Cloud híbrido con componentes gestionados por el proveedor

NetApp Solutions

NetApp
April 26, 2024

Tabla de contenidos

Soluciones de multicloud híbrido de NetApp para cargas de trabajo de contenedores de Red Hat OpenShift

Descripción general

NetApp está viendo un aumento significativo en los clientes que modernizan sus aplicaciones empresariales heredadas y crean nuevas aplicaciones con contenedores y plataformas de orquestación creadas en torno a Kubernetes. Red Hat OpenShift Container Platform es un ejemplo que consideramos adoptado por muchos de nuestros clientes.

A medida que más y más clientes empiezan a adoptar contenedores dentro de sus empresas, NetApp está perfectamente posicionada para poder dar respuesta a las necesidades de almacenamiento persistente de sus aplicaciones con estado y las necesidades de gestión de datos clásicas como la protección de datos, la seguridad de datos y la migración de datos. Sin embargo, estas necesidades se satisfacen utilizando diferentes estrategias, herramientas y métodos.

Las opciones de almacenamiento basado en ONTAP de NetApp que se enumeran a continuación, ofrecen seguridad, protección de datos, fiabilidad y flexibilidad para implementaciones de contenedores y Kubernetes.

- Almacenamiento autogestionado en las instalaciones:
  - Almacenamiento estructural (FAS) de NetApp, cabinas All Flash FAS (AFF), cabina All SAN (ASA) y ONTAP Select
- Almacenamiento gestionado por el proveedor en las instalaciones:
  - NetApp Keystone proporciona almacenamiento como servicio (STaaS)
- Almacenamiento autogestionado en el cloud:
  - Cloud Volumes ONTAP (CVO) de NetApp proporciona almacenamiento autogestionado en los proveedores a hiperescala
- Almacenamiento en el cloud gestionado por el proveedor:
  - Cloud Volumes Service para Google Cloud (CVS), Azure NetApp Files (ANF), Amazon FSx para ONTAP de NetApp ofrecen un almacenamiento totalmente gestionado en los proveedores a hiperescala

ONTAP feature highlights

Storage Administration

- Multi-tenancy
- FlexVol & FlexGroup
- LUN
- Quotas
- ONTAP CLI & API
- System Manager & BlueXP

Performance & Scalability

- FlexCache
- FlexClone
- nconnect, session trunking, multipathing
- Scale-out clusters

Availability & Resilience

- Multi-AZ HA deployment (MetroCluster)
- SnapShot & SnapRestore
- SnapMirror
- SnapMirror Business Continuity
- SnapMirror Cloud

Access Protocols

- NFS –v3, v4, v4.1, v4.2
- SMB – v2, v3
- iSCSI
- Multi-protocol access

Storage Efficiency

- Deduplication & Compression
- Compaction
- Thin provisioning
- Data Tiering (Fabric Pool)

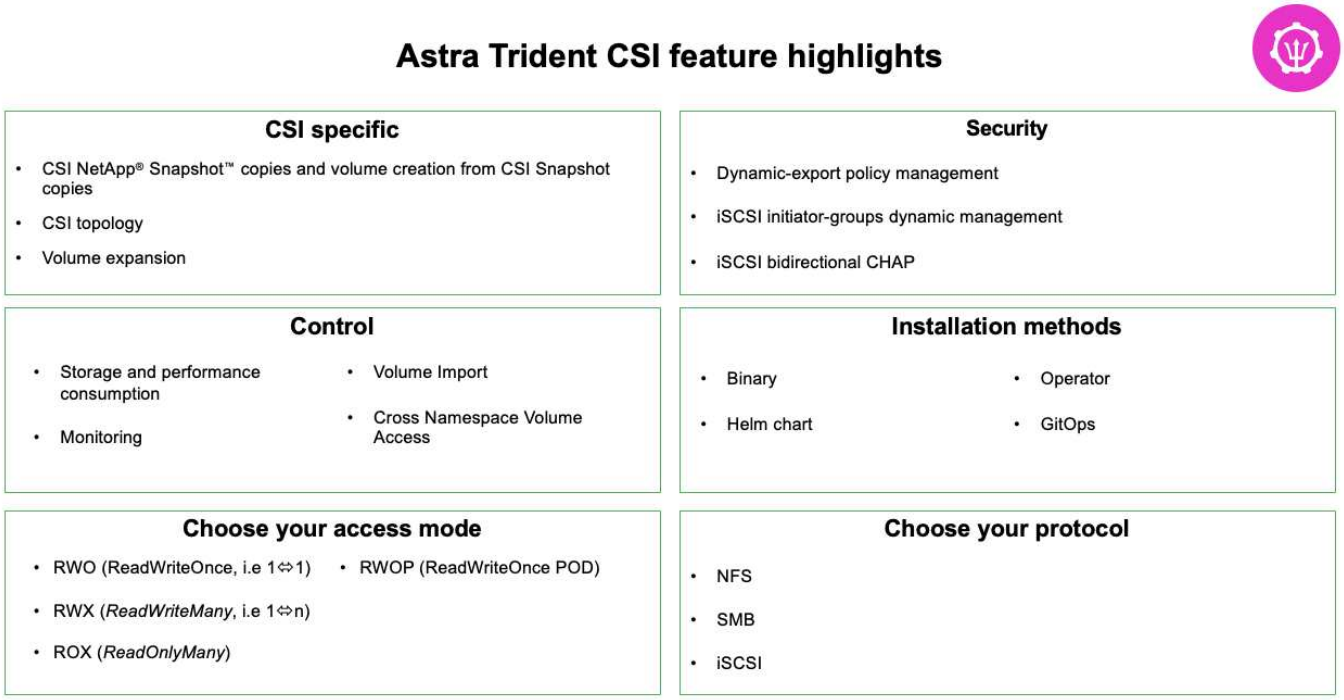
Security & Compliance

- Fpolicy & Vscan
- Active Directory integration
- LDAP & Kerberos
- Certificate based authentication

NetApp BlueXP le permite gestionar todos sus activos de almacenamiento y datos desde una única interfaz o plano de control.

Puedes usar BlueXP para crear y administrar almacenamiento en nube (por ejemplo, Cloud Volumes ONTAP y Azure NetApp Files), para mover, proteger y analizar datos, y para controlar muchos dispositivos de almacenamiento on-premises y en la periferia.

Astra Trident de NetApp es un orquestador de almacenamiento compatible con CSI que permite consumir almacenamiento persistente de forma rápida y sencilla, respaldado por diversas opciones de almacenamiento de NetApp mencionadas anteriormente. Es un software de código abierto que tiene soporte y mantenimiento de NetApp.



Las cargas de trabajo de contenedores vitales para el negocio necesitan más que volúmenes persistentes. Sus requisitos de gestión de datos requieren la protección y la migración de los objetos de aplicaciones kubernetes también.

> Los datos de la aplicación incluyen objetos de kubernetes además de los datos del usuario: Algunos ejemplos son los siguientes: - Objetos de kubernetes como especificaciones de pods, PVCs, despliegues, servicios - objetos de configuración personalizados como mapas de configuración y secretos - datos persistentes como copias Snapshot, copias de seguridad, clones - recursos personalizados como CRS y CRD

Astra Control de NetApp, disponible como software totalmente gestionado y autogestionado, proporciona orquestación para una gestión de datos de aplicaciones sólida. Consulte la "Documentación de Astra" Para obtener más información sobre la familia de productos Astra.

Esta documentación de referencia proporciona la validación de la migración y la protección de aplicaciones basadas en contenedores, puestas en marcha en la plataforma de contenedores RedHat OpenShift, mediante Astra Control Center de NetApp. Además, la solución proporciona detalles de alto nivel para la implementación y el uso de Red Hat Advanced Cluster Management (ACM) para la gestión de las plataformas de contenedores. En el documento también se destacan los detalles de la integración del almacenamiento de NetApp con las plataformas de contenedor Red Hat OpenShift mediante el aprovisionador CSI de Astra Trident. Astra Control Center se pone en marcha en el clúster de concentradores y se utiliza para gestionar las aplicaciones de contenedores y su ciclo de vida de almacenamiento persistente. Por último, proporciona una solución de replicación y conmutación al nodo de respaldo y conmutación de retorno tras recuperación para cargas de trabajo de contenedores en clústeres de Red Hat OpenShift gestionados en AWS (ROSA) utilizando Amazon FSx para NetApp ONTAP (FSxN) como almacenamiento persistente.

Solución de NetApp con cargas de trabajo de la plataforma de contenedores Red Hat OpenShift gestionadas en AWS

Solución de NetApp con cargas de trabajo de la plataforma de contenedores Red Hat OpenShift gestionadas en AWS

