StorageGRID 사용

StorageGRID software

NetApp
July 23, 2026

목차

StorageGRID 테넌트 및 클라이언트 사용

테넌트 계정 사용

StorageGRID 테넌트 계정 사용

테넌트 계정을 사용하면 Simple Storage Service(S3) REST API를 이용하여 StorageGRID 시스템에 객체를 저장하고 검색할 수 있습니다.

테넌트 계정이란 무엇인가요?

각 테넌트 계정은 자체적인 연합 또는 로컬 그룹, 사용자, S3 버킷 및 객체를 보유합니다.

테넌트 계정을 사용하면 저장된 객체를 서로 다른 엔티티별로 구분할 수 있습니다. 예를 들어, 다음과 같은 두 가지 사용 사례에 여러 테넌트 계정을 사용할 수 있습니다.

- 엔터프라이즈 사용 사례: StorageGRID 시스템을 기업 환경에서 사용하는 경우, 그리드의 객체 스토리지는 조직 내 여러 부서별로 분리될 수 있습니다. 예를 들어 마케팅 부서, 고객 지원 부서, 인사 부서 등을 위한 테넌트 계정이 있을 수 있습니다.

> S3 클라이언트 프로토콜을 사용하는 경우 S3 버킷 및 버킷 정책을 사용하여 기업 내 부서 간에 객체를 분리할 수 있습니다. 별도의 테넌트 계정을 생성할 필요가 없습니다. 자세한 내용은 "S3 버킷 및 버킷 정책" 구현 지침을 참조하십시오.

- 서비스 제공업체 사용 사례: StorageGRID 시스템을 서비스 제공업체가 사용하는 경우, 그리드의 객체 스토리지는 스토리지를 임대하는 여러 주체별로 분리될 수 있습니다. 예를 들어, Company A, Company B, Company C 등의 테넌트 계정이 있을 수 있습니다.

테넌트 계정을 만드는 방법

테넌트 계정은 "StorageGRID 그리드 관리자가 Grid Manager를 사용합니다."이(가) 생성합니다. 테넌트 계정을 생성할 때 그리드 관리자는 다음 사항을 지정합니다.

- 테넌트 이름, 클라이언트 유형(S3) 및 선택적 스토리지 할당량을 포함한 기본 정보.
- 테넌트 계정에 대한 권한에는 테넌트 계정이 S3 플랫폼 서비스를 사용할 수 있는지, 자체 ID 소스를 구성할 수 있는지, S3 Select를 사용할 수 있는지, 또는 그리드 페더레이션 연결을 사용할 수 있는지 여부가 포함됩니다.
- 테넌트의 초기 루트 액세스는 StorageGRID 시스템이 로컬 그룹 및 사용자, ID 페더레이션 또는 단일 로그인(SSO)을 사용하는지에 따라 결정됩니다.

또한, 그리드 관리자는 S3 테넌트 계정이 규제 요건을 준수해야 하는 경우 StorageGRID 시스템에 대해 S3 객체 잠금 설정을 활성화할 수 있습니다. S3 객체 잠금이 활성화되면 모든 S3 테넌트 계정은 규정을 준수하는 버킷을 생성하고 관리할 수 있습니다.

S3 테넌트 구성

"S3 테넌트 계정이 생성되었습니다." 후에 테넌트 관리자에 액세스하여 다음과 같은 작업을 수행할 수 있습니다.

- ID 페더레이션을 설정합니다(ID 소스가 그리드와 공유되는 경우는 제외).

- 그룹 및 사용자 관리
- 계정 복제 및 그리드 간 복제를 위해 그리드 페더레이션을 사용합니다.
- S3 액세스 키 관리
- S3 버킷 생성 및 관리
- S3 플랫폼 서비스 사용
- S3 Select 사용
- 스토리지 사용량 모니터링

테넌트 관리자를 사용하여 S3 버킷을 생성하고 관리할 수 있지만, 객체를 수집하고 관리하려면 "S3 클라이언트" 또는 "S3 콘솔"를 사용해야 합니다.

로그인 및 로그아웃 방법

StorageGRID 테넌트 관리자에 로그인

테넌트 관리자에 접속하려면 "지원되는 웹 브라우저"의 주소창에 해당 테넌트의 URL을 입력하십시오.

시작하기 전에

- 로그인 정보를 가지고 계십니다.
- 그리드 관리자가 제공한 Tenant Manager 액세스 URL이 있습니다. 해당 URL은 다음 예시 중 하나와 유사합니다.

  ```
  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
  ```

  URL에는 항상 정규화된 도메인 이름(FQDN), 관리 노드의 IP 주소 또는 관리 노드 HA 그룹의 가상 IP 주소가 포함됩니다. 또한 포트 번호, 20자리 테넌트 계정 ID 또는 둘 다 포함될 수 있습니다.

- URL에 테넌트의 20자리 계정 ID가 포함되어 있지 않으면 해당 계정 ID를 가지고 있습니다.
- 귀하는 "지원되는 웹 브라우저"을(를) 사용하고 있습니다.
- 웹 브라우저에서 쿠키가 활성화되어 있습니다.
- 귀하는 "특정 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.

단계

1. "지원되는 웹 브라우저"을(를) 시작합니다.
2. 브라우저 주소 표시줄에 테넌트 관리자에 액세스하기 위한 URL을 입력합니다.
3. 보안 경고 메시지가 표시되면 브라우저의 설치 마법사를 사용하여 인증서를 설치하십시오.
4. 테넌트 관리자에 로그인합니다.

표시되는 로그인 화면은 입력한 URL과 StorageGRID에 대해 SSO(Single Sign-On)가 구성되었는지 여부에 따라 다릅니다.

SSO를 사용하지 않음

StorageGRID가 SSO를 사용하지 않는 경우 다음 화면 중 하나가 나타납니다.

- 그리드 관리자 로그인 페이지입니다. 테넌트 로그인 링크를 선택합니다.
- 테넌트 관리자 로그인 페이지입니다. 계정 필드가 이미 입력되어 있을 수 있습니다.
  i. 테넌트의 20자리 계정 ID가 표시되지 않으면 최근 계정 목록에 테넌트 계정 이름이 있으면 해당 이름을 선택하거나 계정 ID를 입력하십시오.
  ii. 사용자 이름과 비밀번호를 입력하십시오.
  iii. *로그인*을 선택합니다.

  테넌트 관리자 대시보드가 나타납니다.

  iv. 다른 사람으로부터 초기 비밀번호를 받은 경우, 계정 보안을 위해 ***username*** > *Change password*를 선택하십시오.

SSO 사용

StorageGRID가 SSO를 사용하는 경우 다음 화면 중 하나가 나타납니다.

- 귀사 조직의 SSO 페이지.

  표준 SSO 자격 증명을 입력하고 *로그인*을 선택합니다.

- 테넌트 관리자 SSO 로그인 페이지입니다.
  i. 테넌트의 20자리 계정 ID가 표시되지 않으면 최근 계정 목록에 테넌트 계정 이름이 있으면 해당 이름을 선택하거나 계정 ID를 입력하십시오.
  ii. *로그인*을 선택합니다.
  iii. 조직의 SSO 로그인 페이지에서 표준 SSO 자격 증명을 사용하여 로그인하십시오.

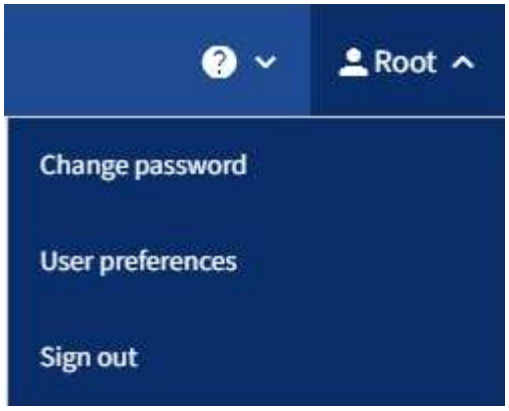
  테넌트 관리자 대시보드가 나타납니다.

StorageGRID 테넌트 관리자에서 로그아웃합니다**.**

테넌트 관리자 사용을 마친 후에는 무단 사용자가 StorageGRID 시스템에 접근하지 못하도록 반드시 로그아웃해야 합니다. 브라우저의 쿠키 설정에 따라 브라우저를 닫는 것만으로는 시스템에서 로그아웃되지 않을 수 있습니다.

단계

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



2. 사용자 이름을 선택한 다음 *로그아웃*을 선택하세요.

   ◦ SSO를 사용하지 않는 경우:

   관리자 노드에서 로그아웃되었습니다. 테넌트 관리자 로그인 페이지가 표시됩니다.

   둘 이상의 관리자 노드에 로그인한 경우, 각 노드에서 로그아웃해야 합니다.

   ◦ SSO가 활성화된 경우:

   현재 접속 중인 모든 관리자 노드에서 로그아웃되었습니다. StorageGRID 로그인 페이지가 표시됩니다. 방금 접속한 테넌트 계정 이름이 최근 계정 드롭다운 메뉴에 기본값으로 표시되고, 테넌트의 *계정 ID*가 나타납니다.

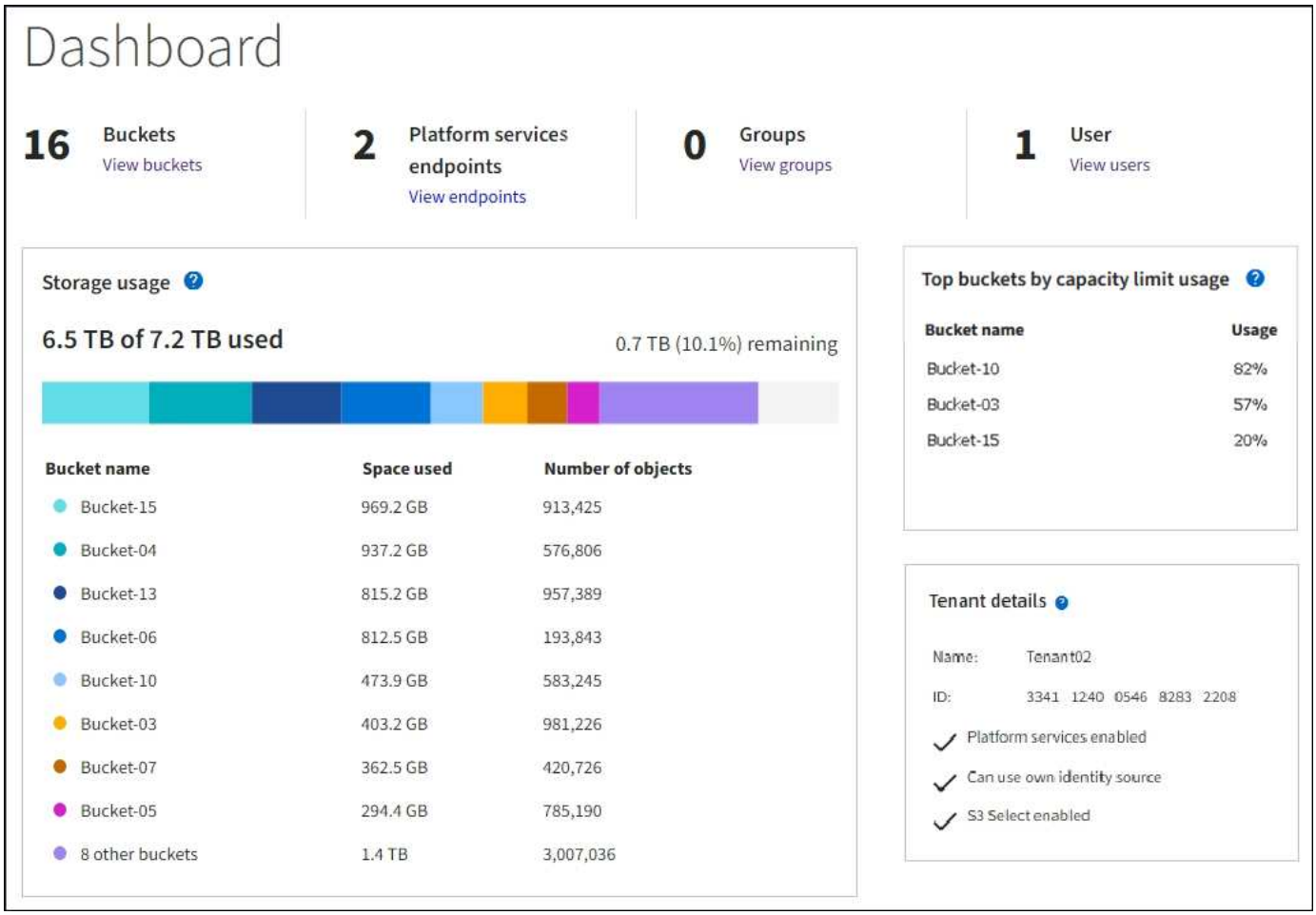
SSO가 활성화되어 있고 Grid Manager에도 로그인되어 있는 경우, SSO에서 로그아웃하려면 Grid Manager에서도 로그아웃해야 합니다.

StorageGRID의 테넌트 관리자 대시보드

테넌트 관리자 대시보드는 테넌트 계정의 구성과 테넌트의 S3 버킷에서 객체가 사용하는 공간 용량에 대한 개요를 제공합니다. 테넌트에 할당량이 있는 경우 대시보드에 할당량 사용량과 남은 용량이 표시됩니다. 테넌트 계정과 관련된 오류가 있는 경우 해당 오류가 대시보드에 표시됩니다.

이 테넌트에 속하는 모든 객체의 논리적 크기에는 미완료 및 진행 중인 멀티파트 업로드가 포함됩니다. 이 크기에는 ILM 정책에 사용되는 추가 물리적 공간은 포함되지 않습니다. 사용된 공간 값은 추정치입니다. 이러한 추정치는 데이터 수집 시간, 네트워크 연결 상태 및 노드 상태의 영향을 받습니다.

객체 업로드가 완료되면 대시보드는 다음 예시와 같이 표시됩니다.



테넌트 계정 정보

대시보드 상단에는 구성된 버킷 또는 컨테이너, 그룹 및 사용자 수가 표시됩니다. 또한 구성된 플랫폼 서비스 엔드포인트가 있는 경우 해당 엔드포인트 수도 표시됩니다. 링크를 선택하여 세부 정보를 확인하십시오.

보유한 "테넌트 관리 권한" 및 구성한 옵션에 따라 대시보드의 나머지 부분에는 다양한 지침, 스토리지 사용량, 객체 정보 및 테넌트 세부 정보의 조합이 표시됩니다.

저장 공간 및 할당량 사용량

저장 공간 사용량 패널에는 다음 정보가 포함되어 있습니다.

- 테넌트의 오브젝트 데이터 양입니다.

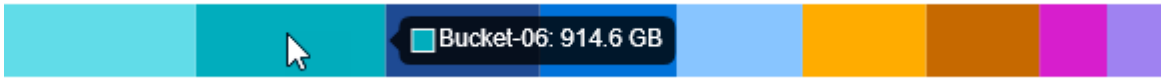
  이 값은 업로드된 객체 데이터의 총량을 나타내며, 해당 객체의 복사본과 메타데이터를 저장하는 데 사용된 공간을 나타내지 않습니다.

- 할당량이 설정된 경우, 객체 데이터에 사용할 수 있는 총 공간과 남은 공간의 양 및 비율이 표시됩니다. 할당량은 수집할 수 있는 객체 데이터의 양을 제한합니다.

할당량 사용량은 내부 추정치를 기반으로 하며 경우에 따라 초과될 수 있습니다. 예를 들어, StorageGRID는 테넌트가 객체 업로드를 시작할 때 할당량을 확인하고, 테넌트가 할당량을 초과한 경우 새로운 수집을 거부합니다. 그러나 StorageGRID는 할당량 초과 여부를 판단할 때 현재 업로드 중인 데이터의 크기를 고려하지 않습니다. 객체가 삭제되면 할당량 사용량이 재계산될 때까지 테넌트가 새 객체를 업로드하지 못할 수 있습니다. 할당량 사용량 계산에는 10분 이상 소요될 수 있습니다.

- 가장 큰 버킷 또는 컨테이너의 상대적인 크기를 나타내는 막대 그래프.

  차트의 각 부분 위에 커서를 올려놓으면 해당 버킷 또는 컨테이너에서 사용된 총 공간을 확인할 수 있습니다.



- 막대 그래프와 함께 가장 큰 버킷 또는 컨테이너 목록을 제공하며, 각 버킷 또는 컨테이너의 총 오브젝트 데이터 양과 오브젝트 수를 포함합니다.

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 버킷 또는 컨테이너가 있는 경우, 나머지 모든 버킷 또는 컨테이너는 목록 맨 아래에 하나의 항목으로 통합됩니다.

테넌트 관리자에 표시되는 저장 용량 값의 단위를 변경하려면 테넌트 관리자 오른쪽 상단에 있는 사용자 드롭다운 메뉴를 선택한 다음 *User preferences*를 선택하십시오.

할당량 사용량 알림

Grid Manager에서 할당량 사용량 알림을 활성화한 경우, 할당량이 부족하거나 초과되면 다음과 같이 Tenant Manager에 알림이 표시됩니다.

- 테넌트 할당량의 90% 이상이 사용된 경우, 테넌트 할당량 사용량 높음 알림이 트리거됩니다.

그리드 관리자에게 할당량을 늘려달라고 요청하는 것을 고려하십시오.

- 할당량을 초과하면 새 개체를 업로드할 수 없다는 알림이 표시됩니다.

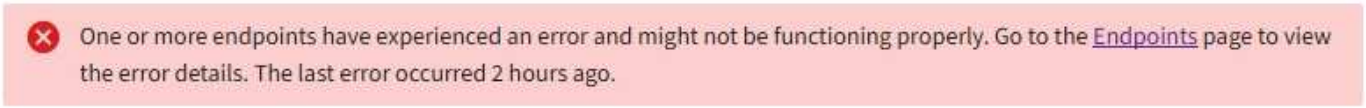
용량 제한 사용량

버킷에 용량 제한을 설정한 경우, 테넌트 관리자 대시보드에 용량 제한 사용량 기준으로 상위 버킷 목록이 표시됩니다.

버킷에 용량 제한이 설정되어 있지 않으면 용량은 무제한입니다. 하지만 테넌트 계정에 총 스토리지 할당량이 있고 해당 할당량에 도달하면 버킷의 남은 용량 제한과 관계없이 더 이상 객체를 수집할 수 없습니다.

엔드포인트 오류

그리드 관리자를 사용하여 플랫폼 서비스와 함께 사용할 하나 이상의 엔드포인트를 구성한 경우, 지난 7일 동안 엔드포인트 오류가 발생하면 테넌트 관리자 대시보드에 알림이 표시됩니다.



"플랫폼 서비스 엔드포인트 오류"에 대한 자세한 내용을 보려면 *Endpoints*를 선택하여 Endpoints 페이지를 표시하십시오.

테넌트 관리 **API**

StorageGRID 테넌트 관리 **API**에 대해 알아보십시오

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

테넌트 관리 API:

- Swagger 오픈 소스 API 플랫폼을 사용합니다. Swagger는 개발자와 비개발자 모두 API와 상호 작용할 수 있는 직관적인 사용자 인터페이스를 제공합니다. Swagger 사용자 인터페이스는 각 API 작업에 대한 자세한 정보와 문서를 제공합니다.
- "중단 없는 업그레이드를 지원하는 버전 관리"을(를) 사용합니다.

테넌트 관리 API에 대한 Swagger 문서에 액세스하려면 다음을 수행합니다.

1. 테넌트 관리자에 로그인합니다.
2. 테넌트 관리자 상단에서 도움말 아이콘을 선택한 다음 *API 문서*를 선택합니다.

API 작업

테넌트 관리 API는 사용 가능한 API 작업을 다음과 같은 섹션으로 구성합니다.

- 계정: 현재 테넌트 계정에 대한 작업(저장 공간 사용량 정보 가져오기 포함).
- **auth**: 사용자 세션 인증을 수행하는 작업입니다.

테넌트 관리 API는 베어러 토큰 인증 방식을 지원합니다. 테넌트 로그인을 위해 인증 요청의 JSON 본문에 사용자 이름, 비밀번호 및 accountId를 제공해야 합니다(즉, `POST /api/v3/authorize`). 사용자가 인증에 성공하면 보안 토큰이 반환됩니다. 이 토큰은 이후 API 요청의 헤더에 포함해야 합니다("Authorization: Bearer token").

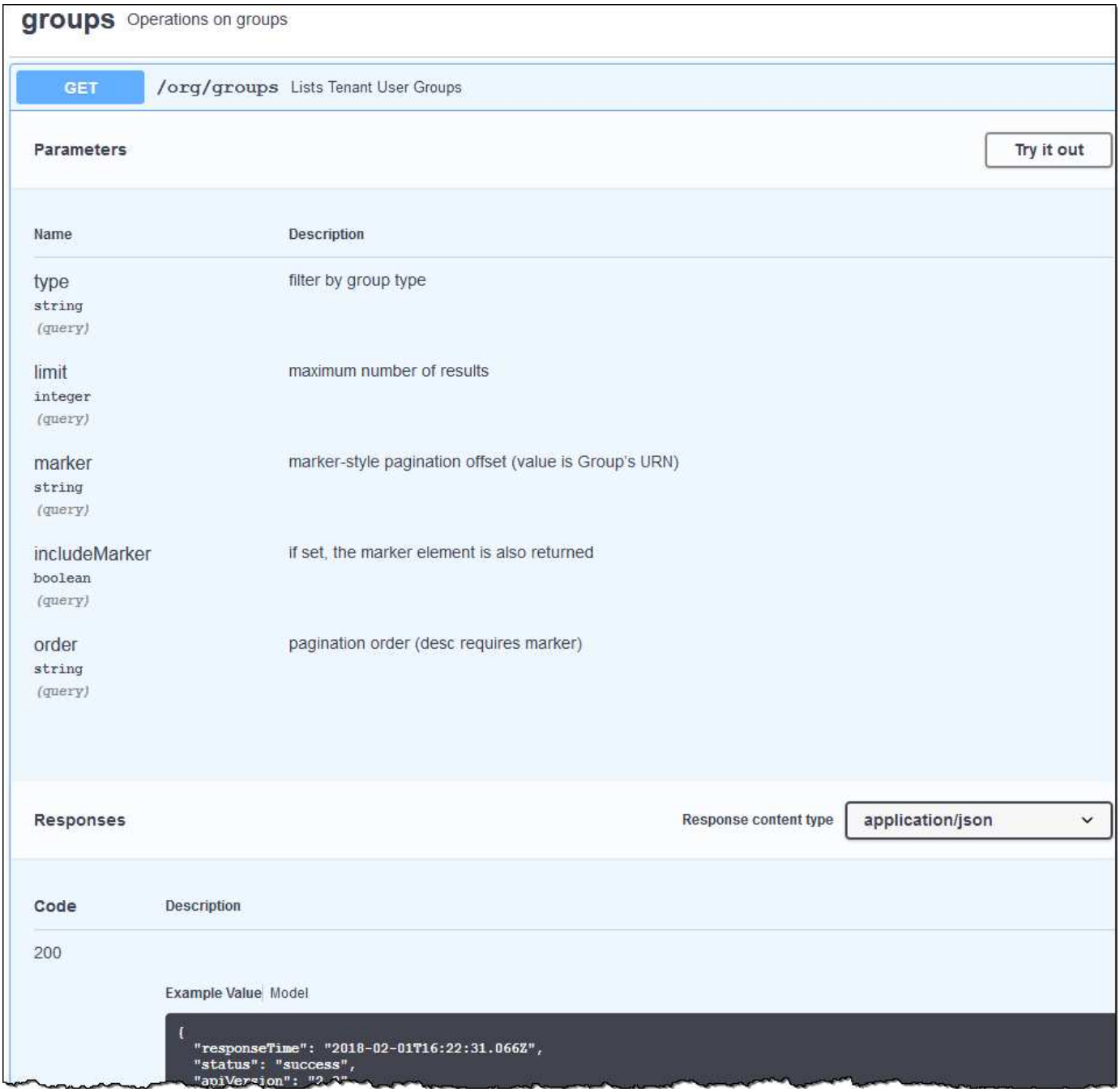
인증 보안 개선에 대한 자세한 내용은 "사이트 간 요청 위조(CSRF) 공격으로부터 보호"을 참조하십시오.

> StorageGRID 시스템에 단일 로그인(SSO)이 활성화된 경우 인증을 위해 다른 단계를 수행해야 합니다. "그리드 관리 API 사용 지침"을 참조하십시오.

- **config**: 테넌트 관리 API의 제품 릴리스 및 버전과 관련된 작업입니다. 제품 릴리스 버전과 해당 릴리스에서 지원하는 API의 주요 버전을 나열할 수 있습니다.
- 컨테이너: S3 버킷에 대한 작업.
- **deactivated-features**: 비활성화되었을 수 있는 기능을 확인하기 위한 작업입니다.
- 엔드포인트: 엔드포인트를 관리하는 작업입니다. 엔드포인트를 통해 S3 버킷은 StorageGRID CloudMirror 복제, 알림 또는 검색 통합을 위해 외부 서비스를 사용할 수 있습니다.
- **grid-federation-connections**: 그리드 페더레이션 연결 및 그리드 간 복제에 대한 작업입니다.
- **groups**: 로컬 테넌트 그룹을 관리하고 외부 ID 소스에서 페더레이션된 테넌트 그룹을 검색하는 작업입니다.
- **identity-source**: 외부 ID 소스를 구성하고 페더레이션된 그룹 및 사용자 정보를 수동으로 동기화하는 작업입니다.
- **ilm**: 정보 라이프사이클 관리(ILM) 설정에 대한 작업.
- **regions**: StorageGRID 시스템에 대해 구성된 지역을 확인하는 작업입니다.
- **s3**: 테넌트 사용자를 위한 S3 액세스 키 관리 작업입니다.
- **s3-object-lock**: 규정 준수를 지원하기 위해 사용되는 전역 S3 Object Lock 설정에 대한 작업입니다.
- 사용자: 테넌트 사용자를 보고 관리하는 작업입니다.

작업 세부 정보

각 API 작업을 펼쳐보면 HTTP 작업, 엔드포인트 URL, 필수 또는 선택적 매개변수 목록, 요청 본문 예시(필요한 경우) 및 가능한 응답을 확인할 수 있습니다.



API 요청 발행

API 문서 웹페이지를 사용하여 수행하는 모든 API 작업은 실시간 작업입니다. 실수로 구성 데이터 또는 기타 데이터를 생성, 업데이트 또는 삭제하지 않도록 주의하십시오.

단계

1. 요청 세부 정보를 보려면 HTTP 작업을 선택하십시오.
2. 요청에 그룹 또는 사용자 ID와 같은 추가 매개변수가 필요한지 확인합니다. 그런 다음 이러한 값을 가져옵니다. 필요한 정보를 얻기 위해 다른 API 요청을 먼저 수행해야 할 수도 있습니다.
3. 예제 요청 본문을 수정해야 하는지 판단하세요. 수정해야 하는 경우, *모델*을 선택하여 각 필드에 대한 요구 사항을 확인할 수 있습니다.

4. *직접 사용해보기*를 선택합니다.
5. 필요한 매개변수를 제공하거나 필요에 따라 요청 본문을 수정하십시오.
6. *실행*을 선택합니다.
7. 요청이 성공했는지 여부를 확인하려면 응답 코드를 검토하십시오.

StorageGRID Tenant Management API 버전 관리

테넌트 관리 API는 버전 관리를 사용하여 서비스 중단 없는 업그레이드를 지원합니다.

예를 들어, 이 요청 URL은 API 버전 4를 지정합니다.

`https://hostname_or_ip_address/api/v4/authorize`

API의 주 버전은 이전 버전과 호환되지 않는 변경 사항이 있을 때 업데이트됩니다. 반대로 API의 부 버전은 이전 버전과 호환되는 변경 사항이 있을 때 업데이트됩니다. 호환되는 변경 사항에는 새로운 엔드포인트 또는 새로운 속성 추가가 포함됩니다.

다음 예시는 변경 유형에 따라 API 버전이 어떻게 증가하는지 보여줍니다.

API 변경 유형	이전 버전	새 버전
이전 버전과 호환됩니다	2.1	2.2
이전 버전과 호환되지 않습니다	2.1	3.0

StorageGRID 소프트웨어를 처음 설치할 때는 최신 버전의 API만 활성화됩니다. 하지만 StorageGRID의 새 기능 릴리스로 업그레이드할 때는 적어도 하나의 StorageGRID 기능 릴리스 동안 이전 버전의 API를 계속 사용할 수 있습니다.

> ⓘ 지원되는 버전을 구성할 수 있습니다. "그리드 관리 API"에 대한 Swagger API 문서의 **config** 섹션을 참조하십시오. 모든 API 클라이언트를 최신 버전으로 업데이트한 후에는 이전 버전에 대한 지원을 비활성화해야 합니다.

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

- 응답 헤더는 "Deprecated: true"입니다.
- JSON 응답 본문에 "deprecated": true가 포함되어 있습니다.
- nms.log 파일에 더 이상 사용되지 않음을 알리는 경고 메시지가 추가됩니다. 예를 들면 다음과 같습니다.

```
Received call to deprecated v2 API at POST "/api/v2/authorize"
```

현재 릴리스에서 지원되는 API 버전 확인

지원되는 API 주요 버전 목록을 반환하려면 `GET /versions` API 요청을 사용하십시오. 이 요청은 Swagger API 문서의 **config** 섹션에 있습니다.

```
GET https://{{IP-Address}}/api/versions
{
  "responseTime": "2023-06-27T22:13:50.750Z",
  "status": "success",
  "apiVersion": "4.0",
  "data": [
    2,
    3,
    4
  ]
}
```

요청에 사용할 **API** 버전 지정

API 버전은 경로 매개변수 (`/api/v4`) 또는 헤더 (`Api-Version: 4`)를 사용하여 지정할 수 있습니다. 두 값을 모두 제공하는 경우 헤더 값이 경로 값을 재정의합니다.

```
curl https://[IP-Address]/api/v4/grid/accounts

curl -H "Api-Version: 4" https://[IP-Address]/api/grid/accounts
```

StorageGRID에서 교차 사이트 요청 위조 방지

쿠키를 사용하는 인증 방식을 강화하기 위해 CSRF 토큰을 사용하면 StorageGRID에 대한 교차 사이트 요청 위조(CSRF) 공격을 방지하는 데 도움이 됩니다. Grid Manager와 Tenant Manager는 이 보안 기능을 자동으로 활성화하며, 다른 API 클라이언트는 로그인 시 활성화 여부를 선택할 수 있습니다.

공격자는 HTTP 폼 POST와 같은 방식으로 다른 사이트에 요청을 보내 로그인한 사용자의 쿠키를 이용해 특정 요청을 발생시킬 수 있습니다.

StorageGRID는 CSRF 토큰을 사용하여 CSRF 공격으로부터 보호합니다. 이 기능이 활성화되면 특정 쿠키의 내용이 특정 헤더 또는 특정 POST 본문 매개변수의 내용과 일치해야 합니다.

