Data Protection of VMs using Trident Protect

NetApp Solutions

NetApp
December 19, 2024

Table of Contents

Data Protection of VMs using Trident Protect

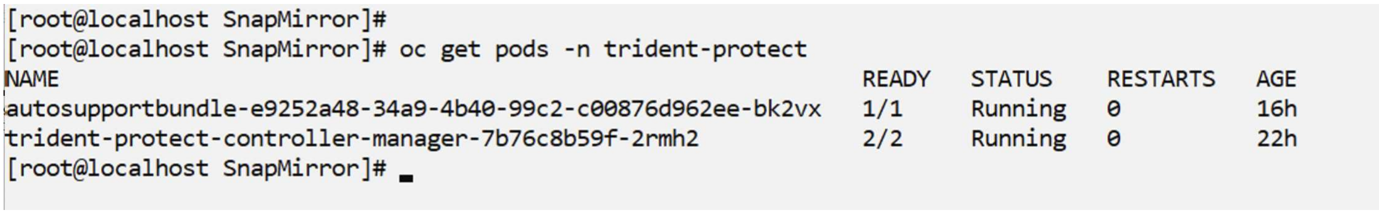
Use Trident protect to implement Failover and Failback for VMs in OpenShift Virtualization

Overview

This section provides details for implementing Failover and Failback of VMs in OpenShift Virtualization using trident protect. The procedures are the same regardless of whether the VMs are on-premises OpenShift clusters or on ROSA clusters.
This section shows the procedures for creating an ontap s3 object storage to use as the appvault for trident protect and create a schedule for app mirror. After that, it shows how to create an app mirror relationship. Finally, it shows how to change state of the app mirror relationship to perform failover and failback.

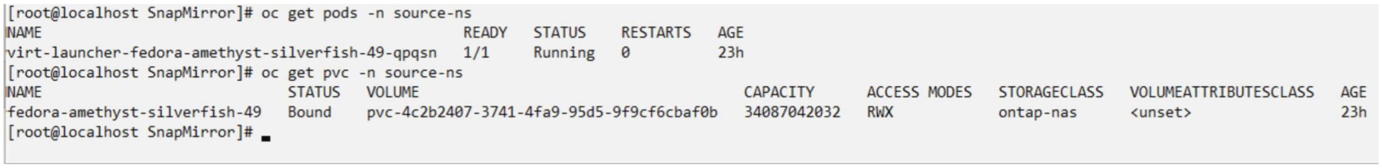
Prerequisites

- Trident must be installed. Backend and storage classes must be created before OpenShift Virtualization is installed on the cluster using the OpenShift Virtualization operator.
- Trident protect must be installed to implement failover and failback operations for the OpenShift VMs. Refer to the instructions here to install trident protect



```
[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]#
```

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.



```
[root@localhost SnapMirror]# oc get pods -n source-ns
NAME                                         READY   STATUS    RESTARTS   AGE
virt-launcher-fedora-amethyst-silverfish-49-qpqsn   1/1     Running   0          23h
[root@localhost SnapMirror]# oc get pvc -n source-ns
NAME                           STATUS   VOLUME                                     CAPACITY      ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
fedora-amethyst-silverfish-49   Bound    pvc-4c2b2407-3741-4fa9-95d5-9f9cf6cbaf0b   34087042032   RWX            ontap-nas      <unset>                 23h
[root@localhost SnapMirror]#
```

Create App Vault using ONTAP S3

This section shows how to set up an app vault in trident protect using ontap S3 Object storage.

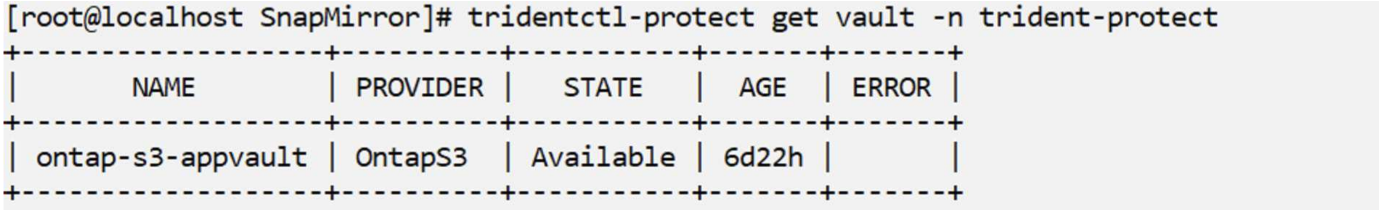
Use oc commands and the yaml files shown below to create a secret and the appvault custom resource for ontap s3. Ensure that you create them in the trident protect namespace.

