[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY04/CASE/01/index.html" \l "page_top_case)

# NSD CLUSTER DAY04

1. [案例1：块存储应用案例](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY04/CASE/01/index.html" \l "case1)
2. [案例2：Ceph文件系统](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY04/CASE/01/index.html" \l "case2)
3. [案例3：创建对象存储服务器](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY04/CASE/01/index.html" \l "case3)

## 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。

1. [root@node1 ~]# rbd create vm1-image --image-feature layering --size 10G
2. [root@node1 ~]# rbd list
3. [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）。

1. [root@node1 ~]# cat /etc/ceph/ceph.conf                    //配置文件
2. [global]
3. mon\_initial\_members = node1, node2, node3
4. mon\_host = 192.168.2.10,192.168.2.20,192.168.2.30
5. auth\_cluster\_required = cephx                                //开启认证
6. auth\_service\_required = cephx                                //开启认证
7. auth\_client\_required = cephx                                //开启认证
8. [root@node1 ~]# cat /etc/ceph/ceph.client.admin.keyring        //账户文件
9. [client.admin]
10. key = AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg==

3）创建KVM虚拟机（注意：这里使用真实机操作！！！）。

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（注意：这里使用真实机操作！！！）。

编写账户信息文件，让KVM知道ceph的账户名称。

1. [root@room9pc01 ~]# vim secret.xml #新建临时文件，内容如下
2. <secret ephemeral='no' private='no'>
3. <usage type='ceph'>
4. <name>client.admin secret</name>
5. </usage>
6. </secret>
7. #使用XML配置文件创建secret
8. [root@room9pc01 ~]# virsh secret-define secret.xml
9. #命令会生成随机的UUID，这个UUID对应的有账户信息
10. [root@room9pc01 ~]# virsh secret-list #查看Secret信息

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

1. [root@room9pc01] virsh secret-set-value \
2. --secret 733f0fd1-e3d6-4c25-a69f-6681fc19802b \
3. --base64 AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg
4. #这里secret后面是前一步secret-fine创建的UUID
5. #base64后面是client.admin账户的密码

5）虚拟机的XML配置文件。

每个虚拟机都会有一个XML配置文件，包括：

虚拟机的名称、内存、CPU、磁盘、网卡等信息。

1. [root@room9pc01 ~]# virsh edit 虚拟机名称
2. //原始模板内容如下：
3. <disk type='file' device='disk'>
4. <driver name='qemu' type='qcow2'/>
5. <source file='/var/lib/libvirt/images/vm1.qcow2'/>
6. <target dev='vda' bus='virtio'/>
7. <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
8. </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：

1. [root@room9pc01] virsh edit tedu\_node01                //tedu\_node01为虚拟机名称
2. <disk type='network' device='disk'>
3. <driver name='qemu' type='raw'/>
4. <auth username='admin'>
5. <secret type='ceph' uuid='733f0fd1-e3d6-4c25-a69f-6681fc19802b'/>
6. </auth>
7. <source protocol='rbd' name='rbd/vm1-image'>\_x000B\_ <host name='192.168.4.11' port='6789'/>\_x000B\_ </source>
8. <target dev='vda' bus='virtio'/>
9. </disk>

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

## 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

添加一台虚拟机，部署MDS节点。

主机的主机名及对应的IP地址如表-1所示。

表－1 主机名称及对应IP地址表

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

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node4，安装ceph-mds软件包

1. [root@node4 ~]# 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 node4
4. //远程nod4，拷贝配置文件，启动mds服务

如果没有密钥和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）

1. [root@node1 ceph-cluster]# ceph-deploy admin node4
2. //同步配置文件和key

3）创建存储池

1. [root@node4 ~]# ceph osd pool create cephfs\_data 128
2. //创建存储池，对应128个PG
3. [root@node4 ~]# ceph osd pool create cephfs\_metadata 128
4. //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。

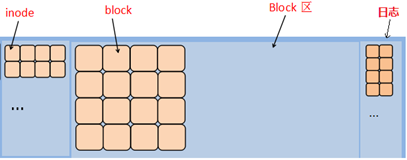


图-1

4）创建Ceph文件系统

1. [root@node4 ~]# ceph fs new myfs1 cephfs\_metadata cephfs\_data
2. new fs with metadata pool 2 and data pool 1
3. //注意，先写medadata池，再写data池
4. //默认，只能创建1个文件系统，多余的会报错
5. [root@node4 ~]# ceph fs ls
6. name: myfs1, metadata pool: cephfs\_metadata, data pools: [cephfs\_data ]

5）客户端挂载

1. [root@client ~]# mount -t ceph 192.168.4.11:6789:/ /mnt/cephfs/ \
2. -o name=admin,secret=AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg==
3. //注意:文件系统类型为ceph
4. //192.168.4.11为MON节点的IP（不是MDS节点）
5. //admin是用户名,secret是密钥
6. //密钥可以在/etc/ceph/ceph.client.admin.keyring中找到

## 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:192.168.4.15

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RGW软件包

1. [root@node1 ~]# yum -y install ceph-radosgw
2. 或者使用脚本自动安装ceph-deploy install --rgw node5.

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 node5

如果没有密钥和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）

1. [root@node1 ceph-cluster]# ceph-deploy admin node4
2. //同步配置文件和key

登陆node5验证服务是否启动

1. [root@node5 ~]# 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.node4 --setuser ceph --setgroup ceph
3. [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

1. [root@node5 ~]# vim /etc/ceph/ceph.conf
2. [client.rgw.node5]
3. host = node5
4. rgw\_frontends = "civetweb port=8000"
5. //node5为主机名
6. //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

1. [root@client ~]# curl 192.168.4.15:8000
2. <?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

1. [root@node5 ~]# 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. ... ...
12. #
13. [root@node5 ~]# radosgw-admin user info --uid=testuser
14. //testuser为用户，key是账户访问密钥

3）客户端安装软件

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\_x000B\_Secret Key: i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6
3. S3 Endpoint [s3.amazonaws.com]: 192.168.4.15:8000
4. [%(bucket)s.s3.amazonaws.com]: %(bucket)s.192.168.4.15:8000
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