搭建ceph存储

环境：rhel7.5系统虚拟机 6台

Server10:192.168.4.10 node01

Server11:192.168.4.11 node02

Server12:192.168.4.12 node03

Server13:192.168.4.13 node04

Server14:192.168.4.14 node05

Server15:192.168.4.15 node06

搭建好基础yum，关闭防火墙、SELinux

#做好免密登陆和主机名映射

[root@room8pc205 ~]# for I in {10..15};do ssh-copy-id [root@192.168.4.$i](mailto:root@192.168.4.$i) done

[root@room8pc205 ~]# for i in {10..15}; do echo -e "192.168.4.$i\tserver$i.tedu.cn\tserver$i" >> /etc/hosts

done

[root@room8pc205 ~]# tail -6 /etc/hosts

192.168.4.10 server10.tedu.cn server10

192.168.4.11 server11.tedu.cn server11

192.168.4.12 server12.tedu.cn server12

192.168.4.13 server13.tedu.cn server13

192.168.4.14 server14.tedu.cn server14

192.168.4.15 server15.tedu.cn server15

#设置ceph源，我这边用外部iso挂载的，dvd源已经在各个虚拟机上做好

[root@room8pc205 ~]# mount /dev/cdrom /var/ftp/pub/iso/ceph/

mount: /dev/sr0 写保护，将以只读方式挂载

[root@room8pc205 ~]# ls /var/ftp/pub/iso/ceph/

rhceph-2.0-rhel-7-x86\_64 rhel-7-server-openstack-9-rpms rhscon-2.0-rhel-7-x86\_64

[root@room8pc205 ~]# ls /var/ftp/pub/iso/ceph/rhceph-2.0-rhel-7-x86\_64/

EULA GPL MON OSD README RPM-GPG-KEY-redhat-release Tools TRANS.TBL

[root@room8pc205 ~]# vim ceph.repo

[root@room8pc205 ~]# cat ceph.repo

[mon]

name=mon

baseurl=ftp://192.168.4.254/pub/iso/ceph/rhceph-2.0-rhel-7-x86\_64/MON

enabled=1

gpgcheck=0

[osd]

name=osd

baseurl=ftp://192.168.4.254/pub/iso/ceph/rhceph-2.0-rhel-7-x86\_64/OSD

enabled=1

gpgcheck=0

[tools]

name=tools

baseurl=ftp://192.168.4.254/pub/iso/ceph/rhceph-2.0-rhel-7-x86\_64/Tools

enabled=1

gpgcheck=0

[root@room8pc205 ~]# for i in {10..15}

> do

> scp ceph.repo 192.168.4.$i:/etc/yum.repos.d/

> done

ceph.repo 100% 326 0.3KB/s 00:00

ceph.repo 100% 326 0.3KB/s 00:00

Warning: Permanently added '192.168.4.12' (ECDSA) to the list of known hosts.

ceph.repo 100% 326 0.3KB/s 00:00

Warning: Permanently added '192.168.4.13' (ECDSA) to the list of known hosts.

ceph.repo 100% 326 0.3KB/s 00:00

Warning: Permanently added '192.168.4.14' (ECDSA) to the list of known hosts.

ceph.repo 100% 326 0.3KB/s 00:00

Warning: Permanently added '192.168.4.15' (ECDSA) to the list of known hosts.