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

```yaml
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
```

```yaml
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
      endpoint: ""
    gcp:
      bucketName: ""
      projectID: ""
    s3:
      bucketName: trident-protect
      endpoint: <data lif to use to access S3>
      secure: "false"
      skipCertValidation: "true"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-secret
  providerType: OntapS3
```

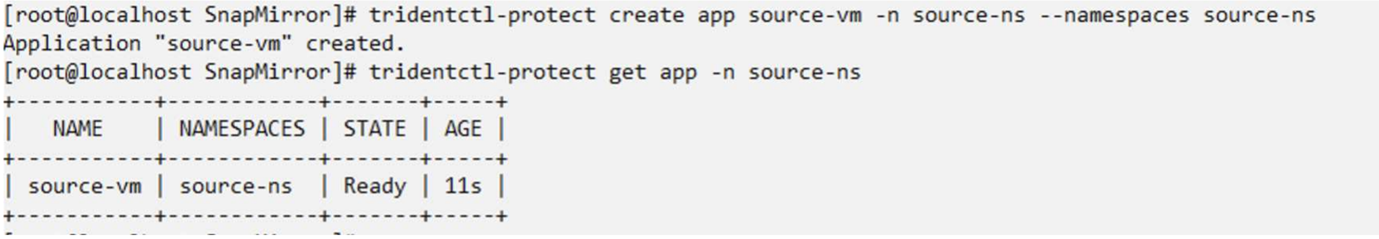
Ensure that ontap S3 vault is created and is in the Available state



```
[root@localhost SnapMirror]# tridentctl-protect get vault -n trident-protect
+------------------+----------+----------+-------+-------+
|       NAME       | PROVIDER |  STATE   |  AGE  | ERROR |
+------------------+----------+----------+-------+-------+
| ontap-s3-appvault | OntapS3  | Available | 6d22h |       |
+------------------+----------+----------+-------+-------+
```

Create a Trident protect app for the VM

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



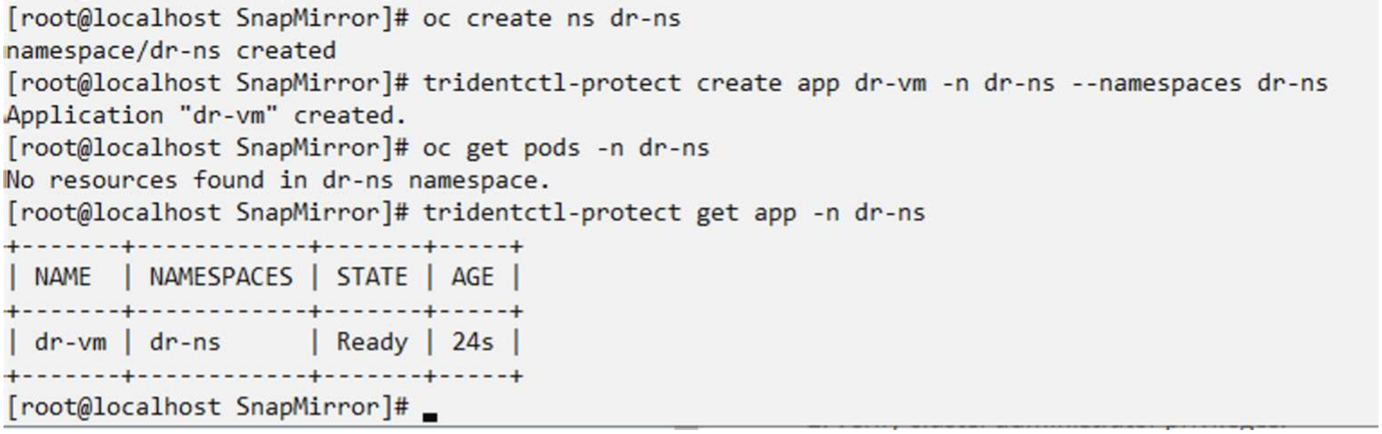
```
[root@localhost SnapMirror]# tridentctl-protect create app source-vm -n source-ns --namespaces source-ns
Application "source-vm" created.
[root@localhost SnapMirror]# tridentctl-protect get app -n source-ns
+-----------+------------+-------+-----+
|   NAME    | NAMESPACES | STATE | AGE |
+-----------+------------+-------+-----+
| source-vm | source-ns  | Ready | 11s |
+-----------+------------+-------+-----+
```

