



# Docker 用のTrident

## Trident

NetApp  
January 15, 2026

# 目次

Docker 用のTrident	1
展開の前提条件	1
要件を確認する	1
NVMeツール	3
FCツール	4
Tridentを展開する	6
Docker マネージド プラグイン方式 (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 のアップグレードまたはアンインストール	10
Upgrade	11
アンインストール	12
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	13
ボリュームをクローニングする	13
外部で作成されたボリュームにアクセスする	15
ドライバー固有のボリュームオプション	15
ログを収集する	21
トラブルシューティングのためにログを収集する	21
一般的なトラブルシューティングのヒント	22
複数のTridentインスタンスを管理する	22
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)	22
従来の手順 (バージョン1.12以前)	23
ストレージ構成オプション	23
グローバル設定オプション	23
ONTAP構成	24
要素ソフトウェア構成	32
既知の問題と制限事項	34
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。	34
ボリューム名は 2 文字以上である必要があります。	35
Docker Swarm には、Trident があらゆるストレージとドライバーの組み合わせをサポートできない原因となる特定の動作がありま す。	35
FlexGroupがプロビジョニングされている場合、2 番目のFlexGroup にプロビジョニングされているFlexGroupと共に 1 つ以上のアグリゲートが存在すると、ONTAP は2 番目のFlexGroup をプロビジョニングしません。	35

# Docker 用のTrident

## 展開の前提条件

Trident を展開する前に、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構成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する["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、構成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
```



確保する `etc/multipath.conf`、含む `find_multipaths no` 下 `defaults`。

5. 確実に `iscsid` そして `multipathd` 実行中:

```
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
```



確保する `/etc/multipath.conf` `含む` `find_multipaths no` `下` `defaults`。

### 5. 確実に `open-iscsi` そして `multipath-tools` 有効になっていて実行中である:

```
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、`discard`インライン スペース再利用を実行するには、StorageClass の mountOption を使用します。参照 "["Red Hat 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. マルチパスを有効にする:

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



確保する `etc/multipath.conf`、含む `'find_multipaths no'` 下 `'defaults'`。

3. 確実に `'multipathd'` 実行中:

```
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
```



確保する `etc/multipath.conf`、含む `'find_multipaths no'` 下 `'defaults'`。

3. 確実に `'multipath-tools'` 有効になっていて実行中:

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

# Tridentを展開する

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。

Tridentの複数の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

参照["展開の前提条件"](#)。前提条件が満たされていることを確認したら、 Trident をデプロイ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



従来のデーモン方式で Docker 1.13/17.03 より前のTrident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、必ず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` ただし、 `config` ファイル名にオプションを指定します。設定ファイルは、 `/etc/netappdvp` ホストシステム上のディレクトリ。
- log-level: ログレベルを指定する(debug、 info、 warn、 error、 fatal)。デフォルトは info。

◦ 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。交換する `<version>` 使用しているプラグインのバージョン (xxx.xx.x) に合わせてください。

```
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、["配布の指示に従ってください"](#)。

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 Volume Plugin をインストールして構成します。

- 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/v25.06.0/trident-  
installer-25.06.0.tar.gz  
tar zxf trident-installer-25.06.0.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. 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
```

- バイナリを配置して設定ファイルを作成したら、目的の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。

- ボリュームを作成します。

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

- 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
```

- 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、次の場所にあります。  
contrib/trident.service.example Git リポジトリ内。 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、ドライバーネーム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。

```
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、`systemctl daemon-reload`変更を認識させるためのコマンドです。

## 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、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中に、`docker volume` プラグインに向けたコマンドは成功せず、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

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

手順

- 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
```

- 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
```

- 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。直接アップグレードする必要があります `netapp/ndvp-plugin` イメージを `netapp/trident-plugin` 画像。

- プラグインを有効にします:

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

- 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 より前) 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"](#)。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、追加の構成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`

