具有提供者管理元件的混合雲
NetApp public and hybrid cloud solutions

NetApp
August 20, 2025

This PDF was generated from https://docs.netapp.com/zh-tw/netapp-solutions-cloud/openshift/os-pm-solution.html on August 20, 2025. Always check docs.netapp.com for the latest.

目錄

具有提供者管理元件的混合雲 1
- 具有託管 Red Hat OpenShift 容器平台工作負載的NetApp解決方案 1
- 在 AWS 上部署和設定託管 Red Hat OpenShift 容器平台 1
- 使用Google Cloud NetApp Volumes在 Google Cloud 上部署和設定 OpenShift Dedicated 4
- 資料保護 6
  - 備份/從備份中恢復 6
  - 快照/從快照恢復 6
  - 部落格 6
  - 建立快照並從中復原的詳細步驟 6
- 資料遷移 21
  - 資料遷移 22
- 適用於 Red Hat OpenShift 工作負載的其他NetApp混合多雲解決方案 23
  - 其他解決方案 23

具有提供者管理元件的混合雲

具有託管 **Red Hat OpenShift** 容器平台工作負載的**NetApp**解決方案

客戶可能“誕生於雲端”，或正處於現代化歷程的某個階段，準備將部分選定的工作負載或所有工作負載從資料中心遷移到雲端。他們可以選擇在雲端中使用供應商管理的 OpenShift 容器和供應商管理的NetApp儲存體來運行他們的工作負載。他們應該在雲端中規劃和部署託管的 Red Hat OpenShift 容器集群，以便為他們的容器工作負載提供成功的生產就緒環境。 NetApp為三大領先公有雲中的託管 Red Hat 解決方案提供完全託管的儲存產品。

Amazon FSx for NetApp ONTAP (FSx ONTAP)

FSx ONTAP為 AWS 中的容器部署提供資料保護、可靠性和靈活性。 Trident作為動態儲存供應器，為客戶的有狀態應用程式使用持久性 FSx ONTAP儲存。

由於 ROSA 可以在 HA 模式下部署，並且控制平面節點分佈在多個可用區域，因此 FSx ONTAP還可以配置多可用區選項，以提供高可用性並防止可用區故障。

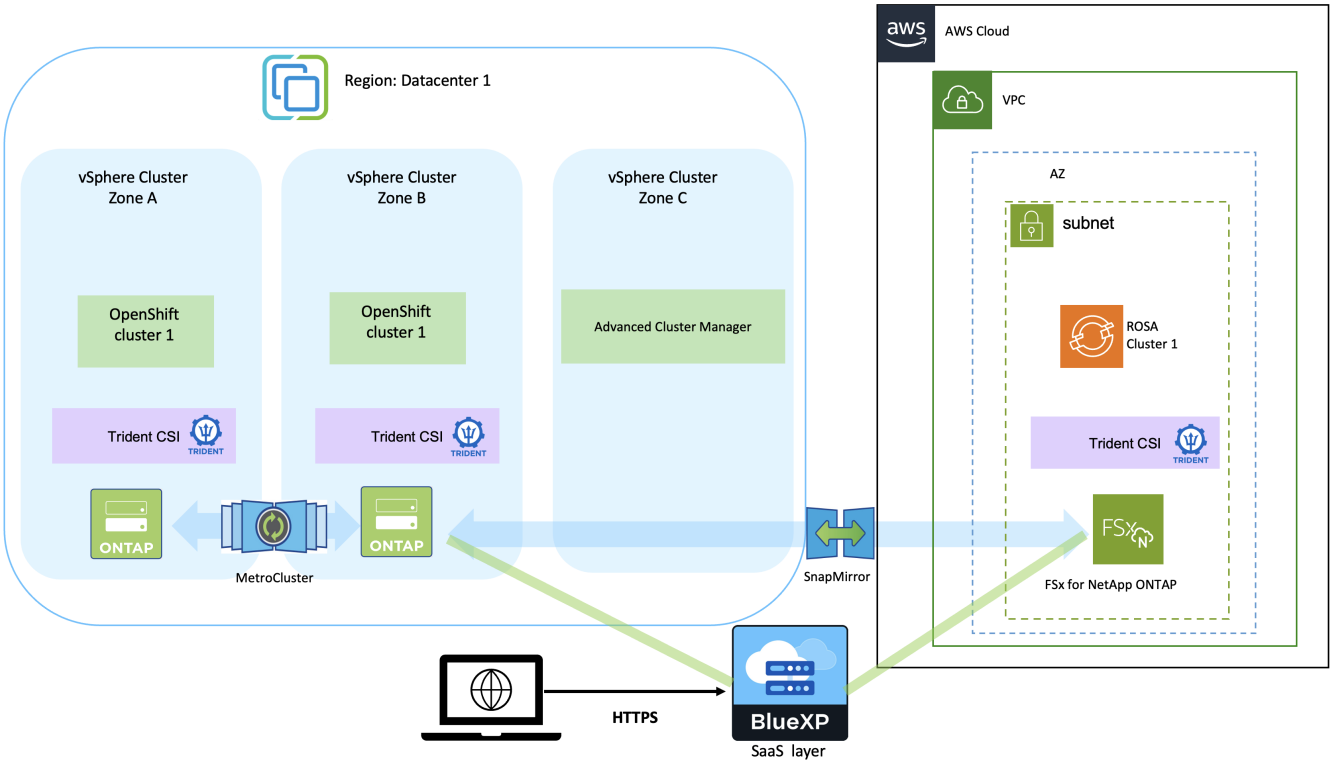
- Google Cloud NetApp Volumes*

Red Hat OpenShift Dedicated 是一個完全託管的應用程式平台，可讓您在混合雲中快速建置、部署和擴展應用程式。 Google Cloud NetApp Volumes提供持久性卷，將 ONTAP 的全套企業資料管理功能帶入 Google Cloud 中的 OpenShift 部署。

在 **AWS** 上部署和設定託管 **Red Hat OpenShift** 容器平台

本節介紹在 AWS（ROSA）上設定託管 Red Hat OpenShift 叢集的進階工作流程。它展示了Trident使用託管的Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為儲存後端來提供持久性磁碟區。提供了有關使用BlueXP在 AWS 上部署 FSx ONTAP的詳細資訊。此外，還提供了有關使用BlueXP和 OpenShift GitOps（Argo CD）為 ROSA 叢集上的有狀態應用程式執行資料保護和遷移活動的詳細資訊。

下圖描述了部署在 AWS 上並使用 FSx ONTAP作為後端儲存的 ROSA 叢集。



該解決方案透過在AWS的兩個VPC中使用兩個ROSA集群進行了驗證。每個 ROSA 叢集都使用Trident與 FSx ONTAP整合。在 AWS 中部署 ROSA 叢集和 FSx ONTAP有幾種方法。此設定的高級描述提供了所使用的特定方法的文檔連結。您可以參考"資源部分"。

設定過程可分為以下步驟：

安裝**ROSA**集群

- 建立兩個 VPC 並在 VPC 之間建立 VPC 對等連線。
- 參考"這裡"有關安裝 ROSA 集群的說明。

安裝 **FSx ONTAP**

- 從BlueXP在 VPC 上安裝 FSx ONTAP 。參考"這裡"用於建立BlueXP帳戶並開始使用。參考"這裡"用於安裝 FSx ONTAP。參考"這裡"用於在 AWS 中建立連接器來管理 FSx ONTAP。
- 使用 AWS 部署 FSx ONTAP 。參考"這裡"使用 AWS 控制台進行部署。

在 **ROSA** 叢集上安裝**Trident** （使用 **Helm** 圖表）

- 使用 Helm chart 在 ROSA 叢集上安裝Trident 。請參閱文件連結： https://docs.netapp.com/us-en/trident/trident-get-started/kubernetes-deploy-helm.html [此處]。

