



툴 및 애플리케이션 가이드

How to enable StorageGRID in your environment

NetApp
March 07, 2024

목차

툴 및 애플리케이션 가이드	1
StorageGRID와 함께 Cloudera Hadoop S3A 커넥터를 사용하십시오	1
S3cmd를 사용하여 StorageGRID에서 S3 액세스를 테스트하고 시연합니다	8
NetApp StorageGRID를 공동 스토리지로 사용하는 Vertica Eon 모드 데이터베이스	9
ELK 스택을 사용한 StorageGRID 로그 분석	22
Prometheus 및 Grafana를 사용하여 메트릭 보존 기간을 연장합니다	28
Datadog SNMP 구성	44
rclone을 사용하여 StorageGRID에서 개체를 마이그레이션, 저장 및 삭제합니다	47
Veeam 백업 및 복제를 사용한 구축에 대한 StorageGRID 모범 사례	59
StorageGRID를 사용하여 Dremio 데이터 소스를 구성합니다	70

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.

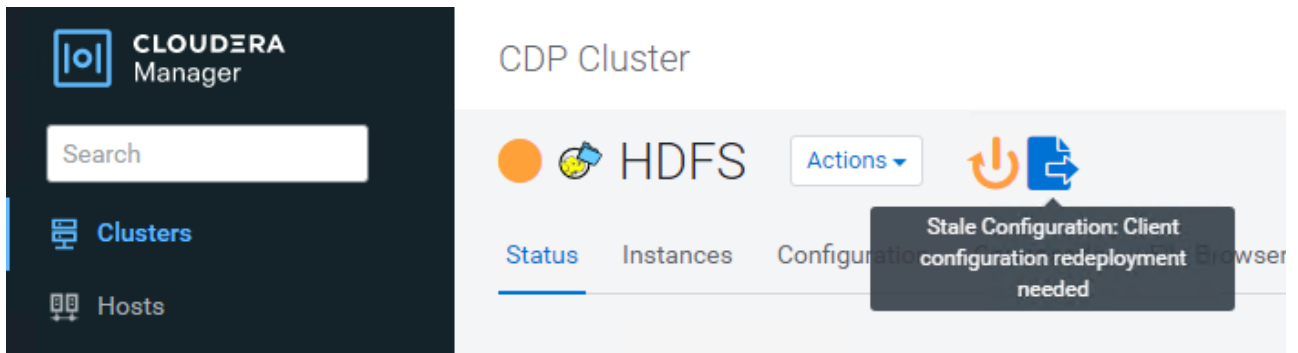
이름	값
fs.s3a.access.key	_<StorageGRID의 테넌트 S3 액세스 키> _
fs.s3a.secret.key	_<StorageGRID의 테넌트 S3 비밀 키> _
FS.s3a.CONNECTION.SSL.ENABLED	[true 또는 false] (이 항목이 누락된 경우 기본값은 https)
FS.s3a.endpoint	_<StorageGRID S3 엔드포인트: port> _
FS.s3a.IMPL	org.apache.hadoop.fs.s3a.s3aFileSystem

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

- 샘플 스크린샷 *

Name	fs.s3a.endpoint	 
Value	sgdemo.netapp.com:10443	
Description	StorageGRID s3 load balancer endpoint	
	<input checked="" type="checkbox"/> Final	
Name	fs.s3a.access.key	 
Value	OMC[REDACTED]BAN	
Description	SG CDP S3 access key	
	<input checked="" type="checkbox"/> Final	
Name	fs.s3a.secret.key	 
Value	mapz[REDACTED]Qfc	
Description	SG CDP S3 secret key	
	<input checked="" type="checkbox"/> Final	
Name	fs.s3a.impl	 
Value	org.apache.hadoop.fs.s3a.S3AFileSystem	
Description		
	<input checked="" type="checkbox"/> Final	
Name	fs.s3a.path.style.access	 
Value	true	
Description		
	<input checked="" type="checkbox"/> Final	

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 filesystem
s3a://hdfs-test/: org.apache.hadoop.fs.s3a.AWSClietIOException:
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 filesystem
s3a://hdfs-test/: org.apache.hadoop.fs.s3a.AWSClietIOException:
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.

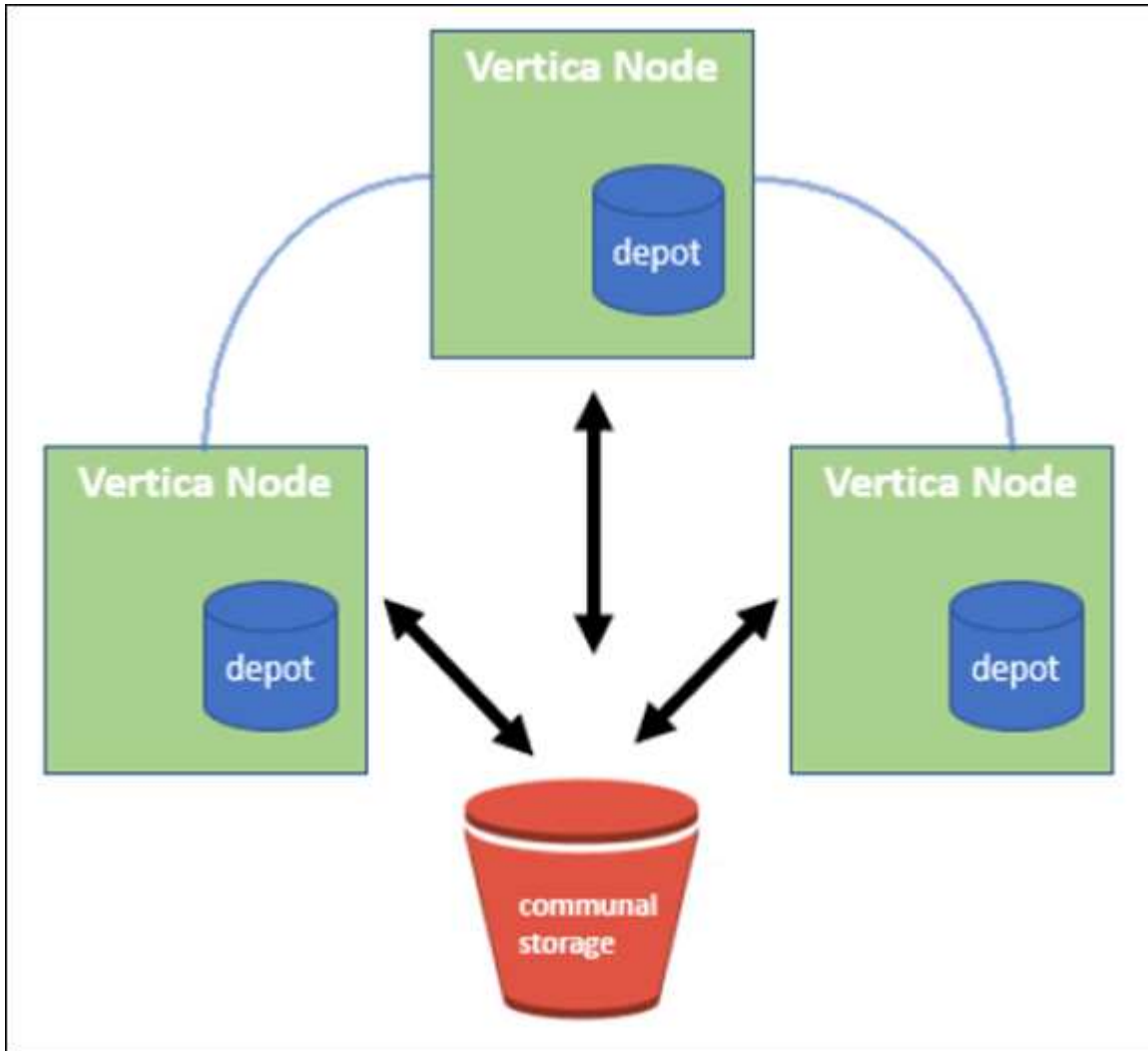
- eon Mode 데이터베이스는 공용 스토리지를 사용하여 데이터를 저장합니다. 이 방법은 Vertica에서 권장합니다.

- 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 건너뛴다. 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 사용자에게 읽기 권한이 있어야 합니다. StorageGRID S3 엔드포인트 SSL 인증서가 공개적으로 알려진 CA에 의해 서명된 경우 이 단계를 건너뛰십시오.

