

# 环境要求

## 虚拟机配置

基础配置					
计费模式	按需计费	区域	上海一	可用区	可用区1
规格	通用计算增强型   c7.large.2   2vCPUs   4GiB	镜像	CentOS 7.9 64bit	主机安全	免费开启主机安全基础防护
系统盘	通用型SSD, 40GiB, SCSI	数据盘	1块   通用型SSD, 30GiB, SCSI		
网络配置					
虚拟私有云	vpc-default(192.168.0.0/16)	安全组	Sys-WebServer	主网卡	subnet-default(192.168.0.0/24)
弹性公网IP	全动态BGP   计费方式: 按流量计费   带宽: 10 ...				
高级配置					
云服务器名称	ceph1	登录凭证	密码	云监控服务	详细监控
云服务器组	--				

## 节点对应IP

<input type="checkbox"/>	名称/ID	监控	可用区	状态	规格/镜像	IP地址	计费模式	标签	操作
<input type="checkbox"/>	ceph2 1008b4e0-de39-4c90-9...		可用区1	运行中	2vCPUs   4GiB CentOS 7.9 64bit	123.60.52.131 (弹性公网) 1... 192.168.0.195 (私有)	按需计费 2022/10/06 14:1...	--	远程登录
<input type="checkbox"/>	ceph3 c003f648-6a06-4c7a-9...		可用区1	运行中	2vCPUs   4GiB CentOS 7.9 64bit	124.70.155.61 (弹性公网) 1... 192.168.0.128 (私有)	按需计费 2022/10/06 14:1...	--	远程登录
<input type="checkbox"/>	ceph1 4dd0bb9-4c14-4fea-89...		可用区1	运行中	2vCPUs   4GiB CentOS 7.9 64bit	123.60.70.37 (弹性公网) 10... 192.168.0.19 (私有)	按需计费 2022/10/06 14:1...	--	远程登录

# 配置部署环境

## 1、配置主机名

### a.配置永久静态主机名

主机配置为ceph1~ceph3

```
hostnamectl --static set-hostname ceph1
```

```
[root@ceph1-0001 ~]# hostnamectl --static set-hostname ceph1
[root@ceph1-0001 ~]#
```

其余节点以此类推。

重启可以看到已经生效

```
[root@ceph1 ~]#
```

其余节点以此类推。

### b.修改域名解析文件

```
vi /etc/hosts
```

在各个节点的“/etc/hosts”中添加如下内容，添加主机名和内网IP的对应关系

```
192.168.0.19    ceph1
192.168.0.195  ceph2
192.168.0.128  ceph3
```

```
1 ceph1 x 2 ceph2 x 3 ceph3 x +
[~]:1 localhost localhost.localdomain localhost6 localhost6.localdom
ain6
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.
localdomain4
127.0.0.1 ceph1-0001 ceph1-0001
192.168.0.19 ceph1
192.168.0.195 ceph2
192.168.0.128 ceph3
```

## 2、配置免密登录

在ceph1节点生成公钥，并发放到各个主机节点

```
ssh-keygen -t rsa
for i in {1..3}; do ssh-copy-id ceph$i; done
```

注意：输入第一条命令“ssh-keygen -t rsa”之后，按回车使用默认配置

```
[root@ceph1 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
```

输入各节点登录密码，将节点添加进来

```
[root@ceph1 ~]# for i in {1..3}; do ssh-copy-id ceph$i; done
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pu
b"
The authenticity of host 'ceph1 (192.168.0.19)' can't be established.
ECDSA key fingerprint is SHA256:bfGwx5LU0fd9L8ccZ1NNQFcG8DR5szsK/QQb8fjvwf4.
ECDSA key fingerprint is MD5:fa:8e:8b:a7:cd:05:24:7d:f9:d4:0b:56:7c:ba:0a:e2.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
root@ceph1's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'ceph1'"
and check to make sure that only the key(s) you wanted were added.
```

可进行测试，显示已成功

```
ssh 'ceph3'
```

```
[root@ceph1 ~]# ssh 'ceph3'
Last login: Fri Oct 7 15:43:35 2022 from 202.119.46.83

Welcome to Huawei Cloud Service
```