## アンインストール

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

### 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

標準のボリューム作成、クローン作成、削除を簡単に行うことができます。`docker volume`必要に応じてTridentドライバー名を指定したコマンド。

### 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します。

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



何も指定しない場合は"options"、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
```



使用する際は `solidfire-san` ドライバーの場合、上記の例ではボリュームを削除して消去します。

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

## ボリュームをクローニングする

使用する際は `ontap-nas`、`ontap-san`、`solidfire-san`、そして `gcp-cvs storage drivers` 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

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を使用するコンテナからアクセスできるのは、そのブロックデバイスにパーティションがなく、そのファイルシステムがTridentでサポートされている場合のみです（例：ext4 -フォーマット済み `/dev/sdc1` Trident経由ではアクセスできなくなります）。

## ドライバー固有のボリュームオプション

各ストレージ ドライバー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。オプションと値を指定するには、`-o`

CLI 操作中のオペレータ。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	説明
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 <code>none</code> (シンプロビジョニング) および <code>volume</code> (シックプロビジョニング)。
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` つまり、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、すべてのONTAPシステムに「default」というポリシーが存在し、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、および2つの週次スナップショットが作成され、保持されます。スナップショットに保存されたデータは、`.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、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。デフォルトでは値がありません。つまり、 <code>snapshotPolicy</code> を選択した場合はONTAP は <code>snapshotReserve</code> (通常 5%) を選択し、 <code>snapshotPolicy</code> が <code>none</code> の場合は 0% を選択します。すべてのONTAPバックエンドの構成ファイルでデフォルトの <code>snapshotReserve</code> 値を設定でき、 <code>ontap-nas-economy</code> 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	ボリュームのクローンを作成すると、ONTAP はクローンをその親から直ちに分割します。デフォルトは <code>false</code> 。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。

オプション	説明
encryption	<p>新しいボリュームでNetAppボリューム暗号化（NVE）を有効にします。デフォルトは <code>false</code>。このオプションを使用するには、NVEのライセンスを取得し、クラスターで有効に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、以下を参照してください。<a href="#">"Trident がNVE および NAE と連携する仕組み"</a>。</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否かが決まります。

次の追加オプションは NFS のみに有効です。

オプション	説明
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、権限は次のように設定されます。`---rwxr-xr-x`、または数値表記0755、および `root` 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	これを設定 `true` 作るだろう `.snapshot` ボリュームにアクセスするクライアントに表示されるディレクトリ。デフォルト値は `false` つまり、`.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、例えば公式のMySQLイメージは、`.snapshot` ディレクトリが表示されます。
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 <code>default</code> 。
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 <code>unix</code> 。有効な値は <code>unix</code> そして <code>mixed</code> 。

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	説明
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 <code>ext4</code> 。有効な値は <code>ext3</code> 、 <code>ext4</code> 、そして <code>xfs</code> 。

オプション	説明
spaceAllocation	これを設定 `false` LUN のスペース割り当て機能をオフにします。デフォルト値は `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 が書き込みを受け付けことができなくなったときに、ONTAP はホストに通知します。また、このオプションで、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` Trident は 0% のスナップショット予約を使用します。

- スナップショット ポリシーとスナップショット 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 ポリシーを使用して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 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 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` 命名法。

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	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"}。
blocksize	512 または 4096 を使用します。デフォルトは 512 または構成エントリ 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

- 推奨される管理プラグイン方式（つまり、`docker plugin`コマンド）については、次のように表示します。

```
docker plugin ls
```

ID	NAME	DESCRIPTION
ENABLED		
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。

- 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。デバッグ ログを有効にするには、以下を指定します。`-debug` プラグインを実行するとき。

## 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`インストールされ実行されています。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`実行中です。`rpcbind`サービスの状態を確認するには、`systemctl status rpcbind`またはそれと同等のもの。

## 複数のTridentインスタンスを管理する

複数のストレージ構成を同時に利用したい場合は、Tridentの複数のインスタンスが必要になります。複数のインスタンスを作成するための鍵は、`--alias`コンテナ化されたプラグインのオプション、または`--volume-driver`ホスト上でTridentをインスタンス化するときのオプション。

### 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
```

