Installare Trident Protect

Trident

NetApp
November 14, 2025

Sommario

Installare Trident Protect

Requisiti di Trident Protect

Inizia subito con la verifica della prontezza del tuo ambiente operativo, dei cluster di applicazioni, delle applicazioni e delle licenze. Assicurati che il tuo ambiente soddisfi questi requisiti per l'implementazione e l'utilizzo di Trident Protect.

Trident protegge la compatibilità del cluster Kubernetes

Trident Protect è compatibile con un'ampia gamma di offerte Kubernetes completamente gestite e autogestite, tra cui:

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

- I backup Trident Protect sono supportati solo sui nodi di elaborazione Linux. I nodi di elaborazione Windows non sono supportati per le operazioni di backup.
- Assicurarsi che il cluster su cui si installa Trident Protect sia configurato con un controller snapshot in esecuzione e i CRD correlati. Per installare un'unità di controllo istantanee, fare riferimento alla "queste istruzioni".

Trident protegge la compatibilità del backend di storage

Trident Protect supporta i seguenti backend di storage:

- Amazon FSX per NetApp ONTAP
- Cloud Volumes ONTAP
- Array storage ONTAP
- Google Cloud NetApp Volumes
- Azure NetApp Files

Verificare che lo storage backend soddisfi i seguenti requisiti:

- Assicurarsi che lo storage NetApp connesso al cluster utilizzi Trident 24.02 o una versione successiva (si consiglia Trident 24.10).
- Verificare di disporre di un back-end dello storage NetApp ONTAP.
- Verificare di aver configurato un bucket dello storage a oggetti per la memorizzazione dei backup.
- Creare spazi dei nomi delle applicazioni che si intende utilizzare per applicazioni o operazioni di gestione dei dati delle applicazioni. Trident Protect non crea questi spazi dei nomi per l'utente; se si specifica uno

spazio dei nomi inesistente in una risorsa personalizzata, l'operazione non verrà eseguita correttamente.

Per i volumi nas-Economy

Trident Protect supporta le operazioni di backup e ripristino su volumi nas-Economy. Al momento snapshot, cloni e replica SnapMirror sui volumi nas-Economy non sono supportati. È necessario abilitare una directory di snapshot per ogni volume economico nas che si intende utilizzare con Trident Protect.

Alcune applicazioni non sono compatibili con volumi che utilizzano una directory snapshot. Per queste applicazioni, è necessario nascondere la directory dello snapshot eseguendo il seguente comando nel sistema di archiviazione ONTAP:

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

Puoi abilitare la directory dello snapshot eseguendo il seguente comando per ogni volume di economia nas, sostituendo `<volume-UUID>` con l'UUID del volume che desideri modificare:

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

Per impostazione predefinita, è possibile abilitare le directory snapshot per i nuovi volumi impostando l'opzione di configurazione back-end Trident `snapshotDir` su `true`. I volumi esistenti non vengono influenzati.

Protezione dei dati con le macchine virtuali KubeVirt

Trident Protect fornisce funzionalità di blocco e sblocco del file system per le macchine virtuali KubeVirt durante le operazioni di protezione dei dati per garantire la coerenza dei dati. Il metodo di configurazione e il comportamento predefinito per le operazioni di congelamento delle VM variano a seconda delle versioni Trident Protect, con le versioni più recenti che offrono una configurazione semplificata tramite i parametri del grafico Helm.

Durante le operazioni di ripristino, qualsiasi `VirtualMachineSnapshots` creati per una macchina virtuale (VM) non vengono ripristinati.

Trident Protect 25.10 e versioni successive

Trident Protect blocca e sblocca automaticamente i file system KubeVirt durante le operazioni di protezione dei dati per garantire la coerenza. A partire da Trident Protect 25.10, è possibile disattivare questo comportamento utilizzando `vm.freeze` parametro durante l'installazione della carta Helm. Il parametro è abilitato per impostazione predefinita.

