



Managen Sie Astra Trident

Astra Trident

NetApp
April 16, 2024

Inhalt

- Managen Sie Astra Trident 1
- Upgrade Astra Trident 1
- Deinstallieren Sie Astra Trident 14
- Downgrade Astra Trident 16

Managen Sie 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 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.

Wählen Sie eine Version aus

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 22.01 auf 23.01 aktualisieren (einschließlich aller Punktversionen, z. B. 22.01.1).
- 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.



- Für alle Upgrades von Version 19.04 und früher ist die Migration der Astra Trident Metadaten von alleine erforderlich `etcd` Zu CRD-Objekten. Prüfen Sie die Dokumentation der 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 von [security Kontexten](#) 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#Lstorage-class-bronze-default.yaml](https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml#Lstorage-class-bronze-default.yaml). Weitere Informationen finden Sie unter "[Bekannte Probleme](#)".

Wählen Sie eine Upgrade-Option aus

Astra Trident bietet zwei Optionen für ein Upgrade. Im Allgemeinen verwenden Sie die gleiche Option, die Sie für die Erstinstallation verwendet haben, wie Sie jedoch können "[Wechseln Sie zwischen den Installationsmethoden](#)".

- "Upgrade über den Trident-Operator"

*



CSI Volume Snapshots ist jetzt eine GA-Funktion, die ab Kubernetes 1.20 beginnt. Beim Upgrade von Astra Trident müssen alle vorherigen Alpha-Snapshot-CRS und CRDs (Volume Snapshot-Klassen, Volume-Snapshots und Volume Snapshot-Inhalt) entfernt werden, bevor das Upgrade durchgeführt wird. Siehe "[Diesem Blog](#)" Schritte zur Migration von Alpha-Snapshots in die Beta/GA-Spezifikation

Änderungen am Operator

In der Version 21.01 von Astra Trident werden dem Betreiber einige wichtige architektonische Änderungen vorgestellt:

- 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 `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

Sie können eine bestehende Astra Trident-Installation ganz einfach mithilfe des Betreibers aufrüsten.

Bevor Sie beginnen

Um mit dem Bediener zu aktualisieren, sollten die folgenden Bedingungen erfüllt sein:

- Sie benötigen eine CSI-basierte Astra Trident-Installation. Alle Versionen von 19.07 sind CSI-basiert. Sie können die Pods in Ihrem Trident Namespace überprüfen und überprüfen.
 - Die Benennung von Pods in Versionen vor 23.01 folgt `A trident-csi-*` übereinkommen.
 - 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.

- Wenn Sie CSI Trident deinstalliert haben und die Metadaten aus der Installation beibehalten werden, können Sie mithilfe des Operators ein Upgrade durchführen.
- Es sollte nur eine Astra Trident Installation über alle Namespaces in einem bestimmten Kubernetes Cluster hinweg vorhanden sein.
- Ein Kubernetes-Cluster sollte ausgeführt werden ["Eine unterstützte Kubernetes Version"](#).
- Wenn Alpha-Snapshot-CRDs vorhanden sind, sollten Sie sie mit entfernen `tridentctl obliviare alpha-snapshot-crd`. Dadurch werden die CRDs für die Alpha-Snapshot-Spezifikation gelöscht. Informationen zu vorhandenen Snapshots, die gelöscht/migriert werden sollen, finden Sie unter ["Diesem Blog"](#).



- Wenn Sie das Upgrade von Trident mithilfe des Betreibers für OpenShift Container Platform vornehmen, sollten Sie ein Upgrade auf Trident 21.01.1 oder höher durchführen. 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"](#).
- Verwenden Sie den Operator zum Aktualisieren von Trident nicht, wenn Sie ein verwenden etcd-Based Trident Release (19.04 oder früher).

Upgrade einer Installation des Trident Operators mit Cluster-Umfang

Führen Sie die folgenden Schritte aus, um eine Installation des Trident Operators mit Cluster-Umfang zu aktualisieren. Alle Astra Trident Versionen 21.01 und höher verwenden einen Operator mit Cluster-Umfang.

