공급자 관리 구성 요소를 지원하는 하이브리드 클라우드

NetApp Solutions

NetApp
April 20, 2024

목차

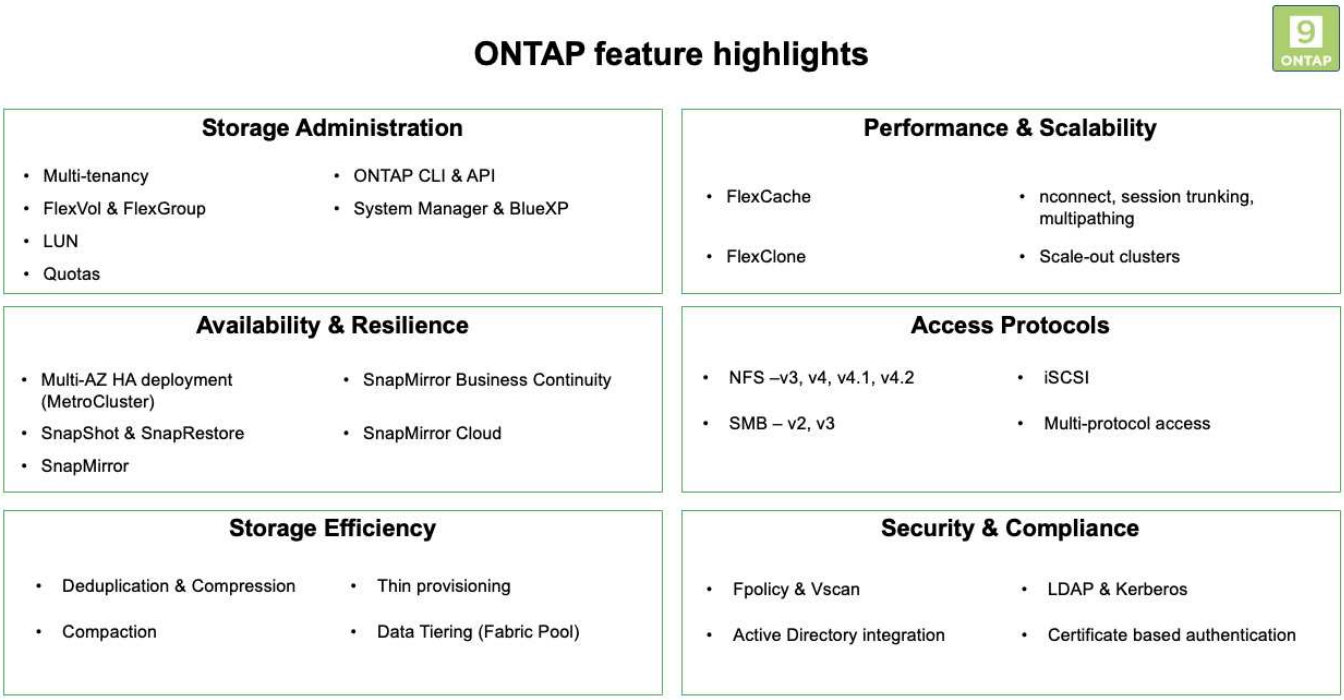
Red Hat OpenShift Container 워크로드를 위한 NetApp 하이브리드 멀티 클라우드 솔루션

개요

NetApp은 기존 엔터프라이즈 애플리케이션을 현대화하고 Kubernetes를 기반으로 구축된 컨테이너 및 오케스트레이션 플랫폼을 사용하여 새로운 애플리케이션을 구축하는 고객이 크게 증가하고 있습니다. Red Hat OpenShift Container Platform은 많은 고객이 채택한 한 가지 예입니다.

점점 더 많은 고객이 기업 내에 컨테이너를 채택하기 시작함에 따라 NetApp은 상태 저장 애플리케이션의 영구 스토리지 요구사항과 데이터 보호, 데이터 보안, 데이터 마이그레이션과 같은 기존의 데이터 관리 요구사항을 충족할 수 있는 완벽한 위치를 선점하고 있습니다. 그러나 이러한 요구 사항은 서로 다른 전략, 도구 및 방법을 사용하여 충족됩니다.

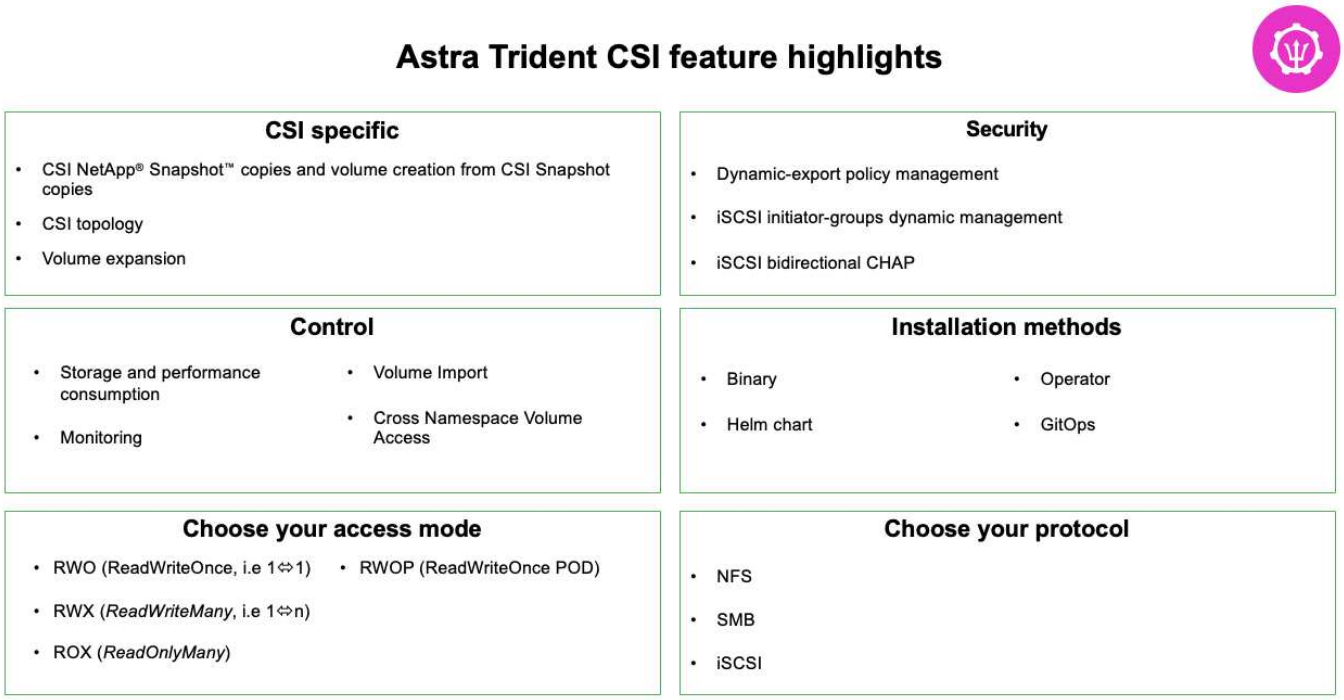
- NetApp ONTAP** 아래에 나열된 스토리지 옵션을 사용하여 컨테이너 및 Kubernetes 구축을 위한 보안, 데이터 보호, 안정성 및 유연성을 확보할 수 있습니다.
  - 사내 자가 관리형 스토리지:
- NetApp 패브릭 연결 스토리지(FAS), NetApp All Flash FAS 어레이(AFF), NetApp All SAN 어레이(ASA) 및 ONTAP Select
  - 온프레미스에서 공급자 관리 스토리지:
- NetApp Keystone, STaaS(서비스형 스토리지) 제공
  - 클라우드에서 자가 관리 스토리지:
- NetApp Cloud Volumes ONTAP(CVO)은 하이퍼스케일러에 자가 관리하는 스토리지를 제공합니다
  - 클라우드 내 공급자 관리 스토리지:
- Cloud Volumes Service for Google Cloud(CVS), Azure NetApp Files(ANF), Amazon FSx for NetApp ONTAP는 하이퍼스케일러에 완전 관리형 스토리지를 제공합니다



- NetApp BlueXP** - 단일 제어 플레인/인터페이스에서 모든 스토리지 및 데이터 자산을 관리할 수 있습니다.

BlueXP를 사용하여 클라우드 스토리지(예: Cloud Volumes ONTAP 및 Azure NetApp Files)를 생성 및 관리하고, 데이터를 이동, 보호 및 분석하며, 많은 사내 및 에지 스토리지 장치를 제어할 수 있습니다.

- NetApp Astra Trident**는 CSI 규정 준수 스토리지 오케스트레이터로서, 위에서 언급한 다양한 NetApp 스토리지 옵션을 통해 영구 스토리지를 빠르고 쉽게 사용할 수 있습니다. NetApp에서 관리 및 지원하는 오픈 소스 소프트웨어입니다.



비즈니스 크리티컬 컨테이너 워크로드에는 영구 볼륨 이상의 용량이 필요합니다. 이들의 데이터 관리 요구사항에 따라 애플리케이션 Kubernetes 객체의 보호 및 마이그레이션이 필요합니다.

