

上一节课搭建的ceph环境是单机模式，如果需要实现高可用则mon、mgr、mds等节点也需要多个才可以保证高可用。

零声教育 Darren

- 1. 先保证mon集群正常
- 2. 然后保证单个osd是正确启动的，再去启动另外两个osd

0 重点推荐的资料

存储策略指南 [Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

管理指南 [Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

操作指南 [Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

架构指南 [Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

1 安装docker

如果已经安装了不需要重新安装。如果没有安装参考docker课程。

2 部署ceph

部署ceph之前，先学习ceph原理，是否不容易理解不同组件之间的关系。

3 准备开发环境

准备三台服务器（ip主机名要设置好），设置ip主机名方法

主机名	IP	功能
master	192.168.1.33	容器主节点mon, osd, mgr, mds, rgw, dashboard
node1	192.168.1.2	容器子节点mon, osd, mgr, mds, rgw
node2	192.168.1.25	容器子节点mon, osd, mgr, mds, rgw

修改主机名方法

输入以下命令以查看当前主机名：

```
hostname
```

使用root权限修改主机名，输入以下命令（将"new_hostname"替换为您想要设置的新主机名）：

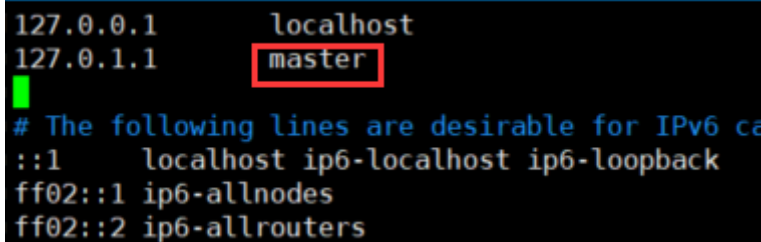
```
sudo hostnamectl set-hostname new_hostname
```

输入系统密码以确认权限。

修改/etc/hosts，有个127.0.0.1或者127.0.1.1

复制后面对应的主机名，修改自己电脑的主机名，比如master

```
sudo vim /etc/hosts
```



```
127.0.0.1    localhost
127.0.1.1    master
# The following lines are desirable for IPv6 capability
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

重新启动系统以使新主机名生效：

```
sudo reboot
```

设置静态IP地址

1. 打开终端并编辑 /etc/network/interfaces 文件，可以使用以下命令打开该文件：

```
sudo vim /etc/network/interfaces
```

1. 在文件的末尾添加以下内容来设置静态IP地址：

```
# 配置 ens33 网络接口
auto ens33
iface ens33 inet static
address 192.168.1.33 # 你想设置的静态IP地址
netmask 255.255.255.0 # 子网掩码
gateway 192.168.1.1 # 网关IP地址,注意自己的网段
dns-nameservers 114.114.114.114 # DNS服务器IP地址，可以根据自己的需要修改
```

注意：

请注意，上述内容是根据默认的ens33网络接口进行设置的。如果您使用的是不同的网络接口，请将上述内容中的ens33替换为您使用的接口名称。

1. 保存文件并退出编辑器。

2. 重启网络服务以应用更改，可以使用以下命令：

```
sudo service networking restart
```

如果不成功则重启系统

```
sudo reboot
```

创建Ceph专用网络

三台机器执行

```
docker network create --driver bridge --subnet 192.168.1.0/16 ceph-network
```

```
docker network inspect ceph-network
```

删除旧的ceph相关容器和清理ceph文件

三台机器执行

删除旧的容器和清理旧的ceph相关目录文件，假如有的话

```
docker rm -f $(docker ps -a | grep ceph | awk '{print $1}')
sudo rm -rf /etc/ceph /var/lib/ceph /var/log/ceph /usr/local/ceph /data/etc/osd
/data/ceph
```

拉取ceph镜像

目前最新的版本是quincy，我们这里使用nautilus版本。官网：

<https://docs.ceph.com/en/latest/releases/nautilus/>

