安裝 Trident Protect

Trident

NetApp
July 01, 2026

目錄

安裝 Trident Protect 1
Trident Protect 要求 1
Trident Protect Kubernetes 叢集相容性 1
Trident Protect 儲存後端相容性 1
nas-economy Volume 的需求 2
利用 KubeVirt VM 保護資料 2
SnapMirror 複寫的需求 3
安裝並設定 Trident Protect 4
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
查看 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
自訂安全內容限制 12
設定其他 Trident Protect Helm Chart 設定 13
將 Trident Protect Pod 限制在特定節點上 15

安裝 Trident Protect

Trident Protect 要求

首先，請驗證您的運作環境、應用程式叢集、應用程式和授權是否已準備就緒。確保您的環境符合這些要求，以部署和執行 Trident Protect。

Trident Protect Kubernetes 叢集相容性

Trident Protect 與各種完全託管和自架的 Kubernetes 產品相容，包括：

- 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
- 上游 Kubernetes

- Trident Protect 備份僅支援 Linux 運算節點。Windows 運算節點不支援備份作業。
- 確保安裝 Trident Protect 的叢集已配置正在執行中的快照控制器和相關的 CRD。若要安裝快照控制器，請參閱 "這些說明"。
- 確保至少存在一個 VolumeSnapshotClass。如需更多資訊，請參閱 "VolumeSnapshotClass"。
- 安裝 Trident Protect 需要 Helm 4.x 或更高版本。

Trident Protect 儲存後端相容性

Trident Protect 支援以下儲存後端：

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

請確保您的儲存後端符合以下要求：

- 確保連接到叢集的 NetApp 儲存裝置使用 Trident 24.02 或更高版本（建議使用 Trident 24.10）。
- 請確保您擁有 NetApp ONTAP 儲存後端。

- 請確保您已設定用於儲存備份的物件儲存貯體。
- 建立您計劃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了不存在的命名空間，則操作將會失敗。

nas-economy Volume 的需求

Trident Protect 支援對 nas-economy 磁碟區進行備份和還原作業。目前不支援對 nas-economy 磁碟區進行快照、複製和 SnapMirror 複寫。您需要為計劃與 Trident Protect 搭配使用的每個 nas-economy 磁碟區啟用快照目錄。

某些應用程式與使用快照目錄的磁碟區不相容。對於這些應用程式，您需要在 ONTAP 儲存系統上執行以下命令來隱藏快照目錄：

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

您可以透過對每個 nas-economy 磁碟區執行以下命令來啟用快照目錄，並將 `<volume-UUID>`替換為您要變更的磁碟區的 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 `snapshotDir`設為 `true`，為新磁碟區預設啟用快照目錄。現有磁碟區不受影響。

利用 KubeVirt VM 保護資料

Trident Protect 在資料保護作業期間為 KubeVirt 虛擬機器提供檔案系統凍結和解凍功能，以確保資料一致性。虛擬機器凍結操作的配置方法和預設行為因 Trident Protect 版本而異，較新版本透過 Helm Chart 參數提供了更簡化的配置方式。

在復原作業期間，不會復原為虛擬機器 (VM) 建立的任何 `VirtualMachineSnapshots`。

Trident Protect 25.10 及更新版本

Trident Protect 會在資料保護作業期間自動凍結和解凍 KubeVirt 檔案系統，以確保資料一致性。從 Trident Protect 25.10 開始，您可以在 Helm chart 安裝過程中使用 `vm.freeze`參數來停用此功能。此參數預設為啟用。

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

Trident Protect 24.10.1 至 25.06

從 Trident Protect 24.10.1 版本開始，Trident Protect 會在資料保護作業期間自動凍結和解凍 KubeVirt 檔案系統。您也可以選擇使用以下命令停用此自動行為：

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

Trident Protect 24.10

Trident Protect 24.10 在資料保護作業期間不會自動確保 KubeVirt VM 檔案系統的一致性狀態。如果您想使用 Trident Protect 24.10 保護 KubeVirt VM 資料，則需要在執行資料保護作業之前手動啟用檔案系統的凍結/解凍功能。這樣可以確保檔案系統處於一致狀態。

