



恢复应用程序 Trident

NetApp
January 15, 2026

目录

- 恢复应用程序 1
 - 使用Trident Protect 恢复应用程序 1
 - 从备份还原到不同的命名空间 1
 - 从备份恢复到原始命名空间 4
 - 从备份恢复到不同的集群 7
 - 从快照还原到不同的命名空间 10
 - 从快照恢复到原始命名空间 13
 - 检查还原操作的状态 15
 - 使用高级Trident Protect 恢复设置 16
 - 恢复和故障转移操作期间的命名空间注释和标签 16
 - 支持的字段 17
 - 支持的注释 17

恢复应用程序

使用Trident Protect 恢复应用程序

您可以使用Trident Protect 从快照或备份中恢复您的应用程序。将应用程序恢复到同一集群时，从现有快照恢复速度会更快。



- 恢复应用程序时，为该应用程序配置的所有执行钩子都会随应用程序一起恢复。如果存在恢复后执行钩子，它将作为恢复操作的一部分自动运行。
- 支持从备份恢复到不同的命名空间或恢复到原始命名空间（适用于 qtree 卷）。但是，对于 qtree 卷，不支持从快照恢复到不同的命名空间或恢复到原始命名空间。
- 您可以使用高级设置来自定义恢复操作。欲了解更多信息，请参阅 ["使用高级Trident Protect 恢复设置"](#)。

从备份还原到不同的命名空间

当您使用 BackupRestore CR 将备份还原到不同的命名空间时，Trident Protect 会在新的命名空间中还原应用程序，并为还原的应用程序创建一个应用程序 CR。为了保护已恢复的应用程序，可以创建按需备份或快照，或者制定保护计划。



将备份还原到具有现有资源的不同命名空间不会更改与备份中资源同名的任何资源。要恢复备份中的所有资源，要么删除并重新创建目标命名空间，要么将备份恢复到新的命名空间。

开始之前

确保 AWS 会话令牌的过期时间足以满足任何长时间运行的 S3 恢复操作。如果在恢复操作期间令牌过期，则操作可能会失败。

- 请参阅 ["AWS API 文档"](#)有关检查当前会话令牌过期时间的更多信息。
- 请参阅 ["AWS IAM 文档"](#)有关 AWS 资源凭证的更多信息。



当您使用 Kopia 作为数据移动器恢复备份时，您可以选择在 CR 中指定注释或使用 CLI 来控制 Kopia 使用的临时存储的行为。请参阅 ["科皮亚文件"](#)有关您可以配置的选项的更多信息。使用 ``tridentctl-protect create --help`` 有关使用Trident Protect CLI 指定注释的更多信息，请参阅命令。

使用 CR

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-backup-restore-cr.yaml`。
2. 在您创建的文件中，配置以下属性：
 - **metadata.name:** (必填) 此自定义资源的名称；请为您的环境选择一个唯一且有意义的名称。
 - **spec.appArchivePath:** AppVault 内存储备份内容的路径。您可以使用以下命令查找此路径：

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

- **spec.appVaultRef:** (必需) 存储备份内容的 AppVault 的名称。
- **spec.namespaceMapping:** 恢复操作的源命名空间到目标命名空间的映射。代替 ``my-source-namespace`` 和 ``my-destination-namespace`` 利用来自周围环境的信息。

```
---  
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. (可选) 如果您只需要选择应用程序中的某些资源进行恢复，请添加筛选条件，以包含或排除带有特定标签的资源：



Trident Protect 会自动选择一些资源，因为它们与您选择的资源存在关联。例如，如果您选择持久卷声明资源并且它有一个关联的 pod，Trident Protect 还会恢复关联的 pod。

- **resourceFilter.resourceSelectionCriteria:** (筛选时必需) 使用 ``Include`` 或者 ``Exclude`` 包含或排除 `resourceMatchers` 中定义的资源。添加以下 `resourceMatchers` 参数以定义要包含或排除的资源：
 - **resourceFilter.resourceMatchers:** `resourceMatcher` 对象数组。如果在该数组中定义多个元素，则它们之间按 OR 运算匹配，每个元素内的字段（组、种类、版本）之间按 AND 运算匹配。
 - **resourceMatchers[].group:** (可选) 要筛选的资源组。
 - **resourceMatchers[].kind:** (可选) 要筛选的资源类型。

- **resourceMatchers[].version:** (可选) 要筛选的资源版本。
- **resourceMatchers[].names:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的名称。
- **resourceMatchers[].namespaces:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的命名空间。
- **resourceMatchers[].labelSelectors:** (可选) 资源的 Kubernetes 元数据.name 字段中的标签选择器字符串，如在以下位置定义：["Kubernetes 文档"](#)。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. 填写完之后 `trident-protect-backup-restore-cr.yaml` 将文件的值正确后，应用 CR：

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

使用 CLI

步骤

1. 将备份还原到不同的命名空间，并将括号中的值替换为您环境中的信息。这 namespace-mapping` 该参数使用冒号分隔的命名空间，将源命名空间映射到正确的目标命名空间，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>
```

