Google Cloud NetApp Volumes

Trident

NetApp
November 02, 2026

This PDF was generated from https://docs.netapp.com/ko-kr/trident-2602/trident-use/gcnv.html on November 02, 2026. Always check docs.netapp.com for the latest.

목차

Google Cloud NetApp Volumes . . . 1
- Google Cloud NetApp Volumes 구성 . . . 1
  - 개요 . . . 1
  - 구성 준비 . . . 1
  - NAS 스토리지 구성 . . . 2
- SAN 워크로드를 위해 Google Cloud NetApp Volumes 구성 . . . 5
  - 개요 . . . 5
  - Flex Unified 스토리지 풀 . . . 5
  - Trident SAN 백엔드 구성 . . . 5
  - StorageClass 생성 . . . 6
  - 블록 볼륨 프로비저닝 . . . 7
  - 블록 볼륨 동작 . . . 8
  - 액세스 모드 . . . 8
  - volumeMode 동작 . . . 9
  - 지원되는 작업 . . . 9
  - 추가 GiB 오버프로비저닝 동작 . . . 9
  - Pod 예 . . . 10
  - 연결 및 마운트 동작 . . . 10
- Google Cloud NetApp Volumes 백엔드 구성을 준비하세요 . . . 11
  - NFS 또는 SMB 볼륨의 사전 요구 사항 . . . 11
- Google Cloud NetApp Volumes 백엔드 구성 옵션 및 예시 . . . 11
  - 백엔드 configuration 옵션 . . . 11
  - 볼륨 프로비저닝 옵션 . . . 13
  - 예시 구성 . . . 13
  - 다음 단계 . . . 18
  - 스토리지 클래스 정의 . . . 19
- Google Cloud NetApp Volumes에 대한 자동 계층화 구성 . . . 21
  - 개요 . . . 22
  - 개념 . . . 22
  - 구성 모델 . . . 22
  - Trident 26.02에서 지원되는 기능 . . . 23
  - Trident 26.02에서 지원되지 않는 기능 . . . 23
  - 백엔드 구성 매개 변수 . . . 23
  - PersistentVolumeClaim 어노테이션을 사용한 볼륨 수준 재정의 . . . 23
  - 동작 및 제한 사항 . . . 24

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes 구성

Google Cloud NetApp Volumes를 Trident의 백엔드로 구성하여 Kubernetes 워크로드용 스토리지를 프로비저닝할 수 있습니다.

개요

Trident는 NAS(NFS 및 SMB)와 블록(iSCSI) 워크로드 모두에 대해 Google Cloud NetApp Volumes를 지원합니다.

- NAS 워크로드는 `google-cloud-netapp-volumes` 백엔드를 사용합니다
- 블록(iSCSI) 워크로드는 `google-cloud-netapp-volumes-san` 백엔드를 사용합니다.

NAS 볼륨은 파일 기반 스토리지를 제공하며 NFS 또는 SMB 프로토콜을 사용하여 액세스합니다. 이러한 볼륨은 여러 파드 또는 노드에서 공유 액세스를 지원합니다.

블록 볼륨은 원시 블록 스토리지를 제공하며 Kubernetes 노드에 연결된 iSCSI 장치로 액세스됩니다. 이러한 볼륨은 애플리케이션에 블록 수준 액세스가 필요한 경우에 사용됩니다.

이는 다음 환경에 적용됩니다.

- Trident 26.02 이상
- Google Kubernetes Engine(GKE) 또는 Red Hat OpenShift
- Google Cloud NetApp Volumes 스토리지 풀

블록(iSCSI) 스토리지를 구성하려면 "블록 스토리지(iSCSI) 구성"을(를) 참조하십시오.

구성 준비

Cloud ID를 사용하면 Kubernetes 워크로드가 정적 자격 증명을 사용하는 대신 워크로드 ID로 인증하여 Google Cloud 리소스에 액세스할 수 있습니다.

Google Cloud NetApp Volumes에서 클라우드 ID를 사용하려면 다음이 필요합니다.

