プロバイダー管理コンポーネントを備えたハイブリッドクラウド

NetApp public and hybrid cloud solutions

NetApp
August 18, 2025

This PDF was generated from https://docs.netapp.com/ja-jp/netapp-solutions-cloud/openshift/os-pm-solution.html on August 18, 2025. Always check docs.netapp.com for the latest.

目次

プロバイダー管理コンポーネントを備えたハイブリッドクラウド . . . 1
マネージド Red Hat OpenShift Container プラットフォーム ワークロードを備えたNetAppソリューション . . . 1
AWS 上でマネージド Red Hat OpenShift Container プラットフォームをデプロイおよび構成する . . . 1
Google Cloud NetApp Volumesを使用して Google Cloud に OpenShift Dedicatedをデプロイおよび構成する . . . 4
データ保護 . . . 6
バックアップ/バックアップからの復元 . . . 7
スナップショット/スナップショットからの復元 . . . 7
ブログ . . . 7
スナップショットを作成し、そこから復元するための詳細な手順 . . . 7
データ移行 . . . 22
データ マイグレーション . . . 23
Red Hat OpenShift ワークロード向けの追加のNetAppハイブリッド マルチクラウド ソリューション . . . 24
追加のソリューション . . . 24

プロバイダー管理コンポーネントを備えたハイブリッドクラウド

マネージド **Red Hat OpenShift Container** プラットフォーム ワークロードを備えた**NetApp**ソリューション

お客様は「クラウド生まれ」の場合もあれば、モダナイゼーションの過程で、一部のワークロードまたはすべてのワークロードをデータセンターからクラウドに移行する準備が整っている場合もあります。ワークロードを実行するために、クラウド内のプロバイダー管理の OpenShift コンテナとプロバイダー管理のNetAppストレージを使用することを選択できます。コンテナ ワークロードに適した本番環境を正常に構築するには、マネージド Red Hat OpenShift コンテナ クラスターをクラウドで計画して導入する必要があります。 NetApp は、3 つの主要なパブリック クラウドすべてにおいて、マネージド Red Hat ソリューション向けに完全に管理されたストレージ サービスを提供します。

- Amazon FSx for NetApp ONTAP (FSx ONTAP)*

FSx ONTAP は、AWS でのコンテナ展開にデータ保護、信頼性、柔軟性を提供します。 Trident は、顧客のステートフル アプリケーション用に永続的な FSx ONTAPストレージを使用する動的ストレージ プロビジョナーとして機能します。

ROSA は、コントロール プレーン ノードを複数のアベイラビリティ ゾーンに分散して HA モードで展開できるため、FSx ONTAPも、高可用性を提供し、AZ 障害から保護するマルチ AZ オプションを使用してプロビジョニングできます。

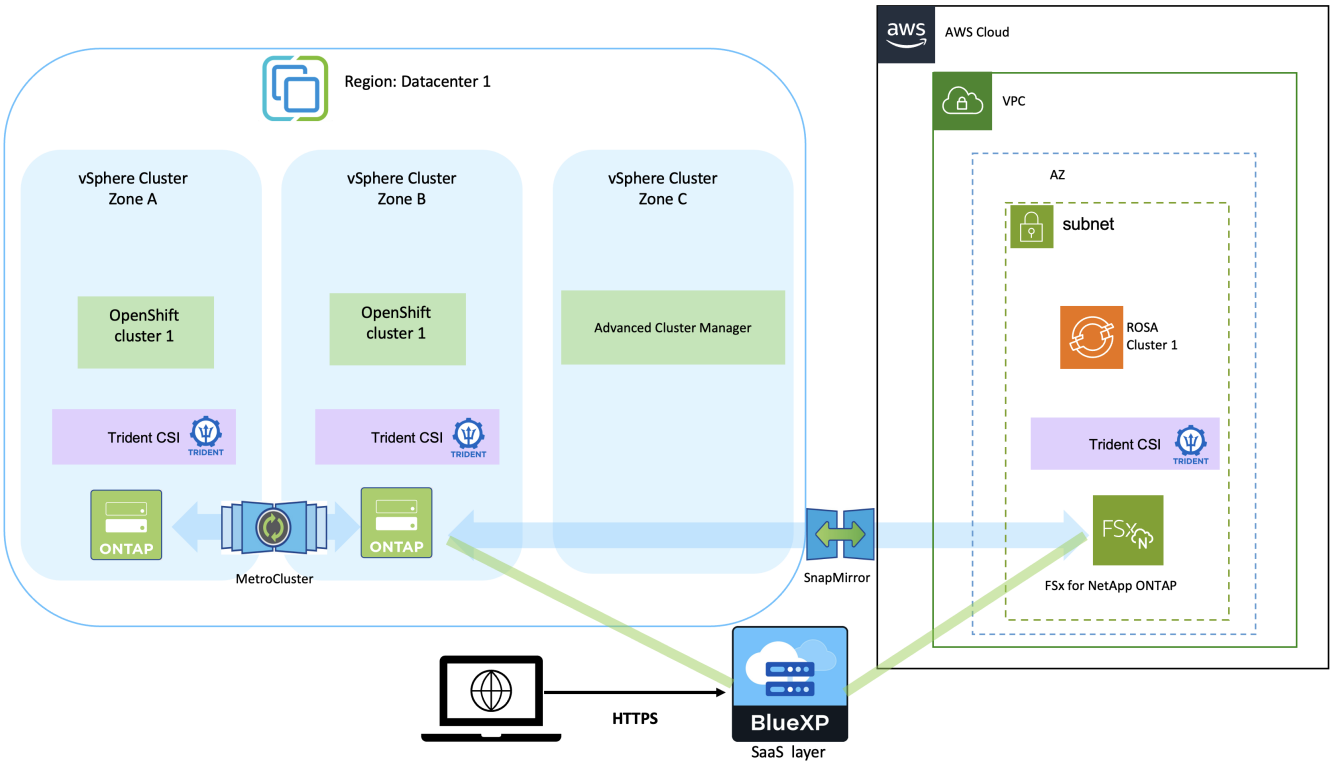
- Google Cloud NetApp Volumes*

Red Hat OpenShift Dedicated は、ハイブリッド クラウド全体でアプリケーションを迅速に構築、デプロイ、拡張できる、完全に管理されたアプリケーション プラットフォームです。 Google Cloud NetApp Volumes は永続ボリュームを提供し、ONTAP のエンタープライズ データ管理機能の完全なスイートを Google Cloud の OpenShift デプロイメントに提供します。

AWS 上でマネージド **Red Hat OpenShift Container** プラットフォームをデプロイおよび構成する

このセクションでは、AWS (ROSA) 上でマネージド Red Hat OpenShift クラスターをセットアップするための高レベルのワークフローについて説明します。これは、永続ボリュームを提供するためにTridentがストレージバックエンドとしてマネージドAmazon FSx for NetApp ONTAP (FSx ONTAP) を使用する方法を示しています。 BlueXPを使用して AWS に FSx ONTAPを導入する方法について詳細に説明します。また、ROSA クラスター上のステートフル アプリケーションのデータ保護と移行アクティビティを実行するためにBlueXPと OpenShift GitOps (Argo CD) を使用する方法についても詳しく説明します。

以下は、AWS にデプロイされ、バックエンドストレージとして FSx ONTAPを使用している ROSA クラスターを示す図です。



このソリューションは、AWS の 2 つの VPC 内の 2 つの ROSA クラスターを使用して検証されました。各 ROSA クラスターは、 Tridentを使用して FSx ONTAPと統合されました。 AWS に ROSA クラスターと FSx ONTAPをデプロイする方法はいくつかあります。このセットアップの概要説明には、使用された特定の方法に関するドキュメント リンクが提供されます。その他の方法については、以下の関連リンクを参照してください。"リソースセクション" 。

セットアッププロセスは次の手順に分けられます。

ROSAクラスタをインストールする

- 2 つの VPC を作成し、VPC 間に VPC ピアリング接続を設定します。
- 参照する"ここをクリックしてください。"ROSA クラスターのインストール手順については、こちらをご覧ください。

FSx ONTAPをインストールする

- BlueXPから VPC に FSx ONTAPをインストールします。参照する"ここをクリックしてください。"BlueXPアカウントを作成して開始するには。参照する"ここをクリックしてください。"FSx ONTAPをインストールするためのものです。参照する"ここをクリックしてください。"FSx ONTAP を管理するためのコネクタを AWS で作成します。
- AWS を使用して FSx ONTAPをデプロイします。参照する"ここをクリックしてください。"AWS コンソールを使用してデプロイします。

