使用 Astra Control Provisioner

Astra Control Center

NetApp
April 25, 2024

目錄

使用 Astra Control Provisioner

設定儲存後端加密

使用 Astra Control Provisioner 、您可以為託管叢集與儲存後端之間的流量啟用加密、藉此改善資料存取安全性。

Astra Control Provisioner 支援兩種儲存後端的 Kerberos 加密：

- * 內部部署 ONTAP * ： Astra 控制備置程式支援透過 NFSv3 和 NFSv4 連線進行 Kerberos 加密、從 Red Hat OpenShift 和上游 Kubernetes 叢集到內部部署 ONTAP 磁碟區。
- * Azure NetApp Files * - Astra 控制備置程式支援透過 NFSv4.1 連線進行 Kerberos 加密、從上游 Kubernetes 叢集到 Azure NetApp Files Volume 。

您可以建立、刪除、調整大小、快照、複製、 唯讀複製及匯入使用 NFS 加密的磁碟區。

使用內部部署的 **ONTAP** 磁碟區來設定在線上 **Kerberos** 加密

您可以在託管叢集與內部部署 ONTAP 儲存後端之間的儲存流量上啟用 Kerberos 加密。

> 內部部署 ONTAP 儲存後端的 NFS 流量 Kerberos 加密僅支援使用 `ontap-nas` 儲存驅動程式：

開始之前

- 確保您擁有 "已啟用 Astra Control Provisioner" 在託管叢集上。
- 請確定您可以存取 `tridentctl` 公用程式：
- 確保您具有 ONTAP 儲存後端的管理員存取權。
- 確保您知道將從 ONTAP 儲存後端共用的磁碟區名稱。
- 請確定您已準備好 ONTAP 儲存 VM 、以支援 NFS 磁碟區的 Kerberos 加密。請參閱 "在資料LIF上啟用Kerberos" 以取得相關指示。
- 請確定您使用 Kerberos 加密的任何 NFSv4 磁碟區都已正確設定。請參閱的「 NetApp NFSv4 網域組態」一節（第 13 頁） "NetApp NFSv4 增強與最佳實務指南"。

新增或修改 **ONTAP** 匯出原則

您需要將規則新增至現有的 ONTAP 匯出原則、或建立新的匯出原則、以支援 ONTAP 儲存 VM 根磁碟區的 Kerberos 加密、以及與上游 Kubernetes 叢集共用的任何 ONTAP 磁碟區。您新增的匯出原則規則或您建立的新匯出原則需要支援下列存取通訊協定和存取權限：

存取傳輸協定

使用 NFS 、 NFSv3 和 NFSv4 存取通訊協定來設定匯出原則。

存取詳細資料

您可以根據對磁碟區的需求、設定 Kerberos 加密的三個不同版本之一：

- * Kerberos 5* - （驗證與加密）

- * Kerberos 5i* - （身分識別保護的驗證與加密）
- * Kerberos 5p* - （身分識別與隱私保護的驗證與加密）

使用適當的存取權限來設定 ONTAP 匯出原則規則。例如、如果叢集將使用 Kerberos 5i 和 Kerberos 5p 加密混合安裝 NFS 磁碟區、請使用下列存取設定：

類型	唯讀存取	讀取 / 寫入存取權	超級使用者存取權
UNIX	已啟用	已啟用	已啟用
Kerberos 5i	已啟用	已啟用	已啟用
Kerberos 5p	已啟用	已啟用	已啟用

請參閱下列文件、瞭解如何建立 ONTAP 匯出原則和匯出原則規則：

- "建立匯出原則"
- "新增規則至匯出原則"

建立儲存後端

您可以建立內含 Kerberos 加密功能的 Astra Control Provisioner 儲存後端組態。

關於這項工作

當您建立設定 Kerberos 加密的儲存後端組態檔案時、可以使用指定三個不同版本的 Kerberos 加密之一 `spec.nfsMountOptions` 參數：

