



Schützen Sie Daten mit Trident Protect

NetApp public and hybrid cloud solutions

NetApp
February 04, 2026

Inhalt

Schützen Sie Daten mit Trident Protect	1
Datenschutz für Container-Apps in der OpenShift Container Platform mit Trident Protect	1
Schützen Sie VMs in Red Hat OpenShift Virtualization mit Trident Protect	14
App Vault für Objektspeicher erstellen	15
Erstellen einer VM in OpenShift Virtualization	17
Erstellen einer App	21
Schützen Sie die App, indem Sie ein Backup erstellen	22
Wiederherstellen aus einer Sicherung	25
Schützen Sie die App mit Snapshots	29
Wiederherstellen aus Snapshot	32
Wiederherstellen einer bestimmten VM	36
Videodemonstration	41

Schützen Sie Daten mit Trident Protect

Datenschutz für Container-Apps in der OpenShift Container Platform mit Trident Protect

Dieser Abschnitt des Referenzdokuments enthält Details zum Erstellen von Snapshots und Backups von Container-Apps mit Trident Protect. NetApp Trident Protect bietet erweiterte Funktionen zur Anwendungsdatenverwaltung, die die Funktionalität und Verfügbarkeit von Stateful-Kubernetes-Anwendungen verbessern, die von NetApp ONTAP Speichersystemen und dem NetApp Trident CSI-Speicherbereitsteller unterstützt werden. Trident Protect erstellt Anwendungs-Snapshots und -Backups. Dies bedeutet, dass nicht nur Snapshots und Backups von Anwendungsdaten in persistenten Volumes erstellt werden, sondern auch Snapshots und Backups von Anwendungsmetadaten. Die von Trident Protect erstellten Snapshots und Backups können in jedem der folgenden Objektspeicher gespeichert und zu einem späteren Zeitpunkt von dort wiederhergestellt werden.

- AWS S3
- Azure Blob-Speicher
- Google Cloud-Speicher
- Ontap S3
- StorageGrid
- jeder andere S3-kompatible Speicher

Trident Protect verwendet das Kubernetes-Modell der rollenbasierten Zugriffskontrolle (RBAC). Standardmäßig stellt Trident Protect einen einzelnen Systemnamespace namens „Trident-Protect“ und das zugehörige Standarddienstkonto bereit. Wenn Ihre Organisation viele Benutzer oder spezielle Sicherheitsanforderungen hat, können Sie die RBAC-Funktionen von Trident Protect nutzen, um eine genauere Kontrolle über den Zugriff auf Ressourcen und Namespaces zu erhalten.

Weitere Informationen zu RBAC in Trident Protect finden Sie im "[Trident Protect-Dokumentation](#)"



Der Clusteradministrator hat Zugriff auf Ressourcen im Standard-Trident-Protect-Namespace und kann auch auf Ressourcen in allen anderen Namespaces zugreifen. Benutzer können im Trident-Protect-Namespace keine benutzerdefinierten Ressourcen (CRs) für die Anwendungsdatenverwaltung wie Snapshot- und Backup-CRs erstellen. Als bewährte Methode müssen Benutzer diese CRs im Anwendungsnamespace erstellen.

Trident Protect kann gemäß den Anweisungen in der Dokumentation installiert werden "[hier](#)." In diesem Abschnitt wird der Workflow für den Datenschutz von Containeranwendungen und die Wiederherstellung der Anwendungen mit Trident Protect gezeigt. 1. Snapshot-Erstellung (auf Anfrage und nach Zeitplan) 2. Wiederherstellen aus Snapshot (Wiederherstellung im gleichen und anderen Namespace) 3. Backup-Erstellung 4. Wiederherstellen aus einer Sicherung

Voraussetzung

Bevor die Snapshots und Backups für eine Anwendung erstellt werden, muss in Trident Protect ein Objektspeicher konfiguriert werden, um die Snapshots und Backups zu speichern. Dies geschieht mithilfe des Bucket CR. Nur Administratoren können einen Bucket CR erstellen und konfigurieren. Der Bucket CR ist in Trident Protect als AppVault bekannt. AppVault-Objekte sind die deklarative Kubernetes-Workflow-Darstellung eines Speicher-Buckets. Ein AppVault CR enthält die Konfigurationen, die für die Verwendung eines Buckets in Schutzvorgängen wie Backups, Snapshots, Wiederherstellungsvorgängen und SnapMirror Replikation erforderlich sind.

In diesem Beispiel zeigen wir die Verwendung von ONTAP S3 als Objektspeicher. Hier ist der Workflow zum Erstellen von AppVault CR für ONTAP S3: 1. Erstellen Sie einen S3-Objektspeicherserver in der SVM im ONTAP -Cluster. 2. Erstellen Sie einen Bucket im Object Store Server. 3. Erstellen Sie einen S3-Benutzer in der SVM. Bewahren Sie den Zugangsschlüssel und den geheimen Schlüssel an einem sicheren Ort auf. 4. Erstellen Sie in OpenShift ein Geheimnis zum Speichern der ONTAP S3-Anmeldeinformationen. 5. Erstellen eines AppVault-Objekts für ONTAP S3

Konfigurieren Sie Trident Protect AppVault für ONTAP S3

Beispiel-YAML-Datei zum Konfigurieren von Trident Protect mit ONTAP S3 als AppVault

```
# alias tp='tridentctl-protect'

appvault-secret.yaml

apiVersion: v1
stringData:
  accessKeyID: "<access key id created for a user to access ONTAP S3
bucket>"
  secretAccessKey: "corresponding Secret Access Key"
#data:
# base 64 encoded values
#  accessKeyID: <base64 access key id created for a user to access
ONTAP S3 bucket>
#  secretAccessKey: <base 64  Secret Access Key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
  type: Opaque

appvault.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: <bucket-name for storing the snapshots and backups>
    endpoint: <endpoint IP for S3>
    secure: "false"
    skipCertValidation: "true"

providerCredentials:
  accessKeyID:
    valueFromSecret:
      key: accessKeyID
      name: appvault-secret

  secretAccessKey:
    valueFromSecret:
      key: secretAccessKey
      name: appvault-secret

providerType: Ontaps3

# oc create -f appvault-secret.yaml -n trident-protect
# oc create -f appvault.yaml -n trident-protect

```

```

[root@localhost openshift-benchmark]#
[root@localhost openshift-benchmark]# tp get appvault -n trident-protect
+-----+-----+-----+-----+
|     NAME      | PROVIDER | STATE   | AGE    | ERROR |
+-----+-----+-----+-----+
| ontap-s3-appvault | OntapS3 | Available | 2d2h |       |
+-----+-----+-----+-----+
[root@localhost openshift-benchmark]# ■

```

Beispiel-YAML-Datei zum Installieren der PostgreSQL-App

```

postgres.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres
spec:
  replicas: 1
  selector:

```

```

matchLabels:
  app: postgres
template:
  metadata:
    labels:
      app: postgres
spec:
  containers:
    - name: postgres
      image: postgres:14
      env:
        - name: POSTGRES_USER
          #value: "myuser"
          value: "admin"
        - name: POSTGRES_PASSWORD
          #value: "mypassword"
          value: "adminpass"
        - name: POSTGRES_DB
          value: "mydb"
        - name: PGDATA
          value: "/var/lib/postgresql/data/pgdata"
      ports:
        - containerPort: 5432
      volumeMounts:
        - name: postgres-storage
          mountPath: /var/lib/postgresql/data
  volumes:
    - name: postgres-storage
      persistentVolumeClaim:
        claimName: postgres-pvc
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgres-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
---
apiVersion: v1
kind: Service
metadata:
  name: postgres

```

```
spec:  
  selector:  
    app: postgres  
  ports:  
    - protocol: TCP  
      port: 5432  
      targetPort: 5432  
  type: ClusterIP
```