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

Trident Protect dal 24.10.1 al 25.06

A partire da Trident Protect 24.10.1, Trident Protect blocca e sblocca automaticamente i file system KubeVirt durante le operazioni di data Protection. Facoltativamente, è possibile disattivare questo comportamento automatico utilizzando il seguente comando:

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

Trident Protect 24,10

Trident Protect 24,10 non garantisce automaticamente uno stato coerente dei file system delle macchine virtuali KubeVirt durante le operazioni di protezione dei dati. Per proteggere i dati delle macchine virtuali KubeVirt utilizzando Trident Protect 24,10, è necessario abilitare manualmente la funzionalità di blocco/sblocco dei file system prima dell'operazione di protezione dei dati. Ciò garantisce che i filesystem siano in uno stato coerente.

È possibile configurare Trident Protect 24,10 per gestire il blocco e lo sblocco del file system della VM durante le operazioni di protezione dei dati "configurazione della virtualizzazione"utilizzando il seguente comando:

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

Requisiti per la replica SnapMirror

La replica di NetApp SnapMirror è disponibile per l'utilizzo con Trident Protect per le seguenti soluzioni ONTAP:

- Cluster NetApp FAS, AFF e ASA on-premise
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSX per NetApp ONTAP

Requisiti del cluster di ONTAP per la replica SnapMirror

Assicurati che il tuo cluster ONTAP soddisfi i seguenti requisiti se intendi utilizzare la replica SnapMirror:

- **NetApp Trident**: NetApp Trident deve essere presente sia sul cluster Kubernetes di origine che su quello di destinazione che utilizzano ONTAP come backend. Trident Protect supporta la replica con la tecnologia NetApp SnapMirror utilizzando classi di storage supportate dai seguenti driver:
  - `ontap-nas` : NFS
  - `ontap-san` : iSCSI
  - `ontap-san` : FC

- `ontap-san` : NVMe/TCP (richiede almeno la versione ONTAP 9.15.1)

- **Licenze**: Le licenze asincrone di ONTAP SnapMirror che utilizzano il bundle di protezione dati devono essere attivate sia sul cluster ONTAP di origine che su quello di destinazione. Per ulteriori informazioni, fare riferimento "Panoramica sulle licenze SnapMirror in ONTAP" a.

  A partire da ONTAP 9.10.1, tutte le licenze vengono fornite come file di licenza NetApp (NLF), che è un singolo file che abilita più funzioni. Per ulteriori informazioni, fare riferimento "Licenze incluse con ONTAP ONE" a.

  È supportata solo la protezione asincrona SnapMirror.

Considerazioni sul peering per la replica SnapMirror

Assicurati che il tuo ambiente soddisfi i seguenti requisiti se intendi utilizzare il peering di back-end dello storage:

- **Cluster e SVM**: I backend dello storage ONTAP devono essere peering. Per ulteriori informazioni, fare riferimento "Panoramica del peering di cluster e SVM" a.

  Assicurati che i nomi delle SVM utilizzati nella relazione di replica tra due cluster ONTAP siano univoci.

- **NetApp Trident e SVM**: le SVM remote peered devono essere disponibili per NetApp Trident sul cluster di destinazione.
- **Backend gestiti**: È necessario aggiungere e gestire i backend di storage ONTAP in Trident Protect per creare una relazione di replica.

Configurazione Trident / ONTAP per la replica SnapMirror

Trident Protect richiede la configurazione di almeno un backend di storage che supporti la replica per i cluster di origine e di destinazione. Se i cluster di origine e di destinazione sono gli stessi, l'applicazione di destinazione deve utilizzare un backend di storage diverso da quello dell'applicazione di origine per ottenere la migliore resilienza.

Requisiti del cluster Kubernetes per la replica SnapMirror

Assicurati che i tuoi cluster Kubernetes soddisfino i seguenti requisiti:

- **Accessibilità ad AppVault**: sia i cluster di origine che quelli di destinazione devono avere accesso alla rete per leggere e scrivere su AppVault per la replica degli oggetti applicativi.
- **Connettività di rete**: configura le regole del firewall, le autorizzazioni dei bucket e le liste consentite di IP per abilitare la comunicazione tra entrambi i cluster e AppVault attraverso le WAN.

Molti ambienti aziendali implementano rigide policy firewall sulle connessioni WAN. Verificare questi requisiti di rete con il team dell'infrastruttura prima di configurare la replica.

Installare e configurare Trident Protect

Se l'ambiente in uso soddisfa i requisiti di Trident Protect, è possibile seguire questa procedura per installare Trident Protect sul cluster. È possibile ottenere Trident Protect da

NetApp o installarlo dal proprio registro privato. L'installazione da un registro privato è utile se il cluster non riesce ad accedere a Internet.

