1.环境信息

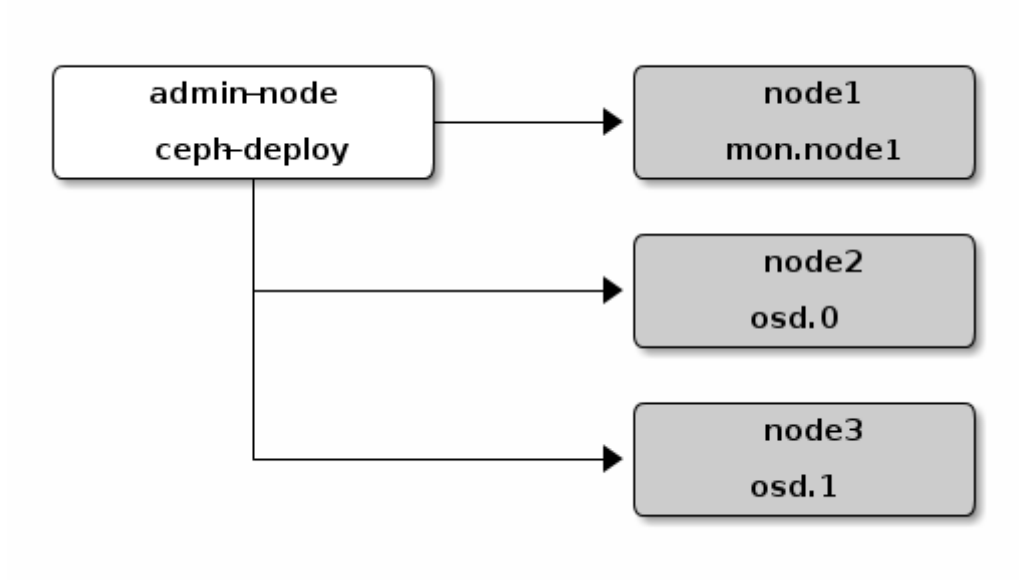
ceph rbd节点

172.16.103.243

172.16.103.247

172.16.103.248

用户名/密码：orcadt/p@ssword



节点列表

节点名称	节点IP	当前版本
node-243	172.16.103.243	1.1.4-2
node-247	172.16.103.247	1.1.4-2
node-248	172.16.103.248	1.1.4-2

2.PV和PVC

Kubernetes PersistentVolumes 持久化存储方案中，提供两种 API 资源方式: PersistentVolume(简称PV) 和 PersistentVolumeClaim(简称PVC)。PV 可理解为集群资源，PVC 可理解为对集群资源的请求，Kubernetes 支持很多种持久化卷存储类型。Ceph 是一个开源的分布式存储系统，支持对象存储、块设备、文件系统，具有可靠性高、管理方便、伸缩性强等特点。在日常工作中，我们会遇到使用 k8s 时后端存储需要持久化，这样不管 Pod 调度到哪个节点，都能挂载同一个卷，从而很容易读取或存储持久化数据，我们可以使用 Kubernetes 结合 Ceph 完成。

3.在Ceph上为Kubernetes创建一个存储池

```
ceph osd pool create rbd 128
```

rados lspools
[orcadt@node-243 ~]$ rados lspools
rbd
testpool
test

4.创建存储镜像

`rbd create bwiot --size 1024`

rbd create --size 1024 mysql_slave --image-feature layering
rbd create --size 1024 mysql_master --image-feature layering
rbd create --size 1024 mongo --image-feature layering

rbd create --size 1024 serverlog --image-feature layering
rbd create --size 1024 brokerstore --image-feature layering
rbd create --size 1024 brokerlog --image-feature layering

[orcadt@node-243 ~]$ rbd ls
brokerlog
brokerstore
bwiot
mongo
mysql-single
mysql_master
mysql_slave
serverlog
test

持久化存储卷

名称	总量	访问模式	回收策略	状态	声明	存储类	原因	已创建
pv-mysql-single	1Gi	ReadWriteOnce	Retain	Bound	bwiot/pv-mysql-sing	-	-	8 天
brokerstore-pv	1Gi	ReadWriteOnce	Retain	Bound	bwiot/brokerstore-p	-	-	8 天
serverlog-pv	1Gi	ReadWriteOnce	Retain	Bound	bwiot/serverlog-pv	-	-	8 天
brokerlog-pv	1Gi	ReadWriteOnce	Retain	Bound	bwiot/brokerlog-pv	-	-	8 天
pv-mysql-slave	1Gi	ReadWriteOnce	Retain	Bound	bwiot/pv-mysql-slav	-	-	8 天
pv-mysql-master	1Gi	ReadWriteOnce	Retain	Bound	bwiot/pv-mysql-mas	-	-	8 天
mongo-pv	1Gi	ReadWriteOnce	Retain	Bound	bwiot/mongo-pvc	-	-	8 天
ceph-rbd-pv	1Gi	ReadWriteOnce	Retain	Bound	default/ceph-rbd-pv	-	-	8 天

