



Management und Monitoring von Astra Trident

Astra Trident

NetApp
January 14, 2026

This PDF was generated from <https://docs.netapp.com/de-de/trident-2406/trident-managing-k8s/upgrade-trident.html> on January 14, 2026. Always check docs.netapp.com for the latest.

Inhalt

Management und Monitoring von Astra Trident	1
Upgrade Astra Trident	1
Upgrade Astra Trident	1
Upgrade mit dem Bediener	2
Upgrade mit tridentctl	7
Managen Sie Astra Trident mit tridentctl	7
Befehle und globale Alarmmeldungen	7
Befehlsoptionen und -Flags	9
Überwachen Sie Astra Trident	15
Überblick	15
Schritt 1: Definieren Sie ein Prometheus-Ziel	15
Schritt: Erstellen Sie einen Prometheus ServiceMonitor	15
Schritt 3: Abfrage der Trident-Kennzahlen mit PromQL	16
AutoSupport Telemetrie von Astra Trident mit Thema	17
Deaktivieren Sie Astra Trident Metriken	18
Deinstallieren Sie Astra Trident	18
Bestimmen Sie die ursprüngliche Installationsmethode	19
Deinstallieren Sie die Installation eines Trident-Operators	19
Deinstallieren Sie eine tridentctl Installation	20

Management und Monitoring von Astra Trident

Upgrade Astra Trident

Upgrade Astra Trident

Ab Version 24.02 folgt Astra Trident einem viermonatigen Release-Intervall und liefert drei wichtige Releases pro Kalenderjahr. Jede neue Version baut auf den vorherigen Versionen auf und bietet neue Funktionen, Performance-Verbesserungen, 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.
- Astra Trident 23.07 und höher benötigt v1-Volume-Snapshots und unterstützt keine Alpha- oder Beta-Snapshots mehr.
- Wenn Sie Cloud Volumes Service für Google Cloud im erstellt "[CVS-Diensttyp](#)" haben, müssen Sie die Backend-Konfiguration aktualisieren, um beim Upgrade von Astra Trident 23.01 den Service-Level oder `zoneredundantstandardsw` zu verwenden `standardsw`. Wenn das im Backend nicht aktualisiert `serviceLevel` wird, kann es zu einem Fehlschlagen der Volumes kommen. Weitere Informationen finden Sie unter "[Beispiele für CVS-Diensttypen](#)".
- Es ist wichtig, dass Sie beim `StorageClasses` Upgrade von Astra Trident verwendet angeben `parameter.fsType`. Sie können löschen und neu erstellen `StorageClasses`, ohne bereits vorhandene Volumes zu unterbrechen.
 - Dies ist eine **Anforderung** für die Durchsetzung von "[Sicherheitskontexte](#)" 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/default-storage-class-aml-^Bronze\[storage-class-bronze-default.yaml\]](https://github.com/NetApp/Trident/blob/master/Trident-Installer/sample-input/Storage-class-Samples/default-storage-class-aml-^Bronze[storage-class-bronze-default.yaml]).
 - Weitere Informationen finden Sie unter "[Bekannte Probleme](#)".

Schritt 1: Wählen Sie eine Version

Astra Trident-Versionen folgen einer datumbasierten Namenskonvention `YY.MM`, wobei „YY“ die letzten beiden Ziffern des Jahres und „MM“ den Monat darstellt. Dot-Releases folgen einer `YY.MM.X` Konvention, wobei „X“ der Patch-Level 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 23.04 (oder einem beliebigen 23.04-Punkt-Release) auf 24.06 aktualisieren.
- Wenn Sie ein Upgrade von einer Version außerhalb des Fensters mit vier Releases durchführen, führen Sie ein Upgrade in mehreren Schritten durch. Verwenden Sie die Upgrade-Anweisungen für das, von dem "[Frühere Version](#)" Sie aktualisieren, um auf die neueste Version zu aktualisieren, die für das Fenster mit vier Versionen passt. Wenn Sie beispielsweise 22.01 verwenden und ein Upgrade auf 24.06 durchführen möchten:

- a. Erstes Upgrade von 22.07 auf 23.04.
- b. Dann Upgrade von 23.04 auf 24.06.



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

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

1. Verwenden Sie, `kubectl get pods -n 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. Falls es einen Pod gibt, mit können Sie `kubectl describe torc` feststellen, 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

Im Allgemeinen sollten Sie mit der gleichen Methode aktualisieren, die Sie für die Erstinstallation verwendet haben, jedoch können Sie ["Wechseln Sie zwischen den Installationsmethoden"](#). Astra Trident bietet zwei Optionen für ein Upgrade.

