도록 구성합니다. 버킷 레벨에서만 정합성 제어를 최후의 수단으로 설정하십시오.

요청 예

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

관련 정보

일관성 제어

버킷 최종 액세스 시간 요청 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 루트가 됩니다.

요청 예

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

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 액세스 시간 요청

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:PutBucketLastAccessTime 권한이 있거나 계정 루트가 됩니다.

> 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. 테넌트 관리자의 Put Bucket Last Access Time 요청, * S3 * > * Bucket * > * Change Last Access Setting * 확인란 또는 테넌트 관리 API를 사용하여 마지막 액세스 시간에 대한 업데이트를 활성화하거나 비활성화할 수 있습니다.

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

- 객체 가져오기, 객체 ACL 가져오기, 객체 태그 지정 가져오기 및 헤드 객체 요청은 마지막 액세스 시간을 업데이트하지 않습니다. ILM(정보 수명 주기 관리) 평가를 위해 객체가 대기열에 추가되지 않습니다.
- Put Object - 메타데이터만 업데이트하는 객체 태그 지정 요청을 복사하고 배치하면 마지막 액세스 시간도 업데이트됩니다. ILM 평가를 위해 오브젝트가 대기열에 추가됩니다.
- 소스 버킷에 대해 마지막 액세스 시간에 대한 업데이트를 사용할 수 없는 경우 객체 복사 요청을 소스 버킷의 마지막 액세스 시간을 업데이트하지 않습니다. 복사된 객체는 소스 버킷에 대한 ILM 평가를 위해 대기열에 추가되지 않습니다. 그러나 대상의 경우, 개체 복사 요청은 항상 마지막 액세스 시간을 업데이트합니다. ILM 평가를 위해

개체의 복사본이 대기열에 추가됩니다.

- 완료 다중 파트 업로드 요청 마지막 액세스 시간 업데이트 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: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

이 예제에서는 버킷의 마지막 액세스 시간을 비활성화합니다.

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

관련 정보

테넌트 계정을 사용합니다

버킷 메타데이터 알림 구성 요청을 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 권한 또는 계정 루트 권한이 있어야 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: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>
```

버킷 메타데이터 알림 구성 요청을 가져옵니다

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 권한 또는 계정 루트 권한이 있어야 합니다.

요청 예

이 요청은 bucket이라는 이름의 버킷에 대한 메타데이터 알림 구성을 검색합니다.

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

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. • 세 번째 요소는 'es'여야 합니다. • URN은 메타데이터가 저장된 인덱스 및 형식으로 domain-name/myindex/MyType 형식으로 끝나야 합니다. 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

"<MetadataNotificationConfiguration></MetadataNotificationConfiguration>" 태그 사이에 포함된 XML은 버킷에 대한 검색 통합 끝점과의 통합이 어떻게 구성되어 있는지 보여줍니다. 이 예에서 객체 메타데이터는 'Current'라는 Elasticsearch 인덱스로 전송되고 있으며 'rest코드'라는 AWS 도메인에서 호스팅되는 '2017'이라는 유형으로 전송됩니다.

```
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 버킷 메타데이터 알림 구성 요청

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 PutBucketMetadataNotification 권한 또는 계정 루트 권한이 있어야 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, 접두어 '/images'가 있는 객체의 메타데이터를 한 대상에 전송하고 접두어 '/videos'가 있는 객체를 다른 대상으로 전송할 수 있습니다.

중복되는 접두사가 있는 구성은 유효하지 않으며 제출될 때 거부됩니다. 예를 들어 접두사 test를 가진 개체에 대해 하나의 규칙을 포함하고 test2 접두사가 있는 개체에 대해 두 번째 규칙을 포함하는 구성은 허용되지 않습니다.

대상은 StorageGRID 끝점의 URN을 사용하여 지정해야 합니다. 메타데이터 알림 설정이 제출되거나 요청이 '400 Bad Request'로 실패하는 경우 단말 장치가 존재해야 한다. "메타데이터 알림(검색) 정책을 저장할 수 없습니다. 지정한 끝점 URN이 없습니다:_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. • 세 번째 요소는 'es'여야 합니다. • URN은 메타데이터가 저장된 인덱스 및 형식으로 domain-name/myindex/MyType 형식으로 끝나야 합니다. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 다음과 같은 형식을 취합니다. • "arn:aws: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: <em>date</em>
Authorization: <em>authorization string</em>
Host: <em>host</em>

<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 예에서는 접두사 /images와 일치하는 객체의 객체 메타데이터가 한 대상으로 전송되고 접두사 /videos와 일치하는 객체의 객체 메타데이터는 두 번째 대상으로 전송됩니다.

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

<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 예에서는 'test'라는 버킷에 'gws/tagging.txt' 키가 있는 객체가 생성될 때 생성될 수 있는 JSON의 예를 보여 줍니다. 시험용 버킷은 버전 관리가 되지 않아 rionId 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	우동동-1 등 버킷 지역
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

- 참고: * 태그 및 사용자 메타데이터의 경우 StorageGRID는 날짜 및 숫자를 Elasticsearch에 문자열로 전달하거나 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.

어카운트 및 해당 버킷에 사용되는 스토리지의 양은 'x-ntap-sg-usage' 쿼리 매개 변수를 사용하는 수정된 Get Service 요청을 통해 얻을 수 있습니다. 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 루트 권한이 있어야 합니다.

요청 예

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

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 응답에서 ObjectCount 및 DataBytes 값에 기여합니다. 삭제 표식이 ObjectCount 합계에 추가되지 않습니다.

관련 정보

일관성 제어

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 오브젝트 잠금을 사용합니다
- 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 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.

- 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.

사용되지 않음: 규정 준수를 위해 버킷 요청 수정을 배치합니다

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.

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

S3 오브젝트 잠금을 사용합니다

ILM을 사용하여 개체를 관리합니다

"NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

Compliance가 설정된 새 버킷을 더 이상 생성할 수 없습니다. 새 준수 버킷을 생성하기 위해 준수 준수를 위해 Put Bucket 요청 수정을 사용하려는 경우 다음 오류 메시지가 반환됩니다.

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

관련 정보

ILM을 사용하여 개체를 관리합니다

테넌트 계정을 사용합니다

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.

S3 오브젝트 잠금을 사용합니다

ILM을 사용하여 개체를 관리합니다

"NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

이 작업을 완료하려면 S3:GetBucketCompliance 권한이 있거나 계정 루트가 됩니다.

요청 예

이 예제 요청을 사용하여 'mybucket'이라는 이름의 버킷에 대한 준수 설정을 확인할 수 있습니다.

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

응답 예

응답 XML에서 "<SGCompliance>"는 버킷에 적용되는 준수 설정을 나열합니다. 이 예제 응답에서는 오브젝트를 그리드에 인제스트하는 시점을 시작으로 각 오브젝트를 1년(525,600분)동안 보존할 버킷의 규정 준수 설정을 보여줍니다. 현재 이 버킷에 대한 법적 보류가 없습니다. 각 개체는 1년 후에 자동으로 삭제됩니다.

```
HTTP/1.1 200 OK
Date: <em>date</em>
Connection: <em>connection</em>
Server: StorageGRID/11.1.0
x-amz-request-id: <em>request ID</em>
Content-Length: <em>length</em>
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.
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

버킷을 규정에 맞게 만들지 않은 경우 응답에 대한 HTTP 상태 코드는 XNoSuchBuckketCompliance의 S3 오류 코드와 함께 404를 찾을 수 없습니다.

관련 정보

ILM을 사용하여 개체를 관리합니다

테넌트 계정을 사용합니다

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.

S3 오브젝트 잠금을 사용합니다

ILM을 사용하여 개체를 관리합니다

"NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

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

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

요청 예

이 예제 요청은 'mybucket'이라는 이름의 버킷에 대한 준수 설정을 수정합니다. 이 예에서는 객체가 그리드에 인제된 후 1년이 아닌 2년(1,051,200분) 동안 mybucket의 객체가 보존됩니다. 이 버킷에는 법적 구속이 없습니다. 각 개체는 2년 후에 자동으로 삭제됩니다.

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: <em>date</em>
Authorization: <em>authorization name</em>
Host: <em>host</em>
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. • 주의: * RetenitionPeriodMinutes에 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 설정을 위한 정합성 보장 레벨

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

유지하도록 보장합니다.

데이터 센터 사이트 또는 사이트의 여러 스토리지 노드를 사용할 수 없어 StorageGRID가 * 강력한 글로벌 * 정합성 수준을 달성할 수 없는 경우 응답에 대한 HTTP 상태 코드는 503 서비스를 사용할 수 없습니다

이 응답을 받으면 그리드 관리자에게 문의하여 필요한 스토리지 서비스를 가능한 빨리 사용할 수 있도록 해야 합니다. 그리드 관리자가 각 사이트에서 충분한 스토리지 노드를 사용할 수 없는 경우, 기술 지원 부서에서 * strong-site * 정합성 보장 수준을 강제로 진행하여 실패한 요청을 다시 시도하도록 할 수 있습니다.

> 기술 지원 부서의 지시가 있는 경우를 제외하고, 이 레벨을 사용할 경우 발생할 수 있는 결과를 이해하지 않는 한 * 강력한 사이트 * 일관성 수준을 강제로 버킷 규정 준수를 강제하지 마십시오.

정합성 보장 수준을 * strong-site * 로 축소하면 StorageGRID는 업데이트된 규정 준수 설정이 사이트 내의 클라이언트 요청에 대해서만 읽기/쓰기 후 일관성을 갖게 됩니다. 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 아래에 놓고 정합성 보장 수준을 낮추면 버킷의 이전 규정 준수 설정(즉, 법적 증거 자료 보관)이 일부 데이터 센터 사이트에서 계속 적용될 수 있습니다. 따라서 보존 기간이 만료되면 사용자나 자동 삭제(활성화된 경우)에 의해 법적 보류라고 생각하는 개체가 삭제될 수 있습니다.

strong-site * 정합성 보장 수준을 강제로 사용하려면 PUT Bucket 준수 요청을 다시 발행하고 다음과 같이 "Consistency-Control" HTTP 요청 헤더를 포함시킵니다.

```
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 404를 찾을 수 없습니다.
- 요청의 RetenitionPeriodMinutes가 버킷의 현재 보존 기간보다 짧으면 HTTP 상태 코드는 400개의 잘못된 요청입니다.

관련 정보

사용되지 않음: 규정 준수를 위해 버킷 요청 수정을 배치합니다

테넌트 계정을 사용합니다

ILM을 사용하여 개체를 관리합니다

버킷 및 그룹 액세스 정책

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 * - 버킷 정책 가져오기, 버킷 정책 적용 및 버킷 정책 삭제 S3 API 작업을 사용하여 구성됩니다. 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.

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.

요소	설명
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 경우를 제외하고 와일드카드 문자는 지원되지 않습니다. 예를 들어 와일드카드(*)를 Principal 값으로 설정합니다.

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

다음 예제에서는 Effect , Principal , Action 및 Resource 요소를 사용합니다. 이 예제는 "Allow" 효과를 사용하여 Principals, 관리 그룹 "federated-group/admin" 및 재무 그룹 "federated-group/finance"에 mbucket이라는 버킷에 대해 Action'3:ListBucket"을 수행할 수 있는 권한 및 해당 버킷 내의 모든 개체에 대해 Action'3:GetObject'를 제공하는 전체 버킷 정책 설명을 보여줍니다.

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

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

관련 정보

테넌트 계정을 사용합니다

정책에 대한 정합성 보장 제어 설정입니다

기본적으로 그룹 정책에 대한 모든 업데이트는 최종적으로 일치합니다. 그룹 정책이 일관되면 정책 캐싱 때문에 변경 내용이 적용되는 데 15분 정도 더 걸릴 수 있습니다. 기본적으로 버킷 정책에 대한 모든 업데이트도 최종적으로 일치합니다.

필요에 따라 버킷 정책 업데이트의 일관성 보장을 변경할 수 있습니다. 예를 들어, 보안상의 이유로 버킷 정책을 최대한 빨리 변경할 수 있습니다.

이 경우 Put Bucket 정책 요청에서 정합성 보장 제어 헤더를 설정하거나 Put Bucket 정합성 보장 요청을 사용할 수 있습니다. 이 요청에 대한 정합성 제어를 변경할 때는 읽기 후 쓰기 정합성을 보장하는 * All * 값을 사용해야 합니다. Put Bucket 정합성 보장 요청의 헤더에 다른 정합성 보장 제어 값을 지정하면 요청이 거부됩니다. Put Bucket 정책 요청에 대해 다른 값을 지정하면 값이 무시됩니다. 버킷 정책이 일관되면 정책 캐싱으로 인해 변경 사항이 적용되는 데 8초가 더 걸릴 수 있습니다.

> 정합성 수준을 * All * 로 설정하면 새 버킷 정책이 더 빨리 발효되도록 하려면 작업이 완료되면 버킷 수준 제어를 원래 값으로 다시 설정해야 합니다. 그렇지 않으면 이후의 모든 버킷 요청은 * All * 설정을 사용합니다.

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"

  Put Bucket 정책 작업의 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 '리소스' 또는 'NotResource'(제외)로 표시됩니다.
- 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.

관련 정보

정책에 변수를 지정합니다

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 교장은 제외에 대해 "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, 알렉스가 퇴사하고 사용자 이름 알렉스가 삭제되었다고 가정해 봅시다. 새 알렉스가 조직에 가입하여 동일한 '알렉스' 사용자 이름을 할당하면 새 사용자는 원래 사용자에게 부여된 권한을 의도치 않게 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.

Amazon S3는 이제 PUT 및 DELETE Bucket 복제 작업 모두에 S3:PutReplicationConfiguration 권한을 사용합니다. StorageGRID는 원래 Amazon S3 사양과 일치하는 각 작업에 대해 별도의 권한을 사용합니다.

