사용자 환경에서 **StorageGRID**를 활성화하는 방법

How to enable StorageGRID in your environment

NetApp
April 26, 2024

목차

사용자 환경에서 **StorageGRID**를 활성화하는 방법

검증된 타사 솔루션

검증된 타사 솔루션: 개요

NetApp은 파트너와 협력하여 이러한 솔루션을 StorageGRID와 함께 사용할 수 있도록 검증했습니다. 이 섹션의 정보를 검토하여 검증된 솔루션을 확인하고 해당하는 경우 추가 지침을 얻습니다.

NetApp과 함께 업계 최고의 검증된 NetApp 솔루션을 구축하여 포트폴리오 혁신을 가속하고 시장 인지도를 높이며 매출을 늘리십시오. "지금 바로 제휴 파트너가 되십시오".

StorageGRID 11.8은 타사 솔루션으로 검증되었습니다

다음 타사 솔루션은 StorageGRID 11.8과 함께 사용할 수 있도록 검증되었습니다. 를 누릅니다 찾고 있는 솔루션이 목록에 없으면 NetApp 계정 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 알럭시오
- 아파치 카프카
- AWS 마운트 지점
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Collibra(최소 Collibra 데이터 품질 버전 2024.02)
- Commvault 11
- Ctera 포털 6
- 다레트
- 다타도비
- Data Dynamics StorageX를 나타냅니다
- DefendX
- 디스크 오버 데이터
- 드리미오
- eMAM
- Fujifilm 개체 아카이브
- GitHub 엔터프라이즈 서버
- IBM Filenet

- IBM Spectrum Protect Plus
- 인터카
- 고마프라이즈
- Microsoft SQL Server 빅 데이터 클러스터
- 모델9
- Modzy
- 문워크 유니버설
- 멋져
- 나스니
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라
- PixitMedia ngenea 를 참조하십시오
- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4
- Primestream
- Quantum StorNext 5.4.0.1
- Reveille v10 빌드 220706 이상
- Rubrik CDM
- s3a 를 참조하십시오
- 용감합니다
- 스노우플레이크
- Spectra Logic On-Premise Glacier의 약어입니다
- Splunk 스마트스토어
- 간편한 보관
- 트리노
- Varnish Enterprise 6.0.4
- Veeam 12
- Veritas Enterprise Vault 14
- Veritas NetBackup 8.0
- Vertica 10.x
- 비딘
- Virtalica StorageFabric

- Weka v3.10 이상

StorageGRID에서 오브젝트 잠금을 통해 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- CommVault 11 기능 릴리스 26
- IBM Filenet
- OpenText Documentum 21.4
- Veeam 12
- Veritas Enterprise Vault 14.2.2
- Veritas NetBackup 10.1.1 이상

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 기트랩주식회사
- 하이랜드 아쿠오
- IBM Aspera
- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

StorageGRID 11.7 검증된 타사 솔루션

다음 타사 솔루션은 StorageGRID 11.7에서 사용하도록 검증되었습니다. + 찾으려는 솔루션이 목록에 없는 경우 NetApp 고객 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 알럭시오
- 아파치 카프카
- AWS 마운트 지점
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Collibra(최소 Collibra 데이터 품질 버전 2024.02)
- Commvault 11
- Ctera 포털 6
- 다레트
- 다타도비
- Data Dynamics StorageX를 나타냅니다
- DefendX
- 디스크 오버 데이터
- 드리미오
- eMAM
- Fujifilm 개체 아카이브
- GitHub 엔터프라이즈 서버
- IBM Filenet
- IBM Spectrum Protect Plus
- 인터카
- 고마프라이즈
- Microsoft SQL Server 빅 데이터 클러스터
- 모델9
- Modzy
- 문워크 유니버설
- 멋져
- 나스니
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7

- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라
- PixitMedia ngenea 를 참조하십시오
- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4
- Primestream
- Quantum StorNext 5.4.0.1
- Reveille v10 빌드 220706 이상
- Rubrik CDM
- s3a 를 참조하십시오
- 용감합니다
- 스노우플레이크
- Spectra Logic On-Premise Glacier의 약어입니다
- Splunk 스마트스토어
- 간편한 보관
- 트리노
- Varnish Enterprise 6.0.4
- Veeam 12
- Veritas Enterprise Vault 14
- Veritas NetBackup 8.0
- Vertica 10.x
- 비딘
- Virtalica StorageFabric
- Weka v3.10 이상

StorageGRID에서 오브젝트 잠금을 통해 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- CommVault 11 기능 릴리스 26
- IBM Filenet
- OpenText Documentum 21.4
- Veeam 12
- Veritas Enterprise Vault 14.2.2
- Veritas NetBackup 10.1.1 이상

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 기트랩주식회사
- 하이랜드 아쿠오
- IBM Aspera
- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

StorageGRID 11.6 검증된 타사 솔루션

다음 타사 솔루션은 StorageGRID 11.6에서 사용하도록 검증되었습니다. + 찾으려는 솔루션이 목록에 없는 경우 NetApp 고객 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 알럭시오
- 아파치 카프카
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Commvault 11

- Ctera 포털 6
- 다레트
- 다타도비
- Data Dynamics StorageX를 나타냅니다
- DefendX
- 디스크 오버 데이터
- 드리미오
- eMAM
- Fujifilm 개체 아카이브
- GitHub 엔터프라이즈 서버
- IBM Filenet
- IBM Spectrum Protect Plus
- 인터카
- 고마프라이즈
- Microsoft SQL Server 빅 데이터 클러스터
- 모델9
- Modzy
- 문워크 유니버설
- 멋져
- 나스니
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라
- PixitMedia ngenea 를 참조하십시오
- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4
- Primestream
- Quantum StorNext 5.4.0.1
- Reveille v10 빌드 220706 이상
- Rubrik CDM
- s3a 를 참조하십시오
- 용감합니다
- 스노우플레이크

- Spectra Logic On-Premise Glacier의 약어입니다
- Splunk 스마트스토어
- 간편한 보관
- 트리노
- Varnish Enterprise 6.0.4
- Veeam 12
- Veritas Enterprise Vault 14
- Veritas NetBackup 8.0
- Vertica 10.x
- 비딘
- Virtalica StorageFabric
- Weka v3.10 이상

StorageGRID에서 오브젝트 잠금을 통해 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- CommVault 11 기능 릴리스 26
- IBM Filenet
- OpenText Documentum 21.4
- Veeam 12
- Veritas Enterprise Vault 14.2.2
- Veritas NetBackup 10.1.1 이상

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 기트랩주식회사
- 하이랜드 아쿠오
- IBM Aspera

- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

StorageGRID 11.5는 타사 솔루션을 검증했습니다

다음 타사 솔루션은 StorageGRID 11.5에서 사용하도록 검증되었습니다. + 찾으려는 솔루션이 목록에 없는 경우 NetApp 고객 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 알럭시오
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Commvault 11
- Ctera 포털 6
- 다레트
- 다타도비
- Data Dynamics StorageX를 나타냅니다
- DefendX
- 인터카
- 고마프라이즈
- 문워크 유니버설
- 멋져
- 나스니
- OpenText Documentum 16.4
- OpenText Documentum 21.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라

- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4
- Primestream
- Quantum StorNext 5.4.0.1
- Rubrik CDM
- s3a 를 참조하십시오
- 용감합니다
- Splunk 스마트스토어
- 트리노
- Varnish Enterprise 6.0.4
- Veeam 11
- Veritas Enterprise Vault 11
- Veritas Enterprise Vault 12
- Veritas NetBackup 8.0
- Vertica 10.x
- 비딘
- Virtalica StorageFabric

StorageGRID에서 오브젝트 잠금을 통해 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- OpenText Documentum 21.4
- Veeam 11

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 기트랩주식회사
- 하이랜드 아쿠오

- IBM Aspera
- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

StorageGRID 11.4는 타사 솔루션의 유효성을 검증했습니다

다음 타사 솔루션은 StorageGRID 11.4에서 사용하도록 검증되었습니다. + 찾으려는 솔루션이 목록에 없는 경우 NetApp 고객 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Commvault 11
- Ctera 포털 6
- 다레트
- 다타도비
- Data Dynamics StorageX를 나타냅니다
- DefendX
- 인터카
- 고마프라이즈
- 멋져
- 나스니
- OpenText Documentum 16.4
- OpenText InfoArchive 16 EP7
- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라
- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4

- Primestream
- Quantum StorNext 5.4.0.1
- Rubrik CDM
- 용감합니다
- Splunk 스마트스토어
- Varnish Enterprise 6.0.4
- Veeam 9.5.4
- Veritas Enterprise Vault 11
- Veritas Enterprise Vault 12
- Veritas NetBackup 8.0
- Vertica 10.x
- 비딘

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 하이랜드 아쿠오
- IBM Aspera
- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

StorageGRID 11.3은 타사 솔루션의 유효성을 검증했습니다

다음 타사 솔루션은 StorageGRID 11.3에서 사용하도록 검증되었습니다. + 찾으려는 솔루션이

목록에 없는 경우 NetApp 고객 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Commvault 11
- Ctera 포털 6
- 다레트
- 다타도비
- Data Dynamics StorageX를 나타냅니다
- DefendX
- 인터카
- 고마프라이즈
- 멋져
- 나스니
- OpenText Documentum 16.4
- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라
- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4
- Primestream
- Quantum StorNext 5.4.0.1
- Rubrik CDM 5.0.1 P1-1342
- 용감합니다
- Splunk 스마트스토어
- Varnish Enterprise 6.0.4
- Veeam 9.5.4
- Veritas Enterprise Vault 11
- Veritas Enterprise Vault 12
- Veritas NetBackup 8.0
- 비딘

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 하이랜드 아쿠오
- IBM Aspera
- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

StorageGRID 11.2는 타사 솔루션을 검증했습니다

다음 타사 솔루션은 StorageGRID 11.2에서 사용하도록 검증되었습니다. + 찾으려는 솔루션이 목록에 없는 경우 NetApp 고객 담당자에게 문의하십시오.

StorageGRID에서 검증된 타사 솔루션

이러한 솔루션은 해당 파트너와 협력하여 테스트를 거쳤습니다.

- Actifio(활성)
- 브리지스톤
- 캔템오
- Citrix 콘텐츠 협업
- Commvault 11
- Ctera 포털 6
- 다레트
- 다타도비

- Data Dynamics StorageX를 나타냅니다
- DefendX
- 인터카
- 고마프라이즈
- 멋져
- 나스니
- OpenText Documentum 16.4
- CyanGate Cloud를 사용한 OpenText Media Management 16.5
- 판주라
- 포인트 아카이브 게이트웨이 2.0
- 포인트 스토리지 관리자 6.4
- Primestream
- Quantum StorNext 5.4.0.1
- Rubrik CDM 5.0.1 P1-1342
- 용감합니다
- Splunk 스마트스토어
- Varnish Enterprise 6.0.4
- Veeam 9.5.4
- Veritas Enterprise Vault 11
- Veritas Enterprise Vault 12
- Veritas NetBackup 8.0
- 비딘

StorageGRID에서 지원되는 타사 솔루션

이러한 솔루션은 테스트를 거쳤습니다.

- Archiware를 참조하십시오
- Axis 통신
- 코너 360
- DataFrameworks
- EcoDigital DIVA 플랫폼
- Encoding.com
- Fujifilm 개체 아카이브
- GE Centricity Enterprise Archive
- 하이랜드 아쿠오
- IBM Aspera

- 마일스톤 시스템
- ONSSI
- 리치 엔진
- SilverTrak
- 소프트NAS
- 품질
- 벨라시아

제품 기능 가이드

AWS 또는 Google Cloud용 클라우드 스토리지 풀을 생성합니다

StorageGRID 오브젝트를 외부 S3 버킷으로 이동하려는 경우 클라우드 스토리지 풀을 사용할 수 있습니다. 외부 버킷은 Amazon S3(AWS) 또는 Google Cloud에 속할 수 있습니다.

필요한 것

- StorageGRID 11.6이 구성되었습니다.
- AWS 또는 Google Cloud에서 외부 S3 버킷을 이미 설정했습니다.

단계

1. Grid Manager에서 * ILM * > * 스토리지 풀 * 으로 이동합니다.
2. 페이지의 클라우드 스토리지 풀 섹션에서 * 생성 * 을 선택합니다.

   Create Cloud Storage Pool 팝업이 나타납니다.

3. 표시 이름을 입력합니다.
4. 공급자 유형 드롭다운 목록에서 * Amazon S3 * 를 선택합니다.

   이 공급자 유형은 AWS S3 또는 Google Cloud에서 작동합니다.

5. 클라우드 스토리지 풀에 사용할 S3 버킷의 URI를 입력합니다.

   다음 두 가지 형식이 허용됩니다.

   "https://host:port`

   "http://host:port`

6. S3 버킷 이름을 입력합니다.

   지정하는 이름은 S3 버킷의 이름과 정확히 일치해야 합니다. 그렇지 않으면 클라우드 스토리지 풀을 생성하지 못합니다. 클라우드 스토리지 풀을 저장한 후에는 이 값을 변경할 수 없습니다.

7. 선택적으로 액세스 키 ID와 비밀 액세스 키를 입력합니다.
8. 드롭다운에서 * 인증서 확인 안 함 * 을 선택합니다.
9. 저장 * 을 클릭합니다.

예상 결과

Amazon S3 또는 Google Cloud에 대한 클라우드 스토리지 풀이 생성되었는지 확인합니다.

Jonathan Wong이 _

Azure Blob Storage용 클라우드 스토리지 풀 생성

StorageGRID 오브젝트를 외부 Azure 컨테이너로 이동하려는 경우 클라우드 스토리지 풀을 사용할 수 있습니다.

필요한 것

- StorageGRID 11.6이 구성되었습니다.
- 외부 Azure 컨테이너를 이미 설정했습니다.

단계

1. Grid Manager에서 * ILM * > * 스토리지 풀 * 으로 이동합니다.
2. 페이지의 클라우드 스토리지 풀 섹션에서 * 생성 * 을 선택합니다.

   Create Cloud Storage Pool 팝업이 나타납니다.

3. 표시 이름을 입력합니다.
4. 공급자 유형 드롭다운 목록에서 * Azure Blob Storage * 를 선택합니다.
5. 클라우드 스토리지 풀에 사용할 S3 버킷의 URI를 입력합니다.

   다음 두 가지 형식이 허용됩니다.

   "https://host:port`

   "http://host:port`

6. Azure 컨테이너 이름을 입력합니다.

   지정하는 이름은 Azure 컨테이너 이름과 정확히 일치해야 합니다. 그렇지 않으면 클라우드 스토리지 풀을 생성하지 못합니다. 클라우드 스토리지 풀을 저장한 후에는 이 값을 변경할 수 없습니다.

7. 필요한 경우 인증을 위해 Azure 컨테이너의 관련 계정 이름 및 계정 키를 입력합니다.
8. 드롭다운에서 * 인증서 확인 안 함 * 을 선택합니다.
9. 저장 * 을 클릭합니다.

예상 결과

Azure Blob Storage용 Cloud Storage Pool이 생성되었는지 확인합니다.

Jonathan Wong이 _

백업에 클라우드 스토리지 풀 사용

ILM 규칙을 생성하여 백업을 위해 오브젝트를 클라우드 스토리지 풀로 이동할 수 있습니다.

필요한 것

- StorageGRID 11.6이 구성되었습니다.
- 외부 Azure 컨테이너를 이미 설정했습니다.

단계

1. Grid Manager에서 * ILM * > * 규칙 * > * 생성 * 으로 이동합니다.
2. 설명을 입력합니다.
3. 규칙을 트리거할 기준을 입력합니다.
4. 다음 * 을 클릭합니다.
5. 오브젝트를 스토리지 노드로 복제합니다.
6. 배치 규칙을 추가합니다.
7. 객체를 클라우드 스토리지 풀에 복제합니다
8. 다음 * 을 클릭합니다.
9. 저장 * 을 클릭합니다.

예상 결과

보존 다이어그램에 백업용 StorageGRID 및 클라우드 스토리지 풀에 로컬로 저장된 객체가 표시되는지 확인합니다.

ILM 규칙이 트리거되면 클라우드 스토리지 풀에 복사본이 존재하므로 오브젝트 복원을 수행하지 않고 로컬에서 개체를 검색할 수 있는지 확인합니다.

Jonathan Wong이 _

StorageGRID 검색 통합 서비스를 구성합니다

이 가이드는 NetApp StorageGRID 11.6 검색 통합 서비스를 Amazon OpenSearch Service 또는 사내 Elasticsearch와 구성하기 위한 자세한 지침을 제공합니다.

소개

StorageGRID는 세 가지 유형의 플랫폼 서비스를 지원합니다.

- * StorageGRID CloudMirror 복제 *. StorageGRID 버킷에서 지정된 외부 대상으로 특정 객체를 미러링합니다.
- * 알림 *. 객체에서 수행한 특정 작업에 대한 알림을 지정된 외부 Amazon SNS(Amazon Simple Notification Service)로 보내는 버킷당 이벤트 알림입니다.
- * 통합 서비스 검색 *. S3(Simple Storage Service) 개체 메타데이터를 지정된 Elasticsearch 인덱스에 전송하여 외부 서비스를 사용하여 메타데이터를 검색하거나 분석할 수 있습니다.

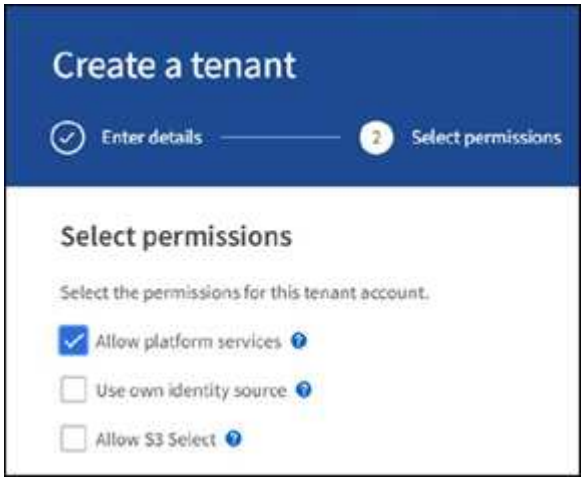
플랫폼 서비스는 테넌트 관리자 UI를 통해 S3 테넌트에서 구성합니다. 자세한 내용은 을 참조하십시오 "플랫폼 서비스 사용에 대한 고려 사항".

이 문서는 에 대한 보충 자료로 사용됩니다 "StorageGRID 11.6 테넌트 가이드" 및 에서는 검색 통합 서비스를 위한 엔드포인트 및 버킷 구성에 대한 단계별 지침과 예제를 제공합니다. 여기에 포함된 AWS(Amazon Web Services) 또는 온프레미스 Elasticsearch 설정 지침은 기본 테스트 또는 데모 전용입니다.

대상 고객은 그리드 관리자, 테넌트 관리자에 익숙해야 하며, StorageGRID 검색 통합 테스트를 위한 기본 업로드(PUT) 및 다운로드(GET) 작업을 수행하기 위해 S3 브라우저에 액세스할 수 있어야 합니다.

테넌트 생성 및 플랫폼 서비스 활성화

1. Grid Manager를 사용하여 S3 테넌트를 생성하고 표시 이름을 입력한 다음 S3 프로토콜을 선택합니다.
2. 사용 권한 페이지에서 플랫폼 서비스 허용 옵션을 선택합니다. 필요한 경우 다른 사용 권한을 선택합니다.



3. 테넌트 루트 사용자 초기 암호를 설정하거나, 격자에서 페더레이션 식별 이 설정된 경우 테넌트 계정을 구성할 루트 액세스 권한이 있는 통합 그룹을 선택합니다.
4. 루트로 로그인 을 클릭하고 버킷:버킷 생성 및 관리 를 선택합니다.

   그러면 Tenant Manager 페이지로 이동합니다.

5. Tenant Manager에서 내 액세스 키를 선택하여 나중에 테스트할 S3 액세스 키를 생성하고 다운로드합니다.

Amazon OpenSearch로 통합 서비스를 검색합니다

Amazon OpenSearch(이전의 Elasticsearch) 서비스 설정

테스트/데모용으로만 OpenSearch 서비스를 빠르고 간편하게 설정하려면 이 절차를 사용하십시오. 온-프레미스 Elasticsearch를 사용하여 검색 통합 서비스를 사용하는 경우 섹션을 참조하십시오 온-프레미스 Elasticsearch와 통합 서비스를 검색합니다.

> OpenSearch 서비스에 가입하려면 유효한 AWS 콘솔 로그인, 액세스 키, 비밀 액세스 키 및 권한이 있어야 합니다.

1. 의 지침에 따라 새 도메인을 만듭니다 "AWS OpenSearch 서비스 시작"다음 사항을 제외한 경우:
   - 4단계. 도메인 이름: sgdemo
   - 10단계. 세분화된 액세스 제어: 세분화된 액세스 제어 사용 옵션을 선택 취소합니다.
   - 12단계. 액세스 정책: 레벨 액세스 정책 구성을 선택하고 JSON 탭을 선택하여 다음 예를 사용하여 액세스 정책을 수정합니다.
     - 강조 표시된 텍스트를 사용자 고유의 AWS ID 및 액세스 관리(IAM) ID 및 사용자 이름으로 바꿉니다.
     - 강조 표시된 텍스트(IP 주소)를 AWS 콘솔에 액세스하는 데 사용한 로컬 컴퓨터의 공용 IP 주소로 바꿉니다.
     - 브라우저 탭을 에 엽니다 "https://checkip.amazonaws.com" 공용 IP를 찾습니다.

```
{

    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal":
        {"AWS": "arn:aws:iam:: nnnnnn:user/xyzabc"},
        "Action": "es:*",
        "Resource": "arn:aws:es:us-east-1:nnnnnn:domain/sgdemo/*"
        },
        {
        "Effect": "Allow",
        "Principal": {"AWS": "*"},
        "Action": [
        "es:ESHttp*"
                ],
        "Condition": {
            "IpAddress": {
                "aws:SourceIp": [ "nnn.nnn.nn.n/nn"
                    ]
                }
        },
        "Resource": "arn:aws:es:us-east-1:nnnnnn:domain/sgdemo/*"
        }
    ]
}
```

Fine-grained access control

Fine-grained access control provides numerous features to help you keep your data secure. Features include document-level security, field-level security, read-only users, and OpenSearch Dashboards/Kibana tenants. Fine-grained access control requires a master user. Learn more

☐ Enable fine-grained access control

SAML authentication for OpenSearch Dashboards/Kibana

SAML authentication lets you use your existing identity provider for single sign-on for OpenSearch Dashboards/Kibana. Learn more

Prepare SAML authentication

ⓘ To use SAML authentication, you must first enable fine-grained access control.

Amazon Cognito authentication

Enable to use Amazon Cognito authentication for OpenSearch Dashboards/Kibana. Amazon Cognito supports a variety of identity providers for username-password authentication. Learn more

☐ Enable Amazon Cognito authentication

Access policy

Access policies control whether a request is accepted or rejected when it reaches the Amazon OpenSearch Service domain. If you specify an account, user, or role in this policy, you must sign your requests. Learn more

Domain access policy

○ Only use fine-grained access control
Allow open access to the domain.

○ Do not set domain level access policy
All requests to the domain will be denied.

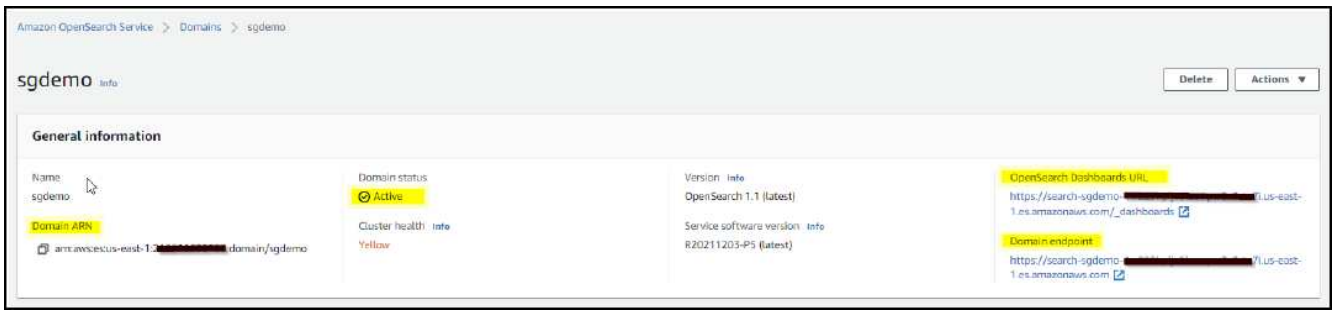
◉ Configure domain level access policy

Visual editor | JSON | Import policy

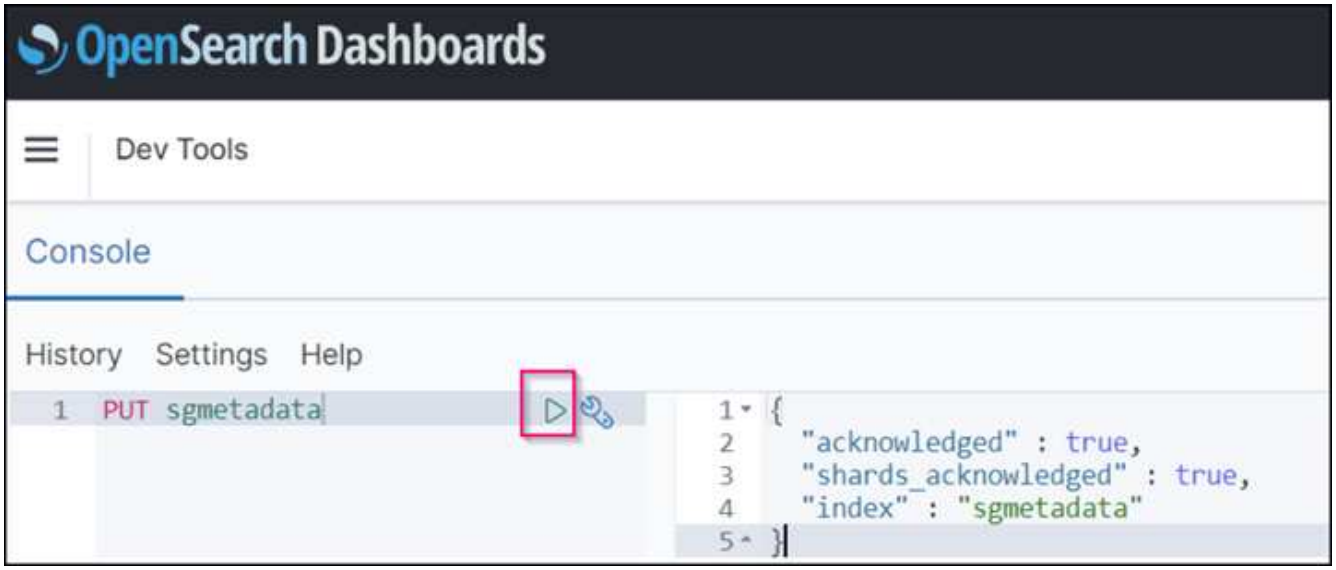
Access policy

```
3  "Statement": [
4    {
5      "Effect": "Allow",
6      "Principal": {
7        "AWS": "arn:aws:iam::2[redacted]:user/[redacted]"
8      },
9      "Action": "es:*",
10     "Resource": "arn:aws:es:us-east-1:2[redacted]:domain/sgdemo/*"
11   },
12   {
13     "Effect": "Allow",
14     "Principal": {
15       "AWS": "*"
16     },
17     "Action": [
18       "es:ESHttp*"
19     ],
20     "Condition": {
21       "IpAddress": {
22         "aws:SourceIp": [
23           "216.[redacted]/24"
24         ]
25       }
26     },
27     "Resource": "arn:aws:es:us-east-1:2[redacted]:domain/sgdemo/*"
28   }
```

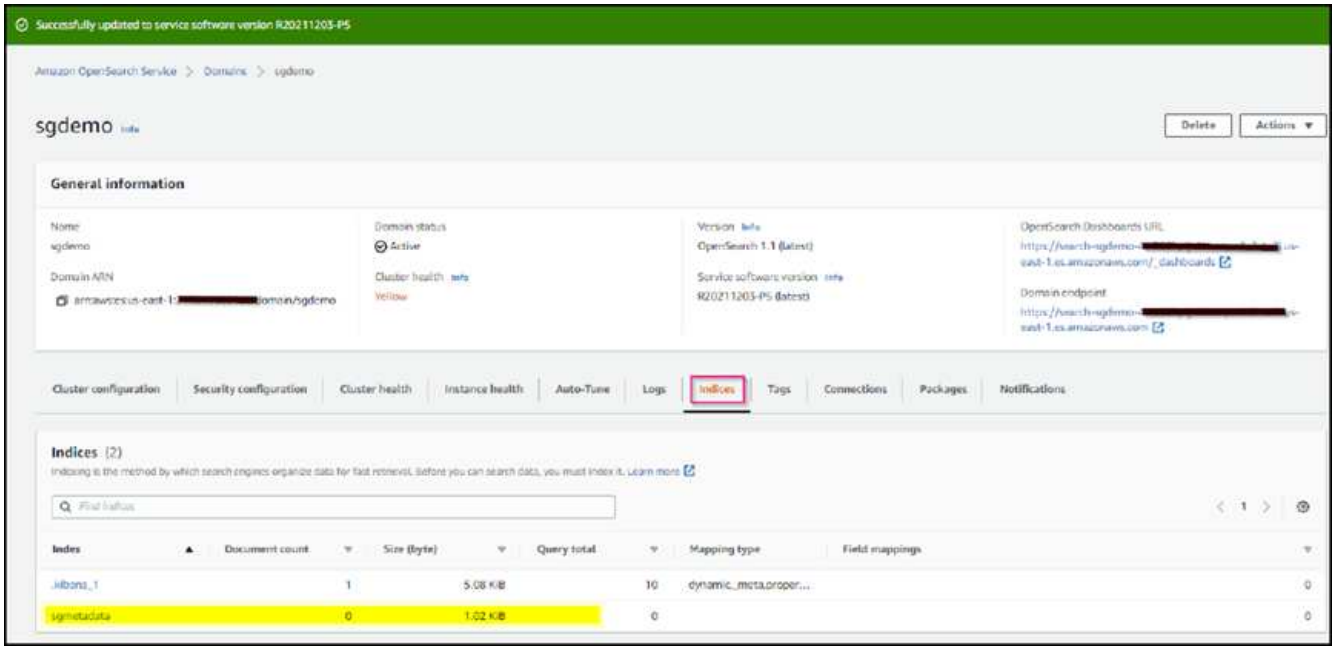
2. 도메인이 활성화될 때까지 15-20분 정도 기다립니다.



3. OpenSearch Dashboards URL 을 클릭하여 새 탭에서 도메인을 열고 대시보드에 액세스합니다. 액세스 거부 오류가 발생하면 도메인 대시보드에 액세스할 수 있도록 액세스 정책 원본 IP 주소가 컴퓨터 공용 IP로 올바르게 설정되어 있는지 확인합니다.
4. 대시보드 시작 페이지에서 직접 탐색 을 선택합니다. 메뉴에서 관리 → 개발 도구 로 이동합니다
5. 개발 도구 → 콘솔에서 StorageGRID 개체 메타데이터를 저장하기 위해 인덱스를 사용하는 'Put <index>'를 입력합니다. 다음 예에서는 인덱스 이름 'gmetadata'를 사용합니다. 작은 삼각형 기호를 클릭하여 PUT 명령을 실행합니다. 다음 예제 스크린샷과 같이 오른쪽 패널에 예상 결과가 표시됩니다.



6. 색인이 sgdomain > Indices 아래의 Amazon OpenSearch UI에서 표시되는지 확인합니다.

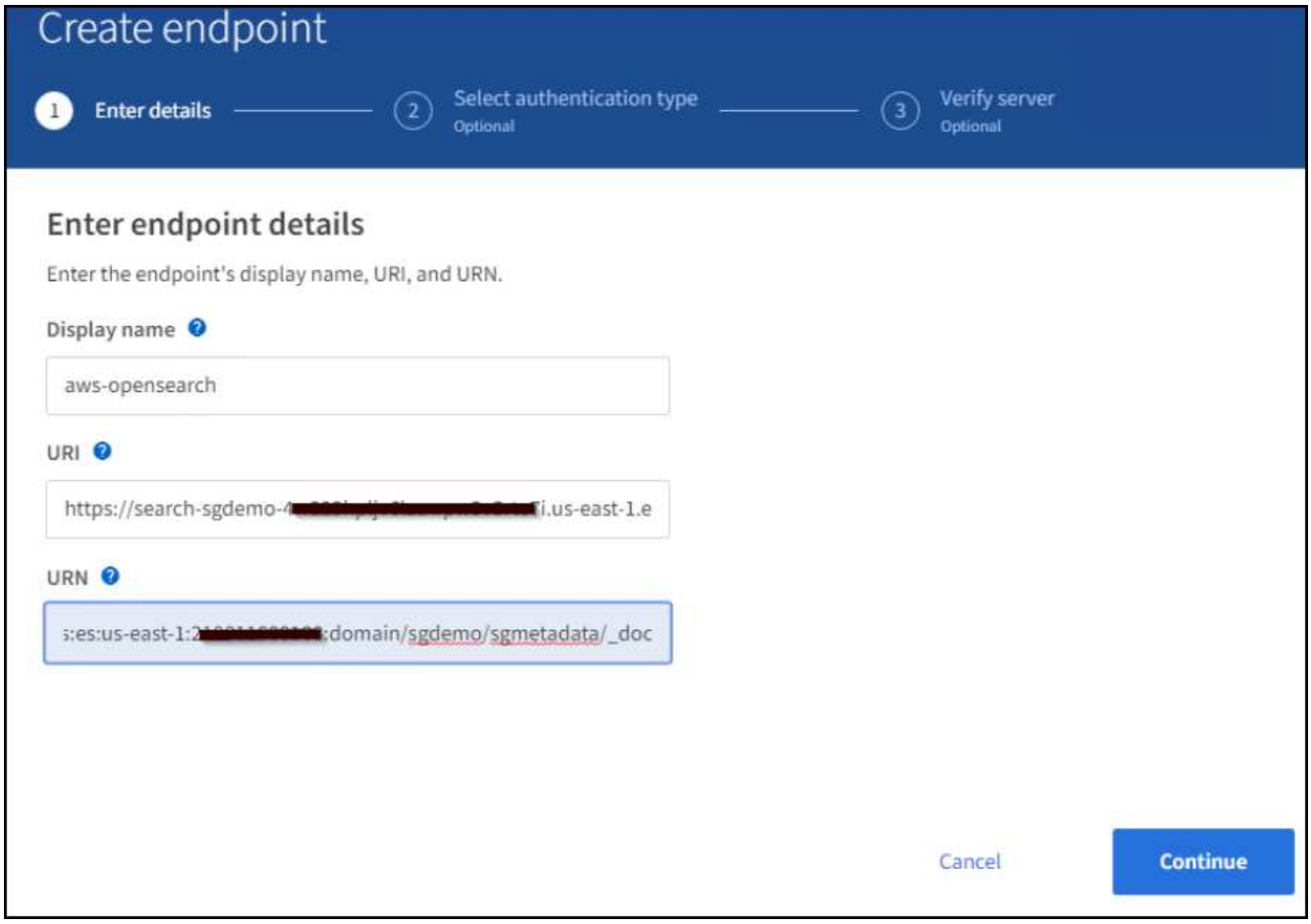


플랫폼 서비스 엔드포인트 구성

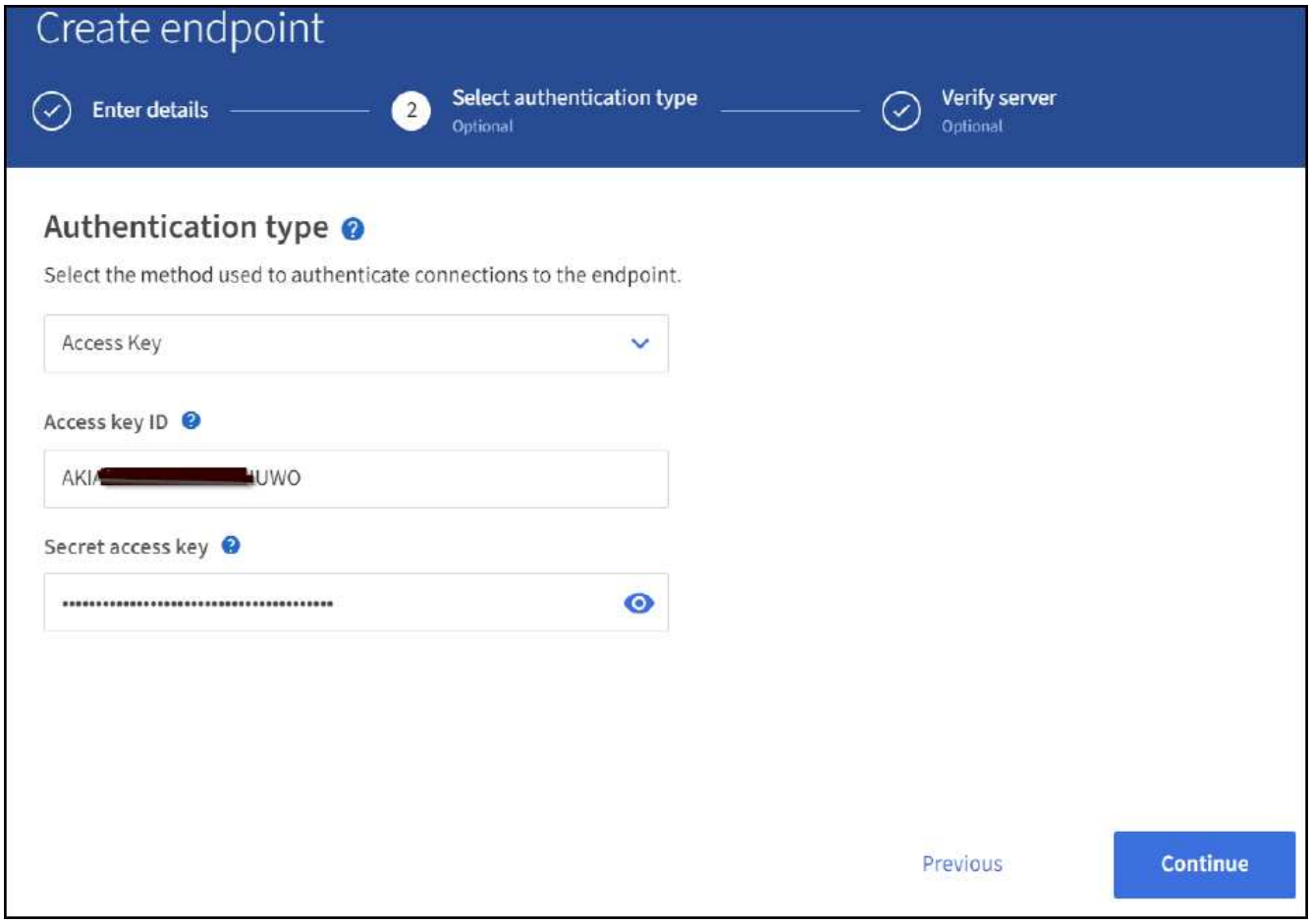
플랫폼 서비스 끝점을 구성하려면 다음 단계를 수행하십시오.

1. 테넌트 관리자 에서 스토리지(S3) > 플랫폼 서비스 엔드포인트 로 이동합니다.
2. 끝점 만들기 를 클릭하고 다음을 입력한 다음 계속 을 클릭합니다.
    - 표시 이름 예 AWS-OpenSearch
    - 예제 스크린샷의 도메인 끝점은 URI 필드의 이전 절차의 2단계 아래에 있습니다.
    - URN 필드의 이전 절차 2단계에서 사용한 ARN 도메인을 ARN의 끝에 추가하는 /<index>/_doc'를 추가한다.

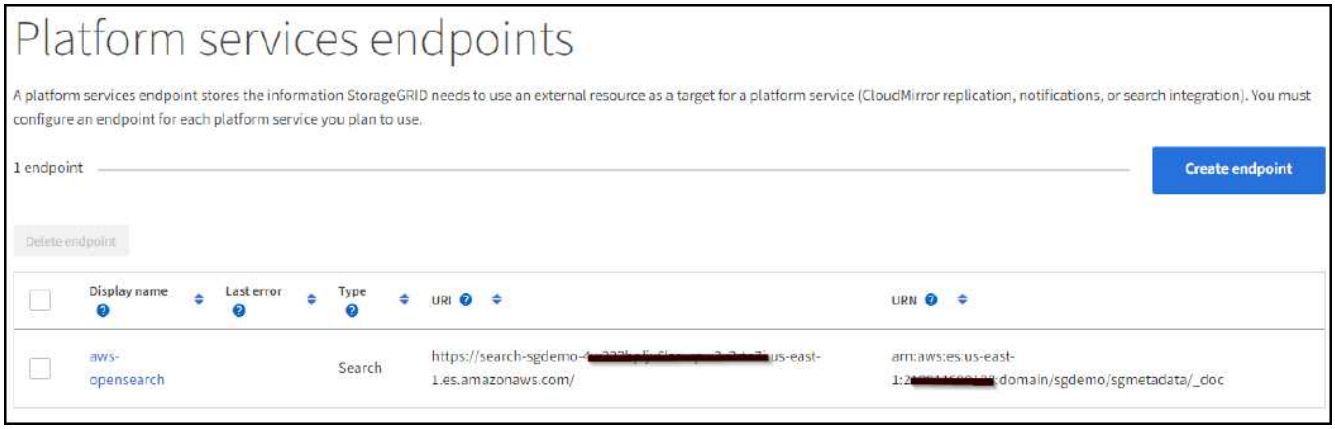
      이 예에서 URN은 'arn:aws:es:us-east-1:211234567890:domain/sgdemo/sgmedata/_doc'가 됩니다.



3. Amazon OpenSearch sgdomain에 액세스하려면 인증 유형으로 Access Key를 선택한 다음 Amazon S3 액세스 키와 암호 키를 입력합니다. 다음 페이지로 이동하려면 계속 을 클릭합니다.



4. 끝점을 확인하려면 운영 체제 CA 인증서 사용 및 끝점 테스트 및 만들기 를 선택합니다. 확인이 성공하면 다음 그림과 유사한 엔드포인트 화면이 표시됩니다. 확인이 실패하면 경로 끝에 URN에 "/<index>/_doc"가 포함되어 있고 AWS 액세스 키와 비밀 키가 올바른지 확인합니다.



온-프레미스 **Elasticsearch**와 통합 서비스를 검색합니다

온-프레미스 **Elasticsearch** 설정

이 절차는 테스트 목적으로만 Docker를 사용하여 사내 Elasticsearch 및 Kibana를 빠르게 설정하기 위한 것입니다. Elasticsearch 및 Kibana 서버가 이미 있는 경우 5단계로 이동합니다.

1. 다음 단계를 따르십시오 "Docker 설치 절차" Docker를 설치합니다. 을 사용합니다 "CentOS Docker 설치 절차" 를

클릭합니다.

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
sudo yum install docker-ce docker-ce-cli containerd.io
sudo systemctl start docker
```

- 재부팅 후 Docker를 시작하려면 다음을 입력합니다.

```
sudo systemctl enable docker
```

- VM.max_map_count 값을 262144로 설정한다.

```
sysctl -w vm.max_map_count=262144
```

- 재부팅 후 설정을 유지하려면 다음을 입력합니다.

