Swift REST API 사용(더 이상 사용 안 함)

StorageGRID 11.7

NetApp
April 12, 2024

목차

Swift REST API 사용(더 이상 사용 안 함)

Swift REST API 사용: 개요

클라이언트 애플리케이션은 OpenStack Swift API를 사용하여 StorageGRID 시스템과 상호 작용할 수 있습니다.

Swift 클라이언트 응용 프로그램에 대한 지원은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

StorageGRID는 다음과 같은 특정 버전의 Swift 및 HTTP를 지원합니다.

항목	버전
SwiFT 사양	2015년 11월 기준 OpenStack Swift Object Storage API v1
HTTP	1.1 HTTP에 대한 자세한 내용은 HTTP/1.1(RFC 7230-35)을 참조하십시오. • 참고 *: StorageGRID는 HTTP/1.1 파이프라이닝을 지원하지 않습니다.

관련 정보

"OpenStack: 오브젝트 스토리지 API"

StorageGRID의 Swift API 지원 기록

Swift REST API에 대한 StorageGRID 시스템의 지원 변경 사항을 숙지해야 합니다.

놓습니다	설명
11.7	Swift 클라이언트 응용 프로그램에 대한 지원은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.
11.6	편집상의 사소한 변경.
11.5	약한 일관성 제어 기능이 제거되었습니다. 대신 사용 가능한 정합성 보장 레벨이 사용됩니다.
11.4	TLS 1.3에 대한 지원이 추가되었습니다. ILM과 정합성 보장 설정 간의 상호 관계에 대한 설명이 추가되었습니다.

놓습니다	설명
11.3	수집 시 동기식 배치를 사용하는 ILM 규칙(Ingest 동작에 대한 균형 및 엄격 옵션)의 영향을 설명하기 위해 PUT 오브젝트 작업이 업데이트되었습니다. 로드 밸런서 끝점 또는 고가용성 그룹을 사용하는 클라이언트 연결에 대한 설명이 추가되었습니다. TLS 1.1 암호가 더 이상 지원되지 않습니다.
11.2	문서에 대한 사소한 편집 변경.
11.1	그리드 노드에 대한 Swift 클라이언트 연결에 HTTP 사용 지원이 추가되었습니다. 일관성 제어의 정의를 업데이트했습니다.
11.0	각 테넌트 계정에 대해 1,000개의 컨테이너에 대한 지원이 추가되었습니다.
10.3	문서의 관리 업데이트 및 수정. 사용자 지정 서버 인증서를 구성하기 위한 섹션이 제거되었습니다.
10.2	StorageGRID 시스템에서 Swift API의 초기 지원 현재 지원되는 버전은 OpenStack Swift Object Storage API v1 입니다.

StorageGRID가 Swift REST API를 구현하는 방법

클라이언트 애플리케이션은 Swift REST API 호출을 사용하여 스토리지 노드 및 게이트웨이 노드에 연결하여 컨테이너를 생성하고 오브젝트를 저장 및 검색할 수 있습니다. 이를 통해 OpenStack Swift용으로 개발된 서비스 중심 애플리케이션을 StorageGRID 시스템에서 제공하는 사내 오브젝트 스토리지에 연결할 수 있습니다.

Swift 오브젝트 관리

Swift 객체가 StorageGRID 시스템에서 수집되면 시스템의 활성 ILM 정책에 있는 ILM(정보 수명 주기 관리) 규칙에 의해 관리됩니다. 를 클릭합니다 "ILM 규칙" 및 "ILM 정책" StorageGRID에서 오브젝트 데이터 복사본을 만들고 배포하는 방법과 시간이 지남에 따라 이러한 복사본을 관리하는 방법을 결정합니다. 예를 들어, ILM 규칙은 특정 Swift 컨테이너의 개체에 적용될 수 있으며 특정 기간 동안 여러 개체 복사본을 여러 데이터 센터에 저장하도록 지정할 수 있습니다.

그리드의 ILM 규칙 및 정책이 Swift 테넌트 계정의 개체에 어떤 영향을 미치는지 알아야 하는 경우 NetApp 프로페셔널 서비스 컨설턴트 또는 StorageGRID 관리자에게 문의하십시오.

클라이언트 요청 충돌

동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "최신" 평가 시기는 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 하며, Swift 클라이언트가 작업을 시작하는 시점이 아닙니다.

일관성 보장 및 제어

기본적으로 StorageGRID는 새로 생성된 객체에 대해 읽기 후 쓰기 정합성을 보장하고 객체 업데이트 및 헤드 작업에 대한 최종 일관성을 제공합니다. 모두 "가져오기" 성공적으로 팔로잉 완료 "를 누릅니다" 새로 작성된 데이터를 읽을 수

있습니다. 기존 오브젝트, 메타데이터 업데이트 및 삭제를 덮어쓰는 것은 결국 일관성이 유지됩니다. 덮어쓰기는 일반적으로 전파되는 데 몇 초 또는 몇 분이 걸리지만 최대 15일이 소요될 수 있습니다.

또한 StorageGRID를 사용하면 컨테이너 단위로 일관성을 제어할 수 있습니다. 일관성 제어는 애플리케이션의 요구에 따라 오브젝트의 가용성과 서로 다른 스토리지 노드 및 사이트 전체에서 오브젝트의 일관성 간의 균형을 제공합니다.

Swift REST API 구축을 위한 권장 사항

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

존재하지 않는 객체에 대한 헤드 권장 사항

응용 프로그램에서 개체가 실제로 존재하지 않을 것으로 예상되는 경로에 개체가 있는지 정기적으로 확인하는 경우 ""사용 가능한"" 일관성 제어를 사용해야 합니다. 예를 들어, 애플리케이션에서 해당 위치에 대한 PUT 작업을 수행하기 전에 헤드 작업을 수행하는 경우 ""사용 가능" 정합성 제어를 사용해야 합니다.

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

를 사용하여 각 컨테이너에 대해 ""사용 가능"" 일관성 제어를 설정할 수 있습니다 "컨테이너 일관성 요청". 를 사용하여 각 컨테이너에 대해 ""사용 가능"" 일관성 제어를 설정하는 것을 볼 수 있습니다 "컨테이너 일관성 요청 가져오기".

오브젝트 이름에 대한 권장사항

StorageGRID 11.4 이상에서 생성된 컨테이너의 경우 성능 모범 사례에 맞게 개체 이름을 제한할 필요가 없습니다. 예를 들어, 이제 개체 이름의 처음 4개 문자에 임의의 값을 사용할 수 있습니다.

StorageGRID 11.4 이전 릴리즈에서 만든 컨테이너의 경우 개체 이름에 대한 다음 권장 사항을 계속 따릅니다.