从备份恢复到原始命名空间

您可以随时将备份恢复到原始命名空间。

开始之前

确保 AWS 会话令牌的过期时间足以满足任何长时间运行的 S3 恢复操作。如果在恢复操作期间令牌过期，则操作可能会失败。

- 请参阅 ["AWS API 文档"](#)有关检查当前会话令牌过期时间的更多信息。
- 请参阅 ["AWS IAM 文档"](#)有关 AWS 资源凭证的更多信息。



当您使用 Kopia 作为数据移动器恢复备份时，您可以选择在 CR 中指定注释或使用 CLI 来控制 Kopia 使用的临时存储的行为。请参阅 ["科皮亚文件"](#)有关您可以配置的选项的更多信息。使用 ``tridentctl-protect create --help``有关使用Trident Protect CLI 指定注释的更多信息，请参阅命令。

使用 CR

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-backup-ipr-cr.yaml`。
2. 在您创建的文件中，配置以下属性：
 - **metadata.name:** (必填) 此自定义资源的名称；请为您的环境选择一个唯一且有意义的名称。
 - **spec.appArchivePath:** AppVault 内存储备份内容的路径。您可以使用以下命令查找此路径：

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

- **spec.appVaultRef:** (必需) 存储备份内容的 AppVault 的名称。

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. (可选) 如果您只需要选择应用程序中的某些资源进行恢复，请添加筛选条件，以包含或排除带有特定标签的资源：



Trident Protect 会自动选择一些资源，因为它们与您选择的资源存在关联。例如，如果您选择持久卷声明资源并且它有一个关联的 pod，Trident Protect 还会恢复关联的 pod。

- **resourceFilter.resourceSelectionCriteria:** (筛选时必需) 使用 `Include` 或者 `Exclude` 包含或排除 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包含或排除的资源：
 - **resourceFilter.resourceMatchers:** resourceMatcher 对象数组。如果在该数组中定义多个元素，则它们之间按 OR 运算匹配，每个元素内的字段（组、种类、版本）之间按 AND 运算匹配。
 - **resourceMatchers[].group:** (可选) 要筛选的资源组。
 - **resourceMatchers[].kind:** (可选) 要筛选的资源类型。
 - **resourceMatchers[].version:** (可选) 要筛选的资源版本。
 - **resourceMatchers[].names:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的名

称。

- **resourceMatchers[].namespaces:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的命名空间。
- **resourceMatchers[].labelSelectors:** (可选) 资源的 Kubernetes 元数据.name 字段中的标签选择器字符串, 如在以下位置定义: "[Kubernetes 文档](#)"。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. 填写完之后 `trident-protect-backup-ipr-cr.yaml` 将文件的值正确后, 应用 CR:

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

使用 CLI

步骤

1. 将备份恢复到原始命名空间, 并将括号中的值替换为您环境中的信息。这 backup` 参数使用命名空间和备份名称, 格式如下 ``<namespace>/<name>`。例如:

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


从备份恢复到不同的集群

如果原始集群出现问题，您可以将备份还原到其他集群。



当您使用 Kopia 作为数据移动器恢复备份时，您可以选择在 CR 中指定注释或使用 CLI 来控制 Kopia 使用的临时存储的行为。请参阅 ["科皮亚文件"](#) 有关您可以配置的选项的更多信息。使用 ``tridentctl-protect create --help`` 有关使用 Trident Protect CLI 指定注释的更多信息，请参阅命令。