기존 값을 덮어쓰는 데 PUT를 사용할 때 삭제가 수행됩니다.

버킷에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:생성 버킷	버킷 을 놓습니다	
S3:삭제 버킷	버킷 삭제	
S3:DeleteBucketMetadataNotification	버킷 메타데이터 알림 구성을 삭제합니다	예
S3:삭제 BucketPolicy	버킷 정책을 삭제합니다	
S3:DeleteReplicationConfiguration	버킷 복제를 삭제합니다	예, PUT 및 DELETE에 대한 별도의 권한 *
S3:GetBuckketAcl	버킷 ACL 가져오기	
S3:GetBucketCompliance	버킷 규정 준수 가져오기(더 이상 사용되지 않음)	예
S3:GetBuckketConsistency	버킷 일관성 확보	예

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:GetBuckketCORS	버킷 CORS를 가져옵니다	
S3:GetEncryptionConfiguration	버킷 암호화 가져오기	
S3:GetBucketLastAccessTime	버킷 최종 액세스 시간 가져오기	예
S3:GetBucketLocation	버킷 위치를 가져옵니다	
S3:GetBuckMetadataNotification 을 참조하십시오	Bucket 메타데이터 알림 구성 가져오기	예
S3:GetBucketNotification 을 참조하십시오	버킷 알림을 받습니다	
S3:GetBuckketObjectLockConfiguration	개체 잠금 구성을 가져옵니다	
S3:GetBuckketPolicy를 참조하십시오	버킷 정책 가져오기	
S3:GetBucketTagging	버킷 태그 지정을 가져옵니다	
S3:GetBucketVersioning	버킷 버전 관리 가져오기	
S3:GetLifecycleConfiguration	버킷 수명 주기 가져오기	
S3:GetReplicationConfiguration	버킷 복제를 가져옵니다	
S3:ListAllMyBucket	• 서비스 받기 • 스토리지 사용량을 가져옵니다	예, 스토리지 사용량 가져오에 대해 가능합니다
S3:목록 버킷	• 버킷 가져오기(객체 나열) • 헤드 버킷 • 사후 개체 복원	
S3:ListBuckketMultipartUploads	• 다중 파트 업로드 나열 • 사후 개체 복원	
S3:목록 BuckketVersions	버킷 버전 가져오기	
S3: PutBucketCompliance	버킷 규정 준수(폐기됨)	예

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3: PutBucketConsistency	버킷 일관성을 유지합니다	예
S3: PutBucketCORS	• 버킷 CORS† 삭제 • 버킷 CORS를 넣습니다	
S3:PutEncryptionConfiguration	• Bucket 암호화를 삭제합니다 • Bucket 암호화를 적용합니다	
S3:PutBucketLastAccessTime	버킷 최종 접근 시간	예
S3:PutBucketMetadataNotification	Put Bucket 메타데이터 알림 구성	예
S3: PutBucketNotification	버킷 통지를 보냅니다	
S3:PutBucketObjectLockConfiguration	• "x-amz-bucket-object-lock-enabled: true" 요청 헤더가 있는 Bucket을 배치합니다(또한 S3:CreateBucket 권한이 필요함). • 개체 잠금 구성을 배치합니다	
S3: PutBucketPolicy	버킷 정책을 적용합니다	
S3: PutBucketTagging	• 버킷 태그 표시 삭제† • Bucket 태그 달기	
S3: PutBucketVersioning	버킷 버전 관리	
S3: PutLifecycleConfiguration	• 버킷 수명 주기 삭제† • 버킷 수명 주기를 놓습니다	
S3:PutReplicationConfiguration	버킷 복제를 배치합니다	예, PUT 및 DELETE에 대한 별도의 권한 *

객체에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:중단멀티업로드입니다	• 멀티파트 업로드를 중단합니다 • 사후 개체 복원	

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:DeleteObject 를 선택합니다	• 개체 삭제 • 여러 개체를 삭제합니다 • 사후 개체 복원	
S3:삭제 ObjectTagging	개체 태그 지정을 삭제합니다	
S3:DeleteObjectVersionTagging	개체 태그 지정 삭제(개체의 특정 버전)	
S3:DeleteObjectVersion	개체 삭제(개체의 특정 버전)	
S3:GetObject	• 객체 가져오기 • 헤드 개체 • 사후 개체 복원 • 개체 내용 을 선택합니다	
S3:GetObjectAcl	객체 ACL을 가져옵니다	
S3:GetObjectLegalHold	객체 법적 증거 자료 보관	
S3:GetObjectRetention	개체 보존 가져오기	
S3:GetObjectTagging	개체 태그 지정을 가져옵니다	
S3:GetObjectVersionTagging	개체 태그 지정 가져오기(개체의 특정 버전)	
S3:GetObjectVersion	개체 가져오기(개체의 특정 버전)	
S3:ListMultipartUploadParts(S3:ListMultimpartUploadParts) 를	부품 나열, POST 개체 복원	
S3:PutObject	• 개체 를 넣습니다 • 개체 - 복사 를 선택합니다 • 사후 개체 복원 • 멀티파트 업로드를 시작합니다 • 멀티파트 업로드를 완료합니다 • 부품 업로드 • 업로드 부품 - 복사	

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:PutObjectLegalHold	개체를 법적 증거 자료 보관	
S3:PutObjectRetention	개체 보존	
S3:PutObjectTagging	개체 태깅을 넣습니다	
S3:PutObjectVersionTagging	개체 태그 지정(개체의 특정 버전)	
S3:PutOverwriteObject	• 개체 를 넣습니다 • 개체 - 복사 를 선택합니다 • 개체 태그 지정 • 개체 태그 지정 삭제 • 멀티파트 업로드를 완료합니다	예
S3:RestoreObject	사후 개체 복원	

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 관리가 활성화된 경우 거부 설정을 사용하면 개체 태그 지정 또는 개체 삭제 태그 지정 작업에서 개체 및 현재 버전이 아닌 개체의 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 * > * 시스템 * > * 그리드 옵션 *) 해당 설정은 PutOverwriteObject 권한 설정을 재정의합니다.

관련 정보

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"
        ...
},
        ...
```

지원되는 조건 연산자

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).

조건 연산자	설명
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	키를 ""보다 큼"" 일치를 기준으로 숫자 값과 비교합니다.
NumericGreaterThanEquals	키를 ""크거나 같음"" 일치를 기준으로 숫자 값과 비교합니다.
NumericLessThan	""보다 작음" 일치를 기준으로 키를 숫자 값과 비교합니다.
NumericLessThanEquals	키를 ""보다 작음 또는 같음" 일치를 기준으로 숫자 값과 비교합니다.
불입니다	"true 또는 false" 일치를 기준으로 키를 부울 값과 비교합니다.
IP 주소	키를 IP 주소 또는 IP 주소 범위와 비교합니다.
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

범주	적용 가능한 조건 키	설명
IP 연산자	AWS: SOURCEIP	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. X-Forwarded-For 헤더는 유효성을 확인할 수 없기 때문에 무시됩니다.
리소스/ID입니다	AWS: 사용자 이름	요청이 전송된 보낸 사람의 사용자 이름과 비교합니다. 버킷 또는 오브젝트 작업에 사용할 수 있습니다.
S3:ListBucket 및 S3: ListBuckketVersions 권한	S3: 구분 기호	버킷 가져오기 또는 버킷 오브젝트 버전 가져오기 요청에 지정된 구분 기호 매개변수와 비교합니다.
S3:ListBucket 및 S3: ListBuckketVersions 권한	S3: 최대 키	버킷 가져오기 또는 버킷 객체 버전 가져오기 요청에 지정된 최대 키 매개변수와 비교합니다.
S3:ListBucket 및 S3: ListBuckketVersions 권한	S3: 접두어	Get Bucket 또는 Get Bucket Object Versions 요청에 지정된 접두사 매개변수와 비교합니다.
S3:PutObject	S3: 오브젝트 잠금 장치 - 남은 보존 기간(일)	'x-amz-object-lock-retain-until-date' 요청 헤더에 지정된 보존 기한 또는 버킷 기본 보존 기간(bucket default retention period)에서 계산된 보존 기한(retain-until-date)과 비교하여 이러한 값이 다음 요청에 대해 허용 가능한 범위 내에 있는지 확인합니다. • 개체 를 넣습니다 • 개체 - 복사 를 선택합니다 • 멀티파트 업로드를 시작합니다

범주	적용 가능한 조건 키	설명
S3:PutObjectRetention	S3: 오브젝트 잠금 장치 - 남은 보존 기간(일)	허용 범위 내에 있는지 확인하기 위해 Put Object Retention 요청에 지정된 Retain-until-date와 비교합니다.

정책에 변수를 지정합니다

정책의 변수를 사용하여 사용 가능한 정책 정보를 채울 수 있습니다. 'Resource' 요소와 'Condition' 요소의 문자열 비교에 정책 변수를 사용할 수 있습니다.

이 예제에서 변수 "${AWS:UserName}"은(는) Resource 요소의 일부입니다.

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

이 예제에서 변수 "${AWS:username}"은 조건 블록의 조건 값의 일부입니다.

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

변수	설명
'${AWS:SOURCEIP}'	SOURCEIP 키를 제공된 변수로 사용합니다.
'${AWS:username}'	제공된 변수로 사용자 이름 키를 사용합니다.
'${s3:prefix}'	서비스별 prefix key를 제공된 variable 로 사용한다.
'${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 * > * 시스템 * > * 그리드 옵션 * 으로 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 오브젝트 허용(Put Object Allow) 작업을 추가합니다.

> S3 정책에서 DeleteObject 를 deny 로 설정해도 ""30일 후 0개 복사본""과 같은 규칙이 있을 때 ILM이 개체를 삭제하는 것을 차단하지 않습니다.

> 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

ILM을 사용하여 개체를 관리합니다

특별한 처리가 필요한 정책을 생성합니다

StorageGRID ILM 규칙이 개체를 관리하는 방법

S3 그룹 정책의 예

S3 정책 예

이 섹션의 예를 사용하여 버킷 및 그룹에 대한 StorageGRID 액세스 정책을 구축합니다.

S3 버킷 정책의 예

버킷 정책은 정책이 연결된 버킷에 대한 액세스 권한을 지정합니다. 버킷 정책은 S3 PutBucketPolicy API를 사용하여 구성됩니다.

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 anonymous 를 비롯한 모든 사람이 버킷에 있는 오브젝트를 나열하고 버킷에 있는 모든 오브젝트에 대해 오브젝트 가져오기 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 '공유/' 개체 키 접두사로 시작하는 버킷의 개체에 대해 버킷을 나열하고 GetObject 작업을 수행할 수만 있습니다.

> 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 예에서는 anonymous를 비롯한 모든 사용자가 버킷을 나열하고 버킷의 모든 오브젝트에 대해 오브젝트 가져오기 작업을 수행할 수 있지만 지정된 계정의 MMarketing 그룹에 속한 사용자만 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 'examplebucket' 버킷과 그 객체에 대한 전체 액세스를 허용합니다. "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 예제에서 PutOverwriteObject 및 DeleteObject 에 대한 Deny 효과를 사용하면 개체의 데이터, 사용자 정의 메타데이터 및 S3 개체 태그를 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/*"
    }
  ]
}
```

관련 정보

버킷 작업

S3 그룹 정책의 예

그룹 정책은 정책이 연결된 그룹에 대한 액세스 권한을 지정합니다. 이 정책에는 주체가 없는 것은 암묵적인 것이기 때문입니다. 그룹 정책은 테넌트 관리자 또는 API를 사용하여 구성됩니다.

예: 테넌트 관리자를 사용하여 그룹 정책을 설정합니다

테넌트 관리자를 사용하여 그룹을 추가 또는 편집할 때 이 그룹의 S3 액세스 권한 구성원이 가질 그룹 정책을 생성하는

방법을 다음과 같이 선택할 수 있습니다.

- * S3 액세스 없음 *: 기본 옵션. 이 그룹의 사용자는 버킷 정책을 통해 액세스가 부여되지 않는 한 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 문자열은 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 문자열은 편집할 수 없습니다.
- * 사용자 정의 *: 그룹의 사용자에게는 텍스트 상자에 지정한 사용 권한이 부여됩니다.

  이 예제에서 그룹의 구성원은 지정된 버킷의 특정 폴더(키 접두사)를 나열하고 액세스할 수만 있습니다.

