



# Trident for Docker

## Trident

NetApp  
February 02, 2026

# 目次

Trident for Docker	1
導入の前提条件	1
要件を確認します	1
NVMeツール	3
FCツール	4
Tridentの導入	6
Docker Managed Plugin メソッド（バージョン 1.13 / 17.03 以降）	6
従来の方法（バージョン 1.12 以前）	8
システム起動時にTridentを起動する	10
Trident をアップグレードまたはアンインストールする	10
アップグレード	11
をアンインストールします	12
ボリュームを操作します	12
ボリュームを作成します	12
ボリュームを削除します	13
ボリュームのクローンを作成します	13
外部で作成されたボリュームにアクセス	15
ドライバ固有のボリュームオプション	15
ログを収集します	20
トラブルシューティング用にログを収集する	20
一般的なトラブルシューティングのヒント	21
複数のTridentインスタンスの管理	21
Docker Managed Plugin（バージョン 1.13 / 17.03 以降）の手順	21
従来の（バージョン 1.12 以前）の場合の手順	22
ストレージ構成オプション	22
グローバル構成オプション	22
ONTAP の設定	23
Element ソフトウェアの設定	32
既知の問題および制限事項	34
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。	35
ボリューム名は 2 文字以上にする必要があります。	35
Docker Swarmには特定の動作があり、ストレージとドライバの組み合わせごとにTridentがDocker Swarmをサポートできないようになっています。	35
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。	35

# Trident for Docker

## 導入の前提条件

Tridentを導入する前に、必要なプロトコルの前提条件をホストにインストールして設定する必要があります。

### 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
```



「/etc/multipath.conf」に「find \_ multipaths no」が「defVaults」に含まれていることを確認します。

5. 「iscsid」と「multipathd」が実行されていることを確認します。

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

6. 'iSCSI' を有効にして開始します

```
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
```



「/etc/multipath.conf」に「find \_ multipaths no」が「defVaults」に含まれていることを確認します。

### 5. 「open-iSCSI」 および「マルチパスツール」が有効で実行されていることを確認します。

```
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
```

## FC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、FCツールをインストールします。

- FC PVSでRHEL / Red Hat Enterprise Linux CoreOS (RHCOS) を実行するワーカーノードを使用する場合は、StorageClassでmountOptionを指定してインラインのスペース再生を実行します discard。を参照してください "[Red Hat のドキュメント](#)"。

## 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 device-mapper-multipath
```

2. マルチパスを有効化：

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



「/etc/multipath.conf」に「find \_ multipaths no」が「defVaults」に含まれていることを確認します。

3. が実行中であることを確認し `multipathd` ます。

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

## 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 lsscsi sg3-utils multipath-tools scsitools
```

2. 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
```



「/etc/multipath.conf」に「find \_ multipaths no」が「defVaults」に含まれていることを確認します。

3. が有効で実行中であることを確認し `multipath-tools` ます。

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

# Tridentの導入

Trident for Dockerは、NetAppストレージプラットフォームの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。

同じホスト上でTridentの複数のインスタンスを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、Tridentを導入する準備が整います。

## 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

従来のデーモンメソッドでDocker 1.13/17.03より前のTridentを使用している場合は、マネージプラグインメソッドを使用する前に、Tridentプロセスを停止して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 'config.json' ですが 'ファイル名に config オプションを指定することで' 選択した任意の名前を使用できます構成ファイルは ' ホスト・システムの /etc/netappdvp ディレクトリに格納されている必要があります
- 「log-level」：ログレベルを指定します（「debug」、「info」、「warn」、「error」、「fatal」



」)。デフォルトは「info」です。

- ° ebug: デバッグロ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。

- 構成ファイルの場所を作成します。

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

- 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. 管理プラグインシステムを使用してTridentを起動します。を、使用しているプラグインのバージョン (xxx.xx.x) に置き換えます <version>。

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

3. Tridentを使用して、構成済みシステムのストレージを消費します。

- 「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
```

- 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/10.0/trident-  
installer-25.10.0.tar.gz  
tar xzf trident-installer-25.10.0.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
```

