

使用FSxN在AWS上运行Red Hat OpenShift服务 NetApp Solutions

NetApp December 19, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/netapp-solutions/containers/rh-osn_use_case_rosa_solution_overview.html on December 19, 2024. Always check docs.netapp.com for the latest.

目录

使用FSxN在AWS上运行Red Hat OpenShift服务	. 1
借助NetApp ONTAP在AWS上运行Red Hat OpenShift服务	. 1
借助NetApp ONTAP在AWS上运行Red Hat OpenShift服务	11

使用FSxN在AWS上运行Red Hat OpenShift服务

借助NetApp ONTAP在AWS上运行Red Hat OpenShift服务

概述

在本节中、我们将介绍如何将FSx for ONTAP用作ROSA上运行的应用程序的永久性存储层。其中将显示 在ROSA集群上安装NetApp Trident CSI驱动程序、配置FSx for ONTAP文件系统以及部署有状态应用程序示例 的过程。同时、还会显示备份和还原应用程序数据的策略。借助这款集成解决方案、您可以建立一个共享存储框 架、轻松地跨多个应用程序进行扩展、从而简化使用Trident CSI驱动程序扩展、保护和还原数据的过程。

前提条件

- "AWS 帐户"
- "Red Hat帐户"
- IAM用户"具有适当的权限"、用于创建和访问ROSA集群
- "AWS命令行界面"
- "罗莎命令行界面"
- "OpenShift命令行界面"(OC)
- Helm 3"文档。"
- "HCP ROSA集群"
- "访问Red Hat OpenShift Web控制台"

此图显示了部署在多个澳大利亚地区的ROSA集群。罗莎集群的主节点、基础架构节点位于Red Hat的VPC中, 而工作节点位于客户帐户的VPC中。我们将在同一个VPC中创建一个FSx for ONTAP文件系统、并在ROSA集群 中安装Trident驱动程序、从而允许此VPC的所有子网连接到文件系统。

Availability Zone 1	Availability Zone 2	Availability Zone 2	OpenShift control plan	e (API server, etcd, contr	olier, scheduler) manage
Private subnet	Private subnet	Private subnet	M5 instance	M5 instance	M5 instance
Fa Instances	OpenShift Worker nodes. (router)	Instances			

初始设置

1.为FSX配置NetApp ONTAP

在与ROSA集群相同的VPC中创建适用于NetApp ONTAP的多可用性FSx。可通过多种方法实现此操作。提供了使用CloudFormation堆栈创建FSxN的详细信息

a.Clone the GitHub re듦 퓚 处

\$ git clone https://github.com/aws-samples/rosa-fsx-netapp-ontap.git

b.Run the CloudFormation Stack 通过将参数值替换为您自己的值来运行以下命令:

\$ cd rosa-fsx-netapp-ontap/fsx

```
$ aws cloudformation create-stack \
  --stack-name ROSA-FSXONTAP \
 --template-body file://./FSxONTAP.yaml \
  --region <region-name> \
  --parameters \setminus
  ParameterKey=Subnet1ID, ParameterValue=[subnet1 ID] \
  ParameterKey=Subnet2ID, ParameterValue=[subnet2 ID] \
  ParameterKey=myVpc,ParameterValue=[VPC ID] \
ParameterKey=FSxONTAPRouteTable, ParameterValue=[routetable1 ID, routetable2
ID] \
  ParameterKey=FileSystemName, ParameterValue=ROSA-myFSxONTAP \
  ParameterKey=ThroughputCapacity,ParameterValue=1024 \
  ParameterKey=FSxAllowedCIDR,ParameterValue=[your allowed CIDR] \
  ParameterKey=FsxAdminPassword, ParameterValue=[Define Admin password] \
  ParameterKey=SvmAdminPassword, ParameterValue=[Define SVM password] \
  --capabilities CAPABILITY NAMED IAM
```