ROSA クラスターに**Trident**をインストールする **(Helm** チャートを使用**)**

- Helm チャートを使用して、ROSA クラスターにTridentをインストールします。ドキュメントリンクを参照してください: ここ。

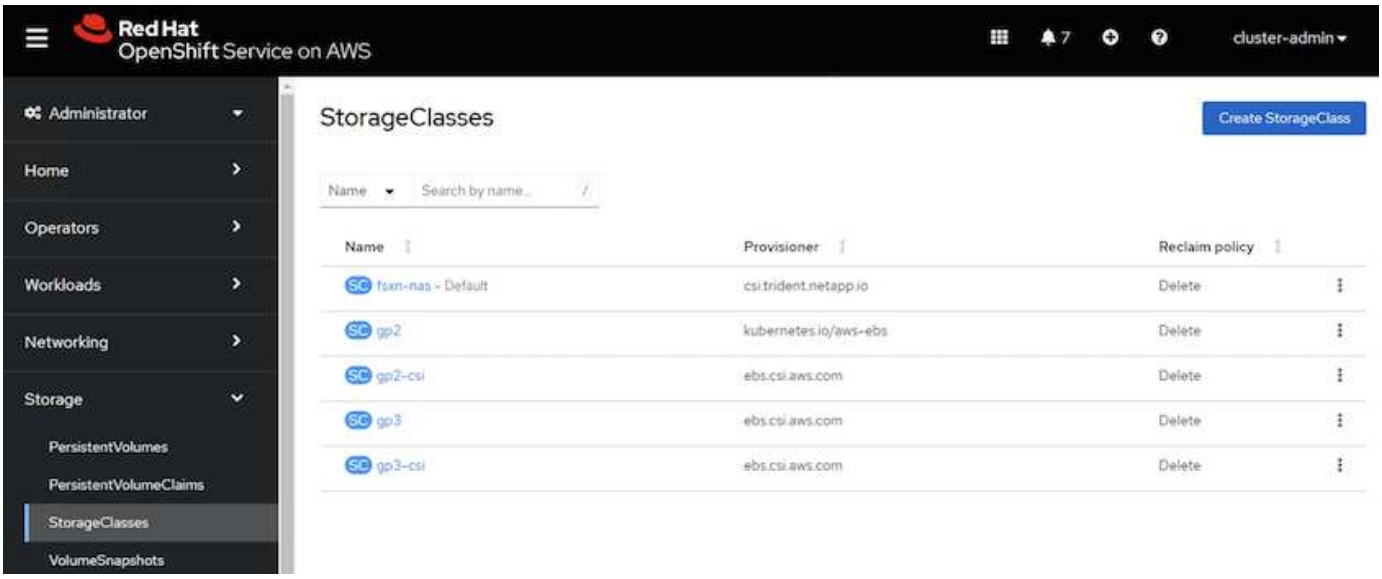
ROSA クラスタ向け FSx ONTAPとTridentの統合

OpenShift GitOps を利用すると、ApplicationSet を使用して ArgoCD に登録されたすべての管理対象クラスターにTrident CSI をデプロイできます。

```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を使用してバックエンドとストレージ クラスを作成する (FSx ONTAP用)

- 参照する"ここをクリックしてください。"バックエンドとストレージ クラスの作成の詳細については、こちらをご覧ください。
- OpenShift コンソールから、 Trident CSI を使用して FsxN 用に作成されたストレージ クラスをデフォルトにします。下のスクリーンショットを参照してください。



OpenShift GitOps (Argo CD) を使用してアプリケーションをデプロイする

- クラスターに OpenShift GitOps オペレーターをインストールします。説明書を参照してください"ここをクリックしてください。"。
- クラスターに新しい Argo CD インスタンスをセットアップします。説明書を参照してください"ここをクリックしてください。"。

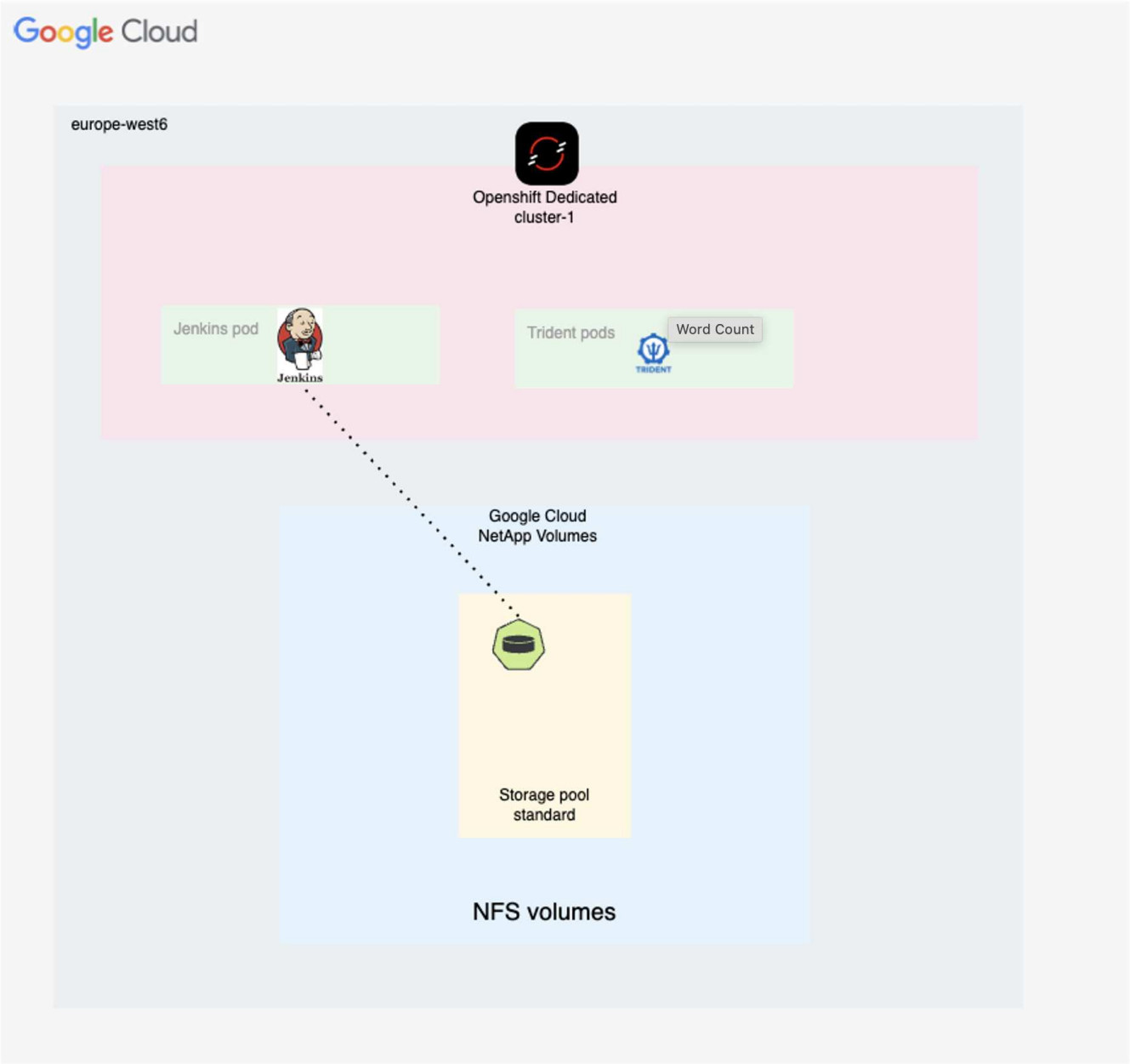
Argo CD のコンソールを開き、アプリを展開します。たとえば、Helm Chart を使用した Argo CD を使用して Jenkins アプリをデプロイできます。アプリケーションを作成するときに、次の詳細が指定されました: プロジェクト: デフォルト クラスター:'https://kubernetes.default.svc' (引用符なし) 名前空間: Jenkins Helm Chart の URL:'https://charts.bitnami.com/bitnami' （引用符なし）

Helm パラメータ: global.storageClass: fsxn-nas

Google Cloud NetApp Volumesを使用して Google Cloud に OpenShift Dedicated をデプロイおよび構成する

このセクションでは、Google Cloud プラットフォームで OpenShift Dedicated（OSD）クラスタを設定するための大まかなワークフローについて説明します。これは、 NetApp TridentがGoogle Cloud NetApp Volumes をストレージ バックエンドとして使用し、Kubernetes で実行されるステートフル アプリケーションに永続ボリュームを提供する様子を示しています。

以下は、Google Cloud にデプロイされ、バックエンド ストレージとしてNetApp Volumes を使用している OSD クラスタを示す図です。



セットアッププロセスは次の手順に分けられます。

Google Cloud に **OSD** クラスタをインストールする

- クラスターに既存の VPC を使用する場合は、OSD クラスター用に VPC、2 つのサブネット、クラウドルーター、および 2 つの GCP クラウド NAT を作成する必要があります。参照する"ここをクリックしてください。"手順についてはこちらをご覧ください。
- 参照する"ここをクリックしてください。"カスタマー クラウド サブスクリプション (CCS) 課金モデルを使用して GCP に OSD クラスターをインストールする手順。 OSD は Google Cloud Marketplace にも含まれています。 Google Cloud Marketplaceソリューションを使用してOSDをインストールする方法を示すビデオは、"ここをクリックしてください。" 。

Google Cloud NetApp Volumesを有効にする

- 参照する"ここをクリックしてください。"Google Cloud NetApp Volumesへのアクセスの設定方法については、こちらをご覧ください。ここまでのすべての手順に従ってください

- ストレージ プールを作成します。参照する"ここをクリックしてください。"Google Cloud NetApp Volumesでストレージ プールを設定する方法については、こちらをご覧ください。 OSD 上で実行されるステートフル Kubernetes アプリケーションのボリュームは、ストレージ プール内に作成されます。

OSD クラスターに**Trident**をインストールする **(Helm** チャートを使用**)**

- Helm チャートを使用して、OSD クラスターにTridentをインストールします。参照する"ここをクリックしてください。"Helm Chart のインストール方法については、こちらをご覧ください。ヘルムチャートは"ここをクリックしてください。"。

OSD クラスタ用の**NetApp Volumes** と**NetApp Trident**の統合

Tridentを使用してバックエンドとストレージ クラスを作成する (Google Cloud NetApp Volumes用)

- バックエンドの作成の詳細については、ここを参照してください。
- Kubernetes の現在のストレージ クラスのいずれかがデフォルトとしてマークされている場合は、ストレージ クラスを編集してその注釈を削除します。
- Trident CSI プロビジョナーを使用して、 NetAppボリューム用に少なくとも 1 つのストレージ クラスを作成します。アノテーションを使用して、ストレージ クラスの 1 つだけをデフォルトにします。これにより、PVC マニフェストで明示的に呼び出されていない場合でも、PVC はこのストレージ クラスを使用できるようになります。注釈付きの例を以下に示します。

