



# Upgrade Astra Trident

## Astra Trident

NetApp  
March 05, 2026

# Inhalt

Upgrade Astra Trident .....	1
Upgrade Astra Trident .....	1
Überlegungen vor dem Upgrade .....	1
Schritt 1: Wählen Sie eine Version .....	1
Schritt 2: Bestimmen Sie die ursprüngliche Installationsmethode .....	2
Schritt 3: Wählen Sie eine Upgrade-Methode .....	2
Änderungen am Operator .....	3
Upgrade mit dem Bediener .....	3
Upgrade über den Trident-Operator .....	3
Upgrade einer Installation des Trident Operators mit Cluster-Umfang .....	4
Aktualisieren einer Installation des Namespace-Scoped-Operators .....	5
Aktualisieren einer Helm-basierten Bedienerinstallation .....	9
Upgrade von einer nicht-Betreiber-Installation .....	9
Upgrade mit tridentctl .....	11
Aktualisieren Sie Astra Trident mit tridentctl .....	11
Aktualisieren Sie Volumes mithilfe von tridentctl .....	12

# Upgrade Astra Trident

## Upgrade Astra Trident

Astra Trident folgt einem vierteljährlichen Release-Intervall mit vier Hauptversionen pro Kalenderjahr. Jede neue Version baut auf den vorherigen Versionen auf und bietet neue Funktionen, Performance-Verbesserungen sowie Bug Fixes und Verbesserungen. Wir empfehlen Ihnen, ein Upgrade mindestens einmal pro Jahr durchzuführen, um von den neuen Funktionen in Astra Trident zu profitieren.

### Überlegungen vor dem Upgrade

Bei einem Upgrade auf die neueste Version von Astra Trident sollten Sie Folgendes berücksichtigen:

- In allen Namespaces in einem Kubernetes-Cluster sollte nur eine Astra Trident Instanz installiert werden.
- Ab Trident 20.01 gilt nur als Beta-Version von ["Volume Snapshots"](#) Wird unterstützt. Kubernetes-Administratoren sollten darauf achten, dass sie die Alpha-Snapshot-Objekte sicher in Beta-Version sichern oder in sie konvertieren können, um die älteren Alpha-Snapshots zu behalten.
  - CSI Volume Snapshots ist jetzt eine GA-Funktion, die ab Kubernetes 1.20 beginnt. Vor der Aktualisierung sollten Sie Alpha-Snapshot-CRDs mit entfernen `tridentctl obliviate alpha-snapshot-crd` So löschen Sie die CRDs für die Alpha-Snapshot-Spezifikation.
  - Die Beta-Version von Volume Snapshots führt zu einer geänderten Menge an CRDs und einem Snapshot Controller, die beide vor dem Upgrade von Astra Trident eingerichtet werden sollten.
  - Weitere Informationen finden Sie unter ["Was Sie Vor Dem Upgrade Ihres Kubernetes-Clusters Beachten Sollten"](#).
- Alle Upgrades von Version 19.04 und früheren Versionen erfordern die Migration der Astra Trident Metadaten von ihrem eigenen System `etcd` Zu CRD-Objekten. Überprüfen Sie die ["Dokumentation für Ihre Astra Trident Version"](#) Um zu verstehen, wie das Upgrade funktioniert.
- Beim Upgrade ist es wichtig, dass Sie das Upgrade durchführen `parameter.fsType In StorageClasses` Verwendet von Astra Trident. Sie können löschen und neu erstellen `StorageClasses` Ohne Unterbrechung vorhandener Volumes
  - Dies ist eine **Anforderung** für die Durchsetzung ["Sicherheitskontexte"](#) Für SAN-Volumes.
  - Das Verzeichnis [sample input](#) enthält Beispiele wie `storage-class-basic.yaml.template` und Link:<https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml>[`storage-class-bronze-default.yaml`^]. Weitere Informationen finden Sie unter ["Bekannt Probleme"](#).

### Schritt 1: Wählen Sie eine Version

Astra Trident Versionen folgen einem datumbasierten `YY.MM` Namensgebungskonvention, wobei „YY“ die letzten beiden Ziffern des Jahres und „MM“ der Monat ist. Dot Releases folgen einem `YY.MM.X` konvention, wo „X“ die Patch-Ebene ist. Sie wählen die Version, auf die Sie aktualisieren möchten, basierend auf der Version aus, von der Sie aktualisieren.

