



Tridentのアップグレード

Trident

NetApp
July 01, 2026

目次

Tridentのアップグレード	1
Tridentのアップグレード	1
アップグレード前の考慮事項	1
ステップ1：バージョンを選択する	1
ステップ2：元のインストール方法を決定する	2
ステップ3：アップグレード方法を選択する	2
オペレータを使用したアップグレード	2
オペレーターのアップグレードワークフローを理解する	2
Trident Operator または Helm を使用して Trident インストールをアップグレードする	3
tridentctl によるアップグレード	7

Tridentのアップグレード

Tridentのアップグレード

24.02 リリース以降、Trident は 4 か月ごとにリリースされ、毎年 3 つのメジャーリリースが提供されます。新しいリリースはそれぞれ以前のリリースに基づいて構築され、新しい機能、パフォーマンスの強化、バグ修正、改善が提供されます。Trident の新機能をご利用いただくために、少なくとも年に 1 回はアップグレードすることをお勧めします。

アップグレード前の考慮事項

Trident の最新リリースにアップグレードする場合は、次の点を考慮してください：

- 特定の Kubernetes クラスター内のすべての名前空間にわたって、Trident インスタンスが 1 つだけインストールされている必要があります。
- Trident 23.07以降では、v1ボリュームスナップショットが必須となり、アルファスナップショットやベータスナップショットはサポートされなくなりました。
- アップグレード時には、Tridentで使用されている `parameter.fsType` に `StorageClasses` を必ず指定することが重要です。既存のボリュームに影響を与えることなく、`StorageClasses` を削除して再作成できます。
 - これは、SAN ボリュームに "[セキュリティコンテキスト](#)" を適用するための要件です。
 - [sample inputディレクトリ](#)には、<https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-basic.yaml.template> [storage-class-basic.yaml.template] や storage-class-bronze-default.yaml などの例が含まれています。
 - 詳細については、"[既知の問題](#)" を参照してください。

ステップ1：バージョンを選択する

Tridentのバージョンは日付ベースの `YY.MM` 命名規則に従います。「YY」は年の下2桁、「MM」は月を表します。ドットリリースは `YY.MM.X` 命名規則に従い、「X」はパッチレベルを表します。アップグレード元のバージョンに基づいて、アップグレード先のバージョンを選択してください。

- インストールされているバージョンから4リリース以内の任意のターゲットリリースへの直接アップグレードを実行できます。たとえば、24.06（または任意の24.06ドットリリース）から25.06に直接アップグレードできます。
- 4つのリリース期間外のリリースからアップグレードする場合は、複数ステップのアップグレードを実行してください。"[以前のバージョン](#)" からアップグレードする場合は、そのバージョンのアップグレード手順を使用して、4リリース期間に収まる最新のリリースにアップグレードします。例えば、23.07 を実行していて、25.06 にアップグレードしたい場合（：）
 - a. 最初に 23.07 から 24.06 にアップグレードします。
 - b. 次に、24.06から25.06にアップグレードします。



OpenShift Container Platform で Trident オペレーターを使用してアップグレードする場合は、Trident 21.01.1 以降にアップグレードする必要があります。21.01.0 でリリースされた Trident オペレーターには既知の問題が含まれていますが、21.01.1 で修正されています。詳細については、"[GitHubの問題の詳細](#)"を参照してください。

ステップ2：元のインストール方法を決定する

最初に Trident をインストールする際に使用したバージョンを確認するには：

1. `kubectl get pods -n trident` を使用してポッドを検査します。
 - オペレーターポッドがない場合、Trident は `tridentctl` を使用してインストールされました。
 - オペレーターポッドがある場合、Trident は Trident オペレーターを使用して手動または Helm を使用してインストールされました。
2. オペレーターポッドがある場合は、`kubectl describe torc` を使用して、TridentがHelmを使用してインストールされたかどうかを確認します。
 - Helmラベルがある場合、TridentはHelm を使用してインストールされました。
 - Helmラベルがない場合、TridentはTridentオペレーターを使用して手動でインストールされました。

ステップ3：アップグレード方法を選択する

一般的には、最初のインストールに使用したのと同じ方法でアップグレードする必要がありますが、"[インストール方法を切り替える](#)"。Trident をアップグレードするには 2 つのオプションがあります。

- "[Tridentオペレータを使用してアップグレードする](#)"



オペレータでアップグレードする前に"[オペレーターのアップグレードワークフローを理解する](#)"を確認することをお勧めします。

*

オペレータを使用したアップグレード

オペレーターのアップグレードワークフローを理解する

Tridentオペレーターを使用してTridentをアップグレードする前に、アップグレード中に発生するバックグラウンド プロセスを理解する必要があります。これには、Tridentコントローラー、コントローラーポッドとノードポッド、およびローリングアップデートを可能にするノードDaemonSetへの変更が含まれます。

