Hybrid Cloud mit vom Provider gemanagten Komponenten

NetApp Solutions

NetApp
April 26, 2024

Inhalt

NetApp Hybrid-Multi-Cloud-Lösungen für Container-Workloads mit Red hat OpenShift . . . 1
Überblick . . . 1
Die NetApp Lösung mit gemanagten Container-Plattform-Workloads aus Red hat OpenShift auf AWS . . . 3
Implementierung und Konfiguration der gemanagten Container-Plattform Red hat OpenShift auf AWS . . . 5
Datensicherung . . . 7
Datenmigration . . . 23

NetApp Hybrid-Multi-Cloud-Lösungen für Container-Workloads mit Red hat OpenShift

Überblick

Bei NetApp beobachten wir eine deutliche Zunahme bei Kunden, die ihre älteren Enterprise-Applikationen modernisieren und neue Applikationen mithilfe von Containern und Orchestrierungsplattformen auf Basis von Kubernetes erstellen. Die Red hat OpenShift Container Platform ist ein Beispiel, das wir von vielen unserer Kunden angenommen sehen.

Immer mehr Kunden beginnen mit der Einführung von Containern in ihrem Unternehmen. NetApp ist perfekt aufgestellt, um die persistenten Storage-Anforderungen ihrer zustandsbehafteten Applikationen und klassischen Datenmanagementanforderungen zu erfüllen, beispielsweise Datensicherung, Datensicherheit und Datenmigration. Diese Bedürfnisse werden jedoch mit verschiedenen Strategien, Werkzeugen und Methoden erfüllt.

NetApp ONTAP basierte Storage-Optionen, die unten aufgeführt sind, bieten Sicherheit, Datensicherung, Zuverlässigkeit und Flexibilität für Container- und Kubernetes-Implementierungen.

- Automatisierter, lokaler Storage:
  - NetApp Fabric Attached Storage (FAS), NetApp All-Flash-FAS-Arrays (AFF), NetApp All-SAN-Arrays (ASA) und ONTAP Select
- Von Providern gemanagter Storage in On-Premises:
  - NetApp Keystone stellt Storage-als-Service (STaaS) bereit
- Automatisierter Storage in der Cloud:
  - NetApp Cloud Volumes ONTAP(CVO) bieten Self-Managed-Storage in den Hyperscalern
- Von Providern gemanagter Storage in der Cloud:
  - Cloud Volumes Service für Google Cloud (CVS), Azure NetApp Files (ANF) und Amazon FSX für NetApp ONTAP bieten vollständig gemanagten Storage in den Hyperscalern

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)

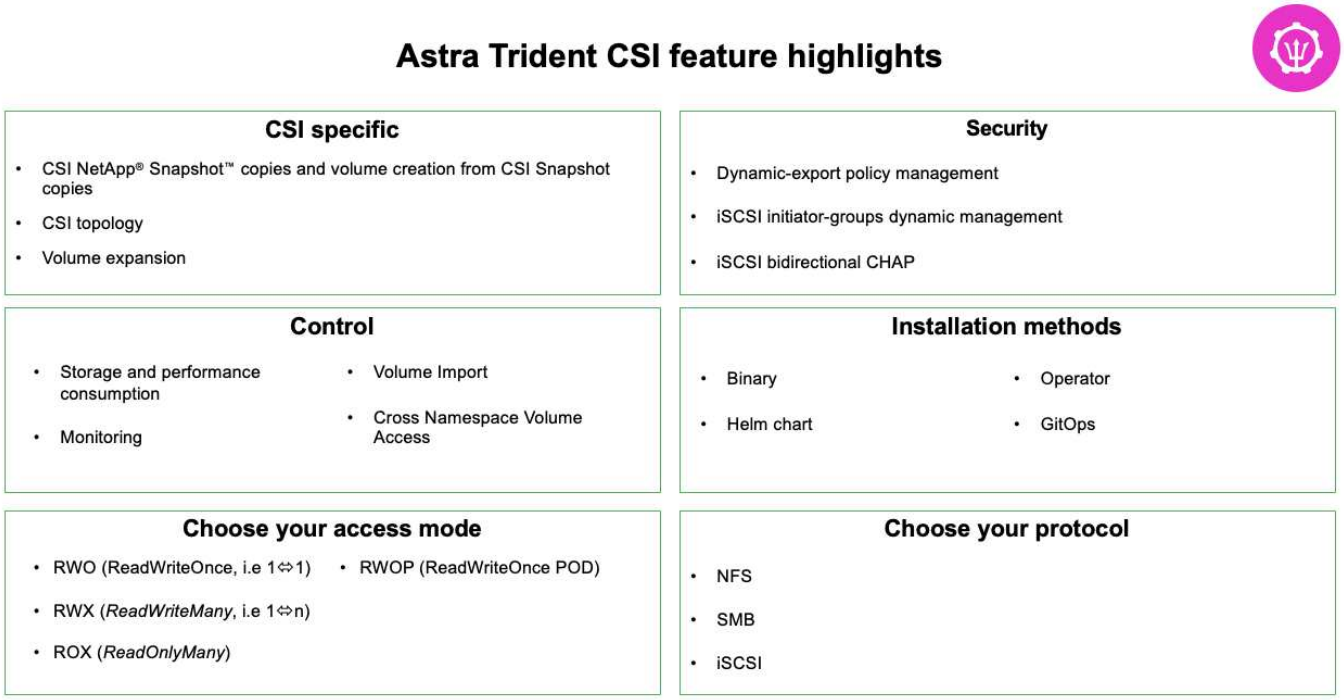
Security & Compliance

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

NetApp BlueXP ermöglicht Ihnen das Management Ihrer gesamten Storage- und Datenbestände über eine einzige Managementoberfläche.

Mit BlueXP können Sie Cloud-Storage erstellen und verwalten (z. B. Cloud Volumes ONTAP und Azure NetApp Files), Daten verschieben, schützen und analysieren sowie zahlreiche lokale und Edge-Storage-Geräte kontrollieren.

NetApp Astra Trident ist ein CSI-konformer Storage Orchestrator, der eine schnelle und einfache Nutzung von persistentem Storage ermöglicht und von einer Vielzahl der oben genannten NetApp Storage-Optionen unterstützt wird. Es handelt sich um eine Open-Source-Software, die von NetApp gewartet und unterstützt wird.



Geschäftskritische Container-Workloads benötigen mehr als nur persistente Volumes. Ihre Datenmanagement-Anforderungen erfordern auch den Schutz und die Migration der Kubernetes-Objekte für die Applikation.

> Neben Benutzerdaten umfassen Applikationsdaten kubernetes-Objekte. Folgende Beispiele sind vorhanden: - kubernetes-Objekte wie Pods-Spezifikationen, PVCs, Implementierungen, Services – benutzerdefinierte Konfigurationsobjekte wie Konfigurationszuordnungen und -Geheimnisse – persistente Daten wie Snapshot-Kopien, Backups, Klone – benutzerdefinierte Ressourcen wie CRS und CRDs

NetApp Astra Control ist sowohl als vollständig gemanagte als auch als selbst gemanagte Software erhältlich und bietet Orchestrierung für solides Applikations-Datenmanagement. Siehe "Astra-Dokumentation" Weitere Informationen zur Astra Produktfamilie.

Diese Referenzdokumentation unterstützt die Validierung der Migration und des Schutzes von Container-basierten Applikationen, die auf der RedHat OpenShift Container-Plattform über das NetApp Astra Control Center implementiert werden. Darüber hinaus bietet die Lösung allgemeine Details zur Bereitstellung und zur Verwendung von Red hat Advanced Cluster Management (ACM) für die Verwaltung der Container-Plattformen. In diesem Dokument werden auch Einzelheiten zur Integration von NetApp Storage in Container-Plattformen mit Red hat OpenShift mithilfe der Astra Trident CSI-bereitstellung erläutert. Astra Control Center wird auf dem Hub-Cluster bereitgestellt und wird für das Management der Container-Applikationen und ihres Lebenszyklus von persistentem Storage verwendet. Schließlich bietet es eine Lösung für Replizierung, Failover und Failback für Container-Workloads auf gemanagten Red hat OpenShift-Clustern in AWS (ROSA), die Amazon FSX für NetApp ONTAP (FSxN) als persistenten Storage verwenden.