注意：测试完成记得输入exit退出！

### 3、关闭防火墙

需在所有Ceph节点依次执行如下命令

```
systemctl stop firewalld
systemctl disable firewalld
systemctl status firewalld
```

```
[root@ceph3 ~]# systemctl stop firewalld
[root@ceph3 ~]# systemctl disable firewalld
[root@ceph3 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
```

### 4、关闭SELinux

需在所有Ceph节点依次执行如下命令

```
vi /etc/selinux/config
```

修改SELINUX=disabled

### 5、配置NTP

Ceph中会自动校验存储节点之间的时间，若不同节点之间时差较大，会有告警，因此执行如下步骤  
每个节点安装ntp

```
yum -y install ntp
```

使ceph2、ceph3节点同步ceph1的时间，在ceph2、ceph3节点执行如下操作

```
vi /etc/ntp.conf
```

注释时间服务器，并添加ceph1节点作为时间服务器

```
server ceph1 iburst
```

```
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_moni

driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict ::1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
# server 0.centos.pool.ntp.org iburst
# server 1.centos.pool.ntp.org iburst
# server 2.centos.pool.ntp.org iburst
# server 3.centos.pool.ntp.org iburst
server ceph1 iburst
#broadcast 192.168.1.255 autokey          # broadcast server
-- INSERT --
```

每个节点开启ntpd，并查看状态

```
systemctl start ntpd
systemctl status ntpd
```

```
[root@ceph1 ~]# systemctl start ntpd
[root@ceph1 ~]# systemctl status ntpd
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-10-07 17:33:15 CST; 3s ago
     Process: 13892 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 13893 (ntpd)
      CGroup: /system.slice/ntpd.service
              └─13893 /usr/sbin/ntpd -u ntp:ntp -g
```

在ceph2、ceph3节点确认ntp服务器指向ceph1节点

```
ntpq -pn
```

```
[root@ceph3 ~]# ntpq -pn
      remote           refid      st t when poll reach   delay   offset  jitter
=====
*192.168.0.19         84.16.73.33    2 u   7   64    1    0.192   -0.074    0.381
```

输入date，可以看到各个节点时间已经同步

```
[root@ceph1 ~]# date
Fri Oct 7 17:38:48 CST 2022
```

```
[root@ceph2 ~]# date
Fri Oct 7 17:38:48 CST 2022
```

```
[root@ceph3 ~]# date
Fri Oct 7 17:38:48 CST 2022
```

## 6、配置Ceph镜像源

## a.在所有节点建立ceph.repo

```
vi /etc/yum.repos.d/ceph.repo
```

加入以下内容

```
[Ceph]
name=Ceph packages for $basearch
baseurl=http://download.ceph.com/rpm-nautilus/el7/$basearch
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://download.ceph.com/keys/release.asc
priority=1

[Ceph-noarch]
name=Ceph noarch packages
baseurl=http://download.ceph.com/rpm-nautilus/el7/noarch
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://download.ceph.com/keys/release.asc
priority=1

[ceph-source]
name=Ceph source packages
baseurl=http://download.ceph.com/rpm-nautilus/el7/SRPMS
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://download.ceph.com/keys/release.asc
priority=1
```

## b.更新yum源

```
yum clean all && yum makecache
```

```
[root@ceph3 ~]# yum clean all && yum makecache
Loaded plugins: fastestmirror
Cleaning repos: Ceph Ceph-noarch base ceph-source epel extras updates
Cleaning up list of fastest mirrors
Loaded plugins: fastestmirror
Determining fastest mirrors
Ceph | 1.5 kB | 00:00
Ceph-noarch | 1.5 kB | 00:00
base | 3.6 kB | 00:00
ceph-source | 1.5 kB | 00:00
epel | 4.7 kB | 00:00
extras | 2.9 kB | 00:00
updates | 2.9 kB | 00:00
```

# 安装Ceph

## 1、安装Ceph软件

### a.在所有集群和客户端节点安装Ceph

```
yum -y install librados2-14.2.10 ceph-14.2.10
```

### b.在ceph1节点额外安装ceph-deploy

```
yum -y install ceph-deploy
```

```
Running transaction
  Installing : ceph-deploy-2.0.1-0.noarch                1/1
  Verifying  : ceph-deploy-2.0.1-0.noarch                1/1

Installed:
  ceph-deploy.noarch 0:2.0.1-0

Complete!
```