您可以設定 Trident Protect 24.10 來管理資料保護作業期間 VM 檔案系統的凍結和解凍，方法是"配置虛擬化"然後使用下列指令：

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

SnapMirror 複寫的需求

NetApp SnapMirror 複寫可用於 Trident Protect，支援以下 ONTAP 解決方案：

- 本地端 NetApp FAS、AFF 和 ASA 系統。目前 ASA r2 系統尚不支援使用 Trident protect 的 SnapMirror 複寫。
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

ONTAP 叢集 SnapMirror 複製要求

如果您打算使用 SnapMirror 複製功能，請確保您的 ONTAP 叢集符合下列要求：

- **NetApp Trident**：NetApp Trident 必須同時存在於使用 ONTAP 作為後端的來源和目標 Kubernetes 叢集上。Trident Protect 支援使用 NetApp SnapMirror 技術，透過以下驅動程式支援的儲存類別進行複寫：
  - `ontap-nas`：NFS
  - `ontap-san`：iSCSI
  - `ontap-san`：FC
  - `ontap-san`：NVMe/TCP （需要最低 ONTAP 版本 9.15.1）
- 授權：必須在來源和目的地 ONTAP 叢集上啟用使用資料保護套件的 ONTAP SnapMirror 非同步授權。如需詳細資訊，請參閱 "SnapMirror 授權總覽（ONTAP）"。

從 ONTAP 9.10.1 開始，所有授權均以 NetApp 授權檔案（NLF）的形式提供，這是一個可啟用多項功能的單一檔案。請參閱 "ONTAP One 隨附的授權" 以取得更多資訊。

僅支援 SnapMirror 非同步保護。

SnapMirror 複寫的對等考量

如果您打算使用儲存後端對等互連，請確保您的環境符合以下要求：

- 叢集和 **SVM**：ONTAP 儲存後端必須建立對等連線。如需詳細資訊，請參閱 "叢集和 SVM 對等連接概述"。

  確保兩個 ONTAP 叢集之間複寫關係中使用的 SVM 名稱是唯一的。

- **NetApp Trident** 和 **SVM**：對等遠端 SVM 必須可供目標叢集上的 NetApp Trident 使用。
- 託管後端：您需要在 Trident Protect 中新增和管理 ONTAP 儲存後端，以建立複製關係。

Trident / ONTAP 用於 **SnapMirror** 複製的配置

Trident Protect 要求您至少設定一個支援來源叢集和目標叢集複寫的儲存後端。如果來源叢集和目標叢集相同，為了獲得最佳的恢復能力，目標應用程式應使用與來源應用程式不同的儲存後端。

Kubernetes 叢集 **SnapMirror** 複製要求

請確保您的 Kubernetes 叢集符合以下要求：

- **AppVault** 可存取性：來源叢集和目標叢集都必須具有網路存取權限，才能從 AppVault 讀取和寫入資料以進行應用程式物件複製。
- 網路連線：設定防火牆規則、儲存桶權限和 IP 允許列表，以啟用兩個叢集之間以及 AppVault 跨 WAN 的通訊。

許多企業環境在 WAN 連線上實施嚴格的防火牆原則。在設定複寫之前，請先與您的基礎架構團隊確認這些網路需求。

安裝並設定 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，也可以從您自己的私人登錄安裝。如果您的叢集無法存取網際網路，從私人登錄安裝會很有幫助。

安裝 Trident Protect

從 **NetApp** 安裝 **Trident Protect**

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. 使用 Helm 安裝 Trident Protect。將 `<name-of-cluster>`替換為叢集名稱，該名稱將分配給叢集並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 100.2602.0 --create
-namespace --namespace trident-protect
```

3. （選用）若要啟用偵錯日誌記錄（建議用於疑難排解），請使用：