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 23.01 für Kubernetes 1.26 installieren, führen Sie den folgenden Befehl aus:

```
kubectl create -f 23.01.1/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 älter "[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

Führen Sie die folgenden Schritte aus, um von einer Instanz von Astra Trident zu aktualisieren, die mit dem Namespace-Scoped Operator (Versionen 20.07 bis 20.10) installiert wurde.

Schritte

1. Überprüfen Sie den Status der vorhandenen Trident Installation. Prüfen Sie dazu den **Status** von `TridentProvisioner`. Der Status sollte sein `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 [23.01.1]
mkdir 23.01.1
cd 23.01.1
wget
https://github.com/NetApp/trident/releases/download/v23.01.1/trident-
installer-23.01.1.tar.gz
tar -xf trident-installer-23.01.1.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. Um diesen Schritt abzuschließen, benötigen Sie die Paket-YAML-Datei, die zum Bereitstellen des Namespace-Scoped-Operators von 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.



Sie sollten die erforderlichen Änderungen an den Trident Installationsparametern vornehmen (z. B. Ändern der 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"](#).

```
#Ensure you are in the right directory
pwd
/root/20.10.1/trident-installer

#Delete the 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

#Confirm the Trident operator was removed
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    PORT(S)
AGE
service/trident-csi                 ClusterIP     10.108.174.125  <none>
34571/TCP,9220/TCP                 105d

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

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

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

In dieser Phase, der `trident-operator-xxxxxxxxxxx-xxxxxx` Pod wurde gelöscht.

4. (Optional) Wenn die Installationsparameter geändert werden müssen, aktualisieren Sie den `TridentProvisioner` Spez. Dies können Änderungen sein, wie z. B. das Ändern der privaten Image-Registry zum Ziehen von Container-Images, das Aktivieren von Debug-Protokollen oder das Festlegen von Image Pull Secrets.

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

5. Installieren Sie den Trident Operator.



Durch die Installation des Operators Cluster-Scoped wird die Migration von initiiert `TridentProvisioner` Objekte an `TridentOrchestrator` Objekte, löscht `TridentProvisioner` Objekte und das `tridentprovisioner` CRD, und aktualisiert Astra Trident auf die Version des verwendeten Cluster-Scoped-Betreibers. Im folgenden Beispiel wurde für Trident ein Upgrade auf 23.01.1 durchgeführt.



Ein Upgrade des Astra Trident mithilfe des Trident Operators führt zu einer Migration von `tridentProvisioner` Zu A `tridentOrchestrator` Objekt mit dem gleichen Namen. Dieser Vorgang wird automatisch vom Betreiber übernommen. Auch Astra Trident ist auf dem Upgrade im selben Namespace wie zuvor installiert.


```

#Ensure you are in the correct directory
pwd
/root/23.01.1/trident-installer

#Install the cluster-scoped operator in the **same namespace**
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

#Examine Trident pods in the namespace
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

#Confirm Trident has been updated to the desired version
kubectl describe torc trident | grep Message -A 3
Message:          Trident installed
Namespace:       trident
Status:          Installed
Version:         v23.01.1

```



Der trident-controller Zudem geben die Pod-Namen die im Jahr 23.01 eingeführte Namenskonvention wieder.

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-23.01.1.tgz` Gibt die Version an, auf die Sie ein Upgrade durchführen möchten.

```
helm upgrade <name> trident-operator-23.01.1.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-23.01.1-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 [23.01.1]
mkdir 23.01.1
cd 23.01.1
wget
https://github.com/NetApp/trident/releases/download/v22.01.1/trident-
installer-23.01.1.tar.gz
tar -xf trident-installer-23.01.1.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 Bediener bereit.

```
#Install the cluster-scoped operator in the same namespace
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.

```

#Create a tridentOrchestrator to initiate a Trident install
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

#Confirm Trident was upgraded to the desired version
kubectl describe torc trident | grep Message -A 3
Message:                             Trident installed
Namespace:                            trident
Status:                                Installed
Version:                               v23.01.1

```

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`.

Überlegungen vor dem Upgrade

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

- 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.
- Die Beta-Version von Volume Snapshots führt einen geänderten Satz von CRDs und einen Snapshot-Controller ein, die beide vor der Installation von Astra Trident eingerichtet werden sollten. ["Diesem Blog"](#) Erläutert die Schritte, die bei der Migration von Alpha-Volume-Snapshots in das Beta-Format erforderlich sind.
- 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.



Unterbrechen Sie beim Upgrade von Astra Trident nicht den Upgrade-Prozess. Vergewissern Sie sich, dass das Installationsprogramm ausgeführt wird.