- Google Kubernetes Engine(GKE)을 사용하여 배포된 Kubernetes 클러스터
- GKE 클러스터에서 워크로드 ID가 활성화되고 노드 풀에서 메타데이터 서버가 활성화되었습니다
- Google Cloud NetApp Volumes 관리자 역할(`roles/netapp.admin`이 있는 Google Cloud 서비스 계정 또는 이에 상응하는 사용자 지정 역할
- Trident가 클라우드 공급자를 `GCP`로 설정하고 클라우드 ID 주석을 구성한 상태로 설치됨

Trident 운영자

Trident 오퍼레이터를 사용하여 Trident를 설치하려면 다음을 편집하십시오 `tridentorchestrator_cr.yaml`:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  namespace: trident
  cloudProvider: "GCP"
  cloudIdentity: "iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com"
```

Helm

Helm을 사용하여 Trident를 설치할 때 클라우드 공급자와 클라우드 ID를 설정하십시오.

```
helm install trident trident-operator-100.6.0.tgz \
  --set cloudProvider=GCP \
  --set cloudIdentity="iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com"
```

tridentctl

클라우드 공급자와 클라우드 ID를 지정하여 Trident를 설치하십시오.

```
tridentctl install \
  --cloud-provider=GCP \
  --cloud-identity="iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com" \
  -n trident
```

NAS 스토리지 구성

Google Cloud NetApp Volumes UNIFIED 스토리지 풀의 경우, Trident는 볼륨 작업 중에 UNIFIED 전용 명명 및 검증 규칙을 적용합니다.

볼륨을 찾을 때 Trident는 여러 호환 가능한 볼륨 이름 변형(예: 하이픈 및 밑줄 형식)을 평가하여 가져오기 및 검색 신뢰성을 향상시킬 수 있습니다.

드라이버 세부 정보

Trident는 `google-cloud-netapp-volumes` Google Cloud NetApp Volumes에서 NAS 스토리지를 프로비저닝하는 드라이버를 제공합니다.

드라이버는 다음과 같은 액세스 모드를 지원합니다.

- ReadWriteOnce(RWO)

- ReadOnlyMany (ROX)
- ReadWriteMany (RWX)
- ReadWriteOncePod(RWOP)

드라이버	프로토콜	**volumeMode**	지원되는 액세스 모드	지원되는 파일 시스템
`google-cloud-netapp-volumes`	NFS SMB	파일 시스템	RWO, ROX, RWX, RWOP	`nfs`, `smb`

Trident NAS 백엔드 구성

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
    network: "<vpc-network>"
```

NAS 볼륨 프로비저닝

NAS 볼륨은 `google-cloud-netapp-volumes` 백엔드를 사용하여 프로비저닝되며 NFS 및 SMB 프로토콜을 지원합니다.

NFS 볼륨용 StorageClass

NFS 볼륨을 프로비저닝하려면 `nasType`을(를) `nfs`(으)로 설정하십시오.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "nfs"
```

```
allowVolumeExpansion: true
```

SMB 볼륨용 StorageClass

SMB 볼륨을 프로비저닝하려면 `nasType`을(를) `smb`(으)로 설정하고 자격 증명을 제공하십시오.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
allowVolumeExpansion: true
```

PersistentVolumeClaim 예시(RWX)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

PersistentVolumeClaim 예시(RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
```

