Docker를 위한 Astra Trident

Astra Trident

NetApp
November 14, 2025

This PDF was generated from https://docs.netapp.com/ko-kr/trident-2301/trident-docker/prereqs-docker.html on November 14, 2025. Always check docs.netapp.com for the latest.

목차

Docker를 위한 Astra Trident . . . 1
배포를 위한 사전 요구 사항 . . . 1
요구 사항을 확인합니다 . . . 1
Astra Trident 구축 . . . 4
Docker 관리형 플러그인 방법(버전 1.13/17.03 이상) . . . 4
기존 방법(버전 1.12 이하) . . . 6
시스템 시작 시 Astra Trident를 시작합니다 . . . 7
Astra Trident를 업그레이드하거나 제거합니다 . . . 8
업그레이드 . . . 8
설치 제거 . . . 10
볼륨 작업 . . . 10
볼륨을 생성합니다 . . . 10
볼륨을 제거합니다 . . . 11
볼륨의 클론을 생성합니다 . . . 11
외부에서 생성된 볼륨에 액세스합니다 . . . 12
드라이버별 볼륨 옵션 . . . 12
로그를 수집합니다 . . . 18
문제 해결을 위해 로그를 수집합니다 . . . 18
일반적인 문제 해결 팁 . . . 18
여러 Astra Trident 인스턴스를 관리합니다 . . . 19
Docker 관리 플러그인 단계(버전 1.13/17.03 이상) . . . 19
기존(버전 1.12 이하) 단계 . . . 19
스토리지 구성 옵션 . . . 20
글로벌 구성 옵션 . . . 20
ONTAP 구성 . . . 21
Element 소프트웨어 구성 . . . 26
알려진 문제 및 제한 사항 . . . 28
이전 버전에서 Trident Docker 볼륨 플러그인을 20.10 이상으로 업그레이드하면 해당 파일 또는 디렉터리 오류가 없는 업그레이드 오류가 발생합니다. . . . 28
볼륨 이름은 최소 2자 이상이어야 합니다. . . . 29
Docker Swarm에는 Astra Trident가 모든 스토리지 및 드라이버 조합에서 이를 지원하지 않는 특정 동작이 있습니다. . . . 29
FlexGroup를 프로비저닝하고 있는 경우, 두 번째 FlexGroup에 프로비저닝되는 FlexGroup와 공통되는 하나 이상의 애그리게이트가 있는 경우 ONTAP는 두 번째 FlexGroup를 프로비저닝하지 않습니다. . . . 29

Docker를 위한 Astra Trident

배포를 위한 사전 요구 사항

Astra Trident를 구축하기 전에 호스트에 필수 프로토콜 사전 요구 사항을 설치하고 구성해야 합니다.

요구 사항을 확인합니다

- 배포가 모든 를 충족하는지 확인합니다 "요구 사항".
- 지원되는 버전의 Docker가 설치되어 있는지 확인합니다. Docker 버전이 오래되었으면 "설치 또는 업데이트합니다".

```
docker --version
```

- 프로토콜 사전 요구 사항이 호스트에 설치 및 구성되어 있는지 확인합니다.

프로토콜	운영 체제	명령
NFS 를 참조하십시오	RHEL 8+	`sudo yum install -y nfs-utils`
NFS 를 참조하십시오	우분투	`sudo apt-get install -y nfs-common`

프로토콜	운영 체제	명령

iSCSI	RHEL 8+	1. 다음 시스템 패키지를 설치합니다.
		
		`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 확인합니다.
		
		`rpm -q iscsi-initiator-utils`
		
		3. 스캔을 수동으로 설정합니다.
		
		`sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/' /etc/iscsi/iscsid.conf`
		
		4. 다중 경로 설정:
		
		`sudo mpathconf --enable --with_multipathd y --find_multipaths n`
		
		ⓘ 확인합니다 `etc/multipath.conf` 포함 `find_multipaths no` 에서 `defaults`.
		
		5. 확인하십시오 `iscsid` 및 `multipathd` 실행 중:
		
		`sudo systemctl enable --now iscsid multipathd`
		
		6. 활성화 및 시작 `iscsi`:
		
		`sudo systemctl enable --now iscsi`

프로토콜	운영 체제	명령

iSCSI	우분투	1. 다음 시스템 패키지를 설치합니다.
		
		`sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools scsitools`
		
		2. open-iscsi 버전이 2.0.874-5ubuntu2.10 이상(bionic) 또는 2.0.874-7.1uubuttu6.1 이상(focal)인지 확인합니다.
		
		`dpkg -l open-iscsi`
		
		3. 스캔을 수동으로 설정합니다.
		
		`sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/' /etc/iscsi/iscsid.conf`
		
		4. 다중 경로 설정:
		
		`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`
		
		ⓘ 확인합니다 `etc/multipath.conf` 포함 `find_multipaths no` 에서 `defaults`.
		
