



# Docker용 Trident

## Trident

NetApp  
July 01, 2026

# 목차

Docker용 Trident	1
배포를 위한 사전 요구 사항	1
요구 사항을 확인하십시오	1
NVMe 도구	3
FC 도구	4
Trident 구축	6
Docker 관리형 플러그인 방식(버전 1.13/17.03 이상)	6
기존 방식(버전 1.12 이하)	8
시스템 시작 시 Trident를 시작합니다	9
Trident 업그레이드 또는 제거	10
업그레이드	10
제거	12
볼륨 작업	12
볼륨 생성	12
볼륨 제거	13
볼륨 복제	13
외부에서 생성된 볼륨에 액세스	15
드라이버별 볼륨 옵션	15
로그 수집	20
문제 해결을 위해 로그를 수집합니다	20
일반적인 문제 해결 팁	21
여러 Trident 인스턴스를 관리합니다	21
Docker 관리형 플러그인(버전 1.13/17.03 이상) 단계	21
기존 버전(1.12 이하)의 단계	22
스토리지 구성 옵션	22
전역 구성 옵션	22
ONTAP 구성	23
Element 소프트웨어 구성	31
알려진 문제 및 제한 사항	33
Trident Docker Volume Plugin을 이전 버전에서 20.10 이상 버전으로 업그레이드하면 no such file or directory 오류와 함께 업그레이드가 실패합니다	33
볼륨 이름은 최소 2자 이상이어야 합니다	34
Docker Swarm에는 Trident가 모든 스토리지 및 드라이버 조합에서 이를 지원하지 못하도록 하는 특정 동작이 있습니다	34
FlexGroup이 프로비저닝되는 동안 ONTAP는 두 번째 FlexGroup이 프로비저닝 중인 FlexGroup과 하나 이상의 애그리게이트를 공유하는 경우 두 번째 FlexGroup을 프로비저닝하지 않습니다	34

# Docker용 Trident

## 배포를 위한 사전 요구 사항

Trident를 배포하기 전에 호스트에 필요한 프로토콜 사전 요구 사항을 설치하고 구성해야 합니다.

### 요구 사항을 확인하십시오

- 배포가 모든 "요구사항"를 충족하는지 확인하십시오.
- 지원되는 버전의 Docker가 설치되어 있는지 확인하십시오. Docker 버전이 최신 버전이 아닌 경우 "[설치하거나 업데이트하십시오](#)".

```
docker --version
```

- 프로토콜 사전 요구 사항이 호스트에 설치 및 구성되었는지 확인하십시오.

### NFS 도구

운영 체제에 맞는 명령을 사용하여 NFS 툴을 설치합니다.

#### RHEL 8+

```
sudo yum install -y nfs-utils
```

#### Ubuntu

```
sudo apt-get install -y nfs-common
```



NFS 도구를 설치한 후 워커 노드를 재부팅하여 컨테이너에 볼륨을 연결할 때 발생하는 오류를 방지하십시오.

### iSCSI 도구

운영 체제에 맞는 명령을 사용하여 iSCSI 툴을 설치합니다.

## RHEL 8+

1. 다음 시스템 패키지를 설치하십시오.

```
sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils device-  
mapper-multipath
```

2. iscsi-initiator-utils 버전이 6.2.0.874-2.el7 이상인지 확인하십시오.

```
rpm -q iscsi-initiator-utils
```

3. 스캔을 수동으로 설정:

```
sudo sed -i 's/^\(node.session.scan\) .*/\1 = manual/'  
/etc/iscsi/iscsid.conf
```

4. 다중 경로 지정 활성화:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



etc/multipath.conf`에 `find\_multipaths no`이(가) `defaults 아래에 포함되어 있는지 확인하십시오.

5. iscsid 및 `multipathd`가 실행 중인지 확인하십시오.

```
sudo systemctl enable --now iscsid multipathd
```

6. 활성화 및 시작 iscsi:

```
sudo systemctl enable --now iscsi
```

## Ubuntu

1. 다음 시스템 패키지를 설치하십시오.

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools  
scsitools
```

2. open-iscsi 버전이 2.0.874-5ubuntu2.10 이상(bionic의 경우) 또는 2.0.874-7.1ubuntu6.1 이상(focal의 경우)인지 확인하십시오.

```
dpkg -l open-iscsi
```

### 3. 스캔을 수동으로 설정:

```
sudo sed -i 's/^\(node.session.scan\).*\/\1 = manual/'  
/etc/iscsi/iscsid.conf
```

### 4. 다중 경로 지정 활성화:

```
sudo tee /etc/multipath.conf <<-EOF  
defaults {  
    user_friendly_names yes  
    find_multipaths no  
}  
EOF  
sudo systemctl enable --now multipath-tools.service  
sudo service multipath-tools restart
```



etc/multipath.conf`에 `find\_multipaths no`이(가) `defaults` 아래에 포함되어 있는지 확인하십시오.

### 5. open-iscsi 및 `multipath-tools`가 활성화되어 실행 중인지 확인하십시오.

```
sudo systemctl status multipath-tools  
sudo systemctl enable --now open-iscsi.service  
sudo systemctl status open-iscsi
```

## NVMe 도구

사용 중인 운영 체제에 맞는 명령을 사용하여 NVMe 툴을 설치하십시오.



- NVMe에는 RHEL 9 이상이 필요합니다.
- Kubernetes 노드의 커널 버전이 너무 오래되었거나 해당 커널 버전에 맞는 NVMe 패키지를 사용할 수 없는 경우 노드의 커널 버전을 NVMe 패키지가 포함된 버전으로 업데이트해야 할 수 있습니다.

## RHEL 9

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

## Ubuntu

```
sudo apt install nvme-cli
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

## FC 도구

운영 체제에 맞는 명령을 사용하여 FC 툴을 설치합니다.

- FC PV를 사용하는 RHEL/Red Hat Enterprise Linux CoreOS(RHCOS) 워커 노드를 사용할 때, 인라인 공간 재확보를 수행하려면 `discard` StorageClass에서 `mountOption`을 지정하십시오. 을 참조하십시오. "[Red Hat 문서](#)"

## RHEL 8+

1. 다음 시스템 패키지를 설치하십시오.

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. 다중 경로 지정 활성화:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



etc/multipath.conf`에 `find\_multipaths no`이(가) `defaults` 아래에 포함되어 있는지 확인하십시오.

3. `multipathd`이(가) 실행 중인지 확인하십시오.

```
sudo systemctl enable --now multipathd
```

## Ubuntu

1. 다음 시스템 패키지를 설치하십시오.

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. 다중 경로 지정 활성화:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```



etc/multipath.conf`에 `find\_multipaths no`이(가) `defaults` 아래에 포함되어 있는지 확인하십시오.