- `spec.nfsMountOptions: sec=krb5` （驗證與加密）
- `spec.nfsMountOptions: sec=krb5i` （身分識別保護的驗證與加密）
- `spec.nfsMountOptions: sec=krb5p` （身分識別與隱私保護的驗證與加密）

只指定一個 Kerberos 層級。如果您在參數清單中指定多個 Kerberos 加密層級、則只會使用第一個選項。

步驟

1. 在託管叢集上、使用下列範例建立儲存後端組態檔案。以您環境的資訊取代括弧 <> 中的值：

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret
```

2. 使用您在上一個步驟中建立的組態檔來建立後端：

```
tridentctl create backend -f <backend-configuration-file>
```

如果後端建立失敗、表示後端組態有問題。您可以執行下列命令來檢視記錄、以判斷原因：

```
tridentctl logs
```

識別並修正組態檔的問題之後、您可以再次執行create命令。

建立儲存類別

您可以建立儲存類別、以使用 Kerberos 加密來配置磁碟區。

關於這項工作

當您建立儲存類別物件時、可以使用指定三個不同版本的 Kerberos 加密之一 `mountOptions` 參數：

- `mountOptions: sec=krb5` （驗證與加密）
- `mountOptions: sec=krb5i` （身分識別保護的驗證與加密）
- `mountOptions: sec=krb5p` （身分識別與隱私保護的驗證與加密）

只指定一個 Kerberos 層級。如果您在參數清單中指定多個 Kerberos 加密層級、則只會使用第一個選項。如果您在儲存後端組態中指定的加密層級與您在儲存類別物件中指定的層級不同、則儲存類別物件會優先。

步驟

1. 使用以下範例建立 StorageClass Kubernetes 物件：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
parameters:
  backendType: "ontap-nas"
  storagePools: "ontapnas_pool"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: True
```

2. 建立儲存類別：

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. 確定已建立儲存類別：

```
kubectl get sc ontap-nas-sc
```

您應該會看到類似下列的輸出：

```
NAME            PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

配置 Volume

建立儲存後端和儲存類別之後、您現在可以配置 Volume 。請參閱以下說明 "資源配置"。

使用 **Azure NetApp Files** 磁碟區設定在線上 **Kerberos** 加密

您可以在託管叢集與單一 Azure NetApp Files 儲存後端或 Azure NetApp Files 儲存後端的虛擬集區之間的儲存流量上啟用 Kerberos 加密。

開始之前

- 確保您已在託管的 Red Hat OpenShift 叢集上啟用 Astra Control Provisioner 。請參閱 "啟用 Astra Control Provisioner" 以取得相關指示。
- 請確定您可以存取 `tridentctl` 公用程式：
- 請注意中的要求並遵循中的指示、以確保您已準備好 Azure NetApp Files 儲存後端進行 Kerberos 加密 "本文檔 Azure NetApp Files"。
- 請確定您使用 Kerberos 加密的任何 NFSv4 磁碟區都已正確設定。請參閱的「 NetApp NFSv4 網域組態」一節（第 13 頁） "NetApp NFSv4 增強與最佳實務指南"。

建立儲存後端

您可以建立包含 Kerberos 加密功能的 Azure NetApp Files 儲存後端組態。

關於這項工作

當您建立儲存後端組態檔案來設定 Kerberos 加密時、您可以加以定義、以便將其套用至下列兩種可能的層級之一：

- 使用的 * 儲存後端層級 * `spec.kerberos` 欄位
- 使用的 * 虛擬集區層級 * `spec.storage.kerberos` 欄位

當您在虛擬集區層級定義組態時、會使用儲存類別中的標籤來選取集區。

在任一層級、您都可以指定 Kerberos 加密的三個不同版本之一：

- `kerberos: sec=krb5` （驗證與加密）
- `kerberos: sec=krb5i` （身分識別保護的驗證與加密）
- `kerberos: sec=krb5p` （身分識別與隱私保護的驗證與加密）

步驟

1. 在託管叢集上、根據您需要定義儲存後端（儲存後端層級或虛擬集區層級）的位置、使用下列其中一個範例建立儲存後端組態檔案。以您環境的資訊取代括弧 <> 中的值：

儲存後端層級範例

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-anf-secret
```