> 애플리케이션 데이터에는 사용자 데이터 외에도 Kubernetes 객체가 포함됩니다. 몇 가지 예는 다음과 같습니다. POD 사양, PVC, 구축, 서비스 맞춤형 구성 개체(예: 구성 맵 및 암호), 스냅샷 복사본, 백업, CRS, CRD와 같은 클론 맞춤형 리소스 등의 영구 데이터)가 있습니다

- NetApp Astra Control**, 완전 관리형 및 자가 관리 소프트웨어로 모두 사용 가능하며, 강력한 애플리케이션 데이터 관리를 위한 오케스트레이션을 제공합니다. 을 참조하십시오 "Astra 문서" Astra 제품군에 대한 자세한 내용은

이 참조 문서는 NetApp Astra Control Center를 사용하여 RedHat OpenShift 컨테이너 플랫폼에 배포된 컨테이너 기반 애플리케이션의 마이그레이션 및 보호를 검증합니다. 또한 이 솔루션은 컨테이너 플랫폼 관리를 위한 Red Hat Advanced Cluster Management(ACM)의 배포 및 사용에 대한 자세한 정보를 제공합니다. 또한, Astra Trident CSI 프로비저닝을 사용하여 NetApp 스토리지를 Red Hat OpenShift 컨테이너 플랫폼과 통합하기 위한 세부 정보도 제공합니다. Astra Control Center는 허브 클러스터에 구축되며 컨테이너 애플리케이션 및 영구 스토리지 라이프사이클을 관리하는 데 사용됩니다. 마지막으로, NetApp FSx for NetApp ONTAP(FSxN)를 영구 스토리지로 사용하는 AWS(Rosa)의 관리되는 Red Hat OpenShift 클러스터에서 복제, 페일오버 및 컨테이너 워크로드에 대한 페일백용 솔루션을 제공합니다.

AWS 기반의 관리되는 Red Hat OpenShift Container 플랫폼 워크로드를 지원하는 NetApp 솔루션

AWS 기반의 관리되는 Red Hat OpenShift Container 플랫폼 워크로드를 지원하는 NetApp 솔루션

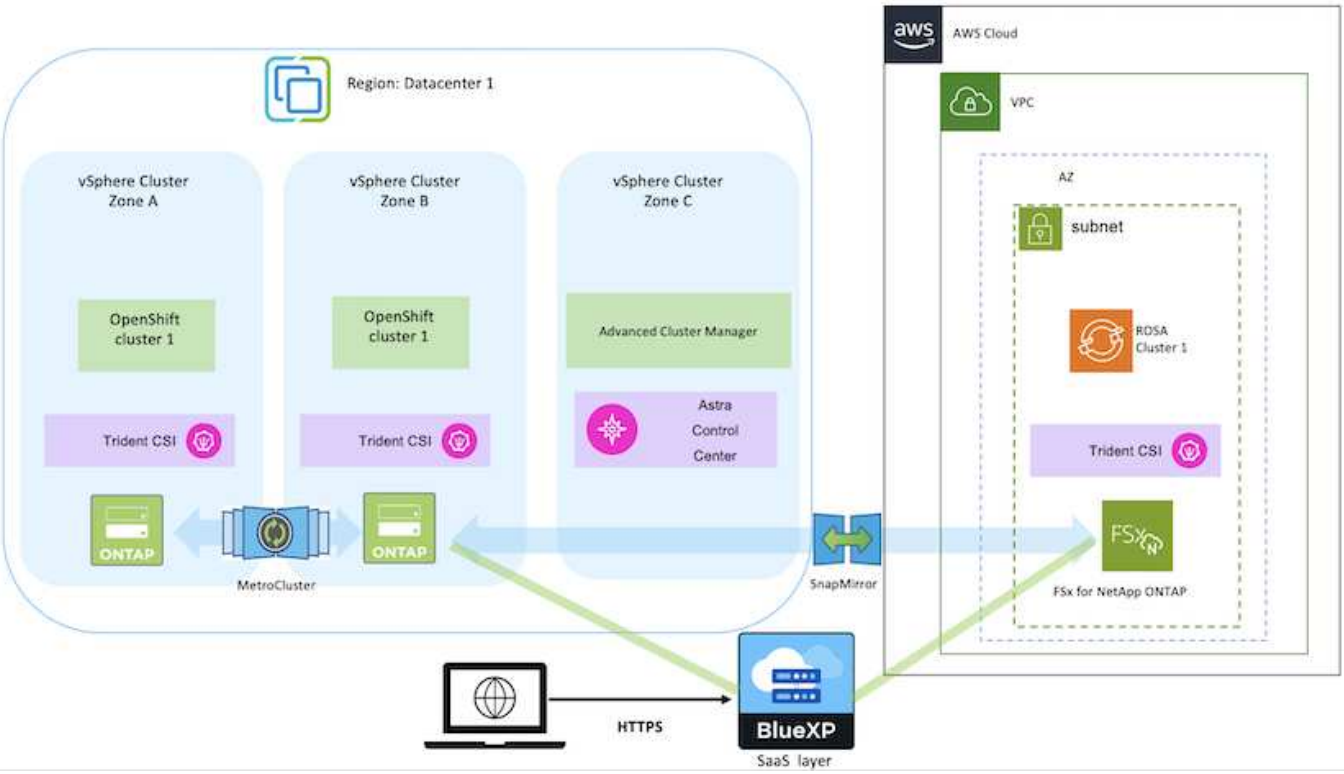
고객은 일부 특정 워크로드 또는 모든 워크로드를 데이터 센터에서 클라우드로 이동할 준비가 되었을 때 "클라우드에서 탄생됨" 또는 현대화 과정에서 일부가 될 수 있습니다. 고객은 클라우드에서 공급자 관리 OpenShift 컨테이너와 공급자 관리 NetApp 스토리지를 사용하여 워크로드를 실행할 수 있습니다. 컨테이너 워크로드를 위한 성공적인 프로덕션 준비 환경을 위해 클라우드에서 관리되는 Red Hat OpenShift 컨테이너 클러스터(Rosa)를 계획하고 배포해야 합니다. AWS 클라우드에 있는 고객은 스토리지 필요에 따라 NetApp ONTAP용 FSx를 구축할 수 있습니다.

NetApp ONTAP용 FSX는 AWS의 컨테이너 구축을 위한 데이터 보호, 안정성 및 유연성을 제공합니다. Astra Trident는 고객의 상태 저장 애플리케이션에 영구 FSxN 스토리지를 사용하는 동적 스토리지 프로비저닝을 수행합니다.

여러 가용성 영역에 컨트롤 플레인 노드가 분산된 상태에서 HA 모드로 Rosa를 구축할 수 있으므로, FSx ONTAP는 고가용성을 제공하고 AZ 장애로부터 보호하는 Multi-AZ 옵션을 통해 구축할 수도 있습니다.

> 파일 시스템의 AZ(Preferred Availability Zone)에서 Amazon FSx 파일 시스템에 액세스할 때 데이터 전송 비용이 발생하지 않습니다. 가격에 대한 자세한 내용은 을 참조하십시오 "여기".

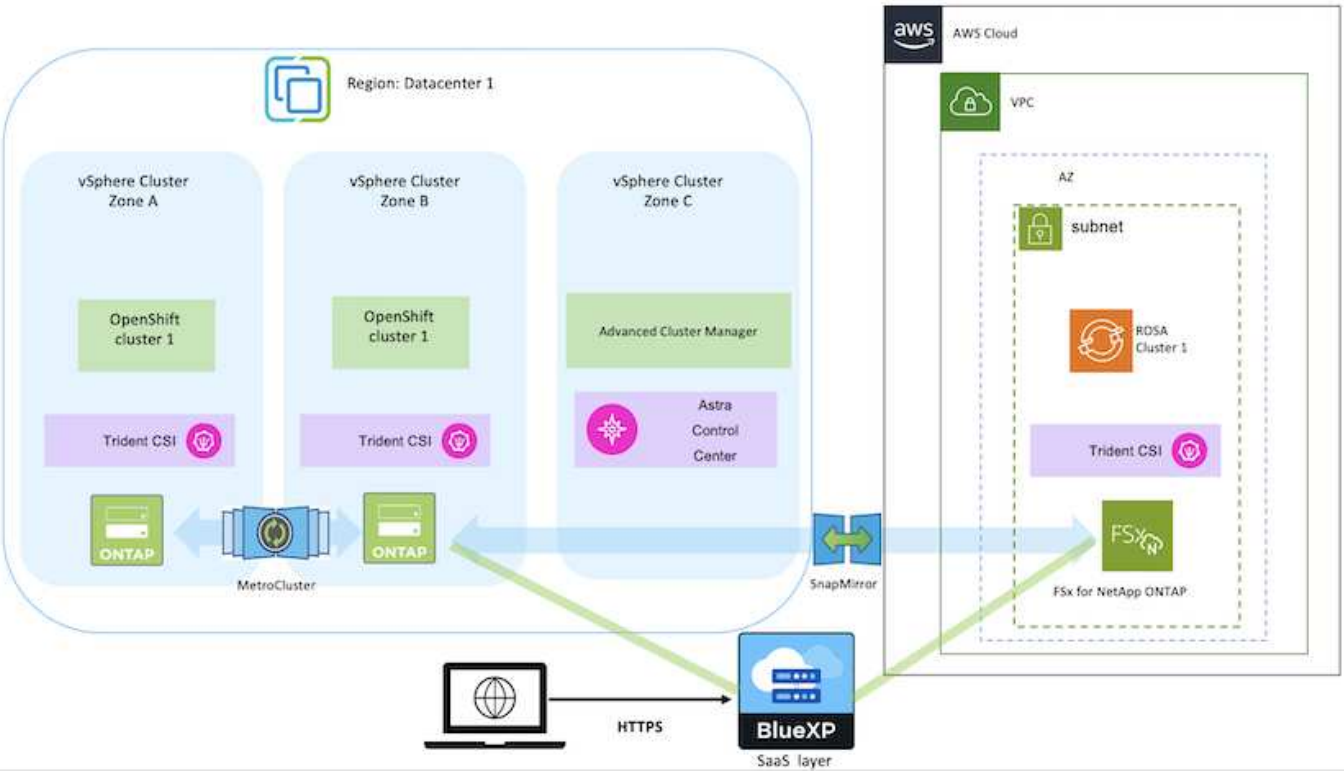
OpenShift Container 워크로드를 위한 데이터 보호 및 마이그레이션 솔루션



AWS에서 관리되는 Red Hat OpenShift Container 플랫폼을 배포하고 구성합니다

이 섹션에서는 AWS(Rosa)에서 관리되는 Red Hat OpenShift 클러스터를 설정하는 고급 워크플로우를 설명합니다. 또한 영구 볼륨을 제공하기 위해 Astra Trident가 NetApp FSx for NetApp ONTAP(FSxN)를 스토리지 백엔드로 사용하는 것을 보여 줍니다. BlueXP를 사용하는 AWS에서 FSxN을 배포하는 방법에 대한 자세한 정보가 제공됩니다. 또한 Rosa 클러스터의 상태 저장 애플리케이션에 대한 데이터 보호 및 마이그레이션 작업을 수행하기 위해 BlueXP 및 OpenShift GitOps(Argo CD)를 사용하는 방법에 대한 세부 정보도 제공됩니다.

