恢復應用程式
Trident

NetApp
March 04, 2026

This PDF was generated from https://docs.netapp.com/zh-tw/trident-2506/trident-protect/trident-protect-restore-apps.html on March 04, 2026. Always check docs.netapp.com for the latest.

目錄

恢復應用程式

使用Trident Protect 恢復應用程式

您可以使用Trident Protect 從快照或備份中還原您的應用程式。將應用程式還原到同一群集時，從現有快照恢復速度會更快。

- 還原應用程式時，為該應用程式配置的所有執行鉤子都會隨應用程式一起復原。如果存在恢復後執行鉤子，它將作為恢復操作的一部分自動運行。
- 支援從備份還原到不同的命名空間或還原到原始命名空間（適用於 qtree 磁碟區）。但是，對於 qtree 卷，不支援從快照還原到不同的命名空間或還原到原始命名空間。
- 您可以使用進階設定來自訂恢復操作。欲了解更多信息，請參閱 "使用進階Trident Protect 恢復設定"。

從備份還原到不同的命名空間

當您使用 BackupRestore CR 將備份還原到不同的命名空間時， Trident Protect 會在新的命名空間中還原應用程序，並為還原的應用程式建立一個應用程式 CR。為了保護已還原的應用程序，可以建立按需備份或快照，或製定保護計劃。

將備份還原到具有現有資源的不同命名空間不會變更與備份中資源同名的任何資源。若要還原備份中的所有資源，請刪除並重新建立目標命名空間，或將備份還原到新的命名空間。

開始之前

確保 AWS 會話令牌的過期時間足以滿足任何長時間運行的 S3 復原操作。如果在恢復操作期間令牌過期，則操作可能會失敗。

- 請參閱 "AWS API 文件"有關檢查當前會話令牌過期時間的詳細資訊。
- 請參閱 "AWS IAM 文件"有關 AWS 資源憑證的詳細資訊。

當您使用 Kopia 作為資料移動器還原備份時，您可以選擇在 CR 中指定註解或使用 CLI 來控制 Kopia 使用的暫存的行為。請參閱 "科皮亞文檔"有關您可以配置的選項的詳細資訊。使用 `tridentctl-protect create --help`有關使用Trident Protect CLI 指定註釋的更多信息，請參閱命令。

使用 **CR**

步驟

1. 建立自訂資源 (CR) 檔案並將其命名為 `trident-protect-backup-restore-cr.yaml`。
2. 在您建立的文件中，配置以下屬性：
   - **metadata.name**: (必填) 此自訂資源的名稱；請為您的環境選擇一個唯一且有意義的名稱。
   - **spec.appArchivePath**: AppVault 內儲存備份內容的路徑。您可以使用以下命令尋找此路徑：

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (必要) 儲存備份內容的 AppVault 的名稱。
   - **spec.namespaceMapping**：恢復作業的來源命名空間到目標命名空間的對應。代替 `my-source-namespace`和 `my-destination-namespace`利用來自周圍環境的資訊。

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: my-cr-name
       namespace: my-destination-namespace
     spec:
       appArchivePath: my-backup-path
       appVaultRef: appvault-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. （可選）如果您只需要選擇應用程式中的某些資源進行恢復，請新增篩選條件，以包含或排除帶有特定標籤的資源：

   > Trident Protect 會自動選擇一些資源，因為它們與您選擇的資源有關聯。例如，如果您選擇持久性磁碟區宣告資源且它有一個關聯的 pod， Trident Protect 也會還原關聯的 pod。

   - **resourceFilter.resourceSelectionCriteria**：（篩選時必備）使用 `Include`或者 `Exclude`包含或排除 resourceMatchers 中定義的資源。新增以下 resourceMatchers 參數以定義要包含或排除的資源：
     - **resourceFilter.resourceMatchers**：resourceMatcher 物件陣列。如果在該數組中定義多個元素，則它們之間按 OR 運算匹配，每個元素內的字段（組、種類、版本）之間按 AND 運算匹配。
       - **resourceMatchers[].group**: (可選) 要篩選的資源群組。
       - **resourceMatchers[].kind**: (可選) 要篩選的資源類型。

