

---

# 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.....	3
1.1	主机.....	3
1.2	前提.....	3
2	Ceph 集群.....	4
2.1	前提.....	4
2.2	mon/mgr/mds/rgw .....	4
2.3	osd .....	6
3	Kubernetes 使用 ceph.....	8
3.1	前提.....	8
3.1.1	Ceph 配置 .....	8
3.1.2	安装 ceph client .....	9
3.1.3	Ceph 池创建 .....	9
3.2	mount 到本地使用.....	11
3.2.1	mount 挂载 ceph.....	11
3.2.2	app 使用挂载路径 .....	12
3.3	pv/pvc 方式使用 .....	14
3.3.1	创建 pv/pvc.....	14
3.3.2	创建 pod.....	17
3.4	StorageClass 方式使用 .....	19
3.4.1	创建 StorageClass.....	19
3.4.2	创建 deployment.....	21
4	附录.....	22
4.1	参考.....	22

---

# 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
```



```
192.168.120.173 - SecureCRT  
192.168.120.176 192.168.120.175 192.168.120.173 x 192.168.120.174 192.168.120.172  
[root@node1 ~]#  
[root@node1 ~]# docker images | grep daemon  
ceph/daemon latest 2953709c5bed 4 weeks ago 657 MB  
[root@node1 ~]# docker image inspect ceph/daemon:latest | grep VERSION  
"CEPH_VERSION=luminous",  
"CEPH_VERSION=luminous",  
[root@node1 ~]#
```

### 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
```

```
192.168.120.173 x 192.168.120.174 192.168.120.172
[root@node1 ~]# mkdir -p /etc/ceph
[root@node1 ~]# mkdir -p /var/lib/ceph/
[root@node1 ~]# docker run -d --net=host --name=mon \
> -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
fc20c63967110d24b9392b1183413f6329a24d2244cb3c1b6b8763f676b63a18
[root@node1 ~]# docker run -d --net=host --name=mgr \
> -v /etc/ceph:/etc/ceph \
> -v /var/lib/ceph:/var/lib/ceph/ \
> ceph/daemon mgr
0f3d99f5dd2b584925f48776c81ef13db660978dae6f1349ea2dc249f5f4243e
[root@node1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS              NAMES
0f3d99f5dd2b        ceph/daemon        "/entrypoint.sh mgr"    4 seconds ago     Up 4 seconds       0.0.0.0:0->0.0.0.0:0  mgr
fc20c6396711        ceph/daemon        "/entrypoint.sh mon"    10 seconds ago    Up 9 seconds       0.0.0.0:0->0.0.0.0:0  mon
[root@node1 ~]# docker exec mon ceph -s
cluster:
  id:           6cf57b12-350f-4185-bc38-bf5b63edb13c
  health:       HEALTH_OK

services:
  mon: 1 daemons, quorum node1
  mgr: node1(active)
  osd: 0 osds: 0 up, 0 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 bytes
  usage:   0 kB used, 0 kB / 0 kB avail
  pgs:

[root@node1 ~]# docker exec mon ceph -v
ceph version 12.2.4 (52085d5249a80c5f5121a76d6288429f35e4e77b) luminous (stable)
[root@node1 ~]#
```