虛擬集區層級範例

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
      kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-anf-secret
```

2. 使用您在上一個步驟中建立的組態檔來建立後端：

```
tridentctl create backend -f <backend-configuration-file>
```

如果後端建立失敗、表示後端組態有問題。您可以執行下列命令來檢視記錄、以判斷原因：

```
tridentctl logs
```

識別並修正組態檔的問題之後、您可以再次執行create命令。

建立儲存類別

您可以建立儲存類別、以使用 Kerberos 加密來配置磁碟區。

步驟

1. 使用以下範例建立 StorageClass Kubernetes 物件：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "nfs"
  selector: "type=encryption"
```

2. 建立儲存類別：

```
kubectl create -f sample-input/storage-class-anf-sc-nfs.yaml
```

3. 確定已建立儲存類別：

```
kubectl get sc anf-sc-nfs
```

您應該會看到類似下列的輸出：

```
NAME          PROVISIONER              AGE
anf-sc-nfs     csi.trident.netapp.io    15h
```

配置 **Volume**

建立儲存後端和儲存類別之後、您現在可以配置 Volume 。請參閱以下說明 "資源配置"。

使用快照恢復 Volume 資料

Astra Control Provisioner 使用從快照提供快速的原位磁碟區還原 `TridentActionSnapshotRestore` （TASR） CR 。此 CR 是 Kubernetes 的必要行動、在作業完成後不會持續存在。

Astra Control Provisioner 支援上的快照還原 `ontap-san`、 `ontap-san-economy`、 `ontap-nas`、 `ontap-nas-flexgroup`、 `azure-netapp-files`、 `gcp-cvs`和 `solidfire-san` 驅動程式：

開始之前

您必須擁有受約束的 PVC 和可用的 Volume 快照。

- 確認 PVC 狀態為「已連結」。

```
kubectl get pvc
```

- 驗證 Volume 快照是否已準備就緒可供使用。

```
kubectl get vs
```

步驟

1. 建立 TASR CR 。此範例會建立用於 PVC 的 CR `pvc1` 和 Volume Snapshot `pvc1-snapshot`。

```
cat tasr-pvc1-snapshot.yaml

apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: this-doesnt-matter
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. 套用 CR 以從快照還原。此範例從快照還原 `pvc1`。

```
kubectl create -f tasr-pvc1-snapshot.yaml

tridentactionsnapshotrestore.trident.netapp.io/this-doesnt-matter
created
```

結果

Astra Control Provisioner 會從快照還原資料。您可以驗證快照還原狀態。

