還原應用程式
Trident

NetApp
February 02, 2026

目錄

還原應用程式

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。

- 當您還原應用程式時，為應用程式設定的所有執行掛鉤都會隨應用程式一起還原。如果存在還原後執行掛鉤，則會在還原作業中自動執行。
- 對於 qtree 卷，支援從備份還原到其他命名空間或原始命名空間。但是，對於 qtree 卷，不支援從快照還原到其他命名空間或原始命名空間。
- 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"。

從備份還原至不同的命名空間

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。

- 將備份還原至具有現有資源的不同命名空間，並不會改變任何與備份中共用名稱的資源。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，或將備份還原至新的命名空間。
- 使用 CR 還原到新命名空間時，必須先手動建立目標命名空間，然後再套用 CR。Trident Protect 僅在使用 CLI 時才會自動建立命名空間。

開始之前

確保 AWS 工作階段權杖到期時間足以執行任何長時間執行的 S3 還原作業。如果 Token 在還原作業期間過期，作業可能會失敗。

- 如需檢查目前工作階段權杖到期時間的詳細資訊，請參閱 "AWS API 文件"。
- 如需 AWS 資源認證的詳細資訊，請參閱 "AWS IAM 文件"。

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 "Kopia 文件"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 ）檔案並命名為 `trident-protect-backup-restore-cr.yaml`。
2. 在您建立的檔案中，設定下列屬性：
   - * metadata.name*: （ _required ）此自訂資源的名稱；為您的環境選擇唯一且合理的名稱。
   - **spec.appArchivePath** ：儲存備份內容的 AppVault 內部路徑。您可以使用下列命令來尋找此路徑：

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

   - **spec.appVaultRef** ：（ _required _ ）儲存備份內容的 AppVault 名稱。
   - **spec.namespaceMapping**: 將還原作業的來源命名空間對應至目的地命名空間。以環境中的資訊取代 `my-source-namespace`和 `my-destination-namespace`。

     ```
     ---
     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. （ *Optional* ）如果您只需要選取應用程式的某些資源來還原，請新增篩選功能，以包含或排除標記有特定標籤的資源：

   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** ：（篩選所需）使用 `Include`或包含或 `Exclude`排除在 resourceMatchers 中定義的資源。新增下列資源配置工具參數、以定義要納入或排除的資源：
     - **resourceFilter.resourceMatchers** ：一組 resourceMatcher 物件。如果您在此陣列中定義多個元素，它們會比對為 OR 作業，而每個元素（群組，種類，版本）內的欄位會比對為 AND 作業。
       - **resourceMatchers[].group** ：（ *Optional* ）要篩選的資源群組。
       - **resourceMatchers[].cher** ：（ *Optional* ）要篩選的資源種類。

- **resourceMatchers[].version** ：（ *Optional* ）要篩選的資源版本。
- 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）名稱。
- 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）命名空間。
- 資源的 Kubernetes metadata.name 欄位中的 *resourceMatchers[].labelSelectors * ：（ *Optional* ） Label 選取器字串，如中所定義 "Kubernetes文件"。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. 在您以正確的值填入檔案之後 `trident-protect-backup-restore-cr.yaml` 、請套用 CR ：