- Sie können ein direktes Upgrade auf jede Zielversion durchführen, die sich innerhalb eines Fensters mit vier Versionen Ihrer installierten Version befindet. Sie können beispielsweise direkt von 22.04 auf 23.04 aktualisieren (einschließlich beliebiger Dot-Versionen, z. B. 22.04.1).

- Wenn Sie eine frühere Version haben, sollten Sie eine mehrstufige Aktualisierung anhand der Dokumentation der jeweiligen Version durchführen, um spezifische Anweisungen zu erhalten. Dazu müssen Sie zuerst ein Upgrade auf die aktuellste Version durchführen, die zu Ihrem vier-Release-Fenster passt. Wenn Sie z. B. 18.07 ausführen und auf Version 20.07 aktualisieren möchten, befolgen Sie den Prozess zum mehrstufigen Upgrade wie unten angegeben:
  - a. Erstes Upgrade von 18.07 auf 19.07.
  - b. Dann aktualisieren Sie von 19.07 auf 20.07.



Wenn Sie ein Upgrade über den Trident-Operator auf der OpenShift Container Platform durchführen, sollten Sie auf Trident 21.01.1 oder höher aktualisieren. Der mit 21.01.0 veröffentlichte Trident-Operator enthält ein bekanntes Problem, das in 21.01.1 behoben wurde. Weitere Informationen finden Sie im ["Details zur Ausgabe auf GitHub"](#).

## Schritt 2: Bestimmen Sie die ursprüngliche Installationsmethode

Im Allgemeinen sollten Sie das Upgrade mit der gleichen Methode durchführen, die Sie für die Erstinstallation verwendet haben, wie Sie es können ["Wechseln Sie zwischen den Installationsmethoden"](#).

So ermitteln Sie, welche Version Sie ursprünglich für Astra Trident verwendet haben:

1. Nutzung `kubectl get pods - trident` Um die Pods zu untersuchen.
  - Wenn es keinen Operator Pod gibt, wurde Astra Trident mit installiert `tridentctl`.
  - Wenn es einen Operator Pod gibt, wurde Astra Trident entweder manuell oder über Helm mit dem Trident Operator installiert.
2. Wenn ein Benutzer-Pod vorhanden ist, verwenden Sie `kubectl describe tproc trident` Um festzustellen, ob Astra Trident mit Helm installiert wurde.
  - Wenn es ein Helm-Label gibt, wurde Astra Trident mit Helm installiert.
  - Wenn es kein Helm-Label gibt, wurde Astra Trident manuell über den Trident Operator installiert.

## Schritt 3: Wählen Sie eine Upgrade-Methode

Für ein Upgrade von Astra Trident gibt es zwei Methoden.

### Zeitpunkt der Aktualisierung mit dem Bediener

Das können Sie ["Upgrade mit dem Trident Operator"](#) Wenn:

- Sie haben Astra Trident ursprünglich mit dem Operator oder mit installiert `tridentctl`.
- Sie haben CSI Trident deinstalliert, und die Metadaten aus der Installation bleiben bestehen.
- Sie verfügen über eine CSI-basierte Astra Trident Installation. Alle Versionen von 19.07 sind CSI-basiert. Sie können die Pods in Ihrem Trident Namespace untersuchen, um Ihre Version zu überprüfen.
  - Pod-Benennung in Versionen vor 23.01 verwendet: `trident-csi-*`
  - Pod-Benennungen in 23.01 und höher:
    - `trident-controller-<generated id>` Für den Controller POD
    - `trident-node-<operating system>-<generated id>` Für die Node-Pods
    - `trident-operator-<generated id>` Für den Bediener-POD



Verwenden Sie den Operator zum Aktualisieren von Trident nicht, wenn Sie ein verwenden `etcd`-Based Trident Release (19.04 oder früher).

## Zeitpunkt für das Upgrade mit `tridentctl`

Das können Sie Falls Sie Astra Trident ursprünglich mit ``tridentctl`` installiert haben.

`tridentctl` ist die herkömmliche Methode zur Installation von Astra Trident und bietet die meisten Optionen für diejenigen, die komplexe Anpassungen erfordern. Weitere Informationen finden Sie unter "[Wählen Sie Ihre Installationsmethode](#)".

