プロバイダ管理コンポーネントを使用したハイブリッドクラウド

NetApp Solutions

NetApp
April 10, 2024

This PDF was generated from https://docs.netapp.com/ja-jp/netapp-solutions/rhhc/rhhc-pm-solution.html on April 10, 2024. Always check docs.netapp.com for the latest.

目次

Red Hat OpenShift Containerワークロード向けのネットアップハイブリッドマルチクラウドソリューション 1
- 概要 1
- NetApp解決策 とAWS上のマネージドRed Hat OpenShift Containerプラットフォームのワークロード 3
- AWSにマネージドRed Hat OpenShift Containerプラットフォームを導入して設定します 4
- データ保護 7
- データ移行 23

Red Hat OpenShift Containerワークロード向けのネットアップハイブリッドマルチクラウドソリューション

概要

ネットアップでは、従来型エンタープライズアプリケーションを最新化し、Kubernetesを中心に構築されたコンテナとオーケストレーションプラットフォームを使用して新しいアプリケーションを構築するお客様が大幅に増えています。Red Hat OpenShift Container Platformは、多くのお客様に採用されている例の1つです。

企業内でコンテナを採用するお客様がますます増えています。ネットアップは、ステートフルアプリケーションの永続的ストレージのニーズに加え、データ保護、データセキュリティ、データ移行などの従来のデータ管理のニーズにも応えることができます。しかし、これらのニーズは、さまざまな戦略、ツール、方法を使用して満たしています。

- NetApp ONTAP **ベースのストレージオプションを次に示します。コンテナとKubernetes環境にセキュリティ、データ保護、信頼性、柔軟性を提供します。
  - オンプレミスの自己管理型ストレージ：
- ネットアップファブリック接続ストレージ（FAS ）、ネットアップオールフラッシュFAS アレイ（AFF ）、ネットアップオールSANアレイ（ASA ）、ONTAP Select
  - オンプレミスのプロバイダ管理ストレージ：
- NetApp Keystone が提供するストレージサービス（STaaS）
  - クラウド内の自己管理型ストレージ：
- NetApp Cloud Volumes ONTAP（CVO）は、ハイパースケーラに自己管理型ストレージを提供します
  - クラウド上のプロバイダが管理するストレージ：
- Cloud Volumes Service for Google Cloud（CVS）、Azure NetApp Files （ANF）、Amazon FSx for NetApp ONTAP は、ハイパースケーラにフルマネージドストレージを提供します

ONTAP feature highlights

Storage Administration

- Multi-tenancy
- FlexVol & FlexGroup
- LUN
- Quotas
- ONTAP CLI & API
- System Manager & BlueXP

Performance & Scalability

- FlexCache
- FlexClone
- nconnect, session trunking, multipathing
- Scale-out clusters

Availability & Resilience

- Multi-AZ HA deployment (MetroCluster)
- SnapShot & SnapRestore
- SnapMirror
- SnapMirror Business Continuity
- SnapMirror Cloud

Access Protocols

- NFS –v3, v4, v4.1, v4.2
- SMB – v2, v3
- iSCSI
- Multi-protocol access

Storage Efficiency

- Deduplication & Compression
- Compaction
- Thin provisioning
- Data Tiering (Fabric Pool)

Security & Compliance

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

- NetApp BlueXP **を使用すると、すべてのストレージ資産とデータ資産を単一のコントロールプレーン/インターフェイスから管理できます。

BlueXPを使用して、クラウドストレージ（Cloud Volumes ONTAP やAzure NetApp Files など）の作成と管理、データの移動、保護、分析、オンプレミスやエッジの多くのストレージデバイスの管理を行うことができます。

- NetApp Astra Trident **はCSI準拠のストレージオーケストレーションツールです。上記のさまざまなネットアップストレージオプションを利用して、永続的ストレージをすばやく簡単に利用できます。ネットアップが保守、サポートしているオープンソースのソフトウェアです。

Astra Trident CSI feature highlights

CSI specific

- CSI NetApp® Snapshot™ copies and volume creation from CSI Snapshot copies
- CSI topology
- Volume expansion

Security

- Dynamic-export policy management
- iSCSI initiator-groups dynamic management
- iSCSI bidirectional CHAP

Control

- Storage and performance consumption
- Monitoring
- Volume Import
- Cross Namespace Volume Access

Installation methods

- Binary
- Helm chart
- Operator
- GitOps

Choose your access mode

- RWO (ReadWriteOnce, i.e 1⇔1)
- RWX (*ReadWriteMany*, i.e 1⇔n)
- ROX (*ReadOnlyMany*)
- RWOP (ReadWriteOnce POD)

Choose your protocol

- NFS
- SMB
- iSCSI

ビジネスクリティカルなコンテナワークロードに必要なのは、永続的ボリュームだけではありません。同社のデータ管理要件では、アプリケーションのKubernetesオブジェクトの保護と移行も必要です。

> アプリケーションデータには、ユーザデータに加えてKubernetesオブジェクトが含まれます。例を次に示します。 -ポッド仕様、PVC、デプロイ、サービスなどのKubernetesオブジェクト- 設定マップやシークレットなどのカスタム設定オブジェクト- Snapshotコピー、バックアップ、クローンなどの永続的データ- CRSやCRDなどのカスタムリソース

- NetApp Astra Control **は、フルマネージドと自己管理型の両方のソフトウェアとして提供され、堅牢なアプリケーションデータ管理のためのオーケストレーションを提供します。を参照してください "Astra のドキュメント" を参照してください。

このリファレンスドキュメントでは、NetApp Astra Control Centerを使用して、Red Hat OpenShiftコンテナプラットフォームに導入されたコンテナベースアプリケーションの移行と保護について検証します。また、解決策 では、コンテナプラットフォームを管理するためのRed Hat Advanced Cluster Management（ACM）の導入と使用に関する詳細についても説明しています。また、Astra Trident CSIプロビジョニングツールを使用して、ネットアップストレージとRed Hat OpenShiftコンテナプラットフォームを統合する方法についても詳しく説明します。Astra Control Centerはハブクラスタに導入され、コンテナアプリケーションとその永続的ストレージライフサイクルの管理に使用されます。最後に、Amazon FSx for NetApp ONTAP （FSxN）を永続的ストレージとして使用し、AWS（ROSA）のマネージドRed Hat OpenShiftクラスタ上のコンテナワークロードのレプリケーション、フェイルオーバー、フェイルバックのための解決策 を提供します。

NetApp解決策 とAWS上のマネージドRed Hat OpenShift Containerプラットフォームのワークロード

NetApp解決策 とAWS上のマネージドRed Hat OpenShift Containerプラットフォームのワークロード

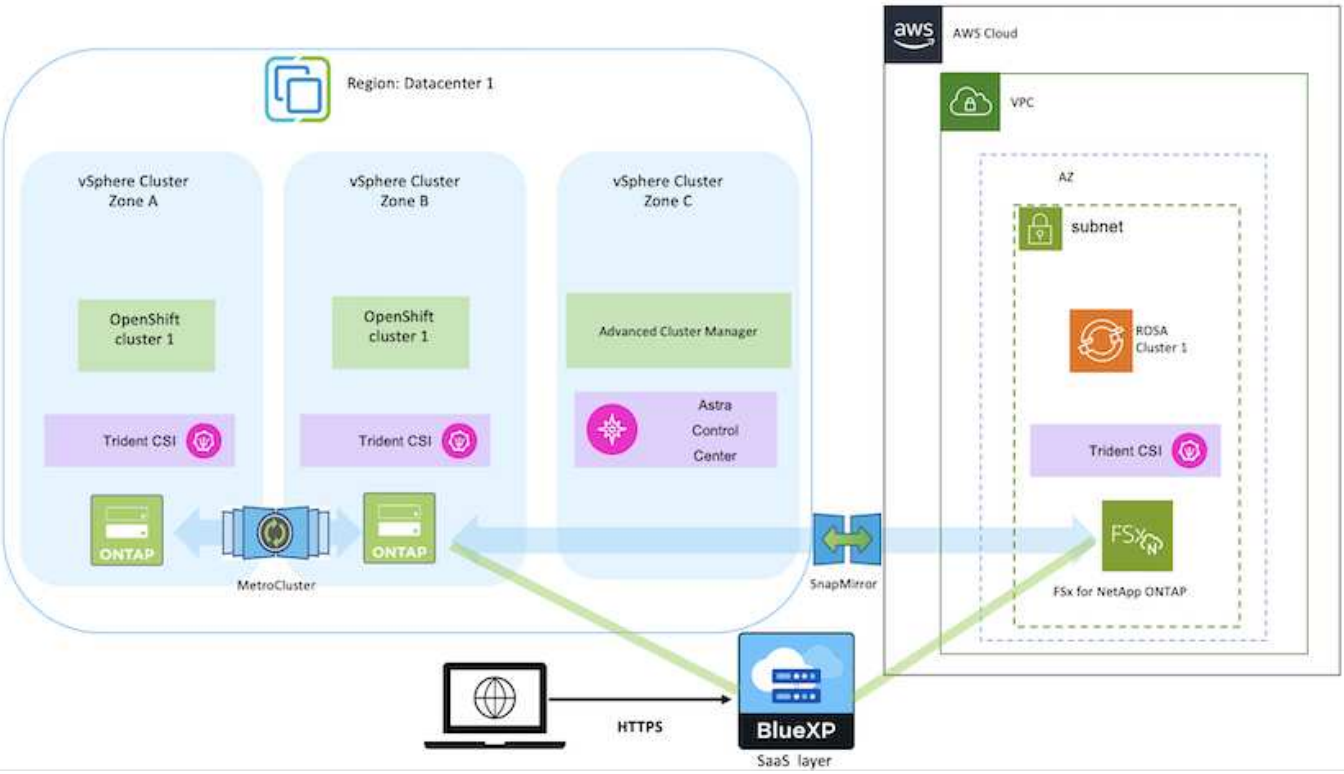
お客様は、「クラウド生まれ」の場合もあれば、一部のワークロードやすべてのワークロードをデータセンターからクラウドに移行する準備ができた時点で、最新化に向けた取り組みを進めている場合もあります。ワークロードの実行に、プロバイダが管理するOpenShiftコンテナとプロバイダが管理するネットアップストレージをクラウドで使用することもできます。コンテナワークロードに対応した本番環境を成功させるためには、マネージドRed Hat OpenShiftコンテナクラスタ（ROSA）をクラウドに計画して導入する必要があります。AWSクラウドにいる場合は、ストレージのニーズに合わせてFSx for NetApp ONTAP を導入することもできます。