- 개체 이름의 처음 네 문자로 임의 값을 사용하면 안 됩니다. 이는 이전 AWS에서 권장하는 이름 접두사와 다릅니다. 대신 와 같이 고유하지 않은 비무작위 접두사를 사용해야 합니다 `image`.
- 이전 AWS 권장 사항에 따라 이름 접두사에 랜덤 및 고유 문자를 사용하려면 오브젝트 이름에 디렉토리 이름을 접두사로 붙여야 합니다. 즉, 다음 형식을 사용합니다.

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

  이 형식 대신:

  ```
  mycontainer/f8e3-image3132.jpg
  ```

""범위 읽기" 권장 사항

를 누릅니다 "저장된 개체를 압축하는 전역 옵션" 이 설정되어 있으면 Swift 클라이언트 애플리케이션이 반환할 바이트 범위를 지정하는 객체 가져오기 작업을 수행하지 않아야 합니다. 이러한 ""범위 읽기"" 작업은 StorageGRID가 요청된 바이트에 액세스하기 위해 개체의 압축을 효과적으로 해제해야 하기 때문에 비효율적입니다. 매우 큰 개체에서 작은 바이트 범위를 요청하는 Get Object 작업은 특히 비효율적입니다. 예를 들어, 50GB의 압축 개체에서 10MB 범위를 읽는 것은 매우 비효율적입니다.

압축된 개체에서 범위를 읽으면 클라이언트 요청이 시간 초과될 수 있습니다.

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

테넌트 계정 및 연결을 구성합니다

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

Swift 테넌트 계정을 생성하고 구성합니다

Swift API 클라이언트가 StorageGRID에 객체를 저장하고 검색하기 전에 Swift 테넌트 계정이 필요합니다. 각 테넌트 계정에는 고유한 계정 ID, 그룹 및 사용자, 컨테이너 및 객체가 있습니다.

Swift 테넌트 계정은 그리드 관리자 또는 그리드 관리 API를 사용하여 StorageGRID 그리드 관리자가 만듭니다.

시기 "Swift 테넌트 계정을 생성하는 중입니다"그리드 관리자는 다음 정보를 지정합니다.

- "테넌트의 표시 이름입니다" (테넌트의 계정 ID는 자동으로 할당되며 변경할 수 없음)
- 선택적으로, a "테넌트 계정의 스토리지 할당량입니다" — 테넌트 객체에 대해 사용 가능한 최대 GB, 테라바이트 또는 PB입니다. 테넌트의 스토리지 할당량은 물리적 크기(디스크 크기)가 아닌 논리적 양(오브젝트 크기)을 나타냅니다.
- If(경우 "SSO(Single Sign-On)" 는 StorageGRID 시스템에서 사용되지 않습니다. 테넌트 계정이 자체 ID 소스를 사용하거나 그리드의 ID 소스를 공유할지 여부 및 테넌트의 로컬 루트 사용자에 대한 초기 암호를 공유할지 여부입니다.
- SSO가 설정된 경우 테넌트 계정을 구성할 수 있는 루트 액세스 권한이 있는 통합 그룹이 있습니다.

Swift 테넌트 계정이 생성된 후 루트 액세스 권한이 있는 사용자는 테넌트 관리자에 액세스하여 다음과 같은 작업을 수행할 수 있습니다.

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

> Swift 사용자는 에 대한 루트 액세스 권한이 있어야 합니다 "테넌트 관리자를 액세스합니다". 그러나 루트 액세스 권한은 사용자가 Swift REST API에 인증하여 컨테이너를 생성하고 객체를 수집하는 것을 허용하지 않습니다. 사용자는 Swift REST API에 인증할 수 있는 Swift 관리자 권한이 있어야 합니다.

클라이언트 연결 구성 방법

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

클라이언트 응용 프로그램은 관리 노드 또는 게이트웨이 노드의 부하 분산 서비스에 연결하거나 선택적으로 관리 노드 또는 게이트웨이 노드의 고가용성(HA) 그룹의 가상 IP 주소에 연결하여 개체를 저장하거나 검색할 수 있습니다.

로드 밸런싱을 제공하기 위해 StorageGRID에 의존하는 모든 애플리케이션은 로드 밸런서 서비스를 사용하여 연결해야 합니다.

- 외부 로드 밸런서가 있거나 없는 스토리지 노드

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

1. 로드 밸런서 서비스의 끝점을 구성합니다.

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

2. 신뢰할 수 없는 클라이언트 네트워크를 구성합니다.

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

3. 고가용성 그룹을 구성합니다.

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

을 참조하십시오 "HA 그룹에 대한 구성 옵션" 를 참조하십시오.

요약**:** 클라이언트 연결을 위한 **IP** 주소 및 포트

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

클라이언트 연결을 만드는 데 필요한 정보입니다

이 표에는 클라이언트가 StorageGRID에 연결할 수 있는 다양한 방법과 각 연결 유형에 사용되는 IP 주소 및 포트가 요약되어 있습니다. 을 참조하십시오 "클라이언트 연결용 IP 주소 및 포트" 자세한 내용은 StorageGRID 관리자에게 문의하십시오.

연결 위치	클라이언트가 연결하는 서비스입니다	**IP** 주소입니다	포트
HA 그룹	로드 밸런서	HA 그룹의 가상 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
관리자 노드	로드 밸런서	관리 노드의 IP 주소입니다	• 로드 밸런서 엔드포인트 포트

연결 위치	클라이언트가 연결하는 서비스입니다	**IP** 주소입니다	포트
게이트웨이 노드	로드 밸런서	게이트웨이 노드의 IP 주소입니다	• 로드 밸런서 엔드포인트 포트
스토리지 노드	LDR	스토리지 노드의 IP 주소입니다	기본 Swift 포트: • HTTPS: 18083 • HTTP: 18085

예

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

- `https://VIP-of-HA-group:LB-endpoint-port`

예를 들어 HA 그룹의 가상 IP 주소가 192.0.2.6이고 Swift 로드 밸런서 끝점의 포트 번호가 10444인 경우 Swift 클라이언트는 다음 URL을 사용하여 StorageGRID에 연결할 수 있습니다.

- `https://192.0.2.6:10444`

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

HTTPS 또는 **HTTP** 연결을 사용하도록 결정합니다

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

기본적으로 클라이언트 애플리케이션이 스토리지 노드에 연결할 때는 모든 연결에 암호화된 HTTPS를 사용해야 합니다. 선택적으로 를 선택하여 보안성이 낮은 HTTP 연결을 활성화할 수 있습니다 "스토리지 노드 연결에 대해 HTTP를 설정합니다" 그리드 관리자의 옵션. 예를 들어, 클라이언트 애플리케이션은 비운영 환경에서 스토리지 노드에 대한 접속을 테스트할 때 HTTP를 사용할 수 있습니다.

