



采用提供商托管组件的混合云 NetApp Solutions

NetApp
April 12, 2024

目录

- 适用于Red Hat OpenShift容器工作负载的NetApp混合云解决方案 1
 - 概述 1
 - NetApp解决方案 在AWS上运行托管Red Hat OpenShift容器平台工作负载 2
 - 在AWS上部署和配置托管Red Hat OpenShift容器平台 3
 - 数据保护..... 6
 - 数据迁移..... 22

适用于Red Hat OpenShift容器工作负载的NetApp混合云解决方案

概述

NetApp发现、越来越多的客户正在利用围绕Kubernetes构建的容器和流程编排平台来打造现代化的传统企业级应用程序以及构建新应用程序。Red Hat OpenShift容器平台就是我们看到许多客户采用的一个示例。

随着越来越多的客户开始在企业中采用容器、NetApp已做好充分准备、可以满足有状态应用程序的持久存储需求以及数据保护、数据安全和数据迁移等传统数据管理需求。但是、可以使用不同的策略、工具和方法来满足这些需求。

****NetApp ONTAP 基于下面列出的存储选项，可为容器和部署提供安全性、数据保护、可靠性和灵活性。**

- 内部环境中的自行管理存储：
 - NetApp光纤连接存储(FAS)、NetApp全闪存FAS 阵列(AFF)、NetApp全SAN阵列(ASA)和ONTAP Select
- 内部部署中由提供商管理的存储：
 - NetApp Keystone 提供存储即服务(STaaS)
- 云中的自行管理存储：
 - NetApp Cloud Volumes ONTAP (CVO)可在超大容量云中提供自行管理的存储
- 云中由提供商管理的存储：
 - Cloud Volumes Service for Google Cloud (CVS)、Azure NetApp Files (ANF)、Amazon FSx for NetApp ONTAP 可在超云中提供完全托管的存储

ONTAP feature highlights



Storage Administration <ul style="list-style-type: none">• Multi-tenancy• FlexVol & FlexGroup• LUN• Quotas• ONTAP CLI & API• System Manager & BlueXP	Performance & Scalability <ul style="list-style-type: none">• FlexCache• FlexClone• nconnect, session trunking, multipathing• Scale-out clusters
Availability & Resilience <ul style="list-style-type: none">• Multi-AZ HA deployment (MetroCluster)• SnapShot & SnapRestore• SnapMirror• SnapMirror Business Continuity• SnapMirror Cloud	Access Protocols <ul style="list-style-type: none">• NFS –v3, v4, v4.1, v4.2• SMB – v2, v3• iSCSI• Multi-protocol access
Storage Efficiency <ul style="list-style-type: none">• Deduplication & Compression• Compaction• Thin provisioning• Data Tiering (Fabric Pool)	Security & Compliance <ul style="list-style-type: none">• Fpolicy & Vscan• Active Directory integration• LDAP & Kerberos• Certificate based authentication

NetApp BlueXP使您能够从一个控制平台/接口管理所有存储和数据资产。

您可以使用BlueXP创建和管理云存储(例如Cloud Volumes ONTAP 和Azure NetApp Files)、移动、保护和分析数据以及控制许多内部和边缘存储设备。

NetApp Asta Trident是一款符合CSI的存储编排程序，支持快速、轻松地使用由上述各种NetApp存储选项提供支持的永久性存储。它是由NetApp维护和支持的开源软件。



Astra Trident CSI feature highlights

CSI specific <ul style="list-style-type: none">• CSI NetApp® Snapshot™ copies and volume creation from CSI Snapshot copies• CSI topology• Volume expansion	Security <ul style="list-style-type: none">• Dynamic-export policy management• iSCSI initiator-groups dynamic management• iSCSI bidirectional CHAP
Control <ul style="list-style-type: none">• Storage and performance consumption• Monitoring• Volume Import• Cross Namespace Volume Access	Installation methods <ul style="list-style-type: none">• Binary• Helm chart• Operator• GitOps
Choose your access mode <ul style="list-style-type: none">• RWO (ReadWriteOnce, i.e 1↔1)• RWX (ReadWriteMany, i.e 1↔n)• ROX (ReadOnlyMany)• RWOP (ReadWriteOnce POD)	Choose your protocol <ul style="list-style-type: none">• NFS• SMB• iSCSI

业务关键型容器工作负载所需的不仅仅是永久性卷。他们的数据管理要求也需要保护和迁移应用程序Kubernetees对象。



除了用户数据之外、应用程序数据还包括Kubernetees对象：以下是一些示例： - Kubernetees对象、例如Pod规格、PVC、部署、服务-自定义配置对象、例如配置映射和密钥-持久数据、例如Snapshot副本、备份、克隆-自定义资源、例如CRS和CRD

NetApp Asta Control作为完全托管和自我管理软件提供，可提供流程编排，实现强大的应用程序数据管理。请参见 ["Astra 文档"](#) 有关Asta系列产品的更多详细信息。

本参考文档使用NetApp Asta Control Center验证了在RedHat OpenShift容器平台上部署的基于容器的应用程序的迁移和保护。此外、解决方案 还提供了有关部署和使用Red Hat高级集群管理(ACM)来管理容器平台的详细信息。本文档还重点介绍了使用Asta Trident CSI配置程序将NetApp存储与Red Hat OpenShift容器平台集成的详细信息。Astra Control Center部署在集线器集群上、用于管理容器应用程序及其永久性存储生命周期。最后、它还提供了一个解决方案、用于在AWS (ROSA)中使用Amazon FSx for NetApp ONTAP (FSxN)作为永久性存储的受管Red Hat OpenShift集群上对容器工作负载进行复制、故障转移和故障恢复。

NetApp解决方案 在AWS上运行托管Red Hat OpenShift容器平台工作负载

NetApp解决方案 在AWS上运行托管Red Hat OpenShift容器平台工作负载

客户可能"生于云"、也可能正处于现代化之旅的某一时刻、准备将部分选定工作负载或所

有工作负载从数据中心迁移到云。他们可以选择在云中使用提供商管理的OpenShift容器和提供商管理的NetApp存储来运行工作负载。他们应该在云中规划和部署托管Red Hat OpenShift容器集群(ROSA)、以便为其容器工作负载提供一个成功的生产就绪环境。在AWS云中、他们还可以部署FSx for NetApp ONTAP 来满足存储需求。

