Kubernetes 使用 ceph-docker 持久化存储

https://github.com/larkguo/Kubernetes-docker/tree/master/ceph

目录

1 部署架构

1.1 主机

主机名	IP	ceph 角色	osd 设备	宿主机目录 (配置和数据)
node1	192.168.120.173/24	mon,mgr, mds, rgw,osd	/dev/sdb	/etc/ceph/ /var/lib/ceph/
node2	192.168.120.174/24	osd	/dev/sdb	/etc/ceph/ /var/lib/ceph/
node3	192.168.120.172/24	osd	/dev/sdb	/etc/ceph/ /var/lib/ceph/
k8s-master1	192.168.120.176/24	client		
k8s-node1	192.168.120.175/24	client		

1.2 前提

部署前在说有主机上进行时钟同步

```
yum clean all
yum install chrony -y
systemctl enable chronyd.service
systemctl restart chronyd.service
```

2 Ceph 集群

Ceph 集群部署到 docker 里.

2.1 前提

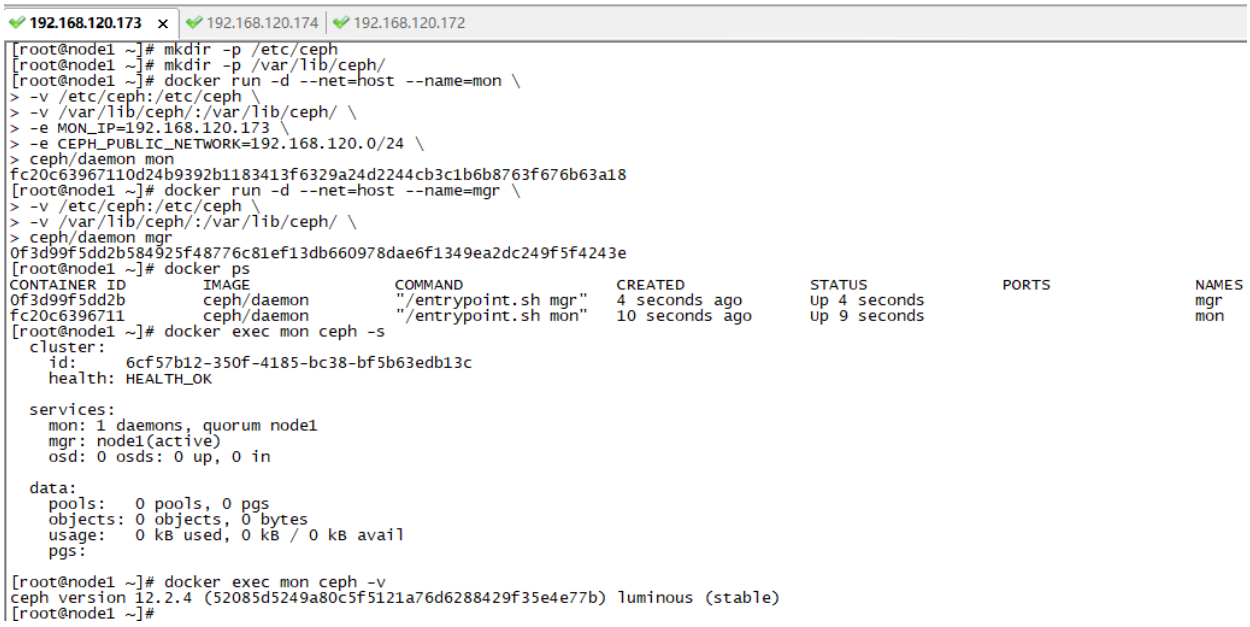
```
#在所有主机上拉取 ceph 镜像,此处下载版本为 ceph version 12.2.4 luminous,
#ceph/daemon 官方镜像地址 https://hub.docker.com/r/ceph/daemon/
docker pull  ceph/daemon
```

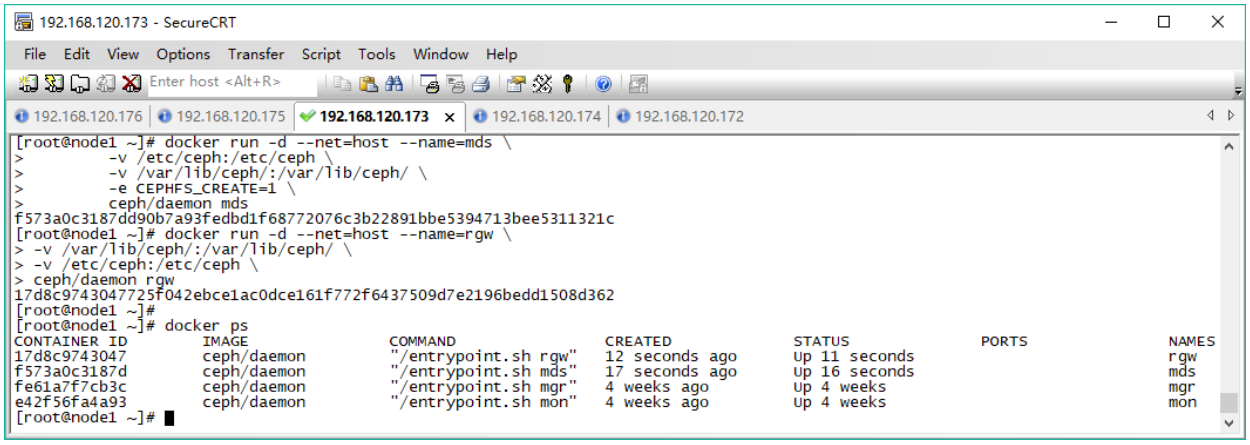
2.2 mon/mgr/mds/rgw

在 node1 安装 mon，mgr，mds 和 rgw

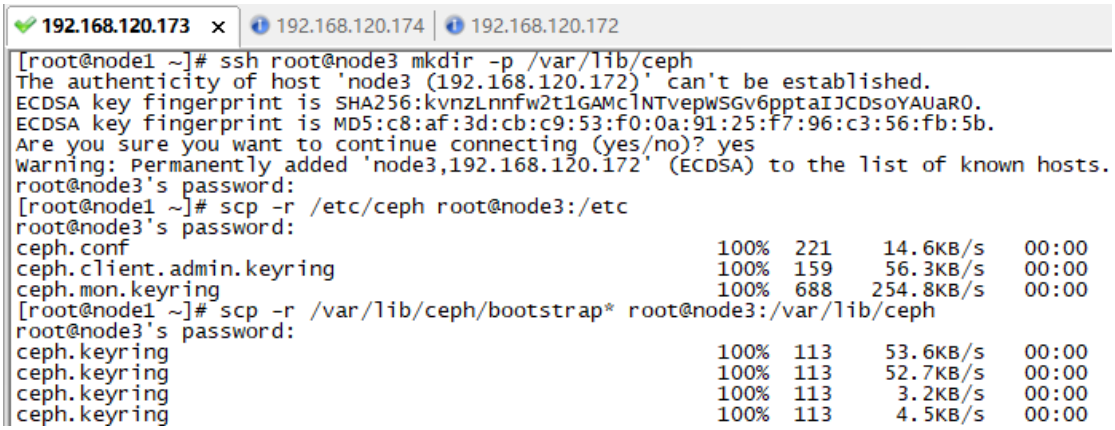
```
#在所有宿主机准备目录
mkdir -p /etc/ceph
mkdir -p /var/lib/ceph/

#清除未用的 container
docker container prune -f
rm -fr /etc/ceph/*
rm -fr /var/lib/ceph/*

# 安装 mon，MON_IP 就是宿主机的 IP 地址
docker run -d --net=host --name=mon --restart=always \
-v /etc/ceph:/etc/ceph \
-v /var/lib/ceph/:/var/lib/ceph/ \
-e MON_IP=192.168.120.173 \
-e CEPH_PUBLIC_NETWORK=192.168.120.0/24 \
ceph/daemon mon

#安装 mgr
```

```
docker run -d --net=host --name=mgr --restart=always \
-v /etc/ceph:/etc/ceph \
-v /var/lib/ceph/:/var/lib/ceph/ \
ceph/daemon mgr

#安装 mds
docker run -d --net=host --name=mds --restart=always \
        -v /etc/ceph:/etc/ceph \
        -v /var/lib/ceph/:/var/lib/ceph/ \
        -e CEPHFS_CREATE=1 \
        ceph/daemon mds

#安装 rgw
docker run -d --net=host --name=rgw --restart=always \
-v /var/lib/ceph/:/var/lib/ceph/ \
-v /etc/ceph:/etc/ceph \
ceph/daemon rgw
```





```
#复制配置文件
#将 node1 上的配置文件复制到 node02 和 node03,复制的路径包含/etc/ceph 和
#/var/lib/ceph/bootstrap-*下的所有内容。

ssh root@node2 mkdir -p /var/lib/ceph
scp -r /etc/ceph root@node2:/etc
scp -r /var/lib/ceph/bootstrap* root@node2:/var/lib/ceph

ssh root@node3 mkdir -p /var/lib/ceph
scp -r /etc/ceph root@node3:/etc
scp -r /var/lib/ceph/bootstrap* root@node3:/var/lib/ceph
```