开始之前

确保满足以下先决条件：

- 目标集群已安装 Trident Protect。
- 目标集群可以访问与源集群相同的 AppVault 的存储桶路径，备份就存储在该存储桶中。
- 确保在运行 AppVault CR 时，本地环境可以连接到 AppVault CR 中定义的对象存储桶。``tridentctl-protect get appvaultcontent`` 命令。如果网络限制阻止访问，请改为从目标集群上的 pod 内运行 Trident Protect CLI。
- 确保 AWS 会话令牌的过期时间足以满足任何长时间运行的恢复操作。如果在恢复操作期间令牌过期，则操作可能会失败。
 - 请参阅 ["AWS API 文档"](#) 有关检查当前会话令牌过期时间的更多信息。
 - 请参阅 ["AWS 文档"](#) 有关 AWS 资源凭证的更多信息。

步骤

1. 使用 Trident Protect CLI 插件检查目标集群上 AppVault CR 的可用性：

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



确保用于应用程序还原的命名空间存在于目标集群上。

2. 查看目标集群中可用 AppVault 的备份内容：

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

运行此命令将显示 AppVault 中可用的备份，包括其来源集群、相应的应用程序名称、时间戳和归档路径。

示例输出：

```

+-----+-----+-----+-----+
+-----+-----+-----+-----+
|  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 名称和归档路径将应用程序还原到目标集群：

使用 CR

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-backup-restore-cr.yaml`。
2. 在您创建的文件中，配置以下属性：
 - **metadata.name:** (必填) 此自定义资源的名称；请为您的环境选择一个唯一且有意义的名称。
 - **spec.appVaultRef:** (必需) 存储备份内容的 AppVault 的名称。
 - **spec.appArchivePath:** AppVault 内存储备份内容的路径。您可以使用以下命令查找此路径：

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



如果 BackupRestore CR 不可用，您可以使用步骤 2 中提到的命令来查看备份内容。

- **spec.namespaceMapping:** 恢复操作的源命名空间到目标命名空间的映射。代替 ``my-source-namespace`` 和 ``my-destination-namespace`` 利用来自周围环境的信息。

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. 填写完之后 ``trident-protect-backup-restore-cr.yaml`` 将文件的值正确后，应用 CR：

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

使用 CLI

1. 使用以下命令恢复应用程序，将括号中的值替换为您环境中的信息。命名空间映射参数使用冒号分隔的命名空间，将源命名空间映射到正确的目标命名空间，格式为 `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>
```

从快照还原到不同的命名空间

您可以使用自定义资源 (CR) 文件从快照恢复数据，恢复到不同的命名空间或原始源命名空间。当您使用 SnapshotRestore CR 将快照还原到不同的命名空间时，Trident Protect 会在新的命名空间中还原应用程序，并为还原的应用程序创建一个应用程序 CR。为了保护已恢复的应用程序，可以创建按需备份或快照，或者制定保护计划。



SnapshotRestore 支持 `spec.storageClassMapping` 属性，但仅当源存储类和目标存储类使用相同的存储后端时才有效。如果您尝试恢复到 `StorageClass` 如果使用不同的存储后端，则恢复操作将失败。

开始之前

确保 AWS 会话令牌的过期时间足以满足任何长时间运行的 S3 恢复操作。如果在恢复操作期间令牌过期，则操作可能会失败。

- 请参阅 ["AWS API 文档"](#) 有关检查当前会话令牌过期时间的更多信息。
- 请参阅 ["AWS IAM 文档"](#) 有关 AWS 资源凭证的更多信息。

使用 CR

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-snapshot-restore-cr.yaml`。
2. 在您创建的文件中，配置以下属性：

- **metadata.name:** (必填) 此自定义资源的名称；请为您的环境选择一个唯一且有意义的名称。
- **spec.appVaultRef:** (必需) 存储快照内容的 AppVault 的名称。
- **spec.appArchivePath:** AppVault 内存储快照内容的路径。您可以使用以下命令查找此路径：