		5. 확인하십시오 `open-iscsi` 및 `multipath-tools` 활성화 및 실행:
		
		`sudo systemctl status multipath-tools`
		`sudo systemctl enable --now open-iscsi.service`
		`sudo systemctl status`

Astra Trident 구축

Docker를 위한 Astra Trident는 NetApp의 스토리지 플랫폼을 위한 Docker 에코시스템과 직접 통합됩니다. 또한 향후 추가 플랫폼을 추가할 수 있도록 스토리지 플랫폼에서 Docker 호스트에 이르는 스토리지 리소스의 프로비저닝 및 관리를 지원합니다. `open-iscsi`

Astra 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

를 참조하십시오 "배포를 위한 사전 요구 사항". 필수 구성 요소가 충족되었는지 확인한 후 Astra Trident를 배포할 준비가 되었습니다.

Docker 관리형 플러그인 방법(버전 1.13/17.03 이상)

> 시작하기 전에
>
> 기존 데몬 방법으로 Astra Trident pRE Docker 1.13/17.03을 사용한 경우, 관리 플러그인 방법을 사용하기 전에 Astra Trident 프로세스를 중지하고 Docker 데몬을 다시 시작해야 합니다.

1. 실행 중인 모든 인스턴스 중지:

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

2. Docker를 다시 시작합니다.

```
systemctl restart docker
```

3. Docker Engine 17.03(새로운 1.13) 이상이 설치되어 있는지 확인합니다.

```
docker --version
```

버전이 최신 버전이 아니면 "설치를 설치하거나 업데이트합니다".

단계

1. 구성 파일을 생성하고 다음과 같이 옵션을 지정합니다.
   - `config`기본 파일 이름은 입니다 `config.json`그러나 를 지정하여 원하는 이름을 사용할 수 있습니다 `config` 옵션을 선택합니다. 구성 파일은 에 있어야 합니다 `/etc/netappdvp` 호스트 시스템의 디렉토리입니다.
   - `log-level`: 로깅 수준을 지정합니다 (`debug`, `info`, `warn`, `error`, `fatal`)를 클릭합니다. 기본값은 입니다 `info`.
   - `debug`: 디버그 로깅이 설정되었는지 여부를 지정합니다. 기본값은 false 입니다. TRUE인 경우 로그 수준을

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. 관리 플러그인 시스템을 사용하여 Astra Trident를 시작합니다. 대치 `<version>` 사용 중인 플러그인 버전(xxx.xx.x)을 사용합니다.

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

3. Astra Trident를 사용하여 구성된 시스템의 스토리지를 소모합니다.

a. "firstVolume"이라는 이름의 볼륨을 생성합니다.

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

b. 컨테이너가 시작될 때 기본 볼륨을 생성합니다.

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

c. "firstVolume" 볼륨을 제거합니다.

```
docker volume rm firstVolume
```

기존 방법(버전 **1.12** 이하)

시작하기 전에

1. Docker 버전 1.10 이상이 설치되어 있는지 확인합니다.

   ```
   docker --version
   ```

   버전이 최신 버전이 아니면 설치를 업데이트하십시오.

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

   또는, "배포 지침을 따릅니다".

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

단계

1. NetApp Docker Volume Plugin 설치 및 구성:

   a. 응용 프로그램 다운로드 및 압축 풀기:

   ```
   wget
   https://github.com/NetApp/trident/releases/download/v23.01.1/trident-
   installer-23.01.1.tar.gz
   tar zxf trident-installer-23.01.1.tar.gz
   ```

   b. 용지함 경로의 위치로 이동:

   ```
   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
```

> 지정하지 않는 한 볼륨 드라이버의 기본 이름은 "NetApp"입니다.

데몬이 시작된 후에는 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 볼륨 프로비저닝:

```
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 **Astra Trident**를 시작합니다

시스템 기반 시스템용 샘플 장치 파일은 에서 찾을 수 있습니다 `contrib/trident.service.example` Git repo에 있습니다. RHEL에서 파일을 사용하려면 다음을 수행하십시오.

1. 파일을 올바른 위치에 복사합니다.

실행 중인 인스턴스가 두 개 이상인 경우 단위 파일에 고유한 이름을 사용해야 합니다.

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

2. 파일을 편집하고 설명(행 2)을 드라이버 이름과 구성 파일 경로(줄 9)에 맞게 변경하여 환경을 반영합니다.
3. IT 부서에서 변경 사항을 수집하도록 시스템 다시 로드:

```
systemctl daemon-reload
```

4. 서비스를 활성화합니다.

   이 이름은 에서 파일의 이름을 지정하는 항목에 따라 달라집니다 `/usr/lib/systemd/system` 디렉토리.

```
systemctl enable trident
```

5. 서비스를 시작합니다.

```
systemctl start trident
```

6. 상태를 봅니다.