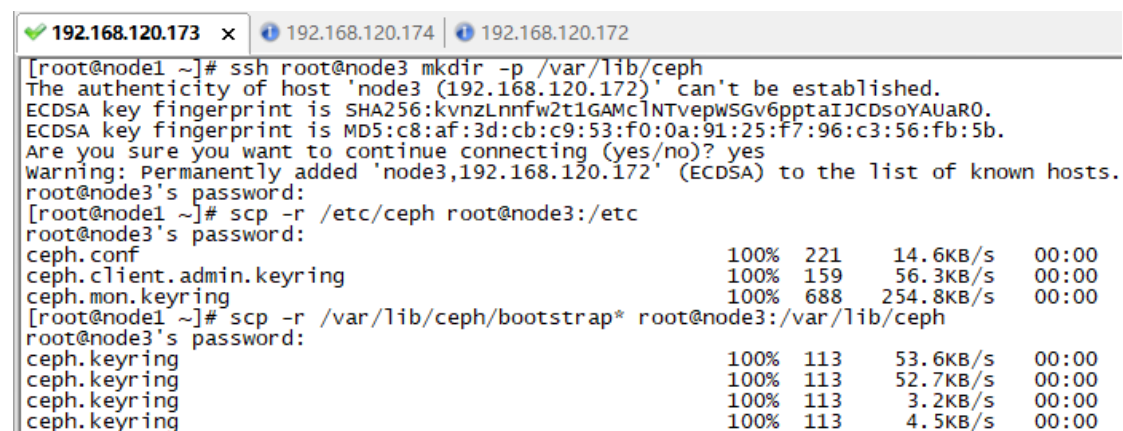
```
192.168.120.173 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.120.176 192.168.120.175 192.168.120.173 x 192.168.120.174 192.168.120.172
[root@node1 ~]# docker run -d --net=host --name=mds \
> -v /etc/ceph:/etc/ceph \
> -v /var/lib/ceph:/var/lib/ceph/ \
> -e CEPHFS_CREATE=1 \
> ceph/daemon mds
f573a0c3187dd90b7a93fedbd1f68772076c3b22891bbe5394713bee5311321c
[root@node1 ~]# docker run -d --net=host --name=rgw \
> -v /var/lib/ceph:/var/lib/ceph/ \
> -v /etc/ceph:/etc/ceph \
> ceph/daemon rgw
17d8c9743047725f042ebce1ac0dce161f772f6437509d7e2196bedd1508d362
[root@node1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS              NAMES
17d8c9743047        ceph/daemon        "/entrypoint.sh rgw"    12 seconds ago     Up 11 seconds      0.0.0.0:0->0.0.0.0:0  rgw
f573a0c3187d        ceph/daemon        "/entrypoint.sh mds"    17 seconds ago     Up 16 seconds      0.0.0.0:0->0.0.0.0:0  mds
fe61a7f7cb3c        ceph/daemon        "/entrypoint.sh mgr"    4 weeks ago        Up 4 weeks          0.0.0.0:0->0.0.0.0:0  mgr
e42f56fa4a93        ceph/daemon        "/entrypoint.sh mon"    4 weeks ago        Up 4 weeks          0.0.0.0:0->0.0.0.0:0  mon
[root@node1 ~]#
```

#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
```



The screenshot shows a terminal window with three tabs: 192.168.120.173 (active), 192.168.120.174, and 192.168.120.172. The terminal output shows the following sequence of commands and responses:

```
[root@node1 ~]# ssh root@node3 mkdir -p /var/lib/ceph
The authenticity of host 'node3 (192.168.120.172)' can't be established.
ECDSA key fingerprint is SHA256:kvnzLnnfw2t1GAMc1NTvepWSGv6pptaIJCDSoyAUaR0.
ECDSA key fingerprint is MD5:c8:af:3d:cb:c9:53:f0:0a:91:25:f7:96:c3:56:fb:5b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node3,192.168.120.172' (ECDSA) to the list of known hosts.
root@node3's password:
[root@node1 ~]# scp -r /etc/ceph root@node3:/etc
root@node3's password:
ceph.conf                                100% 221    14.6KB/s   00:00
ceph.client.admin.keyring                100% 159    56.3KB/s   00:00
ceph.mon.keyring                         100% 688   254.8KB/s   00:00
[root@node1 ~]# scp -r /var/lib/ceph/bootstrap* root@node3:/var/lib/ceph
root@node3's password:
ceph.keyring                             100% 113    53.6KB/s   00:00
ceph.keyring                             100% 113    52.7KB/s   00:00
ceph.keyring                             100% 113     3.2KB/s   00:00
ceph.keyring                             100% 113     4.5KB/s   00:00
```

## 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 x
[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 x 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-eea2-460d-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-eea2-460d-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