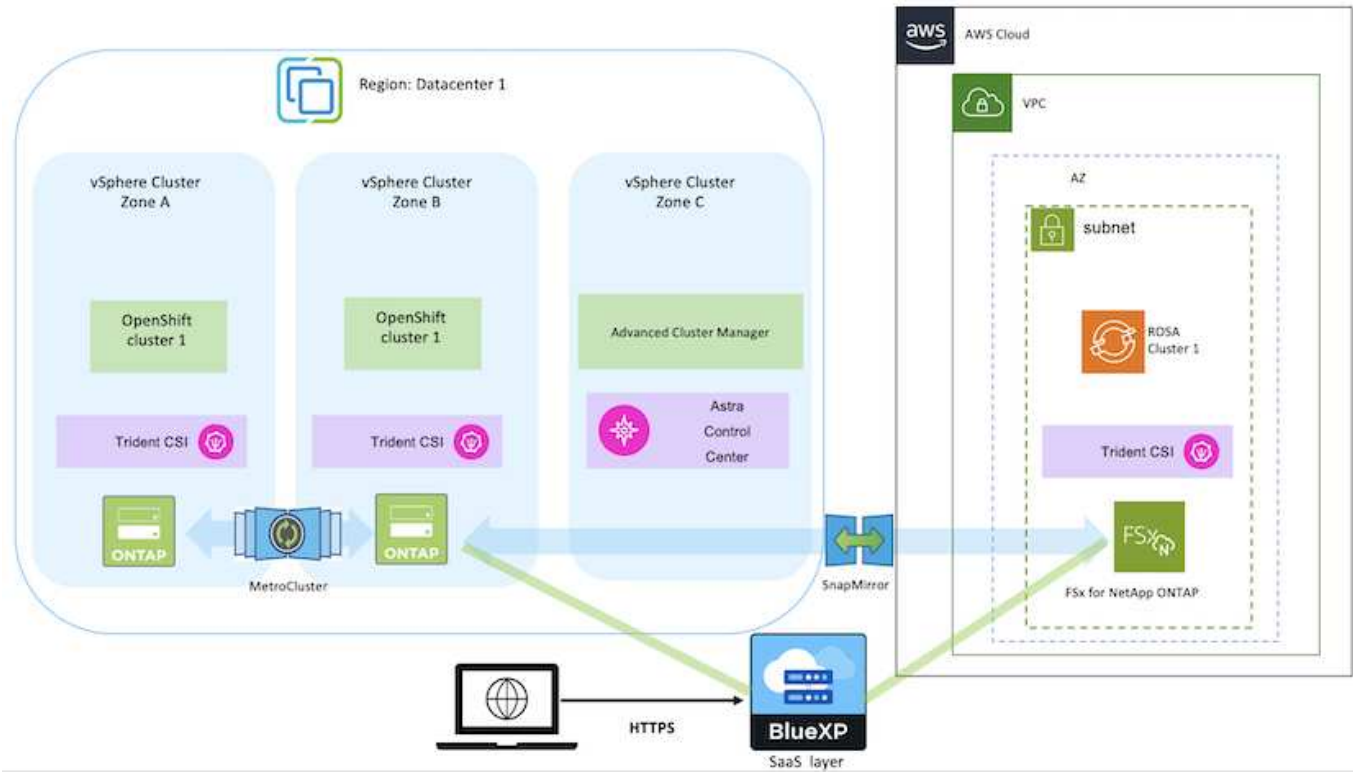
FSx for NetApp ONTAP 可为AWS中的容器部署提供数据保护、可靠性和灵活性。Asta三端存储作为动态存储配置程序、用于为客户的有状态应用程序使用永久性FSxN存储。

由于可以在HA模式下部署ROSA、并且控制平台节点分布在多个可用性区域中、因此FSx ONTAP 还可以配置Multi-AZ选项、以提高可用性并防止出现AZ故障。



从文件系统的首选可用性区域(AZ)访问Amazon FSx文件系统时、无需支付数据传输费用。有关定价的详细信息、请参见 ["此处"](#)。

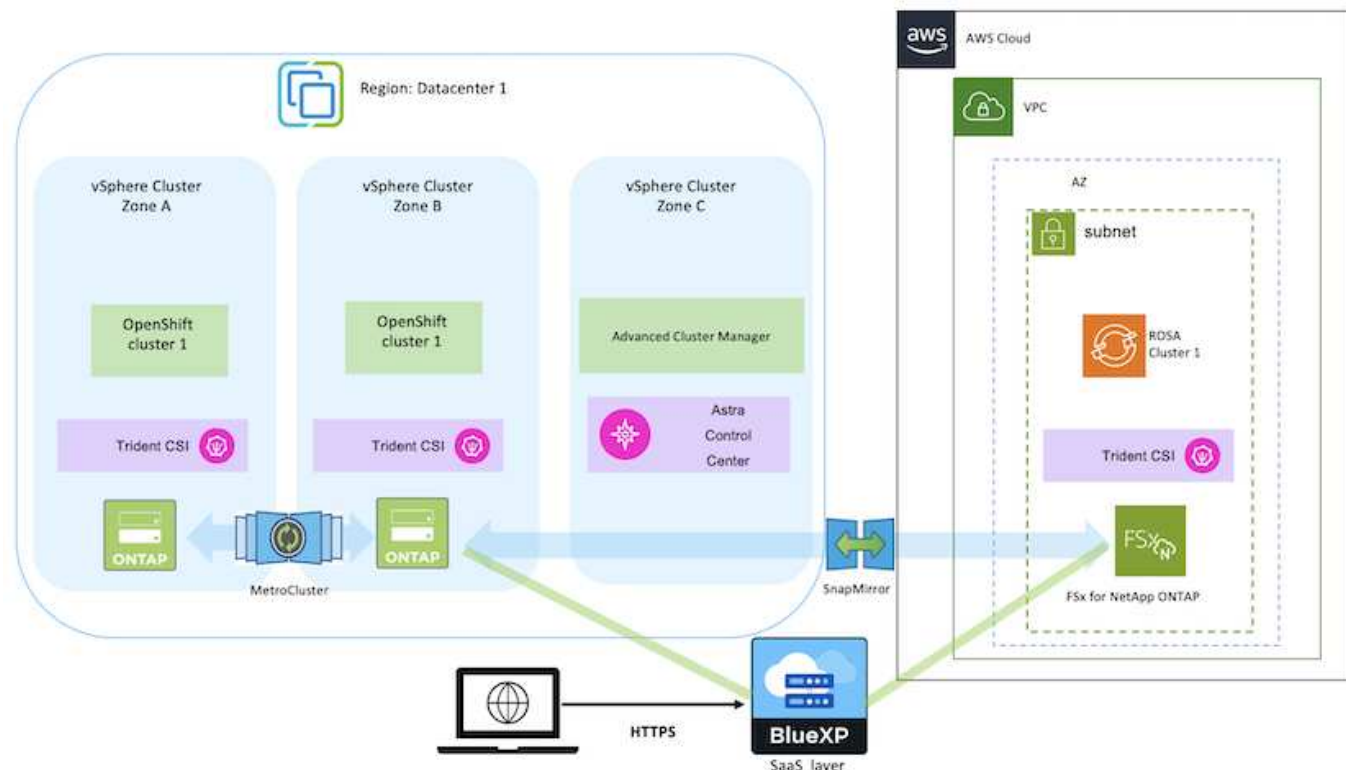
适用于**OpenShift**容器工作负载的数据保护和迁移解决方案



在**AWS**上部署和配置托管**Red Hat OpenShift**容器平台

本节简要介绍了在AWS (ROSA)上设置托管Red Hat OpenShift集群的工作流。其中显示了Asta三端存储使用托管FSx for NetApp ONTAP (FSxN)作为存储后端来提供永久性卷。其中详细介绍了如何使用BlueXP在AWS上部署FSxN。此外、还提供了有关使用BlueXP和OpenShift GitOps (Argo CD)为ROSA集群上有状态应用程序执行数据保护和迁移活动的详细信息。