### c.在各节点查看版本

```
ceph -v
```

```
[root@ceph2 ~]# ceph -v
ceph version 14.2.10 (b340acf629a010a74d90da5782a2c5fe0b54ac20) nautilus (stable)
```

## 2、部署MON节点

注意：只需要在主节点ceph1执行

### a.创建集群

```
cd /etc/ceph
ceph-deploy new ceph1 ceph2 ceph3
```

### b.修改ceph.conf文件

```
vi /etc/ceph/ceph.conf
```

添加内容如下

```
[mon]
mon_allow_pool_delete = true
```

由于本次只涉及到存储之间的数据交互，所以无需配置public network、cluster network，采用网络复用即可

### c.初始化并收集密钥

```
ceph-deploy mon create-initial
```

### d.将“ceph.client.admin.keyring”拷贝到各个节点上

```
ceph-deploy --overwrite-conf admin ceph1 ceph2 ceph3
```

#### e.查看是否配置成功

查看是否配置成功

```
ceph -s
```

```
[root@ceph1 ceph]# ceph -s
cluster:
  id:      c35f6929-f38c-4e28-8dc3-307818a8190b
  health:  HEALTH_OK

services:
  mon: 3 daemons, quorum ceph1,ceph3,ceph2 (age 115s)
  mgr: no daemons active
  osd: 0 osds: 0 up, 0 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage:   0 B used, 0 B / 0 B avail
  pgs:
```

如果遇到以下问题，是时间同步出现偏差

```
[root@ceph1 ceph]# ceph -s
cluster:
  id:      8c580d56-43cb-4cef-bf3c-c93b88180ed7
  health:  HEALTH_WARN
           clock skew detected on mon.ceph2, mon.ceph3

services:
  mon: 3 daemons, quorum ceph1,ceph2,ceph3 (age 3m)
  mgr: ceph1(active, since 3m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 2m), 3 in (since 2m)

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage:   3.0 GiB used, 87 GiB / 90 GiB avail
  pgs:
```

修改ceph.conf配置文件并重启即可

```
vi ceph.conf
```

在global字段添加如下内容

```
mon clock drift allowed = 2
mon clock drift warn backoff = 30
```

```
[global]
fsid = 8c580d56-43cb-4cef-bf3c-c93b88180ed7
mon_initial_members = ceph1, ceph2, ceph3
mon_host = 192.168.0.27,192.168.0.45,192.168.0.174
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx
mon clock drift allowed = 2
mon clock drift warn backoff = 30

[mon]
mon_allow_pool_delete = true
```

将配置文件同步给其他节点

```
ceph-deploy --overwrite-conf config push ceph{1,2,3}
```

重启mon节点

```
systemctl restart ceph-mon.target
```

再次查看集群情况即为成功

```
[root@ceph1 ceph]# systemctl restart ceph-mon.target
[root@ceph1 ceph]# ceph -s
  cluster:
    id:      8c580d56-43cb-4cef-bf3c-c93b88180ed7
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum ceph1,ceph2,ceph3 (age 3s)
    mgr: ceph1(active, since 6m), standbys: ceph2, ceph3
    osd: 3 osds: 3 up (since 6m), 3 in (since 6m)

  data:
    pools:   0 pools, 0 pgs
    objects: 0 objects, 0 B
    usage:    3.0 GiB used, 87 GiB / 90 GiB avail
    pgs:
```

### 3、部署MGR节点

注意：只需要在主节点ceph1执行

#### a.部署MGR节点

```
ceph-deploy mgr create ceph1 ceph2 ceph3
```

#### b.查看MGR是否部署成功

```
ceph -s
```



```
[root@ceph1 ceph]# ceph -s
cluster:
  id:      c35f6929-f38c-4e28-8dc3-307818a8190b
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum ceph1,ceph3,ceph2 (age 4m)
  mgr: ceph1(active, since 16s), standbys: ceph2, ceph3
  osd: 0 osds: 0 up, 0 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage:   0 B used, 0 B / 0 B avail
  pgs:
```

## 4、部署OSD节点

### a.通过命令查看磁盘情况

```
lsblk
```

```
[root@ceph1 ceph]# lsblk
NAME        MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
sda          8:0    0   40G  0  disk
└─sda1       8:1    0   40G  0  part /
sdb          8:16   0   30G  0  disk
```

sda为系统盘不能使用，我们选用sdb作为数据盘

### b.将/dev/sdb磁盘加入集群

```
ceph-deploy osd create ceph1 --data /dev/sdb
ceph-deploy osd create ceph2 --data /dev/sdb
ceph-deploy osd create ceph3 --data /dev/sdb
```