> 요청 및 응답이 암호화되지 않은 상태로 전송되므로 프로덕션 그리드에 대해 HTTP를 설정할 때는 주의해야 합니다.

스토리지 노드 연결에 HTTP 사용 * 옵션을 선택한 경우 클라이언트는 HTTPS에 사용하는 것과 다른 HTTP 포트를 사용해야 합니다.

Swift API 구성에서 연결을 테스트합니다

Swift CLI를 사용하여 StorageGRID 시스템에 대한 연결을 테스트하고 시스템에 개체를 읽고 쓸 수 있는지 확인할 수 있습니다.

시작하기 전에

- Swift 명령줄 클라이언트인 python-swifftclient를 다운로드하여 설치해야 합니다.

"SwiftStack:python-swiftclient"

- StorageGRID 시스템에 Swift 테넌트 계정이 있어야 합니다.

이 작업에 대해

보안을 구성하지 않은 경우 을 추가해야 합니다 `--insecure` 이러한 각 명령에 플래그를 지정합니다.

단계

1. StorageGRID Swift 배포에 대한 정보 URL 쿼리:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/info
capabilities
```

이는 Swift 배포가 제대로 작동하는지 테스트하는 데 충분합니다. 객체를 저장하여 계정 구성을 추가로 테스트하려면 추가 단계를 계속 진행합니다.

2. 컨테이너에 개체 넣기:

```
touch test_object
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
upload test_container test_object
--object-name test_object
```

3. 컨테이너를 내려 개체를 확인합니다.

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
list test_container
```

4. 개체 삭제:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
delete test_container test_object
```

5. 컨테이너를 삭제합니다.

```
swift
-U `<_Tenant_Account_ID:Account_User_Name_>`
-K `<_User_Password_>`
-A `\https://<_FQDN_ | _IP_>:<_Port_>/auth/v1.0'
delete test_container
```

관련 정보

"Swift 테넌트 계정을 생성하고 구성합니다"

"REST API에 대한 보안을 구성합니다"

Swift REST API가 작업을 지원했습니다

StorageGRID 시스템은 OpenStack Swift API에서 대부분의 작업을 지원합니다. Swift REST API 클라이언트를 StorageGRID와 통합하기 전에 계정, 컨테이너 및 오브젝트 작업에 대한 구현 세부 정보를 검토하십시오.

StorageGRID에서 지원되는 작업입니다

다음과 같은 Swift API 작업이 지원됩니다.

- "계정 작업"
- "컨테이너 작업"
- "오브젝트 작업"

모든 작업에 대한 공통 응답 헤더입니다

StorageGRID 시스템은 OpenStack Swift 오브젝트 스토리지 API v1에 정의된 대로 지원되는 작업에 대해 모든 공통 헤더를 구현합니다.

관련 정보

"OpenStack: 오브젝트 스토리지 API"

지원되는 **Swift API** 엔드포인트

StorageGRID는 정보 URL, 인증 URL 및 스토리지 URL과 같은 Swift API 엔드포인트를 지원합니다.

정보 **URL**

/info 경로가 있는 Swift 기본 URL에 GET 요청을 실행하여 StorageGRID Swift 구현의 기능 및 제한 사항을 확인할 수 있습니다.

```
https://FQDN | Node IP:Swift Port/info/
```

요청 시:

- *FQDN* 정규화된 도메인 이름입니다.
- *Node IP* StorageGRID 네트워크의 스토리지 노드 또는 게이트웨이 노드에 대한 IP 주소입니다.
- *Swift Port* 스토리지 노드 또는 게이트웨이 노드의 Swift API 연결에 사용되는 포트 번호입니다.

예를 들어 다음 정보 URL은 IP 주소가 10.99.106.103이고 포트 18083을 사용하는 스토리지 노드에서 정보를 요청합니다.

```
https://10.99.106.103:18083/info/
```

응답에는 JSON 사전으로서 Swift 구현의 기능이 포함됩니다. 클라이언트 도구는 JSON 응답을 구문 분석하여 구현 기능을 결정하고 후속 스토리지 작업의 제약 조건으로 사용할 수 있습니다.

Swift의 StorageGRID 구현을 통해 정보 URL에 대한 인증되지 않은 액세스가 가능합니다.

인증 **URL**

클라이언트는 Swift 인증 URL을 사용하여 테넌트 계정 사용자로 인증할 수 있습니다.

```
https://FQDN | Node IP:Swift Port/auth/v1.0/
```

에서 테넌트 계정 ID, 사용자 이름 및 암호를 매개 변수로 제공해야 합니다 `X-Auth-User` 및 `X-Auth-Key` 다음과 같이 헤더를 요청합니다.

```
X-Auth-User: Tenant_Account_ID:Username
```

```
X-Auth-Key: Password
```

요청 헤더에서:

- *Tenant_Account_ID* Swift 테넌트가 생성될 때 StorageGRID에서 할당한 계정 ID입니다. 테넌트 관리자 로그인 페이지에서 사용되는 것과 동일한 테넌트 계정 ID입니다.
- *Username* 테넌트 관리자에서 생성된 테넌트 사용자의 이름입니다. 이 사용자는 Swift 관리자 권한이 있는 그룹에 속해야 합니다. 테넌트의 루트 사용자는 Swift REST API를 사용하도록 구성할 수 없습니다.

  테넌트 계정에 대해 ID 페더레이션을 사용하도록 설정한 경우 LDAP 서버에서 연결된 사용자의 사용자 이름과 암호를 입력합니다. 또는 LDAP 사용자의 도메인 이름을 제공합니다. 예를 들면 다음과 같습니다.

```
X-Auth-User: Tenant_Account_ID:Username@Domain_Name
```

- *Password* 테넌트 사용자의 암호입니다. 사용자 암호는 테넌트 관리자에서 생성 및 관리됩니다.

인증 요청에 대한 응답은 다음과 같이 스토리지 URL 및 인증 토큰을 반환합니다.

```
X-Storage-Url: https://FQDN | Node_IP:Swift_Port/v1/Tenant_Account_ID

X-Auth-Token: token

X-Storage-Token: token
```

기본적으로 토큰은 생성 후 24시간 동안 유효합니다.

특정 테넌트 계정에 대해 토큰이 생성됩니다. 한 계정에 대해 유효한 토큰이 사용자에게 다른 계정에 액세스할 수 있는 권한을 부여하지 않습니다.

스토리지 **URL**입니다

클라이언트 애플리케이션은 Swift REST API 호출을 실행하여 게이트웨이 노드 또는 스토리지 노드에 대해 지원되는 계정, 컨테이너 및 오브젝트 작업을 수행할 수 있습니다. 저장소 요청은 인증 응답에서 반환된 저장소 URL로 처리됩니다. 또한 요청에는 인증 요청에서 반환된 X-Auth-Token 헤더 및 값이 포함되어야 합니다.