```
192.168.120.173 - SecureCRT
192.168.120.176 192.168.120.175 192.168.120.173 x 192.168.120.174 192.168.120.172
[root@node1 ~]#
[root@node1 ~]# docker logs determined_aryabhata
2018-05-21 02:58:56 /entrypoint.sh: static: does not generate config
HEALTH_WARN 1 osds down; 1 host (1 osds) down; Degraded data redundancy: 274/822 objects degraded (33.333%), 32 pgs degraded, 58 pgs undersized
2018-05-21 02:58:56 /entrypoint.sh: INFO: It looks like /dev/sdb is an OSD
2018-05-21 02:58:56 /entrypoint.sh: You can use the zap_device scenario on the appropriate device to zap it
2018-05-21 02:58:56 /entrypoint.sh: Moving on, trying to activate the OSD now.
main_activate: path = /dev/sdb1
get_dm_uuid: get_dm_uuid /dev/sdb1 uuid path is /sys/dev/block/8:17/dm/uuid
command: Running command: /usr/sbin/blkid -o udev -p /dev/sdb1
command: Running command: /sbin/blkid -p -s TYPE -o value -- /dev/sdb1
command: Running command: /usr/bin/ceph-conf --cluster=ceph --name=osd. --lookup osd_mount_options_xfs
command: Running command: /usr/bin/ceph-conf --cluster=ceph --name=osd. --lookup osd_fs_mount_options_xfs
mount: Mounting /dev/sdb1 on /var/lib/ceph/tmp/mnt.YeTaPI with options noatime,inode64
command_check_call: Running command: /usr/bin/mount -t xfs -o noatime,inode64 -- /dev/sdb1 /var/lib/ceph/tmp/mnt.YeTaPI
command: Running command: /usr/sbin/restorecon -R /var/lib/ceph/tmp/mnt.YeTaPI
activate: Cluster uuid is af844bd2-eaa2-460d-8510-f47a51ed9f01
command: Running command: /usr/bin/ceph-osd --cluster=ceph --show-config-value=fsid
activate: Cluster name is ceph
activate: OSD uuid is 09e2cb17-3ca3-440a-9aa1-dd1a32774873
activate: OSD id is 0
command: Running command: /usr/bin/ceph-conf --cluster=ceph --name=osd. --lookup init
command: Running command: /usr/bin/ceph-detect-init --default sysvinit
activate: Marking with init system none
command: Running command: /usr/sbin/restorecon -R /var/lib/ceph/tmp/mnt.YeTaPI/none
command: Running command: /usr/bin/chown -R ceph:ceph /var/lib/ceph/tmp/mnt.YeTaPI/none
activate: ceph osd.0 data dir is ready at /var/lib/ceph/tmp/mnt.YeTaPI
move_mount: Moving mount to final location..
command_check_call: Running command: /bin/mount -o noatime,inode64 -- /dev/sdb1 /var/lib/ceph/osd/ceph-0
command_check_call: Running command: /bin/umount -l -- /var/lib/ceph/tmp/mnt.YeTaPI
2018-05-21 02:58:58 /entrypoint.sh: SUCCESS
exec: PID 200: spawning /usr/bin/ceph-osd --cluster=ceph -f -i 0 --setuser ceph --setgroup disk
starting osd.0 at - osd_data /var/lib/ceph/osd/ceph-0 /var/lib/ceph/osd/ceph-0/journal
2018-05-21 02:59:01.554581 7f883f996d00 -1 osd.0 124 log_to_monitors {default=true}
[root@node1 ~]# ls /dev/sdb*
/dev/sdb /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4
[root@node1 ~]# fdisk -l /dev/sdb
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use at your own discretion.

Disk /dev/sdb: 8589 MB, 858934592 bytes, 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: gpt
Disk identifier: 3F166F61-CE1C-4A5F-BE07-9EE3DD59DBD6

