



Docker向けTrident

Trident

NetApp
October 29, 2026

This PDF was generated from <https://docs.netapp.com/ja-jp/trident-2606/trident-docker/prereqs-docker.html> on October 29, 2026. Always check docs.netapp.com for the latest.

目次

Docker向けTrident	1
導入の前提条件	1
要件を確認する	1
NVMeツール	3
FCツール	4
Tridentを導入	5
Docker マネージド プラグイン方式（バージョン 1.13/17.03 以降）	5
従来の方法（バージョン1.12以前）	7
システム起動時にTridentを起動する	9
Tridentのアップグレードまたはアンインストール	10
Upgrade	10
アンインストール	11
ボリュームの操作	12
ボリュームの作成	12
ボリュームを削除する	12
ボリュームをクローニングする	12
外部で作成されたボリュームにアクセスする	14
ドライバー固有のボリュームオプション	14
ログを収集する	19
トラブルシューティングのためにログを収集する	19
一般的なトラブルシューティングのヒント	20
複数のTridentインスタンスを管理	20
Docker マネージドプラグインの手順（バージョン 1.13/17.03 以降）	20
従来の手順（バージョン1.12以前）	21
ストレージ構成オプション	21
グローバル設定オプション	21
ONTAPの構成	22
Element ソフトウェア構成	30
既知の問題と制限事項	32
Trident Docker Volume Plugin を古いバージョンから 20.10 以降にアップグレードすると、「そのようなファイルまたはディレクトリはありません」というエラー が発生し、アップグレードが失敗します。	32
ボリューム名は 2 文字以上である必要があります。	32
Docker Swarmには、 Tridentがあらゆるストレージとドライバの組み合わせでサポートすることを妨げる特定の動作があり ます。	32
FlexGroupがプロビジョニングされている場合、2番目のFlexGroupがプロビジョニングされているFI exGroupと1つ以上のアグリゲートを共有している場合、ONTAPは2番目のFlexGroupをプロビジョニ ングしません。	33

Docker向けTrident

導入の前提条件

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

要件を確認する

- 導入がすべての["要件"](#)を満たしていることを確認します。
- 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のバージョンが古い場合は、["インストールまたはアップデートする"](#)。

```
docker --version
```

- プロトコルの前提条件がホストにインストールされ、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
```



ボリュームをコンテナに接続するときに障害が発生しないように、NFSツールをインストールした後、ワーカーノードを再起動します。

iSCSIツール

オペレーティングシステム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します：

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

4. マルチパスを有効にする：

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



`etc/multipath.conf` に `find\_multipaths no` が `defaults` の下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. 有効化して起動 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 のバージョンが 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します：

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

4. マルチパスを有効にする：

```
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`defaults`の下に含まれていることを確認します。

5. `open-iscsi`と`multipath-tools`が有効になっていて実行中であ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の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、またはカーネルバージョンで NVMe パッケージが利用できない場合は、ノードのカーネルバージョンを NVMe パッケージを含むバージョンに更新する必要がある場合があります。

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 PVを使用するRHEL／Red Hat Enterprise Linux CoreOS（RHCOS）上のワーカーノードを利用する場合、インラインスペースリクレーンを実行するために、discard StorageClassでmountOptionを指定してください。["Red Hat ドキュメント"](#)を参照してください。

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. マルチパスを有効にする：

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



‘etc/multipath.conf’に‘find_multipaths no’が‘defaults’の下に含まれていることを確認します。

3. ‘multipathd’が実行されていることを確認します：

```
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. マルチパスを有効にする：

```
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`defaults`の下に含まれていることを確認します。

3. `multipath-tools`が有効になっていて実行中であることを確認します：

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

Tridentを導入

Trident for Dockerは、NetAppストレージプラットフォーム向けのDockerエコシステムとの直接統合を提供します。ストレージプラットフォームからDockerホストへの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、将来的に追加のプラットフォームを追加するためのフレームワークを備えています。

複数のTridentインスタンスを同じホスト上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およびストレージタイプへの同時接続が可能になり、Dockerボリュームに使用されるストレージをカスタマイズできるようになります。