2.3 osd

```
#在三台 osd 主机上分别执行
docker run -d --restart=always \
       --net=host \
       -v /etc/ceph:/etc/ceph \
       -v /var/lib/ceph/:/var/lib/ceph/ \
       -v /dev/:/dev/ \
       --privileged=true \
       -e OSD_FORCE_ZAP=1 \
       -e OSD_DEVICE=/dev/sdb \
       ceph/daemon osd_ceph_disk
```

192.168.120.173 | 192.168.120.174 | 192.168.120.172

```
[root@node3 ~]# docker run -d \
>         --net=host \
>         -v /etc/ceph:/etc/ceph \
>         -v /var/lib/ceph/:/var/lib/ceph/ \
>         -v /dev/:/dev/ \
>         --privileged=true \
>         -e OSD_FORCE_ZAP=1 \
>         -e OSD_DEVICE=/dev/sdc \
>         ceph/daemon osd_ceph_disk
01ca453b5665f6669d1e55c4d48a7b0d6794b889b727ae2fe479dd3e924fb231
[root@node3 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
01ca453b5665        ceph/daemon         "/entrypoint.sh os..."   4 seconds ago       Up 2 seconds                            stoic_wilson
```

#在 mon 主机上检查集群情况

```
docker exec mon ceph -s
```

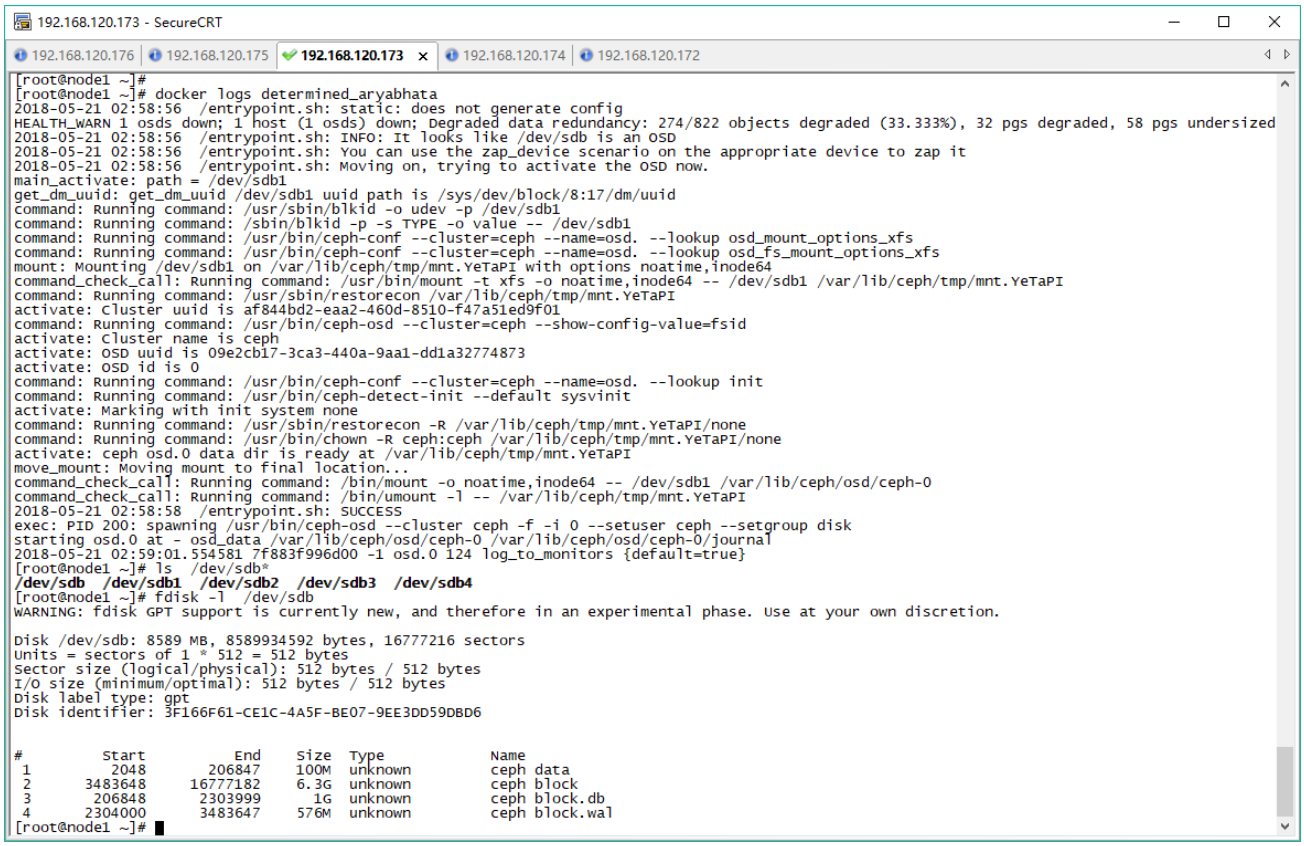
192.168.120.173 - SecureCRT

192.168.120.176 | 192.168.120.175 | 192.168.120.173 | 192.168.120.174 | 192.168.120.172

```
[root@node1 ~]# docker run -d \
>         --net=host \
>         -v /etc/ceph:/etc/ceph \
>         -v /var/lib/ceph/:/var/lib/ceph/ \
>         -v /dev/:/dev/ \
>         --privileged=true \
>         -e OSD_FORCE_ZAP=1 \
>         -e OSD_DEVICE=/dev/sdb \
>         ceph/daemon osd_ceph_disk
56401b301b07471c2792b09e05397609a409fb27d148d17cab5e6388c4c13971
[root@node1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
56401b301b07        ceph/daemon         "/entrypoint.sh os..."   7 seconds ago       Up 6 seconds                            determined_aryabhata
17d8c9743047        ceph/daemon         "/entrypoint.sh rgw"     About a minute ago  Up About a minute                       rgw
f573a0c3187d        ceph/daemon         "/entrypoint.sh mds"     About a minute ago  Up About a minute                       mds
fe61a7f7cb3c        ceph/daemon         "/entrypoint.sh mgr"     4 weeks ago         Up 4 weeks                              mgr