### c.查看集群osd状态

```
ceph -s
```

```
[root@ceph1 ceph]# ceph -s
cluster:
  id: c35f6929-f38c-4e28-8dc3-307818a8190b
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum ceph1,ceph3,ceph2 (age 22m)
  mgr: ceph1(active, since 18m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 39s), 3 in (since 39s)

data:
  pools: 0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage: 3.0 GiB used, 87 GiB / 90 GiB avail
  pgs:
```

至此，ceph集群的部署已经完成

## 验证对象存储

### 1、部署RGW节点

#### a.在所有服务端节点安装RGW组件

```
yum -y install ceph-radosgw-14.2.10
```

#### b.创建RGW实例

注意：只需要在主节点ceph1执行

```
ceph-deploy rgw create ceph1 ceph2 ceph3
```

```
adosgw@rgw.ceph2.service to /usr/lib/systemd/system/ceph-radosgw@.service.
[ceph2][INFO ] Running command: systemctl start ceph-radosgw@rgw.ceph2
[ceph2][INFO ] Running command: systemctl enable ceph.target
[ceph_deploy.rgw][INFO ] The Ceph Object Gateway (RGW) is now running on host ceph2 and
default port 7480
[ceph3][DEBUG ] connected to host: ceph3
[ceph3][DEBUG ] detect platform information from remote host
[ceph3][DEBUG ] detect machine type
[ceph_deploy.rgw][INFO ] Distro info: CentOS Linux 7.9.2009 Core
[ceph_deploy.rgw][DEBUG ] remote host will use systemd
[ceph_deploy.rgw][DEBUG ] deploying rgw bootstrap to ceph3
[ceph3][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph3][WARNIN] rgw keyring does not exist yet, creating one
[ceph3][DEBUG ] create a keyring file
[ceph3][DEBUG ] create path recursively if it doesn't exist
[ceph3][INFO ] Running command: ceph --cluster ceph --name client.bootstrap-rgw --keyrin
g /var/lib/ceph/bootstrap-rgw/ceph.keyring auth get-or-create client.rgw.ceph3 osd allow
rwx mon allow rw -o /var/lib/ceph/radosgw/ceph-rgw.ceph3/keyring
[ceph3][INFO ] Running command: systemctl enable ceph-radosgw@rgw.ceph3
[ceph3][WARNIN] Created symlink from /etc/systemd/system/ceph-radosgw.target.wants/ceph-r
adosgw@rgw.ceph3.service to /usr/lib/systemd/system/ceph-radosgw@.service.
[ceph3][INFO ] Running command: systemctl start ceph-radosgw@rgw.ceph3
[ceph3][INFO ] Running command: systemctl enable ceph.target
[ceph_deploy.rgw][INFO ] The Ceph Object Gateway (RGW) is now running on host ceph3 and
default port 7480
[root@ceph1 ceph]#
```

可以看到默认的端口号为7480

### c.查看集群状态

```
ceph -s
```

```
[root@ceph1 ceph]# ceph -s
cluster:
  id:      c35f6929-f38c-4e28-8dc3-307818a8190b
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum ceph1,ceph3,ceph2 (age 42m)
  mgr: ceph1(active, since 38m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 21m), 3 in (since 21m)
  rgw: 3 daemons active (ceph1, ceph2, ceph3)

task status:

data:
  pools:   4 pools, 128 pgs
  objects: 191 objects, 3.5 KiB
  usage:   3.0 GiB used, 87 GiB / 90 GiB avail
  pgs:     128 active+clean
```

## 2、创建存储池

### a.创建存储池

名称为pooltest

```
ceph osd pool create pooltest 32 32
```

其中两个64分别对应pg\_num和pgp\_num，即存储池对应的pg数量。

Ceph官方文档建议整个集群所有存储池的pg数量之和大约为：(OSD数量 \* 100)/数据冗余因数，数据冗余因数对副本模式而言是副本数，创建存储池默认是副本模式，副本数为3

```
[root@ceph1 ceph]# ceph osd pool create pooltest 32 32
pool 'pooltest' created
```

