공급자 관리 구성 요소를 갖춘 하이브리드 클라우드

NetApp public and hybrid cloud solutions

NetApp
August 18, 2025

목차

공급자 관리 구성 요소를 갖춘 하이브리드 클라우드

관리형 **Red Hat OpenShift** 컨테이너 플랫폼 워크로드를 갖춘 **NetApp** 솔루션

고객은 "클라우드에서 태어났을 수도 있고", 현대화 여정의 어느 시점에 데이터 센터에서 일부 선택된 워크로드나 모든 워크로드를 클라우드로 옮길 준비가 되었을 수도 있습니다. 이들은 워크로드를 실행하기 위해 클라우드에서 공급자가 관리하는 OpenShift 컨테이너와 공급자가 관리하는 NetApp 스토리지를 사용하기로 선택할 수 있습니다. 컨테이너 워크로드를 위한 성공적인 프로덕션 준비 환경을 구축하려면 클라우드에서 관리형 Red Hat OpenShift 컨테이너 클러스터를 계획하고 배포해야 합니다. NetApp 3대 퍼블릭 클라우드에서 관리형 Red Hat 솔루션을 위한 완전 관리형 스토리지 제품을 제공합니다.

- Amazon FSx for NetApp ONTAP (FSx ONTAP)*

FSx ONTAP AWS에서 컨테이너를 배포할 때 데이터 보호, 안정성, 유연성을 제공합니다. Trident 고객의 상태 저장 애플리케이션을 위한 영구 FSx ONTAP 스토리지를 사용하는 동적 스토리지 프로비저너 역할을 합니다.

ROSA는 여러 가용성 영역에 걸쳐 제어 평면 노드를 분산하여 HA 모드로 배포할 수 있는 반면, FSx ONTAP 도 다중 AZ 옵션으로 프로비저닝하여 높은 가용성을 제공하고 AZ 장애로부터 보호할 수 있습니다.

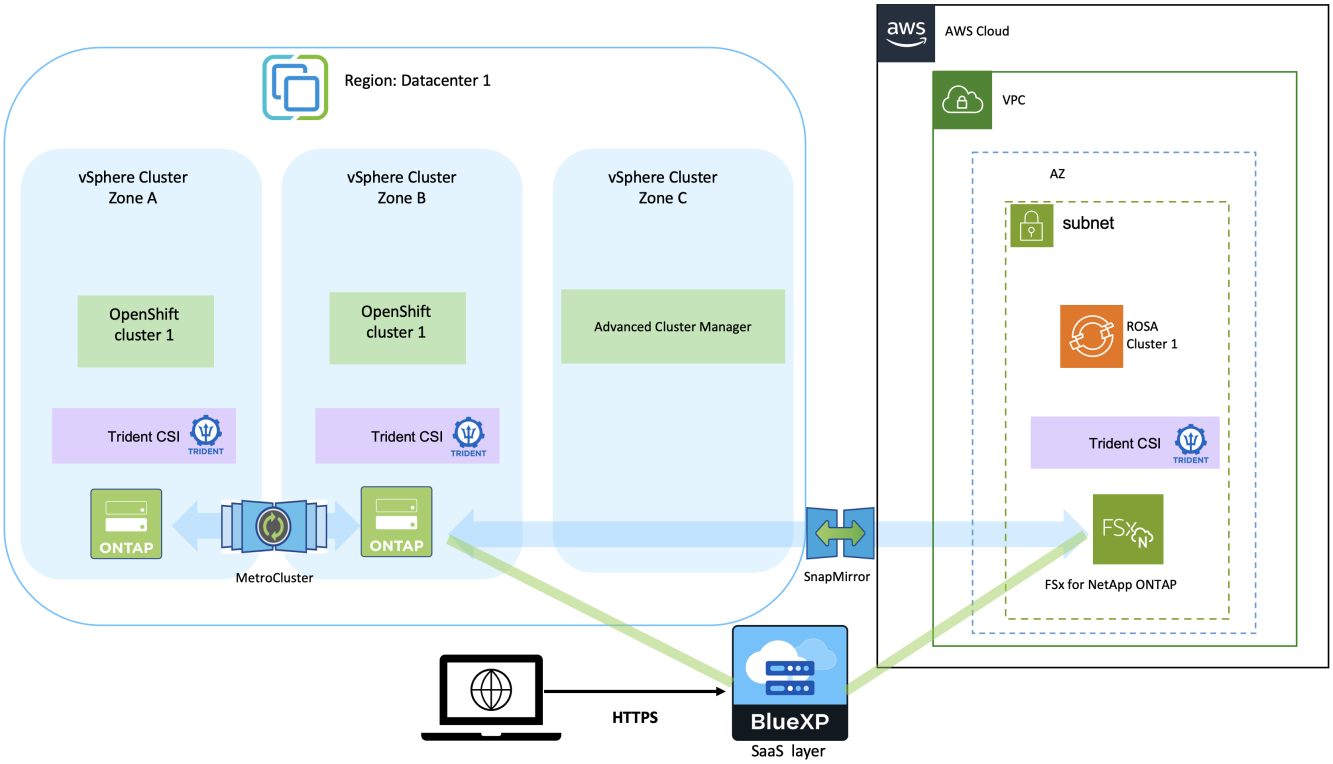
- Google Cloud NetApp Volumes*

Red Hat OpenShift Dedicated는 하이브리드 클라우드 전반에서 애플리케이션을 신속하게 구축, 배포 및 확장할 수 있는 완전 관리형 애플리케이션 플랫폼입니다. Google Cloud NetApp Volumes ONTAP의 엔터프라이즈 데이터 관리 기능 전체를 Google Cloud의 OpenShift 배포에 적용하는 영구 볼륨을 제공합니다.

AWS에서 관리형 **Red Hat OpenShift** 컨테이너 플랫폼 배포 및 구성

이 섹션에서는 AWS(ROSA)에서 관리형 Red Hat OpenShift 클러스터를 설정하는 고급 워크플로를 설명합니다. Trident 가 영구 볼륨을 제공하기 위해 스토리지 백엔드로 관리형 Amazon FSx for NetApp ONTAP (FSx ONTAP)을 사용하는 방법을 보여줍니다. BlueXP 사용하여 AWS에 FSx ONTAP 을 배포하는 방법에 대한 세부 정보가 제공됩니다. 또한, ROSA 클러스터의 상태 저장 애플리케이션에 대한 데이터 보호 및 마이그레이션 활동을 수행하기 위해 BlueXP 와 OpenShift GitOps(Argo CD)를 사용하는 방법에 대한 세부 정보도 제공됩니다.

다음은 AWS에 배포되고 FSx ONTAP 백엔드 스토리지로 사용하는 ROSA 클러스터를 보여주는 다이어그램입니다.



이 솔루션은 AWS의 두 VPC에 있는 두 개의 ROSA 클러스터를 사용하여 검증되었습니다. 각 ROSA 클러스터는 Trident 사용하여 FSx ONTAP 과 통합되었습니다. AWS에 ROSA 클러스터와 FSx ONTAP 배포하는 방법에는 여러 가지가 있습니다. 설정에 대한 이러한 간략한 설명은 사용된 특정 방법에 대한 문서 링크를 제공합니다. 제공된 관련 링크에서 다른 방법을 참조할 수 있습니다."리소스 섹션" .

설정 과정은 다음 단계로 나눌 수 있습니다.

ROSA 클러스터 설치

- 두 개의 VPC를 만들고 VPC 간에 VPC 피어링 연결을 설정합니다.
- 나타내다"여기" ROSA 클러스터를 설치하는 방법에 대한 지침입니다.

FSx ONTAP 설치

- BlueXP 에서 VPC에 FSx ONTAP 설치합니다. 나타내다"여기" BlueXP 계정을 생성하고 시작하려면 여기를 클릭하세요. 나타내다"여기" FSx ONTAP 설치하기 위해서. 나타내다"여기" AWS에서 FSx ONTAP 관리하기 위한 커넥터를 생성합니다.
- AWS를 사용하여 FSx ONTAP 배포합니다. 나타내다"여기" AWS 콘솔을 사용하여 배포합니다.

ROSA 클러스터에 **Trident** 설치**(Helm** 차트 사용**)**

- Helm 차트를 사용하여 ROSA 클러스터에 Trident 설치합니다. 설명서 링크를 참조하세요: https://docs.netapp.com/us-en/trident/trident-get-started/kubernetes-deploy-helm.html [여기].

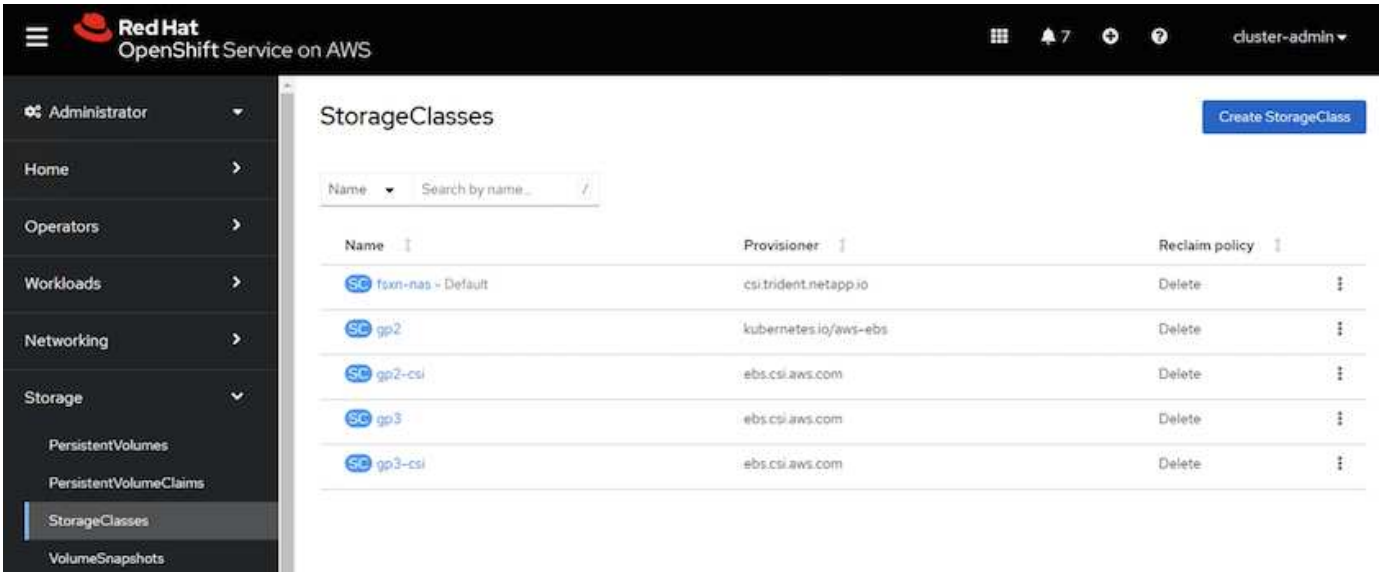
ROSA 클러스터를 위한 Trident 와 FSx ONTAP 통합

OpenShift GitOps를 사용하면 ApplicationSet을 사용하여 ArgoCD에 등록되는 모든 관리형 클러스터에 Trident CSI를 배포할 수 있습니다.

```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
```

Trident (FSx ONTAP 용)를 사용하여 백엔드 및 스토리지 클래스 생성

- 나타내다"여기" 백엔드와 스토리지 클래스 생성에 대한 자세한 내용은 다음을 참조하세요.
- OpenShift 콘솔에서 Trident CSI를 기본값으로 사용하여 FsxN에 대해 생성된 스토리지 클래스를 만듭니다. 아래의 스크린샷을 참조하세요.