```
https://FQDN | IP:Swift_Port/v1/Tenant_Account_ID

[/container][/object]

X-Auth-Token: token
```

사용 통계를 포함하는 일부 스토리지 응답 헤더는 최근에 수정된 개체의 정확한 숫자를 반영하지 않을 수 있습니다. 이 머리글에 정확한 숫자가 표시되려면 몇 분 정도 걸릴 수 있습니다.

계정 및 컨테이너 작업에 대한 다음 응답 머리글은 사용 통계를 포함하는 응답의 예입니다.

- `X-Account-Bytes-Used`
- `X-Account-Object-Count`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`

관련 정보

"테넌트 계정 및 연결을 구성합니다"

"계정 작업"

"컨테이너 작업"

"오브젝트 작업"

계정 작업

다음 Swift API 작업은 어카운트에 대해 수행됩니다.

계정을 가져옵니다

이 작업은 계정 및 계정 사용 통계와 연결된 컨테이너 목록을 검색합니다.

다음 요청 매개 변수가 필요합니다.

- `Account`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

다음과 같은 지원되는 요청 쿼리 매개 변수는 선택 사항입니다.

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`
- `Marker`
- `Prefix`

계정이 발견되어 컨테이너가 없거나 컨테이너 목록이 비어 있는 경우 "HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 성공적으로 실행되면 "HTTP/1.1 200 OK" 응답이 반환됩니다. 계정이 발견되어 컨테이너 목록이 비어 있지 않은 경우 "HTTP/1.1 200 OK" 응답이 반환됩니다.

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Account-Bytes-Used`
- `X-Account-Container-Count`
- `X-Account-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

머리 계정

이 작업은 Swift 계정에서 계정 정보 및 통계를 검색합니다.

다음 요청 매개 변수가 필요합니다.

- `Account`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

성공적으로 실행하면 ""HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 반환됩니다.

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Account-Bytes-Used`
- `X-Account-Container-Count`
- `X-Account-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

관련 정보

"운영 모니터링 및 감사"

컨테이너 작업

StorageGRID는 Swift 계정당 최대 1,000개의 컨테이너를 지원합니다. 컨테이너에서 다음과 같은 Swift API 작업이 수행됩니다.

컨테이너를 삭제합니다

이 작업을 수행하면 StorageGRID 시스템의 Swift 계정에서 빈 컨테이너가 제거됩니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

성공적으로 실행하면 "HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 반환됩니다.

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

컨테이너를 가져옵니다

이 작업은 StorageGRID 시스템의 컨테이너 통계 및 메타데이터와 함께 컨테이너와 연결된 개체 목록을 검색합니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

다음과 같은 지원되는 요청 쿼리 매개 변수는 선택 사항입니다.

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`
- `Marker`
- `Path`
- `Prefix`

성공적으로 실행하면 "HTTP/1.1 200 Success" 또는 "HTTP/1.1 204 No Content" 응답으로 다음 헤더가 반환됩니다.

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

헤드 컨테이너

이 작업은 StorageGRID 시스템에서 컨테이너 통계 및 메타데이터를 검색합니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

성공적으로 실행하면 "HTTP/1.1 204 컨텐츠 없음" 응답이 있는 다음 헤더가 반환됩니다.

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

용기를 놓습니다

이 작업은 StorageGRID 시스템의 계정에 대한 컨테이너를 만듭니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

성공적으로 실행하면 "HTTP/1.1 201 Created" 또는 "HTTP/1.1 202 Accepted"(컨테이너가 이미 이 계정에 있는 경우) 응답으로 다음 헤더가 반환됩니다.

- `Content-Length`
- `Date`
- `X-Timestamp`
- `X-Trans-Id`

컨테이너 이름은 StorageGRID 네임스페이스에서 고유해야 합니다. 컨테이너가 다른 계정 아래에 있는 경우 "HTTP/1.1 409 충돌"이라는 헤더가 반환됩니다.

관련 정보

"운영 모니터링 및 감사"

오브젝트 작업

객체에 대해 다음과 같은 Swift API 작업이 수행됩니다. 이러한 작업은 에서 추적할 수 있습니다 "StorageGRID 감사 로그".

개체를 삭제합니다

이 작업은 StorageGRID 시스템에서 개체의 콘텐츠 및 메타데이터를 삭제합니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`
- `Object`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

성공적으로 실행하면 다음 응답 헤더가 와 함께 반환됩니다 `HTTP/1.1 204 No Content` 응답:

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

오브젝트 삭제 요청을 처리할 때 StorageGRID는 저장된 모든 위치에서 오브젝트의 모든 복사본을 즉시 제거하려고 시도합니다. 성공하면 StorageGRID는 즉시 클라이언트에 응답을 반환합니다. 위치를 일시적으로 사용할 수 없기 때문에 30초 이내에 모든 복사본을 제거할 수 없는 경우 StorageGRID는 제거할 복사본을 대기시킨 다음 클라이언트에 성공 여부를 표시합니다.

자세한 내용은 을 참조하십시오 "오브젝트 삭제 방법".

객체를 가져옵니다

이 작업은 개체 콘텐츠를 검색하고 StorageGRID 시스템에서 개체 메타데이터를 가져옵니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`
- `Object`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

다음 요청 헤더는 선택 사항입니다.

- `Accept-Encoding`
- `If-Match`
- `If-Modified-Since`

- If-None-Match
- If-Unmodified-Since
- Range

성공적으로 실행하면 다음 헤더가 와 함께 반환됩니다 HTTP/1.1 200 OK 응답:

- Accept-Ranges
- Content-Disposition, 다음 경우에만 반환됩니다 Content-Disposition 메타데이터가 설정되었습니다
- Content-Encoding, 다음 경우에만 반환됩니다 Content-Encoding 메타데이터가 설정되었습니다
- Content-Length
- Content-Type
- Date
- ETag
- Last-Modified
- X-Timestamp
- X-Trans-Id

머리 물체

이 작업은 StorageGRID 시스템에서 수집된 개체의 메타데이터 및 속성을 검색합니다.

다음 요청 매개 변수가 필요합니다.

- Account
- Container
- Object

다음 요청 헤더가 필요합니다.

- X-Auth-Token

성공적으로 실행하면 "HTTP/1.1 200 OK" 응답과 함께 다음 헤더가 반환됩니다.

- Accept-Ranges
- Content-Disposition, 다음 경우에만 반환됩니다 Content-Disposition 메타데이터가 설정되었습니다
- Content-Encoding, 다음 경우에만 반환됩니다 Content-Encoding 메타데이터가 설정되었습니다
- Content-Length
- Content-Type
- Date
- ETag
- Last-Modified

- `X-Timestamp`
- `X-Trans-Id`

개체를 넣습니다

이 작업을 실행하면 새 개체가 데이터와 메타데이터로 만들어지거나 기존 개체를 StorageGRID 시스템의 데이터 및 메타데이터로 바꿉니다.

StorageGRID는 최대 5TiB(5,497,558,138,880바이트)의 오브젝트를 지원합니다.

> 동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "최신" 평가 시기는 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 하며, Swift 클라이언트가 작업을 시작하는 시점이 아닙니다.

다음 요청 매개 변수가 필요합니다.

- `Account`
- `Container`
- `Object`

다음 요청 헤더가 필요합니다.

- `X-Auth-Token`

다음 요청 헤더는 선택 사항입니다.

- `Content-Disposition`
- `Content-Encoding`

  청크를 사용하지 마십시오 `Content-Encoding` 개체에 적용되는 ILM 규칙이 크기에 따라 개체를 필터링하고 수집 시 동기식 배치(수집 동작에 대한 균형 또는 엄격 옵션)를 사용하는 경우

- `Transfer-Encoding`

  압축되거나 청크를 사용하지 마십시오 `Transfer-Encoding` 개체에 적용되는 ILM 규칙이 크기에 따라 개체를 필터링하고 수집 시 동기식 배치(수집 동작에 대한 균형 또는 엄격 옵션)를 사용하는 경우

- `Content-Length`

  ILM 규칙이 크기를 기준으로 오브젝트를 필터링하고 수집 시 동기 배치를 사용하는 경우 를 지정해야 합니다 `Content-Length`.

  > 에 대한 다음 지침을 따르지 않는 경우 `Content-Encoding`, `Transfer-Encoding`, 및 `Content-Length`에서 StorageGRID는 개체 크기를 결정하고 ILM 규칙을 적용하기 전에 개체를 저장해야 합니다. 다시 말해, StorageGRID은 수집 중인 오브젝트의 중간 복사본을 기본적으로 생성해야 합니다. 즉, StorageGRID는 Ingest 동작에 대해 이중 커밋 옵션을 사용해야 합니다.

  동기 배치 및 ILM 규칙에 대한 자세한 내용은 을 참조하십시오 "데이터 보호를 위한 수집 옵션".

- `Content-Type`
- `ETag`
- `X-Object-Meta-<name\>` (오브젝트 관련 메타데이터)

  ILM 규칙의 참조 시간으로 * 사용자 정의 생성 시간 * 옵션을 사용하려면 값을 라는 사용자 정의 헤더에 저장해야 합니다 `X-Object-Meta-Creation-Time`. 예를 들면 다음과 같습니다.