이 기능을 활성화하려면 인증 중에 `csrfToken` 매개변수를 `true`으로 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"
```

이 값이 true이면 `GridCsrfToken` 쿠키는 Grid Manager 로그인 시 임의의 값으로 설정되고, `AccountCsrfToken` 쿠키는 Tenant Manager 로그인 시 임의의 값으로 설정됩니다.

쿠키가 존재하는 경우, 시스템 상태를 변경할 수 있는 모든 요청(POST, PUT, PATCH, DELETE)에는 다음 중 하나가 반드시 포함되어야 합니다.

- The `X-Csrf-Token` 헤더이며, 헤더의 값은 CSRF 토큰 쿠키의 값으로 설정됩니다.
- 폼 인코딩된 본문을 허용하는 엔드포인트의 경우: A `csrfToken` 폼 인코딩된 요청 본문 매개변수입니다.

CSRF 보호를 구성하려면 "그리드 관리 API" 또는 "테넌트 관리 API"을(를) 사용하십시오.

> CSRF 토큰 쿠키가 설정된 요청은 CSRF 공격에 대한 추가적인 보호 조치로 JSON 요청 본문을 예상하는 모든 요청에 대해 "Content-Type: application/json" 헤더를 강제 적용합니다.

그리드 페더레이션 연결 사용

StorageGRID에서 테넌트 그룹 및 사용자 복제

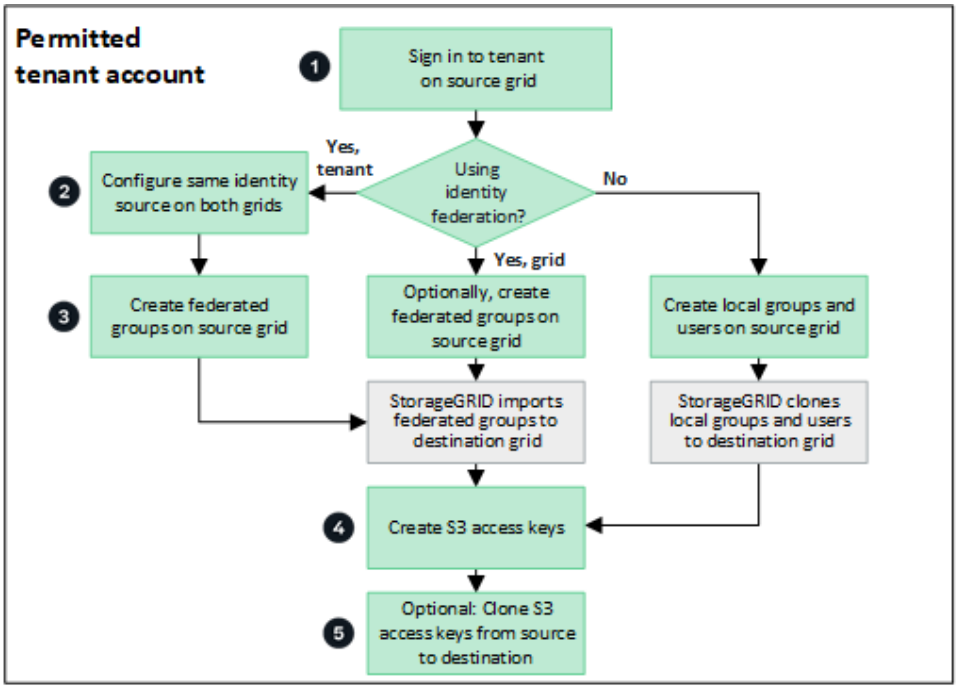
테넌트가 그리드 페더레이션 연결을 사용하도록 생성되거나 편집된 경우, 해당 테넌트는 하나의 StorageGRID 시스템(원본 테넌트)에서 다른 StorageGRID 시스템(복제본 테넌트)으로 복제됩니다. 테넌트가 복제된 후 원본 테넌트에 추가된 모든 그룹 및 사용자는 복제본 테넌트에 복제됩니다.

테넌트가 처음 생성된 StorageGRID 시스템을 테넌트의 _소스 그리드_라고 합니다. 테넌트가 복제되는 StorageGRID 시스템을 테넌트의 _대상 그리드_라고 합니다. 두 테넌트 계정은 동일한 계정 ID, 이름, 설명, 스토리지 할당량 및 할당된 권한을 가지지만, 대상 테넌트에는 초기에 루트 사용자 암호가 없습니다. 자세한 내용은 "계정 복제란 무엇인가요?" 및 "허용된 테넌트 관리"을 참조하십시오.

버킷 객체의 "교차 그리드 복제"을 위해 테넌트 계정 정보 복제가 필요합니다. 두 그리드 모두에 동일한 테넌트 그룹과 사용자가 있어야 각 그리드에서 해당 버킷과 객체에 액세스할 수 있습니다.

계정 복제를 위한 테넌트 워크플로

테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우 워크플로 다이어그램을 검토하여 그룹, 사용자 및 S3 액세스 키를 복제하는 단계를 확인하십시오.



워크플로의 주요 단계는 다음과 같습니다.

1 테넌트에 로그인

소스 그리드(테넌트가 처음 생성된 그리드)의 테넌트 계정에 로그인합니다.

2 선택적으로 **ID** 페더레이션을 구성합니다**.**

테넌트 계정에 페더레이션 그룹 및 사용자를 사용하기 위한 자체 **ID** 소스 사용 권한이 있는 경우, 소스 및 대상 테넌트 계정 모두에 동일한 ID 소스(동일한 설정 포함)를 구성하십시오. 두 그리드가 동일한 ID 소스를 사용하지 않으면 페더레이션 그룹 및 사용자를 복제할 수 없습니다. 자세한 내용은 "ID 페더레이션 사용"을 참조하십시오.

3 그룹 및 사용자 생성

그룹 및 사용자를 생성할 때는 테넌트의 소스 그리드에서 시작하십시오. 새 그룹을 추가하면 StorageGRID가 해당 그룹을 대상 그리드에 자동으로 복제합니다.

- 전체 StorageGRID 시스템 또는 테넌트 계정에 대해 ID 페더레이션이 구성된 경우 "새로운 테넌트 그룹 생성" ID 소스에서 페더레이션된 그룹을 가져옵니다.
- ID 페더레이션을 사용하지 않는 경우, "새 로컬 그룹 만들기" 그런 다음 "로컬 사용자 생성".

4 **S3** 액세스 키 생성

소스 그리드 또는 대상 그리드에서 "액세스 키 생성" 하거나 "다른 사용자의 액세스 키 생성" 하여 해당 그리드의 버킷에 액세스할 수 있습니다.

5 선택적으로 **S3** 액세스 키 복제

두 그리드에서 동일한 액세스 키로 버킷에 액세스해야 하는 경우 소스 그리드에서 액세스 키를 생성한 다음 테넌트 관리자 API를 사용하여 대상 그리드로 수동으로 복제하세요. 자세한 내용은 "API를 사용하여 S3 액세스 키 복제"을 참조하세요.

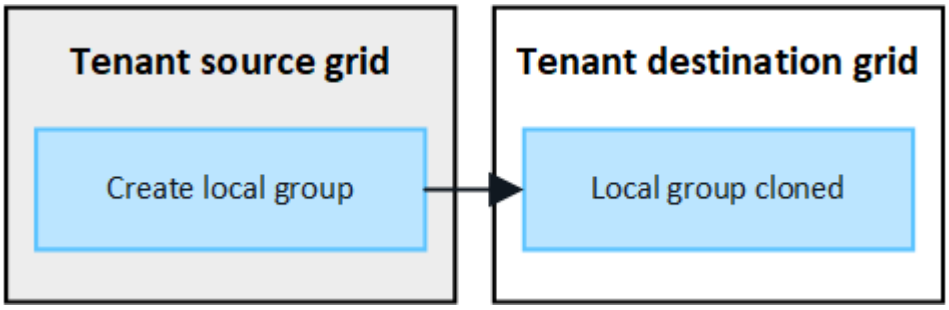
그룹, 사용자 및 **S3** 액세스 키는 어떻게 복제됩니까**?**

이 섹션을 검토하여 그룹, 사용자 및 S3 액세스 키가 테넌트 소스 그리드와 테넌트 대상 그리드 간에 어떻게 복제되는지 이해하십시오.

원본 그리드에 생성된 로컬 그룹은 복제됩니다.

테넌트 계정이 생성되고 대상 그리드로 복제되면 StorageGRID는 테넌트의 소스 그리드에 추가한 모든 로컬 그룹을 테넌트의 대상 그리드로 자동으로 복제합니다.

원본 그룹과 복제본 모두 동일한 액세스 모드, 그룹 권한 및 S3 그룹 정책을 가집니다. 자세한 내용은 "S3 테넌트에 대한 그룹 생성"을 참조하십시오.

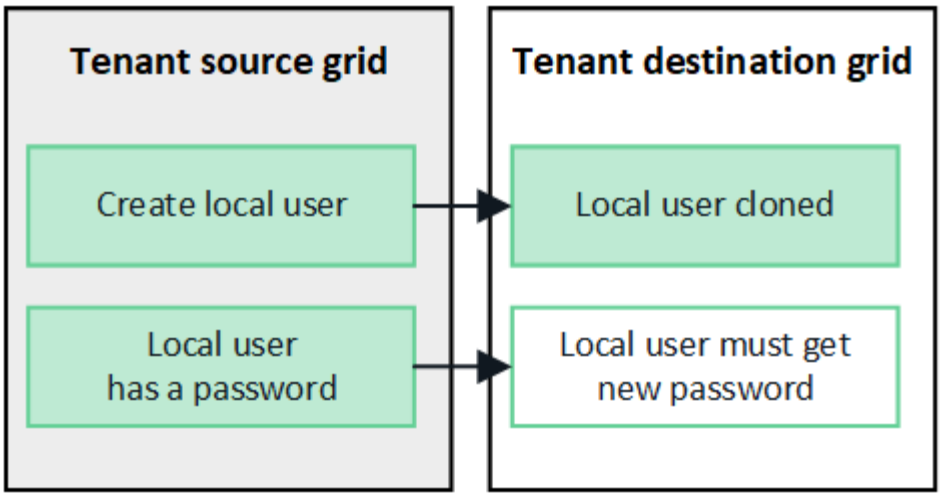


원본 그리드에서 로컬 그룹을 만들 때 선택한 사용자는 해당 그룹이 대상 그리드로 복제될 때 포함되지 않습니다. 따라서 그룹을 만들 때 사용자를 선택하지 마십시오. 대신 사용자를 만들 때 그룹을 선택하십시오.

원본 그리드에 생성된 로컬 사용자가 복제됩니다.

원본 그리드에서 새 로컬 사용자를 생성하면 StorageGRID가 해당 사용자를 대상 그리드로 자동으로 복제합니다. 원본 사용자와 복제된 사용자는 동일한 전체 이름, 사용자 이름 및 액세스 거부 설정을 갖습니다. 또한 두 사용자는 동일한 그룹에 속합니다. 자세한 내용은 "사용자 관리"을 참조하십시오.

보안상의 이유로 로컬 사용자 암호는 대상 그리드로 복제되지 않습니다. 로컬 사용자가 대상 그리드의 테넌트 관리자에 액세스해야 하는 경우, 해당 테넌트 계정의 루트 사용자가 대상 그리드에 해당 사용자의 암호를 추가해야 합니다. 자세한 내용은 "사용자 관리"을 참조하십시오.

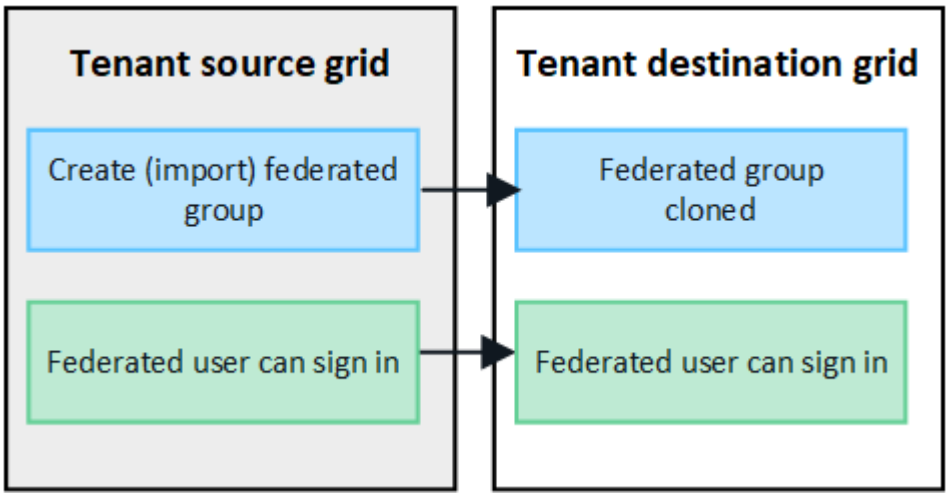


소스 그리드에서 생성된 페더레이션 그룹은 복제됩니다.

계정 복제 기능을 사용하기 위한 요구 사항이 "싱글 사인온" 및 "ID 페더레이션"에 대해 충족되었다고 가정하면, 소스 그리드의 테넌트에 대해 생성(가져오기)한 페더레이션 그룹은 대상 그리드의 테넌트로 자동으로 복제됩니다.

두 그룹 모두 동일한 액세스 모드, 그룹 권한 및 S3 그룹 정책을 가집니다.

원본 테넌트에 대해 페더레이션 그룹이 생성되고 대상 테넌트에 복제된 후, 페더레이션 사용자는 두 그리드 중 하나에서 테넌트에 로그인할 수 있습니다.

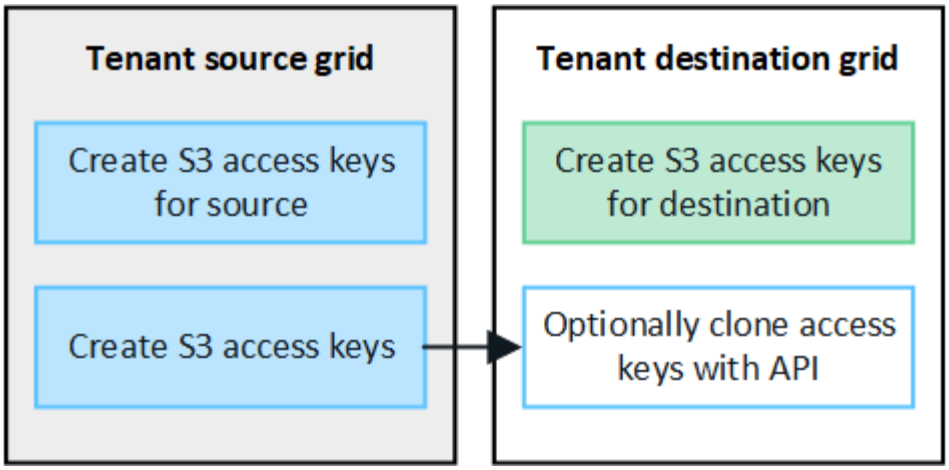


S3 액세스 키는 수동으로 복제할 수 있습니다.

StorageGRID는 각 그리드마다 서로 다른 키를 사용하면 보안이 강화되므로 S3 액세스 키를 자동으로 복제하지 않습니다.

두 그리드의 액세스 키를 관리하려면 다음 방법 중 하나를 사용할 수 있습니다.

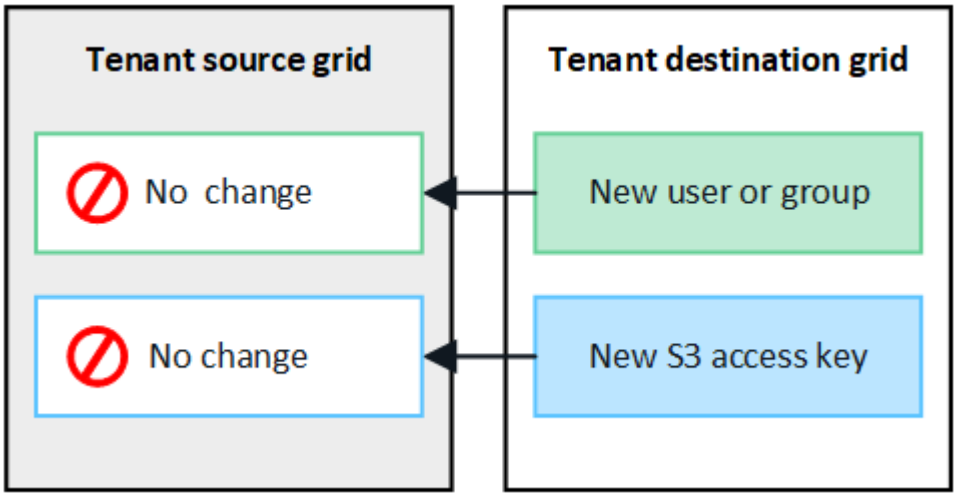
- 각 그리드에 동일한 키를 사용할 필요가 없다면 "액세스 키 생성" 또는 "다른 사용자의 액세스 키 생성" 각 그리드에서 설정할 수 있습니다.
- 두 그리드에서 동일한 키를 사용해야 하는 경우 소스 그리드에서 키를 생성한 다음 테넌트 관리자 API를 사용하여 수동으로 "키를 복제합니다" 대상 그리드에 키를 추가할 수 있습니다.



페더레이션 사용자의 S3 액세스 키를 복제하면 사용자와 S3 액세스 키 모두 대상 테넌트에 복제됩니다.

대상 그리드에 추가된 그룹과 사용자는 복제되지 않습니다.

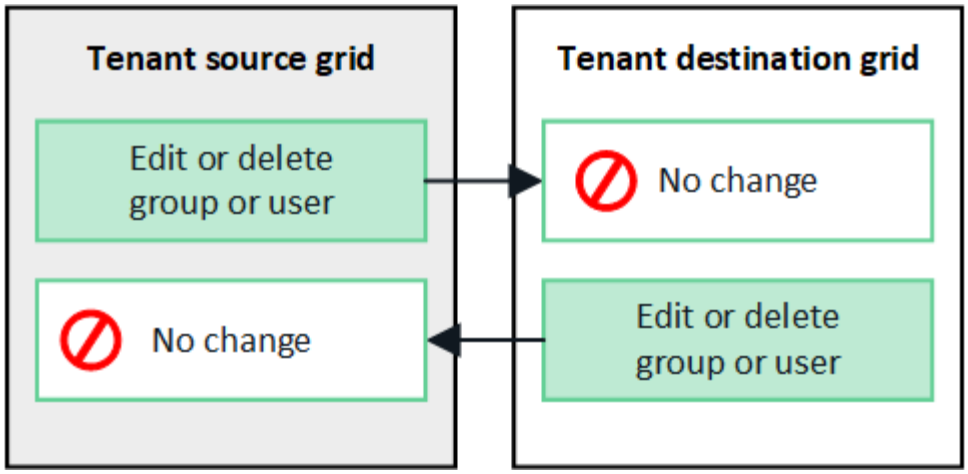
복제는 테넌트의 소스 그리드에서 테넌트의 대상 그리드로만 수행됩니다. 테넌트의 대상 그리드에서 그룹 및 사용자를 생성하거나 가져오는 경우 StorageGRID는 이러한 항목을 테넌트의 소스 그리드로 다시 복제하지 않습니다.



수정 또는 삭제된 그룹, 사용자 및 액세스 키는 복제되지 않습니다

복제는 새 그룹 및 사용자를 생성할 때만 발생합니다.

어느 한쪽 그리드에서 그룹, 사용자 또는 액세스 키를 편집하거나 삭제하면 해당 변경 사항이 다른 그리드에 복제되지 않습니다.



테넌트 관리 **API**를 사용하여 **StorageGRID** 그리드 간 **S3** 액세스 키 복제

StorageGRID 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우 테넌트 관리 API를 사용하여 소스 그리드의 테넌트에서 대상 그리드의 테넌트로 S3 액세스 키를 수동으로 복제할 수 있습니다.

시작하기 전에

- 테넌트 계정에는 그리드 페더레이션 연결 사용 권한이 있습니다.
- 그리드 연합 연결의 *연결 상태*가 *연결됨*입니다.
- 테넌트의 소스 그리드에서 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인되어 있습니다.
- 귀하는 "S3 자격 증명 또는 루트 액세스 권한을 직접 관리합니다."이(가) 있는 사용자 그룹에 속해 있습니다.
- 로컬 사용자의 액세스 키를 복제하는 경우 해당 사용자는 이미 두 그리드 모두에 존재합니다.

> ⓘ 페더레이션 사용자의 S3 액세스 키를 복제하면 사용자와 S3 액세스 키가 모두 대상 테넌트에 추가됩니다.

자신의 액세스 키 복제

두 그리드에서 동일한 버킷에 액세스해야 하는 경우 자체 액세스 키를 복제할 수 있습니다.

단계

1. 소스 그리드의 테넌트 관리자를 사용하여 "액세스 키 생성" 및 `.csv` 파일을 다운로드합니다.
2. 테넌트 관리자 상단에서 도움말 아이콘을 선택한 다음 *API 문서*를 선택합니다.
3. **s3** 섹션에서 다음 엔드포인트를 선택하십시오.

   ```
   POST /org/users/current-user/replicate-s3-access-key
   ```

4. *직접 사용해보기*를 선택합니다.
5. 본문 텍스트 상자에서 **accessKey** 및 **secretAccessKey***의 예시 항목을 다운로드한 ***.csv** 파일의 값으로

바꾸세요.

각 문자열을 둘러싼 큰따옴표를 반드시 유지하십시오.

6. 키가 만료될 예정이라면, 예시 항목인 **expires***를 **ISO 8601** 날짜-시간 형식의 문자열로 된 만료 날짜 및 시간으로 바꾸십시오(예: `2024-02-28T22:46:33-08:00`). 키가 만료되지 않을 예정이라면, ***expires** 항목의 값으로 **null***을 입력하거나 ***Expires** 줄과 그 앞의 쉼표를 삭제하십시오.
7. *실행*을 선택합니다.
8. 서버 응답 코드가 *204*인지 확인하십시오. *204*는 키가 대상 그리드에 성공적으로 복제되었음을 나타냅니다.

다른 사용자의 액세스 키 복제

두 그리드 모두에서 동일한 버킷에 액세스해야 하는 경우 다른 사용자의 액세스 키를 복제할 수 있습니다.

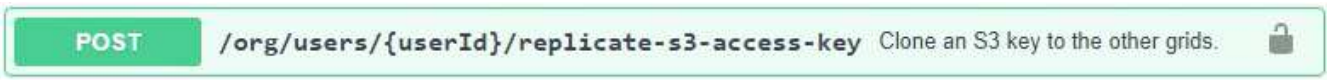
단계

1. 소스 그리드의 테넌트 관리자를 사용하여 "다른 사용자의 S3 액세스 키 생성" 및 `.csv` 파일을 다운로드합니다.
2. 테넌트 관리자 상단에서 도움말 아이콘을 선택한 다음 *API 문서*를 선택합니다.
3. 사용자 ID를 확인하세요. 다른 사용자의 액세스 키를 복제하려면 이 값이 필요합니다.
   a. 사용자 섹션에서 다음 엔드포인트를 선택하십시오.

      `GET /org/users`

   b. *직접 사용해보기*를 선택합니다.
   c. 사용자 조회 시 사용할 매개변수를 지정하십시오.
   d. *실행*을 선택합니다.
   e. 키를 복제하려는 사용자를 찾고 **id** 필드의 번호를 복사합니다.
4. **s3** 섹션에서 다음 엔드포인트를 선택하십시오.

   `POST /org/users/{userId}/replicate-s3-access-key`

5. *직접 사용해보기*를 선택합니다.
6. * userId* 텍스트 상자에 복사한 사용자 ID를 붙여넣으세요.
7. 본문 텍스트 상자에서 예시 액세스 키 및 비밀 액세스 키*의 예시 항목을 해당 사용자의 ***.csv** 파일에 있는 값으로 바꾸십시오.

문자열을 둘러싼 큰따옴표를 반드시 유지하십시오.

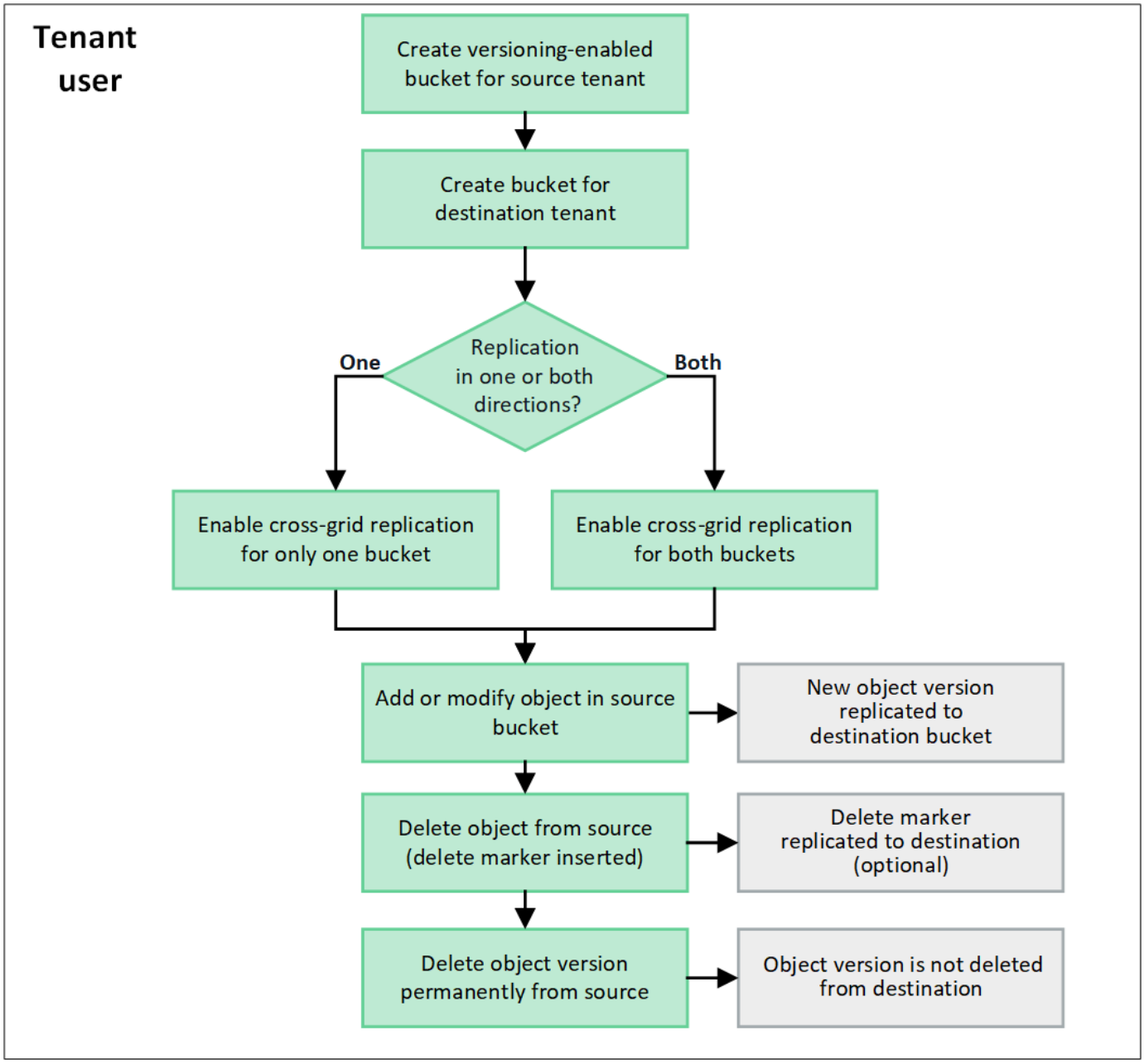
8. 키가 만료될 예정이라면, 예시 항목인 **expires***를 **ISO 8601** 날짜**-**시간 형식의 문자열로 된 만료 날짜 및 시간으로 바꾸십시오**(**예**: `2023-02-28T22:46:33-08:00`).** 키가 만료되지 않을 예정이라면**, *expires** 항목의 값으로 **null***을 입력하거나 ***Expires** 줄과 그 앞의 쉼표를 삭제하십시오.
9. *실행*을 선택합니다.
10. 서버 응답 코드가 *204*인지 확인하십시오. *204*는 키가 대상 그리드에 성공적으로 복제되었음을 나타냅니다.

StorageGRID에서 교차 그리드 복제 관리

테넌트 계정 생성 시 그리드 페더레이션 연결 사용 권한이 부여된 경우, 크로스 그리드 복제를 사용하여 테넌트의 소스 그리드 버킷과 테넌트의 대상 그리드 버킷 간에 객체를 자동으로 복제할 수 있습니다. 크로스 그리드 복제는 단방향 또는 양방향으로 수행될 수 있습니다.

그리드 간 복제 워크플로

워크플로 다이어그램은 두 그리드의 버킷 간 교차 그리드 복제를 구성하기 위해 수행하는 단계를 요약합니다. 이러한 단계는 다이어그램 아래에서 더 자세히 설명합니다.



그리드 간 복제 구성

교차 그리드 복제를 사용하려면 먼저 각 그리드에서 해당 테넌트 계정으로 로그인하고 버킷 두 개를 생성해야 합니다. 그런 다음 하나 또는 두 버킷 모두에서 교차 그리드 복제를 활성화할 수 있습니다.

시작하기 전에

- 그리드 간 복제에 필요한 사항을 검토하셨습니다. "크로스 그리드 복제란 무엇인가요?"을 참조하십시오.
- 현재 "지원되는 웹 브라우저"을(를) 사용하고 있습니다.
- 테넌트 계정에는 그리드 페더레이션 연결 사용 권한이 있으며, 동일한 테넌트 계정이 두 그리드에 모두 존재합니다. 자세한 내용은 "그리드 페더레이션 연결에 허용된 테넌트 관리"을 참조하십시오.
- 로그인하려는 테넌트 사용자는 이미 두 그리드 모두에 존재하며, "루트 액세스 권한"을(를) 가진 사용자 그룹에 속해 있습니다.
- 테넌트의 대상 그리드에 로컬 사용자로 로그인하는 경우, 테넌트 계정의 루트 사용자가 해당 그리드에서 사용자

계정에 대한 암호를 설정해 놓았습니다.

버킷 두 개 만들기

첫 번째 단계로 각 그리드에서 해당 테넌트 계정에 로그인하고 각 그리드에 버킷을 생성합니다.

단계

1. 그리드 연합 연결의 어느 그리드에서든 시작하여 새 버킷을 생성합니다.
   a. 두 그리드 모두에 존재하는 테넌트 사용자의 자격 증명을 사용하여 테넌트 계정에 로그인하십시오.

      로컬 사용자로 테넌트의 대상 그리드에 로그인할 수 없는 경우, 테넌트 계정의 루트 사용자가 사용자 계정에 암호를 설정했는지 확인하십시오.

   b. "S3 버킷을 생성합니다"의 지침을 따르세요.

      각 그리드마다 버킷 이름과 지역이 다를 수 있습니다.

   c. 객체 설정 관리 탭에서 *객체 버전 관리 활성화*를 선택합니다.
   d. StorageGRID 시스템에서 S3 Object Lock이 활성화된 경우 "S3 객체 잠금을 사용한 크로스 그리드 복제"을 참조하십시오.
   e. *버킷 생성*을 선택합니다.
   f. *완료*를 선택합니다.
2. 그리드 페더레이션 연결의 다른 그리드에서 동일한 테넌트 계정에 대한 버킷을 생성하려면 이 단계를 반복하십시오.

   필요에 따라 각 버킷은 서로 다른 지역을 사용할 수 있습니다.

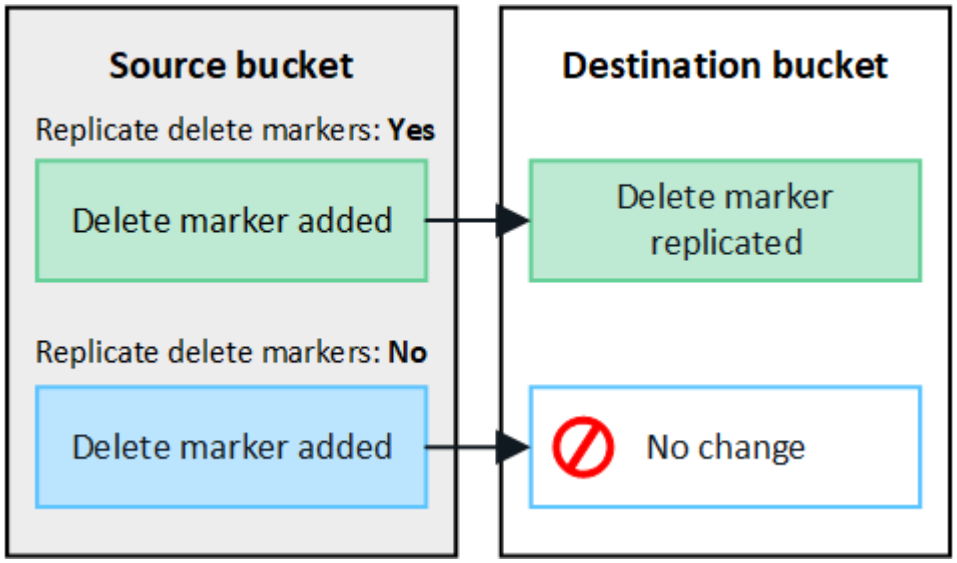
그리드 간 복제 활성화

버킷에 객체를 추가하기 전에 반드시 다음 단계를 수행해야 합니다.

단계

1. 복제하려는 객체가 있는 그리드에서 시작하여 "한 방향으로의 교차 그리드 복제"을(를) 활성화합니다.
   a. 버킷의 테넌트 계정에 로그인합니다.
   b. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
   c. 표에서 버킷 이름을 선택하여 버킷 세부 정보 페이지에 액세스하십시오.
   d. 교차 그리드 복제 탭을 선택합니다.
   e. *활성화*를 선택하고 요구 사항 목록을 검토하십시오.
   f. 모든 요구 사항이 충족되면 사용할 그리드 페더레이션 연결을 선택하십시오.
   g. 선택적으로, 삭제 마커 복제 설정을 변경하여 S3 클라이언트가 버전 ID가 포함되지 않은 삭제 요청을 소스 그리드에 보낼 경우 대상 그리드에서 발생하는 상황을 결정할 수 있습니다.

      - 예(기본값): 삭제 마커가 소스 버킷에 추가되고 대상 버킷으로 복제됩니다.
      - 아니요: 삭제 표시가 소스 버킷에 추가되지만 대상 버킷으로 복제되지 않습니다.



삭제 요청에 버전 ID가 포함된 경우 해당 객체 버전은 소스 버킷에서 영구적으로 제거됩니다. StorageGRID는 버전 ID가 포함된 삭제 요청을 복제하지 않으므로 대상 버킷에서는 동일한 객체 버전이 삭제되지 않습니다.

자세한 내용은 "크로스 그리드 복제란 무엇인가요?"을 참조하십시오.

a. 선택적으로, 감사 메시지 볼륨을 관리하기 위해 **Cross-grid replication** 감사 범주 설정을 변경할 수 있습니다.
   - **Error** (기본값): 감사 출력에는 실패한 그리드 간 복제 요청만 포함됩니다.
   - **Normal**: 모든 그리드 간 복제 요청이 포함되므로 감사 출력량이 크게 증가합니다.

b. 선택 사항을 검토하세요. 두 버킷이 모두 비어 있어야만 이 설정을 변경할 수 있습니다.

c. *활성화 및 테스트*를 선택합니다.

   잠시 후 성공 메시지가 나타납니다. 이제 이 버킷에 추가된 객체가 다른 그리드로 자동으로 복제됩니다. 버킷 세부 정보 페이지에서 *그리드 간 복제*가 활성화된 기능으로 표시됩니다.

2. 선택적으로 다른 그리드의 해당 버킷으로 이동하여 "양방향으로 그리드 간 복제를 활성화합니다.".

그리드 간 복제 테스트

버킷에 대해 그리드 간 복제가 활성화된 경우 연결 및 그리드 간 복제가 올바르게 작동하는지, 그리고 소스 및 대상 버킷이 모든 요구 사항(예: 버전 관리가 여전히 활성화됨)을 충족하는지 확인해야 할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하고 있습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.

단계

1. 버킷의 테넌트 계정에 로그인합니다.
2. 대시보드에서 버킷 보기*를 선택하거나 *스토리지(S3) > *버킷*을 선택합니다.
3. 표에서 버킷 이름을 선택하여 버킷 세부 정보 페이지에 액세스하십시오.

4. 교차 그리드 복제 탭을 선택합니다.
5. *연결 테스트*를 선택합니다.

   연결이 정상적이면 성공 배너가 표시됩니다. 그렇지 않으면 오류 메시지가 표시되며, 이 메시지를 사용하여 사용자와 그리드 관리자가 문제를 해결할 수 있습니다. 자세한 내용은 "그리드 페더레이션 오류 해결"을 참조하십시오.

6. 그리드 간 복제가 양방향으로 구성된 경우, 다른 그리드의 해당 버킷으로 이동하여 *연결 테스트*를 선택하여 그리드 간 복제가 반대 방향으로 제대로 작동하는지 확인하십시오.

그리드 간 복제 비활성화

다른 그리드로 개체를 복사하지 않으려면 그리드 간 복제를 영구적으로 중지할 수 있습니다.

그리드 간 복제를 비활성화하기 전에 다음 사항에 유의하십시오.

- 그리드 간 복제를 비활성화해도 이미 그리드 간에 복사된 객체는 삭제되지 않습니다. 예를 들어, `my-bucket`그리드 1에 있는 객체가 `my-bucket`그리드 2로 복사된 경우, 해당 버킷에 대해 그리드 간 복제를 비활성화해도 객체는 삭제되지 않습니다. 이러한 객체를 삭제하려면 수동으로 삭제해야 합니다.
- 각 버킷에 대해 그리드 간 복제가 활성화된 경우(즉, 양방향 복제가 발생하는 경우), 하나 또는 두 버킷 모두에 대해 그리드 간 복제를 비활성화할 수 있습니다. 예를 들어, `my-bucket`그리드 1에서 `my-bucket`그리드 2로의 객체 복제는 비활성화하고 `my-bucket`그리드 2에서 `my-bucket`그리드 1로의 객체 복제는 계속 유지하도록 설정할 수 있습니다.
- 테넌트 그리드 페더레이션 연결을 사용할 수 있는 테넌트의 권한을 제거하기 전에 먼저 그리드 간 복제를 비활성화해야 합니다. 을(를) 참조하십시오. "허용된 테넌트 관리"
- 객체가 포함된 버킷에 대해 크로스 그리드 복제를 비활성화하면 소스 버킷과 대상 버킷에서 모든 객체를 삭제하지 않는 한 크로스 그리드 복제를 다시 활성화할 수 없습니다.

> ⚠ 버킷 두 개가 모두 비어 있어야 복제를 다시 활성화할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하고 있습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.

단계

1. 더 이상 복제하지 않으려는 객체가 있는 그리드부터 시작하여 해당 버킷에 대한 그리드 간 복제를 중지합니다.
   a. 버킷의 테넌트 계정에 로그인합니다.
   b. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
   c. 표에서 버킷 이름을 선택하여 버킷 세부 정보 페이지에 액세스하십시오.
   d. 교차 그리드 복제 탭을 선택합니다.
   e. *복제 사용 안 함*을 선택합니다.
   f. 이 버킷에 대한 크로스 그리드 복제를 비활성화하려면 텍스트 상자에 *Yes*를 입력하고 *Disable*을 선택하십시오.

      잠시 후 성공 메시지가 나타납니다. 이 버킷에 새로 추가된 객체는 더 이상 다른 그리드로 자동 복제되지

않습니다. 버킷 페이지에서 **Cross-grid replication** 기능이 더 이상 활성화된 기능으로 표시되지 않습니다.

2. 그리드 간 복제가 양방향으로 구성되었다면, 다른 그리드의 해당 버킷으로 이동하여 반대 방향의 그리드 간 복제를 중지하십시오.

StorageGRID에서 그리드 페더레이션 연결 보기

테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우 허용된 연결을 확인할 수 있습니다.

시작하기 전에

- 테넌트 계정에는 그리드 페더레이션 연결 사용 권한이 있습니다.
- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.

단계

1. 스토리지**(S3)** > *그리드 페더레이션 연결*을 선택합니다.

   그리드 페더레이션 연결 페이지가 나타나고 다음 정보를 요약한 표가 포함됩니다.

열	설명
연결 이름	이 테넌트가 사용할 수 있는 그리드 페더레이션 연결입니다.
크로스 그리드 복제를 지원하는 버킷	각 그리드 페더레이션 연결에 대해, 그리드 간 복제가 활성화된 테넌트 버킷입니다. 이러한 버킷에 추가된 객체는 연결된 다른 그리드로 복제됩니다.
마지막 오류	각 그리드 연합 연결에 대해, 다른 그리드로 데이터가 복제되는 동안 발생한 가장 최근의 오류(있는 경우)를 표시합니다. 마지막 오류 지우기을 참조하십시오.

2. 선택적으로 버킷 이름을 "버킷 세부 정보 보기".

마지막 오류 지우기

마지막 오류 열에 오류가 나타날 수 있는 이유는 다음과 같습니다.

- 소스 객체 버전을 찾을 수 없습니다.
- 소스 버킷을 찾을 수 없습니다.
- 대상 버킷이 삭제되었습니다.
- 대상 버킷이 다른 계정에서 다시 생성되었습니다.
- 대상 버킷의 버전 관리가 일시 중단되었습니다.
- 대상 버킷은 동일한 계정으로 다시 생성되었지만 이제 버전 관리가 되지 않습니다.

이 열에는 가장 최근에 발생한 그리드 간 복제 오류만 표시됩니다. 이전에 발생했을 수 있는 오류는 표시되지 않습니다.

단계

1. 마지막 오류 열에 메시지가 표시되면 메시지 내용을 확인하십시오.

   예를 들어, 이 오류는 버전 관리가 일시 중단되었거나 S3 Object Lock이 활성화되었기 때문에 그리드 간 복제 대상 버킷이 유효하지 않은 상태였음을 나타냅니다.

2. 권장되는 조치를 수행하십시오. 예를 들어, 크로스 그리드 복제를 위해 대상 버킷에서 버전 관리가 일시 중단된 경우 해당 버킷에 대해 버전 관리를 다시 활성화하십시오.
3. 표에서 연결을 선택합니다.
4. *오류 지우기*를 선택합니다.
5. 메시지를 지우고 시스템 상태를 업데이트하려면 *예*를 선택하십시오.
6. 5~6분 정도 기다린 후 새 객체를 버킷에 수집합니다. 오류 메시지가 다시 나타나지 않는지 확인하십시오.

   오류 메시지가 완전히 사라지도록 하려면 메시지에 표시된 타임스탬프 이후 최소 5분이 지난 후에 새 객체를 수집하십시오.

7. 버킷 오류로 인해 복제에 실패한 객체가 있는지 확인하려면 "실패한 복제 작업 식별 및 재시도"을 참조하십시오.

그룹 및 사용자 관리

StorageGRID에서 ID 페더레이션 사용

ID 페더레이션을 사용하면 테넌트 그룹 및 사용자 설정이 더 빨라지고, 테넌트 사용자는 익숙한 자격 증명을 사용하여 테넌트 계정에 로그인할 수 있습니다.

테넌트 관리자에 대한 ID 페더레이션 구성

테넌트 그룹 및 사용자를 Active Directory, Microsoft Entra ID, OpenLDAP 또는 Oracle Directory Server와 같은 다른 시스템에서 관리하려면 테넌트 관리자에 대한 ID 페더레이션을 구성할 수 있습니다.

시작하기 전에

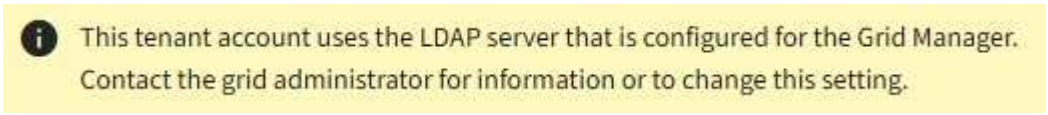
- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.
- 현재 Active Directory, Microsoft Entra ID, OpenLDAP 또는 Oracle Directory Server를 ID 공급자로 사용하고 있습니다.

  목록에 없는 LDAP v3 서비스를 사용하려면 기술 지원 부서에 문의하십시오.

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

이 작업 정보

테넌트에 대한 ID 페더레이션 서비스 구성 가능 여부는 테넌트 계정 설정 방식에 따라 다릅니다. 테넌트는 Grid Manager에 구성된 ID 페더레이션 서비스를 공유할 수도 있습니다. ID 페더레이션 페이지에 액세스할 때 이 메시지가 표시되면 해당 테넌트에 대해 별도의 페더레이션 ID 소스를 구성할 수 없습니다.

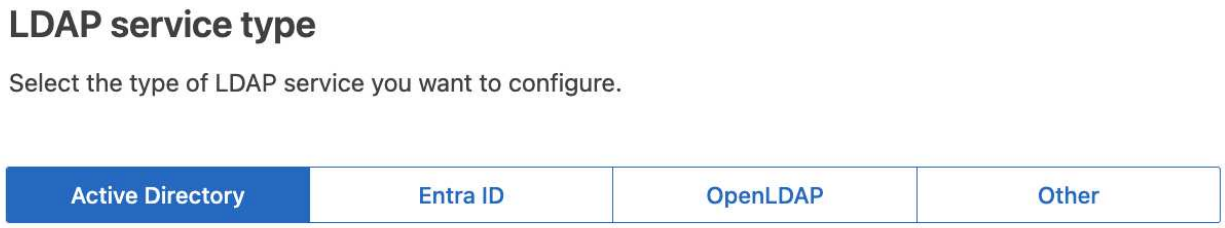


구성 입력

ID 페더레이션을 구성할 때 StorageGRID가 LDAP 서비스에 연결하는 데 필요한 값을 제공합니다.

단계

1. 액세스 관리 > *ID 페더레이션*을 선택합니다.
2. *ID 페더레이션 활성화*를 선택합니다.
3. LDAP 서비스 유형 섹션에서 구성하려는 LDAP 서비스 유형을 선택합니다.



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

4. *기타*를 선택한 경우 LDAP 속성 섹션의 필드를 작성하십시오. 그렇지 않으면 다음 단계로 진행하십시오.
   - 사용자 고유 이름: LDAP 사용자의 고유 식별자를 포함하는 속성 이름입니다. 이 속성은 Active Directory의 경우 `sAMAccountName`, OpenLDAP의 경우 `uid`와 동일합니다. Oracle Directory Server를 구성하는 경우 `uid`를 입력합니다.
   - 사용자 **UUID**: LDAP 사용자의 영구 고유 식별자를 포함하는 속성 이름입니다. 이 속성은 Active Directory의 경우 `objectGUID`와, OpenLDAP의 경우 `entryUUID`와 동일합니다. Oracle Directory Server를 구성하는

경우 `nsuniqueid`를 입력하십시오. 지정된 속성에 대한 각 사용자의 값은 16바이트 또는 문자열 형식의 32자리 16진수여야 하며, 하이픈은 무시됩니다.

- 그룹 고유 이름: LDAP 그룹의 고유 식별자를 포함하는 속성 이름입니다. 이 속성은 Active Directory의 경우 `sAMAccountName`, OpenLDAP의 경우 `cn`와 동일합니다. Oracle Directory Server를 구성하는 경우 `cn`를 입력합니다.
- 그룹 **UUID**: LDAP 그룹의 영구 고유 식별자를 포함하는 속성 이름입니다. 이 속성은 Active Directory의 경우 `objectGUID`, OpenLDAP의 경우 `entryUUID`와 동일합니다. Oracle Directory Server를 구성하는 경우 `nsuniqueid`를 입력합니다. 지정된 속성에 대한 각 그룹의 값은 16바이트 또는 문자열 형식의 32자리 16진수여야 하며, 하이픈은 무시됩니다.

5. 모든 LDAP 서비스 유형에 대해 LDAP 서버 구성 섹션에서 필요한 LDAP 서버 및 네트워크 연결 정보를 입력하십시오.
   - 호스트 이름: LDAP 서버의 정규화된 도메인 이름(FQDN) 또는 IP 주소.
   - 포트: LDAP 서버에 연결하는 데 사용되는 포트입니다.

     STARTTLS의 기본 포트는 389이고, LDAPS의 기본 포트는 636입니다. 하지만 방화벽이 올바르게 구성되어 있다면 어떤 포트든 사용할 수 있습니다.

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

     Active Directory의 경우 하위 레벨 로그온 이름 또는 사용자 주체 이름을 지정할 수도 있습니다.

     지정된 사용자는 그룹 및 사용자 목록을 볼 수 있는 권한과 다음 속성에 접근할 수 있는 권한이 있어야 합니다.

     - `sAMAccountName` 또는 `uid`
     - `objectGUID`, `entryUUID` 또는 `nsuniqueid`
     - `cn`
     - `memberOf` 또는 `isMemberOf`
     - **Active Directory**: `objectSid`, `primaryGroupID`, `userAccountControl`, 및 `userPrincipalName`
     - **Entra ID**: `accountEnabled` 및 `userPrincipalName`

   - **Password**: 사용자 이름과 연결된 비밀번호입니다.

     향후 비밀번호를 변경하는 경우, 반드시 이 페이지에서 업데이트해야 합니다.

   - 그룹 기본 **DN**: 그룹을 검색할 LDAP 하위 트리의 고유 이름(DN)의 전체 경로입니다. 아래 Active Directory 예시에서 고유 이름이 기본 DN(DC=storagegrid,DC=example,DC=com)을 기준으로 하는 모든 그룹은 페더레이션 그룹으로 사용할 수 있습니다.

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

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

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

◦ 바인드 사용자 이름 형식 (선택 사항): 패턴을 자동으로 확인할 수 없는 경우 StorageGRID에서 사용할 기본 사용자 이름 패턴입니다.

*바인딩 사용자 이름 형식*을 제공하는 것이 좋습니다. 이렇게 하면 StorageGRID 서비스 계정과 바인딩할 수 없는 경우에도 사용자가 로그인할 수 있습니다.

다음 패턴 중 하나를 입력하세요:

- **UserPrincipalName** 패턴**(AD** 및 **Entra ID)**: `[USERNAME]@example.com`
- 하위 레벨 로그온 이름 패턴**(AD** 및 **Entra ID)**: `example\[USERNAME]`
- 고유 이름 패턴: `CN=[USERNAME],CN=Users,DC=example,DC=com`

*[USERNAME]*을 정확히 입력하세요.

6. 전송 계층 보안(TLS) 섹션에서 보안 설정을 선택합니다.
   - **STARTTLS** 사용: STARTTLS를 사용하여 LDAP 서버와의 통신을 보호합니다. Active Directory, OpenLDAP 또는 기타 시스템에는 이 옵션을 권장하지만, Microsoft Entra ID에서는 이 옵션이 지원되지 않습니다.
   - **LDAPS** 사용: LDAPS(SSL을 통한 LDAP) 옵션은 TLS를 사용하여 LDAP 서버에 연결을 설정합니다. Microsoft Entra ID를 사용하려면 이 옵션을 선택해야 합니다.
   - **TLS**를 사용하지 마십시오: StorageGRID 시스템과 LDAP 서버 간의 네트워크 트래픽은 보호되지 않습니다. 이 옵션은 Microsoft Entra ID에서 지원되지 않습니다.

> Active Directory 서버에서 LDAP 서명을 강제하는 경우 **TLS**를 사용하지 않음 옵션은 지원되지 않습니다. STARTTLS 또는 LDAPS를 사용해야 합니다.

7. STARTTLS 또는 LDAPS를 선택한 경우 연결을 보호하는 데 사용되는 인증서를 선택하십시오.
   - 운영체제 **CA** 인증서 사용: 연결을 보호하기 위해 운영체제에 설치된 기본 Grid CA 인증서를 사용합니다.
   - 사용자 지정 **CA** 인증서 사용: 사용자 지정 보안 인증서를 사용합니다.

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

연결을 테스트하고 구성을 저장합니다

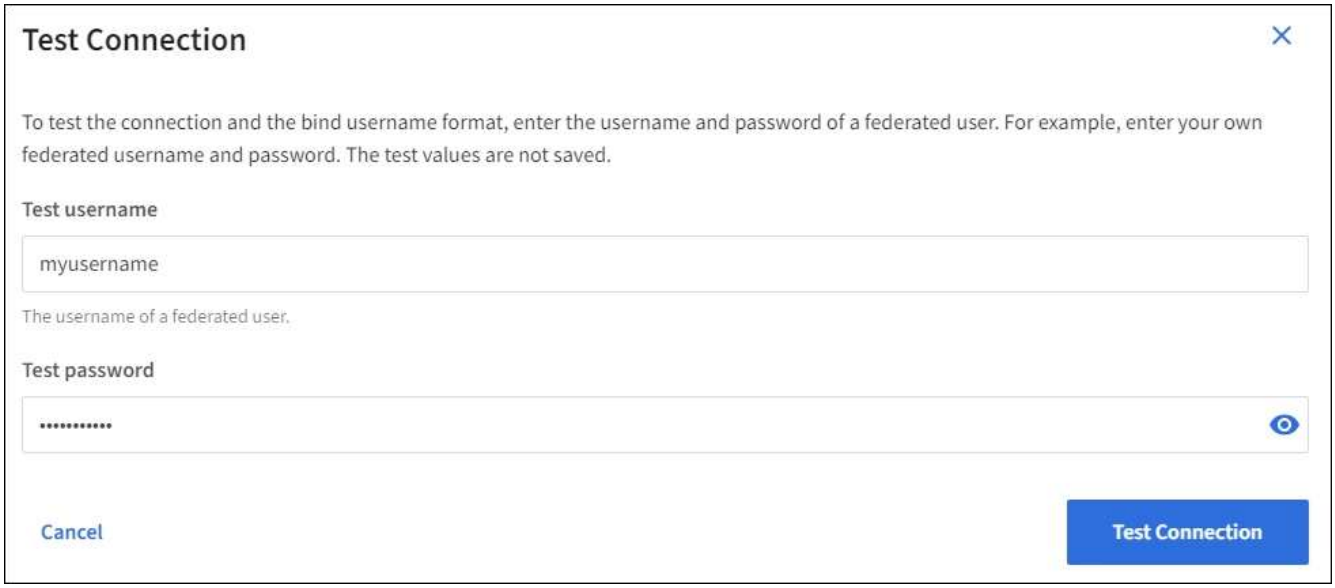
모든 값을 입력한 후 구성을 저장하기 전에 연결을 테스트해야 합니다. StorageGRID는 LDAP 서버의 연결 설정과 바인드 사용자 이름 형식(제공한 경우)을 확인합니다.

단계

1. *연결 테스트*를 선택합니다.
2. 바인딩 사용자 이름 형식을 제공하지 않은 경우:
   - 연결 설정이 유효하면 "연결 테스트 성공" 메시지가 나타납니다. *저장*을 선택하여 구성을 저장합니다.
   - 연결 설정이 올바르지 않으면 "테스트 연결을 설정할 수 없습니다"라는 메시지가 나타납니다. *닫기*를 선택합니다. 그런 다음 문제를 해결하고 연결을 다시 테스트하십시오.
3. 바인딩 사용자 이름 형식을 제공한 경우 유효한 페더레이션 사용자의 사용자 이름과 암호를 입력하십시오.

   예를 들어, 본인의 사용자 이름과 비밀번호를 입력하십시오. 사용자 이름에 @ 또는 /와 같은 특수 문자를 포함하지

마십시오.



- 연결 설정이 유효하면 "연결 테스트 성공" 메시지가 나타납니다. *저장*을 선택하여 구성을 저장합니다.
- 연결 설정, 바인딩 사용자 이름 형식 또는 테스트 사용자 이름 및 암호가 유효하지 않으면 오류 메시지가 나타납니다. 문제를 해결한 후 연결을 다시 테스트하십시오.

ID 소스와의 강제 동기화

StorageGRID 시스템은 ID 소스에서 페더레이션 그룹 및 사용자를 주기적으로 동기화합니다. 사용자 권한을 최대한 빨리 활성화하거나 제한하려면 동기화를 강제로 시작할 수 있습니다.

단계

1. ID 연동 페이지로 이동합니다.
2. 페이지 상단에서 *동기화 서버*를 선택합니다.

   동기화 과정은 사용 환경에 따라 다소 시간이 걸릴 수 있습니다.

   ID 페더레이션 동기화 실패 경고는 ID 소스에서 페더레이션된 그룹 및 사용자를 동기화하는 데 문제가 발생할 경우 발생합니다.

ID 연동 비활성화

그룹 및 사용자에 대한 ID 페더레이션을 일시적 또는 영구적으로 비활성화할 수 있습니다. ID 페더레이션이 비활성화되면 StorageGRID 와 ID 소스 간의 통신이 중단됩니다. 하지만 구성한 설정은 모두 유지되므로 나중에 ID 페더레이션을 쉽게 다시 활성화할 수 있습니다.

이 작업 정보

ID 페더레이션을 비활성화하기 전에 다음 사항을 숙지해야 합니다.

- 페더레이션 사용자는 로그인할 수 없습니다.
- 현재 로그인 중인 페더레이션 사용자는 세션이 만료될 때까지 StorageGRID 시스템에 계속 액세스할 수 있지만, 세션이 만료된 후에는 로그인할 수 없습니다.

- StorageGRID 시스템과 ID 소스 간의 동기화는 발생하지 않으며, 동기화되지 않은 계정에 대한 알림도 표시되지 않습니다.
- 싱글 사인온(SSO) 상태가 '사용' 또는 '샌드박스 모드'인 경우 'ID 페더레이션 사용' 확인란이 비활성화됩니다. ID 페더레이션을 비활성화하려면 싱글 사인온 페이지의 SSO 상태가 '사용 안 함'이어야 합니다. "싱글 사인온 비활성화"을 참조하십시오.

단계

1. ID 연동 페이지로 이동합니다.
2. **ID** 페더레이션 활성화 확인란의 선택을 해제합니다.

OpenLDAP 서버 구성 지침

ID 페더레이션을 위해 OpenLDAP 서버를 사용하려면 OpenLDAP 서버에서 특정 설정을 구성해야 합니다.

> Active Directory 또는 Microsoft Entra ID가 아닌 ID 소스의 경우, StorageGRID는 외부에서 사용이 중지된 사용자의 S3 액세스를 자동으로 차단하지 않습니다. S3 액세스를 차단하려면 해당 사용자의 S3 키를 삭제하거나 모든 그룹에서 해당 사용자를 제거하십시오.

Memberof 및 **refint** 오버레이

memberof 및 refint 오버레이를 활성화해야 합니다. 자세한 내용은 "OpenLDAP 문서: 버전 2.4 관리자 가이드"의 역방향 그룹 멤버십 유지 관리 지침을 참조하십시오.

인덱싱

지정된 인덱스 키워드를 사용하여 다음 OpenLDAP 속성을 구성해야 합니다.

- `olcDbIndex: objectClass eq`
- `olcDbIndex: uid eq,pres,sub`
- `olcDbIndex: cn eq,pres,sub`
- `olcDbIndex: entryUUID eq`

또한, 최적의 성능을 위해 사용자 이름에 대한 도움말에 언급된 필드가 색인화되었는지 확인하십시오.

역방향 그룹 멤버십 유지 관리에 대한 정보는 "OpenLDAP 문서: 버전 2.4 관리자 가이드"을 참조하십시오.

테넌트 그룹 관리

StorageGRID에서 **S3** 테넌트에 대한 그룹 생성

페더레이션 그룹을 가져오거나 로컬 그룹을 생성하여 S3 사용자 그룹의 권한을 관리할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.
- 연합 그룹을 가져오려는 경우, "구성된 ID 페더레이션", 해당 연합 그룹이 구성된 ID 소스에 이미 존재합니다.

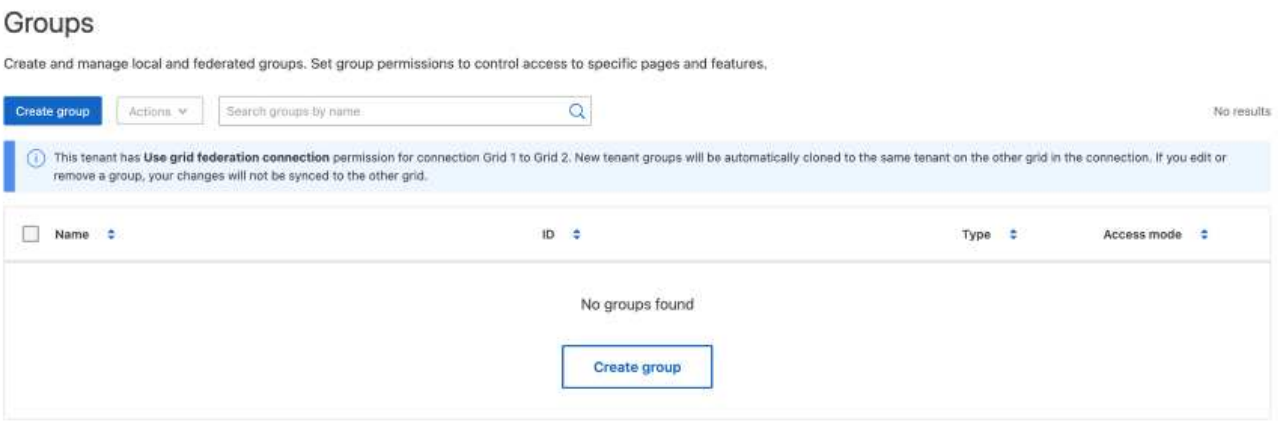
- 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 워크플로 및 고려 사항을 검토했으며 "테넌트 그룹 및 사용자 복제", 테넌트의 소스 그리드에 로그인되어 있어야 합니다.

그룹 생성 마법사에 액세스합니다.

첫 번째 단계로 그룹 생성 마법사에 액세스합니다.

단계

1. 액세스 관리 > *그룹*을 선택합니다.
2. 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 이 그리드에서 생성된 새 그룹이 연결된 다른 그리드의 동일한 테넌트에도 복제된다는 것을 나타내는 파란색 배너가 표시되는지 확인하십시오. 이 배너가 표시되지 않으면 테넌트의 대상 그리드에 로그인되어 있을 수 있습니다.



3. *그룹 생성*을 선택합니다.

그룹 유형을 선택합니다

로컬 그룹을 생성하거나 연합 그룹을 가져올 수 있습니다.

단계

1. 로컬 그룹을 생성하려면 로컬 그룹 탭을 선택하고, 이전에 구성한 ID 소스에서 그룹을 가져오려면 연합 그룹 탭을 선택하십시오.

   StorageGRID 시스템에 단일 로그인(SSO)이 활성화된 경우 로컬 그룹에 속한 사용자는 테넌트 관리자에 로그인할 수 없지만, 그룹 권한에 따라 클라이언트 애플리케이션을 사용하여 테넌트의 리소스를 관리할 수 있습니다.

2. 그룹 이름을 입력하세요.
   - 로컬 그룹: 표시 이름과 고유 이름을 모두 입력하세요. 표시 이름은 나중에 수정할 수 있습니다.

     테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 대상 그리드의 테넌트에 동일한 *고유 이름*이 이미 존재하면 복제 오류가 발생합니다.

   - 연합 그룹: 고유한 이름을 입력하십시오. Active Directory의 경우 고유 이름은 `sAMAccountName` 속성과 연결된 이름입니다. OpenLDAP의 경우 고유 이름은 `uid` 속성과 연결된 이름입니다.
3. *Continue*를 선택합니다.

그룹 권한 관리

그룹 권한은 사용자가 테넌트 관리자 및 테넌트 관리 API에서 수행할 수 있는 작업을 제어합니다.

단계

1. *액세스 모드*의 경우 다음 중 하나를 선택하십시오.
   - 읽기-쓰기 (기본값): 사용자는 Tenant Manager에 로그인하여 테넌트 구성을 관리할 수 있습니다.
   - 읽기 전용: 사용자는 설정 및 기능만 볼 수 있습니다. 테넌트 관리자 또는 테넌트 관리 API에서 변경하거나 작업을 수행할 수 없습니다. 로컬 읽기 전용 사용자는 자신의 비밀번호를 변경할 수 있습니다.

   > ⓘ 사용자가 여러 그룹에 속해 있고 어떤 그룹이라도 읽기 전용으로 설정되어 있는 경우, 해당 사용자는 선택된 모든 설정 및 기능에 대해 읽기 전용 액세스 권한만 갖게 됩니다.

2. 이 그룹에 대해 하나 이상의 권한을 선택하십시오.

   "테넌트 관리 권한"을 참조하십시오.

3. *Continue*를 선택합니다.

S3 그룹 정책 설정

그룹 정책은 사용자가 가질 S3 액세스 권한을 결정합니다.

단계

1. 이 그룹에 적용할 정책을 선택하십시오.

그룹 정책	설명
S3 액세스 불가	기본값입니다. 이 그룹의 사용자는 버킷 정책을 통해 접근 권한이 부여되지 않는 한 S3 리소스에 접근할 수 없습니다. 이 옵션을 선택하면 기본적으로 루트 사용자만 S3 리소스에 접근할 수 있습니다.
읽기 전용 액세스	이 그룹의 사용자는 S3 리소스에 대한 읽기 전용 액세스 권한을 갖습니다. 예를 들어, 이 그룹의 사용자는 객체 목록을 보고 객체 데이터, 메타데이터 및 태그를 읽을 수 있습니다. 이 옵션을 선택하면 읽기 전용 그룹 정책에 대한 JSON 문자열이 텍스트 상자에 나타납니다. 이 문자열은 편집할 수 없습니다.
전체 액세스	이 그룹의 사용자는 버킷을 포함한 S3 리소스에 대한 모든 권한을 갖습니다. 이 옵션을 선택하면 전체 액세스 그룹 정책에 대한 JSON 문자열이 텍스트 상자에 나타납니다. 이 문자열은 편집할 수 없습니다.
랜섬웨어 완화	이 예시 정책은 해당 테넌트의 모든 버킷에 적용됩니다. 이 그룹의 사용자는 일반적인 작업을 수행할 수 있지만, 객체 버전 관리가 활성화된 버킷에서 객체를 영구적으로 삭제할 수는 없습니다. 모든 버킷 관리 권한이 있는 Tenant Manager 사용자는 이 그룹 정책을 재정의할 수 있습니다. 모든 버킷 관리 권한은 신뢰할 수 있는 사용자에게만 부여하고, 가능한 경우 다단계 인증(MFA)을 사용하십시오.

그룹 정책	설명
사용자 지정	${post_edited_translations.segment}

2. *사용자 지정*을 선택한 경우 그룹 정책을 입력합니다. 각 그룹 정책의 크기 제한은 5,120바이트입니다. 유효한 JSON 형식의 문자열을 입력해야 합니다.

   언어 구문 및 예제를 포함한 그룹 정책에 대한 자세한 내용은 "예시 그룹 정책"을 참조하십시오.

3. 로컬 그룹을 생성하는 경우 *계속*을 선택하십시오. 연합 그룹을 생성하는 경우 *그룹 생성*을 선택한 다음 *완료*를 선택하십시오.

사용자 추가 **(**로컬 그룹만 해당**)**

사용자를 추가하지 않고 그룹을 저장할 수도 있고, 선택적으로 이미 존재하는 로컬 사용자를 추가할 수도 있습니다.

> 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 소스 그리드에서 로컬 그룹을 만들 때 선택한 사용자는 해당 그룹이 타겟 클러스터로 복제될 때 포함되지 않습니다. 따라서 그룹을 만들 때 사용자를 선택하지 마십시오. 대신, 사용자를 만들 때 그룹을 선택하십시오.

단계

1. 선택적으로, 이 그룹에 대해 한 명 이상의 로컬 사용자를 선택할 수 있습니다.
2. 그룹 만들기 및 *완료*를 선택합니다.

   생성한 그룹이 그룹 목록에 나타납니다.

   테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에 있는 경우, 새 그룹이 테넌트의 대상 그리드로 복제됩니다. 그룹 세부 정보 페이지의 개요 섹션에서 *복제 상태*가 *성공*으로 표시됩니다.

StorageGRID 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 사용자는 하나 이상의 권한을 가진 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 설정 및 기능을 보기만 할 수 있는지를 결정합니다.

> 사용자가 여러 그룹에 속해 있고 어떤 그룹이라도 읽기 전용으로 설정되어 있는 경우, 해당 사용자는 선택된 모든 설정 및 기능에 대해 읽기 전용 액세스 권한만 갖게 됩니다.

그룹에 다음과 같은 권한을 할당할 수 있습니다.

권한	설명	세부 정보
루트 액세스	테넌트 관리자 및 테넌트 관리 API에 대한 전체 액세스 권한을 제공합니다.	
S3 자격 증명 직접 관리	사용자가 자신의 S3 액세스 키를 생성하고 삭제할 수 있도록 합니다.	이 권한이 없는 사용자는 **STORAGE (S3)** > **My S3 access keys** 메뉴 옵션을 볼 수 없습니다.
모든 버킷 보기	사용자가 모든 버킷과 버킷 구성을 볼 수 있도록 합니다.	모든 버킷 보기 또는 모든 버킷 관리 권한이 없는 사용자는 버킷 메뉴 옵션을 볼 수 없습니다. 이 권한은 '모든 버킷 관리' 권한으로 대체됩니다. S3 클라이언트 또는 S3 Console에서 사용하는 S3 버킷 또는 그룹 정책에는 영향을 미치지 않습니다.
${post_edited_translations.segment}	사용자는 테넌트 관리자 및 테넌트 관리 API를 사용하여 S3 버킷을 생성 및 삭제하고, S3 버킷 또는 그룹 정책과 관계없이 테넌트 계정의 모든 S3 버킷에 대한 설정을 관리할 수 있습니다.	모든 버킷 보기 또는 모든 버킷 관리 권한이 없는 사용자는 버킷 메뉴 옵션을 볼 수 없습니다. 이 권한은 '모든 버킷 보기' 권한보다 우선합니다. S3 클라이언트 또는 S3 Console에서 사용하는 S3 버킷 또는 그룹 정책에는 영향을 미치지 않습니다.
엔드포인트 관리	사용자는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 StorageGRID 플랫폼 서비스의 대상으로 사용되는 플랫폼 서비스 엔드포인트를 생성하거나 편집할 수 있습니다.	이 권한이 없는 사용자는 플랫폼 서비스 엔드포인트 메뉴 옵션을 볼 수 없습니다.
S3 콘솔 탭 사용	'모든 버킷 보기' 또는 '모든 버킷 관리' 권한과 함께 사용하면 사용자는 버킷 세부 정보 페이지의 S3 Console 탭에서 객체를 보고 관리할 수 있습니다.	

StorageGRID 테넌트 그룹 관리

${post_edited_translations.segment}

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.

그룹 보기 또는 편집

각 그룹의 기본 정보와 세부 정보를 확인하고 편집할 수 있습니다.

단계

1. 액세스 관리 > *그룹*을 선택합니다.

2. 그룹 페이지에 제공된 정보를 검토하십시오. 이 페이지에는 해당 테넌트 계정에 대한 모든 로컬 및 페더레이션 그룹의 기본 정보가 나열되어 있습니다.

   테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에서 그룹을 보고 있는 경우:

   - 배너 메시지에는 그룹을 편집하거나 삭제하면 변경 사항이 다른 그리드에 동기화되지 않는다는 내용이 표시됩니다.
   - 필요에 따라, 그룹이 대상 그리드의 테넌트로 복제되지 않은 경우 배너 메시지가 표시됩니다. 실패한 항목을 그룹 클론 재시도 수 있습니다.

3. 그룹 이름을 변경하려면:
   a. 해당 그룹의 확인란을 선택합니다.
   b. * 작업 * > * 그룹 이름 편집 * 을 선택합니다.
   c. 새 이름을 입력합니다.
   d. *변경 사항 저장*을 선택합니다.
4. 더 자세한 내용을 보거나 추가 수정을 하려면 다음 중 하나를 수행하십시오.
   - 그룹 이름을 선택합니다.
   - 해당 그룹의 확인란을 선택한 다음 작업 > *그룹 세부 정보 보기*를 선택합니다.
5. 개요 섹션을 검토하십시오. 이 섹션에는 각 그룹에 대한 다음 정보가 나와 있습니다.
   - 표시 이름
   - 고유한 이름
   - 유형
   - 액세스 모드
   - 권한
   - S3 정책
   - 이 그룹의 사용자 수
   - 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에서 그룹을 보고 있는 경우 추가 필드:
     - 클론 생성 상태 (성공 또는 실패)
     - 이 그룹을 편집하거나 삭제하면 변경 사항이 다른 그리드에 동기화되지 않음을 나타내는 파란색 배너가 표시됩니다.
6. 필요에 따라 그룹 설정을 편집하세요. 입력할 내용에 대한 자세한 내용은 "S3 테넌트에 대한 그룹 생성"을 참조하세요.
   a. 개요 섹션에서 이름 또는 편집 아이콘  을 선택하여 표시 이름을 변경합니다.
   b. 그룹 권한 탭에서 권한을 업데이트하고 *변경 내용 저장*을 선택합니다.
   c. 그룹 정책 탭에서 변경 사항을 적용하고 *변경 내용 저장*을 선택합니다.

      선택적으로 다른 S3 그룹 정책을 선택하거나 필요에 따라 사용자 지정 정책에 대한 JSON 문자열을 입력합니다.

7. 기존 로컬 사용자를 하나 이상 그룹에 추가하려면 다음 단계를 따르십시오.

a. 사용자 탭을 선택합니다.

b. *사용자 추가*를 선택합니다.

c. 추가할 기존 사용자를 선택하고 *사용자 추가*를 선택합니다.

성공 메시지가 오른쪽 상단에 나타납니다.

8. 로컬 사용자를 그룹에서 제거하려면:

a. 사용자 탭을 선택합니다.

b. *사용자 제거*를 선택합니다.

c. 삭제할 사용자를 선택하고 *사용자 삭제*를 선택합니다.

성공 메시지가 오른쪽 상단에 나타납니다.

9. 변경한 각 섹션에 대해 *변경 내용 저장*을 선택했는지 확인하십시오.

중복 그룹

기존 그룹을 복제하면 새 그룹을 더 빠르게 만들 수 있습니다.

테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 원본 그리드에서 그룹을 복제하면 복제된 그룹이 테넌트의 대상 그리드에도 복제됩니다.

단계

1. 액세스 관리 > *그룹*을 선택합니다.
2. 복제하려는 그룹의 확인란을 선택합니다.
3. 작업 > *그룹 복제*를 선택합니다.
4. 입력할 내용에 대한 자세한 내용은 "S3 테넌트에 대한 그룹 생성"을 참조하십시오.
5. *그룹 생성*을 선택합니다.

그룹 클론 재시도

실패한 클론을 다시 시도하려면:

1. 그룹 이름 아래에 _(복제 실패)_라고 표시된 각 그룹을 선택하십시오.
2. 작업 > *그룹 복제*를 선택합니다.
3. 복제하려는 각 그룹의 세부 정보 페이지에서 복제 작업의 상태를 확인할 수 있습니다.

자세한 내용은 "테넌트 그룹 및 사용자 복제"을 참조하십시오.

하나 이상의 그룹 삭제

하나 이상의 그룹을 삭제할 수 있습니다. 삭제된 그룹에만 속한 사용자는 더 이상 테넌트 관리자에 로그인하거나 해당 테넌트 계정을 사용할 수 없습니다.

> 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 그룹을 삭제하면 StorageGRID는 다른 그리드에서 해당 그룹을 삭제하지 않습니다. 이 정보를 동기화 상태로 유지해야 하는 경우, 양쪽 그리드에서 동일한 그룹을 삭제해야 합니다.

단계

1. 액세스 관리 > *그룹*을 선택합니다.
2. 삭제하려는 각 그룹의 확인란을 선택합니다.
3. 작업 > 그룹 삭제 또는 작업 > *그룹 삭제*를 선택합니다.

   확인 대화 상자가 나타납니다.

4. 그룹 삭제 또는 *그룹 삭제*를 선택합니다.

AssumeRole 설정

시작하기 전에

AssumeRole을 설정하려면 관리자여야 합니다.

이 작업 정보

AssumeRole을 설정하려면 대상 그룹이 없는 경우 인수할 대상 그룹을 생성합니다. 그룹의 S3 정책을 편집하여 이 그룹을 인수할 때 허용되는 작업을 지정합니다. 그룹의 S3 트러스트 정책을 편집하여 AssumeRole API를 사용하여 그룹을 인수할 수 있는 신뢰할 수 있는 사용자를 지정합니다.

이 그룹에 대한 권한을 위임받아 생성된 임시 보안 자격 증명은 제한된 기간 동안만 유효합니다. 세션 지속 시간은 15분에서 12시간 사이이며, 기본 세션 지속 시간은 1시간입니다. 사용자를 그룹의 S3 신뢰 정책에서 제거하면 해당 사용자는 더 이상 이 그룹에 대한 권한을 위임받을 수 없습니다.

단계

1. 액세스 관리 > *그룹*을 선택합니다.
2. 그룹 이름을 클릭합니다.
3. **S3** 신뢰 정책 탭을 선택합니다.
4. S3 신뢰 정책을 추가합니다. 여기에는 AssumeRole을 수행할 수 있는 사용자 목록이 포함됩니다.
5. *변경 사항 저장*을 선택합니다.
6. **S3** 그룹 정책 탭을 선택합니다.
7. 이 그룹의 S3 신뢰 정책에 추가된 신뢰 사용자에게 필요한 S3 작업만 지정하도록 S3 정책을 편집하십시오.
8. *변경 사항 저장*을 선택합니다.

AssumeRole S3 신뢰 정책의 예시

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {
                "AWS": [
                    "urn:sgws:identity::1234567890:user/user1",
                    "arn:aws:iam::1234567890:user/user2"
                ]
            }
        }
    ]
}
```