다음은 AWS에 배포되고 FSxN을 백엔드 스토리지로 사용하는 Rosa 클러스터를 보여 주는 다이어그램입니다.



이 솔루션은 AWS의 두 대의 VPC에서 두 개의 Rosa 클러스터를 사용하여 검증되었습니다. 각 Rosa 클러스터는 Astra Trident를 사용하여 FSxN과 통합되었습니다. AWS에서 Rosa 클러스터와 FSxN을 구축하는 방법은 여러 가지가 있습니다. 설정에 대한 이 고급 설명은 사용된 특정 방법에 대한 설명서 링크를 제공합니다. 에 제공된 관련 링크에서 다른 방법을 참조할 수 있습니다 "리소스 섹션을 참조하십시오".

설치 프로세스는 다음 단계로 나눌 수 있습니다.

Rosa 클러스터를 설치합니다

- 2개의 VPC를 생성하고 VPC 간 VPC 피어링 연결을 설정합니다.
- 을 참조하십시오 "여기" Rosa 클러스터를 설치하는 지침은 를 참조하십시오.

FSxN을 설치합니다

- BlueXP에서 VPC에 FSxN을 설치합니다. 을 참조하십시오 "여기" BlueXP 계정 생성 및 시작 을 참조하십시오 "여기" FSxN 설치용. 을 참조하십시오 "여기" FSxN을 관리하기 위해 AWS에 커넥터를 생성하는 데 사용됩니다.
- AWS를 사용하여 FSxN을 구축합니다. 을 참조하십시오 "여기" AWS 콘솔을 사용하여 구축

Rosa 클러스터에 **Trident** 설치(제어 차트 사용)

- 제어 차트를 사용하여 Rosa 클러스터에 Trident를 설치합니다. 제어 차트 URL: https://netapp.github.io/trident-helm-chart

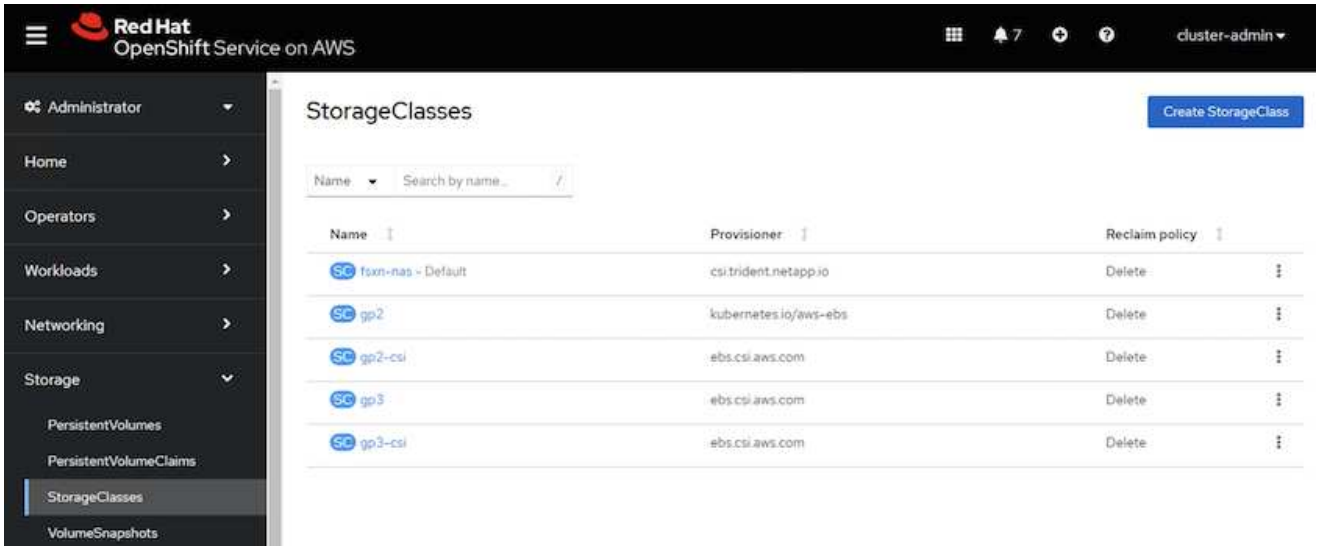
FSxN과 Astra Trident for Rosa 클러스터의 통합

OpenShift GitOps를 사용하면 ApplicationSet을 사용하여 ArgoCD에 등록될 때 모든 관리 클러스터에 Astra Trident CSI를 배포할 수 있습니다.

```
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
```

Trident(FSxN용)를 사용하여 백엔드 및 스토리지 클래스 생성

- 을 참조하십시오 "여기" 백엔드 및 스토리지 클래스 생성에 대한 자세한 내용은 을 참조하십시오.
- OpenShift Console에서 Trident CSI로 FsxN에 대해 생성한 스토리지 클래스를 기본값으로 설정합니다. 아래 스크린샷을 참조하십시오.



OpenShift GitOps(Argo CD)를 사용하여 애플리케이션 배포

- 클러스터에 OpenShift GitOps 운영자를 설치합니다. 지침을 참조하십시오 "여기".
- 클러스터에 대한 새 Argo CD 인스턴스를 설정합니다. 지침을 참조하십시오 "여기".

Argo CD 콘솔을 열고 앱을 배포합니다. 예를 들어, Argo CD와 H제어 차트를 사용하여 Jenkins 앱을 배포할 수 있습니다. 응용 프로그램을 생성할 때 다음과 같은 세부 정보가 제공됩니다. Project: 기본 클러스터: https://kubernetes.default.svc네임스페이스: Jenkins 제어 차트의 URL: https://charts.bitnami.com/bitnami

Helm Parameters:global.storageClass:fsxn-nas

데이터 보호

이 페이지에는 Astra Control Service를 사용하는 AWS(ROSA) 관리형 Red Hat OpenShift 클러스터에 대한 데이터 보호 옵션이 나와 있습니다. Astra Control Service(ACS)는 사용이 간편한 그래픽 사용자 인터페이스를 제공하여 클러스터를 추가하고, 클러스터에서 실행되는 애플리케이션을 정의하고, 애플리케이션 인식 데이터 관리 활동을 수행할 수 있습니다. ACS 기능은 워크플로우 자동화를 지원하는 API를 사용하여 액세스할 수도 있습니다.

Astra Control(ACS 또는 ACC)은 NetApp Astra Trident입니다. Astra Trident는 Red Hat OpenShift, EKS, AKS, SUSE Rancher, Anthos 등과 같은 다양한 유형의 Kubernetes 클러스터를 통합합니다. FAS/AFF, ONTAP Select, CVO, Google Cloud Volumes Service, Azure NetApp Files 및 Amazon FSx for NetApp ONTAP 같은 다양한 유형의 NetApp ONTAP 스토리지를 활용할 수 있습니다.

이 섹션에서는 ACS를 사용하는 다음 데이터 보호 옵션에 대해 자세히 설명합니다.

- 한 지역에서 실행 중인 Rosa 애플리케이션의 백업 및 복원과 다른 지역으로 복원한 비디오를 보여 줍니다.
- Rosa 애플리케이션의 스냅샷 및 복원을 보여주는 비디오
- Rosa 클러스터, Amazon FSx for NetApp ONTAP 설치, NetApp Astra Trident를 사용하여 스토리지 백엔드와 통합, Rosa 클러스터에 PostgreSQL 애플리케이션 설치, ACS를 사용하여 애플리케이션 스냅샷을 생성하고 애플리케이션을 복원하는 방법에 대한 단계별 세부 정보입니다.
- ACS를 사용하는 FSx for ONTAP가 포함된 ROSA 클러스터의 MySQL 애플리케이션에 대한 스냅샷을 생성하고 복원하는 방법에 대한 단계별 세부 정보를 보여주는 블로그

백업에서 백업/복원

다음 비디오에서는 한 지역에서 실행되고 다른 지역으로 복원되는 Rosa 응용 프로그램의 백업을 보여 줍니다.

AWS 기반 FSx NetApp ONTAP for Red Hat OpenShift Service

스냅샷/스냅샷에서 복구

다음 비디오는 Rosa 응용 프로그램의 스냅샷 촬영 및 이후 스냅샷에서 복원하는 방법을 보여 줍니다.

Amazon FSx for NetApp ONTAP 스토리지를 사용하는 AWS(ROSA) 기반 Red Hat OpenShift Service의 애플리케이션을 위한 스냅샷/복원

블로그

- "Amazon FSx 스토리지가 탑재된 Rosa 클러스터에서 앱의 데이터를 관리하는 데 Astra Control Service를 사용합니다"

스냅샷을 생성하고 이 스냅샷에서 복구하는 단계별 세부 정보입니다

사전 요구 사항 설정

- "설치하 고 있습니다"
- "Red Hat OpenShift 계정"
- IAM 사용자 "적절한 사용 권한" Rosa 클러스터를 생성하고 액세스합니다
- "AWS CLI를 참조하십시오"
- "로사 CLI"
- "OpenShift CLI를 참조하십시오"(OC)
- VPC와 서브넷, 적절한 게이트웨이 및 라우트
- "ROSA 클러스터가 설치되었습니다" VPC로 이동합니다
- "NetApp ONTAP용 Amazon FSx" 동일한 VPC에서 생성됨
- 에서 Rosa 클러스터에 액세스합니다 "OpenShift 하이브리드 클라우드 콘솔"

다음 단계

1. admin 사용자를 생성하고 클러스터에 로그인합니다.

