Trident for Docker が必要です

Astra Trident

NetApp
January 14, 2026

This PDF was generated from https://docs.netapp.com/ja-jp/trident-2402/trident-docker/prereqs-docker.html on January 14, 2026. Always check docs.netapp.com for the latest.

目次

Trident for Docker が必要です . . . 1
導入の前提条件 . . . 1
要件を確認します . . . 1
NVMeツール . . . 3
Astra Trident を導入 . . . 4
Docker Managed Plugin メソッド（バージョン 1.13 / 17.03 以降） . . . 4
従来の方法（バージョン 1.12 以前） . . . 6
システム起動時に Astra Trident を起動 . . . 8
Astra Trident をアップグレードまたはアンインストールする . . . 9
アップグレード . . . 9
をアンインストールします . . . 10
ボリュームを操作します . . . 11
ボリュームを作成します . . . 11
ボリュームを削除します . . . 12
ボリュームのクローンを作成します . . . 12
外部で作成されたボリュームにアクセス . . . 13
ドライバ固有のボリュームオプション . . . 13
ログを収集します . . . 19
トラブルシューティング用にログを収集する . . . 19
一般的なトラブルシューティングのヒント . . . 19
複数の Astra Trident インスタンスを管理 . . . 20
Docker Managed Plugin （バージョン 1.13 / 17.03 以降）の手順 . . . 20
従来の（バージョン 1.12 以前）の場合の手順 . . . 20
ストレージ構成オプション . . . 21
グローバル構成オプション . . . 21
ONTAP構成 . . . 22
Element ソフトウェアの設定 . . . 29
既知の問題および制限事項 . . . 31
Trident Docker Volume Plugin を旧バージョンから 20.10 以降にアップグレードすると、該当するファイルエラーまたはディレクトリエラーなしでアップグレードが失敗します。 . . . 31
ボリューム名は 2 文字以上にする必要があります。 . . . 32
Docker Swarm には、 Astra Trident がストレージやドライバのあらゆる組み合わせでサポートしないようにする一定の動作があります。 . . . 32
FlexGroup をプロビジョニングする場合、プロビジョニングする FlexGroup と共通のアグリゲートが 2 つ目の FlexGroup に 1 つ以上あると、 ONTAP は 2 つ目の FlexGroup をプロビジョニングしません。 . . . 32

Trident for Docker が必要です

導入の前提条件

Trident を導入するには、必要なプロトコルをホストにインストールして設定しておく必要があります。

要件を確認します

- の導入がすべてを満たしていることを確認します "要件"。
- サポートされているバージョンの Docker がインストールされていることを確認します。Docker のバージョンが最新でない場合は、 "インストールまたは更新します"。

```
docker --version
```

- プロトコルの前提条件がホストにインストールおよび設定されていることを確認します。

NFSツール

オペレーティングシステム用のコマンドを使用して、NFSツールをインストールします。

RHEL 8以降

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

Ubuntu

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

NFSツールをインストールしたあとにワーカーノードをリブートして、コンテナにボリュームを接続する際の障害を回避します。

iSCSIツール

使用しているオペレーティングシステム用のコマンドを使用して、iSCSIツールをインストールします。

RHEL 8以降

1. 次のシステムパッケージをインストールします。

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

2. iscsi-initiator-utils のバージョンが 6.2.0.874-2.el7 以降であることを確認します。

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

3. スキャンを手動に設定：

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

4. マルチパスを有効化：

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

> 確認します `etc/multipath.conf` が含まれます `find_multipaths no` の下 `defaults`。

5. を確認します `iscsid` および `multipathd` 実行中：

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

6. を有効にして開始します `iscsi`：

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

Ubuntu

1. 次のシステムパッケージをインストールします。

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

2. open-iscsi バージョンが 2.0.874-5ubuntu2.10 以降（ bionic の場合）または 2.0.874-7.1ubuntu6.1 以降（ Focal の場合）であることを確認します。

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

3. スキャンを手動に設定：

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

4. マルチパスを有効化：

```
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
```

> 確認します `etc/multipath.conf` が含まれます `find_multipaths no` の下 `defaults`。

5. を確認します `open-iscsi` および `multipath-tools` 有効になっていて実行中：

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

NVMeツール

オペレーティングシステムに対応したコマンドを使用してNVMeツールをインストールします。

- NVMeにはRHEL 9以降が必要です。
- Kubernetesノードのカーネルバージョンが古すぎる場合や、使用しているカーネルバージョンに対応するNVMeパッケージがない場合は、ノードのカーネルバージョンをNVMeパッケージで更新しなければならないことがあります。