Installare Trident Protect

Installare Trident Protect di NetApp

Fasi

1. Aggiungere il repository Trident Helm:

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

2. Utilizzare Helm per installare Trident Protect. Sostituire `<name-of-cluster>` con un nome cluster, che verrà assegnato al cluster e utilizzato per identificare i backup e gli snapshot del cluster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2510.0 --create
-namespace --namespace trident-protect
```

3. Facoltativamente, per abilitare la registrazione del debug (consigliata per la risoluzione dei problemi), utilizzare:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2510.0 --create-namespace --namespace trident-protect
```

La registrazione del debug aiuta NetApp a risolvere i problemi senza dover modificare il livello di registrazione o riprodurre i problemi.

Installare Trident Protect da un registro privato

È possibile installare Trident Protect da un registro di immagine privata se il cluster Kubernetes non è in grado di accedere a Internet. In questi esempi, sostituire i valori tra parentesi con le informazioni dell'ambiente:

Fasi

1. Estrarre le seguenti immagini sul computer locale, aggiornare i tag e quindi inviarle al registro privato:

```
docker.io/netapp/controller:25.10.0
docker.io/netapp/restic:25.10.0
docker.io/netapp/kopia:25.10.0
docker.io/netapp/kopiablockrestore:25.10.0
docker.io/netapp/trident-autosupport:25.10.0
docker.io/netapp/exechook:25.10.0
docker.io/netapp/resourcebackup:25.10.0
docker.io/netapp/resourcerestore:25.10.0
docker.io/netapp/resourcedelete:25.10.0
docker.io/netapp/trident-protect-utils:v1.0.0
```

Ad esempio:

```
docker pull docker.io/netapp/controller:25.10.0
```

```
docker tag docker.io/netapp/controller:25.10.0 <private-registry-url>/controller:25.10.0
```

```
docker push <private-registry-url>/controller:25.10.0
```

> Per ottenere la tabella Helm, scaricare prima la tabella Helm su un computer con accesso a Internet utilizzando `helm pull trident-protect --version 100.2510.0 --repo https://netapp.github.io/trident-protect-helm-chart`, quindi copia il risultato `trident-protect-100.2510.0.tgz` file nel tuo ambiente offline e installalo utilizzando `helm install trident-protect ./trident-protect-100.2510.0.tgz` invece del riferimento al repository nel passaggio finale.

2. Creare lo spazio dei nomi del sistema Trident Protect:

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

3. Accedere al Registro di sistema:

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

4. Creare un segreto pull da utilizzare per l'autenticazione privata del Registro di sistema:

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

5. Aggiungere il repository Trident Helm:

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

6. Creare un file denominato `protectValues.yaml`. Verificare che contenga le seguenti impostazioni di protezione Trident:

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

IL `imageRegistry` E `imagePullSecrets` i valori si applicano a tutte le immagini dei componenti, comprese `resourcebackup` E `resourcerestore` . Se si inseriscono immagini in un percorso di repository specifico all'interno del registro (ad esempio, `example.com:443/my-repo`), includere il percorso completo nel campo del registro. Ciò garantirà che tutte le immagini vengano estratte da `<private-registry-url>/<image-name>:<tag>`.

7. Utilizzare Helm per installare Trident Protect. Sostituire `<name_of_cluster>` con un nome cluster, che verrà assegnato al cluster e utilizzato per identificare i backup e gli snapshot del cluster:

```
helm install trident-protect netapp-trident-protect/trident-protect --set clusterName=<name_of_cluster> --version 100.2510.0 --create-namespace --namespace trident-protect -f protectValues.yaml
```

8. Facoltativamente, per abilitare la registrazione del debug (consigliata per la risoluzione dei problemi), utilizzare:

```
helm install trident-protect netapp-trident-protect/trident-protect --set clusterName=<name-of-cluster> --set logLevel=debug --version 100.2510.0 --create-namespace --namespace trident-protect -f protectValues.yaml
```

La registrazione del debug aiuta NetApp a risolvere i problemi senza dover modificare il livello di registrazione o riprodurre i problemi.

Per ulteriori opzioni di configurazione del grafico Helm, incluse le impostazioni AutoSupport e il filtraggio dello spazio dei nomi, fare riferimento a "Personalizzare l'installazione di Trident Protect".

