Installieren Sie Trident Protect

Trident

NetApp
July 01, 2026

Inhalt

Installieren Sie Trident Protect . . . 1
- Trident Protect Anforderungen . . . 1
  - Trident Protect Kubernetes-Cluster-Kompatibilität . . . 1
  - Trident Protect Speicher-Backend-Kompatibilität . . . 1
  - Anforderungen für nas-economy Volumes . . . 2
  - Schutz von Daten mit KubeVirt VMs . . . 2
  - Anforderungen für die SnapMirror-Replikation . . . 3
- Trident Protect installieren und konfigurieren . . . 5
  - Installieren Sie Trident Protect . . . 5
- Installieren Sie das Trident Protect CLI-Plugin . . . 9
  - Installieren Sie das Trident Protect CLI-Plugin . . . 9
  - Hilfe zum Trident CLI-Plugin anzeigen . . . 11
  - Automatische Befehlsvervollständigung aktivieren . . . 11
- Trident Protect-Installation anpassen . . . 13
  - Geben Sie die Ressourcenbeschränkungen für den Trident Protect-Container an. . . . 13
  - Sicherheitskontextbeschränkungen anpassen . . . 14
  - Konfigurieren Sie zusätzliche Trident Protect Helm-Chart-Einstellungen . . . 15
  - Trident Protect-Pods auf bestimmte Knoten beschränken . . . 17

Installieren Sie Trident Protect

Trident Protect Anforderungen

Beginnen Sie mit der Überprüfung der Einsatzbereitschaft Ihrer Betriebsumgebung, Anwendungscluster, Anwendungen und Lizenzen. Stellen Sie sicher, dass Ihre Umgebung diese Anforderungen erfüllt, um Trident Protect bereitzustellen und zu betreiben.

Trident Protect Kubernetes-Cluster-Kompatibilität

Trident Protect ist mit einer Vielzahl von vollständig verwalteten und selbstverwalteten Kubernetes-Angeboten kompatibel, einschließlich:

- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)
- Microsoft Azure Kubernetes Service (AKS)
- Red Hat OpenShift
- SUSE Harvester 1.7.0 (ONTAP iSCSI)
- SUSE Rancher
- VMware Tanzu Portfolio
- Upstream Kubernetes

- Trident Protect-Backups werden nur auf Linux-Rechenknoten unterstützt. Windows-Rechenknoten werden für Backup-Vorgänge nicht unterstützt.
- Stellen Sie sicher, dass der Cluster, auf dem Sie Trident Protect installieren, mit einem laufenden Snapshot-Controller und den zugehörigen CRDs konfiguriert ist. Informationen zur Installation eines Snapshot-Controllers finden Sie unter "diese Anweisungen".
- Stellen Sie sicher, dass mindestens eine VolumeSnapshotClass existiert. Weitere Informationen finden sich unter "VolumeSnapshotClass".
- Für die Installation von Trident Protect wird Helm 4.x oder höher benötigt.

Trident Protect Speicher-Backend-Kompatibilität

Trident Protect unterstützt die folgenden Storage-Backends:

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- ONTAP Speicherarrays
- Google Cloud NetApp Volumes
- Azure NetApp Files

Stellen Sie sicher, dass Ihr Storage-Backend die folgenden Anforderungen erfüllt:

- Stellen Sie sicher, dass der mit dem Cluster verbundene NetApp Speicher Trident 24.02 oder neuer verwendet (Trident 24.10 wird empfohlen).
- Stellen Sie sicher, dass Sie über ein NetApp ONTAP storage backend verfügen.
- Stellen Sie sicher, dass Sie einen Objektspeicher-Bucket für das Speichern von Backups konfiguriert haben.
- Erstellen Sie alle Anwendungs-Namespaces, die Sie für Anwendungen oder Anwendungsdatenverwaltungsoperationen verwenden möchten. Trident Protect erstellt diese Namespaces nicht für Sie; wenn Sie in einer benutzerdefinierten Ressource einen nicht vorhandenen Namespace angeben, schlägt der Vorgang fehl.