2. 클러스터에 대한 kubeconfig 파일을 생성합니다.
3. 클러스터에 Astra Trident를 설치합니다.
4. Trident CSI Provisioner를 사용하여 백엔드, 스토리지 클래스 및 스냅샷 클래스 구성을 생성합니다.
5. 클러스터에 PostgreSQL 애플리케이션을 구축합니다.
6. 데이터베이스를 만들고 레코드를 추가합니다.
7. 클러스터를 ACS에 추가합니다.
8. ACS에서 애플리케이션을 정의합니다.
9. ACS를 사용하여 스냅샷을 생성합니다.
10. PostgreSQL 애플리케이션에서 데이터베이스를 삭제합니다.
11. ACS를 사용하여 스냅샷에서 복원합니다.
12. 앱이 스냅숏에서 복원되었는지 확인합니다.

1. 관리자 사용자를 생성하고 클러스터에 로그인합니다

다음 명령을 사용하여 admin 사용자를 생성하여 Rosa 클러스터에 액세스합니다(설치 시 admin 사용자를 생성하지 않은 경우에만 생성 필요).

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

명령은 다음과 같은 출력을 제공합니다. 를 사용하여 클러스터에 로그인합니다 `oc login` 출력에 제공된 명령입니다.

```
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
```

> 토큰을 사용하여 클러스터에 로그인할 수도 있습니다. 클러스터 생성 시 이미 관리자 사용자를 생성한 경우 Red Hat OpenShift Hybrid Cloud 콘솔에서 관리자 자격 증명을 사용하여 클러스터에 로그인할 수 있습니다. 그런 다음, 로그인한 사용자의 이름을 표시하는 오른쪽 상단 모서리를 클릭하여 를 얻을 수 있습니다 `oc login` 명령줄에 대한 명령(토큰 로그인)입니다.

2. 클러스터에 대한 **kubeconfig** 파일을 생성합니다

절차를 따르십시오 "여기" Rosa 클러스터에 대한 kubeconfig 파일을 생성합니다. 이 kubeconfig 파일은 ACS에 클러스터를 추가할 때 나중에 사용됩니다.

3. 클러스터에 **Astra Trident**를 설치합니다

Rosa 클러스터에 Astra Trident(최신 버전)를 설치합니다. 이렇게 하려면 주어진 절차 중 하나를 따를 수 있습니다 " 여기". 클러스터 콘솔에서 Helm을 사용하여 Trident를 설치하려면 먼저 Trident라는 프로젝트를 생성합니다.

그런 다음 개발자 보기에서 Helm 차트 리포지토리를 만듭니다. URL 필드에 을 사용합니다 `'https://netapp.github.io/trident-helm-chart'`. 그런 다음 Trident 운영자에 대한 Helm 릴리즈를 작성합니다.

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

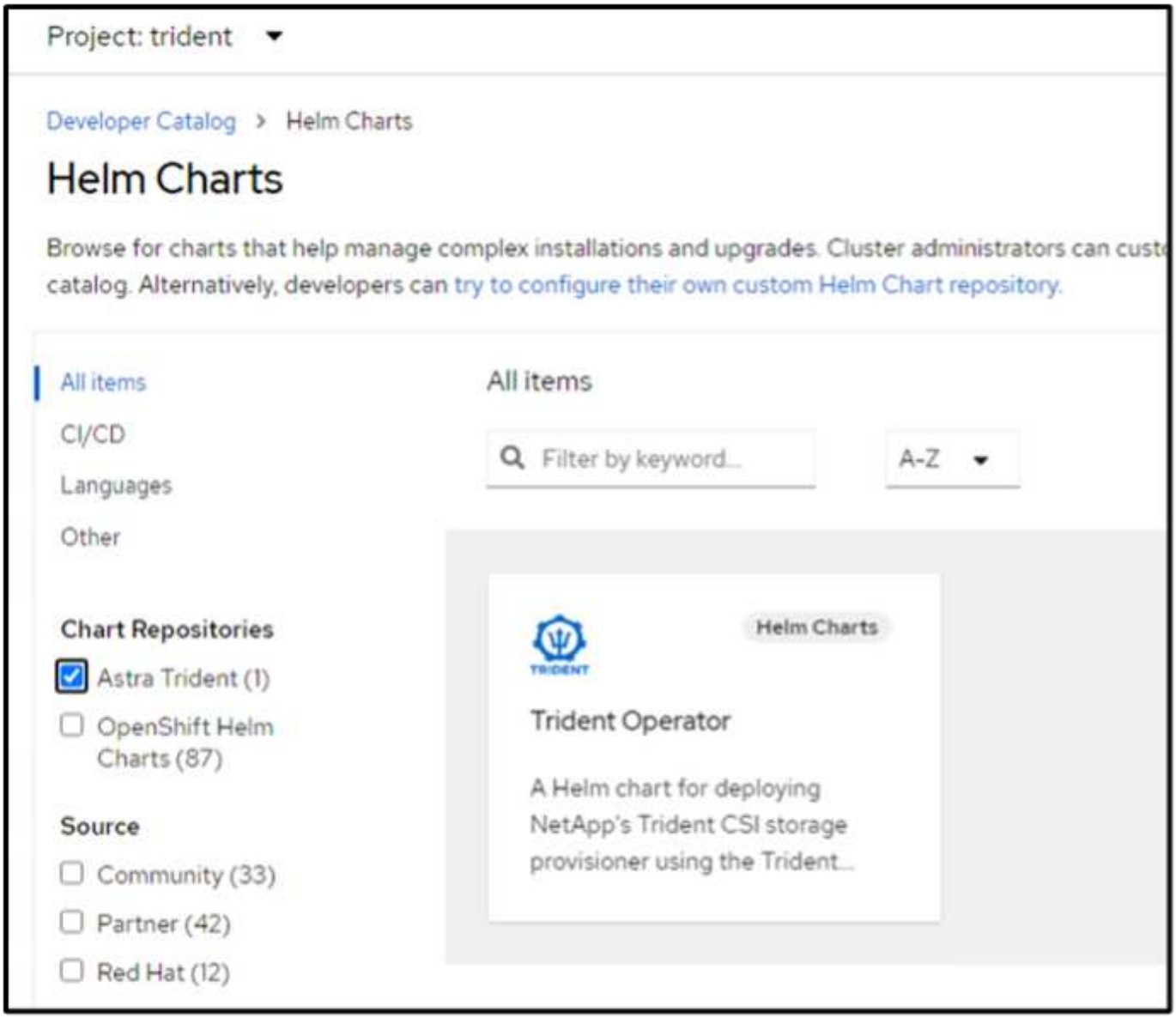
NetApp Astra Trident

A description for the Helm Chart repository.

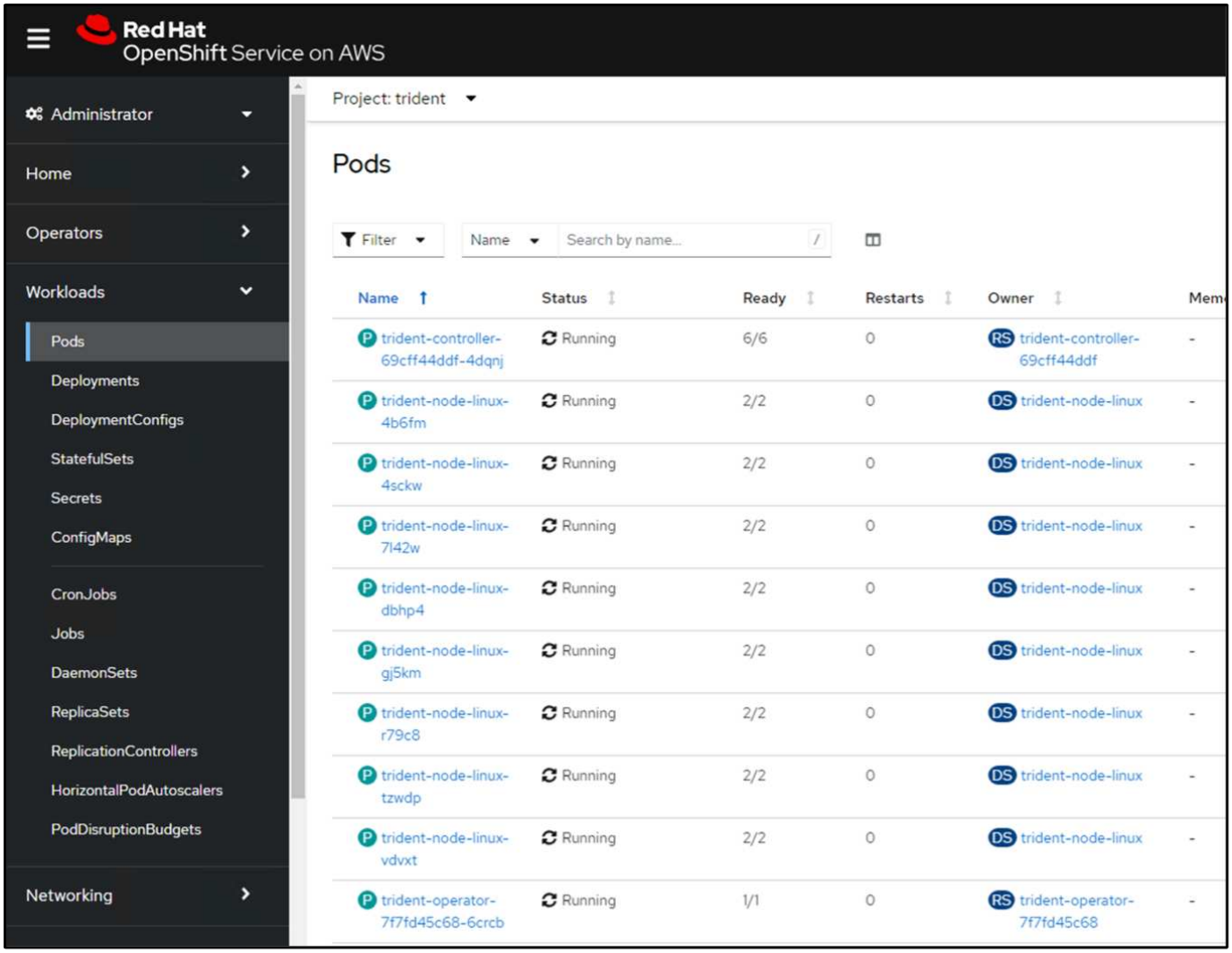
[ ] Disable usage of the repo in the developer catalog.