3. `multipath-tools`이(가) 활성화되어 실행 중인지 확인하십시오.

```
sudo systemctl status multipath-tools
```

# Trident 구축

Trident for Docker는 NetApp 스토리지 플랫폼을 위한 Docker 에코시스템과의 직접적인 통합을 제공합니다. 스토리지 플랫폼에서 Docker 호스트로 스토리지 리소스를 프로비저닝하고 관리할 수 있도록 지원하며, 향후 다른 플랫폼을 추가할 수 있는 프레임워크도 제공합니다.

Trident의 여러 인스턴스를 동일한 호스트에서 동시에 실행할 수 있습니다. 이를 통해 여러 스토리지 시스템 및 스토리지 유형에 동시에 연결할 수 있으며 Docker 볼륨에 사용되는 스토리지를 사용자 지정할 수 있습니다.

필요한 것

"[배포를 위한 사전 요구 사항](#)"을 참조하십시오. 사전 요구 사항이 충족되었는지 확인한 후 Trident를 배포할 준비가 된 것입니다.

## Docker 관리형 플러그인 방식(버전 1.13/17.03 이상)



시작하기 전에

기존 데몬 방식으로 Docker 1.13/17.03 이전 버전의 Trident를 사용한 경우 관리형 플러그인 방식을 사용하기 전에 Trident 프로세스를 중지하고 Docker 데몬을 다시 시작해야 합니다.

1. 실행 중인 모든 인스턴스를 중지합니다.

```
pkill /usr/local/bin/netappdvp
pkill /usr/local/bin/trident
```

2. Docker를 재시작합니다.

```
systemctl restart docker
```

3. Docker Engine 17.03(새 1.13) 이상이 설치되어 있는지 확인하십시오.

```
docker --version
```

사용 중인 버전이 오래된 경우 "[설치 또는 설치 업데이트](#)".

단계

1. 구성 파일을 생성하고 다음과 같이 옵션을 지정하십시오.

- config: 기본 파일 이름은 config.json`이지만 `config 옵션에 파일 이름을 지정하여 원하는 이름을 사용할 수 있습니다. 구성 파일은 호스트 시스템의 /etc/netappdvp 디렉터리에 있어야 합니다.
- log-level: 로깅 수준을 지정합니다((debug, info, warn, error, fatal). 기본값은 `info`입니다.
- debug: 디버그 로깅 활성화 여부를 지정합니다. 기본값은 false입니다. true로 설정하면 log-level을 재정의합니다.
  - i. 구성 파일의 위치를 생성합니다.

```
sudo mkdir -p /etc/netappdvp
```

ii. 구성 파일을 생성합니다.

```
cat << EOF > /etc/netappdvp/config.json
```

```
{  
  "version": 1,  
  "storageDriverName": "ontap-nas",  
  "managementLIF": "10.0.0.1",  
  "dataLIF": "10.0.0.2",  
  "svm": "svm_nfs",  
  "username": "vsadmin",  
  "password": "password",  
  "aggregate": "aggr1"  
}  
EOF
```

2. 관리형 플러그인 시스템을 사용하여 Trident를 시작합니다. ``<version>``를 사용 중인 플러그인 버전(xxx.xx.x)으로 교체합니다.

```
docker plugin install --grant-all-permissions --alias netapp  
netapp/trident-plugin:<version> config=myConfigFile.json
```

3. 구성된 시스템에서 스토리지를 사용하려면 Trident를 사용하기 시작하십시오.

a. "firstVolume"이라는 이름의 볼륨을 생성합니다.

```
docker volume create -d netapp --name firstVolume
```

b. 컨테이너 시작 시 기본 볼륨 생성:

```
docker run --rm -it --volume-driver netapp --volume  
secondVolume:/my_vol alpine ash
```

c. "firstVolume" 볼륨을 제거하세요:

```
docker volume rm firstVolume
```

## 기존 방식(버전 1.12 이하)

시작하기 전에

1. Docker 버전 1.10 이상이 있는지 확인하십시오.

```
docker --version
```

사용 중인 버전이 오래된 경우 설치를 업데이트하십시오.

```
curl -fsSL https://get.docker.com/ | sh
```

또는 "[배포 지침을 따르십시오](#)".

2. 시스템에 NFS 및/또는 iSCSI가 구성되어 있는지 확인하십시오.

단계

1. NetApp Docker 볼륨 플러그인을 설치하고 구성하십시오.
  - a. 애플리케이션을 다운로드하고 압축을 풉니다.

```
wget
https://github.com/NetApp/trident/releases/download/10.0/trident-
installer-26.02.0.tar.gz
tar xzf trident-installer-26.02.0.tar.gz
```

- b. bin 경로의 위치로 이동합니다.

```
sudo mv trident-installer/extras/bin/trident /usr/local/bin/
sudo chown root:root /usr/local/bin/trident
sudo chmod 755 /usr/local/bin/trident
```

- c. 구성 파일의 위치를 생성합니다.

```
sudo mkdir -p /etc/netappdvp
```

- d. 구성 파일을 생성합니다.

```
cat << EOF > /etc/netappdvp/ontap-nas.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1"
}
EOF
```

2. 바이너리를 배치하고 구성 파일을 생성한 후 원하는 구성 파일을 사용하여 Trident 데몬을 시작합니다.

```
sudo trident --config=/etc/netappdvp/ontap-nas.json
```



별도로 지정하지 않으면 볼륨 드라이버의 기본 이름은 "netapp"입니다.

데몬이 시작되면 Docker CLI 인터페이스를 사용하여 볼륨을 생성하고 관리할 수 있습니다.

3. 볼륨 생성:

```
docker volume create -d netapp --name trident_1
```

4. 컨테이너를 시작할 때 Docker 볼륨을 프로비저닝합니다.

```
docker run --rm -it --volume-driver netapp --volume trident_2:/my_vol
alpine ash
```

5. Docker 볼륨 제거:

```
docker volume rm trident_1
```

```
docker volume rm trident_2
```

## 시스템 시작 시 Trident를 시작합니다

systemd 기반 시스템용 샘플 유닛 파일은 Git 저장소의 `contrib/trident.service.example`에서 찾을 수 있습니다. RHEL에서 파일을 사용하려면 다음을 수행합니다.

1. 파일을 올바른 위치에 복사합니다.

여러 인스턴스를 실행하는 경우 유닛 파일에 고유한 이름을 사용해야 합니다.

```
cp contrib/trident.service.example
/usr/lib/systemd/system/trident.service
```

2. 파일을 편집하고 설명(2행)을 드라이버 이름과 일치하도록 변경하고 구성 파일 경로(9행)를 환경에 맞게 변경하십시오.
3. 변경 사항을 수집하도록 `systemd`를 다시 로드합니다.

```
systemctl daemon-reload
```

4. 서비스를 활성화합니다.