  ```
  X-Object-Meta-Creation-Time: 1443399726
  ```

  이 필드는 1970년 1월 1일 이후 초 단위로 평가됩니다.

- `X-Storage-Class: reduced_redundancy`

  수집된 개체와 일치하는 ILM 규칙이 이중 커밋 또는 균형 설정의 수집 동작을 지정하는 경우 이 헤더는 StorageGRID에서 만드는 개체 복사본 수에 영향을 줍니다.

  - * 이중 커밋 *: ILM 규칙이 Ingest 동작에 대한 이중 커밋 옵션을 지정하는 경우 StorageGRID는 오브젝트가 수집될 때(단일 커밋) 단일 임시 복사본을 만듭니다.
  - * 균형 *: ILM 규칙이 균형 옵션을 지정하는 경우 StorageGRID은 시스템에서 규칙에 지정된 모든 사본을 즉시 만들 수 없는 경우에만 단일 중간 복사본을 만듭니다. StorageGRID에서 동기 배치를 수행할 수 있는 경우 이 머리글은 영향을 주지 않습니다.

    를 클릭합니다 `reduced_redundancy` Header는 개체와 일치하는 ILM 규칙이 복제된 단일 복사본을 만들 때 가장 적합합니다. 이 경우 를 사용합니다 `reduced_redundancy` 모든 수집 작업에 대해 불필요한 오브젝트 복사본을 생성 및 삭제할 필요가 없습니다.

    를 사용합니다 `reduced_redundancy` 헤더는 수집 중에 오브젝트 데이터가 손실될 위험이 있기 때문에 다른 상황에서는 권장되지 않습니다. 예를 들어, ILM 평가가 발생하기 전에 실패한 스토리지 노드에 단일 복사본이 처음 저장되는 경우 데이터가 손실될 수 있습니다.

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

  를 지정하는 것에 주의하십시오 `reduced_redundancy` 오브젝트를 처음 인제스트할 때 생성되는 복사본 수에만 영향을 줍니다. 활성 ILM 정책에 따라 개체를 평가할 때 개체의 복사본 수에 영향을 주지 않으며 StorageGRID 시스템의 낮은 수준의 중복성에 데이터가 저장되지 않습니다.

성공적으로 실행하면 "HTTP/1.1 201 created" 응답으로 다음 헤더가 반환됩니다.

- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`

- `X-Trans-Id`

옵션 요청

옵션 요청은 개별 Swift 서비스의 사용 가능 여부를 확인합니다. 옵션 요청은 URL에 지정된 스토리지 노드 또는 게이트웨이 노드에 의해 처리됩니다.

옵션 방법입니다

예를 들어, 클라이언트 애플리케이션은 스토리지 노드의 Swift 인증 자격 증명을 제공하지 않고 스토리지 노드의 Swift 포트에 대한 옵션 요청을 발급하여 스토리지 노드를 사용할 수 있는지 여부를 확인할 수 있습니다. 이 요청을 사용하여 스토리지 노드가 다운된 시점을 모니터링하거나 외부 로드 밸런서가 식별하도록 할 수 있습니다.

info URL 또는 저장소 URL과 함께 사용할 경우 options 메서드는 지정된 URL에 대해 지원되는 동사 목록(예: head, get, options 및 put)을 반환합니다. 옵션 방법은 인증 URL과 함께 사용할 수 없습니다.

다음 요청 매개 변수가 필요합니다.

- `Account`

다음 요청 매개 변수는 선택 사항입니다.

- `Container`
- `Object`

성공적으로 실행하면 HTTP/1.1 204 컨텐츠 없음 응답이 있는 다음 헤더가 반환됩니다. 스토리지 URL에 대한 옵션 요청에는 타겟이 없을 필요가 없습니다.

- `Allow` 지정된 URL에 대해 지원되는 동사 목록(예: head, get, options, 및 PUT)
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

관련 정보

"지원되는 Swift API 엔드포인트"

Swift API 작업에 대한 오류 응답

가능한 오류 응답을 이해하면 작업 문제를 해결하는 데 도움이 됩니다.

작업 중에 오류가 발생하면 다음 HTTP 상태 코드가 반환될 수 있습니다.

SWIFT 오류 이름	**HTTP** 상태입니다
AccountNameTooLong, ContainerNameTooLong, HeaderTooBig, InvalidContainerName, InvalidRequest, InvalidURI, MetadataNameTooLong, MetadataValueTooBig, MissingSecurityHeader, ObjectNameTooLong, TooManyContainers, TooManyMetadataItems, TotalMetadataTooLarge입니다	400 잘못된 요청
액세스가 거부되었습니다	403 사용 금지
ContainerNotEmpty , ContainerAlreadyExists	409 충돌
내부 오류입니다	500 내부 서버 오류입니다
InvalidRange 를 선택합니다	416 요청된 범위가 충분하지 않습니다
MethodNotAllowed 를 참조하십시오	405 메서드를 사용할 수 없습니다
MissingContentLength를 참조하십시오	411 길이 필요
지원되지 않습니다	404를 찾을 수 없습니다
구현되지 않았습니다	501 구현되지 않음
사전 조건에 실패했습니다	412 전제 조건 실패
리소스 NotFound 를 참조하십시오	404를 찾을 수 없습니다
권한이 없습니다	401 승인되지 않음
UnprocessableEntity입니다	422 처리할 수 없는 엔터티

StorageGRID Swift REST API 작업

StorageGRID 시스템별 Swift REST API에 작업이 추가됩니다.

컨테이너 일관성 요청 가져오기

"일관성 제어" 오브젝트의 가용성과 서로 다른 스토리지 노드 및 사이트에서 이러한 오브젝트의 일관성 간의 균형을 제공합니다. 컨테이너 일관성 가져오기 요청을 사용하면 특정 컨테이너에 적용되는 일관성 수준을 확인할 수 있습니다.

요청하십시오

HTTP 헤더를 요청합니다	설명
X-Auth - 토큰	요청에 사용할 계정의 Swift 인증 토큰을 지정합니다.
X-NTAP-sg-정합성	요청 유형을 지정합니다. 여기서 는 `true` = 컨테이너 일관성 확보 및 `false` = 컨테이너를 가져옵니다.
호스트	요청이 전달되는 호스트 이름입니다.

요청 예