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

```
[root@localhost SnapMirror]# tridentctl-protect create app source-vm -n source-ns --namespaces source-ns
Application "source-vm" created.
[root@localhost SnapMirror]# tridentctl-protect get app -n source-ns
+-----------+------------+-------+-----+
|   NAME    | NAMESPACES | STATE | AGE |
+-----------+------------+-------+-----+
| source-vm | source-ns  | Ready | 11s |
+-----------+------------+-------+-----+
```

Create a Trident protect app for the Disaster Recovery VM in a new namespace

```
oc create ns dr-ns
tridentctl-protect create app dr-vm -n dr-ns --namespaces dr-ns
```



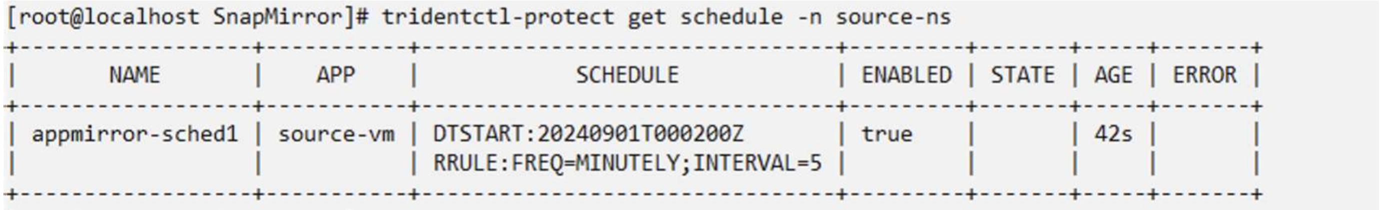
```
[root@localhost SnapMirror]# oc create ns dr-ns
namespace/dr-ns created
[root@localhost SnapMirror]# tridentctl-protect create app dr-vm -n dr-ns --namespaces dr-ns
Application "dr-vm" created.
[root@localhost SnapMirror]# oc get pods -n dr-ns
No resources found in dr-ns namespace.
[root@localhost SnapMirror]# tridentctl-protect get app -n dr-ns
+-------+------------+-------+-----+
| NAME  | NAMESPACES | STATE | AGE |
+-------+------------+-------+-----+
| dr-vm | dr-ns      | Ready | 24s |
+-------+------------+-------+-----+
[root@localhost SnapMirror]#
```

Create an AppMirror Schedule in the source namespace

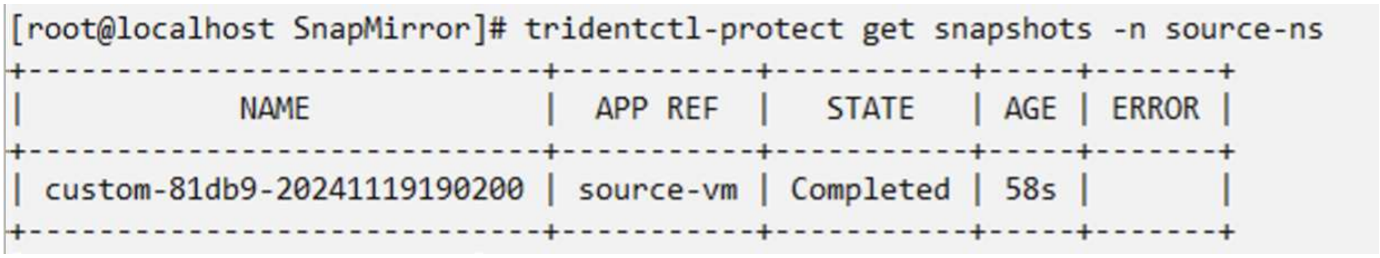
Create a schedule for AppMirror using the yaml as shown. This will create snapshots using the schedule (every 5 minutes) and retain 2 snapshots

```
oc create -f appmirror-schedule.yaml -n source-ns
```

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



```
[root@localhost SnapMirror]# tridentctl-protect get schedule -n source-ns
+------------------+----------+--------------------------------+---------+-------+-----+-------+
|       NAME       |   APP    |            SCHEDULE            | ENABLED | STATE | AGE | ERROR |
+------------------+----------+--------------------------------+---------+-------+-----+-------+
| appmirror-sched1 | source-vm | DTSTART:20240901T000200Z      | true    |       | 42s |       |
|                  |          | RRULE:FREQ=MINUTELY;INTERVAL=5 |         |       |     |       |
+------------------+----------+--------------------------------+---------+-------+-----+-------+
```



