



Benutze den Trident.

Trident

NetApp
January 15, 2026

Inhalt

Benutze den Trident	1
Bereiten Sie den Worker-Knoten vor..	1
Die richtigen Werkzeuge auswählen	1
Knotendiensterkennung	1
NFS -Volumes	2
iSCSI-Volumes	2
NVMe/TCP-Volumes	6
SCSI über FC-Volumes	7
Backends konfigurieren und verwalten	10
Backends konfigurieren	10
Azure NetApp Files	10
Google Cloud NetApp Volumes	30
Konfigurieren eines Cloud Volumes Service für das Google Cloud-Backend	47
Konfigurieren Sie ein NetApp HCI oder SolidFire Backend	59
ONTAP SAN-Treiber	64
ONTAP NAS-Treiber	95
Amazon FSx for NetApp ONTAP	133
Backends mit kubectl erstellen	169
Backends verwalten	177
Speicherklassen erstellen und verwalten	187
Erstellen einer Speicherklasse	187
Speicherklassen verwalten	190
Bereitstellung und Verwaltung von Volumina	192
Bereitstellung eines Volumens	192
Volumen erweitern	196
Importmengen	207
Datenträgernamen und -bezeichnungen anpassen	215
Ein NFS-Volume über Namespaces hinweg freigeben	218
Volumes über Namespaces hinweg klonen	222
Volumes mit SnapMirror replizieren	225
CSI-Topologie verwenden	231
Mit Snapshots arbeiten	239
Arbeiten mit Volumengruppen-Snapshots	247

Benutze den Trident.

Bereiten Sie den Worker-Knoten vor.

Alle Worker-Knoten im Kubernetes-Cluster müssen in der Lage sein, die für Ihre Pods bereitgestellten Volumes einzubinden. Zur Vorbereitung der Worker-Knoten müssen Sie je nach Treiberauswahl NFS-, iSCSI-, NVMe/TCP- oder FC-Tools installieren.

Die richtigen Werkzeuge auswählen

Wenn Sie eine Kombination von Treibern verwenden, sollten Sie alle für Ihre Treiber erforderlichen Tools installieren. Bei den neueren Versionen von Red Hat Enterprise Linux CoreOS (RHCOS) sind die Tools standardmäßig installiert.

NFS -Tools

"[Installieren Sie die NFS-Tools](#)" wenn Sie Folgendes verwenden: `ontap-nas` , `ontap-nas-economy` , `ontap-nas-flexgroup` , `azure-netapp-files` , `gcp-cvs` .

iSCSI-Tools

"[Installieren Sie die iSCSI-Tools](#)" wenn Sie Folgendes verwenden: `ontap-san` , `ontap-san-economy` , `solidfire-san` .

NVMe-Tools

"[Installieren Sie die NVMe-Tools](#)" wenn Sie verwenden `ontap-san` für das NVMe-over-TCP-Protokoll (NVMe/TCP).



NetApp empfiehlt ONTAP 9.12 oder höher für NVMe/TCP.

SCSI über FC-Tools

Siehe "[Möglichkeiten zur Konfiguration von FC- und FC-NVMe-SAN-Hosts](#)" Weitere Informationen zur Konfiguration Ihrer FC- und FC-NVMe-SAN-Hosts finden Sie hier.

"[Installieren Sie die FC-Tools](#)" wenn Sie verwenden `ontap-san` mit `sanType fcp` (SCSI über FC).

Zu beachtende Punkte: * SCSI over FC wird in OpenShift- und KubeVirt-Umgebungen unterstützt. * SCSI over FC wird unter Docker nicht unterstützt. * Die Selbstheilung von iSCSI ist bei SCSI über FC nicht anwendbar.

Knotendiensterkennung

Trident versucht automatisch zu erkennen, ob der Knoten iSCSI- oder NFS-Dienste ausführen kann.



Die Knotendiensterkennung identifiziert zwar gefundene Dienste, garantiert aber nicht, dass die Dienste ordnungsgemäß konfiguriert sind. Umgekehrt bedeutet das Fehlen eines erkannten Dienstes nicht zwangsläufig, dass die Volume-Einbindung fehlschlägt.

Ereignisse überprüfen

Trident erstellt Ereignisse für den Knoten, um die gefundenen Dienste zu identifizieren. Um diese Ereignisse anzusehen, führen Sie Folgendes aus:

```
kubectl get event -A --field-selector involvedObject.name=<Kubernetes node name>
```

Überprüfung entdeckter Dienste

Trident identifiziert die für jeden Knoten auf dem Trident -Knoten-CR aktivierten Dienste. Um die gefundenen Dienste anzuzeigen, führen Sie Folgendes aus:

```
tridentctl get node -o wide -n <Trident namespace>
```

NFS -Volumes

Installieren Sie die NFS-Tools mithilfe der Befehle für Ihr Betriebssystem. Stellen Sie sicher, dass der NFS-Dienst beim Systemstart gestartet wird.

RHEL 8+

```
sudo yum install -y nfs-utils
```

Ubuntu

```
sudo apt-get install -y nfs-common
```



Starten Sie Ihre Worker-Knoten nach der Installation der NFS-Tools neu, um Fehler beim Anhängen von Volumes an Container zu vermeiden.

iSCSI-Volumes

Trident kann automatisch eine iSCSI-Sitzung herstellen, LUNs scannen und Multipath-Geräte erkennen, formatieren und in einen Pod einbinden.

iSCSI-Selbstheilungsfunktionen

Bei ONTAP -Systemen führt Trident alle fünf Minuten eine iSCSI-Selbstheilung durch, um Folgendes zu erreichen:

1. **Identifizieren Sie** den gewünschten iSCSI-Sitzungsstatus und den aktuellen iSCSI-Sitzungsstatus.
2. Vergleichen Sie den gewünschten Zustand mit dem aktuellen Zustand, um notwendige Reparaturen zu ermitteln. Trident legt die Reparaturprioritäten fest und entscheidet, wann Reparaturen vorgezogen werden.
3. Führen Sie die erforderlichen Reparaturen durch, um den aktuellen iSCSI-Sitzungsstatus in den gewünschten iSCSI-Sitzungsstatus zurückzusetzen.



Protokolle der Selbstteilungsaktivitäten befinden sich in der `trident-main` Container im jeweiligen Daemonset-Pod. Um Protokolle anzuzeigen, müssen Sie Folgendes eingestellt haben: `debug` auf "wahr" während der Trident -Installation.

Die Selbstteilungsfunktionen von Trident iSCSI können dazu beitragen, Folgendes zu verhindern:

- Veralterte oder fehlerhafte iSCSI-Sitzungen, die nach einem Netzwerkverbindungsproblem auftreten können. Im Falle einer inaktiven Sitzung wartet Trident sieben Minuten, bevor es sich abmeldet, um die Verbindung mit einem Portal wiederherzustellen.



Wenn beispielsweise CHAP-Geheimnisse auf dem Speichercontroller rotiert wurden und die Netzwerkverbindung abbricht, könnten die alten (veralteten) CHAP-Geheimnisse erhalten bleiben. Die Selbstteilungskräfte können dies erkennen und die Sitzung automatisch wiederherstellen, um die aktualisierten CHAP-Geheimnisse anzuwenden.

- Fehlende iSCSI-Sitzungen
- Fehlende LUNs

Punkte, die Sie vor einem Upgrade von Trident beachten sollten

- Wenn nur pro-Knoten-igroups (eingeführt in 23.04+) verwendet werden, initiiert die iSCSI-Selbstteilung SCSI-Neuscans für alle Geräte im SCSI-Bus.
- Wenn nur Backend-bezogene iGroups verwendet werden (veraltet ab Version 23.04), initiiert die iSCSI-Selbstteilung erneute SCSI-Scans für exakte LUN-IDs im SCSI-Bus.
- Bei Verwendung einer Mischung aus knotenbezogenen igroups und backendbezogenen igroups initiiert die iSCSI-Selbstteilung erneute SCSI-Scans für exakte LUN-IDs im SCSI-Bus.

Installieren Sie die iSCSI-Tools

Installieren Sie die iSCSI-Tools mithilfe der Befehle für Ihr Betriebssystem.

Bevor Sie beginnen

- Jeder Knoten im Kubernetes-Cluster muss einen eindeutigen IQN haben. Dies ist eine notwendige Voraussetzung.
- Bei Verwendung von RHCOS Version 4.5 oder höher oder einer anderen RHEL-kompatiblen Linux-Distribution mit `solidfire-san` Treiber und Element OS 12.5 oder früher, stellen Sie sicher, dass der CHAP-Authentifizierungsalgorithmus auf MD5 eingestellt ist. `/etc/iscsi/iscsid.conf` Mit Element 12.7 stehen sichere, FIPS-konforme CHAP-Algorithmen wie SHA1, SHA-256 und SHA3-256 zur Verfügung.

```
sudo sed -i 's/^\(node.session.auth.chap_algs\) .*/\1 = MD5/'  
/etc/iscsi/iscsid.conf
```

- Bei Verwendung von Worker-Knoten, auf denen RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) mit iSCSI-PVs ausgeführt wird, geben Sie Folgendes an: `discard mountOption` in der StorageClass zur Durchführung der Inline-Speicherplatzfreigabe. Siehe ["Red Hat-Dokumentation"](#).
- Stellen Sie sicher, dass Sie auf die neueste Version aktualisiert haben. `multipath-tools`.

RHEL 8+

1. Installieren Sie die folgenden Systempakete:

```
sudo yum install -y lsscsi iscsi-initiator-utils device-mapper-multipath
```

2. Prüfen Sie, ob die Version von iscsi-initiator-utils 6.2.0.874-2.el7 oder höher ist:

```
rpm -q iscsi-initiator-utils
```

3. Scanvorgang auf manuelles Scannen einstellen:

```
sudo sed -i 's/^\(node.session.scan\) .*/\1 = manual/'  
/etc/iscsi/iscsid.conf
```

4. Multipathing aktivieren:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



Sicherstellen /etc/multipath.conf enthält find_multipaths no unter defaults.

5. Stellen Sie sicher, dass iscsid Und multipathd laufen:

```
sudo systemctl enable --now iscsid multipathd
```

6. Aktivieren und starten iscsi:

```
sudo systemctl enable --now iscsi
```

Ubuntu

1. Installieren Sie die folgenden Systempakete:

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools  
scsitools
```

2. Prüfen Sie, ob die Open-iSCSI-Version 2.0.874-5ubuntu2.10 oder höher (für Bionic) bzw. 2.0.874-7.1ubuntu6.1 oder höher (für Focal) ist:

```
dpkg -l open-iscsi
```

3. Scanvorgang auf manuelles Scannen einstellen:

```
sudo sed -i 's/^\(node.session.scan\).*\/\1 = manual/'  
/etc/iscsi/iscsid.conf
```

4. Multipathing aktivieren:

```
sudo tee /etc/multipath.conf <<-EOF  
defaults {  
    user_friendly_names yes  
    find_multipaths no  
}  
EOF  
sudo systemctl enable --now multipath-tools.service  
sudo service multipath-tools restart
```



Sicherstellen /etc/multipath.conf enthält find_multipaths no unter defaults.

5. Stellen Sie sicher, dass open-iscsi Und multipath-tools sind aktiviert und laufen:

```
sudo systemctl status multipath-tools  
sudo systemctl enable --now open-iscsi.service  
sudo systemctl status open-iscsi
```



Für Ubuntu 18.04 müssen Sie die Zielports mit folgendem Befehl ermitteln: `iscsiadm` vor dem Start `open-iscsi` damit der iSCSI-Daemon startet. Alternativ können Sie die `iscsi` Servicestart `iscsid` automatisch.

iSCSI-Selbsteilung konfigurieren oder deaktivieren

Sie können die folgenden Trident iSCSI-Selbsteilungseinstellungen konfigurieren, um veraltete Sitzungen zu reparieren:

- **iSCSI-Selbsteilungsintervall:** Bestimmt die Häufigkeit, mit der die iSCSI-Selbsteilung aufgerufen wird (Standard: 5 Minuten). Sie können die Ausführungshäufigkeit durch Angabe einer kleineren Zahl oder die Ausführungshäufigkeit durch Angabe einer größeren Zahl konfigurieren.



Durch Setzen des iSCSI-Selbstheilungsintervalls auf 0 wird die iSCSI-Selbstheilung vollständig gestoppt. Wir empfehlen nicht, die iSCSI-Selbstheilung zu deaktivieren; sie sollte nur in bestimmten Szenarien deaktiviert werden, wenn die iSCSI-Selbstheilung nicht wie vorgesehen funktioniert oder zu Debugging-Zwecken.

- **iSCSI-Selbstheilungs-Wartezeit:** Bestimmt die Dauer, die die iSCSI-Selbstheilung wartet, bevor sie sich von einer fehlerhaften Sitzung abmeldet und versucht, sich erneut anzumelden (Standard: 7 Minuten). Sie können den Wert auf eine größere Zahl einstellen, sodass Sitzungen, die als fehlerhaft identifiziert werden, länger warten müssen, bevor sie abgemeldet werden und anschließend ein erneuter Anmeldeversuch unternommen wird, oder auf eine kleinere Zahl, um sich früher abzumelden und wieder anzumelden.

Helm

Um die iSCSI-Selbstheilungseinstellungen zu konfigurieren oder zu ändern, übergeben Sie die `iscsiSelfHealingInterval` Und `iscsiSelfHealingWaitTime` Parameter während der Helm-Installation oder des Helm-Updates.

Im folgenden Beispiel wird das iSCSI-Selbstheilungsintervall auf 3 Minuten und die Selbstheilungs-Wartezeit auf 6 Minuten eingestellt:

```
helm install trident trident-operator-100.2506.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

tridentctl

Um die iSCSI-Selbstheilungseinstellungen zu konfigurieren oder zu ändern, übergeben Sie die `iscsi-self-healing-interval` Und `iscsi-self-healing-wait-time` Parameter während der Installation oder Aktualisierung von `tridentctl`.

Im folgenden Beispiel wird das iSCSI-Selbstheilungsintervall auf 3 Minuten und die Selbstheilungs-Wartezeit auf 6 Minuten eingestellt:

```
tridentctl install --iscsi-self-healing-interval=3m0s --iscsi-self
-healing-wait-time=6m0s -n trident
```

NVMe/TCP-Volumes

Installieren Sie die NVMe-Tools mithilfe der Befehle für Ihr Betriebssystem.



- NVMe erfordert RHEL 9 oder höher.
- Falls die Kernelversion Ihres Kubernetes-Knotens zu alt ist oder das NVMe-Paket für Ihre Kernelversion nicht verfügbar ist, müssen Sie möglicherweise die Kernelversion Ihres Knotens auf eine Version mit dem NVMe-Paket aktualisieren.

RHEL 9

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Ubuntu

```
sudo apt install nvme-cli
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Überprüfen der Installation

Überprüfen Sie nach der Installation mit folgendem Befehl, ob jeder Knoten im Kubernetes-Cluster einen eindeutigen NQN besitzt:

```
cat /etc/nvme/hostnqn
```



Trident modifiziert die `ctrl_device_tmo` Wert, um sicherzustellen, dass NVMe den Pfad nicht aufgibt, falls die Verbindung abbricht. Diese Einstellung darf nicht geändert werden.

SCSI über FC-Volumes

Sie können nun das Fibre Channel (FC)-Protokoll mit Trident verwenden, um Speicherressourcen auf einem ONTAP -System bereitzustellen und zu verwalten.

Voraussetzungen

Konfigurieren Sie die erforderlichen Netzwerk- und Knoteneinstellungen für FC.

Netzwerkeinstellungen

1. Ermitteln Sie die WWPN der Zielschnittstellen. Siehe ["Netzwerkschnittstelle anzeigen"](#) für weitere Informationen.
2. Ermitteln Sie die WWPN für die Schnittstellen auf dem Initiator (Host).

Siehe die entsprechenden Dienstprogramme des Host-Betriebssystems.

3. Konfigurieren Sie die Zoneneinteilung auf dem FC-Switch mithilfe der WWPNs des Hosts und des Ziels.

Weitere Informationen finden Sie in der Dokumentation des jeweiligen Switch-Herstellers.

Weitere Einzelheiten entnehmen Sie bitte der folgenden ONTAP -Dokumentation:

- ["Fibre Channel und FCoE-Zonierungsübersicht"](#)

- ["Möglichkeiten zur Konfiguration von FC- und FC-NVMe-SAN-Hosts"](#)

Installieren Sie die FC-Tools

Installieren Sie die FC-Tools mithilfe der Befehle für Ihr Betriebssystem.

- Bei Verwendung von Worker-Knoten, auf denen RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) mit FC-PVs ausgeführt wird, geben Sie Folgendes an: `discard` mountOption in der StorageClass zur Durchführung der Inline-Speicherplatzfreigabe. Siehe ["Red Hat-Dokumentation"](#) .

RHEL 8+

1. Installieren Sie die folgenden Systempakete:

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. Multipathing aktivieren:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



Sicherstellen /etc/multipath.conf enthält find_multipaths no unter defaults.

3. Stellen Sie sicher, dass multipathd läuft:

```
sudo systemctl enable --now multipathd
```

Ubuntu

1. Installieren Sie die folgenden Systempakete:

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. Multipathing aktivieren:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```



Sicherstellen /etc/multipath.conf enthält find_multipaths no unter defaults.

3. Stellen Sie sicher, dass multipath-tools ist aktiviert und läuft:

```
sudo systemctl status multipath-tools
```

Backends konfigurieren und verwalten

Backends konfigurieren

Ein Backend definiert die Beziehung zwischen Trident und einem Speichersystem. Es teilt Trident mit, wie mit diesem Speichersystem kommuniziert werden soll und wie Trident Datenträger daraus bereitstellen soll.

Trident bietet automatisch Speicherpools von Backends an, die den Anforderungen einer Speicherklasse entsprechen. Erfahren Sie, wie Sie das Backend für Ihr Speichersystem konfigurieren.

- ["Konfigurieren eines Azure NetApp Files -Backends"](#)
- ["Konfigurieren eines Google Cloud NetApp Volumes -Backends"](#)
- ["Konfigurieren eines Cloud Volumes Service für das Google Cloud Platform-Backend"](#)
- ["Konfigurieren Sie ein NetApp HCI oder SolidFire Backend"](#)
- ["Konfigurieren Sie ein Backend mit ONTAP oder Cloud Volumes ONTAP NAS-Treibern"](#)
- ["Konfigurieren Sie ein Backend mit ONTAP oder Cloud Volumes ONTAP SAN-Treibern"](#)
- ["Trident mit Amazon FSx for NetApp ONTAP verwenden"](#)

Azure NetApp Files

Konfigurieren eines Azure NetApp Files -Backends

Sie können Azure NetApp Files als Backend für Trident konfigurieren. Sie können NFS- und SMB-Volumes mithilfe eines Azure NetApp Files Backends einbinden. Trident unterstützt außerdem die Verwaltung von Anmeldeinformationen mithilfe verwalteter Identitäten für Azure Kubernetes Services (AKS)-Cluster.

Details zum Azure NetApp Files Treiber

Trident stellt die folgenden Azure NetApp Files -Speichertreiber für die Kommunikation mit dem Cluster bereit. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	Lautstärke modus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
azure-netapp-files	NFS SMB	Dateisystem	RWO, ROX, RWX, RWOP	nfs, smb

Überlegungen

- Der Azure NetApp Files -Dienst unterstützt keine Volumes unter 50 GiB. Trident erstellt automatisch 50-GiB-Volumes, wenn ein kleineres Volume angefordert wird.
- Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten laufen.

Verwaltete Identitäten für AKS

Trident unterstützt "verwaltete Identitäten" für Azure Kubernetes Services-Cluster. Um die Vorteile der optimierten Anmeldeinformationsverwaltung durch verwaltete Identitäten nutzen zu können, benötigen Sie Folgendes:

- Ein mit AKS bereitgestellter Kubernetes-Cluster
- Auf dem AKS-Kubernetes-Cluster konfigurierte verwaltete Identitäten
- Trident installiert, das Folgendes beinhaltet: `cloudProvider` um zu spezifizieren "Azure" .

Trident -Betreiber

Um Trident mithilfe des Trident -Operators zu installieren, bearbeiten Sie `tridentorchestrator_cr.yaml` einstellen `cloudProvider` Zu "Azure" . Beispiel:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
```

Helm

Das folgende Beispiel installiert Trident -Sets `cloudProvider` für Azure mithilfe der Umgebungsvariablen `$CP` :

```
helm install trident trident-operator-100.2506.0.tgz --create
--namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

<code>tridentctl</code>

Das folgende Beispiel installiert Trident und konfiguriert die `cloudProvider` Flagge an Azure :

```
tridentctl install --cloud-provider="Azure" -n trident
```

Cloud-Identität für AKS

Cloud Identity ermöglicht es Kubernetes-Pods, auf Azure-Ressourcen zuzugreifen, indem sie sich als Workload-Identität authentifizieren, anstatt explizite Azure-Anmeldeinformationen anzugeben.

Um die Vorteile der Cloud-Identität in Azure nutzen zu können, benötigen Sie Folgendes:

- Ein mit AKS bereitgestellter Kubernetes-Cluster
- Workload-Identität und OIDC-Aussteller wurden auf dem AKS Kubernetes-Cluster konfiguriert.
- Trident installiert, das Folgendes beinhaltet: `cloudProvider` um zu spezifizieren "Azure" Und `cloudIdentity` Angabe der Workload-Identität

Trident -Betreiber

Um Trident mithilfe des Trident -Operators zu installieren, bearbeiten Sie

tridentorchestrator_cr.yaml **einstellen** cloudProvider Zu "Azure" und **setzen** cloudIdentity Zu azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx .

Beispiel:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
  cloudIdentity: 'azure.workload.identity/client-id: xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx' # Edit
```

Helm

Legen Sie die Werte für die Flags **cloud-provider (CP)** und **cloud-identity (CI)** mithilfe der folgenden Umgebungsvariablen fest:

```
export CP="Azure"
export CI="'azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx'"
```

Das folgende Beispiel installiert Trident und konfiguriert es. cloudProvider für Azure mithilfe der Umgebungsvariablen \$CP und stellt die cloudIdentity unter Verwendung der Umgebungsvariablen \$CI :

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$CI"
```

<code>tridentctl</code>

Legen Sie die Werte für die Flags **Cloud-Anbieter** und **Cloud-Identität** mithilfe der folgenden Umgebungsvariablen fest:

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx"
```

Das folgende Beispiel installiert Trident und konfiguriert die cloud-provider Flagge an \$CP , Und cloud-identity Zu \$CI :

```
tridentctl install --cloud-provider=$CP --cloud-identity="$CI" -n
trident
```

Bereiten Sie die Konfiguration eines Azure NetApp Files -Backends vor.

Bevor Sie Ihr Azure NetApp Files Backend konfigurieren können, müssen Sie sicherstellen, dass die folgenden Anforderungen erfüllt sind.

Voraussetzungen für NFS- und SMB-Volumes

Wenn Sie Azure NetApp Files zum ersten Mal oder an einem neuen Standort verwenden, ist eine anfängliche Konfiguration erforderlich, um Azure NetApp Files einzurichten und ein NFS-Volume zu erstellen. Siehe ["Azure: Azure NetApp Files einrichten und ein NFS-Volume erstellen"](#) .

Um ein ["Azure NetApp Files"](#) Im Backend benötigen Sie Folgendes:



- `subscriptionID`, `tenantID`, `clientID`, `location`, Und `clientSecret` sind optional bei der Verwendung von verwalteten Identitäten auf einem AKS-Cluster.
- `tenantID`, `clientID`, Und `clientSecret` sind optional, wenn eine Cloud-Identität auf einem AKS-Cluster verwendet wird.

- Ein Kapazitätspool. Siehe ["Microsoft: Erstellen eines Kapazitätspools für Azure NetApp Files"](#) .
- Ein an Azure NetApp Files delegiertes Subnetz. Siehe ["Microsoft: Ein Subnetz an Azure NetApp Files delegieren"](#) .
- ``subscriptionID`` aus einem Azure-Abonnement mit aktiviertem Azure NetApp Files .
- `tenantID`, `clientID`, Und `clientSecret` von einem ["App-Registrierung"](#) in Azure Active Directory mit ausreichenden Berechtigungen für den Azure NetApp Files -Dienst. Für die App-Registrierung sollte eines der folgenden Verfahren verwendet werden:
 - Die Rolle des Eigentümers oder Mitwirkenden ["von Azure vordefiniert"](#) .
 - A ["benutzerdefinierte Mitwirkenderrolle"](#) auf Abonnementebene(`assignableScopes`) mit den folgenden Berechtigungen, die auf das beschränkt sind, was Trident benötigt. Nach dem Erstellen der benutzerdefinierten Rolle, ["Weisen Sie die Rolle über das Azure-Portal zu."](#) .


```
{
  "id": "/subscriptions/<subscription-id>/providers/Microsoft.Authorization/roleDefinitions/<role-definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.NetApp/netAppAccounts/capacityPools/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/write",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/read",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/write",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/delete",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTargets/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",

          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/read",

          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/write",

          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/delete",
        ]
      }
    ]
  }
}
```

```

        "Microsoft.Features/features/read",
        "Microsoft.Features/operations/read",
        "Microsoft.Features/providers/features/read",

        "Microsoft.Features/providers/features/register/action",

        "Microsoft.Features/providers/features/unregister/action",

        "Microsoft.Features/subscriptionFeatureRegistrations/read"
    ],
    "notActions": [],
    "dataActions": [],
    "notDataActions": []
  }
]
}

```

- Die Azur location das mindestens eines enthält "[delegiertes Subnetz](#)". Stand Trident 22.01, location Der Parameter ist ein Pflichtfeld auf oberster Ebene der Backend-Konfigurationsdatei. In virtuellen Pools angegebene Standortwerte werden ignoriert.
- Anwendung Cloud Identity, hol dir die client ID von einem "[vom Benutzer zugewiesene verwaltete Identität](#)" und geben Sie diese ID an in azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx .

Zusätzliche Anforderungen für SMB-Volumina

Zum Erstellen eines SMB-Volumes benötigen Sie Folgendes:

- Active Directory ist konfiguriert und mit Azure NetApp Files verbunden. Siehe "[Microsoft: Erstellen und Verwalten von Active Directory-Verbindungen für Azure NetApp Files](#)".
- Ein Kubernetes-Cluster mit einem Linux-Controller-Knoten und mindestens einem Windows-Worker-Knoten, auf dem Windows Server 2022 ausgeführt wird. Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten laufen.
- Mindestens ein Trident Geheimnis, das Ihre Active Directory-Anmeldeinformationen enthält, damit Azure NetApp Files sich bei Active Directory authentifizieren kann. Um Geheimnisse zu generieren smbcreds :

```

kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'

```

- Ein als Windows-Dienst konfigurierter CSI-Proxy. Um einen zu konfigurieren csi-proxy , siehe "[GitHub: CSI-Proxy](#)" oder "[GitHub: CSI-Proxy für Windows](#)" für Kubernetes-Knoten, die unter Windows laufen.

Konfigurationsoptionen und Beispiele für das Azure NetApp Files Backend

Erfahren Sie mehr über die NFS- und SMB-Backend-Konfigurationsoptionen für Azure

NetApp Files und sehen Sie sich Konfigurationsbeispiele an.

Backend-Konfigurationsoptionen

Trident verwendet Ihre Backend-Konfiguration (Subnetz, virtuelles Netzwerk, Dienstebene und Standort), um Azure NetApp Files Volumes auf Kapazitätspools zu erstellen, die am angeforderten Standort verfügbar sind und der angeforderten Dienstebene und dem Subnetz entsprechen.



* Ab der Version NetApp Trident 25.06 werden manuelle QoS-Kapazitätspools als technische Vorschau unterstützt.*

Die Azure NetApp Files -Backends bieten diese Konfigurationsoptionen.

Parameter	Beschreibung	Standard
version		Immer 1
storageDriverName	Name des Speichertreibers	"azure-netapp-files"
backendName	Benutzerdefinierter Name oder das Speicher-Backend	Fahrername + "_" + zufällige Zeichen
subscriptionID	Die Abonnement-ID Ihres Azure-Abonnements. Optional, wenn verwaltete Identitäten in einem AKS-Cluster aktiviert sind.	
tenantID	Die Mandanten-ID aus einer App-Registrierung ist optional, wenn verwaltete Identitäten oder Cloud-Identitäten auf einem AKS-Cluster verwendet werden.	
clientID	Die Client-ID aus einer App-Registrierung ist optional, wenn verwaltete Identitäten oder Cloud-Identitäten auf einem AKS-Cluster verwendet werden.	
clientSecret	Der Client-Schlüssel aus einer App-Registrierung ist optional, wenn verwaltete Identitäten oder Cloud-Identitäten auf einem AKS-Cluster verwendet werden.	
serviceLevel	Einer von Standard, Premium, oder Ultra	"" (zufällig)
location	Name des Azure-Standorts, an dem die neuen Volumes erstellt werden. Optional, wenn verwaltete Identitäten in einem AKS-Cluster aktiviert sind.	
resourceGroups	Liste der Ressourcengruppen zum Filtern der gefundenen Ressourcen	[] (kein Filter)

Parameter	Beschreibung	Standard
netappAccounts	Liste der NetApp -Konten zum Filtern der gefundenen Ressourcen	[] (kein Filter)
capacityPools	Liste der Kapazitätspools zum Filtern der gefundenen Ressourcen	[] (kein Filter, zufällig)
virtualNetwork	Name eines virtuellen Netzwerks mit einem delegierten Subnetz	""
subnet	Name eines delegierten Subnetzes an Microsoft.Netapp/volumes	""
networkFeatures	Satz von VNet-Funktionen für ein Volume, kann sein Basic oder Standard. Die Netzwerkfunktionen sind nicht in allen Regionen verfügbar und müssen gegebenenfalls im Rahmen eines Abonnements aktiviert werden. Spezifizierung networkFeatures Wenn die Funktionalität nicht aktiviert ist, schlägt die Volumenbereitstellung fehl.	""
nfsMountOptions	Feingranulare Steuerung der NFS-Mount-Optionen. Wird bei SMB-Volumes ignoriert. Um Volumes mit NFS Version 4.1 einzubinden, fügen Sie Folgendes hinzu: nfsvers=4 in der durch Kommas getrennten Mount-Optionsliste die Option NFS v4.1 auswählen. Die in einer Speicherklassendefinition festgelegten Mount-Optionen überschreiben die in der Backend-Konfiguration festgelegten Mount-Optionen.	"nfsvers=3"
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet.	"" (wird nicht standardmäßig erzwungen)
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, <pre>\{"api": false, "method": true, "discovery": true\}</pre> . Verwenden Sie diese Funktion nur, wenn Sie eine Fehlerbehebung durchführen und einen detaillierten Protokollauszug benötigen.	null

Parameter	Beschreibung	Standard
nasType	Konfiguration der Erstellung von NFS- oder SMB-Volumes. Optionen sind <code>nfs</code> , <code>smb</code> oder <code>null</code> . Bei der Einstellung „null“ werden standardmäßig NFS-Volumes verwendet.	<code>nfs</code>
supportedTopologies	Stellt eine Liste der Regionen und Zonen dar, die von diesem Backend unterstützt werden. Weitere Informationen finden Sie unter "CSI-Topologie verwenden" .	
qosType	Stellt den QoS-Typ dar: Automatisch oder Manuell. Technische Vorschau für Trident 25.06	Automatisch
maxThroughput	Legt den maximal zulässigen Durchsatz in MiB/Sek. fest. Wird nur für manuelle QoS-Kapazitätspools unterstützt. Technische Vorschau für Trident 25.06	4 MiB/sec



Weitere Informationen zu Netzwerkfunktionen finden Sie unter ["Konfigurieren von Netzwerkfunktionen für ein Azure NetApp Files Volume"](#).

Erforderliche Berechtigungen und Ressourcen

Wenn beim Erstellen eines PVC die Fehlermeldung „Keine Kapazitätspools gefunden“ angezeigt wird, liegt es wahrscheinlich daran, dass Ihrer App-Registrierung die erforderlichen Berechtigungen und Ressourcen (Subnetz, virtuelles Netzwerk, Kapazitätspool) nicht zugeordnet sind. Wenn der Debug-Modus aktiviert ist, protokolliert Trident die beim Erstellen des Backends erkannten Azure-Ressourcen. Vergewissern Sie sich, dass die richtige Rolle verwendet wird.

Die Werte für `resourceGroups`, `netappAccounts`, `capacityPools`, `virtualNetwork`, Und `subnet` können mit Kurznamen oder vollständigen Namen angegeben werden. In den meisten Fällen empfiehlt es sich, vollständige Namen anzugeben, da kurze Namen zu mehreren Ressourcen mit demselben Namen führen können.

Der `resourceGroups`, `netappAccounts`, Und `capacityPools` Die Werte sind Filter, die die Menge der gefundenen Ressourcen auf diejenigen beschränken, die diesem Speichersystem zur Verfügung stehen, und können in beliebiger Kombination angegeben werden. Vollständige Namen folgen diesem Format:

Typ	Format
Ressourcengruppe	<Ressourcengruppe>
NetApp Konto	<Ressourcengruppe>/<NetApp-Konto>
Kapazitätspool	<Ressourcengruppe>/<NetApp-Konto>/<Kapazitätspool>

Typ	Format
Virtuelles Netzwerk	<Ressourcengruppe>/<virtuelles Netzwerk>
Subnetz	<Ressourcengruppe>/<virtuelles Netzwerk>/<Subnetz>

Volume-Bereitstellung

Sie können die Standard-Volume-Bereitstellung steuern, indem Sie die folgenden Optionen in einem speziellen Abschnitt der Konfigurationsdatei angeben. Siehe [Beispielkonfigurationen](#) für Details.

Parameter	Beschreibung	Standard
<code>exportRule</code>	Exportregeln für neue Volumes. <code>exportRule</code> Es muss sich um eine durch Kommas getrennte Liste beliebiger Kombinationen von IPv4-Adressen oder IPv4-Subnetzen in CIDR-Notation handeln. Wird bei SMB-Volumes ignoriert.	"0.0.0.0/0"
<code>snapshotDir</code>	Steuert die Sichtbarkeit des <code>.snapshot</code> -Verzeichnisses	"true" für NFSv4, "false" für NFSv3
<code>size</code>	Die Standardgröße neuer Volumes	"100G"
<code>unixPermissions</code>	Die Unix-Berechtigungen für neue Datenträger (4 Oktalstellen). Wird bei SMB-Volumes ignoriert.	"" (Vorschaufunktion, erfordert Whitelisting im Abonnement)

Beispielkonfigurationen

Die folgenden Beispiele zeigen Basiskonfigurationen, bei denen die meisten Parameter auf Standardwerte eingestellt bleiben. Dies ist die einfachste Möglichkeit, ein Backend zu definieren.

Minimale Konfiguration

Dies ist die absolute Minimalkonfiguration im Backend. Mit dieser Konfiguration erkennt Trident alle Ihre NetApp -Konten, Kapazitätspools und Subnetze, die an Azure NetApp Files delegiert sind, am konfigurierten Speicherort und platziert neue Volumes zufällig auf einem dieser Pools und Subnetze. Weil `nasType` wird ausgelassen, die `nfs` Es gelten die Standardeinstellungen, und das Backend stellt NFS-Volumes bereit.

Diese Konfiguration ist ideal, wenn Sie gerade erst mit Azure NetApp Files beginnen und verschiedene Funktionen ausprobieren möchten. In der Praxis werden Sie jedoch eine zusätzliche Bereichsdefinition für die von Ihnen bereitgestellten Volumes wünschen.

```
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
  tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
  clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
  clientSecret: SECRET
  location: eastus
```

Verwaltete Identitäten für AKS

Diese Backend-Konfiguration lässt Folgendes aus: `subscriptionID`, `tenantID`, `clientId`, Und `clientSecret`, die bei der Verwendung verwalteter Identitäten optional sind.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
```


Cloud-Identität für AKS

Diese Backend-Konfiguration lässt Folgendes aus: `tenantID`, `clientID`, Und `clientSecret`, die bei Verwendung einer Cloud-Identität optional sind.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

Spezifische Service-Level-Konfiguration mit Kapazitätspoolfiltern

Diese Backend-Konfiguration platziert Volumes in Azure eastus Ort in einem Ultra Kapazitätspool. Trident erkennt automatisch alle an Azure NetApp Files delegierten Subnetze an diesem Standort und platziert ein neues Volume zufällig auf einem davon.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
```

Backend-Beispiel mit manuellen QoS-Kapazitätspools

Diese Backend-Konfiguration platziert Volumes in Azure `eastus` Standort mit manuellen QoS-Kapazitätspools. **Technische Vorschau in NetApp Trident 25.06.**

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: anfl
location: eastus
labels:
  clusterName: test-cluster-1
  cloud: anf
  nasType: nfs
defaults:
  qosType: Manual
storage:
  - serviceLevel: Ultra
    labels:
      performance: gold
    defaults:
      maxThroughput: 10
  - serviceLevel: Premium
    labels:
      performance: silver
    defaults:
      maxThroughput: 5
  - serviceLevel: Standard
    labels:
      performance: bronze
    defaults:
      maxThroughput: 3
```

Erweiterte Konfiguration

Diese Backend-Konfiguration beschränkt den Umfang der Volume-Platzierung weiter auf ein einzelnes Subnetz und ändert außerdem einige Standardeinstellungen für die Volume-Bereitstellung.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
virtualNetwork: my-virtual-network
subnet: my-subnet
networkFeatures: Standard
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 500Gi
defaults:
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  snapshotDir: "true"
  size: 200Gi
  unixPermissions: "0777"
```

Konfiguration eines virtuellen Pools

Diese Backend-Konfiguration definiert mehrere Speicherpools in einer einzigen Datei. Dies ist nützlich, wenn Sie mehrere Kapazitätspools haben, die unterschiedliche Servicelevel unterstützen, und Sie Speicherklassen in Kubernetes erstellen möchten, die diese repräsentieren. Virtuelle Poolbezeichnungen wurden verwendet, um die Pools anhand folgender Kriterien zu unterscheiden: `performance`.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
resourceGroups:
  - application-group-1
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
  - labels:
      performance: gold
      serviceLevel: Ultra
      capacityPools:
        - ultra-1
        - ultra-2
      networkFeatures: Standard
  - labels:
      performance: silver
      serviceLevel: Premium
      capacityPools:
        - premium-1
  - labels:
      performance: bronze
      serviceLevel: Standard
      capacityPools:
        - standard-1
        - standard-2
```

Unterstützte Topologiekonfiguration

Trident ermöglicht die Bereitstellung von Volumes für Workloads basierend auf Regionen und Verfügbarkeitszonen. Der `supportedTopologies` Der Block in dieser Backend-Konfiguration dient dazu, eine Liste von Regionen und Zonen pro Backend bereitzustellen. Die hier angegebenen Regions- und Zonenwerte müssen mit den Regions- und Zonenwerten der Labels auf jedem Kubernetes-Clusterknoten übereinstimmen. Diese Regionen und Zonen stellen die Liste der zulässigen Werte dar, die in einer Speicherklasse angegeben werden können. Für Speicherklassen, die eine Teilmenge der im Backend bereitgestellten Regionen und Zonen enthalten, erstellt Trident Volumes in der genannten Region und Zone. Weitere Informationen finden Sie unter "[CSI-Topologie verwenden](#)".

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
supportedTopologies:
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-1
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-2
```

Speicherklassendefinitionen

Die folgende `StorageClass` Definitionen beziehen sich auf die oben genannten Speicherpools.

Beispieldefinitionen mit `parameter.selector` Feld

Verwenden `parameter.selector` Sie können für jedes einzelne festlegen `StorageClass` Der virtuelle Pool, der zum Hosten eines Volumes verwendet wird. Das Volumen wird die im gewählten Pool definierten Aspekte aufweisen.

```

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze
allowVolumeExpansion: true

```

Beispieldefinitionen für SMB-Volumes

Verwenden `nasType` , `node-stage-secret-name` , Und `node-stage-secret-namespace` Sie können ein SMB-Volume angeben und die erforderlichen Active Directory-Anmeldeinformationen bereitstellen.

Grundkonfiguration im Standard-Namespace

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

Verwendung unterschiedlicher Geheimnisse pro Namensraum

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

Verwendung unterschiedlicher Geheimnisse pro Band

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```



`nasType: smb`Filter für Pools, die SMB-Volumes unterstützen. `nasType: nfs oder nasType: null`Filter für NFS-Pools.`

Backend erstellen

Nachdem Sie die Backend-Konfigurationsdatei erstellt haben, führen Sie folgenden Befehl aus:

```
tridentctl create backend -f <backend-file>
```

Wenn die Backend-Erstellung fehlschlägt, stimmt etwas mit der Backend-Konfiguration nicht. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Befehl zum Erstellen erneut ausführen.

Google Cloud NetApp Volumes

Konfigurieren eines Google Cloud NetApp Volumes -Backends

Sie können jetzt Google Cloud NetApp Volumes als Backend für Trident konfigurieren. Sie können NFS- und SMB-Volumes über ein Google Cloud NetApp Volumes -Backend einbinden.

Details zum Google Cloud NetApp Volumes -Treiber

Trident bietet die `google-cloud-netapp-volumes` Der Treiber soll mit dem Cluster kommunizieren. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	Lautstärke modus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
google-cloud-netapp-volumes	NFS SMB	Dateisystem	RWO, ROX, RWX, RWOP	nfs, smb

Cloud-Identität für GKE

Cloud Identity ermöglicht es Kubernetes-Pods, auf Google Cloud-Ressourcen zuzugreifen, indem sie sich als Workload-Identität authentifizieren, anstatt explizite Google Cloud-Anmeldeinformationen anzugeben.

Um die Cloud-Identität in Google Cloud nutzen zu können, benötigen Sie Folgendes:

- Ein mit GKE bereitgestellter Kubernetes-Cluster.
- Die Workload-Identität wurde im GKE-Cluster konfiguriert und der GKE MetaData Server auf den Knotenpools.
- Ein GCP-Dienstkonto mit der Rolle „Google Cloud NetApp Volumes Admin“ (`roles/netapp.admin`) oder

einer benutzerdefinierten Rolle.

- Trident wurde installiert, einschließlich des Cloud-Providers, der "GCP" angibt, und der Cloud-Identität, die das neue GCP-Dienstkonto angibt. Nachfolgend ein Beispiel.

Trident -Betreiber

Um Trident mithilfe des Trident -Operators zu installieren, bearbeiten Sie `tridentorchestrator_cr.yaml` einstellen `cloudProvider` Zu "GCP" und setzen `cloudIdentity` Zu `iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com`.

