



Tridentのアップグレード

Trident

NetApp
January 14, 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: バージョンを選択します	1
ステップ2: 元のインストール方法を決定します	2
ステップ3: アップグレード方法を選択します	2
オペレータにアップグレードしてください	2
オペレータのアップグレードワークフローについて理解する	2
Trident operatorまたはHelmを使用したTridentインストールのアップグレード	3
tridentctl を使用してアップグレードします	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はリリースサイクルを4カ月に短縮し、毎年3つのメジャーリリースを提供しています。新しいリリースは、以前のリリースに基づいて構築され、新機能、パフォーマンスの強化、バグの修正、および改善が提供されます。Tridentの新機能を利用するには、少なくとも年に1回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クラスタ内のすべてのネームスペースには、Tridentインスタンスを1つだけインストールする必要があります。
- Trident 23.07以降では、v1ボリュームスナップショットが必要です。alphaまたはbetaスナップショットはサポートされません。
- Cloud Volumes Service for Google Cloudを作成した場合"[CVS サービスタイプ](#)"は、Trident 23.01からのアップグレード時にまたは `zoneredundantstandardsw`` サービスレベルを使用するようにバックエンド構成を更新する必要があります。`standardsw。バックエンドで更新しない `serviceLevel` と、ボリュームで障害が発生する可能性があります。詳細については、[を参照してください "CVSサービスタイプのサンプル"](#)。
- をアップグレードする場合は、`StorageClasses`Trident` で使用するために指定することが重要です。`parameter.fsType。既存のボリュームを停止することなく、削除や再作成を実行できます `StorageClasses`。
 - これは、SANボリュームを適用するための要件です "[セキュリティコンテキスト](#)"。
 - <https://github.com/NetApp/Trident/tree/master/storage-installer/sample-input> Trident [sample input^] ディレクトリには、<https://github.com/NetApp/Trident/blob/master/storage-class-samples/storage-class-basic.yaml.template> Tridentやlink：[https://github.com/NetApp/Trident/blob/master/sample-input/storage-class-sample-input/storage-input/storage-class-samples/storage\[storage-class-basic.yaml.template](https://github.com/NetApp/Trident/blob/master/sample-input/storage-class-sample-input/storage-input/storage-class-samples/storage[storage-class-basic.yaml.template) Trident[storage-class-bronze-default.yaml
 - 詳細については、[を参照してください "既知の問題"](#)。

ステップ1：バージョンを選択します

Tridentバージョンは、日付ベースの命名規則に従い `YY.MM` ます。「YY」は年の最後の2桁、「mm」は月です。ドットリリースは規則に従い `YY.MM.X` ます。「X」はパッチレベルです。アップグレード前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の4リリースウィンドウ内にある任意のターゲットリリースに直接アップグレードできます。たとえば、24.06（または任意の24.06 DOTリリース）から25.02に直接アップグレードできます。
- 4つのリリースウィンドウ以外のリリースからアップグレードする場合は、複数の手順でアップグレードを実行します。4リリースのウィンドウに適合する最新リリースにアップグレードするには、アップグレード元のもののアップグレード手順を使用し "[以前のバージョン](#)" ます。たとえば、23.01を実行していて、25.02にアップグレードする場合は、次の手順を実行します。

- a. 23.01から24.02への最初のアップグレード。
- b. その後、24.02から25.02にアップグレードします。



OpenShift Container PlatformでTridentオペレータ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 オペレータには、21.01.1で修正された既知の問題が含まれています。詳細については、を参照して "[GitHub の問題の詳細](#)" ください。

ステップ2:元のインストール方法を決定します

Tridentを最初にインストールしたバージョンを確認するには、次の手順を実行します。

1. ポッドの検査に使用し `kubectl get pods -n trident` ます。
 - オペレータポッドがない場合は、を使用してTridentがインストールされています `tridentctl`。
 - オペレータポッドがある場合、Tridentは手動またはHelmを使用してTridentオペレータを使用してインストールされています。
2. オペレータポッドがある場合は、を使用して、`kubectl describe torc` TridentがHelmを使用してインストールされている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ています。
 - Helmラベルがない場合、TridentはTridentオペレータ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します

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、可能です。"[インストール方法を切り替えます](#)" Tridentをアップグレードする方法は2つあります。

- "[Tridentオペレータを使用してアップグレード](#)"



オペレータとアップグレードする前に、を確認することをお勧めします"[オペレータのアップグレードワークフローについて理解する](#)"。

*

オペレータにアップグレードしてください

オペレータのアップグレードワークフローについて理解する

Tridentオペレータを使用して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について理解しておく必要があります。これには、Tridentコントローラ、コントローラポッドとノードポッド、およびローリング更新を可能にするノードデーモンセットに対する変更が含まれます。

Tridentオペレータのアップグレード処理

Tridentをインストールしてアップグレードするには"[Tridentオペレータを使用するメリット](#)"、既存のマウント

ボリュームを中断することなく、TridentオブジェクトとKubernetesオブジェクトを自動的に処理する必要があります。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。["ローリング更新"](#)TridentオペレータはKubernetesクラスタと通信して次のことを行います。

- Trident Controller環境とノードデーモンセットを削除して再作成します。
- TridentコントローラポッドとTridentノードポッドを新しいバージョンに置き換えます。
 - 更新されていないノードは、残りのノードの更新を妨げません。
 - ボリュームをマウントできるのは、Trident Node Podを実行しているノードだけです。