Installare il plugin Trident Protect CLI

È possibile utilizzare il plug-in della riga di comando Trident Protect, che è un'estensione dell'utilità Trident `tridentctl`, per creare e interagire con le risorse personalizzate Trident Protect (CRS).

Installare il plugin Trident Protect CLI

Prima di utilizzare l'utilità della riga di comando, è necessario installarla sulla macchina utilizzata per accedere

al cluster. Attenersi alla seguente procedura, a seconda che il computer utilizzi una CPU x64 o ARM.

Scarica il plugin per CPU Linux AMD64

Fasi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.10.0/tridentctl-protect-linux-amd64
```

Scarica il plugin per CPU Linux ARM64

Fasi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.10.0/tridentctl-protect-linux-arm64
```

Scarica il plugin per le CPU Mac AMD64

Fasi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.10.0/tridentctl-protect-macos-amd64
```

Scarica il plugin per le CPU Mac ARM64

Fasi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.10.0/tridentctl-protect-macos-arm64
```

1. Abilitare le autorizzazioni di esecuzione per il binario del plugin:

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

2. Copiare il file binario del plugin in una posizione definita nella variabile PATH. Ad esempio, `/usr/bin` o `/usr/local/bin` (potrebbe essere necessario un Privileges elevato):

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

3. Facoltativamente, è possibile copiare il file binario del plugin in una posizione nella propria home directory. In questo caso, si consiglia di assicurarsi che la posizione faccia parte della variabile PATH:

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

Copiare il plugin in una posizione nella variabile PATH consente di utilizzare il plugin digitando `tridentctl-protect` o `tridentctl protect` da qualsiasi posizione.

Visualizza la guida del plugin CLI di Trident

È possibile utilizzare le funzioni della guida del plugin incorporato per ottenere una guida dettagliata sulle funzionalità del plugin:

Fasi

1. Utilizzare la funzione di guida per visualizzare le indicazioni sull'utilizzo:

```
tridentctl-protect help
```

Attivare il completamento automatico del comando

Dopo aver installato il plugin Trident Protect CLI, è possibile abilitare il completamento automatico per alcuni comandi.

Attivare il completamento automatico per la shell Bash

Fasi

1. Creare lo script di completamento:

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

2. Creare una nuova directory nella home directory in modo che contenga lo script:

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

3. Spostare lo script scaricato nella `~/.bash/completions` directory:

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

4. Aggiungere la seguente riga al `~/.bashrc` file nella propria home directory:

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

Attivare il completamento automatico per la shell Z

Fasi

1. Creare lo script di completamento:

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

2. Creare una nuova directory nella home directory in modo che contenga lo script:

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

3. Spostare lo script scaricato nella `~/.zsh/completions` directory:

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

4. Aggiungere la seguente riga al `~/.zprofile` file nella propria home directory:

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

Risultato

Al prossimo login della shell, potete usare il comando auto-completation con il plugin tridentctl-Protect.

Personalizzare l'installazione di Trident Protect

È possibile personalizzare la configurazione predefinita di Trident Protect per soddisfare i requisiti specifici dell'ambiente.

Specificare i limiti delle risorse del contenitore Trident Protect

È possibile utilizzare un file di configurazione per specificare i limiti delle risorse per i contenitori Trident Protect dopo l'installazione di Trident Protect. L'impostazione di limiti delle risorse consente di controllare la quantità di risorse del cluster utilizzata dalle operazioni Trident Protect.

Fasi

1. Creare un file denominato `resourceLimits.yaml`.
2. Popolare il file con opzioni di limite delle risorse per i contenitori Trident Protect in base alle esigenze dell'ambiente.