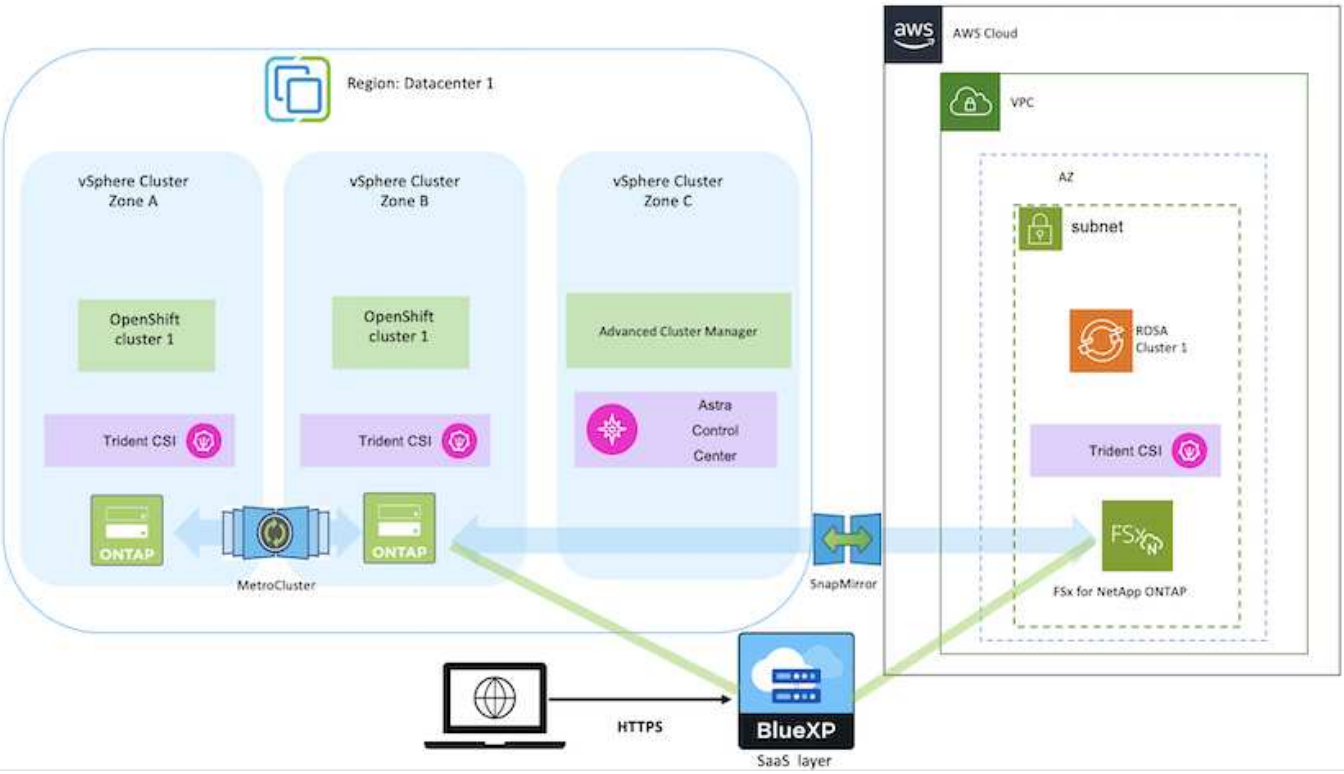
Los clientes pueden haber «nacido en el cloud» o pueden estar en un momento en su proceso de modernización cuando estén listos para mover algunas cargas de trabajo selectas o todas las cargas de trabajo de sus centros de datos al cloud. Pueden elegir usar contenedores OpenShift gestionados por proveedores y almacenamiento NetApp gestionado por proveedores en la nube para ejecutar sus cargas de trabajo. Deben planificar e implementar los clústeres de contenedores Managed Red Hat OpenShift (ROSA) en la nube para un entorno de producción adecuado para sus cargas de trabajo de contenedores. Cuando están en el cloud de AWS, también podrían poner en marcha FSx para ONTAP de NetApp para cubrir las necesidades de almacenamiento.

FSX para ONTAP de NetApp ofrece protección de datos, fiabilidad y flexibilidad para las puestas en marcha de contenedores en AWS. Astra Trident actúa como el aprovisionador de almacenamiento dinámico para consumir el almacenamiento FSxN persistente para las aplicaciones con estado de los clientes.

Como ROSA se puede poner en marcha en modo de alta disponibilidad con nodos del plano de control repartidos por varias zonas de disponibilidad, FSx ONTAP también se puede aprovisionar con la opción Multi-AZ que proporciona alta disponibilidad y protección frente a fallos de AZ.

> No hay cargos de transferencia de datos al acceder a un sistema de archivos Amazon FSx desde la zona de disponibilidad (AZ) preferida del sistema de archivos. Para obtener más información sobre los precios, consulte "aquí".

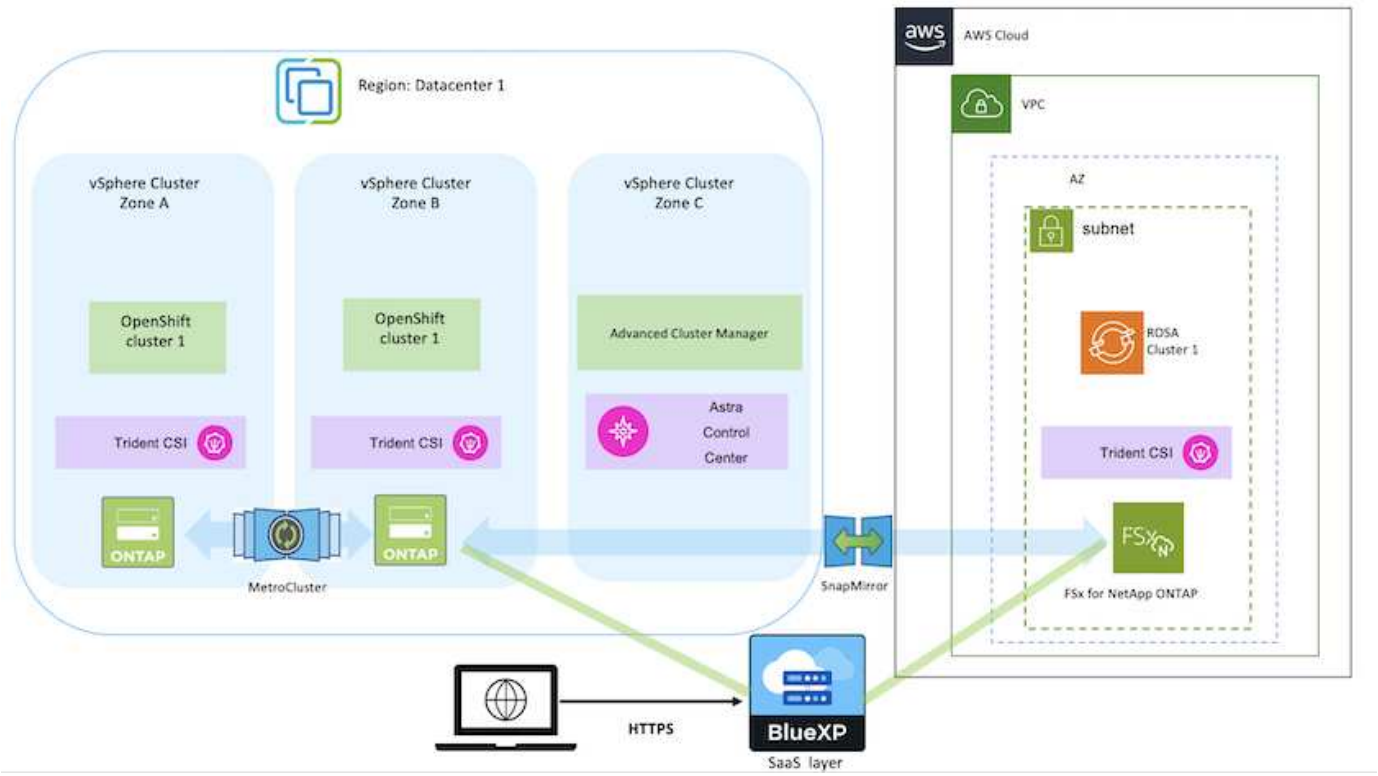
Solución de protección y migración de datos para las cargas de trabajo de contenedores de OpenShift



Implemente y configure la plataforma Managed Red Hat OpenShift Container en AWS

En esta sección se describe un flujo de trabajo de alto nivel de configuración de los clústeres gestionados de Red Hat OpenShift en AWS (ROSA). Muestra el uso de Managed FSx para ONTAP de NetApp (FSxN) como back-end de almacenamiento por parte de Astra Trident para proporcionar volúmenes persistentes. Encontrará más detalles sobre la implementación de FSxN en AWS mediante BlueXP. Además, se incluyen más detalles sobre el uso de BlueXP y OpenShift GitOps (Argo CD) para realizar actividades de protección y migración de datos para las aplicaciones con estado en los clústeres de ROSA.

A continuación se muestra un diagrama que muestra los clústeres ROSA implementados en AWS y utilizando FSxN como almacenamiento back-end.



Esta solución se verificó mediante el uso de dos clústeres ROSA en dos VPC en AWS. Cada clúster ROSA se integró con FSxN mediante Astra Trident. Hay varias formas de implementar los clusters ROSA y FSxN en AWS. Esta descripción de alto nivel de la configuración proporciona enlaces de documentación para el método específico utilizado. Puede consultar los otros métodos en los enlaces correspondientes que se proporcionan en la "sección recursos".

El proceso de configuración puede dividirse en los siguientes pasos:

Instale los clusters ROSA

- Cree dos VPC y configure la conectividad entre iguales entre los VPC.
- Consulte "aquí" Para obtener instrucciones para instalar los clusters ROSA.

Instale FSxN

- Instala FSxN en los PC de BlueXP. Consulte "aquí" Para la creación de cuenta de BlueXP y para comenzar a usarla. Consulte "aquí" Para instalar FSxN. Consulte "aquí" Para crear un conector en AWS para gestionar FSxN.
- Implemente FSxN con AWS. Consulte "aquí" Para la puesta en marcha mediante la consola de AWS.

Instalación de Trident en clústeres ROSA (usando el gráfico Helm)

- Use el gráfico Helm para instalar Trident en clústeres ROSA. url para el diagrama Helm: https://netapp.github.io/trident-helm-chart

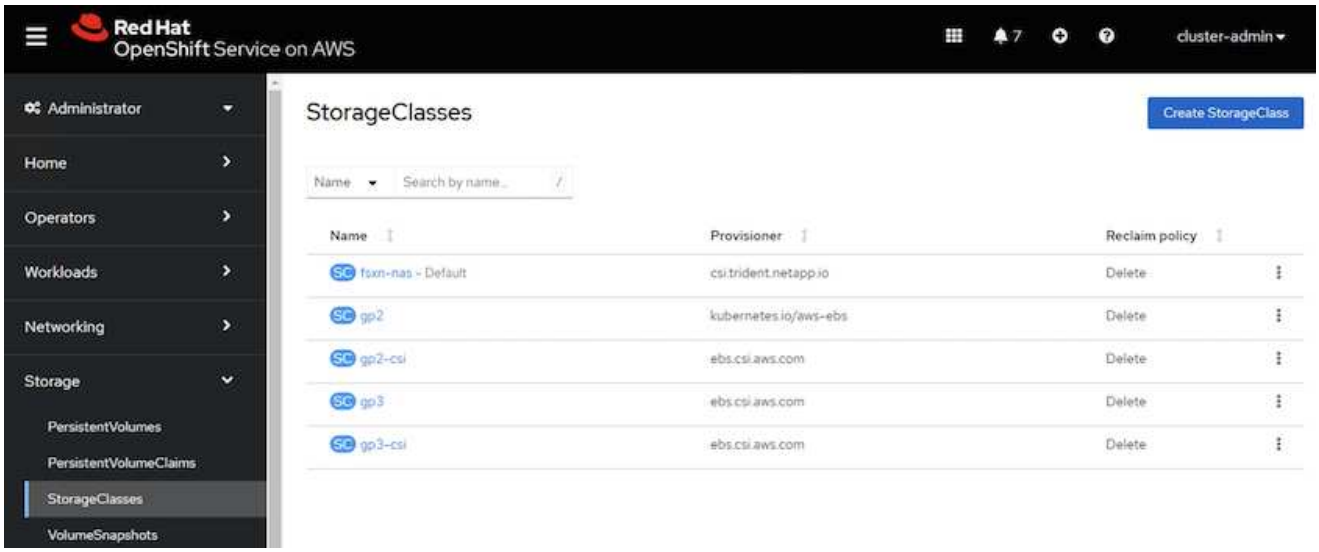
Integración de FSxN con Astra Trident para clústeres ROSA