```
systemctl status trident
```

> 단위 파일을 수정할 때마다 를 실행합니다 `systemctl daemon-reload` IT 부서에서 변경 사항을 인지할 수 있는 명령입니다.

Astra Trident를 업그레이드하거나 제거합니다

사용 중인 볼륨에 영향을 주지 않고 Docker용 Astra Trident를 안전하게 업그레이드할 수 있습니다. 업그레이드 프로세스 중에는 잠시 시간이 있습니다 `docker volume` 플러그인으로 지정된 명령이 성공하지 않으며 플러그인이 다시 실행될 때까지 응용 프로그램이 볼륨을 마운트할 수 없습니다. 대부분의 경우 몇 초 이내에 완료됩니다.

업그레이드

Docker를 위한 Astra Trident를 업그레이드하려면 다음 단계를 수행하십시오.

단계

1. 기존 볼륨 나열:

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

2. 플러그인 비활성화:

```
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
```

> Astra Trident의 18.01 릴리스는 nDVP를 대체합니다. 에서 직접 업그레이드해야 합니다 `netapp/ndvp-plugin` 에 대한 영상 `netapp/trident-plugin` 이미지.

4. 플러그인 활성화:

```
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
```

Astra Trident(20.10 이전)의 이전 버전에서 Astra Trident 20.10 이상으로 업그레이드하는 경우 오류가 발생할 수 있습니다. 자세한 내용은 을 참조하십시오 "알려진 문제". 오류가 발생하면 먼저 플러그인을 비활성화한 다음 플러그인을 제거한 다음 추가 구성 매개 변수를 전달하여 필요한 Astra Trident 버전을 설치해야 합니다. `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

설치 제거

Docker용 Astra Trident를 제거하려면 다음 단계를 수행하십시오.

단계

1. 플러그인이 생성한 모든 볼륨을 제거합니다.
2. 플러그인 비활성화:

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

3. 플러그인 제거:

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

볼륨 작업

표준을 사용하여 볼륨을 쉽게 생성, 클론 복제 및 제거할 수 있습니다 `docker volume` 필요한 경우 지정된 Astra Trident 드라이버 이름의 명령입니다.

볼륨을 생성합니다

- 기본 이름을 사용하여 드라이버로 볼륨을 생성합니다.

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

- 특정 Astra Trident 인스턴스를 사용하여 볼륨 생성:

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

지정하지 않은 경우 "옵션", 드라이버 기본값이 사용됩니다.

- 기본 볼륨 크기를 재정의합니다. 다음 예를 참조하여 드라이버로 20GiB 볼륨을 생성합니다.

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

볼륨 크기는 옵션 단위(예: 10G, 20GB, 3TiB)가 포함된 정수 값이 포함된 문자열로 표시됩니다. 단위를 지정하지 않으면 기본값은 G입니다 크기 단위는 2(B, KiB, MiB, GiB, TiB) 또는 10(B, KB, MB, GB, TB)의 거듭제곱으로 표현될 수 있습니다. 단축 단위는 2의 거듭제곱을 사용합니다(G=GiB, T=TiB,…).

볼륨을 제거합니다

- 다른 Docker 볼륨과 마찬가지로 볼륨을 제거합니다.