```
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) を使用してアプリケーションをデプロイする

- クラスターに OpenShift GitOps オペレーターをインストールします。説明書を参照してください"ここをクリックしてください。"。
- クラスターに新しい Argo CD インスタンスをセットアップします。説明書を参照してください"ここをクリックしてください。"。

Argo CD のコンソールを開き、アプリを展開します。たとえば、Helm Chart を使用した Argo CD を使用して Jenkins アプリをデプロイできます。アプリケーションを作成するときに、次の詳細が指定されました: プロジェクト: デフォルト クラスター:'https://kubernetes.default.svc' (引用符なし) 名前空間: Jenkins Helm Chart の URL:'https://charts.bitnami.com/bitnami' （引用符なし）

データ保護

このページでは、 Astra Control Service を使用した Managed Red Hat OpenShift on

AWS (ROSA) クラスターのデータ保護オプションについて説明します。 Astra Control Service (ACS) は、クラスターを追加したり、クラスター上で実行されるアプリケーションを定義したり、アプリケーションに対応したデータ管理アクティビティを実行したりできる、使いやすいグラフィカル ユーザー インターフェイスを提供します。 ACS 機能には、ワークフローの自動化を可能にする API を使用してアクセスすることもできます。

Astra Control (ACS または ACC) を駆動するのはNetApp Tridentです。 Trident は、Red Hat OpenShift、EKS、AKS、SUSE Rancher、Anthos などの複数の種類の Kubernetes クラスターを、 FAS/ AFF、 ONTAP Select、CVO、Google Google Cloud NetApp Volumes、 Azure NetApp Files 、 Amazon FSx ONTAPなどのさまざまなフレーバーのNetApp ONTAPストレージと統合します。

このセクションでは、ACS を使用した次のデータ保護オプションの詳細について説明します。

- あるリージョンで実行され、別のリージョンに復元される ROSA アプリケーションのバックアップと復元を示すビデオ。
- ROSA アプリケーションのスナップショットと復元を示すビデオ。
- ROSA クラスター、 Amazon FSx ONTAP のインストール、 NetApp Tridentを使用してストレージバックエンドと統合する方法、ROSA クラスターに postgresql アプリケーションをインストールする方法、ACS を使用してアプリケーションのスナップショットを作成し、そこからアプリケーションを復元する方法を、手順ごとに詳しく説明します。
- ACS を使用して FSx ONTAPを搭載した ROSA クラスター上の MySQL アプリケーションのスナップショットを作成し、そこから復元する手順を詳細に示すブログです。

バックアップ/バックアップからの復元

次のビデオは、あるリージョンで実行されている ROSA アプリケーションのバックアップと、別のリージョンへの復元を示しています。

AWS 上の Red Hat OpenShift サービス向け FSx NetApp ONTAP

スナップショット/スナップショットからの復元

次のビデオでは、ROSA アプリケーションのスナップショットを取得し、その後スナップショットから復元する方法について説明します。

Amazon FSx ONTAPストレージを使用した Red Hat OpenShift Service on AWS (ROSA) クラスター上のアプリケーションのスナップショット/リストア

ブログ

- "Amazon FSxストレージを備えた ROSA クラスター上のアプリのデータ管理にAstra Control Service を使用する"

スナップショットを作成し、そこから復元するための詳細な手順

前提条件の設定

- "AWSアカウント"

- "Red Hat OpenShiftアカウント"
- IAMユーザー"適切な権限"ROSAクラスタを作成してアクセスする
- "AWS CLI"
- "ローザCLI"
- "OpenShift CLI"（oc）
- サブネットと適切なゲートウェイおよびルートを備えた VPC
- "ROSAクラスタがインストールされました"VPCに
- "Amazon FSx ONTAP"同じVPC内に作成
- ROSAクラスタへのアクセス"OpenShift ハイブリッドクラウドコンソール"

次のステップ

1. 管理者ユーザーを作成し、クラスターにログインします。
2. クラスターの kubeconfig ファイルを作成します。
3. クラスターにTridentをインストールします。
4. Trident CSI プロビジョナーを使用して、バックエンド、ストレージ クラス、スナップショット クラスの構成を作成します。
5. クラスターに postgresql アプリケーションをデプロイします。
6. データベースを作成し、レコードを追加します。
7. クラスターを ACS に追加します。
8. ACS でアプリケーションを定義します。
9. ACS を使用してスナップショットを作成します。
10. postgresql アプリケーション内のデータベースを削除します。
11. ACS を使用してスナップショットから復元します。
12. アプリがスナップショットから復元されたことを確認します。

1.管理者ユーザーを作成し、クラスターにログインします

次のコマンドで管理者ユーザーを作成して、ROSA クラスターにアクセスします: (インストール時に管理者ユーザーを作成しなかった場合にのみ、管理者ユーザーを作成する必要があります)

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

このコマンドは次のような出力を提供します。クラスタにログインするには、 `oc login`出力に提供されるコマンド。

```
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
```

トークンを使用してクラスターにログインすることもできます。クラスターの作成時にすでに admin-user を作成している場合は、admin-user の認証情報を使用して Red Hat OpenShift Hybrid Cloud コンソールからクラスターにログインできます。次に、ログインしているユーザーの名前が表示されている右上隅をクリックすると、 `oc login`コマンドラインのコマンド (トークン ログイン)。

2.クラスター用の **kubeconfig** ファイルを作成する

手順に従ってください"ここをクリックしてください。"ROSA クラスターの kubeconfig ファイルを作成します。この kubeconfig ファイルは、後でクラスターを ACS に追加するときに使用されます。

3.クラスターに**Trident**をインストールする

ROSA クラスターにTrident (最新バージョン) をインストールします。これを行うには、以下のいずれかの手順に従います。"ここをクリックしてください。" 。クラスターのコンソールから helm を使用してTridentをインストールするには、まずTridentというプロジェクトを作成します。

次に、開発者ビューから Helm チャート リポジトリを作成します。 URLフィールドでは `'https://netapp.github.io/trident-helm-chart'`。次に、 Tridentオペレーターの helm リリースを作成します。

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

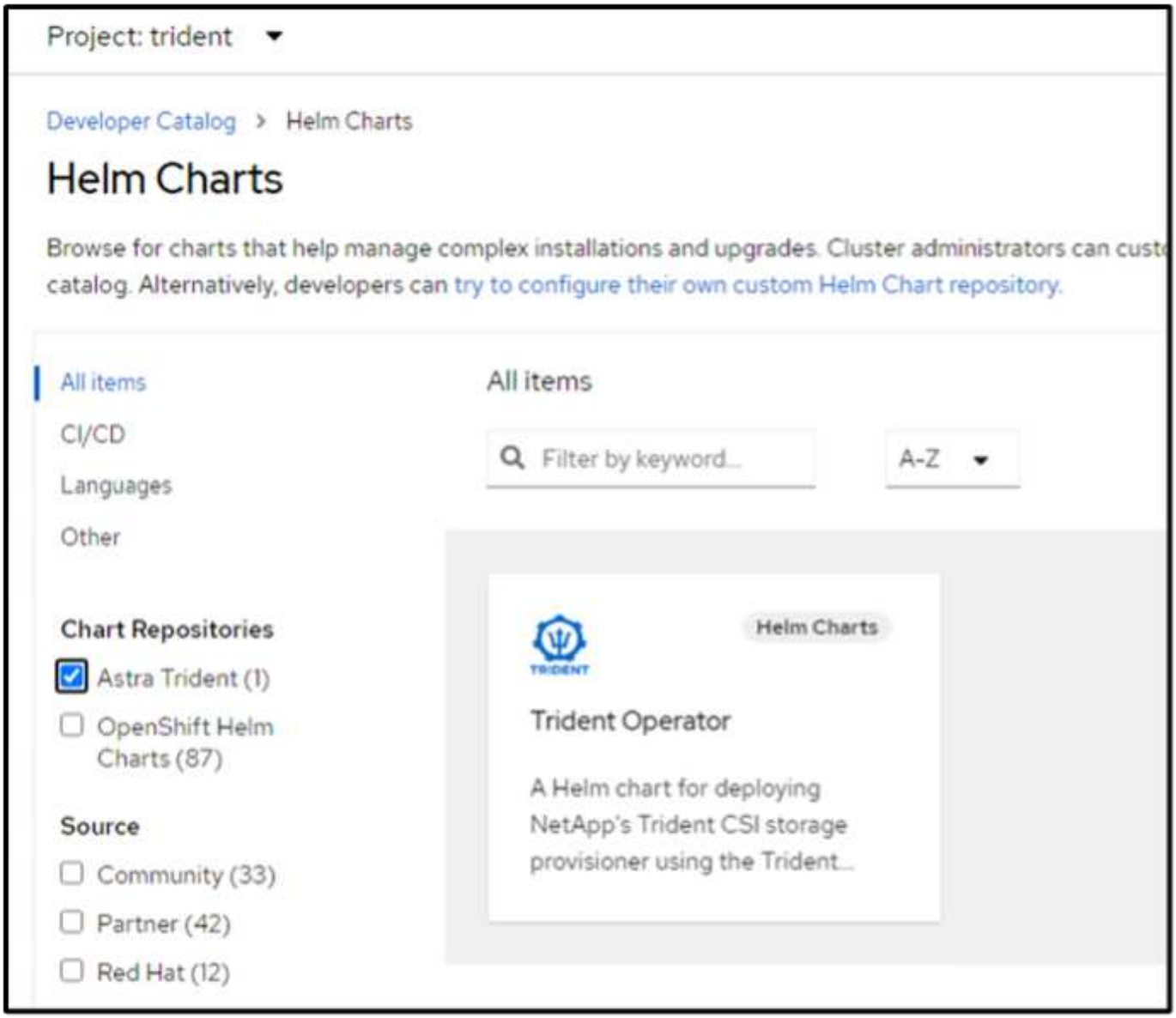
NetApp Astra Trident

A description for the Helm Chart repository.

