Installieren Sie Trident Protect

Trident

NetApp
January 15, 2026

Inhalt

Installieren Sie Trident Protect

Anforderungen von Trident Protect

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 bereitstellen und betreiben zu können.

Trident Protect Kubernetes-Clusterkompatibilität

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

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

- Trident Protect-Backups werden nur auf Linux-Rechenknoten unterstützt. Windows-Rechenknoten werden für Sicherungsvorgänge 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" .

Kompatibilität des Trident Protect-Speicher-Backends

Trident Protect unterstützt die folgenden Speichersysteme:

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

Stellen Sie sicher, dass Ihr Speichersystem die folgenden Anforderungen erfüllt:

- Stellen Sie sicher, dass auf dem mit dem Cluster verbundenen NetApp Speicher Trident 24.02 oder neuer verwendet wird (Trident 24.10 wird empfohlen).
- Stellen Sie sicher, dass Sie über ein NetApp ONTAP -Speicher-Backend verfügen.
- Stellen Sie sicher, dass Sie einen Objektspeicher-Bucket für die Speicherung von Backups konfiguriert haben.

- Erstellen Sie alle Anwendungs-Namespaces, die Sie für Anwendungen oder Anwendungsdatenverwaltungsvorgänge verwenden möchten. Trident Protect erstellt diese Namespaces nicht für Sie; wenn Sie in einer benutzerdefinierten Ressource einen nicht existierenden Namespace angeben, schlägt der Vorgang fehl.

Anforderungen an die Volumina der Nas-Economy

Trident Protect unterstützt Backup- und Wiederherstellungsvorgänge auf nas-economy-Volumes. Snapshots, Klone und die 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.

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

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

Sie können das Snapshot-Verzeichnis aktivieren, indem Sie den folgenden Befehl für jedes nas-economy-Volume ausführen und dabei Folgendes ersetzen: `<volume-UUID>` mit der UUID des Volumes, 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 entsprechende Option in der Trident Backend-Konfiguration festlegen. `snapshotDir` Zu `true` . Bestehende Volumina bleiben unberührt.

Datenschutz mit KubeVirt-VMs

Trident Protect 24.10 und 24.10.1 sowie neuere Versionen verhalten sich unterschiedlich, wenn Sie Anwendungen schützen, die auf KubeVirt VMs ausgeführt werden. Bei beiden Versionen können Sie das Einfrieren und Auftauen des Dateisystems während Datensicherungsvorgängen aktivieren oder deaktivieren.

Während der Wiederherstellungsvorgänge `VirtualMachineSnapshots` Für eine virtuelle Maschine (VM) erstellte Daten werden nicht wiederhergestellt.

Trident Protect 24.10

Trident Protect 24.10 gewährleistet nicht automatisch einen konsistenten Zustand der KubeVirt VM-Dateisysteme während Datensicherungsvorgängen. Wenn Sie Ihre KubeVirt VM-Daten mit Trident Protect 24.10 schützen möchten, müssen Sie die Freeze/Unfreeze-Funktionalität für die Dateisysteme vor dem Datensicherungsvorgang 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 Datensicherungsvorgängen verwaltet wird, indem"Virtualisierung konfigurieren" und anschließend mit folgendem Befehl:

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

Trident Protect 24.10.1 und neuer

Ab Trident Protect 24.10.1 friert Trident Protect KubeVirt-Dateisysteme während Datensicherungsvorgängen automatisch ein und wieder auf. Optional können Sie dieses automatische Verhalten mit folgendem Befehl deaktivieren:

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

Anforderungen für die SnapMirror Replikation

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

- Lokale NetApp FAS, AFF und ASA Cluster
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

ONTAP Clusteranforderungen für die SnapMirror Replikation

Stellen Sie sicher, dass Ihr ONTAP -Cluster die folgenden Anforderungen erfüllt, wenn Sie die SnapMirror Replikation verwenden 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 der 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 -Asynchronlizenzen, die das Data Protection-Bundle verwenden, müssen sowohl auf dem Quell- als auch auf dem Ziel ONTAP -Cluster aktiviert sein. Siehe "SnapMirror -Lizenzübersicht in ONTAP" für weitere Informationen.

  Ab ONTAP 9.10.1 werden alle Lizenzen als NetApp Lizenzdatei (NLF) ausgeliefert. Dabei handelt es sich um eine einzelne Datei, die mehrere Funktionen aktiviert. Siehe"In ONTAP One enthaltene Lizenzen" für weitere Informationen.

  Es wird ausschließlich der asynchrone Schutz von SnapMirror unterstützt.

Peering-Überlegungen für die SnapMirror Replikation

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

- **Cluster und SVM**: Die ONTAP -Speicher-Backends müssen per Peering verbunden sein. Siehe "Cluster- und SVM-Peering-Übersicht" für weitere Informationen.

