Management und Monitoring von Astra Trident

Astra Trident

NetApp
January 14, 2026

This PDF was generated from https://docs.netapp.com/de-de/trident-2310/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 . . . 12
  - Überblick . . . 13
  - Schritt 1: Definieren Sie ein Prometheus-Ziel . . . 13
  - Schritt: Erstellen Sie einen Prometheus ServiceMonitor . . . 13
  - Schritt 3: Abfrage der Trident-Kennzahlen mit PromQL . . . 14
  - AutoSupport Telemetrie von Astra Trident mit Thema . . . 15
  - Deaktivieren Sie Astra Trident Metriken . . . 16
- Deinstallieren Sie Astra Trident . . . 16
  - Bestimmen Sie die ursprüngliche Installationsmethode . . . 17
  - Deinstallieren Sie die Installation eines Trident-Operators . . . 17
  - Deinstallieren Sie A `tridentctl` Installation . . . 18

Management und Monitoring von Astra Trident

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, 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 in erstellt haben "CVS-Diensttyp", Sie müssen die Backend-Konfiguration aktualisieren, um die zu verwenden `standardsw` Oder `zoneredundantstandardsw` Service-Level beim Upgrade von Astra Trident 23.01. Fehler beim Aktualisieren des `serviceLevel` Im Backend kann zu einem Ausfall von Volumes führen. Siehe "Beispiele für CVS-Diensttypen" Entsprechende Details.
- 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.templ` 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 "Bekannte 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 22.10 (oder einem beliebigen 22.10-Punkt-Release) auf 23.10 aktualisieren.
- Wenn Sie ein Upgrade von einer Version außerhalb des Fensters mit vier Releases durchführen, führen Sie ein Upgrade in mehreren Schritten durch. Befolgen Sie die Upgrade-Anweisungen für "Frühere Version" Sie führen ein Upgrade von auf die neueste Version durch, die für das Fenster mit vier Versionen geeignet ist. Wenn Sie beispielsweise 21.10 verwenden und ein Upgrade auf 23.10 durchführen möchten:

a. Erstes Upgrade von 21.10 auf 22.10.

b. Dann Upgrade von 22.10 auf 23.10.

> 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. Nutzung `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. 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

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". Astra Trident bietet zwei Optionen für ein Upgrade.

- "Upgrade über den Trident-Operator"

> Wir empfehlen Ihnen, dies zu überprüfen "Den Upgrade-Workflow für Bediener verstehen" Vor der Aktualisierung mit dem Bediener.

*

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-Bedieners" Die Installation und das Upgrade von Astra Trident erfolgt automatisch für Astra Trident und Kubernetes-Objekte, ohne vorhandene gemountete Volumes zu unterbrechen. So kann Astra Trident Upgrades ohne Ausfallzeiten oder auch ohne "*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 Trident-Bedienerinstallation

Sie können ein Upgrade von Astra Trident mit dem Trident Operator entweder manuell oder mit Helm durchführen. Sie können ein Upgrade von einer Trident Benutzerinstallation auf eine andere Trident Benutzerinstallation durchführen oder von einem durchführen `tridentctl` Installation auf eine Trident-Operatorversion. Prüfen "Wählen Sie eine Aktualisierungsmethode aus" Vor dem Upgrade einer Trident-Benutzerinstallation.

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.

Für ein Upgrade von Astra Trident, das mit dem Namespace-Scoped Operator (Versionen 20.07 bis 20.10) installiert wurde, verwenden Sie die Upgrade-Anweisungen für "Ihre installierte Version" Von Astra Trident zu erhalten.