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 "" 폴더에만 완전히 액세스할 수 있도록 허용합니다

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}/*"
    }
  ]
}
```

관련 정보

테넌트 계정을 사용합니다

REST API에 대한 보안을 구성합니다

REST API에 대해 구현된 보안 조치를 검토하고 시스템 보안 방법을 이해해야 합니다.

StorageGRID에서 REST API에 대한 보안을 제공하는 방법

StorageGRID 시스템이 REST API에 대한 보안, 인증 및 권한 부여를 구현하는 방법을 이해해야 합니다.

StorageGRID는 다음과 같은 보안 조치를 사용합니다.

- 로드 밸런서 끝점에 HTTPS가 구성되어 있는 경우 로드 밸런서 서비스와의 클라이언트 통신은 HTTPS를 사용합니다.

  로드 밸런서 끝점을 구성할 때 HTTP를 선택적으로 활성화할 수 있습니다. 예를 들어, 테스트 또는 기타 비운영 목적으로 HTTP를 사용할 수 있습니다. 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

- 기본적으로 StorageGRID는 게이트웨이 노드에서 스토리지 노드 및 CLB 서비스와의 클라이언트 통신에 HTTPS를 사용합니다.

  이러한 연결에 대해 HTTP를 선택적으로 활성화할 수 있습니다. 예를 들어, 테스트 또는 기타 비운영 목적으로 HTTP를 사용할 수 있습니다. 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

CLB 서비스는 더 이상 사용되지 않습니다.

- StorageGRID와 클라이언트 간의 통신은 TLS를 사용하여 암호화됩니다.
- 로드 밸런서 끝점이 HTTP 또는 HTTPS 연결을 허용하도록 구성되었는지 여부에 관계없이 그리드 내의 로드 밸런서 서비스와 스토리지 노드 간의 통신이 암호화됩니다.
- 클라이언트는 REST API 작업을 수행하기 위해 StorageGRID에 HTTP 인증 헤더를 제공해야 합니다.

보안 인증서 및 클라이언트 응용 프로그램

클라이언트는 게이트웨이 노드 또는 관리 노드의 로드 밸런서 서비스, 스토리지 노드 또는 게이트웨이 노드의 CLB 서비스에 직접 연결할 수 있습니다.

모든 경우에 클라이언트 응용 프로그램은 그리드 관리자가 업로드한 사용자 지정 서버 인증서 또는 StorageGRID 시스템에서 생성한 인증서를 사용하여 TLS 연결을 만들 수 있습니다.

- 클라이언트 응용 프로그램이 로드 밸런서 서비스에 연결되면 연결을 만드는 데 사용되는 특정 로드 밸런서 끝점에 대해 구성된 인증서를 사용합니다. 각 끝점마다 고유한 인증서가 있습니다. 이 인증서는 그리드 관리자가 업로드한 사용자 지정 서버 인증서이거나, 끝점 구성 시 그리드 관리자가 StorageGRID에서 생성한 인증서입니다.
- 클라이언트 응용 프로그램이 게이트웨이 노드의 스토리지 노드 또는 CLB 서비스에 직접 연결할 때 StorageGRID 시스템이 설치될 때 스토리지 노드에 대해 생성된 시스템 생성 서버 인증서(시스템 인증 기관이 서명)를 사용합니다. 또는 그리드 관리자가 그리드에 제공하는 단일 사용자 정의 서버 인증서입니다.

클라이언트가 TLS 연결을 설정하는 데 사용하는 인증서를 신뢰하도록 구성해야 합니다.

로드 밸런서 끝점 구성에 대한 정보와 TLS 연결에 대한 단일 사용자 지정 서버 인증서를 스토리지 노드 또는 게이트웨이 노드의 CLB 서비스에 직접 추가하는 방법에 대한 지침은 StorageGRID 관리 지침을 참조하십시오.

요약

다음 표에서는 S3 및 Swift REST API에서 보안 문제가 구현되는 방식을 보여 줍니다.

보안 문제	**REST API** 구현
연결 보안	TLS
서버 인증	시스템 CA에서 서명한 X.509 서버 인증서 또는 관리자가 제공한 사용자 지정 서버 인증서입니다
클라이언트 인증	• S3:S3 계정(액세스 키 ID 및 비밀 액세스 키) • Swift:Swift 계정(사용자 이름 및 암호)
클라이언트 인증	• S3: 버킷 소유권 및 모든 적용 가능한 액세스 제어 정책 • Swift: 관리자 역할 액세스

관련 정보

StorageGRID 관리

TLS 라이브러리에 대해 지원되는 해시 및 암호화 알고리즘

StorageGRID 시스템은 TLS(전송 계층 보안) 세션을 설정할 때 클라이언트 응용 프로그램에서 사용할 수 있는 제한된 암호화 그룹 세트를 지원합니다.

지원되는 **TLS** 버전입니다

StorageGRID는 TLS 1.2 및 TLS 1.3을 지원합니다.

SSLv3 및 TLS 1.1(또는 이전 버전)은 더 이상 지원되지 않습니다.

지원되는 암호 그룹

TLS 버전입니다	암호화 그룹의 **IANA** 이름입니다
1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
1.2	TLS_ECDHE_RSA_WITH_CHACH20_POLY1305_SHA256
1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
1.3	TLS_AES_256_GCM_SHA384
1.3	TLS_CHACH20_POLY1305_SHA256
1.3	TLS_AES_128_GCM_SHA256

더 이상 사용되지 않는 암호화 그룹

다음 암호화 그룹은 더 이상 사용되지 않습니다. 이러한 암호화에 대한 지원은 이후 릴리스에서 제거됩니다.

IANA 이름입니다
TLS_RSA_with_AES_128_GCM_SHA256
TLS_RSA_WITED_AES_256_GCM_SHA384

관련 정보

클라이언트 연결 구성 방법

운영 모니터링 및 감사

전체 그리드 또는 특정 노드에 대한 트랜잭션 추세를 확인하여 클라이언트 작업의 워크로드 및 효율성을 모니터링할 수 있습니다. 감사 메시지를 사용하여 클라이언트 작업 및 트랜잭션을 모니터링할 수 있습니다.

오브젝트 수집 및 검색 속도와 오브젝트 수, 쿼리, 검증을 위한 메트릭을 모니터링할 수 있습니다. 클라이언트 응용 프로그램이 StorageGRID 시스템에서 개체를 읽고, 쓰고, 수정하는 데 성공한 시도 및 실패한 시도 횟수를 볼 수 있습니다.

단계

1. 를 사용하여 Grid Manager에 로그인합니다 지원되는 웹 브라우저.
2. Dashboard에서 Protocol Operations 섹션을 찾습니다.

   이 섹션에서는 StorageGRID 시스템에서 수행하는 클라이언트 작업의 수를 요약합니다. 프로토콜 속도는 최근 2분 동안의 평균값입니다.

3. 노드 * 를 선택합니다.
4. 노드 홈 페이지(배포 수준)에서 * 로드 밸런서 * 탭을 클릭합니다.

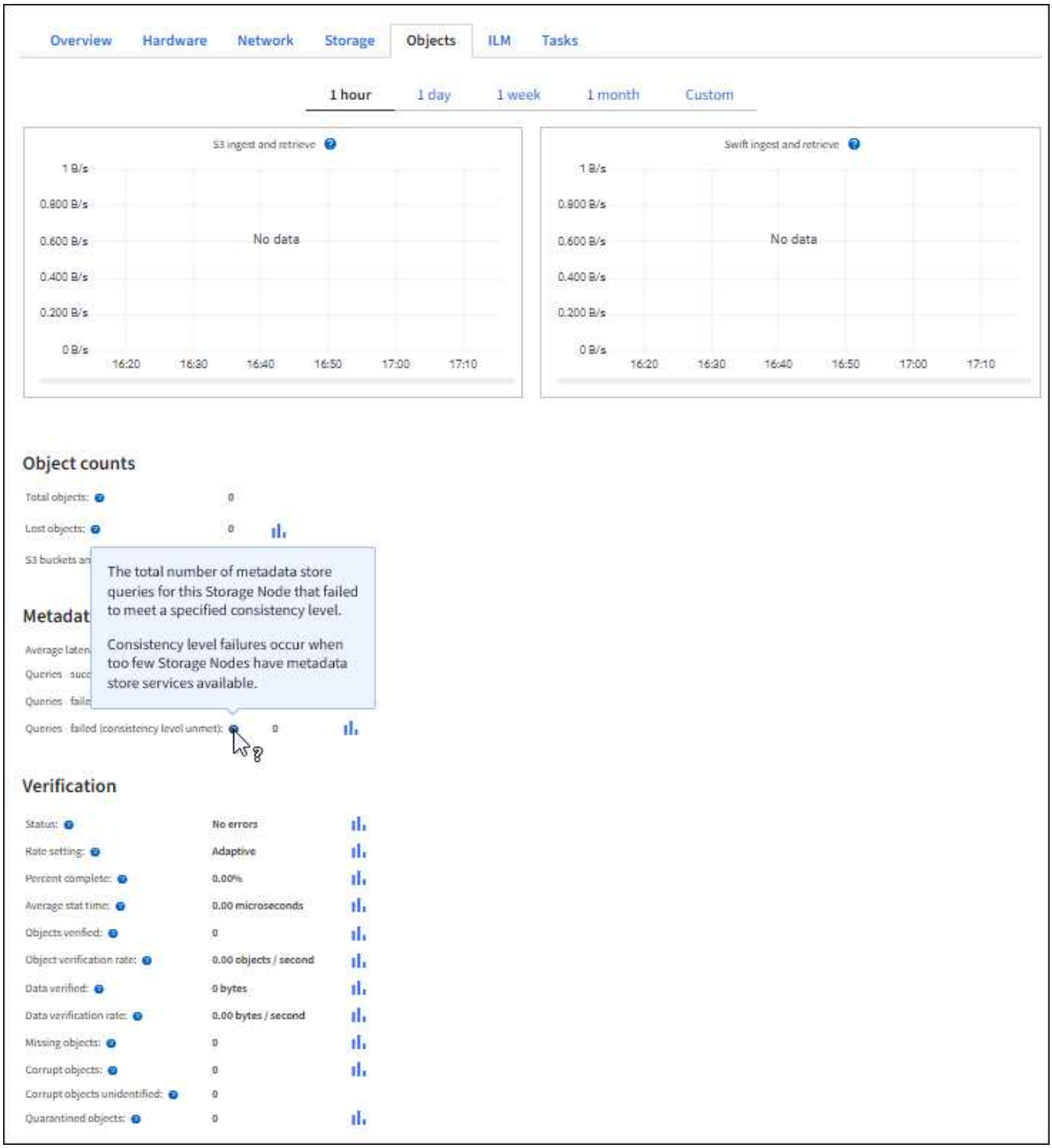
   차트에는 그리드 내의 로드 밸런서 끝점에 대한 모든 클라이언트 트래픽에 대한 추세가 표시됩니다. 시간 간격(시간, 일, 주, 월 또는 년)을 선택할 수 있습니다. 또는 사용자 지정 간격을 적용할 수 있습니다.

5. 노드 홈 페이지(배포 수준)에서 * 개체 * 탭을 클릭합니다.

   이 차트에는 전체 StorageGRID 시스템의 수집 및 검색 속도가 초당 바이트 및 총 바이트 단위로 표시됩니다. 시간 간격(시간, 일, 주, 월 또는 년)을 선택할 수 있습니다. 또는 사용자 지정 간격을 적용할 수 있습니다.

6. 특정 스토리지 노드에 대한 정보를 보려면 왼쪽의 목록에서 노드를 선택하고 * Objects * 탭을 클릭합니다.

   이 차트에는 이 스토리지 노드의 객체 수집 및 검색 속도가 나와 있습니다. 이 탭에는 개체 수, 쿼리 및 검증에 대한 메트릭도 포함되어 있습니다. 레이블을 클릭하여 이러한 메트릭의 정의를 볼 수 있습니다.



7. 더 자세한 내용을 원하는 경우:
   a. 지원 * > * 도구 * > * 그리드 토폴로지 * 를 선택합니다.
   b. site_ * > * Overview * > * Main * 을 선택합니다.

      API 작업 섹션에는 전체 그리드에 대한 요약 정보가 표시됩니다.

   c. 스토리지 노드 * > * LDR * > **CLIENT APPLICATION* * > * Overview * > * Main * 을 선택합니다

      작업 섹션에는 선택한 스토리지 노드에 대한 요약 정보가 표시됩니다.

감사 로그 액세스 및 검토

감사 메시지는 StorageGRID 서비스에서 생성되고 텍스트 로그 파일에 저장됩니다. 감사 로그의 API 관련 감사 메시지는 시스템의 상태를 평가하는 데 도움이 되는 중요한 보안, 운영 및 성능 모니터링 데이터를 제공합니다.

필요한 것

- 특정 액세스 권한이 있습니다.
- "passwords.txt" 파일이 있습니다.
- 관리 노드의 IP 주소를 알고 있습니다.

이 작업에 대해

활성 감사 로그 파일은 AUDIT.LOG라는 이름으로 관리 노드에 저장됩니다.

하루에 한 번 활성 audit.log 파일이 저장되고 새 감사.로그 파일이 시작됩니다. 저장된 파일의 이름은 저장 시기를 나타냅니다('*yyyy-mm-dd.txt*' 형식).

하루 후에는 원래 날짜를 유지하는 형식(*yyyy-mm-dd.txt.gz*)으로 저장된 파일이 압축되고 이름이 변경됩니다.

이 예에서는 액티브 'audit.log' 파일, 전날 파일('2018-04-15.txt') 및 전날 압축 파일('2018-04-14.txt.gz')을 보여줍니다.