通过命令查看现有存储池

```
ceph osd pool ls
```

```
[root@ceph1 ceph]# ceph osd pool ls
.rgw.root
default.rgw.control
default.rgw.meta
default.rgw.log
pooltest
[root@ceph1 ceph]# ceph -s
```

同时，通过ceph -s命令也可看到存储池概况

```
[root@ceph1 ceph]# ceph -s
cluster:
  id:      7951063c-d0c4-4782-8e3f-43411f9b01e1
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum ceph2,ceph1,ceph3 (age 4m)
  mgr: ceph1(active, since 4m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 3m), 3 in (since 3m)
  rgw: 3 daemons active (ceph1, ceph2, ceph3)

task status:

data:
  pools:   5 pools, 192 pgs
  objects: 189 objects, 2.3 KiB
  usage:   3.0 GiB used, 87 GiB / 90 GiB avail
  pgs:    192 active+clean
```

## b.存储池的使用

查看当前pooltest存储池的情况

```
rados -p pooltest ls
```

```
[root@ceph1 ceph]# rados -p pooltest ls
[root@ceph1 ceph]#
```

创建文件，并使用put上传到存储池

```
echo "hello world" > test.txt
rados -p pooltest put test ./test.txt
```

```
[root@ceph1 ceph]# echo "hello world" > test.txt
[root@ceph1 ceph]# cat test.txt
hello world
```

```
[root@ceph1 ceph]# ll
total 132
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-mds.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-mgr.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-osd.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-rgw.keyring
-rw----- 1 root root 151 Oct 8 11:10 ceph.client.admin.keyring
-rw-r--r-- 1 root root 272 Oct 8 11:50 ceph.conf
-rw-r--r-- 1 root root 86035 Oct 8 11:50 ceph-deploy-ceph.log
-rw----- 1 root root 73 Oct 8 10:58 ceph.mon.keyring
-rw-r--r-- 1 root root 155 Oct 8 12:47 pool
-rw-r--r-- 1 root root 92 Jul 10 2020 rbdmap
-rw-r--r-- 1 root root 12 Oct 8 13:48 test.txt
```

上传完成后，再次查看pooltest存储池的情况

```
[root@ceph1 ceph]# rados -p pooltest put test ./test.txt
[root@ceph1 ceph]# rados -p pooltest ls
test
```

并且通过ceph -s可以看到已经存储了一个对象，但是出现警告

```
[root@ceph1 ceph]# ceph -s
cluster:
  id:      7951063c-d0c4-4782-8e3f-43411f9b01e1
  health: HEALTH_WARN
          application not enabled on 1 pool(s)

services:
  mon: 3 daemons, quorum ceph2,ceph1,ceph3 (age 6m)
  mgr: ceph1(active, since 6m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 5m), 3 in (since 5m)
  rgw: 3 daemons active (ceph1, ceph2, ceph3)

task status:

data:
  pools: 5 pools, 192 pgs
  objects: 190 objects, 2.4 KiB
  usage: 3.0 GiB used, 87 GiB / 90 GiB avail
  pgs: 192 active+clean
```

对新加入的存储池进行标记

```
ceph osd pool application enable pooltest rgw
```

再次用ceph -s查看集群情况，已经转为HEALTH

```
[root@ceph1 ceph]# ceph osd pool application enable pooltest rgw
enabled application 'rgw' on pool 'pooltest'
[root@ceph1 ceph]# ceph -s
cluster:
  id:      7951063c-d0c4-4782-8e3f-43411f9b01e1
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum ceph2,ceph1,ceph3 (age 10m)
  mgr: ceph1(active, since 10m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 8m), 3 in (since 8m)
  rgw: 3 daemons active (ceph1, ceph2, ceph3)

task status:

data:
  pools: 5 pools, 192 pgs
  objects: 190 objects, 2.4 KiB
  usage: 3.0 GiB used, 87 GiB / 90 GiB avail
  pgs: 192 active+clean
```

