

# Red Hat OpenShift Service on AWS with FSxN

**NetApp Solutions** 

NetApp January 03, 2025

This PDF was generated from https://docs.netapp.com/us-en/netapp-solutions/containers/rh-osn\_use\_case\_rosa\_solution\_overview.html on January 03, 2025. Always check docs.netapp.com for the latest.

# **Table of Contents**

| Red Hat OpenShift Service on AWS with FSxN          | 1 |
|---|---|
| Red Hat OpenShift Service on AWS with NetApp ONTAP  | 1 |
| Red Hat OpenShift Service on AWS with NetApp ONTAP. | C |

# Red Hat OpenShift Service on AWS with FSxN

# Red Hat OpenShift Service on AWS with NetApp ONTAP

# Overview

In this section, we will show how to utilize FSx for ONTAP as a persistent storage layer for applications running on ROSA. It will show the installation of the NetApp Trident CSI driver on a ROSA cluster, the provisioning of an FSx for ONTAP file system, and the deployment of a sample stateful application. It will also show strategies for backing up and restoring your application data. With this integrated solution, you can establish a shared storage framework that effortlessly scales across AZs, simplifying the processes of scaling, protecting, and restoring your data using the Trident CSI driver.

## Prerequisites

- AWS account
- A Red Hat account
- · IAM user with appropriate permissions to create and access ROSA cluster
- AWS CLI
- ROSA CLI
- OpenShift command-line interface (oc)
- Helm 3 documentation
- A HCP ROSA cluster
- Access to Red Hat OpenShift web console

This diagram shows the ROSA cluster deployed in multiple AZs. ROSA cluster's master nodes, infrastructure nodes are in Red Hat's VPC, while the worker nodes are in a VPC in the customer's account. We'll create an FSx for ONTAP file system within the same VPC and install the Trident driver in the ROSA cluster, allowing all the subnets of this VPC to connect to the file system.

| s AV | VS Cloud<br>Availability Zone 1 | Availability Zone 2              | Availability Zone 2 | AWS Clou<br>OpenShift control plan | d<br>e (API server, etcd, cont | rolier, scheduler) manage |
|------|---------------------------------|----------------------------------|---------------------|------------------------------------|--------------------------------|---------------------------|
|      | Private subnet                  | Private subnet                   | Private subnet      | M5 instance                        | M5 instance                    | M5 instance               |
|      | <b>F</b><br>Instances           | OpenShift Worker nodes. (router) | instances           |                                    |                                |                           |
|      |                                 | FS                               |                     |                                    |                                |                           |

## **Initial Setup**

#### 1. Provision FSx for NetApp ONTAP

Create a multi-AZ FSx for NetApp ONTAP in the same VPC as the ROSA cluster. There are several ways to do this. The details of creating FSxN using a CloudFormation Stack are provided

#### a.Clone the GitHub repository

```
$ git clone https://github.com/aws-samples/rosa-fsx-netapp-ontap.git
```

#### b.Run the CloudFormation Stack

Run the command below by replacing the parameter values with your own values:

\$ cd rosa-fsx-netapp-ontap/fsx

| $ aws$ cloudformation create-stack $\$  |
|---|
| stack-name ROSA-FSXONTAP \  |
| template-body file://./FSxONTAP.yaml \  |
| region <region-name> \</region-name>  |
| parameters \  |
| ParameterKey=Subnet1ID,ParameterValue=[subnet1_ID] \  |
| ParameterKey=Subnet2ID,ParameterValue=[subnet2_ID] \  |
| ParameterKey=myVpc,ParameterValue=[VPC_ID] \  |
| ParameterKey=FSxONTAPRouteTable,ParameterValue=[routetable1_ID,routetable2  |
| _ID] \  |
| ParameterKey=FileSystemName,ParameterValue=ROSA-myFSxONTAP \  |
| ParameterKey=ThroughputCapacity,ParameterValue=1024 $\setminus$   |
| ParameterKey=FSxAllowedCIDR,ParameterValue=[your allowed CIDR] \  |
|   |
| ParameterKey=FsxAdminPassword,ParameterValue=[Define Admin password] \  |
| ParameterKey=FsxAdminPassword,ParameterValue=[Define Admin password] \<br>ParameterKey=SvmAdminPassword,ParameterValue=[Define SVM password] \                                      |
| ParameterKey=FsxAdminPassword,ParameterValue=[Define Admin password] \<br>ParameterKey=SvmAdminPassword,ParameterValue=[Define SVM password] \<br>capabilities CAPABILITY_NAMED_IAM |

Where :

region-name: same as the region where the ROSA cluster is deployed subnet1\_ID : id of the Preferred subnet for FSxN subnet2\_ID: id of the Standby subnet for FSxN VPC\_ID: id of the VPC where the ROSA cluster is deployed routetable1\_ID, routetable2\_ID: ids of the route tables associated with the subnets chosen above your\_allowed\_CIDR: allowed CIDR range for the FSx for ONTAP security groups ingress rules to control access. You can use 0.0.0.0/0 or any appropriate CIDR to allow all traffic to access the specific ports of FSx for ONTAP. Define Admin password: A password to login to FSxN Define SVM password: A password to login to SVM that will be created.