#          Start          End      Size  Type      Name
1          2048          206847    100M  unknown  ceph data
2        3483648        16777182    6.3G  unknown  ceph block
3          206848        2303999     1G  unknown  ceph block.db
4        2304000        3483647    576M  unknown  ceph block.wal
[root@node1 ~]#
```

## 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
```



```
192.168.120.173 - SecureCRT
192.168.120.176 192.168.120.175 192.168.120.173 x 192.168.120.174 192.168.120.172
[root@node1 ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.120.176 k8s-master1
192.168.120.175 k8s-node1
192.168.120.173 node1
192.168.120.174 node2
192.168.120.172 node3
[root@node1 ~]#
[root@node1 ~]# scp -r /etc/ceph root@k8s-master1:/etc
The authenticity of host 'k8s-master1 (192.168.120.176)' can't be established.
ECDSA key fingerprint is SHA256:Cx0JEwI27VhmMkFwUn10e/w5M7JZKV2Nau9E2ot9s6k.
ECDSA key fingerprint is MD5:53:63:b6:b7:2b:f3:0b:fa:5b:ee:bd:2e:8c:7f:c5:96.
Are you sure you want to continue connecting (yes/no)? yes
warning: Permanently added 'k8s-master1' (ECDSA) to the list of known hosts.
root@k8s-master1's password:
ceph.client.admin.keyring          100% 159 108.5KB/s 00:00
ceph.mon.keyring                   100% 688 408.1KB/s 00:00
ceph.conf                          100% 250 211.1KB/s 00:00
[root@node1 ~]# scp -r /etc/ceph root@k8s-node1:/etc
The authenticity of host 'k8s-node1 (192.168.120.175)' can't be established.
ECDSA key fingerprint is SHA256:VL9+FW2b7Mkctu5RQdoPmr6QWYxhgy7/BB03HqpA3i8.
ECDSA key fingerprint is MD5:ca:6c:f7:e3:7a:4b:68:f7:93:87:3e:6f:6a:66:02.
Are you sure you want to continue connecting (yes/no)? yes
warning: Permanently added 'k8s-node1' (ECDSA) to the list of known hosts.
root@k8s-node1's password:
ceph.client.admin.keyring          100% 159 115.0KB/s 00:00
ceph.mon.keyring                   100% 688 264.8KB/s 00:00
ceph.conf                          100% 250 147.2KB/s 00:00
[root@node1 ~]# █
```

### 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
```

```
@node1:/ 192.168.120.173 192.168.120.173 (1) 192.168.120.176 192.168.120.176 (1) 192.168.120.175 x 192.168.120.175 (1)
[root@k8s-node1 ~]# ll /etc/ceph/
total 16
-rw-r--r-- 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-r--r-- 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-aaa2-460d-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          ascii-ci=0 ftype=1
log                 =internal log      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
```

```
192.168.120.176 - SecureCRT
192.168.120.176 x 192.168.120.175 192.168.120.173 192.168.120.174 192.168.120.172
[root@k8s-master1 ceph]# rbd showmapped
id pool image snap device
0 k8s-pool foo - /dev/rbd0
[root@k8s-master1 ceph]# rbd list k8s-pool -l
NAME SIZE PARENT FMT PROT LOCK
foo 1024M 2
kubernetes-dynamic-pvc-158dd5d1-46e6-11e8-aaa9-0a580af4011f 1024M 2
[root@k8s-master1 ceph]# rbd info k8s-pool/foo
rbd image 'foo':
    size 1024 MB in 256 objects
    order 22 (4096 kB objects)
    block_name_prefix: rbd_data.5f33643c9869
    format: 2
    features: layering
    flags:
    create_timestamp: Mon Apr 23 10:16:54 2018
[root@k8s-master1 ceph]# tree /dev/rbd
/dev/rbd
├── k8s-pool
│   └── foo -> ../../rbd0
1 directory, 1 file
[root@k8s-master1 ceph]# rados df
POOL_NAME USED OBJECTS CLONES COPIES MISSING_ON_PRIMARY UNFOUND DEGRADED RD_OPS RD WR_OPS WR
.rgw.root 1113 4 0 12 0 0 0 0 0 4 4096
cephfs_data 0 0 0 0 0 0 0 0 0 0 0
cephfs_metadata 2246 21 0 63 0 0 0 27 24576 42 8192
default.rgw.control 0 8 0 24 0 0 0 0 0 0 0
default.rgw.log 0 207 0 621 0 0 0 27673 27466k 18375 0
default.rgw.meta 0 0 0 0 0 0 0 0 0 0 0
k8s-pool 52836k 34 0 102 0 0 0 1254 5108k 710 90957k