Beispiel:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "GCP"
  cloudIdentity: 'iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com'
```

Helm

Legen Sie die Werte für die Flags **cloud-provider (CP)** und **cloud-identity (CI)** mithilfe der folgenden Umgebungsvariablen fest:

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com'"
```

Das folgende Beispiel installiert Trident und konfiguriert es. `cloudProvider` für GCP unter Verwendung der Umgebungsvariablen `$CP` und stellt die `cloudIdentity` unter Verwendung der Umgebungsvariablen `$ANNOTATION`:

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

<code>tridentctl</code>

Legen Sie die Werte für die Flags **Cloud-Anbieter** und **Cloud-Identität** mithilfe der folgenden Umgebungsvariablen fest:

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com'"
```

Das folgende Beispiel installiert Trident und konfiguriert die `cloud-provider` Flagge an `$CP`, Und `cloud-identity` Zu `$ANNOTATION`:

```
tridentctl install --cloud-provider=$CP --cloud
-identity="$ANNOTATION" -n trident
```

Bereiten Sie die Konfiguration eines Google Cloud NetApp Volumes Backends vor.

Bevor Sie Ihr Google Cloud NetApp Volumes Backend konfigurieren können, müssen Sie sicherstellen, dass die folgenden Voraussetzungen erfüllt sind.

Voraussetzungen für NFS-Volumes

Wenn Sie Google Cloud NetApp Volumes zum ersten Mal oder an einem neuen Standort verwenden, ist eine anfängliche Konfiguration erforderlich, um Google Cloud NetApp Volumes einzurichten und ein NFS-Volume zu erstellen. Siehe ["Bevor Sie beginnen"](#).

Stellen Sie sicher, dass Sie Folgendes haben, bevor Sie das Google Cloud NetApp Volumes -Backend konfigurieren:

- Ein Google Cloud-Konto, das mit dem Google Cloud NetApp Volumes -Dienst konfiguriert ist. Siehe ["Google Cloud NetApp Volumes"](#).
- Projektnummer Ihres Google Cloud-Kontos. Siehe ["Projekte identifizieren"](#).
- Ein Google Cloud-Dienstkonto mit NetApp Volumes-Administratorrechten(`roles/netapp.admin`) Rolle. Siehe ["Rollen und Berechtigungen für Identitäts- und Zugriffsmanagement"](#).
- API-Schlüsseldatei für Ihr GCNV-Konto. Siehe ["Erstellen Sie einen Dienstkontoschlüssel"](#)
- Ein Speicherbecken. Siehe ["Übersicht der Speicherpools"](#).

Weitere Informationen zum Einrichten des Zugriffs auf Google Cloud NetApp Volumes finden Sie unter ["Zugriff auf Google Cloud NetApp Volumes einrichten"](#).

Google Cloud NetApp Volumes Backend-Konfigurationsoptionen und Beispiele

Erfahren Sie mehr über die Backend-Konfigurationsoptionen für Google Cloud NetApp Volumes und sehen Sie sich Konfigurationsbeispiele an.

Backend-Konfigurationsoptionen

Jedes Backend stellt Volumes in einer einzelnen Google Cloud-Region bereit. Um Volumes in anderen Regionen zu erstellen, können Sie zusätzliche Backends definieren.

Parameter	Beschreibung	Standard
version		Immer 1
storageDriverName	Name des Speichertreibers	Der Wert von <code>storageDriverName</code> muss als "google-cloud-netapp-volumes" angegeben werden.

Parameter	Beschreibung	Standard
backendName	(Optional) Benutzerdefinierter Name des Speicher-Backends	Fahrername + "_" + Teil des API-Schlüssels
storagePools	Optionaler Parameter zur Angabe von Speicherpools für die Volume-Erstellung.	
projectNumber	Google Cloud-Konto-Projektnummer. Den Wert finden Sie auf der Startseite des Google Cloud-Portals.	
location	Der Google Cloud-Speicherort, an dem Trident GCNV-Volumes erstellt. Beim Erstellen regionsübergreifender Kubernetes-Cluster werden Volumes, die in einem location kann in Workloads verwendet werden, die auf Knoten in mehreren Google Cloud-Regionen geplant sind. Für den Verkehr über Regionen hinweg fallen zusätzliche Kosten an.	
apiKey	API-Schlüssel für das Google Cloud-Dienstkonto mit dem netapp.admin Rolle. Sie enthält den JSON-formatierten Inhalt der privaten Schlüsseldatei eines Google Cloud-Dienstkontos (wörtlich in die Backend-Konfigurationsdatei kopiert). Der apiKey müssen Schlüssel-Wert-Paare für die folgenden Schlüssel enthalten: type, project_id, client_email, client_id, auth_uri, token_uri, auth_provider_x509_cert_url, Und client_x509_cert_url.	
nfsMountOptions	Feingranulare Steuerung der NFS-Mount-Optionen.	"nfsvers=3"
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet.	"" (wird nicht standardmäßig erzwungen)
serviceLevel	Der Servicegrad eines Speicherpools und seiner Volumes. Die Werte sind flex, standard, premium, oder extreme.	
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Datenträger angewendet werden sollen	""
network	Das Google Cloud-Netzwerk wird für GCNV-Volumes verwendet.	
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, {"api":false, "method":true}. Verwenden Sie diese Funktion nur, wenn Sie eine Fehlerbehebung durchführen und einen detaillierten Protokollauszug benötigen.	null
nasType	Konfiguration der Erstellung von NFS- oder SMB-Volumes. Optionen sind nfs, smb oder null. Bei der Einstellung „null“ werden standardmäßig NFS-Volumes verwendet.	nfs

Parameter	Beschreibung	Standard
supportedTopologies	Stellt eine Liste der Regionen und Zonen dar, die von diesem Backend unterstützt werden. Weitere Informationen finden Sie unter " CSI-Topologie verwenden ". Zum Beispiel: supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

Volumenbereitstellungsoptionen

Sie können die Standard-Volume-Bereitstellung in der `defaults` Abschnitt der Konfigurationsdatei.

Parameter	Beschreibung	Standard
exportRule	Die Ausführbestimmungen für neue Bände. Es muss sich um eine durch Kommas getrennte Liste beliebiger Kombinationen von IPv4-Adressen handeln.	"0.0.0.0/0"
snapshotDir	Zugang zu <code>.snapshot</code> Verzeichnis	"true" für NFSv4, "false" für NFSv3
snapshotReserve	Prozentsatz des für Snapshots reservierten Speichervolumens	"" (Standardwert 0 akzeptieren)
unixPermissions	Die Unix-Berechtigungen für neue Datenträger (4 Oktalstellen).	""

Beispielkonfigurationen

Die folgenden Beispiele zeigen Basiskonfigurationen, bei denen die meisten Parameter auf Standardwerte eingestellt bleiben. Dies ist die einfachste Möglichkeit, ein Backend zu definieren.

Minimale Konfiguration

Dies ist die absolute Minimalkonfiguration im Backend. Mit dieser Konfiguration erkennt Trident alle Ihre Speicherpools, die an Google Cloud NetApp Volumes delegiert sind, am konfigurierten Speicherort und platziert neue Volumes zufällig in einem dieser Pools. Weil `nasType` wird ausgelassen, die `nfs` Es gelten die Standardeinstellungen, und das Backend stellt NFS-Volumes bereit.

Diese Konfiguration ist ideal, wenn Sie gerade erst mit Google Cloud NetApp Volumes beginnen und verschiedene Funktionen ausprobieren möchten. In der Praxis werden Sie jedoch höchstwahrscheinlich zusätzliche Einschränkungen für die von Ihnen bereitgestellten Volumes benötigen.

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret

```

Konfiguration für SMB-Volumes

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```



```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret

```

Konfiguration eines virtuellen Pools

Diese Backend-Konfiguration definiert mehrere virtuelle Pools in einer einzigen Datei. Virtuelle Pools werden definiert in der `storage` Abschnitt. Sie sind nützlich, wenn Sie mehrere Speicherpools haben, die unterschiedliche Servicelevel unterstützen, und Sie Speicherklassen in Kubernetes erstellen möchten, die diese repräsentieren. Virtuelle Poolbezeichnungen dienen zur Unterscheidung der Pools. Zum Beispiel im folgenden Beispiel `performance` Etikett und `serviceLevel` Der Typ wird verwendet, um virtuelle Pools zu unterscheiden.

Sie können auch einige Standardwerte festlegen, die für alle virtuellen Pools gelten, und die Standardwerte für einzelne virtuelle Pools überschreiben. Im folgenden Beispiel `snapshotReserve` Und `exportRule` dienen als Standardwerte für alle virtuellen Pools.

Weitere Informationen finden Sie unter "[Virtuelle Pools](#)".

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
```

```

auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
credentials:
  name: backend-tbc-gcnv-secret
defaults:
  snapshotReserve: "10"
  exportRule: 10.0.0.0/24
storage:
- labels:
  performance: extreme
  serviceLevel: extreme
  defaults:
    snapshotReserve: "5"
    exportRule: 0.0.0.0/0
- labels:
  performance: premium
  serviceLevel: premium
- labels:
  performance: standard
  serviceLevel: standard

```

Cloud-Identität für GKE

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
  storagePool: pool-premium1

```

Unterstützte Topologiekonfiguration

Trident ermöglicht die Bereitstellung von Volumes für Workloads basierend auf Regionen und Verfügbarkeitszonen. Der `supportedTopologies` Block in dieser Backend-Konfiguration dient dazu, eine Liste von Regionen und Zonen pro Backend bereitzustellen. Die hier angegebenen Regions- und Zonenwerte müssen mit den Regions- und Zonenwerten der Labels auf jedem Kubernetes-Clusterknoten übereinstimmen. Diese Regionen und Zonen stellen die Liste der zulässigen Werte dar, die in einer Speicherklasse angegeben werden können. Für Speicherklassen, die eine Teilmenge der im Backend bereitgestellten Regionen und Zonen enthalten, erstellt Trident Volumes in der genannten Region und Zone. Weitere Informationen finden Sie unter "[CSI-Topologie verwenden](#)".

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

Wie geht es weiter?

Nachdem Sie die Backend-Konfigurationsdatei erstellt haben, führen Sie folgenden Befehl aus:

```
kubectl create -f <backend-file>
```

Um zu überprüfen, ob das Backend erfolgreich erstellt wurde, führen Sie folgenden Befehl aus:

```
kubectl get tridentbackendconfig
```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS		
backend-tbc-gcnv	backend-tbc-gcnv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound Success		

Wenn die Backend-Erstellung fehlschlägt, stimmt etwas mit der Backend-Konfiguration nicht. Sie können das Backend mithilfe des folgenden beschreiben: `kubectl get tridentbackendconfig <backend-name>`
Um die Ursache zu ermitteln, führen Sie den folgenden Befehl aus oder überprüfen Sie die Protokolle:

```
tridentctl logs
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie das Backend löschen und den Befehl zum Erstellen erneut ausführen.

Speicherklassendefinitionen

Im Folgenden finden Sie eine grundlegende `StorageClass` Definition, die sich auf das oben genannte Backend bezieht.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

Beispieldefinitionen unter Verwendung der `parameter.selector` Feld:

Verwenden `parameter.selector` Sie können für jedes einzelne festlegen `StorageClass` Die "virtueller Pool" Das wird zum Hosten eines Volumes verwendet. Das Volumen wird die im gewählten Pool definierten Aspekte aufweisen.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes

```

Weitere Einzelheiten zu Speicherklassen finden Sie unter ["Erstellen einer Speicherklasse"](#) .

Beispieldefinitionen für SMB-Volumes

Verwenden `nasType` , `node-stage-secret-name` , Und `node-stage-secret-namespace` Sie können ein SMB-Volume angeben und die erforderlichen Active Directory-Anmeldeinformationen bereitstellen. Für das Node-Stage-Secret kann jeder Active Directory-Benutzer/jedes Passwort mit beliebigen/keinen Berechtigungen verwendet werden.

Grundkonfiguration im Standard-Namespace

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

Verwendung unterschiedlicher Geheimnisse pro Namensraum

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

Verwendung unterschiedlicher Geheimnisse pro Band

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```




`nasType: smb`` Filter für Pools, die SMB-Volumes unterstützen. ``nasType: nfs` oder `nasType: null` Filter für NFS-Pools.

PVC-Definitionsbeispiel

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

Um zu überprüfen, ob die PVC-Verbindung hergestellt ist, führen Sie folgenden Befehl aus:

```
kubectl get pvc gcnv-nfs-pvc
```

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES	STORAGECLASS	AGE	
gcnv-nfs-pvc	Bound	pvc-b00f2414-e229-40e6-9b16-ee03eb79a213	100Gi
RWX	gcnv-nfs-sc	1m	

Konfigurieren eines Cloud Volumes Service für das Google Cloud-Backend

Erfahren Sie anhand der bereitgestellten Beispielkonfigurationen, wie Sie den NetApp Cloud Volumes Service für Google Cloud als Backend für Ihre Trident -Installation konfigurieren.

Details zum Google Cloud-Treiber

Trident bietet die `gcp-cvs` Der Treiber soll mit dem Cluster kommunizieren. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	Lautstärkemodus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
gcp-cvs	NFS	Dateisystem	RWO, ROX, RWX, RWOP	nfs

Erfahren Sie mehr über die Trident Unterstützung für den Cloud Volumes Service für Google Cloud.

Trident kann Cloud Volumes Service Volumes in einem von zwei Formaten erstellen. "[Servicearten](#)":

- **CVS-Performance:** Der standardmäßige Trident Diensttyp. Dieser leistungsoptimierte Servicetyp eignet sich am besten für Produktionsworkloads, bei denen Leistung wichtig ist. Der Servicetyp CVS-Performance ist eine Hardwareoption, die Volumes mit einer Mindestgröße von 100 GiB unterstützt. Sie können eines auswählen **"drei Serviceebenen"** :
 - standard
 - premium
 - extreme
- **CVS:** Der CVS-Diensttyp bietet eine hohe zonale Verfügbarkeit bei begrenzter bis mäßiger Leistungsfähigkeit. Der CVS-Diensttyp ist eine Softwareoption, die Speicherpools nutzt, um Datenmengen ab 1 GiB zu unterstützen. Der Speicherpool kann bis zu 50 Volumes enthalten, wobei sich alle Volumes die Kapazität und Leistung des Pools teilen. Sie können eines auswählen **"zwei Serviceebenen"** :
 - standardsw
 - zoneredundantstandardsw

Was du brauchst

Um die **"Cloud Volumes Service für Google Cloud"** Im Backend benötigen Sie Folgendes:

- Ein mit dem NetApp Cloud Volumes Service konfiguriertes Google Cloud-Konto
- Projektnummer Ihres Google Cloud-Kontos
- Google Cloud-Dienstkonto mit dem `netappcloudvolumes.admin` Rolle
- API-Schlüsseldatei für Ihr Cloud Volumes Service -Konto

Backend-Konfigurationsoptionen

Jedes Backend stellt Volumes in einer einzelnen Google Cloud-Region bereit. Um Volumes in anderen Regionen zu erstellen, können Sie zusätzliche Backends definieren.

Parameter	Beschreibung	Standard
version		Immer 1
storageDriverName	Name des Speichertreibers	"gcp-cvs"
backendName	Benutzerdefinierter Name oder das Speicher-Backend	Fahrername + "_" + Teil des API-Schlüssels
storageClass	Optionaler Parameter zur Angabe des CVS-Diensttyps. Verwenden <code>software</code> um den CVS-Diensttyp auszuwählen. Andernfalls geht Trident vom Diensttyp CVS-Performance aus.(<code>hardware</code>).	
storagePools	Nur CVS-Dienstleistungstyp. Optionaler Parameter zur Angabe von Speicherpools für die Volume-Erstellung.	
projectNumber	Google Cloud-Konto-Projektnummer. Den Wert finden Sie auf der Startseite des Google Cloud-Portals.	

Parameter	Beschreibung	Standard
hostProjectNumber	Erforderlich bei Verwendung eines gemeinsam genutzten VPC-Netzwerks. In diesem Szenario <code>projectNumber</code> ist das Serviceprojekt, und <code>hostProjectNumber</code> ist das Host-Projekt.	
apiRegion	Die Google Cloud-Region, in der Trident Cloud Volumes Service Volumes erstellt. Beim Erstellen regionsübergreifender Kubernetes-Cluster werden Volumes, die in einem <code>apiRegion</code> kann in Workloads verwendet werden, die auf Knoten in mehreren Google Cloud-Regionen geplant sind. Für den Verkehr über Regionen hinweg fallen zusätzliche Kosten an.	
apiKey	API-Schlüssel für das Google Cloud-Dienstkonto mit dem <code>netappcloudvolumes.admin</code> Rolle. Sie enthält den JSON-formatierten Inhalt der privaten Schlüsseldatei eines Google Cloud-Dienstkontos (wörtlich in die Backend-Konfigurationsdatei kopiert).	
proxyURL	Proxy-URL, falls ein Proxy-Server zur Verbindung mit dem CVS-Konto erforderlich ist. Der Proxy-Server kann entweder ein HTTP-Proxy oder ein HTTPS-Proxy sein. Bei einem HTTPS-Proxy wird die Zertifikatsvalidierung übersprungen, um die Verwendung selbstsignierter Zertifikate auf dem Proxy-Server zu ermöglichen. Proxyserver mit aktivierter Authentifizierung werden nicht unterstützt.	
nfsMountOptions	Feingranulare Steuerung der NFS-Mount-Optionen.	"nfsvers=3"
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet.	"" (wird nicht standardmäßig erzwungen)
serviceLevel	Die CVS-Performance- oder CVS-Serviceebene für neue Volumina. Die CVS-Performance-Werte sind <code>standard</code> , <code>premium</code> , oder <code>extreme</code> . CVS-Werte sind <code>standardsw</code> oder <code>zoneredundantstandardsw</code> .	Die Standardeinstellung für CVS-Performance ist "Standard". Der CVS-Standardwert ist "standardsw".
network	Das Google Cloud-Netzwerk wird für Cloud Volumes Service Volumes verwendet.	"Standard"
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, <code>\{"api":false, "method":true\}</code> . Verwenden Sie diese Funktion nur, wenn Sie eine Fehlerbehebung durchführen und einen detaillierten Protokollauszug benötigen.	null

Parameter	Beschreibung	Standard
allowedTopologies	Um regionsübergreifenden Zugriff zu ermöglichen, muss Ihre StorageClass-Definition für allowedTopologies muss alle Regionen umfassen. Zum Beispiel: <ul style="list-style-type: none"> - key: topology.kubernetes.io/region values: - us-east1 - europe-west1 	

Volumenbereitstellungsoptionen

Sie können die Standard-Volume-Bereitstellung in der `defaults` Abschnitt der Konfigurationsdatei.

Parameter	Beschreibung	Standard
exportRule	Die Ausführbestimmungen für neue Bände. Es muss sich um eine durch Kommas getrennte Liste beliebiger Kombinationen von IPv4-Adressen oder IPv4-Subnetzen in CIDR-Notation handeln.	"0.0.0.0/0"
snapshotDir	Zugang zu <code>.snapshot</code> Verzeichnis	"FALSCH"
snapshotReserve	Prozentsatz des für Snapshots reservierten Speichervolumens	"" (CVS-Standardwert 0 akzeptieren)
size	Der Umfang der neuen Bände. Die Mindestgröße für CVS-Performance beträgt 100 GiB. Die Mindestgröße für CVS beträgt 1 GiB.	Der CVS-Performance-Diensttyp ist standardmäßig auf „100 GiB“ eingestellt. Der CVS-Diensttyp legt keinen Standardwert fest, erfordert aber ein Minimum von 1 GiB.

Beispiele für CVS-Performance-Diensttypen

Die folgenden Beispiele enthalten Beispielkonfigurationen für den Diensttyp CVS-Performance.

Beispiel 1: Minimale Konfiguration

Dies ist die minimale Backend-Konfiguration unter Verwendung des Standard-CVS-Performance-Diensttyps mit dem Standard-Dienstlevel „standard“.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: "012345678901"
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: <id_value>
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: "123456789012345678901"
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
```

Beispiel 2: Konfiguration auf Serviceebene

Dieses Beispiel veranschaulicht die Backend-Konfigurationsoptionen, einschließlich Servicelevel und Volumenstandardeinstellungen.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
proxyURL: http://proxy-server-hostname/
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 10Ti
serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '5'
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  size: 5Ti
```

Beispiel 3: Konfiguration eines virtuellen Pools

Dieses Beispiel verwendet `storage` virtuelle Pools und die `StorageClasses` die sich auf sie beziehen. Siehe [Speicherklassendefinitionen](#) um zu sehen, wie die Speicherklassen definiert wurden.

Hier werden spezifische Standardwerte für alle virtuellen Pools festgelegt, die die `snapshotReserve` bei 5 % und der `exportRule` zu 0.0.0.0/0. Die virtuellen Pools sind definiert in der `storage` Abschnitt. Jeder einzelne virtuelle Pool definiert seine eigenen `serviceLevel` und einige Pools überschreiben die Standardwerte. Virtuelle Poolbezeichnungen wurden verwendet, um die Pools anhand folgender Kriterien zu unterscheiden: `performance` Und `protection`.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
region: us-west2
storage:
- labels:
  performance: extreme
  protection: extra
  serviceLevel: extreme
defaults:
```

```

    snapshotDir: 'true'
    snapshotReserve: '10'
    exportRule: 10.0.0.0/24
- labels:
    performance: extreme
    protection: standard
    serviceLevel: extreme
- labels:
    performance: premium
    protection: extra
    serviceLevel: premium
defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
- labels:
    performance: premium
    protection: standard
    serviceLevel: premium
- labels:
    performance: standard
    serviceLevel: standard

```

Speicherklassendefinitionen

Die folgenden StorageClass-Definitionen gelten für das Konfigurationsbeispiel des virtuellen Pools. Verwenden `parameters.selector` Sie können für jede StorageClass den virtuellen Pool angeben, der zum Hosten eines Volumes verwendet wird. Das Volumen wird die im gewählten Pool definierten Aspekte aufweisen.

Beispiel für eine Speicherklasse

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme; protection=extra
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-standard-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=standard
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=extra
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=standard
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-standard
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
```

```

allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: protection=extra
allowVolumeExpansion: true

```

- Die erste Speicherklasse(`cvs-extreme-extra-protection`) wird dem ersten virtuellen Pool zugeordnet. Dies ist der einzige Pool, der extreme Leistung mit einer Momentaufnahme-Reserve von 10% bietet.
- Die letzte Speicherklasse(`cvs-extra-protection`) kennzeichnet jeden Speicherpool, der eine Snapshot-Reserve von 10% bereitstellt. Trident entscheidet, welcher virtuelle Pool ausgewählt wird und stellt sicher, dass die Snapshot-Reservierungsanforderung erfüllt wird.

Beispiele für CVS-Diensttypen

Die folgenden Beispiele enthalten Beispielkonfigurationen für den CVS-Diensttyp.

Beispiel 1: Minimale Konfiguration

Dies ist die minimale Backend-Konfiguration mit `storageClass` um den CVS-Diensttyp und den Standardwert anzugeben `standardsw` Servicelevel.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
storageClass: software
apiRegion: us-east4
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
serviceLevel: standardsw
```

Beispiel 2: Konfiguration des Speicherpools

Diese Beispielkonfiguration für das Backend verwendet `storagePools` einen Speicherpool konfigurieren.

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@cloud-native-
data.iam.gserviceaccount.com
  client_id: '107071413297115343396'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

Wie geht es weiter?

Nachdem Sie die Backend-Konfigurationsdatei erstellt haben, führen Sie folgenden Befehl aus:

```
tridentctl create backend -f <backend-file>
```

Wenn die Backend-Erstellung fehlschlägt, stimmt etwas mit der Backend-Konfiguration nicht. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Befehl zum Erstellen erneut ausführen.

Konfigurieren Sie ein NetApp HCI oder SolidFire Backend

Erfahren Sie, wie Sie ein Element-Backend mit Ihrer Trident -Installation erstellen und verwenden.

Element-Treiberdetails

Trident bietet die `solidfire-san` Speichertreiber zur Kommunikation mit dem Cluster. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Der `solidfire-san` Der Speichertreiber unterstützt die Volume-Modi *file* und *block*. Für die `Filesystem` Im Modus `volumeMode` erstellt Trident ein Volume und ein Dateisystem. Der Dateisystemtyp wird durch die `StorageClass` festgelegt.

Treiber	Protokoll	Lautstärkemode	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
<code>solidfire-san</code>	iSCSI	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem. Rohblockgerät.
<code>solidfire-san</code>	iSCSI	Dateisystem	RWO, RWOP	<code>xfs</code> , <code>ext3</code> , <code>ext4</code>

Bevor Sie beginnen

Bevor Sie ein Element-Backend erstellen, benötigen Sie Folgendes.

- Ein unterstütztes Speichersystem, auf dem die Element-Software läuft.
- Anmeldeinformationen für einen NetApp HCI/ SolidFire Cluster-Administrator oder Mandantenbenutzer, der Volumes verwalten kann.
- Auf allen Ihren Kubernetes-Worker-Knoten sollten die entsprechenden iSCSI-Tools installiert sein. Siehe "[Informationen zur Vorbereitung der Worker-Knoten](#)".

Backend-Konfigurationsoptionen

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
<code>version</code>		Immer 1
<code>storageDriverName</code>	Name des Speichertreibers	Immer " <code>solidfire-san</code> "
<code>backendName</code>	Benutzerdefinierter Name oder das Speicher-Backend	" <code>solidfire_</code> " + Speicher (iSCSI) IP-Adresse

Parameter	Beschreibung	Standard
Endpoint	MVIP für den SolidFire -Cluster mit Mandantenanmeldeinformationen	
SVIP	Speicher (iSCSI) IP-Adresse und Port	
labels	Satz beliebiger, im JSON-Format vorliegender Bezeichnungen, die auf Datenträger angewendet werden sollen.	""
TenantName	Zu verwendender Mandantenname (wird erstellt, falls nicht gefunden)	
InitiatorIFace	iSCSI-Datenverkehr auf eine bestimmte Hostschnittstelle beschränken	"Standard"
UseCHAP	Verwenden Sie CHAP zur Authentifizierung von iSCSI. Trident verwendet CHAP.	true
AccessGroups	Liste der zu verwendenden Zugriffsgruppen-IDs	Findet die ID einer Zugriffsgruppe mit dem Namen "trident"
Types	QoS-Spezifikationen	
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet.	"" (wird nicht standardmäßig erzwungen)
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel: {"api":false, "method":true}	null



Nicht verwenden `debugTraceFlags` Es sei denn, Sie befinden sich in der Fehlersuche und benötigen einen detaillierten Protokollauszug.

Beispiel 1: Backend-Konfiguration für `solidfire-san` Treiber mit drei Lautstärketypen

Dieses Beispiel zeigt eine Backend-Datei, die CHAP-Authentifizierung verwendet und drei Volumentypen mit spezifischen QoS-Garantien modelliert. Höchstwahrscheinlich würden Sie dann Speicherklassen definieren, um jede dieser Klassen mithilfe der folgenden Methode zu nutzen: `IOPS` Speicherklassenparameter.

```

---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: <svip>:3260
TenantName: <tenant>
labels:
  k8scluster: dev1
  backend: dev1-element-cluster
UseCHAP: true
Types:
- Type: Bronze
  Qos:
    minIOPS: 1000
    maxIOPS: 2000
    burstIOPS: 4000
- Type: Silver
  Qos:
    minIOPS: 4000
    maxIOPS: 6000
    burstIOPS: 8000
- Type: Gold
  Qos:
    minIOPS: 6000
    maxIOPS: 8000
    burstIOPS: 10000

```

Beispiel 2: Backend- und Speicherklassenkonfiguration für solidfire-san Fahrer mit virtuellen Pools

Dieses Beispiel zeigt die Backend-Definitionsdatei, die mit virtuellen Pools konfiguriert ist, sowie StorageClasses, die auf diese verweisen.

Trident kopiert die auf einem Speicherpool vorhandenen Labels beim Provisioning auf die Backend-Speicher-LUN. Zur Vereinfachung können Speicheradministratoren Bezeichnungen pro virtuellem Pool definieren und Volumes nach Bezeichnung gruppieren.

In der unten gezeigten Beispiel-Backend-Definitionsdatei sind spezifische Standardwerte für alle Speicherpools festgelegt, die die `type` bei Silver. Die virtuellen Pools sind definiert in der `storage` Abschnitt. In diesem Beispiel legen einige Speicherpools ihren eigenen Typ fest, und einige Pools überschreiben die oben festgelegten Standardwerte.

```

---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: <svip>:3260

```

```

TenantName: <tenant>
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
type: Silver
labels:
  store: solidfire
  k8scluster: dev-1-cluster
region: us-east-1
storage:
  - labels:
      performance: gold
      cost: "4"
    zone: us-east-1a
    type: Gold
  - labels:
      performance: silver
      cost: "3"
    zone: us-east-1b
    type: Silver
  - labels:
      performance: bronze
      cost: "2"
    zone: us-east-1c
    type: Bronze
  - labels:
      performance: silver
      cost: "1"
    zone: us-east-1d

```

Die folgenden StorageClass-Definitionen beziehen sich auf die oben genannten virtuellen Pools. Verwenden des parameters.selector Im Feld „StorageClass“ wird jeweils angegeben, welcher virtuelle Pool (oder

welche virtuellen Pools) zum Hosten eines Volumes verwendet werden kann. Das Volumen wird die im gewählten virtuellen Pool definierten Aspekte aufweisen.

Die erste Speicherklasse(solidfire-gold-four) wird dem ersten virtuellen Pool zugeordnet. Dies ist der einzige Pool, der Gold-Leistung mit einem bietet Volume Type QoS aus Gold. Die letzte Speicherklasse(solidfire-silver) nennt jeden Speicherpool, der eine Silber-Performance bietet. Trident entscheidet, welcher virtuelle Pool ausgewählt wird und stellt sicher, dass die Speichieranforderungen erfüllt werden.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-gold-four
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold; cost=4
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-three
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=3
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-bronze-two
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze; cost=2
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-one
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=1
  fsType: ext4
```

```

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
  fsType: ext4

```

Weitere Informationen

- ["Volumenzugriffsgruppen"](#)

ONTAP SAN-Treiber

ONTAP SAN-Treiberübersicht

Erfahren Sie mehr über die Konfiguration eines ONTAP Backends mit ONTAP und Cloud Volumes ONTAP SAN-Treibern.

ONTAP SAN-Treiberdetails

Trident stellt die folgenden SAN-Speichertreiber zur Verfügung, um mit dem ONTAP Cluster zu kommunizieren. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	Lautstärke modus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
ontap-san	iSCSI SCSI über FC	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem; rohes Blockgerät
ontap-san	iSCSI SCSI über FC	Dateisystem	RWO, RWOP ROX und RWX sind im Dateisystem-Volume- Modus nicht verfügbar.	xfs, ext3 , ext4
ontap-san	NVMe/TCP Siehe Weiter e Überlegungen zu NVMe/TCP .	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem; rohes Blockgerät

Treiber	Protokoll	Lautstärke modus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
ontap-san	NVMe/TCP Siehe Weiter e Überlegungen zu NVMe/TCP .	Dateisystem	RWO, RWOP ROX und RWX sind im Dateisystem-Volume- Modus nicht verfügbar.	xfs, ext3 , ext4
ontap-san-economy	iSCSI	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem; rohes Blockgerät
ontap-san-economy	iSCSI	Dateisystem	RWO, RWOP ROX und RWX sind im Dateisystem-Volume- Modus nicht verfügbar.	xfs, ext3 , ext4



- Verwenden `ontap-san-economy` nur wenn die Anzahl der dauerhaften Speichernutzungen voraussichtlich höher sein wird als "[Unterstützte ONTAP Lautstärkebegrenzungen](#)".
- Verwenden `ontap-nas-economy` nur wenn die Anzahl der dauerhaften Speichernutzungen voraussichtlich höher sein wird als "[Unterstützte ONTAP Lautstärkebegrenzungen](#)" und die `ontap-san-economy` Der Treiber kann nicht verwendet werden.
- Nicht verwenden `ontap-nas-economy` wenn Sie mit einem Bedarf an Datenschutz, Notfallwiederherstellung oder Mobilität rechnen.
- NetApp empfiehlt die Verwendung von Flexvol Autogrow nicht in allen ONTAP -Treibern, außer `ontap-san`. Als Ausweichlösung unterstützt Trident die Verwendung von Snapshot-Reserven und skaliert Flexvol-Volumes entsprechend.

Benutzerberechtigungen

Trident wird voraussichtlich entweder als ONTAP oder SVM-Administrator ausgeführt, typischerweise unter Verwendung von `admin` Clusterbenutzer oder ein `vsadmin` SVM-Benutzer oder ein Benutzer mit einem anderen Namen, der die gleiche Rolle hat. Bei Amazon FSx for NetApp ONTAP Bereitstellungen erwartet Trident , dass es entweder als ONTAP oder SVM-Administrator ausgeführt wird und den Cluster nutzt. `fsxadmin` Benutzer oder ein `vsadmin` SVM-Benutzer oder ein Benutzer mit einem anderen Namen, der die gleiche Rolle hat. Der `fsxadmin` Der Benutzer ist ein eingeschränkter Ersatz für den Cluster-Administratorbenutzer.



Wenn Sie die `limitAggregateUsage` Für diesen Parameter sind Cluster-Administratorrechte erforderlich. Bei der Verwendung von Amazon FSx for NetApp ONTAP mit Trident `limitAggregateUsage` Der Parameter funktioniert nicht mit dem `vsadmin` Und `fsxadmin` Benutzerkonten. Die Konfiguration schlägt fehl, wenn Sie diesen Parameter angeben.

Es ist zwar möglich, innerhalb von ONTAP eine restriktivere Rolle zu erstellen, die ein Trident -Treiber

verwenden kann, wir empfehlen dies jedoch nicht. Die meisten neuen Versionen von Trident werden zusätzliche APIs aufrufen, die berücksichtigt werden müssen, was Aktualisierungen schwierig und fehleranfällig macht.

Weitere Überlegungen zu NVMe/TCP

Trident unterstützt das NVMe-Protokoll (Non-Volatile Memory Express) mithilfe des `ontap-san` Fahrers einschließlich:

- IPv6
- Snapshots und Klone von NVMe-Volumes
- Ändern der Größe eines NVMe-Volumes
- Importieren eines NVMe-Volumes, das außerhalb von Trident erstellt wurde, damit sein Lebenszyklus von Trident verwaltet werden kann.
- NVMe-natives Multipathing
- Geordnetes oder ungeordnetes Herunterfahren der K8s-Knoten (24.06)

Trident unterstützt Folgendes nicht:

- DH-HMAC-CHAP, das nativ von NVMe unterstützt wird
- Gerätemapper (DM) Multipathing
- LUKS-Verschlüsselung



NVMe wird nur mit ONTAP REST APIs unterstützt und nicht mit ONTAPI (ZAPI).

Bereiten Sie die Konfiguration des Backends mit ONTAP SAN-Treibern vor.

Machen Sie sich mit den Anforderungen und Authentifizierungsoptionen für die Konfiguration eines ONTAP -Backends mit ONTAP SAN-Treibern vertraut.

Anforderungen

Für alle ONTAP Backends verlangt Trident , dass mindestens ein Aggregat dem SVM zugewiesen wird.



"ASA r2-Systeme" Sie unterscheiden sich von anderen ONTAP Systemen (ASA, AFF und FAS) in der Implementierung ihrer Speicherschicht. In ASA r2-Systemen werden Speicherverfügbarkeitszonen anstelle von Aggregaten verwendet. Siehe **"Das"** Wissensdatenbankartikel zur Zuordnung von Aggregaten zu SVMs in ASA r2-Systemen.

Denken Sie daran, dass Sie auch mehrere Treiber gleichzeitig ausführen und Speicherklassen erstellen können, die auf den einen oder anderen Treiber verweisen. Beispielsweise könnten Sie Folgendes konfigurieren: `san-dev` Klasse, die die `ontap-san` Fahrer und ein `san-default` Klasse, die die `ontap-san-economy` eins.

Auf allen Ihren Kubernetes-Worker-Knoten müssen die entsprechenden iSCSI-Tools installiert sein. Siehe **"Bereiten Sie den Worker-Knoten vor."** für Details.

Authentifizieren Sie das ONTAP Backend

Trident bietet zwei Modi zur Authentifizierung eines ONTAP Backends.

- Anmeldeinformationsbasiert: Benutzername und Passwort eines ONTAP Benutzers mit den erforderlichen Berechtigungen. Es wird empfohlen, eine vordefinierte Sicherheitsanmelderolle zu verwenden, wie zum Beispiel `admin` oder `vsadmin` um maximale Kompatibilität mit ONTAP Versionen zu gewährleisten.
- Zertifikatsbasiert: Trident kann auch mit einem ONTAP Cluster über ein auf dem Backend installiertes Zertifikat kommunizieren. Hierbei müssen in der Backend-Definition Base64-kodierte Werte des Client-Zertifikats, des Schlüssels und des vertrauenswürdigen CA-Zertifikats (falls verwendet, empfohlen) enthalten sein.

Sie können bestehende Backends aktualisieren, um zwischen anmeldeinformationsbasierten und zertifikatsbasierten Methoden zu wechseln. Es wird jedoch jeweils nur eine Authentifizierungsmethode unterstützt. Um zu einer anderen Authentifizierungsmethode zu wechseln, müssen Sie die bestehende Methode aus der Backend-Konfiguration entfernen.



Wenn Sie versuchen, **sowohl Anmeldeinformationen als auch Zertifikate** anzugeben, schlägt die Backend-Erstellung mit der Fehlermeldung fehl, dass in der Konfigurationsdatei mehr als eine Authentifizierungsmethode angegeben wurde.

Aktivieren Sie die anmeldeinformationsbasierte Authentifizierung

Trident benötigt die Anmeldeinformationen eines SVM-/Cluster-Administrators, um mit dem ONTAP Backend zu kommunizieren. Es wird empfohlen, standardisierte, vordefinierte Rollen zu verwenden, wie zum Beispiel `admin` oder `vsadmin`. Dies gewährleistet die Vorwärtskompatibilität mit zukünftigen ONTAP Versionen, die möglicherweise Feature-APIs zur Verwendung durch zukünftige Trident Versionen bereitstellen. Eine benutzerdefinierte Sicherheitsanmelderolle kann erstellt und mit Trident verwendet werden, dies wird jedoch nicht empfohlen.

Eine beispielhafte Backend-Definition sieht folgendermaßen aus:

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

Beachten Sie, dass die Backend-Definition der einzige Ort ist, an dem die Anmeldeinformationen im Klartext gespeichert werden. Nach der Erstellung des Backends werden Benutzernamen und Passwörter mit Base64 kodiert und als Kubernetes-Secrets gespeichert. Die Erstellung oder Aktualisierung eines Backends ist der einzige Schritt, der Kenntnisse der Zugangsdaten erfordert. Daher handelt es sich um eine ausschließlich für Administratoren zulässige Operation, die vom Kubernetes-/Speicheradministrator durchgeführt werden muss.

Zertifikatbasierte Authentifizierung aktivieren

Neue und bestehende Backends können ein Zertifikat verwenden und mit dem ONTAP Backend kommunizieren. Für die Backend-Definition werden drei Parameter benötigt.

- `clientCertificate`: Base64-kodierter Wert des Clientzertifikats.
- `clientPrivateKey`: Base64-kodierter Wert des zugehörigen privaten Schlüssels.
- `trustedCACertificate`: Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Bei Verwendung einer vertrauenswürdigen Zertifizierungsstelle muss dieser Parameter angegeben werden. Dies kann ignoriert werden, wenn keine vertrauenswürdige Zertifizierungsstelle verwendet wird.

Ein typischer Arbeitsablauf umfasst die folgenden Schritte.

Schritte

1. Generieren Sie ein Clientzertifikat und einen Schlüssel. Beim Generieren muss der allgemeine Name (CN) auf den ONTAP Benutzer gesetzt werden, der sich authentifizieren soll.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. Fügen Sie dem ONTAP -Cluster ein vertrauenswürdiges CA-Zertifikat hinzu. Dies könnte bereits vom Speicheradministrator erledigt werden. Ignorieren, falls keine vertrauenswürdige Zertifizierungsstelle verwendet wird.

```
security certificate install -type server -cert-name <trusted-ca-cert-  
name> -vserver <vserver-name>  
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled  
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca  
<cert-authority>
```

3. Installieren Sie das Clientzertifikat und den Schlüssel (aus Schritt 1) auf dem ONTAP Cluster.

```
security certificate install -type client-ca -cert-name <certificate-  
name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. Bestätigen Sie, dass die ONTAP Sicherheitsanmeldungsrolle die folgenden Funktionen unterstützt: cert Authentifizierungsmethode.

```
security login create -user-or-group-name admin -application ontapi  
-authentication-method cert  
security login create -user-or-group-name admin -application http  
-authentication-method cert
```

5. Testen Sie die Authentifizierung mit dem generierten Zertifikat. Ersetzen Sie < ONTAP Management LIF> und <vserver name> durch die Management LIF IP-Adresse und den SVM-Namen.

```
curl -X POST -Lk https://<ONTAP-Management-  
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key  
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp  
xmlns="http://www.netapp.com/filer/admin" version="1.21"  
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Zertifikat, Schlüssel und vertrauenswürdiges CA-Zertifikat mit Base64 kodieren.

```
base64 -w 0 k8senv.pem >> cert_base64  
base64 -w 0 k8senv.key >> key_base64  
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Erstellen Sie ein Backend unter Verwendung der im vorherigen Schritt erhaltenen Werte.

```
cat cert-backend.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "trustedCACertificate": "QNFinfO...SiqOyN",
  "storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+-----+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                UUID                |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |         0 |
+-----+-----+-----+-----+
+-----+-----+
```

Aktualisieren Sie die Authentifizierungsmethoden oder ändern Sie die Anmeldeinformationen.

Sie können ein bestehendes Backend aktualisieren, um eine andere Authentifizierungsmethode zu verwenden oder um die Anmeldeinformationen zu ändern. Dies funktioniert in beide Richtungen: Backends, die Benutzername/Passwort verwenden, können auf die Verwendung von Zertifikaten umgestellt werden; Backends, die Zertifikate verwenden, können auf Benutzername/Passwort-basiert umgestellt werden. Dazu müssen Sie die bestehende Authentifizierungsmethode entfernen und die neue Authentifizierungsmethode hinzufügen. Verwenden Sie anschließend die aktualisierte Datei backend.json, die die erforderlichen Parameter enthält, um die Ausführung durchzuführen. `tridentctl backend update`.