```
echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
```

2. 를 따릅니다 "Elasticsearch 빠른 시작 가이드" Elasticsearch 및 Kibana Docker를 설치하고 실행하기 위한 자가 관리 섹션입니다. 이 예에서는 버전 8.1을 설치했습니다.

> 참고 Elasticsearch에서 만든 사용자 이름/암호 및 토큰을 아래로 하여 Kibana UI 및 StorageGRID 플랫폼 엔드포인트 인증을 시작해야 합니다.

Elasticsearch Service | Self-managed

Install and run Elasticsearch

1. Install and start Docker Desktop.
2. Run:

```
docker network create elastic
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.1.0
docker run --name es-node01 --net elastic -p 9200:9200 -p 9300:9300 -it
```

When you start Elasticsearch for the first time, the following security configuration occurs automatically:

- Certificates and keys are generated for the transport and HTTP layers.
- The Transport Layer Security (TLS) configuration settings are written to `elasticsearch.yml`.
- A password is generated for the `elastic` user.
- An enrollment token is generated for Kibana.

NOTE: You might need to scroll back a bit in the terminal to view the password and enrollment token.

3. Copy the generated password and enrollment token and save them in a secure location. These values are shown only when you start Elasticsearch for the first time. You'll use these to enroll Kibana with your Elasticsearch cluster and log in.

NOTE: If you need to reset the password for the `elastic` user or other built-in users, run the `elasticsearch-reset-password` tool. To generate new enrollment tokens for Kibana or Elasticsearch nodes, run the `elasticsearch-create-enrollment-token` tool. These tools are available in the Elasticsearch `bin` directory.

Install and run Kibana

To analyze, visualize, and manage Elasticsearch data using an intuitive UI, install Kibana.

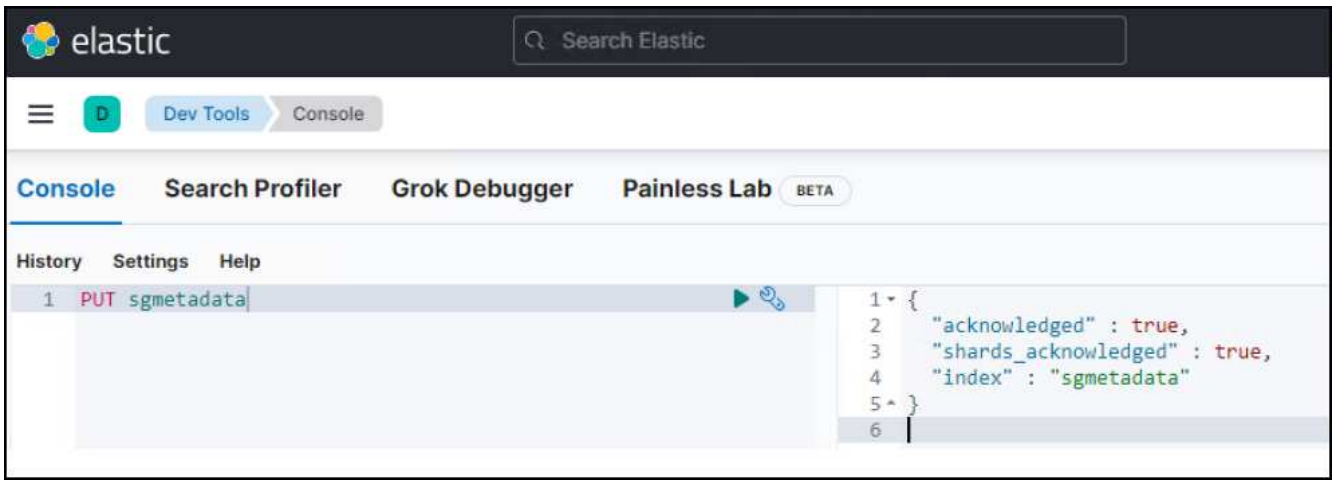
1. In a new terminal session, run:

```
docker pull docker.elastic.co/kibana/kibana:8.1.0
docker run --name kib-01 --net elastic -p 5601:5601 docker.elastic.co/ki
```

When you start Kibana, a unique link is output to your terminal.

2. To access Kibana, click the generated link in your terminal.
   a. In your browser, paste the enrollment token that you copied and click the button to connect your Kibana instance with Elasticsearch.
   b. Log in to Kibana as the `elastic` user with the password that was generated when you started Elasticsearch.

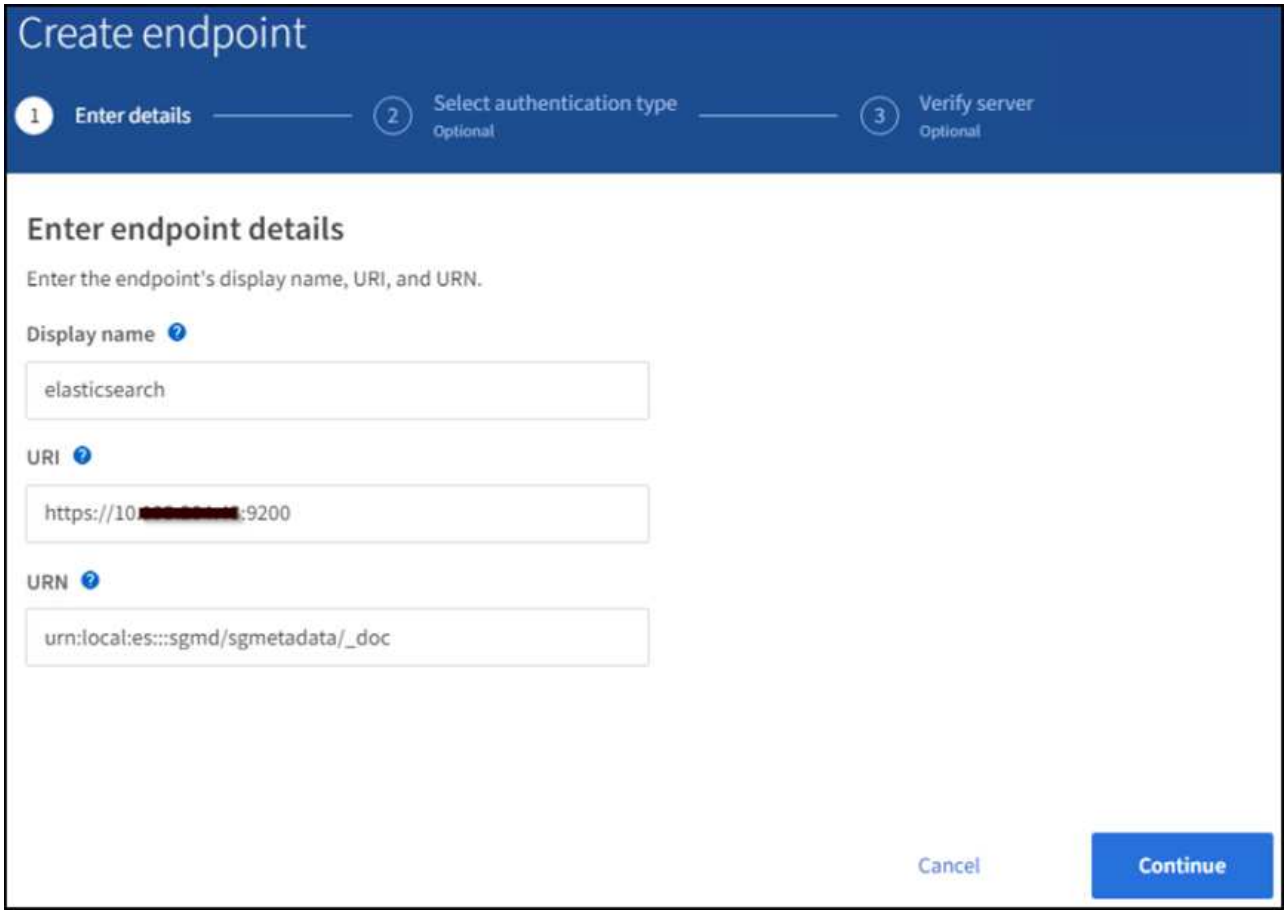
3. Kibana Docker 컨테이너가 시작되면 URL 링크 'https://0.0.0.0:5601` 가 콘솔에 표시됩니다. 0.0.0.0을 URL의 서버 IP 주소로 바꿉니다.
4. 사용자 이름 탄력성과 이전 단계에서 Elastic에 의해 생성된 암호를 사용하여 Kibana UI에 로그인합니다.
5. 처음 로그인하는 경우 대시보드 시작 페이지에서 직접 탐색 을 선택합니다. 메뉴에서 관리 > 개발 도구 를 선택합니다.
6. 개발 도구 콘솔 화면에서 StorageGRID 개체 메타데이터를 저장하기 위해 이 인덱스를 사용하는 "Put <index>"를 입력합니다. 이 예에서는 인덱스 이름 'gmetadata'를 사용합니다. 작은 삼각형 기호를 클릭하여 PUT 명령을 실행합니다. 다음 예제 스크린샷과 같이 오른쪽 패널에 예상 결과가 표시됩니다.



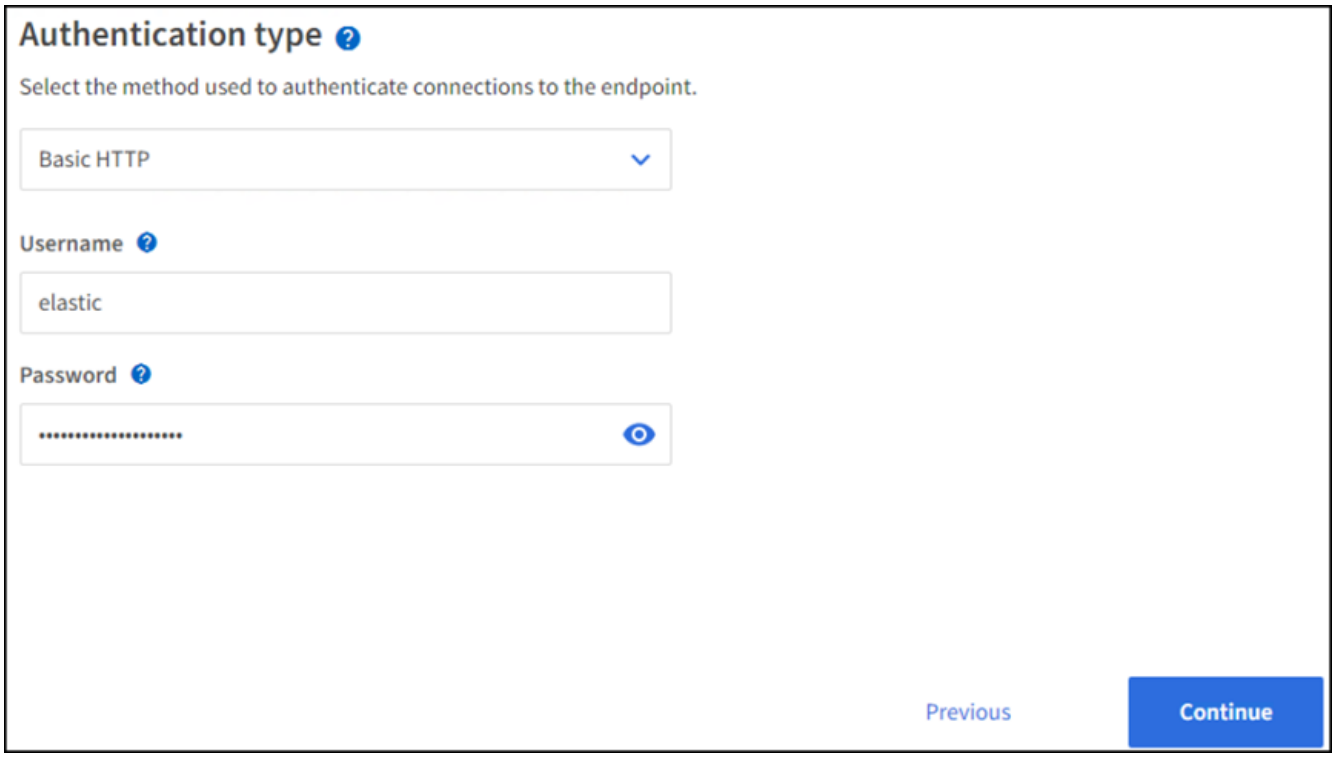
플랫폼 서비스 엔드포인트 구성

플랫폼 서비스에 대한 끝점을 구성하려면 다음 단계를 수행하십시오.

1. 테넌트 관리자에서 스토리지(S3) > 플랫폼 서비스 엔드포인트로 이동합니다
2. 끝점 만들기 를 클릭하고 다음을 입력한 다음 계속 을 클릭합니다.
   - 이름 표시 예: 탄력적인 검색
   - Uri:'https://<elasticsearch-server-ip or hostname>:9200'입니다
   - urn:'urn:<something>:es:::<some-unique-text>/<index-name>/_doc' 여기서 index-name은 Kibana 콘솔에서 사용한 이름입니다. 예: 'urn:local:es::::sgmd/sgmetadata/_doc'

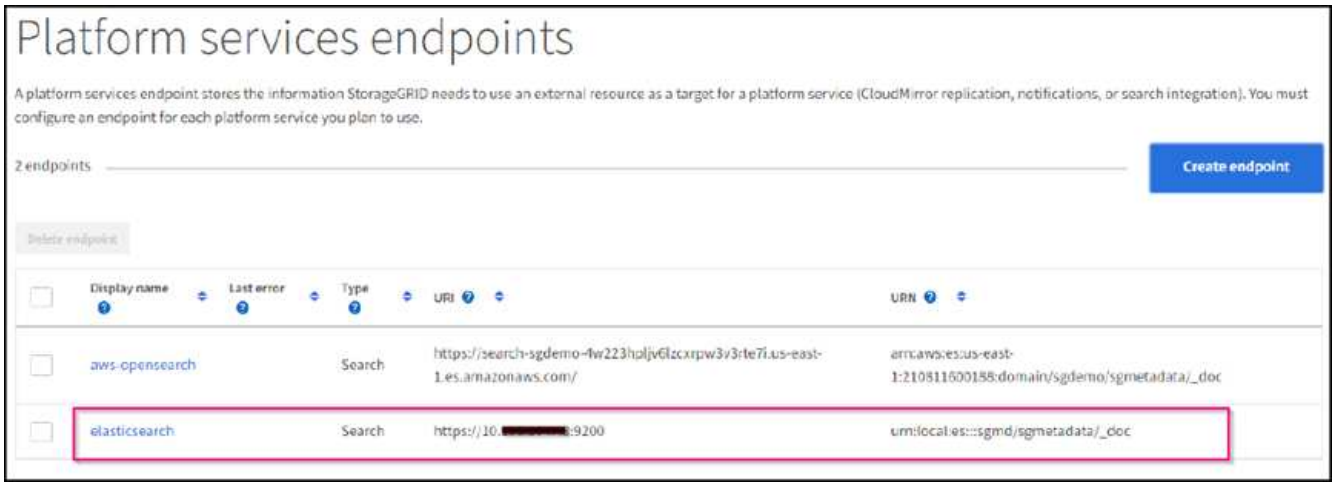


3. 인증 유형으로 기본 HTTP 를 선택하고 Elasticsearch 설치 프로세스에서 생성된 사용자 이름 'elastic'과 암호를 입력합니다. 다음 페이지로 이동하려면 계속 을 클릭합니다.



4. 인증서 확인 안 함 및 테스트 및 끝점 만들기 를 선택하여 끝점을 확인합니다. 확인이 성공하면 다음 스크린샷과

유사한 엔드포인트 화면이 표시됩니다. 확인에 실패하면 URN, URI 및 사용자 이름/암호 항목이 올바른지 확인합니다.



버킷 검색 통합 서비스 구성

플랫폼 서비스 끝점을 만든 후 다음 단계는 개체가 생성, 삭제 또는 해당 메타데이터 또는 태그가 업데이트될 때마다 개체 메타데이터를 정의된 끝점으로 보내도록 버킷 수준에서 이 서비스를 구성하는 것입니다.

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

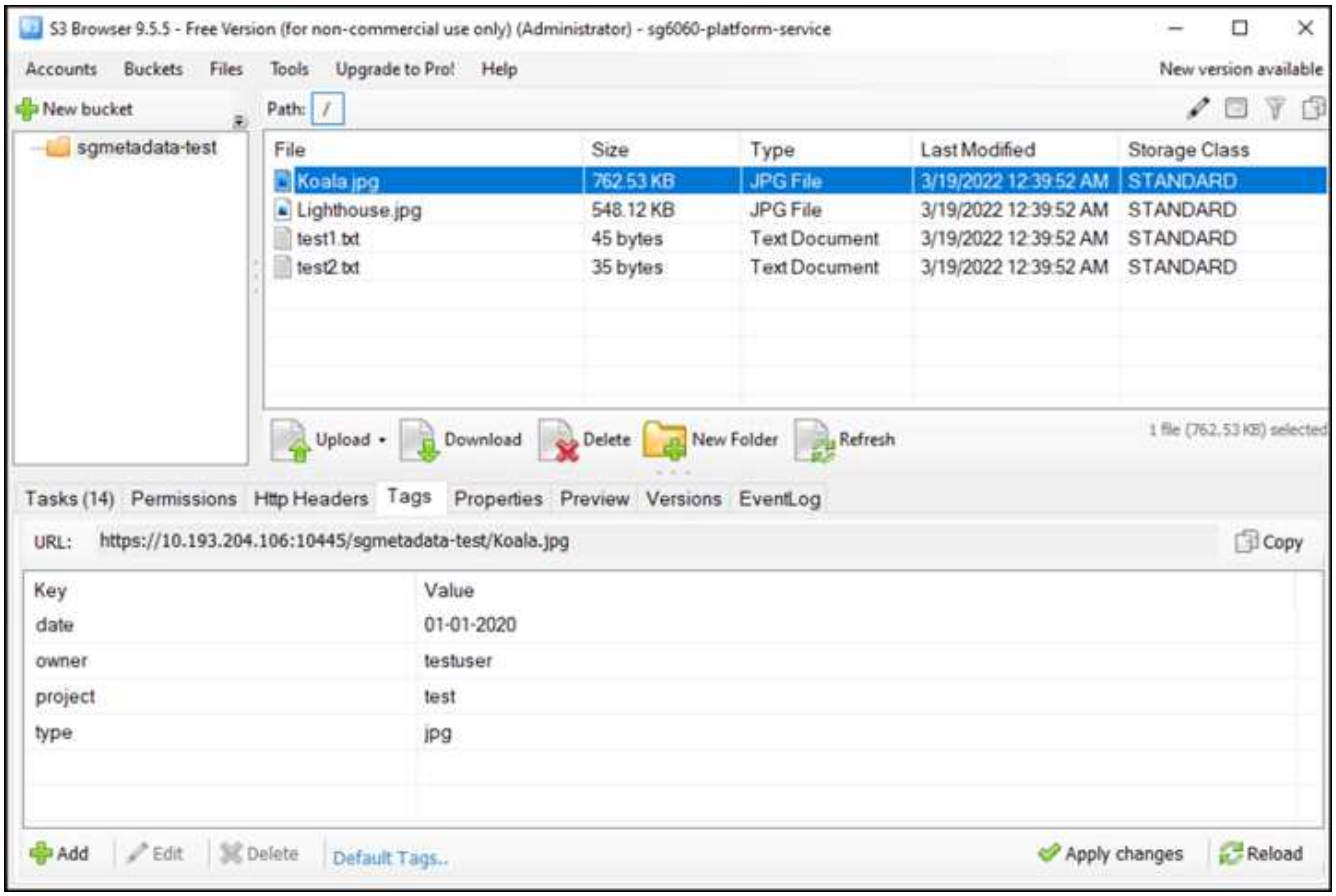
1. 테넌트 관리자 에서 스토리지(S3) > 버킷 으로 이동합니다
2. Create Bucket을 클릭하고 bucket 이름(예: 'gmetadata-test')을 입력한 후 기본 us-east-1 영역을 그대로 사용합니다.
3. 계속 > 버킷 생성 을 클릭합니다.
4. 버킷 개요 페이지를 표시하려면 버킷 이름을 클릭한 다음 플랫폼 서비스를 선택합니다.
5. 검색 통합 활성화 대화 상자를 선택합니다. 제공된 XML 상자에 이 구문을 사용하여 구성 XML을 입력합니다.

   강조 표시된 URN은 사용자가 정의한 플랫폼 서비스 끝점과 일치해야 합니다. 다른 브라우저 탭을 열어 테넌트 관리자에 액세스하고 정의된 플랫폼 서비스 끝점에서 URN을 복사할 수 있습니다.

   이 예에서는 접두어를 사용하지 않았습니다. 즉, 이 버킷의 모든 객체에 대한 메타데이터가 이전에 정의된 Elasticsearch 끝점으로 전송됩니다.

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
            <Urn> urn:local:es:::sgmd/sgmetadata/_doc</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

6. S3 브라우저를 사용하여 테넌트 액세스/암호 키를 사용하여 StorageGRID에 연결하고, 테스트 객체를 '메타데이터 테스트' 버킷에 업로드하고, 태그나 사용자 지정 메타데이터를 객체에 추가합니다.



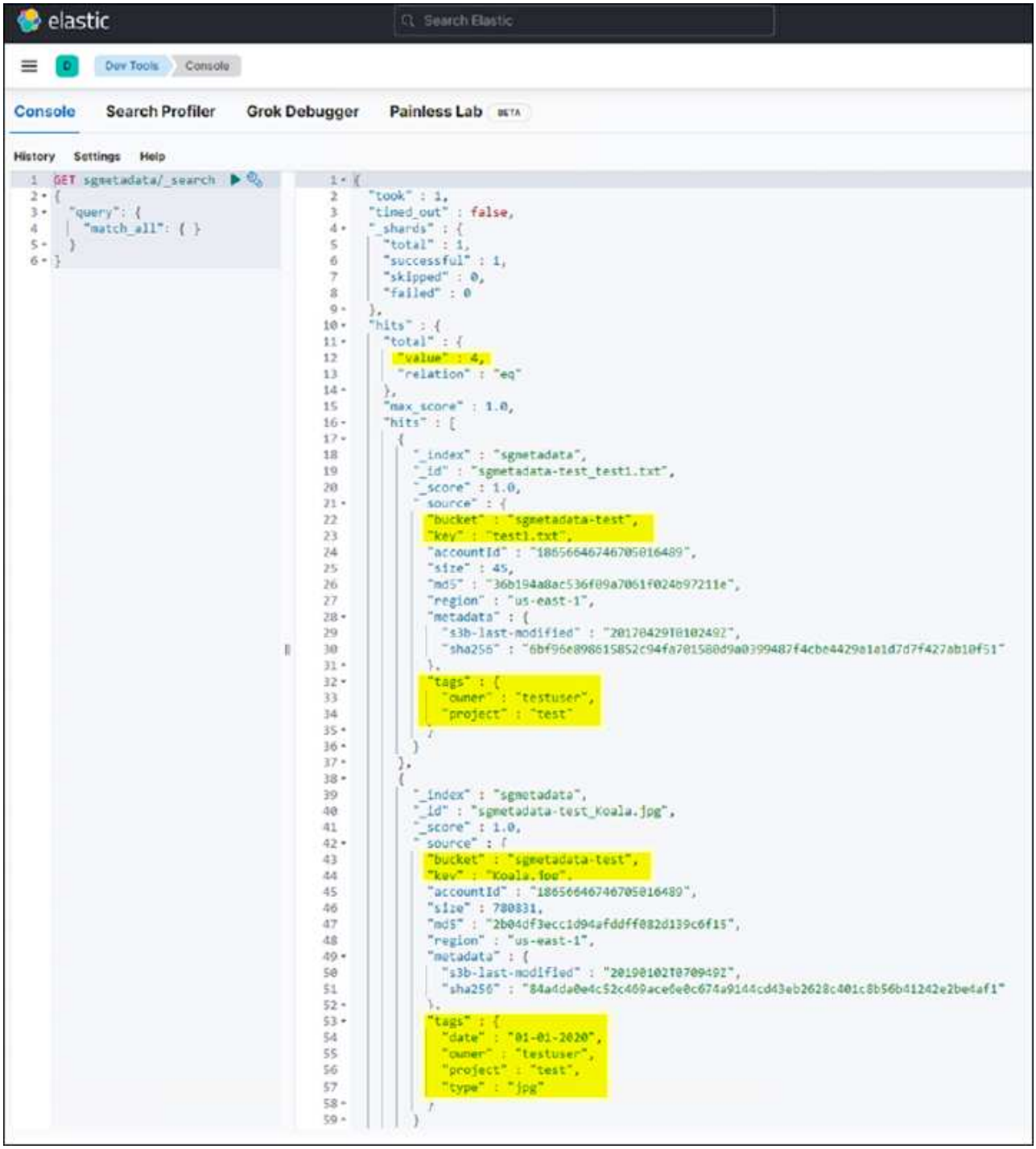
7. Kibana UI를 사용하여 오브젝트 메타데이터가 sgmetadata의 인덱스에 로드되었는지 확인합니다.

   a. 메뉴에서 관리 > 개발 도구 를 선택합니다.

   b. 왼쪽의 콘솔 패널에 샘플 쿼리를 붙여넣고 삼각형 기호를 클릭하여 실행합니다.

      다음 예제 스크린샷의 쿼리 1 예제 결과는 네 개의 레코드를 보여 줍니다. 이는 버킷의 오브젝트 수와 일치합니다.

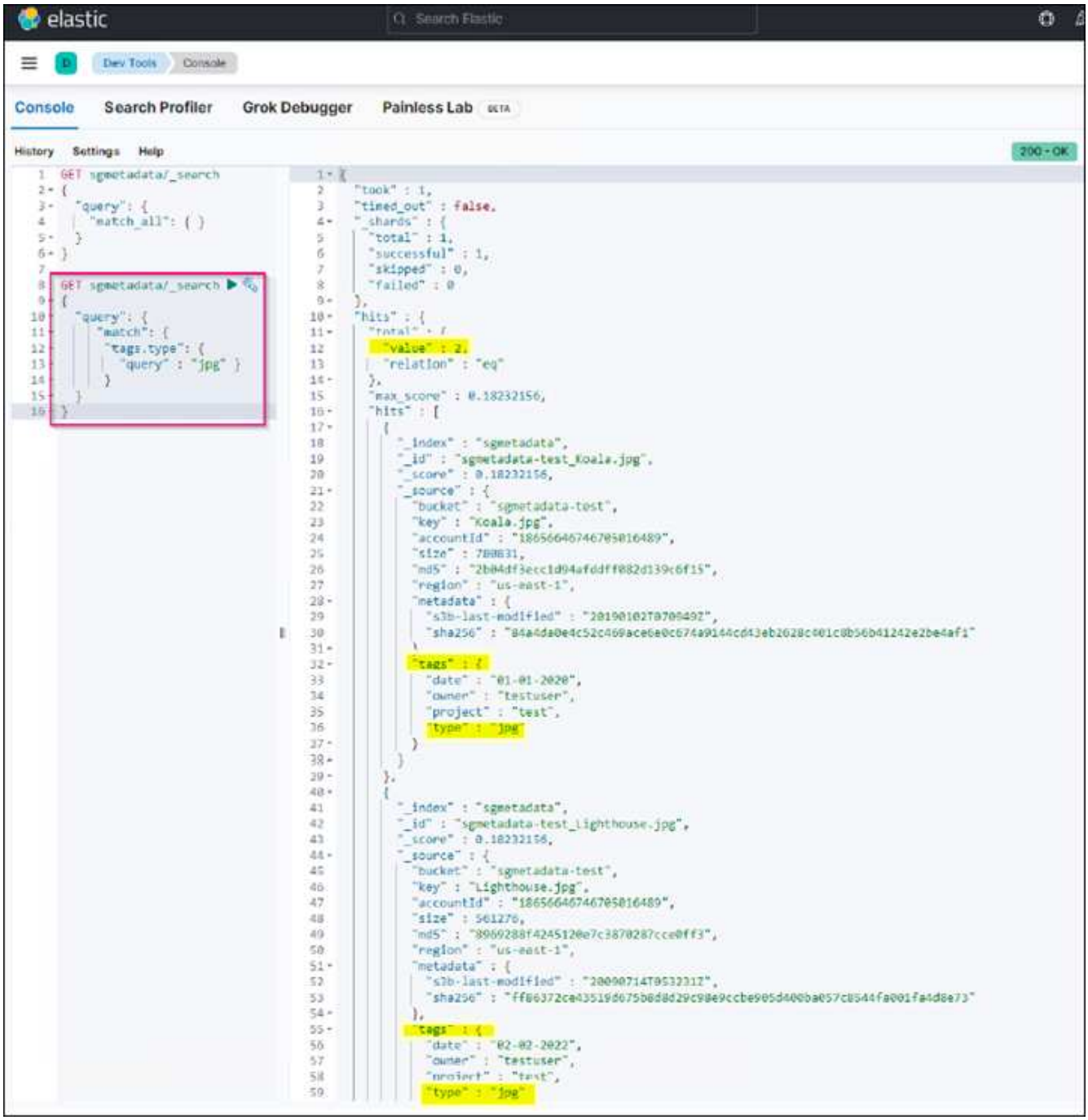
```
GET sgmetadata/_search
{
    "query": {
        "match_all": { }
}
}
```



다음 스크린샷의 쿼리 2 샘플 결과는 태그 유형 jpg의 두 레코드를 보여 줍니다.

```
GET sgmetadata/_search
{
    "query": {
        "match": {
            "tags.type": {
                "query" : "jpg" }
                }
            }
}
```

+



추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대해 자세히 알아보려면 다음 문서 및/또는 웹 사이트를 검토하십시오.

- "플랫폼 서비스란 무엇입니까"
- "StorageGRID 11.6 문서"

안젤라 청 _ 에 의해

노드 클론

노드 클론 고려 사항 및 성능

노드 클론 고려 사항

노드 클론은 기술 업데이트, 용량 증가 또는 StorageGRID 시스템 성능 향상을 위해 기존 어플라이언스 노드를 빠르게 교체할 수 있는 방법이 될 수 있습니다. 노드 클론은 KMS를 사용하여 노드 암호화로 변환하거나 스토리지 노드를 DDP8에서 DDP16으로 변경하는 경우에도 유용합니다.

- 소스 노드의 사용된 용량은 클론 프로세스를 완료하는 데 필요한 시간과 관련이 없습니다. 노드 클론은 노드의 여유 공간을 포함하는 노드의 전체 복사본입니다.
- 소스 및 대상 장비는 동일한 PGE 버전이어야 합니다
- 대상 노드의 용량은 항상 소스보다 커야 합니다
  - 새 대상 어플라이언스의 드라이브 크기가 소스보다 큰지 확인하십시오
  - 대상 어플라이언스의 크기가 동일한 드라이브가 DDP8에 대해 구성된 경우 DDP16의 대상을 구성할 수 있습니다. 소스가 DDP16에 대해 이미 구성되어 있으면 노드 클론을 사용할 수 없습니다.
  - SG5660 또는 SG5760 어플라이언스에서 SG6060 어플라이언스로 이동하는 경우 SG5x60은 60개의 대용량 드라이브를 지원하며 SG6060은 58만 지원합니다.
- 노드 클론 프로세스를 수행하려면 클론 생성 프로세스 동안 소스 노드가 그리드에 대해 오프라인 상태여야 합니다. 이 시간 동안 추가 노드가 오프라인이 되면 클라이언트 서비스에 영향을 줄 수 있습니다.
- 스토리지 노드는 15일 동안만 오프라인 상태가 될 수 있습니다. 복제 프로세스 추정치가 15일에 가깝거나 15일을 초과할 경우 확장 및 서비스 해제 절차를 사용하십시오.
- 확장 쉘프가 있는 SG6060의 경우, 전체 클론 기간을 가져오려면 기본 어플라이언스 시간의 시간에 올바른 쉘프 드라이브 크기에 대한 시간을 추가해야 합니다.
- 타겟 스토리지 어플라이언스의 볼륨 수는 소스 노드의 볼륨 수보다 크거나 같아야 합니다. 오브젝트 저장소 볼륨(rangedb)이 16개인 소스 노드를 12개의 오브젝트 저장소 볼륨이 있는 타겟 스토리지 어플라이언스에 클론 복제할 수 없습니다. 타겟 어플라이언스에 소스 노드보다 용량이 더 큰 경우에도 마찬가지입니다. 오브젝트 저장소 볼륨이 12개뿐인 SGF6112 스토리지 어플라이언스를 제외하고 대부분의 스토리지 어플라이언스에는 16개의 오브젝트 저장소 볼륨이 있습니다. 예를 들어, SG5760에서 SGF6112로 클론을 생성할 수 없습니다.

노드 클론 성능 추정치

다음 표에는 노드 클론 기간에 대해 계산된 추정치가 나와 있습니다. 조건이 다양하므로 * BOLD * 의 항목은 노드 다운에 대해 15일 제한을 초과할 위험이 있습니다.

DDP8

SG5612 → 임의

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	1일	2일	2.5일	3일	4일	4.5일
25GB	1일	2일	2.5일	3일	4일	4.5일

SG5712 → 임의

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	1일	2일	2.5일	3일	4일	4.5일
25GB	1일	2일	2.5일	3일	4일	4.5일

SG5660 → **SG5760**

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	3일	6일	7일	8.5일	11.5일	• 13일 *
25GB	3일	6일	7일	8.5일	11.5일	• 13일 *

SG5660 → **SG6060**

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	2.5일	4.5일	5.5일	6.5일	9일	10일
25GB	2일	4일	5일	6일	8일	9일

SG5760 → **SG5760**

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	3일	6일	7일	8.5일	11.5일	• 13일 *

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
25GB	3일	6일	7일	8.5일	11.5일	• 13일 *

SG5760 → SG6060

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	2.5일	4.5일	5.5일	6.5일	9일	10일
25GB	1.5일	3일	3.5일	4.5일	6일	6.5일

SG6060 → SG6060

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	2.5일	4.5일	5.5일	6.5일	8.5일	9.5일
25GB	1.5일	3일	3.5일	4일	5.5일	6일

DDP16을 참조하십시오

SG5760 → SG5760

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	3.5일	6.5일	8일	9.5일	12.5일	• 14일 *
25GB	3.5일	6.5일	8일	9.5일	12.5일	• 14일 *

SG5760 → SG6060

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	2.5일	5일	6일	7.5일	10일	11일
25GB	2일	3.5일	4일	5일	6.5일	7일

SG6060 → SG6060

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	3.5일	5일	6일	7일	9.5일	10.5일
25GB	2일	3일	4일	4.5일	6일	7일

확장 쉘프(소스 어플라이언스의 각 쉘프에 대해 **SG6060** 위에 추가)

네트워크 인터페이스 속도	**4TB** 드라이브 크기	**8TB** 드라이브 크기	**10TB** 드라이브 크기	**12TB** 드라이브 크기	**16TB** 드라이브 크기	**18TB** 드라이브 크기
10GB	3.5일	5일	6일	7일	9.5일	10.5일
25GB	2일	3일	4일	4.5일	6일	7일

_ 아론 클라인 _

포트 재매핑 사용 방법

여러 가지 이유로 인해 수신 포트 또는 아웃바운드 포트를 다시 매핑해야 할 수 있습니다. 레거시 CLB 로드 밸런서 서비스에서 현재 nginx 서비스 로드 밸런서 끝점으로 이동하고 동일한 포트를 유지하여 클라이언트에 대한 영향을 줄이거나, 관리 노드 클라이언트 네트워크에서 클라이언트 S3에 포트 443을 사용하거나, 방화벽 제한 사항에 대해 사용할 수 있습니다.

포트 재맵을 사용하여 **CLB**에서 **NGINX**로 **S3** 클라이언트를 마이그레이션합니다

StorageGRID 11.3 이전의 릴리스에서는 게이트웨이 노드에 포함된 로드 밸런서 서비스가 CLB(연결 로드 밸런서)입니다. StorageGRID 11.3에서 NetApp은 HTTP(s) 트래픽 로드 밸런싱을 위한 기능이 풍부한 통합 솔루션으로 NGINX 서비스를 도입했습니다. CLB 서비스는 StorageGRID의 현재 릴리스에서 계속 사용할 수 있으므로 새 로드 밸런서 엔드포인트 구성에서 포트 8082를 다시 사용할 수 없습니다. 이 문제를 해결하려면 8082 인바운드 포트를 10443으로 다시 매핑합니다. 이렇게 하면 게이트웨이 리디렉션의 포트 8082에 들어오는 모든 HTTPS 요청이 포트 10443으로 리디렉션되고 CLB 서비스를 우회하여 NGINX 서비스에 연결됩니다. 다음 지침은 VMware에 대한 지침이지만 port_remap 기능은 모든 설치 방법을 사용하며 베어 메탈 배포 및 어플라이언스에 유사한 프로세스를 사용할 수 있습니다.

VMware 가상 머신 게이트웨이 노드 구축

다음 단계는 StorageGRID OVF(Open Virtualization Format)를 사용하여 VMware vSphere 7에서 VM으로 게이트웨이 노드를 구축하는 StorageGRID 구축 단계입니다. 이 프로세스에서는 VM을 소멸적으로 제거하고 동일한 이름 및 구성으로 VM을 다시 배포해야 합니다. VM의 전원을 켜기 전에 vApp 속성을 변경하여 포트를 다시 매핑한 다음 VM의 전원을 켜고 노드 복구 프로세스를 따르십시오.

필수 구성 요소

- StorageGRID 11.3 이상을 실행하고 있습니다
- 설치된 StorageGRID 버전 VMware 설치 파일을 다운로드하여 액세스할 수 있습니다.
- VM의 전원을 켜거나 끄고, VM 및 vApp의 설정을 변경하고, vCenter에서 VM을 제거하고, OVF로 VM을 구축할 수 있는 권한이 있는 vCenter 계정이 있습니다.
- 로드 밸런서 끝점을 만들었습니다
  - 포트가 원하는 리디렉션 포트로 구성되어 있습니다
  - 엔드포인트 SSL 인증서는 Configuration/Server Certificates/Object Storage API Service Endpoints Server Certificate의 CLB 서비스에 설치된 것과 동일하며, 그렇지 않은 경우 클라이언트는 인증서 변경을 수락할 수 있습니다.

```
If your existing certificate is self-signed, you cannot reuse it
in the new endpoint. You must generate a new self-signed
certificate when creating the endpoint and configure the clients
to accept the new certificate.
```

첫 번째 게이트웨이 노드를 제거합니다

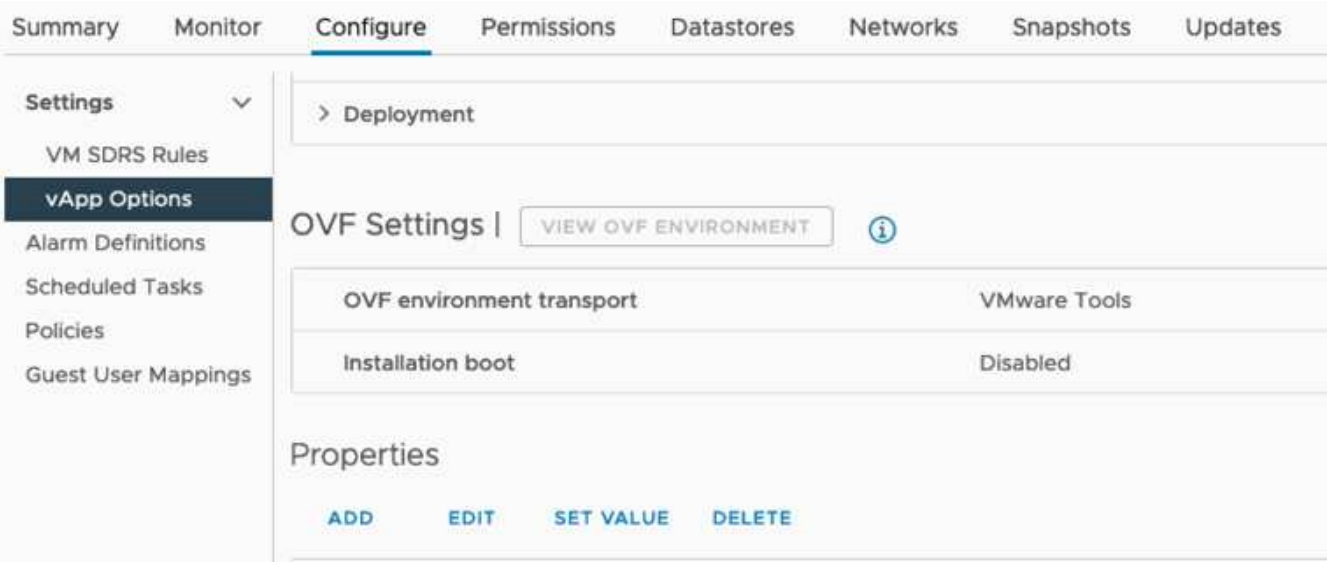
첫 번째 게이트웨이 노드를 제거하려면 다음 단계를 수행하십시오.

1. 그리드에 둘 이상의 노드가 있는 경우 시작할 게이트웨이 노드를 선택합니다.
2. 해당되는 경우 모든 DNS 라운드 로빈 엔터티 또는 로드 밸런서 풀에서 노드 IP를 제거합니다.
3. TTL(Time-to-Live)과 열려 있는 세션이 만료될 때까지 기다립니다.
4. VM 노드의 전원을 끕니다.
5. 디스크에서 VM 노드를 제거합니다.

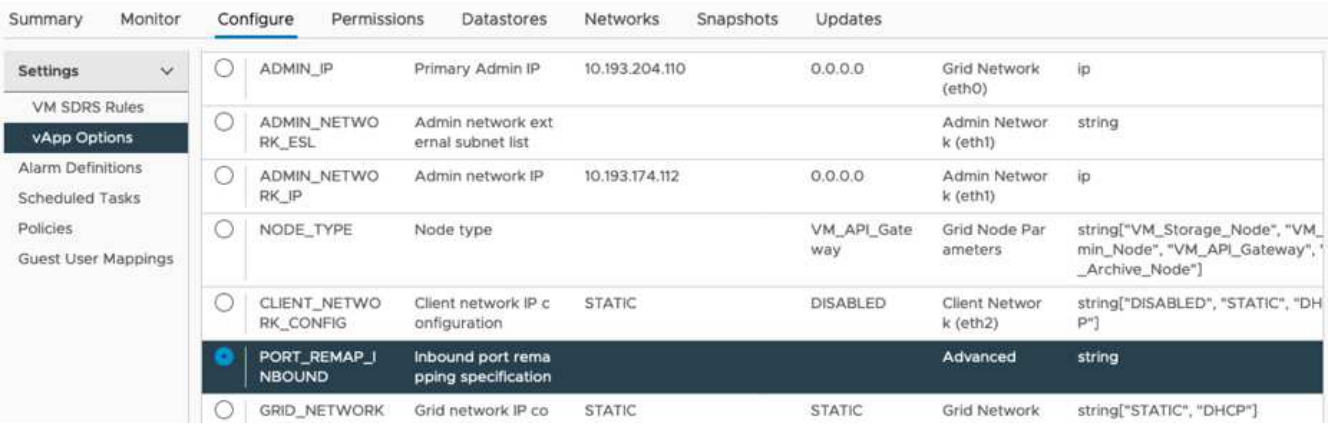
교체용 게이트웨이 노드를 배포합니다

교체 게이트웨이 노드를 배포하려면 다음 단계를 수행하십시오.

1. OVF에서 새 VM을 구축하고 지원 사이트에서 다운로드한 설치 패키지에서 .ovf, .mf 및 .vmdk 파일을 선택합니다.
   - vsphere - gateway.mf
   - vsphere - gateway.ovf
   - NetApp-SG-11.4.0-20200721.1338.d3969b3.vmdk
2. VM이 구축된 후 VM 목록에서 해당 VM을 선택하고 Configure 탭 vApp Options를 선택합니다.



3. 속성 섹션으로 스크롤하고 port_remap_inbound 속성을 선택합니다

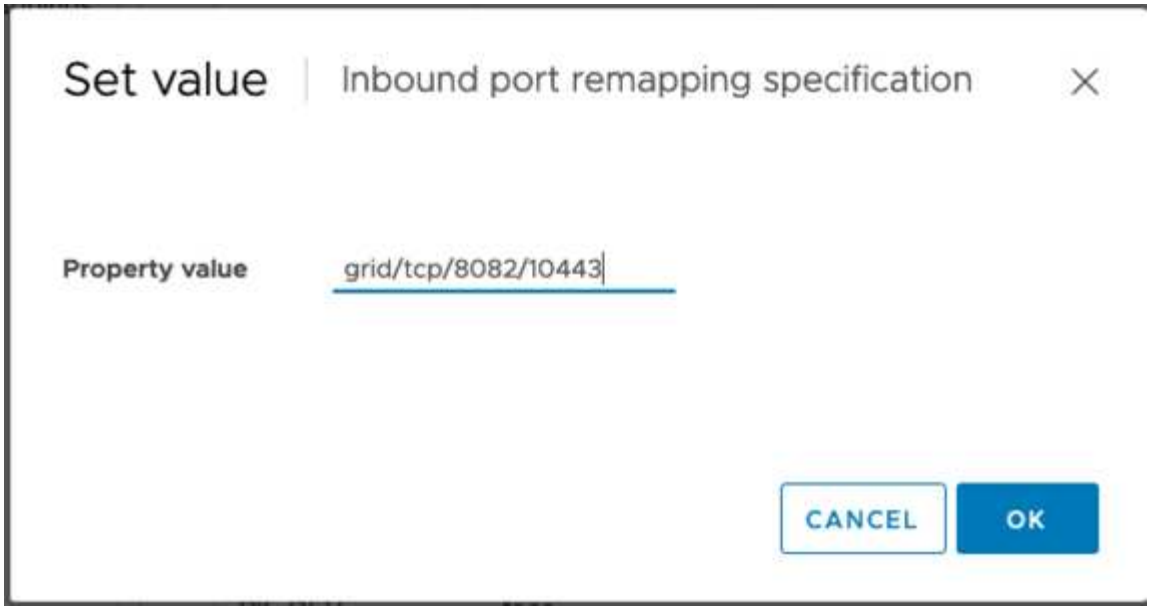


4. 속성 목록 맨 위로 스크롤하여 편집 을 클릭합니다

5. 유형 탭을 선택하고 사용자 구성 가능 확인란이 선택되어 있는지 확인한 다음 저장을 클릭합니다.

6. 속성 목록 상단에서 "port_remap_inbound" 속성을 선택한 상태에서 값 설정 을 클릭합니다.

7. 속성 값 필드에 네트워크(그리드, 관리자 또는 클라이언트), TCP, 원래 포트(8082) 및 새 포트(10443)를 입력합니다(아래 그림에 표시된 대로 각 값 사이에 “/”가 있음).

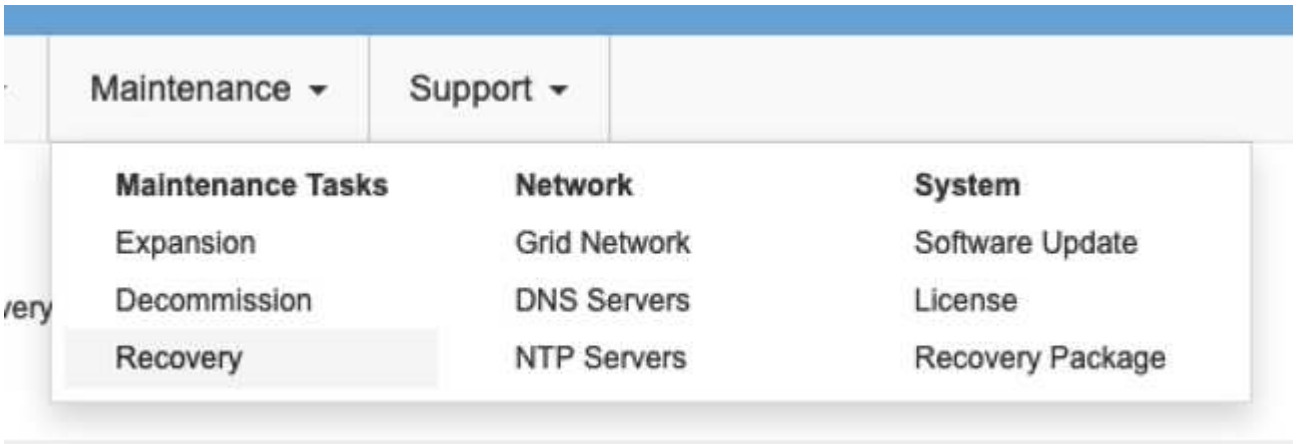


8. 여러 네트워크를 사용하는 경우 네트워크 문자열을 구분하려면 쉼표(,)를 사용합니다(예: GRID/TCP/8082/10443, admin/TCP/8082/10443, client/TCP/8082/10443)

게이트웨이 노드를 복구합니다

게이트웨이 노드를 복구하려면 다음 단계를 수행하십시오.

1. Grid Management UI의 Maintenance/Recovery 섹션으로 이동합니다.



2. VM 노드의 전원을 켜고 Grid Management UI의 Maintenance/Recovery Pending Nodes 섹션에 노드가 나타날 때까지 기다립니다.