Die NetApp Lösung mit gemanagten Container-Plattform-Workloads aus Red hat OpenShift auf AWS

Die NetApp Lösung mit gemanagten Container-Plattform-Workloads aus Red hat OpenShift auf AWS

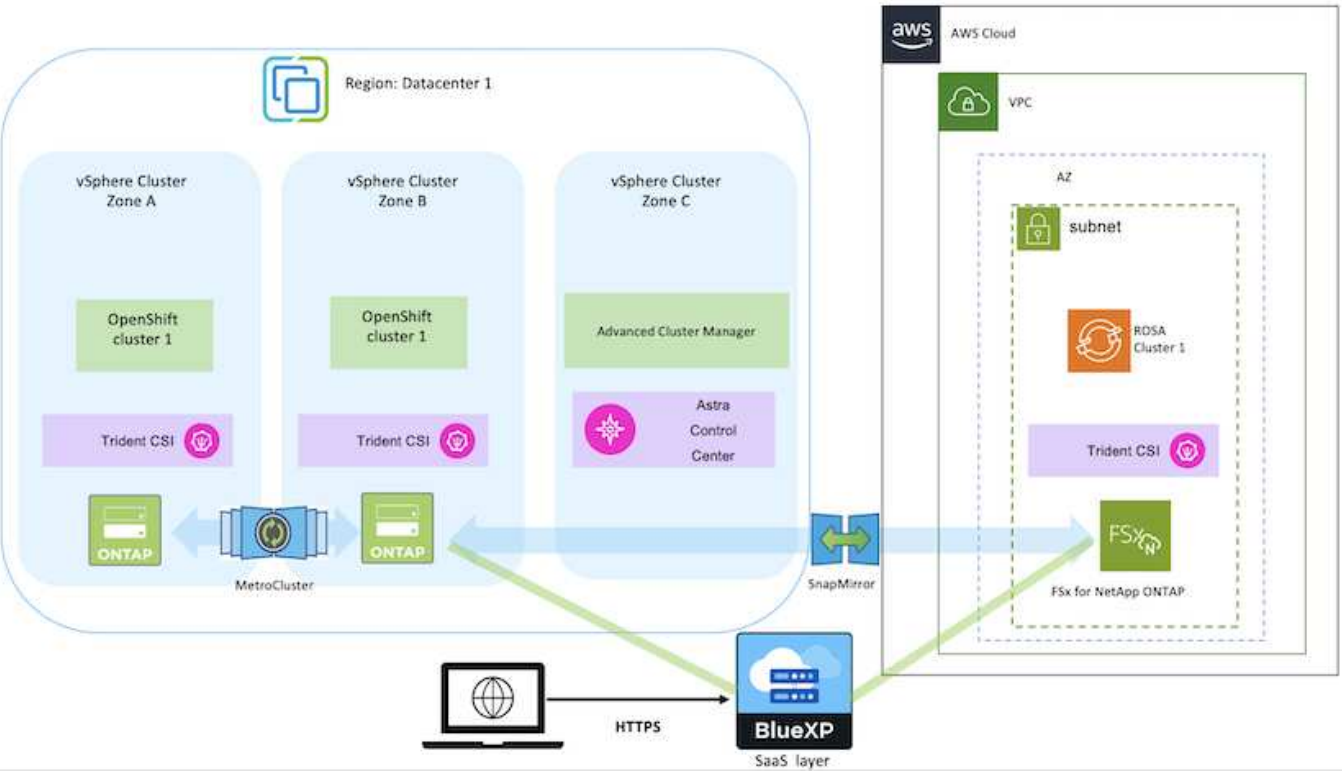
Möglicherweise sind Kunden „aus der Cloud hervorgegangen“ oder bereits an einem Punkt der Modernisierung angelangt, wenn sie bereit sind, einige ausgewählte Workloads oder alle Workloads aus ihrem Datacenter in die Cloud zu verschieben. Sie können dafür wählen, von Providern gemanagte OpenShift-Container und von Providern gemanagten NetApp Storage in der Cloud zu verwenden, um ihre Workloads auszuführen. Sie sollten die verwalteten Container-Cluster (ROSA) von Red hat OpenShift in der Cloud planen und bereitstellen, um eine erfolgreiche produktionsbereite Umgebung für ihre Container-Workloads zu schaffen. In der AWS-Cloud können sie auch FSX für NetApp ONTAP für die Storage-Anforderungen implementieren.

FSX for NetApp ONTAP bietet Datensicherung, Zuverlässigkeit und Flexibilität für Container-Implementierungen in AWS. Astra Trident dient als dynamische Storage-bereitstellung zur Nutzung des persistenten FSxN Storage für zustandsbehaftete Applikationen von Kunden.

DA ROSA im HA-Modus mit Knoten der Kontrollebene über mehrere Verfügbarkeitszonen hinweg implementiert werden kann, kann FSX ONTAP auch mit Multi-AZ-Option bereitgestellt werden, die hohe Verfügbarkeit bietet und AZ-Ausfälle schützt.

> Beim Zugriff auf ein Amazon FSX Filesystem aus der bevorzugten Verfügbarkeitszone (AZ) des Filesystems fallen keine Datenübertragungsgebühren an. Weitere Informationen zur Preisgestaltung finden Sie unter "Hier".

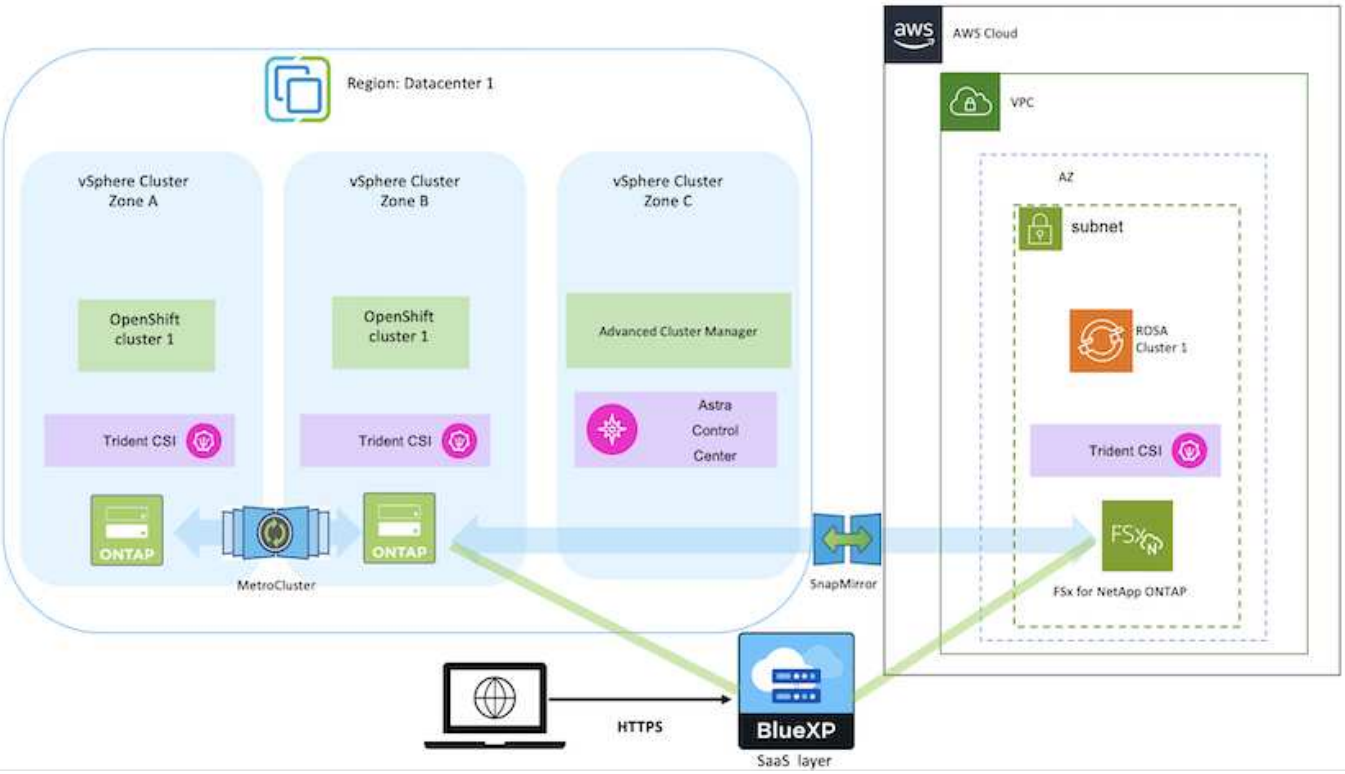
Datensicherungs- und Migrationslösung für OpenShift-Container-Workloads



Implementierung und Konfiguration der gemanagten Container-Plattform Red hat OpenShift auf AWS

In diesem Abschnitt wird ein High-Level-Workflow zur Einrichtung der verwalteten Red hat OpenShift-Cluster auf AWS(ROSA) beschrieben. Es zeigt die Nutzung von Managed FSX for NetApp ONTAP (FSxN) als Storage-Backend von Astra Trident zur Bereitstellung persistenter Volumes. Es werden Details zur Implementierung von FSxN auf AWS mithilfe von BlueXP bereitgestellt. Außerdem werden Einzelheiten zur Verwendung von BlueXP und OpenShift GitOps (Argo CD) bereitgestellt, um Datensicherungs- und Migrationsaktivitäten für die zustandsbehafteten Applikationen auf ROSA Clustern durchzuführen.

Das folgende Diagramm zeigt die AUF AWS implementierten ROSA-Cluster, die FSxN als Back-End-Storage verwenden.



Diese Lösung wurde mit zwei ROSA-Clustern in zwei VPCs in AWS verifiziert. Jeder ROSA Cluster wurde mithilfe von Astra Trident in FSxN integriert. ES gibt mehrere Möglichkeiten, ROSA-Cluster und FSxN in AWS bereitzustellen. Diese allgemeine Beschreibung des Setups enthält Dokumentations-Links für die spezifische verwendete Methode. Weitere Methoden finden Sie in den entsprechenden Links im "Ressourcen".

Der Einrichtungsvorgang kann in die folgenden Schritte unterteilt werden:

INSTALLIEREN SIE ROSA Cluster

- Erstellung von zwei VPCs und Einrichtung der VPC-Peering-Konnektivität zwischen den VPCs.
- Siehe "Hier" Für Anweisungen zur Installation VON ROSA Clustern.