``awsconfigfile=<filepath/filename>'

예를 들어, 다음 샘플 파일을 참조하십시오.

```
awsauth = MNVU4OYFAY2xyz123:03vu04M4KmdfwffT8nqnBmnMVTr78Gu9wANabcxyz
awsendpoint = s3.England.connectlab.io:10443
awsenablehttps = 1
awsconfigfile = /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 매개 변수에 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 여카운트 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 버킷 이름>
- -s <이 데이터베이스에 사용할 Vertica 노드의 심표로 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/051/026d63a9d4a33237bf0e2c2a794a00a000021a07/026d63a9d4a33237bf0e2cf2a794a00a00021a07_0 DFS
145년	S3://vertica/2c4/026d63a9d4a33237bf0e2c2cf2a794a00a000021a3d/026d63ae9d4a33237bf0e2c2cf2a794a00a000021a3d_0 DFS
146	S3://vertica/33c/026d63a9d4a33237bf0e2c2cf2a794a00a000021a1d/026d63a9d4a33237bf0e2c2cf2a794a00a000021a1d_0 DFS
40세	S3://vertica/382/026d63a9d4a33237bf0e2c2a794a00a000021a31/026d63a9d4a33237bf0e2c2a794a00a000021a31_0 DFS
145년	S3://vertica/42F/026d63a9d4a33237bf0e2c2a794a00a000021a21/026d63ae9d4a33237bf0e2c2cf2a794a00a000021a21_0 DFS
34세	S3://vertica/472/026d63a9d4a33237bf0e2c2cf2a794a00a000021a25/026d63a9d4a33237bf0e2c2cf2a794a00a000021a25_0 DFS
41세	S3://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/026d63a9d4a33237bf0e2c2cf2a794a00a000021a15/026d63a9d4a33237bf0e2c2cf2a794a00a0000000021a15_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/Check points/C13/chkpt_1.cat.gz`
254년	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Check points/C13/Completed

오브젝트 크기(바이트)	버킷/객체 키 전체 경로
2958	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Check points/C2_2/chkpt_1.cat.gz`
231	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Check points/C2_2/Completed
822521	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Check points/C4_chkpt_1.cat.gz`
231	S 3://vertica/metadata/VMart/nodes/v_vmart_node0016/Catalog/859703b06a3456d95d0be28575a673/Check points/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/Check points/C14_7/chkpt_1.cat.gz`
232입니다	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/ Catalog/859703b06a3456d95d0be28575a673/Check points/C14_7/Completed
822930	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/ Catalog/859703b06a3456d95d0be28575a673/Txnlog s/txn_14_g7.cat.gz`
755033	S 3://vertica/metadata/VMart/nodes/v_vmart_node0017/ Catalog/859703b06a3456d95d0be28575a673/Txnlog s/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/Check points/C14_7/chkpt_1.cat.gz`
232입니다	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/ Catalog/859703b06a3456d95d0be28575a673/Check points/C14_7/Completed
822930	S 3://vertica/metadata/VMart/nodes/v_vmart_node0018/ Catalog/859703b06a3456d95d0be28575a673/Txnlog s/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 "vsq" 문을 사용합니다. 암호는 "온-프레미스 데이터베이스 만들기"에서 설정한 데이터베이스 암호입니다. 예를 들어, 다음 샘플 출력을 참조하십시오.

```
[dbadmin@vertica-vm1 ~]$ vsq
Password:
Welcome to vsq, the Vertica Analytic Database interactive terminal.
Type:  \h or \? for help with vsq 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.

```
vsq -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 * 5c670755742cfd5a723a596ba087e0153a65baef3934afdb682f61cd278d

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	yuh01IEBVP6KX4EwqcyU
...	 _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
```

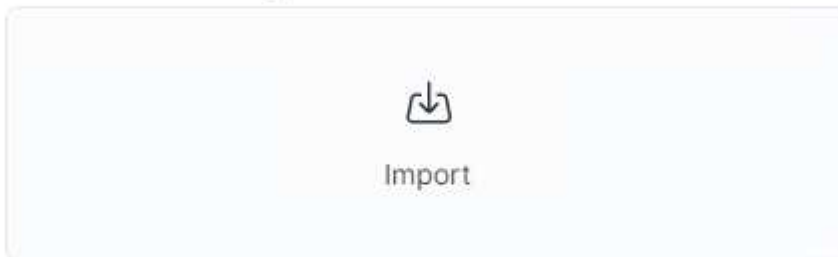
6. StorageGRID 관리자 GUI에서 로그 메시지를 Logstash로 보내도록 외부 syslog 서버를 구성합니다. 을 참조하십시오 "데모 비디오" 를 참조하십시오.
7. 정의된 TCP 포트에 대한 StorageGRID 노드 연결을 허용하려면 Logstash 서버에서 방화벽을 구성하거나 비활성화해야 합니다.
8. Kibana GUI에서 관리 → 개발 도구 를 선택합니다. 콘솔 페이지에서 이 가져오기 명령을 실행하여 Elasticsearch에 새 인덱스가 생성되었는지 확인합니다.

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

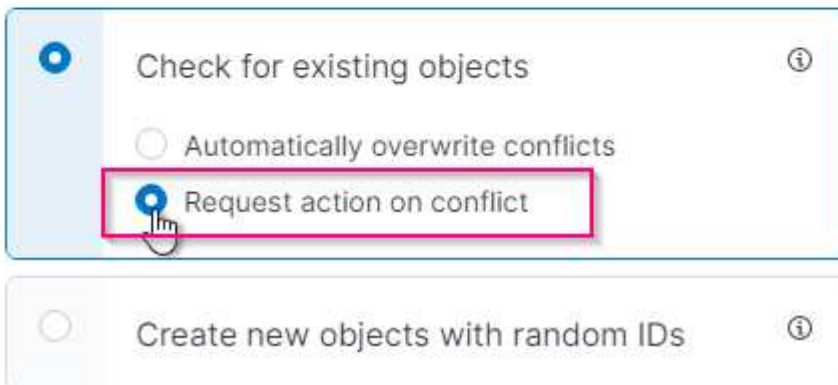
9. 키바나 GUI에서 인덱스 패턴(엘크 7.x) 또는 데이터 뷰(엘크 8.x)를 생성합니다.
10. Kibana GUI의 상단 중앙에 있는 검색 상자에 '저장된 개체'를 입력합니다. + 저장된 개체 페이지에서 가져오기를 선택합니다. 불러오기 옵션 아래에서 '충돌 시 작업 요청'을 선택합니다.

Import saved objects ✕

Select a file to import



Import options



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

Import saved objects ×

Data Views Conflicts

The following saved objects use data views that do not exist. Please select the data views you'd like re-associated with them. You can [create a new data view](#) if necessary.

ID	Count	Sample of aff...	New data view
594f91a0-d192-11ec-b30f-09f67aedd1d9	2		<div style="border: 2px solid #d93025; padding: 5px; display: inline-block;"> sglog ▾ </div>
60cf3620-e5fa-11ec-af71-8f6e980d6eb0	1		<div style="border: 2px solid #d93025; padding: 5px; display: inline-block;"> sglog ▾ </div>

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 스크래핑하도록 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 StorageGRID를 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.

Add a client certificate

1

Enter details

2

Enter details

Certificate details

Certificate name [?](#)

prometheus

Permissions

Allow prometheus [?](#)

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.

Add a client certificate ×

Enter details ————— 2 Enter details

Certificate type

Upload certificate Generate certificate

Domain name ⓘ

prometheus.grid.local

[Add another domain](#)

IP ⓘ

192.168.0.10

[Add another IP address](#)

Subject ⓘ

/CN=Prometheus

Days valid ⓘ

730

[Previous](#)



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.

Generate

Certificate details

Download certificate Copy certificate PEM

Subject DN: /CN=Prometheus
 Serial Number: 72:D9:6E:D7:04:CC:4F:29:66:0A:CA:53:24:79:1B:09:49:3A:BC:56
 Issuer DN: /CN=Prometheus
 Issued On: 2022-08-22T17:54:33.000Z
 Expires On: 2024-08-21T17:54:33.000Z
 SHA-1 Fingerprint: 10:47:6E:FD:67:D8:53:E7:6E:E5:D8:8A:DF:BD:45:94:04:53:47:1E
 SHA-256 Fingerprint: 74:23:C2:02:3A:D9:08:C0:EE:C1:F8:59:8A:7C:AE:18:AB:80:7D:21:31:F3:EB:AF:BF:4F:9E:C7:90:C9:FA:E7
 Alternative Names: DNS:prometheus.grid.local
 IP Address:192.168.0.10

Certificate private key

⚠ You will not be able to view the certificate private key after you close this dialog. To save the keys for future reference, copy and paste the values to another location.