Anforderungen für nas-economy Volumes

Trident Protect unterstützt Backup- und Wiederherstellungsvorgänge auf nas-economy Volumes. Snapshots, Klone und SnapMirror Replikation auf nas-economy Volumes werden derzeit nicht unterstützt. Sie müssen für jedes nas-economy Volume, das Sie mit Trident Protect verwenden möchten, ein Snapshot-Verzeichnis aktivieren.

Einige Anwendungen sind nicht mit Volumes kompatibel, die ein Snapshot-Verzeichnis verwenden. Für diese Anwendungen müssen Sie das Snapshot-Verzeichnis ausblenden, indem Sie den folgenden Befehl auf dem ONTAP Storage-System ausführen:

```
nfs modify -vserver <svm> -v3-hide-snapshot enabled
```

Sie können das Snapshot-Verzeichnis aktivieren, indem Sie für jedes nas-economy-Volume den folgenden Befehl ausführen und dabei `<volume-UUID>` durch die UUID des Volumes ersetzen, das Sie ändern möchten:

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level
=true -n trident
```

Sie können Snapshot-Verzeichnisse standardmäßig für neue Volumes aktivieren, indem Sie die Trident-Backend-Konfigurationsoption `snapshotDir` auf `true` setzen. Vorhandene Volumes sind davon nicht betroffen.

Schutz von Daten mit KubeVirt VMs

Trident Protect bietet Funktionen zum Einfrieren und Auftauen des Dateisystems für KubeVirt virtuelle Maschinen während Datensicherungsoperationen, um die Datenkonsistenz sicherzustellen. Die Konfigurationsmethode und das Standardverhalten für VM-Einfrieroperationen variieren je nach Trident Protect-Version, wobei neuere Versionen eine vereinfachte Konfiguration über Helm-Chart-Parameter bieten.

Während Wiederherstellungsvorgängen werden alle `VirtualMachineSnapshots` für eine virtuelle Maschine (VM) erstellten Dateien nicht wiederhergestellt.

Trident Protect 25.10 und neuer

Trident Protect friert KubeVirt-Dateisysteme während der Datensicherungsoperationen automatisch ein und taut sie wieder auf, um Konsistenz zu gewährleisten. Ab Trident Protect 25.10 können Sie dieses Verhalten mit dem `vm.freeze` Parameter während der Helm-Chart-Installation deaktivieren. Der Parameter ist standardmäßig aktiviert.

```
helm install ... --set vm.freeze=false ...
```

Trident Protect 24.10.1 bis 25.06

Ab Trident Protect 24.10.1 friert Trident Protect KubeVirt-Dateisysteme während Datensicherungsoperationen automatisch ein und taut sie wieder auf. Optional können Sie dieses automatische Verhalten mit dem folgenden Befehl deaktivieren:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=false -n trident-protect
```

Trident Protect 24.10

Trident Protect 24.10 stellt den konsistenten Zustand der KubeVirt VM-Dateisysteme während Datensicherungsoperationen nicht automatisch sicher. Wenn Sie Ihre KubeVirt VM-Daten mit Trident Protect 24.10 schützen möchten, müssen Sie die Funktion zum Einfrieren/Auftauen der Dateisysteme vor der Datensicherungsoperation manuell aktivieren. Dadurch wird sichergestellt, dass sich die Dateisysteme in einem konsistenten Zustand befinden.