Installieren Sie FSxN

- Installieren Sie FSxN auf den VPCs von BlueXP. Siehe "Hier" Für die Erstellung von BlueXP Konten und weitere Schritte. Siehe "Hier" Zur Installation von FSxN. Siehe "Hier" Zum Erstellen eines Connectors in AWS zum Verwalten des FSxN.
- Implementieren Sie FSxN mithilfe von AWS. Siehe "Hier" Für die Implementierung über die AWS-Konsole.

Trident auf ROSA Clustern installieren (mit Helm-Diagramm)

- Verwenden Sie Helm-Diagramm, um Trident auf ROSA Clustern zu installieren. url für das Helm-Diagramm: https://netapp.github.io/trident-helm-chart

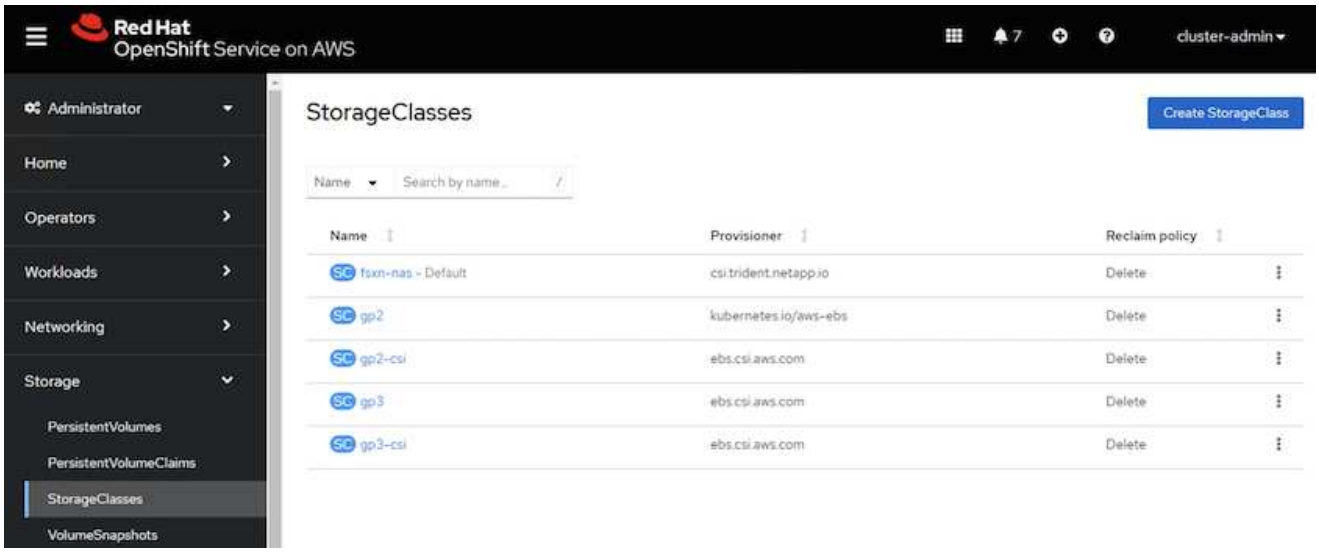
Integration von FSxN mit Astra Trident für ROSA Cluster

OpenShift GitOps kann zur Implementierung von Astra Trident CSI für alle gemanagten Cluster verwendet werden, wenn sie über ApplicationSet auf 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
```

Back-End- und Storage-Klassen mit Trident (für FSxN) erstellen

- Siehe "Hier" Für Details zum Erstellen von Back-End und Storage-Klasse.
- Erstellen Sie die für FsxN erstellte Storage-Klasse mit Trident CSI standardmäßig aus der OpenShift-Konsole. Siehe Abbildung unten:



Anwendung mit OpenShift GitOps (Argo CD) bereitstellen

- 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. Als Beispiel können Sie eine Jenkins-App mithilfe einer Argo-CD mit einem Helm-Diagramm bereitstellen. Beim Erstellen der Anwendung wurden folgende Details angegeben: Projekt: Standardcluster: https://kubernetes.default.svcNamensraum: Jenkins die url für das Helm-Diagramm: https://charts.bitnami.com/bitnami

Helm-Parameter: Global.storageClass: Fsxn-nas

Datensicherung

Auf dieser Seite werden die Datensicherungsoptionen für gemanagte Red hat OpenShift auf AWS (ROSA) Clustern unter Verwendung des Astra Control Service angezeigt. Astra Control Service (ACS) bietet eine intuitive grafische Benutzeroberfläche, mit der Sie Cluster hinzufügen, darauf laufende Applikationen definieren und applikationsorientierte Datenmanagement-Aktivitäten durchführen können. ACS-Funktionen können auch über eine API aufgerufen werden, die die Automatisierung von Workflows ermöglicht.

Astra Control (ACS oder ACC) wird von NetApp Astra Trident angetrieben. Astra Trident integriert mehrere Arten von Kubernetes Clustern wie Red hat OpenShift, EKS, AKS, SUSE Rancher, Anthos usw. mit verschiedenen Ausführungen von NetApp ONTAP-Storage wie FAS/All Flash FAS, ONTAP Select, CVO, Google Cloud Volumes Service, Azure NetApp Files und Amazon FSX for NetApp ONTAP.

Dieser Abschnitt enthält Details zu den folgenden Datenschutzoptionen, die ACS verwenden:

- Ein Video, das Backup und Restore einer ROSA-Anwendung zeigt, die in einer Region ausgeführt wird und in einer anderen Region wiederhergestellt wird.
- Ein Video, das Snapshot und Wiederherstellung einer ROSA-Anwendung zeigt.
- Schritt-für-Schritt-Details zur Installation eines ROSA-Clusters, Amazon FSX for NetApp ONTAP, Verwendung von NetApp Astra Trident zur Integration mit Storage-Backend, Installation einer postgresql-Anwendung auf ROSA-Cluster, Verwendung von ACS zur Erstellung eines Snapshot der Anwendung und Wiederherstellung der Anwendung von ihm.
- Ein Blog, der Schritt-für-Schritt-Details des Erstellens und Wiederherstellen aus einem Snapshot für eine mysql-Anwendung auf einem ROSA-Cluster mit FSX für ONTAP unter Verwendung von ACS zeigt.

Backup/Wiederherstellung aus Backup

Das folgende Video zeigt die Sicherung einer ROSA-Anwendung, die in einer Region ausgeführt wird 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 Sie einen Snapshot einer ROSA-Anwendung erstellen und danach aus dem Snapshot wiederherstellen.

Snapshot/Wiederherstellung für Anwendungen auf Red hat OpenShift-Service auf AWS (ROSA)-Clustern mit Amazon FSX für NetApp ONTAP-Speicher

Blog

- "Nutzung von Astra Control Service zum Datenmanagement von Applikationen auf ROSA Clustern mit Amazon FSX Storage"

Schritt-für-Schritt-Details zum Erstellen von Snapshot und Wiederherstellen von ihm

Vorbereitende Einrichtung

- "AWS Konto"
- "Red hat OpenShift -Konto"
- IAM-Benutzer mit "Entsprechende Berechtigungen" Um ROSA Cluster zu erstellen und darauf zuzugreifen
- "AWS CLI"
- "ROSA CLI"
- "OpenShift-CLI"(oc)
- VPC mit Subnetzen und entsprechenden Gateways und Routen
- "ROSA Cluster installiert" In die VPC
- "Amazon FSX für NetApp ONTAP" Erstellt in derselben VPC
- Zugriff auf den ROSA-Cluster von "OpenShift Hybrid Cloud Console"

Nächste Schritte

1. Erstellen Sie einen Admin-Benutzer und melden Sie sich beim Cluster an.
2. Erstellen Sie eine kubeconfig-Datei für den Cluster.
3. Installieren Sie Astra Trident auf dem Cluster.
4. Mit der CSI-provisionierung von Trident können Sie eine Back-End-, Storage-Klasse- und Snapshot-Klassenkonfiguration erstellen.
5. Implementieren Sie eine postgresql-Anwendung auf dem Cluster.
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. Wiederherstellen von einem Snapshot mit ACS.
12. Überprüfen Sie, ob Ihre App aus dem Snapshot wiederhergestellt wurde.

1. Erstellen Sie einen Admin-Benutzer und melden Sie sich beim Cluster an

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

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

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

```
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 Cluster-Erstellung bereits einen Admin-Benutzer erstellt haben, können Sie sich über die Red hat OpenShift Hybrid Cloud-Konsole mit den Anmeldedaten des Admin-Benutzers beim Cluster anmelden. Klicken Sie dann auf die obere rechte Ecke, wo der Name des angemeldeten Benutzers angezeigt wird, um den zu erhalten `oc login` Befehl (Token Login) für die Befehlszeile.

2. Erstellen Sie eine kubeconfig-Datei für den Cluster

Befolgen Sie die Anweisungen "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 Astra Trident auf dem Cluster

Installieren Sie Astra Trident (neueste Version) im ROSA Cluster. Um dies zu tun, können Sie eine der angegebenen Verfahren befolgen "Hier". Um Trident über das Helm von der Cluster-Konsole zu installieren, erstellen Sie zuerst ein Projekt mit dem Namen Trident.