URL *

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



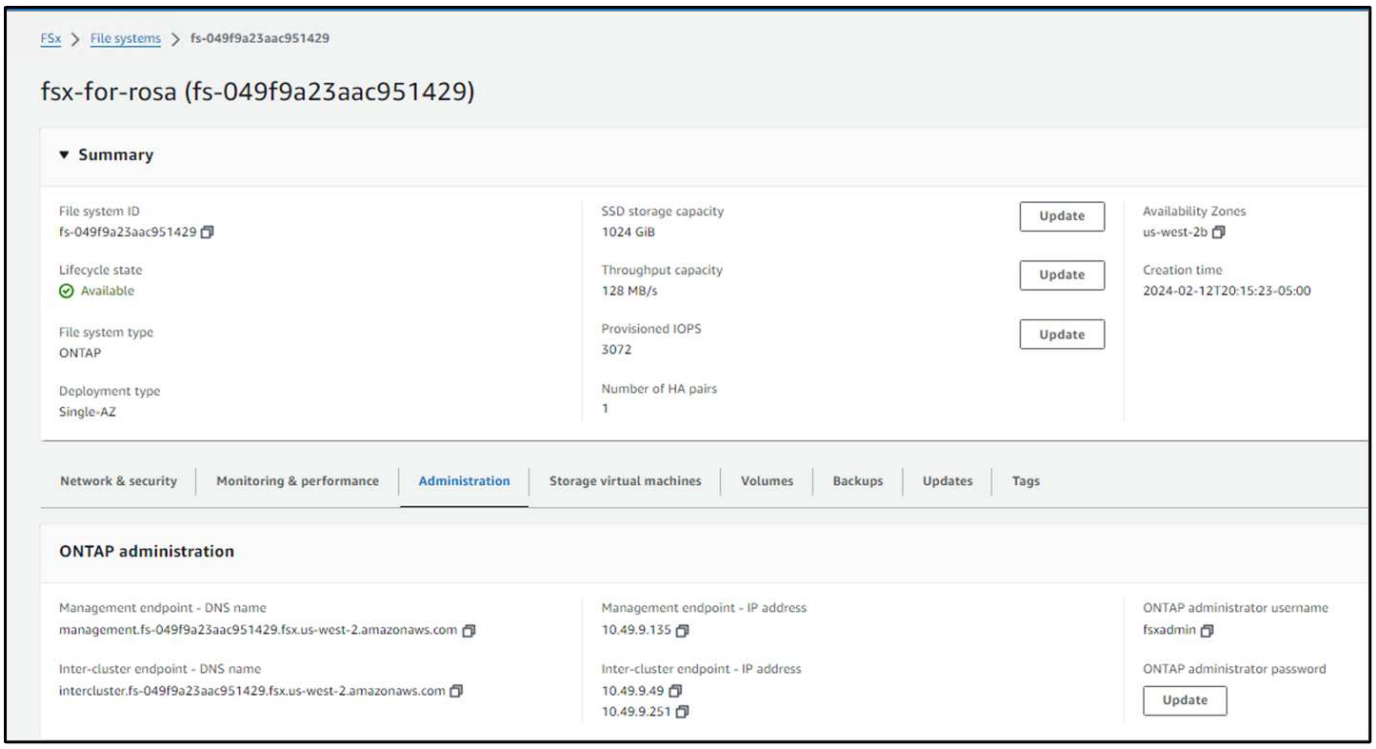
콘솔의 관리자 보기로 돌아가 트라이덴트 프로젝트에서 Pod를 선택하여 모든 트라이덴트 포드가 실행 중인지 확인합니다.



4. Trident CSI Provisioner 를 사용하여 백엔드, 스토리지 클래스 및 스냅샷 클래스 구성을 생성합니다

아래 표시된 YAML 파일을 사용하여 트리덴트 백엔드 객체, 스토리지 클래스 객체 및 Volumesnapshot 객체를 생성합니다. 생성한 Amazon FSx for NetApp ONTAP 파일 시스템에 대한 자격 증명, 백엔드의 YAML 구성에서 파일 시스템의 관리 LIF 및 가상 서버 이름을 제공해야 합니다. 이러한 세부 정보를 보려면 Amazon FSx용 AWS 콘솔로 이동하여 파일 시스템을 선택하고 관리 탭으로 이동합니다. 또한 UPDATE(업데이트)를 클릭하여 의 암호를 설정합니다 `fsxadmin` 사용자.

> 명령줄을 사용하여 개체를 만들거나 하이브리드 클라우드 콘솔에서 YAML 파일을 사용하여 개체를 만들 수 있습니다.



- Trident 백엔드 구성**

```
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
```

- 저장소 클래스**

```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
```

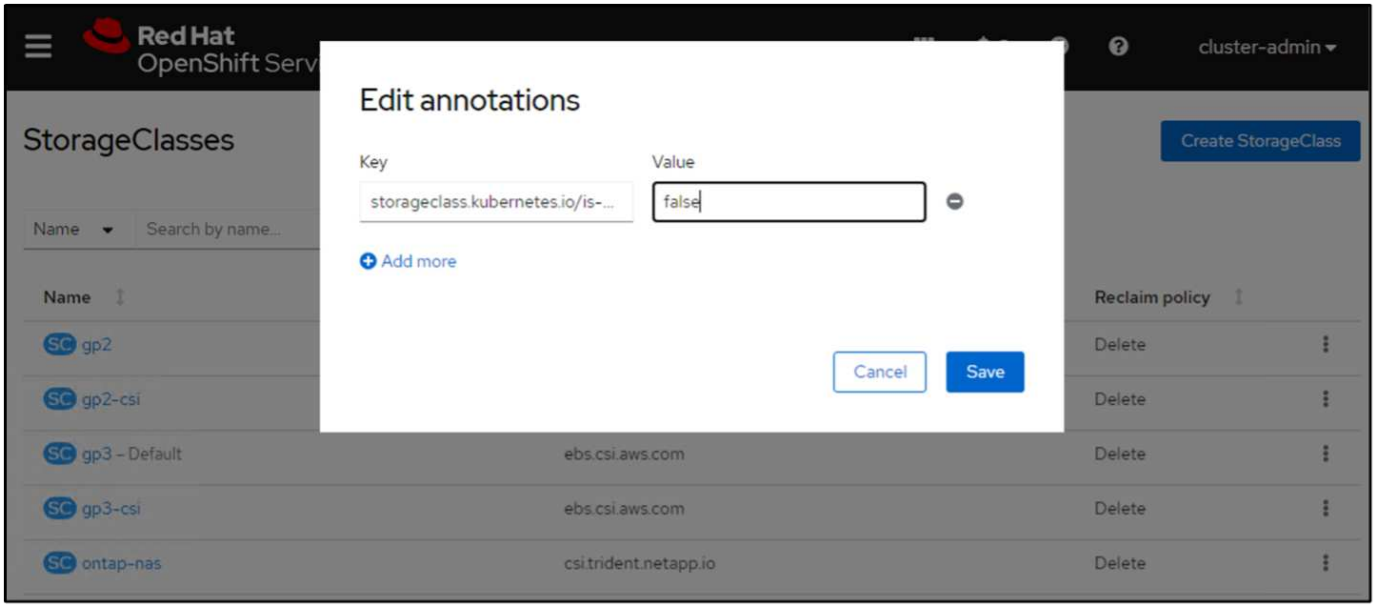
- 스냅샷 클래스**

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

아래 표시된 명령을 실행하여 백엔드, 스토리지 클래스 및 trident-snapshotclass 객체가 생성되었는지 확인합니다.

```
[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]$
```

현재 중요한 수정 사항은 나중에 구축하는 PostgreSQL 앱에서 기본 스토리지 클래스를 사용할 수 있도록 ONTAP-NAS를 GP3이 아닌 기본 스토리지 클래스로 설정하는 것입니다. 클러스터의 OpenShift 콘솔의 Storage에서 StorageClasses를 선택합니다. 현재 기본 클래스의 주석을 false로 편집하고 ONTAP-NAS 스토리지 클래스에 대해 주석 storageclass.cubernetes/is-default-class 세트를 true로 추가하십시오.



5. 클러스터에 **PostgreSQL** 애플리케이션을 구축합니다

다음과 같이 명령줄에서 응용 프로그램을 배포할 수 있습니다.

```
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]$
```

응용 프로그램 포드가 실행되고 있지 않으면 보안 컨텍스트 제약 때문에 발생한 오류가 있을 수 있습니다.

```
[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]$
```

을 편집하여 오류를 수정하십시오 `runAsUser` 및 `fsGroup` 의 필드 `statefuleset.apps/postgresql` 의 출력에 있는 uid 를 가진 개체입니다 `oc get project` 명령을 사용합니다.

```
[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]$
```

PostgreSQL 앱은 Amazon FSx for NetApp ONTAP 스토리지에서 지원하는 영구 볼륨을 실행하고 사용해야 합니다.

```
[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]$
```