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 oder früheren Versionen "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 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 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 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 YAML-Bundle-Datei für Ihre Umgebung, wo *<bundle.yaml>* ist
`bundle_pre_1_25.yaml` Oder `bundle_post_1_25.yaml` Basierend auf Ihrer Kubernetes-Version
Wenn Sie beispielsweise Astra Trident 23.10 installieren, führen Sie den folgenden Befehl aus:

```
kubectl create -f 23.10.0/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 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. Wenn Sie "Astra Trident mit Helm installiert", Können Sie verwenden `helm upgrade trident netapp-trident/trident-operator --version 23.10.0` Um das Upgrade in einem Schritt durchzuführen. Wenn Sie den Helm Repo nicht hinzugefügt haben oder ihn nicht zum Upgrade verwenden

können:

a. Die neueste Version von Astra Trident kann von heruntergeladen werden "Die Sektion *Assets* auf GitHub".

b. Verwenden Sie die `helm upgrade` Befehl wo `trident-operator-23.10.0.tgz` Gibt die Version an, auf die Sie ein Upgrade durchführen möchten.

```
helm upgrade <name> trident-operator-23.10.0.tgz
```

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

2. 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.

Upgrade von einem `tridentctl` Installation zum Trident-Operator

Sie können ein Upgrade auf die neueste Version des Trident-Operators von durchführen `tridentctl` Installation: Die vorhandenen Back-Ends und VES stehen automatisch zur Verfügung.

> Bevor Sie zwischen den Installationsmethoden wechseln, lesen Sie die Informationen "Wechseln zwischen den Installationsmethoden"

Schritte

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

```
# 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. 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:                v23.10.0
```

Upgrade mit tridentctl

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

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.

Managen Sie Astra Trident mit tridentctl

Der "Trident Installationspaket" Umfasst die `tridentctl` Befehlszeilen-Utility für einfachen Zugriff auf Astra Trident. 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 laufen `tridentctl help` Um eine Liste der verfügbaren Befehle für zu erhalten `tridentctl` Oder hängen Sie die an `--help` Markieren Sie einen beliebigen Befehl, um eine Liste mit Optionen und Flags für diesen bestimmten Befehl zu erhalten.

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

Astra Trident `tridentctl` Utility 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 die an `KUBECONFIG` Pfad zur Ausführung von Befehlen lokal oder von einem Kubernetes-Cluster zu einem anderen.

Alternativ können Sie den exportieren `KUBECONFIG` Variable Möglichkeit, auf ein bestimmtes Kubernetes-Cluster und Problem zu verweisen `tridentctl` Befehle zu diesem Cluster.

-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

Verwenden Sie die `create` Befehl zum Hinzufügen einer Ressource zum Astra Trident.

```
tridentctl create [option]
```

Optionen

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

Löschen

Verwenden Sie die `delete` Befehl, um eine oder mehrere Ressourcen aus Astra Trident zu entfernen.

```
tridentctl delete [option]
```

Optionen

`backend`: Löschen Sie ein oder mehrere Storage-Back-Ends von Astra Trident.
`snapshot`: Löschen Sie einen oder mehrere Volumen-Snapshots aus Astra Trident.

`storageclass`: Löschen einer oder mehrerer Speicherklassen von Astra Trident.
`volume`: Löschen Sie ein oder mehrere Storage Volumes von Astra Trident.

Get

Verwenden Sie die `get` Befehl: Sie erhalten eine oder mehrere Ressourcen von Astra Trident.

```
tridentctl get [option]
```

Optionen

`backend`: Holen Sie sich ein oder mehrere Storage Back-Ends von Astra Trident an.
`snapshot`: Holen Sie sich einen oder mehrere Schnappschüsse von Astra Trident.
`storageclass`: Holen Sie sich einen oder mehrere Storage-Kurse von Astra Trident.
`volume`: Holen Sie sich ein oder mehrere Bände von Astra Trident.

Flags

`-h`, `--help`: Hilfe für Volumen.
`--parentOfSubordinate string`: Abfrage auf untergeordnetes Quellvolumen begrenzen.
`--subordinateOf string`: Abfrage auf Untergebene beschränken.

Bilder

Nutzung `images` Flags zum Drucken einer Tabelle der Container-Images, die Astra Trident benötigt.

