Restore applications

Trident

NetApp
June 30, 2026

Table of Contents

Restore applications

Restore applications using Trident Protect

You can use Trident Protect to restore your application from a snapshot or backup. Restoring from an existing snapshot will be faster when restoring the application to the same cluster.

- When you restore an application, all execution hooks configured for the application are restored with the app. If a post-restore execution hook is present, it runs automatically as part of the restore operation.
- Restoring from a backup to a different namespace or to the original namespace is supported for qtree volumes. However, restoring from a snapshot to a different namespace or to the original namespace is not supported for qtree volumes.
- You can use advanced settings to customize restore operations. To learn more, refer to Use advanced Trident Protect restore settings.

Restore from a backup to a different namespace

When you restore a backup to a different namespace using a BackupRestore CR, Trident Protect restores the application in a new namespace and creates an application CR for the restored application. To protect the restored application, create on-demand backups or snapshots, or establish a protection schedule.

- Restoring a backup to a different namespace with existing resources will not alter any resources that share names with those in the backup. To restore all resources in the backup, either delete and re-create the target namespace, or restore the backup to a new namespace.
- When using a CR to restore to a new namespace, you must manually create the destination namespace before applying the CR. Trident Protect automatically creates namespaces only when using the CLI.

Before you begin

Ensure that the AWS session token expiration is sufficient for any long-running s3 restore operations. If the token expires during the restore operation, the operation can fail.

- Refer to the AWS API documentation for more information about checking the current session token expiration.
- Refer to the AWS IAM documentation for more information about credentials with AWS resources.

When you restore backups using Kopia as the data mover, you can optionally specify annotations in the CR or using the CLI to control the behavior of the temporary storage used by Kopia. Refer to the Kopia documentation for more information about the options you can configure. Use the `tridentctl-protect create --help` command for more information about specifying annotations with the Trident Protect CLI.

Use a CR

Steps

1. Create the custom resource (CR) file and name it `trident-protect-backup-restore-cr.yaml`.
2. In the file you created, configure the following attributes:
   - **metadata.name**: (*Required*) The name of this custom resource; choose a unique and sensible name for your environment.
   - **spec.appArchivePath**: The path inside AppVault where the backup contents are stored. You can use the following command to find this path:

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

   - **spec.appVaultRef**: (*Required*) The name of the AppVault where the backup contents are stored.
   - **spec.destinationApplicationName**: (*Optional*) The name for the restored application. If provided, the restored application uses this name. If not provided, the restored application uses the source application name.
   - **spec.namespaceMapping**: The mapping of the source namespace of the restore operation to the destination namespace. Replace `my-source-namespace` and `my-destination-namespace` with information from your environment.

     ```
     ---
     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
       destinationApplicationName: my-new-app-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (*Optional*) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:

   > Trident Protect selects some resources automatically because of their relationship with resources that you select. For example, if you select a persistent volume claim resource and it has an associated pod, Trident Protect will also restore the associated pod.

   - **resourceFilter.resourceSelectionCriteria**: (Required for filtering) Use `Include` or `Exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers

parameters to define the resources to be included or excluded:

- **resourceFilter.resourceMatchers**: An array of resourceMatcher objects. If you define multiple elements in this array, they match as an OR operation, and the fields inside each element (group, kind, version) match as an AND operation.
  - **resourceMatchers[].group**: (*Optional*) Group of the resource to be filtered.
  - **resourceMatchers[].kind**: (*Optional*) Kind of the resource to be filtered.
  - **resourceMatchers[].version**: (*Optional*) Version of the resource to be filtered.
  - **resourceMatchers[].names**: (*Optional*) Names in the Kubernetes metadata.name field of the resource to be filtered.
  - **resourceMatchers[].namespaces**: (*Optional*) Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
  - **resourceMatchers[].labelSelectors**: (*Optional*) Label selector string in the Kubernetes metadata.name field of the resource as defined in the Kubernetes documentation. For example: `"trident.netapp.io/os=linux"`.