Now create the Trident protect application CR for the postgres app. Include the objects in the namespace postgres and create it in the postgres namespace.

```
# tp create app postgres-app --namespaces postgres -n postgres
```

```
[root@localhost RedHat]# tp get app -n postgres  
+-----+-----+-----+-----+  
| NAME | NAMESPACES | STATE | AGE |  
+-----+-----+-----+-----+  
| postgres-app | postgres | Ready | 24s |  
+-----+-----+-----+-----+  
[root@localhost RedHat]# ■
```

Snapshots erstellen

Erstellen eines On-Demand-Snapshots

```
# tp create snapshot postgres-snap1 --app postgres-app --appvault
ontap-s3-appvault -n postgres
Snapshot "postgres-snap1" created.
```

```
[root@localhost RedHat]# tp get snapshot -n postgres
+-----+-----+-----+-----+
|     NAME      | APP REF | STATE   | AGE    | ERROR |
+-----+-----+-----+-----+
| postgres-snap1 | postgres-app | Completed | 19s    |       |
+-----+-----+-----+-----+
[root@localhost RedHat]# ■
```

```
[root@localhost DataProtection]# oc get all,pvc,volumesnapshot -n postgres
apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.1000+
kubvirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.
NAME          READY  STATUS    RESTARTS  AGE
pod/postgres-cd9d6ccb-jfx49  1/1    Running   0          3h47m

NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/postgres  ClusterIP  172.30.132.112  <none>           5432/TCP  3h47m

NAME          READY  UP-TO-DATE  AVAILABLE  AGE
deployment.apps/postgres  1/1    1           1          3h47m

NAME          DESIRED  CURRENT  READY  AGE
replicaset.apps/postgres-cd9d6ccb  1        1        1        3h47m

NAME          VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/data-postgres-postgresql-0  Bound  pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  8G1    RWO      sc-zon
ea-nas  <unset>          4h40m
persistentvolumeclaim/postgres-pvc
ea-nas  <unset>          3h47m

NAME          STATUS      VOLUME          CAPACITY  ACCESS MODES  STORAG
ECLASS          AGE
persistentvolumeclaim/data-postgres-postgresql-0  Bound  pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  5G1    RWO      sc-zon
ea-nas  <unset>          3h47m

NAME          SOURCEPVC      SOURCESNAPSHOTCONTENT  RESTORESIZE  SNAPSHOTCLASS      SNAPSHOTCONTENT  READYTO
CREATIONTIME  AGE
volumesnapshot.snapshot.storage.k8s.io/snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  true
  postgres-pvc          53676Ki      trident-snapshotclass  snapcontent-796ea7f8-59a0-493e-bbd8-3a
e76fe9036c  13m          13m
volumesnapshot.snapshot.storage.k8s.io/snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  true
  data-postgres-postgresql-0          368Ki      trident-snapshotclass  snapcontent-86a464d4-ffd8-4279-9cf7-88
88a097c001  13m          13m
```

Erstellen eines Zeitplans Mit dem folgenden Befehl werden täglich um 15:33 Uhr Snapshots erstellt und zwei Snapshots und Backups aufbewahrt.

```
# tp create schedule schedule1 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres
Schedule "schedule1" created.
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
+-----+-----+-----+-----+-----+-----+
| NAME | APP | SCHEDULE | ENABLED | STATE | AGE | ERROR |
+-----+-----+-----+-----+-----+-----+
| schedule1 | postgres-app | Daily:hour=15,min=33 | true | 17s | | |
+-----+-----+-----+-----+-----+-----+
[root@localhost DataProtection]# ■
```

Erstellen eines Zeitplans mit YAML

```
# tp create schedule schedule2 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres --dry-run >
hourly-snapshotschedule.yaml
```

`cat hourly-snapshotschedule.yaml`

```
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: schedule2
  namespace: postgres
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: postgres-app
  backupRetention: "2"
  dataMover: Restic
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  #hour: "15"
  minute: "33"
  recurrenceRule: ""
  snapshotRetention: "2"
status: {}
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
+-----+-----+-----+-----+-----+-----+
| NAME | APP | SCHEDULE | ENABLED | STATE | AGE | ERROR |
+-----+-----+-----+-----+-----+-----+
| schedule1 | postgres-app | Daily:hour=15,min=33 | true | 8d7h | | |
| schedule2 | postgres-app | Hourly:min=33 | true | 8d7h | | |
+-----+-----+-----+-----+-----+-----+
[root@localhost DataProtection]# ■
```

Ad
Go

Sie können die nach diesem Zeitplan erstellten Snapshots sehen.

```
[root@localhost DataProtection]# tp get snap -n postgres
+-----+-----+-----+-----+
| NAME | APP REF | STATE | AGE | ERROR |
+-----+-----+-----+-----+
| hourly-3f1ee-20250214183300 | postgres-app | Completed | 19s | |
| postgres-snap1 | postgres-app | Completed | 1h25m | |
+-----+-----+-----+-----+
[root@localhost DataProtection]# ■
```

Außerdem werden Volume-Snapshots erstellt.

```
root@localhost DataProtection]# oc get volumesnapshots -n postgres
NAME                                CREATIONTIME   AGE      READYTOUSE   SOURCEPVC
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  114m    true      postgres-pvc
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  114m    true      data-postgres-postgresql-0
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  30m     true      postgres-pvc
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  30m     true      data-postgres-postgresql-0
-3a82-43f6-9868-dcadd2cc04e2          30m     30m
```

Löschen Sie die Anwendung, um den Verlust der Anwendung zu simulieren

```
# oc delete deployment/postgres -n postgres
# oc get pod,pvc -n postgres
No resources found in postgres namespace.
```

Wiederherstellen aus Snapshot im selben Namespace

```
# tp create sir postgres-sir --snapshot postgres/hourly-3f1ee-20250214183300 -n postgres
SnapshotInplaceRestore "postgres-sir" created.
```

[root@localhost DataProtection]# tp get sir -n postgres				
NAME	APPVAULT	STATE	AGE	ERROR
postgres-sir	ontap-s3-appvault	Completed	2m39s	

Die Anwendung und ihr PVC werden im selben Namespace wiederhergestellt.

[root@localhost DataProtection]# oc get pods,pvc -n postgres								
NAME	READY	STATUS	RESTARTS	AGE	CAPACITY	ACCESS MODES	STORAGECLASS	VOLUMEATTRIBUTESCLASS
pod/postgres-cd9d6ccb-x85tg	1/1	Running	0	98s				
NAME		STATUS						
persistentvolumeclaim/data-postgres-postgresql-0	Bound				8Gi	RWO	sc-zonea-nas	<unset>
persistentvolumeclaim/postgres-pvc	Bound				5Gi	RWO	sc-zonea-nas	<unset>
[root@localhost DataProtection]#								

Wiederherstellen aus Snapshot in einem anderen Namespace

```
# tp create snapshotrestore postgres-restore --snapshot postgres/hourly-3f1ee-20250214183300 --namespace-mapping postgres:postgres-restore -n postgres-restore
SnapshotRestore "postgres-restore" created.
```

[root@localhost DataProtection]# tp get snapshotrestore -n postgres-restore				
NAME	APPVAULT	STATE	AGE	ERROR
postgres-restore	ontap-s3-appvault	Completed	1m15s	

Sie können sehen, dass die Anwendung in einem neuen Namespace wiederhergestellt wurde.

[root@localhost DataProtection]#								
[root@localhost DataProtection]# oc get pods,pvc -n postgres-restore								
NAME	READY	STATUS	RESTARTS	AGE	CAPACITY	ACCESS MODES	STORAGECLASS	
pod/postgres-cd9d6ccb-pfxw9	1/1	Running	0	8d				
NAME		STATUS						
persistentvolumeclaim/data-postgres-postgresql-0	Bound				8Gi	RWO	sc-zonea-nas	
persistentvolumeclaim/postgres-pvc	Bound				5Gi	RWO	sc-zonea-nas	
[root@localhost DataProtection]#								

Backups erstellen

Erstellen eines On-Demand-Backups

```
# tp create backup postgres-backup1 --app postgres-app --appvault
ontap-s3-appvault -n postgres
Backup "postgres-backup1" created.
```

```
[root@localhost DataProtection]# tp get backup -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
backup1	postgres-app	Completed	5d12h	
daily-feac1-20250222153300	postgres-app	Completed	1d10h	
daily-feac1-20250223153300	postgres-app	Completed	10h36m	
hourly-3f1ee-20250224003300	postgres-app	Completed	1h36m	
hourly-3f1ee-20250224013300	postgres-app	Completed	36m27s	
postgres-backup1	postgres-app	Completed	6m19s	

Zeitplan für die Sicherung erstellen

Die täglichen und stündlichen Sicherungen in der obigen Liste werden anhand des zuvor eingerichteten Zeitplans erstellt.