Erstellen Sie dann in der Entwickleransicht ein Helmdiagramm-Repository. Verwenden Sie für das URL-Feld `'https://netapp.github.io/trident-helm-chart'`. Erstellen Sie dann ein Helm Release 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

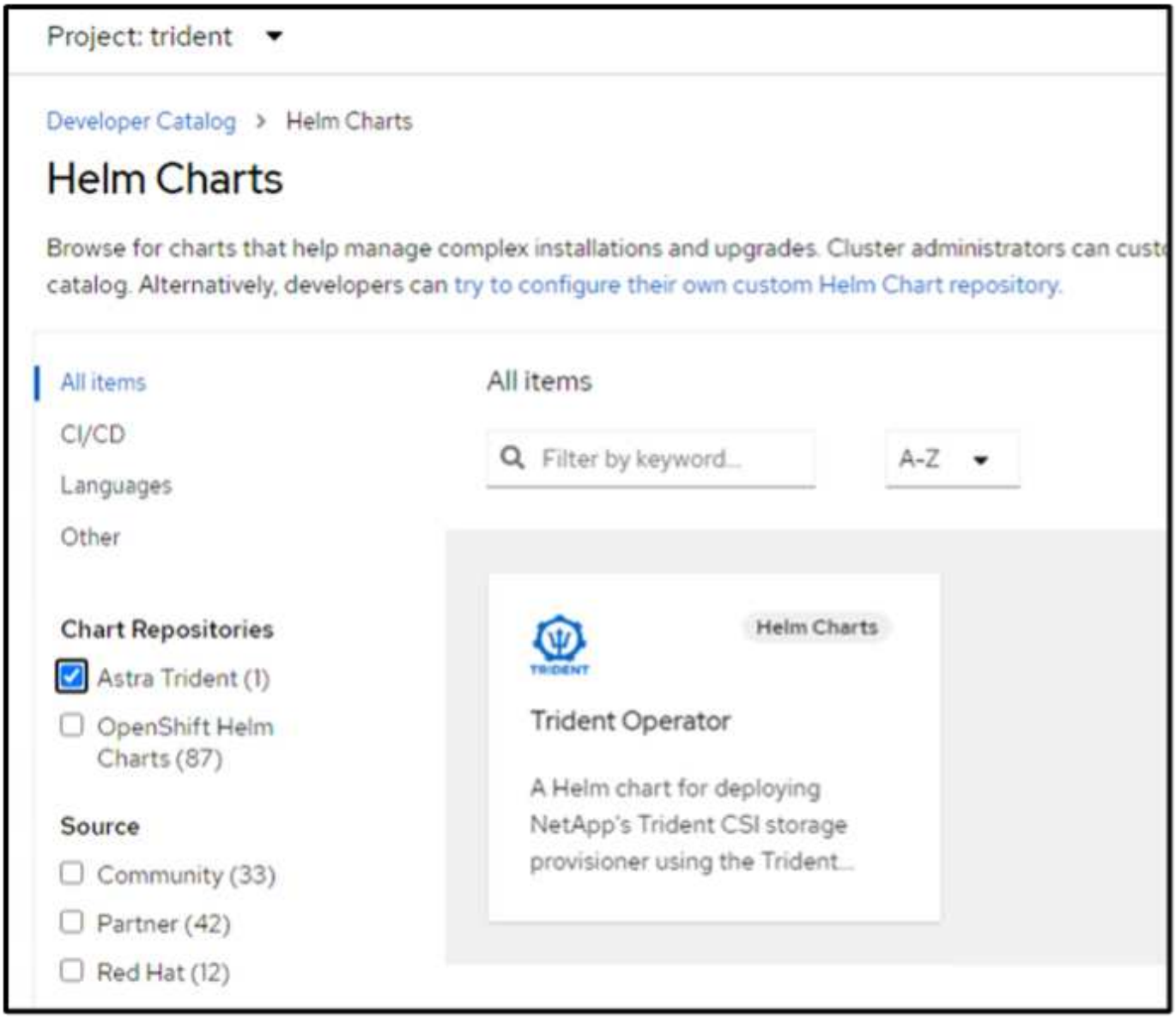
NetApp Astra Trident

A description for the Helm Chart repository.

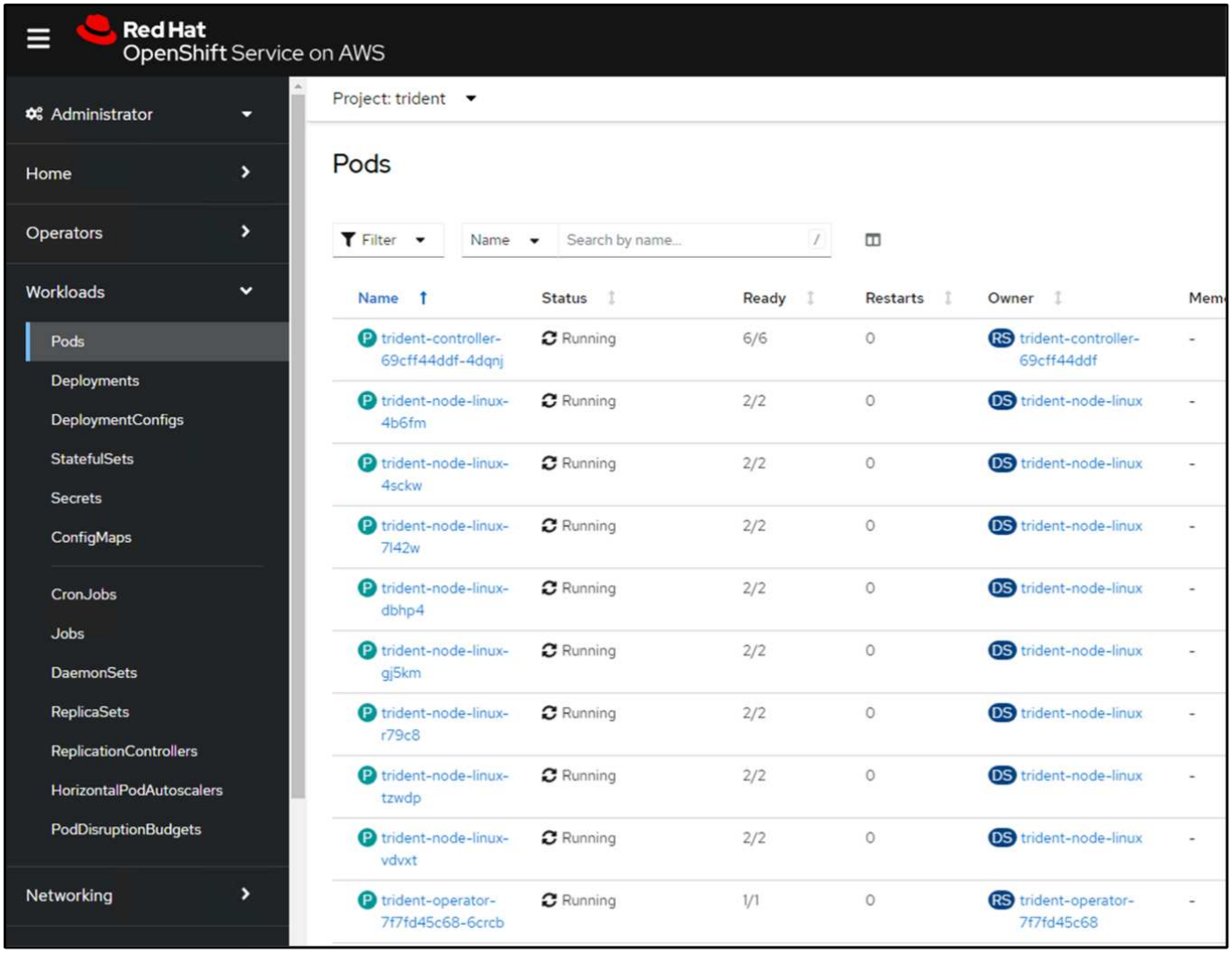
[ ] Disable usage of the repo in the developer catalog.