Sie können Trident Protect 24.10 so konfigurieren, dass das Einfrieren und Auftauen des VM-Dateisystems während Datensicherungsoperationen verwaltet wird, indem "Virtualisierung konfigurieren" und anschließend den folgenden Befehl verwenden:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=true -n trident-protect
```

Anforderungen für die SnapMirror-Replikation

NetApp SnapMirror Replikation ist für die Verwendung mit Trident Protect für die folgenden ONTAP Lösungen verfügbar:

- Lokale NetApp FAS, AFF und ASA Systeme. SnapMirror-Replikation mit Trident protect wird derzeit für ASA r2 Systeme nicht unterstützt.
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

ONTAP Clusteranforderungen für SnapMirror Replikation

Stellen Sie sicher, dass Ihr ONTAP Cluster die folgenden Anforderungen erfüllt, wenn Sie die SnapMirror Replikation nutzen möchten:

- **NetApp Trident**: NetApp Trident muss sowohl auf dem Quell- als auch auf dem Ziel-Kubernetes-Cluster vorhanden sein, die ONTAP als Backend verwenden. Trident Protect unterstützt die Replikation mit NetApp SnapMirror-Technologie unter Verwendung von Speicherklassen, die von den folgenden Treibern unterstützt werden:
  - `ontap-nas`: NFS
  - `ontap-san`: iSCSI
  - `ontap-san`: FC
  - `ontap-san`: NVMe/TCP (erfordert mindestens ONTAP Version 9.15.1)
- **Lizenzen**: ONTAP SnapMirror asynchrone Lizenzen mit dem Data Protection Bundle müssen sowohl auf dem Quell- als auch auf dem Ziel-ONTAP-Cluster aktiviert sein. Weitere Informationen sind unter "SnapMirror Lizenzierungsübersicht in ONTAP" verfügbar.

  Ab ONTAP 9.10.1 werden alle Lizenzen als NetApp Lizenzdatei (NLF) bereitgestellt, bei der es sich um eine einzelne Datei handelt, die mehrere Funktionen aktiviert. Weitere Informationen sind unter "In ONTAP One enthaltene Lizenzen" verfügbar.

  > Es wird ausschließlich SnapMirror asynchroner Schutz unterstützt.

Peering-Überlegungen für die SnapMirror Replikation

Stellen Sie sicher, dass Ihre Umgebung die folgenden Anforderungen erfüllt, wenn Sie Storage-Backend-Peering verwenden möchten:

- **Cluster und SVM**: Die ONTAP-Speicher-Backends müssen per Peering verbunden sein. Weitere Informationen sind unter "Cluster- und SVM-Peering-Übersicht" verfügbar.

  > Stellen Sie sicher, dass die in der Replikationsbeziehung zwischen zwei ONTAP Clustern verwendeten SVM-Namen eindeutig sind.

- **NetApp Trident und SVM**: Die verbundenen Remote-SVMs müssen für NetApp Trident auf dem Ziel-Cluster verfügbar sein.
- **Verwaltete Backends**: Sie müssen ONTAP-Speicher-Backends in Trident Protect hinzufügen und verwalten, um eine Replikationsbeziehung zu erstellen.

Trident / ONTAP-Konfiguration für SnapMirror-Replikation

Trident Protect erfordert, dass Sie mindestens ein Storage-Backend konfigurieren, das die Replikation sowohl für den Quell- als auch für den Ziel-Cluster unterstützt. Wenn der Quell- und der Ziel-Cluster identisch sind, sollte die Zielanwendung für die beste Ausfallsicherheit ein anderes Storage-Backend als die Quellanwendung verwenden.

Kubernetes-Cluster-Anforderungen für SnapMirror Replikation

Stellen Sie sicher, dass Ihre Kubernetes-Cluster die folgenden Anforderungen erfüllen:

- **AppVault-Zugänglichkeit**: Sowohl Quell- als auch Ziel-Cluster müssen Netzwerkzugriff haben, um vom AppVault zu lesen und darauf zu schreiben, damit Anwendungsobjekte repliziert werden können.
- **Netzwerkanbindung**: Konfigurieren Sie Firewall-Regeln, Bucket-Berechtigungen und IP-Zulassungslisten, um die Kommunikation zwischen beiden Clustern und dem AppVault über WANs hinweg zu ermöglichen.

> Viele Unternehmensumgebungen setzen strenge Firewall-Richtlinien für WAN-Verbindungen ein. Klären Sie diese Netzwerkanforderungen mit Ihrem Infrastrukturteam, bevor Sie die Replikation konfigurieren.

Trident Protect installieren und konfigurieren

Wenn Ihre Umgebung die Anforderungen für Trident Protect erfüllt, können Sie Trident Protect mithilfe der folgenden Schritte auf Ihrem Cluster installieren. Sie können Trident Protect von NetApp beziehen oder es aus Ihrer eigenen privaten Registry installieren. Die Installation aus einer privaten Registry ist hilfreich, wenn Ihr Cluster keinen Internetzugang hat.

Installieren Sie Trident Protect

Installieren Sie Trident Protect von NetApp

Schritte

1. Fügen Sie das Trident Helm repository hinzu:

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

2. Verwenden Sie Helm, um Trident Protect zu installieren. Ersetzen Sie `<name-of-cluster>` durch einen Clusternamen, der dem Cluster zugewiesen und zur Identifizierung der Backups und Snapshots des Clusters verwendet wird:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2602.0 --create
-namespace --namespace trident-protect
```

