



Verwenden Sie Trident

Trident

NetApp
October 29, 2026

Inhalt

Verwenden Sie Trident	1
Vorbereitung des Trident Worker-Knotens	1
Die richtigen Werkzeuge auswählen	1
Knotendiensterkennung	1
NFS-Volumes	2
iSCSI-Volumes	2
NVMe/TCP Volumes	6
SCSI über FC-Volumes	7
Bereiten Sie die Bereitstellung von SMB-Volumes vor	9
Backends konfigurieren und verwalten	10
Konfiguration der Trident Backends	10
Azure NetApp Files	11
Google Cloud NetApp Volumes	28
Konfigurieren Sie ein NetApp HCI- oder SolidFire-Backend	56
ONTAP SAN-Treiber	61
ONTAP NAS-Treiber	88
Ausgewogene Platzierung mit ONTAP Backends verwenden	122
Amazon FSx for NetApp ONTAP	124
Backends mit kubectrl erstellen	159
Backends verwalten	165
Speicherklassen erstellen und verwalten	175
Erstellen einer Trident Storage Klasse	175
Verwalten von Trident Storage Klassen	177
Volumes bereitstellen und verwalten	179
Bereitstellung eines Trident Volume	179
Ein Volume verschieben	182
Trident Volumes erweitern	185
Grenzen des RWX NVMe-Subsystems verstehen	195
Controller-Skalierbarkeit	197
Automatische Volumenerweiterung	202
Autogrow-Richtlinien verwalten	209
Importieren von Trident Volumes	218
Anpassung von Trident Volume-Namen und Labels	229
Ein Trident NFS Volume über Namespaces hinweg freigeben	232
Klonen von Trident Volumes über Namespaces hinweg	236
Volumes replizieren mit SnapMirror	238
CSI-Topologie verwenden	245
Arbeiten mit Trident Snapshots	252
Arbeiten mit Trident Volume-Gruppen-Snapshots	260

Verwenden Sie Trident

Vorbereitung des Trident Worker-Knotens

Alle Worker-Knoten im Kubernetes-Cluster müssen die für Ihre Pods bereitgestellten Volumes einbinden können. 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 erforderlichen Tools für Ihre Treiber installieren. Neuere Versionen von Red Hat Enterprise Linux CoreOS (RHCOS) haben die Tools standardmäßig installiert.

NFS-Tools

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

iSCSI-Tools

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

NVMe-Tools

["Installieren Sie die NVMe-Tools"](#) wenn Sie `ontap-san` für Nonvolatile Memory Express (NVMe) über TCP (NVMe/TCP)-Protokoll verwenden.



NetApp empfiehlt ONTAP 9.12 oder höher für NVMe/TCP.

SCSI über FC-Tools

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

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

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

SMB-Tools

["Bereiten Sie die Bereitstellung von SMB-Volumes vor"](#) wenn Sie `ontap-nas` verwenden, um SMB-Volumes bereitzustellen.

Knotendiensterkennung

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



Die Knotendiensterkennung identifiziert gefundene Dienste, garantiert jedoch nicht, dass die Dienste ordnungsgemäß konfiguriert sind. Umgekehrt garantiert das Fehlen eines gefundenen Dienstes nicht, dass die Volume-Einbindung fehlschlägt.

Ereignisse überprüfen

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

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

Entdeckte Dienste überprüfen

Trident identifiziert die für jeden Knoten auf dem Trident node CR aktivierten Dienste. Um die ermittelten Dienste anzuzeigen, führen Sie 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-Service während der Boot-Zeit 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, sie formatieren und in einen Pod einbinden.

iSCSI Selbstreparaturfunktionen

Für ONTAP-Systeme führt Trident alle fünf Minuten eine iSCSI-Selbstreparatur durch, um:

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



Protokolle der Selbstreparaturaktivität befinden sich im `trident-main` Container auf dem

jeweiligen Daemonset-Pod. Um Protokolle anzuzeigen, müssen Sie während der Trident-Installation `debug` auf „true“ gesetzt haben.

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

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



Wenn beispielsweise CHAP-Geheimnisse auf dem Speichercontroller rotiert wurden und die Netzwerkverbindung abbricht, können die alten (*veralteten*) CHAP-Geheimnisse erhalten bleiben. Selbstreparatur kann 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, wird die iSCSI-Selbstreparatur SCSI-Neuscans für alle Geräte im SCSI-Bus initiieren.
- Wenn nur Backend-bezogene igroups (veraltet ab 23.04) verwendet werden, wird die iSCSI-Selbstreparatur SCSI-Scans für exakte LUN-IDs im SCSI-Bus initiieren.
- Wenn eine Mischung aus knotenbezogenen igroups und backendbezogenen igroups verwendet wird, initiiert die iSCSI-Selbstreparatur SCSI-Rescans für exakte LUN-IDs im SCSI-Bus.

Installieren Sie die iSCSI-Tools

Installieren Sie die iSCSI-Tools mit den Befehlen für Ihr Betriebssystem.

Bevor Sie beginnen

- Jeder Knoten im Kubernetes-Cluster muss einen eindeutigen IQN besitzen. **Dies ist eine notwendige Voraussetzung.**
- Wenn Sie RHCOS Version 4.5 oder höher oder eine andere RHEL-kompatible Linux-Distribution mit dem `solidfire-san` Treiber und Element OS 12.5 oder früher verwenden, stellen Sie sicher, dass der CHAP-Authentifizierungsalgorithmus in `/etc/iscsi/iscsid.conf` auf MD5 eingestellt ist. Sichere, FIPS-konforme CHAP-Algorithmen SHA1, SHA-256 und SHA3-256 sind mit Element 12.7 verfügbar.

```
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 die `discard mountOption` in der StorageClass an, um eine Inline-Speicherplatzfreigabe durchzuführen. Weitere Informationen sind unter ["Red Hat Dokumentation"](#) zu finden.
- Stellen Sie sicher, dass Sie auf die neueste Version des `multipath-tools` aktualisiert haben.

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 manuell 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
```



Stellen Sie sicher, dass `/etc/multipath.conf` `find_multipaths no` unter `defaults` enthalten ist.

5. Stellen Sie sicher, dass iscsid und multipathd ausgeführt werden:

```
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) oder 2.0.874-7.1ubuntu6.1 oder höher (für focal) ist:

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

3. Scanvorgang auf manuell 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
```



Stellen Sie sicher, dass `/etc/multipath.conf` `find_multipaths no` unter `defaults` enthalten ist.

5. Stellen Sie sicher, dass `open-iscsi` und `multipath-tools` aktiviert sind und ausgeführt werden:

```
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 `iscsiadm` entdecken, bevor Sie `open-iscsi` starten, damit der iSCSI-Daemon gestartet werden kann. Alternativ können Sie den `iscsi` Dienst so ändern, dass er `iscsid` automatisch startet.

iSCSI-Selbstreparatur konfigurieren oder deaktivieren

Sie können die folgenden Trident iSCSI Selbstreparatur-Einstellungen konfigurieren, um veraltete Sitzungen zu beheben:

- **iSCSI Selbstreparaturintervall:** Legt die Häufigkeit fest, mit der die iSCSI Selbstreparatur aufgerufen wird (Standard: 5 Minuten). Sie können es so konfigurieren, dass es häufiger durch die Angabe einer kleineren Zahl oder seltener durch die Angabe einer größeren Zahl ausgeführt wird.



Durch das Setzen des iSCSI-Selbstreparaturintervalls auf 0 wird die iSCSI-Selbstreparatur vollständig gestoppt. Wir empfehlen nicht, die iSCSI-Selbstreparatur zu deaktivieren; sie sollte nur in bestimmten Szenarien deaktiviert werden, wenn die iSCSI-Selbstreparatur nicht wie

vorgesehen funktioniert oder zu Debugging-Zwecken.

- **iSCSI-Selbstreparatur-Wartezeit:** Bestimmt die Dauer, die die iSCSI-Selbstreparatur wartet, bevor eine fehlerhafte Sitzung ausgeloggt und ein erneuter Anmeldeversuch unternommen wird (Standard: 7 Minuten). Sie können einen höheren Wert einstellen, sodass Sitzungen, die als fehlerhaft erkannt werden, länger warten müssen, bevor sie ausgeloggt werden und dann ein erneuter Anmeldeversuch unternommen wird, oder einen niedrigeren Wert, um früher auszuloggen und sich erneut anzumelden.

Helm

Um die Einstellungen für die iSCSI-Selbstreparatur 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 Selbstreparaturintervall auf 3 Minuten und die Selbstreparatur-Wartezeit auf 6 Minuten eingestellt:

```
helm install trident trident-operator-100.2606.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

tridentctl

Um die Einstellungen für die iSCSI-Selbstreparatur 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 Selbstreparaturintervall auf 3 Minuten und die Selbstreparatur-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 mit den Befehlen für Ihr Betriebssystem.



- NVMe erfordert RHEL 9 oder neuer.
- 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
```

Installation überprüfen

Überprüfen Sie nach der Installation, ob jeder Knoten im Kubernetes-Cluster einen eindeutigen NQN besitzt, indem Sie den folgenden Befehl verwenden:

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



Trident ändert den `ctrl_device_tmo` Wert, um sicherzustellen, dass NVMe den Pfad nicht aufgibt, falls er ausfällt. Ändern Sie diese Einstellung nicht.

SCSI über FC-Volumes

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

Voraussetzungen

Konfigurieren Sie die erforderlichen Netzwerk- und Knoteneinstellungen für FC.

Netzwerkeinstellungen

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

Siehe die entsprechenden Dienstprogramme des Host-Betriebssystems.

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

Informationen finden Sie in der jeweiligen Switch-Anbieter-Dokumentation.

Weitere Einzelheiten entnehmen Sie bitte der folgenden ONTAP-Dokumentation:

- ["Fibre Channel und FCoE-Zoning-Übersicht"](#)
- ["Möglichkeiten zur Konfiguration von FC- 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 werden, geben Sie die `discard mountOption` in der `StorageClass` an, um die Inline-Speicherplatzfreigabe durchzuführen. Weitere Informationen sind unter ["Red Hat Dokumentation"](#) zu finden.

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
```



Stellen Sie sicher, dass `/etc/multipath.conf` `find_multipaths no` unter `defaults` enthalten ist.

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 scsiboot
```

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
```



Stellen Sie sicher, dass `/etc/multipath.conf` `find_multipaths no` unter `defaults` enthalten ist.

3. Stellen Sie sicher, dass `multipath-tools` aktiviert ist und ausgeführt wird:

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

Bereiten Sie die Bereitstellung von SMB-Volumes vor

Sie können SMB-Volumes mit `ontap-nas` Treibern bereitstellen.



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



`autoExportPolicy` is not supported for SMB-Volumes.

Bevor Sie beginnen

Bevor Sie SMB-Volumes bereitstellen können, müssen Sie Folgendes haben.

- Ein Kubernetes-Cluster mit einem Linux-Controller-Knoten und mindestens einem Windows-Worker-Knoten, auf dem Windows Server 2022 läuft. Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten ausgeführt werden.
- Mindestens ein Trident secret, das Ihre Active Directory-Anmeldeinformationen enthält. Um ein secret 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 `csi-proxy` zu konfigurieren, siehe "[GitHub: CSI Proxy](#)" oder "[GitHub: CSI-Proxy für Windows](#)" für Kubernetes-Knoten, die unter Windows laufen.

Schritte

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



SMB-Shares sind für Amazon FSx for ONTAP erforderlich.

Sie können die SMB-Administratorfreigaben auf zwei Arten erstellen: entweder mit dem "[Microsoft Management Console](#)" Shared Folders Snap-In oder mit der ONTAP CLI. Um die SMB-Freigaben mit der ONTAP CLI zu erstellen:

- a. Erstellen Sie bei Bedarf die Verzeichnispfadstruktur für die Freigabe.

Der `vserver cifs share create` Befehl prüft den in der Option `-path` beim Erstellen der Freigabe angegebenen Pfad. Existiert der angegebene Pfad nicht, 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
```



Weitere Einzelheiten finden Sie in ["Erstellen Sie eine SMB-Freigabe"](#).

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

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

Backends konfigurieren und verwalten

Konfiguration der Trident Backends

Ein Backend definiert die Beziehung zwischen Trident und einem Speichersystem. Es gibt Trident vor, wie Trident mit diesem Speichersystem kommuniziert und wie Trident Volumes daraus bereitstellen soll.

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

- ["Konfigurieren eines Azure NetApp Files-Backends"](#)
- ["Konfigurieren eines Google Cloud NetApp Volumes-Backends"](#)
- ["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"
- "Verwenden Sie Trident mit Amazon FSx for NetApp ONTAP"

Azure NetApp Files

Konfigurieren eines Azure NetApp Files-Backends

Verwenden Sie Azure NetApp Files als Backend für Trident. Dieses Backend unterstützt NFS- und SMB-Volumes. Trident unterstützt verwaltete Identitäten und Workload-Identität für Azure Kubernetes Service (AKS)-Cluster.

Unterstützte Azure Cloud-Umgebungen

Trident unterstützt Azure NetApp Files-Backends in mehreren Azure Cloud-Umgebungen.

Unterstützte Azure-Clouds umfassen:

- Azure Commercial
- Azure Government (Azure Government / MAG)
- Azure China

Wenn Sie Trident bereitstellen oder ein Azure NetApp Files-Backend konfigurieren, stellen Sie sicher, dass Azure Resource Manager und Authentifizierung-Endpunkte mit Ihrer Azure-Cloudumgebung übereinstimmen.

Überprüfen Sie die Unterstützung des Azure NetApp Files-Treibers

Trident stellt den folgenden Azure NetApp Files-Speichertreiber bereit.

Unterstützte Zugriffsmodi sind *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX) und *ReadWriteOncePod* (RWOP).

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

Überlegungen zur Überprüfung

- Azure NetApp Files unterstützt keine Volumes unter 50 GiB. Trident erstellt ein 50-GiB-Volume, wenn ein kleineres Volume angefordert wird.
- Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten ausgeführt werden.
- Azure NetApp Files-Bereitstellungen in nicht-kommerziellen Azure-Clouds erfordern cloud-spezifische Azure Resource Manager- und Authentifizierungsendpunkte. Stellen Sie sicher, dass Trident und jede Backend-Konfiguration die für Ihre Azure-Cloud-Umgebung geeigneten Endpunkte verwenden.

Verwenden Sie verwaltete Identitäten für AKS

Trident unterstützt "verwaltete Identitäten" für AKS-Cluster.

Wenn Sie `tridentctl` verwenden, um Azure NetApp Files Backends zu erstellen oder zu verwalten, stellen Sie sicher, dass es für die richtige Azure-Cloudumgebung konfiguriert ist.

Um verwaltete Identitäten zu verwenden, müssen Sie Folgendes haben:

- Ein mit AKS bereitgestellter Kubernetes-Cluster
- Auf dem AKS Kubernetes-Cluster konfigurierte verwaltete Identitäten
- Trident installiert mit `cloudProvider` auf "Azure"

Trident Operator

Bearbeiten `tridentorchestrator_cr.yaml` und setzen `cloudProvider` auf "Azure".

```
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 und setzt `cloudProvider` mithilfe der Umgebungsvariable `$CP`:

```
helm install trident trident-operator-100.2606.0.tgz --create-namespace
--namespace <trident-namespace> --set cloudProvider=$CP
```

`tridentctl`

Das folgende Beispiel installiert Trident und setzt die `cloud-provider-Flag` auf Azure:

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

Workload-Identität für AKS verwenden

Die Workload-Identität ermöglicht es Kubernetes-Pods, auf Azure-Ressourcen zuzugreifen, indem sie sich als Workload-Identität authentifizieren.

Wenn Sie `tridentctl` verwenden, um Azure NetApp Files Backends zu erstellen oder zu verwalten, stellen Sie sicher, dass es für die richtige Azure-Cloudumgebung konfiguriert ist.

Um die Workload-Identität zu verwenden, müssen Sie Folgendes haben:

- Ein mit AKS bereitgestellter Kubernetes-Cluster
- Workload-Identität und oidc-issuer auf dem AKS Kubernetes-Cluster konfiguriert
- Trident wurde mit `cloudProvider` auf "Azure" gesetzt und `cloudIdentity` auf den Wert der Workload-Identität installiert

Trident Operator

Bearbeiten `tridentorchestrator_cr.yaml` und setzen `cloudProvider` auf "Azure". Setzen Sie `cloudIdentity` auf `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`.

```
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-xxxxxxxxxx' # 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-xxxxxxxxxx'"
```

Das folgende Beispiel installiert Trident und setzt `cloudProvider` mit `$CP` und setzt `cloudIdentity` mit `$CI`:

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

`tridentctl`

Legen Sie die Werte für die Flags **cloud provider** und **cloud identity** mithilfe der folgenden Umgebungsvariablen fest:

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

Das folgende Beispiel installiert Trident und setzt `cloud-provider` auf `$CP` und `cloud-identity` auf

\$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.

Unterstützte Azure Cloud-Umgebungen

Trident unterstützt Azure NetApp Files-Backends in mehreren Azure Cloud-Umgebungen.

Unterstützte Azure-Clouds umfassen:

- Azure Commercial
- Azure Government (Azure Government / MAG)
- Azure China

Stellen Sie bei der Vorbereitung Ihrer Umgebung sicher, dass Ihr Azure-Abonnement, Ihre Identitätskonfiguration und Ihre Azure NetApp Files-Ressourcen in der entsprechenden Azure-Cloudumgebung erstellt werden.

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. Weitere Informationen sind unter ["Azure: Azure NetApp Files einrichten und ein NFS-Volume erstellen"](#) zu finden.

Um ein ["Azure NetApp Files"](#) Backend zu konfigurieren und zu verwenden, benötigen Sie Folgendes:



- `subscriptionID`, `tenantID`, `clientID`, `location` und `clientSecret` sind optional, wenn verwaltete Identitäten auf einem AKS-Cluster verwendet werden.
- `tenantID`, `clientID` und `clientSecret` sind optional, wenn eine Cloud-Identität auf einem AKS-Cluster verwendet wird.
- Azure NetApp Files-Bereitstellungen in nicht-kommerziellen Azure-Clouds erfordern cloud-spezifische Azure Resource Manager- und Authentifizierungsendpunkte. Stellen Sie sicher, dass Trident und jede Backend-Konfiguration die für Ihre Azure-Cloud-Umgebung geeigneten Endpunkte verwenden.

- Ein Kapazitätspool. Weitere Informationen enthält ["Microsoft: Erstellen eines Kapazitätspools für Azure NetApp Files"](#).
- Ein an Azure NetApp Files delegiertes Subnetz. Weitere Informationen enthält ["Microsoft: Delegieren Sie ein Subnetz an Azure NetApp Files"](#).
- `subscriptionID` aus einem Azure-Abonnement mit aktivierten Azure NetApp Files.
- `tenantID`, `clientID` und `clientSecret` von einem ["App-Registrierung"](#) in Azure Active Directory mit

ausreichenden Berechtigungen für den Azure NetApp Files Service. Die App-Registrierung sollte entweder Folgendes verwenden:

- Die Rolle Eigentümer oder Mitwirkender ["von Azure vordefiniert"](#).
- A ["benutzerdefinierte Contributor-Rolle"](#) auf Abonnementebene (assignableScopes) mit den folgenden Berechtigungen, die auf nur das beschränkt sind, was Trident benötigt. Nach dem Erstellen der benutzerdefinierten Rolle ["Weisen Sie die Rolle mithilfe des Azure-Portals zu"](#).

▼ Benutzerdefinierte Contributor-Rolle

```
{
  "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/subscriptionFeatureRegistratio
ns/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistratio
ns/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 Azure location, die mindestens einen ["delegiertes Subnetz"](#) enthält. Ab Trident 22.01 ist der location-Parameter ein Pflichtfeld auf oberster Ebene der Backend-Konfigurationsdatei. In virtuellen Pools angegebene Standortwerte werden ignoriert.
- Um Cloud Identity zu verwenden, holen Sie die client ID von einem ["vom Benutzer zugewiesene verwaltete Identität"](#) und geben Sie diese ID in azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx an.

Zusätzliche Anforderungen für SMB-Volumina

Zum Erstellen eines SMB-Volumes müssen Sie Folgendes haben:

- Active Directory ist konfiguriert und mit Azure NetApp Files verbunden. Weitere Informationen enthält ["Microsoft: Active Directory-Verbindungen für Azure NetApp Files erstellen und verwalten"](#).
- Ein Kubernetes-Cluster mit einem Linux-Controller-Knoten und mindestens einem Windows-Worker-Knoten, auf dem Windows Server 2022 läuft. Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten ausgeführt werden.
- Mindestens ein Trident-Geheimnis, das Ihre Active Directory-Anmeldeinformationen enthält, damit Azure NetApp Files sich bei Active Directory authentifizieren kann. Um ein secret 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 csi-proxy zu konfigurieren, siehe ["GitHub"](#):

[CSI Proxy](#)" oder "[GitHub: CSI-Proxy für Windows](#)" für Kubernetes-Knoten, die unter Windows laufen.

Azure NetApp Files Backend-Konfigurationsoptionen und Beispiele

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, Servicelevel und Standort), um Azure NetApp Files Volumes auf Kapazitätspools zu erstellen, die am angeforderten Standort verfügbar sind und dem angeforderten Servicelevel und Subnetz entsprechen.

Azure NetApp Files-Backends bieten diese Konfigurationsoptionen.

Parameter	Beschreibung	Standard
version	Backend-Konfigurationsversion.	Immer 1
storageDriverName	Name des Speichertreibers	"azure-netapp-files"
backendName	Benutzerdefinierter Name für das Speicher-Backend	Driver name + "_" + 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 Optional, wenn verwaltete Identitäten oder Cloud-Identität auf einem AKS-Cluster verwendet wird.	
clientID	Die Client-ID aus einer App-Registrierung Optional, wenn verwaltete Identitäten oder Cloud-Identität auf einem AKS-Cluster verwendet wird.	
clientSecret	Das Client Secret aus einer App Registration ist optional, wenn verwaltete Identitäten oder Cloud Identity auf einem AKS-Cluster verwendet werden.	
serviceLevel	Eines 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.	

Parameter	Beschreibung	Standard
cloudConfiguration.cloudName	Gibt die Azure-Cloudumgebung für das Backend an. Wird auf AzureGovernment festgelegt, wenn Azure Government verwendet wird.	
resourceGroups	Liste der Ressourcengruppen zum Filtern entdeckter Ressourcen	[] (kein Filter)
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 Basic oder Standard sein. Network Features ist nicht in allen Regionen verfügbar und muss möglicherweise in einem Abonnement aktiviert werden. Die Angabe von networkFeatures wenn die Funktionalität nicht aktiviert ist, führt dazu, dass die Volume-Bereitstellung fehlschlägt.	""
nfsMountOptions	Detaillierte Steuerung der NFS-Mountoptionen. Wird für SMB-Volumes ignoriert. Um Volumes mit NFS Version 4.1 zu mounten, schließen Sie nfsvers=4 in die durch Kommas getrennte Liste der Mountoptionen ein, um NFS v4.1 auszuwählen. Mountoptionen, die in einer Speicherklassendefinition festgelegt sind, überschreiben Mountoptionen, die in der Backend-Konfiguration festgelegt sind.	"nfsvers=3"
limitVolumeSize	Fehler bei der Bereitstellung, wenn die angeforderte Volume-Größe über diesem Wert liegt	"" (wird nicht standardmäßig erzwungen)
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, \{"api": false, "method": true, "discovery": true\}. Verwenden Sie dies nur, wenn Sie eine detaillierte Protokollausgabe zur Fehlersuche benötigen.	null

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



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

Berücksichtigen Sie Azure-Cloud-Umgebungen (26.06)

Ab der Version 26.06 unterstützt Trident das Erstellen und Verwalten von Azure NetApp Files-Backends in mehreren Azure-Cloud-Umgebungen.

Unterstützte Azure-Clouds umfassen:

- Azure Commercial
- Azure Government (Azure Government / MAG)
- Azure China

Stellen Sie bei der Bereitstellung von Trident oder der Erstellung eines Azure NetApp Files-Backends sicher, dass Azure Resource Manager und Authentifizierung mit Ihrer Azure-Cloudumgebung übereinstimmen. Stimmen die Endpunkte nicht überein, `tridentctl` kann keine Authentifizierung durchgeführt werden und die Backend-Erstellung schlägt fehl.

Für ein Azure Government Backend wird `cloudConfiguration.cloudName` in der Backend-Konfiguration festgelegt:

```
cloudConfiguration:
  cloudName: AzureGovernment
```

Erforderliche Berechtigungen und Ressourcen

Wenn beim Erstellen eines PVC die Fehlermeldung „Keine Kapazitätspools gefunden“ angezeigt wird, verfügt Ihre App-Registrierung wahrscheinlich nicht über die erforderlichen Berechtigungen und Ressourcen (Subnetz, virtuelles Netzwerk, Kapazitätspool), die zugeordnet sind. Wenn Debug aktiviert ist, protokolliert Trident die

beim Erstellen des Backends ermittelten Azure-Ressourcen. Stellen Sie sicher, dass eine geeignete Rolle verwendet wird.

Die Werte für `resourceGroups`, `netappAccounts`, `capacityPools`, `virtualNetwork` und `subnet` können mithilfe von Kurznamen oder vollqualifizierten Namen angegeben werden. Vollqualifizierte Namen werden in den meisten Situationen empfohlen, da Kurznamen mehreren Ressourcen mit demselben Namen zugeordnet werden können.



Wenn sich das vNet in einer anderen Ressourcengruppe als das Azure NetApp Files (ANF)-Speicherkonto befindet, geben Sie die Ressourcengruppe für das virtuelle Netzwerk bei der Konfiguration der `resourceGroups`-Liste für das Backend an.

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

Typ	Format
Ressourcengruppe	<resource group>
NetApp-Konto	<resource group>/<netapp account>
Kapazitätspool	<resource group>/<netapp account>/<capacity pool>
Virtuelles Netzwerk	<resource group>/<virtual network>
Subnetz	<resource group>/<virtual network>/<subnet>

Volumenbereitstellung

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

Parameter	Beschreibung	Standard
<code>exportRule</code>	Exportregeln für neue Volumes. <code>exportRule</code> muss eine durch Kommas getrennte Liste beliebiger Kombinationen von IPv4-Adressen oder IPv4-Subnetzen in CIDR-Notation sein. Wird für SMB-Volumes ignoriert.	"0.0.0.0/0"
<code>snapshotDir</code>	Zugriff auf das <code>.snapshot</code> Verzeichnis	true, false (explizit festlegen).
<code>size</code>	Die Standardgröße neuer Volumes	"100G"
<code>unixPermissions</code>	Die Unix-Berechtigungen neuer Volumes (4 Oktalstellen). Wird für SMB-Volumes ignoriert.	"" (Vorschaufunktion, erfordert Whitelisting im Abonnement)

Beispielkonfigurationen

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

▼ Minimale Konfiguration

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

Diese Konfiguration ist ideal, wenn Sie gerade erst mit Azure NetApp Files beginnen und Dinge ausprobieren, aber in der Praxis werden Sie 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 `subscriptionID`, `tenantID`, `clientID` und `clientSecret` aus, 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:
    - resource-group-1/netapp-account-1/ultra-pool
  resourceGroups:
    - resource-group-1
  netappAccounts:
    - resource-group-1/netapp-account-1
  virtualNetwork: resource-group-1/eastus-prod-vnet
  subnet: resource-group-1/eastus-prod-vnet/eastus-anf-subnet
```

▼ Cloud-Identität für AKS

Diese Backend-Konfiguration lässt `tenantID`, `clientID` und `clientSecret` aus, 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 im `eastus` Standort von Azure in einem `Ultra` Kapazitätspool. Trident erkennt automatisch alle an Azure NetApp Files in diesem Standort delegierten Subnetze 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 im `eastus` Standort von Azure mit manuellen QoS-Kapazitätspools.

```

---
version: 1
storageDriverName: azure-netapp-files
backendName: anf1
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 Standardwerte 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: application-group-1/eastus-prod-vnet

```

```

subnet: application-group-1/eastus-prod-vnet/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 des virtuellen Pools

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

```

---
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:
        - application-group-1/netapp-account-1/ultra-1
        - application-group-1/netapp-account-1/ultra-2
      networkFeatures: Standard
  - labels:
      performance: silver
      serviceLevel: Premium
      capacityPools:
        - application-group-1/netapp-account-1/premium-1
  - labels:
      performance: bronze
      serviceLevel: Standard

```

```
capacityPools:
  - application-group-1/netapp-account-1/standard-1
  - application-group-1/netapp-account-1/standard-2
```

▼ Unterstützte Topologiekonfiguration

Trident erleichtert die Bereitstellung von Volumes für Workloads basierend auf Regionen und Verfügbarkeitszonen. Der `supportedTopologies` Block in dieser Backend-Konfiguration wird verwendet, um eine Liste von Regionen und Zonen pro Backend bereitzustellen. Die hier angegebenen Regions- und Zonenwerte müssen mit den Regions- und Zonenwerten aus den 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 in einem 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 folgenden `StorageClass` Definitionen beziehen sich auf die oben genannten Speicherpools.

Beispieldefinitionen unter Verwendung des `parameter.selector` Felds

Mit `parameter.selector` können Sie für jedes `StorageClass` den virtuellen Pool angeben, der zum Hosten eines Volumes verwendet wird. Das Volume 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

Mit `nasType`, `node-stage-secret-name` und `node-stage-secret-namespace` können Sie 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 Secrets pro Volume

```
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`filtert für NFS-Pools.

Das Backend erstellen

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

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

Wenn Sie eine nicht-kommerzielle Azure-Cloud verwenden, stellen Sie sicher, dass `tridentctl` für die Verwendung des Azure Resource Manager und der Authentifizierung-Endpunkte für Ihre Azure-Cloudumgebung konfiguriert ist. Wenn die Backend-Erstellung fehlschlägt, überprüfen Sie Ihre Backend-Konfiguration und sehen Sie sich die Protokolle an, um die Ursache zu ermitteln:

```
tridentctl logs
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den `create`-Befehl erneut ausführen.

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes konfigurieren

Sie können Google Cloud NetApp Volumes als Backend für Trident konfigurieren, um Speicher für Kubernetes-Workloads bereitzustellen.

Überblick

Trident unterstützt Google Cloud NetApp Volumes sowohl für NAS- (NFS und SMB) als auch für Block-Workloads (iSCSI).

- NAS-Workloads nutzen das `google-cloud-netapp-volumes` Backend
- Block-Workloads (iSCSI) nutzen das `google-cloud-netapp-volumes-san` Backend

NAS-Volumes bieten dateibasierten Speicher und werden über NFS- oder SMB-Protokolle angesprochen. Diese Volumes unterstützen den gemeinsamen Zugriff durch mehrere Pods oder Knoten.

Block-Volumes bieten rohen Blockspeicher und werden als iSCSI-Geräte angesprochen, die an Kubernetes-Knoten angeschlossen sind. Diese Volumes werden verwendet, wenn Anwendungen Zugriff auf Blockebene benötigen.

Dies gilt für die folgenden Umgebungen:

- Trident 26.02 und später
- Google Kubernetes Engine (GKE) oder Red Hat OpenShift
- Google Cloud NetApp Volumes-Speicherpools

Informationen zur Konfiguration von Block-Speicher (iSCSI) finden Sie unter ["Blockspeicher konfigurieren \(iSCSI\)"](#).

Bereiten Sie die Konfiguration vor

Cloud Identity ermöglicht es Kubernetes-Workloads, auf Google Cloud-Ressourcen zuzugreifen, indem sie sich als Workload-Identität authentifizieren, anstatt statische Anmeldeinformationen zu verwenden.

Um Cloud Identity mit Google Cloud NetApp Volumes zu verwenden, müssen Sie Folgendes haben:

- Ein mit Google Kubernetes Engine (GKE) bereitgestellter Kubernetes-Cluster
- Workload-Identität im GKE-Cluster aktiviert und Metadatenserver auf den Knotenpools aktiviert
- Ein Google Cloud-Dienstkonto mit der Google Cloud NetApp Volumes Admin-Rolle (`roles/netapp.admin` oder einer entsprechenden benutzerdefinierten Rolle)
- Trident wurde mit dem Cloud-Anbieter auf GCP gesetzt und der Cloud-Identitätsannotation konfiguriert installiert

Trident Operator

Um Trident mithilfe des Trident Operators zu installieren, bearbeiten Sie `tridentorchestrator_cr.yaml`:

```

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

```

Helm

Legen Sie beim Installieren von Trident mit Helm den Cloud-Anbieter und die Cloud-Identität fest:

```

helm install trident trident-operator-100.6.0.tgz \
  --set cloudProvider=GCP \
  --set cloudIdentity="iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com"

```

tridentctl

Installieren Sie Trident, indem Sie den Cloud-Anbieter und die Cloud-Identität angeben:

```

tridentctl install \
  --cloud-provider=GCP \
  --cloud-identity="iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com" \
  -n trident

```

NAS-Speicher konfigurieren



Für Google Cloud NetApp Volumes UNIFIED-Speicherpools wendet Trident während der Volume-Operationen UNIFIED-spezifische Namens- und Validierungsregeln an.

Beim Auffinden eines Volumes kann Trident mehrere kompatible Varianten des Volumennamens (z. B. Bindestrich- und Unterstrichformate) auswerten, um die Zuverlässigkeit von Import und Erkennung zu verbessern.

Treiberdetails

Trident stellt den `google-cloud-netapp-volumes` Treiber zur Verfügung, um NAS-Speicher von Google Cloud NetApp Volumes bereitzustellen.

Der Treiber unterstützt die folgenden Zugriffsmodi:

- ReadWriteOnce (RWO)

- ReadOnlyMany (ROX)
- ReadWriteMany (RWX)
- ReadWriteOncePod (RWOP)

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

Konfigurieren Sie ein Trident NAS-Backend

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
    - labels:
        cloud: gcp
        network: "<vpc-network>"

```

NAS-Volumen bereitstellen

NAS-Volumes werden mit dem google-cloud-netapp-volumes Backend bereitgestellt und unterstützen die Protokolle NFS und SMB.

StorageClass für NFS-Volumes

Um NFS-Volumes bereitzustellen, setzen Sie `nasType` auf `nfs`.

```

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

```

```
allowVolumeExpansion: true
```

StorageClass für SMB-Volumes

Um SMB-Volumes bereitzustellen, setzen Sie `nasType` auf `smb` und geben Sie die Anmeldeinformationen an.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-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"
allowVolumeExpansion: true
```

PersistentVolumeClaim-Beispiel (RWX)

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

PersistentVolumeClaim-Beispiel (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
```

```
storage: 100Gi
storageClassName: gcnv-nfs
```



NAS-Volumes verwenden `volumeMode: Filesystem`.

Google Cloud NetApp Volumes für SAN-Workloads konfigurieren

Sie können Trident so konfigurieren, dass Block-Storage-Volumes über das iSCSI-Protokoll von Google Cloud NetApp Volumes bereitgestellt werden. SAN-Volumes werden mithilfe des `google-cloud-netapp-volumes-san` Speichertreibers aus Flex Unified Storage Pools bereitgestellt.



Dieser Treiber ist für Block-Workloads ausgelegt und unterstützt keine NAS-Protokolle.



Das `google-cloud-netapp-volumes-san` Backend ist erforderlich, um iSCSI-Blockvolumes bereitzustellen. Das `google-cloud-netapp-volumes` Backend unterstützt nur NAS-Protokolle und kann nicht für SAN-Workloads verwendet werden.

Überblick

Trident unterstützt Google Cloud NetApp Volumes SAN (iSCSI)-Workloads mithilfe des `google-cloud-netapp-volumes-san` Treibers.

SAN-Volumes werden aus Flex Unified-Speicherpools bereitgestellt und den Kubernetes-Knoten als iSCSI-Blockgeräte präsentiert.

Dies gilt für die folgenden Umgebungen:

- Trident 26.02 und später
- Google Kubernetes Engine (GKE) oder Red Hat OpenShift
- Google Cloud NetApp Volumes Flex – Einheitliche Speicherpools
- iSCSI-basierte Workloads

Flex Unified Speicherpools

Flex Unified-Speicherpools bieten Blockspeicher über das iSCSI-Protokoll und sind für die SAN-Bereitstellung erforderlich:

- Flex Unified REGIONAL-Pools werden unterstützt.
- Flex Unified ZONAL-Pools werden ab Trident 26.02.1 unterstützt.
- Für SAN-Workloads wird ausschließlich die Serviceebene **Flex** unterstützt.

Konfigurieren Sie ein Trident SAN-Backend

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
```

```

name: gcnv-san
namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
      performance: flex
      network: "<vpc-network>"
      serviceLevel: Flex

```

Erstellen Sie eine StorageClass

Nach der Konfiguration des SAN-Backends erstellen Sie eine StorageClass, die auf den `google-cloud-netapp-volumes-san` Treiber verweist.

Der Dateisystemtyp wird in der StorageClass definiert, nicht im Backend.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes-san"
  fsType: "ext4"
allowVolumeExpansion: true

```

Unterstützte Dateisystemtypen:

- ext4 (Standard)
- ext3
- xfs



Der SAN-Treiber unterstützt ausschließlich das Flex-Servicelevel und verwendet keine NAS-spezifischen Backend-Parameter wie `exportRule`, `unixPermissions`, `nasType`, `snapshotDir`, `nfsMountOptions` oder tiering-bezogene Einstellungen.

Block-Volumes bereitstellen

ReadWriteOnce (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadWriteOncePod (RWOP)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwop
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadOnlyMany (ROX)

Ein gängiges Muster für ROX ist es, ein vorhandenes ReadWriteOnce-Volume zu klonen und den Klon als schreibgeschützt einzubinden.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rox
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

```
dataSource:
  kind: PersistentVolumeClaim
  name: gcnv-san-rwo
```

ReadWriteMany (RWX) — nur Rohdatenblock

ReadWriteMany wird nur unterstützt, wenn `volumeMode: Block`.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-raw-rwx
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

Blockvolumenverhalten

Block-Volumes werden als iSCSI-LUNs bereitgestellt und den Kubernetes-Knoten als Blockgeräte präsentiert.

Block-Volumes:

- Verwenden Sie das iSCSI-Protokoll
- Unterstützung der Dateisystem- und Rohblockdarstellung
- Werden von Trident angehängt und verwaltet
- Mehrere Kubernetes-Zugriffsmodi unterstützen

Zugriffsmodi

Von Trident bereitgestellte Block-Volumes unterstützen die folgenden Zugriffsmodi:

- ReadWriteOnce (RWO)
- ReadOnlyMany (ROX)
- ReadWriteOncePod (RWOP)
- ReadWriteMany (RWX), wird nur unterstützt, wenn `volumeMode: Block`

volumeMode-Verhalten

Das `volumeMode` Feld steuert, wie ein Blockvolumen angezeigt wird:

- `Filesystem` Trident formatiert und mountet das Volume.

- Block Trident verbindet das Gerät und stellt es als Raw-Blockgerät bereit.

Unterstützte Operationen

Blockvolumes, die mithilfe des `google-cloud-netapp-volumes-san` Treibers bereitgestellt werden:

- Erstellen
- Löschen
- Klonen
- Schnappschuss
- Größe ändern
- Importieren

Zusätzliches GiB-Überprovisionierungsverhalten

Google Cloud NetApp Volumes Blockvolumes beinhalten einen internen Metadaten-Overhead. Dieser Overhead reduziert die vom Kernel sichtbare Gerätegröße im Vergleich zur bereitgestellten Kapazität.