URL *

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



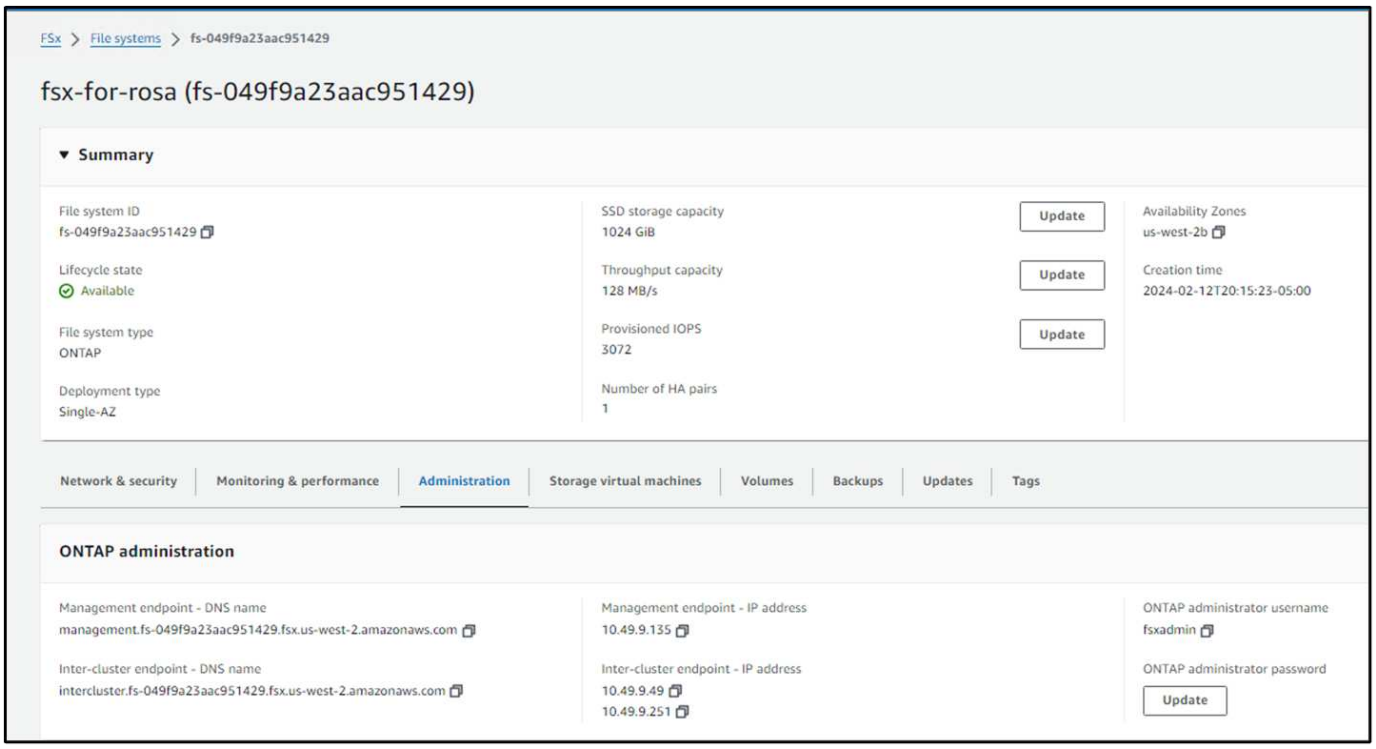
Überprüfen Sie, ob alle Stativpods ausgeführt werden, indem Sie zur Administratoransicht auf der Konsole zurückkehren und Pods im Dreizack-Projekt auswählen.



4. Erstellen Sie mit der Trident CSI-provisionierung eine Back-End-, Storage-Klasse- und Snapshot-Klassenkonfiguration

Verwenden Sie die unten abgebildeten yaml-Dateien, um ein dreigespanntes Backend-Objekt, ein Storage-Klasse-Objekt und das Volumesnapshot-Objekt zu erstellen. Stellen Sie sicher, dass Sie die Anmeldeinformationen für Ihr von Ihnen erstelltes Amazon FSX for NetApp ONTAP-Dateisystem, die Verwaltungs-LIF und den vserver-Namen Ihres Dateisystems in der Konfiguration yaml für das Backend angeben. Um diese Details anzuzeigen, wählen Sie in der AWS-Konsole für Amazon FSX das Dateisystem aus, und wechseln Sie zur Registerkarte Administration. Klicken Sie außerdem auf Aktualisieren, um das Kennwort für das festzulegen `fsxadmin` Benutzer:

> Sie können die Objekte über die Befehlszeile erstellen oder mit den yaml-Dateien von der Hybrid Cloud-Konsole aus erstellen.



Trident Back-End-Konfiguration

```yaml
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
```

Storage-Klasse

```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
```

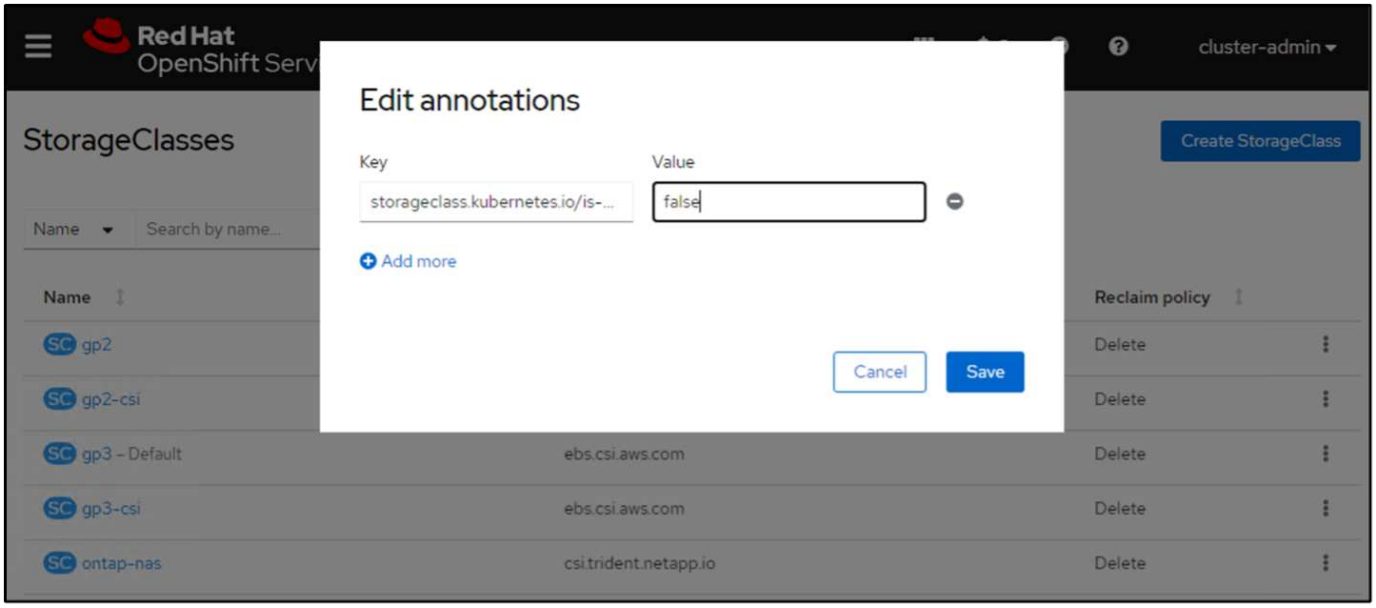
Snapshot-Klasse

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

Stellen Sie sicher, dass die Objekte von Backend, Storage-Klasse und Trident-snapshotclass mit den unten gezeigten Befehlen erstellt werden.

```
[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 ist eine wichtige Änderung erforderlich, ontap-nas statt gp3 als Standard-Storage-Klasse einzustellen, damit die später zu implementierende postgresql-Applikation die Standard-Storage-Klasse verwenden kann. Wählen Sie in der OpenShift-Konsole Ihres Clusters unter Storage StorageClasses aus. Bearbeiten Sie die Annotation der aktuellen Standardklasse mit „false“ und fügen Sie die Annotation storageclass.kubernetes.io/is-default-class für die ontap-nas Storage-Klasse auf „true“ ein.



5. Implementieren Sie eine postgresql-Anwendung auf dem Cluster

Sie können die Anwendung über die Befehlszeile wie folgt 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 Anwendungspads nicht ausgeführt werden, kann es aufgrund von Einschränkungen im Sicherheitskontext zu einem Fehler kommen.

