ONTAP SAN 드라이버

Trident

NetApp
March 04, 2026

목차

ONTAP SAN 드라이버

ONTAP SAN 드라이버 개요

ONTAP 및 Cloud Volumes ONTAP SAN 드라이버를 사용하여 ONTAP 백엔드를 구성하는 방법에 대해 알아보십시오.

ONTAP SAN 드라이버 세부 정보입니다

Trident는 ONTAP 클러스터와 통신할 수 있도록 다음과 같은 SAN 스토리지 드라이버를 제공합니다. 지원되는 액세스 모드는 *ReadWriteOnce*(RWO), *ReadOnlyMany*(ROX), *ReadWriteMany*(rwx), *ReadWriteOncePod*(RWOP)입니다.

드라이버	프로토콜	볼륨 모드	액세스 모드가 지원됩니다	지원되는 파일 시스템
`ontap-san`	FC를 통한 iSCSI SCSI(Trident 24.10의 기술 미리 보기)	블록	RWO, ROX, rwx, RWOP	파일 시스템이 없습니다. 원시 블록 디바이스입니다
`ontap-san`	FC를 통한 iSCSI SCSI(Trident 24.10의 기술 미리 보기)	파일 시스템	RWO, 공화당 파일 시스템 볼륨 모드에서는 ROX 및 rwx를 사용할 수 없습니다.	`xfs`, `ext3`, `ext4`
`ontap-san`	NVMe/TCP 을 NVMe/TCP 에 대한 추가 고려사항참조하십시오.	블록	RWO, ROX, rwx, RWOP	파일 시스템이 없습니다. 원시 블록 디바이스입니다
`ontap-san`	NVMe/TCP 을 NVMe/TCP 에 대한 추가 고려사항참조하십시오.	파일 시스템	RWO, 공화당 파일 시스템 볼륨 모드에서는 ROX 및 rwx를 사용할 수 없습니다.	`xfs`, `ext3`, `ext4`
`ontap-san-economy`	iSCSI	블록	RWO, ROX, rwx, RWOP	파일 시스템이 없습니다. 원시 블록 디바이스입니다

드라이버	프로토콜	볼륨 모드	액세스 모드가 지원됩니다	지원되는 파일 시스템
`ontap-san-economy`	iSCSI	파일 시스템	RWO, 공화당 파일 시스템 볼륨 모드에서는 ROX 및 rwx를 사용할 수 없습니다.	`xfs`, `ext3`, `ext4`

- `ontap-san-economy`영구 볼륨 사용률 수가 보다 높을 것으로 예상되는 경우에만 "지원되는 ONTAP 볼륨 제한"사용합니다.
- `ontap-nas-economy`영구 볼륨 사용 횟수가 다음보다 크고 `ontap-san-economy` 드라이버를 사용할 수 없는 경우에만 "지원되는 ONTAP 볼륨 제한"사용합니다.
- 데이터 보호, 재해 복구 또는 이동성이 필요할 것으로 예상되는 경우에는 사용하지 마십시오 `ontap-nas-economy`.

사용자 권한

Trident는 ONTAP 또는 SVM 관리자로 실행해야 하며, 일반적으로 클러스터 사용자 `vsadmin` 또는 SVM 사용자 또는 같은 역할을 가진 다른 이름의 사용자를 사용할 `admin` 것입니다. Amazon FSx for NetApp ONTAP 배포의 경우 Trident은 클러스터 사용자 또는 `vsadmin` SVM 사용자를 사용하여 ONTAP 또는 SVM 관리자로 실행하거나 `fsxadmin` 동일한 역할을 가진 다른 이름의 사용자를 실행해야 합니다. `fsxadmin`사용자는 클러스터 관리자를 제한적으로 대체합니다.

`limitAggregateUsage`매개 변수를 사용하려면 클러스터 관리 권한이 필요합니다. `Trident`와 함께 `Amazon FSx for NetApp ONTAP`를 사용할 때 `limitAggregateUsage` 매개 변수는 및 `fsxadmin` 사용자 계정에서 작동하지 `vsadmin` 않습니다. 이 매개 변수를 지정하면 구성 작업이 실패합니다.

Trident 드라이버가 사용할 수 있는 더 제한적인 역할을 ONTAP 내에 만들 수 있지만 권장하지 않습니다. Trident의 대부분의 새로운 릴리즈에서는 추가 API를 호출하므로 업그레이드가 어렵고 오류가 발생하기 쉽습니다.

NVMe/TCP에 대한 추가 고려사항

Trident는 다음과 같은 드라이버를 사용하여 비휘발성 메모리 익스프레스(NVMe) 프로토콜을 `ontap-san` 지원합니다.