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
```

Astra Trident を導入

Docker向けAstra Tridentは、NetAppストレージプラットフォーム向けのDockerエコシステムと直接統合できます。ストレージプラットフォームから Docker ホストまで、ストレージリソースのプロビジョニングと管理をサポートします。また、将来プラットフォームを追加するためのフレームワークもサポートします。

Astra Trident の複数のインスタンスを同じホストで同時に実行できます。これにより、複数のストレージシステムとストレージタイプへの同時接続が可能になり、 Docker ボリュームに使用するストレージをカスタマイズできます。

必要なもの

を参照してください "導入の前提条件"。前提条件を満たしていることを確認したら、 Astra Trident を導入する準備ができました。

Docker Managed Plugin メソッド（バージョン 1.13 / 17.03 以降）

作業を開始する前に

従来のデーモン方式で Astra Trident 以前の Docker 1.13 / 17.03 を使用していた場合は、マネージドプラグイン方式を使用する前に Astra Trident プロセスを停止し、 Docker デーモンを再起動してください。

1. 実行中のインスタンスをすべて停止します。

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

2. Docker を再起動します。

```
systemctl restart docker
```

3. Docker Engine 17.03 （新しい 1.13 ）以降がインストールされていることを確認します。

```
docker --version
```

バージョンが最新でない場合は、 "インストール環境をインストールまたは更新します"。

手順

1. 構成ファイルを作成し、次のオプションを指定します。
   - `config`:デフォルトのファイル名はです `config.json`ただし、を指定すると、選択した任意の名前を使用できます `config` オプションを指定してファイル名を指定します構成ファイルはに格納されている必要があります `/etc/netappdvp` ホストシステム上のディレクトリ。
   - `log-level`:ログレベルを指定します (`debug`、 `info`、 `warn`、 `error`、 `fatal`）。デフォルトはです `info`。
   - `debug`:デバッグロギングを有効にするかどうかを指定します。デフォルトは false です。true の場合、ログレベルを上書きします。
     i. 構成ファイルの場所を作成します。

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

     ii. 構成ファイルを作成します

```
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. マネージドプラグインシステムを使用して Astra Trident を起動交換してください `<version>` 使用しているプラグインのバージョン（xxx.xxx.xxx.xxx）を使用している必要があります。

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

3. Astra Trident を使用して、構成したシステムのストレージを使用しましょう。

a. 「 firstVolume 」という名前のボリュームを作成します。

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

b. コンテナの開始時にデフォルトのボリュームを作成します。

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

c. ボリューム「 firstVolume 」を削除します。

```
docker volume rm firstVolume
```

従来の方法（バージョン **1.12** 以前）

作業を開始する前に

1. バージョン 1.10 以降の Docker がインストールされていることを確認します。

```
docker --version
```

使用しているバージョンが最新でない場合は、インストールを更新します。

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

または "ご使用のディストリビューションの指示に従ってください"。

2. NFS または iSCSI がシステムに対して設定されていることを確認します。

手順

1. NetApp Docker Volume Plugin をインストールして設定します。

a. アプリケーションをダウンロードして開梱します。

```
wget
https://github.com/NetApp/trident/releases/download/v24.10.0/trident-
installer-24.02.0.tar.gz
tar zxf trident-installer-24.02.0.tar.gz
```

b. ビンパス内の場所に移動します。

```
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. 構成ファイルの場所を作成します。

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

d. 構成ファイルを作成します

```
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. バイナリを配置して構成ファイルを作成したら、目的の構成ファイルを使用してTridentデーモンを起動します。

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

指定されていない場合、ボリュームドライバのデフォルト名は「NetApp」です。

デーモンが開始されたら、 Docker CLI インターフェイスを使用してボリュームを作成および管理できます

3. ボリュームを作成します

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

4. コンテナの開始時に Docker ボリュームをプロビジョニング：

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

5. Docker ボリュームを削除します。

```
docker volume rm trident_1
docker volume rm trident_2
```

システム起動時に **Astra Trident** を起動

システムベースのシステムのサンプルユニットファイルは、から入手できます `contrib/trident.service.example` Gitリポジトリで実行します。RHELでファイルを使用するには、次の手順を実行します。

1. ファイルを正しい場所にコピーします。

   複数のインスタンスを実行している場合は、ユニットファイルに一意の名前を使用してください。

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

2. ファイルを編集し、概要（ 2 行目）を変更してドライバ名と構成ファイルのパス（ 9 行目）を環境に合わせます。
3. 変更を取り込むためにシステムをリロードします。

```
systemctl daemon-reload
```