e42f56fa4a93        ceph/daemon         "/entrypoint.sh mon"     4 weeks ago         Up 4 weeks                              mon
[root@node1 ~]# docker exec mon ceph -s
  cluster:
    id:     af844bd2-eaa2-460d-8510-f47a51ed9f01
    health: HEALTH_WARN
            Degraded data redundancy: 50/822 objects degraded (6.083%), 2 pgs degraded

  services:
    mon: 1 daemons, quorum node1
    mgr: node1(active)
    mds: cephfs-1/1/1 up  {0=node1=up:active}
    osd: 3 osds: 3 up, 3 in
    rgw: 1 daemon active

  data:
    pools:   7 pools, 58 pgs
    objects: 274 objects, 52835 kB
    usage:   6327 MB used, 8025 MB / 14352 MB avail
    pgs:     50/822 objects degraded (6.083%)
             54 active+clean
             2  active+recovery_wait
             1  active+recovery_wait+degraded
             1  active+recovering+degraded

  io:
    recovery: 0 B/s, 8 objects/s

[root@node1 ~]# docker exec mon ceph -s
  cluster:
    id:     af844bd2-eaa2-460d-8510-f47a51ed9f01
    health: HEALTH_OK

  services:
    mon: 1 daemons, quorum node1
    mgr: node1(active)
    mds: cephfs-1/1/1 up  {0=node1=up:active}
    osd: 3 osds: 3 up, 3 in
```

#在 osd 主机上检查硬盘分区

```
fdisk -l  /dev/sdb
```



3 Kubernetes 使用 ceph

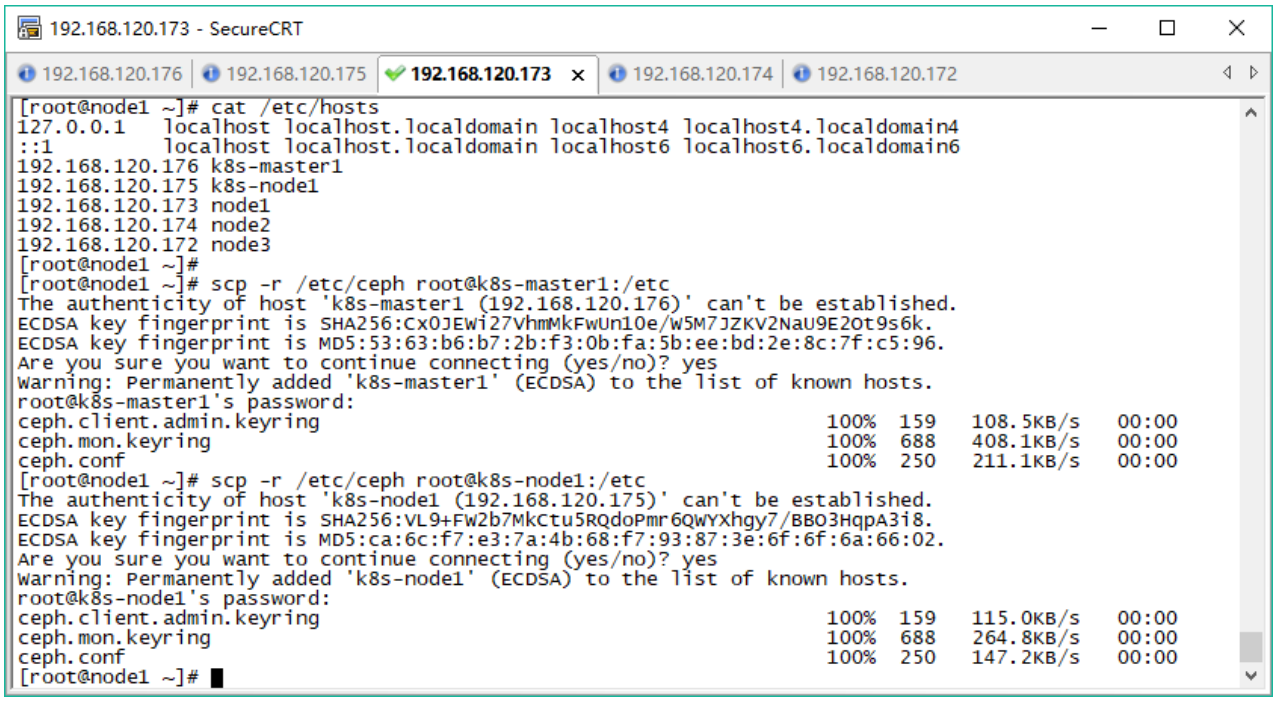
Kubernetes 三种方式使用 ceph 进行持久化：

1. mount 到本地目录使用
2. Pv/pvc 方式持久化存储
3. Storageclass 方式持久化存储

3.1 前提

3.1.1 Ceph 配置

```
#ceph 配置/etc/ceph 拷贝到所有 kubernetes 主机节点，作为 ceph 的 client 使用
scp -r /etc/ceph root@k8s-master1:/etc
scp -r /etc/ceph root@k8s-node1:/etc
```



3.1.2 安装 ceph client

```
#kubernetes 节点安装 ceph 客户端
yum install ceph-common -y
#或指定版本安装
# yum install ceph-common-12.2.4
```

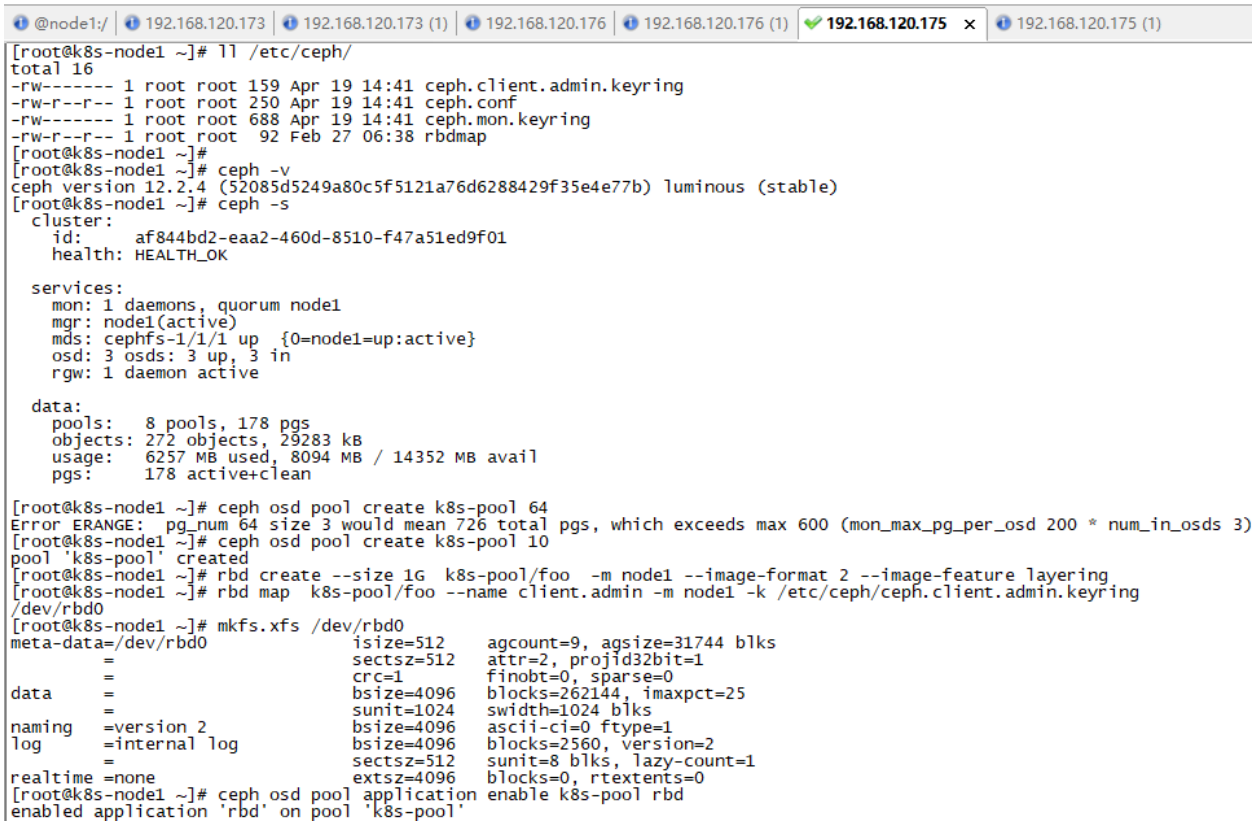
```
[root@k8s-node1 ~]# yum install ceph-common -y
Loaded plugins: fastestmirror, priorities
Loading mirror speeds from cached hostfile
 * base: mirrors.huaweicloud.com
 * epel: mirror.dmmlabs.jp
 * extras: mirrors.huaweicloud.com
 * updates: mirrors.aliyun.com
Package 2:ceph-common-12.2.5-0.el7.x86_64 already installed and latest version
Nothing to do
[root@k8s-node1 ~]# ceph -v
ceph version 12.2.5 (cad919881333ac92274171586c827e01f554a70a) luminous (stable)
```