```
kubectl get tasr -o yaml

apiVersion: trident.netapp.io/v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: this-doesnt-matter
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""
```

- 在大多數情況下、 Astra Control Provisioner 不會在發生故障時自動重試作業。您需要再次執行此作業。
- 不具備管理員存取權限的 Kubernetes 使用者可能必須獲得管理員的權限、才能在其應用程式命名空間中建立 TASR CR 。

使用 SnapMirror 複寫磁碟區

使用 Astra Control Provisioner 、您可以在一個叢集上的來源磁碟區和對等叢集上的目的地磁碟區之間建立鏡射關係、以便複寫資料以進行災難恢復。您可以使用命名的自訂資源定義（ CRD ）來執行下列作業：

- 建立磁碟區之間的鏡射關係（ PVCS ）
- 移除磁碟區之間的鏡射關係
- 中斷鏡射關係
- 在災難情況（容錯移轉）期間提升次要 Volume
- 在計畫性容錯移轉或移轉期間、將應用程式從叢集無損移轉至叢集

複寫先決條件

在您開始之前、請確定符合下列先決條件：

叢集 **ONTAP**

- * Astra Control Provisioner* ： Astra Control Provisioner 版本 23.10 或更新版本、或 "支援的 Astra Trident" 來源叢集和目的地 Kubernetes 叢集必須同時存在、並將 ONTAP 作為後端使用。
- * 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。請參閱 "SnapMirror授權概述ONTAP" 以取得更多資訊。

對等關係

- * 叢集與 SVM* ：必須對 ONTAP 儲存設備的後端進行對等處理。請參閱 "叢集與SVM對等概觀" 以取得更多資訊。

  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 Control Provisioner** 和 **SVM** ：對等的遠端 SVM 必須可供目的地叢集上的 Astra Control Provisioner 使用。

支援的驅動程式

- ONTAP NAS 和 ONTAP SAN 驅動程式支援 Volume 複寫。

建立鏡射 **PVC**

請遵循下列步驟、並使用 CRD 範例在主要和次要磁碟區之間建立鏡射關係。

步驟

1. 在主 Kubernetes 叢集上執行下列步驟：
   a. 使用建立 StorageClass 物件 `trident.netapp.io/replication: true` 參數。

      範例

      ```
      apiVersion: storage.k8s.io/v1
      kind: StorageClass
      metadata:
        name: csi-nas
      provisioner: csi.trident.netapp.io
      parameters:
        backendType: "ontap-nas"
        fsType: "nfs"
        trident.netapp.io/replication: "true"
      ```

   b. 使用先前建立的 StorageClass 建立 PVC 。

範例

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

c. 使用本機資訊建立 MirrorRelationship CR 。

範例

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
```

Astra Control Provisioner 會擷取磁碟區的內部資訊和磁碟區目前的資料保護（ DP ）狀態、然後填入 MirrorRelationship 的狀態欄位。

d. 取得 TridentMirrorRelationship CR 以取得 PVC 的內部名稱和 SVM 。

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
    localVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1
```

2. 在次 Kubernetes 叢集上執行下列步驟：
   a. 使用 trident.netapp.io/replication: true 參數建立 StorageClass 。

      範例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true
```

   b. 使用目的地和來源資訊建立 MirrorRelationship CR 。

      範例

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
  - localPVCName: csi-nas
    remoteVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