4. サービスを有効にします。

   この名前は、ファイルの名前によって異なります `/usr/lib/systemd/system` ディレクトリ。

```
systemctl enable trident
```

5. サービスを開始します。

```
systemctl start trident
```

6. ステータスを確認します。

```
systemctl status trident
```

単位ファイルを変更する場合は、を実行します `systemctl daemon-reload` 変更を認識するためのコマンド。

Astra Trident をアップグレードまたはアンインストールする

使用中のボリュームに影響を与えることなく、 Astra Trident for Docker を安全にアップグレードできます。アップグレードプロセスでは、が短時間実行されます `docker volume` プラグインで指示されたコマンドは正常に実行されず、プラグインが再度実行されるまでアプリケーションはボリュームをマウントできません。ほとんどの場合、これは秒の問題です。

アップグレード

Astra Trident for Docker をアップグレードするには、次の手順を実行します。

手順

1. 既存のボリュームを表示します。

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

2. プラグインを無効にします。

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

3. プラグインをアップグレードします。

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

nDVP に代わる、 Astra Trident の 18.01 リリース。から直接アップグレードする必要があります `netapp/ndvp-plugin` への画像 `netapp/trident-plugin` イメージ（Image）：

4. プラグインを有効にします。

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

5. プラグインが有効になっていることを確認します。

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

6. ボリュームが表示されることを確認します。

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

古いバージョンの Astra Trident （ 20.10 より前）から Astra Trident 20.10 以降にアップグレードすると、エラーが発生する場合があります。詳細については、を参照してください "既知の問題"。このエラーが発生した場合は、まずプラグインを無効にしてからプラグインを削除し、次に追加のconfigパラメータを渡して、必要なAstra Tridentバージョンをインストールします。 `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

をアンインストールします

Astra Trident for Docker をアンインストールするには、次の手順を実行します。

手順

1. プラグインで作成されたボリュームをすべて削除します。
2. プラグインを無効にします。

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

3. プラグインを削除します。

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

ボリュームを操作します

標準のを使用すると、ボリュームを簡単に作成、クローニング、および削除できます `docker volume` 必要に応じてAstra Tridentドライバ名を指定したコマンド。

ボリュームを作成します

- デフォルトの名前を使用して、ドライバでボリュームを作成します。

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

- 特定の Astra Trident インスタンスを使用してボリュームを作成します。

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

何も指定しない場合 "オプション（ Options ）"、ドライバのデフォルトが使用されます。

- デフォルトのボリュームサイズを上書きします。次の例を参照して、ドライバで 20GiB ボリュームを作成してください。

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

ボリュームサイズは、オプションの単位（ 10G 、 20GB 、 3TiB など）を含む整数値で指定します。単位を指定しない場合、デフォルトは g ですサイズの単位は、 2 の累乗（ B 、 KiB 、 MiB 、 GiB 、 TiB ）または 10 の累乗（ B 、 KB 、 MB 、 GB 、 TB ）のいずれかです。略記単位では、 2 の累乗が使用されます（ G=GiB 、 T=TiB 、 ...）。

ボリュームを削除します

- 他の Docker ボリュームと同様にボリュームを削除します。

```
docker volume rm firstVolume
```

> を使用する場合 `solidfire-san` driver、上記の例では、ボリュームを削除およびパージします。

Astra Trident for Docker をアップグレードするには、次の手順を実行します。

ボリュームのクローンを作成します

を使用する場合 `ontap-nas`、 `ontap-san`、 `solidfire-san`および `gcp-cvs storage drivers`Tridentがボリュームをクローニングできます。を使用する場合 `ontap-nas-flexgroup` または `ontap-nas-economy` ドライバ、クローニングはサポートされていません。既存のボリュームから新しいボリュームを作成すると、新しい Snapshot が作成されます。

- ボリュームを調べて Snapshot を列挙します。

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

- 既存のボリュームから新しいボリュームを作成します。その結果、新しい Snapshot が作成されます。

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

- ボリューム上の既存の Snapshot から新しいボリュームを作成します。新しい Snapshot は作成されません。

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

例

```
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
```

外部で作成されたボリュームにアクセス

Tridentを使用すると、外部で作成されたブロックデバイス（またはそのクローン）にTrident *からアクセスできます。Tridentは、パーティションがなく、Astra Tridentでサポートされているファイルシステム（など）の場合にのみ利用できます `ext4`-フォーマット済み `/dev/sdc1` Astra Trident経由ではアクセスできません）。

ドライバ固有のボリュームオプション

ストレージドライバにはそれぞれ異なるオプションがあり、ボリュームの作成時に指定することで結果をカスタマイズできます。構成済みのストレージシステムに適用されるオプションについては、以下を参照してください。