ceph.repo 100% 326 0.3KB/s 00:00

#确认yum源

[root@server10 ~]# yum clean all; yum repolist

源标识 源名称 状态

mon mon 41

osd osd 28

rhel75 rhel75 5,099

tools tools 33

repolist: 5,201

#配置server10为管理节点

##配置域名解析

[root@server10 ~]# for i in {10..15}

> do

> echo -e "192.168.4.$i\tserver$i.tedu.cn\tserver$i" >> /etc/hosts

> done

[root@server10 ~]# tail -6 /etc/hosts

192.168.4.10 server10.tedu.cn server10

192.168.4.11 server11.tedu.cn server11

192.168.4.12 server12.tedu.cn server12

192.168.4.13 server13.tedu.cn server13

192.168.4.14 server14.tedu.cn server14

192.168.4.15 server15.tedu.cn server15

[root@server10 ~]# for i in {10..15}

> do

> ssh-keyscan server$i >> /root/.ssh/known\_hosts

> done

# server10:22 SSH-2.0-OpenSSH\_7.4

# server10:22 SSH-2.0-OpenSSH\_7.4

# server10:22 SSH-2.0-OpenSSH\_7.4

# server11:22 SSH-2.0-OpenSSH\_7.4

# server11:22 SSH-2.0-OpenSSH\_7.4

# server11:22 SSH-2.0-OpenSSH\_7.4

# server12:22 SSH-2.0-OpenSSH\_7.4

# server12:22 SSH-2.0-OpenSSH\_7.4

# server12:22 SSH-2.0-OpenSSH\_7.4

# server13:22 SSH-2.0-OpenSSH\_7.4

# server13:22 SSH-2.0-OpenSSH\_7.4

# server13:22 SSH-2.0-OpenSSH\_7.4

# server14:22 SSH-2.0-OpenSSH\_7.4

# server14:22 SSH-2.0-OpenSSH\_7.4

# server14:22 SSH-2.0-OpenSSH\_7.4

# server15:22 SSH-2.0-OpenSSH\_7.4

# server15:22 SSH-2.0-OpenSSH\_7.4

# server15:22 SSH-2.0-OpenSSH\_7.4

[root@server10 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N ""

Generating public/private rsa key pair.

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:qL8sIqJ68yvyDxLCmgrqK1q8UDmDH/vx0tGvtS5sLgM root@server10.tedu.cn

The key's randomart image is:

+---[RSA 2048]----+

| |

| |

| |

|.. . . |

|+.\* ..S |

|o\* =E.. . |

|\* \* oo o .. |

|O+=+o=+ +... |

|&B+\*=+=\*o+o |

+----[SHA256]-----+

[root@server10 ~]# for i in {10..15}; do ssh-copy-id root@server$i; done

[root@server10 ~]# for i in {10..15}

> do

> scp /etc/hosts server$i:/etc/

> done

hosts 100% 392 132.2KB/s 00:00

hosts 100% 392 84.7KB/s 00:00

hosts 100% 392 73.7KB/s 00:00

hosts 100% 392 98.4KB/s 00:00

hosts 100% 392 70.0KB/s 00:00

hosts 100% 392 61.6KB/s 00:00

##配置ntp服务器server15

[root@room8pc205 ~]# ssh 192.168.4.15

Last login: Wed Dec 26 13:11:21 2018

[root@server15 ~]# yum -y install chrony

[root@server15 ~]# vim /etc/chrony.conf

[root@server15 ~]# cat /etc/chrony.conf | grep -v ^# | grep -v ^$

driftfile /var/lib/chrony/drift

makestep 1.0 3

rtcsync

allow 192.168.4.0/24

local stratum 10

logdir /var/log/chrony

[root@server15 ~]# systemctl restart chrony

chrony-dnssrv@ chronyd.service chrony-wait.service

[root@server15 ~]# systemctl restart chronyd.service

#将server10-server14配置为ntp客户端

[root@server10 ~]# date -s '2010-01-01 12:00:00'

2010年 01月 01日 星期五 12:00:00 CST

[root@server10 ~]# date

2010年 01月 01日 星期五 12:00:01 CST

[root@server10 ~]# ntpdate 192.168.4.15

26 Dec 14:22:14 ntpdate[14112]: step time server 192.168.4.15 offset 283486916.421397 sec

[root@server10 ~]# date

[root@server10 ~]# cat /etc/chrony.conf | grep -v ^# | grep -v ^$

server 192.168.4.15 iburst

driftfile /var/lib/chrony/drift

makestep 1.0 3

rtcsync

logdir /var/log/chrony

2018年 12月 26日 星期三 14:22:29 CST

[root@server10 ~]# for i in {11..14}

> do

> scp /etc/chrony.conf server$i:/etc/

> done

chrony.conf 100% 1130 220.4KB/s 00:00

chrony.conf 100% 1130 150.8KB/s 00:00

chrony.conf 100% 1130 169.0KB/s 00:00

chrony.conf 100% 1130 209.4KB/s 00:00

[root@server10 ~]# for i in {11..14}

> do

> ssh server$i systemctl restart chronyd

> done

##server10、server11、server12各添加三块硬盘

[root@server10 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

├─sda2 8:2 0 1G 0 part [SWAP]

└─sda3 8:3 0 8G 0 part /

vda 253:0 0 5G 0 disk

vdb 253:16 0 5G 0 disk

vdc 253:32 0 5G 0 disk

[root@server11 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

├─sda2 8:2 0 1G 0 part [SWAP]

└─sda3 8:3 0 8G 0 part /

vda 253:0 0 5G 0 disk

vdb 253:16 0 5G 0 disk

vdc 253:32 0 5G 0 disk

[root@server12 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

├─sda2 8:2 0 1G 0 part [SWAP]

└─sda3 8:3 0 8G 0 part /

vda 253:0 0 5G 0 disk

vdb 253:16 0 5G 0 disk

vdc 253:32 0 5G 0 disk

##部署ceph

#安装管理软件

[root@server10 ~]# yum -y install ceph-deploy

[root@server10 ~]# ceph-deploy --help | head -5

usage: ceph-deploy [-h] [-v | -q] [--version] [--username USERNAME]

[--overwrite-conf] [--cluster NAME] [--ceph-conf CEPH\_CONF]

COMMAND ...

