



# Red Hat OpenShift Service en AWS con FSxN

NetApp Solutions

NetApp  
January 06, 2025

This PDF was generated from [https://docs.netapp.com/es-es/netapp-solutions/containers/rh-os-n\\_use\\_case\\_rosa\\_solution\\_overview.html](https://docs.netapp.com/es-es/netapp-solutions/containers/rh-os-n_use_case_rosa_solution_overview.html) on January 06, 2025. Always check docs.netapp.com for the latest.

# Tabla de contenidos

- Red Hat OpenShift Service en AWS con FSxN ..... 1
- Red Hat OpenShift Service en AWS con NetApp ONTAP ..... 1
- Red Hat OpenShift Service en AWS con NetApp ONTAP ..... 11

# Red Hat OpenShift Service en AWS con FSxN

## Red Hat OpenShift Service en AWS con NetApp ONTAP

### Descripción general

En este apartado, mostraremos cómo utilizar FSx para ONTAP como capa de almacenamiento persistente para aplicaciones que se ejecutan en ROSA. Mostrará la instalación del controlador CSI de NetApp Trident en un clúster ROSA, el aprovisionamiento de un sistema de archivos FSX para ONTAP y el despliegue de una aplicación con estado de ejemplo. También mostrará estrategias para realizar backups y restaurar los datos de sus aplicaciones. Con esta solución integrada, puede establecer un marco de almacenamiento compartido que se pueda escalar sin esfuerzo en las Azure, lo que simplifica los procesos de escalado, protección y restauración de sus datos con el controlador CSI de Trident.

### Requisitos previos

- ["Cuenta de AWS"](#)
- ["Una cuenta de Red Hat"](#)
- Usuario de IAM ["con los permisos adecuados"](#) para crear y acceder al clúster ROSA
- ["CLI DE AWS"](#)
- ["ROSA CLI"](#)
- ["Interfaz de línea de comandos de OpenShift" \(oc\)](#)
- Timón 3 ["documentación"](#)
- ["UN GRUPO DE HCP ROSA"](#)
- ["Acceso a la consola web de Red Hat OpenShift"](#)

Este diagrama muestra el cluster ROSA desplegado en varias AZs. Los nodos maestros del clúster ROSA, los nodos de infraestructura están en VPC de Red Hat, mientras que los nodos de trabajo están en una VPC en la cuenta del cliente. Crearemos un sistema de archivos FSx para ONTAP dentro de la misma VPC e instalaremos el controlador Trident en el clúster ROSA, permitiendo que todas las subredes de esta VPC se conecten al sistema de archivos.



## Configuración inicial

### 1. Aprovisionamiento FSx para NetApp ONTAP

Cree un FSX multi-AZ para NetApp ONTAP en el mismo VPC que el clúster ROSA. Hay varias maneras de hacer esto. Se proporcionan los detalles de la creación de FSxN mediante una pila de CloudFormation

#### A. Clone el repositorio de GitHub

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

**B. Ejecute la pila CloudFormation** Ejecute el siguiente comando reemplazando los valores de los parámetros por sus propios valores:

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

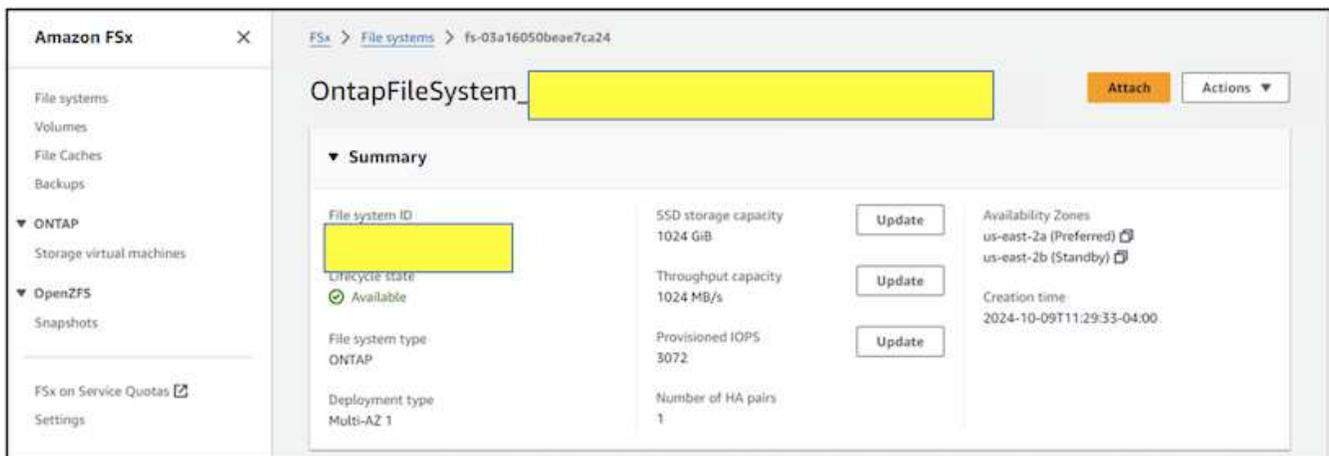
```

$ aws cloudformation create-stack \
  --stack-name ROSA-FSXONTAP \
  --template-body file:///./FSxONTAP.yaml \
  --region <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 \
  ParameterKey=FSxAllowedCIDR,ParameterValue=[your_allowed_CIDR] \
  ParameterKey=FsxAdminPassword,ParameterValue=[Define Admin password] \
  ParameterKey=SvmAdminPassword,ParameterValue=[Define SVM password] \
  --capabilities CAPABILITY_NAMED_IAM

