Hybrid Cloud mit vom Anbieter verwalteten Komponenten

NetApp public and hybrid cloud solutions

NetApp
August 18, 2025

Inhalt

Hybrid Cloud mit vom Anbieter verwalteten Komponenten

NetApp -Lösung mit verwalteten Red Hat OpenShift Container-Plattform-Workloads

Kunden sind möglicherweise „in der Cloud geboren“ oder befinden sich an einem Punkt ihrer Modernisierung, an dem sie bereit sind, einige ausgewählte oder alle Workloads aus ihren Rechenzentren in die Cloud zu verschieben. Sie können sich für die Ausführung ihrer Workloads für die Verwendung von vom Anbieter verwalteten OpenShift-Containern und vom Anbieter verwaltetem NetApp -Speicher in der Cloud entscheiden. Sie sollten die verwalteten Red Hat OpenShift-Containercluster in der Cloud planen und bereitstellen, um eine erfolgreiche produktionsbereite Umgebung für ihre Container-Workloads zu schaffen. NetApp bietet vollständig verwaltete Speicherangebote für Managed Red Hat-Lösungen in allen drei führenden öffentlichen Clouds.

- Amazon FSx for NetApp ONTAP (FSx ONTAP)*

FSx ONTAP bietet Datenschutz, Zuverlässigkeit und Flexibilität für Containerbereitstellungen in AWS. Trident dient als dynamischer Speicherbereitsteller, um den persistenten FSx ONTAP -Speicher für die zustandsbehafteten Anwendungen der Kunden zu nutzen.

Da ROSA im HA-Modus mit über mehrere Verfügbarkeitszonen verteilten Control Plane-Knoten bereitgestellt werden kann, kann FSx ONTAP auch mit der Multi-AZ-Option bereitgestellt werden, die hohe Verfügbarkeit bietet und vor AZ-Ausfällen schützt.

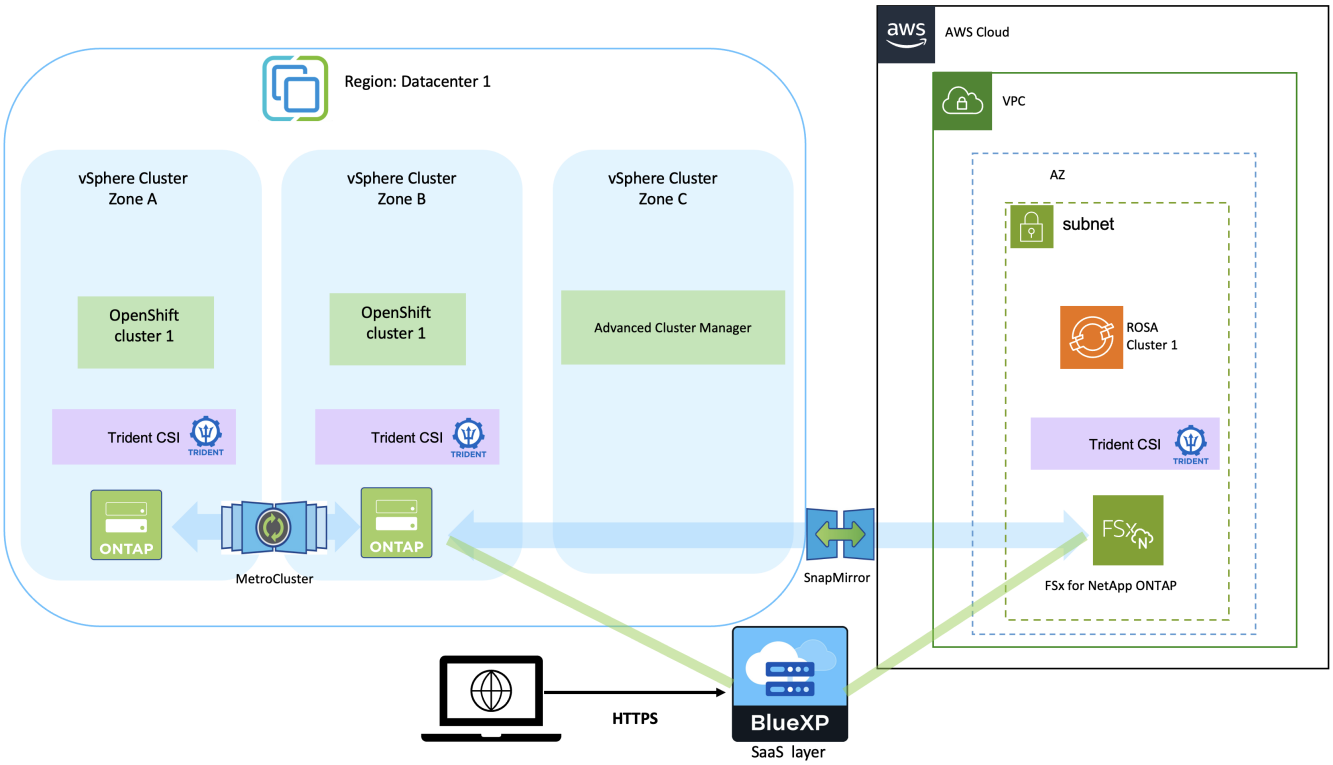
- Google Cloud NetApp Volumes*

Red Hat OpenShift Dedicated ist eine vollständig verwaltete Anwendungsplattform, mit der Sie Anwendungen in der gesamten Hybrid Cloud schnell erstellen, bereitstellen und skalieren können. Google Cloud NetApp Volumes bietet persistente Volumes und bringt die gesamte Palette der Enterprise-Datenverwaltungsfunktionen von ONTAP in OpenShift-Bereitstellungen in Google Cloud.

Bereitstellen und Konfigurieren der Managed Red Hat OpenShift Container-Plattform auf AWS

In diesem Abschnitt wird ein allgemeiner Workflow zum Einrichten der verwalteten Red Hat OpenShift-Cluster auf AWS (ROSA) beschrieben. Es zeigt die Verwendung von verwaltetem Amazon FSx for NetApp ONTAP (FSx ONTAP) als Speicher-Backend von Trident zur Bereitstellung persistenter Volumes. Es werden Details zur Bereitstellung von FSx ONTAP auf AWS mit BlueXP bereitgestellt. Außerdem werden Details zur Verwendung von BlueXP und OpenShift GitOps (Argo CD) bereitgestellt, um Datenschutz- und Migrationsaktivitäten für die zustandsbehafteten Anwendungen auf ROSA-Clustern durchzuführen.

Hier ist ein Diagramm, das die auf AWS bereitgestellten ROSA-Cluster darstellt, die FSx ONTAP als Backend-Speicher verwenden.



Diese Lösung wurde durch die Verwendung von zwei ROSA-Clustern in zwei VPCs in AWS verifiziert. Jeder ROSA-Cluster wurde mithilfe von Trident in FSx ONTAP integriert. Es gibt mehrere Möglichkeiten, ROSA-Cluster und FSx ONTAP in AWS bereitzustellen. Diese allgemeine Beschreibung des Setups enthält Dokumentationslinks für die jeweils verwendete Methode. Weitere Methoden finden Sie in den entsprechenden Links im"Ressourcenbereich" .

Der Einrichtungsprozess kann in die folgenden Schritte unterteilt werden:

Installieren Sie ROSA-Cluster

- Erstellen Sie zwei VPCs und richten Sie eine VPC-Peering-Konnektivität zwischen den VPCs ein.
- Verweisen"hier," Anweisungen zum Installieren von ROSA-Clustern.

Installieren Sie FSx ONTAP

- Installieren Sie FSx ONTAP auf den VPCs von BlueXP. Verweisen"hier," zur Erstellung eines BlueXP Kontos und zum Einstieg. Verweisen"hier," zur Installation von FSx ONTAP. Verweisen"hier," zum Erstellen eines Connectors in AWS zur Verwaltung des FSx ONTAP.
- Stellen Sie FSx ONTAP mit AWS bereit. Verweisen"hier," zur Bereitstellung mithilfe der AWS-Konsole.

Installieren Sie Trident auf ROSA-Clustern (mithilfe des Helm-Diagramms).

- Verwenden Sie das Helm-Diagramm, um Trident auf ROSA-Clustern zu installieren. Siehe den Dokumentationslink: hier.

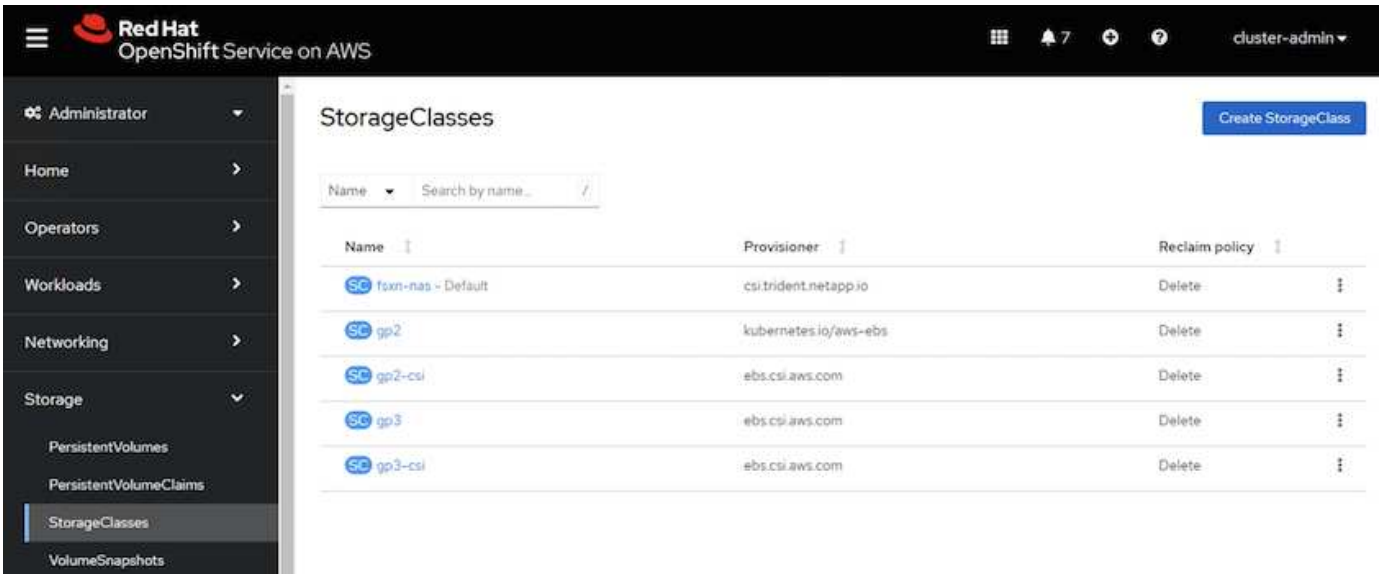
Integration von FSx ONTAP mit Trident für ROSA-Cluster

OpenShift GitOps kann verwendet werden, um Trident CSI auf allen verwalteten Clustern bereitzustellen, da diese mithilfe von ApplicationSet bei ArgoCD registriert werden.

```yaml
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
```

Erstellen Sie Backend- und Speicherklassen mit Trident (für FSx ONTAP)

- Verweisen"hier," für Details zum Erstellen von Backend- und Speicherklassen.
- Legen Sie die für FsxN mit Trident CSI erstellte Speicherklasse als Standard aus der OpenShift-Konsole fest. Siehe Screenshot unten:



Bereitstellen einer Anwendung mit OpenShift GitOps (Argo CD)

- Installieren Sie den OpenShift GitOps-Operator auf dem Cluster. Siehe Anweisungen"hier," .
- Richten Sie eine neue Argo CD-Instanz für den Cluster ein. Siehe Anweisungen"hier," .