```
GET /v1/28544923908243208806/Swift container
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: true
Host: test.com
```

응답

응답 **HTTP** 헤더	설명
날짜	응답의 날짜 및 시간입니다.
연결	서버에 대한 연결이 열려 있는지 또는 닫혀 있는지 여부
X-Trans-ID	요청에 대한 고유한 트랜잭션 식별자입니다.
콘텐츠 - 길이	응답 바디의 길이.

응답 **HTTP** 헤더	설명
X-NTAP-sg-정합성	컨테이너에 적용되는 정합성 보장 제어 레벨입니다. 지원되는 값은 다음과 같습니다. • ALL *: 모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다. • strong-global *: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • strong-site *: 사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • read-after-new-write *: (기본값)는 새 객체에 대한 읽기 후 쓰기 정합성을 보장하고 객체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 대부분의 경우에 권장됩니다. • 사용 가능 *: 새 객체 및 객체 업데이트 모두에 대한 최종 일관성을 제공합니다. S3 버킷의 경우 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.

응답 예

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
x-ntap-sg-consistency: strong-site
```

컨테이너 일관성 요청

저장 컨테이너 일관성 요청을 사용하면 컨테이너에서 수행된 작업에 적용할 일관성 수준을 지정할 수 있습니다. 기본적으로 새 컨테이너는 "새 쓰기 후 다시 쓰기" 일관성 수준을 사용하여 생성됩니다.

요청하십시오

HTTP 헤더를 요청합니다	설명
X-Auth - 토큰	요청에 사용할 계정의 Swift 인증 토큰입니다.

HTTP 헤더를 요청합니다	설명
X-NTAP-sg-정합성	컨테이너의 작업에 적용할 일관성 제어 수준입니다. 지원되는 값은 다음과 같습니다. • ALL *: 모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다. • strong-global *: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • strong-site *: 사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. • read-after-new-write *: (기본값)는 새 객체에 대한 읽기 후 쓰기 정합성을 보장하고 객체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 대부분의 경우에 권장됩니다. • 사용 가능 *: 새 객체 및 객체 업데이트 모두에 대한 최종 일관성을 제공합니다. S3 버킷의 경우 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.
`Host`	요청이 전달되는 호스트 이름입니다.

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

둘 다 선택할 수 있습니다 "일관성 제어" ILM 규칙은 개체의 보호 방법에 영향을 줍니다. 이러한 설정은 상호 작용할 수 있습니다.

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

일관성 제어 및 **ILM** 규칙이 상호 작용하는 방법의 예

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

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

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

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

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

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

요청 예

```
PUT /v1/28544923908243208806/_Swift container_
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: strong-site
Host: test.com
```

응답

응답 **HTTP** 헤더	설명
`Date`	응답의 날짜 및 시간입니다.
`Connection`	서버에 대한 연결이 열려 있는지 또는 닫혀 있는지 여부
`X-Trans-Id`	요청에 대한 고유한 트랜잭션 식별자입니다.
`Content-Length`	응답 바디의 길이.