구성이 완료되면 S3 신뢰 정책에 등록된 사용자는 AssumeRole을 수행하고 자격 증명을 받을 수 있습니다. 최종 권한은 그룹 정책, 버킷 정책 및 세션 정책에 따라 결정됩니다. 자세한 내용은 "액세스 정책 사용"을 참조하십시오.

StorageGRID 테넌트 사용자 관리

로컬 사용자를 생성하고 로컬 그룹에 할당하여 해당 사용자가 액세스할 수 있는 기능을 지정할 수 있습니다. 페더레이션된 사용자를 가져올 수도 있습니다. 테넌트 관리자에는 "root"라는 이름의 미리 정의된 로컬 사용자가 하나 포함되어 있습니다. 로컬 사용자는 추가 및 삭제할 수 있지만, 루트 사용자는 삭제할 수 없습니다.

StorageGRID 시스템에 단일 로그인(SSO)이 활성화된 경우 로컬 사용자는 Tenant Manager 또는 테넌트 관리 API에 로그인할 수 없지만, 그룹 권한에 따라 클라이언트 애플리케이션을 사용하여 테넌트의 리소스에 액세스할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.
- 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 워크플로 및 고려 사항을 검토했으며 "테넌트 그룹 및 사용자 복제", 테넌트의 소스 그리드에 로그인되어 있어야 합니다.

로컬 사용자 생성

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

어떤 그룹에도 속하지 않은 S3 사용자는 관리 권한이 없으며 S3 그룹 정책도 적용되지 않습니다. 이러한 사용자는 버킷 정책을 통해 S3 버킷 액세스 권한을 부여받았을 수 있습니다.

사용자 생성 마법사에 액세스합니다.

단계

1. 액세스 관리 > *사용자*를 선택합니다.

   테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우 파란색 배너가 표시되어 해당 테넌트의 소스 그리드임을 나타냅니다. 이 그리드에서 생성하는 모든 로컬 사용자는 연결된 다른 그리드로 복제됩니다.

2. *사용자 생성*을 선택합니다.

자격 증명을 입력합니다

단계

1. 사용자 자격 증명 입력 단계에서 다음 필드를 입력합니다.

필드	설명
성명	이 사용자의 전체 이름입니다. 예를 들어, 사람의 이름과 성 또는 애플리케이션 이름일 수 있습니다.
사용자 이름	이 사용자가 로그인할 때 사용할 이름입니다. 사용자 이름은 고유해야 하며 변경할 수 없습니다. 참고: 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 동일한 *사용자 이름*이 대상 그리드의 테넌트에 이미 존재하면 복제 오류가 발생합니다.
비밀번호 및 비밀번호 확인	사용자가 처음 로그인할 때 사용할 비밀번호입니다.
액세스 거부	이 사용자가 하나 이상의 그룹에 속해 있더라도 테넌트 계정에 로그인하지 못하도록 하려면 *예*를 선택하십시오. 예를 들어, 사용자의 로그인 기능을 일시적으로 중단하려면 *예*를 선택하세요.

2. *Continue*를 선택합니다.

그룹에 할당

단계

1. 사용자를 하나 이상의 로컬 그룹에 할당하여 수행할 수 있는 작업을 결정합니다.

   사용자를 그룹에 할당하는 것은 선택 사항입니다. 원하시는 경우, 그룹을 생성하거나 편집할 때 사용자를 선택할 수 있습니다.

   어떤 그룹에도 속하지 않은 사용자는 관리 권한이 없습니다. 권한은 누적됩니다. 사용자는 자신이 속한 모든 그룹에 대한 모든 권한을 갖게 됩니다. "테넌트 관리 권한"을 참조하십시오.

2. *사용자 생성*을 선택합니다.

   테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에 있는 경우, 새 로컬 사용자가 테넌트의 대상 그리드로 복제됩니다. 사용자 세부 정보 페이지의 개요 섹션에서 *복제 상태*가 *성공*으로 표시됩니다.

3. *완료*를 선택하여 사용자 페이지로 돌아갑니다.

로컬 사용자 보기 또는 편집

단계

1. 액세스 관리 > *사용자*를 선택합니다.
2. 사용자 페이지에 제공된 정보를 검토하십시오. 이 페이지에는 해당 테넌트 계정의 모든 로컬 및 연합 사용자에 대한 기본 정보가 나열되어 있습니다.

   테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에서 사용자를 보고 있는 경우:

   - 배너 메시지에는 사용자를 편집하거나 삭제해도 변경 사항이 다른 그리드에 동기화되지 않는다는 내용이 표시됩니다.
   - 필요에 따라, 대상 그리드의 테넌트로 사용자가 복제되지 않은 경우 배너 메시지가 표시됩니다. 실패한 사용자 복제 작업을 다시 시도합니다..

3. 사용자의 전체 이름을 변경하려면:
   a. 사용자의 확인란을 선택합니다.
   b. 작업 > *전체 이름 편집*을 선택합니다.
   c. 새 이름을 입력합니다.
   d. *변경 사항 저장*을 선택합니다.
4. 더 자세한 내용을 보거나 추가 수정을 하려면 다음 중 하나를 수행하십시오.
   - 사용자 이름을 선택합니다.
   - 해당 사용자의 확인란을 선택한 다음 작업 > *사용자 세부 정보 보기*를 선택합니다.
5. 개요 섹션을 검토하십시오. 이 섹션에는 각 사용자에 대한 다음 정보가 표시됩니다.
   - 성명
   - 사용자 이름
   - 사용자 유형

- 액세스 거부됨
- 액세스 모드
- 그룹 멤버십
- 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에서 사용자를 보고 있는 경우 추가 필드:
  - 클론 생성 상태 (성공 또는 실패)
  - 이 사용자의 정보를 수정해도 다른 그리드에 변경 사항이 동기화되지 않음을 나타내는 파란색 배너가 표시됩니다.

6. 필요에 따라 사용자 설정을 편집합니다. 입력할 내용에 대한 자세한 내용은 로컬 사용자 생성을 참조하십시오.
   a. 개요 섹션에서 이름 또는 편집 아이콘 을 선택하여 전체 이름을 변경하세요.

      사용자 이름은 변경할 수 없습니다.

   b. 암호 탭에서 사용자의 암호를 변경하고 *변경 사항 저장*을 선택합니다.
   c. 액세스 탭에서 사용자가 로그인할 수 있도록 허용하려면 *아니요*를 선택하고, 로그인을 차단하려면 *예*를 선택합니다. 그런 다음 *변경 사항 저장*을 선택합니다.
   d. 액세스 키 탭에서 *키 생성*을 선택하고 "다른 사용자의 S3 액세스 키 생성"에 대한 지침을 따르십시오.
   e. 그룹 탭에서 *그룹 편집*을 선택하여 사용자를 그룹에 추가하거나 그룹에서 제거합니다. 그런 다음 *변경 내용 저장*을 선택합니다.
7. 변경한 각 섹션에 대해 *변경 내용 저장*을 선택했는지 확인하십시오.

연합 사용자 가져오기

최대 100명까지 연합 사용자를 하나 이상 사용자 페이지로 직접 가져올 수 있습니다.

단계

1. 액세스 관리 > *사용자*를 선택합니다.
2. *연합 사용자 가져오기*를 선택합니다.
3. 연합 사용자 한 명 이상의 UUID 또는 사용자 이름을 입력하십시오.

   여러 항목을 입력하려면 각 UUID 또는 사용자 이름을 새 줄에 추가하십시오.

4. *가져오기*를 선택합니다.

   사용자 필드로의 가져오기가 한 명 이상의 사용자에 대해 실패하는 경우 다음 단계를 수행하십시오.

   a. *가져오지 않은 사용자*를 확장하고 *사용자 복사*를 선택합니다.
   b. 이전*을 선택하고 복사한 사용자를 *연합 사용자 가져오기 대화 상자에 붙여넣어 가져오기를 다시 시도하십시오.

   연합 사용자 가져오기 대화 상자를 닫으면 성공적으로 가져온 사용자에 대한 연합 사용자 정보가 사용자 페이지에 표시됩니다.

로컬 사용자 중복

로컬 사용자를 복제하면 새 사용자를 더 빠르게 생성할 수 있습니다.

> 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있고 테넌트의 소스 그리드에서 사용자를 복제하면 복제된 사용자가 테넌트의 대상 그리드에도 복제됩니다.

단계

1. 액세스 관리 > *사용자*를 선택합니다.
2. 복제하려는 사용자의 확인란을 선택합니다.
3. 작업 > *사용자 복제*를 선택합니다.
4. 입력할 내용에 대한 자세한 내용은 로컬 사용자 생성을 참조하십시오.
5. *사용자 생성*을 선택합니다.

사용자 클론 재시도

실패한 클론을 다시 시도하려면:

1. 사용자 이름 아래에 _(복제 실패)_라고 표시된 사용자를 각각 선택하십시오.
2. 작업 > *사용자 복제*를 선택합니다.
3. 복제 대상 사용자 각각의 상세 페이지에서 복제 작업의 상태를 확인할 수 있습니다.

자세한 내용은 "테넌트 그룹 및 사용자 복제"을 참조하십시오.

로컬 사용자 한 명 이상 삭제

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

> 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우 로컬 사용자를 삭제하면 StorageGRID는 다른 그리드에서 해당 사용자를 삭제하지 않습니다. 이 정보를 동기화해야 하는 경우 두 그리드에서 동일한 사용자를 모두 삭제해야 합니다.

> 페더레이션 사용자를 삭제하려면 페더레이션 ID 소스를 사용해야 합니다.

단계

1. 액세스 관리 > *사용자*를 선택합니다.
2. 삭제하려는 각 사용자의 확인란을 선택하십시오.
3. 작업 > 사용자 삭제 또는 작업 > *사용자 삭제*를 선택합니다.

   확인 대화 상자가 나타납니다.

4. 사용자 삭제 또는 *사용자 삭제*를 선택합니다.

S3 액세스 키 관리

StorageGRID에서 S3 액세스 키 관리

S3 테넌트 계정의 각 사용자는 StorageGRID 시스템에 객체를 저장하고 검색하기 위해 액세스 키를 보유해야 합니다. 액세스 키는 액세스 키 ID와 비밀 액세스 키로 구성됩니다.

S3 액세스 키는 다음과 같이 관리할 수 있습니다.