Download private key Copy private key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA3bIcyIEpMWPk5ritVpMkmIDKLIjaTM3ertq23VcAALwxziaU
...
```



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/fstab에 항목을 추가하면 /dev/sdb의 uuid를 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.pem`을 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
```

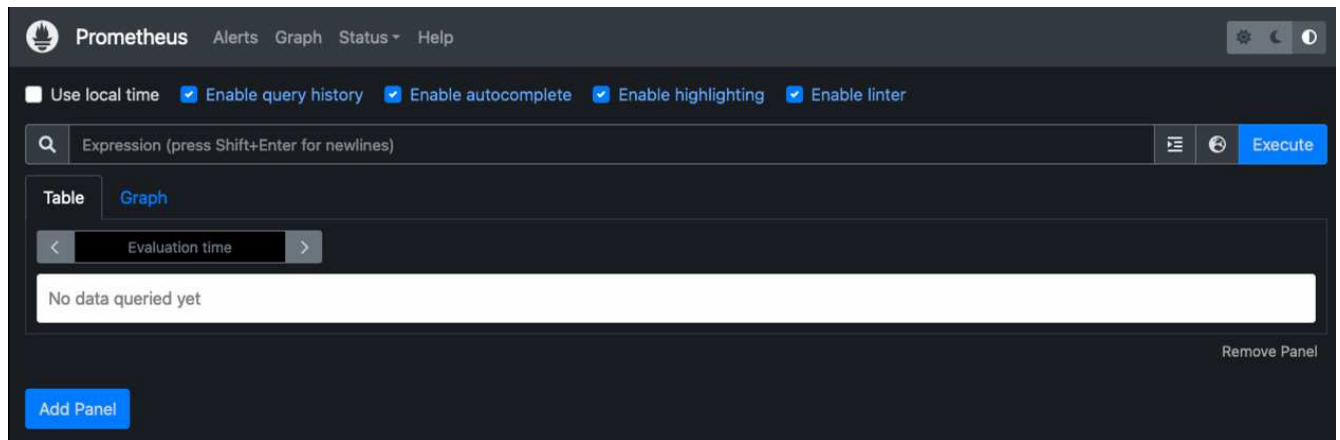
```

● 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>

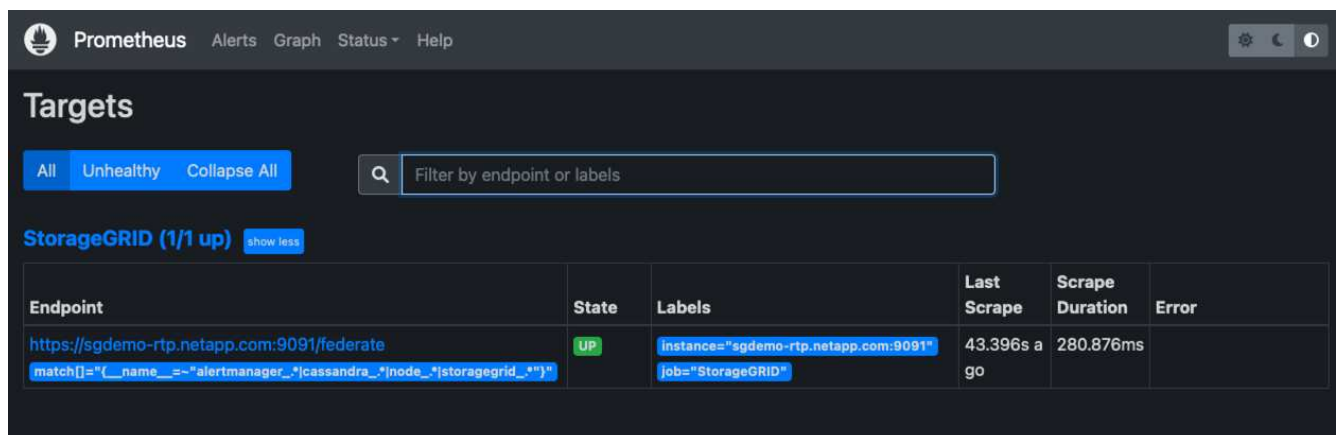
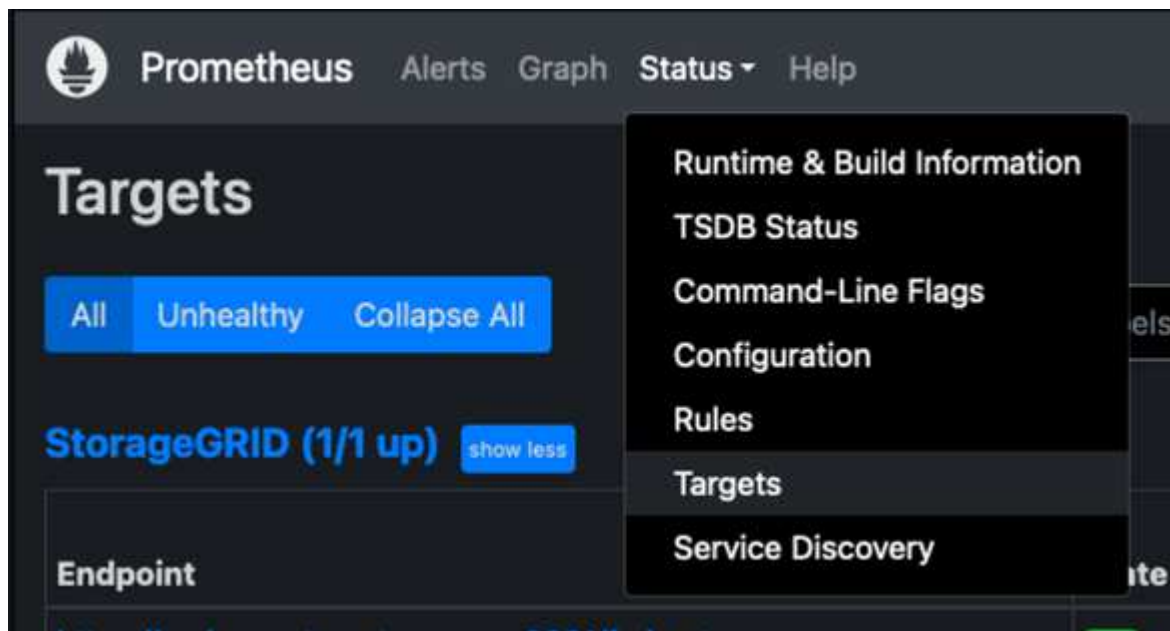
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.510Z caller=head.go:544 level=info component=tsdb
msg="Replaying WAL, this may take a while"
Aug 22 15:14:24 aj-deb-prom01 prometheus[6498]: ts=2022-08-
22T19:14:24.816Z caller=head.go:615 level=info component=tsdb msg="WAL
segment 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..."

```

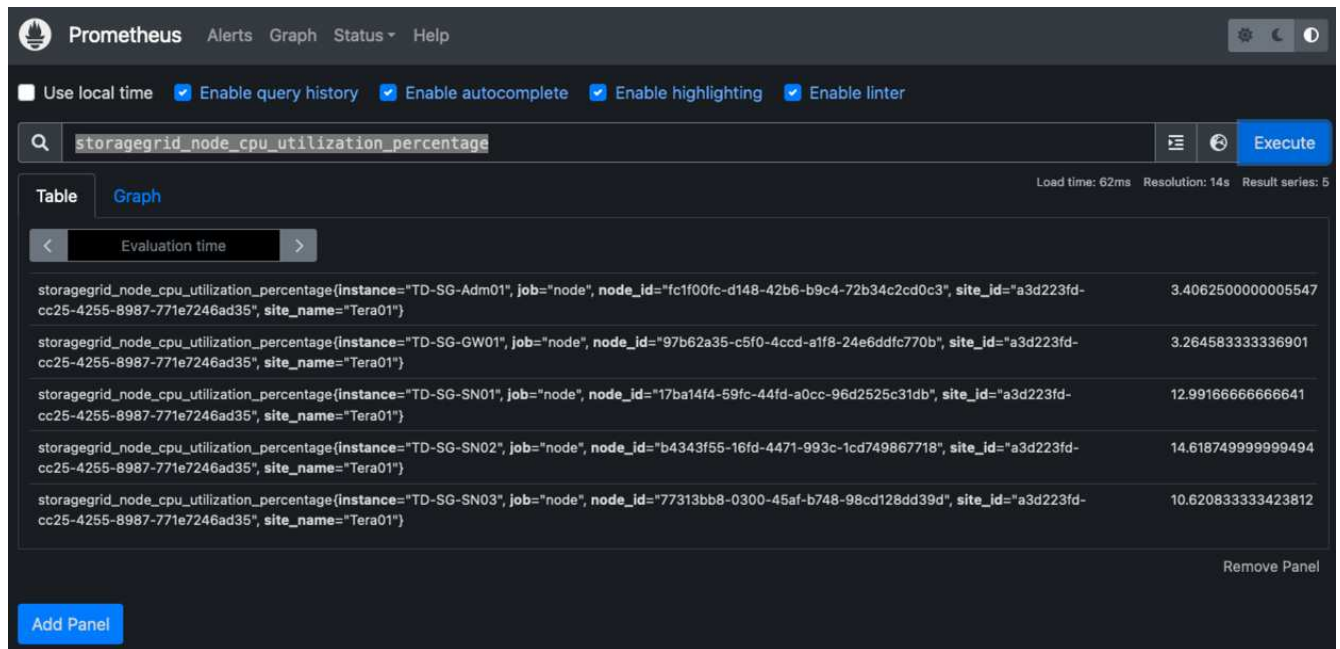
6. 이제 Prometheus 서버의 UI로 이동할 수 있습니다 <http://Prometheus-server:9090> UI를 참조하십시오



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



8. 그래프 페이지에서 테스트 쿼리를 실행하고 데이터가 스크래핑되었는지 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 인스턴스

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.