```
      storage: 100Gi
  storageClassName: gcnv-nfs
```

NAS 볼륨은 `volumeMode: Filesystem`을 사용합니다.

SAN 워크로드를 위해 Google Cloud NetApp Volumes 구성

Google Cloud NetApp Volumes에서 iSCSI 프로토콜을 사용하여 블록 스토리지 볼륨을 프로비저닝하도록 Trident를 구성할 수 있습니다. SAN 볼륨은 `google-cloud-netapp-volumes-san` 스토리지 드라이버를 사용하여 Flex Unified 스토리지 풀에서 프로비저닝됩니다.

이 드라이버는 블록 워크로드 전용이며 NAS 프로토콜을 지원하지 않습니다.

`google-cloud-netapp-volumes-san` 백엔드는 iSCSI 블록 볼륨을 프로비저닝하는 데 필요합니다. `google-cloud-netapp-volumes` 백엔드는 NAS 프로토콜만 지원하며 SAN 워크로드에는 사용할 수 없습니다.

개요

Trident는 `google-cloud-netapp-volumes-san` 드라이버를 사용하여 Google Cloud NetApp Volumes SAN(iSCSI) 워크로드를 지원합니다.

SAN 볼륨은 Flex Unified 스토리지 풀에서 프로비저닝되어 iSCSI 블록 디바이스로 Kubernetes 노드에 제공됩니다.

이는 다음 환경에 적용됩니다.

- Trident 26.02 이상
- Google Kubernetes Engine(GKE) 또는 Red Hat OpenShift
- Google Cloud NetApp Volumes Flex 통합 스토리지 풀
- iSCSI 기반 워크로드

Flex Unified 스토리지 풀

Flex Unified 스토리지 풀은 iSCSI 프로토콜을 사용하여 블록 스토리지를 제공하며 SAN 프로비저닝에 필요합니다.

- Flex Unified REGIONAL 풀이 지원됩니다.
- Flex Unified ZONAL 풀은 Trident 26.02.1부터 지원됩니다.
- SAN 워크로드의 경우 **Flex** 서비스 레벨만 지원됩니다.

Trident SAN 백엔드 구성

```
apiVersion: trident.netapp.io/v1
```

```
kind: TridentBackendConfig
metadata:
  name: gcnv-san
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
      performance: flex
    network: "<vpc-network>"
    serviceLevel: Flex
```

StorageClass 생성

SAN 백엔드를 구성한 후 `google-cloud-netapp-volumes-san` 드라이버를 참조하는 StorageClass를 생성합니다.

파일 시스템 유형은 백엔드가 아닌 StorageClass에서 정의됩니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

지원되는 파일 시스템 유형:

- `ext4` (기본값)
- `ext3`
- `xfs`

SAN 드라이버는 Flex 서비스 레벨만 지원하며 `exportRule`, `unixPermissions`, `nasType`, `snapshotDir`, `nfsMountOptions`와 같은 NAS 관련 백엔드 매개변수나 티어링 관련 설정은 사용하지 않습니다.

블록 볼륨 프로비저닝

ReadWriteOnce(RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadWriteOncePod(RWOP)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwop
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadOnlyMany (ROX)

ROX에서 흔히 사용되는 패턴은 기존 ReadWriteOnce 볼륨을 복제하고 복제본을 읽기 전용으로 마운트하는 것입니다.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rox
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 100Gi
```

```
  storageClassName: gcnv-san
  dataSource:
    kind: PersistentVolumeClaim
    name: gcnv-san-rwo
```

ReadWriteMany (RWX) — 원시 블록 전용

ReadWriteMany는 `volumeMode: Block`인 경우에만 지원됩니다.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-raw-rwx
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

블록 볼륨 동작

블록 볼륨은 iSCSI LUN으로 프로비저닝되고 Kubernetes 노드에 블록 디바이스로 제공됩니다.

블록 볼륨:

- iSCSI 프로토콜 사용
- 파일 시스템 및 원시 블록 프레젠테이션 지원
- Trident에 의해 연결되고 관리됩니다
- 여러 Kubernetes 액세스 모드 지원

액세스 모드

Trident에서 프로비저닝한 블록 볼륨은 다음과 같은 액세스 모드를 지원합니다.

- `ReadWriteOnce` (RWO)
- `ReadOnlyMany` (ROX)
- `ReadWriteOncePod` (RWOP)
- `ReadWriteMany` (RWX), 다음과 같은 경우에만 지원됩니다 `volumeMode: Block`

volumeMode 동작

`volumeMode` 필드는 블록 볼륨이 노출되는 방식을 제어합니다.

- `Filesystem` Trident가 볼륨을 포맷하고 마운트합니다.
- `Block` Trident는 장치를 연결하고 이를 원시 블록 장치로 노출합니다.