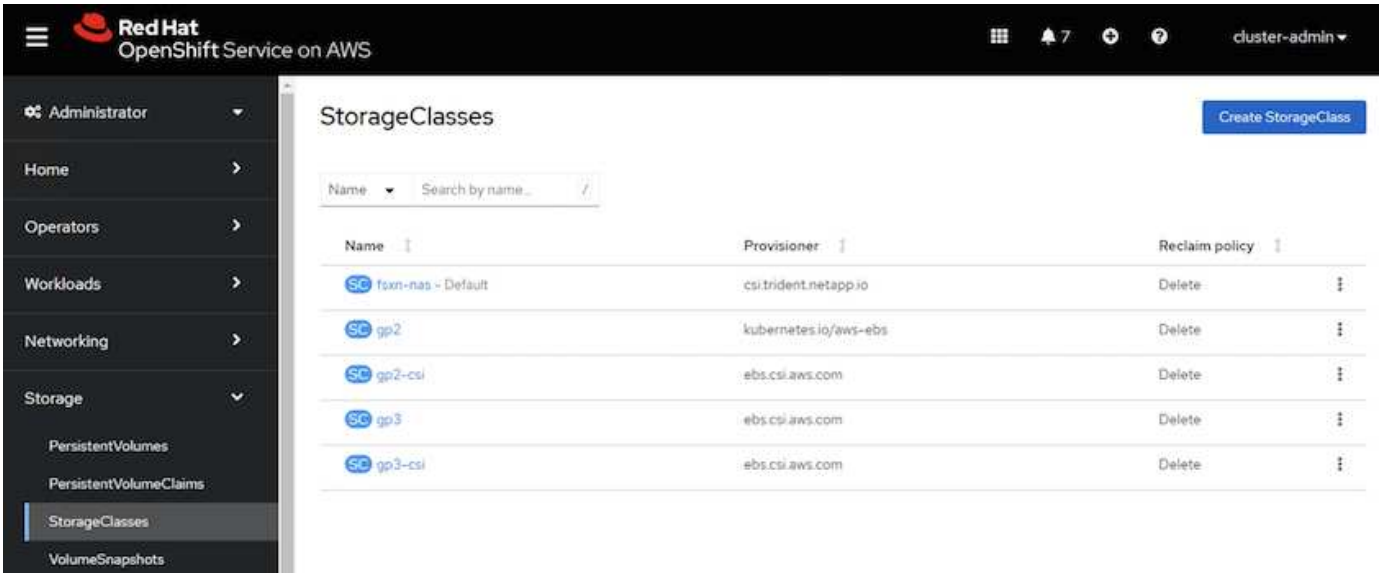
FSx ONTAP與Trident整合用於 ROSA 集群

OpenShift GitOps 可用於將Trident CSI 部署到所有託管集群，因為它們使用 ApplicationSet 註冊到 ArgoCD。

```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
```

使用**Trident**建立後端和儲存類別（適用於 **FSx ONTAP**）

- 參考"這裡"有關建立後端和儲存類別的詳細資訊。
- 從 OpenShift 控制台將使用Trident CSI 為 FsxN 建立的儲存類別設為預設。請參閱下面的截圖：



使用 **OpenShift GitOps**（**Argo CD**）部署應用程式

- 在叢集上安裝 OpenShift GitOps 操作員。參考說明"這裡"。
- 為叢集設定一個新的 Argo CD 實例。參考說明"這裡"。

打開Argo CD的控制台並部署一個應用程式。例如，您可以使用帶有 Helm Chart 的 Argo CD 部署 Jenkins 應用程式。建立應用程式時，提供了以下詳細資訊：專案：預設叢集：'https://kubernetes.default.svc' （不含引號）

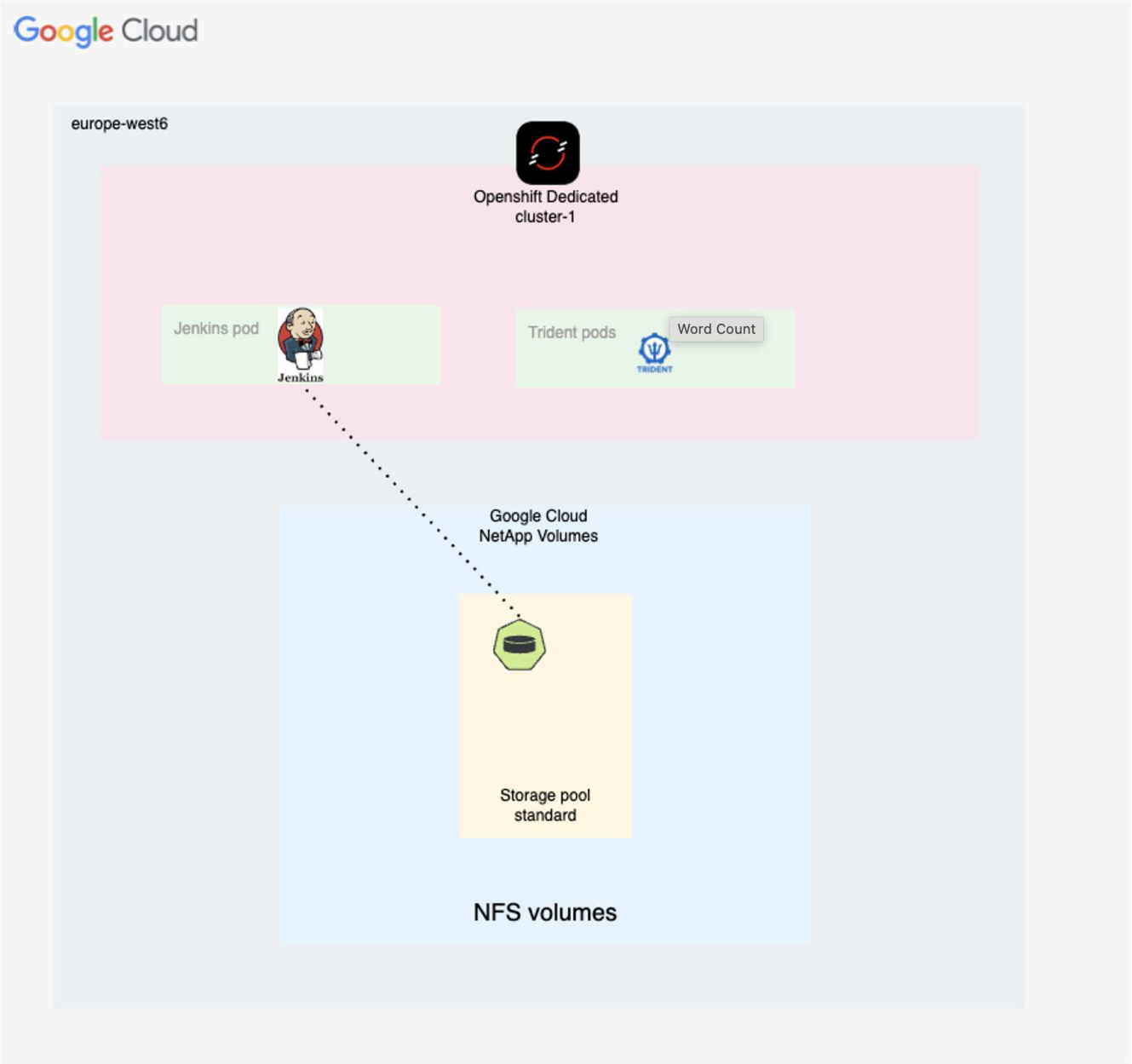
命名空間：Jenkins Helm Chart 的 URL：'https://charts.bitnami.com/bitnami' （不含引號）

Helm 參數：global.storageClass: fsxn-nas

使用Google Cloud NetApp Volumes在 Google Cloud 上部署和設定 OpenShift Dedicated

本節介紹在 Google Cloud 平台上設定 OpenShift Dedicated (OSD) 叢集的高階工作流程。它顯示NetApp Trident使用Google Cloud NetApp Volumes作為儲存後端，為使用 Kubernetes 運行的有狀態應用程式提供持久性磁碟區。

下圖描述了部署在 Google Cloud 上並使用NetApp Volumes 作為後端儲存的 OSD 叢集。



設定過程可分為以下步驟：

在 **Google Cloud** 中安裝 **OSD** 集群

- 如果您希望為叢集使用現有的 VPC，則必須為 OSD 叢集建立 VPC、兩個子網路、一個雲端路由器和兩個 GCP 雲端 NAT。參考"這裡"以取得說明。
- 參考"這裡"有關使用客戶雲端訂閱 (CCS) 計費模型在 GCP 上安裝 OSD 叢集的說明。 OSD 也包含在 Google Cloud Marketplace 中。示範如何使用 Google Cloud Marketplace 解決方案安裝 OSD 的影片位於"這裡"。

啟用**Google Cloud NetApp Volumes**

- 參考"這裡"有關設定對Google Cloud NetApp Volumes 的存取權限的資訊。遵循所有步驟，包括
- 建立儲存池。參考"這裡"有關如何在Google Cloud NetApp Volumes上設定儲存池的資訊。將在儲存池內建立在 OSD 上執行的有狀態 Kubernetes 應用程式的磁碟區。

在 **OSD** 叢集上安裝**Trident** （使用 **Helm** 圖表）

- 使用 Helm 圖表在 OSD 叢集上安裝Trident 。參考"這裡"有關如何安裝 Helm Chart 的說明。舵圖可能位於"這裡"。

將**NetApp Volumes** 與**NetApp Trident**整合用於 **OSD** 集群

使用Trident建立後端和儲存類別（適用於Google Cloud NetApp Volumes）

- 有關創建後端的詳細信息，請參閱此處。
- 如果 kubernetes 中的任何目前儲存類別被標記為默認，請透過編輯儲存類別來刪除該註釋。
- 使用Trident CSI 設定程式為NetApp磁碟區建立至少一個儲存類別。使用註解將其中一個儲存類別設為預設儲存類別。當 PVC 清單中未明確呼叫時，這將允許 PVC 使用此儲存類別。下面顯示了一個帶有註釋的範例。

```
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
```