Tests zeigen:

- Ungefähr 300 KiB Overhead bei der erstmaligen Erstellung
- Bis zu ca. 107 MiB Overhead nach einer Größenänderung

Da Google Cloud NetApp Volumes nur ganze GiB-Zuweisungen akzeptiert, stellt Trident sicher, dass die nutzbare Gerätegröße die PVC-Anforderung immer erfüllt oder übertrifft, indem:

- Aufrunden der angeforderten Größe auf das nächste ganze GiB
- Hinzufügen eines zusätzlichen 1 GiB Puffers

Beispiel:

- PVC-Anfrage: 100 GiB
- Bereitgestellte Größe in Google Cloud NetApp Volumes: 101 GiB
- Für die Anwendung sichtbarer nutzbarer Speicherplatz: mindestens 100 GiB

Pod-Beispiele

Dateisystemgebundenes Blockvolume (RWO)

```
apiVersion: v1
kind: Pod
metadata:
  name: app-rwo
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
```

```

    volumeMounts:
    - name: data
      mountPath: /mnt/data
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-rwo

```

Raw-Block-Gerät (RWX)

```

apiVersion: v1
kind: Pod
metadata:
  name: app-raw-rwx
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeDevices:
    - name: data
      devicePath: /dev/xda
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-raw-rwx

```

Anfüge- und Einhängeverhalten

Für SAN-Volumes, die von Google Cloud NetApp Volumes bereitgestellt wurden:

- Trident erstellt eine Logical Unit Number (LUN) in einem Flex Unified Speicherpool.
- Während der Veröffentlichung ordnet Trident die LUN einer Hostgruppe pro Knoten zu.
- Während der Knoten-Staging-Phase führt Trident Folgendes aus:
 - Meldet sich am iSCSI-Ziel an
 - Entdeckt die LUN
 - Konfiguriert Multipath
- Wenn `volumeMode: Filesystem` erforderlich, formatiert Trident das Gerät und bindet es ein.
- Wenn `volumeMode: Block` Trident das Gerät anschließt, wird es ohne Formatierung oder Einbindung direkt dem Pod bereitgestellt.



SAN-Blockvolumes bieten keine verteilte Sperrung oder Schreibkoordination. Wenn auf ein Blockvolume von mehreren Knoten (ReadWriteMany mit `volumeMode: Block`) zugegriffen wird, muss die Anwendung oder das Dateisystem die Parallelität verwalten.

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- oder SMB-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- oder SMB-Volume zu erstellen. Weitere Informationen enthält ["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 Service konfiguriert ist. Weitere Informationen enthält ["Google Cloud NetApp Volumes"](#).
- Projektnummer Ihres Google Cloud account. Weitere Informationen enthält ["Projekte identifizieren"](#).
- Ein Google Cloud-Dienstkonto mit der NetApp Volumes Admin (`roles/netapp.admin`-Rolle. Weitere Informationen enthält ["Identitäts- und Zugriffsmanagement-Rollen und -Berechtigungen"](#).
- API-Schlüsseldatei für Ihr GCNV-Konto. Siehe ["Erstellen Sie einen Dienstkontoschlüssel"](#)
- Ein Speicherpool. Weitere Informationen enthält ["Übersicht der Storage Pools"](#).

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

Google Cloud NetApp Volumes Backend-Konfigurationsoptionen und Beispiele

Erfahren Sie mehr über 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 <code>storageDriverName</code> muss als "google-cloud-netapp-volumes" angegeben werden.
backendName	(Optional) Benutzerdefinierter Name des Storage-Backends	Driver name + "_" + Teil des API-Schlüssels
storagePools	Optionaler Parameter, der verwendet wird, um Speicherpools für die Volume-Erstellung anzugeben.	
projectNumber	Google Cloud-Konto-Projektnummer. Der Wert ist auf der Startseite des Google Cloud-Portals zu finden.	

Parameter	Beschreibung	Standard
location	Der Google Cloud-Standort, an dem Trident GCNV-Volumes erstellt. Beim Erstellen regionsübergreifender Kubernetes-Cluster können Volumes, die in einem location erstellt wurden, in Workloads verwendet werden, die auf Knoten in mehreren Google Cloud-Regionen geplant sind. Für regionsübergreifenden Datenverkehr fallen zusätzliche Kosten an.	
apiKey	API-Schlüssel für das Google Cloud-Dienstkonto mit der netapp.admin Rolle. Er enthält den JSON-formatierten Inhalt der privaten Schlüsseldatei eines Google Cloud-Dienstkontos (wortwörtlich in die Backend-Konfigurationsdatei kopiert). Die apiKey muss 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	Detaillierte Steuerung der NFS-Mountoptionen.	"nfsvers=3"
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe über diesem Wert liegt.	"" (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 Volumes angewendet werden sollen	""
network	Google Cloud-Netzwerk, das für GCNV-Volumes verwendet wird.	
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, {"api":false, "method":true}. Verwenden Sie dies nur, wenn Sie eine detaillierte Protokollausgabe zur Fehlersuche benötigen.	null
nasType	Konfigurieren Sie die Erstellung von NFS- oder SMB-Volumes. Optionen sind nfs, smb oder null. Die Einstellung auf null verwendet standardmäßig NFS-Volumes.	nfs
supportedTopologies	Stellt eine Liste der Regionen und Zonen dar, die von diesem Backend unterstützt werden. Weitere Informationen finden Sie unter " CSI-Topologie verwenden ". Beispiel: supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

Optionen zur Volumenbereitstellung

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

Parameter	Beschreibung	Standard
<code>exportRule</code>	Die Exportregeln für neue Volumes. Muss eine durch Kommas getrennte Liste beliebiger IPv4-Adressen sein.	"0.0.0.0/0"
<code>snapshotDir</code>	Zugriff auf das <code>.snapshot</code> Verzeichnis	<code>true</code> , <code>false</code> (Das Standardverhalten kann abweichen. Stellen Sie es explizit auf „false“ für NFSv3 ein)
<code>snapshotReserve</code>	Prozentsatz des für Snapshots reservierten Volumens	"" (Standardwert von 0 akzeptieren)
<code>unixPermissions</code>	Die Unix-Berechtigungen neuer Volumes (4 Oktalstellen).	""

Beispielkonfigurationen

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

▼ Minimale Konfiguration

Dies ist die absolute Minimalkonfiguration für das Backend. Mit dieser Konfiguration erkennt Trident alle Ihre an Google Cloud NetApp Volumes delegierten Speicherpools am konfigurierten Standort und platziert neue Volumes zufällig in einem dieser Pools. Da `nasType` ausgelassen wird, gilt die `nfs` Standardeinstellung und das Backend stellt NFS-Volumes bereit.

Diese Konfiguration ist ideal, wenn Sie gerade erst mit Google Cloud NetApp Volumes beginnen und Dinge ausprobieren, aber in der Praxis müssen Sie möglicherweise zusätzliche Einschränkungen für die von Ihnen bereitgestellten Volumes festlegen.



Ersetzen Sie `<id_value>` und `<key_value>` durch Ihre Servicekonto-Zugangsdaten.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----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
  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

```

▼ Konfiguration mit StoragePools Filter

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----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 des virtuellen Pools

Diese Backend-Konfiguration definiert mehrere virtuelle Pools in einer einzigen Datei. Virtuelle Pools werden im `storage` Abschnitt definiert. Sie sind nützlich, wenn Sie mehrere Speicherpools mit unterschiedlichen Service-Levels haben und entsprechende Speicherklassen in Kubernetes erstellen möchten, die diese repräsentieren. Virtuelle Pool-Labels dienen zur Unterscheidung der Pools. Im folgenden Beispiel werden beispielsweise das `performance` Label und der `serviceLevel` Typ verwendet, um virtuelle Pools zu unterscheiden.

Sie können auch Standardwerte festlegen, die für alle virtuellen Pools gelten, und die Standardwerte einzelner virtueller Pools überschreiben. Im folgenden Beispiel dienen `snapshotReserve` und `exportRule` 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: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----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

Bei der Installation von Trident mit einer GKE Cloud-Identität authentifizieren sich die Backends als Workload-Identität und lassen den `apiKey` Block aus. Die Cloud-Identität wird für beide Google Cloud NetApp Volumes Treiber identisch konfiguriert. Der einzige Unterschied ist der `storageDriverName` Wert: `google-cloud-netapp-volumes` wird für NAS (NFS oder SMB) Volumes verwendet und `google-cloud-netapp-volumes-san` für SAN (iSCSI Block) Volumes.



Cloud-Identitätsunterstützung für den `google-cloud-netapp-volumes-san` Treiber erfordert Trident 26.06 oder höher. SAN-Volumes werden aus Flex Unified Speicherpools bereitgestellt, daher `serviceLevel` auf `flex` setzen.

NAS-Treiber (`google-cloud-netapp-volumes`)

```

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
```

SAN Treiber (`google-cloud-netapp-volumes-san`)

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv-san
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: flex
  storagePool: pool-flex1
```

▼ Unterstützte Topologiekonfiguration

Trident erleichtert die Bereitstellung von Volumes für Workloads basierend auf Regionen und Verfügbarkeitszonen. Der `supportedTopologies` Block in dieser Backend-Konfiguration wird verwendet, um eine Liste von Regionen und Zonen pro Backend bereitzustellen. Die hier angegebenen Regions- und Zonenwerte müssen mit den Regions- und Zonenwerten aus den 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 in einem 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`

Was kommt als Nächstes?

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

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

Um zu überprüfen, ob das Backend erfolgreich erstellt wurde, führen Sie den 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		

Schlägt die Backend-Erstellung fehl, liegt ein Fehler in der Backend-Konfiguration vor. Sie können das Backend mit dem `kubectl get tridentbackendconfig <backend-name>`-Befehl beschreiben oder die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie den folgenden Befehl ausführen:

```
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

Nachfolgend ist 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 des `parameter.selector` Feldes:

Mit `parameter.selector` können Sie für jedes `StorageClass` den "virtueller Pool" angeben, der zum Hosten eines Volumes verwendet wird. Das Volume 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 Details zu Speicherklassen finden Sie unter ["Erstellen Sie eine Speicherklasse"](#).

Beispieldefinitionen für SMB-Volumes

Mit `nasType`, `node-stage-secret-name` und `node-stage-secret-namespace` können Sie ein SMB-Volume angeben und die erforderlichen Active Directory-Anmeldeinformationen bereitstellen. Jeder Active Directory-Benutzername/jedes Passwort mit beliebigen oder keinen Berechtigungen kann für das Knotenstufengeheimnis 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 Secrets pro Volume

```

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` filtert 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 gebunden ist, führen Sie den 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	

Automatisches Tiering für Google Cloud NetApp Volumes konfigurieren

Auto-Tiering wird über Trident-Backend-Parameter und PersistentVolumeClaim-Annotationen während der Volume-Bereitstellung konfiguriert. Sie können Auto-Tiering für Google Cloud NetApp Volumes mit Trident konfigurieren.

Überblick

Auto-Tiering ermöglicht es Trident, Volumes bereitzustellen, die inaktive Daten automatisch von einer Performance-Ebene auf eine Kapazitätsebene verschieben. Dadurch werden die Speicherkosten gesenkt, während die Leistung für häufig abgerufene Daten erhalten bleibt.

Trident wendet die Einstellungen für das automatische Tiering nur bei der Volume-Erstellung an. Änderungen nach der Bereitstellung werden in Trident 26.06 nicht unterstützt.

Konzepte

Automatisches Tiering

Auto-Tiering verschiebt selten genutzte Daten basierend auf den Zugriffsmustern von einer Performance-Tier auf eine Kapazitäts-Tier. Die Datenverschiebung erfolgt asynchron und ist nicht sofort.

Tiering-Richtlinie

Die Tiering-Richtlinie legt fest, ob Auto-Tiering für ein Volume aktiviert ist.

Folgende Richtlinien werden unterstützt: * `auto`: Aktiviert die automatische Stufenaufteilung basierend auf Zugriffsmustern * `none`: Deaktiviert die automatische Stufenaufteilung

Kühltage

Die Kühltage legen die Mindestanzahl an Tagen fest, die ein Datenblock inaktiv bleiben muss, bevor er für das Tiering in Frage kommt. Kühltage gelten nur, wenn die Tiering-Richtlinie auf `auto` eingestellt ist.

Konfigurationsmodell

Konfigurationsbereiche

Auto-Tiering kann auf mehreren Ebenen konfiguriert werden:

- **Geltungsbereich des Speicherpools** Gilt für alle Volumes, die aus dem Pool bereitgestellt wurden.
- **Volumenumfang** Gilt für ein einzelnes Volumen durch PersistentVolumeClaim-Annotationen.

Trident ermittelt die effektive Konfiguration basierend darauf, wo jede Einstellung definiert ist.

Konfigurationspriorität

Wenn dieselbe Einstellung in mehreren Bereichen definiert ist, wendet Trident die folgende Rangfolge an:

1. PersistentVolumeClaim Anmerkungen
2. Trident Backend-Konfiguration
3. Standardeinstellungen für Speicherpools

Einstellungen, die auf einer höheren Ebene definiert sind, überschreiben Werte auf niedrigerer Ebene.

Unterstützte Funktionalität in Trident 26.06

Trident 26.06 unterstützt die folgenden Auto-Tiering-Funktionen für Google Cloud NetApp Volumes:

- Aktivieren oder Deaktivieren des automatischen Tierings während der Volume-Bereitstellung
- Definieren einer Tiering-Richtlinie in der Trident Backend-Konfiguration
- Überschreiben der Tiering-Policy und der Cooling Days pro Volume mithilfe von PVC-Anmerkungen
- Konfiguration von Kühltagen für Volumes mit aktiviertem Auto-Tiering

Nicht unterstützte Funktionalität in Trident 26.06

Die folgenden Operationen werden nicht unterstützt:

- Ändern der Auto-Tiering-Einstellungen nach Volume-Erstellung
- Ändern von Tiering-Richtlinien auf bestehenden Volumes mithilfe von Kubernetes-Updates
- Anwenden von Auto-Tiering-Einstellungen außerhalb von Trident-verwalteten Bereitstellungsworkflows

Backend-Konfigurationsparameter

Die folgenden Parameter steuern das Auto-Tiering-Verhalten, wenn sie in der Trident-Backend-Konfiguration definiert sind:

Parameter	Erforderlich	Beschreibung
tieringPolicy	Nein	Staffelungsrichtlinie für Volumes (<code>auto</code> oder <code>none</code>)
tieringMinimumCoolingDays	Nein	Anzahl der inaktiven Tage, bevor Daten gestaffelt werden (Bereich: 2–183, Standard: 31)

Unterstützte Annotationen

PersistentVolumeClaim-Annotationen ermöglichen die individuelle Überschreibung der Auto-Tiering-Einstellungen für jedes Volume.

Anmerkung	Beschreibung
<code>trident.netapp.io/tieringPolicy</code>	Überschreibt die Tiering-Richtlinie für das Volume
<code>trident.netapp.io/tieringMinimumCoolingDays</code>	Überschreibt den Wert für die Kühltage des Volumes

Beispiel: PersistentVolumeClaim mit automatischen Tiering-Überschreibungen

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: auto-tiering-pvc
  annotations:
    trident.netapp.io/tieringPolicy: auto
    trident.netapp.io/tieringMinimumCoolingDays: "45"
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-cloud-netapp-volumes-auto-tiering
  resources:
    requests:
      storage: 500Gi
```

Verhalten und Einschränkungen

Bereitstellungsverhalten

- Die Einstellungen für die automatische Tierierung werden nur zum Zeitpunkt der Volume-Erstellung ausgewertet und angewendet.
- Trident stimmt die Tiering-Konfiguration nach der Bereitstellung nicht ab.
- Kühltage werden ignoriert, wenn die Stufenregelung auf `none` eingestellt ist.

Plattformbeschränkungen

- Auto-tiering wird nur für NAS-Volumes (NFS und SMB) unterstützt.
- Block-Volumes (iSCSI) unterstützen kein Auto-Tiering.
- Der Google Cloud NetApp Volumes-Speicherpool muss das automatische Tiering in Google Cloud aktiviert haben.

Unterstützte Werte

- Gültiger Bereich für `tieringMinimumCoolingDays`: 2 bis 183
- Standardwert: 31

Konfiguration von Google Cloud NetApp Volumes ONTAP-mode Speicherpools

Ab Trident 26.06 können bestehende `ontap-san` und `ontap-nas` Backends so konfiguriert werden, dass sie Speicher auf Google Cloud NetApp Volumes (GCNV) ONTAP-mode Speicherpools bereitstellen. Diese Funktion ist eine Tech Preview.

HINWEIS

Diese Funktion ist eine Tech Preview in Trident 26.06. Tech Preview Funktionen sind für den Produktiveinsatz nicht unterstützt. Die Funktionalität, die Konfigurationsfelder und die unterstützte Matrix können sich vor der allgemeinen Verfügbarkeit (GA) ändern.

So funktioniert es

Trident nutzt die bestehende `ontap-san` und `ontap-nas` Treiberlogik wieder und leitet ONTAP REST-Aufrufe über einen GCNV-Proxy-Endpunkt weiter. Mit diesem Ansatz ist die Bereitstellung und Verwaltung von Speicher auf GCNV-gehosteten ONTAP Clustern möglich, wobei die gleichen Betriebsabläufe wie bei einem direkten ONTAP Backend erhalten bleiben. ONTAP-mode führt keinen neuen Speichertreiber ein. Die Aktivierung erfolgt pro Backend durch Hinzufügen eines `gcnv` Konfigurationsblocks zu einem `ontap-san` oder `ontap-nas` Backend.

Der Bereitstellungspfad ist wie folgt:

PersistentVolumeClaim → Trident (`ontap-san` oder `ontap-nas`) → ONTAP REST Client → GCNV Proxy → ONTAP Cluster im GCNV Pool

Unterstützter und nicht unterstützter Scope

Diese technische Vorschau unterstützt Folgendes:

- Treiber: `ontap-san` (iSCSI) und `ontap-nas` (NFS oder SMB).
- Lebenszyklusoperationen über den Proxy: Backend-Initialisierung, Volume-Erstellung und -Löschung sowie zugehörige ONTAP REST-Flows.
- Authentifizierungsmodelle: Workload Identity Pool (WIP), Dienstkontoschlüssel und Application Default Credentials (ADC) Fallback.

Die folgenden Punkte sind nicht Teil dieser Tech Preview:

- Die `ontap-san-economy`, `ontap-nas-economy` und `ontap-nas-flexgroup`-Treiber.
- Der ASA r2 Persönlichkeitspfad.
- ZAPI Fallback. ONTAP-Modus verwendet ausschließlich ONTAP REST.

Voraussetzungen

Bevor Sie ein ONTAP Mode-Backend konfigurieren, ist Folgendes erforderlich:

- Trident 26.06 oder später.

- Ein GCNV ONTAP-Mode-Speicherpool in Ihrem Zielprojekt und Standort.
- GCP IAM-Berechtigungen für Proxy-Operationen, passend für Ihre Umgebung festgelegt.
- Eine Backend-Konfiguration, die `proxyURL`, `projectNumber`, `location` und `poolID` umfasst.
- Für iSCSI SAN-Workflows gelten die iSCSI- und Multipath-Voraussetzungen auf Knotenseite für Ihre Plattform. Weitere Informationen enthält ["Bereiten Sie den Worker-Knoten vor"](#).

Das Backend konfigurieren

A `TridentBackendConfig` erfordert `credentials`. Für ONTAP-mode enthält dieses Secret die GCP Proxy-Anmeldeinformationen, nicht ONTAP `managementLIF/username/password`. `svm` ist optional: Jeder GCNV ONTAP-mode Pool verfügt derzeit über eine SVM, und Trident leitet sie aus dem Pool ab, wenn sie weggelassen wird.

Dasselbe Dienstkontomuster wie bei nativ GCNV verwenden: nicht sensible Felder in `gcnv.apiKey`, `private_key` und `private_key_id` im Geheimnis. Weitere Informationen in ["Beispiele für die Konfiguration von Google Cloud NetApp Volumes"](#).

ontap-san Backend

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-san
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san
  backendName: gcnv-ontap-san
  credentials:
    name: gcnv-sa-secret
    type: secret
  gcnv:
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
    svm: "<svm-name>"
```

ontap-nas Backend

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-nas
  namespace: trident
spec:
  version: 1
```

```

storageDriverName: ontap-nas
backendName: gcnv-ontap-nas
credentials:
  name: gcnv-sa-secret
  type: secret
gcnv:
  proxyURL: "https://netapp.googleapis.com"
  projectNumber: "<project-number>"
  location: "<region-or-zone>"
  poolID: "<pool-id>"
  apiKey:
    type: service_account
    project_id: "<project-id>"
    client_email: "<service-account-email>"
    client_id: "<client-id>"
    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: "<client-x509-cert-url>"

```

Geheimnis

```

apiVersion: v1
kind: Secret
metadata:
  name: gcnv-sa-secret
  namespace: trident
type: Opaque
stringData:
  private_key_id: "<private-key-id>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <private-key>
    -----END PRIVATE KEY-----

```

HINWEIS

Setzen Sie `private_key` oder `private_key_id` nicht in die `TridentBackendConfig`-Spezifikation, da die CRD sie ablehnt. `ontap-san` sollte mit demselben `gcnv`- und Anmeldeinformationsmuster für Blockspeicher verwendet werden.

```
svm: "<svm-name>"
```

```
== Authentication
```

Trident resolves credentials for proxy access in the following order:

- . Workload Identity Pool (WIP)
- . Service account key
- . Application Default Credentials (ADC)

For this Tech Preview, use the following practices:

- * Prefer Workload Identity Pool where it is available.
- * Do not embed raw private keys in version-controlled backend files. Store credentials in a Kubernetes secret.
- * Apply least-privilege IAM scoping to the service account.

== Validation and initialization behavior

Trident validates an ONTAP-mode backend during initialization and fails fast rather than partially initializing.

Backend initialization fails if a required `gcnv` field is missing, the driver is unsupported for ONTAP-mode, or credential resolution fails.

== Known limitations

- * This feature is a Tech Preview. Behavior and the supported matrix can change before GA.
- * ZAPI fallback is not used in ONTAP-mode.
- * Protocol and node-readiness requirements still apply. For example, iSCSI workflows require the node-side prerequisites described in [xref:{relative_path}../trident-use/worker-node-prep.html](#) [Prepare the worker node].
- * Existing ONTAP and GCNV environment constraints still apply.

== Upgrade and compatibility

ONTAP-mode is opt-in per backend through the `gcnv` configuration block:

- * Existing non-GCNV ONTAP backends are unaffected.
- * Mixed deployments that combine direct ONTAP backends and ONTAP-mode proxy backends are supported through backend-level configuration. Each backend is configured independently.
- * To stop using ONTAP-mode, remove or replace the affected backends. No global switch is required.

== What's next?

Apply secret before TBC; use `kubectl apply` and `-n trident` to match

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 stellt den `solidfire-san` Storage-Treiber zur Kommunikation mit dem Cluster bereit. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

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

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

Bevor Sie beginnen

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

- Ein unterstütztes Speichersystem, das Element-Software ausführt.
- 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. Weitere Informationen enthält ["Informationen zur Vorbereitung der Worker-Knoten"](#).

Backend-Konfigurationsoptionen

Siehe die folgende Tabelle für die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
<code>version</code>		Immer 1
<code>storageDriverName</code>	Name des Speichertreibers	Immer „solidfire-san“
<code>backendName</code>	Benutzerdefinierter Name oder das Speicher-Backend	"solidfire_" + storage (iSCSI) IP-Adresse
<code>Endpoint</code>	MVIP für den SolidFire Cluster mit Mandantenanmeldeinformationen	
<code>SVIP</code>	Speicher (iSCSI) IP-Adresse und Port	

Parameter	Beschreibung	Standard
labels	Satz beliebiger, im JSON-Format vorliegender Bezeichnungen, die auf Volumes angewendet werden sollen.	""
TenantName	Zu verwendender Mandantennamen (wird erstellt, falls nicht gefunden)	
InitiatorIFace	Beschränken Sie den iSCSI-Datenverkehr auf eine bestimmte Host-Schnittstelle	"default"
UseCHAP	Verwenden Sie CHAP zur Authentifizierung von iSCSI. Trident verwendet CHAP.	true
AccessGroups	Liste der zu verwendenden Access Group-IDs	Findet die ID einer Zugriffsgruppe namens "trident"
Types	QoS-Spezifikationen	
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe über diesem Wert liegt	"" (wird nicht standardmäßig erzwungen)
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, {"api":false, "method":true}	null

WARNUNG

Verwenden Sie `debugTraceFlags` nicht, es sei denn, Sie führen eine Fehlerbehebung durch und benötigen einen detaillierten Protokollauszug.

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

Dieses Beispiel zeigt eine Backend-Datei mit CHAP-Authentifizierung und Modellierung von drei Volumentypen mit spezifischen QoS-Garantien. Wahrscheinlich würden Sie anschließend Speicherklassen definieren, um jede dieser mithilfe des `IOPS storage class`-Parameters zu nutzen.

```

---
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`-Treiber mit virtuellen Pools

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

Trident kopiert die im Speicherpool vorhandenen Labels beim Provisionierungsvorgang 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 für alle Speicherpools spezifische Standardwerte festgelegt, die die `type` auf Silver setzen. Die virtuellen Pools sind im `storage` Abschnitt definiert. 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. Mit dem `parameters.selector`-Feld gibt jede StorageClass an, welche virtuellen Pools zum Hosten eines Volumes verwendet werden können. Das Volume weist die im gewählten virtuellen Pool definierten Aspekte auf.

Die erste StorageClass (`solidfire-gold-four`) wird dem ersten virtuellen Pool zugeordnet. Dies ist der einzige Pool, der Gold-Performance mit einer Volume Type QoS von Gold bietet. Die letzte StorageClass (`solidfire-silver`) bezieht sich auf jeden Speicherpool, der eine Silber-Performance bietet. Trident entscheidet, welcher virtuelle Pool ausgewählt wird und stellt sicher, dass die Speicheranforderung erfüllt wird.

```

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 für die Kommunikation mit dem ONTAP Cluster bereit. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	volumeMode	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
ontap-san	iSCSI SCSI über FC	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem; Raw-Blockgerät
ontap-san	iSCSI SCSI über FC	Dateisystem	RWO, RWOP ROX und RWX sind im Filesystem-Volume-Modus nicht verfügbar.	xfs, ext3, ext4
ontap-san	NVMe/TCP Weitere Informationen enthält Weitere Überlegungen zu NVMe/TCP .	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem; Raw-Blockgerät
ontap-san	NVMe/TCP Weitere Informationen enthält Weitere Überlegungen zu NVMe/TCP .	Dateisystem	RWO, RWOP ROX und RWX sind im Filesystem-Volume-Modus nicht verfügbar.	xfs, ext3, ext4
ontap-san-economy	iSCSI	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem; Raw-Blockgerät
ontap-san-economy	iSCSI	Dateisystem	RWO, RWOP ROX und RWX sind im Filesystem-Volume-Modus nicht verfügbar.	xfs, ext3, ext4

WARNUNG

- Verwenden Sie `ontap-san-economy` nur, wenn die Anzahl der persistenten Speichernutzungen voraussichtlich höher ist als "[Unterstützte ONTAP Volume-Grenzwerte](#)".

- Verwenden Sie `ontap-nas-economy` nur, wenn die Anzahl der persistenten Speichernutzungen voraussichtlich höher ist als "[Unterstützte ONTAP Volume-Grenzwerte](#)" und der `ontap-san-economy` Treiber nicht verwendet werden kann.
- Verwenden Sie `ontap-nas-economy` nicht, wenn Sie einen Bedarf an Datenschutz, Notfallwiederherstellung oder Mobilität erwarten.
- NetApp empfiehlt nicht, Flexvol autogrow in allen ONTAP-Treibern zu verwenden, außer `ontap-san`. Als Workaround unterstützt Trident die Verwendung von Snapshot-Reserve und skaliert Flexvol-Volumes entsprechend.

Benutzerberechtigungen

Trident wird entweder als ONTAP- oder SVM-Administrator ausgeführt, typischerweise mit dem `admin` cluster-Benutzer oder einem `vsadmin` SVM-Benutzer oder einem Benutzer mit anderem Namen, der die gleiche Rolle hat. Für Amazon FSx for NetApp ONTAP-Bereitstellungen wird Trident entweder als ONTAP- oder SVM-Administrator ausgeführt, mit dem `cluster fsxadmin` Benutzer oder einem `vsadmin` SVM-Benutzer oder einem Benutzer mit anderem Namen, der die gleiche Rolle hat. Der `fsxadmin` Benutzer ist ein eingeschränkter Ersatz für den cluster-Admin-Benutzer.

HINWEIS

Wenn Sie den `limitAggregateUsage` Parameter verwenden, sind Cluster-Administratorrechte erforderlich. Bei Verwendung von Amazon FSx für NetApp ONTAP mit Trident wird der `limitAggregateUsage` Parameter nicht mit den `vsadmin` und `fsxadmin` Benutzerkonten funktionieren. Der Konfigurationsvorgang schlägt fehl, wenn Sie diesen Parameter angeben.

Zwar ist es möglich, in ONTAP eine restriktivere Rolle zu erstellen, die ein Trident-Treiber verwenden kann, wir raten jedoch davon ab. Die meisten neuen Versionen von Trident rufen zusätzliche APIs auf, die berücksichtigt werden müssten, was Aktualisierungen erschwert und fehleranfällig macht.

Weitere Überlegungen zu NVMe/TCP

Trident unterstützt das Non-Volatile Memory Express (NVMe)-Protokoll mit dem `ontap-san` Treiber, 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-native Multipathing
- Geordnetes oder ungeordnetes Herunterfahren der K8s-Knoten (24.06)

Trident unterstützt nicht:

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

HINWEIS

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 erfordert Trident, dass mindestens ein Aggregat dem SVM zugewiesen wird.

HINWEIS

"[ASA r2-Systeme](#)" 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 "[dies](#)" Knowledge Base Artikel zur Zuweisung von Aggregaten zu SVMs in ASA r2-Systemen.

Beachten Sie, dass Sie auch mehrere Treiber gleichzeitig ausführen und Speicherklassen erstellen können, die auf den einen oder den anderen verweisen. Beispielsweise könnten Sie eine `san-dev` Klasse konfigurieren, die den `ontap-san` Treiber verwendet, und eine `san-default` Klasse, die den `ontap-san-economy` verwendet.

Auf allen Ihren Kubernetes-Worker-Knoten müssen die entsprechenden iSCSI-Tools installiert sein. Details sind in "[Bereiten Sie den Worker-Knoten vor](#)" zu finden.

Authentifizieren Sie das ONTAP Backend

Trident bietet zwei Modi zur Authentifizierung eines ONTAP Backend.

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

Sie können bestehende Backends aktualisieren, um zwischen anmeldeinformationsbasierten und zertifikatsbasierten Authentifizierungsverfahren zu wechseln. Es wird jedoch jeweils nur ein Authentifizierungsverfahren unterstützt. Um zu einem anderen Authentifizierungsverfahren zu wechseln, müssen Sie das bestehende Verfahren aus der Backend-Konfiguration entfernen.

WARNUNG

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

Aktivieren Sie die anmeldeinformationsbasierte Authentifizierung

Trident benötigt die Anmeldeinformationen eines Administrators mit SVM- oder Cluster-Berechtigungen, um mit dem ONTAP Backend zu kommunizieren. Es wird empfohlen, vordefinierte Standardrollen wie `admin` oder `vsadmin` zu verwenden. Dies gewährleistet die Vorwärtskompatibilität mit zukünftigen ONTAP Versionen, die möglicherweise Feature-APIs für zukünftige Trident Versionen bereitstellen. Eine benutzerdefinierte Sicherheits-Anmelderolle kann erstellt und mit Trident verwendet werden, 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 Zugangsdaten im Klartext gespeichert werden. Nach der Backend-Erstellung werden Benutzernamen/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 vom Kubernetes-/Speicheradministrator durchzuführende Operation.

Aktivieren Sie die zertifikatsbasierte Authentifizierung

Neue und bestehende Backends können ein Zertifikat verwenden und mit dem ONTAP Backend kommunizieren. In der Backend-Definition sind drei Parameter erforderlich.

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

Ein typischer Arbeitsablauf umfasst die folgenden Schritte.

Schritte

1. Generieren Sie ein Clientzertifikat und einen Schlüssel. Legen Sie beim Generieren den Common Name (CN) auf den ONTAP Benutzer fest, als den Sie sich authentifizieren möchten.

```
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 wurde möglicherweise bereits vom Storage-Administrator erledigt. Ignorieren Sie dies, falls keine vertrauenswürdige CA 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
```

HINWEIS

Nach Ausführung dieses Befehls fordert ONTAP die Eingabe des Zertifikats an. Fügen Sie den Inhalt der `k8senv.pem` Datei ein, die in Schritt 1 generiert wurde, und drücken Sie `END`, um die Installation abzuschließen.

4. Bestätigen Sie, dass die ONTAP Sicherheitsanmeldungsrolle das `cert` Authentifizierungsverfahren unterstützt.

```
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 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 das 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...Vaaallllluuuueeee",
  "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 Authentifizierungsverfahren oder rotieren Sie die Anmeldeinformationen

Sie können ein bestehendes Backend aktualisieren, um ein anderes Authentifizierungsverfahren zu verwenden oder um die Anmeldeinformationen zu rotieren. Dies funktioniert in beide Richtungen: Backends, die Benutzernamen/Passwörter verwenden, können auf Zertifikate umgestellt werden; Backends, die Zertifikate verwenden, können auf Benutzernamen/Passwörter umgestellt werden. Dazu müssen Sie das bestehende Authentifizierungsverfahren entfernen und das neue Authentifizierungsverfahren hinzufügen. Verwenden Sie dann die aktualisierte backend.json-Datei mit den erforderlichen Parametern, um `tridentctl backend update` auszuführen.

```
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 |
+-----+-----+-----+
+-----+-----+

```

HINWEIS

Beim Rotieren von Passwörtern muss der Speicheradministrator zunächst das Passwort für den Benutzer auf ONTAP aktualisieren. Danach erfolgt ein Backend-Update. Beim Rotieren von Zertifikaten können dem Benutzer mehrere Zertifikate hinzugefügt werden. Das Backend wird dann aktualisiert, um das neue Zertifikat zu verwenden, wonach das alte Zertifikat aus dem ONTAP Cluster gelöscht werden kann.

Die Aktualisierung des Backends beeinträchtigt weder den Zugriff auf bereits erstellte Volumes noch später hergestellte Volume-Verbindungen. Eine erfolgreiche Backend-Aktualisierung zeigt an, dass Trident mit dem ONTAP Backend kommunizieren und zukünftige Volume-Operationen ausführen kann.

Erstellen einer benutzerdefinierten ONTAP Rolle für Trident

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

Weitere Informationen zum Erstellen benutzerdefinierter Trident-Rollen finden Sie unter ["Trident Custom-Role-Generator"](#).

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\>
```

Verwenden von System Manager

Führen Sie die folgenden Schritte im ONTAP System Manager aus:

1. Erstellen Sie eine benutzerdefinierte Rolle:

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

(Oder) Um eine benutzerdefinierte Rolle auf SVM-Ebene zu erstellen, wählen Sie **Storage > Storage VMs > required svm > Settings > Users and Roles**.

- Wählen Sie das Pfeilsymbol (→) neben **Users and Roles** aus.
- Wählen Sie unter **Rollen +Add** aus.
- Definieren Sie die Regeln für die Rolle und klicken Sie auf **Save**.

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

- Wählen Sie das Symbol + unter **Benutzer** aus.
- Wählen Sie den gewünschten Benutzernamen aus und wählen Sie eine Rolle im Dropdown-Menü für **Rolle**.
- Auf **Speichern** klicken.

Weitere Informationen finden Sie auf den folgenden Seiten:

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

Verbindungen mit bidirektionalem CHAP authentifizieren

Trident kann iSCSI-Sitzungen mit bidirektionalem CHAP für die `ontap-san` und `ontap-san-economy` Treiber authentifizieren. Dazu muss die `useCHAP` Option in Ihrer Backend-Definition aktiviert werden. Wenn auf `true` gesetzt, konfiguriert Trident die Standard-Initiator-Sicherheit der SVM auf bidirektionales CHAP und legt den Benutzernamen sowie 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
```

WARNUNG

Der `useCHAP` Parameter ist eine boolesche Option, die nur einmal konfiguriert werden kann. Standardmäßig ist sie auf `false` gesetzt. Nachdem Sie sie auf `true` gesetzt haben, können Sie sie nicht mehr auf `false` zurücksetzen.

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

So funktioniert es

Durch das Setzen von `useCHAP` auf `true` weist der Speicheradministrator Trident an, CHAP auf dem Storage-Backend zu konfigurieren. Dies umfasst Folgendes:

- CHAP auf der SVM einrichten:
 - Wenn der Standard-Initiator-Sicherheitstyp der SVM „none“ ist (Standardeinstellung) **und** keine bereits vorhandenen LUNs im Volume vorhanden sind, setzt Trident den Standard-Sicherheitstyp auf `CHAP` und fährt mit der Konfiguration des CHAP-Initiators sowie des Zielbenutzernamens und der Geheimnisse fort.
 - Wenn die SVM LUNs enthält, aktiviert Trident CHAP nicht auf der SVM. Dadurch wird sichergestellt, dass der Zugriff auf bereits vorhandene LUNs auf der SVM 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 eine entsprechende `tridentbackend` CRD und speichert die CHAP-Secrets und Benutzernamen als Kubernetes-Secrets. Alle von Trident auf diesem Backend erstellten PVs werden über CHAP eingebunden und verbunden.