```
cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "password",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident

+-----+-----+-----+
+-----+-----+
| NAME | STORAGE DRIVER | UUID |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online | 9 |
+-----+-----+-----+
+-----+-----+
```



Beim Ändern von Passwörtern muss der Speicheradministrator zuerst das Passwort für den Benutzer auf ONTAP aktualisieren. Anschließend erfolgt ein Backend-Update. Bei der Zertifikatsrotation können dem Benutzer mehrere Zertifikate hinzugefügt werden. Anschließend wird das Backend aktualisiert, um das neue Zertifikat zu verwenden. Danach kann das alte Zertifikat aus dem ONTAP Cluster gelöscht werden.

Durch die Aktualisierung des Backends wird der Zugriff auf bereits erstellte Volumes nicht beeinträchtigt, und auch später hergestellte Volume-Verbindungen werden nicht beeinträchtigt. Ein erfolgreiches Backend-Update zeigt an, dass Trident mit dem ONTAP -Backend kommunizieren und zukünftige Volumenoperationen bewältigen kann.

Erstellen einer benutzerdefinierten ONTAP Rolle für Trident

Sie können eine ONTAP Clusterrolle mit minimalen Berechtigungen erstellen, sodass Sie für Operationen in Trident nicht die ONTAP Administratorrolle verwenden müssen. Wenn Sie den Benutzernamen in einer Trident Backend-Konfiguration angeben, verwendet Trident die von Ihnen erstellte ONTAP Clusterrolle, um die Operationen durchzuführen.

Siehe "[Trident -Benutzerrollengenerator](#)" Weitere Informationen zum Erstellen benutzerdefinierter Trident -Rollen finden Sie hier.

Verwendung der ONTAP Befehlszeile

1. Erstellen Sie eine neue Rolle mit folgendem Befehl:

```
security login role create <role_name\> -cmddirname "command" -access all  
-vserver <svm_name\>
```

2. Erstellen Sie einen Benutzernamen für den Trident -Benutzer:

```
security login create -username <user_name\> -application ontapi  
-authmethod <password\> -role <name_of_role_in_step_1\> -vserver  
<svm_name\> -comment "user_description"
```

3. Ordnen Sie die Rolle dem Benutzer zu:

```
security login modify username <user_name\> -vserver <svm_name\> -role  
<role_name\> -application ontapi -application console -authmethod  
<password\>
```

Systemmanager verwenden

Führen Sie die folgenden Schritte im ONTAP System Manager aus:

1. Erstellen Sie eine benutzerdefinierte Rolle:

- a. Um eine benutzerdefinierte Rolle auf Clusterebene zu erstellen, wählen Sie **Cluster > Einstellungen**.

(Oder) Um eine benutzerdefinierte Rolle auf SVM-Ebene zu erstellen, wählen Sie **Speicher > Speicher-VMs > required svm > Einstellungen > Benutzer und Rollen**.

- b. Wählen Sie das Pfeilsymbol (→) neben **Benutzer und Rollen** aus.

- c. Wählen Sie unter **Rollen** die Option **+Hinzufügen**.

- d. Definieren Sie die Regeln für die Rolle und klicken Sie auf **Speichern**.

2. **Rolle dem Trident -Benutzer zuordnen:** + Führen Sie die folgenden Schritte auf der Seite **Benutzer und Rollen** aus:

- a. Wählen Sie unter **Benutzer** das Symbol **+** zum Hinzufügen aus.

- b. Wählen Sie den gewünschten Benutzernamen und anschließend eine Rolle im Dropdown-Menü für **Rolle** aus.

- c. Klicken Sie auf **Speichern**.

Weitere Informationen finden Sie auf den folgenden Seiten:

- ["Benutzerdefinierte Rollen für die Administration von ONTAP"](#) oder ["Benutzerdefinierte Rollen definieren"](#)
- ["Mit Rollen und Benutzern arbeiten"](#)

Authentifizieren Sie Verbindungen mit bidirektionalem CHAP

Trident kann iSCSI-Sitzungen mit bidirektionalem CHAP authentifizieren. `ontap-san` Und `ontap-san-economy` Fahrer. Dies erfordert die Aktivierung der `useCHAP` Option in Ihrer Backend-Definition. Wenn eingestellt auf `true` Trident konfiguriert die Standard-Initiator-Sicherheit der SVM auf bidirektionales CHAP

und legt den Benutzernamen und die Geheimnisse aus der Backend-Datei fest. NetApp empfiehlt die Verwendung von bidirektionalem CHAP zur Authentifizierung von Verbindungen. Siehe die folgende Beispielkonfiguration:

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
```



Der `useCHAP` Parameter ist eine boolesche Option, die nur einmal konfiguriert werden kann. Es ist standardmäßig auf „false“ eingestellt. Sobald Sie den Wert auf „true“ gesetzt haben, können Sie ihn nicht mehr auf „false“ setzen.

Zusätzlich zu `useCHAP=true`, Die `chapInitiatorSecret`, `chapTargetInitiatorSecret`, `chapTargetUsername`, Und `chapUsername` Die Felder müssen in der Backend-Definition enthalten sein. Die Geheimnisse können nach der Erstellung eines Backends durch Ausführen von geändert werden.
`tridentctl update`.

So funktioniert es

Durch die Einstellung `useCHAP` Wenn dies der Fall ist, weist der Speicheradministrator Trident an, CHAP auf dem Speicher-Backend zu konfigurieren. Hierzu gehört Folgendes:

- CHAP auf der SVM einrichten:
 - Wenn der Standard-Initiator-Sicherheitstyp der SVM „Keine“ ist (Standardeinstellung) **und** keine LUNs im Volume vorhanden sind, legt Trident den Standard-Sicherheitstyp auf „Keine“ fest. CHAP und fahren Sie mit der Konfiguration des CHAP-Initiators sowie des Zielbenutzernamens und der zugehörigen Geheimnisse fort.
 - Wenn die SVM LUNs enthält, wird Trident CHAP auf der SVM nicht aktivieren. Dadurch wird sichergestellt, dass der Zugriff auf LUNs, die bereits auf der SVM vorhanden sind, nicht eingeschränkt wird.
- Konfiguration des CHAP-Initiators und des Zielbenutzernamens sowie der Geheimnisse; diese Optionen müssen in der Backend-Konfiguration angegeben werden (wie oben gezeigt).

Nachdem das Backend erstellt wurde, erstellt Trident ein entsprechendes `tridentbackend` CRD speichert die CHAP-Geheimnisse und Benutzernamen als Kubernetes-Geheimnisse. Alle von Trident auf diesem Backend erstellten PVs werden über CHAP eingebunden und angehängt.

Rotieren Sie Anmeldeinformationen und aktualisieren Sie Backends

Sie können die CHAP-Zugangsdaten aktualisieren, indem Sie die CHAP-Parameter in der `backend.json` Datei. Dies erfordert eine Aktualisierung der CHAP-Geheimnisse und die Verwendung von `tridentctl update` Befehl, um diese Änderungen widerzuspiegeln.



Beim Aktualisieren der CHAP-Geheimnisse für ein Backend müssen Sie Folgendes verwenden: `tridentctl` um das Backend zu aktualisieren. Aktualisieren Sie die Anmeldeinformationen des Speicherclusters nicht über die ONTAP CLI oder den ONTAP System Manager, da Trident diese Änderungen nicht erkennen kann.

```
cat backend-san.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap_san_chap",
  "managementLIF": "192.168.0.135",
  "svm": "ontap_iscsi_svm",
  "useCHAP": true,
  "username": "vsadmin",
  "password": "password",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}
```

```
./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+-----+-----+
+-----+-----+
|  NAME          | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |      7 |
+-----+-----+-----+-----+
+-----+-----+
```

Bestehende Verbindungen bleiben unberührt; sie bleiben aktiv, wenn die Anmeldeinformationen von Trident auf der SVM aktualisiert werden. Neue Verbindungen verwenden die aktualisierten Zugangsdaten, bestehende Verbindungen bleiben weiterhin aktiv. Durch das Trennen und erneute Verbinden alter PV-Geräte werden diese mit den aktualisierten Zugangsdaten verwendet.

ONTAP SAN-Konfigurationsoptionen und Beispiele

Erfahren Sie, wie Sie ONTAP SAN-Treiber mit Ihrer Trident -Installation erstellen und

verwenden. Dieser Abschnitt enthält Beispiele für die Backend-Konfiguration und Details zur Zuordnung von Backends zu StorageClasses.

"[ASA r2-Systeme](#)" Sie unterscheiden sich von anderen ONTAP Systemen (ASA, AFF und FAS) in der Implementierung ihrer Speicherschicht. Diese Abweichungen wirken sich wie angegeben auf die Verwendung bestimmter Parameter aus. "[Erfahren Sie mehr über die Unterschiede zwischen ASA r2-Systemen und anderen ONTAP Systemen.](#)".




Nur die `ontap-san` Der Treiber (mit iSCSI- und NVMe/TCP-Protokollen) wird für ASA r2-Systeme unterstützt.


In der Trident Backend-Konfiguration müssen Sie nicht angeben, dass Ihr System ASA r2 ist. Wenn Sie auswählen `ontap-san` als die `storageDriverName` Trident erkennt automatisch das ASA r2- oder das herkömmliche ONTAP System. Einige Backend-Konfigurationsparameter sind für ASA r2-Systeme nicht anwendbar, wie in der folgenden Tabelle vermerkt.

Backend-Konfigurationsoptionen


Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
<code>version</code>		Immer 1
<code>storageDriverName</code>	Name des Speichertreibers	<code>ontap-san`</code> oder <code>`ontap-san-economy</code>
<code>backendName</code>	Benutzerdefinierter Name oder das Speicher-Backend	Fahrername + <code>"_"</code> + <code>dataLIF</code>
<code>managementLIF</code>	<p>IP-Adresse eines Cluster- oder SVM-Management-LIF.</p> <p>Es kann ein vollqualifizierter Domänenname (FQDN) angegeben werden.</p> <p>Kann so eingestellt werden, dass IPv6-Adressen verwendet werden, wenn Trident mit dem IPv6-Flag installiert wurde. IPv6-Adressen müssen in eckigen Klammern definiert werden, z. B. [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555] .</p> <p>Für einen nahtlosen MetroCluster Wechsel sieheMetroCluster Beispiel .</p> <div><p>Wenn Sie die Anmeldeinformationen „vsadmin“ verwenden, <code>managementLIF</code> muss die des SVM sein; bei Verwendung von "Admin"-Anmeldeinformationen, <code>managementLIF</code> muss die des Clusters sein.</p></div>	„10.0.0.1“, „[2001:1234:abcd::fefe]“

Parameter	Beschreibung	Standard
dataLIF	IP-Adresse des Protokolls LIF. Kann so eingestellt werden, dass IPv6-Adressen verwendet werden, wenn Trident mit dem IPv6-Flag installiert wurde. IPv6-Adressen müssen in eckigen Klammern definiert werden, z. B. [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555] . Nicht für iSCSI angeben. Trident Anwendungen " ONTAP Selective LUN Map " um die iSCSI LIFs zu ermitteln, die zum Aufbau einer Multipath-Sitzung benötigt werden. Es wird eine Warnung generiert, wenn dataLIF ist explizit definiert. Für Metrocluster auslassen. Siehe die MetroCluster Beispiel .	Abgeleitet durch die SVM
svm	Zu verwendende virtuelle Speichermaschine Für Metrocluster auslassen. Siehe die MetroCluster Beispiel .	Abgeleitet, wenn eine SVM managementLIF wird angegeben
useCHAP	Verwenden Sie CHAP zur Authentifizierung von iSCSI für ONTAP SAN-Treiber [Boolesch]. Auf einstellen true damit Trident bidirektionales CHAP als Standardauthentifizierung für die im Backend angegebene SVM konfiguriert und verwendet. Siehe " Bereiten Sie die Konfiguration des Backends mit ONTAP SAN-Treibern vor. " für Details. Nicht unterstützt für FCP oder NVMe/TCP.	false
chapInitiatorSecret	Geheimnis des CHAP-Initiators. Erforderlich, wenn useCHAP=true	""
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Datenträger angewendet werden sollen	""
chapTargetInitiatorSecret	Geheimnis des CHAP-Zielinitiators. Erforderlich, wenn useCHAP=true	""
chapUsername	Eingehender Benutzername. Erforderlich, wenn useCHAP=true	""
chapTargetUsername	Zielbenutzername. Erforderlich, wenn useCHAP=true	""
clientCertificate	Base64-kodierter Wert des Clientzertifikats. Wird für zertifikatsbasierte Authentifizierung verwendet	""
clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für zertifikatsbasierte Authentifizierung verwendet	""
trustedCACertificate	Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Optional. Wird für die zertifikatsbasierte Authentifizierung verwendet.	""

Parameter	Beschreibung	Standard
username	Für die Kommunikation mit dem ONTAP -Cluster ist ein Benutzername erforderlich. Wird für die auf Anmeldeinformationen basierende Authentifizierung verwendet. Informationen zur Active Directory-Authentifizierung finden Sie unter "Authentifizieren Sie Trident bei einem Backend-SVM mithilfe von Active Directory-Anmeldeinformationen" .	""
password	Für die Kommunikation mit dem ONTAP -Cluster ist ein Passwort erforderlich. Wird für die auf Anmeldeinformationen basierende Authentifizierung verwendet. Informationen zur Active Directory-Authentifizierung finden Sie unter "Authentifizieren Sie Trident bei einem Backend-SVM mithilfe von Active Directory-Anmeldeinformationen" .	""
svm	Zu verwendende virtuelle Speichermaschine	Abgeleitet, wenn eine SVM managementLIF wird angegeben
storagePrefix	Präfix, das beim Bereitstellen neuer Volumes in der SVM verwendet wird. Kann später nicht geändert werden. Um diesen Parameter zu aktualisieren, müssen Sie ein neues Backend erstellen.	trident
aggregate	<p>Aggregat für die Bereitstellung (optional; falls festgelegt, muss es der SVM zugewiesen werden). Für die <code>ontap-nas-flexgroup</code> Treiber, diese Option wird ignoriert. Falls kein Aggregat zugewiesen ist, kann jedes der verfügbaren Aggregate zur Bereitstellung eines FlexGroup Volumes verwendet werden.</p> <div>  <p>Wenn das Aggregat in SVM aktualisiert wird, wird es in Trident automatisch durch Abfrage von SVM aktualisiert, ohne dass der Trident Controller neu gestartet werden muss. Wenn Sie in Trident ein bestimmtes Aggregat zur Bereitstellung von Volumes konfiguriert haben und dieses Aggregat umbenannt oder aus der SVM verschoben wird, wechselt das Backend in Trident in den Fehlerzustand, während es das SVM-Aggregat abfragt. Sie müssen entweder das Aggregat in ein auf der SVM vorhandenes ändern oder es vollständig entfernen, um das Backend wieder online zu bringen.</p> </div> <p>Nicht für ASA r2-Systeme angeben.</p>	""

Parameter	Beschreibung	Standard
limitAggregateUsage	Die Bereitstellung schlägt fehl, wenn die Auslastung diesen Prozentsatz überschreitet. Wenn Sie ein Amazon FSx for NetApp ONTAP -Backend verwenden, geben Sie dies nicht an. limitAggregateUsage . Die bereitgestellten fsxadmin Und vsadmin enthalten nicht die erforderlichen Berechtigungen, um die aggregierte Nutzung abzurufen und sie mit Trident einzuschränken. Nicht für ASA r2-Systeme angeben.	"" (wird nicht standardmäßig erzwungen)
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet. Außerdem wird die maximale Größe der von ihm verwalteten Volumes für LUNs beschränkt.	"" (wird nicht standardmäßig erzwungen)
lunsPerFlexvol	Die maximale Anzahl an LUNs pro Flexvol muss im Bereich [50, 200] liegen.	100
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel: {"api":false, "method":true} Verwenden Sie dies nur, wenn Sie eine Fehlerbehebung durchführen und einen detaillierten Protokollauszug benötigen.	null

Parameter	Beschreibung	Standard
useREST	<p>Boolescher Parameter zur Verwendung von ONTAP REST-APIs.</p> <div> <p>`useREST` Wenn eingestellt auf `true` Trident verwendet ONTAP REST-APIs zur Kommunikation mit dem Backend; wenn eingestellt auf `false` Trident verwendet ONTAPI (ZAPI)-Aufrufe zur Kommunikation mit dem Backend. Diese Funktion erfordert ONTAP 9.11.1 und höher. Darüber hinaus muss die verwendete ONTAP Anmelderolle Zugriff auf die `ontapi` Anwendung. Dies wird durch die vordefinierte Bedingung erfüllt. `vsadmin` Und `cluster-admin` Rollen. Ab der Trident Version 24.06 und ONTAP 9.15.1 oder höher, `useREST` ist eingestellt auf `true` Standardmäßig; ändern `useREST` Zu `false` ONTAPI (ZAPI)-Aufrufe verwenden.</p> </div> <p>`useREST` ist vollständig für NVMe/TCP qualifiziert.</p> <div>  <p>NVMe wird nur mit ONTAP REST APIs unterstützt und nicht mit ONTAPI (ZAPI).</p> </div> <p>Falls angegeben, immer auf setzen <code>true</code> für ASA r2-Systeme.</p>	<p><code>true</code> für ONTAP 9.15.1 oder höher, andernfalls <code>false</code>.</p>
sanType	<p>Zur Auswahl verwenden <code>iscsi</code> für iSCSI, <code>nvme</code> für NVMe/TCP oder <code>fcp</code> für SCSI über Fibre Channel (FC).</p>	<p><code>iscsi</code> falls leer</p>

Parameter	Beschreibung	Standard
formatOptions	Verwenden formatOptions um Befehlszeilenargumente für die mkfs Befehl, der immer dann angewendet wird, wenn ein Datenträger formatiert wird. Dies ermöglicht es Ihnen, die Lautstärke nach Ihren Wünschen zu formatieren. Stellen Sie sicher, dass Sie die Formatoptionen analog zu den Optionen des Befehls mkfs angeben, jedoch ohne den Gerätepfad. Beispiel: "-E nodiscard" Unterstützt für ontap-san Und ontap-san-economy Treiber mit iSCSI-Protokoll. Zusätzlich wird dies für ASA r2-Systeme bei Verwendung der iSCSI- und NVMe/TCP-Protokolle unterstützt.	
limitVolumePoolSize	Maximal anforderbare FlexVol Größe bei Verwendung von LUNs im ontap-san-economy-Backend.	"" (wird nicht standardmäßig erzwungen)
denyNewVolumePools	Beschränkt ontap-san-economy Backends daran zu hindern, neue FlexVol -Volumes zu erstellen, die ihre LUNs enthalten. Für die Bereitstellung neuer PVs werden ausschließlich bereits vorhandene Flexvols verwendet.	

Empfehlungen zur Verwendung von formatOptions

Trident empfiehlt die folgende Option, um den Formatierungsprozess zu beschleunigen:

-E nodiscard:

- Blöcke sollten beim Erstellen des Dateisystems (mkfs) nicht verworfen werden (das anfängliche Verwerfen von Blöcken ist bei Solid-State-Geräten und dünn bereitgestellten Speichern sinnvoll). Dies ersetzt die veraltete Option "-K" und ist auf alle Dateisysteme (xfs, ext3 und ext4) anwendbar.

Authentifizieren Sie Trident bei einem Backend-SVM mithilfe von Active Directory-Anmeldeinformationen

Sie können Trident so konfigurieren, dass es sich mit Active Directory (AD)-Anmeldeinformationen bei einem Back-End-SVM authentifiziert. Bevor ein AD-Konto auf die SVM zugreifen kann, müssen Sie den AD-Domänencontrollerzugriff auf den Cluster oder die SVM konfigurieren. Für die Clusterverwaltung mit einem AD-Konto müssen Sie einen Domänentunnel erstellen. Siehe ["Konfigurieren des Active Directory-Domänencontrollerzugriffs in ONTAP"](#) für Details.

Schritte

1. Konfigurieren Sie die DNS-Einstellungen (Domain Name System) für eine Back-End-SVM:

```
vserver services dns create -vserver <svm_name> -dns-servers
<dns_server_ip1>,<dns_server_ip2>
```

2. Führen Sie den folgenden Befehl aus, um ein Computerkonto für die SVM in Active Directory zu erstellen:

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1
-domain demo.netapp.com
```

3. Verwenden Sie diesen Befehl, um einen AD-Benutzer oder eine AD-Gruppe zum Verwalten des Clusters oder SVM zu erstellen

```
security login create -vserver <svm_name> -user-or-group-name
<ad_user_or_group> -application <application> -authentication-method domain
-role vsadmin
```

4. Legen Sie in der Trident Backend-Konfigurationsdatei Folgendes fest: `username` Und `password` Parameter auf den AD-Benutzer- oder Gruppennamen bzw. das Kennwort.

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

Sie können die Standardbereitstellung mithilfe dieser Optionen steuern. `defaults` Abschnitt der Konfiguration. Ein Beispiel finden Sie in den folgenden Konfigurationsbeispielen.

Parameter	Beschreibung	Standard
<code>spaceAllocation</code>	Speicherplatzzuweisung für LUNs	"true" Falls angegeben, auf setzen true für ASA r2-Systeme.
<code>spaceReserve</code>	Platzreservierungsmodus; "keine" (dünn) oder "Volumen" (dick). Einstellen auf none für ASA r2-Systeme.	"keiner"
<code>snapshotPolicy</code>	Zu verwendende Snapshot-Richtlinie. Einstellen auf none für ASA r2-Systeme.	"keiner"
<code>qosPolicy</code>	Die QoS-Richtliniengruppe soll den erstellten Volumes zugewiesen werden. Wählen Sie pro Speicherpool/Backend entweder <code>qosPolicy</code> oder <code>adaptiveQosPolicy</code> . Die Verwendung von QoS-Richtliniengruppen mit Trident erfordert ONTAP 9.8 oder höher. Sie sollten eine nicht gemeinsam genutzte QoS-Richtliniengruppe verwenden und sicherstellen, dass die Richtliniengruppe auf jeden einzelnen Bestandteil angewendet wird. Eine gemeinsam genutzte QoS-Richtliniengruppe setzt die Obergrenze für den Gesamtdurchsatz aller Workloads durch.	""
<code>adaptiveQosPolicy</code>	Adaptive QoS-Richtliniengruppe, die den erstellten Volumes zugewiesen werden soll. Wählen Sie pro Speicherpool/Backend entweder <code>qosPolicy</code> oder <code>adaptiveQosPolicy</code> aus.	""
<code>snapshotReserve</code>	Prozentsatz des für Snapshots reservierten Speichervolumens. Nicht für ASA r2-Systeme angeben.	"0" wenn <code>snapshotPolicy</code> ist "keine", ansonsten ""
<code>splitOnClone</code>	Beim Erstellen eines Klons diesen von seinem Elternklon trennen	"FALSCH"

Parameter	Beschreibung	Standard
encryption	Aktivieren Sie die NetApp Volumeverschlüsselung (NVE) auf dem neuen Volume; Standardwert ist <code>false</code> . Um diese Option nutzen zu können, muss NVE auf dem Cluster lizenziert und aktiviert sein. Wenn NAE im Backend aktiviert ist, wird jedes in Trident bereitgestellte Volume NAE-fähig sein. Weitere Informationen finden Sie unter: "Wie Trident mit NVE und NAE zusammenarbeitet" .	<code>"false"</code> Falls angegeben, auf <code>setzen</code> . <code>true</code> für ASA r2-Systeme.
luksEncryption	LUKS-Verschlüsselung aktivieren. Siehe "Verwenden Sie Linux Unified Key Setup (LUKS)." .	Einstellen auf <code>false</code> für ASA r2-Systeme.
tieringPolicy	Tiering-Richtlinie auf "keine" setzen Für ASA r2-Systeme nicht angeben .	
nameTemplate	Vorlage zum Erstellen benutzerdefinierter Datenträgernamen.	<code>""</code>

Beispiele für die Volumenbereitstellung

Hier ist ein Beispiel mit vordefinierten Standardwerten:

```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'

```



Für alle mit der `ontap-san` Der Trident -Treiber erweitert die FlexVol -Kapazität um zusätzliche 10 Prozent, um die LUN-Metadaten aufzunehmen. Die LUN wird mit der exakten Größe bereitgestellt, die der Benutzer im PVC anfordert. Trident erhöht den FlexVol um 10 Prozent (wird in ONTAP als verfügbare Größe angezeigt). Die Nutzer erhalten nun die von ihnen angeforderte nutzbare Speicherkapazität. Diese Änderung verhindert auch, dass LUNs schreibgeschützt werden, es sei denn, der verfügbare Speicherplatz wird vollständig genutzt. Dies gilt nicht für `ontap-san-economy`.

Für Backends, die definieren `snapshotReserve` Trident berechnet die Größe von Volumina wie folgt:

```
Total volume size = [(PVC requested size) / (1 - (snapshotReserve percentage) / 100)] * 1.1
```

Die 1.1 sind die zusätzlichen 10 Prozent, die Trident zum FlexVol hinzufügt, um die LUN-Metadaten unterzubringen. Für `snapshotReserve` = 5% und PVC-Anforderung = 5 GiB, die Gesamtvolumengröße beträgt 5,79 GiB und die verfügbare Größe beträgt 5,5 GiB. Der `volume show` Der Befehl sollte ähnliche Ergebnisse wie in diesem Beispiel liefern:

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
		_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4	online	RW	10GB	5.00GB	0%
		_pvc_e42ec6fe_3baa_4af6_996d_134adbbb8e6d	online	RW	5.79GB	5.50GB	0%
		_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba	online	RW	1GB	511.8MB	0%

3 entries were displayed.

Aktuell ist die Größenänderung die einzige Möglichkeit, die neue Berechnung für ein bestehendes Volumen zu nutzen.

Beispiele für minimale Konfigurationen

Die folgenden Beispiele zeigen Basiskonfigurationen, bei denen die meisten Parameter auf Standardwerte eingestellt bleiben. Dies ist die einfachste Möglichkeit, ein Backend zu definieren.



Wenn Sie Amazon FSx auf NetApp ONTAP mit Trident verwenden, empfiehlt NetApp, für LIFs DNS-Namen anstelle von IP-Adressen anzugeben.

ONTAP SAN-Beispiel

Dies ist eine Basiskonfiguration unter Verwendung der `ontap-san` Treiber.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

MetroCluster Beispiel

Sie können das Backend so konfigurieren, dass eine manuelle Aktualisierung der Backend-Definition nach einem Switchover und Switchback vermieden wird. ["SVM-Replikation und -Wiederherstellung"](#) .

Für einen nahtlosen Übergang und Rückwechsel geben Sie die SVM wie folgt an: `managementLIF` und lassen Sie die `svm` Parameter. Beispiel:

```
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

ONTAP SAN Wirtschaftsbeispiel

```
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

Beispiel für zertifikatsbasierte Authentifizierung

In diesem Beispiel für eine einfache Konfiguration `clientCertificate`, `clientPrivateKey`, Und `trustedCACertificate` (optional, falls eine vertrauenswürdige Zertifizierungsstelle verwendet wird) werden in `backend.json` und nehmen Sie die Base64-kodierten Werte des Clientzertifikats, des privaten Schlüssels bzw. des vertrauenswürdigen CA-Zertifikats.

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

Beispiele für bidirektionale CHAP-Programme

Diese Beispiele erstellen ein Backend mit `useCHAP` eingestellt auf `true`.

ONTAP SAN CHAP Beispiel

```
---  
version: 1  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_iscsi  
labels:  
  k8scluster: test-cluster-1  
  backend: testcluster1-sanbackend  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSD6cNwxyz  
username: vsadmin  
password: <password>
```

ONTAP SAN Wirtschaft CHAP Beispiel

```
---  
version: 1  
storageDriverName: ontap-san-economy  
managementLIF: 10.0.0.1  
svm: svm_iscsi_eco  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSD6cNwxyz  
username: vsadmin  
password: <password>
```


NVMe/TCP-Beispiel

Sie benötigen eine SVM, die mit NVMe auf Ihrem ONTAP Backend konfiguriert ist. Dies ist eine grundlegende Backend-Konfiguration für NVMe/TCP.

```
---  
version: 1  
backendName: NVMeBackend  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_nvme  
username: vsadmin  
password: password  
sanType: nvme  
useREST: true
```

SCSI über FC (FCP) Beispiel

Sie benötigen eine SVM, die mit FC auf Ihrem ONTAP Backend konfiguriert ist. Dies ist eine grundlegende Backend-Konfiguration für FC.

```
---  
version: 1  
backendName: fcp-backend  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_fc  
username: vsadmin  
password: password  
sanType: fcp  
useREST: true
```

Backend-Konfigurationsbeispiel mit nameTemplate

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
labels:
  cluster: ClusterA
PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Beispiel für formatOptions für den ontap-san-economy-Treiber

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

Beispiele für Backends mit virtuellen Pools

In diesen Beispiel-Backend-Definitionsdateien sind spezifische Standardwerte für alle Speicherpools festgelegt, wie zum Beispiel `spaceReserve` bei keiner, `spaceAllocation` bei falsch und `encryption` bei falsch. Die virtuellen Pools werden im Speicherbereich definiert.

Trident legt Bereitstellungsbezeichnungen im Feld „Kommentare“ fest. Kommentare werden auf dem FlexVol volume festgelegt. Trident kopiert bei der Bereitstellung alle auf einem virtuellen Pool vorhandenen Labels auf das Speichervolume. Zur Vereinfachung können Speicheradministratoren Bezeichnungen pro virtuellem Pool definieren und Volumes nach Bezeichnung gruppieren.

In diesen Beispielen legen einige der Speicherpools ihre eigenen Einstellungen fest. `spaceReserve` , `spaceAllocation` , Und `encryption` Werte, und einige Pools überschreiben die Standardwerte.



```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "40000"
      zone: us_east_1a
      defaults:
        spaceAllocation: "true"
        encryption: "true"
        adaptiveQosPolicy: adaptive-extreme
  - labels:
      protection: silver
      creditpoints: "20000"
      zone: us_east_1b
      defaults:
        spaceAllocation: "false"
        encryption: "true"
        qosPolicy: premium
  - labels:
      protection: bronze
      creditpoints: "5000"
      zone: us_east_1c
      defaults:
        spaceAllocation: "true"
        encryption: "false"

```

```

---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
region: us_east_1
storage:
  - labels:
      app: oracledb
      cost: "30"
      zone: us_east_1a
      defaults:
        spaceAllocation: "true"
        encryption: "true"
  - labels:
      app: postgresdb
      cost: "20"
      zone: us_east_1b
      defaults:
        spaceAllocation: "false"
        encryption: "true"
  - labels:
      app: mysqldb
      cost: "10"
      zone: us_east_1c
      defaults:
        spaceAllocation: "true"
        encryption: "false"
  - labels:
      department: legal
      creditpoints: "5000"
      zone: us_east_1c

```

```
defaults:
  spaceAllocation: "true"
  encryption: "false"
```

NVMe/TCP-Beispiel

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: "false"
  encryption: "true"
storage:
  - labels:
      app: testApp
      cost: "20"
      defaults:
        spaceAllocation: "false"
        encryption: "false"
```

Backends StorageClasses zuordnen

Die folgenden StorageClass-Definitionen beziehen sich auf die [Beispiele für Backends mit virtuellen Pools](#). Verwenden des `parameters.selector` Im Feld „StorageClass“ wird für jede StorageClass angegeben, welche virtuellen Pools zum Hosten eines Volumes verwendet werden können. Das Volumen wird die im gewählten virtuellen Pool definierten Aspekte aufweisen.

- Der `protection-gold` Die StorageClass wird dem ersten virtuellen Pool im `ontap-san` Backend. Dies ist der einzige Pool, der Schutz auf Goldniveau bietet.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"

```

- Der protection-not-gold Die StorageClass wird dem zweiten und dritten virtuellen Pool zugeordnet. ontap-san Backend. Dies sind die einzigen Pools, die ein anderes Schutzniveau als Gold bieten.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"

```

- Der app-mysqldb Die StorageClass wird dem dritten virtuellen Pool zugeordnet. ontap-san-economy Backend. Dies ist der einzige Pool, der eine Speicherpoolkonfiguration für Anwendungen vom Typ mysqldb bietet.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"

```

- Der protection-silver-creditpoints-20k Die StorageClass wird dem zweiten virtuellen Pool zugeordnet. ontap-san Backend. Dies ist der einzige Pool, der Schutz auf Silber-Niveau und 20000 Kreditpunkte bietet.


```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"

```

- Der creditpoints-5k Die StorageClass wird dem dritten virtuellen Pool zugeordnet. ontap-san Backend und der vierte virtuelle Pool im ontap-san-economy Backend. Dies sind die einzigen Poolangebote mit 5000 Kreditpunkten.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

- Der my-test-app-sc Die StorageClass wird der folgenden zugeordnet: testAPP virtueller Pool im ontap-san Fahrer mit sanType: nvme . Dies ist das einzige Poolangebot testApp .

```

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"

```

Trident entscheidet, welcher virtuelle Pool ausgewählt wird und stellt sicher, dass die Speicheranforderungen erfüllt werden.

ONTAP NAS-Treiber

ONTAP NAS-Treiberübersicht

Erfahren Sie mehr über die Konfiguration eines ONTAP Backends mit ONTAP und Cloud Volumes ONTAP NAS-Treibern.

ONTAP NAS-Treiberdetails

Trident stellt die folgenden NAS-Speichertreiber zur Verfügung, um mit dem ONTAP Cluster zu kommunizieren. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	Lautstärke modus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
ontap-nas	NFS SMB	Dateisystem	RWO, ROX, RWX, RWOP	"", nfs , smb
ontap-nas-economy	NFS SMB	Dateisystem	RWO, ROX, RWX, RWOP	"", nfs , smb
ontap-nas-flexgroup	NFS SMB	Dateisystem	RWO, ROX, RWX, RWOP	"", nfs , smb



- Verwenden `ontap-san-economy` nur wenn die Anzahl der dauerhaften Speichernutzungen voraussichtlich höher sein wird als "[Unterstützte ONTAP Lautstärkebegrenzungen](#)".
- Verwenden `ontap-nas-economy` nur wenn die Anzahl der dauerhaften Speichernutzungen voraussichtlich höher sein wird als "[Unterstützte ONTAP Lautstärkebegrenzungen](#)" und die `ontap-san-economy` Der Treiber kann nicht verwendet werden.
- Nicht verwenden `ontap-nas-economy` wenn Sie mit einem Bedarf an Datenschutz, Notfallwiederherstellung oder Mobilität rechnen.
- NetApp empfiehlt die Verwendung von Flexvol Autogrow nicht in allen ONTAP -Treibern, außer `ontap-san`. Als Ausweichlösung unterstützt Trident die Verwendung von Snapshot-Reserven und skaliert Flexvol-Volumes entsprechend.

Benutzerberechtigungen

Trident wird voraussichtlich entweder als ONTAP oder SVM-Administrator ausgeführt, typischerweise unter Verwendung von `admin` Clusterbenutzer oder ein `vsadmin` SVM-Benutzer oder ein Benutzer mit einem anderen Namen, der die gleiche Rolle hat.

Bei Amazon FSx for NetApp ONTAP Bereitstellungen erwartet Trident , dass es entweder als ONTAP oder SVM-Administrator ausgeführt wird und den Cluster nutzt. `fsxadmin` Benutzer oder ein `vsadmin` SVM-Benutzer oder ein Benutzer mit einem anderen Namen, der die gleiche Rolle hat. Der `fsxadmin` Der Benutzer ist ein eingeschränkter Ersatz für den Cluster-Administratorbenutzer.



Wenn Sie die `limitAggregateUsage` Für diesen Parameter sind Cluster-Administratorrechte erforderlich. Bei der Verwendung von Amazon FSx for NetApp ONTAP mit Trident `limitAggregateUsage` Der Parameter funktioniert nicht mit dem `vsadmin` Und `fsxadmin` Benutzerkonten. Die Konfiguration schlägt fehl, wenn Sie diesen Parameter angeben.

Es ist zwar möglich, innerhalb von ONTAP eine restriktivere Rolle zu erstellen, die ein Trident -Treiber verwenden kann, wir empfehlen dies jedoch nicht. Die meisten neuen Versionen von Trident werden zusätzliche APIs aufrufen, die berücksichtigt werden müssen, was Aktualisierungen schwierig und fehleranfällig macht.

Bereiten Sie die Konfiguration eines Backends mit ONTAP NAS-Treibern vor.

Machen Sie sich mit den Anforderungen, Authentifizierungsoptionen und Exportrichtlinien für die Konfiguration eines ONTAP Backends mit ONTAP -NAS-Treibern vertraut.

Anforderungen

- Für alle ONTAP Backends verlangt Trident , dass mindestens ein Aggregat dem SVM zugewiesen wird.
- Sie können mehrere Treiber gleichzeitig ausführen und Speicherklassen erstellen, die auf den einen oder anderen Treiber verweisen. Beispielsweise könnten Sie eine Gold-Klasse konfigurieren, die Folgendes verwendet: `ontap-nas` Fahrer und eine Bronze-Klasse, die den `ontap-nas-economy` eins.
- Auf allen Ihren Kubernetes-Worker-Knoten müssen die entsprechenden NFS-Tools installiert sein. Siehe ["hier,"](#) für weitere Details.
- Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten laufen. Siehe [Bereiten Sie die Bereitstellung von SMB-Volumes vor](#) für Details.

Authentifizieren Sie das ONTAP Backend

Trident bietet zwei Modi zur Authentifizierung eines ONTAP Backends.

- Anmeldeinformationsbasiert: Dieser Modus erfordert ausreichende Berechtigungen für das ONTAP Backend. Es wird empfohlen, ein Konto zu verwenden, das einer vordefinierten Sicherheitsanmelderolle zugeordnet ist, wie zum Beispiel `admin` oder `vsadmin` um maximale Kompatibilität mit ONTAP Versionen zu gewährleisten.
- Zertifikatsbasiert: In diesem Modus ist ein auf dem Backend installiertes Zertifikat erforderlich, damit Trident mit einem ONTAP Cluster kommunizieren kann. Hierbei müssen in der Backend-Definition Base64-kodierte Werte des Client-Zertifikats, des Schlüssels und des vertrauenswürdigen CA-Zertifikats (falls verwendet, empfohlen) enthalten sein.

Sie können bestehende Backends aktualisieren, um zwischen anmeldeinformationsbasierten und zertifikatsbasierten Methoden zu wechseln. Es wird jedoch jeweils nur eine Authentifizierungsmethode unterstützt. Um zu einer anderen Authentifizierungsmethode zu wechseln, müssen Sie die bestehende Methode aus der Backend-Konfiguration entfernen.



Wenn Sie versuchen, **sowohl Anmeldeinformationen als auch Zertifikate** anzugeben, schlägt die Backend-Erstellung mit der Fehlermeldung fehl, dass in der Konfigurationsdatei mehr als eine Authentifizierungsmethode angegeben wurde.

Aktivieren Sie die anmeldeinformationsbasierte Authentifizierung

Trident benötigt die Anmeldeinformationen eines SVM-/Cluster-Administrators, um mit dem ONTAP Backend zu kommunizieren. Es wird empfohlen, standardisierte, vordefinierte Rollen zu verwenden, wie zum Beispiel `admin` oder `vsadmin` . Dies gewährleistet die Vorwärtskompatibilität mit zukünftigen ONTAP Versionen, die möglicherweise Feature-APIs zur Verwendung durch zukünftige Trident Versionen bereitstellen. Eine benutzerdefinierte Sicherheitsanmelderolle kann erstellt und mit Trident verwendet werden, dies wird jedoch nicht empfohlen.

Eine beispielhafte Backend-Definition sieht folgendermaßen aus:

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
credentials:
  name: secret-backend-creds
```

JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "credentials": {
    "name": "secret-backend-creds"
  }
}
```

Beachten Sie, dass die Backend-Definition der einzige Ort ist, an dem die Anmeldeinformationen im Klartext gespeichert werden. Nach der Erstellung des Backends werden Benutzernamen und Passwörter mit Base64 kodiert und als Kubernetes-Secrets gespeichert. Die Erstellung/Aktualisierung eines Backends ist der einzige Schritt, der Kenntnisse der Zugangsdaten erfordert. Daher handelt es sich um eine ausschließlich für Administratoren zulässige Operation, die vom Kubernetes-/Speicheradministrator durchgeführt werden muss.

Zertifikatsbasierte Authentifizierung aktivieren

Neue und bestehende Backends können ein Zertifikat verwenden und mit dem ONTAP Backend kommunizieren. Für die Backend-Definition werden drei Parameter benötigt.

- `clientCertificate`: Base64-kodierter Wert des Clientzertifikats.
- `clientPrivateKey`: Base64-kodierter Wert des zugehörigen privaten Schlüssels.
- `trustedCACertificate`: Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Bei Verwendung einer vertrauenswürdigen Zertifizierungsstelle muss dieser Parameter angegeben werden. Dies kann ignoriert werden, wenn keine vertrauenswürdige Zertifizierungsstelle verwendet wird.

Ein typischer Arbeitsablauf umfasst die folgenden Schritte.

Schritte

1. Generieren Sie ein Clientzertifikat und einen Schlüssel. Beim Generieren muss der allgemeine Name (CN) auf den ONTAP Benutzer gesetzt werden, der sich authentifizieren soll.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=vsadmin"
```

2. Fügen Sie dem ONTAP -Cluster ein vertrauenswürdiges CA-Zertifikat hinzu. Dies könnte bereits vom Speicheradministrator erledigt werden. Ignorieren, falls keine vertrauenswürdige Zertifizierungsstelle verwendet wird.

```
security certificate install -type server -cert-name <trusted-ca-cert-  
name> -vserver <vserver-name>  
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled  
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca  
<cert-authority>
```

3. Installieren Sie das Clientzertifikat und den Schlüssel (aus Schritt 1) auf dem ONTAP Cluster.

```
security certificate install -type client-ca -cert-name <certificate-  
name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. Bestätigen Sie, dass die ONTAP Sicherheitsanmeldungsrolle die folgenden Funktionen unterstützt: cert Authentifizierungsmethode.

```
security login create -user-or-group-name vsadmin -application ontapi  
-authentication-method cert -vserver <vserver-name>  
security login create -user-or-group-name vsadmin -application http  
-authentication-method cert -vserver <vserver-name>
```

5. Testen Sie die Authentifizierung mit dem generierten Zertifikat. Ersetzen Sie < ONTAP Management LIF> und <vserver name> durch die Management LIF IP-Adresse und den SVM-Namen. Sie müssen sicherstellen, dass die Servicerichtlinie des LIF auf Folgendes eingestellt ist: default-data-management .