Tridentオペレータのアップグレード処理

Trident のインストールとアップグレードの多くの"[Tridentオペレーターを使用する利点](#)"の 1 つは、既存のマウントされたボリュームを中断することなく、Trident と Kubernetes オブジェクトを自動的に処理することです。このようにして、Trident はダウンタイムなしでアップグレードをサポートできます。または"[ローリングアップデート](#)"。特に、Trident オペレーターは Kubernetes クラスターと通信して以下を実行します：

- Trident コントローラの導入とノード DaemonSet を削除して再作成します。
- Trident Controller PodとTrident Node Podを新しいバージョンに置き換えます。
 - 1つのノードが更新されない場合でも、残りのノードの更新は妨げられません。
 - 実行中の Trident Node Pod を持つノードのみがボリュームをマウントできます。



Kubernetesクラスタ上のTridentアーキテクチャの詳細については、"[Tridentアーキテクチャ](#)"を参照してください。

オペレータのアップグレードワークフロー

Trident オペレータを使用してアップグレードを開始すると：

1. **Trident** オペレータ：
 - a. 現在インストールされている Trident のバージョン（バージョン n ）を検出します。
 - b. CRD、RBAC、Trident SVC を含むすべての Kubernetes オブジェクトを更新します。
 - c. バージョン n の Trident Controller デプロイメントを削除します。
 - d. バージョン $n+1$ の Trident Controller デプロイメントを作成します。
2. **Kubernetes** は $n+1$ 用の Trident Controller Pod を作成します。
3. **Trident** オペレータ：
 - a. n の Trident ノード DaemonSet を削除します。オペレーターはノード Pod の終了を待機しません。
 - b. $n+1$ の Trident Node Daemonset を作成します。
4. **Kubernetes** は、Trident Node Pod n を実行していないノードに Trident Node Pod を作成します。これにより、ノード上に任意のバージョンの Trident Node Pod が複数存在しないことが保証されます。

Trident Operator または Helm を使用して Trident インストールをアップグレードする

Trident オペレーターを使用して、手動または Helm を使用して Trident をアップグレードできます。Trident オペレーターのインストールから別の Trident オペレーターのインストールにアップグレードすることも、`tridentctl` のインストールから Trident オペレーターバージョンにアップグレードすることもできます。Trident オペレーターのインストールをアップグレードする前に"[アップグレード方法を選択](#)"を確認してください。

手動インストールのアップグレード

クラスタを対象とした Trident オペレータのインストールから、別のクラスタを対象とした Trident オペレータのインストールにアップグレードできます。すべての Trident バージョンでは、クラスタを対象としたオペレータを使用します。



名前空間スコープのオペレータを使用してインストールされた Trident（バージョン 20.07 から 20.10）からアップグレードするには、"[インストールされているバージョン](#)"の Trident のアップグレード手順を使用してください。

タスク概要

Trident は、オペレーターをインストールし、Kubernetes バージョンに関連付けられたオブジェクトを作成するために使用できるバンドルファイルを提供します。

- Kubernetes 1.25以降を実行しているクラスターの場合は、"[bundle_post_1_25.yaml](#)"を使用します。

開始する前に

"[サポートされている Kubernetes バージョン](#)"を実行しているKubernetesクラスタを使用していることを確認してください。

手順

1. Trident のバージョンを確認してください：

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

2. `operator.yaml`、`tridentorchestrator_cr.yaml`、`post_1_25_bundle.yaml`を、アップグレード先のバージョン（例：25.06）のレジストリとイメージパス、および正しいシークレットで更新します。
3. 現在の Trident インスタンスのインストールに使用された Trident オペレーターを削除します。たとえば、25.02 からアップグレードする場合は、次のコマンドを実行します：

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n trident
```

4. 初期インストールを `TridentOrchestrator` 属性を使用してカスタマイズした場合は、`TridentOrchestrator` オブジェクトを編集してインストールパラメータを変更できます。これには、オフラインモード用のミラー化されたTridentおよびCSIイメージレジストリの指定、デバッグログの有効化、またはイメージプルシークレットの指定などの変更が含まれる場合があります。
5. 環境に適したバンドルYAMLファイルを使用してTridentをインストールします。 `<bundle.yaml>` は `bundle_pre_1_25.yaml` または `bundle_post_1_25.yaml` Kubernetesのバージョンに基づきます。たとえば、Trident 25.06.0をインストールする場合は、次のコマンドを実行します：

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n trident
```