지원되는 작업

`google-cloud-netapp-volumes-san` 드라이버를 사용하여 프로비저닝된 블록 볼륨은 다음을 지원합니다:

- 생성
- 삭제
- 클론
- 스냅샷
- 크기 조정
- 가져오기

추가 GiB 오버프로비저닝 동작

Google Cloud NetApp Volumes 블록 볼륨에는 내부 메타데이터 오버헤드가 포함됩니다. 이 오버헤드로 인해 커널에서 인식되는 장치 크기가 프로비저닝된 용량보다 작아집니다.

테스트 결과:

- 초기 생성 시 약 300 KiB의 오버헤드가 발생합니다
- 크기 조정 후 최대 약 107MiB의 오버헤드

Google Cloud NetApp Volumes는 GiB 단위의 할당만 허용하므로 Trident는 다음과 같은 방법으로 사용 가능한 장치 크기가 항상 PVC 요청을 충족하거나 초과하도록 합니다.

- 요청된 크기를 다음 정수 GiB로 올림합니다
- 추가 1 GiB 버퍼 추가

예:

- PVC 요청: 100 GiB
- Google Cloud NetApp Volumes에 프로비저닝된 크기: 101 GiB
- 애플리케이션에 표시되는 사용 가능한 공간: 최소 100 GiB

Pod 예

파일 시스템 마운트 블록 볼륨**(RWO)**

```
apiVersion: v1
kind: Pod
metadata:
  name: app-rwo
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeMounts:
    - name: data
      mountPath: /mnt/data
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-rwo
```

원시 블록 디바이스**(RWX)**

```
apiVersion: v1
kind: Pod
metadata:
  name: app-raw-rwx
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeDevices:
    - name: data
      devicePath: /dev/xda
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-raw-rwx
```

연결 및 마운트 동작

Google Cloud NetApp Volumes에서 프로비저닝된 SAN 볼륨의 경우:

- Trident는 Flex Unified 스토리지 풀에 LUN(논리 유닛 번호)을 생성합니다.
- 게시 중에 Trident는 LUN을 노드별 호스트 그룹에 매핑합니다.
- 노드 스테이징 중에 Trident는 다음을 수행합니다.
  - iSCSI 타겟에 로그인합니다
  - LUN을 검색합니다
  - 다중 경로를 구성합니다
- `volumeMode: Filesystem`Trident는 필요한 경우 장치를 포맷하고 마운트합니다.
- `volumeMode: Block`인 경우 Trident는 포맷이나 마운트 없이 장치를 연결하고 Pod에 직접 노출합니다.

> SAN 블록 볼륨은 분산 잠금 또는 쓰기 조정을 제공하지 않습니다. 여러 노드에서 블록 볼륨에 액세스할 때(ReadWriteMany 사용 시 `volumeMode: Block`) 애플리케이션 또는 파일 시스템이 동시성을 관리해야 합니다.

Google Cloud NetApp Volumes 백엔드 구성을 준비하세요

Google Cloud NetApp Volumes 백엔드를 구성하기 전에 다음 요구 사항을 충족해야 합니다.

NFS 또는 SMB 볼륨의 사전 요구 사항

Google Cloud NetApp Volumes를 처음 사용하거나 새 위치에서 사용하는 경우 Google Cloud NetApp Volumes를 설정하고 NFS 또는 SMB 볼륨을 생성하려면 몇 가지 초기 구성이 필요합니다. 다음을 참조하십시오. "시작하기 전에".

Google Cloud NetApp Volumes 백엔드를 구성하기 전에 다음 사항을 확인하십시오.

- Google Cloud NetApp Volumes 서비스로 구성된 Google Cloud 계정. 다음을 참조하십시오. "Google Cloud NetApp Volumes".
- Google Cloud 계정의 프로젝트 번호입니다. 다음을 참조하십시오. "프로젝트 식별".
- NetApp Volumes Admin (`roles/netapp.admin` 역할을 가진 Google Cloud 서비스 계정 다음을 참조하십시오. "Identity and Access Management 역할 및 권한".
- GCNV 계정용 API 키 파일입니다. 다음을 참조하십시오. "서비스 계정 키 생성"
- 스토리지 풀. 다음을 참조하십시오. "스토리지 풀 개요".