Verify that your file system and storage virtual machine (SVM) has been created using the Amazon FSx console, shown below:

| File systems                      | OntapFileSystem_               |                                  |        | Attach Actions <b>v</b>                        |
|-----------------------------------|--------------------------------|----------------------------------|--------|--|
| File Caches<br>Backups            | ▼ Summary                      |                                  |        |  |
| ONTAP<br>Storage virtual machines | File system ID                 | SSD storage capacity<br>1024 GiB | Update | Availability Zones<br>us-east-2a (Preferred) 🗇 |
| OpenZF5                           | Chicycle state     Ø Available | Throughput capacity<br>1024 MB/s | Update | Creation time                                  |
| Snapshots                         | File system type<br>ONTAP      | Provisioned IOPS<br>3072         | Update | 000011125350400                                |
| FSx on Service Quotes 🗹           | Deployment type                | Number of HA pairs               |        |  |
| Settings                          | Multi-AZ 1                     | 1                                |        |  |

2.Install and configure Trident CSI driver for the ROSA cluster

\$ helm repo add netapp-trident https://netapp.github.io/trident-helm-chart

#### b.Install trident using helm

```
$ helm install trident netapp-trident/trident-operator --version
100.2406.0 --create-namespace --namespace trident
```



Depending on the version you install, the version parameter will need to be changed in the command shown. Refer to the documentation for the correct version number. For additional methods of installing Trident, refer to the Trident documentation.

#### c.Verify that all Trident pods are in the running state

| [root@localhost hcp-testing]#  |         |            |          |     |
|--|---------|------------|----------|-----|
| [root@localhost hcp-testing]#  |         |            |          |     |
| [root@localhost hcp-testing]# oc ge  | et pods | -n trident |          |     |
| NAME   | READY   | STATUS     | RESTARTS | AGE |
| trident-controller-f5f6796f-vd2sk  | 6/6     | Running    | 0        | 19h |
| trident-node-linux-4svgz   | 2/2     | Running    | 0        | 19h |
| trident-node-linux-dj9j4   | 2/2     | Running    | 0        | 19h |
| trident-node-linux-jlshh   | 2/2     | Running    | 0        | 19h |
| trident-node-linux-sqthw   | 2/2     | Running    | 0        | 19h |
| trident-node-linux-ttj9c   | 2/2     | Running    | 0        | 19h |
| trident-node-linux-vmjr5   | 2/2     | Running    | 0        | 19h |
| trident-node-linux-wvqsf   | 2/2     | Running    | 0        | 19h |
| <pre>trident-operator-545869857c-kgc7p [root@localhost hcp-testing]# _</pre> | 1/1     | Running    | 0        | 19h |

#### 3. Configure the Trident CSI backend to use FSx for ONTAP (ONTAP NAS)

The Trident back-end configuration tells Trident how to communicate with the storage system (in this case, FSx for ONTAP). For creating the backend, we will provide the credentials of the Storage Virtual machine to connect to, along with the Cluster Management and the NFS data interfaces. We will use the ontap-nas driver to provision storage volumes in FSx file system.

#### a. First, create a secret for the SVM credentials using the following yaml

```
apiVersion: v1
kind: Secret
metadata:
    name: backend-fsx-ontap-nas-secret
    namespace: trident
type: Opaque
stringData:
    username: vsadmin
    password: <value provided for Define SVM password as a parameter to the
Cloud Formation Stack>
```

 $(\mathbf{i})$ 

You can also retrieve the SVM password created for FSxN from the AWS Secrets Manager as shown below.