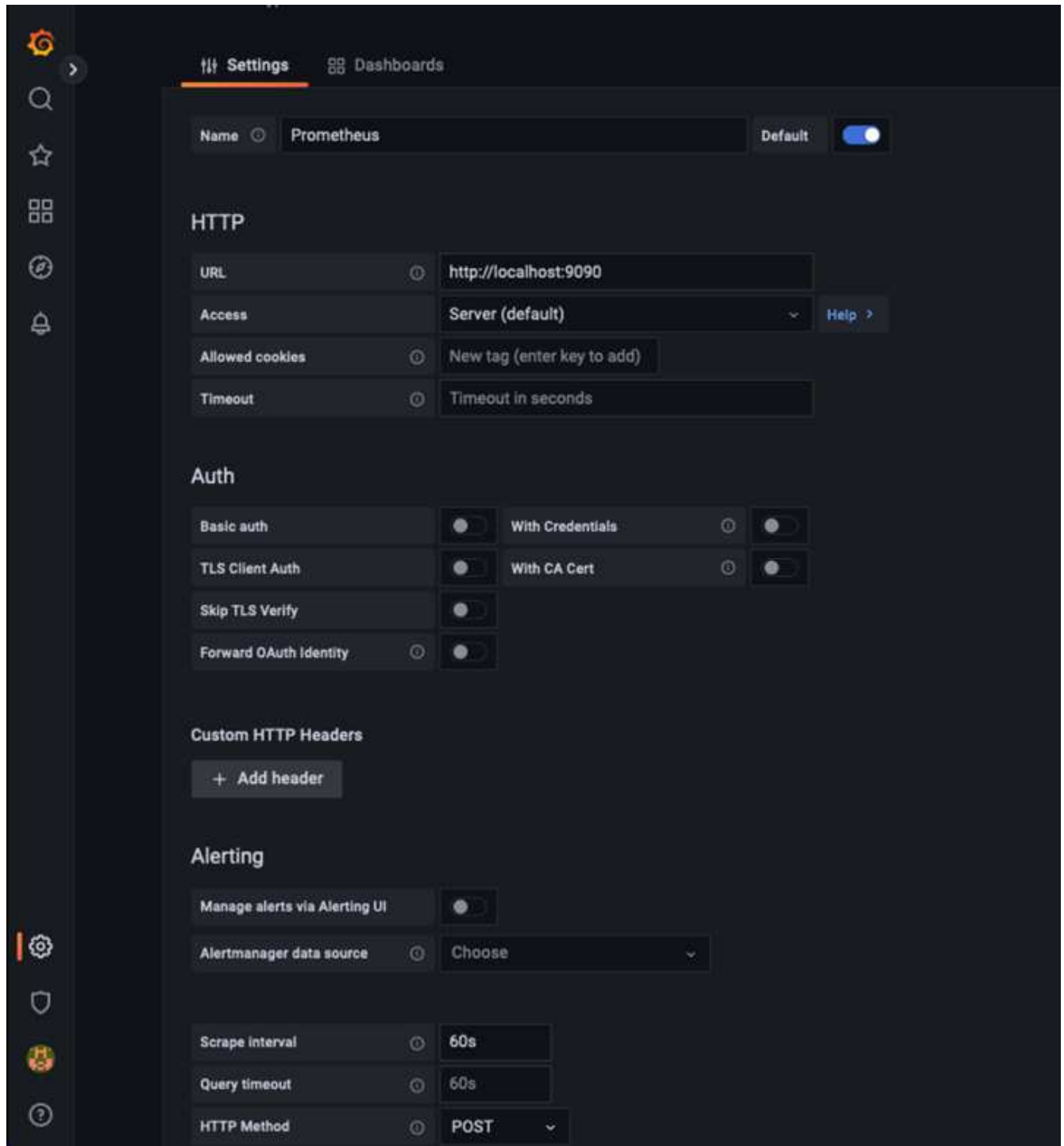
```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl enable grafana-server.service
```

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.

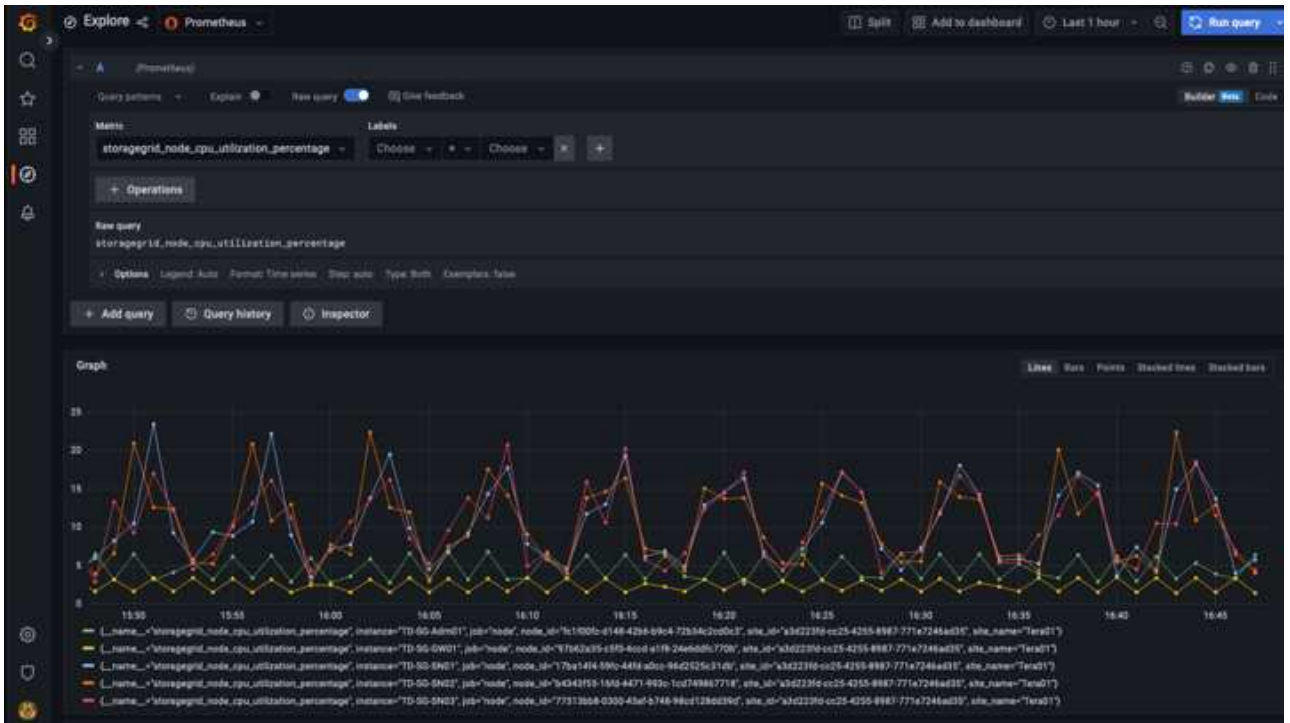
StorageGRID에 대한 Grafana 대시보드를 생성합니다

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 스크래핑 간격을 입력하여 Prometheus 소스를 구성합니다. Prometheus에서 경고 관리자를 구성하지 않았기 때문에 알림 섹션도 비활성화했습니다.