- 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。

デーモンを起動したら、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。

- ボリュームを作成します

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

- 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
```

- 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Tridentを起動する

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 '/usr/lib/systemd/system' ディレクトリ内のファイルの名前によって異なります

```
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
```



ユニット・ファイルを変更するときは '変更を認識するために 'systemctl daemon-reload コマンドを実行します

## 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、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、`docker volume`プラグインへのコマンドが正常に実行されず、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

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
```



Tridentの18.01リリースは、nDVPに代わるものです。イメージからイメージに`netapp/trident-plugin`直接アップグレードする必要がある`netapp/ndvp-plugin`です。

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Trident（20.10より前のバージョン）からTrident 20.10以降にアップグレードすると、エラーが発生することがあります。詳細については、[を参照してください](#) **"既知の問題"**。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、追加のconfigパラメータを渡して必要なTridentバージョンをインストールする必要があります。

```
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

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

必要に応じてTridentドライバ名を指定した標準コマンドを使用すると、ボリュームの作成、クローニング、および削除を簡単に実行でき `docker volume` ます。

### 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。ドライバーを使用して20 GiBのボリュームを作成するには、次の例を参照してください。

```
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
```



「olidfire-san」ドライバーを使用する場合、上記の例ではボリュームを削除してページします。

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

使用する際は ontap-nas、ontap-san、そして `solidfire-san` ストレージドライバーを使用すると、Trident はボリュームを複製できます。使用する際は `ontap-nas-flexgroup` または `ontap-nas-economy` ドライバーの場合、クローン作成はサポートされません。既存のボリュームから新しいボリュームを作成すると、新しいスナップショットが作成されます。

- 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

外部で作成したブロックデバイス（またはそのクローン）には、パーティションがなく、ファイルシステムがTridentでサポートされている場合（例：-formatted /dev/sdc1`はTrident経由でアクセスできません）にのみ、Trident \*を使用してコンテナからアクセスできます `ext4。

## 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。CLI の操作中に '-' 演算子を使用して ' オプションと値を指定しますこれらは、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、iSCSI、およびFCのボリューム作成オプションには、次のものがあります。

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。
「平和のための準備」を参照してください	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 'none( シン・プロビジョニング )' と 'volume( シック・プロビジョニング )' です
「ナップショットポリシー」	Snapshot ポリシーが目的の値に設定されます。デフォルトは、`none`ボリュームのSnapshotは自動的に作成されません。ストレージ管理者が変更しないかぎり、「default」というポリシーは、毎時6個、日次2個、週次2個のSnapshotを作成および保持するすべてのONTAPシステムに存在します。ボリューム内の任意のディレクトリを参照することで、Snapshot に保存されているデータをリカバリでき `snapshot` ます。
「スナップショット予約」	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。

オプション	説明
'plitOnClone	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 <code>false</code> 。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。
「暗号化」	<p>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「false」です。このオプションを使用するには、クラスターで 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。</p> <p>バックエンドで NAE が有効になっている場合、Trident でプロビジョニングされたすべてのボリュームで NAE が有効になります。</p> <p>詳細については、を参照してください<a href="#">"Trident と NVE および NAE との連携"</a>。</p>
階層ポリシー	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 'アクセス権は '--rwxr-xr-x' または数値表記 0755 に設定され 'root' は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。
「スナップショット方向」	これをに設定します <code>true</code> がを作成します <code>.snapshot</code> 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 <code>false</code> の可視性を意味します <code>.snapshot</code> 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、 <code>.snapshot</code> ディレクトリが表示されます。

オプション	説明
「 exportPolicy 」と入力します	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「デフォルト」です。
'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「unix」です。有効な値は「unix」と「immimixed」です。

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 」です	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「ext4」です。有効な値は「ext3」、「ext4」、「xfs」です。
「平和の配分」	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。

- 10 GiB のボリュームを作成します。

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

- スナップショット付きの 100 GiB 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 20 MiB です。

スナップショット予約が指定されておらず、スナップショットポリシーがの場合、`none` Tridentは0%のスナ

ップショット予約を使用します。

- 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ポリシーを使用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%）をそのまま使用します。