```
# tp create schedule schedule1 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres
Schedule "schedule1" created.
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
```

NAME	APP	SCHEDULE	ENABLED	STATE	AGE	ERROR
schedule1	postgres-app	Daily:hour=15,min=33	true		9d8h	
schedule2	postgres-app	Hourly:min=33	true		9d8h	

Wiederherstellen aus einer Sicherung

Löschen Sie die Anwendung und PVCs, um einen Datenverlust zu simulieren.

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME           READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-dftkt  1/1    Running   0          11s
[root@localhost DataProtection]# oc get deployment -n postgres
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
postgres   1/1     1            1            20s
[root@localhost DataProtection]# oc delete deployment/postgres -n postgres
deployment.apps "postgres" deleted
[root@localhost DataProtection]# oc get pods -n postgres
No resources found in postgres namespace.
[root@localhost DataProtection]# oc get pvc -n postgres
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTE
SCLASS   AGE
data-postgres-postgresql-0   Bound   pvc-b2cd67fd-fe4d-49b1-9e06-a53bf7be575e   8Gi        RWO          sc-zonea-nas   <unset>
5d13h
postgres-pvc      Bound   pvc-2d549395-0cc6-4529-b2b9-7361bfb14fa8   5Gi        RWO          sc-zonea-nas   <unset>
5d13h
[root@localhost DataProtection]# oc delete pvc/data-postgres-postgresql-0 -n postgres
persistentvolumeclaim "data-postgres-postgresql-0" deleted
[root@localhost DataProtection]# oc delete pvc/postgres-pvc -n postgres
persistentvolumeclaim "postgres-pvc" deleted
[root@localhost DataProtection]# oc get pvc -n postgres
No resources found in postgres namespace.
[root@localhost DataProtection]#
```

Im selben Namespace wiederherstellen #tp create bir postgres-bir --backup postgres/hourly-3f1ee-20250224023300 -n postgres BackupInplaceRestore „postgres-bir“ erstellt.

```
[root@localhost DataProtection]# tp get bir -n postgres
+-----+-----+-----+-----+
| NAME      | APPVAULT | STATE    | AGE      | ERROR |
+-----+-----+-----+-----+
| postgres-bir | ontap-s3-appvault | Completed | 2m19s |      |
+-----+-----+-----+-----+
```

Die Anwendung und die PVCs werden im selben Namespace wiederhergestellt.

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME           READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-t857w  1/1    Running   0          10m
[root@localhost DataProtection]# oc get pvc -n postgres
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTE
ESCLASS   AGE
data-postgres-postgresql-0   Bound   pvc-0a849c19-16fe-466f-9733-85e82a8b1677   8Gi        RWO          sc-zonea-nas   <unset>
10m
postgres-pvc      Bound   pvc-ded304ea-02d4-4225-b606-63007666ad66   5Gi        RWO          sc-zonea-nas   <unset>
```

In einem anderen Namespace wiederherstellen Erstellen Sie einen neuen Namespace. Stellen Sie aus einer Sicherung im neuen Namespace wieder her.

```
[root@localhost DataProtection]# oc create ns postgres-restore-from-backup
namespace/postgres-restore-from-backup created
[root@localhost DataProtection]# tp create backuprestore postgres-restore-from-backup --backup postgres/postgres-backup1 --namespace-map
ping postgres:postgres-restore-from-backup -n postgres-restore-from-backup
BackupRestore "postgres-restore-from-backup" created.
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME          STATE    ERROR   AGE
postgres-restore-from-backup  Running   37s
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME          STATE    ERROR   AGE
postgres-restore-from-backup  Running   56s
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME          STATE    ERROR   AGE
postgres-restore-from-backup  Completed 2m52s
[root@localhost DataProtection]# oc get pods -n postgres-restore-from-backup
NAME           READY  STATUS    RESTARTS   AGE
postgres-cd9d6ccb-p659p  1/1    Running   0          2m9s
[root@localhost DataProtection]# oc get pvc -n postgres-restore-from-backup
NAME          STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUT
ESCLASS   AGE
data-postgres-postgresql-0  Bound    pvc-36df7399-95da-4c67-a621-af9434015bdb  8Gi       RWO          sc-zonea-nas  <unset>
2m18s
postgres-pvc            Bound    pvc-633de3aa-a4f9-4f3b-93cc-e91afbd4fe02  5Gi       RWO          sc-zonea-nas  <unset>
2m18s
[root@localhost DataProtection]#
```

Activate Windows
sc-zonea-nas <unset>
Go to Settings to activate Windows.

Migrieren von Anwendungen

Um eine Anwendung auf einen anderen Cluster zu klonen oder zu migrieren (durchführen eines Cluster-übergreifenden Klons), erstellen Sie eine Sicherung auf dem Quellcluster und stellen Sie die Sicherung dann auf einem anderen Cluster wieder her. Stellen Sie sicher, dass Trident Protect auf dem Zielcluster installiert ist.

Führen Sie im Quellcluster die im folgenden Bild gezeigten Schritte aus:

```
[root@localhost DataProtection]# tp create backup postgres-backup-cluster1 --app postgres-app --appvault ontap-s3-appvault -n postgres
Backup "postgres-backup-cluster1" created.
[root@localhost DataProtection]# tp get backup -n postgres
+-----+-----+-----+-----+
| NAME | APP REF | STATE | AGE | ERROR |
+-----+-----+-----+-----+
| backup1 | postgres-app | Completed | 5d14h | 
| daily-feac1-20250222153300 | postgres-app | Completed | 1d12h | 
| daily-feac1-20250223153300 | postgres-app | Completed | 12h18m | 
| hourly-3flee-20250224023300 | postgres-app | Completed | 1h18m | 
| hourly-3flee-20250224033300 | postgres-app | Completed | 18m26s | 
| postgres-backup-cluster1 | postgres-app | Running | 35s | 
| postgres-backup1 | postgres-app | Completed | 1h48m | 
+-----+-----+-----+-----+
[root@localhost DataProtection]# tp get backup -n postgres
+-----+-----+-----+-----+
| NAME | APP REF | STATE | AGE | ERROR |
+-----+-----+-----+-----+
| backup1 | postgres-app | Completed | 5d14h | 
| daily-feac1-20250222153300 | postgres-app | Completed | 1d12h | 
| daily-feac1-20250223153300 | postgres-app | Completed | 12h19m | 
| hourly-3flee-20250224023300 | postgres-app | Completed | 1h19m | 
| hourly-3flee-20250224033300 | postgres-app | Completed | 19m41s | 
| postgres-backup-cluster1 | postgres-app | Completed | 1m50s | 
| postgres-backup1 | postgres-app | Completed | 1h49m | 
+-----+-----+-----+-----+
```

Activate Windows
Go to Settings to activate Windows.

Wechseln Sie vom Quellcluster zum Zielcluster. Stellen Sie dann sicher, dass vom Zielclusterkontext aus auf den AppVault zugegriffen werden kann, und rufen Sie den AppVault-Inhalt vom Zielcluster ab.