```
docker pull ceph/daemon:latest-nautilus
```

4 在主节点中编写执行脚本

1、Ceph Monitors (MON)

维护集群状态的映射的守护进程。集群映射是五个映射的集合，其中包含关于集群状态及其配置的信息。Ceph必须处理每个集群事件，更新适当的映射，并将更新的映射复制到每个MON守护进程。

2、Ceph Object Storage Devices (OSD)

Ceph存储集群的底层块设备。osd将存储设备(如硬盘或其他块设备)连接到Ceph存储集群。单个存储服务器可以运行多个OSD守护进程，为集群提供多个OSD。

3、Ceph Managers (MGR)

提供一组集群的统计信息。如果集群中没有MGR，客户端的IO操作不会受到影响，但是尝试查询集群统计信息将会失败。红帽建议你每个集群至少部署2个MGR。

4、Ceph Metadata Server (MDS)

代表Ceph文件系统存储元数据。MDS使cephfs能够与Ceph对象存储进行交互，将一个inode映射到一个对象和Ceph在树中存储数据的位置。访问cephfs文件系统的客户端首先向MDS发送一个请求，MDS提供从正确的osd获取文件内容所需的信息。

5、Ceph Object Gateway(RADOS Gateway、RADOSGW或RGW) (RADOS网关)

Ceph Object Gateway(RADOS Gateway、RADOSGW或RGW)是一个用librados构建的对象存储接口。它使用这个库与Ceph集群通信，并直接写入OSD进程。为应用提供基于RESTful API的网关，支持Amazon S3 和 OpenStack Swift两种接口。

mon

vim start_mon.sh

```
#!/bin/bash
docker run -d --net=host \
    --name=mon \
    -v /etc/localtime:/etc/localtime \
    -v /usr/local/ceph/etc:/etc/ceph \
    -v /usr/local/ceph/lib:/var/lib/ceph \
    -v /usr/local/ceph/logs:/var/log/ceph \
    -e MON_IP=192.168.1.33 \
    -e CEPH_PUBLIC_NETWORK=192.168.1.0/16 \
    ceph/daemon:latest-nautilus mon
```

这个脚本是为了启动监视器，监视器的作用是维护整个Ceph集群的全局状态。一个集群至少要有有一个监视器，最好要有奇数个监视器。方便当一个监视器挂了之后可以选举出其他可用的监视器。

osd

vim start_osd.sh

```
#!/bin/bash
docker run -d \
    --name=osd \
    --net=host \
```

```
--restart=always \  
--privileged=true \  
--pid=host \  
-v /etc/localtime:/etc/localtime \  
-v /usr/local/ceph/etc:/etc/ceph \  
-v /usr/local/ceph/lib:/var/lib/ceph \  
-v /usr/local/ceph/logs:/var/log/ceph \  
-v /data/ceph/osd:/var/lib/ceph/osd \  
ceph/daemon:latest-nautilus osd_directory
```

mds

vim start_mds.sh

```
#!/bin/bash  
docker run -d \  
--net=host \  
--name=mds \  
--privileged=true \  
-v /etc/localtime:/etc/localtime \  
-v /usr/local/ceph/etc:/etc/ceph \  
-v /usr/local/ceph/lib:/var/lib/ceph \  
-v /usr/local/ceph/logs:/var/log/ceph \  
-e CEPHFS_CREATE=0 \  
-e CEPHFS_METADATAPOOL_PG=512 \  
-e CEPHFS_DATAPOOL_PG=512 \  
ceph/daemon:latest-nautilus mds
```

mgr

vim start_mgr.sh

```
#!/bin/bash  
docker run -d --net=host \  
--name=mgr \  
-v /etc/localtime:/etc/localtime \  
-v /usr/local/ceph/etc:/etc/ceph \  
-v /usr/local/ceph/lib:/var/lib/ceph \  
-v /usr/local/ceph/logs:/var/log/ceph \  
ceph/daemon:latest-nautilus mgr
```

rgw

vim start_rgw.sh

```
#!/bin/bash
docker run \
  -d --net=host \
  --name=rgw \
  -v /usr/local/ceph/lib:/var/lib/ceph/ \
  -v /usr/local/ceph/etc:/etc/ceph \
  -v /etc/localtime:/etc/localtime \
  ceph/daemon:latest-nautilus rgw
```

5 准备工作

编写hosts文件

三台机器全部都执行一遍

```
cat >>/etc/hosts <<EOF
192.168.1.33 master
192.168.1.2 node1
192.168.1.25 node2
EOF
```

创建ceph目录

三台机器全部都执行一遍