使用 **OpenShift GitOps**（**Argo CD**）部署應用程式

- 在叢集上安裝 OpenShift GitOps 操作員。參考說明"這裡"。
- 為叢集設定一個新的 Argo CD 實例。參考說明"這裡"。

打開Argo CD的控制台並部署一個應用程式。例如，您可以使用帶有 Helm Chart 的 Argo CD 部署 Jenkins 應用程式。建立應用程式時，提供了以下詳細資訊：專案：預設叢集：'https://kubernetes.default.svc' （不含引號）命名空間：Jenkins Helm Chart 的 URL：'https://charts.bitnami.com/bitnami' （不含引號）

資料保護

此頁面顯示使用Astra Control Service 的託管 Red Hat OpenShift on AWS (ROSA) 叢集的資料保護選項。 Astra Control Service (ACS) 提供了一個易於使用的圖形使用者介面，您可以使用它來新增叢集、定義在其上運行的應用程式以及執行應用程式感知資料管理活動。還可以使用允許工作流程自動化的 API 存取 ACS 功能。

為Astra Control（ACS 或 ACC）提供支援的是NetApp Trident。 Trident整合了多種類型的 Kubernetes 集群，例如 Red Hat OpenShift、EKS、AKS、SUSE Rancher、Anthos 等，以及各種類型的NetApp ONTAP存儲，例如FAS/ AFF、 ONTAP Select、CVO、Google Google Cloud NetApp Volumes、 Azure NetApp Files和Amazon FSx ONTAP。

本節詳細介紹了使用 ACS 的以下資料保護選項：

- 展示在一個區域運行的 ROSA 應用程式的備份和還原以及還原到另一個區域的影片。
- 展示 ROSA 應用程式的快照和復原的影片。
- 安裝 ROSA 叢集、 Amazon FSx ONTAP、使用NetApp Trident與儲存後端整合、在 ROSA 叢集上安裝 postgresql 應用程式、使用 ACS 建立應用程式的快照以及從中恢復應用程式的分步詳細資訊。
- 這篇部落格逐步展示了使用 ACS 在具有 FSx ONTAP的 ROSA 叢集上為 mysql 應用程式建立和復原快照的過程。

備份/從備份中恢復

以下影片展示了在一個區域運行的 ROSA 應用程式的備份以及還原到另一個區域的過程。

AWS 上的 FSx NetApp ONTAP for Red Hat OpenShift 服務

快照/從快照恢復

以下影片展示如何拍攝 ROSA 應用程式的快照以及如何從快照中復原。

使用Amazon FSx ONTAP儲存為 AWS (ROSA) 叢集上的 Red Hat OpenShift Service 應用程式進行快照/還原

部落格

- "使用Astra Control Service 和Amazon FSx儲存對 ROSA 叢集上的應用程式進行資料管理"

建立快照並從中復原的詳細步驟

先決條件設定

- "AWS 帳號"
- "Red Hat OpenShift 帳戶"
- IAM 用戶"適當的權限"建立並存取ROSA集群
- "AWS CLI"
- "ROSA CLI"

- "OpenShift CLI"（oc）
- 具有子網路及適當閘道及路由的 VPC
- "ROSA集群安裝"進入 VPC
- "Amazon FSx ONTAP"在同一個 VPC 中創建
- 從以下位置存取 ROSA 集群"OpenShift 混合雲控制台"

後續步驟

1. 建立管理員使用者並登入叢集。
2. 為叢集建立一個 kubeconfig 檔案。
3. 在叢集上安裝Trident 。
4. 使用Trident CSI 設定器建立後端、儲存類別和快照類別配置。
5. 在叢集上部署postgresql應用程式。
6. 建立資料庫並新增記錄。
7. 將集群新增至 ACS。
8. 在 ACS 中定義應用程式。
9. 使用 ACS 建立快照。
10. 刪除 postgresql 應用程式中的資料庫。
11. 使用 ACS 從快照恢復。
12. 驗證您的應用程式已從快照還原。

1.建立管理員使用者並登入叢集

使用以下命令建立管理員使用者來存取 ROSA 叢集：（僅當您在安裝時未建立管理員使用者時才需要建立管理員使用者）

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

該命令將提供如下所示的輸出。使用 `oc login`輸出中提供的命令。

```
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
```

您也可以使用令牌登入叢集。如果您在建立叢集時已經建立了管理員用戶，則可以使用管理員使用者憑證從 Red Hat OpenShift Hybrid Cloud 控制台登入叢集。然後點擊右上角顯示登入使用者的名稱，您可以取得 `oc login`命令列的命令（令牌登入）。

2.為叢集建立 **kubeconfig** 檔案

遵循程序"這裡"為 ROSA 叢集建立 kubeconfig 檔案。當您將叢集新增至 ACS 時，稍後將使用此 kubeconfig 檔案。

3.在叢集上安裝**Trident**

在 ROSA 叢集上安裝Trident （最新版本）。為此，您可以按照以下任一程序進行操作"這裡"。若要從叢集控制台使用 helm 安裝Trident ，先建立一個名為Trident的專案。

然後從開發人員視圖建立一個 Helm 圖表儲存庫。對於 URL 欄位使用 `'https://netapp.github.io/trident-helm-chart'`。然後為Trident操作員創建一個掌舵版本。

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

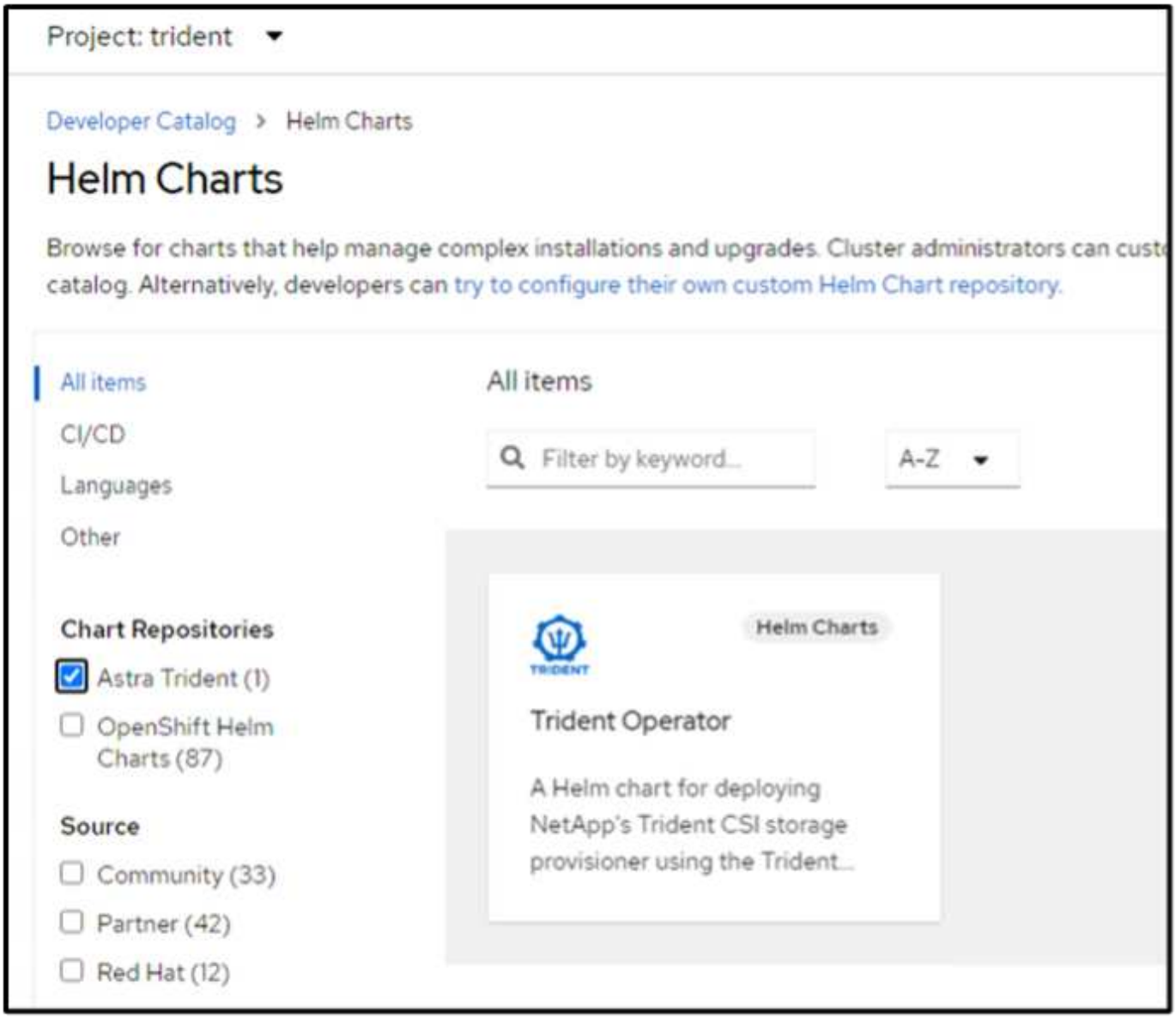
NetApp Astra Trident

A description for the Helm Chart repository.