3. Optional können Sie zur Aktivierung der Debug-Protokollierung (empfohlen zur Fehlerbehebung) Folgendes verwenden:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect
```

   Die Debug-Protokollierung hilft dem NetApp Support, Probleme zu beheben, ohne dass Änderungen des Protokollierungslevels oder eine Reproduktion des Problems erforderlich sind.

Installieren Sie Trident Protect aus einem privaten Registry.

Sie können Trident Protect aus einer privaten Image-Registry installieren, wenn Ihr Kubernetes-Cluster keinen Internetzugang hat. Ersetzen Sie in diesen Beispielen die Werte in Klammern durch Informationen aus Ihrer Umgebung:

Schritte

1. Laden Sie die folgenden Images auf Ihre lokale Maschine herunter, aktualisieren Sie die Tags und laden Sie sie anschließend in Ihre private Registry hoch:

```
docker.io/netapp/controller:26.02.0
docker.io/netapp/restic:26.02.0
docker.io/netapp/kopia:26.02.0
docker.io/netapp/kopiablockrestore:26.02.0
docker.io/netapp/trident-autosupport:26.02.0
docker.io/netapp/exechook:26.02.0
docker.io/netapp/resourcebackup:26.02.0
docker.io/netapp/resourcerestore:26.02.0
docker.io/netapp/resourcedelete:26.02.0
docker.io/netapp/trident-protect-utils:v1.0.0
```

Beispiel:

```
docker pull docker.io/netapp/controller:26.02.0
```

```
docker tag docker.io/netapp/controller:26.02.0 <private-registry-url>/controller:26.02.0
```

```
docker push <private-registry-url>/controller:26.02.0
```

> Um das Helm-Chart zu erhalten, laden Sie zunächst das Helm-Chart auf einem Rechner mit Internetzugang mit `helm pull trident-protect --version 100.2602.0 --repo https://netapp.github.io/trident-protect-helm-chart` herunter, kopieren Sie dann die resultierende `trident-protect-100.2602.0.tgz` Datei in Ihre Offline-Umgebung und installieren Sie es mit `helm install trident-protect ./trident-protect-100.2602.0.tgz` anstelle der Repository-Referenz im letzten Schritt.

2. Erstellen Sie den Trident Protect-Systemnamensraum:

```
kubectl create ns trident-protect
```

3. Melden Sie sich bei der Registry an:

```
helm registry login <private-registry-url> -u <account-id> -p <api-token>
```

4. Erstellen Sie ein Pull-Secret zur Verwendung für die private Registry-Authentifizierung:

```
kubectl create secret docker-registry regcred --docker
-username=<registry-username> --docker-password=<api-token> -n
trident-protect --docker-server=<private-registry-url>
```

5. Fügen Sie das Trident Helm repository hinzu:

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

6. Erstellen Sie eine Datei mit dem Namen `protectValues.yaml`. Stellen Sie sicher, dass sie die folgenden Trident Protect-Einstellungen enthält:

```
---
imageRegistry: <private-registry-url>
imagePullSecrets:
  - name: regcred
```

> Die `imageRegistry` und `imagePullSecrets` Werte gelten für alle Komponentenbilder, einschließlich `resourcebackup` und `resourcerestore`. Wenn Sie Bilder in einen bestimmten Repository-Pfad innerhalb Ihrer Registry hochladen (zum Beispiel `example.com:443/my-repo`), geben Sie den vollständigen Pfad im Registry-Feld an. Dadurch wird sichergestellt, dass alle Bilder aus `<private-registry-url>/<image-name>:<tag>` abgerufen werden.

7. Verwenden Sie Helm, um Trident Protect zu installieren. Ersetzen Sie `<name_of_cluster>` durch einen Clusternamen, der dem Cluster zugewiesen und zur Identifizierung der Backups und Snapshots des Clusters verwendet wird:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2602.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

8. Optional können Sie zur Aktivierung der Debug-Protokollierung (empfohlen zur Fehlerbehebung) Folgendes verwenden:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect -f
protectValues.yaml
```

Die Debug-Protokollierung hilft dem NetApp Support, Probleme zu beheben, ohne dass Änderungen des Protokollierungslevels oder eine Reproduktion des Problems erforderlich sind.