```
[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 den bearbeiten `runAsUser` Und `fsGroup` Felder in `statefuleset.apps/postgresql` Objekt mit der UID, die sich in der Ausgabe des befindet `oc get project` 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 von Amazon FSX für NetApp ONTAP-Storage 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]$
```

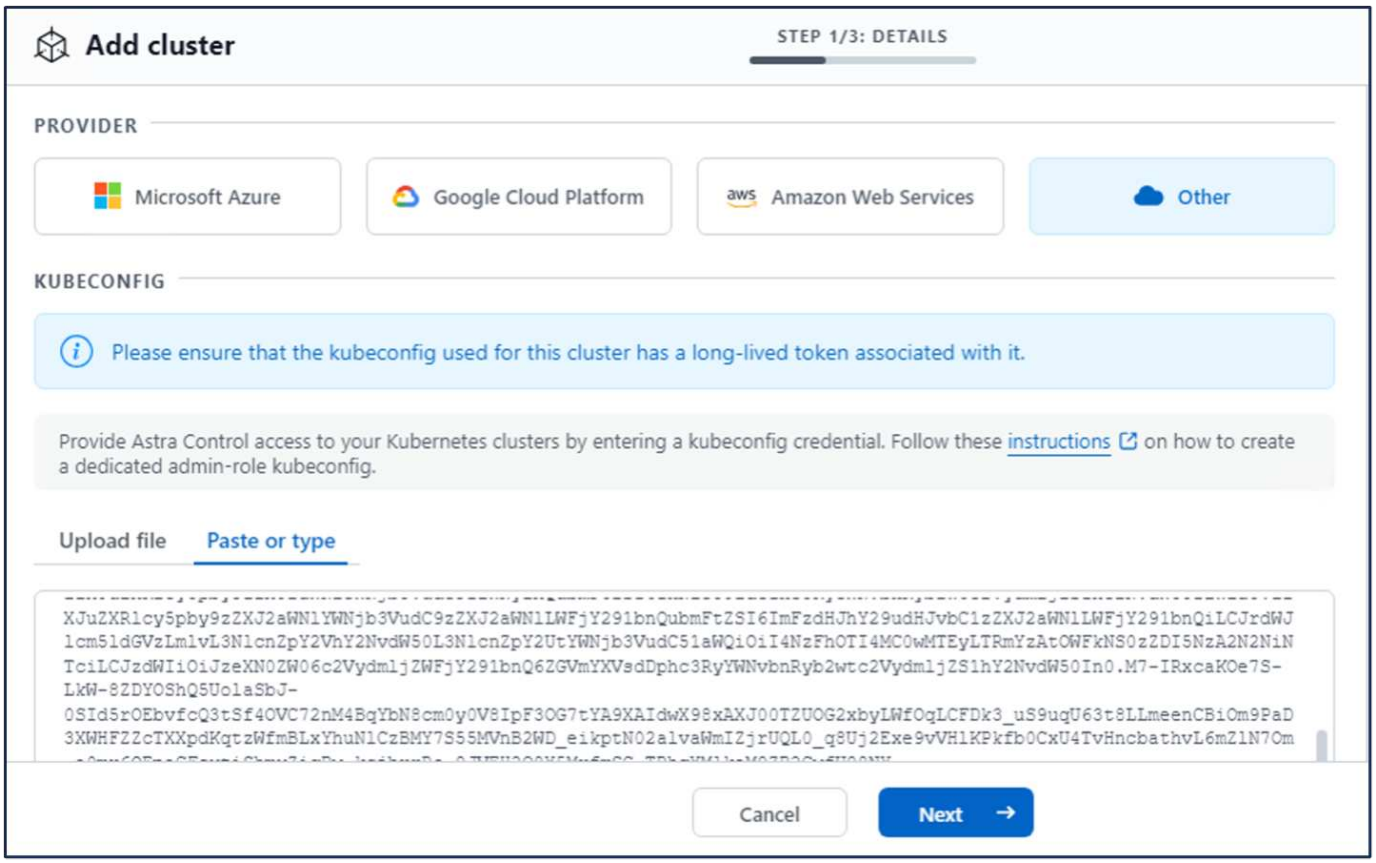
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

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

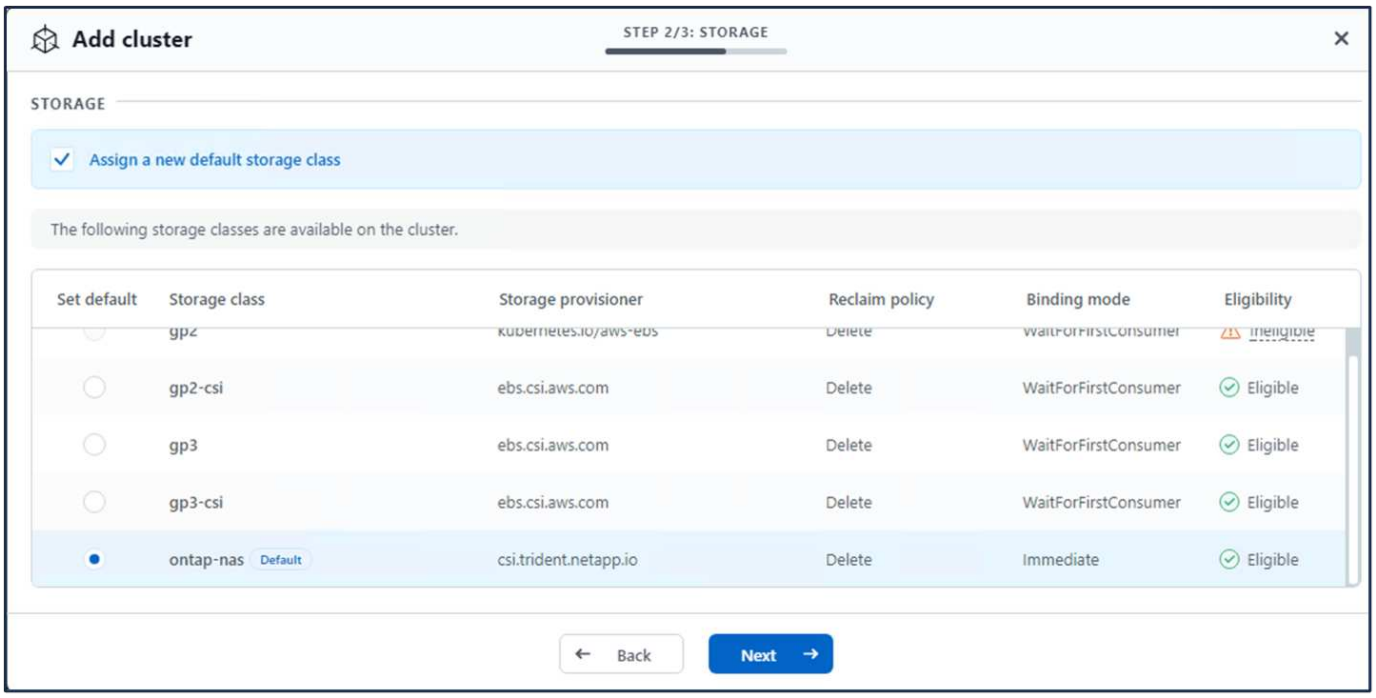
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1 --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 Cluster aus, und klicken Sie auf Hinzufügen. Wählen Sie andere aus, und laden Sie die Datei kubeconfig hoch oder fügen Sie sie ein.



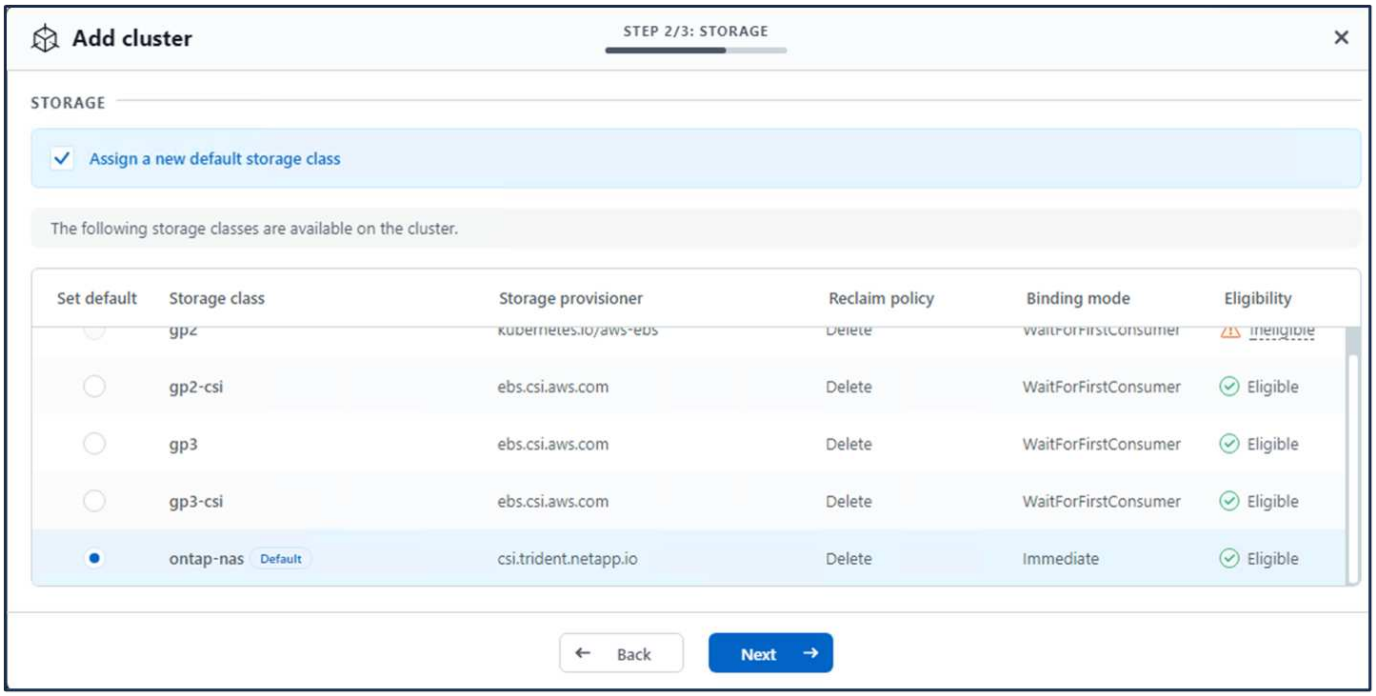
Klicken Sie auf **Weiter** und wählen Sie ontap-nas als Standard-Storage-Klasse für ACS aus. Klicken Sie auf **Weiter**, überprüfen Sie die Details und **Hinzufügen** den Cluster.



8. Definieren Sie die Anwendung in ACS

Definieren Sie die postgresql-Anwendung in ACS. Wählen Sie auf der Landing Page **Applications**, **define** aus und geben Sie die entsprechenden Details ein. Klicken Sie ein paar Mal auf **Weiter**, überprüfen Sie die Details

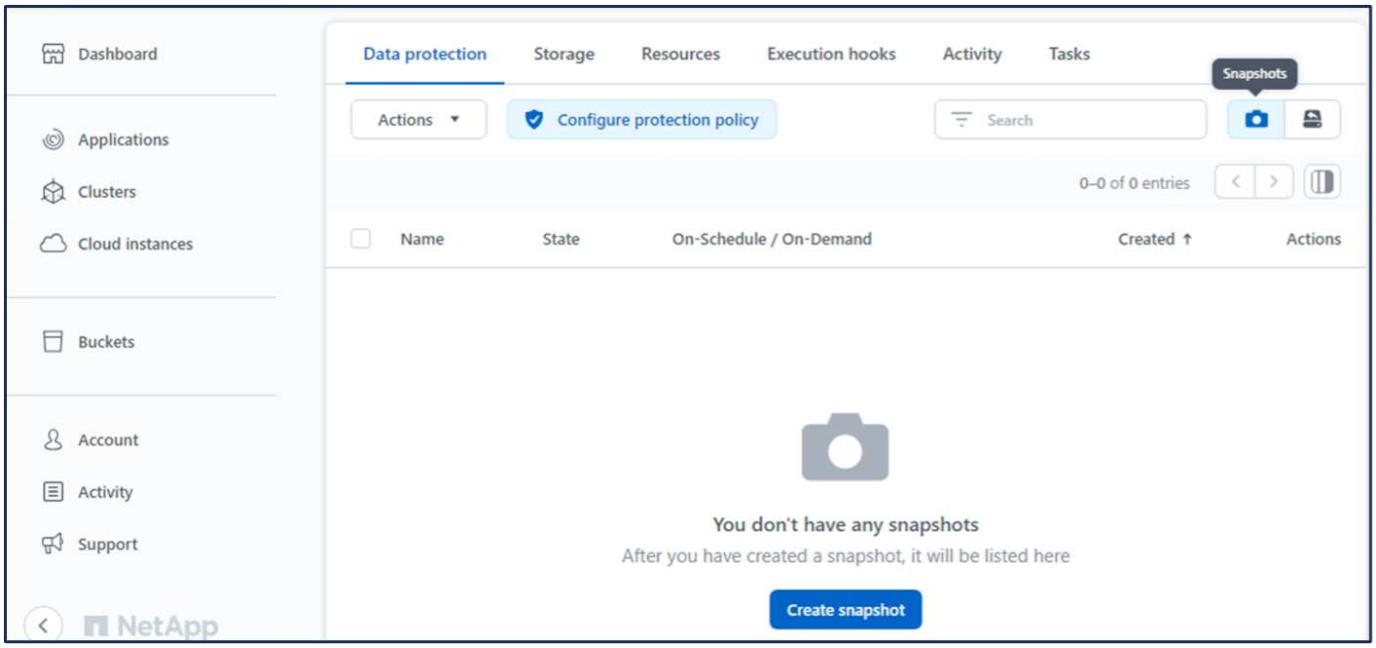
und klicken Sie auf **Definieren**. Die Anwendung wird zu ACS hinzugefügt.

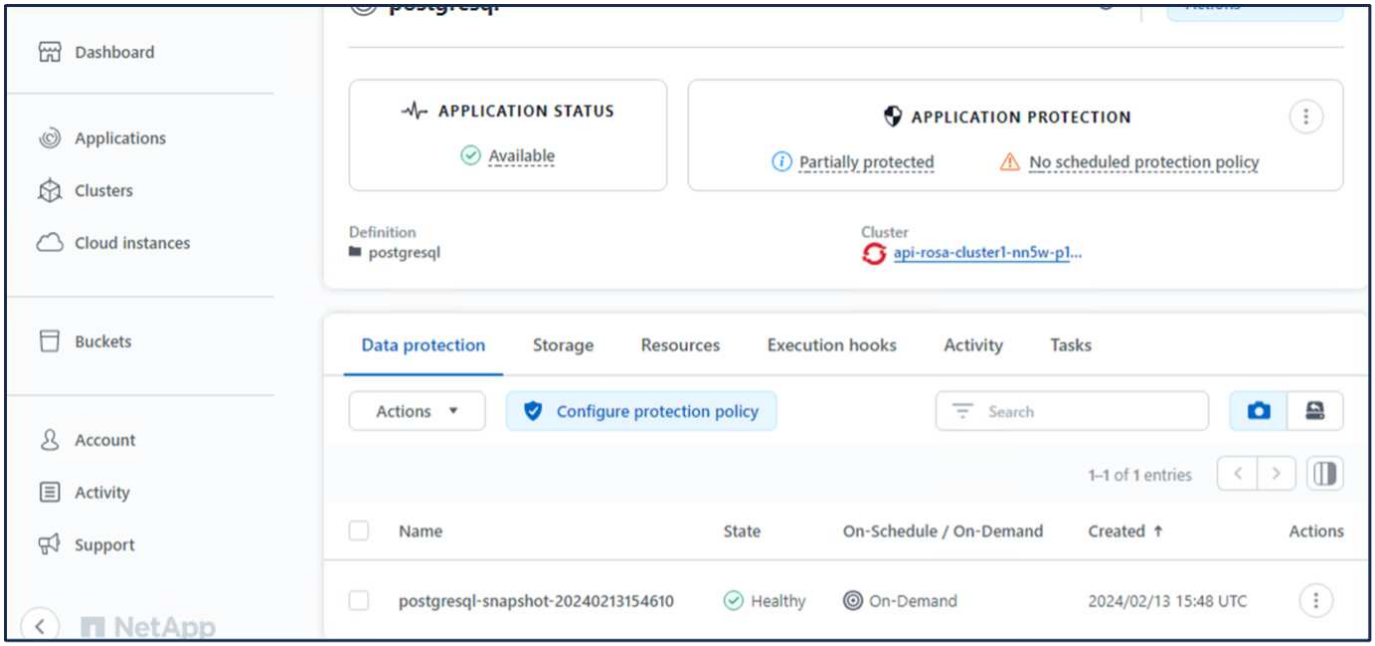


9. Erstellen Sie einen Snapshot mit ACS

Es gibt viele Möglichkeiten, einen Snapshot in ACS zu erstellen. Sie können die Anwendung auswählen und einen Snapshot auf der Seite erstellen, auf der die Details der Anwendung angezeigt werden. 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 **Create Snapshot** klicken, einen Namen angeben, die Details überprüfen und auf **Snapshot** klicken. Nach Abschluss des Vorgangs ändert sich der Snapshot-Status in „funktionstüchtiger Zustand“.





10. Löschen Sie die Datenbank in der postgresql-Anwendung

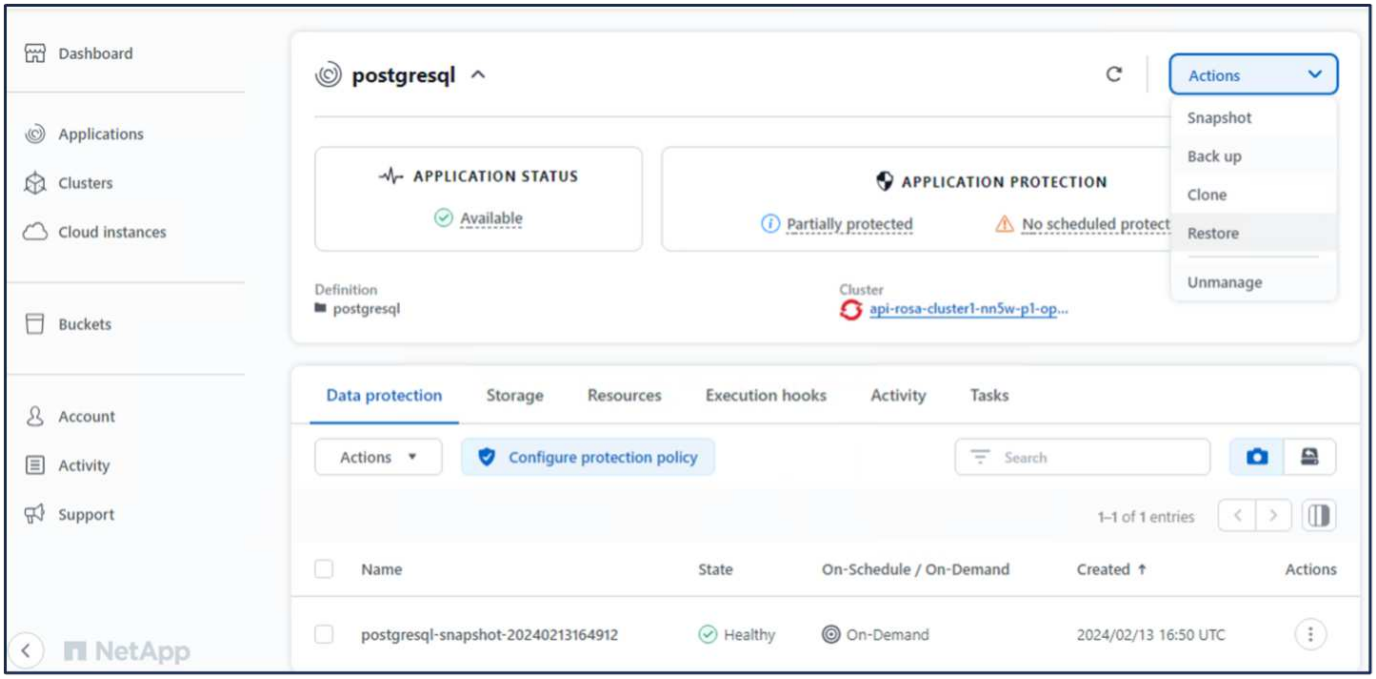
Melden Sie sich wieder bei postgresql an, Listen Sie die verfügbaren Datenbanken auf, löschen Sie die zuvor erstellte Datenbank und führen Sie sie erneut auf, um sicherzustellen, dass die Datenbank gelöscht wurde.

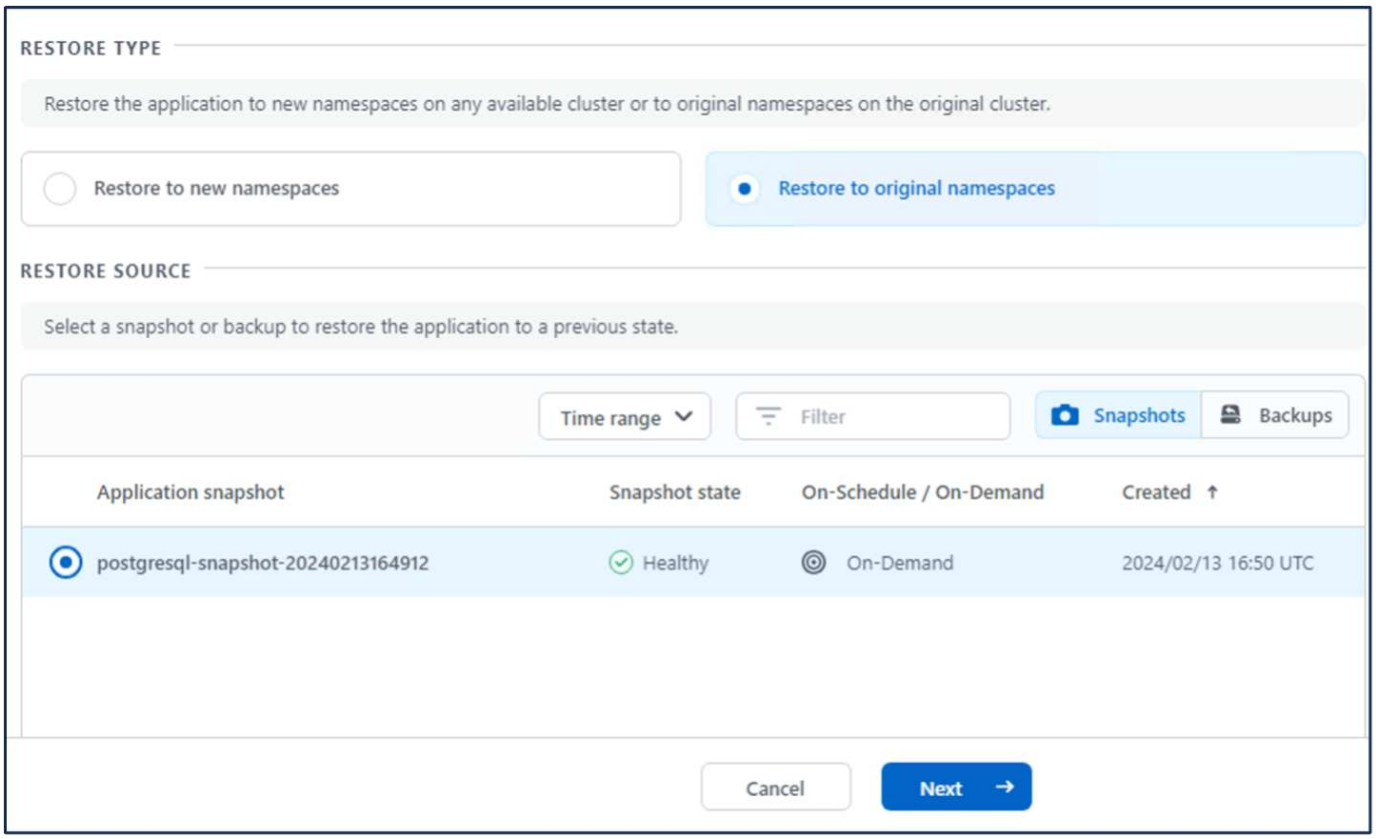
```
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------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