이 이름은 `/usr/lib/systemd/system` 디렉터리에서 파일 이름을 지정한 내용에 따라 달라집니다.

```
systemctl enable trident
```

5. 서비스를 시작합니다.

```
systemctl start trident
```

6. 상태를 확인합니다.

```
systemctl status trident
```



유닛 파일을 수정할 때마다 변경 사항을 인식하도록 `systemctl daemon-reload` 명령을 실행합니다.

## Trident 업그레이드 또는 제거

Trident for Docker를 안전하게 업그레이드할 수 있으며, 사용 중인 볼륨에는 아무런 영향을 미치지 않습니다. 업그레이드 과정 중에는 `docker volume` 플러그인에 대한 명령이 잠시 실패하고, 플러그인이 다시 실행될 때까지 애플리케이션에서 볼륨을 마운트할 수 없는 시간이 발생할 수 있습니다. 대부분의 경우 이 시간은 몇 초에 불과합니다.

### 업그레이드

Docker용 Trident를 업그레이드하려면 아래 단계를 수행하십시오.

## 단계

1. 기존 볼륨을 나열합니다.

```
docker volume ls
DRIVER          VOLUME NAME
netapp:latest   my_volume
```

2. 플러그인을 비활성화합니다.

```
docker plugin disable -f netapp:latest
docker plugin ls
ID              NAME          DESCRIPTION
ENABLED
7067f39a5df5   netapp:latest nDVP - NetApp Docker Volume
Plugin  false
```

3. 플러그인 업그레이드:

```
docker plugin upgrade --skip-remote-check --grant-all-permissions
netapp:latest netapp/trident-plugin:21.07
```



Trident의 18.01 릴리스는 nDVP를 대체합니다. netapp/ndvp-plugin 이미지에서 netapp/trident-plugin 이미지로 직접 업그레이드해야 합니다.

4. 플러그인을 활성화합니다.

```
docker plugin enable netapp:latest
```

5. 플러그인이 활성화되어 있는지 확인합니다.

```
docker plugin ls
ID              NAME          DESCRIPTION
ENABLED
7067f39a5df5   netapp:latest Trident - NetApp Docker Volume
Plugin  true
```

6. 볼륨이 표시되는지 확인합니다.

```
docker volume ls
DRIVER          VOLUME NAME
netapp:latest   my_volume
```



이전 버전의 Trident(20.10 이전)에서 Trident 20.10 이상으로 업그레이드하는 경우 오류가 발생할 수 있습니다. 자세한 내용은 "[알려진 문제](#)"를 참조하십시오. 오류가 발생하는 경우 먼저 플러그인을 비활성화한 다음 플러그인을 제거하고 추가 구성 매개변수를 전달하여 필요한 Trident 버전을 설치해야 합니다: `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

## 제거

Docker용 Trident를 제거하려면 아래 단계를 수행하십시오.

### 단계

1. 플러그인이 생성한 볼륨을 모두 제거하십시오.
2. 플러그인을 비활성화합니다.

```
docker plugin disable netapp:latest
docker plugin ls
ID                NAME                DESCRIPTION
ENABLED
7067f39a5df5     netapp:latest       nDVP - NetApp Docker Volume
Plugin           false
```

3. 플러그인을 제거합니다.

```
docker plugin rm netapp:latest
```

## 볼륨 작업

필요에 따라 Trident 드라이버 이름을 지정하여 표준 `docker volume` 명령을 사용하면 볼륨을 쉽게 생성, 복제 및 제거할 수 있습니다.

### 볼륨 생성

- 기본 이름을 사용하여 드라이버로 볼륨 생성:

```
docker volume create -d netapp --name firstVolume
```

- 특정 Trident 인스턴스를 사용하여 볼륨 생성:

```
docker volume create -d ntap_bronze --name bronzeVolume
```



"옵션"을 지정하지 않으면 드라이버의 기본값이 사용됩니다.

- 기본 볼륨 크기를 재정의합니다. 드라이버를 사용하여 20GiB 볼륨을 생성하는 방법은 다음 예제를 참조하십시오.

```
docker volume create -d netapp --name my_vol --opt size=20G
```



볼륨 크기는 정수 값과 선택적으로 단위를 포함하는 문자열로 표현됩니다(예: 10G, 20GB, 3TiB). 단위가 지정되지 않은 경우 기본값은 G입니다. 크기 단위는 2의 거듭제곱(B, KiB, MiB, GiB, TiB) 또는 10의 거듭제곱(B, KB, MB, GB, TB)으로 표현할 수 있습니다. 간략한 단위 표기에는 2의 거듭제곱이 사용됩니다(G = GiB, T = TiB 등).

## 볼륨 제거

- 다른 Docker 볼륨과 마찬가지로 볼륨을 제거하십시오:

```
docker volume rm firstVolume
```



solidfire-san 드라이버를 사용할 때 위의 예에서는 볼륨을 삭제하고 제거합니다.

Docker용 Trident를 업그레이드하려면 아래 단계를 수행하십시오.

## 볼륨 복제

`ontap-nas`, `ontap-san` 및 `solidfire-san` 스토리지 드라이버를 사용하는 경우 Trident는 볼륨을 복제할 수 있습니다. `ontap-nas-flexgroup` 또는 `ontap-nas-economy` 드라이버를 사용할 때는 복제가 지원되지 않습니다. 기존 볼륨에서 새 볼륨을 생성하면 새 스냅샷이 생성됩니다.

- 볼륨을 검사하여 스냅샷을 열거합니다.

```
docker volume inspect <volume_name>
```

- 기존 볼륨에서 새 볼륨을 생성합니다. 이렇게 하면 새 스냅샷이 생성됩니다.

```
docker volume create -d <driver_name> --name <new_name> -o from
=<source_docker_volume>
```

- 볼륨의 기존 스냅샷에서 새 볼륨을 생성합니다. 새 스냅샷이 생성되지 않습니다.

```
docker volume create -d <driver_name> --name <new_name> -o from
=<source_docker_volume> -o fromSnapshot=<source_snap_name>
```

예

```
docker volume inspect firstVolume
```

```
[
  {
    "Driver": "ontap-nas",
    "Labels": null,
    "Mountpoint": "/var/lib/docker-volumes/ontap-
nas/netappdvp_firstVolume",
    "Name": "firstVolume",
    "Options": {},
    "Scope": "global",
    "Status": {
      "Snapshots": [
        {
          "Created": "2017-02-10T19:05:00Z",
          "Name": "hourly.2017-02-10_1505"
        }
      ]
    }
  }
]
```

```
docker volume create -d ontap-nas --name clonedVolume -o from=firstVolume
clonedVolume
```

```
docker volume rm clonedVolume
```

```
docker volume create -d ontap-nas --name volFromSnap -o from=firstVolume
-o fromSnapshot=hourly.2017-02-10_1505
volFromSnap
```

```
docker volume rm volFromSnap
```

## 외부에서 생성된 볼륨에 액세스

컨테이너는 Trident를 사용하여 외부에서 생성된 블록 디바이스(또는 해당 클론)에 액세스할 수 있습니다. 단, 해당 디바이스에 파티션이 없고 파일 시스템이 Trident에서 지원되는 경우에만 가능합니다(예: ext4`로 포맷된 `/dev/sdc1 디바이스는 Trident를 통해 액세스할 수 없음).