要件

"[導入の前提条件](#)"を参照してください。前提条件が満たされ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 マネージド プラグイン方式（バージョン 1.13/17.03 以降）

開始する前に



Docker 1.13/17.03より前のバージョンで従来のデーモン方式でTridentを使用したことがある場合は、マネージドプラグイン方式を使用する前に、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てください。

```
docker --version
```

バージョンが古い場合は、["インストールまたはインストールを更新する"](#)。

手順

1. 構成ファイルを作成し、次のように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、`config` オプションでファイル名を指定することで任意の名前を使用できます。設定ファイルは、ホストシステムの `/etc/netappdvp` ディレクトリに配置する必要があります。
- `log-level`：ログレベルを指定する(`debug`、`info`、`warn`、`error`、`fatal`)。デフォルトは `info` です。
- `debug`：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。

i. 構成ファイルの場所を作成します：

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

ii. 設定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. 管理されたプラグイン システムを使用して Trident を起動します。`<version>`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。

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

3. 構成されたシステムからストレージを消費するために Trident の使用を開始します。

- 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. コンテナの起動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）

開始する前に

1. Docker バージョン 1.10 以降がインストールされていることを確認してください。

```
docker --version
```

バージョンが古い場合は、インストールを更新してください。

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

または、["ディストリビューションの手順に従ってください"](#)。

2. NFS および/または iSCSI がシステムに設定されていることを確認します。

手順

1. NetApp Docker ボリューム プラグイン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解凍します：

```
wget  
https://github.com/NetApp/trident/releases/download/10.0/trident-  
installer-26.06.0.tar.gz  
tar xzf trident-installer-26.06.0.tar.gz
```

- b. bin 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. 設定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、設定ファイルを作成したら、必要な設定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
```



指定されない限り、ボリューム ドライバーのデフォルト名は「netapp」です。

デーモンを起動したら、Docker CLI インターフェー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. コンテナを起動するときに Docker ボリュームをプロビジョニングします：

```
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Tridentを起動する

systemdベースのシステムのサンプルユニットファイルは、Gitリポジトリの`contrib/trident.service.example`にあります。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する必要があります。

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

2. ファイルを編集し、ドライバー名と一致するように説明（2行目）を変更し、環境を反映するように構成ファイルパス（9行目）を変更します。
3. 変更を取り込むために systemd 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`ディレクトリ内でファイルに付けた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. ステータスを表示します。

```
systemctl status trident
```



ユニットファイルを変更するたびに、変更を認識させるために `systemctl daemon-reload` コマンドを実行します。

Tridentのアップグレードまたはアンインストール

使用中のボリュームに影響を与えることなく、Docker 用 Trident 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中、プラグインに対する `docker volume` コマンドが成功しない短い期間があり、プラグインが再び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、これは数秒の問題です。

Upgrade

Docker用のTridentをアップグレードするには、以下の手順を実行してください。

手順

1. 既存のボリュームを一覧表示します：

```
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 を置き換えます。`netapp/ndvp-plugin` イメージから `netapp/trident-plugin` イメージに直接アップグレードする必要があります。

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てい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以前）からTrident 20.10以降にアップグレードする場合、エラーが発生する可能性があります。詳細については、["既知の問題"](#)を参照してください。エラーが発生した場合は、まずプラグインを無効にし、次にプラグインを削除してから、追加の設定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してください：`docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all -permissions config=config.json`

アンインストール

Docker用のTridentをアンインストールするには、以下の手順を実行してください。

手順

1. プラグインによって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
```

ボリュームの操作

標準の `docker volume` コマンドを使用して、必要に応じてTridentドライバー名を指定することで、ボリュームを簡単に作成、複製、削除できます。

ボリュームの作成

- デフォルト名を使用してドライバー付きのボリュームを作成します：

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

- 特定のTridentインスタンスでボリュームを作成：

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



"options"を指定しない場合は、ドライバー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（例：10G、20GB、3TiB）を持つ整数値を含む文字列として表されます。単位が指定されていない場合、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）で表すことができます。省略単位には2の累乗を使用します（G = GiB、T = TiB、...）。