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

- **spec.namespaceMapping:** 恢复操作的源命名空间到目标命名空间的映射。代替 ``my-source-namespace`` 和 ``my-destination-namespace`` 利用来自周围环境的信息。

```
---
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. (可选) 如果您只需要选择应用程序中的某些资源进行恢复，请添加筛选条件，以包含或排除带有特定标签的资源：



Trident Protect 会自动选择一些资源，因为它们与您选择的资源存在关联。例如，如果您选择持久卷声明资源并且它有一个关联的 pod，Trident Protect 还会恢复关联的 pod。

- **resourceFilter.resourceSelectionCriteria:** (筛选时必需) 使用 ``Include`` 或者 ``Exclude`` 包含或排除 `resourceMatchers` 中定义的资源。添加以下 `resourceMatchers` 参数以定义要包含或排除的资源：
 - **resourceFilter.resourceMatchers:** `resourceMatcher` 对象数组。如果在该数组中定义多个元素，则它们之间按 OR 运算匹配，每个元素内的字段（组、种类、版本）之间按 AND 运算匹配。
 - **resourceMatchers[].group:** (可选) 要筛选的资源组。
 - **resourceMatchers[].kind:** (可选) 要筛选的资源类型。

- **resourceMatchers[].version:** (可选) 要筛选的资源版本。
- **resourceMatchers[].names:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的名称。
- **resourceMatchers[].namespaces:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的命名空间。
- **resourceMatchers[].labelSelectors:** (可选) 资源的 Kubernetes 元数据.name 字段中的标签选择器字符串，如在以下位置定义：["Kubernetes 文档"](#)。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. 填写完之后 `trident-protect-snapshot-restore-cr.yaml` 将文件的值正确后，应用 CR：

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

使用 CLI

步骤

1. 将快照还原到不同的命名空间，并将括号中的值替换为您环境中的信息。
 - 这 snapshot` 参数使用命名空间和快照名称，格式如下 `<namespace>/<name>`。
 - 这 namespace-mapping` 该参数使用冒号分隔的命名空间，将源命名空间映射到正确的目标命名空间，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>
```

从快照恢复到原始命名空间

您可以随时将快照恢复到原始命名空间。

开始之前

确保 AWS 会话令牌的过期时间足以满足任何长时间运行的 S3 恢复操作。如果在恢复操作期间令牌过期，则操作可能会失败。

- 请参阅 ["AWS API 文档"](#) 有关检查当前会话令牌过期时间的更多信息。
- 请参阅 ["AWS IAM 文档"](#) 有关 AWS 资源凭证的更多信息。