```
tridentctl images [flags]
```

Flags

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

Importvolumen

Verwenden Sie die `import volume` Befehl zum Importieren eines vorhandenen Volumes zu Astra Trident

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

Aliase

`volume`, `v`

Flags

`-f`, `--filename string`: Pfad zu YAML oder JSON PVC-Datei.
`-h`, `--help`: Hilfe für Lautstärke.
`--no-manage`: Nur PV/PVC erstellen. Nehmen Sie kein Lifecycle Management für Volumes an.

Installieren

Verwenden Sie die `install` Flags für die Installation von Astra Trident.

```
tridentctl install [flags]
```

Flags

`--autosupport-image string`: Das Container-Image für AutoSupport Telemetrie (Standard „netapp/Trident AutoSupport:<current-version>“).
`--autosupport-proxy string`: Die Adresse/der Port eines Proxy für den Versand von AutoSupport Telemetrie.
`--enable-node-prep`: Versuch, benötigte Pakete auf Knoten zu installieren.
`--generate-custom-yaml`: Erzeugen von YAML-Dateien ohne Installation von irgendetwas.
`-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`: Die Adresse/der Port einer internen Bilddatenbank.
`--k8s-timeout duration`: Die Zeitüberschreitung für alle Kubernetes-Operationen (Standard 3m0s).
`--kubelet-dir string`: Der Host-Standort des internen Status von kubelet (Standard "/var/lib/kubelet").
`--log-format string`: Das Astra Trident Logging-Format (Text, json) (Standard "Text").
`--pv string`: Der Name des alten PV, das von Astra Trident verwendet wird, stellt sicher, dass dies nicht existiert (Standard "Dreizack").
`--pvc string`: Der Name des alten PVC verwendet von Astra Trident, stellt sicher, dass dies nicht existiert (Standard "Dreizack").
`--silence-autosupport`: AutoSupport Bundles nicht automatisch an NetApp senden (standardmäßig wahr).
`--silent`: Während der Installation die meiste Leistung deaktivieren.
`--trident-image string`: Das zu installierende Astra Trident-Image.
`--use-custom-yaml`: Verwenden Sie alle bestehenden YAML-Dateien, die im Setup-Verzeichnis vorhanden sind.
`--use-ipv6`: Nutzen Sie IPv6 für die Kommunikation von Astra Trident.

Protokolle

Nutzung `logs` Flags zum Drucken der Protokolle von Astra Trident.

```
tridentctl logs [flags]
```

Flags

`-a, --archive`: Erstellen Sie ein Stützarchiv mit allen Protokollen, sofern nicht anders angegeben.
`-h, --help`: Hilfe für Protokolle.
`-l, --log string`: Astra Trident Log to Display. Einer der Dreizack-Automatik-Operator ganz (Standard „Auto“).
`--node string`: Der Kubernetes-Knotenname, aus dem Node-Pod-Protokolle erfasst werden.
`-p, --previous`: Holen Sie sich die Protokolle für die frühere Container-Instanz, wenn sie existiert.
`--sidecars`: Holen Sie sich die Protokolle für die Sidecar-Container.

Senden

Verwenden Sie die `send` Befehl zum Senden einer Ressource vom Astra Trident.

```
tridentctl send [option]
```

Optionen

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

Deinstallieren

Nutzung `uninstall` Flags zum Deinstallieren von Astra Trident.

```
tridentctl uninstall [flags]
```

Flags

`-h, --help`: Hilfe zur Deinstallation.
`--silent`: Deaktivieren der meisten Ausgabe während der Deinstallation.

Aktualisierung

Verwenden Sie die `update` Befehl zum Ändern einer Ressource in Astra Trident.

```
tridentctl update [option]
```

Optionen

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

Back-End-Status aktualisieren

Verwenden Sie die `update backend state` Befehl zum Anhalten oder Fortsetzen von Back-End-Vorgängen.

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

Flags

`-h, --help`: Hilfe für Backend-Status.
`--user-state`: Auf eingestellt `suspended` Um Back-End-Vorgänge anzuhalten. Auf einstellen `normal` Um die Back-End-Vorgänge wieder aufzunehmen. Wenn eingestellt auf `suspended`:

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

Version

Nutzung `version` Flags zum Drucken der Version von `tridentctl` Und den Running Trident Service.

```
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.