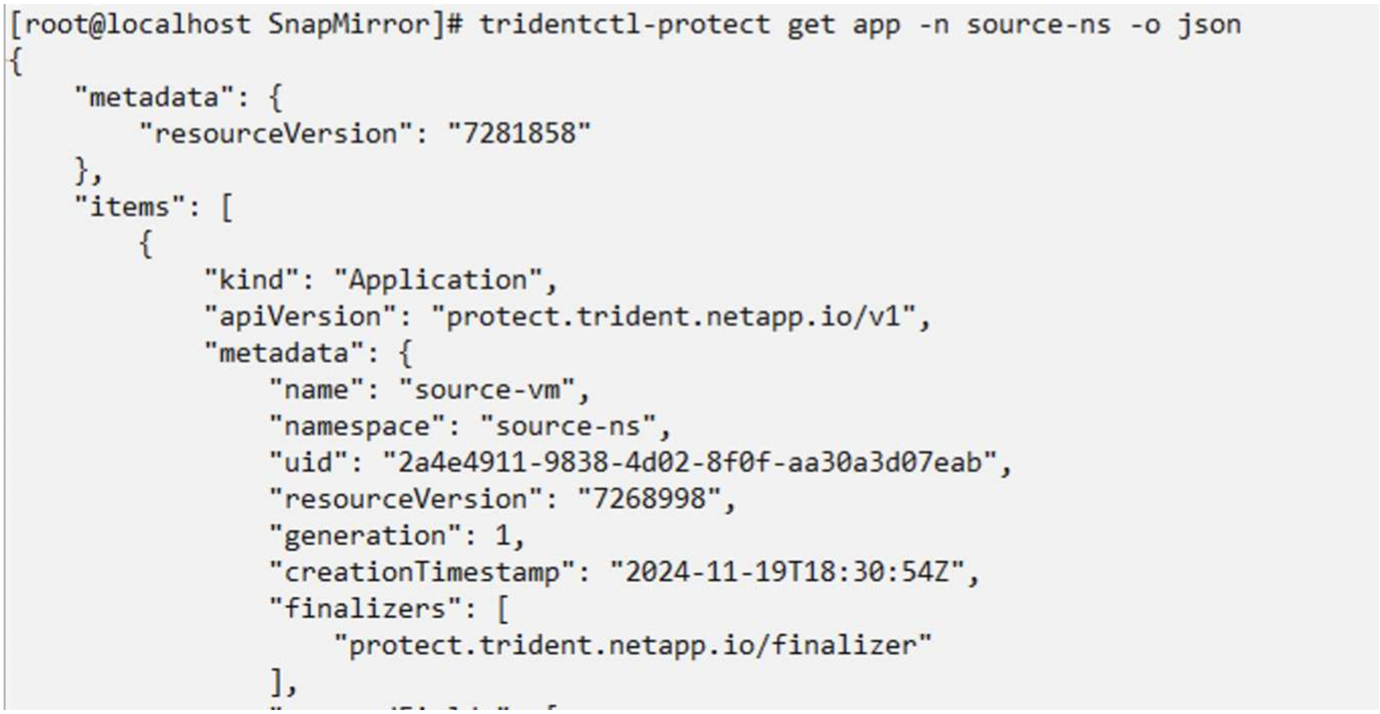
```
[root@localhost SnapMirror]# tridentctl-protect get snapshots -n source-ns
+-----------------------------+-----------+-----------+-----+-------+
|            NAME             |  APP REF  |   STATE   | AGE | ERROR |
+-----------------------------+-----------+-----------+-----+-------+
| custom-81db9-20241119190200 | source-vm | Completed | 58s |       |
+-----------------------------+-----------+-----------+-----+-------+
```

Create an appMirror relationship in the DR namespace

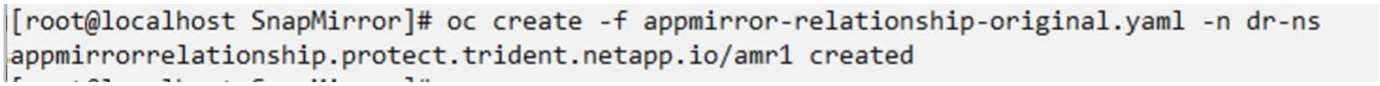
Create an Appmirror relationship in the Disaster Recovery namespace. Set the desiredState to Established.

