分布式存储

存储分类:DAS直连存储 NAS 网络附加存储nfs,samba,ftp[共享文件系统] SAN 存储区域网络iscsi[共享块设备] SDS(software deffine storage)软件定义存储(分部署存储)Ceph

什么是Ceph

Ceph是一个分布式文件系统

具有高扩展,高可用,高性能特点

Ceph可以提供对象存储,块存储,文件系统存储

Ceph可以提供PB级别的存储空间(PB-->TB-->GB)

软件定义存储

ceph组件(默认3副本)

MON(ceph-mon) 集群监控组件

OSD(ceph-osd) 存储设备

MDS(ceph-MDS)文件系统

RGW(ceph-radowsgw) 对象存储网关

1T 1T 1T 1T

node1(osd) node2(osd) node3(osd) node4(osd)

mon mon(另一台安装) mon

client

**案例1:实验环境**

1.1问题

准备四台KVM虚拟机，其三台作为存储集群节点，一台安装为客户端，实现如下功能：

创建1台客户端虚拟机

创建3台存储集群虚拟机

配置主机名、IP地址、YUM源

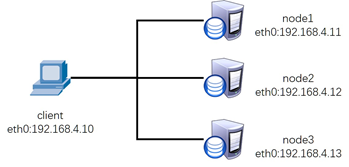
修改所有主机的主机名

配置无密码SSH连接

配置NTP时间同步

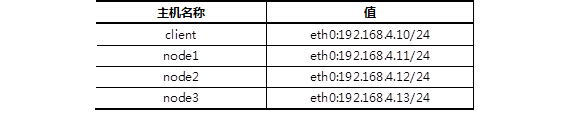
创建虚拟机磁盘

1.2方案

使用4台虚拟机，1台客户端、3台存储集群服务器，拓扑结构如图-1所示。

所有主机的主机名及对应的IP地址如表-1所示。

注意：所有主机基本系统光盘的YUM源必须提前配置好



1.3步骤

实现案例需要按照如下步骤进行

步骤一:安装前准备

[1]物理机为所有节点配置yum源服务器

提示:ceph10.iso在linux-soft/02目录

[root@room9pc01 ~]# mkdir /var/ftp/ceph

[root@room9pc01 ~]# mount ceph10.iso /var/ftp/ceph/

[2]配置无密码连接(包括自己也不需要密码)在node1操作

[root@node1 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N ''

[root@node1 ~]# for i in 10 11 12 13

do

ssh-copy-id 192.168.4.$i

done

[3]修改/etc/hosts并同步到所有主机

警告：/etc/hosts解析的域名必须与本机主机名一致！！！！

[root@node1 ~]# cat /etc/hosts

... ...

192.168.4.10 client

192.168.4.11 node1

192.168.4.12 node2

192.168.4.13 node3

警告：/etc/hosts解析的域名必须与本机主机名一致！！！！

[root@node1 ~]# for i in client node1 node2 node3

do

scp /etc/hosts $i:/etc/

done

[4]修改所有节点都需要配置yum源,并同步到所有主机

[root@node1 ~]# cat /etc/yum.repos.d/ceph.repo

[mon]

name=mon

baseurl=ftp://192.168.4.254/ceph/MON

gpgcheck=0

[osd]

name=osd

baseurl=ftp://192.168.4.254/ceph/OSD

gpgcheck=0

[tools]

name=tools

baseurl=ftp://192.168.4.254/ceph/Tools

gpgcheck=0

[root@node1 ~]# yum repolist #验证YUM源软件数量

源标识 源名称 状态

Dvd redhat 9,911

Mon mon 41

Osd osd 28

Tools tools 33

repolist: 10,013

[root@node1 ~]# for i in client node1 node2 node3

do

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

done

[5]所有节点主机与真实主机的NTP服务器同步时间

提示:默认真实物理机已经配置为NTP服务器

[root@node1 ~]# vim /etc/chrony.conf

… …

server 192.168.4.254 iburst

[root@node1 ~]# for i in client node1 node2 node3

do

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

ssh $i "systemctl restart chronyd" --->**修改配置文件需要重启服务**

done

步骤二:准备存储磁盘

物理机上为每个虚拟机准备3块20G磁盘(可以使用命令,也可以使用图形直接添加).

