Backends konfigurieren und verwalten

Trident

NetApp
January 15, 2026

Inhalt

Backends konfigurieren und verwalten

Backends konfigurieren

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

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

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

Azure NetApp Files

Konfigurieren eines Azure NetApp Files -Backends

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

Details zum Azure NetApp Files Treiber

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

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

Überlegungen

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

Verwaltete Identitäten für AKS

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

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

Trident -Betreiber

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

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

Helm

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

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

<code>tridentctl</code>

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

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

Cloud-Identität für AKS

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

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

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

Trident -Betreiber

Um Trident mithilfe des Trident -Operators zu installieren, bearbeiten Sie `tridentorchestrator_cr.yaml` einstellen `cloudProvider` Zu `"Azure"` und setzen `cloudIdentity` Zu `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx` .

Beispiel:

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

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

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

<code>tridentctl</code>

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

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

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

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

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

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

Voraussetzungen für NFS- und SMB-Volumes

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

Um ein "Azure NetApp Files" Im Backend benötigen Sie Folgendes:

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

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

Benutzerdefinierte Mitwirkenderrolle

```
{
  "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/MountTarge
ts/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/delete",
```

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

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

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

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

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

Zusätzliche Anforderungen für SMB-Volumina

Zum Erstellen eines SMB-Volumes benötigen Sie Folgendes:

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

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

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

Konfigurationsoptionen und Beispiele für das Azure NetApp Files Backend

Erfahren Sie mehr über die NFS- und SMB-Backend-Konfigurationsoptionen für Azure NetApp Files und sehen Sie sich Konfigurationsbeispiele an.

Backend-Konfigurationsoptionen

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

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

Die Azure NetApp Files -Backends bieten diese Konfigurationsoptionen.

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

Parameter	Beschreibung	Standard
`resourceGroups`	Liste der Ressourcengruppen zum Filtern der gefundenen 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 sein `Basic` oder `Standard` . Die Netzwerkfunktionen sind nicht in allen Regionen verfügbar und müssen gegebenenfalls im Rahmen eines Abonnements aktiviert werden. Spezifizierung `networkFeatures` Wenn die Funktionalität nicht aktiviert ist, schlägt die Volumenbereitstellung fehl.	""
`nfsMountOptions`	Feingranulare Steuerung der NFS-Mount-Optionen. Wird bei SMB-Volumes ignoriert. Um Volumes mit NFS Version 4.1 einzubinden, fügen Sie Folgendes hinzu: `nfsvers=4` in der durch Kommas getrennten Mount-Optionsliste die Option NFS v4.1 auswählen. Die in einer Speicherklassendefinition festgelegten Mount-Optionen überschreiben die in der Backend-Konfiguration festgelegten Mount-Optionen.	"nfsvers=3"
`limitVolumeSize`	Die Bereitstellung schlägt fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet.	"" (wird nicht standardmäßig erzwungen)
`debugTraceFlags`	Debug-Flags zur Verwendung bei der Fehlersuche. Beispiel, `\{"api": false, "method": true, "discovery": true}` . Verwenden Sie diese Funktion nur, wenn Sie eine Fehlerbehebung durchführen und einen detaillierten Protokollauszug benötigen.	null

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

Weitere Informationen zu Netzwerkfunktionen finden Sie unter"Konfigurieren von Netzwerkfunktionen für ein Azure NetApp Files Volume" .

Erforderliche Berechtigungen und Ressourcen

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

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

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

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

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

Volume-Bereitstellung

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

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

Beispielkonfigurationen

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

Minimale Konfiguration

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

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

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

Verwaltete Identitäten für AKS

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

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

Cloud-Identität für AKS

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

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

Spezifische Service-Level-Konfiguration mit Kapazitätspoolfiltern

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

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

Backend-Beispiel mit manuellen QoS-Kapazitätspools

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

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: 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 Standardeinstellungen für die Volume-Bereitstellung.

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

Konfiguration eines virtuellen Pools

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

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

Unterstützte Topologiekonfiguration

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

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

Speicherklassendefinitionen

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

Beispieldefinitionen mit `parameter.selector` Feld

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

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

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

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

Beispieldefinitionen für SMB-Volumes

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

Grundkonfiguration im Standard-Namespace

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

Verwendung unterschiedlicher Geheimnisse pro Namensraum

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

Verwendung unterschiedlicher Geheimnisse pro Band

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

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

Backend erstellen

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

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

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

```
tridentctl logs
```

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

Google Cloud NetApp Volumes

Konfigurieren eines Google Cloud NetApp Volumes -Backends

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

Details zum Google Cloud NetApp Volumes -Treiber

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

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

Cloud-Identität für GKE

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

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

- Ein mit GKE bereitgestellter Kubernetes-Cluster.
- Die Workload-Identität wurde im GKE-Cluster konfiguriert und der GKE MetaData Server auf den Knotenpools.

- Ein GCP-Dienstkonto mit der Rolle „Google Cloud NetApp Volumes Admin“ (roles/netapp.admin) oder einer benutzerdefinierten Rolle.
- Trident wurde installiert, einschließlich des Cloud-Providers, der "GCP" angibt, und der Cloud-Identität, die das neue GCP-Dienstkonto angibt. Nachfolgend ein Beispiel.

Trident -Betreiber

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

Beispiel:

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

Helm

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

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

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

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

<code>tridentctl</code>

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

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

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

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

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

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

Voraussetzungen für NFS-Volumes

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

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

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

Weitere Informationen zum Einrichten des Zugriffs auf Google Cloud NetApp Volumes finden Sie unter"Zugriff auf Google Cloud NetApp Volumes einrichten" .

Google Cloud NetApp Volumes Backend-Konfigurationsoptionen und Beispiele

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

Backend-Konfigurationsoptionen

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

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

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

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

Volumenbereitstellungsoptionen

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

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

Beispielkonfigurationen

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

Minimale Konfiguration

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

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

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

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

Konfiguration für SMB-Volumes

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

Konfiguration mit StoragePools-Filter

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

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

Konfiguration eines virtuellen Pools

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

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

Weitere Informationen finden Sie unter"Virtuelle Pools" .

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

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

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

Cloud-Identität für GKE

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

Unterstützte Topologiekonfiguration

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

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

Wie geht es weiter?

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

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

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

```
kubectl get tridentbackendconfig

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

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

```
tridentctl logs
```

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

Speicherklassendefinitionen

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

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

Beispieldefinitionen unter Verwendung der `parameter.selector` Feld:

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

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

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

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

Weitere Einzelheiten zu Speicherklassen finden Sie unter"Erstellen einer Speicherklasse" .

Beispieldefinitionen für SMB-Volumes

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

Grundkonfiguration im Standard-Namespace

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

Verwendung unterschiedlicher Geheimnisse pro Namensraum

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

Verwendung unterschiedlicher Geheimnisse pro Band

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

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

PVC-Definitionsbeispiel

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

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

```
kubectl get pvc gcnv-nfs-pvc

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

Konfigurieren eines Cloud Volumes Service für das Google Cloud-Backend

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

Details zum Google Cloud-Treiber

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

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

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

Trident kann Cloud Volumes Service Volumes in einem von zwei Formaten erstellen."Servicearten" :

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

Was du brauchst

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

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

Backend-Konfigurationsoptionen

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

Parameter	Beschreibung	Standard
`version`		Immer 1
`storageDriverName`	Name des Speichertreibers	"gcp-cvs"
`backendName`	Benutzerdefinierter Name oder das Speicher-Backend	Fahrername + "_" + Teil des API-Schlüssels
`storageClass`	Optionaler Parameter zur Angabe des CVS-Diensttyps. Verwenden `software` um den CVS-Diensttyp auszuwählen. Andernfalls geht Trident vom Diensttyp CVS-Performance aus.(`hardware`).	
`storagePools`	Nur CVS-Dienstleistungstyp. Optionaler Parameter zur Angabe von Speicherpools für die Volume-Erstellung.	

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

Parameter	Beschreibung	Standard

`allowedTopologies`	Um regionsübergreifenden Zugriff zu ermöglichen, muss Ihre StorageClass-Definition für `allowedTopologies` muss alle Regionen umfassen. Zum Beispiel:
	`- key: topology.kubernetes.io/region`
	`values:`
	`- us-east1`
	`- europe-west1`	

Volumenbereitstellungsoptionen

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

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

Beispiele für CVS-Performance-Diensttypen

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

Beispiel 1: Minimale Konfiguration

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

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

Beispiel 2: Konfiguration auf Serviceebene

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

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

Beispiel 3: Konfiguration eines virtuellen Pools

Dieses Beispiel verwendet `storage` virtuelle Pools und die `StorageClasses` die sich auf sie beziehen. SieheSpeicherklassendefinitionen um zu sehen, wie die Speicherklassen definiert wurden.