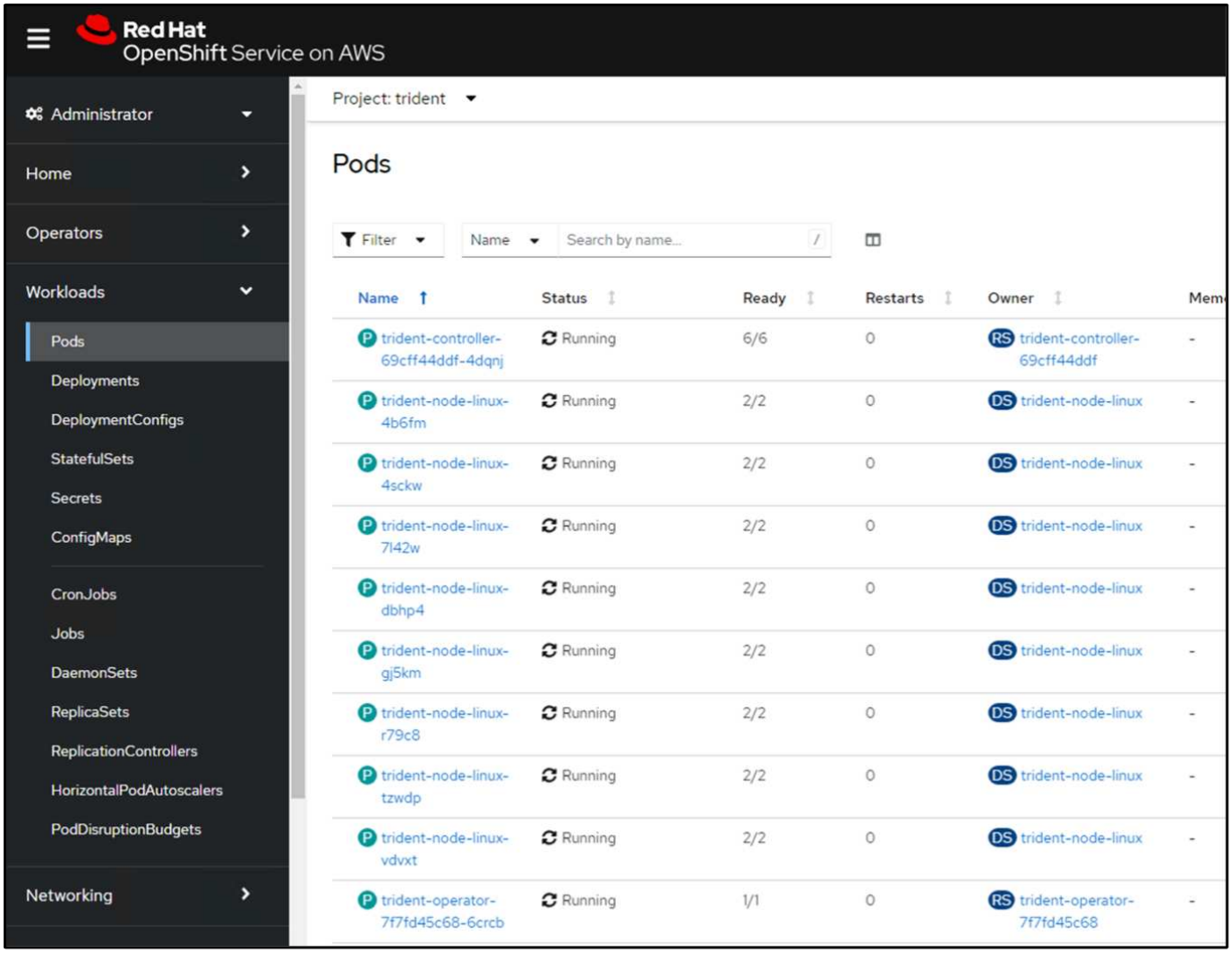
[ ] Disable usage of the repo in the developer catalog.

URL *

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



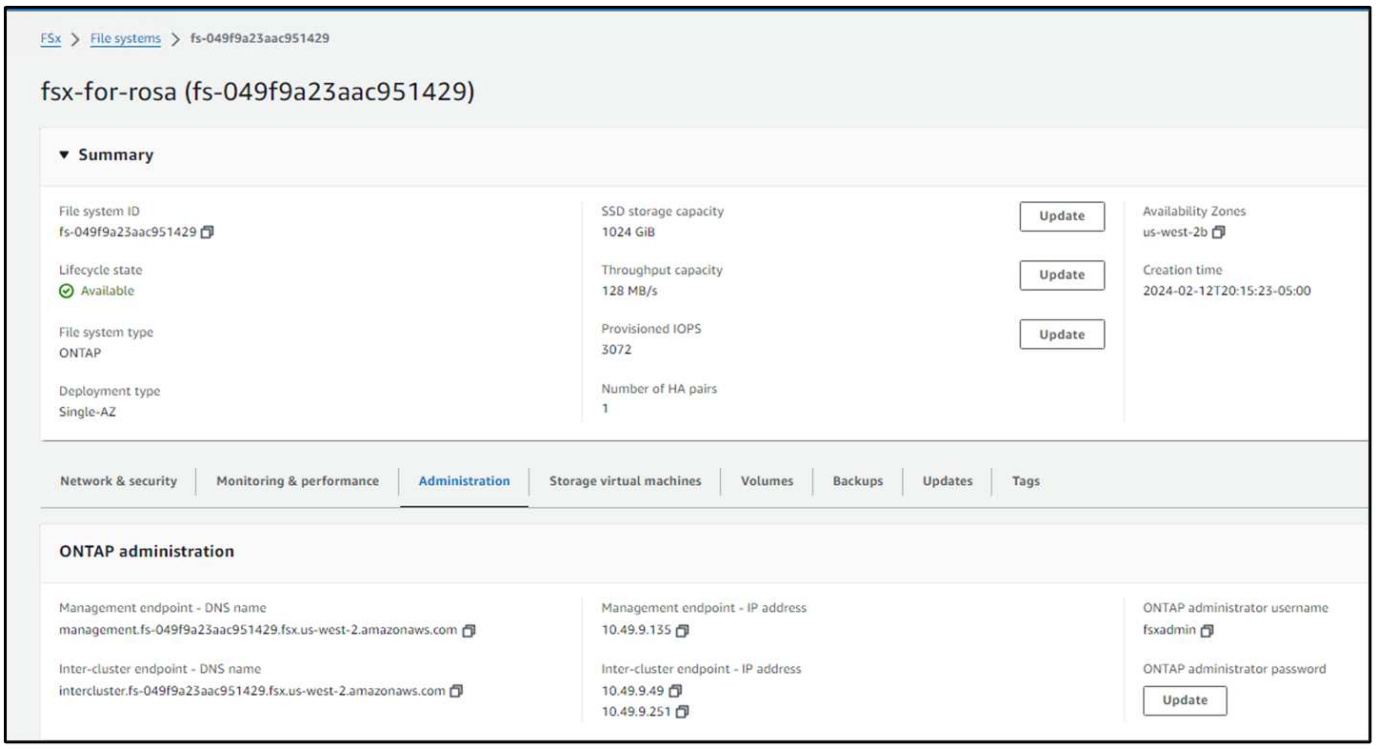
コンソールの管理者ビューに戻り、trident プロジェクト内のポッドを選択して、すべての trident ポッドが実行されていることを確認します。



4.Trident CSI プロビジョナーを使用して、バックエンド、ストレージ クラス、スナップショット クラスの構成を作成します

以下に示す yaml ファイルを使用して、トライデント バックエンド オブジェクト、ストレージ クラス オブジェクト、および Volumesnapshot オブジェクトを作成します。バックエンドの設定 yaml には、作成したAmazon FSx ONTAPファイルシステムの認証情報、管理 LIF、およびファイルシステムの vserver 名を必ず指定してください。これらの詳細を取得するには、 Amazon FSxの AWS コンソールにアクセスし、ファイルシステムを選択して、[管理] タブに移動します。また、更新をクリックしてパスワードを設定するには、`fsxadmin`ユーザー。

> コマンドラインを使用してオブジェクトを作成することも、ハイブリッド クラウド コンソールから yaml ファイルを使用してオブジェクトを作成することもできます。



- Tridentバックエンド構成**

```
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
```