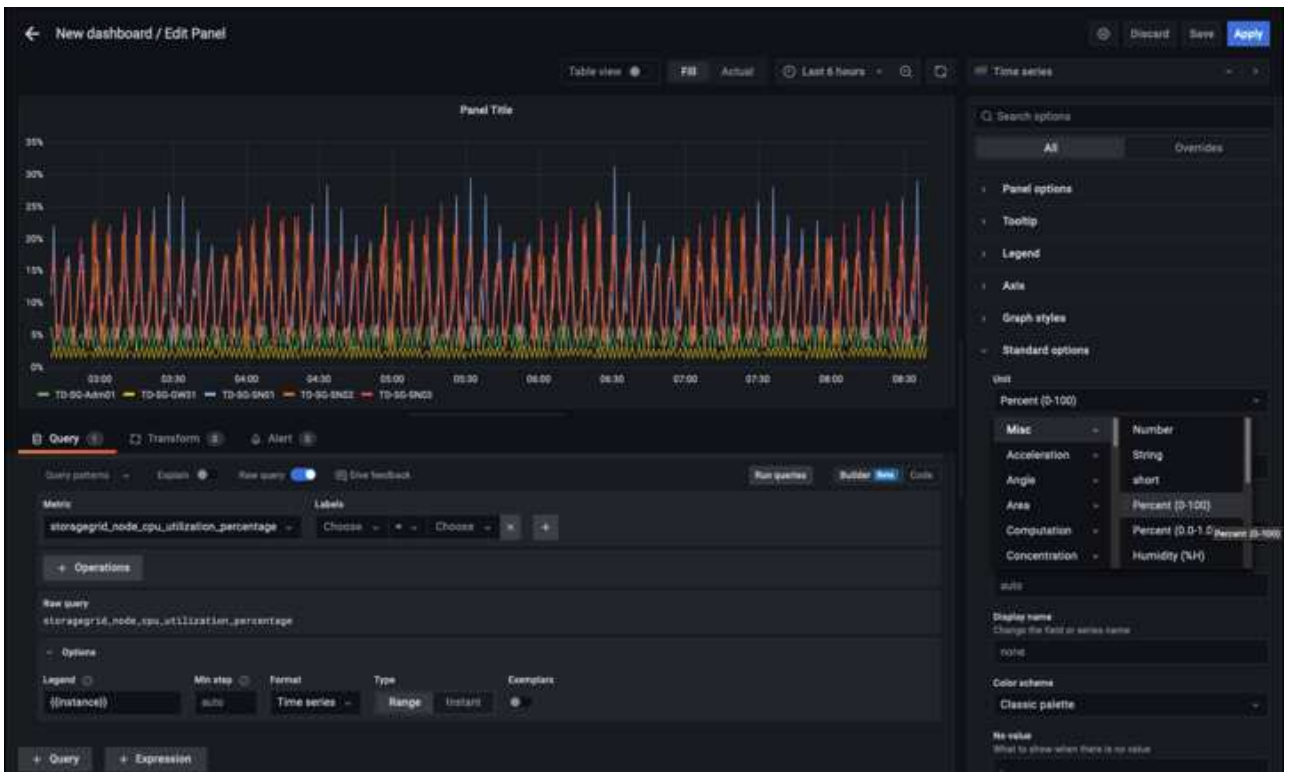


4. 원하는 설정을 입력한 후 아래로 스크롤하여 "Save & Test(저장 및 테스트)"를 클릭합니다.
5. 구성 테스트가 완료되면 탐색 버튼을 클릭합니다.
 - a. 탐색 창에서 Prometheus를 "StorageGrid_node_cpu_Utilization_percentage"로 테스트한 것과 동일한 메트릭을 사용하고 "쿼리 실행" 단추를 클릭할 수 있습니다



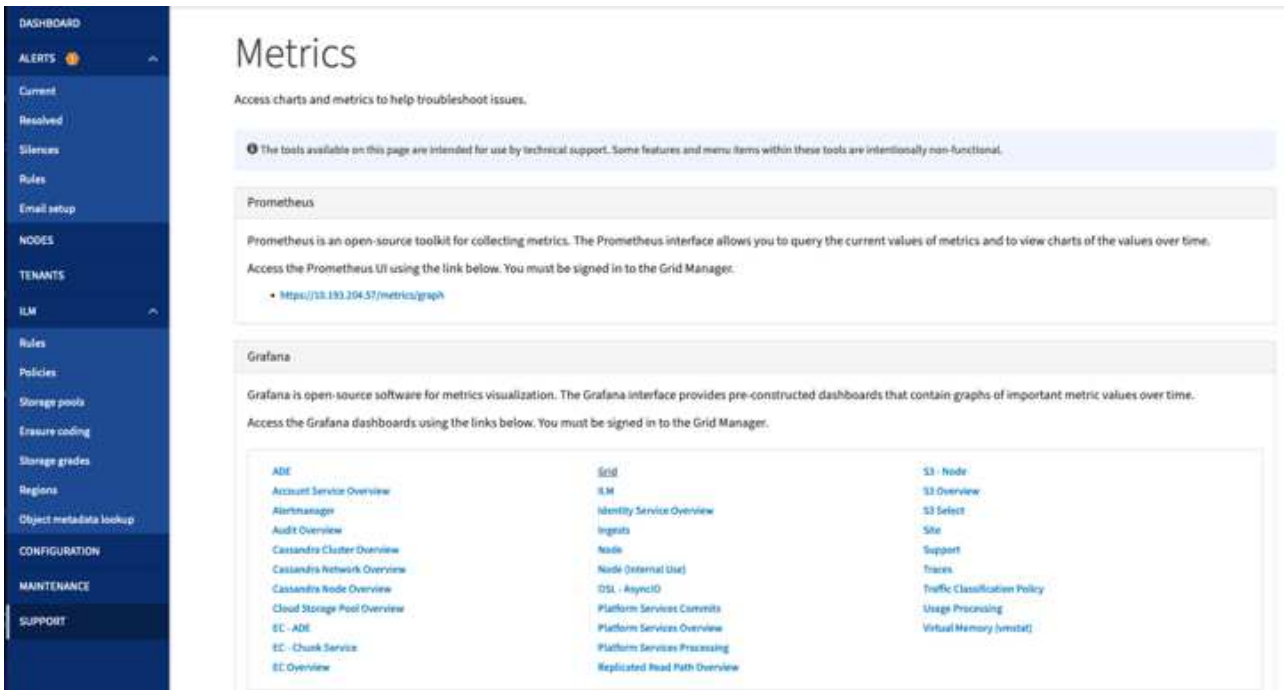
6. 이제 데이터 소스가 구성되었으므로 대시보드를 생성할 수 있습니다.

- 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.

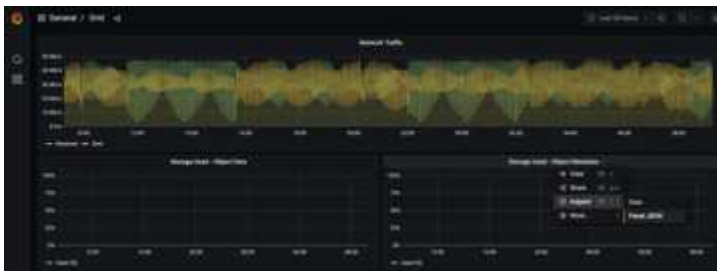


7. 원하는 각 메트릭에 대해 이러한 대시보드를 계속 구축할 수 있지만 다행히 StorageGRID에는 사용자 지정 대시보드에 복사할 수 있는 패널이 포함된 대시보드가 이미 있습니다.

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

Inspect: Storage Used - Object Metadata

4 queries with total query time of 549 ms

Data

Stats

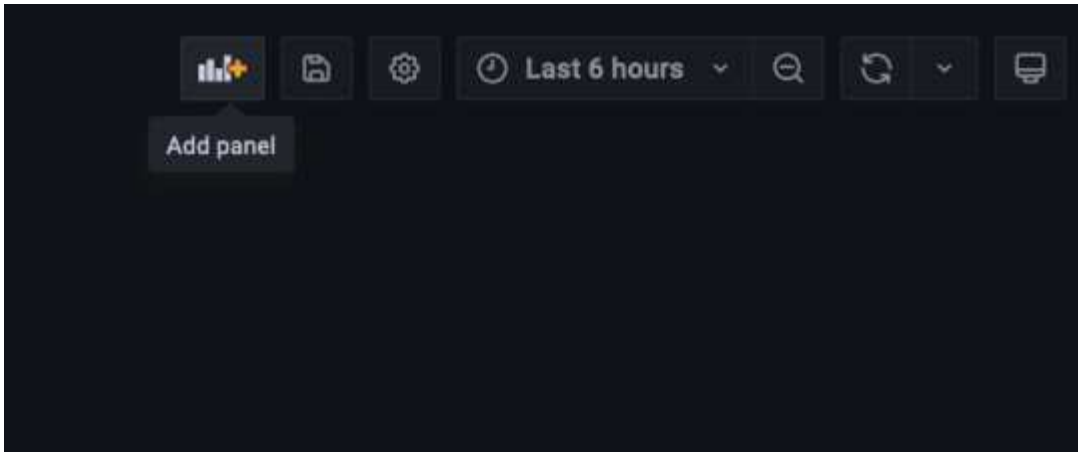
JSON

Select source

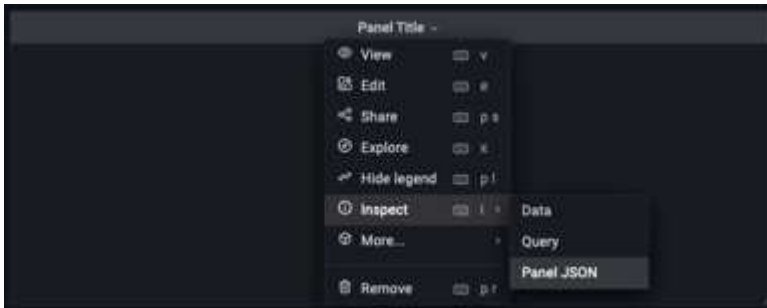
Panel JSON

```
1  [
2  "aliasColors": {},
3  "bars": false,
4  "dashLength": 10,
5  "dashes": false,
6  "datasource": "Prometheus",
7  "decimals": 2,
8  "fill": 1,
9  "fillGradient": 0,
10 "gridPos": {
11   "h": 7,
12   "w": 12,
13   "x": 12,
14   "y": 7
15 },
16 "id": 6,
17 "legend": {
18   "avg": false,
19   "current": false,
20   "max": false,
21   "min": false,
22   "show": true,
23   "total": false,
24   "values": false
25 },
26 "lines": true,
27 "linewidth": 1,
28 "links": [],
29 "nullPointMode": "null",
30 "options": {
31   "alertThreshold": true
32 },
33 "percentage": false,
34 "pointradius": 5,
35 "points": false,
36 "renderer": "flot",
37 "seriesOverrides": [
38   {
39     "alias": "Used",
```

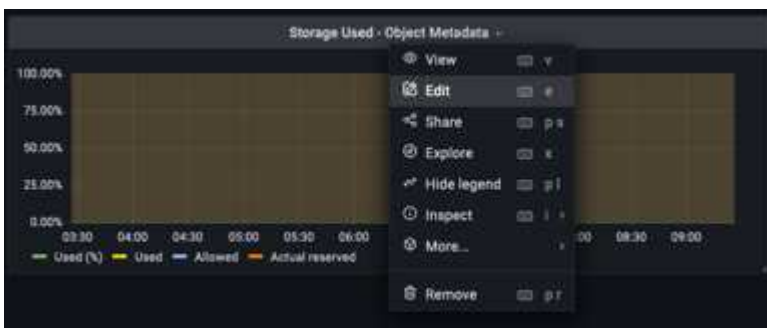
e. 새 대시보드에서 아이콘을 클릭하여 새 패널을 추가합니다.