- ["Upgrade über den Trident-Operator"](#)



Wir empfehlen Ihnen, die Überprüfung ["Den Upgrade-Workflow für Bediener verstehen"](#) durchzuführen, bevor Sie mit dem Betreiber ein Upgrade durchführen.

*

Upgrade mit dem Bediener

Den Upgrade-Workflow für Bediener verstehen

Bevor Sie ein Upgrade von Astra Trident mit dem Trident-Operator durchführen, sollten Sie sich über die während des Upgrades auftretenden Hintergrundprozesse informieren. Dies umfasst Änderungen am Trident Controller, am Controller Pod und an Node-Pods sowie am Node-DemonSet, die Rolling-Updates ermöglichen.

Bearbeitung von Trident Upgrades für Betreiber

Einer der vielen ["Vorteile der Verwendung des Trident-Bediener"](#) Installationen und Upgrades von Astra Trident ist die automatische Verarbeitung von Astra Trident und Kubernetes-Objekten, ohne vorhandene gemountete Volumes zu unterbrechen. So kann Astra Trident Upgrades ohne Ausfallzeiten oder auch ohne Ausfallzeiten unterstützen. ["Rollierende Updates"](#) Insbesondere kommuniziert der Trident Betreiber mit dem

Kubernetes-Cluster, um:

- Löschen Sie die Trident Controller-Implementierung und den Node DemonSet und erstellen Sie sie neu.
- Ersetzen Sie den Trident Controller Pod und die Trident Node Pods durch neue Versionen.
 - Wenn ein Node nicht aktualisiert wird, verhindert dies nicht, dass die verbleibenden Nodes aktualisiert werden.
 - Nur Nodes mit einem laufenden Trident Node Pod können Volumes mounten.



Weitere Informationen zur Architektur von Astra Trident auf dem Kubernetes-Cluster finden Sie unter "[Die Architektur von Astra Trident](#)".

Arbeitsablauf für die Benutzeraktualisierung

Wenn Sie ein Upgrade mit dem Trident Operator initiieren:

1. Der **Trident-Operator**:

- a. Erkennt die aktuell installierte Version von Astra Trident (Version n).
- b. Aktualisiert alle Kubernetes-Objekte einschließlich CRDs, RBAC und Trident SVC.
- c. Löscht die Trident Controller-Bereitstellung für Version n .
- d. Erstellt die Trident-Controller-Bereitstellung für Version $n+1$.

2. **Kubernetes** erstellt Trident Controller Pod für $n+1$.

3. Der **Trident-Operator**:

- a. Löscht das Trident Node DemonSet für n . Der Operator wartet nicht auf die Beendigung des Node-Pod.
- b. Erstellt den Trident Node Demonset für $n+1$.

4. **Kubernetes** erstellt Trident Node Pods auf Nodes, auf denen Trident Node Pod n nicht ausgeführt wird. So wird sichergestellt, dass auf einem Node nie mehr als ein Trident Node Pod einer beliebigen Version vorhanden ist.

Upgrade einer Astra Trident Installation mit dem Trident Operator oder Helm

Sie können ein Upgrade von Astra Trident mit dem Trident Operator entweder manuell oder mit Helm durchführen. Sie können von einer Trident-Bedienerinstallation auf eine andere Trident-Bedienerinstallation aktualisieren oder von einer Installation auf eine Trident-Bedienerversion aktualisieren `tridentctl`. Vor dem Upgrade einer Trident-Bedienerinstallation überprüfen "[Wählen Sie eine Aktualisierungsmethode aus](#)".

Aktualisieren einer manuellen Installation

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



Um ein Upgrade von Astra Trident durchzuführen, das mit dem Namespace-Scoped-Operator (Versionen 20.07 bis 20.10) installiert wurde, verwenden Sie die Upgrade-Anweisungen für "[Ihre installierte Version](#)" Astra Trident.

Über diese Aufgabe

Trident bietet eine Bundle-Datei, mit der Sie den Operator installieren und zugehörige Objekte für Ihre Kubernetes-Version erstellen können.

- Verwenden Sie für Cluster mit Kubernetes 1.24 ["Bundle_pre_1_25.yaml"](#).
- Verwenden Sie für Cluster mit Kubernetes 1.25 oder höher ["Bundle_Post_1_25.yaml"](#).

Bevor Sie beginnen

Stellen Sie sicher, dass Sie ein Kubernetes Cluster verwenden ["Eine unterstützte Kubernetes Version"](#), das ausgeführt wird.

