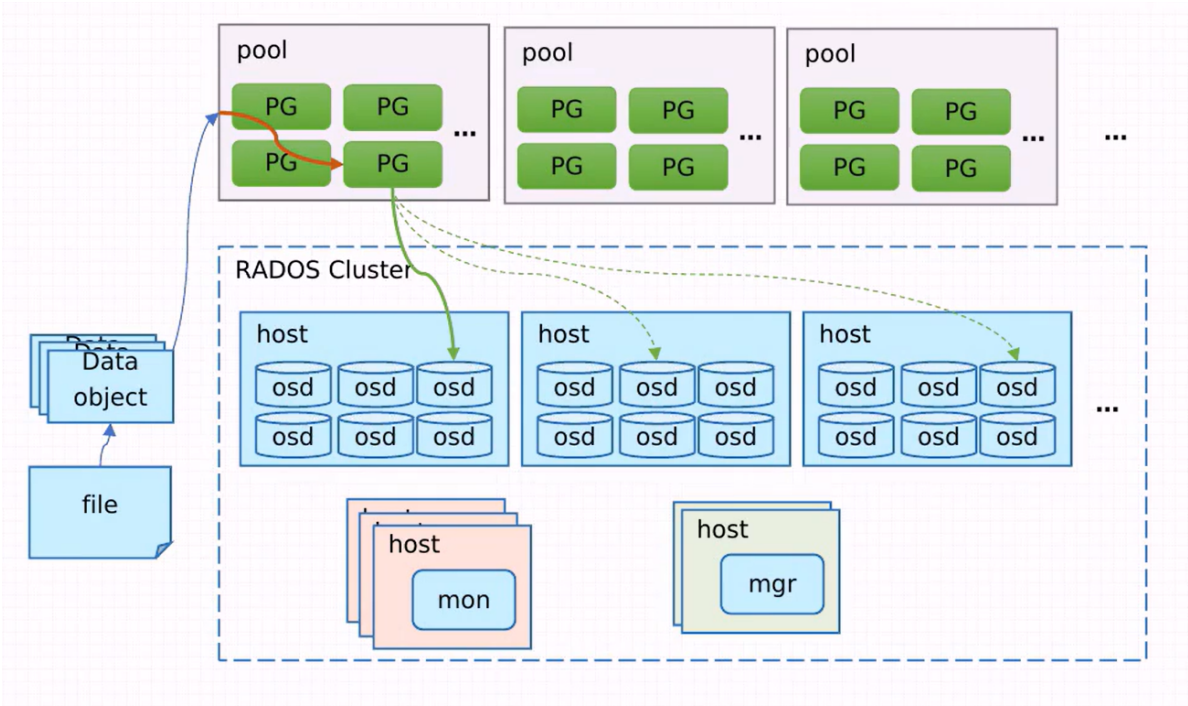
ceph分布式存储集群部署

一、 集群原理图



二、集群规划

主机名	IP地址	角色
ceph-0	100.65.20.11	mon,mgr,osd，ceph-deploy管理节点
ceph-1	100.65.20.12	mon,mgr,osd
ceph-2	100.65.20.13	mon,mgr,osd

三、系统优化

前提条件：

离线安装，需提前配置好局域网yum源： centos os源，epel源，ceph源

1. 禁用selinux

```
setenforce 0
sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

2. 关闭防火墙

```
systemctl stop firewalld ; systemctl disable firewalld
```

3. 配置NTP时间同步

```
yum install chrony -y
sed -i '/iburst/d' /etc/chrony.conf
echo "server 100.65.34.36  iburst"  >>  /etc/chrony.conf
systemctl enable chronyd --now
```

4. 添加本地解析

```
cat <<EOF >> /etc/hosts
100.65.20.11  ceph-0
100.65.20.12  ceph-1
100.65.20.13  ceph-2
EOF
```

5. 创建部署ceph用户

后期部署管理集群都用cephadm这个用户

```
useradd cephadm && echo cephadm123 | passwd --stdin cephadm
echo "cephadm  ALL=(ALL)  NOPASSWD: ALL" | sudo tee /etc/sudoers.d/cephadm
chmod 0440 /etc/sudoers.d/cephadm
```

6. 配置用户密码ssh认证

```
su - cephadm
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
ssh-copy-id -i ~/.ssh/id_rsa.pub cephadm@localhost
for i in ceph-1 ceph-2;do
  scp -rp ~/.ssh/ cephadm@${i}:~
done
```

四、部署RADOS存储集群

1. 在管理节点安装ceph-deploy

这里我们用ceph-0主机作为管理节点

```
# su - cephadm
$ sudo yum -y install ceph-deploy python-setuptools python2-subprocess32
```

2. 初始化RADOS集群

2.1. 在管理节点以cephadm用户创建集群相关的配置文件目录

```
# su - cephadm
$ mkdir ceph-cluster && cd ceph-cluster
```

2.2. 初始化第一个mon节点，准备创建集群