OpenShift GitOps(Argo CD)를 사용하여 애플리케이션 배포

- 클러스터에 OpenShift GitOps 운영자를 설치합니다. 지침을 참조하세요"여기" .
- 클러스터에 대한 새로운 Argo CD 인스턴스를 설정합니다. 지침을 참조하세요"여기" .

Argo CD의 콘솔을 열고 앱을 배포합니다. 예를 들어, Helm Chart와 Argo CD를 사용하여 Jenkins 앱을 배포할 수

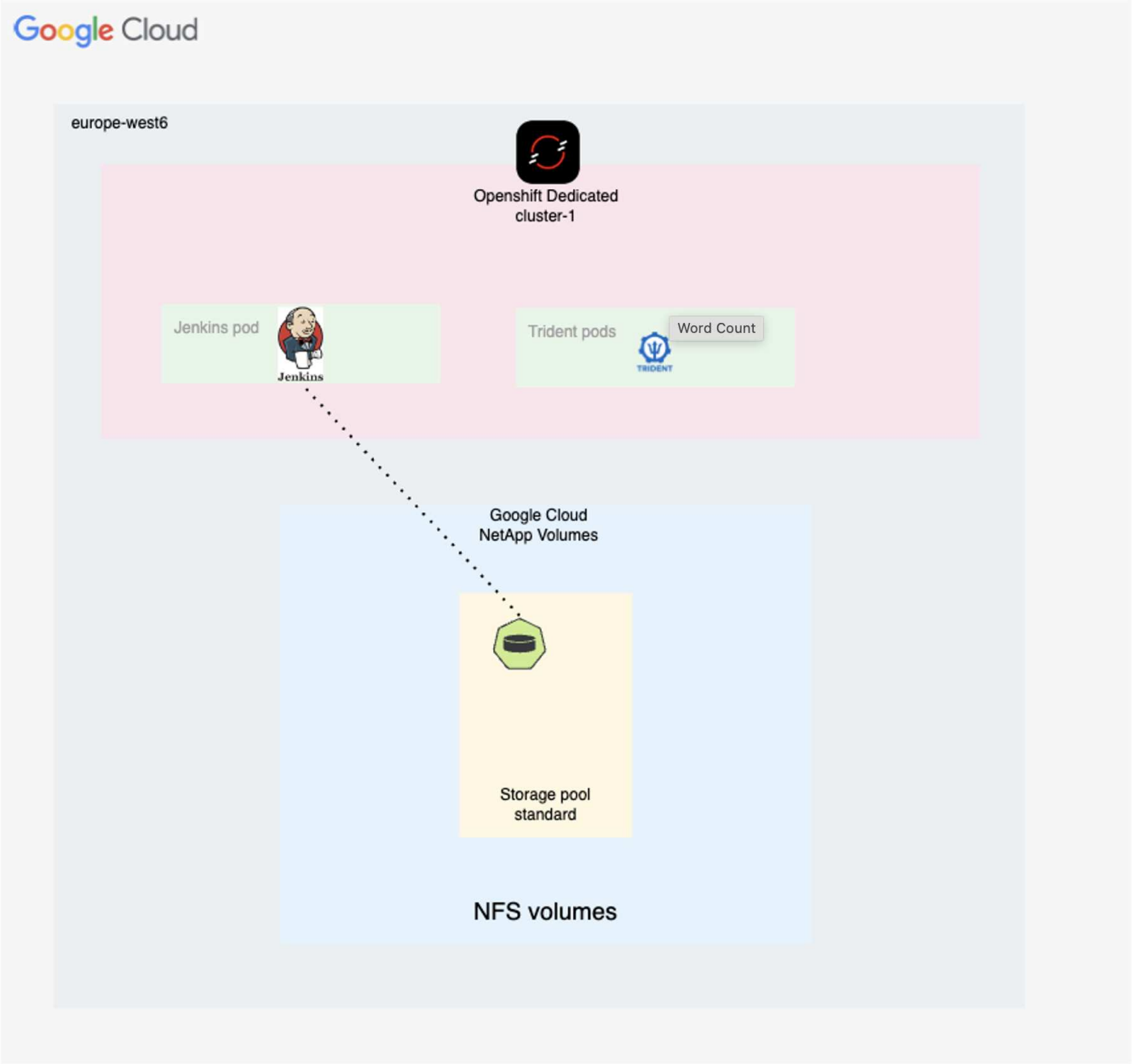
있습니다. 애플리케이션을 생성할 때 다음 세부 정보가 제공되었습니다. 프로젝트: 기본 클러스터 :'https://kubernetes.default.svc' (따옴표 없이) 네임스페이스: Jenkins Helm 차트의 URL:'https://charts.bitnami.com/bitnami' (따옴표 없이)

Helm 매개변수: global.storageClass: fsxn-nas

Google Cloud NetApp Volumes 사용하여 Google Cloud에 OpenShift Dedicated 배포 및 구성

이 섹션에서는 Google Cloud 플랫폼에서 OpenShift Dedicated(OSD) 클러스터를 설정하는 개략적인 워크플로를 설명합니다. NetApp Trident Kubernetes로 실행되는 상태 저장 애플리케이션에 영구 볼륨을 제공하기 위해 스토리지 백엔드로 Google Cloud NetApp Volumes 사용하는 것을 보여줍니다.

다음은 Google Cloud에 배포되고 NetApp Volumes를 백엔드 스토리지로 사용하는 OSD 클러스터를 보여주는 다이어그램입니다.



설정 과정은 다음 단계로 나눌 수 있습니다.

Google Cloud에 **OSD** 클러스터 설치

- 클러스터에 기존 VPC를 사용하려면 OSD 클러스터에 대한 VPC, 두 개의 서브넷, 클라우드 라우터, 두 개의 GCP 클라우드 NAT를 만들어야 합니다. 나타내다"여기" 지침을 보려면 클릭하세요.
- 나타내다"여기" CCS(Customer Cloud Subscription) 청구 모델을 사용하여 GCP에 OSD 클러스터를 설치하는 방법에 대한 지침을 확인하세요. OSD는 Google Cloud Marketplace에도 포함되어 있습니다. Google Cloud Marketplace 솔루션을 사용하여 OSD를 설치하는 방법을 보여주는 비디오가 있습니다."여기" .

Google Cloud NetApp Volumes 활성화

- 나타내다"여기" Google Cloud NetApp Volumes 에 대한 액세스 설정에 대한 정보는 다음을 참조하세요. 다음 단계를 모두 따르세요.
- 스토리지 풀을 생성합니다. 나타내다"여기" Google Cloud NetApp Volumes 에서 스토리지 풀을 설정하는 방법에 대한 정보입니다. OSD에서 실행되는 상태 저장 Kubernetes 애플리케이션의 볼륨은 스토리지 풀 내에 생성됩니다.

OSD 클러스터에 **Trident** 설치**(Helm** 차트 사용**)**

- Helm 차트를 사용하여 OSD 클러스터에 Trident 설치합니다. 나타내다"여기" Helm Chart를 설치하는 방법에 대한 지침은 여기를 참조하세요. 헬름 차트를 찾을 수 있습니다"여기" .

OSD 클러스터를 위한 **NetApp Trident** 와 **NetApp Volumes** 통합

Trident 사용하여 백엔드 및 스토리지 클래스 생성(Google Cloud NetApp Volumes 용)

- 백엔드 생성에 대한 자세한 내용은 여기를 참조하세요.
- 현재 쿠버네티스의 스토리지 클래스 중에 기본값으로 표시된 것이 있다면, 스토리지 클래스를 편집하여 해당 주석을 제거하세요.
- Trident CSI 프로비저너를 사용하여 NetApp 볼륨에 대한 스토리지 클래스를 하나 이상 만듭니다. 주석을 사용하여 저장 클래스 중 하나만 기본값으로 지정합니다. 이를 통해 PVC는 PVC 매니페스트에서 명시적으로 호출되지 않은 경우에도 이 저장 클래스를 사용할 수 있습니다. 주석이 포함된 예는 아래와 같습니다.

```
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
```

OpenShift GitOps(Argo CD)를 사용하여 애플리케이션 배포

- 클러스터에 OpenShift GitOps 운영자를 설치합니다. 지침을 참조하세요"여기" .
- 클러스터에 대한 새로운 Argo CD 인스턴스를 설정합니다. 지침을 참조하세요"여기" .