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 23.07 durchführen, führen Sie den folgenden Befehl aus:

```
kubectl delete -f 23.07.0/trident-installer/deploy/<bundle.yaml> -n trident
```

3. Wenn Sie Ihre Erstinstallation mithilfe von Attributen angepasst haben `TridentOrchestrator`, können Sie das Objekt bearbeiten `TridentOrchestrator`, um die Installationsparameter zu ändern. 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 YAML-Bundle-Datei für Ihre Umgebung, wobei `<bundle.yaml>` auf Ihrer Kubernetes-Version basiert oder ist `bundle_pre_1_25.yaml` `bundle_post_1_25.yaml`. Wenn Sie beispielsweise Astra Trident 24.06 installieren, führen Sie den folgenden Befehl aus:

```
kubectl create -f 24.06.0/trident-installer/deploy/<bundle.yaml> -n trident
```

Aktualisieren einer Helm-Installation

Sie können ein Upgrade für eine Astra Trident Helm Installation durchführen.



Wenn Sie ein Kubernetes-Cluster von 1.24 auf 1.25 oder höher aktualisieren, auf dem Astra Trident installiert ist, müssen Sie `values.yaml` aktualisieren, damit Sie den `helm upgrade` Befehl auf `true` festlegen `excludePodSecurityPolicy` oder hinzufügen `--set excludePodSecurityPolicy=true` können, bevor Sie das Cluster aktualisieren können.

Schritte

1. Wenn Sie ["Astra Trident mit Helm installiert"](#), können Sie verwenden `helm upgrade trident netapp-`

`trident/trident-operator --version 100.2406.0`, um ein Upgrade in einem Schritt. Wenn Sie den Helm Repo nicht hinzugefügt haben oder ihn nicht zum Upgrade verwenden können:

- a. Laden Sie die neueste Version von Astra Trident von ["Die Sektion Assets auf GitHub"](#)herunter.
- b. Verwenden Sie den `helm upgrade` Befehl where zeigt die Version an `trident-operator-24.06.0.tgz`, auf die Sie aktualisieren möchten.

```
helm upgrade <name> trident-operator-24.06.0.tgz
```



Wenn Sie während der Erstinstallation benutzerdefinierte Optionen festlegen (z. B. `private`, gespiegelte Registrierungen für Trident- und CSI-Images angeben), fügen Sie den Befehl mit `--set` an `helm upgrade`, um sicherzustellen, dass diese Optionen im Aktualisierungsbefehl enthalten sind, andernfalls werden die Werte auf die Standardeinstellung zurückgesetzt.

2. Führen Sie aus `helm list`, um zu überprüfen, ob die Karte und die App-Version aktualisiert wurden. Ausführen `tridentctl logs`, um alle Debug-Meldungen zu überprüfen.

Upgrade von einer `tridentctl` Installation auf einen Trident-Operator

Sie können von einer Installation aus auf die neueste Version des Trident-Bedieners aktualisieren `tridentctl`. Die vorhandenen Back-Ends und VES stehen automatisch zur Verfügung.



Bevor Sie zwischen den Installationsmethoden wechseln, lesen Sie ["Wechseln zwischen den Installationsmethoden"](#).

Schritte

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

```
# Download the release required [24.060.0]
mkdir 24.06.0
cd 24.06.0
wget
https://github.com/NetApp/trident/releases/download/v24.06.0/trident-
installer-24.06.0.tar.gz
tar -xf trident-installer-24.06.0.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-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. 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:          v24.06.0
```


Upgrade mit tridentctl

Sie können eine bestehende Astra Trident Installation ganz einfach mit aktualisieren `tridentctl`.

Über diese Aufgabe

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

Überprüfen Sie ["Wählen Sie eine Aktualisierungsmethode aus"](#) vor dem Upgrade mit `tridentctl`.

Schritte

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

```
./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.

Managen Sie Astra Trident mit tridentctl

```
https://github.com/NetApp/trident/releases["Trident
Installationspaket"^]Im ist das Befehlszeilendienstprogramm für den
einfachen Zugriff auf Astra Trident enthalten `tridentctl`. Kubernetes-
Benutzer mit ausreichenden Berechtigungen können damit Astra Trident
installieren oder den Namespace managen, der den Astra Trident Pod
enthält.
```

Befehle und globale Alarmmeldungen

Sie können ausführen `tridentctl help`, um eine Liste der verfügbaren Befehle für `tridentctl` oder hängen Sie das Flag an `--help` einen beliebigen Befehl, um eine Liste der Optionen und Flags für diesen bestimmten Befehl zu erhalten.