OpenShift GitOps se puede utilizar para implementar Astra Trident CSI en todos los clústeres gestionados a medida que se registran en ArgoCD mediante ApplicationSet.

```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: trident-operator
spec:
  generators:
  - clusters: {}
    # selector:
    #   matchLabels:
    #     tridentversion: '23.04.0'
  template:
    metadata:
      name: '{{nameNormalized}}-trident'
    spec:
      destination:
        namespace: trident
        server: '{{server}}'
      source:
        repoURL: 'https://netapp.github.io/trident-helm-chart'
        targetRevision: 23.04.0
        chart: trident-operator
      project: default
      syncPolicy:
        syncOptions:
          - CreateNamespace=true
```

Crear clases de almacenamiento y back-end con Trident (para FSxN)

- Consulte "aquí" para obtener detalles sobre la creación del back-end y la clase de almacenamiento.
- Convierta la clase de almacenamiento creada para FsxN con Trident CSI por defecto en OpenShift Console. Consulte la captura de pantalla a continuación:



Desplegar una aplicación usando OpenShift GitOps (CD de Argo)

- Instale el operador OpenShift GitOps en el clúster. Consulte las instrucciones "aquí".
- Configure una nueva instancia de CD de Argo para el cluster. Consulte las instrucciones "aquí".

Abre la consola del CD de Argo e implementa una aplicación. Como ejemplo, puedes implementar una aplicación Jenkins usando Argo CD con un Helm Chart. Al crear la aplicación, se proporcionaron los siguientes detalles: Proyecto: Clúster predeterminado: https://kubernetes.default.svcEspacio de nombres: Jenkins La url del diagrama Helm: https://charts.bitnami.com/bitnami

Parámetros del timón: Global.storageClass: Fsxn-nas

Protección de datos

Esta página muestra las opciones de protección de datos para clústeres de Red Hat OpenShift gestionados en AWS (ROSA) mediante Astra Control Service. Astra Control Service (ACS) proporciona una interfaz gráfica de usuario fácil de usar con la que puedes añadir clústeres, definir aplicaciones en ellas y realizar actividades de gestión de datos para aplicaciones. También se puede acceder a las funciones de ACS mediante una API que permite la automatización de flujos de trabajo.

Astra Trident de NetApp es el motor de Astra Control (ACS o ACC). Astra Trident integra varios tipos de clústeres de Kubernetes, como Red Hat OpenShift, EKS, AKS, SUSE Rancher, Anthos, etc. con diversos tipos de almacenamiento de NetApp ONTAP, como FAS/AFF, ONTAP Select, CVO, Google Cloud Volumes Service, Azure NetApp Files y Amazon FSx para NetApp ONTAP.

En esta sección se proporcionan detalles sobre las siguientes opciones de protección de datos mediante ACS:

- Un vídeo que muestra la copia de seguridad y restauración de una aplicación ROSA que se ejecuta en una región y la restauración en otra.
- Un vídeo que muestra la instantánea y la restauración de una aplicación ROSA.
- Detalles paso a paso de la instalación de un clúster ROSA, Amazon FSx para NetApp ONTAP, mediante Astra Trident de NetApp para su integración con el back-end de almacenamiento, la instalación de una aplicación posgresql en un clúster ROSA, el uso de ACS para crear una instantánea de la aplicación y la restauración de la aplicación a partir de ella.
- Un blog en el que se muestran detalles paso a paso de la creación y restauración a partir de una instantánea para una aplicación mysql en un clúster ROSA con FSx para ONTAP mediante ACS.

Copia de seguridad/Restaurar desde copia de seguridad

El siguiente vídeo muestra la copia de seguridad de una aplicación ROSA que se ejecuta en una región y se restaura en otra región.

FSX NetApp ONTAP para el servicio Red Hat OpenShift en AWS

Snapshot/Restaurar de la instantánea

En el siguiente vídeo se muestra la toma de una instantánea de una aplicación ROSA y la restauración de la instantánea después.

Snapshot/Restore para aplicaciones en Red Hat OpenShift Service en clústeres de AWS (ROSA) con almacenamiento de Amazon FSx para NetApp ONTAP

Blog

- "Mediante Astra Control Service para la gestión de datos de aplicaciones en CLÚSTERES ROSA con el almacenamiento de Amazon FSx"

Detalles paso a paso para crear la instantánea y restaurarla a partir de ella

Configuración de requisitos previos

- "Cuenta de AWS"
- "Cuenta de Red Hat OpenShift"
- Usuario de IAM con "permisos apropiados" Para crear y acceder al clúster ROSA
- "CLI DE AWS"
- "ROSA CLI"
- "CLI de OpenShift"(oc)
- VPC con subredes y puertas de enlace y rutas correspondientes
- "ROSA Cluster instalado" En el VPC
- "Amazon FSX para ONTAP de NetApp" Creadas en el mismo VPC
- Acceso al clúster ROSA desde "Consola de nube híbrida de OpenShift"

Siguientes pasos

1. Cree un usuario administrador e inicie sesión en el clúster.
2. Cree un archivo kubeconfig para el cluster.
3. Instale Astra Trident en el clúster.
4. Cree una configuración de back-end, clase de almacenamiento y clase de snapshot con el aprovisionador CSI de Trident.
5. Despliegue una aplicación postgresql en el cluster.
6. Cree una base de datos y agregue un registro.
7. Añada el clúster a ACS.
8. Defina la aplicación en ACS.
9. Cree una instantánea mediante ACS.
10. Suprima la base de datos en la aplicación postgresql.
11. Restaurar desde una instantánea mediante ACS.
12. Verifique que su aplicación se ha restaurado de la instantánea.

1. Cree un usuario administrador e inicie sesión en el clúster

Acceda al clúster ROSA creando un usuario administrador con el siguiente comando : (solo necesita crear un usuario administrador si no creó uno en el momento de la instalación).

```
rosa create admin --cluster=<cluster-name>
```

El comando proporcionará un resultado que se parecerá a la siguiente. Inicie sesión en el clúster mediante el `oc login` el comando proporcionado en la salida.

```
W: It is recommended to add an identity provider to login to this cluster.
See 'rosa create idp --help' for more information.
I: Admin account has been added to cluster 'my-rosa-cluster'. It may take up
to a minute for the account to become active.
I: To login, run the following command:
oc login https://api.my-rosa-cluster.abcd.p1.openshiftapps.com:6443 \
--username cluster-admin \
--password FWGYL-2mkJI-00000-00000
```

> También puede iniciar sesión en el clúster mediante un token. Si ya creó un usuario administrador en el momento de la creación del clúster, puede iniciar sesión en el clúster desde la consola de Red Hat OpenShift Hybrid Cloud con las credenciales de usuario administrador. A continuación, haciendo clic en la esquina superior derecha donde se muestra el nombre del usuario que ha iniciado sesión, puede obtener el `oc login` comando (token login) para la línea de comandos.

2. Cree un archivo kubeconfig para el cluster

Siga los procedimientos "aquí" Para crear un archivo kubeconfig para el clúster ROSA. Este archivo kubeconfig se utilizará más adelante cuando agregue el clúster a ACS.

3. Instale Astra Trident en el clúster

Instale Astra Trident (versión más reciente) en el clúster ROSA. Para hacer esto, puede seguir cualquiera de los procedimientos dados "aquí". Para instalar Trident usando helm desde la consola del clúster, cree primero un proyecto denominado Trident.

A continuación, desde la vista Desarrollador, cree un repositorio de gráficos Helm. Para utilizar el campo URL `'https://netapp.github.io/trident-helm-chart'`. A continuación, cree una liberación de timón para el operador Trident.

Project: trident

Create Helm Chart Repository

Add helm chart repository.

Configure via: (•) Form view ( ) YAML view

Scope type

( ) Namespaced scoped (ProjectHelmChartRepository)

Add Helm Chart Repository in the selected namespace.

(•) Cluster scoped (HelmChartRepository)

Add Helm Chart Repository at the cluster level and in all namespaces.

Name *

trident

A unique name for the Helm Chart repository.

Display name

Astra Trident

A display name for the Helm Chart repository.