total_objects 274
total_used 6327M
total_avail 8025M
total_space 14352M
[root@k8s-master1 ceph]# ceph osd df
ID CLASS WEIGHT REWEIGHT SIZE USE AVAIL %USE VAR PGS
0 hdd 0.00719 1.00000 7514M 2109M 5405M 28.07 0.64 58
2 hdd 0.00330 1.00000 3418M 2109M 1309M 61.69 1.40 58
3 hdd 0.00330 1.00000 3418M 2109M 1309M 61.69 1.40 58
TOTAL 14352M 6327M 8025M 44.09
MIN/MAX VAR: 0.64/1.40 STDDEV: 17.09
[root@k8s-master1 ceph]# rados lspools
cephfs_data
cephfs_metadata
.rgw.root
default.rgw.control
default.rgw.meta
default.rgw.log
k8s-pool
[root@k8s-master1 ceph]#
```

## 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
```

```
192.168.120.175 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
192.168.120.176 192.168.120.175 x
[root@k8s-node1 ~]# yum install ceph-common -y
Loaded plugins: fastestmirror, priorities
Loading mirror speeds from cached hostfile
* base: mirrors.huaweicloud.com
* epel: mirror.dnmlabs.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)
[root@k8s-node1 ~]# ceph -s
  cluster:
    id:         af844bd2-eea2-460d-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

  data:
    pools:   7 pools, 58 pgs
    objects: 289 objects, 67483 kb
    usage:   5358 MB used, 7970 MB / 13328 MB avail
    pgs:     58 active+clean

[root@k8s-node1 ~]# rbd showmapped
id pool image snap device
0 k8s-pool foo - /dev/rbd0
1 k8s-pool kubernetes-dynamic-pvc-158dd5d1-46e6-11e8-aaa9-0a580af4011f - /dev/rbd1
[root@k8s-node1 ~]# mount /dev/rbd0 /mnt
mount: /dev/rbd0 is already mounted or /mnt busy
/dev/rbd0 is already mounted on /mnt
[root@k8s-node1 ~]# echo "Hello ceph" > /mnt/index.html
[root@k8s-node1 ~]# cat /mnt/index.html
Hello ceph
[root@k8s-node1 ~]#
```

### 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/
```

The image displays two screenshots of a SecureCRT terminal window connected to 192.168.120.176. The top screenshot shows the output of several `kubectl` commands. The bottom screenshot shows the creation of a deployment and service for 'nginx', followed by a `curl` command and `kubectl exec` commands. A small web browser window is overlaid on the bottom screenshot, showing the URL `192.168.120.175:32001/` and the text 'Hello ceph'.

```
[root@master1 ~]# kubectl get node k8s-node1 --show-labels
NAME      STATUS    ROLES    AGE      VERSION   LABELS
k8s-node1 Ready     <none>    33d      v1.9.2    beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=k8s-node1
[root@master1 ~]# kubectl label node k8s-node1 web=nginx
node "k8s-node1" labeled
[root@master1 ~]# kubectl get node k8s-node1 --show-labels
NAME      STATUS    ROLES    AGE      VERSION   LABELS
k8s-node1 Ready     <none>    33d      v1.9.2    beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=k8s-node1,web=nginx
[root@master1 ~]# kubectl label node k8s-node1 web-
node "k8s-node1" labeled
[root@master1 ~]# kubectl get node k8s-node1 --show-labels
NAME      STATUS    ROLES    AGE      VERSION   LABELS
k8s-node1 Ready     <none>    33d      v1.9.2    beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/hostname=k8s-node1
[root@master1 ~]#
```

```
>   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
[root@master1 ~]# kubectl create -f nginx.yaml
deployment "nginx" created
service "nginx" created
[root@master1 ~]# kubectl get svc,pod -o wide
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE      SELECTOR
svc/kubernetes      ClusterIP    10.96.0.1        <none>            443/TCP           33d      <none>
svc/nginx           NodePort     10.98.224.203    <none>            80:32001/TCP      43s      app=nginx

NAME                READY        STATUS    RESTARTS   AGE      IP            NODE
po/nginx-6db758958-txtv7  1/1         Running   0           43s      10.244.1.41   k8s-node1
[root@master1 ~]# netstat -antup|grep 32001
tcp6               0 :::32001      :::*          LISTEN      1820/kube-proxy