- IPv6를 참조하십시오
- NVMe 볼륨의 스냅샷 및 클론
- NVMe 볼륨 크기 조정
- Trident에서 라이프사이클을 관리할 수 있도록 Trident 외부에서 생성된 NVMe 볼륨을 가져옵니다
- NVMe 네이티브 다중 경로
- K8 노드의 정상 또는 비정상적으로 종료 (24.06)

Trident는 다음을 지원하지 않습니다.

- NVMe에서 기본적으로 지원하는 DH-HMAC-CHAP입니다

- DM(Device Mapper) 경로 다중화
- LUKS 암호화

ONTAP SAN 드라이버를 사용하여 백엔드를 구성할 준비를 합니다

ONTAP SAN 드라이버를 사용하여 ONTAP 백엔드를 구성하기 위한 요구 사항 및 인증 옵션을 이해합니다.

요구 사항

모든 ONTAP 백엔드에 대해 Trident에는 SVM에 할당된 애그리게이트가 하나 이상 필요합니다.

또한 둘 이상의 드라이버를 실행하고 둘 중 하나를 가리키는 스토리지 클래스를 생성할 수도 있습니다. 예를 들어, 드라이버를 사용하는 클래스와 `ontap-san` 드라이버를 `san-default` 사용하는 클래스를 `ontap-san-economy` 구성할 수 `san-dev` 있습니다.

모든 Kubernetes 작업자 노드에는 적절한 iSCSI 툴이 설치되어 있어야 합니다. 자세한 내용은 을 "작업자 노드를 준비합니다" 참조하십시오.

ONTAP 백엔드를 인증합니다

Trident는 ONTAP 백엔드를 인증하는 두 가지 모드를 제공합니다.

- 자격 증명 기반: 필요한 권한이 있는 ONTAP 사용자의 사용자 이름 및 암호입니다. ONTAP 버전과의 호환성을 최대화하려면 또는 과 `vsadmin` 같이 미리 정의된 보안 로그인 역할을 사용하는 것이 좋습니다 `admin`.
- 인증서 기반: Trident는 백엔드에 설치된 인증서를 사용하여 ONTAP 클러스터와 통신할 수도 있습니다. 이 경우 백엔드 정의에는 클라이언트 인증서, 키 및 사용할 경우 신뢰할 수 있는 CA 인증서의 Base64로 인코딩된 값이 있어야 합니다(권장).

자격 증명 기반 방법과 인증서 기반 방법 간에 이동하기 위해 기존 백엔드를 업데이트할 수 있습니다. 그러나 한 번에 하나의 인증 방법만 지원됩니다. 다른 인증 방법으로 전환하려면 백엔드 구성에서 기존 방법을 제거해야 합니다.

> 자격 증명과 인증서 * 를 모두 제공하려고 하면 구성 파일에 둘 이상의 인증 방법이 제공된다는 오류가 발생하여 백엔드 생성이 실패합니다.

자격 증명 기반 인증을 사용합니다

Trident은 ONTAP 백엔드와 통신하기 위해 SVM 범위/클러스터 범위 관리자에 대한 자격 증명이 필요합니다. 또는 `vsadmin` 과 같은 미리 정의된 표준 역할을 사용하는 것이 좋습니다 `admin`. 따라서 향후 Trident 릴리스에서 사용할 기능 API를 노출할 수 있는 향후 ONTAP 릴리즈와의 호환성이 보장됩니다. 사용자 지정 보안 로그인 역할을 만들어 Trident와 함께 사용할 수 있지만 권장하지는 않습니다.

백엔드 정의의 예는 다음과 같습니다.

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON을 참조하십시오

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

백엔드 정의는 자격 증명이 일반 텍스트로 저장되는 유일한 위치라는 점에 유의하십시오. 백엔드가 생성된 후 사용자 이름/암호는 Base64로 인코딩되어 Kubernetes 암호로 저장됩니다. 백엔드의 생성 또는 업데이트는 자격 증명에 대한 지식이 필요한 유일한 단계입니다. 따라서 Kubernetes/스토리지 관리자가 수행할 수 있는 관리 전용 작업입니다.

인증서 기반 인증을 사용합니다

신규 및 기존 백엔드는 인증서를 사용하여 ONTAP 백엔드와 통신할 수 있습니다. 백엔드 정의에는 세 가지 매개 변수가 필요합니다.

- clientCertificate: Base64로 인코딩된 클라이언트 인증서 값입니다.
- clientPrivateKey: Base64 - 연결된 개인 키의 인코딩된 값입니다.
- TrustedCACertificate: 신뢰할 수 있는 CA 인증서의 Base64 인코딩 값입니다. 신뢰할 수 있는 CA를 사용하는 경우 이 매개 변수를 제공해야 합니다. 신뢰할 수 있는 CA가 사용되지 않으면 이 작업을 무시할 수 있습니다.

일반적인 워크플로에는 다음 단계가 포함됩니다.