Argo CD의 콘솔을 열고 앱을 배포합니다. 예를 들어, Helm Chart와 Argo CD를 사용하여 Jenkins 앱을 배포할 수 있습니다. 애플리케이션을 생성할 때 다음 세부 정보가 제공되었습니다. 프로젝트: 기본 클러스터 :'https://kubernetes.default.svc' (따옴표 없이) 네임스페이스: Jenkins Helm 차트의 URL:'https://charts.bitnami.com/bitnami' (따옴표 없이)

데이터 보호

이 페이지에서는 Astra Control Service를 사용하여 AWS(ROSA) 클러스터에서 관리되는 Red Hat OpenShift에 대한 데이터 보호 옵션을 보여줍니다. Astra Control Service(ACS)는 클러스터를 추가하고, 클러스터에서 실행되는 애플리케이션을 정의하고, 애플리케이션 인식 데이터 관리 활동을 수행할 수 있는 사용하기 쉬운 그래픽 사용자 인터페이스를 제공합니다. ACS 기능은 워크플로 자동화를 허용하는 API를 통해서도 접근할 수 있습니다.

Astra Control(ACS 또는 ACC)을 구동하는 것은 NetApp Trident 입니다. Trident Red Hat OpenShift, EKS, AKS, SUSE Rancher, Anthos 등과 같은 여러 유형의 Kubernetes 클러스터를 FAS/ AFF, ONTAP Select, CVO, Google

Google Cloud NetApp Volumes, Azure NetApp Files , Amazon FSx ONTAP 과 같은 다양한 NetApp ONTAP 스토리지와 통합합니다.

이 섹션에서는 ACS를 사용한 다음 데이터 보호 옵션에 대한 세부 정보를 제공합니다.

- 한 지역에서 실행 중인 ROSA 애플리케이션의 백업 및 복원을 보여주고 다른 지역으로 복원하는 방법을 보여주는 비디오입니다.
- ROSA 애플리케이션의 스냅샷과 복원을 보여주는 비디오입니다.
- ROSA 클러스터와 Amazon FSx ONTAP 설치하는 방법, NetApp Trident 사용하여 스토리지 백엔드와 통합하는 방법, ROSA 클러스터에 PostgreSQL 애플리케이션을 설치하는 방법, ACS를 사용하여 애플리케이션 스냅샷을 만드는 방법 및 해당 스냅샷에서 애플리케이션을 복원하는 방법에 대한 단계별 세부 정보입니다.
- ACS를 사용하여 FSx ONTAP 있는 ROSA 클러스터에서 MySQL 애플리케이션의 스냅샷을 만들고 복원하는 방법에 대한 단계별 세부 정보를 보여주는 블로그입니다.

백업/백업에서 복원

다음 비디오는 한 지역에서 실행 중인 ROSA 애플리케이션의 백업을 다른 지역으로 복원하는 과정을 보여줍니다.

AWS에서 Red Hat OpenShift 서비스를 위한 FSx NetApp ONTAP

스냅샷/스냅샷에서 복원

다음 비디오에서는 ROSA 애플리케이션의 스냅샷을 찍고 이후 스냅샷에서 복원하는 방법을 보여줍니다.

Amazon FSx ONTAP 스토리지를 사용하는 AWS(ROSA) 클러스터의 Red Hat OpenShift Service에 대한 애플리케이션 스냅샷/복원

블로그

- "Amazon FSx 스토리지를 사용하여 ROSA 클러스터의 앱 데이터 관리를 위한 Astra Control Service 사용"

스냅샷을 생성하고 이를 통해 복원하는 단계별 세부 정보

필수 설정

- "AWS 계정"
- "Red Hat OpenShift 계정"
- IAM 사용자"적절한 권한" ROSA 클러스터를 생성하고 액세스하려면
- "AWS CLI"
- "로사 CLI"
- "오픈시프트 CLI"(오케이)
- 서브넷과 적절한 게이트웨이 및 경로가 있는 VPC
- "ROSA 클러스터 설치됨"VPC로
- "Amazon FSx ONTAP"동일한 VPC에서 생성됨
- ROSA 클러스터에 대한 액세스"OpenShift 하이브리드 클라우드 콘솔"

다음 단계

1. 관리자 사용자를 만들고 클러스터에 로그인합니다.
2. 클러스터에 대한 kubeconfig 파일을 만듭니다.
3. 클러스터에 Trident 설치합니다.
4. Trident CSI 프로비저너를 사용하여 백엔드, 스토리지 클래스 및 스냅샷 클래스 구성을 만듭니다.
5. 클러스터에 postgresql 애플리케이션을 배포합니다.
6. 데이터베이스를 만들고 레코드를 추가합니다.
7. 클러스터를 ACS에 추가합니다.
8. ACS에서 애플리케이션을 정의합니다.
9. ACS를 사용하여 스냅샷을 만듭니다.
10. postgresql 애플리케이션에서 데이터베이스를 삭제합니다.
11. ACS를 사용하여 스냅샷에서 복원합니다.
12. 스냅샷에서 앱이 복원되었는지 확인하세요.

**1. 관리자 사용자를 생성하고 클러스터에 로그인합니다.

다음 명령을 사용하여 관리자 사용자를 만들어 ROSA 클러스터에 액세스합니다. (설치 시 관리자 사용자를 만들지 않은 경우에만 관리자 사용자를 만들어야 합니다.)

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

이 명령을 실행하면 다음과 같은 출력이 제공됩니다. 클러스터에 로그인하려면 다음을 사용하세요. `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. 클러스터를 생성할 때 이미 관리자 사용자를 생성한 경우 관리자 사용자 자격 증명을 사용하여 Red Hat OpenShift Hybrid Cloud 콘솔에서 클러스터에 로그인할 수 있습니다. 그런 다음 로그인한 사용자 이름이 표시되는 오른쪽 상단 모서리를 클릭하면 다음을 얻을 수 있습니다. `oc login` 명령줄에 대한 명령(토큰 로그인).

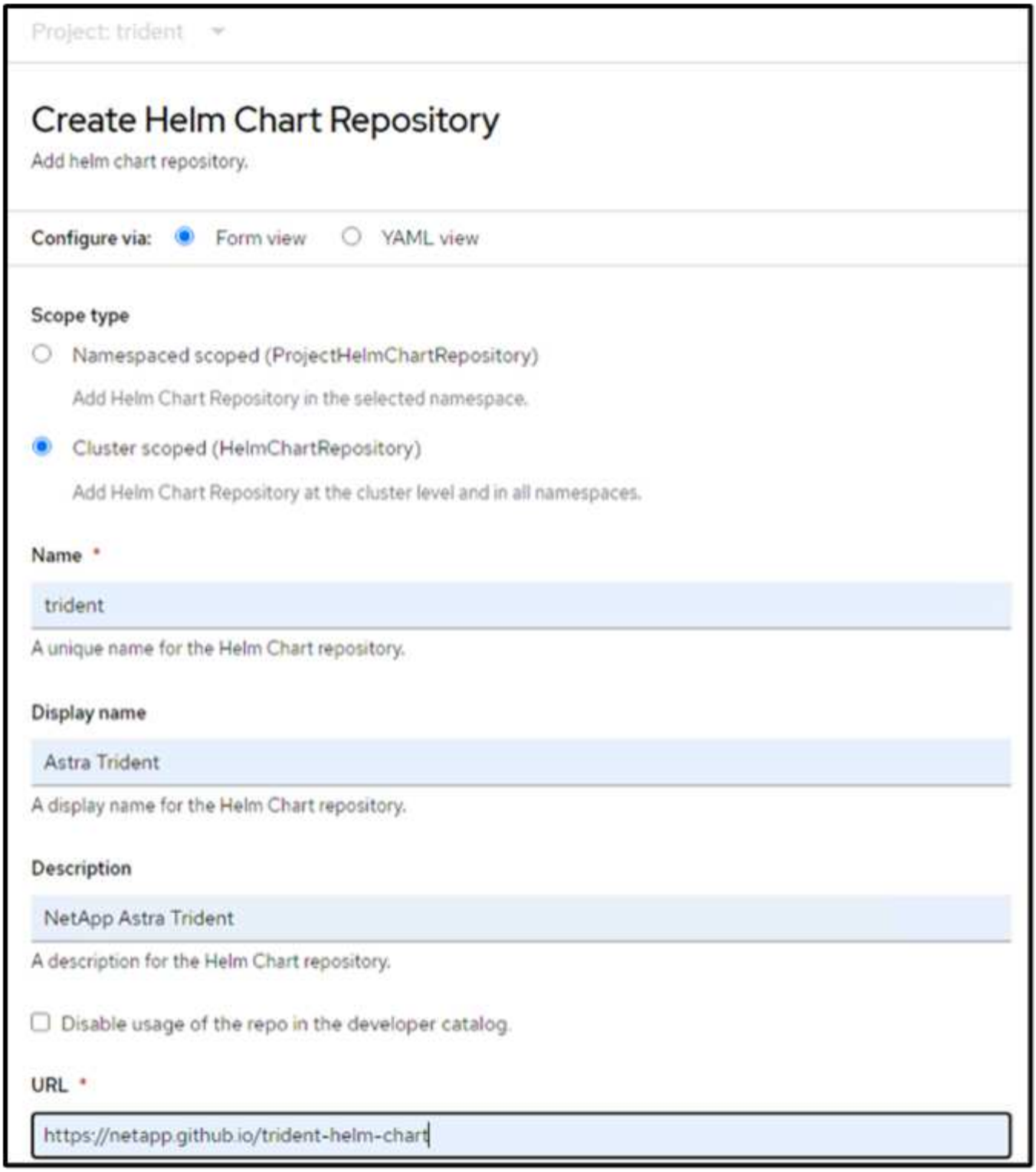
2. 클러스터에 대한 **kubeconfig 파일을 만듭니다.