    For example:

```
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. After you populate the `trident-protect-backup-restore-cr.yaml` file with the correct values, apply the CR:

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

Use the CLI

Steps

1. Restore the backup to a different namespace, replacing values in brackets with information from your environment. The `namespace-mapping` argument uses colon-separated namespaces to map source namespaces to the correct destination namespaces in the format

`source1:dest1,source2:dest2`. For example:

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

Restore from a backup to the original namespace

You can restore a backup to the original namespace at any time. When you perform an in-place restore, Trident Protect automatically manages protection schedules and in-progress operations to prevent invalid recovery points:

- All enabled protection schedules for the application are disabled before the restore begins. This prevents scheduled backups or snapshots from running while the application resources are being restored.
- After the restore completes successfully, only the schedules that were enabled before the restore are re-enabled. Schedules that were already disabled remain disabled.
- Any in-progress backup or snapshot operations are cancelled before the restore begins. If an operation does not cancel within 5 minutes, the restore proceeds and logs a warning in the restore CR status.

Before you begin

Ensure that the AWS session token expiration is sufficient for any long-running s3 restore operations. If the token expires during the restore operation, the operation can fail.

- Refer to the AWS API documentation for more information about checking the current session token expiration.
- Refer to the AWS IAM documentation for more information about credentials with AWS resources.

> When you restore backups using Kopia as the data mover, you can optionally specify annotations in the CR or using the CLI to control the behavior of the temporary storage used by Kopia. Refer to the Kopia documentation for more information about the options you can configure. Use the `tridentctl-protect create --help` command for more information about specifying annotations with the Trident Protect CLI.

Use a CR

Steps

1. Create the custom resource (CR) file and name it `trident-protect-backup-ipr-cr.yaml`.
2. In the file you created, configure the following attributes:
   - **metadata.name**: (*Required*) The name of this custom resource; choose a unique and sensible name for your environment.
   - **spec.appArchivePath**: The path inside AppVault where the backup contents are stored. You can use the following command to find this path:

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

   - **spec.appVaultRef**: (*Required*) The name of the AppVault where the backup contents are stored.

     For example:

     ```
     ---
     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*) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:

   > Trident Protect selects some resources automatically because of their relationship with resources that you select. For example, if you select a persistent volume claim resource and it has an associated pod, Trident Protect will also restore the associated pod.

   - **resourceFilter.resourceSelectionCriteria**: (Required for filtering) Use `Include` or `Exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers parameters to define the resources to be included or excluded:
     - **resourceFilter.resourceMatchers**: An array of resourceMatcher objects. If you define multiple elements in this array, they match as an OR operation, and the fields inside each element (group, kind, version) match as an AND operation.
       - **resourceMatchers[].group**: (*Optional*) Group of the resource to be filtered.
       - **resourceMatchers[].kind**: (*Optional*) Kind of the resource to be filtered.
       - **resourceMatchers[].version**: (*Optional*) Version of the resource to be filtered.

- **resourceMatchers[].names**: (*Optional*) Names in the Kubernetes metadata.name field of the resource to be filtered.
- **resourceMatchers[].namespaces**: (*Optional*) Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label selector string in the Kubernetes metadata.name field of the resource as defined in the Kubernetes documentation. For example: `"trident.netapp.io/os=linux"`.

  For example:

```
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. After you populate the `trident-protect-backup-ipr-cr.yaml` file with the correct values, apply the CR:

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

Use the CLI

Steps

1. Restore the backup to the original namespace, replacing values in brackets with information from your environment. The `backup` argument uses a namespace and backup name in the format `<namespace>/<name>`. For example:

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

Restore from a backup to a different cluster

You can restore a backup to a different cluster if there is an issue with the original cluster.

- When you restore backups using Kopia as the data mover, you can optionally specify annotations in the CR or using the CLI to control the behavior of the temporary storage used by Kopia. Refer to the Kopia documentation for more information about the options you can configure. Use the `tridentctl-protect create --help` command for more information about specifying annotations with the Trident Protect CLI.
- When using a CR to restore to a new namespace, you must manually create the destination namespace before applying the CR. Trident Protect automatically creates namespaces only when using the CLI.

