



# Google Cloud NetApp Volumes

## Trident

NetApp  
January 15, 2026

This PDF was generated from <https://docs.netapp.com/ko-kr/trident-2506/trident-use/gcnv.html> on January 15, 2026. Always check docs.netapp.com for the latest.

# 목차

Google Cloud NetApp Volumes .....	1
Google Cloud NetApp Volumes 백엔드 구성 .....	1
Google Cloud NetApp Volumes 드라이버 세부 정보 .....	1
GKE용 클라우드 ID .....	1
Google Cloud NetApp Volumes 백엔드 구성을 준비하세요 .....	3
NFS 볼륨의 전제 조건 .....	3
Google Cloud NetApp Volumes 백엔드 구성 옵션 및 예시 .....	3
백엔드 구성 옵션 .....	3
볼륨 프로비저닝 옵션 .....	5
구성 예 .....	5
다음은 무엇인가요? .....	13
스토리지 클래스 정의 .....	14

# Google Cloud NetApp Volumes

## Google Cloud NetApp Volumes 백엔드 구성

이제 Google Cloud NetApp Volumes Trident 의 백엔드로 구성할 수 있습니다. Google Cloud NetApp Volumes 백엔드를 사용하여 NFS 및 SMB 볼륨을 연결할 수 있습니다.

### Google Cloud NetApp Volumes 드라이버 세부 정보

Trident 다음을 제공합니다. `google-cloud-netapp-volumes` 클러스터와 통신하는 드라이버. 지원되는 액세스 모드는 다음과 같습니다: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

운전사	규약	볼륨모드	지원되는 액세스 모드	지원되는 파일 시스템
<code>google-cloud-netapp-volumes</code>	NFS SMB	파일 시스템	RWO, ROX, RWX, RWOP	nfs, smb

### GKE용 클라우드 ID

클라우드 ID를 사용하면 Kubernetes 포드가 명시적인 Google Cloud 자격 증명을 제공하는 대신 워크로드 ID로 인증하여 Google Cloud 리소스에 액세스할 수 있습니다.

Google Cloud에서 클라우드 ID를 활용하려면 다음이 필요합니다.

- GKE를 사용하여 배포된 Kubernetes 클러스터.
- GKE 클러스터에 구성된 워크로드 ID와 노드 풀에 구성된 GKE 메타데이터 서버.
- Google Cloud NetApp Volumes 관리자(`roles/netapp.admin`) 역할 또는 사용자 지정 역할이 있는 GCP 서비스 계정.
- "GCP"를 지정하는 `cloudProvider`와 새로운 GCP 서비스 계정을 지정하는 `cloudIdentity`를 포함하는 Trident 설치되었습니다. 아래에 예를 들어보겠습니다.

## Trident 연산자

Trident 연산자를 사용하여 Trident 설치하려면 다음을 편집하세요. `tridentorchestrator_cr.yaml` 설정하다 `cloudProvider` 에게 "GCP" 그리고 설정하다 `cloudIdentity` 에게 `iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com`.

예를 들어:

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

## 지배

다음 환경 변수를 사용하여 **cloud-provider (CP)** 및 **cloud-identity (CI)** 플래그의 값을 설정합니다.

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

다음 예제에서는 Trident 설치하고 설정합니다. `cloudProvider` 환경 변수를 사용하여 GCP에 `$CP` 그리고 설정합니다 `cloudIdentity` 환경 변수 사용 `$ANNOTATION`:

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

## <code>트라이던트ctl</code>

다음 환경 변수를 사용하여 클라우드 공급자 및 클라우드 ID 플래그의 값을 설정합니다.

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

다음 예제에서는 Trident 설치하고 설정합니다. `cloud-provider` 플래그를 `$CP`, 그리고 `cloud-identity` 에게 `$ANNOTATION`:

```
tridentctl install --cloud-provider=$CP --cloud  
-identity="$ANNOTATION" -n trident
```

## Google Cloud NetApp Volumes 백엔드 구성을 준비하세요

Google Cloud NetApp Volumes 백엔드를 구성하기 전에 다음 요구 사항이 충족되는지 확인해야 합니다.

### NFS 볼륨의 전제 조건

Google Cloud NetApp Volumes 처음 사용하거나 새로운 위치에서 사용하는 경우 Google Cloud NetApp Volumes 설정하고 NFS 볼륨을 생성하기 위해 초기 구성이 필요합니다. 참조하다 ["시작하기 전에"](#).

Google Cloud NetApp Volumes 백엔드를 구성하기 전에 다음 사항이 있는지 확인하세요.