- **S3** 자격 증명 관리 권한이 있는 사용자는 자신의 S3 액세스 키를 생성하거나 삭제할 수 있습니다.
- 루트 액세스 권한이 있는 사용자는 S3 루트 계정 및 기타 모든 사용자의 액세스 키를 관리할 수 있습니다. 루트 액세스 키는 버킷 정책에서 명시적으로 비활성화하지 않는 한 테넌트의 모든 버킷 및 객체에 대한 전체 액세스 권한을 제공합니다.

StorageGRID는 서명 버전 2 및 서명 버전 4 인증을 지원합니다. 버킷 정책에서 명시적으로 허용하지 않는 한 계정 간 액세스는 허용되지 않습니다.

StorageGRID에서 자체 S3 액세스 키 생성

S3 테넌트를 사용 중이고 적절한 권한이 있는 경우, S3 액세스 키를 직접 생성할 수 있습니다. 버킷과 객체에 액세스하려면 액세스 키가 필요합니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "S3 자격 증명 또는 루트 액세스 권한을 직접 관리합니다."이(가) 있는 사용자 그룹에 속해 있습니다.

이 작업 정보

테넌트 계정에 대한 버킷을 생성하고 관리할 수 있는 S3 액세스 키를 하나 이상 생성할 수 있습니다. 새 액세스 키를 생성한 후에는 애플리케이션에 새 액세스 키 ID와 비밀 액세스 키를 업데이트하십시오. 보안을 위해 필요한 것보다 많은 키를 생성하지 말고 사용하지 않는 키는 삭제하십시오. 키가 하나만 있고 만료일이 임박한 경우, 만료 전에 새 키를 생성한 다음 기존 키를 삭제하십시오.

각 키에는 특정 만료 시간을 설정하거나 만료 시간을 설정하지 않을 수 있습니다. 만료 시간 설정 시 다음 지침을 따르십시오.

- 액세스 키에 만료 시간을 설정하여 특정 기간 동안만 액세스할 수 있도록 제한하세요. 만료 시간을 짧게 설정하면 액세스 키 ID와 비밀 액세스 키가 실수로 노출될 경우 위험을 줄일 수 있습니다. 만료된 키는 자동으로 삭제됩니다.
- 보안 위험이 낮아 주기적으로 새 키를 생성할 필요가 없다면 키 만료 시간을 설정할 필요가 없습니다. 나중에 새 키를 생성하기로 결정했다면 이전 키는 수동으로 삭제하세요.

> ⚠ 계정에 속한 S3 버킷과 객체는 테넌트 관리자에서 계정에 표시된 액세스 키 ID와 비밀 액세스 키를 사용하여 접근할 수 있습니다. 따라서 액세스 키는 비밀번호처럼 안전하게 보호해야 합니다. 액세스 키는 정기적으로 교체하고, 사용하지 않는 키는 계정에서 삭제하며, 다른 사용자와 절대 공유하지 마십시오.

단계

1. **STORAGE (S3)** > *My access keys*를 선택합니다.

   '내 액세스 키' 페이지가 나타나고 기존 액세스 키 목록이 표시됩니다.

2. *키 생성*을 선택합니다.

3. 다음 중 하나를 수행하십시오.
   - 만료되지 않는 키를 생성하려면 *만료 시간 설정 안 함*을 선택하세요. (기본값)
   - *만료 시간 설정*을 선택하고 만료 날짜와 시간을 설정합니다.

   > 만료일은 현재 날짜로부터 최대 5년까지 가능합니다. 만료 시간은 현재 시간으로부터 최소 1분부터 가능합니다.

4. *액세스 키 생성*을 선택합니다.

   액세스 키 다운로드 대화 상자가 나타나고, 액세스 키 ID와 비밀 액세스 키가 표시됩니다.

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

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

6. *완료*를 선택합니다.

   새 키는 내 액세스 키 페이지에 표시됩니다.

7. 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 선택적으로 테넌트 관리 API를 사용하여 소스 그리드의 테넌트에서 대상 그리드의 테넌트로 S3 액세스 키를 수동으로 복제할 수 있습니다. "API를 사용하여 S3 액세스 키 복제"을 참조하십시오.

StorageGRID에서 **S3** 액세스 키 보기

S3 테넌트를 사용 중이고 "적절한 권한"을(를) 보유하고 있는 경우, S3 액세스 키 목록을 볼 수 있습니다. 만료 시간을 기준으로 목록을 정렬하여 어떤 키가 곧 만료되는지 확인할 수 있습니다. 필요에 따라 더 이상 사용하지 않는 키를 "${post_edited_translations.segment}"하거나 "키 삭제"할 수 있습니다.

> 계정에 속한 S3 버킷과 객체는 테넌트 관리자에서 계정에 표시된 액세스 키 ID와 비밀 액세스 키를 사용하여 접근할 수 있습니다. 따라서 액세스 키는 비밀번호처럼 안전하게 보호해야 합니다. 액세스 키는 정기적으로 교체하고, 사용하지 않는 키는 계정에서 삭제하며, 다른 사용자와 절대 공유하지 마십시오.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 S3 자격 증명을 직접 관리할 수 있는 사용자 그룹에 속해 있습니다"권한".

단계

1. **STORAGE (S3)** > *My access keys*를 선택합니다.
2. '내 액세스 키' 페이지에서 기존 액세스 키를 만료 시간 또는 *액세스 키 ID*별로 정렬할 수 있습니다.
3. 필요에 따라 새 키를 생성하거나 더 이상 사용하지 않는 키를 삭제하세요.

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

만료된 키는 자동으로 삭제됩니다.

StorageGRID에서 자체 S3 액세스 키 삭제

S3 테넌트를 사용 중이고 적절한 권한이 있는 경우, 본인의 S3 액세스 키를 삭제할 수 있습니다. 액세스 키를 삭제하면 해당 키를 사용하여 테넌트 계정의 객체 및 버킷에 더 이상 액세스할 수 없습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 당신은 "S3 자격 증명 권한 직접 관리"을(를) 가지고 있습니다.

> 계정에 속한 S3 버킷과 객체는 테넌트 관리자에서 계정에 표시된 액세스 키 ID와 비밀 액세스 키를 사용하여 접근할 수 있습니다. 따라서 액세스 키는 비밀번호처럼 안전하게 보호해야 합니다. 액세스 키는 정기적으로 교체하고, 사용하지 않는 키는 계정에서 삭제하며, 다른 사용자와 절대 공유하지 마십시오.

단계

1. **STORAGE (S3)** > *My access keys*를 선택합니다.
2. '내 액세스 키' 페이지에서 제거하려는 각 액세스 키의 확인란을 선택합니다.
3. *삭제 키*를 선택합니다.
4. 확인 대화 상자에서 *Delete key*를 선택합니다.

   페이지 오른쪽 상단에 확인 메시지가 나타납니다.

StorageGRID에서 다른 사용자의 S3 액세스 키 생성

S3 테넌트를 사용 중이고 적절한 권한이 있는 경우, 버킷 및 객체에 액세스해야 하는 애플리케이션과 같은 다른 사용자를 위해 S3 액세스 키를 생성할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다.

이 작업 정보

다른 사용자가 자신의 테넌트 계정에 대한 버킷을 생성하고 관리할 수 있도록 하나 이상의 S3 액세스 키를 생성할 수 있습니다. 새 액세스 키를 생성한 후에는 애플리케이션에 새 액세스 키 ID와 비밀 액세스 키를 업데이트해야 합니다. 보안을 위해 사용자가 필요로 하는 것보다 많은 키를 생성하지 말고, 사용하지 않는 키는 삭제하십시오. 키가 하나만 있고 만료일이 임박한 경우, 만료 전에 새 키를 생성하고 기존 키를 삭제하십시오.

각 키에는 특정 만료 시간을 설정하거나 만료 시간을 설정하지 않을 수 있습니다. 만료 시간 설정 시 다음 지침을 따르십시오.

- 사용자의 액세스 권한을 특정 기간으로 제한하려면 키 만료 시간을 설정하세요. 만료 시간을 짧게 설정하면 액세스 키 ID와 비밀 액세스 키가 실수로 노출될 경우 발생할 수 있는 위험을 줄일 수 있습니다. 만료된 키는 자동으로 삭제됩니다.

- 보안 위험이 낮아 주기적으로 새 키를 생성할 필요가 없다면 키 만료 시간을 설정할 필요가 없습니다. 나중에 새 키를 생성하기로 결정했다면 이전 키는 수동으로 삭제하세요.

> 사용자가 소유한 S3 버킷과 객체는 테넌트 관리자에서 해당 사용자에게 표시되는 액세스 키 ID와 비밀 액세스 키를 사용하여 접근할 수 있습니다. 따라서 액세스 키는 비밀번호처럼 안전하게 보호해야 합니다. 액세스 키는 정기적으로 교체하고, 사용하지 않는 키는 계정에서 삭제하며, 다른 사용자와 절대 공유하지 마십시오.

단계

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

   사용자 상세 정보 페이지가 나타납니다.
3. *액세스 키*를 선택한 다음 *키 생성*을 선택합니다.
4. 다음 중 하나를 수행하십시오.
   - 만료되지 않는 키를 생성하려면 *만료 시간 설정 안 함*을 선택하세요. (기본값)
   - *만료 시간 설정*을 선택하고 만료 날짜와 시간을 설정합니다.

     > 만료일은 현재 날짜로부터 최대 5년까지 가능합니다. 만료 시간은 현재 시간으로부터 최소 1분부터 가능합니다.
5. *액세스 키 생성*을 선택합니다.

   액세스 키 다운로드 대화 상자가 나타나고, 액세스 키 ID와 비밀 액세스 키가 표시됩니다.
6. 액세스 키 ID와 비밀 액세스 키를 안전한 위치에 복사하거나, *.csv 다운로드*를 선택하여 액세스 키 ID와 비밀 액세스 키가 포함된 스프레드시트 파일을 저장합니다.

   > 이 정보를 복사하거나 다운로드하기 전까지는 이 대화 상자를 닫지 마십시오. 대화 상자를 닫은 후에는 키를 복사하거나 다운로드할 수 없습니다.
7. *완료*를 선택합니다.

   새 키는 사용자 세부 정보 페이지의 액세스 키 탭에 표시됩니다.
8. 테넌트 계정에 그리드 페더레이션 연결 사용 권한이 있는 경우, 선택적으로 테넌트 관리 API를 사용하여 소스 그리드의 테넌트에서 대상 그리드의 테넌트로 S3 액세스 키를 수동으로 복제할 수 있습니다. "API를 사용하여 S3 액세스 키 복제"을 참조하십시오.

StorageGRID에서 다른 사용자의 **S3** 액세스 키 보기

S3 테넌트를 사용 중이고 적절한 권한이 있는 경우, 다른 사용자의 S3 액세스 키를 볼 수 있습니다. 만료 시간순으로 목록을 정렬하여 곧 만료될 키를 확인할 수 있습니다. 필요에 따라 새 키를 생성하거나 더 이상 사용하지 않는 키를 삭제할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 다음이 있습니다. "루트 액세스 권한".

> 사용자가 소유한 S3 버킷과 객체는 테넌트 관리자에서 해당 사용자에게 표시되는 액세스 키 ID와 비밀 액세스 키를 사용하여 접근할 수 있습니다. 따라서 액세스 키는 비밀번호처럼 안전하게 보호해야 합니다. 액세스 키는 정기적으로 교체하고, 사용하지 않는 키는 계정에서 삭제하며, 다른 사용자와 절대 공유하지 마십시오.

단계

1. 액세스 관리 > *사용자*를 선택합니다.
2. 사용자 페이지에서 S3 액세스 키를 보려는 사용자를 선택합니다.
3. 사용자 세부 정보 페이지에서 *액세스 키*를 선택합니다.
4. 키를 만료 시간 또는 액세스 키 **ID** 순으로 정렬하세요.
5. 필요에 따라 새 키를 생성하고 더 이상 사용하지 않는 키는 수동으로 삭제하십시오.

   기존 키가 만료되기 전에 새 키를 생성하면 사용자는 계정의 개체에 대한 액세스 권한을 일시적으로 잃지 않고 새 키를 사용하여 작업을 시작할 수 있습니다.

   만료된 키는 자동으로 삭제됩니다.

관련 정보

- "다른 사용자의 S3 액세스 키 생성"
- "다른 사용자의 S3 액세스 키 삭제"

StorageGRID에서 다른 사용자의 **S3** 액세스 키 삭제

S3 테넌트를 사용 중이고 적절한 권한이 있는 경우, 다른 사용자의 S3 액세스 키를 삭제할 수 있습니다. 액세스 키가 삭제되면 해당 테넌트 계정의 객체 및 버킷에 더 이상 액세스할 수 없습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 다음이 있습니다. "루트 액세스 권한".

> 사용자가 소유한 S3 버킷과 객체는 테넌트 관리자에서 해당 사용자에게 표시되는 액세스 키 ID와 비밀 액세스 키를 사용하여 접근할 수 있습니다. 따라서 액세스 키는 비밀번호처럼 안전하게 보호해야 합니다. 액세스 키는 정기적으로 교체하고, 사용하지 않는 키는 계정에서 삭제하며, 다른 사용자와 절대 공유하지 마십시오.

단계

1. 액세스 관리 > *사용자*를 선택합니다.
2. 사용자 페이지에서 S3 액세스 키를 관리하려는 사용자를 선택합니다.
3. 사용자 세부 정보 페이지에서 *액세스 키*를 선택한 다음 삭제하려는 각 액세스 키의 확인란을 선택합니다.
4. 작업 > *선택한 키 삭제*를 선택합니다.

5. 확인 대화 상자에서 *Delete key*를 선택합니다.

   페이지 오른쪽 상단에 확인 메시지가 나타납니다.

S3 버킷 관리

StorageGRID S3 버킷을 생성합니다

테넌트 관리자를 사용하여 객체 데이터용 S3 버킷을 생성할 수 있습니다.

시작하기 전에

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

  버킷 또는 객체의 S3 Object Lock 속성을 설정하거나 수정할 수 있는 권한은 "버킷 정책 또는 그룹 정책"을(를) 통해 부여할 수 있습니다.

- 버킷에 S3 객체 잠금을 활성화하려는 경우, 그리드 관리자가 StorageGRID 시스템에 대한 전역 S3 객체 잠금 설정을 활성화했으며 S3 객체 잠금 버킷 및 객체에 대한 요구 사항을 검토한 상태여야 합니다.
- 각 테넌트가 5,000개의 버킷을 보유하게 될 경우, 그리드의 각 스토리지 노드에는 최소 64GB의 RAM이 필요합니다.

  각 그리드는 최대 100,000개의 버킷을 가질 수 있으며, 여기에는 "브랜치 버킷"이(가) 포함됩니다.

마법사에 액세스합니다

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지(S3) > *버킷*을 선택합니다.
2. *버킷 생성*을 선택합니다.

세부 정보를 입력합니다

단계

1. 버킷에 대한 세부 정보를 입력합니다.

필드	설명
버킷 이름	이 규칙을 준수하는 버킷의 이름: • 각 StorageGRID 시스템 전체에서 고유해야 합니다(테넌트 계정 내에서만 고유한 것이 아님). • DNS 규정을 준수해야 합니다. • 최소 3자, 최대 63자여야 합니다. • 각 라벨은 소문자 또는 숫자로 시작하고 끝나야 하며, 소문자, 숫자 및 하이픈만 사용할 수 있습니다. • 가상 호스팅 스타일 요청에는 마침표를 포함해서는 안 됩니다. 마침표는 서버 와일드카드 인증서 검증에 문제를 일으킬 수 있습니다. 자세한 내용은 "Amazon Web Services(AWS) 버킷 이름 지정 규칙에 대한 문서"를 참조하십시오. 참고: 버킷을 생성한 후에는 버킷 이름을 변경할 수 없습니다.
지역	버킷의 영역. StorageGRID 관리자가 사용 가능한 리전을 관리합니다. 버킷의 리전은 객체에 적용되는 데이터 보호 정책에 영향을 줄 수 있습니다. 기본적으로 모든 버킷은 `us-east-1` 리전에 생성됩니다. 기본 리전이 `us-east-1` 이외의 다른 리전으로 구성된 경우, 드롭다운에서 해당 다른 리전이 처음에 선택됩니다. 참고: 버킷 생성 후에는 지역을 변경할 수 없습니다.

2. *Continue*를 선택합니다.

설정 관리

단계

1. 선택적으로 버킷에 대한 객체 버전 관리를 활성화할 수 있습니다.

   이 버킷에 각 객체의 모든 버전을 저장하려면 객체 버전 관리를 활성화하십시오. 그러면 필요에 따라 객체의 이전 버전을 검색할 수 있습니다.

   다음과 같은 경우 객체 버전 관리를 활성화해야 합니다.

   ◦ 해당 버킷은 그리드 간 복제에 사용됩니다.

   ◦ 이 버킷에서 "브랜치 버킷"을(를) 생성하려고 합니다.

2. 전역 S3 Object Lock 설정이 활성화된 경우, 쓰기 1회-읽기 다수(WORM) 모델을 사용하여 객체를 저장하기 위해 버킷에 대해 S3 Object Lock을 선택적으로 활성화할 수 있습니다.

   특정 규제 요건을 충족하는 등 일정 기간 동안 객체를 보존해야 하는 경우에만 버킷에 S3 Object Lock을 활성화하십시오. S3 Object Lock은 객체가 일정 기간 또는 무기한으로 삭제되거나 덮어쓰이는 것을 방지하는 영구적인 설정입니다.

버킷에 대해 S3 Object Lock 설정을 활성화하면 비활성화할 수 없습니다. 올바른 권한을 가진 사용자는 해당 버킷에 변경할 수 없는 객체를 추가할 수 있습니다. 이러한 객체 또는 버킷 자체를 삭제하지 못할 수도 있습니다.

버킷에 S3 객체 잠금을 활성화하면 버킷 버전 관리가 자동으로 활성화됩니다.

3. *S3 객체 잠금 사용*을 선택한 경우, 선택적으로 이 버킷에 대해 *기본 보존*을 사용 설정할 수 있습니다.

그리드 관리자가 "S3 Object Lock의 특정 기능 사용"할 수 있는 권한을 부여해야 합니다.

기본 보존*이 활성화되면 버킷에 새로 추가된 객체는 삭제되거나 덮어쓰기되지 않도록 자동으로 보호됩니다. *기본 보존 설정은 자체 보존 기간이 있는 객체에는 적용되지 않습니다.

a. *기본 보존*이 활성화된 경우 버킷에 대한 *기본 보존 모드*를 지정하십시오.

기본 보존 모드	설명
거버넌스	• `s3:BypassGovernanceRetention` 권한이 있는 사용자는 `x-amz-bypass-governance-retention: true` 요청 헤더를 사용하여 보존 설정을 우회할 수 있습니다. • 이러한 사용자는 보존 기한이 도래하기 전에 개체 버전을 삭제할 수 있습니다. • 이러한 사용자는 개체의 보존 기간을 늘리거나 줄이거나 없앨 수 있습니다.
규정 준수	• 해당 객체는 지정된 보존 기간이 만료될 때까지 삭제할 수 없습니다. • 객체의 보존 기간은 늘릴 수는 있지만 줄일 수는 없습니다. • 해당 날짜에 도달할 때까지 객체의 보존 기한을 제거할 수 없습니다. 참고: 그리드 관리자가 규정 준수 모드 사용을 허용해야 합니다.

b. *기본 보존*이 활성화된 경우 버킷의 *기본 보존 기간*을 지정하십시오.

*기본 보존 기간*은 이 버킷에 새로 추가된 객체를 수집 시점부터 보존할 기간을 나타냅니다. 그리드 관리자가 설정한 테넌트의 최대 보존 기간 이하의 값을 지정하십시오.

최대 보존 기간은 1일에서 100년까지의 값으로 설정할 수 있으며, 그리드 관리자가 테넌트를 생성할 때 설정됩니다. 기본 보존 기간을 설정할 때는 최대 보존 기간에 설정된 값을 초과할 수 없습니다. 필요한 경우 그리드 관리자에게 최대 보존 기간을 늘리거나 줄이도록 요청하십시오.

4. 선택적으로 *용량 제한 활성화*를 선택하고 값을 입력한 다음 용량 단위를 선택합니다.

용량 제한은 해당 버킷의 객체에 사용할 수 있는 최대 용량입니다. 이 값은 물리적 용량(디스크 용량)이 아닌 논리적 용량(객체 크기)을 나타냅니다.

제한이 설정되지 않은 경우 이 버킷의 용량은 무제한입니다. 자세한 내용은 "용량 제한 사용량"을 참조하십시오.

5. 선택적으로 *객체 개수 제한 활성화*를 선택합니다.

객체 개수 제한은 이 버킷에 포함될 수 있는 최대 객체 수입니다. 이 값은 논리적 양(객체 개수)을 나타냅니다. 제한이 설정되지 않은 경우 객체 개수는 무제한입니다.

6. *버킷 생성*을 선택합니다.

   버킷이 생성되어 버킷 페이지의 표에 추가됩니다.

7. 필요에 따라 *버킷 세부 정보 페이지로 이동*을 선택하여 "버킷 세부 정보 보기" 추가 구성을 수행합니다.

또한 필요에 따라 "브랜치 버킷 생성"할 수 있습니다.

StorageGRID에서 **S3** 버킷 세부 정보 보기

버킷은 테넌트 계정에서 확인할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스, 모든 버킷 관리 또는 모든 버킷 보기 권한"을(를) 가진 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.

   버킷 페이지가 나타납니다.

2. 각 버킷의 요약 테이블을 검토하십시오.

   필요에 따라 원하는 열을 기준으로 정보를 정렬하거나 목록을 앞뒤로 넘겨볼 수 있습니다.

   > 표시되는 객체 개수, 사용 공간 및 사용량 값은 추정치입니다. 이러한 추정치는 데이터 수집 시간, 네트워크 연결 상태 및 노드 상태의 영향을 받습니다. 버킷에 버전 관리가 활성화된 경우 삭제된 객체 버전도 객체 개수에 포함됩니다.

   이름
   : 버킷의 고유 이름이며 변경할 수 없습니다.

   활성화된 기능
   : 버킷에 대해 활성화된 기능 목록입니다.

   S3 Object Lock
   : 버킷에 대해 S3 객체 잠금이 활성화되어 있는지 여부입니다.

     이 열은 그리드에 S3 Object Lock이 활성화된 경우에만 표시됩니다. 또한 이 열에는 기존 규정 준수 버킷에 대한 정보도 표시됩니다.

   지역
   : 버킷의 리전은 변경할 수 없습니다. 이 열은 기본적으로 숨겨져 있습니다.

객체 개수

이 버킷에 있는 객체의 개수입니다. 버킷에 버전 관리가 활성화된 경우, 현재 버전이 아닌 객체 버전도 이 값에 포함됩니다.

객체가 추가되거나 삭제될 때 이 값이 즉시 업데이트되지 않을 수 있습니다.

사용된 공간

버킷에 있는 모든 객체의 논리적 크기입니다. 논리적 크기에는 복제본이나 이레이저 코딩된 복사본 또는 객체 메타데이터에 필요한 실제 공간이 포함되지 않습니다.

이 값이 업데이트되는 데 최대 10분이 소요될 수 있습니다.

사용법

버킷 용량 제한이 설정된 경우, 해당 용량 제한에서 사용된 비율입니다.

사용량 값은 내부 추정치를 기반으로 하며 경우에 따라 초과될 수 있습니다. 예를 들어, StorageGRID는 테넌트가 객체 업로드를 시작할 때 용량 제한(설정된 경우)을 확인하고, 테넌트가 용량 제한을 초과한 경우 해당 버킷으로의 새로운 객체 업로드를 거부합니다. 하지만 StorageGRID는 용량 제한 초과 여부를 판단할 때 현재 업로드 중인 객체의 크기는 고려하지 않습니다. 객체가 삭제되면 테넌트는 용량 제한 사용량이 다시 계산될 때까지 해당 버킷에 새로운 객체를 업로드할 수 없게 될 수 있습니다. 계산에는 10분 이상 소요될 수 있습니다.

이 값은 객체와 해당 메타데이터를 저장하는 데 필요한 물리적 크기가 아닌 논리적 크기를 나타냅니다.

용량

설정된 경우, 버킷의 용량 제한입니다.

생성일

버킷이 생성된 날짜 및 시간입니다. 이 열은 기본적으로 숨겨져 있습니다.

3. 특정 버킷의 세부 정보를 보려면 표에서 버킷 이름을 선택하십시오.
   a. 버킷에 대한 자세한 정보(예: 지역 및 객체 수)를 확인하려면 웹 페이지 상단의 요약 정보를 참조하십시오.
   b. 용량 제한 사용량 및 객체 개수 제한 사용량 막대를 확인하십시오. 사용량이 100%이거나 100%에 가까운 경우 제한을 늘리거나 일부 객체를 삭제하는 것을 고려하십시오.
   c. 필요에 따라 버킷의 객체 삭제 및 *버킷 삭제*를 선택하십시오.

      각 옵션을 선택할 때 나타나는 주의 사항을 주의 깊게 살펴보십시오. 자세한 내용은 다음을 참조하십시오.

      - "버킷의 모든 객체를 삭제합니다"
      - "버킷 삭제" (버킷은 비어 있어야 합니다)

   d. 필요에 따라 각 탭에서 버킷 설정을 확인하거나 변경할 수 있습니다.
      - **S3** 콘솔: 버킷의 객체를 볼 수 있습니다. 자세한 내용은 "S3 콘솔 사용"을 참조하십시오.
      - 버킷 옵션: 옵션 설정을 보거나 변경할 수 있습니다. S3 Object Lock과 같은 일부 설정은 버킷 생성 후에는 변경할 수 없습니다.
        - "버킷 일관성 관리"

        - "최근 액세스 시간 업데이트"
        - "용량 제한"
        - "객체 개수 제한"
        - "객체 버전 관리"
        - "S3 Object Lock"
        - "기본 버킷 보존 기간"
        - "그리드 간 복제 관리" (테넌트에 허용된 경우)
    - 플랫폼 서비스: "플랫폼 서비스 관리"(테넌트에 허용된 경우)
    - 버킷 접근: 옵션 설정을 보거나 변경할 수 있습니다. 특정 접근 권한이 필요합니다.
        - "버킷 및 객체에 대한 CORS"을 구성하여 버킷과 버킷 내의 객체가 다른 도메인의 웹 애플리케이션에서 액세스할 수 있도록 합니다.
        - "사용자 접근 제어" S3 버킷 및 해당 버킷에 있는 객체에 대한 정보입니다.
    - **Branches**: 해당 버킷에 대한 분기 버킷 목록을 봅니다. "새 브랜치 버킷을 생성하거나 브랜치 버킷을 관리합니다.".

StorageGRID 브랜치 버킷에 대해 알아보세요

브랜치 버킷은 특정 시점의 버킷 내 객체에 대한 접근을 제공합니다.

기존 버킷에서 브랜치 버킷을 생성할 수 있습니다. 브랜치 버킷을 생성한 후에는 해당 버킷이 생성된 원래 버킷을 _기본 버킷_이라고 합니다. 또한 다른 브랜치 버킷에서 브랜치 버킷을 생성할 수도 있습니다.

브랜치 버킷은 보호된 데이터에 대한 접근을 제공하지만 백업 역할은 하지 않습니다. 데이터를 계속 보호하려면 기본 버킷에서 다음 기능을 사용하십시오.

- "S3 Object Lock"
- "교차 그리드 복제" 기본 버킷용
- "버킷 정책" 버전 관리 버킷에서 오래된 객체 버전을 정리하려면

브랜치 버킷의 다음 특성에 유의하십시오.

- "S3 콘솔에서 객체 다운로드"을 사용하여 브랜치 버킷의 객체에 액세스할 수 있습니다.
- 클라이언트가 브랜치 버킷의 객체에 접근할 때, 기본 버킷의 정책이 아닌 브랜치 버킷의 "${post_edited_translations.segment}"에 따라 접근 허용 여부가 결정됩니다.
- 기본 버킷에서 생성된 객체는 "ILM 규칙"이(가) 기본 버킷에 적용되는 방식을 기반으로 평가됩니다. 브랜치 버킷에서 생성된 객체는 ILM 규칙이 브랜치 버킷에 적용되는 방식을 기반으로 평가됩니다.
- ${post_edited_translations.segment}
- 브랜치 버킷에서는 플랫폼 서비스가 지원되지 않습니다.

${post_edited_translations.segment}

- 손상된 객체를 제거하려면 손상 발생 이전 시점을 기준으로 브랜치 버킷을 생성한 다음, 애플리케이션이 손상된 객체가 포함된 기본 버킷 대신 브랜치 버킷을 가리키도록 설정하면 됩니다.

- 버전 관리 버킷에 데이터를 저장하고 있습니다. 예기치 않은 취약점으로 인해 시간 *T* 이후에 원치 않는 많은 오브젝트가 수집되었습니다. Before 시간 값인 _T_에 대한 브랜치 버킷을 생성하고 클라이언트 작업을 해당 브랜치 버킷으로 재지정할 수 있습니다. 그러면 Before 시간 _T_보다 이전에 수집된 오브젝트만 클라이언트에 노출됩니다.

${post_edited_translations.segment}

- 브랜치 버킷에 대한 PUT 객체 작업은 해당 브랜치에 객체를 생성합니다.
- 브랜치 버킷에 대한 GET 객체 작업은 브랜치에서 객체를 가져옵니다. 객체가 브랜치 버킷에 존재하지 않는 경우, 베이스 버킷에서 객체를 가져옵니다.
- 브랜치 버킷에서 객체가 삭제되는 과정은 다음과 같습니다.

작업	대상	결과	기본 버킷에서의 객체 가시성	브랜치 버킷에서의 객체 가시성
버전 ID 없이 삭제	베이스 버킷	삭제 표시는 기본 버킷에만 생성됩니다.	HEAD/GET 시 객체가 존재하지 않음을 반환하지만 특정 버전에는 여전히 액세스할 수 있습니다	HEAD/GET 반환 시 객체가 존재하며 특정 버전에 계속 액세스할 수 있습니다. 삭제 표시는 브랜치 버킷의 `beforeTime` 이후에 생성되었을 것입니다.
버전 ID로 삭제	베이스 버킷	특정 객체 버전이 기본 버킷과 분기 버킷 모두에서 삭제됩니다.	HEAD/GET이 객체 버전이 존재하지 않음을 반환합니다	HEAD/GET이 객체 버전이 존재하지 않음을 반환합니다
버전 ID 없이 삭제	브랜치 버킷	삭제 표시는 브랜치 버킷에만 생성됩니다.	HEAD/GET 요청은 객체를 반환합니다(기본 버킷 객체는 영향을 받지 않음)	HEAD/GET가 개체가 존재하지 않음을 반환합니다
버전 ID로 삭제	브랜치 버킷	${post_edited_translations.segment}	${post_edited_translations.segment}	HEAD/GET이 객체 버전이 존재하지 않음을 반환합니다

또한 "S3 버전 관리 객체가 삭제되는 방법"을(를) 참조하십시오.

StorageGRID 브랜치 버킷 관리

테넌트 관리자를 사용하여 브랜치 버킷을 생성하고 세부 정보를 볼 수 있습니다.

시작하기 전에

- "지원되는 웹 브라우저"을 사용하여 테넌트 관리자에 로그인했습니다.
- 루트 액세스 또는 "모든 버킷 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.
- 브랜치를 생성하려는 기본 버킷에는 "버전 관리가 활성화됨"이(가) 있습니다.
- 기본 버킷의 소유자입니다.

이 작업 정보

브랜치 버킷에 대한 다음 정보를 참고하십시오.

- 버킷 또는 객체의 S3 Object Lock 속성을 설정하는 권한은 "버킷 정책 또는 그룹 정책"을(를) 통해 부여할 수 있습니다.
- 기본 버킷의 버전 관리를 일시 중지하면 기본 버킷의 내용이 해당 브랜치 버킷에서 더 이상 표시되지 않습니다.

브랜치 버킷을 구성하고 생성한 후에는 구성을 변경할 수 없습니다.

브랜치 버킷 생성

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
2. 브랜치를 생성할 버킷("기본 버킷")을 선택합니다.
3. 버킷 세부 정보 페이지에서 **Branches** > *Create branch bucket*을 선택합니다.

   기본 버킷에 버전 관리가 활성화되어 있지 않으면 브랜치 버킷 생성 버튼이 비활성화됩니다.

세부 정보를 입력합니다

단계

1. 브랜치 버킷에 대한 세부 정보를 입력하세요.

필드	설명
브랜치 버킷 이름	다음 규칙을 준수하는 브랜치 버킷의 이름: • 각 StorageGRID 시스템 전체에서 고유해야 합니다(테넌트 계정 내에서만 고유한 것이 아님). • DNS 규정을 준수해야 합니다. • 최소 3자, 최대 63자여야 합니다. • 각 라벨은 소문자 또는 숫자로 시작하고 끝나야 하며, 소문자, 숫자 및 하이픈만 사용할 수 있습니다. • 가상 호스팅 스타일 요청에는 마침표를 포함해서는 안 됩니다. 마침표는 서버 와일드카드 인증서 검증에 문제를 일으킬 수 있습니다. 자세한 내용은 "Amazon Web Services(AWS) 버킷 이름 지정 규칙에 대한 문서"를 참조하십시오. 참고: 브랜치 버킷을 생성한 후에는 이름을 변경할 수 없습니다.
지역 (브랜치 버킷의 경우 수정할 수 없음)	브랜치 버킷의 리전. 브랜치 버킷의 지역은 베이스 버킷의 지역과 일치해야 하므로, 브랜치 버킷의 경우 이 필드는 비활성화되어 있습니다.

필드	설명
시간 이전	기본 버킷에서 생성된 객체 버전이 브랜치 버킷에서 액세스 가능해지는 마감 시간입니다. 브랜치 버킷은 Before 시간보다 이전에 생성된 객체 버전에 대한 액세스를 제공합니다. '이전' 시간은 반드시 이미 지난 날짜와 시간이어야 합니다. 미래의 날짜는 될 수 없습니다.
브랜치 버킷 유형	• **Read-write**: 브랜치 버킷에서 객체 또는 객체 버전을 추가하거나 삭제할 수 있습니다. • 읽기 전용: 브랜치 버킷의 객체는 수정할 수 없습니다. 참고: 브랜치 버킷이 비어 있는 경우에만 브랜치 버킷 유형을 읽기 전용으로 설정할 수 있습니다. 기존 브랜치 버킷의 유형이 읽기/쓰기로 설정되어 있고 아직 데이터를 쓰지 않은 경우, 유형을 읽기 전용으로 변경할 수 있습니다.

2. *Continue*를 선택합니다.

객체 설정 관리(선택 사항)

브랜치 버킷의 객체 설정은 기본 버킷의 객체 버전에 영향을 미치지 않습니다.

단계

1. 전역 S3 객체 잠금 설정이 활성화된 경우, 선택적으로 브랜치 버킷에 대해 S3 객체 잠금을 활성화할 수 있습니다. S3 객체 잠금을 활성화하려면 브랜치 버킷은 읽기/쓰기 가능한 버킷이어야 합니다.

   특정 규제 요건을 충족하는 등 일정 기간 동안 객체를 보존해야 하는 경우에만 브랜치 버킷에 S3 Object Lock을 활성화하십시오. S3 Object Lock은 객체가 일정 기간 또는 무기한으로 삭제되거나 덮어쓰이는 것을 방지하는 영구 설정입니다.

   > 버킷에 S3 객체 잠금 설정이 활성화되면 비활성화할 수 없습니다. 올바른 권한을 가진 사용자는 브랜치 버킷에 객체를 추가할 수 있지만, 추가된 객체는 변경할 수 없습니다. 이러한 객체 또는 브랜치 버킷 자체를 삭제하지 못할 수도 있습니다.

2. *S3 객체 잠금 활성화*를 선택한 경우, 선택적으로 브랜치 버킷에 대해 *기본 보존*을 활성화할 수 있습니다.

   > 그리드 관리자가 "S3 Object Lock의 특정 기능 사용"할 수 있는 권한을 부여해야 합니다.

   기본 보존*이 활성화되면 브랜치 버킷에 새로 추가된 객체는 삭제되거나 덮어쓰기되지 않도록 자동으로 보호됩니다. *기본 보존 설정은 자체 보존 기간이 있는 객체에는 적용되지 않습니다.

   a. *기본 보존*이 활성화된 경우, 브랜치 버킷에 대한 *기본 보존 모드*를 지정하십시오.

기본 보존 모드	설명
거버넌스	• `s3:BypassGovernanceRetention` 권한이 있는 사용자는 `x-amz-bypass-governance-retention: true` 요청 헤더를 사용하여 보존 설정을 우회할 수 있습니다. • 이러한 사용자는 보존 기한이 도래하기 전에 개체 버전을 삭제할 수 있습니다. • 이러한 사용자는 개체의 보존 기간을 늘리거나 줄이거나 없앨 수 있습니다.
규정 준수	• 해당 객체는 지정된 보존 기간이 만료될 때까지 삭제할 수 없습니다. • 객체의 보존 기간은 늘릴 수는 있지만 줄일 수는 없습니다. • 해당 날짜에 도달할 때까지 객체의 보존 기한을 제거할 수 없습니다. 참고: 그리드 관리자가 규정 준수 모드 사용을 허용해야 합니다.

b. *기본 보존*이 활성화된 경우, 브랜치 버킷에 대한 *기본 보존 기간*을 지정하십시오.

*기본 보존 기간*은 브랜치 버킷에 새로 추가된 객체를 수집 시점부터 보존할 기간을 나타냅니다. 그리드 관리자가 설정한 테넌트의 최대 보존 기간 이하의 값을 지정하십시오.

최대 보존 기간은 1일에서 100년까지의 값으로 설정할 수 있으며, 그리드 관리자가 테넌트를 생성할 때 설정됩니다. 기본 보존 기간을 설정할 때는 최대 보존 기간에 설정된 값을 초과할 수 없습니다. 필요한 경우 그리드 관리자에게 최대 보존 기간을 늘리거나 줄이도록 요청하십시오.

3. 필요에 따라 *용량 제한 활성화*를 선택합니다.

용량 제한은 브랜치 버킷에 사용할 수 있는 최대 용량입니다. 이 값은 물리적 용량(디스크 용량)이 아닌 논리적 용량(객체 크기)을 나타냅니다.

제한을 설정하지 않으면 브랜치 버킷의 용량은 무제한입니다. 자세한 내용은 "용량 제한 사용량"을 참조하십시오.

> ⓘ 이 설정은 브랜치 버킷에 직접 수집된 객체에만 적용되며, 기본 버킷에서 브랜치 버킷을 통해 볼 수 있는 객체에는 적용되지 않습니다.

4. 선택적으로 *객체 개수 제한 활성화*를 선택합니다.

객체 개수 제한은 브랜치 버킷에 포함될 수 있는 최대 객체 수입니다. 이 값은 논리적 양(객체 개수)을 나타냅니다. 제한이 설정되지 않은 경우 객체 개수는 무제한입니다.

> ⓘ 이 설정은 브랜치 버킷에 직접 수집된 객체에만 적용되며, 기본 버킷에서 브랜치 버킷을 통해 볼 수 있는 객체에는 적용되지 않습니다.

5. *버킷 생성*을 선택합니다.

브랜치 버킷이 생성되어 버킷 페이지의 표에 추가됩니다.

6. 필요에 따라 *버킷 세부 정보 페이지로 이동*을 선택하여 "브랜치 버킷 세부 정보 보기" 추가 구성을 수행합니다.

버킷 세부 정보 페이지에서 읽기 전용 버킷의 경우 객체 수정과 관련된 일부 구성 옵션이 비활성화됩니다.

StorageGRID 버킷에 ILM 정책 태그를 적용합니다

객체 스토리지 요구 사항에 따라 버킷에 적용할 ILM 정책 태그를 선택하십시오.

ILM 정책은 객체 데이터가 저장되는 위치와 특정 기간 후 삭제 여부를 제어합니다. 그리드 관리자는 ILM 정책을 생성하고 여러 활성 정책을 사용하는 경우 ILM 정책 태그에 할당합니다.