其中: region-name: sas the region where the ROSA cluster is才会部署subnet1_ID: FSxN subnet2_ID的首选 子网的ID: FSxN的备用子网的ID VPC_ID: 部署ROSA集群的ONTAP的VPC的ID routetable1_ID 、routetable2_ID: 与所选CIDR规则关联的路由表的ID: 允许对所选Sx访问的子网进行控制。您可以使 用0.0.0.0/0或任何适当的CIDR允许所有流量访问FSx for ONTAP的特定端口。定义管理员密码: 用于登录 到FSxN的密码定义SVM密码: 用于登录到要创建的SVM的密码。

验证是否已使用Amazon FSx控制台创建文件系统和Storage Virtual Machine (SVM)、如下所示:

Amazon FSx X	FSx > File systems > fs-03a16050b	eae7ca24		
File systems Volumes	OntapFileSystem_			Attach Actions V
File Caches Backups	▼ Summary			
ONTAP Storage virtual machines	File system ID	SSD storage capacity 1024 GiB	Update	Availability Zones us-east-2a (Preferred) 🗇 us-east-2b (Standby) 🗇
Open2F5 Snapshots	Oriecycle state O Available File system type	Throughput capacity 1024 MB/s Provisioned IOPS	Update	Creation time 2024-10-09T11:29:33-04:00
FSx on Service Quotas 🖸 Settings	ONTAP Deployment type Multi-AZ 1	3072 Number of HA pairs 1		

2.为ROSA群集安装和配置Trident CSI驱动程序

a.Add the Trident Helm re듦 在处

\$ helm repo add netapp-trident https://netapp.github.io/trident-helm-chart

b.Install Trident using Helm

```
$ helm install trident netapp-trident/trident-operator --version
100.2406.0 --create-namespace --namespace trident
```



根据您安装的版本、需要在显示的命令中更改version参数。有关正确的版本号、请参见"文档。 "。有关安装Trident的其他方法,请参阅《Trident》"文档。"。

c.Verify that all Trident Pod are in the running state

[root@localhost hcp-testing]#				
[root@localhost hcp-testing]# oc ge	et pods	-n trident		
NAME	READY	STATUS	RESTARTS	AGE
trident-controller-f5f6796f-vd2sk	6/6	Running	0	19h
trident-node-linux-4svgz	2/2	Running	0	19h
trident-node-linux-dj9j4	2/2	Running	0	19h
trident-node-linux-jlshh	2/2	Running	0	19h
trident-node-linux-sqthw	2/2	Running	0	19h
trident-node-linux-ttj9c	2/2	Running	0	19h
trident-node-linux-vmjr5	2/2	Running	0	19h
trident-node-linux-wvqsf	2/2	Running	0	19h
trident-operator-545869857c-kgc7p [root@localhost hcp-testing]# _	1/1	Running	0	19h

3.配置Trident CSI后端以使用FSx for ONTAP (ONTAP NAS)

Trident后端配置指示Trident如何与存储系统通信(在本示例中为FSx for ONTAP)。要创建后端、我们将提供要连接到的Storage Virtual Machine的凭据、以及集群管理和NFS数据接口。我们将使用"ontap-NAS 驱动程序" 在FSx文件系统中配置存储卷。

**...首先、使用以下YAML"为SVM凭据创建一个密钥

apiVersion: v1 kind: Secret metadata: name: backend-fsx-ontap-nas-secret namespace: trident type: Opaque stringData: username: vsadmin password: <value provided for Define SVM password as a parameter to the Cloud Formation Stack>

您也可以从AWS机密管理器检索为FSxN创建的SVM密码、如下所示。

Secrets		C Store a new secret
Q. Filter secrets by nome, description, tog key, tog value, o	uning service or primary Region	< 1 > @
	Description	Last retrieved (UTC)
Secret name	peacificite	
Secret name HCP-ROSA-FSXDNTAP-SVMAdminPassword	5VMAdminPassword	October 9, 2024

