

# NSD CLUSTER DAY04

1. [案例1：块存储应用案例](#)
2. [案例2：Ceph文件系统](#)
3. [案例3：创建对象存储服务器](#)

## 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。

01. `[root@node1 ~]# rbd create vm1-image --image-feature layering --size 10G`
02. `[root@node1 ~]# rbd list`
03. `[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）。

01. `[root@node1 ~]# cat /etc/ceph/ceph.conf` //配置文件
02. `[global]`
03. `mon_initial_members = node1, node2, node3`
04. `mon_host = 192.168.2.10,192.168.2.20,192.168.2.30`
05. `auth_cluster_required = cephx` //开启认证
06. `auth_service_required = cephx` //开启认证
07. `auth_client_required = cephx` //开启认证

[Top](#)

```

08.
09. [root@node1 ~]# cat /etc/ceph/ceph.client.admin.keyring //账户文件
10. [client.admin]
11.     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。

```

01. [root@room9pc01 ~]# vim secret.xml //新建临时文件，内容如下
02. <secret ephemeral='no' private='no'>
03.     <usage type='ceph'>
04.         <name>client.admin secret</name>
05.     </usage>
06. </secret>
07. #使用XML配置文件创建secret
08. [root@room9pc01 ~]# virsh secret-define secret.xml
09. 733f0fd1-e3d6-4c25-a69f-6681fc19802b
10. //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。

```

01. [root@room9pc01] virsh secret-set-value \
02. --secret 733f0fd1-e3d6-4c25-a69f-6681fc19802b \
03. --base64 AQBTSdRapUxBKRAANXtteNUyoEmQHveb75bISg
04. //这里secret后面是之前创建的secret的UUID
05. //base64后面是client.admin账户的密码
06. //现在secret中既有账户信息又有密钥信息

```

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。

```

01. [root@room9pc01 ~]# vim /etc/libvirt/qemu/vm1.xml
02. //原始模板内容如下：
03. <disk type='file' device='disk'>

```

[Top](#)

```

04.      <driver name='qemu' type='qcow2' />
05.      <source file='/var/lib/libvirt/images/vm1.qcow2' />
06.      <target dev='vda' bus='virtio' />
07.      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
08.      </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：

```

01.  [root@room9pc01] virsh edit tedu_node01           //tedu_node01为虚拟机名称
02.  <disk type='network' device='disk'>
03.      <driver name='qemu' type='raw' />
04.      <auth username='admin'>
05.          <secret type='ceph' uuid='733f0fd1-e3d6-4c25-a69f-6681fc19802b' />
06.      </auth>
07.      <source protocol='rbd' name='rbd/vm1-image'>      <host name='192.168.4.1
08.      <target dev='vda' bus='virtio' />
09.  </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MDS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

主机名称	值
node4	192.168.4.14

### 2.3 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

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

登陆node4，安装ceph-mds软件包

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

登陆node1部署节点操作

```
01. [root@node1 ~]# cd /root/ceph-cluster
02. //该目录，是最早部署ceph集群时，创建的目录
03.
04. [root@node1 ceph-cluster]# ceph-deploy mds create node4
05. //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密钥（备选操作）

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

3) 创建存储池

```
01. [root@node4 ~]# ceph osd pool create cephfs_data 128
02. //创建存储池，对应128个PG
03.
04. [root@node4 ~]# ceph osd pool create cephfs_metadata 128
05. //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。

[Top](#)