```

Where : REGION-NAME: Igual que la región donde se despliega el cluster ROSA subnet1\_ID : Id de la subred preferida para FSxN subnet2\_ID: id de la subred en espera para FSxN VPC\_ID: id de la VPC donde se despliega el cluster ROSA routetable1\_ID, routetable2\_ID: id de las tablas de ruta asociadas a las subredes seleccionadas arriba de your\_allowed\_CIDR: Rango de acceso de ONTAP permitido para las reglas de entrada de los grupos de seguridad de control de FSX. Puedes usar 0,0.0.0/0 o cualquier CIDR apropiado para permitir que todo el tráfico acceda a los puertos específicos de FSX para ONTAP. Define la contraseña de administrador: Una contraseña para iniciar sesión en FSxN Definir la contraseña de SVM: Una contraseña para iniciar sesión en la SVM que se creará.

Compruebe que el sistema de archivos y la máquina virtual de almacenamiento (SVM) se han creado con la consola de Amazon FSx, que se muestra a continuación:



## 2. Instale y configure el controlador Trident CSI para el clúster ROSA

### A. Añada el repositorio Trident Helm

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

## B. Instale Trident usando HELM

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



Según la versión que instale, el parámetro VERSION deberá cambiarse en el comando que se muestra. Consulte la "[documentación](#)" para obtener el número de versión correcto. Para obtener más métodos de instalación de Trident, consulte Trident "[documentación](#)".

## C. Verifique que todos los pods de Trident estén en estado de ejecución

```
[root@localhost hcp-testing]#  
[root@localhost hcp-testing]#  
[root@localhost hcp-testing]# oc get 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  
trident-operator-545869857c-kgc7p   1/1     Running   0           19h  
[root@localhost hcp-testing]#
```

## 3. Configure el backend CSI de Trident para usar FSx para ONTAP (NAS de ONTAP)

La configuración back-end de Trident indica a Trident cómo se comunica con el sistema de almacenamiento (en este caso, FSx para ONTAP). Para crear el backend, proporcionaremos las credenciales de la máquina virtual de almacenamiento a la que conectarse, junto con la gestión del clúster y las interfaces de datos NFS. Usaremos el "[controlador ontap-nas](#)" para aprovisionar volúmenes de almacenamiento en el sistema de archivos FSx.