Google Cloud NetApp Volumes에 대한 액세스 설정 방법에 대한 자세한 내용은 "Google Cloud NetApp Volumes에 대한 액세스 설정"을 참조하십시오.

Google Cloud NetApp Volumes 백엔드 구성 옵션 및 예시

Google Cloud NetApp Volumes의 백엔드 구성 옵션에 대해 알아보고 구성 예제를 검토하십시오.

백엔드 configuration 옵션

각 백엔드는 단일 Google Cloud 리전에 볼륨을 프로비저닝합니다. 다른 리전에 볼륨을 생성하려면 추가 백엔드를

정의할 수 있습니다.

매개변수	설명	기본값
`version`		항상 1
`storageDriverName`	스토리지 드라이버의 이름	`storageDriverName`의 값은 "google-cloud-netapp-volumes"로 지정해야 합니다.
`backendName`	(선택 사항) 스토리지 백엔드의 사용자 지정 이름	드라이버 이름 + "_" + API key의 일부
`storagePools`	볼륨 생성을 위한 스토리지 풀을 지정하는 데 사용되는 선택적 매개 변수입니다.	
`projectNumber`	Google Cloud 계정 프로젝트 번호입니다. 이 값은 Google Cloud 포털 홈페이지에서 확인할 수 있습니다.	
`location`	Trident가 GCNV 볼륨을 생성하는 Google Cloud 위치입니다. 교차 리전 Kubernetes 클러스터를 생성할 때 `location`에서 생성된 볼륨은 여러 Google Cloud 리전의 노드에서 예약된 워크로드에 사용할 수 있습니다. 교차 리전 트래픽에는 추가 비용이 발생합니다.	
`apiKey`	`netapp.admin` 역할이 있는 Google Cloud 서비스 계정의 API 키입니다. 여기에는 Google Cloud 서비스 계정의 개인 키 파일의 JSON 형식 내용이 포함됩니다(백엔드 구성 파일에 그대로 복사됨). `apiKey`에는 다음 키에 대한 키-값 쌍이 포함되어야 합니다: `type`, `project_id`, `client_email`, `client_id`, `auth_uri`, `token_uri`, `auth_provider_x509_cert_url` 및 `client_x509_cert_url`.	
`nfsMountOptions`	NFS 마운트 옵션을 세밀하게 제어할 수 있습니다.	"nfsvers=3"
`limitVolumeSize`	요청된 볼륨 크기가 이 값보다 크면 프로비저닝이 실패합니다.	"" (기본적으로 적용되지 않음)
`serviceLevel`	스토리지 풀 및 해당 볼륨의 서비스 수준입니다. 값은 `flex`, `standard`, `premium` 또는 `extreme`입니다.	
`labels`	볼륨에 적용할 임의의 JSON 형식 레이블 세트	""
`network`	GCNV 볼륨에 사용되는 Google Cloud 네트워크입니다.	
`debugTraceFlags`	문제 해결 시 사용할 디버그 플래그입니다. 예 `{"api":false, "method":true}`. 문제 해결 중이거나 자세한 로그 덤프가 필요한 경우가 아니면 사용하지 마십시오.	null
`nasType`	NFS 또는 SMB 볼륨 생성을 구성합니다. 옵션은 `nfs`, `smb` 또는 null입니다. null로 설정하면 기본적으로 NFS 볼륨이 사용됩니다.	`nfs`

매개변수	설명	기본값
supportedTopologies	이 백엔드에서 지원하는 지역 및 영역 목록을 나타냅니다. 자세한 내용은 "CSI 토폴로지 사용"을 참조하십시오. 예를 들면 다음과 같습니다. supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

볼륨 프로비저닝 옵션

구성 파일의 defaults 섹션에서 기본 볼륨 프로비저닝을 제어할 수 있습니다.