下图展示了在AWS上部署并使用FSxN作为后端存储的ROSA集群。



此解决方案 已通过在AWS中的两个VPC中使用两个ROSA集群进行验证。每个ROSA集群都使用Astra Trdent与FSxN集成。可以通过多种方法在AWS中部署ROSA集群和FSxN。此高级设置问题描述 提供了所用特定方法的文档链接。您可以在中提供的相关链接中参考其他方法 ["资源部分"](#)。

设置过程可细分为以下步骤：

安装ROSA集群

- 创建两个VPC并在VPC之间设置VPC对等连接。
- 请参见 ["此处"](#) 有关安装ROSA集群的说明。

安装FSxN

- 从BlueXP在vPC上安装FSxN。请参见 ["此处"](#) 以便创建BlueXP帐户并开始使用。请参见 ["此处"](#) 用于安装FSxN。请参见 ["此处"](#) 用于在AWS中创建连接器以管理FSxN。
- 使用AWS部署FSxN。请参见 ["此处"](#) 适用于使用AWS控制台进行部署。

在ROSA集群上安装TRIDent (使用Helm图表)

- 使用Helm图表在ROSA集群上安装三端存储。Helm图表的URL: <https://netapp.github.io/trident-helm-chart>

将FSxN与适用于ROSA集群的Asta Trident集成



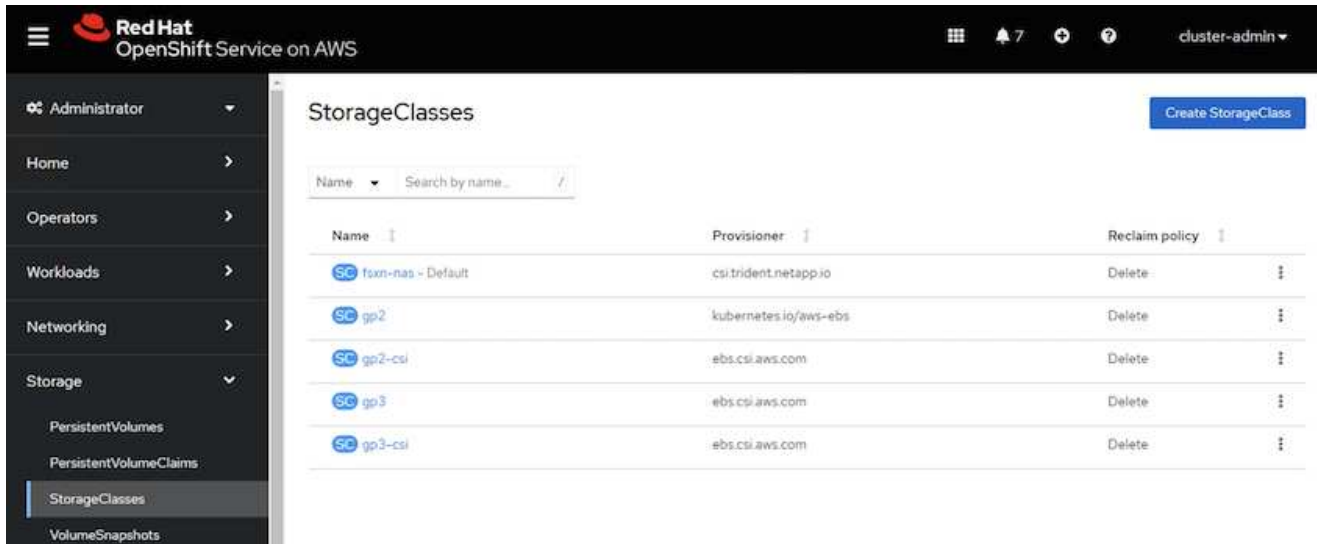
当所有受管集群使用ApplicationSet注册到ArgoCD时、可以使用OpenShift GitOps将Asta Trident CSI部署到这些集群。

```
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创建后端和存储类(适用于FSxN)

- 请参见 ["此处"](#) 有关创建后端和存储类的详细信息、请参见。
- 从OpenShift控制台使用默认的三端CSI为FsxN创建存储类。请参见以下屏幕截图：



使用OpenShift GitOps部署应用程序(Argo CD)

- 在集群上安装OpenShift GitOps Operator。请参阅说明 ["此处"](#)。
- 为集群设置新的Argo CD实例。请参阅说明 ["此处"](#)。

打开Argo CD的控制台并部署应用程序。例如、您可以使用带有Helm Chart的Argo CD部署Jenkins应用程序。创建应用程序时、系统会提供以下详细信息：Project：default cluster： <https://kubernetes.default.svc> 命名空间：Jenkins Helm图表的URL： <https://charts.bitnami.com/bitnami>

Helm参数：globL.storageClass：fsxn-nas

数据保护

此页面显示了使用Astra Control Service在AWS上托管Red Hat OpenShift (ROSA)集群的数据保护选项。Astra Control Service (ACS)提供了一个易于使用的图形用户界面、您可以使用该界面添加集群、定义在其中运行的应用程序以及执行应用程序感知型数据管理活动。此外、还可以使用支持工作流自动化的API访问ACS功能。

NetApp Astra控制(ACS或ACC)由Astra三端驱动。Astra三端集成了多种类型的Kubernetes集群、例如Red Hat OpenShift、EKS、AKS、SUSE缓存器、Anthos等。具有各种NetApp ONTAP存储风格、例如FAS/AFFF、ONTAP Select、CVO、Google Cloud Volumes Service、Azure NetApp Files和Amazon FSx for NetApp ONTAP。

本节详细介绍了使用ACS的以下数据保护选项：

- 显示备份和还原在一个区域运行的ROSA应用程序并还原到另一个区域的视频。

- 显示ROSA应用程序的Snapshot和Restore的视频。
- 安装ROSA集群的分步详细信息、Amazon FSx for NetApp ONTAP、使用NetApp Astra三端集成到存储后端、在ROSA集群上安装PostgreSQL应用程序、使用ACS创建应用程序快照并从中还原应用程序。
- 一篇博客、详细介绍了如何使用ACS在使用FSx for ONTAP的ROSA集群上为mysql应用程序创建快照并从快照中还原。

备份/从备份中还原

以下视频显示了在一个区域运行的ROSA应用程序的备份以及还原到另一个区域的过程。

[FSx NetApp ONTAP for Red Hat OpenShift Service on AWS](#)

快照/从快照还原

以下视频显示了如何创建ROSA应用程序的快照以及之后如何从快照中还原。

[使用Amazon FSx for NetApp ONTAP存储在AWS上的Red Hat OpenShift Service \(ROSA\)集群上为应用程序创建快照/还原](#)

博客

- ["使用Astra Control Service对带有Amazon FSx存储的ROSA集群上的应用程序进行数据管理"](#)

创建快照并从中还原的分步详细信息

前提条件设置

- ["AWS 帐户"](#)
- ["Red Hat OpenShift帐户"](#)
- 使用的IAM用户 ["适当的权限"](#) 创建和访问ROSA集群
- ["AWS命令行界面"](#)
- ["罗莎命令行界面"](#)
- ["OpenShift命令行界面"\(OC\)](#)
- 具有子网以及相应网关和路由的VPC
- ["已安装罗莎群集" VPC](#)
- ["适用于 NetApp ONTAP 的 Amazon FSX"](#) 在同一个VPC中创建
- 从访问ROSA集群 ["OpenShift混合云控制台"](#)

后续步骤

1. 创建管理员用户并登录到集群。
2. 为集群创建一个kubeconfig.文件。
3. 在集群上安装Astra Trident。

4. 使用三端CSI配置程序创建后端、存储类和快照类配置。
5. 在集群上部署PostgreSQL应用程序。
6. 创建数据库并添加记录。
7. 将集群添加到ACS中。
8. 在ACS中定义应用程序。
9. 使用ACS创建快照。
10. 删除PostgreSQL应用程序中的数据库。
11. 使用ACS从快照还原。
12. 验证您的应用程序是否已从快照中还原。

1.创建管理员用户并登录到群集

使用以下命令创建管理员用户以访问ROSA集群：(只有在安装时未创建管理员用户时、才需要创建管理员用户)

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

此命令将提供如下输出。使用登录到集群 `oc login` 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录到集群。如果您在创建集群时已创建管理员用户、则可以使用管理员用户凭据从Red Hat OpenShift Hybrid Cloud控制台登录到集群。然后、通过单击右上角显示已登录用户名称的、您可以获取 `oc login` 命令(令牌登录)。

2.为群集创建kubecfg文件

按照步骤进行操作 "[此处](#)" 为ROSA集群创建kubecfg.稍后在将集群添加到ACS中时、将使用此kubecfg.文件。

3.在群集上安装Asta Trident

在ROSA集群上安装Asta Trident (最新版本)。要执行此操作、您可以按照给定的任一过程进行操作 "[此处](#)"。要从集群控制台使用Helm安装Trident、请先创建一个名为Trident的项目。

Red Hat

OpenShift Service on AWS

2

cluster-admin

Projects

Create Project

Filter

Name

trident

Name

trident

Clear all filters

Name	Display name	Status	Requester	Created
<div>PR</div> trident	trident	<div>Active</div>	rosaadmin	<div>Feb 12, 2024, 9:54 PM</div>

然后、在"开发工具"视图中、创建Helm图表存储库。对于URL字段、请使用
'https://netapp.github.io/trident-helm-chart'。然后为三端操作员创建舵版本。

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

Project: trident ▼

Developer Catalog > Helm Charts

Helm Charts

Browse for charts that help manage complex installations and upgrades. Cluster administrators can customize the catalog. Alternatively, developers can [try to configure their own custom Helm Chart repository](#).

All items

CI/CD

Languages

Other

Chart Repositories

☒ Astra Trident (1)

☐ OpenShift Helm Charts (87)

Source

☐ Community (33)


☐ Partner (42)

☐ Red Hat (12)

All items

Q Filter by keyword...

A-Z ▼

 Helm Charts

Trident Operator

A Helm chart for deploying NetApp's Trident CSI storage provisioner using the Trident...

返回控制台上的"Administrator view"(管理员视图)、然后在三级工程中选择Pod、以验证所有三级工程模块是否正在运行。

11

Red Hat
 OpenShift Service on AWS

Administrator

Home

Operators

Workloads

Pods
 Deployments
 DeploymentConfigs
 StatefulSets
 Secrets
 ConfigMaps
 CronJobs
 Jobs
 DaemonSets
 ReplicaSets
 ReplicationControllers
 HorizontalPodAutoscalers
 PodDisruptionBudgets

Networking

Project: trident

Pods

Filter

Name

Search by name...

Name	Status	Ready	Restarts	Owner	Memory
trident-controller-69cff44ddf-4dqnj	Running	6/6	0	trident-controller-69cff44ddf	-
trident-node-linux-4b6fm	Running	2/2	0	trident-node-linux	-
trident-node-linux-4sckw	Running	2/2	0	trident-node-linux	-
trident-node-linux-7142w	Running	2/2	0	trident-node-linux	-
trident-node-linux-dbhp4	Running	2/2	0	trident-node-linux	-
trident-node-linux-gj5km	Running	2/2	0	trident-node-linux	-
trident-node-linux-r79c8	Running	2/2	0	trident-node-linux	-
trident-node-linux-tzwdp	Running	2/2	0	trident-node-linux	-
trident-node-linux-vdvxt	Running	2/2	0	trident-node-linux	-
trident-operator-7f7fd45c68-6crcb	Running	1/1	0	trident-operator-7f7fd45c68	-

4.使用三端CSI配置程序创建后端、存储类和快照类配置

使用下面显示的YAML文件创建三元后端对象、存储类对象和卷快照对象。请务必为您创建的Amazon FSx for NetApp ONTAP文件系统提供凭据、并在后端的YAML配置中提供管理LIF和文件系统的Vserver名称。要获取这些详细信息、请转到适用于Amazon FSx的AWS控制台并选择文件系统、然后导航到管理选项卡。此外、单击更新以设置的密码 `fsxadmin` 用户。



您可以使用命令行创建对象、也可以从混合云控制台使用YAML文件创建对象。

FSx > File systems > fs-049f9a23aac951429

fsx-for-rosa (fs-049f9a23aac951429)

▼ Summary

File system ID fs-049f9a23aac951429	SSD storage capacity 1024 GiB	<button>Update</button>	Availability Zones us-west-2b
Lifecycle state Available	Throughput capacity 128 MB/s	<button>Update</button>	Creation time 2024-02-12T20:15:23-05:00
File system type ONTAP	Provisioned IOPS 3072	<button>Update</button>	
Deployment type Single-AZ	Number of HA pairs 1		

Network & security | Monitoring & performance | **Administration** | Storage virtual machines | Volumes | Backups | Updates | Tags

ONTAP administration

Management endpoint - DNS name management.fs-049f9a23aac951429.fsx.us-west-2.amazonaws.com	Management endpoint - IP address 10.49.9.135	ONTAP administrator username fsxadmin
Inter-cluster endpoint - DNS name intercluster.fs-049f9a23aac951429.fsx.us-west-2.amazonaws.com	Inter-cluster endpoint - IP address 10.49.9.49	ONTAP administrator password <button>Update</button>
	10.49.9.251	

Trident后端配置