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

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

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

Speicherklassendefinitionen

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

Beispiel für eine Speicherklasse

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

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

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

Beispiele für CVS-Diensttypen

Die folgenden Beispiele enthalten Beispielkonfigurationen für den CVS-Diensttyp.

Beispiel 1: Minimale Konfiguration

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

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

Beispiel 2: Konfiguration des Speicherpools

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

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

Wie geht es weiter?

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

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

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

```
tridentctl logs
```

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

Konfigurieren Sie ein NetApp HCI oder SolidFire Backend

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

Element-Treiberdetails

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

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

Treiber	Protokoll	Lautstärkemodus	Unterstützte Zugriffsmodi	Unterstützte Dateisysteme
`solidfire-san`	iSCSI	Block	RWO, ROX, RWX, RWOP	Kein Dateisystem. Rohblockgerät.
`solidfire-san`	iSCSI	Dateisystem	RWO, RWOP	`xfs`, `ext3` , `ext4`

Bevor Sie beginnen

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

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

Backend-Konfigurationsoptionen

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

Parameter	Beschreibung	Standard
`version`		Immer 1
`storageDriverName`	Name des Speichertreibers	Immer "solidfire-san"

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

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

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

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

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

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

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

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

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

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

```
SVIP: <svip>:3260
TenantName: <tenant>
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
type: Silver
labels:
  store: solidfire
  k8scluster: dev-1-cluster
region: us-east-1
storage:
  - labels:
      performance: gold
      cost: "4"
    zone: us-east-1a
    type: Gold
  - labels:
      performance: silver
      cost: "3"
    zone: us-east-1b
    type: Silver
  - labels:
      performance: bronze
      cost: "2"
    zone: us-east-1c
    type: Bronze
  - labels:
      performance: silver
      cost: "1"
    zone: us-east-1d
```

Die folgenden StorageClass-Definitionen beziehen sich auf die oben genannten virtuellen Pools. Verwenden

des `parameters.selector` Im Feld „StorageClass“ wird jeweils angegeben, welcher virtuelle Pool (oder welche virtuellen Pools) zum Hosten eines Volumes verwendet werden kann. Das Volumen wird die im gewählten virtuellen Pool definierten Aspekte aufweisen.

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

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

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

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

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

```
  fsType: ext4

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

Weitere Informationen

- "Volumenzugriffsgruppen"

ONTAP SAN-Treiber

ONTAP SAN-Treiberübersicht

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

ONTAP SAN-Treiberdetails

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

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

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

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

Benutzerberechtigungen

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

gleiche Rolle hat. Der `fsxadmin` Der Benutzer ist ein eingeschränkter Ersatz für den Cluster-Administratorbenutzer.

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

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

Weitere Überlegungen zu NVMe/TCP

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

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

Trident unterstützt Folgendes nicht:

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

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

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

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

Anforderungen

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

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

Denken Sie daran, dass Sie auch mehrere Treiber gleichzeitig ausführen und Speicherklassen erstellen

können, die auf den einen oder anderen Treiber verweisen. Beispielsweise könnten Sie Folgendes konfigurieren: `san-dev` Klasse, die die `ontap-san` Fahrer und ein `san-default` Klasse, die die `ontap-san-economy` eins.

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

Authentifizieren Sie das ONTAP Backend

Trident bietet zwei Modi zur Authentifizierung eines ONTAP Backends.

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

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

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

Aktivieren Sie die anmeldeinformationsbasierte Authentifizierung

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

Eine beispielhafte Backend-Definition sieht folgendermaßen aus:

YAML

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

JSON

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

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

Zertifikatbasierte Authentifizierung aktivieren

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

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

Ein typischer Arbeitsablauf umfasst die folgenden Schritte.

Schritte

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Aktualisieren Sie die Authentifizierungsmethoden oder ändern Sie die Anmeldeinformationen.

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

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

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

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

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

Erstellen einer benutzerdefinierten ONTAP Rolle für Trident

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

Siehe"Trident -Benutzerrollengenerator" Weitere Informationen zum Erstellen benutzerdefinierter Trident -Rollen finden Sie hier.

Verwendung der ONTAP Befehlszeile

1. Erstellen Sie eine neue Rolle mit folgendem Befehl:

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

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

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

3. Ordnen Sie die Rolle dem Benutzer zu:

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

Systemmanager verwenden

Führen Sie die folgenden Schritte im ONTAP System Manager aus:

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

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

   b. Wählen Sie das Pfeilsymbol (→) neben **Benutzer und Rollen** aus.
   c. Wählen Sie unter **Rollen** die Option **+Hinzufügen**.
   d. Definieren Sie die Regeln für die Rolle und klicken Sie auf **Speichern**.
2. **Rolle dem Trident -Benutzer zuordnen**: + Führen Sie die folgenden Schritte auf der Seite **Benutzer und Rollen** aus:
   a. Wählen Sie unter **Benutzer** das Symbol **+** zum Hinzufügen aus.
   b. Wählen Sie den gewünschten Benutzernamen und anschließend eine Rolle im Dropdown-Menü für **Rolle** aus.
   c. Klicken Sie auf **Speichern**.

Weitere Informationen finden Sie auf den folgenden Seiten:

- "Benutzerdefinierte Rollen für die Administration von ONTAP"oder"Benutzerdefinierte Rollen definieren"
- "Mit Rollen und Benutzern arbeiten"

Authentifizieren Sie Verbindungen mit bidirektionalem CHAP

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

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

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

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

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

So funktioniert es

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

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

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

Rotieren Sie Anmeldeinformationen und aktualisieren Sie Backends

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

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

```
cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "password",
    "chapInitiatorSecret": "cl9qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

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

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

ONTAP SAN-Konfigurationsoptionen und Beispiele

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

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

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

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

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

Backend-Konfigurationsoptionen

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

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

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

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

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

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

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

Empfehlungen zur Verwendung von formatOptions

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

-E nodiscard:

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

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

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

Schritte

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

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

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

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

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

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

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

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

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

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

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

Beispiele für die Volumenbereitstellung

Hier ist ein Beispiel mit vordefinierten Standardwerten:

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

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

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

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

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

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

Beispiele für minimale Konfigurationen

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

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

ONTAP SAN-Beispiel

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

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

MetroCluster Beispiel

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

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

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

ONTAP SAN Wirtschaftsbeispiel

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

Beispiel für zertifikatsbasierte Authentifizierung

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

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

Beispiele für bidirektionale CHAP-Programme

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

ONTAP SAN CHAP Beispiel

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

ONTAP SAN Wirtschaft CHAP Beispiel

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

NVMe/TCP-Beispiel

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

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

SCSI über FC (FCP) Beispiel

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

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

Backend-Konfigurationsbeispiel mit nameTemplate

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

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

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

Beispiele für Backends mit virtuellen Pools

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

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

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

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 Wirtschaftsbeispiel

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
region: us_east_1
storage:
  - labels:
      app: oracledb
      cost: "30"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
  - labels:
      app: postgresdb
      cost: "20"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
  - labels:
      app: mysqldb
      cost: "10"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
  - labels:
      department: legal
      creditpoints: "5000"
    zone: us_east_1c
```

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

NVMe/TCP-Beispiel

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

Backends StorageClasses zuordnen

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ONTAP NAS-Treiber

ONTAP NAS-Treiberübersicht

Erfahren Sie mehr über die Konfiguration eines ONTAP Backends mit ONTAP und Cloud

Volumes ONTAP NAS-Treibern.

ONTAP NAS-Treiberdetails

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

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

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

Benutzerberechtigungen

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

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

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

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

fehleranfällig macht.

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

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

Anforderungen

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

Authentifizieren Sie das ONTAP Backend

Trident bietet zwei Modi zur Authentifizierung eines ONTAP Backends.

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

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

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

Aktivieren Sie die anmeldeinformationsbasierte Authentifizierung

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

Eine beispielhafte Backend-Definition sieht folgendermaßen aus:

YAML

```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

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

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

Zertifikatsbasierte Authentifizierung aktivieren

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

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

Ein typischer Arbeitsablauf umfasst die folgenden Schritte.

Schritte

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

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

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

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

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

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

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

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

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

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

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

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

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

```
cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "NasBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"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 die Authentifizierungsmethoden oder ändern Sie die Anmeldeinformationen.

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

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

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

```
#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

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

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

Erstellen einer benutzerdefinierten ONTAP Rolle für Trident

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

Siehe"Trident -Benutzerrollengenerator" Weitere Informationen zum Erstellen benutzerdefinierter Trident -Rollen finden Sie hier.

