**加我微信可进群学习交流：**

**微信号：**

luckylucky421302

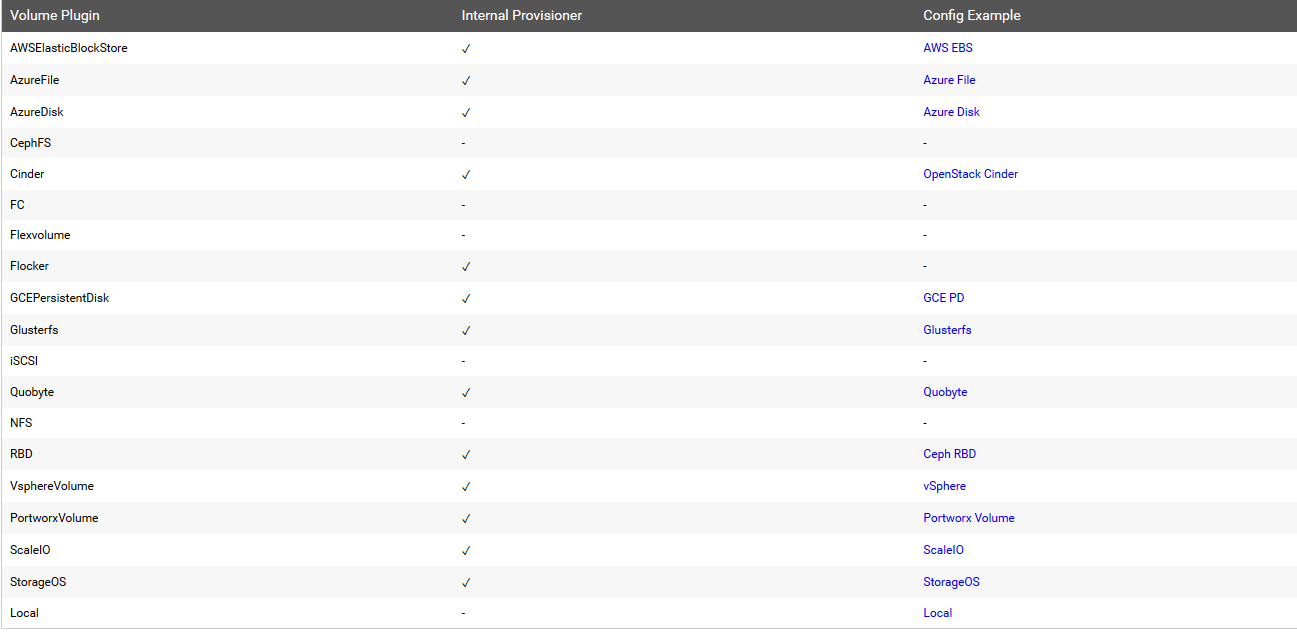
也可通过扫描下面二维码添加



**课程更新的知识点会通过微信公众号免费分享给大家，可以关注我的公众号**

****

kubernetes支持的后端存储接口



**kubernetes pv和pvc绑定流程：**

创建pv，pv需要指定后端存储类型--->创建pvc，指定需要的资源--->pvc和pv绑定--->pod挂载pvc实现持久化存储

**kubernetes storageclass存储类动态生成pv流程：**

首先创建storageclass--->pvc请求已经创建的sc，通过sc来自动创建pv-->这样就达到通过storageclass动态生成一个pv的效果了

**一、测试创建的pod直接挂载ceph rbd**

1.kubernetes要想使用ceph，需要在k8s的每个node节点安装ceph-common，把ceph节点上的ceph.repo文件拷贝到k8s各个节点上，然后yum install ceph-common -y

2.将ceph配置文件拷贝到各个k8s的节点,在ceph的管理节点操作

scp /etc/ceph/\* node1:/etc/

scp /etc/ceph/\* master1:/etc

3.测试pod直接挂载ceph的volume

在master1-admin（ceph的管理节点）上操作

ceph osd pool create k8srbd 256

rbd create rbda -s 1024 -p k8srbd

rbd feature disable k8srbd/rbda object-map fast-diff deep-flatten

4.测试pod直接挂载刚才创建的ceph rbd

cat pod-test.yaml

apiVersion: v1

kind: Pod

metadata:

name: testrbd

spec:

containers:

- image: nginx

name: nginx

volumeMounts:

- name: testrbd

mountPath: /mnt

volumes:

- name: testrbd

rbd:

monitors:

- '192.168.199.201:6789'

pool: k8srbd

image: rbda

fsType: xfs

readOnly: false

user: admin

keyring: /etc/ceph/ceph.client.admin.keyring

kubectl apply -f test.yaml

**二、基于ceph rbd创建pv，pvc**

**1.创建ceph-secret这个k8s secret对象，这个secret对象用于k8s volume插件访问ceph集群**

获取client.admin的keyring值，并用base64编码，在master1-admin（ceph管理节点）操作

ceph auth get-key client.admin | base64

QVFBczlGOWRCVTkrSXhBQThLa1k4VERQQjhVT29wd0FnZkNDQmc9PQ==

**2.创建ceph的secret，在k8s的master1上**

cat ceph-secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: ceph-secret

data:

key: QVFBczlGOWRCVTkrSXhBQThLa1k4VERQQjhVT29wd0FnZkNDQmc9PQ==

kubectl apply -f ceph-secret.yaml

3.回到ceph 管理节点创建pool池

ceph osd pool create k8stest 256

rbd create rbda -s 1024 -p k8stest

rbd feature disable k8stest/rbda object-map fast-diff deep-flatten

**3.创建ceph pv**

cat pv.yaml

apiVersion: v1 kind: PersistentVolume metadata: name: ceph-pv spec: capacity: storage: 1Gi accessModes: - ReadWriteOnce rbd: monitors: - 192.168.199.201:6789 pool: k8stest image: rbda user: admin secretRef: name: ceph-secret fsType: xfs readOnly: false persistentVolumeReclaimPolicy: Recycle

kubectl apply -f pv.yaml

**4.创建ceph pvc**

cat pvc.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: ceph-pvc

spec: accessModes:

- ReadWriteOnce

resources:

requests:

storage: 1Gi

kubectl apply -f pvc.yaml

**5.挂载使用**

cat pod.yaml

apiVersion: apps/v1beta1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 1 # tells deployment to run 2 pods matching the template

template: # create pods using pod definition in this template

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.7.9

ports:

- containerPort: 80

volumeMounts:

- mountPath: "/ceph-data"

name: ceph-data

volumes:

- name: ceph-data

persistentVolumeClaim:

claimName: ceph-pvc

kubectl apply -f pod.yaml

kubectl get pods 查看pod运行状态，如果是running则运行正常

**三、基于storageclass生成pv**

1.创建rbd的provisioner

参考：<https://github.com/kubernetes-incubator/external-storage/tree/master/ceph/rbd/deploy/rbac>

cat rbd-provisioner.yaml

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: rbd-provisioner

rules:

- apiGroups: [""]

resources: ["persistentvolumes"]

verbs: ["get", "list", "watch", "create", "delete"]

- apiGroups: [""]

resources: ["persistentvolumeclaims"]

verbs: ["get", "list", "watch", "update"]

- apiGroups: ["storage.k8s.io"]

resources: ["storageclasses"]

verbs: ["get", "list", "watch"]

- apiGroups: [""]

resources: ["events"]

verbs: ["create", "update", "patch"]

- apiGroups: [""]

resources: ["services"]

resourceNames: ["kube-dns","coredns"]

verbs: ["list", "get"]

---

kind: ClusterRoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: rbd-provisioner

subjects:

- kind: ServiceAccount

name: rbd-provisioner

namespace: default

roleRef:

kind: ClusterRole

name: rbd-provisioner

apiGroup: rbac.authorization.k8s.io

---

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

name: rbd-provisioner

rules:

- apiGroups: [""]

resources: ["secrets"]

verbs: ["get"]

- apiGroups: [""]

resources: ["endpoints"]

verbs: ["get", "list", "watch", "create", "update", "patch"]

---

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: rbd-provisioner

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: Role

name: rbd-provisioner

subjects:

- kind: ServiceAccount

name: rbd-provisioner

namespace: default

---

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: rbd-provisioner

spec:

replicas: 1

strategy:

type: Recreate

template:

metadata:

labels:

app: rbd-provisioner

spec:

containers:

- name: rbd-provisioner

image: quay.io/external\_storage/rbd-provisioner:latest

env:

- name: PROVISIONER\_NAME

value: ceph.com/rbd

serviceAccount: rbd-provisioner

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: rbd-provisioner

kubectl apply -f rbd-provisioner.yaml

2.创建ceph-secret

kubectl delete -f ceph-secret.yaml

cat ceph-secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: ceph-secret

type: "ceph.com/rbd"

data:

key: QVFBczlGOWRCVTkrSXhBQThLa1k4VERQQjhVT29wd0FnZkNDQmc9PQ==

kubectl apply -f ceph-secret.yaml

3.创建storageclass

cat storageclass.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: k8s-rbd

provisioner: kubernetes.io/rbd

parameters:

monitors: 192.168.199.201:6789

adminId: admin

adminSecretName: ceph-secret

pool: k8stest

userId: admin

userSecretName: ceph-secret

fsType: xfs

imageFormat: "2"

imageFeatures: "layering"

kubectl apply -f storageclass.yaml

4.创建pvc

cat rbd-pvc.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: rbd-pvc

spec:

accessModes:

- ReadWriteOnce

volumeMode: Filesystem

resources:

requests:

storage: 1Gi

storageClassName: k8s-rbd

kubectl apply -f rbd-pvc.yaml

5.创建pod

cat pod-sto.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

test: rbd-pod

name: ceph-rbd-pod

spec:

containers:

- name: ceph-rbd-nginx

image: nginx

volumeMounts:

- name: ceph-rbd

mountPath: /mnt

readOnly: false

volumes:

- name: ceph-rbd

persistentVolumeClaim:

claimName: rbd-pvc

kubectl apply -f pod-sto.yaml