Öffnen Sie die Konsole von Argo CD und stellen Sie eine App bereit. Beispielsweise können Sie eine Jenkins-App mithilfe von Argo CD mit einem Helm-Diagramm bereitstellen. Beim Erstellen der Anwendung wurden

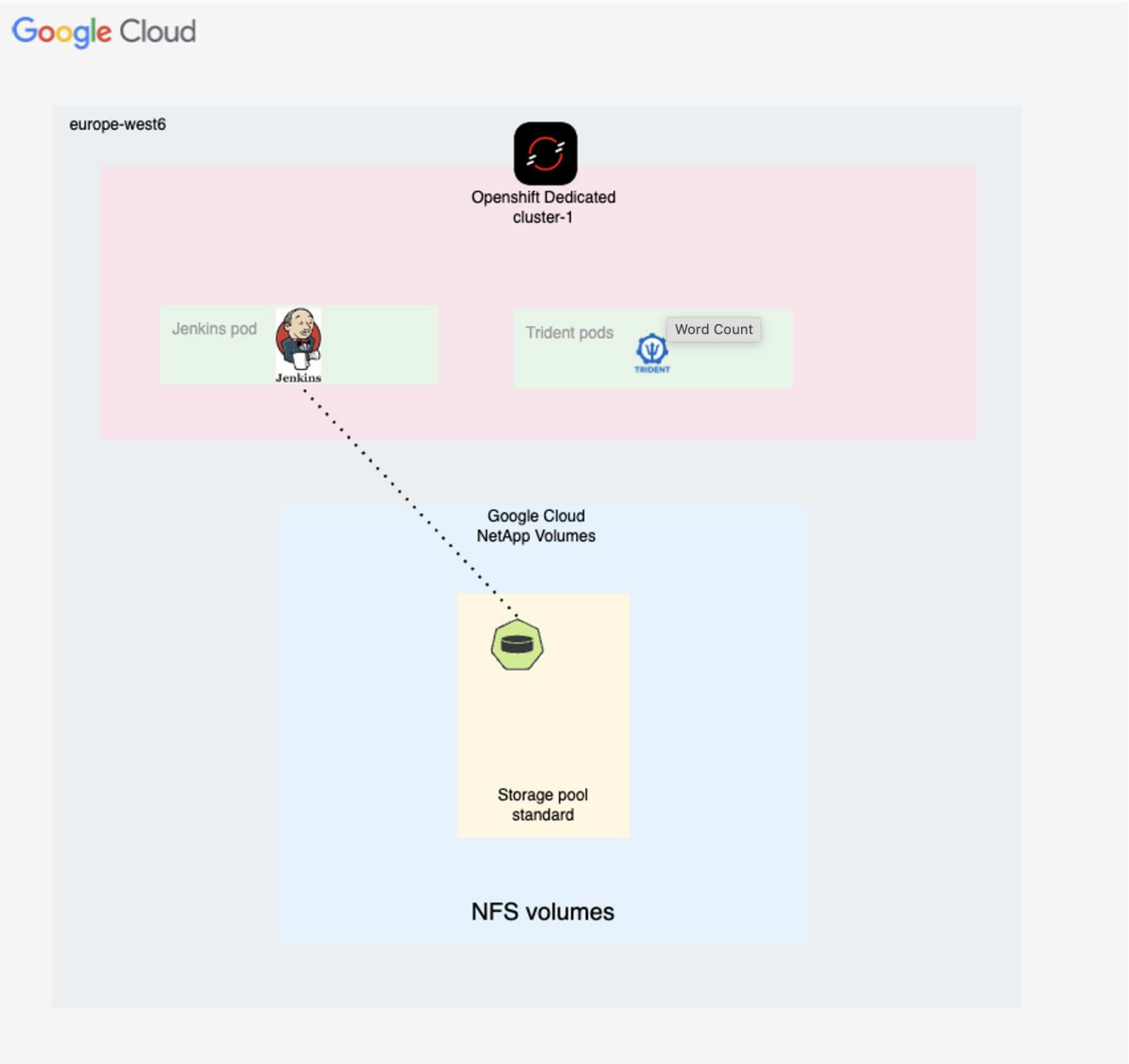
folgende Angaben gemacht: Projekt: Standardcluster:'https://kubernetes.default.svc' (ohne Anführungszeichen) Namespace: Jenkins Die URL für das Helm-Diagramm:'https://charts.bitnami.com/bitnami' (ohne Anführungszeichen)

Helm-Parameter: global.storageClass: fsxn-nas

Bereitstellen und Konfigurieren von OpenShift Dedicated auf Google Cloud mit Google Cloud NetApp Volumes

In diesem Abschnitt wird ein allgemeiner Workflow zum Einrichten von OpenShift Dedicated (OSD)-Clustern auf der Google Cloud-Plattform beschrieben. Es zeigt, wie NetApp Trident Google Cloud NetApp Volumes als Speicher-Backend verwendet, um persistente Volumes für zustandsbehaftete Anwendungen bereitzustellen, die mit Kubernetes ausgeführt werden.

Hier ist ein Diagramm, das einen OSD-Cluster darstellt, der in Google Cloud bereitgestellt wird und NetApp Volumes als Backend-Speicher verwendet.



Der Einrichtungsprozess kann in die folgenden Schritte unterteilt werden:

Installieren Sie OSD-Cluster in Google Cloud

- Wenn Sie eine vorhandene VPC für den Cluster verwenden möchten, müssen Sie die VPC, zwei Subnetze, einen Cloud-Router und zwei GCP-Cloud-NATs für den OSD-Cluster erstellen. Verweisen"hier," Anweisungen hierzu finden Sie unter.
- Verweisen"hier," Anweisungen zum Installieren von OSD-Clustern auf GCP mithilfe des Abrechnungsmodells Customer Cloud Subscription (CCS). OSD ist auch im Google Cloud Marketplace enthalten. Ein Video, das zeigt, wie Sie OSD mithilfe der Google Cloud Marketplace-Lösung installieren, finden Sie"hier," .

Google Cloud NetApp Volumes aktivieren

- Verweisen"hier," Informationen zum Einrichten des Zugriffs auf Google Cloud NetApp Volumes. Befolgen Sie alle Schritte bis einschließlich
- Erstellen Sie einen Speicherpool. Verweisen"hier," Informationen zum Einrichten eines Speicherpools auf

Google Cloud NetApp Volumes. Innerhalb des Speicherpools werden Volumes für die zustandsbehafteten Kubernetes-Anwendungen erstellt, die auf OSD ausgeführt werden.

Installieren Sie Trident auf OSD-Clustern (mithilfe des Helm-Diagramms).

- Verwenden Sie ein Helm-Diagramm, um Trident auf OSD-Clustern zu installieren. Verweisen"hier," Anweisungen zum Installieren des Helm-Diagramms finden Sie unter. Das Steuerdiagramm finden Sie"hier," .

Integration von NetApp Volumes mit NetApp Trident für OSD-Cluster

Erstellen Sie Backend- und Speicherklassen mit Trident (für Google Cloud NetApp Volumes)

