安裝 Trident Protect

Trident

NetApp
March 03, 2025

目錄

安裝 Trident Protect 1
Trident 保護需求 1
Trident 保護 Kubernetes 叢集相容性 1
Trident 保護儲存後端相容性 1
NAS 經濟容量需求 2
使用 KubeVirt VM 保護資料 2
SnapMirror 複寫需求 3
安裝及設定 Trident Protect 3
安裝 Trident Protect 4
安裝 Trident Protect CLI 外掛程式 7
安裝 Trident Protect CLI 外掛程式 7
檢視 Trident CLI 外掛程式說明 9
啟用命令自動完成 9
自訂 Trident Protect 安裝 11
指定 Trident Protect 容器資源限制 11
自訂安全性內容限制 12
設定 Trident Protect 的 NetApp AutoSupport 連線 13
將 Trident 保護 Pod 限制在特定節點 14
停用每日 Trident Protect AutoSupport 套件上傳 15

安裝 Trident Protect

Trident 保護需求

首先，請確認您的營運環境，應用程式叢集，應用程式和授權是否準備就緒。確保您的環境符合這些需求，以部署及操作 Trident Protect 。

Trident 保護 Kubernetes 叢集相容性

Trident Protect 可與多種完全託管且自我管理的 Kubernetes 產品相容，包括：

- Amazon Elastic Kubernetes Service （ EKS ）
- Google Kubernetes Engine（GKE）
- Microsoft Azure Kubernetes服務（英文）
- Red Hat OpenShift
- SUSE Rancher
- VMware Tanzu產品組合
- 上游Kubernetes

請確定您安裝 Trident Protect 的叢集已設定執行中的快照控制器和相關的 CRD 。若要安裝快照控制器，請 "這些指示"參閱。

Trident 保護儲存後端相容性

Trident Protect 支援下列儲存設備後端：

- Amazon FSX for NetApp ONTAP 產品
- Cloud Volumes ONTAP
- ONTAP 儲存陣列
- Google Cloud NetApp Volumes
- Azure NetApp Files

確保您的儲存後端符合下列需求：

- 確保連接至叢集的 NetApp 儲存設備使用 Astra Trident 24.02 或更新版本（建議使用 Trident 24.10 ）。
  - 如果 Astra Trident 早於 24.06.1 版，且您計畫使用 NetApp SnapMirror 災難恢復功能，則需要手動啟用 Astra 控制項資源配置程式。
- 確保您擁有最新的 Astra 控制備份程式（預設為 Astra Trident 24.06.1 ）。
- 確保您擁有 NetApp ONTAP 儲存後端。
- 請確定您已設定物件儲存貯體以儲存備份。
- 建立您計畫用於應用程式或應用程式資料管理作業的任何應用程式命名空間。Trident Protect 不會為您建立這些命名空間；如果您在自訂資源中指定不存在的命名空間，則作業將會失敗。

NAS 經濟容量需求

Trident Protect 支援 NAS 經濟型磁碟區的備份與還原作業。目前不支援快照，複製和 SnapMirror 複寫至 NAS 經濟型磁碟區。您需要為打算搭配 Trident Protect 使用的每個 NAS 經濟型磁碟區啟用快照目錄。

某些應用程式與使用 Snapshot 目錄的磁碟區不相容。對於這些應用程式，您需要在 ONTAP 儲存系統上執行下列命令，以隱藏快照目錄：

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

您可以針對每個 NAS 經濟型磁碟區執行下列命令，以您要變更的磁碟區 UUID 取代，來啟用 Snapshot 目錄 `<volume-UUID>`：

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

您可以將 Trident 後端組態選項設定為，為 `true`新的磁碟區預設啟用快照目錄 `snapshotDir`。現有的磁碟區不受影響。

使用 KubeVirt VM 保護資料

當您保護在 KubeVirt VM 上執行的應用程式時， Trident Protect 24.10 和 24.10.1 及更新版本的行為會有所不同。對於這兩個版本，您可以在資料保護作業期間啟用或停用檔案系統凍結和解除凍結。

Trident Protect 24.10

Trident Protect 24.10 無法在資料保護作業期間，自動確保 KubeVirt VM 檔案系統的狀態一致。如果您想要使用 Trident Protect 24.10 來保護 KubeVirt VM 資料，則必須在資料保護作業之前，手動啟用檔案系統的凍結 / 取消凍結功能。如此可確保檔案系統處於一致的狀態。