```
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。ボリュームが作成されると '-o type=service\_level' という命名法を使用して 'ボリュームに関連付けられた QoS ポリシーが指定されます

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。デフォルトは 1 GiB または構成エントリ... 「defaults」: {"size": "5G"}。
「ブロックサイズ」	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。

- 10 GiB のゴールド ボリュームを作成します。

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

- 100 GiB の 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式を使用して（コマンドを使用して）Tridentを実行している場合は `docker plugin`、次のように表示します。

```
docker plugin ls
```

ID	NAME	DESCRIPTION
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。

「デーモンからのエラー応答：ダイヤル UNIM/run/docx/plugins/<id>/NetApp/smock: 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。

- コンテナへの PV のマウントに問題がある場合は 'rpcbind' がインストールされていて実行されていることを確認してください。ホスト OS に必要なパッケージ・マネージャを使用して 'rpcbind' が実行されているかどうかを確認します。rpcbind サービスのステータスは 'systemctl ステータス rpcbind' またはそれに相当する処理を実行することで確認できます。

## 複数のTridentインスタンスの管理

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の鍵は、コンテナ化されたプラグインでは「--alias」オプション、ホストで Trident をインスタンス化する場合は「--volume-driver」オプションを使用して、それぞれ異なる名前を指定することです。

### 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

Trident構成で使用可能な設定オプションを参照してください。

### グローバル構成オプション

これら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Trident構成に適用されます。

オプション	説明	例
「バージョン」	構成ファイルのバージョン番号	1



オプション	説明	例
'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_
「limitVolumeSize」と入力します	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



Elementバックエンドには（デフォルトを含めて）使用しない`storagePrefix`でください。デフォルトでは、`solidfire-san`ドライバはこの設定を無視し、プレフィックスは使用しません。NetAppでは、Dockerボリュームマッピングに特定のtenantIDを使用するか、名前のマングリングが使用されている可能性がある場合には、DockerからDockerのバージョン、ドライバ情報、およびraw名が入力された属性データを使用することを推奨しています。

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 の設定

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
「管理 LIF」	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
「重複排除	<p>プロトコル LIF の IP アドレス。</p> <ul style="list-style-type: none"> <li>• ONTAP NASドライバ* ：NetAppでは、を指定することを推奨しています dataLIF。指定しない場合、TridentはSVMからデータLIFをフェッチします。NFSの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）を指定すると、ラウンドロビンDNSを作成して複数のデータLIF間で負荷を分散できます。</li> <li>• ONTAP SANドライバ*：iSCSIまたはFCには指定しないでください。Tridentは、を使用して"ONTAP の選択的LUNマップ"、マルチパスセッションの確立に必要なiSCSI LIFまたはFC LIFを検出します。が明示的に定義されている場合は、警告が生成され`dataLIF`ます。</li> </ul>	10.0.0.2
'VM'	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
「ユーザ名」	ストレージデバイスに接続するユーザ名	vsadmin
「password」と入力します	ストレージ・デバイスに接続するためのパスワード	secret
「集約」	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）。ドライバの場合 ontap-nas-flexgroup、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
「AggreglimitateUsage」と入力します	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
「nfsvMountOptions」のように入力します	NFSマウントオプションをきめ細かく制御します。デフォルトは「-o nfsvers=3」です。ドライバと`ontap-nas-economy`ドライバでのみ使用でき`ontap-nas`ます。" <a href="#">ここでは、NFSホストの設定情報を参照してください</a> "です。	-o nfsvers=4
「igroupName」と入力します	Tridentでは、ノードごとにASを`netappdvp`作成および管理し`igroups`ます。  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 <b>ontap-san</b> ドライバ。	netappdvp
「limitVolumeSize」と入力します	要求可能な最大ボリュームサイズ。	300g
qtreesPerFlexvol`	FlexVolあたりの最大mtree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mtree数をカスタマイズできます。	300
sanType	*ドライバでのみサポートされている ontap-san`ます。*iSCSI、`nvme`NVMe/TCP、または`fc`SCSI over Fibre Channel (FC; SCSI over Fibre Channel) に対してを選択します `iscsi。	iscsi 空白の場合
limitVolumePoolSize	*`ontap-san-economy`および`ontap-san-economy`ドライバでのみサポートされています。*ONTAP ONTAP NASエコノミードライバおよびONTAP SANエコノミードライバでFlexVolサイズを制限します。	300g