| VS Secrets Manager > Secrets                                  |  |                      |
|---|--|----------------------|
| Secrets   |  | C Store a new secret |
| Q. Filter secrets by nome, description, tog key, tog value, o | uning service or primary Region  | C 1 3 0              |
| Secret name   | Description  | Last retrieved (UTC) |
| HCP-ROSA-FSXONTAP-SVMAdminPassword                            | SVMAdminPassword   | October 9, 2024      |
|   | and the second |                      |

| Secret details   |                    | C Actions ¥ |
|--|--------------------|-------------|
| Encryption key<br>aws/secretsmanager<br>Secret name<br>HCP-ROSA-FSXONTAP-SVMAdminPassword<br>Secret ARN<br>armaws:secretsmanager:us-east-2:316088182667:secret:HCP-ROSA-FSXONTAP-SVMAdminPassword-<br>ISIUaf | Secret description |             |
| Dverview Rotation Versions Replication Tags  |                    |             |

b.Next, add the secret for the SVM credentials to the ROSA cluster using the following command

```
$ oc apply -f svm_secret.yaml
```

You can verify that the secret has been added in the trident namespace using the following command

\$ oc get secrets -n trident |grep backend-fsx-ontap-nas-secret

[root@localhost hcp-testing]#
[root@localhost hcp-testing]# oc get secrets -n trident | grep backend-fsx-ontap-nas-secret
backend-fsx-ontap-nas-secret Opaque 2 21h
[root@localhost hcp-testing]# \_

#### c. Next, create the backend object

For this, move into the **fsx** directory of your cloned Git repository. Open the file backend-ontap-nas.yaml. Replace the following:

**managementLIF** with the Management DNS name **dataLIF** with the NFS DNS name of the Amazon FSx SVM and **svm** with the SVM name. Create the backend object using the following command.

Create the backend object using the following command.

\$ oc apply -f backend-ontap-nas.yaml



You can get the Management DNS name, NFS DNS name and the SVM name from the Amazon FSx Console as shown in the screenshot below

| Amazon <mark>F</mark> Sx  | × | Summary  |   |                  |
|---|---|--|---|------------------|
| File systems<br>Volumes<br>File Caches<br>Backups<br>V ONTAP<br>Storage virtual machines<br>V OpenZFS<br>Snapshots<br>F5x on Service Quotas [2]<br>Settings |   | SVM ID       Creation time         svm-07a733da2584f2045 ()       Creation time         SVM name       Lifecycle state         SVM1 ()       Ifecycle state         SVM1 ()       Ifecycle state         UUID       Subtype         a845e7bf-8653-11ef-8f27-0f43b1500927       DEFAULT         File system ID       File system ID         fs-03a16050beae7ca24 ()       Resource ARN         amaws:fscus-east-2316088182667:storage-virtual-machine/fs-03a16050beae7ca24/svm-       Ifecource ARN | 00  | Active Directory |
|   |   | Endpoints Administration Volumes Tags Endpoints  |   |                  |
|   |   | Management DNS name<br>svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonavis.com 🗗<br>NFS DNS name<br>svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonavis.com 🗇  | Management IP address<br>198.19.255.182 <b>(7)</b><br>NFS IP address<br>198.19.255.182 <b>(7)</b> |                  |
|   |   | iSCSI DNS name<br>iscsi.svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonaws.com<br>Ø  | iSCSI IP addresses<br>10.10.9.32, 10.10.26.28 🗗   |                  |

d. Now, run the following command to verify that the backend object has been created and Phase is

[root@localhost hcp-testing]# [root@localhost hcp-testing]# [root@localhost hcp-testing]# oc apply -f backend-ontap-nas.yaml tridentbackendconfig.trident.netapp.io/backend-fsx-ontap-nas created [root@localhost hcp-testing]# oc get tbc -n trident BACKEND NAME NAME BACKEND UUID PHASE STATUS acc65405-56be-4719-999d-27b448a50e29 backend-fsx-ontap-nas fsx-ontap Bound Success [root@localhost hcp-testing]# \_

#### 4. Create Storage Class

Now that the Trident backend is configured, you can create a Kubernetes storage class to use the backend. Storage class is a resource object made available to the cluster. It describes and classifies the type of storage that you can request for an application.