FSx for NetApp ONTAP は、AWSのコンテナ導入にデータの保護、信頼性、柔軟性を提供します。Astra Tridentは、お客様のステートフルアプリケーション向けに永続的FSxNストレージを利用するための動的ストレージプロビジョニングツールとして機能します。

ROSAは、コントロールプレーンノードが複数のアベイラビリティゾーンに分散した状態でHAモードで導入できるため、FSx ONTAP は、高可用性を提供し、AZの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することもできます。

> ファイルシステムの優先アベイラビリティゾーン（AZ）からAmazon FSxファイルシステムにアクセスする場合、データ転送料金は発生しません。価格設定の詳細については、を参照してください "こちらをご覧ください"。

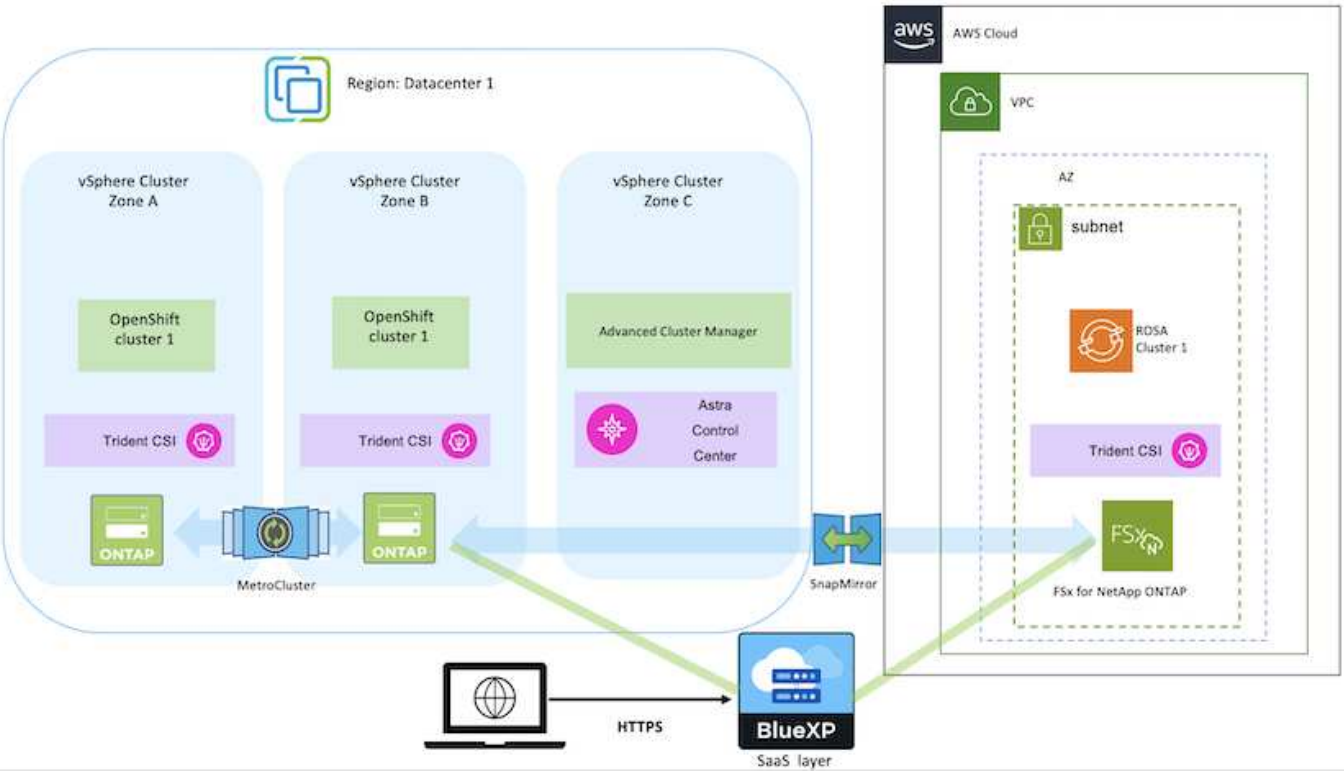
OpenShift Containerワークロード向けのデータ保護と移行用解決策



AWSにマネージド**Red Hat OpenShift Container**プラットフォームを導入して設定します

このセクションでは、AWS（ROSA）でマネージドRed Hat OpenShiftクラスタをセットアップする大まかなワークフローについて説明します。このスライドでは、Astra Tridentによるストレージバックエンドとして Managed FSx for NetApp ONTAP （FSxN）を使用して永続ボリュームを提供しています。BlueXPを使用したAWSへのFSxNの導入について詳しく説明します。また、ROSAクラスタ上のステートフルアプリケーションに対して、BlueXPとOpenShiftのGitOps（Argo CD）を使用してデータ保護と移行のアクティビティを実行する方法についても詳しく説明します。

次の図は、AWSに導入され、FSxNをバックエンドストレージとして使用するROSAクラスタを示しています。



この解決策 は、AWSの2つのVPCで2つのROSAクラスタを使用して検証されました。各ROSAクラスタは、Astra Tridentを使用してFSxNに統合されています。ROSAクラスタとFSxNをAWSに導入するには、いくつかの方法があります。このセットアップの概要概要 には、使用した具体的な方法のドキュメントへのリンクが記載されています。その他の方法については、に記載されている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、次の手順に分けることができます。

ROSAクラスタをインストールします

- 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。
- を参照してください "こちらをご覧ください" ROSAクラスタのインストール手順については、を参照してください。

FSxNをインストールします

- BlueXPからVPCにFSxNをインストールします。を参照してください "こちらをご覧ください"（BlueXPアカウントの作成と使用を開始するため）を参照してください "こちらをご覧ください" FSxNのインストールに使用します。を参照してください "こちらをご覧ください" FSxNを管理するためにAWSでコネクタを作成します。
- AWSを使用してFSxNを導入する。を参照してください "こちらをご覧ください" AWSコンソールを使用した導入用。

ROSAクラスタへの**Trident**のインストール（**Helm**チャートを使用）

- Helmチャートを使用して、ROSAクラスタにTridentをインストールします。HelmチャートのURL：https://netapp.github.io/trident-helm-chart