[root@server10 ~]# mkdir ceph-clu

[root@server10 ~]# cd ceph-clu/

[root@server10 ceph-clu]# ceph-deploy new server10 server11 server12

[root@server10 ceph-clu]# ls

ceph.conf ceph-deploy-ceph.log ceph.mon.keyring

[root@server10 ceph-clu]# cat ceph.conf

[global]

fsid = 2ab79506-42e8-4cc2-a7da-f1561ee83e17

mon\_initial\_members = server10, server11, server12

mon\_host = 192.168.4.10,192.168.4.11,192.168.4.12

auth\_cluster\_required = cephx

auth\_service\_required = cephx

auth\_client\_required = cephx

[root@server10 ceph-clu]# cat ceph.mon.keyring

[mon.]

key = AQD9KSNcAAAAABAA7mK2/eeF9o7Nduh/Kxy9rg==

caps mon = allow \*

#装包

[root@server10 ceph-clu]# ceph-deploy install server10 server11 server12

[server10][DEBUG ] 完毕！

[server11][DEBUG ] 完毕！

[server12][DEBUG ] 完毕！

[server12][INFO ] Running command: ceph --version

[server12][DEBUG ] ceph version 10.2.2-38.el7cp (119a68752a5671253f9daae3f894a90313a6b8e4)

#初始化mon服务

[root@server10 ceph-clu]# ceph-deploy mon create-initial

#创建osd设备

##把server10、server11、server12的vda作为日志盘（为了更好的性能），vdb和dbc存储数据，vda作为vdb和dbc的日志盘

[root@server10 ceph-clu]# for i in {10..12}

> do

> ssh server$i parted /dev/vda mklabel gpt

> done

信息: You may need to update /etc/fstab.

信息: You may need to update /etc/fstab.

信息: You may need to update /etc/fstab.

[root@server10 ceph-clu]# for i in {10..12}

> do

> ssh server$i parted /dev/vda mkpart primary 1M 50%

> done

信息: You may need to update /etc/fstab.

信息: You may need to update /etc/fstab.

信息: You may need to update /etc/fstab.

[root@server10 ceph-clu]# for i in {10..12}

> do

> ssh server$i parted /dev/vda mkpart primary 50% 100%

> done

信息: You may need to update /etc/fstab.

信息: You may need to update /etc/fstab.

信息: You may need to update /etc/fstab.

[root@server10 ceph-clu]# for i in {10..12}

> do

> ssh server$i lsblk

> done

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

├─sda2 8:2 0 1G 0 part [SWAP]

└─sda3 8:3 0 8G 0 part /

vda 253:0 0 5G 0 disk

├─vda1 253:1 0 2.5G 0 part

└─vda2 253:2 0 2.5G 0 part

vdb 253:16 0 5G 0 disk

vdc 253:32 0 5G 0 disk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

├─sda2 8:2 0 1G 0 part [SWAP]

└─sda3 8:3 0 8G 0 part /

vda 253:0 0 5G 0 disk

├─vda1 253:1 0 2.5G 0 part

└─vda2 253:2 0 2.5G 0 part

vdb 253:16 0 5G 0 disk

vdc 253:32 0 5G 0 disk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

├─sda2 8:2 0 1G 0 part [SWAP]

└─sda3 8:3 0 8G 0 part /

vda 253:0 0 5G 0 disk

├─vda1 253:1 0 2.5G 0 part

└─vda2 253:2 0 2.5G 0 part

vdb 253:16 0 5G 0 disk

vdc 253:32 0 5G 0 disk

#修改分区归属

[root@server10 ceph-clu]# ll /dev/vda1

brw-rw---- 1 root disk 253, 1 12月 26 15:30 /dev/vda1

[root@server10 ceph-clu]# id ceph

uid=167(ceph) gid=167(ceph) 组=167(ceph)

[root@server10 ceph-clu]# for i in {10..12}

> do

> ssh server$i chown ceph:ceph /dev/vda?