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
「平和のための準備」を参照してください	スペースリザーベーションモード <code>none</code> （シンプロビジョニング）または <code>volume</code> （シック）	「NONE」
「ナプショットポリシー」	使用するSnapshotポリシー。デフォルトは <code>none</code>	「NONE」
「スナップショット予約」	Snapshotリザーブの割合。デフォルトはONTAPのデフォルトを使用する場合は""です。	10
'plitOnClone	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 <code>false</code>	「偽」
「暗号化」	<p>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「false」です。このオプションを使用するには、クラスターで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。</p> <p>バックエンドでNAEが有効になっている場合、TridentでプロビジョニングされたすべてのボリュームでNAEが有効になります。</p> <p>詳細については、を参照してください"<a href="#">TridentとNVEおよびNAEとの連携</a>"。</p>	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 <code>777</code>	777
「スナップショット方向」	ディレクトリにアクセスするためのNASオプション <code>.snapshot</code> 。	NFSv4の場合は「true」 NFSv3の場合は「false」
「exportPolicy」と入力します	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 <code>default</code>	default
'ecurityStyle'	<p>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。</p> <p>NFSのサポート <code>mixed</code> および <code>unix</code> セキュリティ形式デフォルトは <code>unix</code>。</p>	unix
「filesystemtype」です	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 <code>ext4</code>	xfs
階層ポリシー	使用する階層化ポリシー。デフォルトは <code>none</code> 。	「NONE」

オプション	説明	例
skipRecoveryQueue	ボリュームの削除中は、ストレージ内のリカバリキューをバイパスし、ボリュームを直ちに削除します。	``

## スケーリングオプション

ドライバと `ontap-san` ドライバを使用すると、`ontap-nas` Docker ボリュームごとに ONTAP FlexVol が作成されます。ONTAP は、クラスタノードあたり最大 1000 個の FlexVol、最大 12,000 個の FlexVol をサポートします。Docker ボリュームの要件がこの制限の範囲内に収まる場合、`ontap-nas` Docker ボリューム単位の Snapshot やクローニングなど、FlexVol によって提供される追加機能のため、ドライバが推奨される NAS ソリューションです。

FlexVol の制限で対応できる容量よりも多くの Docker ボリュームが必要な場合は、「ONTAP - NAS - エコノミー」または「ONTAP - SAN - エコノミー」ドライバを選択します。

`ontap-nas-economy` ドライバは、自動的に管理される FlexVol ボリュームのプール内に ONTAP qtrees として Docker ボリュームを作成します。qtrees 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。この `ontap-nas-economy` ドライバは、Docker ボリューム単位の Snapshot やクローニングをサポートしていません。



Docker Swarm で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が行われないため、この `ontap-nas-economy` ドライバは現在 Docker Swarm でサポートされていません。

`ontap-san-economy` ドライバは、自動的に管理される FlexVol ボリュームの共有プール内に ONTAP LUN として Docker ボリュームを作成します。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。

ドライバを選択する `ontap-nas-flexgroup` と、数十億個のファイルを含むペタバイト規模まで拡張可能な単一ボリュームへの並列処理を強化できます。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 では、FlexGroup ボリュームのプロビジョニング時に、SVM に割り当てられているすべてのアグリゲートが使用されます。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 100 GiB です。
- FlexGroupボリュームではクローニングはサポートされていません。

FlexGroupに適したFlexGroupとワークロードについては、を参照してください ["NetApp FlexGroupボリュームベストプラクティスおよび実装ガイド"](#)。