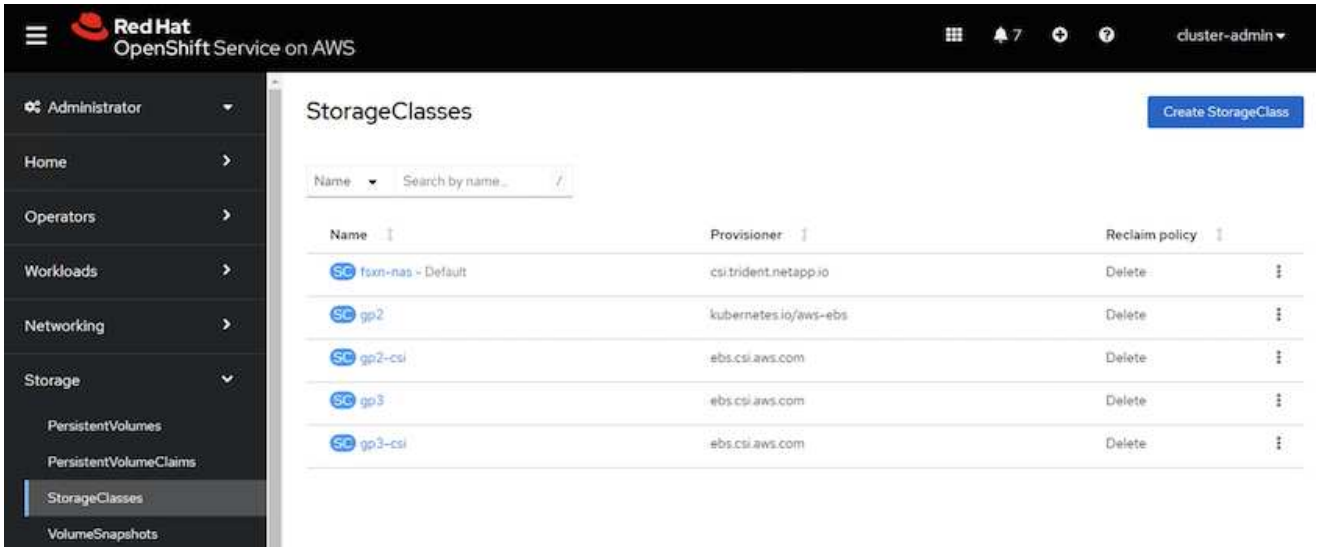
ROSAクラスタ向けのFSxNとAstra Tridentの統合

OpenShift GitOpsを使用すると、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にAstra Trident CSIを導入できます。

```
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を使用したバックエンドとストレージクラスの作成（**FSxN**向け）

- を参照してください "こちらをご覧ください" 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、を参照してください。
- OpenShiftコンソールから、Trident CSIを使用してFsxN用に作成したストレージクラスをデフォルトで作成します。下のスクリーンショットを参照：



OpenShift GitOpsを使用したアプリケーションの導入（**Argo CD**）

- クラスタにOpenShift GitOpsオペレータをインストールします。手順を参照してください "こちらをご覧ください"。
- クラスタ用の新しいArgo CDインスタンスをセットアップします。手順を参照してください "こちらをご覧ください"。

Argo CDのコンソールを開き、アプリをデプロイします。たとえば、Argo CDとHelm Chartを使用してJenkins App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、次の詳細が提供されました。プロジェクト:デフォルトクラスタ: https://kubernetes.default.svc名前空間: Jenkins Helm ChartのURL : https://charts.bitnami.com/bitnami

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

データ保護

このページには、Astra Control Serviceを使用したAWS（ROSA）クラスタでのマネージドRed Hat OpenShiftのデータ保護オプションが表示されます。Astra Control Service（ACS）では、使いやすいグラフィカルユーザインターフェイスを使用して、クラスタの追加、クラスタ上で実行されるアプリケーションの定義、アプリケーション対応のデータ管理アクティビティの実行を行うことができます。ACS関数には、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）に搭載されるのは、NetApp Astra Tridentです。Astra Tridentは、Red Hat

OpenShift、EKS、AKS、SUSE Rancher、Anthosなど、いくつかのタイプのKubernetesクラスタを統合します。 FAS / AFF、ONTAP Select、CVO、Google Cloud Volumes Service、Azure NetApp Files、Amazon FSx for NetApp ONTAPなど、さまざまな種類のNetApp ONTAPストレージを使用できます。

ここでは、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。

- ある地域で実行されているROSAアプリケーションのバックアップと復元と、別の地域への復元を示すビデオ。
- 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（Amazon FSx for NetApp ONTAP）のインストール、NetApp Astra Tridentを使用したストレージバックエンドとの統合、ROSAクラスタへのPostgreSQLアプリケーションのインストール、ACSを使用したアプリケーションのスナップショットの作成とそこからのアプリケーションのリストアの詳細を順を追って説明します。
- ACSを使用してFSx for ONTAPを使用してROSAクラスタ上のMySQLアプリケーションのスナップショットを作成し、そのスナップショットからリストアする手順の詳細を示すブログ。

バックアップ/バックアップからのリストア

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リストアを示しています。

AWSでのRed Hat OpenShift向けFSx NetApp ONTAPサービス

Snapshot / Snapshotからのリストア

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作成してからスナップショットから復元する方法を示します。

Amazon FSx for NetApp ONTAPストレージを使用したRed Hat OpenShift Service on AWS（ROSA）クラスタで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ストレージを使用したROSAクラスタ上のアプリケーションのデータ管理にAstra Control Serviceを使用"

スナップショットを作成してそこからリストアするためのステップバイステップの詳細

セットアップの前提条件

- "AWS アカウント"
- "Red Hat OpenShiftアカウント"
- IAMユーザ "適切な権限" ROSAクラスタを作成してアクセスするには
- "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クラスタインストール済み" VPCに挿入
- "NetApp ONTAP 対応の Amazon FSX" 同じVPC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. 管理者ユーザを作成し、クラスタにログインします。
2. クラスタ用のkubeconfigファイルを作成します。
3. クラスタにAstra Tridentをインストール
4. Trident CSIプロビジョニングツールを使用して、バックエンド、ストレージクラス、Snapshotクラスの構成を作成
5. クラスタにPostgreSQLアプリケーションを導入します。
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アプリケーションで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：管理者ユーザを作成してクラスタにログイン

次のコマンドを使用してadminユーザを作成し、ROSAクラスタにアクセスします（adminユーザを作成する必要があるのは、インストール時にadminユーザを作成しなかった場合だけです）。

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

次のような出力が表示されます。を使用してクラスタにログインします。 `oc login` コマンドは出力に表示されます。

```
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にログインすることもできます。クラスタの作成時にすでにadminユーザを作成している場合は、Red Hat OpenShift Hybrid Cloudコンソールからadminユーザのクレデンシャルを使用してクラスタにログインできます。右上隅にログインしているユーザの名前が表示されていることをクリックすると、 `oc login` コマンドラインのコマンド（トークンログイン）。

2.クラスタの**kubeconfig**ファイルを作成

手順に従います "こちらをご覧ください" ROSAクラスタ用のkubeconfigファイルを作成します。このkubeconfigファイルは、あとでクラスタをACSに追加するときに使用されます。

3.クラスタへの**Astra Trident**のインストール

ROSAクラスタにAstra Trident（最新バージョン）をインストールこれを行うには、以下の手順のいずれかに従うことができます。 "こちらをご覧ください"。クラスタのコンソールから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フィールドの使用 `'https://netapp.github.io/trident-helm-chart'`。次に、Tridentオペレータ用の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

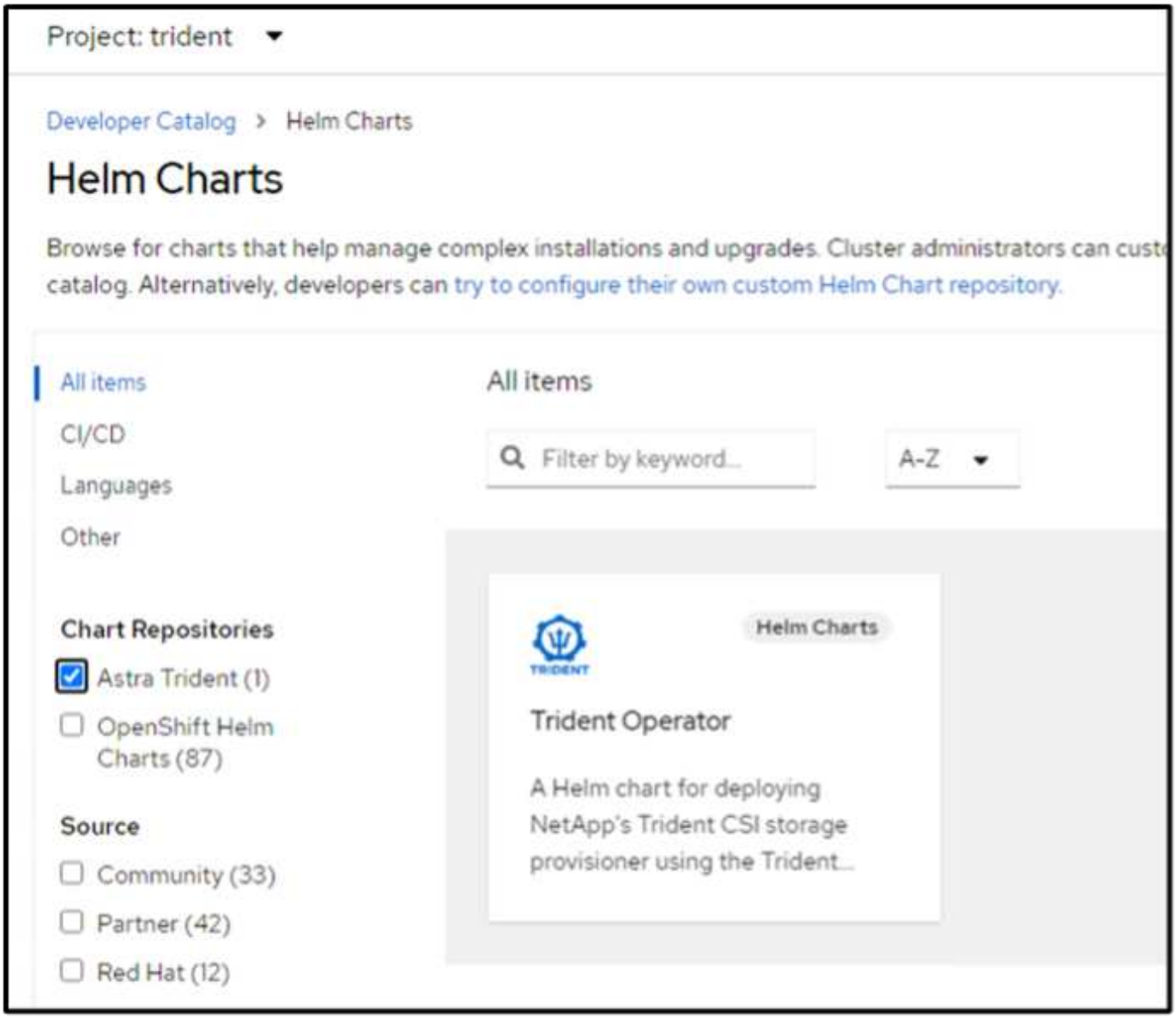
NetApp Astra Trident

A description for the Helm Chart repository.