ボリューム作成処理では、これらのオプションを簡単に使用できます。を使用して、オプションと値を指定し

ます `-o` CLI処理中の演算子。これらは、 JSON 構成ファイルの同等の値よりも優先されます。

ONTAP ボリュームのオプション

NFS と iSCSI のどちらの場合も、 volume create オプションには次のオプションがあります。

オプション	説明
`size`	ボリュームのサイズ。デフォルトは 1GiB です。
`spaceReserve`	ボリュームをシンプロビジョニングまたはシックプロビジョニングします。デフォルトはシンです。有効な値はです `none` （シンプロビジョニング）および `volume` （シックプロビジョニング）。
`snapshotPolicy`	Snapshot ポリシーが目的の値に設定されます。デフォルトはです `none`つまり、ボリュームに対してSnapshotが自動的に作成されることはありません。ストレージ管理者によって変更されていない限り、「 `default` 」という名前のポリシーがすべての `ONTAP` システムに存在し、 `6` 個の時間単位 `Snapshot` 、 `2` 個の日単位 `Snapshot` 、および `2` 個の週単位 `Snapshot` を作成して保持します。Snapshotに保存されているデータは、にアクセスしてリカバリできます `.snapshot` ボリューム内の任意のディレクトリ内のディレクトリ。
`snapshotReserve`	これにより、 Snapshot リザーブの割合が希望する値に設定されます。デフォルト値は no で、 Snapshot ポリシーを選択した場合は ONTAP によって snapshotReserve が選択されます（通常は 5% ）。Snapshot ポリシーがない場合は 0% が選択されます。構成ファイルのすべての ONTAP バックエンドに対して snapshotReserve のデフォルト値を設定できます。また、この値は、 ONTAP-NAS-エコノミー を除くすべての ONTAP バックエンドでボリューム作成オプションとして使用できます。
`splitOnClone`	ボリュームをクローニングすると、そのクローンが原因 ONTAP によって親から即座にスプリットされます。デフォルトはです `false`。クローンボリュームのクローニングは、作成直後に親からクローンをスプリットする方法を推奨します。これは、ストレージ効率化の効果がまったくないためです。たとえば、空のデータベースをクローニングしても時間は大幅に短縮されますが、ストレージはほとんど削減されません。そのため、クローンはすぐにスプリットすることを推奨します。

オプション	説明
`encryption`	新しいボリュームでNetApp Volume Encryption（NVE）を有効にします。デフォルトはです `false`。このオプションを使用するには、クラスタで NVE のライセンスが設定され、有効になっている必要があります。 NAEがバックエンドで有効になっている場合は、Astra Tridentでプロビジョニングされたすべてのボリュームがに有効になります。 詳細については、以下を参照してください。 "Astra TridentとNVEおよびNAEの相互運用性"。
`tieringPolicy`	ボリュームに使用する階層化ポリシーを設定します。これにより、アクセス頻度の低いコールドデータをクラウド階層に移動するかどうかが決まります。

以下は、 NFS * のみ * 用の追加オプションです。

オプション	説明
`unixPermissions`	これにより、ボリューム自体の権限セットを制御できます。デフォルトでは、権限はに設定されます `---rwxr-xr-x`または数値表記`0755`、およびです `root` が所有者になります。テキスト形式または数値形式のどちらかを使用できます。
`snapshotDir`	これをに設定します `true` がを作成します `.snapshot` ボリュームにアクセスしているクライアントから認識できるディレクトリ。デフォルト値はです `false`の可視性を意味します `.snapshot` ディレクトリはデフォルトで無効になっています。一部のイメージ（公式のMySQLイメージなど）が、 `.snapshot` ディレクトリが表示されます。
`exportPolicy`	ボリュームで使用するエクスポートポリシーを設定します。デフォルトはです `default`。
`securityStyle`	ボリュームへのアクセスに使用するセキュリティ形式を設定します。デフォルトはです `unix`。有効な値はです `unix` および `mixed`。

以下の追加オプションは、 iSCSI * のみ * 用です。

オプション	説明
`fileSystemType`	iSCSI ボリュームのフォーマットに使用するファイルシステムを設定します。デフォルトはです `ext4`。有効な値はです `ext3`、 `ext4`および `xfs`。
`spaceAllocation`	これをに設定します `false` LUNのスペース割り当て機能を無効にします。デフォルト値はです `true`つまり、ボリュームのスペースが不足し、ボリューム内のLUNに書き込みを受け付けられなくなったときに、ONTAP からホストに通知されます。また、このオプションで ONTAP 、ホストでデータが削除された時点での自動スペース再生も有効になります。