[root@master1 ~]# curl http://192.168.120.176:32001/
Hello ceph
[root@master1 ~]# kubectl exec -it nginx-6db758958-txtv7 ls /usr/share/nginx/html
index.html
[root@master1 ~]# kubectl exec -it nginx-6db758958-txtv7 cat /usr/share/nginx/html/index.html
Hello ceph
[root@master1 ~]#
```

## 3.3 pv/pvc 方式使用

### 3.3.1 创建 pv/pvc

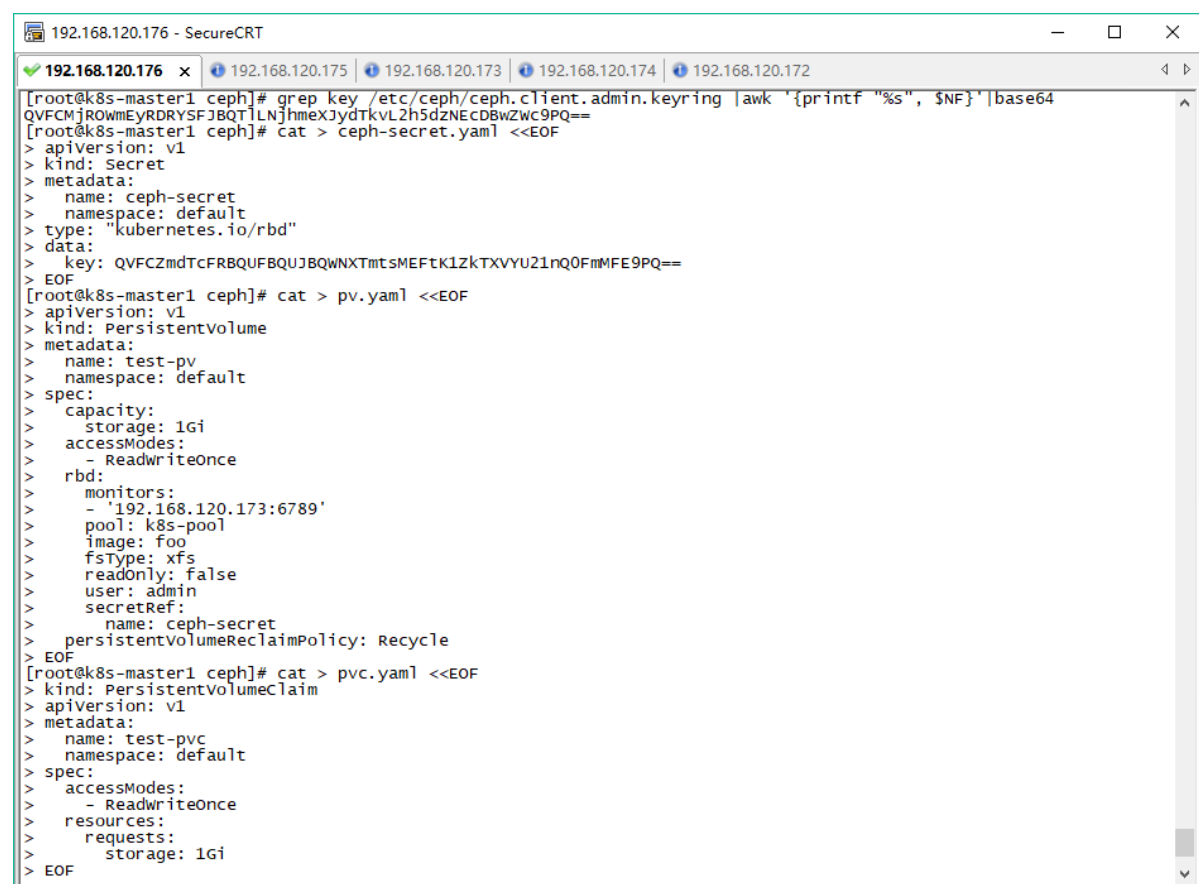
```
grep key /etc/ceph/ceph.client.admin.keyring |awk '{printf "%s", $NF}'|base64
QVFCMjROWmEyRDRYSFJBQT1LNjhmeXJydTkvL2h5dzNEcDBwZWc9PQ==
cat > ceph-secret.yaml <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: ceph-secret
  namespace: default
type: "kubernetes.io/rbd"
data:
  key: QVFCZmdTcFRBQUFBQUJBQWNXTmtsMEFtK1ZkTXVYU21nQ0FmMFE9PQ==
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
```