> 버킷의 정책 태그를 자주 변경하지 마십시오. 그렇지 않으면 성능 문제가 발생할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스, 모든 버킷 관리 또는 모든 버킷 보기 권한"을(를) 가진 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지(S3) > *버킷*을 선택합니다.

   버킷 페이지가 나타납니다. 필요에 따라 원하는 열을 기준으로 정보를 정렬하거나 목록을 앞뒤로 넘겨볼 수 있습니다.

2. ILM 정책 태그를 할당할 버킷 이름을 선택하십시오.

   이미 태그가 할당된 버킷에 대해 ILM 정책 태그 할당을 변경할 수도 있습니다.

   > 표시되는 객체 개수 및 사용 공간 값은 추정치입니다. 이러한 추정치는 데이터 수집 시간, 네트워크 연결 상태 및 노드 상태의 영향을 받습니다. 버킷에 버전 관리가 활성화된 경우 삭제된 객체 버전도 객체 개수에 포함됩니다.

3. 버킷 옵션 탭에서 ILM 정책 태그 아코디언을 확장합니다. 이 아코디언은 그리드 관리자가 사용자 지정 정책 태그 사용을 활성화한 경우에만 나타납니다.
4. 각 정책 태그의 설명을 읽고 버킷에 적용할 태그를 결정하십시오.

   > 버킷의 ILM 정책 태그를 변경하면 버킷 내 모든 객체에 대한 ILM 재평가가 수행됩니다. 새 정책에서 객체를 일정 기간 동안 보존하는 경우, 이전 객체는 삭제됩니다.

5. 버킷에 할당할 태그에 해당하는 라디오 버튼을 선택하십시오.
6. *변경 사항 저장*을 선택합니다. 키 `NTAP-SG-ILM-BUCKET-TAG`와 ILM 정책 태그 이름의 값을 가진 새 S3 버킷 태그가 버킷에 설정됩니다.

   > S3 애플리케이션이 실수로 새 버킷 태그를 덮어쓰거나 삭제하지 않도록 하십시오. 버킷에 새 TagSet을 적용할 때 이 태그를 누락하면 버킷의 객체는 기본 ILM 정책에 따라 평가됩니다.

ILM 정책 태그는 테넌트 관리자 또는 테넌트 관리자 API를 통해서만 설정하고 수정해야 하며, 이때 ILM 정책 태그의 유효성을 검사해야 합니다. `NTAP-SG-ILM-BUCKET-TAG`S3 PutBucketTagging API 또는 S3 DeleteBucketTagging API를 사용하여 ILM 정책 태그를 수정하지 마십시오.

버킷에 할당된 정책 태그를 변경하면 새 ILM 정책을 사용하여 객체를 재평가하는 동안 일시적인 성능 영향이 발생합니다.

StorageGRID 버킷 정책 관리

S3 버킷 및 해당 버킷에 있는 객체에 대한 사용자 액세스를 제어할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다. 모든 버킷 보기 및 모든 버킷 관리 권한은 보기만 허용합니다.
- 필요한 스토리지 노드 및 사이트 수가 확보되었는지 확인했습니다. 사이트 내에서 두 개 이상의 스토리지 노드를 사용할 수 없거나 사이트를 사용할 수 없는 경우, 이러한 설정을 변경하지 못할 수 있습니다.

단계

1. *버킷*을 선택한 다음 관리할 버킷을 선택합니다.
2. ${post_edited_translations.segment}
3. 다음 중 하나를 수행하십시오.
   - 정책 활성화 확인란을 선택하여 버킷 정책을 입력합니다. 그런 다음 유효한 JSON 형식의 문자열을 입력합니다.

     각 버킷 정책에는 20,480바이트의 크기 제한이 있습니다.

   - 기존 정책을 수정하려면 문자열을 편집하세요.
   - *정책 활성화*를 선택 해제하여 정책을 비활성화합니다.

버킷 정책에 대한 자세한 내용(언어 구문 및 예제 포함)은 "예시 버킷 정책"을 참조하십시오.

StorageGRID 버킷 일관성 관리

일관성 값은 버킷 설정 변경의 가용성을 지정하는 데 사용될 수 있으며, 버킷 내 객체의 가용성과 여러 스토리지 노드 및 사이트 간 객체의 일관성 간의 균형을 유지하는 데에도 사용될 수 있습니다. 클라이언트 애플리케이션의 운영 요구 사항에 맞춰 일관성 값을 기본값과 다르게 변경할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

버킷 일관성 지침

버킷 일관성은 해당 S3 버킷 내의 객체에 영향을 미치는 클라이언트 애플리케이션의 일관성을 결정하는 데 사용됩니다. 일반적으로 버킷에는 새 쓰기 후 읽기 일관성을 사용하는 것이 좋습니다.

버킷 일관성 변경

새 쓰기 후 읽기 일관성이 클라이언트 애플리케이션의 요구 사항을 충족하지 못하는 경우, 버킷 일관성을 설정하거나 `Consistency-Control` 헤더를 사용하여 일관성을 변경할 수 있습니다. `Consistency-Control` 헤더를 사용하면 버킷 일관성 설정이 재정의됩니다.

버킷의 일관성을 변경하면 변경 이후에 수집된 객체만 수정된 설정을 충족하는 것으로 보장됩니다.

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
2. 표에서 버킷 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

3. 버킷 옵션 탭에서 ** 아코디언을 선택합니다.
4. ${post_edited_translations.segment}
   - 모두: 최고 수준의 일관성을 제공합니다. 모든 노드가 데이터를 즉시 수신하며, 그렇지 않을 경우 요청이 실패합니다.
   - **Strong-global**: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
   - **Strong-site**: 사이트 내 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
   - **Read-after-new-write** (기본값): 새 객체에 대한 read-after-write 일관성 및 객체 업데이트에 대한 최종 일관성을 제공합니다. 높은 가용성 및 데이터 보호 보장을 제공합니다. 대부분의 경우에 권장됩니다.
   - 사용 가능: 새 객체 생성 및 객체 업데이트 모두에 대해 최종 일관성을 제공합니다. S3 버킷의 경우, 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷 또는 존재하지 않는 키에 대한 HEAD 또는 GET 작업). S3 FabricPool 버킷에서는 지원되지 않습니다.
5. *변경 사항 저장*을 선택합니다.

버킷 설정을 변경하면 어떻게 되나요**?**

버킷에는 버킷 및 버킷 내 객체의 동작에 영향을 미치는 여러 설정이 있습니다.

다음 버킷 설정은 기본적으로 강력한 일관성을 사용합니다. 사이트 내에서 두 개 이상의 스토리지 노드를 사용할 수 없거나 사이트를 사용할 수 없는 경우 이러한 설정을 변경해도 적용되지 않을 수 있습니다.

- "${post_edited_translations.segment}"
- "최종 액세스 시간"
- "버킷 수명 주기"
- "버킷 정책"
- "버킷 태깅"
- "버킷 버전 관리"

- "S3 Object Lock"
- "버킷 암호화"

> 버킷 버전 관리, S3 Object Lock 및 버킷 암호화의 일관성 값은 강력한 일관성을 보장하지 않는 값으로 설정할 수 없습니다.

다음 버킷 설정은 강력한 일관성을 사용하지 않으며 변경 사항에 대한 가용성이 더 높습니다. 이러한 설정의 변경 사항이 적용되는 데는 어느 정도 시간이 걸릴 수 있습니다.

- "플랫폼 서비스 구성: 알림, 복제 또는 검색 통합"
- "버킷 및 객체에 대한 StorageGRID CORS 구성"
- 버킷 일관성 변경

> 버킷 설정을 변경할 때 사용되는 기본 일관성이 클라이언트 애플리케이션의 요구사항을 충족하지 않는 경우, "S3 REST API"에 대한 `Consistency-Control` 헤더를 사용하거나 "테넌트 관리 API"의 `reducedConsistency` 또는 `force` 옵션을 사용하여 일관성을 변경할 수 있습니다.

StorageGRID에서 마지막 액세스 시간 업데이트 활성화 또는 비활성화

그리드 관리자가 StorageGRID 시스템의 정보 라이프사이클 관리(ILM) 규칙을 생성할 때, 객체의 마지막 액세스 시간을 기준으로 해당 객체를 다른 스토리지 위치로 이동할지 여부를 결정하도록 선택적으로 지정할 수 있습니다. S3 테넌트를 사용하는 경우, S3 버킷에 있는 객체의 마지막 액세스 시간 업데이트를 활성화하여 이러한 규칙을 활용할 수 있습니다.

이 지침은 고급 필터 또는 참조 시간으로 마지막 액세스 시간 옵션을 사용하는 ILM 규칙이 하나 이상 포함된 StorageGRID 시스템에만 적용됩니다. StorageGRID 시스템에 이러한 규칙이 없는 경우 이 지침을 무시해도 됩니다. 자세한 내용은 "ILM 규칙에서 마지막 액세스 시간 사용"을 참조하십시오.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업 정보

마지막 액세스 시간*은 **ILM** 규칙의 *참조 시간 배치 지침에 사용할 수 있는 옵션 중 하나입니다. 규칙의 참조 시간을 마지막 액세스 시간으로 설정하면 그리드 관리자는 객체가 마지막으로 검색(읽기 또는 보기)된 시간을 기준으로 특정 스토리지 위치에 객체를 배치하도록 지정할 수 있습니다.

예를 들어, 최근에 본 객체가 더 빠른 스토리지에 유지되도록 하려면 그리드 관리자가 다음과 같이 지정하는 ILM 규칙을 만들 수 있습니다.

- 지난 한 달 동안 검색된 객체는 로컬 스토리지 노드에 유지되어야 합니다.
- 지난 한 달 동안 검색되지 않은 오브젝트는 오프사이트 위치로 이동해야 합니다.

기본적으로 마지막 액세스 시간 업데이트는 비활성화되어 있습니다. StorageGRID 시스템에 마지막 액세스 시간 옵션을 사용하는 ILM 규칙이 포함되어 있고 이 옵션을 해당 버킷의 객체에 적용하려면 해당 규칙에 지정된 S3 버킷에 대해 마지막 액세스 시간 업데이트를 활성화해야 합니다.

객체를 검색할 때 마지막 액세스 시간을 업데이트하면 특히 작은 객체의 경우 StorageGRID 성능이 저하될 수 있습니다.

마지막 액세스 시간 업데이트로 인해 성능 영향이 발생합니다. StorageGRID가 객체를 검색할 때마다 이러한 추가 단계를 수행해야 하기 때문입니다.

- 새 타임스탬프로 객체를 업데이트합니다
- 해당 객체를 ILM 큐에 추가하여 현재 ILM 규칙 및 정책에 따라 재평가될 수 있도록 합니다.

이 표는 마지막 액세스 시간 기능이 비활성화 또는 활성화되었을 때 버킷 내 모든 객체에 적용되는 동작을 요약합니다.

요청 유형	마지막 액세스 시간이 비활성화된 경우의 동작(기본값)		마지막 액세스 시간이 활성화된 경우의 동작	
	마지막 액세스 시간 업데이트 여부?	해당 객체가 ILM 평가 대기열에 추가되었습니까?	마지막 액세스 시간 업데이트 여부?	해당 객체가 ILM 평가 대기열에 추가되었습니까?
HEAD 작업이 실행될 때 객체의 메타데이터를 검색하기 위한 요청입니다.	아니요	아니요	아니요	아니요
객체, 해당 객체의 액세스 제어 목록 또는 메타데이터를 검색하는 요청	아니요	아니요	예	예
객체의 메타데이터 업데이트 요청	예	예	예	예
객체 또는 객체 버전 목록 요청	아니요	아니요	아니요	아니요
한 버킷에서 다른 버킷으로 객체를 복사하는 요청입니다.	• 아니요, 원본 복사본에 대해서입니다. • 예, 대상 복사본에 대해서입니다.	• 아니요, 원본 복사본에 대해서입니다. • 예, 대상 복사본에 대해서입니다.	• 예, 소스 복사본의 경우 • 예, 대상 복사본에 대해서입니다.	• 예, 소스 복사본의 경우 • 예, 대상 복사본에 대해서입니다.
다중 파트 업로드 완료 요청	네, 조립된 객체에 대해서입니다.	네, 조립된 객체에 대해서입니다.	네, 조립된 객체에 대해서입니다.	네, 조립된 객체에 대해서입니다.

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.

2. 표에서 버킷 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

3. 버킷 옵션 탭에서 마지막 액세스 시간 업데이트 아코디언을 선택합니다.
4. 마지막 액세스 시간 업데이트를 활성화 또는 비활성화합니다.
5. *변경 사항 저장*을 선택합니다.

StorageGRID에서 **S3** 버킷에 대한 객체 버전 관리 변경

S3 테넌트를 사용하는 경우 S3 버킷의 버전 관리 상태를 변경할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.
- 필요한 스토리지 노드 및 사이트 수가 확보되었는지 확인했습니다. 사이트 내에서 두 개 이상의 스토리지 노드를 사용할 수 없거나 사이트를 사용할 수 없는 경우, 이러한 설정을 변경하지 못할 수 있습니다.

이 작업 정보

버킷에 대해 객체 버전 관리를 활성화하거나 일시 중지할 수 있습니다. 버킷에 대해 버전 관리를 활성화하면 버전 관리가 해제된 상태로 되돌릴 수 없습니다. 하지만 버킷에 대해 버전 관리를 일시 중지할 수는 있습니다.

- 비활성화됨: 버전 관리가 활성화된 적이 없습니다.
- 활성화됨: 버전 관리가 활성화되었습니다.
- 일시 중단됨: 이전에 버전 관리가 활성화되었으나 현재 일시 중단되었습니다.

자세한 내용은 다음을 참조하십시오.

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

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
2. 표에서 버킷 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

3. 버킷 옵션 탭에서 객체 버전 관리 아코디언을 선택합니다.
4. 이 버킷에 있는 객체에 대한 버전 관리 상태를 선택하십시오.

   그리드 간 복제에 사용되는 버킷의 경우 객체 버전 관리가 활성화된 상태로 유지되어야 합니다. S3 객체 잠금 또는 레거시 규정 준수가 활성화된 경우 * 객체 버전 관리 * 옵션은 비활성화됩니다.

옵션	설명
버전 관리 활성화	이 버킷에 각 객체의 모든 버전을 저장하려면 객체 버전 관리를 활성화하십시오. 그러면 필요에 따라 객체의 이전 버전을 검색할 수 있습니다. 버킷에 이미 있는 객체는 사용자가 수정할 때 버전이 관리됩니다.
버전 관리 일시 중단	새로운 객체 버전 생성을 더 이상 원하지 않는 경우 객체 버전 관리를 일시 중지하세요. 기존 객체 버전은 언제든지 조회할 수 있습니다.

5. *변경 사항 저장*을 선택합니다.

S3 오브젝트 잠금을 사용하여 **StorageGRID**에서 오브젝트 보존

버킷과 객체가 보존에 관한 규제 요건을 준수해야 하는 경우 S3 객체 잠금을 사용할 수 있습니다.

S3 Object Lock의 특정 기능을 사용하려면 그리드 관리자의 권한이 필요합니다.

S3 객체 잠금이란 무엇입니까**?**

StorageGRID S3 객체 잠금 기능은 Amazon Simple Storage Service(Amazon S3)의 S3 객체 잠금과 동일한 객체 보호 솔루션입니다.

StorageGRID 시스템에 대해 전역 S3 객체 잠금 설정이 활성화된 경우, S3 테넌트 계정은 S3 객체 잠금 활성화 여부와 관계없이 버킷을 생성할 수 있습니다. 버킷에 S3 객체 잠금이 활성화된 경우 버킷 버전 관리가 필요하며 자동으로 활성화됩니다.

*S3 객체 잠금이 없는 버킷*에는 보존 설정이 지정되지 않은 객체만 저장할 수 있습니다. 수집된 객체에는 보존 설정이 적용되지 않습니다.

*S3 객체 잠금이 설정된 버킷*에는 S3 클라이언트 애플리케이션에서 지정한 보존 설정이 있는 객체와 없는 객체가 모두 포함될 수 있습니다. 수집된 객체 중 일부에는 보존 설정이 적용됩니다.

*S3 객체 잠금 및 기본 보존 설정이 구성된 버킷*에는 보존 설정이 지정된 업로드된 객체와 보존 설정이 없는 새 객체가 모두 존재할 수 있습니다. 새 객체는 객체 수준에서 보존 설정이 구성되지 않았기 때문에 기본 설정을 사용합니다.

기본 보존 설정이 구성된 경우, 새로 수집되는 모든 객체에는 보존 설정이 적용됩니다. 객체 보존 설정이 없는 기존 객체는 영향을 받지 않습니다.

보존 모드

StorageGRID S3 객체 잠금 기능은 객체에 서로 다른 수준의 보호를 적용할 수 있는 두 가지 보존 모드를 지원합니다. 이러한 모드는 Amazon S3의 보존 모드와 동일합니다.

- 규정 준수 모드:
  - 해당 객체는 지정된 보존 기간이 만료될 때까지 삭제할 수 없습니다.
  - 객체의 보존 기간은 늘릴 수는 있지만 줄일 수는 없습니다.
  - 해당 날짜에 도달할 때까지 객체의 보존 기한을 제거할 수 없습니다.

- 거버넌스 모드에서:
  - 특별 권한을 가진 사용자는 요청에 바이패스 헤더를 사용하여 특정 보존 설정을 수정할 수 있습니다.
  - 이러한 사용자는 보존 기한이 도래하기 전에 개체 버전을 삭제할 수 있습니다.
  - 이러한 사용자는 개체의 보존 기간을 늘리거나 줄이거나 없앨 수 있습니다.

객체 버전 보존 설정

S3 객체 잠금이 활성화된 상태로 버킷이 생성된 경우, 사용자는 S3 클라이언트 애플리케이션을 사용하여 버킷에 추가되는 각 객체에 대해 다음과 같은 보존 설정을 선택적으로 지정할 수 있습니다.

- 보존 모드: 규정 준수 또는 거버넌스 중 하나.
- 보존 기한: 객체 버전의 보존 기한이 미래 날짜로 설정되어 있으면 해당 객체를 검색할 수는 있지만 삭제할 수는 없습니다.
- 법적 보류: 객체 버전에 법적 보류를 적용하면 해당 객체가 즉시 잠깁니다. 예를 들어, 조사 또는 법적 분쟁과 관련된 객체에 법적 보류를 적용해야 할 수 있습니다. 법적 보류에는 만료일이 없으며 명시적으로 해제될 때까지 유지됩니다. 법적 보류는 보존 기한과 무관합니다.

  법적 보존 조치가 적용된 객체는 보존 모드와 관계없이 누구도 삭제할 수 없습니다.

  객체 설정에 대한 자세한 내용은 "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.

버킷에 대한 기본 보존 설정

S3 객체 잠금이 활성화된 상태로 버킷을 생성하는 경우, 사용자는 선택적으로 버킷에 대해 다음과 같은 기본 설정을 지정할 수 있습니다.

- 기본 보존 모드: 규정 준수 또는 거버넌스 중 하나.
- 기본 보존 기간: 이 버킷에 추가된 새 객체 버전을 추가된 날로부터 보존할 기간입니다.

기본 버킷 설정은 자체 보존 설정이 없는 새 객체에만 적용됩니다. 기존 버킷 객체는 이러한 기본 설정을 추가하거나 변경해도 영향을 받지 않습니다.

"S3 버킷 생성" 및 "S3 객체 잠금 기본 보존 기간 업데이트"을(를) 참조하십시오.

S3 객체 잠금 작업

다음은 그리드 관리자와 테넌트 사용자를 위한 S3 객체 잠금 기능 사용에 필요한 주요 작업 목록입니다.

그리드 관리자

- 전체 StorageGRID 시스템에 대한 글로벌 S3 오브젝트 잠금 설정을 활성화합니다.
- 정보 라이프사이클 관리(ILM) 정책이 규정을 준수하는지 확인하십시오. 즉, 정책이 "S3 객체 잠금이 활성화된 버킷의 요구 사항"을 충족하는지 확인하십시오.
- 필요에 따라 테넌트가 Compliance를 보존 모드로 사용할 수 있도록 허용하십시오. 그렇지 않으면 Governance 모드만 허용됩니다.
- 필요에 따라 테넌트의 최대 보존 기간을 설정합니다.

테넌트 사용자

- S3 객체 잠금을 사용할 때 버킷 및 객체에 대한 고려 사항을 검토하십시오.
- 필요한 경우 그리드 관리자에게 문의하여 전역 S3 오브젝트 잠금 설정을 활성화하고 권한을 설정하십시오.
- S3 객체 잠금이 활성화된 버킷을 생성합니다.
- 선택적으로 버킷의 기본 보존 설정을 구성할 수 있습니다.
  - 기본 보존 모드: 그리드 관리자가 허용하는 경우 Governance 또는 Compliance.
  - 기본 보존 기간: 그리드 관리자가 설정한 최대 보존 기간보다 작거나 같아야 합니다.
- S3 클라이언트 애플리케이션을 사용하여 객체를 추가하고 필요에 따라 객체별 보존 기간을 설정할 수 있습니다.
  - 보존 모드. 그리드 관리자가 허용하는 경우 거버넌스 또는 규정 준수.
  - 보존 기한: 그리드 관리자가 설정한 최대 보존 기간에서 허용하는 날짜보다 작거나 같아야 합니다.

S3 객체 잠금이 활성화된 버킷에 대한 요구 사항

- StorageGRID 시스템에 대해 전역 S3 객체 잠금 설정이 활성화된 경우, 테넌트 관리자, 테넌트 관리 API 또는 S3 REST API를 사용하여 S3 객체 잠금이 활성화된 버킷을 생성할 수 있습니다.
- S3 객체 잠금을 사용하려면 버킷 생성 시 S3 객체 잠금을 활성화해야 합니다. 기존 버킷에는 S3 객체 잠금을 활성화할 수 없습니다.
- 버킷에 대해 S3 객체 잠금이 활성화되면 StorageGRID가 해당 버킷에 대한 버전 관리를 자동으로 활성화합니다. 버킷에 대해 S3 객체 잠금을 비활성화하거나 버전 관리를 일시 중지할 수는 없습니다.
- 선택적으로 Tenant Manager, 테넌트 관리 API 또는 S3 REST API를 사용하여 각 버킷에 대한 기본 보존 모드와 보존 기간을 지정할 수 있습니다. 버킷의 기본 보존 설정은 자체 보존 설정이 없는 버킷에 새로 추가되는 객체에만 적용됩니다. 객체 업로드 시 각 객체 버전에 대한 보존 모드와 보존 만료일을 지정하여 이러한 기본 설정을 재정의할 수 있습니다.
- S3 Object Lock이 활성화된 버킷에 대해 버킷 수명 주기 구성이 지원됩니다.
- CloudMirror S3 객체 잠금이 활성화된 버킷의 경우 CloudMirror 복제가 지원되지 않습니다.

S3 객체 잠금이 활성화된 버킷의 객체에 대한 요구 사항

- 객체 버전을 보호하려면 버킷에 대한 기본 보존 설정을 지정하거나 각 객체 버전에 대한 보존 설정을 지정할 수 있습니다. 객체 수준의 보존 설정은 S3 클라이언트 애플리케이션 또는 S3 REST API를 사용하여 지정할 수 있습니다.
- 보존 설정은 개별 객체 버전에 적용됩니다. 객체 버전은 보존 기간 설정과 법적 보존 설정을 모두 가질 수도 있고, 둘 중 하나만 가질 수도 있으며, 둘 다 가지지 않을 수도 있습니다. 객체에 보존 기간 설정 또는 법적 보존 설정을 지정하면 요청에 지정된 버전만 보호됩니다. 객체의 새 버전을 생성할 수 있지만 이전 버전은 계속 잠금 상태로 유지됩니다.

S3 객체 잠금이 활성화된 버킷의 객체 수명 주기

S3 객체 잠금이 활성화된 버킷에 저장되는 각 객체는 다음 단계를 거칩니다.

1. 오브젝트 수집

   S3 객체 잠금이 활성화된 버킷에 객체 버전이 추가되면 보존 설정이 다음과 같이 적용됩니다.

   - 객체에 대한 보존 설정이 지정된 경우 객체 수준 설정이 적용됩니다. 기본 버킷 설정은 무시됩니다.

- 객체에 대한 보존 설정이 지정되지 않은 경우, 기본 버킷 설정이 존재하면 해당 설정이 적용됩니다.
- 객체 또는 버킷에 대한 보존 설정이 지정되지 않은 경우 해당 객체는 S3 Object Lock으로 보호되지 않습니다.

보존 설정이 적용된 경우 객체와 S3 사용자 정의 메타데이터가 모두 보호됩니다.

2. 객체 보존 및 삭제

   보호된 각 객체의 여러 복사본은 지정된 보존 기간 동안 StorageGRID에 저장됩니다. 객체 복사본의 정확한 개수, 유형 및 저장 위치는 활성 ILM 정책의 규정 준수 규칙에 따라 결정됩니다. 보호 대상 객체를 보존 기한 이전에 삭제할 수 있는지 여부는 해당 객체의 보존 모드에 따라 다릅니다.

   - 법적 보존 조치가 적용된 객체는 보존 모드와 관계없이 누구도 삭제할 수 없습니다.

기존 규정 준수 버킷을 계속 관리할 수 있나요?

S3 객체 잠금 기능은 이전 StorageGRID 버전에서 제공되었던 규정 준수 기능을 대체합니다. 이전 버전의 StorageGRID를 사용하여 규정 준수 버킷을 생성한 경우 해당 버킷의 설정을 계속 관리할 수 있지만, 더 이상 새로운 규정 준수 버킷을 생성할 수 없습니다. 자세한 내용은 "NetApp 기술 자료: StorageGRID 11.5에서 기존 규정 준수 버킷을 관리하는 방법"을 참조하십시오.

StorageGRID에서 **S3** 오브젝트 잠금 기본 보존 업데이트

버킷 생성 시 S3 객체 잠금을 활성화한 경우, 버킷을 편집하여 기본 보존 설정을 변경할 수 있습니다. 기본 보존을 활성화(또는 비활성화)하고 기본 보존 모드 및 보존 기간을 설정할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.
- StorageGRID 시스템에 대해 S3 객체 잠금이 전역적으로 활성화되어 있으며, 버킷을 생성할 때 S3 객체 잠금을 활성화했습니다. 을 참조하십시오. "S3 객체 잠금을 사용하여 객체 보존".

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
2. 표에서 버킷 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

3. 버킷 옵션 탭에서 **S3** 객체 잠금 아코디언을 선택합니다.
4. 선택적으로 이 버킷에 대해 *기본 보존*을 활성화 또는 비활성화할 수 있습니다.

   이 설정 변경 사항은 버킷에 이미 있는 객체나 자체 보존 기간이 있는 객체에는 적용되지 않습니다.

5. *기본 보존*이 활성화된 경우 버킷에 대한 *기본 보존 모드*를 지정하십시오.

기본 보존 모드	설명
거버넌스	• `s3:BypassGovernanceRetention` 권한이 있는 사용자는 `x-amz-bypass-governance-retention: true` 요청 헤더를 사용하여 보존 설정을 우회할 수 있습니다. • 이러한 사용자는 보존 기한이 도래하기 전에 개체 버전을 삭제할 수 있습니다. • 이러한 사용자는 개체의 보존 기간을 늘리거나 줄이거나 없앨 수 있습니다.
규정 준수	• 해당 객체는 지정된 보존 기간이 만료될 때까지 삭제할 수 없습니다. • 객체의 보존 기간은 늘릴 수는 있지만 줄일 수는 없습니다. • 해당 날짜에 도달할 때까지 객체의 보존 기한을 제거할 수 없습니다. 참고: 그리드 관리자가 규정 준수 모드 사용을 허용해야 합니다.

6. *기본 보존*이 활성화된 경우 버킷의 *기본 보존 기간*을 지정하십시오.

   *기본 보존 기간*은 이 버킷에 새로 추가된 객체를 수집 시점부터 보존할 기간을 나타냅니다. 그리드 관리자가 설정한 테넌트의 최대 보존 기간 이하의 값을 지정하십시오.

   최대 보존 기간은 1일에서 100년까지의 값으로 설정할 수 있으며, 그리드 관리자가 테넌트를 생성할 때 설정됩니다. 기본 보존 기간을 설정할 때는 최대 보존 기간에 설정된 값을 초과할 수 없습니다. 필요한 경우 그리드 관리자에게 최대 보존 기간을 늘리거나 줄이도록 요청하십시오.

7. *변경 사항 저장*을 선택합니다.

StorageGRID에서 **S3** 버킷에 대한 **CORS** 구성

S3 버킷과 해당 버킷의 객체가 다른 도메인의 웹 애플리케이션에서 액세스할 수 있도록 하려면 CORS(교차 출처 리소스 공유)를 구성할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- GET CORS 구성 요청의 경우, 귀하는 "모든 버킷 관리 또는 모든 버킷 보기 권한"이(가) 있는 사용자 그룹에 속해 있어야 합니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.
- PUT CORS 구성 요청의 경우, "모든 버킷 권한 관리"이 있는 사용자 그룹에 속해 있어야 합니다. 이 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.
- "루트 액세스 권한"은(는) 모든 CORS 구성 요청에 대한 액세스를 제공합니다.

이 작업 정보

CORS(교차 출처 리소스 공유)는 한 도메인의 클라이언트 웹 애플리케이션이 다른 도메인의 리소스에 액세스할 수 있도록 하는 보안 메커니즘입니다. 예를 들어, `Images`이라는 `S3` 버킷을 사용하여 그래픽을 저장한다고 가정해 보겠습니다. `Images` 버킷에 대해 CORS를 구성하면 해당 버킷의 이미지를 웹사이트 `http://www.example.com`에 표시할 수 있습니다.

버킷에 **CORS** 활성화

단계

1. 텍스트 편집기를 사용하여 필요한 XML을 생성합니다. 이 예시는 S3 버킷에 CORS를 활성화하는 데 사용되는 XML을 보여줍니다. 구체적으로 다음과 같습니다.
   - 모든 도메인이 버킷에 GET 요청을 보낼 수 있도록 허용합니다.
   - 해당 `http://www.example.com` 도메인은 GET, POST 및 DELETE 요청만 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 "Amazon Web Services(AWS) 문서: Amazon Simple Storage Service 사용자 가이드"을 참조하십시오.

2. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
3. 표에서 버킷 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

4. 버킷 액세스 탭에서 **CORS(**교차 출처 리소스 공유**)** 아코디언을 선택합니다.
5. **CORS** 활성화 확인란을 선택합니다.
6. CORS 구성 XML을 텍스트 상자에 붙여넣습니다.
7. *변경 사항 저장*을 선택합니다.

CORS 설정 수정

단계

1. 텍스트 상자에서 CORS 구성 XML을 업데이트하거나 *지우기*를 선택하여 처음부터 다시 시작하세요.
2. *변경 사항 저장*을 선택합니다.

CORS 설정 비활성화

단계

1. **CORS** 활성화 확인란의 선택을 해제합니다.
2. *변경 사항 저장*을 선택합니다.

관련 정보

"관리 인터페이스에 대한 StorageGRID CORS 구성"

StorageGRID S3 버킷에서 객체 삭제

테넌트 관리자를 사용하여 하나 이상의 버킷에 있는 객체를 삭제할 수 있습니다.

고려사항 및 요구사항

다음 단계를 수행하기 전에 다음 사항에 유의하십시오.

- 버킷에서 객체를 삭제하면 StorageGRID는 선택한 각 버킷의 모든 객체와 모든 객체 버전을 StorageGRID 시스템의 모든 노드 및 사이트에서 영구적으로 제거합니다. StorageGRID는 또한 관련 객체 메타데이터를 제거합니다. 이 정보는 복구할 수 없습니다.
- 버킷에 있는 모든 객체를 삭제하는 데는 객체 수, 객체 복사본 수, 동시 작업 수에 따라 몇 분, 며칠 또는 몇 주가 걸릴 수 있습니다.
- 버킷에 "S3 객체 잠금 활성화됨"이(가) 있으면 해당 버킷은 수년 동안 객체 삭제 중**:** 읽기 전용 상태로 유지될 수 있습니다.

  S3 Object Lock을 사용하는 버킷은 모든 객체의 보존 날짜가 도래하고 모든 법적 보류가 해제될 때까지 객체 삭제 중**:** 읽기 전용 상태를 유지합니다.

- 객체가 삭제되는 동안 버킷의 상태는 *객체 삭제 중: 읽기 전용*이 됩니다. 이 상태에서는 버킷에 새 객체를 추가할 수 없습니다.
- 모든 객체가 삭제되면 버킷은 읽기 전용 상태로 유지됩니다. 다음 중 하나를 수행할 수 있습니다.
  - 버킷을 쓰기 모드로 되돌리고 새 객체에 대해 재사용하십시오.
  - 버킷을 삭제합니다
  - 버킷의 이름을 향후 사용을 위해 예약하려면 읽기 전용 모드로 유지하십시오.
- 버킷에 객체 버전 관리가 활성화된 경우, StorageGRID 11.8 이상에서 생성된 삭제 마커는 버킷의 객체 삭제 작업을 사용하여 제거할 수 있습니다.
- 버킷에 객체 버전 관리가 활성화된 경우, 객체 삭제 작업은 StorageGRID 11.7 이하 버전에서 생성된 삭제 마커를 제거하지 않습니다. 버킷에서 객체를 삭제하는 방법에 대한 자세한 내용은 "S3 버전 관리 객체가 삭제되는 방법"을 참조하십시오.
- "교차 그리드 복제"를 사용하는 경우 다음 사항에 유의하십시오.
  - 이 옵션을 사용해도 다른 그리드의 버킷에서 객체가 삭제되지는 않습니다.
  - 소스 버킷에 대해 이 옵션을 선택하면 다른 그리드의 대상 버킷에 객체를 추가할 경우 그리드 간 복제 실패 경고가 발생합니다. 다른 그리드의 버킷에 아무도 객체를 추가하지 않도록 보장할 수 없는 경우, "그리드 간 복제 비활성화" 모든 버킷 객체를 삭제하기 전에 해당 버킷에 대해 이 옵션을 선택하십시오.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "루트 액세스 권한"이(가) 있는 사용자 그룹에 속해 있습니다. 이 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.

   버킷 페이지가 나타나고 기존의 모든 S3 버킷이 표시됩니다.

2. 작업 메뉴 또는 특정 버킷의 세부 정보 페이지를 사용합니다.

   작업 메뉴

   a. 객체를 삭제할 각 버킷에 대한 확인란을 선택하십시오.

   b. 작업 > *버킷에서 객체 삭제*를 선택합니다.

   상세 정보 페이지

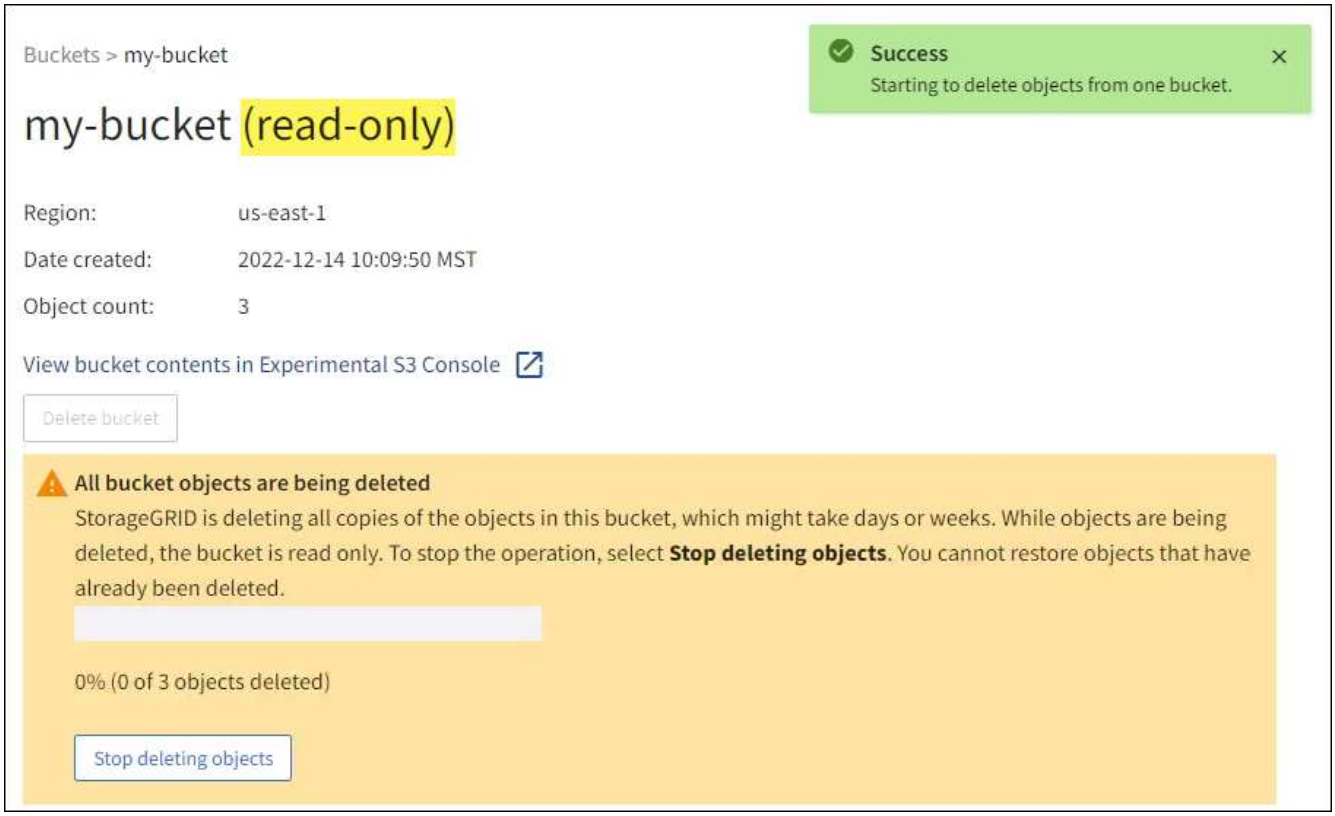
   a. 버킷 이름을 선택하여 세부 정보를 표시합니다.

   b. *버킷에서 객체 삭제*를 선택합니다.

3. 확인 대화 상자가 나타나면 세부 정보를 검토하고 *Yes*를 입력한 다음 *OK*를 선택하십시오.
4. 삭제 작업이 시작될 때까지 기다리십시오.

   몇 분 후:

   - 버킷 세부 정보 페이지에 노란색 상태 배너가 나타납니다. 진행률 표시줄은 삭제된 객체의 비율을 나타냅니다.
   - 버킷 세부 정보 페이지에서 버킷 이름 뒤에 *(읽기 전용)*이 표시됩니다.
   - 버킷 페이지에서 버킷 이름 옆에 *(객체 삭제: 읽기 전용)*이 표시됩니다.