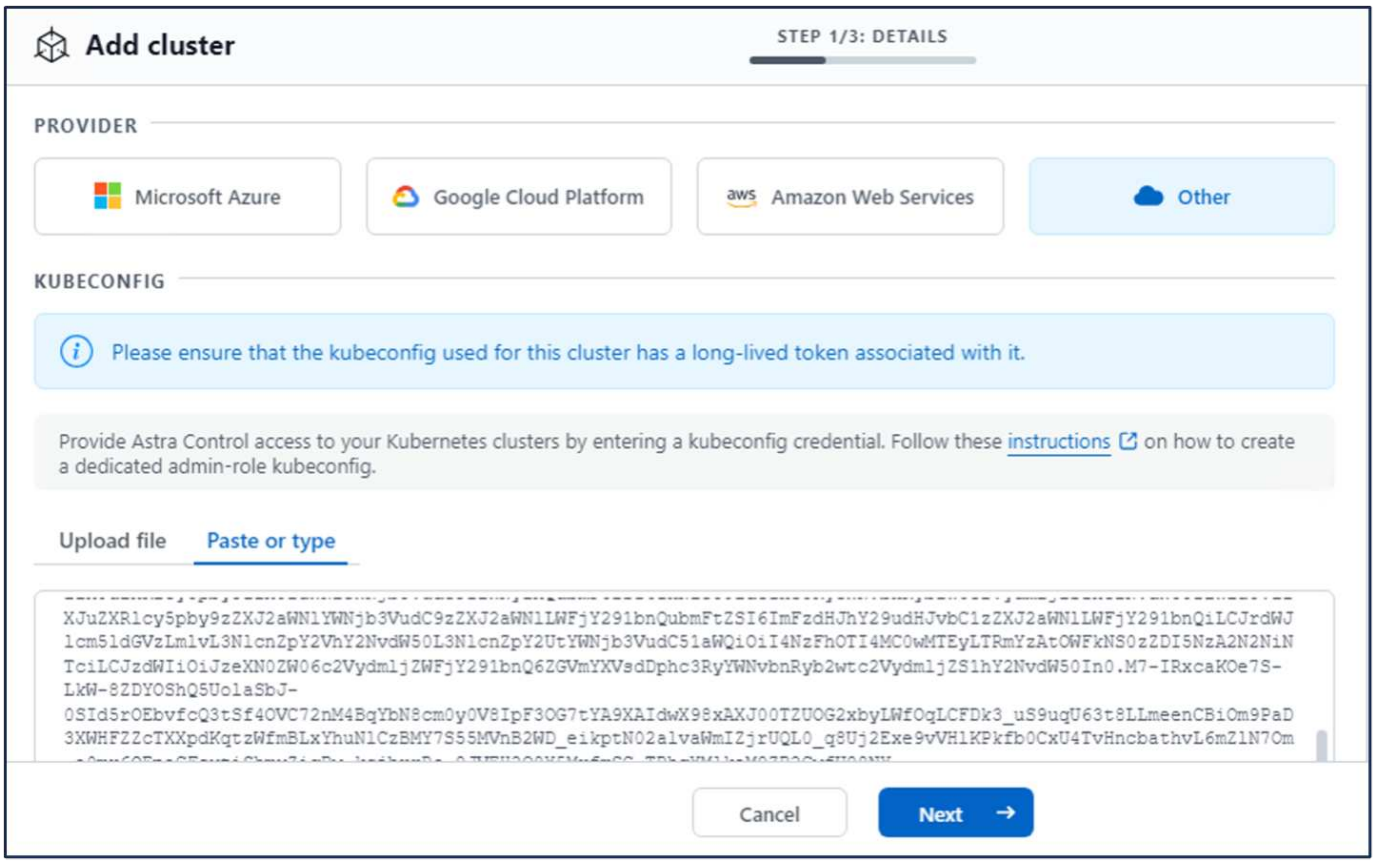
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

6. 데이터베이스를 만들고 레코드를 추가합니다

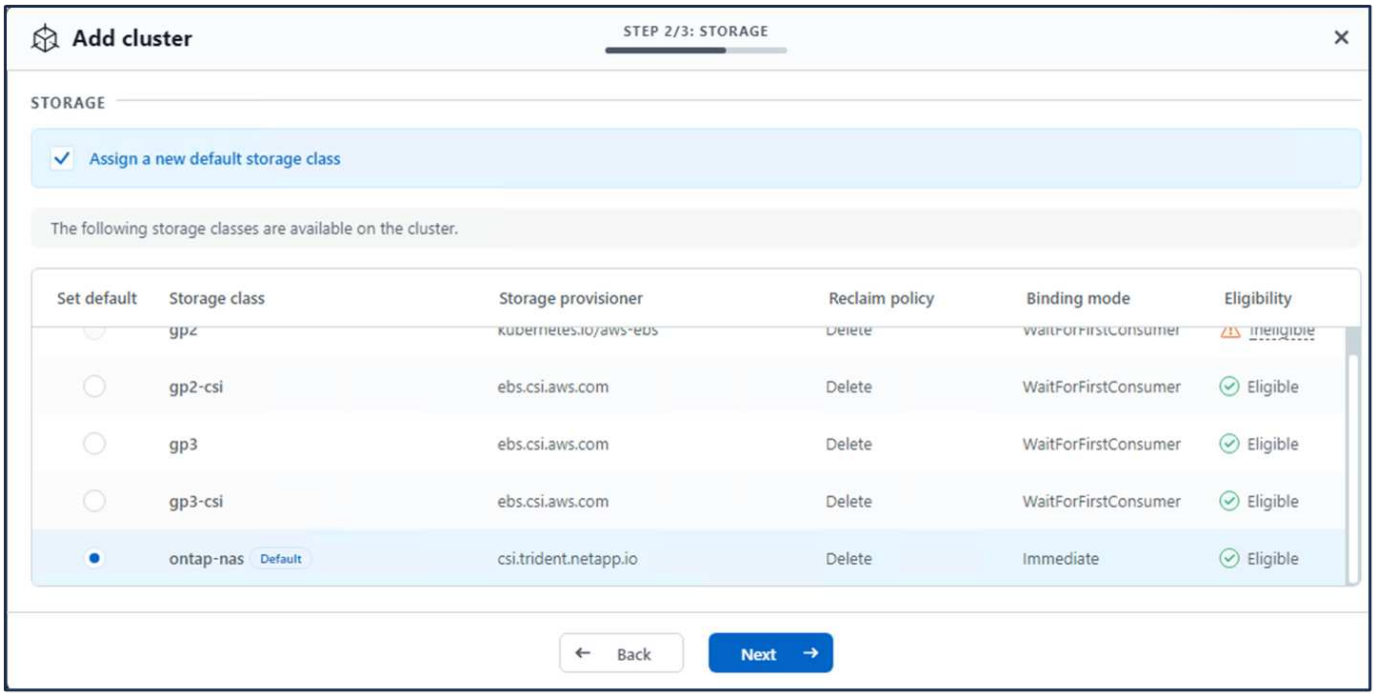
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1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

ACS에 로그인합니다. 클러스터를 선택하고 Add를 클릭합니다. 기타 를 선택하고 kubeconfig 파일을 업로드하거나 붙여 넣습니다.



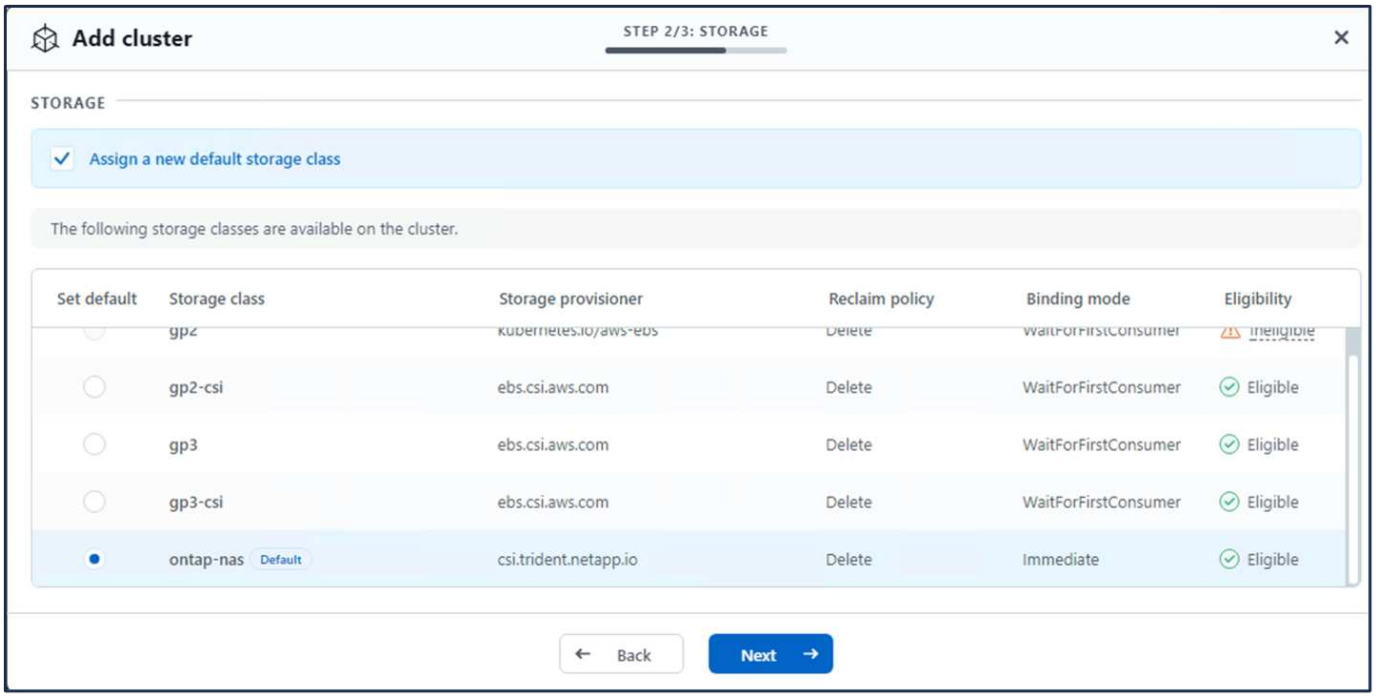
Next * 를 클릭하고 ACS의 기본 스토리지 클래스로 ONTAP-NAS 를 선택합니다. Next * 를 클릭하고 세부 정보를 검토한 후 * Add * the cluster를 클릭합니다.