### a.. Primero, cree un secreto para las credenciales de SVM usando el siguiente 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>
```



También puede recuperar la contraseña de SVM creada para FSxN desde AWS Secrets Manager, como se muestra a continuación.

The screenshot shows the AWS Secrets Manager console. At the top, there's a search bar with the text "Filter secrets by name, description, tag key, tag value, owning service or primary Region". Below the search bar is a table with three columns: "Secret name", "Description", and "Last retrieved (UTC)". The table contains two rows of secrets:

Secret name	Description	Last retrieved (UTC)
HCP-ROSA-FSXONTAP-SVMAdminPassword	SVMAdminPassword	October 9, 2024
HCP-ROSA-FSXONTAP-FsxAdminPassword	FsxAdminPassword	-

The screenshot shows the details page for the secret "HCP-ROSA-FSXONTAP-SVMAdminPassword". The page has a breadcrumb trail: "AWS Secrets Manager > Secrets > HCP-ROSA-FSXONTAP-SVMAdminPassword". The main heading is "HCP-ROSA-FSXONTAP-SVMAdminPassword". Below this is a "Secret details" section with the following information:

- Encryption key: aws/secretsmanager
- Secret name: HCP-ROSA-FSXONTAP-SVMAdminPassword
- Secret ARN: arn:aws:secretsmanager:us-east-2:316088182667:secret:HCP-ROSA-FSXONTAP-SVMAdminPassword-ABIUaf
- Secret description: SVMAdminPassword

At the bottom of the details section, there are tabs for "Overview", "Rotation", "Versions", "Replication", and "Tags". Below the tabs is a "Secret value" section with a "Retrieve secret value" button.

**B.A continuación, agregue el secreto para las credenciales de SVM al clúster ROSA usando el siguiente comando**

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

Puede verificar que el secreto se haya agregado en el espacio de nombres de Trident con el siguiente comando

```
$ 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.. A continuación, cree el objeto **backend** para esto, muévase al directorio **fsx** de su repositorio Git clonado. Abra el archivo **backend-ONTAP-nas.yaml**. Reemplace lo siguiente: **ManagementLIF** por el nombre DNS de administración **dataLIF** por el nombre DNS NFS de la SVM de Amazon FSx y **svm** con el nombre SVM. Cree el objeto backend con el siguiente comando.

Cree el objeto backend con el siguiente comando.

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



Puedes obtener el nombre de DNS de gestión, el nombre de DNS de NFS y el nombre de SVM de Amazon FSx Console, como se muestra en la siguiente captura de pantalla

The screenshot shows the Amazon FSx console interface. On the left, there is a navigation menu with options like File systems, Volumes, File Caches, Backups, ONTAP, OpenZFS, and Settings. The main area displays the 'Summary' page for a specific SVM. Key details include: SVM ID (svm-07a733da2584f2045), SVM name (SVM1), UUID (a845e7bf-8653-11ef-8f27-0f43b1500927), File system ID (fs-03a16050beae7ca24), and Resource ARN. The lifecycle state is 'Created'. Below the summary, there are tabs for Endpoints, Administration, Volumes, and Tags. The 'Endpoints' tab is active, showing Management DNS name, NFS DNS name, iSCSI DNS name, Management IP address, NFS IP address, and iSCSI IP addresses.

Field	Value
SVM ID	svm-07a733da2584f2045
Creation time	2024-10-09T11:31:46-04:00
Active Directory	-
SVM name	SVM1
Lifecycle state	Created
UUID	a845e7bf-8653-11ef-8f27-0f43b1500927
Subtype	DEFAULT
File system ID	fs-03a16050beae7ca24
Resource ARN	arn:aws:fsx:us-east-2:316088182667:storage-virtual-machine/fs-03a16050beae7ca24/svm-07a733da2584f2045
Management DNS name	svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonaws.com
Management IP address	198.19.255.182
NFS DNS name	svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonaws.com
NFS IP address	198.19.255.182
iSCSI DNS name	iscsi.svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonaws.com
iSCSI IP addresses	10.10.9.32, 10.10.26.28

d.. Ahora, ejecute el siguiente comando para verificar que se ha creado el objeto backend y que la fase muestra **Bound** y **Status** es **Success**

```
[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
NAME                BACKEND NAME  BACKEND UUID                                PHASE  STATUS
backend-fsx-ontap-nas  fsx-ontap    acc65405-56be-4719-999d-27b448a50e29    Bound  Success
[root@localhost hcp-testing]#
```

**4. Crear clase de almacenamiento** Ahora que el backend de Trident está configurado, puede crear una clase de almacenamiento de Kubernetes para usar el backend. La clase de almacenamiento es un objeto de recurso disponible para el clúster. Describe y clasifica el tipo de almacenamiento que se puede solicitar para una aplicación.