例

以下の例を参照してください。

- 10GiB ボリュームを作成します。

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

- Snapshot を使用して 100GiB のボリュームを作成します。

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

- setuid ビットが有効になっているボリュームを作成します。

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

最小ボリュームサイズは 20MiB です。

Snapshotリザーブが指定されていない場合、Snapshotポリシーはです `none`Tridentは0%のSnapshotリザーブを使用します。

- Snapshot ポリシーがなく、 Snapshot リザーブがないボリュームを作成します。

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

- Snapshot ポリシーがなく、カスタムの Snapshot リザーブが 10% のボリュームを作成します。

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

- Snapshot ポリシーを使用し、カスタムの Snapshot リザーブを 10% に設定してボリュームを作成します。

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

- Snapshot ポリシーを設定してボリュームを作成し、 ONTAP のデフォルトの Snapshot リザーブ（通常は 5% ）を受け入れます。

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

Element ソフトウェアのボリュームオプション

Element ソフトウェアのオプションでは、ボリュームに関連付けられているサービス品質（ QoS ）ポリシーのサイズと QoS を指定できます。ボリュームの作成時に、関連付けられているQoSポリシーをを使用して指定します `-o type=service_level` 名称。

Element ドライバを使用して QoS サービスレベルを定義する最初の手順は、少なくとも 1 つのタイプを作成し、構成ファイル内の名前に関連付けられた最小 IOPS 、最大 IOPS 、バースト IOPS を指定することです。

Element ソフトウェアのその他のボリューム作成オプションは次のとおりです。

オプション	説明
`size`	ボリュームのサイズ。デフォルト値は 1GiB または設定エントリ … 「 defaults 」： 「 size 」：「 5G 」 。
`blocksize`	512 または 4096 のいずれかを使用します。デフォルトは 512 または config エントリ DefaultBlockSize です。

例

QoS 定義を含む次のサンプル構成ファイルを参照してください。

```
{
    "...": "..."
    "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
            }
        }
    ]
}
```

上記の構成では、 Bronze 、 Silver 、 Gold の 3 つのポリシー定義を使用します。これらの名前は任意です。

- 10GiB の Gold ボリュームを作成します。

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

- 100GiB Bronze ボリュームを作成します。

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

ログを収集します

トラブルシューティングに役立つログを収集できます。ログの収集方法は、 Docker プラグインの実行方法によって異なります。

トラブルシューティング用にログを収集する

手順

1. 推奨される管理プラグイン方法（を使用）でAstra Tridentを実行している場合 `docker plugin` コマンド）で表示される情報は次のとおりです。

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

   標準的なロギングレベルでは、ほとんどの問題を診断できます。十分でない場合は、デバッグロギングを有効にすることができます。

2. デバッグロギングをイネーブルにするには、デバッグロギングをイネーブルにしてプラグインをインストールします。

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

   または、プラグインがすでにインストールされている場合にデバッグログを有効にします。

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

3. ホストでバイナリ自体を実行している場合、ログはホストの `/var/log/netappdvp` ディレクトリ。デバッグロギングを有効にするには、を指定します `-debug` プラグインを実行すると、

一般的なトラブルシューティングのヒント

- 新しいユーザーが実行する最も一般的な問題は、プラグインの初期化を妨げる構成ミスです。この場合、プラグインをインストールまたは有効にしようとすると、次のようなメッセージが表示されることがあります。

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

これは、プラグインの起動に失敗したことを意味します。幸い、このプラグインには、発生する可能性の高い問題のほとんどを診断するのに役立つ包括的なログ機能が組み込まれています。

- PVをコンテナにマウントする際に問題が発生する場合は、を確認してください `rpcbind` をインストールして実行しておきます。ホストOSに必要なパッケージマネージャを使用して、かどうかを確認します `rpcbind` を実行しています。rpcbindサービスのステータスを確認するには、を実行します `systemctl status rpcbind` またはそれと同等のものです。

複数の Astra Trident インスタンスを管理

複数のストレージ構成を同時に使用する必要がある場合は、 Trident の複数のインスタンスが必要です。複数のインスタンスを作成するには、を使用して異なる名前を付けます `--alias` オプションにコンテナ化プラグインを指定するか、を指定します `--volume-driver` ホストでTridentをインスタンス化する際のオプション。