Verwendung der ONTAP Befehlszeile

1. Erstellen Sie eine neue Rolle mit folgendem Befehl:

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

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

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

3. Ordnen Sie die Rolle dem Benutzer zu:

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

Systemmanager verwenden

Führen Sie die folgenden Schritte im ONTAP System Manager aus:

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

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

   b. Wählen Sie das Pfeilsymbol (→) neben **Benutzer und Rollen** aus.
   c. Wählen Sie unter **Rollen** die Option **+Hinzufügen**.
   d. Definieren Sie die Regeln für die Rolle und klicken Sie auf **Speichern**.
2. **Rolle dem Trident -Benutzer zuordnen**: + Führen Sie die folgenden Schritte auf der Seite **Benutzer und Rollen** aus:
   a. Wählen Sie unter **Benutzer** das Symbol **+** zum Hinzufügen aus.
   b. Wählen Sie den gewünschten Benutzernamen und anschließend eine Rolle im Dropdown-Menü für **Rolle** aus.
   c. Klicken Sie auf **Speichern**.

Weitere Informationen finden Sie auf den folgenden Seiten:

- "Benutzerdefinierte Rollen für die Administration von ONTAP"oder"Benutzerdefinierte Rollen definieren"
- "Mit Rollen und Benutzern arbeiten"

NFS-Exportrichtlinien verwalten

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

Trident bietet zwei Optionen für die Arbeit mit Exportrichtlinien:

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

Exportrichtlinien dynamisch verwalten

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

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

Beispiel

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

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

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

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

- `autoExportPolicy`ist eingestellt auf `true`. Dies deutet darauf hin, dass Trident für jedes

mit diesem Backend bereitgestellte Volume eine Exportrichtlinie erstellt. `svm1` SVM und handhaben das Hinzufügen und Löschen von Regeln mithilfe von `autoexportCIDRs` Adressblöcke. Solange ein Volume nicht an einen Knoten angehängt ist, verwendet das Volume eine leere Exportrichtlinie ohne Regeln, um unerwünschten Zugriff auf dieses Volume zu verhindern. Wenn ein Volume auf einem Knoten veröffentlicht wird, erstellt Trident eine Exportrichtlinie mit demselben Namen wie der zugrunde liegende Qtree, der die Knoten-IP innerhalb des angegebenen CIDR-Blocks enthält. Diese IPs werden auch der Exportrichtlinie hinzugefügt, die vom übergeordneten FlexVol volume verwendet wird.

    - Beispiel:
        - Backend-UUID 403b5326-8482-40db-96d0-d83fb3f4daec
        - `autoExportPolicy`eingestellt auf `true`
        - Speicherpräfix `trident`
        - PVC UUID a79bcf5f-7b6d-4a40-9876-e2551f159c1c
        - Ein Qtree mit dem Namen trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c erstellt eine Exportrichtlinie für den FlexVol mit dem Namen `trident-403b5326-8482-40db96d0-d83fb3f4daec` , eine Exportrichtlinie für den Qtree namens `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` und eine leere Exportpolitik namens `trident_empty` auf der SVM. Die Regeln für die FlexVol -Exportrichtlinie bilden eine Obermenge aller in den Qtree-Exportrichtlinien enthaltenen Regeln. Die leere Exportrichtlinie wird von allen Volumes wiederverwendet, die nicht angehängt sind.

- `autoExportCIDRs`enthält eine Liste von Adressblöcken. Dieses Feld ist optional und hat standardmäßig den Wert ["0.0.0.0/0", "::/0"]. Falls nicht definiert, fügt Trident alle global gültigen Unicast-Adressen hinzu, die auf den Worker-Knoten mit Veröffentlichungen gefunden werden.

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

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

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

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

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

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

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

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

Bereiten Sie die Bereitstellung von SMB-Volumes vor

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

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

`autoExportPolicy`wird für SMB-Volumes nicht unterstützt.

Bevor Sie beginnen

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

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

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

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

Schritte

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

   Für Amazon FSx for ONTAP werden SMB-Freigaben benötigt.

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

   a. Erstellen Sie gegebenenfalls die Verzeichnispfadstruktur für die Freigabe.

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

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

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

   c. Überprüfen Sie, ob die Freigabe erstellt wurde:

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

      Siehe"Erstellen einer SMB-Freigabe" Für alle Details.

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

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

Sichere SMB-Verbindungen aktivieren

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

Wichtige Punkte

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

Schritte

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

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

2. Fügen Sie die Annotation in der Speicherklasse hinzu.

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

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

1. Erstellen Sie ein PVC.

   Das folgende Beispiel erzeugt eine PVC:

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

ONTAP NAS-Konfigurationsoptionen und Beispiele

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

Backend-Konfigurationsoptionen

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

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

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

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

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

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

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

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

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

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

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

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

Beispiele für die Volumenbereitstellung

Hier ist ein Beispiel mit vordefinierten Standardwerten:

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

Für `ontap-nas` Und `ontap-nas-flexgroups` Trident verwendet nun eine neue Berechnung, um sicherzustellen, dass FlexVol mit dem SnapshotReserve-Prozentsatz und PVC korrekt dimensioniert wird. Wenn der Benutzer ein PVC anfordert, erstellt Trident das ursprüngliche FlexVol mit mehr Speicherplatz

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

Die Berechnung erfolgt wie folgt:

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

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

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

Beispiele für minimale Konfigurationen

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

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

ONTAP NAS Wirtschaftsbeispiel

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

ONTAP NAS Flexgroup-Beispiel

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

MetroCluster Beispiel

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

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

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

Beispiel für SMB-Volumes

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

Beispiel für zertifikatsbasierte Authentifizierung

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

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

Beispiel für eine automatische Exportrichtlinie

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

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

Beispiel für IPv6-Adressen

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

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

Amazon FSx für ONTAP mit SMB-Volumes – Beispiel

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

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

Backend-Konfigurationsbeispiel mit nameTemplate

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

Beispiele für Backends mit virtuellen Pools

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

Trident legt Bereitstellungsbezeichnungen im Feld „Kommentare“ fest. Kommentare sind auf FlexVol für

`ontap-nas` oder FlexGroup für `ontap-nas-flexgroup` . Trident kopiert bei der Bereitstellung alle im virtuellen Pool vorhandenen Labels auf das Speichervolume. Zur Vereinfachung können Speicheradministratoren Bezeichnungen pro virtuellem Pool definieren und Volumes nach Bezeichnung gruppieren.

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

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: mysqldb
      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 Wirtschaftsbeispiel

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

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

Backends StorageClasses zuordnen

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

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

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

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

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

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

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

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

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

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

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

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

Aktualisieren `dataLIF` nach der ersten Konfiguration

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

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

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

Beispiele für sichere KMU

Backend-Konfiguration mit dem ONTAP-NAS-Treiber

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

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

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

Backend-Konfiguration mit Speicherpool

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

Speicherklassenbeispiel mit dem ONTAP-NAS-Treiber

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

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

Speicherklassenbeispiel mit dem Treiber ontap-nas-economy

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

PVC-Beispiel mit einem einzelnen AD-Benutzer

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

PVC-Beispiel mit mehreren AD-Benutzern

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

Amazon FSx for NetApp ONTAP

Trident mit Amazon FSx for NetApp ONTAP verwenden

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

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

Das Dateisystem ist die primäre Ressource in Amazon FSx, analog zu einem ONTAP Cluster vor Ort.

Innerhalb jeder SVM können Sie ein oder mehrere Volumes erstellen. Dabei handelt es sich um Datencontainer, in denen die Dateien und Ordner Ihres Dateisystems gespeichert werden. Mit Amazon FSx for NetApp ONTAP wird ein verwaltetes Dateisystem in der Cloud bereitgestellt. Der neue Dateisystemtyp heißt * NetApp ONTAP*.

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

Anforderungen

Zusätzlich zu"Trident Anforderungen" Um FSx für ONTAP mit Trident zu integrieren, benötigen Sie:

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

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

Überlegungen

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

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

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

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

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

Authentifizierung

Trident bietet zwei Authentifizierungsmodi an.

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

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

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

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

- "ONTAP NAS-Authentifizierung"
- "ONTAP SAN-Authentifizierung"

Getestete Amazon Machine Images (AMIs)

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

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

BOTTLEROCKET_ARM_64	Ja**	Ja	k. A.	k. A.

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

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

Tests durchgeführt mit:

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

Durchgeführte Tests:

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

Weitere Informationen

- "Amazon FSx for NetApp ONTAP -Dokumentation"
- "Blogbeitrag über Amazon FSx for NetApp ONTAP"

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

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

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

AWS Secrets Manager-Geheimnis erstellen

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

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

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

IAM-Richtlinie erstellen

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

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

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

Beispiel für eine Richtlinien-JSON-Datei:

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

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

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

Pod-Identität

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

Installieren Sie Pod Identity auf Ihrem EKS-Cluster:

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

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

Weitere Informationen finden Sie unter"Amazon EKS Pod Identity Agent einrichten" .