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

使用CLI

步驟

1. 將備份還原至不同的命名空間，以環境中的資訊取代括弧中的值。此 `namespace-mapping`引數使用以冒號分隔的命名空間，以格式將來源命名空間對應至正確的目的地命名空間 `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>
```

從備份還原至原始命名空間

您可以隨時將備份還原至原始命名空間。

開始之前

確保 AWS 工作階段權杖到期時間足以執行任何長時間執行的 S3 還原作業。如果 Token 在還原作業期間過期，作業可能會失敗。

- 如需檢查目前工作階段權杖到期時間的詳細資訊，請參閱 "AWS API 文件"。
- 如需 AWS 資源認證的詳細資訊，請參閱 "AWS IAM 文件"。

> 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 "Kopia 文件"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 ）檔案並命名為 `trident-protect-backup-ipr-cr.yaml` 。
2. 在您建立的檔案中，設定下列屬性：
   - * metadata.name*: （ _required ）此自訂資源的名稱；為您的環境選擇唯一且合理的名稱。
   - **spec.appArchivePath** ：儲存備份內容的 AppVault 內部路徑。您可以使用下列命令來尋找此路徑：

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

   - **spec.appVaultRef** ：（ _required _ ）儲存備份內容的 AppVault 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. （ *Optional* ）如果您只需要選取應用程式的某些資源來還原，請新增篩選功能，以包含或排除標記有特定標籤的資源：

   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** ：（篩選所需）使用 `Include`或包含或 `Exclude`排除在 resourceMatchers 中定義的資源。新增下列資源配置工具參數、以定義要納入或排除的資源：
     - **resourceFilter.resourceMatchers** ：一組 resourceMatcher 物件。如果您在此陣列中定義多個元素，它們會比對為 OR 作業，而每個元素（群組，種類，版本）內的欄位會比對為 AND 作業。
       - **resourceMatchers[].group** ：（ *Optional* ）要篩選的資源群組。
       - **resourceMatchers[].cher** ：（ *Optional* ）要篩選的資源種類。
       - **resourceMatchers[].version** ：（ *Optional* ）要篩選的資源版本。
       - 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（

Optional ）名稱。

- 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）命名空間。
- 資源的 Kubernetes metadata.name 欄位中的 *resourceMatchers[].labelSelectors * ：（ *Optional* ） Label 選取器字串，如中所定義 "Kubernetes文件"。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. 在您以正確的值填入檔案之後 `trident-protect-backup-ipr-cr.yaml` 、請套用 CR ：

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

使用CLI

步驟

1. 將備份還原至原始命名空間，以環境中的資訊取代括弧中的值。 `backup`引數使用的名稱空間和備份名稱格式為 `<namespace>/<name>`。例如：

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

從備份還原至不同的叢集

如果原始叢集發生問題，您可以將備份還原至不同的叢集。

- 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 "Kopia 文件"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 還原到新命名空間時，必須先手動建立目標命名空間，然後再套用 CR。Trident Protect 僅在使用 CLI 時才會自動建立命名空間。

開始之前

確保符合下列先決條件：

- 目標叢集已安裝Trident Protect。
- 目的地叢集可存取與儲存備份的來源叢集相同 AppVault 的儲存區路徑。
- 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 工作階段權杖到期時間足以執行任何長時間執行的還原作業。如果 Token 在還原作業期間過期，作業可能會失敗。
  - 如需檢查目前工作階段權杖到期時間的詳細資訊，請參閱 "AWS API 文件"。
  - 如需 AWS 資源認證的詳細資訊，請參閱 "AWS 文件"。

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>
```

確保目的地叢集上存在用於應用程式還原的命名空間。

2. 從目的地叢集檢視可用 AppVault 的備份內容：

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

執行此命令會顯示 AppVault 中的可用備份，包括其原始叢集，對應的應用程式名稱，時間戳記和歸檔路徑。

- 輸出範例： *

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   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至目的地叢集：

使用 **CR**

1. 建立自訂資源（ CR ）檔案並命名為 `trident-protect-backup-restore-cr.yaml` 。
2. 在您建立的檔案中，設定下列屬性：
   - * metadata.name*: （ _required ）此自訂資源的名稱；為您的環境選擇唯一且合理的名稱。
   - **spec.appVaultRef** ：（ _required _ ）儲存備份內容的 AppVault 名稱。
   - **spec.appArchivePath** ：儲存備份內容的 AppVault 內部路徑。您可以使用下列命令來尋找此路徑：

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

     如果無法使用 BackupRestore CR ，您可以使用步驟 2 所述的命令來檢視備份內容。

   - **spec.namespaceMapping**: 將還原作業的來源命名空間對應至目的地命名空間。以環境中的資訊取代 `my-source-namespace`和 `my-destination-namespace` 。

     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. 在您以正確的值填入檔案之後 `trident-protect-backup-restore-cr.yaml` 、請套用 CR ：