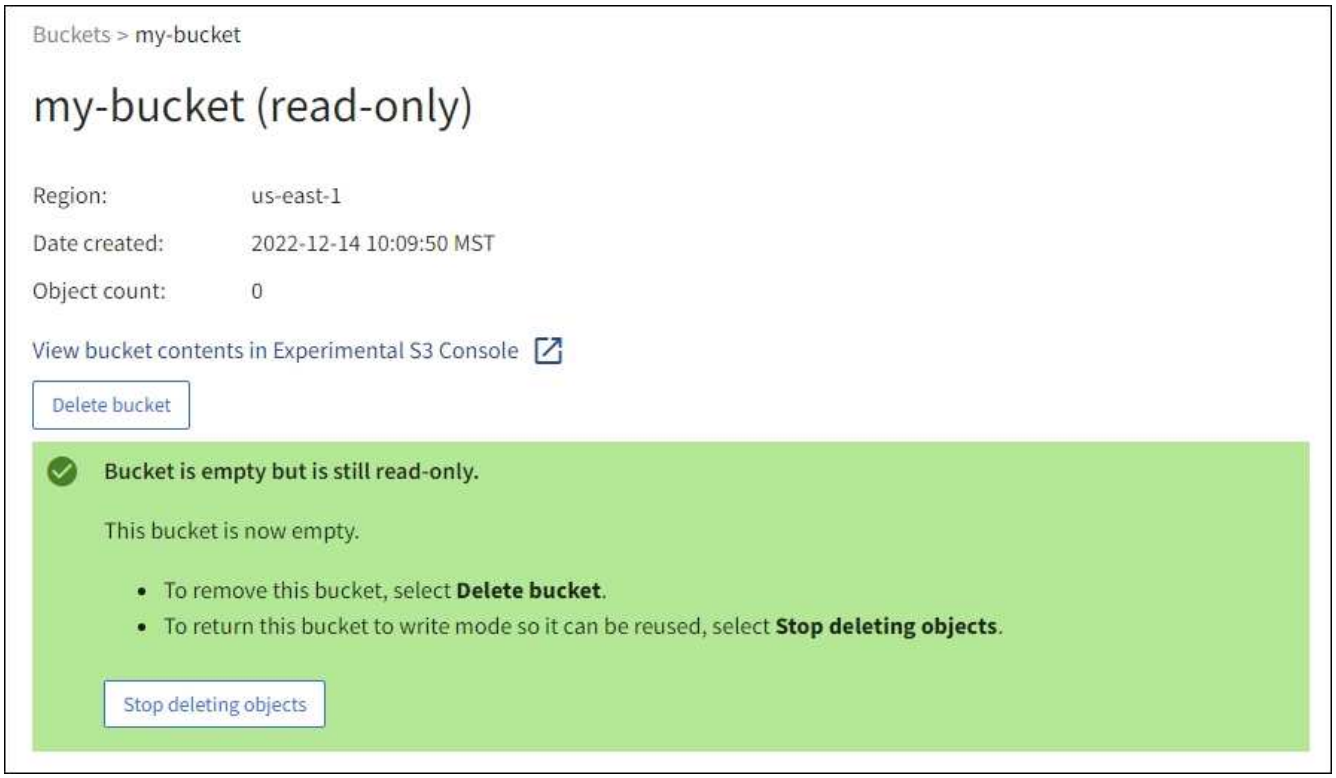


5. 작업 실행 중에 필요에 따라 *객체 삭제 중지*를 선택하여 프로세스를 중지할 수 있습니다. 그런 다음 선택적으로 *버킷에서 객체 삭제*를 선택하여 프로세스를 다시 시작할 수 있습니다.

   *객체 삭제 중지*를 선택하면 버킷이 쓰기 모드로 돌아가지만, 삭제된 객체에 액세스하거나 복원할 수는 없습니다.

6. 작업이 완료될 때까지 기다리세요.

   버킷이 비어 있으면 상태 배너가 업데이트되지만 버킷은 읽기 전용 상태로 유지됩니다.



7. 다음 중 하나를 수행하십시오.
   - 페이지를 종료하면 버킷이 읽기 전용 모드로 유지됩니다. 예를 들어, 나중에 사용할 버킷 이름을 예약하기 위해 빈 버킷을 읽기 전용 모드로 유지할 수 있습니다.
   - 버킷을 삭제합니다. 버킷 하나를 삭제하려면 버킷 삭제*를 선택하고, 여러 버킷을 삭제하려면 버킷 페이지로 돌아가서 *작업 > *버킷 삭제*를 선택합니다.

     모든 객체를 삭제한 후에도 버전 관리 버킷을 삭제할 수 없는 경우 삭제 마커가 남아 있을 수 있습니다. 버킷을 삭제하려면 남아 있는 모든 삭제 마커를 제거해야 합니다.

   - 버킷을 쓰기 모드로 되돌리고 필요에 따라 새 객체에 재사용할 수 있습니다. 단일 버킷에 대해 객체 삭제 중지*를 선택하거나, 버킷 페이지로 돌아가서 여러 버킷에 대해 *작업 > *객체 삭제 중지*를 선택할 수 있습니다.

StorageGRID S3 버킷 삭제

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

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다. 이러한 권한은 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.
- 삭제하려는 버킷이 비어 있습니다. 삭제하려는 버킷이 비어 있지 않은 경우, "버킷에서 객체를 삭제합니다".

이 작업 정보

이 지침에서는 테넌트 관리자를 사용하여 S3 버킷을 삭제하는 방법을 설명합니다. "테넌트 관리 API" 또는 "S3 REST API"를 사용하여 S3 버킷을 삭제할 수도 있습니다.

객체, 현재 버전이 아닌 객체 버전 또는 삭제 마커가 포함된 S3 버킷은 삭제할 수 없습니다. S3 버전 관리 객체 삭제 방법에 대한 자세한 내용은 "객체가 삭제되는 방법"을 참조하십시오.

단계

1. 대시보드에서 버킷 보기*를 선택하거나 *스토리지(S3) > *버킷*을 선택합니다.

   버킷 페이지가 나타나고 기존의 모든 S3 버킷이 표시됩니다.

2. 작업 메뉴 또는 특정 버킷의 세부 정보 페이지를 사용합니다.

   작업 메뉴

   a. 삭제하려는 각 버킷의 확인란을 선택합니다.

   b. 작업 > *버킷 삭제*를 선택합니다.

   상세 정보 페이지

   a. 버킷 이름을 선택하여 세부 정보를 표시합니다.

   b. *버킷 삭제*를 선택합니다.

3. 확인 대화 상자가 나타나면 *예*를 선택하십시오.

   StorageGRID는 각 버킷이 비어 있는지 확인한 후 각 버킷을 삭제합니다. 이 작업은 몇 분 정도 소요될 수 있습니다.

   버킷이 비어 있지 않으면 오류 메시지가 표시됩니다. 버킷을 삭제하려면 먼저 "버킷에 있는 모든 객체와 삭제 표시를 삭제합니다."해야 합니다.

StorageGRID에서 **S3** 콘솔 사용

S3 콘솔을 사용하여 S3 버킷의 객체를 보고 관리할 수 있습니다.

S3 콘솔을 사용하면 다음을 수행할 수 있습니다.

- 객체 업로드, 다운로드, 이름 변경, 복사, 이동 및 삭제
- 객체 버전 보기, 되돌리기, 다운로드 및 삭제
- 접두사로 객체 검색
- 객체 태그 관리
- 객체 메타데이터 보기
- 폴더를 보고, 만들고, 이름을 바꾸고, 복사하고, 이동하고, 삭제할 수 있습니다.

S3 Console은 가장 일반적인 경우에 대해 향상된 사용자 인터페이스 환경을 제공합니다. 모든 상황에서 CLI 또는 API 작업을 대체하도록 설계된 것은 아닙니다.

S3 콘솔을 사용하여 작업하는 데 시간이 너무 오래 걸리는 경우(예: 몇 분 또는 몇 시간) 다음 사항을 고려하십시오.

- 선택된 객체 수 줄이기
- 그래픽 방식이 아닌 (API 또는 CLI) 방법을 사용하여 데이터에 액세스

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 객체를 관리하려면 루트 액세스 권한이 있는 사용자 그룹에 속해야 합니다. 또는 S3 콘솔 탭 사용 권한과 모든 버킷 보기 권한 또는 모든 버킷 관리 권한 중 하나를 가진 사용자 그룹에 속해야 합니다. "테넌트 관리 권한"을 참조하십시오.
- 사용자에게 S3 그룹 또는 버킷 정책이 구성되었습니다. "버킷 및 그룹 액세스 정책 사용"을 참조하십시오.
- 사용자 액세스 키 ID와 비밀 액세스 키를 알고 있습니다. 선택적으로, `.csv` 이 정보가 포함된 파일이 있을 수도 있습니다. "액세스 키 생성 지침"을 참조하십시오.

단계

1. 스토리지 > 버킷 > *버킷 이름*을 선택합니다.
2. S3 콘솔 탭을 선택합니다.
3. 액세스 키 ID와 비밀 액세스 키를 해당 필드에 붙여넣으세요. 또는 *액세스 키 업로드*를 선택하고 사용자의 `.csv` 파일을 선택하세요.
4. *로그인*을 선택합니다.
5. 버킷 객체 목록이 표시됩니다. 필요에 따라 객체를 관리할 수 있습니다.

추가 정보

- 접두사로 검색: 접두사 검색 기능은 현재 폴더를 기준으로 특정 단어로 시작하는 개체만 검색합니다. 해당 단어가 다른 곳에 포함된 개체는 검색에서 제외됩니다. 이 규칙은 폴더 내의 개체에도 적용됩니다. 예를 들어, `folder1/folder2/somefile-`를 검색하면 `folder1/folder2/`폴더 내에 있고 `somefile-`단어로 시작하는 개체가 반환됩니다.
- 드래그 앤 드롭: 컴퓨터의 파일 관리자에서 S3 콘솔로 파일을 드래그 앤 드롭할 수 있습니다. 단, 폴더는 업로드할 수 없습니다.
- 폴더 작업: 폴더를 이동, 복사 또는 이름을 변경하면 폴더 안의 모든 개체가 하나씩 순차적으로 업데이트되므로 시간이 다소 걸릴 수 있습니다.
- 버킷 버전 관리가 비활성화된 경우 영구 삭제: 버전 관리가 비활성화된 버킷에서 객체를 덮어쓰거나 삭제하면 해당 작업은 영구적으로 적용됩니다. "버킷의 객체 버전 관리 변경"을 참조하십시오.

S3 플랫폼 서비스 관리

S3 플랫폼 서비스

StorageGRID용 플랫폼 서비스에 대해 알아보십시오

플랫폼 서비스를 구현하기 전에 해당 서비스 사용에 대한 개요 및 고려 사항을 검토하십시오.

S3에 대한 자세한 내용은 "S3 REST API 사용"을 참조하십시오.

플랫폼 서비스 개요

StorageGRID 플랫폼 서비스를 이용하면 이벤트 알림과 S3 객체 및 객체 메타데이터의 복사본을 외부 대상으로 전송할 수 있으므로 하이브리드 클라우드 전략을 구현하는 데 도움이 됩니다.

플랫폼 서비스의 대상 위치는 일반적으로 StorageGRID 배포 환경 외부에 있기 때문에, 플랫폼 서비스를 통해 외부 스토리지 리소스, 알림 서비스, 데이터 검색 또는 분석 서비스를 활용하여 강력한 기능과 유연성을 확보할 수 있습니다.

하나의 S3 버킷에 대해 다양한 플랫폼 서비스를 조합하여 구성할 수 있습니다. 예를 들어, StorageGRID S3 버킷에서 "CloudMirror 서비스" 및 "알림"를 모두 구성하여 특정 객체를 Amazon Simple Storage Service(S3)에 미러링하는 동시에 각 객체에 대한 알림을 타사 모니터링 애플리케이션으로 전송하여 AWS 비용을 추적할 수 있습니다.

플랫폼 서비스 사용은 StorageGRID 관리자가 Grid Manager 또는 Grid Management API를 사용하여 각 테넌트 계정에 대해 활성화해야 합니다.

플랫폼 서비스 구성 방법

플랫폼 서비스는 "테넌트 관리자" 또는 "테넌트 관리 API"를 사용하여 구성한 외부 엔드포인트와 통신합니다. 각 엔드포인트는 StorageGRID S3 버킷, Amazon Web Services 버킷, Amazon SNS 토픽, 웹훅 엔드포인트 또는 로컬, AWS 또는 다른 곳에 호스팅된 Elasticsearch 클러스터와 같은 외부 대상을 나타냅니다.

외부 엔드포인트를 생성한 후 버킷에 XML 구성을 추가하여 버킷에 대한 플랫폼 서비스를 활성화할 수 있습니다. XML 구성은 버킷이 처리해야 할 객체, 버킷이 수행해야 할 작업, 그리고 서비스에 사용할 엔드포인트를 지정합니다.

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

- 키가 `/images`로 시작하는 모든 객체를 Amazon S3 버킷으로 복제하려면 소스 버킷에 복제 구성을 추가해야 합니다.
- 이러한 객체가 버킷에 저장될 때 알림을 보내려면 알림 구성을 추가해야 합니다.
- 이러한 객체의 메타데이터를 색인화하려면 검색 통합을 구현하는 데 사용되는 메타데이터 알림 구성을 추가해야 합니다.

구성 XML의 형식은 StorageGRID 플랫폼 서비스를 구현하는 데 사용되는 S3 REST API에 의해 결정됩니다.

플랫폼 서비스	**S3 REST API**	참조하십시오
CloudMirror 복제	• GetBucketReplication • PutBucketReplication	• "CloudMirror 복제" • "버킷에 대한 작업"
알림	• GetBucketNotificationConfiguration • PutBucketNotificationConfiguration	• "알림" • "버킷에 대한 작업"
검색 통합	• GET 버킷 메타데이터 알림 구성 • PUT 버킷 메타데이터 알림 구성	• "검색 통합" • "StorageGRID 사용자 지정 작업"

플랫폼 서비스 사용 시 고려 사항

고려 사항	세부 정보
대상 엔드포인트 모니터링	각 대상 엔드포인트의 가용성을 모니터링해야 합니다. 대상 엔드포인트에 대한 연결이 장시간 끊어지고 요청이 대량으로 누적된 경우, StorageGRID에 대한 추가 클라이언트 요청(예: PUT 요청)이 실패할 수 있습니다. 엔드포인트에 다시 연결할 수 있게 되면 이러한 실패한 요청을 재시도해야 합니다.
대상 엔드포인트 스로틀링	StorageGRID 소프트웨어는 버킷으로 들어오는 S3 요청 속도가 대상 엔드포인트가 요청을 수신할 수 있는 속도를 초과하는 경우 해당 요청 속도를 제한할 수 있습니다. 속도 제한은 대상 엔드포인트로 전송될 요청이 누적되어 대기 중인 경우에만 발생합니다. 눈에 띄는 유일한 영향은 수신 S3 요청 실행 시간이 더 오래 걸린다는 것입니다. 성능 저하가 현저하게 감지되기 시작하면 수집 속도를 줄이거나 용량이 더 큰 엔드포인트를 사용해야 합니다. 요청 백로그가 계속 증가하면 클라이언트의 S3 작업(예: PUT 요청)이 결국 실패하게 됩니다. CloudMirror 요청은 일반적으로 검색 통합 또는 이벤트 알림 요청보다 더 많은 데이터 전송을 포함하므로 대상 엔드포인트의 성능에 더 큰 영향을 받을 가능성이 높습니다.
순서 보장	StorageGRID는 사이트 내 객체에 대한 작업 순서를 보장합니다. 객체에 대한 모든 작업이 동일한 사이트 내에서 이루어지는 한, 최종 객체 상태(복제용)는 항상 StorageGRID의 상태와 동일합니다. StorageGRID는 StorageGRID 사이트 간 작업 시 요청 순서를 최대한 정확하게 유지하려고 노력합니다. 예를 들어, 처음에 A 사이트에 객체를 기록한 후 나중에 B 사이트에서 동일한 객체를 덮어쓰는 경우, CloudMirror가 대상 버킷에 복제하는 최종 객체가 최신 객체라는 보장은 없습니다.
ILM 기반 객체 삭제	AWS CRR 및 Amazon Simple Notification Service의 삭제 동작과 일치하도록, StorageGRID ILM 규칙으로 인해 소스 버킷의 객체가 삭제될 경우 CloudMirror 및 이벤트 알림 요청이 전송되지 않습니다. 예를 들어, ILM 규칙에 따라 객체가 14일 후에 삭제되는 경우 CloudMirror 또는 이벤트 알림 요청이 전송되지 않습니다. 이와 대조적으로, ILM으로 인해 객체가 삭제될 때 검색 통합 요청이 전송됩니다.

고려 사항	세부 정보
Kafka 엔드포인트 사용	Kafka 엔드포인트의 경우 상호 TLS는 지원되지 않습니다. 따라서 Kafka 브로커 구성에서 `ssl.client.auth`을(를) `required`로 설정한 경우 Kafka 엔드포인트 구성 문제가 발생할 수 있습니다. Kafka 엔드포인트 인증에는 다음과 같은 인증 유형이 사용됩니다. 이러한 유형은 Amazon SNS와 같은 다른 엔드포인트 인증에 사용되는 유형과 다르며, 사용자 이름과 암호 자격 증명이 필요합니다. • SASL/PLAIN • SASL/SCRAM-SHA-256 • SASL/SCRAM-SHA-512 참고**:** 구성된 스토리지 프록시 설정은 Kafka 플랫폼 서비스 엔드포인트에는 적용되지 않습니다.

CloudMirror 복제 서비스 사용 시 고려 사항

고려 사항	세부 정보
복제 상태	StorageGRID는 `x-amz-replication-status` 헤더를 지원하지 않습니다.
객체 크기	CloudMirror 복제 서비스를 통해 대상 버킷으로 복제할 수 있는 객체의 최대 크기는 5TiB이며, 이는 지원되는 최대 객체 크기와 동일합니다. 참고: 단일 PutObject 작업의 최대 권장 크기는 5GiB(5,368,709,120바이트)입니다. 객체 크기가 5GiB보다 큰 경우 멀티파트 업로드를 사용하십시오.
버킷 버전 관리 및 버전 ID	StorageGRID의 소스 S3 버킷에 버전 관리가 활성화되어 있는 경우 대상 버킷에도 버전 관리를 활성화해야 합니다. 버전 관리를 사용할 때, S3 프로토콜의 제한 사항으로 인해 대상 버킷에서 객체 버전의 순서는 CloudMirror 서비스에서 보장되지 않고 최선을 다해 유지된다는 점에 유의하십시오. 참고: StorageGRID의 소스 버킷에 대한 버전 ID는 대상 버킷의 버전 ID와 관련이 없습니다.
객체 버전에 대한 태깅	CloudMirror 서비스는 S3 프로토콜의 제한 사항으로 인해 버전 ID를 제공하는 PutObjectTagging 또는 DeleteObjectTagging 요청을 복제하지 않습니다. 소스와 대상의 버전 ID가 서로 관련이 없기 때문에 특정 버전 ID에 대한 태그 업데이트가 복제되도록 보장할 방법이 없습니다. 반면, CloudMirror 서비스는 버전 ID를 지정하지 않는 PutObjectTagging 요청이나 DeleteObjectTagging 요청은 복제합니다. 이러한 요청은 최신 키(또는 버킷에 버전이 지정된 경우 최신 버전)에 대한 태그를 업데이트합니다. 태그가 포함된 일반적인 데이터 수집(태그 업데이트 제외)도 복제됩니다.

고려 사항	세부 정보
멀티파트 업로드 및 `ETag` 값	멀티파트 업로드를 사용하여 업로드된 객체를 미러링할 때 CloudMirror 서비스는 파트를 보존하지 않습니다. 따라서 미러링된 객체의 `ETag` 값은 원본 객체의 `ETag` 값과 다릅니다.
SSE-C로 암호화된 객체(고객이 제공한 키를 사용한 서버 측 암호화)	CloudMirror 서비스는 SSE-C로 암호화된 객체를 지원하지 않습니다. CloudMirror 복제를 위해 소스 버킷에 객체를 수집하려고 할 때 요청에 SSE-C 요청 헤더가 포함되어 있으면 작업이 실패합니다.
S3 객체 잠금이 활성화된 버킷	S3 객체 잠금이 활성화된 소스 또는 대상 버킷의 경우 복제가 지원되지 않습니다.

StorageGRID CloudMirror 복제 서비스에 대해 알아보세요

StorageGRID 버킷에 추가된 특정 객체를 하나 이상의 외부 대상 버킷으로 복제하도록 하려면 S3 버킷에 대해 CloudMirror 복제를 활성화할 수 있습니다.

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

CloudMirror 소스 버킷에 S3 객체 잠금이 활성화된 경우 CloudMirror 복제가 지원되지 않습니다.

CloudMirror 및 **ILM**

CloudMirror 복제는 그리드의 활성 ILM 정책과 독립적으로 작동합니다. CloudMirror 서비스는 객체가 소스 버킷에 저장되는 즉시 복제하여 가능한 한 빨리 대상 버킷으로 전달합니다. 복제된 객체의 전달은 객체 수집이 성공했을 때 시작됩니다.

CloudMirror 및 크로스그리드 복제

CloudMirror 복제는 크로스 그리드 복제 기능과 중요한 유사점과 차이점을 가지고 있습니다. 자세한 내용은 "크로스 그리드 복제와 CloudMirror 복제 비교"을 참조하십시오.

CloudMirror 및 **S3** 버킷

CloudMirror 복제는 일반적으로 외부 S3 버킷을 대상으로 구성됩니다. 하지만 다른 StorageGRID 배포 또는 S3 호환 서비스를 대상으로 복제를 구성할 수도 있습니다.

기존 버킷

CloudMirror 복제를 기존 버킷에 대해 활성화하면 해당 버킷에 새로 추가된 객체만 복제됩니다. 버킷에 이미 있는 객체는 복제되지 않습니다. 기존 객체를 강제로 복제하려면 객체 복사를 수행하여 기존 객체의 메타데이터를 업데이트하면 됩니다.

CloudMirror 복제를 사용하여 객체를 Amazon S3 대상으로 복사하는 경우, Amazon S3는 각 PUT 요청 헤더 내 사용자 정의 메타데이터의 크기를 2KB로 제한한다는 점에 유의하십시오. 객체의 사용자 정의 메타데이터가 2KB를 초과하는 경우 해당 객체는 복제되지 않습니다.

여러 대상 버킷

단일 버킷의 객체를 여러 대상 버킷으로 복제하려면 복제 구성 XML에서 각 규칙에 대한 대상을 지정해야 합니다. 객체를 동시에 두 개 이상의 버킷으로 복제할 수는 없습니다.

버전 관리 버킷 또는 버전 관리되지 않는 버킷

버전 관리되는 버킷 또는 버전 관리되지 않는 버킷에서 CloudMirror 복제를 구성할 수 있습니다. 대상 버킷은 버전 관리되는 버킷이거나 버전 관리되지 않는 버킷일 수 있습니다. 버전 관리되는 버킷과 버전 관리되지 않는 버킷의 어떠한 조합도 사용할 수 있습니다. 예를 들어, 버전 관리되지 않는 소스 버킷의 대상으로 버전 관리되는 버킷을 지정하거나 그 반대로 지정할 수 있습니다. 버전 관리되지 않는 버킷 간에 복제할 수도 있습니다.

삭제, 복제 루프 및 이벤트

삭제 동작

이는 Amazon S3 서비스의 교차 리전 복제(CRR)와 동일한 삭제 동작입니다. 소스 버킷에서 객체를 삭제해도 대상 버킷의 복제된 객체는 삭제되지 않습니다. 소스 및 대상 버킷 모두 버전 관리가 적용된 경우 삭제 마커가 복제됩니다. 대상 버킷에 버전 관리가 적용되지 않은 경우 소스 버킷에서 객체를 삭제해도 삭제 마커가 대상 버킷으로 복제되지 않으며 대상 객체도 삭제되지 않습니다.

복제 루프로부터의 보호

객체가 대상 버킷으로 복제될 때 StorageGRID는 해당 객체를 "복제본"으로 표시합니다. 대상 StorageGRID 버킷은 복제본으로 표시된 객체를 다시 복제하지 않으므로 의도치 않은 복제 루프를 방지할 수 있습니다. 이 복제본 표시는 StorageGRID 내부적으로 처리되며, Amazon S3 버킷을 대상으로 사용할 때 AWS CRR을 활용하는 데 영향을 미치지 않습니다.

복제본을 표시하는 데 사용되는 사용자 지정 헤더는 `x-ntap-sg-replica`입니다. 이 표시는 계단식 미러링을 방지합니다. StorageGRID는 두 그리드 간의 양방향 CloudMirror를 지원합니다.

대상 버킷의 이벤트

대상 버킷의 이벤트 고유성 및 순서는 보장되지 않습니다. 전송 성공을 보장하기 위해 수행되는 작업으로 인해 소스 객체의 동일한 복사본이 대상에 두 개 이상 전달될 수 있습니다. 드물지만, 동일한 객체가 두 개 이상의 StorageGRID 사이트에서 동시에 업데이트되는 경우 대상 버킷의 작업 순서가 소스 버킷의 이벤트 순서와 일치하지 않을 수 있습니다.

S3 버킷에 대한 **StorageGRID** 이벤트 알림에 대해 알아보십시오

StorageGRID가 지정된 이벤트에 대한 알림을 대상 Kafka 클러스터, 웹훅 엔드포인트 또는 Amazon Simple Notification Service로 전송하도록 하려면 S3 버킷에 대한 이벤트 알림을 활성화할 수 있습니다.

예를 들어, 버킷에 추가되는 각 객체에 대해 관리자에게 알림을 보내도록 구성할 수 있습니다. 여기서 객체는 중요한 시스템 이벤트와 관련된 로그 파일을 나타냅니다.

이벤트 알림은 알림 구성에 지정된 대로 소스 버킷에서 생성되어 대상으로 전달됩니다. 객체와 관련된 이벤트가 성공적으로 처리되면 해당 이벤트에 대한 알림이 생성되어 전달 대기열에 추가됩니다.

알림의 고유성과 순서는 보장되지 않습니다. 전달 성공을 보장하기 위해 수행되는 작업으로 인해 동일한 이벤트에 대한 알림이 대상에 두 개 이상 전달될 수 있습니다. 또한 전달이 비동기적으로 이루어지기 때문에, 특히 서로 다른 StorageGRID 사이트에서 시작된 작업의 경우, 대상에서의 알림 시간 순서가 소스 버킷의 이벤트 순서와 일치한다는 보장은 없습니다. Amazon S3 설명서에 설명된 대로 이벤트 메시지의 `sequencer` 키를 사용하여 특정 객체에 대한 이벤트 순서를 확인할 수 있습니다.

StorageGRID 이벤트 알림은 일부 제한 사항이 있지만 Amazon S3 API를 따릅니다.

- 지원되는 이벤트 유형은 다음과 같습니다.
  - s3:ObjectCreated:
  - s3:ObjectCreated:Put
  - s3:ObjectCreated:Post
  - s3:ObjectCreated:Copy
  - s3:ObjectCreated:CompleteMultipartUpload
  - s3:ObjectRemoved:
  - s3:ObjectRemoved:Delete
  - s3:ObjectRemoved:DeleteMarkerCreated
  - s3:ObjectRestore:Post
- StorageGRID에서 전송된 이벤트 알림은 표준 JSON 형식을 사용하지만, 아래 표와 같이 일부 키는 포함하지 않고 다른 키에는 특정 값을 사용합니다.

키 이름	**StorageGRID** 값
eventSource	`sgws:s3`
awsRegion	포함되지 않음
x-amz-id-2	포함되지 않음
arn	`urn:sgws:s3:::bucket_name`

StorageGRID 검색 통합 서비스에 대해 알아보십시오

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

검색 통합 서비스는 StorageGRID 의 사용자 지정 서비스로, 객체가 생성되거나 삭제될 때, 또는 메타데이터나 태그가 업데이트될 때마다 S3 객체 메타데이터를 대상 엔드포인트로 자동으로 비동기 전송합니다. 대상 서비스에서 제공하는 정교한 검색, 데이터 분석, 시각화 또는 머신러닝 도구를 사용하여 객체 데이터를 검색, 분석하고 유용한 정보를 얻을 수 있습니다.

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

S3 객체 잠금이 활성화된 버킷에서 Elasticsearch 통합을 구성할 수 있지만, 객체의 S3 객체 잠금 메타데이터(보존 기한 및 법적 보존 상태 포함)는 Elasticsearch로 전송되는 메타데이터에 포함되지 않습니다.

> 검색 통합 서비스는 객체 메타데이터를 대상으로 전송하기 때문에 해당 구성 XML을 "메타데이터 알림 구성 XML"이라고 합니다. 이 구성 XML은 이벤트 알림을 활성화하는 데 사용되는 "알림 구성 XML"과는 다릅니다.

검색 통합 및 **S3** 버킷

버전 관리 여부와 관계없이 모든 버킷에 대해 검색 통합 서비스를 활성화할 수 있습니다. 검색 통합은 작업할 객체와 객체 메타데이터의 대상을 지정하는 메타데이터 알림 구성 XML을 버킷에 연결하여 구성합니다.

메타데이터 알림은 버킷 이름, 객체 이름, 그리고 버전 ID(있는 경우)로 구성된 JSON 문서 형식으로 생성됩니다. 각 메타데이터 알림에는 객체의 모든 태그 및 사용자 메타데이터 외에도 객체에 대한 표준 시스템 메타데이터 세트가 포함됩니다.

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

검색 알림

메타데이터 알림은 다음과 같은 경우에 생성되어 전송 대기열에 추가됩니다.

- 객체가 생성되었습니다.
- 객체가 삭제됩니다. 여기에는 그리드의 ILM 정책 작동으로 인해 객체가 삭제되는 경우도 포함됩니다.
- 객체 메타데이터 또는 태그가 추가, 업데이트 또는 삭제됩니다. 메타데이터 업데이트 시에는 변경된 값뿐만 아니라 전체 메타데이터 및 태그 세트가 항상 전송됩니다.

버킷에 메타데이터 알림 구성 XML을 추가하면 새로 생성하는 모든 객체와 데이터, 사용자 메타데이터 또는 태그를 업데이트하여 수정하는 모든 객체에 대해 알림이 전송됩니다. 그러나 버킷에 이미 있는 객체에 대해서는 알림이 전송되지 않습니다. 버킷에 있는 모든 객체의 객체 메타데이터가 대상으로 전송되도록 하려면 다음 중 하나를 수행해야 합니다.

- 버킷을 생성한 직후, 그리고 객체를 추가하기 전에 검색 통합 서비스를 구성하십시오.
- 버킷에 이미 있는 모든 객체에 대해 메타데이터 알림 메시지가 대상으로 전송되도록 하는 작업을 수행합니다.

검색 통합 서비스 및 **Elasticsearch**

StorageGRID 검색 통합 서비스는 Elasticsearch 클러스터를 대상으로 지원합니다. 다른 플랫폼 서비스와 마찬가지로, 대상은 서비스의 구성 XML에서 URN이 사용되는 엔드포인트에 지정됩니다. "NetApp 상호 운용성 매트릭스 도구"을 사용하여 지원되는 Elasticsearch 버전을 확인하십시오.

플랫폼 서비스 엔드포인트 관리

StorageGRID에서 플랫폼 서비스 엔드포인트 구성

버킷에 대한 플랫폼 서비스를 구성하기 전에 플랫폼 서비스의 대상으로 사용할 엔드포인트를 하나 이상 구성해야 합니다.

플랫폼 서비스에 대한 액세스는 StorageGRID 관리자가 테넌트별로 활성화합니다. 플랫폼 서비스 엔드포인트를 생성하거나 사용하려면, 스토리지 노드가 외부 엔드포인트 리소스에 액세스할 수 있도록 네트워킹이 구성된 그리드에서 엔드포인트 관리 또는 루트 액세스 권한이 있는 테넌트 사용자여야 합니다. 단일 테넌트에 대해 최대 500개의 플랫폼 서비스 엔드포인트를 구성할 수 있습니다. 자세한 내용은 StorageGRID 관리자에게 문의하십시오.

플랫폼 서비스 엔드포인트란 무엇인가요?

플랫폼 서비스 엔드포인트는 StorageGRID가 외부 대상에 액세스하는 데 필요한 정보를 지정합니다.

예를 들어 StorageGRID 버킷의 객체를 Amazon S3 버킷으로 복제하려면 StorageGRID가 Amazon의 대상 버킷에 액세스하는 데 필요한 정보와 자격 증명이 포함된 플랫폼 서비스 엔드포인트를 생성해야 합니다.

각 플랫폼 서비스 유형에는 고유한 엔드포인트가 필요하므로 사용하려는 각 플랫폼 서비스에 대해 최소 하나 이상의 엔드포인트를 구성해야 합니다. 플랫폼 서비스 엔드포인트를 정의한 후에는 해당 엔드포인트의 URN을 서비스 활성화에 사용되는 구성 XML의 대상으로 사용합니다.

여러 소스 버킷에 대해 동일한 엔드포인트를 대상으로 사용할 수 있습니다. 예를 들어, 여러 소스 버킷에서 객체 메타데이터를 동일한 검색 통합 엔드포인트로 전송하도록 구성하여 여러 버킷에서 검색을 수행할 수 있습니다. 또한 소스 버킷이 여러 엔드포인트를 대상으로 사용하도록 구성할 수 있으므로 객체 생성 알림은 하나의 Amazon Simple Notification Service(Amazon SNS) 주제로, 객체 삭제 알림은 다른 Amazon SNS 주제로 전송하는 등의 작업을 수행할 수 있습니다.

CloudMirror 복제용 엔드포인트

StorageGRID는 S3 버킷을 나타내는 복제 엔드포인트를 지원합니다. 이러한 버킷은 Amazon Web Services, 동일하거나 원격의 StorageGRID 배포 환경, 또는 다른 서비스에 호스팅될 수 있습니다.

알림 엔드포인트

StorageGRID는 Amazon SNS, Kafka 및 웹훅 엔드포인트를 지원합니다. Simple Queue Service(SQS) 및 AWS Lambda 엔드포인트는 지원하지 않습니다.

Kafka 엔드포인트의 경우 상호 TLS는 지원되지 않습니다. 따라서 Kafka 브로커 구성에서 `ssl.client.auth`을(를) `required`로 설정한 경우 Kafka 엔드포인트 구성 문제가 발생할 수 있습니다.

검색 통합 서비스의 엔드포인트

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

검색 통합 엔드포인트는 특정 Elasticsearch 인덱스 및 유형을 참조합니다. StorageGRID에서 엔드포인트를 생성하기 전에 Elasticsearch에서 인덱스를 생성해야 합니다. 그렇지 않으면 엔드포인트 생성이 실패합니다. 유형은 엔드포인트를 생성하기 전에 생성할 필요가 없습니다. StorageGRID는 객체 메타데이터를 엔드포인트로 전송할 때 필요한 경우 유형을 생성합니다.

관련 정보

"StorageGRID 관리"

StorageGRID 플랫폼 서비스 엔드포인트에 대한 **URN**을 지정하십시오

플랫폼 서비스 엔드포인트를 생성할 때는 고유 리소스 이름(URN)을 지정해야 합니다. 이 URN은 플랫폼 서비스의 구성 XML을 생성할 때 엔드포인트를 참조하는 데 사용됩니다. 각 엔드포인트의 URN은 고유해야 합니다.

StorageGRID는 플랫폼 서비스 엔드포인트를 생성할 때 유효성을 검사합니다. 플랫폼 서비스 엔드포인트를 생성하기 전에 엔드포인트에 지정된 리소스가 존재하고 접근 가능한지 확인하십시오.

URN 요소

플랫폼 서비스 엔드포인트의 URN은 다음과 같이 `arn:aws` 또는 `urn:mysite`로 시작해야 합니다.

- 서비스가 Amazon Web Services(AWS)에서 호스팅되는 경우 다음을 사용하십시오 `arn:aws`
- 서비스가 Google Cloud Platform(GCP)에서 호스팅되는 경우 다음을 사용하세요. `arn:aws`
- 서비스가 로컬에서 호스팅되는 경우 다음을 사용하십시오 `urn:mysite`

예를 들어 StorageGRID에서 호스팅되는 CloudMirror 엔드포인트의 URN을 지정하는 경우 URN은 `urn:sgws`으로 시작할 수 있습니다.

URN의 다음 요소는 다음과 같이 플랫폼 서비스 유형을 지정합니다.

서비스	유형
CloudMirror 복제	`s3`
알림	`sns`, `kafka` 또는 `webhook`
검색 통합	`es`

예를 들어 StorageGRID에서 호스팅되는 CloudMirror 엔드포인트의 URN을 계속 지정하려면 `s3`을 추가하여 `urn:sgws:s3`을 얻습니다.

대부분의 엔드포인트에서 URN의 마지막 요소는 대상 URI의 특정 대상 리소스를 식별합니다(예: `sns-topic-name`).

웹훅 엔드포인트의 경우 대상 리소스는 대상 URI 자체입니다.

서비스	특정 리소스
CloudMirror 복제	`bucket-name`
알림	`sns-topic-name` 또는 `kafka-topic-name` 참고**:** 웹훅 엔드포인트의 경우, URN의 마지막 요소는 엔드포인트의 URN이 고유하기만 하면 어떤 문자열이든 될 수 있습니다.
검색 통합	`domain-name/index-name/type-name` 참고**:** Elasticsearch 클러스터가 인덱스를 자동으로 생성하도록 구성되어 있지 않은 경우, 엔드포인트를 생성하기 전에 인덱스를 수동으로 생성해야 합니다.

AWS 및 **GCP**에서 호스팅되는 서비스에 대한 **URN**

AWS 및 GCP 엔티티의 경우 전체 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-name`에는 여기에 표시된 것처럼 리터럴 문자열 `domain/`이 포함되어야 합니다.

로컬에서 호스팅되는 서비스에 대한 **URN**

클라우드 서비스 대신 로컬 호스팅 서비스를 사용하는 경우, 세 번째와 마지막 위치에 필수 요소가 포함되어 있는 한, 유효하고 고유한 URN을 생성하는 어떤 방식으로든 URN을 지정할 수 있습니다. 선택 사항으로 표시된 요소는 비워두거나, 리소스를 식별하고 URN을 고유하게 만드는 데 도움이 되는 방식으로 지정할 수 있습니다. 예를 들면 다음과 같습니다.

- CloudMirror 복제:

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

StorageGRID에서 호스팅되는 CloudMirror 엔드포인트의 경우, 다음으로 시작하는 유효한 URN을 지정할 수 있습니다 `urn:sgws`:

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

- 알림:

Amazon Simple Notification Service 엔드포인트를 지정하십시오.

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

Kafka 엔드포인트를 지정합니다:

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

웹훅 엔드포인트를 지정합니다.

```
urn:mysite:webhook:optional:optional:webhook-name
```

- 검색 통합:

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

> 로컬에서 호스팅되는 검색 통합 엔드포인트의 경우, 해당 `domain-name` 요소는 엔드포인트의 URN이 고유한 한 어떤 문자열이든 될 수 있습니다.

StorageGRID 플랫폼 서비스 엔드포인트를 생성합니다.

플랫폼 서비스를 활성화하려면 먼저 올바른 유형의 엔드포인트를 하나 이상 생성해야 합니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 플랫폼 서비스는 StorageGRID 관리자가 테넌트 계정에 대해 활성화했습니다.
- 귀하는 "엔드포인트 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다.
- ${post_edited_translations.segment}
  - CloudMirror 복제: S3 버킷
  - 이벤트 알림: Amazon Simple Notification Service(Amazon SNS) 토픽, Kafka 토픽 또는 웹훅 엔드포인트
  - 검색 알림: 타겟 클러스터가 인덱스를 자동으로 생성하도록 구성되어 있지 않은 경우 Elasticsearch 인덱스에 대한 알림이 표시됩니다.
- 대상 리소스에 대한 정보를 가지고 있습니다.
  - URI(Uniform Resource Identifier)의 호스트 및 포트

    > StorageGRID 시스템에서 호스팅되는 버킷을 CloudMirror 복제의 엔드포인트로 사용하려는 경우, 입력해야 할 값을 확인하려면 그리드 관리자에게 문의하십시오.

  - 고유 리소스 이름(URN)

    "플랫폼 서비스 엔드포인트에 대한 URN을 지정하십시오."

  - 인증 자격 증명(필요한 경우):

검색 통합 엔드포인트

검색 통합 엔드포인트의 경우 다음 자격 증명을 사용할 수 있습니다.