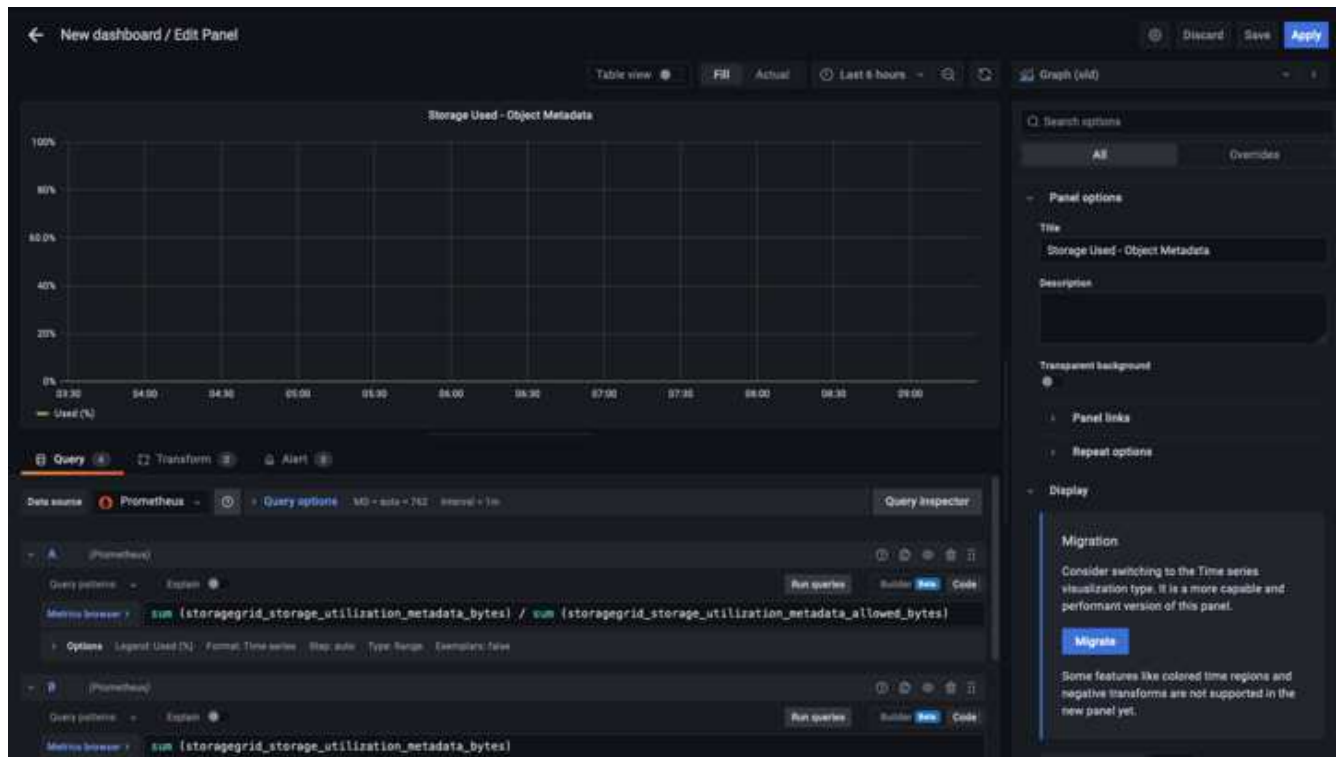


- f. 변경하지 않고 새 패널을 적용합니다
- g. StorageGRID 패널과 마찬가지로 JSON을 검사하십시오. JSON 코드를 모두 제거하고 StorageGRID 패널에서 복사한 코드로 교체합니다.



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





- 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 독 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.YAML`에 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.

SNMP Agent

You can configure SNMP for read-only MIB access and notifications. SNMPv1, SNMPv2c, SNMPv3 are supported. For SNMPv3, only User Security Model (USM) authentication is supported. All nodes in the grid share the same SNMP configuration.

Enable SNMP

System Contact

System Location

lab

Enable SNMP Agent Notifications

Enable Authentication Traps

Community Strings

Default Trap Community

st0r@gegrid

Read-Only Community

String 1

st0r@gegrid

+

Other Configurations

Agent Addresses (0)

USM Users (0)

Trap Destinations (1)

+ Create

Edit

X Remove

Version	Type	Host	Port	Protocol	Community/USM User
<input type="radio"/> SNMPv2C	Inform	10.193.92.241	9162	UDP	Default Community: st0r@gegrid

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"](https://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.

- * 버킷과 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 StorageGRID와 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 Repository](#)를 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, Loadbalancer.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.

Configuration > S3 Object Lock

S3 Object Lock

i S3 Object Lock has been enabled for the grid and cannot be disabled.

Enable S3 Object Lock for your entire StorageGRID system if S3 tenant accounts need to satisfy regulatory compliance requirements when saving object data. After this setting is enabled, it cannot be disabled.

Before enabling S3 Object Lock, you must ensure that the default rule in the active ILM policy is compliant. A compliant rule satisfies the requirements of buckets with S3 Object Lock enabled.

- It must create at least two replicated object copies or one erasure-coded copy.
- These copies must exist on Storage Nodes for the entire duration of each line in the placement instructions.
- Object copies cannot be saved in a Cloud Storage Pool.
- Object copies cannot be saved on Archive Nodes.
- At least one line of the placement instructions must start at day 0, using Ingest Time as the reference time.
- At least one line of the placement instructions must be "forever".

Enable S3 Object Lock


Apply

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.

Manage object settings Optional

Object versioning

Enable object versioning if you want to store every version of each object in this bucket. You can then retrieve previous versions of an object as needed.

 Object versioning has been enabled automatically because this bucket has S3 Object Lock enabled.

Enable object versioning

S3 Object Lock

S3 Object Lock allows you to specify retention and legal hold settings for the objects ingested into a bucket. If you want to use S3 Object Lock, you must enable this setting when you create the bucket. You cannot add or disable S3 Object Lock after a bucket is created.

If S3 Object Lock is enabled, object versioning is enabled for the bucket automatically and cannot be suspended.

Enable S3 Object Lock

Default retention 

Automatically protect new objects put into this bucket from being deleted or overwritten.

Disable

Enable

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

Buckets > veeam12

veeam12

Region: us-east-1
 S3 Object Lock: Enabled
 Date created: 2023-09-21 08:01:38 GMT
 Object count: 0

[View bucket contents in Experimental S3 Console](#)

[Delete objects in bucket](#) [Delete bucket](#)

Bucket options | [Bucket access](#) | [Platform services](#)

Consistency level	Read-after-new-write (default)	▼
Last access time updates	Disabled	▼
Object versioning	Enabled	▼
S3 Object Lock	Enabled	▼

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.

Bucket options | [Bucket access](#) | [Platform services](#)

Consistency level Read-after-new-write (default) ▲

Change the consistency control for operations performed on the objects in the bucket. Consistency levels provide a balance between the availability of objects and the consistency of those objects across different Storage Nodes and sites.

In general, use the **Read-after-new-write** consistency level for your buckets. Then, if objects do not meet availability or consistency requirements, change the client application's behavior, or set the Consistency-Control header for an individual API request, which overrides the bucket setting.

- All
Provides the highest guarantee of consistency. All nodes receive the data immediately, or the request will fail.
- Strong-global
Guarantees read-after-write consistency for all client requests across all sites.
- Strong-site
Guarantees read-after-write consistency for all client requests within a site.
- Read-after-new-write (default)
Provides read-after-write consistency for new objects and eventual consistency for object updates. Offers high availability and data protection guarantees. Recommended for most cases.
- Available
Provides eventual consistency for both new objects and object updates. For S3 buckets, use only as required (for example, for a bucket that contains log values that are rarely read, or for HEAD or GET operations on keys that do not exist). Not supported for FabricPool buckets.

[Save changes](#)

Last access time updates Disabled ▼

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.

