Management und Monitoring von Trident

Trident

NetApp
January 14, 2026

Inhalt

Management und Monitoring von Trident

Upgrade von Trident

Upgrade von Trident

Ab Version 24.02 folgt 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, mindestens einmal pro Jahr ein Upgrade durchzuführen, um von den neuen Funktionen von Trident zu profitieren.

Überlegungen vor dem Upgrade

Beachten Sie beim Upgrade auf die neueste Version von Trident Folgendes:

- In allen Namespaces in einem Kubernetes-Cluster sollte nur eine Trident Instanz installiert werden.
- Trident 23.07 und höher erfordert v1-Volumen-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 den oder `zoneredundantstandardsw` Service-Level beim Upgrade von Trident 23.01 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" .
- Beim Upgrade ist es wichtig, dass Sie `StorageClasses` von 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.templ` 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`].
  - Weitere Informationen finden Sie unter "Bekannte Probleme".

Schritt 1: Wählen Sie eine Version

Trident-Versionen folgen einer datumbasierten Namenskonvention `YY.MM`, wobei „YY“ die letzten beiden Ziffern des Jahres und „MM“ der Monat ist. 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 24.06 (oder einer beliebigen 24.06-Dot-Version) auf 25.02 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 23.01 verwenden und ein Upgrade auf 25.02 durchführen möchten:

a. Erstes Upgrade von 23.01 auf 24.02.

b. Dann Upgrade von 24.02 auf 25.02.

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 bestimmen Sie, welche Version Sie ursprünglich für die Installation von Trident verwendet haben:

1. Verwenden Sie, `kubectl get pods -n trident` um die Pods zu untersuchen.
   - Wenn kein Operator Pod vorhanden ist, wurde Trident mit installiert `tridentctl`.
   - Wenn es einen Operator-Pod gibt, wurde Trident entweder manuell oder über Helm mit dem Trident-Operator installiert.
2. Wenn ein Benutzer-POD vorhanden ist, verwenden Sie `kubectl describe torc`, um zu ermitteln, ob Trident mit Helm installiert wurde.
   - Wenn ein Helm-Label vorhanden ist, wurde Trident mit Helm installiert.
   - Wenn kein Helm-Etikett vorhanden ist, wurde Trident manuell mit dem 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". Es gibt zwei Optionen für ein Upgrade von Trident.

- "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 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

Eine der vielen "Vorteile der Verwendung des Trident-Bedieners"Installationen und Upgrades von Trident ist die automatische Handhabung von Trident- und Kubernetes-Objekten ohne Unterbrechung vorhandener gemountete Volumes. So kann Trident Upgrades ohne Ausfallzeiten oder"*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 Trident-Architektur auf dem Kubernetes-Cluster finden Sie unter "Architektur von 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 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.

Aktualisieren Sie eine Trident-Installation mit Trident Operator oder Helm

Sie können Trident mit dem Trident-Operator entweder manuell oder mit Helm aktualisieren. 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 Trident-Versionen 21.01 und höher verwenden einen Clusteroperator.

Um ein Upgrade von 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" von 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 Ihre Trident-Version:

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

2. Löschen Sie den Trident-Operator, der zur Installation der aktuellen 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 Trident mit der richtigen YAML-Bundle-Datei für Ihre Umgebung, wobei *<bundle.yaml>* `bundle_pre_1_25.yaml` `bundle_post_1_25.yaml` auf Ihrer Kubernetes-Version basiert. Wenn Sie beispielsweise Trident 25.02 installieren, führen Sie den folgenden Befehl aus:

```
kubectl create -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

Aktualisieren einer Helm-Installation

Sie können eine Trident Helm-Installation aktualisieren.

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

Wenn Sie Ihr Kubernetes-Cluster bereits von 1.24 auf 1.25 aktualisiert haben, ohne das Trident Helm zu aktualisieren, schlägt das Helm Upgrade fehl. Führen Sie die folgenden Schritte aus, damit das Ruder-Upgrade durchgeführt wird:

1. Installieren Sie das Helm-mapkubeapis Plugin von https://github.com/helm/helm-mapkubeapis.

2. Führen Sie einen Probelauf für die Trident-Version im Namespace durch, in dem Trident installiert ist. Hier werden die Ressourcen aufgelistet, die bereinigt werden.

```
helm mapkubeapis --dry-run trident  --namespace trident
```

3. Führen Sie einen vollständigen Durchlauf mit Ruder durch, um die Bereinigung durchzuführen.

```
helm mapkubeapis trident  --namespace trident
```

Schritte

1. Wenn Sie "Trident mit Helm installiert", können Sie verwenden `helm upgrade trident netapp-trident/trident-operator --version 100.2502.0` , 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 Trident-Version von "Die Sektion *Assets* auf GitHub"herunter.
   b. Verwenden Sie den `helm upgrade` Befehl where zeigt die Version an `trident-operator-25.02.0.tgz` , auf die Sie aktualisieren möchten.