단계

1. 클라이언트 인증서 및 키를 생성합니다. 생성 시 CN(일반 이름)을 ONTAP 사용자로 설정하여 인증하십시오.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. 신뢰할 수 있는 CA 인증서를 ONTAP 클러스터에 추가합니다. 이는 스토리지 관리자가 이미 처리한 것일 수 있습니다. 트러스트된 CA가 사용되지 않으면 무시합니다.

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. ONTAP 클러스터에 클라이언트 인증서 및 키(1단계)를 설치합니다.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

이 명령을 실행하면 ONTAP에서 인증서 입력을 요청합니다. 1단계에서 생성된 `k8senv.pem` 파일의 내용을 붙여넣은 다음 `END`를 입력하여 설치를 완료하십시오.

4. ONTAP 보안 로그인 역할이 인증 방법을 지원하는지 `cert` 확인합니다.

```
security login create -user-or-group-name admin -application ontapi
-authentication-method cert
security login create -user-or-group-name admin -application http
-authentication-method cert
```

5. 생성된 인증서를 사용하여 인증을 테스트합니다. ONTAP 관리 LIF> 및 <SVM 이름>을 관리 LIF IP 및 SVM 이름으로 바꿉니다.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Base64로 인증서, 키 및 신뢰할 수 있는 CA 인증서를 인코딩합니다.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. 이전 단계에서 얻은 값을 사용하여 백엔드를 생성합니다.

```
cat cert-backend.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"trustedCACertificate": "QNFinfO...SiqOyN",
"storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       0 |
+------------+----------------+--------------------------------------
+--------+---------+
```

인증 방법을 업데이트하거나 자격 증명을 회전합니다

다른 인증 방법을 사용하거나 자격 증명을 회전하도록 기존 백엔드를 업데이트할 수 있습니다. 이렇게 하면 사용자 이름/암호를 사용하는 백엔드를 인증서를 사용하도록 업데이트할 수 있고 인증서를 사용하는 백엔드는 사용자 이름/암호 기반으로 업데이트할 수 있습니다. 이렇게 하려면 기존 인증 방법을 제거하고 새 인증 방법을 추가해야 합니다. 그런 다음 실행할 필수 매개 변수가 포함된 업데이트된 backend.json 파일을 `tridentctl backend update` 사용합니다.

```
cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

암호를 회전할 때 스토리지 관리자는 먼저 ONTAP에서 사용자의 암호를 업데이트해야 합니다. 그 다음에는 백엔드 업데이트가 있습니다. 인증서를 회전할 때 여러 인증서를 사용자에게 추가할 수 있습니다. 그런 다음 백엔드가 업데이트되어 새 인증서를 사용합니다. 그러면 ONTAP 클러스터에서 이전 인증서를 삭제할 수 있습니다.

백엔드를 업데이트해도 이미 생성된 볼륨에 대한 액세스가 중단되거나 이후에 생성된 볼륨 연결에 영향을 미치지 않습니다. 백엔드 업데이트에 성공하면 Trident가 ONTAP 백엔드와 통신하여 향후 볼륨 작업을 처리할 수 있음을 나타냅니다.

Trident에 대한 사용자 지정 ONTAP 역할을 생성합니다

Privileges에서 작업을 수행할 때 ONTAP 관리자 역할을 사용할 필요가 없도록 최소 Trident로 ONTAP 클러스터 역할을 생성할 수 있습니다. Trident 백엔드 구성에 사용자 이름을 포함하면 Trident은 사용자가 생성한 ONTAP 클러스터 역할을 사용하여 작업을 수행합니다.

Trident 사용자 지정 역할 생성에 대한 자세한 내용은 을 "Trident 사용자 지정 역할 생성기"참조하십시오.

ONTAP CLI 사용

1. 다음 명령을 사용하여 새 역할을 생성합니다.

```
security login role create <role_name\> -cmddirname "command" -access all
-vserver <svm_name\>
```

2. Trident 사용자에 대한 사용 이름 만들기:

```
security login create -username <user_name\> -application ontapi
-authmethod <password\> -role <name_of_role_in_step_1\> -vserver
<svm_name\> -comment "user_description"
```

3. 역할을 사용자에게 매핑:

```
security login modify username <user_name\> -vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

System Manager 사용

ONTAP System Manager에서 다음 단계를 수행하십시오.

1. * 사용자 지정 역할 생성 *:
   a. 클러스터 레벨에서 사용자 지정 역할을 생성하려면 * 클러스터 > 설정 * 을 선택합니다.

      SVM 레벨에서 사용자 지정 역할을 생성하려면 * 스토리지 > 스토리지 VM >> 설정 > 사용자 및 역할 * 을 선택합니다 `required SVM`.

   b. 사용자 및 역할 * 옆의 화살표 아이콘(*→ *)을 선택합니다.
   c. 역할 * 아래에서 * + 추가 * 를 선택합니다.
   d. 역할에 대한 규칙을 정의하고 * 저장 * 을 클릭합니다.
2. * 역할을 Trident 사용자에게 매핑 *: + * 사용자 및 역할 * 페이지에서 다음 단계를 수행하십시오.
   a. 사용자 * 아래에서 추가 아이콘 * + * 를 선택합니다.
   b. 필요한 사용자 이름을 선택하고 * Role * 에 대한 드롭다운 메뉴에서 역할을 선택합니다.
   c. 저장 * 을 클릭합니다.