WS Secrets Manager > Secrets > HCP-ROSA-FSXONTAP-SVMAdminPassword		
Secret details		C Actions ¥
Encryption key Encryption key aws/secretimanager Secret name HCP-ROSA-FSXONTAP-SVMAdminPassword Secret AIRN amiawssecretismanager:us-east-2:316088182667:secret.HCP-ROSA-FSXONTAP-SVMAdminPassword- ABiUaf	Secret description	
Overview Rotation Versions Replication Tags Secret value into		Retrieve secret value

b.Next,使用以下命令将SVM凭据的密钥添加到ROSA集群中

(i)

\$ oc apply -f svm_secret.yaml

您可以使用以下命令验证是否已在Trident命名空间中添加此密钥

\$ oc get secrets -n trident |grep backend-fsx-ontap-nas-secret

```
[root@localhost hcp-testing]#
[root@localhost hcp-testing]# oc get secrets -n trident | grep backend-fsx-ontap-nas-secret
backend-fsx-ontap-nas-secret Opaque 2 21h
[root@localhost hcp-testing]# _
```

 C.接下来,创建后端对象为此,移至克隆的Git储存库的FSX目录。打开文件backend-ams-naS.yaml ONTAP。将以下内容: ManagementLIF替换为管理DNS名称 dataLIF,替换为Amazon FSx SVM的NFS DNS名称,并将 SVM**替换为SVM名称。使用以下命令创建后端对象。

使用以下命令创建后端对象。

\$ oc apply -f backend-ontap-nas.yaml



您可以从Amazon FSx控制台获取管理DNS名称、NFS DNS名称和SVM名称、如以下屏幕截图所示

Amazon FSx	×	Summary		
File systems Volumes		SVM ID Creation time svm-07a733da2584f2045 👩 2024-10-09T11:51:46-04	:00	Active Directory
File Caches Backups	40	SVM name Lifecycle state SVM1 🗗 💿 Created		
 ONTAP Storage virtual machines 		UUID Subtype a845e7bf-8653-11ef-8/27-0f43b1500927 DEFAULT		
 OpenZFS Snapshots 		File system ID fs=03a16050beae7ca24		
F5x on Service Quotas 🗹 Settings		Resource ARN armaws:fsxcus-east-2:316088182667:storage-virtual- machine/fs-03a16050beae7ca24/svm- 07a733da2584f2045		
		Endpoints Administration Volumes Tags		
		Endpoints		
		Management DNS name svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonavs.com 🗗	Management IP address 198.19.255.182 🗇	
		NFS DNS name svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonavs.com	NFS IP address 198.19.255.182	
		ISCSI DNS name iscsi.svm-07a733da2584f2045.fs-03a16050beae7ca24.fsx.us-east-2.amazonaws.com	iSCSI IP addresses 10.10.9.32, 10.10.26.28	

*现在,运行以下命令以验证是否已创建后端对象,并且Phase (阶段)显示bound and Status (绑定)为Success (成功)



4.创建存储类配置Trident后端后,您可以创建一个Kubbernetes存储类以使用后端。存储类是可供集群使用的资源对象。它介绍并分类您可以为应用程序请求的存储类型。