ボリュームを削除する

- 他の Docker ボリュームと同じようにボリュームを削除します：

```
docker volume rm firstVolume
```



`solidfire-san` ドライバを使用する場合、上記の例ではボリュームが削除およびパーズされます。

Docker用のTridentをアップグレードするには、以下の手順を実行してください。

ボリュームをクローニングする

```
`ontap-nas`、`ontap-san`、および `solidfire-
```

san`ストレージドライバを使用する場合、Tridentはボリュームをクローニングできます。
`ontap-nas-flexgroup`または `ontap-nas-economy` ドライバーを使用する場合、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。

- ボリュームを検査してスナップショットを列挙します：

```
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。これにより、新しいスナップショットが作成されます：

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

- ボリューム上の既存のスナップショットから新しいボリュームを作成します。これにより、新しいスナップショットは作成されません：

```
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する

コンテナは、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に*のみ*、Tridentを使用して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）にアクセスできます（例：`ext4`でフォーマットされた`/dev/sdc1`は、Trident経由ではアクセスできません）。

ドライバー固有のボリュームオプション

各ストレージドライバには異なるオプションセットがあり、ボリュームの作成時に指定して結果をカスタマイズできます。構成された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。

ボリューム作成操作中にこれらのオプションを使用するのは簡単です。CLI 操作中に`-o`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を上書きします。

ONTAPボリュームオプション

NFS、iSCSI、FC 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	概要
size	ボリュームのサイズ。デフォルトは 1 GiB 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。デフォルトはシンプロビジョニングです。有効な値は 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）です。

オプション	概要
snapshotPolicy	これにより、スナップショットポリシーが目的の値に設定されます。デフォルトは `none` で、ボリュームのスナップショットは自動的に作成されません。ストレージ管理者によって変更されない限り、「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つの時間別スナップショット、2つの日次スナップショット、2つの週次スナップショットを作成して保持します。スナップショットに保存されたデータは、ボリューム内の任意のディレクトリにある `.snapshot` ディレクトリを参照することで復元できます。
snapshotReserve	これにより、スナップショットの予約が希望のパーセンテージに設定されます。デフォルトでは値なし、つまり ONTAP が snapshotReserve (通常 5%) を選択します (snapshotPolicy を選択した場合)。snapshotPolicy が none の場合は 0% です。すべての ONTAP バックエンドの設定ファイルでデフォルトの snapshotReserve 値を設定でき、ontap-nas-economy を除くすべての ONTAP バックエンドの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ンする場合、これにより ONTAP はクローンを親からすぐに分離します。デフォルトは `false` です。ボリュームのクローン作成の一部のユースケースでは、ストレージ効率を高める機会がほとんどないため、作成後すぐにクローンを親から分割するのが最適です。たとえば、空のデータベースのクローンを作成すると、時間は大幅に節約できますが、ストレージの節約はほとんどないため、クローンをすぐに分割するのが最適です。
encryption	<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、クラスター上で NVE のライセンスを取得し、有効化する必要があります。</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ボリュームはすべて NAE が有効になります。</p> <p>詳細については、次を参照してください: "Trident と NVE および NAE の連携"</p>
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、データが非アクティブ (コールド) になったときにクラウド層に移動するかどうかが決まります。

次の追加オプションは NFS *のみ*に有効です。

オプション	概要
unixPermissions	これは、ボリューム自体の権限セットを制御します。デフォルトでは、権限は ``---rwxr-xr-x` に設定されます。または数値表記では0755となり、`root` が所有者になります。テキスト形式または数値形式のいずれかが機能します。
snapshotDir	これを `true` に設定すると、ボリュームにアクセスするクライアントに `.snapshot` ディレクトリが表示されるようになります。デフォルト値は `false` で、`.snapshot` ディレクトリの表示はデフォルトで無効になっています。公式のMySQLイメージなど一部のイメージは、`.snapshot` ディレクトリが表示されていると期待どおりに機能しません。
exportPolicy	ボリュームに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 `default` 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 スタイルを設定します。デフォルトは `unix` です。有効な値は `unix` と `mixed` です。