```

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

```

存储类


```

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

```

快照类

```

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

```

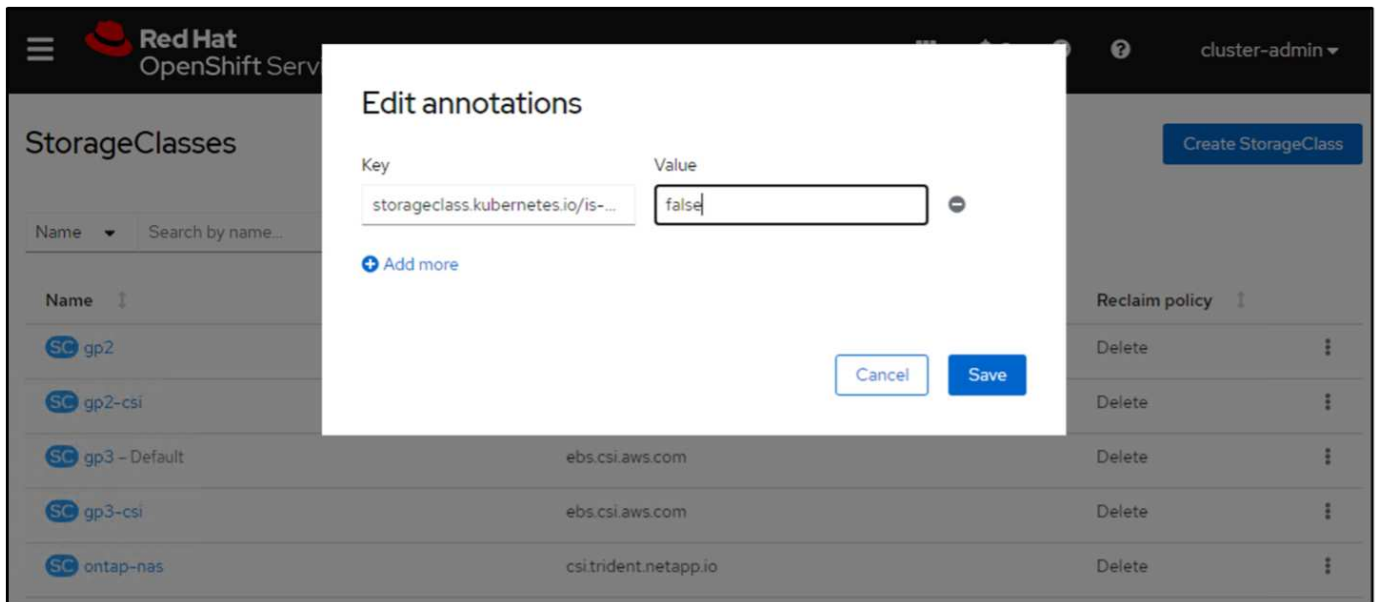
发出下面所示的命令、验证是否已创建后端、存储类和trident-snapshotclass对象。

```

[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]$

```

此时、您需要进行的一项重要修改是将ONTAP NAS设置为默认存储类、而不是GP3、以便您稍后部署的PostgreSQL应用程序可以使用默认存储类。在集群的OpenShift控制台中、在"Storage"下选择"StorageClasses"。将当前默认类的标注编辑为false、并将ONTAP NAS存储类的标注storageclass.Kubernetes.io/is-default-class设置为true。



Name	Provisioner	Reclaim policy
SC gp2	kubernetes.io/aws-ebs	Delete
SC gp2-csi	ebs.csi.aws.com	Delete
SC gp3	ebs.csi.aws.com	Delete
SC gp3-csi	ebs.csi.aws.com	Delete
SC ontap-nas - Default	csi.trident.netapp.io	Delete

5.在群集上部署PostgreSQL应用程序

您可以从命令行部署此应用程序、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:
BASIC CHART NAME: postgresql
BASIC CHART VERSION: 14.0.4
BASIC 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="PGPASSWORD=$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]$
```

如果您看不到应用程序Pod正在运行、则可能是由于安全上下文约束而导致错误。

```
[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]$
```

编辑以修复此错误 runAsUser 和 fsGroup 中的字段 statefulset.apps/postgresql 具有的输出中的_id的对象 oc get project postgresql 命令、如下所示。

```
[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应用程序应正在运行、并使用Amazon FSx支持的永久性卷作为NetApp ONTAP存储。

```
[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.创建数据库并添加记录

```
[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:vl.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.将集群添加到ACS中

登录到ACS。选择cluster、然后单击Add。选择其他并上传或粘贴kubecnfig。

Add cluster

STEP 1/3: DETAILS

PROVIDER

Microsoft Azure

Google Cloud Platform

Amazon Web Services

Other

KUBECONFIG

Please ensure that the kubeconfig used for this cluster has a long-lived token associated with it.

Provide Astra Control access to your Kubernetes clusters by entering a kubeconfig credential. Follow these [instructions](#) on how to create a dedicated admin-role kubeconfig.

Upload file

Paste or type