Anmeldeinformationen rotieren und Backends aktualisieren

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

WARNUNG

Beim Aktualisieren der CHAP-Geheimnisse für ein Backend müssen Sie `tridentctl` verwenden, um das Backend zu aktualisieren. Aktualisieren Sie die Anmeldeinformationen auf dem Storage-Cluster nicht mit der ONTAP CLI oder dem ONTAP System Manager, da Trident diese Änderungen nicht übernehmen 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-4378-8a33-1a472ffbe5c |
online | 7 |
+-----+-----+-----+-----+
+-----+-----+
```

Bestehende Verbindungen bleiben unberührt; sie bleiben aktiv, wenn die Anmeldeinformationen von Trident auf der SVM aktualisiert werden. Neue Verbindungen verwenden die aktualisierten Anmeldeinformationen und bestehende Verbindungen bleiben weiterhin aktiv. Das Trennen und erneute Verbinden alter PVs führt dazu, dass sie die aktualisierten Anmeldeinformationen verwenden.

ONTAP SAN-Konfigurationsoptionen und Beispiele

ONTAP SAN-Treiber können mit Ihrer Trident Installation erstellt und verwendet werden. Dieser Abschnitt enthält Beispiele für die Backend-Konfiguration und Details zur Zuordnung von Backends zu StorageClasses. ["ASA r2-Systeme"](#) unterscheiden sich von anderen ONTAP Systemen (ASA, AFF und FAS) in der Implementierung ihrer Speicherschicht. Diese Unterschiede wirken sich auf die Verwendung bestimmter Parameter aus, wie angegeben. ["Erfahren Sie mehr über die Unterschiede zwischen ASA r2-Systemen und anderen ONTAP Systemen"](#). In der Trident Backend-Konfiguration muss nicht angegeben werden, dass das System ASA r2 ist. Wenn `ontap-san` als `storageDriverName` ausgewählt wird, erkennt Trident automatisch ASA r2 oder andere ONTAP Systeme. Einige Backend-Konfigurationsparameter sind für ASA r2

Systeme nicht anwendbar, wie in der folgenden Tabelle aufgeführt.

HINWEIS

Nur der `ontap-san` driver (mit iSCSI-, NVMe/TCP- und FC-Protokollen) wird für ASA r2-Systeme unterstützt.

Backend-Konfigurationsoptionen

Ab der Version 26.06 kann die ONTAP Balanced Placement-Funktion aktiviert werden, sodass ONTAP das Aggregat für neue Volumes auswählt. Siehe ["Ausgewogene Platzierung mit ONTAP Backends verwenden"](#) für Anforderungen und Beispiele.

Siehe die folgende Tabelle für die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
<code>version</code>		Immer 1
<code>storageDriverName</code>	Name des Speichertreibers. Kann nach der Erstellung nicht mehr geändert werden. Zum Aktualisieren dieses Parameters ist die Erstellung eines neuen Backends erforderlich.	<code>ontap-san</code> oder <code>ontap-san-economy</code>
<code>backendName</code>	Benutzerdefinierter Name oder das Speicher-Backend	Treibername + "_" + dataLIF
<code>managementLIF</code>	<p>IP-Adresse einer Cluster- oder SVM-Management-LIF.</p> <p>Es kann ein vollqualifizierter Domain-Name (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 angegeben werden, wie <code>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</code>.</p> <p>Für einen reibungslosen MetroCluster-Switchover siehe MetroCluster Beispiel.</p> <div>HINWEIS Wenn Sie „vsadmin“-Anmeldeinformationen verwenden, <code>managementLIF</code> muss dies die der SVM sein; wenn Sie „admin“-Anmeldeinformationen verwenden, <code>managementLIF</code> muss dies die des Clusters sein.</div>	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parameter	Beschreibung	Standard
dataLIF	IP-Adresse des Protokoll-LIF. 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, wie <code>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</code> . Nicht für iSCSI angeben. Trident verwendet "ONTAP Selective LUN Map" zur Ermittlung der iSCSI-LIFs, die für den Aufbau einer Multipath-Sitzung benötigt werden. Eine Warnung wird generiert, wenn dataLIF explizit definiert ist. Für MetroCluster weglassen. Siehe MetroCluster Beispiel .	Abgeleitet von der SVM
svm	Zu verwendende virtuelle Speichermaschine Für MetroCluster auslassen. Siehe MetroCluster Beispiel .	Wird abgeleitet, wenn eine SVM managementLIF angegeben ist
useCHAP	Verwenden Sie CHAP zur Authentifizierung von iSCSI für ONTAP SAN-Treiber [Boolescher Wert]. Setzen Sie auf <code>true</code> , damit Trident bidirektionales CHAP als Standardauthentifizierung für die im Backend angegebene SVM konfiguriert und verwendet. Details sind in "Bereiten Sie die Konfiguration des Backends mit ONTAP SAN-Treibern vor" zu finden. Nicht unterstützt für FCP oder NVMe/TCP.	false
chapInitiatorSecret	CHAP-Initiatorgeheimnis. Erforderlich, wenn <code>useCHAP=true</code>	""
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Volumes angewendet werden sollen	""
chapTargetInitiatorSecret	CHAP-Zielinitiatorgeheimnis. Erforderlich, wenn <code>useCHAP=true</code>	""
chapUsername	Benutzername für eingehende Anfragen. Erforderlich, wenn <code>useCHAP=true</code>	""
chapTargetUsername	Zielbenutzername. Erforderlich, wenn <code>useCHAP=true</code>	""
clientCertificate	Base64-kodierter Wert des Clientzertifikats. Wird für zertifikatbasierte Authentifizierung verwendet	""
clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für zertifikatbasierte Authentifizierung verwendet	""
trustedCACertificate	Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Optional. Wird für zertifikatbasierte Authentifizierung verwendet.	""

Parameter	Beschreibung	Standard
username	Benutzername, der für die Kommunikation mit dem ONTAP Cluster benötigt wird. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet. Für die Active Directory Authentifizierung siehe "Authentifizieren Sie Trident bei einer Backend-SVM mit Active Directory-Anmeldeinformationen" .	""
password	Passwort erforderlich, um mit dem ONTAP Cluster zu kommunizieren. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet. Für die Active Directory Authentifizierung siehe "Authentifizieren Sie Trident bei einer Backend-SVM mit Active Directory-Anmeldeinformationen" .	""
svm	Zu verwendende Storage Virtual Machine	Wird abgeleitet, wenn eine SVM managementLIF angegeben ist
storagePrefix	Präfix, das beim Bereitstellen neuer Volumes in der SVM verwendet wird. Kann später nicht mehr geändert werden. Zum Aktualisieren dieses Parameters ist die Erstellung eines neuen Backends erforderlich.	trident

Parameter	Beschreibung	Standard
aggregate	<p>Aggregat für die Bereitstellung (optional; falls festgelegt, muss es der SVM zugewiesen werden). Für den <code>ontap-nas-flexgroup</code> Treiber wird diese Option ignoriert. Wenn kein Aggregat zugewiesen ist, kann jedes der verfügbaren Aggregate zur Bereitstellung eines FlexGroup Volumes verwendet werden.</p> <div> <div>HINWEIS</div> <p>Wird das Aggregat in SVM aktualisiert, erfolgt die Aktualisierung in Trident automatisch durch regelmäßiges Abfragen von SVM, ohne dass der Trident Controller neu gestartet werden muss. Wenn in Trident ein bestimmtes Aggregat für die Volume-Bereitstellung konfiguriert wurde und dieses Aggregat umbenannt oder aus der SVM entfernt wird, wechselt das Backend in Trident beim Abfragen des SVM-Aggregats in den Fehlerzustand. Das Aggregat muss entweder auf eines geändert werden, das in der SVM vorhanden ist, oder vollständig entfernt werden, damit das Backend wieder online ist.</p> </div> <p>Für ASA r2 Systeme nicht angeben.</p> <p>Nicht angeben, wenn <code>useBalancedPlacement</code> auf <code>true</code> gesetzt ist.</p>	""
useBalancedPlacement	<p>Aggregatauswahl für neue Volumes an ONTAP [Boolescher Wert]. Erfordert ONTAP 9.19.1 oder höher und <code>useREST</code> auf <code>true</code> gesetzt. Kann nicht mit <code>aggregate</code> kombiniert werden.</p> <p>Siehe "Ausgewogene Platzierung mit ONTAP Backends verwenden" für Anforderungen und Beispiele.</p>	false
limitAggregateUsage	<p>Die Bereitstellung schlägt fehl, wenn die Nutzung diesen Prozentsatz überschreitet. Wenn Sie ein Amazon FSx for NetApp ONTAP-Backend verwenden, geben Sie <code>limitAggregateUsage</code> nicht an. Die bereitgestellten <code>fsxadmin</code> und <code>vsadmin</code> enthalten nicht die erforderlichen Berechtigungen, um die aggregierte Nutzung abzurufen und sie mit Trident zu begrenzen. Für ASA r2 Systeme nicht angeben.</p>	"" (wird nicht standardmäßig erzwungen)

Parameter	Beschreibung	Standard
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet. Außerdem wird die maximale Größe der von ihr für LUNs verwalteten Volumes beschränkt.	"" (wird nicht standardmäßig erzwungen)
lunsPerFlexvol	Maximale LUNs pro FlexVol, muss im Bereich [50, 200] liegen	100
debugTraceFlags	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, {"api":false, "method":true} nicht verwenden, es sei denn, Sie führen eine Fehlersuche durch und benötigen eine detaillierte Protokollausgabe.	null
useREST	<p>Boolescher Parameter zur Verwendung von ONTAP REST APIs.</p> <div> <p><code>`useREST`</code> Wenn auf <code>`true`</code> gesetzt, verwendet Trident ONTAP REST APIs zur Kommunikation mit dem Backend; wenn auf <code>`false`</code> gesetzt, verwendet Trident ONTAPI (ZAPI)-Aufrufe zur Kommunikation mit dem Backend. Diese Funktion erfordert ONTAP 9.11.1 und höher. Zusätzlich muss die verwendete ONTAP-Anmelderolle Zugriff auf die <code>`ontapi`</code> Anwendung haben. Dies wird durch die vordefinierten <code>`vsadmin`</code> und <code>`cluster-admin`</code> Rollen erfüllt. Ab der Trident 24.06-Version und ONTAP 9.15.1 oder höher ist <code>`useREST`</code> standardmäßig auf <code>`true`</code> gesetzt; ändern Sie <code>`useREST`</code> auf <code>`false`</code>, um ONTAPI (ZAPI)-Aufrufe zu verwenden.</p> </div> <p>useREST ist vollqualifiziert für NVMe/TCP.</p> <div> <p>HINWEIS</p> <p>NVMe wird nur mit ONTAP REST APIs unterstützt und nicht mit ONTAPI (ZAPI).</p> </div> <p>Falls angegeben, immer auf true für ASA r2-Systeme setzen.</p>	true für ONTAP 9.15.1 oder höher, andernfalls false.

Parameter	Beschreibung	Standard
sanType	Verwenden Sie zum Auswählen <code>iscsi</code> für iSCSI, <code>nvme</code> für NVMe/TCP oder <code>fc</code> für SCSI über Fibre Channel (FC).	<code>iscsi</code> falls leer
formatOptions	Verwenden Sie <code>formatOptions</code> zur Angabe von Befehlszeilenargumenten für den <code>mkfs</code> Befehl, der immer dann angewendet wird, wenn ein Volume formatiert wird. Dadurch kann das Volume entsprechend den eigenen Präferenzen formatiert werden. Es ist darauf zu achten, dass <code>formatOptions</code> ähnlich wie die Optionen des Befehls <code>mkfs</code> angegeben werden, jedoch ohne den Gerätepfad. Beispiel: <code>"-E nodiscard"</code> Unterstützt für <code>ontap-san</code> und <code>ontap-san-economy</code> Treiber mit iSCSI-Protokoll. Zusätzlich unterstützt für ASA r2 Systeme bei Verwendung der Protokolle iSCSI und NVMe/TCP.	
limitVolumePoolSize	Maximal anforderbare FlexVol-Größe bei Verwendung von LUNs im <code>ontap-san-economy</code> Backend.	<code>""</code> (wird nicht standardmäßig erzwungen)
denyNewVolumePools	Beschränkt <code>ontap-san-economy</code> Backends darauf, neue FlexVol Volumes zur Aufnahme ihrer LUNs zu erstellen. Für die Bereitstellung neuer PVs werden ausschließlich bereits vorhandene FlexVols verwendet.	

Empfehlungen zur Verwendung von `formatOptions`

Trident empfiehlt die folgenden Optionen, um den Formatierungsprozess zu beschleunigen:

- **-E nodiscard (ext3, ext4):** Versuche nicht, Blöcke während der Erstellung des Dateisystems (`mkfs`) zu verwerfen (das anfängliche Verwerfen von Blöcken ist auf Solid-State-Geräten und dünnbesetzten/dünnprovisionierten Speichern sinnvoll). Dies ersetzt die veraltete Option `"-K"` und ist für `ext3`- und `ext4`-Dateisysteme anwendbar.
- **-K (xfs):** Versuche nicht, Blöcke beim `mkfs` zu verwerfen. Diese Option ist für das `xfs`-Dateisystem anwendbar.

Authentifizieren Sie Trident bei einer Backend-SVM mit Active Directory-Anmeldeinformationen

Sie können Trident so konfigurieren, dass es sich mit Active Directory (AD)-Anmeldeinformationen an einer Backend-SVM authentifiziert. Bevor ein AD-Konto auf die SVM zugreifen kann, müssen Sie den Zugriff des AD-Domänencontrollers auf den Cluster oder die SVM konfigurieren. Für die Clusterverwaltung mit einem AD-Konto müssen Sie einen Domämentunnel erstellen. Details sind in ["Konfigurieren des Zugriffs auf Active Directory-Domänencontroller in ONTAP"](#) zu finden.

Schritte

1. Konfigurieren Sie die Domain Name System (DNS)-Einstellungen für eine Backend-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 Gruppe zu erstellen, die das Cluster oder die SVM verwalten.

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

4. In der Trident-Backend-Konfigurationsdatei setzen Sie die `username` und `password` Parameter auf den AD-Benutzer- bzw. Gruppennamen und das Passwort.

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

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

Parameter	Beschreibung	Standard
<code>spaceAllocation</code>	Speicherplatzzuweisung für LUNs	"true" Falls angegeben, auf true für ASA r2-Systeme setzen.
<code>spaceReserve</code>	Speicherplatzreservierungsmodus: „none“ (dünn) oder „volume“ (dick). Auf none für ASA r2-Systeme einstellen.	"none"
<code>snapshotPolicy</code>	Zu verwendende Snapshot-Richtlinie. Für none ASA r2-Systeme festlegen.	"none"
<code>qosPolicy</code>	QoS-Richtliniengruppe, die für erstellte Volumes zugewiesen werden soll. Wählen Sie eine von <code>qosPolicy</code> oder <code>adaptiveQosPolicy</code> pro Storage-Pool/Backend. 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 jedes einzelne Element angewendet wird. Eine gemeinsam genutzte QoS-Richtliniengruppe erzwingt die Obergrenze für den Gesamtdurchsatz aller Workloads.	""
<code>adaptiveQosPolicy</code>	Adaptive QoS-Richtliniengruppe zur Zuweisung für erstellte Volumes. Wählen Sie eine der <code>qosPolicy</code> oder <code>adaptiveQosPolicy</code> pro Speicherpool/Backend aus.	""
<code>snapshotReserve</code>	Prozentsatz des für Snapshots reservierten Speichervolumens. Für ASA r2 Systeme nicht angeben.	"0", falls <code>`snapshotPolicy`</code> "none", andernfalls ""
<code>splitOnClone</code>	Trennen Sie einen Klon bei seiner Erstellung von seinem übergeordneten Objekt	"false"

Parameter	Beschreibung	Standard
encryption	Aktivieren Sie NetApp Volume Encryption (NVE) auf dem neuen Volume; standardmäßig <code>false</code> . NVE muss auf dem Cluster lizenziert und aktiviert sein, damit diese Option verwendet werden kann. Wenn NAE im Backend aktiviert ist, ist jedes in Trident bereitgestellte Volume NAE-aktiviert. Weitere Informationen unter: "Wie Trident mit NVE und NAE zusammenarbeitet" .	<code>"false"</code> Falls angegeben, auf <code>true</code> für ASA r2-Systeme setzen.
luksEncryption	LUKS-Verschlüsselung aktivieren. Weitere Informationen enthält "Verwenden Sie Linux Unified Key Setup (LUKS)" .	<code>""</code> Auf <code>false</code> für ASA r2-Systeme setzen.
tieringPolicy	Tiering-Richtlinie auf "keine" setzen Für ASA r2-Systeme nicht angeben.	
nameTemplate	Vorlage zum Erstellen benutzerdefinierter Volume-Namen.	<code>""</code>

Beispiele für die Volume-Bereitstellung

Hier ist ein Beispiel mit definierten 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'

```

HINWEIS

Für alle Volumes, die mit dem `ontap-san` Treiber erstellt werden, fügt Trident dem FlexVol zusätzlich 10 Prozent Kapazität hinzu, um die LUN-Metadaten aufzunehmen. Die LUN wird mit genau der Größe bereitgestellt, die der Benutzer im PVC anfordert. Trident fügt dem FlexVol 10 Prozent hinzu (wird als verfügbare Größe in ONTAP angezeigt).

Benutzer erhalten nun die von ihnen angeforderte nutzbare Kapazität. Diese Änderung verhindert außerdem, dass LUNs schreibgeschützt werden, solange der verfügbare Speicherplatz nicht vollständig genutzt ist. Dies gilt nicht für ontap-san-economy.

Für Backends, die `snapshotReserve` definieren, berechnet Trident die Größe der Volumes wie folgt:

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

Die 1,1 ist die zusätzlichen 10 Prozent, die Trident dem FlexVol hinzufügt, um die LUN-Metadaten aufzunehmen. Für `snapshotReserve` = 5 % und eine PVC-Anforderung von 5 GiB beträgt die Gesamtgröße des Volumes 5,79 GiB und die verfügbare Größe 5,5 GiB. Der `volume show`-Befehl sollte Ergebnisse ähnlich diesem Beispiel anzeigen:

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 Volume zu nutzen.

Minimale Konfigurationsbeispiele

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

HINWEIS

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

▼ ONTAP SAN-Beispiel

Dies ist eine Basiskonfiguration mit dem `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 Sie die Backend-Definition nach einem Switchover und Switchback während "SVM-Replikation und -Wiederherstellung" nicht manuell aktualisieren müssen.

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

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

▼ ONTAP SAN economy Beispiel

```
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 Basiskonfigurationsbeispiel `clientCertificate`, `clientPrivateKey`, und `trustedCACertificate` (optional, wenn eine vertrauenswürdige CA verwendet wird) werden in `backend.json` gefüllt und nehmen die base64-kodierten Werte des Clientzertifikats, des privaten Schlüssels bzw. des Zertifikats der vertrauenswürdigen CA an.

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

▼ Beispiele für bidirektionales CHAP

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

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 economy 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>
```

▼ Beispiel für eine ausgewogene Platzierung

Dieses Beispiel aktiviert die ausgewogene Platzierung in ONTAP, sodass ONTAP das Aggregat für neue Volumes auswählt. Die ausgewogene Platzierung erfordert ONTAP 9.19.1 oder höher. Details sind in ["Ausgewogene Platzierung mit ONTAP Backends verwenden"](#) zu finden.

```
---
version: 1
storageDriverName: ontap-san
backendName: san-balanced-placement
managementLIF: 10.0.0.1
svm: svm_iscsi
username: vsadmin
```

```
password: <password>
useBalancedPlacement: true
```

▼ NVMe/TCP-Beispiel

Sie müssen eine SVM mit NVMe auf Ihrem ONTAP Backend konfiguriert haben. 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
```

▼ Beispiel für die Backend-Konfiguration 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}}_{{.vo\
lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"

```

▼ formatOptions example 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 `spaceReserve` bei `none`, `spaceAllocation` bei `false` und `encryption` bei `false`. Die virtuellen Pools werden im Speicherabschnitt definiert.

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

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

▼ ONTAP SAN-Beispiel

```

---
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"

```

▼ ONTAP SAN economy 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>
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 zu StorageClasses zuordnen

Die folgenden StorageClass-Definitionen beziehen sich auf die [Beispiele für Backends mit virtuellen Pools](#). Mit dem `parameters.selector`-Feld wird bei jeder StorageClass angegeben, welche virtuellen Pools zum Hosten eines Volumes verwendet werden können. Das Volume verfügt nun über die im ausgewählten virtuellen Pool definierten Aspekte.

- Die `protection-gold` StorageClass verweist auf den ersten virtuellen Pool im `ontap-san` Backend. Dies ist der einzige Pool, der Schutz auf Gold-Niveau bietet.

```

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

```

- Die `protection-not-gold` StorageClass verweist auf den zweiten und dritten virtuellen Pool im `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"
```

- Die app-mysqldb StorageClass wird dem dritten virtuellen Pool im ontap-san-economy Backend zugeordnet. Dies ist der einzige Pool, der eine Speicherpoolkonfiguration für den Anwendungstyp 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 StorageClass verweist auf den zweiten virtuellen Pool im 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"
```

- Die creditpoints-5k StorageClass verweist auf den dritten virtuellen Pool im ontap-san Backend und auf den vierten virtuellen 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"
```

- Die my-test-app-sc StorageClass wird dem testAPP virtuellen Pool im ontap-san Treiber mit sanType: nvme zugeordnet. Dies ist der einzige Pool, der testApp anbietet.

```

---
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 Speicheranforderung erfüllt wird.

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 für die Kommunikation mit dem ONTAP Cluster bereit. Unterstützte Zugriffsmodi sind: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Treiber	Protokoll	volumeMode	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

WARNUNG

- Verwenden Sie `ontap-san-economy` nur, wenn die Anzahl der persistenten Speichernutzungen voraussichtlich höher ist als "[Unterstützte ONTAP Volume-Grenzwerte](#)".
- Verwenden Sie `ontap-nas-economy` nur, wenn die Anzahl der persistenten Speichernutzungen voraussichtlich höher ist als "[Unterstützte ONTAP Volume-Grenzwerte](#)" und der `ontap-san-economy` Treiber nicht verwendet werden kann.
- Verwenden Sie `ontap-nas-economy` nicht, wenn Sie einen Bedarf an Datenschutz, Notfallwiederherstellung oder Mobilität erwarten.
- NetApp empfiehlt nicht, Flexvol autogrow in allen ONTAP-Treibern zu verwenden, außer `ontap-san`. Als Workaround unterstützt Trident die Verwendung von Snapshot-Reserve und skaliert Flexvol-Volumes entsprechend.

Benutzerberechtigungen

Trident wird entweder als ONTAP- oder SVM-Administrator ausgeführt, typischerweise mit dem `admin cluster`-Benutzer oder einem `vsadmin` SVM-Benutzer oder einem Benutzer mit anderem Namen, der die gleiche Rolle hat.

Für Amazon FSx for NetApp ONTAP-Bereitstellungen wird Trident entweder als ONTAP- oder SVM-Administrator ausgeführt, mit dem `cluster fsxadmin` Benutzer oder einem `vsadmin` SVM-Benutzer oder einem Benutzer mit anderem Namen, der die gleiche Rolle hat. Der `fsxadmin` Benutzer ist ein eingeschränkter Ersatz für den `cluster-Admin`-Benutzer.

HINWEIS

Wenn Sie den `limitAggregateUsage` Parameter verwenden, sind Cluster-Administratorrechte erforderlich. Bei Verwendung von Amazon FSx für NetApp ONTAP mit Trident wird der `limitAggregateUsage` Parameter nicht mit den `vsadmin` und `fsxadmin` Benutzerkonten funktionieren. Der Konfigurationsvorgang schlägt fehl, wenn Sie diesen Parameter angeben.

Zwar ist es möglich, in ONTAP eine restriktivere Rolle zu erstellen, die ein Trident-Treiber verwenden kann, wir raten jedoch davon ab. Die meisten neuen Versionen von Trident rufen zusätzliche APIs auf, die berücksichtigt werden müssten, was Aktualisierungen erschwert und fehleranfällig macht.

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

Verstehen Sie die Anforderungen, Authentifizierungsoptionen und Exportrichtlinien für die Konfiguration eines ONTAP Backends mit ONTAP NAS-Treibern. Ab der Version 25.10 unterstützt NetApp Trident "[NetApp AFX-Speichersystem](#)". NetApp AFX-Speichersysteme unterscheiden sich von anderen ONTAP-Systemen (ASA, AFF und FAS) in der Implementierung ihrer Speicherschicht. In der Trident-Backend-Konfiguration müssen Sie nicht angeben, dass Ihr System AFX ist. Wenn Sie `ontap-nas` als `storageDriverName` auswählen, erkennt Trident die AFX-Systeme automatisch.

HINWEIS

Nur der `ontap-nas` Treiber (mit NFS-Protokoll) wird für AFX-Systeme unterstützt; das SMB-Protokoll wird nicht unterstützt.

Anforderungen

- Für alle ONTAP Backends erfordert Trident, dass mindestens ein Aggregat dem SVM zugewiesen wird.
- Sie können mehr als einen Treiber ausführen und Speicherklassen erstellen, die auf den einen oder den anderen verweisen. Beispielsweise könnten Sie eine Gold-Klasse konfigurieren, die den `ontap-nas` Treiber verwendet, und eine Bronze-Klasse, die den `ontap-nas-economy` verwendet.
- Auf allen Ihren Kubernetes-Worker-Knoten müssen die entsprechenden NFS-Tools installiert sein. Weitere Einzelheiten sind in "[hier](#)," enthalten.
- Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten ausgeführt werden. Details sind in [Bereiten Sie die Bereitstellung von SMB-Volumes vor](#) zu finden.

Authentifizieren Sie das ONTAP Backend

Trident bietet zwei Modi zur Authentifizierung eines ONTAP Backend.

- Anmeldeinformationsbasiert: Dieser Modus erfordert ausreichende Berechtigungen für das ONTAP

Backend. Es wird empfohlen, ein Konto zu verwenden, das einer vordefinierten Sicherheitsanmelderolle zugeordnet ist, wie `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. Hier muss die Backend-Definition Base64-kodierte Werte des Client-Zertifikats, des Schlüssels und, falls verwendet (empfohlen), des vertrauenswürdigen CA-Zertifikats enthalten.

Sie können bestehende Backends aktualisieren, um zwischen anmeldeinformationsbasierten und zertifikatsbasierten Authentifizierungsverfahren zu wechseln. Es wird jedoch jeweils nur ein Authentifizierungsverfahren unterstützt. Um zu einem anderen Authentifizierungsverfahren zu wechseln, müssen Sie das bestehende Verfahren aus der Backend-Konfiguration entfernen.

WARNUNG

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

Aktivieren Sie die anmeldeinformationsbasierte Authentifizierung

Trident benötigt die Anmeldeinformationen eines Administrators mit SVM- oder Cluster-Berechtigungen, um mit dem ONTAP Backend zu kommunizieren. Es wird empfohlen, vordefinierte Standardrollen wie `admin` oder `vsadmin` zu verwenden. Dies gewährleistet die Vorwärtskompatibilität mit zukünftigen ONTAP Versionen, die möglicherweise Feature-APIs für zukünftige Trident Versionen bereitstellen. Eine benutzerdefinierte Sicherheits-Anmelderolle kann erstellt und mit Trident verwendet werden, 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 Zugangsdaten im Klartext gespeichert werden. Nach der Backend-Erstellung werden Benutzernamen/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 vom Kubernetes-/Speicheradministrator durchzuführende Operation.

Zertifikatsbasierte Authentifizierung aktivieren

Neue und bestehende Backends können ein Zertifikat verwenden und mit dem ONTAP Backend kommunizieren. In der Backend-Definition sind drei Parameter erforderlich.

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

Ein typischer Arbeitsablauf umfasst die folgenden Schritte.

Schritte

1. Generieren Sie ein Clientzertifikat und einen Schlüssel. Legen Sie beim Generieren den Common Name (CN) auf den ONTAP Benutzer fest, als den Sie sich authentifizieren möchten.

```
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 wurde möglicherweise bereits vom Storage-Administrator erledigt. Ignorieren Sie dies, falls keine vertrauenswürdige CA 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 das cert Authentifizierungsverfahren unterstützt.

```
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 und den SVM-Namen. Sie müssen sicherstellen, dass die Service-Richtlinie des LIF auf default-data-management gesetzt ist.

```
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 das 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          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |          9 |
+-----+-----+-----+
+-----+-----+

```

Aktualisieren Sie Authentifizierungsverfahren oder rotieren Sie die Anmeldeinformationen

Sie können ein bestehendes Backend aktualisieren, um ein anderes Authentifizierungsverfahren zu verwenden oder um die Anmeldeinformationen zu rotieren. Dies funktioniert in beide Richtungen: Backends, die Benutzername/Passwort verwenden, können auf Zertifikate umgestellt werden; Backends, die Zertifikate verwenden, können auf Benutzername/Passwort umgestellt werden. Dazu müssen Sie das bestehende Authentifizierungsverfahren entfernen und das neue Authentifizierungsverfahren hinzufügen. Verwenden Sie dann die aktualisierte backend.json-Datei mit den erforderlichen Parametern, um `tridentctl update backend` auszuführen.

```
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          |
STATE | VOLUMES |
+-----+-----+-----+

```

```
+-----+-----+
| NasBackend | ontap-nas          | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |          9 |
+-----+-----+-----+-----+
+-----+-----+
```

HINWEIS

Beim Rotieren von Passwörtern muss der Speicheradministrator zunächst das Passwort für den Benutzer auf ONTAP aktualisieren. Danach erfolgt ein Backend-Update. Beim Rotieren von Zertifikaten können dem Benutzer mehrere Zertifikate hinzugefügt werden. Das Backend wird dann aktualisiert, um das neue Zertifikat zu verwenden, wonach das alte Zertifikat aus dem ONTAP Cluster gelöscht werden kann.

Die Aktualisierung des Backends beeinträchtigt weder den Zugriff auf bereits erstellte Volumes noch später hergestellte Volume-Verbindungen. Eine erfolgreiche Backend-Aktualisierung zeigt an, dass Trident mit dem ONTAP Backend kommunizieren und zukünftige Volume-Operationen ausführen kann.

Erstellen einer benutzerdefinierten ONTAP Rolle für Trident

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

Weitere Informationen zum Erstellen benutzerdefinierter Trident-Rollen finden Sie unter "[Trident Custom-Role-Generator](#)".

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>
```

Verwenden von System Manager

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 > Settings**.

(Oder) Um eine benutzerdefinierte Rolle auf SVM-Ebene zu erstellen, wählen Sie **Storage > Storage VMs > required svm > Settings > Users and Roles**.

- b. Wählen Sie das Pfeilsymbol (→) neben **Users and Roles** aus.
 - c. Wählen Sie unter **Rollen +Add** aus.
 - d. Definieren Sie die Regeln für die Rolle und klicken Sie auf **Save**.
2. **Rolle dem Trident-Benutzer zuordnen:** + Führen Sie die folgenden Schritte auf der Seite **Benutzer und Rollen** aus:
- a. Wählen Sie das Symbol + unter **Benutzer** aus.
 - b. Wählen Sie den gewünschten Benutzernamen aus und wählen Sie eine Rolle im Dropdown-Menü für **Rolle**.
 - c. Auf **Speichern** klicken.

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-Exportregeln, um den Zugriff auf die Volumes zu steuern, die es bereitstellt.

Trident bietet zwei Optionen für die Arbeit mit Exportrichtlinien:

- Trident kann die Exportregel dynamisch selbst verwalten; in diesem Betriebsmodus gibt der Speicheradministrator eine Liste von CIDR-Blöcken an, die zulässige IP-Adressen repräsentieren. Trident fügt anwendbare Knoten-IPs, die in diese Bereiche fallen, der Exportregel beim Veröffentlichen automatisch hinzu. Alternativ, wenn keine CIDRs angegeben werden, werden alle globalen Unicast-IPs, die auf dem Knoten gefunden werden, auf den das Volume veröffentlicht wird, zur Exportregel hinzugefügt.
- Speicheradministratoren können eine Exportregel erstellen und Regeln manuell hinzufügen. Trident verwendet die Standard-Exportregel, sofern in der Konfiguration kein anderer Name für die Exportregel angegeben ist.

Exportregeln dynamisch verwalten

Trident bietet die Möglichkeit, Exportregeln für ONTAP-Backends dynamisch zu verwalten. Dadurch erhält der Speicheradministrator die Möglichkeit, einen zulässigen Adressraum für Worker-Knoten-IPs anzugeben, anstatt explizite Regeln manuell zu definieren. Dies vereinfacht die Verwaltung der Exportregel erheblich; Änderungen an der Exportregel erfordern keine manuelle Intervention mehr am Storage-Cluster. Außerdem hilft dies, den Zugriff auf den Storage-Cluster nur auf Worker-Knoten zu beschränken, die Volumes einbinden und deren IPs im angegebenen Bereich liegen, was eine feingranulare und automatisierte Verwaltung unterstützt.

HINWEIS

Verwenden Sie keine Netzwerkadressübersetzung (NAT), wenn Sie dynamische Exportregeln verwenden. Mit NAT sieht der Speicherkontroller die Frontend-NAT-Adresse und nicht die tatsächliche IP-Hostadresse, sodass der Zugriff verweigert wird, 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
```

HINWEIS

Bei Verwendung dieser Funktion müssen Sie sicherstellen, dass die Root-Verbindung in Ihrer SVM über eine zuvor erstellte Exportregel verfügt, die den CIDR-Block des Knotens zulässt (wie z. B. die Standard-Exportregel). Befolgen Sie stets die von NetApp empfohlenen Best Practices, um eine SVM für Trident zu dedizieren.

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

- `autoExportPolicy` ist auf `true` gesetzt. Dies bedeutet, dass Trident für jedes mit diesem Backend bereitgestellte Volume für die `svm1` SVM eine Exportregel erstellt und das Hinzufügen sowie Löschen von Regeln mithilfe von `autoexportCIDRs` Adressblöcken verwaltet. Solange ein Volume nicht an einen Knoten angehängt ist, verwendet das Volume eine leere Exportregel ohne Regeln, um unerwünschten Zugriff auf dieses Volume zu verhindern. Wenn ein Volume auf einem Knoten veröffentlicht wird, erstellt Trident eine Exportregel mit demselben Namen wie der zugrunde liegende `qtree`, der die Knoten-IP innerhalb des angegebenen CIDR-Blocks enthält. Diese IPs werden auch der Exportregel 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
 - Der Qtree mit dem Namen `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` erstellt eine Exportrichtlinie für das FlexVol mit dem Namen `trident-403b5326-8482-40db96d0-d83fb3f4daec`, eine Exportrichtlinie für den Qtree mit dem Namen `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` und eine leere Exportrichtlinie mit dem Namen `trident_empty` auf dem SVM. Die Regeln der FlexVol-Exportrichtlinie bilden eine Obermenge aller in den Qtree-Exportrichtlinien enthaltenen Regeln. Die leere Exportrichtlinie wird von allen nicht angehängten Volumes wiederverwendet.
- `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 wurden.

In diesem Beispiel wird der 192.168.0.0/24 Adressraum bereitgestellt. Dies bedeutet, dass Kubernetes-Knoten-IPs, die in diesen Adressbereich fallen und Veröffentlichungen enthalten, der von Trident erstellten Exportregel hinzugefügt werden. Wenn Trident einen Knoten registriert, auf dem es ausgeführt wird, ruft es die IP-Adressen des Knotens ab und prüft sie anhand der in `autoExportCIDRs` bereitgestellten Adressblöcke. Zum Veröffentlichungszeitpunkt erstellt Trident nach dem Filtern der IPs die Exportregel für die Client-IPs des Knotens, auf den veröffentlicht wird.

Sie können `autoExportPolicy` und `autoExportCIDRs` für Backends nach deren Erstellung aktualisieren. Sie können neue CIDRs für ein automatisch verwaltetes Backend hinzufügen oder bestehende CIDRs löschen. Gehen Sie beim Löschen von CIDRs vorsichtig vor, um sicherzustellen, dass bestehende Verbindungen nicht getrennt werden. Sie können auch wählen, `autoExportPolicy` für ein Backend zu deaktivieren und auf eine manuell erstellte Exportrichtlinie zurückzugreifen. Dies erfordert das Setzen des `exportPolicy` Parameters in Ihrer Backend-Konfiguration.

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

```
./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 Exportregeln, um die Zugriffsregeln zu entfernen, die dem Knoten entsprechen. Durch das Entfernen dieser Knoten-IP aus den Exportregeln der verwalteten Backends verhindert Trident unerwünschte Einbindungen, es sei denn, diese IP wird von einem neuen Knoten im Cluster wiederverwendet.

Bei bereits vorhandenen Backends stellt die Aktualisierung des Backends mit `tridentctl update backend` sicher, dass Trident die Exportregeln automatisch verwaltet. Dabei werden bei Bedarf zwei neue Exportregeln erstellt, die nach der UUID und dem Qtree-Namen des Backends benannt sind. Volumes, die auf dem Backend vorhanden sind, verwenden nach dem Aushängen und erneuten Einhängen die neu erstellten Exportregeln.

HINWEIS

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

Wird die IP-Adresse eines aktiven Knotens geändert, muss der Trident Pod auf dem Knoten neu gestartet werden. Trident aktualisiert anschließend die Exportregel für die von ihm verwalteten Backends, um diese IP-Änderung widerzuspiegeln.

Bereiten Sie die Bereitstellung von SMB-Volumes vor

Mit ein wenig zusätzlicher Vorbereitung können Sie SMB-Volumes mit `ontap-nas` Treibern bereitstellen.

WARNUNG

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

HINWEIS

`autoExportPolicy` is not supported for SMB-Volumes.

Bevor Sie beginnen

Bevor Sie SMB-Volumes bereitstellen können, müssen Sie Folgendes haben.

- Ein Kubernetes-Cluster mit einem Linux-Controller-Knoten und mindestens einem Windows-Worker-Knoten, auf dem Windows Server 2022 läuft. Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten ausgeführt werden.
- Mindestens ein Trident secret, das Ihre Active Directory-Anmeldeinformationen enthält. Um ein secret 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 `csi-proxy` zu konfigurieren, siehe "[GitHub: CSI Proxy](#)" oder "[GitHub: CSI-Proxy für Windows](#)" für Kubernetes-Knoten, die unter Windows laufen.

Schritte

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

HINWEIS

SMB-Shares sind für Amazon FSx for ONTAP erforderlich.

Sie können die SMB-Administratorfreigaben auf zwei Arten erstellen: entweder mit dem "[Microsoft Management Console](#)" Shared Folders Snap-In oder mit der ONTAP CLI. Um die SMB-Freigaben mit der ONTAP CLI zu erstellen:

- a. Erstellen Sie bei Bedarf die Verzeichnispfadstruktur für die Freigabe.

Der `vserver cifs share create` Befehl prüft den in der Option `-path` beim Erstellen der Freigabe angegebenen Pfad. Existiert der angegebene Pfad nicht, 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
```

HINWEIS

Weitere Einzelheiten finden Sie in ["Erstellen Sie eine SMB-Freigabe"](#).

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

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

Sichere SMB-Verbindungen aktivieren

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

Wichtige Punkte

- Das Importieren `ontap-nas-economy` von Volumes wird nicht unterstützt.
- Es werden nur schreibgeschützte Klone für `ontap-nas-economy volumes` unterstützt.

- 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 share ACL.
- Die in der Annotation des Klon-PVC angegebene SMB-Share-ACL hat Vorrang vor denen im Quell-PVC.
- Stellen Sie sicher, dass Sie gültige AD-Benutzer angeben, wenn Sie sicheres SMB aktivieren. Ungültige Benutzer werden nicht zur ACL hinzugefügt.
- Wenn Sie dem gleichen AD-Benutzer im Backend, in der Speicherklasse und im PVC unterschiedliche Berechtigungen zuweisen, ist die Berechtigungsriorität: PVC, Speicherklasse und dann Backend.
- Secure SMB wird für `ontap-nas` managed Volume-Importe unterstützt und ist für unmanaged Volume-Importe nicht anwendbar.

Schritte

1. Geben Sie `adAdminUser` in `TridentBackendConfig` wie im folgenden Beispiel 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 die Speicherklasse ein.

Fügen Sie die `trident.netapp.io/smbShareAdUser` Annotation zur Storage Class hinzu, um sicheres SMB ohne Fehler zu aktivieren. Der für die Annotation `trident.netapp.io/smbShareAdUser` angegebene Benutzerwert sollte derselbe sein wie der im `smbcreds` Secret angegebene Benutzername. Sie können eines der folgenden für `smbShareAdUserPermission` auswählen: `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 erstellt 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

ONTAP NAS-Treiber können mit Ihrer Trident Installation erstellt und verwendet werden. Dieser Abschnitt enthält Beispiele für die Backend-Konfiguration und Details zur Zuordnung von Backends zu StorageClasses. Ab der Version 25.10 unterstützt NetApp Trident ["NetApp AFX-Speichersysteme"](#). NetApp AFX-Speichersysteme unterscheiden sich von anderen ONTAP-basierten Systemen (ASA, AFF und FAS) in der Implementierung ihrer Speicherschicht.

HINWEIS

Nur der `ontap-nas` Treiber (mit NFS-Protokoll) wird für NetApp AFX-Systeme unterstützt; das SMB-Protokoll wird nicht unterstützt.

Backend-Konfigurationsoptionen

In der Trident-Backend-Konfiguration müssen Sie nicht angeben, dass Ihr System ein NetApp AFX-Speichersystem ist. Wenn Sie `ontap-nas` als `storageDriverName` auswählen, erkennt Trident das AFX-Speichersystem automatisch. Einige Backend-Konfigurationsparameter sind für AFX-Speichersysteme nicht anwendbar.

Ab der Version 26.06 kann die ONTAP Balanced Placement-Funktion aktiviert werden, sodass ONTAP das Aggregat für neue Volumes auswählt. Balanced Placement ist für AFX Storage-Systeme mit ONTAP 9.19.1 oder neuer erforderlich. Siehe ["Ausgewogene Platzierung mit ONTAP Backends verwenden"](#) für Anforderungen und Beispiele.