您可以將 Trident Protect 24.10 設定為在資料保護作業期間管理 VM 檔案系統的凍結和取消凍結"設定虛擬化"，然後使用下列命令：

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

Trident Protect 24.10.1 及更新版本

從 Trident Protect 24.10.1 開始， Trident Protect 會在資料保護作業期間，自動凍結和取消凍結 KubeVirt 檔案系統。您也可以使用下列命令停用此自動行為：

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

SnapMirror 複寫需求

NetApp SnapMirror 複寫可與 Trident Protect 搭配使用，適用於下列 ONTAP 解決方案：

- 內部部署 NetApp FAS ， AFF 和 ASA 叢集
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSX for NetApp ONTAP 產品

SnapMirror 複寫的 ONTAP 叢集需求

如果您打算使用 SnapMirror 複寫，請確保 ONTAP 叢集符合下列需求：

- * Astra 控制備份程式或 Trident * ： Astra 控制備份程式或 Trident 必須同時存在於使用 ONTAP 作為後端的來源叢集和目的地 Kubernetes 叢集上。Trident Protect 使用下列驅動程式所支援的儲存類別，以 NetApp SnapMirror 技術支援複寫：
  - 「ONTAP-NAS」
  - 「ONTAP-SAN」
- * 授權 * ：使用資料保護套件的 ONTAP SnapMirror 非同步授權必須同時在來源和目的地 ONTAP 叢集上啟用。如需詳細資訊、請參閱 "SnapMirror授權概述ONTAP" 。

SnapMirror 複寫的對等考量

如果您計畫使用儲存後端對等，請確保您的環境符合下列需求：

- * 叢集與 SVM* ：必須對 ONTAP 儲存設備的後端進行對等處理。如需詳細資訊、請參閱 "叢集與SVM對等概觀" 。

  確保兩個 ONTAP 叢集之間複寫關係中使用的 SVM 名稱是唯一的。

- **Astra** 控制資源配置程式或 **Trident** 和 **SVM** ：對等的遠端 SVM 必須可用於目的地叢集上的 Astra 控制資源配置程式或 Trident 。
- * 託管後端 * ：您需要在 Trident Protect 中新增及管理 ONTAP 儲存後端，才能建立複寫關係。
- **NVMe over TCP** ： Trident Protect 不支援 NetApp SnapMirror 複寫，用於使用 NVMe over TCP 傳輸協定的儲存後端。

用於 SnapMirror 複寫的 Trident / ONTAP 組態

Trident Protect 要求您至少設定一個儲存後端，以支援來源叢集和目的地叢集的複寫。如果來源叢集和目的地叢集相同、則目的地應用程式應使用不同於來源應用程式的儲存後端、以獲得最佳恢復能力。

安裝及設定 Trident Protect

如果您的環境符合 Trident Protect 的要求，您可以依照下列步驟在叢集上安裝 Trident Protect 。您可以從 NetApp 取得 Trident Protect ，或從您自己的私有登錄安裝。如果您的叢集無法存取網際網路，從私有登錄安裝會很有幫助。

安装 Trident Protect

安裝 **Trident Protect from NetApp**

步驟

1. 新增Trident Helm儲存庫：

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

2. 安裝 Trident Protect 客戶需求日：

```
helm install trident-protect-crds netapp-trident-protect/trident-
protect-crds --version 100.2502.0 --create-namespace --namespace
trident-protect
```

3. 使用 Helm 安裝 Trident Protect 。以叢集名稱取代 `<name_of_cluster>`，該名稱將指派給叢集，用於識別叢集的備份和快照：

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2502.0 --create
-namespace --namespace trident-protect
```

從私有登錄安裝 **Trident Protect**

如果 Kubernetes 叢集無法存取網際網路，您可以從私有映像登錄安裝 Trident Protect 。在這些範例中，請將方括號中的值取代為環境中的資訊：

步驟

1. 將下列影像拉到您的本機電腦，更新標記，然後將它們推送到您的私人登錄：

```
netapp/controller:25.02.0
netapp/restic:25.02.0
netapp/kopia:25.02.0
netapp/trident-autosupport:25.02.0
netapp/exechook:25.02.0
netapp/resourcebackup:25.02.0
netapp/resourcerestore:25.02.0
netapp/resourcedelete:25.02.0
bitnami/kubectl:1.30.2
kubebuilder/kube-rbac-proxy:v0.16.0
```

例如：