```
For information and directions for node recovery, see the
https://docs.netapp.com/sgws-114/topic/com.netapp.doc.sg-
maint/GUID-7E22B1B9-4169-4800-8727-75F25FC0FFB1.html[Recovery
and Maintenance guide]
```

3. 노드가 복구된 후에는 해당하는 경우 모든 DNS 라운드 로빈 엔터티 또는 로드 밸런서 풀에 IP를 포함할 수 있습니다.

   이제 포트 8082의 모든 HTTPS 세션이 포트 10443으로 이동합니다

관리 노드에서 클라이언트 **S3** 액세스를 위한 포트 **443**을 다시 매핑합니다

관리 노드 또는 관리 노드가 포함된 HA 그룹에 대한 StorageGRID 시스템의 기본 구성은 포트 443 및 80을 관리 및 테넌트 관리자 UI용으로 예약하기 위한 것이며 로드 밸런서 끝점에 사용할 수 없습니다. 이에 대한 해결 방법은 포트 재매핑 기능을 사용하고 인바운드 포트 443을 로드 밸런서 끝점으로 구성할 새 포트로 리디렉션하는 것입니다. 이 작업이 완료되면 클라이언트 S3 트래픽에서 포트 443을 사용할 수 있고, 그리드 관리 UI는 포트 8443을 통해서만 액세스할 수 있으며, 테넌트 관리 UI는 포트 9443에서만 액세스할 수 있습니다. 재매핑 포트 기능은 노드 설치 시에만 구성할 수 있습니다. 활성 노드의 포트 재맵을 그리드에서 구현하려면 미리 설치된 상태로 재설정해야 합니다. 이 작업은 구성을 변경한 후 노드 복구를 포함하는 제거 절차입니다.

백업 로그 및 데이터베이스

관리 노드에는 속성, 경보 및 경고에 대한 내역 정보뿐만 아니라 감사 로그, Prometheus 메트릭이 포함됩니다. admin 노드가 여러 개인 경우 이 데이터의 복사본이 여러 개 있습니다. 그리드에 admin 노드가 여러 개 없는 경우, 이 프로세스가 끝날 때 노드를 복구한 후에 이 데이터를 보존하여 복원해야 합니다. 그리드에 다른 관리 노드가 있는 경우 복구 프로세스 중에 해당 노드의 데이터를 복사할 수 있습니다. 그리드에 다른 관리 노드가 없는 경우 노드를 삭제하기 전에 다음 지침에 따라 데이터를 복사할 수 있습니다.

감사 로그를 복사합니다

1. 관리자 노드에 로그인합니다.
   a. 다음 명령을 입력합니다. `ssh admin@grid_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.
   e. SSH 에이전트에 SSH 개인 키를 추가합니다. 입력: `ssh-add`

f. 에 나열된 SSH 액세스 암호를 입력합니다 `Passwords.txt` 파일.

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 디렉토리를 생성하여 모든 감사 로그 파일을 별도의 그리드 노드의 임시 위치에 복사합니다. use_storage_node_01_:
   a. `ssh admin@storage_node_01_IP`
   b. `mkdir -p /var/local/tmp/saved-audit-logs`
3. 관리 노드로 돌아가서 AMS 서비스를 중지하여 새 로그 파일을 생성하지 않도록 합니다. `service ams stop`
4. audit.log 파일을 복구된 관리 노드에 복사할 때 기존 파일을 덮어쓰지 않도록 파일 이름을 바꿉니다.
   a. audit.log 이름을 yyyy-mm-dd.txt.1과 같이 번호가 지정된 고유한 파일 이름으로 바꿉니다. 예를 들어 감사 로그 파일의 이름을 2015-10-25.txt.1로 바꿀 수 있습니다

```
cd /var/local/audit/export
ls -l
mv audit.log 2015-10-25.txt.1
```

5. AMS 서비스를 다시 시작합니다. `service ams start`
6. 모든 감사 로그 파일 복사: `scp * admin@storage_node_01_IP:/var/local/tmp/saved-audit-logs`

Prometheus 데이터를 복사합니다

> Prometheus 데이터베이스를 복사하는 데 1시간 이상이 걸릴 수 있습니다. 일부 그리드 관리자 기능은 관리 노드에서 서비스가 중지되는 동안 사용할 수 없습니다.

1. 디렉토리를 생성하여 Prometheus 데이터를 별도의 그리드 노드의 임시 위치에 복사합니다. 다시 한 번 사용자_storage_node_01_입니다.
   a. 스토리지 노드에 로그인합니다.
      i. 다음 명령을 입력합니다. `ssh admin@storage_node_01_IP`
      ii. 에 나열된 암호를 입력합니다 `Passwords.txt` 파일.
      iii. mkdir -p /var/local/tmp/Prometheus'입니다
2. 관리자 노드에 로그인합니다.
   a. 다음 명령을 입력합니다. `ssh admin@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.
   e. SSH 에이전트에 SSH 개인 키를 추가합니다. 입력: `ssh-add`

f. 에 나열된 SSH 액세스 암호를 입력합니다 `Passwords.txt` 파일.

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

3. 관리 노드에서 Prometheus 서비스를 중지합니다. `service prometheus stop`

   a. 소스 관리 노드에서 스토리지 노드 백업 위치로 Prometheus 데이터베이스를 복사합니다. 노드: `/rsync -azh --stats "/var/local/mysql_ibdata/prometheus/data" " storage_node_01_IP:/var/local/tmp/prometheus/"`

4. 소스 관리 노드에서 Prometheus 서비스를 다시 시작합니다.`service prometheus start`

내역 정보 백업

내역 정보는 MySQL 데이터베이스에 저장됩니다. 데이터베이스 복사본을 덤프하려면 NetApp의 사용자 및 암호가 필요합니다. 그리드에 다른 관리 노드가 있는 경우 이 단계는 필요하지 않으며 복구 프로세스 중에 나머지 관리 노드에서 데이터베이스를 복제할 수 있습니다.

1. 관리자 노드에 로그인합니다.

   a. 다음 명령을 입력합니다. `ssh admin@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.

   e. SSH 에이전트에 SSH 개인 키를 추가합니다. 입력: `ssh-add`

   f. 에 나열된 SSH 액세스 암호를 입력합니다 `Passwords.txt` 파일.

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 관리자 노드에서 StorageGRID 서비스를 중지하고 NTP 및 MySQL을 시작합니다

   a. 모든 서비스 중지: `service servermanager stop`

   b. NTP 서비스 다시 시작: `service ntp start`.. MySQL 서비스를 다시 시작합니다. `service mysql start`

3. mi 데이터베이스를 /var/local/tmp에 덤프합니다

   a. 다음 명령을 입력합니다. `mysqldump -u username -p password mi > /var/local/tmp/mysql-mi.sql`

4. MySQL dump 파일을 대체 노드에 복사합니다. *storage_node_01*을 사용합니다. *`scp /var/local/tmp/mysql-mi.sql _storage_node_01_IP:/var/local/tmp/mysql-mi.sql`*

   a. 다른 서버에 대한 암호 없는 액세스가 더 이상 필요하지 않으면 SSH 에이전트에서 개인 키를 제거합니다. 입력: `ssh-add -D`

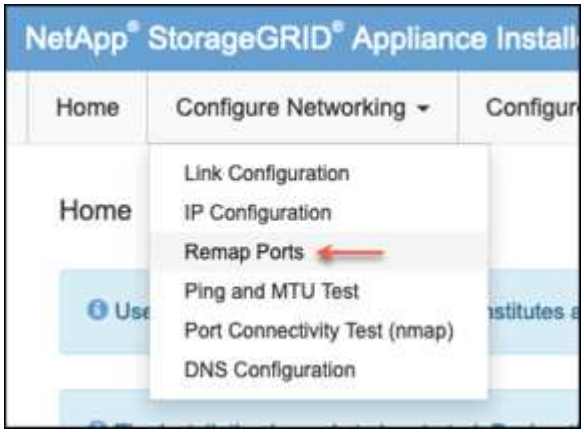
관리 노드를 재구축합니다

이제 원하는 모든 데이터의 백업 복사본이 있으며 그리드의 다른 관리 노드에 기록하거나 임시 위치에 저장되었으므로 어플라이언스를 재설정하여 포트 재맵을 구성할 수 있습니다.

1. 어플라이언스를 재설정하면 사전 설치된 상태로 돌아가고 호스트 이름, IP 및 네트워크 구성만 유지됩니다. 모든 데이터가 손실되므로 중요한 정보를 백업하도록 했습니다.
   a. 다음 명령을 입력합니다. `sgareinstall`

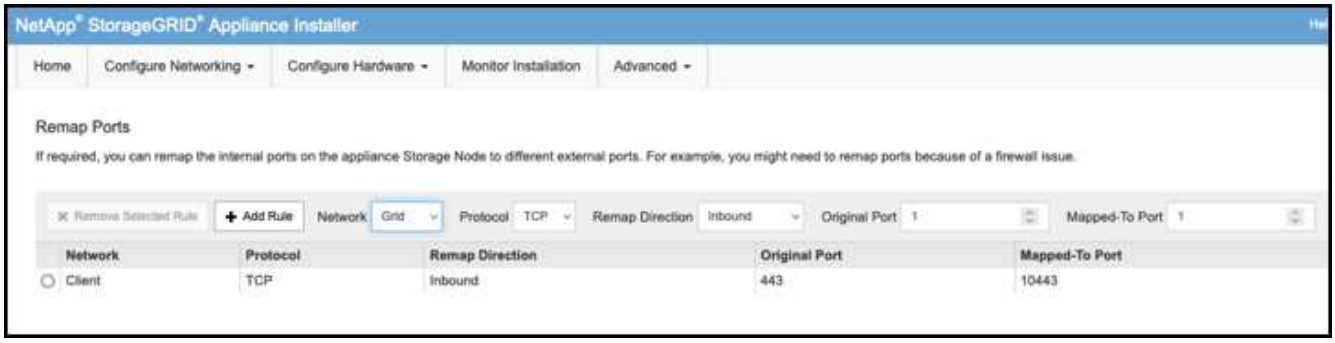
```
root@sg100-01:~ # sgareinstall
WARNING: All StorageGRID Webscale services on this node will be shut
down.
WARNING: Data stored on this node may be lost.
WARNING: You will have to reinstall StorageGRID Webscale to this
node.

After running this command and waiting a few minutes for the node to
reboot,
browse to one of the following URLs to reinstall StorageGRID Webscale
on
this node:

    https://10.193.174.192:8443
    https://10.193.204.192:8443
    https://169.254.0.1:8443

Are you sure you want to continue (y/n)? y
Renaming SG installation flag file.
Initiating a reboot to trigger the StorageGRID Webscale appliance
installation wizard.
```

2. 잠시 후 어플라이언스가 재부팅되고 노드 PGE UI에 액세스할 수 있습니다.
3. Configure Networking으로 이동합니다



4. 원하는 네트워크, 프로토콜, 방향 및 포트를 선택한 다음 규칙 추가 버튼을 클릭합니다.

> 그리드 네트워크에서 인바운드 포트 443을 다시 매핑하면 설치와 확장 절차가 중단됩니다. 그리드 네트워크에서 포트 443을 다시 매핑하지 않는 것이 좋습니다.



5. 원하는 포트 재맵이 추가되었습니다. 홈 탭으로 돌아가 설치 시작 버튼을 클릭합니다.

이제 의 관리 노드 복구 절차를 수행할 수 있습니다 "제품 설명서"

데이터베이스 및 로그 복원

이제 관리 노드가 복구되었으므로 메트릭, 로그 및 기간별 정보를 복구할 수 있습니다. 그리드에 다른 관리 노드가 있는 경우, 에 따르십시오 "제품 설명서" Prometheus-clone-db.sh_and_mi-clone-db.sh_scripts를 사용합니다. 이 노드가 유일한 관리 노드이고 이 데이터를 백업하도록 선택한 경우 다음 단계에 따라 정보를 복원할 수 있습니다.

감사 로그를 다시 복사합니다

1. 관리자 노드에 로그인합니다.
   a. 다음 명령을 입력합니다. `ssh admin@grid_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.
   e. SSH 에이전트에 SSH 개인 키를 추가합니다. 입력: `ssh-add`
   f. 에 나열된 SSH 액세스 암호를 입력합니다 `Passwords.txt` 파일.

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 보존된 감사 로그 파일을 복구된 관리 노드에 복사합니다. `scp admin@grid_node_IP:/var/local/tmp/saved-audit-logs/YYYY* .`
3. 보안을 위해 장애가 발생한 그리드 노드에서 복구된 관리 노드에 성공적으로 복사되었는지 확인한 후 감사 로그를 삭제합니다.
4. 복구된 관리 노드에서 감사 로그 파일의 사용자 및 그룹 설정을 업데이트합니다. `chown ams-user:bycast *`

또한 감사 공유에 대한 기존 클라이언트 액세스도 복원해야 합니다. 자세한 내용은 StorageGRID 관리 지침을 참조하십시오.

Prometheus 메트릭을 복원합니다

> Prometheus 데이터베이스를 복사하는 데 1시간 이상이 걸릴 수 있습니다. 일부 그리드 관리자 기능은 관리 노드에서 서비스가 중지되는 동안 사용할 수 없습니다.

1. 관리자 노드에 로그인합니다.
   a. 다음 명령을 입력합니다. `ssh admin@grid_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.
   e. SSH 에이전트에 SSH 개인 키를 추가합니다. 입력: `ssh-add`
   f. 에 나열된 SSH 액세스 암호를 입력합니다 `Passwords.txt` 파일.

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 관리 노드에서 Prometheus 서비스를 중지합니다. `service prometheus stop`
   a. 임시 백업 위치에서 관리자 노드로 Prometheus 데이터베이스를 복사합니다. `/rsync -azh --stats "backup_node:/var/local/tmp/prometheus/" "/var/local/mysql_ibdata/prometheus/"`
   b. 데이터가 올바른 경로에 있고 완전한지 확인합니다 `ls /var/local/mysql_ibdata/prometheus/data/`
3. 소스 관리 노드에서 Prometheus 서비스를 다시 시작합니다.`service prometheus start`

내역 정보를 복원합니다

1. 관리자 노드에 로그인합니다.
   a. 다음 명령을 입력합니다. `ssh admin@grid_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.
   e. SSH 에이전트에 SSH 개인 키를 추가합니다. 입력: `ssh-add`
   f. 에 나열된 SSH 액세스 암호를 입력합니다 `Passwords.txt` 파일.

```
When you are logged in as root, the prompt changes from `$` to `#`.
```

2. 대체 노드에서 MySQL 덤프 파일을 복사합니다. `scp grid_node_IP_:/var/local/tmp/mysql-mi.sql /var/local/tmp/mysql-mi.sql`
3. 관리자 노드에서 StorageGRID 서비스를 중지하고 NTP 및 MySQL을 시작합니다

a. 모든 서비스 중지: `service servermanager stop`

b. NTP 서비스 다시 시작: `service ntp start`.. MySQL 서비스를 다시 시작합니다. `service mysql start`

4. mi 데이터베이스를 드롭하고 비어 있는 새 데이터베이스를 생성합니다. `mysql -u *username* -p *password* -A mi -e "drop database mi; create database mi;"`
5. 데이터베이스 덤프에서 MySQL 데이터베이스 복원: `mysql -u *username* -p *password* -A mi < /var/local/tmp/mysql-mi.sql`
6. 다른 서비스를 모두 다시 시작합니다 `service servermanager start`

_ 아론 클라인 _

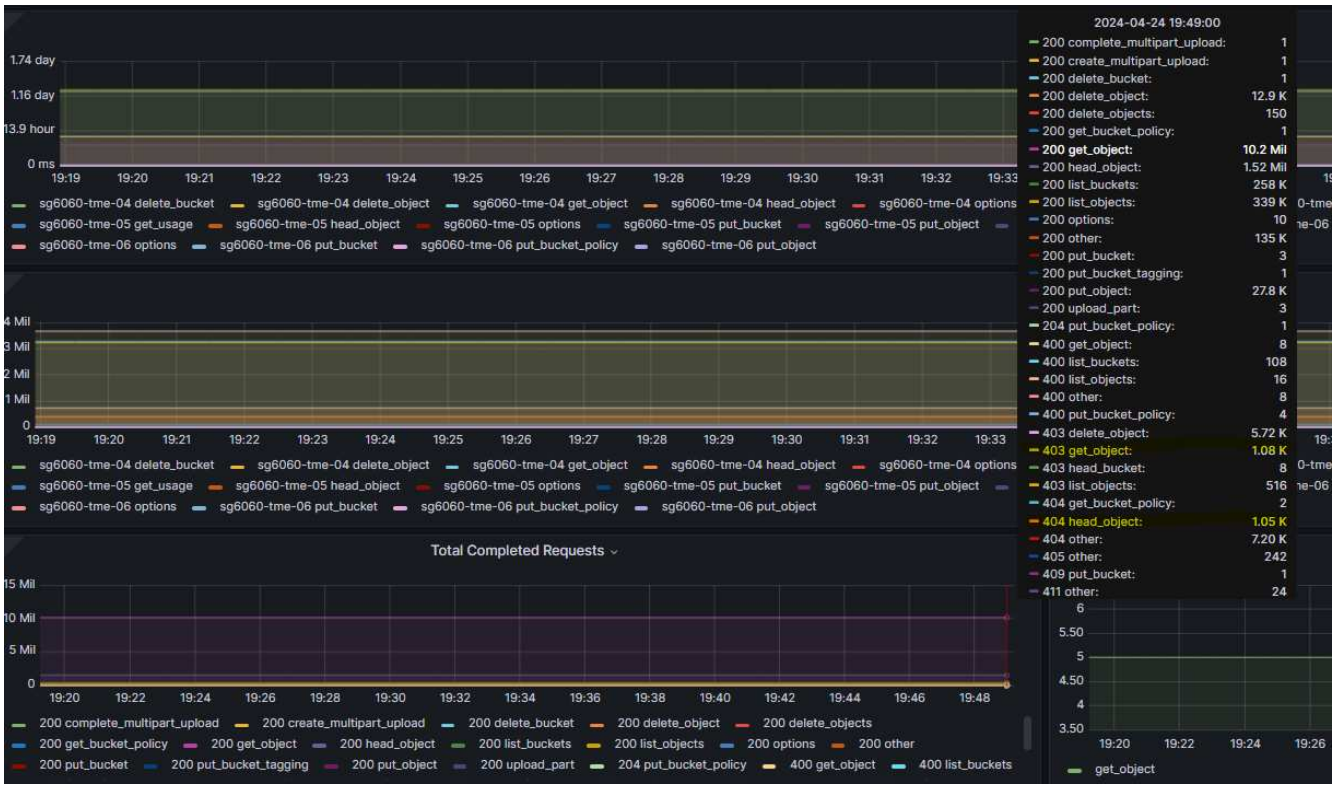
그리드 사이트 재배치 및 사이트 전체 네트워크 변경 절차

이 가이드에서는 다중 사이트 그리드에서 StorageGRID 사이트 재배치를 위한 준비 및 절차에 대해 설명합니다. 이 절차를 완전히 이해하고 원활한 프로세스를 보장하고 고객의 중단을 최소화할 수 있도록 미리 계획해야 합니다.

전체 그리드의 그리드 네트워크를 변경해야 하는 경우 을 참조하십시오
"그리드의 모든 노드에 대한 IP 주소를 변경합니다".

사이트 재배치 전 고려 사항

- 사이트 이동을 완료하고 모든 노드를 15일 이내에 온라인으로 전환하여 Cassandra 데이터베이스가 재구축되지 않도록 합니다.
  "스토리지 노드를 15일 이상 복구합니다"
- 활성 정책의 ILM 규칙이 엄격한 수집 동작을 사용하고 있는 경우, 고객이 사이트 재배치 중에 그리드에 개체를 계속 넣으려는 경우 이를 밸런스 또는 이중 커밋으로 변경하는 것이 좋습니다.
- 60개 이상의 드라이브가 있는 스토리지 어플라이언스의 경우 디스크 드라이브가 설치된 상태로 쉘프를 이동하지 마십시오. 포장/이동 전에 각 디스크 드라이브에 레이블을 지정하고 스토리지 인클로저에서 분리하십시오.
- StorageGRID 어플라이언스 변경 그리드 네트워크 VLAN은 관리 네트워크 또는 클라이언트 네트워크를 통해 원격으로 수행할 수 있습니다. 또는 재배치 이전 또는 이후에 변경을 수행하기 위해 현장에 있을 계획입니다.
- 고객 응용 프로그램이 HEAD를 사용하고 있는지 또는 넣기 전에 존재하지 않는 개체를 가져오는지 확인합니다. 그렇다면 HTTP 500 오류를 방지하기 위해 버킷 일관성을 강력한 사이트로 변경합니다. 확실하지 않은 경우, S3 개요 Grafana Charts * 그리드 매니저 > 지원 > 메트릭 * 에서 '총 완료된 요청' 차트 위에 마우스를 올려 놓습니다. 404 Get Object 또는 404 head object의 개수가 매우 많으면 하나 이상의 응용 프로그램이 head 또는 get nonexistance object를 사용하고 있을 가능성이 높습니다. 카운트가 누적됩니다. 다른 타임라인에 마우스를 갖다 대면 차이를 확인할 수 있습니다.



사이트 재배치 전 그리드 **IP** 주소를 변경하는 절차

단계

1. 새 그리드 네트워크 서브넷을 새 위치에서 사용할 경우 "그리드 네트워크 서브넷 목록에 서브넷을 추가합니다"
2. 기본 관리자 노드에 로그인하고 change-IP를 사용하여 그리드 IP를 변경합니다. 변경을 위해 노드를 종료하기 전에 * 단계 * 해야 합니다.
   a. 그리드 IP 변경에 대해 2와 1을 차례로 선택합니다

```
Editing: Node IP/subnet and gateway

Use up arrow to recall a previously typed value, which you can then edit
Use d or 0.0.0.0/0 as the IP/mask to delete the network from the node
Use q to complete the editing session early and return to the previous menu
Press <enter> to use the value shown in square brackets

================================================================================
Site: LONDON
================================================================================
LONDON-ADM1 Grid   IP/mask [     10.45.74.14/26 ]: 10.45.74.24/26
LONDON-S1   Grid   IP/mask [     10.45.74.16/26 ]: 10.45.74.26/26
LONDON-S2   Grid   IP/mask [     10.45.74.17/26 ]: 10.45.74.27/26
LONDON-S3   Grid   IP/mask [     10.45.74.18/26 ]: 10.45.74.28/26
================================================================================
LONDON-ADM1 Grid   Gateway [         10.45.74.1 ]:
LONDON-S1   Grid   Gateway [         10.45.74.1 ]:
LONDON-S2   Grid   Gateway [         10.45.74.1 ]:
LONDON-S3   Grid   Gateway [         10.45.74.1 ]:
================================================================================
================================================================================
Site: OXFORD
================================================================================
OXFORD-ADM1 Grid   IP/mask [     10.45.75.14/26 ]:
OXFORD-S1   Grid   IP/mask [     10.45.75.16/26 ]:
OXFORD-S2   Grid   IP/mask [     10.45.75.17/26 ]:
OXFORD-S3   Grid   IP/mask [     10.45.75.18/26 ]:
================================================================================
OXFORD-ADM1 Grid   Gateway [         10.45.75.1 ]:
OXFORD-S1   Grid   Gateway [         10.45.75.1 ]:
OXFORD-S2   Grid   Gateway [         10.45.75.1 ]:
OXFORD-S3   Grid   Gateway [         10.45.75.1 ]:
================================================================================
Finished editing. Press Enter to return to menu.
```

b. 5를 선택하여 변경 사항을 표시합니다

```
================================================================================
Site: LONDON
================================================================================
LONDON-ADM1 Grid   IP      [     10.45.74.14/26 ]: 10.45.74.24/26
LONDON-S1   Grid   IP      [     10.45.74.16/26 ]: 10.45.74.26/26
LONDON-S2   Grid   IP      [     10.45.74.17/26 ]: 10.45.74.27/26
LONDON-S3   Grid   IP      [     10.45.74.18/26 ]: 10.45.74.28/26
Press Enter to continue
```

c. 10을 선택하여 변경 사항을 확인하고 적용합니다.

```
Welcome to the StorageGRID IP Change Tool.

Selected nodes: all

1:   SELECT NODES to edit
2:   EDIT IP/mask and gateway
3:   EDIT admin network subnet lists
4:   EDIT grid network subnet list
5:   SHOW changes
6:   SHOW full configuration, with changes highlighted
7:   VALIDATE changes
8:   SAVE changes, so you can resume later
9:   CLEAR all changes, to start fresh
10:  APPLY changes to the grid
0:   Exit

Selection: 10
```

d. 이 단계에서 * stage * 를 선택해야 합니다.

```
Validating new networking configuration... PASSED.
Checking for Grid Network IP address swaps... PASSED.

Applying these changes will update the following nodes:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

The following nodes will also require restarting:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

Select one of the following options:

  apply:  apply all changes and automatically restart nodes (if necessary)
  stage:  stage the changes; no changes will take effect until the nodes are restarted
  cancel: do not make any network changes at this time

[apply/stage/cancel]> stage
```

e. 위 변경에 기본 관리 노드가 포함되어 있는 경우 * 'a'를 입력하여 운영 관리 노드를 수동으로 다시 시작합니다 *

```
10.45.74.14 - PuTTY
Validating new networking configuration... PASSED.
Checking for Grid Network IP address swaps... PASSED.

Applying these changes will update the following nodes:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

The following nodes will also require restarting:

  LONDON-ADM1
  LONDON-S1
  LONDON-S2
  LONDON-S3

Select one of the following options:

  apply:  apply all changes and automatically restart nodes (if necessary)
  stage:  stage the changes; no changes will take effect until the nodes are restarted
  cancel: do not make any network changes at this time

[apply/stage/cancel]> stage

Generating new grid networking description file... PASSED.
Running provisioning... PASSED.
Updating network configuration on LONDON-S1... PASSED.
Updating network configuration on LONDON-S2... PASSED.
Updating network configuration on LONDON-S3... PASSED.
Updating network configuration on LONDON-ADM1... PASSED.
Finished staging network changes. You must manually restart these nodes for the changes to take effect:

  LONDON-ADM1 (has IP 10.45.74.14 until restart)
  LONDON-S1 (has IP 10.45.74.16 until restart)
  LONDON-S2 (has IP 10.45.74.17 until restart)
  LONDON-S3 (has IP 10.45.74.18 until restart)

Importing bundles... PASSED.
*****************************************************************
*                          IMPORTANT                            *
*                                                               *
*  A new recovery package has been generated as a result of the *
*  configuration change. Select Maintenance > Recovery Package  *
*  in the Grid Manager to download it.                          *
*****************************************************************

Network Update Complete.  Primary admin restart required. Select 'continue' to restart this node immediately, 'abort' to restart manually.
Enter a to abort, c to continue [a/c]>
```

f. 이전 메뉴로 돌아가고 change-IP 인터페이스에서 나가려면 Enter 키를 누릅니다.

```
Network Update Complete.  Primary admin restart required. Select 'continue' to restart this node immediately, 'abort' to restart manually.
Enter a to abort, c to continue [a/c]> a
Restart aborted. You must manually restart this node as soon as possible
Press Enter to return to the previous menu.
```

3. Grid Manager에서 새 복구 패키지를 다운로드합니다. * 그리드 관리자 * > * 유지 관리 * > * 복구 패키지 *
4. StorageGRID 어플라이언스에서 VLAN 변경이 필요한 경우 섹션을 참조하십시오 어플라이언스 VLAN 변경.
5. 사이트의 모든 노드 및/또는 어플라이언스를 종료하고, 필요한 경우 디스크 드라이브에 레이블을 붙이거나 제거하고, 랙을 해제하고, 포장하고, 이동합니다.
6. 관리자 네트워크 IP 및/또는 클라이언트 VLAN 및 IP 주소를 변경하려는 경우 재배치 후 변경 작업을 수행할 수 있습니다.

어플라이언스 **VLAN** 변경

아래 절차에서는 StorageGRID 어플라이언스의 관리자 또는 클라이언트 네트워크에 원격으로 액세스하여 원격으로 변경을 수행하는 것으로 가정합니다.

단계

1. 제품을 종료하기 전에 "제품을 유지보수 모드로 두십시오".
2. 브라우저를 사용하여 를 사용하여 StorageGRID 어플라이언스 설치 프로그램 GUI에 액세스합니다

https://<admin-or-client-network-ip>:8443. 어플라이언스가 유지보수 모드로 부팅된 후 새 그리드 IP가 이미 있으므로 그리드 IP를 사용할 수 없습니다.

3. 그리드 네트워크의 VLAN을 변경합니다. 클라이언트 네트워크를 통해 어플라이언스에 액세스하는 경우 지금은 클라이언트 VLAN을 변경할 수 없으며 이동 후 변경할 수 있습니다.
4. 어플라이언스에 SSH로 연결하고 'shutdoown -h now'를 사용하여 노드를 종료합니다.
5. 어플라이언스가 새 사이트에서 준비되면 를 사용하여 StorageGRID 어플라이언스 설치 프로그램 GUI에 액세스합니다 https://<grid-network-ip>:8443. GUI에서 ping/nmap 툴을 사용하여 스토리지가 최적의 상태이고 다른 그리드 노드에 대한 네트워크 연결인지 확인합니다.
6. 클라이언트 네트워크 IP를 변경하려는 경우 이 단계에서 클라이언트 VLAN을 변경할 수 있습니다. 클라이언트 네트워크는 이후 단계에서 change-IP 도구를 사용하여 클라이언트 네트워크 IP를 업데이트할 때까지 준비되지 않습니다.
7. 유지보수 모드를 종료합니다. StorageGRID 어플라이언스 설치 프로그램에서 * 고급 * > * 컨트롤러 재부팅 * 을 선택한 다음 * StorageGRID * 으로 재부팅 * 을 선택합니다.
8. 모든 노드가 가동되고 그리드에 연결 문제가 표시되지 않으면 필요에 따라 change-IP를 사용하여 어플라이언스 관리 네트워크와 클라이언트 네트워크를 업데이트합니다.

툴 및 애플리케이션 가이드

StorageGRID와 함께 Cloudera Hadoop S3A 커넥터를 사용하십시오

Hadoop은 한동안 데이터 과학자들이 선호하는 분야입니다. Hadoop을 사용하면 간단한 프로그래밍 프레임워크를 사용하여 컴퓨터 클러스터 전반에 걸쳐 대규모 데이터 세트를 분산 처리할 수 있습니다. Hadoop은 단일 서버에서 수천 개의 시스템으로 스케일업할 수 있도록 설계되었으며, 각 시스템은 로컬 컴퓨팅과 스토리지를 소유합니다.

Hadoop 워크플로우에 S3A를 사용하는 이유는 무엇입니까?

시간이 지나면서 데이터 양이 증가함에 따라 자체 컴퓨팅 및 스토리지로 새 시스템을 추가하는 방식이 비효율적으로 되었습니다. 선형적으로 확장하면 리소스를 효율적으로 사용하고 인프라를 관리하는 데 어려움이 발생합니다.

이러한 과제를 해결하기 위해 Hadoop S3A 클라이언트는 S3 오브젝트 스토리지에 대한 고성능 I/O를 제공합니다. S3A를 사용하여 Hadoop 워크플로우를 구축하면 오브젝트 스토리지를 데이터 저장소로 활용할 수 있으며, 컴퓨팅과 스토리지를 독립적으로 확장할 수 있는 분리된 컴퓨팅 및 스토리지를 사용할 수 있습니다. 또한 컴퓨팅과 스토리지를 분리하여 컴퓨팅 작업에 적절한 양의 리소스를 할당하고 데이터 세트 크기에 따라 용량을 제공할 수 있습니다. 따라서 Hadoop 워크플로우의 전체 TCO를 줄일 수 있습니다.

StorageGRID를 사용하도록 S3A 커넥터를 구성합니다

필수 구성 요소

- StorageGRID S3 엔드포인트 URL, 테넌트 S3 액세스 키 및 Hadoop S3A 연결 테스트를 위한 암호 키입니다.
- Java 패키지를 설치하기 위해 클러스터의 각 호스트에 대한 Cloudera 클러스터 및 루트 또는 sudo 권한입니다.

2022년 4월 현재, Cloudera 7.1.7을 사용한 Java 11.0.14는 StorageGRID 11.5 및 11.6을 대상으로 테스트를 마쳤습니다. 그러나 Java 버전 번호는 새로 설치할 때 다를 수 있습니다.

Java 패키지를 설치합니다

1. 를 확인하십시오 "Cloudera 지원 매트릭스" 지원되는 JDK 버전.
2. 를 다운로드합니다 "Java 11.x 패키지" 이 운영 체제는 Cloudera 클러스터 운영 체제와 일치합니다. 이 패키지를 클러스터의 각 호스트에 복사합니다. 이 예에서는 CentOS에 rpm 패키지가 사용됩니다.
3. 각 호스트에 루트로 로그인하거나 sudo 권한이 있는 계정을 사용합니다. 각 호스트에서 다음 단계를 수행합니다.
   a. 패키지 설치:

   ```
   $ sudo rpm -Uvh jdk-11.0.14_linux-x64_bin.rpm
   ```

   b. Java가 설치된 위치를 확인합니다. 여러 버전이 설치된 경우 새로 설치된 버전을 기본값으로 설정합니다.

```
alternatives --config java

There are 2 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
 +1           /usr/java/jre1.8.0_291-amd64/bin/java
  2           /usr/java/jdk-11.0.14/bin/java

Enter to keep the current selection[+], or type selection number: 2
```

c. 이 줄을 '/etc/profile' 끝에 추가합니다. 경로는 위의 선택 경로와 일치해야 합니다.

```
export JAVA_HOME=/usr/java/jdk-11.0.14
```

d. 프로파일을 적용하려면 다음 명령을 실행합니다.

```
source /etc/profile
```

Cloudera HDFS S3A 구성

- 단계 *

1. Cloudera Manager GUI에서 클러스터 > HDFS를 선택하고 Configuration을 선택합니다.
2. 범주 아래에서 고급을 선택하고 아래로 스크롤하여 core-site.xml에 대한 클러스터 차원의 고급 구성 조각(안전 밸브)을 찾습니다.
3. (+) 기호를 클릭하고 다음 값 쌍을 추가합니다.

이름	값
fs.s3a.access.key	_<StorageGRID의 테넌트 S3 액세스 키> _
fs.s3a.secret.key	_<StorageGRID의 테넌트 S3 비밀 키> _
FS.s3a.CONNECTION.SSL.ENABLED	[true 또는 false] (이 항목이 누락된 경우 기본값은 https)
FS.s3a.endpoint	_<StorageGRID S3 엔드포인트: port> _
FS.s3a.IMPL	org.apache.하둡.fs.s3a.s3aFileSystem

이름	값
FS.s3a.path.style.access	[TRUE 또는 FALSE](이 항목이 누락된 경우 기본값은 가상 호스트 스타일)

- 샘플 스크린샷 *

Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml

core_site_safety_valve

HDFS (Service-Wide) Undo

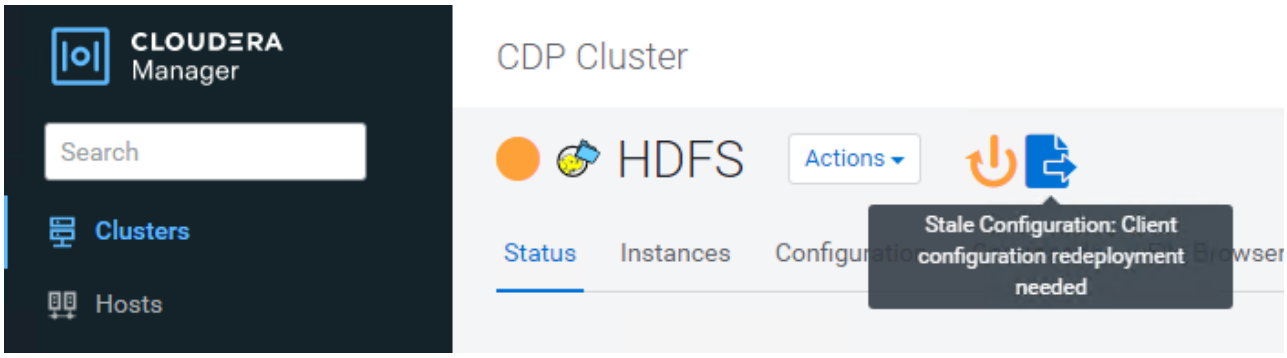
View as XML

Name	fs.s3a.endpoint
Value	sgdemo.netapp.com:10443
Description	StorageGRID s3 load balancer endpoint
	☑ Final
Name	fs.s3a.access.key
Value	OM[redacted]BAN
Description	SG CDP S3 access key
	☑ Final
Name	fs.s3a.secret.key
Value	mapz[redacted]Qfc
Description	SG CDP S3 secret key
	☑ Final
Name	fs.s3a.impl
Value	org.apache.hadoop.fs.s3a.S3AFileSystem
Description	
	☑ Final
Name	fs.s3a.path.style.access
Value	true
Description	
	☑ Final

Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml

Save Changes(CTRL+S)

1. 변경 내용 저장 단추를 클릭합니다. HDFS 메뉴 표시줄에서 오래된 구성 아이콘을 선택하고 다음 페이지에서 오래된 서비스 다시 시작 을 선택한 다음 지금 다시 시작 을 선택합니다.



StorageGRID에 대한 S3A 연결을 테스트합니다

기본 연결 테스트를 수행합니다

Cloudera 클러스터의 호스트 중 하나에 로그인하고 'Hadoop fs-ls s3a://<*bucket-name*>/'를 입력합니다.

다음 예에서는 경로 syle을 기존 HDFS 테스트 버킷과 테스트 객체와 함께 사용합니다.

```
[root@ce-n1 ~]# hadoop fs -ls s3a://hdfs-test/
22/02/15 18:24:37 WARN impl.MetricsConfig: Cannot locate configuration:
tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
22/02/15 18:24:37 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
22/02/15 18:24:37 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system started
22/02/15 18:24:37 INFO Configuration.deprecation: No unit for
fs.s3a.connection.request.timeout(0) assuming SECONDS
Found 1 items
-rw-rw-rw-   1 root root       1679 2022-02-14 16:03 s3a://hdfs-test/test
22/02/15 18:24:38 INFO impl.MetricsSystemImpl: Stopping s3a-file-system
metrics system...
22/02/15 18:24:38 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system stopped.
22/02/15 18:24:38 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system shutdown complete.
```

문제 해결

시나리오 1

StorageGRID에 대한 HTTPS 연결을 사용하고 15분 시간 제한 후 "shake_failure" 오류가 발생합니다.

- 이유: * StorageGRID 연결을 위해 오래되었거나 지원되지 않는 TLS 암호 제품군을 사용하는 이전 JRE/JDK 버전.
- 샘플 오류 메시지 *

```
[root@ce-n1 ~]# hadoop fs -ls s3a://hdfs-test/
22/02/15 18:52:34 WARN impl.MetricsConfig: Cannot locate configuration:
tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
22/02/15 18:52:34 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
22/02/15 18:52:34 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system started
22/02/15 18:52:35 INFO Configuration.deprecation: No unit for
fs.s3a.connection.request.timeout(0) assuming SECONDS
22/02/15 19:04:51 INFO impl.MetricsSystemImpl: Stopping s3a-file-system
metrics system...
22/02/15 19:04:51 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system stopped.
22/02/15 19:04:51 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system shutdown complete.
22/02/15 19:04:51 WARN fs.FileSystem: Failed to initialize fileystem
s3a://hdfs-test/: org.apache.hadoop.fs.s3a.AWSClientIOException:
doesBucketExistV2 on hdfs: com.amazonaws.SdkClientException: Unable to
execute HTTP request: Received fatal alert: handshake_failure: Unable to
execute HTTP request: Received fatal alert: handshake_failure
ls: doesBucketExistV2 on hdfs: com.amazonaws.SdkClientException: Unable to
execute HTTP request: Received fatal alert: handshake_failure: Unable to
execute HTTP request: Received fatal alert: handshake_failure
```

- 해상도: * JDK 11.x 이상이 설치되어 있는지 확인하고 Java 라이브러리를 기본값으로 설정합니다. 을 참조하십시오 Java 패키지를 설치합니다 섹션을 참조하십시오.

시나리오 **2:**

"요청한 대상에 대한 유효한 인증 경로를 찾을 수 없습니다."라는 오류 메시지와 함께 StorageGRID에 연결하지 못했습니다.

- 이유: * StorageGRID S3 엔드포인트 서버 인증서가 Java 프로그램에서 신뢰되지 않습니다.

샘플 오류 메시지:

```
[root@hdp6 ~]# hadoop fs -ls s3a://hdfs-test/
22/03/11 20:58:12 WARN impl.MetricsConfig: Cannot locate configuration:
tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
22/03/11 20:58:13 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
22/03/11 20:58:13 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system started
22/03/11 20:58:13 INFO Configuration.deprecation: No unit for
fs.s3a.connection.request.timeout(0) assuming SECONDS
22/03/11 21:12:25 INFO impl.MetricsSystemImpl: Stopping s3a-file-system
metrics system...
22/03/11 21:12:25 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system stopped.
22/03/11 21:12:25 INFO impl.MetricsSystemImpl: s3a-file-system metrics
system shutdown complete.
22/03/11 21:12:25 WARN fs.FileSystem: Failed to initialize fileystem
s3a://hdfs-test/: org.apache.hadoop.fs.s3a.AWSClientIOException:
doesBucketExistV2 on hdfs: com.amazonaws.SdkClientException: Unable to
execute HTTP request: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target: Unable to execute HTTP
request: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

- 해결 방법: * 알려진 공개 인증서 서명 기관에서 발급한 서버 인증서를 사용하여 인증이 보안되는지 확인하는 것이 좋습니다. 또는 사용자 지정 CA 또는 서버 인증서를 Java 신뢰 저장소에 추가합니다.

StorageGRID 사용자 지정 CA 또는 서버 인증서를 Java 신뢰 저장소에 추가하려면 다음 단계를 수행하십시오.

1. 기존 기본 Java cacerts 파일을 백업합니다.

```
cp -ap $JAVA_HOME/lib/security/cacerts
$JAVA_HOME/lib/security/cacerts.orig
```

2. StorageGRID S3 끝점 인증서를 Java 신뢰 저장소로 가져옵니다.

```
keytool -import -trustcacerts -keystore $JAVA_HOME/lib/security/cacerts
-storepass changeit -noprompt -alias sg-lb -file <StorageGRID CA or
server cert in pem format>
```

문제 해결 팁

1. 디버깅하려면 Hadoop 로그 수준을 높입니다.

   export hadoop_root_logger=hadoop.root.logger=debug, console

2. 명령을 실행하고 로그 메시지를 error.log로 전달합니다.

   'Hadoop fs-ls s3a://<*bucket-name*>/&> error.log'

안젤라 청 _ 에 의해

S3cmd를 사용하여 StorageGRID에서 S3 액세스를 테스트하고 시연합니다

S3cmd 는 S3 작업을 위한 무료 명령줄 도구 및 클라이언트입니다. s3cmd를 사용하여 StorageGRID에서 S3 액세스를 테스트하고 시연할 수 있습니다.

S3cmd를 설치하고 구성합니다

워크스테이션이나 서버에 S3cmd를 설치하려면 에서 다운로드합니다 "명령줄 S3 클라이언트". s3cmd 는 문제 해결을 지원하기 위한 도구로 각 StorageGRID 노드에 미리 설치되어 있습니다.

초기 구성 단계

1. s3cmd — 구성
2. access_key와 secret_key만 제공하십시오. 나머지는 기본값을 유지합니다.
3. 제공된 자격 증명으로 액세스를 테스트하시겠습니까? [Y/n]:n(실패하므로 테스트 생략)
4. 설정을 저장하시겠습니까? [y/N]y입니다
   a. 구성이 '/root/.s3cfg'에 저장되었습니다.
5. s3cfg에서 "=" 기호 다음에 host_base 및 host_bucket 필드가 비어 있도록 합니다.
   a. host_base=
   b. host_bucket=

> 4단계에서 host_base 및 host_bucket을 지정하는 경우 CLI에서 -host를 사용하여 엔드포인트를 지정할 필요가 없습니다. 예:

```
host_base = 192.168.1.91:8082
host_bucket = bucketX.192.168.1.91:8082
s3cmd ls s3://bucketX --no-check-certificate
```

기본 명령 예

- * 버킷 생성: *

  S 3cmd MB S3://s3cmdbucket — host=<endpoint>:<port> — no-check-certificate

- * 모든 버킷 나열: *

  S 3cmd ls — host=<endpoint>:<port> — no-check-certificate

- * 모든 버킷과 해당 내용을 나열합니다. *

  S 3cmd la—host=<endpoint>:<port>--no-check-certificate

- * 특정 버킷의 오브젝트 목록: *

  's3cmd ls s3://<bucket>--host=<endpoint>:<port>--no-check-certificate'

- * 버킷 삭제: *

  S 3cmd rb s3://s3cmdbucket — host=<endpoint>:<port> — no-check-certificate

- * 개체 넣기: *

  S 3cmd put <file>S3:/<bucket>--host=<endpoint>:<port>--no-check-certificate

- * 개체 가져오기: *

  's3cmd get s3://<bucket>/<object><file>--host=<endpoint>:<port>--no-check-certificate'

- * 개체 삭제: *

  S 3cmd del S3://<bucket>/<object>--host=<endpoint>:<port>--no-check-certificate

_ 아론 클라인 _

NetApp StorageGRID를 공동 스토리지로 사용하는 Vertica Eon 모드 데이터베이스

이 가이드에서는 NetApp StorageGRID에서 공용 스토리지를 사용하는 Vertica Eon Mode 데이터베이스를 생성하는 절차를 설명합니다.

소개

Vertica는 분석 데이터베이스 관리 소프트웨어입니다. 대량의 데이터를 처리하도록 설계된 columnar 스토리지 플랫폼으로서, 기존의 집약적인 시나리오에서 매우 빠른 쿼리 성능을 지원합니다. Vertica 데이터베이스는 Eon 또는 Enterprise의 두 가지 모드 중 하나로 실행됩니다. 두 모드를 모두 사내 또는 클라우드에 구축할 수 있습니다.

EON 및 엔터프라이즈 모드는 주로 데이터를 저장하는 위치에 따라 다릅니다.

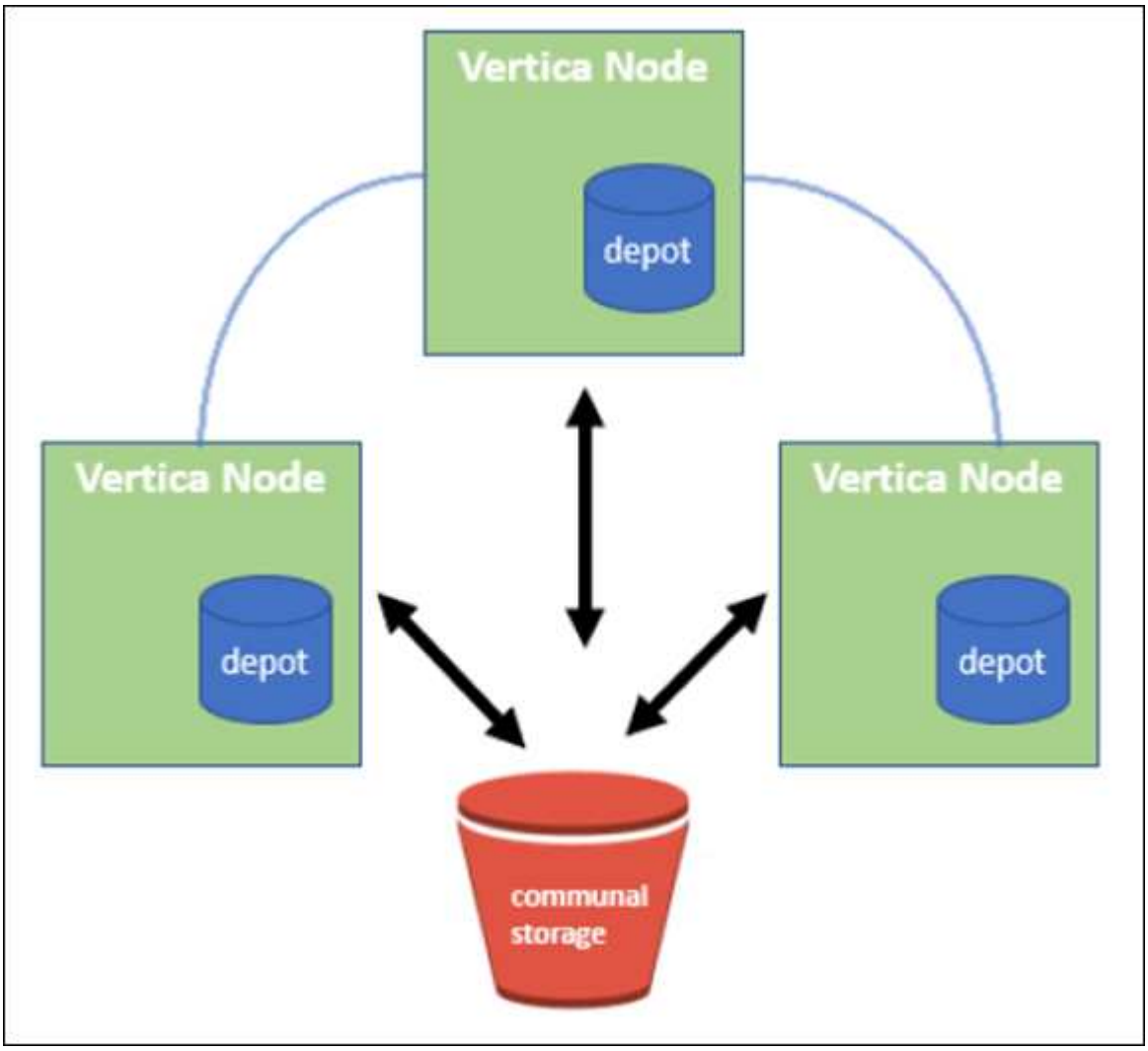
- eon Mode 데이터베이스는 공용 스토리지를 사용하여 데이터를 저장합니다. 이 방법은 Vertica에서 권장합니다.