응답 예

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
```

REST API에 대한 보안을 구성합니다

REST API에 대해 구현된 보안 조치를 검토하고 시스템 보안 방법을 이해해야 합니다.

StorageGRID가 REST API에 보안을 제공하는 방법

StorageGRID 시스템이 REST API에 대한 보안, 인증 및 권한 부여를 구현하는 방법을 이해해야 합니다.

StorageGRID는 다음과 같은 보안 조치를 사용합니다.

- 로드 밸런서 끝점에 HTTPS가 구성되어 있는 경우 로드 밸런서 서비스와의 클라이언트 통신은 HTTPS를 사용합니다.

  언제 "로드 밸런서 끝점을 구성합니다", 선택적으로 HTTP를 활성화할 수 있습니다. 예를 들어, 테스트 또는 기타 비운영 목적으로 HTTP를 사용할 수 있습니다.

- 기본적으로 StorageGRID는 스토리지 노드와의 클라이언트 통신에 HTTPS를 사용합니다.

  필요한 경우 "이러한 연결에 대해 HTTP를 활성화합니다". 예를 들어, 테스트 또는 기타 비운영 목적으로 HTTP를 사용할 수 있습니다.

- StorageGRID와 클라이언트 간의 통신은 TLS를 사용하여 암호화됩니다.
- 로드 밸런서 끝점이 HTTP 또는 HTTPS 연결을 허용하도록 구성되었는지 여부에 관계없이 그리드 내의 로드 밸런서 서비스와 스토리지 노드 간의 통신이 암호화됩니다.
- 클라이언트는 REST API 작업을 수행하기 위해 StorageGRID에 HTTP 인증 헤더를 제공해야 합니다.

보안 인증서 및 클라이언트 응용 프로그램

클라이언트는 게이트웨이 노드 또는 관리 노드의 로드 밸런서 서비스에 직접 스토리지 노드에 연결할 수 있습니다.

모든 경우에 클라이언트 응용 프로그램은 그리드 관리자가 업로드한 사용자 지정 서버 인증서 또는 StorageGRID 시스템에서 생성한 인증서를 사용하여 TLS 연결을 만들 수 있습니다.

- 클라이언트 응용 프로그램이 로드 밸런서 서비스에 연결되면 연결을 만드는 데 사용되는 특정 로드 밸런서 끝점에 대해 구성된 인증서를 사용합니다. 각 끝점마다 고유한 인증서가 있습니다. 이 인증서는 그리드 관리자가 업로드한 사용자 지정 서버 인증서이거나, 끝점 구성 시 그리드 관리자가 StorageGRID에서 생성한 인증서입니다.
- 클라이언트 애플리케이션이 스토리지 노드에 직접 접속하면 시스템 인증 기관에서 서명한 StorageGRID 시스템을 설치할 때 스토리지 노드에 대해 생성된 시스템 생성 서버 인증서를 사용합니다. 또는 그리드 관리자가 그리드에 제공하는 단일 사용자 정의 서버 인증서입니다.

클라이언트가 TLS 연결을 설정하는 데 사용하는 인증서를 신뢰하도록 구성해야 합니다.

을 참조하십시오 "부하 분산 장치 엔드포인트 구성" 및 "단일 사용자 지정 서버 인증서 추가" 스토리지 노드에 직접 연결된 TLS 연결의 경우

요약

다음 표에서는 S3 및 Swift REST API에서 보안 문제가 구현되는 방식을 보여 줍니다.

보안 문제	**REST API** 구현
연결 보안	TLS
서버 인증	시스템 CA에서 서명한 X.509 서버 인증서 또는 관리자가 제공한 사용자 지정 서버 인증서입니다
클라이언트 인증	• S3:S3 계정(액세스 키 ID 및 비밀 액세스 키) • Swift:Swift 계정(사용자 이름 및 암호)
클라이언트 인증	• S3: 버킷 소유권 및 모든 적용 가능한 액세스 제어 정책 • Swift: 관리자 역할 액세스

TLS 라이브러리에 대해 지원되는 해시 및 암호화 알고리즘

StorageGRID 시스템은 TLS(전송 계층 보안) 세션을 설정할 때 클라이언트 응용 프로그램에서 사용할 수 있는 제한된 암호화 그룹 세트를 지원합니다. 암호를 구성하려면 * 구성 * > * 보안 * > * 보안 설정 * 으로 이동하여 * TLS 및 SSH 정책 * 을 선택합니다.

지원되는 **TLS** 버전입니다

StorageGRID는 TLS 1.2 및 TLS 1.3을 지원합니다.

SSLv3 및 TLS 1.1(또는 이전 버전)은 더 이상 지원되지 않습니다.

관련 정보

"테넌트 계정 및 연결을 구성합니다"

운영 모니터링 및 감사

전체 그리드 또는 특정 노드에 대한 트랜잭션 추세를 확인하여 클라이언트 작업의 워크로드 및 효율성을 모니터링할 수 있습니다. 감사 메시지를 사용하여 클라이언트 작업 및 트랜잭션을 모니터링할 수 있습니다.

오브젝트 수집 및 검색 속도 모니터링

오브젝트 수집 및 검색 속도와 오브젝트 수, 쿼리, 검증을 위한 메트릭을 모니터링할 수 있습니다. 클라이언트 응용 프로그램이 StorageGRID 시스템에서 개체를 읽고, 쓰고, 수정하는 데 성공한 시도 및 실패한 시도 횟수를 볼 수 있습니다.

단계

1. 를 사용하여 Grid Manager에 로그인합니다 "지원되는 웹 브라우저".
2. 대시보드에서 * 성능 * > * S3 작업 * 또는 * 성능 * > * Swift 작업 * 을 선택합니다.

   이 섹션에서는 StorageGRID 시스템에서 수행하는 클라이언트 작업의 수를 요약합니다. 프로토콜 속도는 최근 2분 동안의 평균값입니다.

3. 노드 * 를 선택합니다.
4. 노드 홈 페이지(배포 수준)에서 * 로드 밸런서 * 탭을 클릭합니다.

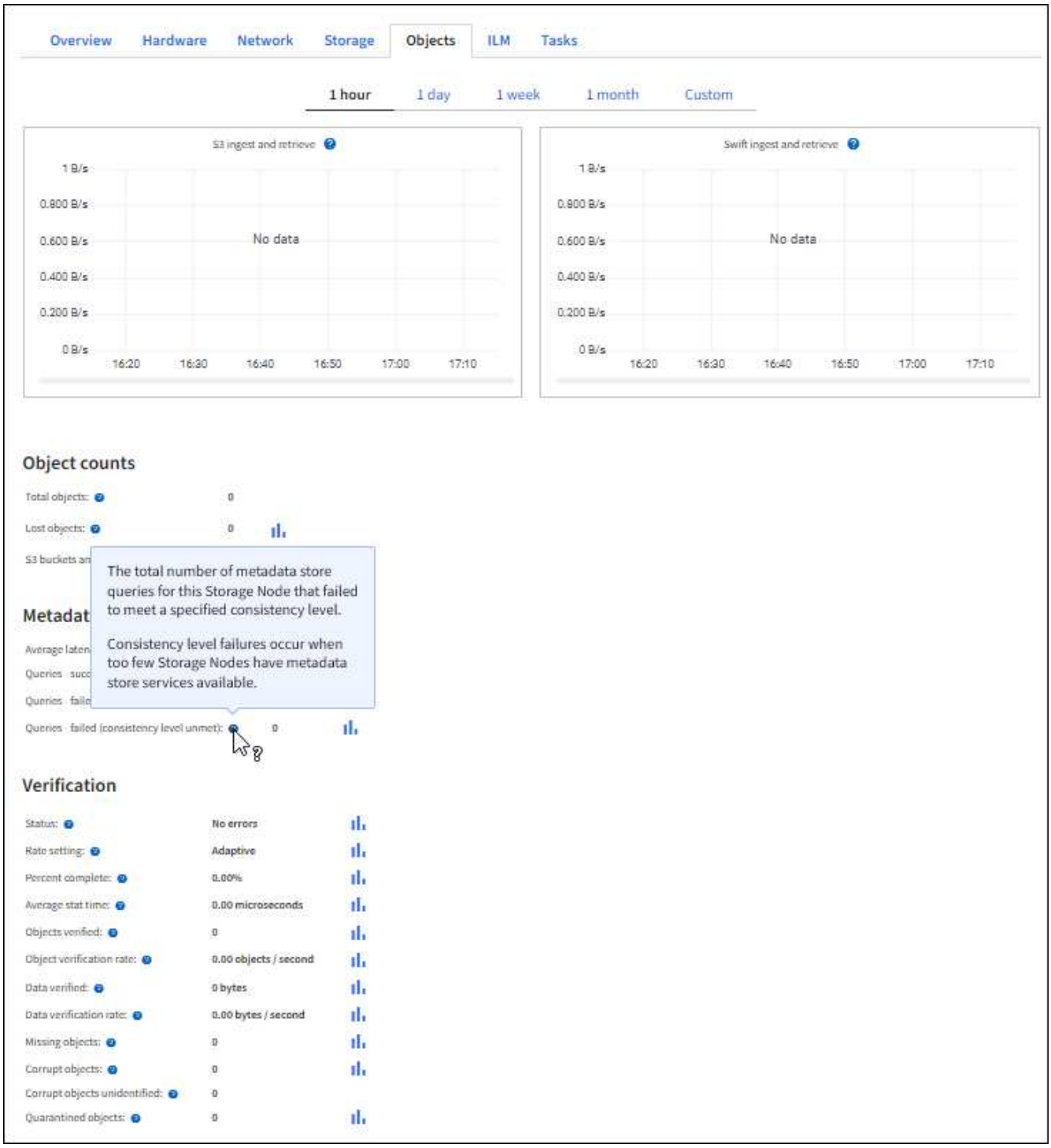
   차트에는 그리드 내의 로드 밸런서 끝점에 대한 모든 클라이언트 트래픽에 대한 추세가 표시됩니다. 시간 간격(시간, 일, 주, 월 또는 년)을 선택할 수 있습니다. 또는 사용자 지정 간격을 적용할 수 있습니다.

5. 노드 홈 페이지(배포 수준)에서 * 개체 * 탭을 클릭합니다.

   이 차트에는 전체 StorageGRID 시스템의 수집 및 검색 속도가 초당 바이트 및 총 바이트 단위로 표시됩니다. 시간 간격(시간, 일, 주, 월 또는 년)을 선택할 수 있습니다. 또는 사용자 지정 간격을 적용할 수 있습니다.

6. 특정 스토리지 노드에 대한 정보를 보려면 왼쪽의 목록에서 노드를 선택하고 * Objects * 탭을 클릭합니다.

   이 차트에는 이 스토리지 노드의 객체 수집 및 검색 속도가 나와 있습니다. 이 탭에는 개체 수, 쿼리 및 검증에 대한 메트릭도 포함되어 있습니다. 레이블을 클릭하여 이러한 메트릭의 정의를 볼 수 있습니다.



7. 더 자세한 내용을 원하는 경우:
   a. 지원 * > * 도구 * > * 그리드 토폴로지 * 를 선택합니다.
   b. site_ * > * Overview * > * Main * 을 선택합니다.

      API 작업 섹션에는 전체 그리드에 대한 요약 정보가 표시됩니다.

   c. 스토리지 노드 * > * LDR * > **CLIENT APPLICATION* * > * Overview * > * Main * 을 선택합니다

      작업 섹션에는 선택한 스토리지 노드에 대한 요약 정보가 표시됩니다.

감사 로그 액세스 및 검토

감사 메시지는 StorageGRID 서비스에서 생성되고 텍스트 로그 파일에 저장됩니다. 감사 로그의 API 관련 감사 메시지는 시스템의 상태를 평가하는 데 도움이 되는 중요한 보안, 운영 및 성능 모니터링 데이터를 제공합니다.

시작하기 전에

- 특정 액세스 권한이 있어야 합니다.
- 에 가 있어야 합니다 `Passwords.txt` 파일.
- 관리 노드의 IP 주소를 알아야 합니다.

이 작업에 대해

를 클릭합니다 "활성 감사 로그 파일" 이름이 지정됩니다 `audit.log`, 및 은 관리 노드에 저장됩니다.

하루에 한 번 활성 audit.log 파일이 저장되고 새 audit.log 파일이 시작됩니다. 저장된 파일의 이름은 저장 시기를 형식으로 나타냅니다 `yyyy-mm-dd.txt`.

하루 후에는 저장된 파일이 압축되고 이름이 파일 형식으로 변경됩니다 `yyyy-mm-dd.txt.gz`원래 날짜를 유지합니다.

이 예제에서는 활성 audit.log 파일, 이전 날짜의 파일(2018-04-15.txt) 및 이전 날짜의 압축 파일을 보여 줍니다 (`2018-04-14.txt.gz`)를 클릭합니다.

