Installa Trident Protect

Trident

NetApp
January 15, 2026

Sommario

Installa Trident Protect

Requisiti Trident Protect

Per iniziare, verifica la prontezza 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 Trident Protect

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

- Servizio Amazon Elastic Kubernetes (EKS)
- Motore Google Kubernetes (GKE)
- Servizio Microsoft Azure Kubernetes (AKS)
- Red Hat OpenShift
- SUSE Rancher
- Portafoglio VMware Tanzu
- Kubernetes a monte

- 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 relativi CRD. Per installare un controller snapshot, fare riferimento a "queste istruzioni" .

Compatibilità del backend di archiviazione Trident Protect

Trident Protect supporta i seguenti backend di archiviazione:

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

Assicurati che il tuo backend di archiviazione 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).
- Assicurati di disporre di un backend di archiviazione NetApp ONTAP .
- Assicurati di aver configurato un bucket di archiviazione degli oggetti per l'archiviazione dei backup.
- Crea tutti gli spazi dei nomi delle applicazioni che intendi utilizzare per le applicazioni o per le operazioni di gestione dei dati delle applicazioni. Trident Protect non crea questi namespace per te; se specifichi uno

namespace inesistente in una risorsa personalizzata, l'operazione non riuscirà.

Requisiti per i volumi nas-economy

Trident Protect supporta le operazioni di backup e ripristino sui volumi nas-economy. Snapshot, cloni e replica SnapMirror su volumi nas-economy non sono attualmente supportati. È necessario abilitare una directory snapshot per ogni volume nas-economy che si intende 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 degli snapshot eseguendo il seguente comando sul sistema di archiviazione ONTAP :

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

È possibile abilitare la directory snapshot eseguendo il seguente comando per ogni volume nas-economy, sostituendo `<volume-UUID>` con l'UUID del volume che vuoi 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` A `true` . I volumi esistenti non sono interessati.

Protezione dei dati con le VM KubeVirt

Trident Protect 24.10 e 24.10.1 e versioni successive hanno un comportamento diverso quando si proteggono le applicazioni in esecuzione su VM KubeVirt. Per entrambe le versioni è possibile abilitare o disabilitare il blocco e lo sblocco del file system durante le operazioni di protezione dei dati.

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

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 della VM KubeVirt utilizzando Trident Protect 24.10, è necessario abilitare manualmente la funzionalità di congelamento/scongelamento 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 tramite"configurazione della virtualizzazione" e quindi utilizzando il seguente comando:

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

Trident Protect 24.10.1 e versioni successive

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, 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
```

Requisiti per la replica SnapMirror

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

- Cluster NetApp FAS, AFF e ASA on-premise
- ONTAP Select NetApp ONTAP
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

Requisiti del cluster ONTAP per la replica SnapMirror

Se si prevede di utilizzare la replica SnapMirror, assicurarsi che il cluster ONTAP soddisfi i seguenti requisiti:

- * 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 archiviazione 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 ONTAP SnapMirror che utilizzano il bundle Data Protection devono essere abilitate sia sul cluster ONTAP di origine che su quello di destinazione. Fare riferimento a "Panoramica delle licenze SnapMirror in ONTAP" per maggiori informazioni.

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

  È supportata solo la protezione asincrona SnapMirror .

Considerazioni sul peering per la replica SnapMirror

Se intendi utilizzare il peering backend di archiviazione, assicurati che il tuo ambiente soddisfi i seguenti requisiti:

- **Cluster e SVM**: i backend di archiviazione ONTAP devono essere peered. Fare riferimento a "Panoramica del peering di cluster e SVM" per maggiori 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 archiviazione 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 archiviazione che supporti la replica sia per i cluster di origine che per quelli di destinazione. Se i cluster di origine e di destinazione sono gli stessi, l'applicazione di destinazione dovrebbe utilizzare un backend di archiviazione 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.

Installa e configura Trident Protect

Se il tuo ambiente soddisfa i requisiti per Trident Protect, puoi seguire questi passaggi per installare Trident Protect sul tuo cluster. Puoi ottenere Trident Protect da NetApp oppure installarlo dal tuo registro privato. L'installazione da un registro privato è utile se il cluster non riesce ad accedere a Internet.

Installa Trident Protect

Installa Trident Protect da NetApp

Passi

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 100.2506.0 --create
-namespace --namespace trident-protect
```