```
audit.log
2018-04-15.txt
2018-04-14.txt.gz
```

단계

1. 관리자 노드에 로그인:
   a. 'ssh admin@*primary_Admin_Node_IP*' 명령을 입력합니다
   b. "passwords.txt" 파일에 나열된 암호를 입력합니다.
2. 감사 로그 파일이 포함된 디렉토리로 이동합니다.

   ```
   cd /var/local/audit/export
   ```

3. 필요에 따라 현재 또는 저장된 감사 로그 파일을 봅니다.

감사 로그에서 **S3** 작업을 추적했습니다

StorageGRID 감사 로그에서 여러 버킷 작업 및 오브젝트 작업을 추적합니다.

감사 로그에서 버킷 작업을 추적했습니다

- 버킷 삭제
- 버킷 태그 지정을 삭제합니다

- 여러 개체를 삭제합니다
- 버킷 가져오기(객체 나열)
- 버킷 객체 버전을 가져옵니다
- 버킷 태그 지정을 가져옵니다
- 헤드 버킷
- 버킷 을 놓습니다
- 버킷 규정 준수
- Bucket 태그 달기
- 버킷 버전 관리

감사 로그에서 추적된 객체 작업입니다

- 멀티파트 업로드를 완료합니다
- 파트 업로드(ILM 규칙이 Strict 또는 Balanced 수집 동작을 사용하는 경우)
- Upload Part-Copy (ILM 규칙이 Strict 또는 Balanced 수집 동작을 사용하는 경우)
- 개체 삭제
- 객체 가져오기
- 헤드 개체
- 사후 개체 복원
- 개체 를 넣습니다
- 개체 - 복사 를 선택합니다

관련 정보

버킷 작업

객체에 대한 작업

활성, 유휴 및 동시 **HTTP** 연결의 이점

HTTP 연결을 구성하는 방법은 StorageGRID 시스템의 성능에 영향을 줄 수 있습니다. 구성은 HTTP 연결이 활성 상태인지 유휴 상태인지 또는 여러 개의 동시 연결이 있는지 여부에 따라 달라집니다.

다음과 같은 유형의 HTTP 연결에 대한 성능 이점을 확인할 수 있습니다.

- 유휴 HTTP 연결
- 활성 HTTP 연결
- 동시 HTTP 연결

유휴 **HTTP** 연결을 열어 두면 얻을 수 있는 이점

클라이언트 응용 프로그램이 열려 있는 연결을 통해 후속 트랜잭션을 수행할 수 있도록 클라이언트 응용 프로그램이 유휴 상태인 경우에도 HTTP 연결을 열어 두어야 합니다. 시스템 측정 및 통합 경험을 바탕으로 유휴 HTTP 연결을 최대 10분 동안 열어 두어야 합니다. StorageGRID는 열려 있고 10분 이상 유휴 상태로 유지되는 HTTP 연결을 자동으로 닫을 수 있습니다.

개방 및 유휴 HTTP 연결은 다음과 같은 이점을 제공합니다.

- StorageGRID 시스템이 HTTP 트랜잭션을 수행해야 한다고 결정하는 시간부터 StorageGRID 시스템이 트랜잭션을 수행할 수 있는 시간까지 지연 시간을 줄였습니다

  지연 시간 감소는 특히 TCP/IP 및 TLS 연결을 설정하는 데 필요한 시간의 주요 장점입니다.

- 이전에 수행된 전송을 사용하여 TCP/IP 저속 시작 알고리즘을 프라이밍하여 데이터 전송 속도를 높였습니다
- 클라이언트 응용 프로그램과 StorageGRID 시스템 간의 연결을 중단하는 여러 가지 장애 조건에 대한 즉각적인 알림

유휴 연결을 유지하는 기간을 결정하는 것은 기존 연결과 관련된 느린 시작의 이점과 내부 시스템 리소스에 대한 연결의 이상적인 할당을 절충하는 것입니다.

활성 **HTTP** 연결의 이점

스토리지 노드 또는 게이트웨이 노드의 CLB 서비스(더 이상 사용되지 않음)에 직접 연결하는 경우 HTTP 연결이 지속적으로 트랜잭션을 수행하더라도 활성 HTTP 연결 기간을 최대 10분으로 제한해야 합니다.

연결을 열어 두어야 하는 최대 기간을 결정하는 것은 연결 지속성의 이점과 내부 시스템 리소스에 대한 연결을 이상적으로 할당하는 것입니다.

클라이언트가 스토리지 노드 또는 CLB 서비스에 접속할 경우 활성 HTTP 연결을 제한하면 다음과 같은 이점이 있습니다.

- StorageGRID 시스템 전체에서 최적의 로드 밸런싱을 지원합니다.

  CLB 서비스를 사용할 때는 오래 지속되는 TCP/IP 연결을 방지하여 StorageGRID 시스템 전체의 로드 밸런싱을 최적화해야 합니다. HTTP 연결을 다시 설정하고 재조정할 수 있도록 클라이언트 응용 프로그램을 구성하여 각 HTTP 연결 기간을 추적하고 설정된 시간 후에 HTTP 연결을 닫아야 합니다.

  CLB 서비스는 클라이언트 응용 프로그램이 HTTP 연결을 설정할 때 StorageGRID 시스템 전체의 로드 균형을 조정합니다. 시간이 지남에 따라 로드 밸런싱 요구 사항이 변경됨에 따라 HTTP 연결이 더 이상 최적화되지 않을 수 있습니다. 시스템은 클라이언트 애플리케이션이 각 트랜잭션에 대해 별도의 HTTP 연결을 설정할 때 최상의 로드 밸런싱을 수행하지만, 이 경우 영구 연결과 관련된 훨씬 더 가치 있는 이득을 얻을 수 없습니다.

  ⓘ CLB 서비스는 더 이상 사용되지 않습니다.

- 클라이언트 응용 프로그램이 사용 가능한 공간이 있는 LDR 서비스로 HTTP 트랜잭션을 보낼 수 있도록 합니다.
- 유지보수 절차를 시작할 수 있습니다.

일부 유지 관리 절차는 진행 중인 모든 HTTP 연결이 완료된 후에만 시작됩니다.

부하 분산 서비스에 대한 클라이언트 연결의 경우 일부 유지 관리 절차를 즉시 시작할 수 있도록 개방 연결 기간을 제한하는 것이 유용할 수 있습니다. 클라이언트 연결 기간이 제한되지 않으면 활성 연결이 자동으로 종료되는 데 몇 분이 걸릴 수 있습니다.

동시 **HTTP** 연결의 이점

병렬 처리를 허용하도록 StorageGRID 시스템에 대한 여러 TCP/IP 연결을 열린 상태로 유지하여 성능을 향상시켜야 합니다. 최적의 병렬 연결 수는 다양한 요인에 따라 달라집니다.

동시 HTTP 연결은 다음과 같은 이점을 제공합니다.

- 지연 시간 단축

  다른 트랜잭션이 완료될 때까지 기다리지 않고 즉시 트랜잭션을 시작할 수 있습니다.

- 처리량 향상

  StorageGRID 시스템은 병렬 트랜잭션을 수행하고 총 트랜잭션 처리량을 늘릴 수 있습니다.

클라이언트 응용 프로그램은 여러 HTTP 연결을 설정해야 합니다. 클라이언트 응용 프로그램은 트랜잭션을 수행해야 하는 경우 트랜잭션을 현재 처리하지 않는 설정된 연결을 선택하여 즉시 사용할 수 있습니다.

각 StorageGRID 시스템의 토폴로지에는 성능이 저하되기 전에 동시 트랜잭션 및 연결에 대해 서로 다른 최대 처리량이 있습니다. 최대 처리량은 컴퓨팅 리소스, 네트워크 리소스, 스토리지 리소스, WAN 링크 등의 요인에 따라 달라집니다. StorageGRID 시스템에서 지원하는 서버 및 서비스 수와 애플리케이션 수도 고려해야 합니다.

StorageGRID 시스템은 종종 여러 클라이언트 애플리케이션을 지원합니다. 클라이언트 응용 프로그램에서 사용하는 최대 동시 연결 수를 결정할 때 이 점에 유의해야 합니다. 클라이언트 응용 프로그램이 StorageGRID 시스템에 대한 연결을 설정하는 여러 소프트웨어 엔터티로 구성된 경우 엔터티에 대한 모든 연결을 추가해야 합니다. 다음과 같은 경우 최대 동시 연결 수를 조정해야 할 수 있습니다.

- StorageGRID 시스템의 토폴로지는 시스템에서 지원할 수 있는 최대 동시 트랜잭션 및 연결 수에 영향을 줍니다.
- 대역폭이 제한된 네트워크에서 StorageGRID 시스템과 상호 작용하는 클라이언트 응용 프로그램은 개별 트랜잭션이 적절한 시간 내에 완료되도록 동시성 정도를 줄여야 할 수 있습니다.
- 많은 클라이언트 응용 프로그램이 StorageGRID 시스템을 공유하는 경우 시스템의 제한을 초과하지 않도록 동시성 정도를 줄여야 할 수 있습니다.

읽기 및 쓰기 작업을 위한 **HTTP** 연결 풀 분리

읽기 및 쓰기 작업에 별도의 HTTP 연결 풀을 사용하고 각 풀에 사용할 풀 수를 제어할 수 있습니다. 별도의 HTTP 연결 풀을 통해 트랜잭션을 보다 효율적으로 제어하고 로드 밸런싱을 수행할 수 있습니다.

클라이언트 애플리케이션은 검색 가능(읽기) 또는 저장 가능(쓰기) 부하를 생성할 수 있습니다. 읽기 및 쓰기 트랜잭션을 위한 별도의 HTTP 연결 풀을 사용하여 읽기 또는 쓰기 트랜잭션에 사용할 각 풀의 양을 조정할 수 있습니다.

Swift를 사용합니다

Swift:개요 를 사용합니다

클라이언트 애플리케이션은 OpenStack Swift API를 사용하여 StorageGRID 시스템과 상호 작용할 수 있습니다.

StorageGRID는 다음과 같은 특정 버전의 Swift 및 HTTP를 지원합니다.

항목	버전
SwiFT 사양	2015년 11월 기준 OpenStack Swift Object Storage API v1
HTTP	1.1 HTTP에 대한 자세한 내용은 HTTP/1.1(RFC 7230-35)을 참조하십시오. • 참고 *: StorageGRID는 HTTP/1.1 파이프라이닝을 지원하지 않습니다.

관련 정보

"OpenStack: 오브젝트 스토리지 API"

StorageGRID의 **Swift API** 지원 기록

Swift REST API에 대한 StorageGRID 시스템의 지원 변경 사항을 숙지해야 합니다.

놓습니다	설명
11.6	편집상의 사소한 변경.
11.5	약한 일관성 제어 기능이 제거되었습니다. 대신 사용 가능한 정합성 보장 레벨이 사용됩니다.
11.4	TLS 1.3에 대한 지원 및 지원되는 TLS 암호 제품군의 업데이트된 목록이 추가되었습니다. CLB는 사용되지 않습니다. ILM과 정합성 보장 설정 간의 상호 관계에 대한 설명이 추가되었습니다.
11.3	수집 시 동기식 배치를 사용하는 ILM 규칙(Ingest 동작에 대한 균형 및 엄격 옵션)의 영향을 설명하기 위해 PUT 오브젝트 작업이 업데이트되었습니다. 로드 밸런서 끝점 또는 고가용성 그룹을 사용하는 클라이언트 연결에 대한 설명이 추가되었습니다. 지원되는 TLS 암호 그룹 목록이 업데이트되었습니다. TLS 1.1 암호가 더 이상 지원되지 않습니다.

놓습니다	설명
11.2	문서에 대한 사소한 편집 변경.
11.1	그리드 노드에 대한 Swift 클라이언트 연결에 HTTP 사용 지원이 추가되었습니다. 일관성 제어의 정의를 업데이트했습니다.
11.0	각 테넌트 계정에 대해 1,000개의 컨테이너에 대한 지원이 추가되었습니다.
10.3	문서의 관리 업데이트 및 수정. 사용자 지정 서버 인증서를 구성하기 위한 섹션이 제거되었습니다.
10.2	StorageGRID 시스템에서 Swift API의 초기 지원 현재 지원되는 버전은 OpenStack Swift Object Storage API v1 입니다.

StorageGRID가 **Swift REST API**를 구현하는 방법

클라이언트 애플리케이션은 Swift REST API 호출을 사용하여 스토리지 노드 및 게이트웨이 노드에 연결하여 컨테이너를 생성하고 오브젝트를 저장 및 검색할 수 있습니다. 이를 통해 OpenStack Swift용으로 개발된 서비스 중심 애플리케이션을 StorageGRID 시스템에서 제공하는 사내 오브젝트 스토리지에 연결할 수 있습니다.

Swift 오브젝트 관리

Swift 객체가 StorageGRID 시스템에서 수집되면 시스템의 활성 ILM 정책에 있는 ILM(정보 수명 주기 관리) 규칙에 의해 관리됩니다. ILM 규칙 및 정책은 StorageGRID이 오브젝트 데이터 복사본을 만들고 배포하는 방법과 시간이 지남에 따라 이러한 복사본을 관리하는 방법을 결정합니다. 예를 들어, ILM 규칙은 특정 Swift 컨테이너의 개체에 적용될 수 있으며 특정 기간 동안 여러 개체 복사본을 여러 데이터 센터에 저장하도록 지정할 수 있습니다.

그리드의 ILM 규칙 및 정책이 Swift 테넌트 계정의 개체에 어떤 영향을 미치는지 알아야 하는 경우 StorageGRID 관리자에게 문의하십시오.

클라이언트 요청 충돌

동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "최신" 평가 시기는 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 하며, Swift 클라이언트가 작업을 시작하는 시점이 아닙니다.

일관성 보장 및 제어

기본적으로 StorageGRID는 새로 생성된 객체에 대해 읽기 후 쓰기 정합성을 보장하고 객체 업데이트 및 헤드 작업에 대한 최종 일관성을 제공합니다. 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.

또한 StorageGRID를 사용하면 컨테이너 단위로 일관성을 제어할 수 있습니다. 애플리케이션의 필요에 따라 일관성 제어를 변경하여 오브젝트의 가용성과 서로 다른 스토리지 노드 및 사이트 전체에서 오브젝트의 일관성 간의 균형을

유지할 수 있습니다.

관련 정보

ILM을 사용하여 개체를 관리합니다

컨테이너 일관성 요청 가져오기

컨테이너 일관성 요청

Swift REST API 구축을 위한 권장 사항

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

존재하지 않는 객체에 대한 헤드 권장 사항

응용 프로그램에서 개체가 실제로 존재하지 않을 것으로 예상되는 경로에 개체가 있는지 정기적으로 확인하는 경우 ""사용 가능한"" 일관성 제어를 사용해야 합니다. 예를 들어, 애플리케이션에서 해당 위치에 대한 PUT 작업을 수행하기 전에 헤드 작업을 수행하는 경우 ""사용 가능" 정합성 제어를 사용해야 합니다.

그렇지 않으면 헤드 작업에서 개체를 찾지 못할 경우 하나 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다.

Put container consistency request를 사용하여 각 컨테이너에 대해 ""사용 가능"" 정합성 제어를 설정할 수 있습니다.

오브젝트 이름에 대한 권장사항

StorageGRID 11.4 이상에서 생성된 컨테이너의 경우 성능 모범 사례에 맞게 개체 이름을 제한할 필요가 없습니다. 예를 들어, 이제 개체 이름의 처음 4개 문자에 임의의 값을 사용할 수 있습니다.

StorageGRID 11.4 이전 릴리즈에서 만든 컨테이너의 경우 개체 이름에 대한 다음 권장 사항을 계속 따릅니다.

- 개체 이름의 처음 네 문자로 임의 값을 사용하면 안 됩니다. 이는 이전 AWS에서 권장하는 이름 접두사와 다릅니다. 대신 "이미지"와 같은 비무작위, 고유하지 않은 접두사를 사용해야 합니다.
- 이전 AWS 권장 사항에 따라 이름 접두사에 랜덤 및 고유 문자를 사용하려면 오브젝트 이름에 디렉토리 이름을 접두사로 붙여야 합니다. 즉, 다음 형식을 사용합니다.

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

  이 형식 대신:

  ```
  mycontainer/f8e3-image3132.jpg
  ```

""범위 읽기" 권장 사항

저장된 객체 압축 * 옵션을 선택한 경우(* 구성 * > * 시스템 * > * 그리드 옵션 *) Swift 클라이언트 응용 프로그램은 바이트 범위를 지정하는 객체 가져오기 작업을 수행하지 않아야 합니다. 이러한 ""범위 읽기"" 작업은 StorageGRID가 요청된 바이트에 액세스하기 위해 개체의 압축을 효과적으로 해제해야 하기 때문에 비효율적입니다. 매우 큰 개체에서 작은 바이트 범위를 요청하는 Get Object 작업은 특히 비효율적입니다. 예를 들어, 50GB의 압축 개체에서 10MB

범위를 읽는 것은 매우 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.

관련 정보

컨테이너 일관성 요청 가져오기

컨테이너 일관성 요청

StorageGRID 관리

테넌트 계정 및 연결을 구성합니다

클라이언트 응용 프로그램에서 연결을 허용하도록 StorageGRID를 구성하려면 하나 이상의 테넌트 계정을 만들고 연결을 설정해야 합니다.

Swift 테넌트 계정을 생성하고 구성합니다

Swift API 클라이언트가 StorageGRID에 객체를 저장하고 검색하기 전에 Swift 테넌트 계정이 필요합니다. 각 테넌트 계정에는 고유한 계정 ID, 그룹 및 사용자, 컨테이너 및 객체가 있습니다.

Swift 테넌트 계정은 그리드 관리자 또는 그리드 관리 API를 사용하여 StorageGRID 그리드 관리자가 만듭니다.

Swift 테넌트 계정을 생성할 때 그리드 관리자는 다음 정보를 지정합니다.

- 테넌트의 표시 이름(테넌트의 계정 ID가 자동으로 할당되며 변경할 수 없음)
- 필요한 경우 테넌트 계정의 스토리지 할당량 — 테넌트의 객체에 사용할 수 있는 최대 GB, 테라바이트 또는 PB입니다. 테넌트의 스토리지 할당량은 물리적 크기(디스크 크기)가 아닌 논리적 양(오브젝트 크기)을 나타냅니다.
- StorageGRID 시스템에서 SSO(Single Sign-On)를 사용하지 않는 경우 테넌트 계정이 자체 ID 소스를 사용할지 또는 그리드의 ID 소스를 공유할지 여부 및 테넌트의 로컬 루트 사용자의 초기 암호를 공유할지 여부
- SSO가 설정된 경우 테넌트 계정을 구성할 수 있는 루트 액세스 권한이 있는 통합 그룹이 있습니다.

Swift 테넌트 계정이 생성된 후 루트 액세스 권한이 있는 사용자는 테넌트 관리자에 액세스하여 다음과 같은 작업을 수행할 수 있습니다.

- ID 통합을 설정하고(ID 소스를 그리드와 공유하지 않는 경우) 로컬 그룹 및 사용자를 만듭니다
- 스토리지 사용량 모니터링

> Swift 사용자는 테넌트 관리자에 액세스하려면 루트 액세스 권한이 있어야 합니다. 그러나 루트 액세스 권한은 사용자가 Swift REST API에 인증하여 컨테이너를 생성하고 객체를 수집하는 것을 허용하지 않습니다. 사용자는 Swift REST API에 인증할 수 있는 Swift 관리자 권한이 있어야 합니다.

관련 정보

StorageGRID 관리

테넌트 계정을 사용합니다

지원되는 Swift API 엔드포인트

클라이언트 연결 구성 방법

그리드 관리자는 Swift 클라이언트가 StorageGRID에 연결하여 데이터를 저장 및 검색하는 방법에 영향을 주는 구성을 선택합니다. 연결에 필요한 특정 정보는 선택한 구성에 따라 다릅니다.

클라이언트 응용 프로그램은 다음 중 하나를 연결하여 개체를 저장하거나 검색할 수 있습니다.

- 관리 노드 또는 게이트웨이 노드의 부하 분산 서비스 또는 선택적으로 관리 노드 또는 게이트웨이 노드의 고가용성(HA) 그룹의 가상 IP 주소입니다
- 게이트웨이 노드의 CLB 서비스 또는 게이트웨이 노드의 고가용성 그룹의 가상 IP 주소(선택 사항)입니다

> CLB 서비스는 더 이상 사용되지 않습니다. StorageGRID 11.3 릴리스 전에 구성된 클라이언트는 게이트웨이 노드에서 CLB 서비스를 계속 사용할 수 있습니다. 로드 밸런싱을 제공하기 위해 StorageGRID에 의존하는 다른 모든 클라이언트 애플리케이션은 로드 밸런서 서비스를 사용하여 연결해야 합니다.

- 외부 로드 밸런서가 있거나 없는 스토리지 노드

StorageGRID를 구성할 때 그리드 관리자는 그리드 관리자 또는 그리드 관리 API를 사용하여 다음 단계를 수행할 수 있습니다. 이 모든 단계는 선택 사항입니다.

1. 로드 밸런서 서비스의 끝점을 구성합니다.

   로드 밸런서 서비스를 사용하려면 끝점을 구성해야 합니다. 관리 노드 또는 게이트웨이 노드의 부하 분산 서비스는 들어오는 네트워크 연결을 클라이언트 애플리케이션에서 스토리지 노드로 분산합니다. 로드 밸런서 끝점을 만들 때 StorageGRID 관리자는 포트 번호, 엔드포인트가 HTTP 또는 HTTPS 연결을 수락하는지 여부, 엔드포인트를 사용할 클라이언트 유형(S3 또는 Swift) 및 HTTPS 연결에 사용할 인증서(해당하는 경우)를 지정합니다.

2. 신뢰할 수 없는 클라이언트 네트워크를 구성합니다.

   StorageGRID 관리자가 노드의 클라이언트 네트워크를 신뢰할 수 없도록 구성하는 경우 노드는 로드 밸런서 끝점으로 명시적으로 구성된 포트의 클라이언트 네트워크에서 인바운드 연결만 허용합니다.

3. 고가용성 그룹을 구성합니다.

   관리자가 HA 그룹을 생성하면 여러 관리 노드 또는 게이트웨이 노드의 네트워크 인터페이스가 액티브-백업 구성에 배치됩니다. HA 그룹의 가상 IP 주소를 사용하여 클라이언트 연결이 이루어집니다.

각 옵션에 대한 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

요약: 클라이언트 연결을 위한 **IP** 주소 및 포트

클라이언트 애플리케이션은 그리드 노드의 IP 주소와 해당 노드의 서비스 포트 번호를 사용하여 StorageGRID에 접속합니다. HA(고가용성) 그룹이 구성되어 있는 경우 클라이언트 애플리케이션은 HA 그룹의 가상 IP 주소를 사용하여 연결할 수 있습니다.

클라이언트 연결을 만드는 데 필요한 정보입니다

이 표에는 클라이언트가 StorageGRID에 연결할 수 있는 다양한 방법과 각 연결 유형에 사용되는 IP 주소 및 포트가 요약되어 있습니다. 자세한 내용은 StorageGRID 관리자에게 문의하거나 StorageGRID 관리 지침 에서 그리드 관리자에서 이 정보를 찾는 방법에 대한 설명을 참조하십시오.

연결 위치	클라이언트가 연결하는 서비스입니다	**IP** 주소입니다	포트
HA 그룹	로드 밸런서	HA 그룹의 가상 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
HA 그룹	CLB • 참고: * CLB 서비스는 더 이상 사용되지 않습니다.	HA 그룹의 가상 IP 주소입니다	기본 Swift 포트: • HTTPS: 8083 • HTTP: 8085
관리자 노드	로드 밸런서	관리 노드의 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
게이트웨이 노드	로드 밸런서	게이트웨이 노드의 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
게이트웨이 노드	CLB • 참고: * CLB 서비스는 더 이상 사용되지 않습니다.	게이트웨이 노드의 IP 주소입니다 • 참고: * 기본적으로 CLB 및 LDR용 HTTP 포트는 사용되지 않습니다.	기본 Swift 포트: • HTTPS: 8083 • HTTP: 8085
스토리지 노드	LDR	스토리지 노드의 IP 주소입니다	기본 Swift 포트: • HTTPS: 18083 • HTTP: 18085

예

Swift 클라이언트를 게이트웨이 노드 HA 그룹의 로드 밸런서 끝점에 연결하려면 아래와 같이 구조화된 URL을 사용합니다.

- "https://*VIP-of-HA-group:LB-endpoint-port*`

예를 들어 HA 그룹의 가상 IP 주소가 192.0.2.6이고 Swift 로드 밸런서 끝점의 포트 번호가 10444인 경우 Swift 클라이언트는 다음 URL을 사용하여 StorageGRID에 연결할 수 있습니다.

- 'https://192.0.2.6:10444`

클라이언트가 StorageGRID에 연결하는 데 사용하는 IP 주소에 대한 DNS 이름을 구성할 수 있습니다. 로컬 네트워크 관리자에게 문의하십시오.

HTTPS 또는 HTTP 연결을 사용하도록 결정합니다

로드 밸런서 끝점을 사용하여 클라이언트 연결을 만들 때는 해당 끝점에 지정된 프로토콜(HTTP 또는 HTTPS)을 사용하여 연결해야 합니다. 스토리지 노드 또는 게이트웨이 노드의 CLB 서비스에 대한 클라이언트 연결에 HTTP를 사용하려면 해당 사용을 설정해야 합니다.

기본적으로 클라이언트 응용 프로그램이 게이트웨이 노드의 스토리지 노드 또는 CLB 서비스에 연결할 때는 모든 연결에 암호화된 HTTPS를 사용해야 합니다. 선택적으로 Grid Manager에서 * HTTP Connection * 그리드 사용 옵션을 선택하여 보안성이 떨어지는 HTTP 연결을 활성화할 수 있습니다. 예를 들어, 클라이언트 애플리케이션은 비운영 환경에서 스토리지 노드에 대한 접속을 테스트할 때 HTTP를 사용할 수 있습니다.

> 요청은 암호화되지 않은 상태로 전송되므로 프로덕션 그리드에 대해 HTTP를 설정할 때는 주의해야 합니다.

> CLB 서비스는 더 이상 사용되지 않습니다.

HTTP 연결 사용 * 옵션을 선택한 경우 클라이언트는 HTTPS에 사용하는 것과 다른 HTTP 포트를 사용해야 합니다. StorageGRID 관리 지침을 참조하십시오.

관련 정보

StorageGRID 관리

Swift API 구성에서 연결을 테스트합니다

Swift CLI를 사용하여 StorageGRID 시스템에 대한 연결을 테스트하고 시스템에 개체를 읽고 쓸 수 있는지 확인할 수 있습니다.

필요한 것

- Swift 명령줄 클라이언트인 python-swifftclient를 다운로드하여 설치해야 합니다.

  "SwiftStack:python-swiftclient"

- StorageGRID 시스템에 Swift 테넌트 계정이 있어야 합니다.

이 작업에 대해

보안을 구성하지 않은 경우 각 명령에 '--insecure' 플래그를 추가해야 합니다.

단계

1. StorageGRID Swift 배포에 대한 정보 URL 쿼리:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/info
capabilities
```

   이는 Swift 배포가 제대로 작동하는지 테스트하는 데 충분합니다. 객체를 저장하여 계정 구성을 추가로 테스트하려면 추가 단계를 계속 진행합니다.

2. 컨테이너에 개체 넣기:

```
touch test_object
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
upload test_container test_object
--object-name test_object
```

3. 컨테이너를 내려 개체를 확인합니다.

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
list test_container
```

4. 개체 삭제:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
delete test_container test_object
```

5. 컨테이너를 삭제합니다.

```
swift
-U `<_Tenant_Account_ID:Account_User_Name_>`
-K `<_User_Password_>`
-A `\https://<_FQDN_ | _IP_>:<_Port_>/auth/v1.0'
delete test_container
```

관련 정보

Swift 테넌트 계정을 생성하고 구성합니다

REST API에 대한 보안을 구성합니다

Swift REST API가 작업을 지원했습니다

StorageGRID 시스템은 OpenStack Swift API에서 대부분의 작업을 지원합니다. Swift REST

API 클라이언트를 StorageGRID와 통합하기 전에 계정, 컨테이너 및 오브젝트 작업에 대한 구현 세부 정보를 검토하십시오.

StorageGRID에서 지원되는 작업입니다

다음과 같은 Swift API 작업이 지원됩니다.