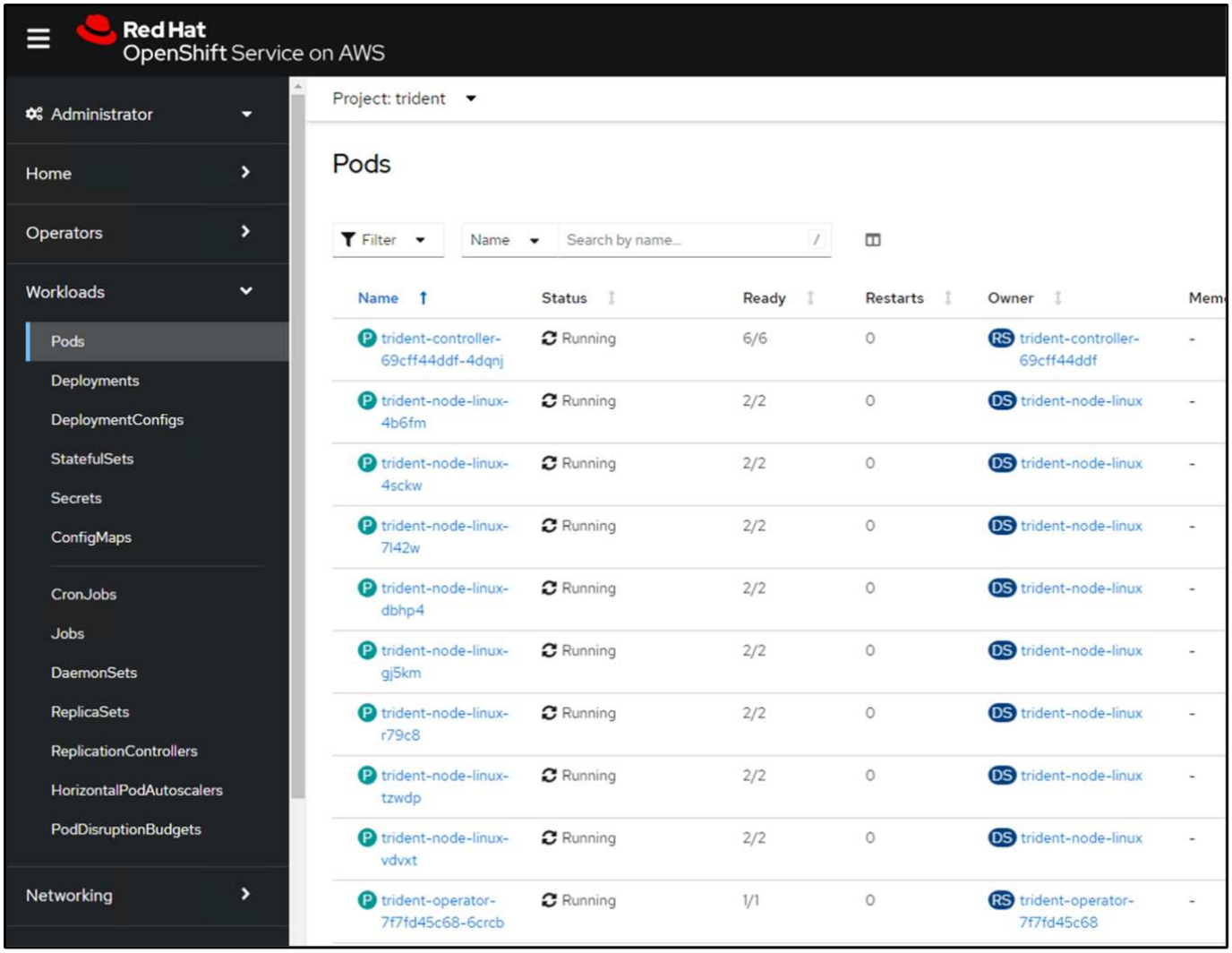
[ ] Disable usage of the repo in the developer catalog.

URL *

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



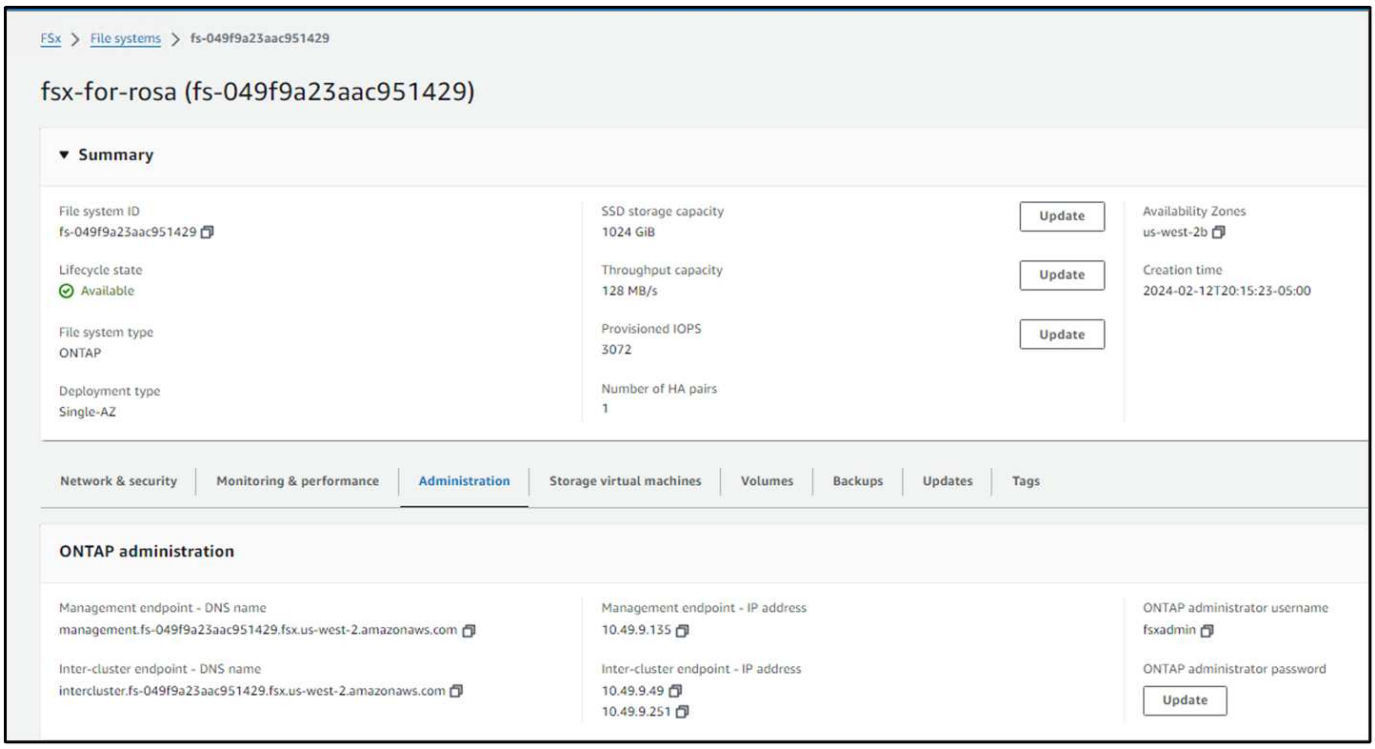
透過返回控制台上的管理員視圖並選擇 trident 專案中的 pod，驗證所有 trident pod 是否正在運行。



4.使用**Trident CSI** 設定器建立後端、儲存類別和快照類別配置

使用下面顯示的 yaml 檔案建立 trident 後端物件、儲存類別物件和 Volumesnapshot 物件。確保在後端的設定 yaml 中提供您建立的Amazon FSx ONTAP檔案系統、管理 LIF 和檔案系統的 vserver 名稱的憑證。要獲取這些詳細信息，請轉到Amazon FSx的 AWS 控制台並選擇檔案系統，導航至「管理」標籤。另外，點擊更新以設定 `fsxadmin`用戶。