3.1.3 Ceph 池创建

在执行创建该 Pod 之前，先在 ceph client 主机手动创建 pool k8s-pool 和 image foo，后续用到。

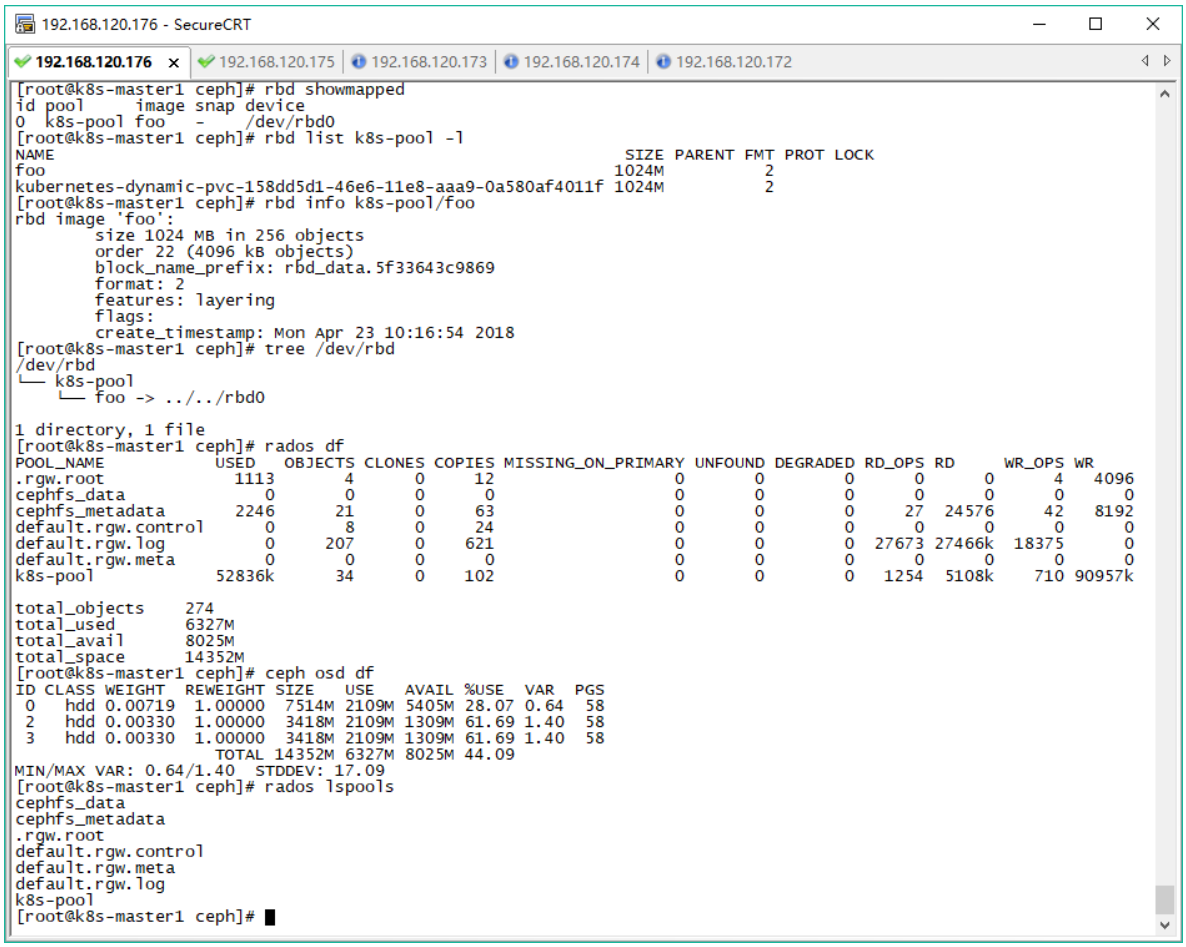
```
#ceph 集群创建一个新的存储池 k8s-pool， Placement Group 的个数为 64
ceph osd pool create k8s-pool 64

#在 ceph 池 k8s-pool 中创建一个块设备 image 镜像 foo
```

```
rbd create --size 1G  k8s-pool/foo  -m node1 --image-format 2 --image-feature
layering

# map 映射到本地
rbd map  k8s-pool/foo --name client.admin -m node1 -k
/etc/ceph/ceph.client.admin.keyring

#格式化块设备
mkfs.xfs /dev/rbd0

#指定 pool 应用类型为 rbd
ceph osd pool application enable k8s-pool rbd
```



```
[root@k8s-node1 ~]# ll /etc/ceph/
total 16
-rw------- 1 root root 159 Apr 19 14:41 ceph.client.admin.keyring
-rw-r--r-- 1 root root 250 Apr 19 14:41 ceph.conf
-rw------- 1 root root 688 Apr 19 14:41 ceph.mon.keyring
-rw-r--r-- 1 root root  92 Feb 27 06:38 rbdmap
[root@k8s-node1 ~]#
[root@k8s-node1 ~]# ceph -v
ceph version 12.2.4 (52085d5249a80c5f5121a76d6288429f35e4e77b) luminous (stable)
[root@k8s-node1 ~]# ceph -s
  cluster:
    id:     af844bd2-eaa2-460d-8510-f47a51ed9f01
    health: HEALTH_OK

  services:
    mon: 1 daemons, quorum node1
    mgr: node1(active)
    mds: cephfs-1/1/1 up  {0=node1=up:active}
    osd: 3 osds: 3 up, 3 in
    rgw: 1 daemon active

  data:
    pools:   8 pools, 178 pgs
    objects: 272 objects, 29283 kB
    usage:   6257 MB used, 8094 MB / 14352 MB avail
    pgs:     178 active+clean

[root@k8s-node1 ~]# ceph osd pool create k8s-pool 64
Error ERANGE:  pg_num 64 size 3 would mean 726 total pgs, which exceeds max 600 (mon_max_pg_per_osd 200 * num_in_osds 3)
[root@k8s-node1 ~]# ceph osd pool create k8s-pool 10
pool 'k8s-pool' created
[root@k8s-node1 ~]# rbd create --size 1G  k8s-pool/foo  -m node1 --image-format 2 --image-feature layering
[root@k8s-node1 ~]# rbd map  k8s-pool/foo --name client.admin -m node1 -k /etc/ceph/ceph.client.admin.keyring
/dev/rbd0
[root@k8s-node1 ~]# mkfs.xfs /dev/rbd0
meta-data=/dev/rbd0              isize=512    agcount=9, agsize=31744 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=1024   swidth=1024 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=8 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@k8s-node1 ~]# ceph osd pool application enable k8s-pool rbd
enabled application 'rbd' on pool 'k8s-pool'
```