```
docker volume rm firstVolume
```

를 사용할 때 `solidfire-san` 드라이버, 위의 예에서는 볼륨을 삭제 및 삭제합니다.

Docker를 위한 Astra Trident를 업그레이드하려면 다음 단계를 수행하십시오.

볼륨의 클론을 생성합니다

를 사용할 때 `ontap-nas`, `ontap-san`, `solidfire-san`, 및 `gcp-cvs storage drivers`, Astra Trident는 볼륨을 복제할 수 있습니다. 를 사용할 때 `ontap-nas-flexgroup` 또는 `ontap-nas-economy` 드라이버, 복제는 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 스냅샷은 생성하지 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 액세스합니다

파티션이 없고 파일 시스템이 Astra Trident(예: an)에서 지원되는 경우 Trident * Oonly * 를 사용하여 컨테이너를 통해 외부에서 생성된 블록 장치(또는 해당 클론)에 액세스할 수 있습니다 `ext4`형식 지정됨 `/dev/sdc1` Astra Trident를 통해 액세스할 수 없음).

드라이버별 볼륨 옵션

각 스토리지 드라이버에는 다양한 옵션이 있으며, 볼륨 생성 시 이를 지정하여 결과를 사용자 지정할 수 있습니다. 구성된 스토리지 시스템에 적용되는 옵션은 아래를 참조하십시오.

볼륨 생성 작업 중에 이러한 옵션을 사용하는 것은 간단합니다. 을 사용하여 옵션과 값을 제공합니다 `-o` CLI 작업 중

operator이다. 이러한 값은 JSON 구성 파일의 모든 등가 값을 재정의합니다.

ONTAP 볼륨 옵션

NFS 및 iSCSI에 대한 볼륨 생성 옵션은 다음과 같습니다.

옵션을 선택합니다	설명
`size`	볼륨의 크기는 기본적으로 1GiB로 설정됩니다.
`spaceReserve`	볼륨을 씬 또는 일반 프로비저닝합니다. 기본값은 Thin 입니다. 유효한 값은 입니다 `none` (씬 프로비저닝됨) 및 을 누릅니다 `volume` (일반 프로비저닝).
`snapshotPolicy`	그러면 스냅샷 정책이 원하는 값으로 설정됩니다. 기본값은 입니다 `none`즉, 볼륨에 대해 스냅샷이 자동으로 생성되지 않습니다. 스토리지 관리자가 수정하지 않는 한, "`default`"라는 정책이 모든 `ONTAP` 시스템에 존재하며, 이 정책은 `6`시간, `2`일, `2`주 스냅샷을 생성하고 유지합니다. 스냅샷에 보존된 데이터는 로 이동하여 복구할 수 있습니다 `.snapshot` 디렉토리에 저장됩니다.
`snapshotReserve`	이렇게 하면 스냅숏 예비 공간이 원하는 비율로 설정됩니다. 기본값은 값이 없습니다. 즉, snapshotPolicy를 선택한 경우 ONTAP가 snapshotReserve(일반적으로 5%)를 선택하거나 snapshotPolicy가 none인 경우 0%를 선택합니다. 모든 ONTAP 백엔드에 대한 구성 파일에서 기본 snapshotReserve 값을 설정할 수 있으며 ONTAP-NAS-이코노미 를 제외한 모든 ONTAP 백엔드에 대한 볼륨 생성 옵션으로 사용할 수 있습니다.
`splitOnClone`	볼륨을 클론 생성할 때 ONTAP가 상위 클론에서 즉시 클론을 분할합니다. 기본값은 입니다 `false`. 볼륨을 클론 복제하는 사용 사례에는 스토리지 효율성을 높일 기회가 없을 것 같기 때문에 생성 즉시 클론을 상위 볼륨에서 분리하는 것이 가장 좋습니다. 예를 들어 빈 데이터베이스를 복제하면 많은 시간을 절약할 수 있지만 스토리지 절감 효과는 거의 없으므로 즉시 클론을 분리하는 것이 좋습니다.

옵션을 선택합니다	설명
`encryption`	새 볼륨에 NVE(NetApp Volume Encryption)를 활성화합니다. 기본값은 입니다 `false`. 이 옵션을 사용하려면 NVE 라이센스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Astra Trident에 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다. 자세한 내용은 다음을 참조하십시오. "Astra Trident가 NVE 및 NAE와 연동되는 방식".
`tieringPolicy`	볼륨에 사용할 계층화 정책을 설정합니다. 비활성(콜드) 상태일 때 데이터를 클라우드 계층으로 이동할지 결정합니다.

다음 추가 옵션은 NFS * 에만 적용됩니다 *.

옵션을 선택합니다	설명
`unixPermissions`	볼륨 자체에 대한 권한 집합을 제어합니다. 기본적으로 사용 권한은 로 설정됩니다 `---rwxr-xr-x`, 또는 숫자 표기 0755, 및 `root` 소유자가 됩니다. 텍스트 또는 숫자 형식이 작동합니다.
`snapshotDir`	를 로 설정합니다 `true` 가 을(를) 만듭니다 `.snapshot` 볼륨을 액세스하는 클라이언트가 볼 수 있는 디렉토리입니다. 기본값은 입니다 `false`이는 의 가시성을 의미합니다 `.snapshot` 디렉토리는 기본적으로 사용되지 않습니다. 예를 들어, 공식 MySQL 이미지와 같은 일부 이미지는 에서 예상대로 작동하지 않습니다 `.snapshot` 디렉토리가 표시됩니다.
`exportPolicy`	볼륨에 사용할 엑스포트 정책을 설정합니다. 기본값은 입니다 `default`.
`securityStyle`	볼륨에 액세스하는 데 사용할 보안 스타일을 설정합니다. 기본값은 입니다 `unix`. 유효한 값은 입니다 `unix` 및 `mixed`.

다음 추가 옵션은 iSCSI * 에만 적용됩니다 *.

옵션을 선택합니다	설명
`fileSystemType`	iSCSI 볼륨을 포맷하는 데 사용되는 파일 시스템을 설정합니다. 기본값은 입니다 `ext4`. 유효한 값은 입니다 `ext3`, `ext4`, 및 `xfs`.

옵션을 선택합니다	설명
`spaceAllocation`	를 로 설정합니다 `false` 에서는 LUN의 공간 할당 기능을 해제합니다. 기본값은 입니다 `true`즉, ONTAP는 볼륨에 공간이 부족하고 볼륨의 LUN이 쓰기를 허용할 수 없을 때 호스트에 알립니다. 또한 이 옵션을 사용하면 호스트가 데이터를 삭제할 때 ONTAP에서 자동으로 공간을 재확보할 수 있습니다.

예

아래 예를 참조하십시오.

- 10GiB 볼륨 생성:

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

- 스냅샷을 사용하여 100GiB 볼륨 생성:

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

- setuid 비트가 설정된 볼륨을 생성합니다.

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

최소 볼륨 크기는 20MiB입니다.

스냅숏 예비 공간이 지정되지 않은 경우 스냅숏 정책이 입니다 `none`, Trident는 0%의 스냅숏 예비 공간을 사용합니다.

- 스냅숏 정책이 없고 스냅숏 예비 공간이 없는 볼륨을 생성합니다.

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

- 스냅샷 정책이 없는 볼륨 및 10%의 사용자 지정 스냅숏 예비 공간을 생성합니다.

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

- 스냅샷 정책 및 10%의 사용자 지정 스냅숏 예비 공간이 있는 볼륨을 생성합니다.

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

- 스냅샷 정책을 사용하여 볼륨을 생성하고 ONTAP의 기본 스냅샷 예약 공간(일반적으로 5%)을 적용합니다.

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

Element 소프트웨어 볼륨 옵션

Element 소프트웨어 옵션은 볼륨과 연관된 서비스 품질(QoS) 정책의 크기 및 크기를 표시합니다. 볼륨이 생성되면 볼륨을 사용하여 연결된 QoS 정책을 지정합니다 `-o type=service_level` 명칭.

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

기타 Element 소프트웨어 볼륨 생성 옵션에는 다음이 포함됩니다.

옵션을 선택합니다	설명
`size`	볼륨 크기, 기본값 1GiB 또는 구성 항목… "기본값":{" 크기":"5G"}.
`blocksize`	512 또는 4096 중 하나를 사용합니다. 기본값은 512 또는 구성 항목 DefaultBlockSize 입니다.

예

QoS 정의가 포함된 다음 샘플 구성 파일을 참조하십시오.