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

使用**CLI**

1. 使用下列命令還原應用程式，將方括號中的值取代為您環境中的資訊。命名空間對應引數使用以冒號分隔的命名空間，將來源命名空間對應到正確的目的地命名空間，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>
```

從快照還原至不同的命名空間

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，但僅當來源和目標儲存類別使用相同的儲存後端。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。
- 使用 CR 還原到新命名空間時，必須先手動建立目標命名空間，然後再套用 CR。Trident Protect 僅在使用 CLI 時才會自動建立命名空間。

開始之前

確保 AWS 工作階段權杖到期時間足以執行任何長時間執行的 S3 還原作業。如果 Token 在還原作業期間過期，作業可能會失敗。

- 如需檢查目前工作階段權杖到期時間的詳細資訊，請參閱 "AWS API 文件"。
- 如需 AWS 資源認證的詳細資訊，請參閱 "AWS IAM 文件"。

使用 **CR**

步驟

1. 建立自訂資源（ CR ）檔案並命名為 `trident-protect-snapshot-restore-cr.yaml`。
2. 在您建立的檔案中，設定下列屬性：
   - * metadata.name*: （ _required ）此自訂資源的名稱；為您的環境選擇唯一且合理的名稱。
   - **spec.appVaultRef** ：（ _required _ ）儲存快照內容的 AppVault 名稱。
   - **spec.appArchivePath** ：在 AppVault 中儲存快照內容的路徑。您可以使用下列命令來尋找此路徑：

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

   - **spec.namespaceMapping**: 將還原作業的來源命名空間對應至目的地命名空間。以環境中的資訊取代 `my-source-namespace`和 `my-destination-namespace`。

     ```
     ---
     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. （ *Optional* ）如果您只需要選取應用程式的某些資源來還原，請新增篩選功能，以包含或排除標記有特定標籤的資源：

   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** ：（篩選所需）使用 `Include`或包含或 `Exclude`排除在 resourceMatchers 中定義的資源。新增下列資源配置工具參數、以定義要納入或排除的資源：
     - **resourceFilter.resourceMatchers** ：一組 resourceMatcher 物件。如果您在此陣列中定義多個元素，它們會比對為 OR 作業，而每個元素（群組，種類，版本）內的欄位會比對為 AND 作業。
       - **resourceMatchers[].group** ：（ *Optional* ）要篩選的資源群組。
       - **resourceMatchers[].cher** ：（ *Optional* ）要篩選的資源種類。

- **resourceMatchers[].version** ：（ *Optional* ）要篩選的資源版本。
- 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）名稱。
- 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）命名空間。
- 資源的 Kubernetes metadata.name 欄位中的 *resourceMatchers[].labelSelectors * ：（ *Optional* ） Label 選取器字串，如中所定義 "Kubernetes文件"。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. 在您以正確的值填入檔案之後 `trident-protect-snapshot-restore-cr.yaml` 、請套用 CR ：

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

使用CLI

步驟

1. 將快照還原至不同的命名空間，以環境中的資訊取代方括號中的值。

   ◦ `snapshot`引數使用格式的命名空間和快照名稱 `<namespace>/<name>`。

   ◦ 此 `namespace-mapping`引數使用以冒號分隔的命名空間，以格式將來源命名空間對應至正確的目的地命名空間 `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>
```