Description

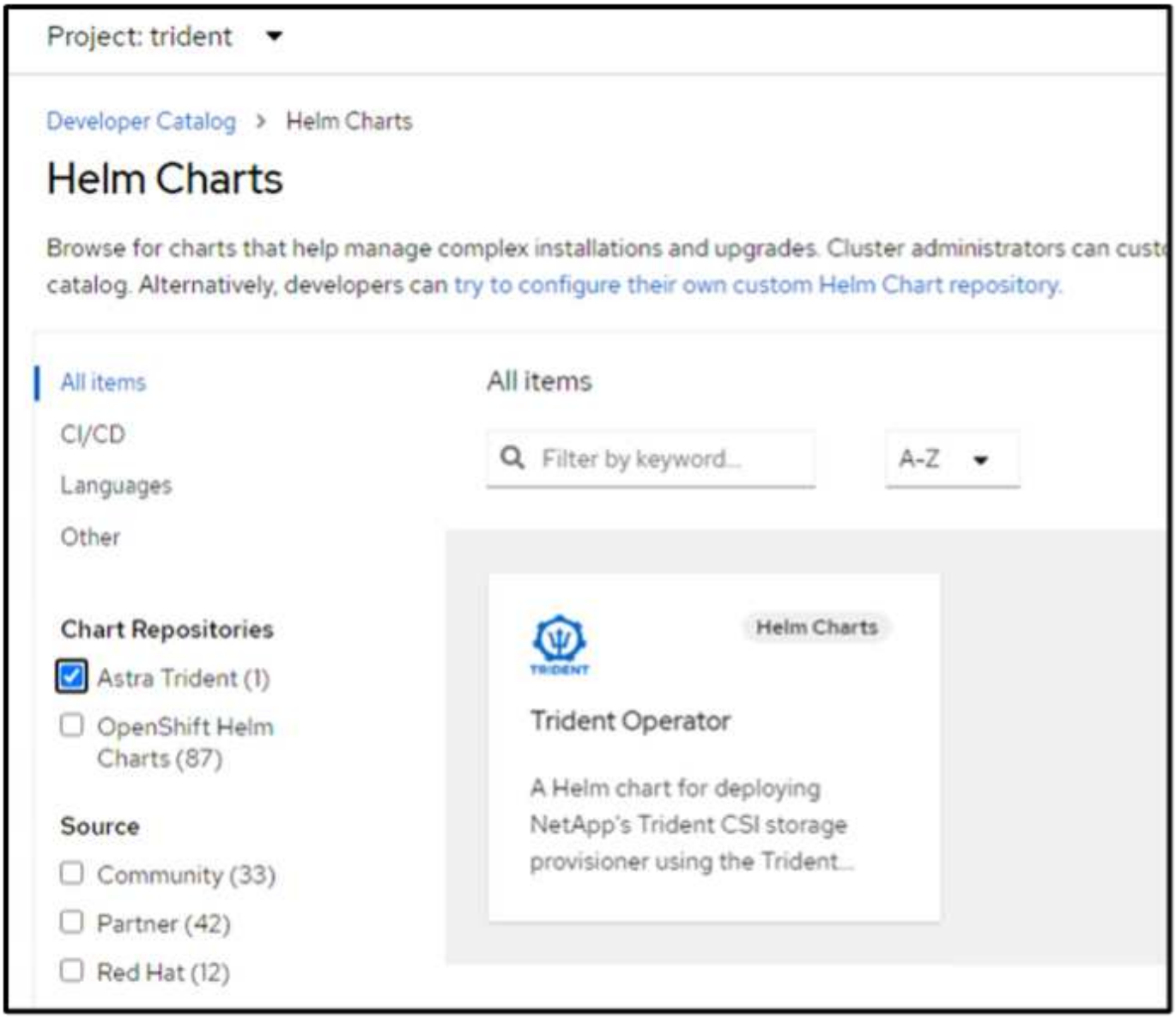
NetApp Astra Trident

A description for the Helm Chart repository.

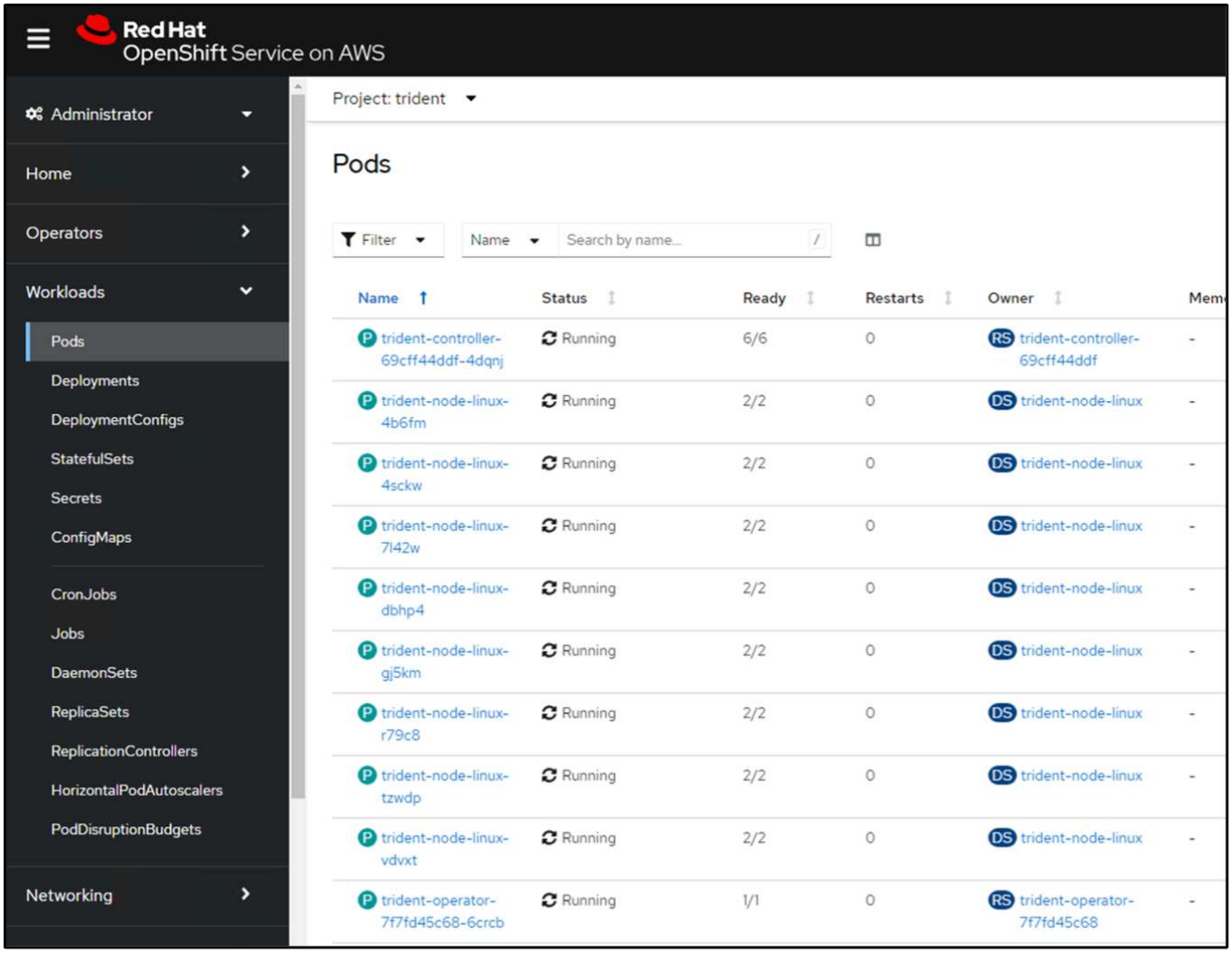
[ ] Disable usage of the repo in the developer catalog.

URL *

https://netapp.github.io/trident-helm-chart



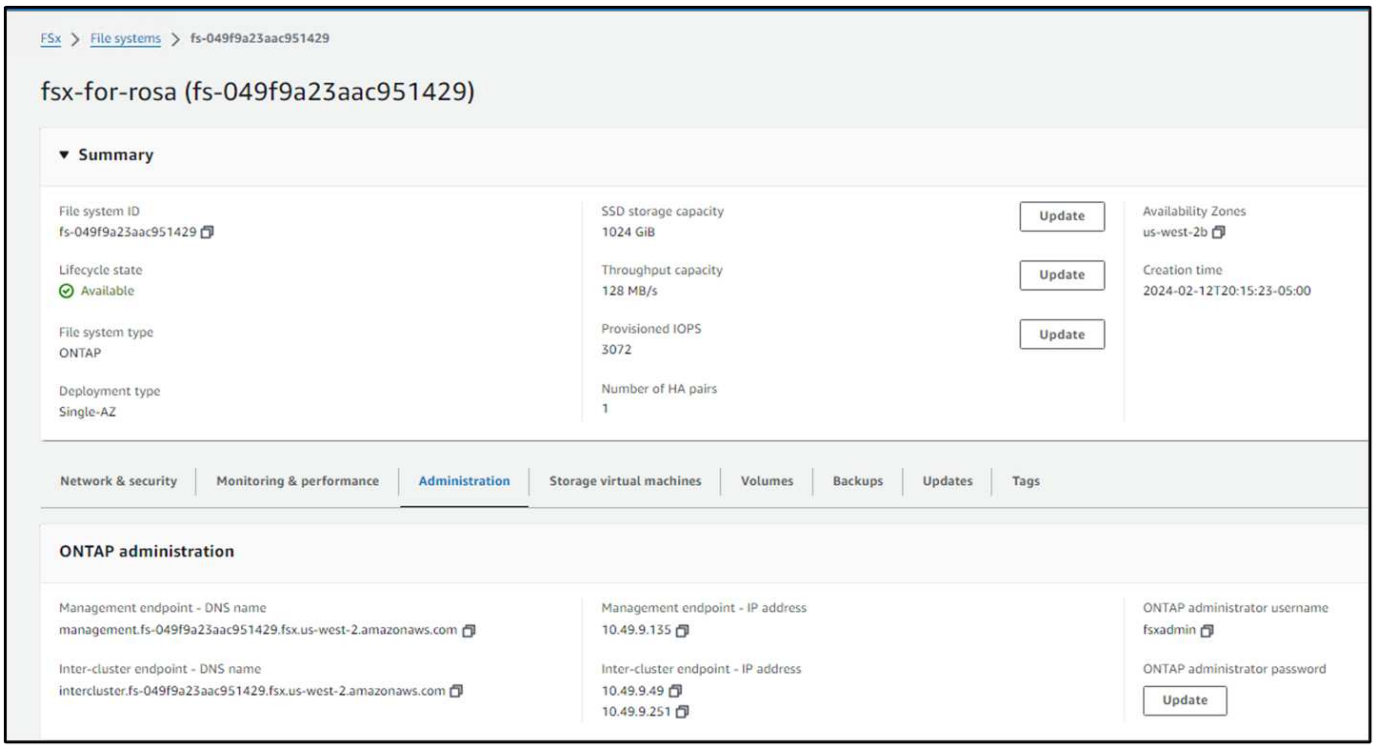
Compruebe que todos los pods de trident se están ejecutando volviendo a la vista Administrador en la consola y seleccionando pods en el proyecto de trident.