- 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. 各ドライバーアイナスの 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構成で利用可能な構成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

これらの構成オプションは、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	説明	例
version	設定ファイルのバージョン番号	1
storageDriverName	ストレージドライバーの名前	ontap-nas、ontap-san、ontap-nas-economy、ontap-nas-flexgroup、solidfire-san

オプション	説明	例
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： <code>netappdvp_</code> 。	<code>staging_</code>
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: <code>""</code> (強制されません)	<code>10G</code>

使用しないでください `storagePrefix`(デフォルトを含む) Element バックエンド用。デフォルトでは、`solidfire-san` ドライバーはこの設定を無視し、プレフィックスを使用しません。NetApp、名前の変更が行われた可能性がある場合には、Docker ボリューム マッピングに特定の `tenantID` を使用するか、Docker のバージョン、ドライバー情報、および Docker からの生の名前が設定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。その `size` このオプションはすべてのコントローラー 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	説明	例
<code>size</code>	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： <code>1G</code>	<code>10G</code>

## ONTAP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を使用できます。

オプション	説明	例
<code>managementLIF</code>	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。	<code>10.0.0.1</code>

オプション	説明	例
dataLIF	<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 は以下を指定することを推奨します <code>dataLIF</code>。指定されない場合、Trident は SVM から <code>dataLIF</code> 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) を指定できるため、複数のデータ LIF 間で負荷分散を行うラウンドロビン DNS を作成できます。</li> <li>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選択的 LUN マップ" 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、`<code>dataLIF</code>` 明示的に定義されています。</li> </ul>	10.0.0.2
svm	使用するストレージ仮想マシン (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ー名	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)。のために `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%

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。のみ利用可能 `ontap-nas` そして `ontap-nas-economy` ドライバー。"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
igroupName	Tridentはノードごとに作成および管理します <code>igroups</code> 、そして <code>netappdvp</code> 。 この値は変更または省略できません。 のみ利用可能 `ontap-san` ドライバー。	netappdvp
limitVolumeSize	要求可能な最大ボリューム サイズ。	300g
qtreesPerFlexvol	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。  *については `ontap-nas-economy` ドライバーでは、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	*対応機種 `ontap-san` ドライバーのみ。*選択するには使用 `iscsi` `iSCSI` の場合、`nvme` NVMe/TCP の場合または `fcp` SCSI over Fibre Channel (FC) 用。	`iscsi` 空白の場合
limitVolumePoolSize	対応機種 `ontap-san-economy` そして `ontap-san-economy` ドライバーのみ。ONTAP ontap-nas-economy および ontap-SAN-economy ドライバーの FlexVol 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	説明	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 (厚い)	none
snapshotPolicy	使用するスナップショットポリシー、デフォルトは none	none
snapshotReserve	スナップショットの予約率。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。デフォルトでは false	false
encryption	<p>新しいボリュームでNetAppボリューム暗号化 (NVE) を有効にします。デフォルトは false。このオプションを使用するには、NVE のライセンスを取得し、クラスターで有効に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、以下を参照してください。<a href="#">"Trident がNVE および NAE と連携する仕組み"</a>。</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	アクセスのためのNASオプション `snapshot` ディレクトリ。	NFSv4の場合は「true」、NFSv3 の場合は「false」
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	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サポート mixed、そして `unix` セキュリティスタイル。デフォルトは `unix`。</p>	unix
fileSystemType	SANオプションでファイルシステムの種類を選択します。デフォルトは ext4	xfs
tieringPolicy	使用する階層化ポリシー、デフォルトは none。	none

## スケーリングオプション

その `ontap-nas` そして `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はクラスタノードあたり最大1000個のFlexVolをサポートし、クラスタの最大FlexVol数は12,000個です。Dockerボリュームの要件がこの制限内に収まる場合は、`ontap-nas` FlexVol が提供する Docker ボリュームのきめ細かなスナップショットやクローン作成などの追加機能により、このドライバーは推奨される NAS ソリューションです。