- 계정 작업
- 컨테이너 작업
- 오브젝트 작업

모든 작업에 대한 공통 응답 헤더입니다

StorageGRID 시스템은 OpenStack Swift 오브젝트 스토리지 API v1에 정의된 대로 지원되는 작업에 대해 모든 공통 헤더를 구현합니다.

관련 정보

"OpenStack: 오브젝트 스토리지 API"

지원되는 **Swift API** 엔드포인트

StorageGRID는 정보 URL, 인증 URL 및 스토리지 URL과 같은 Swift API 엔드포인트를 지원합니다.

정보 **URL**

/info 경로가 있는 Swift 기본 URL에 GET 요청을 실행하여 StorageGRID Swift 구현의 기능 및 제한 사항을 확인할 수 있습니다.

' https://*FQDN* | _ Node IP: Swift Port_/info/'

요청 시:

- '*FQDN*'은 정규화된 도메인 이름입니다.
- '*Node IP*'는 StorageGRID 네트워크의 스토리지 노드 또는 게이트웨이 노드에 대한 IP 주소입니다.
- '*Swift Port*'는 스토리지 노드 또는 게이트웨이 노드의 Swift API 연결에 사용되는 포트 번호입니다.

예를 들어 다음 정보 URL은 IP 주소가 10.99.106.103이고 포트 18083을 사용하는 스토리지 노드에서 정보를 요청합니다.

'https://10.99.106.103:18083/info/`

응답에는 JSON 사전으로서 Swift 구현의 기능이 포함됩니다. 클라이언트 도구는 JSON 응답을 구문 분석하여 구현 기능을 결정하고 후속 스토리지 작업의 제약 조건으로 사용할 수 있습니다.

Swift의 StorageGRID 구현을 통해 정보 URL에 대한 인증되지 않은 액세스가 가능합니다.

인증 URL

클라이언트는 Swift 인증 URL을 사용하여 테넌트 계정 사용자로 인증할 수 있습니다.

' https://*FQDN* | _ Node IP: Swift Port_/auth/v1.0/'

다음과 같이 X-Auth-User와 X-Auth-Key 요청 헤더의 매개 변수로 테넌트 계정 ID, 사용자 이름 및 암호를 제공해야 합니다.

'X-Auth-User:_Tenant_Account_ID: Username _'

'X-Auth-Key:_Password_'

요청 헤더에서:

- `*Tenant_Account_ID*'는 Swift 테넌트가 생성될 때 StorageGRID에서 할당한 계정 ID입니다. 테넌트 관리자 로그인 페이지에서 사용되는 것과 동일한 테넌트 계정 ID입니다.
- '*Username*'은 테넌트 관리자에서 생성된 테넌트 사용자의 이름입니다. 이 사용자는 Swift 관리자 권한이 있는 그룹에 속해야 합니다. 테넌트의 루트 사용자는 Swift REST API를 사용하도록 구성할 수 없습니다.

  테넌트 계정에 대해 ID 페더레이션을 사용하도록 설정한 경우 LDAP 서버에서 연결된 사용자의 사용자 이름과 암호를 입력합니다. 또는 LDAP 사용자의 도메인 이름을 제공합니다. 예를 들면 다음과 같습니다.

  'X-Auth-User:_Tenant_Account_ID: 사용자 이름@Domain_Name_'

- '*Password*'는 테넌트 사용자의 암호입니다. 사용자 암호는 테넌트 관리자에서 생성 및 관리됩니다.

인증 요청에 대한 응답은 다음과 같이 스토리지 URL 및 인증 토큰을 반환합니다.

'X-Storage-URL: https://FQDN |Node_IP:Swift_Port/v1/Tenant_Account_ID'

'X-Auth-Token:_token_'

'X-Storage-Token:_token_'

기본적으로 토큰은 생성 후 24시간 동안 유효합니다.

특정 테넌트 계정에 대해 토큰이 생성됩니다. 한 계정에 대해 유효한 토큰이 사용자에게 다른 계정에 액세스할 수 있는 권한을 부여하지 않습니다.

스토리지 URL입니다

클라이언트 애플리케이션은 Swift REST API 호출을 실행하여 게이트웨이 노드 또는 스토리지 노드에 대해 지원되는 계정, 컨테이너 및 오브젝트 작업을 수행할 수 있습니다. 저장소 요청은 인증 응답에서 반환된 저장소 URL로 처리됩니다. 또한 요청에는 인증 요청에서 반환된 X-Auth-Token 헤더 및 값이 포함되어야 합니다.

'https://FQDN |*IP:Swift_Port*/v1/*Tenant_Account_ID*'

'[*/container*][*/object*]'

'X-Auth-Token:_token_'

사용 통계를 포함하는 일부 스토리지 응답 헤더는 최근에 수정된 개체의 정확한 숫자를 반영하지 않을 수 있습니다. 이

머리글에 정확한 숫자가 표시되려면 몇 분 정도 걸릴 수 있습니다.

계정 및 컨테이너 작업에 대한 다음 응답 머리글은 사용 통계를 포함하는 응답의 예입니다.

- X-Account-Bytes-Used
- X-Account-Object-Count'입니다
- X-Container-Bytes-Used(X-Container-Bytes-Used)
- X-Container-Object-Count

관련 정보

테넌트 계정 및 연결을 구성합니다

계정 작업

컨테이너 작업

오브젝트 작업

계정 작업

다음 Swift API 작업은 어카운트에 대해 수행됩니다.

계정을 가져옵니다

이 작업은 계정 및 계정 사용 통계와 연결된 컨테이너 목록을 검색합니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다

다음 요청 헤더가 필요합니다.

- X-Auth-Token

다음과 같은 지원되는 요청 쿼리 매개 변수는 선택 사항입니다.

- "암월터"
- 'End_marker'입니다
- 형식
- 제한
- '마커'
- 접두사

계정이 발견되어 컨테이너가 없거나 컨테이너 목록이 비어 있는 경우 "HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 성공적으로 실행되면 "HTTP/1.1 200 OK" 응답이 반환됩니다. 계정이 발견되어 컨테이너 목록이 비어 있지 않은 경우 "HTTP/1.1 200 OK" 응답이 반환됩니다.

- '수용 범위'

- 콘텐츠 길이
- 콘텐츠 유형
- 다
- X-Account-Bytes-Used
- X-계정-컨테이너-카운트
- X-Account-Object-Count'입니다
- X-타임스탬프
- X-Trans-ID

머리 계정

이 작업은 Swift 계정에서 계정 정보 및 통계를 검색합니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 ""HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 반환됩니다.

- '수용 범위'
- 콘텐츠 길이
- 다
- X-Account-Bytes-Used
- X-계정-컨테이너-카운트
- X-Account-Object-Count'입니다
- X-타임스탬프
- X-Trans-ID

관련 정보

운영 모니터링 및 감사

컨테이너 작업

StorageGRID는 Swift 계정당 최대 1,000개의 컨테이너를 지원합니다. 컨테이너에서 다음과 같은 Swift API 작업이 수행됩니다.

컨테이너를 삭제합니다

이 작업을 수행하면 StorageGRID 시스템의 Swift 계정에서 빈 컨테이너가 제거됩니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 "HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 반환됩니다.

- 콘텐츠 길이
- 콘텐츠 유형
- 다
- X-Trans-ID

컨테이너를 가져옵니다

이 작업은 StorageGRID 시스템의 컨테이너 통계 및 메타데이터와 함께 컨테이너와 연결된 개체 목록을 검색합니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너

다음 요청 헤더가 필요합니다.

- X-Auth-Token

다음과 같은 지원되는 요청 쿼리 매개 변수는 선택 사항입니다.

- "암월터"
- 'End_marker'입니다
- 형식
- 제한
- '마커'
- "경로"
- 접두사

성공적으로 실행하면 "HTTP/1.1 200 Success" 또는 "HTTP/1.1 204 No Content" 응답으로 다음 헤더가 반환됩니다.

- '수용 범위'
- 콘텐츠 길이
- 콘텐츠 유형
- 다
- X-Container-Bytes-Used(X-Container-Bytes-Used)

- X-Container-Object-Count
- X-타임스탬프
- X-Trans-ID

헤드 컨테이너

이 작업은 StorageGRID 시스템에서 컨테이너 통계 및 메타데이터를 검색합니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 "HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 반환됩니다.

- '수용 범위'
- 콘텐츠 길이
- 다
- X-Container-Bytes-Used(X-Container-Bytes-Used)
- X-Container-Object-Count
- X-타임스탬프
- X-Trans-ID

용기를 놓습니다

이 작업은 StorageGRID 시스템의 계정에 대한 컨테이너를 만듭니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 "HTTP/1.1 201 Created" 또는 "HTTP/1.1 202 Accepted"(컨테이너가 이미 이 계정에 있는 경우) 응답으로 다음 헤더가 반환됩니다.

- 콘텐츠 길이
- 다
- X-타임스탬프

- X-Trans-ID

컨테이너 이름은 StorageGRID 네임스페이스에서 고유해야 합니다. 컨테이너가 다른 계정 아래에 있는 경우 "HTTP/1.1 409 충돌"이라는 헤더가 반환됩니다.

관련 정보

운영 모니터링 및 감사

오브젝트 작업

객체에 대해 다음과 같은 Swift API 작업이 수행됩니다.

개체를 삭제합니다

이 작업은 StorageGRID 시스템에서 개체의 콘텐츠 및 메타데이터를 삭제합니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너
- '개체'

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 HTTP/1.1 204 No Content 응답이 있는 다음 응답 헤더가 반환됩니다.

- 콘텐츠 길이
- 콘텐츠 유형
- 다
- X-Trans-ID

오브젝트 삭제 요청을 처리할 때 StorageGRID는 저장된 모든 위치에서 오브젝트의 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.

개체 삭제 방법에 대한 자세한 내용은 정보 수명 주기 관리를 사용하여 개체 관리 지침을 참조하십시오.

객체를 가져옵니다

이 작업은 개체 콘텐츠를 검색하고 StorageGRID 시스템에서 개체 메타데이터를 가져옵니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너
- '개체'

다음 요청 헤더가 필요합니다.

- X-Auth-Token

다음 요청 헤더는 선택 사항입니다.

- 인코딩 수락
- IF-MATCH
- If-Modified-Since
- "If-None-Match"
- "수정되지 않은 경우 - 이후"
- "범위"

성공적으로 실행하면 HTTP/1.1 200 OK 응답이 있는 다음 헤더가 반환됩니다.

- '수용 범위'
- Content-Disposition은 Content-Disposition 메타데이터가 설정된 경우에만 반환됩니다
- Content-Encoding은 Content-Encoding 메타데이터가 설정된 경우에만 반환됩니다
- 콘텐츠 길이
- 콘텐츠 유형
- 다
- 'ETag'
- 마지막 수정일
- X-타임스탬프
- X-Trans-ID

머리 물체

이 작업은 StorageGRID 시스템에서 수집된 개체의 메타데이터 및 속성을 검색합니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너
- '개체'

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 "HTTP/1.1 200 OK" 응답과 함께 다음 헤더가 반환됩니다.

- '수용 범위'
- Content-Disposition은 Content-Disposition 메타데이터가 설정된 경우에만 반환됩니다

- Content-Encoding은 Content-Encoding 메타데이터가 설정된 경우에만 반환됩니다
- 콘텐츠 길이
- 콘텐츠 유형
- 다
- 'ETag'
- 마지막 수정일
- X-타임스탬프
- X-Trans-ID

개체를 넣습니다

이 작업을 실행하면 새 개체가 데이터와 메타데이터로 만들어지거나 기존 개체를 StorageGRID 시스템의 데이터 및 메타데이터로 바꿉니다.

StorageGRID는 최대 5TiB(5,497,558,138,880바이트)의 오브젝트를 지원합니다.

> 동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "최신" 평가 시기는 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 하며, Swift 클라이언트가 작업을 시작하는 시점이 아닙니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다
- 컨테이너
- '개체'

다음 요청 헤더가 필요합니다.

- X-Auth-Token

다음 요청 헤더는 선택 사항입니다.

- 'Content-Disposition'
- 콘텐츠 인코딩

  개체에 적용되는 ILM 규칙이 크기에 따라 개체를 필터링하고 수집 시 동기식 배치(Ingest 동작에 대한 균형 또는 엄격 옵션)를 사용하는 경우 청크된 "콘텐츠 인코딩"을 사용하지 마십시오.

- 전송 인코딩

  개체에 적용되는 ILM 규칙이 크기에 따라 개체를 필터링하고 수집 시 동기식 배치(Ingest 동작에 대한 균형 또는 엄격 옵션)를 사용하는 경우 압축 또는 청크된 "전송 인코딩"을 사용하지 마십시오.

- 콘텐츠 길이

  ILM 규칙이 크기를 기준으로 오브젝트를 필터링하고 수집 시 동기 배치를 사용하는 경우 'Content-Length'를 지정해야 합니다.

> Content-Encoding, Transfer-Encoding, Content-Length에 대한 지침을 따르지 않을 경우 StorageGRID는 개체 크기를 결정하고 ILM 규칙을 적용하기 전에 개체를 저장해야 합니다. 다시 말해, StorageGRID은 수집 중인 오브젝트의 중간 복사본을 기본적으로 생성해야 합니다. 즉, StorageGRID는 Ingest 동작에 대해 이중 커밋 옵션을 사용해야 합니다.

동기 배치 및 ILM 규칙에 대한 자세한 내용은 정보 수명 주기 관리를 통해 개체 관리 지침을 참조하십시오.