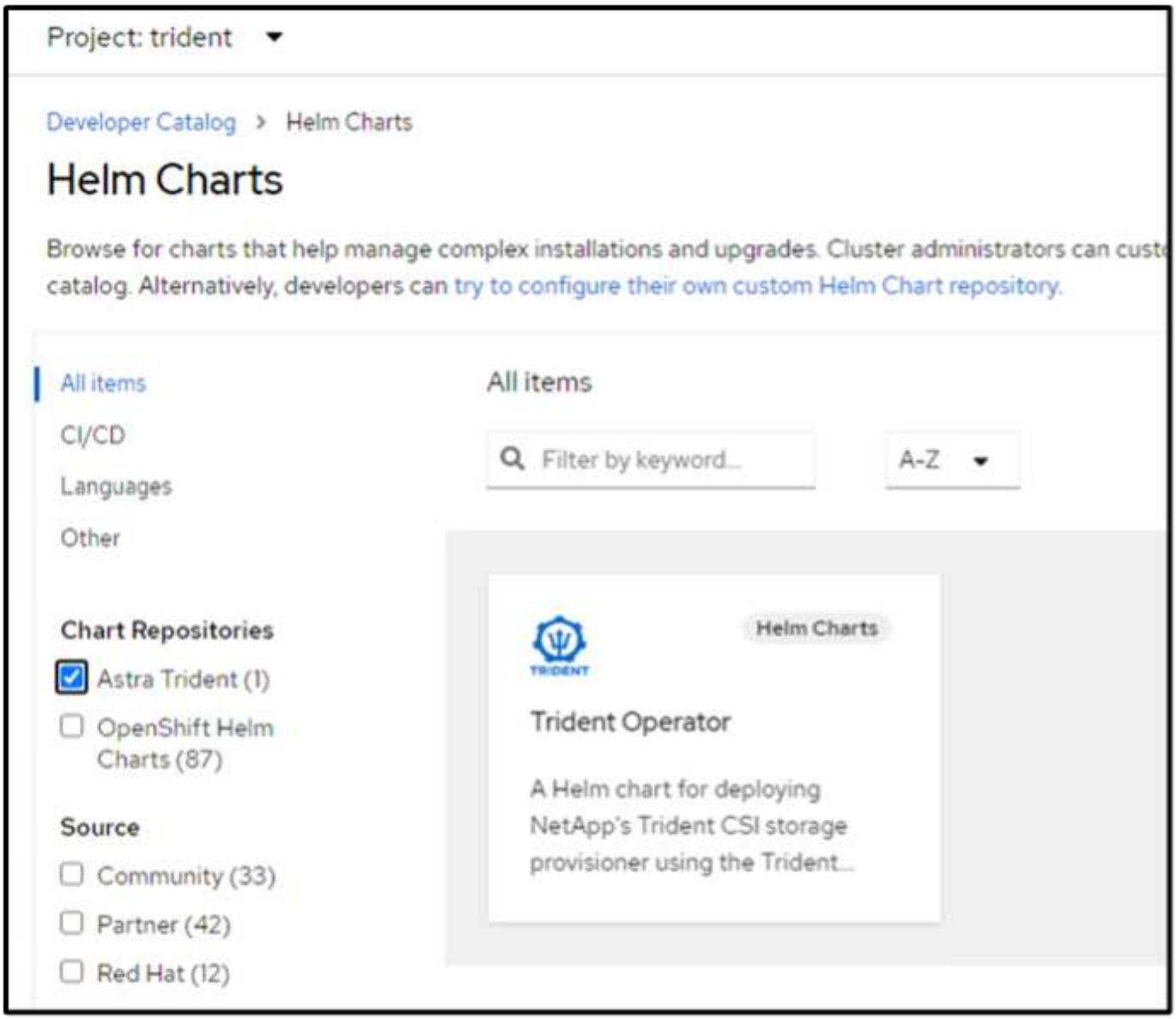
절차를 따르세요"여기" ROSA 클러스터에 대한 kubeconfig 파일을 생성합니다. 이 kubeconfig 파일은 나중에 ACS에 클러스터를 추가할 때 사용됩니다.

3. 클러스터에 **Trident** 설치

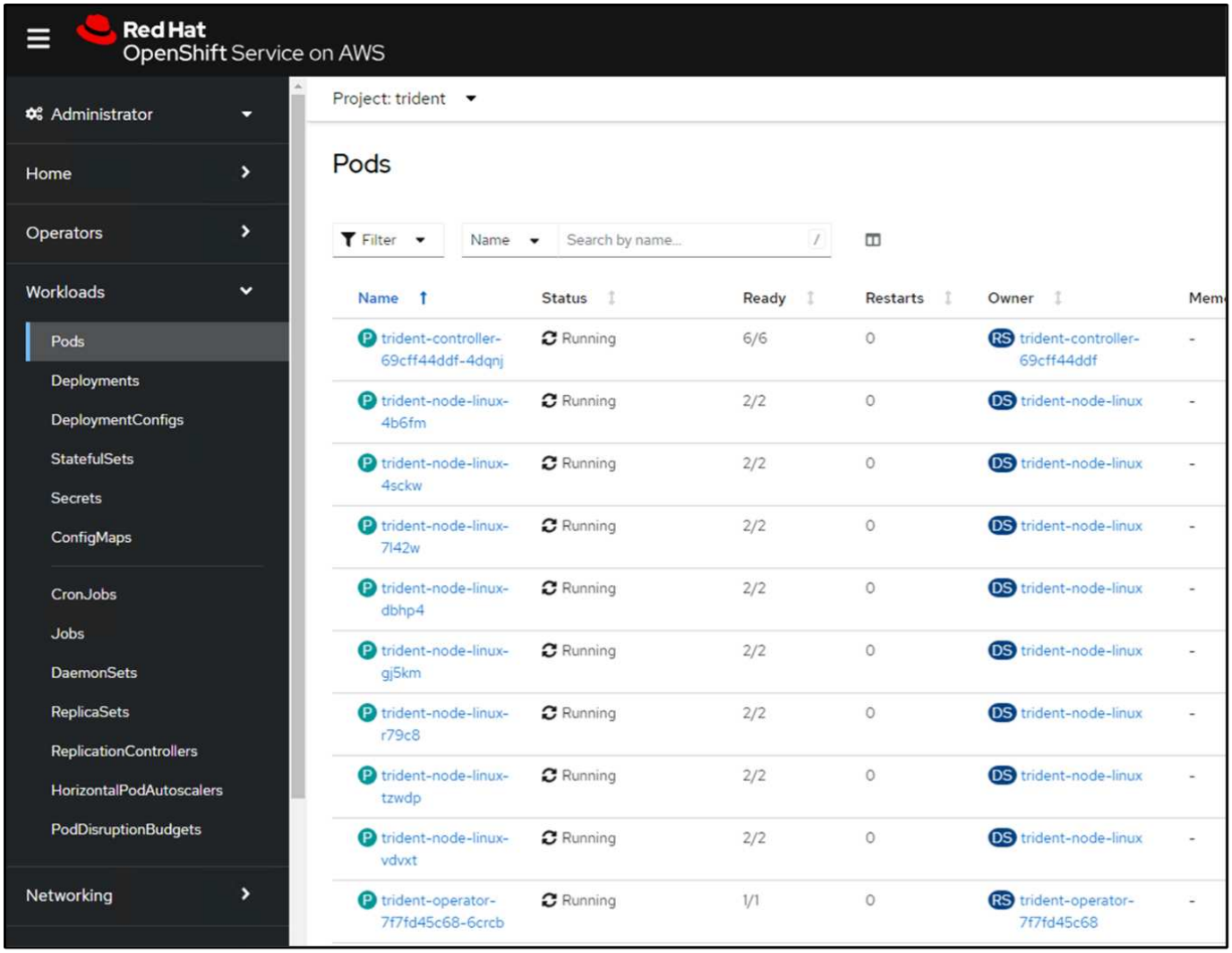
ROSA 클러스터에 Trident (최신 버전)를 설치합니다. 이를 위해 다음 절차 중 하나를 따르세요."여기" . 클러스터 콘솔에서 helm을 사용하여 Trident 설치하려면 먼저 Trident 라는 프로젝트를 만듭니다.

그런 다음 개발자 보기에서 Helm 차트 저장소를 만듭니다. URL 필드에는 다음을 사용합니다. `'https://netapp.github.io/trident-helm-chart'` . 그런 다음 Trident 운영자에 대한 헬름 릴리스를 만듭니다.


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



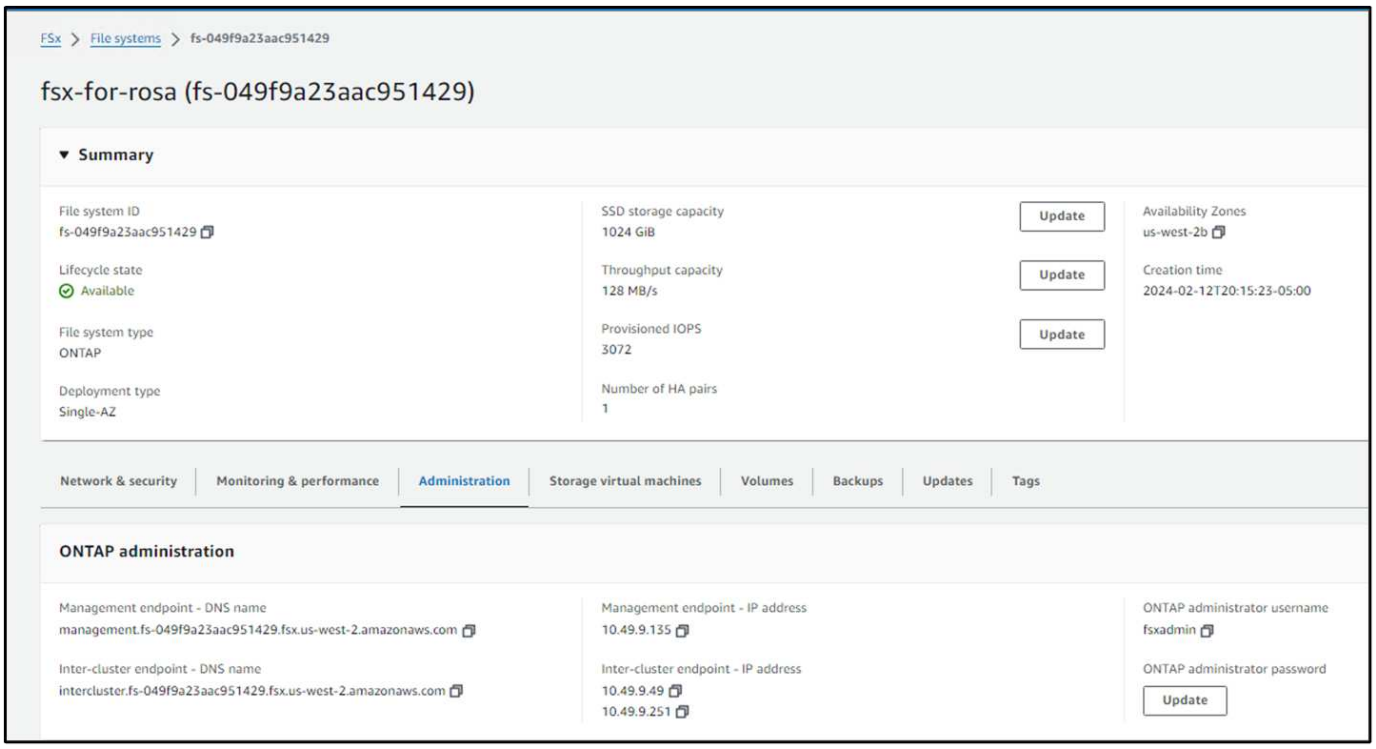
콘솔의 관리자 보기로 돌아가서 트라이던트 프로젝트에서 포드를 선택하여 모든 트라이던트 포드가 실행 중인지 확인합니다.