- Weitere Informationen zum Erstellen des Backends finden Sie hier.
- Wenn eine der aktuellen Speicherklassen in Kubernetes als Standard markiert ist, entfernen Sie diese Anmerkung, indem Sie die Speicherklasse bearbeiten.
- Erstellen Sie mit dem Trident CSI Provisioner mindestens eine Speicherklasse für NetApp -Volumes. Machen Sie mithilfe einer Anmerkung genau eine der Speicherklassen zum Standard. Dadurch kann ein PVC diese Speicherklasse verwenden, wenn sie im PVC-Manifest nicht explizit aufgerufen wird. Ein Beispiel mit der Anmerkung wird unten angezeigt.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-standard-k8s
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: true
```

Bereitstellen einer Anwendung mit OpenShift GitOps (Argo CD)

- Installieren Sie den OpenShift GitOps-Operator auf dem Cluster. Siehe Anweisungen"hier," .
- Richten Sie eine neue Argo CD-Instanz für den Cluster ein. Siehe Anweisungen"hier," .

Öffnen Sie die Konsole von Argo CD und stellen Sie eine App bereit. Beispielsweise können Sie eine Jenkins-App mithilfe von Argo CD mit einem Helm-Diagramm bereitstellen. Beim Erstellen der Anwendung wurden folgende Angaben gemacht: Projekt: Standardcluster:'https://kubernetes.default.svc' (ohne Anführungszeichen) Namespace: Jenkins Die URL für das Helm-Diagramm:'https://charts.bitnami.com/bitnami' (ohne Anführungszeichen)

Datenschutz

Auf dieser Seite werden die Datenschutzoptionen für Managed Red Hat OpenShift on AWS (ROSA)-Cluster mit Astra Control Service angezeigt. Astra Control Service (ACS) bietet eine benutzerfreundliche grafische Benutzeroberfläche, mit der Sie Cluster hinzufügen, darauf ausgeführte Anwendungen definieren und anwendungsbezogene

Datenverwaltungsaktivitäten durchführen können. Auf ACS-Funktionen kann auch über eine API zugegriffen werden, die die Automatisierung von Arbeitsabläufen ermöglicht.

Astra Control (ACS oder ACC) wird von NetApp Trident angetrieben. Trident integriert mehrere Arten von Kubernetes-Clustern wie Red Hat OpenShift, EKS, AKS, SUSE Rancher, Anthos usw. mit verschiedenen Varianten von NetApp ONTAP -Speicher wie FAS/ AFF, ONTAP Select, CVO, Google Google Cloud NetApp Volumes, Azure NetApp Files und Amazon FSx ONTAP.

Dieser Abschnitt enthält Details zu den folgenden Datenschutzoptionen mit ACS:

- Ein Video, das die Sicherung und Wiederherstellung einer ROSA-Anwendung zeigt, die in einer Region ausgeführt und in einer anderen Region wiederhergestellt wird.
- Ein Video, das Snapshot und Wiederherstellung einer ROSA-Anwendung zeigt.
- Schrittweise Anleitung zur Installation eines ROSA-Clusters, Amazon FSx ONTAP, Verwendung von NetApp Trident zur Integration mit dem Speicher-Backend, Installation einer PostgreSQL-Anwendung auf dem ROSA-Cluster, Verwendung von ACS zum Erstellen eines Snapshots der Anwendung und Wiederherstellen der Anwendung daraus.
- Ein Blog, der Schritt für Schritt Details zum Erstellen und Wiederherstellen eines Snapshots für eine MySQL-Anwendung auf einem ROSA-Cluster mit FSx ONTAP unter Verwendung von ACS zeigt.

Sichern/Aus einer Sicherung wiederherstellen

Das folgende Video zeigt die Sicherung einer ROSA-Anwendung, die in einer Region ausgeführt und in einer anderen Region wiederhergestellt wird.

FSx NetApp ONTAP für Red Hat OpenShift Service auf AWS

Snapshot/Wiederherstellung aus Snapshot

Das folgende Video zeigt, wie ein Snapshot einer ROSA-Anwendung erstellt und anschließend aus dem Snapshot wiederhergestellt wird.

Snapshot/Wiederherstellung für Anwendungen auf Red Hat OpenShift Service auf AWS (ROSA)-Clustern mit Amazon FSx ONTAP -Speicher

Der Blog

- "Verwenden des Astra Control Service zur Datenverwaltung von Apps auf ROSA-Clustern mit Amazon FSx Speicher"

Schritt-für-Schritt-Anleitung zum Erstellen und Wiederherstellen eines Snapshots

Erforderliche Einrichtung

- "AWS-Konto"
- "Red Hat OpenShift-Konto"
- IAM-Benutzer mit"entsprechende Berechtigungen" zum Erstellen und Zugreifen auf den ROSA-Cluster
- "AWS CLI"
- "ROSA CLI"

- "OpenShift-Befehlszeilenschnittstelle"(oc)
- VPC mit Subnetzen und entsprechenden Gateways und Routen
- "ROSA-Cluster installiert"in die VPC
- "Amazon FSx ONTAP"im selben VPC erstellt
- Zugriff auf den ROSA-Cluster von"OpenShift Hybrid Cloud-Konsole"

Nächste Schritte

1. Erstellen Sie einen Administratorbenutzer und melden Sie sich beim Cluster an.
2. Erstellen Sie eine Kubeconfig-Datei für den Cluster.
3. Installieren Sie Trident auf dem Cluster.
4. Erstellen Sie mit dem Trident CSI Provisioner eine Backend-, Speicherklassen- und Snapshotklassenkonfiguration.
5. Stellen Sie eine PostgreSQL-Anwendung auf dem Cluster bereit.
6. Erstellen Sie eine Datenbank und fügen Sie einen Datensatz hinzu.
7. Fügen Sie den Cluster zu ACS hinzu.
8. Definieren Sie die Anwendung in ACS.
9. Erstellen Sie einen Snapshot mit ACS.
10. Löschen Sie die Datenbank in der PostgreSQL-Anwendung.
11. Wiederherstellung aus einem Snapshot mithilfe von ACS.
12. Überprüfen Sie, ob Ihre App aus dem Snapshot wiederhergestellt wurde.

1. Erstellen Sie einen Administratorbenutzer und melden Sie sich beim Cluster an

Greifen Sie auf den ROSA-Cluster zu, indem Sie mit dem folgenden Befehl einen Administratorbenutzer erstellen: (Sie müssen nur dann einen Administratorbenutzer erstellen, wenn Sie zum Zeitpunkt der Installation noch keinen erstellt haben.)

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

Der Befehl liefert eine Ausgabe, die wie folgt aussieht. Melden Sie sich beim Cluster an mit dem `oc login` Befehl in der Ausgabe bereitgestellt.