```
[root@localhost DataProtection]# kubectl config use-context default/api-bm-cluster5-min-ocpv-sddc-netapp-com:6443/kube:admin
Switched to context "default/api-bm-cluster5-min-ocpv-sddc-netapp-com:6443/kube:admin".
[root@localhost DataProtection]# tp get appvault -n trident-protect
+-----+-----+-----+-----+
| NAME | PROVIDER | STATE | AGE | ERROR |
+-----+-----+-----+-----+
| ontap-s3-appvault | OntapS3 | Available | 3d6h | 
+-----+-----+-----+-----+
[root@localhost DataProtection]# tp get appvaultcontent ontap-s3-appvault --show-resources backup --show-paths
+-----+-----+-----+-----+
| CLUSTER | APP PATH | TYPE | NAME | TIMESTAMP | 
+-----+-----+-----+-----+
| ocp-cluster11 | bbox | backup | bboxbackup1 | 2025-01-17 15:57:49 (UTC) | bbox_68b4f05f-d5f1-4304-8c69-ad67514fe393/backups/bboxbackup1_3960c945-8ee0-42fe-945c-c57b0a3af6f |
| ocp-cluster11 | postgres-app | backup | backup1 | 2025-02-18 13:31:50 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/backup1_28e1bd9a-9b04-4412-8b96-811f9b62e2e3 |
| ocp-cluster11 | postgres-app | backup | daily-feac1-20250222153300 | 2025-02-22 15:34:44 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/daily-feac1-20250222153300_23d1386b-09f0-456f-aa49-a5865fd48abd |
| ocp-cluster11 | postgres-app | backup | daily-feac1-20250223153300_c492a4d1-38a9-4472-9684-4705c12a206d | 2025-02-23 15:34:42 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/daily-feac1-20250223153300_c492a4d1-38a9-4472-9684-4705c12a206d |
| ocp-cluster11 | postgres-app | backup | hourly-3flee-20250224033300 | 2025-02-24 03:34:44 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/hourly-3flee-20250224033300_3d09ab81-f6fe-47fa-a699-28006160cdbe |
| ocp-cluster11 | postgres-app | backup | hourly-3flee-20250224043300 | 2025-02-24 04:34:47 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/hourly-3flee-20250224043300_66805e4f-7631-48a6-98f7-d34bb8626031 |
| ocp-cluster11 | postgres-app | backup | postgres-backup-cluster1 | 2025-02-24 03:52:36 (UTC) | postgres-app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/postgres-backup-cluster1_ec0ed3f3-5500-4e72-afa8-117a04a0b1c3 |
```

Verwenden Sie den Sicherungspfad aus der Liste und erstellen Sie ein Backuprestore-CR-Objekt, wie im folgenden Befehl gezeigt.

```
# tp create backuprestore backup-restore-cluster2 --namespace-mapping
postgres:postgres --appvault ontap-s3-appvault --path postgres-
app_4d798ed5-cfa8-a5b6-c5e2d89aeb89/backups/postgres-backup-
cluster1_ec0ed3f3-5500-4e72-afa8-117a04a0b1c3 -n postgres
BackupRestore "backup-restore-cluster2" created.
```

```
[root@localhost DataProtection]# tp get backuprestore -n postgres
```

NAME	APPVAULT	STATE	AGE	ERROR
backup-restore-cluster2	ontap-s3-appvault	Completed	12m41s	

Sie können jetzt sehen, dass die Anwendungspods und die PVCs im Zielcluster erstellt werden.

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME           READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-2lvcq   1/1     Running   0          13m
[root@localhost DataProtection]# oc get pvc -n postgres
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUT
ESCLASS   AGE
data-postgres-postgresql-0   Bound    pvc-872a5182-601b-4848-b410-fef368337d07   8Gi        RWO          sc-zoneb-san   <unset>
13m
postgres-pvc            Bound    pvc-caf9fa71-76a8-4645-9bb5-2ed72e72948b   5Gi        RWO          sc-zoneb-san   <unset>
13m
[root@localhost DataProtection]#
```

Schützen Sie VMs in Red Hat OpenShift Virtualization mit Trident Protect

Schützen Sie VMs in OpenShift Virtualization mithilfe von Snapshots und Backups. Dieses Verfahren umfasst das Erstellen eines AppVault mithilfe des ONTAP S3-Objektspeichers, das Konfigurieren von Trident Protect zum Erfassen von VM-Daten, einschließlich Kubernetes-Ressourcenobjekten, persistenten Volumes und internen Images, sowie das Wiederherstellen der Daten bei Bedarf.

Virtuelle Maschinen in der OpenShift-Virtualisierungsumgebung sind containerisierte Anwendungen, die in den Worker-Knoten Ihrer OpenShift-Containerplattform ausgeführt werden. Es ist wichtig, die VM-Metadaten sowie die persistenten Datenträger der VMs zu schützen, damit Sie sie wiederherstellen können, wenn sie verloren gehen oder beschädigt werden.

Die persistenten Festplatten der OpenShift Virtualization VMs können durch ONTAP Speicher gesichert werden, der in den OpenShift-Cluster integriert ist, indem "[Trident CSI](#)". In diesem Abschnitt verwenden wir "[Trident -Schutz](#)" um Snapshots und Backups von VMs einschließlich ihrer Datenvolumes in ONTAP Object Storage zu erstellen.

Bei Bedarf stellen wir dann eine Wiederherstellung aus einem Snapshot oder einem Backup her.

Trident Protect ermöglicht Snapshots, Backups, Wiederherstellung und Notfallwiederherstellung von Anwendungen und VMs auf einem OpenShift-Cluster. Zu den Daten, die bei OpenShift-Virtualisierungs-VMs mit Trident Protect geschützt werden können, gehören mit den VMs verknüpfte Kubernetes-

Ressourcenobjekte, persistente Volumes und interne Images.

Im Folgenden sind die Versionen der verschiedenen Komponenten aufgeführt, die für die Beispiele in diesem Abschnitt verwendet wurden.

- "OpenShift Cluster 4.17"
- "OpenShift-Virtualisierung installiert über OpenShift Virtualization Operator, bereitgestellt von Red Hat"
- "Trident 25.02"
- "Trident 25.02"
- "ONTAP 9.16"

App Vault für Objektspeicher erstellen

AppVault erstellen

Bevor die Snapshots und Backups für eine Anwendung oder eine VM erstellt werden, muss in Trident Protect ein Objektspeicher konfiguriert werden, um die Snapshots und Backups zu speichern. Dies geschieht mithilfe des Bucket CR. Nur Administratoren können einen Bucket CR erstellen und konfigurieren. Der Bucket CR ist in Trident Protect als AppVault bekannt. AppVault-Objekte sind die deklarative Kubernetes-Workflow-Darstellung eines Speicher-Buckets. Ein AppVault CR enthält die Konfigurationen, die für die Verwendung eines Buckets in Schutzzvorgängen wie Backups, Snapshots, Wiederherstellungsvzvorgängen und SnapMirror Replikation erforderlich sind.

In diesem Beispiel zeigen wir die Verwendung von ONTAP S3 als Objektspeicher. Hier ist der Workflow zum Erstellen von AppVault CR für ONTAP S3: 1. Erstellen Sie einen S3-Objektspeicherserver in der SVM im ONTAP -Cluster. 2. Erstellen Sie einen Bucket im Object Store Server. 3. Erstellen Sie einen S3-Benutzer in der SVM. Bewahren Sie den Zugangsschlüssel und den geheimen Schlüssel an einem sicheren Ort auf. 4. Erstellen Sie in OpenShift ein Geheimnis zum Speichern der ONTAP S3-Anmeldeinformationen. 5. Erstellen eines AppVault-Objekts für ONTAP S3

Konfigurieren Sie Trident Protect AppVault für ONTAP S3

```
# alias tp='tridentctl-protect'

# cat appvault-secret.yaml
apiVersion: v1
stringData:
  accessKeyID: "<access key of S3>"
  secretAccessKey: "<secret access key of S3>"
# you can also provide base 64 encoded values instead of string values
#data:
# base 64 encoded values
#  accessKeyID: < base 64 encoded access key>
#  secretAccessKey: <base 64 encoded secretAccess key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque

# cat appvault.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: <lif for S3 access>
  secure: "false"
  skipCertValidation: "true"
providerCredentials:
  accessKeyID:
    valueFromSecret:
      key: accessKeyID
      name: appvault-secret
  secretAccessKey:
    valueFromSecret:
      key: secretAccessKey
      name: appvault-secret
providerType: Ontaps3

# oc create -f appvault-secret.yaml -n trident-protect
# oc create -f appvault.yaml -n trident-protect