Die folgende Tabelle zeigt die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
<code>version</code>		Immer 1
<code>storageDriverName</code>	<p>Name des Speichertreibers.</p> <p>Kann nach der Erstellung nicht mehr geändert werden. Zum Aktualisieren dieses Parameters ist die Erstellung eines neuen Backends erforderlich.</p> <div>HINWEIS Für NetApp AFX-Systeme wird nur <code>ontap-nas</code> unterstützt.</div>	<code>ontap-nas</code> , <code>ontap-nas-economy</code> , oder <code>ontap-nas-flexgroup</code>
<code>backendName</code>	Benutzerdefinierter Name oder das Speicher-Backend	Treibername + "_" + <code>dataLIF</code>
<code>managementLIF</code>	<p>IP-Adresse eines Clusters oder einer SVM-Management-LIF. Ein vollqualifizierter Domänenname (FQDN) kann 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, wie <code>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</code>. Für einen reibungslosen MetroCluster-Switchover siehe MetroCluster Beispiel.</p>	"10.0.0.1", "[2001:1234:abcd::fefe]"
<code>dataLIF</code>	<p>IP-Adresse des Protokoll-LIF. NetApp empfiehlt, <code>dataLIF</code> anzugeben. Falls nicht angegeben, ruft Trident <code>dataLIFs</code> vom SVM ab. Ein vollqualifizierter Domainname (FQDN) kann für die NFS-Mount-Operationen angegeben werden, wodurch die Erstellung eines Round-Robin-DNS zur Lastverteilung über mehrere <code>dataLIFs</code> ermöglicht wird. Kann nach der Ersteinrichtung geändert werden. Weitere Informationen enthält . 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, wie <code>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</code>. Für MetroCluster weglassen. Siehe MetroCluster Beispiel.</p>	Angegebene Adresse oder abgeleitet von SVM, falls nicht angegeben (nicht empfohlen)

Parameter	Beschreibung	Standard
svm	Zu verwendende virtuelle Speichermaschine Für MetroCluster auslassen. Siehe MetroCluster Beispiel .	Wird abgeleitet, wenn eine SVM managementLIF angegeben ist
autoExportPolicy	Automatische Erstellung und Aktualisierung von Exportrichtlinien aktivieren [Boolean]. Mit den Optionen autoExportPolicy und autoExportCIDRs kann Trident Exportrichtlinien automatisch verwalten.	false
autoExportCIDRs	Liste der CIDRs, anhand derer die Node-IPs von Kubernetes gefiltert werden, wenn autoExportPolicy aktiviert ist. Mit den Optionen autoExportPolicy und autoExportCIDRs kann Trident Exportrichtlinien automatisch verwalten.	["0.0.0.0/0", ":::0"]
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Volumes angewendet werden sollen	""
clientCertificate	Base64-kodierter Wert des Clientzertifikats. Wird für zertifikatbasierte Authentifizierung verwendet	""
clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für zertifikatbasierte Authentifizierung verwendet	""
trustedCACertificate	Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Optional. Wird für zertifikatbasierte Authentifizierung verwendet	""
username	Benutzername für die Verbindung zum Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet. Für die Active Directory Authentifizierung siehe "Authentifizieren Sie Trident bei einer Backend-SVM mit Active Directory-Anmeldeinformationen" .	
password	Passwort für die Verbindung zum Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet. Für die Active Directory Authentifizierung siehe "Authentifizieren Sie Trident bei einer Backend-SVM mit Active Directory-Anmeldeinformationen" .	
storagePrefix	<p>Präfix, das beim Bereitstellen neuer Volumes in der SVM verwendet wird. Kann nach der Festlegung nicht mehr geändert werden.</p> <div> <div>HINWEIS</div> <div>Bei Verwendung von ontap-nas-economy und einem storagePrefix, der 24 oder mehr Zeichen umfasst, ist das Speicherpräfix nicht in die Qtrees eingebettet, obwohl es im Volume-Namen enthalten ist.</div> </div>	"Trident"

Parameter	Beschreibung	Standard
aggregate	<p>Aggregat für die Bereitstellung (optional; falls festgelegt, muss es der SVM zugewiesen werden). Für den <code>ontap-nas-flexgroup</code> Treiber wird diese Option ignoriert. Wenn kein Aggregat zugewiesen ist, kann jedes der verfügbaren Aggregate zur Bereitstellung eines FlexGroup Volumes verwendet werden.</p> <div> <div>HINWEIS</div> <p>Wird das Aggregat in SVM aktualisiert, erfolgt die Aktualisierung in Trident automatisch durch regelmäßiges Abfragen von SVM, ohne dass der Trident Controller neu gestartet werden muss. Wenn in Trident ein bestimmtes Aggregat für die Volume-Bereitstellung konfiguriert wurde und dieses Aggregat umbenannt oder aus der SVM verschoben wird, wechselt das Backend in Trident beim Abfragen des SVM-Aggregats in den Fehlerzustand. Das Aggregat muss entweder auf eines geändert werden, das in der SVM vorhanden ist, oder vollständig entfernt werden, damit das Backend wieder online ist.</p> </div> <p>Nicht für AFX storage systems angeben.</p> <p>Nicht angeben, wenn <code>useBalancedPlacement</code> auf <code>true</code> gesetzt ist.</p>	""
useBalancedPlacement	<p>Aggregatauswahl für neue Volumes an ONTAP [Boolescher Wert]. Erfordert ONTAP 9.19.1 oder höher und <code>useREST</code> auf <code>true</code> gesetzt. Kann nicht mit <code>aggregate</code> kombiniert werden.</p> <p>Siehe "Ausgewogene Platzierung mit ONTAP Backends verwenden" für Anforderungen und Beispiele.</p> <p>Falls angegeben, immer auf <code>true</code> für AFX Storage-Systeme unter ONTAP 9.19.1 oder höher setzen.</p>	false; true für AFX Speichersysteme auf ONTAP 9.19.1 oder höher
limitAggregateUsage	<p>Die Bereitstellung schlägt fehl, wenn die Nutzung diesen Prozentsatz überschreitet. Gilt nicht für Amazon FSx for ONTAP. Nicht für AFX storage systems angeben.</p>	"" (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 Aggregate, die der SVM zugewiesen sind, werden zur Bereitstellung eines FlexGroup-Volumes verwendet. Unterstützt für den ontap-nas-flexgroup-Speichertreiber.</p> <p>HINWEIS Wird die Aggregatliste in SVM aktualisiert, wird sie in Trident automatisch durch Abfrage von SVM aktualisiert, ohne dass der Trident Controller neu gestartet werden muss. Wenn in Trident eine bestimmte Aggregatliste für die Volume-Bereitstellung konfiguriert wurde und diese Aggregatliste umbenannt oder aus SVM verschoben wird, wechselt das Backend in Trident beim Abfragen der SVM-Aggregatliste in den Fehlerzustand. Entweder muss die Aggregatliste durch eine in SVM vorhandene ersetzt oder vollständig entfernt werden, damit das Backend wieder online ist.</p>	""
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} Verwenden Sie debugTraceFlags nicht, es sei denn, Sie führen eine Fehlersuche durch und benötigen eine detaillierte Protokollausgabe.	null
nasType	Konfigurieren Sie die Erstellung von NFS- oder SMB-Volumes. Optionen sind <code>nfs</code> , <code>smb</code> oder <code>null</code> . Die Einstellung auf <code>null</code> verwendet standardmäßig NFS-Volumes. Falls angegeben, immer auf <code>nfs</code> für AFX-Speichersysteme setzen.	<code>nfs</code>
nfsMountOptions	Kommagetrennte Liste der NFS-Mountoptionen. Die Mountoptionen für Kubernetes-persistente Volumes werden normalerweise in Speicherklassen angegeben, aber wenn in einer Speicherklasse keine Mountoptionen angegeben sind, greift Trident auf die Mountoptionen zurück, die in der Konfigurationsdatei des Storage-Backends angegeben sind. Wenn weder in der Speicherklasse noch in der Konfigurationsdatei Mountoptionen angegeben sind, setzt Trident keine Mountoptionen für das zugehörige persistente Volume.	""

Parameter	Beschreibung	Standard
qtreesPerFlexvol	Maximale Qtrees pro FlexVol, muss im Bereich [50, 300] liegen	"200"
smbShare	Sie können eines der folgenden angeben: den Namen einer SMB-Freigabe, die mit der Microsoft Management Console oder ONTAP CLI erstellt wurde; einen Namen, damit Trident die SMB-Freigabe erstellen kann; oder Sie können den Parameter leer lassen, um den gemeinsamen Freigabezugriff auf Volumes zu verhindern. Dieser Parameter ist optional für lokale ONTAP. 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. <code>useREST</code> Wenn auf <code>true</code> gesetzt, verwendet Trident ONTAP REST APIs zur Kommunikation mit dem Backend; wenn auf <code>false</code> gesetzt, verwendet Trident ONTAPI (ZAPI)-Aufrufe zur Kommunikation mit dem Backend. Diese Funktion erfordert ONTAP 9.11.1 und höher. Zusätzlich muss die verwendete ONTAP-Anmelderolle Zugriff auf die <code>ontapi</code> Anwendung haben. Dies wird durch die vordefinierten <code>vsadmin</code> und <code>cluster-admin</code> Rollen erfüllt. Ab der Trident 24.06-Version und ONTAP 9.15.1 oder höher ist <code>useREST</code> standardmäßig auf <code>true</code> gesetzt; ändern Sie <code>useREST</code> auf <code>false</code> , um ONTAPI (ZAPI)-Aufrufe zu verwenden. Falls angegeben, immer auf <code>true</code> für AFX-Speichersysteme setzen.	<code>true</code> für ONTAP 9.15.1 oder höher, andernfalls <code>false</code> .
limitVolumePoolSize	Maximal anforderbare FlexVol-Größe bei Verwendung von Qtrees im <code>ontap-nas-economy</code> Backend.	"" (wird nicht standardmäßig erzwungen)
denyNewVolumePools	Beschränkt <code>ontap-nas-economy</code> Backends darauf, neue FlexVol Volumes zu erstellen, um ihre Qtrees zu enthalten. Für die Bereitstellung neuer PVs werden ausschließlich bereits vorhandene FlexVols verwendet.	
adAdminUser	Active Directory-Administratorbenutzer oder -Benutzergruppe mit vollem Zugriff auf SMB-Freigaben. Verwenden Sie diesen Parameter, um Administratorrechte für die SMB-Freigabe mit vollständiger Kontrolle zu erteilen.	

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

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

Parameter	Beschreibung	Standard
spaceAllocation	Speicherplatzzuweisung für Qtrees	"true"
spaceReserve	Platzreservierungsmodus; "none" (dünn) oder "volume" (dick)	"none"
snapshotPolicy	Zu verwendende Snapshot-Richtlinie	"none"
qosPolicy	QoS-Richtliniengruppe, die für erstellte Volumes zugewiesen werden soll. Wählen Sie eine der qosPolicy oder adaptiveQosPolicy pro Speicherpool/Backend aus.	""
adaptiveQosPolicy	Adaptive QoS-Richtliniengruppe zur Zuweisung für erstellte Volumes. Wählen Sie eine von qosPolicy oder adaptiveQosPolicy pro Storage-Pool/Backend. Wird von ontap-nas-economy nicht unterstützt.	""
snapshotReserve	Prozentsatz des für Snapshots reservierten Volumens	"0", falls `snapshotPolicy` "none", andernfalls ""
splitOnClone	Trennen Sie einen Klon bei seiner Erstellung von seinem übergeordneten Objekt	"false"
encryption	Aktivieren Sie NetApp Volume Encryption (NVE) auf dem neuen Volume; standardmäßig false. NVE muss auf dem Cluster lizenziert und aktiviert sein, damit diese Option verwendet werden kann. Wenn NAE im Backend aktiviert ist, ist jedes in Trident bereitgestellte Volume NAE-aktiviert. Weitere Informationen unter: "Wie Trident mit NVE und NAE zusammenarbeitet" .	"false"
tieringPolicy	Tiering-Richtlinie auf "none" setzen	
unixPermissions	Modus für neue Volumes	"777" für NFS-Volumes; leer (nicht zutreffend) für SMB-Volumes
snapshotDir	Steuert den Zugriff auf das .snapshot Verzeichnis	true, false (explizit festlegen).
exportPolicy	Zu verwendende Exportrichtlinie	"default"
securityStyle	Sicherheitsstil für neue Volumes. NFS unterstützt mixed und unix Sicherheitsstile. SMB unterstützt mixed und ntfs Sicherheitsstile.	NFS-Standard ist unix. SMB-Standard ist ntfs.
nameTemplate	Vorlage zum Erstellen benutzerdefinierter Volume-Namen.	""

HINWEIS

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 jede Komponente einzeln angewendet wird. Eine gemeinsam genutzte QoS-Richtliniengruppe erzwingt die Obergrenze für den Gesamtdurchsatz aller Workloads.

Beispiele für die Volume-Bereitstellung

Hier ist ein Beispiel mit definierten 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` verwendet Trident nun eine neue Berechnung, um sicherzustellen, dass die FlexVol mit dem `snapshotReserve`-Prozentsatz und dem PVC korrekt dimensioniert ist. Wenn der Benutzer einen PVC anfordert, erstellt Trident das ursprüngliche FlexVol mit mehr Speicherplatz mithilfe der neuen Berechnung. Diese Berechnung stellt sicher, dass der Benutzer den beschreibbaren Speicherplatz erhält, den er im PVC angefordert hat, und nicht weniger als angefordert. Vor v21.07, wenn der Benutzer einen PVC anfordert (zum Beispiel 5 GiB) und der `snapshotReserve` auf 50 Prozent gesetzt ist, erhält er nur 2,5 GiB beschreibbaren Speicherplatz. Dies liegt daran, dass das, was der Benutzer anfordert, das gesamte Volume ist und `snapshotReserve` ein Prozentsatz davon ist. Mit Trident 21.07 ist das, was der Benutzer anfordert, der beschreibbare Speicherplatz, und Trident definiert die `snapshotReserve` Zahl als Prozentsatz des gesamten Volumens. Dies gilt nicht für `ontap-nas-economy`. Siehe das folgende Beispiel, um zu sehen, wie dies funktioniert:

Die Berechnung erfolgt wie folgt:

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

Für `snapshotReserve = 50 %`, und eine PVC-Anforderung = 5 GiB, beträgt die Gesamtgröße des Volumes

5/0,5 = 10 GiB und die verfügbare Größe 5 GiB, was der vom Benutzer in der PVC-Anforderung angeforderten Größe entspricht. Der `volume show`-Befehl sollte Ergebnisse ähnlich diesem Beispiel anzeigen:

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.

Bestehende Backends aus früheren Installationen stellen Volumes wie oben beschrieben bereit, wenn Trident aktualisiert wird. Für Volumes, die vor dem Upgrade erstellt wurden, sollte deren Größe angepasst werden, damit die Änderung wirksam wird. Beispielsweise führte ein 2-GiB-PVC mit `snapshotReserve=50` zuvor zu einem Volume, das 1 GiB beschreibbaren Speicherplatz bereitstellt. Eine Größenänderung des Volumes auf beispielsweise 3 GiB stellt der Anwendung 3 GiB beschreibbaren Speicherplatz auf einem 6-GiB-Volume zur Verfügung.

Minimale Konfigurationsbeispiele

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

HINWEIS

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

▼ ONTAP NAS economy Beispiel

```
---
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 Sie die Backend-Definition nach einem Switchover und

Switchback während "SVM-Replikation und -Wiederherstellung" nicht manuell aktualisieren müssen.

Für einen nahtlosen Switchover und Switchback geben Sie die SVM mit `managementLIF` an und lassen Sie die `dataLIF` und `svm` Parameter weg. 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` befüllt und nehmen die Base64-kodierten Werte des Clientzertifikats, des privaten Schlüssels und des Zertifikats der vertrauenswürdigen Zertifizierungsstelle an.

```
---
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 gleichermaßen für die `ontap-nas-economy` und `ontap-nas-flexgroup` Treiber.

```
---
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 eine ausgewogene Platzierung

Dieses Beispiel aktiviert die ausgewogene Platzierung in ONTAP, sodass ONTAP das Aggregat für neue Volumes auswählt. Die ausgewogene Platzierung erfordert ONTAP 9.19.1 oder höher. Details sind in ["Ausgewogene Platzierung mit ONTAP Backends verwenden"](#) zu finden.

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-balanced-placement
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
useBalancedPlacement: true
```

▼ Beispiel für IPv6 addresses

Dieses Beispiel zeigt `managementLIF` die 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 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_
```

▼ Beispiel für die Backend-Konfiguration 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 für alle Speicherpools spezifische Standardwerte festgelegt, wie `spaceReserve` bei `none`, `spaceAllocation` bei `false` und `encryption` bei `false`. Die virtuellen Pools werden im Speicherabschnitt definiert.

Trident legt Bereitstellungsbezeichnungen im Feld „Kommentare“ fest. Kommentare werden auf FlexVol für `ontap-nas` oder auf FlexGroup für `ontap-nas-flexgroup` festgelegt. Trident kopiert alle auf einem virtuellen Pool vorhandenen Bezeichnungen bei der Bereitstellung auf das Speichervolume. Zur Vereinfachung können Speicheradministratoren Bezeichnungen pro virtuellem Pool definieren und Volumes nach Bezeichnung gruppieren.

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

▼ ONTAP NAS-Beispiel

```
---
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"

```

▼ ONTAP NAS FlexGroup Beispiel

```

---
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"

```

▼ ONTAP NAS economy Beispiel

```

---
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 zu StorageClasses zuordnen

Die folgenden StorageClass-Definitionen beziehen sich auf [Beispiele für Backends mit virtuellen Pools](#). Mit dem `parameters.selector`-Feld wird bei jeder StorageClass angegeben, welche virtuellen Pools zum Hosten eines Volumes verwendet werden können. Das Volume besitzt die im gewählten virtuellen Pool definierten Aspekte.

- Die `protection-gold` StorageClass wird den ersten und zweiten virtuellen Pool im `ontap-nas-flexgroup` Backend zugeordnet. Nur diese beiden Pools bieten Schutz auf Goldniveau.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass

```

```
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- Die protection-not-gold StorageClass verweist auf den dritten und vierten virtuellen Pool im 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"
```

- Die app-mysqldb StorageClass wird dem vierten virtuellen Pool im ontap-nas Backend zugeordnet. 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 StorageClass verweist auf den dritten virtuellen Pool im 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"
```

- Die `creditpoints-5k` StorageClass entspricht dem dritten virtuellen Pool im `ontap-nas` Backend und dem zweiten virtuellen 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 Speicheranforderung erfüllt wird.

Aktualisieren dataLIF nach der Erstkonfiguration

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>
```

HINWEIS

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

Beispiele für sicheres SMB

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 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
```

HINWEIS

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

Speicherklassenbeispiel mit ontap-nas-economy-Treiber

```
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
```

Ausgewogene Platzierung mit ONTAP Backends verwenden

Sie können die ONTAP Balanced Placement auf ONTAP Backends aktivieren. Mit Balanced Placement wählt ONTAP für jedes neue Volume das optimale Aggregat aus, sodass die Aggregatzuweisung in Trident nicht verwaltet werden muss.

Wie eine ausgewogene Platzierung funktioniert

Standardmäßig erkennt Trident die einem SVM zugewiesenen Aggregate, behandelt jedes Aggregat als Speicherpool und wählt ein Aggregat aus, wenn ein Volume bereitgestellt wird.

Wenn die ausgewogene Platzierung aktiviert ist:

- Trident meldet die gesamte SVM als einen einzigen Speicherpool statt einzelner Aggregate.
- ONTAP wählt für jedes neue Volume das Aggregat aus und gleicht die Volumes über die verfügbaren Aggregate aus.
- Trident erstellt Volumes mithilfe der ONTAP `/application/containers` REST API.

Balanced Placement ist für Unified ONTAP Systeme (AFF, FAS und ASA) und NetApp AFX Speichersysteme verfügbar.

Anforderungen

- Das Backend muss ONTAP 9.19.1 oder neuer verwenden.
- Trident muss mit ONTAP über REST kommunizieren. Ausgewogene Platzierung ist mit ZAPI nicht verfügbar.
- Die Backend-Konfiguration darf keine Aggregate angeben. Eine Kombination aus ausgeglichener Platzierung und expliziter Aggregatauswahl ist nicht möglich.

Standardwerte

Der Standardwert von `useBalancedPlacement` hängt vom Speichersystem ab:

Speichersystem	Standard	Hinweise
Unified ONTAP (AFF, FAS, ASA)	<code>false</code>	Trident wählt Aggregate aus, sofern die ausgewogene Platzierung nicht aktiviert ist.
NetApp AFX (ONTAP 9.19.1 oder höher)	<code>true</code>	Eine ausgewogene Platzierung ist erforderlich. Das Setzen von <code>useBalancedPlacement</code> auf <code>false</code> führt dazu, dass die Backend-Validierung fehlschlägt.
NetApp AFX (vor ONTAP 9.19.1)	<code>false</code>	Eine ausgewogene Platzierung ist nicht verfügbar.

Konfigurationsoption

Parameter	Beschreibung	Standard
useBalancedPlacement	Delegiert die Auswahl des Aggregats für neue Volumes an ONTAP. Diese Option wird im Stammverzeichnis der Backend-Konfiguration oder für einzelne virtuelle Pools festgelegt.	false; true für AFX auf ONTAP 9.19.1 oder höher

Ausgewogene Platzierung mit virtuellen Pools

Ausgewogene Platzierung funktioniert mit oder ohne virtuelle Pools:

- Ohne virtuelle Pools meldet Trident die SVM als einen einzigen Speicherpool und erstellt alle neuen Volumes mit ausgewogener Platzierung.
- Mit virtuellen Pools meldet Trident die virtuellen Pools und ordnet ihnen wie gewohnt Volumes zu. Wenn für den entsprechenden virtuellen Pool die Balanced Placement aktiviert ist, wählt ONTAP das Aggregat für das Volume aus.

Die Einstellung `useBalancedPlacement` kann im Stammverzeichnis der Backend-Konfiguration gesetzt werden, um sie auf alle virtuellen Pools anzuwenden, oder auf einzelnen virtuellen Pools, um sie selektiv zu aktivieren.

Beispiele

Das folgende Beispiel ermöglicht eine ausgewogene Platzierung auf einem einheitlichen ONTAP Backend:

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-balanced-placement
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: cluster-admin
password: <password>
useBalancedPlacement: true
```

Das folgende Beispiel ermöglicht eine ausgewogene Platzierung mithilfe von TridentBackendConfig:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: nas-balanced-placement
  managementLIF: 10.0.0.1
  svm: trident_svm
```

```

useBalancedPlacement: true
credentials:
  name: backend-tbc-ontap-nas-secret

```

Das folgende Beispiel ermöglicht die ausgewogene Platzierung nur im `gold` virtuellen Pool. Trident wählt weiterhin Aggregate für Volumes aus, die dem `bronze` virtuellen Pool entsprechen:

```

---
version: 1
storageDriverName: ontap-nas
backendName: nas-virtual-pools
managementLIF: 10.0.0.1
svm: trident_svm
username: cluster-admin
password: <password>
storage:
- labels:
    performance: gold
  useBalancedPlacement: true
- labels:
    performance: bronze

```

Backend-Validierungsfehler

Die Erstellung oder Aktualisierung des Backends schlägt in folgenden Fällen fehl:

Konfiguration	Ergebnis
<code>useBalancedPlacement: true</code> auf ONTAP vor 9.19.1	Die Validierung ist fehlgeschlagen. Ein Upgrade auf ONTAP 9.19.1 oder höher ist erforderlich.
<code>useBalancedPlacement: true</code> auf einem Backend mit ZAPI	Die Validierung ist fehlgeschlagen. Das Backend ist für die Verwendung von REST zu konfigurieren.
<code>useBalancedPlacement: true</code> kombiniert mit spezifischen Aggregaten	Die Validierung ist fehlgeschlagen. Die Aggregatkonfiguration muss entfernt oder die ausgeglichene Platzierung deaktiviert werden.
<code>useBalancedPlacement: false</code> auf AFX mit ONTAP 9.19.1 oder höher	Die Validierung ist fehlgeschlagen. Für AFX auf ONTAP 9.19.1 oder höher ist eine ausgewogene Platzierung erforderlich.

Amazon FSx for NetApp ONTAP

Verwenden Sie Trident mit Amazon FSx for NetApp ONTAP

"[Amazon FSx for NetApp ONTAP](#)" ist ein vollständig verwalteter AWS-Service, der Dateisysteme betreibt, die vom NetApp ONTAP-Speicherbetriebssystem unterstützt werden. Er bietet ONTAP-Funktionen, Leistung und Administration mit der Skalierbarkeit

und dem einfachen Betrieb von AWS. Ein Dateisystem ist die primäre Ressource in Amazon FSx und entspricht einem lokalen ONTAP-Cluster. Jedes Dateisystem enthält eine oder mehrere Storage Virtual Machines (SVMs), und jede SVM enthält ein oder mehrere Volumes, die Dateien und Verzeichnisse speichern. Diese Integration ermöglicht es Kubernetes-Clustern, die in Amazon Elastic Kubernetes Service (EKS) ausgeführt werden, ONTAP-gestützte persistente Volumes für Block- und Dateiworkloads bereitzustellen.

Anforderungen

Zusätzlich zu ["Trident-Anforderungen"](#) benötigen Sie zur Integration von FSx for ONTAP mit Trident Folgendes:

- 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 für ["NFS oder iSCSI"](#) vorbereitet sind.

HINWEIS

Stellen Sie sicher, dass Sie die für Amazon Linux und Ubuntu ["Amazon Machine Images"](#) (AMIs) erforderlichen Node-Vorbereitungsschritte je nach Ihrem EKS-AMI-Typ befolgen.

Überlegungen

- SMB-Volumes:
 - SMB-Volumes werden nur mit dem `ontap-nas` Treiber unterstützt.
 - SMB-Volumes werden mit dem Trident EKS-Add-on nicht unterstützt.
 - Trident unterstützt SMB-Volumes nur, wenn sie in Pods eingebunden sind, die auf Windows-Knoten ausgeführt werden. Details sind in ["Bereiten Sie die Bereitstellung von SMB-Volumes vor"](#) zu finden.
- Vor Trident 24.02 konnten Volumes, die auf Amazon FSx-Dateisystemen mit aktivierten automatischen Backups erstellt wurden, nicht von Trident gelöscht werden. Um dieses Problem in Trident 24.02 oder höher zu verhindern, geben Sie die `fsxFileSystemID`, `AWS apiRegion`, `AWS apikey` und `AWS secretKey` in der Backend-Konfigurationsdatei für AWS FSx for ONTAP an.

HINWEIS

Wenn Sie eine IAM-Rolle für Trident angeben, können Sie das explizite Angeben der Felder `apiRegion`, `apiKey` und `secretKey` für Trident weglassen. Weitere Informationen finden Sie unter ["FSx for ONTAP Konfigurationsoptionen und Beispiele"](#).

Gleichzeitige Verwendung 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 die erforderliche Multipath-Konfiguration auf den Knoten mit dem Amazon Elastic Block Store (EBS) CSI-Treiber in Konflikt geraten. Um sicherzustellen, dass Multipathing ohne Beeinträchtigung der EBS-Festplatten auf demselben Knoten funktioniert, müssen Sie EBS in Ihrer Multipathing-Konfiguration ausschließen. Dieses Beispiel zeigt eine `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 Modi der Authentifizierung.

- Anmeldeinformationsbasiert (Empfohlen): Speichert Anmeldeinformationen sicher im AWS Secrets Manager. Sie können den `fsxadmin` Benutzer für Ihr Dateisystem oder den `vsadmin` Benutzer, der für Ihre SVM konfiguriert ist, verwenden.

WARNUNG

Trident erwartet, als `vsadmin` SVM-Benutzer oder als Benutzer mit einem anderen Namen, aber derselben Rolle ausgeführt zu werden. Amazon FSx for NetApp ONTAP hat einen `fsxadmin` Benutzer, der ein eingeschränkter Ersatz für den ONTAP `admin` Cluster-Benutzer ist. Wir empfehlen dringend, `vsadmin` mit Trident zu verwenden.

- Zertifikatsbasiert: Trident wird mit der SVM auf Ihrem FSx-Dateisystem unter Verwendung eines auf Ihrer SVM installierten Zertifikats kommunizieren.

Einzelheiten zur Aktivierung der Authentifizierung finden Sie in der 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-economy	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 PV kann nicht ohne Neustart des Knotens gelöscht werden
- ** Funktioniert nicht mit NFSv3 mit Trident Version 25.02.

HINWEIS

Wenn Ihr gewünschtes AMI hier nicht aufgeführt ist, bedeutet das nicht, dass es nicht unterstützt wird; es wurde lediglich noch nicht getestet. Diese Liste dient als Orientierungshilfe 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, pod
- Löschen: pod, pvc (regulär, qtree/lun – economy, NAS mit AWS backup)

Weitere Informationen

- ["Amazon FSx for NetApp ONTAP Dokumentation"](#)
- ["Blogpost zu 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.

HINWEIS

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

AWS Secrets Manager-Geheimnis erstellen

Da Trident APIs gegen einen FSx vserver ausführt, um den Speicher für Sie zu verwalten, benötigt es hierfür Anmeldeinformationen. Der sichere Weg, diese Anmeldeinformationen zu übermitteln, ist über ein AWS Secrets Manager secret. Wenn Sie also noch keines haben, müssen Sie ein AWS Secrets Manager secret erstellen, das die Anmeldeinformationen für das vsadmin account enthält.

Dieses Beispiel erstellt ein AWS Secrets Manager secret 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 zu funktionieren. Daher müssen Sie eine Richtlinie erstellen, die Trident die erforderlichen 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"
```

Policy JSON-Beispiel:

```
{  
  "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 Service account association (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 alle AWS-Services zugreifen, auf die die Rolle Berechtigungen hat.

Pod-Identität

Amazon EKS Pod Identity associations bieten die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon EC2 instance profiles Anmeldeinformationen für Amazon EC2 instances bereitstellen.

Installieren Sie Pod Identity auf Ihrem EKS-Cluster:

Sie können eine Pod-Identität über die AWS-Konsole oder mit dem folgenden AWS CLI-Befehl erstellen:

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

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

Erstellen Sie trust-relationship.json:

Erstellen Sie trust-relationship.json, um dem EKS Service Principal zu ermöglichen, diese Rolle für Pod Identity zu übernehmen. Erstellen Sie dann 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 an die IAM-Rolle anhängen:

Hängen Sie die Rollenrichtlinie aus dem vorherigen Schritt an die erstellte IAM-Rolle an:

```
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 Sie eine 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 role für Service account association (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>** - Die OIDC Ihres EKS-Clusters. Sie können den `oidc_provider` durch Ausführen des folgenden Befehls abrufen:

```
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, hängen Sie die Richtlinie (die im vorherigen Schritt erstellt wurde) mit diesem Befehl an die Rolle an:

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

Überprüfen Sie, ob der OICD provider zugeordnet ist:

Überprüfen Sie, ob 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, verwenden Sie das folgende Beispiel, um eine IAM-Rolle für das 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 Amazon FSx for NetApp ONTAP Speicherverwaltung 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. Weitere Informationen finden Sie unter ["Snapshot-Funktionalität für CSI-Volumes aktivieren"](#).

Trident über helm installieren

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 den `helm list` Befehl verwenden, um Installationsdetails wie Name, Namespace, Chart, Status, App-Version und Revisionsnummer zu überprüfen.

```
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		

Service-Account-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 provider** und **cloud identity** 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 den `helm list` Befehl verwenden, um Installationsdetails wie Name, Namespace, Chart, Status, App-Version und Revisionsnummer zu überprüfen.

```
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-
100.2510.0	25.10.0		

Wenn Sie iSCSI verwenden möchten, stellen Sie sicher, dass iSCSI auf Ihrem Client-Rechner aktiviert ist. Wenn Sie das AL2023 Worker node OS verwenden, können Sie die Installation des iSCSI-Clients automatisieren, indem Sie den `node prep` Parameter in der `helm` Installation hinzufügen:

HINWEIS

```
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, Fehlerbehebungen und ist von AWS für die Verwendung mit Amazon EKS validiert. Das EKS-Add-on ermöglicht es Ihnen, konsequent sicherzustellen,

dass Ihre Amazon EKS-Cluster sicher und stabil sind, und reduziert den Arbeitsaufwand, den Sie für die Installation, Konfiguration und Aktualisierung von Add-ons aufwenden müssen.

Voraussetzungen

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

- Ein Amazon EKS-Clusterkonto mit Add-on-Abonnement
- AWS-Berechtigungen für den AWS Marketplace:
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AMI type: 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 file system

Aktivieren Sie das Trident Add-on für AWS

Verwaltungskonsole

1. Öffnen Sie die Amazon EKS-Konsole unter <https://console.aws.amazon.com/eks/home#/clusters>.
2. Wählen Sie im linken Navigationsbereich **Clusters** 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 dann **Weitere Add-ons abrufen**.
5. Führen Sie die folgenden Schritte aus, 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 Kästchen in der oberen rechten Ecke des Trident by NetApp-Feldes.
 - c. **Weiter** auswählen.
6. Gehen Sie auf der Seite **Ausgewählte Add-ons konfigurieren** wie folgt vor:

HINWEIS

Überspringen Sie diese Schritte, wenn Sie die Pod Identity association 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 **configurationValues**-Parameter im Abschnitt **Konfigurationswerte** auf den role-arn fest, den Sie im vorherigen Schritt erstellt haben (Wert sollte folgendes Format haben):

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

```
"cloudProvider": "AWS"

}
```

+

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

7. Wählen Sie **Weiter**.

8. Auf der Seite **Review and add** wählen Sie **Create**.

Nach Abschluss der Add-on-Installation sehen Sie Ihr installiertes Add-on.

AWS CLI

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

Für Pod Identity verwenden Sie das folgende Format:

HINWEIS | Verwenden Sie die

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

Für die IRSA-Authentifizierung verwenden Sie das folgende 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"
  }
}
```

HINWEIS

Ersetzen Sie `<role ARN>` durch den 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 Beispielbefehl 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. Wählen Sie im linken Navigationsbereich **Clusters** aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das NetApp Trident CSI Add-on aktualisieren möchten.
4. Wählen Sie die Registerkarte **Add-ons** aus.
5. Wählen Sie **Trident by NetApp** und dann **Bearbeiten** aus.
6. Gehen Sie auf der Seite **Trident konfigurieren von NetApp** wie folgt vor:
 - a. Wählen Sie die **Version** aus, die Sie verwenden möchten.
 - b. Erweitern Sie die **Optional configuration settings** und nehmen Sie bei Bedarf Anpassungen vor.
 - c. **Änderungen speichern** auswählen.

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-on. Ersetzen Sie `my-cluster` durch Ihren 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 unter UPDATE AVAILABLE in der Ausgabe des vorherigen Schritts zurückgegeben wurde.

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

Wenn Sie die `--force` Option entfernen und eine der Amazon EKS Add-on-Einstellungen mit Ihren bestehenden Einstellungen in Konflikt steht, schlägt die Aktualisierung des Amazon EKS Add-ons fehl; Sie erhalten eine Fehlermeldung, die Ihnen hilft, den Konflikt zu beheben. Bevor Sie diese Option festlegen, stellen Sie sicher, dass das Amazon EKS Add-on keine Einstellungen verwaltet, die Sie verwalten müssen, da diese Einstellungen mit dieser Option überschrieben werden. Weitere Informationen zu anderen Optionen für diese Einstellung finden Sie unter ["Add-ons"](#). Weitere Informationen zur Amazon EKS Kubernetes-Feldverwaltung finden Sie unter ["Kubernetes-Feldmanagement"](#).

Deinstallieren/entfernen Sie das Trident EKS Add-on

Sie haben zwei Optionen, ein Amazon EKS add-on zu entfernen:

- **Add-on-Software auf Ihrem Cluster beibehalten** – Diese Option entfernt die Amazon EKS-Verwaltung aller Einstellungen. Außerdem wird die Möglichkeit entfernt, dass Amazon EKS Sie über Updates benachrichtigt und das Amazon EKS-Add-on nach einer Aktualisierung automatisch aktualisiert. Die Add-on-Software selbst bleibt jedoch auf Ihrem Cluster erhalten. Mit dieser Option wird das Add-on zu einer selbstverwalteten Installation, anstatt ein Amazon EKS-Add-on zu sein. Mit dieser Option gibt es keine Ausfallzeiten für das Add-on. Behalten Sie die `--preserve` Option im Befehl bei, um das Add-on beizubehalten.
- **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 auf Ihrem Cluster davon abhängig sind. Entfernen Sie die `--preserve` Option aus dem `delete` Befehl, um das Add-on zu entfernen.

HINWEIS

Wenn dem Add-on eine 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. Wählen Sie im linken Navigationsbereich **Clusters** 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 dann **Trident by NetApp**.*

5. Wählen Sie **Remove**.
6. Führen Sie im Dialogfeld **Remove netapp_trident-operator confirmation** die folgenden Schritte aus:
 - a. Wenn Amazon EKS die Einstellungen für das Add-on nicht mehr verwalten soll, wählen Sie **Preserve on cluster**. Tun Sie dies, wenn Sie die Add-on-Software auf Ihrem Cluster behalten und alle Einstellungen des Add-ons selbst verwalten möchten.
 - b. Geben Sie **netapp_trident-operator** ein.
 - c. Wählen Sie **Remove**.

AWS CLI

Ersetzen Sie `my-cluster` durch den 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 einer Speicherklasse

Der "[Kubernetes StorageClass-Objekt](#)" identifiziert einen Provisioner und weist den Provisioner an, wie Volumes bereitgestellt werden sollen. In diesem Abschnitt wird gezeigt, wie Sie ein Kubernetes-StorageClass-Objekt konfigurieren, das Trident als Provisioner angibt.

Erstellen Sie ein StorageClass-Objekt

Wenn Sie eine StorageClass für FSx for ONTAP erstellen, erstellt Trident automatisch die Backend-Konfiguration.

HINWEIS

Wenn Sie das Speicher-Backend manuell konfigurieren möchten, lesen Sie bitte den [Erstellen Sie eine Kubernetes StorageClass ohne automatische Backend-Konfiguration](#) Abschnitt, um zu erfahren, wie Sie das Trident-Backend und die Speicherklasse separat erstellen.

Geben Sie die erforderlichen StorageClass-Parameter an

Die folgenden drei Parameter müssen beim Erstellen einer StorageClass definiert werden:

Parameter	Erforderlich	Typ	Beschreibung
<code>fsxFilesystemID</code>	Ja	Zeichenkette	FSx für NetApp ONTAP filesystem ID

Parameter	Erforderlich	Typ	Beschreibung
storageDriverName	Ja	Zeichenkette	Trident-Speichertreiber (zum Beispiel, <code>ontap-nas</code> oder <code>ontap-san</code>)
credentialsName	Ja	Zeichenkette	Name des Kubernetes-Secrets, das die FSx for ONTAP Anmeldeinformationen enthält

Optionale Parameter angeben

Sie können optionale Backend-Parameter über die StorageClass übergeben. Definieren Sie alle optionalen Werte als Zeichenketten im StorageClass `parameters` Abschnitt. Eine vollständige Liste der Backend-Parameter finden Sie unter: ["FSx für NetApp ONTAP Backend-Konfiguration"](#).

Beispiel-StorageClass-Konfigurationsdateien.

Das folgende Beispiel zeigt eine StorageClass, die eine automatische Backend-Konfiguration auslöst.