```
audit.log
2018-04-15.txt
2018-04-14.txt.gz
```

단계

1. 관리자 노드에 로그인:
   a. 다음 명령을 입력합니다. `ssh admin@primary_Admin_Node_IP`
   b. 에 나열된 암호를 입력합니다 `Passwords.txt` 파일.
   c. 루트로 전환하려면 다음 명령을 입력합니다. `su -`
   d. 에 나열된 암호를 입력합니다 `Passwords.txt` 파일.

      루트로 로그인하면 프롬프트가 에서 변경됩니다 `$` 를 선택합니다 `#`.
2. 감사 로그 파일이 포함된 디렉토리로 이동합니다.`cd /var/local/audit/export`
3. 필요에 따라 현재 또는 저장된 감사 로그 파일을 봅니다.

감사 로그에서 **Swift** 작업이 추적되었습니다

스토리지 삭제, 가져오기, 헤드, POST 및 PUT 작업이 모두 에서 추적됩니다 "StorageGRID 감사 로그". 실패는 기록되지 않으며 정보, 인증 또는 옵션 요청도 기록되지 않습니다.

다음 Swift 작업에 대한 정보가 추적됩니다.

계정 작업

- "계정을 가져옵니다"
- "머리 계정"

컨테이너 작업

- "컨테이너를 삭제합니다"
- "컨테이너를 가져옵니다"
- "헤드 컨테이너"
- "용기를 놓습니다"

오브젝트 작업

- "개체를 삭제합니다"
- "객체를 가져옵니다"
- "머리 물체"
- "개체를 넣습니다"