```
sudo mkdir -p /usr/local/ceph/{admin,data,etc,lib,logs}
sudo chmod 777 -R /usr/local/ceph/admin
sudo chmod 777 -R /usr/local/ceph/data
sudo chmod 777 -R /usr/local/ceph/etc
sudo chmod 777 -R /usr/local/ceph/lib
sudo chmod 777 -R /usr/local/ceph/logs
```

该命令会一次创建5个指定的目录，注意逗号分隔，不能有空格，其中：

- admin文件夹下用于存储启动脚本
- data文件夹用于挂载文件
- etc文件夹下存放了ceph.conf等配置文件
- lib文件夹下存放了各组件的库文件
- logs文件夹下存放了ceph的日志文件

创建osd挂载目录

三台机器全部都执行一遍

```
sudo mkdir -p /data/etc/osd
sudo chmod 777 -R /data/etc/osd
```

设置免密登录

master执行即可，设置免密登录

```
ssh-keygen
```

```
ssh-copy-id node1
```

```
ssh-copy-id node2
```

安装ceph-mon

查看容器日志

```
docker logs -f -t --tail=100 a92b36(容器id)
```

在master执行

```
sh start_mon.sh
```

在master执行，查看是否运行成功

```
docker ps
```

在master执行，检查Ceph状态

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

在master执行

```
sudo vim /usr/local/ceph/etc/ceph.conf
```

```
[global]
fsid = b6677a25-699d-4000-96ab-8f34fedafb4f
mon initial members = master
mon host = 192.168.1.33,192.168.1.2,192.168.1.25
public network = 192.168.1.0/16
cluster network = 192.168.1.0/16
osd journal size = 100

# 容忍更多的时钟误差
mon clock drift allowed = 2
mon clock drift warn backoff = 30
mon_max_pg_per_osd = 1000

# 推送到各节点:
```

```
# 允许删除pool
mon allow pool delete = true
osd max object name len = 256
osd max object namespace len = 64

[mgr]
# 开启WEB仪表盘
mgr modules = dashboard
[client.rgw.ceph1]

# 设置rgw网关的web访问端口
rgw_frontends = "civetweb port=7480"
```

在master执行

复制文件到节点，这里最好是用root用户

```
sudo scp -r /usr/local/ceph 你的用户名@node1:/usr/local
sudo scp -r /usr/local/ceph 你的用户名@node2:/usr/local
比如
sudo scp -r /usr/local/ceph lqf@node1:/usr/local
sudo scp -r /usr/local/ceph lqf@node2:/usr/local
```

如果有**权限的问题**可以先拷贝到home目录，比如~/bin，然后再拷贝到/usr/local/

```
sudo scp -r /usr/local/ceph lqf@node1:~/bin
```

node 1执行

```
sudo mkdir /usr/local/ceph
sudo mv ~/bin/* /usr/local/ceph/
```

```
sudo scp -r /usr/local/ceph lqf@node2:~/bin
```

node 2执行

```
sudo mkdir /usr/local/ceph
sudo mv ~/bin/* /usr/local/ceph/
```

拷贝脚本到另外两台机器

```
sudo scp -r ~/ceph_sh lqf@node1:~/
sudo scp -r ~/ceph_sh lqf@node2:~/
```

分别在两台node上执行


```
sh start_mon.sh
```

在任意机器执行将会看到下面的内容

检查Ceph状态

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

cluster:

id: b6677a25-699d-4000-96ab-8f34fedafb4f

health: HEALTH_WARN

mons are allowing insecure global_id reclaim

services:

mon: 3 daemons, quorum master,node1,node2 (age 1m)

安装ceph-osd

分别在三台机器上执行生成osd的密钥信息（三台mon节点都需执）

```
docker exec -it mon ceph auth get client.bootstrap-osd -o /var/lib/ceph/bootstrap-osd/ceph.keyring
```

分别在三台机器上执行

```
sh start_osd.sh
```

在任意机器执行将会看到下面的内容, 检查Ceph状态

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

cluster:

id: b6677a25-699d-4000-96ab-8f34fedafb4f

health: HEALTH_WARN

mons are allowing insecure global_id reclaim

services:

mon: 3 daemons, quorum master,node1,node2 (age 2m)

osd: 3 osds: 3 up (since 2d), 3 in (since 1m)

安装ceph-mgr

分别在三台机器上执行

```
sh start_mgr.sh
```

在任意机器执行将会看到下面的内容, 检查Ceph状态

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