```
curl -X POST -Lk https://<ONTAP-Management-  
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key  
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp  
xmlns="http://www.netapp.com/filer/admin" version="1.21"  
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Zertifikat, Schlüssel und vertrauenswürdiges CA-Zertifikat mit Base64 kodieren.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Erstellen Sie ein Backend unter Verwendung der im vorherigen Schritt erhaltenen Werte.

```
cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
```

NAME	STORAGE DRIVER	UUID
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214

Aktualisieren Sie die Authentifizierungsmethoden oder ändern Sie die Anmeldeinformationen.

Sie können ein bestehendes Backend aktualisieren, um eine andere Authentifizierungsmethode zu verwenden oder um die Anmeldeinformationen zu ändern. Dies funktioniert in beide Richtungen: Backends, die Benutzernamen/Passwörter verwenden, können auf die Verwendung von Zertifikaten umgestellt werden; Backends, die Zertifikate verwenden, können auf Benutzernamen/Passwort-basiert umgestellt werden. Dazu müssen Sie die bestehende Authentifizierungsmethode entfernen und die neue Authentifizierungsmethode hinzufügen. Verwenden Sie anschließend die aktualisierte Datei `backend.json`, die die erforderlichen Parameter enthält, um die Ausführung durchzuführen. `tridentctl update backend`.

```
cat cert-backend-updated.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "password",
  "storagePrefix": "myPrefix_"
}
```

```
#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
```

NAME	STORAGE DRIVER	UUID
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214
online	9	



Beim Ändern von Passwörtern muss der Speicheradministrator zuerst das Passwort für den Benutzer auf ONTAP aktualisieren. Anschließend erfolgt ein Backend-Update. Bei der Zertifikatsrotation können dem Benutzer mehrere Zertifikate hinzugefügt werden. Anschließend wird das Backend aktualisiert, um das neue Zertifikat zu verwenden. Danach kann das alte Zertifikat aus dem ONTAP Cluster gelöscht werden.

Durch die Aktualisierung des Backends wird der Zugriff auf bereits erstellte Volumes nicht beeinträchtigt, und auch später hergestellte Volume-Verbindungen werden nicht beeinträchtigt. Ein erfolgreiches Backend-Update zeigt an, dass Trident mit dem ONTAP -Backend kommunizieren und zukünftige Volumenoperationen bewältigen kann.

Erstellen einer benutzerdefinierten ONTAP Rolle für Trident

Sie können eine ONTAP Clusterrolle mit minimalen Berechtigungen erstellen, sodass Sie für Operationen in Trident nicht die ONTAP Administratorrolle verwenden müssen. Wenn Sie den Benutzernamen in einer Trident Backend-Konfiguration angeben, verwendet Trident die von Ihnen erstellte ONTAP Clusterrolle, um die

Operationen durchzuführen.

Siehe ["Trident -Benutzerrollengenerator"](#) Weitere Informationen zum Erstellen benutzerdefinierter Trident -Rollen finden Sie hier.

Verwendung der ONTAP Befehlszeile

1. Erstellen Sie eine neue Rolle mit folgendem Befehl:

```
security login role create <role_name\> -cmddirname "command" -access all  
-vserver <svm_name\>
```

2. Erstellen Sie einen Benutzernamen für den Trident -Benutzer:

```
security login create -username <user_name\> -application ontapi  
-authmethod <password\> -role <name_of_role_in_step_1\> -vserver  
<svm_name\> -comment "user_description"
```

3. Ordnen Sie die Rolle dem Benutzer zu:

```
security login modify username <user_name\> -vserver <svm_name\> -role  
<role_name\> -application ontapi -application console -authmethod  
<password\>
```

Systemmanager verwenden

Führen Sie die folgenden Schritte im ONTAP System Manager aus:

1. **Erstellen Sie eine benutzerdefinierte Rolle:**

- a. Um eine benutzerdefinierte Rolle auf Clusterebene zu erstellen, wählen Sie **Cluster > Einstellungen**.

(Oder) Um eine benutzerdefinierte Rolle auf SVM-Ebene zu erstellen, wählen Sie **Speicher > Speicher-VMs > required svm > Einstellungen > Benutzer und Rollen**.

- b. Wählen Sie das Pfeilsymbol (→) neben **Benutzer und Rollen** aus.
- c. Wählen Sie unter **Rollen** die Option **+Hinzufügen**.
- d. Definieren Sie die Regeln für die Rolle und klicken Sie auf **Speichern**.

2. **Rolle dem Trident -Benutzer zuordnen:** + Führen Sie die folgenden Schritte auf der Seite **Benutzer und Rollen** aus:

- a. Wählen Sie unter **Benutzer** das Symbol **+** zum Hinzufügen aus.
- b. Wählen Sie den gewünschten Benutzernamen und anschließend eine Rolle im Dropdown-Menü für **Rolle** aus.
- c. Klicken Sie auf **Speichern**.

Weitere Informationen finden Sie auf den folgenden Seiten:

- ["Benutzerdefinierte Rollen für die Administration von ONTAP"](#) oder ["Benutzerdefinierte Rollen definieren"](#)
- ["Mit Rollen und Benutzern arbeiten"](#)

NFS-Exportrichtlinien verwalten

Trident verwendet NFS-Exportrichtlinien, um den Zugriff auf die von ihm bereitgestellten Volumes zu steuern.

Trident bietet zwei Optionen für die Arbeit mit Exportrichtlinien:

- Trident kann die Exportrichtlinie selbst dynamisch verwalten; in diesem Betriebsmodus gibt der Speicheradministrator eine Liste von CIDR-Blöcken an, die zulässige IP-Adressen darstellen. Trident fügt bei der Veröffentlichung automatisch die entsprechenden Knoten-IPs, die in diese Bereiche fallen, zur Exportrichtlinie hinzu. Alternativ werden, wenn keine CIDRs angegeben sind, alle globalen Unicast-IPs, die auf dem Knoten gefunden werden, auf dem das Volume veröffentlicht wird, der Exportrichtlinie hinzugefügt.
- Speicheradministratoren können eine Exportrichtlinie erstellen und Regeln manuell hinzufügen. Trident verwendet die Standardexportrichtlinie, es sei denn, in der Konfiguration ist ein anderer Exportrichtlinienname angegeben.

Exportrichtlinien dynamisch verwalten

Trident bietet die Möglichkeit, Exportrichtlinien für ONTAP Backends dynamisch zu verwalten. Dies gibt dem Speicheradministrator die Möglichkeit, einen zulässigen Adressraum für Worker-Knoten-IPs festzulegen, anstatt explizite Regeln manuell zu definieren. Es vereinfacht die Verwaltung der Exportrichtlinien erheblich; Änderungen an den Exportrichtlinien erfordern keinen manuellen Eingriff mehr in den Speichercluster. Darüber hinaus trägt dies dazu bei, den Zugriff auf den Speichercluster auf Worker-Knoten zu beschränken, die Volumes einbinden und über IPs im angegebenen Bereich verfügen, wodurch eine feingranulare und automatisierte Verwaltung unterstützt wird.



Verwenden Sie keine Netzwerkadressübersetzung (NAT), wenn Sie dynamische Exportrichtlinien verwenden. Bei Verwendung von NAT sieht der Speicherkontroller die Frontend-NAT-Adresse und nicht die tatsächliche IP-Hostadresse. Daher wird der Zugriff verweigert, wenn in den Exportregeln keine Übereinstimmung gefunden wird.

Beispiel

Es gibt zwei Konfigurationsoptionen, die verwendet werden müssen. Hier ist ein Beispiel für eine Backend-Definition:

```
---
version: 1
storageDriverName: ontap-nas-economy
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
  - 192.168.0.0/24
autoExportPolicy: true
```



Bei Verwendung dieser Funktion müssen Sie sicherstellen, dass für den Root-Junction in Ihrer SVM eine zuvor erstellte Exportrichtlinie mit einer Exportregel existiert, die den CIDR-Block des Knotens zulässt (z. B. die Standardexportrichtlinie). Befolgen Sie stets die von NetApp empfohlenen Best Practices, um eine SVM für Trident zu dedizieren.

Hier ist eine Erklärung, wie diese Funktion funktioniert, anhand des obigen Beispiels:

- `autoExportPolicy` ist eingestellt auf `true`. Dies deutet darauf hin, dass Trident für jedes mit diesem Backend bereitgestellte Volume eine Exportrichtlinie erstellt. `svm1` SVM und handhaben das Hinzufügen und Löschen von Regeln mithilfe von `autoexportCIDRs` Adressblöcke. Solange ein Volume nicht an einen Knoten angehängt ist, verwendet das Volume eine leere Exportrichtlinie ohne Regeln, um unerwünschten Zugriff auf dieses Volume zu verhindern. Wenn ein Volume auf einem Knoten veröffentlicht wird, erstellt Trident eine Exportrichtlinie mit demselben Namen wie der zugrunde liegende Qtree, der die Knoten-IP innerhalb des angegebenen CIDR-Blocks enthält. Diese IPs werden auch der Exportrichtlinie hinzugefügt, die vom übergeordneten FlexVol volume verwendet wird.
 - Beispiel:
 - Backend-UUID 403b5326-8482-40db-96d0-d83fb3f4daec
 - `autoExportPolicy` eingestellt auf `true`
 - Speicherpräfix `trident`
 - PVC UUID a79bcf5f-7b6d-4a40-9876-e2551f159c1c
 - Ein Qtree mit dem Namen `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` erstellt eine Exportrichtlinie für den FlexVol mit dem Namen `trident-403b5326-8482-40db96d0-d83fb3f4daec`, eine Exportrichtlinie für den Qtree namens `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` und eine leere Exportpolitik namens `trident_empty` auf der SVM. Die Regeln für die FlexVol -Exportrichtlinie bilden eine Obermenge aller in den Qtree-Exportrichtlinien enthaltenen Regeln. Die leere Exportrichtlinie wird von allen Volumes wiederverwendet, die nicht angehängt sind.
- `autoExportCIDRs` enthält eine Liste von Adressblöcken. Dieses Feld ist optional und hat standardmäßig den Wert `["0.0.0.0/0", ":::/0"]`. Falls nicht definiert, fügt Trident alle global gültigen Unicast-Adressen hinzu, die auf den Worker-Knoten mit Veröffentlichungen gefunden werden.

In diesem Beispiel, `192.168.0.0/24` Ein Adressraum wird bereitgestellt. Dies bedeutet, dass Kubernetes-Knoten-IPs, die in diesen Adressbereich fallen und Veröffentlichungen enthalten, der von Trident erstellten Exportrichtlinie hinzugefügt werden. Wenn Trident einen Knoten registriert, auf dem es ausgeführt wird, ruft es die IP-Adressen des Knotens ab und überprüft sie anhand der bereitgestellten Adressblöcke.

`autoExportCIDRs` Zum Zeitpunkt der Veröffentlichung erstellt Trident nach dem Filtern der IPs die Exportrichtlinienregeln für die Client-IPs des Knotens, auf dem die Veröffentlichung erfolgt.

Sie können aktualisieren `autoExportPolicy` Und `autoExportCIDRs` für Backends, nachdem Sie diese erstellt haben. Sie können neue CIDRs für ein automatisch verwaltetes Backend hinzufügen oder bestehende CIDRs löschen. Beim Löschen von CIDRs ist darauf zu achten, dass bestehende Verbindungen nicht unterbrochen werden. Sie können diese Option auch deaktivieren. `autoExportPolicy` für ein Backend und greifen Sie auf eine manuell erstellte Exportrichtlinie zurück. Dies erfordert die Einstellung der `exportPolicy` Parameter in Ihrer Backend-Konfiguration.

Nachdem Trident ein Backend erstellt oder aktualisiert hat, können Sie das Backend mit folgendem Befehl überprüfen: `tridentctl` oder dem entsprechenden `tridentbackend` CRD:

```
./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      fileType: ext4
```

Wenn ein Knoten entfernt wird, überprüft Trident alle Exportrichtlinien, um die dem Knoten entsprechenden Zugriffsregeln zu entfernen. Durch das Entfernen dieser Knoten-IP aus den Exportrichtlinien der verwalteten Backends verhindert Trident unerwünschte Mounts, es sei denn, diese IP wird von einem neuen Knoten im Cluster wiederverwendet.

Bei bereits bestehenden Backends wird das Backend aktualisiert mit `tridentctl update backend` stellt sicher, dass Trident die Exportrichtlinien automatisch verwaltet. Dadurch werden bei Bedarf zwei neue Exportrichtlinien erstellt, die nach der UUID und dem Qtree-Namen des Backends benannt sind. Auf dem Backend vorhandene Volumes verwenden nach dem Aushängen und erneuten Einhängen die neu erstellten Exportrichtlinien.



Das Löschen eines Backends mit automatisch verwalteten Exportrichtlinien löscht die dynamisch erstellte Exportrichtlinie. Wird das Backend neu erstellt, wird es als neues Backend behandelt und führt zur Erstellung einer neuen Exportrichtlinie.

Wenn die IP-Adresse eines aktiven Knotens aktualisiert wird, müssen Sie den Trident Pod auf dem Knoten neu starten. Trident wird anschließend die Exportrichtlinie für die von ihm verwalteten Backends aktualisieren, um diese IP-Änderung widerzuspiegeln.

Bereiten Sie die Bereitstellung von SMB-Volumes vor

Mit ein wenig zusätzlicher Vorbereitung können Sie SMB-Volumes bereitstellen mithilfe von `ontap-nas` Fahrer.



Sie müssen sowohl das NFS- als auch das SMB/CIFS-Protokoll auf der SVM konfigurieren, um eine `ontap-nas-economy` SMB-Volume für ONTAP On-Premises-Cluster. Wenn eines dieser Protokolle nicht konfiguriert wird, schlägt die Erstellung des SMB-Volumes fehl.



`autoExportPolicy` wird für SMB-Volumes nicht unterstützt.

Bevor Sie beginnen

Bevor Sie SMB-Volumes bereitstellen können, benötigen Sie Folgendes.

- Ein Kubernetes-Cluster mit einem Linux-Controller-Knoten und mindestens einem Windows-Worker-Knoten, auf dem Windows Server 2022 ausgeführt wird. Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten laufen.
- Mindestens ein Trident Geheimnis, das Ihre Active Directory-Anmeldeinformationen enthält. Um Geheimnisse zu generieren `smbcreds` :

```
kubectl create secret generic smbcreds --from-literal username=user  
--from-literal password='password'
```

- Ein als Windows-Dienst konfigurierter CSI-Proxy. Um einen zu konfigurieren `csi-proxy` , siehe "[GitHub: CSI-Proxy](#)" oder "[GitHub: CSI-Proxy für Windows](#)" für Kubernetes-Knoten, die unter Windows laufen.

Schritte

1. Bei On-Premises ONTAP können Sie optional eine SMB-Freigabe erstellen oder Trident kann eine für Sie erstellen.



Für Amazon FSx for ONTAP werden SMB-Freigaben benötigt.

Sie können die SMB-Administratorfreigaben auf zwei Arten erstellen, entweder mithilfe von "[Microsoft Management Console](#)" über das Snap-In „Freigegebene Ordner“ oder über die ONTAP-Befehlszeilenschnittstelle. So erstellen Sie die SMB-Freigaben mithilfe der ONTAP-Befehlszeilenschnittstelle:

- a. Erstellen Sie gegebenenfalls die Verzeichnispfadstruktur für die Freigabe.

Der `vserver cifs share create` Der Befehl überprüft den Pfad, der bei der Erstellung der Freigabe in der Option `-path` angegeben wurde. Wenn der angegebene Pfad nicht existiert, schlägt der Befehl fehl.

- b. Erstellen Sie eine SMB-Freigabe, die dem angegebenen SVM zugeordnet ist:

```
vserver cifs share create -vserver vserver_name -share-name  
share_name -path path [-share-properties share_properties,...]  
[other_attributes] [-comment text]
```

- c. Überprüfen Sie, ob die Freigabe erstellt wurde:

```
vserver cifs share show -share-name share_name
```



Siehe "[Erstellen einer SMB-Freigabe](#)" Für alle Details.

2. Bei der Erstellung des Backends müssen Sie Folgendes konfigurieren, um SMB-Volumes anzugeben. Alle Konfigurationsoptionen für das FSx for ONTAP Backend finden Sie unter ["FSx für ONTAP: Konfigurationsoptionen und Beispiele"](#) .

Parameter	Beschreibung	Beispiel
smbShare	Sie können eine der folgenden Optionen angeben: den Namen einer SMB-Freigabe, die mit der Microsoft Management Console oder der ONTAP CLI erstellt wurde; einen Namen, unter dem Trident die SMB-Freigabe erstellen kann; oder Sie können den Parameter leer lassen, um den Zugriff auf Volumes durch die gemeinsame Freigabe zu verhindern. Dieser Parameter ist für On-Premises ONTAP optional. Dieser Parameter ist für Amazon FSx for ONTAP -Backends erforderlich und darf nicht leer sein.	smb-share
nasType	Muss eingestellt werden auf smb . Wenn null, wird standardmäßig der Wert verwendet. <code>nfs</code> .	smb
securityStyle	Sicherheitsstil für neue Bänder. Muss eingestellt sein auf ntfs oder mixed für SMB-Volumes.	ntfs oder mixed für SMB-Volumes
unixPermissions	Modus für neue Volumes. Muss bei SMB-Volumes leer bleiben.	""

Sichere SMB-Verbindungen aktivieren

Ab Version 25.06 unterstützt NetApp Trident die sichere Bereitstellung von SMB-Volumes, die mit `ontap-nas` Und `ontap-nas-economy` Backends. Wenn Secure SMB aktiviert ist, können Sie Active Directory (AD)-Benutzern und Benutzergruppen mithilfe von Zugriffssteuerungslisten (ACLs) einen kontrollierten Zugriff auf die SMB-Freigaben gewähren.

Wichtige Punkte

- Importieren `ontap-nas-economy` Volumen werden nicht unterstützt.
- Es werden nur schreibgeschützte Klone unterstützt für `ontap-nas-economy` Bänder.
- Wenn Secure SMB aktiviert ist, ignoriert Trident die im Backend angegebene SMB-Freigabe.
- Das Aktualisieren der PVC-Annotation, der Speicherklassenannotation und des Backend-Felds aktualisiert nicht die SMB-Freigabe-ACL.
- Die in der Annotation des Klon-PVC angegebene SMB-Freigabe-ACL hat Vorrang vor denjenigen im Quell-PVC.
- Stellen Sie sicher, dass Sie gültige AD-Benutzer angeben, während Sie Secure SMB aktivieren. Ungültige Benutzer werden nicht zur Zugriffskontrollliste (ACL) hinzugefügt.
- Wenn Sie dem gleichen AD-Benutzer im Backend, in der Speicherklasse und im PVC unterschiedliche Berechtigungen zuweisen, ergibt sich folgende Berechtigungspriorität: PVC, Speicherklasse und dann Backend.
- Secure SMB wird unterstützt für `ontap-nas` Gilt für verwaltete Volume-Importe und nicht für nicht verwaltete Volume-Importe.

Schritte

1. Geben Sie `adAdminUser` in `TridentBackendConfig` wie im folgenden Beispiel gezeigt an:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.193.176.x
  svm: svm0
  useREST: true
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

2. Fügen Sie die Annotation in der Speicherklasse hinzu.

Füge die `trident.netapp.io/smbShareAdUser` Annotation der Speicherklasse, um sicheres SMB ohne Ausfall zu ermöglichen. Der für die Annotation angegebene Benutzerwert `trident.netapp.io/smbShareAdUser` sollte mit dem im `smbcreds` Geheimnis. Sie können eine der folgenden Optionen auswählen: `smbShareAdUserPermission: full_control`, `change`, oder `read`. Die Standardberechtigung ist `full_control`.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

1. Erstellen Sie ein PVC.

Das folgende Beispiel erzeugt eine PVC:

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/snapshotDirectory: "true"
    trident.netapp.io/smbShareAccessControl: |
      read:
        - tridentADtest
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc

```

ONTAP NAS-Konfigurationsoptionen und Beispiele



Lernen Sie, wie Sie ONTAP NAS-Treiber mit Ihrer Trident -Installation erstellen und verwenden. Dieser Abschnitt enthält Beispiele für die Backend-Konfiguration und Details zur Zuordnung von Backends zu StorageClasses.


Backend-Konfigurationsoptionen

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
version		Immer 1
storageDriverName	Name des Speichertreibers	ontap-nas, ontap-nas-economy , oder ontap-nas-flexgroup
backendName	Benutzerdefinierter Name oder das Speicher-Backend	Fahrername + "_" + dataLIF
managementLIF	IP-Adresse eines Clusters oder SVM-Management-LIF. Es kann ein vollqualifizierter Domänenname (FQDN) angegeben werden. Kann so eingestellt werden, dass IPv6-Adressen verwendet werden, wenn Trident mit dem IPv6-Flag installiert wurde. IPv6-Adressen müssen in eckigen Klammern definiert werden, z. B. [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555] . Für einen nahtlosen MetroCluster Wechsel siehe MetroCluster Beispiel .	„10.0.0.1“, „[2001:1234:abcd::fefe]“

Parameter	Beschreibung	Standard
dataLIF	IP-Adresse des Protokolls LIF. NetApp empfiehlt die Angabe <code>dataLIF</code> . Falls keine Daten angegeben werden, ruft Trident die dataLIFs vom SVM ab. Sie können einen vollqualifizierten Domännennamen (FQDN) angeben, der für die NFS-Mount-Operationen verwendet werden soll. Dadurch können Sie ein Round-Robin-DNS erstellen, um die Last auf mehrere DataLIFs zu verteilen. Kann nach der Ersteinrichtung geändert werden. Siehe . Kann so eingestellt werden, dass IPv6-Adressen verwendet werden, wenn Trident mit dem IPv6-Flag installiert wurde. IPv6-Adressen müssen in eckigen Klammern definiert werden, z. B. <code>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</code> . Für Metrocluster auslassen. Siehe die MetroCluster Beispiel .	Angegebene Adresse oder abgeleitet von SVM, falls nicht angegeben (nicht empfohlen).
svm	Zu verwendende virtuelle Speichermaschine Für Metrocluster auslassen. Siehe die MetroCluster Beispiel .	Abgeleitet, wenn eine SVM <code>managementLIF</code> wird angegeben
autoExportPolicy	Automatische Erstellung und Aktualisierung von Exportrichtlinien aktivieren [Boolesch]. Verwenden des <code>autoExportPolicy</code> Und <code>autoExportCIDRs</code> Optionen: Trident kann Exportrichtlinien automatisch verwalten.	FALSCH
autoExportCIDRs	Liste der CIDRs, anhand derer die Kubernetes-Knoten-IPs gefiltert werden sollen, wenn <code>autoExportPolicy</code> ist aktiviert. Verwenden des <code>autoExportPolicy</code> Und <code>autoExportCIDRs</code> Optionen: Trident kann Exportrichtlinien automatisch verwalten.	<code>["0.0.0.0/0", ":::0"]</code>
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Datenträger angewendet werden sollen	""
clientCertificate	Base64-kodierter Wert des Clientzertifikats. Wird für zertifikatsbasierte Authentifizierung verwendet	""
clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für zertifikatsbasierte Authentifizierung verwendet	""
trustedCACertificate	Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Optional. Wird für zertifikatsbasierte Authentifizierung verwendet	""
username	Benutzername für die Verbindung mit dem Cluster/SVM. Wird für die Authentifizierung auf Basis von Anmeldeinformationen verwendet. Informationen zur Active Directory-Authentifizierung finden Sie unter "Authentifizieren Sie Trident bei einem Backend-SVM mithilfe von Active Directory-Anmeldeinformationen" .	

Parameter	Beschreibung	Standard
password	Kennwort für die Verbindung mit dem Cluster/SVM. Wird für die Authentifizierung auf Basis von Anmeldeinformationen verwendet. Informationen zur Active Directory-Authentifizierung finden Sie unter "Authentifizieren Sie Trident bei einem Backend-SVM mithilfe von Active Directory-Anmeldeinformationen" .	
storagePrefix	<p>Präfix, das beim Bereitstellen neuer Volumes in der SVM verwendet wird. Kann nach der Konfiguration nicht mehr aktualisiert werden.</p> <div>  <p>Bei Verwendung von ontap-nas-economy und einem storagePrefix mit 24 oder mehr Zeichen wird das storagePrefix nicht in die Qtrees eingebettet, sondern nur im Volume-Namen.</p> </div>	"Dreizack"
aggregate	<p>Aggregat für die Bereitstellung (optional; falls festgelegt, muss es der SVM zugewiesen werden). Für die ontap-nas-flexgroup Treiber, diese Option wird ignoriert. Falls kein Aggregat zugewiesen ist, kann jedes der verfügbaren Aggregate zur Bereitstellung eines FlexGroup Volumes verwendet werden.</p> <div>  <p>Wenn das Aggregat in SVM aktualisiert wird, wird es in Trident automatisch durch Abfrage von SVM aktualisiert, ohne dass der Trident Controller neu gestartet werden muss. Wenn Sie in Trident ein bestimmtes Aggregat zur Bereitstellung von Volumes konfiguriert haben und dieses Aggregat umbenannt oder aus der SVM verschoben wird, wechselt das Backend in Trident in den Fehlerzustand, während es das SVM-Aggregat abfragt. Sie müssen entweder das Aggregat in ein auf der SVM vorhandenes ändern oder es vollständig entfernen, um das Backend wieder online zu bringen.</p> </div>	""
limitAggregateUsage	Die Bereitstellung schlägt fehl, wenn die Auslastung diesen Prozentsatz überschreitet. Gilt nicht für Amazon FSx für ONTAP.	"" (wird nicht standardmäßig erzwungen)

Parameter	Beschreibung	Standard
flexgroupAggregateList	<p>Liste der Aggregate für die Bereitstellung (optional; falls festgelegt, muss sie der SVM zugewiesen werden). Alle dem SVM zugewiesenen Aggregate werden zur Bereitstellung eines FlexGroup Volumes verwendet. Unterstützt für den Speichertreiber ontap-nas-flexgroup.</p> <div>  <p>Wenn die Aggregatliste in SVM aktualisiert wird, wird die Liste in Trident automatisch durch Abfrage von SVM aktualisiert, ohne dass der Trident Controller neu gestartet werden muss. Wenn Sie in Trident eine bestimmte Aggregatliste für die Bereitstellung von Volumes konfiguriert haben und diese Aggregatliste umbenannt oder aus SVM verschoben wird, wechselt das Backend in Trident beim Abfragen des SVM-Aggregats in den Fehlerzustand. Sie müssen entweder die Aggregatliste durch eine auf der SVM vorhandene Liste ersetzen oder sie vollständig entfernen, um das Backend wieder online zu bringen.</p> </div>	""
limitVolumeSize	<p>Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet. Beschränkt außerdem die maximale Größe der von ihm verwalteten Volumina für Qtrees, und die <code>qtreesPerFlexvol</code>. Diese Option ermöglicht die Anpassung der maximalen Anzahl von Qtrees pro FlexVol volume.</p>	"" (wird nicht standardmäßig erzwungen)
debugTraceFlags	<p>Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel: {"api":false, "method":true} Nicht verwenden <code>debugTraceFlags</code> Es sei denn, Sie befinden sich in der Fehlersuche und benötigen einen detaillierten Protokollauszug.</p>	null
nasType	<p>Konfiguration der Erstellung von NFS- oder SMB-Volumes. Optionen sind <code>nfs</code>, <code>smb</code> oder <code>null</code>. Bei der Einstellung „null“ werden standardmäßig NFS-Volumes verwendet.</p>	<code>nfs</code>

Parameter	Beschreibung	Standard
nfsMountOptions	Durch Kommas getrennte Liste der NFS-Mount-Optionen. Die Mount-Optionen für Kubernetes-persistente Volumes werden normalerweise in Speicherklassen angegeben. Wenn jedoch in einer Speicherklasse keine Mount-Optionen angegeben sind, greift Trident auf die in der Konfigurationsdatei des Speicher-Backends angegebenen Mount-Optionen zurück. Wenn in der Speicherklasse oder der Konfigurationsdatei keine Mount-Optionen angegeben sind, setzt Trident keine Mount-Optionen auf einem zugehörigen persistenten Volume.	""
qtreesPerFlexvol	Die maximale Anzahl an Qtrees pro FlexVol muss im Bereich [50, 300] liegen.	"200"
smbShare	Sie können eine der folgenden Optionen angeben: den Namen einer SMB-Freigabe, die mit der Microsoft Management Console oder der ONTAP CLI erstellt wurde; einen Namen, unter dem Trident die SMB-Freigabe erstellen kann; oder Sie können den Parameter leer lassen, um den Zugriff auf Volumes durch die gemeinsame Freigabe zu verhindern. Dieser Parameter ist für On-Premises ONTAP optional. Dieser Parameter ist für Amazon FSx for ONTAP -Backends erforderlich und darf nicht leer sein.	smb-share
useREST	Boolescher Parameter zur Verwendung von ONTAP REST-APIs. useREST Wenn eingestellt auf true Trident verwendet ONTAP REST-APIs zur Kommunikation mit dem Backend; wenn eingestellt auf false Trident verwendet ONTAPI (ZAPI)-Aufrufe zur Kommunikation mit dem Backend. Diese Funktion erfordert ONTAP 9.11.1 und höher. Darüber hinaus muss die verwendete ONTAP Anmelderolle Zugriff auf die ontapi Anwendung. Dies wird durch die vordefinierte Bedingung erfüllt. vsadmin Und cluster-admin Rollen. Ab der Trident Version 24.06 und ONTAP 9.15.1 oder höher, useREST ist eingestellt auf true Standardmäßig; ändern useREST Zu false ONTAPI (ZAPI)-Aufrufe verwenden.	true für ONTAP 9.15.1 oder höher, andernfalls false.
limitVolumePoolSize	Maximal anforderbare FlexVol Größe bei Verwendung von Qtrees im ontap-nas-economy-Backend.	"" (wird nicht standardmäßig erzwungen)
denyNewVolumePools	Beschränkt ontap-nas-economy Backends daran zu hindern, neue FlexVol -Volumes zu erstellen, die ihre Qtrees enthalten. Für die Bereitstellung neuer PVs werden ausschließlich bereits vorhandene Flexvols verwendet.	

Parameter	Beschreibung	Standard
adAdminUser	Active Directory-Administratorbenutzer oder Benutzergruppe mit vollem Zugriff auf SMB-Freigaben. Verwenden Sie diesen Parameter, um Administratorrechte für die SMB-Freigabe mit voller Kontrolle zu erteilen.	

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

Sie können die Standardbereitstellung mithilfe dieser Optionen steuern. `defaults` Abschnitt der Konfiguration. Ein Beispiel finden Sie in den folgenden Konfigurationsbeispielen.

Parameter	Beschreibung	Standard
spaceAllocation	Speicherplatzbelegung für Qtrees	"WAHR"
spaceReserve	Platzreservierungsmodus; "keine" (dünn) oder "Volumen" (dick)	"keiner"
snapshotPolicy	Zu verwendende Snapshot-Richtlinie	"keiner"
qosPolicy	Die QoS-Richtliniengruppe soll den erstellten Volumes zugewiesen werden. Wählen Sie pro Speicherpool/Backend entweder qosPolicy oder adaptiveQosPolicy aus.	""
adaptiveQosPolicy	Adaptive QoS-Richtliniengruppe, die den erstellten Volumes zugewiesen werden soll. Wählen Sie pro Speicherpool/Backend entweder qosPolicy oder adaptiveQosPolicy. Wird von ontap-nas-economy nicht unterstützt.	""
snapshotReserve	Prozentsatz des für Snapshots reservierten Speichervolumens	"0" wenn snapshotPolicy ist "keine", ansonsten ""
splitOnClone	Beim Erstellen eines Klons diesen von seinem Elternklon trennen	"FALSCH"
encryption	Aktivieren Sie die NetApp Volumeverschlüsselung (NVE) auf dem neuen Volume; Standardwert ist <code>false</code> . Um diese Option nutzen zu können, muss NVE auf dem Cluster lizenziert und aktiviert sein. Wenn NAE im Backend aktiviert ist, wird jedes in Trident bereitgestellte Volume NAE-fähig sein. Weitere Informationen finden Sie unter: "Wie Trident mit NVE und NAE zusammenarbeitet" .	"FALSCH"
tieringPolicy	Stufenrichtlinie: "keine" verwenden	
unixPermissions	Modus für neue Volumes	„777“ für NFS-Volumes; leer (nicht zutreffend) für SMB-Volumes
snapshotDir	Steuert den Zugriff auf die <code>.snapshot</code> Verzeichnis	"true" für NFSv4, "false" für NFSv3

Parameter	Beschreibung	Standard
exportPolicy	Exportrichtlinie zu verwenden	"Standard"
securityStyle	Sicherheitsstil für neue Bände. NFS unterstützt <code>mixed</code> Und <code>unix</code> Sicherheitsstile. SMB-Unterstützung <code>mixed</code> Und <code>ntfs</code> Sicherheitsstile.	NFS-Standard ist <code>unix</code> . SMB-Standard ist <code>ntfs</code> .
nameTemplate	Vorlage zum Erstellen benutzerdefinierter Datenträgernamen.	""



Die Verwendung von QoS-Richtliniengruppen mit Trident erfordert ONTAP 9.8 oder höher. Sie sollten eine nicht gemeinsam genutzte QoS-Richtliniengruppe verwenden und sicherstellen, dass die Richtliniengruppe auf jeden einzelnen Bestandteil angewendet wird. Eine gemeinsam genutzte QoS-Richtliniengruppe setzt die Obergrenze für den Gesamtdurchsatz aller Workloads durch.

Beispiele für die Volumenbereitstellung

Hier ist ein Beispiel mit vordefinierten Standardwerten:

```
---
version: 1
storageDriverName: ontap-nas
backendName: customBackendName
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
labels:
  k8scluster: dev1
  backend: dev1-nasbackend
svm: trident_svm
username: cluster-admin
password: <password>
limitAggregateUsage: 80%
limitVolumeSize: 50Gi
nfsMountOptions: nfsvers=4
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: premium
  exportPolicy: myk8scluster
  snapshotPolicy: default
  snapshotReserve: "10"
```

Für `ontap-nas` Und `ontap-nas-flexgroups` Trident verwendet nun eine neue Berechnung, um sicherzustellen, dass FlexVol mit dem SnapshotReserve-Prozentsatz und PVC korrekt dimensioniert wird.

Wenn der Benutzer ein PVC anfordert, erstellt Trident das ursprüngliche FlexVol mit mehr Speicherplatz mithilfe der neuen Berechnung. Diese Berechnung stellt sicher, dass der Benutzer den im PVC angeforderten beschreibbaren Speicherplatz erhält und nicht weniger. Vor Version 21.07 erhielt der Benutzer, wenn er ein PVC (z. B. 5 GiB) mit einem `SnapshotReserve` von 50 Prozent anforderte, nur 2,5 GiB beschreibbaren Speicherplatz. Dies liegt daran, dass der Benutzer das gesamte Volumen angefordert hat. `snapshotReserve` ist ein Prozentsatz davon. Mit Trident 21.07 fordert der Benutzer den beschreibbaren Speicherplatz an, und Trident definiert diesen. `snapshotReserve` Zahl als Prozentsatz des Gesamtvolumens. Dies gilt nicht für `ontap-nas-economy`. Wie das funktioniert, sehen Sie im folgenden Beispiel:

Die Berechnung erfolgt wie folgt:

```
Total volume size = (PVC requested size) / (1 - (snapshotReserve
percentage) / 100)
```

Bei `SnapshotReserve` = 50 % und PVC-Anforderung = 5 GiB beträgt die Gesamtgröße des Volumes $5/0.5 = 10$ GiB und die verfügbare Größe beträgt 5 GiB, was der vom Benutzer in der PVC-Anforderung angeforderten Größe entspricht. Der `volume show` Befehl sollte ähnliche Ergebnisse wie in diesem Beispiel liefern:

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
	_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4		online	RW	10GB	5.00GB	0%
	_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba		online	RW	1GB	511.8MB	0%

2 entries were displayed.

Vorhandene Backends aus früheren Installationen stellen beim Upgrade von Trident Volumes wie oben beschrieben bereit. Für Volumes, die Sie vor dem Upgrade erstellt haben, sollten Sie die Größe der Volumes anpassen, damit die Änderung berücksichtigt wird. Zum Beispiel ein 2 GiB PVC mit `snapshotReserve=50`. Das vorherige Ergebnis war ein Volumen mit 1 GiB beschreibbarem Speicherplatz. Wenn Sie die Größe des Volumes beispielsweise auf 3 GiB ändern, stehen der Anwendung 3 GiB beschreibbarer Speicherplatz auf einem 6-GiB-Volumen zur Verfügung.

Beispiele für minimale Konfigurationen

Die folgenden Beispiele zeigen Basiskonfigurationen, bei denen die meisten Parameter auf Standardwerte eingestellt bleiben. Dies ist die einfachste Möglichkeit, ein Backend zu definieren.



Wenn Sie Amazon FSx auf NetApp ONTAP mit Trident verwenden, wird empfohlen, DNS-Namen für LIFs anstelle von IP-Adressen anzugeben.

ONTAP NAS Wirtschaftsbeispiel

```
---  
version: 1  
storageDriverName: ontap-nas-economy  
managementLIF: 10.0.0.1  
dataLIF: 10.0.0.2  
svm: svm_nfs  
username: vsadmin  
password: password
```

ONTAP NAS Flexgroup-Beispiel

```
---  
version: 1  
storageDriverName: ontap-nas-flexgroup  
managementLIF: 10.0.0.1  
dataLIF: 10.0.0.2  
svm: svm_nfs  
username: vsadmin  
password: password
```

MetroCluster Beispiel

Sie können das Backend so konfigurieren, dass eine manuelle Aktualisierung der Backend-Definition nach einem Switchover und Switchback vermieden wird. ["SVM-Replikation und -Wiederherstellung"](#) .

Für einen nahtlosen Übergang und Rückwechsel geben Sie die SVM wie folgt an: `managementLIF` und lassen Sie die `dataLIF` Und `svm` Parameter. Beispiel:

```
---  
version: 1  
storageDriverName: ontap-nas  
managementLIF: 192.168.1.66  
username: vsadmin  
password: password
```

Beispiel für SMB-Volumes

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
nasType: smb
securityStyle: ntfs
unixPermissions: ""
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

Beispiel für zertifikatsbasierte Authentifizierung

Dies ist ein minimales Beispiel für eine Backend-Konfiguration. `clientCertificate`, `clientPrivateKey`, Und `trustedCACertificate` (optional, falls eine vertrauenswürdige Zertifizierungsstelle verwendet wird) werden in `backend.json` und nehmen Sie die Base64-kodierten Werte des Clientzertifikats, des privaten Schlüssels bzw. des vertrauenswürdigen CA-Zertifikats.

```
---
version: 1
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```


Beispiel für eine automatische Exportrichtlinie

Dieses Beispiel zeigt Ihnen, wie Sie Trident anweisen können, dynamische Exportrichtlinien zu verwenden, um die Exportrichtlinie automatisch zu erstellen und zu verwalten. Dies funktioniert genauso für die `ontap-nas-economy` Und `ontap-nas-flexgroup` Fahrer.

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-nasbackend
autoExportPolicy: true
autoExportCIDRs:
- 10.0.0.0/24
username: admin
password: password
nfsMountOptions: nfsvers=4
```

Beispiel für IPv6-Adressen

Dieses Beispiel zeigt `managementLIF` unter Verwendung einer IPv6-Adresse.

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas_ipv6_backend
managementLIF: "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]"
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-ontap-ipv6
svm: nas_ipv6_svm
username: vsadmin
password: password
```

Amazon FSx für ONTAP mit SMB-Volumes – Beispiel

Der smbShare Dieser Parameter ist für FSx for ONTAP mit SMB-Volumes erforderlich.

```
---
version: 1
backendName: SMBBackend
storageDriverName: ontap-nas
managementLIF: example.mgmt.fqdn.aws.com
nasType: smb
dataLIF: 10.0.0.15
svm: nfs_svm
smbShare: smb-share
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

Backend-Konfigurationsbeispiel mit nameTemplate

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap-nas-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
      lume.RequestName}}"
labels:
  cluster: ClusterA
PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Beispiele für Backends mit virtuellen Pools

In den unten gezeigten Beispiel-Backend-Definitionsdateien sind spezifische Standardwerte für alle Speicherpools festgelegt, wie zum Beispiel: `spaceReserve` bei keiner, `spaceAllocation` bei falsch und `encryption` bei falsch. Die virtuellen Pools werden im Speicherbereich definiert.

Trident legt Bereitstellungsbezeichnungen im Feld „Kommentare“ fest. Kommentare sind auf FlexVol für `ontap-nas` oder FlexGroup für `ontap-nas-flexgroup`. Trident kopiert bei der Bereitstellung alle im

virtuellen Pool vorhandenen Labels auf das Speichervolume. Zur Vereinfachung können Speicheradministratoren Bezeichnungen pro virtuellem Pool definieren und Volumes nach Bezeichnung gruppieren.

In diesen Beispielen legen einige der Speicherpools ihre eigenen Einstellungen fest. `spaceReserve` , `spaceAllocation` , Und `encryption` Werte, und einige Pools überschreiben die Standardwerte.