- 액세스 키: 액세스 키 ID 및 비밀 액세스 키
- 기본 HTTP: 사용자 이름과 비밀번호

CloudMirror 복제 엔드포인트

CloudMirror 복제 엔드포인트의 경우 다음 자격 증명을 사용할 수 있습니다.

- 액세스 키: 액세스 키 ID 및 비밀 액세스 키
- ${post_edited_translations.segment}

Amazon SNS 엔드포인트

Amazon SNS 엔드포인트의 경우 다음 자격 증명을 사용할 수 있습니다.

- 액세스 키: 액세스 키 ID 및 비밀 액세스 키

Kafka 엔드포인트

${post_edited_translations.segment}

- SASL/PLAIN: 사용자 이름 및 비밀번호
- SASL/SCRAM-SHA-256: 사용자 이름 및 비밀번호
- ${post_edited_translations.segment}

◦ 보안 인증서(인증서 확인이 필요한 경우)

- Elasticsearch 보안 기능이 활성화된 경우 연결 테스트를 위한 클러스터 모니터링 권한과 문서 업데이트를 위한 쓰기 인덱스 권한 또는 인덱스 및 인덱스 삭제 권한 모두가 필요합니다.

단계

1. **STORAGE (S3)** > *Platform services endpoints*를 선택합니다. Platform services endpoints 페이지가 나타납니다.
2. *엔드포인트 생성*을 선택합니다.
3. 엔드포인트와 그 용도를 간략하게 설명하는 표시 이름을 입력하십시오.

   ${post_edited_translations.segment}

4. ${post_edited_translations.segment}

   ${post_edited_translations.segment}

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

   포트를 지정하지 않으면 다음 기본 포트가 사용됩니다.

- HTTPS URI의 경우 포트 443, HTTP URI의 경우 포트 80(대부분의 엔드포인트)
- HTTPS 및 HTTP URI용 포트 9092 (Kafka 엔드포인트에만 해당)

예를 들어 StorageGRID에서 호스팅되는 버킷의 URI는 다음과 같을 수 있습니다.

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

이 예에서 `s3.example.com`은 StorageGRID 고가용성(HA) 그룹의 가상 IP(VIP)에 대한 DNS 엔트리를 나타내며, `10443`은 로드 밸런서 엔드포인트에 정의된 포트를 나타냅니다.

> 가능한 경우 단일 장애 지점을 방지하기 위해 로드 밸런싱 노드의 HA 그룹에 연결해야 합니다.

마찬가지로 AWS에서 호스팅되는 버킷의 URI는 다음과 같을 수 있습니다.

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

> 엔드포인트가 CloudMirror 복제 서비스에 사용되는 경우, URI에 버킷 이름을 포함하지 마십시오. 버킷 이름은 **URN** 필드에 포함합니다.

5. 엔드포인트의 고유 리소스 이름(URN)을 입력하십시오.

> 엔드포인트가 생성된 후에는 해당 엔드포인트의 URN을 변경할 수 없습니다.

6. *Continue*를 선택합니다.
7. *인증 유형*에 대한 값을 선택하십시오.

> 웹훅 엔드포인트에 대한 인증을 사용하려면 9단계에서 mTLS(상호 전송 계층 보안)를 구성하십시오.

검색 통합 엔드포인트

검색 통합 엔드포인트에 대한 자격 증명을 입력하거나 업로드하십시오.

제공하는 자격 증명에는 대상 리소스에 대한 쓰기 권한이 있어야 합니다.

${post_edited_translations.segment}	설명	**${post_edited_translations.segment}**
익명의	대상에 대한 익명 액세스를 제공합니다. 보안이 비활성화된 엔드포인트에만 작동합니다.	인증이 필요하지 않습니다.
액세스 키	대상과의 연결을 인증하기 위해 AWS 스타일 자격 증명을 사용합니다.	• 액세스 키 ID • ${post_edited_translations.segment}
기본 HTTP	${post_edited_translations.segment}	• 사용자 이름 • 비밀번호

CloudMirror 복제 엔드포인트

CloudMirror 복제 엔드포인트에 대한 자격 증명을 입력하거나 업로드하세요.

제공하는 자격 증명에는 대상 리소스에 대한 쓰기 권한이 있어야 합니다.

${post_edited_translations.segment}	설명	**${post_edited_translations.segment}**
익명의	대상에 대한 익명 액세스를 제공합니다. 보안이 비활성화된 엔드포인트에만 작동합니다.	인증이 필요하지 않습니다.
액세스 키	대상과의 연결을 인증하기 위해 AWS 스타일 자격 증명을 사용합니다.	• 액세스 키 ID • ${post_edited_translations.segment}

${post_edited_translations.segment}	설명	**${post_edited_translations.segment}**
${post_edited_translations.segment}	대상과의 연결을 인증하기 위해 인증서와 키를 사용합니다.	• 임시 자격 증명 URL • ${post_edited_translations.segment} • 클라이언트 인증서(PEM 파일 업로드) • 클라이언트 개인 키(PEM 파일 업로드, OpenSSL 암호화 형식 또는 암호화되지 않은 개인 키 형식) • 클라이언트 개인 키 암호(선택 사항)

Amazon SNS 엔드포인트

Amazon SNS 엔드포인트에 대한 자격 증명을 입력하거나 업로드하십시오.

제공하는 자격 증명에는 대상 리소스에 대한 쓰기 권한이 있어야 합니다.

${post_edited_translations.segment}	설명	**${post_edited_translations.segment}**
익명의	대상에 대한 익명 액세스를 제공합니다. 보안이 비활성화된 엔드포인트에만 작동합니다.	인증이 필요하지 않습니다.
액세스 키	대상과의 연결을 인증하기 위해 AWS 스타일 자격 증명을 사용합니다.	• 액세스 키 ID • ${post_edited_translations.segment}

Kafka 엔드포인트

Kafka 엔드포인트에 대한 자격 증명을 입력하거나 업로드합니다.

제공하는 자격 증명에는 대상 리소스에 대한 쓰기 권한이 있어야 합니다.

${post_edited_translations.segment}	설명	**${post_edited_translations.segment}**
익명의	대상에 대한 익명 액세스를 제공합니다. 보안이 비활성화된 엔드포인트에만 작동합니다.	인증이 필요하지 않습니다.

${post_edited_translations.segment}	설명	**${post_edited_translations.segment}**
SASL/PLAIN	${post_edited_translations.segment}	• 사용자 이름 • 비밀번호
SASL/SCRAM-SHA-256	${post_edited_translations.segment}	• 사용자 이름 • 비밀번호
SASL/SCRAM-SHA-512	챌린지-응답 프로토콜과 SHA-512 해싱을 사용하는 사용자 이름과 암호를 사용하여 대상에 대한 연결을 인증합니다.	• 사용자 이름 • 비밀번호

사용자 이름과 비밀번호가 Kafka 클러스터에서 얻은 위임 토큰에서 파생된 경우 *위임된 토큰 인증 사용*을 선택하십시오.

8. *Continue*를 선택합니다.
9. 인증서 확인 라디오 버튼을 선택하여 엔드포인트에 대한 TLS 연결을 확인하는 방법을 선택하십시오.

대부분의 엔드포인트

검색 통합, CloudMirror 복제, Amazon SNS 또는 Kafka 엔드포인트에 대한 TLS 연결을 확인하십시오.

인증서 검증 유형	설명
TLS	엔드포인트 리소스에 대한 TLS 연결에 사용되는 서버 인증서를 검증합니다.
비활성화됨	인증서 확인 기능이 비활성화되었습니다. 이 옵션은 안전하지 않습니다.
사용자 지정 CA 인증서 사용	사용자 지정 CA 인증서는 엔드포인트에 연결할 때 서버의 신원을 확인하는 데 사용됩니다.
운영 체제 CA 인증서 사용	${post_edited_translations.segment}

웹훅 엔드포인트만 해당

웹훅 엔드포인트에 대한 TLS 연결을 확인하십시오.

인증서 검증 유형	설명
TLS	엔드포인트 리소스에 대한 TLS 연결에 사용되는 서버 인증서를 검증합니다.
mTLS	엔드포인트 리소스에 대한 상호 TLS 연결에 사용되는 클라이언트 및 서버 인증서를 검증합니다.
비활성화됨	인증서 확인 기능이 비활성화되었습니다. 이 옵션은 안전하지 않습니다.
사용자 지정 CA 인증서 사용	사용자 지정 CA 인증서는 엔드포인트에 연결할 때 서버의 신원을 확인하는 데 사용됩니다.

${post_edited_translations.segment}

인증서 검증 유형	설명
서버 인증서를 확인하지 않음	서버 인증서 검증을 비활성화합니다. 즉, 서버의 신원이 확인되지 않습니다. 이 옵션은 보안에 취약합니다.
클라이언트 인증서	클라이언트 인증서는 엔드포인트에 연결할 때 클라이언트의 신원을 확인하는 데 사용됩니다.
클라이언트 개인 키	클라이언트 인증서의 개인 키입니다. 암호화된 경우 기존 PKCS #1 형식을 사용해야 합니다(PKCS #8 형식은 지원되지 않습니다).

인증서 검증 유형	설명
클라이언트 개인 키 암호	클라이언트 개인 키를 복호화하는 데 사용할 암호입니다. 개인 키가 암호화되지 않은 경우 이 필드를 비워 두십시오.

10. *테스트 및 엔드포인트 생성*을 선택합니다.
    - 지정된 자격 증명을 사용하여 엔드포인트에 연결할 수 있으면 성공 메시지가 표시됩니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.
    - 엔드포인트 유효성 검사에 실패하면 오류 메시지가 표시됩니다. 오류를 수정하기 위해 엔드포인트를 수정해야 하는 경우, *엔드포인트 세부 정보로 돌아가기*를 선택하고 정보를 업데이트합니다. 그런 다음 *엔드포인트 테스트 및 생성*을 선택합니다.

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

엔드포인트를 구성한 후에는 해당 URN을 사용하여 플랫폼 서비스를 구성할 수 있습니다.

관련 정보

- "플랫폼 서비스 엔드포인트에 대한 URN을 지정하십시오."
- "CloudMirror 복제 구성"
- "이벤트 알림 구성"
- "검색 통합 서비스 구성"

StorageGRID 플랫폼 서비스 엔드포인트의 연결을 테스트합니다

플랫폼 서비스에 대한 연결이 변경된 경우 엔드포인트에 대한 연결을 테스트하여 대상 리소스가 존재하고 지정한 자격 증명을 사용하여 접근할 수 있는지 확인할 수 있습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "엔드포인트 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다.

이 작업 정보

StorageGRID는 자격 증명에 올바른 권한이 있는지 검증하지 않습니다.

단계

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

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

2. 연결을 테스트할 엔드포인트를 선택합니다.

   엔드포인트 세부 정보 페이지가 나타납니다.

3. *연결 테스트*를 선택합니다.

◦ 지정된 자격 증명을 사용하여 엔드포인트에 연결할 수 있으면 성공 메시지가 표시됩니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.

◦ 엔드포인트 유효성 검사에 실패하면 오류 메시지가 표시됩니다. 오류를 수정하기 위해 엔드포인트를 수정해야 하는 경우, *구성*을 선택하고 정보를 업데이트합니다. 그런 다음 *테스트 및 변경 사항 저장*을 선택합니다.

StorageGRID에서 플랫폼 서비스 엔드포인트 편집

플랫폼 서비스 엔드포인트의 구성을 편집하여 이름, URI 또는 기타 세부 정보를 변경할 수 있습니다. 예를 들어 만료된 자격 증명을 업데이트하거나 장애 조치를 위해 백업 Elasticsearch 인덱스를 가리키도록 URI를 변경해야 할 수 있습니다. 플랫폼 서비스 엔드포인트의 URN은 변경할 수 없습니다.

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "엔드포인트 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다.

단계

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

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

2. 편집할 엔드포인트를 선택합니다.

   엔드포인트 세부 정보 페이지가 나타납니다.

3. *구성*을 선택합니다.
4. 필요에 따라 엔드포인트의 구성을 변경하십시오.

   엔드포인트가 생성된 후에는 해당 엔드포인트의 URN을 변경할 수 없습니다.

   a. 엔드포인트의 표시 이름을 변경하려면 편집 아이콘 을 선택합니다.
   b. 필요에 따라 URI를 변경하십시오.
   c. 필요에 따라 인증 유형을 변경하십시오.
      - 액세스 키 인증의 경우, *S3 키 편집*을 선택하고 새 액세스 키 ID와 비밀 액세스 키를 붙여넣어 필요에 따라 키를 변경하십시오. 변경 사항을 취소하려면 *S3 키 편집 되돌리기*를 선택하십시오.
      - CAP(C2S 액세스 포털) 인증의 경우, 필요에 따라 임시 자격 증명 URL 또는 선택적 클라이언트 개인 키 암호를 변경하고 새 인증서 및 키 파일을 업로드하십시오.

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

   d. 필요에 따라 인증서 확인 방법을 변경하십시오.
5. *테스트 후 변경 사항 저장*을 선택합니다.

   ◦ 지정된 자격 증명을 사용하여 엔드포인트에 연결할 수 있으면 성공 메시지가 표시됩니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.

◦ 엔드포인트 유효성 검사에 실패하면 오류 메시지가 표시됩니다. 오류를 수정하려면 엔드포인트를 수정하고 *테스트 및 변경 사항 저장*을 선택하십시오.

StorageGRID 플랫폼 서비스 엔드포인트를 삭제합니다

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

시작하기 전에

- 현재 "지원되는 웹 브라우저"을(를) 사용하여 테넌트 관리자에 로그인했습니다.
- 귀하는 "엔드포인트 또는 루트 액세스 권한 관리"이(가) 있는 사용자 그룹에 속해 있습니다.

단계

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

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

2. 삭제하려는 각 엔드포인트의 확인란을 선택합니다.

   사용 중인 플랫폼 서비스 엔드포인트를 삭제하면 해당 엔드포인트를 사용하는 모든 버킷에 대해 연결된 플랫폼 서비스가 비활성화됩니다. 아직 완료되지 않은 요청은 모두 삭제됩니다. 삭제된 URN을 더 이상 참조하지 않도록 버킷 구성을 변경하기 전까지는 새로운 요청이 계속 생성됩니다. StorageGRID는 이러한 요청을 복구할 수 없는 오류로 보고합니다.

3. 작업 > *엔드포인트 삭제*를 선택합니다.

   확인 메시지가 나타납니다.

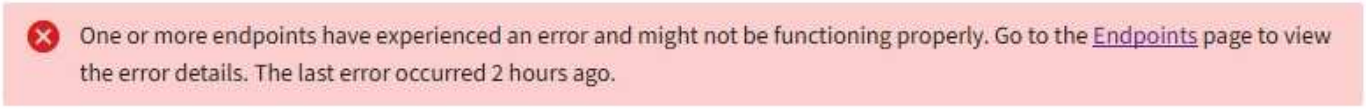
4. *엔드포인트 삭제*를 선택합니다.

StorageGRID 플랫폼 서비스 엔드포인트 오류 해결

StorageGRID가 플랫폼 서비스 엔드포인트와 통신하려고 할 때 오류가 발생하면 대시보드에 메시지가 표시됩니다. 플랫폼 서비스 엔드포인트 페이지의 마지막 오류 열에는 오류가 발생한 시간이 표시됩니다. 엔드포인트 자격 증명과 연결된 권한이 올바르지 않은 경우에는 오류가 표시되지 않습니다.

오류가 발생했는지 확인합니다.

지난 7일 동안 플랫폼 서비스 엔드포인트 오류가 발생한 경우 테넌트 관리자 대시보드에 알림 메시지가 표시됩니다. 플랫폼 서비스 엔드포인트 페이지에서 오류에 대한 자세한 내용을 확인할 수 있습니다.



대시보드에 표시되는 것과 동일한 오류가 플랫폼 서비스 엔드포인트 페이지 상단에도 나타납니다. 자세한 오류 메시지를 보려면 다음을 참조하십시오.

단계

1. 엔드포인트 목록에서 오류가 발생한 엔드포인트를 선택합니다.
2. 엔드포인트 세부 정보 페이지에서 *연결*을 선택합니다. 이 탭에는 엔드포인트에 대한 가장 최근 오류만 표시되며 오류 발생 시점이 표시됩니다. 빨간색 X 아이콘 이 포함된 오류는 지난 7일 이내에 발생한 오류입니다.

오류가 여전히 유효한지 확인하십시오.

일부 오류는 해결된 후에도 마지막 오류 열에 계속 표시될 수 있습니다. 오류가 현재 발생 중인지 확인하거나 해결된 오류를 표에서 강제로 제거하려면 다음 단계를 따르세요.

단계

1. 엔드포인트를 선택합니다.

   엔드포인트 세부 정보 페이지가 나타납니다.

2. * 연결* > *연결 테스트*를 선택합니다.

*연결 테스트*를 선택하면 StorageGRID가 플랫폼 서비스 엔드포인트가 존재하고 현재 자격 증명으로 연결할 수 있는지 확인합니다. 엔드포인트에 대한 연결은 각 사이트의 한 노드에서 검증됩니다.

엔드포인트 오류 해결

엔드포인트 세부 정보 페이지에서 마지막 오류 메시지를 확인하여 오류 원인을 파악할 수 있습니다. 일부 오류는 엔드포인트를 수정해야 해결할 수 있습니다. 예를 들어, StorageGRID가 올바른 액세스 권한이 없거나 액세스 키가 만료되어 대상 S3 버킷에 액세스할 수 없는 경우 CloudMirroring 오류가 발생할 수 있습니다. 이 경우 메시지는 "엔드포인트 자격 증명 또는 대상 액세스를 업데이트해야 합니다."이며, 세부 정보는 "AccessDenied" 또는 "InvalidAccessKeyId"로 표시됩니다.

오류를 해결하기 위해 엔드포인트를 수정해야 하는 경우, *테스트 및 변경 사항 저장*을 선택하면 StorageGRID가 업데이트된 엔드포인트를 검증하고 현재 자격 증명으로 연결할 수 있는지 확인합니다. 엔드포인트 연결은 각 사이트의 한 노드에서 검증됩니다.

단계

1. 엔드포인트를 선택합니다.
2. 엔드포인트 세부 정보 페이지에서 *구성*을 선택합니다.
3. 필요에 따라 엔드포인트 구성을 편집하십시오.
4. * 연결* > *연결 테스트*를 선택합니다.

${post_edited_translations.segment}

StorageGRID 플랫폼 서비스 엔드포인트를 검증할 때, 해당 엔드포인트의 자격 증명을 사용하여 대상 리소스에 연결할 수 있는지 확인하고 기본적인 권한 검사를 수행합니다. 그러나 StorageGRID는 특정 플랫폼 서비스 작업에 필요한 모든 권한을 검증하지는 않습니다. 따라서 플랫폼 서비스를 사용하려고 할 때 오류(예: "403 Forbidden")가 발생하는 경우, 엔드포인트 자격 증명과 연결된 권한을 확인하십시오.

관련 정보

- **StorageGRID** 관리 › 플랫폼 서비스 문제 해결
- "플랫폼 서비스 엔드포인트 생성"
- "플랫폼 서비스 엔드포인트에 대한 연결 테스트"

- "플랫폼 서비스 엔드포인트 편집"

StorageGRID에서 **CloudMirror** 복제 구성

버킷에 대한 CloudMirror 복제를 활성화하려면 유효한 버킷 복제 구성 XML을 생성하여 적용해야 합니다.

시작하기 전에

- 플랫폼 서비스는 StorageGRID 관리자가 테넌트 계정에 대해 활성화했습니다.
- 이미 복제 소스 역할을 할 버킷을 생성했습니다.
- CloudMirror 복제 대상으로 사용하려는 엔드포인트가 이미 존재하며 해당 URN을 알고 있습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"을(를) 가진 사용자 그룹에 속해 있습니다. 이러한 권한은 테넌트 관리자를 사용하여 버킷을 구성할 때 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업 정보

CloudMirror 복제는 소스 버킷의 객체를 엔드포인트에 지정된 대상 버킷으로 복사합니다.

버킷 복제 및 구성 방법에 대한 일반 정보는 "Amazon Simple Storage Service (S3) 문서: 객체 복제"을 참조하십시오. StorageGRID가 GetBucketReplication, DeleteBucketReplication 및 PutBucketReplication을 구현하는 방법에 대한 자세한 내용은 "버킷에 대한 작업"을 참조하십시오.

> CloudMirror 복제는 크로스 그리드 복제 기능과 중요한 유사점과 차이점을 가지고 있습니다. 자세한 내용은 "크로스 그리드 복제와 CloudMirror 복제 비교"을 참조하십시오.

CloudMirror 복제를 구성할 때 다음 요구 사항 및 특징에 유의하십시오.

- 유효한 버킷 복제 구성 XML을 생성하고 적용할 때는 각 대상에 대해 S3 버킷 엔드포인트의 URN을 사용해야 합니다.
- S3 객체 잠금이 활성화된 소스 또는 대상 버킷의 경우 복제가 지원되지 않습니다.
- CloudMirror 복제를 객체가 포함된 버킷에서 활성화하면 버킷에 새로 추가된 객체는 복제되지만, 버킷에 이미 있는 객체는 복제되지 않습니다. 복제를 시작하려면 기존 객체를 업데이트해야 합니다.
- 복제 구성 XML에서 스토리지 클래스를 지정하면 StorageGRID는 대상 S3 엔드포인트에 대한 작업을 수행할 때 해당 클래스를 사용합니다. 대상 엔드포인트 또한 지정된 스토리지 클래스를 지원해야 합니다. 대상 시스템 공급업체에서 제공하는 권장 사항을 반드시 따르십시오.

단계

1. 원본 버킷에 대한 복제를 활성화합니다.
   - S3 복제 API에 명시된 대로 복제를 활성화하는 데 필요한 복제 구성 XML을 텍스트 편집기를 사용하여 생성합니다.
   - XML을 구성할 때:
     - StorageGRID는 복제 구성의 V1만 지원합니다. 즉, StorageGRID는 규칙에 대한 `Filter`요소 사용을 지원하지 않으며, 객체 버전 삭제 시 V1 규칙을 따릅니다. 자세한 내용은 Amazon의 복제 구성 관련 문서를 참조하십시오.
     - 대상으로 S3 버킷 엔드포인트의 URN을 사용합니다.
     - 선택적으로 `<StorageClass>` 요소를 추가하고 다음 중 하나를 지정하십시오.

- `STANDARD`: 기본 스토리지 클래스입니다. 객체를 업로드할 때 스토리지 클래스를 지정하지 않으면 `STANDARD` 스토리지 클래스가 사용됩니다.
- `STANDARD_IA`: (Standard - infrequent access.) 액세스 빈도는 낮지만 필요할 때 신속한 액세스가 필요한 데이터에 이 스토리지 클래스를 사용하십시오.
- `REDUCED_REDUNDANCY`: 이 스토리지 클래스는 중요하지 않고 재현 가능한 데이터, 즉 `STANDARD` 스토리지 클래스보다 중복성을 줄여도 되는 데이터에 사용하십시오.

- 구성 XML에 `Role`을 지정하더라도 해당 값은 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. 대시보드에서 버킷 보기*를 선택하거나 *스토리지**(S3)** > *버킷*을 선택합니다.
3. 소스 버킷의 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

4. 플랫폼 서비스 > *복제*를 선택합니다.
5. 복제 활성화 확인란을 선택하십시오.
6. 복제 구성 XML을 텍스트 상자에 붙여넣고 *변경 내용 저장*을 선택합니다.

   각 테넌트 계정에 대해 플랫폼 서비스를 활성화하려면 Grid Manager 또는 Grid Management API를 사용하는 StorageGRID 관리자가 필요합니다. 구성 XML을 저장할 때 오류가 발생하면 StorageGRID 관리자에게 문의하십시오.

7. 복제 구성이 올바르게 되어 있는지 확인하십시오.
   a. 복제 구성에 지정된 복제 요구 사항을 충족하는 객체를 소스 버킷에 추가합니다.

      앞서 보여드린 예시에서처럼 접두사 "2020"과 일치하는 객체가 복제됩니다.

   b. 객체가 대상 버킷에 복제되었는지 확인하십시오.

      작은 오브젝트의 경우 복제가 빠르게 이루어집니다.

관련 정보

"플랫폼 서비스 엔드포인트 생성"

버킷에 대한 알림을 활성화하려면 알림 구성 XML을 생성하고 Tenant Manager를 사용하여 해당 XML을 버킷에 적용하면 됩니다.

시작하기 전에

- 플랫폼 서비스는 StorageGRID 관리자가 테넌트 계정에 대해 활성화했습니다.
- 이미 알림 소스로 사용할 버킷을 생성했습니다.
- 이벤트 알림 대상으로 사용하려는 엔드포인트가 이미 존재하며, 해당 URN을 알고 있습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"을(를) 가진 사용자 그룹에 속해 있습니다. 이러한 권한은 테넌트 관리자를 사용하여 버킷을 구성할 때 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업 정보

이벤트 알림은 알림 구성 XML을 소스 버킷과 연결하여 구성합니다. 알림 구성 XML은 버킷 알림 구성을 위한 S3 규칙을 따르며, 대상 Amazon SNS 토픽, Kafka 토픽 또는 웹훅 엔드포인트는 엔드포인트의 URN으로 지정됩니다.

이벤트 알림 및 구성 방법에 대한 일반 정보는 "Amazon 문서"을 참조하십시오. StorageGRID가 S3 버킷 알림 구성 API를 구현하는 방법에 대한 자세한 내용은 "S3 클라이언트 애플리케이션 구현 지침"을 참조하십시오.

버킷에 대한 이벤트 알림을 구성할 때 다음 요구 사항 및 특징을 참고하십시오.

- 유효한 알림 구성 XML을 생성하고 적용할 때는 각 대상에 대해 이벤트 알림 엔드포인트의 URN을 사용해야 합니다.
- S3 Object Lock이 활성화된 버킷에서 이벤트 알림을 구성할 수 있지만, 알림 메시지에는 객체의 S3 Object Lock 메타데이터(보존 기한 및 법적 보존 상태 포함)가 포함되지 않습니다.
- 이벤트 알림을 구성하면 소스 버킷의 객체에 대해 지정된 이벤트가 발생할 때마다 알림이 생성되어 대상으로 사용된 Amazon SNS 토픽, Kafka 토픽 또는 웹훅 엔드포인트로 전송됩니다.
- 객체가 포함된 버킷에 대해 이벤트 알림을 활성화하면 알림 구성이 저장된 후에 수행된 작업에 대해서만 알림이 전송됩니다.

단계

1. 소스 버킷에 대한 알림을 활성화합니다.
    - 텍스트 편집기를 사용하여 S3 알림 API에 명시된 대로 이벤트 알림을 활성화하는 데 필요한 알림 구성 XML을 생성합니다.
    - XML을 구성할 때 대상 토픽으로 이벤트 알림 엔드포인트의 URN을 사용하십시오.

```
<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 **STORAGE (S3)** > *Buckets*를 선택합니다.
3. 소스 버킷의 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

4. 플랫폼 서비스 > *이벤트 알림*을 선택합니다.
5. 이벤트 알림 활성화 확인란을 선택합니다.
6. 알림 구성 XML을 텍스트 상자에 붙여넣고 *변경 사항 저장*을 선택합니다.

   각 테넌트 계정에 대해 플랫폼 서비스를 활성화하려면 Grid Manager 또는 Grid Management API를 사용하는 StorageGRID 관리자가 필요합니다. 구성 XML을 저장할 때 오류가 발생하면 StorageGRID 관리자에게 문의하십시오.

7. 이벤트 알림이 올바르게 구성되었는지 확인하십시오.

   a. 구성 XML에 설정된 대로 알림을 트리거하기 위한 요구 사항을 충족하는 소스 버킷의 객체에 대해 작업을 수행합니다.

      예시에서는 `images/` 접두사가 붙은 객체가 생성될 때마다 이벤트 알림이 전송됩니다.

   b. 알림이 대상 Amazon SNS 토픽, Kafka 토픽 또는 웹훅 엔드포인트로 전달되었는지 확인하십시오.

      예를 들어, 대상 주제가 Amazon SNS에 호스팅되어 있는 경우, 알림이 전달될 때 이메일을 보내도록 서비스를 구성할 수 있습니다.

```
{
   "Records":[
      {
         "eventVersion":"2.0",
         "eventSource":"sgws:s3",
         "eventTime":"2017-08-08T23:52:38Z",
         "eventName":"ObjectCreated:Put",
         "userIdentity":{
            "principalId":"11111111111111111"
         },
         "requestParameters":{
            "sourceIPAddress":"193.51.100.20"
         },
         "responseElements":{
            "x-amz-request-id":"122047343"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"Image-created",
            "bucket":{
               "name":"test1",
               "ownerIdentity":{
                  "principalId":"11111111111111111"
               },
               "arn":"arn:sgws:s3:::test1"
            },
            "object":{
               "key":"images/cat.jpg",
               "size":0,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
               "sequencer":"14D90402421461C7"
            }
         }
      }
   ]
}
```

+
대상 토픽에서 알림을 수신하면 StorageGRID 알림에 대한 소스 버킷 구성이 성공적으로 완료된 것입니다.

관련 정보

- "버킷에 대한 알림 이해"
- "S3 REST API 사용"
- "플랫폼 서비스 엔드포인트 생성"

버킷에 대한 검색 통합을 활성화하려면 검색 통합 XML을 생성하고 Tenant Manager를 사용하여 해당 XML을 버킷에 적용하면 됩니다.

시작하기 전에

- 플랫폼 서비스는 StorageGRID 관리자가 테넌트 계정에 대해 활성화했습니다.
- 이미 S3 버킷을 생성했으며, 해당 버킷의 콘텐츠를 인덱싱하려고 합니다.
- 검색 통합 서비스의 대상으로 사용하려는 엔드포인트가 이미 존재하며, 해당 URN을 가지고 있습니다.
- 귀하는 "모든 버킷 또는 루트 액세스 권한 관리"을(를) 가진 사용자 그룹에 속해 있습니다. 이러한 권한은 테넌트 관리자를 사용하여 버킷을 구성할 때 그룹 또는 버킷 정책의 권한 설정을 재정의합니다.

이 작업 정보

소스 버킷에 대한 검색 통합 서비스를 구성한 후 객체를 생성하거나 객체의 메타데이터 또는 태그를 업데이트하면 객체 메타데이터가 대상 엔드포인트로 전송됩니다.

이미 객체가 포함된 버킷에 대해 검색 통합 서비스를 활성화하면 기존 객체에 대한 메타데이터 알림이 자동으로 전송되지 않습니다. 대상 검색 인덱스에 메타데이터가 추가되도록 하려면 기존 객체를 업데이트해야 합니다.

단계

1. 버킷에 대한 검색 통합을 활성화합니다.

   - 메타데이터 검색 통합을 활성화하는 데 필요한 메타데이터 알림 XML을 생성하려면 텍스트 편집기를 사용하십시오.
   - XML을 구성할 때 대상으로는 검색 통합 엔드포인트의 URN을 사용하십시오.

     객체는 객체 이름의 접두사를 기준으로 필터링할 수 있습니다. 예를 들어, 접두사가 `images`인 객체의 메타데이터는 한 대상으로 보내고, 접두사가 `videos`인 객체의 메타데이터는 다른 대상으로 보낼 수 있습니다. 접두사가 중복되는 구성은 유효하지 않으며 제출 시 거부됩니다. 예를 들어, 접두사가 `test`인 객체에 대한 규칙 하나와 접두사가 `test2`인 객체에 대한 두 번째 규칙이 포함된 구성은 허용되지 않습니다.

     필요에 따라 메타데이터 구성 XML의 예시를 참조하십시오.

     ```
     <MetadataNotificationConfiguration>
      <Rule>
         <Status>Enabled</Status>
         <Prefix></Prefix>
         <Destination>
            <Urn>/Urn>
            </Destination>
      </Rule>
     </MetadataNotificationConfiguration>
     ```

     메타데이터 알림 구성 XML의 요소:

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

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

2. 테넌트 관리자에서 **STORAGE (S3)** > *Buckets*를 선택합니다.
3. 소스 버킷의 이름을 선택합니다.

   버킷 상세 정보 페이지가 나타납니다.

4. 플랫폼 서비스 * > * 검색 통합 * 을 선택합니다.
5. 검색 통합 활성화 확인란을 선택합니다.
6. 메타데이터 알림 구성을 텍스트 상자에 붙여넣고 *변경 사항 저장*을 선택합니다.

   각 테넌트 계정에 대한 플랫폼 서비스는 StorageGRID 관리자가 Grid Manager 또는 관리 API를 사용하여 활성화해야 합니다. 구성 XML을 저장할 때 오류가 발생하면 StorageGRID 관리자에게 문의하십시오.

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

      앞서 보여드린 예시에서처럼 버킷에 추가되는 모든 객체는 메타데이터 알림을 트리거합니다.

   b. 해당 객체의 메타데이터와 태그가 포함된 JSON 문서가 엔드포인트에 지정된 검색 인덱스에 추가되었는지 확인하십시오.

완료한 후

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

- **STORAGE (S3)** > **Buckets***를 선택하고 ***Enable search integration** 확인란을 선택 해제합니다.

- S3 API를 직접 사용하는 경우, DELETE Bucket 메타데이터 알림 요청을 사용하십시오. S3 클라이언트 애플리케이션 구현 지침을 참조하십시오.

예시: 모든 객체에 적용되는 메타데이터 알림 구성

이 예시에서는 모든 객체의 객체 메타데이터가 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>
```

예시: 두 개의 규칙이 포함된 메타데이터 알림 구성

이 예시에서는 접두사 `/images`와 일치하는 객체의 객체 메타데이터는 한 대상으로 전송되고, 접두사 `/videos`와 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>
```

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

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

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

JSON 문서에 포함된 필드

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

버킷 및 객체 정보

`bucket`: 버킷 이름

`key`: 객체 키 이름

`versionID`: 버전 관리 버킷에 있는 객체의 객체 버전입니다.

`region`: 버킷 영역, 예를 들어 `us-east-1`

시스템 메타데이터

`size`: HTTP 클라이언트에 표시되는 객체 크기(바이트)

`md5`: 객체 해시

사용자 메타데이터

`metadata`: 객체에 대한 모든 사용자 메타데이터(키-값 쌍 형식)

`key:value`

태그

`tags`: 키-값 쌍으로 나타낸 객체에 대해 정의된 모든 객체 태그

