Disaster Recovery

NetApp virtualization solutions

NetApp
July 24, 2026

Table of Contents

Disaster Recovery

Trident Protect/AppMirror prerequisites for OpenShift Virtualization Disaster Recovery

This section provides information about the prerequisites for using NetApp Trident Protect and AppMirror for disaster recovery of OpenShift Virtualization workloads. This includes the necessary cluster configurations, storage requirements, and installation steps to set up Trident Protect and AppMirror for effective disaster recovery of VMs running on OpenShift Virtualization.

Prerequisites

1. A Red Hat OpenShift cluster with OpenShift Virtualization installed and configured. For installation instructions, see the documentation here.
2. Two NetApp ONTAP storage systems with an SVM configured on each, to provide storage for the active and the disaster recovery OpenShift clusters. Each SVM must have at least one management LIF and one data LIF configured and accessible from the corresponding OpenShift cluster.
3. Cluster peering and SVM peering configured between the two ONTAP systems to allow replication of data from the source cluster's SVM to the target cluster's SVM. For instructions on how to set up cluster peering and SVM peering, see the ONTAP cluster peering documentation here.
4. NetApp Trident installed and configured on the OpenShift cluster to provision persistent storage from the ONTAP SVM. For installation instructions, see the Trident documentation here.
5. Trident backend configuration created for the ONTAP SVM and a StorageClass configured on the OpenShift cluster to use that Trident backend as the provisioner. For instructions on how to create a Trident backend configuration and StorageClass, see the Trident backend configuration documentation here.
6. Cluster-admin access to the OpenShift cluster and admin access to the ONTAP storage system to perform the necessary configurations.
7. An admin workstation with `tridentctl` and `oc` tools installed and added to the $PATH environment variable to interact with the OpenShift cluster and Trident.
8. Adequate hardware resources on the OpenShift cluster to support the OpenShift Virtualization operator and the VMs that will be provisioned. For more details on resource requirements, see the documentation here.
9. Optionally, configure node placement rules for OpenShift Virtualization to specify a subset of cluster nodes to host the OpenShift Virtualization operators, controllers, and VMs. For instructions on how to configure node placement rules, see the documentation here.
10. For the storage backing OpenShift Virtualization, it is recommended to have a dedicated StorageClass that requests storage from a particular Trident backend, which in turn is backed by a dedicated SVM. This maintains a level of multitenancy with regard to the data being served for VM-based workloads on the OpenShift cluster.
11. A VM must be available in OpenShift Virtualization. For details about deploying a new VM, or migrating an existing VM into OpenShift Virtualization, see the appropriate section in the documentation.

Install NetApp Trident Protect for OpenShift Virtualization

To enable disaster recovery capabilities for OpenShift Virtualization workloads using NetApp Trident Protect,

you must have Trident Protect installed and configured on the OpenShift cluster. Trident Protect can be installed using a Helm chart.

Trident Protect installation must be done after installing Trident.

For complete installation instructions, see the Trident Protect documentation here.

Show example


```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# oc get pods -n trident-protect
NAME                                                READY   STATUS    RESTARTS   AGE
autosupportbundle-e9252a48-34a9-4b40-99c2-c00876d962ee-bk2vx   1/1     Running   0          16h
trident-protect-controller-manager-7b76c8b59f-2rmh2            2/2     Running   0          22h
[root@localhost SnapMirror]#
```

Source Cluster Setup for Red Hat OpenShift Virtualization Disaster Recovery

This section provides information about setting up the source cluster for disaster recovery of OpenShift Virtualization workloads using NetApp Trident Protect and AppMirror. This includes creating an AppVault using ONTAP S3, creating a Trident Protect Application for the source VM, and creating on-demand and scheduled snapshots of the VM using Trident Protect. These steps are essential to ensure that the source cluster is properly configured to replicate data to the disaster recovery target cluster and enable recovery of VMs in case of a disaster.

Setup Disaster Recovery Source Cluster for OpenShift Virtualization

All the following steps are performed on the source cluster, which is the cluster where the protected VM is currently running.

Create AppVault using ONTAP S3

This section shows how to set up an AppVault object in Trident Protect using ONTAP S3 object storage. The AppVault created in this step will be used to store replicated data from the source cluster, and that data will be used for recovery in the disaster recovery target cluster.

Use oc commands and the yaml files shown below to create a secret and the AppVault custom resource for ONTAP s3. Ensure that you create them in the trident-protect namespace.