4. Cree una configuración de backend, clase de almacenamiento y clase de snapshot usando el aprovisionador CSI de Trident

Utilice los archivos yaml que se muestran a continuación para crear un objeto backend trident, un objeto de clase de almacenamiento y el objeto Volumesnapshot. Asegúrese de proporcionar las credenciales a su sistema de archivos Amazon FSx para NetApp ONTAP que creó, la LIF de gestión y el nombre Vserver de su sistema de archivos en la configuración yaml para el backend. Para obtener esos detalles, ve a la consola de AWS para Amazon FSx y selecciona el sistema de archivos, navega a la pestaña Administración. También, haga clic en Actualizar para establecer la contraseña del `fsxadmin` usuario.

> Puede utilizar la línea de comandos para crear los objetos o crearlos con los archivos yaml desde la consola de la nube híbrida.



Configuración de backend Trident

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nas-secret
type: Opaque
stringData:
  username: fsxadmin
  password: <password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <management lif>
  backendName: ontap-nas
  svm: fsx
  credentials:
    name: backend-tbc-ontap-nas-secret
```

Clase de almacenamiento

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

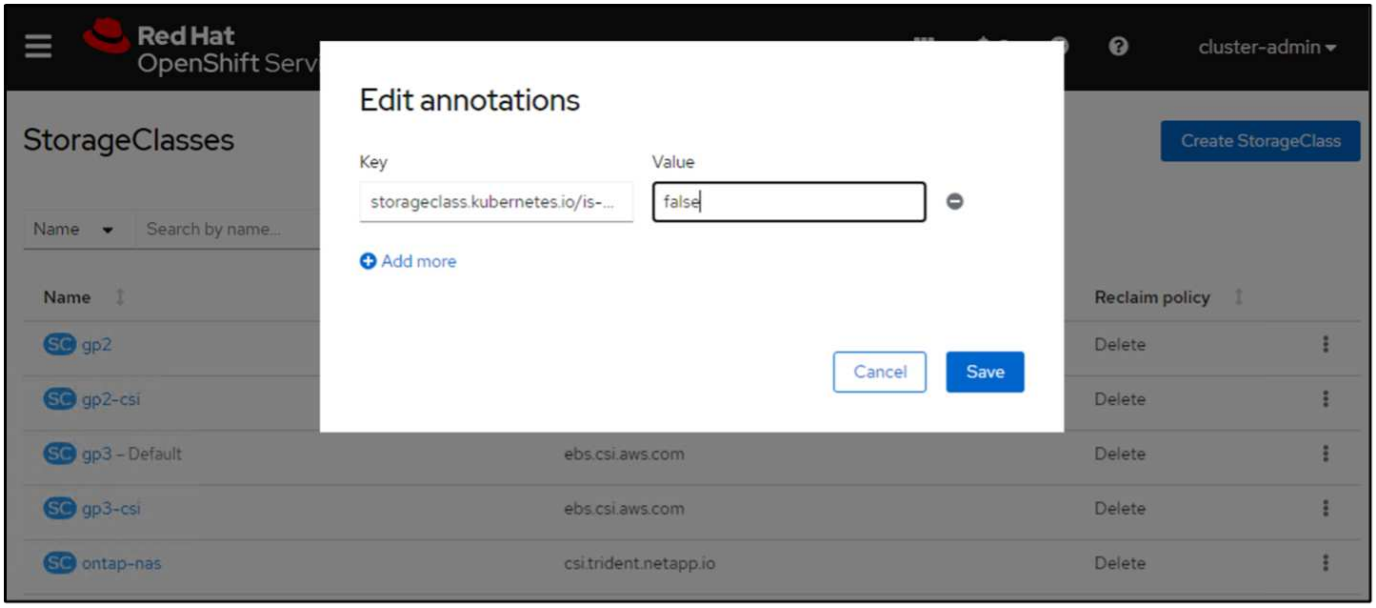
clase de instantánea

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

Verifique que el back-end, la clase storage y los objetos trident-snapshotclass se han creado utilizando los comandos que se muestran a continuación.

```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get tbc -n trident
NAME        BACKEND NAME   BACKEND UUID                           PHASE   STATUS
ontap-nas   ontap-nas      8a5e4583-2dac-46bb-b01e-fa7c3816f121   Bound   Success
[ec2-user@ip-10-49-11-132 storage]$ kubectl get sc
NAME            PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
gp2             kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   true                   3h23m
gp2-csi         ebs.csi.aws.com         Delete          WaitForFirstConsumer   true                   3h19m
gp3 (default)   ebs.csi.aws.com         Delete          WaitForFirstConsumer   true                   3h23m
gp3-csi         ebs.csi.aws.com         Delete          WaitForFirstConsumer   true                   3h19m
ontap-nas       csi.trident.netapp.io   Delete          Immediate              true                   141m
[ec2-user@ip-10-49-11-132 storage]$ kubectl get Volumesnapshotclass
NAME                    DRIVER                  DELETIONPOLICY   AGE
csi-aws-vsc             ebs.csi.aws.com         Delete           3h19m
trident-snapshotclass   csi.trident.netapp.io   Delete           6m56s
[ec2-user@ip-10-49-11-132 storage]$
```

En este momento, una importante modificación que hay que realizar es establecer ontap-nas como la clase de almacenamiento predeterminada en lugar de GP3 para que la aplicación postgresql que ponga en marcha más adelante pueda utilizar la clase de almacenamiento predeterminada. En la consola de OpenShift de su clúster, en Storage seleccione StorageClasses. Edite la anotación de la clase predeterminada actual como false y añada la anotación storageclass.kubernetes.io/is-default-class establecida como true para la clase de almacenamiento ontap-nas.



5. Implementar una aplicación postgresql en el clúster

Puede desplegar la aplicación desde la línea de comandos de la siguiente manera:

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

```
[ec2-user@ip-10-49-11-132 astra]$ helm install postgresql bitnami/postgresql -n postgresql --create-namespace
NAME: postgresql
LAST DEPLOYED: Tue Feb 13 14:46:16 2024
NAMESPACE: postgresql
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: postgresql
CHART VERSION: 14.0.4
APP VERSION: 16.2.0

** 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:16.2.0-debian-11-r1 --env="PG
PASSWORD=$POSTGRES_PASSWORD" \
      --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 avoid
the error "psql: local user with ID 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. In that
 case, old PVC will have an old password, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.
[ec2-user@ip-10-49-11-132 astra]$
```

Si no ve los pods de la aplicación en ejecución, es posible que haya un error debido a las restricciones del contexto de seguridad.

```
[ec2-user@ip-10-49-11-132 astra]$ kubectl get all -n postgresql
NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/postgresql      ClusterIP   172.30.245.50   <none>        5432/TCP   12m
service/postgresql-hl   ClusterIP   None            <none>        5432/TCP   12m

NAME                          READY   AGE
statefulset.apps/postgresql   0/1     12m
[ec2-user@ip-10-49-11-132 astra]$ kubectl get events -n postgresql
LAST SEEN   TYPE      REASON                  OBJECT                                  MESSAGE
2m39s       Normal    WaitForFirstConsumer    persistentvolumeclaim/data-postgresql-0   waiting for first consumer to be created before binding
12m         Normal    SuccessfulCreate        statefulset/postgresql                  create Claim data-postgresql-0 Pod postgresql-0 in StatefulSet postg
resql success
107s        Warning   FailedCreate            statefulset/postgresql                  create Pod postgresql-0 in StatefulSet postgresql failed error: pods
 "postgresql-0" is forbidden: unable to validate against any security context constraint: [provider "trident-controller": Forbidden: not usable by user or
serviceaccount, provider "anyuid": Forbidden: not usable by user or serviceaccount, provider restricted-v2: .spec.securityContext.fsGroup: Invalid value: [
]int64{1001}: 1001 is not an allowed group, provider restricted-v2: .containers[0].runAsUser: Invalid value: 1001: must be in the ranges: [1001010000, 1001
019999], provider "restricted": Forbidden: not usable by user or serviceaccount, provider "nonroot-v2": Forbidden: not usable by user or serviceaccount, pr
ovider "nonroot": Forbidden: not usable by user or serviceaccount, provider "pcap-dedicated-admins": Forbidden: not usable by user or serviceaccount, provi
der "hostmount-anyuid": Forbidden: not usable by user or serviceaccount, provider "machine-api-termination-handler": Forbidden: not usable by user or servi
ceaccount, provider "hostnetwork-v2": Forbidden: not usable by user or serviceaccount, provider "hostnetwork": Forbidden: not usable by user or serviceacco
unt, provider "hostaccess": Forbidden: not usable by user or serviceaccount, provider "splunkforwarder": Forbidden: not usable by user or serviceaccount, p
rovider "trident-node-linux": Forbidden: not usable by user or serviceaccount, provider "node-exporter": Forbidden: not usable by user or serviceaccount, p
rovider "privileged": Forbidden: not usable by user or serviceaccount]
[ec2-user@ip-10-49-11-132 astra]$
```

Corrija el error editando `runAsUser` y.. `fsGroup` campos de la `statefuleset.apps/postgresql` objeto con el uid que se encuentra en la salida del `oc get project` comando como se muestra a continuación.

```
[ec2-user@ip-10-49-11-132 astra]$ oc get project postgresql -o yaml | grep uid-range
    openshift.io/sa.scc.uid-range: 1001010000/10000
[ec2-user@ip-10-49-11-132 astra]$ oc edit -n postgresql statefulset.apps/postgresql
statefulset.apps/postgresql edited
[ec2-user@ip-10-49-11-132 astra]$
```

La aplicación de postgresql debería ejecutar y utilizar volúmenes persistentes respaldados por Amazon FSx para el almacenamiento de NetApp ONTAP.