ストレージクラス

```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
```

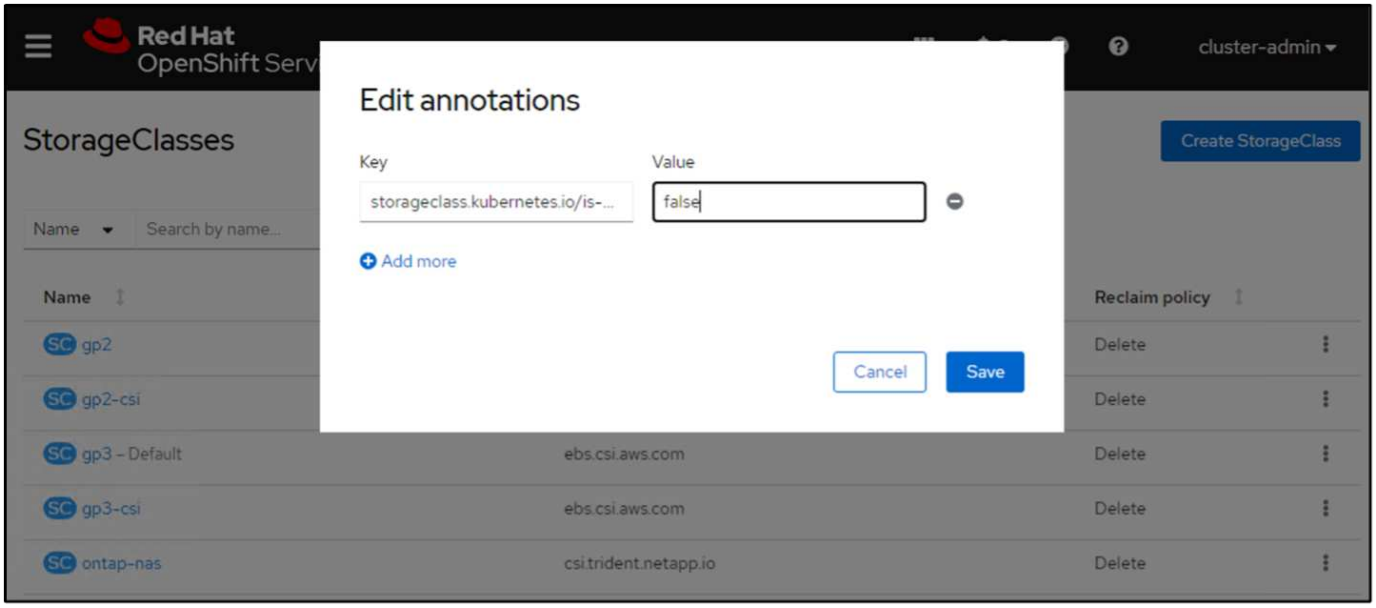
スナップショットクラス

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

以下のコマンドを発行して、バックエンド、ストレージ クラス、および trident-snapshotclass オブジェクトが作成されていることを確認します。

```
[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]$
```

この時点で行う必要がある重要な変更は、後でデプロイする postgresql アプリがデフォルトのストレージ クラスを使用できるように、gp3 ではなく ontap-nas をデフォルトのストレージ クラスとして設定することです。クラスターの Openshift コンソールで、[Storage] の下にある [StorageClasses] を選択します。現在のデフォルト クラスのアノテーションを false に編集し、ontap-nas ストレージ クラスのアノテーション storageclass.kubernetes.io/is-default-class を true に設定して追加します。



5.クラスターに **PostgreSQL** アプリケーションをデプロイする

次のようにして、コマンドラインからアプリケーションをデプロイできます。

```
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]$
```

アプリケーション ポッドが実行されていない場合は、セキュリティ コンテキストの制約によりエラーが発生している可能性があります。

```
[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]$
```

編集してエラーを修正してください `runAsUser`そして `fsGroup`フィールド `statefuleset.apps/postgresql`出力にあるuidを持つオブジェクト `oc get project`以下のようにコマンドを実行します。

```
[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 アプリが実行され、 Amazon FSx ONTAPストレージによってバックアップされた永続ボリュームを使用している必要があります。