Show example

```
apiVersion: v1
# You can provide the keys either as stringData or base 64 encoded data
stringData:
  accessKeyID: "<access key id as obtained from ONTAP>"
  secretAccessKey: "<secret access key as obtained from ONTAP>"
#data:
  #accessKeyID: <base 64 encoded value of access key>
  #secretAccessKey: <base 64 encoded value of secret access key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque
```

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    s3:
      bucketName: oc-dr
      endpoint: <data lif to use to access S3>
      secure: "false" # Setting this to "false" will disable TLS, which
is not recommended for production environments.
      skipCertValidation: "true" # When TLS is enabled, setting
skipCertValidation to "true" will skip TLS certificate validation,
which is not recommended for production environments.
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-secret
  providerType: OntapS3
```

Ensure that ONTAP S3 vault is created and is in the Available state

```
oc create -f app-vault-secret.yaml -n trident-protect
oc create -f app-vault.yaml -n trident-protect
```

```
[root@00-50-56-b5-2b-9b dr]# oc get appvault -n trident-protect
NAME                STATE       ERROR   MESSAGE   AGE
ontap-s3-appvault   Available                     2d23h
```

Create a Trident Protect App for the source VM

Create an Application custom resource in the namespace where the source VM is located.

Show example

```
# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: test-source-app
  namespace: test-dr-ns
spec:
  includedNamespaces:
  - namespace: test-dr-ns
```

```
tridentctl-protect create app <source-vm> -n <source-ns> --namespaces
<source-ns>
```

```
[root@00-50-56-b5-2b-9b dr]# oc get applications.protect.trident.netapp.io -n test-dr-ns
NAME              PROTECTION STATE   AGE
test-source-app   Partial            25h
```

The VM and its PVC in the namespace included in the App custom resource is shown below.

```
[root@00-50-56-b5-2b-9b dr]# oc get vm,pods,pvc -n test-dr-ns
NAME                                  AGE   STATUS    READY
virtualmachine.kubevirt.io/test-dr-vm1   25h   Running   True

NAME                                READY   STATUS    RESTARTS   AGE
pod/virt-launcher-test-dr-vm1-ns9qq   1/1     Running   0          25h

NAME                                  STATUS   VOLUME                                     CAPACITY     ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/test-dr-vm1     Bound    pvc-43b51d12-ce76-4063-bd59-6e74588ee266   34144990004  RWX            sc-nas         <unset>                 25h
```

Create a snapshot of the VM using Trident Protect

You can create an on-demand snapshot of the VM using Trident Protect CLI or API. This snapshot will be stored in the AppVault created in the previous step and can be used for recovery in the disaster recovery target cluster.

Show example

```
# snapshot.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  name: test-source-dr-init-snapshot
  namespace: test-dr-ns
spec:
  applicationRef: test-source-app
  appVaultRef: ontap-s3-appvault
  reclaimPolicy: Delete
```

```
[root@00-50-56-b5-2b-9b dr]# oc get snapshot -n test-dr-ns
NAME                          APP              RECLAIM POLICY   STATE       ERROR   AGE
test-source-dr-init-snapshot  test-source-app   Delete           Completed           25h
```

Create a scheduled snapshot of the VM using Trident Protect

You can also create a scheduled snapshot by creating a Schedule custom resource in the same namespace as the Application custom resource. The schedule will automatically create snapshots of the VM at the specified schedule and store them in the AppVault.

Show example

```
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  name: appmirror-sched1
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: test-source-app
  backupRetention: "0"
  enabled: true
  granularity: Custom
  recurrenceRule: |-
    DTSTART:20240901T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  snapshotRetention: "2"
```

Target Cluster Setup for Red Hat OpenShift Virtualization Disaster Recovery

This section provides information about setting up the target cluster for disaster recovery of OpenShift Virtualization workloads using NetApp Trident Protect and AppMirror. This includes creating an AppVault using ONTAP S3 and creating a new namespace for the destination VM. These steps are essential to ensure that the destination cluster is properly configured to receive replicated data from the source cluster and enable recovery of VMs in case of a disaster.

1. Create an AppVault Object in the destination cluster. This should point to the same S3 as the source so that it can get the snapshot from the S3 where the source cluster placed it.

Show example

```
apiVersion: v1
# You can provide the keys either as stringData or base 64 encoded
data
stringData:
  accessKeyID: "<access key id as obtained from ONTAP>"
  secretAccessKey: "<secret access key as obtained from ONTAP>"
#data:
  #accessKeyID: <base 64 encoded value of access key>
  #secretAccessKey: <base 64 encoded value of secret access key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque
```

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    s3:
      bucketName: oc-dr
      endpoint: <data lif to use to access S3>
      secure: "false" # Setting this to "false" will disable TLS,
which is not recommended for production
      skipCertValidation: "true" # When TLS is enabled, setting
skipCertValidation to "true" will skip TLS certificate validation,
which is not recommended for production environments.
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-secret
  providerType: OntapS3
```

```
oc create -f app-vault-secret.yaml -n trident-protect
oc create -f app-vault.yaml -n trident-protect
```

```
[root@00-50-56-b5-01-5b dr]# oc get appvault -A
NAMESPACE          NAME                                STATE       ERROR   MESSAGE   AGE
trident-protect    ontap-s3-appvault                   Available                     29h
```