次の追加オプションは iSCSI *のみ*に適用されます。

オプション	概要
fileSystemType	iSCSI ボリュームをフォーマットするため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`、および `xfs` です。
spaceAllocation	これを `false` に設定すると、LUNのスペース割り当て機能がオフになります。デフォルト値は `true` で、ボリュームの容量が不足し、ボリューム内のLUNが書き込みを受け付けることができなくなった場合にONTAPがホストに通知することを意味します。このオプションでは、ホストがデータを削除したときにONTAPがスペースを自動的に再利用することもできます。

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20MiBです。

スナップショットリザーブが指定されておらず、スナップショットポリシーが `none` の場合、Tridentはスナップショットリザーブとして0%を使用します。

- スナップショットポリシーとスナップショットリザーブのないボリュームを作成します：

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

- スナップショットポリシーがなく、カスタムスナップショット予約が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
```

- スナップショットポリシーと10%のカスタムスナップショット予約を持つ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
```

- スナップショットポリシーでボリュームを作成し、ONTAPのデフォルトのスナップショット予約量（通常は5%）を使用します：

```
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 ソフトウェアオプションは、ボリュームに関連付けられたサイズとサービス品質（QoS）ポリシーを公開します。ボリュームが作成されると、それに関連付けられた QoS ポリシーは `-o type=service_level` の命名法を使用して指定されます。

Element ドライバーを使用して 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、およびバースト IOPS を指定することです。

その他の Element ソフトウェア 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	概要
size	ボリュームのサイズ。デフォルトは 1 GiB または config エントリです..."defaults":{"size": "5G"}。
blocksize	512または4096を使用します。デフォルトは512または設定エントリ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
      }
    }
  ]
}
```

上記の構成には、Bronze、Silver、Gold の 3 つのポリシー定義があります。これらの名前は任意です。

- 10 GiB 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
```

- 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
```

ログを収集する

トラブルシューティングに役立つログを収集できます。ログを収集するために使用する方法は、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のためにログを収集する

手順

1. 推奨される管理プラグイン方式を使用してTridentを実行している場合（つまり、`docker plugin` コマンドを使用）、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		

標準のログレベルでは、ほとんどの問題を診断できます。それだけでは不十分な場合は、デバッグログを有効にすることができます。

2. デバッグ ログを有効にするには、デバッグ ログを有効にした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は、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>
```

- バイナリ自体をホスト上で実行している場合、ログはホストの `/var/log/netappdvp` ディレクトリで利用できます。デバッグログを有効にするには、プラグインを実行するときに `-debug` を指定します。

一般的なトラブルシューティングのヒント

- 新しいユーザーが遭遇する最も一般的な問題は、プラグインの初期化を妨げる誤った構成です。このような場合、プラグインをインストールまたは有効化しようとすると、次のようなメッセージが表示される可能性があります：

```
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。幸いなことに、このプラグインには包括的なログ機能が組み込まれているため、発生する可能性のあるほとんどの問題の診断に役立ちます。

- PVをコンテナにマウントする際に問題がある場合は、`rpcbind` がインストールされ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ーを使用して、`rpcbind` が実行中かどうかを確認してください。rpcbindサービスのステータスを確認するには、`systemctl status rpcbind` またはそれに相当するコマンドを実行します。

複数のTridentインスタンスを管理

複数のストレージ構成を同時に利用したい場合は、複数の Trident 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ための鍵は、コンテナ化されたプラグインでは `--alias` オプションを使用して、またはホスト上で Trident をインスタンス化する際には `--volume-driver` オプションを使用して、それぞれに異なる名前を付けることです。

Docker マネージドプラグインの手順（バージョン 1.13/17.03 以降）

- エイリアスと設定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
```

- 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
```