4. Trident CSI 프로비저너를 사용하여 백엔드, 스토리지 클래스 및 스냅샷 클래스 구성을 만듭니다.

아래에 표시된 yaml 파일을 사용하여 트라이던트 백엔드 객체, 스토리지 클래스 객체 및 Volumesnapshot 객체를 만듭니다. 백엔드의 구성 YAML에 생성한 Amazon FSx ONTAP 파일 시스템에 대한 자격 증명, 관리 LIF 및 파일 시스템의 vserver 이름을 제공해야 합니다. 해당 세부 정보를 얻으려면 Amazon FSx 용 AWS 콘솔로 이동하여 파일 시스템을 선택하고 관리 탭으로 이동하세요. 또한 업데이트를 클릭하여 비밀번호를 설정하세요. `fsxadmin` 사용자.

명령줄을 사용하여 객체를 만들거나 하이브리드 클라우드 콘솔에서 yaml 파일을 사용하여 객체를 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
```

저장 클래스

```
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
```

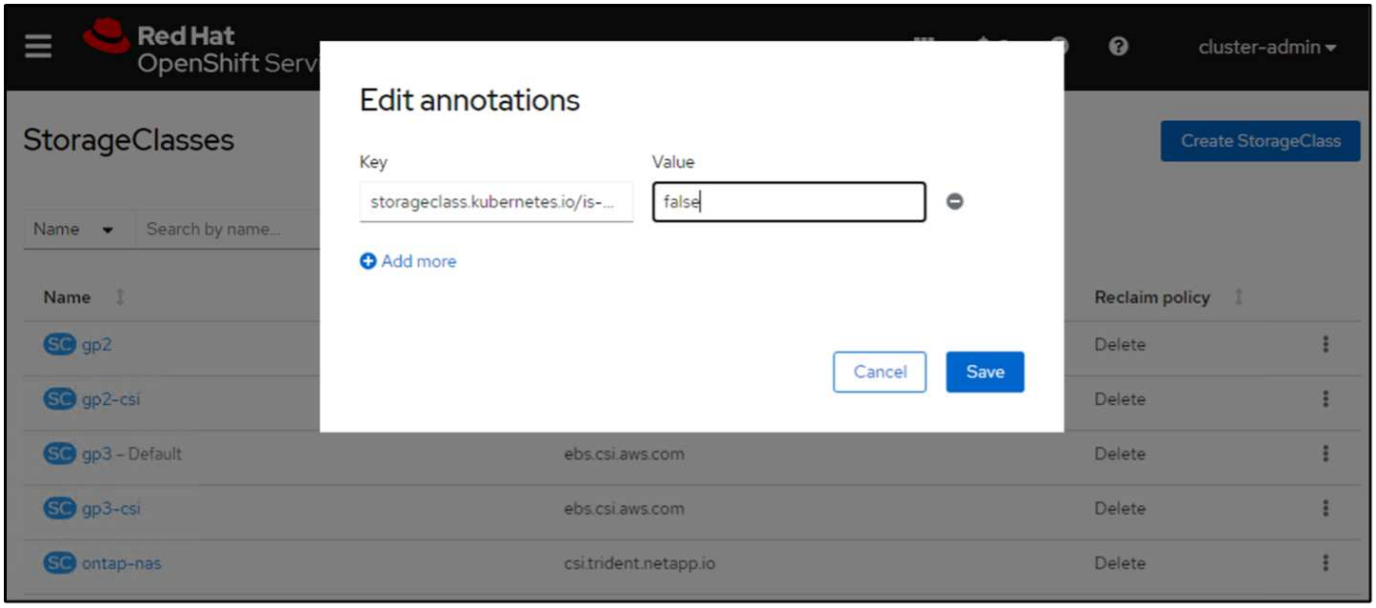
스냅샷 수업

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

아래 명령을 실행하여 백엔드, 스토리지 클래스 및 trident-snapshotclass 개체가 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]$
```

이때 해야 할 중요한 수정 사항은 gp3 대신 ontap-nas를 기본 스토리지 클래스로 설정하는 것입니다. 이렇게 하면 나중에 배포하는 postgresql 앱에서 기본 스토리지 클래스를 사용할 수 있습니다. 클러스터의 Openshift 콘솔에서 저장소 아래에 있는 StorageClasses를 선택합니다. 현재 기본 클래스의 주석을 false로 편집하고 ontap-nas 스토리지 클래스에 대해 true로 설정된 주석 storageclass.kubernetes.io/is-default-class를 추가합니다.



5. 클러스터에 **postgresql** 애플리케이션 배포

다음과 같이 명령줄에서 애플리케이션을 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]$
```

애플리케이션 포드가 실행되고 있지 않은 경우 보안 컨텍스트 제약으로 인해 오류가 발생했을 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]$
```

편집하여 오류를 수정하세요. `runAsUser` 그리고 `fsGroup` 의 필드 `statefuleset.apps/postgresql` 출력에 있는 uid를 갖는 객체 `oc get project` 아래와 같이 명령을 내립니다

.

```
[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 실행 중이어야 하며 Amazon FSx ONTAP 스토리지로 지원되는 영구 볼륨을 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]$
```

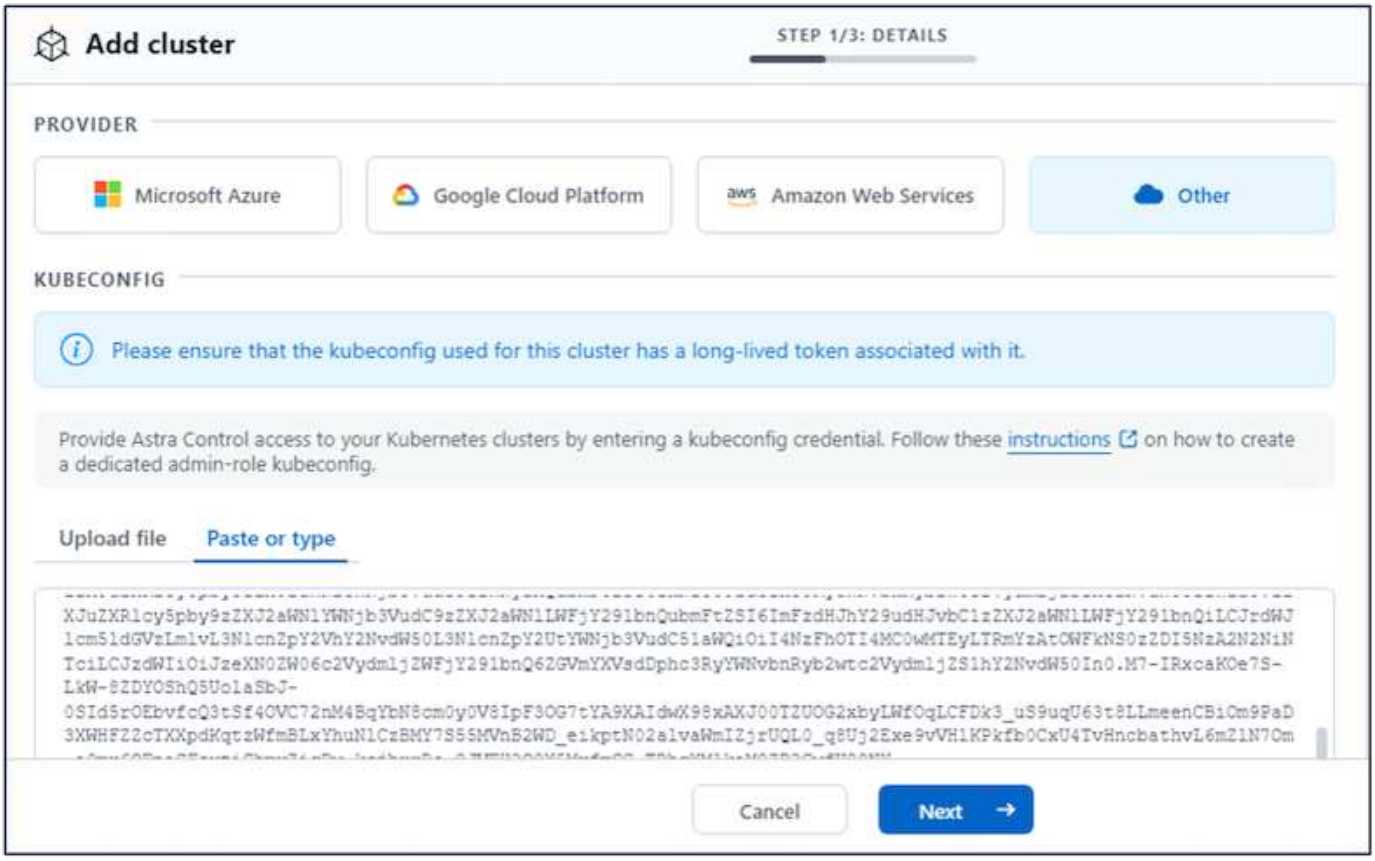
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

6. 데이터베이스를 생성하고 레코드를 추가합니다.