Weitere Helm-Chart-Konfigurationsoptionen, einschließlich AutoSupport-Einstellungen und Namespace-Filterung, finden Sie unter "Trident Protect-Installation anpassen".

Installieren Sie das Trident Protect CLI-Plugin

Sie können das Trident Protect Befehlszeilen-Plugin verwenden, eine Erweiterung des Trident `tridentctl`-Dienstprogramms, um Trident Protect benutzerdefinierte Ressourcen (CRs) zu erstellen und mit ihnen zu interagieren.

Installieren Sie das Trident Protect CLI-Plugin

Bevor Sie das Befehlszeilenprogramm verwenden, müssen Sie es auf dem Rechner installieren, mit dem Sie auf Ihren Cluster zugreifen. Führen Sie die folgenden Schritte aus, je nachdem, ob Ihr Rechner eine x64- oder ARM-CPU verwendet.

Plugin für Linux AMD64-CPUs herunterladen

Schritte

1. Laden Sie das Trident Protect CLI-Plugin herunter:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-linux-amd64
```

Plugin für Linux ARM64 CPUs herunterladen

Schritte

1. Laden Sie das Trident Protect CLI-Plugin herunter:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-linux-arm64
```

Plugin für Mac AMD64-CPUs herunterladen

Schritte

1. Laden Sie das Trident Protect CLI-Plugin herunter:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-macos-amd64
```

Plugin für Mac ARM64 CPUs herunterladen

Schritte

1. Laden Sie das Trident Protect CLI-Plugin herunter:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-macos-arm64
```

1. Aktivieren Sie die Ausführungsberechtigungen für die Plugin-Binärdatei:

```
chmod +x tridentctl-protect
```

2. Kopieren Sie die Plugin-Binärdatei in ein Verzeichnis, das in Ihrer PATH-Variablen definiert ist. Zum Beispiel, `/usr/bin` oder `/usr/local/bin` (möglicherweise benötigen Sie erhöhte Berechtigungen):

```
cp ./tridentctl-protect /usr/local/bin/
```

3. Optional können Sie die Plugin-Binärdatei in ein Verzeichnis in Ihrem Home-Verzeichnis kopieren. In diesem Fall wird empfohlen, sicherzustellen, dass sich das Verzeichnis in Ihrer PATH-Variablen befindet:

```
cp ./tridentctl-protect ~/bin/
```

Durch das Kopieren des Plugins an einen Ort in Ihrer PATH-Variablen können Sie das Plugin verwenden, indem Sie `tridentctl-protect` oder `tridentctl protect` von jedem beliebigen Ort aus eingeben.

Hilfe zum Trident CLI-Plugin anzeigen

Sie können die integrierten Plugin-Hilfefunktionen verwenden, um detaillierte Hilfe zu den Funktionen des Plugins zu erhalten:

Schritte

1. Verwenden Sie die Hilfefunktion, um Nutzungshinweise anzuzeigen:

```
tridentctl-protect help
```

Automatische Befehlsvervollständigung aktivieren

Nachdem Sie das Trident Protect CLI-Plugin installiert haben, können Sie die automatische Vervollständigung für bestimmte Befehle aktivieren.

Aktivieren Sie die automatische Vervollständigung für die Bash-Shell

Schritte

1. Erstellen Sie das Vervollständigungsskript:

```
tridentctl-protect completion bash > tridentctl-completion.bash
```

2. Erstellen Sie ein neues Verzeichnis in Ihrem Home-Verzeichnis, um das Skript darin zu speichern:

```
mkdir -p ~/.bash/completions
```

3. Verschieben Sie das heruntergeladene Skript in das `~/.bash/completions` Verzeichnis:

```
mv tridentctl-completion.bash ~/.bash/completions/
```

4. Fügen Sie die folgende Zeile in die `~/.bashrc` Datei in Ihrem Home-Verzeichnis ein:

```
source ~/.bash/completions/tridentctl-completion.bash
```