## Änderungen am Operator

Mit Version 21.01 von Astra Trident wurden Architekturänderungen für den Betreiber eingeführt:

- Der Operator ist jetzt **Cluster-scoped**. Vorherige Instanzen des Trident Operators (Versionen 20.04 bis 20.10) waren **Namespace-Scoped**. Ein Operator mit Cluster-Scoped ist aus den folgenden Gründen von Vorteil:
  - Resource Accountability: Der Operator managt jetzt die mit einer Astra Trident-Installation verbundenen Ressourcen auf Cluster-Ebene. Im Rahmen der Installation von Astra Trident erstellt und verwaltet der Bediener mehrere Ressourcen mit `ownerReferences`. Wartung `ownerReferences` Auf Cluster-Scoped-Ressourcen können Fehler bei bestimmten Kubernetes-Distributoren wie OpenShift auftreten. Diese Option wird durch einen Operator mit Cluster-Umfang entschärft. Für die automatische Reparatur und das Patching von Trident-Ressourcen ist dies eine wesentliche Anforderung.
  - Aufräumarbeiten während der Deinstallation: Eine vollständige Entfernung von Astra Trident würde alle damit verbundenen Ressourcen zu löschen benötigen. Ein Operator mit Namespace-Scoped kann Probleme beim Entfernen von Cluster-Scoped-Ressourcen (wie `clusterRole`, `ClusterRoleBinding` und `PodSecurityPolicy`) haben und eine unvollständige Bereinigung zur Folge haben. Ein Operator mit Cluster-Umfang beseitigt dieses Problem. Benutzer können Astra Trident vollständig deinstallieren und bei Bedarf neu installieren.
- `TridentProvisioner` Wird nun durch ersetzt `TridentOrchestrator` Als benutzerdefinierte Ressource, die für die Installation und das Management von Astra Trident verwendet wird. Darüber hinaus wird dem ein neues Feld vorgestellt `TridentOrchestrator` Spez. Benutzer können angeben, dass der Namespace Trident über den installiert/aktualisiert werden muss `spec.namespace` Feld. Sie können sich ein Beispiel ansehen "[Hier](#)".

## Upgrade mit dem Bediener

Eine vorhandene Astra Trident Installation lässt sich ganz einfach entweder manuell oder über Helm aufrüsten.

## Upgrade über den Trident-Operator

Generell sollten Sie Astra Trident mit der gleichen Methode aktualisieren, die zur ursprünglichen Installation verwendet wurde. Prüfen "[Wählen Sie eine Aktualisierungsmethode aus](#)" Bevor Sie ein Upgrade mit dem Trident-Operator durchführen.

Bei einem Upgrade von einer Instanz von Astra Trident, die über den Namespace-Scoped-Operator (Versionen 20.07 bis 20.10) installiert wurde, führt der Trident-Operator automatisch folgende Schritte aus:



- Migriert `tridentProvisioner` Zu A `tridentOrchestrator` Objekt mit demselben Namen,
- Löscht `TridentProvisioner` Objekte und das `tridentprovisioner` CRD.
- Aktualisiert Astra Trident auf die Version des verwendeten Cluster-Scoped-Operators
- Astra Trident sollte in dem Namespace installiert werden, in dem er ursprünglich installiert wurde

## Upgrade einer Installation des Trident Operators mit Cluster-Umfang

Sie können eine Installation eines Trident-Operators mit Cluster-Umfang aktualisieren. Alle Astra Trident Versionen 21.01 und höher verwenden einen Operator mit Cluster-Umfang.

### Bevor Sie beginnen

Stellen Sie sicher, dass Sie ein Kubernetes-Cluster ausführen "[Eine unterstützte Kubernetes Version](#)".