매개변수	설명	기본값
exportRule	새 볼륨에 대한 내보내기 규칙입니다. IPv4 주소의 모든 조합을 쉼표로 구분한 목록이어야 합니다.	"0.0.0.0/0"
snapshotDir	.snapshot 디렉토리에 대한 액세스	true, false (기본 동작은 다를 수 있습니다. 명시적으로 설정) NFSv3의 경우 "false"
snapshotReserve	스냅샷용으로 예약된 볼륨의 비율	""(기본값 0 적용)
unixPermissions	새 볼륨의 Unix 권한(8진수 4자리).	""

예시 구성

다음 예시들은 대부분의 매개변수를 기본값으로 유지하는 기본 구성을 보여줍니다. 이것이 백엔드를 정의하는 가장 쉬운 방법입니다.

▼ 최소 구성

이는 최소한의 백엔드 구성입니다. 이 구성을 사용하면 Trident는 구성된 위치에서 Google Cloud NetApp Volumes에 위임된 모든 스토리지 풀을 검색하고 새 볼륨을 해당 풀 중 하나에 무작위로 배치합니다. nasType`이 생략되었으므로 `nfs 기본값이 적용되어 백엔드에서 NFS 볼륨을 프로비저닝합니다.

이 구성은 Google Cloud NetApp Volumes를 처음 사용하고 테스트할 때 이상적이지만, 실제로는 프로비저닝하는 볼륨에 대해 추가적인 범위 지정이 필요할 수 있습니다.

<id_value> 및 `<key_value>`를 서비스 계정 자격 증명으로 바꾸십시오.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
```

```
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ **SMB 볼륨 구성**

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
```

```
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ **StoragePools** 필터를 사용한 구성

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
```

```
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ 가상 풀 구성

이 백엔드 구성은 단일 파일에 여러 개의 가상 풀을 정의합니다. 가상 풀은 `storage` 섹션에 정의됩니다. 여러 개의 스토리지 풀이 서로 다른 서비스 수준을 지원하고, 이를 나타내는 스토리지 클래스를 Kubernetes에 생성하려는 경우에 유용합니다. 가상 풀 레이블은 풀을 구분하는 데 사용됩니다. 예를 들어, 아래 예시에서는 `performance` 레이블과 `serviceLevel` 유형을 사용하여 가상 풀을 구분합니다.

모든 가상 풀에 적용할 기본값을 설정하고 개별 가상 풀의 기본값을 재정의할 수도 있습니다. 다음 예에서는 `snapshotReserve`와 `exportRule`가 모든 가상 풀의 기본값으로 사용됩니다.

자세한 내용은 "가상 풀"을 참조하십시오.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
```

```
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
  defaults:
    snapshotReserve: "10"
    exportRule: 10.0.0.0/24
  storage:
    - labels:
        performance: extreme
      serviceLevel: extreme
      defaults:
        snapshotReserve: "5"
        exportRule: 0.0.0.0/0
    - labels:
        performance: premium
      serviceLevel: premium
    - labels:
        performance: standard
      serviceLevel: standard
```

▼ **GKE용 클라우드 ID**

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
```

```
  storagePool: pool-premium1
```

▼ 지원되는 토폴로지 구성

Trident는 리전 및 가용 영역을 기반으로 워크로드용 볼륨 프로비저닝을 지원합니다. 이 백엔드 구성의 `supportedTopologies` 블록은 백엔드별 리전 및 영역 목록을 제공하는 데 사용됩니다. 여기에 지정된 리전 및 영역 값은 각 Kubernetes 클러스터 노드의 레이블에 있는 리전 및 영역 값과 일치해야 합니다. 이러한 리전 및 영역은 스토리지 클래스에 제공될 수 있는 허용 값 목록을 나타냅니다. 백엔드에 제공된 리전 및 영역의 하위 집합을 포함하는 스토리지 클래스의 경우 Trident는 언급된 리전 및 영역에 볼륨을 생성합니다. 자세한 내용은 "CSI 토폴로지 사용"을 참조하십시오.