```
cluster:
  id: b6677a25-699d-4000-96ab-8f34fedafb4f
  health: HEALTH_WARN
        mons are allowing insecure global_id reclaim

services:
  mon: 3 daemons, quorum master,node1,node2 (age 3m)
  osd: 3 osds: 3 up (since 2d), 3 in (since 2m)
  mgr: master(active, since 2d), standbys: node1, node2
```

安装ceph-rgw

分别在三台机器上执行生成rgw的密钥信息

```
docker exec mon ceph auth get client.bootstrap-rgw -o /var/lib/ceph/bootstrap-rgw/ceph.keyring
```

分别在三台机器上执行

```
sh start_rgws.sh
```

#单节点时，rgw启动之后，ceph -s 查看可能会出现Degraded降级的情况，我们需要手动设置rgw pool的size 和 min_size为最小1

```
ceph osd pool set .rgw.root min_size 1
ceph set .rgw.root size 1

ceph osd pool set default.rgw.control min_size 1
ceph osd pool set default.rgw.control size 1

ceph osd pool set default.rgw.meta min_size 1
ceph osd pool set default.rgw.meta size 1

ceph osd pool set default.rgw.log min_size 1
ceph osd pool set default.rgw.log size 1
```

在任意机器执行将会看到下面的内容

检查Ceph状态

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

```
cluster:
  id: b6677a25-699d-4000-96ab-8f34fedafb4f
  health: HEALTH_WARN
        mons are allowing insecure global_id reclaim
```

services:

mon: 3 daemons, quorum master,node1,node2 (age 3m)

osd: 3 osds: 3 up (since 2d), 3 in (since 2m)

mgr: master(active, since 2d), standbys: node1, node2

rgw: 3 daemons active (master, node1, node2)

安装ceph-mds

分别在三台机器上执行

```
sh start_mds.sh
```

在任意机器执行将会看到下面的内容

检查Ceph状态

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

cluster:

id: b6677a25-699d-4000-96ab-8f34fedafb4f

health: HEALTH_WARN

mons are allowing insecure global_id reclaim

services:

mon: 3 daemons, quorum master,node1,node2 (age 3m)

osd: 3 osds: 3 up (since 2d), 3 in (since 2m)

mds: cephfs:1 {0=master=up:active} 2 up:standby

mgr: master(active, since 2d), standbys: node1, node2

rgw: 3 daemons active (master, node1, node2)

6 CephFS部署

ceph FS 即 ceph filesystem，可以实现文件系统共享功能,客户端通过 ceph 协议挂载并使用 ceph 集群作为数据存储服务器。

在master执行 创建Data Pool

```
docker exec osd ceph osd pool create cephfs_data 64 64
```

在master执行 创建Metadata Pool

```
docker exec osd ceph osd pool create cephfs_metadata 32 32
```

在master执行 创建CephFS

```
docker exec osd ceph fs new cephfs cephfs_metadata cephfs_data
```

在master执行 查看FS信息

```
docker exec osd ceph fs ls
```

7 安装Dashboard管理后台

在master执行 开启dashboard功能

```
docker exec mgr ceph mgr module enable dashboard
```

在master执行 创建证书

```
docker exec mgr ceph dashboard create-self-signed-cert
```

在master执行 设置用户名为admin, 密码为123456。

```
docker exec mgr ceph dashboard set-login-credentials admin 123456
```

注意我使用这条命令的时候报错了dashboard set-login-credentials : Set the login credentials. Password read from -i

我手动在mgr容器中创建了/tmp/ceph-password.txt 在里面写入了密码123456
然后执行如下命令就成功了：

```
# 进入容器shell命令行
docker exec -it mgr bash
#编译密码文件
vi /tmp/ceph-password.txt
123456
# 退出容器
exit
```

```
docker exec mgr ceph dashboard ac-user-create admin -i /tmp/ceph-password.txt administrator
```

在master执行 配置外部访问端口

```
docker exec mgr ceph config set mgr mgr/dashboard/server_port 18080
```

在master执行 配置外部访问IP

```
docker exec mgr ceph config set mgr mgr/dashboard/server_addr 192.168.1.33
```

在master执行 关闭https(如果没有证书或内网访问，可以关闭)