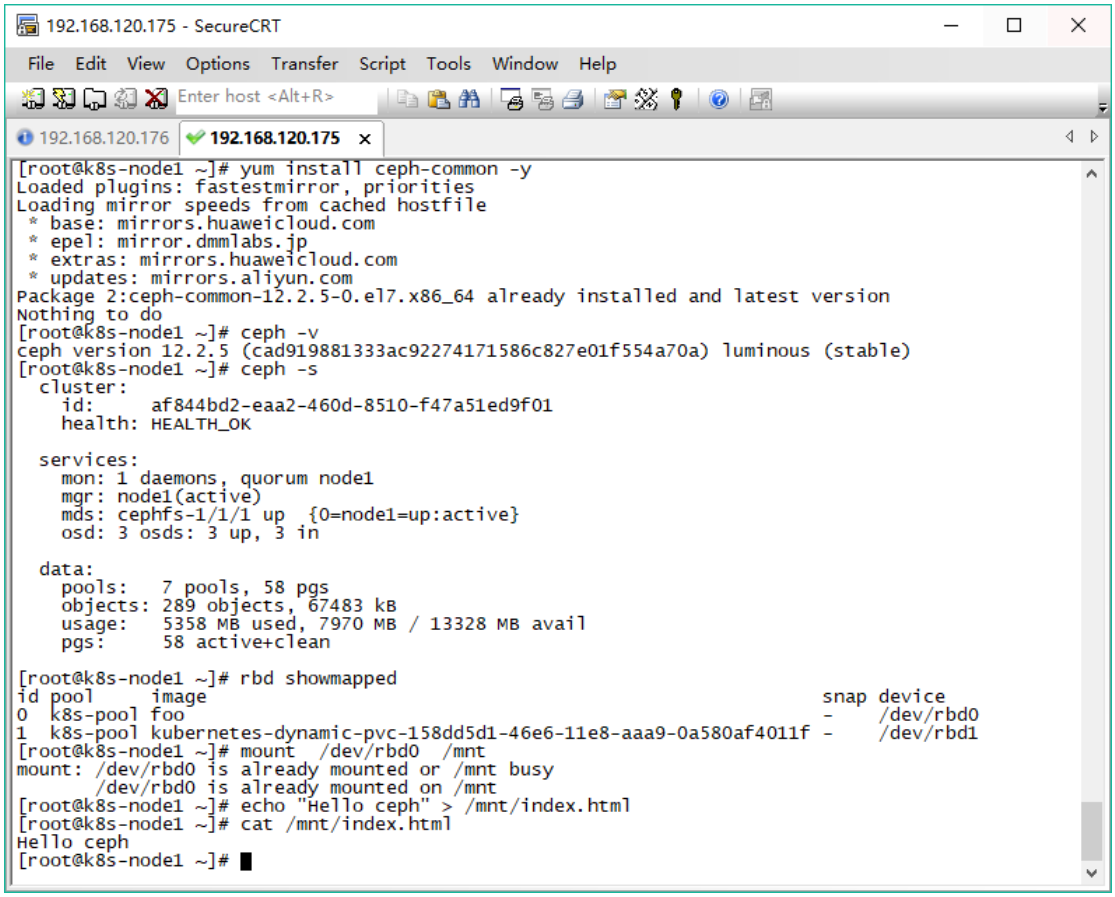
```
#查看已经映射的 Block Device 信息
rbd showmapped

#查看 k8s-pool/foo 的详细信息
rbd info  k8s-pool/foo
```



3.2 mount 到本地使用

3.2.1 mount 挂载 ceph

```
#查看已经映射的 Block Device 信息
rbd showmapped

#ceph Client 的 User Space 挂载(Mount)该 RBD 设备 /dev/rbd0 到本地目录/mnt
#把 rbd0 挂载到本地目录
mount  /dev/rbd0  /mnt

#查询
mount |grep /dev/rbd0
```