```
[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]$
```

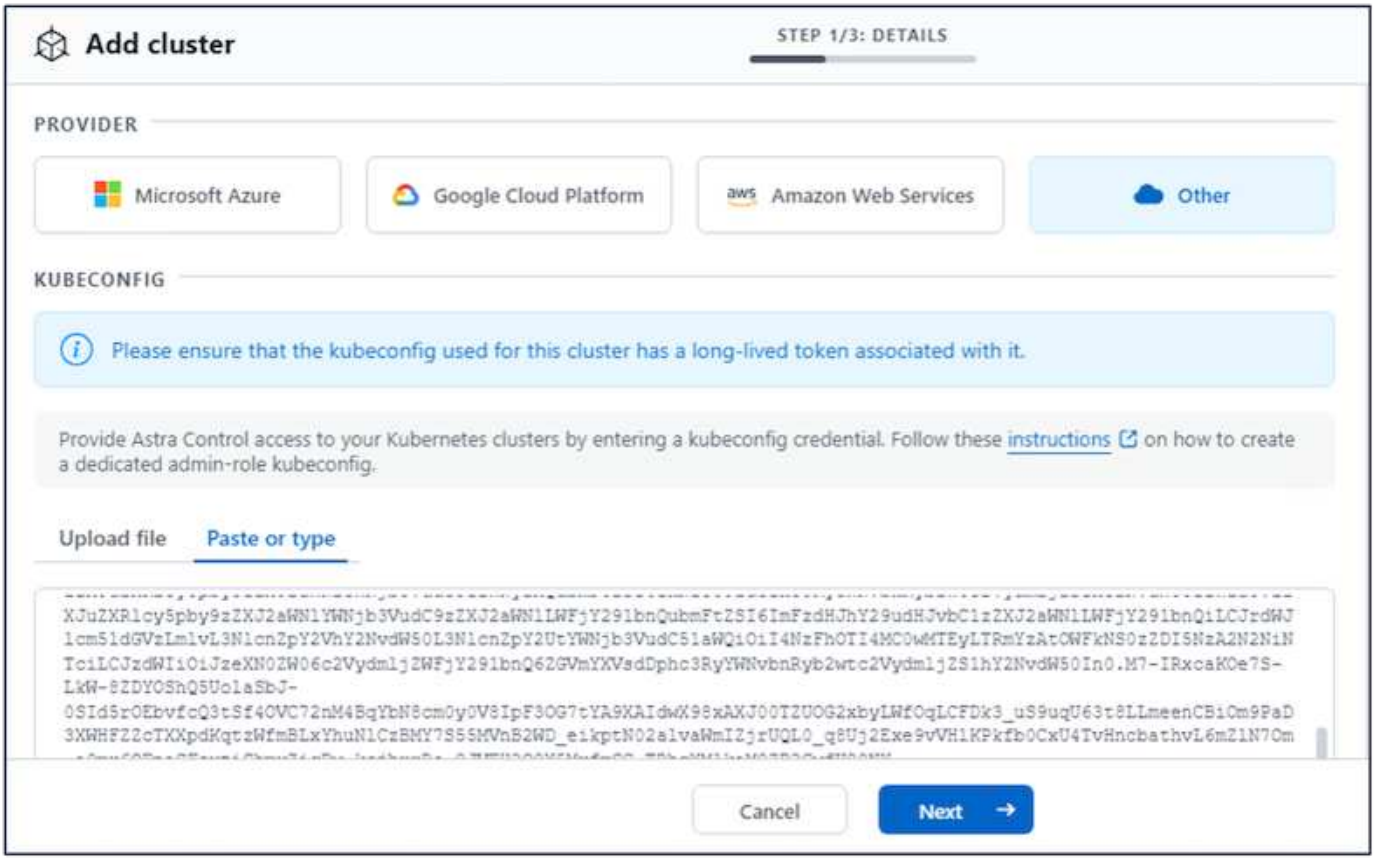
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

6.データベースを作成し、レコードを追加する

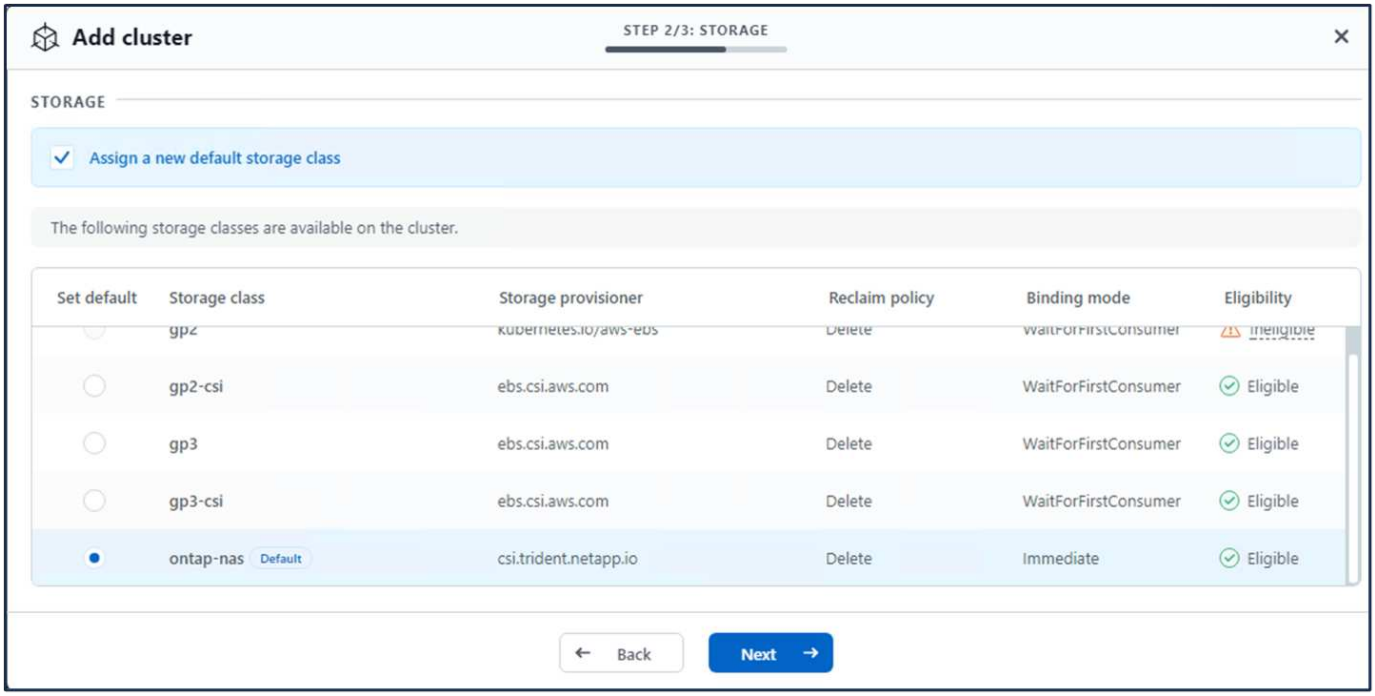
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1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

ACS にログインします。クラスターを選択し、「追加」をクリックします。 「その他」を選択し、kubeconfig ファイルをアップロードまたは貼り付けます。



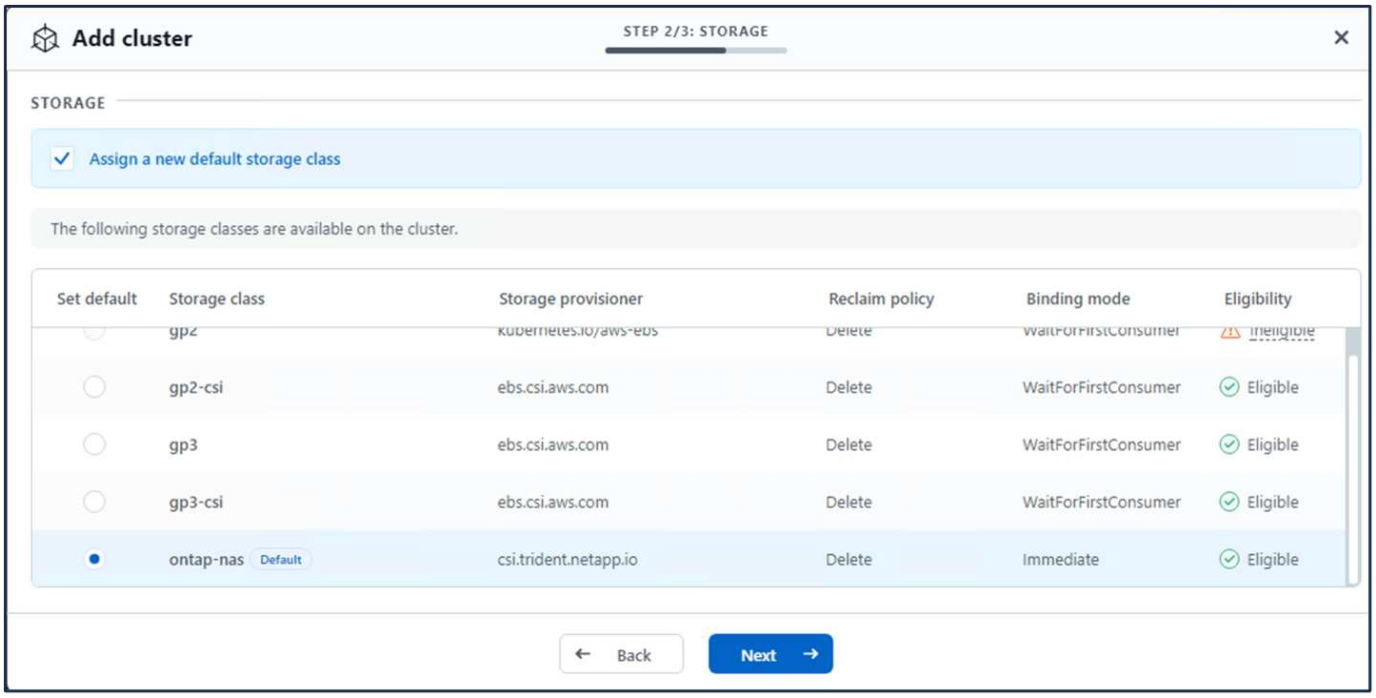
[次へ] をクリックし、ACS のデフォルトのストレージ クラスとして ontap-nas を選択します。 [次へ] をクリックし、詳細を確認してクラスターを [追加] します。



8.ACSでアプリケーションを定義する

ACS で postgresql アプリケーションを定義します。ランディング ページから、アプリケーション、定義 を選択し、適切な詳細を入力します。 *次へ*を数回クリックし、詳細を確認して*定義*をクリックします。アプリ

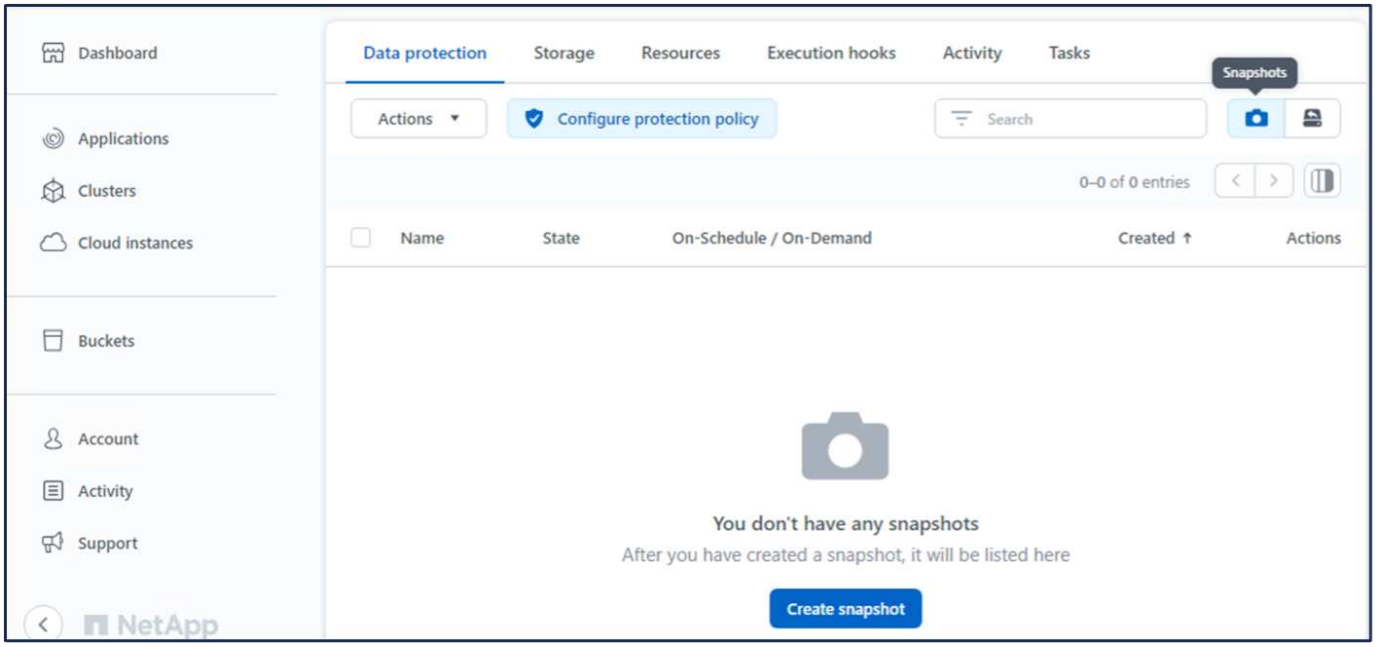