Docker Managed Plugin （バージョン 1.13 / 17.03 以降）の手順

1. エイリアスと構成ファイルを指定して、最初のインスタンスを起動します。

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

2. 別のエイリアスと構成ファイルを指定して、 2 番目のインスタンスを起動します。

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

3. ドライバ名としてエイリアスを指定するボリュームを作成します。

   たとえば、 gold ボリュームの場合：

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

   たとえば、 Silver ボリュームの場合：

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

従来の（バージョン 1.12 以前）の場合の手順

1. カスタムドライバ ID を使用して NFS 設定でプラグインを起動します。

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

2. カスタムドライバ ID を使用して、 iSCSI 構成でプラグインを起動します。

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

3. ドライバインスタンスごとに Docker ボリュームをプロビジョニングします。

   たとえば、 NFS の場合：

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

   たとえば、 iSCSI の場合：

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

ストレージ構成オプション

Astra Trident 構成で使用できる設定オプションを確認してください。

グローバル構成オプション

以下の設定オプションは、使用するストレージプラットフォームに関係なく、すべての Astra Trident 構成に適用されます。

オプション	説明	例
`version`	構成ファイルのバージョン番号	`1`
`storageDriverName`	ストレージドライバの名前	`ontap-nas`、 `ontap-san`、 `ontap-nas-economy`、 `ontap-nas-flexgroup`、 `solidfire-san`
`storagePrefix`	ボリューム名のオプションのプレフィックス。デフォルト： `netappdvp_`。	`staging_`

オプション	説明	例
`limitVolumeSize`	ボリュームサイズに関するオプションの制限。デフォルト：""（強制なし）	`10g`

使用しないでください `storagePrefix` （デフォルトを含む）をElementバックエンドに使用します。デフォルトでは、が表示されます `solidfire-san` ドライバはこの設定を無視し、プレフィックスを使用しません。Docker ボリュームマッピングには特定の tenantID を使用するか、 Docker バージョン、ドライバ情報、名前の munging が使用されている可能性がある場合には Docker から取得した属性データを使用することを推奨します。

作成するすべてのボリュームでデフォルトのオプションを指定しなくても済むようになっています。。 `size` オプションはすべてのコントローラタイプで使用できます。デフォルトのボリュームサイズの設定方法の例については、 ONTAP の設定に関するセクションを参照してください。

オプション	説明	例
`size`	新しいボリュームのオプションのデフォルトサイズ。デフォルト：`1G`	`10G`

ONTAP構成

ONTAP を使用する場合は、上記のグローバル構成値に加えて、次のトップレベルオプションを使用できます。

オプション	説明	例
`managementLIF`	ONTAP 管理 LIF の IP アドレス。Fully Qualified Domain Name （ FQDN ；完全修飾ドメイン名）を指定できます。	`10.0.0.1`

オプション	説明	例
`dataLIF`	プロトコル LIF の IP アドレス。 • ONTAP NASドライバ*:を指定することをお勧めします `dataLIF`。指定しない場合は、Astra TridentがSVMからデータLIFを取得します。NFSマウント処理に使用するFully Qualified Domain Name（FQDN；完全修飾ドメイン名）を指定して、ラウンドロビンDNSを作成して複数のデータLIF間で負荷を分散することができます。 • ONTAP SANドライバ*: iSCSIには指定しないでください Astra Tridentが使用 "ONTAP の選択的LUNマップ" iSCSI LIFを検出するには、マルチパスセッションを確立する必要があります。の場合は警告が生成されます `dataLIF` は明示的に定義されます。	`10.0.0.2`
`svm`	使用する Storage Virtual Machine（管理 LIF がクラスタ LIF である場合は必須）	`svm_nfs`
`username`	ストレージデバイスに接続するユーザ名	`vsadmin`
`password`	ストレージ・デバイスに接続するためのパスワード	`secret`
`aggregate`	プロビジョニング用のアグリゲート（オプション。設定する場合は SVM に割り当てる必要があります）。をクリックします `ontap-nas-flexgroup` ドライバ。このオプションは無視されます。SVM に割り当てられたすべてのアグリゲートを使用して FlexGroup ボリュームがプロビジョニングされます。	`aggr1`
`limitAggregateUsage`	オプション。使用率がこの割合を超えている場合は、プロビジョニングを失敗させます	`75%`