```

[root@localhost VM-DataProtection]# tp get appvault						
NAME	PROVIDER	STATE	ERROR	MESSAGE	AGE	
ontap-s3-appvault	OntapS3	Available			8d17h	

[root@localhost VM-DataProtection]# ■

Erstellen einer VM in OpenShift Virtualization

Erstellen einer VM in OpenShift Virtualization

Die folgenden Screenshots zeigen die Erstellung der VM (Demo-Fedora im Namespace Demo) von der Konsole aus mithilfe der Vorlage. Die Root-Festplatte wählt die Standardspeicherklasse automatisch aus. Überprüfen Sie daher, ob die Standardspeicherklasse richtig eingestellt ist. In diesem Setup ist die Standardspeicherklasse **sc-zonea-san**. Stellen Sie sicher, dass Sie beim Erstellen der zusätzlichen Festplatte die Speicherklasse sc-zonea-san auswählen und das Kontrollkästchen „**Optimierte Speichereinstellungen anwenden**“ aktivieren. Dadurch werden die Zugriffsmodi auf RWX und der Volume-Modus auf Block gesetzt.



Trident unterstützt den RWX-Zugriffsmodus im Block-Volume-Modus für SAN (iSCSI, NVMe/TCP und FC). (Dies ist der Standardzugriffsmodus für NAS). Der RWX-Zugriffsmodus ist erforderlich, wenn Sie zu einem späteren Zeitpunkt eine Livemigration der VMs durchführen müssen.

StorageClasses

Name	Provisioner
SC sc-zonea-san – Default	csi.trident.netapp.io
SC thin-csi	csi.vsphere.vmware.com

Catalog

- VirtualMachines
- Templates
- InstanceTypes
- Preferences
- Bootable volumes
- MigrationPolicies
- Checkups

VirtualMachines

Project: demo

No VirtualMachines found

Click Create VirtualMachine to create your first VirtualMachine or view the catalog tab to...

Create VirtualMachine

- From InstanceType
- From template
- With YAML

Fedora VM

fedora-server-small

Template info

Operating system
Fedora VM

Workload type
Server (default)

Description
Template for Fedora Linux 39 VM or newer. A PVC with the Fedora disk image must be

Storage

Boot from CD ?

Disk source * ?

Template default

Disk size *

- 30 + GiB

Quick create VirtualMachine

VirtualMachine name * demo-fedora

Project demo **Public SSH key** Not configured edit

Start this VirtualMachine after creation

Activate Windows Go to Settings to activate Windows.

Quick create VirtualMachine **Customize VirtualMachine**

Catalog

Customize and create VirtualMachine YAML

Template: Fedora VM

Overview YAML Scheduling Environment Network interfaces **Disks** Scripts Metadata

Add disk ▼

Empty disk (blank)
Create a disk with no contents.

Use existing

Ephemeral disk (Container image)
Any changes are lost upon reboot.

Volume
Add a volume already available on the cluster.

Volume snapshot
Add a snapshot available on the cluster to the VirtualMachine.

Clone volume
Clone a volume available on the cluster and add it to the VirtualMachine.

Mount Windows drivers disk

Drive	Interface	Storage class
Disk	virtio	-
Disk	virtio	-

Create VirtualMachine **Cancel**

Activate Windows Go to Settings to activate Windows.

Add disk

disk1



Disk size *

- 30 + GiB ▾

Type

Disk



Hot plug is enabled only for "Disk" and "Lun" types

Interface *

VirtIO



Hot plug is enabled only for "SCSI" interface

StorageClass

SC SC-zonea-san



Apply optimized StorageProfile settings

Optimized values Access mode: ReadWriteMany, Volume mode: Block.



Save

Cancel

Project: demo

Overview YAML Scheduling Environment Network interfaces Disks Scripts Metadata

Add disk ▾

Filter Search by name... Mount Windows drivers disk

Name	Source	Size	Drive	Interface	Storage class
cloudinitdisk	Other	-	Disk	virtio	-
disk1	Other	30 GiB	Disk	virtio	sc-zonea-san
rootdisk	PVC (auto import)	30 GiB	Disk	virtio	-
	bootable				

Start this VirtualMachine after creation

Activate Windows
Go to Settings to activate V

Create VirtualMachine Cancel

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                                AGE      STATUS    READY
virtualmachine.kubevirt.io/demo-fedora 2m54s   Running   True

NAME          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-demo-fedora-69cts  1/1     Running   0          110s

NAME                           STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-fedora  Bound    pvc-406d9d57-4a5e-4123-93c1-13070b7b0140  30Gi     RWO         sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-311f2f81-1d25-4a9a-b0cb-836b4e702f04  30Gi     RWO         sc-zonea-san
[root@localhost VM-DataProtection]#
```

Erstellen einer App

App erstellen

Erstellen Sie eine Trident Protect-App für die VM

Im Beispiel verfügt der Demo-Namespace über eine VM und alle Ressourcen des Namespace werden beim Erstellen der App einbezogen.

```
# alias tp='tridentctl-protect'
# tp create app demo-vm --namespaces demo -n demo --dry-run > app.yaml

# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: demo-vm
  namespace: demo
spec:
  includedNamespaces:
  - namespace: demo
# oc create -f app.yaml -n demo
```

```
[root@localhost VM-DataProtection]# tp get app -n demo
+-----+-----+-----+
| NAME | NAMESPACES | STATE | AGE |
+-----+-----+-----+
| demo-vm | demo | Ready | 45s |
+-----+-----+-----+
[root@localhost VM-DataProtection]# ■
```

Schützen Sie die App, indem Sie ein Backup erstellen

Backups erstellen

Erstellen Sie ein On-Demand-Backup

Erstellen Sie ein Backup für die zuvor erstellte App (Demo-VM), das alle Ressourcen im Demo-Namespace enthält. Geben Sie den Appvault-Namen an, in dem die Sicherungen gespeichert werden.

```
# tp create backup demo-vm-backup-on-demand --app demo-vm --appvault  
ontap-s3-appvault -n demo  
Backup "demo-vm-backup-on-demand" created.
```

```
[root@localhost VM-DataProtection]# tp get backup -n demo  
+-----+-----+-----+-----+-----+  
| NAME | APP | RECLAIM POLICY | STATE | ERROR | AGE |  
+-----+-----+-----+-----+-----+  
| demo-vm-backup-on-demand | demo-vm | Retain | Completed | | 12m53s |  
+-----+-----+-----+-----+-----+  
[root@localhost VM-DataProtection]# ■
```

Erstellen Sie Backups nach einem Zeitplan

Erstellen Sie einen Zeitplan für die Sicherungen und geben Sie dabei die Granularität und die Anzahl der aufzubewahrenden Sicherungen an.

```

# tp create schedule backup-schedule1 --app demo-vm --appvault ontap-s3-appvault --granularity Hourly --minute 45 --backup-retention 1 -n demo --dry-run>backup-schedule-demo-vm.yaml
schedule.protect.trident.netapp.io/backup-schedule1 created

#cat backup-schedule-demo-vm.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: backup-schedule1
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  backupRetention: "1"
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  hour: ""
  minute: "45"
  recurrenceRule: ""
  snapshotRetention: "0"
status: {}
# oc create -f backup-schedule-demo-vm.yaml -n demo

```

```
[root@localhost VM-DataProtection]# tp get schedule -n demo
```

NAME	APP	SCHEDULE	ENABLED	STATE	ERROR	AGE
backup-schedule1	demo-vm	Hourly:min=45	true			9s

```
[root@localhost VM-DataProtection]# tp get backups -n demo
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
demo-vm-backup-on-demand	demo-vm	Retain	Completed		44m4s
hourly-4c094-20250312144500	demo-vm	Retain	Completed		20m34s

Wiederherstellen aus einer Sicherung

Wiederherstellen aus Backups

Stellen Sie die VM im selben Namespace wieder her

Im Beispiel enthält das Backup demo-vm-backup-on-demand das Backup mit der Demo-App für die Fedora-VM.