- **resourceMatchers[].version**: (可選) 要篩選的資源版本。
- **resourceMatchers[].names**: (可選) 要過濾的資源的 Kubernetes 元資料.name 欄位中的名稱。
- **resourceMatchers[].namespaces**: (可選) 要篩選的資源的 Kubernetes 元資料.name 欄位中的命名空間。
- **resourceMatchers[].labelSelectors**: (可選) 資源的 Kubernetes 元資料.name 欄位中的標籤選擇器字串，如在下列位置定義： "Kubernetes 文檔" 。例如： `"trident.netapp.io/os=linux"` 。

  例如：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 填寫完後 `trident-protect-backup-restore-cr.yaml`將檔案的值正確後，套用 CR：

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

使用 CLI

步驟

1. 將備份還原到不同的命名空間，並將括號中的值替換為您環境中的資訊。這 `namespace-mapping`此參數使用冒號分隔的命名空間，將來源命名空間對應到正確的目標命名空間，格式如下： `source1:dest1,source2:dest2` 。例如：

```
tridentctl-protect create backuprestore <my_restore_name> \
--backup <backup_namespace>/<backup_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

從備份還原到原始命名空間

您可以隨時將備份還原到原始命名空間。

開始之前

確保 AWS 會話令牌的過期時間足以滿足任何長時間運行的 S3 復原操作。如果在恢復操作期間令牌過期，則操作可能會失敗。

- 請參閱 "AWS API 文件"有關檢查當前會話令牌過期時間的詳細資訊。
- 請參閱 "AWS IAM 文件"有關 AWS 資源憑證的詳細資訊。

當您使用 Kopia 作為資料移動器還原備份時，您可以選擇在 CR 中指定註解或使用 CLI 來控制 Kopia 使用的暫存的行為。請參閱 "科皮亞文檔"有關您可以配置的選項的詳細資訊。使用 `tridentctl-protect create --help`有關使用Trident Protect CLI 指定註釋的更多信息，請參閱命令。

使用 **CR**

步驟

1. 建立自訂資源 (CR) 檔案並將其命名為 `trident-protect-backup-ipr-cr.yaml`。
2. 在您建立的文件中，配置以下屬性：
   - **metadata.name**: (必填) 此自訂資源的名稱；請為您的環境選擇一個唯一且有意義的名稱。
   - **spec.appArchivePath**: AppVault 內儲存備份內容的路徑。您可以使用以下命令尋找此路徑：

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (必要) 儲存備份內容的 AppVault 的名稱。

     例如：

```
---
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appArchivePath: my-backup-path
  appVaultRef: appvault-name
```

3. （可選）如果您只需要選擇應用程式中的某些資源進行恢復，請新增篩選條件，以包含或排除帶有特定標籤的資源：

   Trident Protect 會自動選擇一些資源，因為它們與您選擇的資源有關聯。例如，如果您選擇持久性磁碟區宣告資源且它有一個關聯的 pod， Trident Protect 也會還原關聯的 pod。

   - **resourceFilter.resourceSelectionCriteria**：（篩選時必備）使用 `Include`或者 `Exclude`包含或排除 resourceMatchers 中定義的資源。新增以下 resourceMatchers 參數以定義要包含或排除的資源：
     - **resourceFilter.resourceMatchers**：resourceMatcher 物件陣列。如果在該數組中定義多個元素，則它們之間按 OR 運算匹配，每個元素內的字段（組、種類、版本）之間按 AND 運算匹配。
       - **resourceMatchers[].group**: (可選) 要篩選的資源群組。
       - **resourceMatchers[].kind**: (可選) 要篩選的資源類型。
       - **resourceMatchers[].version**: (可選) 要篩選的資源版本。
       - **resourceMatchers[].names**: (可選) 要過濾的資源的 Kubernetes 元資料.name 欄位中的名

稱。

- **resourceMatchers[].namespaces**: (可選) 要篩選的資源的 Kubernetes 元資料.name 欄位中的命名空間。
- **resourceMatchers[].labelSelectors**: (可選) 資源的 Kubernetes 元資料.name 欄位中的標籤選擇器字串，如在下列位置定義： "Kubernetes 文檔" 。例如： `"trident.netapp.io/os=linux"` 。

  例如：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 填寫完後 `trident-protect-backup-ipr-cr.yaml`將檔案的值正確後，套用 CR：