Die Metriken von Trident sind standardmäßig auf dem Ziel-Port offengelegt `8001` Am `/metrics` endpunkt: 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 Prometheus als ein 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. Das "Blog" Erläutert, wie Sie mithilfe von Prometheus und Grafana mit Astra Trident Kennzahlen abrufen können. 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 verwenden, sollten Sie ein Prometheus ServiceMonitor erstellen, das überwacht `trident-csi` Service und wartet auf den `metrics` Port: 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 zurückgegebene Kennzahlen ab `trident-csi` Service und insbesondere sucht nach dem `metrics` endpunkt des Dienstes: Daher ist Prometheus jetzt so konfiguriert, dass es die von Astra Trident verstehen wird
Kennzahlen.

Neben den direkt bei Astra Trident verfügbaren Kennzahlen gibt kubelet auch viele andere Lösungen auf `kubelet_volume_*` Kennzahlen über den Endpunkt der IT-eigenen Kennzahlen. 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 die Prometheus Kennzahlen und grundlegende Backend-Informationen an NetApp sendet, bestehen Sie am `--silence-autosupport` Fahne während der Installation von Astra Trident.
- Astra Trident kann auch Container-Protokolle per On-Demand an den NetApp Support senden `tridentctl send autosupport`. Sie müssen Astra Trident auslösen, um seine Protokolle hochzuladen. Bevor Sie Protokolle einreichen, sollten Sie die von NetApp akzeptieren "datenschutzrichtlinie".
- Sofern nicht angegeben, ruft Astra Trident die Protokolle der letzten 24 Stunden ab.

- Sie können den Zeitrahmen für die Protokollaufbewahrung mit festlegen `--since` Flagge. Beispiel: `tridentctl send autosupport --since=1h`. Diese Informationen werden über ein gesammelt und versendet `trident-autosupport` Container
  Das wird neben Astra Trident installiert. Sie können das Container-Image 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 zum Speichern von Container-Images eine private Registrierung verwenden, können Sie das verwenden `--image-registry` Flagge.
- Sie können auch Proxy-URLs konfigurieren, indem Sie die Installation YAML-Dateien erstellen. Dies kann mit erfolgen `tridentctl install --generate-custom-yaml` So erstellen Sie die YAML-Dateien und fügen die hinzu `--proxy-url` Argument für das `trident-autosupport` Container in `trident-deployment.yaml`.

Deaktivieren Sie Astra Trident Metriken

Um**-Metriken von der Meldung zu deaktivieren, sollten Sie benutzerdefinierte YAML generieren (mit dem `--generate-custom-yaml` Markieren) und bearbeiten, um die zu entfernen `--metrics` Flagge wird für das aufgerufen `trident-main`
Container:

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 eine Fehlerbehebung für Fehler benötigen, die nach einem Upgrade, Abhängigkeitsproblemen oder einem nicht erfolgreichen oder unvollständigen Upgrade beobachtet wurden, 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 zugehörigen Daten vollständig entfernen müssen, finden Sie weitere Informationen unter "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. Nutzung `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. 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.

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. **Bearbeiten `TridentOrchestrator` CR und stellen Sie das Deinstallationsflag** ein:

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

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

2. **Löschen `TridentOrchestrator`**: Durch Entfernen der `TridentOrchestrator` CR, das zur Implementierung von Astra Trident verwendet wurde, weisen Sie den Bediener an, Trident zu deinstallieren. Der Bediener verarbeitet die Entfernung von `TridentOrchestrator` Außerdem wird die Implementierung und das Dämonenset Astra Trident entfernt und die im Rahmen der Installation erstellten Trident-Pods gelöscht.

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

Deinstallieren Sie Helm-Installation

Wenn Sie Astra Trident mithilfe von Helm installiert haben, können Sie es mit deinstallieren `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 A `tridentctl` Installation

Verwenden Sie die `uninstall` Befehl in `tridentctl` So entfernen Sie alle Ressourcen, die Astra Trident mit Ausnahme der CRDs und zugehörigen Objekte zugeordnet sind:

```
./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 IRGENDEINER 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.