> 您可以使用命令列建立對象，也可以使用混合雲控制台中的 yaml 檔案建立對象。



- Trident後端設定**

```
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
```

存儲類別

```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
```

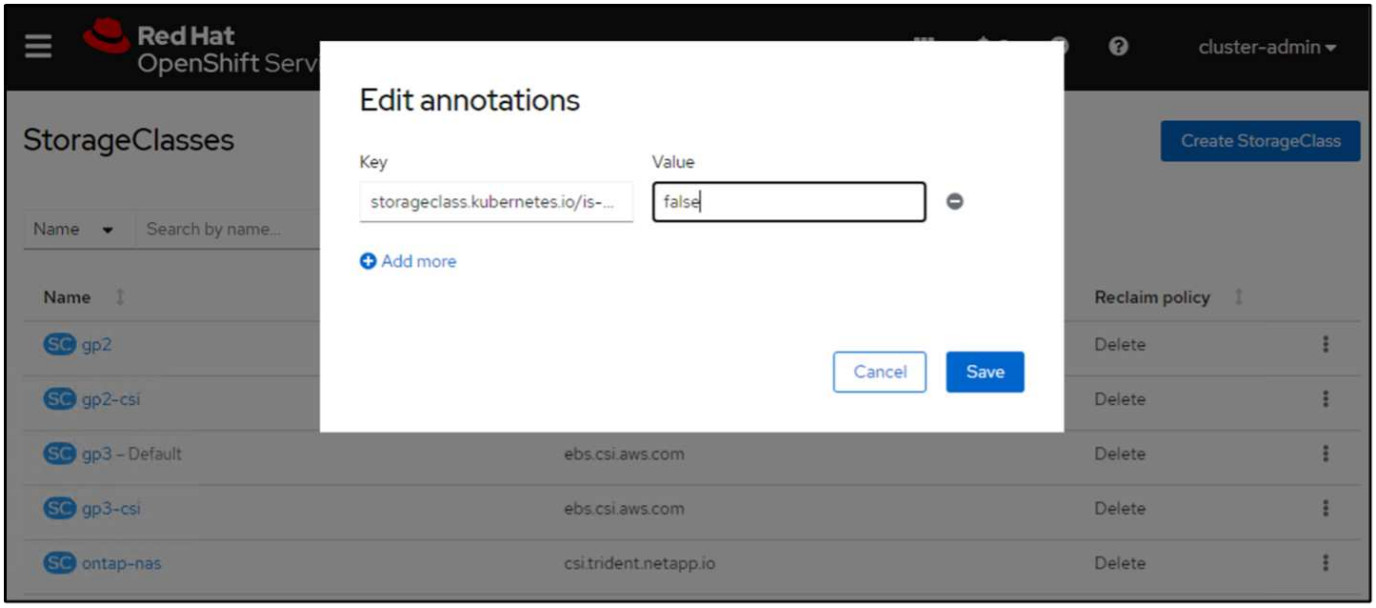
快照類別

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

透過發出下方顯示的命令來驗證後端、儲存類別和 trident-snapshotclass 物件是否已建立。

```
[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]$
```

這時候你需要做的一個重要的修改就是將 ontap-nas 設定為預設儲存類，而不是 gp3，這樣你後面部署的 postgresql app 就可以使用預設儲存類別。在叢集的 Openshift 控制台中，在「儲存」下選擇「StorageClasses」。將目前預設類別的註解編輯為 false，並為 ontap-nas 儲存類別新增註解 storageclass.kubernetes.io/is-default-class 設定為 true。



5.在叢集上部署 **postgresql** 應用程式

您可以從命令列部署應用程序，如下所示：

```
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]$
```

如果您沒有看到應用程式 pod 正在運行，那麼可能是由於安全上下文約束導致的錯誤。

```
[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]$
```

透過編輯 `runAsUser`和 `fsGroup`中的字段 `statefuleset.apps/postgresql`具有輸出中的 uid 的對象 `oc get project`命令如下圖所示。

```
[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 應用程式應該正在運行並使用Amazon FSx ONTAP儲存支援的持久性磁碟區。