使用 CR

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-snapshot-ipr-cr.yaml`。
2. 在您创建的文件中，配置以下属性：
 - **metadata.name:** (必填) 此自定义资源的名称；请为您的环境选择一个唯一且有意义的名称。
 - **spec.appVaultRef:** (必需) 存储快照内容的 AppVault 的名称。
 - **spec.appArchivePath:** AppVault 内存储快照内容的路径。您可以使用以下命令查找此路径：

```
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. (可选) 如果您只需要选择应用程序中的某些资源进行恢复，请添加筛选条件，以包含或排除带有特定标签的资源：



Trident Protect 会自动选择一些资源，因为它们与您选择的资源存在关联。例如，如果您选择持久卷声明资源并且它有一个关联的 pod，Trident Protect 还会恢复关联的 pod。

- **resourceFilter.resourceSelectionCriteria:** (筛选时必需) 使用 `Include` 或者 `Exclude` 包含或排除 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包含或排除的资源：
 - **resourceFilter.resourceMatchers:** resourceMatcher 对象数组。如果在该数组中定义多个元素，则它们之间按 OR 运算匹配，每个元素内的字段（组、种类、版本）之间按 AND 运算匹配。
 - **resourceMatchers[].group:** (可选) 要筛选的资源组。
 - **resourceMatchers[].kind:** (可选) 要筛选的资源类型。
 - **resourceMatchers[].version:** (可选) 要筛选的资源版本。
 - **resourceMatchers[].names:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的名称。
 - **resourceMatchers[].namespaces:** (可选) 要过滤的资源的 Kubernetes 元数据.name 字段中的命名空间。

- **resourceMatchers[].labelSelectors:** (可选) 资源的 Kubernetes 元数据.name 字段中的标签选择器字符串，如在以下位置定义：["Kubernetes 文档"](#)。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. 填写完之后 `trident-protect-snapshot-ipr-cr.yaml` 将文件的值正确后，应用 CR：

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

使用 CLI

步骤

1. 将快照恢复到原始命名空间，并将括号中的值替换为您环境中的信息。例如：

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

检查还原操作的状态

您可以使用命令行来检查正在进行、已完成或已失败的还原操作的状态。

步骤

1. 使用以下命令检索恢复操作的状态，将方括号中的值替换为您环境中的信息：

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

使用高级Trident Protect 恢复设置

您可以使用高级设置（例如注释、命名空间设置和存储选项）自定义恢复操作，以满足您的特定要求。

恢复和故障转移操作期间的命名空间注释和标签

在恢复和故障转移操作期间，目标命名空间中的标签和注释将与源命名空间中的标签和注释相匹配。源命名空间中不存在的目标命名空间中的标签或注释将被添加，并且任何已存在的标签或注释都将被覆盖以匹配源命名空间中的值。仅存在于目标命名空间中的标签或注释保持不变。



如果您使用 Red Hat OpenShift，请务必注意命名空间注释在 OpenShift 环境中的重要作用。命名空间注释确保恢复的 pod 遵守 OpenShift 安全上下文约束 (SCC) 定义的适当权限和安全配置，并且可以访问卷而不会出现权限问题。欲了解更多信息，请参阅 ["OpenShift 安全上下文约束文档"](#)。

您可以通过设置 Kubernetes 环境变量来防止目标命名空间中的特定注解被覆盖。
`RESTORE\_SKIP\_NAMESPACE\_ANNOTATIONS` 在执行恢复或故障转移操作之前。例如：

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



执行恢复或故障转移操作时，任何命名空间注释和标签都将生效。
`restoreSkipNamespaceAnnotations` 和 `restoreSkipNamespaceLabels` 不参与恢复或故障转移操作。确保在初始 Helm 安装期间配置这些设置。欲了解更多信息，请参阅 ["配置AutoSupport和命名空间过滤选项"](#)。

如果您使用 Helm 安装了源应用程序，`--create-namespace` 国旗，给予特殊待遇 `name` 标签键。在恢复或故障转移过程中，Trident Protect 会将此标签复制到目标命名空间，但如果源命名空间的值与源命名空间的值匹配，则会将值更新为目标命名空间的值。如果此值与源命名空间不匹配，则会将其复制到目标命名空间，而不做任何更改。

示例

以下示例展示了源命名空间和目标命名空间，每个命名空间都有不同的注释和标签。您可以查看操作前后目标命名空间的状态，以及目标命名空间中的注释和标签是如何组合或覆盖的。

在恢复或故障转移操作之前

下表说明了恢复或故障转移操作之前示例源命名空间和目标命名空间的状态：

命名空间	标注	标签
命名空间 ns-1 (源)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• 环境=生产 • 合规性=HIPAA • 名称=ns-1
命名空间 ns-2 (目标)	• annotation.one/key: "true" • annotation.three/key: "false"	• 角色=数据库

恢复操作后

下表说明了恢复或故障转移操作后示例目标命名空间的状态。有些键已被添加，有些键已被覆盖，并且 `name` 标签已更新，以匹配目标命名空间：

命名空间	标注	标签
命名空间 ns-2 (目标)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• 名称=ns-2 • 合规性=HIPAA • 环境=生产 • 角色=数据库

支持的字段

本节介绍可用于恢复操作的其他字段。

存储类别映射

这 `spec.storageClassMapping` 属性定义了从源应用程序中存在的存储类到目标集群上新存储类的映射。您可以在具有不同存储类别的集群之间迁移应用程序时或更改 BackupRestore 操作的存储后端时使用此功能。

例子：

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

支持的注释

本节列出了系统中用于配置各种行为的支持注解。如果用户没有明确设置注解，系统将使用默认值。

标注	类型	描述	默认值
protect.trident.netapp.io/data-mover-timeout-sec	string	数据移动器操作允许停止的最长时间（以秒为单位）。	“300”
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	Kopia 内容缓存的最大大小限制（以兆字节为单位）。	“1000”

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。