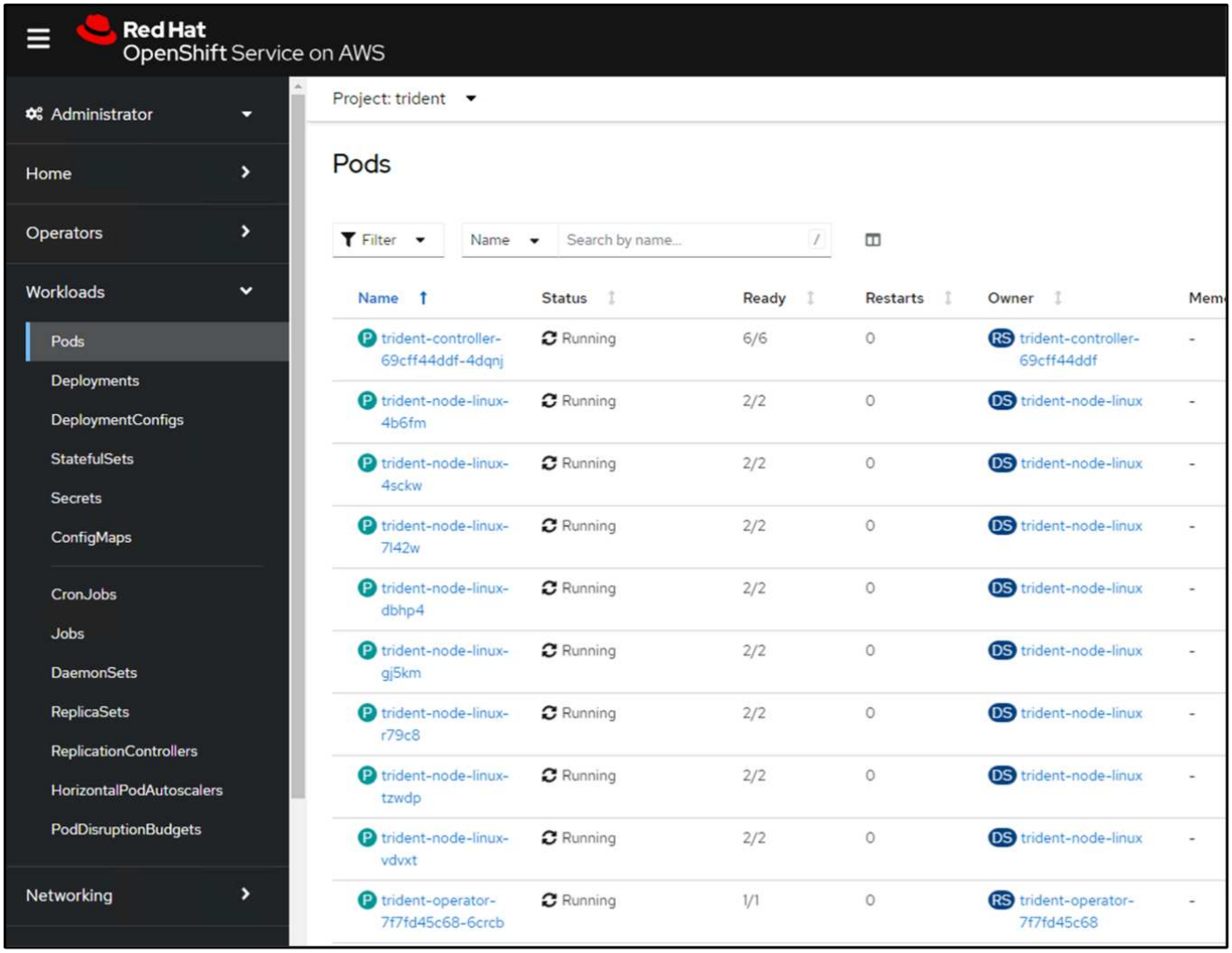
[ ] Disable usage of the repo in the developer catalog.

URL *

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



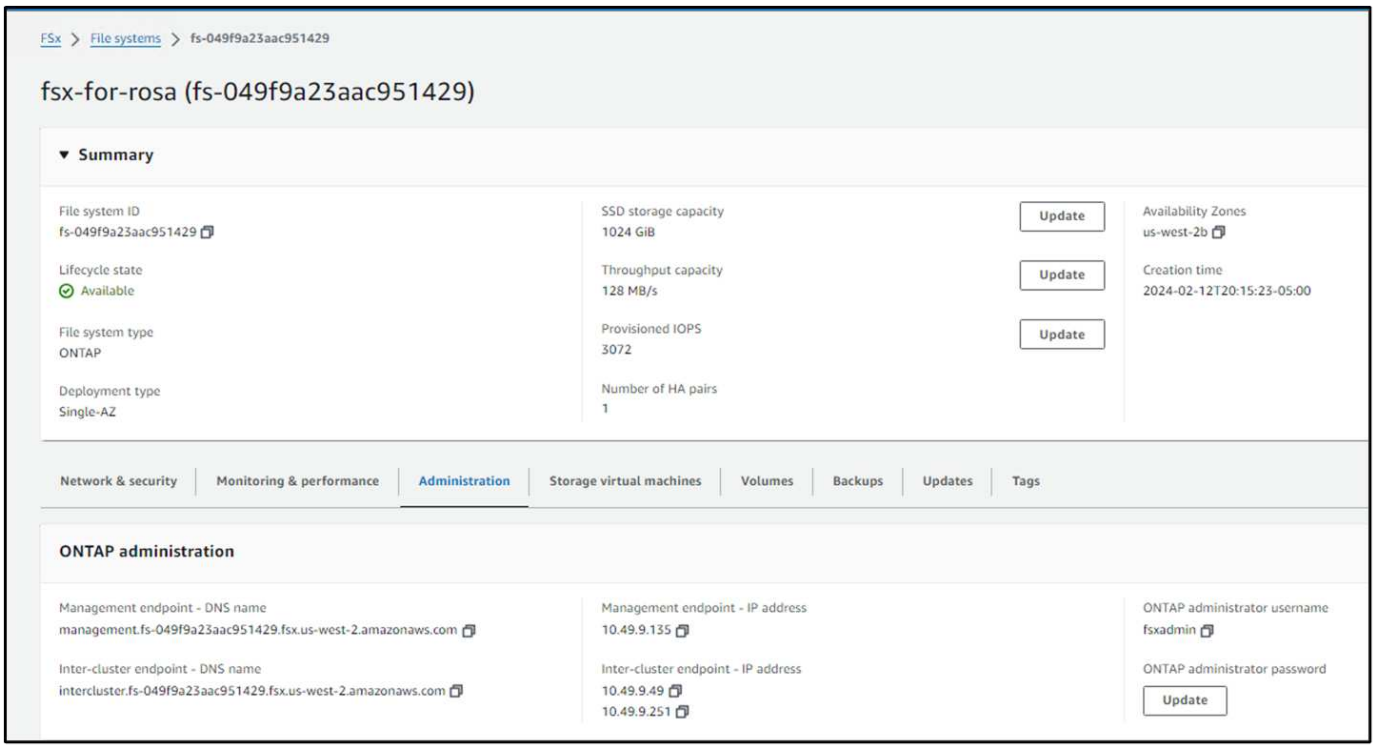
コンソールの管理者ビューに戻り、Tridentプロジェクトでポッドを選択して、すべてのTridentポッドが実行されていることを確認します。



4.Trident CSIプロビジョニングツールを使用して、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

以下のYAMLファイルを使用して、Trident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。作成したAmazon FSx for NetApp ONTAPファイルシステム、管理LIF、およびファイルシステムのSVM名のクレデンシャルを、バックエンドの構成YAMLで指定してください。これらの詳細を確認するには、Amazon FSxのAWSコンソールに移動し、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、[UPDATE]をクリックして、 `fsxadmin` ユーザ：