[root@room9pc01]#virt-manager

案例2:部署Ceph集群

2.1问题

沿用练习一,部署Ceph集群服务器,实现以下目标

安装部署工具ceph-deploy

创建ceph集群

准备日志磁盘分区

创建OSD存储空间

查看ceph状态，验证

2.2步骤

实现此案例需要按照如下步骤进行

步骤一:安装部署软件ceph-deploy

[1]在node1安装部署工具，学习工具的语法格式。

[root@node1 ~]# yum -y install ceph-deploy

[root@node1 ~]# ceph-deploy --help

[root@node1 ~]# ceph-deploy mon --help

[2]创建目录

[root@node1 ~]# mkdir ceph-cluster

[root@node1 ~]# cd ceph-cluster/

步骤二:部署Ceph集群

[1]创建Ceph集群配置,在ceph-cluster目录下生成Ceph配置文件.

**在ceph.conf配置文件中定义monitor主机是谁**

[root@node1 ceph-cluster]# ceph-deploy new node1 node2 node3

[2]给所有节点安装ceph相关软件包。

[root@node1 ceph-cluster]# for i in node1 node2 node3

do

ssh $i "yum -y install ceph-mon ceph-osd ceph-mds ceph-radosgw"

done

[3]初始化所有节点的mon服务,也就是启动mon服务,(主机名解析必须对)

[root@node1 ceph-cluster]# ceph-deploy mon create-initial--->启动mon

常见错误及解决方法(非必要操作.有错误可以参考):

如果提示如下错误信息:

[node1][ERROR ] admin\_socket: exception getting command descriptions: [Error 2] No such file or directory

解决方案如下（在node1操作）：

先检查自己的命令是否是在ceph-cluster目录下执行的！！！！如果确认是在该目录下执行的create-initial命令，依然报错，可以使用如下方式修复。

[root@node1 ceph-cluster]# vim ceph.conf #文件最后追加以下内容

public\_network = 192.168.4.0/24

修改后重新推送配置文件

[root@node1 ceph-cluster]# ceph-deploy --overwrite-conf config push node1 node2 node3

步骤三:创建OSD

备注:vdb1和vdb2这两个分区用来做存储服务器的journal缓存盘.

[root@node1 ceph-cluster]# for i in node1 node2 node3

do

ssh $i "parted /dev/vdb mklabel gpt"

ssh $i "parted /dev/vdb mkpart primary 1 50%"

ssh $i "parted /dev/vdb mkpart primary 50% 100%"

done

[2]磁盘分区后的默认权限无法让ceph软件对其尽心高度写操作,需要修改权限,node1,node2node3都需要操作,这里以node1为例

[root@node1 ceph-cluster]# chown ceph.ceph /dev/vdb1

[root@node1 ceph-cluster]# chown ceph.ceph /dev/vdb2

#上面的权限修改为临时操作，重启计算机后，权限会再次被重置。

#我们还需要将规则写到配置文件实现永久有效。

#规则：如果设备名称为/dev/vdb1则设备文件的所有者和所属组都设置为ceph。

#规则：如果设备名称为/dev/vdb2则设备文件的所有者和所属组都设置为ceph。

**[root@node1 ceph-cluster]# vim /etc/udev/rules.d/70-vdb.rules--->配置永久有效**

ENV{DEVNAME}=="/dev/vdb1",OWNER="ceph",GROUP="ceph"

ENV{DEVNAME}=="/dev/vdb2",OWNER="ceph",GROUP="ceph"

[3]初始化清空磁盘数据(仅node1操作即可)