자세한 내용은 다음 페이지를 참조하십시오.

- "ONTAP 관리를 위한 사용자 지정 역할" 또는 "사용자 지정 역할을 정의합니다"
- "역할 및 사용자 작업"

양방향 CHAP를 사용하여 연결을 인증합니다

Trident는 및 `ontap-san-economy` 드라이버에 대해 양방향 CHAP를 사용하여 iSCSI 세션을 인증할 수 `ontap-san` 있습니다. 이를 위해서는 백엔드 정의에서 옵션을 활성화해야 `useCHAP` 합니다. 로 `true` 설정하면 Trident는 SVM의 기본 이니시에이터 보안을 양방향 CHAP로 구성하고 백엔드 파일의 사용자 이름과 암호를 설정합니다. 양방향

CHAP를 사용하여 연결을 인증하는 것이 좋습니다. 다음 샘플 구성을 참조하십시오.

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

`useCHAP`매개 변수는 한 번만 구성할 수 있는 부울 옵션입니다. 기본적으로 false로 설정되어 있습니다. true 로 설정한 후에는 false 로 설정할 수 없습니다.

또한 `useCHAP=true` `chapInitiatorSecret` , `chapTargetInitiatorSecret`, `chapTargetUsername` 및 `chapUsername` 필드는 백엔드 정의에 포함되어야 합니다. 을 실행하여 백엔드를 생성한 후 암호를 변경할 수 `tridentctl update` 있습니다.

작동 방식

`useCHAP`true로 설정하면 스토리지 관리자가 `Trident`에 스토리지 백엔드에서 `CHAP`를 구성하도록 지시합니다. 여기에는 다음이 포함됩니다.

- SVM에서 CHAP 설정:
  - SVM의 기본 이니시에이터 보안 유형이 NONE(기본값 설정) * 이고 * 이미 존재하는 LUN이 볼륨에 없는 경우 Trident는 기본 보안 유형을 로 설정하고 CHAP 이니시에이터 및 타겟 사용자 이름과 암호를 구성합니다. `CHAP`
  - SVM에 LUN이 포함된 경우 Trident은 SVM에서 CHAP를 사용하도록 설정하지 않습니다. 따라서 SVM에 이미 있는 LUN에 대한 액세스가 제한되지 않습니다.
- CHAP 이니시에이터 및 타겟 사용자 이름과 암호를 구성합니다. 이러한 옵션은 백엔드 구성에 지정해야 합니다(위 참조).

백엔드가 생성된 후 Trident는 해당 `tridentbackend` CRD를 생성하고 CHAP 암호 및 사용자 이름을 Kubernetes 비밀로 저장합니다. 이 백엔드에서 Trident에 의해 생성된 모든 PVS가 CHAP를 통해 마운트되고 연결됩니다.

자격 증명을 회전하고 백엔드를 업데이트합니다

파일에서 CHAP 매개 변수를 업데이트하여 CHAP 자격 증명을 업데이트할 수 `backend.json` 있습니다. 이렇게 하려면 CHAP 암호를 업데이트하고 명령을 사용하여 이러한 변경 사항을 반영해야 `tridentctl update` 합니다.

백엔드의 CHAP 암호를 업데이트할 때 을 사용하여 백엔드를 업데이트해야 `tridentctl` 합니다. Trident은 이러한 변경 사항을 파악할 수 없으므로 CLI/ONTAP UI를 통해 스토리지 클러스터의 자격 증명을 업데이트하지 마십시오.

```
cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "password",
    "chapInitiatorSecret": "cl9qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+----------------+----------------+--------------------------------------
+--------+---------+
|   NAME         | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+----------------+----------------+--------------------------------------
+--------+---------+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+----------------+----------------+--------------------------------------
+--------+---------+
```

기존 연결은 영향을 받지 않고 그대로 유지되며 SVM의 Trident가 자격 증명을 업데이트하는 경우 계속 활성 상태로 유지됩니다. 새 연결은 업데이트된 자격 증명을 사용하며 기존 연결은 계속 활성 상태를 유지합니다. 기존 PVS를 연결 해제하고 다시 연결하면 업데이트된 자격 증명을 사용하게 됩니다.

ONTAP SAN 구성 옵션 및 예

Trident 설치 시 ONTAP SAN 드라이버를 생성하고 사용하는 방법에 대해 알아봅니다. 이 섹션에서는 백엔드 구성 예제 및 Backend를 StorageClasses에 매핑하는 방법에 대한 세부 정보를 제공합니다.