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

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、次の処理が実行されます。

1. Trident演算子*：
 - a. 現在インストールされているTridentのバージョン (version_n_) を検出します。
 - b. CRD、RBAC、Trident SVCなど、すべてのKubernetesオブジェクトを更新
 - c. version_n_用のTrident Controller環境を削除します。
 - d. version_n+1_用のTrident Controller環境を作成します。
2. * Kubernetes *は、_n+1_用にTridentコントローラポッドを作成します。
3. Trident演算子*：
 - a. _n_のTridentノードデーモンセットを削除します。オペレータは、Node Podが終了するのを待たない。
 - b. _n+1_のTridentノードデーモンセットを作成します。
4. * Kubernetes * Trident Node Pod_n_を実行していないノードにTridentノードポッドを作成します。これにより、1つのノードに複数のTrident Node Pod (バージョンに関係なく) が存在することがなくなります。

Trident operatorまたはHelmを使用したTridentインストールのアップグレード

Tridentは、Tridentオペレータを使用して手動でアップグレードすることも、Helmを使用してアップグレードすることもできます。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することも、インストールからTridentオペレータのバージョンにアップグレードすることもできます `tridentctl`。Trident Operatorのインストールをアップグレードする前に[を参照してください"アップグレード方法を選択します"](#)。

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オペレータインストールから、クラスタを対象とした別のTridentオペレータインストールにアップグレードできます。バージョン21.01以降のTridentでは、すべてクラスタを対象とした演算子が使用されます。



名前空間を対象とした演算子（バージョン20.07～20.10）を使用してインストールされたTridentからアップグレードするには、Tridentのアップグレード手順を使用します"[インストールされているバージョン](#)"。

タスク概要

Tridentにはバンドルファイルが用意されています。このファイルを使用して、オペレータをインストールしたり、Kubernetesバージョンに対応する関連オブジェクトを作成したりできます。

- クラスタでKubernetes 1.24を実行している場合は、を使用し "[Bundle_pre_1_25.yaml](#)"ます。
- クラスタでKubernetes 1.25以降を実行している場合は、を使用します "[bundle_post_1_25.yaml](#)"。

開始する前に

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"[サポートされるKubernetesバージョン](#)"ます。

手順

1. Tridentのバージョンを確認します。

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

2. 現在のTridentインスタンスのインストールに使用したTridentオペレータを削除します。たとえば、23.07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 23.07.0/trident-installer/deploy/<bundle.yaml> -n trident
```

3. 属性を使用して初期インストールをカスタマイズした場合 `TridentOrchestrator``は、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 ``TridentOrchestrator`。これには、ミラーリングされたTridentおよびCSIイメージレジストリをオフラインモードに指定したり、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ます。
4. ご使用の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（`_``は、または ``bundle_post_1_25.yaml``使用している`<bundle.yaml>` ``bundle_pre_1_25.yaml``のバージョンに基づいています）。たとえば、Trident 25.02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 25.02.0/trident-installer/deploy/<bundle.yaml> -n trident
```

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 `helm upgrade`、クラスタをアップグレードする前に、`values.yaml`を `true` に設定するかコマンドに追加する `--set excludePodSecurityPolicy=true` ように更新する必要があります。 `excludePodSecurityPolicy`

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にアップグレードした場合、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、次の手順を前提条件として実行します。

1. から `helm-mapkubeapis` プラグインをインストールします <https://github.com/helm/helm-mapkubeapis>。
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。リソースが一覧表示され、クリーンアップされます。

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

3. クリーンアップを実行するには、`helm`を使用してフルランを実行します。

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

手順

1. を使用する "[Helmを使用してTridentをインストール](#)"と、を使用してワンステップでアップグレードできます `helm upgrade trident netapp-trident/trident-operator --version 100.2502.0`。Helmリポジトリを追加しなかった場合、またはHelmリポジトリを使用してアップグレードできない場合は、次の手順を実行します。
 - a. から最新のTridentリリースをダウンロードし "[GitHubの_Assets_sectionを参照してください](#)" ます。
 - b. コマンドを使用し `helm upgrade` ます。は、 `trident-operator-25.02.0.tgz` アップグレード先のバージョンを反映しています。

```
helm upgrade <name> trident-operator-25.02.0.tgz
```



初期インストール時にカスタムオプションを設定した場合（TridentおよびCSIイメージのプライベートなミラーレジストリの指定など）は、を使用してコマンドを追加し `helm upgrade`、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ます。含まれていない場合は、`--set` 値がデフォルトにリセットされます。

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

インストールから **Trident Operator** へのアップグレード `tridentctl`

インストールからTrident Operatorを最新リリースにアップグレードできます `tridentctl`。既存のバックエンドとPVCは自動的に使用可能になります。



インストール方法を切り替える前に、[を参照してください"インストール方法を切り替える"](#)。

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 [25.02.0]
mkdir 25.02.0
cd 25.02.0
wget
https://github.com/NetApp/trident/releases/download/v25.02.0/trident-
installer-25.02.0.tar.gz
tar -xf trident-installer-25.02.0.tar.gz
cd trident-installer
```

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

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

3. クラスタを対象としたオペレータを同じ名前スペースに導入します。

```
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CRを作成し `TridentOrchestrator` 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が目的のバージョンにアップグレードされたことを確認

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

Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v25.02.0
```

tridentctl を使用してアップグレードします

を使用して、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のアンインストールと再インストールは、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環境で使用されている永続的ボリューム要求 (PVC) と永続的ボリューム (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PVCは、Tridentがオフラインの間も使用できます。また、その間に作成されたPVCがオンラインに戻ったあとも、Tridentはボリュームをプロビジョニングします。

開始する前に

を使用してアップグレードする前に `tridentctl` を参照["アップグレード方法を選択します"](#)

手順

1. のuninstallコマンドを実行し `tridentctl` で、CRDと関連オブジェクトを除くTridentに関連付けられている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。を参照してください ["tridentctlを使用したTridentのインストール"](#)。



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ることを確認します。

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。