[root@node1 ceph-cluster]# ceph-deploy disk zap node1:vdc node1:vdd

[root@node1 ceph-cluster]# ceph-deploy disk zap node2:vdc node2:vdd

[root@node1 ceph-cluster]# ceph-deploy disk zap node3:vdc node3:vdd

[4]创建OSD

重要：很多同学在这里会出错！将主机名、设备名称输入错误！！！

[root@node1 ceph-cluster]# ceph-deploy osd create \

node1:vdc:/dev/vdb1 node1:vdd:/dev/vdb2

//创建osd存储设备，vdc为集群提供存储空间，vdb1提供JOURNAL缓存，

//每个存储设备对应一个缓存设备，缓存需要SSD，不需要很大

[root@node1 ceph-cluster]# ceph-deploy osd create \

node2:vdc:/dev/vdb1 node2:vdd:/dev/vdb2---->启动osd

[root@node1 ceph-cluster]# ceph-deploy osd create \

node3:vdc:/dev/vdb1 node3:vdd:/dev/vdb2 ---->启动osd

常见错误及解决方法（非必须操作）。

使用osd create创建OSD存储空间时，如提示下面的错误提示：

[ceph\_deploy][ERROR ] RuntimeError: bootstrap-osd keyring not found; run 'gatherkeys'

可以使用如下命令修复文件，重新配置ceph的密钥文件：

[root@node1 ceph-cluster]# ceph-deploy gatherkeys node1 node2 node3

步骤四:验证测试

[1]查看集群状态.

[root@node1 ~]# ceph -s

[2]常见错误（非必须操作）。

如果查看状态包含如下信息：

health: HEALTH\_WARN

clock skew detected on node2, node3…

clock skew表示时间不同步，解决办法：请先将所有主机的时间都使用NTP时间同步！！！

Ceph要求所有主机时差不能超过0.05s，否则就会提示WARN。

如果状态还是失败，可以尝试执行如下命令，重启ceph服务：

[root@node1 ~]# systemctl restart ceph\\*.service ceph\\*.target

[root@node1]ceph osd tree #查看osd在的位置

**共享池概念**

共享镜像1 1G----->client:1G

共享镜像2 100G----->client:100G

共享镜像3 1000G---->client:100G

... ...

**案例3:创建Ceph块存储**

3.1问题

沿用练习一，使用Ceph集群的块存储功能，实现以下目标：

创建块存储镜像

客户端映射镜像

创建镜像快照

使用快照还原数据

使用快照克隆镜像

删除快照与镜像

3.2步骤

实现此案例需要按照如下步骤进行.

步骤一:创建镜像

[1]查看存储池

[root@node1 ~]# ceph osd lspools #查看存储池

0 rbd,

[2]创建镜像,查看镜像

[root@node1 ~]# rbd create demo-image --image-feature layering --size 10G

[root@node1 ~]# rbd create rbd/image --image-feature layering --size 10G

#这里的demo-image和image为创建的镜像名称，可以为任意字符。

#--image-feature参数指定我们创建的镜像有哪些功能，layering是开启COW功能。

#提示：ceph镜像支持很多功能，但很多是操作系统不支持的，我们只开启layering。

[root@node1 ~]# rbd list #查看镜像列表

[root@node1 ~]# rbd info demo-image #查看某个镜像详细信息

rbd image 'demo-image':

size 10240 MB in 2560 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.d3aa2ae8944a

format: 2

features: layering

步骤二:动态调整

[1]缩小容量

[root@node1 ~]# rbd resize --size 7G image --allow-shrink

[root@node1 ~]# rbd info image

[2]扩展容量

[root@node1 ~]# rbd resize --size 15G image

[root@node1 ~]# rbd info image

步骤三:通过KRBD访问

[1]客户端通过KRBD访问

#客户端需要安装ceph-common软件包

#拷贝配置文件（否则不知道集群在哪）

#拷贝连接密钥（否则无连接权限）

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

[root@client ~]# scp 192.168.4.11:/etc/ceph/ceph.conf /etc/ceph/

[root@client ~]# scp 192.168.4.11:/etc/ceph/ceph.client.admin.keyring/etc/ceph/

[root@client ~]# rbd map image

[root@client ~]# lsblk

[root@client ~]# rbd showmapped

id pool image snap device

0 rbd image - /dev/rbd0

[2]客户端格式化,挂在分区

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

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

[root@client ~]# echo "test" > /mnt/test.txt

步骤四:创建镜像快照

[1]查看镜像快照(默认所有镜像都没有有快照)