### Schritte

1. Überprüfen Sie die Astra Trident Version:

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

2. Löschen Sie den Trident-Operator, der zur Installation der aktuellen Astra Trident-Instanz verwendet wurde. Wenn Sie beispielsweise ein Upgrade von 22.01 durchführen, führen Sie den folgenden Befehl aus:

```
kubectl delete -f 22.01/trident-installer/deploy/bundle.yaml -n trident
```

3. Wenn Sie Ihre Erstinstallation mit angepasst haben `TridentOrchestrator` Attribute, können Sie die bearbeiten `TridentOrchestrator` Objekt zum Ändern der Installationsparameter. Dies kann auch Änderungen umfassen, die an der Angabe gespiegelter Trident- und CSI-Image-Register für den Offline-Modus vorgenommen wurden, Debug-Protokolle aktivieren oder Geheimnisse für die Bildausziehung angeben.
4. Installieren Sie Astra Trident mit der richtigen Paket-YAML-Datei für Ihre Umgebung und Astra Trident-Version. Wenn Sie beispielsweise Astra Trident 23.04 für Kubernetes 1.27 installieren, führen Sie den folgenden Befehl aus:

```
kubectl create -f 23.04.0/trident-installer/deploy/bundle_post_1_25.yaml -n trident
```



Trident stellt eine Paketdatei bereit, mit der der Operator installiert und damit verknüpfte Objekte für die Kubernetes-Version erstellt werden können.

- Verwenden Sie für Cluster mit Kubernetes 1.24 oder früheren Versionen "[Bundle\\_pre\\_1\\_25.yaml](#)".
- Verwenden Sie für Cluster mit Kubernetes 1.25 oder höher "[Bundle\\_Post\\_1\\_25.yaml](#)".

## Ergebnisse

Der Trident-Operator identifiziert eine bestehende Astra Trident-Installation und aktualisiert sie auf die gleiche Version wie der Operator.

## Aktualisieren einer Installation des Namespace-Scoped-Operators

Sie können ein Upgrade von einer Instanz von Astra Trident, die mit dem Namespace-Scoped Operator (Versionen 20.07 bis 20.10) installiert ist, auf eine Cluster-Scoped Operator-Installation durchführen.

### Bevor Sie beginnen

Sie benötigen die Paket-YAML-Datei, die zum Bereitstellen des Namespace-Scoped-Operators verwendet wird <https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/BUNDLE.YAML> Wo *vXX.XX* Ist die Versionsnummer und *BUNDLE.YAML* Ist der Name der Paket-YAML-Datei.

### Schritte

1. Überprüfen Sie die `TridentProvisioner` Der Status der bestehenden Trident-Installation lautet `Installed`.

```
kubectl describe tprov trident -n trident | grep Message: -A 3

Message:  Trident installed
Status:   Installed
Version:  v20.10.1
```



Wenn der Status angezeigt wird `Updating`, Stellen Sie sicher, dass Sie es lösen, bevor Sie fortfahren. Eine Liste möglicher Statuswerte finden Sie unter "[Hier](#)".

2. Erstellen Sie die `TridentOrchestrator` CRD mit dem Manifest, das mit dem Trident-Installer bereitgestellt wurde.

```
# Download the release required [23.04.0]
mkdir 23.04.0
cd 23.04.0
wget
https://github.com/NetApp/trident/releases/download/v23.04.0/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.tar.gz
cd trident-installer
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Löschen Sie den Operator Namespace-Scoped mithilfe des Manifests.

a. Stellen Sie sicher, dass Sie sich im richtigen Verzeichnis befinden.

```
pwd
/root/20.10.1/trident-installer
```

b. Löschen Sie den Namespace-Scoped-Operator.

```
kubectl delete -f deploy/<BUNDLE.YAML> -n trident

serviceaccount "trident-operator" deleted
clusterrole.rbac.authorization.k8s.io "trident-operator" deleted
clusterrolebinding.rbac.authorization.k8s.io "trident-operator"
deleted
deployment.apps "trident-operator" deleted
podsecuritypolicy.policy "tridentoperatorpods" deleted
```

c. Bestätigen Sie, dass der Trident-Operator entfernt wurde.