ケーションが ACS に追加されます。

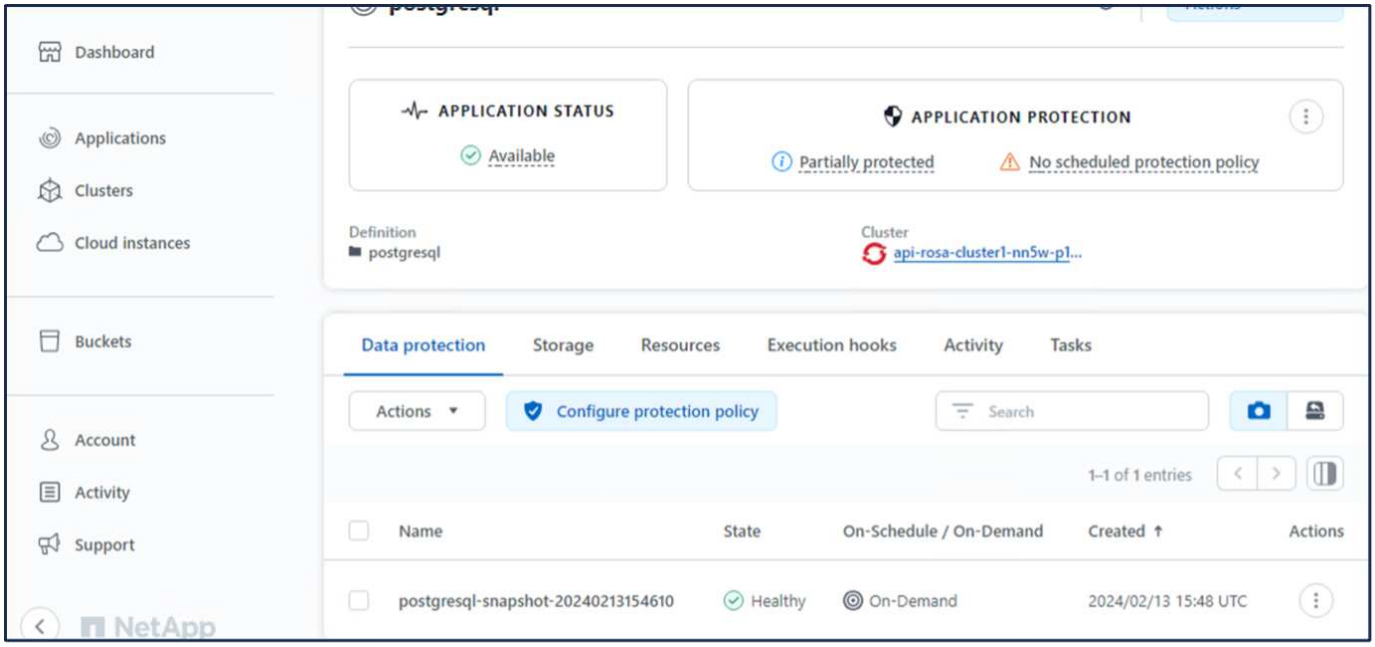


9.ACS を使用してスナップショットを作成する

ACS でスナップショットを作成する方法は多数あります。アプリケーションの詳細を表示するページからアプリケーションを選択し、スナップショットを作成できます。 「スナップショットの作成」をクリックすると、オンデマンドのスナップショットを作成したり、保護ポリシーを構成したりできます。

*スナップショットの作成*をクリックし、名前を入力して詳細を確認し、*スナップショット*をクリックするだけで、オンデマンドのスナップショットを作成できます。操作が完了すると、スナップショットの状態は「正常」に変わります。





10.PostgreSQLアプリケーションでデータベースを削除します

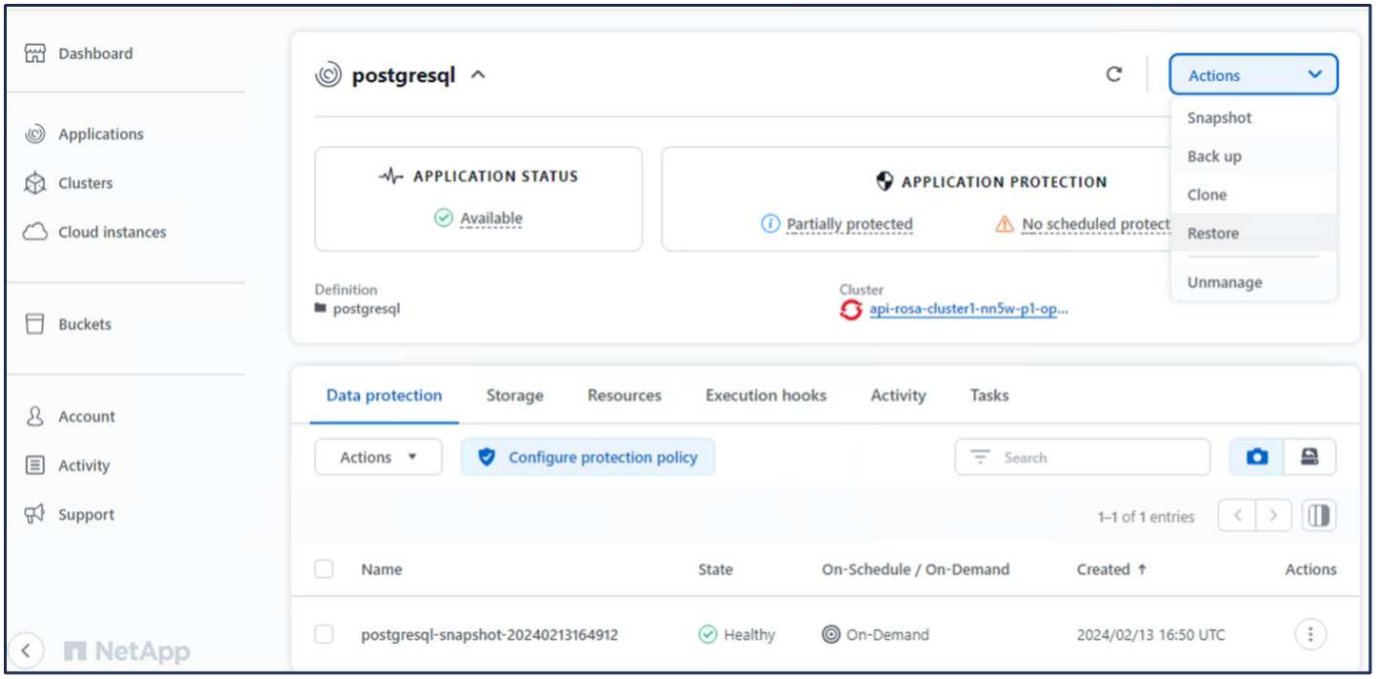
postgresql に再度ログインし、利用可能なデータベースを一覧表示し、以前に作成したデータベースを削除して再度一覧表示し、データベースが削除されたことを確認します。

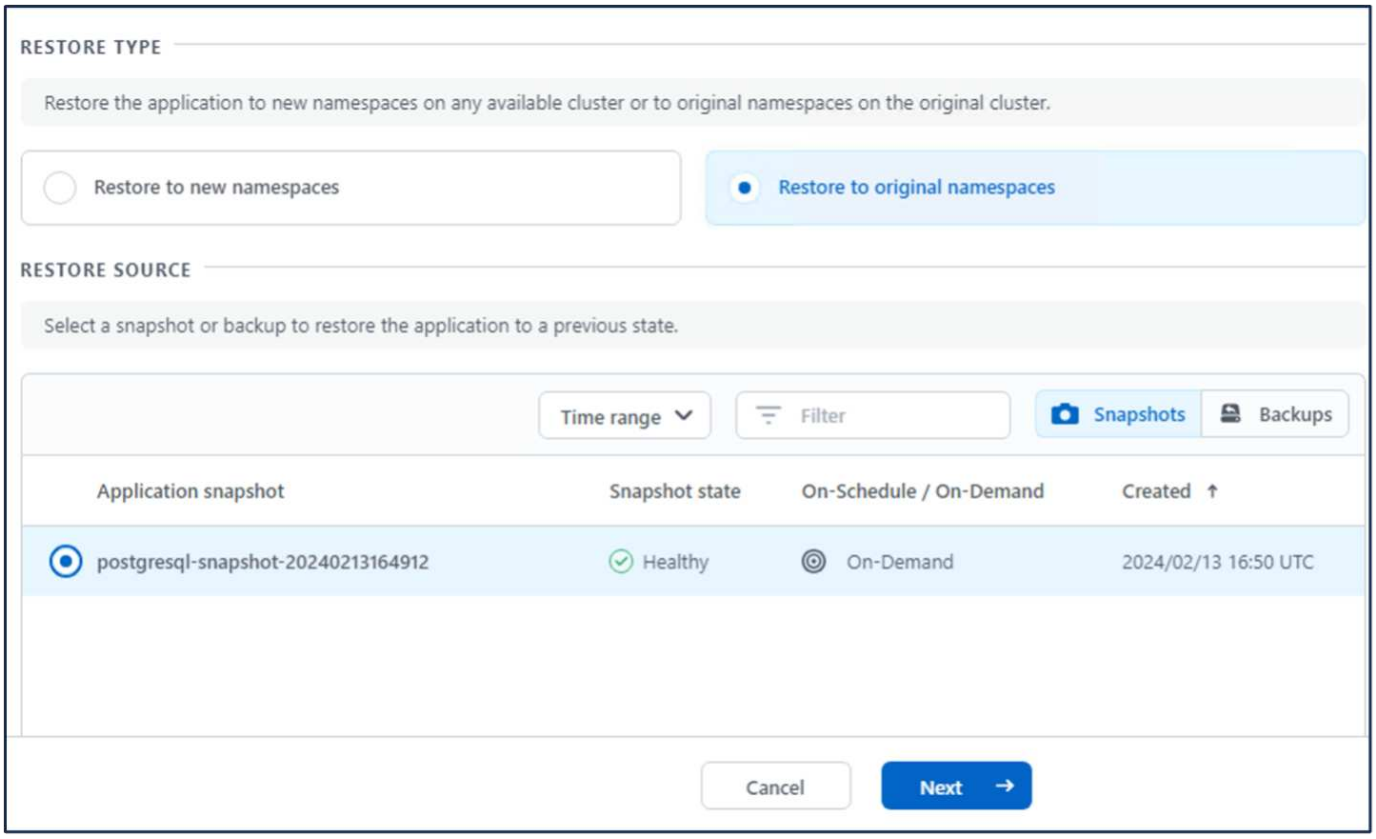
```
postgres=# \l
                                                List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------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