Löschen Sie zunächst die VM und stellen Sie sicher, dass die PVCs, Pods und VM-Objekte aus dem Namespace „demo“ gelöscht werden.

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                           AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora  59m   Running  True

NAME          READY  STATUS  RESTARTS  AGE
pod/virt-launcher-demo-fedora-69cts  1/1    Running  0          58m

NAME          STATUS  VOLUME
CLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-fedora      Bound   pvc-406d9d57-4a5e-4123-93c1-13070b7b0140  30Gi
a-san <unset>           59m
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-311f2f81-1d25-4a9a-b0cb-836b4e702f04  30Gi
a-san <unset>           59m
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc delete vm demo-fedora -n demo
virtualmachine.kubevirt.io "demo-fedora" deleted
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
No resources found in demo namespace.
[root@localhost VM-DataProtection]# ■
```

Activate
Go to Settings

Erstellen Sie jetzt ein Backup-in-Place-Wiederherstellungsobjekt.

```

# tp create bir demo-fedora-restore --backup demo/demo-vm-backup-on-demand -n demo --dry-run>vm-demo-bir.yaml

# cat vm-demo-bir.yaml
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  annotations:
    protect.trident.netapp.io/max-parallel-restore-jobs: "25"
  creationTimestamp: null
  name: demo-fedora-restore
  namespace: demo
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-0a7b3d353e13/backups/demo-vm-backup-on-demand_f6af3513-9739-480e-88c7-4cca45808a80
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-bir.yaml -n demo
backupinplacorestore.protect.trident.netapp.io/demo-fedora-restore
created

```

```

[root@localhost VM-DataProtection]# tp get bir -n demo
+-----+-----+-----+-----+
|     NAME      | APPVAULT      | STATE      | ERROR | AGE      |
+-----+-----+-----+-----+
| demo-fedora-restore | ontap-s3-appvault | Completed |       | 28m17s |
+-----+-----+-----+-----+
[root@localhost VM-DataProtection]#

```

Überprüfen Sie, ob die VM, Pods und PVCs wiederhergestellt sind

```

[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                           AGE   STATUS  READY
virtualmachine.kubervirt.io/demo-fedora  116s  Running  True

NAME          READY  STATUS    RESTARTS  AGE
pod/virt-launcher-demo-fedora-9kfxh  1/1    Running   0          116s

NAME                           STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-fedora  Bound    pvc-6f69a62c-285c-4980-b0dd-6c85baccf346  30Gi     RWX          sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8  30Gi     RWX          sc-zonea-san
[root@localhost VM-DataProtection]#

```

Stellen Sie die VM in einem anderen Namespace wieder her

Erstellen Sie zunächst einen neuen Namespace, in dem Sie die App wiederherstellen möchten, in diesem Beispiel demo2. Erstellen Sie dann ein Backup-Wiederherstellungsobjekt

```
# tp create br demo2-fedora-restore --backup demo/hourly-4c094-20250312154500 --namespace-mapping demo:demo2 -n demo2 --dry-run>vm-demo2-br.yaml

# cat vm-demo2-br.yaml
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  annotations:
    protect.trident.netapp.io/max-parallel-restore-jobs: "25"
  creationTimestamp: null
  name: demo2-fedora-restore
  namespace: demo2
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-0a7b3d353e13/backups/hourly-4c094-20250312154500_aaa14543-a3fa-41f1-a04c-44b1664d0f81
  appVaultRef: ontap-s3-appvault
  namespaceMapping:
    - destination: demo2
      source: demo
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo2-br.yaml -n demo2
```

```
[root@localhost VM-DataProtection]# tp get br -n demo2
+-----+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+-----+
| demo2-fedora-restore | ontap-s3-appvault | Completed | | 38m52s |
+-----+-----+-----+-----+-----+
```

Überprüfen Sie, ob die VM, Pods und PVCs im neuen Namespace „Demo2“ erstellt werden.

```
[root@localhost VM-DataProtection]#  
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2  
NAME AGE STATUS READY  
virtualmachine.kubevirt.io/demo-fedora 5m8s Running True  
  
NAME READY STATUS RESTARTS AGE  
pod/virt-launcher-demo-fedora-c7xc6 1/1 Running 0 5m7s  
  
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS  
persistentvolumeclaim/demo-fedora Bound pvc-4d278ae2-76cc-46f1-bbf8-071ae75e4a82 30Gi RWX sc-zonea-san  
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87 Bound pvc-9b96d09c-7226-4ffc-829b-2cee88e7a117 30Gi RWX sc-zonea-san  
[root@localhost VM-DataProtection]#
```

Schützen Sie die App mit Snapshots

Snapshots erstellen

Erstellen Sie einen On-Demand-Snapshot Erstellen Sie einen Snapshot für die App und geben Sie den App-Tresor an, in dem er gespeichert werden soll.

```
# tp create snapshot demo-vm-snapshot-on-demand --app demo-vm --appvault
ontap-s3-appvault -n demo --dry-run
# cat demo-vm-snapshot-on-demand.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  creationTimestamp: null
  name: demo-vm-snapshot-on-demand
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  completionTimeout: 0s
  volumeSnapshotsCreatedTimeout: 0s
  volumeSnapshotsReadyToUseTimeout: 0s
status:
  conditions: null
  postSnapshotExecHooksRunResults: null
  preSnapshotExecHooksRunResults: null
  state: ""

# oc create -f demo-vm-snapshot-on-demand.yaml
snapshot.protect.trident.netapp.io/demo-vm-snapshot-on-demand created
```

```
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
NAME                           AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-fedora   5m8s  Running  True

NAME          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-demo-fedora-c7xc6  1/1     Running   0          5m7s

NAME                           STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
persistentvolumeclaim/demo-fedora   Bound    pvc-4d278ae2-76cc-46f1-bbf8-071ae75e4a82   30Gi      RWX           sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87   Bound    pvc-9b96d09c-7226-4ffc-829b-2cee88e7a117   30Gi      RWX           sc-zonea-san
[root@localhost VM-DataProtection]#
```

Erstellen Sie einen Zeitplan für Snapshots Erstellen Sie einen Zeitplan für die Snapshots. Geben Sie die Granularität und die Anzahl der aufzubewahrenen Snapshots an.

```

# tp create Schedule snapshot-schedule1 --app demo-vm --appvault ontap-s3-appvault --granularity Hourly --minute 50 --snapshot-retention 1 -n demo --dry-run>snapshot-schedule-demo-vm.yaml

# cat snapshot-schedule-demo-vm.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: snapshot-schedule1
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  backupRetention: "0"
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  hour: ""
  minute: "50"
  recurrenceRule: ""
  snapshotRetention: "1"
status: {}

# oc create -f snapshot-schedule-demo-vm.yaml
schedule.protect.trident.netapp.io/snapshot-schedule1 created

```

```
[root@localhost VM-DataProtection]# tp get schedule -n demo
```

NAME	APP	SCHEDULE	ENABLED	STATE	ERROR	AGE
backup-schedule1	demo-vm	Hourly:min=45	true			5d23h
snapshot-schedule1	demo-vm	Hourly:min=50	true			12s

```
[root@localhost VM-DataProtection]#
```

```
[root@localhost VM-DataProtection]# tp get snapshots -n demo
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
backup-39b67e1c-f875-4045-93df-78634bae9dfb	demo-vm	Delete	Completed		6m29s
demo-vm-snapshot-on-demand	demo-vm	Delete	Completed		21m30s
hourly-51839-20250318135000	demo-vm	Delete	Completed		1m29s

```
[root@localhost VM-DataProtection]#
```

Wiederherstellen aus Snapshot

Wiederherstellen aus Snapshot

Stellen Sie die VM aus dem Snapshot im selben Namespace wieder her. Löschen Sie die VM demo-fedora aus dem Namespace demo2.