- 엔터프라이즈 모드 데이터베이스는 데이터베이스를 구성하는 노드의 파일 시스템에 로컬로 데이터를 저장합니다.

EON 모드 아키텍처

eon Mode는 컴퓨팅 리소스를 데이터베이스의 공용 스토리지 계층과 분리하여 컴퓨팅과 스토리지를 별도로 확장할 수 있도록 합니다. Eon Mode의 Vertica는 다양한 워크로드를 처리하고 별도의 컴퓨팅 및 스토리지 리소스를 사용하여 서로 격리하도록 최적화되어 있습니다.

eon Mode는 사내 또는 Amazon S3에 호스팅된 공유 오브젝트 저장소인 S3 버킷에 데이터를 저장합니다.



공용 저장 장치

Eon Mode는 데이터를 로컬에 저장하는 대신 모든 데이터와 카탈로그(메타데이터)에 단일 공동 스토리지 위치를 사용합니다. 공용 스토리지는 데이터베이스 노드 간에 공유되는 데이터베이스의 중앙 집중식 저장소 위치입니다.

공용 스토리지의 속성은 다음과 같습니다.

- 클라우드 또는 온프레미스 오브젝트 스토리지의 공용 스토리지는 개별 시스템의 디스크 스토리지에 비해 스토리지 장애로 인해 데이터 손실이 발생할 가능성이 더 적고 복구 성능이 더 낮습니다.
- 모든 데이터는 동일한 경로를 사용하여 모든 노드에서 읽을 수 있습니다.

- 용량은 노드의 디스크 공간에 의해 제한되지 않습니다.
- 데이터는 적소에 저장되므로 변화하는 요구사항에 따라 클러스터를 탄력적으로 확장할 수 있습니다. 데이터가 노드에 로컬로 저장된 경우 노드를 추가하거나 제거하려면 노드 간에 상당한 양의 데이터를 이동해야 하며, 제거할 노드나 새로 생성된 노드로 이동해야 합니다.

물류창고입니다

공용 스토리지의 한 가지 단점은 속도이다. 공유 클라우드 위치에서 데이터에 액세스하는 것은 로컬 디스크에서 데이터를 읽는 것보다 속도가 느립니다. 또한 많은 노드에서 데이터를 한 번에 읽는 경우에는 공용 스토리지에 대한 연결이 병목 현상을 일으킬 수 있습니다. 데이터 액세스 속도를 높이기 위해 Eon Mode 데이터베이스의 노드는 데이터 센터라는 데이터의 로컬 디스크 캐시를 유지합니다. 쿼리를 실행할 때 노드는 먼저 필요한 데이터가 서비스 센터에 있는지 확인합니다. 이 경우 데이터의 로컬 복사본을 사용하여 쿼리를 완료합니다. 데이터가 서비스 센터에 없는 경우 노드는 공용 스토리지에서 데이터를 가져와 서비스 센터에 복사본을 저장합니다.

NetApp StorageGRID 권장 사항

Vertica는 데이터베이스 데이터를 오브젝트 스토리지에 수천 또는 수백만 개의 압축된 오브젝트(관찰 크기는 오브젝트당 200~500MB임)로 저장합니다. 사용자가 데이터베이스 쿼리를 실행하면 Vertica는 바이트 범위 가져오기 호출을 사용하여 이러한 압축된 개체에서 선택한 데이터 범위를 병렬로 검색합니다. 각 바이트 범위 GET는 약 8KB입니다.

10TB 데이터베이스 서비스 센터에서 사용자 쿼리 테스트를 수행하는 동안 초당 4,000 - 10,000개의 GET(바이트 범위 가져오기) 요청이 그리드에 전송되었습니다. SG6060 어플라이언스를 사용하여 이 테스트를 실행할 때 어플라이언스 노드당 CPU % 활용도가 낮지만(약 20% ~ 30%) CPU 시간의 2/3 이상이 I/O를 기다리고 있습니다 SGF6024에서는 I/O 대기가 0% ~ 0.5%로 매우 적습니다.

지연 시간이 매우 짧은 소규모 IOPS의 높은 수요(평균 0.01초 미만)로 인해 오브젝트 스토리지 서비스에 SFG6024를 사용하는 것이 좋습니다. SG6060이 매우 큰 데이터베이스 크기에 필요한 경우 고객은 적극적으로 쿼리한 데이터 세트를 지원하기 위해 서비스 센터 사이징에 대해 Vertica 계정 팀과 협력해야 합니다.

Admin Node 및 API Gateway Node의 경우 고객은 SG100 또는 SG1000을 사용할 수 있습니다. 병렬 및 데이터베이스 크기의 사용자 쿼리 요청 수에 따라 선택이 달라집니다. 고객이 타사 로드 밸런싱 장치를 사용하려는 경우, 성능 수요가 높은 워크로드를 위한 전용 로드 밸런싱 장치를 사용하는 것이 좋습니다. StorageGRID 사이징의 경우 NetApp 세일즈 팀에 문의하십시오.

기타 StorageGRID 구성 권장 사항은 다음과 같습니다.

- * 그리드 토폴로지 *. SGF6024를 동일한 그리드 사이트에서 다른 스토리지 어플라이언스 모델과 혼합하지 마십시오. 장기간 아카이브 보호를 위해 SG6060을 사용하려는 경우 성능을 향상시키기 위해 활성 데이터베이스의 전용 그리드 로드 밸런서가 있는 SGF6024를 자체 그리드 사이트(물리적 또는 논리적 사이트)에 보관하십시오. 여러 가지 어플라이언스 모델을 동일한 사이트에서 혼합하여 사이트에서의 전반적인 성능을 줄일 수 있습니다.
- * 데이터 보호 *. 보호를 위해 복제 복사본을 사용합니다. 활성 데이터베이스에 대해 삭제 코딩을 사용하지 마십시오. 고객은 비활성 데이터베이스를 장기간 보호하기 위해 삭제 코딩을 사용할 수 있습니다.
- * 그리드 압축 사용 안 함 *. Vertica 개체 저장소에 저장하기 전에 개체를 압축합니다. 그리드 압축을 사용하면 스토리지 사용량이 추가로 절감되지 않고 바이트 범위 가져오기 성능이 크게 저하됩니다.
- * HTTP 대 HTTPS S3 엔드포인트 연결 *. 벤치마크 테스트 중에 Vertica 클러스터에서 StorageGRID 로드 밸런서 엔드포인트로 HTTP S3 연결을 사용할 경우 성능이 약 5% 향상되는 것을 확인했습니다. 이 선택은 고객의 보안 요구 사항을 기반으로 해야 합니다.

Vertica 구성을 위한 권장 사항은 다음과 같습니다.

- * Vertica 데이터베이스 기본 서비스 센터 설정은 읽기 및 쓰기 작업에 대해 활성화(값 = 1)됩니다 *. 성능 향상을

위해 이러한 서비스 센터 설정을 유지할 것을 적극 권장합니다.

- * 스트리밍 제한 비활성화 *. 구성에 대한 자세한 내용은 섹션을 참조하십시오 스트리밍 제한 비활성화.

StorageGRID에서 공용 스토리지를 사용하는 온프레미스 Eon 모드 설치

다음 섹션에서는 StorageGRID에서 공용 스토리지를 사용하여 Eon 모드를 사내에 설치하는 절차에 대해 설명합니다. 사내 S3(Simple Storage Service) 호환 오브젝트 스토리지를 구성하는 절차는 Vertica 가이드의 절차와 유사합니다. "Eon 모드 데이터베이스를 온-프레미스에 설치합니다".

기능 테스트에 사용된 설정은 다음과 같습니다.

- StorageGRID 11.4.0.4
- Vertica 10.1.0
- Vertica 노드용 CentOS 7.x OS가 있는 3개의 가상 시스템(VM)이 클러스터를 구성합니다. 이 설정은 기능 테스트 전용이며, Vertica 운영 데이터베이스 클러스터용이 아닙니다.

이 세 노드는 SSH(Secure Shell) 키로 설정되어 클러스터 내의 노드 간에 암호 없이 SSH를 사용할 수 있습니다.

NetApp StorageGRID에 필요한 정보입니다

StorageGRID에서 공용 스토리지를 사용하는 사내에 Eon 모드를 설치하려면 다음 필수 정보가 있어야 합니다.

- StorageGRID S3 엔드포인트의 IP 주소 또는 FQDN(정규화된 도메인 이름) 및 포트 번호입니다. HTTPS를 사용하는 경우 StorageGRID S3 엔드포인트에 구현된 사용자 지정 인증 기관(CA) 또는 자체 서명된 SSL 인증서를 사용합니다.
- 버킷 이름. 미리 존재해야 하며 비어 있어야 합니다.
- 버킷에 대한 읽기 및 쓰기 액세스를 통해 키 ID 및 비밀 액세스 키에 액세스합니다.

S3 엔드 포인트에 액세스하기 위한 권한 부여 파일 생성

S3 끝점에 액세스하기 위한 권한 부여 파일을 생성할 때는 다음과 같은 사전 요구 사항이 적용됩니다.

- Vertica가 설치되어 있습니다.
- 클러스터가 설정, 구성 및 준비되면 데이터베이스를 생성할 수 있습니다.

S3 끝점에 액세스하기 위한 권한 부여 파일을 생성하려면 다음 단계를 수행하십시오.

1. 'admintools'를 실행하여 Eon Mode 데이터베이스를 생성할 Vertica 노드에 로그인합니다.

   기본 사용자는 Vertica 클러스터 설치 중에 생성된 dbadmin입니다.

2. 텍스트 편집기를 사용하여 '/home/dbadmin' 디렉토리 아래에 파일을 만듭니다. 파일 이름은 'sg_auth.conf'와 같이 원하는 모든 것이 될 수 있습니다.
3. S3 엔드포인트가 표준 HTTP 포트 80 또는 HTTPS 포트 443을 사용하는 경우 포트 번호를 건너뜁니다. HTTPS를 사용하려면 다음 값을 설정합니다.

   - "awsenablehttps=1"을 선택하지 않으면 값을 "0"으로 설정합니다.
   - ``awauth=<S3 access key ID>:<secret access key>'

- ``awsendpoint=<StorageGRID S3 endpoint>:<port>'

  StorageGRID S3 엔드포인트 HTTPS 연결에 사용자 지정 CA 또는 자체 서명된 SSL 인증서를 사용하려면 인증서의 전체 파일 경로와 파일 이름을 지정합니다. 이 파일은 각 Vertica 노드의 동일한 위치에 있어야 하며 모든 사용자에 대한 읽기 권한이 있어야 합니다. StorageGRID S3 엔드포인트 SSL 인증서가 공개적으로 알려진 CA에 의해 서명된 경우 이 단계를 건너뜁니다.

  ``awscfile=<filepath/filename>'

  예를 들어, 다음 샘플 파일을 참조하십시오.

```
awsauth = MNVU4OYFAY2xyz123:03vuO4M4KmdfwffT8nqnBmnMVTr78Gu9wANabcxyz
awsendpoint = s3.england.connectlab.io:10443
awsenablehttps = 1
awscafile = /etc/custom-cert/grid.pem
```

+

> 운영 환경에서 고객은 공개적으로 알려진 CA가 서명한 서버 인증서를 StorageGRID S3 로드 밸런서 끝점에 구현해야 합니다.

모든 **Vertica** 노드에서 서비스 센터 경로를 선택합니다

서비스 센터 스토리지 경로에 대해 각 노드에서 디렉토리를 선택하거나 생성합니다. 서비스 센터 스토리지 경로 매개변수에 대해 제공한 디렉토리는 다음과 같아야 합니다.

- 클러스터의 모든 노드에서 동일한 경로(예: '/home/dbadmin/depot')
- dbadmin 사용자가 읽고 쓸 수 있습니다
- 충분한 보관

  기본적으로 Vertica는 depot 스토리지에 대한 디렉토리를 포함하는 파일 시스템 공간의 60%를 사용합니다. create_db 명령에서 '--depot-size' 인수를 사용하여 서비스 센터 크기를 제한할 수 있습니다. 을 참조하십시오 "Eon 모드 데이터베이스에 대한 Vertica 클러스터 크기 조정" 일반 Vertica 사이징 지침을 참조하거나 Vertica 어카운트 매니저에게 문의하십시오.

  admintools create_db" 도구는 서비스 센터 경로가 없는 경우 해당 경로를 생성하려고 시도합니다.

Eon 온프레미스 데이터베이스 생성

Eon 온프레미스 데이터베이스를 만들려면 다음 단계를 수행하십시오.

1. 데이터베이스를 생성하려면 admintools create_db 툴을 사용합니다.

   다음 목록에서는 이 예제에 사용된 인수에 대해 간략하게 설명합니다. 필수 인수와 선택적 인수에 대한 자세한 설명은 Vertica 문서를 참조하십시오.

   - 에서 생성된 권한 부여 파일의 -x <경로/파일 이름 "S3 끝점에 액세스하기 위한 권한 부여 파일 생성" 를 누릅니다.

인증 세부 정보는 성공적으로 생성된 후 데이터베이스 내에 저장됩니다. 이 파일을 제거하여 S3 비밀 키가 노출되지 않도록 할 수 있습니다.

- 공용 스토리지 위치 <S3://StorageGrid 버켓 이름>
- -s <이 데이터베이스에 사용할 Vertica 노드의 쉼표로 구분된 목록>
- d <생성할 데이터베이스 이름>
- 이 새 데이터베이스에 대해 설정할 -p <암호>. 예를 들어, 다음 샘플 명령을 참조하십시오.

```
admintools -t create_db -x sg_auth.conf --communal-storage
-location=s3://vertica --depot-path=/home/dbadmin/depot --shard
-count=6 -s vertica-vm1,vertica-vm2,vertica-vm3 -d vmart -p
'<password>'
```

새 데이터베이스를 생성하는 데는 데이터베이스의 노드 수에 따라 몇 분 정도 소요됩니다. 데이터베이스를 처음 만들 때 사용권 계약에 동의하라는 메시지가 표시됩니다.

예를 들어 다음 샘플 권한 부여 파일 및 'db 생성' 명령을 참조하십시오.

```
[dbadmin@vertica-vm1 ~]$ cat sg_auth.conf
awsauth = MNVU4OYFAY2CPKVXVxxxx:03vuO4M4KmdfwffT8nqnBmnMVTr78Gu9wAN+xxxx
awsendpoint = s3.england.connectlab.io:10445
awsenablehttps = 1

[dbadmin@vertica-vm1 ~]$ admintools -t create_db -x sg_auth.conf
--communal-storage-location=s3://vertica --depot-path=/home/dbadmin/depot
--shard-count=6 -s vertica-vm1,vertica-vm2,vertica-vm3 -d vmart -p
'xxxxxxxx'
Default depot size in use
Distributing changes to cluster.
    Creating database vmart
    Starting bootstrap node v_vmart_node0007 (10.45.74.19)
    Starting nodes:
        v_vmart_node0007 (10.45.74.19)
    Starting Vertica on all nodes. Please wait, databases with a large
catalog may take a while to initialize.
    Node Status: v_vmart_node0007: (DOWN)
    Node Status: v_vmart_node0007: (DOWN)
    Node Status: v_vmart_node0007: (DOWN)
    Node Status: v_vmart_node0007: (UP)
    Creating database nodes
    Creating node v_vmart_node0008 (host 10.45.74.29)
    Creating node v_vmart_node0009 (host 10.45.74.39)
    Generating new configuration information
    Stopping single node db before adding additional nodes.
```

```
    Database shutdown complete
    Starting all nodes
Start hosts = ['10.45.74.19', '10.45.74.29', '10.45.74.39']
    Starting nodes:
        v_vmart_node0007 (10.45.74.19)
        v_vmart_node0008 (10.45.74.29)
        v_vmart_node0009 (10.45.74.39)
    Starting Vertica on all nodes. Please wait, databases with a large
catalog may take a while to initialize.
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (DOWN) v_vmart_node0008: (DOWN)
v_vmart_node0009: (DOWN)
    Node Status: v_vmart_node0007: (UP) v_vmart_node0008: (UP)
v_vmart_node0009: (UP)
Creating depot locations for 3 nodes
Communal storage detected: rebalancing shards

Waiting for rebalance shards. We will wait for at most 36000 seconds.
Installing AWS package
    Success: package AWS installed
Installing ComplexTypes package
    Success: package ComplexTypes installed
Installing MachineLearning package
    Success: package MachineLearning installed
Installing ParquetExport package
    Success: package ParquetExport installed
Installing VFunctions package
    Success: package VFunctions installed
Installing approximate package
    Success: package approximate installed
Installing flextable package
    Success: package flextable installed
Installing kafka package
    Success: package kafka installed
Installing logsearch package
    Success: package logsearch installed
Installing place package
    Success: package place installed
Installing txtindex package
    Success: package txtindex installed
Installing voltagesecure package
```

```
    Success: package voltagesecure installed
Syncing catalog on vmart with 2000 attempts.
Database creation SQL tasks completed successfully. Database vmart created
successfully.
```

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
61입니다	S3://vertica/051/026d63a9d4a33237bf0e2c2a794a00a000021a07/026d63a9d4a33237bf0e2cf2a794a00a000021a07_0 DFS
145년	S 3://vertica/2c4/026d63a9d4a33237bf0e2c2cf2a794a00a000021a3d/026d63ae9d4a33237bf0e2c2cf2a794a00a000021a3d_0 DFS
146	S3://vertica/33c/026d63a9d4a33237bf0e2c2cf2a794a00a000021a1d/026d63a9d4a33237bf0e2c2cf2a794a00a000021a1d_0 DFS
40세	S3://vertica/382/026d63a9d4a33237bf0e2c2a794a00a000021a31/026d63a9d4a33237bf0e2c2a794a00a000021a31_0 DFS
145년	S 3://vertica/42F/026d63a9d4a33237bf0e2c2a794a00a000021a21/026d63ae9d4a33237bf0e2c2cf2a794a00a000021a21_0 DFS
34세	S3://vertica/472/026d63a9d4a33237bf0e2c2cf2a794a00a000021a25/026d63a9d4a33237bf0e2c2cf2a794a00a000021a25_0 DFS
41세	S 3://vertica/476/026d63a9d4a33237bf0e2c2cf2a794a00a000021a2d/026d63ae9d4a33237bf0e2c2cf2a794a00a000021a2d_0 DFS
61입니다	S3://vertica/52A/026d63a9d4a33237bf0e2c2cf2a794a00a000021a5d/026d63a9d4a33237bf0e2c2cf2a794a00a000021a5d_0 DFS
131입니다	S3://vertica/5d2/026d63a9d4a33237bf0e2c2cf2a794a00a000021a19/026d63a9d4a33237bf0e2c2cf2a794a00a000021a19_0 DFS

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
91세	S3://vertica/5f7/026d63a9d4a33237bf0e2c2cf2a794a00a000021a11/026d63a9d4a33237bf0e2c2cf2a794a00a000021a11_0 DFS
118입니다	S3://vertica/82d/026d63a9d4a33237bf0e2c2cf2a794a00a000021a15/026d63a9d4a33237bf0e2cf2a794a00a0000000021a15_0 DFS
115년	S3://vertica/9a2/026d63a9d4a33237bf0e2c2cf2a794a00a000021a61/026d63a9d4a33237bf0e2c2cf2a794a00a000021a61_0 DFS
33세	S 3://vertica/ACD/026d63a9d4a33237bf0e2c2a794a00a000021a29/026d63a9d4a33237bf0e2c2a794a00a000021a29_0 DFS
133입니다	S3://vertica/b98/026d63a9d4a33237bf0e2c2cf2a794a00a000021a4d/026d63a9d4a33237bf0e2c2cf2a794a00a000021a4d_0 DFS
38세	S 3://vertica/DB3/026d63a9d4a33237bf0e2c2cf2a794a00a000021a49/026d63ae9d4a33237bf0e2c2cf2a794a00a000021a49_0 DFS
38세	S3://vertica/EBA/026d63a9d4a33237bf0e2c2a794a00a000021a59/026d63ae9d4a33237bf0e2c2cf2a794a00a0000000021a59_0 DFS
21521920	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2c2cf2a794a00a0000215e2/026d63a33237bf0e2c2cf2a794a00a0000215e2
6865408	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2c2a794a00a000021602/026d63ae9d4a33237bf0e2c2cf2a794a00a000021602
204217344	S3://vertica/metadata/VMart/Libraries/026d63a9d4a33237bf0e2c2a794a00a000021610/026d63ae9d4a33237bf0e2c2cf2a794a00a000021610
16109056	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a0000217e0/026d63a9d4a33237bf0e2c2a794a00a0000217e0

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
12853248	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a000021800/026d63ae9d4a33237bf0e2c2cf2a794a00a000021800 tar
8937984	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2c2cf2a794a00a00002187a/026d63a33237bf0e2c2cf2a794a00a00002187a.
56260608	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a0000218b2/026d63a9d4a33237bf0e2c2a794a00a0000218b2
53947904	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2c2cf2a794a00a0000219ba/026d63a33237bf0e2c2cf2a794a00a0000219ba
44932608	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a0000219de/026d63a33237bf0e2c2cf2a794a00a0000219de
256306688	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a000021a6e/026d63a33237bf0e2c2a794a00a000021a6e
8062464	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a000021e34/026d63a9d4a33237bf0e2c2cf2a794a00a000021e34
20024832	S3://vertica/metadata/VMart/Libraries/026d63ae9d4a33237bf0e2cf2a794a00a000021e70/026d63a9d4a33237bf0e2c2cf2a794a00a000021e70
10444	S 3://vertica/metadata/VMart/cluster_config.json
823266	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C13/chkpt_1.cat.gz`
254년	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C13/Completed

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
2958	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C2_2/chkpt_1.cat.gz`
231	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C2_2/Completed
822521	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C4_chkpt_1.cat.gz`
231	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C4/Completed
746513	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_14_g14.cat`
2596	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_3_g3.cat.gz`
821065	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_4_g4.cat.gz`
6440	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_5_g5.cat`
8518	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_8_g8.cat`

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
0	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/tiered_catalog.cat`
822922	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C14_7/chkpt_1.cat.gz`
232입니다	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C14_7/Completed
822930	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_14_g7.cat.gz`
755033	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_15_g8.cat`
0	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/Catalog/859703b06a3456d95d0be28575a673/tiered_catalog.cat`
822922	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C14_7/chkpt_1.cat.gz`
232입니다	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Checkpoints/C14_7/Completed
822930	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_14_g7.cat.gz`

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
755033	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/Txnlogs/txn_15_g8.cat`
0	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/Catalog/859703b06a3456d95d0be28575a673/tiered_catalog.cat`

스트리밍 제한 비활성화

이 절차는 다른 온프레미스 오브젝트 스토리지에 대한 Vertica 가이드를 기반으로 하며 StorageGRID에 적용할 수 있어야 합니다.

1. 데이터베이스를 만든 후 AWSStreamingConnectionPercentage 구성 매개 변수를 0으로 설정하여 비활성화합니다. 이 설정은 공용 스토리지가 있는 Eon 모드 온-프레미스 설치에는 필요하지 않습니다. 이 구성 매개 변수는 Vertica가 스트리밍 읽기에 사용하는 개체 저장소에 대한 연결 수를 제어합니다. 클라우드 환경에서 이 설정은 오브젝트 저장소에서 데이터를 스트리밍하는 데 사용 가능한 모든 파일 핸들을 사용하지 않도록 도와줍니다. 이 경우 일부 파일 핸들을 다른 오브젝트 저장소 작업에 사용할 수 있습니다. 온프레미스 오브젝트 저장소의 대기 시간이 짧기 때문에 이 옵션이 필요하지 않습니다.
2. 매개 변수 값을 업데이트하려면 "vsql" 문을 사용합니다. 암호는 “온-프레미스 데이터베이스 만들기”에서 설정한 데이터베이스 암호입니다. 예를 들어, 다음 샘플 출력을 참조하십시오.

```
[dbadmin@vertica-vm1 ~]$ vsql
Password:
Welcome to vsql, the Vertica Analytic Database interactive terminal.
Type:   \h or \? for help with vsql commands
        \g or terminate with semicolon to execute query
        \q to quit
dbadmin=> ALTER DATABASE DEFAULT SET PARAMETER
AWSStreamingConnectionPercentage = 0; ALTER DATABASE
dbadmin=> \q
```

물류창고 설정을 확인하는 중입니다

Vertica 데이터베이스 기본 서비스 센터 설정은 읽기 및 쓰기 작업에 대해 활성화됩니다(값 = 1). 성능 향상을 위해 이러한 서비스 센터 설정을 유지할 것을 적극 권장합니다.

```
vsql -c 'show current all;' | grep -i UseDepot
DATABASE | UseDepotForReads | 1
DATABASE | UseDepotForWrites | 1
```

샘플 데이터 로드(옵션)

이 데이터베이스가 테스트용으로 제거되는 경우 테스트를 위해 이 데이터베이스에 샘플 데이터를 로드할 수 있습니다. Vertica는 각 Vertica 노드의 '/opt/vertica/examples/VMart_Schema/'에 있는 샘플 데이터 세트 VMart와 함께 제공됩니다. 이 샘플 데이터 집합에 대한 자세한 내용을 확인할 수 있습니다 "여기".

다음 단계에 따라 샘플 데이터를 로드합니다.

1. Vertica 노드 중 하나에 dbadmin으로 로그인합니다. cd /opt/vertica/examples/VMart_Schema/
2. 데이터베이스에 예제 데이터를 로드하고 하위 단계 c 및 d에 프롬프트가 표시되면 데이터베이스 암호를 입력합니다.
   a. 'cd/opt/vertica/examples/VMart_Schema'를 선택합니다
   b. './vmart_gen'
   c. "vsql<vmart_define_schema.sql"을 참조하십시오
   d. "vsql <vmart_load_data.sql"을 선택합니다
3. 미리 정의된 SQL 쿼리가 여러 개 있습니다. 일부 쿼리를 실행하여 테스트 데이터가 데이터베이스에 성공적으로 로드되었는지 확인할 수 있습니다. 예: ``vsql<vmart_queries1.sql'

추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대해 자세히 알아보려면 다음 문서 및/또는 웹 사이트를 검토하십시오.

- "NetApp StorageGRID 11.7 제품 문서"
- "StorageGRID 데이터 시트"
- "Vertica 10.1 제품 설명서"

버전 기록

버전	날짜	문서 버전 기록
버전 1.0	2021년 9월	최초 릴리스.

안젤라 청 _ 에 의해

ELK 스택을 사용한 StorageGRID 로그 분석

StorageGRID 11.6 syslog 전달 기능을 사용하면 외부 syslog 서버를 구성하여 StorageGRID 로그 메시지를 수집하고 분석할 수 있습니다. 엘크(Elasticsearch, Logstash, Kibana)는 가장 인기 있는 로그 분석 솔루션 중 하나가 되었습니다. 를 참조하십시오 "ELK 비디오를 사용한 StorageGRID 로그 분석" 샘플 ELK 구성 및 장애 발생 S3 요청을 식별하고 해결하는 데 사용할 수 있는 방법을 확인합니다. 이 문서에서는 Logstash 구성, Kibana 쿼리, 차트 및 대시보드의 예제 파일을 제공하여 StorageGRID 로그 관리 및 분석을 빠르게 시작할 수 있도록 합니다.

요구 사항

- StorageGRID 11.6.0.2 이상
- 엘크(Elasticsearch, Logstash 및 Kibana) 7.1x 이상 설치 및 작동 중

샘플 파일

- "Logstash 7.x 샘플 파일 패키지를 다운로드합니다" + * MD5 체크섬 * 148c23d0021d9a4b4a6c0287464deab + * SHA256 체크섬 * f51ec9e2e3f842d5a7861566ba561bbbb4373038b4e7b3b3b3d522adf2d6
- "Logstash 8.x 샘플 파일 패키지를 다운로드합니다" + * MD5 체크섬 * e11bae3a662f87c310ef363d0fe06835+ * SHA256 체크섬 * 5c670755742cfdf5a723a596ba087e0153a65baef3934afdb682f61cd278d

가정

독자는 StorageGRID 및 ELK 용어와 운영에 대해 잘 알고 있습니다.

지침

두 가지 샘플 버전은 그로크 패턴으로 정의된 이름의 차이로 인해 제공됩니다. 예를 들어, Logstash 구성 파일의 SYSLOGBASE 그로크 패턴은 설치된 Logstash 버전에 따라 필드 이름을 다르게 정의합니다.

```
match => {"message" => '<%{POSINT:syslog_pri}>%{SYSLOGBASE}
%{GREEDYDATA:msg-details}'}
```

- Logstash 7.17 샘플 *

Field	Value
_id	7C1MaYEBRH8UbfKnIls8
_index	sgrid2-2022.06.15
_score	-
_type	_doc
@timestamp	Jun 15, 2022 @ 17:36:46.038
host	grid2-site2-s1
logsource	SITE2-S1
msg-details	Reloading syslog service
pid	628
program	update-sysl
syslog_pri	37
timestamp	Jun 15 21:36:46

- Logstash 8.23 샘플 *

Table JSON

Search field names

Actions	Field	Value
	_id	yuhOiIEBVP6KX4EwqcyU
	_index	sglog-2022.06.21
	_score	-
	@timestamp	Jun 21, 2022 @ 18:07:45.444
	event.original	<28>Jun 21 22:07:45 SITE2-S3 ADE: syslog messages being dropped
	host.hostname	SITE2-S3
	msg-details	syslog messages being dropped
	process.name	ADE
	syslog_pri	28
	timestamp	Jun 21 22:07:45

- 단계 *

1. 설치된 엘크 버전을 기반으로 제공된 샘플의 압축을 풉니다. + 샘플 폴더에는 + * sglog-2-file.conf: * 이 구성 파일은 데이터 변환 없이 Logstash의 파일에 StorageGRID 로그 메시지를 출력합니다. 이 옵션을 사용하여 로그 스타재가 StorageGRID 메시지를 수신하는지 확인하거나 StorageGRID 로그 패턴을 이해하는 데 도움을 줄 수 있습니다. + * sglog-2-es.conf: * 이 구성 파일은 다양한 패턴과 필터를 사용하여 StorageGRID 로그 메시지를 변환합니다. 여기에는 패턴 또는 필터를 기반으로 메시지를 드롭하는 Drop 문 예가 포함되어 있습니다. 인덱싱을 위해 Elasticsearch로 출력이 전송됩니다. + 파일 내의 명령에 따라 선택한 구성 파일을 사용자 지정합니다.

2. 사용자 지정 구성 파일 테스트:

```
/usr/share/logstash/bin/logstash --config.test_and_exit -f <config-
file-path/file>
```

마지막으로 반환된 줄이 아래 줄과 비슷하면 구성 파일에 구문 오류가 없는 것입니다.

```
[LogStash::Runner] runner - Using config.test_and_exit mode. Config
Validation Result: OK. Exiting Logstash
```

3. /etc/logstash/logstash.yml에서 config.reload.automatic 를 활성화하지 않은 경우 Logstash 서버의 config:/etc/logstash/conf.d+에 사용자 지정된 conf 파일을 복사합니다. 그렇지 않으면 구성 다시 로드 간격이 경과될 때까지 기다립니다.

```
 grep reload /etc/logstash/logstash.yml
# Periodically check if the configuration has changed and reload the
pipeline
config.reload.automatic: true
config.reload.interval: 5s
```

4. /var/log/logstash/logstash-plain.log 를 확인하고 새 구성 파일로 Logstash를 시작하는 데 오류가 없는지 확인합니다.

5. TCP 포트가 시작되고 수신 중인지 확인합니다. + 이 예에서는 TCP 포트 5000이 사용됩니다.

```
netstat -ntpa | grep 5000
tcp6       0      0 :::5000                 :::*
LISTEN      25744/java
```

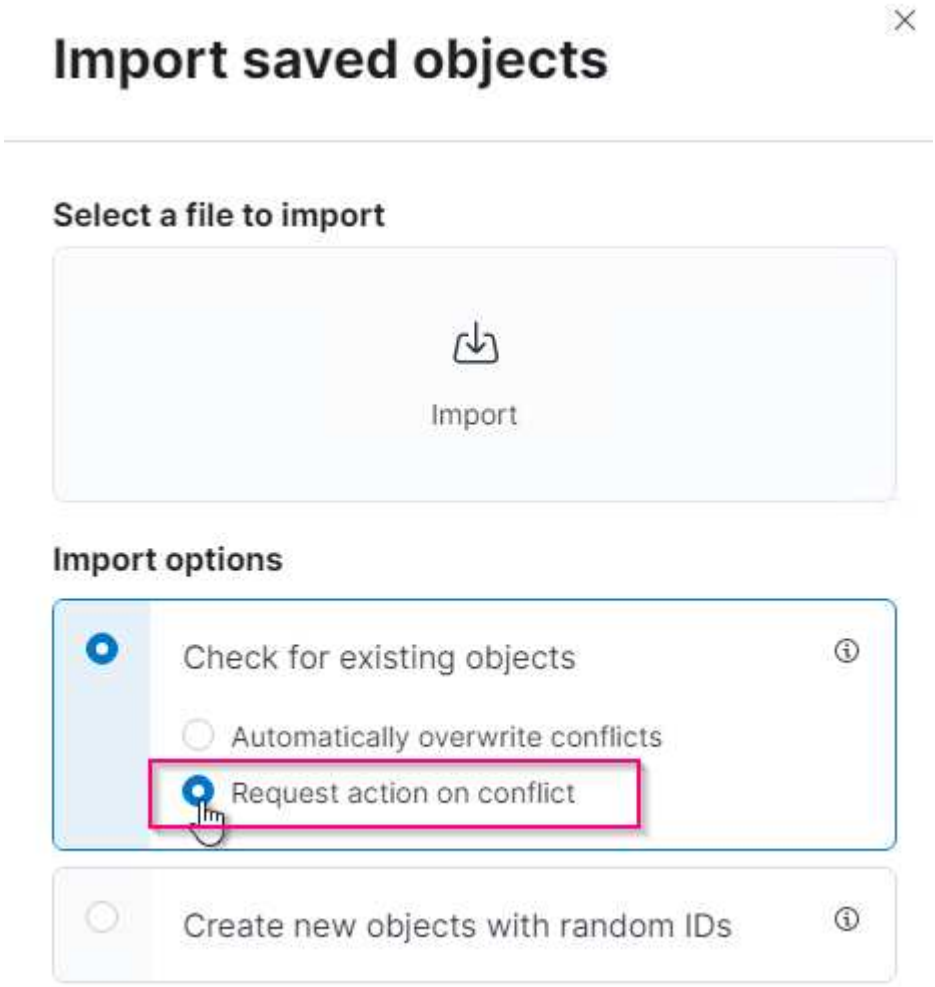
6. StorageGRID 관리자 GUI에서 로그 메시지를 Logstash로 보내도록 외부 syslog 서버를 구성합니다. 을 참조하십시오 "데모 비디오" 를 참조하십시오.

7. 정의된 TCP 포트에 대한 StorageGRID 노드 연결을 허용하려면 Logstash 서버에서 방화벽을 구성하거나 비활성화해야 합니다.

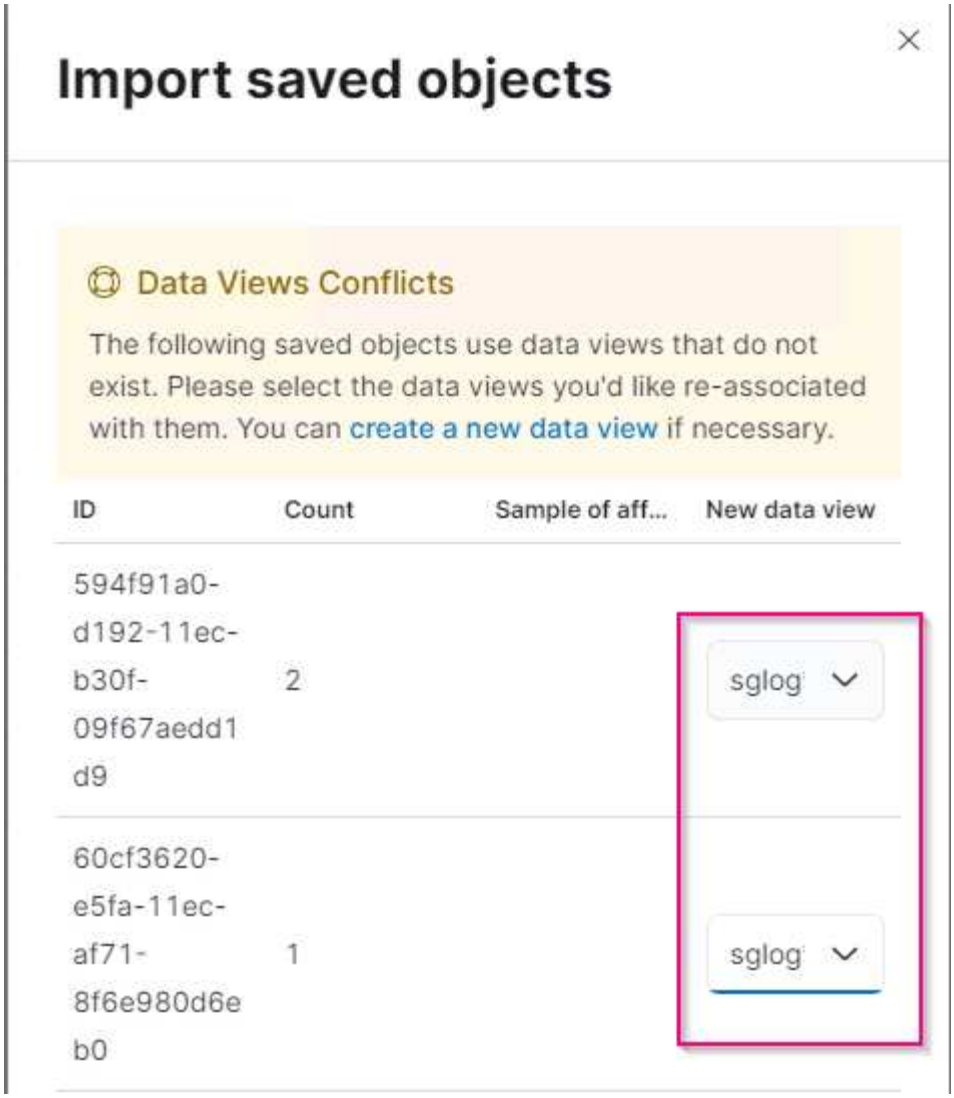
8. Kibana GUI에서 관리 → 개발 도구 를 선택합니다. 콘솔 페이지에서 이 가져오기 명령을 실행하여 Elasticsearch에 새 인덱스가 생성되었는지 확인합니다.

```
GET /_cat/indices/*?v=true&s=index
```

9. 키바나 GUI에서 인덱스 패턴(엘크 7.x) 또는 데이터 뷰(엘크 8.x)를 생성합니다.
10. Kibana GUI의 상단 중앙에 있는 검색 상자에 '저장된 개체'를 입력합니다. + 저장된 개체 페이지에서 가져오기를 선택합니다. 불러오기 옵션 아래에서 '충돌 시 작업 요청'을 선택합니다.



ELK <version> -query-chart-sample.ndjson을 가져옵니다. + 충돌을 해결할 것인지 묻는 메시지가 나타나면 8단계에서 만든 색인 패턴이나 데이터 보기를 선택합니다.



다음 Kibana 객체를 가져옵니다. + * 쿼리 * + * 감사-메시지 -s3rq -orlm + * bycast 로그 S3 관련 메시지 + * 로그 수준 경고 이상 + * 실패한 보안 이벤트 + * 차트 * + * bycast.log 기반 S3 요청 수 + * HTTP 상태 코드 + * 유형별 감사 메시지 분석 + * 평균 S3 응답 위 차트를 사용하는 시간 + * 대시보드 * + * S3 요청 대시보드

이제 Kibana를 사용하여 StorageGRID 로그 분석을 수행할 준비가 되었습니다.

추가 리소스

- "syslog101"
- "엘크 스택"
- "Grok 패턴 목록"
- "Logstash:Grok에 대한 초보자용 가이드"
- "Logstash: syslog 심층 탐구 실습 가이드"
- "Kibana 가이드 – 문서 탐색"
- "StorageGRID 감사 로그 메시지 참조"

안젤라 청 _ 에 의해

Prometheus 및 Grafana를 사용하여 메트릭 보존 기간을 연장합니다

이 기술 보고서에서는 외부 Prometheus 및 Grafana 서비스를 사용하여 NetApp StorageGRID 11.6을 구성하는 방법에 대해 자세히 설명합니다.

소개

StorageGRID는 Prometheus를 사용하여 메트릭을 저장하고 내장된 Grafana 대시보드를 통해 이러한 메트릭의 시각화를 제공합니다. Prometheus 메트릭은 클라이언트 액세스 인증서를 구성하고 지정된 클라이언트에 대한 Prometheus 액세스를 활성화하여 StorageGRID에서 안전하게 액세스할 수 있습니다. 오늘날 이 메트릭 데이터의 보존은 관리 노드의 스토리지 용량에 의해 제한됩니다. 이러한 메트릭의 사용자 지정 시각화를 생성하는 데 더 오랜 시간이 걸릴 뿐만 아니라 사용자 지정 시각화를 만들기 위해 새 Prometheus 및 Grafana 서버를 구축하고, StorageGRID 인스턴스에서 메트릭을 스크레핑하도록 새 서버를 구성하고, 우리에게 중요한 메트릭이 포함된 대시보드를 만들 것입니다. 에서 수집된 Prometheus 메트릭에 대한 자세한 정보를 확인할 수 있습니다 "StorageGRID 설명서".