```
{
    "...": "..."
    "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 구성에서는 Bronze, Silver, Gold의 세 가지 정책 정의가 있습니다. 이러한 이름은 임의로 지정됩니다.

- 10GiB 골드 볼륨 생성:

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

- 100GiB Bronze 볼륨 생성:

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

로그를 수집합니다

문제 해결에 도움이 되는 로그를 수집할 수 있습니다. 로그를 수집하는 방법은 Docker 플러그인을 실행하는 방법에 따라 다릅니다.

문제 해결을 위해 로그를 수집합니다

단계

1. 권장 관리 플러그인 방법(예: 사용)을 사용하여 Astra Trident를 실행하는 경우 `docker plugin` 명령 참조), 다음과 같이 봅니다.

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

   표준 로깅 수준에서는 대부분의 문제를 진단할 수 있어야 합니다. 충분하지 않은 경우 디버그 로깅을 활성화할 수 있습니다.

2. 디버그 로깅을 사용하려면 디버그 로깅을 사용하도록 설정한 플러그인을 설치합니다.

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

   또는 플러그인이 이미 설치된 경우 디버그 로깅을 활성화합니다.

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

3. 호스트에서 바이너리 자체를 실행하는 경우 호스트 에서 로그를 사용할 수 있습니다 `/var/log/netappdvp` 디렉토리. 디버그 로깅을 활성화하려면 를 지정합니다 `-debug` 플러그인을 실행할 때