- エイリアスをドライバー名として指定してボリュームを作成します。

たとえば、金の数量の場合：

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

たとえば、シルバーのボリュームの場合：

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

従来の手順（バージョン1.12以前）

1. カスタム ドライバー ID を使用して NFS 構成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. カスタムドライバー 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. 各ドライバインスタンスの 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。

グローバル設定オプション

これらの設定オプションは、使用されてい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	概要	例
version	設定ファイルのバージョン番号	1

オプション	概要	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	ボリュームサイズに対するオプションの制限。デフォルト：""（強制されません）	10g



storagePrefix（デフォルトを含む）は Element バックエンドには使用しないでください。デフォルトでは、`solidfire-san`ドライバはこの設定を無視し、プレフィックスを使用しません。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、名前の変更が使用された可能性がある場合は Docker バージョン、ドライバ情報、および Docker からの元の名前が入力された属性データを使用することを推奨します。

作成するボリュームごとにオプションを指定する必要がないように、デフォルトのオプションが用意されています。`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を設定する方法の例については、ONTAP構成セクションを参照してください。

オプション	概要	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の構成

上記のグローバル設定値に加えて、ONTAPを使用する場合は、次の最上位オプションが利用可能です。

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

オプション	概要	例
dataLIF	<p>プロトコル LIF の IP アドレス。</p> <p>ONTAP NAS ドライバー：NetApp では指定することを推奨します dataLIF。指定しない場合、Trident は SVM から dataLIF を取得します。NFS マウント操作に使用する完全修飾ドメイン名 (FQDN) を指定することで、ラウンドロビン DNS を作成し、複数の dataLIF 間で負荷分散を行うことができます。</p> <p>ONTAP SAN ドライバー：iSCSI または FC の場合は指定しないでください。Trident は"ONTAP セレクティブLUNマップ"を使用して、マルチパス セッションを確立するために必要な iSCSI または FC LIF を検出します。警告が発生するのは、`dataLIF` が明示的に定義されている場合です。</p>	10.0.0.2
svm	使用する Storage Virtual Machine (管理LIFがクラスタLIFの場合は必須)	svm_nfs
username	ストレージデバイスに接続するためのユーザ名	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)。`ontap-nas-flexgroup` ドライバーの場合、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は、FlexGroupボリュームのプロビジョニングに使用されます。	aggr1
limitAggregateUsage	オプション：使用率がこのパーセンテージを超える場合はプロビジョニングを失敗します	75%
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。* `ontap-nas` および `ontap-nas-economy` ドライバーでのみ使用可能* " NFSホストの構成情報についてはここを参照してください "。	-o nfsvers=4

オプション	概要	例
igroupName	<p>Tridentはノードごとに `igroups` を `netappdvp` として作成および管理します。</p> <p>この値は変更または省略できません。</p> <ul style="list-style-type: none"> • `ontap-san` ドライバでのみ使用可能*。 	netappdvp
limitVolumeSize	要求可能な最大ボリュームサイズ。	300g
qtreesPerFlexvol	<p>FlexVolあたりの最大qtree数は [50、300] の範囲内である必要があります、デフォルトは200です。</p> <ul style="list-style-type: none"> • `ontap-nas-economy` ドライバの場合、このオプションを使用すると、FlexVolあたりのqtreeの最大数をカスタマイズできます*。 	300
sanType	* `ontap-san` ドライバーのみでサポートされます。* iSCSI の場合は `iscsi`、NVMe/TCP の場合は `nvme`、SCSI over Fibre Channel (FC) の場合は `fcp` を選択します。	iscsi 空白の場合
limitVolumePoolSize	* `ontap-san-economy` および `ontap-san-economy` ドライバーでのみサポートされます。* ONTAP の `ontap-nas-economy` および `ontap-san-economy` ドライバーで FlexVol のサイズを制限します。	300g

作成するボリュームごとに指定する必要がないように、デフォルトのオプションが用意されています。

オプション	概要	例
spaceReserve	スペース予約モード： none (シンプロビジョニング) または volume (シック)	none
snapshotPolicy	使用するSnapshotポリシー。デフォルトは none	none
snapshotReserve	Snapshotリザーブの割合。デフォルトは「」でONTAPのデフォルトを受け入れます	10
splitOnClone	クローン作成時に親からクローンを分割します。デフォルトは false	false