> done

[root@server10 ceph-clu]# for i in {10..12}; do ssh server$i ls -l /dev/vda\* ; done

brw-rw---- 1 root disk 253, 0 12月 26 15:30 /dev/vda

brw-rw---- 1 ceph ceph 253, 1 12月 26 15:30 /dev/vda1

brw-rw---- 1 ceph ceph 253, 2 12月 26 15:30 /dev/vda2

brw-rw---- 1 root disk 253, 0 12月 26 15:30 /dev/vda

brw-rw---- 1 ceph ceph 253, 1 12月 26 15:30 /dev/vda1

brw-rw---- 1 ceph ceph 253, 2 12月 26 15:30 /dev/vda2

brw-rw---- 1 root disk 253, 0 12月 26 15:30 /dev/vda

brw-rw---- 1 ceph ceph 253, 1 12月 26 15:30 /dev/vda1

brw-rw---- 1 ceph ceph 253, 2 12月 26 15:30 /dev/vda2

#初始化磁盘

[root@server10 ceph-clu]# for i in {10..12}

> do

> ceph-deploy disk zap server$i:vdb server$i:vdc

> done

#做好数据盘和日志盘的映射

[root@server10 ceph-clu]# for i in {10..12}; do ceph-deploy osd create server$i:vdb:/dev/vda1 server$i:vdc:/dev/vda2; done

[ceph\_deploy.osd][DEBUG ] Host server10 is now ready for osd use.

[ceph\_deploy.osd][DEBUG ] Host server11 is now ready for osd use.

[ceph\_deploy.osd][DEBUG ] Host server12 is now ready for osd use.

#测试

[root@server10 ceph-clu]# ceph -s

cluster 2ab79506-42e8-4cc2-a7da-f1561ee83e17

health HEALTH\_OK

monmap e1: 3 mons at {server10=192.168.4.10:6789/0,server11=192.168.4.11:6789/0,server12=192.168.4.12:6789/0}

election epoch 8, quorum 0,1,2 server10,server11,server12

osdmap e31: 6 osds: 6 up, 6 in

flags sortbitwise

pgmap v46: 64 pgs, 1 pools, 0 bytes data, 0 objects

202 MB used, 30450 MB / 30653 MB avail

64 active+clean

##见到health HEALTH\_OK基础环境搭建完毕。

#ceph应用

1. 块存储：使用最多的一种方式
2. cephFS：了解，生产环境用的不多 ，不成熟
3. 对象存储：了解，使用亚马逊的s3，适合程序员用，直接调用API接口

#使用RBD（Rados块设备）

#查看存储池

[root@server10 ceph-clu]# ceph osd lspools

0 rbd,

查看到0号镜像池，名字是rbd

#创建第一个镜像

[root@server10 ceph-clu]# rbd create demo-img --image-feature layering --size 5G

[root@server10 ceph-clu]# rbd list

demo-img

[root@server10 ceph-clu]# rbd info demo-img

rbd image 'demo-img':

size 5120 MB in 1280 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.102f238e1f29

format: 2

features: layering

flags:

#创建第二个镜像，指定位置在rbd池下

[root@server10 ceph-clu]# rbd create rbd/image --image-feature layering --size 5G

[root@server10 ceph-clu]# rbd list

demo-img

image

[root@server10 ceph-clu]# rbd info image

rbd image 'image':

size 5120 MB in 1280 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.1035238e1f29

format: 2

features: layering

flags:

#创建udev规则，防止重启后日志磁盘的归属发生改变

[root@server10 ~]# vim /etc/udev/rules.d/90-cephdisk.rules

[root@server10 ~]# scp /etc/udev/rules.d/90-cephdisk.rules 192.168.4.11:/etc/udev/rules.d/

90-cephdisk.rules 100% 61 4.0KB/s 00:00

[root@server10 ~]# scp /etc/udev/rules.d/90-cephdisk.rules 192.168.4.12:/etc/udev/rules.d/

90-cephdisk.rules 100% 61 13.3KB/s 00:00

[root@server10 ~]# cat /etc/udev/rules.d/90-cephdisk.rules

ACTION=="add", KERNEL=="vda[12]", OWNER="ceph", GROUP="ceph"

#重启系统

[root@server10 ~]# ceph -s

cluster 2ab79506-42e8-4cc2-a7da-f1561ee83e17

health HEALTH\_OK

monmap e1: 3 mons at {server10=192.168.4.10:6789/0,server11=192.168.4.11:6789/0,server12=192.168.4.12:6789/0}

election epoch 16, quorum 0,1,2 server10,server11,server12

osdmap e53: 6 osds: 6 up, 6 in

flags sortbitwise

pgmap v116: 64 pgs, 1 pools, 32 bytes data, 5 objects

203 MB used, 30449 MB / 30653 MB avail

64 active+clean

##此处如果是health HAEALT\_WARN等待一会儿，看是否会变成ok，ok表示ceph可用

##如一直是warn，重启所有ceph相关服务

#缩减镜像

[root@server10 ceph-clu]# rbd resize --size 3G image --allow-shrink

Resizing image: 100% complete...done.

[root@server10 ceph-clu]# rbd info image

rbd image 'image':

size 3072 MB in 768 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.3734238e1f29

format: 2

features: layering

flags:

#扩容镜像

[root@server10 ceph-clu]# rbd resize --size 10G image

Resizing image: 100% complete...done.