```
[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]$
```

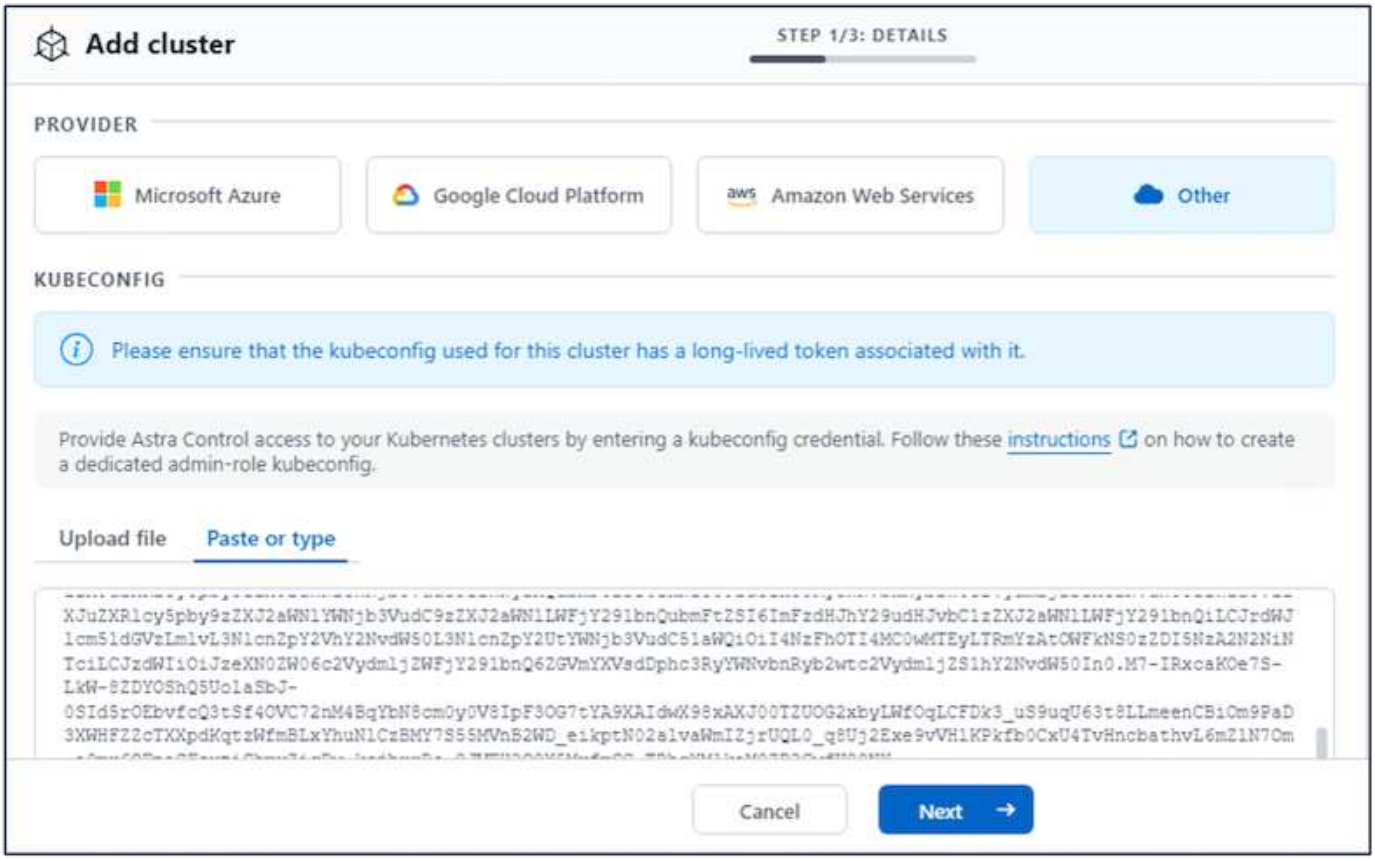
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s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

6.建立資料庫並新增記錄

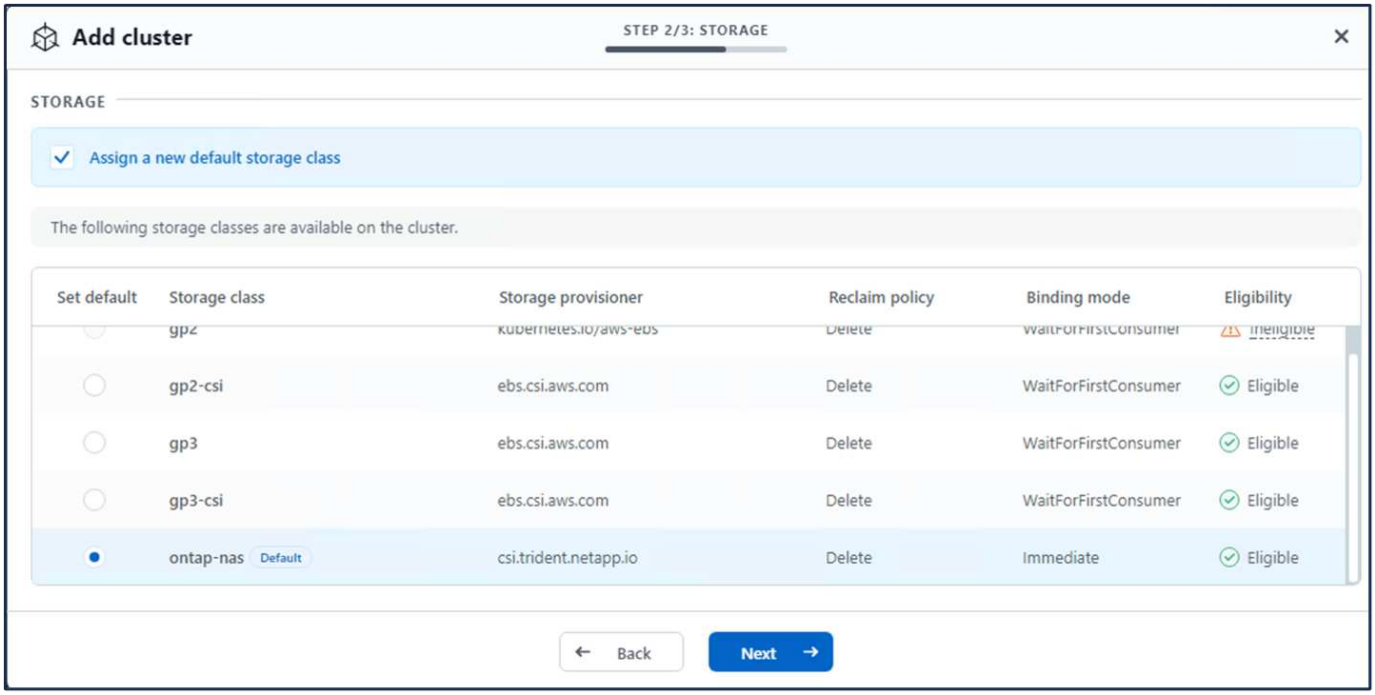
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1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

登入 ACS。選擇集群並點擊新增。選擇其他並上傳或貼上 kubeconfig 檔案。

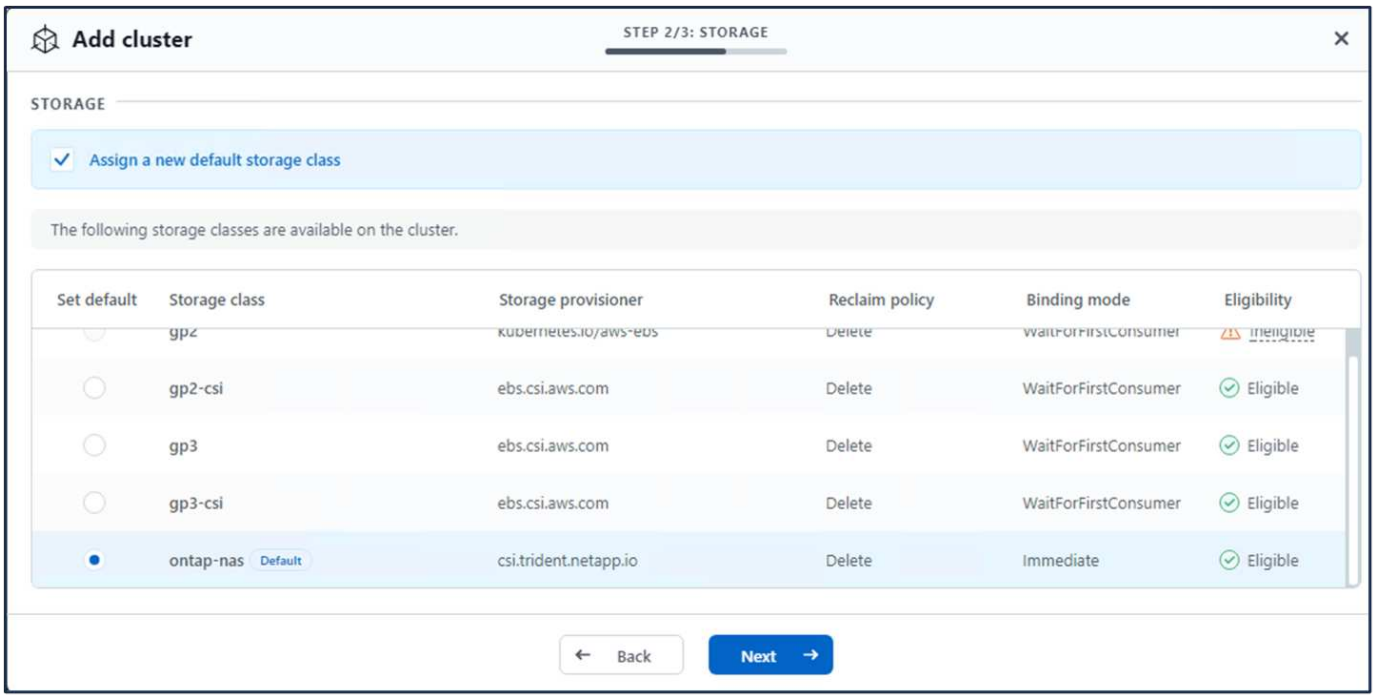


按一下“下一步”並選擇 ontap-nas 作為 ACS 的預設儲存類別。按一下“下一步”，查看詳細資料並“新增”叢集。



8.在 **ACS** 中定義應用程式

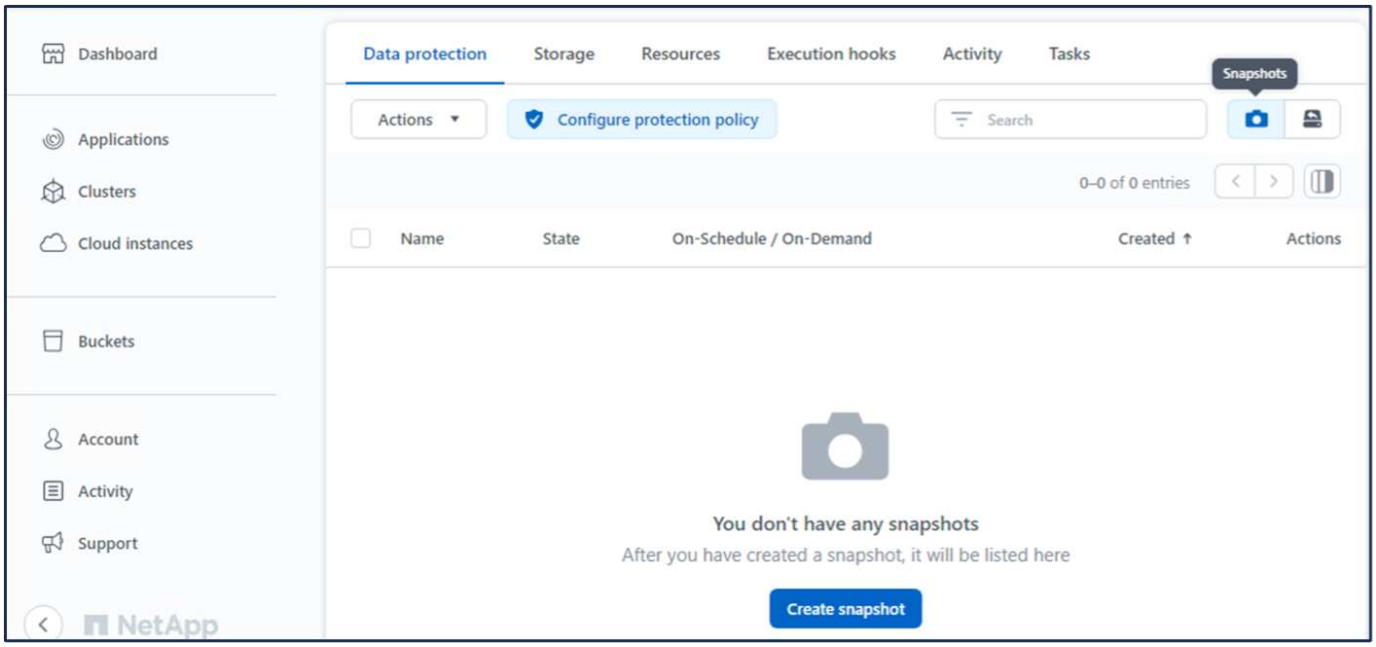
