Notfallwiederherstellung

NetApp virtualization solutions

NetApp
July 21, 2026

Inhalt

Notfallwiederherstellung

Trident Protect/AppMirror Voraussetzungen für die OpenShift Virtualisierungs-Notfallwiederherstellung

Dieser Abschnitt enthält Informationen zu den Voraussetzungen für die Verwendung von NetApp Trident Protect und AppMirror zur Notfallwiederherstellung von OpenShift Virtualisierungs-Workloads. Dies umfasst die notwendigen Clusterkonfigurationen, Speicheranforderungen und Installationsschritte, um Trident Protect und AppMirror für eine effektive Notfallwiederherstellung von VMs, die auf OpenShift Virtualisierung ausgeführt werden, einzurichten.

Voraussetzungen

1. Ein Red Hat OpenShift Cluster mit installierter und konfigurierter OpenShift Virtualisierung. Anweisungen zur Installation finden sich im "Dokumentation hier".
2. Zwei NetApp ONTAP Speichersysteme mit jeweils einer konfigurierten SVM, um Speicher für die aktiven und die Disaster Recovery OpenShift Cluster bereitzustellen. Jede SVM muss mindestens eine Management-LIF und eine Daten-LIF konfiguriert und vom jeweiligen OpenShift Cluster aus zugänglich haben.
3. Cluster-Peering und SVM-Peering wurden zwischen den beiden ONTAP-Systemen konfiguriert, um die Replikation von Daten von der SVM des Quell-Clusters zur SVM des Ziel-Clusters zu ermöglichen. Anleitungen zur Einrichtung von Cluster-Peering und SVM-Peering sind in der "ONTAP Cluster-Peering-Dokumentation hier" zu finden.
4. NetApp Trident wurde auf dem OpenShift Cluster installiert und konfiguriert, um persistenten Speicher von der ONTAP SVM bereitzustellen. Installationsanweisungen finden sich in der "Trident Dokumentation hier".
5. Trident Backend-Konfiguration für die ONTAP SVM erstellt und eine StorageClass auf dem OpenShift Cluster konfiguriert, sodass dieses Trident Backend als Provisioner verwendet wird. Anleitungen zum Erstellen einer Trident Backend-Konfiguration und einer StorageClass finden sich in der "Trident Backend-Konfigurationsdokumentation hier".
6. Cluster-Administratorzugriff auf den OpenShift Cluster und Administratorzugriff auf das ONTAP Storage-System, um die erforderlichen Konfigurationen vorzunehmen.
7. Eine Administrator-Workstation mit `tridentctl` und `oc` installierten Tools, die der $PATH Umgebungsvariablen hinzugefügt wurden, um mit dem OpenShift Cluster und Trident zu interagieren.
8. Ausreichende Hardware-Ressourcen auf dem OpenShift Cluster zur Unterstützung des OpenShift Virtualization Operators und der bereitzustellenden VMs. Weitere Details zu den Ressourcenanforderungen finden sich in der "Dokumentation hier".
9. Optional können Knotenplatzierungsregeln für OpenShift Virtualization konfiguriert werden, um eine Teilmenge der Cluster-Knoten für die OpenShift Virtualization Operatoren, Controller und VMs festzulegen. Anweisungen zur Konfiguration von Knotenplatzierungsregeln finden sich in der "Dokumentation hier".
10. Für die Speicherunterstützung von OpenShift Virtualization wird empfohlen, eine dedizierte StorageClass zu verwenden, die Speicher von einem bestimmten Trident Backend anfordert, das wiederum von einer dedizierten SVM unterstützt wird. Dadurch bleibt ein Maß an Mandantenfähigkeit in Bezug auf die Daten erhalten, die für VM-basierte Workloads im OpenShift Cluster bereitgestellt werden.
11. Eine VM muss in OpenShift Virtualization verfügbar sein. Einzelheiten zum Bereitstellen einer neuen VM oder zum Migrieren einer vorhandenen VM in OpenShift Virtualization finden Sie im entsprechenden

Abschnitt der Dokumentation.