```
kubectl get all -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
pod/trident-csi-68d979fb85-dsrmn	6/6	Running	12	99d
pod/trident-csi-8jfhf	2/2	Running	6	105d
pod/trident-csi-jtnjz	2/2	Running	6	105d
pod/trident-csi-lcxvh	2/2	Running	8	105d

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/trident-csi	ClusterIP	10.108.174.125	<none>
34571/TCP, 9220/TCP	105d		

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AGE
daemonset.apps/trident-csi	3	3	3	3	
3					105d
					kubernetes.io/arch=amd64, kubernetes.io/os=linux

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/trident-csi	1/1	1	1	105d

NAME	DESIRED	CURRENT	READY
replicaset.apps/trident-csi-68d979fb85	1	1	1
105d			

4. (Optional) Wenn die Installationsparameter geändert werden müssen, aktualisieren Sie den TridentProvisioner Spez. Dies kann Änderungen wie Ändern beinhalten: Die Werte für tridentImage, autosupportImage, Ein privates Image Repository von und das imagePullSecrets) Nach dem Löschen des Operator Namespace-scoped und vor der Installation des Operators Cluster-scoped. Eine vollständige Liste der Parameter, die aktualisiert werden können, finden Sie im ["Konfigurationsoptionen"](#).

```
kubectl patch tprov <trident-provisioner-name> -n <trident-namespace> --type=merge -p '{"spec":{"debug":true}}'
```

5. Installieren Sie den Trident-Operator für den Cluster-Umfang.

- a. Stellen Sie sicher, dass Sie sich im richtigen Verzeichnis befinden.

```
pwd  
/root/23.04.0/trident-installer
```

- b. Installieren Sie den Cluster-scoped-Operator im gleichen Namespace.

Trident stellt eine Paketdatei bereit, mit der der Operator installiert und damit verknüpfte Objekte für die Kubernetes-Version erstellt werden können.



- Verwenden Sie für Cluster mit Kubernetes 1.24 oder früheren Versionen ["Bundle\\_pre\\_1\\_25.yaml"](#).
- Verwenden Sie für Cluster mit Kubernetes 1.25 oder höher ["Bundle\\_Post\\_1\\_25.yaml"](#).

```
kubectl create -f deploy/<BUNDLE.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

#All tridentProvisioners will be removed, including the CRD itself
kubectl get tprov -n trident
Error from server (NotFound): Unable to list "trident.netapp.io/v1,
Resource=tridentprovisioners": the server could not find the
requested resource (get tridentprovisioners.trident.netapp.io)

#tridentProvisioners are replaced by tridentOrchestrator
kubectl get torc
NAME          AGE
trident      13s
```

- c. Untersuchen Sie die Trident Pods im Namespace. Der `trident-controller` Zudem geben die Pod-Namen die im Jahr 23.01 eingeführte Namenskonvention wieder.

```
kubectl get pods -n trident

NAME                                READY   STATUS    RESTARTS
AGE
trident-controller-79df798bdc-m79dc  6/6    Running   0
1m41s
trident-node-linux-xrst8            2/2    Running   0
1m41s
trident-operator-5574dbbc68-nthjv    1/1    Running   0
1m52s
```

- d. Bestätigen Sie, dass Trident auf die beabsichtigte Version aktualisiert wurde.

```
kubectl describe torc trident | grep Message -A 3
Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v23.04.0
```

## Aktualisieren einer Helm-basierten Bedienerinstallation

Führen Sie die folgenden Schritte durch, um eine Helm-basierte Bedienerinstallation zu aktualisieren.



Wenn Sie ein Kubernetes-Cluster von 1.24 auf 1.25 oder höher aktualisieren, auf das Astra Trident installiert ist, müssen Sie Werte.yaml aktualisieren `excludePodSecurityPolicy` Bis `true` Oder hinzufügen `--set excludePodSecurityPolicy=true` Bis zum `helm upgrade` Befehl bevor Sie ein Upgrade des Clusters durchführen können.

### Schritte

1. Laden Sie die neueste Version von Astra Trident herunter.
2. Verwenden Sie die `helm upgrade` Befehl wo `trident-operator-23.04.0.tgz` Gibt die Version an, auf die Sie ein Upgrade durchführen möchten.

```
helm upgrade <name> trident-operator-23.04.0.tgz
```

Wenn Sie während der Erstinstallation alle nicht standardmäßigen Optionen festlegen (z. B. Private, gespiegelte Registries für Trident- und CSI-Images), verwenden Sie `--set` Um sicherzustellen, dass diese Optionen im Upgrade-Befehl enthalten sind, werden die Werte andernfalls auf die Standardeinstellung zurückgesetzt.