...查看FSx文件夹中的storage-class-CSI-NAS. yaml文件。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: trident-csi
provisioner: csi.trident.netapp.io
parameters:
   backendType: "ontap-nas"
   fsType: "ext4"
allowVolumeExpansion: True
reclaimPolicy: Retain
```

• b.在ROSA集群中创建存储类、并验证是否Trident已创建ROSA-CSI存储类。**

[root@localhost hc [root@localhost hc	p-testing]# p-testing]#		(c		
[root@localhost hc	o-testing]# oc apply -f :	storage-class-cs	i-nas.yaml		
storageclass.stora	ge.k8s.io/trident-csi cro	eated			
[root@localhost hc	p-testing]# oc get sc				
NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
gp2-csi	ebs.csi.aws.com	Delete	WaitForFirstConsumer	true	2d16h
gp3-csi (default)	ebs.csi.aws.com	Delete	WaitForFirstConsumer	true	2d16h
trident-csi	csi.trident.netapp.io	Retain	Immediate	true	4s
[root@localhost hc	p-testing]#				

至此、Trident CSI驱动程序的安装完成、并完成了它与FSx for ONTAP文件系统的连接。现在、您可以使用FSx for ONTAP上的文件卷在ROSA上部署示例PostgreSQL有状态应用程序。

• C.确认没有使用PVC-sl存储类创建Trident和PV。**

<pre>froot@localhost hcp-testing]# [root@localhost hcp-testing]# [root@localhost hcp-testing]# openshift wonitering openshift virtualization-os-immges openshift virtualization-os-immges openshift virtualization-os-immges openshift virtualization-os-immges</pre>	pvc -A NAME prometheus-data centos-streami- centos-streami- fedoro-21a0f3e6 rhe18-06326fdeb rhe18-06321bd136 rm	- prometheus - kds - i - prometheus - kds - i oae111cdd5a1 382d 382d 38cd 259 e64	STATUS V e Bound p Bound p Bound p Bound p Bound p Bound p Bound p	OLUME NC - 944553a5 NC - 7d949aef NC - 82b0e84a NC - 82b0e84a NC - 64375ad NC - 2d06de48 NE - 74374ce7	-07x5-440a-8x50-554384c97528 +8663-659a-854-514455575bab2 -cb37-4409-8574-55495496c10 -c557-4425-1079-348447415521 -d377-4563-8340-3650+3134679c -5936-431e-053-925876350e4134679c	CAPACITY 100G1 100G1 30G1 30G1 30G1 30G1 30G1	ACCESS MODES Rido Rido Rido Rido Rido Rido Rido Rido	STORAGECLAS gp3-csi gp3-csi gp3-csi gp3-csi gp3-csi gp3-csi gp3-csi	S VOLUMEAT Curiset> Curiset> Curiset> Curiset> Curiset> Curiset>	TRIBUTESCLASS	AGE 2d16h 2d16h 2dh 44b 44h 44h 44h
Proceduce almost ncp-testing a oc get NAME pvc-2dc6de48-5916-411e-9cb3-99598f50 pvc-6df375ad-d377-456d-83a0-368e413a pvc-7d949aef-e00d-4d9a-8b54-514e385f1	CAPACITY beAC 38GI 579c 38G1 sab2 100G1	ACCESS MODES RUO RUO RUO	RECLAIM POLIC Delete Delete Delete	Y STATUS Bound Bound Bound	CLAIM openshift-virtualization-os openshift-virtualization-os openshift-monitoring/promet)	-images/rhe -images/fed heus-data-p	18-052df0eb25 ora-2146f3e628 rometheus-k8s-	9 cd 1	TORAGECLASS p3-CS1 p3-CS1 p3-CS1 p3-CS1	VOLUMEATTREE cunset> cunset> cunset>	NUTESCLASS
pvc-82b8c84a-e5ef-452b-bf98-1eae4fe1 pvc-9a4553a5-07e9-440a-8a90-99e384c9	12c1 3861 7624 10861	RuD RuO	Delete Delete	Bound	openshift-virtualization-os openshift-monitoring/prometH	-images/cen heus-data-p	tos-stream9-d8 rometheus-k8s-l	2f4a141a4 0	p3-cs1 p3-cs1	cunset> cunset>	
pvc-deb61444-cb3f-449b-807d-39d02849 pvc-f4374ce7-568d-4afc-b635-0328cf45 [root@localhost-hcp-testing]# _	5c16 30G1 1434 30G1	RLC RLC	Delete Delete	Bound	openshift-virtualization-os- openshift-virtualization-os-	-inages/cen -inages/rhe	tos-stream9-ba 19-2521bd116e6	e111cdd5a	p3-cs1 p3-cs1	conset> conset>	

*验证应用程序是否可以使用Trident Csi.创建PV

使用FSX文件夹中提供的PVC-AML.YAML文件创建Trident。