[root@node1 ~]# rbd snap ls image

[2]给镜像创建快照

[root@node1 ~]# rbd snap create image --snap image-snap1

#为image镜像创建快照，快照名称为image-snap1

[root@node1 ~]# rbd snap ls image

SNAPID NAME SIZE

4 image-snap1 15360 MB

[3]删除客户端写入测试文件

[root@client ~]# rm -rf /mnt/test.txt

[root@client ~]# umount /mnt

[4]还原快照

[root@node1 ~]# rbd snap rollback image --snap image-snap1

#客户端重新挂载分区

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

[root@client ~]# ls /mnt

创建一个新镜像,(rbd create)20G

运维部门1:

步骤五:创建快照克隆

[1]克隆快照

[root@node1 ~]# rbd snap protect image --snap image-snap1

[root@node1 ~]# rbd snap rm image --snap image-snap1 //会失败

[root@node1 ~]# rbd clone \

image --snap image-snap1 image-clone --image-feature layering

//使用image的快照image-snap1克隆一个新的名称为image-clone镜像

[2]查看克隆镜像与父镜像快照的关系

[root@node1 ~]# rbd info image-clone

rbd image 'image-clone':

size 15360 MB in 3840 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.d3f53d1b58ba

format: 2

features: layering

flags:

parent: rbd/image@image-snap1

#克隆镜像很多数据都来自于快照链

#如果希望克隆镜像可以独立工作，就需要将父快照中的数据，全部拷贝一份，但比较耗时！！！

[root@node1 ~]# rbd flatten image-clone #脱离快照,拷贝快照信息,使其成为独立镜像

[root@node1 ~]# rbd info image-clone

rbd image 'image-clone':

size 15360 MB in 3840 objects

order 22 (4096 kB objects)

block\_name\_prefix: rbd\_data.d3f53d1b58ba

format: 2

features: layering

flags:

**#注意，父快照信息没了！**

[root@node1 ~]# rbd snap unprotect image --snap image-snap1 #取消快照保护

[root@node1 ~]# rbd snap rm image --snap image-snap1 #可以删除快照

步骤六:其他操作

[1]客户端撤销磁盘映射

[root@client ~]# umount /mnt

[root@client ~]# rbd showmapped

id pool image snap device

0 rbd image - /dev/rbd0

//语法格式:

[root@client ~]# rbd unmap /dev/rbd0

day04

**案例1:快存储应用案例**

1.1问题

延续Day03的实验内容，演示块存储在KVM虚拟化中的应用案例，实现以下功能：

Ceph创建块存储镜像

客户端安装部署ceph软件

客户端部署虚拟机

客户端创建secret

设置虚拟机配置文件，调用ceph存储

1.2方案

使用Ceph存储创建镜像。

KVM虚拟机调用Ceph镜像作为虚拟机的磁盘。

1.3步骤

实现此案例需要按照如下步骤进行

[1]创建磁盘镜像

[root@node1 ~]# rbd create vm1-image --image-feature layering --size 10G

[root@node1 ~]# rbd list

[root@node1 ~]# rbd info vm1-image

[2]Ceph认证账户（仅查看即可）。

Ceph默认开启用户认证，客户端需要账户才可以访问，默认账户名称为client.admin，key是账户的密钥。

可以使用ceph auth添加新账户（案例我们使用默认账户）。

[root@node1 ~]# cat /etc/ceph/ceph.conf //配置文件

[global]

mon\_initial\_members = node1, node2, node3

mon\_host = 192.168.2.10,192.168.2.20,192.168.2.30

auth\_cluster\_required = cephx //开启认证

auth\_service\_required = cephx //开启认证

auth\_client\_required = cephx //开启认证

[root@node1 ~]# cat /etc/ceph/ceph.client.admin.keyring //账户文件

[client.admin]

key = AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg==

[3]创建KVM虚拟机(注意这里使用真机操作)

创建2台的KVM虚拟机，或者直接使用现有的虚拟机也可以。

[4]配置libvirt secret (真机操作)