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

ⓘ You can get the application UID of the source VM from the json output of the source app as shown below

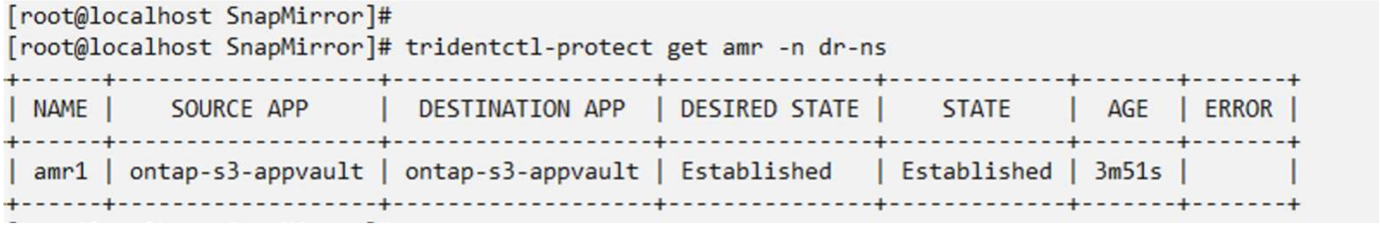


```
[root@localhost SnapMirror]# tridentctl-protect get app -n source-ns -o json
{
    "metadata": {
        "resourceVersion": "7281858"
    },
    "items": [
        {
            "kind": "Application",
            "apiVersion": "protect.trident.netapp.io/v1",
            "metadata": {
                "name": "source-vm",
                "namespace": "source-ns",
                "uid": "2a4e4911-9838-4d02-8f0f-aa30a3d07eab",
                "resourceVersion": "7268998",
                "generation": 1,
                "creationTimestamp": "2024-11-19T18:30:54Z",
                "finalizers": [
                    "protect.trident.netapp.io/finalizer"
                ],
```

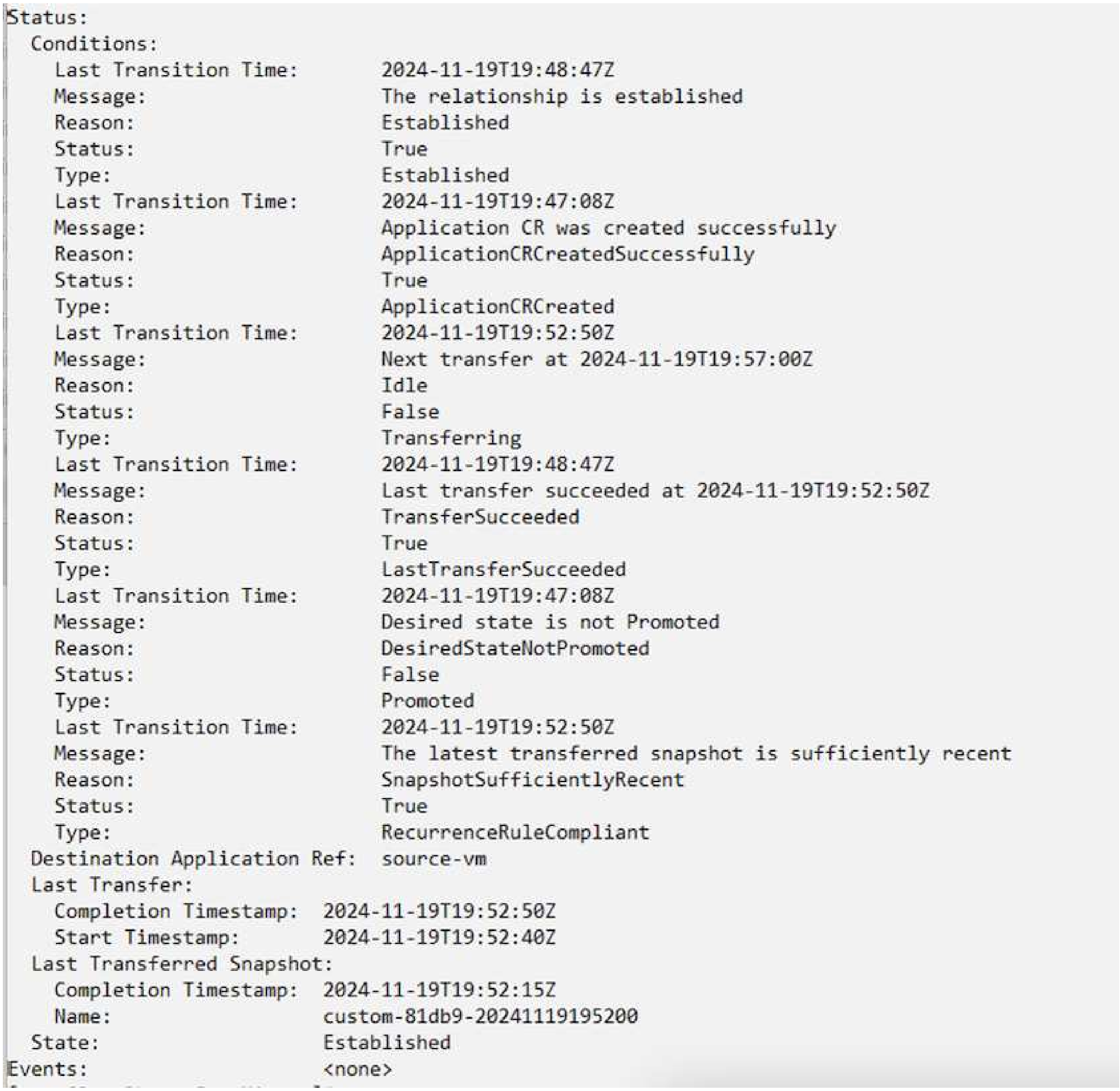


```
[root@localhost SnapMirror]# oc create -f appmirror-relationship-original.yaml -n dr-ns
appmirrorrelationship.protect.trident.netapp.io/amr1 created
```