オプション	概要	例
encryption	<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、クラスター上で NVE のライセンスを取得し、有効化する必要があります。</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ボリュームはすべてNAEが有効になります。</p> <p>詳細については、次を参照してください："TridentとNVE および NAE の連携"</p>	true
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 777	777
snapshotDir	<div>`.snapshot`ディレクトリへのアクセスのためのNASオプション。</div>	<div>`.snapshot`ディレクトリへのアクセスを有効または無効にします。</div> <p>有効な値： true、 false。デフォルトの動作は、ONTAPおよびTridentのバージョンによって異なる場合があります。一貫した動作を保証するために、明示的に設定してください。</p>
exportPolicy	使用するNFSエクスポートポリシーのNASオプション。デフォルトは `default` です	default
securityStyle	<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は `mixed` および `unix` セキュリティ スタイルをサポートします。デフォルトは `unix` です。</p>	unix
fileSystemType	SANオプションでファイルシステムのタイプを選択します。デフォルトは `ext4` です	xfs
tieringPolicy	使用する階層化ポリシー。デフォルトは `none` です。	none
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

`ontap-nas` および `ontap-san` ドライバーは、Docker ボリュームごとに ONTAP FlexVol を作成します。ONTAP は、クラスターノードあたり最大 1000 個の FlexVolsをサポートし、クラスター全体では最大 12,000 個の

FlexVolボリュームをサポートします。Docker ボリュームの要件がこの制限内に収まる場合、`ontap-nas`ドライバは、Docker ボリューム単位の Snapshot やクローン作成など、FlexVolsが提供する追加機能により、NAS ソリューションとして推奨されます。

FlexVolの制限で収容できる数よりも多くのDockerボリュームが必要な場合は、`ontap-nas-economy`または`ontap-san-economy`ドライバを選択してください。

`ontap-nas-economy`ドライバは、自動管理されるFlexVolボリュームのプール内で、DockerボリュームをONTAP Qtreeとして作成します。Qtreeは、一部の機能を犠牲にして、クラスターあたり最大100,000、クラスターあたり最大2,400,000という、はるかに優れたスケーリングを提供します。`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。



`ontap-nas-economy`ドライバは、Docker Swarmが複数のノードにわたるボリューム作成を調整しないため、現在Docker Swarmではサポートされていません。

`ontap-san-economy`ドライバは、自動管理されるFlexVolボリュームの共有プール内で、DockerボリュームをONTAP LUNとして作成します。これにより、各FlexVolは1つのLUNに制限されず、SANワークロードに対してより優れたスケーラビリティを提供します。ストレージレイによっては、ONTAPはクラスターあたり最大16384個のLUNをサポートします。ボリュームは内部的にLUNであるため、このドライバはDockerボリューム単位のスナップショットとクローニングをサポートします。

`ontap-nas-flexgroup`ドライバーを選択して、1つのボリュームへの並列性を高め、数十億のファイルを持つペタバイト規模まで拡張できます。FlexGroupsの理想的なユースケースには、AI/ML/DL、ビッグデータと分析、ソフトウェアビルド、ストリーミング、ファイルリポジトリなどがあります。Tridentは、FlexGroupボリュームをプロビジョニングする際に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サポートには、以下の考慮事項もあります：

- ONTAP バージョン 9.2 以降が必要です。
- 本稿執筆時点では、FlexGroupsはNFS v3のみをサポートします。
- SVM に対して 64 ビット NFSv3 識別子を有効にすることを推奨します。
- 推奨される最小のFlexGroupメンバー/ボリュームサイズは100GiBです。
- クローニングはFlexGroupボリュームではサポートされていません。

FlexGroupsおよびFlexGroupsに適したワークロードの詳細については、["NetApp FlexGroupボリュームのベストプラクティスと実装ガイド"](#)を参照してください。