- Google Cloud NetApp Volumes 서비스로 구성된 Google Cloud 계정입니다. 참조하다 ["Google Cloud NetApp Volumes"](#).
- Google Cloud 계정의 프로젝트 번호입니다. 참조하다 ["프로젝트 식별"](#).
- NetApp Volumes Admin이 있는 Google Cloud 서비스 계정(roles/netapp.admin) 역할. 참조하다 ["ID 및 액세스 관리 역할 및 권한"](#).
- GCNV 계정의 API 키 파일입니다. 참조하다 ["서비스 계정 키 생성"](#).
- 저장 풀. 참조하다 ["스토리지 풀 개요"](#).

Google Cloud NetApp Volumes 에 대한 액세스를 설정하는 방법에 대한 자세한 내용은 다음을 참조하세요. ["Google Cloud NetApp Volumes 에 대한 액세스 설정"](#).

## Google Cloud NetApp Volumes 백엔드 구성 옵션 및 예시

Google Cloud NetApp Volumes 의 백엔드 구성 옵션에 대해 알아보고 구성 예를 검토하세요.

### 백엔드 구성 옵션

각 백엔드는 단일 Google Cloud 지역에서 볼륨을 프로비저닝합니다. 다른 지역에 볼륨을 생성하려면 추가 백엔드를 정의할 수 있습니다.

매개변수	설명	기본
version		항상 1
storageDriverName	저장 드라이버의 이름	의 가치 storageDriverName "google-cloud-netapp-volumes"로 지정해야 합니다.

매개변수	설명	기본
backendName	(선택 사항) 스토리지 백엔드의 사용자 정의 이름	드라이버 이름 + "_" + API 키의 일부
storagePools	볼륨 생성을 위한 스토리지 풀을 지정하는 데 사용되는 선택적 매개변수입니다.	
projectNumber	Google Cloud 계정 프로젝트 번호. 해당 값은 Google Cloud 포털 홈페이지에서 확인할 수 있습니다.	
location	Trident GCNV 볼륨을 생성하는 Google Cloud 위치입니다. 지역 간 Kubernetes 클러스터를 생성할 때 볼륨이 생성됩니다. location 여러 Google Cloud 지역의 노드에 예약된 워크로드에 사용할 수 있습니다. 지역 간 트래픽에는 추가 비용이 발생합니다.	
apiKey	Google Cloud 서비스 계정에 대한 API 키 netapp.admin 역할. 여기에는 Google Cloud 서비스 계정의 개인 키 파일의 JSON 형식 내용이 포함됩니다 (백엔드 구성 파일에 그대로 복사됨). 그만큼 apiKey 다음 키에 대한 키-값 쌍을 포함해야 합니다. type, project_id, client_email, client_id, auth_uri, token_uri, auth_provider_x509_cert_url, 그리고 client_x509_cert_url.	
nfsMountOptions	NFS 마운트 옵션에 대한 세부적인 제어.	"nfsvers=3"
limitVolumeSize	요청된 볼륨 크기가 이 값보다 크면 프로비저닝에 실패합니다.	"" (기본적으로 적용되지 않음)
serviceLevel	스토리지 풀의 서비스 수준과 볼륨입니다. 값은 다음과 같습니다 flex, standard, premium, 또는 extreme.	
labels	볼륨에 적용할 임의의 JSON 형식 레이블 세트	""
network	GCNV 볼륨에 Google Cloud 네트워크가 사용됩니다.	
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예, {"api":false, "method":true}. 문제 해결을 위해 자세한 로그 덤프가 필요한 경우가 아니면 이 기능을 사용하지 마세요.	널
nasType	NFS 또는 SMB 볼륨 생성을 구성합니다. 옵션은 다음과 같습니다 nfs, smb 또는 null. null로 설정하면 기본적으로 NFS 볼륨이 사용됩니다.	nfs
supportedTopologies	이 백엔드에서 지원하는 지역 및 영역 목록을 나타냅니다. 자세한 내용은 다음을 참조하세요. " <a href="#">CSI 토폴로지 사용</a> ". 예를 들어: supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

## 볼륨 프로비저닝 옵션

기본 볼륨 프로비저닝을 제어할 수 있습니다. `defaults` 구성 파일의 섹션.

매개변수	설명	기본
<code>exportRule</code>	새로운 볼륨에 대한 내보내기 규칙. IPv4 주소의 조합을 심표로 구분하여 나열해야 합니다.	"0.0.0.0/0"
<code>snapshotDir</code>	에 대한 액세스 <code>.snapshot</code> 예배 규칙서	NFSv4의 경우 "true", NFSv3의 경우 "false"
<code>snapshotReserve</code>	스냅샷을 위해 예약된 볼륨의 백분율	"" (기본값 0 허용)
<code>unixPermissions</code>	새로운 볼륨의 유닉스 권한(8진수 4자리).	""

## 구성 예

다음 예에서는 대부분의 매개변수를 기본값으로 두는 기본 구성을 보여줍니다. 백엔드를 정의하는 가장 쉬운 방법입니다.