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                     AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora    28h   Running  True

NAME                                     STATUS  VOLUME
ORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-fedora          Bound   pvc-e0d5f79d-dff9-450d-be0e-90ab6880b7af
-zonea-san <unset>                     28h
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-e6f7890a-70c7-4538-9035-5e2e9379511d
-zonea-san <unset>                     28h
[root@localhost RedHat]# oc delete virtualmachine.kubevirt.io/demo-fedora -n demo
virtualmachine.kubevirt.io "demo-fedora" deleted
[root@localhost RedHat]# ■
```

Act
Go t

Erstellen Sie aus dem Snapshot der VM ein Snapshot-in-Place-Restore-Objekt.

```
# tp create sir demo-fedora-restore-from-snapshot --snapshot demo/demo-
vm-snapshot-on-demand -n demo --dry-run>vm-demo-sir.yaml

# cat vm-demo-sir.yaml
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  creationTimestamp: null
  name: demo-fedora-restore-from-snapshot
  namespace: demo
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-
0a7b3d353e13/snapshots/20250318132959_demo-vm-snapshot-
on-demand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-sir.yaml
snapshotinplacelorestore.protect.trident.netapp.io/demo-fedora-restore-
from-snapshot created
```

```
[root@localhost VM-DataProtection]# tp get sir -n demo
+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+
| demo-fedora-restore-from-snapshot | ontap-s3-appvault | Completed |      | 58m17s |
+-----+-----+-----+-----+
[root@localhost VM-DataProtection]#
```

Überprüfen Sie, ob die VM und ihre PVCs im Demo-Namespace erstellt werden.

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                AGE      STATUS    READY
virtualmachine.kubevirt.io/demo-fedora 5m17s   Running   True

NAME                                         STATUS    VOLUME
persistentvolumeclaim/demo-fedora          Bound    pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-db085154-079f-45ad-9e62-9656e913d01c
[root@localhost RedHat]#
[root@localhost RedHat]#
```

Stellen Sie die VM aus dem Snapshot in einem anderen Namespace wieder her

Löschen Sie die VM im Demo2-Namespace, die zuvor aus der Sicherung wiederhergestellt wurde.

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
NAME                                AGE      STATUS    READY
virtualmachine.kubevirt.io/demo-centos 3m12s   Running   True
virtualmachine.kubevirt.io/demo-fedora  3m11s   Running   True

NAME          READY  STATUS    RESTARTS  AGE
pod/virt-launcher-demo-centos-w77rr  1/1    Running   0          3m11s
pod/virt-launcher-demo-fedora-wwdtc  1/1    Running   0          3m11s

NAME           STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-centos    sc-zonea-san <unset>            3m26s
persistentvolumeclaim/demo-fedora    sc-zonea-san <unset>            3m25s
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 sc-zonea-san <unset>            3m20s
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87     sc-zonea-san <unset>            3m16s
[persistentvolumeclaim/demo-fedora-fuchsia-shrew-87]# oc delete virtualmachine.kubevirt.io/demo-fedora -n demo2
virtualmachine.kubevirt.io "demo-fedora" deleted
```

Erstellen Sie das Snapshot-Wiederherstellungsobjekt aus dem Snapshot und geben Sie die Namespace-Zuordnung an.

```

# tp create sr demo2-fedora-restore-from-snapshot --snapshot demo/demo-
vm-snapshot-on-demand --namespace-mapping demo:demo2 -n demo2 --dry
-run>vm-demo2-sr.yaml

# cat vm-demo2-sr.yaml
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  creationTimestamp: null
  name: demo2-fedora-restore-from-snapshot
  namespace: demo2
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-
0a7b3d353e13/snapshots/20250318132959_demo-vm-snapshot-
on-demand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  namespaceMapping:
    - destination: demo2
      source: demo
  resourceFilter: {}
status:
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo2-sr.yaml
snapshotrestore.protect.trident.netapp.io/demo2-fedora-restore-from-
snapshot created

```

[root@localhost VM-DataProtection]# tp get sr -n demo2				
NAME	APPVAULT	STATE	ERROR	AGE
demo2-fedora-restore-from-snapshot	ontap-s3-appvault	Completed		15m22s

Überprüfen Sie, ob die VM und ihre PVCs im neuen Namespace „Demo2“ wiederhergestellt sind.

[root@localhost RedHat]# oc get vm,pvc -n demo2			
NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-fedora	29h	Running	True
NAME	AGE	STATUS	VOLUME
pvcdcd9b2-4fca-486c-af9e-596bc5b4dc15		Bound	pvc-35dc
persistentvolumeclaim/demo-fedora			
-zonea-san <unset>	29h		
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87		Bound	pvc-575a3111-382f-4933-a778-0089fa1ea2af
-zonea-san <unset>	29h		

Wiederherstellen einer bestimmten VM

Auswählen bestimmter VMs in einem Namespace zum Erstellen von Snapshots/Backups und Wiederherstellen

Im vorherigen Beispiel hatten wir eine einzelne VM innerhalb eines Namespace. Durch die Einbeziehung des gesamten Namespace in die Sicherung wurden alle mit dieser VM verknüpften Ressourcen erfasst. Im folgenden Beispiel fügen wir demselben Namespace eine weitere VM hinzu und erstellen mithilfe eines Label-Selektors eine App nur für diese neue VM.

Erstellen Sie eine neue VM (Demo-Centos-VM) im Demo-Namespace

```
[root@localhost VM-DataProtection]# oc get vm,pod,pvc -n demo
NAME          AGE     STATUS    READY
virtualmachine.kubevirt.io/demo-centos   2m47s   Running   True
virtualmachine.kubevirt.io/demo-fedora   81m     Running   True

NAME          READY   STATUS    RESTARTS   AGE
pod/virt-launcher-demo-centos-2ng6g   1/1     Running   0          113s
pod/virt-launcher-demo-fedora-9kfxh   1/1     Running   0          81m

NAME                           STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-centos  Bound    pvc-ed0f492b-0109-471d-b395-9077ae5f1fa7  30Gi     RWX         sc-zonea-san
persistentvolumeclaim/demo-fedora  Bound    pvc-6f69a62c-285c-4980-b0dd-6c85baccf346  30Gi     RWX         sc-zonea-san
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34  Bound    pvc-3c01142a-4344-4293-ae67-7d3925c56211  30Gi     RWX         sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87    Bound    pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8  30Gi     RWX         sc-zonea-san
[root@localhost VM-DataProtection]#
```

Beschriften Sie die Demo-Centos-VM und die zugehörigen Ressourcen

```
[root@localhost VM-DataProtection]# oc label vm demo-centos category=protect-demo-centos -n demo
virtualmachine.kubevirt.io/demo-centos labeled
[root@localhost VM-DataProtection]# oc label pvc demo-centos category=protect-demo-centos -n demo
persistentvolumeclaim/demo-centos labeled
[root@localhost VM-DataProtection]# oc label pvc dv-demo-centos-lavender-tortoise-34 category=protect-demo-centos -n demo
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 labeled
[root@localhost VM-DataProtection]#
```

Überprüfen Sie, ob die Demo-CentOS-VM und die PVCs die Bezeichnungen erhalten haben.