프로메테우스 연방

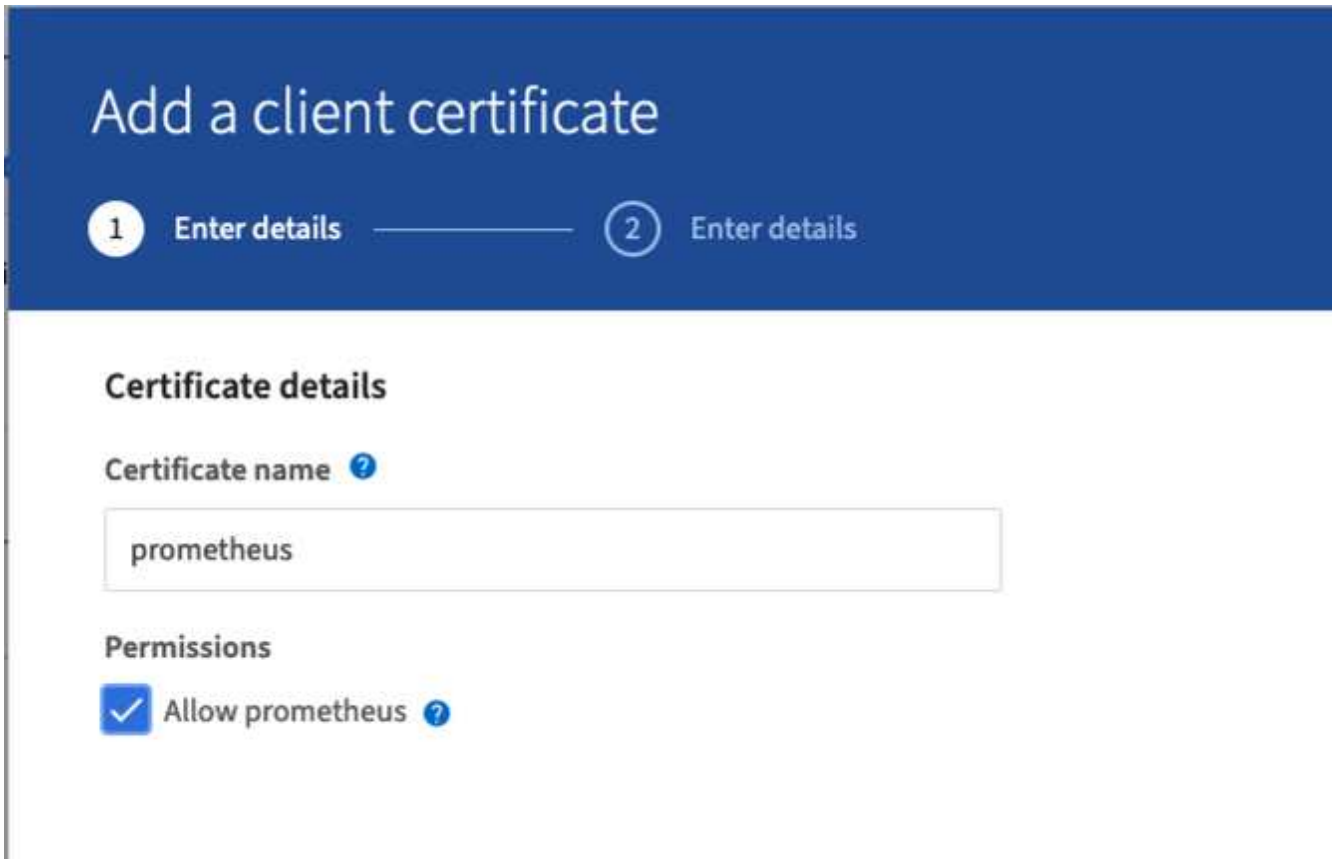
실습 세부 정보

이 예제에서는 StorageGRID 11.6 노드와 Debian 11 서버에 모든 가상 시스템을 사용합니다. StorageGRID 관리 인터페이스는 공개적으로 신뢰할 수 있는 CA 인증서로 구성됩니다. 이 예제에서는 StorageGRID 시스템 또는 Debian Linux 설치의 설치 및 구성을 사용하지 않습니다. Prometheus 및 Grafana에서 지원하기를 원하는 Linux의 맛을 사용할 수 있습니다. Prometheus와 Grafana는 모두 Docker 컨테이너로 설치하거나, 소스에서 구축하거나, 사전 컴파일된 바이너리로 구축할 수 있습니다. 이 예제에서는 Prometheus와 Grafana 바이너리를 동일한 Debian 서버에 직접 설치합니다. 의 기본 설치 지침을 다운로드하여 따릅니다 https://prometheus.io 및 https://grafana.com/grafana/ 각각

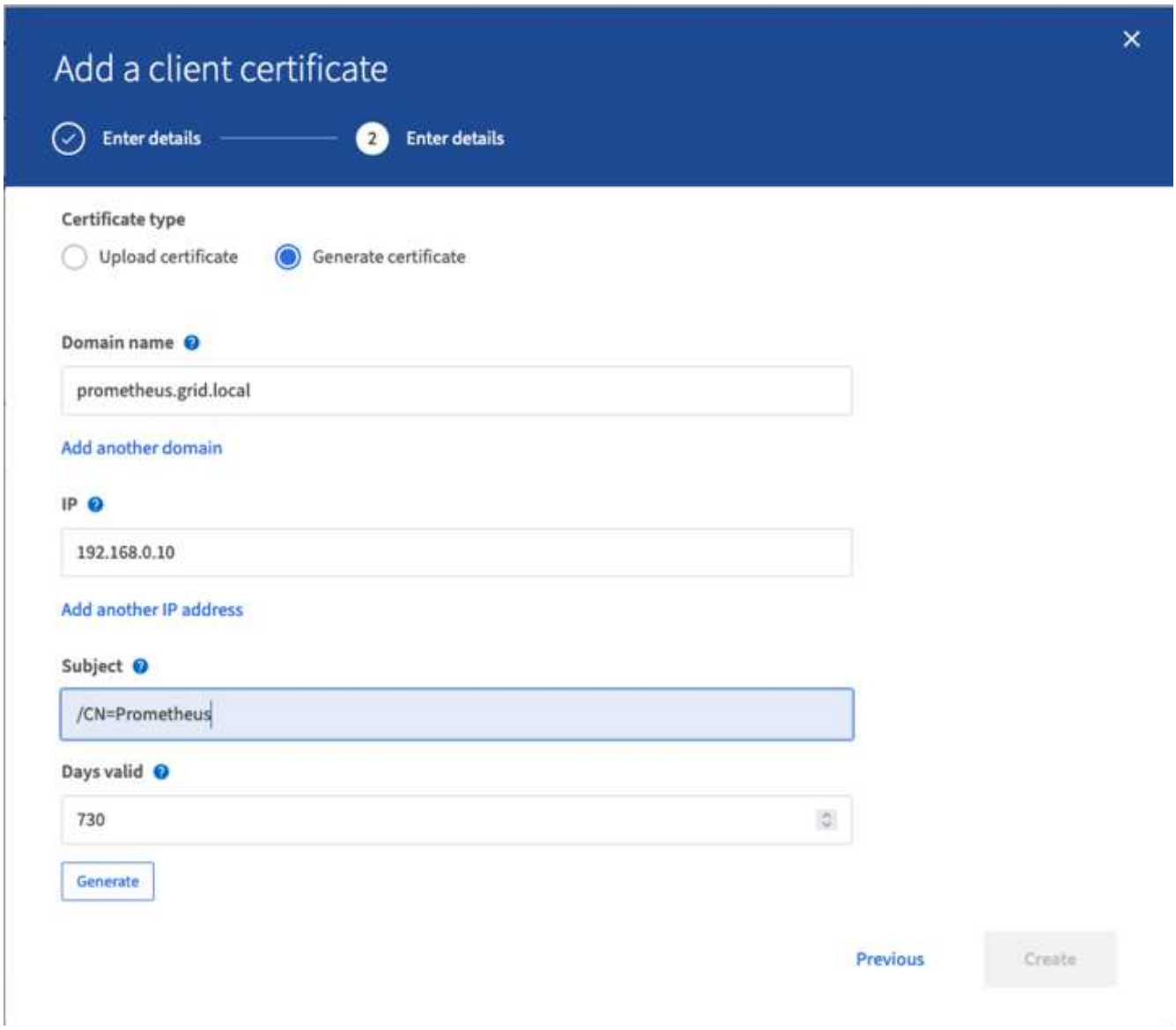
Prometheus 클라이언트 액세스를 위해 **StorageGRID**를 구성합니다

StorageGRID에 저장된 Prometheus 메트릭에 액세스하려면 개인 키가 있는 클라이언트 인증서를 생성하거나 업로드하고 클라이언트에 대한 권한을 활성화해야 합니다. StorageGRID 관리 인터페이스에는 SSL 인증서가 있어야 합니다. 이 인증서는 신뢰할 수 있는 CA에서 또는 자체 서명된 경우 수동으로 신뢰할 수 있는 Prometheus 서버에 의해 신뢰되어야 합니다. 자세한 내용은 를 참조하십시오 "StorageGRID 설명서".

1. StorageGRID 관리 인터페이스의 왼쪽 아래에서 "구성"을 선택하고 "보안" 아래의 두 번째 열에서 인증서를 클릭합니다.
2. 인증서 페이지에서 "클라이언트" 탭을 선택하고 "추가" 버튼을 클릭합니다.
3. 액세스 권한이 부여된 클라이언트의 이름을 제공하고 이 인증서를 사용합니다. "권한" 아래의 "Prometheus 허용" 앞의 상자를 클릭하고 계속 단추를 클릭합니다.

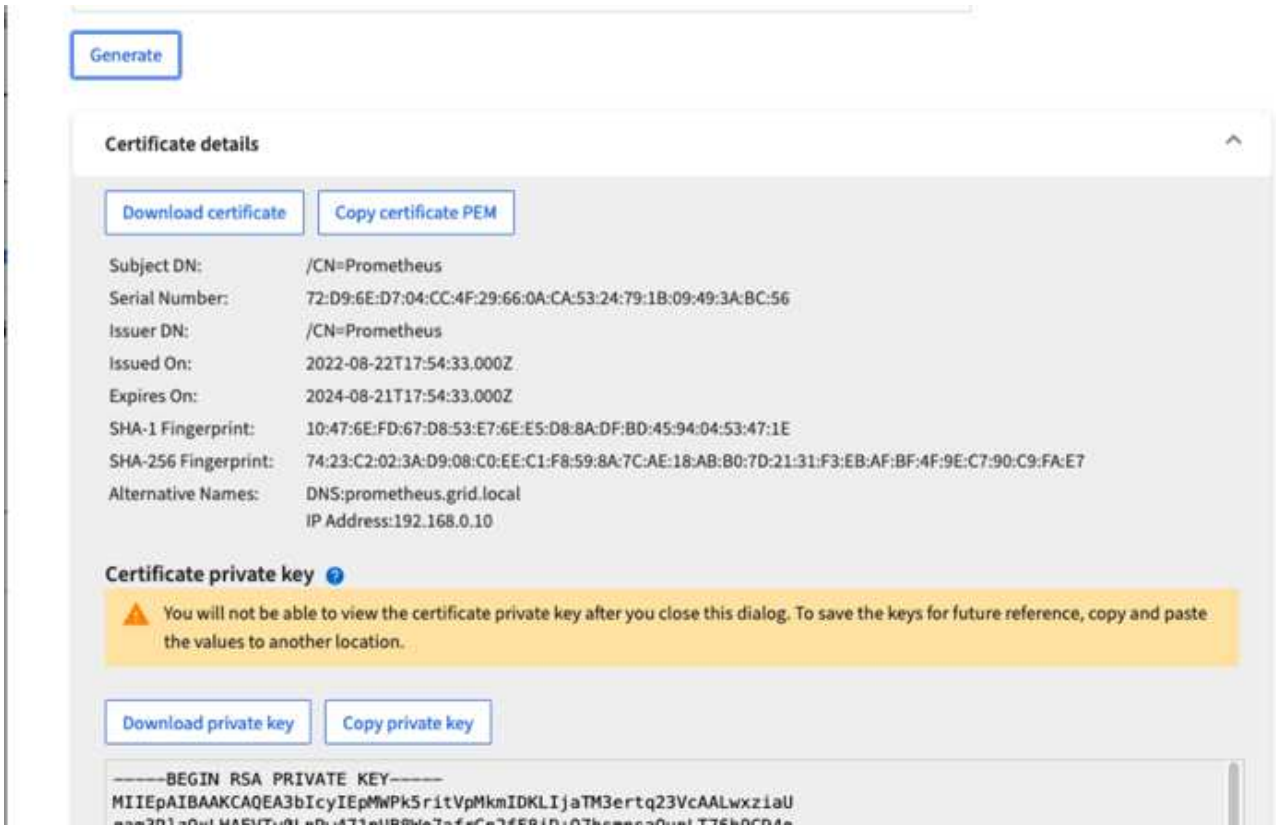


4. CA 서명 인증서가 있는 경우 "인증서 업로드" 라디오 버튼을 선택할 수 있지만, 여기서는 "인증서 생성" 라디오 버튼을 선택하여 StorageGRID가 클라이언트 인증서를 생성하도록 할 것입니다. 필수 필드가 입력되어 표시됩니다. 클라이언트 서버의 FQDN, 서버의 IP, 제목 및 유효한 날짜를 입력합니다. 그런 다음 "생성" 버튼을 클릭합니다.




```
Be mindful of the certificate days valid entry as you will need
to renew this certificate in both StorageGRID and the Prometheus
server before it expires to maintain uninterrupted collection.
```

1. 인증서 PEM 파일과 개인 키 PEM 파일을 다운로드합니다.



> This is the only time you can download the private key, so make sure you do not skip this step.

Prometheus 설치를 위해 **Linux** 서버를 준비합니다

Prometheus를 설치하기 전에 Prometheus 사용자, 디렉토리 구조를 사용하여 환경을 준비하고 메트릭 스토리지 위치의 용량을 구성하려고 합니다.

1. Prometheus 사용자를 생성합니다.

```
sudo useradd -M -r -s /bin/false Prometheus
```

2. Prometheus, 클라이언트 인증서 및 메트릭 데이터용 디렉토리를 생성합니다.

```
sudo mkdir /etc/Prometheus /etc/Prometheus/cert /var/lib/Prometheus
```

3. ext4 파일 시스템을 사용하여 메트릭 보존을 위해 사용 중인 디스크를 포맷했습니다.

```
mkfs -t ext4 /dev/sdb
```

4. 그런 다음 Prometheus 메트릭 디렉토리에 파일 시스템을 마운트했습니다.

```
sudo mount -t auto /dev/sdb /var/lib/prometheus/
```

5. 메트릭 데이터에 사용 중인 디스크의 uuid를 가져옵니다.

```
sudo ls -al /dev/disk/by-uuid/
   lrwxrwxrwx 1 root root   9 Aug 18 17:02 9af2c5a3-bfc2-4ec1-85d9-
ebab850bb4a1 -> ../../sdb
```

6. /etc/fstab/에 항목을 추가하면 /dev/sdb 의 uuuid 를 사용하여 재부팅 후에도 마운트가 유지됩니다.

```
/etc/fstab
UUID=9af2c5a3-bfc2-4ec1-85d9-ebab850bb4a1 /var/lib/prometheus   ext4
defaults    0   0
```

Prometheus를 설치하고 구성합니다

이제 서버가 준비되었으므로 Prometheus 설치를 시작하고 서비스를 구성할 수 있습니다.

1. Prometheus 설치 패키지의 압축을 풉니다

```
tar xzf prometheus-2.38.0.linux-amd64.tar.gz
```

2. 바이너리를 /usr/local/bin에 복사하고 소유권을 이전에 만든 Prometheus 사용자로 변경합니다

```
sudo cp prometheus-2.38.0.linux-amd64/{prometheus,promtool}
/usr/local/bin
sudo chown prometheus:prometheus /usr/local/bin/{prometheus,promtool}
```

3. 콘솔 및 라이브러리를 /etc/Prometheus에 복사합니다

```
sudo cp -r prometheus-2.38.0.linux-amd64/{consoles,console_libraries}
/etc/prometheus/
```

4. StorageGRID에서 이전에 다운로드한 클라이언트 인증서 및 개인 키 PEM 파일을 /etc/Prometheus/certs로 복사합니다
5. Prometheus 구성 YAML 파일을 생성합니다

```
sudo nano /etc/prometheus/prometheus.yml
```

6. 다음 설정을 삽입합니다. 작업 이름은 원하는 모든 것이 될 수 있습니다. "-targets:[']"를 관리 노드의 FQDN으로 변경하고 인증서 및 개인 키 파일 이름의 이름을 변경한 경우 TLS_config 섹션이 일치하도록 업데이트하십시오. 그런 다음 파일을 저장합니다. 그리드 관리 인터페이스에서 자체 서명된 인증서를 사용하는 경우 인증서를 다운로드하여 고유한 이름의 클라이언트 인증서와 함께 놓고 TLS_config 섹션에서 ca_file:/etc/Prometheus/cert/UCERT.pem을 추가합니다

   a. 이 예에서는 alertmanager, cassandra, node 및 StorageGRID로 시작하는 모든 메트릭을 수집합니다. Prometheus 메트릭에 대한 자세한 내용은 에서 확인할 수 있습니다 "StorageGRID 설명서".

```
# my global config
global:
  scrape_interval: 60s # Set the scrape interval to every 15 seconds.
Default is every 1 minute.

scrape_configs:
  - job_name: 'StorageGRID'
    honor_labels: true
    scheme: https
    metrics_path: /federate
    scrape_interval: 60s
    scrape_timeout: 30s
    tls_config:
      cert_file: /etc/prometheus/cert/certificate.pem
      key_file: /etc/prometheus/cert/private_key.pem
    params:
      match[]:
        -
'{__name__=~"alertmanager_.*|cassandra_.*|node_.*|storagegrid_.*"}'
    static_configs:
    - targets: ['sgdemo-rtp.netapp.com:9091']
```

> 그리드 관리 인터페이스에서 자체 서명된 인증서를 사용하는 경우 인증서를 다운로드하여 고유한 이름의 클라이언트 인증서와 함께 배치합니다. TLS_config 섹션에서 클라이언트 인증서 및 개인 키 줄 위에 인증서를 추가합니다
>
> ```
> ca_file: /etc/prometheus/cert/UIcert.pem
> ```

1. /etc/Prometheus 및 /var/lib/Prometheus에 있는 모든 파일 및 디렉토리의 소유권을 Prometheus 사용자로 변경합니다

```
sudo chown -R prometheus:prometheus /etc/prometheus/
sudo chown -R prometheus:prometheus /var/lib/prometheus/
```

2. /etc/systemd/system에서 Prometheus 서비스 파일을 생성합니다

```
sudo nano /etc/systemd/system/prometheus.service
```

3. 다음 줄을 삽입하고 메트릭 데이터의 보존 기간을 1년으로 설정하는 #- storage.tsdb.retention.time=1y#를 확인합니다. 또는 #- storage.sdb.retention.size=300GiB#를 사용하여 스토리지 제한에 따라 기본 보존을 수행할 수도 있습니다. 메트릭 보존을 설정할 수 있는 유일한 위치입니다.

```
[Unit]
Description=Prometheus Time Series Collection and Processing Server
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
        --config.file /etc/prometheus/prometheus.yml \
        --storage.tsdb.path /var/lib/prometheus/ \
        --storage.tsdb.retention.time=1y \
        --web.console.templates=/etc/prometheus/consoles \
        --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

4. 새 Prometheus 서비스를 등록하려면 시스템 서비스를 다시 로드하십시오. 그런 다음 Prometheus 서비스를 시작하고 활성화합니다.

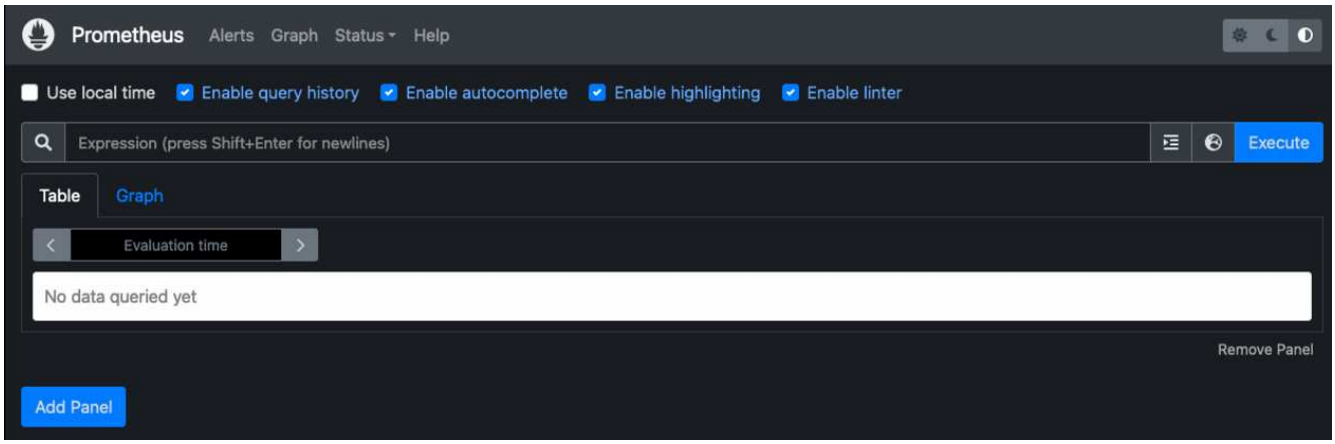
```
sudo systemctl daemon-reload
sudo systemctl start prometheus
sudo systemctl enable prometheus
```

5. 서비스가 올바르게 실행되는지 확인합니다

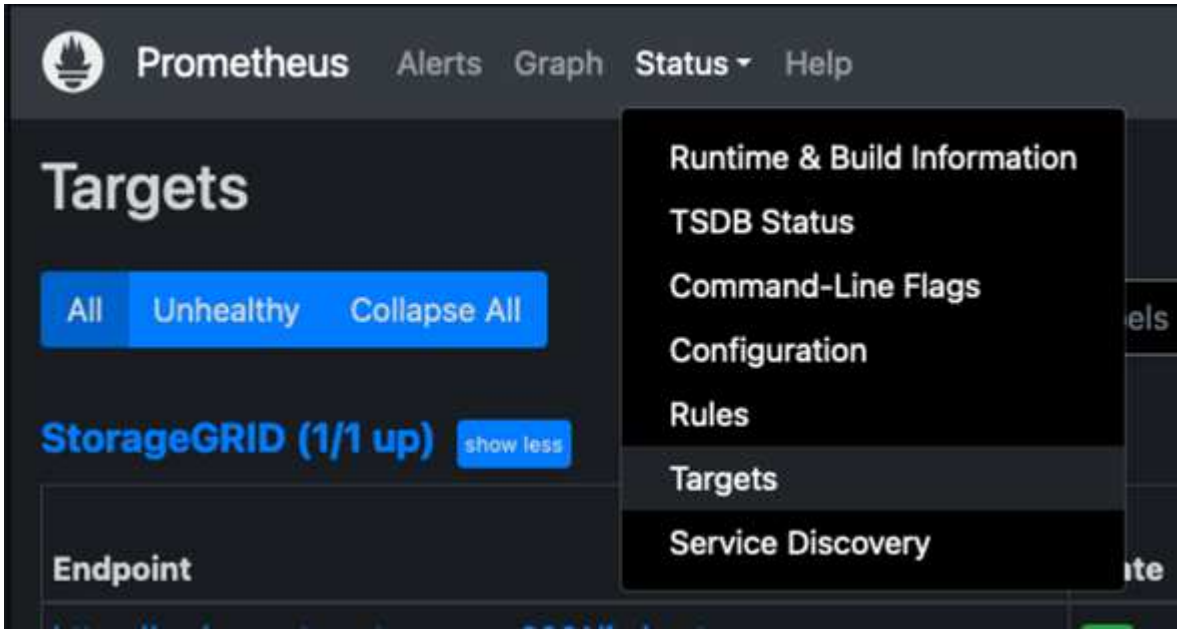
```
sudo systemctl status prometheus
```

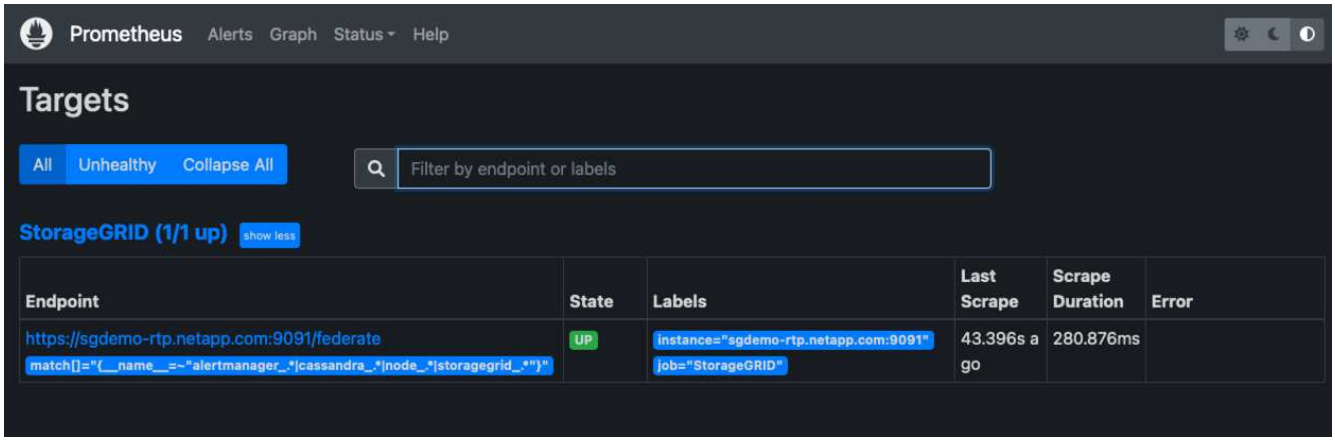
```
● prometheus.service - Prometheus Time Series Collection and Processing
Server
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled;
vendor preset: enabled)
     Active: active (running) since Mon 2022-08-22 15:14:24 EDT; 2s ago
   Main PID: 6498 (prometheus)
      Tasks: 13 (limit: 28818)
     Memory: 107.7M
        CPU: 1.143s
     CGroup: /system.slice/prometheus.service
             └─6498 /usr/local/bin/prometheus --config.file
/etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/
--web.console.templates=/etc/prometheus/consoles --web.con>

Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.510Z caller=head.go:544 level=info component=tsdb
msg="Replaying WAL, this may take a while"
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:615 level=info component=tsdb msg="WAL
segment loaded" segment=0 maxSegment=1
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:615 level=info component=tsdb msg="WAL
segment loaded" segment=1 maxSegment=1
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:621 level=info component=tsdb msg="WAL
replay completed" checkpoint_replay_duration=55.57µs wal_rep>
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.831Z caller=main.go:997 level=info fs_type=EXT4_SUPER_MAGIC
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.831Z caller=main.go:1000 level=info msg="TSDB started"
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.831Z caller=main.go:1181 level=info msg="Loading
configuration file" filename=/etc/prometheus/prometheus.yml
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.832Z caller=main.go:1218 level=info msg="Completed loading
of configuration file" filename=/etc/prometheus/prometheus.y>
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.832Z caller=main.go:961 level=info msg="Server is ready to
receive web requests."
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.832Z caller=manager.go:941 level=info component="rule
manager" msg="Starting rule manager..."
```

6. 이제 Prometheus 서버의 UI로 이동할 수 있습니다 http://Prometheus-server:9090 UI를 참조하십시오

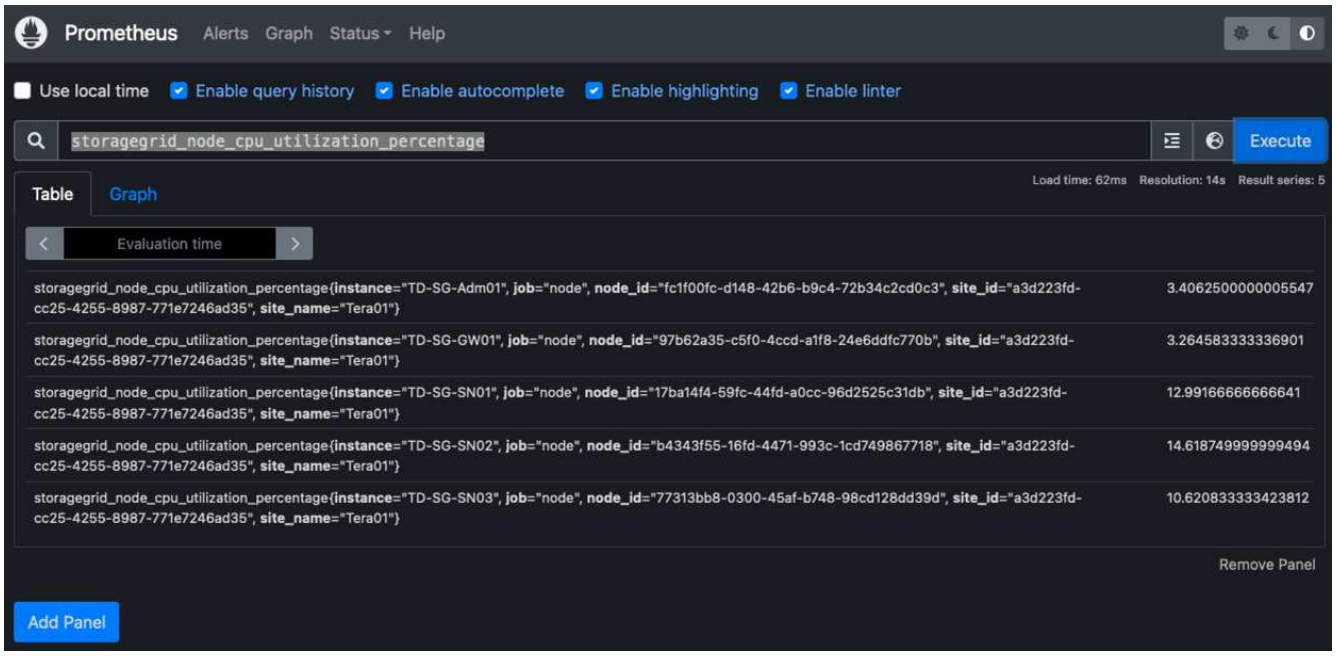


7. "상태" 대상 아래에서 Prometheus.yml에서 구성한 StorageGRID 끝점의 상태를 볼 수 있습니다





8. 그래프 페이지에서 테스트 쿼리를 실행하고 데이터가 스크레핑되었는지 확인할 수 있습니다. 예를 들어 쿼리 표시줄에 "StorageGrid_node_cpu_Utilization_percentage"를 입력하고 실행 단추를 클릭합니다.



Grafana 설치 및 구성

Prometheus가 설치되고 작동되었으므로 Grafana 설치 및 대시보드 구성으로 이동할 수 있습니다

Grafana 인스턴션

1. Grafana의 최신 Enterprise Edition을 설치합니다

```
sudo apt-get install -y apt-transport-https
sudo apt-get install -y software-properties-common wget
sudo wget -q -O /usr/share/keyrings/grafana.key
https://packages.grafana.com/gpg.key
```

2. 안정적인 릴리스를 위해 이 리포지토리를 추가합니다.

```
echo "deb [signed-by=/usr/share/keyrings/grafana.key]
https://packages.grafana.com/enterprise/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list
```

3. 리포지토리를 추가한 후

```
sudo apt-get update
sudo apt-get install grafana-enterprise
```

4. 새 이식편 서비스를 등록하려면 시스템 서비스를 다시 로드하십시오. 그런 다음 Grafana 서비스를 시작 및 활성화합니다.

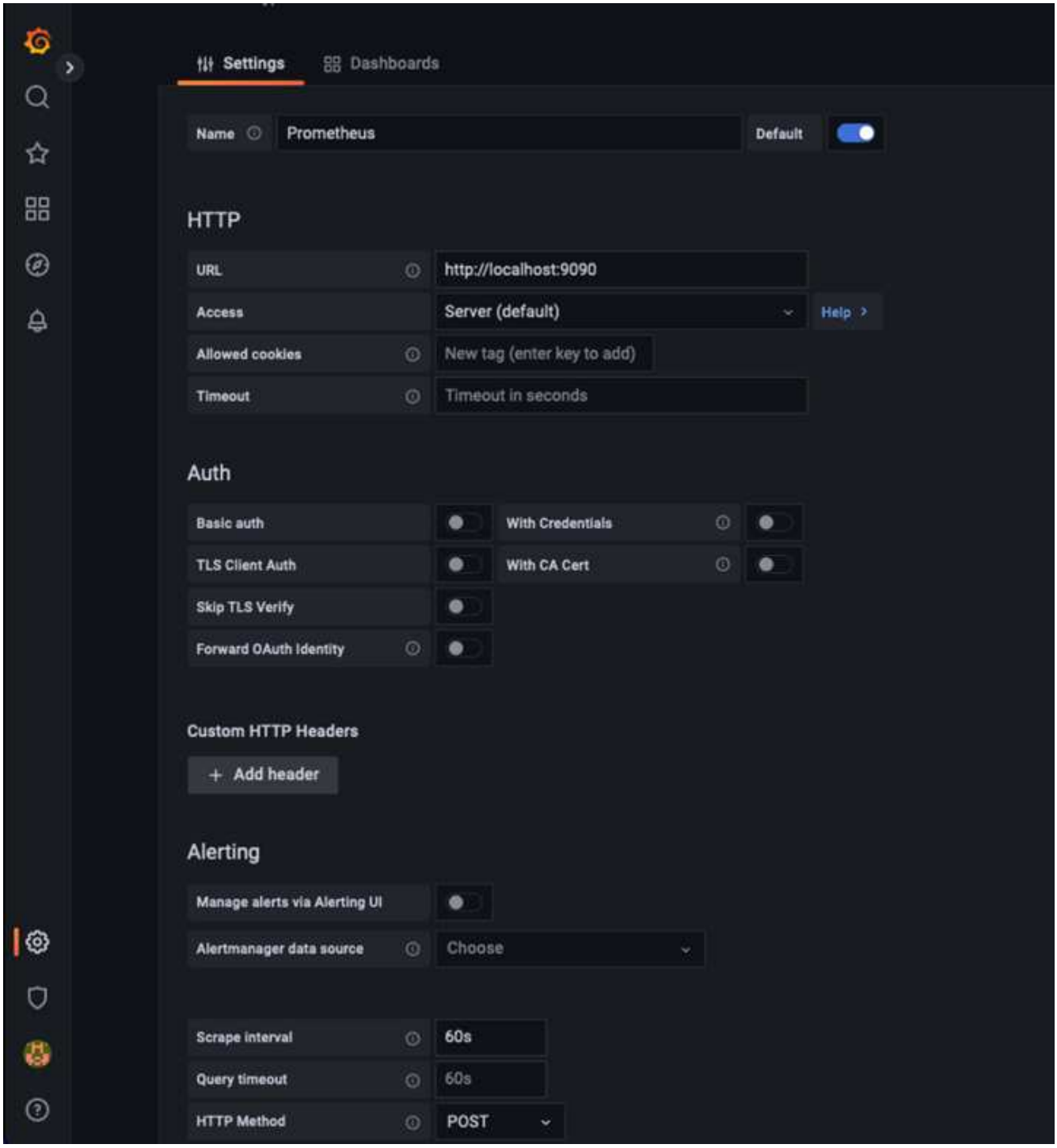
```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl enable grafana-server.service
```

5. Grafana가 이제 설치 및 실행 중입니다. 브라우저를 열고 HTTP://Prometheus-server:3000을 열면 Grafana 로그인 페이지가 표시됩니다.
6. 기본 로그인 자격 증명은 admin/admin이며, 메시지가 표시되면 새 암호를 설정해야 합니다.

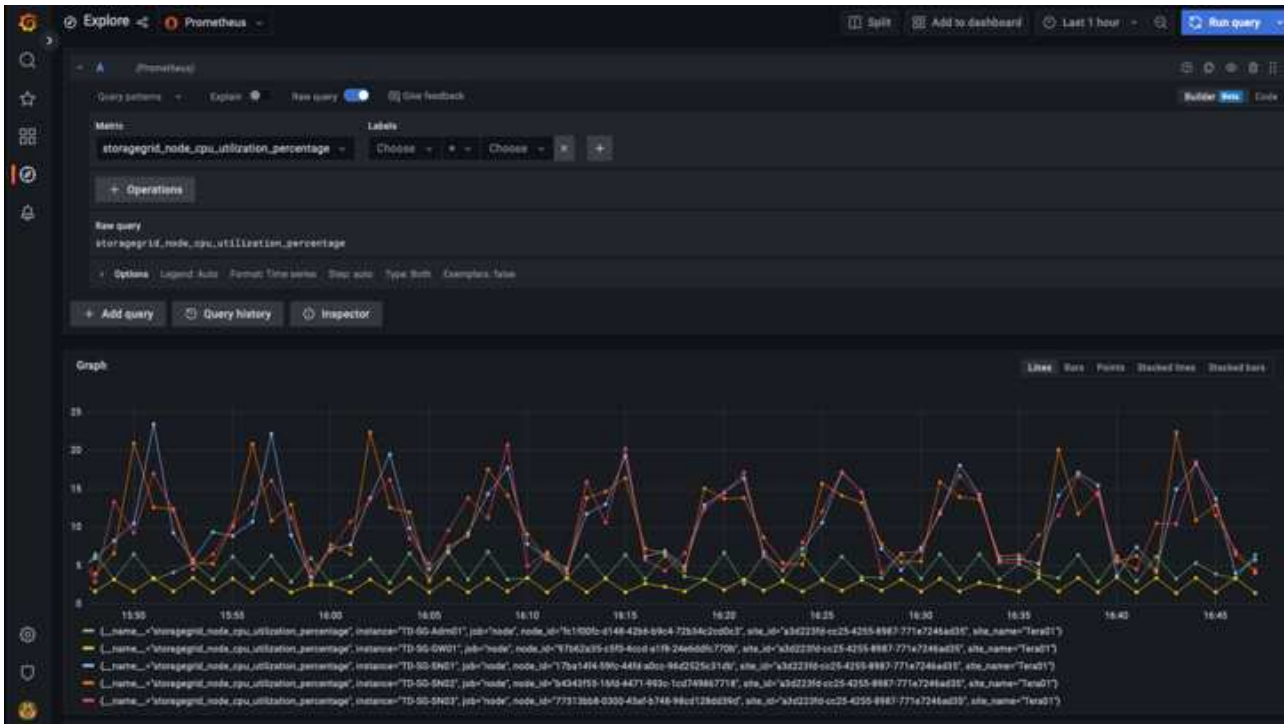
StorageGRID에 대한 **Grafana** 대시보드를 생성합니다

Grafana와 Prometheus가 설치 및 실행되었으므로 이제 데이터 소스를 생성하고 대시보드를 구축하여 두 가지를 연결할 시간입니다

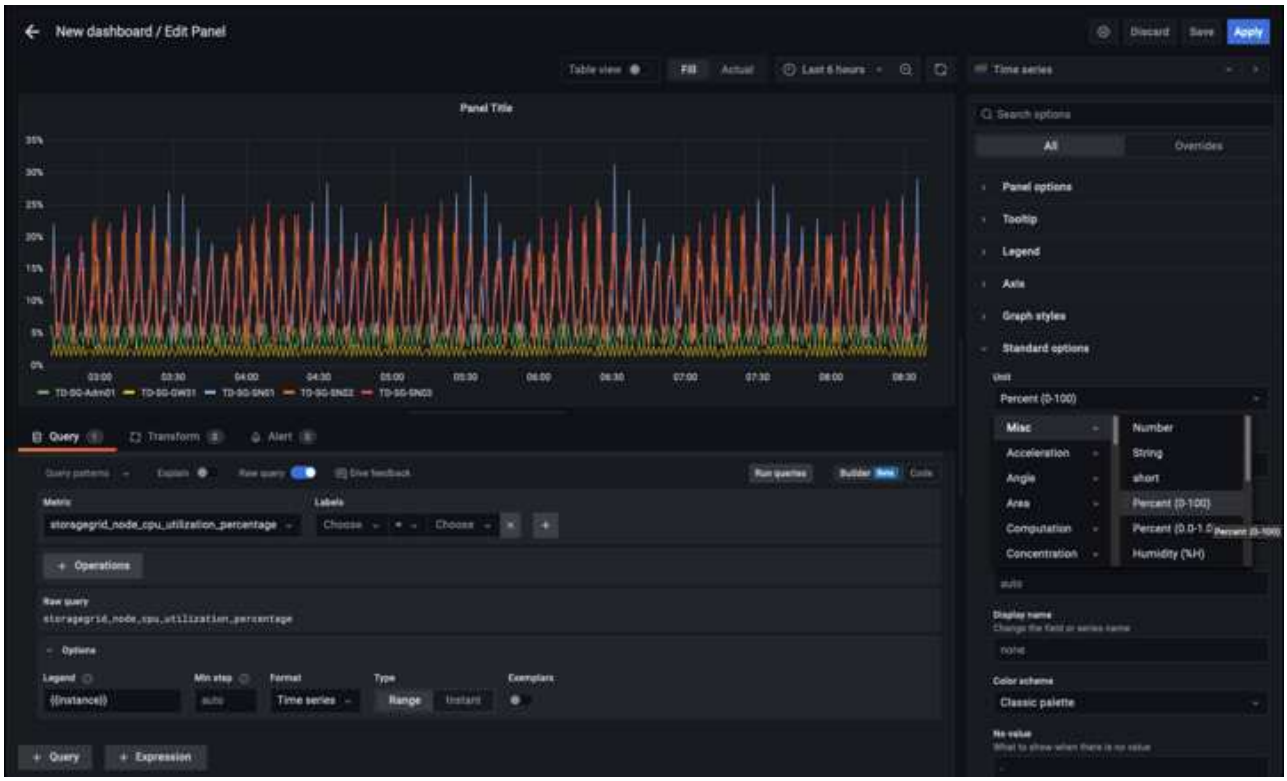
1. 왼쪽 창에서 "구성"을 확장하고 "데이터 소스"를 선택한 다음 "데이터 소스 추가" 버튼을 클릭합니다
2. Prometheus는 최고의 데이터 소스 중 하나가 될 것입니다. 그렇지 않은 경우 검색 표시줄을 사용하여 "Prometheus"를 찾습니다.
3. Prometheus 인스턴스의 URL과 Prometheus 간격에 맞게 스크레핑 간격을 입력하여 Prometheus 소스를 구성합니다. Prometheus에서 경고 관리자를 구성하지 않았기 때문에 알림 섹션도 비활성화했습니다.



4. 원하는 설정을 입력한 후 아래로 스크롤하여 "Save & Test(저장 및 테스트)"를 클릭합니다.
5. 구성 테스트가 완료되면 탐색 버튼을 클릭합니다.
   a. 탐색 창에서 Prometheus를 "StorageGrid_node_cpu_Utilization_percentage"로 테스트한 것과 동일한 메트릭을 사용하고 "쿼리 실행" 단추를 클릭할 수 있습니다



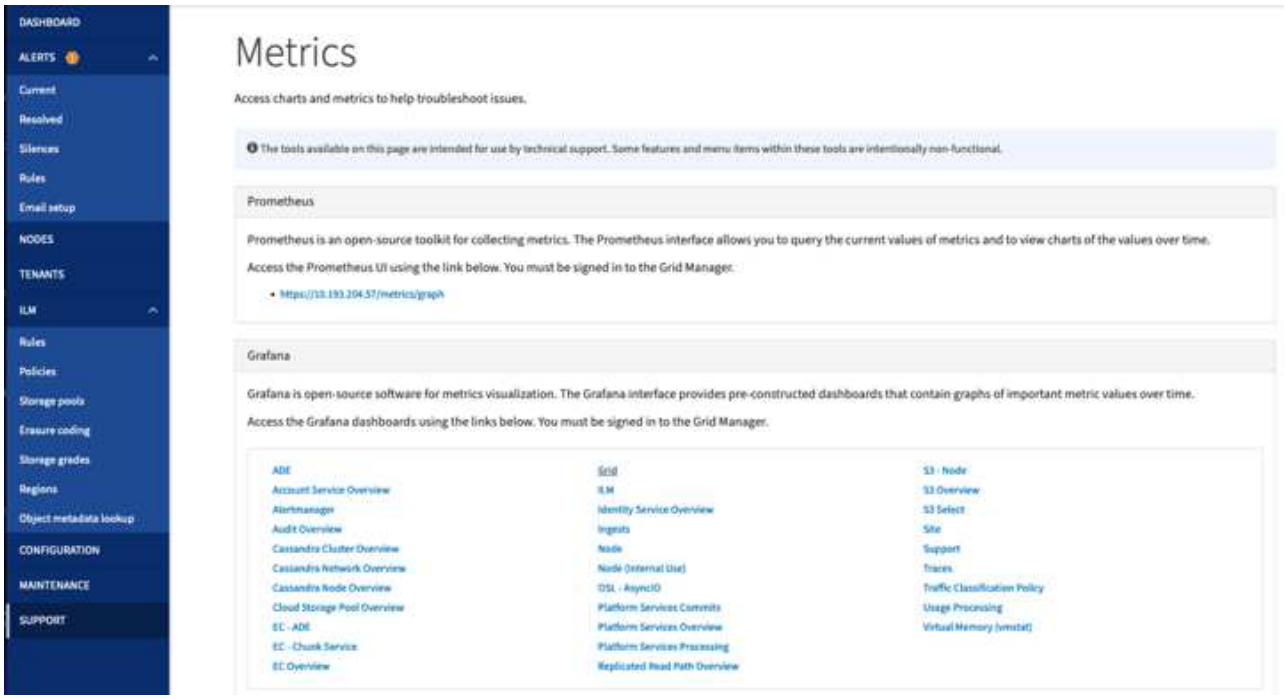
6. 이제 데이터 소스가 구성되었으므로 대시보드를 생성할 수 있습니다.
   a. 왼쪽 창에서 "대시보드"를 확장하고 "+새 대시보드"를 선택합니다.
   b. "Add a new panel(새 패널 추가)"을 선택합니다.
   c. 메트릭을 선택하여 새 패널을 구성합니다. 다시 "StorageGrid_node_cpu_Utilization_percentage"를 사용하고, 패널 제목을 입력하고, 하단에 있는 "Options"를 확장하고, 범례를 사용자 지정으로 변경하려면 "{{instance}"를 입력하고, 오른쪽 창에 "Standard options"에서 "Unit"을 "Misc/Percent(0-100)"로 설정합니다. 그런 다음 "적용"을 클릭하여 패널을 대시보드에 저장합니다.



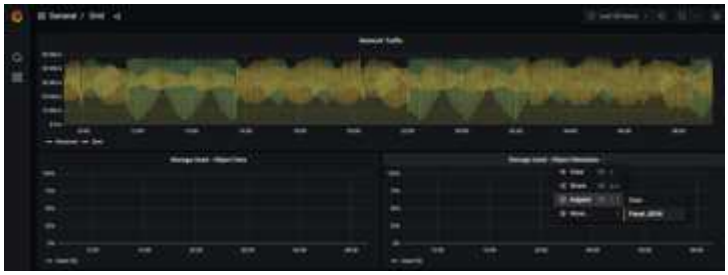
7. 원하는 각 메트릭에 대해 이러한 대시보드를 계속 구축할 수 있지만 다행히 StorageGRID에는 사용자 지정 대시보드에 복사할 수 있는 패널이 포함된 대시보드가 이미 있습니다.

   a. StorageGRID 관리 인터페이스의 왼쪽 창에서 "지원"을 선택하고 "도구" 열 아래쪽에서 "메트릭"을 클릭합니다.

   b. 메트릭스 내에서 중간 열의 맨 위에 있는 "Grid" 링크를 선택하겠습니다.

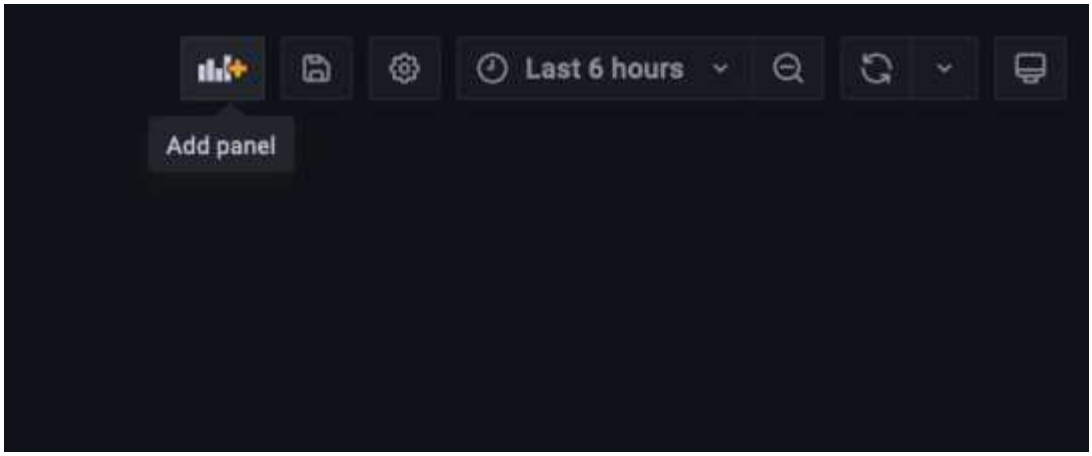


   c. Grid 대시보드에서 "Storage Used - Object Metadata" 패널을 선택합니다. 작은 아래쪽 화살표 및 패널 제목 끝을 클릭하여 메뉴를 드롭다운합니다. 이 메뉴에서 "검사" 및 "패널 JSON"을 선택합니다.



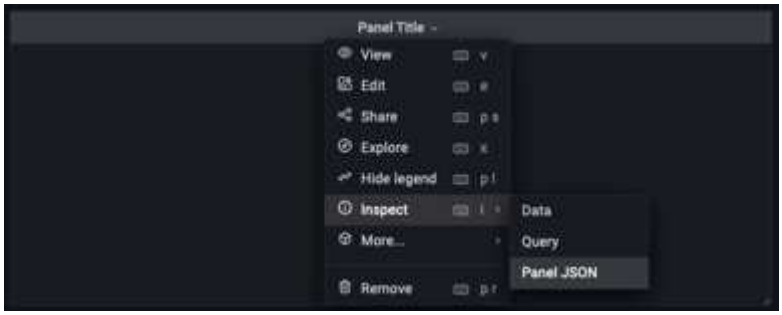
   d. JSON 코드를 복사하고 창을 닫습니다.

e. 새 대시보드에서 아이콘을 클릭하여 새 패널을 추가합니다.

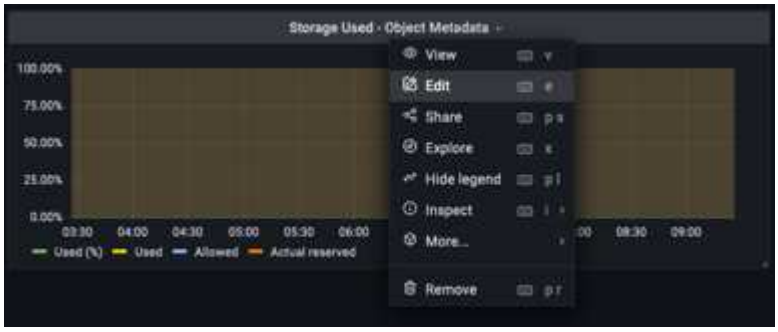


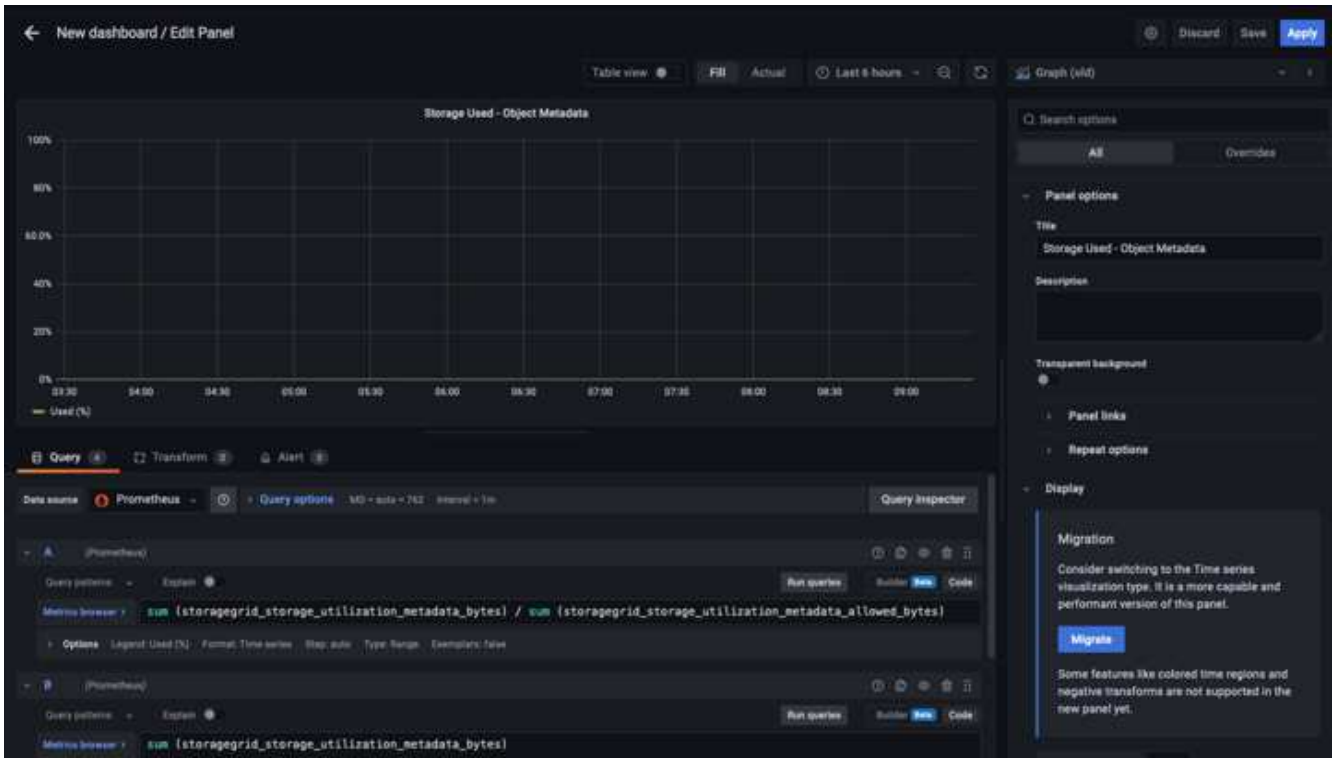
f. 변경하지 않고 새 패널을 적용합니다

g. StorageGRID 패널과 마찬가지로 JSON을 검사하십시오. JSON 코드를 모두 제거하고 StorageGRID 패널에서 복사한 코드로 교체합니다.



h. 새 패널을 편집하면 오른쪽에 "migrate(마이그레이션)" 버튼이 있는 Migration(마이그레이션) 메시지가 표시됩니다. 버튼을 클릭한 다음 "적용" 버튼을 클릭합니다.





8. 모든 패널이 제자리에 있고 원하는 대로 구성되면 오른쪽 위에 있는 디스크 아이콘을 클릭하여 대시보드를 저장하고 대시보드에 이름을 지정합니다.

결론

이제 Prometheus 서버에 맞춤형 데이터 보존 및 스토리지 용량을 추가할 수 있습니다. 이를 통해 운영 관련 메트릭이 포함된 자체 대시보드를 지속적으로 구축할 수 있습니다. 에서 수집된 Prometheus 메트릭에 대한 자세한 정보를 확인할 수 있습니다 "StorageGRID 설명서".

_ 아론 클라인 _

Datadog SNMP 구성

StorageGRID SNMP 메트릭 및 트랩을 수집하도록 데이터 독을 구성합니다.

데이터 독을 구성합니다

Datadog는 메트릭, 시각화 및 알림을 제공하는 모니터링 솔루션입니다. 다음 구성은 Ubuntu 22.04.1 호스트에서 Linux 에이전트 버전 7.43.1을 사용하여 StorageGRID 시스템에 로컬로 배포되었습니다.

StorageGRID MIB 파일에서 생성된 **Datadog** 프로파일 및 트랩 파일입니다

Datadog는 제품 MIB 파일을 SNMP 메시지를 매핑하는 데 필요한 datadog 참조 파일로 변환하는 방법을 제공합니다.

발견된 지침에 따라 생성된 데이터 독그 트랩 해결 매핑에 대한 StorageGRID YAML 파일입니다 "여기". +이 파일을 /etc/datadog-agent/conf.d/snmp.d/trap_db/+에 넣습니다

- "TRAP YAML 파일을 다운로드합니다" 를 누릅니다

- * MD5 체크섬 * 42e27e4210719945a46172b98c379517+
- * SHA256 checksum * d0f5c8e6c3c902d054f854b70a85f928cba8b7c76391d356f05d2cf73b6887+

이 StorageGRID 프로필 YAML 파일은 발견된 지침에 따라 생성된 데이터 독그 메트릭 매핑에 대해 생성됩니다 "여기". +이 파일을 /etc/datadog-agent/conf.d/snmp.d/profiles/+에 넣습니다

- "YAML 프로필 파일을 다운로드합니다" 를 누릅니다
  - * MD5 체크섬 * 72bb7784f4801adda4e0c3ea77df19aa+
  - * SHA256 체크섬 * b6b7fadd33063422a8b8e39b3ead8ab38349ee0229926eadc8585f0087b8cee+

메트릭의 **SNMP** 데이터 독이 구성됩니다

메트릭에 대한 SNMP 구성은 두 가지 방법으로 관리할 수 있습니다. StorageGRID 시스템이 포함된 네트워크 주소 범위를 제공하거나 개별 장치의 IP를 정의하여 자동 검색을 구성할 수 있습니다. 구성 위치는 결정에 따라 다릅니다. 자동 검색은 데이터 로그 에이전트 YAML 파일에서 정의됩니다. 명시적 장치 정의는 SNMP 구성 YAML 파일에 구성되어 있습니다. 다음은 동일한 StorageGRID 시스템에 대한 각 의 예입니다.

자동 검색

구성은 /etc/datadog-agent/datadog.YAML에 있습니다