Um beispielsweise den Standardwert von `tridentDebug` zu ändern, Ausführen des folgenden Befehls:

```
helm upgrade <name> trident-operator-23.04.0-custom.tgz --set
tridentDebug=true
```

3. Laufen `helm list` Um zu überprüfen, ob sowohl die Karten- als auch die App-Version aktualisiert wurden. Laufen `tridentctl logs` Um alle Debug-Nachrichten zu überprüfen.

### Ergebnisse

Der Trident-Operator identifiziert eine bestehende Astra Trident-Installation und aktualisiert sie auf die gleiche Version wie der Operator.

## Upgrade von einer nicht-Betreiber-Installation

Sie können ein Upgrade auf die neueste Version des Trident-Operators von durchführen `tridentctl` Installation:

## Schritte

1. Laden Sie die neueste Version von Astra Trident herunter.

```
# Download the release required [23.04.0]
mkdir 23.04.0
cd 23.04.0
wget
https://github.com/NetApp/trident/releases/download/v22.01.1/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.tar.gz
cd trident-installer
```

2. Erstellen Sie die tridentorchestrator CRD aus dem Manifest.

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

3. Stellen Sie den Clusteroperator im selben Namespace bereit.

```
kubectl create -f deploy/<BUNDLE.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. Erstellen Sie ein TridentOrchestrator CR für die Installation von Astra Trident.

```

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. Bestätigen Sie, dass das Upgrade von Trident auf die beabsichtigte Version durchgeführt wurde.

```

kubectl describe torc trident | grep Message -A 3

Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v23.04.0

```

## Ergebnisse

Die vorhandenen Back-Ends und PVCs stehen automatisch zur Verfügung.

## Upgrade mit tridentctl

Sie können mithilfe von ganz einfach eine bestehende Astra Trident Installation aufrüsten tridentctl.

### Aktualisieren Sie Astra Trident mit tridentctl

Deinstallation und Neuinstallation von Astra Trident fungiert als Upgrade. Bei der Deinstallation von Trident werden die von der Astra Trident Implementierung verwendeten Persistent Volume Claim (PVC) und Persistent Volume (PV) nicht gelöscht. PVS, die bereits bereitgestellt wurden, bleiben verfügbar, während Astra Trident offline ist. Astra Trident stellt Volumes für alle PVCs bereit, die in der Zwischenzeit erstellt werden, sobald sie wieder online sind.

### Bevor Sie beginnen

Prüfen ["Wählen Sie eine Aktualisierungsmethode aus"](#) Vor der Aktualisierung mit tridentctl.

## Schritte

1. Führen Sie den Deinstallationsbefehl in aus `tridentctl` So entfernen Sie alle mit Astra Trident verbundenen Ressourcen mit Ausnahme der CRDs und zugehörigen Objekte.

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

2. Installieren Sie Astra Trident Neu. Siehe "[Installieren Sie Astra Trident mit tridentctl](#)".



Unterbrechen Sie den Upgrade-Prozess nicht. Stellen Sie sicher, dass das Installationsprogramm bis zum Abschluss ausgeführt wird.

## Aktualisieren Sie Volumes mithilfe von `tridentctl`

Nach einem Upgrade können Sie die umfassenden Funktionen nutzen, die in neueren Trident Versionen verfügbar sind (wie beispielsweise On-Demand Volume Snapshots), die Sie für ein Upgrade der Volumes verwenden können `tridentctl upgrade` Befehl.

Wenn bereits ältere Volumes vorhanden sind, sollten Sie diese von einem NFS- oder iSCSI-Typ auf den CSI-Typ aktualisieren, um die umfassenden neuen Funktionen von Astra Trident zu verwenden. Ein von Trident bereitgestelltes Legacy-PV unterstützt die herkömmlichen Funktionen.

### Bevor Sie beginnen

Beachten Sie Folgendes, bevor Sie sich für ein Upgrade von Volumes auf den CSI-Typ entscheiden:

- Möglicherweise müssen Sie nicht alle Volumes aktualisieren. Zuvor erstellte Volumes sind weiterhin zugänglich und funktionieren ordnungsgemäß.
- Ein PV kann als Teil einer Deployment/StatupfulSet installiert werden. Es ist nicht erforderlich, die Implementierung/StatesSet herunterzufahren.
- Sie können ein PV nicht an einen eigenständigen Pod anschließen, wenn Sie ein Upgrade durchführen. Sie sollten den POD herunterfahren, bevor Sie das Volume aktualisieren.
- Sie können nur ein Volume aktualisieren, das an eine PVC gebunden ist. Volumes, die nicht an PVCs gebunden sind, sollten vor dem Upgrade entfernt und importiert werden.