Erstelle trust-relationship.json:

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

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

trust-relationship.json-Datei:

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

Die Rollenrichtlinie der IAM-Rolle zuordnen:

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

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

Eine Pod-Identitätszuordnung erstellen:

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

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

IAM-Rolle für die Dienstkontozuordnung (IRSA)

Verwendung der AWS CLI:

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

trust-relationship.json-Datei:

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

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

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

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

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

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

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

Überprüfen Sie, ob der OICD-Anbieter zugeordnet ist:

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

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

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

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

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

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

Trident installieren

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

Sie können Trident mit einer der folgenden Methoden installieren:

- Helm
- EKS-Add-on

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

Installieren Sie Trident über Helm

Pod-Identität

1. Fügen Sie das Trident Helm-Repository hinzu:

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

2. Installieren Sie Trident anhand des folgenden Beispiels:

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

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

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                           APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2502.0    25.02.0
```

Servicekonto-Zuordnung (IRSA)

1. Fügen Sie das Trident Helm-Repository hinzu:

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

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

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

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

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                          APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2506.0    25.06.0
```

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

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

Installieren Sie Trident über das EKS-Add-on

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

Voraussetzungen

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

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

Aktivieren Sie das Trident Add-on für AWS.

Verwaltungskonsole

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

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

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

```
{

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

}
```

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

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

AWS CLI

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

Für die Pod-Identität verwenden Sie bitte folgendes Format:

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

Für die IRSA-Authentifizierung verwenden Sie bitte folgendes Format:

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

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

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

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

eksctl

Der folgende 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

Verwaltungskonsole

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

AWS CLI

Das folgende Beispiel aktualisiert das EKS-Add-on:

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

eksctl

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

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

  Beispielausgabe:

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

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

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

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

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

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

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

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

Verwaltungskonsole

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

AWS CLI

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

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

eksctl

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

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

Konfigurieren des Speicher-Backends

ONTAP SAN- und NAS-Treiberintegration

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

YAML

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

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

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

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

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

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

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

```
NAME                          BACKEND NAME          BACKEND UUID
PHASE   STATUS

backend-tbc-ontap-nas         tbc-ontap-nas         933e0071-66ce-4324-
b9ff-f96d916ac5e9   Bound   Success
```

FSx für ONTAP Treiberdetails

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

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

Weitere Fahrerdetails finden Sie unter"NAS-Treiber" Und"SAN-Treiber" .

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

```
kubectl create -f configuration_file
```

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

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

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

Erweiterte Backend-Konfiguration und Beispiele

Die folgenden Tabellen enthalten die Backend-Konfigurationsoptionen:

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

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

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

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

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

Backend-Konfigurationsoptionen für die Bereitstellung von Volumes

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

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

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

Bereiten Sie die Bereitstellung von SMB-Volumes vor

Sie können SMB-Volumes mithilfe von … bereitstellen. `ontap-nas` Treiber. Bevor Sie fertigstellenONTAP SAN- und NAS-Treiberintegration Führen Sie die folgenden Schritte aus.

Bevor Sie beginnen

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

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

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

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

Schritte

1. SMB-Freigaben erstellen. Sie können die SMB-Administratorfreigaben auf zwei Arten erstellen, entweder mithilfe von"Microsoft Management Console" Über das Snap-In „Freigegebene Ordner“ oder über die ONTAP -Befehlszeilenschnittstelle. So erstellen Sie die SMB-Freigaben mithilfe der ONTAP -Befehlszeilenschnittstelle:
   a. Erstellen Sie gegebenenfalls die Verzeichnispfadstruktur für die Freigabe.

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

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

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

c. Überprüfen Sie, ob die Freigabe erstellt wurde:

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

Siehe"Erstellen einer SMB-Freigabe" Für alle Details.

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

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

Konfigurieren Sie eine Speicherklasse und einen PVC.

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

Erstellen einer Speicherklasse

Konfigurieren eines Kubernetes StorageClass-Objekts

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

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

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

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

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

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

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

Erstellen einer Speicherklasse

Schritte

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

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

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

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

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

PVC erstellen

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

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

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

Beispielmanifeste

Beispielmanifeste für PersistentVolumeClaim

Diese Beispiele zeigen grundlegende PVC-Konfigurationsoptionen.

PVC mit RWX-Zugang

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

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

PVC mit iSCSI-Beispiel

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

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

PVC erstellen

Schritte

1. Erstellen Sie die PVC.

```
kubectl create -f pvc.yaml
```

2. Überprüfen Sie den PVC-Status.

```
kubectl get pvc
```

```
NAME          STATUS VOLUME     CAPACITY ACCESS MODES STORAGECLASS AGE
pvc-storage Bound  pv-name 2Gi      RWO                           5m
```

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

Trident Eigenschaften

Diese Parameter legen fest, welche von Trident verwalteten Speicherpools zur Bereitstellung von Volumes eines bestimmten Typs verwendet werden sollen.

Attribut	Typ	Werte	Angebot	Anfrage	Unterstützt von
media[1]	Schnur	HDD, Hybrid, SSD	Der Pool enthält Medien dieses Typs; hybrid bedeutet beides	Medientyp angegeben	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
Bereitstellungstyp	Schnur	dünn, dick	Pool unterstützt diese Bereitstellungsmethode	Bereitstellungsmethode angegeben	dick: alles vom Fass; dünn: alles vom Fass & Solidfire-San
Backend-Typ	Schnur	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Pool gehört zu dieser Art von Backend.	Backend spezifiziert	Alle Fahrer
Momentaufnahmen	bool	wahr, falsch	Pool unterstützt Volumes mit Snapshots	Volume mit aktivierten Snapshots	ontap-nas, ontap-san, solidfire-san, gcp-cvs
Klone	bool	wahr, falsch	Pool unterstützt das Klonen von Volumes	Volume mit aktivierten Klonen	ontap-nas, ontap-san, solidfire-san, gcp-cvs

Attribut	Typ	Werte	Angebot	Anfrage	Unterstützt von
Verschlüsselung	bool	wahr, falsch	Pool unterstützt verschlüsselte Volumes	Volume mit aktivierter Verschlüsselung	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	int	positive ganze Zahl	Pool ist in der Lage, IOPS in diesem Bereich zu garantieren.	Volumen garantiert diese IOPS	solidfire-san

[1]: Wird von ONTAP Select -Systemen nicht unterstützt.

Beispielanwendung bereitstellen

Sobald die Speicherklasse und das PVC erstellt sind, können Sie das PV an einem Pod montieren. Dieser Abschnitt listet den Beispielbefehl und die Konfiguration zum Anhängen des PV an einen Pod auf.

Schritte

1. Montieren Sie das Volume in einem Gehäuse.

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

Diese Beispiele zeigen grundlegende Konfigurationen zum Anbringen des PVC an eine Kapsel:
Grundkonfiguration:

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
       claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```

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

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

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

```
Filesystem                                                      Size
Used  Avail Use%  Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06  1.1G
320K  1.0G   1%   /my/mount/path
```

Sie können den Pod jetzt löschen. Die Pod-Anwendung wird nicht mehr existieren, das Volume bleibt jedoch erhalten.

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

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

NetApp Trident optimiert die Amazon FSx for NetApp ONTAP in Kubernetes, damit sich Ihre Entwickler und Administratoren auf die Anwendungsbereitstellung konzentrieren können. Das NetApp Trident EKS-Add-on enthält die neuesten Sicherheitspatches und Fehlerbehebungen und ist von AWS für die Verwendung mit Amazon EKS validiert. Mit dem EKS-Add-on können Sie sicherstellen, dass Ihre Amazon EKS-Cluster stets sicher und stabil sind und den Aufwand für die Installation, Konfiguration und Aktualisierung von Add-ons reduzieren.

Voraussetzungen

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

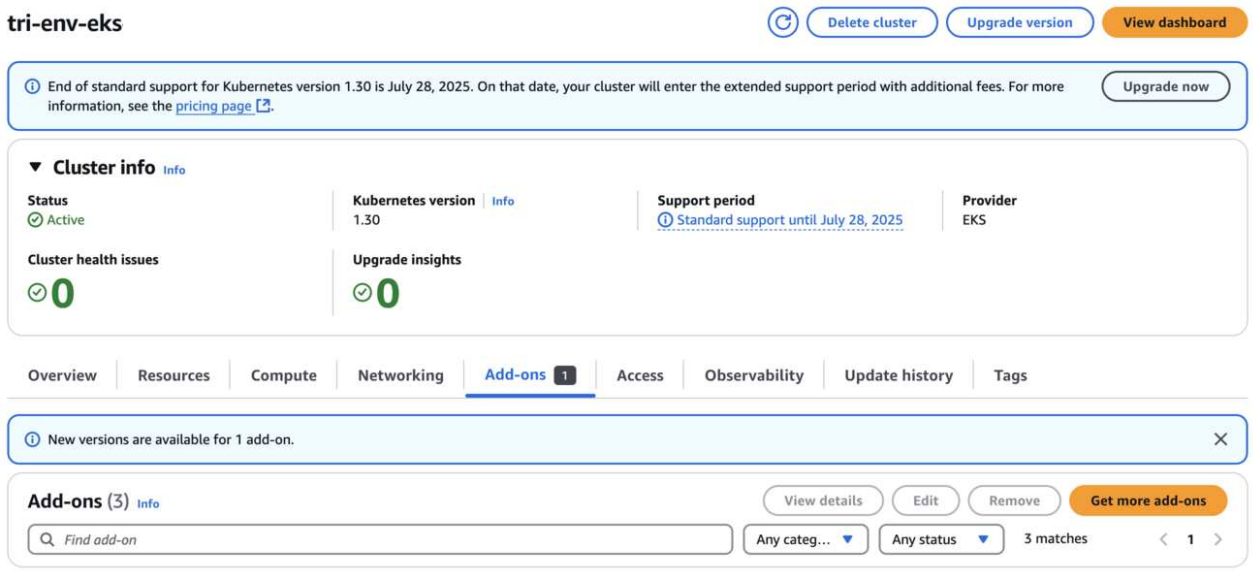
- Ein Amazon EKS-Cluster-Konto mit Berechtigungen zur Arbeit mit Add-ons. Siehe"Amazon EKS-Add-ons" .
- AWS-Berechtigungen für den AWS Marketplace:
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- AMI-Typ: Amazon Linux 2 (AL2_x86_64) oder Amazon Linux 2 Arm (AL2_ARM_64)
- Knotentyp: AMD oder ARM
- Ein bestehendes Amazon FSx for NetApp ONTAP Dateisystem