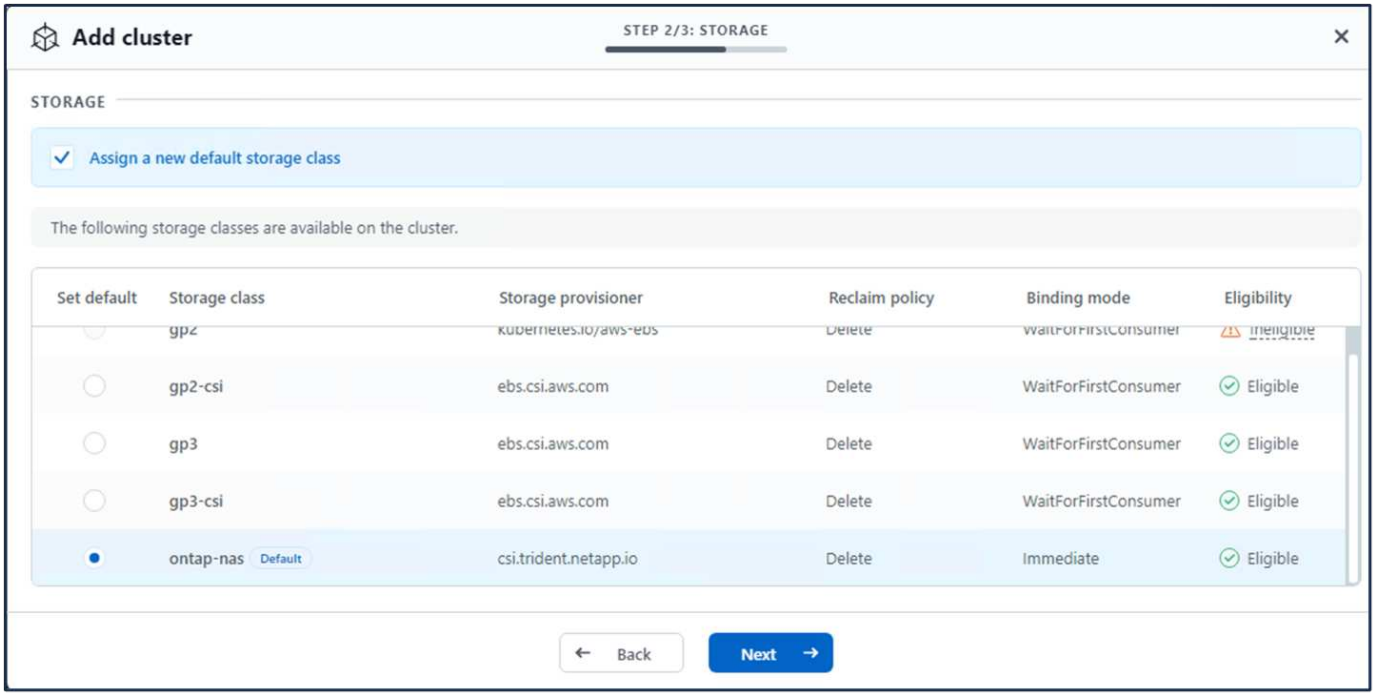
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1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

ACS에 로그인하세요. 클러스터를 선택하고 추가를 클릭합니다. 다른 것을 선택하고 kubeconfig 파일을 업로드하거나 붙여넣습니다.



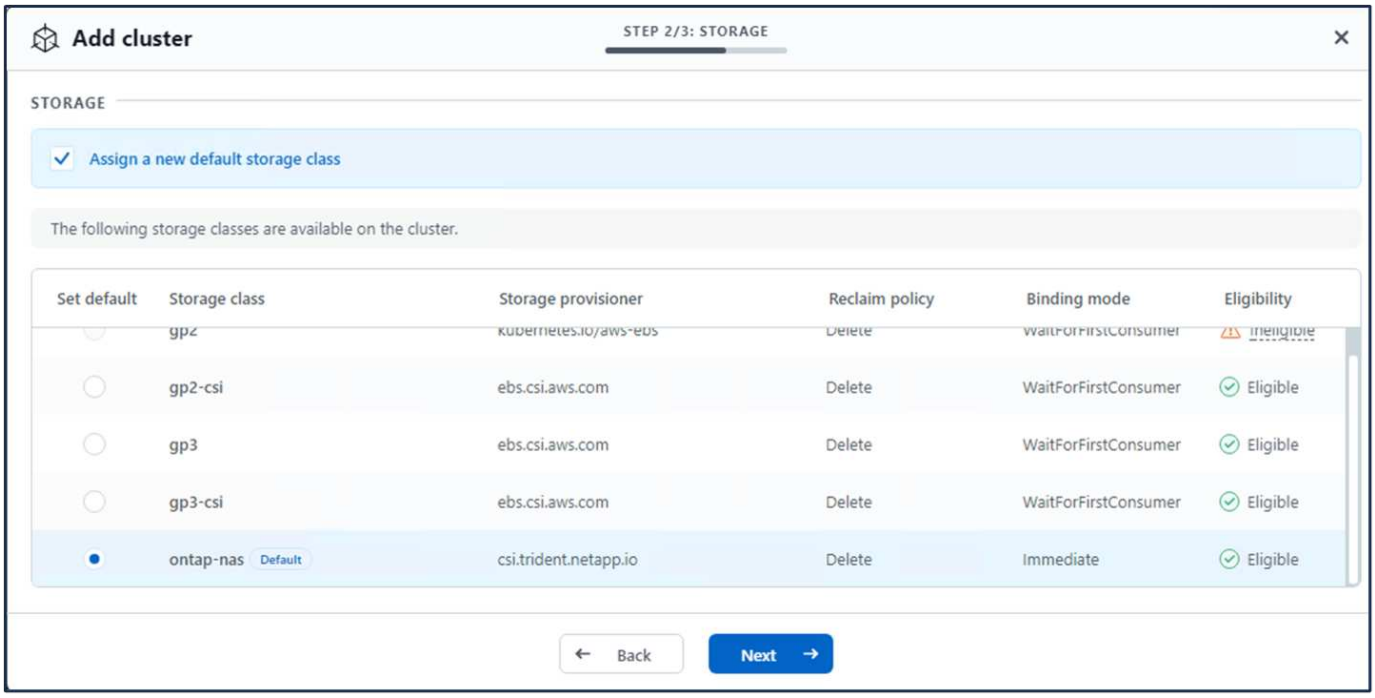
*다음*을 클릭하고 ACS의 기본 스토리지 클래스로 ontap-nas를 선택합니다. *다음*을 클릭하고 세부 정보를 검토한 후 클러스터를 *추가*합니다.



8. ACS에서 애플리케이션을 정의합니다.

ACS에서 postgresql 애플리케이션을 정의합니다. 랜딩 페이지에서 애플리케이션, *정의*를 선택하고 적절한 세부 정보를 입력합니다. *다음*을 두 번 클릭하고, 세부 정보를 검토한 후 *정의*를 클릭합니다. 해당 애플리케이션이 ACS에

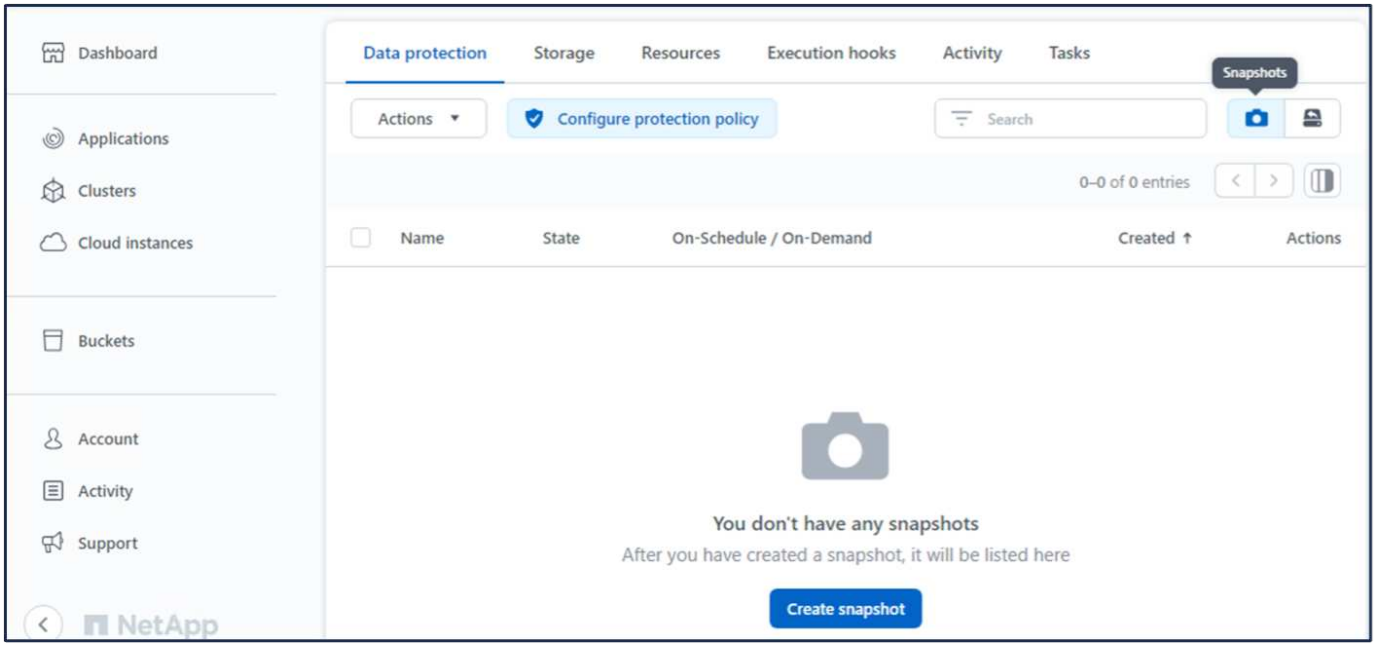
추가됩니다.