```
listeners:
  - name: snmp
snmp_listener:
  workers: 100  # number of workers used to discover devices concurrently
  discovery_interval: 3600  # interval between each autodiscovery in
seconds
  loader: core  # use core check implementation of SNMP integration.
recommended
  use_device_id_as_hostname: true  # recommended
  configs:
    - network_address: 10.0.0.0/24  # CIDR subnet
      snmp_version: 2
      port: 161
      community_string: 'st0r@gegrid'  # enclose with single quote
      profile: netapp-storagegrid
```

개별 장치

/etc/datadog-agent/conf.d/snmp.d/conf.yaML

```
init_config:
  loader: core  # use core check implementation of SNMP integration.
recommended
  use_device_id_as_hostname: true  # recommended
instances:
- ip_address: '10.0.0.1'
  profile: netapp-storagegrid
  community_string: 'st0r@gegrid'  # enclose with single quote
- ip_address: '10.0.0.2'
  profile: netapp-storagegrid
  community_string: 'st0r@gegrid'
- ip_address: '10.0.0.3'
  profile: netapp-storagegrid
 community_string: 'st0r@gegrid'
- ip_address: '10.0.0.4'
  profile: netapp-storagegrid
  community_string: 'st0r@gegrid'
```

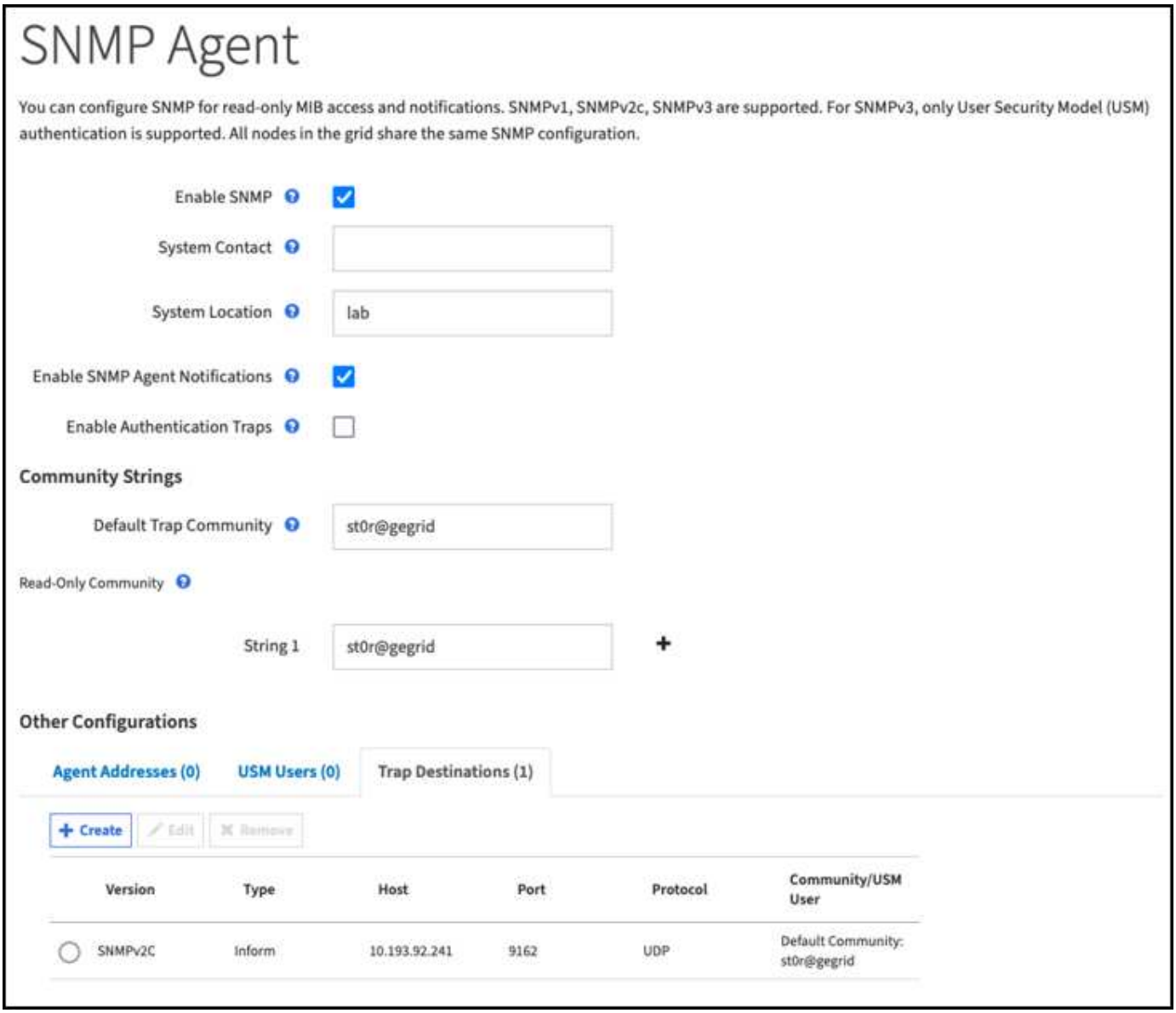
트랩에 대한 **SNMP** 구성

SNMP 트랩에 대한 구성은 datadog 구성 YAML 파일 /etc/datadog-agent/datadog.YAML에서 정의됩니다

```
network_devices:
  namespace:  # optional, defaults to “default”.
  snmp_traps:
    enabled: true
    port: 9162 # on which ports to listen for traps
    community_strings: # which community strings to allow for v2 traps
      - st0r@gegrid
```

StorageGRID SNMP 구성의 예

StorageGRID 시스템의 SNMP 에이전트는 구성 탭, 모니터링 열 아래에 있습니다. SNMP를 활성화하고 원하는 정보를 입력합니다. 트랩을 구성하려면 "트랩 대상"을 선택하고 트랩 구성을 포함하는 데이터 독그 에이전트 호스트의 대상을 생성합니다.



_ 아론 클라인 _

rclone을 사용하여 StorageGRID에서 개체를 마이그레이션, 저장 및 삭제합니다

rclone은 S3 작업을 위한 무료 명령줄 도구 및 클라이언트입니다. rclone을 사용하여 StorageGRID에서 오브젝트 데이터를 마이그레이션, 복사, 삭제할 수 있습니다. rclone에는 아래 예와 같이 "퍼지" 기능을 사용하여 비어 있지 않은 경우에도 버킷을 삭제할 수 있는 기능이 포함되어 있습니다.

rclone을 설치하고 구성합니다

워크스테이션 또는 서버에 rclone을 설치하려면 에서 다운로드하십시오 "rclone.org".

초기 구성 단계

1. config 스크립트를 실행하거나 수동으로 파일을 생성하여 rclone 구성 파일을 생성합니다.
2. 이 예에서는 rclone 구성에서 원격 StorageGRID S3 엔드포인트 이름에 sgdemo를 사용합니다.
   a. 구성 파일 ~/.config/rclone/rclone.conf를 생성합니다

```
[sgdemo]
type = s3
provider = Other
access_key_id = ABCDEFGH123456789JKL
secret_access_key = 123456789ABCDEFGHIJKLMN0123456789PQRST+V
endpoint = sgdemo.netapp.com
```

   b. rclone 구성을 실행합니다

rclone config

```
2023/04/13 14:22:45 NOTICE: Config file
"/root/.config/rclone/rclone.conf" not found - using defaults
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> sgdemo
```

```
Option Storage.
Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
 1 / 1Fichier
   \ "fichier"
 2 / Alias for an existing remote
   \ "alias"
 3 / Amazon Drive
   \ "amazon cloud drive"
 4 / Amazon S3 Compliant Storage Providers including AWS,
Alibaba, Ceph, Digital Ocean, Dreamhost, IBM COS, Minio,
SeaweedFS, and Tencent COS
   \ "s3"
 5 / Backblaze B2
   \ "b2"
 6 / Better checksums for other remotes
   \ "hasher"
 7 / Box
   \ "box"
 8 / Cache a remote
   \ "cache"
 9 / Citrix Sharefile
   \ "sharefile"
10 / Compress a remote
   \ "compress"
11 / Dropbox
   \ "dropbox"
12 / Encrypt/Decrypt a remote
   \ "crypt"
13 / Enterprise File Fabric
   \ "filefabric"
14 / FTP Connection
```

```
   \ "ftp"
15 / Google Cloud Storage (this is not Google Drive)
   \ "google cloud storage"
16 / Google Drive
   \ "drive"
17 / Google Photos
   \ "google photos"
18 / Hadoop distributed file system
   \ "hdfs"
19 / Hubic
   \ "hubic"
20 / In memory object storage system.
   \ "memory"
21 / Jottacloud
   \ "jottacloud"
22 / Koofr
   \ "koofr"
23 / Local Disk
   \ "local"
24 / Mail.ru Cloud
   \ "mailru"
25 / Mega
   \ "mega"
26 / Microsoft Azure Blob Storage
   \ "azureblob"
27 / Microsoft OneDrive
   \ "onedrive"
28 / OpenDrive
   \ "opendrive"
29 / OpenStack Swift (Rackspace Cloud Files, Memset Memstore,
OVH)
   \ "swift"
30 / Pcloud
   \ "pcloud"
31 / Put.io
   \ "putio"
32 / QingCloud Object Storage
   \ "qingstor"
33 / SSH/SFTP Connection
   \ "sftp"
34 / Sia Decentralized Cloud
   \ "sia"
35 / Sugarsync
   \ "sugarsync"
36 / Tardigrade Decentralized Cloud Storage
   \ "tardigrade"
```

```
37 / Transparently chunk/split large files
   \ "chunker"
38 / Union merges the contents of several upstream fs
   \ "union"
39 / Uptobox
   \ "uptobox"
40 / Webdav
   \ "webdav"
41 / Yandex Disk
   \ "yandex"
42 / Zoho
   \ "zoho"
43 / http Connection
   \ "http"
44 / premiumize.me
   \ "premiumizeme"
45 / seafile
   \ "seafile"
```

```
Storage> 4
```

```
Option provider.
Choose your S3 provider.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
 1 / Amazon Web Services (AWS) S3
   \ "AWS"
 2 / Alibaba Cloud Object Storage System (OSS) formerly Aliyun
   \ "Alibaba"
 3 / Ceph Object Storage
   \ "Ceph"
 4 / Digital Ocean Spaces
   \ "DigitalOcean"
 5 / Dreamhost DreamObjects
   \ "Dreamhost"
 6 / IBM COS S3
   \ "IBMCOS"
 7 / Minio Object Storage
   \ "Minio"
 8 / Netease Object Storage (NOS)
   \ "Netease"
 9 / Scaleway Object Storage
   \ "Scaleway"
10 / SeaweedFS S3
   \ "SeaweedFS"
11 / StackPath Object Storage
   \ "StackPath"
12 / Tencent Cloud Object Storage (COS)
   \ "TencentCOS"
13 / Wasabi Object Storage
   \ "Wasabi"
14 / Any other S3 compatible provider
   \ "Other"
provider> 14
```

```
Option env_auth.
Get AWS credentials from runtime (environment variables or
EC2/ECS meta data if no env vars).
Only applies if access_key_id and secret_access_key is blank.
Enter a boolean value (true or false). Press Enter for the
default ("false").
Choose a number from below, or type in your own value.
 1 / Enter AWS credentials in the next step.
   \ "false"
 2 / Get AWS credentials from the environment (env vars or IAM).
   \ "true"
env_auth> 1
```

```
Option access_key_id.
AWS Access Key ID.
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
access_key_id> ABCDEFGH123456789JKL
```

```
Option secret_access_key.
AWS Secret Access Key (password).
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
secret_access_key> 123456789ABCDEFGHIJKLMN0123456789PQRST+V
```

```
Option region.
Region to connect to.
Leave blank if you are using an S3 clone and you don't have a
region.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
   / Use this if unsure.
 1 | Will use v4 signatures and an empty region.
   \ ""
   / Use this only if v4 signatures don't work.
 2 | E.g. pre Jewel/v10 CEPH.
   \ "other-v2-signature"
region> 1
```

```
Option endpoint.
Endpoint for S3 API.
Required when using an S3 clone.
Enter a string value. Press Enter for the default ("").
endpoint> sgdemo.netapp.com
```

```
Option location_constraint.
Location constraint - must be set to match the Region.
Leave blank if not sure. Used when creating buckets only.
Enter a string value. Press Enter for the default ("").
location_constraint>
```

```
Option acl.
Canned ACL used when creating buckets and storing or copying
objects.
This ACL is used for creating objects and if bucket_acl isn't
set, for creating buckets too.
For more info visit
https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-
overview.html#canned-acl
Note that this ACL is applied when server-side copying objects as
S3
doesn't copy the ACL from the source but rather writes a fresh
one.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value.
   / Owner gets FULL_CONTROL.
 1 | No one else has access rights (default).
   \ "private"
   / Owner gets FULL_CONTROL.
 2 | The AllUsers group gets READ access.
   \ "public-read"
   / Owner gets FULL_CONTROL.
 3 | The AllUsers group gets READ and WRITE access.
   | Granting this on a bucket is generally not recommended.
   \ "public-read-write"
   / Owner gets FULL_CONTROL.
 4 | The AuthenticatedUsers group gets READ access.
   \ "authenticated-read"
   / Object owner gets FULL_CONTROL.
 5 | Bucket owner gets READ access.
   | If you specify this canned ACL when creating a bucket,
Amazon S3 ignores it.
   \ "bucket-owner-read"
   / Both the object owner and the bucket owner get FULL_CONTROL
over the object.
 6 | If you specify this canned ACL when creating a bucket,
Amazon S3 ignores it.
   \ "bucket-owner-full-control"
acl>
```

```
Edit advanced config?
y) Yes
n) No (default)
y/n> n
```

```
--------------------
[sgdemo]
type = s3
provider = Other
access_key_id = ABCDEFGH123456789JKL
secret_access_key = 123456789ABCDEFGHIJKLMN0123456789PQRST+V
endpoint = sgdemo.netapp.com:443
--------------------
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d>
```

```
Current remotes:
```

```
Name                 Type
====                 ====
sgdemo               s3
```

```
e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q
```

기본 명령 예

- * 버킷 생성: *

  `rclone mkdir remote:bucket`

  ```
  rclone mkdir sgdemo:test01
  ```

  SSL 인증서를 무시해야 하는 경우 — 확인 안 함 - 인증서를 사용합니다.

- * 모든 버킷 나열: *

```
rclone lsd remote:
```

rclone LSD sgdemo 수:

- * 특정 버킷의 오브젝트 목록: *

```
rclone ls remote:bucket
```

rclone ls sgdemo: test01

```
     65536 TestObject.0
     65536 TestObject.1
     65536 TestObject.10
     65536 TestObject.12
     65536 TestObject.13
     65536 TestObject.14
     65536 TestObject.15
     65536 TestObject.16
     65536 TestObject.17
     65536 TestObject.18
     65536 TestObject.2
     65536 TestObject.3
     65536 TestObject.5
     65536 TestObject.6
     65536 TestObject.7
     65536 TestObject.8
     65536 TestObject.9
  33554432 bigobj
       102 key.json
        47 locked01.txt
4294967296 sequential-read.0.0
        15 test.txt
       116 version.txt
```

- * 버킷 삭제: *

```
rclone rmdir remote:bucket
```

rclone rmdir sgdemo:test02

- * 개체 넣기: *

```
rclone copy filename remote:bucket
```

rclone copy~/test/testfile.txt sgdemo:test01

- * 개체 가져오기: *

```
rclone copy remote:bucket/objectname filename
```

rclone copy sgdemo:test01/testfile.txt~/test/testfileS3.txt

- * 개체 삭제: *

```
rclone delete remote:bucket/objectname
```

rclone delete sgdemo:test01/testfile.txt

- * 버킷에서 오브젝트 마이그레이션 *

```
rclone sync source:bucket destination:bucket --progress

rclone sync source_directory destination:bucket --progress
```

rclone sync sgdemo: test01 sgdemo: clone01 — 진행률

```
Transferred:        4.032 GiB / 4.032 GiB, 100%, 95.484 KiB/s, ETA
0s
Transferred:           22 / 22, 100%
Elapsed time:       1m4.2s
```

progress(진행) 또는 -P를 사용하여 작업의 진행 상황을 표시합니다. 그렇지 않으면 출력이 없습니다.

- * 버켓과 모든 오브젝트 내용 삭제 *

```
rclone purge remote:bucket --progress
```

rclone purge sgdemo: test01 — 진행률

```
Transferred:              0 B / 0 B, -, 0 B/s, ETA -
Checks:                46 / 46, 100%
Deleted:               23 (files), 1 (dirs)
Elapsed time:        10.2s
```

rclone ls sgdemo: test01