```
[ec2-user@ip-10-49-11-132 astra]$ oc get pods -n postgresql
NAME            READY   STATUS    RESTARTS   AGE
postgresql-0    1/1     Running   0          2m46s
[ec2-user@ip-10-49-11-132 astra]$
```

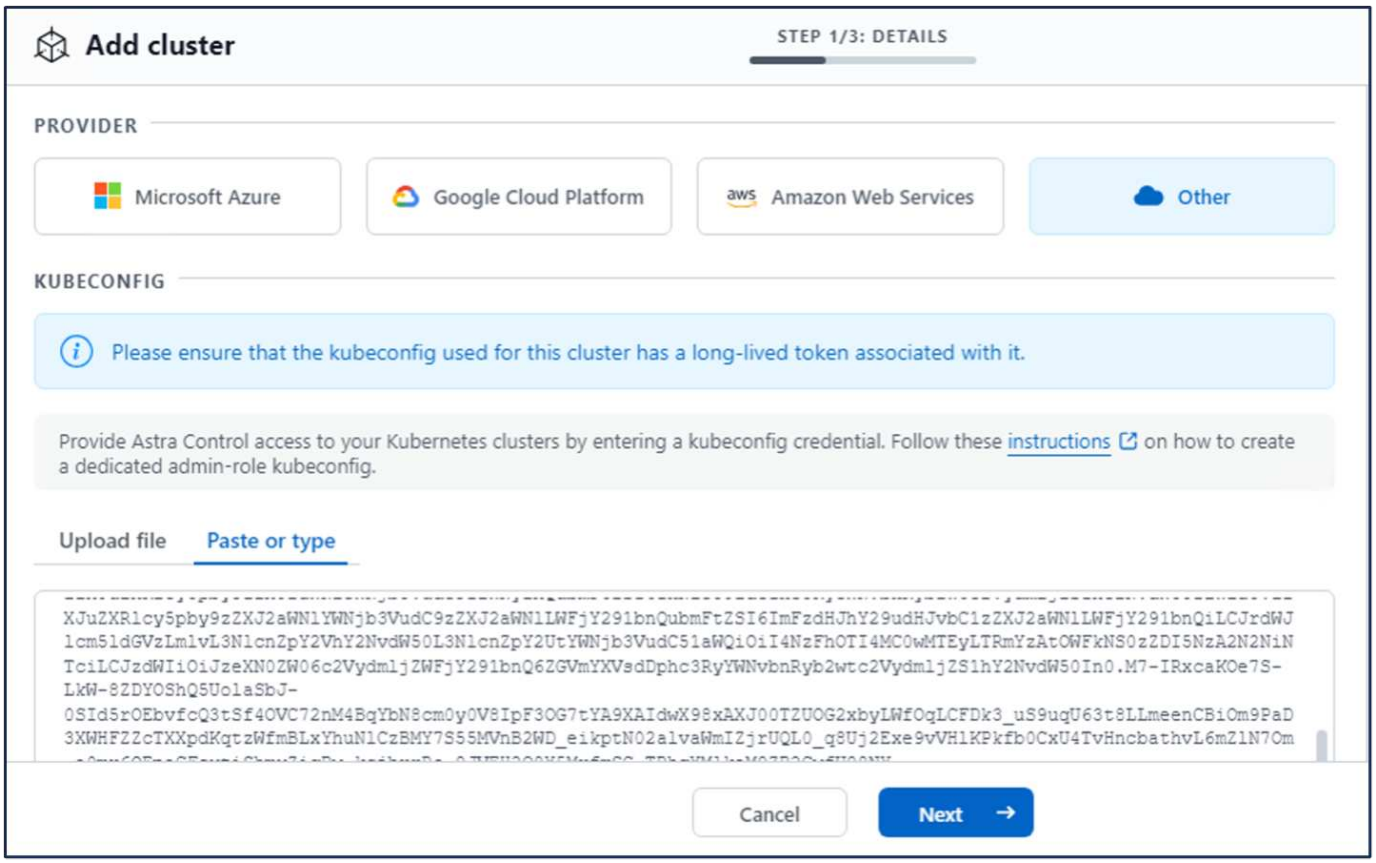
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s
[ec2-user@ip-10-49-11-132 storage]$
```