```

---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
svm: svm_nfs
username: admin
password: <password>
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: "false"
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      app: msoffice
      cost: "100"
      zone: us_east_1a
      defaults:
        spaceReserve: volume
        encryption: "true"
        unixPermissions: "0755"
        adaptiveQosPolicy: adaptive-premium
  - labels:
      app: slack
      cost: "75"
      zone: us_east_1b
      defaults:
        spaceReserve: none
        encryption: "true"
        unixPermissions: "0755"
  - labels:
      department: legal
      creditpoints: "5000"
      zone: us_east_1b
      defaults:
        spaceReserve: none
        encryption: "true"
        unixPermissions: "0755"
  - labels:
      app: wordpress

```

```
    cost: "50"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
- labels:
    app: mysqlldb
    cost: "25"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

```

---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "50000"
      zone: us_east_1a
      defaults:
        spaceReserve: volume
        encryption: "true"
        unixPermissions: "0755"
  - labels:
      protection: gold
      creditpoints: "30000"
      zone: us_east_1b
      defaults:
        spaceReserve: none
        encryption: "true"
        unixPermissions: "0755"
  - labels:
      protection: silver
      creditpoints: "20000"
      zone: us_east_1c
      defaults:
        spaceReserve: none
        encryption: "true"
        unixPermissions: "0775"
  - labels:
      protection: bronze
      creditpoints: "10000"
      zone: us_east_1d
      defaults:

```

```
spaceReserve: volume  
encryption: "false"  
unixPermissions: "0775"
```

```

---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: nas_economy_store
region: us_east_1
storage:
  - labels:
      department: finance
      creditpoints: "6000"
      zone: us_east_1a
      defaults:
        spaceReserve: volume
        encryption: "true"
        unixPermissions: "0755"
  - labels:
      protection: bronze
      creditpoints: "5000"
      zone: us_east_1b
      defaults:
        spaceReserve: none
        encryption: "true"
        unixPermissions: "0755"
  - labels:
      department: engineering
      creditpoints: "3000"
      zone: us_east_1c
      defaults:
        spaceReserve: none
        encryption: "true"
        unixPermissions: "0775"
  - labels:
      department: humanresource
      creditpoints: "2000"
      zone: us_east_1d
      defaults:
        spaceReserve: volume

```



```
encryption: "false"
unixPermissions: "0775"
```

Backends StorageClasses zuordnen

Die folgenden StorageClass-Definitionen beziehen sich auf [Beispiele für Backends mit virtuellen Pools](#) . Verwenden des `parameters.selector` Im Feld „StorageClass“ wird für jede StorageClass angegeben, welche virtuellen Pools zum Hosten eines Volumes verwendet werden können. Das Volumen wird die im gewählten virtuellen Pool definierten Aspekte aufweisen.

- Der `protection-gold` Die StorageClass wird dem ersten und zweiten virtuellen Pool zugeordnet. `ontap-nas-flexgroup` Backend. Dies sind die einzigen Pools, die einen Schutz auf Goldniveau bieten.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- Der `protection-not-gold` Die StorageClass wird dem dritten und vierten virtuellen Pool zugeordnet. `ontap-nas-flexgroup` Backend. Dies sind die einzigen Pools, die ein anderes Schutzniveau als Gold bieten.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- Der `app-mysqldb` Die StorageClass wird dem vierten virtuellen Pool zugeordnet. `ontap-nas` Backend. Dies ist der einzige Pool, der eine Speicherpoolkonfiguration für Anwendungen vom Typ `mysqldb` bietet.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"

```

- Die protection-silver-creditpoints-20k Die StorageClass wird dem dritten virtuellen Pool zugeordnet. ontap-nas-flexgroup Backend. Dies ist der einzige Pool, der Schutz auf Silber-Niveau und 20000 Kreditpunkte bietet.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"

```

- Der creditpoints-5k Die StorageClass wird dem dritten virtuellen Pool zugeordnet. ontap-nas Backend und der zweite virtuelle Pool im ontap-nas-economy Backend. Dies sind die einzigen Poolangebote mit 5000 Kreditpunkten.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

Trident entscheidet, welcher virtuelle Pool ausgewählt wird und stellt sicher, dass die Speicheranforderungen erfüllt werden.

Aktualisieren dataLIF nach der ersten Konfiguration

Sie können die dataLIF nach der Erstkonfiguration ändern, indem Sie den folgenden Befehl ausführen, um die neue Backend-JSON-Datei mit der aktualisierten dataLIF bereitzustellen.

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-with-updated-dataLIF>
```



Wenn PVCs an einem oder mehreren Pods angeschlossen sind, müssen Sie alle entsprechenden Pods herunterfahren und anschließend wieder hochfahren, damit die neue dataLIF-Regelung wirksam wird.

Beispiele für sichere KMU

Backend-Konfiguration mit dem ONTAP-NAS-Treiber

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Backend-Konfiguration mit dem ontap-nas-economy-Treiber

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret

```

Backend-Konfiguration mit Speicherpool

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm0
  useREST: false
  storage:
    - labels:
        app: msoffice
      defaults:
        adAdminUser: tridentADuser
  nasType: smb
  credentials:
    name: backend-tbc-ontap-invest-secret

```

Speicherklassenbeispiel mit dem ONTAP-NAS-Treiber

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADtest
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate

```



Stellen Sie sicher, dass Sie hinzufügen annotations um sichere KMU zu ermöglichen. Secure SMB funktioniert nicht ohne die Annotationen, unabhängig von den im Backend oder PVC festgelegten Konfigurationen.

Speicherklassenbeispiel mit dem Treiber ontap-nas-economy

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser3
parameters:
  backendType: ontap-nas-economy
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate

```

PVC-Beispiel mit einem einzelnen AD-Benutzer

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      change:
        - tridentADtest
      read:
        - tridentADuser
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

PVC-Beispiel mit mehreren AD-Benutzern

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

```

Amazon FSx for NetApp ONTAP

Trident mit Amazon FSx for NetApp ONTAP verwenden

"Amazon FSx for NetApp ONTAP" ist ein vollständig verwalteter AWS-Service, der es Kunden ermöglicht, Dateisysteme zu starten und auszuführen, die auf dem Speicherbetriebssystem NetApp ONTAP basieren. FSx for ONTAP ermöglicht es Ihnen, die Ihnen vertrauten Funktionen, die Leistung und die administrativen Möglichkeiten von NetApp zu nutzen und gleichzeitig die Einfachheit, Agilität, Sicherheit und Skalierbarkeit der Datenspeicherung auf AWS in Anspruch zu nehmen. FSx für ONTAP unterstützt die Funktionen des ONTAP -Dateisystems und die Administrations-APIs.

Sie können Ihr Amazon FSx for NetApp ONTAP Dateisystem mit Trident integrieren, um sicherzustellen, dass Kubernetes-Cluster, die im Amazon Elastic Kubernetes Service (EKS) ausgeführt werden, persistente Block- und Dateivolumes bereitstellen können, die von ONTAP unterstützt werden.

Das Dateisystem ist die primäre Ressource in Amazon FSx, analog zu einem ONTAP Cluster vor Ort. Innerhalb jeder SVM können Sie ein oder mehrere Volumes erstellen. Dabei handelt es sich um

Datencontainer, in denen die Dateien und Ordner Ihres Dateisystems gespeichert werden. Mit Amazon FSx for NetApp ONTAP wird ein verwaltetes Dateisystem in der Cloud bereitgestellt. Der neue Dateisystemtyp heißt * NetApp ONTAP*.

Durch die Verwendung von Trident mit Amazon FSx for NetApp ONTAP können Sie sicherstellen, dass Kubernetes-Cluster, die im Amazon Elastic Kubernetes Service (EKS) ausgeführt werden, persistente Block- und Dateivolumes bereitstellen können, die von ONTAP unterstützt werden.

Anforderungen

Zusätzlich zu ["Trident Anforderungen"](#) Um FSx für ONTAP mit Trident zu integrieren, benötigen Sie:

- Ein bestehender Amazon EKS-Cluster oder ein selbstverwalteter Kubernetes-Cluster mit `kubectl` installiert.
- Ein vorhandenes Amazon FSx for NetApp ONTAP Dateisystem und eine Storage Virtual Machine (SVM), die von den Worker-Knoten Ihres Clusters aus erreichbar ist.
- Worker-Knoten, die vorbereitet sind für ["NFS oder iSCSI"](#) .



Stellen Sie sicher, dass Sie die für Amazon Linux und Ubuntu erforderlichen Schritte zur Knotenvorbereitung befolgen. ["Amazon Machine Images"](#) (AMIs) abhängig von Ihrem EKS-AMI-Typ.

Überlegungen

- SMB-Volumes:
 - SMB-Volumes werden mithilfe von `ontap-nas` Nur für den Fahrer.
 - SMB-Volumes werden vom Trident EKS-Add-on nicht unterstützt.
 - Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten laufen. Siehe ["Bereiten Sie die Bereitstellung von SMB-Volumes vor"](#) für Details.
- Vor Trident 24.02 konnten Volumes, die auf Amazon FSx Dateisystemen mit aktivierter automatischer Datensicherung erstellt wurden, von Trident nicht gelöscht werden. Um dieses Problem in Trident 24.02 oder höher zu vermeiden, geben Sie Folgendes an: `fsxFileSystemID` `AWS apiRegion` `AWS apikey` und `AWS secretKey` in der Backend-Konfigurationsdatei für AWS FSx für ONTAP.



Wenn Sie Trident eine IAM-Rolle zuweisen, können Sie die Angabe der `apiRegion` , `apiKey` , Und `secretKey` Felder explizit an Trident übergeben. Weitere Informationen finden Sie unter ["FSx für ONTAP: Konfigurationsoptionen und Beispiele"](#) .

Gleichzeitige Nutzung von Trident SAN/iSCSI und EBS-CSI-Treiber

Wenn Sie `ontap-san`-Treiber (z. B. iSCSI) mit AWS (EKS, ROSA, EC2 oder einer anderen Instanz) verwenden möchten, kann es bei der auf den Knoten erforderlichen Multipath-Konfiguration zu Konflikten mit dem CSI-Treiber von Amazon Elastic Block Store (EBS) kommen. Um sicherzustellen, dass Multipathing funktioniert, ohne EBS-Festplatten auf demselben Knoten zu beeinträchtigen, müssen Sie EBS aus Ihrem Multipathing-Setup ausschließen. Dieses Beispiel zeigt ein `multipath.conf` Datei, die die erforderlichen Trident Einstellungen enthält und gleichzeitig EBS-Festplatten vom Multipathing ausschließt:


```
defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

Authentifizierung

Trident bietet zwei Authentifizierungsmodi an.

- Anmeldeinformationsbasiert (Empfohlen): Speichert Anmeldeinformationen sicher im AWS Secrets Manager. Sie können die `fsxadmin` Benutzer für Ihr Dateisystem oder die `vsadmin` Benutzerkonfiguriert für Ihre SVM.



Trident geht davon aus, als ein `vsadmin` SVM-Benutzer oder als Benutzer mit einem anderen Namen, der die gleiche Rolle hat. Amazon FSx for NetApp ONTAP hat ein `fsxadmin` Benutzer, der einen eingeschränkten Ersatz für ONTAP darstellt. `admin` Clusterbenutzer. Wir empfehlen dringend die Verwendung `vsadmin` mit Trident.

- Zertifikatsbasiert: Trident kommuniziert mit der SVM auf Ihrem FSx-Dateisystem mithilfe eines auf Ihrer SVM installierten Zertifikats.

Einzelheiten zur Aktivierung der Authentifizierung finden Sie in der Dokumentation zur Authentifizierung für Ihren Treibertyp:

- ["ONTAP NAS-Authentifizierung"](#)
- ["ONTAP SAN-Authentifizierung"](#)

Getestete Amazon Machine Images (AMIs)

Der EKS-Cluster unterstützt verschiedene Betriebssysteme, aber AWS hat bestimmte Amazon Machine Images (AMIs) für Container und EKS optimiert. Die folgenden AMIs wurden mit NetApp Trident 25.02 getestet.

AMI	NAS	NAS-Wirtschaft	iSCSI	iSCSI-Economy
AL2023_x86_64_STANDARD	Ja	Ja	Ja	Ja
AL2_x86_64	Ja	Ja	Ja*	Ja*
BOTTLEROCKET_x86_64	Ja**	Ja	k. A.	k. A.
AL2023_ARM_64_STANDARD	Ja	Ja	Ja	Ja
AL2_ARM_64	Ja	Ja	Ja*	Ja*

BOTTLEROCKET_A RM_64	Ja**	Ja	k. A.	k. A.
-------------------------	------	----	-------	-------

- * Das Löschen des PV ist ohne Neustart des Knotens nicht möglich
- ** Funktioniert nicht mit NFSv3 mit Trident Version 25.02.



Wenn Ihr gewünschtes AMI hier nicht aufgeführt ist, bedeutet das nicht, dass es nicht unterstützt wird; es bedeutet lediglich, dass es nicht getestet wurde. Diese Liste dient als Leitfaden für AMIs, von denen bekannt ist, dass sie funktionieren.

Tests durchgeführt mit:

- EKS-Version: 1.32
- Installationsmethode: Helm 25.06 und als AWS-Add-On 25.06
- Für NAS wurden sowohl NFSv3 als auch NFSv4.1 getestet.
- Für SAN wurde nur iSCSI getestet, nicht NVMe-oF.

Durchgeführte Tests:

- Erstellen: Speicherklasse, PVC, Kapsel
- Löschen: Pod, PVC (regulär, Qtree/LUN – Economy, NAS mit AWS-Backup)

Weitere Informationen

- ["Amazon FSx for NetApp ONTAP -Dokumentation"](#)
- ["Blogbeitrag über Amazon FSx for NetApp ONTAP"](#)

Erstellen Sie eine IAM-Rolle und ein AWS-Geheimnis.

Sie können Kubernetes-Pods so konfigurieren, dass sie auf AWS-Ressourcen zugreifen, indem sie sich als AWS-IAM-Rolle authentifizieren, anstatt explizite AWS-Anmeldeinformationen anzugeben.



Zur Authentifizierung mit einer AWS IAM-Rolle benötigen Sie einen Kubernetes-Cluster, der mit EKS bereitgestellt wurde.

AWS Secrets Manager-Geheimnis erstellen

Da Trident APIs an einen FSx vServer ausgibt, um den Speicher für Sie zu verwalten, benötigt es hierfür Anmeldeinformationen. Die sicherste Methode zur Übermittlung dieser Zugangsdaten ist die Verwendung eines AWS Secrets Manager-Geheimnisses. Wenn Sie also noch keines haben, müssen Sie ein AWS Secrets Manager-Geheimnis erstellen, das die Anmeldeinformationen für das vsadmin-Konto enthält.

Dieses Beispiel erstellt ein AWS Secrets Manager-Geheimnis zum Speichern von Trident CSI-Anmeldeinformationen:

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials"\
  --secret-string
"{\"username\": \"vsadmin\", \"password\": \"<svmpassword>\"}"
```

IAM-Richtlinie erstellen

Trident benötigt außerdem AWS-Berechtigungen, um korrekt ausgeführt werden zu können. Daher müssen Sie eine Richtlinie erstellen, die Trident die benötigten Berechtigungen erteilt.

Die folgenden Beispiele erstellen eine IAM-Richtlinie mithilfe der AWS CLI:

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy
-document file://policy.json
  --description "This policy grants access to Trident CSI to FSxN and
Secrets manager"
```

Beispiel für eine Richtlinien-JSON-Datei:

```

{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx>DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}

```

Pod-Identität oder IAM-Rolle für die Dienstkontozuordnung (IRSA) erstellen

Sie können ein Kubernetes-Servicekonto so konfigurieren, dass es eine AWS Identity and Access Management (IAM)-Rolle mit EKS Pod Identity oder IAM role for Service account association (IRSA) übernimmt. Alle Pods, die für die Verwendung des Servicekontos konfiguriert sind, können dann auf jeden AWS-Service zugreifen, für den die Rolle Berechtigungen besitzt.

Pod-Identität

Amazon EKS Pod Identity-Zuordnungen bieten die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon EC2-Instanzprofile Anmeldeinformationen für Amazon EC2-Instanzen bereitstellen.

Installieren Sie Pod Identity auf Ihrem EKS-Cluster:

Sie können eine Pod-Identität über die AWS-Konsole oder mithilfe des folgenden AWS CLI-Befehls erstellen:

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

Weitere Informationen finden Sie unter ["Amazon EKS Pod Identity Agent einrichten"](#) .

Erstelle trust-relationship.json:

Erstellen Sie eine trust-relationship.json-Datei, um dem EKS-Dienstprinzipal zu ermöglichen, diese Rolle für die Pod-Identität zu übernehmen. Erstellen Sie anschließend eine Rolle mit dieser Vertrauensrichtlinie:

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

trust-relationship.json-Datei:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

Die Rollenrichtlinie der IAM-Rolle zuordnen:

Weisen Sie der erstellten IAM-Rolle die Rollenrichtlinie aus dem vorherigen Schritt zu:

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \  
  --role-name fsxn-csi-role
```

Eine Pod-Identitätszuordnung erstellen:

Erstellen einer Pod-Identitätszuordnung zwischen der IAM-Rolle und dem Trident -Dienstkonto (trident-controller).

```
aws eks create-pod-identity-association \  
  --cluster-name <EKS_CLUSTER_NAME> \  
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \  
  --namespace trident --service-account trident-controller
```

IAM-Rolle für die Dienstkontozuordnung (IRSA)

Verwendung der AWS CLI:

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \  
  --assume-role-policy-document file://trust-relationship.json
```

trust-relationship.json-Datei:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-
provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub":
"system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

Aktualisieren Sie die folgenden Werte in der `trust-relationship.json` Datei:

- **<account_id>** - Ihre AWS-Konto-ID
- **<oidc_provider>** - Der OIDC Ihres EKS-Clusters. Sie können den `oidc_provider` durch Ausführen folgender Befehl erhalten:

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\
--output text | sed -e "s/^https:\\/\\/\\/"
```

Verknüpfen Sie die IAM-Rolle mit der IAM-Richtlinie:

Sobald die Rolle erstellt wurde, ordnen Sie die (im vorherigen Schritt erstellte) Richtlinie der Rolle mit diesem Befehl zu:

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

Überprüfen Sie, ob der OICD-Anbieter zugeordnet ist:

Vergewissern Sie sich, dass Ihr OIDC-Anbieter mit Ihrem Cluster verknüpft ist. Sie können dies mit diesem Befehl überprüfen:

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

Wenn die Ausgabe leer ist, verwenden Sie den folgenden Befehl, um IAM OIDC mit Ihrem Cluster zu verknüpfen:

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name  
--approve
```

Wenn Sie eksctl verwenden, nutzen Sie das folgende Beispiel, um eine IAM-Rolle für ein Dienstkonto in EKS zu erstellen:

```
eksctl create iamserviceaccount --name trident-controller --namespace  
trident \  
  --cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole  
--role-only \  
  --attach-policy-arn <IAM-Policy ARN> --approve
```

Trident installieren

Trident optimiert die Amazon FSx for NetApp ONTAP in Kubernetes, damit sich Ihre Entwickler und Administratoren auf die Anwendungsbereitstellung konzentrieren können.

Sie können Trident mit einer der folgenden Methoden installieren:

- Helm
- EKS-Add-on

Wenn Sie die Snapshot-Funktionalität nutzen möchten, installieren Sie das CSI Snapshot Controller Add-on. Siehe ["Snapshot-Funktionalität für CSI-Volumes aktivieren"](#) für weitere Informationen.

Installieren Sie Trident über Helm

Pod-Identität

1. Fügen Sie das Trident Helm-Repository hinzu:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Installieren Sie Trident anhand des folgenden Beispiels:

```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace
```

Sie können die `helm list` Befehl zum Überprüfen von Installationsdetails wie Name, Namespace, Chart, Status, App-Version und Revisionsnummer.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300 IDT		deployed	trident-operator-
100.2502.0	25.02.0		

Servicekonto-Zuordnung (IRSA)

1. Fügen Sie das Trident Helm-Repository hinzu:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Legen Sie die Werte für **Cloud-Anbieter** und **Cloud-Identität** fest:

```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 \  
--set cloudProvider="AWS" \  
--set cloudIdentity="'eks.amazonaws.com/role-arn:  
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>' " \  
--namespace trident \  
--create-namespace
```

Sie können die `helm list` Befehl zum Überprüfen von Installationsdetails wie Name, Namespace, Chart, Status, App-Version und Revisionsnummer.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300	IDT	deployed	trident-operator-
100.2506.0	25.06.0		

Wenn Sie iSCSI verwenden möchten, stellen Sie sicher, dass iSCSI auf Ihrem Client-Rechner aktiviert ist. Wenn Sie AL2023 Worker Node OS verwenden, können Sie die Installation des iSCSI-Clients automatisieren, indem Sie den Parameter „node prep“ in die Helm-Installation einfügen:



```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace --  
set nodePrep={iscsi}
```

Installieren Sie Trident über das EKS-Add-on

Das Trident EKS-Add-on enthält die neuesten Sicherheitspatches und Fehlerbehebungen und ist von AWS für die Verwendung mit Amazon EKS validiert. Mit dem EKS-Add-on können Sie sicherstellen, dass Ihre Amazon EKS-Cluster stets sicher und stabil sind und den Aufwand für die Installation, Konfiguration und Aktualisierung von Add-ons reduzieren.

Voraussetzungen

Stellen Sie sicher, dass Sie Folgendes haben, bevor Sie das Trident Add-on für AWS EKS konfigurieren:

- Ein Amazon EKS-Clusterkonto mit Zusatzabonnement
- AWS-Berechtigungen für den AWS Marketplace:
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AMI-Typ: Amazon Linux 2 (AL2_x86_64) oder Amazon Linux 2 Arm (AL2_ARM_64)
- Knotentyp: AMD oder ARM
- Ein bestehendes Amazon FSx for NetApp ONTAP Dateisystem

Aktivieren Sie das Trident Add-on für AWS.

Verwaltungskontrolle

1. Öffnen Sie die Amazon EKS-Konsole unter <https://console.aws.amazon.com/eks/home#/clusters>.
2. Im linken Navigationsbereich wählen Sie **Cluster** aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das NetApp Trident CSI-Add-on konfigurieren möchten.
4. Wählen Sie **Add-ons** und anschließend **Weitere Add-ons abrufen**.
5. Gehen Sie wie folgt vor, um das Add-on auszuwählen:
 - a. Scrollen Sie nach unten zum Abschnitt **AWS Marketplace Add-ons** und geben Sie **"Trident"** in das Suchfeld ein.
 - b. Aktivieren Sie das Kontrollkästchen in der oberen rechten Ecke des Feldes „Trident by NetApp“.
 - c. Wählen Sie **Weiter**.
6. Gehen Sie auf der Einstellungsseite **Ausgewählte Add-ons konfigurieren** wie folgt vor:



Überspringen Sie diese Schritte, wenn Sie die Pod Identity-Zuordnung verwenden.

- a. Wählen Sie die **Version** aus, die Sie verwenden möchten.
- b. Wenn Sie die IRSA-Authentifizierung verwenden, stellen Sie sicher, dass Sie die in den optionalen Konfigurationseinstellungen verfügbaren Konfigurationswerte festlegen:
 - Wählen Sie die **Version** aus, die Sie verwenden möchten.
 - Folgen Sie dem **Add-on-Konfigurationsschema** und legen Sie den Parameter **configurationValues** im Abschnitt **Konfigurationswerte** auf den Rollen-ARN fest, den Sie im vorherigen Schritt erstellt haben (der Wert sollte folgendes Format haben):

```
{  
  
  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",  
  "cloudProvider": "AWS"  
  
}
```

+

Wenn Sie bei der Konfliktlösungsmethode die Option „Überschreiben“ auswählen, können eine oder mehrere Einstellungen des bestehenden Add-ons mit den Einstellungen des Amazon EKS-Add-ons überschrieben werden. Wenn Sie diese Option nicht aktivieren und es zu einem Konflikt mit Ihren bestehenden Einstellungen kommt, schlägt der Vorgang fehl. Sie können die resultierende Fehlermeldung zur Fehlerbehebung des Konflikts verwenden. Bevor Sie diese Option auswählen, vergewissern Sie sich, dass das Amazon EKS-Add-on keine Einstellungen verwaltet, die Sie selbst verwalten müssen.

7. Wählen Sie **Weiter**.
8. Auf der Seite **Überprüfen und hinzufügen** wählen Sie **Erstellen**.

Nach Abschluss der Add-on-Installation wird Ihnen das installierte Add-on angezeigt.

AWS CLI

1. Erstellen Sie die add-on.json Datei:

Für die Pod-Identität verwenden Sie bitte folgendes Format:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

Für die IRSA-Authentifizierung verwenden Sie bitte folgendes Format:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```



Ersetzen <role ARN> mit dem ARN der Rolle, die im vorherigen Schritt erstellt wurde.

2. Installieren Sie das Trident EKS-Add-on.

```
aws eks create-addon --cli-input-json file://add-on.json
```

eksctl

Der folgende Beispielfehl installiert das Trident EKS-Add-on:

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

Aktualisieren Sie das Trident EKS-Add-on

Verwaltungskonsolle

1. Öffnen Sie die Amazon EKS-Konsole. <https://console.aws.amazon.com/eks/home#/clusters>.
2. Im linken Navigationsbereich wählen Sie **Cluster** aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das NetApp Trident CSI-Add-on aktualisieren möchten.
4. Wählen Sie den Reiter **Add-ons**.
5. Wählen Sie * Trident by NetApp* und anschließend **Bearbeiten**.
6. Führen Sie auf der Seite * Trident von NetApp konfigurieren* folgende Schritte aus:
 - a. Wählen Sie die **Version** aus, die Sie verwenden möchten.
 - b. Erweitern Sie die **Optionalen Konfigurationseinstellungen** und nehmen Sie bei Bedarf Anpassungen vor.
 - c. Wählen Sie **Änderungen speichern**.

AWS CLI

Das folgende Beispiel aktualisiert das EKS-Add-on:

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

eksctl

- Überprüfen Sie die aktuelle Version Ihres FSxN Trident CSI-Add-ons. Ersetzen my-cluster mit Ihrem Clusternamen.

```
eksctl get addon --name netapp_trident-operator --cluster my-cluster
```

Beispielausgabe:

NAME	VERSION	STATUS	ISSUES
IAMROLE	UPDATE AVAILABLE	CONFIGURATION VALUES	
netapp_trident-operator	v25.6.0-eksbuild.1	ACTIVE	0
{\"cloudIdentity\": \"'eks.amazonaws.com/role-arn: arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'\"}			

- Aktualisieren Sie das Add-on auf die Version, die im Ergebnis des vorherigen Schritts unter UPDATE AVAILABLE angezeigt wurde.

```
eksctl update addon --name netapp_trident-operator --version  
v25.6.0-eksbuild.1 --cluster my-cluster --force
```

Wenn Sie die `--force` Wenn eine der Optionen und eine der Einstellungen des Amazon EKS-Add-ons mit Ihren bestehenden Einstellungen in Konflikt steht und das Aktualisieren des Amazon EKS-Add-ons fehlschlägt, erhalten Sie eine Fehlermeldung, die Ihnen bei der Behebung des Konflikts hilft. Bevor Sie diese Option angeben, vergewissern Sie sich, dass das Amazon EKS-Add-on keine Einstellungen verwaltet, die Sie selbst verwalten müssen, da diese Einstellungen durch diese Option überschrieben werden. Weitere Informationen zu anderen Optionen für diese Einstellung finden Sie unter "[Addons](#)". Weitere Informationen zur Feldverwaltung von Amazon EKS Kubernetes finden Sie unter "[Kubernetes-Feldmanagement](#)".

Deinstallieren/entfernen Sie das Trident EKS-Add-on.

Sie haben zwei Möglichkeiten, ein Amazon EKS-Add-on zu entfernen:

- **Zusatzsoftware auf Ihrem Cluster beibehalten** – Diese Option entfernt die Verwaltung aller Einstellungen durch Amazon EKS. Außerdem entfällt dadurch die Möglichkeit für Amazon EKS, Sie über Aktualisierungen zu benachrichtigen und das Amazon EKS-Add-on automatisch zu aktualisieren, nachdem Sie eine Aktualisierung initiiert haben. Die Zusatzsoftware auf Ihrem Cluster bleibt jedoch erhalten. Diese Option macht das Add-on zu einer selbstverwalteten Installation und nicht zu einem Amazon EKS-Add-on. Bei dieser Option gibt es keine Ausfallzeiten für das Add-on. Behalten Sie die `--preserve` Option im Befehl zum Beibehalten des Add-ons.
- **Entfernen Sie die Add-on-Software vollständig aus Ihrem Cluster** – NetApp empfiehlt, das Amazon EKS-Add-on nur dann aus Ihrem Cluster zu entfernen, wenn keine Ressourcen in Ihrem Cluster davon abhängig sind. Entfernen Sie die `--preserve` Option aus der `delete` Befehl zum Entfernen des Add-ons.



Wenn dem Add-on ein IAM-Konto zugeordnet ist, wird das IAM-Konto nicht entfernt.

Verwaltungskonsolle

1. Öffnen Sie die Amazon EKS-Konsole unter <https://console.aws.amazon.com/eks/home#/clusters>.
2. Im linken Navigationsbereich wählen Sie **Cluster** aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das NetApp Trident CSI-Add-on entfernen möchten.
4. Wählen Sie die Registerkarte **Add-ons** und anschließend * Trident by NetApp*.
5. Wählen Sie **Entfernen**.
6. Führen Sie im Dialogfeld „Bestätigung zum Entfernen des netapp_trident-Operators“ folgende Schritte aus:
 - a. Wenn Sie nicht möchten, dass Amazon EKS die Einstellungen für das Add-on verwaltet, wählen Sie **Auf Cluster beibehalten**. Tun Sie dies, wenn Sie die Zusatzsoftware auf Ihrem Cluster behalten möchten, um alle Einstellungen des Zusatzes selbst verwalten zu können.
 - b. Geben Sie **netapp_trident-operator** ein.
 - c. Wählen Sie **Entfernen**.

AWS CLI

Ersetzen `my-cluster` mit dem Namen Ihres Clusters und führen Sie dann den folgenden Befehl aus.

```
aws eks delete-addon --cluster-name my-cluster --addon-name  
netapp_trident-operator --preserve
```

eksctl

Der folgende Befehl deinstalliert das Trident EKS-Add-on:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Konfigurieren des Speicher-Backends

ONTAP SAN- und NAS-Treiberintegration

Um ein Speicher-Backend zu erstellen, müssen Sie eine Konfigurationsdatei entweder im JSON- oder im YAML-Format erstellen. Die Datei muss den gewünschten Speichertyp (NAS oder SAN), das Dateisystem und die SVM, von der die Daten bezogen werden sollen, sowie die Art der Authentifizierung angeben. Das folgende Beispiel zeigt, wie Sie NAS-basierten Speicher definieren und ein AWS-Secret verwenden, um die Anmeldeinformationen für die gewünschte SVM zu speichern:

YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name"
    type: awsarn
```

JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

Führen Sie die folgenden Befehle aus, um die Trident Backend-Konfiguration (TBC) zu erstellen und zu validieren:

- Erstellen Sie eine Trident-Backend-Konfiguration (TBC) aus einer YAML-Datei und führen Sie folgenden Befehl aus:

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Überprüfen Sie, ob die Trident-Backend-Konfiguration (TBC) erfolgreich erstellt wurde:

```
Kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-tbc-ontap-nas	tbc-ontap-nas	933e0071-66ce-4324-
b9ff-f96d916ac5e9	Bound	Success

FSx für ONTAP Treiberdetails

Sie können Trident mit Amazon FSx for NetApp ONTAP mithilfe der folgenden Treiber integrieren:

- `ontap-san`: Jedes bereitgestellte PV ist eine LUN innerhalb eines eigenen Amazon FSx for NetApp ONTAP Volumes. Empfohlen für Blockspeicherung.
- `ontap-nas`: Jedes bereitgestellte PV ist ein vollständiges Amazon FSx for NetApp ONTAP -Volume. Empfohlen für NFS und SMB.
- `ontap-san-economy`: Jedes bereitgestellte PV ist eine LUN mit einer konfigurierbaren Anzahl von LUNs pro Amazon FSx for NetApp ONTAP Volume.
- `ontap-nas-economy`: Jedes bereitgestellte PV ist ein Qtree, wobei die Anzahl der Qtrees pro Amazon FSx for NetApp ONTAP Volume konfigurierbar ist.
- `ontap-nas-flexgroup`: Jedes bereitgestellte PV ist ein vollständiges Amazon FSx for NetApp ONTAP FlexGroup Volume.

Weitere Fahrerdetails finden Sie unter "[NAS-Treiber](#)" Und "[SAN-Treiber](#)".

Sobald die Konfigurationsdatei erstellt wurde, führen Sie diesen Befehl aus, um sie in Ihrem EKS zu erstellen:

```
kubectl create -f configuration_file
```

Um den Status zu überprüfen, führen Sie folgenden Befehl aus:

```
kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS		
backend-fsx-ontap-nas	backend-fsx-ontap-nas	7a551921-997c-4c37-a1d1-f2f4c87fa629
Bound	Success	

Erweiterte Backend-Konfiguration und Beispiele

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Beispiel
version		Immer 1
storageDriverName	Name des Speichertreibers	ontap-nas, ontap-nas-economy , ontap-nas-flexgroup , ontap-san , ontap-san-economy
backendName	Benutzerdefinierter Name oder das Speicher-Backend	Fahrername + "_" + dataLIF
managementLIF	IP-Adresse eines Clusters oder SVM-Management-LIF. Es kann ein vollqualifizierter Domänenname (FQDN) angegeben werden. Kann so eingestellt werden, dass IPv6-Adressen verwendet werden, wenn Trident mit dem IPv6-Flag installiert wurde. IPv6-Adressen müssen in eckigen Klammern angegeben werden, z. B. [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Wenn Sie die fsxFilesystemID unter dem aws Das Feld müssen Sie nicht angeben. managementLIF weil Trident die SVM abruft managementLIF Informationen von AWS. Sie müssen also Anmeldeinformationen für einen Benutzer unter der SVM angeben (z. B. vsadmin), und dieser Benutzer muss über die folgenden Berechtigungen verfügen: vsadmin Rolle.	„10.0.0.1“, „[2001:1234:abcd::fefe]“

Parameter	Beschreibung	Beispiel
dataLIF	IP-Adresse des Protokolls LIF. * ONTAP NAS-Treiber*: NetApp empfiehlt die Angabe von dataLIF. Falls keine Daten angegeben werden, ruft Trident die dataLIFs vom SVM ab. Sie können einen vollqualifizierten Domännennamen (FQDN) angeben, der für die NFS-Mount-Operationen verwendet werden soll. Dadurch können Sie ein Round-Robin-DNS erstellen, um die Last auf mehrere DataLIFs zu verteilen. Kann nach der Ersteinrichtung geändert werden. Siehe . * ONTAP SAN-Treiber*: Nicht für iSCSI angeben. Trident verwendet ONTAP Selective LUN Map, um die iSCSI LIFs zu ermitteln, die zum Aufbau einer Multipath-Sitzung benötigt werden. Es wird eine Warnung generiert, wenn dataLIF explizit definiert ist. Kann so eingestellt werden, dass IPv6-Adressen verwendet werden, wenn Trident mit dem IPv6-Flag installiert wurde. IPv6-Adressen müssen in eckigen Klammern angegeben werden, z. B. [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].	
autoExportPolicy	Automatische Erstellung und Aktualisierung von Exportrichtlinien aktivieren [Boolesch]. Verwenden des autoExportPolicy Und autoExportCIDRs Optionen: Trident kann Exportrichtlinien automatisch verwalten.	false
autoExportCIDRs	Liste der CIDRs, anhand derer die Kubernetes-Knoten-IPs gefiltert werden sollen, wenn autoExportPolicy ist aktiviert. Verwenden des autoExportPolicy Und autoExportCIDRs Optionen: Trident kann Exportrichtlinien automatisch verwalten.	"["0.0.0.0/0", "::/0"]"
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Datenträger angewendet werden sollen	""

Parameter	Beschreibung	Beispiel
clientCertificate	Base64-kodierter Wert des Clientzertifikats. Wird für zertifikatsbasierte Authentifizierung verwendet	""
clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für zertifikatsbasierte Authentifizierung verwendet	""
trustedCACertificate	Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Optional. Wird für die zertifikatsbasierte Authentifizierung verwendet.	""
username	Benutzername für die Verbindung zum Cluster oder zur SVM. Wird für die auf Anmeldeinformationen basierende Authentifizierung verwendet. Zum Beispiel vsadmin.	
password	Passwort zum Verbinden mit dem Cluster oder der SVM. Wird für die auf Anmeldeinformationen basierende Authentifizierung verwendet.	
svm	Zu verwendende virtuelle Speichermaschine	Wird abgeleitet, wenn ein SVM managementLIF angegeben ist.
storagePrefix	Präfix, das beim Bereitstellen neuer Volumes in der SVM verwendet wird. Kann nach der Erstellung nicht mehr geändert werden. Um diesen Parameter zu aktualisieren, müssen Sie ein neues Backend erstellen.	trident
limitAggregateUsage	Nicht für Amazon FSx for NetApp ONTAP angeben. Die bereitgestellten fsxadmin Und vsadmin enthalten nicht die erforderlichen Berechtigungen, um die aggregierte Nutzung abzurufen und sie mit Trident einzuschränken.	Nicht verwenden.

Parameter	Beschreibung	Beispiel
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet. Beschränkt außerdem die maximale Größe der von ihm verwalteten Volumes für Qtrees und LUNs, und die qtreesPerFlexvol. Diese Option ermöglicht die Anpassung der maximalen Anzahl von Qtrees pro FlexVol volume.	"" (wird nicht standardmäßig erzwungen)
lunsPerFlexvol	Die maximale Anzahl an LUNs pro Flexvol-Volume muss im Bereich [50, 200] liegen. Nur SAN.	"100"
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel: {"api":false, "method":true} Nicht verwenden debugTraceFlags. Es sei denn, Sie befinden sich in der Fehlersuche und benötigen einen detaillierten Protokollauszug.	null
nfsMountOptions	Durch Kommas getrennte Liste der NFS-Mount-Optionen. Die Mount-Optionen für Kubernetes-persistente Volumes werden normalerweise in Speicherklassen angegeben. Wenn jedoch in einer Speicherklasse keine Mount-Optionen angegeben sind, greift Trident auf die in der Konfigurationsdatei des Speicher-Backends angegebenen Mount-Optionen zurück. Wenn in der Speicherklasse oder der Konfigurationsdatei keine Mount-Optionen angegeben sind, setzt Trident keine Mount-Optionen auf einem zugehörigen persistenten Volume.	""
nasType	Konfiguration der Erstellung von NFS- oder SMB-Volumes. Optionen sind nfs, smb oder null. Muss eingestellt werden auf smb für SMB-Volumes. Bei der Einstellung „null“ werden standardmäßig NFS-Volumes verwendet.	nfs
qtreesPerFlexvol	Die maximale Anzahl an Qtrees pro FlexVol volume muss im Bereich [50, 300] liegen.	"200"

Parameter	Beschreibung	Beispiel
smbShare	Sie können entweder den Namen einer SMB-Freigabe angeben, die mit der Microsoft Management Console oder der ONTAP CLI erstellt wurde, oder einen Namen, unter dem Trident die SMB-Freigabe erstellen kann. Dieser Parameter ist für Amazon FSx for ONTAP -Backends erforderlich.	smb-share
useREST	Boolescher Parameter zur Verwendung von ONTAP REST-APIs. Wenn eingestellt auf <code>true</code> Trident wird ONTAP REST APIs zur Kommunikation mit dem Backend verwenden. Diese Funktion erfordert ONTAP 9.11.1 und höher. Darüber hinaus muss die verwendete ONTAP Anmelderolle Zugriff auf die <code>ontap</code> Anwendung. Dies wird durch die vordefinierte Bedingung erfüllt. <code>vsadmin</code> Und <code>cluster-admin</code> Rollen.	false
aws	In der Konfigurationsdatei für AWS FSx für ONTAP können Sie Folgendes angeben: - <code>fsxFilesystemID</code> : Geben Sie die ID des AWS FSx-Dateisystems an. - <code>apiRegion</code> : Name der AWS-API-Region. - <code>apikey</code> : AWS-API-Schlüssel. - <code>secretKey</code> : AWS-Geheimschlüssel.	"" "" ""
credentials	Geben Sie die FSx SVM-Anmeldeinformationen an, die im AWS Secrets Manager gespeichert werden sollen. - <code>name</code> : Amazon Resource Name (ARN) des Geheimnisses, das die Anmeldeinformationen von SVM enthält. - <code>type</code> : Aufstellen <code>awsarn</code> . Siehe "Erstellen Sie ein AWS Secrets Manager-Geheimnis" für weitere Informationen.	

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

Sie können die Standardbereitstellung mithilfe dieser Optionen steuern. `defaults` Abschnitt der Konfiguration. Ein Beispiel finden Sie in den folgenden Konfigurationsbeispielen.

Parameter	Beschreibung	Standard
spaceAllocation	Speicherplatzzuweisung für LUNs	true
spaceReserve	Platzreservierungsmodus; "keine" (dünn) oder "Volumen" (dick)	none
snapshotPolicy	Zu verwendende Snapshot-Richtlinie	none
qosPolicy	Die QoS-Richtliniengruppe soll den erstellten Volumes zugewiesen werden. Wählen Sie pro Speicherpool oder Backend entweder qosPolicy oder adaptiveQosPolicy aus. Die Verwendung von QoS-Richtliniengruppen mit Trident erfordert ONTAP 9.8 oder höher. Sie sollten eine nicht gemeinsam genutzte QoS-Richtliniengruppe verwenden und sicherstellen, dass die Richtliniengruppe auf jeden einzelnen Bestandteil angewendet wird. Eine gemeinsam genutzte QoS-Richtliniengruppe setzt die Obergrenze für den Gesamtdurchsatz aller Workloads durch.	""
adaptiveQosPolicy	Adaptive QoS-Richtliniengruppe, die den erstellten Volumes zugewiesen werden soll. Wählen Sie pro Speicherpool oder Backend entweder qosPolicy oder adaptiveQosPolicy aus. Wird von ontap-nas-economy nicht unterstützt.	""
snapshotReserve	Prozentsatz des für Snapshots reservierten Speichervolumens „0“	Wenn snapshotPolicy ist none , else ""
splitOnClone	Beim Erstellen eines Klons diesen von seinem Elternklon trennen	false
encryption	Aktivieren Sie die NetApp Volumeverschlüsselung (NVE) auf dem neuen Volume; Standardwert ist false . Um diese Option nutzen zu können, muss NVE auf dem Cluster lizenziert und aktiviert sein. Wenn NAE im Backend aktiviert ist, wird jedes in Trident bereitgestellte Volume NAE-fähig sein. Weitere Informationen finden Sie unter: "Wie Trident mit NVE und NAE zusammenarbeitet" .	false

Parameter	Beschreibung	Standard
luksEncryption	LUKS-Verschlüsselung aktivieren. Siehe " Verwenden Sie Linux Unified Key Setup (LUKS) ". Nur SAN.	""
tieringPolicy	zu verwendende Stufenrichtlinie none	
unixPermissions	Modus für neue Volumes. Für SMB-Volumes leer lassen.	""
securityStyle	Sicherheitsstil für neue Bände. NFS unterstützt <code>mixed</code> Und <code>unix</code> Sicherheitsstile. SMB-Unterstützung <code>mixed</code> Und <code>ntfs</code> Sicherheitsstile.	NFS-Standard ist <code>unix</code> . SMB-Standard ist <code>ntfs</code> .