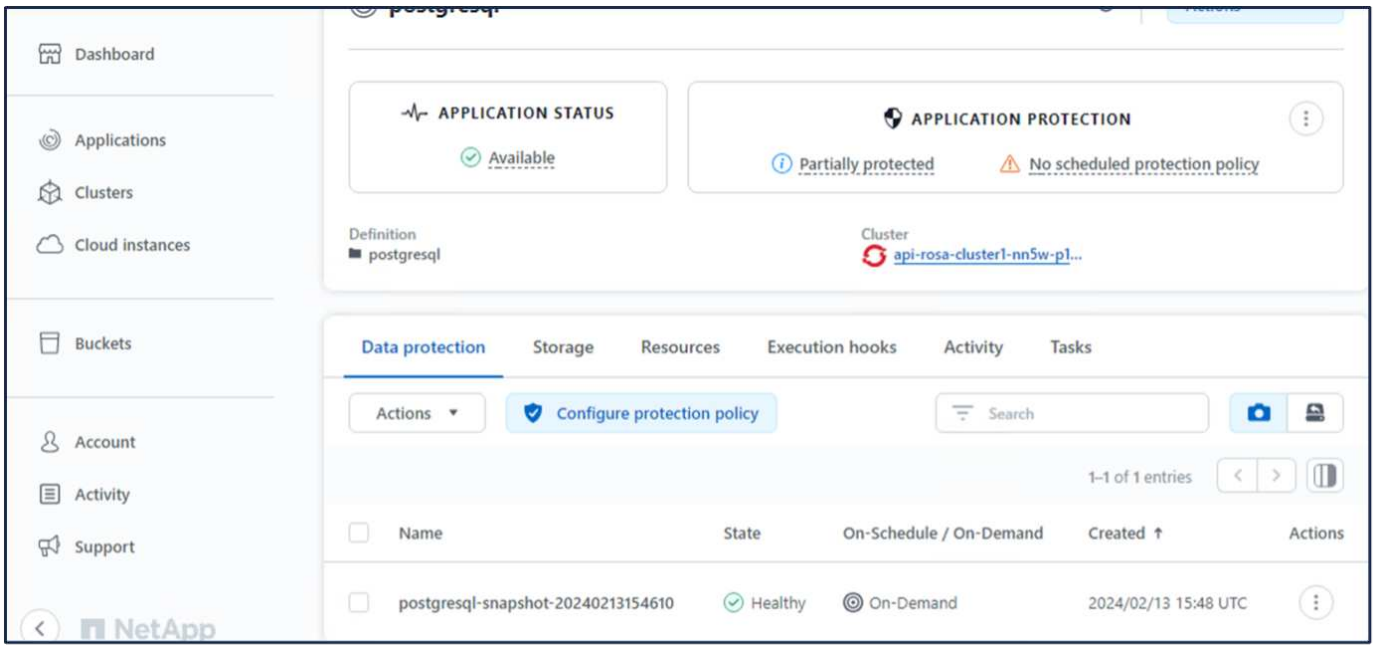


9. ACS를 사용하여 스냅샷을 만듭니다.

ACS에서 스냅샷을 만드는 방법은 여러 가지가 있습니다. 해당 페이지에서 애플리케이션을 선택하고, 애플리케이션의 세부 정보를 보여주는 스냅샷을 만들 수 있습니다. 스냅샷 만들기를 클릭하면 주문형 스냅샷을 만들거나 보호 정책을 구성할 수 있습니다.

*스냅샷 만들기*를 클릭하고, 이름을 지정하고, 세부 정보를 검토한 다음 *스냅샷*을 클릭하면 간단히 주문형 스냅샷을 만들 수 있습니다. 작업이 완료되면 스냅샷 상태가 정상으로 변경됩니다.





10. postgresql 애플리케이션에서 데이터베이스를 삭제합니다.

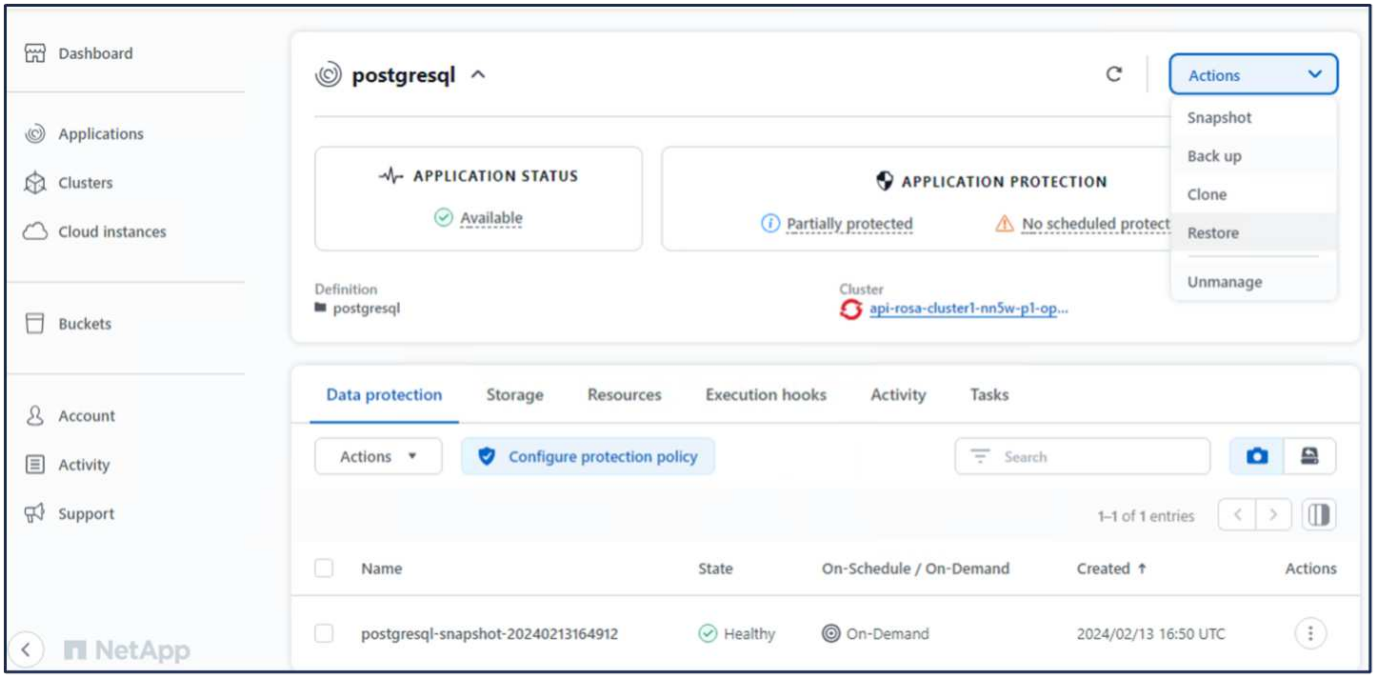
postgresql에 다시 로그인하여 사용 가능한 데이터베이스를 나열하고, 이전에 만든 데이터베이스를 삭제한 후 다시 나열하여 데이터베이스가 삭제되었는지 확인하세요.