일반적인 문제 해결 팁

- 새 사용자가 실행하는 가장 일반적인 문제는 플러그 인을 초기화할 수 없도록 잘못 구성된 것입니다. 이 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
  ```

  즉, 플러그인을 시작하지 못했습니다. 다행히 플러그인은 사용자가 겪을 수 있는 대부분의 문제를 진단하는 데

도움이 되는 포괄적인 로깅 기능을 갖추고 있습니다.

- 컨테이너에 PV를 장착하는 데 문제가 있는 경우, 반드시 확인하십시오 `rpcbind` 가 설치되어 실행 중입니다. 호스트 OS에 필요한 패키지 관리자를 사용하여 확인합니다 `rpcbind` 가 실행 중입니다. 를 실행하여 rpcbind 서비스의 상태를 확인할 수 있습니다 `systemctl status rpcbind` 또는 그와 동등한 제품이라는 것입니다.

여러 **Astra Trident** 인스턴스를 관리합니다

여러 스토리지 구성을 동시에 사용할 수 있도록 하려는 경우 Trident의 여러 인스턴스가 필요합니다. 여러 인스턴스의 키는 를 사용하여 다른 이름을 지정하는 것입니다 `--alias` 컨테이너형 플러그인 옵션, 또는 `--volume-driver` 호스트에서 Trident를 인스턴스화할 때 선택할 수 있는 옵션입니다.

Docker 관리 플러그인 단계(버전 **1.13/17.03** 이상)

1. 별칭 및 구성 파일을 지정하는 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 지정하는 볼륨을 생성합니다.

   예를 들어 금괴 볼륨의 경우:

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

   예를 들어, 은 볼륨의 경우:

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

기존(버전 **1.12** 이하) 단계

1. 사용자 지정 드라이버 ID를 사용하여 NFS 구성으로 플러그인을 시작합니다.

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

2. 사용자 지정 드라이버 ID를 사용하여 iSCSI 구성으로 플러그인을 시작합니다.

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

3. 각 드라이버 인스턴스에 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

Astra Trident 구성에 사용할 수 있는 구성 옵션을 확인하십시오.

글로벌 구성 옵션

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

옵션을 선택합니다	설명	예
`version`	구성 파일 버전 번호입니다	`1`
`storageDriverName`	스토리지 드라이버 이름입니다	`ontap-nas`, `ontap-san`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `solidfire-san`
`storagePrefix`	볼륨 이름에 대한 선택적 접두사입니다. 기본값: `netappdvp_`.	`staging_`
`limitVolumeSize`	볼륨 크기에 대한 선택적 제한. 기본값: ""(강제 적용 안 됨)	`10g`

사용하지 마십시오 `storagePrefix` 요소 백엔드에 대한 (기본값 포함). 기본적으로 은(는) 입니다 `solidfire-san` 드라이버는 이 설정을 무시하고 접두어를 사용하지 않습니다. Docker 볼륨 매핑에 특정 tenantID를 사용하거나 이름 문지가 사용된 경우 Docker의 Docker 버전, 드라이버 정보 및 원시 이름으로 채워진 특성 데이터를 사용하는 것이 좋습니다.

생성하는 모든 볼륨에서 기본 옵션을 지정하지 않아도 됩니다. 를 클릭합니다 `size` 옵션은 모든 컨트롤러 유형에 사용할 수 있습니다. 기본 볼륨 크기를 설정하는 방법은 ONTAP 구성 섹션을 참조하십시오.

옵션을 선택합니다	설명	예
`size`	새 볼륨의 선택적 기본 크기입니다. 기본값: `1G`	`10G`

ONTAP 구성

위의 글로벌 구성 값 외에도 ONTAP를 사용할 경우 다음과 같은 최상위 옵션을 사용할 수 있습니다.

옵션을 선택합니다	설명	예
`managementLIF`	ONTAP 관리 LIF의 IP 주소입니다. FQDN(정규화된 도메인 이름)을 지정할 수 있습니다.	`10.0.0.1`
`dataLIF`	프로토콜 LIF의 IP 주소입니다. • ONTAP NAS 드라이버 *: 지정할 것을 권장합니다 `dataLIF`. 제공되지 않는 경우 Astra Trident는 SVM에서 데이터 LIF를 가져옵니다. NFS 마운트 작업에 사용할 FQDN(정규화된 도메인 이름)을 지정하면 여러 데이터 LIF에서 로드 밸런싱을 위해 라운드 로빈 DNS를 생성할 수 있습니다. • ONTAP SAN 드라이버 *: iSCSI에 대해 지정하지 마십시오. Astra Trident가 사용합니다 "ONTAP 선택적 LUN 맵" 다중 경로 세션을 설정하는 데 필요한 iSCI LIF를 검색하려면 경고 발생 시 `dataLIF` 명시적으로 정의됩니다.	`10.0.0.2`
`svm`	사용할 스토리지 가상 머신(관리 LIF가 클러스터 LIF인 경우 필요)	`svm_nfs`
`username`	스토리지 디바이스에 접속할 사용자 이름입니다	`vsadmin`
`password`	스토리지 디바이스에 연결하는 암호입니다	`secret`

옵션을 선택합니다	설명	예
aggregate	프로비저닝을 위한 애그리게이트(선택 사항, SVM에 셋팅해야 하는 경우) 의 경우 ontap-nas-flexgroup 드라이버, 이 옵션은 무시됩니다. SVM에 할당된 모든 애그리게이트는 FlexGroup 볼륨을 프로비저닝하는 데 사용됩니다.	aggr1
limitAggregateUsage	선택 사항으로, 사용량이 이 백분율보다 높을 경우 용량 할당에 실패합니다	75%
nfsMountOptions	NFS 마운트 옵션을 세밀하게 제어합니다. 기본값은 "-o nfsvers=3"입니다. * 에서만 사용할 수 있습니다 ontap-nas 및 ontap-nas-economy 드라이버 *. "NFS 호스트 구성 정보는 여기 를 참조하십시오".	-o nfsvers=4