Nächste Schritte nach dem Upgrade

Um die umfangreichen Funktionen neuerer Trident Versionen (wie On-Demand Volume Snapshots) nutzen zu können, können Sie die Volumes mit der aktualisieren `tridentctl upgrade` Befehl.

Wenn es Legacy-Volumes gibt, sollten Sie diese von einem NFS/iSCSI-Typ auf den CSI-Typ upgraden, um die umfassenden neuen Funktionen in Astra Trident zu nutzen. Ein von Trident bereitgestelltes Legacy-PV unterstützt die herkömmlichen Funktionen.

Beachten Sie bei der Entscheidung, Volumes auf den CSI-Typ zu aktualisieren:

- 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.

Beispiel für ein Volume-Upgrade

Dies ist ein Beispiel, das die Durchführung eines Volume-Upgrades zeigt.

1. Laufen `kubect1 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   19h
Bound   default/pvc-1                        standard
default-pvc-2-a8486                        1073741824   RWO          Delete   19h
Bound   default/pvc-2                        standard
default-pvc-3-a849e                        1073741824   RWO          Delete   19h
Bound   default/pvc-3                        standard
default-pvc-4-a84de                        1073741824   RWO          Delete   19h
Bound   default/pvc-4                        standard
trident                                     2Gi         RWO          Retain   19h
Bound   trident/trident
```

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

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

```
kubectl 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 erstellten `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.

```

kubect1 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>

```

Auf diese Weise können Sie Volumes des von Astra Trident erstellten NFS-/iSCSI-Typs auf Basis der einzelnen Volumes auf CSI-Typ aufrüsten.

Deinstallieren Sie Astra Trident

Je nachdem, wie Astra Trident installiert ist, gibt es mehrere Optionen, um es zu deinstallieren.

Deinstallieren Sie mit Helm

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 die Deinstallation mit dem Trident-Operator

Wenn Sie Astra Trident über den Operator installiert haben, können Sie es deinstallieren, indem Sie einen der folgenden Schritte durchführen:

- **Bearbeiten TridentOrchestrator So legen Sie die Deinstallationsflag fest:** können Sie bearbeiten TridentOrchestrator Und gesetzt `spec.uninstall=true`. Bearbeiten Sie das TridentOrchestrator CR und stellen Sie den ein `uninstall` Flag wie unten gezeigt:

```
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.

- **Löschen TridentOrchestrator:** durch Entfernen des `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. Um den Astra Trident komplett zu entfernen (einschließlich der von ihm erstellten CRDs) und effektiv den Schiefer sauber zu löschen, können Sie bearbeiten `TridentOrchestrator` Um die zu bestehen `wipeout` Option. Das folgende Beispiel zeigt:

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

Damit wird Astra Trident vollständig deinstalliert und alle Metadaten gelöscht, die mit den gemanagten Back-Ends und Volumes zusammenhängen. Nachfolgende Installationen werden als frische Installationen behandelt.



Sie sollten nur erwägen, die CRDs zu löschen, wenn Sie eine vollständige Deinstallation durchführen. Dieser Vorgang kann nicht rückgängig gemacht werden. **Wischen Sie die CRDs nicht ab, es sei denn, Sie möchten von vorne beginnen und eine neue Astra Trident Installation erstellen.**

Deinstallieren Sie mit `tridentctl`

Führen Sie die aus `uninstall` Befehl in `tridentctl` Wie folgt entfernt alle mit Astra Trident verbundenen Ressourcen außer den CRDs und verwandten Objekten, so dass es einfach ist, das Installationsprogramm erneut auszuführen, um auf eine neuere Version zu aktualisieren.

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

Um Astra Trident vollständig zu entfernen, sollten Sie die Finalizer für die von Astra Trident erstellten CRDs entfernen und die CRDs löschen.

Downgrade Astra Trident

Erfahren Sie mehr über die Schritte beim Downgrade auf eine frühere Version von Astra Trident.

Wenn sie heruntergestuft werden müssen

Sie können aus verschiedenen Gründen ein Downgrade in Betracht ziehen, beispielsweise aus folgenden Gründen:

- Verfügbarkeitsplanung
- Sofortige Behebung von Fehlern, die als Folge eines Upgrades beobachtet wurden
- Abhängigkeitsprobleme, nicht erfolgreiche und unvollständige Upgrades

Beim Wechsel zu einer Astra Trident-Version mit CRDs sollten Sie ein Downgrade in Betracht ziehen. Astra Trident verwendet CRDs zur Statuswahrung. Daher verfügen alle erstellten Storage-Einheiten (Back-Ends, Storage-Klassen, PV und Volume Snapshots) über zugehörige CRD-Objekte anstatt in die geschriebene Daten `trident PV` (verwendet von der früheren installierten Version von Astra Trident). Neu erstellte PVS, Back-Ends und Storage-Klassen werden als CRD-Objekte verwaltet.

Nur eine Downgrade für eine Version von Astra Trident versuchen, die mit CRDs (19.07 und höher) ausgeführt wird. Dadurch wird sichergestellt, dass die im aktuellen Astra Trident Release ausgeführten Vorgänge nach dem Downgrade sichtbar sind.

Wenn Sie nicht herunterstufen

Sie sollten kein Downgrade auf eine Version von Trident durchführen, die verwendet wird `etcd` Zustand beibehalten (19.04 und früher). Alle Operationen, die mit der aktuellen Astra Trident Version ausgeführt werden, werden nach der Herabstufung nicht berücksichtigt. Neu erstellte PVS können nicht verwendet werden, wenn Sie zurück auf eine frühere Version wechseln. Änderungen an Objekten wie Back-Ends, PVS, Storage-Klassen und Volume Snapshots (erstellt/aktualisiert/gelöscht) sind für Astra Trident nicht sichtbar, wenn sie zurück auf eine frühere Version verschoben werden. Wenn Sie zu einer früheren Version zurückkehren, wird der Zugriff auf PVS, die bereits mit der älteren Version erstellt wurden, nicht unterbrochen,

es sei denn, sie wurden aktualisiert.

Downgrade bei der Installation von Astra Trident mit dem Operator

Bei Installationen, die mit dem Trident Operator abgeschlossen wurden, ist das Downgrade-Verfahren anders und erfordert nicht die Verwendung von `tridentctl`.

Bei Installationen, die mit dem Trident-Operator durchgeführt wurden, kann Astra Trident auf eine der folgenden Werte heruntergestuft werden:

- Eine Version, die mithilfe des Namespace-Scoped-Operators installiert wird (20.07 - 20.10).
- Eine Version, die mit dem Cluster-Scoped Operator (21.01 und höher) installiert wird.

Downgrade auf einen Operator mit Cluster-Umfang

Führen Sie zum Downgrade von Astra Trident auf eine Version mit dem Operator Cluster-Scoped die unten aufgeführten Schritte aus.

Schritte

1. **"Deinstallieren Sie Astra Trident". Löschen Sie die CRDs nicht, es sei denn, Sie möchten eine vorhandene Installation vollständig entfernen.**
2. Der Trident-Operator kann mithilfe des Operator-Manifests gelöscht werden, das mit Ihrer Trident-Version verknüpft ist. Beispiel: <https://github.com/NetApp/trident/tree/stable/vXX.XX> /`deploy/bundle.yaml` Wo `vXX.XX` Ist die Versionsnummer (zum Beispiel `v22.10`) Und `bundle.yaml` Ist der Name der Paket-YAML-Datei.
3. Fahren Sie mit der Installation der gewünschten Version von Astra Trident fort. Befolgen Sie die Dokumentation für das gewünschte Release.

Downgrade auf Operator mit Namespace-Scoped

In diesem Abschnitt werden die Schritte für die Herabstufung auf eine Astra Trident-Version zusammengefasst, die im Bereich 20.07 bis 20.10 liegt, die über den Operator Namespace-Scoped installiert werden.

Schritte

1. **"Deinstallieren Sie Astra Trident". Die CRDs dürfen nicht gelöscht werden, es sei denn, Sie möchten eine vorhandene Installation vollständig entfernen.** stellen Sie sicher, dass die `tridentorchestrator` Wird gelöscht.