Automatische Vervollständigung für die Z shell aktivieren

Schritte

1. Erstellen Sie das Vervollständigungsskript:

```
tridentctl-protect completion zsh > tridentctl-completion.zsh
```

2. Erstellen Sie ein neues Verzeichnis in Ihrem Home-Verzeichnis, um das Skript darin zu speichern:

```
mkdir -p ~/.zsh/completions
```

3. Verschieben Sie das heruntergeladene Skript in das `~/.zsh/completions` Verzeichnis:

```
mv tridentctl-completion.zsh ~/.zsh/completions/
```

4. Fügen Sie die folgende Zeile in die `~/.zprofile` Datei in Ihrem Home-Verzeichnis ein:

```
source ~/.zsh/completions/tridentctl-completion.zsh
```

Ergebnis

Beim nächsten Login in die Shell können Sie die Befehlsvervollständigung mit dem tridentctl-protect plugin nutzen.

Trident Protect-Installation anpassen

Sie können die Standardkonfiguration von Trident Protect an die spezifischen Anforderungen Ihrer Umgebung anpassen.

Geben Sie die Ressourcenbeschränkungen für den Trident Protect-Container an.

Sie können eine Konfigurationsdatei verwenden, um Ressourcenlimits für Trident Protect-Container festzulegen, nachdem Sie Trident Protect installiert haben. Durch das Festlegen von Ressourcenlimits können Sie steuern, wie viele Ressourcen des Clusters von Trident Protect-Operationen verbraucht werden.

Schritte

1. Erstellen Sie eine Datei mit dem Namen `resourceLimits.yaml`.
2. Füllen Sie die Datei mit Ressourcenlimitoptionen für Trident Protect-Container entsprechend den Anforderungen Ihrer Umgebung.

   Die folgende Beispiel-Konfigurationsdatei zeigt die verfügbaren Einstellungen und enthält die Standardwerte für jede Ressourcenbegrenzung:

```
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
  resticVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  resticVolumeRestore:
    limits:
      cpu: ""
      memory: ""
```

```
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
```

3. Wenden Sie die Werte aus der `resourceLimits.yaml` Datei an:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f resourceLimits.yaml --reuse-values
```

Sicherheitskontextbeschränkungen anpassen

Sie können eine Konfigurationsdatei verwenden, um die OpenShift Security Context Constraint (SCCs) für Trident Protect-Container nach der Installation von Trident Protect zu ändern. Diese Beschränkungen definieren Sicherheitsvorgaben für Pods in einem Red Hat OpenShift Cluster.

Schritte

1. Erstellen Sie eine Datei mit dem Namen `sccconfig.yaml`.
2. Fügen Sie die SCC-Option zur Datei hinzu und ändern Sie die Parameter entsprechend den Anforderungen Ihrer Umgebung.

   Das folgende Beispiel zeigt die Standardwerte der Parameter für die SCC-Option:

```
scc:
  create: true
  name: trident-protect-job
  priority: 1
```

Diese Tabelle beschreibt die Parameter für die SCC-Option:

Parameter	Beschreibung	Standard
erstellen	Legt fest, ob eine SCC-Ressource erstellt werden kann. Eine SCC-Ressource wird nur erstellt, wenn `scc.create` auf `true` gesetzt ist und der Helm-Installationsprozess eine OpenShift-Umgebung identifiziert. Wenn nicht auf OpenShift gearbeitet wird oder wenn `scc.create` auf `false` gesetzt ist, wird keine SCC-Ressource erstellt.	true
Name	Gibt den Namen des SCC an.	trident-protect-job
Priorität	Legt die Priorität der SCC fest. SCCs mit höheren Prioritätswerten werden vor solchen mit niedrigeren Werten bewertet.	1

3. Wenden Sie die Werte aus der `sccconfig.yaml` Datei an:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f sccconfig.yaml --reuse-values
```

Dadurch werden die Standardwerte durch die in der `sccconfig.yaml` Datei angegebenen Werte ersetzt.

Konfigurieren Sie zusätzliche Trident Protect Helm-Chart-Einstellungen

Sie können die AutoSupport-Einstellungen und die Namensraumfilterung an Ihre spezifischen Anforderungen anpassen. Die folgende Tabelle beschreibt die verfügbaren Konfigurationsparameter:

Parameter	Typ	Beschreibung
autoSupport.proxy	Zeichenkette	Konfiguriert eine Proxy-URL für NetApp AutoSupport-Verbindungen. Verwenden Sie dies, um Support-Bundle-Uploads über einen Proxyserver zu leiten. Beispiel: `http://my.proxy.url`.