```
docker exec mgr ceph config set mgr mgr/dashboard/ssl false
```

在master执行 重启Mgr DashBoard服务

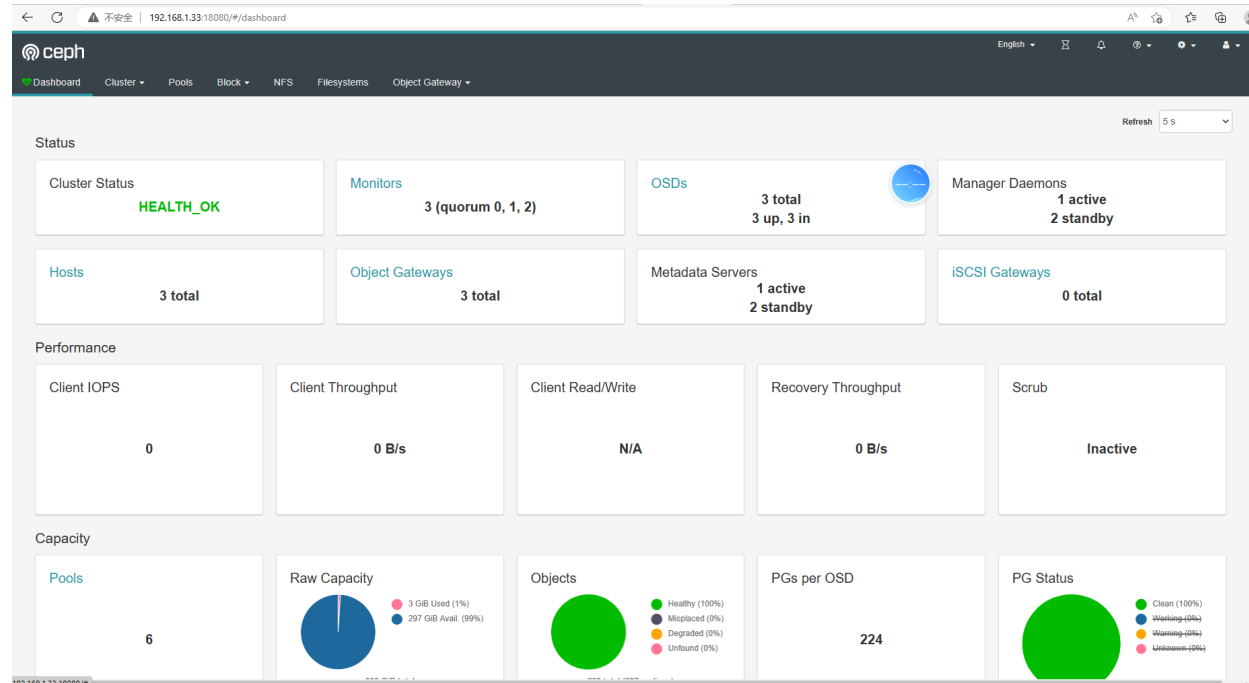
```
docker restart mgr
```

在master执行 查看Mgr DashBoard服务信息

```
docker exec mgr ceph mgr services
```

```
{  
  "dashboard": "http://master:18080/"  
}
```

管理控制台界面：



8 测试

8.1 添加rgw用户

添加账号

```
docker exec rgw radosgw-admin user create --uid="testuser" --display-name="lqf User"
```

下面这些信息在使用s3cmd工具和程序上传下载的时候需要使用。

```
{  
  "user": "testuser",  
  "access_key": "M9Q3I1BWRZVFC3AEO2XB",  
  "secret_key": "7U186Co06mD8CamPGV0r4qp5KrRnxqTaH2AeWEBM"  
}
```

###

查询账号信息

后续使用s3cmd、程序访问ceph时需要用到user对应的access_key和secret_key

```
docker exec rgw radosgw-admin user info --uid=testuser
```

8.2 客户端测试

8.2.1 安装s3cmd

客户机执行 `sudo apt install s3cmd`

8.2.2 配置s3cmd

配置分两个步骤：

1. 执行 `s3cmd --configure` （可以在普通权限执行）
2. 修改配置文件 `/root/.s3cfg`，（**要注意的是如果不是root的权限**，这个文件在home目录，比如 `/home/lqf/.s3cfg`）

2.1 s3cmd --configure

客户机执行 `s3cmd --configure`，开始配置，根据提示输入accessKey,securityKey 生成基本的配置文件

```
lqf@ubuntu:~/ceph/go-ceph$ 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: D3HTA2TRXBBE514USEQT ##此处填上一步获取到的access_key

Secret Key: AZxoYU6u3DkLUw9OMnRewfx73DxhjpdICwSjEIwH #此处填上一步获取到的secret_key

Default Region [US]: #默认即可，直接回车

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.
S3 Endpoint [s3.amazonaws.com]: #默认即可，直接回车