```
key:value
```

Elasticsearch에서 결과를 보는 방법

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

S3 REST API 사용

StorageGRID S3 REST API 지원 버전 및 업데이트

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

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

지원되는 버전

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

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

S3 REST API 지원 업데이트

릴리스	댓글
12.0	• 관리 인터페이스에 대한 CORS(Cross-Origin Resource Sharing)에 대한 지원이 추가되어 다른 도메인에서 관리 API를 사용하여 StorageGRID의 데이터에 액세스할 수 있습니다. "더 알아보기". • STS AssumeRole 및 세션 정책 지원이 추가되었습니다. "세션 정책의 예시"을 참조하십시오. 테넌트 그룹에서 AssumeRole을 설정할 수 있습니다.
11.9	• 다음 요청 및 지원되는 헤더에 대해 미리 계산된 SHA-256 체크섬 값에 대한 지원이 추가되었습니다. 이 기능을 사용하여 업로드된 객체의 무결성을 확인할 수 있습니다. ◦ CompleteMultipartUpload: `x-amz-checksum-sha256` ◦ CreateMultipartUpload: `x-amz-checksum-algorithm` ◦ GetObject: `x-amz-checksum-mode` ◦ HeadObject: `x-amz-checksum-mode` ◦ ListParts ◦ PutObject: `x-amz-checksum-sha256` ◦ UploadPart: `x-amz-checksum-sha256` • 그리드 관리자가 테넌트 수준의 보존 및 규정 준수 설정을 제어할 수 있는 기능을 추가했습니다. 이러한 설정은 S3 객체 잠금 설정에 영향을 미칩니다. ◦ 버킷 기본 보존 모드 및 객체 보존 모드: 그리드 관리자가 허용하는 경우 Governance 또는 Compliance. ◦ 버킷 기본 보존 기간 및 객체 보존 만료일: 그리드 관리자가 설정한 최대 보존 기간에서 허용하는 값보다 작거나 같아야 합니다. • 향상된 `aws-chunked` 콘텐츠 인코딩 및 스트리밍 `x-amz-content-sha256` 값 지원. 제한 사항: ◦ 존재하는 경우 `chunk-signature`은 선택 사항이며 유효성 검사를 거치지 않습니다. ◦ 존재하는 경우, `x-amz-trailer` 콘텐츠는 무시됩니다
11.8	S3 작업 이름을 "Amazon Web Services(AWS) 문서: Amazon Simple Storage Service API 참조"에 사용된 이름과 일치하도록 업데이트했습니다.
11.7	• 추가됨 "빠른 참조: 지원되는 S3 API 요청". • S3 객체 잠금과 함께 거버넌스 모드를 사용할 수 있도록 지원이 추가되었습니다. • GET Object 및 HEAD Object 요청에 대한 StorageGRID 관련 `x-ntap-sg-cgr-replication-status` 응답 헤더 지원이 추가되었습니다. 이 헤더는 그리드 간 복제를 위한 객체의 복제 상태를 제공합니다. • SelectObjectContent 요청에서 이제 Parquet 객체를 지원합니다.

릴리스	댓글
11.6	• GET Object 및 HEAD Object 요청에서 `partNumber` request 매개변수를 사용할 수 있도록 지원이 추가되었습니다. • S3 Object Lock에 대해 버킷 수준에서 기본 보존 모드 및 기본 보존 기간에 대한 지원이 추가되었습니다. • `s3:object-lock-remaining-retention-days` 정책 조건 키에 대한 지원이 추가되어 객체에 허용되는 보존 기간 범위를 설정할 수 있습니다. • 단일 PUT Object 작업의 최대 권장 크기를 5GiB(5,368,709,120바이트)로 변경했습니다. 5GiB보다 큰 객체가 있는 경우 멀티파트 업로드를 사용하십시오.
11.5	• 버킷 암호화 관리 지원이 추가되었습니다. • S3 객체 잠금 지원이 추가되었으며, 기존의 규정 준수 요청 기능은 더 이상 사용되지 않습니다. • 버전 관리 버킷에서 DELETE Multiple Objects 사용에 대한 지원이 추가되었습니다. • The `Content-MD5` 요청 헤더가 이제 올바르게 지원됩니다.
11.4	• DELETE Bucket tagging, GET Bucket tagging 및 PUT Bucket tagging에 대한 지원이 추가되었습니다. 비용 할당 태그는 지원되지 않습니다. • StorageGRID 11.4에서 생성된 버킷의 경우, 성능 모범 사례를 충족하기 위해 객체 키 이름을 제한할 필요가 더 이상 없습니다. • `s3:ObjectRestore:Post` 이벤트 유형에 대한 버킷 알림 지원이 추가되었습니다. • 이제 AWS 멀티파트 파트의 크기 제한이 적용됩니다. 멀티파트 업로드의 각 파트는 5MiB에서 5GiB 사이여야 합니다. 마지막 파트는 5MiB보다 작을 수 있습니다. • TLS 1.3 지원이 추가되었습니다.
11.3	• 고객이 제공하는 키를 사용한 서버 측 객체 데이터 암호화(SSE-C) 지원이 추가되었습니다. • 버킷 수명 주기 작업(만료 작업에만 해당) 및 `x-amz-expiration` 응답 헤더에 대한 DELETE, GET, PUT Bucket 작업 지원이 추가되었습니다. • PUT Object, PUT Object - Copy 및 Multipart Upload가 업데이트되어 수집 시 동기 배치를 사용하는 ILM 규칙의 영향을 설명합니다. • TLS 1.1 암호화 방식은 더 이상 지원되지 않습니다.
11.2	클라우드 스토리지 풀에서 사용할 수 있도록 POST 객체 복원 지원이 추가되었습니다. 그룹 및 버킷 정책에서 ARN, 정책 조건 키 및 정책 변수에 AWS 구문을 사용할 수 있도록 지원이 추가되었습니다. StorageGRID 구문을 사용하는 기존 그룹 및 버킷 정책은 계속 지원됩니다. 참고**:** 사용자 지정 StorageGRID 기능에 사용되는 것을 포함하여 다른 구성 JSON/XML에서 ARN/URN의 사용 방식은 변경되지 않았습니다.
11.1	CORS(교차 출처 리소스 공유) 지원, 그리드 노드에 대한 S3 클라이언트 연결용 HTTP 지원, 버킷의 규정 준수 설정 기능이 추가되었습니다.

릴리스	댓글
11.0	버킷에 대한 플랫폼 서비스(CloudMirror 복제, 알림 및 Elasticsearch 검색 통합) 구성 지원이 추가되었습니다. 또한 버킷에 대한 객체 태깅 위치 제약 조건 및 Available 일관성 지원도 추가되었습니다.
10.4	버전 관리, 엔드포인트 도메인 이름 페이지 업데이트, 정책의 조건 및 변수, 정책 예제, PutOverwriteObject 권한에 대한 ILM 스캔 변경 사항에 대한 지원이 추가되었습니다.
10.3	버전 관리 기능이 추가되었습니다.
10.2	그룹 및 버킷 액세스 정책과 멀티파트 복사(Upload Part - Copy)에 대한 지원이 추가되었습니다.
10.1	멀티파트 업로드, 가상 호스팅 방식 요청 및 v4 인증에 대한 지원이 추가되었습니다.
10.0	StorageGRID 시스템에서 S3 REST API를 처음으로 지원합니다. 현재 지원되는 *Simple Storage Service API Reference* 버전은 2006-03-01입니다.

StorageGRID에서 지원되는 S3 API 요청

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

이 페이지에는 StorageGRID에서 지원하는 S3 작업만 포함되어 있습니다.

> 각 작업에 대한 AWS 설명서를 보려면 제목의 링크를 선택하십시오.

일반적인 **URI** 쿼리 매개변수 및 요청 헤더

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

- `versionId` (객체 연산에 필요한 사항)

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

- `Authorization`
- `Connection`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Date`
- `Expect`
- `Host`

- `x-amz-date`

관련 정보

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

"AbortMultipartUpload"

URI 쿼리 매개변수 및 요청 헤더

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

- `uploadId`

요청 본문

None

StorageGRID 문서

"멀티파트 업로드 작업"

"CompleteMultipartUpload"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 문서

"CompleteMultipartUpload"

"CopyObject"

URI 쿼리 매개변수 및 요청 헤더

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

- `x-amz-copy-source`

- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-modified-since`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-server-side-encryption-customer-algorithm`
- `x-amz-copy-source-server-side-encryption-customer-key`
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`
- `x-amz-metadata-directive`
- `x-amz-object-lock-legal-hold`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-server-side-encryption`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-storage-class`
- `x-amz-tagging`
- `x-amz-tagging-directive`
- `x-amz-meta-<metadata-name>`

요청 본문

None

StorageGRID 문서

"CopyObject"

"CreateBucket"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문

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

StorageGRID 문서

"버킷에 대한 작업"

"CreateMultipartUpload"

URI 쿼리 매개변수 및 요청 헤더

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

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-algorithm`
- `x-amz-server-side-encryption`
- `x-amz-storage-class`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-tagging`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`
- `x-amz-meta-<metadata-name>`

요청 본문

None

StorageGRID 문서

"CreateMultipartUpload"

"DeleteBucket"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketCors"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketEncryption"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketLifecycle"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

- "버킷에 대한 작업"
- "S3 라이프사이클 구성 생성"

"DeleteBucketPolicy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketReplication"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteObject"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"DeleteObjects"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문

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

StorageGRID 문서

"객체에 대한 연산"

"DeleteObjectTagging"

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"GetBucketAcl"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketCors"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketEncryption"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketLifecycleConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

- "버킷에 대한 작업"
- "S3 라이프사이클 구성 생성"

"GetBucketLocation"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketNotificationConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketPolicy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketReplication"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketVersioning"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetObject"

URI 쿼리 매개변수 및 요청 헤더

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

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

그리고 다음과 같은 추가 요청 헤더가 있습니다.

- `Range`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`

요청 본문

None

StorageGRID 문서

"GetObject"

"GetObjectAcl"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"GetObjectLegalHold"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"GetObjectLockConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"GetObjectRetention"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"GetObjectTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"HeadBucket"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"HeadObject"

URI 쿼리 매개변수 및 요청 헤더

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

- `x-amz-checksum-mode`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`
- `Range`

요청 본문

None

StorageGRID 문서

"HeadObject"

"ListBuckets"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

서비스 › **ListBuckets**에 대한 작업

"ListMultipartUploads"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문

None

StorageGRID 문서

"ListMultipartUploads"

"ListObjects"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"ListObjectsV2"

URI 쿼리 매개변수 및 요청 헤더

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

- `continuation-token`
- `delimiter`

- `encoding-type`
- `fetch-owner`
- `max-keys`
- `prefix`
- `start-after`

요청 본문
None

StorageGRID 문서
"버킷에 대한 작업"

"ListObjectVersions"

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

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

요청 본문
None

StorageGRID 문서
"버킷에 대한 작업"

"ListParts"

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

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

요청 본문
None

StorageGRID 문서
"ListMultipartUploads"

"PutBucketCors"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketEncryption"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketLifecycleConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

- `And`
- `Days`
- `Expiration`
- `ExpiredObjectDeleteMarker`
- `Filter`
- `ID`
- `Key`
- `LifecycleConfiguration`
- `NewerNoncurrentVersions`

- NoncurrentDays
- NoncurrentVersionExpiration
- Prefix
- Rule
- Status
- Tag
- Value

StorageGRID 문서

- "버킷에 대한 작업"
- "S3 라이프사이클 구성 생성"

"PutBucketNotificationConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketPolicy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

"PutBucketReplication"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

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

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketVersioning"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 매개변수

StorageGRID는 다음과 같은 요청 본문 매개변수를 지원합니다.

- `VersioningConfiguration`
- `Status`

StorageGRID 문서

"버킷에 대한 작업"

"PutObject"

URI 쿼리 매개변수 및 요청 헤더

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

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-sha256`
- `x-amz-server-side-encryption`
- `x-amz-storage-class`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-tagging`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`
- `x-amz-meta-<metadata-name>`

요청 본문

- 객체의 이진 데이터

StorageGRID 문서

"PutObject"

"PutObjectLegalHold"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"PutObjectLockConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"PutObjectRetention"

URI 쿼리 매개변수 및 요청 헤더

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

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

요청 본문

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

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"PutObjectTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

StorageGRID 문서

"객체에 대한 연산"

"RestoreObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

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

"SelectObjectContent"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

지원되는 본문 필드에 대한 자세한 내용은 다음을 참조하십시오.

- "S3 Select 사용"
- "SelectObjectContent"

"UploadPart"

URI 쿼리 매개변수 및 요청 헤더

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

- `partNumber`
- `uploadId`

그리고 다음과 같은 추가 요청 헤더가 있습니다.

- `x-amz-checksum-sha256`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`

요청 본문

- 해당 부분의 이진 데이터

StorageGRID 문서

"UploadPart"

"UploadPartCopy"

URI 쿼리 매개변수 및 요청 헤더

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

- `partNumber`
- `uploadId`

그리고 다음과 같은 추가 요청 헤더가 있습니다.

- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-modified-since`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-range`
- `x-amz-server-side-encryption-customer-algorithm`

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

요청 본문

None

StorageGRID 문서

"UploadPartCopy"

StorageGRID S3 REST API 구성 테스트

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

시작하기 전에

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

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

단계

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

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

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

   버킷 생성이 성공적으로 완료되면 다음 예시와 같이 버킷의 위치가 반환됩니다.

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

3. 객체를 업로드합니다.

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
put-object --bucket testbucket --key s3.pdf --body C:\s3-
test\upload\s3.pdf
```

객체 업로드가 성공적으로 완료되면 객체 데이터의 해시값인 Etag가 반환됩니다.

4. 버킷의 내용을 나열하여 객체가 업로드되었는지 확인하십시오.

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

5. 개체를 삭제합니다.

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

6. 버킷을 삭제합니다.

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

StorageGRID의 S3 REST API 구현 방식

StorageGRID S3 REST API가 충돌하는 클라이언트 요청을 해결하는 방법

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

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

StorageGRID S3 REST API가 가용성과 일관성의 균형을 유지하는 방법

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

기본적으로 StorageGRID는 새로 생성된 객체에 대해 쓰기 후 읽기 일관성을 보장합니다. PUT 작업이 성공적으로

완료된 후 발생하는 모든 GET 요청은 새로 기록된 데이터를 읽을 수 있습니다. 기존 객체 덮어쓰기, 메타데이터 업데이트 및 삭제는 최종 일관성을 보장합니다.

객체 작업을 다른 일관성 수준으로 수행하려면 다음을 수행할 수 있습니다.

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

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

일관성 값

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

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

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

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

FabricPool 버킷에서는 지원되지 않습니다.

"새로운 쓰기 후 읽기" 및 "사용 가능" 일관성 사용

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

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

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

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

API 작업에 대한 일관성 지정

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

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

PutObject 및 GetObject 작업 모두에 동일한 일관성을 유지해야 합니다.

버킷에 대한 일관성 지정

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

버킷의 일관성을 설정할 때 다음 사항에 유의하십시오.

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

일관성과 **ILM** 규칙이 상호 작용하여 데이터 보호에 영향을 미치는 방식

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

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

다음 "수집 옵션"은(는) ILM 규칙에 사용할 수 있습니다.

이중 커밋

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

엄격한

ILM 규칙에 명시된 모든 사본이 생성되어야만 클라이언트에 성공이 반환됩니다.

균형 잡힌

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

일관성 규칙과 **ILM** 규칙이 상호 작용하는 방식의 예시

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

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

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

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

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

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

StorageGRID에서 객체 버전 관리를 사용하는 방법

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

StorageGRID 시스템은 대부분의 기능을 지원하는 버전 관리 기능을 구현하지만, 몇 가지 제한 사항이 있습니다.

StorageGRID는 각 객체에 대해 최대 10,000개의 버전을 지원합니다.

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

다단계 인증(MFA)을 사용하는 경우 삭제는 지원되지 않습니다.

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

ILM 및 버전 관리

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

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

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

관련 정보

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

S3 REST API를 사용하여 StorageGRID S3 Object Lock을 구성합니다

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

버킷에 S3 Object Lock을 활성화하는 방법

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

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

버킷에 S3 Object Lock을 활성화하려면 다음 방법 중 하나를 사용하십시오.

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

S3 객체 잠금에는 버킷 버전 관리가 필요하며, 이는 버킷 생성 시 자동으로 활성화됩니다. 버킷의 버전 관리를 일시

중지할 수 없습니다. "객체 버전 관리"을 참조하십시오.

버킷의 기본 보존 설정

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

기본 보존 모드

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

기본 보존 기간

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

버킷의 기본 보존 기간을 설정하는 방법

버킷의 기본 보존 기간을 설정하려면 다음 방법 중 하나를 사용하십시오.

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

PutObjectLockConfiguration

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

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

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

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

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

요청 예시

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

```
PUT /bucket?object-lock HTTP/1.1
Accept-Encoding: identity
Content-Length: 308
Host: host
Content-MD5: request header
User-Agent: s3sign/1.0.0 requests/2.24.0 python/3.8.2
X-Amz-Date: date
X-Amz-Content-SHA256: authorization-string
Authorization: authorization-string

<ObjectLockConfiguration>
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

버킷의 기본 보존 기간을 확인하는 방법

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

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

GetObjectLockConfiguration

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

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

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

요청 예시

```
GET /bucket?object-lock HTTP/1.1
Host: host
Accept-Encoding: identity
User-Agent: aws-cli/1.18.106 Python/3.8.2 Linux/4.4.0-18362-Microsoft
botocore/1.17.29
x-amz-date: date
x-amz-content-sha256: authorization-string
Authorization: authorization-string
```

응답 예시

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB7OXXJRkRH1FiVq1151/T24gRfpwpuZrEG11Bb9ImOMAAe98oxSpXlknabA0LTvBYJpSIX
k=
x-amz-request-id: B34E94CACB2CEF6D
Date: Fri, 04 Sep 2020 22:47:09 GMT
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

객체에 대한 보존 설정을 지정하는 방법

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

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

각 객체에 대해 다음과 같은 설정을 지정할 수 있습니다.

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

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

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

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

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

다음을 사용할 수 있습니다.

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

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

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

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

이러한 요청 헤더를 사용하는 경우 다음 제한 사항에 유의하십시오.

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

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

객체의 보존 설정을 업데이트하는 방법

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

- `PutObjectLegalHold`

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

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

거버넌스 모드 사용 방법

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

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

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

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

  규정 준수 모드에서 거버넌스 모드로 변경하는 것은 절대 허용되지 않습니다.

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

관련 정보

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

StorageGRID의 **S3** 라이프사이클 구성에 대해 알아보십시오

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

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

작업만 지원하며 전환 작업은 지원하지 않습니다.

라이프사이클 구성이란 무엇인가요?

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

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

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

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

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

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

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

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

라이프사이클 구성 생성

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

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

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

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

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

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

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

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

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

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

성공적인 응답에는 방금 적용한 라이프사이클 구성이 나열됩니다.

버킷 수명 주기 만료가 객체에 적용되는지 확인합니다.

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

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

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

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

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

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

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

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

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

```
{
      "AcceptRanges": "bytes",
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:48 GMT\", rule-
id=\"rule2\"",
      "LastModified": "2020-06-23T09:07:48+00:00",
      "ContentLength": 921,
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
      "ContentType": "binary/octet-stream",
      "Metadata": {}
}
```

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

StorageGRID로 **S3 REST API** 구현 권장 사항

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

존재하지 않는 객체에 대한 **HEAD** 권장 사항

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

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

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

객체 키에 대한 권장 사항

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

StorageGRID 11.4 이하 버전에서 생성된 버킷

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

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

이 형식 대신에:

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

StorageGRID 11.4 이상에서 생성된 버킷

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

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

"범위 읽기"에 대한 권장 사항

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

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

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

Amazon S3 REST API 지원

StorageGRID에서 지원되는 **S3 REST API** 작업 및 제한 사항

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

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

날짜 처리

StorageGRID에서 구현한 S3 REST API는 유효한 HTTP 날짜 형식만 지원합니다.

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

일반적인 요청 헤더

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

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

일반적인 응답 헤더

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

응답 헤더	구현
x-amz-id-2	사용하지 않음

StorageGRID S3 REST API가 요청을 인증하는 방법

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

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

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

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

HTTP Authorization 헤더 사용

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

쿼리 매개변수 사용

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

StorageGRID에서 지원되는 서비스 작업

StorageGRID 시스템은 서비스에 대해 다음과 같은 작업을 지원합니다.

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

StorageGRID의 **S3** 버킷 작업 및 구현 세부 정보

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

각 그리드는 최대 100,000개의 버킷을 가질 수 있습니다.

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

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

자세한 내용은 다음을 참조하십시오.

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

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

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

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

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

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

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

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

작업	구현

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

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

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

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

객체에 대한 연산

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

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

다음 조건은 모든 객체 작업에 적용됩니다.

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

- `OPTIONS /`
- PutObjectLegalHold
- PutObjectRetention
- SelectObjectContent

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

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

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

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

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

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

목록에 없는 항목은 지원되지 않습니다.

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

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

조항

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

데이터 유형

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

운영자

논리 연산자

- AND
- NOT
- 또는

비교 연산자

- <
- >
- ⇐
- >=
- =

- =
- <>
- !=
- 사이
- IN

패턴 매칭 연산자

- LIKE
- _
- %

단일 연산자

- IS NULL
- NULL이 아닙니다

수학 연산자

- +
- -
- *
- /
- %

StorageGRID는 Amazon S3 Select 연산자 우선순위를 따릅니다.

집계 함수

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

조건 함수

- 사례
- COALESCE
- NULLIF

변환 함수

- CAST(지원되는 데이터 형식의 경우)

날짜 함수

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

문자열 함수

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

StorageGRID에서 서버 측 암호화를 사용하는 방법

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

${post_edited_translations.segment}

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

  ${post_edited_translations.segment}

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

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

${post_edited_translations.segment}

${post_edited_translations.segment}

`x-amz-server-side-encryption`

SSE 요청 헤더는 다음 객체 작업에서 지원됩니다.

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

SSE-C 사용

${post_edited_translations.segment}

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

${post_edited_translations.segment}

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

${post_edited_translations.segment}

SSE-C를 사용하기 전에 다음 사항을 고려하십시오.

- https를 사용해야 합니다.

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

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

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

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

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

관련 정보

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

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

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

충돌 해결

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

객체 크기

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

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

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

사용자 메타데이터의 **UTF-8** 문자

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

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

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

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

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

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

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

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

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

    서버 측 암호화에 대한 요청 헤더을 참조하십시오

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

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

- If-None-Match

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

- x-amz-checksum-algorithm

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

- x-amz-website-redirect-location

스토리지 클래스 옵션

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

- STANDARD

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

- REDUCED_REDUNDANCY

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

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

CopyObject에서 x-amz-copy-source 사용

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

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

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

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

버전 관리

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

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

S3 GetObject 요청을 사용하여 S3 버킷에서 객체를 검색할 수 있습니다.

GetObject 및 멀티파트 객체

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

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

사용자 메타데이터의 **UTF-8** 문자

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

지원되는 요청 헤더

다음 요청 헤더가 지원됩니다.

- `x-amz-checksum-mode`: 지정 `ENABLED`

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

지원되지 않는 요청 헤더

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

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

버전 관리

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

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

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

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

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

클라우드 스토리지 풀 객체에 대한 **GetObject**의 동작

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

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

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

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

클라우드 스토리지 풀의 멀티파트 또는 분할된 객체

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

이러한 경우:

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

GetObject 및 그리드 간 복제

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

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

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

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

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

HeadObject 및 멀티파트 객체

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

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

사용자 메타데이터의 **UTF-8** 문자

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

지원되는 요청 헤더

다음 요청 헤더가 지원됩니다.

- `x-amz-checksum-mode`

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

지원되지 않는 요청 헤더

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

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

버전 관리

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

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

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

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

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

HeadObject 클라우드 스토리지 풀 객체에 대한 응답

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

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

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

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

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

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

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

객체의 상태	**HeadObject**에 대한 응답

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

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

클라우드 스토리지 풀의 멀티파트 또는 분할된 객체

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

HeadObject 및 그리드 간 복제

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

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

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

S3 PutObject 요청을 사용하여 버킷에 객체를 추가할 수 있습니다.

충돌 해결

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

객체 크기

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

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

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

사용자 메타데이터 크기

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

사용자 메타데이터의 **UTF-8** 문자

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

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

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

객체 태그 제한

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

객체 소유권

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

생성한 객체도 포함됩니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

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

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

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

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

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

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

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

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

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

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

    서버 측 암호화에 대한 요청 헤더을 참조하십시오

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

- `If-Match`

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

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

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

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

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

다음 값을 사용할 수 있습니다 x-amz-storage-class:

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

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

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

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

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

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

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

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

버전 관리

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

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

Authorization 헤더에 대한 서명 계산

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

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

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

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

관련 정보

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

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

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

지원되는 요청 유형

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

버전 관리

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

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

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

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

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

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

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

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

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

시작하기 전에

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

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

CSV 요청 구문 예시

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

Parquet 요청 구문 예시

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

SQL 쿼리 예시

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

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

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

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

AWS-CLI 사용 예시 **(CSV)**

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

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

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

AWS-CLI 사용 예시 (Parquet)

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

출력 파일인 changes.csv의 처음 몇 줄은 다음과 같습니다.

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

멀티파트 업로드 작업

StorageGRID에서 지원되는 멀티파트 업로드 작업

이 섹션에서는 StorageGRID가 멀티파트 업로드 작업을 지원하는 방법을 설명합니다.

모든 멀티파트 업로드 작업에는 다음과 같은 조건과 유의사항이 적용됩니다.

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

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

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

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

StorageGRID S3 REST API가 멀티파트 업로드를 완료하는 방법

CompleteMultipartUpload 작업은 이전에 업로드된 부분들을 조합하여 객체의 멀티파트 업로드를 완료합니다.

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

충돌 해결

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

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

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

- `STANDARD`

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

- `REDUCED_REDUNDANCY`

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

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

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

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

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

- `If-Match`

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

- `If-None-Match`

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

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

버전 관리

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

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

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

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

복제, 알림 또는 메타데이터 알림 실패

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

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

"플랫폼 서비스 문제 해결"을 참조하십시오.

StorageGRID의 S3 CreateMultipartUpload 요청 헤더 및 옵션

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

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

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

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

- `STANDARD` (기본값)

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

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

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

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

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

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

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

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

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

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

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

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

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

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

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

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

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

  "S3 REST API를 사용하여 S3 Object Lock 구성"

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

    서버 측 암호화에 대한 요청 헤더

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

서버 측 암호화에 대한 요청 헤더

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

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

UploadPart 요청)에 이 세 가지 헤더를 모두 사용하십시오.

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

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

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

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

버전 관리

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

S3 ListMultipartUploads 작업 및 StorageGRID에서 지원되는 매개 변수

ListMultipartUploads 작업은 버킷에 대해 진행 중인 멀티파트 업로드 목록을 표시합니다.

다음 요청 매개변수가 지원됩니다.

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

버전 관리

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

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

UploadPart 작업은 객체의 멀티파트 업로드에서 파트를 업로드합니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

버전 관리

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

S3 UploadPartCopy 작업을 사용하여 **StorageGRID**에 객체의 일부를 업로드합니다

UploadPartCopy 작업은 기존 객체의 데이터를 데이터 소스로 복사하여 객체의 일부를

업로드합니다.

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

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

다음과 같은 요청 헤더가 지원됩니다.

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

버전 관리

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

StorageGRID S3 REST API 오류 응답

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

지원되는 **S3 API** 오류 코드

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

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

StorageGRID 사용자 지정 오류 코드

이름	설명	**HTTP** 상태
XBucketLifecycleNotAllowed	기존 규정 준수 버킷에서는 버킷 수명 주기 구성이 허용되지 않습니다.	400 잘못된 요청

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

StorageGRID 사용자 지정 작업

StorageGRID에서 지원되는 사용자 지정 버킷 작업

StorageGRID 시스템은 S3 REST API에 추가되는 사용자 지정 작업을 지원합니다.

다음 표에는 StorageGRID에서 지원하는 사용자 지정 작업이 나열되어 있습니다.

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

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

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

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

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

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

요청 예시

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

응답

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

일관성	설명
모두	모든 노드가 데이터를 즉시 수신해야 하며, 그렇지 않으면 요청이 실패합니다.
strong-global	모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.

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

응답 예시

```
HTTP/1.1 200 OK
Date: Fri, 18 Sep 2020 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/11.5.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<Consistency xmlns="http://s3.storagegrid.com/doc/2015-02-01/">read-after-
new-write</Consistency>
```

관련 정보

"일관성"

PUT Bucket 정합성 보장 요청을 사용하여 **StorageGRID**에서 정합성 보장 수준 지정

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

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

시작하기 전에

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

요청

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

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

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

요청 예시

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

관련 정보

"일관성"

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

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

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

요청 예시

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

응답 예시

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

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

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

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

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

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

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

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

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

요청 예시

이 예제는 버킷의 마지막 액세스 시간을 활성화합니다.

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

이 예제는 버킷의 마지막 액세스 시간을 비활성화합니다.

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

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

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

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

요청 예시

이 예시는 버킷에 대한 검색 통합 서비스를 비활성화하는 방법을 보여줍니다.

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

GET 버킷 메타데이터 알림 구성 요청을 사용하여 **StorageGRID**에서 검색 통합을 구성합니다

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

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

요청 예시

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

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

응답

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

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:_region:account-
ID_:domain/_mydomain/myindex/mytype_</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

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

이름	설명	필수
MetadataNotificationConfiguration	메타데이터 알림의 객체와 대상을 지정하는 데 사용되는 규칙의 컨테이너 태그입니다. 하나 이상의 Rule 요소를 포함합니다.	예
규칙	지정된 인덱스에 메타데이터를 추가해야 하는 객체를 식별하는 규칙에 대한 컨테이너 태그입니다. 접두사가 겹치는 규칙은 허용되지 않습니다. MetadataNotificationConfiguration 요소에 포함되어 있습니다.	예

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

응답 예시

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

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

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

관련 정보

"테넌트 계정 사용"

PUT Bucket 메타데이터 알림 구성 요청을 사용하여 **StorageGRID**에서 검색 통합 서비스를 활성화합니다

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

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

요청

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

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

접두사가 겹치는 구성은 유효하지 않으며 제출 시 거부됩니다. 예를 들어, 접두사가 `test`인 객체에 대한 규칙 하나와

접두사가 `test2`인 객체에 대한 규칙 하나를 포함하는 구성은 허용되지 않습니다.

대상은 StorageGRID 엔드포인트의 URN을 사용하여 지정해야 합니다. 메타데이터 알림 구성이 제출될 때 엔드포인트가 존재해야 하며, 그렇지 않으면 요청이 실패합니다 `400 Bad Request`. 오류 메시지는 다음과 같습니다. `Unable to save the metadata notification (search) policy. The specified endpoint URN does not exist: URN`.

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:region:account-
ID:domain/mydomain/myindex/mytype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

이 표는 메타데이터 알림 구성 XML의 요소를 설명합니다.

이름	설명	필수
MetadataNotificationConfiguration	메타데이터 알림의 객체와 대상을 지정하는 데 사용되는 규칙의 컨테이너 태그입니다. 하나 이상의 Rule 요소를 포함합니다.	예
규칙	지정된 인덱스에 메타데이터를 추가해야 하는 객체를 식별하는 규칙에 대한 컨테이너 태그입니다. 접두사가 겹치는 규칙은 허용되지 않습니다. MetadataNotificationConfiguration 요소에 포함되어 있습니다.	예
ID	규칙에 대한 고유 식별자입니다. 규칙 요소에 포함됩니다.	아니요

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

요청 예시

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

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

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

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

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

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Images-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/images</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-1:3333333:domain/es-
domain/graphics/imagetype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Videos-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/videos</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-west-1:22222222:domain/es-
domain/graphics/videotype</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

검색 통합 서비스에서 생성된 **JSON**

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

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

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

메타데이터 알림에 포함된 객체 메타데이터

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

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

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

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

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

관련 정보

"테넌트 계정 사용"

GET Storage Usage 요청을 사용하여 **StorageGRID**에서 계정 및 버킷 스토리지 사용량을 조회합니다

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

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

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

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

요청 예시

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

응답 예시

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

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 00:49:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/10.2.0
x-amz-request-id: 727237123
Content-Length: 427
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<UsageResult xmlns="http://s3.storagegrid.com/doc/2015-02-01">
<CalculationTime>2014-11-19T05:30:11.000000Z</CalculationTime>
<ObjectCount>4</ObjectCount>
<DataBytes>12</DataBytes>
<Buckets>
<Bucket>
<Name>bucket1</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
<Bucket>
<Name>bucket2</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
</Buckets>
</UsageResult>
```

버전 관리

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

관련 정보

"일관성"

기존 규정 준수를 위한 더 이상 사용되지 않는 버킷 요청

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

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

규정 준수 기능이 더 이상 사용되지 않습니다.

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

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

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

더 이상 사용되지 않는 규정 준수 요청:

- "사용 중단됨 - 규정 준수를 위한 PUT 버킷 요청 수정"

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

- "사용 중단됨 - GET 버킷 규정 준수"

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

- "사용 중단됨 - PUT Bucket compliance"

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

StorageGRID의 더 이상 사용되지 않는 **SGCompliance** 요소

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

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

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

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

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

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

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

요청 예시

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

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

응답 예시

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

```
HTTP/1.1 200 OK
Date: date
Connection: connection
Server: StorageGRID/11.1.0
x-amz-request-id: request ID
Content-Length: length
Content-Type: application/xml

<SGCompliance>
  <RetentionPeriodMinutes>525600</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

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

오류 응답

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

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

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

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

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

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

요청 예시

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

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: date
Authorization: authorization name
Host: host
Content-Length: 152

<SGCompliance>
  <RetentionPeriodMinutes>1051200</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

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

규정 준수 설정의 일관성

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

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

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

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

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

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

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

오류 응답

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

관련 정보

"사용 중단 예정: 규정 준수를 위한 PUT 버킷 요청 수정"

${post_edited_translations.segment}

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

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

액세스 정책 개요

StorageGRID는 세 가지 유형의 액세스 정책을 지원합니다.

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

정책에서 부여된 권한 외에 사용자가 갖는 권한을 추가로 정의합니다.

그룹, 버킷 및 세션 정책 간에는 우선순위 차이가 없습니다.

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

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

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

각 정책 요소는 특정 기능을 위해 사용됩니다.

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

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

```
s3:*Object
```

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

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

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

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

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

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::27233906934684427525:federated-group/admin",
          "arn:aws:iam::27233906934684427525:federated-group/finance"
        ]
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::mybucket",
        "arn:aws:s3:::mybucket/*"
      ]
    }
  ]
}
```

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

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

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

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

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

세션 정책이란 무엇인가요?

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

${post_edited_translations.segment}

정책 설명에서 ARN은 Principal 요소와 Resource 요소에 사용됩니다.

- ${post_edited_translations.segment}

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

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

```
arn:aws:iam::account_id:root
arn:aws:iam::account_id:user/user_name
arn:aws:iam::account_id:group/group_name
arn:aws:iam::account_id:federated-user/user_name
arn:aws:iam::account_id:federated-group/group_name
```

기타 고려 사항:

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

  "RFC 2141 URN 구문"

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

정책에 리소스 지정

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

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

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

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

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

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

정책에서 보안 주체 지정

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

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

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

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

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

- 계정 루트만 지정할 수도 있습니다.

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

- 특정 연합 사용자("Alex")를 지정할 수 있습니다.

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

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

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

- 익명 주체를 지정할 수 있습니다.

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

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

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

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

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

정책에서 권한 지정

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

해당 표에는 버킷에 적용되는 권한과 객체에 적용되는 권한이 나열되어 있습니다.

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

> 삭제는 put 작업을 사용하여 기존 값을 덮어쓸 때 수행됩니다.

버킷에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:CreateBucket	CreateBucket	예. 참고: 그룹 정책에서만 사용합니다.
s3:DeleteBucket	DeleteBucket	
s3:DeleteBucketMetadataNotification	DELETE Bucket 메타데이터 알림 구성	예
s3:DeleteBucketPolicy	DeleteBucketPolicy	
s3:DeleteReplicationConfiguration	DeleteBucketReplication	예, PUT 및 DELETE에 대한 별도의 권한
s3:GetBucketAcl	GetBucketAcl	
s3:GetBucketCompliance	GET Bucket compliance(지원 중단됨)	예
s3:GetBucketConsistency	GET Bucket 일관성	예
s3:GetBucketCORS	GetBucketCors	
s3:GetEncryptionConfiguration	GetBucketEncryption	
s3:GetBucketLastAccessTime	GET 버킷 마지막 액세스 시간	예
s3:GetBucketLocation	GetBucketLocation	
s3:GetBucketMetadataNotification	GET 버킷 메타데이터 알림 구성	예
s3:GetBucketNotification	GetBucketNotificationConfiguration	
s3:GetBucketObjectLockConfiguration	GetObjectLockConfiguration	
s3:GetBucketPolicy	GetBucketPolicy	
s3:GetBucketTagging	GetBucketTagging	
s3:GetBucketVersioning	GetBucketVersioning	
s3:GetLifecycleConfiguration	GetBucketLifecycleConfiguration	

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:GetReplicationConfiguration	GetBucketReplication	
s3:ListAllMyBuckets	• ListBuckets • 저장 공간 사용량 가져오기	예, GET Storage Usage의 경우 그렇습니다. 참고: 그룹 정책에서만 사용합니다.
s3:ListBucket	• ListObjects • HeadBucket • RestoreObject	
s3:ListBucketMultipartUploads	• ListMultipartUploads • RestoreObject	
s3:ListBucketVersions	GET Bucket 버전	
s3:PutBucketCompliance	PUT Bucket compliance(지원 중단됨)	예
s3:PutBucketConsistency	PUT 버킷 일관성	예
s3:PutBucketCORS	• DeleteBucketCors† • PutBucketCors	
s3:PutEncryptionConfiguration	• DeleteBucketEncryption • PutBucketEncryption	
s3:PutBucketLastAccessTime	PUT Bucket 마지막 액세스 시간	예
s3:PutBucketMetadataNotification	PUT 버킷 메타데이터 알림 구성	예
s3:PutBucketNotification	PutBucketNotificationConfiguration	
s3:PutBucketObjectLockConfiguration	• CreateBucket with the `x-amz-bucket-object-lock-enabled: true` 요청 헤더(s3:CreateBucket 권한도 필요) • PutObjectLockConfiguration	
s3:PutBucketPolicy	PutBucketPolicy	

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:PutBucketTagging	• DeleteBucketTagging† • PutBucketTagging	
s3:PutBucketVersioning	PutBucketVersioning	
s3:PutLifecycleConfiguration	• DeleteBucketLifecycle† • PutBucketLifecycleConfiguration	
s3:PutReplicationConfiguration	PutBucketReplication	예, PUT 및 DELETE에 대한 별도의 권한

객체에 적용되는 권한

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

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:GetObjectLegalHold	GetObjectLegalHold	
s3:GetObjectRetention	GetObjectRetention	
s3:GetObjectTagging	GetObjectTagging	
s3:GetObjectVersionTagging	GetObjectTagging(객체의 특정 버전)	
s3:GetObjectVersion	GetObject(객체의 특정 버전)	
s3:ListMultipartUploadParts	ListParts, RestoreObject	
s3:PutObject	• PutObject • CopyObject • RestoreObject • CreateMultipartUpload • CompleteMultipartUpload • UploadPart • UploadPartCopy	
s3:PutObjectLegalHold	PutObjectLegalHold	
s3:PutObjectRetention	PutObjectRetention	
s3:PutObjectTagging	PutObjectTagging	
s3:PutObjectVersionTagging	PutObjectTagging(객체의 특정 버전)	
s3:PutOverwriteObject	• PutObject • CopyObject • PutObjectTagging • DeleteObjectTagging • CompleteMultipartUpload	예
s3:RestoreObject	RestoreObject	

PutOverwriteObject 권한 사용

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

이 권한에 대한 가능한 설정은 다음과 같습니다.

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

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

정책에 조건 지정

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

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

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

다음 예에서 IpAddress 조건은 SourceIp 조건 키를 사용합니다.

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

지원되는 조건 연산자

조건 연산자는 다음과 같이 분류됩니다.

- 문자열

- 숫자
- 부울
- IP 주소
- 널 검사

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

조건 연산자	설명
IfExists	Null 조건을 제외한 모든 조건 연산자에 추가하여 해당 조건 키의 부재 여부를 확인합니다. 조건 키가 없으면 TRUE를 반환합니다.

지원되는 조건 키

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

조건 키	작업	설명
s3:ExistingObjectTag/<tag-key>	s3:DeleteObjectTagging s3:DeleteObjectVersionTagging s3:GetObject s3:GetObjectAcl 3:GetObjectTagging s3:GetObjectVersion s3:GetObjectVersionAcl s3:GetObjectVersionTagging s3:PutObjectAcl s3:PutObjectTagging s3:PutObjectVersionAcl s3:PutObjectVersionTagging	기존 객체에 특정 태그 키와 값이 있어야 합니다.
s3:max-keys	s3:ListBucket 및 s3:ListBucketVersions 권한	ListObjects 또는 ListObjectVersions 요청에 지정된 max-keys 매개변수와 비교합니다.
s3:object-lock-mode	s3:PutObject	요청 헤더에서 확장된 `object-lock-mode`을(를) PutObject, CopyObject 및 CreateMultipartUpload 요청과 비교합니다.
s3:object-lock-mode	s3:PutObjectRetention	`object-lock-mode`PutObjectRetention 요청의 XML 본문에서 확장된 내용과 비교합니다.

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

정책에 변수 지정

정책에서 변수를 사용하여 정책 정보가 있을 때 이를 채울 수 있습니다. 정책 변수는 `Resource` 요소 내에서 또는 `Condition` 요소 내의 문자열 비교에서 사용할 수 있습니다.

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

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

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

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

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

특별 처리가 필요한 정책 생성

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

정책 설명	정책 유형	**Amazon** 동작	**StorageGRID** 동작
루트 계정에 대한 모든 권한을 스스로 거부합니다.	버킷	유효하고 시행 중이지만, 루트 사용자 계정은 모든 S3 버킷 정책 작업에 대한 권한을 유지합니다.	동일
사용자/그룹에 대한 모든 권한을 자신에게 거부	그룹	유효하고 강제력이 있습니다	동일
외부 계정 그룹에 모든 권한 허용	버킷	잘못된 주체	유효하지만 정책에서 허용된 경우에도 모든 S3 버킷 정책 작업에 대한 권한이 405 Method Not Allowed 오류를 반환합니다

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

한 번만 쓰고 여러 번 읽을 수 있는**(WORM)** 보호

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

덮어쓰기가 항상 거부되도록 하려면 다음을 수행할 수 있습니다.

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

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

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

상황 **A**: 동시 쓰기(방지되지 않음)

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

상황 **B**: 순차적으로 완료된 덮어쓰기(방지됨)

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

관련 정보

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

StorageGRID S3 REST API 세션 정책 예

다음 예제를 사용하여 StorageGRID 세션 정책을 구성하십시오.

예시: 객체 검색을 허용하는 세션 정책 설정

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

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

StorageGRID S3 REST API 버킷 정책 예

이 섹션의 예제를 사용하여 버킷에 대한 StorageGRID 액세스 정책을 구성하십시오.

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

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

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

예시: 모든 사용자에게 버킷에 대한 읽기 전용 액세스 권한 허용

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

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

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

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

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

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "95390887230002558202"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::examplebucket/shared/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::examplebucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "shared/*"
        }
      }
    }
  ]
}
```

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

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

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/Marketing"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:ListBucket","s3:GetObject"],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

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

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

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadWriteAccessIfInSourceIpRange",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:*Object", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"],
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
        "NotIpAddress": {"aws:SourceIp": "54.240.143.188"}
      }
    }
  ]
}
```

예시: 지정된 페더레이션 사용자에게만 버킷에 대한 전체 액세스 권한 허용

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

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

예시: **PutOverwriteObject** 권한

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

```
{
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:PutOverwriteObject",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::wormbucket/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::wormbucket"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::wormbucket/*"
    }
  ]
}
```

StorageGRID S3 REST API 그룹 정책 예

이 섹션의 예제를 사용하여 그룹에 대한 StorageGRID 액세스 정책을 구성하십시오.

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

예시**: Tenant Manager**를 사용하여 그룹 정책 설정

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

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

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

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

예시: 모든 버킷에 대한 그룹의 전체 액세스 권한 부여

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

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

예시: 모든 버킷에 대해 그룹별 읽기 전용 액세스 권한 허용

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

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

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

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

```
{
  "Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::department-bucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::department-bucket/${aws:username}/*"
    }
  ]
}
```

StorageGRID 감사 로그에 기록된 S3 작업

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

감사 로그에서 추적되는 버킷 작업

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

감사 로그에 기록된 객체 작업

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

관련 정보

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

Swift REST API 사용 (지원 종료)

StorageGRID의 Swift REST API 지원

Swift API 지원은 수명이 종료되었으며 향후 릴리스에서 제거될 예정입니다.

Swift 관련 세부 정보는 이 버전의 문서 사이트에서 삭제되었습니다. 을 참조하십시오 "StorageGRID 11.8: Swift REST API 사용".