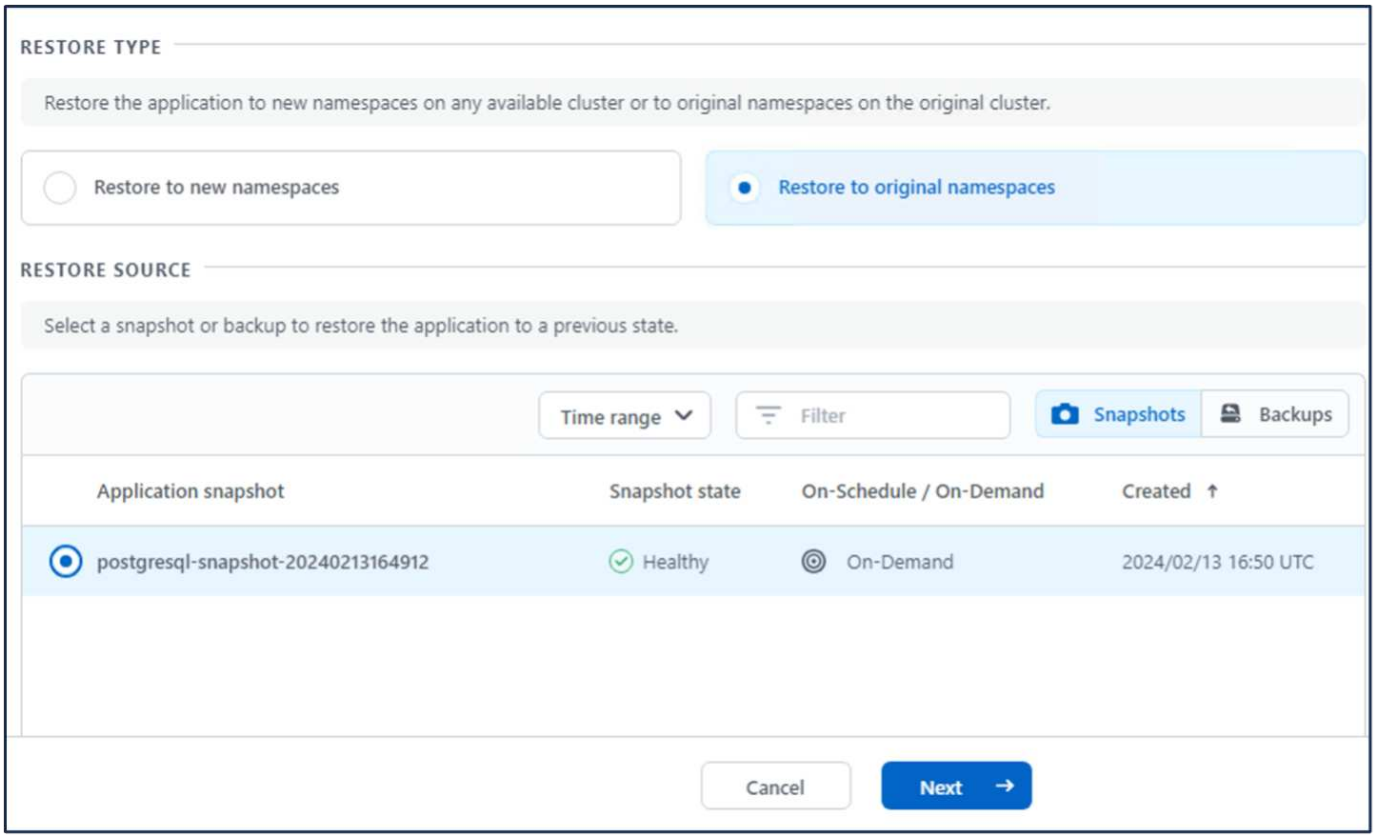
```
postgres=# \l
                                                 List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-------------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

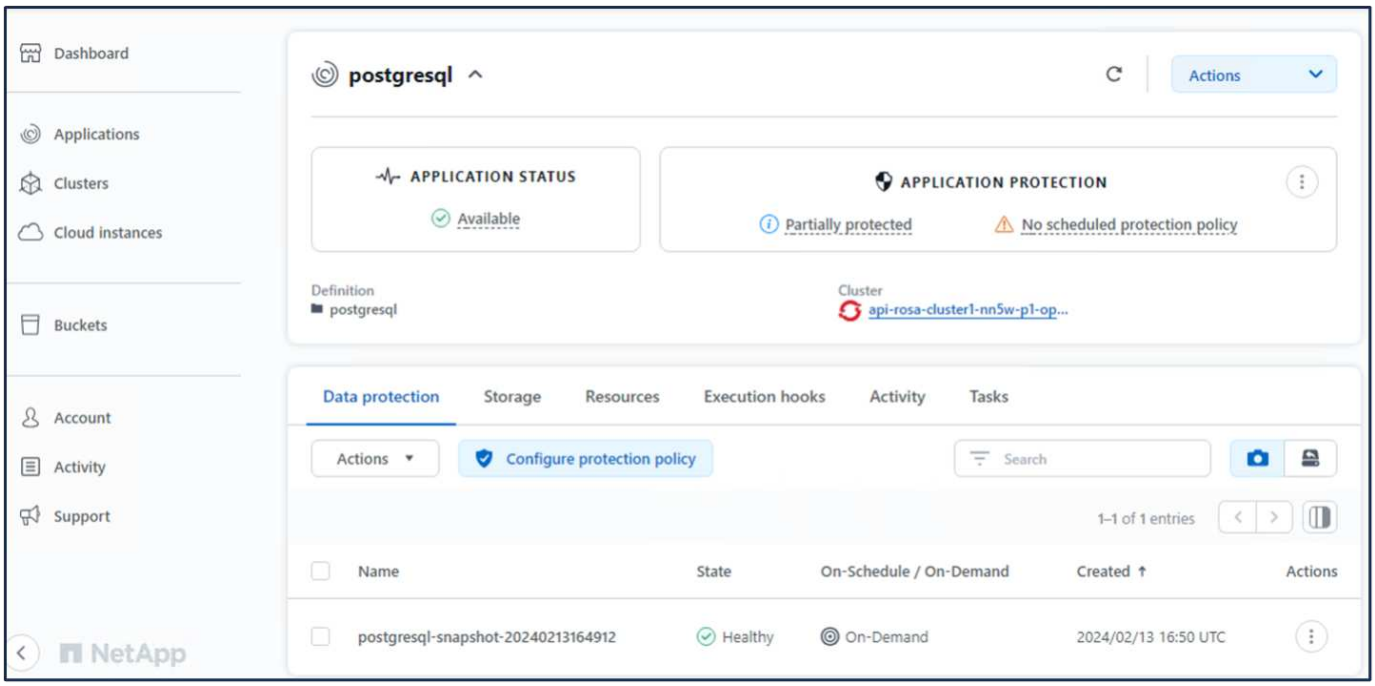
11. ACS를 사용하여 스냅샷에서 복원

스냅샷에서 애플리케이션을 복원하려면 ACS UI 랜딩 페이지로 이동하여 애플리케이션을 선택하고 복원을 선택하세요.

복원할 스냅샷이나 백업을 선택해야 합니다. (일반적으로, 귀하가 구성한 정책에 따라 여러 개가 생성됩니다). 다음 화면에서 적절한 선택을 한 후 *복원*을 클릭하세요. 스냅샷에서 복원된 후 애플리케이션 상태가 복원 중에서 사용 가능으로 변경됩니다.







12. 스냅샷에서 앱이 복원되었는지 확인하세요**

PostgreSQL 클라이언트에 로그인하면 이전에 가지고 있던 테이블과 테이블의 레코드를 볼 수 있습니다. 그게 다예요. 버튼 하나만 클릭하면 귀하의 애플리케이션이 이전 상태로 복원됩니다. Astra Control을 통해 우리는 고객에게 이렇게 쉬운 서비스를 제공합니다.

```
[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 페이지에서는 영구 저장소로 FSx ONTAP 사용하여 관리형 Red Hat OpenShift 클러스터의 컨테이너 워크로드에 대한 데이터 마이그레이션 옵션을 보여줍니다.

데이터 마이그레이션

AWS의 Red Hat OpenShift 서비스와 Amazon FSx for NetApp ONTAP (FSx ONTAP)은 AWS의 서비스 포트폴리오에 속합니다. FSx ONTAP 단일 AZ 또는 다중 AZ 옵션으로 제공됩니다. Multi-Az 옵션은 가용성 영역 장애로부터 데이터를 보호합니다. FSx ONTAP Trident 와 통합되어 ROSA 클러스터의 애플리케이션에 대한 영구 저장소를 제공할 수 있습니다.

Helm 차트를 사용하여 **FSx ONTAP** 과 **Trident** 통합

Amazon FSx ONTAP 과 ROSA 클러스터 통합

컨테이너 애플리케이션 마이그레이션에는 다음이 포함됩니다.

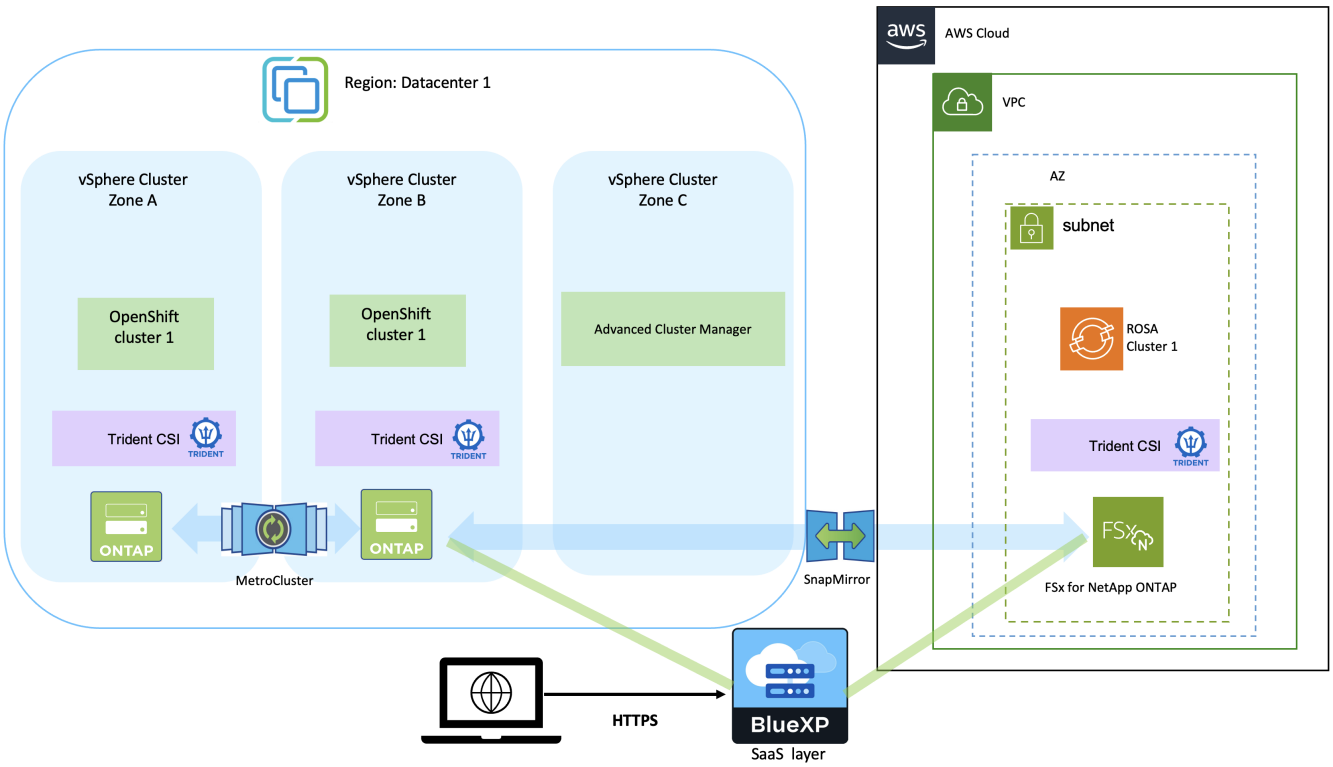
- 영구 볼륨: BlueXP 사용하여 이를 구현할 수 있습니다. 또 다른 옵션은 Trident Protect를 사용하여 온프레미스에서 클라우드 환경으로 컨테이너 애플리케이션을 마이그레이션하는 것입니다. 자동화는 같은 목적으로 사용될 수 있습니다.
- 애플리케이션 메타데이터: OpenShift GitOps(Argo CD)를 사용하여 이를 달성할 수 있습니다.

FSx ONTAP 사용하여 영구 저장소를 위한 **ROSA** 클러스터의 애플리케이션 장애 조치 및 장애 복구

다음 비디오는 BlueXP 와 Argo CD를 사용한 애플리케이션 장애 조치 및 장애 복구 시나리오를 보여줍니다.

ROSA 클러스터에서 애플리케이션의 장애 조치 및 장애 복구

OpenShift 컨테이너 워크로드를 위한 데이터 보호 및 마이그레이션 솔루션



Red Hat OpenShift 워크로드를 위한 추가 **NetApp** 하이브리드 멀티클라우드 솔루션

추가 솔루션

추가 솔루션은 다음과 같이 다른 섹션에서 사용할 수 있습니다.

Red Hat OpenShift Container 솔루션에 대해서는 다음을 참조하세요."여기" .

온프레미스 Red Hat OpenShift Virtualization 솔루션의 경우 다음을 참조하세요."여기" .