编写想好信息文件,让KVM知道ceph的账户名称

[root@room9pc01 ~]# vim secret.xml //新建临时文件，内容如下

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

<usage type='ceph'>

<name>client.admin secret</name>

</usage>

</secret>

#使用XML配置文件创建secret

[root@room9pc01 ~]# virsh secret-define secret.xml

733f0fd1-e3d6-4c25-a69f-6681fc19802b

**//随机的UUID，这个UUID对应的有账户信息**

给secret绑定admin账户的密码，密码参考ceph.client.admin.keyring文件。

[root@room9pc01] virsh secret-set-value \

--secret 733f0fd1-e3d6-4c25-a69f-6681fc19802b \

--base64 AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg

//这里secret后面是之前创建的secret的UUID

//base64后面是client.admin账户的密码

//现在secret中既有账户信息又有密钥信息

1. 虚拟机XML配置文件

每个虚拟机都会有一个xml配置文件,包括:

虚拟机名称,内存,CPU,磁盘,网卡等信息

[root@room9pc01 ~]# vim /etc/libvirt/qemu/vm1.xml

//原始模板内容如下：

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

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

<source file='/var/lib/libvirt/images/vm1.qcow2'/>

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

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

</disk>

不推荐直接使用vim修改配置文件，推荐使用virsh edit修改配置文件，效果如下：

[root@room9pc01] virsh edit tedu\_node01 //tedu\_node01为虚拟机名称

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

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

<auth username='admin'>

<secret type='ceph' uuid='733f0fd1-e3d6-4c25-a69f-6681fc19802b'/>

</auth>

<source protocol='rbd' name='rbd/vm1-image'>  
 <host name='192.168.4.11' port='6789'/>  
 </source>

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

</disk>

备注：修改secret的UUID，修改source中的共享名name，修改dev设备名称。

**案例2:ceph文件系统**

2.1问题

延续前面实验,实现Ceph文件系统的功能.具体实现有以下功能:

部署MDSs节点

创建ceph文件系统

客户端挂载文件系统

2.2方案

添加一台虚拟机,部署MDS节点

主机的主机名及对应的IP地址如下表所示:

IMG_256

2.3步骤

实现此案例需要按照如下步骤进行

[1]添加一台新的虚拟机,要求如下:

IP地址:192.168.4.14

主机名:node4

配置yum源(包括rhel,ceph的源)

与client主机同步时间

node1允许无密码远程node4

[2]部署元数据服务器

登录node4安装ceph-mds软件包

[root@node4 ~]# yum -y install ceph-mds

登录node1部署节点操作

[root@node1 ~]# cd /root/ceph-cluster

//该目录，是最早部署ceph集群时，创建的目录

[root@node1 ceph-cluster]# ceph-deploy mds create node4

//远程nod4，拷贝配置文件，启动mds服务

如果没有配置文件则可以通过admin命令重新发送配置和秘钥(备选操作)

[root@node1 ceph-cluster]# ceph-deploy admin node4

//同步配置文件和key

[3]创建存储池

[root@node4 ~]# ceph osd pool create cephfs\_data 128

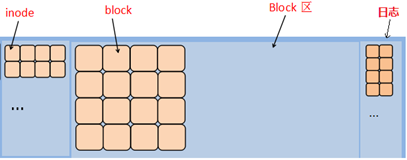
//创建存储池，对应128个PG

[root@node4 ~]# ceph osd pool create cephfs\_metadata 128

//创建存储池，对应128个PG

备注：一个文件系统是由inode和block两部分组成，效果如图-1所示。

inode存储文件的描述信息（metadata元数据），block中存储真正的数据。



[4]创建ceph文件系统

[root@node4 ~]# ceph mds stat //查看mds状态

e2:, 1 up:standby

[root@node4 ~]# ceph fs new myfs1 cephfs\_metadata cephfs\_data

new fs with metadata pool 2 and data pool 1

//注意，先写medadata池，再写data池

//默认，只能创建1个文件系统，多余的会报错

[root@node4 ~]# ceph fs ls

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

[root@node4 ~]# ceph mds stat

e4: 1/1/1 up {0=node4=up:creating}

[5]客户端挂载

[root@client ~]# mount -t ceph 192.168.4.11:6789:/ /mnt/cephfs/ \

-o name=admin,secret=AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg==

//注意:文件系统类型为ceph

//192.168.4.11为MON节点的IP（不是MDS节点）

//admin是用户名,secret是密钥

//密钥可以在/etc/ceph/ceph.client.admin.keyring中找到

案例3:创建对象存储服务器

块共享: map挂载

文件系统: mount

对象存储

必须使用使用API访问的一个存储形式

百度云盘[软件]pan.baidu.com

3.1问题

延续前面的实验,实现ceph对象存储的功能.具体实现有以下功能:

安装部署Rados Gateway

启动RGW服务

设置RGW的前端服务与端口

客户端测试

3.2步骤

步骤一:部署对象存储服务器

[1]准备实验环境,要求如下

IP地址:192.168.4.15

主机名:node5

配置yum源（包括rhel、ceph的源）

与Client主机同步时间

node1允许无密码远程node5

修改node1的/etc/hosts，并同步到所有node主机

[2]部署GRW软件包

[root@node1 ~]# ceph-deploy install --rgw node5

或者登陆node5手动yum安装软件包ceph-radosgw.

[3]新建网关实例

拷贝设置文件,启动一个rgw服务

[root@node1 ~]# cd /root/ceph-cluster

[root@node1 ~]# ceph-deploy rgw create node5

如果没有配置文件则可以通过admin命令重新发送配置和密钥（备选操作）

[root@node1 ceph-cluster]# ceph-deploy admin node4

//同步配置文件和key

登陆node5验证服务是否启动

[root@node5 ~]# ps aux |grep radosgw

ceph 4109 0.2 1.4 2289196 14972 ? Ssl 22:53 0:00 /usr/bin/radosgw -f --cluster ceph --name client.rgw.node4 --setuser ceph --setgroup ceph

[root@node5 ~]# systemctl status ceph-radosgw@\\*

[4]修改服务端口

登陆node5，RGW默认服务端口为7480，修改为8000或80更方便客户端记忆和使用

[root@node5 ~]# vim /etc/ceph/ceph.conf

[client.rgw.node5]

host = node5

rgw\_frontends = "civetweb port=8000"

//node5为主机名

//civetweb是RGW内置的一个web服务

步骤二:客户端测试(扩展选做实验)

[1]curl测试

[root@client ~]# curl 192.168.4.15:8000

<?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>

[2]使用第三方软件访问

登陆node5（RGW）创建账户

[root@node5 ~]# radosgw-admin user create \

--uid="testuser" --display-name="First User"

… …

"keys": [

{

"user": "testuser",

"access\_key": "5E42OEGB1M95Y49IBG7B",

"secret\_key": "i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6"

}

],

... ...

#

[root@node5 ~]# radosgw-admin user info --uid=testuser

//testuser为用户，key是账户访问密钥

[3]客户端安装软件

[root@client ~]# yum install s3cmd-2.0.1-1.el7.noarch.rpm

修改软件配置文件,(注意:除了下面设置的内容,其他提示默认回车)

[root@client ~]# s3cmd --configure

Access Key: 5E42OEGB1M95Y49IBG7B  
Secret Key: i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6

S3 Endpoint [s3.amazonaws.com]: 192.168.4.15:8000

[%(bucket)s.s3.amazonaws.com]: %(bucket)s.192.168.4.15:8000

Use HTTPS protocol [Yes]: No

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

Save settings? [y/N] y

//注意，其他提示都默认回车

[4]创建存储数据的bucket(类似于存储数据的目录)

[root@client ~]# s3cmd ls

[root@client ~]# s3cmd mb s3://my\_bucket

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

[root@client ~]# s3cmd ls

2018-05-09 08:14 s3://my\_bucket

[root@client ~]# s3cmd put /var/log/messages s3://my\_bucket/log/

[root@client ~]# s3cmd ls

2018-05-09 08:14 s3://my\_bucket

[root@client ~]# s3cmd ls s3://my\_bucket

DIR s3://my\_bucket/log/

[root@client ~]# s3cmd ls s3://my\_bucket/log/

2018-05-09 08:19 309034 s3://my\_bucket/log/messages

[5]测试下载功能

[root@client ~]# s3cmd get s3://my\_bucket/log/messages /tmp/

[6]测试删除功能

[root@client ~]# s3cmd del s3://my\_bucket/log/messages