```
192.168.120.176 - SecureCRT
192.168.120.176 x 192.168.120.175 192.168.120.173 192.168.120.174 192.168.120.172
[root@k8s-master1 ceph]# grep key /etc/ceph/ceph.client.admin.keyring |awk '{printf "%s", $NF}'|base64
QVFCMjR0WmEyRDY5FjBQTLNjhmexJydTkVL2h5dzNEcDBwZwc9PQ==
[root@k8s-master1 ceph]# cat > ceph-secret.yaml <<EOF
> apiVersion: v1
> kind: Secret
> metadata:
>   name: ceph-secret
>   namespace: default
> type: "kubernetes.io/rbd"
> data:
>   key: QVFCZmdTcFRBQUFBQUJBQWNTMTsMEFTk1ZkTXVYU2lnQ0FmMFE9PQ==
> EOF
[root@k8s-master1 ceph]# 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
[root@k8s-master1 ceph]# cat > pvc.yaml <<EOF
> kind: PersistentVolumeClaim
> apiVersion: v1
> metadata:
>   name: test-pvc
>   namespace: default
> spec:
>   accessModes:
>     - ReadWriteOnce
>   resources:
>     requests:
>       storage: 1Gi
> EOF
```



```
192.168.120.176 - SecureCRT
192.168.120.176 x 192.168.120.175 192.168.120.173 192.168.120.174 192.168.120.172
[root@k8s-master1 ceph]# kubectl create -f pv.yaml
persistentvolume "test-pv" created
[root@k8s-master1 ceph]# kubectl create -f pvc.yaml
persistentvolumeclaim "test-pvc" created
[root@k8s-master1 ceph]# kubectl get pv,pvc
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM          STORAGECLASS  REASON  AGE
pv/test-pv    1Gi       RWO           Recycle         Bound   default/test-pvc  STORAGECLASS  REASON  AGE
pvc/test-pvc  Bound     test-pv       1Gi            RWO          STORAGECLASS  AGE
[root@k8s-master1 ceph]# ll
total 28
-rw-r--r-- 1 root root 159 May 21 14:02 ceph.client.admin.keyring
-rw-r--r-- 1 root root 250 May 21 14:02 ceph.conf
-rw-r--r-- 1 root root 688 May 21 14:02 ceph.mon.keyring
-rw-r--r-- 1 root root 177 May 21 14:22 ceph-secret.yaml
-rw-r--r-- 1 root root 178 May 21 14:23 pv.yaml
-rw-r--r-- 1 root root 372 May 21 14:22 pv.yaml
-rw-r--r-- 1 root root 92 Feb 27 06:38 rbdmap
[root@k8s-master1 ceph]#
[root@k8s-master1 ceph]# kubectl create -f ceph-secret.yaml
secret "ceph-secret" created
[root@k8s-master1 ceph]# kubectl create -f pv.yaml
Error from server (AlreadyExists): error when creating "pv.yaml": persistentvolumes "test-pv" already exists
[root@k8s-master1 ceph]# kubectl delete -f ./
secret "ceph-secret" deleted
persistentvolume "test-pv" deleted
persistentvolumeclaim "test-pvc" deleted
[root@k8s-master1 ceph]#
[root@k8s-master1 ceph]# kubectl create -f ceph-secret.yaml
secret "ceph-secret" created
[root@k8s-master1 ceph]# kubectl create -f pv.yaml
persistentvolume "test-pv" created
[root@k8s-master1 ceph]# kubectl create -f pvc.yaml
persistentvolumeclaim "test-pvc" created
[root@k8s-master1 ceph]# kubectl get secret,pv,pvc
NAME          TYPE          DATA  AGE
secrets/ceph-secret  kubernetes.io/rbd  1      2m
secrets/default-token-qzqfb  kubernetes.io/service-account-token  3      32d
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM          STORAGECLASS  REASON  AGE
pv/test-pv    1Gi       RWO           Recycle         Bound   default/test-pvc  STORAGECLASS  REASON  AGE
pvc/test-pvc  Bound     test-pv       1Gi            RWO          STORAGECLASS  AGE
[root@k8s-master1 ceph]#
```

### 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