YAML

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-fsx-demo
  annotations:
    description: "Demo StorageClass for FSx for NetApp ONTAP"
provisioner: csi.trident.netapp.io
parameters:
  fsxFileSystemID: "fs-0abc123"
  storageDriverName: "ontap-nas"
  credentialsName: trident-fsx-credentials
allowVolumeExpansion: true
reclaimPolicy: Delete
volumeBindingMode: Immediate

```

JSON

```

{
  "apiVersion": "storage.k8s.io/v1",
  "kind": "StorageClass",
  "metadata": {
    "name": "ontap-fsx-demo",
    "annotations": {
      "description": "Demo StorageClass for FSx for NetApp ONTAP"
    }
  },
  "provisioner": "csi.trident.netapp.io",
  "parameters": {

```

```

    "fsxFilesystemID": "fs-0abc123",
    "storageDriverName": "ontap-nas",
    "credentialsName": "trident-fsx-credentials"
  },
  "allowVolumeExpansion": true,
  "reclaimPolicy": "Delete",
  "volumeBindingMode": "Immediate"
}

```

Erstelle die StorageClass

Sobald Sie Ihre Konfigurationsdatei erstellt haben, führen Sie den folgenden Befehl aus, um die Speicherklasse zu erstellen.

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

Sie sollten nun eine **basic-csi** Storage-Class sowohl in Kubernetes als auch in Trident 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

Nachdem Sie die StorageClass angewendet haben, erstellt Trident das Backend automatisch. Sie können dann PersistentVolumeClaims erstellen, die auf diese StorageClass verweisen.

Überprüfen Sie den Status der Backend-Konfiguration

Trident speichert das Ergebnis der Backend-Erstellung in StorageClass-Annotationen.

Anmerkung	Beschreibung
trident.netapp.io/configuratorStatus	Konfigurationsergebnis (Success or Failure)
trident.netapp.io/configuratorMessage	Detaillierte Status- oder Fehlermeldung
trident.netapp.io/configuratorName	Name der internen Konfigurator-Ressource
trident.netapp.io/managed	Gibt an, dass die StorageClass von Trident verwaltet wird
trident.netapp.io/additionalStoragePools	Für dieses Backend erstellte Speicherpools

Um den Status zu überprüfen, führen Sie Folgendes aus:

```
kubectl get storageclass ontap-fsx-demo -o yaml
```

Bestätigen Sie, dass `trident.netapp.io/configuratorStatus` auf `Success` gesetzt ist. Wenn der Wert `Failure` ist, überprüfen Sie `trident.netapp.io/configuratorMessage` auf den Fehler.

Fügen Sie zusätzliche FSxN-Dateisysteme hinzu

Wenn Sie zusätzliche Speicherkapazität benötigen, während Sie weiterhin dieselbe StorageClass verwenden, fügen Sie zusätzliche FSxN-Dateisystem-IDs hinzu.

Bearbeiten Sie die StorageClass und fügen Sie die folgende Anmerkung hinzu:

```
metadata:
  annotations:
    trident.netapp.io/additionalFsxnFileSystemID: '["fs-
xxxxxxxxxxxxxxxxxxxxxx"] '
```

Nachdem Sie die Änderung angewendet haben, aktualisiert Trident die Backend-Konfiguration und die StorageClass-Annotationen.

Betriebliche Überlegungen und Einschränkungen

- Das Löschen einer StorageClass, die als automatische Backend-Konfiguration verwendet wird, löscht in der Regel auch das zugehörige Trident-Backend. Dies kann die Speicherkonnektivität beeinträchtigen und laufende Workloads unterbrechen. Prüfen Sie die Auswirkungen, bevor Sie eine verwaltete StorageClass löschen.
- Die automatische Backend-Konfiguration wird nur für AWS FSx für NetApp ONTAP unterstützt.

Erstellen Sie eine Kubernetes StorageClass ohne automatische Backend-Konfiguration

Wenn Sie das Trident Backend und StorageClass separat erstellen möchten, folgen Sie diesen Schritten.

Verstehen Sie, wie die automatische Backend-Konfiguration funktioniert

Trident leitet die Backend-Konfiguration aus der StorageClass-Definition ab. Wenn Sie die StorageClass anwenden, validiert Trident die erforderlichen Parameter, erstellt das Backend und versieht die StorageClass mit dem Status.

Trident erstellt die VolumeSnapshotClass nur einmal. Trident verwendet dieselbe VolumeSnapshotClass für nachfolgende StorageClasses wieder.

Erstelle das Trident Backend

Um ein Trident-Backend zu erstellen, benötigen Sie eine Konfigurationsdatei im JSON- oder YAML-Format. Die Datei muss den gewünschten Speichertyp (NAS oder SAN), das Dateisystem, die SVM, von der Sie es beziehen möchten, und die Authentifizierungsmethode angeben. Das folgende Beispiel zeigt, wie Sie NAS-basierten Speicher definieren und mithilfe eines AWS-Secrets die Anmeldeinformationen für die SVM speichern, die Sie verwenden möchten:

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"
    }
  }
}
```

FSx for ONTAP-Treiberdetails

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

Treibername	Beschreibung
ontap-san	Jedes bereitgestellte PV ist eine LUN innerhalb eines eigenen Amazon FSx for NetApp ONTAP-Volumes. Empfohlen für Blockspeicher.
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 Informationen zum Treiber finden Sie unter ["NAS-Treiber"](#) und ["SAN-Treiber"](#).

Das Backend erstellen

Nach dem Erstellen der Konfigurationsdatei 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 den 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

Weitere Informationen zu anderen Konfigurationsoptionen finden Sie im [Erweiterte Backend Konfiguration und Beispiele](#) Abschnitt unten.

Konfigurieren einer Storage Class ohne automatische Backend-Konfiguration

Nachfolgend finden Sie Beispiele für Speicherklassenkonfigurationen zur Verwendung mit Trident und FSx für ONTAP.

Speicherklasse für NFS

Sie können dieses Beispiel verwenden, um StorageClass für Volumes mit NFS einzurichten (eine 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"
```

Speicherklasse für iSCSI

Verwenden Sie dieses Beispiel, um StorageClass 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"
```

Speicherklasse mit NFSv3 und AWS Bottlerocket

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

```
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

```

Trident StorageClass-Attribute

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
Medien ¹	Zeichenkette	hdd, hybrid, ssd	Pool enthält Medien dieses Typs; hybrid bedeutet beides	Medientyp angeben	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
provisioningType	Zeichenkette	dünn, dick	Pool unterstützt diese Bereitstellungsmethode	Bereitstellungsmethode angeben	dick: alle ontap; dünn: alle ontap & solidfire-san
backendType	Zeichenkette	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, azure-netapp-files, ontap-san-economy	Pool gehört zu dieser Art von Backend	Backend angeben	Alle Treiber
Momentaufnahmen	bool	wahr, falsch	Pool unterstützt Volumes mit Snapshots	Volume mit aktivierten Snapshots	ontap-nas, ontap-san, solidfire-san
Klone	bool	wahr, falsch	Pool unterstützt das Klonen von Volumes	Volume mit aktivierten Klonen	ontap-nas, ontap-san, solidfire-san
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	Volume garantiert diese IOPS	solidfire-san

¹: Wird von ONTAP Select oder FSx für ONTAP-Systeme nicht unterstützt

Siehe "[Kubernetes- und Trident-Objekte](#)" für Details dazu, wie Storage-Klassen mit dem PersistentVolumeClaim interagieren und Parameter zur Steuerung, wie Trident Volumes bereitstellt.

Erstellen Sie die Speicherklasse

Sobald Sie die StorageClass konfiguriert haben, können Sie sie in Kubernetes erstellen.

Schritte

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

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

2. Sie sollten nun eine **basic-csi** Storage-Class sowohl in Kubernetes als auch in Trident 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

SMB-Volumen bereitstellen

Sie können SMB-Volumen mithilfe des `ontap-nas` Treibers bereitstellen. Dazu müssen Sie jedoch die folgenden Schritte ausführen: "[Bereiten Sie die Bereitstellung von SMB-Volumen vor](#)".

Erweiterte Backend-Konfiguration und Beispiele

Siehe die folgende Tabelle für 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	Treibername + "_" + dataLIF

Parameter	Beschreibung	Beispiel
managementLIF	<p>IP-Adresse eines Clusters oder einer SVM-Management-LIF. Ein vollqualifizierter Domänenname (FQDN) kann 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, zum Beispiel [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Wenn Sie die fsxFilesystemID unter dem aws Feld angeben, müssen Sie die managementLIF nicht angeben, da Trident die SVM managementLIF Informationen von AWS abrufen. Sie müssen daher Anmeldeinformationen für einen Benutzer unter der SVM (zum Beispiel: vsadmin) angeben, und der Benutzer muss die vsadmin Rolle haben.</p>	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parameter	Beschreibung	Beispiel
dataLIF	IP-Adresse des Protokoll-LIF. ONTAP NAS-Treiber: NetApp empfiehlt, dataLIF anzugeben. Falls nicht angegeben, ruft Trident dataLIFs vom SVM ab. Ein vollqualifizierter Domainname (FQDN) kann für die NFS-Mount-Operationen angegeben werden, wodurch die Erstellung eines Round-Robin-DNS zur Lastverteilung über mehrere dataLIFs ermöglicht wird. Kann nach der Ersteinrichtung geändert werden. ONTAP SAN-Treiber: Für iSCSI nicht angeben. Trident verwendet ONTAP Selective LUN Map, um die für den Aufbau einer Multipath-Sitzung benötigten iSCSI-LIFs zu ermitteln. Eine Warnung wird ausgegeben, 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, zum Beispiel [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].	
autoExportPolicy	Automatische Erstellung und Aktualisierung von Exportrichtlinien aktivieren [Boolean]. Mit den Optionen autoExportPolicy und autoExportCIDRs kann Trident Exportrichtlinien automatisch verwalten.	false
autoExportCIDRs	Liste der CIDRs, anhand derer die Node-IPs von Kubernetes gefiltert werden, wenn autoExportPolicy aktiviert ist. Mit den Optionen autoExportPolicy und autoExportCIDRs kann Trident Exportrichtlinien automatisch verwalten.	"["0.0.0.0/0", "::/0"]"
labels	Satz beliebiger JSON-formatierter Bezeichnungen, die auf Volumes angewendet werden sollen	""

Parameter	Beschreibung	Beispiel
clientCertificate	Base64-kodierter Wert des Clientzertifikats. Wird für zertifikatbasierte Authentifizierung verwendet	""
clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für zertifikatbasierte Authentifizierung verwendet	""
trustedCACertificate	Base64-kodierter Wert des vertrauenswürdigen CA-Zertifikats. Optional. Wird für zertifikatbasierte Authentifizierung verwendet.	""
username	Benutzername für die Verbindung zum Cluster oder zur SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet. Zum Beispiel vsadmin.	
password	Passwort zur Verbindung mit dem Cluster oder der SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet.	
svm	Zu verwendende Storage Virtual Machine	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 für NetApp ONTAP angeben. Die bereitgestellten <code>fsxadmin</code> und <code>vsadmin</code> enthalten nicht die erforderlichen Berechtigungen, um die aggregierte Nutzung abzurufen und sie mit Trident zu begrenzen.	Nicht verwenden.

Parameter	Beschreibung	Beispiel
limitVolumeSize	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet. Außerdem wird die maximale Größe der von ihr verwalteten Volumes für qtrees und LUNs beschränkt, und die qtreesPerFlexvol Option ermöglicht die Anpassung der maximalen Anzahl von qtrees pro FlexVol volume.	"" (wird nicht standardmäßig erzwungen)
lunsPerFlexvol	Maximale 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} Verwenden Sie debugTraceFlags nicht, es sei denn, Sie führen eine Fehlersuche durch und benötigen eine detaillierte Protokollausgabe.	null
nfsMountOptions	Kommagetrennte Liste der NFS-Mountoptionen. Die Mountoptionen für persistente Kubernetes-Volumes werden normalerweise in Speicherklassen angegeben, aber wenn keine Mountoptionen in einer Speicherklasse angegeben sind, verwendet Trident die Mountoptionen, die in der Konfigurationsdatei des Storage-Backends angegeben sind. Wenn weder in der Speicherklasse noch in der Konfigurationsdatei Mountoptionen angegeben sind, setzt Trident keine Mountoptionen für das zugehörige persistente Volume.	""
nasType	Konfigurieren Sie die Erstellung von NFS- oder SMB-Volumes. Optionen sind nfs, smb oder null. Muss auf smb gesetzt werden für SMB-Volumes. Die Einstellung auf null verwendet standardmäßig NFS-Volumes.	nfs
qtreesPerFlexvol	Maximale 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 ONTAP CLI erstellt wurde, oder einen Namen, damit 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 auf <code>true</code> gesetzt, verwendet Trident ONTAP REST APIs, um mit dem Backend zu kommunizieren. Diese Funktion erfordert ONTAP 9.11.1 und höher. Zusätzlich muss die verwendete ONTAP-Anmelderolle Zugriff auf die <code>ontap</code> Anwendung haben. Dies wird durch die vordefinierten <code>vsadmin</code> und <code>cluster-admin</code> Rollen erfüllt.	false
aws	Sie können Folgendes in der Konfigurationsdatei für AWS FSx for ONTAP angeben: - fsxFilesystemID: Geben Sie die ID des AWS FSx-Dateisystems an. - apiRegion: Name der AWS API-Region. - apikey: AWS API-Schlüssel. - secretKey: Geheimer AWS-Schlüssel.	"" "" ""
credentials	Geben Sie die FSx SVM-Anmeldeinformationen an, die im AWS Secrets Manager gespeichert werden sollen. - name: Amazon Resource Name (ARN) des Secrets, das die Anmeldeinformationen der SVM enthält. - type: Auf <code>awsarn</code> setzen. Weitere Informationen finden Sie unter "Erstellen Sie ein AWS Secrets Manager secret" .	

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

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

Parameter	Beschreibung	Standard
spaceAllocation	Speicherplatzzuweisung für LUNs	true
spaceReserve	Platzreservierungsmodus; "none" (dünn) oder "volume" (dick)	none
snapshotPolicy	Zu verwendende Snapshot-Richtlinie	none
qosPolicy	QoS-Richtliniengruppe, die für erstellte Volumes zugewiesen werden soll. Wählen Sie eine der qosPolicy oder adaptiveQosPolicy pro Speicherpool oder Backend. 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 jedes einzelne Element angewendet wird. Eine gemeinsam genutzte QoS-Richtliniengruppe erzwingt die Obergrenze für den Gesamtdurchsatz aller Workloads.	""
adaptiveQosPolicy	Adaptive QoS-Richtliniengruppe zur Zuweisung für erstellte Volumes. Wählen Sie eine der qosPolicy oder adaptiveQosPolicy pro Speicherpool oder Backend. Wird von ontap-nas-economy nicht unterstützt.	""
snapshotReserve	Prozentsatz des für Snapshots reservierten Volumens "0"	Wenn snapshotPolicy ist none, else ""
splitOnClone	Trennen Sie einen Klon bei seiner Erstellung von seinem übergeordneten Objekt	false
encryption	Aktivieren Sie NetApp Volume Encryption (NVE) auf dem neuen Volume; standardmäßig false. NVE muss auf dem Cluster lizenziert und aktiviert sein, damit diese Option verwendet werden kann. Wenn NAE im Backend aktiviert ist, wird jedes in Trident bereitgestellte Volume NAE-aktiviert. Weitere Informationen unter: "Wie Trident mit NVE und NAE zusammenarbeitet" .	false

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

PVC konfigurieren

Dieser Abschnitt enthält Anweisungen zum Erstellen eines PersistentVolumeClaim (PVC), der die konfigurierte Kubernetes-StorageClass verwendet, um ein PV anzufordern. Im Erfolgsfall kann das PV anschließend in einen Pod eingebunden werden.

PVC erstellen

Ein ["PersistentVolumeClaim"](#) (PVC) ist eine Anfrage für den Zugriff auf das PersistentVolume im Cluster. Die PVC kann so konfiguriert werden, dass sie Speicherplatz einer bestimmten Größe oder eines bestimmten Zugriffsmodus anfordert. Mithilfe der zugehörigen StorageClass kann der Clusteradministrator mehr als nur PersistentVolume-Größe und Zugriffsmodus steuern – zum Beispiel Leistung oder Servicelevel.

Nachdem Sie das Trident-Backend und die StorageClass erstellt haben, können Sie ein PVC erstellen. Sobald das PVC erstellt ist, können Sie das Volume in einem Pod einbinden.

Beispielmanifeste

Die folgenden Beispiele zeigen grundlegende PVC-Konfigurationsoptionen.

PVC mit RWX-Zugriff

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

```
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 protection-gold zugeordnet ist.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

PVC erstellen

Schritte

1. Die PVC wird erstellt.

```
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](#)" für Details dazu, wie Storage-Klassen mit dem PersistentVolumeClaim interagieren und Parameter zur Steuerung, wie Trident Volumes bereitstellt.

Eine Anwendung bereitstellen

Sobald die Storage Class und die PVC erstellt sind, können Sie das PV an einen Pod anhängen. In diesem Abschnitt finden Sie den Beispielbefehl und die Konfiguration, um das PV an einen Pod anzuhängen.

Eine Beispielanwendung bereitstellen

Schritte

1. Mountieren Sie das Volume in einem Pod.

```
kubectl create -f pv-pod.yaml
```

Diese Beispiele zeigen grundlegende Konfigurationen, um das PVC an ein Pod anzuhängen:

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
```

HINWEIS

Sie können den Fortschritt mit `kubectl get pod --watch` überwachen.

2. Überprüfen Sie, ob das Volume auf `/my/mount/path` eingebunden ist.

```
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 nun löschen. Die Pod-Anwendung existiert dann nicht mehr, aber das Volume bleibt erhalten.

```
kubectl delete pod pv-pod
```

Konfigurieren Sie das Trident EKS add-on auf einem EKS Cluster

NetApp Trident vereinfacht die Verwaltung von Amazon FSx for NetApp ONTAP Storage in Kubernetes, sodass sich Ihre Entwickler und Administratoren auf die Anwendungsbereitstellung konzentrieren können. Das NetApp Trident EKS Add-on enthält die neuesten Sicherheitspatches, Fehlerbehebungen und ist von AWS für die Zusammenarbeit mit Amazon EKS validiert. Das EKS-Add-on ermöglicht es Ihnen, konsequent sicherzustellen, dass Ihre Amazon EKS-Cluster sicher und stabil sind, und reduziert den Arbeitsaufwand, den Sie für die Installation, Konfiguration und Aktualisierung von Add-ons aufwenden müssen.

Voraussetzungen

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

- Ein Amazon EKS-Clusterkonto mit Berechtigungen zur Verwendung von Add-ons. Weitere Informationen enthält ["Amazon EKS Add-ons"](#).
- AWS-Berechtigungen für den AWS Marketplace:
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AML type: 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 file system

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 sich unter ["Erstellen Sie eine IAM-Rolle und ein AWS-Geheimnis"](#).
2. Navigieren Sie in Ihrem EKS Kubernetes-Cluster zur Registerkarte **Add-ons**.



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

Kubernetes version [Info](#)

1.30

Support period

① Standard support until July 28, 2025

Provider

EKS

Cluster health issues

✓ 0

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 *storage*.

AWS Marketplace add-ons (1)

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. ✕

< 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, und klicken Sie auf **Weiter**.

5. Wählen Sie die gewünschte Version des Add-on.

Configure selected add-ons settings

Configure the add-ons for your cluster by selecting settings.

NetApp Trident

Remove add-on

Listed by

Category

Status

NetApp

storage

Ready to install

You're subscribed to this software

View subscription

You can view the terms and pricing details for this product or choose another offer if one is available.

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.		

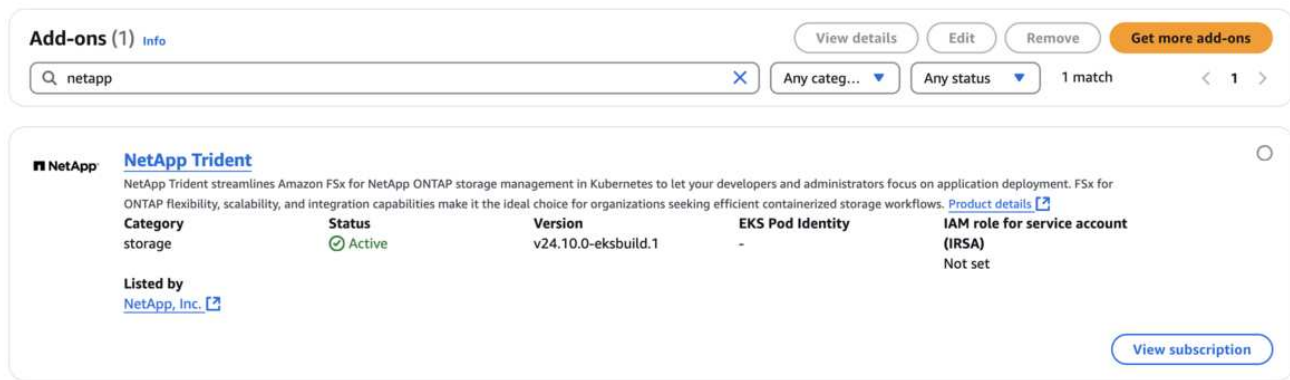
Cancel

Previous

Create

- Wenn Sie IRSA (IAM roles for service account) verwenden, beachten Sie die zusätzlichen Konfigurationsschritte ["hier"](#).
- Die Option **Create** wird ausgewählt.

9. Überprüfen Sie, ob der Status des Add-ons *Active* ist.



10. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Trident ordnungsgemäß auf dem Cluster installiert ist:

```
kubectl get pods -n trident
```

11. Fahren Sie mit der Einrichtung fort und konfigurieren Sie das Storage-Backend. Informationen finden sich unter "[Konfigurieren Sie das Storage-Backend](#)".

Installieren/Deinstallieren des Trident EKS Add-ons über die CLI

Installieren Sie das NetApp Trident EKS-Add-on mithilfe der CLI:

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 dedizierten Version)
```

Der folgende Beispielbefehl installiert das Trident EKS-Add-on Version 25.6.1:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.1-eksbuild.1 (mit einer dedizierten Version)
```

Der folgende Beispielbefehl installiert das Trident EKS-Add-on Version 25.6.2:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.2-eksbuild.1 (mit einer dedizierten Version)
```

Deinstallieren Sie das NetApp Trident EKS add-on über die CLI:

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 gibt Trident vor, wie Trident mit diesem Speichersystem kommuniziert und wie Trident Volumes daraus bereitstellen soll. Nachdem Trident installiert wurde, besteht der nächste Schritt darin, ein Backend zu erstellen. Die `TridentBackendConfig Custom Resource`

Definition (CRD) ermöglicht es Ihnen, Trident-Backends direkt über die Kubernetes-Oberfläche zu erstellen und zu verwalten. Sie können dies tun, indem Sie `kubectl` oder das entsprechende CLI-Tool für Ihre Kubernetes-Distribution verwenden.

TridentBackendConfig

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) ist ein Frontend mit Namespaces, das Ihnen ermöglicht, Trident-Backends mithilfe von `kubectl` zu verwalten. Kubernetes- und Speicheradministratoren können jetzt Backends direkt über die Kubernetes-CLI erstellen und verwalten, ohne ein spezielles Befehlszeilenprogramm zu benötigen (`tridentctl`).

Bei der Erstellung eines `TridentBackendConfig` Objekts geschieht Folgendes:

- Ein Backend wird von Trident automatisch auf Basis der von Ihnen bereitgestellten Konfiguration erstellt. Dies wird intern als `TridentBackend` (`tbe`, `tridentbackend`) CR dargestellt.
- Das `TridentBackendConfig` ist eindeutig an ein `TridentBackend` gebunden, das von Trident erstellt wurde.

Jedes `TridentBackendConfig` pflegt eine Eins-zu-Eins-Zuordnung zu 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 repräsentiert.

WARNUNG

`TridentBackend`-CRs werden automatisch von Trident erstellt. Sie **sollten** diese nicht bearbeiten. Wenn Sie Aktualisierungen an Backends vornehmen möchten, tun Sie dies, indem Sie das `TridentBackendConfig` Objekt bearbeiten.

Siehe das folgende Beispiel für 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 im "[trident-installer](#)" directory für Beispielfunktionen für die gewünschte Speicherplattform bzw. den gewünschten Speicherdienst ansehen.

Die `spec` nimmt backendspezifische Konfigurationsparameter entgegen. In diesem Beispiel verwendet das Backend den `ontap-san` Speichertreiber und verwendet die hier tabellarisch aufgeführten

Konfigurationsparameter. Eine Liste der Konfigurationsoptionen für Ihren gewünschten Speichertreiber finden Sie in der ["Backend-Konfigurationsinformationen für Ihren Speichertreiber"](#).

Der `spec` Abschnitt umfasst außerdem `credentials` und `deletionPolicy` Felder, die neu im `TridentBackendConfig` CR eingeführt wurden:

- `credentials`: Dieser Parameter ist ein Pflichtfeld und enthält die Anmeldeinformationen, die zur Authentifizierung beim Speichersystem/-dienst verwendet werden. Dies wird auf ein vom Benutzer erstelltes Kubernetes Secret gesetzt. Die Anmeldeinformationen dürfen nicht im Klartext übergeben werden und führen sonst zu einem Fehler.
- `deletionPolicy`: Dieses Feld definiert, was geschehen soll, wenn der `TridentBackendConfig` gelöscht wird. Es kann einen von zwei möglichen Werten annehmen:
 - `delete`: Dies führt zur Löschung sowohl des `TridentBackendConfig` CR als auch des zugehörigen Backends. Dies ist der Standardwert.
 - `retain`: Wenn ein `TridentBackendConfig` CR gelöscht wird, bleibt die Backend-Definition weiterhin vorhanden und kann mit `tridentctl` verwaltet werden. Das Festlegen der Löschrichtlinie auf `retain` ermöglicht es Benutzern, auf eine frühere Version (vor 21.04) downzugraden und die erstellten Backends beizubehalten. Der Wert für dieses Feld kann aktualisiert werden, nachdem ein `TridentBackendConfig` erstellt wurde.

HINWEIS

Der Name eines Backends wird mit `spec.backendName` festgelegt. Wenn nicht angegeben, wird der Name des Backends auf den Namen des `TridentBackendConfig` Objekts (`metadata.name`) gesetzt. Es wird empfohlen, Backend-Namen explizit mit `spec.backendName` festzulegen.

TIPP

Backends, die mit `tridentctl` erstellt wurden, haben kein zugeordnetes `TridentBackendConfig` Objekt. Sie können solche Backends mit `kubect1` verwalten, indem Sie eine `TridentBackendConfig` CR erstellen. Achten Sie darauf, identische Konfigurationsparameter (wie `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName` usw.) anzugeben. Trident bindet die neu erstellte `TridentBackendConfig` automatisch an das bereits vorhandene Backend.

Schrittübersicht

Um ein neues Backend mithilfe von `kubect1` zu erstellen, sollten Sie Folgendes tun:

1. Erstellen Sie ein ["Kubernetes Secret"](#). Das Geheimnis enthält die Anmeldeinformationen, die Trident benötigt, um mit dem Speichercluster/-dienst zu kommunizieren.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieses enthält spezifische Informationen zum Speicher-Cluster/-Dienst und verweist auf das im vorherigen Schritt erstellte Geheimnis.

Nachdem Sie ein Backend erstellt haben, können Sie dessen Status mithilfe von `kubect1 get tbc <tbc-name> -n <trident-namespace>` beobachten und weitere Details abrufen.

Schritt 1: Erstellen Sie ein Kubernetes-Secret

Erstellen Sie ein Secret, das die Zugangsdaten für das Backend enthält. Dies ist für jeden Speicherdienst bzw. jede Plattform einzigartig. 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 im Secret für jede Speicherplattform enthalten sein müssen:

Beschreibung der Secret Fields der Storage-Plattform	Geheimnis	Feldbeschreibung
Azure NetApp Files	clientID	Die Client-ID aus einer App-Registrierung
Element (NetApp HCI/SolidFire)	Endpunkt	MVIP für den SolidFire Cluster mit Mandantenanmeldeinformationen
ONTAP	Benutzername	Benutzername für die Verbindung zum Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet
ONTAP	Passwort	Passwort für die Verbindung zum Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet
ONTAP	clientPrivateKey	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für die zertifikatsbasierte Authentifizierung verwendet
ONTAP	chapUsername	Benutzername für eingehende Anfragen. Erforderlich, wenn useCHAP=true. Für ontap-san und ontap-san-economy
ONTAP	chapInitiatorSecret	CHAP-Initiatorgeheimnis. 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

Beschreibung der Secret Fields der Storage-Plattform	Geheimnis	Feldbeschreibung
ONTAP	chapTargetInitiatorSecret	CHAP-Zielinitiatorgeheimnis. Erforderlich, wenn useCHAP=true. Für ontap-san und ontap-san-economy

Das in diesem Schritt erstellte Geheimnis wird im `spec.credentials` Feld des `TridentBackendConfig` Objekts referenziert, das im nächsten Schritt erstellt wird.

Schritt 2: Erstellen Sie die `TridentBackendConfig` CR

Sie sind nun bereit, Ihre `TridentBackendConfig` CR zu erstellen. In diesem Beispiel wird ein Backend, das den `ontap-san` Treiber verwendet, mithilfe des unten gezeigten `TridentBackendConfig` Objekts erstellt:

```
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 der `TridentBackendConfig` CR

Nachdem Sie die `TridentBackendConfig` CR erstellt haben, können Sie den Status überprüfen. Siehe das folgende Beispiel:

```
kubectl -n trident get tbc backend-tbc-ontap-san
NAME                                BACKEND NAME                        BACKEND UUID
PHASE    STATUS
backend-tbc-ontap-san  ontap-san-backend  8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8          Bound              Success
```

Ein Backend wurde erfolgreich erstellt und an das `TridentBackendConfig` CR gebunden.

Die Phase kann einen der folgenden Werte annehmen:

- **Bound:** Der `TridentBackendConfig` CR ist mit einem Backend verknüpft, und dieses Backend enthält `configRef`, das auf die `TridentBackendConfig` UID des CR festgelegt ist.
- **Unbound:** Dargestellt durch `""`. Das `TridentBackendConfig` Objekt ist nicht an ein Backend gebunden. Alle neu erstellten `TridentBackendConfig` CRs befinden sich standardmäßig in dieser Phase. Nach dem Phasenwechsel kann es nicht wieder in den Zustand `Unbound` zurückversetzt werden.
- **Deleting:** Der `TridentBackendConfig` CR `deletionPolicy` war zum Löschen markiert. Wenn der `TridentBackendConfig` CR gelöscht wird, wechselt er in den Status „Wird gelöscht“.
 - Wenn auf dem Backend keine persistenten Volumenansprüche (PVCs) vorhanden sind, führt das Löschen des `TridentBackendConfig` dazu, dass Trident sowohl das Backend als auch den `TridentBackendConfig` CR löscht.
 - Sind ein oder mehrere PVCs im Backend vorhanden, wechselt es in den Löschstadium. Die `TridentBackendConfig` CR wechselt anschließend ebenfalls in die Löschphase. Das Backend und `TridentBackendConfig` werden erst gelöscht, nachdem alle PVCs gelöscht wurden.
- **Lost:** Das mit dem `TridentBackendConfig` CR verknüpfte Backend wurde versehentlich oder absichtlich gelöscht und der `TridentBackendConfig` CR enthält weiterhin eine Referenz auf das gelöschte Backend. Der `TridentBackendConfig` CR kann weiterhin gelöscht werden, unabhängig vom `deletionPolicy` Wert.
- **Unknown:** Trident kann den Status oder die Existenz des mit der `TridentBackendConfig` CR verknüpften Backends nicht feststellen. Zum Beispiel, wenn der API-Server nicht antwortet oder die `tridentbackends.trident.netapp.io` CRD fehlt. Dies könnte einen Eingriff erfordern.

In diesem Stadium ist ein Backend erfolgreich erstellt worden! Es gibt mehrere Operationen, die zusätzlich durchgeführt werden können, wie "[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	PHASE	STATUS	STORAGE DRIVER	BACKEND NAME	DELETION POLICY	BACKEND UUID
backend-tbc-ontap-san				ontap-san-backend		8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8		Bound	Success	ontap-san		delete

Zusätzlich können Sie auch einen YAML/JSON-Dump von `TridentBackendConfig` erhalten.

```
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 die backendName und die backendUUID des Backends, das als Reaktion auf die TridentBackendConfig CR erstellt wurde. Das lastOperationStatus Feld repräsentiert den Status der letzten Operation der TridentBackendConfig CR, die entweder vom Benutzer ausgelöst wurde (zum Beispiel, wenn der Benutzer etwas in spec geändert hat) oder von Trident ausgelöst wurde (zum Beispiel während Trident-Neustarts). Es kann entweder Erfolgreich oder Fehlgeschlagen sein. phase repräsentiert den Status der Beziehung zwischen der TridentBackendConfig CR und dem Backend. Im obigen Beispiel phase hat den Wert Bound, was bedeutet, dass die TridentBackendConfig CR mit dem Backend verknüpft ist.

Sie können den `kubectl -n trident describe tbc <tbc-cr-name>` Befehl ausführen, um Details zu den Ereignisprotokollen zu erhalten.

WARNUNG

Sie können ein Backend, das ein zugehöriges TridentBackendConfig Objekt enthält, nicht mit `tridentctl` aktualisieren oder löschen. Um die Schritte zum Wechsel zwischen `tridentctl` und TridentBackendConfig zu verstehen, "[Siehe hier](#)".

Backends verwalten

Trident Backends mit kubectl verwalten

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge mithilfe von `kubectl` durchführen.

Ein Backend löschen

Durch das Löschen eines `TridentBackendConfig` weisen Sie Trident an, Backends zu löschen oder beizubehalten (basierend auf `deletionPolicy`). Um ein Backend zu löschen, stellen Sie sicher, dass `deletionPolicy` auf „delete“ gesetzt ist. Um nur das `TridentBackendConfig` zu löschen, stellen Sie sicher, dass `deletionPolicy` auf „retain“ gesetzt ist. Dadurch wird sichergestellt, dass das Backend weiterhin vorhanden ist und mit `tridentctl` verwaltet werden kann.

Führen Sie den folgenden Befehl aus:

```
kubectl delete tbc <tbc-name> -n trident
```

Trident löscht die Kubernetes-Secrets, die von `TridentBackendConfig` verwendet wurden, nicht. Der Kubernetes-Benutzer ist für die Bereinigung der Secrets verantwortlich. Beim Löschen von Secrets ist Vorsicht geboten. Sie sollten Secrets nur löschen, wenn sie von den Backends nicht verwendet werden.

Vorhandene Backends anzeigen

Führen Sie den folgenden Befehl aus:

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

Sie können auch `tridentctl get backend -n trident` oder `tridentctl get backend -o yaml -n trident` ausführen, um eine Liste aller vorhandenen Backends zu erhalten. Diese Liste enthält auch Backends, die mit `tridentctl` erstellt wurden.

Aktualisieren Sie ein Backend

Es kann mehrere Gründe geben, ein Backend zu aktualisieren:

- Die Zugangsdaten für das Speichersystem haben sich geändert. Um die Zugangsdaten zu aktualisieren, muss das Kubernetes-Secret, das im `TridentBackendConfig` Objekt verwendet wird, aktualisiert werden. Trident aktualisiert das Backend automatisch mit den neuesten Zugangsdaten. Führen Sie den 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 `TridentBackendConfig` Objekte direkt über Kubernetes mit dem folgenden Befehl aktualisieren:

```
kubectl apply -f <updated-backend-file.yaml>
```

- Alternativ können Sie Änderungen an der bestehenden `TridentBackendConfig` CR mit dem folgenden Befehl vornehmen:

```
kubectl edit tbc <tbc-name> -n trident
```

HINWEIS

- Schlägt ein Backend-Update fehl, bleibt das Backend in seiner zuletzt bekannten Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie `kubectl get tbc <tbc-name> -o yaml -n trident` oder `kubectl describe tbc <tbc-name> -n trident` ausführen.
- Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Aktualisierungsbefehl erneut ausführen.

Trident Backends mit tridentctl verwalten

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge mithilfe von `tridentctl` durchführen.

Erstelle ein Backend

Nachdem Sie eine ["Konfigurationsdatei für das Backend"](#) erstellt haben, führen Sie den 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 den folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie einfach den `create` Befehl erneut ausführen.

Ein 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
```

HINWEIS

Falls Trident Volumes und Snapshots von diesem Backend bereitgestellt hat, die noch

existieren, verhindert das Löschen des Backends, dass neue Volumes von ihm bereitgestellt werden. Das Backend verbleibt im Status „Wird gelöscht“.

Vorhandene Backends anzeigen

Um die von Trident bekannten Backends anzuzeigen, gehen Sie wie folgt vor:

- Um eine Zusammenfassung zu erhalten, führen Sie den folgenden Befehl aus:

```
tridentctl get backend -n trident
```

- Um alle Details zu erhalten, führen Sie den folgenden Befehl aus:

```
tridentctl get backend -o json -n trident
```

Aktualisieren Sie ein Backend

Nachdem Sie eine neue Backend-Konfigurationsdatei erstellt haben, führen Sie den folgenden Befehl aus:

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Wenn die Backend-Aktualisierung fehlschlägt, lag ein Fehler in der Backend-Konfiguration vor oder Sie haben eine ungültige Aktualisierung versucht. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie den folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie einfach den `update` Befehl erneut ausführen.

Identifizieren Sie die Speicherklassen, die ein Backend verwenden

Dies ist ein Beispiel für die Art von Fragen, die Sie mit dem JSON, das `tridentctl` für Backend-Objekte ausgibt, beantworten können. Dies verwendet das `jq` Hilfsprogramm, das Sie installieren müssen.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

Dies gilt auch für Backends, die mit `TridentBackendConfig` erstellt wurden.

Auswahl einer Methode zur Verwaltung des Trident Backends

Erfahren Sie mehr über die verschiedenen Möglichkeiten, Backends in Trident zu verwalten.

Optionen zur Verwaltung von Backends

Mit der Einführung von `TridentBackendConfig` haben Administratoren nun zwei einzigartige Möglichkeiten, Backends zu verwalten. Dies wirft folgende Fragen auf:

- Können Backends, die mit `tridentctl` erstellt wurden, mit `TridentBackendConfig` verwaltet werden?
- Können Backends, die mit `TridentBackendConfig` erstellt wurden, mit `tridentctl` verwaltet werden?

Backends mit `tridentctl` verwalten unter Verwendung von `TridentBackendConfig`

Dieser Abschnitt beschreibt die Schritte, die erforderlich sind, um Backends zu verwalten, die mit `tridentctl` direkt über die Kubernetes-Schnittstelle durch das Erstellen von `TridentBackendConfig` Objekten erstellt wurden.

Dies gilt für die folgenden Szenarien:

- Vorhandene Backends, die kein `TridentBackendConfig` haben, weil sie mit `tridentctl` erstellt wurden.
- Neue Backends, die mit `tridentctl` erstellt wurden, während andere `TridentBackendConfig` Objekte existieren.

In beiden Szenarien bleiben die Backends weiterhin vorhanden, wobei Trident die Volumes plant und auf ihnen arbeitet. Administratoren haben hier zwei Möglichkeiten:

- Verwenden Sie `tridentctl` weiterhin, um Backends zu verwalten, die damit erstellt wurden.
- Binden Sie Backends, die mit `tridentctl` erstellt wurden, an ein neues `TridentBackendConfig` Objekt. Dadurch werden die Backends mit `kubectl` und nicht mit `tridentctl` verwaltet.

Um ein bestehendes Backend mit `kubectl` zu verwalten, müssen Sie eine `TridentBackendConfig` erstellen, die an das bestehende Backend gebunden ist. Hier ist eine Übersicht, wie das funktioniert:

1. Erstellen Sie ein Kubernetes-Secret. Das Secret enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster bzw. -dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieses enthält spezifische Informationen zum Speicher-Cluster/-Dienst und verweist auf das im vorherigen Schritt erstellte Geheimnis. Achten Sie darauf, identische Konfigurationsparameter anzugeben (wie `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName` usw.). `spec.backendName` muss auf den Namen des vorhandenen Backends gesetzt werden.

Schritt 0: Backend identifizieren

Um ein `TridentBackendConfig` zu erstellen, das an ein bestehendes Backend gebunden ist, müssen Sie die Backend-Konfiguration abrufen. In diesem Beispiel nehmen wir an, 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 Secret, 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 einen `TridentBackendConfig` CR

Der nächste Schritt besteht darin, eine `TridentBackendConfig` CR zu erstellen, die automatisch an die bereits vorhandene `ontap-nas-backend` gebunden wird (wie in diesem Beispiel). Stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Der gleiche Backend-Name ist definiert in `spec.backendName`.
- Die Konfigurationsparameter sind identisch mit dem ursprünglichen Backend.
- Virtuelle Pools (sofern vorhanden) müssen die gleiche Reihenfolge wie im ursprünglichen Backend beibehalten.
- Anmeldeinformationen werden über ein Kubernetes Secret und nicht im Klartext bereitgestellt.

In diesem Fall wird die `TridentBackendConfig` 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: mysqldb
      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 der `TridentBackendConfig` CR

Nachdem das `TridentBackendConfig` erstellt wurde, muss seine Phase Bound sein. 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
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
ontap-nas-backend	ontap-nas	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7
online	25	

Das Backend wird nun vollständig über das `tbc-ontap-nas-backend` `TridentBackendConfig` Objekt verwaltet.

Backends mit `TridentBackendConfig` verwalten unter Verwendung von `tridentctl`

`tridentctl` kann verwendet werden, um Backends aufzulisten, die mit `TridentBackendConfig` erstellt wurden. Darüber hinaus können Administratoren solche Backends auch vollständig über `tridentctl` verwalten, indem sie `TridentBackendConfig` löschen und sicherstellen, dass `spec.deletionPolicy` auf `retain` gesetzt ist.

Schritt 0: Backend identifizieren

Nehmen wir beispielsweise an, das folgende Backend wurde mit `TridentBackendConfig` erstellt:

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82
Bound	Success	

PHASE	STATUS	STORAGE DRIVER	DELETION POLICY
		ontap-san	delete

```
tridentctl get backend ontap-san-backend -n trident
```

NAME	STORAGE DRIVER	UUID
ontap-san-backend	ontap-san	81abcb27-ea63-49bb-b606-0a5315ac5f82

```
tridentctl get backend ontap-san-backend -n trident
```

NAME	STORAGE DRIVER	UUID
ontap-san-backend	ontap-san	81abcb27-ea63-49bb-b606-0a5315ac5f82

Aus der Ausgabe geht hervor, dass `TridentBackendConfig` erfolgreich erstellt wurde und mit einem Backend verbunden ist [beachten Sie die UUID des Backends].