Installation von NetApp Trident Protect für OpenShift Virtualization

Um die Notfallwiederherstellungsfunktionen für OpenShift Virtualisierungs-Workloads mit NetApp Trident Protect zu ermöglichen, muss Trident Protect auf dem OpenShift Cluster installiert und konfiguriert sein. Trident Protect kann mithilfe eines Helm-Charts installiert werden.

Die Installation von Trident Protect muss nach der Installation von Trident erfolgen.

Vollständige Installationsanweisungen sind in der "Trident Protect Dokumentation hier" zu finden.

Beispiel anzeigen


```
[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]#
```

Quell-Cluster-Einrichtung für Red Hat OpenShift Virtualization Disaster Recovery

Dieser Abschnitt enthält Informationen zur Einrichtung des Quell-Cluster für die Notfallwiederherstellung von OpenShift Virtualization-Workloads mit NetApp Trident Protect und AppMirror. Dazu gehört die Erstellung eines AppVault mit ONTAP S3, die Erstellung einer Trident Protect Application für die Quell-VM sowie die Erstellung von bedarfsgesteuerten und geplanten Snapshots der VM mit Trident Protect. Diese Schritte sind unerlässlich, damit der Quell-Cluster korrekt konfiguriert ist, um Daten in den Ziel-Cluster für die Notfallwiederherstellung zu replizieren und die Wiederherstellung von VMs im Katastrophenfall zu ermöglichen.

Einrichtung des Quell-Clusters für die Notfallwiederherstellung für OpenShift Virtualisierung

Alle nachfolgenden Schritte werden auf dem Quell-Cluster ausgeführt, also dem Cluster, auf dem die geschützte VM aktuell läuft.

Erstellen Sie AppVault mit ONTAP S3

In diesem Abschnitt wird gezeigt, wie ein AppVault-Objekt in Trident Protect mit ONTAP S3 Object Storage eingerichtet wird. Das in diesem Schritt erstellte AppVault-Objekt wird zur Speicherung replizierter Daten aus dem Quell-Cluster verwendet, und diese Daten werden für die Wiederherstellung im Ziel-Cluster für die Notfallwiederherstellung genutzt.

Verwenden Sie die oc Befehle und die unten aufgeführten yaml-Dateien, um ein Secret und die AppVault benutzerdefinierte Ressource für ONTAP s3 zu erstellen. Diese sind im Namespace trident-protect zu erstellen.

Beispiel anzeigen

```
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
```

Stellen Sie sicher, dass ONTAP S3 vault erstellt wurde und sich im Status Verfügbar befindet

```
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
```

Eine Trident Protect App für die Quell-VM erstellen

Eine benutzerdefinierte Anwendungsressource im Namespace erstellen, in dem sich die Quell-VM befindet.

Beispiel anzeigen

```
# 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
```

Die VM und ihr PVC im Namespace, der in der benutzerdefinierten App-Ressource enthalten ist, sind unten dargestellt.

```
[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   341144990004 RWX            sc-nas         <unset>                 25h
```

Einen Snapshot der VM mit Trident Protect erstellen

Sie können mithilfe der Trident Protect CLI oder API einen bedarfsgesteuerten Snapshot der VM erstellen. Dieser Snapshot wird im AppVault gespeichert, das im vorherigen Schritt erstellt wurde, und kann zur

Wiederherstellung im Ziel-Cluster für die Notfallwiederherstellung verwendet werden.

Beispiel anzeigen

```
# 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
```

Einen geplanten Snapshot der VM mit Trident Protect erstellen

Es besteht auch die Möglichkeit, einen geplanten Snapshot zu erstellen, indem eine benutzerdefinierte Ressource „Schedule“ im selben Namespace wie die benutzerdefinierte Ressource „Application“ erstellt wird. Der Zeitplan erstellt automatisch Snapshots der VM zum angegebenen Zeitpunkt und speichert sie im AppVault.