## 드라이버별 볼륨 옵션

각 스토리지 드라이버는 볼륨 생성 시 지정할 수 있는 다양한 옵션 세트를 제공하여 결과물을 사용자 지정할 수 있습니다. 구성된 스토리지 시스템에 적용되는 옵션은 아래를 참조하십시오.

볼륨 생성 작업 중에 이러한 옵션을 사용하는 것은 간단합니다. CLI 작업 중에 -o 연산자를 사용하여 옵션과 값을 제공합니다. 이러한 값은 JSON 구성 파일의 동등한 값을 재정의합니다.

## ONTAP 볼륨 옵션

NFS, iSCSI 및 FC에 대한 볼륨 생성 옵션은 다음과 같습니다.

옵션	설명
size	볼륨의 크기로, 기본값은 1GiB입니다.
spaceReserve	볼륨을 씬 프로비저닝 또는 씩 프로비저닝할지 결정하며, 기본값은 씬 프로비저닝입니다. 유효한 값은 none(씬 프로비저닝)과 volume(씩 프로비저닝)입니다.
snapshotPolicy	이렇게 하면 스냅샷 정책이 원하는 값으로 설정됩니다. 기본값은 none`이며, 볼륨에 대해 스냅샷이 자동으로 생성되지 않음을 의미합니다. 스토리지 관리자가 수정하지 않는 한, 모든 ONTAP 시스템에는 "default"라는 정책이 존재하며, 이 정책은 6개의 시간별, 2개의 일별, 2개의 주별 스냅샷을 생성하고 보존합니다. 스냅샷에 보존된 데이터는 볼륨 내 모든 디렉터리의 `.snapshot 디렉터리를 탐색하여 복구할 수 있습니다.
snapshotReserve	이렇게 하면 스냅샷 예약 공간이 원하는 비율로 설정됩니다. 기본값은 값이 없음을 의미하며, snapshotPolicy를 선택한 경우 ONTAP가 snapshotReserve(일반적으로 5%)를 선택하고, snapshotPolicy가 none인 경우 0%를 선택합니다. 모든 ONTAP 백엔드에 대해 구성 파일에서 기본 snapshotReserve 값을 설정할 수 있으며, ontap-nas-economy를 제외한 모든 ONTAP 백엔드의 볼륨 생성 옵션으로 사용할 수 있습니다.

옵션	설명
splitOnClone	볼륨을 복제할 때 이 옵션을 선택하면 ONTAP가 복제본을 상위 볼륨에서 즉시 분할합니다. 기본값은 `false`입니다. 볼륨 복제를 위한 일부 사용 사례는 스토리지 효율성을 위한 기회가 거의 없기 때문에 생성 즉시 복제본을 상위 볼륨에서 분할하는 것이 가장 좋습니다. 예를 들어, 빈 데이터베이스를 복제하면 시간은 크게 절약할 수 있지만 스토리지 절약 효과는 거의 없으므로 복제본을 즉시 분할하는 것이 가장 좋습니다.
encryption	새 볼륨에서 NetApp Volume Encryption(NVE)을 활성화합니다. 기본값은 `false`입니다. 이 옵션을 사용하려면 클러스터에서 NVE 라이선스가 있고 활성화되어 있어야 합니다.  백엔드에서 NAE가 활성화된 경우 Trident에서 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다.  자세한 내용은 다음을 참조하십시오. " <a href="#">Trident가 NVE 및 NAE와 작동하는 방식</a> "
tieringPolicy	볼륨에 사용할 티어링 정책을 설정합니다. 이 정책은 데이터가 비활성화(콜드)될 때 클라우드 티어로 이동할지 여부를 결정합니다.

다음 추가 옵션은 NFS \*전용\*입니다.

옵션	설명
unixPermissions	이 설정은 볼륨 자체에 대한 권한 설정을 제어합니다. 기본적으로 권한은 `---rwxr-xr-x` 또는 숫자 표기법으로 0755로 설정되며, `root`가 소유자입니다. 텍스트 또는 숫자 형식 모두 사용 가능합니다.
snapshotDir	이 값을 `true`로 설정하면 `.snapshot` 디렉터리가 볼륨에 액세스하는 클라이언트에 표시됩니다. 기본값은 `false`이며, 이는 `.snapshot` 디렉터리의 가시성이 기본적으로 비활성화되어 있음을 의미합니다. 일부 이미지(예: 공식 MySQL 이미지)는 `.snapshot` 디렉터리가 표시될 때 예상대로 작동하지 않습니다.
exportPolicy	볼륨에 사용할 내보내기 정책을 설정합니다. 기본값은 `default`입니다.
securityStyle	볼륨 접근에 사용할 보안 스타일을 설정합니다. 기본값은 `unix`입니다. 유효한 값은 `unix` 및 `mixed`입니다.

다음 추가 옵션은 iSCSI \*전용\*입니다.

옵션	설명
fileSystemType	iSCSI 볼륨 포맷에 사용되는 파일 시스템을 설정합니다. 기본값은 ext4입니다. 유효한 값은 `ext3, ext4` 및 `xfs`입니다.
spaceAllocation	이 값을 `false`로 설정하면 LUN의 공간 할당 기능이 비활성화됩니다. 기본값은 `true`이며, ONTAP이 볼륨의 공간이 부족해져 해당 볼륨의 LUN에서 쓰기 작업을 더 이상 수행할 수 없을 때 호스트에 알림을 보낸다는 의미입니다. 또한 이 옵션을 사용하면 호스트에서 데이터를 삭제할 때 ONTAP이 자동으로 공간을 회수할 수 있습니다.

예

아래 예를 참조하십시오.

- 10 GiB 볼륨을 생성합니다.

```
docker volume create -d netapp --name demo -o size=10G -o encryption=true
```

- 스냅샷이 있는 100GiB 볼륨 생성:

```
docker volume create -d netapp --name demo -o size=100G -o snapshotPolicy=default -o snapshotReserve=10
```

- setUID 비트가 활성화된 볼륨을 생성합니다.

```
docker volume create -d netapp --name demo -o unixPermissions=4755
```

최소 볼륨 크기는 20MiB입니다.