Schritt 1: Bestätigen Sie, dass `deletionPolicy` auf `retain` gesetzt ist

Betrachten wir den Wert von `deletionPolicy`. Dieser muss auf `retain` gesetzt werden. Dadurch wird sichergestellt, dass beim Löschen eines `TridentBackendConfig` CR die Backend-Definition weiterhin vorhanden ist und mit `tridentctl` verwaltet werden kann.

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82

```
# 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
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82

HINWEIS

Fahren Sie nicht mit dem nächsten Schritt fort, es sei denn, `deletionPolicy` ist auf `retain` gesetzt.

Schritt 2: Löschen Sie das TridentBackendConfig CR

Der letzte Schritt besteht darin, die `TridentBackendConfig` CR zu löschen. Nachdem Sie bestätigt haben, dass das `deletionPolicy` auf `retain` gesetzt ist, können Sie mit dem Löschen fortfahren:

```
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 |                               |
+-----+-----+-----+-----+
```

Beim Löschen des `TridentBackendConfig` Objekts entfernt Trident dieses einfach, ohne das Backend selbst zu löschen.

Speicherklassen erstellen und verwalten

Erstellen einer Trident Storage Klasse

Konfigurieren Sie ein Kubernetes-StorageClass-Objekt und erstellen Sie die Storage-Klasse, um Trident anzuweisen, wie Volumes bereitgestellt werden.

Konfigurieren eines Kubernetes StorageClass Objekts

Das "[Kubernetes StorageClass-Objekt](#)" identifiziert Trident als den Provisioner, der für diese Klasse verwendet wird, und weist Trident an, wie ein Volume bereitgestellt werden soll. 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](#)" für Details dazu, wie Storage-Klassen mit dem PersistentVolumeClaim interagieren und Parameter zur Steuerung, wie Trident Volumes bereitstellt.

Erstellen Sie eine Speicherklasse

Nachdem Sie das StorageClass-Objekt erstellt haben, können Sie die Speicherklasse erstellen. [Storage-Class-Beispiele](#) bietet einige grundlegende Beispiele, die Sie verwenden oder anpassen können.

Schritte

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

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

2. Sie sollten nun eine **basic-csi** Storage-Class sowohl in Kubernetes als auch in Trident 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"
    ]
}
}
]
}

```

Storage-Class-Beispiele

Trident bietet "einfache Speicherklassendefinitionen für spezifische Backends".

Alternativ können Sie die `sample-input/storage-class-csi.yaml.template` Datei, die mit dem Installationsprogramm geliefert wird, bearbeiten und `BACKEND_TYPE` durch den Namen des Speichertreibers ersetzen.

```

./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.template 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

```

Verwalten von Trident Storage Klassen

Sie können vorhandene Speicherklassen anzeigen, eine Standard-Speicherklasse festlegen, die Speicherklassen-Backends identifizieren und Speicherklassen löschen.

Vorhandene Speicherklassen anzeigen

- Um vorhandene Kubernetes-Speicherklassen anzuzeigen, führen Sie den folgenden Befehl aus:

```
kubectl get storageclass
```

- Um Details zur Kubernetes-Speicherklasse anzuzeigen, führen Sie den folgenden Befehl aus:

```
kubectl get storageclass <storage-class> -o json
```

- Um die synchronisierten Speicherklassen von Trident anzuzeigen, führen Sie den folgenden Befehl aus:

```
tridentctl get storageclass
```

- Um die Details der synchronisierten Speicherklasse von Trident anzuzeigen, führen Sie den folgenden Befehl aus:

```
tridentctl get storageclass <storage-class> -o json
```

Legen Sie eine Standardspeicherklasse fest

Kubernetes 1.6 führte die Möglichkeit ein, eine Standard-Speicherklasse festzulegen. Dies ist die Speicherklasse, die für die Bereitstellung eines Persistent Volume verwendet wird, wenn ein Benutzer in einem Persistent Volume Claim (PVC) keine angibt.

- Definieren Sie eine Standard-Speicherklasse, indem Sie die Annotation `storageclass.kubernetes.io/is-default-class` auf `true` in der Speicherklassendefinition setzen. Gemäß der Spezifikation wird jeder andere Wert oder das Fehlen der Annotation als `false` 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 dem folgenden 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 enthalten.

HINWEIS

Es sollte immer nur eine Standard-Speicherklasse in Ihrem Cluster vorhanden sein. Kubernetes verhindert dies technisch zwar nicht, aber es verhält sich so, als gäbe es überhaupt keine Standard-Speicherklasse.

Identifizieren Sie das Backend für eine Storage Class

Dies ist ein Beispiel für die Art von Fragen, die Sie mit dem JSON, das `tridentctl` für Trident Backend-Objekte ausgibt, beantworten können. Dies verwendet das `jq` Hilfsprogramm, das Sie möglicherweise zuerst installieren müssen.

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass:
.Config.name, backends: [.storage]|unique}]'
```

Eine Storage Class löschen

Um eine Speicherklasse aus Kubernetes zu löschen, führen Sie den 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 Trident wird sie weiterhin verwalten.

HINWEIS

Trident erzwingt ein `fsType` für die von ihm erstellten Volumes. Für iSCSI-Backends wird empfohlen, `parameters.fsType` in der StorageClass zu erzwingen. Sie sollten vorhandene StorageClasses löschen und sie mit `parameters.fsType` neu erstellen.

Volumes bereitstellen und verwalten

Bereitstellung eines Trident Volume

Erstellen Sie einen PersistentVolumeClaim (PVC), der die konfigurierte Kubernetes-StorageClass verwendet, um Zugriff auf das PV anzufordern. Anschließend können Sie das PV in einen Pod einbinden.

Überblick

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

Die PVC kann so konfiguriert werden, dass sie Speicherplatz einer bestimmten Größe oder eines bestimmten Zugriffsmodus anfordert. Mithilfe der zugehörigen StorageClass kann der Clusteradministrator mehr als nur PersistentVolume-Größe und Zugriffsmodus steuern – zum Beispiel Leistung oder Servicelevel.

Nachdem Sie das PVC erstellt haben, können Sie das Volume in einem Pod einbinden.

PVC erstellen

Schritte

1. Die PVC wird erstellt.

```
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. Mountieren Sie das Volume in einem Pod.

```
kubectl create -f pv-pod.yaml
```

HINWEIS

Sie können den Fortschritt mit `kubectl get pod --watch` überwachen.

2. Überprüfen Sie, ob das Volume auf `/my/mount/path` eingebunden ist.

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

3. Sie können den Pod nun löschen. Die Pod-Anwendung existiert dann nicht mehr, aber das Volume bleibt erhalten.

```
kubectl delete pod pv-pod
```

Beispielmanifeste

▼ PersistentVolumeClaim-Beispielmanifeste

Diese Beispiele zeigen grundlegende PVC-Konfigurationsoptionen.

PVC mit RWO-Zugriff

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

```
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 protection-gold zugeordnet ist.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

▼ Pod-Manifestbeispiele

Diese Beispiele zeigen grundlegende Konfigurationen, um das PVC an ein Pod anzuhängen.

Grundkonfiguration

```
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"](#) für Details dazu, wie Storage-Klassen mit dem PersistentVolumeClaim interagieren und Parameter zur Steuerung, wie Trident Volumes bereitstellt.

Ein Volume verschieben

Ab Trident 26.06 kann ein Speicheradministrator ein iSCSI-Volume unterbrechungsfrei auf einen anderen ONTAP-Knoten und ein anderes Aggregat innerhalb desselben ONTAP Clusters verschieben, während die darauf basierende Arbeitslast weiterläuft. Die Verschiebung wird deklarativ mit einer `TridentVolumeMove` benutzerdefinierten Ressource oder dem `tridentctl` Äquivalent angefordert, und Trident koordiniert ONTAP und den Host, sodass die I/O niemals unterbrochen wird.

Anforderungen

Volume Move gilt nur für den `ontap-san` Treiber für iSCSI-Volumes. NVMe und NAS Protokolle werden nicht unterstützt.

Bevor Sie ein Volume verschieben, muss sichergestellt sein, dass die folgenden Voraussetzungen erfüllt sind:

- Das Backend verwendet den `ontap-san` Treiber mit der ONTAP REST API.
- Der ONTAP Cluster verfügt über mindestens zwei Knoten.
- Sie verfügen über einen der folgenden Berechtigungssätze:
 - ONTAP 9.19.1 oder höher, wenn die dedizierte benutzerdefinierte ONTAP Rolle verwendet wird.
 - Cluster-Administrator-Anmeldeinformationen in früheren ONTAP Releases.

So funktioniert es

Trident verschiebt das Volume (LUN) zwischen zwei Standorten innerhalb desselben ONTAP Clusters. Bevor die Daten verschoben werden, fügt Trident dem Ziel neben den bestehenden Pfaden neue iSCSI-Pfade hinzu, sodass das Multipath-Gerät des Hosts jederzeit erreichbar bleibt. Nachdem ONTAP die Verschiebung abgeschlossen hat, entfernt Trident die veralteten Quellpfade.

Trident steuert den Vorgang durch eine geordnete Reihe von Zuständen, die in der `TridentVolumeMove` Ressource sichtbar sind, sodass der Fortschritt des Vorgangs verfolgt werden kann.

Ein Volume verschieben

Das Volumen und die Standorte identifizieren

1. Den Namen des zu verschiebenden PV ermitteln. Der PV-Name ist auch der Name der Move-Ressource.
2. Im ONTAP Cluster den aktuellen (Quell-)Knoten und das Aggregat für dieses Volume notieren und den Zielknoten sowie das Ziel-Aggregat auswählen.

Starten Sie den Umzug

Beide Methoden können verwendet werden. Beide erstellen dieselbe `TridentVolumeMove` Ressource, deren Name der PV-Name ist.

Verwendung von tridentctl

```
tridentctl create volume-move \  
  --volume      pvc-8a814d62-bd58-4253-b0d1-82f2885db671 \  
  --source-node ontap-node-01 \  
  --source-pool aggr1_node01 \  
  --target-node ontap-node-02 \  
  --target-pool aggr1_node02 \  
  -n trident
```

tvm ist ein Alias für volume-move. --delete-after-success 10m hinzufügen, damit die Ressource nach erfolgreichem Verschieben automatisch entfernt wird.

Verwendung einer TridentVolumeMove Ressource

```
apiVersion: trident.netapp.io/v1  
kind: TridentVolumeMove  
metadata:  
  name: pvc-8a814d62-bd58-4253-b0d1-82f2885db671    # the PV to move  
  namespace: trident  
spec:  
  sourceNode: ontap-node-01  
  sourcePool: aggr1_node01  
  targetNode: ontap-node-02  
  targetPool: aggr1_node02  
  # deleteAfterSuccess: 10m    # optional
```

Ressource anwenden:

```
kubectl apply -f tridentvolumemove.yaml
```

Spezifikationsfelder

sourceNode / targetNode

ONTAP Knotennamen (nicht UUIDs).

sourcePool / targetPool

ONTAP Aggregatnamen.

deleteAfterSuccess

Optionale Dauer, wie 30s oder 10m. Auf 0 setzen, um die Ressource sofort zu löschen, oder weglassen, um die Ressource beizubehalten.

Die Verschiebung überwachen

Der Fortschritt kann anhand des `tvm`Kurznamens` oder ``tridentvolumemove` verfolgt werden:

```
kubectl get tridentvolumemove -n trident
```

Der aktuelle Status und der Fortschritt pro Knoten werden angezeigt, wenn die Ressource beschrieben wird:

```
kubectl describe tridentvolumemove <pv-name> -n trident
```

Das Volume kann während des `Moving`-Zustands mehrere gleichzeitige Anhänge haben, da ONTAP für die Verschiebung eines großen Volumes einige Zeit benötigen kann. Für eine reibungslose Verschiebung koordiniert Trident die Abläufe zwischen dem Trident Controller und dem Trident Node. Jeder Zustand führt die folgenden Schritte in der angegebenen Reihenfolge aus und endet entweder in `Succeeded` oder `Failed`.

Status	Was geschieht
Pending	Trident akzeptiert und validiert die Umzugsanfrage und sammelt die benötigten Informationen, bevor Änderungen am Speicher vorgenommen werden.
ControllerStaging	Trident bereitet den Zielort vor. Es ordnet die Speicherschnittstellen des Zielknotens der LUN zu und aktualisiert Kubernetes, sodass ein zukünftiger Neustart des Knotens die Verbindung zum richtigen Standort wiederherstellt.
NodeStaging	Auf jedem Host, auf dem das Volume eingebunden ist, fügt Trident die neuen Speicherpfade vom Zielknoten neben den bestehenden Pfaden hinzu, sodass die E/A des Pods über beide Pfadsätze hinweg fortgesetzt wird.
Moving	ONTAP kopiert und überträgt die Volumendaten auf das Zielaggregat. Dies ist die längste Phase, und ONTAP übernimmt sie vollständig, während Trident den Vorgang abfragt, bis er abgeschlossen ist. Neue Veröffentlichungen sind während dieses Status zulässig.
ControllerUnstaging	Die Datenübertragung ist abgeschlossen. Trident aktualisiert Kubernetes, um die Quellpfade aus dem gewünschten Zustand zu entfernen, und hebt anschließend die Zuordnung der Speicherschnittstellen des Quellknotens von der LUN auf.

Status	Was geschieht
NodeUnstaging	Auf jedem Host entfernt Trident die veralteten Quellpfade. Dem Host verbleiben nur noch die Zielpfade, und das Multipath-Gerät verarbeitet weiterhin E/A ohne Unterbrechung.
Succeeded	Das Volume ist vollständig verschoben und am Ziel online. Trident behält die <code>TridentVolumeMove</code> Ressource bei oder löscht sie automatisch, wenn Sie <code>deleteAfterSuccess</code> festlegen.
Failed	Ein nicht behebbarer Fehler hat den Verschiebevorgang gestoppt oder dessen Start verhindert. Trident setzt den Volume-Status zurück, sodass der normale Betrieb fortgesetzt werden kann, und behält die Ressource mit einer Fehlermeldung. Die Meldung sollte überprüft, die Ursache behoben, die Ressource gelöscht und der Vorgang erneut versucht werden.

Die `.status.attachments``Liste zeigt den Fortschritt jedes angeschlossenen Knotens in seinem ``state``Feld:

Anhangsstatus	Bedeutung
Pending	Knoten als verbunden identifiziert; die Arbeit hat noch nicht begonnen.
Bridged	Neue Zielpfade werden neben den bestehenden Pfaden hinzugefügt.
Migrated	Der Datenverkehr wird jetzt über die neuen Pfade bereitgestellt.
Cleaned	Veraltete Quellpfade vom Knoten entfernt.
Detached	Der Knoten wurde während der Bewegung abgetrennt und von der weiteren Betrachtung ausgeschlossen, während Trident prüft, ob die Verbindung bereit ist, mit dem nächsten Schritt fortzufahren.
Failed	Terminalfehler auf diesem Knoten. Die Meldung anzeigen.

Die Umzugsressource bereinigen

Wenn `deleteAfterSuccess` festgelegt ist, entfernt Trident die Ressource automatisch. Andernfalls kann sie manuell gelöscht werden. Bevor eine weitere Verschiebung auf demselben Volume angefordert wird, ist sicherzustellen, dass für dieses Volume keine `TridentVolumeMove` Ressource im `Succeeded`- oder `Failed`-Status vorhanden ist.

HINWEIS

Trident löscht fehlgeschlagene Verschiebungen niemals automatisch. Trident speichert eine `Failed` Ressource als Prüfprotokoll. Nachdem die Ursache behoben wurde, sollte die Ressource manuell gelöscht werden, bevor ein erneuter Versuch erfolgt.

Trident Volumes erweitern

Trident ermöglicht Kubernetes-Nutzern die nachträgliche Erweiterung ihrer Volumes. Finden Sie Informationen zu den erforderlichen Konfigurationen, um iSCSI-, NFS-, SMB-, NVMe/TCP- und FC-Volumes zu erweitern.

Ein iSCSI-Volume erweitern

Sie können ein iSCSI Persistent Volume (PV) mithilfe des CSI provisioner erweitern.

HINWEIS

Die iSCSI-Volume-Erweiterung wird von den `ontap-san`, `ontap-san-economy`, `solidfire-san` Treibern unterstützt und erfordert Kubernetes 1.16 oder höher.

Schritt 1: Konfigurieren Sie die StorageClass, um die Volumenerweiterung zu unterstützen

Bearbeiten Sie die StorageClass-Definition, um das `allowVolumeExpansion` Feld auf `true` zu setzen.

```
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 bestehende StorageClass, um den `allowVolumeExpansion` Parameter hinzuzufügen.

Schritt 2: Erstellen Sie eine PVC mit der StorageClass, die Sie erstellt haben

Bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage`, um die neu gewünschte Größe anzugeben, 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 ordnet es diesem Persistent Volume Claim (PVC) zu.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY
san-pvc	Bound	pvc-8a814d62-bd58-4253-b0d1-82f2885db671	1Gi

```
kubectl get pv
```

NAME	RECLAIM POLICY	STATUS	CLAIM	CAPACITY	ACCESS MODES	AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671	Delete	Bound	default/san-pvc	1Gi	RWO	10s

Schritt 3: Definieren Sie ein Pod, das das PVC anbindet

Befestigen Sie das PV an einem Pod, damit es in der Größe angepasst werden kann. Es gibt zwei Szenarien, wenn die Größe eines iSCSI PV geändert wird:

- Wenn das PV an einen Pod angehängt ist, erweitert Trident das Volume auf dem Storage-Backend, scannt das Gerät erneut und passt die Größe des Dateisystems an.
- Beim Versuch, ein nicht zugeordnetes PV zu vergrößern, erweitert Trident das Volume im Storage-Backend. Nachdem das PVC an einen Pod gebunden wurde, scannt Trident das Gerät erneut und passt die Größe des Dateisystems an. Kubernetes aktualisiert dann die PVC-Größe, nachdem der Erweiterungsvorgang erfolgreich abgeschlossen wurde.

In diesem Beispiel wird ein Pod erstellt, der die `san-pvc` verwendet.

```
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 1Gi auf 2Gi zu ändern, bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage` auf 2Gi.

```
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: Validieren Sie die Erweiterung

Sie können überprüfen, ob die Erweiterung korrekt funktioniert hat, indem Sie die Größe des PVC, PV und des Trident volume 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 |
+-----+-----+-----+-----+
|          BACKEND UUID  | STATE | MANAGED  |
+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san |
| block      | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Ein FC-Volume erweitern

Sie können ein FC Persistent Volume (PV) mithilfe des CSI provisioner erweitern.

HINWEIS

FC-Volume-Erweiterung wird vom `ontap-san` Treiber unterstützt und erfordert Kubernetes 1.16 und höher.

Schritt 1: Konfigurieren Sie die StorageClass, um die Volumenerweiterung zu unterstützen

Bearbeiten Sie die StorageClass-Definition, um das `allowVolumeExpansion` Feld auf `true` zu setzen.

```
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 bestehende StorageClass, um den `allowVolumeExpansion` Parameter

hinzuzufügen.

Schritt 2: Erstellen Sie eine PVC mit der StorageClass, die Sie erstellt haben

Bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage`, um die neu gewünschte Größe anzugeben, 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 ordnet es diesem Persistent Volume Claim (PVC) zu.

```
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 ein Pod, das das PVC anbindet

Befestigen Sie das PV an einem Pod, damit es in der Größe angepasst werden kann. Es gibt zwei Szenarien, wenn ein FC-PV in der Größe geändert wird:

- Wenn das PV an einen Pod angehängt ist, erweitert Trident das Volume auf dem Storage-Backend, scannt das Gerät erneut und passt die Größe des Dateisystems an.
- Beim Versuch, ein nicht zugeordnetes PV zu vergrößern, erweitert Trident das Volume im Storage-Backend. Nachdem das PVC an einen Pod gebunden wurde, scannt Trident das Gerät erneut und passt die Größe des Dateisystems an. Kubernetes aktualisiert dann die PVC-Größe, nachdem der

Erweiterungsvorgang erfolgreich abgeschlossen wurde.

In diesem Beispiel wird ein Pod erstellt, der die `san-pvc` verwendet.

```
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 1Gi auf 2Gi zu ändern, bearbeiten Sie die PVC-Definition und aktualisieren Sie die `spec.resources.requests.storage` auf 2Gi.

```
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: Validieren Sie die Erweiterung

Sie können überprüfen, ob die Erweiterung korrekt funktioniert hat, indem Sie die Größe des PVC, PV und des Trident volume 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
tridentctl get volumes -n trident
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE  | STORAGE CLASS |
+-----+-----+-----+-----+
|          BACKEND UUID  | STATE | MANAGED  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san |
| block      | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Ein NFS-Volume erweitern

Trident unterstützt die Volume-Erweiterung für NFS PVs, die auf `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup` und `azure-netapp-files` Backends bereitgestellt werden.

Schritt 1: Konfigurieren Sie die StorageClass, um die Volumenerweiterung zu unterstützen

Um die Größe eines NFS-PV zu ändern, muss der Administrator zunächst die Speicherklasse so konfigurieren, dass eine Volumenerweiterung möglich ist, indem er das `allowVolumeExpansion` Feld auf `true` setzt:

```
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
```

Falls Sie bereits eine Speicherklasse ohne diese Option erstellt haben, können Sie die bestehende Speicherklasse einfach bearbeiten, indem Sie `kubectl edit storageclass` verwenden, um die Volumenerweiterung zu ermöglichen.

Schritt 2: Erstellen Sie eine PVC mit der StorageClass, die Sie erstellt haben

```
cat pvc-ontapnas.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Trident sollte eine 20 MiB NFS PV für diese PVC 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 setzen Sie `spec.resources.requests.storage` auf 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: Validieren Sie die Erweiterung

Sie können validieren, ob die Größenänderung korrekt funktioniert hat, indem Sie die Größe des PVC, PV und des Trident-Volumes 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 |
+-----+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
+-----+-----+-----+-----+
| file          | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Grenzen des RWX NVMe-Subsystems verstehen

ReadWriteMany (RWX)-Volumes, die das NVMe-Protokoll verwenden, haben eine Skalierbarkeitsgrenze von 64 Knoten pro Volume. Im Folgenden werden die Einschränkungen erläutert, die Architektur des beteiligten NVMe-Subsystems erklärt und die erforderlichen Schritte zur Behebung aufgezeigt.

Verstehen Sie das 64-Knoten-Limit

Wenn Sie planen, ReadWriteMany (RWX)-Volumes mit dem NVMe-Protokoll zu verwenden, kann ein einzelnes RWX-NVMe-Volume von nicht mehr als 64 Knoten in einem Kubernetes-Cluster eingebunden werden.

Planen Sie keine Workloads ein, die dieselbe RWX NVMe PersistentVolumeClaim auf mehr als 64 Knoten einbinden.

Diese Einschränkung gilt nur für RWX-Volumes, die das NVMe-Protokoll verwenden.

NVMe-Subsystemmodelle verstehen

Subsystemmodell pro Volumen (Trident Releases früher als 26.02)

In Trident-Versionen vor 26.02 werden RWX-NVMe-Volumes mithilfe eines Subsystemmodells pro Volume bereitgestellt. Jedes RWX-NVMe-Volume ist seinem eigenen dedizierten NVMe-Subsystem auf ONTAP zugeordnet.

Dieses Modell ist einfach, hat aber eine geringere Skalierbarkeitsgrenze. In großen Kubernetes-Clustern werden die Grenzen der Subsystem-Controller schnell erreicht, da jedes RWX-Volume ein eigenes Subsystem belegt.

Super-Subsystem-Modell (eingeführt in Trident 26.02)

Ab Trident 26.02 verwenden RWX NVMe-Volumes ein gemeinsames Super-Subsystem-Modell. Mehrere RWX NVMe-Volumes teilen sich dasselbe NVMe-Subsystem.

Jedes Super-Subsystem unterstützt bis zu 1024 Namensräume (Volumes). Dieses Modell verbessert die Skalierbarkeit für RWX-Workloads erheblich und verringert die Wahrscheinlichkeit, die ONTAP-Subsystem-Grenzen zu erreichen.

Jedes RWX NVMe-Volume unterstützt bis zu 64 Knoten.

Fehlersymptome identifizieren

Wenn Sie RWX NVMe-Volumes in großem Umfang erstellen oder einbinden, können Fehler wie die folgenden auftreten:

```
Maximum number of controllers reached. No more controllers can be created.
```

Dieser Fehler weist darauf hin, dass das ONTAP NVMe-Subsystem-Controller-Limit erreicht wurde.

Beheben Sie Subsystem-Limit-Fehler

Um die Beschränkungen des Subsystems pro Volume zu überwinden und das Super-Subsystem-Modell nutzen zu können, aktualisieren Sie auf Trident 26.02 oder höher.

Aktualisieren Sie Trident, um das Super-Subsystem-Modell anzuwenden

Um das Super-Subsystem-Modell für RWX NVMe-Volumes anzuwenden:

1. Aktualisieren Sie Trident auf Version 26.02 oder höher.

2. Skalieren Sie alle Pods, die RWX NVMe-Volumes verwenden, auf null Replikate herunter.
3. Stellen Sie sicher, dass keine Workloads aktiv RWX NVMe-Volumes verwenden.
4. Skalieren Sie die Pods wieder hoch.

Diese Neustartsequenz stellt sicher, dass RWX NVMe volumes mithilfe des Super-Subsystem-Modells eingebunden werden.

- Diese Einschränkung gilt nur für RWX-Volumes, die das NVMe-Protokoll verwenden.
- Die Beschränkung auf 64 Knoten gilt pro RWX NVMe volume.
- Andere Zugriffsmodi und andere Protokolle sind nicht betroffen.

Controller-Skalierbarkeit

Trident verbessert die Controller Skalierbarkeit durch erhöhte Parallelität über mehrere Speichertreiber hinweg. Bei aktivierter Controller Skalierbarkeit verarbeitet der Trident Controller Speicheroperationen parallel statt sie zu serialisieren, was den Durchsatz in Kubernetes-Umgebungen mit vielen gleichzeitigen Operationen erhöht.

Vor der Bereitstellung sollte ermittelt werden, welche Trident-Treiber die Controller Skalierbarkeit bei allgemeiner Verfügbarkeit unterstützen und welche Treiber in Trident 26.06 als technische Vorschau verfügbar sind. Dies unterstützt fundierte Entscheidungen zur Bereitstellung und ein effektives Risikomanagement. Die Controller Skalierbarkeit ist standardmäßig deaktiviert.

Schlüsselbegriffe und Definitionen

Controller-Skalierbarkeit

Controller-Skalierbarkeit bezeichnet die Fähigkeit des Trident Controllers, mehrere Speicheroperationen parallel zu verarbeiten, anstatt sie hinter einer einzigen Sperre zu serialisieren. Zu diesen Operationen gehören das Erstellen, Löschen und Ändern der Größe von Volumes, das Erstellen und Löschen von Snapshots, das Veröffentlichen und Aufheben der Veröffentlichung von Volumes sowie die Backend-Verwaltung.

Wenn die Controller-Skalierbarkeit aktiviert ist, werden Vorgänge auf verschiedenen Volumes und Backends gleichzeitig ausgeführt. Dadurch wird der Durchsatz erhöht und die End-to-End-Betriebszeit in Umgebungen mit einer hohen Anzahl gleichzeitiger PersistentVolumeClaim und VolumeSnapshot Vorgänge reduziert.

Standardverhalten (serieller Modus)

Standardmäßig verarbeitet der Trident-Controller Operationen nacheinander. Jede Anfrage zum Erstellen, Löschen, Ändern der Größe oder Erstellen eines Snapshots wird abgeschlossen, bevor Trident die nächste startet. Der serielle Modus ist die unterstützte Standardeinstellung für alle Installationen und erfordert keine Konfiguration.

Der serielle Modus ist für die meisten Arbeitslasten ausreichend. Die Controller Skalierbarkeit sollte nur aktiviert werden, wenn das Verarbeitungsvolumen bei der seriellen Verarbeitung zu einem Rückstau führt.

Unterstützung der Controller-Skalierbarkeit

Trident unterstützt die Skalierbarkeit des Controllers auf unterschiedlichen Reifegraden für verschiedene Speichertreiber.

Allgemeine Verfügbarkeit

Die folgenden Treiber unterstützen die Controller-Skalierbarkeit bei allgemeiner Verfügbarkeit in Trident 26.06:

- `ontap-san`
- `ontap-nas`
- `ontap-nas-economy`
- `ontap-san-economy`
- `google-cloud-netapp-volumes`
- `azure-netapp-files`
- `solidfire-san`

HINWEIS

Die `google-cloud-netapp-volumes` und `google-cloud-netapp-volumes-san` Treiber unterscheiden sich. Trident unterstützt nur `google-cloud-netapp-volumes`. `google-cloud-netapp-volumes-san` sollte in Backend-Konfigurationen oder Beispielen nicht verwendet werden.

Technische Vorschau

Der folgende Treiber unterstützt die Controller Skalierbarkeit als technische Vorschaufunktion in Trident 26.06:

- `asa-r2` (SAN und NVMe)

Dieser Treiber hat folgende Einschränkungen:

- Die Controller-Parallelität ist nur zu Evaluierungs- und Testzwecken verfügbar.
- Das Verhalten kann sich von einer Version zur nächsten ändern.
- NetApp empfiehlt die Verwendung in Produktionsumgebungen nicht.

Wie `enableConcurrency` funktioniert

Wenn `enableConcurrency` auf `true` gesetzt wird, wendet Trident parallele Verarbeitung auf alle vom Controller verwalteten Backends an. Die Einstellung gilt für alle Backends gleichzeitig. Eine Aktivierung für einzelne Backends oder einzelne Treiber ist nicht möglich.

Jedes konfigurierte Backend muss einen Treiber aus der Liste der allgemein verfügbaren oder der technischen Vorschauversionen verwenden. Falls ein Backend einen nicht unterstützten Treiber verwendet, startet Trident nicht und fügt auch kein Backend hinzu, das einen nicht unterstützten Treiber verwendet.

Um den Startvorgang wiederherzustellen, müssen alle Backends entfernt oder neu konfiguriert werden, die einen nicht unterstützten Treiber verwenden, oder die Controller Skalierbarkeit wird deaktiviert.

Bevor Sie aktivieren

WICHTIG

Bevor Sie die Controller Skalierbarkeit aktivieren, vergewissern Sie sich, dass jedes konfigurierte Backend einen Treiber aus der Liste der allgemein verfügbaren oder technischen Vorschauversionen verwendet. Falls ein Backend einen nicht unterstützten Treiber verwendet, startet Trident nach der Aktivierung der Funktion nicht.

Die folgende Tabelle unterstützt die Entscheidung, ob die Controller Skalierbarkeit aktiviert werden soll.

Wenn Ihre Umgebung	Dann
Verwendet nur unterstützte Treiber und weist einen Rückstau bei Controller-Operationen auf	Controller-Skalierbarkeit aktivieren.
Verwendet einen nicht unterstützten Treiber	Der standardmäßige serielle Modus bleibt erhalten. Die Controller-Skalierbarkeit nicht aktivieren.
Bewältigt geringes Arbeitsaufkommen ohne Rückstände	Der standardmäßige serielle Modus bleibt erhalten.

Controller-Skalierbarkeit aktivieren

Die `enableConcurrency` Konfigurationsoption steuert die Skalierbarkeit des Controllers. Diese Option muss während der Trident Installation oder bei der Aktualisierung einer bestehenden Bereitstellung explizit aktiviert werden.

Trident Operator-Bereitstellung

Um die Controller Skalierbarkeit mit dem Trident Operator zu ermöglichen, `enableConcurrency` auf `true` in der `TridentOrchestrator` benutzerdefinierten Ressource (CR) setzen.

Neuinstallation

Erstellen oder bearbeiten Sie die `TridentOrchestrator` CR und setzen Sie `enableConcurrency` auf `true`:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  namespace: trident
  enableConcurrency: true
```

Wenden Sie die CR an:

```
kubectl apply -f tridentorchestrator_cr.yaml
```

Vorhandene Installation

Patchen Sie die bestehende `TridentOrchestrator` CR, um die Skalierbarkeit des Controllers zu ermöglichen:

```
kubectl patch torc trident --type=merge -p
'{"spec":{"enableConcurrency":true}}'
```

Es sollte überprüft werden, ob Trident die Einstellung angewendet hat:

```
kubectl get torc trident -o  
jsonpath='{.status.currentInstallationParams.enableConcurrency}'
```

Helm-Bereitstellung

Um die Controller-Skalierbarkeit mit Helm zu aktivieren, setzen Sie den `enableConcurrency` Wert auf `true`.

Neuinstallation

```
helm install trident netapp-trident/trident-operator --namespace trident  
--create-namespace --set enableConcurrency=true
```

Vorhandene Installation

```
helm upgrade trident netapp-trident/trident-operator --namespace trident  
--set enableConcurrency=true
```

Alternativ setzen Sie `enableConcurrency` auf `true` in einer benutzerdefinierten `values.yaml` Datei:

```
# values.yaml  
enableConcurrency: true
```

Anschließend mit der Wertedatei installieren oder aktualisieren:

```
helm install trident netapp-trident/trident-operator --namespace trident  
--create-namespace -f values.yaml
```

tridentctl-Bereitstellung

Um die Skalierbarkeit des Controllers mit `tridentctl` zu ermöglichen, übergeben Sie das `--enable-concurrency` Flag während der Installation.

Neuinstallation

```
tridentctl install -n trident --enable-concurrency
```

Vorhandene Installation

Um die Controller Skalierbarkeit in einer bestehenden `tridentctl` Bereitstellung zu aktivieren, muss Trident deinstalliert und anschließend mit dem Flag erneut installiert werden:

```
tridentctl uninstall -n trident
tridentctl install -n trident --enable-concurrency
```

Überprüfen, ob die Controller Skalierbarkeit aktiviert ist

Nachdem Sie die Controller Skalierbarkeit aktiviert haben, sollte überprüft werden, ob der Trident Controller mit aktivierter Parallelverarbeitung ausgeführt wird. Die Pod-Logs des Controllers können überprüft werden:

```
kubectl logs -n trident deploy/trident-controller | grep -i concurrency
```

Die Ausgabe enthält einen Protokolleintrag, der bestätigt, dass die Parallelverarbeitung aktiviert ist.

Parallelitätsverhalten

Wenn die Controller-Skalierbarkeit aktiviert ist, wendet der Trident-Controller das folgende Verhalten an:

- Trident ersetzt die einzelne globale Sperre durch eine feingranulare, ressourcenbezogene Sperrung.
- Trident serialisiert Operationen, die dieselbe Ressource verändern, um die Datenkonsistenz zu gewährleisten.
- Operationen, die nur von einer Ressource lesen, laufen gleichzeitig mit anderen Leseoperationen auf dieser Ressource ab.
- Trident begrenzt gleichzeitige ONTAP-API-Anfragen auf 20 pro Management-LIF, um eine Überlastung der Backend-Speichersysteme zu verhindern.
- Wenn mehrere Backends dieselbe Management-LIF verwenden, teilen sie sich dieses Limit von 20 Anfragen.

Vorbehalte und Einschränkungen

Für die Skalierbarkeit von Controllern in Trident 26.06 gelten folgende Überlegungen:

- Die Controller Skalierbarkeit unterstützt nur die Treiber, die in den Listen für allgemeine Verfügbarkeit und technische Vorschau aufgeführt sind. Einzelheiten enthält [Bevor Sie aktivieren](#).
- Der Trident Controller verwaltet die Parallelverarbeitung intern. Diese Version bietet keine vom Benutzer konfigurierbaren Parallelitätsgrenzen.
- Der Gesamtdurchsatz hängt vom verwendeten Speichertreiber, der Reaktionsfähigkeit des Backends und der Leistung des Kubernetes API-Servers ab.
- Hohe Parallelität kann die Belastung der Backend-Speichersysteme erhöhen.
- Das Skalierbarkeitsverhalten des Controllers ist nicht bei allen Treibern identisch.
- Der Treiber in der technischen Vorschau kann unter hoher Last eine inkonsistente Performance aufweisen und das Verhalten kann sich zwischen den Versionen ändern.
- Die Fehlersuche bei gleichzeitig laufenden Operationen kann aufgrund der parallelen Ausführung komplexer sein. Metriken und Protokolle können verschachtelte Operationsausgaben anzeigen.

Empfehlungen

Bevor Sie die Controller Skalierbarkeit aktivieren, sind die folgenden Schritte abzuschließen:

1. Bestätigt werden muss, dass jedes konfigurierte Backend einen Treiber aus der Liste der allgemein verfügbaren oder technischen Vorschau-Treiber verwendet.
2. Die Änderung sollte in einem Nicht-Produktionscluster getestet werden, bevor sie in der Produktionsumgebung angewendet wird.
3. Es sollte sichergestellt sein, dass die Controller Skalierbarkeit nach der Anwendung der Änderung aktiviert ist.

Die folgenden allgemeinen Empfehlungen sind beim Betrieb mit Controller-Skalierbarkeit zu berücksichtigen:

- Allgemein verfügbare Treiber eignen sich für Produktionsumgebungen, die eine hohe Skalierbarkeit erfordern.
- Der technische Vorschau-Treiber ist ausschließlich in Nicht-Produktionsumgebungen zu evaluieren.
- Die Backend- und Controller-Performance bei großem Maßstab im Blick behalten.
- In Automatisierungsskripten sollte nicht von einer bestimmten Reihenfolge der Vorgänge ausgegangen werden.

Automatische Volumenerweiterung

Die automatische Volumenerweiterung ermöglicht es, dass von Trident bereitgestellte Persistent Volumes automatisch wachsen, sobald die genutzte Kapazität einen definierten Schwellenwert erreicht. Diese Funktion reduziert den Betriebsaufwand und trägt dazu bei, Anwendungsunterbrechungen aufgrund von Kapazitätsengpässen zu vermeiden. Die automatische Volumenerweiterung wird mithilfe von Autogrow Policies implementiert. Eine Autogrow Policy definiert:

- Die Auslastungsschwelle, die die Erweiterung auslöst
- Die Menge, um die das Volumen wächst
- Die maximale Größe, die das Volume erreichen kann

HINWEIS

Volumes erhöhen sich automatisch, wenn der festgelegte Auslastungsschwellenwert überschritten wird. Volumes werden niemals automatisch reduziert.

Anforderungen

Bevor Sie die automatische Volumenerweiterung konfigurieren, stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind:

- Trident 26.02 oder später
- Rollenbasierte Zugriffssteuerungsberechtigungen zum Erstellen `TridentAutogrowPolicy` benutzerdefinierter Ressourcen
- StorageClasses konfiguriert mit `allowVolumeExpansion: true`
- Unterstützte ONTAP-Protokolle:
 - Netzwerkdateisystem (NFS)
 - Internet Small Computer Systems Interface (iSCSI)
 - Non-Volatile Memory Express (NVMe)

- Fibre Channel Protocol (FCP)

Einschränkungen

- ONTAP Non-Volatile Memory Express Raw-Block-Volumes früher als ONTAP 9.16.1 unterstützen keine automatische Erweiterung.
- Für SAN-Volumes, wenn `growthAmount` kleiner oder gleich 50 Mebibyte ist, erhöht Trident den Wert automatisch auf 51 Mebibyte, bevor die Größe angepasst wird, vorausgesetzt, die resultierende Größe überschreitet nicht `maxSize`.
- In Brownfield-Umgebungen funktioniert die automatische Erweiterung möglicherweise nicht für bestimmte vorhandene Volumes aufgrund des Migrationsverhaltens bei der Volume-Veröffentlichung.
- Wenn ein Volumen `maxSize` erreicht, findet keine weitere Erweiterung statt.
- Unterstützte Protokolle für die automatische Volumenerweiterung:
 - Netzwerkdateisystem (NFS)
 - Internet Small Computer Systems Interface (iSCSI)
 - Non-Volatile Memory Express (NVMe)
 - Fibre Channel Protocol (FCP)

Volumes mit Autogrow-Richtlinie bereitstellen

Die Autogrow Policy kann auf zwei Ebenen konfiguriert werden:

- Speicherebene: Legt den Standardwert für alle Volumes fest (mittels Annotation)
- PVC-Ebene: Überschreibt die Standardeinstellung der Speicherklasse (using annotation)

Erstellen Sie eine Autogrow-Richtlinie

Autogrow-Richtlinien ermöglichen die automatische Volumenerweiterung, wenn Volumes einen definierten Kapazitätsschwellenwert erreichen.

Stellen Sie sicher, dass Sie Folgendes haben:

- Trident 26.02 oder höher installiert
- Rollenbasierte Zugriffssteuerungsberechtigungen zum Erstellen `TridentAutogrowPolicy` von Ressourcen
- Verständnis der Anforderungen an das Arbeitslastwachstum

Eine Autogrow Policy definiert, wie sich Volumes automatisch vergrößern, wenn sie einen festgelegten Kapazitätsschwellenwert erreichen.

Sie können Autogrow Policies an jeder beliebigen Stelle in Ihrem Workflow erstellen:

- Bevor StorageClasses und Volumes erstellt werden
- Nachdem StorageClasses existieren
- Nachdem die Volumes bereitgestellt wurden

Diese Flexibilität ermöglicht es Ihnen, eine automatische Erweiterung einzuführen, ohne bestehende Ressourcen neu zu erstellen.

Spezifikationen der Autogrow Policy

Autogrow Policies sind benutzerdefinierte Kubernetes-Ressourcen, die wie folgt definiert sind:

Feld	Beschreibung	Format	Erforderlich	Beispiel	Standard
Name	Eindeutige Richtlinienkennung	Zeichenkette	Ja	Produktionsdatenbankrichtlinie	Keine
usedThreshold	Kapazitätsprozentsatz, der die Erweiterung auslöst	Prozentzeichenkette	Ja	"80%"	Keine
growthAmount	Betrag, um den erhöht wird, wenn der Schwellenwert erreicht ist	Prozentualer Anteil oder Größe	Nein	"10%" oder "5Gi"	"10%"
maxSize	Maximale Volumengrößenbeschränkung	Kubernetes-Menge	Nein	"500Gi"	Unbegrenzt

Erstellen Sie eine Autogrow-Richtlinie

Schritte

1. Erstellen Sie eine YAML-Datei, die Ihre Autogrow Policy definiert:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: standard-autogrow
spec:
  usedThreshold: "80%"
  growthAmount: "10%"
  maxSize: "500Gi"
```

2. Wenden Sie die Richtlinie auf Ihren Cluster an:

```
kubectl apply -f autogrow-policy.yaml
```

3. Überprüfen Sie, ob die Richtlinie erstellt wurde:

```
kubectl get tridentautogrowpolicy standard-autogrow
```

Erwartete Ausgabe

NAME	USED THRESHOLD	GROWTH AMOUNT	STATE
standard-autogrow	80%	10%	Success

Richtlinienstatus

Nachdem Sie eine Richtlinie erstellt haben, validiert Trident die Spezifikation und weist einen der folgenden Zustände zu:

Status	Beschreibung	Aktion erforderlich
Erfolg	Die Richtlinie ist validiert und einsatzbereit.	Keine.
Fehlgeschlagen	Validierungsfehler festgestellt.	Spezifikation prüfen und korrigieren.
Löschen	Löschung wird durchgeführt.	Warten Sie auf den Abschluss.