**a.. Revise el archivo storage-class-csi-nas.yaml en la carpeta fsx.**

```
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.. Cree una clase de almacenamiento en el clúster ROSA y verifique que se haya creado la clase de almacenamiento Trident-csi.**

```
[root@localhost hcp-testing]#
[root@localhost hcp-testing]#
[root@localhost hcp-testing]# oc apply -f storage-class-csi-nas.yaml
storageclass.storage.k8s.io/trident-csi created
[root@localhost hcp-testing]# oc get sc
NAME                PROVISIONER          RECLAIMPOLICY  VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
gp2-csi             ebs.csi.aws.com      Delete         WaitForFirstConsumer  true                  2d16h
gp3-csi (default)  ebs.csi.aws.com      Delete         WaitForFirstConsumer  true                  2d16h
trident-csi        csi.trident.netapp.io  Retain         Immediate           true                   4s
[root@localhost hcp-testing]#
```

Esto completa la instalación del controlador CSI de Trident y su conectividad con el sistema de archivos FSx para ONTAP. Ahora puedes implementar una aplicación de estado PostgreSQL de muestra en ROSA usando volúmenes de archivos en FSx para ONTAP.

**c.. Verifique que no haya EVs ni VP creados con la clase de almacenamiento Trident-csi.**

```

[root@localhost hcp-testing]#
[root@localhost hcp-testing]#
[root@localhost hcp-testing]# oc get pvc -A
NAMESPACE          NAME                                     STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS          VOLUMEATTRIBUTESCLASS  AGE
openshift-monitoring  prometheus-data-prometheus-k8s-0      Bound    pvc-9a4553a5-07e9-440a-8a90-99e384c97624  100Gi      RWO            gp3-csi               <unset>                  2d16h
openshift-monitoring  prometheus-data-prometheus-k8s-1      Bound    pvc-79949aef-e00d-4d9a-8b54-514ed85fbab2  100Gi      RWO            gp3-csi               <unset>                  2d16h
openshift-visualization-os-images  centos-stream9-bee11cd5a1              Bound    pvc-96b01a44-cb3f-449b-bd7d-39d020499c16  30Gi       RWO            gp3-csi               <unset>                  24h
openshift-visualization-os-images  centos-stream9-d0244a141a44            Bound    pvc-82908044-c5e8-452b-bf90-10e0e4f102c1  30Gi       RWO            gp3-csi               <unset>                  44h
openshift-visualization-os-images  fedora-21a0f2e628cd                    Bound    pvc-64f375a8-d377-456d-83a0-268e413ae79c  30Gi       RWO            gp3-csi               <unset>                  44h
openshift-visualization-os-images  rhel8-0652df0eb359                      Bound    pvc-2dc6de48-5916-411e-0cb3-99598f50be4c  30Gi       RWO            gp3-csi               <unset>                  44h
openshift-visualization-os-images  rhel9-2521bd116e64                      Bound    pvc-f4374ce7-568d-4afc-b635-0228cf4544d4  30Gi       RWO            gp3-csi               <unset>                  44h
[root@localhost hcp-testing]# oc get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                                                                                               STORAGECLASS          VOLUMEATTRIBUTESCLASS
pvc-2dc6de48-5916-411e-0cb3-99598f50be4c  30Gi      RWO            Delete           Bound    openshift-visualization-os-images/rhel8-0652df0eb359      gp3-csi               <unset>
pvc-64f375a8-d377-456d-83a0-268e413ae79c  30Gi      RWO            Delete           Bound    openshift-visualization-os-images/fedora-21a0f2e628cd     gp3-csi               <unset>
pvc-74949aef-e00d-4d9a-8b54-514ed85fbab2  100Gi     RWO            Delete           Bound    openshift-monitoring/prometheus-data-prometheus-k8s-1    gp3-csi               <unset>
pvc-82b0e84a-e5ef-452b-bf90-10e0e4f102c1  30Gi      RWO            Delete           Bound    openshift-visualization-os-images/centos-stream9-d0244a141a44  gp3-csi               <unset>
pvc-9a4553a5-07e9-440a-8a90-99e384c97624  100Gi     RWO            Delete           Bound    openshift-monitoring/prometheus-data-prometheus-k8s-0    gp3-csi               <unset>
pvc-deb01a44-cb3f-449b-bd7d-39d020499c16  30Gi      RWO            Delete           Bound    openshift-visualization-os-images/centos-stream9-bee11cd5a1  gp3-csi               <unset>
pvc-f4374ce7-568d-4afc-b635-0228cf4544d4  30Gi      RWO            Delete           Bound    openshift-visualization-os-images/rhel9-2521bd116e64     gp3-csi               <unset>
[root@localhost hcp-testing]#

```

**d.. Verifique que las aplicaciones pueden crear PV usando Trident CSI.**

Cree un PVC usando el archivo pvc-Trident.yaml proporcionado en la carpeta **fsx**.