Bereiten Sie die Bereitstellung von SMB-Volumes vor

Sie können SMB-Volumes mithilfe von ... bereitstellen. `ontap-nas` Treiber. Bevor Sie fertigstellen [ONTAP SAN- und NAS-Treiberintegration](#) Führen Sie die folgenden Schritte aus.

Bevor Sie beginnen

Bevor Sie SMB-Volumes mithilfe von `ontap-nas` Als Fahrer benötigen Sie Folgendes:

- Ein Kubernetes-Cluster mit einem Linux-Controller-Knoten und mindestens einem Windows-Worker-Knoten, auf dem Windows Server 2019 ausgeführt wird. Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten laufen.
- Mindestens ein Trident Geheimnis, das Ihre Active Directory-Anmeldeinformationen enthält. Um Geheimnisse zu generieren `smbcreds` :

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Ein als Windows-Dienst konfigurierter CSI-Proxy. Um einen zu konfigurieren `csi-proxy` , siehe "[GitHub: CSI-Proxy](#)" oder "[GitHub: CSI-Proxy für Windows](#)" für Kubernetes-Knoten, die unter Windows laufen.

Schritte

1. SMB-Freigaben erstellen. Sie können die SMB-Administratorfreigaben auf zwei Arten erstellen, entweder mithilfe von "[Microsoft Management Console](#)" Über das Snap-In „Freigegebene Ordner“ oder über die ONTAP -Befehlszeilenschnittstelle. So erstellen Sie die SMB-Freigaben mithilfe der ONTAP -Befehlszeilenschnittstelle:

- a. Erstellen Sie gegebenenfalls die Verzeichnispfadstruktur für die Freigabe.

Der `vserver cifs share create` Der Befehl überprüft den Pfad, der bei der Erstellung der Freigabe in der Option `-path` angegeben wurde. Wenn der angegebene Pfad nicht existiert, schlägt der Befehl fehl.

- b. Erstellen Sie eine SMB-Freigabe, die dem angegebenen SVM zugeordnet ist:

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

c. Überprüfen Sie, ob die Freigabe erstellt wurde:

```
vserver cifs share show -share-name share_name
```



Siehe ["Erstellen einer SMB-Freigabe"](#) Für alle Details.

- Bei der Erstellung des Backends müssen Sie Folgendes konfigurieren, um SMB-Volumes anzugeben. Alle Konfigurationsoptionen für das FSx for ONTAP Backend finden Sie unter ["FSx für ONTAP: Konfigurationsoptionen und Beispiele"](#).

Parameter	Beschreibung	Beispiel
smbShare	Sie können entweder den Namen einer SMB-Freigabe angeben, die mit der Microsoft Management Console oder der ONTAP CLI erstellt wurde, oder einen Namen, unter dem Trident die SMB-Freigabe erstellen kann. Dieser Parameter ist für Amazon FSx for ONTAP -Backends erforderlich.	smb-share
nasType	Muss eingestellt werden auf smb . Wenn null, wird standardmäßig der Wert verwendet. <code>nfs</code> .	smb
securityStyle	Sicherheitsstil für neue Bände. Muss eingestellt sein auf ntfs oder mixed für SMB-Volumes.	ntfs oder mixed für SMB-Volumes
unixPermissions	Modus für neue Volumes. Muss bei SMB-Volumes leer bleiben.	""

Konfigurieren Sie eine Speicherklasse und einen PVC.

Konfigurieren Sie ein Kubernetes StorageClass-Objekt und erstellen Sie die Storage-Klasse, um Trident anzuweisen, wie Volumes bereitgestellt werden sollen. Erstellen Sie einen PersistentVolumeClaim (PVC), der die konfigurierte Kubernetes StorageClass verwendet, um Zugriff auf das PV anzufordern. Anschließend können Sie die PV-Anlage an einem Pod montieren.

Erstellen einer Speicherklasse

Konfigurieren eines Kubernetes StorageClass-Objekts

Der **"Kubernetes StorageClass-Objekt"** Das Objekt identifiziert Trident als den für diese Klasse verwendeten Provisionierer und weist Trident an, wie ein Volume zu provisionieren ist. Verwenden Sie dieses Beispiel, um die Speicherklasse für Volumes mit NFS einzurichten (die vollständige Liste der Attribute finden Sie im Abschnitt „Trident -Attribute“ weiter unten):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"
```

Verwenden Sie dieses Beispiel, um die Speicherklasse für Volumes mit iSCSI einzurichten:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  provisioningType: "thin"
  snapshots: "true"
```

Um NFSv3-Volumes auf AWS Bottlerocket bereitzustellen, fügen Sie die erforderlichen Komponenten hinzu. mountOptions zur Speicherklasse:

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock

```

Siehe "[Kubernetes- und Trident Objekte](#)" Einzelheiten darüber, wie Speicherklassen mit dem interagieren, finden Sie hier. PersistentVolumeClaim und Parameter zur Steuerung der Volumenbereitstellung Trident .

Erstellen einer Speicherklasse

Schritte

1. Dies ist ein Kubernetes-Objekt, also verwenden Sie `kubectl` um es in Kubernetes zu erstellen.

```
kubectl create -f storage-class-ontapnas.yaml
```

2. Sie sollten nun sowohl in Kubernetes als auch in Trident eine **basic-csi**-Speicherklasse sehen, und Trident sollte die Pools im Backend erkannt haben.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

PVC erstellen

A "[PersistentVolumeClaim](#)" (PVC) ist eine Anfrage für den Zugriff auf das PersistentVolume im Cluster.

Das PVC kann so konfiguriert werden, dass es Speicherplatz einer bestimmten Größe oder einen bestimmten Zugriffsmodus anfordert. Mithilfe der zugehörigen StorageClass kann der Clusteradministrator mehr als nur die Größe und den Zugriffsmodus des PersistentVolumes steuern – beispielsweise die Leistung oder das Servicelevel.

Nachdem Sie das PVC erstellt haben, können Sie das Volumen in einem Gehäuse montieren.

Beispielmanifeste

Beispielmanifeste für PersistentVolumeClaim

Diese Beispiele zeigen grundlegende PVC-Konfigurationsoptionen.

PVC mit RWX-Zugang

Dieses Beispiel zeigt eine einfache PVC mit RWX-Zugriff, die einer StorageClass namens zugeordnet ist. `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

PVC mit iSCSI-Beispiel

Dieses Beispiel zeigt eine einfache PVC für iSCSI mit RWO-Zugriff, die einer StorageClass namens zugeordnet ist. `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

PVC erstellen

Schritte

1. Erstellen Sie die PVC.

```
kubectl create -f pvc.yaml
```

2. Überprüfen Sie den PVC-Status.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	2Gi	RWO		5m

Siehe "[Kubernetes- und Trident Objekte](#)" Einzelheiten darüber, wie Speicherklassen mit dem interagieren, finden Sie hier. `PersistentVolumeClaim` und Parameter zur Steuerung der Volumenbereitstellung Trident .

Trident Eigenschaften

Diese Parameter legen fest, welche von Trident verwalteten Speicherpools zur Bereitstellung von Volumes eines bestimmten Typs verwendet werden sollen.

Attribut	Typ	Werte	Angebot	Anfrage	Unterstützt von
media ¹	Schnur	HDD, Hybrid, SSD	Der Pool enthält Medien dieses Typs; hybrid bedeutet beides	Medientyp angeben	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
Bereitstellungstyp	Schnur	dünn, dick	Pool unterstützt diese Bereitstellungsmethode	Bereitstellungsmethode angeben	dick: alles vom Fass; dünn: alles vom Fass & Solidfire-San
Backend-Typ	Schnur	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Pool gehört zu dieser Art von Backend.	Backend spezifiziert	Alle Fahrer
Momentaufnahme	bool	wahr, falsch	Pool unterstützt Volumes mit Snapshots	Volume mit aktivierten Snapshots	ontap-nas, ontap-san, solidfire-san, gcp-cvs
Klone	bool	wahr, falsch	Pool unterstützt das Klonen von Volumes	Volume mit aktivierten Klonen	ontap-nas, ontap-san, solidfire-san, gcp-cvs

Attribut	Typ	Werte	Angebot	Anfrage	Unterstützt von
Verschlüsselung	bool	wahr, falsch	Pool unterstützt verschlüsselte Volumes	Volume mit aktivierter Verschlüsselung	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	int	positive ganze Zahl	Pool ist in der Lage, IOPS in diesem Bereich zu garantieren.	Volumen garantiert diese IOPS	solidfire-san

¹: Wird von ONTAP Select -Systemen nicht unterstützt.

Beispielanwendung bereitstellen

Sobald die Speicherklasse und das PVC erstellt sind, können Sie das PV an einem Pod montieren. Dieser Abschnitt listet den Beispielbefehl und die Konfiguration zum Anhängen des PV an einen Pod auf.

Schritte

1. Montieren Sie das Volume in einem Gehäuse.

```
kubectl create -f pv-pod.yaml
```

Diese Beispiele zeigen grundlegende Konfigurationen zum Anbringen des PVC an eine Kapsel:

Grundkonfiguration:

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
        claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```



Sie können den Fortschritt überwachen mit `kubectl get pod --watch`.

2. Überprüfen Sie, ob das Volume eingebunden ist. `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

Filesystem	Size
Used Avail Use% Mounted on	
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06	1.1G
320K 1.0G 1% /my/mount/path	

Sie können den Pod jetzt löschen. Die Pod-Anwendung wird nicht mehr existieren, das Volume bleibt jedoch erhalten.

```
kubectl delete pod pv-pod
```

Konfigurieren Sie das Trident EKS-Add-on auf einem EKS-Cluster

NetApp Trident optimiert die Amazon FSx for NetApp ONTAP in Kubernetes, damit sich Ihre Entwickler und Administratoren auf die Anwendungsbereitstellung konzentrieren können. Das NetApp Trident EKS-Add-on enthält die neuesten Sicherheitspatches und Fehlerbehebungen und ist von AWS für die Verwendung mit Amazon EKS validiert. Mit dem EKS-Add-on können Sie sicherstellen, dass Ihre Amazon EKS-Cluster stets sicher und stabil sind und den Aufwand für die Installation, Konfiguration und Aktualisierung von Add-ons reduzieren.

Voraussetzungen

Stellen Sie sicher, dass Sie Folgendes haben, bevor Sie das Trident Add-on für AWS EKS konfigurieren:

- Ein Amazon EKS-Cluster-Konto mit Berechtigungen zur Arbeit mit Add-ons. Siehe ["Amazon EKS-Add-ons"](#).
- AWS-Berechtigungen für den AWS Marketplace:
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AMI-Typ: Amazon Linux 2 (AL2_x86_64) oder Amazon Linux 2 Arm (AL2_ARM_64)
- Knotentyp: AMD oder ARM
- Ein bestehendes Amazon FSx for NetApp ONTAP Dateisystem

Schritte

1. Stellen Sie sicher, dass Sie eine IAM-Rolle und ein AWS-Secret erstellen, damit EKS-Pods auf AWS-Ressourcen zugreifen können. Anweisungen finden Sie unter ["Erstellen Sie eine IAM-Rolle und ein AWS-](#)

Geheimnis." .

2. Navigieren Sie in Ihrem EKS Kubernetes-Cluster zum Tab **Add-ons**.

tri-env-eks Refresh Delete cluster Upgrade version View dashboard

① End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the [pricing page](#). Upgrade now

▼ Cluster info [Info](#)
Status
✓ Active
Cluster health issues
✓ 0
Kubernetes version [Info](#)
1.30
Support period
① Standard support until July 28, 2025
Provider
EKS
Upgrade insights
✓ 0

Overview

Resources

Compute

Networking

Add-ons 1

Access

Observability

Update history

Tags

① New versions are available for 1 add-on. ×

Add-ons (3) [Info](#) View details Edit Remove Get more add-ons

Any categ... Any status 3 matches < 1 >

3. Gehen Sie zu **AWS Marketplace Add-ons** und wählen Sie die Kategorie *Speicher* aus.


AWS Marketplace add-ons (1) Refresh

Discover, subscribe to and configure EKS add-ons to enhance your EKS clusters.

Filtering options

Any category NetApp, Inc. Any pricing model Clear filters

NetApp, Inc. X < 1 >

**NetApp Trident**

NetApp Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)

Standard Contract

Category storage	Listed by NetApp, Inc.	Supported versions 1.31, 1.30, 1.29, 1.28, 1.27, 1.26, 1.25, 1.24, 1.23	Pricing starting at View pricing details
----------------------------	--------------------------------------------------	-----------------------------------------------------------------------------------------	--------------------------------------------------------------------

Cancel Next

4. Suchen Sie * NetApp Trident* und aktivieren Sie das Kontrollkästchen für das Trident Add-on. Klicken Sie anschließend auf **Weiter**.

5. Wählen Sie die gewünschte Version des Add-ons.

Configure selected add-ons settings


Configure the add-ons for your cluster by selecting settings.

NetApp TridentRemove add-on

Listed by

Category

Status



storage

Ready to install

You're subscribed to this software

You can view the terms and pricing details for this product or choose another offer if one is available.

View subscription

×

Version

Select the version for this add-on.

v25.6.0-eksbuild.1

Optional configuration settings

Cancel

Previous

Next

6. Konfigurieren Sie die erforderlichen Add-On-Einstellungen.

Review and add

Step 1: Select add-ons

[Edit](#)

Selected add-ons (1)

Find add-on

< 1 >

Add-on name	Type	Status
netapp_trident-operator	storage	Ready to install

Step 2: Configure selected add-ons settings

[Edit](#)

Selected add-ons version (1)

< 1 >

Add-on name	Version	IAM role for service account (IRSA)
netapp_trident-operator	v24.10.0-eksbuild.1	Not set

EKS Pod Identity (0)

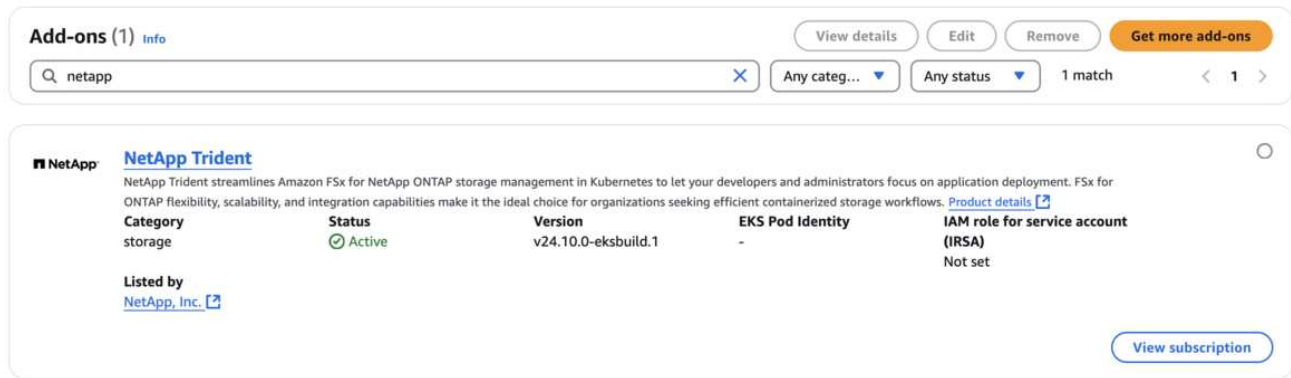
< 1 >

Add-on name	IAM role	Service account
No Pod Identity associations None of the selected add-on(s) have Pod Identity associations.		

[Cancel](#)[Previous](#)[Create](#)

- Wenn Sie IRSA (IAM-Rollen für Dienstkonten) verwenden, beachten Sie die zusätzlichen Konfigurationsschritte. ["hier,"](#) .
- Wählen Sie **Erstellen**.

9. Überprüfen Sie, ob der Status des Add-ons *Aktiv* lautet.



10. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Trident ordnungsgemäß auf dem Cluster installiert ist:

```
kubectl get pods -n trident
```

11. Setzen Sie die Einrichtung fort und konfigurieren Sie das Speicher-Backend. Weitere Informationen finden Sie unter "[Konfigurieren des Speicher-Backends](#)".

Installation/Deinstallation des Trident EKS-Add-ons über die Befehlszeile

Installieren Sie das NetApp Trident EKS-Add-on über die Befehlszeile:

Der folgende Beispielbefehl installiert das Trident EKS-Add-on:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.0-eksbuild.1 (mit einer speziellen Version)
```

Deinstallieren Sie das NetApp Trident EKS-Add-on über die Befehlszeile:

Der folgende Befehl deinstalliert das Trident EKS-Add-on:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Backends mit kubectl erstellen

Ein Backend definiert die Beziehung zwischen Trident und einem Speichersystem. Es teilt Trident mit, wie mit diesem Speichersystem kommuniziert werden soll und wie Trident Datenträger daraus bereitstellen soll. Nach der Installation von Trident besteht der nächste Schritt darin, ein Backend zu erstellen. Der `TridentBackendConfig` Mit Custom Resource Definition (CRD) können Sie Trident -Backends direkt über die Kubernetes-Schnittstelle erstellen und verwalten. Dies können Sie tun, indem Sie `kubectl` oder das entsprechende CLI-Tool für Ihre Kubernetes-Distribution.

TridentBackendConfig

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) ist ein Frontend mit Namensräumen, das es Ihnen ermöglicht, Trident -Backends zu verwalten. `kubectl`. Kubernetes- und Speicheradministratoren

können Backends jetzt direkt über die Kubernetes-CLI erstellen und verwalten, ohne dass ein separates Befehlszeilenprogramm erforderlich ist.(`tridentctl`).

Bei der Schaffung eines `TridentBackendConfig` Wenn man sich das Objekt ansieht, geschieht Folgendes:

- Ein Backend wird von Trident automatisch auf Basis der von Ihnen bereitgestellten Konfiguration erstellt. Dies wird intern dargestellt als `TridentBackend (tbe , tridentbackend)` CR.
- Der `TridentBackendConfig` ist auf einzigartige Weise an ein `TridentBackend` Das wurde von Trident entwickelt.

Jede `TridentBackendConfig` pflegt eine Eins-zu-Eins-Zuordnung mit einem `TridentBackend` Ersteres ist die dem Benutzer zur Verfügung gestellte Schnittstelle zum Entwerfen und Konfigurieren von Backends; letzteres ist die Art und Weise, wie Trident das eigentliche Backend-Objekt darstellt.



`TridentBackend`CRs` werden von Trident automatisch erstellt. Sie **sollten** sie nicht verändern. Wenn Sie Aktualisierungen an den Backends vornehmen möchten, tun Sie dies durch Ändern der ``TridentBackendConfig` Objekt.

Das folgende Beispiel zeigt das Format des `TridentBackendConfig` CR:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Sie können sich auch die Beispiele in der "[Dreizack-Installateur](#)" Verzeichnis mit Beispielkonfigurationen für die gewünschte Speicherplattform/den gewünschten Speicherdienst.

Der `spec` benötigt backendspezifische Konfigurationsparameter. In diesem Beispiel verwendet das Backend die `ontap-san` Der Speichertreiber verwendet die hier tabellarisch aufgeführten Konfigurationsparameter. Eine Liste der Konfigurationsoptionen für Ihren gewünschten Speichertreiber finden Sie unter "[Backend-Konfigurationsinformationen für Ihren Speichertreiber](#)".

Der `spec` Dieser Abschnitt umfasst auch `credentials` Und `deletionPolicy` Felder, die neu eingeführt wurden in der `TridentBackendConfig` CR:

- ``credentials`` Dieser Parameter ist ein Pflichtfeld und enthält die Anmeldeinformationen, die zur Authentifizierung beim Speichersystem/-dienst verwendet werden. Dies ist auf ein vom Benutzer erstelltes Kubernetes-Secret eingestellt. Die Zugangsdaten dürfen nicht im Klartext übermittelt werden und führen zu

einem Fehler.

- `deletionPolicy` Dieses Feld definiert, was geschehen soll, wenn `TridentBackendConfig` gelöscht wird. Es kann einen von zwei möglichen Werten annehmen:
 - `delete` Dies führt zur Löschung beider `TridentBackendConfig` CR und das zugehörige Backend. Dies ist der Standardwert.
 - `retain` Wenn ein `TridentBackendConfig` CR gelöscht wird, bleibt die Backend-Definition weiterhin vorhanden und kann verwaltet werden mit `tridentctl`. Die Löschroutine festlegen `retain` Ermöglicht es Benutzern, auf eine frühere Version (vor 21.04) zurückzustufen und die erstellten Backends beizubehalten. Der Wert dieses Feldes kann nach einem `TridentBackendConfig` erstellt werden.



Der Name eines Backends wird wie folgt festgelegt: `spec.backendName`. Wenn kein Name angegeben ist, wird der Name des Backends auf den Namen des Backends gesetzt. `TridentBackendConfig` Objekt (`metadata.name`). Es wird empfohlen, Backend-Namen explizit festzulegen. `spec.backendName`.



Backends, die mit `tridentctl` haben keine zugehörige `TridentBackendConfig` Objekt. Sie können diese Backends mit folgender Funktion verwalten: `kubectl` durch die Erstellung eines `TridentBackendConfig` CR. Es ist darauf zu achten, dass identische Konfigurationsparameter angegeben werden (wie z. B. `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, und so weiter). Trident wird das neu erstellte automatisch binden. `TridentBackendConfig` mit dem bereits bestehenden Backend.

Schrittübersicht

Um ein neues Backend zu erstellen, indem man `kubectl` Sie sollten Folgendes tun:

1. Erstellen Sie ein **"Kubernetes-Geheimnis"** Das Geheimnis enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster/-dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieser Eintrag enthält spezifische Informationen zum Speichercluster/Speicherdienst und verweist auf das im vorherigen Schritt erstellte Geheimnis.

Nachdem Sie ein Backend erstellt haben, können Sie dessen Status mithilfe von [Name des Dienstes/der Funktion] beobachten. `kubectl get tbc <tbc-name> -n <trident-namespace>` und weitere Details einholen.

Schritt 1: Erstellen Sie ein Kubernetes-Secret.

Erstellen Sie ein Geheimnis, das die Zugangsdaten für das Backend enthält. Dies ist für jeden Speicherdienst bzw. jede Speicherplattform individuell. Hier ist ein Beispiel:

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```

apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password

```

Diese Tabelle fasst die Felder zusammen, die für jede Speicherplattform im Secret enthalten sein müssen:

Beschreibung der geheimen Felder der Speicherplattform	Geheimnis	Feldbeschreibung
Azure NetApp Files	Client-ID	Die Client-ID aus einer App-Registrierung
Cloud Volumes Service für GCP	private_key_id	ID des privaten Schlüssels. Teil des API-Schlüssels für ein GCP-Dienstkonto mit CVS-Administratorrolle
Cloud Volumes Service für GCP	privater_key	Privater Schlüssel. Teil des API-Schlüssels für ein GCP-Dienstkonto mit CVS-Administratorrolle
Element (NetApp HCI/ SolidFire)	Endpunkt	MVIP für den SolidFire -Cluster mit Mandantenanmeldeinformationen
ONTAP	Benutzername	Benutzername für die Verbindung mit dem Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet
ONTAP	Passwort	Kennwort für die Verbindung mit dem Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet
ONTAP	Client-Privatschlüssel	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für die zertifikatsbasierte Authentifizierung verwendet

Beschreibung der geheimen Felder der Speicherplattform	Geheimnis	Feldbeschreibung
ONTAP	chapUsername	Eingehender Benutzername. Erforderlich, wenn useCHAP=true. Für ontap-san Und ontap-san-economy
ONTAP	KapitelInitiatorGeheimnis	Geheimnis des CHAP-Initiators. Erforderlich, wenn useCHAP=true. Für ontap-san Und ontap-san-economy
ONTAP	chapTargetUsername	Zielbenutzername. Erforderlich, wenn useCHAP=true. Für ontap-san Und ontap-san-economy
ONTAP	Kapitel „Zielinitiatorgeheimnis“	Geheimnis des CHAP-Zielinitiators. Erforderlich, wenn useCHAP=true. Für ontap-san Und ontap-san-economy

Das in diesem Schritt erstellte Geheimnis wird in der `spec.credentials` Feld des `TridentBackendConfig` Objekt, das im nächsten Schritt erstellt wird.

Schritt 2: Erstellen Sie die `TridentBackendConfig` CR

Sie können nun Ihre eigene erstellen. `TridentBackendConfig` CR. In diesem Beispiel verwendet ein Backend die `ontap-san` Der Treiber wird mithilfe des `TridentBackendConfig` Das unten abgebildete Objekt:

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret

```

Schritt 3: Überprüfen Sie den Status des TridentBackendConfig CR

Nachdem Sie nun die TridentBackendConfig CR, Sie können den Status überprüfen. Siehe das folgende Beispiel:

```

kubectl -n trident get tbc backend-tbc-ontap-san

```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
Bound	Success	

Es wurde erfolgreich ein Backend erstellt und angebunden. TridentBackendConfig CR.

Die Phase kann einen der folgenden Werte annehmen:

- **Bound:** Der TridentBackendConfig CR ist mit einem Backend verknüpft, und dieses Backend enthält configRef auf die TridentBackendConfig CR's uid.
- **Unbound:** Dargestellt durch "". Der TridentBackendConfig Das Objekt ist nicht an ein Backend gebunden. Alle neu erstellt TridentBackendConfig CRs befinden sich standardmäßig in dieser Phase. Nach den Phasenübergängen kann der Zustand „Ungebunden“ nicht mehr rückgängig gemacht werden.
- **Deleting:** Der TridentBackendConfig CRs deletionPolicy wurde zum Löschen ausgewählt. Wenn die TridentBackendConfig Wenn CR gelöscht wird, wechselt es in den Status „Wird gelöscht“.
 - Wenn im Backend keine persistenten Volumenansprüche (PVCs) vorhanden sind, wird deren gelöscht. TridentBackendConfig wird dazu führen, dass Trident sowohl das Backend als auch das TridentBackendConfig CR.
 - Wenn im Backend ein oder mehrere PVCs vorhanden sind, wechselt das System in den Löschzustand. Der TridentBackendConfig Anschließend tritt CR auch in die Löschphase ein. Das Backend und TridentBackendConfig werden erst gelöscht, nachdem alle PVCs gelöscht wurden.
- **Lost:** Das zugehörige Backend TridentBackendConfig CR wurde versehentlich oder absichtlich gelöscht und die TridentBackendConfig CR enthält noch einen Verweis auf das gelöschte Backend.

Der `TridentBackendConfig` CR kann unabhängig davon weiterhin gelöscht werden. `deletionPolicy` Wert.

- Unknown`Trident kann den Status oder die Existenz des zugehörigen Backends nicht feststellen. `TridentBackendConfig CR. Wenn beispielsweise der API-Server nicht antwortet oder wenn der `tridentbackends.trident.netapp.io` CRD fehlt. Dies könnte ein Eingreifen erfordern.

In diesem Stadium ist das Backend erfolgreich erstellt worden! Es gibt mehrere weitere Vorgänge, die zusätzlich durchgeführt werden können, wie zum Beispiel ["Backend-Aktualisierungen und Backend-Löschungen"](#) .

(Optional) Schritt 4: Weitere Details einholen.

Sie können den folgenden Befehl ausführen, um weitere Informationen über Ihr Backend zu erhalten:

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

NAME	BACKEND NAME	BACKEND UUID		
PHASE	STATUS	STORAGE DRIVER	DELETION POLICY	
backend-tbc-ontap-san	ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-		
bab2699e6ab8	Bound	Success	ontap-san	delete

Darüber hinaus können Sie auch einen YAML/JSON-Dump erhalten von `TridentBackendConfig` .

```
kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound

```

backendInfo`enthält `backendName und die backendUUID des Backends, das als Reaktion auf die TridentBackendConfig CR. Der lastOperationStatus Das Feld repräsentiert den Status der letzten Operation des TridentBackendConfig CR, die durch einen Benutzer ausgelöst werden kann (z. B. wenn ein Benutzer etwas geändert hat in spec oder ausgelöst durch Trident (zum Beispiel während eines Trident -Neustarts). Es kann entweder ein Erfolg oder ein Misserfolg sein. phase stellt den Status der Beziehung zwischen den dar TridentBackendConfig CR und das Backend. Im obigen Beispiel phase hat den Wert „Gebunden“, was bedeutet, dass TridentBackendConfig CR ist mit dem Backend verbunden.

Sie können die `kubectl -n trident describe tbc <tbc-cr-name>` Befehl zum Abrufen von Details zu den Ereignisprotokollen.



Ein Backend, das ein zugehöriges Backend enthält, kann nicht aktualisiert oder gelöscht werden. TridentBackendConfig Objekt verwenden `tridentctl`. Um die einzelnen Schritte beim Wechsel zwischen `tridentctl` Und `TridentBackendConfig`, "[siehe hier](#)".

Backends verwalten

Führen Sie die Backend-Verwaltung mit `kubectl` durch.

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge durchführen, indem Sie `kubectl` .

Backend löschen

Durch das Löschen eines `TridentBackendConfig` Sie weisen Trident an, Backends zu löschen/beizubehalten (basierend auf `deletionPolicy`). Um ein Backend zu löschen, stellen Sie sicher, dass `deletionPolicy` ist auf Löschen eingestellt. Nur das löschen `TridentBackendConfig` , sicherstellen, dass `deletionPolicy` ist auf Beibehaltung eingestellt. Dadurch wird sichergestellt, dass das Backend weiterhin vorhanden ist und mithilfe von verwaltet werden kann. `tridentctl` .

Führen Sie den folgenden Befehl aus:

```
kubectl delete tbc <tbc-name> -n trident
```

Trident löscht die Kubernetes-Secrets nicht, die von diesem Unternehmen verwendet wurden. `TridentBackendConfig` . Der Kubernetes-Benutzer ist für das Bereinigen von Geheimnissen verantwortlich. Beim Löschen von Geheimnissen ist Vorsicht geboten. Geheimnisse sollten nur dann gelöscht werden, wenn sie von den Backends nicht verwendet werden.

Die vorhandenen Backends ansehen

Führen Sie den folgenden Befehl aus:

```
kubectl get tbc -n trident
```

Sie können auch ausführen `tridentctl get backend -n trident` oder `tridentctl get backend -o yaml -n trident` um eine Liste aller existierenden Backends zu erhalten. Diese Liste wird auch Backends enthalten, die mit `tridentctl` .

Aktualisieren Sie ein Backend

Es kann mehrere Gründe für ein Backend-Update geben:

- Die Zugangsdaten zum Speichersystem haben sich geändert. Um die Anmeldeinformationen zu aktualisieren, muss das Kubernetes-Secret, das im `TridentBackendConfig` Das Objekt muss aktualisiert werden. Trident aktualisiert das Backend automatisch mit den zuletzt angegebenen Zugangsdaten. Führen Sie folgenden Befehl aus, um das Kubernetes-Secret zu aktualisieren:

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- Parameter (wie zum Beispiel der Name der verwendeten ONTAP SVM) müssen aktualisiert werden.
 - Sie können aktualisieren `TridentBackendConfig` Objekte direkt über Kubernetes mit folgendem Befehl:

```
kubectl apply -f <updated-backend-file.yaml>
```

- Alternativ können Sie Änderungen an der bestehenden Datei vornehmen. `TridentBackendConfig` CR mit folgendem Befehl:

```
kubectl edit tbc <tbc-name> -n trident
```



- Wenn ein Backend-Update fehlschlägt, bleibt das Backend in seiner zuletzt bekannten Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie Folgendes ausführen: `kubectl get tbc <tbc-name> -o yaml -n trident` oder `kubectl describe tbc <tbc-name> -n trident`.
- Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Aktualisierungsbefehl erneut ausführen.

Führen Sie die Backend-Verwaltung mit `tridentctl` durch.

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge durchführen, indem Sie `tridentctl`.

Erstelle ein Backend

Nachdem Sie ein "[Backend-Konfigurationsdatei](#)" Führen Sie folgenden Befehl aus:

```
tridentctl create backend -f <backend-file> -n trident
```

Wenn die Backend-Erstellung fehlschlägt, gab es ein Problem mit der Backend-Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie es einfach ausführen. `create` Den Befehl erneut geben.

Backend löschen

Um ein Backend aus Trident zu löschen, gehen Sie wie folgt vor:

1. Den Backend-Namen abrufen:

```
tridentctl get backend -n trident
```

2. Backend löschen:

```
tridentctl delete backend <backend-name> -n trident
```



Falls Trident über dieses Backend Volumes und Snapshots bereitgestellt hat, die noch existieren, verhindert das Löschen des Backends, dass neue Volumes über dieses Backend bereitgestellt werden. Das Backend wird weiterhin im Status „Löschen“ verbleiben.

Die vorhandenen Backends ansehen

Um die von Trident bekannten Backends anzuzeigen, gehen Sie wie folgt vor:

- Um eine Zusammenfassung zu erhalten, führen Sie folgenden Befehl aus:

```
tridentctl get backend -n trident
```

- Um alle Details zu erhalten, führen Sie folgenden Befehl aus:

```
tridentctl get backend -o json -n trident
```

Aktualisieren Sie ein Backend

Nachdem Sie eine neue Backend-Konfigurationsdatei erstellt haben, führen Sie folgenden Befehl aus:

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Wenn das Backend-Update fehlschlägt, lag entweder ein Fehler in der Backend-Konfiguration vor oder Sie haben versucht, ein ungültiges Update durchzuführen. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie es einfach ausführen. `update` Den Befehl erneut geben.

Identifizieren Sie die Speicherklassen, die ein Backend verwenden.

Dies ist ein Beispiel für die Art von Fragen, die Sie mit JSON beantworten können. `tridentctl` Ausgaben für Backend-Objekte. Dies verwendet die `jq` Dienstprogramm, das Sie installieren müssen.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

Dies gilt auch für Backends, die durch die Verwendung von [fehlende Information] erstellt wurden.

`TridentBackendConfig`.

Wechseln Sie zwischen verschiedenen Backend-Verwaltungsoptionen.

Erfahren Sie mehr über die verschiedenen Möglichkeiten zur Backend-Verwaltung in Trident.

Optionen zur Verwaltung von Backends

Mit der Einführung von `TridentBackendConfig` Administratoren haben nun zwei unterschiedliche Möglichkeiten, Backends zu verwalten. Dies wirft folgende Fragen auf:

- Können Backends erstellt werden mit `tridentctl` verwaltet werden mit `TridentBackendConfig` ?
- Können Backends erstellt werden mit `TridentBackendConfig` verwaltet werden mit `tridentctl` ?

Verwalten `tridentctl` Backends verwenden `TridentBackendConfig`

Dieser Abschnitt beschreibt die erforderlichen Schritte zur Verwaltung von Backends, die mit folgendem Werkzeug erstellt wurden: `tridentctl` direkt über die Kubernetes-Schnittstelle durch Erstellen `TridentBackendConfig` Objekte.

Dies gilt für folgende Szenarien:

- Vorhandene Backends, die keine haben `TridentBackendConfig` weil sie mit `tridentctl`.
- Neue Backends, die mit `tridentctl` während andere `TridentBackendConfig` Objekte existieren.

In beiden Szenarien bleiben die Backend-Systeme weiterhin vorhanden, über die Trident die Datenmengen plant und verarbeitet. Administratoren haben hier zwei Möglichkeiten:

- Weiter verwenden `tridentctl` zur Verwaltung von Backends, die damit erstellt wurden.
- Bind-Backends, die mit `tridentctl` zu einem neuen `TridentBackendConfig` Objekt. Dies würde bedeuten, dass die Backends wie folgt verwaltet werden: `kubectl` und nicht `tridentctl`.

Um ein bereits bestehendes Backend zu verwalten `kubectl` Sie müssen einen erstellen `TridentBackendConfig` das an das bestehende Backend angebunden wird. Hier ist eine Übersicht, wie das funktioniert:

1. Erstelle ein Kubernetes-Secret. Das Geheimnis enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster/-dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieser Eintrag enthält spezifische Informationen zum Speichercluster/Speicherdienst und verweist auf das im vorherigen Schritt erstellte Geheimnis. Es ist darauf zu achten, dass identische Konfigurationsparameter angegeben werden (wie z. B. `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, und so weiter). `spec.backendName` muss auf den Namen des bestehenden Backends eingestellt werden.

Schritt 0: Backend identifizieren

Um ein `TridentBackendConfig` Um eine Verbindung zu einem bestehenden Backend herzustellen, müssen Sie die Backend-Konfiguration abrufen. In diesem Beispiel gehen wir davon aus, dass ein Backend mit der folgenden JSON-Definition erstellt wurde:

```
tridentctl get backend ontap-nas-backend -n trident
```

```
+-----+-----+
+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+
| ontap-nas-backend      | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+-----+-----+
+-----+-----+
```

```
cat ontap-nas-backend.json
```

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}

```


Schritt 1: Erstellen Sie ein Kubernetes-Secret.

Erstellen Sie ein Geheimnis, das die Anmeldeinformationen für das Backend enthält, wie in diesem Beispiel gezeigt:

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

Schritt 2: Erstellen Sie ein TridentBackendConfig CR

Der nächste Schritt besteht darin, ein/e zu erstellen `TridentBackendConfig` CR, das automatisch an das bereits vorhandene gebunden wird `ontap-nas-backend` (wie in diesem Beispiel). Stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Derselbe Backend-Name ist definiert in `spec.backendName`.
- Die Konfigurationsparameter sind identisch mit denen des ursprünglichen Backends.
- Virtuelle Pools (sofern vorhanden) müssen die gleiche Reihenfolge wie im ursprünglichen Backend beibehalten.
- Die Zugangsdaten werden über ein Kubernetes-Secret und nicht im Klartext bereitgestellt.

In diesem Fall `TridentBackendConfig` wird folgendermaßen aussehen:

```
cat backend-tbc-ontap-nas.yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
    region: us_east_1
  storage:
  - labels:
      app: msoffice
      cost: '100'
      zone: us_east_1a
      defaults:
        spaceReserve: volume
        encryption: 'true'
        unixPermissions: '0755'
  - labels:
      app: mysqlldb
      cost: '25'
      zone: us_east_1d
      defaults:
        spaceReserve: volume
        encryption: 'false'
        unixPermissions: '0775'

```

```

kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created

```

Schritt 3: Überprüfen Sie den Status des TridentBackendConfig CR

Nach dem TridentBackendConfig wurde erstellt, seine Phase muss sein Bound . Es sollte außerdem denselben Backend-Namen und dieselbe UUID wie das bestehende Backend aufweisen.

```
kubectl get tbc tbc-ontap-nas-backend -n trident
```

NAME	BACKEND NAME	BACKEND UUID
tbc-ontap-nas-backend	ontap-nas-backend	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7

```

PHASE    STATUS
Bound    Success

#confirm that no new backends were created (i.e., TridentBackendConfig did
not end up creating a new backend)
tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-nas-backend      | ontap-nas      | 52f2eb10-e4c6-4160-99fc-96b3be5ab5d7 |
| online |      25 |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Das Backend wird nun vollständig über die Software verwaltet. tbc-ontap-nas-backend TridentBackendConfig Objekt.

Verwalten TridentBackendConfig **Backends verwenden** tridentctl

`tridentctl` kann verwendet werden, um Backends aufzulisten, die mit folgendem Werkzeug erstellt wurden: `TridentBackendConfig`. Darüber hinaus können Administratoren diese Backends auch vollständig selbst verwalten durch `tridentctl` durch Löschen `TridentBackendConfig` und sicherstellen `spec.deletionPolicy` ist eingestellt auf `retain`.

Schritt 0: Backend identifizieren

Nehmen wir beispielsweise an, das folgende Backend wurde erstellt mit TridentBackendConfig:

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                                BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san    delete

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-----+-----+
+-----+-----+-----+-----+
```

Aus der Ausgabe geht hervor, dass `TridentBackendConfig` wurde erfolgreich erstellt und ist an ein Backend gebunden [siehe die UUID des Backends].

Schritt 1: Bestätigen `deletionPolicy` ist eingestellt auf `retain`

Werfen wir einen Blick auf den Wert von `deletionPolicy`. Dies muss auf `retain` eingestellt werden. Dies stellt sicher, dass, wenn ein `TridentBackendConfig` gelöscht wird, bleibt die Backend-Definition weiterhin vorhanden und kann verwaltet werden mit `tridentctl`.

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                                BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san    delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                                BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san    retain
```



Fahren Sie nicht mit dem nächsten Schritt fort, es sei denn `deletionPolicy` ist eingestellt auf `retain`.

Schritt 2: Löschen Sie die `TridentBackendConfig` CR

Der letzte Schritt besteht darin, die `TridentBackendConfig` CR. Nach Bestätigung der `deletionPolicy` ist eingestellt auf `retain` Sie können die Löschung nun durchführen:

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                      UUID                      |
| STATE  | VOLUMES |                      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-0a5315ac5f82 |
| online |      33 |                      |
+-----+-----+-----+-----+
```

Nach der Löschung der `TridentBackendConfig` Trident entfernt das Objekt einfach, ohne das Backend selbst zu löschen.

Speicherklassen erstellen und verwalten

Erstellen einer Speicherklasse

Konfigurieren Sie ein Kubernetes `StorageClass`-Objekt und erstellen Sie die Storage-Klasse, um Trident anzuweisen, wie Volumes bereitgestellt werden sollen.

Konfigurieren eines Kubernetes `StorageClass`-Objekts

Der "[Kubernetes StorageClass-Objekt](#)" Identifiziert Trident als den für diese Klasse verwendeten Provisionierer und weist Trident an, wie ein Volume zu provisionieren ist. Beispiel:

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
mountOptions:
  - nfsvers=3
  - nolock
parameters:
  backendType: "ontap-nas"
  media: "ssd"
allowVolumeExpansion: true
volumeBindingMode: Immediate

```

Siehe "[Kubernetes- und Trident Objekte](#)" Einzelheiten darüber, wie Speicherklassen mit dem interagieren, finden Sie hier. PersistentVolumeClaim und Parameter zur Steuerung der Volumenbereitstellung Trident .

Erstellen einer Speicherklasse

Nachdem Sie das StorageClass-Objekt erstellt haben, können Sie die Speicherklasse erstellen. [Lagerklassenmuster](#) bietet einige grundlegende Beispiele, die Sie verwenden oder modifizieren können.