```
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
```

Sie können sich auch mit einem Token beim Cluster anmelden. Wenn Sie zum Zeitpunkt der Clustererstellung bereits einen Administratorbenutzer erstellt haben, können Sie sich mit den Anmeldeinformationen des Administratorbenutzers über die Red Hat OpenShift Hybrid Cloud-Konsole beim Cluster anmelden. Klicken Sie dann oben rechts auf den Namen des angemeldeten Benutzers, um die `oc login` Befehl (Token-Login) für die Kommandozeile.

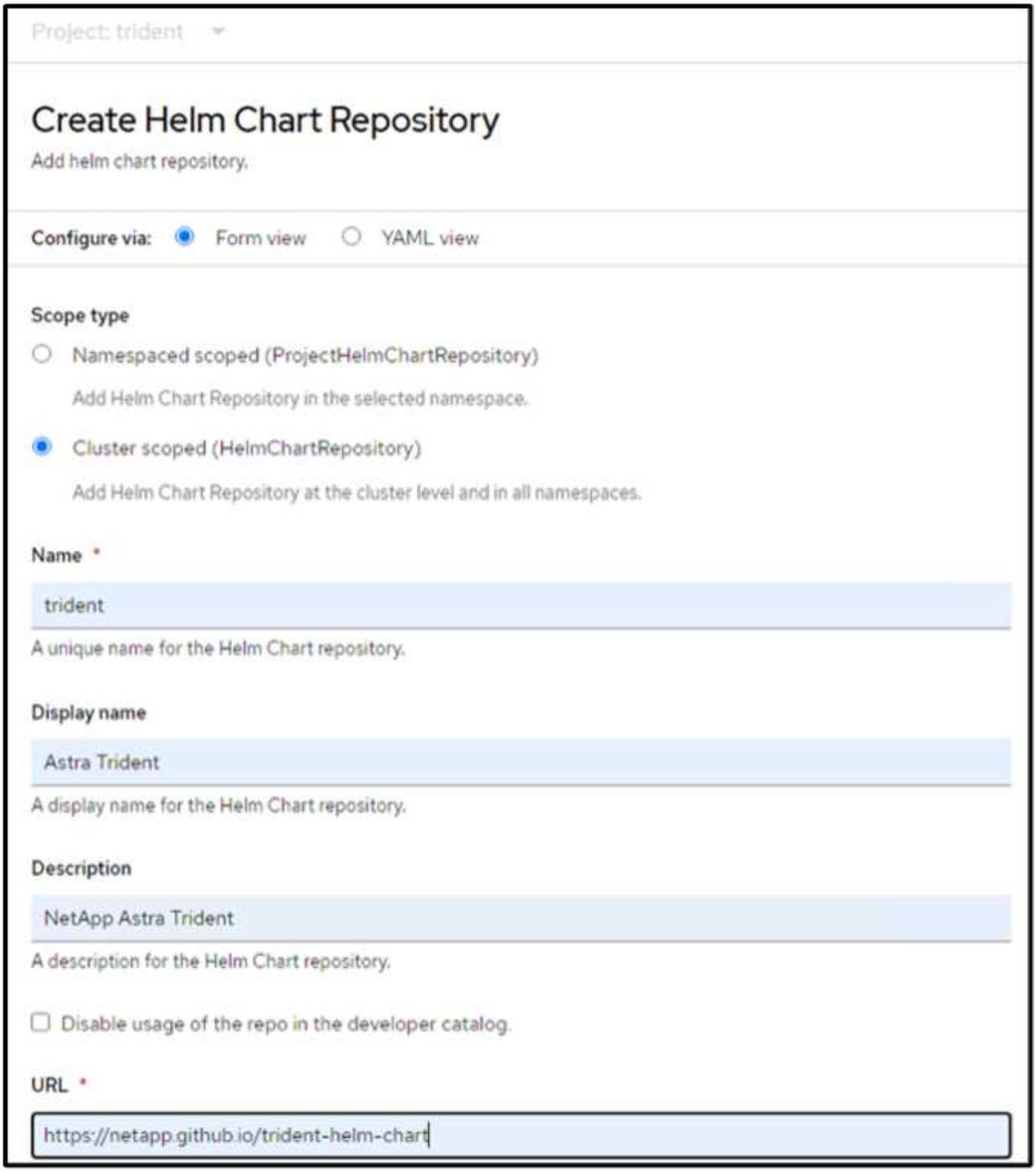
2. Erstellen Sie eine Kubeconfig-Datei für den Cluster

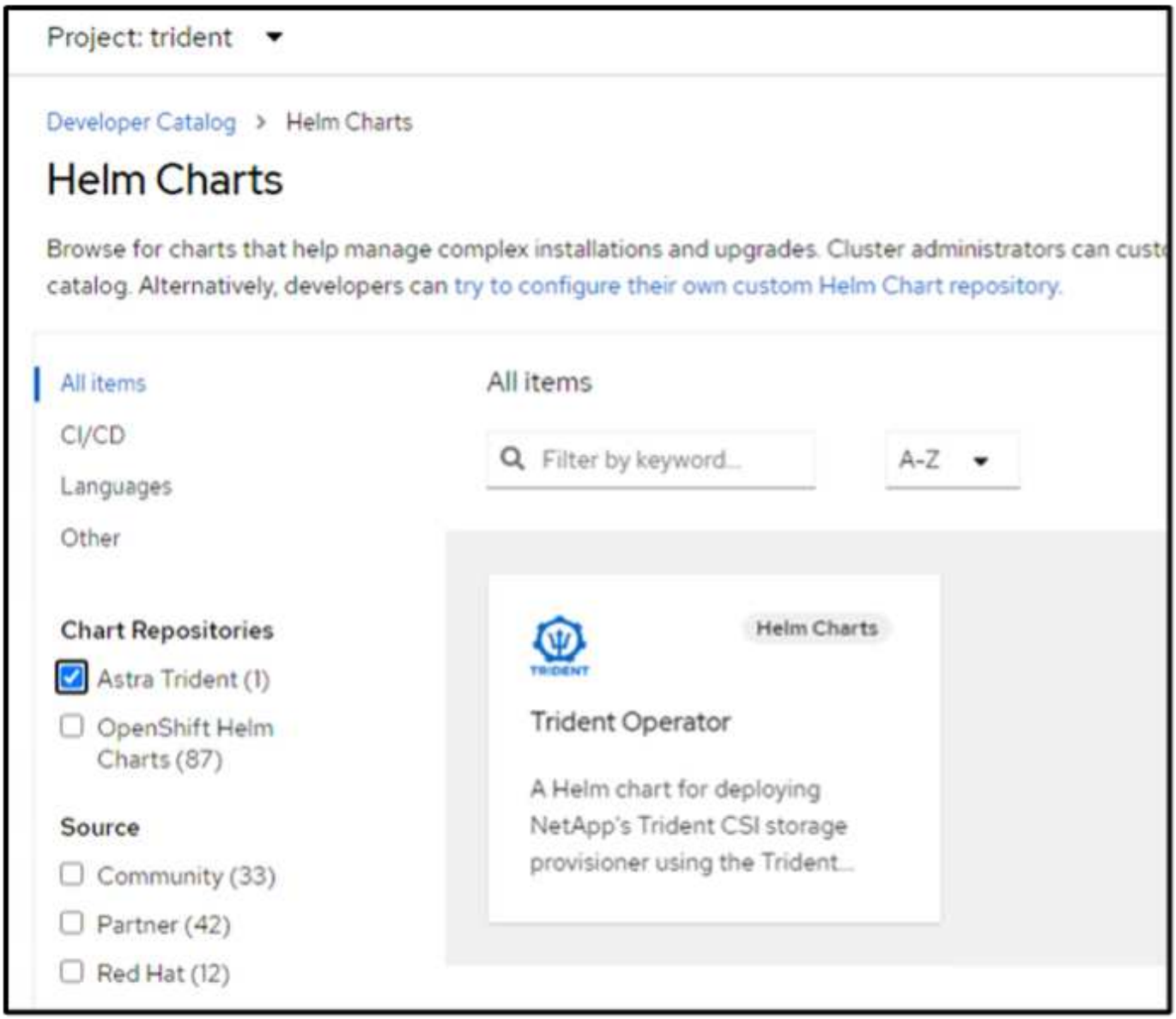
Befolgen Sie die Verfahren"hier," um eine Kubeconfig-Datei für den ROSA-Cluster zu erstellen. Diese Kubeconfig-Datei wird später verwendet, wenn Sie den Cluster zu ACS hinzufügen.

3. Installieren Sie Trident auf dem Cluster

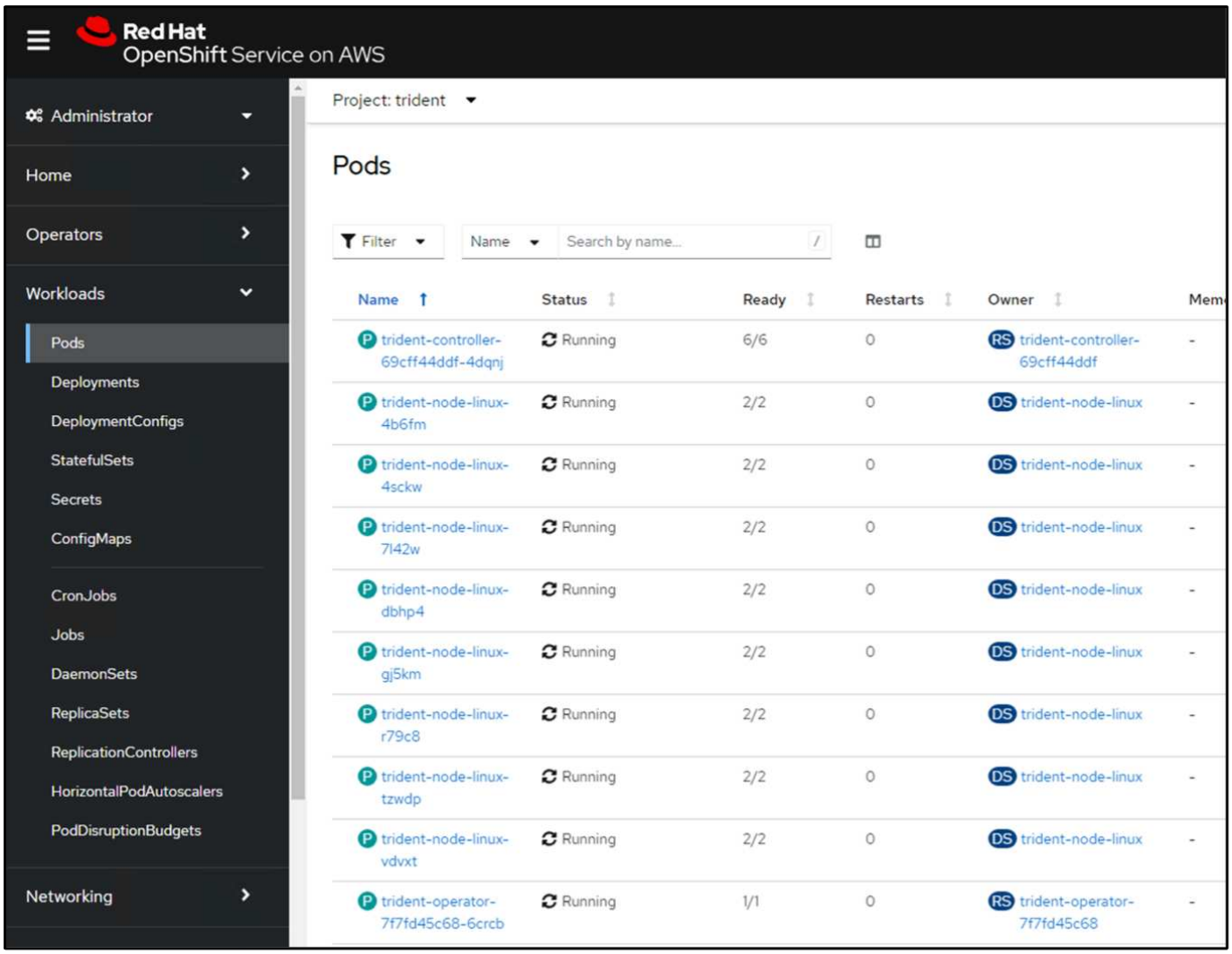
Installieren Sie Trident (neueste Version) auf dem ROSA-Cluster. Dazu können Sie eines der folgenden Verfahren befolgen"hier," . Um Trident mithilfe von Helm von der Konsole des Clusters aus zu installieren, erstellen Sie zunächst ein Projekt namens Trident.

Erstellen Sie dann in der Entwickleransicht ein Helm-Diagramm-Repository. Verwenden Sie für das URL-Feld `'https://netapp.github.io/trident-helm-chart'` . Erstellen Sie dann eine Helmversion für den Trident -Operator.


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



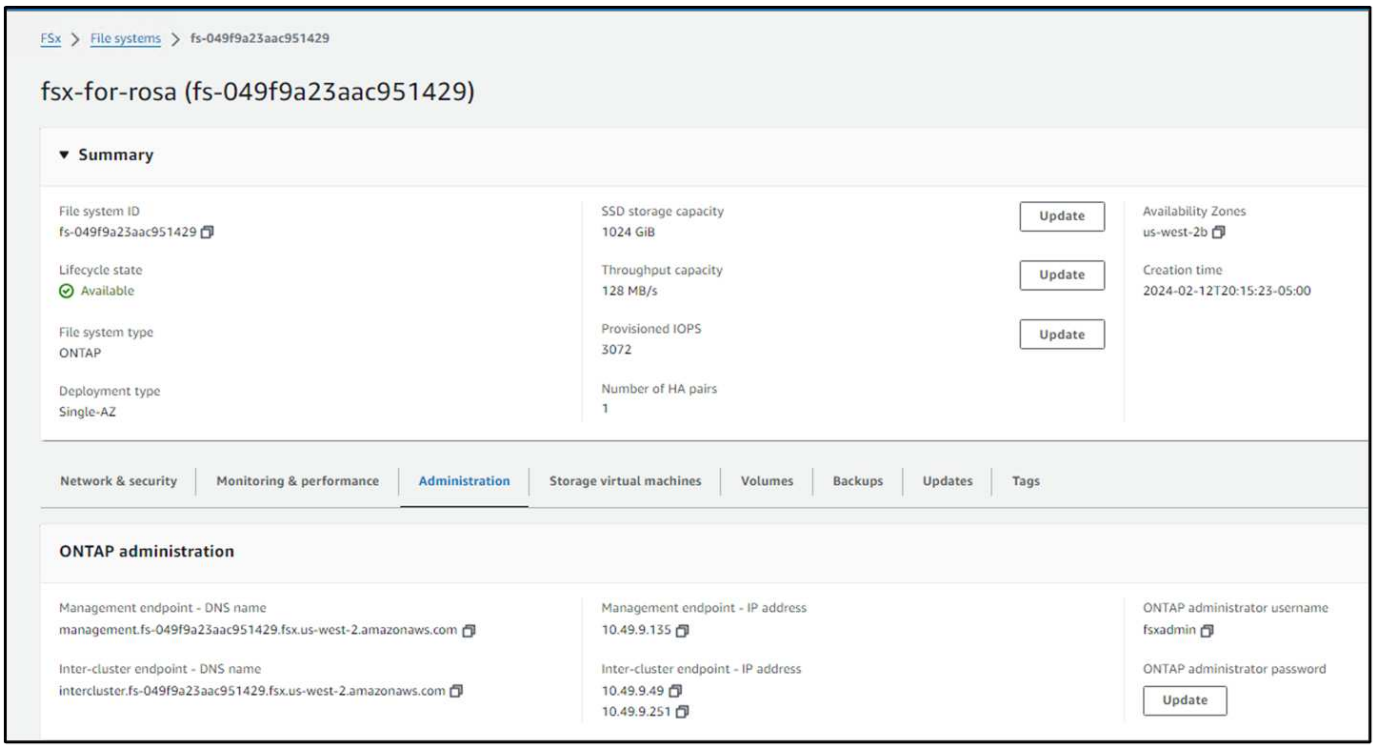
Überprüfen Sie, ob alle Trident-Pods ausgeführt werden, indem Sie zur Administratoransicht auf der Konsole zurückkehren und Pods im Trident-Projekt auswählen.



4. Erstellen Sie eine Backend-, Speicherklassen- und Snapshotklassenkonfiguration mit dem Trident CSI Provisioner

Verwenden Sie die unten gezeigten YAML-Dateien, um ein Trident-Backend-Objekt, ein Speicherklassenobjekt und das Volumesnapshot-Objekt zu erstellen. Geben Sie unbedingt die Anmeldeinformationen für Ihr von Ihnen erstelltes Amazon FSx ONTAP -Dateisystem, das Verwaltungs-LIF und den VServer-Namen Ihres Dateisystems im Konfigurations-YAML für das Backend an. Um diese Details zu erhalten, gehen Sie zur AWS-Konsole für Amazon FSx , wählen Sie das Dateisystem aus und navigieren Sie zur Registerkarte „Administration“. Klicken Sie außerdem auf „Aktualisieren“, um das Kennwort für die `fsxadmin` Benutzer.

Sie können die Objekte über die Befehlszeile erstellen oder sie mit den YAML-Dateien aus der Hybrid Cloud-Konsole erstellen.



- Trident -Backend-Konfiguration**