> コマンドラインを使用して、ハイブリッドクラウドコンソールからオブジェクトを作成したり、YAMLファイルを使用してオブジェクトを作成したりできます。



- 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
```

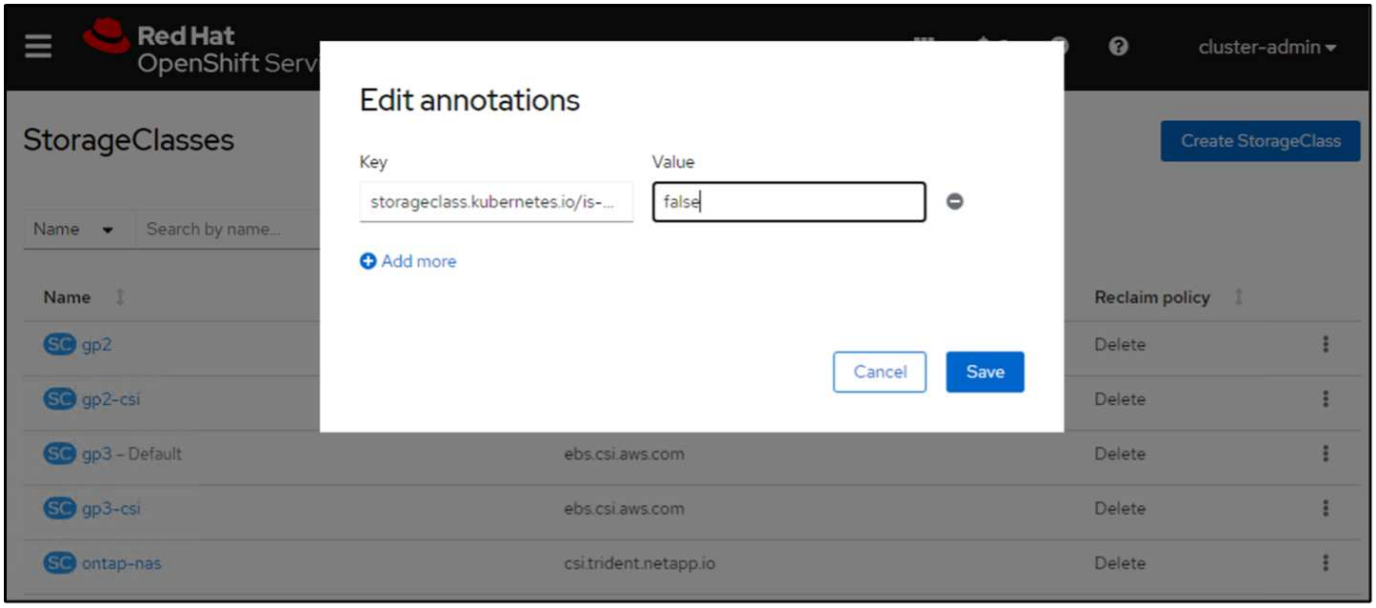
スナップショットクラス

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

以下のコマンドを実行して、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たことを確認します。

```
[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]$
```

この時点で重要な変更点は、あとで導入するPostgreSQLアプリケーションでデフォルトのストレージクラスを使用できるように、ONTAP-NASをgp3ではなくデフォルトのストレージクラスに設定することです。クラスタのOpenShiftコンソールで、[Storage]で[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ストレージクラスに対してstorageclass.kubernetes.io/is-default-classをtrueに設定して追加します。



5.クラスタに**PostgreSQL**アプリケーションを導入する

次のように、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が原因で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]$
```

を編集してエラーを修正します。 `runAsUser` および `fsGroup` フィールド `statefuleset.apps/postgresql` の出力にあるuidを持つオブジェクト `oc get project` 次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アプリケーションを実行し、Amazon FSx for NetApp ONTAPストレージを基盤とする永続ボリュームを使用する必要があります。

```
[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]$
```

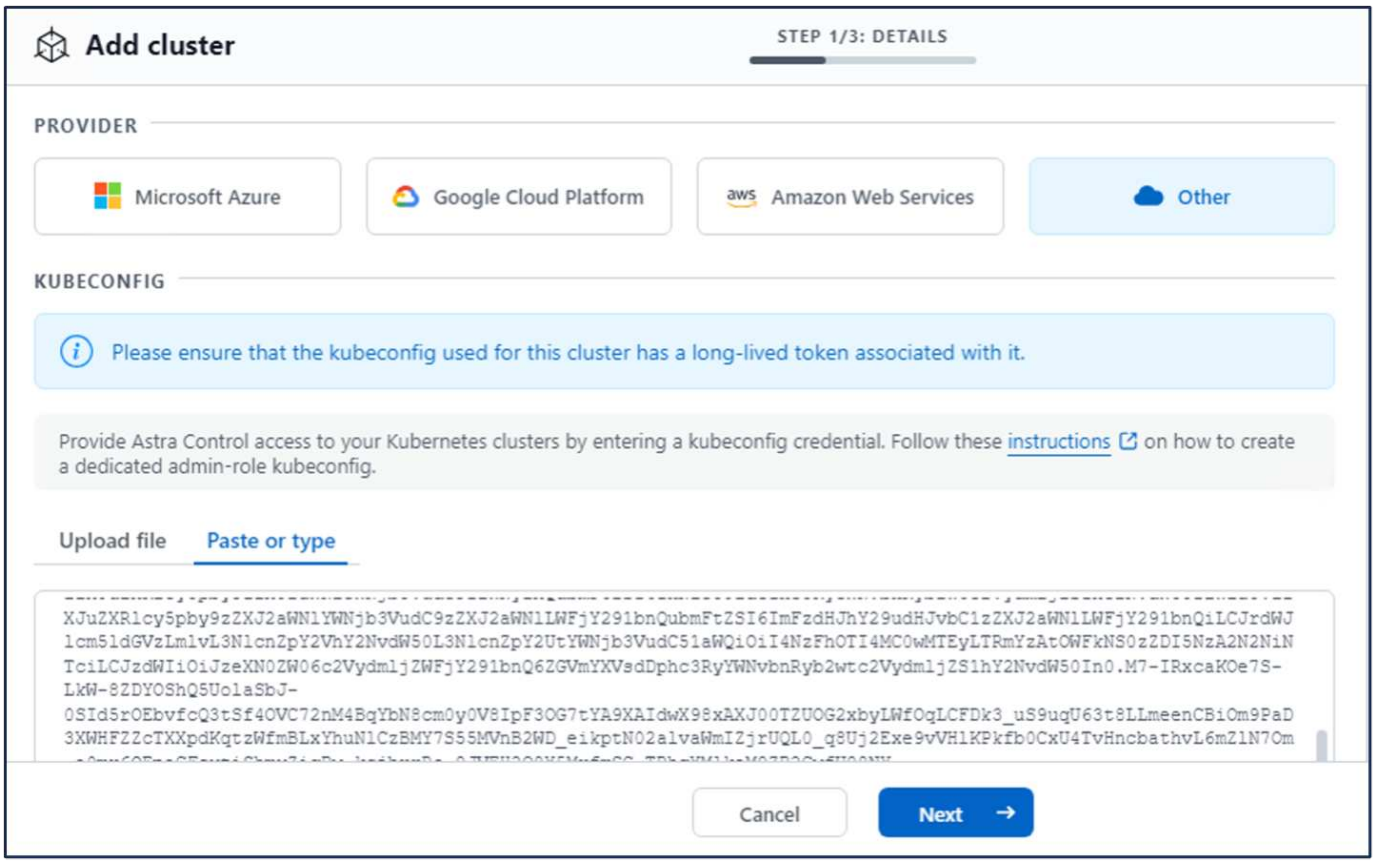
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