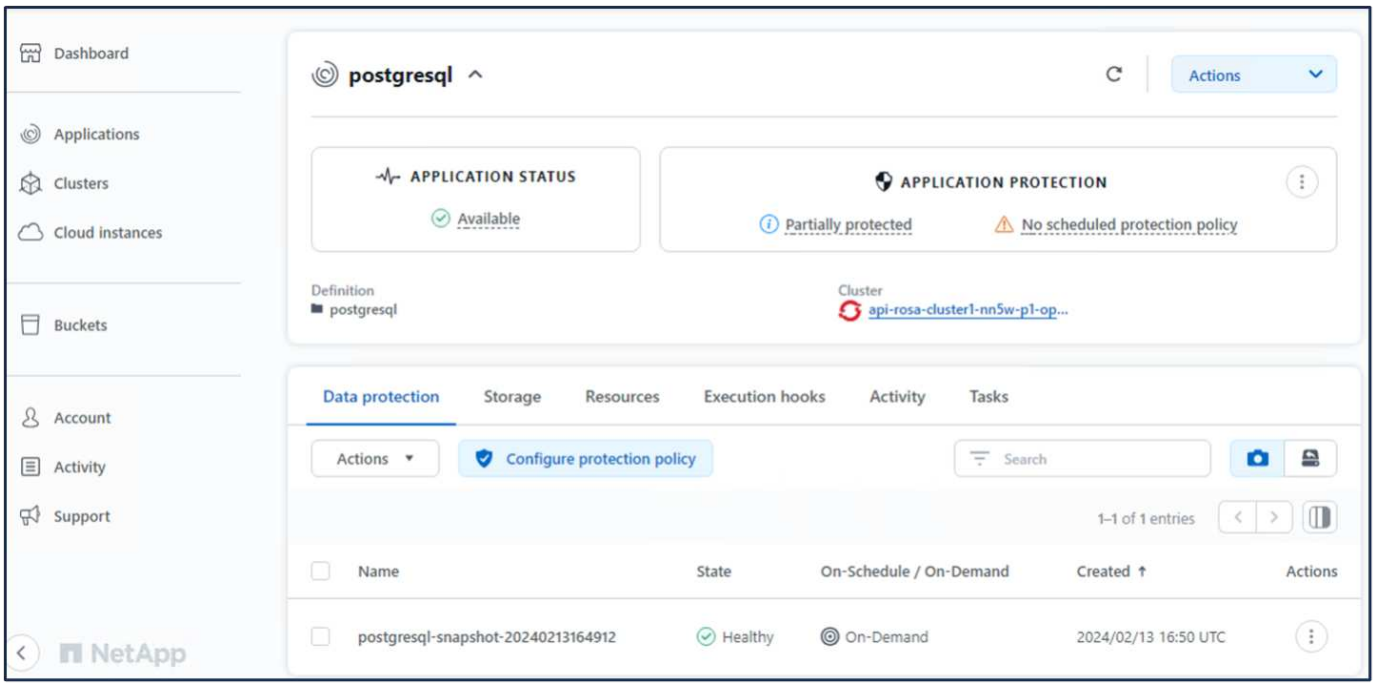
11. Wiederherstellen von einem Snapshot mit ACS