```
2023/04/14 09:40:51 Failed to ls: directory not found
```

지그프리드 헤프와 아론 클라인 작사

Veeam 백업 및 복제를 사용한 구축에 대한 StorageGRID 모범 사례

이 가이드에서는 NetApp StorageGRID 구성과 일부 Veeam 백업 및 복제를 중점적으로 다룹니다. 이 문서는 Linux 시스템에 익숙하고 Veeam 백업 및 복제와 함께 NetApp StorageGRID 시스템의 유지 관리 또는 구축을 담당하는 스토리지 및 네트워크 관리자를 위해 작성되었습니다.

개요

스토리지 관리자는 가용성, 빠른 복구 목표, 요구 사항에 맞게 확장 및 장기 데이터 보존을 위한 정책을 자동화하는 솔루션을 사용하여 증가하는 데이터를 관리할 수 있습니다. 이러한 솔루션은 손실 또는 악의적인 공격으로부터 보호되어야 합니다. Veeam과 NetApp은 파트너십을 통해 Veeam 백업 및 복구를 사내 오브젝트 스토리지에 NetApp StorageGRID과 결합하는 데이터 보호 솔루션을 만들었습니다.

Veeam과 NetApp StorageGRID는 전 세계적으로 빠르게 증가하는 데이터 및 늘어나는 규정 요구 사항을 충족하는 사용하기 쉬운 솔루션을 제공합니다. 클라우드 기반 오브젝트 스토리지는 복원력, 확장 기능, 운영 및 비용 효율성으로 인해 백업 대상으로 자연스럽게 선택할 수 있는 것으로 유명합니다. 이 문서는 Veeam 백업 솔루션 및 StorageGRID 시스템 구성에 대한 지침과 권장사항을 제공합니다.

Veeam의 오브젝트 워크로드에는 작은 오브젝트의 여러 동시 배치, 삭제 및 목록 작업이 생성됩니다. 불변성을 설정하면 보존 및 목록 버전을 설정하기 위한 요청 수가 개체 저장소에 추가됩니다. 백업 작업의 프로세스에는 일일 변경 사항에 대한 객체 쓰기가 포함되며 새 쓰기가 완료된 후 작업은 백업의 보존 정책에 따라 모든 객체를 삭제합니다. 백업 작업의 스케줄링은 거의 항상 중복됩니다. 이렇게 겹치면 객체 저장소의 50/50 PUT/DELETE 워크로드로 구성된 백업 윈도우의 상당 부분이 발생합니다. Veeam에서 작업 슬롯 설정을 사용하여 동시 작업 수를 조정하면 백업 작업 블록 크기를 늘리고 다중 개체 삭제 요청의 객체 수를 줄여 객체 크기를 늘릴 수 있습니다. 또한 작업을 완료할 최대 기간을 선택하면 성능 및 비용에 맞게 솔루션을 최적화할 수 있습니다.

에 대한 제품 설명서를 읽어야 합니다 "Veeam 백업 및 복제" 및 "StorageGRID" 시작하기 전에. Veeam을 사용하면 StorageGRID 솔루션을 사이징하기 전에 사용해야 할 Veeam 인프라 및 용량 요구사항의 크기를 이해할 수 있습니다. 의 Veeam Ready 프로그램 웹 사이트에서 Veeam-NetApp의 검증된 구성을 항상 확인하십시오 "Veeam Ready Object, Object Immutability 및 Repository를 사용할 수 있습니다".

Veeam 구성

권장 버전

Veeam Backup & Replication 12 시스템에 최신 핫픽스를 적용하는 것이 좋습니다. 현재는 최소한 Veeam 패치 P20230718을 설치할 것을 권장합니다.

S3 저장소 구성

스케일아웃 백업 저장소(SOBR)는 S3 오브젝트 스토리지의 용량 계층입니다. 용량 계층은 기본 저장소의 확장 기능으로, 데이터 보존 기간이 길고 스토리지 솔루션이 저렴합니다. Veeam은 S3 Object Lock API를 통해 불변성을 제공하는 기능을 제공합니다. Veeam 12는 스케일아웃 저장소에서 여러 버킷을 사용할 수 있습니다. StorageGRID은 단일 버킷의 오브젝트 또는 용량에 대한 제한이 없습니다. 여러 버킷을 사용하면 백업 데이터가 오브젝트에서 페타바이트 규모로 증가할 수 있는 대규모 데이터 세트를 백업할 때 성능이 향상될 수 있습니다.

특정 솔루션 및 요구 사항의 사이징에 따라 동시 작업을 제한해야 할 수 있습니다. 기본 설정에서는 각 CPU 코어와 각 작업 슬롯에 대해 하나의 리포지토리 작업 슬롯을 지정하고 동시 작업 슬롯 제한은 64입니다. 예를 들어 서버에 2개의 CPU 코어가 있는 경우 총 128개의 동시 스레드가 개체 저장소에 사용됩니다. 여기에는 PUT, GET, BATCH Delete가 포함됩니다. Veeam 백업이 새로운 백업 및 백업 데이터의 안정적 상태에 도달하고 만료 예정인 경우 시작할 작업 슬롯에 대해 보수적인 제한을 선택하고 이 값을 조정하는 것이 좋습니다. NetApp 어카운트 팀과 협력하여 원하는 시간 및 성능을 만족하도록 StorageGRID 시스템의 크기를 적절하게 조정해 주십시오. 최적의 솔루션을 제공하기 위해 슬롯당 작업 슬롯의 수와 작업 제한을 조정해야 할 수 있습니다.

백업 작업 구성입니다

Veeam 백업 작업은 신중하게 고려해야 하는 다양한 블록 크기 옵션으로 구성할 수 있습니다. 기본 블록 크기는 1MB이며, 압축 및 중복제거를 통해 Veeam이 제공하는 스토리지 효율성을 통해 초기 전체 백업에는 약 500KB의 오브젝트 크기와 증분 작업에 대해서는 100~200kB 오브젝트를 생성합니다. NetApp은 더 큰 백업 블록 크기를 선택하여 성능을 크게 향상시키고 오브젝트 저장소 요구 사항을 축소할 수 있습니다. 블록 크기가 클수록 오브젝트 저장소의 성능이 크게 향상되지만, 스토리지 효율성 성능의 저하로 인해 기본 스토리지 용량 요구사항이 증가할 가능성이 있습니다. 전체 백업에 대해 약 2MB의 객체를 생성하는 4MB 블록 크기로 백업 작업을 구성하고 증가분에 대해 700kB-1MB 객체 크기를 생성하는 것이 좋습니다. 고객은 Veeam 지원의 도움을 받아 8MB 블록 크기를 사용하여 백업 작업을 구성하는 것도 고려할 수 있습니다.

변경 불가능한 백업을 구현하면 오브젝트 저장소에서 S3 오브젝트 잠금을 사용합니다. 불변성 옵션은 객체에 대한 목록 및 보존 업데이트를 위해 객체 저장소에 대한 요청을 더 많이 생성합니다.

백업 보존 기간이 만료되면 백업 작업이 객체 삭제를 처리합니다. Veeam은 요청당 1,000개의 오브젝트가 포함된 다중 오브젝트 삭제 요청의 삭제 요청을 오브젝트 저장소로 전송합니다. 소규모 솔루션의 경우 요청당 객체 수를 줄이기 위해 조정해야 할 수 있습니다. 이 값을 낮추면 삭제 요청을 StorageGRID 시스템의 노드에 고르게 분산시킬 수 있는 이점이 추가됩니다. 다중 개체 삭제 제한을 구성할 때는 아래 표의 값을 시작점으로 사용하는 것이 좋습니다. 표의 값에 선택한 어플라이언스 유형의 노드 수를 곱하여 Veeam의 설정 값을 구합니다. 이 값이 1000보다 크거나 같으면 기본값을 조정할 필요가 없습니다. 이 값을 조정해야 하는 경우, Veeam 지원에 문의하여 변경하십시오.

어플라이언스 모델	노드별 **S3MultiObjectDeleteLimit**
SG5712를 참조하십시오	34
SG5760입니다	75를
SG6060입니다	200

특정 요구 사항에 맞는 권장 구성을 위해 NetApp 세일즈 팀과 협력하십시오. Veeam 구성 권장 사항에는 다음이 포함됩니다.

- 백업 작업 블록 크기 = 4MB
- SOBR 작업 슬롯 제한 = 2-16
- 다중 개체 삭제 제한 = 34-1000

StorageGRID 구성

권장 버전

최신 핫픽스가 포함된 NetApp StorageGRID 11.6 또는 11.7은 Veeam 구축에 권장되는 버전입니다. StorageGRID 11.6.0.11 및 11.7.0.4에 많은 최적화 기능이 도입되어 Veeam 워크로드에 유용합니다. 항상 최신 상태를 유지하고 StorageGRID 시스템에 최신 핫픽스를 적용하는 것이 좋습니다.

로드 밸런서 및 **S3** 엔드포인트 구성

Veeam을 사용하면 HTTPS를 통해서만 엔드포인트를 연결해야 합니다. 암호화되지 않은 연결은 Veeam에서 지원되지 않습니다. SSL 인증서는 자체 서명된 인증서, 신뢰할 수 있는 개인 인증 기관 또는 신뢰할 수 있는 공용 인증 기관일 수 있습니다. S3 저장소에 대한 지속적인 액세스를 보장하려면 HA 구성에서 로드 밸런서를 2개 이상 사용하는 것이 좋습니다. 로드 밸런서는 모든 관리 노드 및 게이트웨이 노드에 있는 StorageGRID에서 제공하는 통합 로드 밸런서 서비스이거나 F5, Kemp, Hafroxy, Loadbalanacer.org 등과 같은 타사 솔루션일 수 있습니다 StorageGRID 로드 밸런서를 사용하면 Veeam 워크로드의 우선순위를 지정할 수 있는 트래픽 분류자(QoS 규칙)를 설정하거나, Veeam을 StorageGRID 시스템에서 우선순위가 높은 워크로드에 영향을 미치지 않도록 제한할 수 있습니다.

S3 버킷

StorageGRID는 안전한 멀티 테넌트 스토리지 시스템입니다. Veeam 워크로드를 위한 전용 테넌트를 생성하는 것이 좋습니다. 필요에 따라 스토리지 할당량을 할당할 수 있습니다. 최선의 방법으로 "자체 ID 소스 사용"을 활성화합니다. 적절한 암호를 사용하여 테넌트 루트 관리 사용자를 보호합니다. Veeam Backup 12는 S3 버킷의 경우 강력한 일관성이 필요합니다. StorageGRID은 버킷 수준에서 구성된 다양한 정합성 보장 옵션을 제공합니다. Veeam이 여러 위치의 데이터에 액세스할 수 있는 멀티 사이트 배포의 경우 "강력한 글로벌"을 선택하십시오. Veeam 백업 및 복원을 단일 사이트에서만 수행할 경우 일관성 수준을 "강력한 사이트"로 설정해야 합니다. 버킷 일관성 수준에 대한 자세한 내용은 을 참조하십시오 "문서화". Veeam 불변성 백업을 위해 StorageGRID를 사용하려면 S3 오브젝트 잠금을 글로벌로 사용하도록 설정하고 버킷 생성 중에 버킷에 구성해야 합니다.

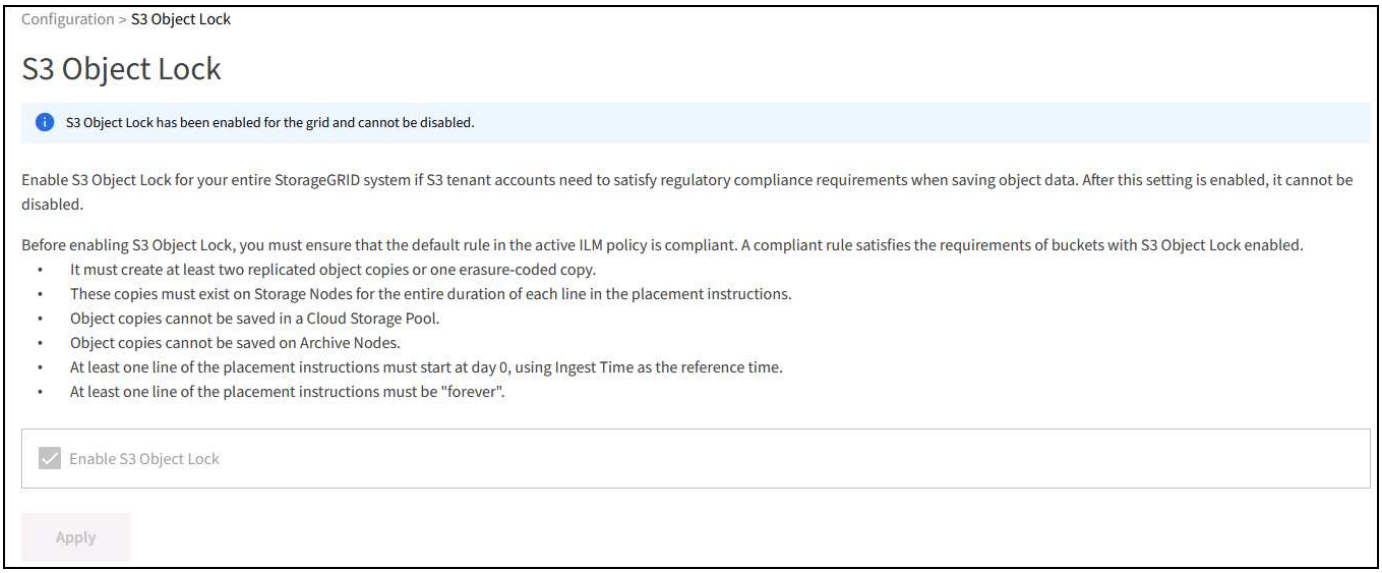
라이프사이클 관리

StorageGRID는 StorageGRID 노드와 사이트에서 오브젝트 레벨의 보호를 위해 복제 및 삭제 코딩을 지원합니다. 삭제 코딩에는 최소 200kB 오브젝트 크기가 필요합니다. Veeam의 1MB에 대한 기본 블록 크기는 Veeam의 스토리지 효율성 후 종종 이 200kB 권장 최소 크기보다 작을 수 있는 오브젝트 크기를 생성합니다. 솔루션의 성능을 위해 사이트 간 연결이 지연 시간을 추가하거나 StorageGRID 시스템의 대역폭을 제한하지 않는 한 여러 사이트에 걸쳐 있는 삭제 코딩 프로필을 사용하지 않는 것이 좋습니다. 다중 사이트 StorageGRID 시스템에서는 각 사이트에 단일 복제본을 저장하도록 ILM 규칙을 구성할 수 있습니다. 내구성을 최대화하기 위해 각 사이트에 삭제 코딩 복사본을 저장하도록 규칙을 구성할 수 있습니다. 이 워크로드를 위해 Veeam Backup 서버에 로컬에 2개의 복제본을 사용하는 것이 가장 좋습니다.

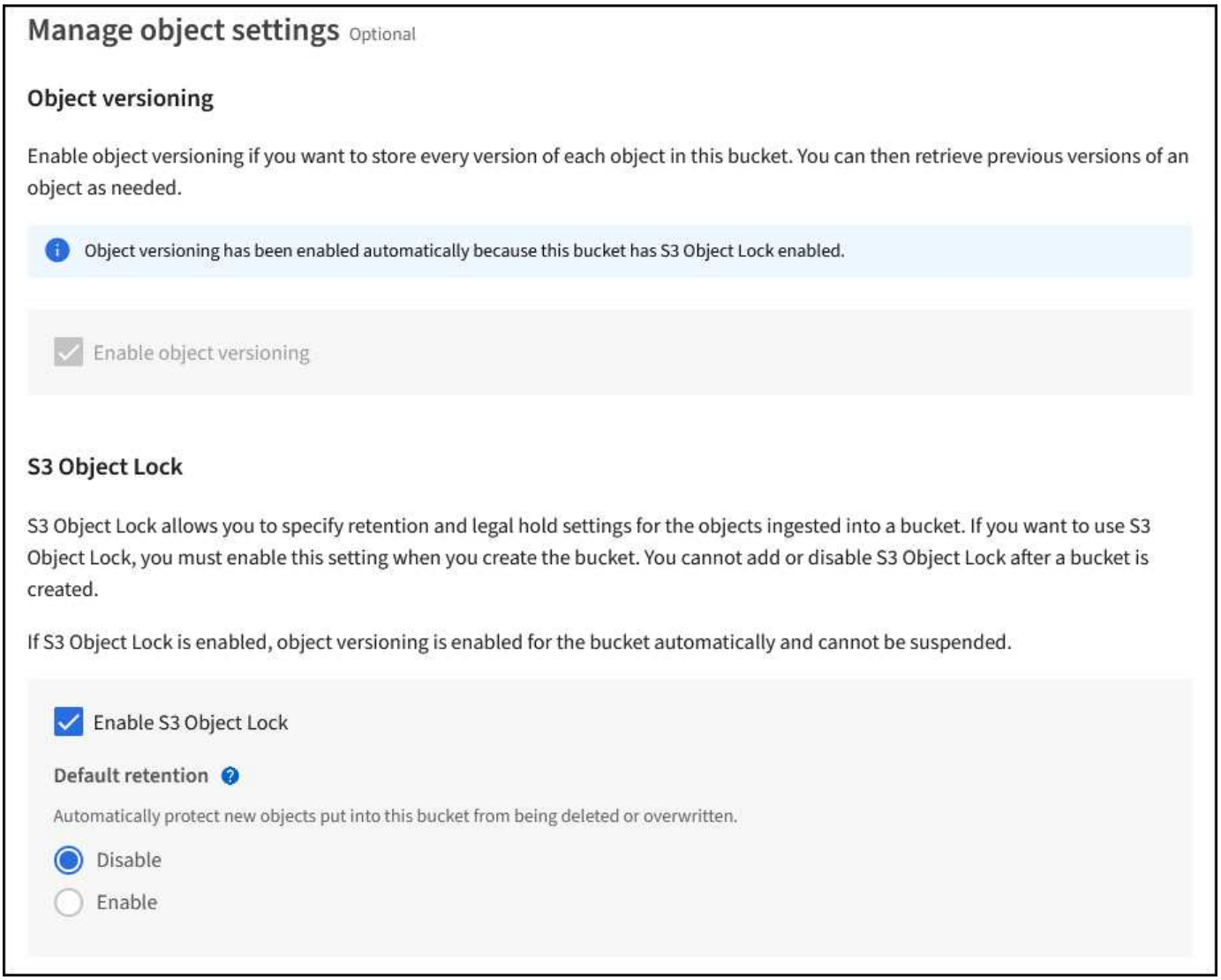
구현 핵심 사항

StorageGRID

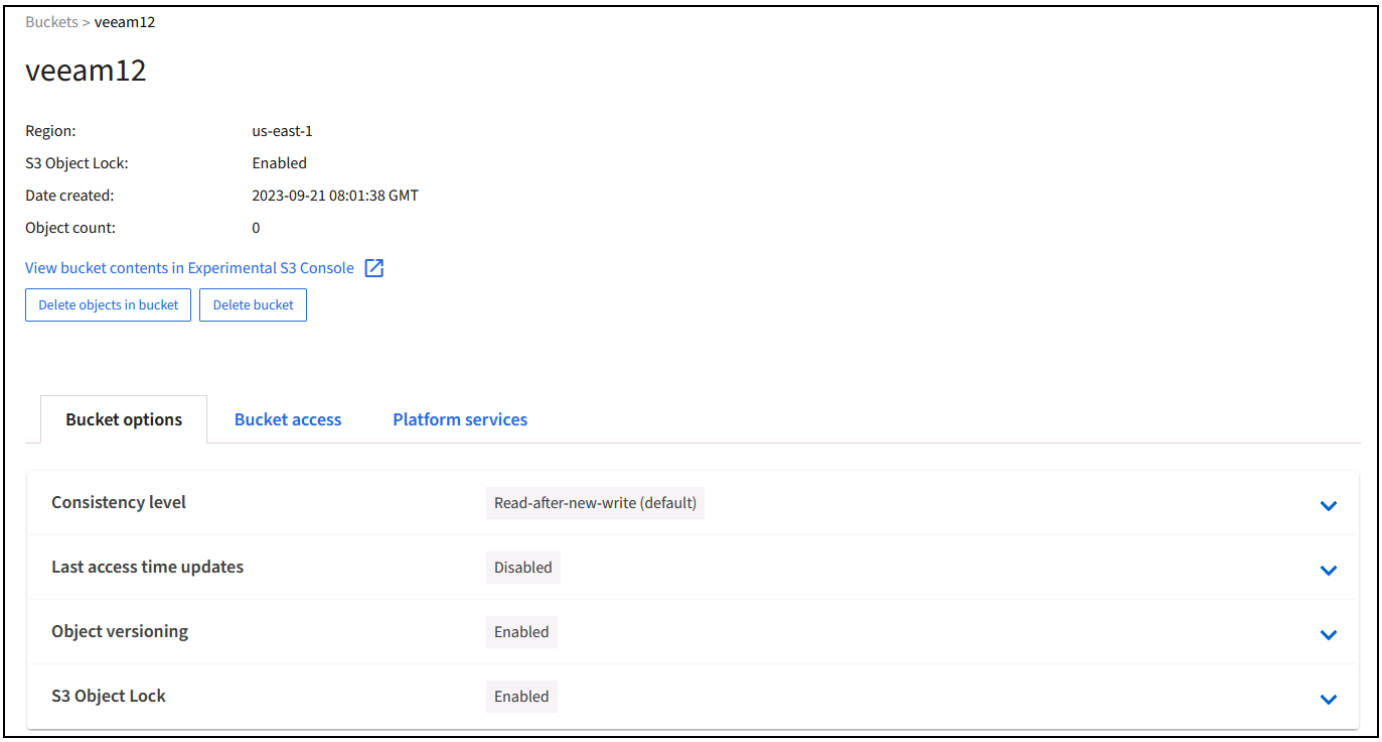
불변성이 필요한 경우 StorageGRID 시스템에서 오브젝트 잠금이 활성화되어 있는지 확인합니다. 관리 UI의 구성/S3 오브젝트 잠금 아래에서 옵션을 찾습니다.



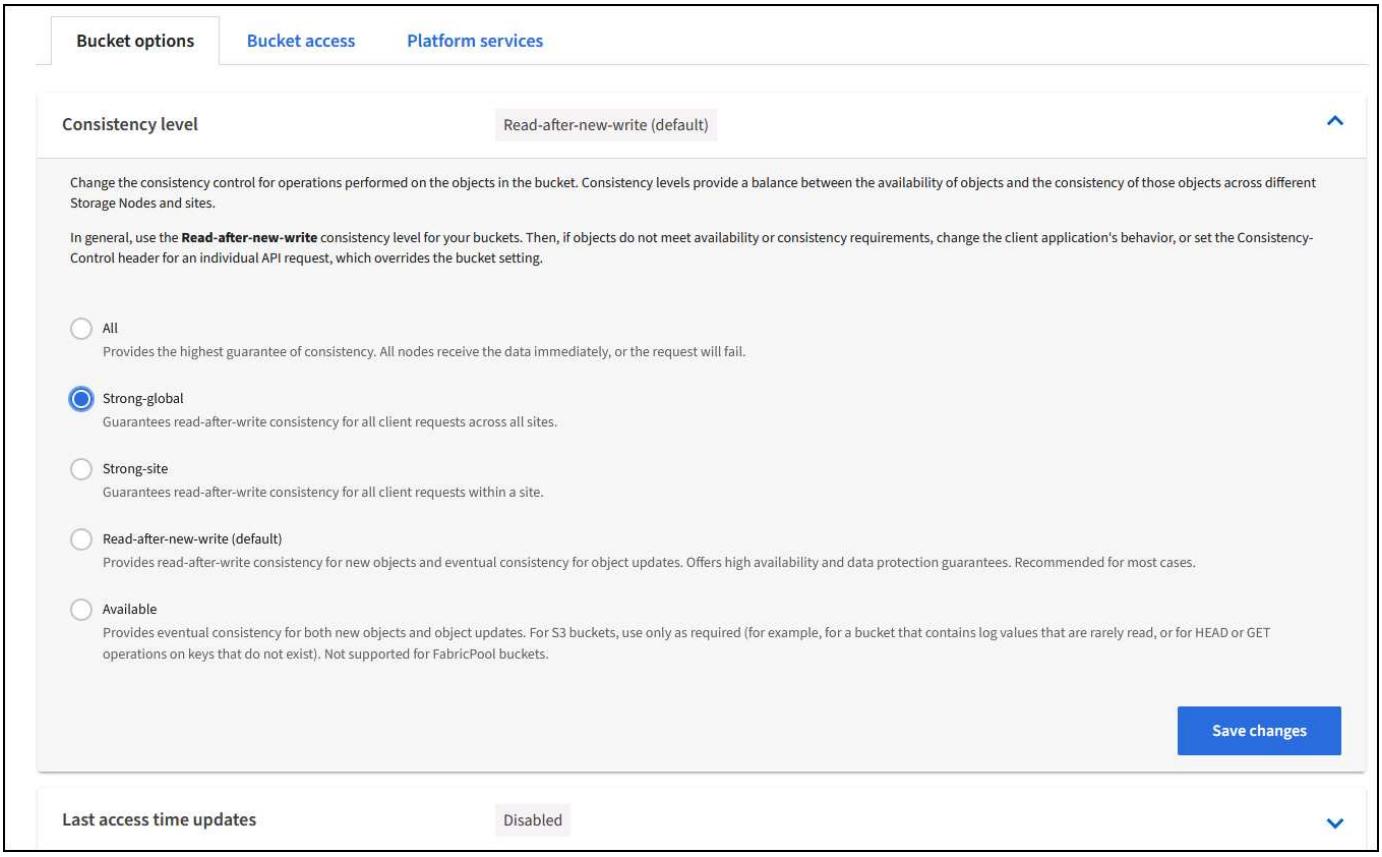
버킷을 생성할 때 불변의 백업에 이 버킷을 사용하려면 "S3 오브젝트 잠금 활성화"를 선택하십시오. 이렇게 하면 버킷 버전 관리가 자동으로 활성화됩니다. Veeam에서 객체 보존을 명시적으로 설정하므로 기본 보존을 사용하지 않도록 설정합니다. Veeam에서 변경 불가능한 백업을 생성하지 않는 경우 버전 관리 및 S3 오브젝트 잠금을 선택하지 않아야 합니다.



버킷이 생성되면 생성된 버킷의 세부 정보 페이지로 이동합니다. 정합성 보장 수준을 선택합니다.



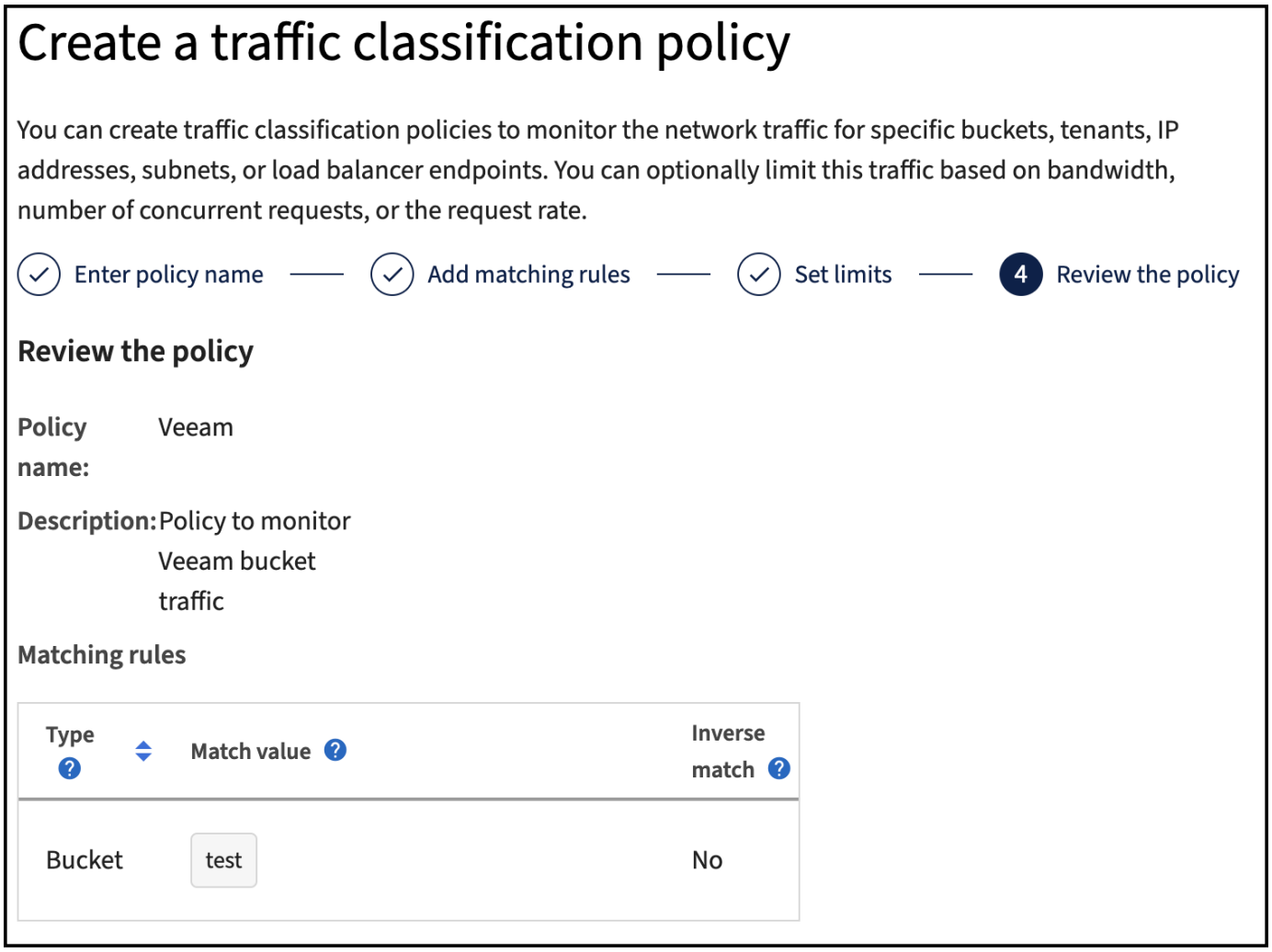
Veeam을 사용하려면 S3 버킷에 대해 강력한 일관성이 필요합니다. 따라서 Veeam을 통해 여러 위치의 데이터에 액세스할 수 있는 멀티 사이트 배포의 경우 "강력한 글로벌"을 선택하십시오. Veeam 백업 및 복원을 단일 사이트에서만 수행할 경우 일관성 수준을 "강력한 사이트"로 설정해야 합니다. 변경 사항을 저장합니다.



StorageGRID는 모든 관리 노드와 전용 게이트웨이 노드에서 통합 로드 밸런서 서비스를 제공합니다. 이 로드 밸런서를 사용하면 QoS(트래픽 분류 정책)를 구성할 수 있다는 이점이 많습니다. 이러한 기능은 다른 클라이언트 워크로드에

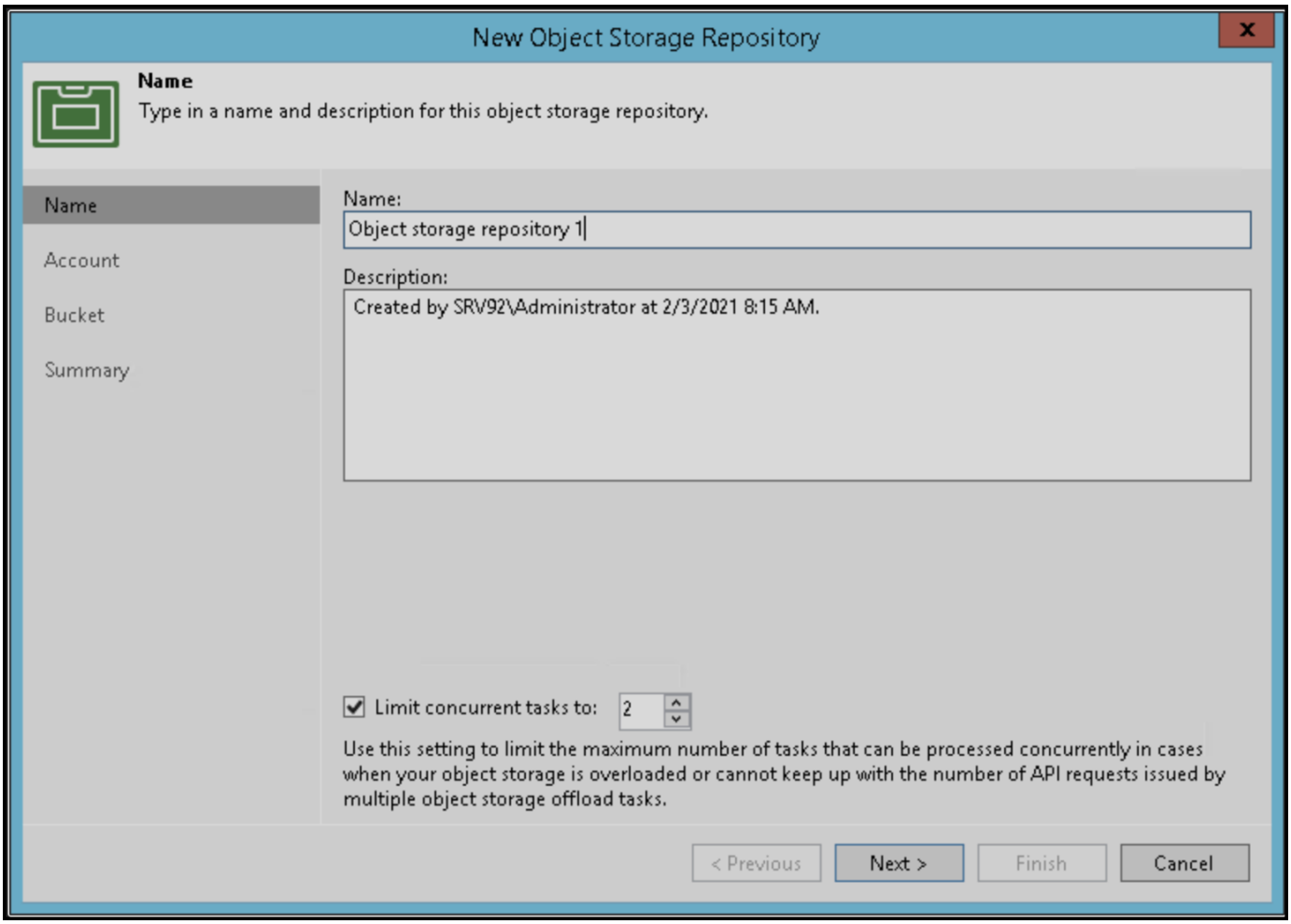
대한 애플리케이션 영향을 제한하거나 다른 워크로드에 대한 우선 순위를 지정하는 데 주로 사용되지만 모니터링에 도움이 되는 추가 메트릭 수집도 제공합니다.

구성 탭에서 "트래픽 분류"를 선택하고 새 정책을 생성합니다. 규칙의 이름을 지정하고 유형으로 버킷 또는 테넌트를 선택합니다. 버킷 또는 테넌트의 이름을 입력하십시오. QoS가 필요한 경우 제한을 설정하지만 대부분의 구현에서는 모니터링 이점을 추가하려고 하므로 제한을 설정하지 마십시오.

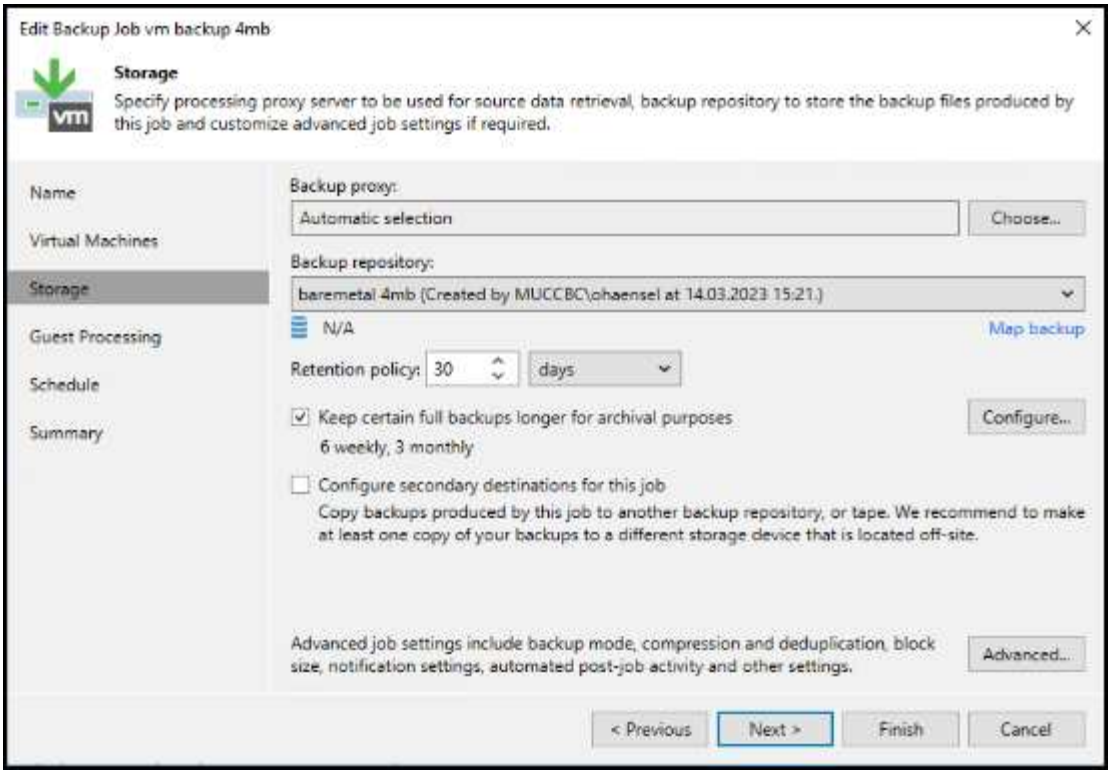


Veeam을 선택합니다

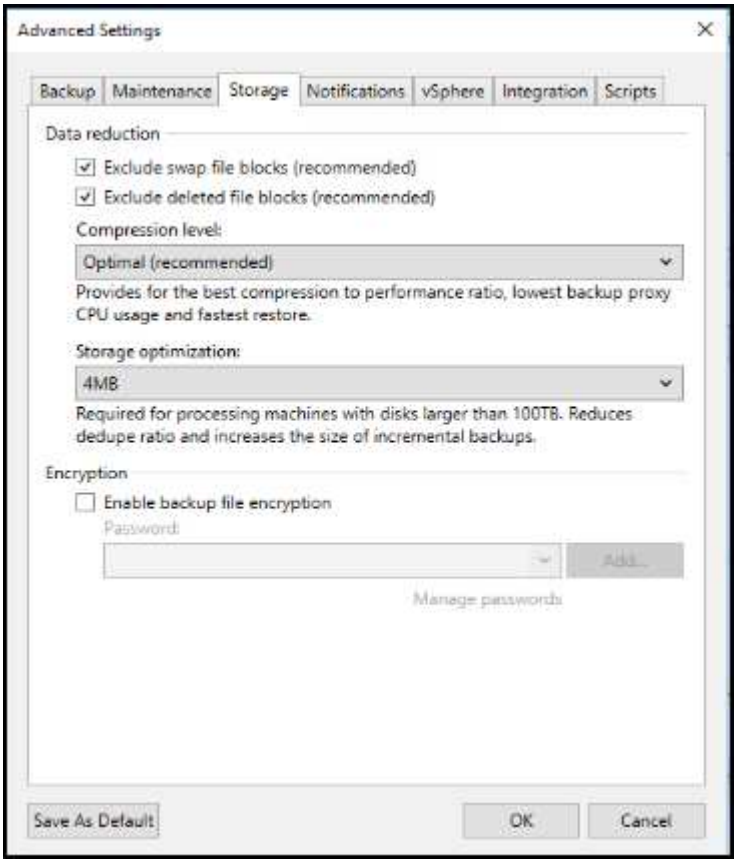
StorageGRID 어플라이언스의 모델 및 수량에 따라 버킷에서 동시 작업 수의 제한을 선택하고 구성해야 할 수 있습니다.



Veeam 콘솔의 백업 작업 구성에 관한 Veeam 설명서를 따라 마법사를 시작합니다. VM을 추가한 후 SOBR 리포지토리를 선택합니다.



고급 설정 을 클릭하고 저장소 최적화 설정을 4MB 이상으로 변경합니다. 압축 및 중복제거가 활성화되어야 합니다. 요구사항에 따라 게스트 설정을 변경하고 백업 작업 일정을 구성합니다.



StorageGRID 모니터링

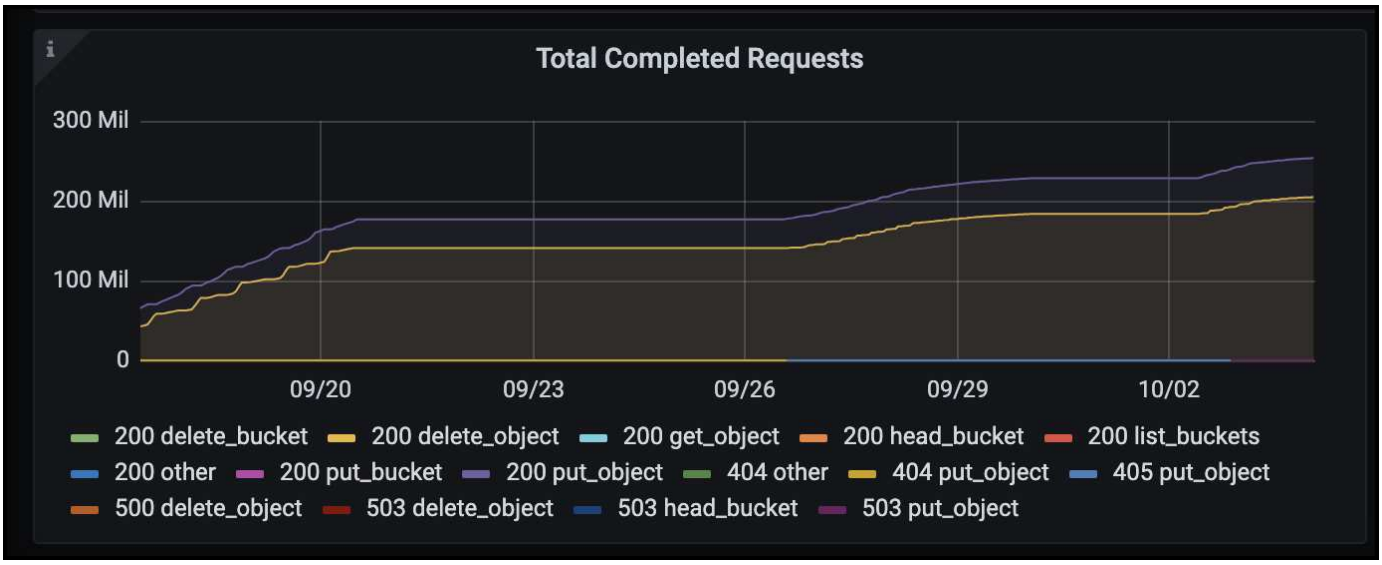
Veeam과 StorageGRID가 함께 작동하는 방식을 자세히 보려면 첫 번째 백업의 보존 시간이 만료될 때까지 기다려야 합니다. 지금까지는 Veeam 워크로드가 주로 PUT 작업으로 구성되며 삭제가 발생하지 않습니다. 백업 데이터가 만료되고 정리가 시작되면 오브젝트 저장소에서 전체 일관된 사용량을 확인하고 필요한 경우 Veeam에서 설정을 조정할 수 있습니다.

StorageGRID는 지원 탭 메트릭 페이지에 있는 시스템 작동을 모니터링하는 편리한 차트를 제공합니다. 주요 대시보드는 정책을 생성한 경우 S3 개요, ILM 및 트래픽 분류 정책입니다. S3 개요 대시보드에서 S3 작업 속도, 지연 시간 및 요청 응답에 대한 정보를 확인할 수 있습니다.

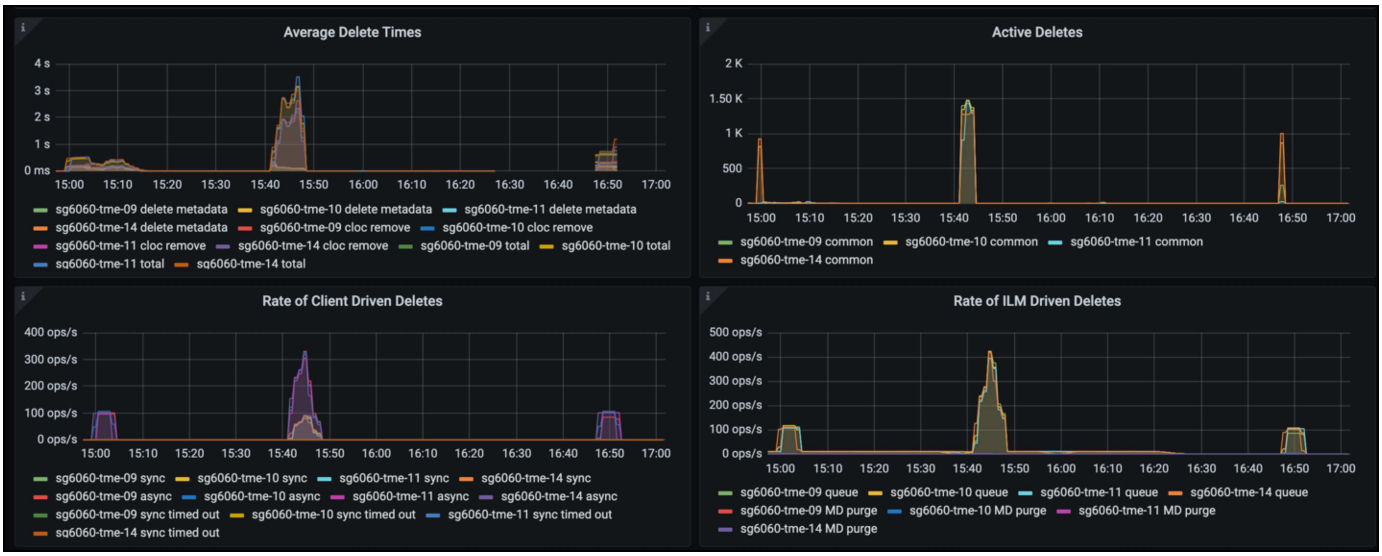
S3 속도 및 활성 요청을 보면 각 노드가 처리 중인 로드의 양과 유형별로 전체 요청 수를 확인할 수 있습니다.

Average Duration(평균 기간) 차트에는 각 노드가 각 요청 유형에 대해 걸리는 평균 시간이 표시됩니다. 이는 요청의 평균 대기 시간이며 추가 튜닝이 필요하거나 StorageGRID 시스템이 더 많은 로드를 처리할 수 있는 공간이 있음을 나타내는 좋은 지표가 될 수 있습니다.

총 완료된 요청 차트에서 유형 및 응답 코드별로 요청을 볼 수 있습니다. 응답에 대해 200(OK)이 아닌 응답이 표시되면 StorageGRID 시스템이 503(느린 속도) 응답을 보내면서 로드가 과중하게 로드되고 있는 것과 같은 문제일 수 있으며 추가적인 튜닝이 필요하거나 로드가 증가하기 위해 시스템을 확장할 시간이 되었을 수 있습니다.



ILM 대시보드에서 StorageGRID 시스템의 삭제 성능을 모니터링할 수 있습니다. StorageGRID는 각 노드에서 동기 및 비동기 삭제를 결합하여 모든 요청의 전반적인 성능을 최적화하고 시도합니다.



트래픽 분류 정책을 사용하면 로드 밸런서에 대한 메트릭을 볼 수 있습니다. 요청 처리량, 속도, 기간, Veeam이 전송 및 수신하는 객체 크기 등을 확인할 수 있습니다.

추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대해 자세히 알아보려면 다음 문서 및/또는 웹 사이트를 검토하십시오.

- "NetApp StorageGRID 11.7 제품 문서"
- "Veeam 백업 및 복제"

올리버 헨셀과 아론 클라인 작사

StorageGRID를 사용하여 Dremio 데이터 소스를 구성합니다

Dremio는 클라우드 기반 또는 온프레미스 오브젝트 스토리지를 비롯한 다양한 데이터 소스를 지원합니다. StorageGRID를 오브젝트 스토리지 데이터 소스로 사용하도록 Dremio를 구성할 수 있습니다.

Dremio 데이터 소스를 구성합니다

필수 구성 요소

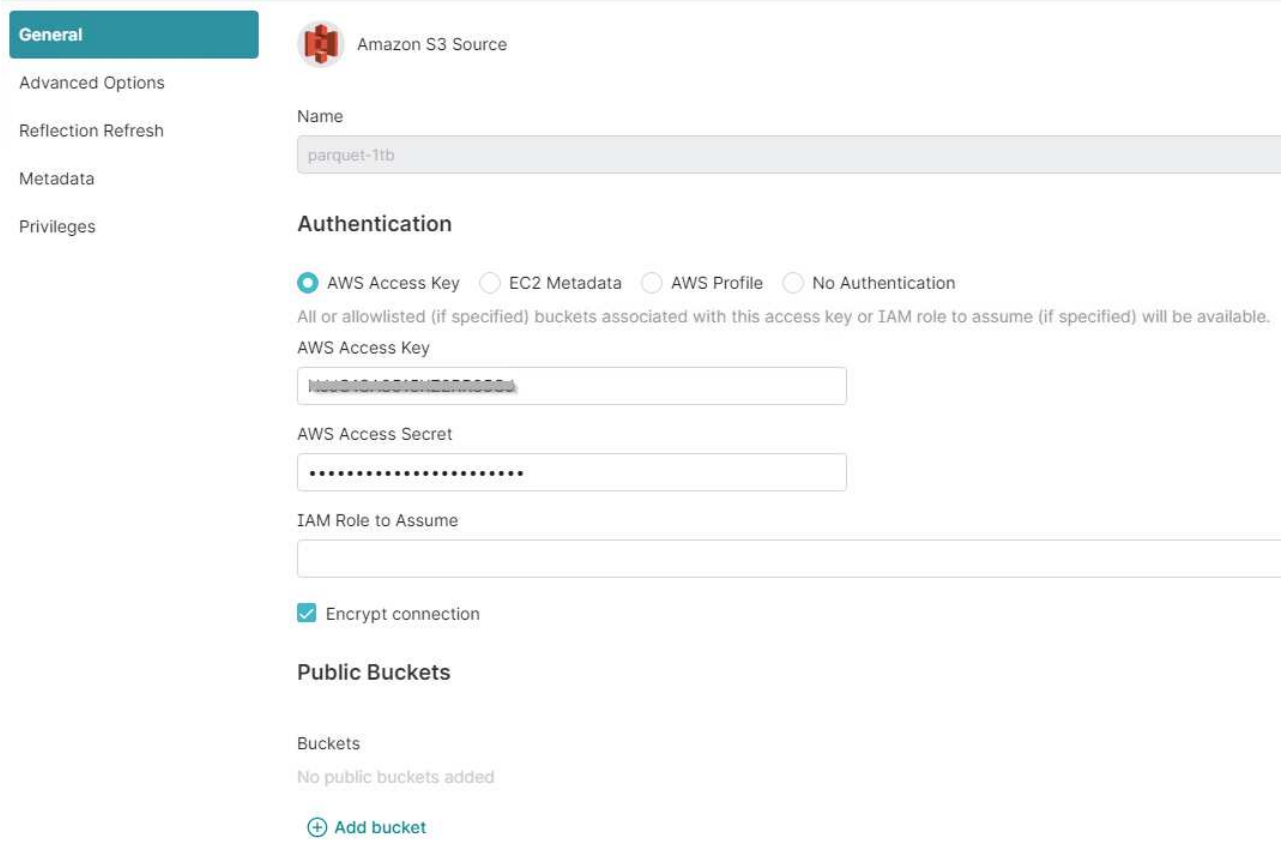
- StorageGRID S3 끝점 URL, 테넌트 S3 액세스 키 ID 및 보안 액세스 키
- StorageGRID 구성 권장 사항: 압축을 사용하지 않도록 설정(기본적으로 해제됨) 를 누릅니다 Dremio는 쿼리 중에 동일한 개체 내에서 다른 바이트 범위를 동시에 가져오기 위해 바이트 범위 GET를 사용합니다. 일반적으로 바이트 범위 요청의 크기는 1MB입니다. 압축된 객체는 바이트 범위 가져오기 성능을 저하시킵니다.

Dremio 가이드

"Amazon S3에 연결 - S3 호환 스토리지 구성".

지침

1. Dremio Datasets 페이지에서 + 기호를 클릭하여 소스를 추가하고 'Amazon S3'를 선택합니다.
2. 이 새 데이터 소스의 이름, StorageGRID S3 테넌트 액세스 키 ID 및 비밀 액세스 키를 입력합니다.
3. StorageGRID S3 끝점에 연결하기 위해 https를 사용하는 경우 '연결 암호화' 확인란을 선택합니다. 를 누릅니다 이 S3 끝점에 대해 자체 서명된 CA 인증서를 사용하는 경우 Dremio 가이드 지침에 따라 이 CA 인증서를 Dremio 서버의 <JAVA_HOME>/JRE/lib/security+에 추가합니다
    - 샘플 스크린샷 *

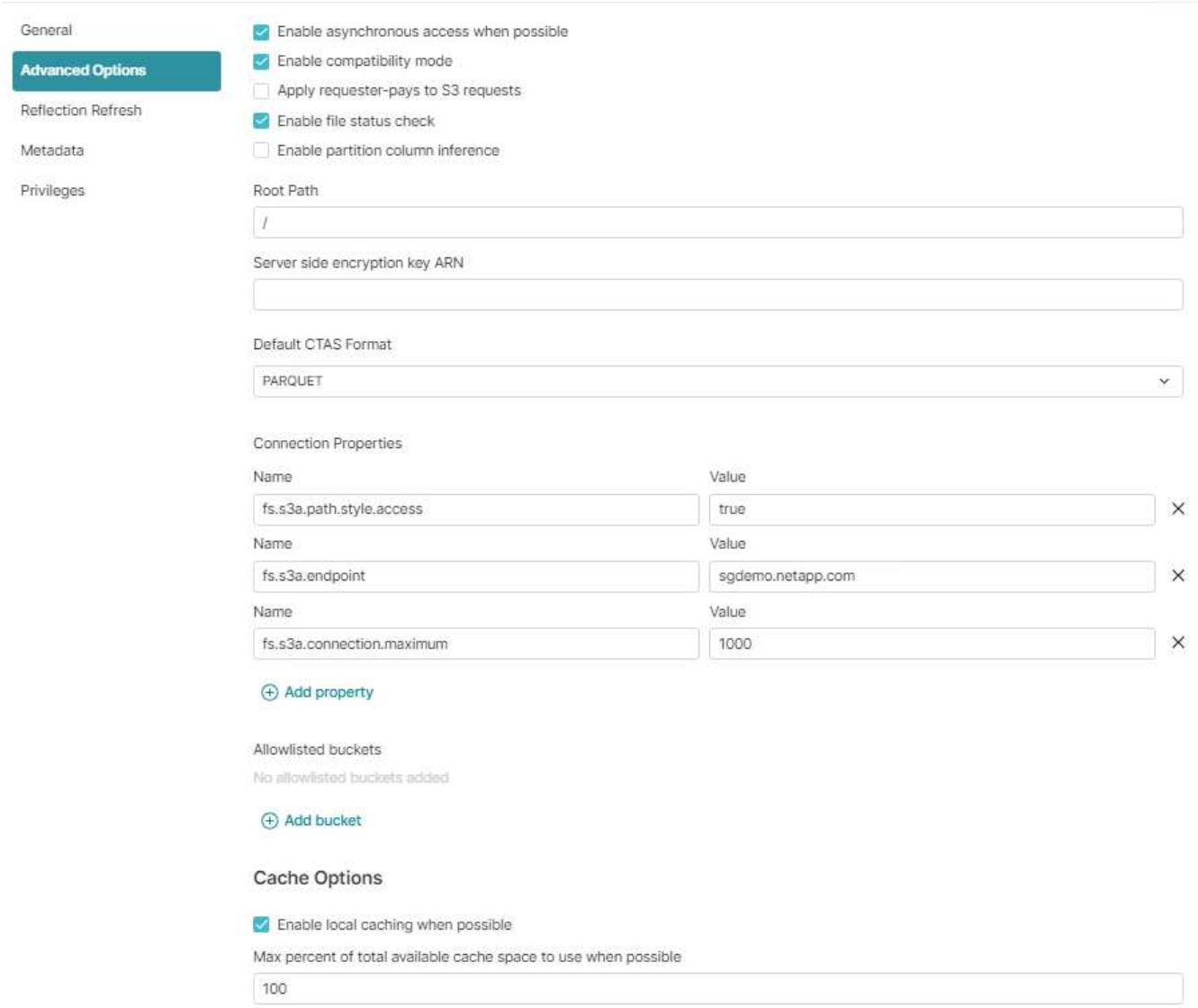


4. '고급 옵션'을 클릭하고 '호환 모드 사용'을 선택합니다.
5. 연결 속성에서 + 속성 추가를 클릭하고 이러한 s3a 속성을 추가합니다.

6. FS.s3a.connection. 최대 기본값은 100입니다. S3 데이터 세트에 100개 이상의 열이 있는 대형 Parquet 파일이 포함된 경우 에서 100보다 큰 값을 입력해야 합니다. 이 설정은 Dremio 가이드를 참조하십시오.

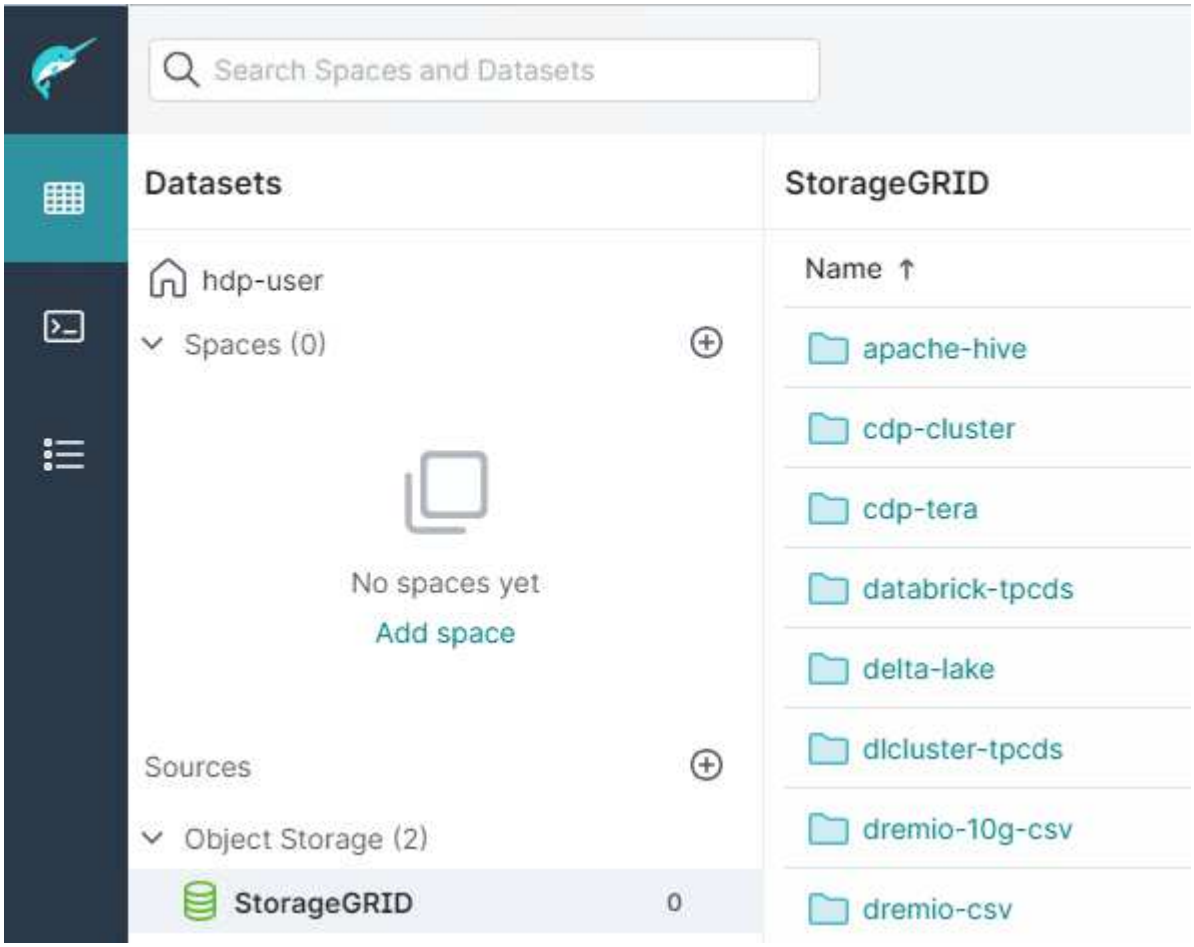
이름	값
FS.s3a.endpoint	_<StorageGRID S3 엔드포인트: port> _
FS.s3a.path.style.access	참
FS.s3a.connection.maximum입니다	_< 100보다 큰 값 > _

◦ 샘플 스크린샷 *



7. 조직 또는 응용 프로그램 요구 사항에 따라 다른 Dremio 옵션을 구성합니다.
8. 이 새 데이터 원본을 만들려면 저장 단추를 클릭합니다.
9. StorageGRID 데이터 소스가 성공적으로 추가되면 버킷 목록이 왼쪽 패널에 표시됩니다. 를 누릅니다

◦ 샘플 스크린샷 *



안젤라 청 _ 에 의해

GitLab을 사용한 NetApp StorageGRID

NetApp은 GitLab에서 StorageGRID를 테스트했습니다. 아래의 GitLab 구성 샘플을 참조하십시오. 을 참조하십시오 "GitLab 객체 스토리지 구성 가이드" 를 참조하십시오.

객체 저장소 연결 예

Linux 패키지 설치의 경우, 이 예제는 의 예입니다 `connection` 통합 양식의 설정입니다. 편집 `/etc/gitlab/gitlab.rb` 원하는 값으로 다음 줄을 추가합니다.

```
# Consolidated object storage configuration
gitlab_rails['object_store']['enabled'] = true
gitlab_rails['object_store']['proxy_download'] = true
gitlab_rails['object_store']['connection'] = {
  'provider' => 'AWS',
  'region' => 'us-east-1',
  'endpoint' => 'https://<storagegrid-s3-endpoint:port>',
  'path_stype' => 'true',
  'aws_access_key_id' => '<AWS_ACCESS_KEY_ID>',
  'aws_secret_access_key' => '<AWS_SECRET_ACCESS_KEY>'
}
# OPTIONAL: The following lines are only needed if server side encryption
is required
gitlab_rails['object_store']['storage_options'] = {
  'server_side_encryption' => 'AES256'
}
gitlab_rails['object_store']['objects']['artifacts']['bucket'] = 'gitlab-
artifacts'
gitlab_rails['object_store']['objects']['external_diffs']['bucket'] =
'gitlab-mr-diffs'
gitlab_rails['object_store']['objects']['lfs']['bucket'] = 'gitlab-lfs'
gitlab_rails['object_store']['objects']['uploads']['bucket'] = 'gitlab-
uploads'
gitlab_rails['object_store']['objects']['packages']['bucket'] = 'gitlab-
packages'
gitlab_rails['object_store']['objects']['dependency_proxy']['bucket'] =
'gitlab-dependency-proxy'
gitlab_rails['object_store']['objects']['terraform_state']['bucket'] =
'gitlab-terraform-state'
gitlab_rails['object_store']['objects']['pages']['bucket'] = 'gitlab-
pages'
```

절차 및 API 예

StorageGRID에서 S3 암호화 옵션 테스트 및 시연

StorageGRID와 S3 API를 사용하면 유휴 데이터를 다양한 방법으로 암호화할 수 있습니다. 자세한 내용은 을 참조하십시오 "StorageGRID 암호화 방법을 검토합니다".

이 가이드에서는 S3 API 암호화 방법을 보여 줍니다.

서버 측 암호화(SSE)

SSE를 사용하면 클라이언트가 개체를 저장하고 StorageGRID에서 관리하는 고유 키로 암호화할 수 있습니다. 개체가 요청되면 StorageGRID 에 저장된 키에 의해 개체가 해독됩니다.

SSE 예

- SSE가 있는 개체를 넣습니다

```
aws s3api put-object --bucket <bucket>  --key <file> --body "<file>"
--server-side-encryption AES256 --endpoint-url https://s3.example.com
```

- 객체를 확인하여 암호화를 확인합니다

```
aws s3api head-object --bucket <bucket>  --key <file>   --endpoint-url
https://s3.example.com
```

```
{
    "AcceptRanges": "bytes",
    "LastModified": "2022-05-02T19:03:03+00:00",
    "ContentLength": 47,
    "ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
    "ContentType": "text/plain",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
```

- 객체를 가져옵니다

```
aws s3api get-object --bucket <bucket>  --key <file> <file> --endpoint
-url https://s3.example.com
```

고객 제공 키(SSE-C)를 사용한 서버측 암호화

SSE를 사용하면 클라이언트가 객체를 저장하고 해당 객체와 함께 제공된 고유 키를 사용하여 해당 객체를 암호화할 수 있습니다. 개체가 요청될 때 개체를 해독하고 반환하려면 동일한 키를 제공해야 합니다.

SSE-C의 예

- 테스트 또는 데모용으로 암호화 키를 만들 수 있습니다
  - 암호화 키를 만듭니다

```
openssl enc -aes-128-cbc -pass pass:secret -P`
```

```
salt=E9DBB6603C7B3D2A
key=23832BAC16516152E560F933F261BF03
iv =71E87C0F6EC3C45921C2754BA131A315
```

- 생성된 키가 있는 개체를 넣습니다

```
aws s3api put-object --bucket <bucket> --key <file> --body "file" --sse
-customer-algorithm AES256 --sse-customer-key
23832BAC16516152E560F933F261BF03 --endpoint-url https://s3.example.com
```

- 물체를 향하십시오

```
aws s3api head-object --bucket <bucket> --key <file> --sse-customer
-algorithm AES256 --sse-customer-key 23832BAC16516152E560F933F261BF03
--endpoint-url https://s3.example.com
```

```
            {
                "AcceptRanges": "bytes",
                "LastModified": "2022-05-02T19:20:02+00:00",
                "ContentLength": 47,
                "ETag": "\"f92ef20ab87e0e13951d9bee862e9f9a\"",
                "ContentType": "binary/octet-stream",
                "Metadata": {},
                "SSECustomerAlgorithm": "AES256",
                "SSECustomerKeyMD5": "rjGuMdjLpPV1eRuotNaPMQ=="
            }
```

암호화 키를 제공하지 않으면 "HeadObject 작업을 호출할 때 오류 발생(404): 찾을 수 없음" 오류가 발생합니다.

- 객체를 가져옵니다

```
aws s3api get-object --bucket <bucket> --key <file> <file> --sse
-customer-algorithm AES256 --sse-customer-key
23832BAC16516152E560F933F261BF03 --endpoint-url https://s3.example.com
```

암호화 키를 제공하지 않으면 "GetObject 작업을 호출할 때 오류가 발생했습니다(InvalidRequest). 즉, 개체가 서버측 암호화 형식을 사용하여 저장되었습니다."라는 오류가 발생합니다. 객체를 검색하려면 올바른 매개 변수를 제공해야 합니다."

버킷 서버 측 암호화(SSE-S3)

SSE-S3를 사용하면 클라이언트가 버킷에 저장된 모든 오브젝트에 대해 기본 암호화 동작을 정의할 수 있습니다. 개체는 StorageGRID 에서 관리하는 고유 키로 암호화됩니다. 개체가 요청되면 StorageGRID 에 저장된 키에 의해 개체가 해독됩니다.

버킷 **SSE-S3**의 예

- 새 버킷을 생성하고 기본 암호화 정책을 설정합니다
  - 새 버킷을 생성합니다

```
aws s3api create-bucket --bucket <bucket> --region us-east-1
--endpoint-url https://s3.example.com
```

  - 버킷 암호화

```
aws s3api put-bucket-encryption --bucket <bucket> --server-side
-encryption-configuration '{"Rules":
[{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm":
"AES256"}}]}' --endpoint-url https://s3.example.com
```

- 물체를 버킷에 넣으십시오

```
aws s3api put-object --bucket <bucket> --key <file> --body "file"
--endpoint-url https://s3.example.com
```

- 물체를 향하십시오

```
aws s3api head-object --bucket <bucket> --key <file> --endpoint-url
https://s3.example.com
```

```
{
"AcceptRanges": "bytes",
"LastModified": "2022-05-02T20:16:23+00:00",
"ContentLength": 47,
"ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
"ContentType": "binary/octet-stream",
"ServerSideEncryption": "AES256",
"Metadata": {}
}
```

- 객체를 가져옵니다

```
aws s3api get-object --bucket <bucket>  --key <file> <file> --endpoint
-url https://s3.example.com
```

_ 아론 클라인 _

StorageGRID에서 S3 오브젝트 잠금을 테스트하고 시연합니다

Object Lock은 개체가 삭제되거나 덮어써지는 것을 방지하는 WORM 모델을 제공합니다. StorageGRID의 객체 잠금 구현은 규제 요구사항을 충족하고, 객체 보존에 대한 법적 보류 및 규정 준수 모드를 지원하고, 기본 버킷 보존 정책을 지원하는 데 도움이 되는 코호트 평가입니다.

이 가이드에서는 S3 오브젝트 잠금 API를 보여 줍니다.

법적 증거 자료 보관

- 개체 잠금 법적 보류는 개체에 적용되는 간단한 켜기/끄기 상태입니다.

