Installa Trident Protect

Trident

NetApp
July 01, 2026

Sommario

Installa Trident Protect

Requisiti di Trident Protect

Per iniziare, verifica la disponibilità del tuo ambiente operativo, dei cluster applicativi, delle applicazioni e delle licenze. Assicurati che il tuo ambiente soddisfi questi requisiti per distribuire e utilizzare Trident Protect.

Compatibilità del cluster Kubernetes di Trident Protect

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)
- Microsoft Azure Kubernetes Service (AKS)
- Red Hat OpenShift
- SUSE Harvester 1.7.0 (ONTAP iSCSI)
- SUSE Rancher
- VMware Tanzu Portfolio
- Kubernetes upstream

- I backup di 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 relativi CRD. Per installare un controller snapshot, fare riferimento a "queste istruzioni".
- Assicurarsi che esista almeno un VolumeSnapshotClass. Per ulteriori informazioni, fai riferimento a "VolumeSnapshotClass".
- Per installare Trident Protect è necessario Helm 4.x o versioni successive.

Compatibilità del backend di storage Trident Protect

Trident Protect supporta i seguenti storage back-end:

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- Array di storage ONTAP
- Google Cloud NetApp Volumes
- Azure NetApp Files

Assicurati che il tuo storage backend soddisfi i seguenti requisiti:

- Assicurarsi che lo storage NetApp connesso al cluster utilizzi Trident 24.02 o una versione successiva (Trident 24.10 è consigliato).

- Assicurati di disporre di un backend di storage NetApp ONTAP.
- Assicurati di aver configurato un bucket di storage a oggetti per l'archiviazione dei backup.
- Crea tutti gli spazi dei nomi applicativi che intendi utilizzare per le applicazioni o per le operazioni di gestione dei dati applicativi. Trident Protect non crea questi spazi dei nomi per te; se specifichi uno spazio dei nomi inesistente in una risorsa personalizzata, l'operazione non riuscirà.

Requisiti per i volumi nas-economy

Trident Protect supporta le operazioni di backup e ripristino su volumi nas-economy. Snapshot, cloni e SnapMirror e la replica su volumi nas-economy non sono attualmente supportati. È necessario abilitare una directory snapshot per ogni volume nas-economy che si prevede di utilizzare con Trident Protect.

Alcune applicazioni non sono compatibili con i volumi che utilizzano una directory snapshot. Per queste applicazioni, è necessario nascondere la directory snapshot eseguendo il seguente comando sul sistema storage ONTAP:

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

È possibile abilitare la directory snapshot eseguendo il seguente comando per ciascun volume nas-economy, sostituendo `<volume-UUID>` con l'UUID del volume che si desidera modificare:

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

È possibile abilitare le directory snapshot per impostazione predefinita per i nuovi volumi impostando l'opzione di configurazione del backend Trident `snapshotDir` su `true`. I volumi esistenti non sono interessati.

Protezione dei dati con le VM KubeVirt

Trident Protect offre 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 blocco delle VM variano tra le versioni di Trident Protect, con le release più recenti che offrono una configurazione semplificata tramite i parametri del chart Helm.

Durante le operazioni di ripristino, qualsiasi `VirtualMachineSnapshots` creato per una macchina virtuale (VM) non viene ripristinato.

Trident Protect 25.10 e versioni successive

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

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

Trident Protect 24.10.1 a 25.06

A partire da Trident Protect 24.10.1, Trident Protect blocca e sblocca automaticamente i file system KubeVirt durante le operazioni di protezione dei dati. Facoltativamente, puoi disabilitare 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 per i file system delle VM KubeVirt durante le operazioni di protezione dei dati. Se si desidera proteggere i dati delle VM KubeVirt utilizzando Trident Protect 24.10, è necessario abilitare manualmente la funzionalità di freeze/unfreeze per i file system prima dell'operazione di protezione dei dati. Ciò garantisce che i file system 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" e quindi utilizzare il seguente comando:

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

Requisiti per la replicazione SnapMirror

NetApp SnapMirror la replica è disponibile per l'uso con Trident Protect per le seguenti soluzioni ONTAP:

- Sistemi NetApp FAS, AFF e ASA on-premises. La replica SnapMirror con Trident protect non è attualmente supportata per i sistemi ASA r2.
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

Requisiti del cluster ONTAP per la replica SnapMirror

Assicurarsi che il cluster ONTAP soddisfi i seguenti requisiti se si prevede di utilizzare la replica SnapMirror:

- **NetApp Trident**: NetApp Trident deve essere presente sia sui cluster Kubernetes di origine che su quelli 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 la versione minima di ONTAP 9.15.1)
- **Licenze**: Le licenze asincrone ONTAP SnapMirror che utilizzano il bundle Data Protection devono essere abilitate sia sul cluster ONTAP di origine che su quello di destinazione. Consulta "Panoramica delle licenze SnapMirror in ONTAP" per ulteriori informazioni.

  A partire da ONTAP 9.10.1, tutte le licenze vengono fornite come NetApp license file (NLF), ovvero un singolo file che abilita più funzionalità. Consulta "Licenze incluse con ONTAP One" per maggiori informazioni.

  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 backend di storage:

- **Cluster e SVM**: I backend di storage ONTAP devono essere sottoposti a peering. Consulta "Panoramica del peering di cluster e SVM" per ulteriori informazioni.

  Assicurarsi che i nomi 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 di Trident / ONTAP per la replica SnapMirror

Trident Protect richiede che tu configuri almeno un backend di storage che supporti la replica sia per il cluster di origine che per il cluster di destinazione. Se il cluster di origine e il cluster di destinazione sono gli stessi, l'applicazione di destinazione dovrebbe utilizzare un backend di storage diverso da quello dell'applicazione di origine per la migliore resilienza.

Requisiti del cluster Kubernetes per la replica SnapMirror

Assicurati che i tuoi cluster Kubernetes soddisfino i seguenti requisiti:

- **AppVault accessibilità**: sia il cluster di origine che quello di destinazione devono avere accesso alla rete per leggere dalla e scrivere sulla AppVault per la replica degli oggetti dell'applicazione.

- **Connettività di rete**: configura le regole del firewall, le autorizzazioni dei bucket e le liste consentite di IP per consentire la comunicazione tra entrambi i cluster e il AppVault attraverso le WAN.

> Molti ambienti aziendali implementano rigide policy firewall sulle connessioni WAN. Verificate questi requisiti di rete con il vostro team infrastrutturale prima di configurare la replica.

Installa e configura Trident Protect

Se l'ambiente soddisfa i requisiti per Trident Protect, è possibile seguire questi passaggi per installare Trident Protect sul cluster. È possibile ottenere Trident Protect da NetApp, oppure installarlo dal proprio registro privato. L'installazione da un registro privato è utile se il cluster non può accedere a Internet.

Installa Trident Protect

Installa Trident Protect da NetApp

Passaggi

1. Aggiungi il repository Trident Helm:

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

2. Utilizza Helm per installare Trident Protect. Sostituisci `<name-of-cluster>` con un nome di 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 100.2602.0 --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
100.2602.0 --create-namespace --namespace trident-protect
```

   La registrazione del debug aiuta il supporto NetApp a risolvere i problemi senza richiedere modifiche al livello di registro o la riproduzione del problema.

Installa Trident Protect da un registro privato

Puoi installare Trident Protect da un registro immagini privato se il tuo cluster Kubernetes non è in grado di accedere a Internet. In questi esempi, sostituisci i valori tra parentesi con le informazioni del tuo ambiente:

Passaggi

1. Scarica le seguenti immagini sulla tua macchina locale, aggiorna i tag e poi caricale nel tuo registro privato:

```
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
```

Ad esempio:

```
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
```

> Per ottenere il grafico Helm, scarica innanzitutto il grafico Helm su una macchina con accesso a Internet utilizzando `helm pull trident-protect --version 100.2602.0 --repo https://netapp.github.io/trident-protect-helm-chart`, quindi copia il file risultante `trident-protect-100.2602.0.tgz` nel tuo ambiente offline e installalo utilizzando `helm install trident-protect ./trident-protect-100.2602.0.tgz` invece del riferimento al repository nel passaggio finale.

2. Crea lo spazio dei nomi di sistema Trident Protect:

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

3. Accedi al registro:

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

4. Crea un pull secret da utilizzare per l'autenticazione del registro privato:

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

5. Aggiungi il repository Trident Helm:

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

6. Crea un file denominato `protectValues.yaml`. Assicurati che contenga le seguenti impostazioni di Trident Protect:

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

> I `imageRegistry` e `imagePullSecrets` valori si applicano a tutte le immagini dei componenti, inclusi `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 full path nel campo del registro. Questo garantirà che tutte le immagini vengano estratte da `<private-registry-url>/<image-name>:<tag>`.

7. Utilizza Helm per installare Trident Protect. Sostituisci `<name_of_cluster>` con un nome di 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 100.2602.0 --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
100.2602.0 --create-namespace --namespace trident-protect -f
protectValues.yaml
```

La registrazione del debug aiuta il supporto NetApp a risolvere i problemi senza richiedere modifiche al livello di registro o la riproduzione del problema.

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

Installa il plugin Trident Protect CLI

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

Installa il plugin Trident Protect CLI

Prima di utilizzare l'utilità della riga di comando, è necessario installarla sul computer utilizzato per accedere al cluster. Seguire questi passaggi, a seconda che il computer utilizzi una CPU x64 o ARM.

Scarica il plugin per CPU Linux AMD64

Passaggi

1. Scarica il plugin Trident Protect CLI:

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

Scarica il plugin per CPU Linux ARM64

Passaggi

1. Scarica il plugin Trident Protect CLI:

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

Scarica il plugin per CPU Mac AMD64

Passaggi

1. Scarica il plugin Trident Protect CLI:

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

Scarica il plugin per CPU Mac ARM64

Passaggi

1. Scarica il plugin Trident Protect CLI:

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

1. Abilita i permessi di esecuzione per il binario del plugin:

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

2. Copia il file binario del plugin in una posizione definita nella variabile PATH. Ad esempio, `/usr/bin` o `/usr/local/bin` (potresti aver bisogno di privilegi elevati):

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

3. Facoltativamente, puoi copiare il file binario del plugin in una posizione nella tua home directory. In questo caso, è consigliato assicurarsi che la posizione faccia parte della variabile PATH:

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

Copiando il plugin in una posizione nella variabile PATH, puoi utilizzare il plugin digitando `tridentctl-protect` o `tridentctl protect` da qualsiasi posizione.

Visualizza la guida del plugin Trident CLI

È possibile utilizzare le funzionalità di aiuto integrate nel plugin per ottenere assistenza dettagliata sulle capacità del plugin:

Passaggi

1. Utilizzare la funzione di aiuto per visualizzare le istruzioni sull'utilizzo:

```
tridentctl-protect help
```

Abilita il completamento automatico dei comandi

Dopo aver installato il plugin Trident Protect CLI, puoi abilitare il completamento automatico per determinati comandi.

Abilita il completamento automatico per la shell Bash

Passaggi

1. Crea lo script di completamento:

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

2. Crea una nuova directory nella tua home directory per contenere lo script:

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

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

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

4. Aggiungi la seguente riga al `~/.bashrc` file nella tua home directory:

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

Abilita il completamento automatico per la Z shell

Passaggi

1. Crea lo script di completamento:

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

2. Crea una nuova directory nella tua home directory per contenere lo script:

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

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

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

4. Aggiungi la seguente riga al `~/.zprofile` file nella tua home directory:

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

Risultato

Al prossimo accesso alla shell, puoi utilizzare il completamento automatico dei comandi con il plugin tridentctl-protect.

Personalizza l'installazione di Trident Protect

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

Specificare i limiti delle risorse del container Trident Protect

È possibile utilizzare un file di configurazione per specificare i limiti delle risorse per i container Trident Protect dopo aver installato Trident Protect. L'impostazione dei limiti delle risorse consente di controllare quante risorse del cluster vengono consumate dalle operazioni di Trident Protect.