Ordnen Sie eine Richtlinie einer StorageClass zu

Sie können eine Autogrow-Richtlinie mit einer StorageClass durch die `trident.netapp.io/autogrowPolicy` Annotation verknüpfen. Alle Volumes, die aus dieser StorageClass bereitgestellt werden, erben die Richtlinie.

HINWEIS Die StorageClass muss `allowVolumeExpansion: true` haben.

Schritte

1. Erstellen oder ändern Sie eine StorageClass mit der Autogrow Policy-Annotation:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
  annotations:
    trident.netapp.io/autogrowPolicy: "production-db-policy"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

2. Wenden Sie die StorageClass an:

```
kubectl apply -f storageclass.yaml
```

3. Überprüfen Sie die Annotation:

```
kubectl get storageclass ontap-gold -o
jsonpath='{.metadata.annotations.trident\.netapp\.io/autogrowPolicy}'
```

Erwartete Ausgabe

```
production-db-policy
```

Richtlinienvorrang

Wenn Autogrow-Richtlinien-Annotationen sowohl für eine StorageClass als auch für ein PVC festgelegt sind, wendet Trident die folgenden Vorrangregeln an:

1. **PVC-Annotationen haben Vorrang.** Wenn eine PVC `trident.netapp.io/autogrowPolicy` festlegt, wird dieser Wert immer verwendet.
2. **StorageClass-Anmerkung gilt nur, wenn die PVC keine Anmerkung enthält.**
3. **Wenn keines der beiden die Annotation hat,** wird keine Autogrow Policy angewendet.

StorageClass Anmerkung	PVC-Anmerkung	Effektives Verhalten
<code>trident.netapp.io/autogrowPolicy: standard-agp</code>	Nicht festgelegt	Anwendungsgebiete <code>standard-agp</code> .
<code>trident.netapp.io/autogrowPolicy: standard-agp</code>	<code>trident.netapp.io/autogrowPolicy: logs-policy</code>	Verwendet <code>logs-policy</code> (PVC überschreibt StorageClass).
<code>trident.netapp.io/autogrowPolicy: standard-agp</code>	<code>trident.netapp.io/autogrowPolicy: "none"</code>	Keine Autogrow-Richtlinie (PVC deaktiviert autogrow).
Nicht festgelegt	<code>trident.netapp.io/autogrowPolicy: dev-policy</code>	Anwendungsgebiete <code>dev-policy</code> .
Nicht festgelegt	Nicht festgelegt	Keine Autogrow Policy.

Konfigurationsbeispiele

Die folgenden Beispiele zeigen gängige Autogrow Policy-Konfigurationen für verschiedene Anwendungsfälle.

Konservative Richtlinie für Produktionsdatenbanken

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: production-db-policy
spec:
  usedThreshold: "75%"
  growthAmount: "20%"
  maxSize: "5Ti"
```

Protokollspeicherung mit festen Wachstumsschritten

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: log-storage-policy
spec:
  usedThreshold: "90%"
  growthAmount: "10Gi"
  maxSize: "100Gi"
```

Minimalrichtlinie mit Standardeinstellungen

Wenn Sie growthAmount und maxSize weglassen, verwendet Trident die Standardeinstellungen (10% Wachstum, unbegrenzte Größe):

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: logs-policy
spec:
  usedThreshold: "85%"
```

Richtlinie mit benutzerdefiniertem maxSize und standardmäßigem growthAmount

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: default-ga-policy
spec:
  usedThreshold: "70%"
  maxSize: "100Gi"
```

Aggressives Wachstum mit unbegrenztem maxSize

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: aggressive-growth-policy
spec:
  usedThreshold: "80%"
  growthAmount: "150%"
```

Richtlinie mit Bruchteilprozenten

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: precise-policy
spec:
  usedThreshold: "80.28%"
  growthAmount: "10.65%"
  maxSize: "100Gi"
```

HINWEIS

Bruchprozentsätze werden unterstützt. Wenn Sie mehr als drei Dezimalstellen angeben, rundet Trident den Wert auf drei Dezimalstellen.

NAS StorageClass mit Autogrow

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-autogrow
  annotations:
    trident.netapp.io/autogrowPolicy: "standard-autogrow"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: true
```

SAN StorageClass mit Autogrow

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: database-storage
  annotations:
    trident.netapp.io/autogrowPolicy: "production-db-policy"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

Autogrow-Richtlinien verwalten

Nachdem Sie Autogrow Policies erstellt haben, können Sie diese anzeigen, aktualisieren und löschen. Es besteht außerdem die Möglichkeit zu überwachen, welche Volumes eine bestimmte Policy verwenden.

Autogrow-Richtlinien anzeigen

Alle Richtlinien auflisten

Verwenden Sie `kubectl`, um alle Autogrow Policies in Ihrem Cluster aufzulisten:

```
kubectl get tridentautogrowpolicy
```

Alternativ verwenden Sie `tridentctl`:

```
tridentctl get autogrowpolicy
```

Richtliniendetails anzeigen

Um die vollständige Spezifikation und den Status einer Richtlinie anzuzeigen:

```
kubectl describe tridentautogrowpolicy production-db-policy
```

So zeigen Sie eine Richtlinie mit den zugehörigen Volumes im YAML-Format an:

```
tridentctl get autogrowpolicy production-db-policy -o yaml
```

Eine Autogrow-Richtlinie aktualisieren

Sie können eine bestehende Richtlinie anpassen, um deren Schwellenwert, Wachstumsbetrag oder maximale Größe zu ändern. Änderungen werden sofort für alle Volumes wirksam, die diese Richtlinie verwenden.

WICHTIG

Änderungen betreffen alle Volumes, die diese Richtlinie verwenden. Änderungen sollten nach Möglichkeit zunächst in einer Nicht-Produktionsumgebung getestet werden.

Schritte

1. Richtlinie bearbeiten:

```
kubectl edit tridentautogrowpolicy production-db-policy
```

2. Ändern Sie die `spec` Felder nach Bedarf:

```
spec:
  usedThreshold: "75%"      # Changed from 80%
  growthAmount: "20%"      # Changed from 10%
  maxSize: "1Ti"           # Changed from 500Gi
```

3. Speichern und beenden. Die Änderungen werden sofort wirksam.

Aktualisierungsüberlegungen

- **Sofortige Wirkung:** Alle Volumina, die die Richtlinie anwenden, übernehmen die neuen Parameter bei der nächsten Wachstumsbewertung.
- **Kein Neustart des Volumes erforderlich:** Änderungen gelten für den nächsten Wachstumsvorgang.
- **Erst testen:** Validieren Sie Änderungen nach Möglichkeit in einer Nicht-Produktionsumgebung.
- **Änderungen kommunizieren:** Benachrichtigen Sie die Teams, wenn Sie gemeinsam genutzte Richtlinien ändern.

Löschen einer Autogrow-Richtlinie

Autogrow Policies verwenden Finalizer-Schutz, um eine versehentliche Löschung zu verhindern, während Volumes sie aktiv verwenden.

Schritte

1. Löschen Sie die Richtlinie:

```
kubectl delete tridentautogrowpolicy production-db-policy
```

2. Wenn die Volumes die Richtlinie noch verwenden, wechselt der Löschvorgang in einen `Deleting` Status. Es kann geprüft werden, welche Volumes betroffen sind:

```
tridentctl get autogrowpolicy production-db-policy -o yaml
```

3. Entfernen Sie die Richtlinie von jedem betroffenen Volume. Wählen Sie eine der folgenden Optionen:

- **Option A: Automatisches Wachstum explizit deaktivieren**, indem Sie die Annotation auf "none" setzen:

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="none" \
  --overwrite
```

- **Option B: Die Annotation vollständig entfernen:**

```
kubectl annotate pvc <pvc-name> \
```

```
trident.netapp.io/autogrowPolicy-
```

Löschverhalten

Szenario	Verhalten
Keine Volumes verwenden die Richtlinie	Die Richtlinie wird sofort gelöscht.
Volumes verwenden die Richtlinie	Die Richtlinie tritt in <code>Deleting</code> Zustand ein. Ein Finalisierer blockiert den Abschluss, bis alle Volumes entfernt sind.
Alle Volumes werden aus der Richtlinie entfernt	Finalisierer werden entfernt und die Richtlinie wird gelöscht.

Autogrow-Richtliniennutzung überwachen

Überprüfen Sie Volumes mithilfe einer Richtlinie

```
tridentctl get autogrowpolicy production-db-policy -o json | jq '.volumes'
```

Ermitteln Sie, welche Richtlinie ein Volume verwendet

```
kubectl get pvc database-pvc -o  
jsonpath='{.metadata.annotations.trident\.netapp\.io/autogrowPolicy}'
```

Richtlinienereignisse überwachen

```
kubectl get events --field-selector  
involvedObject.kind=TridentAutogrowPolicy
```

Unterstützte Protokolle

Autogrow unterstützt die folgenden Speicherprotokolle:

- NFS
- iSCSI
- FCP
- NVMe

HINWEIS

Für SAN-Volumes, wenn die konfigurierte `growthAmount` 50 MiB oder weniger beträgt, erhöht Trident den Wachstumsbetrag für die Größenänderung automatisch auf 51 MB, solange die resultierende Größe `maxSize` nicht überschreitet.

Unterstützte Backends

Autogrow unterstützt ONTAP Backends. Autogrow unterstützt außerdem Google Cloud NetApp Volumes (GCNV) sowohl für NAS- als auch für SAN Volumes.

Bekannte Einschränkungen

- **ONTAP NVMe raw block volumes:** Volumes, die mit ONTAP Versionen vor 9.16.1 erstellt wurden, unterstützen kein automatisches Wachstum.
- **Vorhandene Volumes (Brownfield-Bereitstellungen):** Die automatische Vergrößerung funktioniert möglicherweise nicht für vorhandene Volumes, selbst wenn eine gültige Autogrow-Richtlinie angewendet wird. Diese Einschränkung ist auf eine laufende Migration der Volume-Veröffentlichungen zurückzuführen. Zur Bestätigung, dass die Migration abgeschlossen ist, sind die Trident Controller-Protokolle auf "Migration completed" Meldungen zu prüfen.

Häufig gestellte Fragen

Wann wertet Trident den Schwellenwert aus?

Trident überwacht kontinuierlich die Speicherauslastung. Wenn die belegte Kapazität die `usedThreshold` überschreitet, erstellt Trident eine interne Größenänderungsanforderung und erweitert das Volume um die konfigurierte `growthAmount`.

Diese Richtlinie löst beispielsweise eine Erweiterung bei 80 % Kapazität aus und erhöht das Volumen jedes Mal um 10 %, bis zu einem Maximum von 500 GiB:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: standard-autogrow
spec:
  usedThreshold: "80%"
  growthAmount: "10%"
  maxSize: "500Gi"
```

Kann ich eine Richtlinie anwenden, nachdem Volumes bereits bereitgestellt wurden?

Ja. Sie können jederzeit eine Autogrow-Richtlinie erstellen und diese auf bestehende PVCs anwenden, indem Sie die `trident.netapp.io/autogrowPolicy` Annotation hinzufügen oder aktualisieren. Sie müssen weder die PVC noch die StorageClass neu erstellen.

So wenden Sie eine Richtlinie auf ein bestehendes PVC an:

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="production-db-policy" \
  --overwrite
```

Um eine Richtlinie auf eine bestehende StorageClass anzuwenden:

```
kubectl annotate storageclass ontap-gold \
  trident.netapp.io/autogrowPolicy="production-db-policy" \
  --overwrite
```

Was passiert, wenn ich sowohl für die StorageClass als auch für das PVC eine Autogrow-Richtlinie einrichte?

Die PVC-Annotation hat immer Vorrang. Wenn eine PVC die `trident.netapp.io/autogrowPolicy` Annotation enthält, verwendet Trident diesen Wert unabhängig davon, was die StorageClass angibt. Details sind in ["Richtlinienvorrang"](#) zu finden.

Zum Beispiel, gegeben diese StorageClass:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
  annotations:
    trident.netapp.io/autogrowPolicy: "standard-agp"
provisioner: csi.trident.netapp.io
allowVolumeExpansion: true
```

Und dieses PVC, das die StorageClass-Richtlinie außer Kraft setzt:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: database-pvc
  annotations:
    trident.netapp.io/autogrowPolicy: "logs-policy"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 50Gi
  storageClassName: ontap-gold
```

Trident verwendet `logs-policy` für `database-pvc`, nicht `standard-agp`.

Wie deaktiviere ich Autogrow für ein bestimmtes Volume?

Die PVC-Annotation wird auf `"none"` gesetzt. Diese Einstellung überschreibt jede Richtlinie auf StorageClass-Ebene für dieses Volume:

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="none" \
  --overwrite
```

Sie können überprüfen, ob autogrow deaktiviert ist:

```
kubectl get pvc <pvc-name> -o jsonpath
='{.metadata.annotations.trident\.netapp\.io/autogrowPolicy}'
```

Erwartete Ausgabe

```
none
```

Was passiert, wenn ein Volumen maxSize erreicht?

Trident stellt die Erweiterung des Volumes ein. Es werden keine weiteren Größenänderungsanforderungen für dieses Volume erstellt, selbst wenn die Nutzung über den `usedThreshold` hinaus weiter zunimmt.

Beispielsweise wird mit dieser Richtlinie das Wachstum des Volumens durch Trident gestoppt, sobald es 100 GiB erreicht:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: capped-policy
spec:
  usedThreshold: "90%"
  growthAmount: "10Gi"
  maxSize: "100Gi"
```

Um unbegrenztes Wachstum zu ermöglichen, lassen Sie `maxSize` weg oder setzen Sie es auf 0:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: unlimited-policy
spec:
  usedThreshold: "85%"
  growthAmount: "10%"
```

Kann ich eine Richtlinie ändern, ohne die Volumes neu zu starten?

Ja. Wenn eine Richtlinie aktualisiert wird, übernehmen alle Volumes, die diese Richtlinie verwenden, die neuen

Parameter bei der nächsten Wachstumsbewertung. Ein Neustart der Volumes ist nicht erforderlich.

Um eine Richtlinie zu aktualisieren:

```
kubectl edit tridentautogrowpolicy production-db-policy
```

Passen Sie die Felder nach Bedarf an:

```
spec:
  usedThreshold: "75%"      # Changed from 80%
  growthAmount: "20%"      # Changed from 10%
  maxSize: "1Ti"           # Changed from 500Gi
```

Speichern und beenden. Überprüfen Sie die aktualisierte Richtlinie:

```
kubectl get tridentautogrowpolicy production-db-policy
```

Erwartete Ausgabe

NAME	USED THRESHOLD	GROWTH AMOUNT	STATE
production-db-policy	75%	20%	Success

Warum befindet sich meine Richtlinie im Status „Fehlgeschlagen“?

Ein Failed Status zeigt an, dass die Richtlinienspezifikation Validierungsfehler enthält. Führen Sie den folgenden Befehl aus, um die Fehlerdetails anzuzeigen:

```
kubectl describe tridentautogrowpolicy <policy-name>
```

Häufige Ursachen sind ein ungültiger `usedThreshold` (muss zwischen 1–99 % liegen), ein `growthAmount`, der `maxSize` überschreitet, oder ein ungültiges Kubernetes-Mengenformat. Korrigieren Sie die Spezifikation und wenden Sie sie erneut an.

```
kubectl apply -f autogrow-policy.yaml
```

Warum kann ich eine Richtlinie nicht löschen?

Richtlinien verwenden einen Finalisierungsschutz. Wenn Volumes die Richtlinie noch verwenden, wechselt der Löschvorgang in einen `Deleting` Status und wartet, bis alle Volumes aus der Richtlinie entfernt wurden.

Identifizieren Sie die betroffenen Volumes:

```
tridentctl get autogrowpolicy production-db-policy -o yaml
```

Entfernen Sie anschließend die Annotation von jedem PVC:

```
# Option A: Explicitly disable autogrow
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="none" \
  --overwrite

# Option B: Remove the annotation entirely
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy-
```

Nachdem alle Volumes entfernt wurden, gibt Trident den Finalizer frei und löscht die Richtlinie.

Welche Backends unterstützt Autogrow?

Autogrow unterstützt ONTAP Backends. Unterstützt werden außerdem Google Cloud NetApp Volumes (GCNV) sowohl für NAS- als auch für SAN-Volumes.

Autogrow unterstützt die Protokolle NFS, iSCSI, FCP und NVMe. NVMe Raw-Block-Volumes erfordern ONTAP 9.16.1 oder höher.

Bei bestehenden Volumes in Brownfield-Umgebungen muss die Migration der Volume-Veröffentlichung möglicherweise abgeschlossen sein, bevor das automatische Wachstum wirksam wird. Zum Überprüfen des Migrationsstatus sind die Trident Controller-Protokolle zu prüfen:

```
kubectl logs -l app=trident-controller -n trident | grep "Migration
completed"
```

Die folgenden StorageClass-Beispiele zeigen Autogrow für NAS- und SAN-Backends konfiguriert:

NAS-Backend

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-autogrow
  annotations:
    trident.netapp.io/autogrowPolicy: "standard-autogrow"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: true
```

SAN-Backend

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: database-storage
  annotations:
    trident.netapp.io/autogrowPolicy: "production-db-policy"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

Was ist der minimale Zuwachs für SAN-Volumes?

Für SAN-Volumes beträgt die effektive Mindestwachstumsgröße 51 MB. Wenn Sie eine `growthAmount` von 50 MiB oder weniger konfigurieren, erhöht Trident das Wachstum für die Größenänderung automatisch auf 51 MB.

Diese Richtlinie legt beispielsweise ein `growthAmount` von "40Mi" fest, aber Trident wendet ein Wachstum von 51 MB für jedes SAN-Volume an, das sie verwendet:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: san-minimal-policy
spec:
  usedThreshold: "85%"
  growthAmount: "40Mi"
  maxSize: "100Gi"
```

Um diese automatische Anpassung zu vermeiden, setzen Sie `growthAmount` auf einen Wert größer als 50 MiB:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: san-policy
spec:
  usedThreshold: "85%"
  growthAmount: "100Mi"
  maxSize: "500Gi"
```

Importieren von Trident Volumes

Sie können vorhandene Speichervolumes als Kubernetes PV mit `tridentctl import` importieren oder indem Sie einen Persistent Volume Claim (PVC) mit Trident-Importannotationen erstellen.

Überblick und Überlegungen

Sie könnten ein Volume in Trident importieren, um:

- Eine Anwendung containerisieren und ihren vorhandenen Datensatz wiederverwenden
- Verwenden Sie einen Klon eines Datensatzes für eine kurzlebige Anwendung
- Einen ausgefallenen Kubernetes-Cluster wiederherstellen
- Anwendungsdaten während der Notfallwiederherstellung migrieren

Überlegungen

Bevor Sie ein Volume importieren, beachten Sie die folgenden Hinweise.

- Trident kann nur ONTAP-Volumes vom Typ RW (Read-Write) importieren. Volumes vom Typ DP (data protection) sind SnapMirror Ziel-Volumes. 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 dann den Import durch.

WARNUNG

Dies ist besonders wichtig für Block-Volumes, da Kubernetes die vorherige Verbindung nicht erkennen und leicht ein aktives Volume an einen Pod anhängen könnte. Dies kann zu Datenbeschädigung führen.

- Obwohl `StorageClass` auf einem PVC angegeben werden muss, verwendet Trident diesen Parameter beim Import nicht. Speicherklassen werden während der Volume-Erstellung verwendet, um anhand der Speichereigenschaften aus den verfügbaren Pools auszuwählen. Da das Volume bereits existiert, ist beim Import keine Poolauswahl erforderlich. Daher schlägt der Import nicht fehl, selbst wenn das Volume auf einem Backend oder in einem Pool existiert, der nicht der im PVC angegebenen Speicherklasse entspricht.
- Die vorhandene Volumengröße wird ermittelt und im PVC festgelegt. Nachdem das Volumen vom Speichertreiber importiert wurde, wird das PV mit einer `ClaimRef` auf das PVC erstellt.
 - Die Rückgewinnungsrichtlinie ist anfänglich auf `retain` im PV festgelegt. Nachdem Kubernetes PVC und PV erfolgreich gebunden hat, wird die Rückgewinnungsrichtlinie aktualisiert, um der Rückgewinnungsrichtlinie der Storage Class zu entsprechen.
 - Wenn die Rückgewinnungsrichtlinie der Storage Class `delete` ist, löscht Trident das Speichervolume, wenn das PV gelöscht wird.
- Standardmäßig verwaltet Trident die PVC und benennt das FlexVol Volume sowie die LUN oder den Qtree im Backend um. Es besteht die Möglichkeit, das `--no-manage` Flag zu verwenden, um ein nicht verwaltetes Volume zu importieren, und das `--no-rename` Flag, um den Volume-Namen beizubehalten.
 - `--no-manage*` - Wenn Sie das `--no-manage` Flag verwenden, führt Trident während des gesamten Lebenszyklus der Objekte keine zusätzlichen Operationen an der PVC oder PV durch. Das Speichervolume wird beim Löschen der PV nicht gelöscht und andere Operationen wie das Klonen und die Größenänderung von Volumes werden ebenfalls ignoriert.
 - `--no-rename*` - Wenn das `--no-rename` Flag verwendet wird, behält Trident beim Importieren von

Volumes den vorhandenen Volume-Namen bei und verwaltet den Lebenszyklus der Volumes. Diese Option wird nur für die `ontap-nas`, `ontap-san` (einschließlich ASA r2-Systeme), `ontap-san-economy` und `ontap-nas-economy` (`qtree import`) Treiber unterstützt.

TIPP

Diese Optionen sind nützlich, wenn Sie Kubernetes für containerisierte Workloads verwenden möchten, aber ansonsten den Lebenszyklus des Speichervolumens außerhalb von Kubernetes verwalten möchten.

- Dem PVC und PV wird eine Anmerkung hinzugefügt, die zwei Zwecken dient: Sie zeigt an, dass das Volume importiert wurde und ob der PVC und PV verwaltet werden. Diese Anmerkung sollte nicht verändert oder entfernt werden.

Importieren Sie ein Volume

Sie können ein Volume entweder mit `tridentctl import` oder durch das Erstellen einer PVC mit Trident-Importanmerkungen importieren.

HINWEIS

Wenn Sie PVC-Anmerkungen verwenden, müssen Sie `tridentctl` nicht herunterladen oder verwenden, um das Volume zu importieren.

Verwendung von `tridentctl`

Schritte

1. Erstellen Sie eine PVC-Datei (zum Beispiel `pvc.yaml`), die zur Erstellung des PVC verwendet wird. Die PVC-Datei sollte `name`, `namespace`, `accessModes` und `storageClassName` enthalten. Optional können Sie `unixPermissions` in Ihrer PVC-Definition angeben.

Das Folgende ist ein Beispiel für eine Mindestspezifikation:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class
```

HINWEIS

Geben Sie nur die erforderlichen Parameter an. Zusätzliche Parameter wie PV-Name oder Volume-Größe können dazu führen, dass der Importbefehl fehlschlägt.

2. Verwenden Sie den `tridentctl import` Befehl, um den Namen des Trident Backend, das das Volume enthält, und den Namen, der das Volume auf dem Storage eindeutig identifiziert (zum Beispiel: ONTAP FlexVol, Element Volume), anzugeben. Das `-f` Argument ist erforderlich, um den Pfad zur PVC-Datei anzugeben.

```
tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-file>
```

Verwendung von PVC-Anmerkungen

Schritte

1. Erstellen Sie eine PVC-YAML-Datei (zum Beispiel `pvc.yaml`) mit den erforderlichen Trident-Importannotationen. Die PVC-Datei sollte Folgendes enthalten:

- name and namespace in Metadaten
- accessModes, resources.requests.storage, und storageClassName in Spezifikation
- Anmerkungen:
 - `trident.netapp.io/importOriginalName`: Volumenname auf dem Backend
 - `trident.netapp.io/importBackendUUID`: Backend-UUID, wo das Volume existiert
 - `trident.netapp.io/notManaged` (*Optional*): Auf "true" für nicht verwaltete Volumes festlegen. Standardmäßig ist "false".

Im Folgenden finden Sie eine Beispielspezifikation für den Import eines verwalteten Volumes:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <pvc-name>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "<volume-name>"
    trident.netapp.io/importBackendUUID: "<backend-uuid>"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: <size>
  storageClassName: <storage-class-name>
```

2. Wenden Sie die PVC YAML-Datei auf Ihren Kubernetes-Cluster an:

```
kubectl apply -f <pvc-file>.yaml
```

Trident wird das Volumen automatisch importieren und es an das PVC binden.

Beispiele

Überprüfen Sie die folgenden Beispiele für den Volumenimport für unterstützte Treiber.

ONTAP NAS und ONTAP NAS FlexGroup

Trident unterstützt den Volumenimport mithilfe der `ontap-nas` und `ontap-nas-flexgroup` Treiber.

HINWEIS

- Der `'ontap-nas-economy'` Treiber verwaltet Speicher auf Qtree-Ebene, daher wird das eigenständige Importieren eines FlexVol Volumes nicht unterstützt. Zum Importieren von Qtrees mit dem `'ontap-nas-economy'` Treiber siehe [Qtrees importieren](#) (Tech Preview). Wenn Trident einen Qtree importiert, wird auch das übergeordnete FlexVol als Bucket importiert, das den Qtree enthält.
- Die `ontap-nas` und `'ontap-nas-flexgroup'` Treiber erlauben keine doppelten Datenträgernamen.

Jedes mit dem `ontap-nas` Treiber erstellte Volume ist ein FlexVol Volume auf dem ONTAP Cluster. Das Importieren von FlexVol Volumes mit dem `ontap-nas` Treiber funktioniert genauso. Ein FlexVol Volume, das bereits auf einem ONTAP Cluster existiert, kann als `ontap-nas` PVC importiert werden. Ebenso können FlexGroup Volumes als `ontap-nas-flexgroup` PVCs importiert werden.

ONTAP NAS-Beispiele mit `tridentctl`

Die folgenden Beispiele zeigen, wie verwaltete und nicht verwaltete Volumes mit `tridentctl` importiert werden.

Verwaltetes Volume

Das folgende Beispiel importiert ein Volume mit dem Namen `managed_volume` auf einem Backend mit dem Namen `ontap_nas`:

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>
```

PROTOCOL	NAME	BACKEND	UUID	SIZE	STATE	STORAGE CLASS	MANAGED
file	pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7	c5a6f6a4-b052-423b-80d4-8fb491a14a22		1.0 GiB	online	standard	true

Unverwaltetes Volume

Bei Verwendung des `--no-manage` Arguments benennt Trident das Volume nicht um.

Das folgende Beispiel importiert `unmanaged_volume` auf dem `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 NAS-Beispiele mit PVC-Annotationen

Die folgenden Beispiele zeigen, wie verwaltete und nicht verwaltete Volumes mithilfe von PVC-Annotationen importiert werden.

Verwaltetes Volume

Das folgende Beispiel importiert ein 1GiB ontap-nas Volume namens `ontap_volume1` vom Backend `81abcb27-ea63-49bb-b606-0a5315ac5f21` mit RWO-Zugriffsmodus, der mithilfe von PVC-Annotationen festgelegt ist:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <managed-imported-volume>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "ontap_volume1"
    trident.netapp.io/importBackendUUID: "81abcb27-ea63-49bb-b606-
0a5315ac5f21"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: <storage-class-name>
```

Unverwaltetes Volume

Das folgende Beispiel importiert ein 1GiB ontap-nas Volume mit dem Namen `ontap-volume2` vom Backend `34abcb27-ea63-49bb-b606-0a5315ac5f34` mit festgelegtem RWO-Zugriffsmodus mithilfe

von PVC-Annotationen:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <umanaged-imported-volume>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "ontap-volume2"
    trident.netapp.io/importBackendUUID: "34abcb27-ea63-49bb-b606-0a5315ac5f34"
    trident.netapp.io/notManaged: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: <storage-class-name>
```

Qtrees importieren

HINWEIS

Qtree Import ist eine Tech Preview Funktion in Trident 26.06. Tech Preview Funktionen sind für den Produktiveinsatz nicht unterstützt.

Ab Trident 26.06 ist es möglich, bestehende ONTAP qtrees mithilfe des `ontap-nas-economy` Treibers in Trident zu importieren. Der Qtree wird durch seinen FlexVol Volume- und Qtree-Namen im `flexvol/qtree` Format sowie durch die UUID des Backends, das ihn enthält, identifiziert. Die allgemeinen Importüberlegungen in [Überblick und Überlegungen](#) gelten auch für den Qtree-Import.

Einen Qtree mit `tridentctl` importieren

Geben Sie den Qtree im `flexvol/qtree` Format als Volume-Namen an. Im folgenden Beispiel wird der Qtree `qtreet1` aus dem FlexVol Volume `flexvol1` auf dem Backend `ontap_nas_economy` importiert:

```
tridentctl import volume ontap_nas_economy flexvol1/qtreet1 -f <path-to-pvc-file>
```

Um den Qtree zu importieren, ohne ihn umzubenennen, das `--no-rename` Flag hinzufügen. Um den Qtree zu importieren, ohne dass Trident seinen Lebenszyklus verwaltet, das `--no-manage` Flag hinzufügen.

Import eines Qtrees mithilfe von PVC-Annotationen

Eine PVC mit den folgenden Annotationen erstellen:

```
kind: PersistentVolumeClaim
```

```

apiVersion: v1
metadata:
  name: <pvc-name>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "flexvoll1/qtree1"
    trident.netapp.io/importBackendUUID: "<backend-uuid>"
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: <size>
    storageClassName: <storage-class-name>

```

Um die bestehenden FlexVol volume- und qtree-Namen beizubehalten, die `trident.netapp.io/importNoRename: "true"` Annotation hinzufügen. Um den qtree als nicht verwaltet zu importieren, die `trident.netapp.io/notManaged: "true"` Annotation hinzufügen.

Benennungs- und Managementverhalten

Verwalteter Import (Standard)

Trident benennt den Qtree entsprechend dem PVC-Namen um und benennt das FlexVol volume gemäß der `ontap-nas-economy` Namenskonvention um. Der Verbindungspfad bleibt unverändert.

--no-rename

Trident behält die bestehenden FlexVol volume- und qtree-Namen bei. Die Namen müssen bereits den Trident Namenskonventionen entsprechen, sonst kann Trident keine Lösch- oder Mount-Operationen für den qtree verwalten.

--no-manage

Trident referenziert den qtree, verwaltet aber nicht dessen Lebenszyklus. Die Namen müssen den Trident Namenskonventionen entsprechen, und Trident verwendet oder verwaltet keine Auto-Export-Richtlinien für den qtree.

Verhalten der Export-Policy

Wenn das Backend automatische Exportrichtlinien verwendet, nutzt der importierte Qtree das Exportrichtlinienmodell pro Qtree, bei dem jeder Qtree seine eigene Exportrichtlinie besitzt. Trident weist dem Qtree beim Import eine leere Exportrichtlinie zu. Beim Veröffentlichen aktualisiert Trident das FlexVol volume auf die Exportrichtlinie pro Backend, falls der automatische Export aktiviert ist und die bestehende FlexVol Exportrichtlinie `default` oder leer ist.

ACHTUNG

Die Aktualisierung der Export-Richtlinie des FlexVol Volumes kann andere Qtrees im selben FlexVol Volume beeinträchtigen, die derzeit eingebunden sind. Qtrees sollten nur importiert werden, wenn die Qtrees im FlexVol Volume ausgehängt sind.

WARNUNG

Zwei Kubernetes-Cluster dürfen nicht auf dieselbe aktive SVM verweisen, und es darf kein Qtree importiert werden, der noch von einem anderen Cluster verwaltet wird. Wenn beide Cluster dasselbe Volume verwalten, können widersprüchliche Vorgänge die Daten

beschädigen oder zerstören.

ONTAP SAN

Trident unterstützt den Volume-Import mit den `ontap-san` (iSCSI, NVMe/TCP und FC) und `ontap-san-economy`-Treibern.

Trident kann ONTAP SAN FlexVol Volumes importieren, die eine einzelne LUN enthalten. Dies entspricht dem `ontap-san` Treiber, der für jede PVC ein FlexVol Volume erstellt und eine LUN innerhalb des FlexVol Volumes anlegt. Trident importiert das FlexVol Volume und ordnet es der PVC-Definition zu. Trident kann `ontap-san-economy` Volumes importieren, die mehrere LUNs enthalten.

Die folgenden Beispiele zeigen, wie verwaltete und nicht verwaltete Volumes importiert werden:

Verwaltetes Volume

Für verwaltete Volumes benennt Trident das FlexVol Volume in das `pvc-<uuid>` Format um und die LUN innerhalb des FlexVol Volume in `lun0` um.

Das folgende Beispiel importiert das `ontap-san-managed` FlexVol Volume, das sich auf dem `ontap_san_default` Backend befindet:

```
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 Volume

Das folgende Beispiel importiert `unmanaged_example_volume` auf dem `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 denselben IQN wie ein Kubernetes-Knoten-IQN verwenden (siehe folgendes Beispiel), erhalten Sie die Fehlermeldung: `LUN already mapped to initiator(s) in this group`. Sie müssen den Initiator entfernen oder die Zuordnung der LUN aufheben, um das Volume zu importieren.

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

ONTAP SAN-economy Beispiele

Die folgenden Beispiele zeigen, wie verwaltete und nicht verwaltete Volumes für das `ontap-san-economy` Backend importiert werden.

Verwaltetes Volume

Wenn Sie ein verwaltetes Volume importieren, übernimmt Trident die Eigentümerschaft des FlexVol und benennt es um. Sie müssen diese Umbenennung berücksichtigen, wenn Sie mehrere LUNs aus demselben FlexVol importieren.

Das folgende Beispiel importiert `lun1` aus der FlexVol `toimport` als verwaltetes Volume mit dem Namen `vol-managed-saneco`:

```
tridentctl import volume vol-managed-saneco toimport/lun1 -f
import1.yaml
```

Nach dem Import `lun1` benennt Trident die FlexVol (zum Beispiel in `trident_lun_pool_xyz`) um. Um zusätzliche LUNs aus derselben FlexVol zu importieren, verwenden Sie den neuen FlexVol-Namen:

```
tridentctl import volume vol-managed-saneco trident_lun_pool_xyz/lun2
-f import2.yaml
```

HINWEIS

Das `ontap-san-economy` Backend importiert jeweils eine LUN. Sie können mehrere Importe mithilfe eines Skripts automatisieren.

Unverwaltetes Volume

Wenn Sie ein nicht verwaltetes Volume importieren, übernimmt Trident nicht die Besitzrechte an dem FlexVol. Allerdings müssen das FlexVol und die LUN den Trident-Namenskonventionen entsprechen.

FlexVol-Benennungsformat

```
trident_lun_pool_STORAGEPREFIX_RANDOMSTRING
```

- `STORAGEPREFIX` ist der Wert von `storagePrefix` in Ihrer Backend-Konfiguration. Der Standardwert ist `trident`.
- `RANDOMSTRING` ist eine beliebige Zeichenkette Ihrer Wahl.

LUN-Benennungsanforderung

Der LUN muss mit einem Namen versehen werden `lun0`.

Beispiel

Wenn Ihre `storagePrefix xyz` ist, lautet der vollständige Pfad zur LUN:

```
trident_lun_pool_xyz_randomstring/lun0
```

Element

Trident unterstützt NetApp Element Software und NetApp HCI Volume-Import mit dem `solidfire-san` Treiber.

HINWEIS

Der Element-Treiber unterstützt doppelte Volume-Namen. Trident gibt jedoch einen Fehler zurück, wenn doppelte Volume-Namen vorhanden sind. Als Workaround klonen Sie das Volume, vergeben einen eindeutigen Volume-Namen und importieren das geklonte Volume.

Das folgende Beispiel importiert ein `element-managed` Volume auf 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  |  MANAGED  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block      | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Azure NetApp Files

Trident unterstützt den Volumenimport mithilfe des `azure-netapp-files` Treibers.

HINWEIS

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 der Mount-Pfad beispielsweise `10.0.0.2:/importvoll1` ist, ist der Volume-Pfad `importvoll1`.

