



Protect data using Trident Protect

NetApp public and hybrid cloud solutions

NetApp

February 04, 2026

This PDF was generated from <https://docs.netapp.com/us-en/netapp-solutions-cloud/openshift/os-dp-tp-solution.html> on February 04, 2026. Always check docs.netapp.com for the latest.

Table of Contents

- Protect data using Trident Protect 1
 - Data protection for Container Apps in OpenShift Container Platform using Trident protect 1
 - Protect VMs in Red Hat OpenShift Virtualization using Trident protect 14
 - Create App Vault for Object Storage 15
 - Create a VM in OpenShift Virtualization. 17
 - Create an app 21
 - Protect the app by creating a backup 22
 - Restore from Backup 25
 - Protect the app using Snapshots 29
 - Restore from Snapshot 32
 - Restore a Specific VM 36
 - Video Demonstration 41

Protect data using Trident Protect

Data protection for Container Apps in OpenShift Container Platform using Trident protect

This section of the reference document provides details for creating Snapshots and backups of Container Apps using Trident protect. NetApp Trident protect provides advanced application data management capabilities that enhance the functionality and availability of stateful Kubernetes applications backed by NetApp ONTAP storage systems and the NetApp Trident CSI storage provisioner.

Trident protect creates application snapshots and backups, which means not only is the snapshot and backups of application data in persistent volumes are created, but snapshots and backups of application metadata are also created. The snapshots and backups created by Trident protect can be stored in any of the following Object Storage and restored from them at a later point.

- AWS S3
- Azure Blob storage
- Google Cloud Storage
- Ontap S3
- StorageGrid
- any other S3 compatible storage

Trident protect uses the Kubernetes model of role-based access control (RBAC).

By default, Trident protect provides a single system namespace called trident-protect and its associated default service account. If you have an organization with many users or specific security needs, you can use the RBAC features of Trident protect to gain more granular control over access to resources and namespaces.

Additional information on RBAC in Trident protect can be found in the [Trident protect documentation](#)



The cluster administrator has access to resources in the default trident-protect namespace and can also access resources in all other namespaces. Users cannot create application data management custom resource (CRs) like Snapshot and Backup CRs in the trident-protect namespace. As a best practice, users will need to create those CRs in the application namespace.

Trident protect can be installed using the instructions provided in the documentation [here](#). This section will show the workflow for the data protection of container applications and restoration of the applications using Trident protect.

1. Snapshot creation (on demand on scheduled)
2. Restore from Snapshot (restore to same and different namespace)
3. Backup creation
4. Restore from Backup

Pre-requisite

Prior to creating the Snapshots and backups for an application, an Object Storage must be configured in Trident protect to store the snapshots and backups. This is done using the bucket CR. Only administrators can create a bucket CR and configure it.

The bucket CR is known as AppVault in Trident protect. AppVault objects are the declarative Kubernetes workflow representation of a storage bucket. An AppVault CR contains the configurations necessary for a bucket to be used in protection operations, such as backups, snapshots, restore operations, and SnapMirror replication.

In this example, we will show the use of ONTAP S3 as Object storage.

Here is the workflow for creating AppVault CR for ONTAP S3:

1. Create S3 object store server in the SVM in ONTAP Cluster.
2. Create a bucket in the Object Store Server.
3. Create an S3 user in the SVM. Keep the access Key and the Secret Key in a safe location.
4. In OpenShift, create a secret to store the ONTAP S3 credentials.
5. Create an AppVault Object for ONTAP S3

Configure Trident protect AppVault for ONTAP S3

Sample yaml file for configuring Trident protect with ONTAP S3 as the 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]# █
```

Sample yaml file for installing 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          | NAMESPACE | STATE | AGE |
+-----+-----+-----+-----+
| postgres-app  | postgres  | Ready | 24s |
+-----+-----+-----+-----+
[root@localhost RedHat]#
```

```
# 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 v4.10000+

kubevirt.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	ECLASS	VOLUMEATTRIBUTESCLASS	AGE	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
persistentvolumeclaim/data-postgres-postgresql-0	ea-nas	<unset>	4h40m	Bound	pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00	8Gi	RWO	sc-zon
persistentvolumeclaim/postgres-pvc	ea-nas	<unset>	3h47m	Bound	pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3	5Gi	RWO	sc-zon

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

Using the following command, Snapshots will be created daily at 15:33 and two snapshots and backups will be retained.