Beispiel anzeigen

```
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"
```

Ziel-Cluster-Einrichtung für Red Hat OpenShift Virtualization Disaster Recovery

Dieser Abschnitt enthält Informationen zur Einrichtung des Ziel-Clusters für die Notfallwiederherstellung von OpenShift Virtualization-Workloads mit NetApp Trident Protect und AppMirror. Dies umfasst die Erstellung eines AppVault mit ONTAP S3 und die Erstellung eines neuen Namespace für die Ziel-VM. Diese Schritte sind unerlässlich, um sicherzustellen, dass der Ziel-Cluster ordnungsgemäß konfiguriert ist, um replizierte Daten vom Quell-Cluster zu empfangen und die Wiederherstellung von VMs im Katastrophenfall zu ermöglichen.

1. Ein AppVault Objekt im Ziel-Cluster erstellen. Dieses sollte auf dasselbe S3 wie der Quell-Cluster verweisen, damit der Snapshot aus dem S3 abgerufen werden kann, in dem der Quell-Cluster ihn abgelegt hat.

Beispiel anzeigen

```
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. Erstellen Sie einen neuen Namespace im Ziel-Cluster, um die übernommene VM zu hosten. Erstellen Sie eine App im Ziel-Cluster, die auf den in Schritt 1 erstellten AppVault und den neuen Namespace verweist. Diese App wird verwendet, um die AppMirror-Beziehung zwischen Quell- und Ziel-Cluster herzustellen und die Replikation der Daten vom Quell-Cluster zum Ziel-Cluster zu verwalten.

Beispiel anzeigen

```
# 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 einer VM in OpenShift Virtualization Disaster Recovery

In diesem Abschnitt wird beschrieben, wie ein Failover einer VM von einem Quell-Cluster auf einen Disaster-Recovery-Ziel-Cluster mithilfe von NetApp Trident Protect und AppMirror durchgeführt werden kann. Dies ist hilfreich zum Testen des Failover-Prozesses sowie für geplante Wartungsarbeiten am Quell-Cluster. Mit diesen Schritten kann sichergestellt werden, dass Ihre VM nach einem Disaster-Recovery-Ereignis mit allen Änderungen im Disaster-Recovery-Ziel-Cluster ausgeführt wird.

Ausfallsicherung ohne Standortausfall für OpenShift Virtualisierung

In diesem Abschnitt wird beschrieben, wie ein Failover einer VM von einem Quell-Cluster ohne Standortausfall durchgeführt werden kann. Dies ist hilfreich, um den Failover-Prozess zu testen und geplante

Wartungsarbeiten am Quell-Cluster durchzuführen. +

Um ein Failover ohne Standortausfall durchzuführen, muss zunächst sichergestellt werden, dass der Quell-Cluster und der Ziel-Cluster wie in den vorherigen Abschnitten beschrieben für die Notfallwiederherstellung eingerichtet sind. Anschließend sind die folgenden Schritte auszuführen:

1. Die AppMirror-Beziehung zwischen dem Quell-Cluster und dem Ziel-Cluster wird hergestellt. Wenn die AppMirror-Beziehung hergestellt ist, wird der aktuellste Snapshot in den Ziel-Namespace übertragen. Der PVC wird für die VM im Ziel-Namespace erstellt, jedoch wird das VM-Pod im Ziel-Namespace noch nicht erstellt.

   Beispiel anzeigen

```
# 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
```

> Die AppMirror-Beziehung sollte über die CLI des Ziel-Cluster erstellt werden. Sie sollte mit dem gewünschten Status „Established“ erstellt werden. So wird sichergestellt, dass die neuesten Snapshot-Daten an den Ziel-Cluster repliziert werden.

2. Die AppMirror-Beziehung wird hochgestuft. Der gewünschte Status der Beziehung wird auf „Hochgestuft“ geändert, um die VM im Ziel-Namespace zu erstellen. Die VM läuft weiterhin im Quell-Namespace. Die neuesten Snapshot-Daten werden in den Ziel-Cluster repliziert und die VM wird im Ziel-Cluster gestartet. Mit dieser Methode lässt sich ein Failover testen und die VM-Verfügbarkeit aufrechterhalten, ohne dass der gesamte Quell-Cluster außer Betrieb genommen werden muss.