2. Create a new namespace on the destination cluster to host the failed over VM. Create an App in the destination cluster that references the AppVault created in step 1 and the new namespace. This App will be used to establish the AppMirror relationship between the source and destination cluster, and to manage the replication of data from the source cluster to the destination cluster.

Show example

```
# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: test-dest-app
  namespace: test-dr-ns-dest
spec:
  includedNamespaces:
  - namespace: test-dr-ns-dest
```

```
tridentctl-protect create app <dest-app-name> -n <dest-ns>
--namespaces <dest-ns>
```

Failover of a VM in OpenShift Virtualization Disaster Recovery

This section describes how to perform a failover of a VM from a source cluster to a disaster recovery target cluster using NetApp Trident Protect and AppMirror. This is useful for testing the failover process, and for performing planned maintenance on the source cluster. By following these steps, you can ensure that your VM is running on the disaster recovery target cluster with all changes intact after a disaster recovery event.

Failover without site outage for OpenShift Virtualization

This section describes how to perform a failover of a VM from a source cluster without a site outage. This is useful for testing the failover process, and for performing planned maintenance on the source cluster. +

To perform a failover without a site outage, first ensure that the source and target clusters are set up for disaster recovery as described in the previous sections. Then, perform the following steps:

1. Establish the AppMirror relationship between the source and the target cluster. When the AppMirror relationship is established, the most recent snapshot is transferred to the destination namespace. The PVC is created for the VM in the destination namespace, however, the VM pod is not yet created in the destination namespace.

   Show example

   ```
   # application-mirror-relationship.yaml
   apiVersion: protect.trident.netapp.io/v1
   kind: AppMirrorRelationship
   metadata:
     name: amr1
   spec:
     desiredState: Established
     destinationAppVaultRef: ontap-s3-appvault
     destinationApplicationRef: test-dest-app
     namespaceMapping:
     - destination: test-dr-ns-dest
       source: test-dr-ns
     recurrenceRule: |-
       DTSTART:20240901T000200Z
       RRULE:FREQ=MINUTELY;INTERVAL=5
     sourceAppVaultRef: ontap-s3-appvault
     sourceApplicationName: test-source-app
     sourceApplicationUID: "<application-uid-of-the-source-app>"
     storageClassName: "sc-nas"
   ```

   ```
   [root@00-50-56-b5-2b-9b dr]# oc get amr -n test-dr-ns
   NAME   DESIRED STATE   STATE          ERROR   AGE
   amr1   Established     Establishing           115s
   ```

   ```
   [root@00-50-56-b5-2b-9b dr]# oc get amr -A
   NAMESPACE     NAME   DESIRED STATE   STATE         ERROR   AGE
   test-dr-ns    amr1   Established     Established           94s
   ```

   The AppMirror relationship should be created from the cli of the destination cluster. It should be created with the desired state as Established. This will ensure that the latest snapshot data is replicated to the destination cluster.

2. Promote the AppMirror relationship. Change the desired state of the relationship to "Promoted" to create the VM in the destination namespace. The VM is still running in the source namespace. The latest snapshot data is replicated to the destination cluster and the VM is started on the destination cluster. This

method allows you to test failover and maintain VM availability without needing to take down the entire source cluster.

Show example

```
oc patch amr amr1 -n test-dr-ns-dest --type=merge -p
'{"spec":{"desiredState":"Promoted"}}'
```

```
[root@00-50-56-b5-01-5b dr]# oc get amr -n test-dr-ns-dest
NAME   DESIRED STATE   STATE      ERROR   AGE
amr2   Promoted        Promoted           25h
```

```
oc get pod, pvc,vm -n test-dr-ns-dest
oc get pod, pvc,vm -n test-dr-ns
```

Failback of a VM in OpenShift Virtualization Disaster Recovery

This section provides information about the failback process for a VM in OpenShift Virtualization disaster recovery using NetApp Trident Protect and AppMirror. After a failover operation has been performed to switch to a disaster recovery target cluster, you can use the failback process to reverse the replication direction and restore the original source cluster as the primary cluster for the VM. This involves resyncing any changes made to the VM during the failover period back to the original source cluster and then promoting the replication relationship in the reverse direction. By following these steps, you can ensure that your VM is running on the original source cluster with all changes intact after a disaster recovery event.