Schritte

1. Dies ist ein Kubernetes-Objekt, also verwenden Sie `kubectl` um es in Kubernetes zu erstellen.

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

2. Sie sollten nun sowohl in Kubernetes als auch in Trident eine **basic-csi**-Speicherklasse sehen, und Trident sollte die Pools im Backend erkannt haben.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

```
./tridentctl -n trident get storageclass basic-csi -o json
```

```

{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic-csi",
        "attributes": {
          "backendType": "ontap-nas"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "ontapnas_10.0.0.1": [
          "aggr1",
          "aggr2",
          "aggr3",
          "aggr4"
        ]
      }
    }
  ]
}

```

Lagerklassenmuster

Trident bietet "einfache Speicherklassendefinitionen für spezifische Backends" .

Alternativ können Sie bearbeiten `sample-input/storage-class-csi.yaml.tmpl` Die mit dem Installationsprogramm gelieferte Datei ersetzen `BACKEND_TYPE` mit dem Namen des Speichertreibers.

```
./tridentctl -n trident get backend
+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| nas-backend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |         0 |
+-----+-----+-----+
+-----+-----+

cp sample-input/storage-class-csi.yaml.templ sample-input/storage-class-
basic-csi.yaml

# Modify __BACKEND_TYPE__ with the storage driver field above (e.g.,
ontap-nas)
vi sample-input/storage-class-basic-csi.yaml
```

Speicherklassen verwalten

Sie können vorhandene Speicherklassen anzeigen, eine Standard-Speicherklasse festlegen, das Backend der Speicherklasse identifizieren und Speicherklassen löschen.

Die vorhandenen Speicherklassen anzeigen

- Um vorhandene Kubernetes-Speicherklassen anzuzeigen, führen Sie folgenden Befehl aus:

```
kubectl get storageclass
```

- Um Details zur Kubernetes-Speicherklasse anzuzeigen, führen Sie folgenden Befehl aus:

```
kubectl get storageclass <storage-class> -o json
```

- Um die synchronisierten Speicherklassen von Trident anzuzeigen, führen Sie folgenden Befehl aus:

```
tridentctl get storageclass
```

- Um die Details der synchronisierten Speicherklasse von Trident anzuzeigen, führen Sie folgenden Befehl aus:

```
tridentctl get storageclass <storage-class> -o json
```


Legen Sie eine Standardspeicherklasse fest

Mit Kubernetes 1.6 wurde die Möglichkeit eingeführt, eine Standard-Speicherklasse festzulegen. Dies ist die Speicherklasse, die zur Bereitstellung eines Persistent Volume verwendet wird, wenn ein Benutzer in einem Persistent Volume Claim (PVC) kein Persistent Volume angibt.

- Definieren Sie eine Standard-Speicherklasse durch Festlegen der Annotation.
storageclass.kubernetes.io/is-default-class in der Definition der Speicherklasse auf „true“ setzen. Gemäß der Spezifikation wird jeder andere Wert oder das Fehlen der Annotation als falsch interpretiert.
- Sie können eine vorhandene Speicherklasse als Standardspeicherklasse konfigurieren, indem Sie den folgenden Befehl verwenden:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- Ebenso können Sie die Standard-Speicherklassenannotation mit folgendem Befehl entfernen:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Im Trident -Installationspaket sind auch Beispiele enthalten, die diese Annotation beinhalten.



Es sollte immer nur eine Standard-Speicherklasse in Ihrem Cluster vorhanden sein. Kubernetes verhindert technisch gesehen nicht, dass Sie mehr als eine Speicherklasse haben, verhält sich aber so, als gäbe es überhaupt keine Standard-Speicherklasse.

Identifizieren Sie das Backend für eine Speicherklasse

Dies ist ein Beispiel für die Art von Fragen, die Sie mit JSON beantworten können. tridentctl Ausgaben für Trident Backend-Objekte. Dies verwendet die jq Dienstprogramm, das Sie möglicherweise zuerst installieren müssen.

```
tridentctl get storageclass -o json | jq ' [.items[] | {storageClass:  
.Config.name, backends: [.storage]|unique}] '
```

Eine Speicherklasse löschen

Um eine Speicherklasse aus Kubernetes zu löschen, führen Sie folgenden Befehl aus:

```
kubectl delete storageclass <storage-class>
```

`<storage-class>` sollte durch Ihre Speicherklasse ersetzt werden.

Alle persistenten Volumes, die über diese Speicherklasse erstellt wurden, bleiben unberührt und werden weiterhin von Trident verwaltet.



Trident erzwingt eine leere `fsType` für die dadurch entstehenden Mengen. Für iSCSI-Backends wird empfohlen, Folgendes zu erzwingen `parameters.fsType` in der `StorageClass`. Sie sollten die vorhandenen `StorageClasses` löschen und neu erstellen mit `parameters.fsType` angegeben.

Bereitstellung und Verwaltung von Volumina

Bereitstellung eines Volumens

Erstellen Sie einen `PersistentVolumeClaim` (PVC), der die konfigurierte Kubernetes `StorageClass` verwendet, um Zugriff auf das PV anzufordern. Anschließend können Sie die PV-Anlage an einem Pod montieren.

Überblick

A "[*PersistentVolumeClaim*](#)" (PVC) ist eine Anfrage für den Zugriff auf das `PersistentVolume` im Cluster.

Das PVC kann so konfiguriert werden, dass es Speicherplatz einer bestimmten Größe oder einen bestimmten Zugriffsmodus anfordert. Mithilfe der zugehörigen `StorageClass` kann der Clusteradministrator mehr als nur die Größe und den Zugriffsmodus des `PersistentVolumes` steuern – beispielsweise die Leistung oder das Servicelevel.

Nachdem Sie das PVC erstellt haben, können Sie das Volumen in einem Gehäuse montieren.

PVC erstellen

Schritte

1. Erstellen Sie die PVC.

```
kubectl create -f pvc.yaml
```

2. Überprüfen Sie den PVC-Status.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	1Gi	RWO		5m

1. Montieren Sie das Volume in einem Gehäuse.

```
kubectl create -f pv-pod.yaml
```



Sie können den Fortschritt überwachen mit `kubectl get pod --watch`.

2. Überprüfen Sie, ob das Volume eingebunden ist. `/my/mount/path`.

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

3. Sie können den Pod jetzt löschen. Die Pod-Anwendung wird nicht mehr existieren, das Volume bleibt jedoch erhalten.

```
kubectl delete pod pv-pod
```

Beispielmanifeste

Beispielmanifeste für PersistentVolumeClaim

Diese Beispiele zeigen grundlegende PVC-Konfigurationsoptionen.

PVC mit RWO-Zugang

Dieses Beispiel zeigt ein einfaches PVC mit RWO-Zugriff, das einer StorageClass namens zugeordnet ist. `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

PVC mit NVMe/TCP

Dieses Beispiel zeigt eine einfache PVC für NVMe/TCP mit RWO-Zugriff, die einer StorageClass namens zugeordnet ist. `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

Pod-Manifestproben

Diese Beispiele zeigen grundlegende Konfigurationen zur Befestigung des PVC an einer Kapsel.

Basiskonfiguration

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: storage
      persistentVolumeClaim:
        claimName: pvc-storage
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: storage
```

Grundlegende NVMe/TCP-Konfiguration

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-nginx
spec:
  volumes:
    - name: basic-pvc
      persistentVolumeClaim:
        claimName: pvc-san-nvme
  containers:
    - name: task-pv-container
      image: nginx
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: basic-pvc
```

Siehe "[Kubernetes- und Trident Objekte](#)" Einzelheiten darüber, wie Speicherklassen mit dem interagieren, finden Sie hier. PersistentVolumeClaim und Parameter zur Steuerung der Volumenbereitstellung Trident .

Volumen erweitern

Trident bietet Kubernetes-Benutzern die Möglichkeit, ihre Volumes nach der Erstellung zu erweitern. Hier finden Sie Informationen zu den Konfigurationen, die zum Erweitern von iSCSI-, NFS-, SMB-, NVMe/TCP- und FC-Volumes erforderlich sind.

Erweitern eines iSCSI-Volumes

Sie können ein iSCSI Persistent Volume (PV) mithilfe des CSI-Provisioners erweitern.



Die iSCSI-Volume-Erweiterung wird unterstützt von `ontap-san`, `ontap-san-economy`, `solidfire-san` Treiber und erfordert Kubernetes 1.16 oder höher.

Schritt 1: Konfigurieren Sie die StorageClass zur Unterstützung der Volume-Erweiterung.

Bearbeiten Sie die StorageClass-Definition, um die `allowVolumeExpansion` Feld zu `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Bearbeiten Sie eine bereits vorhandene StorageClass, um Folgendes einzuschließen:
`allowVolumeExpansion` Parameter.

Schritt 2: Erstellen Sie ein PVC mit der von Ihnen erstellten StorageClass.

Bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage` um die neu gewünschte Größe widerzuspiegeln, die größer als die ursprüngliche Größe sein muss.

```
cat pvc-ontapsan.yaml
```

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san

```

Trident erstellt ein Persistent Volume (PV) und verknüpft es mit diesem Persistent Volume Claim (PVC).

```

kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi
RWO          ontap-san    8s

kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY   STATUS   CLAIM                                     STORAGECLASS  REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi      RWO          Delete           Bound    default/san-pvc                         ontap-san    10s

```

Schritt 3: Definieren Sie eine Halterung, an der das PVC befestigt wird.

Befestigen Sie das PV an einer Kapsel, damit es in seiner Größe angepasst werden kann. Beim Ändern der Größe eines iSCSI PV gibt es zwei Szenarien:

- Wenn das PV an einen Pod angehängt ist, erweitert Trident das Volume im Speicher-Backend, scannt das Gerät erneut und passt die Größe des Dateisystems an.
- Beim Versuch, die Größe eines nicht zugeordneten PV zu ändern, erweitert Trident das Volumen im Speicher-Backend. Nachdem die PVC an einen Pod gebunden wurde, scannt Trident das Gerät erneut und passt die Größe des Dateisystems an. Kubernetes aktualisiert anschließend die PVC-Größe, nachdem der Erweiterungsvorgang erfolgreich abgeschlossen wurde.

In diesem Beispiel wird ein Pod erstellt, der Folgendes verwendet: `san-pvc`.

```
kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
ubuntu-pod    1/1     Running   0           65s

kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
               csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Schritt 4: PV erweitern

Um die Größe des erstellten PV von 1 GiB auf 2 GiB zu ändern, bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage` bis 2 GiB.

```
kubectl edit pvc san-pvc
```



```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...

```

Schritt 5: Die Erweiterung validieren

Sie können überprüfen, ob die Erweiterung korrekt funktioniert hat, indem Sie die Größe des PVC, PV und des Trident -Volumens kontrollieren:

```
kubectl get pvc san-pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi
RWO           ontap-san    11m

kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi        RWO
Delete              Bound      default/san-pvc  ontap-san    12m

tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san    |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

FC-Volume erweitern

Sie können ein FC Persistent Volume (PV) mithilfe des CSI-Provisioners erweitern.



Die FC-Volumenerweiterung wird unterstützt durch ontap-san Treiber und erfordert Kubernetes 1.16 oder höher.

Schritt 1: Konfigurieren Sie die StorageClass zur Unterstützung der Volume-Erweiterung.

Bearbeiten Sie die StorageClass-Definition, um die `allowVolumeExpansion` Feld zu `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Bearbeiten Sie eine bereits vorhandene StorageClass, um Folgendes einzuschließen:
allowVolumeExpansion Parameter.

Schritt 2: Erstellen Sie ein PVC mit der von Ihnen erstellten StorageClass.

Bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage` um die neu gewünschte Größe widerzuspiegeln, die größer als die ursprüngliche Größe sein muss.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Trident erstellt ein Persistent Volume (PV) und verknüpft es mit diesem Persistent Volume Claim (PVC).

```
kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi
RWO           ontap-san    8s

kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY   STATUS    CLAIM                                STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi      RWO           Delete           Bound     default/san-pvc  ontap-san                10s
```

Schritt 3: Definieren Sie eine Halterung, an der das PVC befestigt wird.

Befestigen Sie das PV an einer Kapsel, damit es in seiner Größe angepasst werden kann. Bei der Größenänderung einer FC-PV gibt es zwei Szenarien:

- Wenn das PV an einen Pod angehängt ist, erweitert Trident das Volume im Speicher-Backend, scannt das Gerät erneut und passt die Größe des Dateisystems an.
- Beim Versuch, die Größe eines nicht zugeordneten PV zu ändern, erweitert Trident das Volumen im Speicher-Backend. Nachdem die PVC an einen Pod gebunden wurde, scannt Trident das Gerät erneut

und passt die Größe des Dateisystems an. Kubernetes aktualisiert anschließend die PVC-Größe, nachdem der Erweiterungsvorgang erfolgreich abgeschlossen wurde.

In diesem Beispiel wird ein Pod erstellt, der Folgendes verwendet: `san-pvc` .

```
kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
ubuntu-pod    1/1     Running   0           65s

kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
               csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Schritt 4: PV erweitern

Um die Größe des erstellten PV von 1 GiB auf 2 GiB zu ändern, bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage` bis 2 GiB.

```
kubectl edit pvc san-pvc
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...

```

Schritt 5: Die Erweiterung validieren

Sie können überprüfen, ob die Erweiterung korrekt funktioniert hat, indem Sie die Größe des PVC, PV und des Trident -Volumens kontrollieren:

```
kubectl get pvc san-pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi
RWO           ontap-san    11m

kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi        RWO
Delete              Bound      default/san-pvc  ontap-san    12m

tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san    |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
```

Erweitern eines NFS-Volumes

Trident unterstützt die Volumenerweiterung für NFS-PVs, die auf `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `gcp-cvs`, Und `azure-netapp-files` Backends.

Schritt 1: Konfigurieren Sie die StorageClass zur Unterstützung der Volume-Erweiterung.

Um ein NFS-PV zu vergrößern oder zu verkleinern, muss der Administrator zunächst die Speicherklasse so konfigurieren, dass eine Volumenerweiterung möglich ist. `allowVolumeExpansion` Feld zu `true` :

```
cat storageclass-ontapnas.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontapnas
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
allowVolumeExpansion: true
```

Wenn Sie bereits eine Speicherklasse ohne diese Option erstellt haben, können Sie die vorhandene

Speicherklasse einfach bearbeiten, indem Sie `kubectl edit storageclass` um eine Volumenausdehnung zu ermöglichen.

Schritt 2: Erstellen Sie ein PVC mit der von Ihnen erstellten StorageClass.

```
cat pvc-ontapnas.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Trident sollte für dieses PVC ein 20 MiB NFS PV erstellen:

```
kubectl get pvc
NAME                STATUS    VOLUME
CAPACITY            ACCESS MODES  STORAGECLASS  AGE
ontapnas20mb        Bound       pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi
RWO                  ontapnas      9s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                CAPACITY  ACCESS MODES
RECLAIM POLICY      STATUS    CLAIM                STORAGECLASS  REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi      RWO
Delete              Bound     default/ontapnas20mb  ontapnas
2m42s
```

Schritt 3: PV erweitern

Um die neu erstellte 20 MiB PV auf 1 GiB zu skalieren, bearbeiten Sie die PVC und stellen Sie sie entsprechend ein. `spec.resources.requests.storage` bis 1 GiB:

```
kubectl edit pvc ontapnas20mb
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: 2018-08-21T18:26:44Z
  finalizers:
  - kubernetes.io/pvc-protection
  name: ontapnas20mb
  namespace: default
  resourceVersion: "1958015"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/ontapnas20mb
  uid: c1bd7fa5-a56f-11e8-b8d7-fa163e59eaab
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
# ...

```

Schritt 4: Die Erweiterung validieren

Sie können überprüfen, ob die Größenänderung korrekt durchgeführt wurde, indem Sie die Größe des PVC, PV und des Trident -Volumens überprüfen:


```
kubectl get pvc ontapnas20mb
NAME          STATUS    VOLUME
CAPACITY     ACCESS MODES  STORAGECLASS  AGE
ontapnas20mb  Bound      pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  1Gi
RWO          ontapnas      4m44s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  1Gi      RWO
Delete          Bound      default/ontapnas20mb  ontapnas
5m35s

tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n trident
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE  | STORAGE CLASS |
+-----+-----+-----+-----+
| PROTOCOL | BACKEND UUID | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
| file      | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Importmengen

Sie können vorhandene Speichervolumes als Kubernetes PV importieren, indem Sie `tridentctl import`.

Überblick und Überlegungen

Sie könnten beispielsweise ein Volume in Trident importieren, um:

- Eine Anwendung containerisieren und ihren vorhandenen Datensatz wiederverwenden
- Verwenden Sie eine Kopie eines Datensatzes für eine kurzlebige Anwendung.
- Einen ausgefallenen Kubernetes-Cluster wiederherstellen
- Anwendungsdaten während der Notfallwiederherstellung migrieren

Überlegungen

Bevor Sie ein Volume importieren, beachten Sie bitte die folgenden Hinweise.

- Trident kann nur ONTAP -Volumes vom Typ RW (Read-Write) importieren. DP-Volumes (Datenschutz-Volumes) sind SnapMirror Zielvolumes. Sie sollten die Spiegelungsbeziehung aufheben, bevor Sie das Volume in Trident importieren.

- Wir empfehlen, Volumes ohne aktive Verbindungen zu importieren. Um ein aktiv genutztes Volume zu importieren, klonen Sie das Volume und führen Sie anschließend den Import durch.



Dies ist besonders wichtig für Block-Volumes, da Kubernetes die vorherige Verbindung nicht erkennen würde und leicht ein aktives Volume an einen Pod anhängen könnte. Dies kann zu Datenbeschädigung führen.

- Obwohl `StorageClass` muss auf einem PVC angegeben werden, Trident verwendet diesen Parameter beim Import nicht. Speicherklassen werden bei der Erstellung von Volumes verwendet, um anhand der Speichereigenschaften aus den verfügbaren Pools auszuwählen. Da das Volume bereits existiert, ist während des Imports keine Poolauswahl erforderlich. Daher schlägt der Import auch dann nicht fehl, wenn sich das Volume auf einem Backend oder Pool befindet, der nicht der im PVC angegebenen Speicherklasse entspricht.
- Das vorhandene Volumen wird ermittelt und im PVC festgelegt. Nachdem das Volume vom Speichertreiber importiert wurde, wird das PV mit einem `ClaimRef` zum PVC erstellt.
 - Die Rückforderungsrichtlinie ist anfänglich so eingestellt, dass `retain` im PV. Nachdem Kubernetes PVC und PV erfolgreich gebunden hat, wird die Rückgewinnungsrichtlinie aktualisiert, um der Rückgewinnungsrichtlinie der Speicherklasse zu entsprechen.
 - Wenn die Rückgewinnungsrichtlinie der Speicherklasse lautet `delete` Das Speichervolumen wird gelöscht, wenn das PV gelöscht wird.
- Standardmäßig verwaltet Trident die PVC und benennt das FlexVol volume und die LUN im Backend um. Du kannst die `--no-manage` Flag zum Importieren eines nicht verwalteten Volumes. Wenn Sie `--no-manage` Trident führt während des gesamten Lebenszyklus der Objekte keine zusätzlichen Bearbeitungsschritte an den PVC- oder PV-Materialien durch. Das Speichervolumen wird beim Löschen des PV nicht gelöscht, und auch andere Operationen wie das Klonen von Volumes und das Ändern der Volume-Größe werden ignoriert.



Diese Option ist nützlich, wenn Sie Kubernetes für containerisierte Workloads verwenden möchten, ansonsten aber den Lebenszyklus des Speichervolumens außerhalb von Kubernetes verwalten möchten.

- Dem PVC und PV wird eine Anmerkung hinzugefügt, die einen doppelten Zweck erfüllt: Sie gibt an, dass das Volumen importiert wurde und ob PVC und PV verwaltet werden. Diese Anmerkung darf nicht verändert oder entfernt werden.

Importieren Sie ein Volume

Sie können verwenden `tridentctl import` ein Volume importieren.

Schritte

1. Erstellen Sie die Persistent Volume Claim (PVC)-Datei (zum Beispiel, `pvc.yaml`) das zur Herstellung des PVC verwendet wird. Die PVC-Datei sollte Folgendes enthalten: `name` , `namespace` , `accessModes` , Und `storageClassName` . Optional können Sie Folgendes angeben `unixPermissions` in Ihrer PVC-Definition.

Nachfolgend ein Beispiel für eine Mindestanforderung:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class
```



Fügen Sie keine zusätzlichen Parameter wie PV-Name oder Volumengröße hinzu. Dies kann dazu führen, dass der Importbefehl fehlschlägt.

2. Verwenden Sie die `tridentctl import volume` Befehl zur Angabe des Namens des Trident -Backends, das das Volume enthält, und des Namens, der das Volume auf dem Speicher eindeutig identifiziert (z. B. ONTAP FlexVol, Element Volume, Cloud Volumes Service -Pfad). Der `-f` Ein Argument ist erforderlich, um den Pfad zur PVC-Datei anzugeben.

```
tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-
file>
```

Beispiele

Prüfen Sie die folgenden Beispiele für den Volumenimport auf unterstützte Treiber.

ONTAP NAS und ONTAP NAS FlexGroup

Trident unterstützt den Volumenimport über die `ontap-nas` Und `ontap-nas-flexgroup` Fahrer.



- Trident unterstützt keinen Volumenimport mit dem `ontap-nas-economy` Treiber.
- Der `ontap-nas` Und `ontap-nas-flexgroup` Treiber erlauben keine doppelten Datenträgernamen.

Jeder mit dem `ontap-nas` Der Treiber ist ein FlexVol volume auf dem ONTAP Cluster. Importieren von FlexVol -Volumes mit dem `ontap-nas` Der Treiber funktioniert genauso. Ein FlexVol -Volume, das bereits auf einem ONTAP Cluster vorhanden ist, kann als importiert werden `ontap-nas` PVC. In ähnlicher Weise können FlexGroup Volumes importiert werden als `ontap-nas-flexgroup` PVCs.

ONTAP NAS-Beispiele

Nachfolgend werden Beispiele für den Import eines verwalteten und eines nicht verwalteten Volumes gezeigt.

Verwaltetes Volumen

Das folgende Beispiel importiert ein Volume mit dem Namen `managed_volume` auf einem Backend namens `ontap_nas`:

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>
```

NAME	SIZE	STORAGE CLASS
PROTOCOL	BACKEND	UUID
STATE	MANAGED	
pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7	1.0 GiB	standard
file	c5a6f6a4-b052-423b-80d4-8fb491a14a22	online
		true

Unverwaltetes Volumen

Bei der Verwendung des `--no-manage` Argument: Trident benennt das Volume nicht um.

Das folgende Beispiel importiert `unmanaged_volume` auf der `ontap_nas` Backend:

```
tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-file> --no-manage
```

NAME	SIZE	STORAGE CLASS
PROTOCOL	BACKEND	UUID
STATE	MANAGED	
pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7	1.0 GiB	standard
file	c5a6f6a4-b052-423b-80d4-8fb491a14a22	online
		false

ONTAP SAN

Trident unterstützt den Volumenimport mithilfe der `ontap-san` (iSCSI, NVMe/TCP und FC) und `ontap-san-economy` Treiber.

Trident kann ONTAP SAN FlexVol -Volumes importieren, die eine einzelne LUN enthalten. Dies steht im Einklang mit der `ontap-san` Treiber, der für jedes PVC ein FlexVol volume und eine LUN innerhalb des FlexVol volume erstellt. Trident importiert das FlexVol volume und verknüpft es mit der PVC-Definition. Trident

kann importieren `ontap-san-economy` Volumes, die mehrere LUNs enthalten.

ONTAP SAN-Beispiele

Nachfolgend werden Beispiele für den Import eines verwalteten und eines nicht verwalteten Volumes gezeigt.

Verwaltetes Volumen

Bei verwalteten Volumes benennt Trident das FlexVol volume um in `pvc-<uuid>` Format und die LUN innerhalb des FlexVol volume zu `lun0`.

Das folgende Beispiel importiert die `ontap-san-managed` FlexVol volume, die auf dem `ontap_san_default` Backend:

```
tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-  
basic-import.yaml -n trident -d
```

NAME	SIZE	STORAGE CLASS
PROTOCOL BACKEND UUID	STATE	MANAGED
pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a	20 MiB	basic
block cd394786-ddd5-4470-adc3-10c5ce4ca757	online	true

Unverwaltetes Volumen

Das folgende Beispiel importiert `unmanaged_example_volume` auf der `ontap_san` Backend:

```
tridentctl import volume -n trident san_blog unmanaged_example_volume  
-f pvc-import.yaml --no-manage
```

NAME	SIZE	STORAGE CLASS
PROTOCOL BACKEND UUID	STATE	MANAGED
pvc-1fc999c9-ce8c-459c-82e4-ed4380a4b228	1.0 GiB	san-blog
block e3275890-7d80-4af6-90cc-c7a0759f555a	online	false

Wenn Sie LUNs Igroups zugeordnet haben, die einen IQN mit einem Kubernetes-Knoten-IQN teilen, wie im folgenden Beispiel gezeigt, erhalten Sie die Fehlermeldung: `LUN already mapped to initiator(s) in`

this group. Sie müssen den Initiator entfernen oder die LUN-Zuordnung aufheben, um das Volume importieren zu können.

Vserver	Igroup	Protocol	OS Type	Initiators
svm0	k8s-nodename.example.com-fe5d36f2-cded-4f38-9eb0-c7719fc2f9f3	iscsi	linux	iqn.1994-05.com.redhat:4c2e1cf35e0
svm0	unmanaged-example-igroup	mixed	linux	iqn.1994-05.com.redhat:4c2e1cf35e0

Element

Trident unterstützt die NetApp Element Software und den NetApp HCI Volume-Import mithilfe von `solidfire-san` Treiber.



Der Element-Treiber unterstützt doppelte Datenträgernamen. Trident gibt jedoch eine Fehlermeldung aus, wenn doppelte Datenträgernamen vorhanden sind. Als Ausweidlösung können Sie das Volume klonen, einen eindeutigen Volume-Namen vergeben und das geklonte Volume importieren.

Elementbeispiel

Das folgende Beispiel importiert ein `element-managed` Lautstärke im Backend `element_default`.

```
tridentctl import volume element_default element-managed -f pvc-basic-import.yaml -n trident -d
```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
+-----+-----+-----+-----+
|          BACKEND UUID  |      | STATE         |
+-----+-----+-----+-----+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
| block      | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true         |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Google Cloud Platform

Trident unterstützt den Volumenimport über die `gcp-cvs` Treiber.



Um ein vom NetApp Cloud Volumes Service unterstütztes Volume in die Google Cloud Platform zu importieren, identifizieren Sie das Volume anhand seines Volume-Pfads. Der Volume-Pfad ist der Teil des Exportpfads des Volumes nach dem `:/`. Wenn beispielsweise der Exportpfad lautet `10.0.0.1:/adroit-jolly-swift` Der Volumenpfad ist `adroit-jolly-swift`.

Beispiel für die Google Cloud Platform

Das folgende Beispiel importiert ein `gcp-cvs` Lautstärke im Backend `gcpcvs_YEppr` mit dem Volumenpfad von `adroit-jolly-swift`.

```
tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-
file> -n trident
```

PROTOCOL	NAME	BACKEND	UUID	SIZE	STATE	STORAGE CLASS	MANAGED
	pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55			93 GiB	online	gcp-storage	file
	e1a6e65b-299e-4568-ad05-4f0a105c888f				true		

Azure NetApp Files

Trident unterstützt den Volumenimport über die `azure-netapp-files` Treiber.



Um ein Azure NetApp Files Volume zu importieren, identifizieren Sie das Volume anhand seines Volume-Pfads. Der Volume-Pfad ist der Teil des Exportpfads des Volumes nach dem `:/`. Wenn beispielsweise der Mount-Pfad lautet `10.0.0.2:/importvoll1` Der Volumenpfad ist `importvoll1`.

Azure NetApp Files -Beispiel

Das folgende Beispiel importiert ein `azure-netapp-files` Lautstärke im Backend `azurenetappfiles_40517` mit dem Volumenpfad `importvoll1`.

```
tridentctl import volume azurenetappfiles_40517 importvoll1 -f <path-to-pvc-file> -n trident
```

```
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
| PROTOCOL |      BACKEND UUID      | STATE | MANAGED |
+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab | 100 GiB | anf-storage |
file      | 1c01274f-d94b-44a3-98a3-04c953c9a51e | online | true      |
+-----+-----+-----+
+-----+-----+-----+-----+
```

Google Cloud NetApp Volumes

Trident unterstützt den Volumenimport über die `google-cloud-netapp-volumes` Treiber.

Google Cloud NetApp Volumes Beispiel

Das folgende Beispiel importiert ein `google-cloud-netapp-volumes` Lautstärke im Backend `backend-tbc-gcnv1` mit dem Volumen `testvoleasiaeast1`.

```
tridentctl import volume backend-tbc-gcnv1 "testvoleasiaeast1" -f < path-to-pvc> -n trident
```

```
+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
| PROTOCOL |      BACKEND UUID      | STATE | MANAGED |
+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 | 10 GiB | gcnv-nfs-sc-
identity | file      | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Das folgende Beispiel importiert ein `google-cloud-netapp-volumes` Volumen, wenn zwei Volumina im selben Bereich vorhanden sind:


```
tridentctl import volume backend-tbc-gcnv1
"projects/123456789100/locations/asia-east1-a/volumes/testvoleasiaeast1"
-f <path-to-pvc> -n trident
```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS
| PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 | 10 GiB | gcnv-nfs-sc-
identity | file      | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
```

Datenträgernamen und -bezeichnungen anpassen

Mit Trident können Sie den von Ihnen erstellten Datenträgern aussagekräftige Namen und Bezeichnungen zuweisen. Dies hilft Ihnen, Volumes zu identifizieren und sie einfach ihren jeweiligen Kubernetes-Ressourcen (PVCs) zuzuordnen. Sie können auch auf Backend-Ebene Vorlagen definieren, um benutzerdefinierte Volumennamen und benutzerdefinierte Bezeichnungen zu erstellen; alle Volumes, die Sie erstellen, importieren oder klonen, werden den Vorlagen entsprechen.

Bevor Sie beginnen

Unterstützung für anpassbare Datenträgernamen und -bezeichnungen:

1. Vorgänge zum Erstellen, Importieren und Klonen von Volumes.
2. Im Falle des `ontap-nas-economy`-Treibers entspricht nur der Name des Qtree-Volumes der Namensvorlage.
3. Im Falle des Treibers `ontap-san-economy` entspricht nur der LUN-Name der Namensvorlage.

Einschränkungen

1. Anpassbare Volumennamen sind nur mit ONTAP -Treibern für lokale Installationen kompatibel.
2. Anpassbare Volumennamen gelten nicht für bereits vorhandene Volumes.

Wichtige Verhaltensweisen von anpassbaren Volumennamen

1. Wenn aufgrund ungültiger Syntax in einer Namensvorlage ein Fehler auftritt, schlägt die Backend-Erstellung fehl. Falls die Anwendung der Vorlage jedoch fehlschlägt, wird das Volume gemäß der

bestehenden Namenskonvention benannt.

2. Das Speicherpräfix ist nicht anwendbar, wenn ein Volume mithilfe einer Namensvorlage aus der Backend-Konfiguration benannt wird. Der gewünschte Präfixwert kann direkt in die Vorlage eingefügt werden.

Backend-Konfigurationsbeispiele mit Namensvorlage und Bezeichnungen

Benutzerdefinierte Namensvorlagen können auf Stamm- und/oder Poolebene definiert werden.

Beispiel auf Wurzelebene

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "defaults": {
    "nameTemplate":
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
  },
  "labels": {
    "cluster": "ClusterA",
    "PVC": "{{.volume.Namespace}}_{{.volume.RequestName}}"
  }
}
```

Beispiel auf Poolebene

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "useREST": true,
  "storage": [
    {
      "labels": {
        "labelname": "label1",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool01_{{ .volume.Name }}_{{ .labels.cluster }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    },
    {
      "labels": {
        "cluster": "label2",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool02_{{ .volume.Name }}_{{ .labels.cluster }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    }
  ]
}
```

Beispiele für Namensvorlagen

Beispiel 1:

```
"nameTemplate": "{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{ .config.BackendName }}"
```

Beispiel 2:

```
"nameTemplate": "pool_{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{ slice .volume.RequestName 1 5 }}"
```

Zu berücksichtigende Punkte

1. Bei Volumenimporten werden die Etiketten nur dann aktualisiert, wenn das vorhandene Volumen Etiketten in einem bestimmten Format aufweist. Zum Beispiel: {"provisioning":{"Cluster":"ClusterA", "PVC": "pvcname"}} .
2. Bei verwalteten Volume-Importen folgt der Volume-Name der Namensvorlage, die auf der obersten Ebene in der Backend-Definition definiert ist.
3. Trident unterstützt die Verwendung eines Slice-Operators mit dem Speicherpräfix nicht.
4. Falls die Vorlagen keine eindeutigen Datenträgernamen ergeben, fügt Trident einige zufällige Zeichen hinzu, um eindeutige Datenträgernamen zu erstellen.
5. Wenn der benutzerdefinierte Name für ein NAS Economy-Volume länger als 64 Zeichen ist, benennt Trident die Volumes gemäß der bestehenden Namenskonvention. Bei allen anderen ONTAP -Treibern schlägt der Volume-Erstellungsprozess fehl, wenn der Volume-Name die Namensbeschränkung überschreitet.

Ein NFS-Volume über Namespaces hinweg freigeben

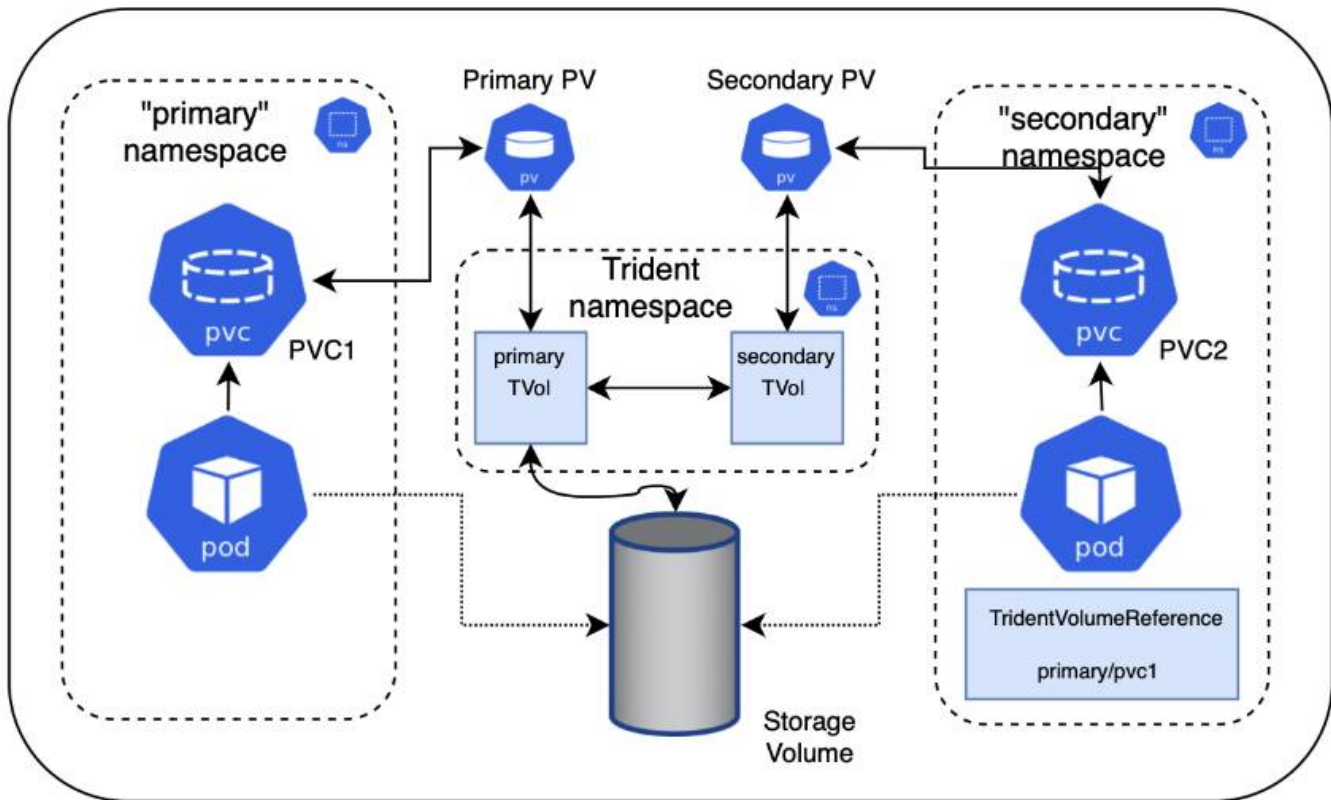
Mit Trident können Sie ein Volume in einem primären Namespace erstellen und es in einem oder mehreren sekundären Namespaces freigeben.

Features

Mit dem TridentVolumeReference CR können Sie ReadWriteMany (RWX) NFS-Volumes sicher über einen oder mehrere Kubernetes-Namespaces hinweg freigeben. Diese Kubernetes-native Lösung bietet folgende Vorteile:

- Mehrere Zugriffskontrollebenen zur Gewährleistung der Sicherheit
- Funktioniert mit allen Trident NFS-Volume-Treibern
- Keine Abhängigkeit von tridentctl oder anderen nicht-nativen Kubernetes-Funktionen.

Dieses Diagramm veranschaulicht die gemeinsame Nutzung von NFS-Volumes über zwei Kubernetes-Namespaces hinweg.



Schnellstart

Die Einrichtung der NFS-Volume-Freigabe ist in wenigen Schritten erledigt.

1

Konfigurieren Sie den Quell-PVC so, dass er das Volume gemeinsam nutzt.

Der Inhaber des Quell-Namespace erteilt die Berechtigung zum Zugriff auf die Daten im Quell-PVC.

2

Erteilen Sie die Berechtigung zum Erstellen eines CR im Ziel-Namespace

Der Clusteradministrator erteilt dem Eigentümer des Ziel-Namespace die Berechtigung, die **TridentVolumeReference** CR zu erstellen.

3

Erstellen Sie eine **TridentVolumeReference im Ziel-Namespace.**

Der Eigentümer des Ziel-Namespace erstellt die **TridentVolumeReference** CR, um auf die Quell-PVC zu verweisen.

4

Erstellen Sie den untergeordneten PVC im Ziel-Namespace

Der Eigentümer des Ziel-Namensraums erstellt den untergeordneten PVC, um die Datenquelle aus dem Quell-PVC zu verwenden.

Konfigurieren Sie die Quell- und Ziel-Namespace

Um die Sicherheit zu gewährleisten, erfordert die gemeinsame Nutzung von Namespaces über verschiedene Namensräume hinweg die Zusammenarbeit und das Handeln des Quell-Namespace-Inhabers, des Cluster-Administrators und des Ziel-Namespace-Inhabers. Die Benutzerrolle wird in jedem Schritt festgelegt.

Schritte

1. **Quellnamespace-Inhaber:** PVC erstellen(`pvc1`) im Quell-Namespace, der die Berechtigung zur gemeinsamen Nutzung mit dem Ziel-Namespace erteilt(`namespace2`) unter Verwendung der `shareToNamespace` Anmerkung.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/shareToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Trident erstellt das PV und sein zugehöriges NFS-Speichervolume.



- Sie können die PVC mithilfe einer durch Kommas getrennten Liste an mehrere Namensräume weitergeben. Zum Beispiel, `trident.netapp.io/shareToNamespace: namespace2, namespace3, namespace4`.
- Sie können die Freigabe für alle Namensräume nutzen `*`. Zum Beispiel, `trident.netapp.io/shareToNamespace: *`
- Sie können das PVC aktualisieren, um Folgendes einzuschließen: `shareToNamespace` Anmerkungen jederzeit möglich.

2. **Clusteradministrator:** Stellen Sie sicher, dass das richtige RBAC vorhanden ist, um dem Besitzer des Zielnamespace die Berechtigung zum Erstellen des `TridentVolumeReference` CR im Zielnamespace zu erteilen.
3. **Inhaber des Ziel-Namespace:** Erstellen Sie im Ziel-Namespace eine `TridentVolumeReference` CR, die auf den Quell-Namespace verweist. `pvc1`.

```

apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1

```

4. **Inhaber des Ziel-Namensraums:** Erstellen Sie eine PVC(pvc2) im Ziel-Namespace(namespace2) unter Verwendung der shareFromPVC Anmerkung zur Kennzeichnung des Quell-PVC.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/shareFromPVC: namespace1/pvc1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi

```



Die Größe des Ziel-PVC-Rohrs muss kleiner oder gleich der Größe des Quell-PVC-Rohrs sein.

Ergebnisse

Trident liest die shareFromPVC Es wird eine Annotation auf dem Ziel-PVC vorgenommen und das Ziel-PV als untergeordnetes Volume ohne eigene Speicherressource erstellt, das auf das Quell-PV verweist und die Speicherressource des Quell-PV mitnutzt. Die Ziel-PVC und PV scheinen normal verbunden zu sein.

Ein freigegebenes Volume löschen

Sie können ein Volume löschen, das von mehreren Namespaces gemeinsam genutzt wird. Trident wird den Zugriff auf das Volume im Quell-Namespace entfernen und den Zugriff für andere Namespaces, die das Volume gemeinsam nutzen, aufrechterhalten. Wenn alle Namespaces, die auf das Volume verweisen, entfernt werden, löscht Trident das Volume.

Verwenden tridentctl get Abfrage untergeordneter Volumes

Verwenden des[tridentctl Mit diesem Dienstprogramm können Sie das ausführen get Befehl zum Abrufen

untergeordneter Volumes. Weitere Informationen finden Sie unter folgendem Link: [../trident-reference/tridentctl.html](#) [tridentctl Befehle und Optionen].