igroupName	플러그인에서 사용되는 igroup. 기본값은 입니다 netappdvp. * ONTAP-SAN driver * 에서만 사용 가능.	myigroup
limitVolumeSize	최대 요청 가능 볼륨 크기 및 qtree 상위 볼륨 크기. 의 경우 * ontap-nas-economy 드라이버, 이 옵션은 생성된 FlexVol의 크기를 추가로 제한합니다.	300g
qtreesPerFlexvol	FlexVol당 최대 qtree는 범위 [50, 300]에 있어야 하며 기본값은 200입니다. 의 경우 * ontap-nas-economy 드라이버, 이 옵션을 사용하면 FlexVol*당 최대 qtree 수를 사용자 지정할 수 있습니다.	300

생성하는 모든 볼륨에 기본 옵션을 지정하지 않아도 됩니다.

옵션을 선택합니다	설명	예
spaceReserve	공간 예약 모드; none (씬 프로비저닝됨) 또는 을 클릭합니다 volume (두꺼운)	none

옵션을 선택합니다	설명	예
`snapshotPolicy`	사용할 스냅샷 정책이며 기본값은 입니다 `none`	`none`
`snapshotReserve`	스냅숏 예비 공간 비율, 기본값은 ""이며 ONTAP 기본값을 사용합니다	`10`
`splitOnClone`	생성 시 상위 클론에서 클론을 분할합니다. 기본값은 입니다 `false`	`false`
`encryption`	새로운 볼륨에 NVE(NetApp Volume Encryption)를 지원하며 기본값은 로 설정됩니다 `false`. 이 옵션을 사용하려면 NVE 라이센스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Astra Trident에 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다. 자세한 내용은 다음을 참조하십시오. "Astra Trident가 NVE 및 NAE와 연동되는 방식".	참
`unixPermissions`	프로비저닝된 NFS 볼륨에 대한 NAS 옵션, 기본값은 로 설정됩니다 `777`	`777`
`snapshotDir`	에 액세스하기 위한 NAS 옵션입니다 `.snapshot` Directory(디렉터리), 기본값 `false`	`true`
`exportPolicy`	NFS 엑스포트 정책에서 사용할 NAS 옵션이며 기본값은 입니다 `default`	`default`
`securityStyle`	프로비저닝된 NFS 볼륨에 액세스하기 위한 NAS 옵션입니다. NFS를 지원합니다 `mixed` 및 `unix` 보안 스타일. 기본값은 입니다 `unix`.	`unix`
`fileSystemType`	SAN 옵션 - 파일 시스템 유형을 선택합니다. 기본값은 입니다 `ext4`	`xfs`
`tieringPolicy`	사용할 계층화 정책, 기본값은 입니다 `none`; `snapshot-only` ONTAP 9.5 이전 SVM-DR 구성용	`none`

축척 옵션

를 클릭합니다 `ontap-nas` 및 `ontap-san` 드라이버는 각 Docker 볼륨에 대해 ONTAP FlexVol를 생성합니다. ONTAP는 최대 12,000개의 FlexVol 클러스터를 사용하여 클러스터 노드당 최대 1,000개의 FlexVol을 지원합니다. Docker 볼륨 요구 사항이 이러한 제한 범위 내에 있으면 를 참조하십시오 `ontap-nas` 드라이버는 Docker 볼륨 세부 스냅샷 및 클론 복제와 같은 FlexVol에서 제공되는 추가 기능 때문에 선호되는 NAS 솔루션입니다.

FlexVol 제한에 수용 가능한 것보다 더 많은 Docker 볼륨이 필요한 경우 를 선택합니다 `ontap-nas-economy` 또는 을 누릅니다 `ontap-san-economy` 드라이버.

를 클릭합니다 `ontap-nas-economy` 드라이버는 자동으로 관리되는 FlexVol 풀 내에 Docker 볼륨을 ONTAP qtree로 생성합니다. qtree는 일부 기능을 희생하여 클러스터 노드당 최대 100,000 및 클러스터당 2,400,000까지 훨씬 더 뛰어난 확장을 제공합니다. 를 클릭합니다 `ontap-nas-economy` 드라이버는 Docker 볼륨 세부 스냅샷 또는 클론 복제를 지원하지 않습니다.

> 를 클릭합니다 `ontap-nas-economy` Swarm은 여러 노드에서 볼륨 생성을 오케스트레이션하지 않으므로 현재 Docker Swarm에서 드라이버가 지원되지 않습니다.

를 클릭합니다 `ontap-san-economy` 드라이버는 자동으로 관리되는 FlexVol의 공유 풀 내에 Docker 볼륨을 ONTAP LUN으로 생성합니다. 이렇게 하면 각 FlexVol가 하나의 LUN에만 제한되지 않으며 SAN 워크로드에 더 나은 확장성을 제공합니다. 스토리지 시스템에 따라 ONTAP는 클러스터당 최대 16384개의 LUN을 지원합니다. 볼륨이 그 아래에 LUN이 있으므로 이 드라이버는 Docker 볼륨 세부 스냅샷 및 클론 복제를 지원합니다.

를 선택합니다 `ontap-nas-flexgroup` 수십억 개의 파일로 구성된 페타바이트 규모로 성장할 수 있는 단일 볼륨으로 병렬 처리를 늘립니다. FlexGroups의 이상적인 사용 사례로는 AI/ML/DL, 빅데이터 및 분석, 소프트웨어 빌드, 스트리밍, 파일 저장소 등이 있습니다. Trident는 FlexGroup 볼륨을 프로비저닝할 때 SVM에 할당된 모든 애그리게이트를 사용합니다. Trident의 FlexGroup 지원에도 다음과 같은 고려 사항이 있습니다.

- ONTAP 버전 9.2 이상이 필요합니다.
- 이번 작부터 FlexGroups는 NFS v3만 지원합니다.
- SVM에 대해 64비트 NFSv3 식별자를 사용하는 것이 좋습니다.
- 최소 권장 FlexGroup 크기는 100GB입니다.
- FlexGroup 볼륨에 대해서는 클론 생성이 지원되지 않습니다.

FlexGroups에 적합한 FlexGroups 및 워크로드에 대한 자세한 내용은 를 참조하십시오 "NetApp FlexGroup 볼륨 모범 사례 및 구현 가이드 를 참조하십시오".

같은 환경에서 고급 기능과 대규모 확장을 위해 Docker Volume Plugin의 여러 인스턴스를 를 사용하여 실행할 수 있습니다 `ontap-nas` 또 다른 방법은 을 사용하는 것입니다 `ontap-nas-economy`.

ONTAP 구성 파일의 예

- NFS의 예 `ontap-nas` 드라이버 *