a. Review the file storage-class-csi-nas.yaml in the fsx folder.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: trident-csi
provisioner: csi.trident.netapp.io
parameters:
   backendType: "ontap-nas"
   fsType: "ext4"
allowVolumeExpansion: True
reclaimPolicy: Retain
```

b. Create Storage Class in ROSA cluster and verify that trident-csi storage class has been created.

| [root@localhost hcp<br>[root@localhost hcp<br>[root@localhost hcp | p-testing]#<br>p-testing]#<br>p-testing]# oc apply -f : | storage-class-cs | i-nas.vaml           |                        |       |
|---|---|------------------|----------------------|------------------------|-------|
| storageclass.stora  | ge.k8s.io/trident-csi cro                               | eated            |                      |                        |       |
| LLOOT@TOC9TUO21 UC  | PROVISIONER   | RECLATMPOLTCY    | VOLUMERTNDTNGMODE    | ALL OWNOLLIMEEXPANSTON | AGE   |
| p2-csi  | ebs.csi.aws.com   | Delete           | WaitForFirstConsumer | true                   | 2d16h |
| p3-csi (default)  | ebs.csi.aws.com   | Delete           | WaitForFirstConsumer | true                   | 2d16h |
| trident-csi<br>[root@localhost hcj                                | csi.trident.netapp.io<br>p-testing]# _                  | Retain           | Immediate            | true                   | 4s    |

This completes the installation of Trident CSI driver and its connectivity to FSx for ONTAP file system. Now you can deploy a sample Postgresql stateful application on ROSA using file volumes on FSx for ONTAP.

c. Verify that there are no PVCs and PVs created using the trident-csi storage class.

| rootglocalnost ncp-tetting #<br>[rootglocalhost hcp-testing]# oc get<br>NaWESPACE<br>openshift-monitoring<br>openshift-wirtualization-os-inages<br>openshift-virtualization-os-inages<br>openshift-virtualization-os-inages<br>openshift-virtualization-os-inages<br>openshift-virtualization-os-inages | pvc -A<br>NAME<br>prometheus-data<br>prometheus-data<br>centos-stream9-i<br>centos-stream9-i<br>fedora-21a0f3e6<br>rhe18-0052df0eb<br>rhe18-052df0eb | prometheus-kdts-<br>prometheus-kdts-<br>baellicddSal<br>802441241044<br>8026<br>259<br>664 | STATUS V<br>e Bound p<br>Bound p<br>Bound p<br>Bound p<br>Bound p<br>Bound p<br>Bound p | 0LUME<br>vc-9a4553a5<br>vc-7d949aef<br>vc-8bb01464<br>vc-8bb084a<br>vc-64f375ad<br>vc-2dc6de48<br>vc-74374ce7 | +07e9-440a-8200-994384c97624<br>+0904-409a-856-514e4055fbab2<br>-cb3f-4409-8074-394028496c16<br>+5ef-4522-1690-10004f4102c1<br>4077-4566-3308-56864138e79c<br>5936-411e-0c33-09206f308046<br>55684-447c-0535-9220cf45444 | CAPACITY<br>100Gi<br>100Gi<br>30Gi<br>30Gi<br>30Gi<br>30Gi<br>30Gi | ACCESS HODES<br>RHO<br>RHO<br>RHO<br>RHO<br>RHO<br>RHO<br>RHO<br>RHO<br>RHO | STORAGECLAS<br>gp3-csi<br>gp3-csi<br>gp3-csi<br>gp3-csi<br>gp3-csi<br>gp3-csi<br>gp3-csi | S VOLUMEATT<br>CURSETS<br>CURSETS<br>CURSETS<br>CURSETS<br>CURSETS<br>CURSETS | RIBUTESCLASS                                  | AGE<br>2d16h<br>2d16h<br>2dh<br>44b<br>44h<br>44h |
|---|--|--|---|---|--|--|---|--|---|---|---|
| [root@localhost hcp-testing]# oc_get<br>NAME<br>prc-2dc6de48-5916-411e-9cb3-99598f50b<br>prc-647375d-d377-456d-83a8-306e433a8<br>prc-7d949aef=e00d-409a-8154-514e485f<br>prc-820498ta-65ef-452b-bf90-teae6fe10  | pv CAPACITY<br>e4c 3061<br>7% 3061<br>ab2 10061<br>2c1 3061  | ACCESS MODES<br>RAD<br>RAD<br>RAD  | RECLAIM POLIC<br>Delete<br>Delete<br>Delete   | Y STATUS<br>Bound<br>Bound<br>Bound   | CLAIM<br>openshift-virtualization-os<br>openshift-virtualization-os<br>openshift-monitoring/prometi<br>openshift-virtualization-os   | -images/rhe<br>-images/fed<br>heus-data-p<br>-images/ceo           | 18-0652df0e6259<br>ora-21a6f3e628c<br>rometheus-k0s-1<br>tos-stream-d82     | d B<br>ff4a141a4   | TORAGECLASS<br>p3-C51<br>p3-C51<br>p3-C51<br>p3-C51                           | VOLUMEATTREE<br>Cunset><br>Cunset><br>Cunset> | UTESCLASS   |
| pvc-9a4553a5-0709-440a-8a90-990384c07<br>pvc-deb61444-cb3f-4496-807d-390028400<br>pvc-f4374ce7-568d-4afc-bb35-0228cf454<br>[root@localbost hcp-testing]# _  | 624 10001<br>c16 3001<br>4d4 3001  | Ruo<br>Ruo<br>Ruo  | Delete<br>Delete  | Bound<br>Bound<br>Bound   | openshift-monitoring/promet<br>openshift-virtualization-os<br>openshift-virtualization-os  | heus-data-p<br>-images/cen<br>-images/rhe                          | rometheus-kBs-0<br>tos-stream9-bae<br>19-2521bd116e64                       | 111cdd5a g   | p3-csi<br>p3-csi<br>p3-csi  | cunset><br>cunset><br>cunset>                 |   |

#### d. Verify that applications can create PV using Trident CSI.

Create a PVC using the pvc-trident.yaml file provided in the **fsx** folder.