在 ACS 中定義 postgresql 應用程式。從登陸頁面，選擇*應用程式*、*定義*並填寫適當的詳細資訊。按一下“下一步”幾次，查看詳細信息，然後按一下“定義”。應用程式被加入到 ACS。

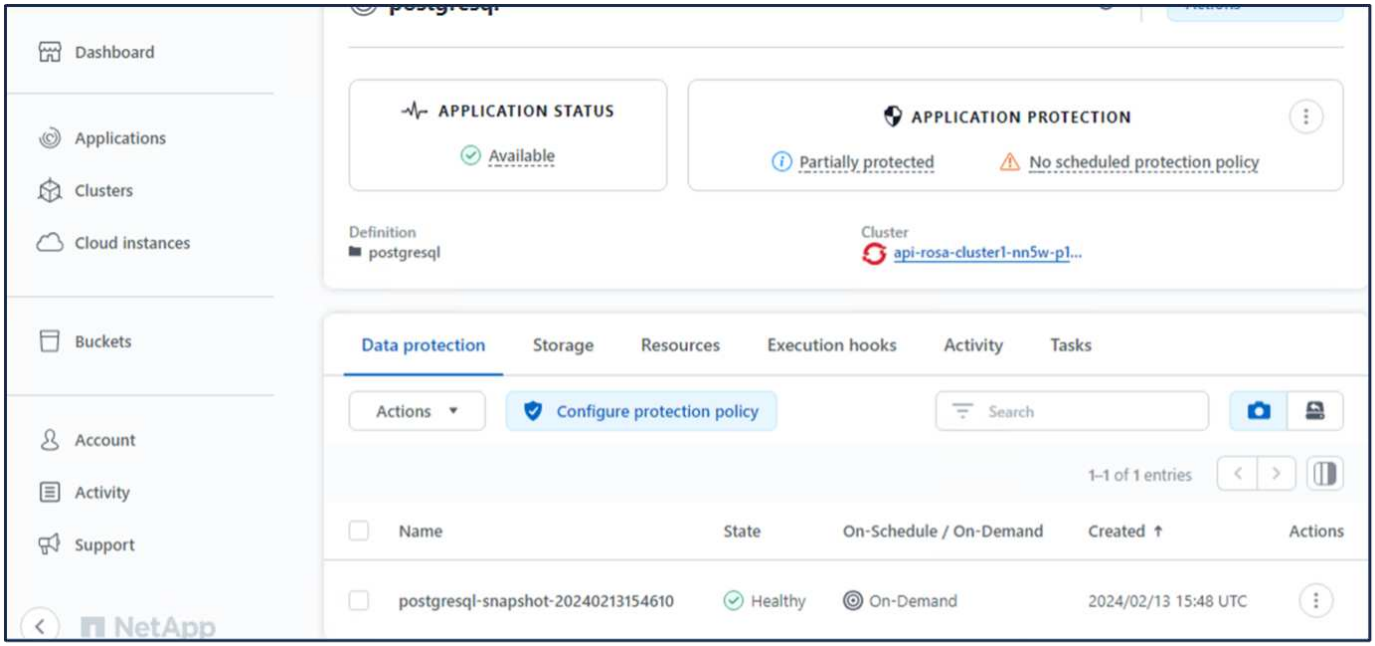


9.使用 **ACS** 建立快照

在 ACS 中建立快照的方法有很多種。您可以從顯示應用程式詳細資訊的頁面中選擇應用程式並建立快照。您可以點擊建立快照來建立按需快照或配置保護策略。

只需按一下“建立快照”，提供名稱，查看詳細信息，然後按一下“快照”即可建立按需快照。操作完成後，快照狀態變成「健康」。





10.刪除 **postgresql** 應用程式中的資料庫

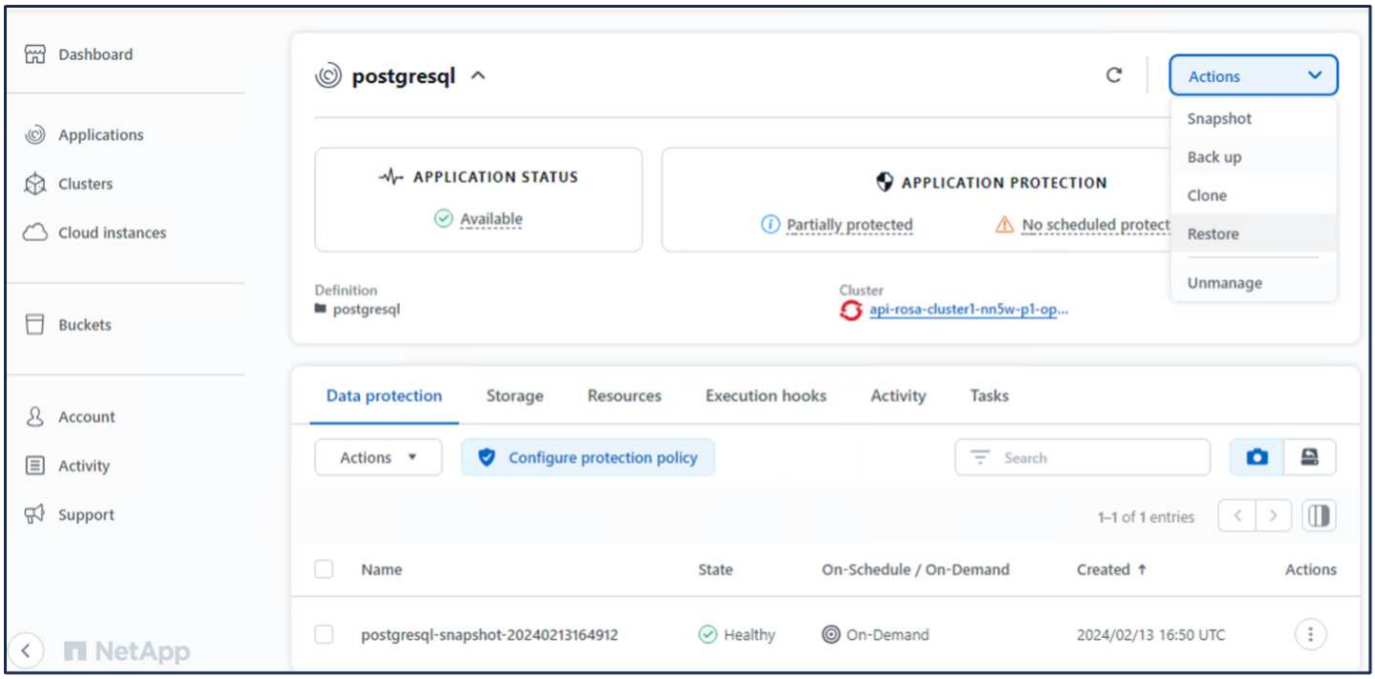
重新登入 postgresql，列出可用的資料庫，刪除先前建立的資料庫，然後再次列出以確保資料庫已被刪除。

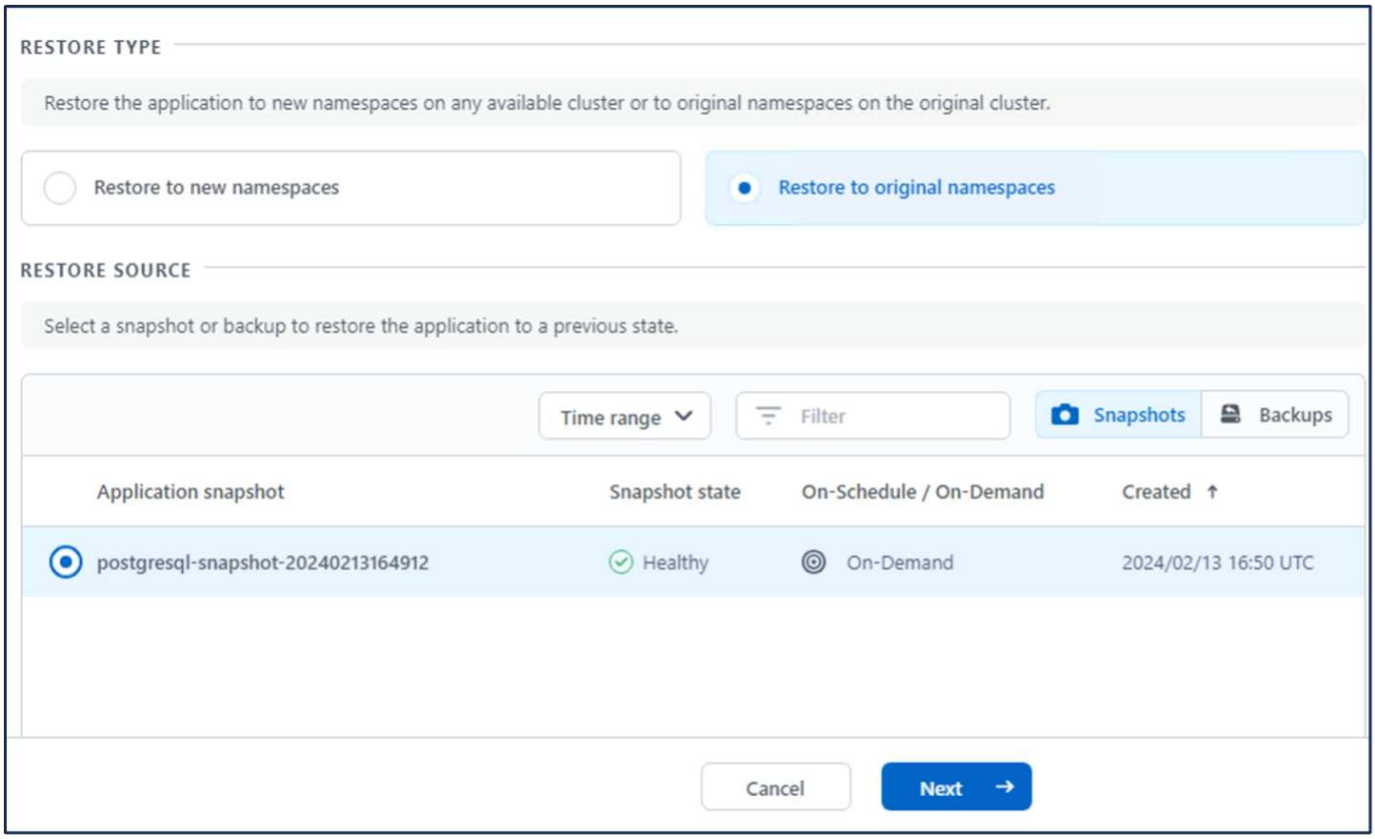
```
postgres=# \l
                                                   List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------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