從快照還原至原始命名空間

您可以隨時將快照還原至原始命名空間。

開始之前

確保 AWS 工作階段權杖到期時間足以執行任何長時間執行的 S3 還原作業。如果 Token 在還原作業期間過期，作業可能會失敗。

- 如需檢查目前工作階段權杖到期時間的詳細資訊，請參閱 "AWS API 文件"。
- 如需 AWS 資源認證的詳細資訊，請參閱 "AWS IAM 文件"。

使用 **CR**

步驟

1. 建立自訂資源（ CR ）檔案並命名為 `trident-protect-snapshot-ipr-cr.yaml`。
2. 在您建立的檔案中，設定下列屬性：
   - * metadata.name*: （ _required ）此自訂資源的名稱；為您的環境選擇唯一且合理的名稱。
   - **spec.appVaultRef** ：（ _required _ ）儲存快照內容的 AppVault 名稱。
   - **spec.appArchivePath** ：在 AppVault 中儲存快照內容的路徑。您可以使用下列命令來尋找此路徑：

```
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. （ *Optional* ）如果您只需要選取應用程式的某些資源來還原，請新增篩選功能，以包含或排除標記有特定標籤的資源：

> 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** ：（篩選所需）使用 `Include`或包含或 `Exclude`排除在 resourceMatchers 中定義的資源。新增下列資源配置工具參數、以定義要納入或排除的資源：
     - **resourceFilter.resourceMatchers** ：一組 resourceMatcher 物件。如果您在此陣列中定義多個元素，它們會比對為 OR 作業，而每個元素（群組，種類，版本）內的欄位會比對為 AND 作業。
       - **resourceMatchers[].group** ：（ *Optional* ）要篩選的資源群組。
       - **resourceMatchers[].cher** ：（ *Optional* ）要篩選的資源種類。
       - **resourceMatchers[].version** ：（ *Optional* ）要篩選的資源版本。
       - 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）名稱。
       - 要篩選之資源的 Kubernetes metadata.name 欄位中的 * resourceMatchers[].names* ：（ *Optional* ）命名空間。

- 資源的 Kubernetes metadata.name 欄位中的 *resourceMatchers[].labelSelectors * ：（ *Optional* ） Label 選取器字串，如中所定義 "Kubernetes文件"。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. 在您以正確的值填入檔案之後 `trident-protect-snapshot-ipr-cr.yaml` 、請套用 CR ：

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

使用CLI

步驟

1. 將快照還原至原始命名空間，以環境中的資訊取代方括號中的值。例如：

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

檢查還原作業的狀態

您可以使用命令列來檢查進行中，已完成或已失敗的還原作業狀態。

步驟

1. 使用下列命令可擷取還原作業的狀態，以環境中的資訊取代方括號中的值：

```
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。

還原和容錯移轉作業期間的命名空間註釋和標籤

在還原和容錯移轉作業期間，目的地命名空間中的標籤和註釋會與來源命名空間中的標籤和註釋相符。會新增來源命名空間中不存在的標籤或註釋，並覆寫已存在的任何標籤或註釋，以符合來源命名空間的值。只存在於目的地命名空間上的標籤或註釋會保持不變。

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 安全性內容限制文件"。

您可以在執行還原或容錯移轉作業之前，先設定 Kubernetes 環境變數，以避免覆寫目的地命名空間中的特定註釋 `RESTORE_SKIP_NAMESPACE_ANNOTATIONS`。例如：

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set-string
restoreSkipNamespaceAnnotations="{<annotation_key_to_skip_1>,<annotation_k
ey_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 "設定其他Trident Protect 舵圖設置"。

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。

範例

以下範例提供來源和目的地命名空間，每個命名空間都有不同的註釋和標籤。您可以查看作業前後目的地命名空間的狀態，以及註釋和標籤在目的地命名空間中的組合或覆寫方式。

還原或容錯移轉作業之前

下表說明還原或容錯移轉作業之前的範例來源和目的地命名空間狀態：

命名空間	註釋	標籤
命名空間 nS-1 （來源）	• annotation.one / 機碼：「 updatedvalue 」 • annotation.b2/key ：「 true 」	• 環境 = 正式作業 • Compliance = HIPAA • NAME=ns-1
命名空間 nS-2 （目的地）	• annotation.one / 機碼：「 true 」 • annotation.the/key ：「 FALSE 」	• role = 資料庫

還原作業之後

下表說明還原或容錯移轉作業之後範例目的地命名空間的狀態。某些金鑰已新增，部分已覆寫， `name`標籤已更新以符合目的地命名空間：

命名空間	註釋	標籤
命名空間 nS-2 （目的地）	• annotation.one / 機碼：「 updatedvalue 」 • annotation.b2/key ：「 true 」 • annotation.the/key ：「 FALSE 」	• NAME=nS-2 • Compliance = HIPAA • 環境 = 正式作業 • role = 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`屬性定義從來源應用程式中的現有儲存類別到目標叢集上的新儲存類別的對應。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。

範例：

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

支持的註釋

本節列出了系統中支援配置各種行為的註解。如果使用者未明確設定註解，系統將使用預設值。

註釋	類型	說明	預設值
protected.trident.netapp.io/data-mover-timeout-sec	字串	允許資料移動設備操作停止的最長時間（以秒為單位）。	“300”
protected.trident.netapp.io/kopia-content-cache-size-limit-mb	字串	Kopia 內容快取的最大大小限制（以兆位元組為單位）。	“1000”
protect.trident.netapp.io/pvc-bind-timeout-sec	字串	等待新建立的持久卷聲明 (PVC) 到達的最大時間（以秒為單位）`Bound`操作失敗前的階段。適用於所有還原 CR 類型（備份還原、備份就地還原、快照還原、快照就地還原）。如果您的儲存後端或叢集經常需要更多時間，請使用更高的值。	1200（20分鐘）