```
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
```

Speicherklasse

```
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
```

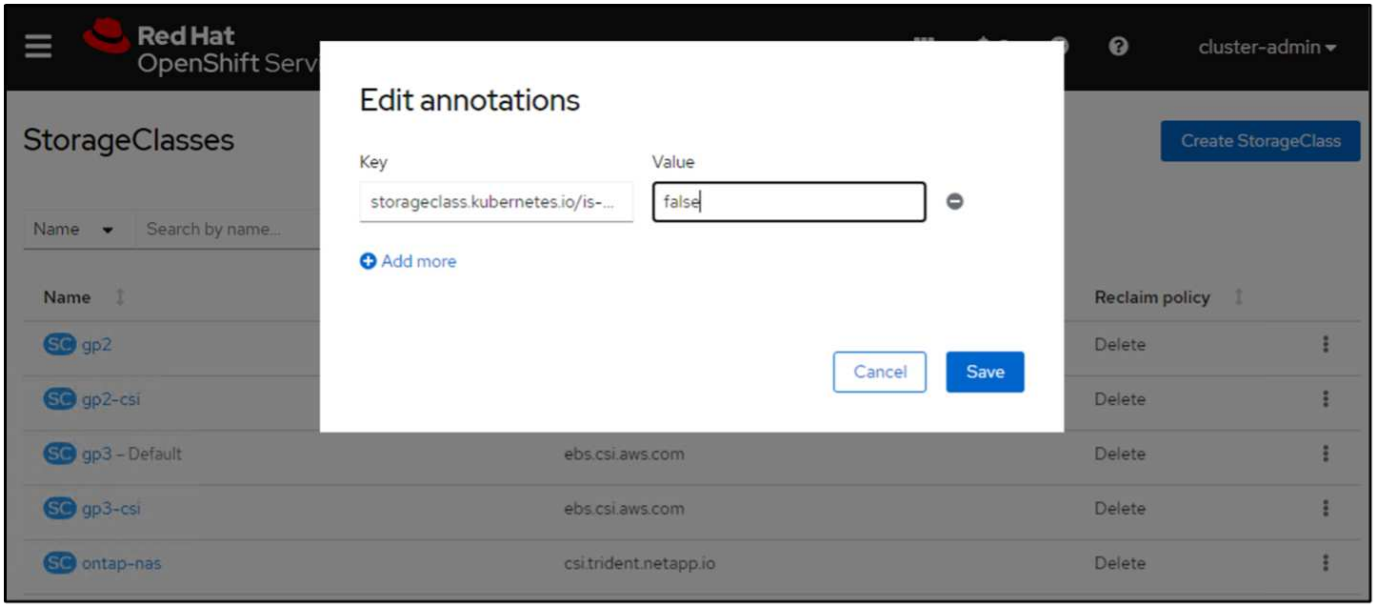
Snapshot-Klasse

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

Überprüfen Sie, ob das Backend, die Speicherklasse und die Trident-Snapshotclass-Objekte erstellt wurden, indem Sie die unten gezeigten Befehle ausführen.

```
[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]$
```

Zu diesem Zeitpunkt müssen Sie eine wichtige Änderung vornehmen: Legen Sie „ontap-nas“ anstelle von „gp3“ als Standardspeicherklasse fest, damit die PostgreSQL-App, die Sie später bereitstellen, die Standardspeicherklasse verwenden kann. Wählen Sie in der Openshift-Konsole Ihres Clusters unter „Speicher“ die Option „StorageClasses“ aus. Bearbeiten Sie die Annotation der aktuellen Standardklasse, sodass sie „false“ ist, und fügen Sie die Annotation storageclass.kubernetes.io/is-default-class hinzu, die für die Speicherklasse ontap-nas auf „true“ gesetzt ist.



5. Stellen Sie eine PostgreSQL-Anwendung auf dem Cluster bereit

Sie können die Anwendung wie folgt über die Befehlszeile bereitstellen:

```
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]$
```

Wenn die Anwendungspods nicht ausgeführt werden, liegt möglicherweise ein Fehler vor, der auf Sicherheitskontextbeschränkungen zurückzuführen ist.

```
[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]$
```

Beheben Sie den Fehler, indem Sie die `runAsUser` Und `fsGroup` Felder in `statefuleset.apps/postgresql` Objekt mit der UID, die in der Ausgabe des `oc get project` Befehl wie unten
gezeigt.

```
[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]$
```

Die PostgreSQL-App sollte ausgeführt werden und persistente Volumes verwenden, die durch Amazon FSx ONTAP Speicher unterstützt werden.