```
pvc-trident.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
   name: basic
spec:
   accessModes:
    - ReadWriteMany
resources:
    requests:
     storage: 10Gi
storageClassName: trident-csi
```

You can issue the following commands to create a pvc and verify that it has been created.

| [root@<br>[root@<br>persist<br>[root@ | localhost<br>localhost<br>tentvolume<br>localhost | hcp-testing]#<br>hcp-testing]# oc create -f pvc-trident.yam]<br>eclaim/basic created<br>hcp-testing]# oc get pvc -n trident | l -n triden | t            |              |                       |     |
|---------------------------------------|---|---|-------------|--------------|--------------|-----------------------|-----|
| NAME                                  | STATUS  | VOLUME  | CAPACITY    | ACCESS MODES | STORAGECLASS | VOLUMEATTRIBUTESCLASS | AGE |
| basic                                 | Bound   | pvc-adb709b8-fe12-4d4e-9a6b-2afb345bad29  | 10Gi        | RWX          | trident-csi  | <unset></unset>       | 95  |

#### 5. Deploy a sample Postgresql stateful application

#### a. Use helm to install postgresql

```
$ helm install postgresql bitnami/postgresql -n postgresql --create
-namespace
```

| vod@localhost hcp-testing]# helm install postgresql bitnami/postgresql -n postgresqlcreate-namespace  |
|---|
| WE: postgresql  |
| ST DEPLOYED: Non OCt 14 96:52:58 2024   |
| VEDYALE: postgresq1<br>Attic: den land  |
|   |
| ST SUITE Mone   |
| IFS:  |
| NART NAME: postgresal   |
| IART VERSION: 15.5.21   |
| P VERSION: 16.4.0   |
| Please be patient while the chart is being deployed **  |
| stgreSQL can be accessed via port 5432 on the following DNS names from within your cluster:   |
| postgresql.postgresql.svc.cluster.local - Read/Write connection   |
| get the password for "postgres" run:  |
| export POSTGRES_PASSWORD=\$(kubectl get secretnamespace postgresql postgresql -o jsonpath="(.data.postgres-password)"   base64 -d)  |
| connect to your database run the following command:   |
| <pre>kubectl run postgresql-clientrmtty -irestart='Never'namespace postgresqlimage docker.io/bitnami/postgresql:16.4.0-debian-12-r0<br/>command psqlhost postgresql -U postgres -d postgres -p 5432</pre>   |
| > NOTE: If you access the container using bash, make sure that you execute "/opt/bitnami/scripts/postgresql/entrypoint.sh /bin/bash" in order to<br>1001} does not exist"   |
| connect to your database from outside the cluster execute the following commands:   |
| kubectl port-forwardnamespace postgresql svc/postgresql 5432:5432 &<br>PGPASSWORD="\$POSTGRES_PASSWORD" psqlhost 127.0.0.1 -U postgres -d postgres -p 5432  |
| RNING: The configured password will be ignored on new installation in case when previous PostgreSQL release was deleted through the helm command.<br>word, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue. |

#### b. Verify that the application pod is running, and a PVC and PV is created for the application.

| [root@localhos | t hcp-te | esting]# oc | get pods | -n postgresql |  |
|----------------|----------|-------------|----------|---------------|--|
| NAME           | READY    | STATUS      | RESTARTS | AGE           |  |
| postgresql-0   | 1/1      | Running     | 0        | 29m           |  |

| NAME                   | STATUS     | VOLUME               | a7-4a4a-b935-f1c090fd8db6 | CAPACITY | ACCESS MODES | STORAGECLASS |
|------------------------|------------|----------------------|---------------------------|----------|--------------|--------------|
| data-postgresql-0      | Bound      | pvc-e3ddd9bd-e6a     |                           | 8Gi      | RWO          | trident-csi  |
| [root@]ocalbost_bon_to | sting]# or | get ny   gren nostgr | aral                      |          |              |              |

#### c. Deploy a Postgresql client

Use the following command to get the password for the postgresql server that was installed.