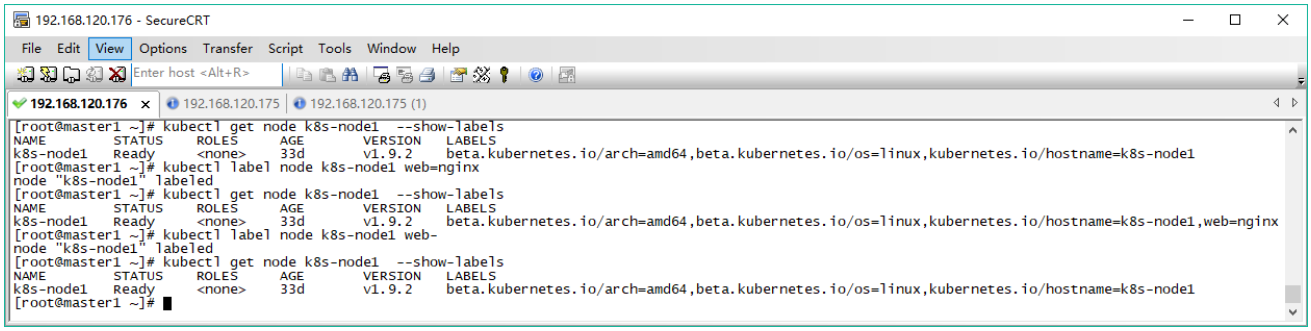
3.2.2 app 使用挂载路径

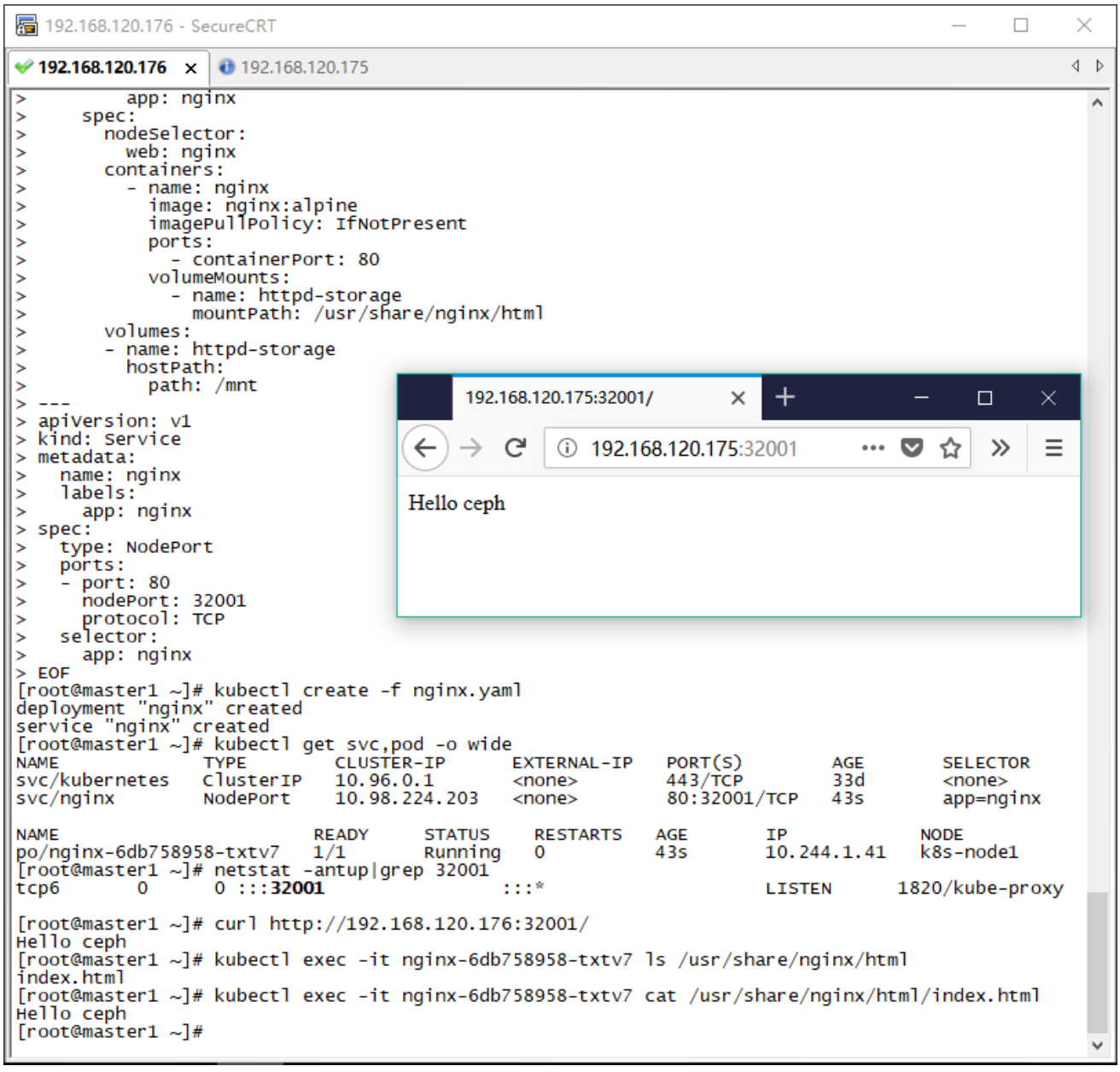
nginx 使用 ceph 挂载目录/mnt 进行持久化存储，在 master 主机节点配置。

```
#给 k8s-node1 节点添加标签
kubectl label node k8s-node1 web=nginx
```

```
#定义 nginx 应用
cat > nginx.yaml <<EOF
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
  #namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
```

```
      nodeSelector:
        web: nginx
      containers:
        - name: nginx
          image: nginx:alpine
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 80
          volumeMounts:
            - name: httpd-storage
              mountPath: /usr/share/nginx/html
      volumes:
      - name: httpd-storage
        hostPath:
          path: /mnt
---
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  type: NodePort
  ports:
  - port: 80
    nodePort: 32001
    protocol: TCP
  selector:
    app: nginx
EOF

#创建 app
kubectl create -f nginx.yaml

#测试 app
curl http://192.168.120.176:32001/
```





