



Astra Trident をアップグレード

Astra Trident

NetApp
April 03, 2024

目次

Astra Trident をアップグレード	1
Astra Trident をアップグレード	1
オペレータにアップグレードしてください	2
tridentctl を使用してアップグレードします	7

Astra Trident をアップグレード

Astra Trident をアップグレード

Astra Trident は四半期ごとにリリースサイクルを実施し、毎年 4 つのメジャーリリースをリリースしています。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。ネットアップでは、Astra Tridentの新機能を活用するために、1年に1回以上アップグレードすることを推奨しています。

アップグレード前の考慮事項

最新リリースの Astra Trident にアップグレードする際は、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、Astra 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。
- Astra Trident 23.07以降では、v1ボリュームSnapshotが必要です。アルファSnapshotまたはベータSnapshotはサポートされなくなりました。
- Cloud Volumes Service for Google Cloudを **"CVS サービスタイプ"**を使用するには、バックエンド構成を更新する必要があります。 `standardsw` または `zoneredundantstandardsw` Astra Trident 23.01からアップグレードする場合のサービスレベル。の更新に失敗しました `serviceLevel` バックエンドでは、原因ボリュームで障害が発生する可能性があります。を参照してください **"CVSサービスタイプのサンプル"**を参照してください。
- アップグレードするときは、この作業を行うことが重要です `parameter.fsType` インチ `StorageClasses` Astra Tridentが使用。削除して再作成することができます `StorageClasses` 実行前のボリュームの中断はなし。
 - これは、強制的 要件 です **"セキュリティコンテキスト"** SAN ボリュームの場合。
 - `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](https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml[storage-class-bronze-default.yaml)]をクリックします。
 - 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

Astra Tridentバージョンは日付ベースです `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、の後に続きます `YY.MM.X` 条約。ここで、「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、22.10（または任意の22.10 DOTリリース）から23.10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。のアップグレード手順を使用します。 **"以前のバージョン"** から、4つのリリースウィンドウに適合する最新のリリースにアップグレードします。たとえば、21.10を実行していて、23.10にアップ

グレードする場合は、次の手順を実行します。

- a. 21.10から22.10への最初のアップグレード。
- b. その後、22.10から23.10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、を参照してください "[GitHub の問題の詳細](#)"。

ステップ2:元のインストール方法を決定します

Astra Tridentの最初のインストールに使用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. 使用 `kubectl get pods -n trident` ポッドを検査するために。
 - オペレータポッドがない場合は、を使用してAstra 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、Astra Tridentは手動またはHelm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、を使用します `kubectl describe tproc trident` をクリックし、Helmを使用してAstra Tridentがインストールされたかどうかを確認します。
 - Helmラベルがある場合は、Helmを使用してAstra Tridentがインストールされています。
 - Helmラベルがない場合は、Astra TridentをTridentオペレータを使用して手動でインストールしています。

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](#)"



レビュー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](#)" オペレータでアップグレードする前に。

*

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Astra 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、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

多数のうちの1つ ["Tridentオペレータを使用するメリット"](#) Astra Tridentのインストールとアップグレードは、既存のマウントボリュームを停止することなく、Astra TridentとKubernetesのオブジェクトを自動的に処理します。これにより、Astra 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Astra Tridentアーキテクチャの詳細については、 ["Astra 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Astra 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オペレータのインストールをアップグレード

Astra Tridentは、Tridentオペレータを使用して手動またはHelmを使用してアップグレードできます。Tridentオペレータのインストール環境から別のTridentオペレータのインストール環境へのアップグレード、または 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ます"](#) Tridentオペレータの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。すべてのAstra Tridentバージョン21.01以降では、クラスタを対象と

した演算子を使用します。



ネームスペースを対象としたオペレータ（バージョン20.07~20.10）を使用してインストールされたAstra Tridentからアップグレードするには、次のアップグレード手順を使用してください：["インストールされているバージョン"](#)実績があります。

このタスクについて

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以前を実行している場合は、を使用します ["Bundle_pre_1_25.yaml"](#)。
- 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"](#)。

作業を開始する前に

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ます ["サポートされるKubernetesバージョン"](#)。

手順

1. Astra Tridentのバージョンを確認します。

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

2. 現在のAstra 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. 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` 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、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. 環境に応じた適切なバンドルYAMLファイルを使用してAstra Tridentをインストールします（`_YAML <bundle.yaml>`）は

`bundle_pre_1_25.yaml` または `bundle_post_1_25.yaml` 使用しているKubernetesのバージョンに基づきます。たとえば、Astra Trident 23.10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 23.10.0/trident-installer/deploy/<bundle.yaml> -n trident
```

Helmインストールのアップグレード

Astra Trident Helmのインストールをアップグレードできます。



Astra 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、value.yamlを更新して設定する必要があります
excludePodSecurityPolicy 終了: true または、を追加します --set
excludePodSecurityPolicy=true に移動します helm upgrade コマンドを実行してから
クラスタをアップグレードしてください。

手順

1. あなたの場合同様 **"Helmを使用したAstra Tridentのインストール"**を使用できます。helm upgrade trident netapp-trident/trident-operator --version 23.10.0 1つの手順でアップグレードできます。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. 次のサイトからAstra Tridentの最新リリースをダウンロードしてください: **"GitHubの_Assets_sectionを参照してください"**。
 - b. を使用します helm upgrade コマンドを入力します trident-operator-23.10.0.tgz アップグレード後のバージョンが反映されます。

```
helm upgrade <name> trident-operator-23.10.0.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イメージと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、helm upgrade コマンド --set これらのオプションがupgradeコマンドに含まれるようにするため、それらのオプションの値をdefaultにリセットします。

2. を実行します helm list グラフとアプリのバージョンが両方ともアップグレードされていることを確認します。を実行します tridentctl logs デバッグメッセージを確認します。

からのアップグレード tridentctl **Trident**オペレータへのインストール

からTridentの最新リリースにアップグレードできます tridentctl インストール: 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、**"インストール方法を切り替える"**

手順

1. 最新の Astra Trident リリースをダウンロード

```
# Download the release required [23.10.0]
mkdir 23.10.0
cd 23.10.0
wget
https://github.com/NetApp/trident/releases/download/v23.10.0/trident-
installer-23.10.0.tar.gz
tar -xf trident-installer-23.10.0.tar.gz
cd trident-installer
```

2. を作成します 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. クラスタを対象としたオペレータを同じネーム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. を作成します TridentOrchestrator Astra Tridentのインストール用に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:          v23.10.0
```

tridentctl を使用してアップグレードします

を使用すると、既存のAstra Tridentインストールを簡単にアップグレードできます tridentctl。

このタスクについて

Astra 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、Astra Trident 環境で使用されている Persistent Volume Claim (PVC ; 永続的ボリューム要求) と Persistent Volume (PV ; 永続的ボリューム) は削除されません。Astra Trident がオフラインの間は、すでにプロビジョニング済みの PVS を引き続き使用でき、Astra Trident は、オンラインに戻った時点で作成された PVC に対してボリュームをプロビジョニングします。

作業を開始する前に

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

手順

1. のアンインストールコマンドを実行します tridentctl CRDと関連オブジェクトを除くAstra Tridentに関連付けられているすべてのリソースを削除する。

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

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



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 © 2024 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。