**1 案例1：块存储应用案例**

**1.1 问题**

延续Day03的实验内容，演示块存储的应用案例，实现以下功能：

* 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

**1.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一、创建镜像快照**

1) 查看镜像快照（默认所有镜像都没有快照）。

1. **[**root@node1 **~]**# rbd snap ls jacob
2. #查看某个镜像有没有快照，jacob是镜像的名称，ls是list查看

2) 给镜像创建快照。

1. **[**root@node1 **~]**# rbd snap create jacob **--**snap jacob**-**snap1
2. #为jacob镜像创建快照，**--**snap指定快照名称，快照名称为jacob**-**snap1，快照名称可以任意
3. **[**root@node1 **~]**# rbd snap ls jacob
4. SNAPID NAME SIZE
5. **4** jacob**-**snap1 **15360** MB

3) 删除客户端写入的测试文件

1. **[**root@client **~]**# rm **-**rf **/**mnt**/**test**.**txt
2. **[**root@client **~]**# umount **/**mnt

4) 还原快照

1. **[**root@node1 **~]**# rbd snap rollback jacob **--**snap jacob**-**snap1
2. # rollback是回滚的意思，使用jacob**-**snap1快照回滚数据，对jacob镜像进行回滚数据
3. **[**root@client **~]**# mount **/**dev**/**rbd0 **/**mnt**/**        #客户端重新挂载分区
4. **[**root@client **~]**# ls **/**mnt     #查看数据是否被恢复

**步骤二：创建快照克隆**

1）克隆快照

1. **[**root@node1 **~]**# rbd snap protect jacob **--**snap jacob**-**snap1    #保护快照
2. #jacob是镜像名称，jacob**-**snap1是前面创建的快照（被保护的快照，不可以被删除）
3. **[**root@node1 **~]**# rbd snap rm jacob **--**snap jacob**-**snap1 #删除被保护的快照，会失败
4. **[**root@node1 **~]**# rbd clone **\**
5. jacob **--**snap jacob**-**snap1 jacob**-**clone **--**image**-**feature layering
6. #使用jacob镜像的快照jacob**-**snap1克隆一个新的名称为jacob**-**clone的镜像
7. #新镜像名称可以任意

2）查看克隆镜像与父镜像快照的关系

1. **[**root@node1 **~]**# rbd info jacob**-**clone
2. rbd image 'jacob-clone'**:**
3. size **15360** MB **in** **3840** objects
4. order **22** **(4096** kB objects**)**
5. block\_name\_prefix**:** rbd\_data**.**d3f53d1b58ba
6. format**:** **2**
7. features**:** layering
8. flags**:**
9. parent**:** rbd**/**jacob@jacob**-**snap1
10. #克隆镜像的很多数据都来自于快照链
11. #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！！！
12. **[**root@node1 **~]**# rbd flatten jacob**-**clone        #让新克隆的镜像与快照脱离关系
13. **[**root@node1 **~]**# rbd info jacob**-**clone            #查看镜像信息
14. rbd image 'jadob-clone'**:**
15. size **15360** MB **in** **3840** objects
16. order **22** **(4096** kB objects**)**
17. block\_name\_prefix**:** rbd\_data**.**d3f53d1b58ba
18. format**:** **2**
19. features**:** layering
20. flags**:**
21. #注意，父快照信息没了！

3）删除快照

1. **[**root@node1 **~]**# rbd snap unprotect jacob **--**snap jacob**-**snap1 #取消快照保护
2. **[**root@node1 **~]**# rbd snap rm jacob **--**snap jacob**-**snap1         #可以删除快照

**2 案例2：Ceph文件系统**

**2.1 问题**

延续前面的实验，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**

前面的块共享，仅允许同时一个客户端访问，无法实现多人同时使用块设备。

而Ceph的文件系统共享则允许多人同时使用。

下面假设使用虚拟机node3，部署MDS节点。

主机的主机名及对应的IP地址如表-1所示。

表－1 主机名称及对应IP地址表



**2.3 步骤**

实现此案例需要按照如下步骤进行。

1）添加一台虚拟机node3，要求如下：

IP地址:192.168.4.13

主机名:node3

配置yum源（包括操作系统的源、ceph的源）

与Client主机同步时间

node1允许无密码远程node3

修改node1的/etc/hosts，并同步到所有node主机

2）部署元数据服务器

登陆node3，安装ceph-mds软件包（如果前面课程已经安装，此步骤可以忽略）

1. **[**root@node3 **~]**# yum **-**y install ceph**-**mds

登陆node1部署节点操作

1. **[**root@node1 **~]**# cd **/**root**/**ceph**-**cluster
2. #该目录，是最早部署ceph集群时，创建的目录
3. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy mds create node3
4. #远程nod3，拷贝集群配置文件，启动mds服务

