



# 安裝 Trident Protect

## Trident

NetApp  
January 14, 2026

# 目錄

安裝Trident Protect	1
Trident保護要求	1
Trident Protect Kubernetes 叢集相容性	1
Trident Protect 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
安裝並設定Trident Protect	3
安裝Trident Protect	4
安裝Trident Protect CLI 插件	7
安裝Trident Protect CLI 插件	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NetApp AutoSupport連線以啟用Trident Protect	13
將Trident Protect Pod 限制在特定節點上	14
停用每日Trident ProtectAutoSupport捆綁包上傳。	15

# 安裝 Trident Protect

## Trident 保護要求

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

### 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 服務 (英文)
- Red Hat OpenShift
- SUSE Rancher
- VMware Tanzu 產品組合
- 上游 Kubernetes



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 ["這些指示"](#)。

### Trident Protect 儲存後端相容性

Trident Protect 支援以下儲存後端：

- 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 經濟容量需求

Trident Protect 支援對 nas-economy 磁碟區進行備份和復原作業。目前不支援將快照、克隆和SnapMirror複製到 nas-economy 磁碟區。您需要為計劃與Trident Protect 一起使用的每個 nas-economy 磁碟區啟用快照目錄。



某些應用程式與使用 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

Trident Protect 24.10 和 24.10.1 及更高版本在保護運行在 KubeVirt VM 上的應用程式時，行為有所不同。對於這兩個版本，您都可以在資料保護操作期間啟用或停用檔案系統凍結和解凍。

### 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 保護您的 KubeVirt VM 數據，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 檔案系統的凍結與解凍。["設定虛擬化"](#)然後使用以下命令：

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

### Trident Protect 24.10.1 及更高版本

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解凍 KubeVirt 檔案系統。您也可以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
```

## SnapMirror 複寫需求

NetApp SnapMirror 複製功能可與 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 Control Provisioner 或 Trident \***：使用 ONTAP 作為後端的來源 Kubernetes 叢集和目標 Kubernetes 叢集上必須同時存在 Astra Control Provisioner 或 Trident。Trident Protect 支援使用 NetApp SnapMirror 技術進行複製，該技術使用以下驅動程式支援的儲存類別：
  - 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 中新增和管理 ONTAP 儲存後端，以建立複製關係。
- **NVMe over TCP**：Trident Protect 不支援使用 NVMe over TCP 協定的儲存後端的 NetApp SnapMirror 複製。

### 用於 SnapMirror 複寫的 Trident / ONTAP 組態

Trident Protect 要求您至少設定一個支援來源叢集和目標叢集複製的儲存後端。如果來源叢集和目標叢集相同，為了獲得最佳彈性，目標應用程式應該使用與來源應用程式不同的儲存後端。

## 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，或從您自己的私人註冊表中安裝它。如果您的叢集無法存取互聯網，從私有註冊表安裝會很有幫助。

# 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. 安裝Trident Protect CRD：

```
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。代替 ``<name-of-cluster>`` 集群名稱將分配給集群，並用於標識集群的備份和快照：

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

## 從私人註冊表安裝Trident Protect

如果您的 Kubernetes 叢集無法存取互聯網，您可以從私人鏡像倉庫安裝Trident Protect。在這些範例中，請將括號中的值替換為您環境中的資訊：

### 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. 建立一個名為的文件 `protectValues.yaml`。請確保其中包含以下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 CRD :

```

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。代替 ``<name_of_cluster>`` 集群名稱將分配給集群，並用於標識集群的備份和快照：

```

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 插件

您可以使用Trident Protect 命令列插件，它是Trident的一個擴充。`tridentctl`用於建立和與Trident Protect 自訂資源 (CR) 互動的實用程式。

## 安裝Trident Protect CLI 插件

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：

```
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：

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

## 下載 **Mac AMD64 CPU** 的外掛程式

### 步驟

1. 下載Trident Protect CLI 外掛：

```
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：

```
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/
```

- 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 外掛程式後，您可以為某些指令啟用自動補全功能。

## 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 的預設配置，以符合您環境的特定要求。

### 指定Trident Protect 容器資源限制

安裝Trident Protect 後，您可以使用設定檔來指定Trident Protect 容器的資源限制。設定資源限制可以控制Trident Protect 作業消耗叢集資源的程度。

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 後，您可以使用設定檔來修改 Trident Protect 容器的 OpenShift 安全上下文約束 (SCC)。這些約束定義了 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 NetApp AutoSupport 連線以啟用 Trident Protect

您可以設定連線代理程式來變更 Trident Protect 連線到 NetApp Support 以上傳支援包的方式。您可以根據需要將代理配置為使用安全連線或不安全連線。

## 設定安全 Proxy 連線

### 步驟

1. 為Trident Protect 支援包上傳設定安全代理連線：

```
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. 設定一個不安全的代理連接，用於上傳Trident Protect 支援包，跳過 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 Protect Pod 限制在特定節點上

您可以使用 Kubernetes nodeSelector 節點來選擇約束，根據節點標籤來控制哪些節點有資格執行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限制條件。

### 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 ProtectAutoSupport捆綁包上傳。

您也可以選擇停用每日定期上傳的Trident Protect AutoSupport支援包。



預設情況下，Trident Protect 會收集有助於處理您可能提交的任何NetApp支援案例的支援訊息，包括叢集和受管應用程式的日誌、指標和拓撲資訊。Trident Protect 會按方案每日傳送這些支援包給NetApp。您可以手動操作["產生支援服務組合"](#)隨時。

### 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

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。