Passaggi

1. Crea un file denominato `resourceLimits.yaml`.
2. Compila il file con le opzioni di limitazione delle risorse per i container Trident Protect in base alle esigenze del tuo ambiente.

   Il seguente file di configurazione mostra le impostazioni disponibili e contiene i valori predefiniti per ciascun limite di risorsa:

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

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

Personalizza i vincoli del contesto di sicurezza

È possibile utilizzare un file di configurazione per modificare i vincoli di contesto di sicurezza (OpenShift SCC) per i container Trident Protect dopo aver installato Trident Protect. Questi vincoli definiscono le restrizioni di sicurezza per i pod in un cluster OpenShift.

Passaggi

1. Crea un file denominato `sccconfig.yaml`.
2. Aggiungi l'opzione SCC al file e modifica i parametri secondo le esigenze del tuo ambiente.

   Il seguente esempio mostra 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
crea	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 si opera su OpenShift, o se `scc.create` è impostato su `false`, non verrà creata alcuna risorsa SCC.	true
nome	Specifica il nome dell'SCC.	trident-protect-job
priorità	Definisce la priorità dell'SCC. Gli SCC con valori di priorità più alti vengono valutati prima di quelli con valori più bassi.	1

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

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

Questo sostituirà i valori predefiniti con quelli specificati nel `sccconfig.yaml` file.

Configura impostazioni aggiuntive dell'helm chart Trident Protect

È possibile personalizzare le impostazioni di AutoSupport e il filtraggio dello spazio dei nomi per soddisfare i requisiti specifici. La tabella seguente descrive i parametri di configurazione disponibili:

Parametro	Tipo	Descrizione
AutoSupport.proxy	stringa	Configura un URL proxy per le connessioni NetApp AutoSupport. Usa questa opzione per instradare i caricamenti dei pacchetti di supporto attraverso un proxy server. Esempio: `http://my.proxy.url`.

Parametro	Tipo	Descrizione
AutoSupport.insecure	booleano	Ignora la verifica TLS per le connessioni proxy AutoSupport quando impostato su `true`. Utilizzare solo per connessioni proxy non sicure. (predefinito: `false`)
AutoSupport.enabled	booleano	Abilita o disabilita i caricamenti giornalieri dei bundle AutoSupport di Trident Protect. Quando impostato su `false`, i caricamenti giornalieri programmati sono disabilitati, ma è comunque possibile generare manualmente i bundle di supporto. (predefinito: `true`)
restoreSkipNamespaceAnnotations	stringa	Elenco separato da virgole di annotazioni dei namespace da escludere dalle operazioni di backup e ripristino. Consente di filtrare i namespace in base alle annotazioni.
restoreSkipNamespaceLabels	stringa	Elenco separato da virgole di etichette di namespace da escludere dalle operazioni di backup e ripristino. Consente di filtrare i namespace in base alle etichette.

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

Usa il file YAML

Passaggi

1. Creare un file di configurazione e chiamarlo `values.yaml`.
2. Nel file che hai 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 file `values.yaml` con i valori corretti, applica il file di configurazione:

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

Utilizzare il flag CLI

Passaggi

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

```
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
```

Limitare i pod Trident Protect a nodi specifici

È possibile utilizzare il vincolo di selezione dei nodi Kubernetes nodeSelector per controllare quali dei propri nodi sono idonei a 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.

Passaggi

1. Crea un file denominato `nodeSelectorConfig.yaml`.
2. Aggiungere l'opzione nodeSelector al file e modificare il file per aggiungere o cambiare le etichette dei nodi da limitare in base alle esigenze dell'ambiente. Ad esempio, il file seguente contiene la restrizione

predefinita per il sistema operativo, ma anche una regione e un nome di app specifici:

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

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

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

Questo sostituisce le restrizioni predefinite con quelle specificate nel `nodeSelectorConfig.yaml` file.