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

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

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 Zielcluster unterstützt. Wenn Quell- und Zielcluster identisch sind, sollte die Zielanwendung für eine optimale Ausfallsicherheit ein anderes Speichersystem als die Quellanwendung verwenden.

Anforderungen an einen Kubernetes-Cluster für die SnapMirror Replikation

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

- **AppVault-Zugänglichkeit**: Sowohl der Quell- als auch der Zielcluster müssen über Netzwerkzugriff verfügen, um für die Replikation von Anwendungsobjekten aus dem AppVault lesen und in diesen schreiben zu 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.

  In vielen Unternehmensumgebungen werden strenge Firewall-Richtlinien für WAN-Verbindungen implementiert. Klären Sie diese Netzwerkanforderungen mit Ihrem Infrastrukturteam ab, bevor Sie die Replikation konfigurieren.

Trident Protect installieren und konfigurieren

Wenn Ihre Umgebung die Anforderungen für Trident Protect erfüllt, können Sie die folgenden Schritte befolgen, um Trident Protect auf Ihrem Cluster zu 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 `<name-of-cluster>` mit einem Clusternamen, der dem Cluster zugewiesen wird 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 100.2506.0 --create
-namespace --namespace trident-protect
```

Installieren Sie Trident Protect aus einer privaten Registry.

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

Schritte

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

```
netapp/controller:25.06.0
netapp/restic:25.06.0
netapp/kopia:25.06.0
netapp/trident-autosupport:25.06.0
netapp/exechook:25.06.0
netapp/resourcebackup:25.06.0
netapp/resourcerestore:25.06.0
netapp/resourcedelete:25.06.0
bitnami/kubectl:1.30.2
kubebuilder/kube-rbac-proxy:v0.16.0
```

   Beispiel:

```
docker pull netapp/controller:25.06.0
```

```
docker tag netapp/controller:25.06.0 <private-registry-
url>/controller:25.06.0
```

```
docker push <private-registry-url>/controller:25.06.0
```

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 es die folgenden Trident Protect-Einstellungen enthält:

```
---
image:
  registry: <private-registry-url>
imagePullSecrets:
  - name: regcred
controller:
  image:
    registry: <private-registry-url>
rbacProxy:
  image:
    registry: <private-registry-url>
crCleanup:
  imagePullSecrets:
    - name: regcred
webhooksCleanup:
  imagePullSecrets:
    - name: regcred
```

7. Verwenden Sie Helm, um Trident Protect zu installieren. Ersetzen `<name_of_cluster>` mit einem Clusternamen, der dem Cluster zugewiesen wird 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 100.2506.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

Installieren Sie das Trident Protect CLI-Plugin.

Sie können das Trident Protect-Befehlszeilen-Plugin verwenden, das eine Erweiterung von Trident ist. `tridentctl` Hilfsprogramm zum Erstellen und Interagieren mit benutzerdefinierten Ressourcen (CRs) von Trident Protect.

Installieren Sie das Trident Protect CLI-Plugin.

Bevor Sie das Befehlszeilenprogramm verwenden können, müssen Sie es auf dem Rechner installieren, mit dem Sie auf Ihren Cluster zugreifen. Befolgen Sie diese Schritte, je nachdem, ob Ihr Rechner eine x64- oder eine 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/25.06.0/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/25.06.0/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/25.06.0/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/25.06.0/tridentctl-protect-macos-arm64
```

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

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

2. Kopieren Sie die Plugin-Binärdatei an einen Ort, der 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 der Speicherort in Ihrer PATH-Variablen enthalten ist:

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

Durch Kopieren des Plugins in einen Ordner Ihrer PATH-Variablen können Sie das Plugin durch Eingabe von … verwenden. `tridentctl-protect` oder `tridentctl protect` von jedem beliebigen Ort aus.

Hilfe zum Trident CLI-Plugin anzeigen

Mithilfe der integrierten Plugin-Hilfefunktionen erhalten Sie detaillierte Informationen zu den Funktionen des Plugins:

Schritte

1. Nutzen Sie die Hilfefunktion, um Hinweise zur Verwendung zu erhalten:

```
tridentctl-protect help
```

Automatische Befehlsvervollständigung aktivieren

Nach der Installation des Trident Protect CLI-Plugins können Sie die automatische Vervollständigung für bestimmte Befehle aktivieren.

Aktivieren Sie die automatische Vervollständigung für die Bash-Shell.