```
kubectl apply -f trident-protect-backup-ipr-cr.yaml
```

使用 **CLI**

步驟

1. 將備份還原到原始命名空間，並將括號中的值替換為您環境中的資訊。這 `backup`參數使用命名空間和備份名稱，格式如下 `<namespace>/<name>`。例如：

```
tridentctl-protect create backupinplacerestore <my_restore_name> \
--backup <namespace/backup_to_restore> \
-n <application_namespace>
```

從備份還原到不同的集群

如果原始叢集出現問題，您可以將備份還原到其他叢集。

> 當您使用 Kopia 作為資料移動器還原備份時，您可以選擇在 CR 中指定註解或使用 CLI 來控制 Kopia 使用的暫存的行為。請參閱 "科皮亞文檔"有關您可以配置的選項的詳細資訊。使用 `tridentctl-protect create --help`有關使用Trident Protect CLI 指定註釋的更多信息，請參閱命令。

開始之前

確保滿足以下先決條件：

- 目標叢集已安裝Trident Protect。
- 目標叢集可以存取與來源叢集相同的 AppVault 的儲存桶路徑，備份就儲存在該儲存桶中。
- 執行 AppVault CR 時，請確保本機環境可以連接到 AppVault CR 中定義的物件儲存桶。 `tridentctl-protect get appvaultcontent`命令。如果網路限制阻止訪問，請改為從目標叢集上的 pod 內執行Trident Protect CLI。
- 確保 AWS 會話令牌的過期時間足以滿足任何長時間運行的復原操作。如果在恢復操作期間令牌過期，則操作可能會失敗。
  - 請參閱 "AWS API 文件"有關檢查當前會話令牌過期時間的詳細資訊。
  - 請參閱 "AWS 文件"有關 AWS 資源憑證的詳細資訊。

步驟

1. 使用Trident Protect CLI 外掛程式檢查目標叢集上 AppVault CR 的可用性：

```
tridentctl-protect get appvault --context <destination_cluster_name>
```

> 確保用於應用程式還原的命名空間存在於目標叢集上。

2. 查看目標叢集中可用 AppVault 的備份內容：

```
tridentctl-protect get appvaultcontent <appvault_name> \
--show-resources backup \
--show-paths \
--context <destination_cluster_name>
```

執行此命令將顯示 AppVault 中可用的備份，包括其來源叢集、對應的應用程式名稱、時間戳記和歸檔路徑。

範例輸出：

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   CLUSTER   |    APP    |  TYPE  |      NAME       |        TIMESTAMP
|    PATH     |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
| production1 | wordpress | backup | wordpress-bkup-1| 2024-10-30
08:37:40 (UTC)| backuppath1 |
| production1 | wordpress | backup | wordpress-bkup-2| 2024-10-30
08:37:40 (UTC)| backuppath2 |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
```

3. 使用 AppVault 名稱和歸檔路徑將應用程式還原到目標叢集：

使用 **CR**

1. 建立自訂資源 (CR) 檔案並將其命名為 `trident-protect-backup-restore-cr.yaml`。
2. 在您建立的文件中，配置以下屬性：
   - **metadata.name**: (必填) 此自訂資源的名稱；請為您的環境選擇一個唯一且有意義的名稱。
   - **spec.appVaultRef**: (必要) 儲存備份內容的 AppVault 的名稱。
   - **spec.appArchivePath**: AppVault 內儲存備份內容的路徑。您可以使用以下命令尋找此路徑：

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

     如果 BackupRestore CR 不可用，您可以使用步驟 2 中提到的指令來查看備份內容。

   - **spec.namespaceMapping**：恢復作業的來源命名空間到目標命名空間的對應。代替 `my-source-namespace`和 `my-destination-namespace`利用來自周圍環境的資訊。

     例如：

```
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  name: my-cr-name
  namespace: my-destination-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-backup-path
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

3. 填寫完後 `trident-protect-backup-restore-cr.yaml`將檔案的值正確後，套用 CR：

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

使用 **CLI**

1. 使用以下命令恢復應用程序，將括號中的值替換為您環境中的資訊。命名空間映射參數使用冒號分隔的命名空間，將來源命名空間對應到正確的目標命名空間，格式為 source1:dest1,source2:dest2。例如：