```

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.

```

image:redhat_openshift_container_rosa_image11.png["Cree la PVC de prueba usando Trident"]

```

**5. Implementar una aplicación PostgreSQL de muestra**

**a.. Utilice el timón para instalar postgresql**

```

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

```

```

[root@localhost hcp-testing]# helm install postgresql bitnami/postgresql -n postgresql --create-namespace
NAME: postgresql
LAST DEPLOYED: Mon Oct 14 06:52:58 2024
NAMESPACE: postgresql
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: postgresql
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 secret --namespace postgresql postgresql -o jsonpath="{.data.postgres-password}" | base64 -d)

To connect to your database run the following command:

    kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql:16.4.0-debian-12-r0 --command -- psql --host postgresql -U postgres -d postgres -p 5432

    > NOTE: If you access the container using bash, make sure that you execute "/opt/bitnami/scripts/postgresql/entrypoint.sh /bin/bash" in order to
    1001) does not exist"

To connect to your database from outside the cluster execute the following commands:

    kubectl port-forward --namespace postgresql svc/postgresql 5432:5432 &
    PGPASSWORD="$POSTGRES_PASSWORD" psql --host 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 through the helm command,
password, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.

```

b.. Verifique que el pod de la aplicación se está ejecutando y que se ha creado un PVC y un PV para la aplicación.

```

[root@localhost hcp-testing]# oc get pods -n postgresql
NAME                READY   STATUS    RESTARTS   AGE
postgresql-0       1/1    Running   0           29m

```

```

[root@localhost hcp-testing]# oc get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
data-postgresql-0  Bound   pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8db6   8Gi        RWO             trident-csi

```

```

[root@localhost hcp-testing]# oc get pv | grep postgresql
pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8db6   8Gi        RWO             Retain        Bound        postgresql/data-postgresql-0
csi                                     4h20m
[root@localhost hcp-testing]#

```

c.. Implementar un cliente PostgreSQL

Utilice el siguiente comando para obtener la contraseña del servidor postgresql que se instaló.

```

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

```

Utilice el siguiente comando para ejecutar un cliente postgresql y conectarse al servidor usando la contraseña

```
$ kubectl run postgresql-client --rm --tty -i --restart='Never'  
--namespace postgresql --image docker.io/bitnami/postgresql:16.2.0-debian-  
11-r1 --env="PGPASSWORD=$POSTGRES_PASSWORD" \  
> --command -- psql --host postgresql -U postgres -d postgres -p 5432
```

```
[root@localhost hcp-testing]# kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql:16.2.0-debian-11-r1 --env="PGPASSWORD=$POSTGRES_PASSWORD" \  
> --command -- psql --host postgresql -U postgres -d postgres -p 5432  
Warning: would violate PodSecurity "restricted:vl.24": allowPrivilegeEscalation != false (container "postgresql-client" must set securityContext.allowPrivilegeEscalation to true), runAsNonRoot != true (pod or container "postgresql-client" must set securityContext.runAsNonRoot to true), seccompProfile (pod or container "postgresql-client" must set securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost"), If you don't see a command prompt, try pressing enter.
```

d.. Crear una base de datos y una tabla. Cree un esquema para la tabla e inserte 2 filas de datos en la tabla.

```
postgres=# CREATE DATABASE erp;  
CREATE DATABASE  
postgres=# \c erp  
psql (16.2, server 16.4)  
You are now connected to database "erp" as user "postgres".  
erp=# CREATE TABLE PERSONS(ID INT PRIMARY KEY NOT NULL, FIRSTNAME TEXT NOT NULL, LASTNAME TEXT NOT NULL);  
CREATE TABLE  
erp=# INSERT INTO PERSONS VALUES(1, 'John', 'Doe');  
INSERT 0 1  
erp=# \dt  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | persons | table | postgres  
(1 row)
```

```
erp=# SELECT * FROM PERSONS;  
 id | firstname | lastname  
----+-----+-----  
  1 | John      | Doe  
(1 row)
```

```

erp=# INSERT INTO PERSONS VALUES(2, 'Jane', 'Scott');
INSERT 0 1
erp=# SELECT * from PERSONS;
 id | first_name | last_name
-----+-----+-----
  1 | John       | Doe
  2 | Jane       | Scott
(2 rows)

```

## Red Hat OpenShift Service en AWS con NetApp ONTAP

En este documento se describe cómo utilizar NetApp ONTAP con el servicio Red Hat OpenShift en AWS (ROSA).