オプション	説明	例
nfsMountOptions	NFS マウントオプションのきめ細かな制御。デフォルトは「 -o nfsvers=3 」です。でのみ使用できます **ontap-nas** および **ontap-nas-economy** ドライバ。 "ここでは、 NFS ホストの設定情報を参照してください"。	-o nfsvers=4
igroupName	Astra Tridentはノード単位で作成、管理します igroups として netappdvp。 この値は変更したり省略したりすることはできません。 でのみ使用できます **ontap-san** ドライバ。	netappdvp
limitVolumeSize	最大要求可能ボリュームサイズとqtree 親ボリュームサイズ。*のため ontap-nas-economy また、このオプションを使用すると、作成するFlexVol *のサイズも制限されます。	300g
qtreesPerFlexvol	FlexVol あたりの最大 qtree 数は [50 、 300] の範囲で指定する必要があります。デフォルトは 200 です。 *のため ontap-nas-economy ドライバ。このオプションを使用すると、FlexVol あたりの最大qtree数をカスタマイズできます。	300
sanType	サポート対象 **ontap-san** ドライバーのみ。 を使用して選択 iscsi iSCSIの場合または nvme （NVMe/TCPの場合）。	iscsi 空白の場合

作成するすべてのボリュームでデフォルトのオプションを指定しなくても済むようになっています。

オプション	説明	例
spaceReserve	スペースリザベーションモード none （シンプロビジョニング）または volume （シック）	none

オプション	説明	例
snapshotPolicy	使用するSnapshotポリシー。デフォルトはです none	none
snapshotReserve	Snapshotリザーブの割合。デフォルトはONTAP のデフォルトをそのまま使用する場合はです	10
splitOnClone	作成時に親からクローンをスプリットします。デフォルトはです false	false
encryption	新しいボリュームでNetApp Volume Encryption（NVE）を有効にします。デフォルトはです false。このオプションを使用するには、クラスタで NVE のライセンスが設定され、有効になっている必要があります。 NAEがバックエンドで有効になっている場合は、Astra Tridentでプロビジョニングされたすべてのボリュームがに有効になります。 詳細については、以下を参照してください。 "Astra TridentとNVEおよびNAEの相互運用性"。	正しいです
unixPermissions	プロビジョニングされたNFSボリュームのNASオプション。デフォルトはです 777	777
snapshotDir	にアクセスするためのNASオプション .snapshot ディレクトリ。デフォルトはです false	true
exportPolicy	NFSエクスポートポリシーで使用するNASオプション。デフォルトはです default	default
securityStyle	プロビジョニングされたNFSボリュームにアクセスするためのNASオプション。 NFSのサポート mixed および unix セキュリティ形式デフォルトはです unix。	unix
fileSystemType	ファイルシステムタイプを選択するためのSANオプション。デフォルトはです ext4	xfs
tieringPolicy	使用する階層化ポリシー。デフォルトはです none; snapshot-only ONTAP 9.5より前のSVM-DR構成の場合	none

スケーリングオプション

。 `ontap-nas` および `ontap-san` ドライバによって、DockerボリュームごとにONTAP FlexVol が作成されます。ONTAP では、クラスタノードあたり最大 1 、 000 個の FlexVol がサポートされます。クラスタの最大 FlexVol 数は 12 、 000 です。この制限内にDockerボリュームの要件が収まる場合は、を参照してください `ontap-nas` FlexVolで提供されるDockerボリューム単位のSnapshotやクローニングなどの機能が追加されているため、NAS解決策 がドライバとして推奨されます。

FlexVol の制限で対応できない数のDockerボリュームが必要な場合は、を選択します `ontap-nas-economy` または `ontap-san-economy` ドライバ。

。 `ontap-nas-economy` ドライバによって、自動管理されるFlexVolのプール内に、Dockerボリュームがが作成される。qtree の拡張性は、クラスタノードあたり最大 10 、 000 、クラスタあたり最大 2 、 40 、 000 で、一部の機能を犠牲にすることで大幅に向上しています。。 `ontap-nas-economy` ドライバは、Dockerボリューム単位のスナップショットやクローニングをサポートしていません。

。 `ontap-nas-economy` ドライバは現在Docker Swarmではサポートされていません。Swarmは複数のノード間でのボリューム作成のオーケストレーションを行わないためです。

。 `ontap-san-economy` ドライバによって、自動で管理されるFlexVolの共有プール内にDockerボリュームがONTAP LUNとして作成される。この方法により、各 FlexVol が 1 つの LUN に制限されることはなく、 SAN ワークロードのスケーラビリティが向上します。ストレージアレイに応じて、 ONTAP はクラスタあたり最大 16384 個の LUN をサポートします。このドライバは、ボリュームが下位の LUN であるため、 Docker ボリューム単位の Snapshot とクローニングをサポートします。

を選択します `ontap-nas-flexgroup` 数十億個のファイルを含むペタバイト規模に拡張可能な1つのボリュームへの並列処理能力を高めるドライバ。FlexGroup のユースケースとしては、 AI / ML / DL 、ビッグデータと分析、ソフトウェアのビルド、ストリーミング、ファイルリポジトリなどが考えられます。Trident は、 FlexGroup ボリュームのプロビジョニング時に SVM に割り当てられたすべてのアグリゲートを使用します。Trident での FlexGroup のサポートでは、次の点も考慮する必要があります。