Create a traffic classification policy

You can create traffic classification policies to monitor the network traffic for specific buckets, tenants, IP addresses, subnets, or load balancer endpoints. You can optionally limit this traffic based on bandwidth, number of concurrent requests, or the request rate.

① Enter policy name — ② Add matching rules — ③ Set limits — ④ Review the policy

Review the policy

Policy name: Veeam

Description: Policy to monitor Veeam bucket traffic

Matching rules

Type ?	Match value ?	Inverse match ?
Bucket	test	No

Veeam을 선택합니다

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

New Object Storage Repository X

Name
Type in a name and description for this object storage repository.

Name:
Object storage repository 1

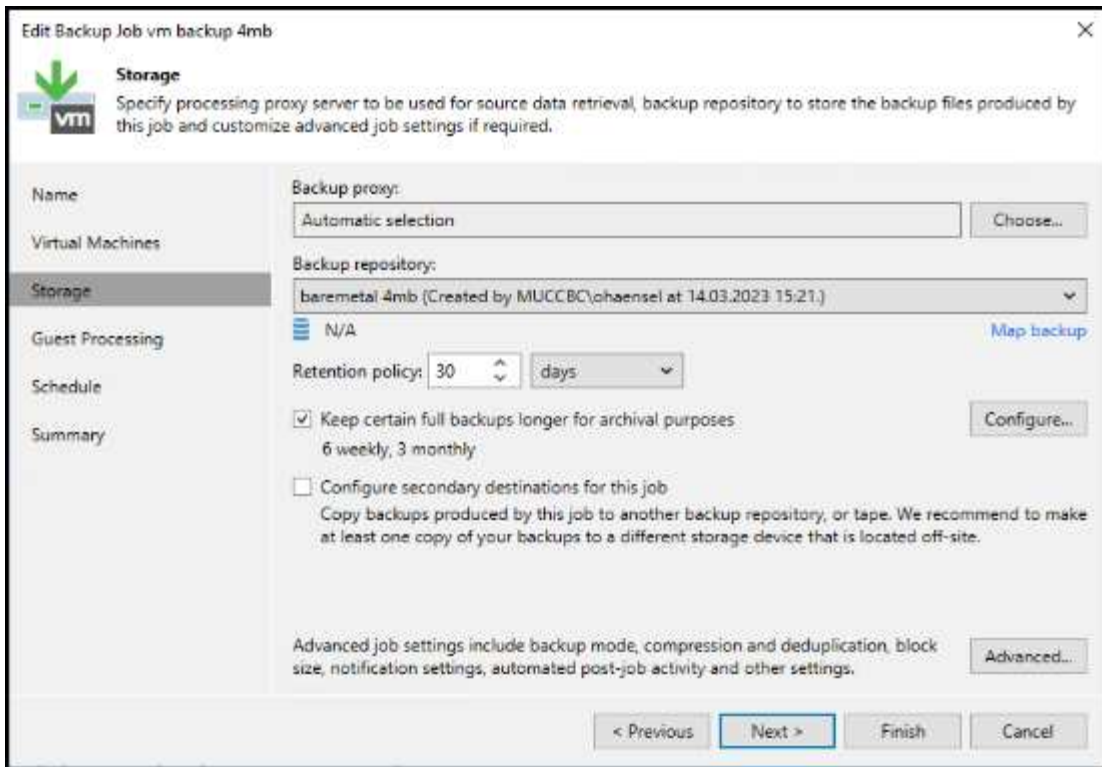
Description:
Created by SRV92\Administrator at 2/3/2021 8:15 AM.

Limit concurrent tasks to: 2 ▲▼

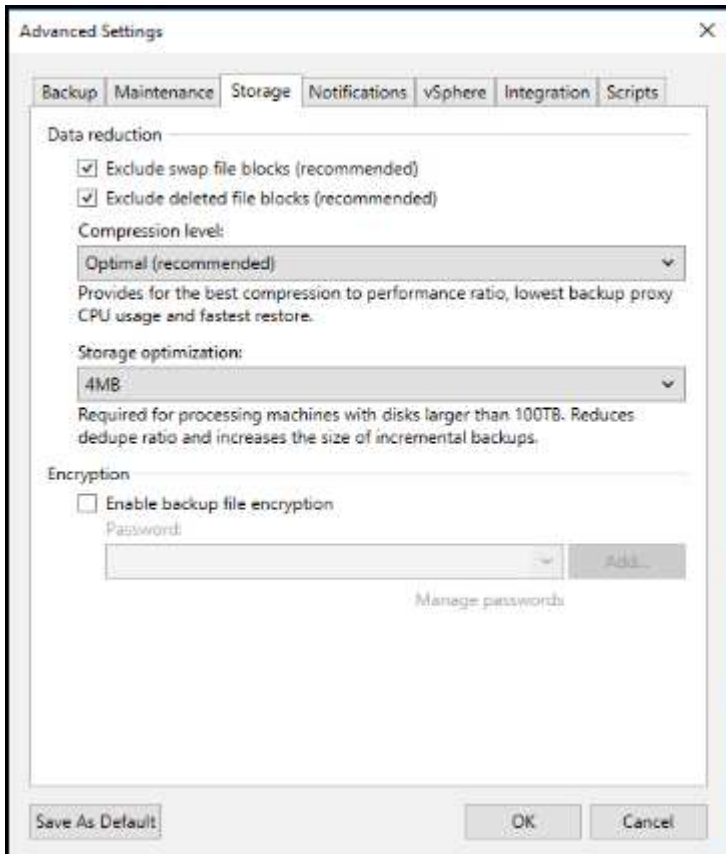
Use this setting to limit the maximum number of tasks that can be processed concurrently in cases when your object storage is overloaded or cannot keep up with the number of API requests issued by multiple object storage offload tasks.

< Previous
 Next >
 Finish
 Cancel

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. 요구 사항에 따라 게스트 설정을 변경하고 백업 작업 일정을 구성합니다.



StorageGRID 모니터링

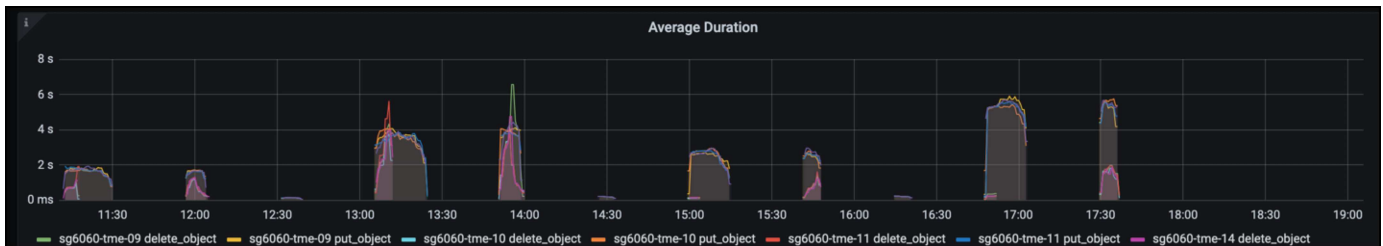
Veeam과 StorageGRID가 함께 작동하는 방식을 자세히 보려면 첫 번째 백업의 보존 시간이 만료될 때까지 기다려야 합니다. 지금까지는 Veeam 워크로드가 주로 PUT 작업으로 구성되며 삭제가 발생하지 않습니다. 백업 데이터가 만료되고 정리가 시작되면 오브젝트 저장소에서 전체 일관된 사용량을 확인하고 필요한 경우 Veeam에서 설정을 조정할 수 있습니다.

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.

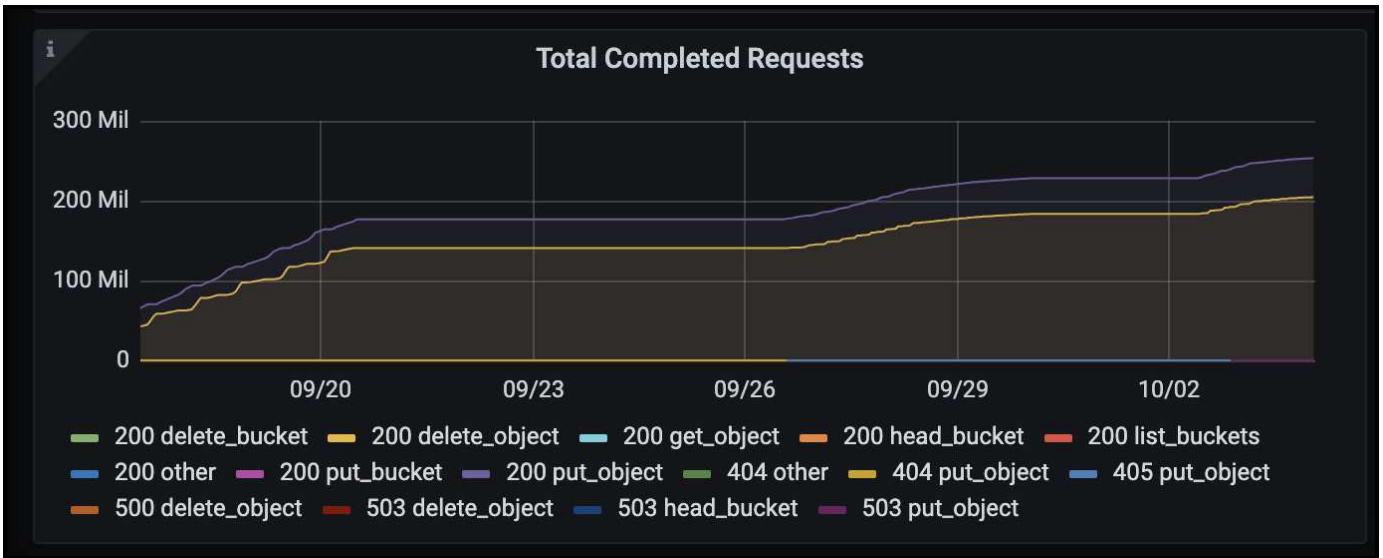
S3 속도 및 활성 요청을 보면 각 노드가 처리 중인 로드와 유형별로 전체 요청 수를 확인할 수 있습니다.