```
docker pull netapp/controller:25.02.0
```

```
docker tag netapp/controller:25.02.0 <private-registry-
url>/controller:25.02.0
```

```
docker push <private-registry-url>/controller:25.02.0
```

2. 建立 Trident Protect 系統命名空間：

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

3. 登入登錄：

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

4. 建立用於私人登錄驗證的拉出密碼：

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

5. 新增Trident Helm儲存庫：

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

6. 建立名為的檔案 `protectValues.yaml`。請確定其中包含下列 Trident Protect 設定：

```
---
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. 安裝 Trident Protect 客戶需求日：

```
helm install trident-protect-crds netapp-trident-protect/trident-protect-crds --version 100.2502.0 --create-namespace --namespace trident-protect
```

8. 使用 Helm 安裝 Trident Protect 。以叢集名稱取代 `<name_of_cluster>`，該名稱將指派給叢集，用於識別叢集的備份和快照：

```
helm install trident-protect netapp-trident-protect/trident-protect --set clusterName=<name_of_cluster> --version 100.2502.0 --create-namespace --namespace trident-protect -f protectValues.yaml
```

安裝 Trident Protect CLI 外掛程式

您可以使用 Trident Protect 命令列外掛程式（ Trident 公用程式的延伸 `tridentctl`）來建立自訂資源（ CRS ），並與 Trident 互動。

安裝 Trident Protect CLI 外掛程式

在使用命令列公用程式之前，您必須先將其安裝在用來存取叢集的機器上。根據您的機器使用的是 x64 或 ARM CPU ，請遵循下列步驟。

下載適用於 **Linux AMD64 CPU** 的外掛程式

步驟

1. 下載 Trident Protect CLI 外掛程式：

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-linux-amd64
```

下載適用於 **Linux ARM64 CPU** 的外掛程式

步驟

1. 下載 Trident Protect CLI 外掛程式：

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-linux-arm64
```

下載適用於 **Mac AMD64 CPU** 的外掛程式

步驟

1. 下載 Trident Protect CLI 外掛程式：

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-macos-amd64
```

下載 **Mac ARM64 CPU** 的外掛程式

步驟

1. 下載 Trident Protect CLI 外掛程式：

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-macos-arm64
```

1. 啟用外掛程式二進位檔的執行權限：

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

2. 將外掛程式二進位檔複製到路徑變數中定義的位置。例如， `/usr/bin`或 `/usr/local/bin`（您可能需要提升的 Privileges ）：

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

3. 您也可以選擇將外掛程式二進位檔複製到主目錄中的某個位置。在這種情況下，建議您確保位置是 PATH 變數的一部分：

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

將外掛程式複製到 PATH 變數中的某個位置，可讓您輸入或 `tridentctl protect`從任何位置使用外掛程式 `tridentctl-protect`。

檢視 **Trident CLI** 外掛程式說明

您可以使用內建的外掛程式說明功能，取得外掛程式功能的詳細說明：

步驟

1. 使用說明功能檢視使用指南：

```
tridentctl-protect help
```

啟用命令自動完成

安裝 Trident Protect CLI 外掛程式之後，您可以啟用某些命令的自動完成功能。

啟用 **Bash Shell** 的自動完成功能

步驟

1. 下載完成指令碼：

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.02.0/tridentctl-completion.bash
```

2. 在主目錄中建立新目錄以包含指令碼：

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

3. 將下載的指令碼移至 `~/.bash/completions`目錄：

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

4. 將下列行新增至 `~/.bashrc`主目錄中的檔案：

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

啟用 **Z Shell** 的自動完成功能

步驟

1. 下載完成指令碼：

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.02.0/tridentctl-completion.zsh
```

2. 在主目錄中建立新目錄以包含指令碼：

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

3. 將下載的指令碼移至 `~/.zsh/completions`目錄：

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

4. 將下列行新增至 `~/.zprofile`主目錄中的檔案：

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

結果

下次登入 Shell 時，您可以將命令自動完成功能與 tridentctl-Protect 外掛程式搭配使用。

自訂 Trident Protect 安裝

您可以自訂 Trident Protect 的預設組態，以符合您環境的特定需求。

指定 Trident Protect 容器資源限制

安裝 Trident Protect 之後，您可以使用組態檔來指定 Trident Protect 容器的資源限制。設定資源限制可讓您控制 Trident Protect 作業消耗多少叢集資源。