Schritte

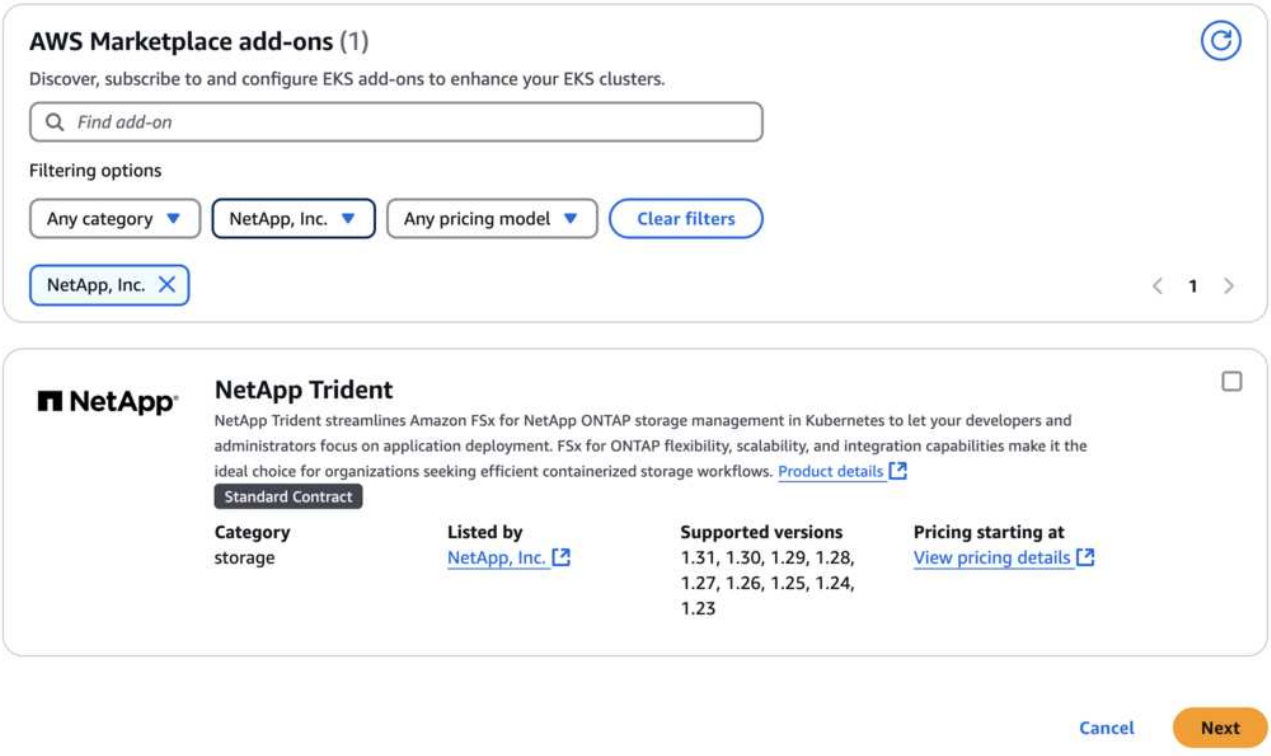
1. Stellen Sie sicher, dass Sie eine IAM-Rolle und ein AWS-Secret erstellen, damit EKS-Pods auf AWS-

Ressourcen zugreifen können. Anweisungen finden Sie unter"Erstellen Sie eine IAM-Rolle und ein AWS-Geheimnis." .

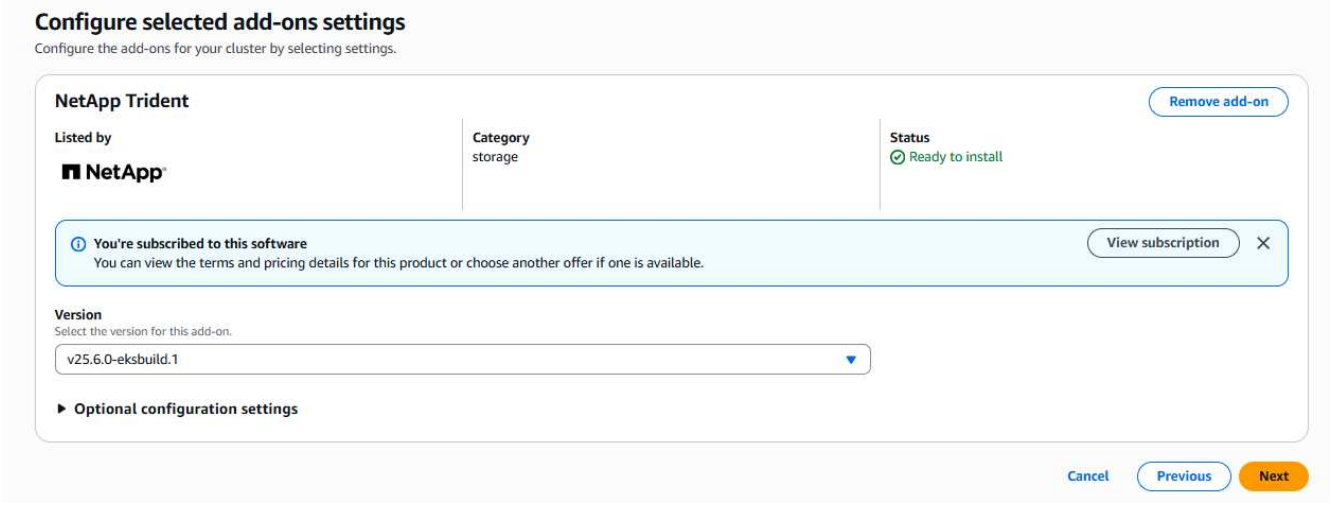
2. Navigieren Sie in Ihrem EKS Kubernetes-Cluster zum Tab **Add-ons**.