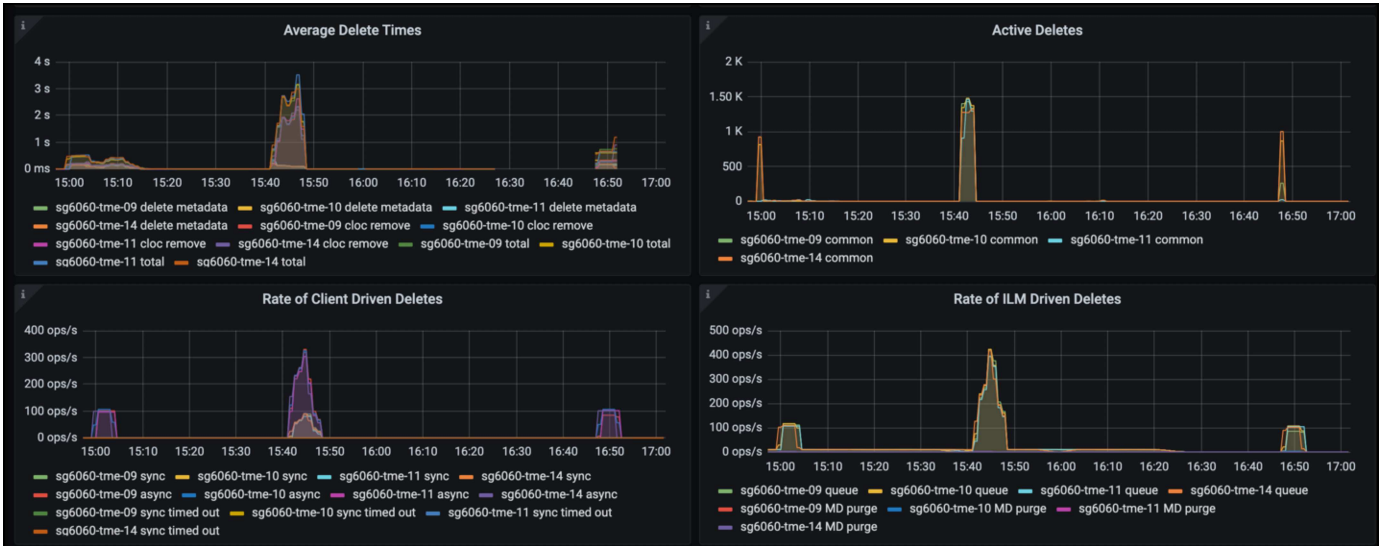
Average Duration(평균 기간) 차트에는 각 노드가 각 요청 유형에 대해 걸리는 평균 시간이 표시됩니다. 이는 요청의 평균 대기 시간이며 추가 튜닝이 필요하거나 StorageGRID 시스템이 더 많은 로드를 처리할 수 있는 공간이 있음을 나타내는 좋은 지표가 될 수 있습니다.



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 *

The screenshot shows the configuration interface for an Amazon S3 Source in Dremio. The 'General' tab is active. The 'Name' field is set to 'parquet-1tb'. In the 'Authentication' section, 'AWS Access Key' is selected. The 'AWS Access Key' field contains 'AKIAIOSFODNN7EXAMPLE' and the 'AWS Access Secret' field contains 'wJalrXU3FJOQJ4dQrThjVz6zS6pUwZbywcq'. The 'IAM Role to Assume' field is empty. The 'Encrypt connection' checkbox is checked. In the 'Public Buckets' section, there are no buckets added and an 'Add bucket' button is visible.

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 *

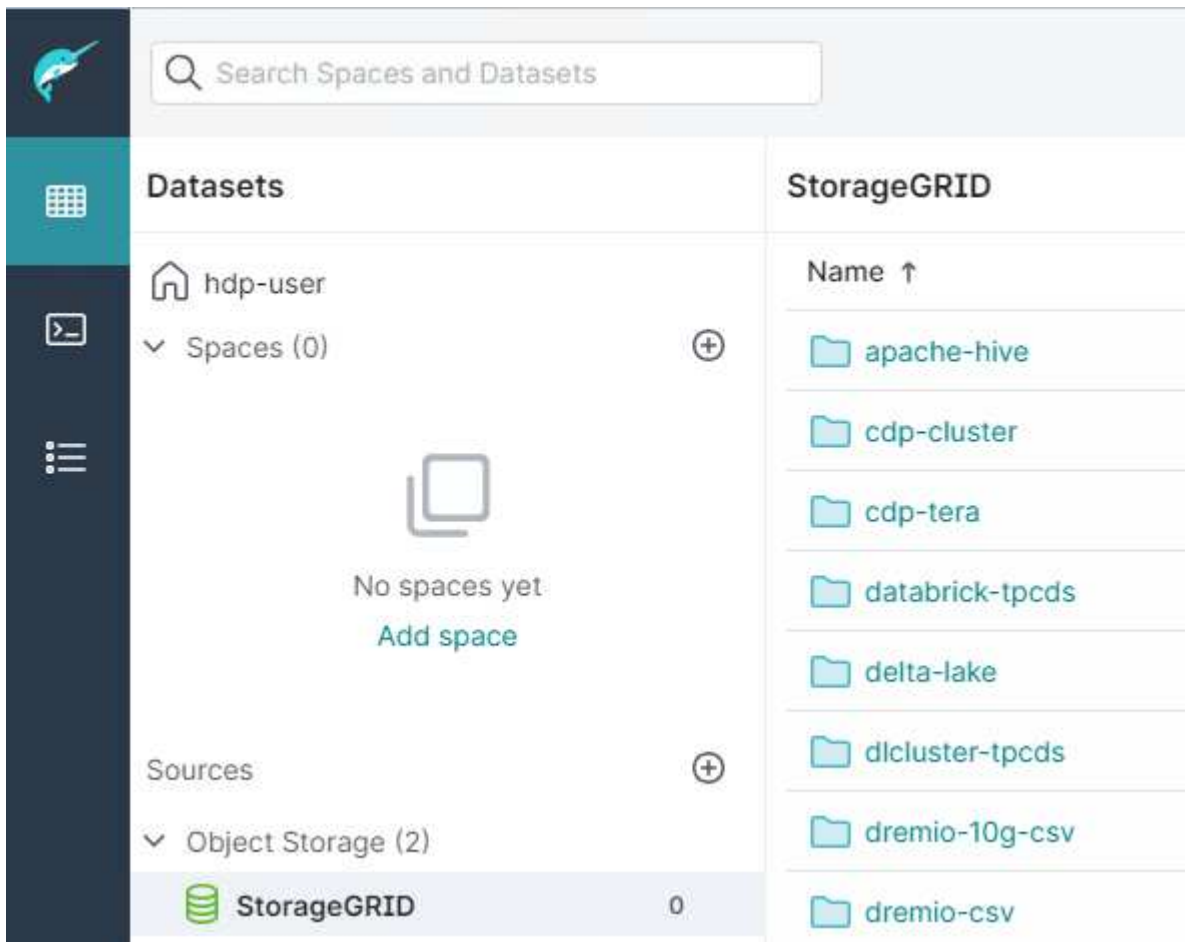
The screenshot shows the Dremio configuration interface for FS.s3a.connection. The 'Advanced Options' tab is selected. The configuration includes several checkboxes and input fields:

- Enable asynchronous access when possible
- Enable compatibility mode
- Apply requester-pays to S3 requests
- Enable file status check
- Enable partition column inference
- Root Path:
- Server side encryption key ARN:
- Default CTAS Format:
- Connection Properties:

Name	Value
fs.s3a.path.style.access	true
fs.s3a.endpoint	sgdemo.netapp.com
fs.s3a.connection.maximum	1000
- Allowedlisted buckets: No allowlisted buckets added
- Cache Options:
 - Enable local caching when possible
 - Max percent of total available cache space to use when possible:

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

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.