스냅샷 예약량이 지정되지 않고 스냅샷 정책이 `none`인 경우 Trident는 0%의 스냅샷 예약량을 사용합니다.

- 스냅샷 정책 및 스냅샷 예약이 없는 볼륨을 생성합니다.

```
docker volume create -d netapp --name my_vol --opt snapshotPolicy=none
```

- 스냅샷 정책이 없고 사용자 지정 스냅샷 예약률이 10%인 볼륨을 생성합니다.

```
docker volume create -d netapp --name my_vol --opt snapshotPolicy=none --opt snapshotReserve=10
```

- 스냅샷 정책과 10%의 사용자 지정 스냅샷 예약 공간을 사용하여 볼륨을 생성합니다.

```
docker volume create -d netapp --name my_vol --opt
snapshotPolicy=myPolicy --opt snapshotReserve=10
```

- 스냅샷 정책이 적용된 볼륨을 생성하고 ONTAP의 기본 스냅샷 예약값(일반적으로 5%)을 수락하십시오.

```
docker volume create -d netapp --name my_vol --opt
snapshotPolicy=myPolicy
```

## Element 소프트웨어 볼륨 옵션

Element 소프트웨어 옵션은 볼륨과 관련된 크기 및 서비스 품질(QoS) 정책을 표시합니다. 볼륨을 생성할 때 해당 볼륨과 연결된 QoS 정책은 `-o type=service_level` 명명법을 사용하여 지정됩니다.

Element 드라이버를 사용하여 QoS 서비스 수준을 정의하는 첫 번째 단계는 하나 이상의 유형을 생성하고 구성 파일에서 이름과 연결된 최소, 최대 및 버스트 IOPS를 지정하는 것입니다.

기타 Element 소프트웨어 볼륨 생성 옵션은 다음과 같습니다.

옵션	설명
size	볼륨의 크기는 기본값으로 1GiB 또는 구성 항목 ... "기본값": {"size": "5G"}.
blocksize	512 또는 4096 중 하나를 사용하십시오. 기본값은 512 또는 구성 항목 DefaultBlockSize입니다.

예

다음은 QoS 정의가 포함된 샘플 구성 파일입니다.

```

{
  "Types": [
    {
      "Type": "Bronze",
      "Qos": {
        "minIOPS": 1000,
        "maxIOPS": 2000,
        "burstIOPS": 4000
      }
    },
    {
      "Type": "Silver",
      "Qos": {
        "minIOPS": 4000,
        "maxIOPS": 6000,
        "burstIOPS": 8000
      }
    },
    {
      "Type": "Gold",
      "Qos": {
        "minIOPS": 6000,
        "maxIOPS": 8000,
        "burstIOPS": 10000
      }
    }
  ]
}

```

위 구성에는 Bronze, Silver, Gold라는 세 가지 정책 정의가 있습니다. 이 이름은 임의로 지정한 것입니다.

- 10GiB Gold 볼륨을 생성합니다.

```
docker volume create -d solidfire --name sfGold -o type=Gold -o size=10G
```

- 100 GiB Bronze 볼륨 생성:

```
docker volume create -d solidfire --name sfBronze -o type=Bronze -o
size=100G
```

# 로그 수집

문제 해결에 도움이 되도록 로그를 수집할 수 있습니다. 로그 수집 방법은 Docker 플러그인을 실행하는 방식에 따라 다릅니다.

## 문제 해결을 위해 로그를 수집합니다

단계

1. 권장되는 관리형 플러그인 방식(즉, `docker plugin` 명령을 사용)을 사용하여 Trident를 실행하는 경우 다음과 같이 확인합니다.

```
docker plugin ls
```

ID	NAME	DESCRIPTION
ENABLED		
4fb97d2b956b	netapp:latest	nDVP - NetApp Docker Volume
Plugin	false	
journalctl -u docker   grep 4fb97d2b956b		

기본 로깅 수준을 사용하면 대부분의 문제를 진단할 수 있습니다. 만약 그것으로 충분하지 않다고 판단되면 디버그 로깅을 활성화할 수 있습니다.

2. 디버그 로깅을 활성화하려면 디버그 로깅이 활성화된 플러그인을 설치하십시오.

```
docker plugin install netapp/trident-plugin:<version> --alias <alias>  
debug=true
```

또는 플러그인이 이미 설치된 경우 디버그 로깅을 활성화하십시오.

```
docker plugin disable <plugin>
```

```
docker plugin set <plugin> debug=true
```

```
docker plugin enable <plugin>
```

3. 호스트에서 바이너리 자체를 실행하는 경우 로그는 호스트의 `/var/log/netappdvp` 디렉터리에서 확인할 수 있습니다. 디버그 로깅을 활성화하려면 플러그인을 실행할 때 `-debug`를 지정하십시오.

## 일반적인 문제 해결 팁

- 새로운 사용자들이 가장 흔히 겪는 문제는 플러그인 초기화를 방해하는 잘못된 구성입니다. 이러한 경우 플러그인을 설치하거나 활성화하려고 할 때 다음과 같은 메시지가 표시될 수 있습니다:

```
Error response from daemon: dial unix /run/docker/plugins/<id>/netapp.sock:
connect: no such file or directory
```

이는 플러그인이 시작되지 않았음을 의미합니다. 다행히 이 플러그인은 포괄적인 로깅 기능을 갖추고 있어 발생할 수 있는 대부분의 문제를 진단하는 데 도움이 될 것입니다.

- PV를 컨테이너에 마운트하는 데 문제가 있는 경우, `rpcbind`가 설치되어 실행 중인지 확인하십시오. 호스트 OS에 맞는 패키지 관리자를 사용하여 `rpcbind`가 실행 중인지 확인하십시오. `rpcbind` 서비스의 상태는 `systemctl status rpcbind` 또는 이와 동등한 명령을 실행하여 확인할 수 있습니다.

## 여러 Trident 인스턴스를 관리합니다

여러 스토리지 구성을 동시에 사용하려면 Trident의 여러 인스턴스가 필요합니다. 여러 인스턴스의 핵심은 컨테이너화된 플러그인의 `--alias` 옵션을 사용하거나 호스트에서 Trident를 인스턴스화할 때 `--volume-driver` 옵션을 사용하여 각 인스턴스에 서로 다른 이름을 지정하는 것입니다.

