Restore Kubernetes applications

NetApp Backup and Recovery

NetApp
February 11, 2026

Table of Contents

Restore Kubernetes applications

Restore Kubernetes applications using the web UI

NetApp Backup and Recovery enables you to restore applications that you have protected with a protection policy. To restore an application, an application needs to have at least one restore point available. A restore point consists of either the local snapshot or the backup to the object store (or both). You can restore an application using the local, secondary, or object store archive.

Before you begin

If you are restoring an application that was backed up using Trident Protect, ensure that Trident Protect is installed on both the source and destination clusters.

Required NetApp Console role

Organization admin or SnapCenter admin. Learn about NetApp Backup and Recovery access roles. Learn about NetApp Console access roles for all services.

Steps

1. In the NetApp Backup and Recovery menu, select **Restore**.
2. Choose a Kubernetes application from the list, and select **View and Restore** for that application.

   The list of restore points appears.

3. Select the **Restore** button for the restore point you want to use.

General settings

1. Choose the source location to restore from.
2. Choose the destination cluster from the **Cluster** list.

   > ⓘ Restoring a local snapshot created by Trident Protect to a different cluster is not supported at this time.

3. Choose to restore to the original namespaces or new namespaces.
4. If you chose to restore to new namespaces, enter the destination namespace or namespaces to use.
5. Select **Next**.

Resource selection

1. Choose whether you want to restore all resources associated with the application or use a filter to select specific resources to restore:

Restore all resources

a. Select **Restore all resources**.

b. Select **Next**.

Restore specific resources

a. Select **Selective resources**.

b. Choose the behavior of the resource filter. If you choose **Include**, the resources you select are restored. If you choose **Exclude**, the resources you select are not restored.

c. Select **Add rules** to add rules that define filters for selecting resources. You need at least one rule to filter resources.

   Each rule can filter on criteria such as the resource namespace, labels, group, version, and kind.

d. Select **Save** to save each rule.

e. When you have added all the rules you need, select **Search** to see the resources available in the backup archive that match your filter criteria.

   > The resources shown are the resources that currently exist on the cluster.

f. When satisfied with the results, select **Next**.

Destination settings

1. Expand the **Destination settings** section and choose to restore either to the default storage class, a different storage class, or if you are restoring to a different cluster, to map the storage classes to the destination cluster.
2. If you chose to restore to a different storage class, select a destination storage class to match each source storage class.
3. Optionally, if you are restoring a backup or snapshot that was made using Trident Protect, view the details of the AppVault used as the storage bucket for the restore operation. If there is a change in your environment or the AppVault status, select **Sync App Vault** to refresh the details.

   > If you need to create an AppVault on a Kubernetes cluster to facilitate restoring a backup or snapshot created using Trident Protect, refer to Use Trident Protect AppVault objects to manage buckets.

4. Optionally, expand the **Restore scripts** section and enable the **Postscript** option to choose an execution hook template that will run after the restore operation is complete. If needed, enter any arguments that the script needs and add label selectors to filter resources based on resource labels.
5. Select **Restore**.

Restore Kubernetes applications using a custom resource

You can use custom resources to restore your applications 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 custom resource restore settings.

Restore a backup to a different namespace

When you restore a backup to a different namespace using a BackupRestore CR, Backup and Recovery 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. Backup and Recovery automatically creates namespaces only when using the CLI.

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-sessiontoken-note.adoc[]

When you restore backups using Kopia as the data mover, you can optionally specify annotations in the CR 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.

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.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
  namespaceMapping: [{"source": "my-source-namespace", "destination": "my-destination-namespace"}]
```

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-selective-restore.adoc[]

3. 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
```

Restore a backup to the original namespace

You can restore a backup to the original namespace at any time.

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-sessiontoken-note.adoc[]

When you restore backups using Kopia as the data mover, you can optionally specify annotations in the CR 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.

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
```

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-selective-restore.adoc[]

3. 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
```

Restore 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 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.
- When using a CR to restore to a new namespace, you must manually create the destination namespace before applying the CR.

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 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. Check the availability of the AppVault CR on the destination cluster using Trident Protect CLI plugin:

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

Ensure that the namespace intended for the application restore exists on the destination cluster.

2. View the backup contents of the available AppVault from the destination cluster:

   ```
   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:

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**: 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}'
     ```

     If BackupRestore CR is not available, you can use the command mentioned in step 2 to view the backup contents.

- **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
    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
   ```

Restore 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, Backup and Recovery 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.

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-sessiontoken-note.adoc[]

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.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"}]
```

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-selective-restore.adoc[]

3. 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
```

Restore a snapshot to the original namespace

You can restore a snapshot to the original namespace at any time.

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-sessiontoken-note.adoc[]

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
```

Unresolved directive in br-use-restore-kubernetes-applications-cr.adoc - include::../_include/restore-include-selective-restore.adoc[]

3. 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 advanced custom resource restore settings

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

Unresolved directive in br-use-kubernetes-advanced-restore-settings.adoc - include::../_include/namespace-anno-labels.adoc[]

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