```
[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]#
```

Creating a Schedule using 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]#
```

You can see snapshots created on this schedule.

```
[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]#
```

Volume snapshots are also created.

```
[root@localhost DataProtection]# oc get volumesnapshots -n postgres
```

NAME	CREATIONTIME	AGE	READYTOUSE	SOURCEPVC
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107	114m	114m	true	postgres-pvc
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107	114m	114m	true	data-postgres-postgresql-0
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1	30m	30m	true	postgres-pvc
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1	30m	30m	true	data-postgres-postgresql-0

Delete the Application to simulate loss of application

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

Restore from Snapshot to the same namespace

```
# tp create sir postgres-sir --snapshot postgres/hourly-3flee-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	

Application and its PVCs are restored to the same namespace.

Restore from Snapshot to a different namespace

```
# tp create snapshotrestore postgres-restore --snapshot postgres/hourly-3flee-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	

You can see that the application has been restored to a new namespace.

Create Backups

Creating an on-demand Backup

```
# 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	

Creating Schedule for Backup

The daily and the hourly backups in the list above are created from the schedule set up previously.

```
# 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	

Restore from backup

Delete the application and PVCs to simulate a data loss.

```
[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]#
```

Activate Windows
Go to Settings to activate Windows.

Restore to same namespace

#tp create bir postgres-bir --backup postgres/hourly-3f1ee-20250224023300 -n postgres
BackupInplaceRestore "postgres-bir" created.

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

The application and the PVCs are restored in the same namespace.

```
[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
SCLASS   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>
10m
```

Restore to a different namespace

Create a new namespace.

Restore from a backup to the new namespace.

```

[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   VOLUMEATTRIBUTION
postgres-pvc                        Bound       pvc-633de3aa-a4f9-4f3b-93cc-e91afbd4fe02  5Gi        RWO             sc-zonea-nas   <unset>
data-postgres-postgresql-0          Bound       pvc-36df7399-95da-4c67-a621-af9434015bdb    8Gi        RWO             sc-zonea-nas   <unset>

```

Migrate Applications

To clone or migrate an application to a different cluster (perform a cross-cluster clone), create a backup on the source cluster, and then restore the backup to a different cluster. Make sure that Trident protect is installed on the destination cluster.

On the source cluster, perform the steps as shown in the image below:

```
[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-3f1ee-20250224023300	postgres-app	Completed	1h18m	
hourly-3f1ee-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-3f1ee-20250224023300	postgres-app	Completed	1h19m	
hourly-3f1ee-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.

From the source cluster, switch context to the destination cluster.

Then, ensure that the AppVault is accessible from the destination cluster context and get the AppVault contents from the destination cluster.

```
[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)
ocp-cluster11	postgres-app	backup	backup1	2025-02-18 13:31:50 (UTC)
ocp-cluster11	postgres-app	backup	daily-feac1-20250222153300	2025-02-22 15:34:44 (UTC)
ocp-cluster11	postgres-app	backup	hourly-3f1ee-20250224033300	2025-02-24 03:34:44 (UTC)
ocp-cluster11	postgres-app	backup	postgres-backup-cluster1	2025-02-24 03:52:36 (UTC)

Use the backup path from the list and create a backupprestore CR object as shown in the command below.

```
# tp create backuprestore backup-restore-cluster2 --namespace-mapping
postgres:postgres --appvault ontap-s3-appvault --path postgres-
app_4d798ed5-cfa8-49ff-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	

You can now see that the application pods and the pvcs are created in the destination cluster.

```
[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
data-postgres-postgresql-0	Bound	pvc-872a5182-601b-4848-b410-fef368337d07	8Gi	RWO	sc-zoneb-san	<unset>
postgres-pvc	Bound	pvc-caf9fa71-76a8-4645-9bb5-2ed72e72948b	5Gi	RWO	sc-zoneb-san	<unset>

Protect VMs in Red Hat OpenShift Virtualization using Trident protect

Protect VMs in OpenShift Virtualization using snapshots and backup. This procedure includes creating an AppVault using ONTAP S3 object storage, configuring Trident Protect to capture VM data, including Kubernetes resource objects, persistent volumes, and internal images, and restoring the data when necessary.

Virtual machines in the OpenShift Virtualization environment are containerized applications that run in the worker nodes of your OpenShift Container platform. It is important to protect the VM metadata as well as the persistent disks of the VMs, so that when they are lost or corrupted, you can recover them.

The persistent disks of the OpenShift Virtualization VMs can be backed by ONTAP storage integrated to the OpenShift Cluster using [Trident CSI](#). In this section we use [Trident protect](#) to create snapshots and backups of VMs including its data volumes to ONTAP Object Storage.

We then restore from a snapshot or a backup when needed.

Trident protect enables snapshots, backups, restore, and disaster recovery of applications and VMs on an OpenShift cluster. For OpenShift Virtualization VMs, data that can be protected with Trident protect include Kubernetes resource objects associated with the VMs, persistent volumes, and internal images.