Astra Control Provisioner 將使用設定的關係原則名稱（或 ONTAP 的預設名稱）建立 SnapMirror 關係、並將其初始化。

c. 使用先前建立的 StorageClass 建立 PVC 、作為次要（ SnapMirror 目的地）。

範例

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
  - ReadWriteMany
resources:
  requests:
    storage: 1Gi
storageClassName: csi-nas
```

Astra Control Provisioner 會檢查 TridentMirrorRelationship CRD 、如果關係不存在、則無法建立 Volume 。如果存在這種關係、 Astra Control Provisioner 將確保新的 FlexVol 磁碟區放置在與 MirrorRelationship 中定義的遠端 SVM 對等的 SVM 上。

Volume 複寫狀態

Trident Mirror Relationship （ TMR ）是一種 CRD 、代表 PVC 之間複寫關係的一端。目的地 TMR 具有狀態、可告知 Astra Control Provisioner 所需的狀態。目的地 TMR 有下列狀態：

- * 建立 * ：本機 PVC 是鏡射關係的目的地 Volume 、這是新的關係。
- * 升級 * ：本機 PVC 為可讀寫且可掛載、目前無鏡射關係。
- * 重新建立 * ：本機 PVC 是鏡射關係的目的地 Volume 、先前也屬於該鏡射關係。
  - 如果目的地磁碟區與來源磁碟區有任何關係、則必須使用重新建立的狀態、因為它會覆寫目的地磁碟區內容。
  - 如果磁碟區先前未與來源建立關係、則重新建立的狀態將會失敗。

在非計畫性容錯移轉期間升級次要 PVC

在次 Kubernetes 叢集上執行下列步驟：

- 將 TridentMirrorRelationship 的 *spec.state* 欄位更新至 `promoted`。

在規劃的容錯移轉期間升級次要 PVC

在計畫性容錯移轉（移轉）期間、請執行下列步驟來升級次要 PVC ：

步驟

1. 在主要 Kubernetes 叢集上、建立 PVC 的快照、並等待快照建立完成。
2. 在主要 Kubernetes 叢集上、建立 SnapshotInfo CR 以取得內部詳細資料。

   範例

   ```
   kind: SnapshotInfo
   apiVersion: trident.netapp.io/v1
   metadata:
     name: csi-nas
   spec:
     snapshot-name: csi-nas-snapshot
   ```

3. 在次要 Kubernetes 叢集上、將 *TridentMirrorRelationationship _ CR* 的 *_spec.state* 欄位更新為 *updated* 、 *spec.promotedSnapshotHandle* 更新為快照的內部名稱。
4. 在次要 Kubernetes 叢集上、確認要升級的 TridentMirrorRelationship 狀態（ STATUS.STATUS 欄位）。

在容錯移轉後還原鏡射關係

還原鏡射關係之前、請先選擇要設為新主要的一面。

步驟

1. 在次要 Kubernetes 叢集上、確保已更新 TridentMirrorRelationship 上 *spec.remoteVolumeHandle* 欄位的值。
2. 在次要 Kubernetes 叢集上、將 TridentMirrorRelationship 的 _spec.mirror 欄位更新至 `reestablished`。

其他作業

Astra Control Provisioner 支援在主要和次要磁碟區上執行下列作業：

將主要 **PVC** 複製到新的次要 **PVC**

請確定您已擁有主要 PVC 和次要 PVC 。

步驟

1. 從已建立的次要（目的地）叢集刪除 PersistentVolume Claim 和 TridentMirrorRelationship CRD 。
2. 從主（來源）叢集刪除 TridentMirrorRelationship CRD 。
3. 在主要（來源）叢集上建立新的 TridentMirrorRelationship CRD 、以用於您要建立的新次要（目的地）PVC 。

調整鏡射、主要或次要 **PVC** 的大小

PVC 可以正常調整大小、如果資料量超過目前大小、 ONTAP 會自動擴充任何目的地 flevxols 。

從 **PVC** 移除複寫

若要移除複寫、請在目前的次要磁碟區上執行下列其中一項作業：

- 刪除次要 PVC 上的 MirrorRelationship 。這會中斷複寫關係。
- 或者、將 spec.state 欄位更新為 *updated* 。

刪除 **PVC** （先前已鏡射）

Astra Control Provisioner 會檢查複寫的 PVCS 、並在嘗試刪除磁碟區之前先釋放複寫關係。

刪除 **TMR**

在鏡射關係的一側刪除 TMR 會導致其餘 TMR 在 Astra Control Provisioner 完成刪除之前轉換至 升 遷狀態。如果選取要刪除的 *TMR* 已處於 _ 升級 _ 狀態、則沒有現有的鏡射關係、 *TMR* 將會移除、 *Astra Control Provisioner* 會將本機 *PVC* 升級為 *_ReadWrite* 。此刪除作業會在 ONTAP 中針對本機磁碟區釋出 SnapMirror 中繼資料。如果此磁碟區在未來的鏡射關係中使用、則在建立新的鏡射關係時、它必須使用具有 _ 建立 _ 磁碟區複寫狀態的新 TMR 。

當 ONTAP 連線時、請更新鏡射關係

建立鏡射關係之後、可以隨時更新它們。您可以使用 `state: promoted` 或 `state: reestablished` 更新關聯的欄位。
將目的地 Volume 升級為一般 ReadWrite Volume 時、您可以使用 *promotedSnapshotHandle* 來指定特定快照、將目前的 Volume 還原至。

當 ONTAP 離線時更新鏡射關係

您可以使用 CRD 來執行 SnapMirror 更新、而無需 Astra Control 直接連線至 ONTAP 叢集。請參閱下列 TridentActionMirrorUpdate 範例格式：

範例

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

`status.state` 反映 TridentActionMirrorUpdate CRD 的狀態。它可以取自 *sued* 、 *in progress* 或 *Failed* 的值。