```
$ ceph-deploy new --cluster-network=100.65.20.0/24 --public-
network=100.65.20.0/24 ceph-0
```

2.3. 安装ceph集群：

```
$ sudo yum -y install ceph ceph-radosgw  python-enum34
```

建议在每台服务器上手动执行，因如下命令不会自动安装python-enum34包，经测试python-enum34是必要依赖包，不然后期可能会出现Module 'volumes' has failed dependency: No module named enum报错。

```
$ ceph-deploy install --no-adjust-repos ceph-0 ceph-1 ceph-2
```

注释：--no-adjust-repos 直接使用本地源，不生成官方源

2.4. 初始化mon节点

```
$ ceph-deploy mon create-initial
$ ceph-deploy --overwrite-conf mon create-initial  # 如有手动修改ceph.conf配置文件，请执行此命令
```

2.5. 把配置文件和admin秘钥拷贝到ceph集群各个节点

```
$ ceph-deploy admin ceph-0 ceph-1 ceph-2
$ sudo setfacl -m u:cephadm:r /etc/ceph/ceph.client.admin.keyring  # 在3台服务器都要执行
```

2.6. 配置manager节点，启动ceph-mgr进程

```
$ ceph-deploy mgr create ceph-0
```

2.7. 查看集群状态

```
$ ceph -s
$ ceph health
HEALTH_WARN OSD count 0 < osd_pool_default_size 3
```

如没有其他报错，表示集群安装完成。

五、向RADOS集群添加OSD

1. 列举磁盘

命令格式： ceph-deploy disk list {node-name [node-name]...}

```
$ ceph-deploy disk list ceph-0 ceph-1 ceph-2
```

2. 擦净磁盘（删除分区表）

命令格式：ceph-deploy disk zap {osd-server-name}:{disk-name}

```
$ ceph-deploy disk zap ceph-0 /dev/sda
$ ceph-deploy disk zap ceph-1 /dev/sda
$ ceph-deploy disk zap cehp-2 /dev/sda
```

注释：因我们的环境系统盘安装在sdy/sdz，所以第一块是从sda开始，可以用系统命令lsblk来查看服务器有多少块硬盘

3. 添加OSD

早期ceph-deploy 命令支持在将添加OSD的过程分为两个步骤：准备OSD和激活OSD，在新版本中，此种操作方式已经被废除，添加OSD的步骤只能由命令："ceph-deploy osd create {node} --data {data-disk}" 一次完成，默认使用的存储引擎bluestore

```
$ ceph-deploy osd create ceph-0 --data /dev/sda
$ ceph-deploy osd create ceph-1 --data /dev/sda
$ ceph-deploy osd create ceph-2 --data /dev/sda
```

4. 列出指定节点上的OSD

```
$ ceph-deploy osd list ceph-0 ceph-1 ceph-2
```

5. 查看OSD的相关信息

```
ceph osd stat
ceph osd ls
ceph osd tree
```

6. 添加其余OSD

把sdb--sdx其余23块盘添加到ceph

```
$ cat <<'EOF' > ceph_add_osd.sh
#!/bin/bash
for i in {b..x}
do
  for node in ceph-0 ceph-1 ceph-2
  do
    ceph-deploy disk zap ${node} /dev/sd${i}
    ceph-deploy osd create ${node} --data /dev/sd${i}
  done
done
EOF

$ bash ceph_add_osd.sh
```

六、测试上传下载数据对象

存储数据时，客户端必须首先连接至RADOS集群上某存储磁，而后根据对象名称由相关的CRUSH规则完成数据对象寻址。

1. 创建数据pool

```
$ ceph osd pool create mypool 16
```

2. 上传数据

```
$ echo "TEST" > test.log
$ rados put test.log test.log --pool=mypool
```

3. 列出数据

```
$ rados ls --pool mypool
```

4. 获取存储池中数据对象的具体位置信息

```
$ ceph osd map mypool test.log
osdmap e20 pool 'mypool' (1) object 'a.jpg' -> pg 1.942cc0fc (1.c) -> up
([2,0,1], p2) acting ([2,0,1], p2)
```

5. 删除数据对象

```
$ rados rm test.log --pool=mypool
或者：
$ rados rm test.log  -p mypool
```

6. 删除存储池

```
$ ceph osd pool ls
$ ceph config set mon mon_allow_pool_delete true
$ ceph osd pool rm mypool mypool --yes-i-really-really-mean-it
$ ceph config set mon mon_allow_pool_delete false
```

七、扩展ceph集群

1. 扩展mon监视器节点

```
$ ceph-deploy mon add ceph-1
$ ceph-deploy mon add ceph-2
```

2. 添加mgr节点

```
$ ceph-deploy mgr create ceph-1 ceph-2
```

八、开启MGR模块

1. 安装dashboard

```
$ sudo yum install ceph-mgr-dashboard
```

2. 开启mgr功能

```
$ ceph mgr module enable dashboard
```

3. 查看mgr开启的功能

```
$ ceph mgr module ls
```

4. web登录配置

默认情况下，仪表板的所有HTTP连接均使用SSL/TLS进行保护。

要快速启动并运行仪表板，可以使用以下内置命令生成并安装自签名证书:

```
$ ceph dashboard create-self-signed-cert
```

5. 创建具有管理员角色的用户

```
$ ceph dashboard set-login-credentials admin admin
```

6. 查看ceph-mgr服务

```
$ ceph mgr services
{
    "dashboard": "https://ceph-0:8443/"
}
```

以上配置完成后，浏览器输入https://ceph-0:8443输入用户名admin，密码admin登录即可查看

7. 扩展，修改默认配置

指定集群dashboard的访问端口和IP

```
ceph config-key set mgr/dashboard/server_port 8443
ceph config-key set mgr/dashboard/server_addr $IP
```

九、PG规置组计算方法

```
                  (OSDs * 100)

Total PGs =  ----------------------

                   pool size #副本数量
```

1. PG和PGP查看

```
ceph osd pool get {pool-name} pg_num
ceph osd pool get {pool-name} pgp_num
```

2. PG和PGP修改方法

```
ceph osd pool set {pool-name} pg_num {pg_num}
ceph osd pool set {pool-name} pgp_num {pgp_num}
```

十、创建RBD pool测试

1. 创建rbd的pool池

```
$ ceph osd pool create rbd-pool 1024
```

2. 创建10G大小块设备

```
$ rbd create image01 --size 10240 --pool rbd
```

3. ceph存储配置（映射RBD镜像到客户端）：

3.1. 安装ceph-common

客户端上操作:

```
yum -y install ceph-common
modprobe nbd   # 检查是否成功加载
```

3.2. 客户端授权

ceph 服务器上操作:

```
ceph auth get-or-create client.clt3422 mon 'allow r' osd 'allow class-read
object_prefix rbd_children,allow rwx pool=rbd-pool'
```

3.3. 拷贝 client.clt3422到客户端/etc/ceph目录下

```
cd ceph-cluster/
ceph auth get-or-create client.clt3422 | tee ceph.client.clt3422.keyring
scp ceph.client.clt3422.keyring ceph.conf root@100.65.34.22:/etc/ceph
```

4. 挂载RBD设备块到客户端

客户端上操作:

4.1. 查看设备块信息

```
# rbd --image rbd-pool/image01 info --name client.clt3422
rbd image 'image01':
    size 10 GiB in 2560 objects
    order 22 (4 MiB objects)
    snapshot_count: 0
    id: 85e03d414bda
    block_name_prefix: rbd_data.85e03d414bda
    format: 2
    features: layering
    op_features:
    flags:
    create_timestamp: Wed Sep 22 15:25:12 2021
    access_timestamp: Wed Sep 22 15:25:12 2021
    modify_timestamp: Wed Sep 22 15:25:12 2021

# ceph -s --name client.clt3422
  cluster:
    id:     d0e1a433-db32-4c67-8257-fb19744da26a
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum ceph-0,ceph-1,ceph-2 (age 2h)
    mgr: ceph-0(active, since 4h), standbys: ceph-1, ceph-2
    osd: 72 osds: 72 up (since 4h), 72 in (since 4h)

  task status:

  data:
    pools:   2 pools, 4096 pgs
    objects: 4 objects, 35 B
    usage:   87 GiB used, 1.0 PiB / 1.0 PiB avail
    pgs:     4096 active+clean
```

4.2. map设备块到客户端

```
# rbd map rbd-pool/image01 --name client.clt3422
/dev/rbd0

# rbd showmapped --name client.clt3422
id pool     namespace image   snap device
0  rbd-pool           image01 -    /dev/rbd0
```

4.3. 格式设备块

```
# 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
```

4.4 挂载设备块

```
# mkdir /rbd_data
# mount /dev/rbd0 /rbd_data
# df -h /rbd_data
Filesystem      Size  Used Avail Use% Mounted on
/dev/rbd0        10G   33M   10G   1% /rbd_data
```

4.5 测试数据写入

```
# echo test > /rbd_data/test.txt
# cat /rbd_data/test.txt
test
```

5. 取消map

```
umount /rbd_data
rbd unmap rbd-pool/image01 --name client.clt3422
```

6. 如何查看rbd映射的设备被哪个客户端使用

```
$ rbd info rbd-pool/image01
rbd image 'image01':
    size 10 GiB in 2560 objects
    order 22 (4 MiB objects)
    snapshot_count: 0
    id: 85e03d414bda
    block_name_prefix: rbd_data.85e03d414bda
    format: 2
    features: layering
    op_features:
    flags:
```

```
    create_timestamp: Wed Sep 22 15:25:12 2021
    access_timestamp: Wed Sep 22 15:25:12 2021
    modify_timestamp: Wed Sep 22 15:25:12 2021

$ rados -p rbd-pool listwatchers rbd_header.85e03d414bda
watcher=100.65.34.22:0/2652720761 client.44410 cookie=18446462598732840961
```