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

다음 단계

백엔드 구성 파일을 생성한 후 다음 명령을 실행하십시오.

```
kubectl create -f <backend-file>
```

백엔드가 성공적으로 생성되었는지 확인하려면 다음 명령을 실행하십시오.

```
kubectl get tridentbackendconfig

NAME                BACKEND NAME        BACKEND UUID
PHASE   STATUS
backend-tbc-gcnv    backend-tbc-gcnv    b2fd1ff9-b234-477e-88fd-713913294f65
Bound   Success
```

백엔드 생성에 실패하면 백엔드 구성에 문제가 있는 것입니다. `kubectl get tridentbackendconfig <backend-name>` 명령을 사용하여 백엔드를 설명하거나 다음 명령을 실행하여 로그를 보고 원인을 확인할 수 있습니다.

```
tridentctl logs
```

구성 파일의 문제를 식별하고 수정한 후 백엔드를 삭제하고 생성 명령을 다시 실행할 수 있습니다.

스토리지 클래스 정의

다음은 위의 백엔드를 참조하는 기본 `StorageClass` 정의입니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

- `parameter.selector` 필드를 사용한 예시 정의:*

  `parameter.selector`을 사용하면 각 `StorageClass`에 대해 볼륨을 호스팅하는 데 사용되는 xref:{relative_path}../trident-concepts/virtual-storage-pool.html["가상 풀"]을 지정할 수 있습니다. 볼륨은 선택한 풀에 정의된 속성을 갖게 됩니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes
```

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes
```

스토리지 클래스에 대한 자세한 내용은 "스토리지 클래스를 생성합니다"을(를) 참조하십시오.

SMB 볼륨에 대한 정의 예

`nasType`, `node-stage-secret-name` 및 `node-stage-secret-namespace`를 사용하여 SMB 볼륨을 지정하고 필요한 Active Directory 자격 증명을 제공할 수 있습니다. 노드 단계 비밀에는 권한이 있거나 없는 모든 Active Directory 사용자/암호를 사용할 수 있습니다.

▼ 기본 네임스페이스의 기본 구성

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

▼ 네임스페이스별로 서로 다른 시크릿 사용

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

▼ 볼륨마다 다른 시크릿 사용

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

`nasType: smb` SMB 볼륨을 지원하는 풀에 대한 필터입니다. `nasType: nfs` 또는 `nasType: null` NFS 풀에 대한 필터입니다.

PVC 정의 예

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

PVC가 바인딩되었는지 확인하려면 다음 명령을 실행합니다.

```
kubectl get pvc gcnv-nfs-pvc

NAME           STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS AGE
gcnv-nfs-pvc   Bound    pvc-b00f2414-e229-40e6-9b16-ee03eb79a213   100Gi
RWX            gcnv-nfs-sc  1m
```

Google Cloud NetApp Volumes에 대한 자동 계층화 구성

자동 계층화는 볼륨 프로비저닝 중에 Trident 백엔드 매개변수와 PersistentVolumeClaim 주석을 통해 구성됩니다. Trident를 사용하여 Google Cloud NetApp Volumes에 대한 자동

계층화를 구성할 수 있습니다.

개요

자동 계층화 기능을 통해 Trident는 비활성 데이터를 성능 계층에서 용량 계층으로 자동으로 이동시키는 볼륨을 프로비저닝할 수 있습니다. 이를 통해 스토리지 비용을 절감하는 동시에 자주 액세스하는 데이터의 성능을 유지할 수 있습니다.

Trident는 볼륨 생성 시에만 자동 계층화 설정을 적용합니다. Trident 26.02에서는 프로비저닝 후 변경이 지원되지 않습니다.

개념

자동 계층화

자동 계층화는 접근 패턴에 따라 접근 빈도가 낮은 데이터를 성능 계층에서 용량 계층으로 이동시킵니다. 데이터 이동은 비동기적으로 발생하며 즉시 이루어지지 않습니다.

계층화 정책

계층화 정책은 볼륨에 대해 자동 계층화를 활성화할지 여부를 결정합니다.