```
helm upgrade <name> trident-operator-25.02.0.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 Trident Version herunter.

```
# Download the release required [25.02.0]
mkdir 25.02.0
cd 25.02.0
wget
https://github.com/NetApp/trident/releases/download/v25.02.0/trident-
installer-25.02.0.tar.gz
tar -xf trident-installer-25.02.0.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 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:                v25.02.0
```

Upgrade mit tridentctl

Sie können eine vorhandene Trident-Installation ganz einfach mit aktualisieren `tridentctl`.

Über diese Aufgabe

Die Deinstallation und Neuinstallation von Trident dient als Upgrade. Wenn Sie Trident deinstallieren, werden die Persistent Volume Claim (PVC) und das Persistent Volume (PV), die von der Trident-Bereitstellung verwendet werden, nicht gelöscht. Bereits bereitgestellte PVS bleiben verfügbar, während Trident offline ist, und Trident stellt Volumes für alle PVCs bereit, die in der Zwischenzeit erstellt werden, nachdem 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 Trident verbundenen Ressourcen mit Ausnahme der CRDs und zugehörigen Objekte zu entfernen.

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

2. Installieren Sie Trident neu. Siehe "Installieren Sie Trident mit tridentctl".

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

Managen Sie Trident mit tridentctl

Im ist das "Trident Installationspaket" Befehlszeilendienstprogramm für den einfachen Zugriff auf Trident enthalten `tridentctl`. Kubernetes-Benutzer mit genügend Privileges können es verwenden, um Trident zu installieren oder den Namespace zu managen, der den 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 Trident `tridentctl` unterstützt die folgenden Befehle und Global Flags.

Befehle

`create`
Fügen Sie eine Ressource zu Trident hinzu.

`delete`
Entfernen Sie eine oder mehrere Ressourcen aus Trident.

`get`
Holen Sie sich eine oder mehrere Ressourcen von Trident.

`help`
Hilfe zu jedem Befehl.

`images`
Drucken Sie eine Tabelle der Container-Bilder, die Trident benötigt.

`import`
Importieren Sie eine vorhandene Ressource in Trident.

`install`
Installation Von Trident:

`logs`
Drucken Sie die Protokolle aus Trident.

`send`
Senden Sie eine Ressource von Trident.

`uninstall`
Deinstallieren Sie Trident.

`update`
Ändern Sie eine Ressource in Trident.

`update backend state`
Vorübergehende Unterbrechung der Back-End-Vorgänge.

`upgrade`
Aktualisieren Sie eine Ressource in Trident.

`version`
Drucken Sie die Version von 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 der Trident-Implementierung:

-o, --output string
Ausgabeformat. Einer von json yaml-Namen natürlich Ärmellos (Standard).

-s, --server string
Adresse/Port der 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 den `create` Befehl, um Trident eine Ressource hinzuzufügen.

```
tridentctl create [option]
```

Optionen

`backend`: Fügen Sie ein Backend zu Trident.

Löschen

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

```
tridentctl delete [option]
```

Optionen

`backend`: Löschen Sie eine oder mehrere Speicher-Backends aus Trident.
`snapshot`: Löschen Sie einen oder mehrere Volume-Snapshots aus Trident.

`storageclass`: Löschen Sie eine oder mehrere Speicherklassen aus Trident.
`volume`: Löschen eines oder mehrerer Speichervolumes aus Trident.

Get

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

```
tridentctl get [option]
```

Optionen

`backend`: Holen Sie sich ein oder mehrere Speicher-Backends von Trident.
`snapshot`: Holen Sie sich einen oder mehrere Schnappschüsse von Trident.
`storageclass`: Holen Sie sich eine oder mehrere Speicherklassen von Trident.
`volume`: Holen Sie sich einen oder mehrere Bände von 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` Markierungen, um eine Tabelle der Container-Bilder zu drucken, die 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 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 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 Trident-Logging-Format (Text, json) (Standard "Text").
`--node-prep`: Ermöglicht Trident, die Knoten des Kubernetes-Clusters vorzubereiten, um Volumes mit dem angegebenen Datenspeicherprotokoll zu verwalten. **Derzeit `iscsi` wird nur der Wert unterstützt.**
`--pv string`: der Name des von Trident verwendeten Legacy-PV stellt sicher, dass dies nicht existiert (Standard "Trident").
`--pvc string`: Der Name des von 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 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 Trident.

Protokolle

Verwenden Sie `logs` Markierungen, um die Protokolle aus 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`: 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

Verwenden Sie den `send` Befehl, um eine Ressource von Trident zu senden.

```
tridentctl send [option]
```

Optionen

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

Deinstallieren

Verwenden Sie `uninstall` Flags, um 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 Trident zu ändern.

```
tridentctl update [option]
```

Optionen

`backend`: Aktualisieren Sie ein Backend in 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, das via tridentctl nach dem Setzen über tbc wieder einzustellen `userState`, muss das Feld aus dem tbc `userState` entfernt werden. Dies kann mit dem Befehl erfolgen `kubectl edit tbc` . Nachdem 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.