Das folgende Beispiel importiert ein `azure-netapp-files` Volume auf Backend `azurenetaappfiles_40517` mit dem Volume-Pfad `importvoll1`.

```
tridentctl import volume azurenetaappfiles_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 mithilfe des `google-cloud-netapp-volumes` Treibers.

Das folgende Beispiel importiert ein Volume auf Backend `backend-tbc-gcnv1` mit dem Volume `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` Volume, wenn sich zwei Volumes in derselben Region befinden:

```
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
|
+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
```

Anpassung von Trident Volume-Namen und Labels

Mit Trident können Sie den von Ihnen erstellten Volumes aussagekräftige Namen und Bezeichnungen zuweisen. Dies hilft Ihnen, Volumes zu identifizieren und sie einfach ihren jeweiligen Kubernetes-Ressourcen (PVCs) zuzuordnen. Sie können außerdem auf Backend-Ebene Vorlagen für benutzerdefinierte Volume-Namen und benutzerdefinierte Bezeichnungen definieren; alle Volumes, die Sie erstellen, importieren oder klonen, halten sich an diese Vorlagen.

Bevor Sie beginnen

Unterstützung für anpassbare Volume-Namen und Labels:

- Volume-Erstellungs-, Import- und Klonvorgänge.
- Im Fall des `ontap-nas-economy`-Treibers entspricht nur der Name des Qtree-Volumes der Namensvorlage.
- Im Falle des `ontap-san-economy` Treibers entspricht nur der LUN-Name der Namensvorlage.

Einschränkungen

- Benutzerdefinierte Volumennamen sind nur mit ONTAP-Treibern für lokale Installationen kompatibel.
- Benutzerdefinierte Bezeichnungen werden nur für die `ontap-san`, `ontap-nas` und `ontap-nas-flexgroup` Treiber unterstützt.
- Benutzerdefinierte Volumennamen gelten nicht für vorhandene Volumen.

Wichtige Verhaltensweisen von anpassbaren Volumennamen

- Tritt aufgrund ungültiger Syntax in einer Namensvorlage ein Fehler auf, schlägt die Backend-Erstellung fehl. Schlägt jedoch die Vorlagenanwendung fehl, wird das Volume gemäß der bestehenden Namenskonvention benannt.
- Das Speicherpräfix ist nicht anwendbar, wenn ein Volume mithilfe einer Namensvorlage aus der Backend-Konfiguration benannt wird. Jeder gewünschte Präfixwert kann direkt zur Vorlage hinzugefügt werden.

Backend-Konfigurationsbeispiele mit Namensvorlage und Labels

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 }}_{{ .config.BackendName }}"
```

```
slice .volume.RequestName 1 5 }}""
```

Zu berücksichtigende Punkte

1. Bei Datenträgerimporten werden die Bezeichnungen nur dann aktualisiert, wenn der vorhandene Datenträger Bezeichnungen in einem bestimmten Format aufweist. Beispiel:
`{"provisioning":{"Cluster":"ClusterA", "PVC": "pvcname"}}`.
2. Bei verwalteten Volume-Importen folgt der Volume-Name der Namensvorlage, die auf der Root-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 Volume-Namen ergeben, fügt Trident einige zufällige Zeichen hinzu, um eindeutige Volume-Namen 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 Trident NFS Volume über Namespaces hinweg freigeben

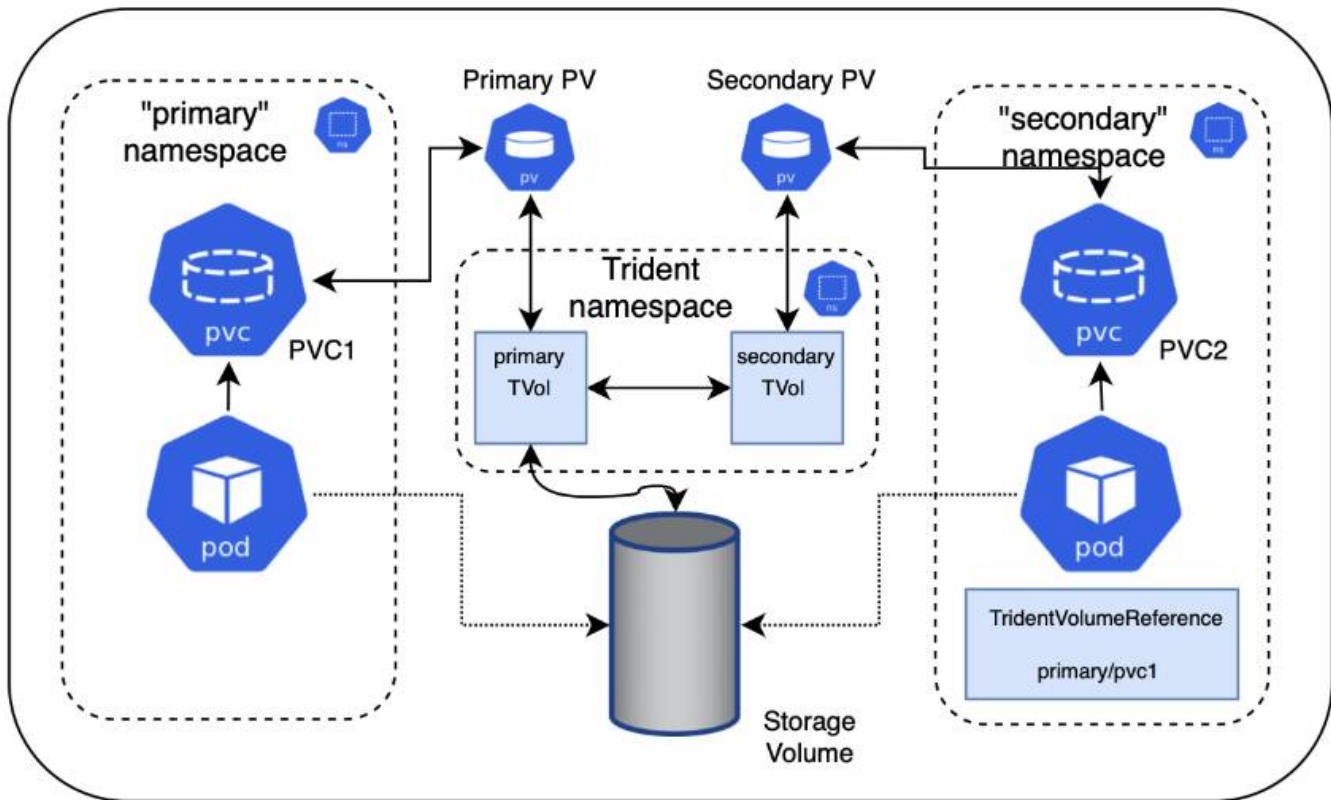
Mit Trident können Sie ein Volume in einem primären Namespace erstellen und es in einem oder mehreren sekundären Namespaces freigeben.

Features

Das TridentVolumeReference CR ermöglicht es Ihnen, ReadWriteMany (RWX)-NFS-Volumes sicher über einen oder mehrere Kubernetes-Namespaces hinweg freizugeben. 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

Sie können die NFS-Volume-Freigabe in nur wenigen Schritten einrichten.

1

Konfigurieren Sie den Quell-PVC, um das Volume freizugeben

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 TridentVolumeReference im Ziel-Namespace

Der Eigentümer des Ziel-Namensraums 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 über verschiedene Namespaces 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:** Erstellen Sie die PVC (pvc1 im Quellnamespace, die die Berechtigung zur gemeinsamen Nutzung mit dem Zielnamespace (namespace2 mithilfe der `shareToNamespace` Annotation gewährt.

```
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 Backend-NFS-Speichervolume.

HINWEIS

- Sie können die PVC mithilfe einer durch Kommas getrennten Liste für mehrere Namensräume freigeben. Zum Beispiel
`trident.netapp.io/shareToNamespace: namespace2, namespace3, namespace4.`
- Sie können für alle Namensräume freigeben *. Zum Beispiel,
`trident.netapp.io/shareToNamespace: *`
- Sie können die PVC jederzeit aktualisieren, um die `shareToNamespace` Annotation einzuschließen.

2. **Cluster-Administrator:** Stellen Sie sicher, dass die korrekte RBAC-Konfiguration vorhanden ist, um dem Besitzer des Ziel-Namespace die Berechtigung zum Erstellen des `TridentVolumeReference` CR im Ziel-Namespace zu erteilen.
3. **Ziel-Namespace-Inhaber:** Erstellen Sie eine `TridentVolumeReference` CR im Ziel-Namespace, 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. **Ziel-Namespace-Inhaber:** Erstellen Sie einen PVC (pvc2 im Ziel-Namespace (namespace2 unter Verwendung der shareFromPVC-Annotation, um den Quell-PVC zu kennzeichnen.

```

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

```

HINWEIS

Die Größe des Ziel-PVC muss kleiner oder gleich der des Quell-PVC sein.

Ergebnisse

Trident liest die shareFromPVC Annotation auf dem Ziel-PVC und erstellt das Ziel-PV als untergeordnetes Volume ohne eigene Speicherressource, das auf das Quell-PV verweist und die Speicherressource des Quell-PV mitnutzt. Das Ziel-PVC und -PV erscheinen wie gewohnt verbunden.

Löschen eines freigegebenen Volumes

Sie können ein Volume löschen, das von mehreren Namespaces gemeinsam genutzt wird. Trident entfernt den Zugriff auf das Volume im Quell-Namespace und erhält den Zugriff für andere Namespaces, die das Volume gemeinsam nutzen. Wenn alle Namespaces, die auf das Volume verweisen, entfernt wurden, löscht Trident das Volume.

Verwenden Sie `tridentctl get` zur Abfrage untergeordneter Volumes

Mit dem [tridentctl-Dienstprogramm kann der `get`-Befehl ausgeführt werden, um untergeordnete Volumes abzurufen. Weitere Informationen finden Sie unter `tridentctl` [commands and options](#).

```

Usage:
  tridentctl get [option]

```

Flags:

- `-h, --help`: Hilfe für Bände.
- `--parentOfSubordinate string`: Beschränken Sie die Abfrage auf das untergeordnete Quellvolume.
- `--subordinateOf string`: Beschränken Sie die Abfrage auf untergeordnete Elemente des Volumens.

Einschränkungen

- Trident kann nicht verhindern, dass Ziel-Namespace auf das gemeinsam genutzte Volume schreiben. Sie sollten Dateisperren oder andere Verfahren verwenden, um das Überschreiben von Daten auf dem gemeinsam genutzten Volume zu verhindern.
- Sie können den Zugriff auf die Quell-PVC nicht widerrufen, indem Sie die `shareToNamespace` oder `shareFromNamespace` Anmerkungen entfernen oder die `TridentVolumeReference` CR löschen. Um den Zugriff zu widerrufen, müssen Sie die untergeordnete PVC löschen.
- Snapshots, Klone und Spiegelungen sind auf untergeordneten Volumes nicht möglich.

Für weitere Informationen

Um mehr über den Volume-Zugriff über Namespaces hinweg zu erfahren:

- Sehen Sie sich die Demo auf ["NetAppTV"](#) an.

Klonen von Trident Volumes über Namespaces hinweg

Mit Trident können Sie neue Volumes erstellen, indem Sie vorhandene Volumes oder Volumesnapshots aus einem anderen Namespace innerhalb desselben Kubernetes-Clusters verwenden.

Voraussetzungen

Bevor Sie Volumes klonen, stellen Sie sicher, dass das Quell- und das Ziel-Backend vom gleichen Typ sind und die gleiche Speicherklasse haben.

HINWEIS

Das Klonen über Namensräume hinweg wird nur für die `ontap-san` und `ontap-nas` Speichertreiber unterstützt. Schreibgeschützte Klone werden nicht unterstützt.

Schnellstart

Das Klonen von Volumes lässt sich in wenigen Schritten einrichten.

1

Konfigurieren Sie die Quell-PVC, um das Volume zu klonen

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 TridentVolumeReference im Ziel-Namespace

Der Eigentümer des Ziel-Namensraums 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-Inhabers, des Cluster-Administrators und des Ziel-Namespace-Inhabers. Die Benutzerrolle wird in jedem Schritt festgelegt.

Schritte

1. **Quellnamespace-Inhaber:** Erstellen Sie die PVC (pvc1 im Quellnamespace (namespace1, die die Berechtigung zur gemeinsamen Nutzung mit dem Zielnamespace (namespace2 mithilfe der cloneToNamespace-Annotation gewährt.

```
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 dessen Backend-Speichervolume.

HINWEIS

- Sie können die PVC mithilfe einer durch Kommas getrennten Liste für mehrere Namensräume freigeben. Zum Beispiel
trident.netapp.io/cloneToNamespace:
namespace2, namespace3, namespace4.
- Sie können für alle Namensräume freigeben *. Zum Beispiel,
trident.netapp.io/cloneToNamespace: *
- Sie können die PVC jederzeit aktualisieren, um die cloneToNamespace Annotation einzuschließen.

2. **Cluster-Administrator:** Stellen Sie sicher, dass die korrekte RBAC-Konfiguration vorhanden ist, um dem Besitzer des Ziel-Namespace die Berechtigung zum Erstellen des TridentVolumeReference CR im Ziel-Namespace zu erteilen (namespace2).
3. **Ziel-Namespace-Inhaber:** Erstellung einer TridentVolumeReference CR im Ziel-Namespace, die auf die Quell-PVC `pvc1` verweist.

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Ziel-Namespace-Inhaber:** Erstellen Sie einen PVC (`pvc2`) im Ziel-Namespace (`namespace2`) mit der Verwendung von `cloneFromPVC` oder `cloneFromSnapshot` und `cloneFromNamespace` Annotationen, um den Quell-PVC zu kennzeichnen.

```
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, sind schreibgeschützte Klone nicht unterstützt.

Volumes replizieren mit SnapMirror

Trident unterstützt Spiegelungsbeziehungen zwischen einem Quellvolume in einem Cluster und dem Zielvolume im verbundenen Cluster zur Datenreplikation für die Notfallwiederherstellung. Sie können eine namensbasierte Custom Resource Definition

(CRD), genannt Trident Mirror Relationship (TMR), verwenden, um die folgenden Operationen durchzuführen:

- Spiegelbeziehungen zwischen Volumes (PVCs) erstellen
- Spiegelbeziehungen zwischen Volumes entfernen
- Spiegelbeziehungen aufheben
- Das sekundäre Volume während Katastrophenbedingungen (Failover) promoten
- 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:** Trident Version 22.10 oder höher muss auf beiden Quell- und Ziel-Kubernetes-Clustern vorhanden sein, die ONTAP als Backend nutzen.
- **Lizenzen:** ONTAP SnapMirror asynchrone Lizenzen mit dem Data Protection Bundle müssen sowohl auf dem Quell- als auch auf dem Ziel-ONTAP-Cluster aktiviert sein. Weitere Informationen finden Sie unter ["SnapMirror Lizenzierungsübersicht in ONTAP"](#).

Ab ONTAP 9.10.1 werden alle Lizenzen als NetApp Lizenzdatei (NLF) bereitgestellt, bei der es sich um eine einzelne Datei handelt, die mehrere Funktionen aktiviert. Weitere Informationen finden Sie unter ["In ONTAP One enthaltene Lizenzen"](#).

HINWEIS

Es wird ausschließlich SnapMirror asynchroner Schutz unterstützt.

Peering

- **Cluster und SVM:** Die ONTAP-Speicher-Backends müssen per Peering verbunden sein. Weitere Informationen finden Sie unter ["Cluster- und SVM-Peering-Übersicht"](#).

WICHTIG

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 Ziel-Cluster verfügbar sein.

Unterstützte Treiber

NetApp Trident unterstützt die Volume-Replikation mit NetApp SnapMirror-Technologie unter Verwendung von Storage-Klassen, die von den folgenden Treibern unterstützt werden: **ontap-nas: NFS** `ontap-san: iSCSI` **ontap-san: FC** `ontap-san: NVMe/TCP` (erfordert mindestens ONTAP Version 9.15.1)

HINWEIS

Die Volume-Replikation mit SnapMirror wird für ASA r2-Systeme nicht unterstützt. Weitere Informationen zu ASA r2-Systemen finden Sie unter ["Erfahren Sie mehr über ASA r2-Speichersysteme"](#).

Erstellen Sie ein gespiegeltes PVC

Folgen Sie diesen Schritten und verwenden Sie die CRD-Beispiele, um eine Spiegelbeziehung zwischen

primären und sekundären Volumes zu erstellen.

Schritte

1. Führen Sie die folgenden Schritte auf dem primären Kubernetes-Cluster aus:
 - a. Erstelle 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 zuvor erstellter 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 des MirrorRelationship.

- d. Rufen Sie das TridentMirrorRelationship CR ab, um den internen Namen und die SVM des PVC zu erhalten.

```
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 aus:

- a. 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 ein MirrorRelationship CR mit Ziel- und Quellinformationen.

Beispiel

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
```

```

name: csi-nas
spec:
  state: established
  replicationPolicy: MirrorAllSnapshots
  replicationSchedule: 5min
  volumeMappings:
  - localPVCName: csi-nas
    remoteVolumeHandle:
      "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"

```

Das TMR spec verfügt über vier Felder der obersten Ebene: state, das den gewünschten Beziehungsstatus festlegt (siehe [Volumenreplikationszustände](#)); volumeMappings, das lokale PVCs ihren Remote-Volume-Handles zuordnet; sowie die optionalen replicationPolicy und replicationSchedule Felder, die unten beschrieben sind.

Die replicationPolicy und replicationSchedule Felder sind optional. Als Wert für replicationPolicy ist der Name einer asynchronen SnapMirror Richtlinie auf ONTAP und für replicationSchedule der Name eines ONTAP Zeitplans anzugeben. Wenn eines der Felder ausgelassen wird, verwendet Trident die entsprechende Replikationsrichtlinie bzw. den entsprechenden Zeitplan, die bzw. der im Backend konfiguriert ist. Diese Felder sind beim Erstellen der Beziehung festzulegen; die Richtlinie oder der Zeitplan können nach dem Einrichten der Beziehung nicht mehr geändert werden. Wenn andere Einstellungen verwendet werden sollen, ist die TMR zu löschen und neu zu erstellen.

Trident erstellt und initialisiert eine SnapMirror Beziehung anhand der konfigurierten Replikationsrichtlinie und des Zeitplans.

1. Erstellen Sie eine PVC mit zuvor erstellter StorageClass, um als sekundäres (SnapMirror Ziel) zu fungieren.

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 die Erstellung des Volumes fehl, falls die Beziehung nicht existiert. Wenn die Beziehung besteht, stellt Trident sicher, dass das neue FlexVol volume auf einer SVM platziert wird, die mit der Remote-SVM, die in der MirrorRelationship definiert ist, verbunden ist.

Volumenreplikationszustände

Eine Trident Mirror Relationship (TMR) ist eine CRD, die ein Ende einer Replikationsbeziehung zwischen PVCs darstellt. Die Ziel-TMR hat einen Zustand, der Trident den gewünschten Zustand mitteilt. Die Ziel-TMR hat die folgenden Zustände:

- **Festgestellt:** Das lokale PVC ist das Zielvolume einer SnapMirror Beziehung, und dies ist eine neue Beziehung.
- **Promoted:** Die lokale PVC ist ReadWrite und mountbar, es besteht derzeit keine Spiegelbeziehung.
- **Wiederhergestellt:** Das lokale PVC ist das Zielvolume einer SnapMirror Beziehung und war auch zuvor in dieser SnapMirror Beziehung.
 - Der wiederhergestellte Zustand muss verwendet werden, wenn das Zielvolume jemals in einer Beziehung mit dem Quellvolume stand, da er den Inhalt des Zielvolumes überschreibt.
 - Der wiederhergestellte Zustand schlägt fehl, wenn das Volume zuvor nicht in einer Beziehung mit der Quelle stand.

Sekundäres PVC während eines ungeplanten Failovers promoten

Führen Sie den folgenden Schritt auf dem sekundären Kubernetes-Cluster durch:

- Aktualisiere das Feld `spec.state` von `TridentMirrorRelationship` auf `promoted`.

Sekundäre PVC während eines geplanten Failovers promoten

Führen Sie während eines geplanten Failovers (Migration) die folgenden Schritte aus, um den sekundären PVC zu promoten:

Schritte

1. Erstellen Sie im primären Kubernetes-Cluster einen Snapshot des PVC und warten Sie, bis der Snapshot erstellt wurde.
2. Im primären Kubernetes-Cluster wird der `TridentTridentSnapshotInfoCR` erstellt, um interne Details zu erhalten.

Beispiel

```
kind: TridentSnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshotName csi-nas-snapshot
```

- a. Im sekundären Kubernetes-Cluster das Feld `spec.state` der `TridentMirrorRelationship` CR auf `promoted` und `spec.volumeMappings[].promotedSnapshotHandle` auf den `internalName` des Snapshots aktualisieren.
3. Bestätigen Sie auf dem sekundären Kubernetes-Cluster den Status (`status.state`-Feld) von `TridentMirrorRelationship` auf `promoted`.

Eine Spiegelbeziehung nach einem Failover wiederherstellen

Bevor Sie die Spiegelbeziehung wiederherstellen, wählen Sie die Seite aus, die Sie als neue primäre Seite festlegen möchten.

Schritte

1. Stellen Sie auf dem sekundären Kubernetes-Cluster sicher, dass die Werte für das Feld `spec.remoteVolumeHandle` auf der `TridentMirrorRelationship` aktualisiert werden.
2. Auf dem sekundären Kubernetes-Cluster muss das Feld `spec.state` von `TridentMirrorRelationship` auf `reestablished` aktualisiert werden.

Zusätzliche Operationen

Trident unterstützt die folgenden Operationen auf den primären und sekundären Volumes:

Primäres PVC auf ein neues sekundäres PVC replizieren

Stellen Sie sicher, dass Sie bereits ein primäres PVC und ein sekundäres PVC 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 das neue sekundäre (Ziel-)PVC, das Sie einrichten möchten.

Ändern Sie die 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 von 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 ein PVC (das 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 Spiegelungsbeziehung führt dazu, dass der verbleibende TMR in den Status `promoted` wechselt, bevor Trident den Löschvorgang abschließt. Befindet sich der zum Löschen ausgewählte TMR bereits im Status `promoted`, besteht keine Spiegelungsbeziehung mehr, der TMR wird entfernt und Trident stuft den lokalen PVC auf `ReadWrite` hoch. Durch diese Löschung werden SnapMirror-Metadaten für das lokale Volume in ONTAP freigegeben. Wird dieses Volume zukünftig in einer Spiegelungsbeziehung verwendet, muss beim Erstellen der neuen Spiegelungsbeziehung ein neuer TMR mit

einem *established* Volume-Replikationsstatus verwendet werden.

Spiegelbeziehungen aktualisieren, wenn ONTAP online ist

Spiegelungsbeziehungen können jederzeit nach ihrer Einrichtung aktualisiert werden. Sie können die Felder `state: promoted` oder `state: reestablished` verwenden, um die Beziehungen zu aktualisieren. Wenn Sie ein Zielvolume zu einem regulären ReadWrite-Volume hochstufen, 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 benötigt. Beachten Sie das folgende Beispielformat von `TridentActionMirrorUpdate`:

Beispiel

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

`status.state` spiegelt den Status der `TridentActionMirrorUpdate` CRD wider. Es kann den Wert *Succeeded*, *In Progress* oder *Failed* annehmen.

CSI-Topologie verwenden

Trident kann selektiv Volumes erstellen und an Knoten in einem Kubernetes-Cluster anhängen, indem es die ["CSI-Topologiefunktion"](#) verwendet.

Überblick

Mithilfe der CSI-Topologie-Funktion lässt sich der Zugriff auf Volumes auf eine Teilmenge von Knoten beschränken, basierend auf Regionen und Verfügbarkeitszonen. Cloud-Anbieter ermöglichen es Kubernetes-Administratoren heute, zonenbasierte Knoten zu erstellen. Knoten können sich in verschiedenen Verfügbarkeitszonen innerhalb einer Region oder über verschiedene Regionen hinweg befinden. Um die Bereitstellung von Volumes für Workloads in einer Multi-Zonen-Architektur zu erleichtern, verwendet Trident die CSI-Topologie.

TIPP | Erfahren Sie mehr über die CSI Topology-Funktion ["hier"](#).

Kubernetes bietet zwei einzigartige Volume-Bindungsmodi:

- Mit `VolumeBindingMode` auf `Immediate` gesetzt, erstellt Trident das Volume ohne jegliche Topologiebewusstheit. Volume-Bindung und dynamische Bereitstellung erfolgen, wenn das PVC erstellt wird. Dies ist die Standardeinstellung `VolumeBindingMode` und eignet sich für Cluster, die keine Topologiebeschränkungen erzwingen. Persistente Volumes werden erstellt, ohne von den

Planungsanforderungen des anfordernden Pods abhängig zu sein.

- Mit `VolumeBindingMode` auf `WaitForFirstConsumer` gesetzt, wird die Erstellung und Bindung eines Persistent Volume für eine PVC verzögert, bis ein Pod, der die PVC verwendet, geplant und erstellt wird. So werden Volumes erstellt, um die durch Topologieanforderungen erzwungenen Planungsbeschränkungen zu erfüllen.

HINWEIS

Der `WaitForFirstConsumer` Bindungsmodus benötigt keine Topologiebezeichnungen. Dies kann unabhängig von der CSI-Topologiefunktion verwendet werden.

Was Sie benötigen

Um die CSI Topology 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:"1e11e4a2108024935ecfcb2912226cedaafd99df",
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:"1e11e4a2108024935ecfcb2912226cedaafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- Die Knoten im Cluster sollten Labels aufweisen, die Topologiebewusstsein ermöglichen (`topology.kubernetes.io/region` und `topology.kubernetes.io/zone`). Diese Labels **müssen auf den Knoten im Cluster vorhanden sein**, bevor Trident installiert wird, damit Trident topologiebewusst funktioniert.

```
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 eines topologiebewussten Backends

Trident-Speicher-Backends können so konzipiert werden, dass sie Volumes selektiv basierend auf Verfügbarkeitszonen bereitstellen. Jedes Backend kann einen optionalen `supportedTopologies` Block enthalten, der eine Liste der unterstützten Zonen und Regionen repräsentiert. Für `StorageClasses`, die ein solches Backend verwenden, würde ein Volume nur erstellt werden, 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"
    }
  ]
}

```

HINWEIS

`supportedTopologies` wird verwendet, um 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 in einem Backend bereitgestellten Regionen und Zonen enthalten, erstellt Trident ein Volume auf dem Backend.

Sie können `supportedTopologies` pro Speicherpool ebenfalls definieren. Siehe das folgende Beispiel:

```

---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-central1
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-a
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-b

```

In diesem Beispiel stehen die `region` und `zone` Bezeichnungen für den Speicherort des Speicherpools. `topology.kubernetes.io/region` und `topology.kubernetes.io/zone` legen fest, von wo aus auf die Speicherpools zugegriffen werden kann.

Schritt 2: Definieren Sie StorageClasses, die topologiebewusst sind

Anhand der den Knoten im Cluster zugewiesenen Topologiebezeichnungen 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:
  name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions:
    - 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 wird `volumeBindingMode` auf `WaitForFirstConsumer` gesetzt. PVCs, die mit dieser StorageClass angefordert werden, werden erst verarbeitet, wenn sie in einem Pod referenziert werden. Und `allowedTopologies` gibt die zu verwendenden Zonen und die Region an. Die `netapp-san-us-east1` StorageClass erstellt PVCs auf dem `san-backend-us-east1` oben definierten Backend.

Schritt 3: Eine PVC erstellen und verwenden

Nachdem die StorageClass erstellt und einem Backend zugeordnet wurde, können Sie jetzt 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

```

Das Erstellen eines PVC mit diesem Manifest würde zu Folgendem führen:

```

kubectl create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubectl get pvc

```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
pvc-san	Pending				netapp-san-us-east1

```

2s
kubectl 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 Volume zu erstellen und es an das PVC zu binden, verwenden Sie das PVC in einem Pod. 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

```

Dieses podSpec weist Kubernetes an, den Pod auf Knoten zu planen, die in der us-east1 Region vorhanden sind, und aus beliebigen Knoten auszuwählen, die in der us-east1-a oder us-east1-b Zone vorhanden sind.

Siehe die 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
```

Backends aktualisieren, um `supportedTopologies` einzuschließen

Vorhandene Backends können aktualisiert werden, um eine Liste von `supportedTopologies` mit `tridentctl backend update` einzuschließen. Dies hat keine Auswirkungen auf bereits bereitgestellte Volumes und wird nur für nachfolgende PVCs verwendet.

Weitere Informationen

- ["Ressourcen für Container verwalten"](#)
- ["nodeSelector"](#)
- ["Affinität und Anti-Affinität"](#)
- ["Taints und Tolerations"](#)

Arbeiten mit Trident Snapshots

Kubernetes-Volume-Snapshots von Persistent Volumes (PVs) ermöglichen zeitpunktgenaue 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-Snapshot wird von `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `azure-netapp-files` und `google-cloud-netapp-volumes` Treibern unterstützt.

Bevor Sie beginnen

Sie benötigen einen externen Snapshot-Controller und benutzerdefinierte Ressourcendefinitionen (CRDs), um mit Snapshots zu arbeiten. Dies ist die Aufgabe des Kubernetes-Orchestrators (zum Beispiel: Kubeadm, GKE, OpenShift).

Falls Ihre Kubernetes-Distribution den Snapshot-Controller und die CRDs nicht enthält, lesen Sie [Einen Volume Snapshot Controller bereitstellen](#).

HINWEIS

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` verweist auf den Trident CSI driver.

- `deletionPolicy` kann `Delete` oder `Retain` sein. Wenn auf `Retain` gesetzt, bleibt der zugrunde liegende physische Snapshot auf dem Speichercluster auch dann erhalten, wenn das `VolumeSnapshot` Objekt gelöscht wird.

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 Snapshot 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 `VolumeSnapshot`-Objekt für eine PVC mit dem Namen `pvc1` und der Name des Snapshots wird auf `pvc1-snap` gesetzt. Ein `VolumeSnapshot` ist analog zu einer PVC und ist mit einem `VolumeSnapshotContent` Objekt verknüpft, das den eigentlichen Snapshot darstellt.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME                AGE
pvc1-snap           50s
```

- Sie können das `VolumeSnapshotContent` Objekt für das `pvc1-snap` `VolumeSnapshot`

identifizieren, indem Sie es beschreiben. Das `Snapshot Content Name` identifiziert das `VolumeSnapshotContent` Objekt, das diesem Snapshot dient. Der `Ready To Use` Parameter zeigt 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 Volume-Snapshot

Sie können `dataSource` verwenden, um eine PVC mithilfe eines `VolumeSnapshot` mit dem Namen `<pvc-name>` als Datenquelle zu erstellen. Nachdem die PVC erstellt wurde, kann sie einem Pod zugeordnet und wie jede andere PVC verwendet werden.

WARNUNG

Die PVC wird im selben Backend wie das Quellvolume erstellt. Weitere Informationen enthält ["KB: Die Erstellung eines PVC aus einem Trident PVC Snapshot kann in einem alternativen Backend nicht durchgeführt werden"](#).

Im folgenden Beispiel wird die PVC unter Verwendung von `pvc1-snap` als Datenquelle erstellt.

```
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: pvc1-snap
  kind: VolumeSnapshot
  apiGroup: snapshot.storage.k8s.io

```

Importieren eines Volume-Snapshots

Trident unterstützt die ["Kubernetes pre-provisioned Snapshot-Prozess"](#), damit der Cluster-Administrator ein VolumeSnapshotContent-Objekt erstellen und außerhalb von Trident erstellte Snapshots importieren kann.

Bevor Sie beginnen

Trident muss das übergeordnete Volume des Snapshots erstellt oder importiert haben.

Schritte

1. **Cluster-Administrator:** Erstellen Sie ein VolumeSnapshotContent Objekt, das auf den Backend-Snapshot verweist. Dadurch wird der Snapshot-Workflow in Trident gestartet.
 - Geben Sie den Namen des Backend-Snapshots in annotations als `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name"> an.`
 - Geben Sie `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` in `snapshotHandle` an. Dies ist die einzige Information, die Trident vom externen Snapshotter im `ListSnapshots`-Aufruf bereitgestellt wird.

HINWEIS

Der `<volumeSnapshotContentName>` kann aufgrund von Namensbeschränkungen der CRs nicht immer mit dem Backend-Snapshot-Namen übereinstimmen.

Beispiel

Das folgende Beispiel erstellt ein VolumeSnapshotContent Objekt, das auf einen Backend-Snapshot `snap-01` verweist.

```

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 das VolumeSnapshot CR, das auf das VolumeSnapshotContent Objekt verweist. Dadurch wird der Zugriff auf die Verwendung des VolumeSnapshot in einem bestimmten Namespace angefordert.

Beispiel

Das folgende Beispiel erstellt ein VolumeSnapshot CR mit dem Namen import-snap, das auf das VolumeSnapshotContent mit dem Namen import-snap-content verweist.

```
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 führt den ListSnapshots-Aufruf aus. Trident erstellt das TridentSnapshot.
 - Der externe Snapshotter setzt das VolumeSnapshotContent auf readyToUse und das VolumeSnapshot auf true.
 - Trident kehrt zurück readyToUse=true.
4. **Beliebiger Benutzer:** Erstellen Sie ein PersistentVolumeClaim, um auf das neue VolumeSnapshot zu verweisen, wobei der spec.dataSource (oder spec.dataSourceRef) Name der VolumeSnapshot Name ist.

Beispiel

Das folgende Beispiel erstellt eine PVC, die auf die VolumeSnapshot mit dem Namen import-snap verweist.

```
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
```

Stellen Sie Volumendaten mithilfe von Snapshots wieder her

Das Snapshot-Verzeichnis ist standardmäßig ausgeblendet, um maximale Kompatibilität von Volumes zu gewährleisten, die mit den `ontap-nas` und `ontap-nas-economy`-Treibern bereitgestellt wurden. Aktivieren Sie das `.snapshot`-Verzeichnis, um Daten direkt aus Snapshots wiederherzustellen.

Verwenden Sie die ONTAP CLI `volume snapshot restore`, um ein Volume auf einen Zustand wiederherzustellen, der in einem früheren Snapshot aufgezeichnet wurde.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

HINWEIS

Beim Wiederherstellen einer Snapshot-Kopie wird die bestehende Volume-Konfiguration überschrieben. Änderungen an den Volume-Daten, die nach der Erstellung der Snapshot-Kopie vorgenommen wurden, gehen verloren.

In-Place-Volume-Wiederherstellung aus einem Snapshot

Trident ermöglicht die schnelle, direkte Wiederherstellung von Volumes aus einem Snapshot mithilfe des `TridentActionSnapshotRestore` (TASR) CR. Dieser CR fungiert als imperative Kubernetes-Aktion und bleibt nach Abschluss des Vorgangs nicht erhalten.

Trident unterstützt die Snapshot-Wiederherstellung auf den `ontap-san`, `ontap-san-economy`, `ontap-nas`, `ontap-nas-flexgroup`, `azure-netapp-files`, `google-cloud-netapp-volumes` und `solidfire-san`-Treibern.

Bevor Sie beginnen

Sie müssen ein gebundenes PVC und einen verfügbaren Volume Snapshot haben.

- Überprüfen Sie, ob der PVC-Status gebunden ist.

```
kubectl get pvc
```

- Prüfen Sie, ob der Volume-Snapshot einsatzbereit ist.

```
kubectl get vs
```

Schritte

1. Erstellen Sie den TASR CR. Dieses Beispiel erstellt einen CR für PVC `pvc1` und Volume Snapshot `pvc1-snapshot`.

HINWEIS

Der TASR CR muss sich in einem Namespace befinden, in dem 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 aus dem Snapshot wiederherzustellen. Dieses Beispiel stellt aus dem Snapshot `pvc1` wieder her.

```
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: ""

```

HINWEIS

- In den meisten Fällen wird Trident den Vorgang bei einem Fehler nicht automatisch wiederholen. Sie müssen den Vorgang erneut ausführen.
- 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 Persistent Volume mit zugehörigen Snapshots wird das entsprechende Trident volume in den Status „Deleting state“ versetzt. Entfernen Sie die Volume-Snapshots, um das Trident volume zu löschen.

Einen Volume Snapshot Controller bereitstellen

Falls Ihre Kubernetes-Distribution den Snapshot-Controller und die CRDs nicht enthält, können Sie sie 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.yam
l
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
```

HINWEIS

Öffnen Sie bei Bedarf `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` und aktualisieren Sie namespace auf Ihren Namespace.

Verwandte Links

- ["Volume Snapshots"](#)
- ["VolumeSnapshotClass"](#)

Arbeiten mit Trident Volume-Gruppen-Snapshots

Kubernetes-Volume-Gruppen-Snapshots von Persistent Volumes (PVs) NetApp Trident bietet die Möglichkeit, Snapshots mehrerer Volumes (einer Gruppe von Volume-Snapshots) zu erstellen. Dieser Volume-Gruppen-Snapshot stellt Kopien mehrerer Volumes dar, die zum gleichen Zeitpunkt erstellt wurden.

HINWEIS

VolumeGroupSnapshot ist eine Beta-Funktion in Kubernetes mit Beta-APIs. Kubernetes 1.32 ist die Mindestversion, die für VolumeGroupSnapshot erforderlich ist.

Snapshots von Volumengruppen erstellen

Snapshot für Volumengruppen wird mit den folgenden Speichertreibern unterstützt:

- `ontap-san` Treiber – nur für die iSCSI- und FC-Protokolle, nicht für das NVMe/TCP-Protokoll.
- `ontap-san-economy` - nur für das iSCSI-Protokoll.
- `ontap-nas`

HINWEIS

Volume-Gruppen-Snapshot wird für NetApp ASA r2 oder AFX-Speichersysteme nicht unterstützt.

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 zu arbeiten. Dies ist die Aufgabe des Kubernetes-Orchestrators (zum Beispiel: Kubeadm, GKE, OpenShift).

Falls Ihre Kubernetes-Distribution den externen Snapshot-Controller und die CRDs nicht enthält, lesen Sie [Einen Volume Snapshot Controller bereitstellen](#).

HINWEIS

Erstellen Sie keinen Snapshot-Controller, wenn Sie in einer GKE-Umgebung bedarfsgesteuerte Snapshots von Volumengruppen erstellen. GKE verwendet einen integrierten, versteckten Snapshot-Controller.

- Setzen Sie im Snapshot-Controller-YAML das `CSIVolumeGroupSnapshot` Feature-Gate auf 'true', um sicherzustellen, dass Volume Group Snapshot 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 demselben SVM befinden, um VolumeGroupSnapshot erstellen zu können.

Schritte

- Erstellen Sie eine VolumeGroupSnapshotClass, bevor Sie eine 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 Labels unter Verwendung vorhandener Storage Classes, oder fügen Sie diese Labels zu bestehenden PVCs hinzu.

Das folgende Beispiel erstellt die PVC unter Verwendung von `pvc1-group-snap` als Datenquelle und der Bezeichnung `consistentGroupSnapshot: groupA`. Definieren Sie den Bezeichnungsschlüssel und -wert entsprechend Ihren Anforderungen.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1-group-snap
  labels:
    consistentGroupSnapshot: groupA
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

```
storageClassName: sc1-1
```

- Erstellen Sie ein VolumeGroupSnapshot mit der gleichen Bezeichnung (consistentGroupSnapshot: groupA wie im PVC angegeben.

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
```

Volumendaten mithilfe eines Gruppen-Snapshots wiederherstellen

Sie können einzelne Persistent Volumes mithilfe der einzelnen Snapshots wiederherstellen, die als Teil des Volume Group Snapshot erstellt wurden. Sie können den Volume Group Snapshot nicht als Einheit wiederherstellen.

Verwenden Sie die ONTAP CLI `volume snapshot restore`, um ein Volume auf einen Zustand wiederherzustellen, der in einem früheren Snapshot aufgezeichnet wurde.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

HINWEIS

Beim Wiederherstellen einer Snapshot-Kopie wird die bestehende Volume-Konfiguration überschrieben. Änderungen an den Volume-Daten, die nach der Erstellung der Snapshot-Kopie vorgenommen wurden, gehen verloren.

In-Place-Volume-Wiederherstellung aus einem Snapshot

Trident ermöglicht die schnelle, direkte Wiederherstellung von Volumes aus einem Snapshot mithilfe des `TridentActionSnapshotRestore` (TASR) CR. Dieser CR fungiert als imperative Kubernetes-Aktion und bleibt nach Abschluss des Vorgangs nicht erhalten.

Weitere Informationen finden Sie unter ["In-Place-Volume-Wiederherstellung aus einem Snapshot"](#).

Löschen eines PV mit zugehörigen Gruppensnapshots

Beim Löschen eines Gruppenvolume-Snapshots:

- Sie können VolumeGroupSnapshots als Ganzes löschen, nicht einzelne Snapshots in der Gruppe.
- Wenn PersistentVolumes gelöscht werden, während ein Snapshot für dieses PersistentVolume existiert, versetzt Trident dieses Volume in den Status „wird gelöscht“, da der Snapshot entfernt werden muss, bevor das Volume sicher entfernt 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 sie 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
```

HINWEIS

Öffnen Sie bei Bedarf `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` und aktualisieren Sie namespace auf Ihren Namespace.

Verwandte Links

- ["VolumeGroupSnapshotClass"](#)
- ["Volume 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.