Schritte

1. Laden Sie das Vervollständigungsskript herunter:

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.bash
```

2. Erstellen Sie in Ihrem Home-Verzeichnis ein neues Verzeichnis, das das Skript enthalten soll:

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

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

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

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

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

Automatische Vervollständigung für die Z-Shell aktivieren

Schritte

1. Laden Sie das Vervollständigungsskript herunter:

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.zsh
```

2. Erstellen Sie in Ihrem Home-Verzeichnis ein neues Verzeichnis, das das Skript enthalten soll:

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

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

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

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

```
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.

Legen Sie die Ressourcenbeschränkungen für Trident Protect-Container fest.

Sie können nach der Installation von Trident Protect eine Konfigurationsdatei verwenden, um Ressourcenlimits für Trident Protect-Container festzulegen. Durch das Festlegen von Ressourcenlimits können Sie steuern, wie viele der Clusterressourcen von Trident Protect-Operationen verbraucht werden.

Schritte

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

   Die folgende Beispielkonfigurationsdatei zeigt die verfügbaren Einstellungen und enthält die Standardwerte für jedes Ressourcenlimit:

```
---
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 dem `resourceLimits.yaml` Datei:

```
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-Sicherheitskontextbeschränkungen (SCCs) für Trident Protect-Container nach der Installation von Trident Protect zu ändern. Diese Einschränkungen definieren Sicherheitsbeschränkungen 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 passen Sie die Parameter an die Bedürfnisse Ihrer Umgebung an.

   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	Ermittelt, ob eine SCC-Ressource erstellt werden kann. Eine SCC-Ressource wird nur dann erstellt, wenn `scc.create` ist eingestellt auf `true` und der Helm-Installationsprozess erkennt eine OpenShift-Umgebung. Wenn der Betrieb nicht auf OpenShift erfolgt, oder wenn `scc.create` ist eingestellt auf `false` Es wird keine SCC-Ressource erstellt.	true
Name	Gibt den Namen des SCC an.	trident-protect-job
Priorität	Definiert die Priorität des SCC. SCCs mit höheren Prioritätswerten werden vor solchen mit niedrigeren Werten bewertet.	1

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

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

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

Konfigurieren Sie zusätzliche Trident Protect Helm-Chart-Einstellungen

Sie können die AutoSupport Einstellungen und die Namespace-Filterung an Ihre spezifischen Anforderungen anpassen. In der folgenden Tabelle werden die verfügbaren Konfigurationsparameter beschrieben:

Parameter	Typ	Beschreibung
autoSupport.proxy	Schnur	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.unsicher	boolescher Wert	Überspringt die TLS-Verifizierung für AutoSupport Proxy-Verbindungen, wenn diese Option aktiviert ist. `true` . Nur für unsichere Proxy-Verbindungen verwenden. (Standard: `false`)
autoSupport.enabled	boolescher Wert	Aktiviert oder deaktiviert tägliche Trident Protect AutoSupport Bundle-Uploads. Wenn eingestellt auf `false` Tägliche Uploads sind deaktiviert, Support-Bundles können aber weiterhin manuell generiert werden. (Standard: `true`)
restoreSkipNamespaceAnnotations	Schnur	Durch Kommas getrennte Liste von Namespace-Anmerkungen, die von Sicherungs- und Wiederherstellungsvorgängen ausgeschlossen werden sollen. Ermöglicht Ihnen, Namespaces basierend auf Anmerkungen zu filtern.
restoreSkipNamespaceLabels	Schnur	Durch Kommas getrennte Liste von Namespace-Bezeichnungen, die von Sicherungs- und Wiederherstellungsvorgängen ausgeschlossen werden sollen. Ermöglicht Ihnen, Namespaces basierend auf Labels zu filtern.

Sie können diese Optionen entweder mithilfe einer YAML-Konfigurationsdatei oder mithilfe von Befehlszeilenflags konfigurieren:

YAML-Datei verwenden

Schritte

1. Erstellen Sie eine Konfigurationsdatei und benennen Sie 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, Konfigurationsdatei anwenden:

```
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` Flagge zur Angabe einzelner Parameter:

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

Trident Protect-Pods auf bestimmte Knoten beschränken

Mithilfe der Kubernetes-Knotenauswahlbeschränkung nodeSelector können Sie anhand der Knotenbezeichnungen steuern, welche Ihrer Knoten für die Ausführung von Trident Protect-Pods geeignet sind. Standardmäßig ist Trident Protect auf Knoten beschränkt, auf denen Linux ausgeführt wird. Sie können diese Einschränkungen je nach Ihren Bedürfnissen weiter anpassen.

Schritte

1. Erstellen Sie eine Datei mit dem Namen `nodeSelectorConfig.yaml` .
2. Fügen Sie die Option nodeSelector zur Datei hinzu und modifizieren Sie die Datei, um Knotenbezeichnungen hinzuzufügen oder zu ändern und diese entsprechend den Anforderungen Ihrer

Umgebung einzuschränken. Die folgende Datei enthält beispielsweise die Standardbeschränkung 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 dem `nodeSelectorConfig.yaml` Datei:

```
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 angegebenen Beschränkungen ersetzt. `nodeSelectorConfig.yaml` Datei.