



# **Tridentをアップグレード**

## Trident

NetApp  
January 15, 2026

# 目次

Tridentをアップグレード	1
Tridentをアップグレード	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
オペレーターによるアップグレード	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をアップグレード

## Tridentをアップグレード

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の新機能を活用できるよう、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。
- Google CloudのCloud Volumes Serviceを["CVSサービスタイプ"](#)バックエンド構成を更新して、`standardsw` または `zoneredundantstandardsw` Trident 23.01 からアップグレードする場合のサービス レベル。更新に失敗した `serviceLevel` バックエンドでボリュームが失敗する可能性があります。参照 ["CVSサービスタイプのサンプル"](#) 詳細については。
- アップグレードする際には、`parameter.fsType` で `StorageClasses` Tridentによって使用されます。削除して再作成できます `StorageClasses` 既存のボリュームを中断することなく。
  - これは、強制するための要件です ["セキュリティコンテキスト"](#) SAN ボリューム用。
  - [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>などの例が含まれています。[storage-class-basic.yaml.template ^]とリンク:<https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml>[storage-class-bronze-default.yaml ^]。
  - 詳細については、["既知の問題"](#)。

### 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 `YY.MM` 命名規則。「YY」は年の最後の 2 桁、「MM」は月です。ドット リリースは `YY.MM.X` 規則では、「X」はパッチ レベルです。アップグレード元のバージョンに基づいて、アップグレード後のバージョンを選択します。

- 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 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、複数段階のアップグレードを実行します。アップグレード手順については、["以前のバージョン"](#) アップグレード元のバージョンを、4 つのリリースの期間に収まる最新のリリースにアップグレードします。たとえば、23.07 を実行していて、25.06 にアップグレードする場合:
  - 23.07 から 24.06 への最初のアップグレード。

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 上の問題の詳細"](#)。

## 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. 使用 `kubectl get pods -n trident` ポッドを検査します。
  - オペレーター ポッドがない場合、Trident は次のようにインストールされます。 `tridentctl`。
  - オペレーター ポッドがある場合、Trident は手動または Helm を使用して Trident オペレーター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 svc Trident` が Helm を使用してインストールされているかどうかを判断します。
  - 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ています。

## ステップ3: アップグレード方法を選択する

通常は、最初のインストール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 オペレーターを使用してアップグレードする"](#)



確認することをお勧めします ["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"](#) オペレーターにアップグレードする前に。

\*

## オペレーターによるアップグレード

### 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 プロセスを理解しておく必要があります。これには、ローリング アップデートを有効にする Trident コントローラー、コントローラー Pod とノード Pod、およびノード DaemonSet への変更が含まれます。

### Trident オペレーターのアップグレード処理

数ある中の一つ ["Trident オペレーターを使用する利点"](#) Trident をインストールおよびアップグレードすると、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 および Kubernetes オブジェクトが自動的に処理

されます。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。["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次の操作を行います。

- Tridentコントローラーのデプロイメントとノード DaemonSet を削除して再作成します。
- Tridentコントローラー ポッドとTridentノード ポッド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クラスター上のTridentアーキテクチャの詳細については、以下を参照してください。["Tridentアーキテクチャ"](#)。

## オペレーター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、次のようになります。

### 1. トライデントTrident:

- 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。
- CRD、RBAC、Trident SVCを含むすべてのKubernetesオブジェクトを更新します。
- バージョン  $n$  のTridentコントローラーのデプロイメントを削除します。
- バージョン  $n+1$  のTridentコントローラーのデプロイメントを作成します。

### 2. Kubernetes は $n+1$ 用のTridentコントローラー ポッドを作成します。

### 3. トライデントTrident:

- $n$  のTrident Node DaemonSetを削除します。オペレーターはノード ポッドの終了を待機しません。
- $n+1$  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1つしか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、手動でも Helm を使用しても Trident をアップグレードできます。Tridentオペレータのインストールから別のTridentオペレータのインストールにアップグレードしたり、tridentctl Tridentオペレータ バージョンへのインストール。レビュー["アップグレード方法を選択してください"](#)Tridentオペレータインストールをアップグレードする前に。

## 手動インストールのアップグレード

クラスター スコープのTridentオペレーターインストールから別のクラスター スコープのTridentオペレーターインストールにアップグレードできます。すべてのTridentバージョンは、クラスター スコープの演算子を使用します。



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"](#)Tridentの。

## 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使用できるバンドルファイルを提供します。

- Kubernetes 1.24を実行しているクラスターの場合は、"バンドル\_pre\_1\_25.yaml"。
- Kubernetes 1.25以降を実行しているクラスターの場合は、"バンドルポスト1\_25.yaml"。

## 開始する前に

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

## 手順

1. Tridentのバージョンを確認してください:

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

2. 更新する operator.yaml、tridentorchestrator\_cr.yaml、そして `post\_1\_25\_bundle.yaml` アップグレードするバージョン(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を入力します。
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カスタマイズした場合 `TridentOrchestrator` 属性を編集できます `TridentOrchestrator` インストールパラメータを変更するオブジェクト。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を指定したり、デバッグログを有効にしたり、イメージ プルシークレットを指定したりするため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. トライデントトルクを編集して、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は、values.yamlを更新して設定する必要があります。`excludePodSecurityPolicy`に`true`または追加`--set excludePodSecurityPolicy=true`に`helm upgrade`クラスターをアップグレードする前にコマンドを実行する必要があります。

Trident helm をアップグレードせずに Kubernetes クラスターを 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. helm-mapkubeapis プラグインをインストールする <https://github.com/helm/helm-mapkubeapis>。
2. Tridentがインストールされている名前空間で、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リストされます。

```
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. もしあなたが "Helmを使用してTridentをインストールしました"、使用することができます `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` 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は、次の手順を実行します。
  - a. 最新のTridentリリースをダウンロードするには、["GitHubの\\_Assets\\_セクション"](#)。
  - b. 使用 `helm upgrade` コマンドの場所 `trident-operator-25.06.0.tgz` アップグレードするバージョンを反映します。

```
helm upgrade <name> trident-operator-25.06.0.tgz
```



初期インストール時にカスタムオプションを設定する場合（TridentおよびCSIイメージのプライベートミラーレジストリを指定するなど）、`helm upgrade` コマンド使用`--set`これらのオプションがアップグレード コマンドに含まれていることを確認してください。含まれていない場合、値はデフォルトにリセットされます。

2. 走る `helm list` 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。走る `tridentctl logs` デバッグ メッセージを確認します。

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

Tridentオペレーターの最新リリースにアップグレードするには、`tridentctl` インストール。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、["インストール方法の変更"](#)。

## 手順

1. 最新のTridentリリースをダウンロードしてください。

```
# Download the release required [25.06.0]
mkdir 25.06.0
cd 25.06.0
wget
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
installer-25.06.0.tar.gz
tar -xf trident-installer-25.06.0.tar.gz
cd trident-installer
```

2. 作成する `tridentorchestrator` マニフェストからの CRD。

```
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と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. 作成する `TridentOrchestrator` Trident をインストールするための CR。

```

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:          v25.06.0

```

## tridentctl によるアップグレード

既存のTridentインストールを簡単にアップグレードできます。 tridentctl。

### 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 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

レビュー "アップグレード方法を選択してください" アップグレードする前に tridentctl。

### 手順

1. アンインストールコマンドを実行する `tridentctl` 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。参照["tridentctlを使用してTridentをインストールする"](#)。



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するまで実行されたことを確認します。

## 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。