[root@server10 ceph-clu]# rbd info image

rbd image 'image':

size 10240 MB in 2560 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.3734238e1f29

format: 2

features: layering

flags:

##将server15作为客户端，使用ceph创建的镜像作为存储

[root@server15 ~]# yum -y install ceph-common

#拷贝配置文件

[root@server10 ceph-clu]# scp /etc/ceph/ceph.conf /etc/ceph/ceph.client.admin.keyring server15:/etc/ceph/

ceph.conf //记录ceph集群的访问方式和地址 100% 244 60.6KB/s 00:00

ceph.client.admin.keyring //client.admin用户的秘钥 100% 63 20.1KB/s 00:00

#映射image到本地

[root@server15 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 9G 0 part

├─rhel-root 253:0 0 8G 0 lvm /

└─rhel-swap 253:1 0 1G 0 lvm [SWAP]

[root@server15 ~]# rbd map image

/dev/rbd0

[root@server15 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 10G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 9G 0 part

├─rhel-root 253:0 0 8G 0 lvm /

└─rhel-swap 253:1 0 1G 0 lvm [SWAP]

rbd0 252:0 0 10G 0 disk

[root@server15 ~]# rbd showmapped

id pool image snap device

0 rbd image - /dev/rbd0

#格式化，挂载

[root@server15 ~]# mkfs.xfs /dev/rbd0

meta-data=/dev/rbd0 isize=512 agcount=16, agsize=163840 blks

= sectsz=512 attr=2, projid32bit=1

= crc=1 finobt=0, sparse=0

data = bsize=4096 blocks=2621440, 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@server15 ~]# mount /dev/rbd0 /mnt/

[root@server15 ~]# df -hT | grep rbd

/dev/rbd0 xfs 10G 33M 10G 1% /mnt

[root@server15 ~]# echo "hello world" > /mnt/hello.txt

[root@server15 ~]# cat /mnt/hello.txt

hello world

#给镜像创建快照

[root@server15 ~]# rbd snap ls image

[root@server15 ~]# rbd snap create image --snap image-snap1

[root@server15 ~]# rbd snap ls image

SNAPID NAME SIZE

4 image-snap1 10240 MB

#模拟误删除，用快照回复数据

[root@server15 ~]# ls /mnt/

hello.txt

[root@server15 ~]# rm -rf /mnt/hello.txt

[root@server15 ~]# umount /mnt/

[root@server15 ~]# rbd snap rollback image --snap image-snap1

Rolling back to snapshot: 100% complete...done.

[root@server15 ~]# mount /dev/rbd0 /mnt/

[root@server15 ~]# ls /mnt/

hello.txt

#官方说可以在线直接回滚快照，但实际中多数不成功，所以需要先卸载镜像，然后回滚，在将镜像挂载到挂载点

##克隆快照

#首先要保护快照，防止误删除

[root@server15 ~]# rbd snap protect image --snap image-snap1

[root@server15 ~]# rbd snap ls image

SNAPID NAME SIZE

4 image-snap1 10240 MB

[root@server15 ~]# rbd snap rm image --snap image-snap1

rbd: snapshot 'image-snap1' is protected from removal.

2018-12-28 01:11:18.029032 7f2147255d80 -1 librbd::Operations: snapshot is protected

#克隆image-snap1快照

[root@server15 ~]# rbd clone image --snap image-snap1 image-clone1 --image-feature layering

[root@server15 ~]# rbd list

demo-img

image

image-clone1

[root@server15 ~]# rbd info image-clone1

rbd image 'image-clone1':

size 10240 MB in 2560 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.37553d1b58ba

format: 2

features: layering

flags:

parent: rbd/image@image-snap1

overlap: 10240 MB

#合并克隆文件

[root@server15 ~]# rbd flatten image-clone1

Image flatten: 100% complete...done.

[root@server15 ~]# rbd info image-clone1

rbd image 'image-clone1':

size 10240 MB in 2560 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.37553d1b58ba

format: 2

features: layering

flags:

#与之前相比没有了parent那一项

##删除映射

[root@server15 ~]# umount /mnt/

[root@server15 ~]# rbd showmapped

id pool image snap device

0 rbd image - /dev/rbd0

[root@server15 ~]# rbd unmap /dev/rbd/rbd/image

[root@server15 ~]# rbd showmapped

##ceph实战，使用ceph存储提供的镜像作为kvm虚拟机的硬盘

#创建虚拟机镜像

[root@server10 ceph-clu]# rbd create vm1-image --size 5G --image-feature layering

[root@server10 ceph-clu]# rbd info vm1-image

rbd image 'vm1-image':

size 5120 MB in 1280 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.1057238e1f29

format: 2

features: layering

flags:

#以物理机作为客户端安装软件包ceph-common，拷贝配置文件

[root@room8pc205 ~]# yum -y install ceph-common

[root@room8pc205 ~]# scp 192.168.4.10:/etc/ceph/ceph.conf /etc/ceph/

ceph.conf 100% 244 73.1KB/s 00:00

[root@room8pc205 ~]# scp 192.168.4.10:/etc/ceph/ceph.client.admin.keyring /etc/ceph/

ceph.client.admin.keyring 100% 63 18.4KB/s 00:00

[root@room8pc205 ~]# ls /etc/ceph/

ceph.client.admin.keyring ceph.conf rbdmap

[root@room8pc205 ~]# rbd list

demo-img

image

image-clone1

vm1-image

#手动创建一个kvm虚拟机，默认配置即可（目的是拿到虚拟机的配置文件）

#创建一个名为ceph的虚拟机，向导结束后强制关机即可

[root@room8pc205 ~]# virsh list --all | grep ceph

- ceph 关闭

#导出ceph虚拟机的声明文件，删除ceph虚拟机

[root@room8pc205 ~]# virsh dumpxml ceph > /tmp/ceph.xml

[root@room8pc205 ~]# ll /tmp/ceph.xml

-rw-r--r-- 1 root root 4209 12月 28 14:51 /tmp/ceph.xml

##虚拟机使用ceph存储，需要认证，方式是虚拟机先生成secret，在将secret与ceph账户映射

#编写账户信息文件（位置和名称没有必然要求，但内容必须准确）

[root@room8pc205 ~]# vim /tmp/secret.xml

[root@room8pc205 ~]# cat /tmp/secret.xml

<secret ephemeral='no' private='no'>

<usage type='ceph'>

<name>client.admin secret</name>

</usage>

</secret>

#生成secret

[root@room8pc205 ~]# virsh secret-define --file /tmp/secret.xml

生成 secret e8d4d800-e5e2-419a-ae31-4f4ff773820d

#查看secret

[root@room8pc205 ~]# virsh secret-list

UUID 用量

-------------------------------------------------------------------------------

e8d4d800-e5e2-419a-ae31-4f4ff773820d ceph client.admin secret

#将虚拟机软件的secret和ceph的管理员用户关联

[root@room8pc205 ~]# cat /etc/ceph/ceph.client.admin.keyring

[client.admin]

key = AQA8lCVcAaOaMxAA3UsCkq5fO6RoV4ATJnRVOg==

[root@room8pc205 ~]# virsh secret-list

UUID 用量

--------------------------------------------------------------------------------

e8d4d800-e5e2-419a-ae31-4f4ff773820d ceph client.admin secret

[root@room8pc205 ~]# virsh secret-set-value –secret e8d4d800-e5e2-419a-ae31-4f4ff773820d --base64 AQA8lCVcAaOaMxAA3UsCkq5fO6RoV4ATJnRVOg==