Parameter	Typ	Beschreibung
autoSupport.insecure	boolescher Wert	Überspringt die TLS-Verifizierung für AutoSupport Proxyserver-Verbindungen, wenn auf `true` gesetzt. Nur für unsichere Proxyserver-Verbindungen verwenden. (Standard: `false`)
autoSupport.enabled	boolescher Wert	Aktiviert oder deaktiviert tägliche Trident Protect AutoSupport Bundle-Uploads. Wenn auf `false` gesetzt, werden geplante tägliche Uploads deaktiviert, aber Sie können weiterhin Support-Bundles manuell generieren. (Standard: `true`)
restoreSkipNamespaceAnnotations	Zeichenkette	Kommagetrennte Liste von Namespace-Annotationen, die von Sicherungs- und Wiederherstellungsvorgängen ausgeschlossen werden sollen. Ermöglicht das Filtern von Namespaces anhand von Annotationen.
restoreSkipNamespaceLabels	Zeichenkette	Durch Kommas getrennte Liste von Namespace-Labels, die von Sicherungs- und Wiederherstellungsvorgängen ausgeschlossen werden. Ermöglicht das Filtern von Namespaces anhand von Labels.

Sie können diese Optionen entweder mit einer YAML-Konfigurationsdatei oder Befehlszeilen-Flags konfigurieren:

YAML-Datei verwenden

Schritte

1. Erstelle eine Konfigurationsdatei und benenne sie `values.yaml`.
2. Fügen Sie in der von Ihnen erstellten Datei die Konfigurationsoptionen hinzu, die Sie anpassen möchten.

```
autoSupport:
  enabled: false
  proxy: http://my.proxy.url
  insecure: true
restoreSkipNamespaceAnnotations: "annotation1,annotation2"
restoreSkipNamespaceLabels: "label1,label2"
```

3. Nachdem Sie die `values.yaml` Datei mit den korrekten Werten gefüllt haben, wenden Sie die Konfigurationsdatei an:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f values.yaml --reuse-values
```

CLI-Flag verwenden

Schritte

1. Verwenden Sie den folgenden Befehl mit dem `--set` Flag, um einzelne Parameter anzugeben:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set autoSupport.enabled=false \
  --set autoSupport.proxy=http://my.proxy.url \
  --set-string
restoreSkipNamespaceAnnotations="{annotation1,annotation2}" \
  --set-string restoreSkipNamespaceLabels="{label1,label2}" \
  --reuse-values
```

Trident Protect-Pods auf bestimmte Knoten beschränken

Sie können die Kubernetes nodeSelector Knotenauswahlbeschränkung verwenden, um anhand von Knotenbezeichnungen zu steuern, welche Ihrer Knoten berechtigt sind, Trident Protect-Pods auszuführen. Standardmäßig ist Trident Protect auf Knoten beschränkt, auf denen Linux ausgeführt wird. Sie können diese Beschränkungen je nach Bedarf weiter anpassen.

Schritte

1. Erstellen Sie eine Datei mit dem Namen `nodeSelectorConfig.yaml`.

2. Fügen Sie die nodeSelector-Option zur Datei hinzu und bearbeiten Sie die Datei, um Knotenbezeichnungen hinzuzufügen oder zu ändern, um die Einschränkungen entsprechend den Anforderungen Ihrer Umgebung anzupassen. Die folgende Datei enthält beispielsweise die Standardeinschränkung des Betriebssystems, zielt aber auch auf eine bestimmte Region und einen bestimmten Anwendungsnamen ab:

```
nodeSelector:
  kubernetes.io/os: linux
  region: us-west
  app.kubernetes.io/name: mysql
```

3. Wenden Sie die Werte aus der `nodeSelectorConfig.yaml` Datei an:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f nodeSelectorConfig.yaml --reuse-values
```

Dadurch werden die Standardbeschränkungen durch die von Ihnen in der `nodeSelectorConfig.yaml` Datei angegebenen Beschränkungen ersetzt.