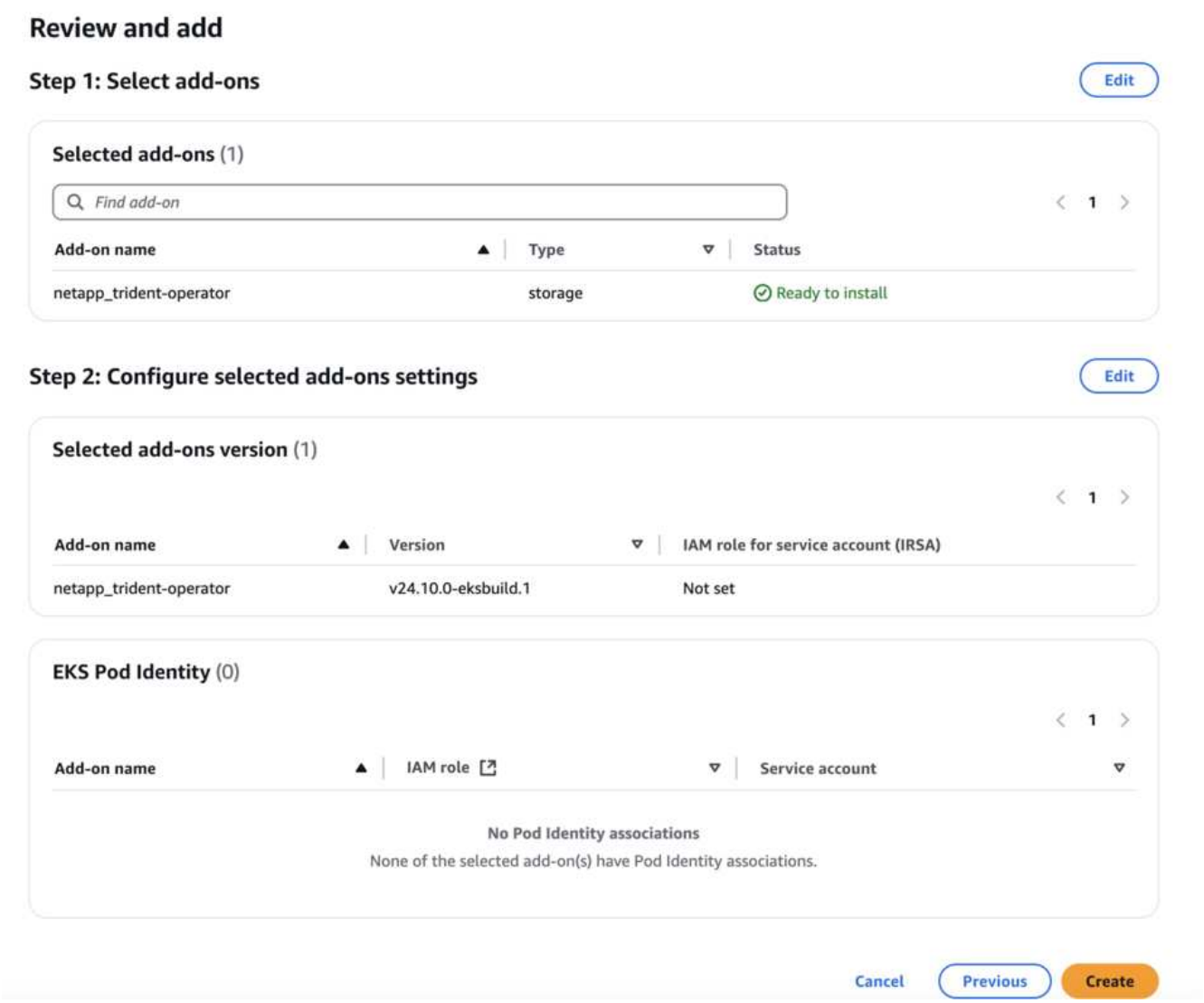
3. Gehen Sie zu **AWS Marketplace Add-ons** und wählen Sie die Kategorie *Speicher* aus.



4. Suchen Sie * NetApp Trident* und aktivieren Sie das Kontrollkästchen für das Trident Add-on. Klicken Sie anschließend auf **Weiter**.
5. Wählen Sie die gewünschte Version des Add-ons.

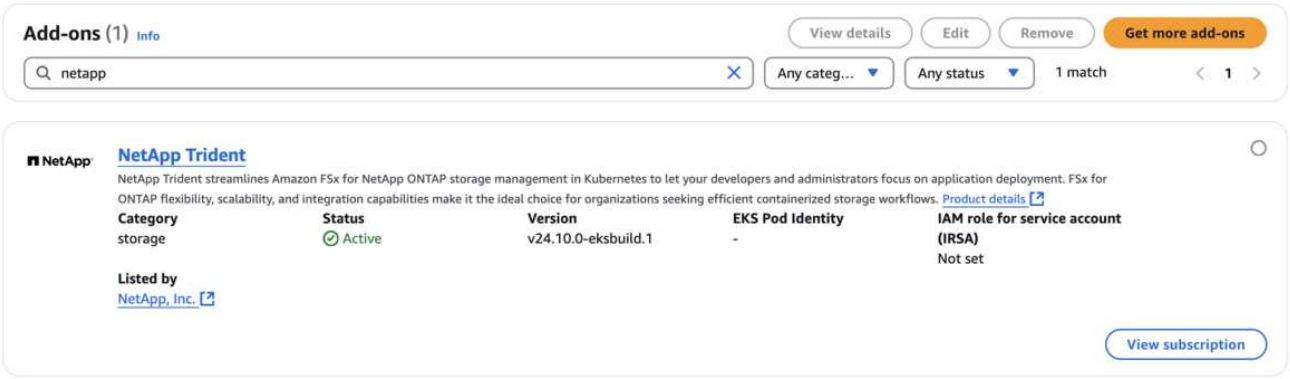


6. Konfigurieren Sie die erforderlichen Add-On-Einstellungen.



7. Wenn Sie IRSA (IAM-Rollen für Dienstkonten) verwenden, beachten Sie die zusätzlichen Konfigurationsschritte."hier," .
8. Wählen Sie **Erstellen**.

9. Überprüfen Sie, ob der Status des Add-ons *Aktiv* lautet.



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

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

11. Setzen Sie die Einrichtung fort und konfigurieren Sie das Speicher-Backend. Weitere Informationen finden Sie unter"Konfigurieren des Speicher-Backends" .

Installation/Deinstallation des Trident EKS-Add-ons über die Befehlszeile

Installieren Sie das NetApp Trident EKS-Add-on über die Befehlszeile:

Der folgende Beispielbefehl installiert das Trident EKS-Add-on: `eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.0-eksbuild.1` (mit einer speziellen Version)

Deinstallieren Sie das NetApp Trident EKS-Add-on über die Befehlszeile:

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

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

Backends mit kubectl erstellen

Ein Backend definiert die Beziehung zwischen Trident und einem Speichersystem. Es teilt Trident mit, wie mit diesem Speichersystem kommuniziert werden soll und wie Trident Datenträger daraus bereitstellen soll. Nach der Installation von Trident besteht der nächste Schritt darin, ein Backend zu erstellen. Der `TridentBackendConfig` Mit Custom Resource Definition (CRD) können Sie Trident -Backends direkt über die Kubernetes-Schnittstelle erstellen und verwalten. Dies können Sie tun, indem Sie `kubectl` oder das entsprechende CLI-Tool für Ihre Kubernetes-Distribution.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig` , `tbackendconfig`) ist ein Frontend mit Namensräumen, das

es Ihnen ermöglicht, Trident -Backends zu verwalten. `kubectl` . Kubernetes- und Speicheradministratoren können Backends jetzt direkt über die Kubernetes-CLI erstellen und verwalten, ohne dass ein separates Befehlszeilenprogramm erforderlich ist.(`tridentctl`).

Bei der Schaffung eines `TridentBackendConfig` Wenn man sich das Objekt ansieht, geschieht Folgendes:

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

Jede `TridentBackendConfig` pflegt eine Eins-zu-Eins-Zuordnung mit einem `TridentBackend` Ersteres ist die dem Benutzer zur Verfügung gestellte Schnittstelle zum Entwerfen und Konfigurieren von Backends; letzteres ist die Art und Weise, wie Trident das eigentliche Backend-Objekt darstellt.

> `TridentBackend`CRs werden von Trident automatisch erstellt. Sie **sollten** sie nicht verändern. Wenn Sie Aktualisierungen an den Backends vornehmen möchten, tun Sie dies durch Ändern der `TridentBackendConfig` Objekt.

Das folgende Beispiel zeigt das Format des `TridentBackendConfig` CR:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Sie können sich auch die Beispiele in der "Dreizack-Installateur" Verzeichnis mit Beispielkonfigurationen für die gewünschte Speicherplattform/den gewünschten Speicherdienst.

Der `spec` benötigt backendspezifische Konfigurationsparameter. In diesem Beispiel verwendet das Backend die `ontap-san` Der Speichertreiber verwendet die hier tabellarisch aufgeführten Konfigurationsparameter. Eine Liste der Konfigurationsoptionen für Ihren gewünschten Speichertreiber finden Sie unter"Backend-Konfigurationsinformationen für Ihren Speichertreiber" .

Der `spec` Dieser Abschnitt umfasst auch `credentials` Und `deletionPolicy` Felder, die neu eingeführt wurden in der `TridentBackendConfig` CR:

- `credentials`Dieser Parameter ist ein Pflichtfeld und enthält die Anmeldeinformationen, die zur Authentifizierung beim Speichersystem/-dienst verwendet werden. Dies ist auf ein vom Benutzer erstelltes

Kubernetes-Secret eingestellt. Die Zugangsdaten dürfen nicht im Klartext übermittelt werden und führen zu einem Fehler.

- `deletionPolicy`Dieses Feld definiert, was geschehen soll, wenn `TridentBackendConfig` wird gelöscht. Es kann einen von zwei möglichen Werten annehmen:
  - `delete`Dies führt zur Löschung beider `TridentBackendConfig` CR und das zugehörige Backend. Dies ist der Standardwert.
  - `retain`Wenn ein `TridentBackendConfig` Wenn CR gelöscht wird, bleibt die Backend-Definition weiterhin vorhanden und kann verwaltet werden mit `tridentctl` . Die Löschrichtlinie festlegen `retain` Ermöglicht es Benutzern, auf eine frühere Version (vor 21.04) zurückzustufen und die erstellten Backends beizubehalten. Der Wert dieses Feldes kann nach einem `TridentBackendConfig` wird erstellt.