6. Crear una base de datos y agregar un registro

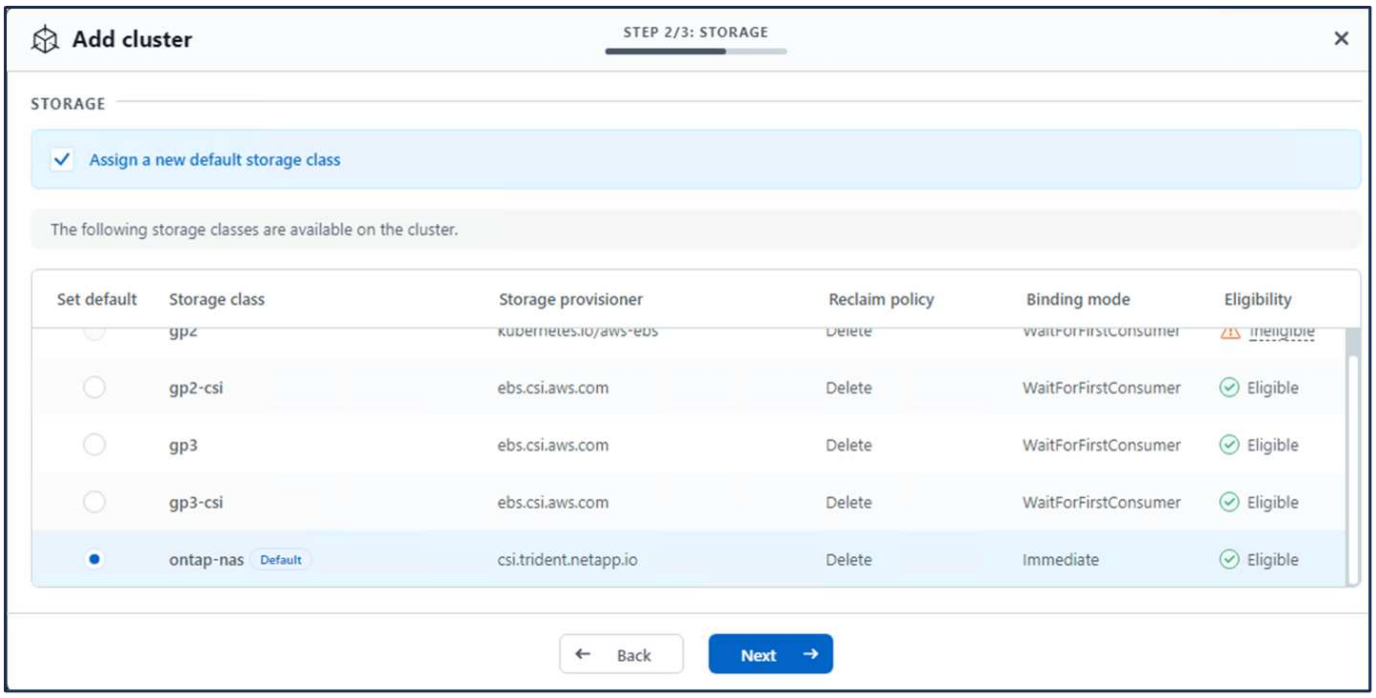
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1 --env="PGPASSWORD=$POSTGRES_PASSWORD" \
> --command -- psql --host postgresql -U postgres -d postgres -p 5432
Warning: would violate PodSecurity "restricted:v1.24": allowPrivilegeEscalation != false (container "postgresql-client" must se
t securityContext.allowPrivilegeEscalation=false), unrestricted capabilities (container "postgresql-client" must set securityCo
ntext.capabilities.drop=["ALL"]), runAsNonRoot != true (pod or container "postgresql-client" must set securityContext.runAsNonR
oot=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=# CREATE DATABASE erp;
CREATE DATABASE
postgres=# \c erp
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)
```

7. Agregue el clúster a ACS

Inicie sesión en ACS. Seleccione cluster y haga clic en Add. Seleccione Otro y cargue o pegue el archivo kubeconfig.



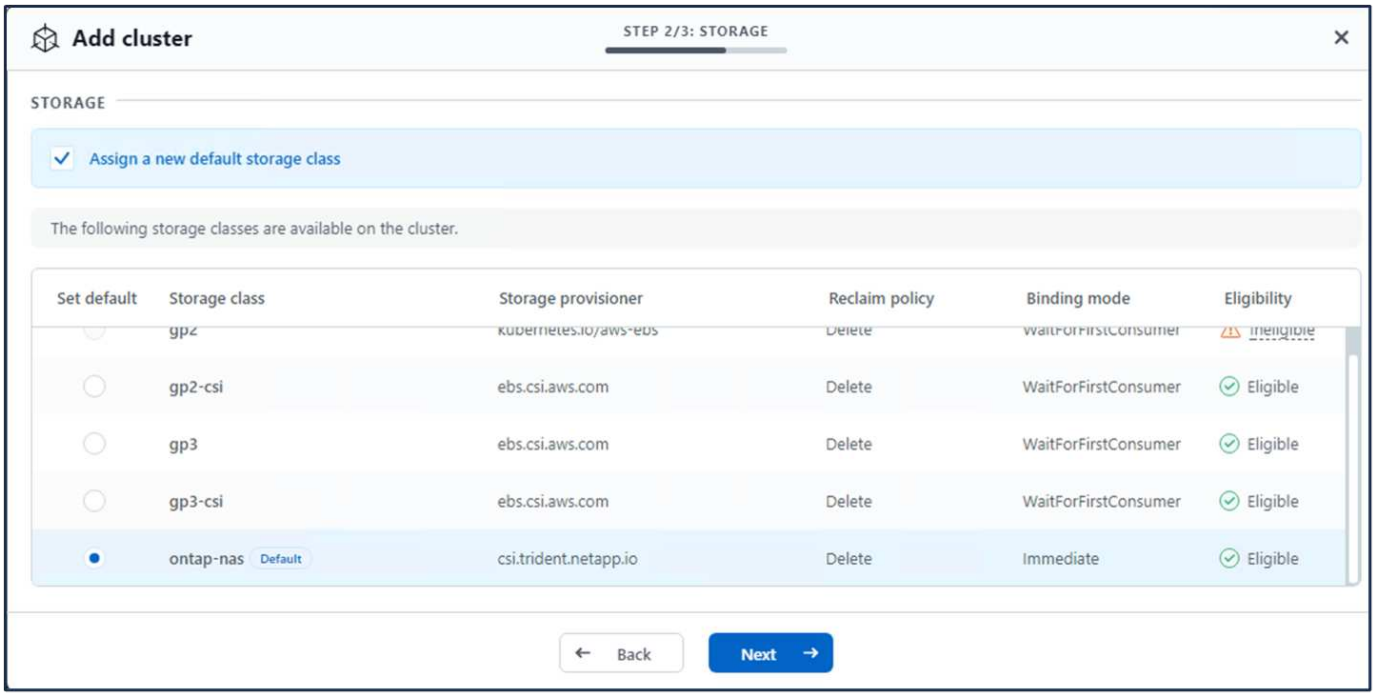
Haga clic en **Next** y seleccione ontap-nas como la clase de almacenamiento predeterminada para ACS. Haga clic en **Siguiente**, revise los detalles y **Agregar** el clúster.



8. Defina la aplicación en ACS

Defina la aplicación postgresql en ACS. En la página de destino, selecciona **Aplicaciones**, **Definir** y rellena los detalles apropiados. Haga clic en **Siguiente** un par de veces, revise los detalles y haga clic en **Definir**. La

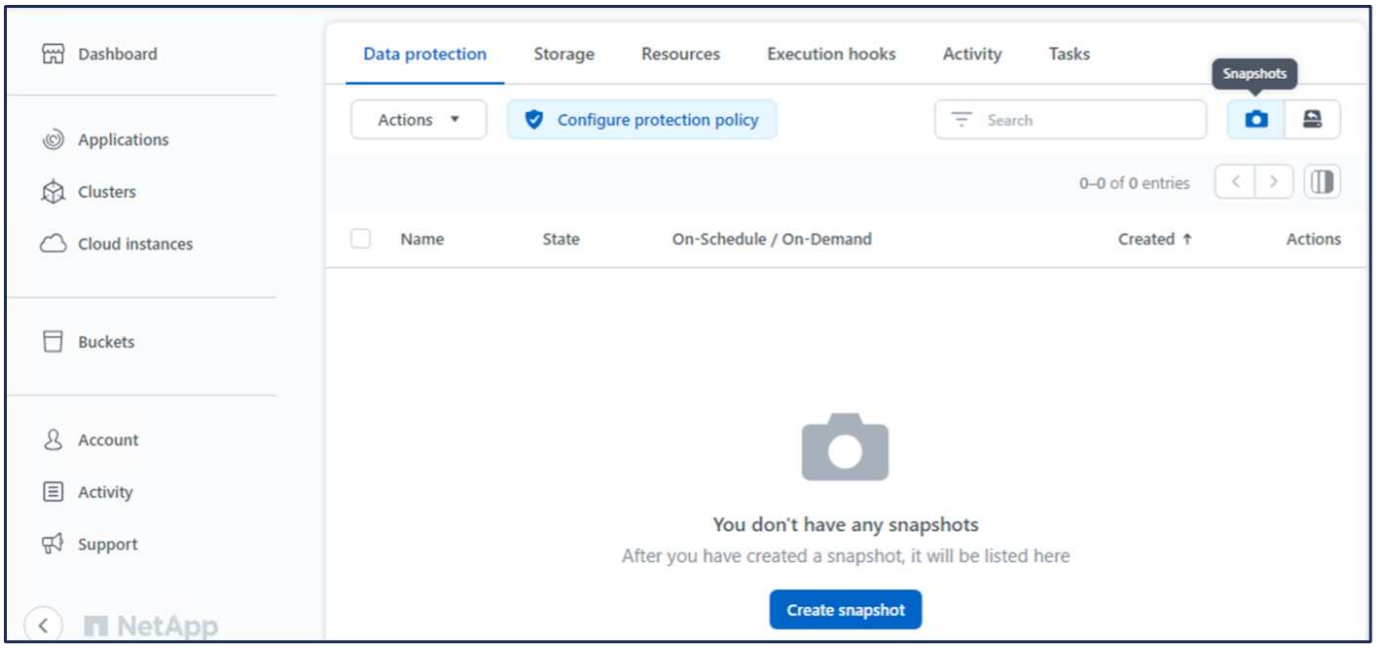
aplicación se agrega a ACS.

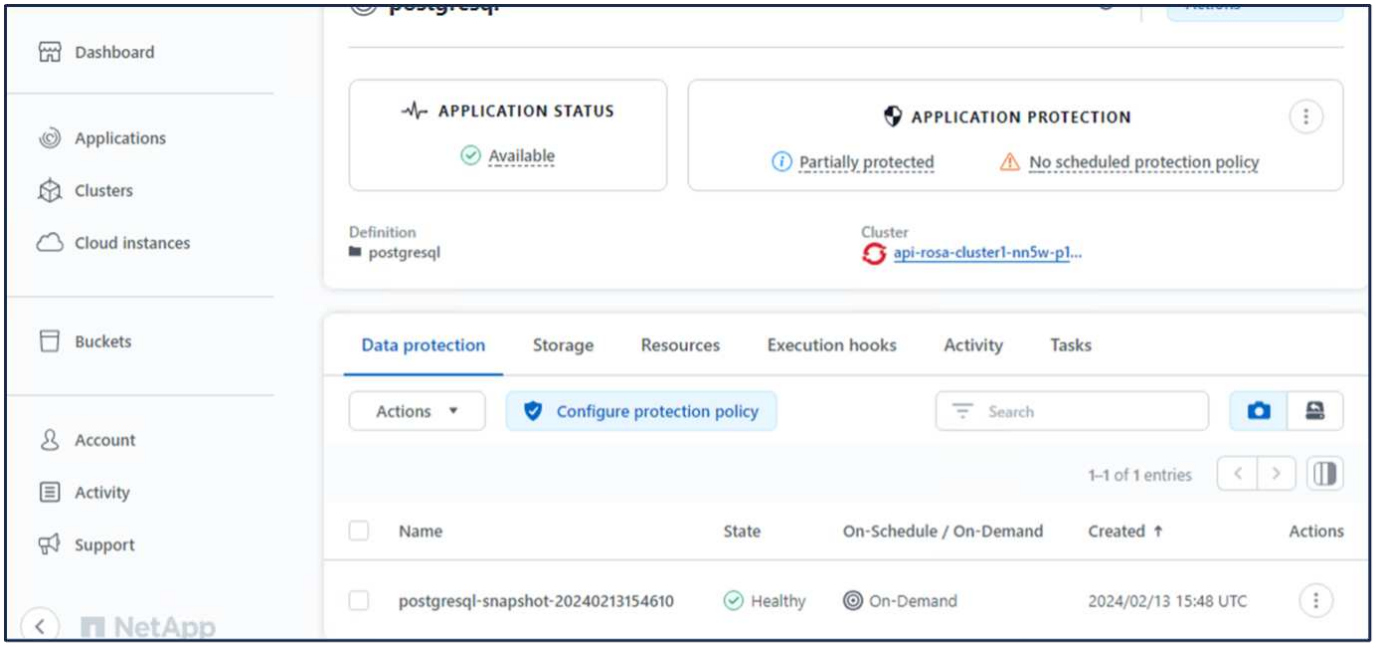


9. Cree una instantánea con ACS

Hay muchas maneras de crear una instantánea en ACS. Puede seleccionar la aplicación y crear una instantánea desde la página que muestra los detalles de la aplicación. Puede hacer clic en Crear snapshot para crear una snapshot bajo demanda o configurar una política de protección.

Cree una instantánea bajo demanda simplemente haciendo clic en **Crear instantánea**, proporcionando un nombre, revisando los detalles y haciendo clic en **Instantánea**. El estado de la Snapshot cambia a correcto una vez que se completa la operación.





10. Elimine la base de datos en la aplicación postgresql

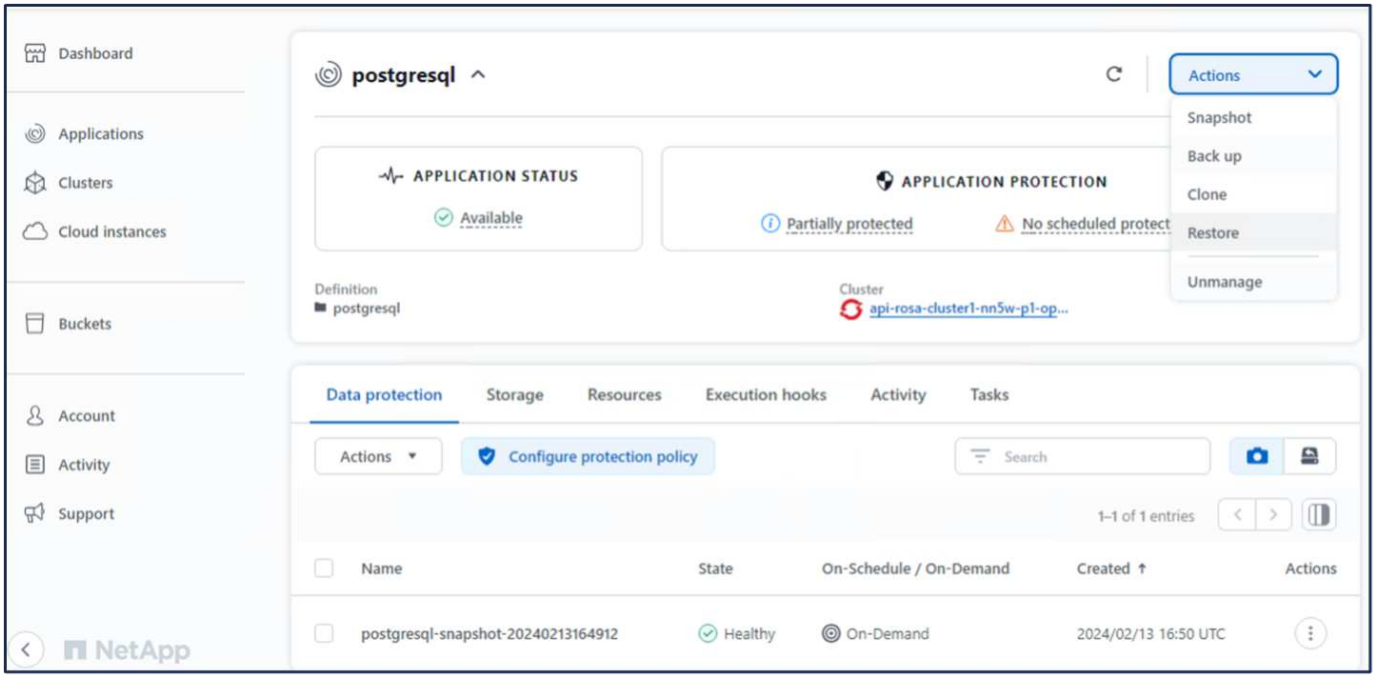
Vuelva a conectarse a postgresql, enumere las bases de datos disponibles, suprima la que creó anteriormente y vuelva a listar para asegurarse de que la base de datos se ha eliminado.

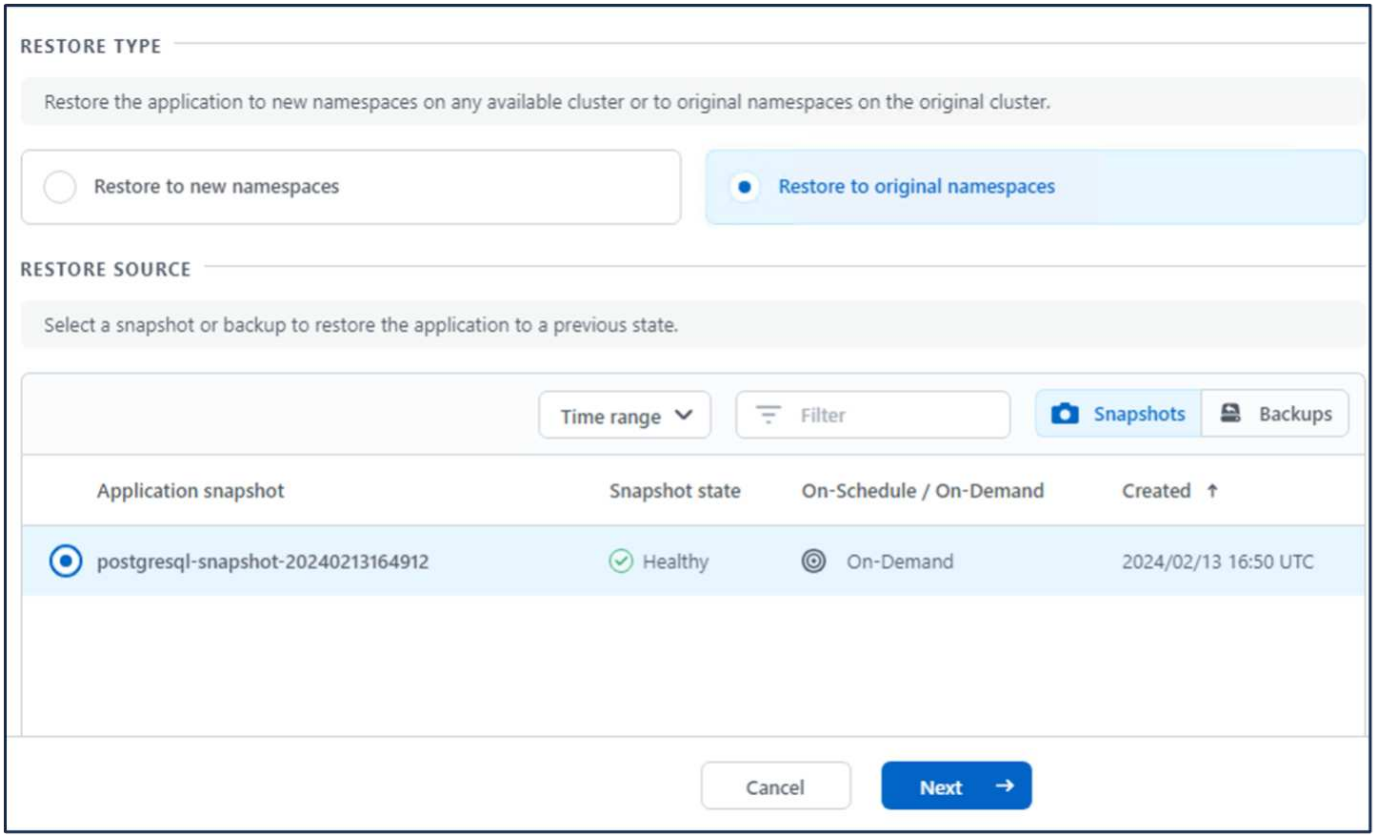
```
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------
 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/
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres
+
           |          |          |                 |             |             |            |           | postgres=CTc/
(4 rows)