```
pvc-trident.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
   name: basic
spec:
   accessModes:
    - ReadWriteMany
   resources:
      requests:
       storage: 10Gi
   storageClassName: trident-csi
```

You can issue the following commands to create a pvc and verify that it has been created. image:redhat openshift container rosa image11.png["使用Trident创建测试PVC"]

5.部署示例PostgreSQL有状态应用程序

…使用HeIm安装PostgreSQL

```
$ helm install postgresql bitnami/postgresql -n postgresql --create
-namespace
```

[root@localhost hcp-testing]# helm install postgresql bitnami/postgresql -n postgresqlcreate-namespace NAME: postgresql LAST DEPLOYED: Mon Oct 14 06:52:58 2024
NAMESPACE: postgresq1
STATUS: deployed Bevrstow - 1
TEST SUITE: None
NOTES:
CHART NAME: postgresql
CHART VERSION: 15.5.21
40-9 VERSION: 16.4.0
** 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=\$(kubect1 get secretnamespace postgresq1 postgresq1 -o jsonpath="(.data.postgres-password)" base64 -d)
To connect to your database run the following command:
kubectl run postgresql-clientrmtty -irestart='Never'namespace postgresqlimage docker.io/bitnami/postgresql:16.4.0-debian-12-r0 command psqlhost 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 1001) does not exist"
To connect to your database from outside the cluster execute the following commands:
kubectl port-forwardnamespace postgresql svc/postgresql 5432:5432 & PGPASSWORD="\$POSTGRES_PASSWORD" psqlhost 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. sword, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.

• b.确认应用程序POD正在运行,并且为应用程序创建了PVC和PV。**

[root@localhos	t hcp-te	esting]# oc	get pods	-n postgresql	
NAME	READY	STATUS	RESTARTS	AGE	
postgresql-0	1/1	Running	0	29m	

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
data-postgresql-0	Bound	pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8db6	8Gi	RWO	trident-csi

<pre>[root@localhost hcp-testing]# oc get p</pre>	v grep post	gresql			
pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8d	66 8Gi	RWO	Retain	Bound	postgresql/data-postgresql-0
csi <unset></unset>	4h20m				
<pre>[root@localhost hcp-testing]# _</pre>					

• C.部署PostgreSQL客户机**

使用以下命令获取已安装的PostgreSQL服务器的口令。

```
$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql
postgresql -o jsoata.postgres-password}" | base64 -d)
```

使用以下命令运行PostgreSQL客户机,并使用口令连接到服务器

<pre>\$ kubectl run postgresql-clientrmtty -irestart='Never'</pre>
namespace postgresqlimage docker.io/bitnami/postgresql:16.2.0-debian-
11-r1env="PGPASSWORD=\$POSTGRES_PASSWORD" \
>command psqlhost postgresql -U postgres -d postgres -p 5432



*创建数据库和表。创建表的纲要并将2行数据插入表中。



借助NetApp ONTAP在AWS上运行Red Hat OpenShift服务

本文档将概述如何将NetApp ONTAP与基于AWS的Red Hat OpenShift服务(ROSA)结合使用。

创建卷快照

1.创建应用程序卷的快照在本节中,我们将介绍如何创建与应用程序关联的卷的Trident快照。这将是应用程序数 据的时间点副本。如果应用程序数据丢失、我们可以从此时间点副本恢复数据。注意:此快照与ONTAP中的原 始卷存储在同一聚合中(内部或云中)。因此、如果ONTAP存储聚合丢失、我们将无法从其快照中恢复应用程序数 据。

**...创建卷快照类将以下清单保存在名为volume-snapshot-class.yaml的文件中

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

使用上述清单创建快照。

[root@localhost hcp-testing]# oc create -f volume-snapshot-class.yaml
volumesnapshotclass.snapshot.storage.k8s.io/fsx-snapclass created
[root@localhost hcp-testing]# _

b.接下来,创建快照**通过创建卷快照创建现有PVC的快照,以创建PostgreSQL数据的时间点副本。这将创建FSx快照、该快照在文件系统后端几乎不占用任何空间。将以下清单保存在名为volume-Snapshot.yaml的文件中:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: postgresql-volume-snap-01
spec:
  volumeSnapshotClassName: fsx-snapclass
  source:
    persistentVolumeClaimName: data-postgresql-0
```