```
tridentctl [command] [--optional-flag]
```

Das Dienstprogramm Astra Trident `tridentctl` unterstützt die folgenden Befehle und Global Flags.

Befehle

create

Ressource zu Astra Trident hinzufügen.

delete

Entfernen Sie eine oder mehrere Ressourcen aus Astra Trident.

get

Holen Sie sich eine oder mehrere Ressourcen von Astra Trident.

help

Hilfe zu jedem Befehl.

images

Drucken Sie eine Tabelle der Container-Images, die Astra Trident benötigt.

import

Importieren Sie eine vorhandene Ressource in Astra Trident.

install

Installation Von Astra Trident:

logs

Protokolle aus Astra Trident drucken.

send

Senden Sie eine Ressource von Astra Trident.

uninstall

Deinstallieren Sie Astra Trident.

update

Ändern Sie eine Ressource in Astra Trident.

update backend state

Vorübergehende Unterbrechung der Back-End-Vorgänge.

upgrade

Aktualisieren Sie eine Ressource in Astra Trident.

version

Drucken Sie die Version von Astra Trident.

Globale Alarmmeldungen

-d, --debug

Debug-Ausgabe.

-h, --help

Hilfe für `tridentctl`.

-k, --kubeconfig string

Geben Sie den Pfad an, über den Befehle lokal oder von einem Kubernetes-Cluster zu einem anderen ausgeführt werden `KUBECONFIG` sollen.



Alternativ können Sie die Variable exportieren `KUBECONFIG`, um auf ein bestimmtes Kubernetes-Cluster zu verweisen und Befehle an dieses Cluster auszugeben `tridentctl`.

-n, --namespace string

Namespace für die Astra Trident-Implementierung.

-o, --output string

Ausgabeformat. Einer von `json` `yaml`-Namen natürlich `Ärmellos` (Standard).

-s, --server string

Adresse/Port der Astra Trident REST-Schnittstelle



Die Trident REST-Schnittstelle kann nur für die Wiedergabe unter `127.0.0.1` (für IPv4) oder `[: 1]` (für IPv6) konfiguriert werden.

Befehlsoptionen und -Flags

Erstellen

Mit dem `create` Befehl fügen Sie Astra Trident eine Ressource hinzu.

```
tridentctl create [option]
```

Optionen

`backend`: Fügen Sie ein Backend zu Astra Trident.

Löschen

Mit dem `delete` Befehl entfernen Sie eine oder mehrere Ressourcen aus Astra Trident.

```
tridentctl delete [option]
```

Optionen

`backend`: Löschen eines oder mehrerer Speicher-Backends aus Astra Trident.

`snapshot`: Löschen Sie einen oder mehrere Volume-Snapshots aus Astra Trident.

`storageclass`: Löschen einer oder mehrerer Storage-Klassen aus Astra Trident.
`volume`: Löschen eines oder mehrerer Speichervolumes aus Astra Trident.

Get

Mit dem `get` Befehl rufen Sie eine oder mehrere Ressourcen aus Astra Trident ab.

```
tridentctl get [option]
```

Optionen

`backend`: Holen Sie sich ein oder mehrere Speicher-Backends von Astra Trident.
`snapshot`: Holen Sie sich einen oder mehrere Schnappschüsse von Astra Trident.
`storageclass`: Holen Sie sich eine oder mehrere Storage-Klassen von Astra Trident.
`volume`: Holen Sie sich einen oder mehrere Bände von Astra Trident.

Flags

`-h, --help`: Hilfe für Bände.
`--parentOfSubordinate string`: Abfrage auf untergeordneten Quellvolume beschränken.
`--subordinateOf string`: Abfrage auf Untergebene des Volumens beschränken.

Bilder

Verwenden Sie `images` Flags, um eine Tabelle der Container-Bilder zu drucken, die Astra Trident benötigt.

```
tridentctl images [flags]
```

Flags

`-h, --help`: Hilfe für Bilder.
`-v, --k8s-version string`: Semantische Version des Kubernetes-Clusters.

Importvolumen

Importieren Sie ein vorhandenes Volume mit dem `import volume` Befehl in Astra Trident.

```
tridentctl import volume <backendName> <volumeName> [flags]
```

Aliase

`volume, v`

Flags

`-f, --filename string`: Pfad zur YAML- oder JSON-PVC-Datei.
`-h, --help`: Hilfe für Volumen.
`--no-manage`: Erstellen Sie nur PV/PVC. Nehmen Sie kein Lifecycle Management für Volumes an.

Installieren

Verwenden Sie die `install` Flags, um Astra Trident zu installieren.