验证存储池下载文件

首先删除文件中的test.txt文件

```
rm -f test.txt
```

```
[root@ceph1 ceph]# rm -f test.txt
[root@ceph1 ceph]# ll
total 128
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-mds.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-mgr.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-osd.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-rgw.keyring
-rw----- 1 root root 151 Oct 8 11:10 ceph.client.admin.keyring
-rw-r--r-- 1 root root 272 Oct 8 11:50 ceph.conf
-rw-r--r-- 1 root root 86035 Oct 8 11:50 ceph-deploy-ceph.log
-rw----- 1 root root 73 Oct 8 10:58 ceph.mon.keyring
-rw-r--r-- 1 root root 155 Oct 8 12:47 pool
-rw-r--r-- 1 root root 92 Jul 10 2020 rbdmap
```

使用get从存储池中下载

```
rados -p pooltest get test ./test.txt
```

```
[root@ceph1 ceph]# rados -p pooltest get test ./test.txt
[root@ceph1 ceph]# ll
total 132
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-mds.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-mgr.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-osd.keyring
-rw----- 1 root root 113 Oct 8 11:09 ceph.bootstrap-rgw.keyring
-rw----- 1 root root 151 Oct 8 11:10 ceph.client.admin.keyring
-rw-r--r-- 1 root root 272 Oct 8 11:50 ceph.conf
-rw-r--r-- 1 root root 86035 Oct 8 11:50 ceph-deploy-ceph.log
-rw----- 1 root root 73 Oct 8 10:58 ceph.mon.keyring
-rw-r--r-- 1 root root 155 Oct 8 12:47 pool
-rw-r--r-- 1 root root 92 Jul 10 2020 rbdmap
-rw-r--r-- 1 root root 12 Oct 8 14:01 test.txt
```

### 3、验证S3接口

#### a.在ceph1节点上创建rgw用户

用户名为admin并且指定access\_key和secret\_key字段，不指定就按照rgw默认分配

```
radosgw-admin user create --uid=admin --display-name=admin --access-key=test1 --secret-key=test1
```

查询账户信息

```
radosgw-admin user info --uid=admin
```

```
[root@ceph1 ceph]# radosgw-admin user info --uid=admin
{
  "user_id": "admin",
  "display_name": "admin",
  "email": "",
  "suspended": 0,
  "max_buckets": 1000,
  "subusers": [],
  "keys": [
    {
      "user": "admin",
      "access_key": "test1",
      "secret_key": "test1"
    }
  ],
  "swift_keys": [],
}
```

#### b.在ceph1节点上安装s3cmd

```
yum -y install s3cmd
```

#### c.更新S3配置文件

在ceph1节点上编辑“~/.s3cfg”

```
vi ~/.s3cfg
```

并添加如下内容

```
[default]
access_key = test1
secret_key = test1
host_base = ceph1_IP:7480
host_bucket = ceph1_IP:7480
use_https = False
```

```
[default]
access_key = test1
secret_key = test1
host_base = 192.168.0.19:7480
host_bucket = 192.168.0.19:7480
use_https = False
```

#### d.在ceph1节点上创建bucket

```
s3cmd mb s3://testbucket
```

查看当前接口中桶的情况

```
s3cmd ls
```

```
[root@ceph1 ceph]# s3cmd ls
2022-10-10 13:57 s3://testbucket
```



### e.进行文件的上传和下载

将本目录下test.txt文件上传至testbucket

```
s3cmd put test.txt s3://testbucket
```

```
[root@ceph1 ceph]# s3cmd put test.txt s3://testbucket
upload: 'test.txt' -> 's3://testbucket/test.txt' [1 of 1]
 12 of 12  100% in    1s    7.05 B/s  done
[root@ceph1 ceph]#
```

可以查看桶下的存储情况

```
s3cmd ls s3://testbucket
```

```
[root@ceph1 ceph]# s3cmd ls s3://testbucket
2022-10-10 11:38          12  s3://testbucket/test.txt
```

将文件下载到指定目录

```
s3cmd get s3://testbucket/test.txt /usr/local
```

```
[root@ceph1 ceph]# s3cmd get s3://testbucket/test.txt /usr/local
download: 's3://testbucket/test.txt' -> '/usr/local/test.txt' [1 of 1]
 12 of 12  100% in    0s   292.61 B/s  done
```