```
XJu2XR1cy5phy9zZXJ2aWN1YWNjb3VudC9zZXJ2aWN1LWFjY291bnQubmFtZSI6ImFzdHJhY29udHJvbC1zZXJ2aWN1LWFjY291bnQ1LCJrdWJ1cm5ldGVzLmlvL3N1cnZpY2VhY2NvdW50L3N1cnZpY2UuYWNjb3VudC51aWQiOiI4NzFhOTI4MC0wMTEyLTRmYzAtOWFkNS0zZDI5NzA2N2NiInR0LCJzdWIiOiJzeXN0ZW06c2VydmljZWVjY291bnQ6ZGVmYXVudDphc3RyYWNvbnRyb2wtc2VydmljZS1hY2NvdW50In0.M7-IRxcaKOe7S-LkW-8ZDYOShQ5UolaSbJ-0SIdSrOEbvfcQ3tSf40VC72nM4BqYbN8cm0y0V8IpF3OG7tYA9XAIdwX98xAXJ00T2UOG2xbyLWfOqLCFDk3_uS9uqU63t8LLmeenCBiOm9PaD3XWHF2ZcTXXpdKqtzWfmbLxYhuN1CzBMY7S55MVnB2WD_eikptN02alvaWmIZjrUQL0_q8Uj2Exe9vVH1KPkb0CxU4TvHncbathvL6mZ1N7Om
```

Cancel

Next →

单击*Next*并选择ONTAP—NAS作为ACS的默认存储类。单击*Next*(下一步*)，查看详细信息，然后单击*Add*(添加)群集。

Add cluster

STEP 2/3: STORAGE

STORAGE

☒
Assign a new default storage class

The following storage classes are available on the cluster.

Set default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligibility
<input type="radio"/>	gp2	kubernetes.io/aws-ebs	Delete	WaitForFirstConsumer	Ineligible
<input type="radio"/>	gp2-csi	ebs.csi.aws.com	Delete	WaitForFirstConsumer	Eligible
<input type="radio"/>	gp3	ebs.csi.aws.com	Delete	WaitForFirstConsumer	Eligible
<input type="radio"/>	gp3-csi	ebs.csi.aws.com	Delete	WaitForFirstConsumer	Eligible
<input checked="" type="radio"/>	ontap-nas Default	csi.trident.netapp.io	Delete	Immediate	Eligible

← Back

Next →

8.在ACS中定义应用程序

在ACS中定义PostgreSQL应用程序。在登录页面中，选择*Applications*、**Define***并填写相应的详细信息。单击“*下一步”几次，查看详细信息，然后单击“定义”。应用程序将添加到ACS。

Add cluster

STEP 2/3: STORAGE

×

STORAGE

☒ Assign a new default storage class

The following storage classes are available on the cluster.

Set default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligibility
<input type="radio"/>	gp2	kubernetes.io/aws-ebs	Delete	WaitForFirstConsumer	Ineligible
<input type="radio"/>	gp2-csi	ebs.csi.aws.com	Delete	WaitForFirstConsumer	Eligible
<input type="radio"/>	gp3	ebs.csi.aws.com	Delete	WaitForFirstConsumer	Eligible
<input type="radio"/>	gp3-csi	ebs.csi.aws.com	Delete	WaitForFirstConsumer	Eligible
<input checked="" type="radio"/>	ontap-nas <small>Default</small>	csi.trident.netapp.io	Delete	Immediate	Eligible

← Back

Next →

9.使用ACS创建快照

可通过多种方法在ACS中创建快照。您可以从显示应用程序详细信息的页面中选择应用程序并创建快照。您可以单击创建快照来创建按需快照或配置保护策略。

只需单击*创建快照*、提供名称、查看详细信息并单击*快照*、即可创建按需快照。操作完成后、快照状态将更改为"运行状况良好"。

Dashboard

Applications

Clusters

Cloud instances

Buckets

Account

Activity

Support

NetApp

Data protection

Storage

Resources

Execution hooks

Activity

Tasks

Actions

Configure protection policy

Search

Snapshots

0-0 of 0 entries

☐

Name

State

On-Schedule / On-Demand

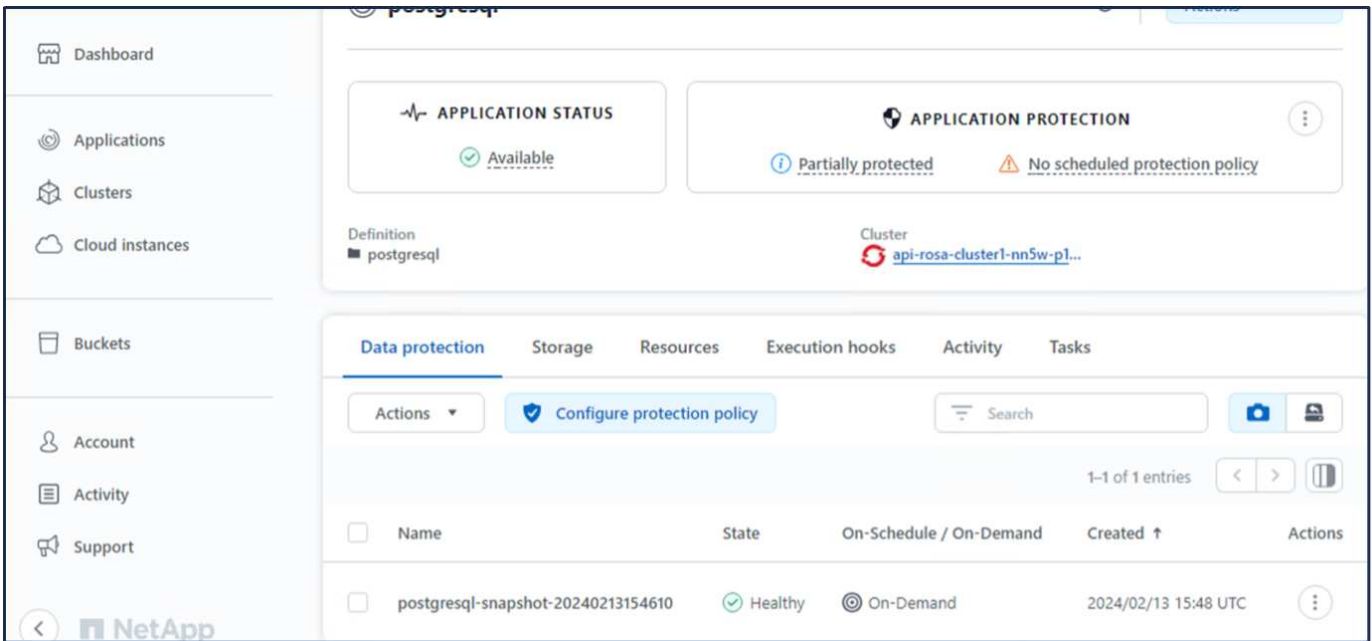
Created ↑

Actions

You don't have any snapshots

After you have created a snapshot, it will be listed here

Create snapshot



10.删除PostgreSQL应用程序中的数据库

重新登录到PostgreSQL、列出可用数据库、删除先前创建的数据库并重新列出、以确保数据库已被删除。

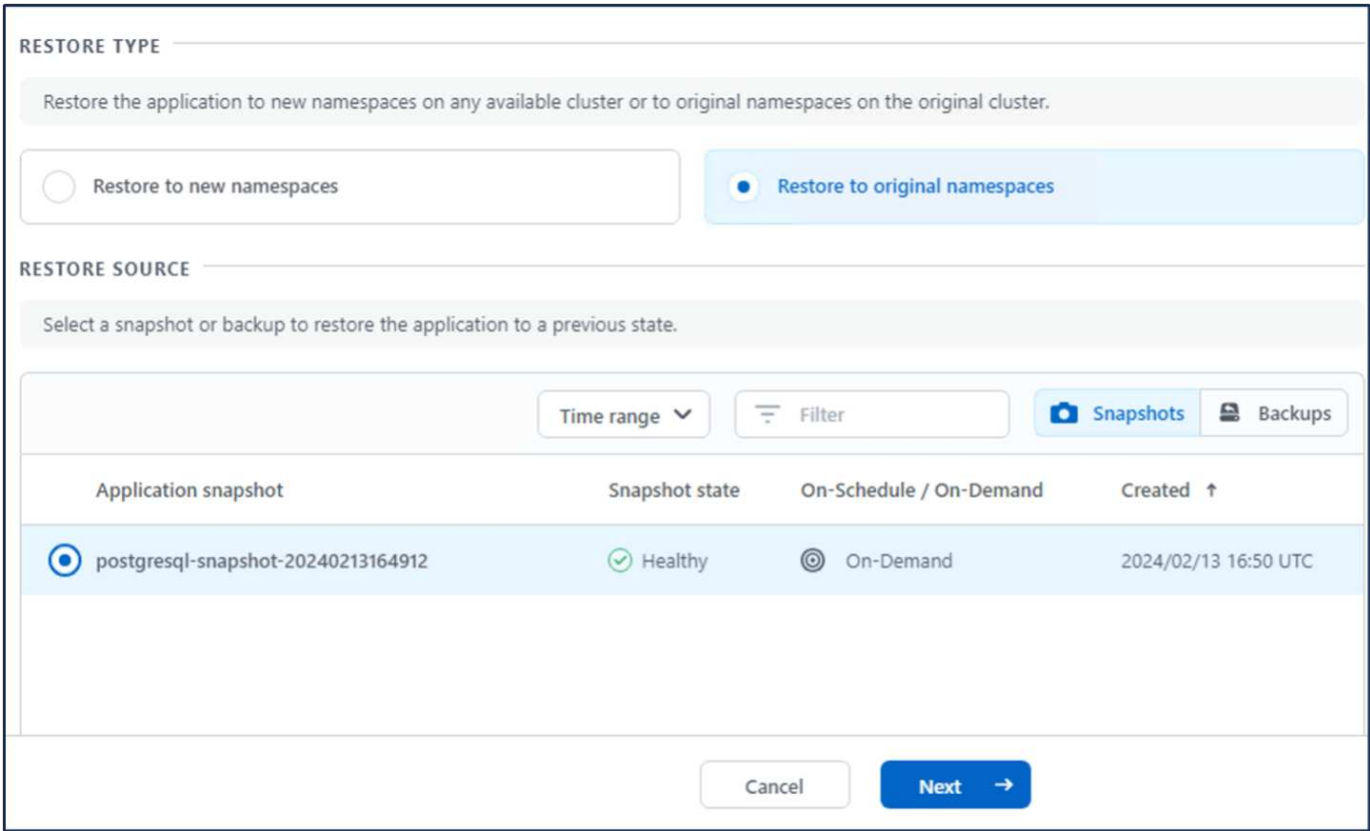
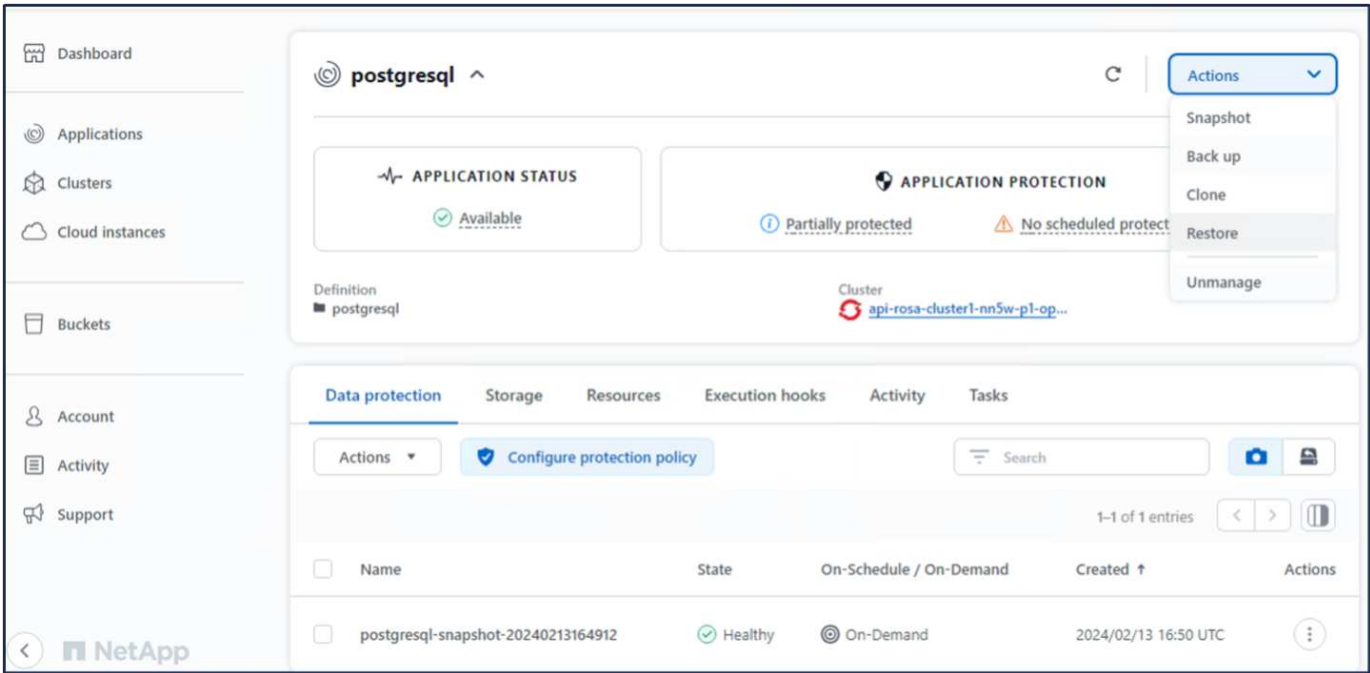
```
postgres=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | ICU Locale | ICU Rules | Access priv
-----+-----+-----+-----+-----+-----+-----+-----+-----
erp    | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
postgres | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
template0 | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
template1 | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
(4 rows)