Before you begin

Ensure the following prerequisites are met:

- The destination cluster has Trident Protect installed.
- The destination cluster has access to the bucket path of the same AppVault as the source cluster, where the backup is stored.
- Ensure that your local environment can connect to the object storage bucket defined in the AppVault CR when running the `tridentctl-protect get appvaultcontent` command. If network restrictions prevent access, run the Trident Protect CLI from within a pod on the destination cluster instead.
- Ensure that the AWS session token expiration is sufficient for any long-running restore operations. If the token expires during the restore operation, the operation can fail.
  - Refer to the AWS API documentation for more information about checking the current session token expiration.
  - Refer to the AWS documentation for more information about credentials with AWS resources.

Steps

1. Verify that the AppVault CR exists on the destination cluster using the Trident Protect CLI plugin:

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

   If the AppVault CR does not exist on the destination cluster, create it by following the steps in Use Trident Protect AppVault objects to manage buckets.

2. View the backup contents of the available AppVault on the destination cluster, and note `appArchivePath` of the backup you want to restore:

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

   Running this command displays the available backups in the AppVault, including their originating clusters, corresponding application names, timestamps, and archive paths.

Example output:

```
+------------+----------+--------+----------------
+--------------------------+-------------+
|   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. Restore the application to the destination cluster using the AppVault name and archive path:

> When using a CR, ensure that the namespace intended for the application restore exists on the destination cluster.

Use a CR

4. Create the custom resource (CR) file and name it `trident-protect-backup-restore-cr.yaml`.
5. In the file you created, configure the following attributes:
   - **metadata.name**: (*Required*) The name of this custom resource; choose a unique and sensible name for your environment.
   - **spec.appVaultRef**: (*Required*) The name of the AppVault where the backup contents are stored.
   - **spec.appArchivePath**: (*Required*) The path inside AppVault where the backup contents are stored. Use the command from step 2 to view the backup contents and find `appArchivePath` for the backup you want to restore.
   - **spec.destinationApplicationName**: (*Optional*) The name for the restored application. If provided, the restored application uses this name. If not provided, the restored application uses the source application name.
   - **spec.namespaceMapping**: The mapping of the source namespace of the restore operation to the destination namespace. Replace `my-source-namespace` and `my-destination-namespace` with information from your environment.

     For example:

     ```
     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
       destinationApplicationName: my-new-app-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

6. After you populate the `trident-protect-backup-restore-cr.yaml` file with the correct values, apply the CR:

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

Use the CLI

4. Use the following command to restore the application, replacing values in brackets with information from your environment. The namespace-mapping argument uses colon-separated namespaces to map source namespaces to the correct destination namespaces in the format source1:dest1,source2:dest2. For example:

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

Restore from a snapshot to a different namespace

You can restore data from a snapshot using a custom resource (CR) file either to a different namespace or the original source namespace. When you restore a snapshot to a different namespace using a SnapshotRestore CR, Trident Protect restores the application in a new namespace and creates an application CR for the restored application. To protect the restored application, create on-demand backups or snapshots, or establish a protection schedule.

- SnapshotRestore supports the `spec.storageClassMapping` attribute, but only when the source and destination storage classes use the same storage backend. If you attempt to restore to a `StorageClass` that uses a different storage backend, the restore operation will fail.
- When using a CR to restore to a new namespace, you must manually create the destination namespace before applying the CR. Trident Protect automatically creates namespaces only when using the CLI.

Before you begin

Ensure that the AWS session token expiration is sufficient for any long-running s3 restore operations. If the token expires during the restore operation, the operation can fail.

- Refer to the AWS API documentation for more information about checking the current session token expiration.
- Refer to the AWS IAM documentation for more information about credentials with AWS resources.

Use a CR

Steps

1. Create the custom resource (CR) file and name it `trident-protect-snapshot-restore-cr.yaml`.
2. In the file you created, configure the following attributes:
   - **metadata.name**: (*Required*) The name of this custom resource; choose a unique and sensible name for your environment.
   - **spec.appVaultRef**: (*Required*) The name of the AppVault where the snapshot contents are stored.
   - **spec.appArchivePath**: The path inside AppVault where the snapshot contents are stored. You can use the following command to find this path:

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

   - **spec.destinationApplicationName**: (*Optional*) The name for the restored application. If provided, the restored application uses this name. If not provided, the restored application uses the source application name.
   - **spec.namespaceMapping**: The mapping of the source namespace of the restore operation to the destination namespace. Replace `my-source-namespace` and `my-destination-namespace` with information from your environment.

     ```
     ---
     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*) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:

   > Trident Protect selects some resources automatically because of their relationship with resources that you select. For example, if you select a persistent volume claim resource and it has an associated pod, Trident Protect will also restore the associated pod.

   - **resourceFilter.resourceSelectionCriteria**: (Required for filtering) Use `Include` or `Exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers

parameters to define the resources to be included or excluded:

- **resourceFilter.resourceMatchers**: An array of resourceMatcher objects. If you define multiple elements in this array, they match as an OR operation, and the fields inside each element (group, kind, version) match as an AND operation.
  - **resourceMatchers[].group**: (*Optional*) Group of the resource to be filtered.
  - **resourceMatchers[].kind**: (*Optional*) Kind of the resource to be filtered.
  - **resourceMatchers[].version**: (*Optional*) Version of the resource to be filtered.
  - **resourceMatchers[].names**: (*Optional*) Names in the Kubernetes metadata.name field of the resource to be filtered.
  - **resourceMatchers[].namespaces**: (*Optional*) Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
  - **resourceMatchers[].labelSelectors**: (*Optional*) Label selector string in the Kubernetes metadata.name field of the resource as defined in the Kubernetes documentation. For example: `"trident.netapp.io/os=linux"`.

    For example:

```
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. After you populate the `trident-protect-snapshot-restore-cr.yaml` file with the correct values, apply the CR:

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

Use the CLI

Steps

1. Restore the snapshot to a different namespace, replacing values in brackets with information from your environment.

- The `snapshot` argument uses a namespace and snapshot name in the format `<namespace>/<name>`.
- The `namespace-mapping` argument uses colon-separated namespaces to map source namespaces to the correct destination namespaces in the format `source1:dest1,source2:dest2`.

  For example:

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

Restore from a snapshot to the original namespace

You can restore a snapshot to the original namespace at any time. When you perform an in-place restore, Trident Protect automatically manages protection schedules and in-progress operations to prevent invalid recovery points:

- All enabled protection schedules for the application are disabled before the restore begins. This prevents scheduled backups or snapshots from running while the application resources are being restored.
- After the restore completes successfully, only the schedules that were enabled before the restore are re-enabled. Schedules that were already disabled remain disabled.
- Any in-progress backup or snapshot operations are cancelled before the restore begins. If an operation does not cancel within 5 minutes, the restore proceeds and logs a warning in the restore CR status.

Before you begin

Ensure that the AWS session token expiration is sufficient for any long-running s3 restore operations. If the token expires during the restore operation, the operation can fail.

- Refer to the AWS API documentation for more information about checking the current session token expiration.
- Refer to the AWS IAM documentation for more information about credentials with AWS resources.

Use a CR

Steps

1. Create the custom resource (CR) file and name it `trident-protect-snapshot-ipr-cr.yaml`.
2. In the file you created, configure the following attributes:
   - **metadata.name**: (*Required*) The name of this custom resource; choose a unique and sensible name for your environment.
   - **spec.appVaultRef**: (*Required*) The name of the AppVault where the snapshot contents are stored.
   - **spec.appArchivePath**: The path inside AppVault where the snapshot contents are stored. You can use the following command to find this path:

     ```
     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*) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:

   > Trident Protect selects some resources automatically because of their relationship with resources that you select. For example, if you select a persistent volume claim resource and it has an associated pod, Trident Protect will also restore the associated pod.

   - **resourceFilter.resourceSelectionCriteria**: (Required for filtering) Use `Include` or `Exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers parameters to define the resources to be included or excluded:
     - **resourceFilter.resourceMatchers**: An array of resourceMatcher objects. If you define multiple elements in this array, they match as an OR operation, and the fields inside each element (group, kind, version) match as an AND operation.
       - **resourceMatchers[].group**: (*Optional*) Group of the resource to be filtered.
       - **resourceMatchers[].kind**: (*Optional*) Kind of the resource to be filtered.
       - **resourceMatchers[].version**: (*Optional*) Version of the resource to be filtered.
       - **resourceMatchers[].names**: (*Optional*) Names in the Kubernetes metadata.name field of the resource to be filtered.