- ONTAP バージョン 9.2 以降が必要です。
- 本ドキュメントの執筆時点では、 FlexGroup は NFS v3 のみをサポートしています。
- SVM で 64 ビットの NFSv3 ID を有効にすることを推奨します。
- 推奨されるFlexGroupメンバー/ボリュームの最小サイズは100GiBです。
- FlexGroup Volume ではクローニングはサポートされていません。

FlexGroup と FlexGroup に適したワークロードの詳細については、を参照してください "NetApp FlexGroup Volume Best Practices and Implementation Guide"。

同じ環境で高度な機能と大規模な拡張性を実現するために、を使用して、Docker Volume Pluginの複数のインスタンスを実行できます `ontap-nas` を使用しています `ontap-nas-economy`。

ONTAP 構成ファイルの例

<code>ontap-nas</code> ドライバの**NFS**の例

```
{
    "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> ドライバの**NFS**の例

```
{
    "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> ドライバの**NFS**の例

```
{
    "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> ドライバの**iSCSI**の例

```
{
    "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> ドライバの**NFS**の例

```
{
    "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> ドライバの**NVMe/TCP**の例

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

Element ソフトウェアの設定

Element ソフトウェア（ NetApp HCI / SolidFire ）を使用する場合は、グローバルな設定値のほかに、以下のオプションも使用できます。

オプション	説明	例
`Endpoint`	\https：//<login>：<password>@<mvip>/ JSON -RPC /<element-version>	https://admin:admin@192.168.160.3/json-rpc/8.0
`SVIP`	iSCSI の IP アドレスとポート	10.0.0.7 ： 3260
`TenantName`	使用する SolidFire テナント（見つからない場合に作成）	`docker`
`InitiatorIFace`	iSCSI トラフィックをデフォルト以外のインターフェイスに制限する場合は、インターフェイスを指定します	`default`
`Types`	QoS の仕様	以下の例を参照してください

オプション	説明	例
`LegacyNamePrefix`	アップグレードされた Trident インストールのプレフィックス。1.3.2 より前のバージョンのTridentを使用していて、既存のボリュームでアップグレードを実行した場合は、volume-nameメソッドでマッピングされた古いボリュームにアクセスするためにこの値を設定する必要があります。	`netappdvp-`

。 `solidfire-san` ドライバはDocker Swarmをサポートしていません。

Element ソフトウェア構成ファイルの例

```
{
    "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
            }
        }
    ]
}
```

既知の問題および制限事項

Astra Trident と Docker を使用する際の既知の問題と制限事項について説明しています。

Trident Docker Volume Plugin を旧バージョンから **20.10** 以降にアップグレードすると、該当するファイルエラーまたはディレクトリエラーなしでアップグレードが失敗します。

回避策

1. プラグインを無効にします。

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

2. プラグインを削除します。

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

3. 追加を指定してプラグインを再インストールします `config` パラメータ

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

ボリューム名は 2 文字以上にする必要があります。

これは Docker クライアントの制限事項です。クライアントは、 1 文字の名前を Windows パスと解釈します。 "バグ 25773 を参照"。

Docker Swarm には、 Astra Trident がストレージやドライバのあらゆる組み合わせでサポートしないようにする一定の動作があります。

- Docker Swarm は現在、ボリューム ID ではなくボリューム名を一意のボリューム識別子として使用します。
- ボリューム要求は、 Swarm クラスタ内の各ノードに同時に送信されます。
- ボリュームプラグイン（ Astra Trident を含む）は、 Swarm クラスタ内の各ノードで個別に実行する必要があります。
  これは、ONTAP の仕組みとの仕組みによるものです `ontap-nas` および `ontap-san` ドライバ機能は、これらの制限内で動作できるようになる唯一の機能です。

その他のドライバには、競合状態などの問題があります。このような問題が発生すると、ボリュームを同じ名前で異なる ID にする機能が Element に備わっているため、「勝者」を明確にせずに 1 回の要求で大量のボリュームを作成できるようになります。

ネットアップは Docker チームにフィードバックを提供しましたが、今後の変更の兆候はありません。

FlexGroup をプロビジョニングする場合、プロビジョニングする FlexGroup と共通のアグリゲートが 2 つ目の FlexGroup に 1 つ以上あると、 ONTAP は 2 つ目の FlexGroup をプロビジョニングしません。

著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、http://www.netapp.com/TMに記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。