```
{
    "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"
    }
}
```

- NFS의 예 `ontap-nas-flexgroup` 드라이버 *

```
{
    "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"
    }
}
```

- NFS의 예 `ontap-nas-economy` 드라이버 *

```
{
    "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"
}
```

- iSCSI의 예 `ontap-san` 드라이버 *

```
{
    "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": "myigroup"
}
```

- NFS의 예 `ontap-san-economy` 드라이버 *

```
{
    "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": "myigroup"
}
```

Element 소프트웨어 구성

Element 소프트웨어(NetApp HCI/SolidFire)를 사용하는 경우 글로벌 구성 값 외에도 이러한 옵션을 사용할 수 있습니다.

옵션을 선택합니다	설명	예
`Endpoint`	<a href="https://<login>:<password>@<mvip>/json-rpc/<element-version>" class="bare">https://<login>:<password>@<mvip>/json-rpc/<element-version>; 으로 문의하십시오	https://admin:admin@192.168.160.3/json-rpc/8.0 으로 문의하십시오
`SVIP`	iSCSI IP 주소 및 포트	10.0.0.7:3260
`TenantName`	사용할 SolidFireF 테넌트(찾을 수 없는 경우 생성됨)	`docker`
`InitiatorIFace`	iSCSI 트래픽을 기본 인터페이스가 아닌 인터페이스로 제한할 때 인터페이스를 지정합니다	`default`
`Types`	QoS 사양	아래 예를 참조하십시오
`LegacyNamePrefix`	업그레이드된 Trident 설치의 접두사 1.3.2 이전 버전의 Trident를 사용하고 기존 볼륨을 업그레이드하는 경우 이 값을 설정하여 볼륨 이름 방법을 통해 매핑된 이전 볼륨에 액세스해야 합니다.	`netappdvp-`

를 클릭합니다 `solidfire-san` 드라이버가 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

Docker에서 Astra Trident를 사용할 때 알려진 문제 및 제한 사항에 대한 정보를 확인하십시오.

이전 버전에서 **Trident Docker** 볼륨 플러그인을 **20.10** 이상으로 업그레이드하면 해당 파일 또는 디렉터리 오류가 없는 업그레이드 오류가 발생합니다.

해결 방법

1. 플러그인을 비활성화합니다.

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

2. 플러그인을 제거합니다.

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

3. 추가 를 제공하여 플러그인을 다시 설치합니다 `config` 매개 변수.

```
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 단일 문자 이름을 Windows 경로로 해석합니다. "버그 25773을 참조하십시오".

Docker Swarm에는 Astra Trident가 모든 스토리지 및 드라이버 조합에서 이를 지원하지 않는 특정 동작이 있습니다.

- 현재 Docker Swarm은 볼륨 ID 대신 볼륨 이름을 고유한 볼륨 식별자로 사용합니다.
- 볼륨 요청은 Swarm 클러스터의 각 노드로 동시에 전송됩니다.
- 볼륨 플러그인(Astra Trident 포함)은 Swarm 클러스터의 각 노드에서 독립적으로 실행해야 합니다. ONTAP의 작동 방식 및 방법 때문입니다 `ontap-nas` 및 `ontap-san` 운전자 기능, 이러한 제한 내에서 작동할 수 있는 유일한 운전자이다.

나머지 드라이버에는 명확한 "승자" 없이 단일 요청에 대해 대량의 볼륨을 생성할 수 있는 경합 상태와 같은 문제가 있습니다. 예를 들어, Element에는 볼륨의 이름이 같지만 ID가 다를 수 있는 기능이 있습니다.

NetApp은 Docker 팀에 피드백을 제공했지만, 향후 소구에 대한 표시는 제공하지 않습니다.

FlexGroup를 프로비저닝하고 있는 경우, 두 번째 FlexGroup에 프로비저닝되는 FlexGroup와 공통되는 하나 이상의 애그리게이트가 있는 경우 ONTAP는 두 번째 FlexGroup를 프로비저닝하지 않습니다.

저작권 정보

Copyright © 2025 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 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.