



# tridentctlを使用してインストールする

## Trident

NetApp  
July 01, 2026

# 目次

tridentctlを使用してインストールする	1
tridentctlを使用してインストールする	1
Trident10に関する重要な情報	1
Tridentを使用してインストール tridentctl	1
ステップ1：Tridentインストーラーパッケージをダウンロードする	2
ステップ2：Tridentをインストールする	2
インストールの確認	3
サンプル構成	4
tridentctl インストールのカスタマイズ	5
インストーラについて	5

# tridentctlを使用してインストールする

## tridentctlを使用してインストールする

Tridentは`tridentctl`を使用してインストールできます。このプロセスは、Tridentに必要なコンテナイメージがプライベートレジストリに保存されているかどうかに関係なく適用されます。`tridentctl`の導入をカスタマイズするには、"[tridentctl 導入環境のカスタマイズ](#)"を参照してください。

### Trident10に関する重要な情報

Trident に関する以下の重要な情報を必ずお読みください。

#### <strong>Tridentに関する重要な情報</strong>

- Kubernetes 1.27 が Trident でサポートされるようになりました。Kubernetes をアップグレードする前に Trident をアップグレードしてください。
- Trident は SAN 環境でのマルチパス構成の使用を厳格に強制し、multipath.conf ファイル内の推奨値は`find\_multipaths: no`です。

非マルチパス構成の使用、または multipath.conf ファイルでの`find\_multipaths: yes`または`find\_multipaths: smart`値の使用は、マウントの失敗を引き起こします。Trident は、21.07 リリース以降`find\_multipaths: no`の使用を推奨しています。

### Tridentを使用してインストール tridentctl

"[インストールの概要](#)"を確認して、インストールの前提条件を満たしていること、および環境に適したインストールオプションが選択されていることを確認します。

開始する前に

インストールを始める前に、Linuxホストにログインし、正常に動作している"[サポートされている Kubernetes クラスタ](#)"を管理していること、および必要な権限を持っていることを確認します。



OpenShift では、以下のすべての例で oc を kubectl の代わりに使用し、最初に oc login -u system:admin または oc login -u kube-admin を実行して **system:admin** としてログインしてください。

1. Kubernetes のバージョンを確認します：

```
kubectl version
```

2. クラスタ管理者権限を確認します：

```
kubectl auth can-i '*' '*' --all-namespaces
```

3. Docker Hub からイメージを使用するポッドを起動し、ポッド ネットワーク経由でストレージ システムにアクセスできることを確認します：

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

## ステップ1：Tridentインストーラーパッケージをダウンロードする

Tridentインストーラーパッケージは、Tridentポッドを作成し、その状態を維持するために使用されるCRDオブジェクトを構成し、CSIサイドカーを初期化して、ボリュームのプロビジョニングやクラスターホストへの接続などのアクションを実行します。最新バージョンのTridentインストーラーを["GitHub の Assets セクション"](#)からダウンロードして解凍します。例の\_`<trident-installer-XX.XX.X.tar.gz>`を選択したTridentバージョンで更新します。

```
wget https://github.com/NetApp/trident/releases/download/v26.02.0/trident-
installer-26.02.0.tar.gz
tar -xf trident-installer-26.02.0.tar.gz
cd trident-installer
```

## ステップ2：Tridentをインストールする

目的のネームスペースにTridentをインストールするには、``tridentctl install``コマンドを実行します。追加の引数を追加して、イメージレジストリの場所を指定できます。

## 標準モード

```
./tridentctl install -n trident
```

## 1つのレジストリ内の画像

```
./tridentctl install -n trident --image-registry <your-registry>  
--autosupport-image <your-registry>/trident-autosupport:26.02 --trident  
-image <your-registry>/trident:26.02.0
```

## 異なるレジストリ内のイメージ

```
./tridentctl install -n trident --image-registry <your-registry>  
--autosupport-image <your-registry>/trident-autosupport:26.02 --trident  
-image <your-registry>/trident:26.02.0
```