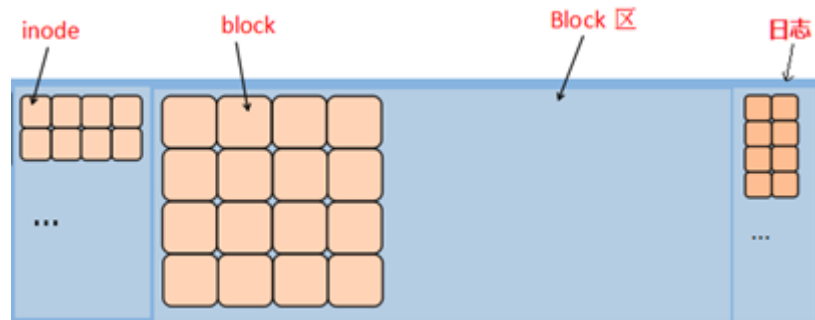


图-1

#### 4) 创建Ceph文件系统

```

01. [root@node4 ~]# ceph mds stat           //查看mds状态
02. e2: 1 up:standby
03.
04. [root@node4 ~]# ceph fs new myfs1 cephfs_metadata cephfs_data
05. new fs with metadata pool 2 and data pool 1
06. //注意，先写medadata池，再写data池
07. //默认，只能创建1个文件系统，多余的会报错
08.
09. [root@node4 ~]# ceph fs ls
10. name: myfs1, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
11.
12. [root@node4 ~]# ceph mds stat
13. e4: 1/1/1 up {0=node4=up:creating}

```

#### 5) 客户端挂载

```

01. [root@client ~]# mount -t ceph 192.168.4.11:6789:/ /mnt/cephfs/ \
02. -o name=admin,secret=AQBTsdRapUxBKRAANXtteNUyoEmQHveb75bISg==
03. //注意:文件系统类型为ceph
04. //192.168.4.11为MON节点的IP (不是MDS节点)
05. //admin是用户名,secret是密钥
06. //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：

[Top](#)

- 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

01. `[root@node1 ~]# ceph-deploy install --rgw node5`
02. 或者登陆node5手动yum安装软件包ceph-radosgw.

3) 新建网关实例

拷贝配置文件，启动一个rgw服务

01. `[root@node1 ~]# cd /root/ceph-cluster`
- 02.
03. `[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）

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

登陆node5验证服务是否启动

01. `[root@node5 ~]# ps aux |grep radosgw`
02. `ceph 4109 0.2 1.4 2289196 14972 ? Ssl 22:53 0:00 /usr/bin/radosgw -f`
03. `[root@node5 ~]# systemctl status ceph-radosgw@*`

[Top](#)

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

01. [root@node5 ~]# vim /etc/ceph/ceph.conf
02. [client.rgw.node5]
03. host = node5
04. rgw\_frontends = "civetweb port=8000"
05. //node5为主机名
06. //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

01. [root@client ~]# curl 192.168.4.15:8000
02. <?xml version="1.0" encoding="UTF-8"?><ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">



### 2) 使用第三方软件访问

登陆node5（RGW）创建账户

01. [root@node5 ~]# radosgw-admin user create \
02. --uid="testuser" --display-name="First User"
03. ... ..
04. "keys": [
05. {
06. "user": "testuser",
07. "access\_key": "5E42OEGb1M95Y49IBG7B",
08. "secret\_key": "i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6"
09. }
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

[Top](#)

01. [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）

01. [root@client ~]# s3cmd --configure
02. Access Key: 5E420EGB1M95Y49IBG7B Secret Key: i8YtM8cs7QDCK3rTRopb0TTPBFJV
03. S3 Endpoint [s3.amazonaws.com]: 192.168.4.15:8000
04. [% (bucket)s.s3.amazonaws.com]: %(bucket)s.192.168.4.15:8000
05. Use HTTPS protocol [Yes]: No
06. Test access with supplied credentials? [Y/n] n
07. Save settings? [y/N] y
08. //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）

01. [root@client ~]# s3cmd ls
02. [root@client ~]# s3cmd mb s3://my\_bucket
03. Bucket 's3://my\_bucket/' created
- 04.
05. [root@client ~]# s3cmd ls
06. 2018-05-09 08:14 s3://my\_bucket
- 07.
08. [root@client ~]# s3cmd put /var/log/messages s3://my\_bucket/log/
- 09.
10. [root@client ~]# s3cmd ls
11. 2018-05-09 08:14 s3://my\_bucket
12. [root@client ~]# s3cmd ls s3://my\_bucket
13. DIR s3://my\_bucket/log/
14. [root@client ~]# s3cmd ls s3://my\_bucket/log/
15. 2018-05-09 08:19 309034 s3://my\_bucket/log/messages

#### 5) 测试下载功能

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

#### 6) 测试删除功能

[Top](#)



01. `[root@client ~]# s3cmd del s3://my_bucket/log/messages`

[Top](#)