• C.创建卷快照并确认已创建**

删除数据库以模拟数据丢失(由于各种原因可能会发生数据丢失、此处我们只是通过删除数据库来模拟它)



*删除数据库以模拟数据丢失(由于各种原因可能会发生数据丢失,此处我们只是通过删除数据库来模拟它)





从卷快照还原

1.从Snap照恢复在本节中,我们将介绍如何从应用程序卷的Trident快照恢复应用程序。

...从快照创建卷克隆

要将卷还原到其先前状态、您必须根据创建的快照中的数据创建一个新的PVC。为此、请将以下清单保存在名为pvC-CLONE .yaml的文件中

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: postgresql-volume-clone
spec:
accessModes:
    - ReadWriteOnce
storageClassName: trident-csi
resources:
    requests:
    storage: 8Gi
dataSource:
name: postgresql-volume-snap-01
kind: VolumeSnapshot
apiGroup: snapshot.storage.k8s.io
```

通过使用快照作为源并使用上述清单创建PVC来创建卷的克隆。应用清单并确保已创建克隆。

[root@localhost hcp-test persistentvolumeclaim/po [root@localhost hcp-test	ing]# oc stgresql- ing]# oc	create -f postgresql-pvc-clone.yaml -n post volume-clone created get pvc -n postgresql	gresql		
NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
data-postgresql-0	Bound	pvc-e3ddd9bd-e6a7-4a4a-b935-f1c090fd8db6	8Gi	RWO	trident-csi
postgresql-volume-clone [root@localhost hcp-test:	Bound ing]# _	pvc-b38fbc54-55dc-47e8-934d-47f181fddac6	8Gi	RWO	trident-csi

• b.删除初始PostgreSQL安装**

```
[root@localhost hcp-testing]#
[root@localhost hcp-testing]# helm uninstall postgresql -n postgresql
release "postgresql" uninstalled
[root@localhost hcp-testing]# oc get pods -n postgresql
No resources found in postgresql namespace.
[root@localhost hcp-testing]# _
```

• C.使用新的克隆PVC**创建新的PostgreSQL应用程序