The following are the versions of the various components used for the examples in this section

- [OpenShift Cluster 4.17](#)
- [OpenShift Virtualization installed via OpenShift Virtualization Operator provided by Red Hat](#)
- [Trident 25.02](#)
- [Trident protect 25.02](#)
- [ONTAP 9.16](#)

Create App Vault for Object Storage

Create AppVault

Prior to creating the snapshots and backups for an application or a VM, an Object Storage must be configured in Trident protect to store the snapshots and backups. This is done using the bucket CR. Only administrators can create a bucket CR and configure it.

The bucket CR is known as AppVault in Trident protect. AppVault objects are the declarative Kubernetes workflow representation of a storage bucket. An AppVault CR contains the configurations necessary for a bucket to be used in protection operations, such as backups, snapshots, restore operations, and SnapMirror replication.

In this example, we will show the use of ONTAP S3 as Object storage.

Here is the workflow for creating AppVault CR for ONTAP S3:

1. Create S3 object store server in the SVM in ONTAP Cluster.
2. Create a bucket in the Object Store Server.
3. Create an S3 user in the SVM. Keep the access Key and the Secret Key in a safe location.
4. In OpenShift, create a secret to store the ONTAP S3 credentials.
5. Create an AppVault Object for ONTAP S3

Configure Trident protect AppVault for 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]#

```

Create a VM in OpenShift Virtualization

Create a VM in OpenShift Virtualization

The following screen shots show the creation of the VM (demo-fedora in namespace demo) from the console using the template. The root disk chooses the default storage class automatically, so, verify that the default storage class is set appropriately. In this setup, the default storage class is **sc-zonea-san**. Ensure that when you create the additional disk, you choose the storage class sc-zonea-san and check the **"Apply optimized storage settings"** checkbox. This will set the Access modes to RWX and Volume Mode to Block.



Trident supports RWX access mode in Block Volume mode for SAN (iSCSI, NVMe/TCP and FC). (It is the default access mode for NAS). RWX Access mode is required if you need to do Live migration of the VMs at a later point.

StorageClasses

Name ▾ Search by name... /

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

Catalog

VirtualMachines

Templates

InstanceTypes

Preferences

Bootable volumes

MigrationPolicies

Checkups

Migration >

Networking >

Storage ▾

PersistentVolumes

Project: demo ▾

VirtualMachines



No VirtualMachines found

Click Create VirtualMachine to create your first VirtualMachine or view the [catalog](#) tab to c

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 

☒ Start this VirtualMachine after creation

Quick create VirtualMachine

Customize VirtualMachine

Activate Windows

Go to Settings to activate Windows.

Cancel

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.

/

☐ Mount Windows drivers disk

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.

Create VirtualMachine

Cancel

Drive

Interface

Storage class

Disk	virtio	-	⋮
Disk	virtio	-	⋮

Activate Windows

Go to Settings to activate Windows

19

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

OverviewYAMLSchedulingEnvironmentNetwork interfacesDisksScriptsMetadata

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-zones-san
rootdisk	PVC (auto import)	30 GiB	Disk	virtio	-

bootable

☒ Start this VirtualMachine after creation

Create VirtualMachineCancel

Activate Windows
Go to Settings to activate V

Create an app

Create App

Create a trident protect app for the VM

In the example, the demo namespace has one VM and all resources of the namespace is included when creating the app.

```
# 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]#
```

Protect the app by creating a backup

Create Backups

Create an On-demand Backup

Create a backup for the app (demo-vm) created previously that includes all resources in the demo namespace. Provide the appvault name where the backups will be stored.

```
# 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]#
```

Create Backups on a Schedule

Create a schedule for the backups specifying the granularity and the number of backups to retain.

```
# 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

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

Restore from Backup

Restore from Backups

Restore the VM to the same namespace

In the example the backup demo-vm-backup-on-demand contains the backup with the demo-app for the fedora VM.

First, delete the VM and ensure that the PVCs, pod and the VM objects are deleted from the namespace "demo"

```
[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                                     CAPACITY
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

Now, create a backup-in-place restore object.

```
# 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
backupinplacerestore.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]#
```

Verify that the VM, pods and PVCs are restored

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.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]#
```

Restore the VM to a different namespace

First create a new namespace to which you want to restore the app to, in this example demo2. Then create a backup restore object

```
# 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

Verify that the VM, pods and pvcs are created in the new namespace demo2.

```
[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]#
```

Protect the app using Snapshots

Create Snapshots

Create an on-demand snapshot

Create a snapshot for the app and specify the appvault where it needs to be stored.