## 최소 구성

이는 백엔드의 최소 구성입니다. 이 구성을 사용하면 Trident 구성된 위치에서 Google Cloud NetApp Volumes에 위임된 모든 스토리지 풀을 검색하고 해당 풀 중 하나에 무작위로 새 볼륨을 배치합니다. 왜냐하면 `nasType` 생략됩니다. `nfs` 기본값이 적용되고 백엔드가 NFS 볼륨을 프로비저닝합니다.

이 구성은 Google Cloud NetApp Volumes 처음 시작하고 여러 가지를 시도해 볼 때 이상적이지만 실제로는 프로비저닝하는 볼륨에 대한 추가 범위를 제공해야 할 가능성이 높습니다.



```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
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

```

```
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: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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 다음에 정의됩니다. `storage` 부분. 여러 개의 스토리지 풀이 서로 다른 서비스 수준을 지원하고 이를 나타내는 Kubernetes의 스토리지 클래스를 만들려는 경우에 유용합니다. 가상 풀 레이블은 풀을 구별하는 데 사용됩니다. 예를 들어, 아래 예에서 `performance` 라벨 및 `serviceLevel` 유형은 가상 풀을 구별하는 데 사용됩니다.

모든 가상 풀에 적용할 수 있는 기본값을 설정하고, 개별 가상 풀의 기본값을 덮어쓸 수도 있습니다. 다음 예에서, `snapshotReserve` 그리고 `exportRule` 모든 가상 풀의 기본값으로 사용됩니다.

자세한 내용은 다음을 참조하세요. "[가상 풀](#)".

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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

```

Trident 지역 및 가용성 영역에 따라 워크로드에 대한 볼륨 프로비저닝을 용이하게 합니다. 그만큼 supportedTopologies 이 백엔드 구성의 블록은 백엔드당 지역 및 영역 목록을 제공하는 데 사용됩니다. 여기에 지정된 지역 및 영역 값은 각 Kubernetes 클러스터 노드의 레이블에 있는 지역 및 영역 값과 일치해야 합니다. 이러한 지역과 영역은 저장 클래스에서 제공될 수 있는 허용 가능한 값 목록을 나타냅니다. 백엔드에서 제공하는 지역 및 영역의 하위 집합을 포함하는 스토리지 클래스의 경우, Trident 언급된 지역 및 영역에 볼륨을 생성합니다. 자세한 내용은 다음을 참조하세요. "[CSI 토폴로지 사용](#)".

```
---
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
```

## 다음은 무엇인가요?

백엔드 구성 파일을 만든 후 다음 명령을 실행합니다.

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

백엔드가 성공적으로 생성되었는지 확인하려면 다음 명령을 실행하세요.

```
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. 백엔드를 사용하여 설명할 수 있습니다. `kubectl get tridentbackendconfig <backend-name>` 다음 명령을 실행하여 원인을 파악하려면 명령을 실행하거나 로그를 확인하세요.

```
tridentctl logs
```

구성 파일의 문제를 파악하고 수정한 후 백엔드를 삭제하고 create 명령을 다시 실행할 수 있습니다.

## 스토리지 클래스 정의

다음은 기본입니다 StorageClass 위의 백엔드를 참조하는 정의입니다.

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

- 다음을 사용한 정의 예 parameter.selector 필드:\*

사용 중 parameter.selector 각각에 대해 지정할 수 있습니다 StorageClass 그만큼 "가상 풀" 볼륨을 호스팅하는 데 사용됩니다. 볼륨에는 선택된 풀에서 정의된 측면이 있습니다.



```

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

```

저장 클래스에 대한 자세한 내용은 다음을 참조하세요. "[스토리지 클래스 생성](#)".

### SMB 볼륨에 대한 예제 정의

사용 중 nasType , node-stage-secret-name , 그리고 node-stage-secret-namespace SMB 볼륨을 지정하고 필요한 Active Directory 자격 증명을 제공할 수 있습니다. 노드 단계 비밀에는 권한이 있거나 없는 모든 Active Directory 사용자/비밀번호를 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 다른 비밀 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}
```

## 볼륨별로 다른 비밀 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 필터.

## 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 실행하세요.

```
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	

## 저작권 정보

Copyright © 2026 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.