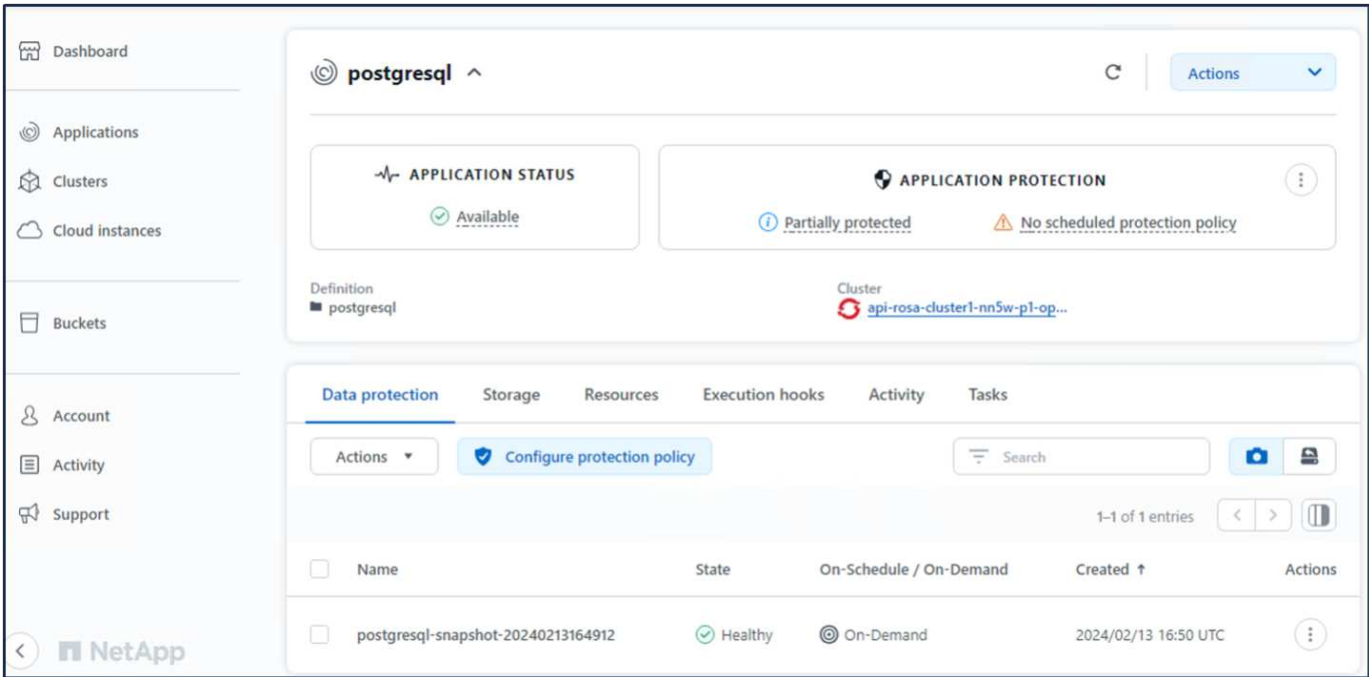
postgres=# DROP DATABASE erp;
DROP DATABASE
postgres=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | ICU Locale | ICU Rules | Access priv
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
template0 | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
template1 | postgres | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | | =c/postgres
(3 rows)
```

11.使用ACs从快照恢复

要从快照还原应用程序、请转到ACS UI登录页面、选择应用程序、然后选择还原。您需要选择要从中还原的快照或备份。(通常、您会根据所配置的策略创建多个)。在接下来的几个屏幕中做出适当的选择，然后单

击*Restore*。从快照还原后、应用程序状态将从还原变为可用。





12.验证您的应用程序是否已从快照中恢复

登录到PostgreSQL客户端、您现在应该可以看到表以及以前的表中的记录。就是这样。只需单击一个按钮、您的应用程序便已恢复到先前的状态。这就是我们使用Astra Control为客户实现的简单体验。