Beispiel anzeigen

```
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 einer VM in OpenShift Virtualization Disaster Recovery

Dieser Abschnitt enthält Informationen über den Failback-Prozess für eine VM in OpenShift Virtualization Disaster Recovery unter Verwendung von NetApp Trident Protect und AppMirror. Nach einer Failover-Operation zum Wechsel auf einen Disaster Recovery Ziel-Cluster kann der Failback-Prozess genutzt werden, um die Replikationsrichtung umzukehren und den ursprünglichen Quell-Cluster als primären Cluster für die VM wiederherzustellen. Dabei werden alle während des Failover-Zeitraums an der VM vorgenommenen Änderungen zurück auf den ursprünglichen Quell-Cluster synchronisiert und anschließend die Replikationsbeziehung in umgekehrter Richtung aktiviert. Durch diese Vorgehensweise wird sichergestellt, dass die VM nach einem Disaster Recovery Ereignis auf dem ursprünglichen Quell-Cluster mit allen Änderungen ausgeführt wird.

Failback für OpenShift Virtualisierung Notfallwiederherstellung

Mit Trident Protect können Sie „Failback“ nach einem Failover-Vorgang durchführen, indem Sie die folgende Abfolge von Schritten ausführen. In diesem Workflow zur Wiederherstellung der ursprünglichen Replikationsrichtung repliziert (resynchronisiert) Trident Protect alle Anwendungsänderungen zurück in die ursprüngliche Quellanwendung, bevor die Replikationsrichtung umgekehrt wird.

Dieser Prozess beginnt mit einer Beziehung, die einen Failover auf ein Ziel abgeschlossen hat und umfasst die folgenden Schritte:

1. Beginnen Sie mit einem übergewechselten Zustand.

2. Die Replikationsbeziehung umgekehrt resynchronisieren

   Wenn Sie eine fehlgeschlagene Replikationsbeziehung rücksynchronisieren, wird die Zielanwendung zur Quellanwendung und die Quellanwendung zur Zielanwendung. Änderungen, die während des Failovers an der Zielanwendung vorgenommen wurden, bleiben erhalten.

   a. Löschen Sie auf dem ursprünglichen Ziel-Cluster die AppMirrorRelationship CR. Dadurch wird das Ziel-Cluster zum Quell-Cluster. Wenn auf dem neuen Ziel-Cluster noch Datensicherungsstrategien vorhanden sind, entfernen Sie diese.

      Beispiel anzeigen

      ```
      [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. Auf dem neuen Quell-Cluster (ursprüngliches Ziel) wird ein Snapshot erstellt, um alle während des Failover-Zeitraums vorgenommenen Änderungen zu erfassen. So wird sichergestellt, dass alle Änderungen während des Resync-Prozesses zurück auf den ursprünglichen Quell-Cluster repliziert werden.

Beispiel anzeigen

```
# 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. Eine Replikationsbeziehung vom neuen Quell-Cluster (ursprüngliches Ziel) zum ursprünglichen Quell-Cluster einrichten und Werte für die umgekehrte Richtung konfigurieren.

Beispiel anzeigen

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. Die Replikationsbeziehung vom neuen Quell-Cluster (ursprüngliches Ziel) zum ursprünglichen Quell-Cluster wird durch Aktualisierung des desiredState-Felds in der AppMirrorRelationship-CR auf „Promoted“ hochgestuft.

Beispiel anzeigen

```
[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
```

Die VM läuft nun im ursprünglichen Quell-Cluster. Der Failback-Prozess ist jetzt abgeschlossen. Nun können die Ressourcen im ursprünglichen Ziel-Cluster bereinigt werden. Es besteht außerdem die Möglichkeit, eine neue Replikationsbeziehung einzurichten, wenn nach dem Failback-Prozess weiterhin Disaster-Recovery-Funktionen erhalten bleiben sollen.

Führen Sie keine normale Resynchronisierungsoperation durch, da dadurch Daten, die während des Failover-Vorgangs auf den Ziel-Cluster geschrieben wurden, verworfen werden.