```
tridentctl-protect create backuprestore <restore_name> \
--namespace-mapping <source_to_destination_namespace_mapping> \
--appvault <appvault_name> \
--path <backup_path> \
--context <destination_cluster_name> \
-n <application_namespace>
```

從快照還原到不同的命名空間

您可以使用自訂資源 (CR) 檔案從快照還原數據，還原到不同的命名空間或原始來源命名空間。當您使用 SnapshotRestore CR 將快照還原到不同的命名空間時， Trident Protect 會在新的命名空間中還原應用程序，並為還原的應用程式建立應用程式 CR。為了保護已還原的應用程序，可以建立按需備份或快照，或製定保護計劃。

> SnapshotRestore 支持 `spec.storageClassMapping`屬性，但僅當來源儲存類別和目標儲存類別使用相同的儲存後端時才有效。如果您嘗試恢復到 `StorageClass`如果使用不同的儲存後端，則復原操作將會失敗。

開始之前

確保 AWS 會話令牌的過期時間足以滿足任何長時間運行的 S3 復原操作。如果在恢復操作期間令牌過期，則操作可能會失敗。

- 請參閱 "AWS API 文件"有關檢查當前會話令牌過期時間的詳細資訊。
- 請參閱 "AWS IAM 文件"有關 AWS 資源憑證的詳細資訊。

使用 **CR**

步驟

1. 建立自訂資源 (CR) 檔案並將其命名為 `trident-protect-snapshot-restore-cr.yaml`。
2. 在您建立的文件中，配置以下屬性：
   - **metadata.name**: (必填) 此自訂資源的名稱；請為您的環境選擇一個唯一且有意義的名稱。
   - **spec.appVaultRef**: (必要) 儲存快照內容的 AppVault 的名稱。
   - **spec.appArchivePath**: AppVault 內儲存快照內容的路徑。您可以使用以下命令尋找此路徑：

     ```
     kubectl get snapshots <SNAPHOT_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.namespaceMapping**：恢復作業的來源命名空間到目標命名空間的對應。代替 `my-source-namespace`和 `my-destination-namespace`利用來自周圍環境的資訊。

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: my-cr-name
       namespace: my-app-namespace
     spec:
       appVaultRef: appvault-name
       appArchivePath: my-snapshot-path
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. （可選）如果您只需要選擇應用程式中的某些資源進行恢復，請新增篩選條件，以包含或排除帶有特定標籤的資源：

   > Trident Protect 會自動選擇一些資源，因為它們與您選擇的資源有關聯。例如，如果您選擇持久性磁碟區宣告資源且它有一個關聯的 pod， Trident Protect 也會還原關聯的 pod。

   - **resourceFilter.resourceSelectionCriteria**：（篩選時必備）使用 `Include`或者 `Exclude`包含或排除 resourceMatchers 中定義的資源。新增以下 resourceMatchers 參數以定義要包含或排除的資源：
     - **resourceFilter.resourceMatchers**：resourceMatcher 物件陣列。如果在該數組中定義多個元素，則它們之間按 OR 運算匹配，每個元素內的字段（組、種類、版本）之間按 AND 運算匹配。
       - **resourceMatchers[].group**: (可選) 要篩選的資源群組。
       - **resourceMatchers[].kind**: (可選) 要篩選的資源類型。

- **resourceMatchers[].version**: (可選) 要篩選的資源版本。
- **resourceMatchers[].names**: (可選) 要過濾的資源的 Kubernetes 元資料.name 欄位中的名稱。
- **resourceMatchers[].namespaces**: (可選) 要篩選的資源的 Kubernetes 元資料.name 欄位中的命名空間。
- **resourceMatchers[].labelSelectors**: (可選) 資源的 Kubernetes 元資料.name 欄位中的標籤選擇器字串，如在下列位置定義： "Kubernetes 文檔" 。例如： `"trident.netapp.io/os=linux"` 。

  例如：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 填寫完後 `trident-protect-snapshot-restore-cr.yaml`將檔案的值正確後，套用 CR：

```
kubectl apply -f trident-protect-snapshot-restore-cr.yaml
```

使用 CLI

步驟

1. 將快照還原到不同的命名空間，並將括號中的值替換為您環境中的資訊。

   - 這 `snapshot`參數使用命名空間和快照名稱，格式如下 `<namespace>/<name>`。
   - 這 `namespace-mapping`此參數使用冒號分隔的命名空間，將來源命名空間對應到正確的目標命名空間，格式如下： `source1:dest1,source2:dest2` 。

     例如：