Um die Anwendung von einem Snapshot wiederherzustellen, gehen Sie zur ACS-UI-Landing Page, wählen Sie die Anwendung aus und wählen Sie Wiederherstellen. Sie müssen einen Snapshot oder ein Backup

auswählen, von dem aus wiederhergestellt werden soll. (In der Regel würden auf Basis einer von Ihnen konfigurierten Richtlinie mehrere erstellt werden.) Treffen Sie in den nächsten Bildschirmanzeigen die richtige Auswahl und klicken Sie dann auf **Wiederherstellen**. Der Anwendungsstatus wechselt von Wiederherstellen zu verfügbar, nachdem er aus dem Snapshot wiederhergestellt wurde.







12. Überprüfen Sie, ob Ihre App aus der Momentaufnahme wiederhergestellt wurde

Melden Sie sich beim postgresql-Client an und Sie sollten nun die Tabelle und den Datensatz in der Tabelle sehen, die Sie zuvor hatten. Das ist alles. Durch Klicken auf eine Schaltfläche wurde Ihre Anwendung in einen früheren Zustand zurückgesetzt. 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 unter Verwendung von FSX for NetApp ONTAP für persistenten Storage angezeigt.

Datenmigration

Red hat OpenShift-Service auf AWS sowie FSX for NetApp ONTAP (FSxN) sind Teil ihres Service-Portfolios von AWS. FSxN ist mit Single AZ- oder Multi-AZ-Optionen verfügbar. Die Multi-Az-Option bietet Datenschutz bei Ausfall einer Verfügbarkeitszone. FSxN kann in Astra Trident integriert werden, um persistenten Storage für Applikationen auf ROSA Clustern bereitzustellen.

Integration von FSxN mit Trident mit Helm Chart

ROSA Cluster Integration mit Amazon FSX for ONTAP

Die Migration von Container-Applikationen umfasst Folgendes:

- Persistente Volumes: Dies ist mit BlueXP möglich. Eine weitere Option ist der Einsatz von Astra Control Center für die Migration von Container-Applikationen von On-Premises- in die Cloud-Umgebung. Automatisierung kann für den gleichen Zweck eingesetzt werden.
- Applikations-Metadaten: Dies kann mithilfe von OpenShift GitOps (Argo CD) erreicht werden.

Failover und Failback von Anwendungen auf ROSA-Cluster mit FSxN für persistenten Speicher

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

Failover und Failback von Anwendungen auf ROSA Cluster

Datensicherungs- und Migrationslösung für OpenShift-Container-Workloads