8. ACS에서 응용 프로그램을 정의합니다

ACS에서 PostgreSQL 애플리케이션을 정의합니다. 시작 페이지에서 * 응용 프로그램 *, * 정의 * 를 선택하고 적절한 세부 정보를 입력합니다. 다음 * 을 두 번 클릭하고 세부 정보를 검토한 후 * 정의 * 를 클릭합니다. 응용 프로그램이

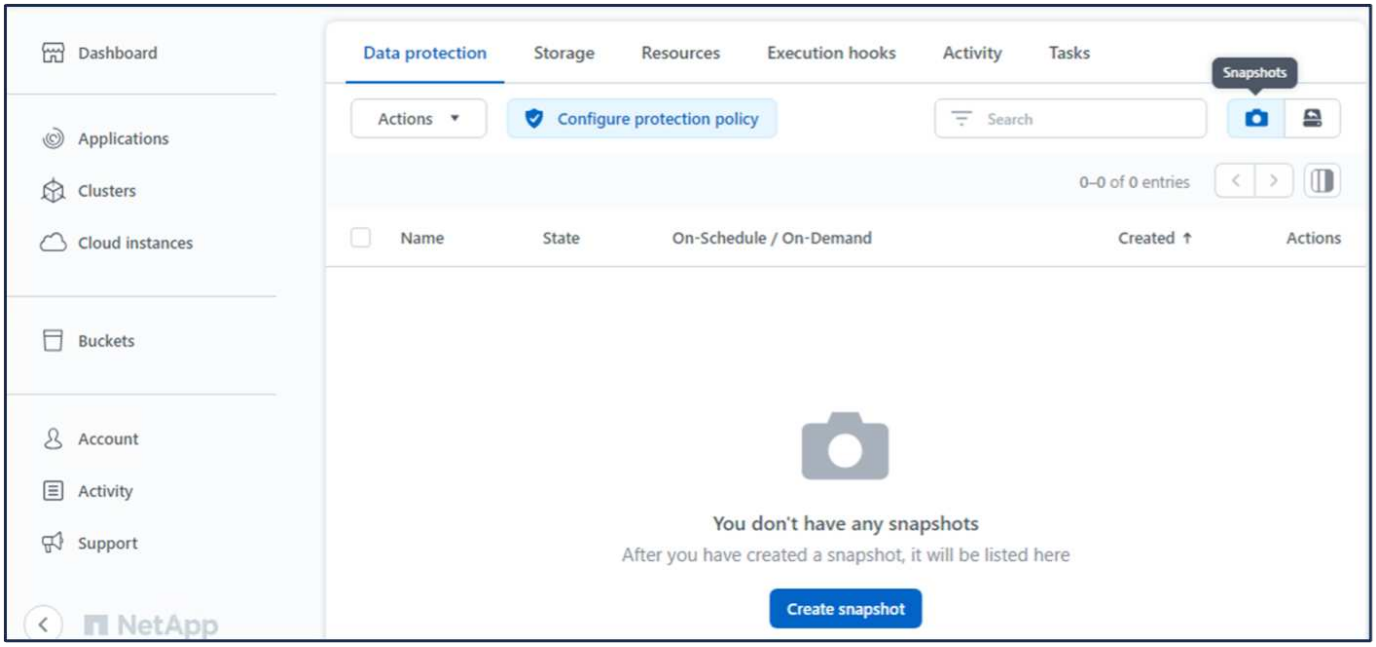
ACS에 추가됩니다.

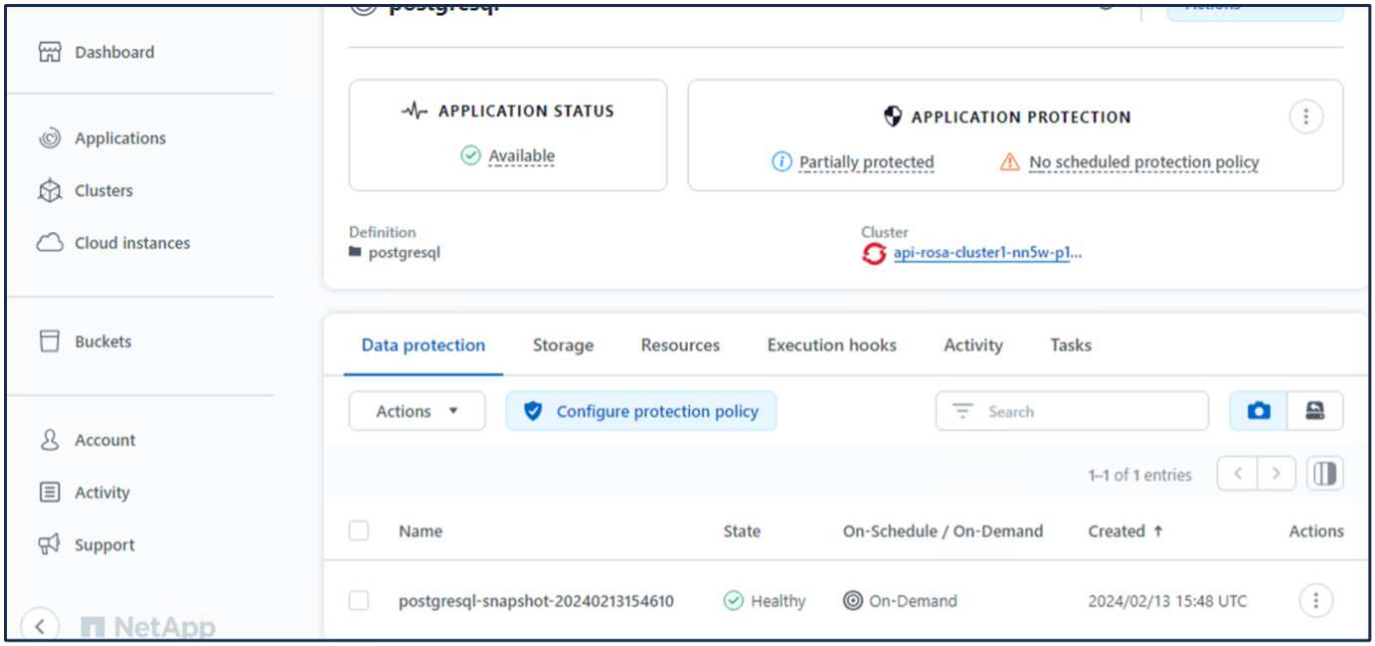


9. ACS를 사용하여 스냅샷을 생성합니다

ACS에서 스냅샷을 생성하는 방법은 여러 가지가 있습니다. 응용 프로그램을 선택하고 페이지에서 응용 프로그램의 세부 정보를 보여 주는 스냅샷을 만들 수 있습니다. 스냅샷 생성 을 클릭하여 필요 시 스냅샷을 생성하거나 보호 정책을 구성할 수 있습니다.

스냅샷 생성 * 을 클릭하고 이름을 입력하고 세부 정보를 검토한 후 * 스냅샷 * 을 클릭하여 주문형 스냅샷을 생성합니다. 작업이 완료되면 스냅샷 상태가 정상으로 변경됩니다.





10. PostgreSQL 응용 프로그램에서 데이터베이스를 삭제합니다

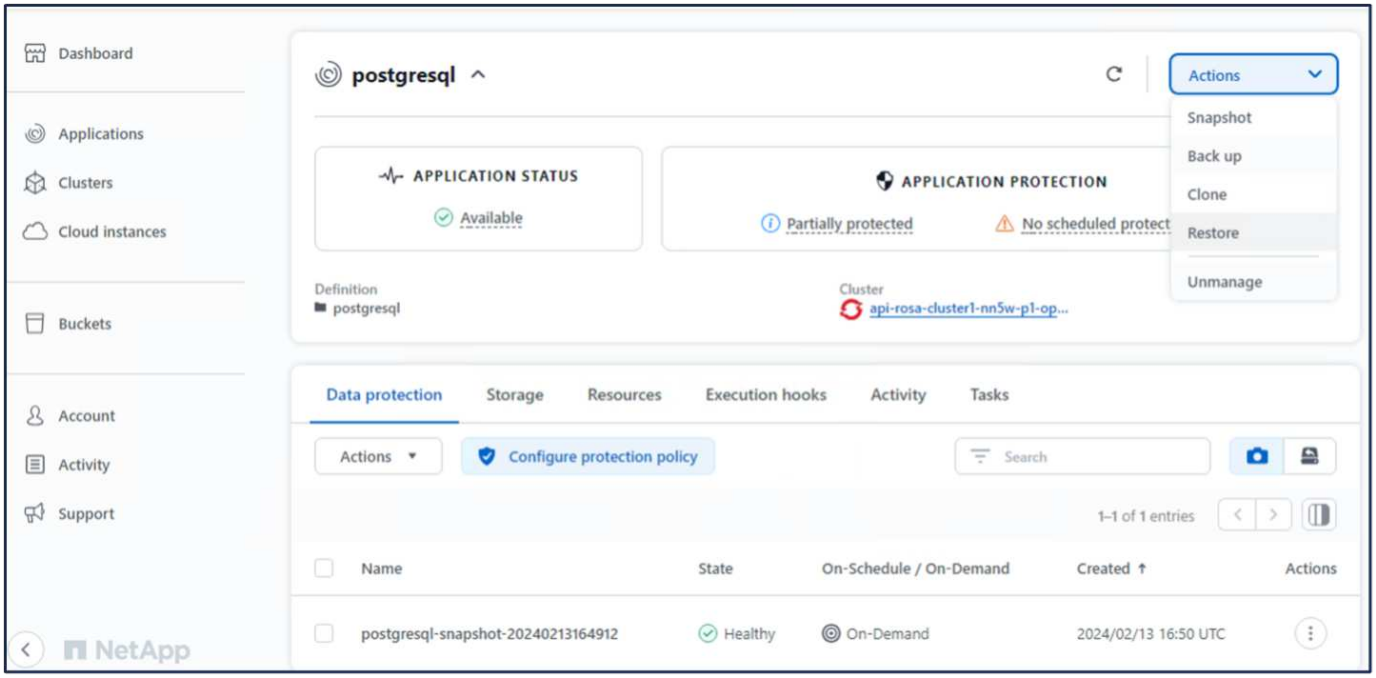
PostgreSQL에 다시 로그인하고 사용 가능한 데이터베이스를 나열한 다음 이전에 만든 데이터베이스를 삭제하고 다시 나열하여 데이터베이스가 삭제되었는지 확인합니다.