- 콘텐츠 유형
- 'ETag'
- 'X-Object-Meta-<name\>'(객체 관련 메타데이터)

  ILM 규칙의 참조 시간으로 * 사용자 정의 작성 시간 * 옵션을 사용하려면 값을 사용자 정의 헤더("X-Object-Meta-Creation-Time")에 저장해야 합니다. 예를 들면 다음과 같습니다.

  ```
  X-Object-Meta-Creation-Time: 1443399726
  ```

  이 필드는 1970년 1월 1일 이후 초 단위로 평가됩니다.

- X-Storage-Class: reduced_redundancy가 있습니다

  수집된 개체와 일치하는 ILM 규칙이 이중 커밋 또는 균형 설정의 수집 동작을 지정하는 경우 이 헤더는 StorageGRID에서 만드는 개체 복사본 수에 영향을 줍니다.

  - * 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.

    reducted_redundancy' 헤더는 개체와 일치하는 ILM 규칙이 복제된 단일 복사본을 만들 때 가장 적합합니다. 이 경우 'REDED_READITORY'를 사용하면 모든 수집 작업에 대해 불필요한 오브젝트 복사본을 생성하고 삭제할 필요가 없습니다.

    다른 상황에서는 수집 중에 오브젝트 데이터가 손실될 위험이 있기 때문에 reducted_redundancy" 헤더를 사용하지 않는 것이 좋습니다. 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.

  reducted_redundancy를 지정하면 개체를 처음 인제스트할 때 생성되는 복사본 수에만 영향을 줍니다. 활성 ILM 정책에 따라 개체를 평가할 때 개체의 복사본 수에 영향을 주지 않으며 StorageGRID 시스템의 낮은 수준의 중복성에 데이터가 저장되지 않습니다.

성공적으로 실행하면 "HTTP/1.1 201 created" 응답으로 다음 헤더가 반환됩니다.

- 콘텐츠 길이
- 콘텐츠 유형
- 다
- 'ETag'
- 마지막 수정일
- X-Trans-ID

관련 정보

ILM을 사용하여 개체를 관리합니다

운영 모니터링 및 감사

옵션 요청

옵션 요청은 개별 Swift 서비스의 사용 가능 여부를 확인합니다. 옵션 요청은 URL에 지정된 스토리지 노드 또는 게이트웨이 노드에 의해 처리됩니다.

옵션 방법입니다

예를 들어, 클라이언트 애플리케이션은 스토리지 노드의 Swift 인증 자격 증명을 제공하지 않고 스토리지 노드의 Swift 포트에 대한 옵션 요청을 발급하여 스토리지 노드를 사용할 수 있는지 여부를 확인할 수 있습니다. 이 요청을 사용하여 스토리지 노드가 다운된 시점을 모니터링하거나 외부 로드 밸런서가 식별하도록 할 수 있습니다.

info URL 또는 저장소 URL과 함께 사용할 경우 options 메서드는 지정된 URL에 대해 지원되는 동사 목록(예: head, get, options 및 put)을 반환합니다. 옵션 방법은 인증 URL과 함께 사용할 수 없습니다.

다음 요청 매개 변수가 필요합니다.

- '계정'입니다

다음 요청 매개 변수는 선택 사항입니다.

- 컨테이너
- '개체'

성공적으로 실행하면 HTTP/1.1 204 컨텐츠 없음 응답이 있는 다음 헤더가 반환됩니다. 스토리지 URL에 대한 옵션 요청에는 타겟이 없을 필요가 없습니다.

- "Allow"(지정된 URL에 대해 지원되는 동사 목록, 예: head, get, options, 및 PUT)
- 콘텐츠 길이
- 콘텐츠 유형
- 다
- X-Trans-ID

관련 정보

지원되는 Swift API 엔드포인트

Swift API 작업에 대한 오류 응답

가능한 오류 응답을 이해하면 작업 문제를 해결하는 데 도움이 됩니다.

작업 중에 오류가 발생하면 다음 HTTP 상태 코드가 반환될 수 있습니다.

SWIFT 오류 이름	**HTTP** 상태입니다
AccountNameTooLong, ContainerNameTooLong, HeaderTooBig, InvalidContainerName, InvalidRequest, InvalidURI, MetadataNameTooLong, MetadataValueTooBig, MissingSecurityHeader, ObjectNameTooLong, TooManyContainers, TooManyMetadataItems, TotalMetadataTooLarge입니다	400 잘못된 요청
액세스가 거부되었습니다	403 사용 금지
ContainerNotEmpty , ContainerAlreadyExists	409 충돌
내부 오류입니다	500 내부 서버 오류입니다
InvalidRange 를 선택합니다	416 요청된 범위가 충분하지 않습니다
MethodNotAllowed 를 참조하십시오	405 메서드를 사용할 수 없습니다
MissingContentLength를 참조하십시오	411 길이 필요
지원되지 않습니다	404를 찾을 수 없습니다
구현되지 않았습니다	501 구현되지 않음
사전 조건에 실패했습니다	412 전제 조건 실패
리소스 NotFound 를 참조하십시오	404를 찾을 수 없습니다
권한이 없습니다	401 승인되지 않음
UnprocessableEntity입니다	422 처리할 수 없는 엔터티

StorageGRID Swift REST API 작업

StorageGRID 시스템별 Swift REST API에 작업이 추가됩니다.

컨테이너 일관성 요청 가져오기

정합성 보장 레벨은 오브젝트의 가용성과 서로 다른 스토리지 노드 및 사이트에서 이러한 오브젝트의 일관성 간의 균형을 제공합니다. 컨테이너 일관성 가져오기 요청을 사용하면 특정 컨테이너에 적용되는 일관성 수준을 확인할 수 있습니다.

요청하십시오

HTTP 헤더를 요청합니다	설명
X-Auth-Token	요청에 사용할 계정의 Swift 인증 토큰을 지정합니다.
'x-ntap-sg-consistency'	요청 유형을 지정합니다. 여기서 true는 컨테이너 일관성을 얻고 false는 컨테이너를 가져옵니다.
호스트	요청이 전달되는 호스트 이름입니다.

요청 예

```
GET /v1/28544923908243208806/Swift container
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: true
Host: test.com
```

응답

응답 **HTTP** 헤더	설명
다	응답의 날짜 및 시간입니다.
'연결'을 선택합니다	서버에 대한 연결이 열려 있는지 또는 닫혀 있는지 여부
X-Trans-ID	요청에 대한 고유한 트랜잭션 식별자입니다.
콘텐츠 길이	응답 바디의 길이.

응답 **HTTP** 헤더	설명
'x-ntap-sg-consistency'	컨테이너에 적용되는 정합성 보장 제어 레벨입니다. 지원되는 값은 다음과 같습니다. • * ALL *: 모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다. • * strong-global *: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • * strong-site *: 사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • * read-after-new-write *: 새 객체에 대한 읽기 후 쓰기 정합성을 보장하고 객체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 • 참고 *: 응용 프로그램이 존재하지 않는 개체에 대한 헤드 요청을 사용하는 경우 하나 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다. 이러한 오류를 방지하려면 "사용 가능" 수준을 사용하십시오. • * Available * (헤드 작업의 최종 일관성): "read-after-new-write" 정합성 수준과 동일하게 동작하지만 헤드 작업에 대한 최종 정합성 보장만 제공합니다. 스토리지 노드를 사용할 수 없는 경우 "새 쓰기 후"보다 헤드 작업에 더 높은 가용성을 제공합니다.

응답 예

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
x-ntap-sg-consistency: strong-site
```

관련 정보

테넌트 계정을 사용합니다

컨테이너 일관성 요청

저장 컨테이너 일관성 요청을 사용하면 컨테이너에서 수행된 작업에 적용할 일관성 수준을 지정할 수 있습니다. 기본적으로 새 컨테이너는 "새 쓰기 후 다시 쓰기" 일관성 수준을 사용하여 생성됩니다.

요청하십시오

HTTP 헤더를 요청합니다	설명
X-Auth-Token	요청에 사용할 계정의 Swift 인증 토큰입니다.
'x-ntap-sg-consistency'	컨테이너의 작업에 적용할 일관성 제어 수준입니다. 지원되는 값은 다음과 같습니다. • * ALL *: 모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다. • * strong-global *: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • * strong-site *: 사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • * read-after-new-write *: 새 객체에 대한 읽기 후 쓰기 정합성을 보장하고 객체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 • 참고 *: 응용 프로그램이 존재하지 않는 개체에 대한 헤드 요청을 사용하는 경우 하나 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다. 이러한 오류를 방지하려면 "사용 가능" 수준을 사용하십시오. • * Available * (헤드 작업의 최종 일관성): "read-after-new-write" 정합성 수준과 동일하게 동작하지만 헤드 작업에 대한 최종 정합성 보장만 제공합니다. 스토리지 노드를 사용할 수 없는 경우 "새 쓰기 후"보다 헤드 작업에 더 높은 가용성을 제공합니다.
호스트	요청이 전달되는 호스트 이름입니다.

일관성 제어 및 **ILM** 규칙이 상호 작용하여 데이터 보호에 영향을 미치는 방식

일관성 제어와 ILM 규칙 모두 오브젝트의 보호 방법에 영향을 미칩니다. 이러한 설정은 상호 작용할 수 있습니다.

예를 들어, 개체가 저장될 때 사용되는 일관성 컨트롤은 오브젝트 메타데이터의 초기 배치에 영향을 미치는 반면 ILM 규칙에 대해 선택된 수집 동작은 오브젝트 복사본의 초기 배치에 영향을 줍니다. StorageGRID에서는 클라이언트 요청을 이행하기 위해 오브젝트의 메타데이터와 해당 데이터에 모두 액세스해야 하므로 일관성 수준과 수집 동작에 적합한 보호 수준을 선택하면 초기 데이터 보호 수준을 높이고 시스템 응답을 더욱 정확하게 예측할 수 있습니다.

ILM 규칙에 대해 다음과 같은 수집 동작을 사용할 수 있습니다.

- * Strict * : ILM 규칙에 지정된 모든 사본은 클라이언트에 반환되기 전에 만들어야 합니다.
- * 균형 *: StorageGRID는 수집 시 ILM 규칙에 지정된 모든 복제본을 생성하려고 합니다. 그렇지 않을 경우 중간 복사본이 만들어지고 클라이언트에 성공적으로 반환됩니다. ILM 규칙에 지정된 복사본은 가능한 경우 만들어집니다.
- * 이중 커밋*: StorageGRID는 즉시 개체의 임시 복사본을 만들고 클라이언트에 성공을 반환합니다. ILM 규칙에 지정된 복사본은 가능한 경우 만들어집니다.

ILM 규칙의 수집 동작을 선택하기 전에 정보 수명 주기 관리를 통해 개체를 관리하기 위한 지침에서 이러한 설정에 대한 전체 설명을 읽어보십시오.

일관성 제어 및 **ILM** 규칙이 상호 작용하는 방법의 예

다음 ILM 규칙 및 다음 일관성 수준 설정이 있는 두 사이트 그리드가 있다고 가정합니다.

- * ILM 규칙 *: 로컬 사이트와 원격 사이트에 각각 하나씩, 두 개의 오브젝트 복사본을 만듭니다. Strict 수집 동작이 선택됩니다.
- * Consistency level *: "trong-global"(개체 메타데이터가 모든 사이트에 즉시 배포됩니다.)

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 규칙 및 "'strong-site' 정합성 보장 수준을 사용한 경우 객체 데이터가 원격 사이트에 복제되었지만 객체 메타데이터가 그 위치에 배포되기 전에 클라이언트에 성공 메시지가 표시될 수 있습니다. 이 경우 오브젝트 메타데이터의 보호 수준이 오브젝트 데이터의 보호 수준과 일치하지 않습니다. 수집 후 곧바로 로컬 사이트가 손실되면 오브젝트 메타데이터가 손실됩니다. 객체를 검색할 수 없습니다.

일관성 수준과 ILM 규칙 간의 상호 관계는 복잡할 수 있습니다. 도움이 필요한 경우 NetApp에 문의하십시오.

요청 예

```
PUT /v1/28544923908243208806/_Swift container_
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: strong-site
Host: test.com
```

응답

응답 **HTTP** 헤더	설명
다	응답의 날짜 및 시간입니다.
'연결'을 선택합니다	서버에 대한 연결이 열려 있는지 또는 닫혀 있는지 여부
X-Trans-ID	요청에 대한 고유한 트랜잭션 식별자입니다.
콘텐츠 길이	응답 바디의 길이.

응답 예

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
```

관련 정보

테넌트 계정을 사용합니다

REST API에 대한 보안을 구성합니다

REST API에 대해 구현된 보안 조치를 검토하고 시스템 보안 방법을 이해해야 합니다.

StorageGRID에서 REST API에 대한 보안을 제공하는 방법

StorageGRID 시스템이 REST API에 대한 보안, 인증 및 권한 부여를 구현하는 방법을 이해해야 합니다.

StorageGRID는 다음과 같은 보안 조치를 사용합니다.

- 로드 밸런서 끝점에 HTTPS가 구성되어 있는 경우 로드 밸런서 서비스와의 클라이언트 통신은 HTTPS를 사용합니다.

  로드 밸런서 끝점을 구성할 때 HTTP를 선택적으로 활성화할 수 있습니다. 예를 들어, 테스트 또는 기타 비운영 목적으로 HTTP를 사용할 수 있습니다. 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

- 기본적으로 StorageGRID는 게이트웨이 노드에서 스토리지 노드 및 CLB 서비스와의 클라이언트 통신에 HTTPS를 사용합니다.

  이러한 연결에 대해 HTTP를 선택적으로 활성화할 수 있습니다. 예를 들어, 테스트 또는 기타 비운영 목적으로 HTTP를 사용할 수 있습니다. 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

  > CLB 서비스는 더 이상 사용되지 않습니다.