```
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
```

偵錯日誌有助於 NetApp 支援人員在無需更改日誌等級或重現問題的情況下排查問題。

從私人註冊表安裝 **Trident Protect**

如果您的 Kubernetes 叢集無法存取網際網路，您可以從私人映像登錄安裝 Trident Protect。在這些範例中，請將方括號中的值替換為您環境中的資訊：

步驟

1. 將以下映像拉取到本機、更新標籤、然後將其推送到您的私有登錄：

```
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
```

例如：

```
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
```

若要取得 Helm chart，請先在可存取互聯網的機器上使用 `helm pull trident-protect --version 100.2602.0 --repo https://netapp.github.io/trident-protect-helm-chart`下載 Helm chart，然後將產生的 `trident-protect-100.2602.0.tgz`檔案複製到您的離線環境中，並在最後一步中使用 `helm install trident-protect ./trident-protect-100.2602.0.tgz`而不是儲存庫引用進行安裝。

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. 建立用於私有登錄驗證的 pull secret ：

```
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. 建立一個名為 `protectValues.yaml`的檔案。確保該檔案包含以下 Trident Protect 設定：

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

> ⓘ imageRegistry 和 imagePullSecrets 值適用於所有組件鏡像，包括 resourcebackup 和 resourcerestore。如果您將鏡像推送到登錄中的特定儲存庫路徑（例如， example.com:443/my-repo），請在登錄欄位中包含完整路徑。這將確保所有鏡像都從 <private-registry-url>/<image-name>:<tag> 拉取。

7. 使用 Helm 安裝 Trident Protect。將 `<name_of_cluster>`替換為叢集名稱，該名稱將分配給叢集並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 100.2602.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

8. （選用）若要啟用偵錯日誌記錄（建議用於疑難排解），請使用：

```
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
```

偵錯日誌有助於 NetApp 支援人員在無需更改日誌等級或重現問題的情況下排查問題。

> ⓘ 有關 Helm chart 配置的其他選項，包括 AutoSupport 設定和命名空間過濾，請參閱 "自訂 Trident Protect 安裝"。

安裝 Trident Protect CLI 外掛程式

您可以使用 Trident Protect 命令列外掛程式（ Trident tridentctl 實用程式的擴充功能）來建立和與 Trident Protect 自訂資源（ CR ）進行互動。

安裝 Trident Protect CLI 外掛程式

在使用命令列實用程式之前，需要將其安裝到用於存取叢集的電腦上。請根據您的電腦使用的是 x64 還是 ARM CPU，依照以下步驟操作。

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/26.02.0/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/26.02.0/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/26.02.0/tridentctl-protect-macos-amd64
```

下載適用於 **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/26.02.0/tridentctl-protect-macos-arm64
```

1. 啟用外掛程式二進位檔案的執行權限：

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

2. 將插件二進位檔案複製到 PATH 環境變數中定義的某個位置。例如， `/usr/bin` 或 `/usr/local/bin`（您可能需要管理員權限）：

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

3. 您也可以選擇將插件二進位檔案複製到您主目錄下的某個位置。在這種情況下，建議確保該位置已新增至您的 PATH 環境變數：

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

> 將外掛程式複製到 PATH 變數中的某個位置,即可透過鍵入 `tridentctl-protect` 或 `tridentctl protect` 從任何位置使用該外掛程式。

查看 **Trident CLI** 外掛程式說明

您可以使用內建的外掛程式說明功能，以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 外掛程式後、您可以為某些命令啟用自動完成功能。