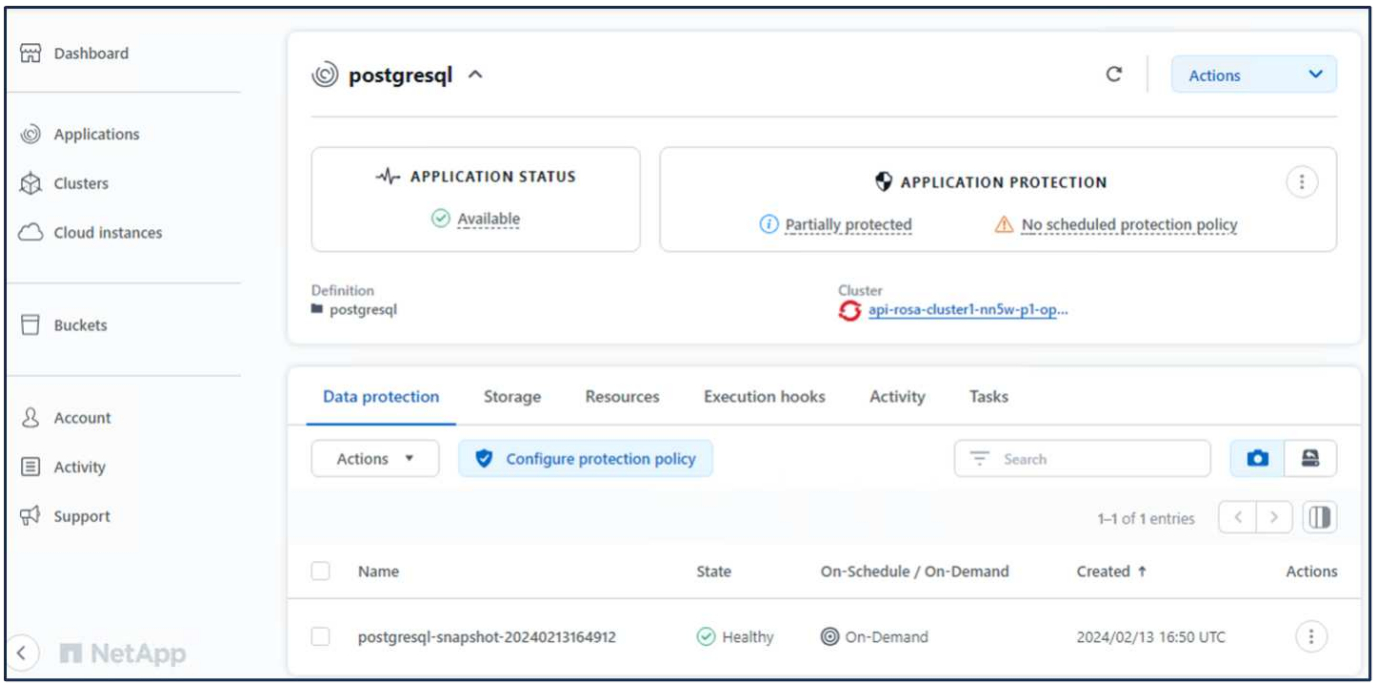
11.使用 **ACS** 從快照還原

若要從快照還原應用程序，請前往 ACS UI 登入頁面，選擇應用程式並選擇復原。您需要選擇一個快照或備份來從中還原。 （通常，您會根據已配置的策略建立多個）。在接下來的幾個畫面中做出適當的選擇，然後按一

下「恢復」。從快照恢復後，應用程式狀態從「正在恢復」變為「可用」。







12.驗證您的應用程式已從快照還原

登入 postgresql 客戶端，您現在應該看到您之前擁有的表和表中的記錄。就是這樣。只需單擊一個按鈕，您的應用程式就會恢復到以前的狀態。這就是我們利用Astra Control 為客戶提供的便利。

```
[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)
```

資料遷移

此頁面顯示使用 FSx ONTAP進行持久性儲存的託管 Red Hat OpenShift 叢集上的容器工作負載的資料遷移選項。

資料遷移

AWS 上的 Red Hat OpenShift 服務以及Amazon FSx for NetApp ONTAP （FSx ONTAP）都是 AWS 服務組合的一部分。 FSx ONTAP可在單一可用區或多可用區選項上使用。 Multi-Az 選項提供可用區域故障時的資料保護。 FSx ONTAP可以與Trident集成，為 ROSA 叢集上的應用程式提供持久性儲存。

使用 **Helm chart** 將 **FSx ONTAP**與**Trident**集成

ROSA 叢集與Amazon FSx ONTAP集成

容器應用的遷移主要包括：

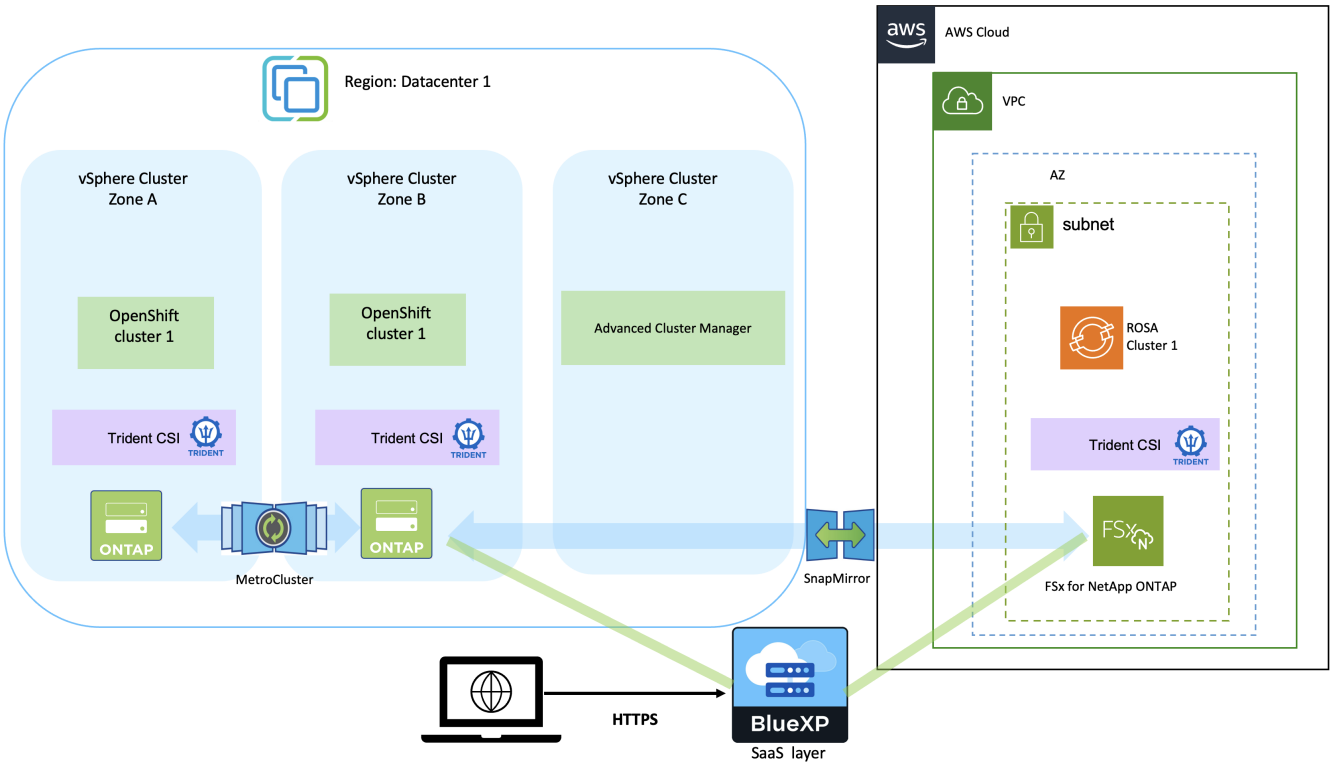
- 持久性卷：這可以使用BlueXP來實現。另一個選擇是使用Trident Protect 來處理從本地到雲端環境的容器應用程式遷移。自動化可以用於相同的目的。
- 應用程式元資料：這可以使用 OpenShift GitOps（Argo CD）來完成。

使用 **FSx ONTAP**進行 **ROSA** 叢集上應用程式的故障轉移和故障恢復，以實現持久存儲

以下影片示範了使用BlueXP和 Argo CD 的應用程式故障轉移和故障復原場景。

ROSA 叢集上應用程式的故障轉移和故障復原

OpenShift Container 工作負載的資料保護與遷移解決方案



適用於 Red Hat OpenShift 工作負載的其他NetApp混合多雲解決方案

其他解決方案

其他部分提供了其他解決方案，如下所示：

有關 Red Hat OpenShift Container 解決方案，請參閱"這裡"。

有關 Red Hat OpenShift 虛擬化解決方案，請參閱"這裡"。

版權資訊

Copyright © 2025 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。 美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 http://www.netapp.com/TM 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。