```
[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:vl.24": allowPrivilegeEscalation != false (container "postgresql-client" must set securityContext.allowPrivilegeEscalation=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 securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost")
If you don't see a command prompt, try pressing enter.

postgresql=# \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 |              |              |
template1 | postgres | UTF8     | libc             | en_US.UTF-8 | en_US.UTF-8 |              |              |
=c/postgres +
postgres=Ctc/postgres
=c/postgres +
postgres=Ctc/postgres
(4 rows)

postgresql=# \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)
```

数据迁移

此页面显示了使用FSx for NetApp ONTAP 作为永久性存储的托管Red Hat OpenShift集群上容器工作负载的数据迁移选项。

数据迁移

AWS上的Red Hat OpenShift服务以及适用于NetApp ONTAP 的FSx (FSxN)是AWS服务产品组合的一部分。FSxN可用于单AZ或多AZ选项。Multi-Az选项可防止数据受到可用性区域故障的影响。FSxN可以与Astra Trident集成、为ROSA集群上的应用程序提供永久性存储。

使用Helm将FSxN与TRident集成图表

Rosa集群与Amazon FSx for ONTAP集成

容器应用程序的迁移涉及：

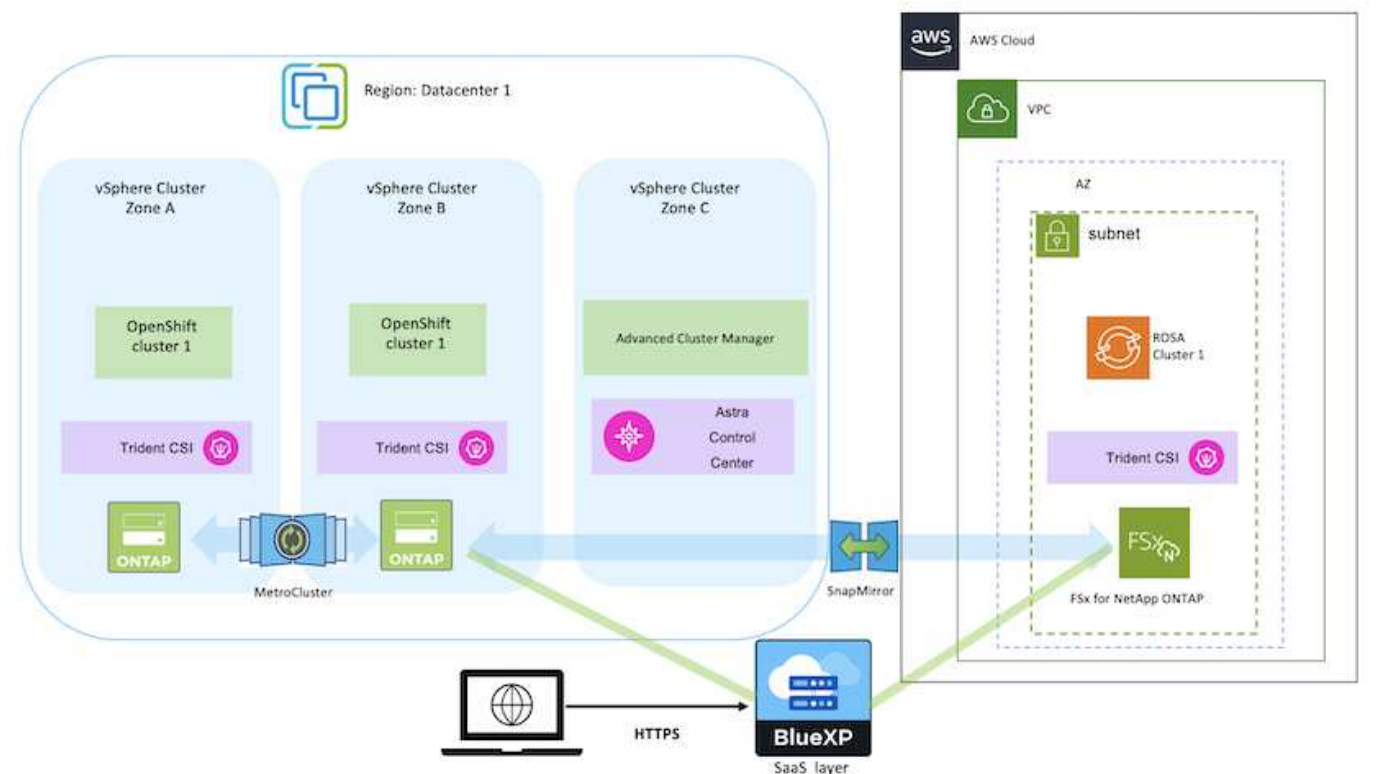
- 永久性卷：可使用BlueXP来实现。另一种选择是使用Astra Control Center处理从内部环境到云环境的容器应用程序迁移。自动化也可以用于相同目的。
- 应用程序元数据：可以使用OpenShift GitOps (Argo CD)来实现。

使用FSxN对ROSA集群上的应用程序进行故障转移和故障恢复、以实现永久性存储

以下视频演示了使用BlueXP和Argo CD的应用程序故障转移和故障恢复场景。

对ROSA集群上的应用程序进行故障转移和故障恢复

适用于OpenShift容器工作负载的数据保护和迁移解决方案



版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。