3）创建存储池

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。

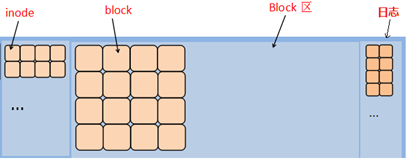


图-1

1. **[**root@node3 **~]**# ceph osd pool create cephfs\_data **64**
2. #创建存储池，共享池的名称为cephfs\_data，对应有**64**个PG
3. #共享池名称可以任意
4. **[**root@node3 **~]**# ceph osd pool create cephfs\_metadata **64**
5. #创建存储池，共享池的名称为cephfs\_metadata，对应有**64**个PG

PG拓扑如图-2所示。

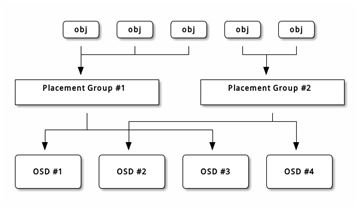


图-2

PG是一个逻辑概念，没有对应的物质形态，是为了方便管理OSD而设计的概念。

为了方便理解，可以把PG想象成为是目录，可以创建32个目录来存放OSD，也可以创建64个目录来存放OSD。

4）创建Ceph文件系统

1. **[**root@node3 **~]**# ceph fs **new** myfs1 cephfs\_metadata cephfs\_data
2. #myfs1是名称，名称可以任意，注意，先写metadata池，再写data池
3. #fs是filesystem的缩写，filesystem中文是文件系统
4. #默认，只能创建**1**个文件系统，多余的会报错
5. **[**root@node3 **~]**# ceph fs ls
6. name**:** myfs1**,** metadata pool**:** cephfs\_metadata**,** data pools**:** **[**cephfs\_data **]**

5）客户端挂载（客户端需要安装ceph-common，前面的课程已经安装）

1. **[**root@client **~]**# mount **-**t ceph **192.168.4.11:6789:**/ /mnt **\**
2. **-**o name**=**admin**,**secret**=**AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg**==**
3. #注意**:-**t（type）指定文件系统类型，文件系统类型为ceph
4. #**-**o（option）指定mount挂载命令的选项，选项包括name账户名和secret密码
5. #**192.168.4.11**为MON节点的IP（不是MDS节点），**6789**是MON服务的端口号
6. #admin是用户名**,**secret后面是密钥
7. #密钥可以在/etc/ceph**/**ceph**.**client**.**admin**.**keyring中找到

##扩展知识：创建ceph用户，查看用户##

1. **[**root@node1 **~]**# ceph auth get**-**or**-**create client**.**nb **\**
2. osd 'allow \*' **\**
3. mds 'allow \*' **\**
4. mon 'allow \*' **>** 文件名
5. # **>**是重定向导出，后面的文件名可以任意，没有文件会创建，有文件则会覆盖文件的内容
6. **[**root@node1 **~]**# ceph auth list #查看所有用户列表

**3 案例3：创建对象存储服务器**

**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:192.168.4.13

主机名:node3

配置yum源（包括操作系统的源、ceph的源）

与Client主机同步时间

node1允许无密码远程node3

修改node1的/etc/hosts，并同步到所有node主机

2）部署RGW软件包

1. **[**root@node3 **~]**# yum **-**y install ceph**-**radosgw

3）新建网关实例

拷贝配置文件，启动一个rgw服务

1. **[**root@node1 **~]**# cd **/**root**/**ceph**-**cluster
2. **[**root@node1 **~]**# ceph**-**deploy rgw create node3        #远程mode3启动rgw服务

登陆node3验证服务是否启动

1. **[**root@node3 **~]**# ps aux **|**grep radosgw
2. ceph **4109** **0.2** **1.4** **2289196** **14972** **?** Ssl **22:53** **0:00** **/**usr**/**bin**/**radosgw **-**f **--**cluster ceph **--**name client**.**rgw**.**node3 **--**setuser ceph **--**setgroup ceph
3. **[**root@node3 **~]**# systemctl status ceph**-**radosgw@**\\***

4）修改服务端口

登陆node3，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

1. **[**root@node3 **~]**# vim **/**etc**/**ceph**/**ceph**.**conf
2. **[**client**.**rgw**.**node3**]**
3. host **=** node3
4. rgw\_frontends **=** "civetweb port=8000"
5. #node3为主机名
6. #civetweb是RGW内置的一个web服务
7. **[**root@node3 **~]**# systemctl restart ceph**-**radosgw@**\\***