```
[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]$
```

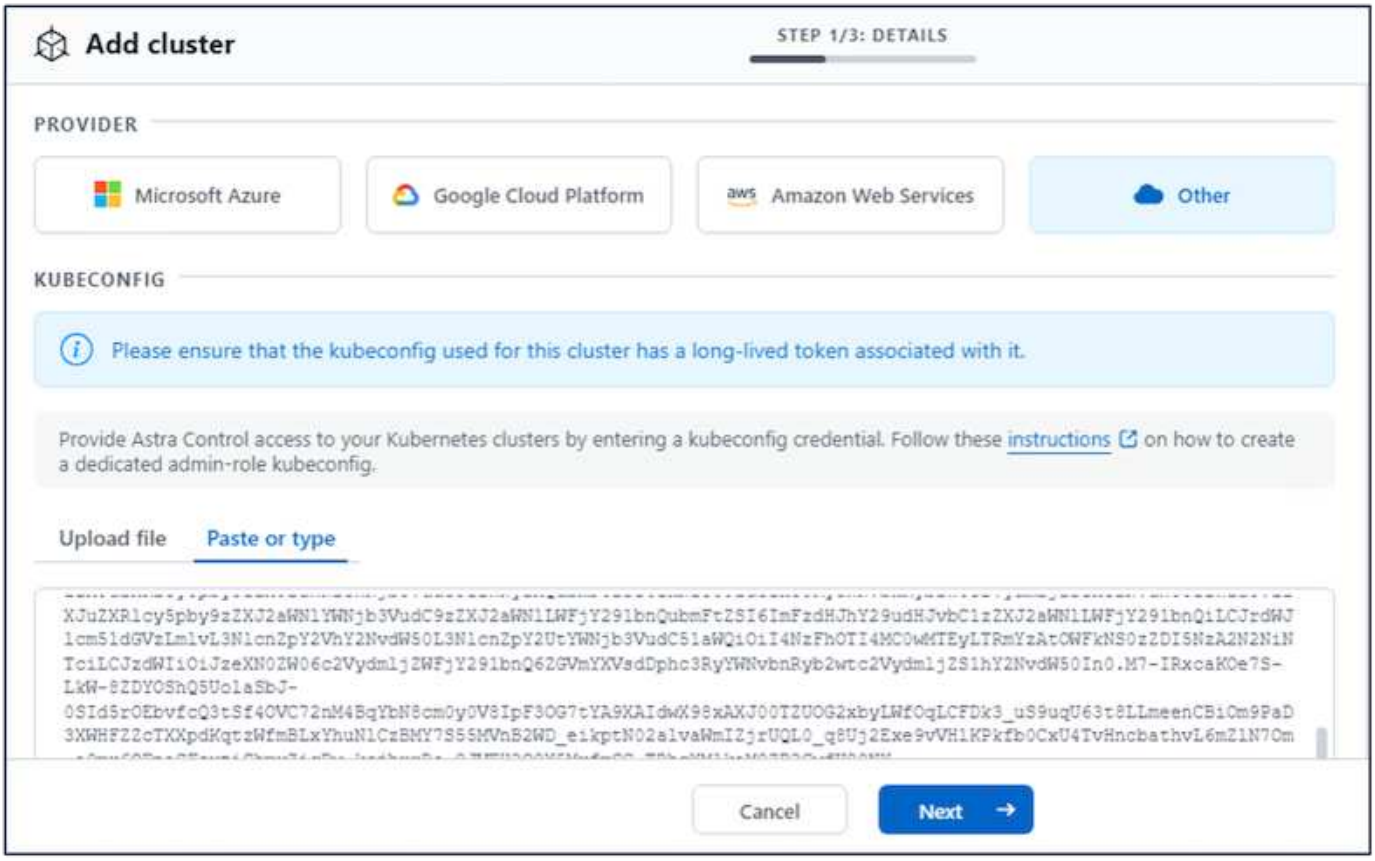
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME               STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0  Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

6. Erstellen Sie eine Datenbank und fügen Sie einen Datensatz hinzu

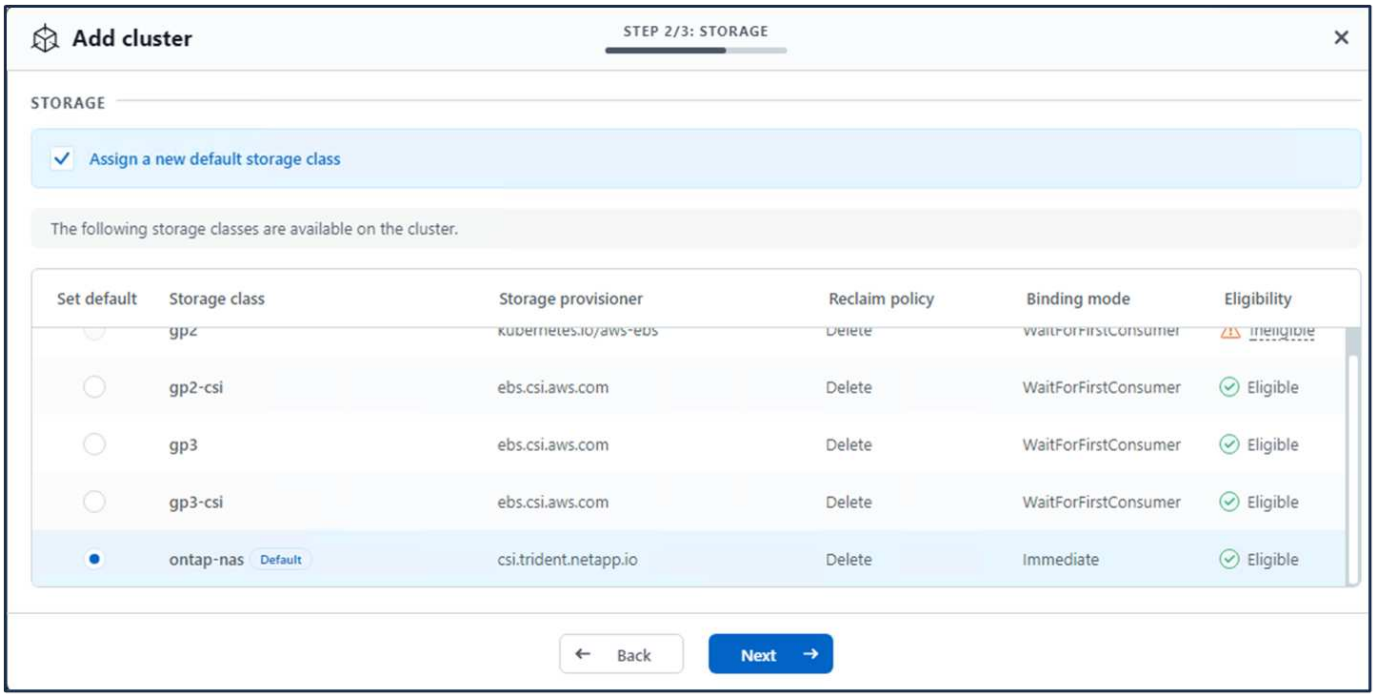
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1 --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. Fügen Sie den Cluster zu ACS hinzu

Melden Sie sich bei ACS an. Wählen Sie den Cluster aus und klicken Sie auf „Hinzufügen“. Wählen Sie „Andere“ aus und laden Sie die Kubeconfig-Datei hoch oder fügen Sie sie ein.



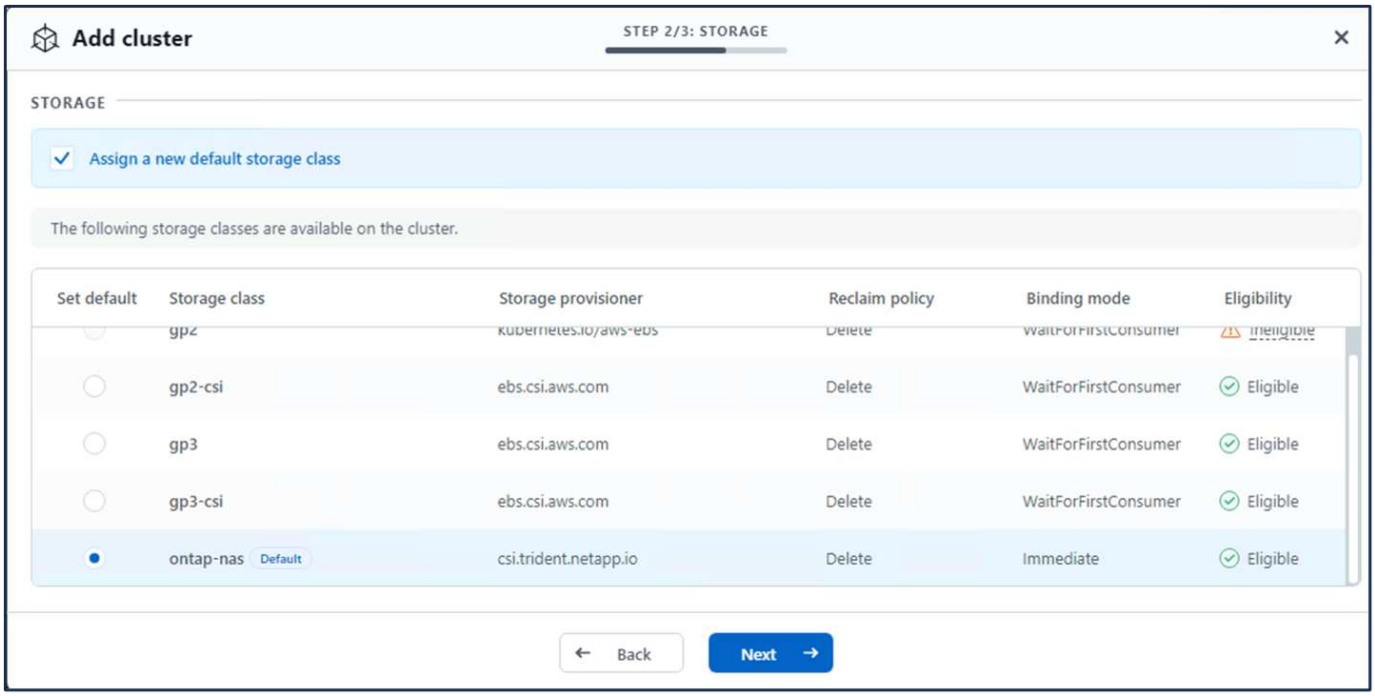
Klicken Sie auf **Weiter** und wählen Sie ontap-nas als Standardspeicherklasse für ACS. Klicken Sie auf **Weiter**, überprüfen Sie die Details und **Fügen Sie** den Cluster hinzu.



8. Definieren Sie die Anwendung in ACS

Definieren Sie die PostgreSQL-Anwendung in ACS. Wählen Sie auf der Zielseite **Anwendungen**, **Definieren** aus und geben Sie die entsprechenden Details ein. Klicken Sie einige Male auf **Weiter**, überprüfen Sie die

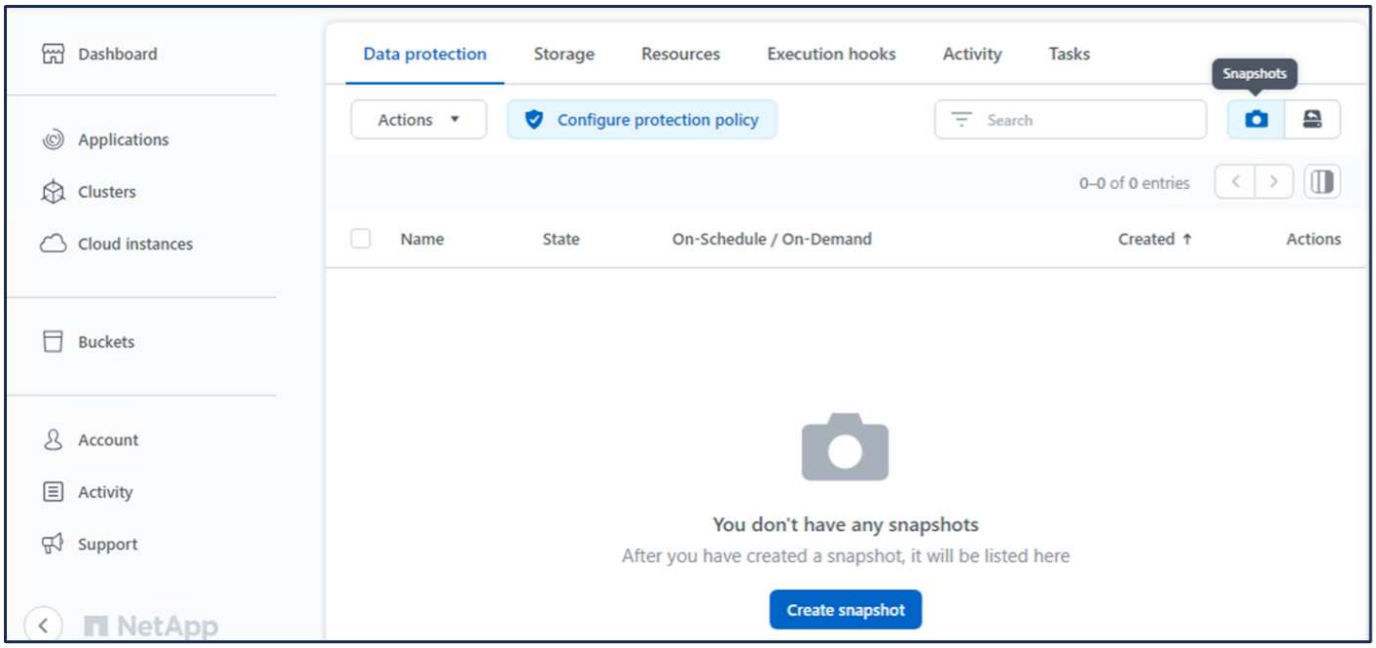
Details und klicken Sie auf **Definieren**. Die Anwendung wird zu ACS hinzugefügt.

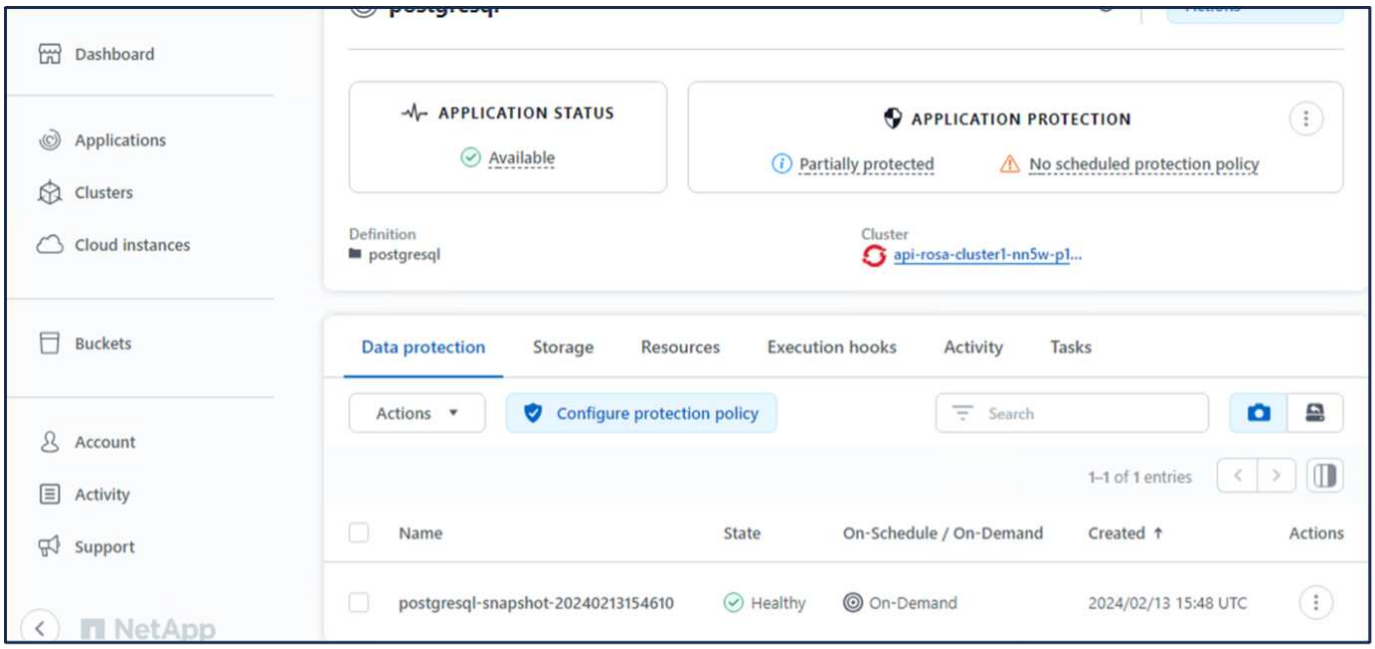


9. Erstellen Sie einen Snapshot mit ACS

Es gibt viele Möglichkeiten, einen Snapshot in ACS zu erstellen. Sie können die Anwendung auswählen und von der Seite aus einen Snapshot erstellen, der die Details der Anwendung anzeigt. Sie können auf „Snapshot erstellen“ klicken, um einen On-Demand-Snapshot zu erstellen oder eine Schutzrichtlinie zu konfigurieren.