### Docker 관리형 플러그인(버전 1.13/17.03 이상) 단계

1. 별칭과 구성 파일을 지정하여 첫 번째 인스턴스를 시작합니다.

```
docker plugin install --grant-all-permissions --alias silver
netapp/trident-plugin:21.07 config=silver.json
```

2. 다른 별칭과 구성 파일을 지정하여 두 번째 인스턴스를 시작합니다.

```
docker plugin install --grant-all-permissions --alias gold
netapp/trident-plugin:21.07 config=gold.json
```

3. 별칭을 드라이버 이름으로 지정하여 볼륨을 생성합니다.

예를 들어 금 볼륨의 경우:

```
docker volume create -d gold --name ntapGold
```

예를 들어, silver 볼륨의 경우:

```
docker volume create -d silver --name ntapSilver
```

## 기존 버전(1.12 이하)의 단계

1. 사용자 지정 드라이버 ID를 사용하는 NFS 구성으로 플러그인을 실행하세요.

```
sudo trident --volume-driver=netapp-nas --config=/path/to/config-nfs.json
```

2. 사용자 지정 드라이버 ID를 사용하는 iSCSI 구성으로 플러그인을 실행하십시오.

```
sudo trident --volume-driver=netapp-san --config=/path/to/config-iscsi.json
```

3. 각 드라이버 인스턴스에 대한 Docker 볼륨 프로비저닝:

예를 들어 NFS의 경우:

```
docker volume create -d netapp-nas --name my_nfs_vol
```

예를 들어 iSCSI의 경우:

```
docker volume create -d netapp-san --name my_iscsi_vol
```

## 스토리지 구성 옵션

Trident 구성에 사용할 수 있는 구성 옵션을 확인하세요.

### 전역 구성 옵션

이러한 구성 옵션은 사용 중인 스토리지 플랫폼과 관계없이 모든 Trident 구성에 적용됩니다.

옵션	설명	예
version	구성 파일 버전 번호	1
storageDriverName	스토리지 드라이버 이름	ontap-nas, ontap-san, ontap-nas-economy, ontap-nas-flexgroup, solidfire-san
storagePrefix	볼륨 이름에 사용할 수 있는 선택적 접두사입니다. 기본값: netappdvp_.	staging_

옵션	설명	예
limitVolumeSize	볼륨 크기에 대한 선택적 제한입니다. 기본값: "" (적용되지 않음)	10g



Element 백엔드에는 storagePrefix(기본값 포함)을 사용하지 마십시오. 기본적으로 solidfire-san 드라이버는 이 설정을 무시하고 접두사를 사용하지 않습니다. NetApp에서는 Docker 볼륨 매핑에 특정 tenantID를 사용하거나 이름 변환이 사용된 경우 Docker 버전, 드라이버 정보 및 Docker의 원시 이름으로 채워진 속성 데이터를 사용할 것을 권장합니다.

볼륨을 생성할 때마다 설정을 일일이 지정할 필요 없이 기본 옵션을 사용할 수 있습니다. size 옵션은 모든 컨트롤러 유형에서 사용할 수 있습니다. 기본 볼륨 크기를 설정하는 방법에 대한 예시는 ONTAP 구성 섹션을 참조하십시오.

옵션	설명	예
size	새 볼륨의 선택적 기본 크기입니다. 기본값: 1G	10G

## ONTAP 구성

위의 전역 구성 값 외에도 ONTAP를 사용할 때 다음과 같은 최상위 옵션을 사용할 수 있습니다.

옵션	설명	예
managementLIF	ONTAP 관리 LIF의 IP 주소입니다. 정규화된 도메인 이름(FQDN)을 지정할 수 있습니다.	10.0.0.1
dataLIF	프로토콜 LIF의 IP 주소입니다.  <b>ONTAP NAS</b> 드라이버: NetApp에서는 `dataLIF`를 지정하는 것을 권장합니다. 지정하지 않으면 Trident는 SVM에서 dataLIF를 가져옵니다. NFS 마운트 작업에 사용할 정규화된 도메인 이름 (FQDN)을 지정하여 여러 dataLIF에 걸쳐 로드 밸런싱을 수행하는 라운드 로빈 DNS를 구성할 수 있습니다.  <b>ONTAP SAN</b> 드라이버: iSCSI 또는 FC에 대해 지정하지 마십시오. Trident는 "ONTAP 선택적 LUN 매핑"를 사용하여 다중 경로 세션을 설정하는 데 필요한 iSCSI 또는 FC LIF를 검색합니다. `dataLIF`가 명시적으로 정의된 경우 경고가 생성됩니다.	10.0.0.2

옵션	설명	예
svm	사용할 스토리지 가상 머신(관리 LIF가 클러스터 LIF인 경우 필수)	svm_nfs
username	스토리지 장치에 연결할 사용자 이름	vsadmin
password	스토리지 장치에 연결하기 위한 암호	secret
aggregate	프로비저닝용 애그리게이트(선택 사항, 설정된 경우 SVM에 할당해야 함). ontap-nas-flexgroup 드라이버의 경우 이 옵션은 무시됩니다. SVM에 할당된 모든 애그리게이트는 FlexGroup 볼륨을 프로비저닝하는 데 사용됩니다.	aggr1
limitAggregateUsage	선택 사항이며, 사용량이 이 비율을 초과하면 프로비저닝이 실패합니다	75%
nfsMountOptions	NFS 마운트 옵션을 세밀하게 제어할 수 있습니다. 기본값은 "-o nfsvers=3"입니다. * ontap-nas 및 ontap-nas-economy 드라이버에서만 사용 가능합니다*. <a href="#">"NFS 호스트 구성 정보는 여기를 참조하십시오"</a>	-o nfsvers=4
igroupName	Trident는 노드별 igroups `netappdvp`로 생성 및 관리합니다.  이 값은 변경하거나 생략할 수 없습니다.  • ontap-san 드라이버에만 사용할 수 있습니다*.	netappdvp
limitVolumeSize	최대 요청 가능 볼륨 크기.	300g
qtreesPerFlexvol	FlexVol당 최대 qtree 수는 [50, 300] 범위 내에 있어야 하며 기본값은 200입니다.  • ontap-nas-economy 드라이버의 경우, 이 옵션을 사용하면 FlexVol당 최대 qtree 수를 사용자 지정할 수 있습니다*.	300

옵션	설명	예
sanType	* ontap-san 드라이버에서만 지원됩니다.* iSCSI의 경우 iscsi, NVMe/TCP의 경우 nvme 또는 Fibre Channel(FC)을 통한 SCSI의 경우 `fc`를 선택하는 데 사용합니다.	iscsi 공백인 경우
limitVolumePoolSize	* ontap-san-economy 및 ontap-san-economy 드라이버에서만 지원됩니다.* ONTAP ontap-nas-economy 및 ontap-san-economy 드라이버에서 FlexVol 크기를 제한합니다.	300g

볼륨을 생성할 때마다 지정할 필요가 없도록 기본 옵션을 사용할 수 있습니다.