다음 정책이 지원됩니다. * `auto`: 액세스 패턴에 따른 자동 계층화 활성화 * `none`: 자동 계층화 비활성화

냉방 일수

냉각일은 데이터 블록이 계층화 대상이 되기 전에 비활성 상태로 유지되어야 하는 최소 일수를 지정합니다. 냉각일은 계층화 정책이 `auto`로 설정된 경우에만 적용됩니다.

구성 모델

구성 범위

자동 계층화는 여러 범위에서 구성할 수 있습니다.

- 스토리지 풀 범위 풀에서 프로비저닝된 모든 볼륨에 적용됩니다.
- 볼륨 범위 PersistentVolumeClaim 주석을 통해 단일 볼륨에 적용됩니다.

Trident는 각 설정이 정의된 위치를 기반으로 유효한 구성을 결정합니다.

구성 우선 순위

동일한 설정이 여러 범위에서 정의된 경우 Trident는 다음 우선순위를 적용합니다.

1. PersistentVolumeClaim 주석
2. Trident 백엔드 구성
3. 스토리지 풀 기본값

우선 순위가 높은 설정이 하위 수준 값을 재정의합니다.

Trident 26.02에서 지원되는 기능

Trident 26.02는 Google Cloud NetApp Volumes에 대해 다음과 같은 자동 계층화 기능을 지원합니다.

- 볼륨 프로비저닝 중 자동 계층화 설정 또는 해제
- Trident 백엔드 구성에서 계층화 정책 정의
- PVC 주석을 사용하여 볼륨별 계층화 정책 및 냉각 일수 재정의
- 자동 계층화가 활성화된 볼륨에 대한 냉각 일수 구성

Trident 26.02에서 지원되지 않는 기능

다음 작업은 지원되지 않습니다.

- 볼륨 생성 후 자동 계층화 설정 수정
- Kubernetes 업데이트를 사용하여 기존 볼륨의 계층화 정책 변경
- Trident에서 관리하는 프로비저닝 워크플로 외부에서 자동 계층화 설정 적용

백엔드 구성 매개 변수

다음 매개변수는 Trident 백엔드 구성에 정의될 때 자동 계층화 동작을 제어합니다.

매개변수	필수	설명
tieringPolicy	아니요	볼륨에 대한 계층화 정책 (`auto` 또는 `none`)
tieringMinimumCoolingDays	아니요	데이터가 계층화되기 전의 비활성 일수(범위: 2-183, 기본값: 31)

PersistentVolumeClaim 어노테이션을 사용한 볼륨 수준 재정의

지원되는 주석

PersistentVolumeClaim 어노테이션을 사용하면 볼륨별로 자동 계층화 설정을 재정의할 수 있습니다.

주석	설명
`trident.netapp.io/tieringPolicy`	볼륨에 대한 티어링 정책을 재정의합니다
`trident.netapp.io/tieringMinimumCoolingDays`	볼륨에 대한 쿨링 일수 값을 재정의합니다

예시: 자동 계층화 재정의가 적용된 **PersistentVolumeClaim**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: auto-tiering-pvc
  annotations:
```

```
    trident.netapp.io/tieringPolicy: auto
    trident.netapp.io/tieringMinimumCoolingDays: "45"
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-cloud-netapp-volumes-auto-tiering
  resources:
    requests:
      storage: 500Gi
```

동작 및 제한 사항

프로비저닝 동작

- 자동 계층화 설정은 볼륨 생성 시에만 평가 및 적용됩니다.
- Trident는 프로비저닝 후 계층화 구성을 조정하지 않습니다.
- 계층화 정책이 `none`로 설정된 경우 쿨링 기간은 무시됩니다.

플랫폼 제한 사항

- 자동 계층화는 NAS 볼륨(NFS 및 SMB)에서만 지원됩니다.
- 블록 볼륨(iSCSI)은 자동 계층화를 지원하지 않습니다.
- Google Cloud NetApp Volumes 스토리지 풀은 Google Cloud에서 자동 계층화가 활성화되어 있어야 합니다.

지원되는 값

- `tieringMinimumCoolingDays`의 유효 범위: 2~183
- 기본값: 31

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 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 라이센스와 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 라이센스도 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 라이센스를 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 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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