```
aws s3api put-object-legal-hold --bucket <bucket> --key <file> --legal
-hold Status=ON --endpoint-url https://s3.company.com
```

- 가져오기 작업으로 확인합니다.

```
aws s3api get-object-legal-hold --bucket <bucket> --key <file>
--endpoint-url https://s3.company.com
```

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- 법적 증거 자료 보관 기능을 끕니다

```
aws s3api put-object-legal-hold --bucket <bucket> --key <file> --legal
-hold Status=OFF --endpoint-url https://s3.company.com
```

- 가져오기 작업으로 확인합니다.

```
aws s3api get-object-legal-hold --bucket <bucket> --key <file>
--endpoint-url https://s3.company.com
```

```
{
    "LegalHold": {
        "Status": "OFF"
    }
}
```

준수 모드

- 개체 보존은 타임스탬프까지 보존하여 수행됩니다.

```
aws s3api put-object-retention --bucket <bucket> --key <file>
--retention '{"Mode":"COMPLIANCE", "RetainUntilDate": "2025-06-
10T16:00:00"}' --endpoint-url https://s3.company.com
```

- 보존 상태를 확인합니다

```
aws s3api get-object-retention --bucket <bucket> --key <file> --endpoint
-url https://s3.company.com
+
```

```
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2025-06-10T16:00:00+00:00"
    }
}
```

기본 보존

- 객체 API에 정의된 보존 종료 날짜를 기준으로 보존 기간을 일 및 년 단위로 설정합니다.

```
aws s3api put-object-lock-configuration --bucket <bucket> --object-lock
-configuration '{"ObjectLockEnabled": "Enabled", "Rule": {
"DefaultRetention": { "Mode": "COMPLIANCE", "Days": 10 }}}' --endpoint
-url https://s3.company.com
```

- 보존 상태를 확인합니다

```
aws s3api get-object-lock-configuration --bucket <bucket> --endpoint-url
https://s3.company.com
```

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 10
            }
        }
    }
}
```

- 물체를 버킷에 넣으십시오

```
aws s3api put-object --bucket <bucket> --key <file> --body "file"
--endpoint-url https://s3.example.com
```

- 버킷에 설정된 보존 기간은 객체의 보존 타임 스탬프로 변환됩니다.

```
aws s3api get-object-retention --bucket <bucket> --key <file> --endpoint
-url https://s3.company.com
```

```
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-03-02T15:22:47.202000+00:00"
    }
}
```

정의된 보존 개체가 있는 개체 삭제 테스트

오브젝트 잠금은 버전 관리를 기반으로 합니다. 보존은 개체 버전에 정의됩니다. 보존이 정의되어 있고 버전이 지정되지 않은 개체를 삭제하려고 하면 삭제 표식이 개체의 현재 버전으로 만들어집니다.

- 보존이 정의된 개체를 삭제합니다

```
aws s3api delete-object --bucket <bucket> --key <file> --endpoint-url
https://s3.example.com
```

- 버킷의 물체를 나열합니다

```
aws s3api list-objects --bucket <bucket> --endpoint-url
https://s3.example.com
```

  - 개체가 나열되지 않은 것을 확인합니다.
- 버전 목록을 사용하여 삭제 마커와 원래 잠긴 버전을 확인합니다

```
aws s3api list-object-versions --bucket <bucket> --prefix <file>
--endpoint-url https://s3.example.com
```

```
{
    "Versions": [
        {
            "ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
            "Size": 47,
            "StorageClass": "STANDARD",
            "Key": "file.txt",
            "VersionId":
"RDVDMjYwMTQtQkNDQS0xMUVDLThGOEUtNjQ3NTAwQzAxQTk1",
            "IsLatest": false,
            "LastModified": "2022-04-15T14:46:29.734000+00:00",
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "56622399308951294926"
            }
        }
    ],
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "56622399308951294926"
            },
            "Key": "file01.txt",
            "VersionId":
"QjVDQzgzOTAtQ0FGNi0xMUVDLThFMzgtQ0RGMjAwQjk0MjM1",
            "IsLatest": true,
            "LastModified": "2022-05-03T15:35:50.248000+00:00"
        }
    ]
}
```

- 객체의 잠긴 버전을 삭제합니다

```
aws s3api delete-object  --bucket <bucket> --key <file> --version-id
"<VersionId>" --endpoint-url https://s3.example.com
```

```
An error occurred (AccessDenied) when calling the DeleteObject
operation: Access Denied
```

_ 아론 클라인 _

버킷 및 그룹(IAM) 정책의 예

다음은 버킷 정책 및 그룹 정책(IAM 정책)의 예입니다.

그룹 정책(IAM)

홈 디렉토리 스타일 버킷 액세스

이 그룹 정책은 사용자가 사용자 이름이 인 버킷의 객체에 액세스하는 것만 허용합니다.

```
"Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::home",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::home/?/?/${aws:username}/*"
    }

  ]
}
```

오브젝트 잠금 버킷 생성을 거부합니다

이 그룹 정책은 사용자가 버킷에 개체 잠금이 설정된 버킷을 생성할 수 없도록 제한합니다.

> 이 정책은 StorageGRID UI에서 적용되지 않으며 S3 API에서만 적용됩니다.

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

개체 잠금 보존 제한

이 버킷 정책은 객체 잠금 보존 기간을 10일 이하로 제한합니다

```
{
 "Version":"2012-10-17",
 "Id":"CustSetRetentionLimits",
 "Statement": [
   {
    "Sid":"CustSetRetentionPeriod",
    "Effect":"Deny",
    "Principal":"*",
    "Action": [
      "s3:PutObjectRetention"
    ],
    "Resource":"arn:aws:s3:::testlock-01/*",
    "Condition": {
      "NumericGreaterThan": {
        "s3:object-lock-remaining-retention-days":"10"
      }
    }
   }
  ]
}
```

버전 **ID**를 기준으로 개체를 삭제하지 못하도록 제한합니다

이 그룹 정책은 버전 ID를 기준으로 버전이 지정된 개체를 삭제하지 못하도록 제한합니다

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

이 버킷 정책은 사용자 ID "56622399308951294926"으로 식별된 사용자(versionID로 식별됨)가 버전 ID로 버전이 지정된 객체를 삭제하지 못하도록 제한합니다

```
{
  "Statement": [
    {
      "Action": [
        "s3:DeleteObjectVersion"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::verdeny/*",
      "Principal": {
        "AWS": [
          "56622399308951294926"
        ]
      }
    },
    {
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::verdeny/*",
      "Principal": {
        "AWS": [
          "56622399308951294926"
        ]
      }
    }
  ]
}
```

읽기 전용 액세스 권한이 있는 단일 사용자로 버킷을 제한합니다

이 정책을 사용하면 단일 사용자가 버킷에 대한 읽기 전용 액세스를 가질 수 있고 다른 모든 사용자에 대한 액세스를 명시적으로 부인할 수 있습니다. 정책 맨 위에 있는 Deny 문을 그룹화하는 것은 보다 빠른 평가를 위한 좋은 방법입니다.

```
{
    "Statement": [
        {
            "Sid": "Deny non user1",
            "Effect": "Deny",
            "NotPrincipal": {
                "AWS":
"urn:sgws:identity::34921514133002833665:user/user1"
            },
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "urn:sgws:s3:::bucket1",
                "urn:sgws:s3:::bucket1/*"
            ]
        },
        {
            "Sid": "Allow user1 read access to bucket bucket1",
            "Effect": "Allow",
            "Principal": {
                "AWS":
"urn:sgws:identity::34921514133002833665:user/user1"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "urn:sgws:s3:::bucket1",
                "urn:sgws:s3:::bucket1/*"
            ]
        }
    ]
}
```

그룹을 읽기 전용 권한으로 단일 하위 디렉토리(접두사)로 제한합니다

이 정책을 사용하면 그룹의 구성원이 버킷 내의 하위 디렉터리(접두사)에 읽기 전용 액세스 권한을 가질 수 있습니다. 버킷 이름은 "study"이고 하위 디렉토리는 "study01"입니다.

```
{
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
```

```
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowRootAndstudyListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3::: study"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "",
                        "study01/"
                    ],
                    "s3:delimiter": [
                        "/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfstudy01",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::study"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "study01/*"
                    ]
                }
            }
        },
```

```
        {
            "Sid": "AllowAllS3ActionsInstudy01Folder",
            "Effect": "Allow",
            "Action": [
                "s3:Getobject"
            ],
            "Resource": [
                "arn:aws:s3:::study/study01/*"
            ]
        }
    ]
}
```

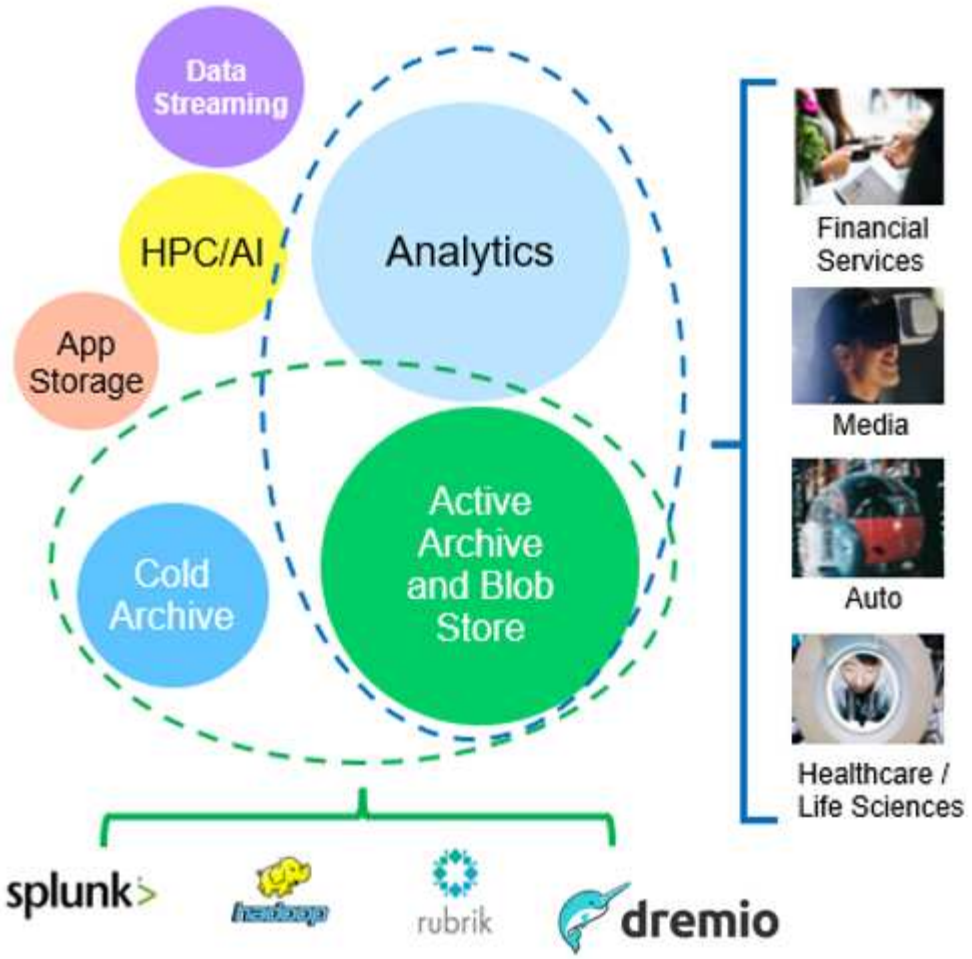
기술 보고서

NetApp StorageGRID 및 빅데이터 분석

NetApp StorageGRID 사용 사례

NetApp StorageGRID 오브젝트 스토리지 솔루션은 확장성, 데이터 가용성, 보안 및 고성능을 제공합니다. 모든 규모와 다양한 산업 분야의 조직이 광범위한 사용 사례에 StorageGRID S3를 사용합니다. 몇 가지 일반적인 시나리오를 살펴보겠습니다.

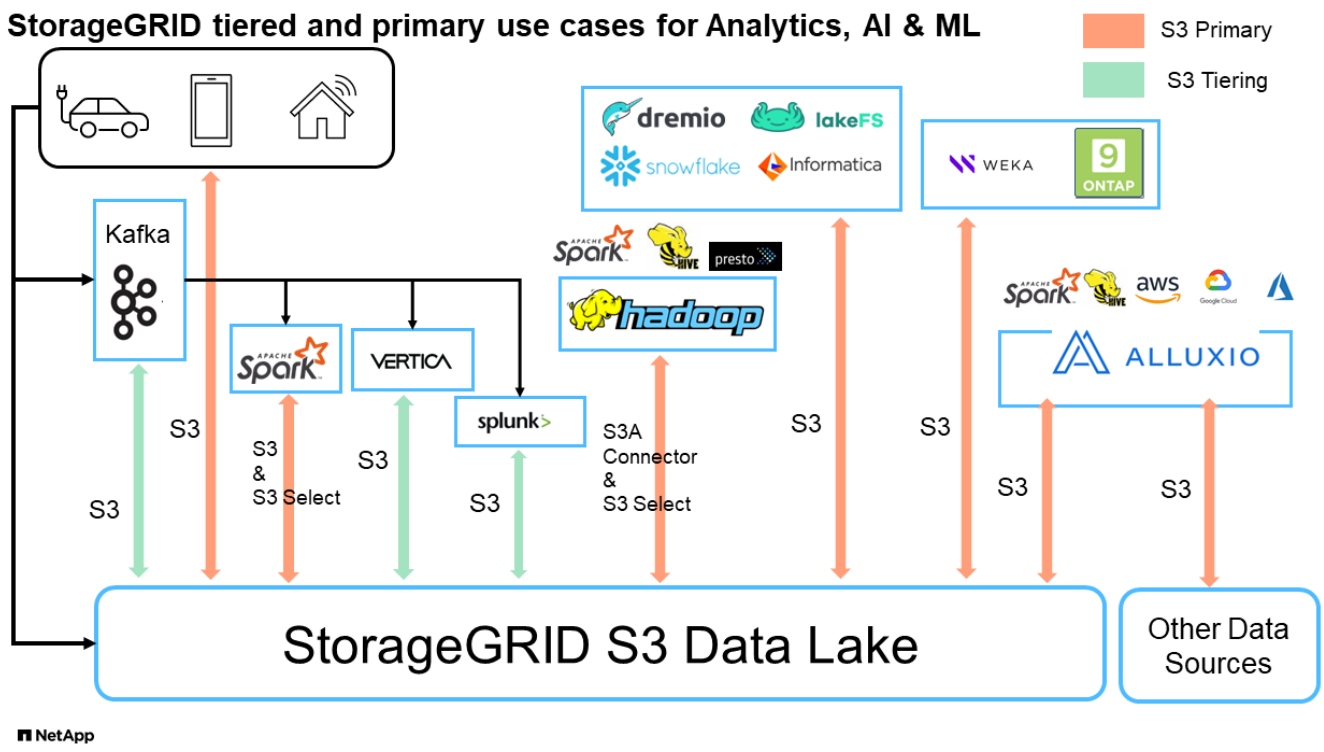
- 빅 데이터 분석: * StorageGRID S3는 기업에서 Apache Spark, Splunk Smartstore 및 Dremio와 같은 도구를 사용하여 분석을 위해 대량의 정형 및 비정형 데이터를 저장하는 데이터 레이크로 자주 사용됩니다.
- 데이터 계층화: * NetApp 고객은 ONTAP의 FabricPool 기능을 사용하여 고성능 로컬 계층 간에 데이터를 자동으로 StorageGRID로 이동합니다. 계층화하면 콜드 데이터를 저렴한 오브젝트 스토리지에서 즉시 사용 가능한 상태로 유지하면서 고가의 플래시 스토리지를 핫 데이터용으로 확보할 수 있습니다. 따라서 성능과 비용 절감 효과가 극대화됩니다.
- 데이터 백업 및 재해 복구: * 기업에서는 StorageGRID S3를 안정적이고 비용 효율적인 솔루션으로 사용하여 중요한 데이터를 백업하고 재해 발생 시 복구할 수 있습니다.
- 응용 프로그램용 데이터 저장: * StorageGRID S3는 응용 프로그램용 스토리지 백엔드로 사용할 수 있으므로 개발자가 파일, 이미지, 비디오 및 기타 유형의 데이터를 쉽게 저장하고 검색할 수 있습니다.
- 콘텐츠 전송: * StorageGRID S3를 사용하여 정적 웹 사이트 콘텐츠, 미디어 파일 및 소프트웨어 다운로드를 전 세계 사용자에게 저장하고 제공할 수 있으며, StorageGRID의 지리적 분산 및 글로벌 네임스페이스를 활용하여 빠르고 안정적인 콘텐츠 전송을 수행할 수 있습니다.
- 데이터 계층화: * NetApp 고객은 ONTAP FabricPool 기능을 사용하여 고성능 로컬 계층 간에 데이터를 자동으로 StorageGRID로 이동합니다. 계층화하면 콜드 데이터를 저렴한 오브젝트 스토리지에서 즉시 사용 가능한 상태로 유지하면서 고가의 플래시 스토리지를 핫 데이터용으로 확보할 수 있습니다. 따라서 성능과 비용 절감 효과가 극대화됩니다.
- 데이터 아카이브: * StorageGRID는 다양한 스토리지 유형을 제공하고 공용 장기 저비용 스토리지 옵션에 대한 계층화를 지원하며, 규정 준수 또는 기간별 목적으로 보존해야 하는 데이터의 보관 및 장기 보존에 이상적인 솔루션입니다.
- 오브젝트 스토리지 사용 사례 *



위 중 빅 데이터 분석은 최상위 사용 사례 중 하나이며 사용량이 증가하고 있습니다.

데이터 레이크에 **StorageGRID**를 사용해야 하는 이유

- 협업 증가 - 업계 표준 API 액세스를 지원하는 대규모 공유 멀티 사이트 멀티 테넌시
- 운영 비용 절감 - 자동 복구, 자동화된 단일 스케일아웃 아키텍처를 통해 운영 간소화
- 확장성 - 기존 Hadoop 및 데이터 웨어하우스 솔루션과 달리 StorageGRID S3 오브젝트 스토리지는 스토리지를 컴퓨팅과 데이터와 분리하므로 기업이 필요에 따라 스토리지 요구사항을 확장할 수 있습니다.
- 내구성 및 안정성 - StorageGRID는 99.9999999%의 내구성을 제공하여 저장된 데이터가 데이터 손실에 대한 저항성이 높습니다. 또한 고가용성을 제공하여 데이터에 항상 액세스할 수 있도록 보장합니다.
- 보안-StorageGRID는 암호화, 액세스 제어 정책, 데이터 라이프사이클 관리, 오브젝트 잠금 및 버전 관리와 같은 다양한 보안 기능을 제공하여 S3 버킷에 저장된 데이터를 보호합니다
- StorageGRID S3 데이터 레이크 *



S3 오브젝트 스토리지에서 가장 잘 작동하는 데이터 웨어하우스 또는 데이터 레이크가 있습니다

NetApp는 Hive, Delta Lake 및 Dremio 등 3개의 데이터 웨어하우스/호수 하우스 에코시스템을 통해 StorageGRID의 벤치마크를 수행했습니다. "아파치 아이스버그: 확실한 가이드" 데이터 웨어하우스 및 데이터 레이크 하우스에 대한 간략한 소개와 이 두 아키텍처의 장점 포함

- 벤치마크 도구 - TPC-DS - https://www.tpc.org/tpcds/
- 빅 데이터 에코시스템
  - VM 5개로 구성된 클러스터, 각각 128G RAM 및 24개의 vCPU, 시스템 디스크용 SSD 스토리지
  - Hive 3.1.3 포함 Hadoop 3.3.5(이름 노드 1개 + 데이터 노드 4개)
  - Spark 3.0.0(마스터 1명 + 작업자 4명) 및 Hadoop 3.3.5 지원 델타 레이크
  - Dremio v23(마스터 1개 + 실행자 4개)
- 오브젝트 스토리지
  - NetApp ® StorageGRID ® 11.6(SG6060 + SG1000 로드 밸런서 3개 포함
  - 오브젝트 보호 - 복사본 2개
- 데이터베이스 크기 1000GB
- 각 쿼리 테스트에 대해 일관된 결과를 얻기 위해 3개 에코시스템에서 캐시를 모두 사용할 수 없습니다.

TPC-DS에는 쿼리 벤치마킹을 위한 99개의 복잡한 SQL 쿼리가 포함되어 있습니다. 전체 시간(분)을 측정하여 99개의 쿼리를 모두 완료했으며 결과를 분석하기 위한 S3 요청의 유형과 수를 세분화하여 더 깊이 있게 조사했습니다. 아래의 첫 번째 표는 99개의 모든 쿼리의 총 기간을 보여 주며 두 번째 표는 각 에코시스템에서 StorageGRID로 전송된 S3 요청의 수와 유형을 요약한 것입니다.

- TPC-DS 쿼리 결과 *

에코시스템	하이브	델타 레이크	드리미오
지원합니다	NetApp (영어) StorageGRID (영어	NetApp (영어) StorageGRID (영어	NetApp (영어) StorageGRID (영어
드라이브 유형입니다	HDD	HDD	HDD
표 형식	마루	마루	마루 1
데이터베이스 크기	1000g	1000g	1000g
TPCDS 99 쿼리 총 시간(분)	1084 - 2도	55	47

파케(Parquet)와 아이스버그(Iceberg) 테이블 형식을 모두 테스트한 결과도 이와 유사합니다.

1. 조회 번호 72를 완료할 수 없습니다.
    - TPC-DS 쿼리 - S3 요청 분석 *

S3 요청	하이브	델타 레이크	드리미오
가져오기	1,117,184	2,074,610	4,414,227
관찰: 모든 제품군 GET	80% 범위 32MB 객체에서 2KB ~ 2MB, 초당 50 ~ 100회의 요청	73% 범위는 32MB 개체에서 100KB 미만으로, 초당 1000 - 1400개의 요청 수입니다	90% 1M 바이트 범위는 256MB 객체, 초당 2000-2300개 요청에서 가져옵니다
개체 나열	312,053입니다	24,158입니다	240
머리 (존재하지 않는 객체)	156,027	12,103	192
머리 (존재하는 객체)	982,126	922,732	1,845
총 요청 수입니다	2,567,390입니다	3,033,603입니다	4,416,504입니다

첫 번째 테이블에서, 우리는 델타 호수와 Dremio가 Hive보다 훨씬 더 빠르다는 것을 볼 수 있습니다. 두 번째 표에서 Hive는 많은 S3 목록 오브젝트 요청을 전송했습니다. 이 요청은 모든 오브젝트 스토리지 플랫폼에서 일반적으로 느리며, 특히 많은 오브젝트가 포함된 버킷을 다룰 경우 매우 느립니다. 따라서 전체 쿼리 기간이 크게 증가합니다. 또 다른 관찰은 Dremio가 Hive에서 초당 50-100개의 요청을 처리하는 데 비해 초당 2,000-2,300개의 요청을 동시에 보낼 수 있다는 것입니다. Hive 및 Hadoop S3A는 표준 파일 시스템을 모방하여 S3 오브젝트 스토리지에 Hive 느림 효과를 제공합니다.

Hive 또는 Spark와 함께 Hadoop(HDFS 또는 S3 오브젝트 스토리지)을 사용하려면 Hadoop 및 Hive/Spark와 각 서비스의 설정이 상호 작용하는 방법에 대한 폭넓은 지식이 필요합니다. 이러한 두 서비스의 설정이 1,000개 이상인 경우입니다. 설정은 서로 관련이 있는 경우가 매우 많으며 단독으로 변경할 수 없습니다. 사용할 설정과 값의 최적 조합을 찾기 위해서는 엄청난 시간과 노력이 필요합니다.

Dremio는 완벽한 Apache Arrow를 사용하여 쿼리 성능을 획기적으로 향상하는 데이터 레이크 엔진입니다. Apache Arrow는 효율적인 데이터 공유와 빠른 분석을 위해 표준화된 원주 메모리 형식을 제공합니다. Arrow는 데이터 serialization 및 deserialization의 필요성을 제거하여 복잡한 데이터 프로세스와 시스템 간의 성능 및 상호 운용성을 향상시키도록 설계된 언어 독립적 접근 방식을 사용합니다.

Dremio의 성능은 주로 Dremio 클러스터의 컴퓨팅 성능에 의해 좌우됩니다. Dremio는 S3 오브젝트 스토리지 연결에 Hadoop의 S3A 커넥터를 사용하지만 Hadoop은 필요하지 않으며 대부분의 Hadoop의 fs.s3a 설정은 Dremio에서

사용되지 않습니다. 따라서 다양한 Hadoop s3a 설정을 배우고 테스트하는 데 시간을 들이지 않고도 Dremio 성능을 손쉽게 튜닝할 수 있습니다.

이러한 벤치마크 결과에서 알 수 있듯이 S3 기반 워크로드에 최적화된 빅데이터 분석 시스템이 주요 성능 요인이라는 결론을 내릴 수 있습니다. Dremio는 쿼리 실행을 최적화하고, 메타데이터를 효율적으로 사용하며, S3 데이터에 대한 원활한 액세스를 제공하므로 S3 스토리지로 작업할 때 Hive에 비해 성능이 향상됩니다. 이를 참조하십시오 "페이지" StorageGRID를 사용하여 Dremio S3 데이터 소스를 구성합니다.

아래 링크를 방문하여 StorageGRID와 Dremio가 함께 작동하여 현대적이고 효율적인 데이터 레이크 인프라를 제공하는 방법과 NetApp가 Hive+ HDFS에서 Dremio+ StorageGRID로 마이그레이션하여 빅데이터 분석 효율성을 획기적으로 개선한 방법에 대해 자세히 알아보십시오.

- "NetApp StorageGRID로 빅데이터의 성능을 향상하십시오"
- "StorageGRID 및 Dremio를 사용하는 현대적이고 강력하고 효율적인 데이터 레이크 인프라"
- "NetApp이 제품 분석을 통해 고객 경험을 재정의하는 방법"

Hadoop S3A 튜닝

Hadoop S3A 커넥터는 Hadoop 기반 애플리케이션과 S3 오브젝트 스토리지 간의 원활한 상호 작용을 지원합니다. S3 오브젝트 스토리지로 작업할 때 성능을 최적화하려면 Hadoop S3A Connector를 튜닝해야 합니다. 세부 조정을 시작하기에 앞서, Hadoop과 그 구성 요소에 대한 기본적인 이해를 갖췄습니다.

Hadoop이란?

- Hadoop * 은 대규모 데이터 처리 및 스토리지를 처리하도록 설계된 강력한 오픈 소스 프레임워크입니다. 이를 통해 컴퓨터 클러스터 간에 분산 스토리지 및 병렬 처리가 가능합니다.

Hadoop의 3가지 핵심 구성 요소는 다음과 같습니다.

- * Hadoop HDFS (Hadoop 분산 파일 시스템) * : 스토리지를 처리하고 데이터를 블록으로 분할하여 노드에 분산시킵니다.
- * Hadoop MapReduce *: 작업을 작은 청크로 분할하고 병렬로 실행하여 데이터를 처리합니다.
- * Hadoop YARN (또 다른 리소스 협상 담당자) : * "리소스를 관리하고 작업을 효율적으로 예약합니다"

Hadoop HDFS 및 S3A 커넥터

HDFS는 Hadoop 에코시스템의 핵심 구성 요소로, 효율적인 빅 데이터 처리에 중요한 역할을 합니다. HDFS는 안정적인 스토리지 및 관리를 지원합니다. 병렬 처리 및 최적화된 데이터 스토리지를 보장하여 데이터 액세스 및 분석 속도가 빨라집니다.

빅데이터 처리 시 HDFS는 대규모 데이터 세트를 위한 내결함성 스토리지를 제공하는 데 탁월합니다. 이 점은 데이터 복제를 통해 실현됩니다. 데이터 웨어하우스 환경에서 대량의 정형 데이터와 비정형 데이터를 저장하고 관리할 수 있습니다. 또한 Apache Spark, Hive, Pig 및 Flink와 같은 선도적인 빅 데이터 처리 프레임워크와 원활하게 통합되어 확장 가능하고 효율적인 데이터 처리가 가능합니다. Unix 기반(Linux) 운영 체제와 호환되기 때문에 빅 데이터 처리를 위해 Linux 기반 환경을 사용하는 것을 선호하는 기업에 이상적인 선택입니다.

시간이 지나면서 데이터 볼륨이 증가함에 따라 자체 컴퓨팅 및 스토리지를 사용하여 Hadoop 클러스터에 새 시스템을 추가하는 방식이 비효율적이 되었습니다. 선형적으로 확장하면 리소스를 효율적으로 사용하고 인프라를 관리하는 데 어려움이 발생합니다.

이러한 과제를 해결하기 위해 Hadoop S3A 커넥터는 S3 오브젝트 스토리지에 대한 고성능 I/O를 제공합니다. S3A를 사용하여 Hadoop 워크플로우를 구축하면 오브젝트 스토리지를 데이터 저장소로 활용할 수 있으며, 컴퓨팅과 스토리지를 독립적으로 확장할 수 있는 분리된 컴퓨팅 및 스토리지를 사용할 수 있습니다. 또한 컴퓨팅과 스토리지를 분리하면 컴퓨팅 작업에 적절한 양의 리소스를 투입하고 데이터 세트의 크기에 따라 용량을 제공할 수 있습니다. 따라서 Hadoop 워크플로우의 전체 TCO를 줄일 수 있습니다.

Hadoop S3A 커넥터 튜닝

S3는 HDFS와 다르게 동작하며 파일 시스템의 모양을 유지하려는 일부 시도는 공격적으로 최적화되지 않습니다. S3 리소스를 가장 효율적으로 활용하려면 신중한 튜닝/테스트/실험이 필요합니다.

이 문서의 Hadoop 옵션은 Hadoop 3.3.5를 기반으로 합니다. 을 참조하십시오 "Hadoop 3.3.5 core-site.xml" 사용 가능한 모든 옵션

참고 – 일부 Hadoop fs.s3a 설정의 기본값은 Hadoop 버전마다 다릅니다. 현재 Hadoop 버전과 관련된 기본값을 확인하십시오. 이러한 설정이 Hadoop core-site.xml에 지정되지 않은 경우 기본값이 사용됩니다. Spark 또는 Hive 구성 옵션을 사용하여 런타임에 값을 재정의할 수 있습니다.

이 페이지로 이동해야 합니다 "아파치 하둡 페이지" 각 fs.s3a 옵션을 이해합니다. 가능한 경우 비운영 Hadoop 클러스터에서 테스트하여 최적의 값을 찾습니다.

읽어야 합니다 "S3A 커넥터로 작업할 때 성능을 극대화합니다" 기타 튜닝 권장 사항

몇 가지 주요 고려 사항을 살펴보겠습니다.

- 1. 데이터 압축 *

StorageGRID 압축을 활성화하지 마십시오. 대부분의 빅 데이터 시스템은 전체 객체를 검색하는 대신 바이트 범위 GET를 사용합니다. 압축된 객체와 함께 GET 바이트 범위를 사용하면 GET 성능이 크게 저하됩니다.

- 2. S3A 커밋 *

일반적으로 magic s3a committer가 권장됩니다. 이를 참조하십시오 "일반 S3A 커밋 옵션 페이지" 마법 커밋 및 관련 s3a 설정에 대한 더 나은 이해를 얻기 위해.

매직 커미터:

Magic Committer는 특히 S3Guard에 의존하여 S3 오브젝트 저장소에서 일관된 디렉토리 목록을 제공합니다.

이제 일관된 S3(이 경우)를 통해 Magic Committer를 모든 S3 버킷과 함께 안전하게 사용할 수 있습니다.

선택 및 실험:

사용 사례에 따라 클러스터 HDFS 파일 시스템에 의존하는 스테이징 커밋자와 Magic committer 중에서 선택할 수 있습니다.

두 가지를 모두 실험하여 귀사의 워크로드 및 요구사항에 가장 적합한 솔루션을 결정하십시오.

요약하면, S3A committers는 S3에 대한 일관적이고, 고성능의 안정적인 출력 약속이라는 근본적인 과제에 대한 솔루션을 제공합니다. 내부 설계로 데이터 무결성을 유지하면서 효율적인 데이터 전송을 보장합니다.

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.committer.name	Committer to create for output to S3A, one of: "file", "directory", "partitioned", "magic".	file
fs.s3a.buffer.dir	Local filesystem directory for data being written and/or staged.	${env.LOCAL_DIRS:-${hadoop.tmp.dir}}/s3a
fs.s3a.committer.magic.enabled	Enable “magic committer” support in the filesystem.	true
fs.s3a.committer.abort.pending.uploads	list and abort all pending uploads under the destination path when the job is committed or aborted.	true
fs.s3a.committer.threads	Number of threads in committers for parallel operations on files.	8
fs.s3a.committer.generate.uuid	Generate a Job UUID if none is passed down from Spark	false
fs.s3a.committer.require.uuid	Require the Job UUID to be passed down from Spark	false
mapreduce.fileoutputcommitter.marksuccessfuljobs	Write a _SUCCESS file on the successful completion of the job.	true
mapreduce.outputcommitter.factory.scheme.s3a	The committer factory to use when writing data to S3A filesystems. If mapreduce.outputcommitter.factory.class is set, it will override this property. (This property is set in mapred-default.xml)	org.apache.hadoop.fs.s3a.commit.S3ACommitterFactory

- 3. 스레드, 연결 풀 크기 및 블록 크기 *
- 단일 버킷과 상호 작용하는 각 * S3A * 클라이언트에는 업로드 및 복사 작업을 위한 개방형 HTTP 1.1 연결 전용 풀과 스레드가 있습니다.
- "이러한 풀 크기를 조정하여 성능과 메모리/스레드 사용량 간의 균형을 맞출 수 있습니다".
- S3에 데이터를 업로드하면 블록으로 나뉩니다. 기본 블록 크기는 32MB입니다. fs.s3a.block.size 속성을 설정하여 이 값을 사용자 지정할 수 있습니다.
- 블록 크기가 클수록 업로드 시 다중 파트 관리 오버헤드를 줄여 대용량 데이터 업로드의 성능을 향상시킬 수 있습니다. 대용량 데이터 세트의 권장 값은 256MB 이상입니다.

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.threads.max	The total number of threads available in the filesystem for data uploads *or any other queued filesystem operation*.	64
fs.s3a.connection.maximum	Controls the maximum number of simultaneous connections to S3. This must be bigger than the value of fs.s3a.threads.max so as to stop threads being blocked waiting for new HTTPS connections. Why not equal? The AWS SDK transfer manager also uses these connections.	96
fs.s3a.max.total.tasks	The number of operations which can be queued for execution. This is in addition to the number of active threads in fs.s3a.threads.max.	32
fs.s3a.committer.threads	Number of threads in committers for parallel operations on files (upload, commit, abort, delete...)	8
fs.s3a.executor.capacity	The maximum number of submitted tasks which is a single operation (e.g. rename(), delete()) may submit simultaneously for execution -excluding the IO-heavy block uploads, whose capacity is set in "fs.s3a.fast.upload.active.blocks" All tasks are submitted to the shared thread pool whose size is set in "fs.s3a.threads.max"; the value of capacity should be less than that of the thread pool itself, as the goal is to stop a single operation from overloading that thread pool.	16
fs.s3a.fast.upload.active.blocks (see also related fs.s3a.fast.upload.buffer option)	Maximum Number of blocks a single output stream can have active (uploading, or queued to the central FileSystem instance's pool of queued operations. This stops a single stream overloading the shared thread pool.	4
fs.s3a.block.size	Block size to use when reading files using s3a: file system. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value.	32MB (tested 1TB data set with 256MB and 512MB block size shows significant improvement in both read and write)

- 4. 멀티 파트 업로드 *

s3a committers * 항상 * MPU(멀티 파트 업로드)를 사용하여 데이터를 S3 버킷에 업로드합니다. 작업 실패, 추측에 의한 작업 실행, 커밋 전 작업 중단 등을 허용하기 위해 필요합니다. 다음은 다중 파트 업로드와 관련된 몇 가지 주요 사양입니다.

- 최대 개체 크기: 5TiB(테라바이트)
- 업로드당 최대 부품 수: 10,000.
- 부품 번호: 1 ~ 10,000 범위(포함).
- 부품 크기: 5MiB에서 5GiB 사이. 멀티 파트 업로드의 마지막 부분에 대한 최소 크기 제한은 없습니다.

S3 멀티 파트 업로드에 더 작은 파트 크기를 사용하면 장점과 단점이 모두 있습니다.

- 장점 *:
- 네트워크 문제에서 빠른 복구: 작은 부품을 업로드하면 네트워크 오류로 인해 실패한 업로드를 다시 시작할 때의 영향이 최소화됩니다. 부품에 오류가 발생하면 전체 오브젝트가 아닌 특정 부분만 다시 업로드하면 됩니다.
- 향상된 병렬 처리: 다중 스레딩 또는 동시 연결을 활용하여 더 많은 파트를 병렬로 업로드할 수 있습니다. 이 병렬화는 특히 큰 파일을 처리할 때 성능을 향상시킵니다.

- 단점 *:
- 네트워크 오버헤드: 파트 크기가 작을수록 업로드할 파트가 더 많아지며 각 파트마다 자체 HTTP 요청이 필요합니다. HTTP 요청이 많을수록 개별 요청을 시작 및 완료하는 데 따르는 오버헤드가 증가합니다. 많은 수의 작은 파트를 관리하면 성능에 영향을 줄 수 있습니다.
- 복잡성: 주문 관리, 부품 추적, 성공적인 업로드 보장은 번거로울 수 있습니다. 업로드를 중단해야 하는 경우 이미 업로드한 모든 부품을 추적하고 제거해야 합니다.

Hadoop의 경우 fs.s3a.multipart.size에 256MB 이상의 파트 크기가 권장됩니다. 항상 fs.s3a.mutlipart.threshold 값을 2 x fs.s3a.multipart.size 값으로 설정하십시오. 예를 들어 fs.s3a.multipart.size=256M,fs.s3a.mullipart.threshold는 512M이어야 합니다.

대형 데이터 세트에 더 큰 파트 크기를 사용합니다. 특정 사용 사례와 네트워크 상태에 따라 이러한 요소의 균형을 맞추는 부품 크기를 선택하는 것이 중요합니다.

다중 부분 업로드는 입니다 "3단계 프로세스":

1. 업로드가 시작되면 StorageGRID에서 업로드 ID를 반환합니다.
2. 개체 부분은 upload-id를 사용하여 업로드됩니다.
3. 모든 객체 부분이 업로드되면 는 업로드 ID와 함께 완전한 멀티 파트 업로드 요청을 보냅니다. StorageGRID는 업로드된 부분에서 객체를 생성하며 클라이언트는 객체에 액세스할 수 있습니다.

전체 다중 파트 업로드 요청이 성공적으로 전송되지 않으면 부품은 StorageGRID에 남아 있고 객체를 생성하지 않습니다. 이 문제는 작업이 중단, 실패 또는 중단될 때 발생합니다. 업로드가 시작된 후 15일이 경과하면 멀티 파트 업로드가 완료되거나 중단되거나 StorageGRID가 이러한 부품을 제거할 때까지 파트가 그리드에 남아 있습니다. 버킷에 여러 개의(수억 ~ 수백만) 진행 중인 멀티 파트 업로드가 있는 경우 Hadoop이 'list-multipart-uploads'를 전송할 때(이 요청은 업로드 ID로 필터링되지 않음) 요청을 완료하는 데 시간이 오래 걸리거나 시간이 초과될 수 있습니다. 적절한 fs.s3a.mutlipart.purge를 true로 설정하여 적절한 fs.s3a.multipart.purge.age 값을 설정할 수 있습니다(예: 5-7일, 기본값 86400, 즉 1일을 사용하지 마십시오). 또는 NetApp 지원 팀에 문의하여 상황을 조사하십시오.

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.multipart.size	How big (in bytes) to split upload or copy operations up into. A suffix from the set {K,M,G,T,P} may be used to scale the numeric value.	64M
fs.s3a.multipart.threshold	How big (in bytes) to split upload or copy operations up into. This also controls the partition size in renamed files, as rename() involves copying the source file(s). A suffix from the set {K,M,G,T,P} may be used to scale the numeric value.	128M
fs.s3a.multipart.purge	True if you want to purge existing multipart uploads that may not have been completed/aborted correctly. The corresponding purge age is defined in fs.s3a.multipart.purge.age. If set, when the filesystem is instantiated then all outstanding uploads older than the purge age will be terminated -across the entire bucket. This will impact multipart uploads by other applications and users. so should be used sparingly, with an age value chosen to stop failed uploads, without breaking ongoing operations.	false
fs.s3a.multipart.purge.age	Minimum age in seconds of multipart uploads to purge on startup if "fs.s3a.multipart.purge" is true	86400

- 5. 메모리의 버퍼 쓰기 데이터 *

성능을 높이기 위해 쓰기 데이터를 S3에 업로드하기 전에 메모리에 버퍼링할 수 있습니다. 이렇게 하면 작은 쓰기 수를 줄이고 효율성을 높일 수 있습니다.

Option	Meaning	Default (Hadoop 3.3.5)
fs.s3a.fast.upload.buffer	The buffering mechanism to for data being written. Values: disk, array, bytebuffer. "disk" will use the directories listed in fs.s3a.buffer.dir as the location(s) to save data prior to being uploaded. "array" uses arrays in the JVM heap "bytebuffer" uses off-heap memory within the JVM. Both "array" and "bytebuffer" will consume memory in a single stream up to the number of blocks set by: fs.s3a.multipart.size * fs.s3a.fast.upload.active.blocks. If using either of these mechanisms, keep this value low The total number of threads performing work across all threads is set by fs.s3a.threads.max, with fs.s3a.max.total.tasks values setting the number of queued work items.	disk

S3 및 HDFS는 서로 다른 방식으로 작동한다는 점을 기억하십시오. S3 리소스를 가장 효율적으로 활용하려면 신중한

튜닝/테스트/실험이 필요합니다.

NetApp StorageGRID 블로그

여기에서 몇 가지 훌륭한 NetApp StorageGRID 블로그를 찾을 수 있습니다.

- 5월 10일: "Lab on Demand는 StorageGRID를 위한 최고의 영업 툴입니다"
- 5월 24일: "NetApp과 Alluxio로 분석 워크로드를 현대화하십시오"
- 5월 26일: "StorageGRID: 사내 백업 및 복제 데이터 저장 및 관리"
- 6월 9일: "StorageGRID와 함께 Cloudera Hadoop S3A 커넥터를 사용하십시오"
- 7월 26일: "StorageGRID의 검증된 파트너 솔루션 목록 을 확인하십시오"
- 8월 5일: "NetApp StorageGRID는 일반 조건 보안 인증을 획득했습니다"
- 8월 16일: "StorageGRID를 오픈 소스 엘크 스택과 통합하여 고객 경험을 개선합니다"
- 8월 17일: "이 모든 것은 개체 잠금에서 시작됩니다… 중요 백업 애플리케이션을 위한 S3 스토리지 에코시스템 구축"
- 8월 23일: "StorageGRID에서 데이터 레이크를 구축합니다"
- 9월 1일: "다음 메트릭을 사용하여 그래프로 표시합니다"
- 9월 19일: "StorageGRID용 DataLock 및 랜섬웨어 보호 지원"
- 9월 26일: "서비스 공급자를 위한 NetApp StorageGRID"
- 10월 5일: "스노우플레이크(Snowflake)용 StorageGRID에서 데이터를 해동하십시오"
- 10월 5일: "NetApp Cloud Insights에는 StorageGRID 갤러리 대시보드가 추가되었습니다"
- 11월 7일: "StorageGRID 및 ONTAP S3 지원: 차이점, 유사점 및 통합"
- 11월 23일: "NetApp과 Modzy가 제공하는 MLOps를 통한 설명 가능한 AI"
- 12월 6일: "StorageGRID는 KPMG 규정 준수 인증을 획득했습니다"
- 1월 16일: "StorageGRID는 NF203 및 ISO/IEC 25051 준수 인증을 갱신합니다"
- 1월 18일: "StorageGRID S3 오브젝트 잠금은 Veritas NetBackup에서 검증되었습니다"
- 2월 14일: "초콜릿, 스키, 시계, 메인프레임의 공통점은 무엇입니까?"
- 3월 14일: "3:2:1 호환 아키텍처에서 하나의 명령으로 Epic Systems EHR 데이터베이스를 백업하는 방법"
- 3월 30일: "BlueXP를 사용하여 3:2:1 호환 백업 정책으로 Epic EHR을 보호합니다"
- 3월 30일: "StorageGRID를 사용한 Amazon S3 알파 릴리스 마운트 지점"
- 5월 16일: "StorageGRID 오브젝트 스토리지 제품군의 새로운 기능"
- 5월 16일: "StorageGRID 11.7 및 새로운 All-Flash 오브젝트 스토리지 어플라이언스 SGF6112를 소개합니다"
- 8월 30일: "Amazon S3 파일 시스템의 마운트 지점이 이제 GA로 변경되었습니다"
- 9월 1일: "Cloud Insights를 활용하여 Fluent Bit을 사용하여 로그를 모니터링하고 수집합니다"
- 10월 17일: "하둡을 통한 전환: Dremio 및 StorageGRID로 데이터 분석 현대화"
- 11월 7일: "StorageGRID를 이용한 온프레미스 글레이셔 스펙트럼 로직"
- 12월 12일: "StorageGRID에 대한 빅 데이터 분석: Dremio는 Apache Hive보다 23배 빠른 성능을 제공합니다"

- 2월 2일: "StorageGRID+ lakeFS 솔루션 요약 발표"
- 2월 16일: "StorageGRID 11.8 소개: 향상된 보안, 단순성 및 사용자 환경"
- 2월 16일: "StorageGRID 11.8 소개"

NetApp StorageGRID 문서

각 NetApp StorageGRID 릴리스에 대한 전체 설명서는 여기에서 확인할 수 있습니다.

- "StorageGRID 어플라이언스"
- "StorageGRID 11.8"
- "StorageGRID 11.7"
- "StorageGRID 11.6"
- "StorageGRID 11.5"
- "StorageGRID 11.4"
- "StorageGRID 11.3"
- "StorageGRID 11.2"

법적 고지

법적 고지 사항은 저작권 선언, 상표, 특허 등에 대한 액세스를 제공합니다.

저작권

"https://www.netapp.com/company/legal/copyright/"

상표

NetApp, NetApp 로고, NetApp 상표 페이지에 나열된 마크는 NetApp Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.

"https://www.netapp.com/company/legal/trademarks/"

특허

NetApp 소유 특허 목록은 다음 사이트에서 확인할 수 있습니다.

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

개인 정보 보호 정책

"https://www.netapp.com/company/legal/privacy-policy/"

오픈 소스

통지 파일은 NetApp 소프트웨어에 사용된 타사의 저작권 및 라이센스에 대한 정보를 제공합니다.

https://library.netapp.com/ecm/ecm_download_file/2879263

https://library.netapp.com/ecm/ecm_download_file/2881511