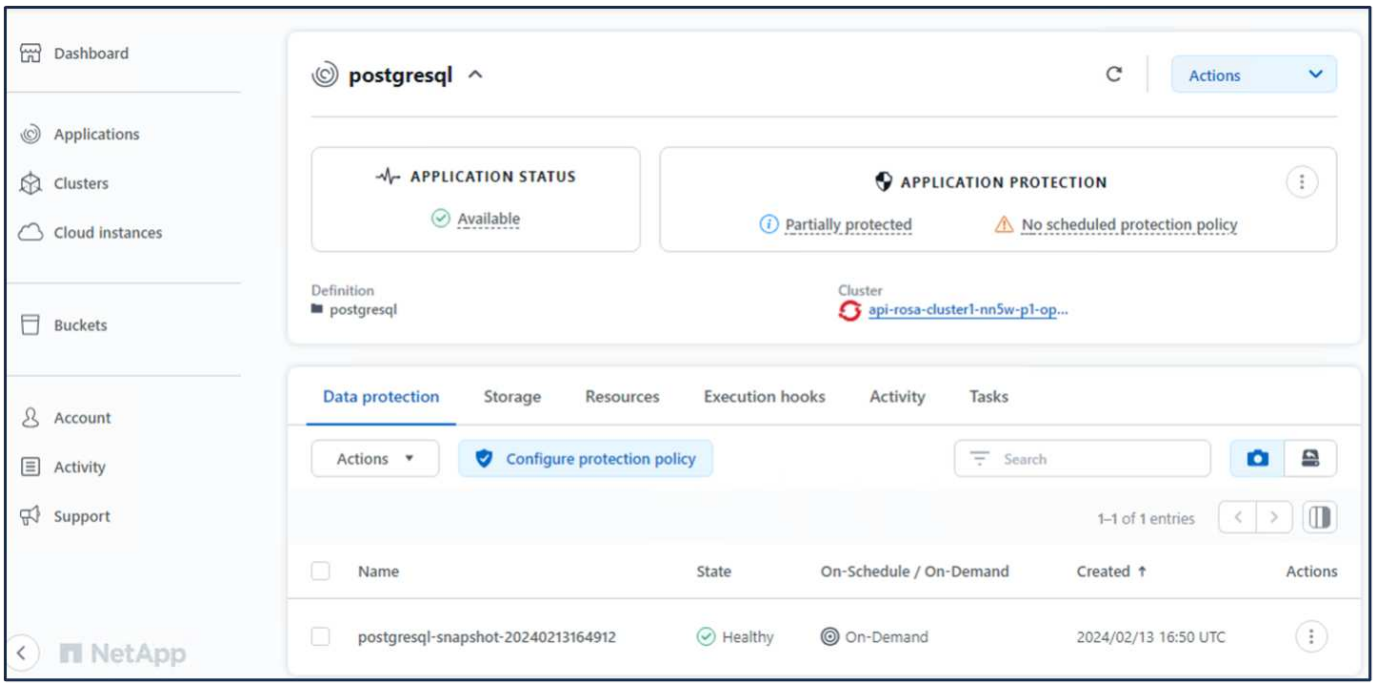
11.ACS を使用してスナップショットから復元する

スナップショットからアプリケーションを復元するには、ACS UI ランディング ページに移動し、アプリケー

ションを選択して [復元] を選択します。復元するスナップショットまたはバックアップを選択する必要があります。 (通常は、設定したポリシーに基づいて複数作成されます)。次のいくつかの画面で適切な選択を行い、「復元」をクリックします。スナップショットから復元されると、アプリケーションのステータスは「復元中」から「使用可能」に変わります。







12.アプリがスナップショットから復元されたことを確認します

postgresql クライアントにログインすると、以前持っていたテーブルとテーブル内のレコードが表示されるはずです。それでおしまい。ボタンをクリックするだけで、アプリケーションが以前の状態に復元されます。Astra Control を使用すると、お客様にとってそれがいかに簡単になります。

```
[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)
```

データ移行

このページでは、永続ストレージとして FSx ONTAPを使用するマネージド Red Hat OpenShift クラスター上のコンテナ ワークロードのデータ移行オプションについて説明します。

データ マイグレーション

AWS 上の Red Hat OpenShift サービスとAmazon FSx for NetApp ONTAP (FSx ONTAP) は、AWS のサービスポートフォリオの一部です。 FSx ONTAP は、シングル AZ またはマルチ AZ オプションで利用できます。 Multi-Az オプションは、アベイラビリティーゾーンの障害からのデータ保護を提供します。 FSx ONTAP をTridentと統合して、ROSA クラスター上のアプリケーションに永続的なストレージを提供できます。

Helm チャートを使用した **FSx ONTAP**と**Trident**の統合

ROSA クラスターとAmazon FSx ONTAP の統合

コンテナ アプリケーションの移行には次の作業が含まれます。

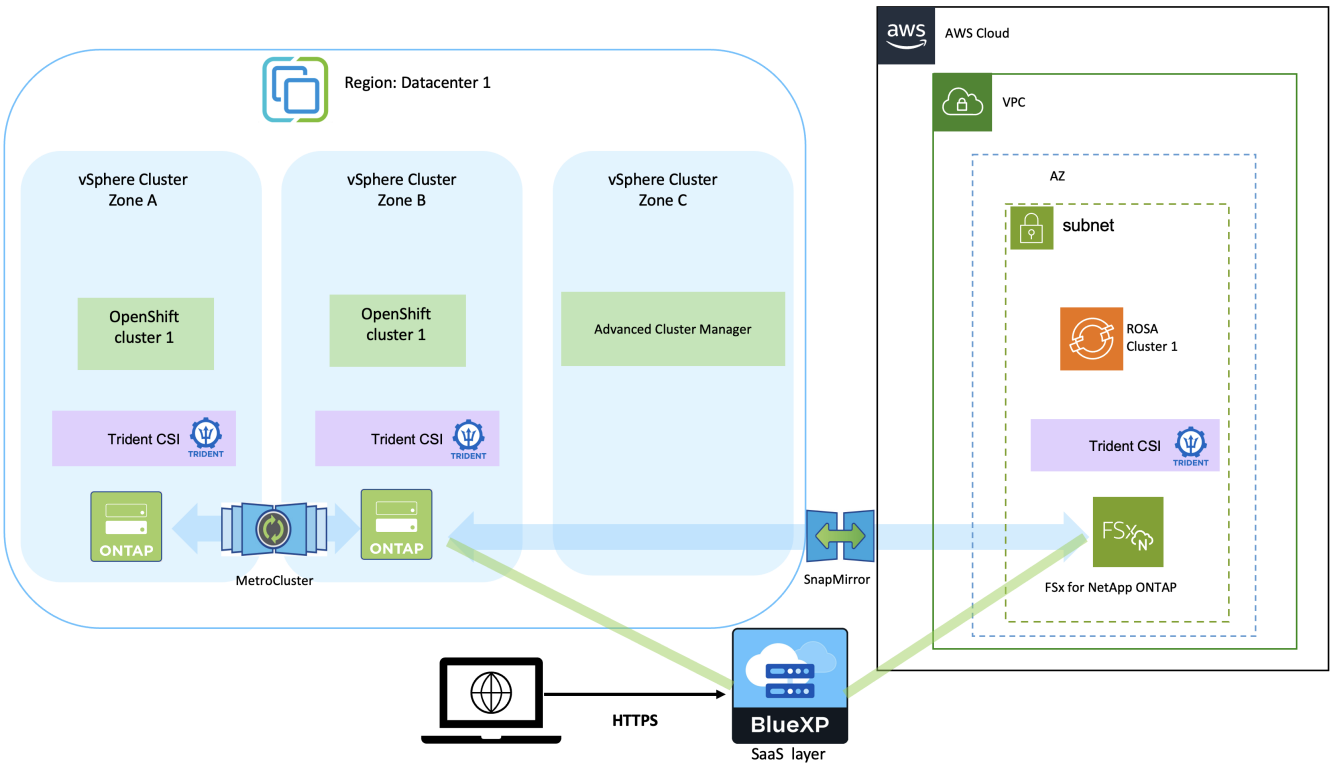
- 永続ボリューム: これはBlueXPを使用して実現できます。もう 1 つのオプションは、 Trident Protect を使用して、オンプレミスからクラウド環境へのコンテナ アプリケーションの移行を処理することです。同じ目的で自動化を使用することもできます。
- アプリケーション メタデータ: これは OpenShift GitOps (Argo CD) を使用して実現できます。

永続ストレージとして **FSx ONTAP**を使用した **ROSA** クラスタ上のアプリケーションのフェイルオーバーとフェイルバック

次のビデオは、 BlueXPと Argo CD を使用したアプリケーションのフェールオーバーとフェールバックのシナリオのデモです。

ROSA クラスタ上のアプリケーションのフェイルオーバーとフェイルバック

OpenShift Container ワークロード向けのデータ保護および移行ソリューション



Red Hat OpenShift ワークロード向けの追加のNetAppハイブリッド マルチクラウド ソリューション

追加のソリューション

追加のソリューションは、次の他のセクションで利用できます。

Red Hat OpenShift Container ソリューションについては、"ここをクリックしてください。" 。

オンプレミスのRed Hat OpenShift Virtualizationソリューションについては、"ここをクリックしてください。" 。

著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、http://www.netapp.com/TMに記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。