Der Name eines Backends wird wie folgt festgelegt: `spec.backendName` . Wenn kein Name angegeben ist, wird der Name des Backends auf den Namen des Backends gesetzt. `TridentBackendConfig` Objekt (metadata.name). Es wird empfohlen, Backend-Namen explizit festzulegen. `spec.backendName` .

Backends, die mit `tridentctl` haben keine zugehörige `TridentBackendConfig` Objekt. Sie können diese Backends mit folgender Funktion verwalten: `kubectl` durch die Erstellung eines `TridentBackendConfig` CR. Es ist darauf zu achten, dass identische Konfigurationsparameter angegeben werden (wie z. B. `spec.backendName` , `spec.storagePrefix` , `spec.storageDriverName` , und so weiter). Trident wird das neu erstellte automatisch binden. `TridentBackendConfig` mit dem bereits bestehenden Backend.

Schrittübersicht

Um ein neues Backend zu erstellen, indem man `kubectl` Sie sollten Folgendes tun:

1. Erstellen Sie ein "Kubernetes-Geheimnis" Das Geheimnis enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster/-dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieser Eintrag enthält spezifische Informationen zum Speichercluster/Speicherdienst und verweist auf das im vorherigen Schritt erstellte Geheimnis.

Nachdem Sie ein Backend erstellt haben, können Sie dessen Status mithilfe von [Name des Dienstes/der Funktion] beobachten. `kubectl get tbc <tbc-name> -n <trident-namespace>` und weitere Details einholen.

Schritt 1: Erstellen Sie ein Kubernetes-Secret.

Erstellen Sie ein Geheimnis, das die Zugangsdaten für das Backend enthält. Dies ist für jeden Speicherdienst bzw. jede Speicherplattform individuell. Hier ist ein Beispiel:

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password
```

Diese Tabelle fasst die Felder zusammen, die für jede Speicherplattform im Secret enthalten sein müssen:

Beschreibung der geheimen Felder der Speicherplattform	Geheimnis	Feldbeschreibung
Azure NetApp Files	Client-ID	Die Client-ID aus einer App-Registrierung
Cloud Volumes Service für GCP	private_key_id	ID des privaten Schlüssels. Teil des API-Schlüssels für ein GCP-Dienstkonto mit CVS-Administratorrolle
Cloud Volumes Service für GCP	privater_key	Privater Schlüssel. Teil des API-Schlüssels für ein GCP-Dienstkonto mit CVS-Administratorrolle
Element (NetApp HCI/ SolidFire)	Endpunkt	MVIP für den SolidFire -Cluster mit Mandantenanmeldeinformationen
ONTAP	Benutzername	Benutzername für die Verbindung mit dem Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet
ONTAP	Passwort	Kennwort für die Verbindung mit dem Cluster/SVM. Wird für die anmeldeinformationsbasierte Authentifizierung verwendet
ONTAP	Client-Privatschlüssel	Base64-kodierter Wert des privaten Client-Schlüssels. Wird für die zertifikatsbasierte Authentifizierung verwendet

Beschreibung der geheimen Felder der Speicherplattform	Geheimnis	Feldbeschreibung
ONTAP	chapUsername	Eingehender Benutzername. Erforderlich, wenn useCHAP=true. Für `ontap-san` Und `ontap-san-economy`
ONTAP	KapitelInitiatorGeheimnis	Geheimnis des CHAP-Initiators. Erforderlich, wenn useCHAP=true. Für `ontap-san` Und `ontap-san-economy`
ONTAP	chapTargetUsername	Zielbenutzername. Erforderlich, wenn useCHAP=true. Für `ontap-san` Und `ontap-san-economy`
ONTAP	Kapitel „Zielinitiatorgeheimnis“	Geheimnis des CHAP-Zielinitiators. Erforderlich, wenn useCHAP=true. Für `ontap-san` Und `ontap-san-economy`

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

Schritt 2: Erstellen Sie die `TridentBackendConfig` CR

Sie können nun Ihre eigene erstellen. `TridentBackendConfig` CR. In diesem Beispiel verwendet ein Backend die `ontap-san` Der Treiber wird mithilfe des `TridentBackendConfig` Das unten abgebildete Objekt:

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Schritt 3: Überprüfen Sie den Status des `TridentBackendConfig` CR

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

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

Es wurde erfolgreich ein Backend erstellt und angebunden. `TridentBackendConfig` CR.

Die Phase kann einen der folgenden Werte annehmen:

- `Bound`: Der `TridentBackendConfig` CR ist mit einem Backend verknüpft, und dieses Backend enthält `configRef` auf die `TridentBackendConfig` CR's uid.
- `Unbound`: Dargestellt durch `""` . Der `TridentBackendConfig` Das Objekt ist nicht an ein Backend gebunden. Alle neu erstellt `TridentBackendConfig` CRs befinden sich standardmäßig in dieser Phase. Nach den Phasenübergängen kann der Zustand „Ungebunden“ nicht mehr rückgängig gemacht werden.
- `Deleting`: Der `TridentBackendConfig` CRs `deletionPolicy` wurde zum Löschen ausgewählt. Wenn die `TridentBackendConfig` Wenn CR gelöscht wird, wechselt es in den Status „Wird gelöscht“.
  - Wenn im Backend keine persistenten Volumenansprüche (PVCs) vorhanden sind, wird deren gelöscht. `TridentBackendConfig` wird dazu führen, dass Trident sowohl das Backend als auch das `TridentBackendConfig` CR.
  - Wenn im Backend ein oder mehrere PVCs vorhanden sind, wechselt das System in den Löschzustand. Der `TridentBackendConfig` Anschließend tritt CR auch in die Löschphase ein. Das Backend und `TridentBackendConfig` werden erst gelöscht, nachdem alle PVCs gelöscht wurden.
- `Lost`: Das zugehörige Backend `TridentBackendConfig` CR wurde versehentlich oder absichtlich gelöscht und die `TridentBackendConfig` CR enthält noch einen Verweis auf das gelöschte Backend.

Der `TridentBackendConfig` CR kann unabhängig davon weiterhin gelöscht werden. `deletionPolicy` Wert.

- `Unknown`Trident kann den Status oder die Existenz des zugehörigen Backends nicht feststellen. `TridentBackendConfig` CR. Wenn beispielsweise der API-Server nicht antwortet oder wenn der `tridentbackends.trident.netapp.io` CRD fehlt. Dies könnte ein Eingreifen erfordern.

In diesem Stadium ist das Backend erfolgreich erstellt worden! Es gibt mehrere weitere Vorgänge, die zusätzlich durchgeführt werden können, wie zum Beispiel"Backend-Aktualisierungen und Backend-Löschungen" .

(Optional) Schritt 4: Weitere Details einholen.

Sie können den folgenden Befehl ausführen, um weitere Informationen über Ihr Backend zu erhalten:

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

```
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8   Bound   Success   ontap-san        delete
```

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

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

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound
```

`backendInfo`enthält `backendName` und die `backendUUID` des Backends, das als Reaktion auf die `TridentBackendConfig` CR. Der `lastOperationStatus` Das Feld repräsentiert den Status der letzten Operation des `TridentBackendConfig` CR, die durch einen Benutzer ausgelöst werden kann (z. B. wenn ein Benutzer etwas geändert hat in `spec` oder ausgelöst durch Trident (zum Beispiel während eines Trident -Neustarts). Es kann entweder ein Erfolg oder ein Misserfolg sein. `phase` stellt den Status der Beziehung zwischen den dar `TridentBackendConfig` CR und das Backend. Im obigen Beispiel `phase` hat den Wert „Gebunden“, was bedeutet, dass `TridentBackendConfig` CR ist mit dem Backend verbunden.

Sie können die `kubectl -n trident describe tbc <tbc-cr-name>` Befehl zum Abrufen von Details zu den Ereignisprotokollen.

Ein Backend, das ein zugehöriges Backend enthält, kann nicht aktualisiert oder gelöscht werden. `TridentBackendConfig` Objekt verwenden `tridentctl` . Um die einzelnen Schritte beim Wechsel zwischen `tridentctl` Und `TridentBackendConfig` ,"siehe hier" .

Backends verwalten

Führen Sie die Backend-Verwaltung mit kubectl durch.

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge durchführen, indem Sie `kubectl` .

Backend löschen

Durch das Löschen eines `TridentBackendConfig` Sie weisen Trident an, Backends zu löschen/beizubehalten (basierend auf `deletionPolicy`). Um ein Backend zu löschen, stellen Sie sicher, dass `deletionPolicy` ist auf Löschen eingestellt. Nur das löschen `TridentBackendConfig` , sicherstellen, dass `deletionPolicy` ist auf Beibehaltung eingestellt. Dadurch wird sichergestellt, dass das Backend weiterhin vorhanden ist und mithilfe von verwaltet werden kann. `tridentctl` .