백엔드 구성 옵션

백엔드 구성 옵션은 다음 표를 참조하십시오.

매개 변수	설명	기본값
`version`		항상 1
`storageDriverName`	스토리지 드라이버의 이름입니다	`ontap-nas` `ontap-nas-economy`, , `ontap-nas-flexgroup`, , , `ontap-san` `ontap-san-economy`
`backendName`	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + dataLIF
`managementLIF`	클러스터 또는 SVM 관리 LIF의 IP 주소입니다. FQDN(정규화된 도메인 이름)을 지정할 수 있습니다. IPv6 플래그를 사용하여 Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다. IPv6 주소는 과 같이 대괄호로 정의해야 `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` 합니다. 원활한 MetroCluster 전환은 를 [mcc-best]참조하십시오.	“10.0.0.1”, “[2001:1234:ABCD::fee]”
`dataLIF`	프로토콜 LIF의 IP 주소입니다. IPv6 플래그를 사용하여 Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다. IPv6 주소는 과 같이 대괄호로 정의해야 `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` 합니다. **iSCSI**에 대해 지정하지 마십시오. Trident는 를 사용하여 "ONTAP 선택적 LUN 맵"다중 경로 세션을 설정하는 데 필요한 iSCI LIF를 검색합니다. 이 명시적으로 정의된 경우 경고가 `dataLIF` 생성됩니다. * MetroCluster의 경우 생략합니다. * 를 [mcc-best]참조하십시오.	SVM에서 파생됩니다
`svm`	사용할 스토리지 가상 머신 * MetroCluster에는 생략함 * 를 [mcc-best]참조하십시오.	SVM이 지정된 경우 파생됩니다 `managementLIF`
`useCHAP`	CHAP를 사용하여 ONTAP SAN 드라이버에 대한 iSCSI 인증 [Boolean]. 백엔드에 제공된 SVM에 대한 기본 인증으로 양방향 CHAP를 구성하고 사용하려면 Trident에 대해 으로 `true` 설정합니다. 자세한 내용은 을 "ONTAP SAN 드라이버를 사용하여 백엔드를 구성할 준비를 합니다" 참조하십시오.	`false`
`chapInitiatorSecret`	CHAP 이니시에이터 암호입니다. 필요한 경우 `useCHAP=true`	""
`labels`	볼륨에 적용할 임의의 JSON 형식 레이블 세트입니다	""
`chapTargetInitiatorSecret`	CHAP 타겟 이니시에이터 암호입니다. 필요한 경우 `useCHAP=true`	""
`chapUsername`	인바운드 사용자 이름입니다. 필요한 경우 `useCHAP=true`	""
`chapTargetUsername`	대상 사용자 이름입니다. 필요한 경우 `useCHAP=true`	""
`clientCertificate`	Base64 - 클라이언트 인증서의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""

매개 변수	설명	기본값
`clientPrivateKey`	Base64 - 클라이언트 개인 키의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
`trustedCACertificate`	Base64 - 신뢰할 수 있는 CA 인증서의 인코딩된 값입니다. 선택 사항. 인증서 기반 인증에 사용됩니다.	""
`username`	ONTAP 클러스터와 통신하는 데 필요한 사용자 이름입니다. 자격 증명 기반 인증에 사용됩니다.	""
`password`	ONTAP 클러스터와 통신하는 데 필요한 암호입니다. 자격 증명 기반 인증에 사용됩니다.	""
`svm`	사용할 스토리지 가상 머신입니다	SVM이 지정된 경우 파생됩니다 `managementLIF`
`storagePrefix`	SVM에서 새 볼륨을 프로비저닝할 때 사용되는 접두사 나중에 수정할 수 없습니다. 이 매개 변수를 업데이트하려면 새 백엔드를 생성해야 합니다.	`trident`
`aggregate`	프로비저닝을 위한 애그리게이트(선택 사항, SVM에 셋팅해야 하는 경우) 드라이버의 경우 `ontap-nas-flexgroup` 이 옵션은 무시됩니다. 할당되지 않은 경우 사용 가능한 애그리게이트를 사용하여 FlexGroup 볼륨을 프로비저닝할 수 있습니다. ⓘ SVM에서 Aggregate를 업데이트하면 Trident 컨트롤러를 다시 시작하지 않고도 SVM을 폴링하여 Trident에서 자동으로 업데이트됩니다. 볼륨을 프로비저닝하기 위해 Trident의 특정 애그리게이트를 구성한 경우, 애그리게이트의 이름을 바꾸거나 SVM에서 이동할 경우 SVM 애그리게이트를 폴링하는 동안 백엔드가 Trident에서 오류 상태로 전환됩니다. Aggregate를 SVM에 있는 Aggregate로 변경하거나 완전히 제거하여 백엔드를 다시 온라인 상태로 전환해야 합니다.	""
`limitAggregateUsage`	사용량이 이 비율을 초과하면 프로비저닝이 실패합니다. Amazon FSx for NetApp ONTAP 백엔드를 사용하는 경우 을 지정하지 `limitAggregateUsage`마십시오. 제공된 및 `vsadmin` 에는 `fsxadmin` 애그리게이트 사용량을 검색하고 Trident를 사용하여 제한하는 데 필요한 권한이 포함되어 있지 않습니다.	""(기본적으로 적용되지 않음)
`limitVolumeSize`	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다. 또한 LUN에 대해 관리하는 볼륨의 최대 크기를 제한합니다.	""(기본적으로 적용되지 않음)
`lunsPerFlexvol`	FlexVol당 최대 LUN 수는 범위[50, 200]에 있어야 합니다.	`100`