```

EOF

#创建 pod

kubectl create -f nginx.yaml

#进入 pod 操作

```

192.168.120.176 - SecureCRT
192.168.120.176 x 192.168.120.175 192.168.120.173 192.168.120.174 192.168.120.172
[root@k8s-master1 ceph]# kubectl create -f nginx.yaml
deployment "nginx-dm" created
[root@k8s-master1 ceph]# kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE
nginx-dm-67b9cddf4b-qb8ng 1/1 Running 0 34s 10.244.1.39 k8s-node1
[root@k8s-master1 ceph]# kubectl exec -it nginx-dm-67b9cddf4b-qb8ng sh
/ # ifconfig
eth0 Link encap:Ethernet Hwaddr 0A:58:0A:F4:01:27
      inet addr:10.244.1.39 Bcast:0.0.0.0 Mask:255.255.255.0
      inet6 addr: fe80::385e:74ff:fe4e:acb9/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1450 Metric:1
      RX packets:8 errors:0 dropped:0 overruns:0 frame:0
      TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:648 (648.0 B) TX bytes:648 (648.0 B)

lo Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 Scope:Host
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:30 errors:0 dropped:0 overruns:0 frame:0
     TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1
     RX bytes:2635 (2.5 KiB) TX bytes:2635 (2.5 KiB)

/ # ps -ef
PID USER TIME COMMAND
1 root 0:00 nginx: master process nginx -g daemon off;
5 nginx 0:00 nginx: worker process
6 root 0:00 sh
22 root 0:00 sh
31 root 0:00 sh
36 root 0:00 ps -ef
/ # netstat -antup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 1/nginx: master pro
/ # mount |grep /usr/share/nginx/html
/dev/rbd0 on /usr/share/nginx/html type xfs (rw,relatime,attr2,inode64,sunit=8192,swidth=8192,noquota)
/ # ls -l /usr/share/nginx/html
total 0
/ # wget http://localhost/
Connecting to localhost (127.0.0.1:80)
wget: server returned error: HTTP/1.1 403 Forbidden
/ # echo "Hello pv-ceph" >/usr/share/nginx/html/index.html
/ # wget http://localhost/
Connecting to localhost (127.0.0.1:80)
index.html 100% |*****
/ # exit
[root@k8s-master1 ceph]#

```

---

## 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
QVFCMjROWmEyRDRYSFJBQT1LNjhmeXJydTkvL2h5dzNEcDBwZWc9PQ==
cat > ceph-secret.yaml <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: ceph-secret-admin
  namespace: kube-system
type: kubernetes.io/rbd
data:
  key: QVFCMjROWmEyRDRYSFJBQT1LNjhmeXJydTkvL2h5dzNEcDBwZWc9PQ==

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
```

```
192.168.120.176 - SecureCRT
192.168.120.176 x 192.168.120.175 192.168.120.173 192.168.120.174 192.168.120.172
[root@k8s-master1 k8s-ceph-sc]#
[root@k8s-master1 k8s-ceph-sc]# kubectl create -f ceph-secret.yaml
secret "ceph-secret-admin" created
[root@k8s-master1 k8s-ceph-sc]# kubectl create -f rbd-class.yaml
storageclass "fast" created
[root@k8s-master1 k8s-ceph-sc]# kubectl create -f ceph-pvc.yaml
persistentvolumeclaim "ceph-claim-dynamic" created
[root@k8s-master1 k8s-ceph-sc]# kubectl get pvc --all-namespaces
NAMESPACE   NAME                STATUS   VOLUME          CAPACITY   ACCESS MODES   STORAGECLASS   AGE
kube-system  ceph-claim-dynamic  Pending  pvc-3d1a60ae-5cd1-11e8-aed7-000c2949828f  1Gi        RWO            fast           1m
[root@k8s-master1 k8s-ceph-sc]# kubectl create -f rbd-provisioner.yaml
deployment "rbd-provisioner" created
[root@k8s-master1 k8s-ceph-sc]# kubectl get pvc --all-namespaces
NAMESPACE   NAME                STATUS   VOLUME          CAPACITY   ACCESS MODES   STORAGECLASS   AGE
kube-system  ceph-claim-dynamic  Bound    pvc-3d1a60ae-5cd1-11e8-aed7-000c2949828f  1Gi        RWO            fast           1m
[root@k8s-master1 k8s-ceph-sc]# kubectl get StorageClass --all-namespaces
NAME          PROVISIONER   AGE
fast          ceph.com/rbd  2m
[root@k8s-master1 k8s-ceph-sc]# kubectl get deployment --all-namespaces
NAMESPACE   NAME                DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kube-system  kube-dns            1         1         1             1           32d
kube-system  kubernetes-dashboard 1         1         1             1           31d
kube-system  rbd-provisioner     1         1         1             1           1m
[root@k8s-master1 k8s-ceph-sc]#
```

### 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
```

```
kubectl create -f deployment.yaml
```

22 / 22