secret 值设定

#修改虚拟机的配置文件（/tmp/ceph.xml）把管理员信息写入该文件，并指定虚拟机使用的ceph磁盘

[root@room8pc205 ~]# vim /tmp/ceph.xml

#原始文件

<disk type='file' device='disk'>

<driver name='qemu' type='qcow2'/>

<source file='/var/lib/libvirt/images/ceph.qcow2'/>

<target dev='vda' bus='virtio'/>

<address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>

</disk>

#修改后的文件

<disk type='network' device='disk'>

<driver name='qemu' type='raw'/>

<auth username='admin'>

<secret type='ceph' uuid='e8d4d800-e5e2-419a-ae31-4f4ff773820d'/>

</auth>

<source protocol='rbd' name='rbd/vm1-image'>

<host name='192.168.4.10' port='6789' />

</source>

<target dev='vda' bus='virtio'/>

<address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>

</disk>

#利用xml文件生成虚拟机

[root@room8pc205 ~]# virsh define /tmp/ceph.xml

定义域 ceph（从 /tmp/ceph.xml）

[root@room8pc205 ~]# virsh list --all | grep ceph

- ceph 关闭

#修改虚拟机的启动项，由光盘引导，可安装系统即可

#表示虚拟机的磁盘文件放到ceph存储上了，并不在本地的/var/lib/libvirt/images/

#系统安装完成后关闭虚拟机，调整启动项为硬盘引导

##CEPHFS可以想nfs、Samba那样，提供共享文件夹，客户端通过挂载目录的方式使用ceph存储，目前技术还不成熟，只做介绍参考

#把server13作为MDS元数据服务器

#元数据（Metadata）是指用来描述一个文件的特性的系统数据

#安装mds服务

[root@server13 ~]# yum -y install ceph-mds

#到ceph管理节点配置mds

[root@server10 ceph-clu]# ceph-deploy mds create server13

Unhandled exception in thread started by

#将管理秘钥同步到mds服务器

[root@server10 ceph-clu]# ceph-deploy admin server13

[root@server13 ~]# ls /etc/ceph/

ceph.client.admin.keyring ceph.conf rbdmap tmp6dzBUv

#创建两个池，一个用于存储数据，一个用于存储元数据

[root@server10 ceph-clu]# ceph osd pool create cephfs\_data 128

pool 'cephfs\_data' created

[root@server10 ceph-clu]# ceph osd pool create cephfs\_metadata 128

pool 'cephfs\_metadata' created

[root@server10 ceph-clu]# ceph osd pool ls

rbd

cephfs\_data

cephfs\_metadata

#注意，128表示PG数目是128.PG是归置组，文件存到PG中，PG在池中

#创建名为myfs1的文件系统

[root@server10 ceph-clu]# ceph fs new myfs1 cephfs\_metadata cephfs\_data

new fs with metadata pool 2 and data pool 1

[root@server10 ceph-clu]# ceph fs ls

name: myfs1, metadata pool: cephfs\_metadata, data pools: [cephfs\_data ]

[root@server10 ceph-clu]# ceph mds stat

e5: 1/1/1 up {0=server13=up:active}

#使用cephfs文件系统 server15节点

[root@server15 ~]# mkdir /mnt/cephfs

[root@server15 ~]# cat /etc/ceph/ceph.client.admin.keyring

[client.admin]

key = AQA8lCVcAaOaMxAA3UsCkq5fO6RoV4ATJnRVOg==

[root@server15 ~]# df -hT | grep ceph

[root@server15 ~]# mount -t ceph 192.168.4.10:6789:/ /mnt/cephfs/ -o name=admin,secret=AQA8lCVcAaOaMxAA3UsCkq5fO6RoV4ATJnRVOg==

[root@server15 ~]# df -hT | grep ceph

192.168.4.10:6789:/ ceph 30G 4.5G 26G 15% /mnt/cephfs

#cephfs成功挂载

##ceph对象存储

#server14做成rgw设备

#使用ceph对象存储，需要RGW，即ceph网关，配置server14是RGW

[root@server10 ceph-clu]# ceph-deploy install --rgw server14

[server14][DEBUG ] 完毕！

#该命令等同于[root@server14 ~]# yum -y install ceph-radosgw

#同步配置文件和秘钥文件

[root@server14 ~]# ls /etc/ceph/

Rbdmap

[root@server10 ceph-clu]# ceph-deploy admin server14

[root@server14 ~]# ls /etc/ceph/

ceph.client.admin.keyring ceph.conf rbdmap tmpkj1Aum

[root@server10 ceph-clu]# ceph-deploy rgw create server14

[ceph\_deploy.rgw][INFO ] The Ceph Object Gateway (RGW) is now running on host server14 and default port 7480

[root@server14 ~]# netstat -antpu | grep 7480

tcp 0 0 0.0.0.0:7480 0.0.0.0:\* LISTEN 5074/radosgw

#rgw内建了一个civetweb的web服务器，与apache、nginx类，因为rgw默认使用7480端口，可以修改为80端口，方便访问

[root@server14 ~]# vim /etc/ceph/ceph.conf

[root@server14 ~]# tail -3 /etc/ceph/ceph.conf

[client.rgw.server14]

host = server14

rgw\_frontends = "civetweb port=80"

[root@server14 ~]# systemctl restart ceph-radosgw\\*

[root@server14 ~]# netstat -antpu | grep radosgw

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 5534/radosgw

tcp 0 0 192.168.4.14:52028 192.168.4.11:6800 ESTABLISHED 5534/radosgw

tcp 0 0 192.168.4.14:40858 192.168.4.12:6800 ESTABLISHED 5534/radosgw

tcp 0 0 192.168.4.14:43458 192.168.4.11:6789 ESTABLISHED 5534/radosgw

tcp 0 0 192.168.4.14:50258 192.168.4.10:6804 ESTABLISHED 5534/radosgw

tcp 0 0 192.168.4.14:38258 192.168.4.10:6800 ESTABLISHED 5534/radosgw

tcp 0 0 192.168.4.14:48666 192.168.4.11:6804 ESTABLISHED 5534/radosgw

tcp 0 0 192.168.4.14:37830 192.168.4.12:6804 ESTABLISHED 5534/radosgw

#验证端口

[root@server15 ~]# curl http://192.168.4.14

<?xml version="1.0" encoding="UTF-8"?><ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Owner><ID>anonymous</ID><DisplayName></DisplayName></Owner><Buckets></Buckets></ListAllMyBucketsResult>[root@server15 ~]#