6.データベースの作成とレコードの追加

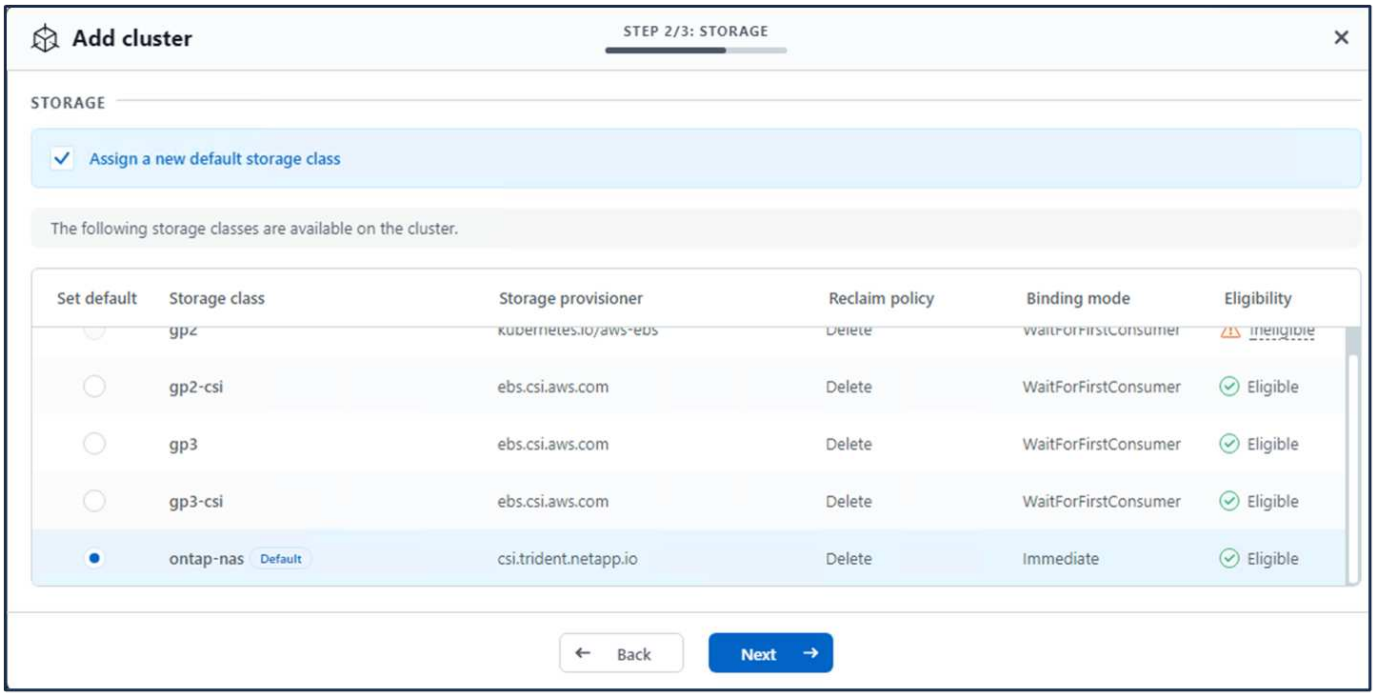
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1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

ACSにログインします。クラスタを選択し、[Add]をクリックします。[Other]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。



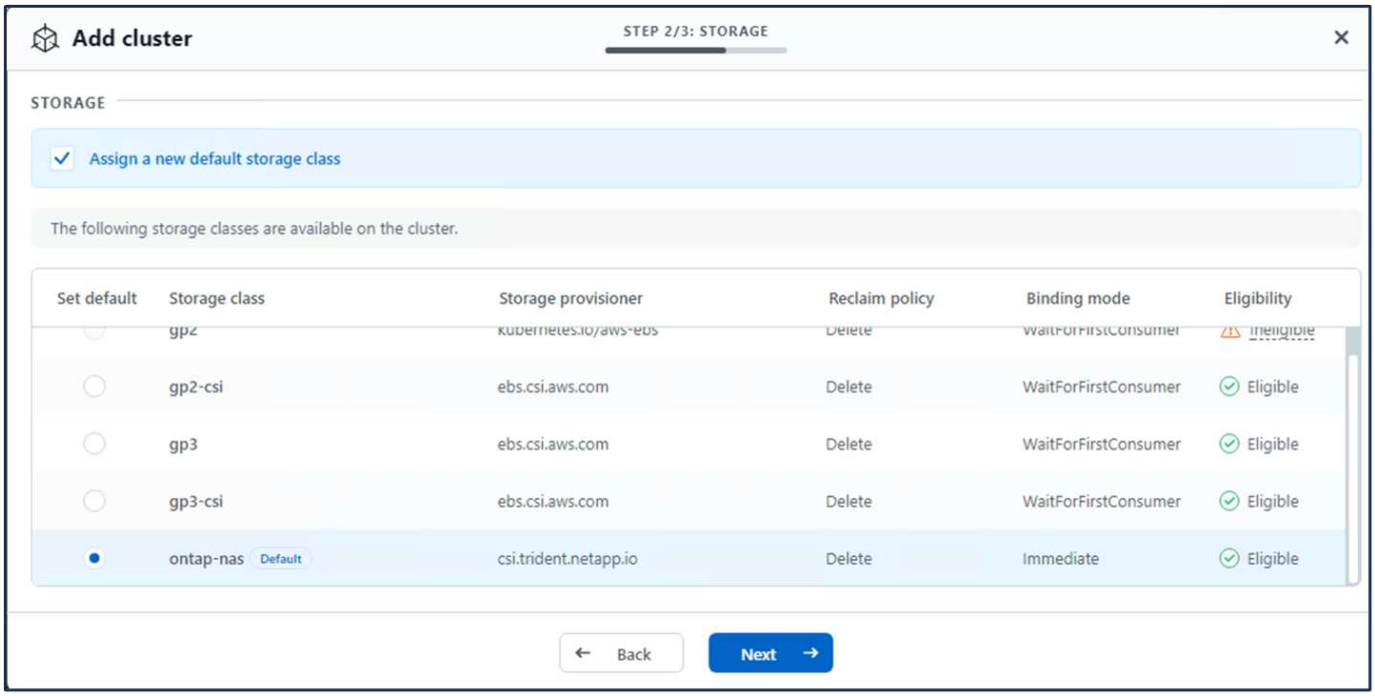
をクリックし、**ACS**のデフォルトのストレージクラスとして**[ONTAP-NAS]**を選択します。**[**次へ**]***をクリックし、詳細を確認して[クラスタを追加]*をクリックします。



8.ACSでのアプリケーションの定義

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で*、[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