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]:

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]:      #默认即可，直接回车

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      #不使用https，填写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:      #默认即可，直接回车

New settings:
Access Key: D3HTA2TRXBBE514USEQT
Secret Key: AZxoYU6u3DkLUw9OMnRewfx73DxhjpdICwsjEIwH
Default Region: US
S3 Endpoint: s3.amazonaws.com
DNS-style bucket+hostname:port template for accessing a bucket: %
(bucket)s.s3.amazonaws.com
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] n      #测试访问，此时还没配置完，不测试，填写n

Save settings? [y/N] y      #是否保存，是，填写y
Configuration saved to '/root/.s3cfg'      (具体文件路径根据实际显示)

```

2.2 修改/root/.s3cfg配置文件

还没结束，还要修改刚生成的/root/.s3cfg（具体路径看上一步，比如我的是/home/lqf/.s3cfg）中的三处配置

```

cloudfront_host = [serverIP] (改成自己的服务端的IP)
host_base = [serverIP]:[Port] (改成自己的服务端的IP和端口)
host_bucket = [serverIP]:[Port]/%(bucket) (改成自己的服务端的IP和端口)

```

示例：我服务器本地的ceph集群环境，rgw默认端口为7480，ip填写自己服务器的ip即可

```

cloudfront_host = 192.168.1.27
host_base = 192.168.1.27:7480

```

host_bucket = %(bucket)192.168.1.27:7480

8.2.3 测试s3cmd命令

创建名为test-bucket的bucket

```
lqf@ubuntu:~/ceph/go-ceph$ s3cmd mb s3://test-bucket  
Bucket 's3://test-bucket/' created
```

异常情况：ERROR: S3 error: 416 (InvalidRange)，重启 mon

docker restart mon

然后再创建桶。

查看bucket桶列表

```
lqf@ubuntu:~/ceph/go-ceph$ s3cmd ls  
2023-03-07 07:52 s3://test-bucket  
即s3配置正常，可正常连接集群
```

8.2.4 s3cmd常用命令

针对bucket桶的操作

创建bucket

```
$ s3cmd mb s3://{bucket_name}
```

删除bucket (bucket需为空)

```
$ s3cmd rb s3://{bucket_name}
```

查看bucket列表或bucket内文件列表

```
s3cmd ls
```

```
s3cmd ls s3://{bucket_name}
```

针对bucket中文件的操作

上传文件到bucket中

```
$ s3cmd put fio-fio-3.10.zip s3://test-bucket
```


删除文件

```
s3cmd del s3://test-bucket/file.txt
```

批量删除文件

```
s3cmd del s3://test-bucket/aa*  
s3cmd del s3://test-bucket/test/*
```

批量上传文件

```
$ s3cmd put test/* s3://test-bucket
```

递归上传文件（可上传整个文件夹-包含文件夹）

```
#-r 递归参数，全称为：--recursive  
$ s3cmd put -r /root/test s3://test-bucket
```

同步目录下文件至bucket中（应该类似于git合流代码）

```
s3cmd sync ./test/ s3://test-bucket
```

复制bucket中文件到其他bucket中

```
s3cmd cp s3://test-bucket/aaaa s3://test-bucket-2
```

下载文件

```
s3cmd get s3://test-bucket/file.txt  
s3cmd get s3://test-bucket/file.txt /root/test/
```

针对权限的操作

将文件权限设置为所有人可读

```
$ s3cmd setacl --acl-public s3://test-bucket/file.txt
```

将bucket中整个文件夹设置权限为私有读（递归权限，文件夹下所有文件都生效）

```
$ s3cmd setacl --acl-private -r s3://test-bucket/test/
```

1. 更多参见官方

<https://tecadmin.net/install-s3cmd-manage-amazon-s3-buckets/>

9 重点推荐的资料

[存储策略指南 Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

[管理指南 Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

[操作指南 Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

[架构指南 Red Hat Ceph Storage 6 | Red Hat Customer Portal](#)

10 运维参考

解决mons are allowing insecure global_id reclaim问题:

禁用不安全模式:

```
$ ceph config set mon auth_allow_insecure_global_id_reclaim false
```

docker:

```
docker exec mon ceph config set mon auth_allow_insecure_global_id_reclaim false
```

11 参考

[遇到的问题 · 王康宁的笔记 · 看云 \(kancloud.cn\)](#)