3.3 pv/pvc 方式使用

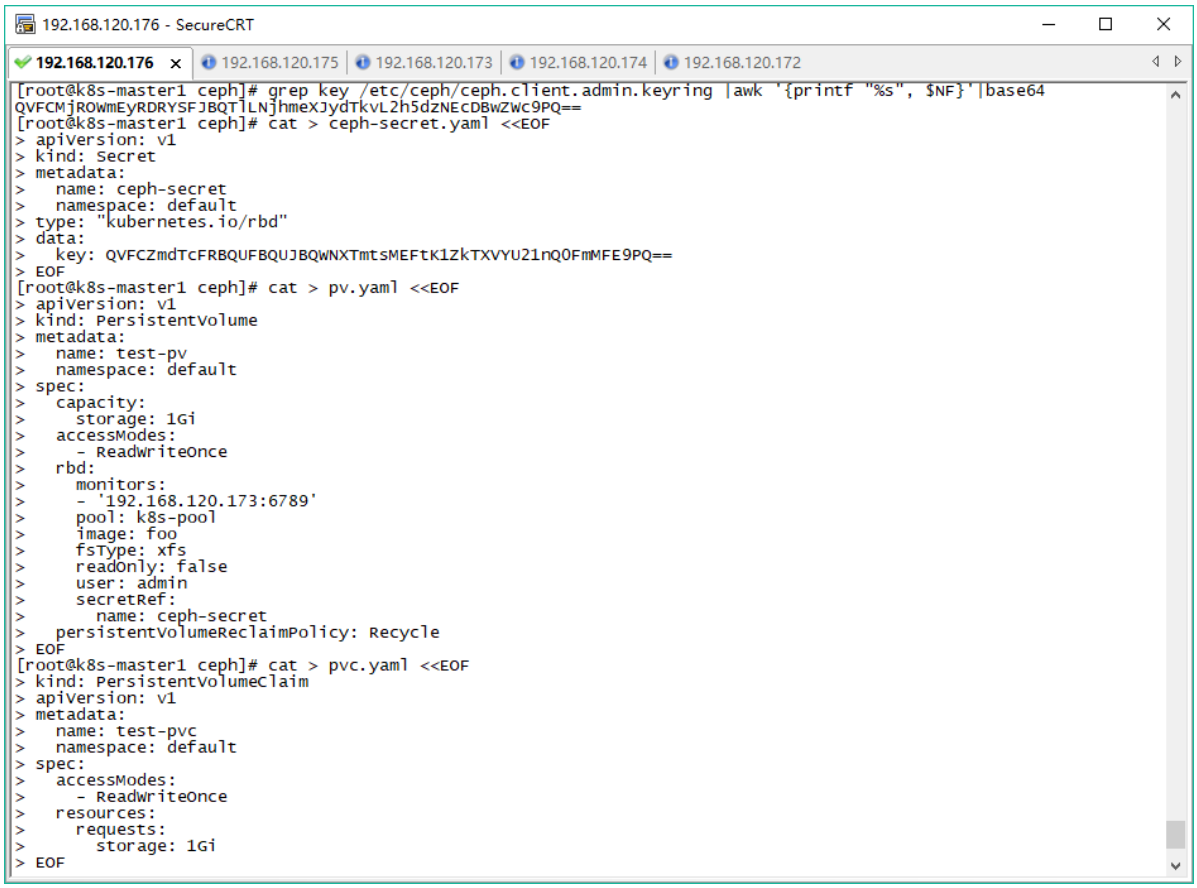
3.3.1 创建 pv/pvc

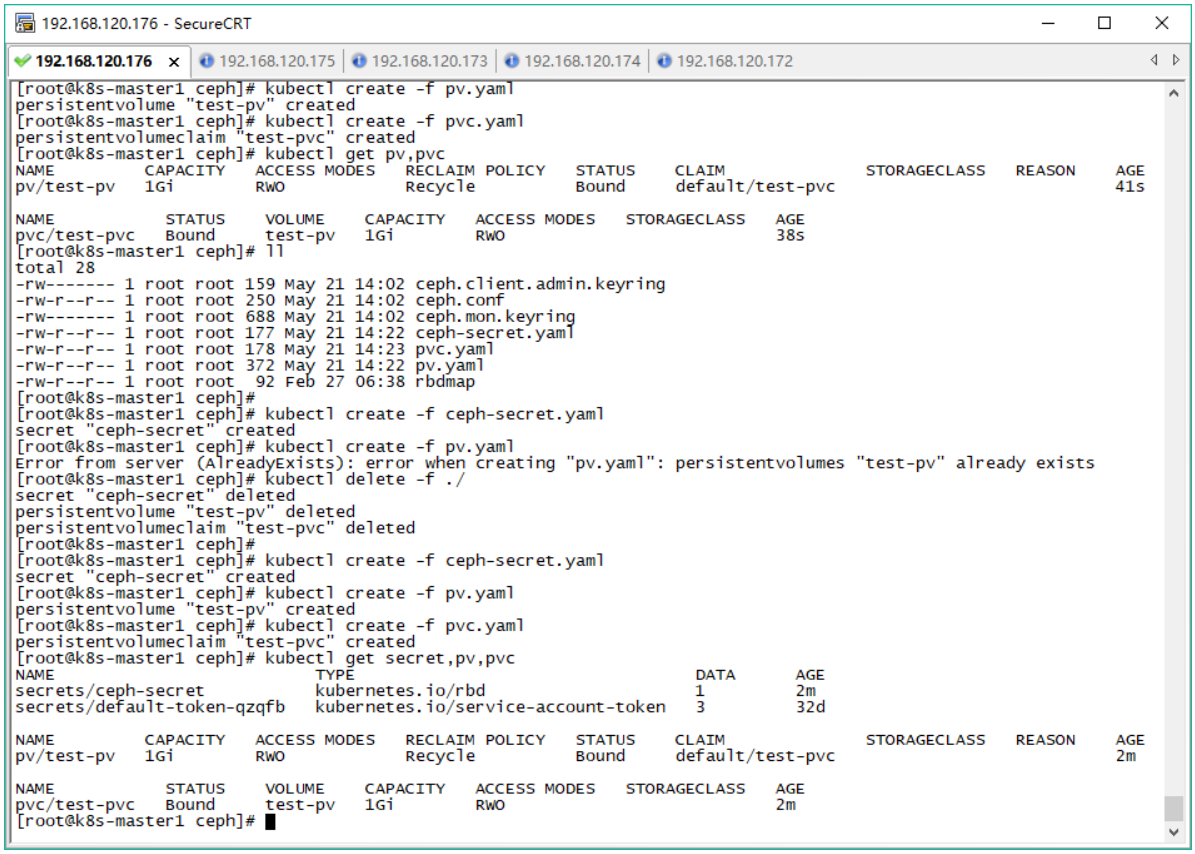
```
grep key /etc/ceph/ceph.client.admin.keyring |awk '{printf "%s", $NF}'|base64
QVFCMjROWmEyRDRYSFJBQTlLNjhmeXJydTkvL2h5dzNEcDBwZWc9PQ==
cat > ceph-secret.yaml <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: ceph-secret
  namespace: default
type: "kubernetes.io/rbd"
data:
  key: QVFCZmdTcFRBQUFBQUJBQWNXTmtsMEFtK1ZkTXVYU21nQ0FmMFE9PQ==
EOF
```

```
cat > pv.yaml <<EOF
apiVersion: v1
kind: PersistentVolume
metadata:
  name: test-pv
  namespace: default
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  rbd:
    monitors:
    - '192.168.120.173:6789'
    pool: k8s-pool
    image: foo
    fsType: xfs
    readOnly: false
    user: admin
    secretRef:
      name: ceph-secret
  persistentVolumeReclaimPolicy: Recycle
EOF
```

```
cat > pvc.yaml <<EOF
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc
  namespace: default
spec:
```

```
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
EOF

#创建 pvc/pv
kubectl create -f ceph-secret.yaml
kubectl create –f pv.yaml
kubectl create –f pvc.yaml
```

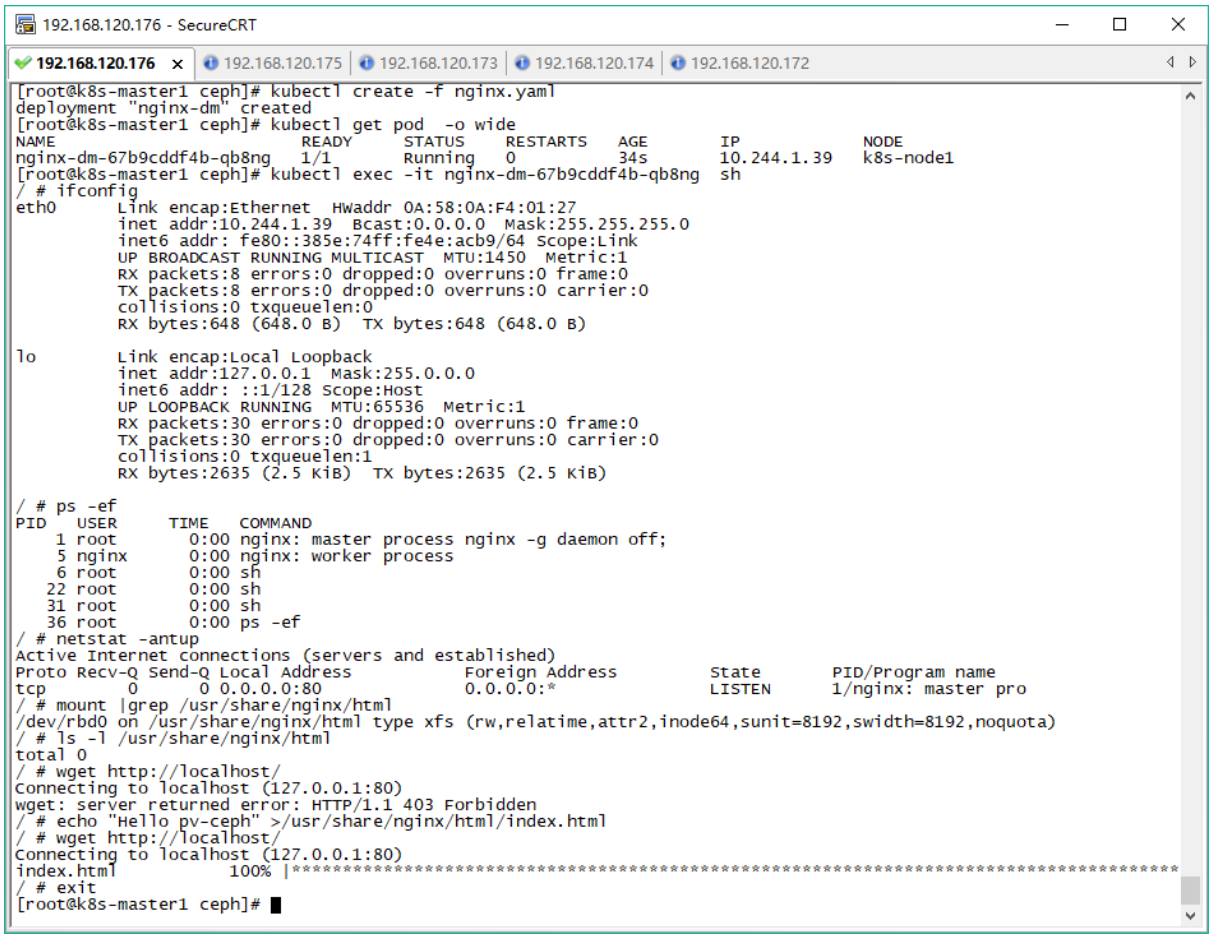




3.3.2 创建 pod

```
# pod 中目录/usr/share/nginx/html 映射到 ceph 集群
cat > nginx.yaml <<EOF
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-dm
  namespace: default
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:alpine
```

```
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 80
          volumeMounts:
            - name: ceph-rbd-volume
              mountPath: "/usr/share/nginx/html"
      volumes:
      - name: ceph-rbd-volume
        persistentVolumeClaim:
          claimName: test-pvc
EOF

#创建 pod
kubectl create -f nginx.yaml

#进入 pod 操作
```