## Schritte

1. Laufen `kubectl get pv` So Listen Sie die PVS auf.

```

kubect1 get pv
NAME                                CAPACITY    ACCESS MODES    RECLAIM POLICY
STATUS    CLAIM                                STORAGECLASS    REASON    AGE
default-pvc-1-a8475                1073741824    RWO                                Delete
Bound    default/pvc-1                        standard
default-pvc-2-a8486                1073741824    RWO                                Delete
Bound    default/pvc-2                        standard
default-pvc-3-a849e                1073741824    RWO                                Delete
Bound    default/pvc-3                        standard
default-pvc-4-a84de                1073741824    RWO                                Delete
Bound    default/pvc-4                        standard
trident                             2Gi          RWO                                Retain
Bound    trident/trident                      19h

```

Derzeit gibt es vier PVS, die von Trident 20.07 mit dem erstellt wurden `netapp.io/trident` bereitstellung:

## 2. Laufen `kubect1 describe pv` Um die Details zum PV zu erhalten.

```

kubect1 describe pv default-pvc-2-a8486

Name:                default-pvc-2-a8486
Labels:              <none>
Annotations:         pv.kubernetes.io/provisioned-by: netapp.io/trident
                    volume.beta.kubernetes.io/storage-class: standard
Finalizers:          [kubernetes.io/pv-protection]
StorageClass:        standard
Status:              Bound
Claim:               default/pvc-2
Reclaim Policy:      Delete
Access Modes:        RWO
VolumeMode:          Filesystem
Capacity:            1073741824
Node Affinity:       <none>
Message:
Source:
  Type:              NFS (an NFS mount that lasts the lifetime of a pod)
  Server:            10.xx.xx.xx
  Path:              /trid_1907_alpha_default_pvc_2_a8486
  ReadOnly:          false

```

Das PV wurde mit Hilfe des erstellt `netapp.io/trident` bereitstellung vom Typ NFS. Um alle neuen Funktionen von Astra Trident zu unterstützen, sollte dieses PV auf den CSI-Typ aufgerüstet werden.

## 3. Führen Sie die aus `tridentctl upgrade volume <name-of-trident-volume>` Befehl zum

## Upgrade eines alten Astra Trident Volumes auf die CSI-Spezifikation.

```
./tridentctl get volumes -n trident
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS | PROTOCOL |
BACKEND UUID            | STATE  | MANAGED      |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| default-pvc-2-a8486 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
| default-pvc-3-a849e | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
| default-pvc-1-a8475 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
| default-pvc-4-a84de | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

./tridentctl upgrade volume default-pvc-2-a8486 -n trident
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS | PROTOCOL |
BACKEND UUID            | STATE  | MANAGED      |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| default-pvc-2-a8486 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

4. A ausführen `kubectl describe pv` Um zu überprüfen, ob es sich bei dem Volumen um ein CSI-Volumen handelt.

```
kubectl describe pv default-pvc-2-a8486
Name:                default-pvc-2-a8486
Labels:              <none>
Annotations:         pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
                    volume.beta.kubernetes.io/storage-class: standard
Finalizers:          [kubernetes.io/pv-protection]
StorageClass:        standard
Status:              Bound
Claim:               default/pvc-2
Reclaim Policy:      Delete
Access Modes:        RWO
VolumeMode:          Filesystem
Capacity:            1073741824
Node Affinity:       <none>
Message:
Source:
  Type:              CSI (a Container Storage Interface (CSI) volume
source)
  Driver:            csi.trident.netapp.io
  VolumeHandle:      default-pvc-2-a8486
  ReadOnly:          false
  VolumeAttributes:  backendUUID=c5a6f6a4-b052-423b-80d4-
8fb491a14a22

internalName=trid_1907_alpha_default_pvc_2_a8486
                    name=default-pvc-2-a8486
                    protocol=file
Events:              <none>
```

## Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtlich geschützten Urhebers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFT SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

## Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.