- StorageGRID와 클라이언트 간의 통신은 TLS를 사용하여 암호화됩니다.
- 로드 밸런서 끝점이 HTTP 또는 HTTPS 연결을 허용하도록 구성되었는지 여부에 관계없이 그리드 내의 로드 밸런서 서비스와 스토리지 노드 간의 통신이 암호화됩니다.
- 클라이언트는 REST API 작업을 수행하기 위해 StorageGRID에 HTTP 인증 헤더를 제공해야 합니다.

보안 인증서 및 클라이언트 응용 프로그램

클라이언트는 게이트웨이 노드 또는 관리 노드의 부하 분산 서비스, 스토리지 노드 또는 게이트웨이 노드의 더 이상 사용되지 않는 CLB 서비스에 직접 연결할 수 있습니다.

모든 경우에 클라이언트 응용 프로그램은 그리드 관리자가 업로드한 사용자 지정 서버 인증서 또는 StorageGRID 시스템에서 생성한 인증서를 사용하여 TLS 연결을 만들 수 있습니다.

- 클라이언트 응용 프로그램이 로드 밸런서 서비스에 연결되면 연결을 만드는 데 사용되는 특정 로드 밸런서 끝점에 대해 구성된 인증서를 사용합니다. 각 끝점마다 고유한 인증서가 있습니다. 이 인증서는 그리드 관리자가 업로드한

사용자 지정 서버 인증서이거나, 끝점 구성 시 그리드 관리자가 StorageGRID에서 생성한 인증서입니다.

- 클라이언트 응용 프로그램이 게이트웨이 노드의 스토리지 노드 또는 CLB 서비스에 직접 연결할 때 StorageGRID 시스템이 설치될 때 스토리지 노드에 대해 생성된 시스템 생성 서버 인증서(시스템 인증 기관이 서명)를 사용합니다. 또는 그리드 관리자가 그리드에 제공하는 단일 사용자 정의 서버 인증서입니다.

클라이언트가 TLS 연결을 설정하는 데 사용하는 인증서를 신뢰하도록 구성해야 합니다.

로드 밸런서 끝점 구성에 대한 정보와 TLS 연결에 대한 단일 사용자 지정 서버 인증서를 스토리지 노드 또는 게이트웨이 노드의 CLB 서비스에 직접 추가하는 방법에 대한 지침은 StorageGRID 관리 지침을 참조하십시오.

요약

다음 표에서는 S3 및 Swift REST API에서 보안 문제가 구현되는 방식을 보여 줍니다.

보안 문제	**REST API** 구현
연결 보안	TLS
서버 인증	시스템 CA에서 서명한 X.509 서버 인증서 또는 관리자가 제공한 사용자 지정 서버 인증서입니다
클라이언트 인증	• S3:S3 계정(액세스 키 ID 및 비밀 액세스 키) • Swift:Swift 계정(사용자 이름 및 암호)
클라이언트 인증	• S3: 버킷 소유권 및 모든 적용 가능한 액세스 제어 정책 • Swift: 관리자 역할 액세스

관련 정보

StorageGRID 관리

TLS 라이브러리에 대해 지원되는 해시 및 암호화 알고리즘

StorageGRID 시스템은 TLS(전송 계층 보안) 세션을 설정할 때 클라이언트 응용 프로그램에서 사용할 수 있는 제한된 암호화 그룹 세트를 지원합니다.

지원되는 **TLS** 버전입니다

StorageGRID는 TLS 1.2 및 TLS 1.3을 지원합니다.

> SSLv3 및 TLS 1.1(또는 이전 버전)은 더 이상 지원되지 않습니다.

지원되는 암호 그룹

TLS 버전입니다	암호화 그룹의 **IANA** 이름입니다
1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TLS 버전입니다	암호화 그룹의 **IANA** 이름입니다
TLS_ECDHE_RSA_WITH_CHACH20_POLY1305_SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
1.3	TLS_AES_256_GCM_SHA384
TLS_CHACH20_POLY1305_SHA256	TLS_AES_128_GCM_SHA256

더 이상 사용되지 않는 암호화 그룹

다음 암호화 그룹은 더 이상 사용되지 않습니다. 이러한 암호화에 대한 지원은 이후 릴리스에서 제거됩니다.

IANA 이름입니다
TLS_RSA_with_AES_128_GCM_SHA256
TLS_RSA_WITED_AES_256_GCM_SHA384

관련 정보

테넌트 계정 및 연결을 구성합니다

운영 모니터링 및 감사

전체 그리드 또는 특정 노드에 대한 트랜잭션 추세를 확인하여 클라이언트 작업의 워크로드 및 효율성을 모니터링할 수 있습니다. 감사 메시지를 사용하여 클라이언트 작업 및 트랜잭션을 모니터링할 수 있습니다.

오브젝트 수집 및 검색 속도 모니터링

오브젝트 수집 및 검색 속도와 오브젝트 수, 쿼리, 검증을 위한 메트릭을 모니터링할 수 있습니다. 클라이언트 응용 프로그램이 StorageGRID 시스템에서 개체를 읽고, 쓰고, 수정하는 데 성공한 시도 및 실패한 시도 횟수를 볼 수 있습니다.

단계

1. 를 사용하여 Grid Manager에 로그인합니다 지원되는 웹 브라우저.
2. Dashboard에서 Protocol Operations 섹션을 찾습니다.

   이 섹션에서는 StorageGRID 시스템에서 수행하는 클라이언트 작업의 수를 요약합니다. 프로토콜 속도는 최근 2분 동안의 평균값입니다.

3. 노드 * 를 선택합니다.
4. 노드 홈 페이지(배포 수준)에서 * 로드 밸런서 * 탭을 클릭합니다.

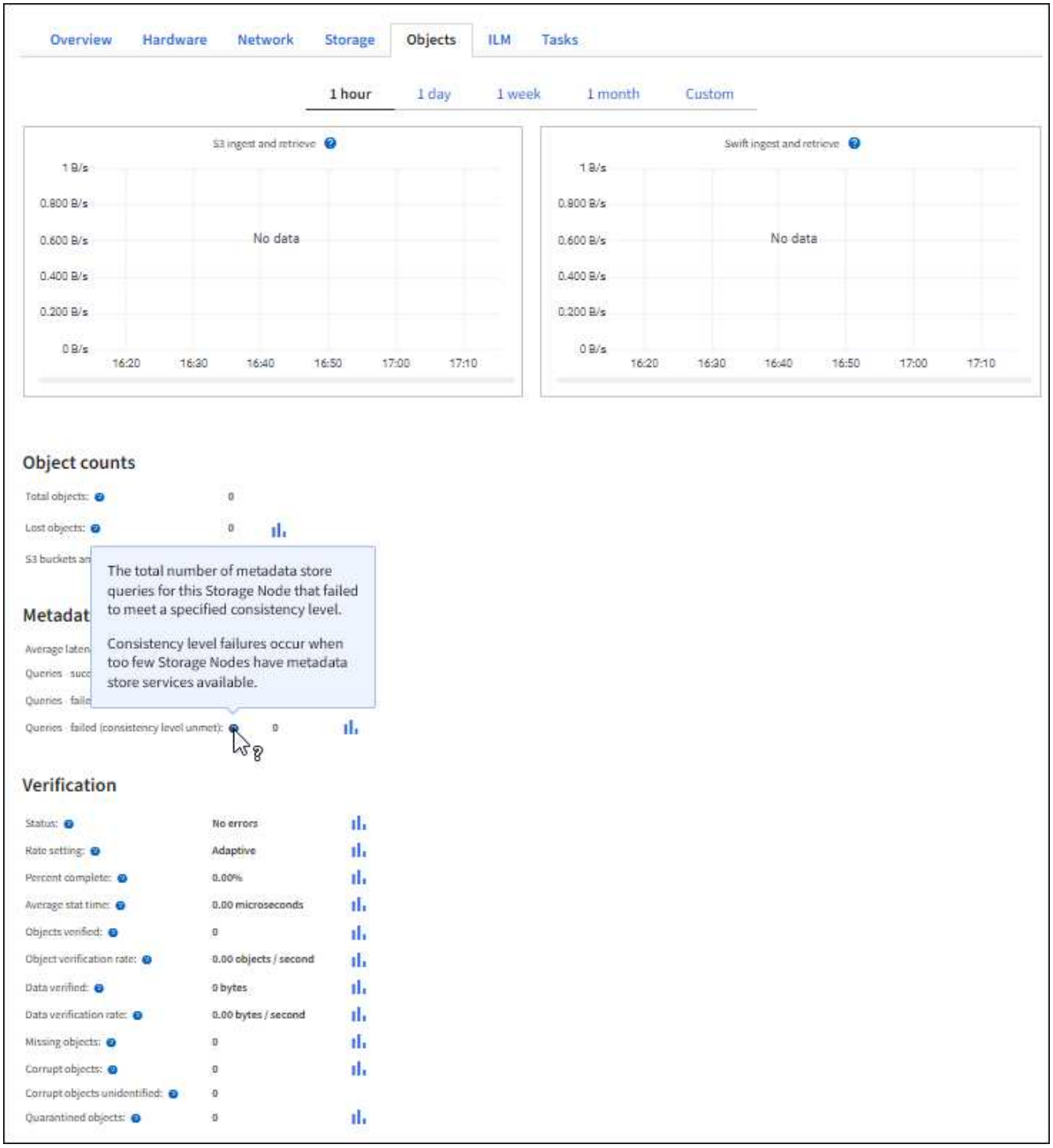
   차트에는 그리드 내의 로드 밸런서 끝점에 대한 모든 클라이언트 트래픽에 대한 추세가 표시됩니다. 시간 간격(시간, 일, 주, 월 또는 년)을 선택할 수 있습니다. 또는 사용자 지정 간격을 적용할 수 있습니다.

5. 노드 홈 페이지(배포 수준)에서 * 개체 * 탭을 클릭합니다.

   이 차트에는 전체 StorageGRID 시스템의 수집 및 검색 속도가 초당 바이트 및 총 바이트 단위로 표시됩니다. 시간 간격(시간, 일, 주, 월 또는 년)을 선택할 수 있습니다. 또는 사용자 지정 간격을 적용할 수 있습니다.

6. 특정 스토리지 노드에 대한 정보를 보려면 왼쪽의 목록에서 노드를 선택하고 * Objects * 탭을 클릭합니다.

   이 차트에는 이 스토리지 노드의 객체 수집 및 검색 속도가 나와 있습니다. 이 탭에는 개체 수, 쿼리 및 검증에 대한 메트릭도 포함되어 있습니다. 레이블을 클릭하여 이러한 메트릭의 정의를 볼 수 있습니다.



7. 더 자세한 내용을 원하는 경우:

a. 지원 * > * 도구 * > * 그리드 토폴로지 * 를 선택합니다.

b. site_ * > * Overview * > * Main * 을 선택합니다.

   API 작업 섹션에는 전체 그리드에 대한 요약 정보가 표시됩니다.

c. 스토리지 노드 * > * LDR * > **CLIENT APPLICATION* * > * Overview * > * Main * 을 선택합니다

   작업 섹션에는 선택한 스토리지 노드에 대한 요약 정보가 표시됩니다.

감사 로그 액세스 및 검토

감사 메시지는 StorageGRID 서비스에서 생성되고 텍스트 로그 파일에 저장됩니다. 감사 로그의 API 관련 감사 메시지는 시스템의 상태를 평가하는 데 도움이 되는 중요한 보안, 운영 및 성능 모니터링 데이터를 제공합니다.

필요한 것

- 특정 액세스 권한이 있어야 합니다.
- "passwords.txt" 파일이 있어야 합니다.
- 관리 노드의 IP 주소를 알아야 합니다.

이 작업에 대해

활성 감사 로그 파일은 AUDIT.LOG라는 이름으로 관리 노드에 저장됩니다.

하루에 한 번 활성 audit.log 파일이 저장되고 새 audit.log 파일이 시작됩니다. 저장된 파일의 이름은 저장 시기를 yyyy-mm-dd.txt 형식으로 나타냅니다.

하루 후에는 원래 날짜를 유지하는 형식(yyyy-mm-dd.txt.gz)으로 저장된 파일이 압축되고 이름이 변경됩니다.

이 예에서는 활성 audit.log 파일, 이전 날짜의 파일(2018-04-15.txt) 및 이전 날짜의 압축 파일('2018-04-14.txt.gz')을 보여 줍니다.

```
audit.log
2018-04-15.txt
2018-04-14.txt.gz
```

단계

1. 관리자 노드에 로그인:

   a. 'ssh_admin@primary_Admin_Node_IP_' 명령을 입력합니다

   b. "passwords.txt" 파일에 나열된 암호를 입력합니다.

2. 감사 로그 파일이 들어 있는 디렉터리('cd/var/local/audit/export')로 이동합니다

3. 필요에 따라 현재 또는 저장된 감사 로그 파일을 봅니다.

관련 정보

감사 로그를 검토합니다

성공한 모든 스토리지 삭제, 가져오기, 헤드, POST 및 PUT 작업은 StorageGRID 감사 로그에서 추적됩니다. 실패는 기록되지 않으며 정보, 인증 또는 옵션 요청도 기록되지 않습니다.

다음 Swift 작업에 대해 추적되는 정보에 대한 자세한 내용은 _감사 메시지 이해_를 참조하십시오.

계정 작업

- 계정을 가져옵니다
- 머리 계정

컨테이너 작업

- 컨테이너를 삭제합니다
- 컨테이너를 가져옵니다
- 헤드 컨테이너
- 용기를 놓습니다

오브젝트 작업

- 개체를 삭제합니다
- 객체를 가져옵니다
- 머리 물체
- 개체를 넣습니다

관련 정보

감사 로그를 검토합니다

계정 작업

컨테이너 작업

오브젝트 작업