옵션	설명	예
spaceReserve	공간 예약 모드; none(싹 프로비저닝) 또는 volume(싹 프로비저닝)	none
snapshotPolicy	사용할 스냅샷 정책, 기본값은 none	none
snapshotReserve	스냅샷 예약 비율, 기본값은 ""이며 ONTAP 기본값을 수락합니다	10
splitOnClone	생성 시 상위 클론에서 클론을 분할합니다. 기본값은 false	false
encryption	새 볼륨에서 NetApp Volume Encryption(NVE)을 활성화합니다. 기본값은 `false`입니다. 이 옵션을 사용하려면 클러스터에서 NVE 라이선스가 있고 활성화되어 있어야 합니다.  백엔드에서 NAE가 활성화된 경우 Trident에서 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다.  자세한 내용은 다음을 참조하십시오. " <a href="#">Trident가 NVE 및 NAE와 작동하는 방식</a> "	true
unixPermissions	프로비저닝된 NFS 볼륨에 대한 NAS 옵션, 기본값 777	777

옵션	설명	예
snapshotDir	`.snapshot` 디렉토리에 액세스하기 위한 NAS 옵션.	<p>`.snapshot` 디렉터리에 대한 액세스를 활성화 또는 비활성화합니다.</p> <p>유효값: true, false. 기본 동작은 ONTAP 및 Trident 버전에 따라 다를 수 있습니다. 일관된 동작을 보장하려면 명시적으로 설정하십시오.</p>
exportPolicy	사용할 NFS 익스포트 정책에 대한 NAS 옵션, 기본값은 default	default
securityStyle	<p>프로비저닝된 NFS 볼륨에 액세스하기 위한 NAS 옵션입니다.</p> <p>NFS는 mixed 및 unix 보안 스타일을 지원합니다. 기본값은 `unix`입니다.</p>	unix
fileSystemType	파일 시스템 유형을 선택하는 SAN 옵션, 기본값은 ext4	xfx
tieringPolicy	사용할 계층화 정책, 기본값은 `none`입니다.	none
skipRecoveryQueue	볼륨 삭제 시 스토리지의 복구 큐를 건너뛰고 볼륨을 즉시 삭제합니다.	``

## 확장 옵션

`ontap-nas` 및 `ontap-san` 드라이버는 각 Docker 볼륨에 대해 ONTAP FlexVol을 생성합니다. ONTAP는 클러스터 노드당 최대 1,000개의 FlexVols를 지원하며 클러스터 최대 12,000개의 FlexVol 볼륨을 지원합니다. Docker 볼륨 요구 사항이 이 제한 내에 있는 경우 `ontap-nas` 드라이버는 Docker 볼륨 단위 스냅샷 및 클론 생성과 같은 FlexVols에서 제공하는 추가 기능으로 인해 선호되는 NAS 솔루션입니다.

FlexVol 제한으로 수용할 수 있는 것보다 더 많은 Docker 볼륨이 필요한 경우 ontap-nas-economy 또는 ontap-san-economy 드라이버를 선택하십시오.

`ontap-nas-economy` 드라이버는 자동으로 관리되는 FlexVol 볼륨 풀 내에서 ONTAP Qtree로 Docker 볼륨을 생성합니다. Qtree는 클러스터 노드당 최대 100,000개, 클러스터당 최대 2,400,000개까지 훨씬 더 큰 확장성을 제공하지만 일부 기능이 제한됩니다. `ontap-nas-economy` 드라이버는 Docker 볼륨 단위의 스냅샷 또는 클론 생성을 지원하지 않습니다.



ontap-nas-economy 드라이버는 Docker Swarm이 여러 노드에 걸쳐 볼륨 생성을 조정하지 않기 때문에 현재 Docker Swarm에서 지원되지 않습니다.

`ontap-san-economy` 드라이버는 자동으로 관리되는 FlexVol 볼륨의 공유 풀 내에서 Docker 볼륨을 ONTAP LUN으로 생성합니다. 이러한 방식으로 각 FlexVol은 하나의 LUN으로만 제한되지 않으며 SAN 워크로드에 더 나은 확장성을 제공합니다. 스토리지 어레이에 따라 ONTAP는 클러스터당 최대 16384개의 LUN을 지원합니다. 볼륨이 내부적으로 LUN이기 때문에 이 드라이버는 Docker 볼륨 단위의 스냅샷 및 클론 생성을 지원합니다.

`ontap-nas-flexgroup` 드라이버를 선택하여 수십억 개의 파일로 페타바이트 범위까지 확장할 수 있는 단일 볼륨에 대한 병렬 처리를 향상시키십시오. FlexGroups의 이상적인 사용 사례로는 AI/ML/DL, 빅 데이터 및 분석, 소프트웨어 빌드, 스트리밍, 파일 리포지토리 등이 있습니다. Trident는 FlexGroup 볼륨을 프로비저닝할 때 SVM에 할당된 모든 애그리게이트를 사용합니다. Trident의 FlexGroup 지원에는 다음과 같은 고려 사항도 있습니다.

- ONTAP 버전 9.2 이상이 필요합니다.
- 현재(작성 시점 기준) FlexGroups는 NFS v3만 지원합니다.
- SVM에 대해 64비트 NFSv3 식별자를 활성화하는 것이 좋습니다.
- 최소 권장 FlexGroup 멤버/볼륨 크기는 100GiB입니다.
- FlexGroup 볼륨은 복제가 지원되지 않습니다.

FlexGroups 및 FlexGroups에 적합한 워크로드에 대한 자세한 내용은 "[NetApp FlexGroup 볼륨 모범 사례 및 구현 가이드](#)"를 참조하십시오.

고급 기능과 대규모 확장성을 동일한 환경에서 구현하려면 Docker Volume Plugin의 여러 인스턴스를 실행할 수 있습니다. 하나는 `ontap-nas`을 사용하고 다른 하나는 `ontap-nas-economy`을 사용합니다.

### Trident용 사용자 지정 ONTAP 역할

Trident에서 작업을 수행하기 위해 ONTAP 관리자 역할을 사용할 필요가 없도록 최소 권한으로 ONTAP 클러스터 역할을 생성할 수 있습니다. Trident 백엔드 구성에 사용자 이름을 포함하면 Trident는 생성한 ONTAP 클러스터 역할을 사용하여 작업을 수행합니다.

Trident 사용자 지정 역할 생성에 대한 자세한 내용은 "[Trident 사용자 지정 역할 생성기](#)"를 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 사용자의 사용자 이름을 생성합니다.

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

3. 사용자에게 역할 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 수준에서 사용자 지정 역할을 생성하려면 \* Cluster > Settings \* 를 선택합니다.