6. Trident `torc` を編集して、イメージ 25.06.0 を含めます。

Helmインストールのアップグレード

Trident Helm のインストールをアップグレードできます。



TridentがインストールされているKubernetesクラスタを1.24から1.25以降にアップグレードする場合、クラスタをアップグレードする前に、`values.yaml`を更新して `excludePodSecurityPolicy` を `true` に設定するか、 `--set excludePodSecurityPolicy=true` を `helm upgrade` コマンドに追加する必要があります。

Trident helm をアップグレードせずに Kubernetes クラスタを 1.24 から 1.25 にアップグレード済みの場

合、helm のアップグレードは失敗します。helm のアップグレードを実行するには、前提条件として次の手順を実行してください：

1. <https://github.com/helm/helm-mapkubeapis>からhelm-mapkubeapisプラグインをインストールします。
2. Trident がインストールされているネームスペースで、Trident リリースのドライランを実行します。クリーンアップされるリソースが一覧表示されます。

```
helm mapkubeapis --dry-run trident --namespace trident
```

3. helm を使用して完全実行を実行し、クリーンアップを実行します。

```
helm mapkubeapis trident --namespace trident
```

手順

1. "Helmを使用してTridentをインストールしました"の場合は、`helm upgrade trident netapp-trident/trident-operator --version 100.2602.0`を使用してワンステップでアップグレードできます。Helmリポジトリを追加していない場合、またはアップグレードに使用できない場合：
 - a. 最新のTridentリリースを"[GitHub の Assets セクション](#)"からダウンロードします。
 - b. `helm upgrade`コマンドを使用します。`trident-operator-26.02.0.tgz`にはアップグレード先のバージョンを指定します。

```
helm upgrade <name> trident-operator-26.02.0.tgz
```



初期インストール時にカスタムオプション（TridentおよびCSIイメージ用のプライベートミラーリポジトリの指定など）を設定した場合は、それらのオプションがアップグレードコマンドに含まれるように、`helm upgrade`コマンドに`--set`を追加してください。そうしないと、値がデフォルトにリセットされます。

2. `helm list`を実行して、チャートとアプリのバージョンが両方ともアップグレードされたことを確認します。`tridentctl logs`を実行して、デバッグメッセージを確認します。

`tridentctl`インストールからTridentオペレーターへのアップグレード

Tridentオペレーターの最新リリースには、`tridentctl`インストールからアップグレードできます。既存のバックエンドとPVCは自動的に利用できるようになります。



インストール方法を切り替える前に、"[インストール方法間の移行](#)"を確認してください。

手順

1. 最新の Trident リリースをダウンロードします。

```
# Download the release required [26.02.0]
mkdir 26.02.0
cd 26.02.0
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
```

2. マニフェストから tridentorchestrator CRD を作成します。

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. クラスタースコープのオペレーターを同じ名前空間にデプロイします。

```
kubectl create -f deploy/<bundle-name.yaml>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc 6/6     Running   0           150d
trident-node-linux-xrst8             2/2     Running   0           150d
trident-operator-5574dbbc68-nthjv    1/1     Running   0           1m30s
```

4. Trident をインストールするための TridentOrchestrator CR を作成します。

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml
```

```
#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6    Running   0           1m
trident-csi-xrst8                    2/2    Running   0           1m
trident-operator-5574dbbc68-nthjv    1/1    Running   0           5m41s
```

5. Trident が意図したバージョンにアップグレードされたことを確認します。

```
kubectl describe torc trident | grep Message -A 3

Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v26.02.0
```

tridentctl によるアップグレード

既存のTridentインストールは `tridentctl` を使用して簡単にアップグレードできます。

タスク概要

Trident をアンインストールして再インストールすると、アップグレードとして機能します。Trident をアンインストールしても、Trident デプロイメントで使用される永続ボリューム要求 (PVC) と永続ボリューム (PV) は削除されません。すでにプロビジョニングされている PV は、Trident がオフラインの間も引き続き使用可能であり、Trident はオンラインに戻った後、その間に作成されたすべての PVC のボリュームをプロビジョニングします。

開始する前に

"[アップグレード方法を選択](#)"を使用してアップグレードする前に `tridentctl` を確認してください。

手順

1. `tridentctl` でアンインストールコマンドを実行して、CRDと関連オブジェクトを除く、Tridentに関連付けられたすべてのリソースを削除します。

```
./tridentctl uninstall -n <namespace>
```

2. Trident を再インストールします。"[tridentctl を使用して Trident をインストールする](#)"を参照してください。



アップグレードプロセスを中断しないでください。インストーラが完了まで実行されることを確認してください。

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。