FlexVolの制限を超える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。

その `ontap-nas-economy` ドライバーは、自動的に管理されるFlexVolボリュームのプール内にONTAP Qtreeとして Docker ボリュームを作成します。Qtree は、一部の機能を犠牲にして、クラスター ノードあたり最大100,000、クラスターあたり最大 2,400,000 という、はるかに大きなスケーリングを提供します。その `ontap-nas-economy` ドライバーは、Docker ボリュームの粒度スナップショットまたはクローン作成をサポートしていません。



その `ontap-nas-economy` Docker Swarm は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、このドライバーは現在 Docker Swarm ではサポートされていません。

その `ontap-san-economy` ドライバーは、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。これにより、各FlexVol は1つの LUN のみに制限されず、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 はクラスタごとに最大 16384 個の LUN をサポートします。ボリュームは下にある LUN であるため、このドライバーは Docker ボリューム単位のスナップショットとクローン作成をサポートします。

選択してください `ontap-nas-flexgroup` 数十億のファイルでペタバイト単位にまで拡張可能な単一ボリュームへの並列処理を増やすドライバーです。FlexGroups の理想的な使用例には、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ときに、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のFlexGroupサポートには、次の考慮事項もあります。

- ONTAPバージョン 9.2 以上が必要です。
- この記事の執筆時点では、FlexGroups は NFS v3 のみをサポートしています。
- SVM に対して 64 ビット NFSv3 識別子を有効にすることをお勧めします。
- 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ン作成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、["NetApp FlexGroupボリュームのベストプラクティスと実装ガイド"](#)。

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。`ontap-nas` そしてもう一つは `ontap-nas-economy`。

## TridentのカスタムONTAPロール

最小限の権限を持つONTAPクラスタ ロールを作成すると、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。

参照 "Tridentカスタムロールジェネレーター" Tridentカスタム ロールの作成の詳細については、こちらをご覧ください。

### ONTAP CLI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. クラスター レベルでカスタム ロールを作成するには、クラスター > 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 > ストレージVM > を選択します。 **required SVM** > 設定 > ユーザーとロール。

- 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. 役割\*の下の+追加\*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. 役割を Tridentユーザーにマップします: + ユーザーと役割 ページで次の手順を実行します。

- a. ユーザー\*の下の追加アイコン+\*を選択します。
- b. 必要なユーザー名を選択し、\*役割\*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"
- "役割とユーザーを操作する"

## 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> ドライバーの 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> ドライバーの 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": "agg1"  
}
```

<code>ontap-san</code> ドライバーの 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": "agg1",  
    "igroupName": "netappdvp"  
}
```

<code>ontap-san-economy</code> ドライバーの 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": "agg1",  
    "igroupName": "netappdvp"  
}
```

<code>ontap-san</code> ドライバーの 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": "fcp",  
    "svm": "trident_svm",  
    "username": "vsadmin",  
    "password": "password",  
    "useREST": true  
}
```

## 要素ソフトウェア構成

Element ソフトウェア (NetApp HCI/ SolidFire) を使用する場合、グローバル設定値に加えて、次のオプションが使用できます。

オプション	説明	例
Endpoint	https://<ログイン>:<パスワード>@<mvip>/json-rpc/<要素バージョン>	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

オプション	説明	例
TenantName	使用する SolidFireF テナント (見つからない場合は作成)	docker
InitiatorIFace	iSCSI トライフィックをデフォルト以外のインターフェースに制限する場合にインターフェースを指定する	default
Types	QoS仕様	下記の例をご覧ください
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する場合は、ボリューム名方式でマップされた古いボリュームにアクセスするには、この値を設定する必要があります。	netappdvp-

その `solidfire-san` ドライバーは 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見つけます。

**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. 追加の情報を提供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ーの組み合わせをサポートできない原因とな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` そして `ontap-san` ドライバー機能は、これらの制限内で動作できる唯一のものです。

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**にプロビジョニングされている**FlexGroup**と共に 1 つ以上のアグリゲートが存在すると、**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隨時、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 5225.227-19（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。