### Crear snapshot de volumen

**1. Crear una instantánea del volumen de la aplicación** En esta sección, mostraremos cómo crear una instantánea de Trident del volumen asociado a la aplicación. Esta será una copia de punto en el tiempo de los datos de la aplicación. Si se pierden los datos de la aplicación, podemos recuperar los datos de esta copia puntual. **NOTA:** Esta copia Snapshot se almacena en el mismo agregado que el volumen original en ONTAP (en las instalaciones o en el cloud). Por lo tanto, si se pierde el agregado de almacenamiento de ONTAP, no podemos recuperar los datos de la aplicación a partir de su instantánea.

**\*\*a..** Cree un VolumeSnapshotClass Guarde el siguiente manifiesto en un archivo denominado volume-snapshot-class.yaml

```

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

```

Cree una instantánea mediante el manifiesto anterior.

```

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

```

**b.. A continuación, cree una instantánea** Cree una instantánea de la RVP existente mediante la creación de VolumeSnapshot para tomar una copia puntual de sus datos PostgreSQL. Esto crea una instantánea de FSX que casi no ocupa espacio en el backend del sistema de archivos. Guarde el siguiente manifiesto en un

archivo llamado 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.. Cree la snapshot del volumen y confirme que se ha creado

Eliminar la base de datos para simular la pérdida de datos (la pérdida de datos puede ocurrir debido a una variedad de razones, aquí solo estamos simulando borrando la base de datos)

```
[root@localhost hcp-testing]#
[root@localhost hcp-testing]# oc create -f postgresql-volume-snapshot.yaml -n postgresql
volumesnapshot.snapshot.storage.k8s.io/postgresql-volume-snap-01 created
[root@localhost hcp-testing]# oc get VolumeSnapshot -n postgresql
NAME                                READYTOUSE SOURCEPVC          SOURCESNAPSHOTCONTENT  RESTORESIZE  SNAPSHOTCLASS  SNAPSHOTCONTENT
postgresql-volume-snap-01          true          data-postgresql-0    41500Ki          fsx-snapclass  snapcontent-5baf4337-922e-4318-be82-6db822082339
[root@localhost hcp-testing]#
```

d.. Eliminar la base de datos para simular la pérdida de datos (la pérdida de datos puede ocurrir debido a una variedad de razones, aquí solo estamos simulando borrando la base de datos)

```
postgres=# \c erp;
psql (16.2, server 16.4)
You are now connected to database "erp" as user "postgres".
erp=# SELECT * FROM persons;
 id | firstname | lastname
----+-----+-----
  1 | John      | Doe
  2 | Jane      | Scott
(2 rows)
```

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

## Restaurar desde Snapshot de volumen

1. Restaurar desde la instantánea En esta sección, mostraremos cómo restaurar una aplicación desde la instantánea de Trident del volumen de la aplicación.

### a.. Cree un clon de volumen a partir de la copia de Snapshot

Para restaurar el volumen a su estado anterior, debe crear una RVP nueva basada en los datos de la copia de Snapshot realizada. Para ello, guarde el siguiente manifiesto en un archivo llamado pvc-clone.yaml

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

Cree un clon del volumen creando una RVP utilizando la snapshot como origen mediante el manifiesto anterior. Aplique el manifiesto y asegúrese de que se haya creado el clon.

```
[root@localhost hcp-testing]# oc create -f postgresql-pvc-clone.yaml -n postgresql
persistentvolumeclaim/postgresql-volume-clone created
[root@localhost hcp-testing]# oc get pvc -n postgresql
NAME                                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
data-postgresql-0                   Bound    pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8db6   8Gi        RWO            trident-csi
postgresql-volume-clone             Bound    pvc-b38fbc54-55dc-47e8-934d-47f181fddac6   8Gi        RWO            trident-csi
[root@localhost hcp-testing]#
```

### b.. Elimine la instalación postgresql original

```
[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.. Cree una nueva aplicación postgresql usando el nuevo clon PVC

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

```

[root@localhost hcp-testing]#
[root@localhost hcp-testing]# helm install postgresql bitnami/postgresql --set primary.persistence.enabled=true \
> --set primary.persistence.existingClaim=postgresql-volume-clone -n postgresql
NAME: postgresql
LAST DEPLOYED: Mon Oct 14 12:03:31 2024
NAMESPACE: postgresql
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: postgresql
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 secret --namespace postgresql postgresql -o jsonpath="{.data.postgres-password}" | base64 -d)

To connect to your database run the following command:

    kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql:16
    --command -- psql --host postgresql -U postgres -d postgres -p 5432

    > NOTE: If you access the container using bash, make sure that you execute "/opt/bitnami/scripts/postgresql/entrypoint.sh /bin/bash"
    instead of "/bin/bash" to ensure that the environment is properly initialized.

