



管理後端 Trident

NetApp
October 30, 2026

目錄

管理後端	1
使用 kubectl 管理 Trident 後端	1
刪除後端	1
檢視現有的後端	1
更新後端	1
使用 tridentctl 管理 Trident 後端	2
建立後端	2
刪除後端	2
檢視現有的後端	3
更新後端	3
識別使用後端的儲存類別	3
選擇 Trident 後端管理方法	3
管理後端的選項	3
使用 tridentctl`後端管理`TridentBackendConfig	4
使用 TridentBackendConfig`後端管理`tridentctl	8

管理後端

使用 **kubectl** 管理 **Trident** 後端

了解如何使用 `kubectl` 執行後端管理操作。

刪除後端

刪除 `TridentBackendConfig` 會指示 `Trident` 刪除/保留後端（取決於 `deletionPolicy`）。若要刪除後端，請確保 `deletionPolicy` 設定為刪除。若要僅刪除 `TridentBackendConfig`，請確保 `deletionPolicy` 設定為保留。這樣可以確保後端仍然存在，並且可以使用 `tridentctl` 進行管理。

執行下列命令：

```
kubectl delete tbc <tbc-name> -n trident
```

`Trident` 不會刪除 `Kubernetes` 正在使用的 `Secret TridentBackendConfig`。`Kubernetes` 用戶負責清理 `Secret`。刪除 `Secret` 時必須格外小心。只有當 `Secret` 不再被後端使用時，才應刪除。

檢視現有的後端

執行下列命令：

```
kubectl get tbc -n trident
```

您也可以執行 `tridentctl get backend -n trident` 或 `tridentctl get backend -o yaml -n trident` 來取得所有現有後端的清單。此清單還將包括使用 `tridentctl` 建立的後端。

更新後端

更新後端的原因可能有很多：

- 儲存系統的憑證已更改。若要更新憑證，必須更新 `TridentBackendConfig` 物件中使用的 `Kubernetes Secret`。`Trident` 將使用提供的最新憑證自動更新後端。執行以下命令更新 `Kubernetes Secret`：

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- 需要更新參數（例如正在使用的 `ONTAP SVM` 的名稱）。
 - 您可以使用以下命令直接透過 `Kubernetes` 更新 `TridentBackendConfig` 物件：

```
kubectl apply -f <updated-backend-file.yaml>
```

- 或者，您可以使用以下命令對現有 `TridentBackendConfig` CR 進行更改：

```
kubectl edit tbc <tbc-name> -n trident
```



- 如果後端更新失敗，後端將繼續保持其上次已知的配置。您可以透過執行 ``kubectl get tbc <tbc-name> -o yaml -n trident`` 或 ``kubectl describe tbc <tbc-name> -n trident`` 來查看日誌以確定原因。
- 在您識別並修正組態檔的問題後，您可以重新執行更新命令。

使用 `tridentctl` 管理 Trident 後端

了解如何使用 ``tridentctl`` 執行後端管理操作。

建立後端

建立 "[後端組態檔](#)" 後，執行以下命令：

```
tridentctl create backend -f <backend-file> -n trident
```

如果後端建立失敗，則表示後端配置存在問題。您可以執行以下命令查看日誌以確定原因：

```
tridentctl logs -n trident
```

在您發現並修正設定檔中的問題後，您可以再次執行 `create` 命令。

刪除後端

若要從 Trident 刪除後端、請執行下列操作：

1. 取得後端名稱：

```
tridentctl get backend -n trident
```

2. 刪除後端：

```
tridentctl delete backend <backend-name> -n trident
```



如果 Trident 已從此後端配置了仍然存在的磁碟區和快照，則刪除該後端將阻止其配置新的磁碟區。該後端將繼續以「正在刪除」狀態存在。

檢視現有的後端

若要檢視 Trident 已知的後端、請執行下列步驟：

- 若要取得摘要、請執行下列命令：

```
tridentctl get backend -n trident
```

- 若要取得所有詳細資料、請執行下列命令：

```
tridentctl get backend -o json -n trident
```

更新後端

建立新的後端組態檔後，執行以下命令：

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

如果後端更新失敗，則表示後端配置存在問題，或者您嘗試了無效的更新操作。您可以執行以下命令查看日誌以確定原因：

```
tridentctl logs -n trident
```

在您發現並修正設定檔中的問題後，您可以再次執行 update 命令。

識別使用後端的儲存類別

這是一個可以使用 tridentctl 為後端物件輸出的 JSON 資料來回答的問題範例。這需要用到 jq 實用程式，您需要先安裝該程式。

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

這也適用於使用 TridentBackendConfig 建立的後端。

選擇 Trident 後端管理方法

了解在 Trident 中管理後端的不同方式。

管理後端的選項

隨著 `TridentBackendConfig` 的推出，管理員現在有兩種獨特的後端管理方式。這引發了以下問題：

- 使用 `tridentctl` 建立的後端可以用 `TridentBackendConfig` 來管理嗎？
- 使用 `TridentBackendConfig` 建立的後端可以使用 `tridentctl` 來管理嗎？

使用 `tridentctl` 後端管理 `TridentBackendConfig`

本節說明如何通過 Kubernetes 介面直接建立 `tridentctl` 並建立 `TridentBackendConfig` 物件來管理後端的步驟。

這適用於以下情況：

- 已存在的後端，沒有 `TridentBackendConfig`，因為它們是用 `tridentctl` 建立的。
- 使用 `tridentctl` 建立的新後端，同時存在其他 `TridentBackendConfig` 物件。