```
[root@localhost VM-DataProtection]# oc get vm --show-labels -n demo
NAME          AGE     STATUS    READY   LABELS
demo-centos   6m1s   Running   True    app=proto-centos,category=protect-demo-centos,kubevirt.io/dynamic-credentials-support=true,vm.kubevirt.io/template.namespace=openshift,vm.kubevirt.io/template.revision=1,vm.kubevirt.io/template.version=v0.31.1,vm.kubevirt.io/template=centos-stream0-server-small
demo-fedora   85m    Running   True    app=proto-fedora,vm.kubevirt.io/template.namespace=openshift,vm.kubevirt.io/template.revision=1,vm.kubevirt.io/template.version=v0.31.1,vm.kubevirt.io/template=fedora-server-small
[root@localhost VM-DataProtection]#
```

```
[root@localhost VM-DataProtection]# oc get pvc --show-labels -n demo
NAME          STATUS    VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTECLASS  AGE      LABELS
demo-centos   Bound    pvc-ed0f492b-0109-471d-b395-9077ae5f1fa7  30Gi     RWX         sc-zonea-san  <unset>    7m41s  app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app=containerized-data-importer,category=protect-demo-centos,instancetype.kubevirt.io/default-instance-type.ul.medium,instance-type.kubevirt.io/default-preference=centos,preferencenCentos.stream0,kubevirt.io/created-by=a6a7b49c-669a-4e21-aa78-20743671b284
demo-fedora   Bound    pvc-6f69a62c-285c-4980-b0dd-6c85baccf346  30Gi     RWX         sc-zonea-san  <unset>    80m    app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app=containerized-data-importer,instance-type.kubevirt.io/default-instance-type=ul.medium,instance-type.kubevirt.io/default-preference=fedora,kubevirt.io/created-by=7d5184e9-22f8-4456-9afe-3d1984c430f9
dv-demo-centos-lavender-tortoise-34  Bound    pvc-3c01142a-4344-4293-ae67-7d3925c56211  30Gi     RWX         sc-zonea-san  <unset>    7m41s  app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app=containerized-data-importer,category=protect-demo-centos,kubevirt.io/created-by=a6a7b49c-669a-4e21-aa78-20743671b284
dv-demo-fedora-fuchsia-shrew-87    Bound    pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8  30Gi     RWX         sc-zonea-san  <unset>    80m    app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app=containerized-data-importer,kubevirt.io/created-by=7d5184e9-22f8-4456-9afe-3d1984c430f9
[root@localhost VM-DataProtection]#
```

Erstellen Sie mithilfe des Label-Selektors eine App nur für eine bestimmte VM (Demo-Centos)

```

# tp create app demo-centos-app --namespaces 'demo(category=protect-
demo-centos)' -n demo --dry-run>demo-centos-app.yaml

# cat demo-centos-app.yaml

apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: demo-centos-app
  namespace: demo
spec:
  includedNamespaces:
  - labelSelector:
    matchLabels:
      category: protect-demo-centos
      namespace: demo
status:
  conditions: null

# oc create -f demo-centos-app.yaml -n demo
application.protect.trident.netapp.io/demo-centos-app created

```

[root@localhost VM-DataProtection]# tp get app -n demo				
NAME	NAMESPACES	STATE	AGE	
demo-centos-app	demo	Ready	56s	
demo-vm	demo	Ready	4h6m	

Die Methode zum Erstellen von Backups und Snapshots auf Abruf und nach Zeitplan ist dieselbe wie zuvor gezeigt. Da die zum Erstellen der Snapshots oder Backups verwendete Trident-Protect-App nur die spezifische VM aus dem Namespace enthält, wird bei der Wiederherstellung von dort nur eine bestimmte VM wiederhergestellt. Nachfolgend wird ein Beispiel für einen Sicherungs-/Wiederherstellungsvorgang gezeigt.

Erstellen Sie ein Backup einer bestimmten VM in einem Namespace, indem Sie die entsprechende App verwenden

In den vorherigen Schritten wurde mithilfe von Label-Selektoren eine App erstellt, um nur die CentOS-VM in den Demo-Namespace aufzunehmen. Erstellen Sie ein Backup (in diesem Beispiel ein On-Demand-Backup) für diese App.

```
# tp create backup demo-centos-backup-on-demand --app demo-centos-app
--appvault ontap-s3-appvault -n demo
Backup "demo-centos-backup-on-demand" created.
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
demo-centos-backup-on-demand	demo-centos-app	Retain	Completed		13m22s
demo-vm-backup-on-demand	demo-vm	Retain	Completed		4h19m
hourly-4c094-20250312174500	demo-vm	Retain	Completed		56m17s

Eine bestimmte VM im selben Namespace wiederherstellen Das Backup einer bestimmten VM (CentOS) wurde mit der entsprechenden App erstellt. Wenn daraus ein Backup-in-Place-Restore oder ein Backup-Restore erstellt wird, wird nur diese spezielle VM wiederhergestellt. Löschen Sie die Centos-VM.

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                           AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-centos  4m27s  Running True
virtualmachine.kubevirt.io/demo-fedora  4m27s  Running True

NAME                                     STATUS  VOLUME
STORAGECLASS    VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-centos        Bound   pvc-e8faeaf8-fc0c-4d92-96de-c83a335a7a17
  sc-zonea-san <unset>                4m33s
persistentvolumeclaim/demo-fedora        Bound   pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
  sc-zonea-san <unset>                4m33s
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34  Bound   pvc-66eb7996-1420-4513-a67c-2824f08534da
  sc-zonea-san <unset>                4m33s
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87       Bound   pvc-db085154-079f-45ad-9e62-9656e913d01c
  sc-zonea-san <unset>                4m32s
```

```
[root@localhost RedHat]# oc delete virtualmachine.kubevirt.io/demo-centos -n demo
virtualmachine.kubevirt.io "demo-centos" deleted
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                           AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora  5m17s  Running True

NAME                                     STATUS  VOLUME
persistentvolumeclaim/demo-fedora        Bound   pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-db085154-079f-45ad-9e62-9656e913d01c
[root@localhost RedHat]#
```

Erstellen Sie eine direkte Sicherungswiederherstellung von „Demo-Centos-Backup-on-Demand“ und überprüfen Sie, ob die Centos-VM neu erstellt wurde.

```
#tp create bir demo-centos-restore --backup demo/demo-centos-backup-on-
demand -n demo
BackupInplaceRestore "demo-centos-restore" created.
```

```
[root@localhost RedHat]# tp get bir -n demo
+-----+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+
| demo-centos-restore | ontap-s3-appvault | Completed | | 57m9s |
| demo-fedora-restore | ontap-s3-appvault | Completed | | 7d5h |
+-----+-----+-----+-----+
```

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME AGE STATUS READY
virtualmachine.kubevirt.io/demo-centos 29m Running True
virtualmachine.kubevirt.io/demo-fedora 85m Running True

NAME STATUS VOLUME
STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
persistentvolumeclaim/demo-centos Bound pvc-82954bf7-4a7e-4e0c-9a04-4fa152e1b0ef
sc-zonea-san <unset> 29m
persistentvolumeclaim/demo-fedora Bound pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
sc-zonea-san <unset> 85m
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 Bound pvc-2a8d4eb5-ed6d-4408-b85d-e218e9a5d4b0
sc-zonea-san <unset> 29m
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87 Bound pvc-db085154-079f-45ad-9e62-9656e913d01c
sc-zonea-san <unset> 85m
[root@localhost RedHat]#
```

Stellen Sie eine bestimmte VM in einem anderen Namespace wieder her. Erstellen Sie eine Sicherungswiederherstellung in einem anderen Namespace (Demo3) von Demo-Centos-Backup-on-Demand und überprüfen Sie, ob die Centos-VM neu erstellt wurde.

```
# tp create br demo2-centos-restore --backup demo/demo-centos-backup-on-demand --namespace-mapping demo:demo3 -n demo3
BackupRestore "demo2-centos-restore" created.
```

```
[root@localhost RedHat]#
[root@localhost RedHat]# tp get br -n demo3
+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+
| demo2-centos-restore | ontap-s3-appvault | Completed | | 52m57s |
+-----+-----+-----+-----+
[root@localhost RedHat]#
```

```
[root@localhost RedHat]#
[root@localhost RedHat]# oc get vm,pvc -n demo3
NAME AGE STATUS READY
virtualmachine.kubevirt.io/demo-centos 19m Running True

NAME STATUS VOLUME
STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
persistentvolumeclaim/demo-centos Bound pvc-0a14e38f-07de-4e09-8f88-14a9a8bb45c2
sc-zonea-san <unset> 19m
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 Bound pvc-d4f9cf2f-264c-4d02-94bf-0db28b427acc
sc-zonea-san <unset> 19m
[root@localhost RedHat]#
```

Videodemonstration

Das folgende Video zeigt eine Demonstration zum Schutz einer VM mit Snapshots

[Schützen einer VM](#)

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFFE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGENDERWEINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.