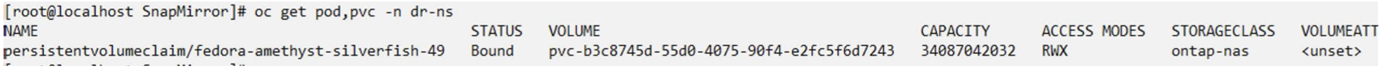
When the AppMirror relationship is established, the most recent snapshot is transferred to the destination namespace. The PVC is created for the VM in the dr namespace, however, the VM pod is not yet created in the dr namespace.



```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# tridentctl-protect get amr -n dr-ns
+------+------------------+------------------+---------------+-------------+-------+-------+
| NAME |    SOURCE APP    | DESTINATION APP  | DESIRED STATE |    STATE    |  AGE  | ERROR |
+------+------------------+------------------+---------------+-------------+-------+-------+
| amr1 | ontap-s3-appvault | ontap-s3-appvault | Established   | Established | 3m51s |       |
+------+------------------+------------------+---------------+-------------+-------+-------+
```



```
Status:
  Conditions:
    Last Transition Time:       2024-11-19T19:48:47Z
    Message:                    The relationship is established
    Reason:                     Established
    Status:                     True
    Type:                       Established
    Last Transition Time:       2024-11-19T19:47:08Z
    Message:                    Application CR was created successfully
    Reason:                     ApplicationCRCreatedSuccessfully
    Status:                     True
    Type:                       ApplicationCRCreated
    Last Transition Time:       2024-11-19T19:52:50Z
    Message:                    Next transfer at 2024-11-19T19:57:00Z
    Reason:                     Idle
    Status:                     False
    Type:                       Transferring
    Last Transition Time:       2024-11-19T19:48:47Z
    Message:                    Last transfer succeeded at 2024-11-19T19:52:50Z
    Reason:                     TransferSucceeded
    Status:                     True
    Type:                       LastTransferSucceeded
    Last Transition Time:       2024-11-19T19:47:08Z
    Message:                    Desired state is not Promoted
    Reason:                     DesiredStateNotPromoted
    Status:                     False
    Type:                       Promoted
    Last Transition Time:       2024-11-19T19:52:50Z
    Message:                    The latest transferred snapshot is sufficiently recent
    Reason:                     SnapshotSufficientlyRecent
    Status:                     True
    Type:                       RecurrenceRuleCompliant
  Destination Application Ref:  source-vm
  Last Transfer:
    Completion Timestamp:  2024-11-19T19:52:50Z
    Start Timestamp:       2024-11-19T19:52:40Z
  Last Transferred Snapshot:
    Completion Timestamp:  2024-11-19T19:52:15Z
    Name:                  custom-81db9-20241119195200
  State:                   Established
Events:                    <none>
```

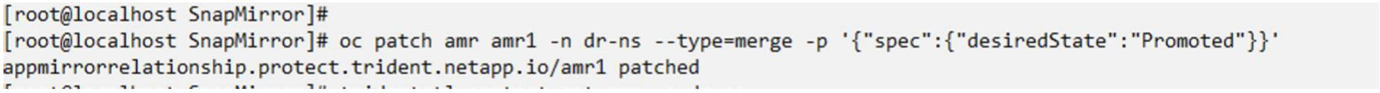


```
[root@localhost SnapMirror]# oc get pod,pvc -n dr-ns
NAME                                                 STATUS   VOLUME                                     CAPACITY      ACCESS MODES   STORAGECLASS   VOLUMEATT
persistentvolumeclaim/fedora-amethyst-silverfish-49  Bound    pvc-b3c8745d-55d0-4075-90f4-e2fc5f6d7243   34087042032   RWX            ontap-nas      <unset>
```