Erstellen Sie einen On-Demand-Snapshot, indem Sie einfach auf **Snapshot erstellen** klicken, einen Namen angeben, die Details überprüfen und auf **Snapshot** klicken. Der Snapshot-Status ändert sich nach Abschluss des Vorgangs in „Fehlerfrei“.





10. Löschen Sie die Datenbank in der PostgreSQL-Anwendung

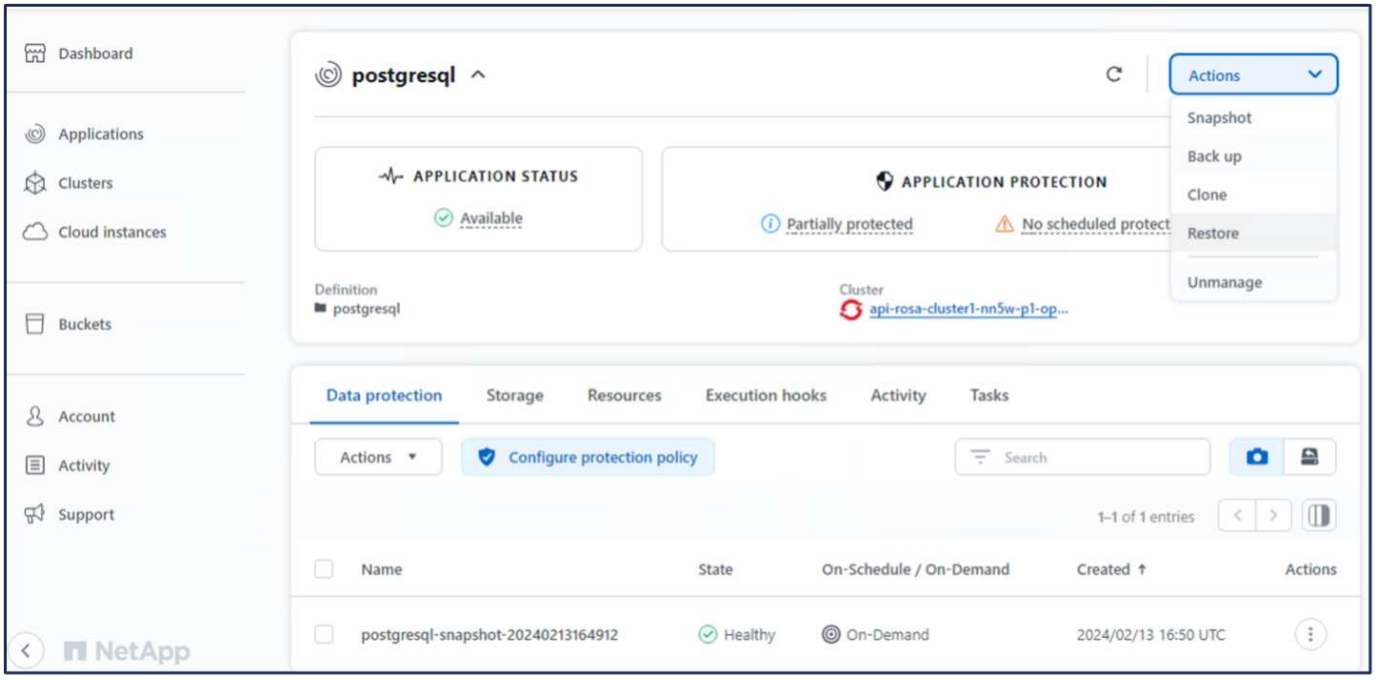
Melden Sie sich erneut bei PostgreSQL an, listen Sie die verfügbaren Datenbanken auf, löschen Sie die zuvor erstellte Datenbank und listen Sie sie erneut auf, um sicherzustellen, dass die Datenbank gelöscht wurde.

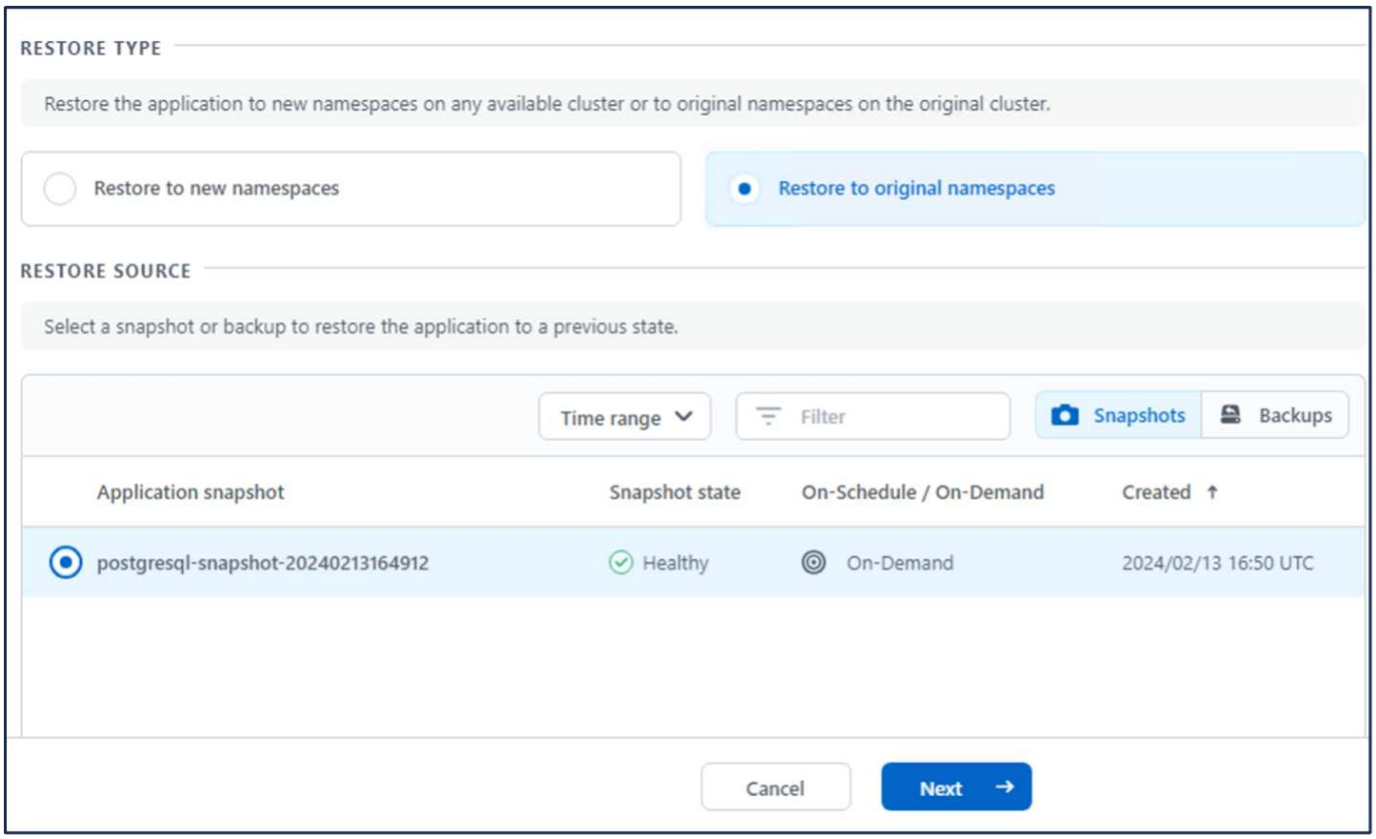
```
postgres=# \l
                                                 List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-------------
 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)
```

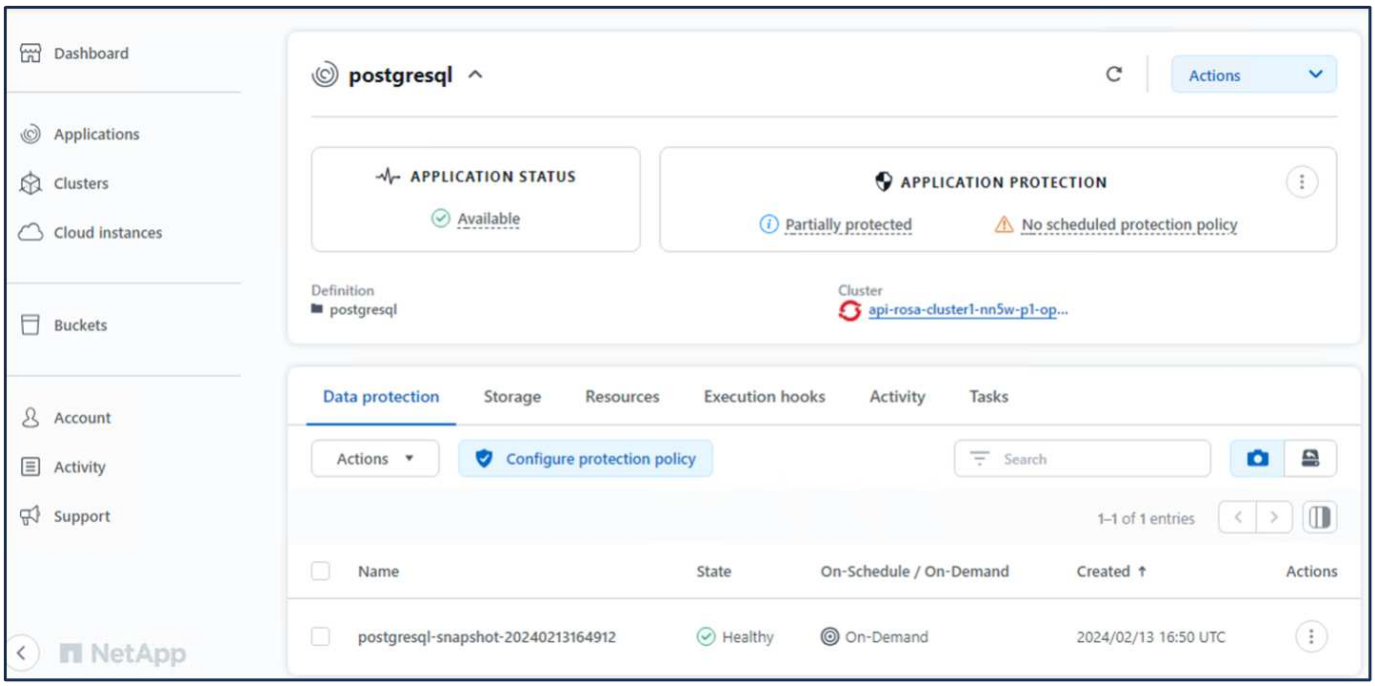
11. Wiederherstellung aus einem Snapshot mit ACS

Um die Anwendung aus einem Snapshot wiederherzustellen, gehen Sie zur ACS-UI-Startseite, wählen Sie die Anwendung aus und wählen Sie „Wiederherstellen“. Sie müssen einen Snapshot oder ein Backup auswählen,

aus dem die Wiederherstellung erfolgen soll. (Normalerweise werden basierend auf einer von Ihnen konfigurierten Richtlinie mehrere erstellt.) Treffen Sie auf den nächsten Bildschirmen die entsprechenden Entscheidungen und klicken Sie dann auf **Wiederherstellen**. Der Anwendungsstatus ändert sich von „Wird wiederhergestellt“ in „Verfügbar“, nachdem die Anwendung aus dem Snapshot wiederhergestellt wurde.







12. Überprüfen Sie, ob Ihre App aus dem Snapshot wiederhergestellt wurde

Melden Sie sich beim PostgreSQL-Client an. Sie sollten nun die Tabelle und den Datensatz in der Tabelle sehen, die Sie zuvor hatten. Das ist es. Mit nur einem Mausklick wird Ihre Anwendung in den vorherigen Zustand zurückversetzt. So einfach machen wir es unseren Kunden mit 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)
```