```
$ helm install postgresql bitnami/postgresql --set
primary.persistence.enabled=true --set
primary.persistence.existingClaim=postgresql-volume-clone -n postgresql
```

[root@localhost hcp-testing]# [root@localhost hcp-testing]# helm install postgresql bitnami/postgresqlset primary.persistence.enabled=true \ >set primary.persistence.existingClaim=postgresql-volume-clone -n postgresql NAME: postgresql
LAST DEPLOYED: Mon Oct 14 12:03:31 2024 NAMESPACE: postgresql
STATUS: deployed REVISION: 1 TEST SUTTE: None
NOTES: CHART NAME: postgresql
CHART VERSION: 15.5.21 APP VERSION: 16.4.0
** 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 secretnamespace postgresql postgresql -o jsonpath="{.data.postgres-password}" bas
To connect to your database run the following command:
<pre>kubectl run postgresql-clientrmtty -irestart='Never'namespace postgresqlimage docker.io/bitnami/postgresql:16 command psqlhost postgresql -U postgres -d postgres -p 5432</pre>
> NOTE: If you access the container using bash, make sure that you execute "/opt/bitnami/scripts/postgresql/entrypoint.sh /b 1001} does not exist"
To connect to your database from outside the cluster execute the following commands:
kubectl port-forwardnamespace postgresql svc/postgresql 5432:5432 & PGPASSWORD="\$POSTGRES_PASSWORD" psqlhost 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 throug sword, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.
WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For proving to your workload needs: - primary resources
- readReplicas.resources
[root@localhost hcp-testing]# _

*验证应用程序POD是否处于running状态

[root@localhos	st hcp-te	esting]# oc	get pods	-n postgresql
NAME	READY	STATUS	RESTARTS	AGE
oostgresql-0	1/1	Running	0	2m1s
[root@localhos	st hcp-te	esting]#		

验证POD是否使用克隆作为其PVC

root@localhost hcp-testing]# root@localhost hcp-testing]# oc describe pod/postgresql-0 -n postgresql_

Contai	nersRea	dy Tru	ie			
PodSch	eduled	Tru	ie			
Volumes:						
empty-	dir:					
Type Medi	: um:	EmptyDir (a tempo	orary di	irectory that shares a poo	i's lifetime)	
Size	Limit:	<unset></unset>				
dshm:						
Type Medi	: um:	EmptyDir (a tempo Memory	orary di	irectory that shares a poo	i's lifetime)	
- Size	Limit:	<pre><unset></unset></pre>				
data:						
Type	: mName:	PersistentVolume	Claim (a reference to a Persiste	entVolumeClaim in the same namespace)	
Read	Only:	false	ic croite			
DoS Clas	5:	Burstable				
Node-Sel	ectors:	<none></none>				
Tolerati	ons:	node.kubernetes. node.kubernetes. node.kubernetes.	io/memo io/not- io/unre	ory-pressure:NoSchedule op ready:NoExecute op=Exists eachable:NoExecute op=Exist	D=Exists 5 for 300s sts for 300s	
Events:						
Туре	Reaso	n	Age	From	Message	
Normal	Sched	uled	3m55s	default-scheduler	Successfully assigned postgresql/postgr	es
.us-east	-2.comp	ute.internal				
Normal	Succe	ssfulAttachVolume	3m54s	attachdetach-controller	AttachVolume.Attach succeeded for volum	e
8-934d-4	7f181fd	dac6"				
Normal	Added.	Interface	3m43s	multus	Add eth0 [10.129.2.126/23] from ovn-kub	er
Normal	Pulle	d	3m43s	kubelet	Container image "docker.io/bitnami/post	gr
r0" alre	ady pre	sent on machine				
Normal	Creat	ed	3m42s	kubelet	Created container postgresql Activ	at
Normal	Start	ed	3m42s	kubelet	Started container postgresql Go to S	ret
root@lo	calhost	hcp-testing]#				

f)要验证数据库是否已按预期还原、请返回容器控制台并显示现有数据库

[root@localhost hcp-testing]# kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql: \$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.allowPri capabilities (container "postgresql-client" must set securityContext.capabilities.drop=["ALL"]), runAsNonRoot != true (pod or container "postgresql-client" must set securityContext.allowPri Root=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=#	νı.			List of da	tabases			
Name	Owner	Encoding	Locale Provider	Collate	Ctype	ICU Locale	ICU Rules	Access privileges
erp postgres template0	postgres postgres postgres	UTF8 UTF8 UTF8	libc libc libc	en_US.UTF-8 en_US.UTF-8 en_US.UTF-8	en_US.UTF-8 en_US.UTF-8 en_US.UTF-8			=c/postgres
template1	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres postgres=CTc/postgres
erp=# \dt L Schema	ist of rela Name Ty	tions pe Owner						
You are now erp=# \dt	connected	to database	"erp" as user "po	stgres".				
public p (1 row)	ersons ta	ble postgr	res					
erp=# SELEC	T * FROM PE	RSONS; name						
id first	name 1050							

演示视频

Amazon FSx for NetApp ONTAP使用托管控制平台在AWS上运行Red Hat OpenShift服务

有关Red Hat OpenShift和OpenShift解决方案的更多视频,请参见"此处"。

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。