```
$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql
postgresql -o jsoata.postgres-password}" | base64 -d)
```

Use the following command to run a postgresql client and connect to the server using the password





d. Create a database and a table. Create a schema for the table and insert 2 rows of data into the table.



# Red Hat OpenShift Service on AWS with NetApp ONTAP

This document will outline how to use NetApp ONTAP with the Red Hat OpenShift Service on AWS (ROSA).

## **Create Volume Snapshot**

1. Create a Snapshot of the app volume

In this section, we will show how to create a trident snapshot of the volume associated with the app. This will be a point in time copy of the app data. If the application data is lost, we can recover the data from this point in time copy.

NOTE: This snapshot is stored in the same aggregate as the original volume in ONTAP(on-premises or in the cloud). So if the ONTAP storage aggregate is lost, we cannot recover the app data from its snapshot.

\*\*a. Create a VolumeSnapshotClass

Save the following manifest in a file called volume-snapshot-class.yaml

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
   name: fsx-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

Create a snapshot by using the above manifest.

[root@localhost hcp-testing]# oc create -f volume-snapshot-class.yaml
volumesnapshotclass.snapshot.storage.k8s.io/fsx-snapclass created
[root@localhost hcp-testing]# \_

#### b. Next, create a snapshot

Create a snapshot of the existing PVC by creating VolumeSnapshot to take a point-in-time copy of your Postgresql data. This creates an FSx snapshot that takes almost no space in the filesystem backend. Save the following manifest in a file called volume-snapshot.yaml:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: postgresql-volume-snap-01
spec:
  volumeSnapshotClassName: fsx-snapclass
  source:
    persistentVolumeClaimName: data-postgresql-0
```

#### c. Create the volume snapshot and confirm that it is created

Delete the database to simulate the loss of data (data loss can happen due to a variety of reasons, here we are just simulating it by deleting the database)



d. Delete the database to simulate the loss of data (data loss can happen due to a variety of reasons, here we are just simulating it by deleting the database)



connection to server at "postgresql" (172.30.103.67), port 5432 failed: FATAL: database "erp" does not exist Previous connection kept postgres=# \_

### **Restore from Volume Snapshot**

#### 1. Restore from Snapshot

In this section, we will show how to restore an application from the trident snapshot of the app volume.

#### a. Create a volume clone from the snapshot

To restore the volume to its previous state, you must create a new PVC based on the data in the snapshot you took. To do this, save the following manifest in a file named pvc-clone.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgresql-volume-clone
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: trident-csi
  requests:
    requests:
    storage: 8Gi
dataSource:
    name: postgresql-volume-snap-01
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Create a clone of the volume by creating a PVC using the snapshot as the source using the above manifest. Apply the manifest and ensure that the clone is created.

| [root@localhost hcp-test<br>persistentvolumeclaim/po<br>[root@localhost hcp-test | ing]# oc<br>stgresql-<br>ing]# oc | create -f postgresql-pvc-clone.yaml -n post<br>volume-clone created<br>get pvc -n postgresql | gresql   |              |              |
|--|-----------------------------------|--|----------|--------------|--------------|
| NAME   | STATUS                            | VOLUME   | CAPACITY | ACCESS MODES | STORAGECLASS |
| data-postgresql-0  | Bound                             | pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8db6   | 8Gi      | RWO          | trident-csi  |
| postgresql-volume-clone<br>[root@localhost hcp-test:                             | Bound<br>ing]# _                  | pvc-b38fbc54-55dc-47e8-934d-47f181fddac6   | 8Gi      | RWO          | trident-csi  |

#### b. Delete the original postgresql installation

```
[root@localhost hcp-testing]#
[root@localhost hcp-testing]# helm uninstall postgresql -n postgresql
release "postgresql" uninstalled
[root@localhost hcp-testing]# oc get pods -n postgresql
No resources found in postgresql namespace.
[root@localhost hcp-testing]# _
```