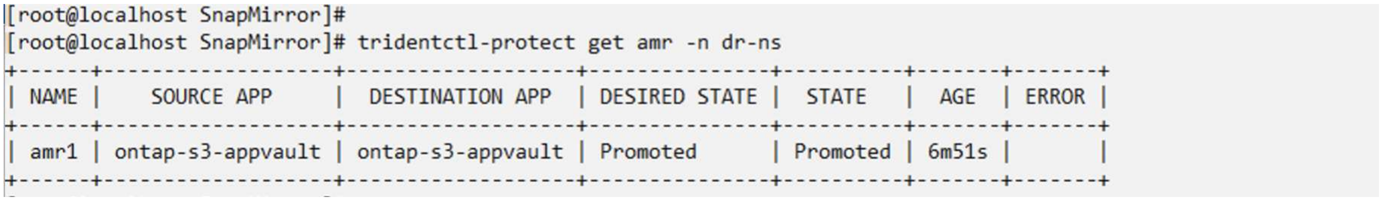
Promote the relationship to Failover

Change the desired state of the relationship to "Promoted" to create the VM in the DR namespace. The VM is still running in the source namespace.

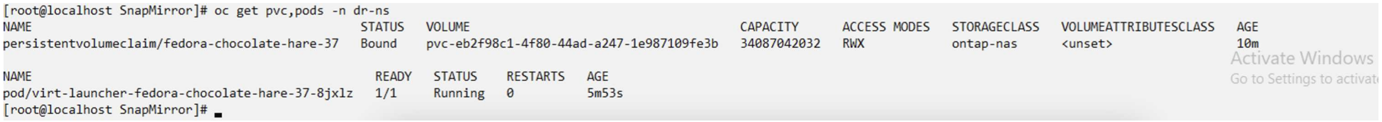
```
oc patch amr amr1 -n dr-ns --type=merge -p
'{"spec":{"desiredState":"Promoted"}}'
```



```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# oc patch amr amr1 -n dr-ns --type=merge -p '{"spec":{"desiredState":"Promoted"}}'
appmirrorrelationship.protect.trident.netapp.io/amr1 patched
```

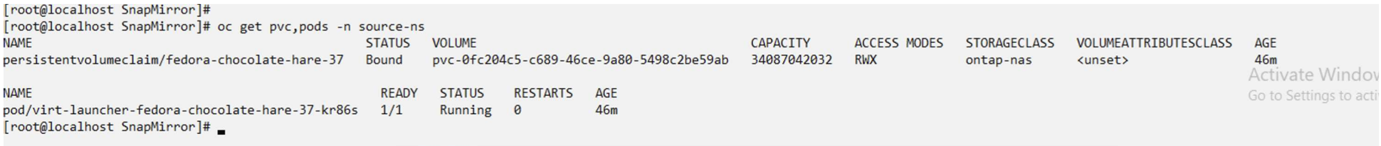


```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# tridentctl-protect get amr -n dr-ns
+------+-------------------+-------------------+---------------+----------+-------+-------+
| NAME |    SOURCE APP     |  DESTINATION APP  | DESIRED STATE |  STATE   |  AGE  | ERROR |
+------+-------------------+-------------------+---------------+----------+-------+-------+
| amr1 | ontap-s3-appvault | ontap-s3-appvault | Promoted      | Promoted | 6m51s |       |
+------+-------------------+-------------------+---------------+----------+-------+-------+
```



```
[root@localhost SnapMirror]# oc get pvc,pods -n dr-ns
NAME                                         STATUS   VOLUME                                     CAPACITY      ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/fedora-chocolate-hare-37   Bound    pvc-eb2f98c1-4f80-44ad-a247-1e987109fe3b   34087042032   RWX            ontap-nas      <unset>                 10m

NAME                                          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-fedora-chocolate-hare-37-8jxlz   1/1     Running   0          5m53s
[root@localhost SnapMirror]#
```



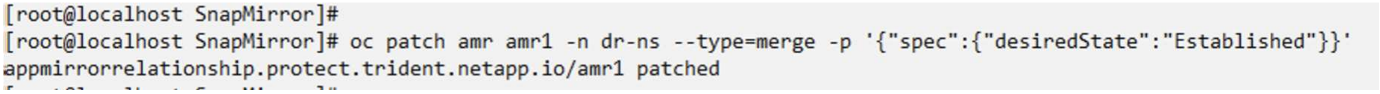
```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# oc get pvc,pods -n source-ns
NAME                                         STATUS   VOLUME                                     CAPACITY      ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/fedora-chocolate-hare-37   Bound    pvc-0fc204c5-c689-46ce-9a80-5498c2be59ab   34087042032   RWX            ontap-nas      <unset>                 46m

NAME                                          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-fedora-chocolate-hare-37-kr86s   1/1     Running   0          46m
[root@localhost SnapMirror]#
```