Führen Sie den folgenden Befehl aus:

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

Trident löscht die Kubernetes-Secrets nicht, die von diesem Unternehmen verwendet wurden. `TridentBackendConfig` . Der Kubernetes-Benutzer ist für das Bereinigen von Geheimnissen verantwortlich. Beim Löschen von Geheimnissen ist Vorsicht geboten. Geheimnisse sollten nur dann gelöscht werden, wenn sie von den Backends nicht verwendet werden.

Die vorhandenen Backends ansehen

Führen Sie den folgenden Befehl aus:

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

Sie können auch ausführen `tridentctl get backend -n trident` oder `tridentctl get backend -o yaml -n trident` um eine Liste aller existierenden Backends zu erhalten. Diese Liste wird auch Backends enthalten, die mit `tridentctl` .

Aktualisieren Sie ein Backend

Es kann mehrere Gründe für ein Backend-Update geben:

- Die Zugangsdaten zum Speichersystem haben sich geändert. Um die Anmeldeinformationen zu aktualisieren, muss das Kubernetes-Secret, das im `TridentBackendConfig` Das Objekt muss aktualisiert werden. Trident aktualisiert das Backend automatisch mit den zuletzt angegebenen Zugangsdaten. Führen Sie folgenden Befehl aus, um das Kubernetes-Secret zu aktualisieren:

  ```
  kubectl apply -f <updated-secret-file.yaml> -n trident
  ```

- Parameter (wie zum Beispiel der Name der verwendeten ONTAP SVM) müssen aktualisiert werden.
  - Sie können aktualisieren `TridentBackendConfig` Objekte direkt über Kubernetes mit folgendem Befehl:

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

- Alternativ können Sie Änderungen an der bestehenden Datei vornehmen. `TridentBackendConfig` CR mit folgendem Befehl:

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

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

Führen Sie die Backend-Verwaltung mit tridentctl durch.

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge durchführen, indem Sie `tridentctl`.

Erstelle ein Backend

Nachdem Sie ein"Backend-Konfigurationsdatei" Führen Sie folgenden Befehl aus:

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

Wenn die Backend-Erstellung fehlschlägt, gab es ein Problem mit der Backend-Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

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

Backend löschen

Um ein Backend aus Trident zu löschen, gehen Sie wie folgt vor:

1. Den Backend-Namen abrufen:

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

2. Backend löschen:

```
tridentctl delete backend <backend-name> -n trident
```

Falls Trident über dieses Backend Volumes und Snapshots bereitgestellt hat, die noch existieren, verhindert das Löschen des Backends, dass neue Volumes über dieses Backend bereitgestellt werden. Das Backend wird weiterhin im Status „Löschen“ verbleiben.

Die vorhandenen Backends ansehen

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

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

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

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

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

Aktualisieren Sie ein Backend

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

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

Wenn das Backend-Update fehlschlägt, lag entweder ein Fehler in der Backend-Konfiguration vor oder Sie haben versucht, ein ungültiges Update durchzuführen. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

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

Identifizieren Sie die Speicherklassen, die ein Backend verwenden.

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

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

Dies gilt auch für Backends, die durch die Verwendung von [fehlende Information] erstellt wurden.

`TridentBackendConfig` .

Wechseln Sie zwischen verschiedenen Backend-Verwaltungsoptionen.

Erfahren Sie mehr über die verschiedenen Möglichkeiten zur Backend-Verwaltung in Trident.

Optionen zur Verwaltung von Backends

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

- Können Backends erstellt werden mit `tridentctl` verwaltet werden mit `TridentBackendConfig` ?
- Können Backends erstellt werden mit `TridentBackendConfig` verwaltet werden mit `tridentctl` ?

Verwalten `tridentctl` Backends verwenden `TridentBackendConfig`

Dieser Abschnitt beschreibt die erforderlichen Schritte zur Verwaltung von Backends, die mit folgendem Werkzeug erstellt wurden: `tridentctl` direkt über die Kubernetes-Schnittstelle durch Erstellen `TridentBackendConfig` Objekte.

Dies gilt für folgende Szenarien:

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

In beiden Szenarien bleiben die Backend-Systeme weiterhin vorhanden, über die Trident die Datenmengen plant und verarbeitet. Administratoren haben hier zwei Möglichkeiten:

- Weiter verwenden `tridentctl` zur Verwaltung von Backends, die damit erstellt wurden.
- Bind-Backends, die mit `tridentctl` zu einem neuen `TridentBackendConfig` Objekt. Dies würde bedeuten, dass die Backends wie folgt verwaltet werden: `kubectl` und nicht `tridentctl` .

Um ein bereits bestehendes Backend zu verwalten `kubectl` Sie müssen einen erstellen `TridentBackendConfig` das an das bestehende Backend angebunden wird. Hier ist eine Übersicht, wie das funktioniert:

1. Erstelle ein Kubernetes-Secret. Das Geheimnis enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster/-dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieser Eintrag enthält spezifische Informationen zum Speichercluster/Speicherdienst und verweist auf das im vorherigen Schritt erstellte Geheimnis. Es ist darauf zu achten, dass identische Konfigurationsparameter angegeben werden (wie z. B. `spec.backendName` , `spec.storagePrefix` , `spec.storageDriverName` , und so weiter). `spec.backendName` muss auf den Namen des bestehenden Backends eingestellt werden.

Schritt 0: Backend identifizieren

Um ein `TridentBackendConfig` Um eine Verbindung zu einem bestehenden Backend herzustellen, müssen Sie die Backend-Konfiguration abrufen. In diesem Beispiel gehen wir davon aus, dass ein Backend mit der folgenden JSON-Definition erstellt wurde:

```
tridentctl get backend ontap-nas-backend -n trident
+---------------------+----------------
+--------------------------------------+--------+---------+
|          NAME       | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+---------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas-backend   | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+---------------------+----------------
+--------------------------------------+--------+---------+
```

```
cat ontap-nas-backend.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}
```

Schritt 1: Erstellen Sie ein Kubernetes-Secret.

Erstellen Sie ein Geheimnis, das die Anmeldeinformationen für das Backend enthält, wie in diesem Beispiel gezeigt:

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

Schritt 2: Erstellen Sie ein `TridentBackendConfig` CR

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

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

In diesem Fall `TridentBackendConfig` wird folgendermaßen aussehen:

```
cat backend-tbc-ontap-nas.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
  region: us_east_1
  storage:
  - labels:
      app: msoffice
      cost: '100'
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: 'true'
      unixPermissions: '0755'
  - labels:
      app: 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 des `TridentBackendConfig` CR

Nach dem `TridentBackendConfig` wurde erstellt, seine Phase muss sein `Bound` . Es sollte außerdem denselben Backend-Namen und dieselbe UUID wie das bestehende Backend aufweisen.

```
kubectl get tbc tbc-ontap-nas-backend -n trident
NAME                   BACKEND NAME          BACKEND UUID
PHASE   STATUS
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
| STATE  | VOLUMES |
+---------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas-backend   | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+---------------------+----------------
+--------------------------------------+--------+---------+
```

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

Verwalten `TridentBackendConfig` Backends verwenden `tridentctl`

```
`tridentctl`kann verwendet werden, um Backends aufzulisten, die mit
folgendem Werkzeug erstellt wurden: `TridentBackendConfig` .  Darüber
hinaus können Administratoren diese Backends auch vollständig selbst
verwalten durch `tridentctl` durch Löschen `TridentBackendConfig` und
sicherstellen `spec.deletionPolicy` ist eingestellt auf `retain` .
```

Schritt 0: Backend identifizieren

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

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        delete

tridentctl get backend ontap-san-backend -n trident
+-------------------+----------------
+--------------------------------------+--------+---------+
|       NAME        | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+-------------------+----------------
+--------------------------------------+--------+---------+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-------------------+----------------
+--------------------------------------+--------+---------+
```

Aus der Ausgabe geht hervor, dass `TridentBackendConfig` wurde erfolgreich erstellt und ist an ein Backend gebunden [siehe die UUID des Backends].

Schritt 1: Bestätigen `deletionPolicy` ist eingestellt auf `retain`

Werfen wir einen Blick auf den Wert von `deletionPolicy` . Dies muss auf eingestellt werden `retain` . Dies stellt sicher, dass, wenn ein `TridentBackendConfig` Wenn CR gelöscht wird, bleibt die Backend-Definition weiterhin vorhanden und kann verwaltet werden mit `tridentctl` .

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        retain
```

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

Schritt 2: Löschen Sie die `TridentBackendConfig` CR

Der letzte Schritt besteht darin, die `TridentBackendConfig` CR. Nach Bestätigung der `deletionPolicy` ist eingestellt auf `retain` Sie können die Löschung nun durchführen:

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+------------------+----------------
+--------------------------------------+--------+---------+
|       NAME        | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------+----------------
+--------------------------------------+--------+---------+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+------------------+----------------
+--------------------------------------+--------+---------+
```

Nach der Löschung der `TridentBackendConfig` Trident entfernt das Objekt einfach, ohne das Backend selbst zu löschen.