Plug-in-Unterstützung

Tridentctl unterstützt Plugins ähnlich wie kubectl. Tridentctl erkennt ein Plugin, wenn der binäre Dateiname des Plugins dem Schema "tridentctl-<plugin>" folgt, und die Binärdatei befindet sich in einem Ordner, der die Umgebungsvariable PATH aufführt. Alle erkannten Plugins sind im Plugin-Abschnitt der tridentctl-Hilfe aufgeführt. Optional können Sie die Suche auch einschränken, indem Sie in der Enviorment-Variable TRIDENTCTL_PLUGIN_PATH einen PLUGIN-Ordner angeben (Beispiel: `TRIDENTCTL_PLUGIN_PATH=~/tridentctl-plugins/`). Wenn die Variable verwendet wird, sucht tridenctl nur im angegebenen Ordner.

Monitoring von Trident

Trident bietet eine Reihe von Prometheus Kennzahlen-Endpunkten zur Überwachung der Trident-Performance.

Überblick

Mit den von Trident bereitgestellten Metriken können Sie Folgendes tun:

- Überwachen Sie den Zustand und die Konfiguration von 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. Hier sehen Sie, wie lange Trident benötigt, um mit Back-Ends zu kommunizieren und Vorgänge auszuführen.

> 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

- Ein Kubernetes-Cluster mit installiertem 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 erfassen und Informationen über die von

Trident gemanagten Back-Ends, die erstellten Volumes usw. zu erhalten. Dies "Blog" erklärt, wie Sie Prometheus und Grafana mit Trident verwenden können, um Metriken abzurufen. Im Blog erfahren Sie, wie Sie Prometheus als Betreiber in Ihrem Kubernetes-Cluster ausführen und einen Service Monitor erstellen können, um 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. Daher ist Prometheus jetzt so konfiguriert, dass es die Kennzahlen von Trident versteht.

Zusätzlich zu den direkt aus Trident verfügbaren Kennzahlen legt Kubelet viele `kubelet_volume_*` Metriken über seinen eigenen Endpunkt für Kennzahlen dar. 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 Trident**

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

- **Prozentsatz der REST-Antworten von Trident über Statuscode**

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

- **Durchschnittliche Dauer in ms der von 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 Trident-Nutzungsinformationen

- **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
```

Informationen zur Trident AutoSupport Telemetrie

Standardmäßig sendet Trident im täglichen Rhythmus Prometheus Kennzahlen und grundlegende Back-End-Informationen an NetApp.

- Um zu verhindern, dass Trident Prometheus-Metriken und grundlegende Backend-Informationen an NetApp sendet, übergeben Sie das `--silence-autosupport` Flag während der Trident-Installation.

- Trident kann auch Container-Protokolle an NetApp-Support On-Demand senden über `tridentctl send autosupport`. Sie müssen Trident auslösen, um die Protokolle hochzuladen. Bevor Sie Protokolle senden, sollten Sie NetApp's akzeptierenhttps://www.netapp.com/company/legal/privacy-policy/["datenschutzrichtlinie"^].
- Sofern nicht angegeben, ruft 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 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.

Ein Beispiel für eine Nutzlast, die von Trident gesendet wird, sieht wie folgt aus:

```
---
items:
  - backendUUID: ff3852e1-18a5-4df4-b2d3-f59f829627ed
    protocol: file
    config:
      version: 1
      storageDriverName: ontap-nas
      debug: false
      debugTraceFlags: null
      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 Trident-Kennzahlen

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 Trident

Sie sollten dieselbe Methode verwenden, um Trident zu deinstallieren, die Sie bei der Installation von 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 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 die Deinstallation von Trident nicht die von Trident erstellten CRDs oder zugehörigen Objekte. Wenn Sie vollständig entfernen müssen Trident und alle seine Daten, siehe "Entfernen Sie Trident und CRDs vollständig".

Bevor Sie beginnen

Wenn Sie Kubernetes-Cluster ausmustern, müssen Sie alle Applikationen löschen, die Volumes verwenden, die von 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 dieselbe Methode verwenden, um Trident zu deinstallieren, die Sie bei der Installation verwendet haben. Überprüfen Sie vor der Deinstallation, welche Version Sie ursprünglich für die Installation von Trident verwendet haben.

1. Verwenden Sie, `kubectl get pods -n trident` um die Pods zu untersuchen.
   - Wenn kein Operator Pod vorhanden ist, wurde Trident mit installiert `tridentctl`.
   - Wenn es einen Operator-Pod gibt, wurde 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 zu ermitteln, ob Trident mit Helm installiert wurde.
   - Wenn ein Helm-Label vorhanden ist, wurde Trident mit Helm installiert.
   - Wenn kein Helm-Etikett vorhanden ist, wurde Trident manuell mit dem 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

Wenn Sie 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 zum Bereitstellen von Trident verwendet wurde, weisen Sie den Bediener an, Trident zu deinstallieren. Der Bediener verarbeitet die Entfernung von `TridentOrchestrator` Trident Deployment und demonset und entfernt 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 Trident mit Helm installiert haben, können Sie es mit deinstallieren `helm uninstall`.

```
#List the Helm release corresponding to the 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 Trident verbundenen Ressourcen mit Ausnahme der CRDs und zugehörigen Objekte zu entfernen:

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