#### c. Create a new postgresql application using the new clone PVC

```
$ helm install postgresql bitnami/postgresql --set
primary.persistence.enabled=true --set
primary.persistence.existingClaim=postgresql-volume-clone -n postgresql
```

| [root@localhost hcp-testing]#<br>[root@localhost hcp-testing]# helm install postgresql bitnami/postgresqlset primary.persistence.enabled=true \<br>>set primary.persistence.existingClaim=postgresql-volume-clone -n postgresql<br>NAME: noctangesql |
|--|
| LAST DEPLOYED: Mon Oct 14 12:03:31 2024  |
| NAMESPACE: postgresgi<br>STATUS: deployed  |
| REVISION: 1<br>TEST SUTTE: None  |
| NOTES:   |
| CHART NAME: postgresql<br>CHART VERSION: 15.5.21   |
| APP VERSION: 16.4.0  |
| ** Please be patient while the chart is being deployed **  |
| PostgreSQL can be accessed via port 5432 on the following DNS names from within your cluster:  |
| postgresql.postgresql.svc.cluster.local - Read/Write connection  |
| To get the password for "postgres" run:  |
| export POSTGRES_PASSWORD=\$(kubectl get secretnamespace postgresql postgresql -o jsonpath="{.data.postgres-password}"   bas  |
| To connect to your database run the following command:   |
| <pre>kubectl run postgresql-clientrmtty -irestart='Never'namespace postgresqlimage docker.io/bitnami/postgresql:16<br/>command psqlhost postgresql -U postgres -d postgres -p 5432</pre>   |
| > NOTE: If you access the container using bash, make sure that you execute "/opt/bitnami/scripts/postgresql/entrypoint.sh /b:<br>1001} does not exist"   |
| To connect to your database from outside the cluster execute the following commands:   |
| kubectl port-forwardnamespace postgresql svc/postgresql 5432:5432 &<br>PGPASSWORD="\$POSTGRES_PASSWORD" psqlhost 127.0.0.1 -U postgres -d postgres -p 5432   |
| WARNING: The configured password will be ignored on new installation in case when previous PostgreSQL release was deleted throug<br>sword, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.    |
| WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production  |
| - primary.resources  |
| - readReplicas.resources   |
| <pre>[root@localhost hcp-testing]# _</pre>   |
|  |

### d. Verify that the application pod is in the running state

| [root@localhos | st hcp-te | esting]# oc | get pods | -n postgresql |
|----------------|-----------|-------------|----------|---------------|
| NAME           | READY     | STATUS      | RESTARTS | AGE           |
| oostgresql-0   | 1/1       | Running     | 0        | 2m1s          |
| [root@localhos | st hcp-te | esting]#    |          |               |

e. Verify that the pod uses the clone as its PVC

root@localhost hcp-testing]# root@localhost hcp-testing]# oc describe pod/postgresql-0 -n postgresql\_

| ContainersRead   | ly Tru                      | ie       |                            |                                     |            |
|------------------|-----------------------------|----------|----------------------------|-------------------------------------|------------|
| PodScheduled     | Tru                         | ie       |                            |                                     |            |
| Volumes:         |                             |          |                            |                                     |            |
| empty-dir:       |                             |          |                            |                                     |            |
| Type:            | EmptyDir (a tempo           | orary di | rectory that shares a poo  | l's lifetime)                       |            |
| Medium:          |                             |          |                            |                                     |            |
| SizeLimit:       | <unset></unset>             |          |                            |                                     |            |
| dshm:            |                             |          |                            |                                     |            |
| Type:<br>Medium: | EmptyDir (a tempo<br>Memory | orary di | rectory that shares a poo  | l's lifetime)                       |            |
| SizeLimit:       | <unset></unset>             |          |                            |                                     |            |
| data:            |                             |          |                            |                                     |            |
| Type:            | PersistentVolume            | Claim (  | a reference to a Persiste  | entVolumeClaim in the same namespa  | ace)       |
| ClaimName:       | postgresql-volum            | e-clone  |                            |                                     |            |
| ReadOnly:        | false                       |          |                            |                                     |            |
| QoS Class:       | Burstable                   |          |                            |                                     |            |
| Node-Selectors:  | <none></none>               |          | =                          |                                     |            |
| Tolerations:     | node.kubernetes.            | io/memc  | ory-pressure:NoSchedule op | )=Exists                            |            |
|                  | node.kubernetes.            | io/not-  | ready:NoExecute op=Exists  | for 300s                            |            |
|                  | node.kubernetes.            | 10/unre  | achable:Notxecute op=txis  | its for 300s                        |            |
| events:          |                             |          |                            |                                     |            |
| Type Reason      | 1                           | Age      | From                       | message                             |            |
| Normal Cohode    |                             | 3-55-    | default estedules          | Consectually and extensed           | 1/2224-222 |
| wormal Schedu    | ited                        | 30555    | default-scheduler          | Successfully assigned postgresd     | L/postgres |
| Normal Succos    | cfulAttachVolumo            | 20540    | attachdotach_controllor    | AttachVolume Attach succeeded for   |            |
| R-03/d-17f181fd  | ac6"                        | 5111545  | actachuetach-controller    | Actacityorume. Actacit succeeded to | of vorume  |
| Normal Added     | nterface                    | 3m/13c   | multus                     | Add ath0 [10 129 2 126/23] from     | ovn-kuber  |
| Normal Pulled    | incerrace.                  | 3m43s    | kubelet                    | Container image "docker io/bitn     | ami/nostgr |
| r0" already pres | ent on machine              | 200,02.2 | RODELES                    | concurrer image worker i zo/ bi en  | mat hoseB. |
| Normal Create    | d                           | 3m42s    | kubelet                    | Created container postgresgl        | Activat    |
| Normal Starte    | ed .                        | 3m42s    | kubelet                    | Started container postgresql        | Go to Set  |
| [root@localhost  | hcp-testing]#               |          |                            |                                     |            |