매개 변수	설명	기본값
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예제, {"api":false, "method":true} 문제 해결을 진행하고 자세한 로그 덤프가 필요한 경우가 아니면 사용하지 마십시오.	null
useREST	ONTAP REST API를 사용하는 부울 매개 변수입니다. useREST 로 설정된 true 경우 Trident는 ONTAP REST API를 사용하여 백엔드와 통신하고, 로 설정된 경우 false Trident는 ONTAP ZAPI 호출을 사용하여 백엔드와 통신합니다. 이 기능을 사용하려면 ONTAP 9.11.1 이상이 필요합니다. 또한 사용되는 ONTAP 로그인 역할에는 애플리케이션에 대한 액세스 권한이 있어야 ontap 합니다. 이는 미리 정의된 역할과 역할에 의해 충족됩니다. vsadmin cluster-admin Trident 24.06 릴리스 및 ONTAP 9.15.1 이상부터 는 useREST 기본적으로 로 설정되어 true 있으며 ONTAP ZAPI 호출을 사용하도록 로 false 변경합니다. useREST useREST NVMe/TCP에 대해 완전한 자격을 갖추고 있음	true ONTAP 9.15.1 이상, 그렇지 않은 경우 false.
sanType	iSCSI, nvme NVMe/TCP 또는 fcp FC(SCSI over Fibre Channel)를 선택할 때 iscsi 사용합니다. * 'FCP'(SCSI over FC)는 Trident 24.10 릴리스의 기술 미리 보기 기능입니다. *	iscsi 비어 있는 경우
formatOptions	`formatOptions`볼륨을 포맷할 때마다 적용되는 명령에 대한 명령줄 인수를 지정하는 데 `mkfs` 사용합니다. 이렇게 하면 기본 설정에 따라 볼륨을 포맷할 수 있습니다. 장치 경로를 제외하고 mkfs 명령 옵션과 비슷한 formatOptions를 지정해야 합니다. 예: "-E NODEARD" • ontap-san`및 `ontap-san-economy 드라이버에서만 지원됩니다.*	
limitVolumePoolSize	ONTAP-SAN-Economy 백엔드에서 LUN을 사용할 때 요청될 수 있는 최대 FlexVol 크기입니다.	""(기본적으로 적용되지 않음)
denyNewVolumePools	백엔드가 LUN을 포함하도록 새 FlexVol 볼륨을 생성하지 못하도록 ontap-san-economy 제한합니다. 기존 FlexVol만 새 PVS 프로비저닝에 사용됩니다.	

포맷옵션 사용에 대한 권장 사항

Trident에서는 다음 옵션을 사용하여 서식 프로세스를 신속하게 수행할 것을 권장합니다.

- -E nobiscard: *
- 유지, mkfs 시간에 블록 삭제를 시도하지 마십시오(처음에는 솔리드 스테이트 디바이스 및 스파스/씬 프로비저닝된 스토리지에서 블록 삭제가 유용함). 이 옵션은 사용되지 않는 "-K" 옵션을 대체하며 모든 파일 시스템(xfs, ext3 및

ext4)에 적용됩니다.

볼륨 프로비저닝을 위한 백엔드 구성 옵션

구성 섹션에서 이러한 옵션을 사용하여 기본 프로비저닝을 제어할 수 `defaults` 있습니다. 예를 들어, 아래 구성 예제를 참조하십시오.