(또는) SVM 수준에서 사용자 지정 역할을 생성하려면 \*스토리지 > 스토리지 VM > required SVM>  
설정 > 사용자 및 역할\*을 선택하십시오.

- b. 사용자 및 역할 옆에 있는 화살표 아이콘(→)을 선택합니다.
- c. 역할 에서 +추가 를 선택합니다.
- d. 역할에 대한 규칙을 정의하고 \*저장\*을 클릭합니다.

2. **Trident** 사용자에게 역할 매핑: + 사용자 및 역할 페이지에서 다음 단계를 수행합니다.

- a. 사용자 아래에서 추가 아이콘 \*\*를 선택합니다.
- b. 필요한 사용자 이름을 선택하고 역할 드롭다운 메뉴에서 역할을 선택합니다.
- c. \*저장\*을 클릭합니다.

자세한 내용은 다음 페이지를 참조하십시오.

- ["ONTAP 관리를 위한 사용자 지정 역할"](#) 또는 ["사용자 지정 역할 정의"](#)
- ["역할 및 사용자 작업"](#)

## ONTAP 구성 파일의 예

### <code>ontap-nas</code> 드라이버의 NFS 예

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "defaults": {
    "size": "10G",
    "spaceReserve": "none",
    "exportPolicy": "default"
  }
}
```

### <code>ontap-nas-flexgroup</code> 드라이버의 NFS 예

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "defaults": {
    "size": "100G",
    "spaceReserve": "none",
    "exportPolicy": "default"
  }
}
```

**<code>ontap-nas-economy</code>** 드라이버의 NFS 예

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1"
}
```

**<code>ontap-san</code>** 드라이버의 iSCSI 예

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "igroupName": "netappdvp"
}
```

**<code>ontap-san-economy</code>** 드라이버의 NFS 예

```
{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi_eco",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "igroupName": "netappdvp"
}
```

### <code>ontap-san</code> 드라이버의 NVMe/TCP 예

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

### <code>ontap-san</code> 드라이버의 FC 기반 SCSI 예

```
{
  "version": 1,
  "backendName": "ontap-san-backend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "sanType": "fcp",
  "svm": "trident_svm",
  "username": "vsadmin",
  "password": "password",
  "useREST": true
}
```

## Element 소프트웨어 구성

전역 구성 값 외에도 Element 소프트웨어(NetApp HCI/SolidFire)를 사용할 때 다음과 같은 옵션을 사용할 수 있습니다.

옵션	설명	예
Endpoint	https://<login>:<password>@<mvip>/json-rpc/<element-version>	https://admin:admin@192.168.160.3/json-rpc/8.0
SVIP	iSCSI IP 주소 및 포트	10.0.0.7:3260
TenantName	사용할 SolidFire 테넌트(찾을 수 없으면 생성됨)	docker

옵션	설명	예
InitiatorIFace	iSCSI 트래픽을 기본이 아닌 인터페이스로 제한할 때 인터페이스를 지정합니다	default
Types	QoS 사양	아래 예를 참조하십시오
LegacyNamePrefix	업그레이드된 Trident 설치에 사용되는 접두사입니다. 1.3.2 이전 버전의 Trident를 사용했고 기존 볼륨으로 업그레이드를 수행하는 경우 volume-name 방법을 통해 매핑된 이전 볼륨에 액세스하려면 이 값을 설정해야 합니다.	netappdvp-

``solidfire-san`` 드라이버는 Docker Swarm을 지원하지 않습니다.

#### Element 소프트웨어 구성 파일의 예

```

{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://admin:admin@192.168.160.3/json-rpc/8.0",
  "SVIP": "10.0.0.7:3260",
  "TenantName": "docker",
  "InitiatorIFace": "default",
  "Types": [
    {
      "Type": "Bronze",
      "Qos": {
        "minIOPS": 1000,
        "maxIOPS": 2000,
        "burstIOPS": 4000
      }
    },
    {
      "Type": "Silver",
      "Qos": {
        "minIOPS": 4000,
        "maxIOPS": 6000,
        "burstIOPS": 8000
      }
    },
    {
      "Type": "Gold",
      "Qos": {
        "minIOPS": 6000,
        "maxIOPS": 8000,
        "burstIOPS": 10000
      }
    }
  ]
}

```

## 알려진 문제 및 제한 사항

Docker와 함께 Trident를 사용할 때 발생하는 알려진 문제 및 제한 사항에 대한 정보를 찾아보세요.

**Trident Docker Volume Plugin**을 이전 버전에서 **20.10** 이상 버전으로 업그레이드하면 **no such file or directory** 오류와 함께 업그레이드가 실패합니다.

해결 방법

1. 플러그인을 비활성화합니다.

```
docker plugin disable -f netapp:latest
```

2. 플러그인을 제거합니다.

```
docker plugin rm -f netapp:latest
```

3. 추가 config 매개변수를 제공하여 플러그인을 다시 설치하십시오.

```
docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant  
-all-permissions config=config.json
```

볼륨 이름은 최소 **2자** 이상이어야 합니다.



이는 Docker 클라이언트의 제한 사항입니다. 클라이언트는 한 글자로 된 이름을 Windows 경로로 해석합니다 "[버그 25773](#)을 참조하십시오".

**Docker Swarm**에는 **Trident**가 모든 스토리지 및 드라이버 조합에서 이를 지원하지 못하도록 하는 특정 동작이 있습니다.

- Docker Swarm은 현재 고유 볼륨 식별자로 볼륨 ID 대신 볼륨 이름을 사용합니다.
- 볼륨 요청은 Swarm 클러스터의 각 노드로 동시에 전송됩니다.
- 볼륨 플러그인(Trident 포함)은 Swarm 클러스터의 각 노드에서 독립적으로 실행되어야 합니다. ONTAP 작동 방식과 `ontap-nas` 및 `ontap-san` 드라이버의 기능 특성상, 이러한 제약 조건 내에서 작동할 수 있는 플러그인은 해당 드라이버뿐입니다.

나머지 드라이버는 경쟁 조건과 같은 문제에 직면할 수 있으며, 이로 인해 명확한 "승자" 없이 단일 요청에 대해 다수의 볼륨이 생성될 수 있습니다. 예를 들어 Element에는 볼륨이 동일한 이름을 갖지만 서로 다른 ID를 가질 수 있도록 하는 기능이 있습니다.

NetApp은 Docker 팀에 피드백을 제공했지만 향후 조치에 대한 표시는 없습니다.

**FlexGroup**이 프로비저닝되는 동안 **ONTAP**는 두 번째 **FlexGroup**이 프로비저닝 중인 **FlexGroup**과 하나 이상의 애그리게이트를 공유하는 경우 두 번째 **FlexGroup**을 프로비저닝하지 않습니다.

## 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.