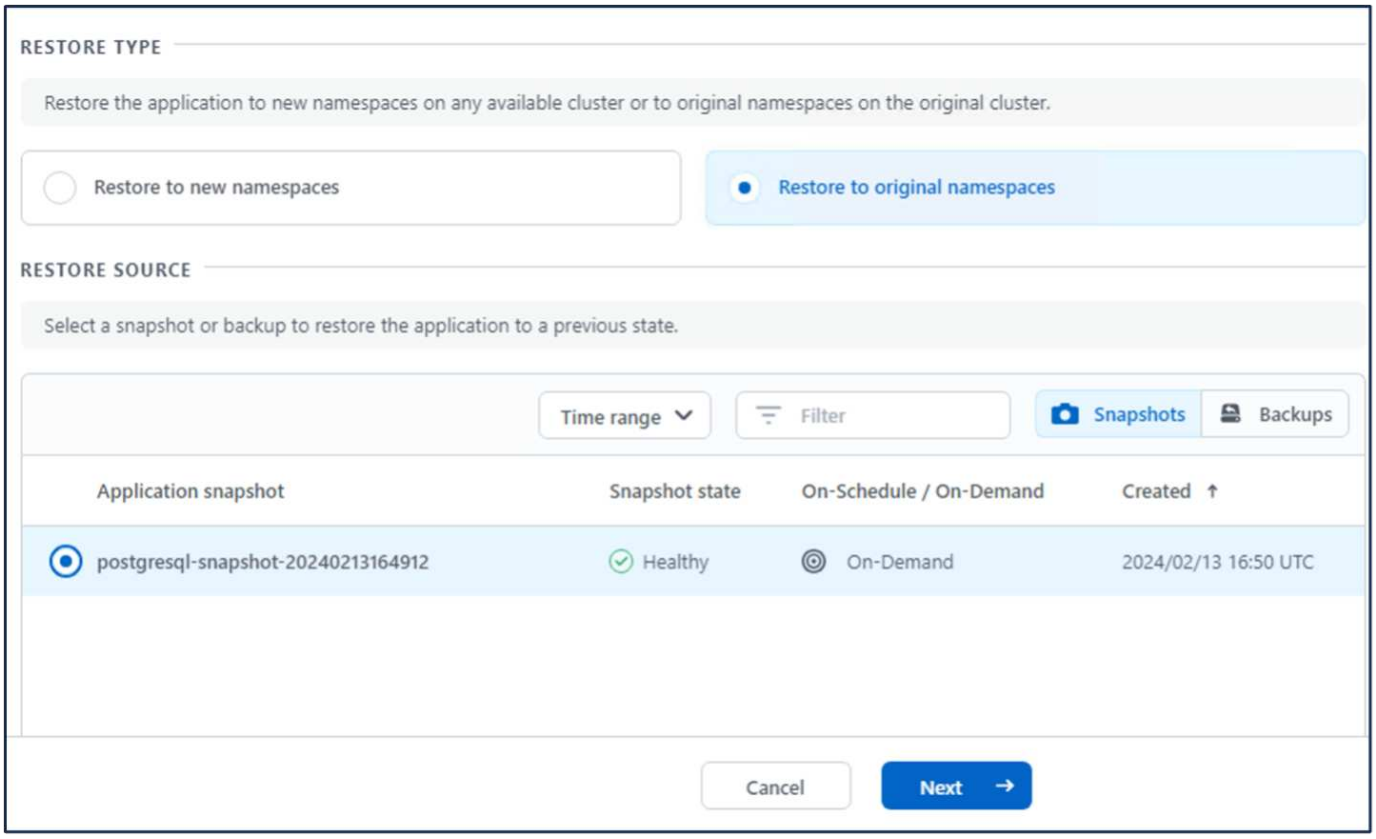
```
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------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

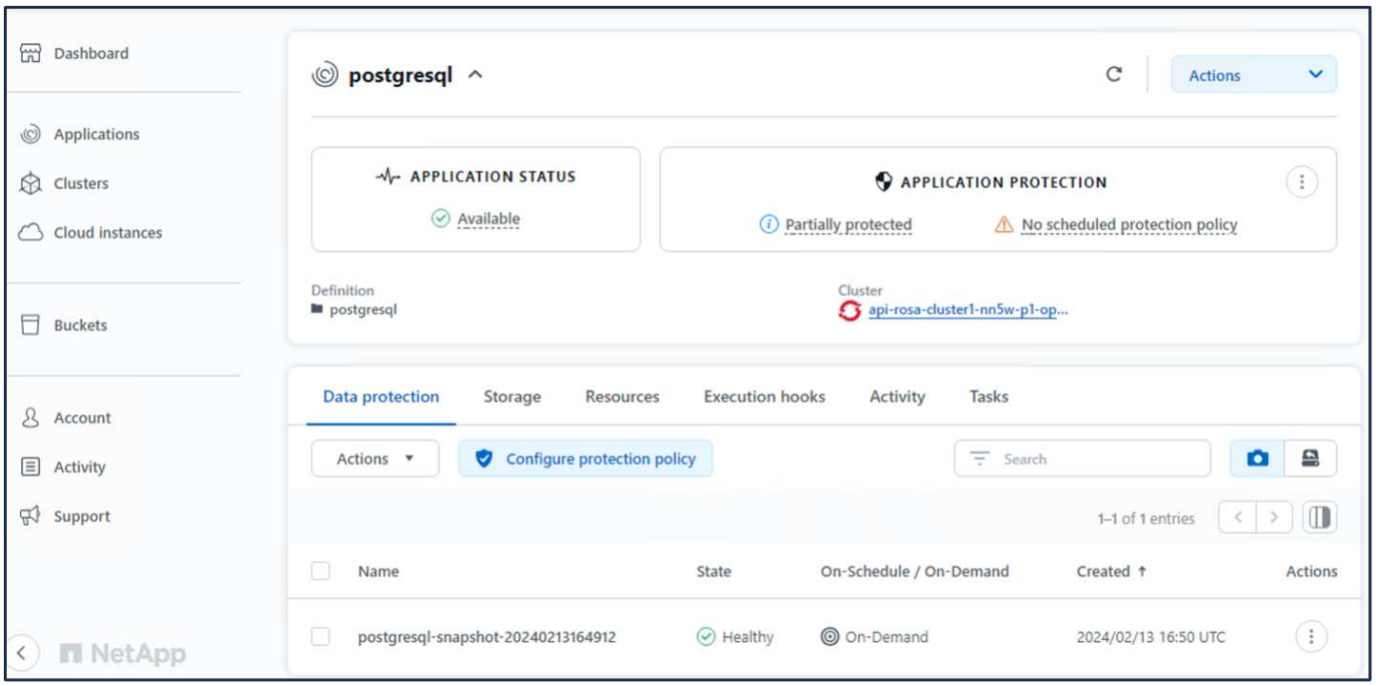
11. ACS를 사용하여 스냅샷에서 복원합니다

스냅샷에서 애플리케이션을 복원하려면 ACS UI 시작 페이지로 이동하여 애플리케이션을 선택하고 Restore(복원) 를

선택합니다. 복원할 스냅샷 또는 백업을 선택해야 합니다. (일반적으로 구성한 정책에 따라 여러 개의 를 생성할 수 있습니다.) 다음 두 화면에서 적절한 항목을 선택한 다음 * Restore * 를 클릭합니다. 스냅샷에서 복구된 후 애플리케이션 상태가 복원 중 에서 사용 가능 으로 이동합니다.







12. 앱이 스냅샷에서 복원되었는지 확인합니다

PostgreSQL 클라이언트에 로그인하면 이전에 사용했던 테이블과 레코드가 테이블에 표시됩니다. 이상입니다. 버튼을 클릭하기만 하면 프로그램이 이전 상태로 복원됩니다. Astra Control을 사용하는 고객은 이렇게 손쉽게 이용할 수 있습니다.

```
[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)
```

데이터 마이그레이션

이 페이지에는 영구 스토리지용 NetApp ONTAP용 FSx를 사용하는 관리형 Red Hat OpenShift 클러스터의 컨테이너 워크로드에 대한 데이터 마이그레이션 옵션이 나와 있습니다.

데이터 마이그레이션

AWS의 Red Hat OpenShift 서비스와 NetApp FSxN(ONTAP)용 FSx는 AWS의 서비스 포트폴리오에 포함됩니다. FSxN은 단일 AZ 또는 Multi-AZ 옵션에서 사용할 수 있습니다. Multi-AZ 옵션은 가용성 영역 장애로부터 데이터를 보호합니다. FSxN을 Astra Trident와 통합하여 Rosa 클러스터의 애플리케이션에 영구 스토리지를 제공할 수 있습니다.

제어 차트를 사용하여 **FSxN**과 **Trident** 통합

Amazon FSx for ONTAP와 Rosa Cluster 통합

컨테이너 애플리케이션 마이그레이션에는 다음이 포함됩니다.

- 영구 볼륨: BlueXP를 사용하여 이 작업을 수행할 수 있습니다. 또 다른 옵션은 Astra Control Center를 사용하여 사내에서 클라우드 환경으로 컨테이너 애플리케이션 마이그레이션을 처리하는 것입니다. 자동화는 같은 용도로 사용할 수 있습니다.
- 애플리케이션 메타데이터: OpenShift GitOps(Argo CD)를 사용하여 이 작업을 수행할 수 있습니다.

영구 스토리지에 **FSxN**을 사용하여 **Rosa** 클러스터에서 애플리케이션의 페일오버 및 페일백

다음 비디오에서는 BlueXP 및 Argo CD를 사용한 애플리케이션 장애 조치 및 장애 복구 시나리오에 대해 설명합니다.

ROSA 클러스터에서 애플리케이션의 장애 조치 및 장애 복구

OpenShift Container 워크로드를 위한 데이터 보호 및 마이그레이션 솔루션