**步骤二：客户端测试（扩展选做实验）**

1）curl测试

1. **[**root@client **~]**# curl **192.168.4.13:8000**

2）使用第三方软件访问

登陆node3（RGW）创建账户

1. **[**root@node3 **~]**# radosgw**-**admin user create **\**
2. **--**uid**=**"testuser" **--**display**-**name**=**"First User"
3. … …
4. "keys"**:** **[**
5. **{**
6. "user"**:** "testuser"**,**
7. "access\_key"**:** "5E42OEGB1M95Y49IBG7B"**,**
8. "secret\_key"**:** "i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6"
9. **}**
10. **],**
11. **[**root@node5 **~]**# radosgw**-**admin user info **--**uid**=**testuser
12. //testuser为用户名，access\_key和secret\_key是账户密钥

3）客户端安装软件（软件需要自己上网搜索下载）

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

修改软件配置（注意，除了下面设置的内容，其他提示都默认回车）

1. **[**root@client **~]**# s3cmd **--**configure
2. Access Key**:** 5E42OEGB1M95Y49IBG7B  
   Secret Key**:** i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6
3. S3 Endpoint **[**s3**.**amazonaws**.**com**]:** **192.168.4.13:8000**
4. **[%(**bucket**)**s**.**s3**.**amazonaws**.**com**]:** **%(**bucket**)**s**.192.168.4.13:8000**
5. Use HTTPS protocol **[**Yes**]:** No
6. Test access **with** supplied credentials**?** **[**Y**/**n**]** n
7. Save settings**?** **[**y**/**N**]** y
8. #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）

1. **[**root@client **~]**# s3cmd ls
2. **[**root@client **~]**# s3cmd mb s3**:**//my\_bucket
3. Bucket 's3://my\_bucket/' created
4. **[**root@client **~]**# s3cmd ls
5. **2018-05-09** **08:14** s3**:**//my\_bucket
6. **[**root@client **~]**# s3cmd put **/var**/log/messages s3**:**//my\_bucket/log/
7. **[**root@client **~]**# s3cmd ls s3**:**//my\_bucket
8. DIR s3**:**//my\_bucket/log/
9. **[**root@client **~]**# s3cmd ls s3**:**//my\_bucket/log/
10. **2018-05-09** **08:19** **309034** s3**:**//my\_bucket/log/messages

5）测试下载功能

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

6）测试删除功能

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

附加知识总结：（Ceph操作思路）

一、准备工作：

IP，主机名，hosts解析，ssh密钥，时间同步，yum源，防火墙，selinux

二、部署ceph：

1.安装软件

1. ceph**-**deploy**(**脚本**)**
2. ceph**-**mon ceph**-**osd ceph**-**mds ceph**-**radosgw**(**集群**)**

2.修改配置启动服务mon

1. mkdir 目录；cd 目录
2. ceph**-**deploy **new** node1 node2 node3 **(**生成配置文件**)**
3. ceph**-**deploy mon create**-**initial **(**启动服务**)**

3.启动osd共享硬盘

1. ceph**-**deploy disk zap 主机名**:**磁盘名 **...** **...**
2. ceph**-**deploy osd create 主机名**:**磁盘 **...** **...**

三、使用Ceph思路:

1.块共享

1. 服务器**:** rbd create 创建一个共享镜像
2. 客户端**:** 安装cpeh**-**common**;** cp 配置文件和密钥
3. rbd map **|** rbd unmap

2.文件系统共享(文件系统由inode和block)

服务器: 创建两个共享池(名称任意)

使用两个共享池合并一个文件系统

安装ceph-mds软件，并启动服务(ceph-deploy mds create node3)

1. 客户端： mount **-**t MON的IP**:6789:**/ /挂载点 **-**o name**=**用户名**,**secret**=**密码

3.对象存储

服务器启动一个radosgw即可(RGW)

1. ceph**-**deploy rgw create node3

四、ceph-deploy脚本用法：

1. ceph**-**deploy **new** node1 node2 node3 #生成配置文件
2. ceph**-**deploy mon create**-**initial #远程所有主机启动mon服务
3. ceph**-**deploy disk zap 主机名**:**磁盘名 #初始化磁盘
4. ceph**-**deploy osd create 主机名**:**磁盘名 #远程主机并启动osd服务
5. ceph**-**deploy mds create 主机名 #远程主机并启动mds服务
6. ceph**-**deploy rgw create 主机名 #远程主机并启动RGW服务