- **resourceMatchers[].namespaces**: (*Optional*) Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label selector string in the Kubernetes metadata.name field of the resource as defined in the Kubernetes documentation. For example: `"trident.netapp.io/os=linux"`.

  For example:

```
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. After you populate the `trident-protect-snapshot-ipr-cr.yaml` file with the correct values, apply the CR:

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

Use the CLI

Steps

1. Restore the snapshot to the original namespace, replacing values in brackets with information from your environment. For example:

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

Check the status of a restore operation

You can use the command line to check the status of a restore operation that is in progress, has completed, or has failed.

Steps

1. Use the following command to retrieve status of the restore operation, replacing values in brackes with information from your environment:

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

Use advanced Trident Protect restore settings

You can customize restore operations using advanced settings such as annotations, namespace settings, and storage options to meet your specific requirements.

Namespace annotations and labels during restore and failover operations

During restore and failover operations, labels and annotations in the destination namespace are made to match the labels and annotations in the source namespace. Labels or annotations from the source namespace that don't exist in the destination namespace are added, and any labels or annotations that already exist are overwritten to match the value from the source namespace. Labels or annotations that exist only on the destination namespace remain unchanged.

If you use Red Hat OpenShift, it's important to note the critical role of namespace annotations in OpenShift environments. Namespace annotations ensure that restored pods adhere to the appropriate permissions and security configurations defined by OpenShift security context constraints (SCCs) and can access volumes without permission issues. For more information, refer to the OpenShift security context constraints documentation.

You can prevent specific annotations in the destination namespace from being overwritten by setting the Kubernetes environment variable `RESTORE_SKIP_NAMESPACE_ANNOTATIONS` before you perform the restore or failover operation. For example:

```
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
```

When performing restore or failover operation, any namespace annotations and labels specified in `restoreSkipNamespaceAnnotations` and `restoreSkipNamespaceLabels` are excluded from the restore or failover operation. Ensure these settings are configured during the initial Helm installation. To learn more, refer to Configure additional Trident Protect helm chart settings.

If you installed the source application using Helm with the `--create-namespace` flag, special treatment is given to the `name` label key. During the restore or failover process, Trident Protect copies this label to the destination namespace, but updates the value to the destination namespace value if the value from source

matches the source namespace. If this value doesn’t match the source namespace it is copied to the destination namespace with no changes.

Example

The following example presents a source and destination namespace, each with different annotations and labels. You can see the state of the destination namespace before and after the operation, and how the annotations and labels are combined or overwritten in the destination namespace.

Before the restore or failover operation

The following table illustrates the state of the example source and destination namespaces before the restore or failover operation:

Namespace	Annotations	Labels
Namespace ns-1 (source)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• environment=production • compliance=hipaa • name=ns-1
Namespace ns-2 (destination)	• annotation.one/key: "true" • annotation.three/key: "false"	• role=database

After the restore operation

The following table illustrates the state of the example destination namespace after the restore or failover operation. Some keys have been added, some have been overwritten, and the `name` label has been updated to match the destination namespace:

Namespace	Annotations	Labels
Namespace ns-2 (destination)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• name=ns-2 • compliance=hipaa • environment=production • role=database

Supported fields

This section describes additional fields available for restore operations.

Storage class mapping

The `spec.storageClassMapping` attribute defines a mapping from a storage class present in the source application to a new storage class on the target cluster. You can use this when migrating applications between clusters with different storage classes or when changing the storage backend for BackupRestore operations.

Example:

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

Supported annotations

This section lists the supported annotations for configuring various behaviors in the system. If an annotation is not explicitly set by the user, the system will use the default value.

Annotation	Type	Description	Default value
protect.trident.netapp.io/data-mover-timeout-sec	string	The maximum time (in seconds) allowed for data mover operation to be stalled.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	The maximum size limit (in megabytes) for the Kopia content cache.	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	string	Maximum time (in seconds) to wait for any newly created PersistentVolumeClaims (PVCs) to reach the `Bound` phase before the operations fails. Applies to all restore CR types (BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore). Use a higher value if your storage backend or cluster often requires more time.	"1200" (20 minutes)

Copyright information

Trademark information