```
tridentctl-protect create snapshotrestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

從快照還原到原始命名空間

您可以隨時將快照還原到原始命名空間。

如果您的應用程式使用多個命名空間，且這些命名空間中存在同名的PVC，則快照還原操作（無論是就地還原或還原到新命名空間）都將無法正常運作。所有還原的磁碟區將包含來自單一命名空間而不是每個命名空間對應的正確資料。請使用備份還原而不是快照還原，或升級至包含修復此問題的版本。

開始之前

確保 AWS 會話令牌的過期時間足以滿足任何長時間運行的 S3 復原操作。如果在恢復操作期間令牌過期，則操作可能會失敗。

- 請參閱 "AWS API 文件"有關檢查當前會話令牌過期時間的詳細資訊。
- 請參閱 "AWS IAM 文件"有關 AWS 資源憑證的詳細資訊。

使用 **CR**

步驟

1. 建立自訂資源 (CR) 檔案並將其命名為 `trident-protect-snapshot-ipr-cr.yaml`。
2. 在您建立的文件中，配置以下屬性：
   - **metadata.name**: (必填) 此自訂資源的名稱；請為您的環境選擇一個唯一且有意義的名稱。
   - **spec.appVaultRef**: (必要) 儲存快照內容的 AppVault 的名稱。
   - **spec.appArchivePath**: AppVault 內儲存快照內容的路徑。您可以使用以下命令尋找此路徑：

```
kubectl get snapshots <SNAPSHOT_NAME> -n my-app-namespace -o
jsonpath='{.status.appArchivePath}'
```

```
---
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appVaultRef: appvault-name
    appArchivePath: my-snapshot-path
```

3. （可選）如果您只需要選擇應用程式中的某些資源進行恢復，請新增篩選條件，以包含或排除帶有特定標籤的資源：

> Trident Protect 會自動選擇一些資源，因為它們與您選擇的資源有關聯。例如，如果您選擇持久性磁碟區宣告資源且它有一個關聯的 pod， Trident Protect 也會還原關聯的 pod。

   - **resourceFilter.resourceSelectionCriteria**：（篩選時必備）使用 `Include`或者 `Exclude`包含或排除 resourceMatchers 中定義的資源。新增以下 resourceMatchers 參數以定義要包含或排除的資源：
     - **resourceFilter.resourceMatchers**：resourceMatcher 物件陣列。如果在該數組中定義多個元素，則它們之間按 OR 運算匹配，每個元素內的字段（組、種類、版本）之間按 AND 運算匹配。
       - **resourceMatchers[].group**: (可選) 要篩選的資源群組。
       - **resourceMatchers[].kind**: (可選) 要篩選的資源類型。
       - **resourceMatchers[].version**: (可選) 要篩選的資源版本。
       - **resourceMatchers[].names**: (可選) 要過濾的資源的 Kubernetes 元資料.name 欄位中的名稱。
       - **resourceMatchers[].namespaces**: (可選) 要篩選的資源的 Kubernetes 元資料.name 欄位中的命名空間。

- **resourceMatchers[].labelSelectors**: (可選) 資源的 Kubernetes 元資料.name 欄位中的標籤選擇器字串，如在下列位置定義： "Kubernetes 文檔" 。例如： `"trident.netapp.io/os=linux"` 。

  例如：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 填寫完後 `trident-protect-snapshot-ipr-cr.yaml`將檔案的值正確後，套用 CR：

```
kubectl apply -f trident-protect-snapshot-ipr-cr.yaml
```

使用 CLI

步驟

1. 將快照還原到原始命名空間，並將括號中的值替換為您環境中的資訊。例如：

```
tridentctl-protect create snapshotinplacerestore <my_restore_name> \
--snapshot <snapshot_to_restore> \
-n <application_namespace>
```

檢查還原操作的狀態

您可以使用命令列來檢查正在進行、已完成或已失敗的還原作業的狀態。

步驟

1. 使用以下命令檢索恢復操作的狀態，將方括號中的值替換為您環境中的資訊：

```
kubectl get backuprestore -n <namespace_name> <my_restore_cr_name> -o
jsonpath='{.status}'
```