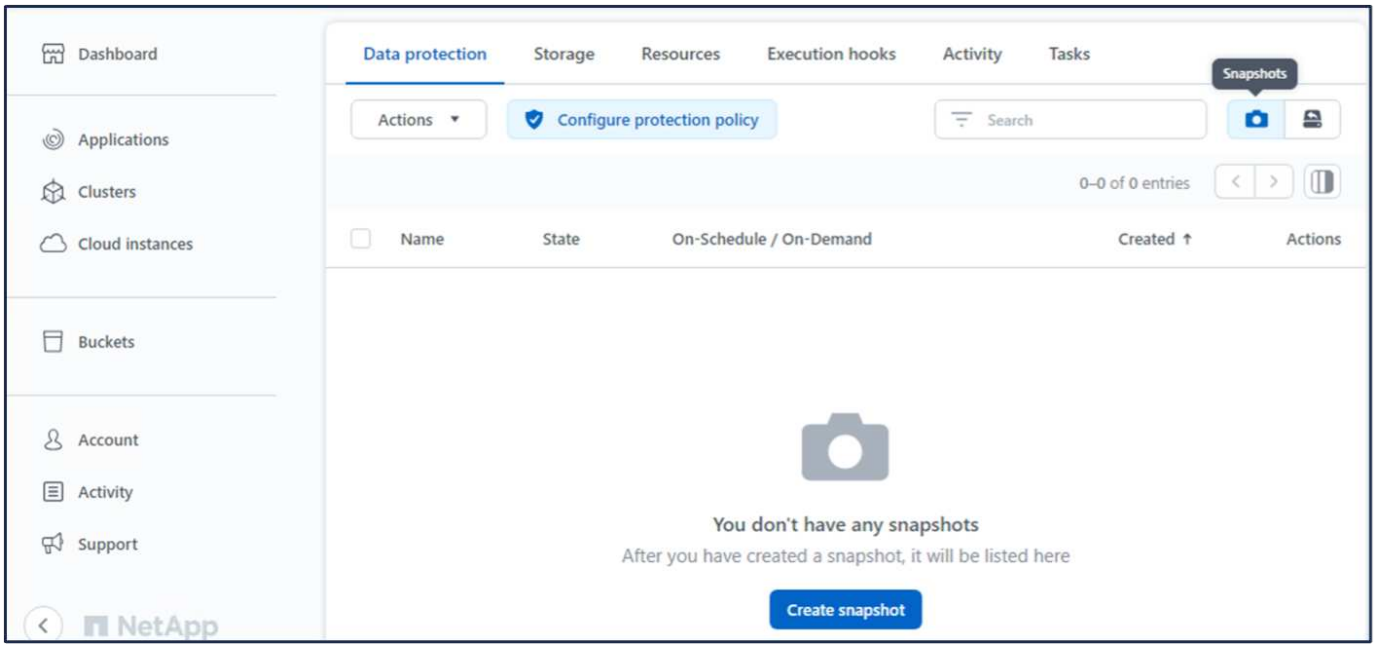
追加されます。

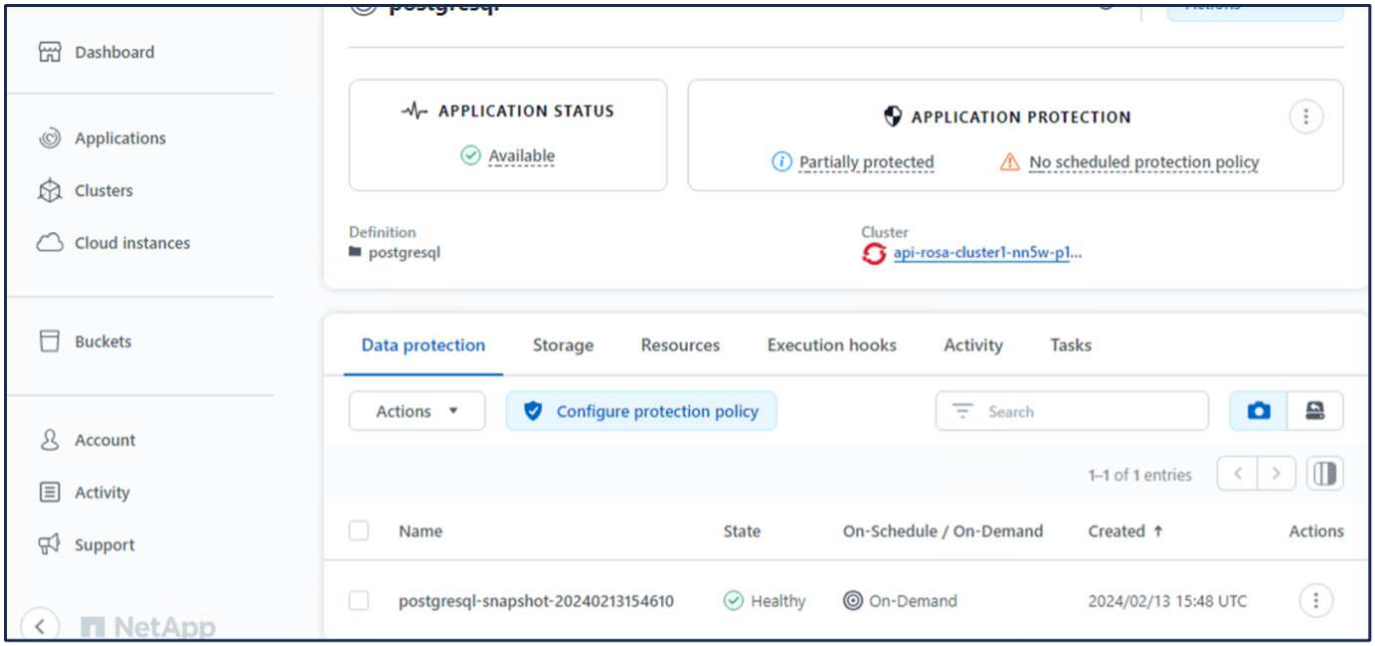


9.ACSを使用したスナップショットの作成

ACSでスナップショットを作成するには、さまざまな方法があります。アプリケーションを選択し、アプリケーションの詳細が表示されたページからスナップショットを作成できます。[Create snapshot]をクリックすると、オンデマンドSnapshotを作成したり、保護ポリシーを設定したりできます。

をクリックして名前を指定し、詳細を確認して[Snapshot]*をクリックするだけで、オンデマンドSnapshotを作成できます。処理が完了すると、Snapshotの状態が「Healthy」に変わります。





10。PostgreSQLアプリケーション内のデータベースの削除

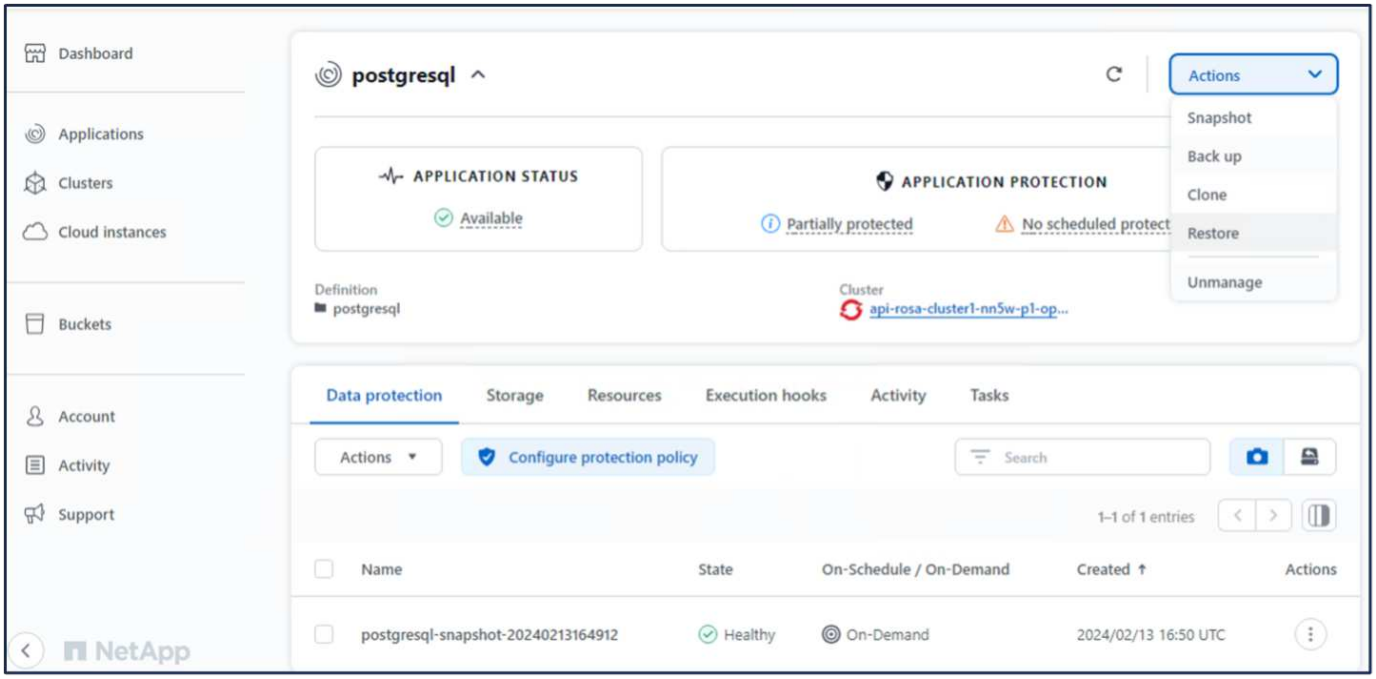
PostgreSQLに再度ログインし、利用可能なデータベースを一覧表示し、以前に作成したデータベースを削除して、データベースが削除されたことを確認します。