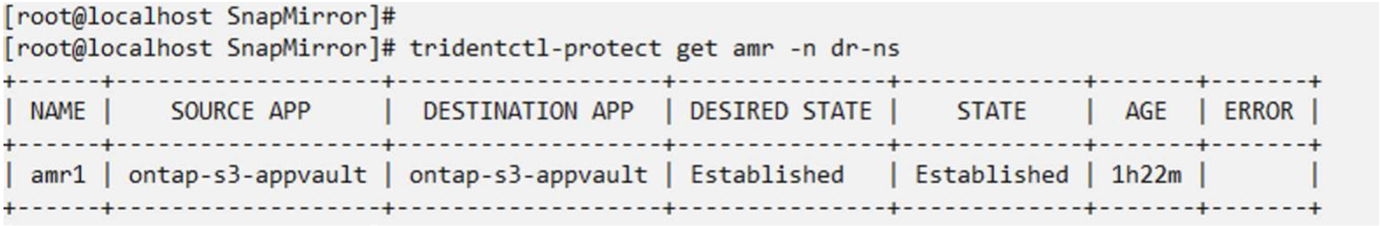
Establish the relationship again to Failback

Change the desired state of the relationship to "Established". The VM is deleted in the DR namespace. The pvc still exists in the DR namespace. The VM is still running in the source namespace. The original relationship from source namespace to DR ns is established. .

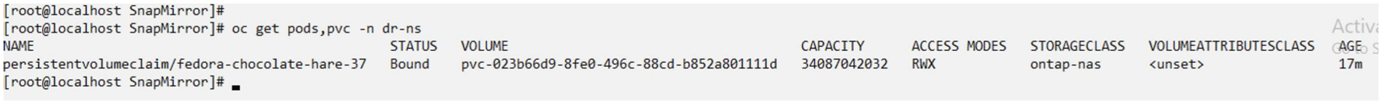
```
oc patch amr amr1 -n dr-ns --type=merge -p
'{"spec":{"desiredState":"Established"}}'
```



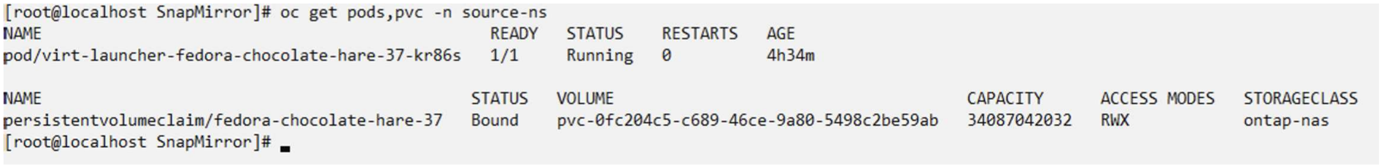
```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# oc patch amr amr1 -n dr-ns --type=merge -p '{"spec":{"desiredState":"Established"}}'
appmirrorrelationship.protect.trident.netapp.io/amr1 patched
```



```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# tridentctl-protect get amr -n dr-ns
+------+-------------------+-------------------+---------------+-------------+-------+-------+
| NAME |    SOURCE APP     |  DESTINATION APP  | DESIRED STATE |    STATE    |  AGE  | ERROR |
+------+-------------------+-------------------+---------------+-------------+-------+-------+
| amr1 | ontap-s3-appvault | ontap-s3-appvault | Established   | Established | 1h22m |       |
+------+-------------------+-------------------+---------------+-------------+-------+-------+
```



```
[root@localhost SnapMirror]#
[root@localhost SnapMirror]# oc get pods,pvc -n dr-ns
NAME                                          STATUS   VOLUME                                     CAPACITY      ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/fedora-chocolate-hare-37   Bound    pvc-023b66d9-8fe0-496c-88cd-b852a801111d   34087042032   RWX            ontap-nas      <unset>                 17m
[root@localhost SnapMirror]#
```



```
[root@localhost SnapMirror]# oc get pods,pvc -n source-ns
NAME                                         READY   STATUS    RESTARTS   AGE
pod/virt-launcher-fedora-chocolate-hare-37-kr86s   1/1     Running   0          4h34m

NAME                                          STATUS   VOLUME                                     CAPACITY      ACCESS MODES   STORAGECLASS
persistentvolumeclaim/fedora-chocolate-hare-37   Bound    pvc-0fc204c5-c689-46ce-9a80-5498c2be59ab   34087042032   RWX            ontap-nas
[root@localhost SnapMirror]#
```

Copyright information

Trademark information