postgres=# DROP DATABASE erp;
DROP DATABASE
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------
 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/
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres
+
           |          |          |                 |             |             |            |           | postgres=CTc/
(3 rows)
```

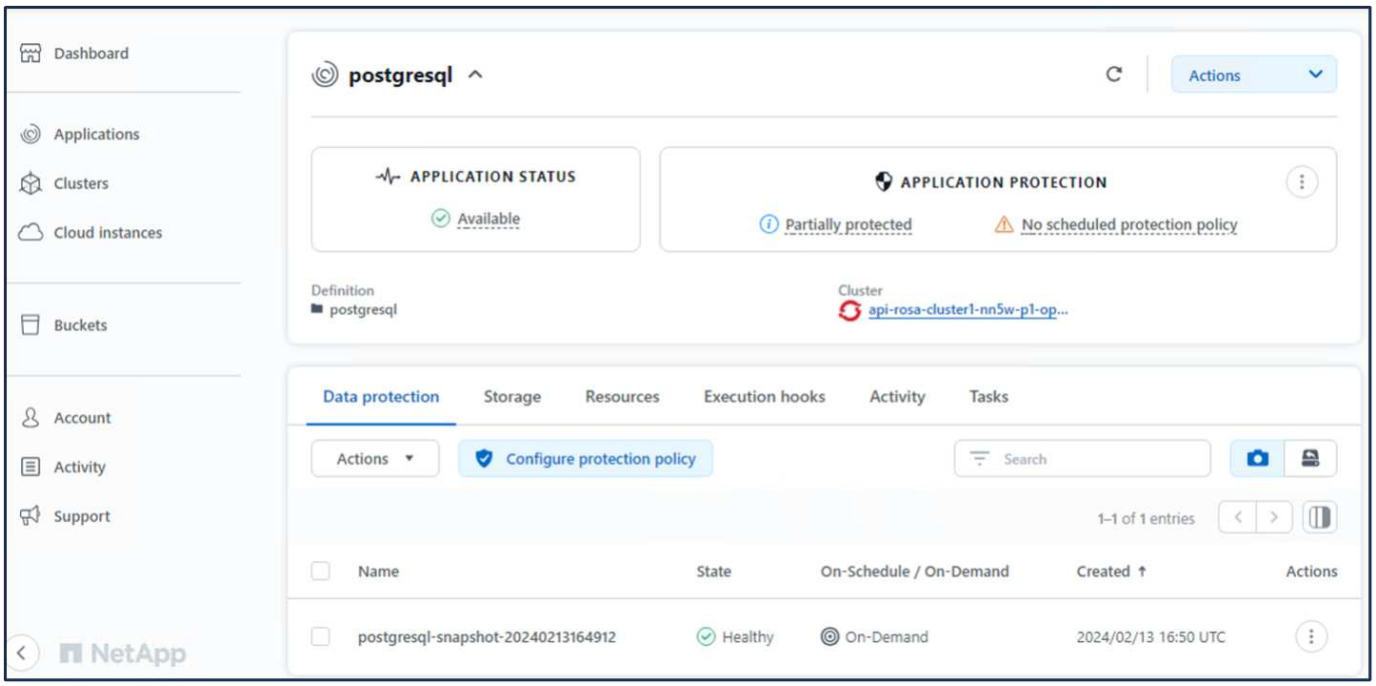
11. Restaurar desde una instantánea mediante ACS

Para restaurar la aplicación desde una instantánea, vaya a la página de inicio de la interfaz de usuario de ACS, seleccione la aplicación y seleccione Restaurar. Debe elegir la copia Snapshot o un backup desde el

que desea restaurar. (Por lo general, tendría varios creados en función de una política que haya configurado). Tome las decisiones adecuadas en el próximo par de pantallas y luego haga clic en **Restaurar**. El estado de la aplicación pasa de restaurar a Disponible después de que se ha restaurado de la copia de Snapshot.







12. Verifique que su aplicación se ha restaurado a partir de la instantánea

Inicie sesión en el cliente postgresql y ahora debería ver la tabla y el registro en la tabla que tenía anteriormente. Eso es todo. Con solo hacer clic en un botón, su aplicación se ha restaurado a un estado anterior. Es así de fácil que conseguimos a nuestros clientes con Astra Control.

```
[ec2-user@ip-10-49-11-132 ~]$ 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:v1.24": allowPrivilegeEscalation != false (container "postgresql-client" must set securityContext.allowPrivi
legeEscalation=false), unrestricted capabilities (container "postgresql-client" must set securityContext.capabilities.drop=["ALL"]), runAsNonRoot != true (
pod or container "postgresql-client" must set securityContext.runAsNonRoot=true), seccompProfile (pod or container "postgresql-client" must set securityCon
text.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
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 | firstname | lastname
----+-----------+----------
  1 | John      | Doe
(1 row)
```

Migración de datos

Esta página muestra las opciones de migración de datos para las cargas de trabajo de contenedor en clústeres de Red Hat OpenShift gestionados mediante FSx para NetApp ONTAP para el almacenamiento persistente.

Migración de datos

Red Hat OpenShift Service en AWS, así como FSx para ONTAP de NetApp (FSxN) forman parte de su cartera de servicios de AWS. FSxN está disponible en las opciones de AZ única o Multi-AZ. La opción Multi-AZ proporciona protección de datos frente a un fallo en la zona de disponibilidad. FSxN puede integrarse con Astra Trident para proporcionar almacenamiento persistente para aplicaciones en clústeres de ROSA.

Integración de FSxN con Trident mediante el gráfico Helm

Integración de clústeres ROSA con Amazon FSx para ONTAP

La migración de las aplicaciones de contenedores implica:

- Volúmenes persistentes: Se puede realizar con BlueXP. Otra opción consiste en utilizar Astra Control Center para gestionar las migraciones de aplicaciones de contenedores desde las instalaciones al entorno de cloud. La automatización se puede usar para el mismo propósito.
- Metadatos de la aplicación: Esto se puede realizar con OpenShift GitOps (CD de Argo).

Recuperación tras fallos y conmutación por error de aplicaciones en el cluster ROSA utilizando FSxN para el almacenamiento persistente

El siguiente vídeo es una demostración de los escenarios de conmutación al nodo de respaldo y conmutación de retorno tras recuperación en las aplicaciones con BlueXP y Argo CD.

Failover y failover de aplicaciones en el cluster ROSA

Solución de protección y migración de datos para las cargas de trabajo de contenedores de OpenShift