步驟

1. 建立名為的檔案 `resourceLimits.yaml`。
2. 根據您的環境需求，在檔案中填入 Trident Protect 容器的資源限制選項。

   以下範例組態檔顯示可用的設定，並包含每個資源限制的預設值：

```
---
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. 套用檔案中的值 `resourceLimits.yaml`：

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

自訂安全性內容限制

安裝 Trident Protect 之後，您可以使用組態檔來修改 Trident Protect 容器的 OpenShift 安全性內容限制（ SCC ）。這些限制定義了 Red Hat OpenShift 叢集中 Pod 的安全性限制。

步驟

1. 建立名為的檔案 `sccconfig.yaml`。
2. 將 SCC 選項新增至檔案，並根據環境需求修改參數。

   以下範例顯示 SCC 選項參數的預設值：

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

下表說明 SCC 選項的參數：

參數	說明	預設
建立	決定是否可以建立 SCC 資源。只有在設定為 `true`且 `Helm` 安裝程序識別 `OpenShift` 環境時，才會建立 `SCC` 資源 `scc.create`。如果未在 OpenShift 上操作，或如果設為 `false`，則 `scc.create`不會建立任何 SCC 資源。	是的
名稱	指定SCC的名稱。	Trident 保護工作
優先順序	定義 SCC 的優先順序。優先順序值較高的 SCC 會在值較低之前進行評估。	1

3. 套用檔案中的值 `sccconfig.yaml`：

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

這會將預設值取代為檔案中指定的值 `sccconfig.yaml`。

設定 Trident Protect 的 NetApp AutoSupport 連線

您可以設定連線的 Proxy ，變更 Trident Protect 連線至 NetApp 支援的方式，以上傳支援套件。您可以根據需要將 Proxy 設定為使用安全連線或不安全連線。

設定安全 **Proxy** 連線

步驟

1. 設定安全的 Proxy 連線以上傳 Trident Protect 支援服務包：

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect --set autoSupport.proxy=http://my.proxy.url
--reuse-values
```

設定不安全的 **Proxy** 連線

步驟

1. 設定不安全的 Proxy 連線，以進行 Trident Protect 支援服務套件上傳，以略過 TLS 驗證：

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect --set autoSupport.proxy=http://my.proxy.url
--set autoSupport.insecure=true --reuse-values
```

將 **Trident** 保護 **Pod** 限制在特定節點

您可以使用 Kubernetes nodeSelector 節點選擇限制，根據節點標籤來控制哪些節點符合執行 Trident Protect Pod 的資格。根據預設， Trident Protect 僅限於執行 Linux 的節點。您可以根據自己的需求，進一步自訂這些限制。

步驟

1. 建立名為的檔案 `nodeSelectorConfig.yaml`。
2. 將 nodeSelector 選項新增至檔案，並修改檔案以新增或變更節點標籤，以根據環境需求加以限制。例如，下列檔案包含預設的作業系統限制，但也針對特定區域和應用程式名稱：

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

3. 套用檔案中的值 `nodeSelectorConfig.yaml`：

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

這會將預設限制取代為您在檔案中指定的限制 `nodeSelectorConfig.yaml`。

停用每日 **Trident Protect AutoSupport** 套件上傳

您也可以停用排定的每日 Trident Protect AutoSupport 支援服務套件上傳。

根據預設， Trident Protect 會收集支援資訊，協助處理您可能開啟的任何 NetApp 支援案例，包括叢集和託管應用程式的記錄，度量和拓撲資訊。Trident Protect 會根據每日排程將這些支援套裝組合傳送至 NetApp 。您可以隨時手動"產生支援服務組合"進行。

步驟

1. 建立名為的檔案 `autosupportconfig.yaml`。
2. 將 AutoSupport 選項新增至檔案，並根據環境需求修改參數。

   下列範例顯示 AutoSupport 選項參數的預設值：

   ```
   autoSupport:
     enabled: true
   ```

   當 `autoSupport.enabled`設為 `false`時， AutoSupport 支援套裝組合的每日上傳會停用。

3. 套用檔案中的值 `autosupportconfig.yaml`：

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

版權資訊

商標資訊