매개 변수	설명	기본값
`spaceAllocation`	LUN에 대한 공간 할당	"참"
`spaceReserve`	공간 예약 모드, "없음"(씬) 또는 "볼륨"(일반)	"없음"
`snapshotPolicy`	사용할 스냅샷 정책입니다	"없음"
`qosPolicy`	생성된 볼륨에 할당할 QoS 정책 그룹입니다. 스토리지 풀/백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. Trident에서 QoS 정책 그룹을 사용하려면 ONTAP 9 .8 이상이 필요합니다. 비공유 QoS 정책 그룹을 사용하고 정책 그룹이 각 구성 요소에 개별적으로 적용되도록 해야 합니다. 공유 QoS 정책 그룹은 모든 워크로드의 총 처리량에 대한 제한을 적용합니다.	""
`adaptiveQosPolicy`	생성된 볼륨에 할당할 적응형 QoS 정책 그룹입니다. 스토리지 풀/백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다	""
`snapshotReserve`	스냅숏용으로 예약된 볼륨의 백분율입니다	"없음"인 경우 "0", 그렇지 않은 경우 `snapshotPolicy`"
`splitOnClone`	생성 시 상위 클론에서 클론을 분할합니다	"거짓"
`encryption`	새 볼륨에서 NetApp 볼륨 암호화(NVE)를 활성화하고, 기본값은 로 설정합니다. `false` 이 옵션을 사용하려면 NVE 라이센스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Trident에서 프로비저닝된 모든 볼륨은 NAE가 사용됩니다. 자세한 내용은 다음을 "Trident가 NVE 및 NAE와 작동하는 방법"참조하십시오.	"거짓"
`luksEncryption`	LUKS 암호화를 사용합니다. 을 "LUKS(Linux Unified Key Setup) 사용"참조하십시오. NVMe/TCP에 대해서는 LUKS 암호화가 지원되지 않습니다.	""
`securityStyle`	새로운 볼륨에 대한 보안 스타일	`unix`
`tieringPolicy`	"없음"을 사용하는 계층화 정책	ONTAP 9.5 SVM-DR 이전 구성의 경우 "스냅샷 전용
`nameTemplate`	사용자 지정 볼륨 이름을 생성하는 템플릿입니다.	""

볼륨 프로비저닝의 예

다음은 기본값이 정의된 예입니다.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

드라이버를 사용하여 생성된 모든 볼륨의 경우 `ontap-san` Trident는 LUN 메타데이터를 수용하기 위해 FlexVol에 10%의 용량을 추가합니다. LUN은 사용자가 PVC에서 요청하는 정확한 크기로 프로비저닝됩니다. Trident는 FlexVol에 10%를 추가합니다(ONTAP에서 사용 가능한 크기로 표시됨). 이제 사용자가 요청한 가용 용량을 얻을 수 있습니다. 또한 이 변경으로 인해 사용 가능한 공간이 완전히 활용되지 않는 한 LUN이 읽기 전용이 되는 것을 방지할 수 있습니다. ONTAP-SAN-경제에는 적용되지 않습니다.

을 정의하는 백엔드의 경우 `snapshotReserve` Trident는 다음과 같이 볼륨 크기를 계산합니다.

```
Total volume size = [(PVC requested size) / (1 - (snapshotReserve
percentage) / 100)] * 1.1
```

1.1은 LUN 메타데이터를 수용하기 위해 FlexVol에 추가되는 10%의 Trident입니다. = 5%, PVC 요청 = 5GiB의 경우 `snapshotReserve` 총 볼륨 크기는 5.79GiB이고 사용 가능한 크기는 5.5GiB입니다. 이 `volume show` 명령은 다음 예제와 유사한 결과를 표시해야 합니다.

현재 기존 볼륨에 대해 새 계산을 사용하는 유일한 방법은 크기 조정입니다.

최소 구성의 예

다음 예에서는 대부분의 매개 변수를 기본값으로 두는 기본 구성을 보여 줍니다. 이는 백엔드를 정의하는 가장 쉬운 방법입니다.

NetApp ONTAP on Trident와 함께 Amazon FSx를 사용하는 경우 IP 주소 대신 LIF에 대한 DNS 이름을 지정하는 것이 좋습니다.

ONTAP SAN의 예

드라이버를 사용하는 기본 구성입니다. `ontap-san`

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

ONTAP SAN 경제 예

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

1. 예

전환 및 전환 중에 백엔드 정의를 수동으로 업데이트할 필요가 없도록 백엔드를 구성할 수 "SVM 복제 및 복구"있습니다.

원활한 스위치오버 및 스위치백의 경우 및 `svm` 매개 변수를 사용하여 SVM을 지정하고 `managementLIF` 생략합니다. `dataLIF` 예를 들면 다음과 같습니다.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

인증서 기반 인증의 예

이 기본 구성 예에서는 `clientCertificate` `clientPrivateKey` 및 `trustedCACertificate` (트러스트된 CA를 사용하는 경우 선택 사항)가 에 채워지고 `backend.json` 클라이언트 인증서, 개인 키 및 트러스트된 CA 인증서의 base64로 인코딩된 값을 각각 가져옵니다.

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

양방향 **CHAP** 예

이 예에서는 로 설정된 `true` 백엔드를 `useCHAP` 생성합니다.