```
tridentctl install [flags]
```

Flags

`--autosupport-image` string: Das Containerbild für die AutoSupport Telemetrie (Standard "NetApp/Trident AutoSupport:<current-version>").

`--autosupport-proxy` string: Adresse/Port eines Proxys zum Senden von AutoSupport Telemetrie.

`--enable-node-prep`: Versuch, benötigte Pakete auf Knoten zu installieren.

`--generate-custom-yaml`: Generieren Sie YAML-Dateien ohne etwas zu installieren.

`-h, --help`: Hilfe zur Installation.

`--http-request-timeout`: Das HTTP-Anforderungs-Timeout für die REST-API des Trident-Controllers überschreiben (Standard 1m30s).

`--image-registry` string: Adresse/Port einer internen Image-Registry.

`--k8s-timeout` duration: Das Timeout für alle Kubernetes-Operationen (Standard 3m0s).

`--kubelet-dir` string: Der Host-Speicherort des internen Status von kubelet (Default "/var/lib/kubelet").

`--log-format` string: Das Protokollierungsformat Astra Trident (Text, json) (Standard "Text").

`--pv` string: Der Name des Legacy-PV, das Astra Trident verwendet, stellt sicher, dass es nicht existiert (Standard "Trident").

`--pvc` string: Der Name des von Astra Trident verwendeten Legacy-PVC, stellt sicher, dass dies nicht existiert (Standard "Trident").

`--silence-autosupport`: Senden Sie AutoSupport-Pakete nicht automatisch an NetApp (Standard TRUE).

`--silent`: Deaktivieren Sie die meisten Ausgaben während der Installation.

`--trident-image` string: Das zu installierende Astra Trident-Image.

`--use-custom-yaml`: Verwenden Sie alle vorhandenen YAML-Dateien, die im Setup-Verzeichnis vorhanden sind.

`--use-ipv6`: Verwenden Sie IPv6 für die Kommunikation von Astra Trident.

Protokolle

Verwenden Sie `logs` Flags, um die Protokolle aus Astra Trident zu drucken.

```
tridentctl logs [flags]
```

Flags

`-a, --archive`: Erstellen Sie ein Support-Archiv mit allen Protokollen, sofern nicht anders angegeben.

`-h, --help`: Hilfe für Protokolle.

`-l, --log` string: Astra Trident-Protokoll zur Anzeige. Eine von Trident/Trident-Operator/alle (Standard „Auto“).

`--node` string: Der Name des Kubernetes-Knotens, von dem aus die POD-Protokolle des Knotens erfasst werden.

`-p, --previous`: Holen Sie sich die Protokolle für die vorherige Container-Instanz, wenn sie existiert.

`--sidecars`: Holen Sie sich die Protokolle für die Beiwagen-Container.

Senden

Mit dem `send` Befehl senden Sie eine Ressource aus Astra Trident.

```
tridentctl send [option]
```

Optionen

`autosupport`: Senden Sie ein AutoSupport-Archiv an NetApp.

Deinstallieren

Verwenden Sie `uninstall` Flags, um Astra Trident zu deinstallieren.

```
tridentctl uninstall [flags]
```

Flags

- `-h, --help`: Hilfe zur Deinstallation.
- `--silent`: Deaktivieren Sie die meisten Ausgaben während der Deinstallation.

Aktualisierung

Verwenden Sie den `update` Befehl, um eine Ressource in Astra Trident zu ändern.

```
tridentctl update [option]
```

Optionen

`backend`: Aktualisieren Sie ein Backend in Astra Trident.

Back-End-Status aktualisieren

Verwenden Sie den `update backend state` Befehl, um die Back-End-Vorgänge anzuhalten oder fortzusetzen.

```
tridentctl update backend state <backend-name> [flag]
```

Zu berücksichtigende Aspekte

- Wenn ein Backend mit einem `TridentBackendConfig` (tbc) erstellt wird, kann das Backend nicht mit einer Datei aktualisiert werden `backend.json`.
- Wenn der `userState` in einem tbc gesetzt wurde, kann er nicht mit dem Befehl geändert werden `tridentctl update backend state <backend-name> --user-state suspended/normal`.
- Um die Möglichkeit, die Via `tridentctl` nach der Einstellung über tbc wieder einzustellen `userState`, muss das Feld aus dem tbc `userState` entfernt werden. Dies kann mit dem Befehl erfolgen `kubectl edit tbc`. Sobald das `userState` Feld entfernt wurde, können Sie mit dem `tridentctl update backend state` Befehl das eines Backends ändern `userState`.
- Verwenden Sie die `tridentctl update backend state`, um die zu ändern `userState`. Sie können auch die Using- oder -Datei aktualisieren `userState TridentBackendConfig backend.json`; dies löst eine vollständige Neuinitialisierung des Backends aus und kann zeitaufwändig sein.

Flags

- `-h, --help`: Hilfe für Backend-Status.
- `--user-state`: Auf Pause gesetzt `suspended`. Legen Sie fest `normal`, um die Back-End-Vorgänge fortzusetzen. Wenn eingestellt auf `suspended`:

- `AddVolume` Und `Import Volume` werden angehalten.
- `CloneVolume`, `ResizeVolume`, `PublishVolume`, `UnPublishVolume`, `CreateSnapshot`, `GetSnapshot` `RestoreSnapshot`, `DeleteSnapshot`, `RemoveVolume`, `GetVolumeExternal`, `ReconcileNodeAccess` verfügbar bleiben.

Sie können den Backend-Status auch über das Feld in der Backend-Konfigurationsdatei oder aktualisieren

`userState TridentBackendConfig backend.json`. Weitere Informationen finden Sie unter ["Optionen für das Management von Back-Ends"](#) und ["Führen Sie das Back-End-Management mit kubectl durch"](#).

Beispiel:

JSON

Führen Sie die folgenden Schritte aus, um die mit der Datei zu aktualisieren `userState backend.json` :

1. Bearbeiten Sie die `backend.json` Datei, um das Feld mit dem Wert „suspendiert“ aufzunehmen `userState`.
2. Aktualisieren Sie das Backend mit dem `tridentctl backend update` Befehl und dem Pfad zur aktualisierten `backend.json` Datei.

Beispiel: `tridentctl backend update -f /<path to backend JSON file>/backend.json`

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "<redacted>",
  "svm": "nas-svm",
  "backendName": "customBackend",
  "username": "<redacted>",
  "password": "<redacted>",
  "userState": "suspended",
}
```