Datenmigration

Auf dieser Seite werden die Datenmigrationsoptionen für Container-Workloads auf verwalteten Red Hat OpenShift-Clustern angezeigt, die FSx ONTAP für persistenten Speicher verwenden.

Datenmigration

Der Red Hat OpenShift-Dienst auf AWS sowie Amazon FSx for NetApp ONTAP (FSx ONTAP) sind Teil des Serviceportfolios von AWS. FSx ONTAP ist in den Optionen Single AZ oder Multi-AZ verfügbar. Die Multi-Az-Option bietet Datenschutz vor Verfügbarkeitszonenausfällen. FSx ONTAP kann mit Trident integriert werden, um dauerhaften Speicher für Anwendungen auf ROSA-Clustern bereitzustellen.

Integration von FSx ONTAP mit Trident mithilfe des Helm-Diagramms

ROSA-Cluster-Integration mit Amazon FSx ONTAP

Die Migration von Containeranwendungen umfasst:

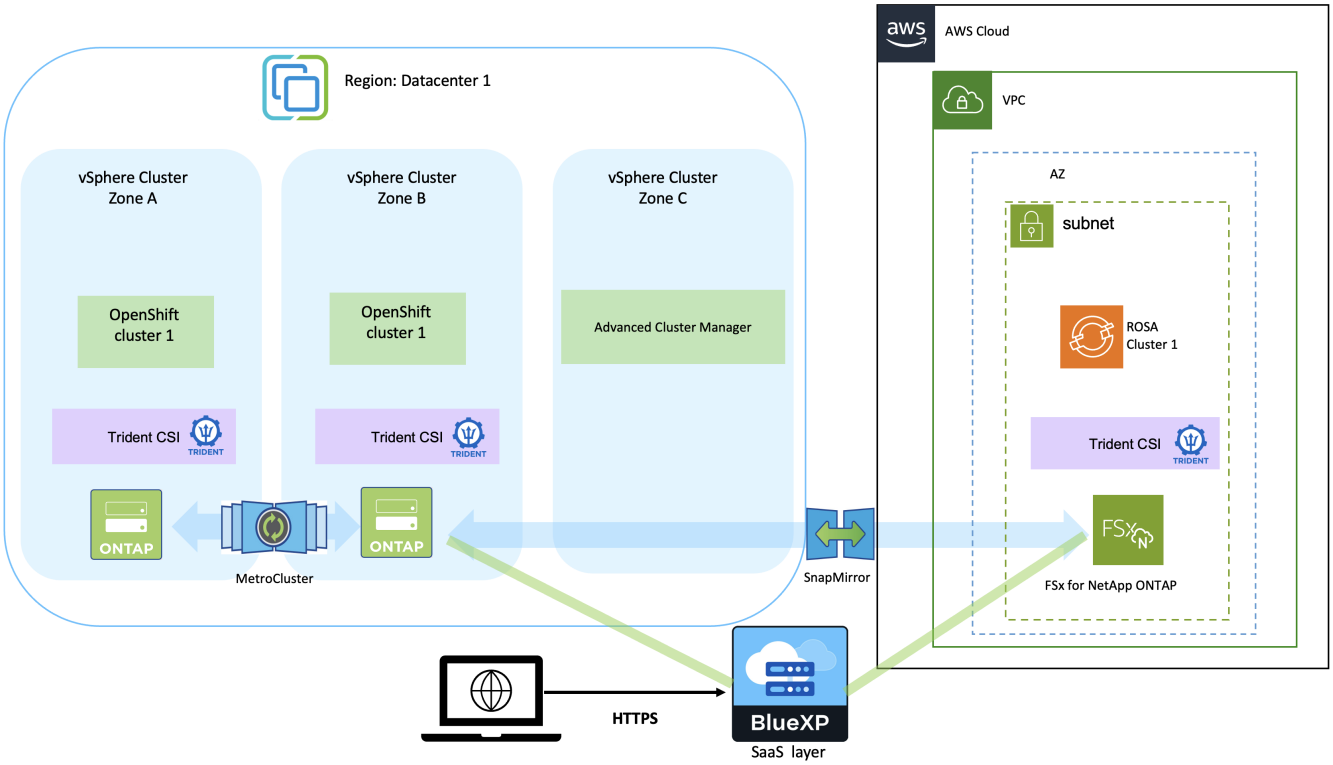
- Persistente Volumes: Dies kann mit BlueXP erreicht werden. Eine weitere Option besteht darin, Trident Protect für die Migration von Containeranwendungen von lokalen in die Cloud-Umgebung zu verwenden. Für denselben Zweck kann auch die Automatisierung eingesetzt werden.
- Anwendungsmetadaten: Dies kann mit OpenShift GitOps (Argo CD) erreicht werden.

Failover und Failback von Anwendungen auf ROSA-Clustern mit FSx ONTAP für persistente Speicherung

Das folgende Video ist eine Demonstration von Anwendungs-Failover- und Failback-Szenarien mit BlueXP und Argo CD.

Failover und Failback von Anwendungen auf dem ROSA-Cluster

Datenschutz- und Migrationslösung für OpenShift-Container-Workloads



Zusätzliche NetApp Hybrid Multicloud-Lösungen für Red Hat OpenShift-Workloads

Zusätzliche Lösungen

Weitere Lösungen sind in den folgenden Abschnitten verfügbar:

Informationen zu Red Hat OpenShift Container-Lösungen finden Sie unter"hier," .

Informationen zu lokalen Red Hat OpenShift-Virtualisierungslösungen finden Sie unter"hier," .