```
# tp create snapshot demo-vm-snapshot-ondemand --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-ondemand
  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-ondemand 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]#
```

Create a schedule for snapshots

Create schedule for the snapshots. Specify the granularity and the number of snapshots to be retained.

```
# 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-ondemand	demo-vm	Delete	Completed		21m30s
hourly-51839-20250318135000	demo-vm	Delete	Completed		1m29s

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

Restore from Snapshot

Restore from Snapshot

Restore the VM from the snapshot to the same namespace

Delete the VM demo-fedora from the demo2 namespace.

```
[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
Got

Create a snapshot-in-place-restore object from the snapshot of the VM.

```
# tp create sir demo-fedora-restore-from-snapshot --snapshot demo/demo-vm-snapshot-ondemand -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-ondemand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-sir.yaml
snapshotinplacerestore.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]#
```

Verify that the VM and its PVCs are created in the demo namespace.

```
[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]#
```

Restore the VM from the snapshot to a different namespace

Delete the VM in the demo2 namespace previously restored from the backup.

Create the snapshot restore object from the snapshot and provide the namespace mapping.

```
# tp create sr demo2-fedora-restore-from-snapshot --snapshot demo/demo-vm-snapshot-ondemand --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-ondemand_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

Verify that the VM and its PVCs are restored in the new namespace demo2.

```
[root@localhost RedHat]# oc get vm,pvc -n demo2
```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-fedora	29h	Running	True

NAME	ORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE	STATUS	VOLUME
persistentvolumeclaim/demo-fedora				Bound	pvc-35dcd9b2-4fca-486c-af9e-596bc5b4dc15
-zonea-san	<unset>		29h		
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87				Bound	pvc-575a3111-382f-4933-a778-0089fa1ea2af
-zonea-san	<unset>		29h		

```
[root@localhost RedHat]#
```

Restore a Specific VM

Selecting specific VMs in a namespace to create snapshots/backups and restore

In the previous example, we had a single VM within a namespace. By including the entire namespace in the backup, all resources associated with that VM were captured. In the following example, we add another VM to the same namespace and create an app just for this new VM by using a label selector.

Create a new VM (demo-centos vm) in the 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-2nq6g 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]#
```

Label the demo-centos vm and its associated resources

Verify that the demo-centos vm and pvcs got the labels

```
[root@localhost VM-DataProtection]# oc get vm --show-labels -n demo
NAME                                AGE      STATUS    READY    LABELS
demo-centos 6m1s    Running   True     app=demo-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-stream9-server-small
demo-fedora 85m      Running   True     app=demo-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    VOLUMEATTRIBUTESCLASS    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-instancetype=ul.medium,instancetype.kubevirt.io/default-preference=centos-stream9,kubevirt.io/created-by=a6a7b49c-669a-4e21-a478-20743671b284
demo-fedora    Bound     pvc-6f69a62c-285c-4980-b0dd-6c85baccf346 30Gi         RWX            sc-zonea-san    <unset>                85m      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,instancetype.kubevirt.io/default-instancetype=ul.medium,instancetype.kubevirt.io/default-preference=fedora,kubevirt.io/created-by=7d5184e9-22f0-4456-9afe-3d1904c430f9
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-a478-20743671b284
dv-demo-fedora-fuchsia-shrew-87    Bound     pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8 30Gi         RWX            sc-zonea-san    <unset>                85m      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=7d5184e9-22f0-4456-9afe-3d1904c430f9
[root@localhost VM-DataProtection]#
```

Create an app for only a specific VM (demo-centos) using the label selector

```
# 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

The method of creating backups and snapshots on-demand and on a schedule is the same as shown previously.

Since the trident-protect app that is being used to create the snapshots or backups only contains the specific VM from the namespace, restoring from them only restores a specific VM.

A sample backup/restore operation is shown as an example below.

Create a backup of a specific VM in a namespace by using its corresponding app

In the previous steps, an app was created using label selectors to include only the centos vm in the demo namespace. Create a backup (on-demand backup, in this example) for this 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

Restore a specific VM to the same namespace

The backup of a specific VM (centos) was created using the corresponding app.

If a backup-in-place-restore or a backup-restore is created from this, only this specific VM is restored.

Delete the 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]#
```

Create a backup in-place restore from demo-centos-backup-on-demand and verify that the centos VM has been recreated.

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

```
[root@localhost RedHat]#
```

Restore a specific VM to a different namespace

Create a backup restore to a different namespace (demo3) from demo-centos-backup-on-demand and verify that the centos VM has been recreated.

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

```
[root@localhost RedHat]#
```

Video Demonstration

The following video shows a demonstration of protecting a VM using Snapshots

[Protecting a VM](#)

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.