f) To validate that the database has been restored as expected, go back to the container console and show the existing databases

| [root@local<br>\$POSTGRES_P/<br>Warning: wor<br>capabilitic<br>Root=true),<br>If you don't<br>postgres=#  | host hcp-te<br>ASSWORD" -<br>uld violate<br>es (contain<br>seccompPro<br>t see a com<br>\l   | sting]# kub<br>-command<br>PodSecurity<br>er "postgre<br>file (pod o<br>mand prompt | ectl run postgresq<br>psglhost postg<br>y "restricted:v1.2<br>sql-client" must s<br>r container "postg<br>, try pressing ent | l-clientrm<br>resql -U postg<br>4": allowPrivi<br>et securityCon<br>resql-client"<br>er.<br>List of da | tty -ires<br>res -d postgre<br>legeEscalation<br>text.capabilit<br>must set secur<br>tabases | tart='Never'<br>s -p 5432<br>l= false (co<br>ies.drop=["AL<br>ityContext.se | namespace<br>ntainer "pos<br>L"]), runAsN<br>ccompProfile | postgresqlimage docke<br>tgresql-client" must set<br>onRoot != true (pod or c<br>.type to "RuntimeDefault" | r.io/bitnami/postgresql:<br>securityContext.allowPr<br>ontainer "postgresql-cli<br>" or "Localhost") |
|---|--|---|--|--|--|---|---|--|--|
|   |  | +   |  |  |  |   | +   | +  |  |
| erp   | postgres   | L UTER  | 1100   | en_us.urr-8  | en_us.urr-8  | -2  |   | *  |  |
| postgres  | postgres   |   | 1100   | en_us.urr-8  | en_US.UIF-8  |   |   | and another and  |  |
| cempiaceo   | postgres   | 0170  | 1100   | en_05.01F-8  | en_us.urr-8  |   |   | hostanas-CTe/nostanas  |  |
| Second Second   | 0000000000000000   | 11750   | 1160   | IN US INTO O   | AT US ITTE O   |   | - 1   | postgresscic/postgres  |  |
| cempiatei   | postgres   | Ulra  | 1100   | en_us.uir-a  | en_us.uir-a  |   |   | <pre>postgres + postgres_CTc/postgres</pre>  |  |
| (A rous)  | <u>,</u>   |   | t i  |  | k 0.   |   |   | postgres=cit/postgres  |  |
| postgres=# '<br>psql (16.2,<br>You are now<br>erp=# \dt<br>L:<br>Schema   1<br>public   pu<br>(1 row)<br>erp=# SELEC<br>id   firstu<br>1   John<br>2   Jane<br>(2 rows) | <pre>\c erp;<br/>server 16.<br/>connected<br/>Name   Ty<br/>ersons   ta<br/>T * FROM PE<br/>name   last<br/>  Doe<br/>  Scot</pre> | 4)<br>to database<br>tions<br>pe   Owne<br>ble   postg<br>ksons;<br>name<br>t       | "erp" as user "po<br>r<br>res  | stgres".   |  |   |   |  |  |

# Demo video

Amazon FSx for NetApp ONTAP wth Red Hat OpenShift Service on AWS using Hosted Control Plane

More videos on Red Hat OpenShift and OpenShift solutions can be found here.

#### **Copyright information**

Copyright © 2024 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.