無論哪種情況，後端都將繼續存在，Trident 負責排程磁碟區並對其進行操作。管理員此時有兩種選擇：

- 繼續使用 `tridentctl` 來管理使用它建立的後端。
- 將使用 `tridentctl` 建立的後端繫結至新的 `TridentBackendConfig` 物件。這樣做意味著後端將使用 `kubectl` 而非 `tridentctl` 進行管理。

若要使用 `kubectl` 管理現有後端，您需要建立一個 `TridentBackendConfig` 綁定到現有後端。以下概述了其工作原理：

1. 建立 Kubernetes Secret。此 Secret 包含 Trident 與儲存叢集/服務通訊所需的認證資料。
2. 建立 `TridentBackendConfig` 物件。其中包含儲存叢集 / 服務的詳細資訊、以及在上一步中建立的機密參照。必須小心指定相同的組態參數（例如 `spec.backendName`、`spec.storagePrefix`、`spec.storageDriverName` 等）。`spec.backendName` 必須設定為現有後端的名稱。

步驟 0：識別後端

若要建立 `TridentBackendConfig` 綁定到現有後端，您需要取得後端配置。在本例中，我們假設後端是使用以下 JSON 定義建立的：

```
tridentctl get backend ontap-nas-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                      UUID                      |
| STATE  | VOLUMES |                      |                      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-nas-backend      | ontap-nas      | 52f2eb10-e4c6-4160-99fc-96b3be5ab5d7 |
| online |      25 |                      |                      |
+-----+-----+-----+-----+
```

```
cat ontap-nas-backend.json
```

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}

```

步驟 1：建立 Kubernetes Secret

建立一個包含後端憑證的 Secret，如下例所示：

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

步驟 2：建立 TridentBackendConfig CR

下一步是建立 TridentBackendConfig CR，使其自動綁定到現有 ontap-nas-backend（如本例所示）。請確保滿足以下要求：

- 相同的後端名稱已在 `spec.backendName` 中定義。
- 配置參數與原始後端相同。
- 虛擬資源池（如果存在）必須保持與原始後端相同的順序。
- 認證資訊透過 Kubernetes Secret 提供，而非以純文字形式提供。

在這種情況下，`TridentBackendConfig` 看起來會像這樣：

```
cat backend-tbc-ontap-nas.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
```

```

backendName: ontap-nas-backend
svm: trident_svm
credentials:
  name: mysecret
defaults:
  spaceReserve: none
  encryption: 'false'
labels:
  store: nas_store
region: us_east_1
storage:
- labels:
  app: msoffice
  cost: '100'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  app: mysqldb
  cost: '25'
  zone: us_east_1d
  defaults:
    spaceReserve: volume
    encryption: 'false'
    unixPermissions: '0775'

```

```

kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created

```

步驟 3：驗證 **TridentBackendConfig** CR 的狀態

在 **TridentBackendConfig** 建立之後，其階段必須為 `Bound`。它還應反映與現有後端相同的後端名稱和 UUID。

```

kubectl get tbc tbc-ontap-nas-backend -n trident

```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS		
tbc-ontap-nas-backend	ontap-nas-backend	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7
Bound Success		

```

#confirm that no new backends were created (i.e., TridentBackendConfig did not end up creating a new backend)

```

```
tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+-----+-----+
| ontap-nas-backend      | ontap-nas      | 52f2eb10-e4c6-4160-99fc-96b3be5ab5d7 |
| online |      25 |
+-----+-----+
+-----+-----+-----+-----+
```

現在後端將完全使用 `tbc-ontap-nas-backend TridentBackendConfig` 物件進行管理。

使用 `TridentBackendConfig` 後端管理 `tridentctl`

`tridentctl` 可用於列出使用 `TridentBackendConfig` 建立的後端。此外，管理員也可以選擇透過 `tridentctl` 完全管理這些後端，方法是刪除 `TridentBackendConfig` 並確保 `spec.deletionPolicy` 設定為 `retain`。

步驟 0：識別後端

例如，假設使用 `TridentBackendConfig` 建立了以下後端：

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME          BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-0a5315ac5f82
Bound Success  ontap-san          delete

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend      | ontap-san      | 81abcb27-ea63-49bb-b606-0a5315ac5f82 |
| online |      33 |
+-----+-----+
+-----+-----+-----+-----+
```

從輸出結果可以看出，`TridentBackendConfig` 已成功創建，並綁定到後端（觀察後端的 UUID）。

步驟 1：確認 `deletionPolicy` 已設定為 `retain`

讓我們來看一下 `deletionPolicy` 的值。這需要設定為 `retain`。這可確保刪除 `TridentBackendConfig` CR 時，後端定義仍會存在，並可使用 `tridentctl` 進行管理。

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME          BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-0a5315ac5f82    Bound    Success    ontap-san    delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME          BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-0a5315ac5f82    Bound    Success    ontap-san    retain
```



除非 `deletionPolicy` 設定為 `retain`，否則不要進行下一步。

步驟 2：刪除 `TridentBackendConfig` CR

最後一步是刪除 `TridentBackendConfig` CR。確認 `deletionPolicy` 設定為 `retain` 後，即可執行刪除操作：

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|      NAME      | STORAGE DRIVER |                               UUID
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-0a5315ac5f82 |
| online | 33      |
+-----+-----+
+-----+-----+-----+-----+
```

刪除 `TridentBackendConfig` 物件時，`Trident` 只是將其移除，而不會實際刪除後端本身。

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。