```
#Check to see if there are any tridentorchestrators present
kubectl get torc
NAME          AGE
trident      20h

#Looks like there is a tridentorchestrator that needs deleting
kubectl delete torc trident
tridentorchestrator.trident.netapp.io "trident" deleted
```

2. Der Trident-Operator kann mithilfe des Operator-Manifests gelöscht werden, das mit Ihrer Trident-Version verknüpft ist. Beispiel: <https://github.com/NetApp/trident/tree/stable/vXX.XX>

`/deploy/bundle.yaml` Wo `vXX.XX` Ist die Versionsnummer (zum Beispiel `v22.10`) Und `bundle.yaml` Ist der Name der Paket-YAML-Datei.

3. Löschen Sie die `tridentorchestrator` CRD.-

```
#Check to see if ``tridentorchestrators.trident.netapp.io`` CRD is present and delete it.
```

```
kubectl get crd tridentorchestrators.trident.netapp.io
```

```
NAME                                CREATED AT
tridentorchestrators.trident.netapp.io  2021-01-21T21:11:37Z
```

```
kubectl delete crd tridentorchestrators.trident.netapp.io
```

```
customresourcedefinition.apiextensions.k8s.io
"tridentorchestrators.trident.netapp.io" deleted
```

Astra Trident wurde deinstalliert.

4. Setzen Sie den Downgrade fort, indem Sie die gewünschte Version installieren. Befolgen Sie die Dokumentation für das gewünschte Release.

Downgrade mit Helm

Verwenden Sie zum Downgrade den `helm rollback` Befehl. Das folgende Beispiel zeigt:

```
helm rollback trident [revision #]
```

Downgrade bei Installation von Astra Trident mit `tridentctl`

Falls Sie Astra Trident mit installiert haben `tridentctl`, Der Downgrade-Prozess umfasst die folgenden Schritte. In dieser Folge werden Sie durch die Herabstufung von Astra Trident 21.07 auf 20.07 geleitet.



Bevor Sie mit dem Downgrade beginnen, sollten Sie einen Schnappschuss Ihres Kubernetes-Clusters machen `etcd`. So können Sie den aktuellen Status der CRDs von Astra Trident sichern.

Schritte

1. Stellen Sie sicher, dass Trident mit installiert wird `tridentctl`. Wenn Sie sich nicht sicher sind, wie Astra Trident installiert ist, führen Sie diesen einfachen Test aus:
 - a. Listen Sie die im Trident Namespace vorhandenen Pods auf.
 - b. Ermitteln Sie die Version des Astra Trident in Ihrem Cluster. Sie können entweder verwenden `tridentctl` Oder werfen Sie einen Blick auf das Image, das in den Trident Pods verwendet wird.
 - c. Wenn Sie * nicht sehen* a `tridentOrchestrator`, (Oder) a `tridentprovisioner`, (Oder) ein POD mit dem Namen `trident-operator-xxxxxxxxxx-xxxxx`, Astra Trident **ist installiert** mit

```
tridentctl.
```

2. Deinstallieren Sie Astra Trident mit den vorhandenen `tridentctl` Binär: In diesem Fall werden Sie mit dem 21.07 Binary deinstallieren.

```
tridentctl version -n trident
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 21.07.0        | 21.07.0        |
+-----+-----+

tridentctl uninstall -n trident
INFO Deleted Trident deployment.
INFO Deleted Trident daemonset.
INFO Deleted Trident service.
INFO Deleted Trident secret.
INFO Deleted cluster role binding.
INFO Deleted cluster role.
INFO Deleted service account.
INFO Deleted pod security policy.
podSecurityPolicy=tridentpods
INFO The uninstaller did not delete Trident's namespace in case it is
going to be reused.
INFO Trident uninstallation succeeded.
```

3. Nachdem diese abgeschlossen ist, holen Sie sich die Trident-Binärdatei für die gewünschte Version (in diesem Beispiel, 20.07), und installieren Sie Astra Trident. Sie können benutzerdefinierte YAML für ein generieren "[Benutzerdefinierte Installation](#)" Wenn nötig.

```
cd 20.07/trident-installer/
./tridentctl install -n trident-ns
INFO Created installer service account.
serviceaccount=trident-installer
INFO Created installer cluster role.                clusterrole=trident-
installer
INFO Created installer cluster role binding.
clusterrolebinding=trident-installer
INFO Created installer configmap.                  configmap=trident-
installer
...
...
INFO Deleted installer cluster role binding.
INFO Deleted installer cluster role.
INFO Deleted installer service account.
```

Der Downgrade-Vorgang ist abgeschlossen.

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtinhabers 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.