```
Usage:
  tridentctl get [option]
```

Flaggen:

- `--h, --help` Hilfe für Bände.
- `--parentOfSubordinate string`: Beschränken Sie die Abfrage auf das untergeordnete Quellvolume.
- `--subordinateOf string`: Beschränke die Abfrage auf untergeordnete Elemente des Volumens.

Einschränkungen

- Trident kann nicht verhindern, dass Ziel-Namespace auf das gemeinsam genutzte Volume schreiben. Sie sollten Dateisperrungen oder andere Verfahren verwenden, um das Überschreiben von gemeinsam genutzten Datenträgerdaten zu verhindern.
- Der Zugriff auf die Quell-PVC kann nicht durch Entfernen der `shareToNamespace` oder `shareFromNamespace` Anmerkungen oder das Löschen der `TridentVolumeReference` CR. Um den Zugriff zu widerrufen, müssen Sie die untergeordnete PVC löschen.
- Snapshots, Klone und Spiegelungen sind auf untergeordneten Volumes nicht möglich.

Weitere Informationen

Um mehr über den Volume-Zugriff über Namespaces hinweg zu erfahren:

- Besuchen ["Gemeinsame Nutzung von Volumes zwischen Namespaces: Lernen Sie den Namespace-übergreifenden Volume-Zugriff kennen."](#) .
- Sehen Sie sich die Demo an auf ["NetAppTV"](#) .

Volumes über Namespaces hinweg klonen

Mit Trident können Sie neue Volumes erstellen, indem Sie vorhandene Volumes oder Volume-Snapshots aus einem anderen Namespace innerhalb desselben Kubernetes-Clusters verwenden.

Voraussetzungen

Vor dem Klonen von Volumes muss sichergestellt werden, dass Quell- und Ziel-Backend vom gleichen Typ sind und die gleiche Speicherklasse aufweisen.



Das Klonen über Namensräume hinweg wird nur für die `ontap-san` und `ontap-nas` Speichertreiber. Schreibgeschützte Klone werden nicht unterstützt.

Schnellstart

Das Klonen von Volumes lässt sich in wenigen Schritten einrichten.

1**Konfigurieren Sie die Quell-PVC zum Klonen des Volumes**

Der Inhaber des Quell-Namespace erteilt die Berechtigung zum Zugriff auf die Daten im Quell-PVC.

2**Erteilen Sie die Berechtigung zum Erstellen eines CR im Ziel-Namespace**

Der Clusteradministrator erteilt dem Eigentümer des Ziel-Namespace die Berechtigung, die TridentVolumeReference CR zu erstellen.

3**Erstellen Sie eine TridentVolumeReference im Ziel-Namespace.**

Der Eigentümer des Ziel-Namespace erstellt die TridentVolumeReference CR, um auf die Quell-PVC zu verweisen.

4**Erstellen Sie den Klon-PVC im Ziel-Namespace**

Der Eigentümer des Ziel-Namensraums erstellt einen PVC, um den PVC aus dem Quell-Namensraum zu klonen.

Konfigurieren Sie die Quell- und Ziel-Namespace

Um die Sicherheit zu gewährleisten, erfordert das Klonen von Volumes über Namespaces hinweg die Zusammenarbeit und das Eingreifen des Quell-Namespace-Besitzers, des Cluster-Administrators und des Ziel-Namespace-Besitzers. Die Benutzerrolle wird in jedem Schritt festgelegt.

Schritte

1. **Quellnamespace-Inhaber:** PVC erstellen(`pvc1`) im Quell-Namespace(`namespace1`) die die Berechtigung zur gemeinsamen Nutzung mit dem Ziel-Namespace erteilt(`namespace2`) unter Verwendung der `cloneToNamespace` Anmerkung.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/cloneToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Trident erstellt das PV und sein Backend-Speichervolumen.



- Sie können die PVC mithilfe einer durch Kommas getrennten Liste an mehrere Namensräume weitergeben. Zum Beispiel,
trident.netapp.io/cloneToNamespace:
namespace2,namespace3,namespace4 .
- Sie können die Freigabe für alle Namensräume nutzen * . Zum Beispiel,
trident.netapp.io/cloneToNamespace: *
- Sie können das PVC aktualisieren, um Folgendes einzuschließen: cloneToNamespace
Anmerkungen jederzeit möglich.

2. **Cluster-Administrator:** Stellen Sie sicher, dass die korrekte rollenbasierte Zugriffssteuerung (RBAC) eingerichtet ist, um dem Besitzer des Ziel-Namespace die Berechtigung zum Erstellen der TridentVolumeReference-CR im Ziel-Namespace zu erteilen.(namespace2).
3. **Inhaber des Ziel-Namespace:** Erstellen Sie im Ziel-Namespace eine TridentVolumeReference CR, die auf den Quell-Namespace verweist. pvc1 .

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Inhaber des Ziel-Namensraums:** Erstellen Sie eine PVC(pvc2) im Ziel-Namespace(namespace2) unter Verwendung der cloneFromPVC oder cloneFromSnapshot , Und cloneFromNamespace
Anmerkungen zur Kennzeichnung des Quell-PVC.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/cloneFromPVC: pvc1
    trident.netapp.io/cloneFromNamespace: namespace1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Einschränkungen

- Für PVCs, die mit ontap-nas-economy-Treibern bereitgestellt wurden, werden schreibgeschützte Klone nicht unterstützt.

Volumes mit SnapMirror replizieren

Trident unterstützt Spiegelungsbeziehungen zwischen einem Quellvolume auf einem Cluster und dem Zielvolume auf dem verbundenen Cluster zur Datenreplikation für die Notfallwiederherstellung. Sie können eine namens Trident Mirror Relationship (TMR) mit einem Namensraum verwenden, um die folgenden Operationen durchzuführen:

- Spiegelbeziehungen zwischen Volumina (PVCs) erzeugen
- Spiegelbeziehungen zwischen Volumen entfernen
- Zerstöre die Spiegelbeziehungen
- Förderung des sekundären Datenvolumens während Katastrophensituationen (Ausfallereignisse).
- Führen Sie einen verlustfreien Übergang von Anwendungen von Cluster zu Cluster durch (während geplanter Failover oder Migrationen).

Replikationsvoraussetzungen

Stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind, bevor Sie beginnen:

ONTAP -Cluster

- *** Trident ***: Auf den Quell- und Ziel-Kubernetes-Clustern, die ONTAP als Backend verwenden, muss Trident Version 22.10 oder höher vorhanden sein.
- **Lizenzen**: ONTAP SnapMirror -Asynchronlizenzen, die das Data Protection-Bundle verwenden, müssen sowohl auf dem Quell- als auch auf dem Ziel ONTAP -Cluster aktiviert sein. Siehe "[SnapMirror -Lizenzübersicht in ONTAP](#)" für weitere Informationen.

Ab ONTAP 9.10.1 werden alle Lizenzen als NetApp Lizenzdatei (NLF) ausgeliefert. Dabei handelt es sich um eine einzelne Datei, die mehrere Funktionen aktiviert. Siehe "[In ONTAP One enthaltene Lizenzen](#)" für weitere Informationen.



Es wird ausschließlich der asynchrone Schutz von SnapMirror unterstützt.

Peering

- **Cluster und SVM**: Die ONTAP -Speicher-Backends müssen per Peering verbunden sein. Siehe "[Cluster- und SVM-Peering-Übersicht](#)" für weitere Informationen.



Stellen Sie sicher, dass die in der Replikationsbeziehung zwischen zwei ONTAP Clustern verwendeten SVM-Namen eindeutig sind.

- *** Trident und SVM***: Die verbundenen Remote-SVMs müssen für Trident auf dem Zielcluster verfügbar sein.

Unterstützte Treiber

NetApp Trident unterstützt die Volume-Replikation mit der NetApp SnapMirror -Technologie unter Verwendung von Speicherklassen, die von den folgenden Treibern unterstützt werden: **ontap-nas : NFS** ontap-san :



Die Volume-Replikation mit SnapMirror wird für ASA r2-Systeme nicht unterstützt. Informationen zu ASA r2-Systemen finden Sie unter ["Erfahren Sie mehr über ASA R2-Speichersysteme"](#).

Erstellen Sie ein spiegelndes PVC

Folgen Sie diesen Schritten und verwenden Sie die CRD-Beispiele, um eine Spiegelbeziehung zwischen primären und sekundären Volumes herzustellen.

Schritte

1. Führen Sie die folgenden Schritte auf dem primären Kubernetes-Cluster durch:
 - a. Erstellen Sie ein StorageClass-Objekt mit dem `trident.netapp.io/replication: true` Parameter.

Beispiel

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

- b. Erstellen Sie eine PVC mit einer zuvor erstellten StorageClass.

Beispiel

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. Erstellen Sie eine MirrorRelationship-CR mit lokalen Informationen.

Beispiel

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
```

Trident ruft die internen Informationen für das Volume und den aktuellen Datenschutzstatus (DP) des Volumes ab und füllt dann das Statusfeld der MirrorRelationship aus.

- d. Ermitteln Sie mithilfe der TridentMirrorRelationship CR den internen Namen und die SVM der PVC.

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
    localVolumeHandle:
      "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1
```

2. Führen Sie die folgenden Schritte auf dem sekundären Kubernetes-Cluster durch:
- Erstellen Sie eine StorageClass mit dem Parameter `trident.netapp.io/replication: true`.

Beispiel

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true
```

- b. Erstellen Sie eine MirrorRelationship-CR mit Ziel- und Quellinformationen.

Beispiel

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
    - localPVCName: csi-nas
      remoteVolumeHandle:
        "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

Trident erstellt eine SnapMirror Beziehung mit dem konfigurierten Beziehungsrichtliniennamen (oder dem Standardnamen für ONTAP) und initialisiert diese.

- c. Erstellen Sie eine PVC mit einer zuvor erstellten StorageClass, die als sekundäres Ziel (SnapMirror Ziel) fungieren soll.

Beispiel

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Gi
storageClassName: csi-nas
```

Trident prüft, ob die TridentMirrorRelationship CRD vorhanden ist, und schlägt fehl, das Volume zu erstellen, wenn die Beziehung nicht existiert. Wenn die Beziehung besteht, sorgt Trident dafür, dass das neue FlexVol volume auf einer SVM platziert wird, die mit der in der MirrorRelationship definierten Remote-SVM verbunden ist.

Volumenreplikationszustände

Eine Trident Mirror Relationship (TMR) ist eine CRD, die ein Ende einer Replikationsbeziehung zwischen PVCs darstellt. Das Ziel-TMR verfügt über einen Zustand, der Trident mitteilt, welcher Zustand gewünscht ist. Das Ziel-TMR hat folgende Zustände:

- **Festgestellt:** Das lokale PVC ist das Zielvolumen einer Spiegelbeziehung, und dies ist eine neue Beziehung.
- **Verbessert:** Die lokale PVC ist lesbar und beschreibbar und kann eingebunden werden; derzeit besteht keine Spiegelungsbeziehung.
- **Wiederhergestellt:** Das lokale PVC ist das Zielvolumen einer Spiegelbeziehung und war auch zuvor Teil dieser Spiegelbeziehung.
 - Der wiederhergestellte Zustand muss verwendet werden, wenn das Zielvolumen jemals in einer Beziehung zum Quellvolumen stand, da er den Inhalt des Zielvolumens überschreibt.
 - Der wiederhergestellte Zustand schlägt fehl, wenn das Volumen zuvor nicht in Beziehung zur Quelle stand.

Förderung der sekundären PVC-Funktion bei einem ungeplanten Ausfall

Führen Sie die folgenden Schritte auf dem sekundären Kubernetes-Cluster durch:

- Aktualisiere das Feld `spec.state` der TridentMirrorRelationship auf `promoted`.

Sekundäre PVC während eines geplanten Ausfalls fördern

Führen Sie während eines geplanten Failovers (Migration) die folgenden Schritte durch, um den sekundären PVC zu fördern:

Schritte

1. Auf dem primären Kubernetes-Cluster wird ein Snapshot des PVC erstellt und gewartet, bis der Snapshot erstellt ist.
2. Erstellen Sie auf dem primären Kubernetes-Cluster die SnapshotInfo CR, um interne Details zu erhalten.

Beispiel

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. Aktualisieren Sie auf dem sekundären Kubernetes-Cluster das Feld `spec.state` der TridentMirrorRelationship CR auf `promoted` und `spec.promotedSnapshotHandle` auf den internen Namen

des Snapshots.

4. Bestätigen Sie auf dem sekundären Kubernetes-Cluster, dass der Status (status.state-Feld) der TridentMirrorRelationship auf „promoted“ gesetzt ist.

Nach einem Failover eine Spiegelbeziehung wiederherstellen

Bevor Sie die Spiegelbeziehung wiederherstellen, wählen Sie die Seite aus, die Sie zur neuen primären Seite machen möchten.

Schritte

1. Stellen Sie auf dem sekundären Kubernetes-Cluster sicher, dass die Werte für das Feld *spec.remoteVolumeHandle* in der TridentMirrorRelationship aktualisiert werden.
2. Aktualisieren Sie auf dem sekundären Kubernetes-Cluster das Feld *spec.mirror* der TridentMirrorRelationship auf *reestablished*.

Zusätzliche Operationen

Trident unterstützt die folgenden Operationen auf den primären und sekundären Datenträgern:

Primäres PVC in ein neues sekundäres PVC replizieren

Stellen Sie sicher, dass Sie bereits einen primären PVC-Anschluss und einen sekundären PVC-Anschluss haben.

Schritte

1. Löschen Sie die PersistentVolumeClaim- und TridentMirrorRelationship-CRDs aus dem eingerichteten sekundären (Ziel-)Cluster.
2. Löschen Sie die TridentMirrorRelationship CRD aus dem primären (Quell-)Cluster.
3. Erstellen Sie eine neue TridentMirrorRelationship CRD auf dem primären (Quell-)Cluster für die neue sekundäre (Ziel-)PVC, die Sie einrichten möchten.

Ändern der Größe eines gespiegelten, primären oder sekundären PVC

Die PVC-Größe kann wie gewohnt angepasst werden. ONTAP erweitert automatisch alle Ziel-Flexvols, wenn die Datenmenge die aktuelle Größe überschreitet.

Replikation aus einem PVC entfernen

Um die Replikation zu entfernen, führen Sie eine der folgenden Operationen auf dem aktuellen sekundären Volume durch:

- Löschen Sie die MirrorRelationship auf dem sekundären PVC. Dadurch wird die Replikationsbeziehung unterbrochen.
- Oder aktualisieren Sie das Feld *spec.state* auf *promoted*.

Löschen Sie eine PVC (die zuvor gespiegelt wurde)

Trident prüft, ob PVCs repliziert sind, und gibt die Replikationsbeziehung frei, bevor versucht wird, das Volume zu löschen.

Löschen eines TMR

Das Löschen eines TMR auf einer Seite einer gespiegelten Beziehung führt dazu, dass der verbleibende TMR in den Status *promoted* wechselt, bevor Trident den Löschvorgang abschließt. Wenn sich das zum Löschen ausgewählte TMR bereits im Status *promoted* befindet, besteht keine Spiegelbeziehung und das TMR wird entfernt. Trident wird dann den lokalen PVC auf *ReadWrite* hochstufen. Durch diese Löschung werden SnapMirror Metadaten für das lokale Volume in ONTAP freigegeben. Wenn dieses Volume in Zukunft in einer Spiegelbeziehung verwendet wird, muss beim Erstellen der neuen Spiegelbeziehung ein neues TMR mit einem *etablierten* Volume-Replikationsstatus verwendet werden.

Spiegelbeziehungen aktualisieren, wenn ONTAP online ist

Spiegelbeziehungen können jederzeit nach ihrer Einrichtung aktualisiert werden. Sie können die `state: promoted` oder `state: reestablished` Felder, um die Beziehungen zu aktualisieren. Beim Hochstufen eines Zielvolumes zu einem regulären ReadWrite-Volume können Sie mit *promotedSnapshotHandle* einen bestimmten Snapshot angeben, auf den das aktuelle Volume wiederhergestellt werden soll.

Spiegelbeziehungen aktualisieren, wenn ONTAP offline ist

Sie können eine CRD verwenden, um ein SnapMirror -Update durchzuführen, ohne dass Trident eine direkte Verbindung zum ONTAP -Cluster haben muss. Beachten Sie das folgende Beispielformat für TridentActionMirrorUpdate:

Beispiel

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

Der `status.state` spiegelt den Zustand des TridentActionMirrorUpdate CRD wider. Es kann einen Wert von *Succeeded*, *In Progress* oder *Failed* annehmen.

CSI-Topologie verwenden

Trident kann mithilfe der folgenden Technologie gezielt Volumes erstellen und an Knoten in einem Kubernetes-Cluster anhängen: "[CSI-Topologiefunktion](#)".

Überblick

Mithilfe der CSI-Topologiefunktion kann der Zugriff auf Volumes basierend auf Regionen und Verfügbarkeitszonen auf eine Teilmenge von Knoten beschränkt werden. Cloud-Anbieter ermöglichen es Kubernetes-Administratoren heute, zonenbasierte Knoten zu erzeugen. Knoten können sich in verschiedenen Verfügbarkeitszonen innerhalb einer Region oder über mehrere Regionen verteilt befinden. Um die Bereitstellung von Volumes für Workloads in einer Multizonenarchitektur zu erleichtern, verwendet Trident die CSI-Topologie.



Erfahren Sie mehr über die CSI-Topologiefunktion. "[hier](#)," .

Kubernetes bietet zwei einzigartige Volume-Bindungsmodi:

- Mit `VolumeBindingMode` eingestellt auf `Immediate` Trident erzeugt das Volumen ohne Kenntnis der Topologie. Die Volumenbindung und die dynamische Bereitstellung werden bei der Erstellung des PVCs verarbeitet. Dies ist die Standardeinstellung. `VolumeBindingMode` eignet sich für Cluster, die keine Topologiebeschränkungen erzwingen. Persistente Volumes werden ohne Abhängigkeit von den Planungsanforderungen des anfordernden Pods erstellt.
- Mit `VolumeBindingMode` eingestellt auf `WaitForFirstConsumer` Die Erstellung und Bindung eines Persistent Volume für ein PVC wird verzögert, bis ein Pod, der das PVC verwendet, geplant und erstellt wurde. Auf diese Weise werden Volumes erstellt, die den Planungsbeschränkungen entsprechen, die durch die Topologieanforderungen erzwungen werden.



Der `WaitForFirstConsumer` Der Bindungsmodus erfordert keine Topologiebezeichnungen. Dies kann unabhängig von der CSI-Topologiefunktion verwendet werden.

Was du brauchst

Um die CSI-Topologie nutzen zu können, benötigen Sie Folgendes:

- Ein Kubernetes-Cluster, der einen ["Unterstützte Kubernetes-Version"](#)

```
kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:50:19Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- Die Knoten im Cluster sollten Bezeichnungen haben, die die Topologie verdeutlichen. (`topology.kubernetes.io/region` Und `topology.kubernetes.io/zone`). Diese Bezeichnungen **müssen auf den Knoten im Cluster vorhanden sein**, bevor Trident installiert wird, damit Trident topologiebewusst ist.

```
kubectl get nodes -o=jsonpath='{range .items[*]}[{.metadata.name},
{.metadata.labels}]{"\n"}{end}' | grep --color "topology.kubernetes.io"
[node1,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node1","kubernetes.io/os":"linux","node-role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/os":"linux","node-role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/os":"linux","node-role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-c"}]
```

Schritt 1: Erstellen Sie ein topologiebewusstes Backend

Trident -Speicher-Backends können so konzipiert werden, dass sie Volumes selektiv auf Basis von Verfügbarkeitszonen bereitstellen. Jedes Backend kann optional `supportedTopologies` Block, der eine Liste der unterstützten Zonen und Regionen darstellt. Für `StorageClasses`, die ein solches Backend verwenden, wird ein Volume nur erstellt, wenn es von einer Anwendung angefordert wird, die in einer unterstützten Region/Zone geplant ist.

Hier ist ein Beispiel für eine Backend-Definition:

YAML

```
---
version: 1
storageDriverName: ontap-san
backendName: san-backend-us-east1
managementLIF: 192.168.27.5
svm: iscsi_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-a
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-b
```

JSON

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "password",
  "supportedTopologies": [
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-a"
    },
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-b"
    }
  ]
}
```



`supportedTopologies` dient dazu, eine Liste von Regionen und Zonen pro Backend bereitzustellen. Diese Regionen und Zonen stellen die Liste der zulässigen Werte dar, die in einer StorageClass angegeben werden können. Für StorageClasses, die eine Teilmenge der im Backend bereitgestellten Regionen und Zonen enthalten, erstellt Trident ein Volume im Backend.

Sie können definieren `supportedTopologies` ebenfalls pro Speicherpool. Siehe das folgende Beispiel:

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-centrall
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-centrall
    topology.kubernetes.io/zone: us-centrall-a
  - topology.kubernetes.io/region: us-centrall
    topology.kubernetes.io/zone: us-centrall-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-centrall
        topology.kubernetes.io/zone: us-centrall-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-centrall
        topology.kubernetes.io/zone: us-centrall-b
```

In diesem Beispiel, `region` und `zone` sind die Etiketten, die den Standort des Speicherbeckens kennzeichnen. `topology.kubernetes.io/region` und `topology.kubernetes.io/zone` geben an, von wo aus die Speicherpools genutzt werden können.

Schritt 2: Topologiebewusste Speicherklassen definieren

Basierend auf den Topologiebezeichnungen, die den Knoten im Cluster zugewiesen werden, können `StorageClasses` definiert werden, die Topologieinformationen enthalten. Dadurch werden die Speicherpools bestimmt, die als Kandidaten für PVC-Anfragen dienen, sowie die Teilmenge der Knoten, die die von Trident bereitgestellten Volumes nutzen können.

Siehe das folgende Beispiel:

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata: null
name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions: null
  - key: topology.kubernetes.io/zone
    values:
      - us-east1-a
      - us-east1-b
  - key: topology.kubernetes.io/region
    values:
      - us-east1
parameters:
  fsType: ext4

```

In der oben angegebenen StorageClass-Definition, volumeBindingMode ist eingestellt auf WaitForFirstConsumer . PVCs, die mit dieser StorageClass angefordert werden, werden erst dann verarbeitet, wenn sie in einem Pod referenziert werden. Und, allowedTopologies gibt die zu verwendenden Zonen und Regionen an. Der netapp-san-us-east1 StorageClass erstellt PVCs auf dem san-backend-us-east1 Das Backend ist oben definiert.

Schritt 3: PVC-Rohr erstellen und verwenden

Nachdem die StorageClass erstellt und einem Backend zugeordnet wurde, können Sie nun PVCs erstellen.

Siehe das Beispiel. spec unten:

```

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata: null
name: pvc-san
spec: null
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: netapp-san-us-east1

```

Die Erstellung eines PVC mithilfe dieses Manifests hätte folgende Folgen:

```

kubect1 create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubect1 get pvc
NAME          STATUS      VOLUME      CAPACITY    ACCESS MODES    STORAGECLASS
AGE
pvc-san      Pending                                netapp-san-us-east1
2s
kubect1 describe pvc
Name:          pvc-san
Namespace:     default
StorageClass:  netapp-san-us-east1
Status:        Pending
Volume:
Labels:        <none>
Annotations:   <none>
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    <none>
Events:
  Type      Reason              Age    From
  ----      -
  Normal    WaitForFirstConsumer  6s     persistentvolume-controller
waiting
for first consumer to be created before binding

```

Um mit Trident ein Volumen zu erzeugen und es an das PVC zu binden, verwenden Sie das PVC in einer Kapsel. Siehe das folgende Beispiel:

```

apiVersion: v1
kind: Pod
metadata:
  name: app-pod-1
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: topology.kubernetes.io/region
                operator: In
                values:
                  - us-east1
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            matchExpressions:
              - key: topology.kubernetes.io/zone
                operator: In
                values:
                  - us-east1-a
                  - us-east1-b
  securityContext:
    runAsUser: 1000
    runAsGroup: 3000
    fsGroup: 2000
  volumes:
    - name: voll
      persistentVolumeClaim:
        claimName: pvc-san
  containers:
    - name: sec-ctx-demo
      image: busybox
      command: [ "sh", "-c", "sleep 1h" ]
      volumeMounts:
        - name: voll
          mountPath: /data/demo
      securityContext:
        allowPrivilegeEscalation: false

```

Diese PodSpec weist Kubernetes an, den Pod auf Knoten zu planen, die im Netzwerk vorhanden sind. us-east1 Region und wähle einen beliebigen Knoten aus, der in der Region vorhanden ist. us-east1-a oder us-east1-b Zonen.

Siehe folgende Ausgabe:

```
kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP              NODE
NOMINATED NODE READINESS GATES
app-pod-1     1/1     Running   0           19s   192.168.25.131  node2
<none>        <none>
kubectl get pvc -o wide
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS          AGE   VOLUMEMODE
pvc-san       Bound     pvc-ecb1e1a0-840c-463b-8b65-b3d033e2e62b  300Mi
RWO           netapp-san-us-east1  48s   Filesystem
```

Aktualisieren Sie die Backends, um Folgendes einzuschließen `supportedTopologies`

Vorhandene Backends können aktualisiert werden, um eine Liste von `supportedTopologies` mit `tridentctl backend update`. Dies betrifft nicht bereits bereitgestellte Volumes und wird nur für nachfolgende PVCs verwendet.

Weitere Informationen

- ["Ressourcen für Container verwalten"](#)
- ["Knotenselektor"](#)
- ["Affinität und Anti-Affinität"](#)
- ["Makel und Duldungen"](#)

Mit Snapshots arbeiten

Kubernetes-Volume-Snapshots von Persistent Volumes (PVs) ermöglichen die Erstellung von zeitpunktbezogenen Kopien von Volumes. Sie können einen Snapshot eines mit Trident erstellten Volumes erstellen, einen außerhalb von Trident erstellten Snapshot importieren, ein neues Volume aus einem vorhandenen Snapshot erstellen und Volume-Daten aus Snapshots wiederherstellen.

Überblick

Volume-Snapshots werden unterstützt von `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `gcp-cvs`, `azure-netapp-files`, Und `google-cloud-netapp-volumes` Fahrer.

Bevor Sie beginnen

Sie benötigen einen externen Snapshot-Controller und benutzerdefinierte Ressourcendefinitionen (CRDs), um mit Snapshots arbeiten zu können. Dies fällt in den Verantwortungsbereich des Kubernetes-Orchestrators (z. B. Kubeadm, GKE, OpenShift).

Falls Ihre Kubernetes-Distribution den Snapshot-Controller und die CRDs nicht enthält, lesen Sie bitte weiter unter [Einen Volume-Snapshot-Controller bereitstellen](#).



Erstellen Sie keinen Snapshot-Controller, wenn Sie in einer GKE-Umgebung bedarfsgesteuerte Volume-Snapshots erstellen. GKE verwendet einen integrierten, versteckten Snapshot-Controller.

Erstellen Sie einen Volume-Snapshot

Schritte

1. Erstellen Sie ein `VolumeSnapshotClass` Weitere Informationen finden Sie unter "[VolumeSnapshotClass](#)".
 - Der `driver` zeigt auf den Trident CSI-Fahrer.
 - `deletionPolicy` kann sein `Delete` oder `Retain`. Wenn eingestellt auf `Retain` Der zugrunde liegende physische Snapshot des Speicherclusters bleibt auch dann erhalten, wenn `VolumeSnapshot` Das Objekt wurde gelöscht.

Beispiel

```
cat snap-sc.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

2. Erstellen Sie einen Snapshot eines bestehenden PVC.

Beispiele

- Dieses Beispiel erstellt eine Momentaufnahme eines bestehenden PVC.

```
cat snap.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvc1-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvc1
```

- Dieses Beispiel erstellt ein Volume-Snapshot-Objekt für eine PVC namens `pvc1` und der Name des Snapshots ist festgelegt auf `pvc1-snap`. Ein `VolumeSnapshot` ist analog zu einem PVC und ist mit einem verknüpft `VolumeSnapshotContent` Objekt, das die tatsächliche Momentaufnahme darstellt.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME                                AGE
pvc1-snap                          50s
```

- Sie können die `VolumeSnapshotContent` Objekt für das `pvc1-snap` `VolumeSnapshot` durch Beschreibung. Der `Snapshot Content Name` identifiziert das `VolumeSnapshotContent`-Objekt, das diesen Snapshot bereitstellt. Der `Ready To Use` Parameter gibt an, dass der Snapshot zur Erstellung eines neuen PVC verwendet werden kann.

```
kubectl describe volumesnapshots pvc1-snap
Name:          pvc1-snap
Namespace:     default
...
Spec:
  Snapshot Class Name:    pvc1-snap
  Snapshot Content Name:  snapcontent-e8d8a0ca-9826-11e9-9807-
525400f3f660
  Source:
    API Group:
    Kind:      PersistentVolumeClaim
    Name:      pvc1
Status:
  Creation Time:  2019-06-26T15:27:29Z
  Ready To Use:   true
  Restore Size:   3Gi
...
```

Erstellen Sie eine PVC aus einem Volumen-Snapshot

Sie können verwenden `dataSource` um ein PVC mithilfe eines `VolumeSnapshots` zu erstellen, der `<pvc-name>` als Quelle der Daten. Sobald das PVC hergestellt ist, kann es an einer Kapsel befestigt und wie jedes andere PVC verwendet werden.



Die PVC wird im selben Backend wie das Quellvolume erstellt. Siehe "[KB: Die Erstellung eines PVC aus einem Trident PVC-Snapshot ist in einem alternativen Backend nicht möglich.](#)" .

Das folgende Beispiel erstellt die PVC-Datei mithilfe von `pvc1-snap` als Datenquelle.

```
cat pvc-from-snap.yaml
```

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: golden
  resources:
    requests:
      storage: 3Gi
  dataSource:
    name: pvcl-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io

```

Importieren eines Volume-Snapshots

Trident unterstützt die ["Kubernetes-Vorbereitungs-Snapshot-Prozess"](#) um dem Cluster-Administrator die Möglichkeit zu geben, einen `VolumeSnapshotContent` Objekt- und Import-Snapshots, die außerhalb von Trident erstellt wurden.

Bevor Sie beginnen

Trident muss das übergeordnete Volume des Snapshots erstellt oder importiert haben.

Schritte

1. **Cluster-Administrator:** Erstellen Sie einen `VolumeSnapshotContent` Objekt, das auf den Backend-Snapshot verweist. Dadurch wird der Snapshot-Workflow in Trident gestartet.
 - Geben Sie den Namen des Backend-Snapshots an in annotations als `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name">`.
 - Angeben `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` In `snapshotHandle` Dies sind die einzigen Informationen, die Trident vom externen Snapshotter erhält. `ListSnapshots` Anruf.



Der `<volumeSnapshotContentName>` Aufgrund von Namensbeschränkungen für CRs kann der Name des Backend-Snapshots nicht immer übereinstimmen.

Beispiel

Das folgende Beispiel erstellt ein `VolumeSnapshotContent` Objekt, das auf einen Backend-Snapshot verweist `snap-01`.

```

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotContent
metadata:
  name: import-snap-content
  annotations:
    trident.netapp.io/internalSnapshotName: "snap-01" # This is the
name of the snapshot on the backend
spec:
  deletionPolicy: Retain
  driver: csi.trident.netapp.io
  source:
    snapshotHandle: pvc-f71223b5-23b9-4235-bbfe-e269ac7b84b0/import-
snap-content # <import PV name or source PV name>/<volume-snapshot-
content-name>
  volumeSnapshotRef:
    name: import-snap
    namespace: default

```

2. **Cluster-Administrator:** Erstellen Sie die VolumeSnapshot CR, das sich auf die VolumeSnapshotContent Objekt. Hiermit wird der Zugriff zur Nutzung des/der/der/des VolumeSnapshot in einem gegebenen Namensraum.

Beispiel

Das folgende Beispiel erstellt ein VolumeSnapshot CR benannt import-snap das sich auf das bezieht VolumeSnapshotContent namens import-snap-content .

```

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: import-snap
spec:
  # volumeSnapshotClassName: csi-snapclass (not required for pre-
provisioned or imported snapshots)
  source:
    volumeSnapshotContentName: import-snap-content

```

3. **Interne Verarbeitung (keine Aktion erforderlich):** Der externe Snapshotter erkennt den neu erstellten VolumeSnapshotContent und leitet die ListSnapshots Anruf. Trident erschafft die TridentSnapshot .
 - Der externe Snapshotter setzt die VolumeSnapshotContent Zu readyToUse und die VolumeSnapshot Zu true .
 - Trident kehrt zurück readyToUse=true .
4. **Jeder Benutzer:** Erstellen Sie ein PersistentVolumeClaim um auf das Neue Bezug zu nehmen VolumeSnapshot , wo die spec.dataSource (oder spec.dataSourceRef) Name ist der

VolumeSnapshot Name.

Beispiel

Das folgende Beispiel erstellt eine PVC, die auf die VolumeSnapshot namens import-snap .

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: simple-sc
  resources:
    requests:
      storage: 1Gi
  dataSource:
    name: import-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Wiederherstellung von Volumendaten mithilfe von Snapshots

Das Snapshot-Verzeichnis ist standardmäßig ausgeblendet, um eine maximale Kompatibilität der mit dem bereitgestellten Volumes zu gewährleisten. `ontap-nas` Und `ontap-nas-economy` Fahrer. Aktivieren Sie die `.snapshot` Verzeichnis zum direkten Wiederherstellen von Daten aus Snapshots.

Verwenden Sie den Befehl „volume snapshot restore ONTAP CLI“, um ein Volume auf einen Zustand zurückzusetzen, der in einem vorherigen Snapshot aufgezeichnet wurde.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```



Beim Wiederherstellen einer Snapshot-Kopie wird die bestehende Volume-Konfiguration überschrieben. Änderungen, die an den Volumendaten vorgenommen wurden, nachdem die Snapshot-Kopie erstellt wurde, gehen verloren.

Wiederherstellung des Volumens direkt aus einem Snapshot

Trident ermöglicht die schnelle, direkte Wiederherstellung des Volumens aus einer Momentaufnahme mithilfe der `TridentActionSnapshotRestore` (TASR) CR. Diese CR fungiert als imperative Kubernetes-Aktion und bleibt nach Abschluss der Operation nicht erhalten.

Trident unterstützt die Wiederherstellung von Snapshots auf dem `ontap-san` , `ontap-san-economy` , `ontap-nas` , `ontap-nas-flexgroup` , `azure-netapp-files` , `gcp-cvs` , `google-cloud-netapp-volumes` , Und `solidfire-san` Fahrer.

Bevor Sie beginnen

Sie benötigen ein gebundenes PVC-Gehäuse und einen verfügbaren Volumen-Snapshot.

- Prüfen Sie, ob der PVC-Status „gebunden“ ist.

```
kubectl get pvc
```

- Prüfen Sie, ob der Volume-Snapshot einsatzbereit ist.

```
kubectl get vs
```

Schritte

1. TASR CR erstellen. Dieses Beispiel erstellt eine CR für PVC. `pvc1` und Volumen-Snapshot `pvc1-snapshot`.



Der TASR CR muss sich in einem Namensraum befinden, in dem auch der PVC und der VS existieren.

```
cat tasr-pvc1-snapshot.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: trident-snap
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Wenden Sie die CR an, um die Wiederherstellung aus dem Snapshot durchzuführen. Dieses Beispiel stellt eine Wiederherstellung aus einem Snapshot her. `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml
```

```
tridentactionsnapshotrestore.trident.netapp.io/trident-snap created
```

Ergebnisse

Trident stellt die Daten aus dem Snapshot wieder her. Sie können den Status der Snapshot-Wiederherstellung überprüfen:

```
kubectl get tasr -o yaml
```

```
apiVersion: trident.netapp.io/v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: trident-snap
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""
```



- In den meisten Fällen wird Trident den Vorgang im Fehlerfall nicht automatisch wiederholen. Sie müssen den Vorgang wiederholen.
- Kubernetes-Benutzer ohne Administratorrechte müssen möglicherweise vom Administrator die Berechtigung erhalten, einen TASR CR in ihrem Anwendungs-Namespace zu erstellen.

Löschen eines PV mit zugehörigen Snapshots

Beim Löschen eines persistenten Volumes mit zugehörigen Snapshots wird das entsprechende Trident Volume auf den Status „Wird gelöscht“ aktualisiert. Um das Trident -Volume zu löschen, müssen die Volume-Snapshots entfernt werden.

Einen Volume-Snapshot-Controller bereitstellen

Falls Ihre Kubernetes-Distribution den Snapshot-Controller und die CRDs nicht enthält, können Sie diese wie folgt bereitstellen.

Schritte

1. Erstellen Sie Volume-Snapshot-CRDs.

```
cat snapshot-setup.sh
```



```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Erstellen Sie den Snapshot-Controller.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



Öffnen Sie gegebenenfalls `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` und aktualisieren `namespace` zu Ihrem Namensraum.

Weiterführende Links

- ["Volumen-Snapshots"](#)
- ["VolumeSnapshotClass"](#)

Arbeiten mit Volumengruppen-Snapshots

Kubernetes-Volume-Gruppen-Snapshots von Persistent Volumes (PVs) NetApp Trident bietet die Möglichkeit, Snapshots mehrerer Volumes (eine Gruppe von Volume-Snapshots) zu erstellen. Dieser Snapshot einer Volumengruppe stellt Kopien von mehreren Volumes dar, die zum gleichen Zeitpunkt erstellt wurden.



VolumeGroupSnapshot ist eine Beta-Funktion in Kubernetes mit Beta-APIs. Für VolumeGroupSnapshot ist mindestens die Version Kubernetes 1.32 erforderlich.

Erstellen von Snapshots von Volumengruppen

Die Snapshot-Funktion für Volumengruppen wird unterstützt. `ontap-san` Treiber nur für das iSCSI-Protokoll, noch nicht unterstützt mit Fibre Channel (FCP) oder NVMe/TCP. Bevor Sie beginnen

- Stellen Sie sicher, dass Ihre Kubernetes-Version K8s 1.32 oder höher ist.
- Sie benötigen einen externen Snapshot-Controller und benutzerdefinierte Ressourcendefinitionen (CRDs), um mit Snapshots arbeiten zu können. Dies fällt in den Verantwortungsbereich des Kubernetes-Orchestrators (z. B. Kubeadm, GKE, OpenShift).

Falls Ihre Kubernetes-Distribution den externen Snapshot-Controller und die CRDs nicht enthält, lesen Sie bitte weiter unter [Einen Volume-Snapshot-Controller bereitstellen](#).



Erstellen Sie keinen Snapshot-Controller, wenn Sie in einer GKE-Umgebung bedarfsgesteuerte Volume-Group-Snapshots erstellen. GKE verwendet einen integrierten, versteckten Snapshot-Controller.

- Im Snapshot-Controller-YAML legen Sie Folgendes fest: `CSIVolumeGroupSnapshot` Feature-Gate auf „true“ setzen, um sicherzustellen, dass die Snapshot-Funktion für Volumengruppen aktiviert ist.
- Erstellen Sie die erforderlichen Volume-Group-Snapshot-Klassen, bevor Sie einen Volume-Group-Snapshot erstellen.
- Stellen Sie sicher, dass sich alle PVCs/Volumes auf derselben SVM befinden, um `VolumeGroupSnapshot` erstellen zu können.

Schritte

- Erstellen Sie eine `VolumeGroupSnapshotClass`, bevor Sie einen `VolumeGroupSnapshot` erstellen. Weitere Informationen finden Sie unter ["VolumeGroupSnapshotClass"](#).

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshotClass
metadata:
  name: csi-group-snap-class
  annotations:
    kubernetes.io/description: "Trident group snapshot class"
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

- Erstellen Sie PVCs mit den erforderlichen Etiketten unter Verwendung vorhandener Speicherklassen oder fügen Sie diese Etiketten zu vorhandenen PVCs hinzu.

Das folgende Beispiel erstellt die PVC-Datei mithilfe von `pvc1-group-snap` als Datenquelle und Bezeichnung `consistentGroupSnapshot: groupA`. Definieren Sie den Label-Schlüssel und den Wert entsprechend Ihren Anforderungen.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvcl-group-snap
  labels:
    consistentGroupSnapshot: groupA
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: sc1-1

```

- Erstellen Sie einen VolumeGroupSnapshot mit derselben Bezeichnung(`consistentGroupSnapshot: groupA`) im PVC spezifiziert.

Dieses Beispiel erstellt einen Snapshot einer Volumengruppe:

```

apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshot
metadata:
  name: "vgs1"
  namespace: trident
spec:
  volumeGroupSnapshotClassName: csi-group-snap-class
  source:
    selector:
      matchLabels:
        consistentGroupSnapshot: groupA

```

Wiederherstellung von Volumendaten mithilfe eines Gruppen-Snapshots

Einzelne persistente Volumes können mithilfe der einzelnen Snapshots wiederhergestellt werden, die im Rahmen des Volume Group Snapshots erstellt wurden. Der Volume-Gruppen-Snapshot kann nicht als Einheit wiederhergestellt werden.

Verwenden Sie den Befehl „`volume snapshot restore ONTAP CLI`“, um ein Volume auf einen Zustand zurückzusetzen, der in einem vorherigen Snapshot aufgezeichnet wurde.

```

cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive

```



Beim Wiederherstellen einer Snapshot-Kopie wird die bestehende Volume-Konfiguration überschrieben. Änderungen, die an den Volumendaten vorgenommen wurden, nachdem die Snapshot-Kopie erstellt wurde, gehen verloren.

Wiederherstellung des Volumens direkt aus einem Snapshot

Trident ermöglicht die schnelle, direkte Wiederherstellung des Volumens aus einer Momentaufnahme mithilfe der `TridentActionSnapshotRestore` (TASR) CR. Diese CR fungiert als imperative Kubernetes-Aktion und bleibt nach Abschluss der Operation nicht erhalten.

Weitere Informationen finden Sie unter ["Wiederherstellung des Volumens direkt aus einem Snapshot"](#).

Löschen eines PV mit zugehörigen Gruppen-Snapshots

Beim Löschen eines Gruppenvolume-Snapshots:

- Sie können `VolumeGroupSnapshots` als Ganzes löschen, nicht einzelne Snapshots innerhalb der Gruppe.
- Wenn `PersistentVolumes` gelöscht werden, während für dieses `PersistentVolume` ein Snapshot existiert, versetzt Trident dieses Volume in den Status "wird gelöscht", da der Snapshot entfernt werden muss, bevor das Volume sicher gelöscht werden kann.
- Wenn ein Klon mithilfe eines gruppierten Snapshots erstellt wurde und die Gruppe anschließend gelöscht werden soll, wird ein Split-on-Clone-Vorgang gestartet, und die Gruppe kann erst gelöscht werden, wenn der Split abgeschlossen ist.

Einen Volume-Snapshot-Controller bereitstellen

Falls Ihre Kubernetes-Distribution den Snapshot-Controller und die CRDs nicht enthält, können Sie diese wie folgt bereitstellen.

Schritte

1. Erstellen Sie Volume-Snapshot-CRDs.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshots.yaml
```

2. Erstellen Sie den Snapshot-Controller.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



Öffnen Sie gegebenenfalls `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` und aktualisieren namespace zu Ihrem Namensraum.

Weiterführende Links

- ["VolumeGroupSnapshotClass"](#)
- ["Volumen-Snapshots"](#)

Copyright-Informationen

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

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

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

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

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

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

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

Markeninformationen

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