Installa Trident Protect da un registro privato

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

Passi

1. Estrai le seguenti immagini sul tuo computer locale, aggiorna i tag e poi inseriscile nel tuo registro privato:

```
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
```

   Per esempio:

```
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. Creare lo spazio dei nomi del 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 segreto pull 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. Aggiungere 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` . Assicurarsi che contenga le seguenti impostazioni Trident Protect:

```
---
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. 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 100.2506.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

Installa il plugin Trident Protect CLI

È possibile utilizzare il plugin della riga di comando Trident Protect, che è un'estensione di Trident `tridentctl` utilità, per creare e interagire con le risorse personalizzate (CR) 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 le CPU Linux AMD64

Passi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-amd64
```

Scarica il plugin per le CPU Linux ARM64

Passi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-arm64
```

Scarica il plugin per le CPU AMD64 di Mac

Passi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-amd64
```

Scarica il plugin per le CPU Mac ARM64

Passi

1. Scarica il plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/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. Per esempio, `/usr/bin` O `/usr/local/bin` (potrebbero essere necessari privilegi elevati):

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

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

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

Copiando il plugin in una posizione nella variabile PATH è possibile utilizzare il plugin digitando `tridentctl-protect` O `tridentctl protect` da qualsiasi luogo.

Visualizza la guida del plugin Trident CLI

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

Passi

1. Utilizzare la funzione di aiuto per visualizzare le istruzioni per l'uso:

```
tridentctl-protect help
```

Abilita il completamento automatico dei comandi

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

Abilita il completamento automatico per la shell Bash

Passi

1. Scarica lo script di completamento:

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

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

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

3. Sposta lo script scaricato in `~/.bash/completions` elenco:

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

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

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

Abilita il completamento automatico per la shell Z

Passi

1. Scarica lo script di completamento:

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

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

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

3. Sposta lo script scaricato in `~/.zsh/completions` elenco:

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

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

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

Risultato

Al successivo accesso alla shell, è possibile utilizzare il completamento automatico dei comandi con il plugin tridentctl-protect.

Personalizza l'installazione Trident Protect

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

Specificare i limiti delle risorse del contenitore Trident Protect

Dopo aver installato Trident Protect, è possibile utilizzare un file di configurazione per specificare i limiti delle risorse per i contenitori Trident Protect. Impostando i limiti delle risorse è possibile controllare la quantità di risorse del cluster consumata dalle operazioni Trident Protect.

Passi

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

   Il seguente file di configurazione di esempio mostra le impostazioni disponibili e contiene i valori predefiniti per ciascun 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 da `resourceLimits.yaml` file:

```
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 (SCC) di OpenShift per i contenitori Trident Protect dopo aver installato Trident Protect. Questi vincoli definiscono le restrizioni di sicurezza per i pod in un cluster Red Hat OpenShift.

Passi

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

   L'esempio seguente 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
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 si opera su OpenShift, o se `scc.create` è impostato su `false` , non verrà creata alcuna risorsa SCC.	VERO
nome	Specifica il nome dell'SCC.	trident-protect-job
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 da `sccconfig.yaml` file:

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

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

Configurare le impostazioni aggiuntive del grafico del timone Trident Protect

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	corda	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` .

Parametro	Tipo	Descrizione
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`)
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	corda	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	corda	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

Passi

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

Passi

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 Kubernetes nodeSelector per controllare quali 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.

Passi

1. Crea un file denominato `nodeSelectorConfig.yaml` .
2. Aggiungere l'opzione nodeSelector al file e modificare il file per aggiungere o modificare le etichette dei nodi in modo da limitare le restrizioni in base alle esigenze del proprio ambiente. Ad esempio, il file seguente contiene la restrizione predefinita del sistema operativo, ma ha anche come target una regione

specifica e un nome di app:

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

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

```
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.

Informazioni sul copyright

Informazioni sul marchio commerciale