1つの環境で高度な機能と大規模な機能を利用するには、を使用して、別のを使用 `ontap-nas-economy` して、Docker Volume Pluginの複数のインスタンスを実行し `ontap-nas` ます。

### Trident用のカスタムONTAPロール

Tridentで処理を実行するためにONTAP adminロールを使用する必要がないように、最小Privilegesを持つONTAPクラスタロールを作成できます。Tridentバックエンド構成にユーザ名を含めると、Trident作成したONTAPクラスタロールが使用されて処理が実行されます。

Tridentカスタムロールの作成の詳細については、を参照してください ["Tridentカスタムロールジェネレータ"](#)。

## ONTAP CLIノシヨウ

1. 次のコマンドを使用して新しいロールを作成します。

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

2. Tridentユーザのユーザ名を作成します。

```
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. ユーザにロールをマッピングします。

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

## System Managerの使用

ONTAPシステムマネージャで、次の手順を実行します。

1. カスタムロールの作成：
  - a. クラスタレベルでカスタムロールを作成するには、\*[クラスタ]>[設定]\*を選択します。  
  
(または) SVMレベルでカスタムロールを作成するには、\*[ストレージ]>[Storage VM]>[設定]>[ユーザとロール]\*を選択し`required SVM`ます。
  - b. の横にある矢印アイコン (→\*) を選択します。
  - c. [Roles]\*で[+Add]\*を選択します。
  - d. ロールのルールを定義し、\*[保存]\*をクリックします。
2. ロールを**Trident**ユーザにマップする:+[ユーザとロール]ページで次の手順を実行します。
  - a. で[アイコンの追加]\*を選択します。
  - b. 必要なユーザ名を選択し、\* Role \*のドロップダウンメニューでロールを選択します。
  - c. [保存 ( Save ) ] をクリックします。

詳細については、次のページを参照してください。

- ["ONTAPの管理用のカスタムロール"または"カスタムロールの定義"](#)
- ["ロールとユーザを使用する"](#)

## ONTAP 構成ファイルの例

### `ontap-nas` 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"
  }
}
```

### `ontap-nas-flexgroup` 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
}
```

## SCSI-SAN <code> ONTAP </code> ドライバの例

```
{
  "version": 1,
  "backendName": "ontap-san-backend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "sanType": "fc",
  "svm": "trident_svm",
  "username": "vsadmin",
  "password": "password",
  "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
「エンドポイント」	<code>\&lt;a href="https://&amp;lt;login&amp;gt;:&amp;lt;password&amp;gt;@&amp;lt;mvip&amp;gt;/json-rpc/&amp;lt;element-version&amp;gt;" class="bare"&gt;https://&amp;lt;login&amp;gt;:&amp;lt;password&amp;gt;@&amp;lt;mvip&amp;gt;/json-rpc/&amp;lt;element-version&amp;gt;&lt;/a&gt;</code>	<code>\ https://admin:admin@192.168.160.3/json-rpc/8.0</code>

オプション	説明	例
「VIP」	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
「InitiatorIFCace」	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
「タイプ」	QoS の仕様	以下の例を参照してください
「LegacyNamePrefix」のように入力します	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-

「olidfire -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

ここでは、TridentでDockerを使用する場合の既知の問題および制限事項について説明します。

**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. 追加の 'config' パラメータを指定して 'プラグインを再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
```

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**には特定の動作があり、ストレージとドライバの組み合わせごとに**TridentがDocker Swarm**をサポートできないようになっています。

- Docker Swarm は現在、ボリューム ID ではなくボリューム名を一意的な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（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。ONTAPの動作方法とドライバと `ontap-san` ドライバの機能により、`ontap-nas`これらの制限内で動作できるのはこれらのドライバだけです。

残りのドライバはレースコンディションなどの問題の影響を受け、明確な「勝者」なしで1回の要求で多数のボリュームが作成される可能性があります。たとえば、Elementには、同じ名前で複数のIDを使用できる機能があります。

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权を有し、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权については、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。