為 **Bash shell** 啟用自動完成功能

步驟

1. 建立完成指令碼：

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

2. 在您的使用者目錄下建立一個新目錄，用於存放腳本：

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

3. 將下載的腳本移至 `~/.bash/completions` 目錄：

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

4. 將以下行新增至您主目錄中的 `~/.bashrc` 檔案：

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

為 **Z shell** 啟用自動完成

步驟

1. 建立完成指令碼：

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

2. 在您的使用者目錄下建立一個新目錄，用於存放腳本：

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

3. 將下載的腳本移至 `~/.zsh/completions` 目錄：

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

4. 將以下行新增至您主目錄中的 `~/.zprofile` 檔案：

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

結果

下次登入 shell 時、您可以使用 tridentctl-protect 外掛程式進行命令自動完成。

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 後，您可以使用組態檔為 Trident Protect 容器指定資源限制。設定資源限制可讓您控制 Trident Protect 作業所消耗的叢集資源量。

步驟

1. 建立一個名為 `resourceLimits.yaml`的檔案。
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. 應用 `resourceLimits.yaml` 檔案中的值：

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

自訂安全內容限制

安裝 Trident Protect 後，您可以使用組態檔修改 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. 建立一個名為 `sccconfig.yaml`的檔案。
2. 在檔案中新增 SCC 選項，並根據您的環境需求修改參數。

   以下範例顯示 SCC 選項的參數預設值：

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

   下表描述了 SCC 選項的參數：

參數	說明	預設
建立	確定是否可以建立 SCC 資源。僅當 `scc.create` 設定為 `true` 且 Helm 安裝程序識別 OpenShift 環境時，才會建立 SCC 資源。如果未在 OpenShift 中執行，或 `scc.create` 設定為 `false`，則不會建立 SCC 資源。	true
姓名	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 會在優先順序值較低的 SCC 之前進行評估。	1

3. 應用 `sccconfig.yaml` 檔案中的值：

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

這將把預設值替換為 `sccconfig.yaml` 檔案中指定的值。

設定其他 Trident Protect Helm Chart 設定

您可以自訂 AutoSupport 設定和命名空間篩選，以滿足您的特定需求。下表說明可用的組態參數：

參數	類型	說明
autoSupport.proxy	字串	設定 NetApp AutoSupport 連線的代理 URL。使用此功能可將支援包上傳路由至代理伺服器。例如：`http://my.proxy.url`。
autoSupport.insecure	布林值	設定為 `true` 時，會跳過 AutoSupport Proxy 連線的 TLS 驗證。僅用於不安全的 Proxy 連線。（預設值： `false`）
autoSupport.enabled	布林值	啟用或停用每日 Trident Protect AutoSupport 捆綁包上傳。設定為 `false` 時，每日定時上傳將被停用，但您仍然可以手動產生支援捆綁包。（預設值： `true`）
restoreSkipNamespaceAnnotations	字串	以逗號分隔的命名空間註解清單，用於從備份和還原作業中排除。讓您根據註解篩選命名空間。

參數	類型	說明
restoreSkipNamespaceLabels	字串	以逗號分隔的命名空間標籤清單，用於從備份和還原作業中排除。允許您根據標籤篩選命名空間。

您可以使用 YAML 組態檔或命令列旗標來設定這些選項：

使用 **YAML** 檔案

步驟

1. 建立一個設定檔並將其命名為 `values.yaml`。
2. 在您建立的檔案中，新增您想要自訂的組態選項。

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

3. 在 `values.yaml` 檔案中填入正確的值後，套用設定檔：

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

使用 **CLI** 標誌

步驟

1. 使用以下命令並加上 `--set` 標誌位元來指定各個參數：

```
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
```

將 **Trident Protect Pod** 限制在特定節點上

您可以使用 Kubernetes nodeSelector 節點選擇約束，根據節點標籤來控制哪些節點可以執行 Trident Protect Pod。預設情況下，Trident Protect 僅限於執行 Linux 的節點。您可以根據需要進一步自訂這些約束。

步驟

1. 建立一個名為 `nodeSelectorConfig.yaml`的檔案。
2. 將 nodeSelector 選項新增至檔案中，並修改該檔案以新增或變更節點標籤，從而根據您的環境需求進行限制。例如，以下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. 應用 `nodeSelectorConfig.yaml` 檔案中的值：

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

這將預設限制替換為您在 `nodeSelectorConfig.yaml` 文件中指定的限制。