Failback for OpenShift Virtualization Disaster Recovery

Using Trident Protect, you can achieve "fail back" after a failover operation by using the following sequence of operations. In this workflow to restore the original replication direction, Trident Protect replicates (resyncs) any application changes back to the original source application before reversing the replication direction.

This process starts from a relationship that has completed a failover to a destination and involves the following steps:

1. Start with a failed over state.
2. Reverse resync the replication relationship

   When you reverse resync a failed over replication relationship, the destination application becomes the source application, and the source becomes the destination. Changes made to the destination application during failover are kept.

a. On the original destination cluster, delete the AppMirrorRelationship CR. This causes the destination to become the source. If there are any protection schedules remaining on the new destination cluster, remove them.

Show example

```
[root@00-50-56-b5-01-5b dr]# oc delete -f app-mirror-relationship.yaml -n test-dr-ns-dest
appmirrorrelationship.protect.trident.netapp.io "amr2" deleted from test-dr-ns-dest namespace
```

b. Create a Snapshot on the new source cluster (original destination) to capture any changes made during the failover period. This ensures that all changes are replicated back to the original source cluster during the resync process.

Show example

```
# snapshot.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  name: test-source-dr-init-snapshot-failback
  namespace: test-dr-ns
spec:
  applicationRef: test-dest-app
  appVaultRef: ontap-s3-appvault
  reclaimPolicy: Delete
```

```
[root@00-50-56-b5-01-5b dr]# cat failback-snapshot.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  name: test-source-dr-init-snapshot-failback
  namespace: test-dr-ns-dest
spec:
  applicationRef: test-dest-app
  appVaultRef: ontap-s3-appvault
  reclaimPolicy: Delete
```

```
[root@00-50-56-b5-01-5b dr]# oc get snapshot -n test-dr-ns-dest
NAME                                    APP            RECLAIM POLICY   STATE     ERROR   AGE
test-source-dr-init-snapshot-failback   test-dest-app   Delete           Running           10s
```

```
[root@00-50-56-b5-01-5b dr]# oc get snapshot -n test-dr-ns-dest
NAME                                    APP            RECLAIM POLICY   STATE       ERROR   AGE
test-source-dr-init-snapshot-failback   test-dest-app   Delete           Completed           8m50s
test-source-dr-init-snapshot-failback-2 test-dest-app   Delete           Completed           32s
```

c. Set up a replication relationship from the new source cluster (original destination) to the original source cluster, configuring values for the reverse direction.

Show example

application-mirror-relationship.yaml

```
apiVersion: protect.trident.netapp.io/v1
kind: AppMirrorRelationship
metadata:
  name: amr1
spec:
  desiredState: Established
  destinationAppVaultRef: ontap-s3-appvault
  destinationApplicationRef: dr-source-vm-failback
  namespaceMapping:
  - destination: test-dr-ns
    source: test-dr-ns-dest
  recurrenceRule: |-
    DTSTART:20240901T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  sourceAppVaultRef: ontap-s3-appvault
  sourceApplicationName: test-dest-app
  sourceApplicationUID: "uid of the app in the original source
cluster"
  storageClassName: "sc-nas"
```

```
[root@00-50-56-b5-2b-9b dr]# oc get amr -n test-dr-ns
NAME   DESIRED STATE   STATE          ERROR   AGE
amr1   Established     Establishing           115s
[root@00-50-56-b5-2b-9b dr]# oc get amr -A
NAMESPACE    NAME   DESIRED STATE   STATE         ERROR   AGE
test-dr-ns   amr1   Established     Established           94s
```

d. Promote the replication relationship from the new source cluster (original destination) to the original source cluster by updating the desiredState field in the AppMirrorRelationship CR to "Promoted".

Show example

```
[root@00-50-56-b5-2b-9b dr]# oc get vm -n test-dr-ns
No resources found in test-dr-ns namespace.
[root@00-50-56-b5-2b-9b dr]# oc patch amr amr1 -n test-dr-ns --type=merge -p '{"spec":{"desiredState":"Promoted"}}'
appmirrorrelationship.protect.trident.netapp.io/amr1 patched
```

You will now see the VM running in the original source cluster. The failback process is now complete. You can now clean up the resources in the original destination cluster. You can also set up a new replication relationship if you want to maintain disaster recovery capabilities after the failback process.

Do not perform a normal resync operation, as this will discard data written to the destination cluster during the fail over procedure.

Copyright information

Trademark information