ONTAP SAN CHAP의 예

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

ONTAP SAN 이코노미 **CHAP**의 예

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

NVMe/TCP 예

ONTAP 백엔드에서 NVMe로 구성된 SVM이 있어야 합니다. NVMe/TCP에 대한 기본 백엔드 구성입니다.

```
---
version: 1
backendName: NVMeBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nvme
username: vsadmin
password: password
sanType: nvme
useREST: true
```

nameTemplate이 포함된 백엔드 구성 예

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults: {
    "nameTemplate":
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.R
equestName}}"
},
"labels": {"cluster": "ClusterA", "PVC":
"{{.volume.Namespace}}_{{.volume.RequestName}}"}
```

<code> ONTAP-SAN-Economy </code> 드라이버에 대한 옵션 예

```
version: 1
storageDriverName: ontap-san-economy
managementLIF: ''
svm: svm1
username: ''
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: "-E nodiscard"
```

가상 풀의 백엔드 예

이러한 샘플 백엔드 정의 파일에서 특정 기본값은 모든 스토리지 풀에 대해 설정(예: `spaceReserve` 없음, `spaceAllocation` 거짓, 거짓 `encryption`) 가상 풀은 스토리지 섹션에 정의됩니다.

Trident는 "Comments" 필드에 프로비저닝 레이블을 설정합니다. FlexVol에 주석이 설정됩니다. Trident는 프로비저닝 시 가상 풀에 있는 모든 레이블을 스토리지 볼륨에 복제합니다. 편의를 위해 스토리지 관리자는 가상 풀 및 그룹 볼륨별로 레이블을 레이블별로 정의할 수 있습니다.

이 예에서 일부 스토리지 풀은 자체, `spaceAllocation` 및 `encryption` 값을 설정하고 `spaceReserve` 일부 풀은 기본값을 재정의합니다.

ONTAP SAN의 예

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: 'false'
  encryption: 'false'
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
- labels:
    protection: gold
    creditpoints: '40000'
  zone: us_east_1a
  defaults:
    spaceAllocation: 'true'
    encryption: 'true'
    adaptiveQosPolicy: adaptive-extreme
- labels:
    protection: silver
    creditpoints: '20000'
  zone: us_east_1b
  defaults:
    spaceAllocation: 'false'
    encryption: 'true'
    qosPolicy: premium
- labels:
    protection: bronze
    creditpoints: '5000'
  zone: us_east_1c
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'
```

ONTAP SAN 경제 예

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: 'false'
  encryption: 'false'
labels:
  store: san_economy_store
region: us_east_1
storage:
- labels:
    app: oracledb
    cost: '30'
  zone: us_east_1a
  defaults:
    spaceAllocation: 'true'
    encryption: 'true'
- labels:
    app: postgresdb
    cost: '20'
  zone: us_east_1b
  defaults:
    spaceAllocation: 'false'
    encryption: 'true'
- labels:
    app: mysqldb
    cost: '10'
  zone: us_east_1c
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'
- labels:
    department: legal
    creditpoints: '5000'
```

```
    zone: us_east_1c
    defaults:
      spaceAllocation: 'true'
      encryption: 'false'
```

NVMe/TCP 예

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: 'false'
  encryption: 'true'
storage:
- labels:
    app: testApp
    cost: '20'
  defaults:
    spaceAllocation: 'false'
    encryption: 'false'
```

백엔드를 **StorageClasses**에 매핑합니다

다음 StorageClass 정의는 을 가상 풀의 백엔드 예참조하십시오. 각 StorageClass 는 필드를 사용하여 `parameters.selector` 볼륨을 호스팅하는 데 사용할 수 있는 가상 풀을 호출합니다. 선택한 가상 풀에 볼륨이 정의되어 있습니다.

- `protection-gold`StorageClass는 백엔드의 첫 번째 가상 풀에 매핑됩니다. `ontap-san` 골드 레벨 보호 기능을 제공하는 유일한 풀입니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- protection-not-gold`StorageClass는 백엔드의 두 번째 및 세 번째 가상 풀에 매핑됩니다. `ontap-san 금 이외의 보호 수준을 제공하는 유일한 풀입니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- app-mysqldb`StorageClass는 백엔드의 세 번째 가상 풀에 매핑됩니다. `ontap-san-economy mysqldb 유형 앱에 대한 스토리지 풀 구성을 제공하는 유일한 풀입니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- protection-silver-creditpoints-20k`StorageClass는 백엔드의 두 번째 가상 풀에 매핑됩니다. `ontap-san 실버 레벨 보호 및 20,000포인트 적립을 제공하는 유일한 풀입니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- `creditpoints-5k`StorageClass는 백엔드의 세 번째 가상 풀과 백엔드의 네 번째 가상 `ontap-san-economy` 풀에 매핑됩니다. `ontap-san` 5000 크레딧 포인트를 보유한 유일한 풀 서비스입니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

- `my-test-app-sc`StorageClass는 를 사용하여 드라이버의 `sanType: nvme` 가상 풀에 `ontap-san` 매핑됩니다. `testAPP` 이것은 유일한 풀 제안입니다. `testApp`

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"
```

Trident는 어떤 가상 풀이 선택되었는지 결정하고 스토리지 요구 사항이 충족되는지 확인합니다.