To connect to your database from outside the cluster execute the following commands:

    kubectl port-forward --namespace postgresql svc/postgresql 5432:5432 &
    PGPASSWORD="$POSTGRES_PASSWORD" psql --host 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 through
WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production
ing to your workload needs:
- primary.resources
- readReplicas.resources
+info https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
[root@localhost hcp-testing]#

```

d.. Compruebe que el pod de aplicación está en estado de ejecución

```

[root@localhost hcp-testing]# oc get pods -n postgresql
NAME                READY   STATUS    RESTARTS   AGE
postgresql-0       1/1    Running   0          2m1s
[root@localhost hcp-testing]#

```

e.. Compruebe que el pod utiliza el clon como PVC

```

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

```

```

ContainersReady      True
PodScheduled         True
Volumes:
empty-dir:
  Type:          EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
  SizeLimit:    <unset>
dshm:
  Type:          EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:        Memory
  SizeLimit:    <unset>
data:
  Type:          PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
  ClaimName:    postgresql-volume-clone
  ReadOnly:     false
QoS Class:           Burstable
Node-Selectors:     <none>
Tolerations:        node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason          Age   From          Message
  ----    -
Normal   Scheduled       3m55s default-scheduler   Successfully assigned postgresql/postgres to ip-10-0-1-1.us-east-2.compute.internal
Normal   SuccessfulAttachVolume  3m54s attachdetach-controller   AttachVolume.Attach succeeded for volume "pvc-83-934d-47f181fddac6"
Normal   AddedInterface   3m43s multus         Add eth0 [10.129.2.126/23] from ovn-kubernetes
Normal   Pulled           3m43s kubelet        Container image "docker.io/bitnami/postgresql" already present on machine
Normal   Created          3m42s kubelet        Created container postgresql
Normal   Started          3m42s kubelet        Started container postgresql
[root@localhost hcp-testing]#

```

f) Para validar que la base de datos se ha restaurado del modo esperado, vuelva a la consola de contenedores y muestre las bases de datos existentes

```

[root@localhost hcp-testing]# kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql:16.4 --env POSTGRES_PASSWORD=postgres --command -- psql --host postgresql -U postgres -d postgres -p 5432
Warning: would violate PodSecurity "restricted:v1.24": allowPrivilegeEscalation != false (container "postgresql-client" must set securityContext.allowPrivilegeEscalation to false), capabilities (container "postgresql-client" must set securityContext.capabilities.drop=["ALL"]), runAsNonRoot != true (pod or container "postgresql-client" must set securityContext.runAsNonRoot to true), seccompProfile (pod or container "postgresql-client" must set securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost")
If you don't see a command prompt, try pressing enter.
postgres=# \l
          List of databases
 name | owner  | encoding | locale provider | collate | ctype  | icu locale | icu rules | access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
 erp  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |              |              |
 postgres | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |              |              |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |              |              | =c/postgres,+postgres=Ctc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |              |              | =c/postgres,+postgres=Ctc/postgres
(4 rows)

postgres=# \c erp;
psql (16.2, server 16.4)
You are now connected to database "erp" as user "postgres".
erp=# \dt
          List of relations
 schema | name  | type  | owner
-----+-----+-----+-----
 public | persons | table | postgres
(1 row)

erp=# SELECT * FROM PERSONS;
 id | first_name | last_name
----+-----+-----
  1 | John       | Doe
  2 | Jane       | Scott
(2 rows)

```

## Vídeo de demostración

Amazon FSx para NetApp ONTAP con el servicio Red Hat OpenShift en AWS mediante el plano de control alojado

Puede encontrar más vídeos sobre las soluciones de Red Hat OpenShift y OpenShift "[aquí](#)".

## Información de copyright

Copyright © 2025 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPTIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

## Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.