持久化存储卷声明

名称	状态	存储卷	总量	访问模式	存储类	已创建
pv-mysql-single	Bound	pv-mysql-single	1Gi	ReadWriteOnce	-	8 天
serverlog-pv	Bound	serverlog-pv	1Gi	ReadWriteOnce	-	8 天
brokerstore-pv	Bound	brokerstore-pv	1Gi	ReadWriteOnce	-	8 天
brokerlog-pv	Bound	brokerlog-pv	1Gi	ReadWriteOnce	-	8 天
pv-mysql-slave	Bound	pv-mysql-slave	1Gi	ReadWriteOnce	-	8 天
pv-mysql-master	Bound	pv-mysql-master	1Gi	ReadWriteOnce	-	8 天
mongo-pvc	Bound	mongo-pv	1Gi	ReadWriteOnce	-	8 天

mongo-pv

详情

名称: mongo-pv

注释: pv.kubernetes.io/bound-by-controller: yes

创建时间: 2019-02-14T01:49 UTC

状态: Bound

声明: bwiot/mongo-pvc

回收策略: Retain

访问模式: ReadWriteOnce

存储类: -

原因: -

消息: -

来源

RBD

监视器: 172.16.103.243:6789

镜像: mongo

用户: admin

密钥环: /etc/ceph/keyring

SecretRef: ceph-secret

只读: -

总量

资源名称	数量
Storage	1Gi

5.将k8s用户的key进行base64编码

[orcadt@node-243 ceph]$ sudo cat ceph.client.admin.keyring
[client.admin]
    key = AQCZ005chQnXBBAAzBJaXqyVP6J8AO6iocJyuQ==
    auid = 0
    caps mds = "allow"
    caps mon = "allow *"
    caps osd = "allow *"

echo AQCZ005chQnXBBAAzBJaXqyVP6J8AO6iocJyuQ==|base64
QVFDWjAwNWNoUW5YQkJBQXpCSmFYcXlWUDZKOEFPNmlvY0p5dVE9PQo=

6.在Kubernetes创建访问Ceph的Secret

```
[root@k8s-master2 mongo-rbd]# cat ceph-secret.yml
apiVersion: v1
kind: Secret
metadata:
  name: ceph-secret
  namespace: bwiot
type: "kubernetes.io/rbd"
data:
  key:
QVFDWjAwNWNoUW5YQkJBQXpCSmFYcXlWUDZKOEFPNmlvY0p5dVE9PQo=
```

7.创建一个PersistentVolume

```
[root@k8s-master2 mongo-rbd]# cat mongo-pv.yml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongo-pv
  namespace: bwiot
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  rbd:
    monitors:
      - 172.16.103.243:6789
    pool: rbd
```

```
    image: mongo
    user: admin
    secretRef:
      name: ceph-secret
    fsType: ext4
    readOnly: false
```

8.创建一个PersistentVolumeClaim

```
[root@k8s-master2 mongo-rbd]# cat mongo-pvc.yml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongo-pvc
  namespace: bwiot
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

9.创建rc挂载PVC

```
[root@k8s-master2 mongo-rbd]# cat mongo-rc.yml
apiVersion: v1
kind: ReplicationController
metadata:
  name: mongo
  namespace: bwiot
  labels:
    name: mongo
spec:
  replicas: 1
  selector:
    name: mongo
  template:
    metadata:
```

```
    labels:
      name: mongo
  spec:
    nodeName: k8s-node1
    containers:
    - name: mongo
      image: 172.16.103.246:5000/mongo
      imagePullPolicy: IfNotPresent
      ports:
        - containerPort: 27017
      volumeMounts:
        - mountPath: /data/db
          name: mongo
    volumes:
    - name: mongo
      persistentVolumeClaim:
        claimName: mongo-pvc
```

10.在node节点上需要安装ceph-common

yum -y install ceph-common

11.在对应节点 df-h

/dev/rbd1 976M 334M 627M 35% /var/lib/kubelet/plugins/kubernetes.io/rbd/mounts/rbd-image-mongo

12.进入容器查看挂载情况

```
[root@k8s-node1 ceph]# docker exec -ti 7b sh
#
# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay         420G  160G  261G  38% /
tmpfs            64M     0   64M   0% /dev
tmpfs            63G     0   63G   0% /sys/fs/cgroup
/dev/rbd1       976M  334M  627M  35% /data/db
```

/dev/mapper/zstack-root 420G 160G 261G 38% /etc/hosts
shm 64M 0 64M 0% /dev/shm
tmpfs 63G 12K 63G 1% /run/secrets/kubernetes.io/serviceaccount
tmpfs 63G 0 63G 0% /proc/acpi
tmpfs 63G 0 63G 0% /proc/scsi
tmpfs 63G 0 63G 0% /sys/firmware

可以看到宿主机通过map rbd的image到/dev下，然后挂载到对应的pod里面，所以在没有安装ceph集群的节点上需要安装ceph-common，否则rbd的映射挂载会失败。

问题:

使用静态PV创建pod，pod一直处于ContainerCreating状态:

```
# kubectl get pod ceph-pod1
NAME        READY     STATUS              RESTARTS   AGE
ceph-pod1   0/1       ContainerCreating   0          10s
......
# kubectl describe pod ceph-pod1
Warning  FailedMount            41s (x8 over 1m)  kubelet, node01
MountVolume.WaitForAttach failed for volume "ceph-pv" : fail to check rbd image
status with: (executable file not found in $PATH), rbd output: ()
Warning  FailedMount            0s                kubelet, node01           Unable to mount
volumes for pod "ceph-pod1_default(14e3a07d-93a8-11e8-95f6-000c29b1ec26)":
timeout expired waiting for volumes to attach or mount for pod "default"/"ceph-
pod1". list of unmounted volumes=[ceph-vol1]. list of unattached volumes=[ceph-
vol1 default-token-v9flt]
```

解决:node节点安装最新版的ceph-common解决该问题，ceph集群使用的是最新的mimic版本，而base源的版本太陈旧，故出现该问题

```
sudo rbd map bwiot /dev/rbd0
sudo rbd map bwiot
/dev/rbd0
```