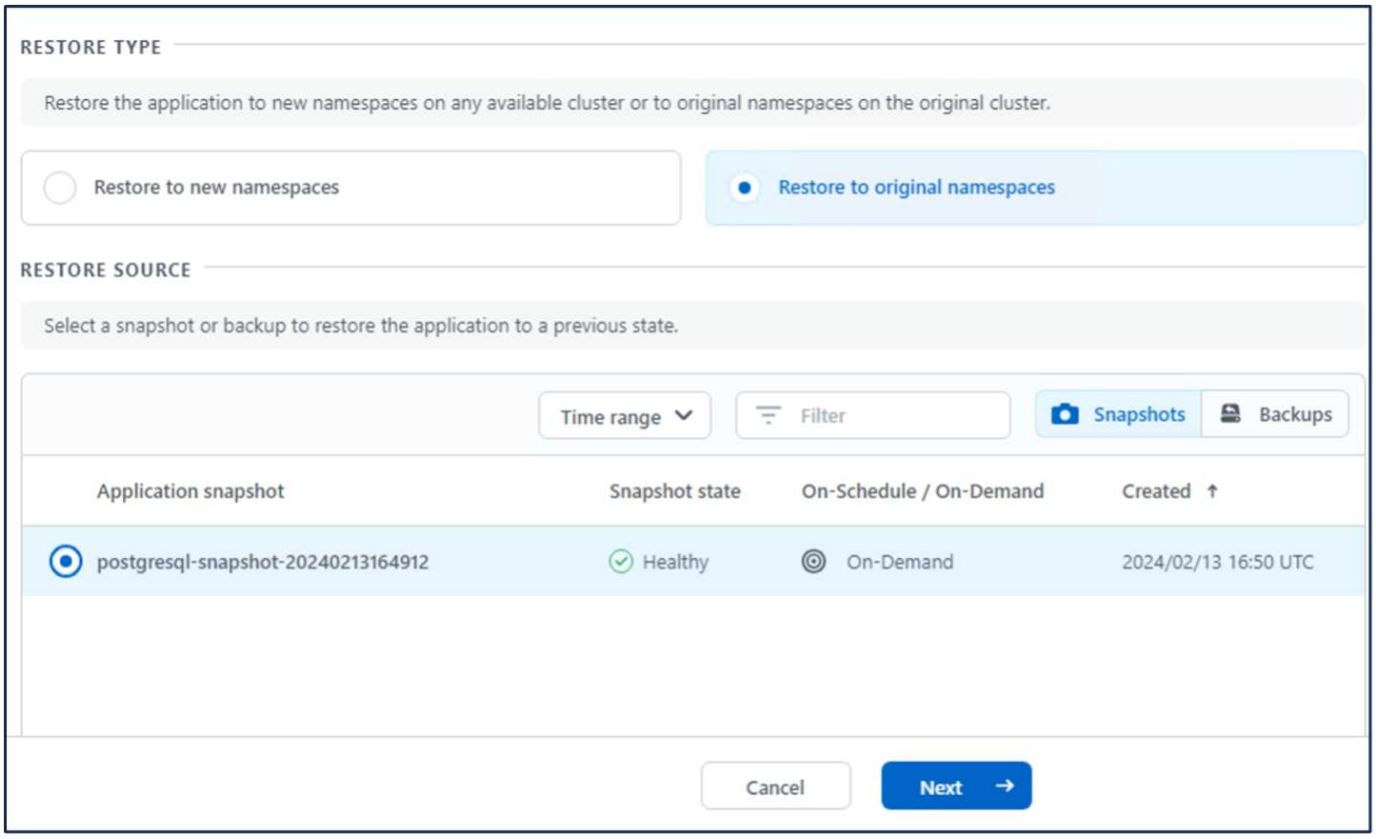
```
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------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

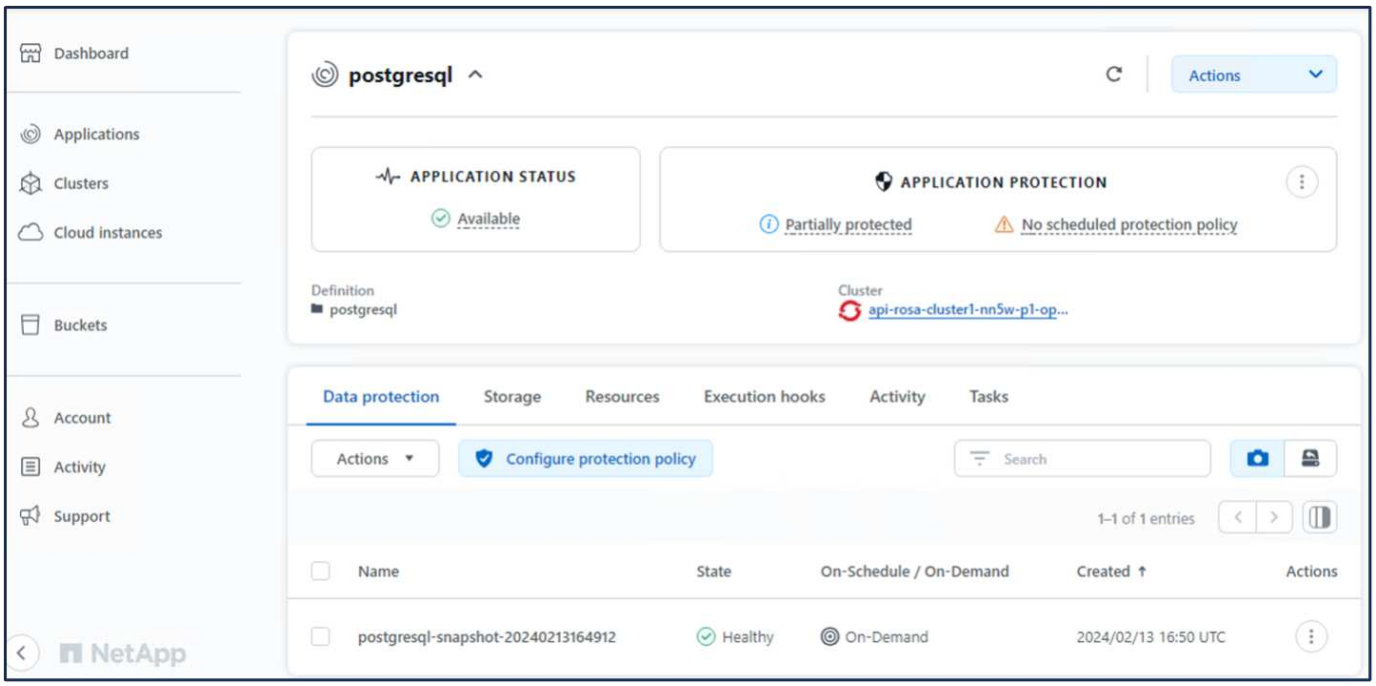
11.ACSを使用したスナップショットからのリストア

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[Restore]を選択します。リストア元の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のが作成されます）。次の2つ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リストアされると、アプリケーションのステータスがRestoring（復元中）からAvailable（使用可能）に変わります。







12.アプリケーション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、以前に使用していたテーブルとレコードが表示されます。 これで終わりです。ボタンをクリックするだけで、アプリケーションは以前の状態に復元されます。Astra Controlを使用することで、お客様はそれを簡単に実現できます。

```
[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、永続的ストレージにFSx for NetApp ONTAP を使用したマネージドRed Hat OpenShiftクラスタでのコンテナワークロードのデータ移行オプションを示します。

データ移行

AWS上のRed Hat OpenShiftサービスとFSx for NetApp ONTAP （FSxN）は、AWSによるサービスポートフォリオに含まれています。FSxNは、単一のAZまたは複数のAZオプションで使用できます。複数のAZオプションを使用すると、アベイラビリティゾーンの障害からデータを保護できます。FSxNをAstra Tridentと統合することで、ROSAクラスタ上のアプリケーションに永続的ストレージを提供できます。

Helmチャートを使用した**FSxN**と**Trident**の統合

RosaクラスタとAmazon FSx for 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。

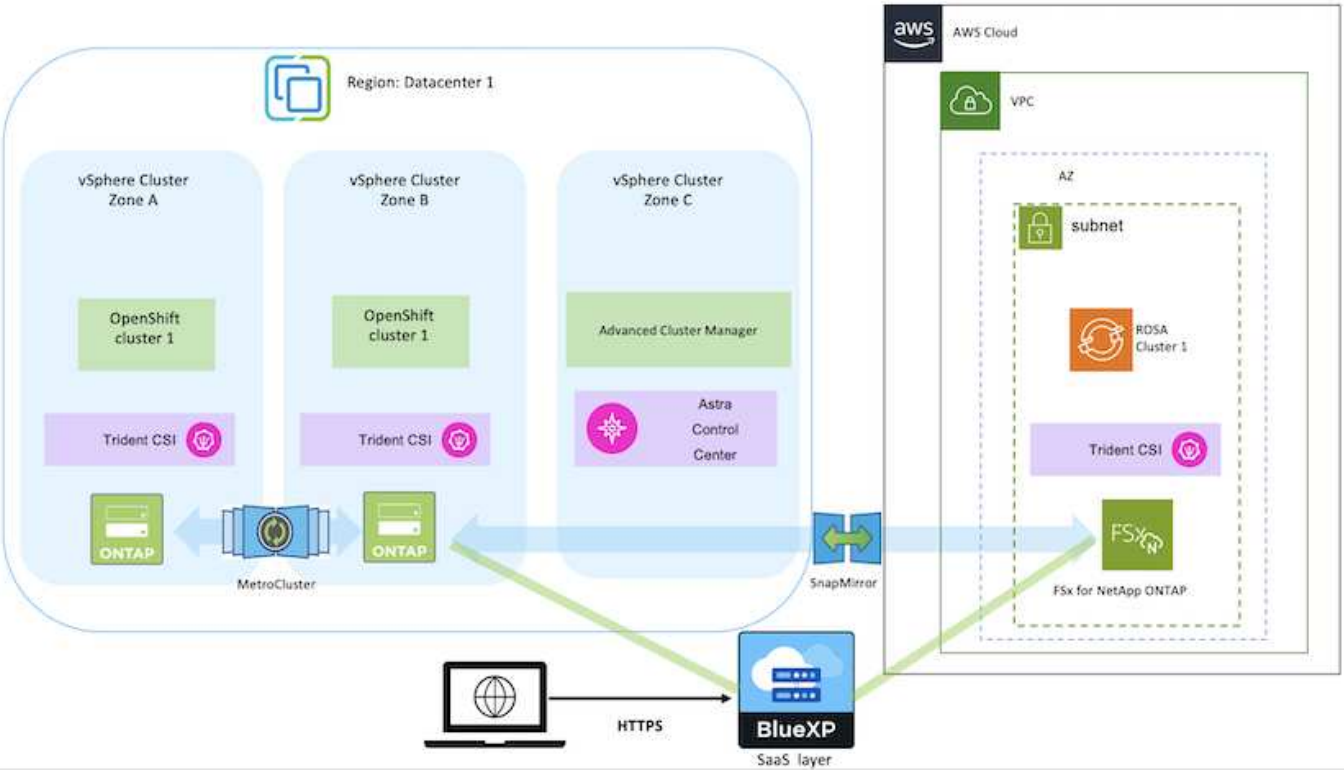
- 永続ボリューム：これはBlueXPを使用して実行できます。もう1つの方法は、Astra Control Centerを使用して、オンプレミスからクラウド環境へのコンテナアプリケーションの移行を処理する方法です。自動化も同じ目的で使用できます。
- アプリケーションメタデータ:これはOpenShift GitOps (Argo CD)を使用して実行できます。

永続的ストレージに**FSxN**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イルオーバーとフェイ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ールオーバーとフェールバック

OpenShift Containerワークロード向けのデータ保護と移行用解決策



著作権に関する情報

Copyright © 2024 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。