インストールのステータスは次のようになります。

```
....  
INFO Starting Trident installation.                namespace=trident  
INFO Created service account.  
INFO Created cluster role.  
INFO Created cluster role binding.  
INFO Added finalizers to custom resource definitions.  
INFO Created Trident service.  
INFO Created Trident secret.  
INFO Created Trident deployment.  
INFO Created Trident daemonset.  
INFO Waiting for Trident pod to start.  
INFO Trident pod started.                          namespace=trident  
pod=trident-controller-679648bd45-cv2mx  
INFO Waiting for Trident REST interface.  
INFO Trident REST interface is up.                 version=26.10.0  
INFO Trident installation succeeded.  
....
```

## インストールの確認

ポッド作成ステータスを使用してインストールを確認するか、`tridentctl`を使用できます。

## ポッド作成ステータスの使用

作成されたポッドのステータスを確認することで、Tridentのインストールが完了したかどうかを確認できます：

```
kubectl get pods -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-controller-679648bd45-cv2mx	6/6	Running	0	5m29s
trident-node-linux-vgc8n	2/2	Running	0	5m29s



インストーラが正常に完了しない場合、または `trident-controller-<generated id>` (`trident-csi-<generated id>` (23.01より前のバージョン) が\*Running\*ステータスでない場合、プラットフォームはインストールされていません。`-d`を使用して"デバッグモードをオンにする"問題のトラブルシューティングを行います。

## 使用 tridentctl

`tridentctl`を使用して、インストールされているTridentのバージョンを確認できます。

```
./tridentctl -n trident version
```

```
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 26.02.0       | 26.02.0       |
+-----+-----+
```

## サンプル構成

以下の例は、`tridentctl`を使用してTridentをインストールするためのサンプル構成を示しています。

### Windows ノード

Windows ノードで Trident を実行できるようにするには：

```
tridentctl install --windows -n trident
```

## 強制的に切り離す

詳細については、"[Tridentを使用したステートフルアプリケーションのフェイルオーバーの自動化](#)"を参照してください。

```
tridentctl install --enable-force-detach=true -n trident
```

## 同時実行Tridentコントローラ操作を有効にする

スループットを向上させるために同時実行Tridentコントローラ処理を有効にするには、この例に示すように、インストール中に`--enable-concurrency`オプションを追加します。



**Tech Preview**：この機能は試験的なもので、現在、ONTAP-NAS (NFSのみ) およびONTAP-SAN (統合ONTAP 9のNVMe) ドライバを使用した限定的な並列ワークフローをサポートしています。これは、既存のONTAP-SANドライバ (統合ONTAP 9のiSCSIおよびFCPストレージ プロトコル) のTech Previewに加えてのもので、

```
tridentctl install --enable-concurrency -n trident
```

# tridentctl インストールのカスタマイズ

Trident インストーラを使用してインストールをカスタマイズできます。

## インストーラについて

Tridentインストーラを使用すると、属性をカスタマイズできます。たとえば、Tridentイメージをプライベートリポジトリにコピーした場合は、`--trident-image`を使用してイメージ名を指定できます。Tridentイメージと必要なCSIサイドカーイメージをプライベートリポジトリにコピーした場合は、`--image-registry`スイッチを使用してそのリポジトリの場所を指定することをお勧めします。このスイッチは`<registry FQDN>[:port]`の形式をとります。



プライベートリポジトリにTridentをインストールする際、`--image-registry`スイッチを使用してリポジトリの場所を指定する場合は、リポジトリパスに`/netapp/`を使用しないでください。例：`./tridentctl install --image-registry <image-registry> -n <namespace>`

Kubernetesのディストリビューションを使用している場合、`kubelet`が通常の`/var/lib/kubelet`とは異なるパスにデータを保存する場合は、`--kubelet-dir`を使用して代替パスを指定できます。

インストーラの引数で許可されている範囲を超えてインストールをカスタマイズする必要がある場合は、展開ファイルをカスタマイズすることもできます。`--generate-custom-yaml`パラメータを使用すると、インストーラの`setup`ディレクトリに以下のYAMLファイルが作成されます：

- trident-clusterrolebinding.yaml

- trident-deployment.yaml
- trident-crds.yaml
- trident-clusterrole.yaml
- trident-daemonset.yaml
- trident-service.yaml
- trident-namespace.yaml
- trident-serviceaccount.yaml
- trident-resourcequota.yaml \*

これらのファイルを生成したら、必要に応じて変更し、`--use-custom-yaml`を使用してカスタム展開をインストールできます。

```
./tridentctl install -n trident --use-custom-yaml
```

## 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。