同じ環境で高度な機能と大規模なスケールを実現するには、Dockerボリュームプラグインの複数のインスタ

ンスを実行することができます。1つは `ontap-nas` を使用し、もう1つは `ontap-nas-economy` を使用します。

TridentのカスタムONTAP ロール

最小限の権限を持つONTAPクラスタロールを作成することで、Tridentで操作を実行するためにONTAP管理者ロールを使用する必要がなくなります。Tridentバックエンド構成にユーザー名を含めると、Tridentは作成したONTAPクラスタロールを使用して操作を実行します。

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

ONTAPコマンドラインの使用

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ユーザーのユーザー名を作成します：

```
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. ロールをユーザーにマップします：

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

System Managerを使用

ONTAP System Managerで次の手順を実行します。

1. カスタムロールを作成する：
 - a. クラスタレベルでカスタムロールを作成するには、*** Cluster > Settings ***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 > ストレージVM > required SVM> 設定 > ユーザーとロール***を選択します。
 - b. ユーザーとロール*の横にある矢印アイコン (→*) を選択します。
 - c. **Roles***の下に**+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. Tridentユーザーに役割をマッピングする：**+*ユーザーとロール***ページで次の手順を実行します：
 - a. ユーザー*の下にある追加アイコン+*を選択します。
 - b. 必要なユーザー名を選択し、***Role***のドロップダウンメニューで役割を選択します。

c. *保存*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"](#)
- ["ロールとユーザーを操作する"](#)

ONTAP設定ファイルの例

▼ `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"
  }
}
```

▼ `ontap-nas-economy` 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"
}
```

▼ `ontap-san` 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"
}
```

▼ `ontap-san-economy` 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"
}
```

▼ `ontap-san` 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
}

```

▼ <code>ontap-san</code> ドライバの SCSI over FC 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 ソフトウェア構成

グローバル設定値に加えて、Element ソフトウェア（NetApp HCI/SolidFire）を使用する場合、これらのオプションが利用可能です。

オプション	概要	例
Endpoint	https://<login> : <password>@<mvip>/json- rpc/<element-version>	https://admin:admin@192.168.160. 3/json-rpc/8.0
SVIP	iSCSI IPアドレスとポート	10.0.0.7:3260
TenantName	使用する SolidFire テナント（見つからない場合は作成）	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にインターフェイスを指定する	default
Types	QoS仕様	下記の例をご覧ください

オプション	概要	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、既存のボリュームでアップグレードを実行する場合は、ボリューム名方式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
      }
    }
  ]
}
```

```
}  
}  
]  
}
```

既知の問題と制限事項

Docker で Trident を使用する際の既知の問題と制限に関する情報を確認してください。

Trident Docker Volume Plugin を古いバージョンから **20.10** 以降にアップグレードすると、「そのようなファイルまたはディレクトリはありません」というエラーが発生し、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** である必要があります。



これは Docker クライアントの制限です。クライアントは、1 文字の名前を Windows パスとして解釈します ["バグ25773を参照"](#)。

Docker Swarm には、**Trident** があらゆるストレージとドライバの組み合わせでサポートすることを妨げる特定の動作があります。

- Docker Swarm は現在、一意のボリューム識別子としてボリューム ID ではなくボリューム名を使用しています。
- ボリュームリクエストは、Swarm クラスター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 クラスター内の各ノードで独立して実行する必要があります。ONTAP の仕組みと `ontap-nas` および `ontap-san` ドライバの機能により、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。

残りのドライバーは、競合状態などの問題の影響を受けやすく、その結果、明確な「勝者」なしに単一のリクエストに対して大量のボリュームが作成されることがあります。たとえば、Elementには、ボリュームに同じ名前を付けてIDを異ならせることができる機能があります。

NetAppはDockerチームにフィードバックを提供しましたが、今後の対応については何も示されていません。

FlexGroupがプロビジョニングされている場合、**2番目のFlexGroup**がプロビジョニングされている**FlexGroup**と**1つ以上のアグリゲート**を共有している場合、**ONTAP**は**2番目の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。