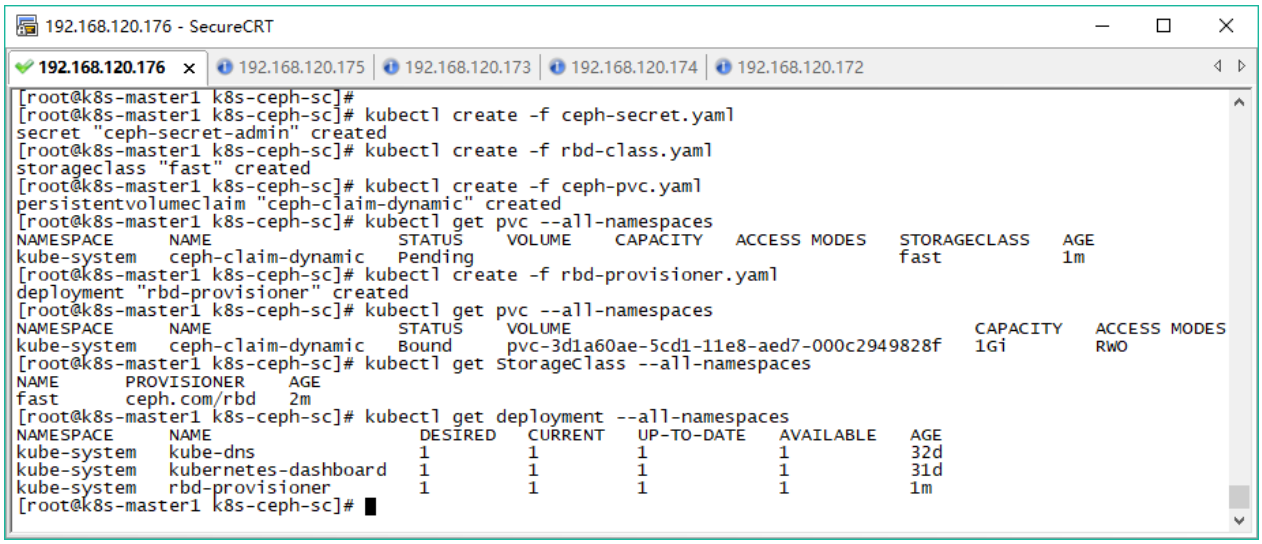


3.4 StorageClass 方式使用

3.4.1 创建 StorageClass

```
grep key /etc/ceph/ceph.client.admin.keyring |awk '{printf "%s", $NF}'|base64
QVFCMjROWmEyRDRYSFJBQTlLNjhmeXJydTkvL2h5dzNEcDBwZWc9PQ==
cat > ceph-secret.yaml <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: ceph-secret-admin
  namespace: kube-system
type: kubernetes.io/rbd
data:
  key: QVFCMjROWmEyRDRYSFJBQTlLNjhmeXJydTkvL2h5dzNEcDBwZWc9PQ==

EOF

cat > rbd-class.yaml <<EOF
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: fast
   namespace: kube-system
provisioner: ceph.com/rbd
parameters:
  monitors: 192.168.120.173:6789
  adminId: admin
  adminSecretName: ceph-secret-admin
  adminSecretNamespace: kube-system
  pool: k8s-pool
  userId: admin
  userSecretName: ceph-secret-admin
  imageFormat: "2"
  imageFeatures: layering
  fsType: xfs
EOF

cat > ceph-pvc.yaml <<EOF
kind: PersistentVolumeClaim
apiVersion: v1
```

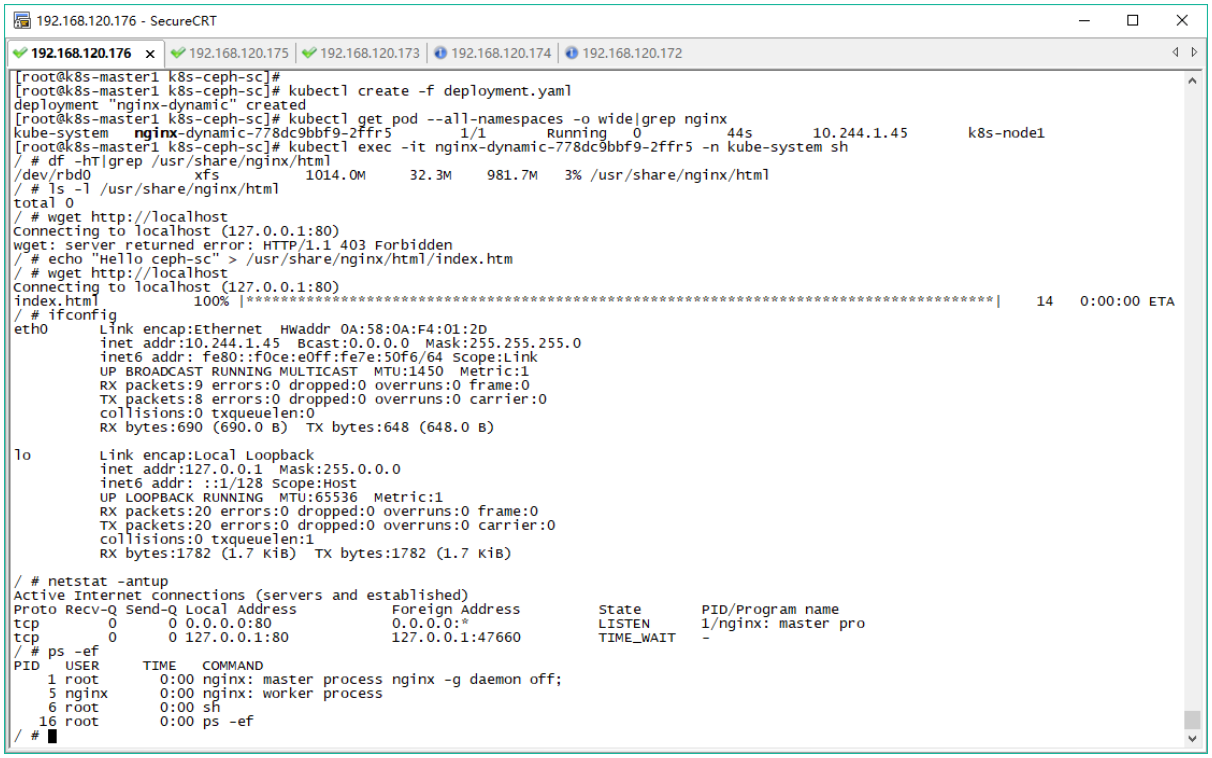
```
metadata:
 name: ceph-claim-dynamic
 namespace: kube-system
spec:
 accessModes:
    - ReadWriteOnce
 resources:
   requests:
     storage: 1Gi
 storageClassName: fast
EOF

cat > rbd-provisioner.yaml <<EOF
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: rbd-provisioner
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: rbd-provisioner
    spec:
      containers:
      - name: rbd-provisioner
        image: "quay.io/external_storage/rbd-provisioner:latest"
      serviceAccountName: persistent-volume-binder
EOF

#创建 storageclass，rbd-provisioner.yaml 创建成功后 pvc 状态变为 Bound
kubectl create -f ceph-secret.yaml
kubectl create -f rbd-class.yaml
kubectl create -f ceph-pvc.yaml
kubectl create -f rbd-provisioner.yaml
```



3.4.2 创建 deployment

```
#pod 中目录/usr/share/nginx/html 映射到 ceph 集群
cat > deployment.yaml <<EOF
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-dynamic
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:alpine
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 80
          volumeMounts:
            - name: ceph-rbd-dynamic-volume
              mountPath: /usr/share/nginx/html
      volumes:
      - name: ceph-rbd-dynamic-volume
```

```
        persistentVolumeClaim:
          claimName: ceph-claim-dynamic
EOF

kubectl create -f deployment.yaml
```



4 附录

4.1 参考

- 基于 docker 部署 ceph 以及修改 docker image

https://ceph.com/planet/%E5%9F%BA%E4%BA%8Edocker%E9%83%A8%E7%BD%B2ceph%E4%BB%A5%E5%8F%8A%E4%BF%AE%E6%94%B9docker-image/