使用進階Trident Protect 恢復設定

您可以使用進階設定（例如註解、命名空間設定和儲存選項）自訂復原操作，以滿足您的特定要求。

復原和故障轉移操作期間的命名空間註釋和標籤

在復原和故障轉移操作期間，目標命名空間中的標籤和註釋將與來源命名空間中的標籤和註釋相符。來源命名空間中不存在的目標命名空間中的標籤或註釋將被添加，並且任何已存在的標籤或註釋都將被覆蓋以匹配來源命名空間中的值。僅存在於目標命名空間中的標籤或註解保持不變。

> 如果您使用 Red Hat OpenShift，請務必注意命名空間註解在 OpenShift 環境中的重要角色。命名空間註解可確保復原的 pod 遵守 OpenShift 安全性情境約束 (SCC) 定義的適當權限和安全性配置，並且可以存取磁碟區而不會出現權限問題。欲了解更多信息，請參閱 "OpenShift 安全上下文約束文檔"。

您可以透過設定 Kubernetes 環境變數來防止目標命名空間中的特定註解被覆蓋。`RESTORE_SKIP_NAMESPACE_ANNOTATIONS`在執行復原或故障轉移操作之前。例如：

```
helm upgrade trident-protect --set
restoreSkipNamespaceAnnotations=<annotation_key_to_skip_1>,<annotation_key
_to_skip_2> --reuse-values
```

> 執行復原或故障轉移操作時，任何命名空間註解和標籤都將生效。`restoreSkipNamespaceAnnotations` 和 `restoreSkipNamespaceLabels` 不參與恢復或故障轉移操作。確保在初始 Helm 安裝期間配置這些設定。欲了解更多信息，請參閱 "配置AutoSupport和命名空間過濾選項"。

如果您使用 Helm 安裝了來源應用程序， `--create-namespace`國旗，給予特殊待遇 `name`標籤鍵。在復原或故障轉移過程中， Trident Protect 會將此標籤複製到目標命名空間，但如果來源命名空間的值與來源命名空間的值匹配，則會將值更新為目標命名空間的值。如果此值與來源命名空間不匹配，則會將其複製到目標命名空間，而不做任何變更。

例子

以下範例展示了來源命名空間和目標命名空間，每個命名空間都有不同的註解和標籤。您可以查看操作前後目標命名空間的狀態，以及目標命名空間中的註解和標籤是如何組合或覆蓋的。

在恢復或故障轉移操作之前

下表說明了復原或故障轉移操作之前範例來源命名空間和目標命名空間的狀態：

命名空間	註解	標籤
命名空間 ns-1（來源）	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• 環境=生產 • 合規性=HIPAA • 名稱=ns-1
命名空間 ns-2（目標）	• annotation.one/key: "true" • annotation.three/key: "false"	• 角色=資料庫

恢復操作後

下表說明了復原或故障轉移作業後範例目標命名空間的狀態。有些按鍵已被添加，有些按鍵已被覆蓋，並且 `name`標籤已更新，以符合目標命名空間：

命名空間	註解	標籤
命名空間 ns-2（目標）	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• 名稱=ns-2 • 合規性=HIPAA • 環境=生產 • 角色=資料庫

支援的字段

本節介紹可用於恢復操作的其他欄位。

儲存類別映射

這 `spec.storageClassMapping`屬性定義了從來源應用程式中存在的儲存類別到目標叢集上新儲存類別的對應。您可以在具有不同儲存類別的叢集之間移轉應用程式時或變更 BackupRestore 作業的儲存後端時使用此功能。

例：

```
storageClassMapping:
  - destination: "destinationStorageClass1"
    source: "sourceStorageClass1"
  - destination: "destinationStorageClass2"
    source: "sourceStorageClass2"
```

支持的註釋

本節列出了系統中用於配置各種行為的支援註解。如果使用者沒有明確設定註釋，系統將使用預設值。

註解	類型	描述	預設值
protect.trident.netapp.io/data-mover-timeout-sec	細繩	資料移動器操作允許停止的最長時間（以秒為單位）。	“300”
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	細繩	Kopia 內容快取的最大大小限制（以兆位元組為單位）。	“1000”

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。