#屏幕有html代码表示成功

#server15上安装s3客户端软件

[root@server15 ~]# ls s3cmd-2.0.1-1.el7.noarch.rpm

s3cmd-2.0.1-1.el7.noarch.rpm

[root@server15 ~]# rpm -ivh s3cmd-2.0.1-1.el7.noarch.rpm

#创建用户

[root@server14 ~]# radosgw-admin user create --uid="mark" --display-name="mark"

{

"user\_id": "mark",

"display\_name": "mark",

"email": "",

"suspended": 0,

"max\_buckets": 1000,

"auid": 0,

"subusers": [],

"keys": [

{

"user": "mark",

"access\_key": "OGU86G30639GAH1SKDVK",

"secret\_key": "cvCfmQUveTJ5veApi5WDK44EeqBuGqW6Zp7CQ4EZ"

}

],

"swift\_keys": [],

"caps": [],

"op\_mask": "read, write, delete",

"default\_placement": "",

"placement\_tags": [],

"bucket\_quota": {

"enabled": false,

"max\_size\_kb": -1,

"max\_objects": -1

},

"user\_quota": {

"enabled": false,

"max\_size\_kb": -1,

"max\_objects": -1

},

"temp\_url\_keys": []

}

#配置s3客户端

[root@server15 ~]# s3cmd --configure

Enter new values or accept defaults in brackets with Enter.

Refer to user manual for detailed description of all options.

Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.

Access Key: OGU86G30639GAH1SKDVK

Secret Key: cvCfmQUveTJ5veApi5WDK44EeqBuGqW6Zp7CQ4EZ

Default Region [US]: //注意不要修改

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.

S3 Endpoint [s3.amazonaws.com]: 192.168.4.14

Use "%(bucket)s.s3.amazonaws.com" to the target Amazon S3. "%(bucket)s" and "%(location)s" vars can be used

if the target S3 system supports dns based buckets.

DNS-style bucket+hostname:port template for accessing a bucket [%(bucket)s.s3.amazonaws.com]: %(bucket)s.192.168.4.14 //固定格式

Encryption password is used to protect your files from reading

by unauthorized persons while in transfer to S3

Encryption password:

Path to GPG program [/usr/bin/gpg]:

When using secure HTTPS protocol all communication with Amazon S3

servers is protected from 3rd party eavesdropping. This method is

slower than plain HTTP, and can only be proxied with Python 2.7 or newer

Use HTTPS protocol [Yes]: No

On some networks all internet access must go through a HTTP proxy.

Try setting it here if you can't connect to S3 directly

HTTP Proxy server name:

New settings:

Access Key: OGU86G30639GAH1SKDVK

Secret Key: cvCfmQUveTJ5veApi5WDK44EeqBuGqW6Zp7CQ4EZ

Default Region: US

S3 Endpoint: 192.168.4.14

DNS-style bucket+hostname:port template for accessing a bucket: %(bucket)s.192.168.4.14

Encryption password:

Path to GPG program: /usr/bin/gpg

Use HTTPS protocol: False

HTTP Proxy server name:

HTTP Proxy server port: 0

Test access with supplied credentials? [Y/n] Y

Please wait, attempting to list all buckets...

Success. Your access key and secret key worked fine :-)

Now verifying that encryption works...

Not configured. Never mind.

Save settings? [y/N] y

Configuration saved to '/root/.s3cfg'

##客户端测试

#创建bucket，注意格式my\_dir

[root@server15 ~]# s3cmd ls

[root@server15 ~]# s3cmd mb s3://my\_dir

Bucket 's3://my\_dir/' created

[root@server15 ~]# s3cmd ls

2018-12-28 09:45 s3://my\_dir

#上传文件

[root@server15 ~]# s3cmd put /etc/hosts s3://my\_dir

upload: '/etc/hosts' -> 's3://my\_dir/hosts' [1 of 1]

392 of 392 100% in 2s 159.56 B/s done

[root@server15 ~]# s3cmd ls s3://my\_dir

2018-12-28 09:46 392 s3://my\_dir/hosts

#下载文件

[root@server15 ~]# ls

s3cmd-2.0.1-1.el7.noarch.rpm

[root@server15 ~]# s3cmd get s3://my\_dir/hosts ./zhuji

download: 's3://my\_dir/hosts' -> './zhuji' [1 of 1]

392 of 392 100% in 0s 11.84 kB/s done

[root@server15 ~]# ls

s3cmd-2.0.1-1.el7.noarch.rpm zhuji

[root@server15 ~]# tail -1 zhuji

192.168.4.15 server15.tedu.cn server15

#删除文件

[root@server15 ~]# s3cmd ls s3://my\_dir

2018-12-28 09:46 392 s3://my\_dir/hosts

[root@server15 ~]# s3cmd del s3://my\_dir/hosts

delete: 's3://my\_dir/hosts'

[root@server15 ~]# s3cmd ls s3://my\_dir

##完毕