YAML

Sie können den `tbc` bearbeiten, nachdem er angewendet wurde, indem Sie den Befehl verwenden `kubectl edit <tbc-name> -n <namespace>`. Im folgenden Beispiel wird der Back-End-Status mit der Option zum Anhalten aktualisiert `userState: suspended`:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  backendName: customBackend
  storageDriverName: ontap-nas
  managementLIF: <redacted>
  svm: nas-svm
userState: suspended
credentials:
  name: backend-tbc-ontap-nas-secret
```


Version

Verwenden Sie `version` Flags, um die Version von und den laufenden Trident-Dienst zu drucken `tridentctl`.

```
tridentctl version [flags]
```

Flags

- `--client`: Nur Client-Version (kein Server erforderlich).
- `-h`, `--help`: Hilfe zur Version.

Überwachen Sie Astra Trident

Astra Trident bietet eine Reihe von Prometheus Kennzahlen-Endpunkten, mit denen Sie die Performance von Astra Trident überwachen können.

Überblick

Mit den von Astra Trident bereitgestellten Metriken können Sie:

- Bleiben Sie auf dem Laufenden über den Zustand und die Konfiguration von Astra Trident. Sie können prüfen, wie erfolgreich Vorgänge sind und ob sie wie erwartet mit den Back-Ends kommunizieren können.
- Untersuchen Sie die Back-End-Nutzungsinformationen und erfahren Sie, wie viele Volumes auf einem Back-End bereitgestellt werden, sowie den belegten Speicherplatz usw.
- Erstellt eine Zuordnung der Anzahl von Volumes, die über verfügbare Back-Ends bereitgestellt werden.
- Verfolgen Sie die Leistung. Sie können sich ansehen, wie lange Astra Trident für die Kommunikation mit Back-Ends und die Durchführung von Vorgängen benötigt.



Standardmäßig sind die Trident-Kennzahlen auf dem Zielport am `/metrics` Endpunkt sichtbar 8001. Diese Metriken sind bei der Installation von Trident standardmäßig aktiviert.

Was Sie benötigen

- Kubernetes-Cluster mit installiertem Astra Trident
- Eine Prometheus Instanz. Dies kann ein sein "[Implementierung von Container-Prometheus](#)", oder Sie können wählen, Prometheus als ausführen "[Native Applikation](#)".

Schritt 1: Definieren Sie ein Prometheus-Ziel

Sie sollten ein Prometheus Ziel definieren, um die Kennzahlen zu sammeln und Informationen über das Management von Back-Ends Astra Trident, die von ihm erstellten Volumes usw. zu erhalten. Dies "[Blog](#)" erklärt, wie Sie Prometheus und Grafana mit Astra Trident verwenden können, um Metriken abzurufen. Im Blog erfahren Sie, wie Sie Prometheus als Betreiber in Ihrem Kubernetes-Cluster ausführen und einen ServiceMonitor erstellen können, um Astra Trident-Kennzahlen zu erhalten.

Schritt: Erstellen Sie einen Prometheus ServiceMonitor

Um die Trident-Kennzahlen zu nutzen, sollten Sie einen Prometheus ServiceMonitor erstellen, der den Service überwacht `trident-csi` und den Port abhört `metrics`. Ein Beispiel für ServiceMonitor sieht so aus:

```

apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - port: metrics
      interval: 15s

```

Diese ServiceMonitor-Definition ruft vom Dienst zurückgegebene Kennzahlen `trident-csi` ab und sucht gezielt nach dem `metrics` Endpunkt des Dienstes. Das Ergebnis: Prometheus ist jetzt so konfiguriert, dass sie die Kennzahlen von Astra Trident verstehen.

Zusätzlich zu den Kennzahlen, die direkt aus Astra Trident zur Verfügung stehen, legt Kubelet viele `kubelet_volume_*` Kennzahlen über seinen eigenen Endpunkt für Kennzahlen offen. Kubelet kann Informationen über verbundene Volumes bereitstellen und Pods und andere interne Vorgänge, die er übernimmt. Siehe "[Hier](#)".

Schritt 3: Abfrage der Trident-Kennzahlen mit PromQL

PromQL ist gut geeignet, um Ausdrücke zu erstellen, die Zeitreihen- oder tabellarische Daten zurückgeben.

Im Folgenden finden Sie einige PromQL-Abfragen, die Sie verwenden können:

Abrufen des Integritätsinformationen zu Trident

- **Prozentsatz der HTTP 2XX-Antworten von Astra Trident**

```

(sum (trident_rest_ops_seconds_total_count{status_code=~"2.."} OR on()
vector(0)) / sum (trident_rest_ops_seconds_total_count)) * 100

```

- **Prozentualer Anteil DER REST-Antworten von Astra Trident über Statuscode**

```

(sum (trident_rest_ops_seconds_total_count) by (status_code) / scalar
(sum (trident_rest_ops_seconds_total_count))) * 100

```

- **Durchschnittsdauer in ms der von Astra Trident durchgeführten Operationen**

```
sum by (operation)
(trident_operation_duration_milliseconds_sum{success="true"}) / sum by
(operation)
(trident_operation_duration_milliseconds_count{success="true"})
```

Holen Sie sich Informationen zur Nutzung von Astra Trident

- **Mittlere Volumengröße**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Gesamter Volume-Speicherplatz, der von jedem Backend bereitgestellt wird**

```
sum (trident_volume_allocated_bytes) by (backend_uuid)
```

Individuelle Volume-Nutzung



Dies ist nur aktiviert, wenn auch kubelet-Kennzahlen gesammelt werden.

- **Prozentsatz des verwendeten Speicherplatzes für jedes Volumen**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

AutoSupport Telemetrie von Astra Trident mit Thema

Standardmäßig sendet Astra Trident in einem täglichen Intervall Prometheus-Kennzahlen und grundlegende Backend-Informationen an NetApp.

- Um zu verhindern, dass Astra Trident Prometheus-Kennzahlen und grundlegende Backend-Informationen an NetApp sendet, übergeben Sie das `--silence-autosupport` Flag während der Astra Trident Installation.
- Astra Trident kann auch bei Bedarf Container-Logs an den NetApp-Support über senden `tridentctl send autosupport`. Sie müssen Astra Trident auslösen, um seine Protokolle hochzuladen. Bevor Sie Protokolle senden, sollten Sie NetApp's akzeptieren <https://www.netapp.com/company/legal/privacy-policy/>["datenschutzrichtlinie"].
- Sofern nicht angegeben, ruft Astra Trident die Protokolle der letzten 24 Stunden ab.
- Sie können den Zeitrahmen für die Protokollaufbewahrung mit dem Flag angeben `--since`. Zum Beispiel: `tridentctl send autosupport --since=1h`. Diese Informationen werden gesammelt und über einen Container gesendet `trident-autosupport`, der zusammen mit Astra Trident installiert wird. Sie können das Container-Bild unter abrufen ["Trident AutoSupport"](#).

- Trident AutoSupport erfasst oder übermittelt keine personenbezogenen Daten oder personenbezogenen Daten. Sie wird mit einem geliefert ["EULA"](#) , das sich nicht für das Trident Container-Image selbst eignet. Weitere Informationen zum Engagement von NetApp für Datensicherheit und Vertrauen finden ["Hier"](#) Sie hier.

Eine von Astra Trident gesendete Beispiellast sieht folgendermaßen aus:

```
---
items:
- backendUUID: ff3852e1-18a5-4df4-b2d3-f59f829627ed
  protocol: file
  config:
    version: 1
    storageDriverName: ontap-nas
    debug: false
    debugTraceFlags:
    disableDelete: false
    serialNumbers:
    - nwkvzfanek_SN
    limitVolumeSize: ''
  state: online
  online: true
```

- Die AutoSupport Meldungen werden an den AutoSupport Endpunkt von NetApp gesendet. Wenn Sie eine private Registrierung zum Speichern von Container-Images verwenden, können Sie das Flag verwenden `--image-registry`.
- Sie können auch Proxy-URLs konfigurieren, indem Sie die Installation YAML-Dateien erstellen. Dies kann getan werden, indem `tridentctl install --generate-custom-yaml` Sie die YAML-Dateien erstellen und das Argument für den `trident-autosupport` Container in `trident-deployment.yaml` hinzufügen `--proxy-url`.

Deaktivieren Sie Astra Trident Metriken

Um **die Meldung von**-Metriken zu deaktivieren, sollten Sie benutzerdefinierte YAMLs (mit dem Flag) generieren `--generate-custom-yaml` und diese bearbeiten, um das Flag für den `trident-main` Container zu entfernen `--metrics`.

Deinstallieren Sie Astra Trident

Sie sollten die gleiche Methode verwenden, um Astra Trident zu deinstallieren, die Sie zur Installation von Astra Trident verwendet haben.

Über diese Aufgabe

- Wenn Sie nach einem Upgrade, Abhängigkeitsproblemen oder einem nicht erfolgreichen oder unvollständigen Upgrade eine Korrektur für Fehler benötigen, sollten Sie Astra Trident deinstallieren und die frühere Version mithilfe der entsprechenden Anweisungen neu installieren ["Version"](#). Dies ist die einzige empfohlene Möglichkeit, *Downgrade* auf eine frühere Version zu übertragen.

- Für eine einfache Aktualisierung und Neuinstallation entfernt das Deinstallieren von Astra Trident nicht die CRDs oder damit verbundene Objekte, die von Astra Trident erstellt wurden. Wenn Sie Astra Trident und alle seine Daten vollständig entfernen müssen, lesen Sie ["Entfernen Sie Astra Trident und CRDs vollständig"](#).

Bevor Sie beginnen

Falls Sie Kubernetes-Cluster stilllegen, müssen Sie alle Applikationen löschen, die Volumes verwenden, die von Astra Trident erstellt wurden, bevor Sie sie deinstallieren. Dadurch wird sichergestellt, dass PVCs auf Kubernetes-Nodes nicht veröffentlicht werden, bevor sie gelöscht werden.

Bestimmen Sie die ursprüngliche Installationsmethode

Sie sollten die gleiche Methode verwenden, um Astra Trident zu deinstallieren, die Sie verwendet haben, um es zu installieren. Überprüfen Sie vor der Deinstallation, mit welcher Version Sie Astra Trident ursprünglich installiert haben.

1. Verwenden Sie, `kubectl get pods -n 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. Falls es einen Pod gibt, mit können Sie `kubectl describe tproc trident` feststellen, 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.

Deinstallieren Sie die Installation eines Trident-Operators

Sie können die Installation eines Dreizack-Bedieners manuell oder mithilfe von Helm deinstallieren.

Deinstallieren Sie die manuelle Installation

Falls Sie Astra Trident mit dem Operator installiert haben, können Sie es deinstallieren, indem Sie einen der folgenden Schritte ausführen:

1. **CR bearbeiten `TridentOrchestrator` und das Deinstallationsflag einstellen:**

```
kubectl patch torc <trident-orchestrator-name> --type=merge -p
'{"spec":{"uninstall":true}}'
```

Wenn das `uninstall` Flag auf gesetzt ist `true`, deinstalliert der Trident-Operator Trident, entfernt aber nicht den `TridentOrchestrator` selbst. Sie sollten den `TridentOrchestrator` aufräumen und einen neuen erstellen, wenn Sie Trident erneut installieren möchten.

2. **Löschen `TridentOrchestrator`:** Durch Entfernen des `TridentOrchestrator` CR, der zur Bereitstellung von Astra Trident verwendet wurde, weisen Sie den Bediener an, Trident zu deinstallieren. Der Operator verarbeitet die Entfernung von `TridentOrchestrator` Astra Trident Deployment und demonset und löscht die Trident-Pods, die er im Rahmen der Installation erstellt hatte.

```
kubectl delete -f deploy/<bundle.yaml> -n <namespace>
```

Deinstallieren Sie Helm-Installation

Wenn Sie Astra Trident mit Helm installiert haben, können Sie es mit `helm uninstall`.

```
#List the Helm release corresponding to the Astra Trident install.
helm ls -n trident
NAME                NAMESPACE      REVISION      UPDATED
STATUS              CHART           APP VERSION
trident             trident         1             2021-04-20
00:26:42.417764794 +0000 UTC deployed    trident-operator-21.07.1
21.07.1

#Uninstall Helm release to remove Trident
helm uninstall trident -n trident
release "trident" uninstalled
```

Deinstallieren Sie eine tridentctl Installation

Verwenden Sie den `uninstall` Befehl in `tridentctl`, um alle mit Astra Trident verbundenen Ressourcen mit Ausnahme der CRDs und zugehörigen Objekte zu entfernen:

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

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers 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, BEISPIELHAFTE 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.