   Il seguente file di configurazione di esempio mostra le impostazioni disponibili e contiene i valori predefiniti per ogni limite di risorse:

```
---
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. Applicare i valori dal `resourceLimits.yaml` file:

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

Personalizzare i vincoli del contesto di protezione

È possibile utilizzare un file di configurazione per modificare il vincolo del contesto di protezione OpenShift per i contenitori Trident Protect dopo l'installazione di Trident Protect. Questi vincoli definiscono le restrizioni di sicurezza per i pod in un cluster Red Hat OpenShift.

Fasi

1. Creare un file denominato `sccconfig.yaml`.
2. Aggiungere l'opzione SCC al file e modificare i parametri in base alle esigenze dell'ambiente.

   Nell'esempio seguente vengono mostrati i valori predefiniti dei parametri per l'opzione SCC:

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

Questa tabella descrive i parametri per l'opzione SCC:

Parametro	Descrizione	Predefinito
creare	Determina se è possibile creare una risorsa SCC. Una risorsa SCC verrà creata solo se `scc.create` è impostato su `true` e il processo di installazione di Helm identifica un ambiente OpenShift. Se non funziona su OpenShift, o se `scc.create` è impostato su `false`, non verrà creata alcuna risorsa SCC.	vero
nome	Specifica il nome della SCC.	processo-di-protezione-Trident
priorità	Definisce la priorità dell'SCC. Gli scc con valori di priorità più elevati vengono valutati prima di quelli con valori più bassi.	1

3. Applicare i valori dal `sccconfig.yaml` file:

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

In questo modo i valori predefiniti verranno sostituiti con quelli specificati nel `sccconfig.yaml` file.

Configurare le impostazioni aggiuntive del grafico del timone di protezione Trident

È possibile personalizzare le impostazioni AutoSupport e il filtraggio degli spazi dei nomi in base alle proprie esigenze specifiche. La tabella seguente descrive i parametri di configurazione disponibili:

Parametro	Tipo	Descrizione
autoSupport.proxy	stringa	Configura un URL proxy per le connessioni NetApp AutoSupport . Utilizzare questa opzione per instradare i caricamenti dei pacchetti di supporto tramite un server proxy. Esempio: `http://my.proxy.url` .
autoSupport.insicuro	booleano	Salta la verifica TLS per le connessioni proxy AutoSupport quando impostato su `true` . Utilizzare solo per connessioni proxy non sicure. (predefinito: `false`)

Parametro	Tipo	Descrizione
autoSupport.abilitato	booleano	Abilita o disabilita i caricamenti giornalieri del bundle Trident Protect AutoSupport . Quando impostato su `false` , i caricamenti giornalieri programmati sono disabilitati, ma puoi comunque generare manualmente i pacchetti di supporto. (predefinito: `true`)
restoreSkipNamespaceAnnotations	stringa	Elenco separato da virgole di annotazioni dello spazio dei nomi da escludere dalle operazioni di backup e ripristino. Consente di filtrare gli spazi dei nomi in base alle annotazioni.
ripristina Salta le etichette dello spazio dei nomi	stringa	Elenco separato da virgole delle etichette degli spazi dei nomi da escludere dalle operazioni di backup e ripristino. Consente di filtrare gli spazi dei nomi in base alle etichette.

È possibile configurare queste opzioni utilizzando un file di configurazione YAML o i flag della riga di comando:

Utilizzare il file YAML

Fasi

1. Crea un file di configurazione e assegnagli un nome `values.yaml` .
2. Nel file creato, aggiungi le opzioni di configurazione che desideri personalizzare.

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

3. Dopo aver popolato il `values.yaml` file con i valori corretti, applicare il file di configurazione:

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

Usa il flag CLI

Fasi

1. Utilizzare il seguente comando con il `--set` flag per specificare parametri individuali:

```
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
```

Limita i pod Trident Protect a nodi specifici

Puoi utilizzare il vincolo di selezione dei nodi di Kubernetes nodeSelector per controllare quali nodi sono idonei per eseguire i pod Trident Protect, in base alle etichette dei nodi. Per impostazione predefinita, Trident Protect è limitato ai nodi che eseguono Linux. È possibile personalizzare ulteriormente questi vincoli in base alle proprie esigenze.

Fasi

1. Creare un file denominato `nodeSelectorConfig.yaml`.
2. Aggiungere l'opzione nodeSelector al file e modificare il file per aggiungere o modificare le etichette dei nodi da limitare in base alle esigenze dell'ambiente. Ad esempio, il seguente file contiene la restrizione predefinita del sistema operativo, ma riguarda anche una regione e un nome dell'applicazione specifici:

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

3. Applicare i valori dal `nodeSelectorConfig.yaml` file:

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

In questo modo, le restrizioni predefinite vengono sostituite da quelle specificate nel `nodeSelectorConfig.yaml` file.

Informazioni sul copyright

Informazioni sul marchio commerciale