可以在该目录下看到已经下载的文件

```
[root@ceph1 ceph]# ll /usr/local/
total 52
drwxr-xr-x.  2 root root 4096 Feb 10  2022 bin
drwxr-xr-x.  2 root root 4096 Apr 11  2018 etc
drwxr-xr-x.  2 root root 4096 Apr 11  2018 games
drwxr-xr-x. 10 root root 4096 Oct  6 14:20 hostguard
drwxr-xr-x.  2 root root 4096 Apr 11  2018 include
drwxr-xr-x.  3 root root 4096 Feb 10  2022 lib
drwxr-xr-x.  3 root root 4096 Feb 10  2022 lib64
drwxr-xr-x.  2 root root 4096 Apr 11  2018 libexec
drwxr-xr-x.  2 root root 4096 Apr 11  2018 sbin
drwxr-xr-x.  5 root root 4096 Feb 10  2022 share
drwxr-xr-x.  2 root root 4096 Apr 11  2018 src
-rw-r--r--.  1 root root  12 Oct 10 11:38 test.txt
drwx-----  7 root root 4096 Oct  6 14:21 uniagent
[root@ceph1 ceph]#
```

此时再查看ceph集群情况仍然为HEALTH



```
drwxr-xr-x. 2 root root 4096 Apr 11 2018 src
-rw-r--r-- 1 root root 12 Oct 10 15:29 test.txt
drwx----- 7 root root 4096 Oct 10 22:41 uniagent
[root@ceph1 ceph]# ceph -s
cluster:
  id:      8c580d56-43cb-4cef-bf3c-c93b88180ed7
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum ceph1,ceph2,ceph3 (age 20m)
  mgr: ceph1(active, since 27m), standbys: ceph2, ceph3
  osd: 3 osds: 3 up (since 26m), 3 in (since 26m)
  rgw: 3 daemons active (ceph1, ceph2, ceph3)

task status:

data:
  pools:   7 pools, 224 pgs
  objects: 199 objects, 4.2 KiB
  usage:   3.0 GiB used, 87 GiB / 90 GiB avail
  pgs:     224 active+clean

io:
  client:  255 B/s rd, 0 op/s rd, 0 op/s wr
```

后续与大数据侧连接也可采用S3接口进行

## 进行测试

安装测试所需依赖包

```
yum -y install python-boto
```

创建测试脚本

```
vim s3.py
```

添加如下内容

```
import boto
import boto.s3.connection
access_key = 'test1'
secret_key = 'test1'
conn = boto.connect_s3(
    aws_access_key_id = access_key,
    aws_secret_access_key = secret_key,
    host = 'ceph1', port= 7480,
    is_secure=False,
    calling_format = boto.s3.connection.OrdinaryCallingFormat(),
)
bucket = conn.create_bucket('mybucket')
for bucket in conn.get_all_buckets():
    print "{name}\t{created}".format(
        name = bucket.name,
        created = bucket.creation_date,
```

)

```
import boto
import boto.s3.connection
access_key = 'test1'
secret_key = 'test1'
conn = boto.connect_s3(
    aws_access_key_id = access_key,
    aws_secret_access_key = secret_key,
    host = 'ceph1', port= 7480,
    is_secure=False,
    calling_format = boto.s3.connection.OrdinaryCallingFormat(),
)
bucket = conn.create_bucket('mybucket')
for bucket in conn.get_all_buckets():
    print "{name}\t{created}".format(
        name = bucket.name,
        created = bucket.creation_date,
    )
```

运行脚本

```
python s3.py
```

如图所示即为成功

```
[root@ceph1 ceph]# python s3.py
mybucket      2022-10-08T06:32:59.514Z
```

并且保证ceph集群状态为HEALTH

```
[root@ceph1 ceph]# python s3.py
mybucket      2022-10-10T15:34:11.759Z
testbucket    2022-10-10T15:29:10.265Z
[root@ceph1 ceph]# ceph -s
  cluster:
    id:      8c580d56-43cb-4cef-bf3c-c93b88180ed7
    health:  HEALTH_OK

  services:
    mon: 3 daemons, quorum ceph1,ceph2,ceph3 (age 21m)
    mgr: ceph1(active, since 28m), standbys: ceph2, ceph3
    osd: 3 osds: 3 up (since 27m), 3 in (since 27m)
    rgw: 3 daemons active (ceph1, ceph2, ceph3)

  task status:

  data:
    pools:   7 pools, 224 pgs
    objects: 202 objects, 4.6 KiB
    usage:   3.0 GiB used, 87 GiB / 90 GiB avail
    pgs:     224 active+clean

  io:
    client:  170 B/s wr, 0 op/s rd, 0 op/s wr
```