7_Cluster04 块存储应用案例\分布式文件系统\对象存储

一 案例：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

1）创建磁盘镜像。

```
node1 ~]# rbd create vm1-image --image-feature  layering --size
10G
node1 ~]# rbd  list
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）。

```
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


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。

```
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

```
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 node1 上的 /etc/ceph/ceph.client.admin.keyring 文件。

```
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

5）编辑虚拟机的 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。

```
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：

```
room9pc01 ~] virsh edit client  #编辑虚拟机 client 的配置文件
<disk type='network' device='disk'>
     <driver name='qemu' type='raw'/>
     <auth username='admin'>
     <secret type='ceph' uuid='733f0fd1-e3d6-4c25-a69f-6681fc1
9802b'/>
     </auth>
     <source protocol='rbd' name='rbd/vm1-image'><host name='19
2.168.4.11' 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。

二 案例：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

主机名称	值
node4	192.168.4.14

2.3 步骤

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

```
node4 ~]# yum -y install ceph-mds
```

登陆 node1 部署节点操作

node1 ~]# cd /root/ceph-cluster

#该目录，是最早部署 ceph 集群时，创建的目录

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密钥（备选操作）

node1 ceph-cluster]# ceph-deploy admin node4

#同步配置文件和 key

3）创建存储池

node4 ~]# **ceph osd pool create** cephfs_data 128

#创建存储池，对应 128 个 PG

PG 为存储对象的最小单位,1 个对象只能属于 1 个 PG,1 个 PG 可包含对个对象

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。

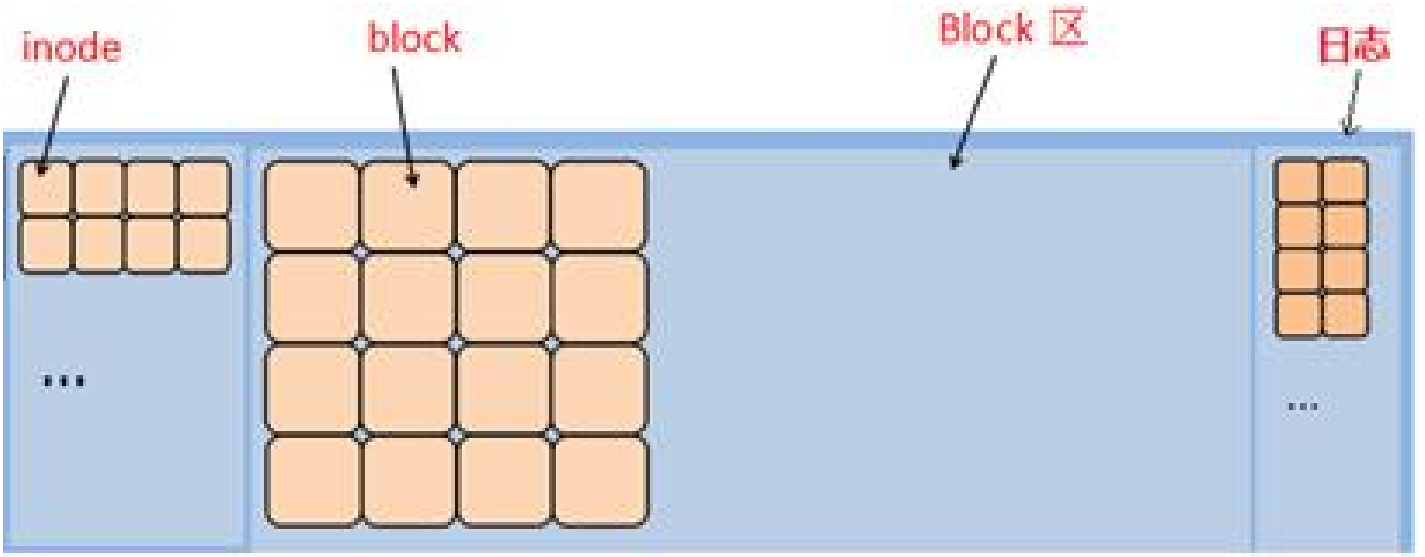

图-1

inode 默认 128byte,block 默认 4096byte

4）创建 Ceph 文件系统

node4 ~]# **ceph mds stat** #查看 mds 状态

e2:, 1 up:standby

node4 ~]# **ceph fs new** myfs1 cephfs_metadata cephfs_data

new fs with metadata pool 2 and data pool 1

#创建 ceph 文件系统,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

node4 ~]# **ceph fs ls**

name: myfs1, metadata pool: cephfs_metadata, data pools: [cephfs_data]

node4 ~]# ceph mds stat

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

5）客户端挂载

```
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

三 案例：创建对象存储服务器

对象存储:必须使用 API 访问的一个存储形式

示例:百度云盘,通过百度云盘这个软件访问存储的数据

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

```
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

拷贝配置文件，启动一个 rgw 服务

```
node1 ~]# cd /root/ceph-cluster
```

```
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）

```
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

```
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
```

```
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

```
node5 ~]#  vim  /etc/ceph/ceph.conf
[client.rgw.node5]
host = node5
rgw_frontends = "civetweb port=8000"
#配置文件内追加以上 3 行,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

```
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>anon
ymous</ID><DisplayName></DisplayName></Owner><Buckets></Bucket
s></ListAllMyBucketsResult>
```

2）使用第三方软件访问

登陆 node5（RGW）创建账户

```
node5 ~]#  radosgw-admin user create \
```

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

… …

"keys": [

       {

          "user": "testuser",

          "access_key": "5E42OEGB1M95Y49IBG7B",

          "secret_key":

"i8YtM8cs7QDCK3rTRopb0TTPBFJVXdEryRbeLGK6"

       }

    ],

... ...

#

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

```
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）

```
client ~]#  s3cmd --configure

Access     Key:     5E42OEGB1M95Y49IBG7B_x000B_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）

```
client ~]# s3cmd ls

client ~]# s3cmd mb s3:#my_bucket

Bucket 's3:#my_bucket/' created


client ~]# s3cmd ls

2018-05-09 08:14 s3:#my_bucket


client ~]# s3cmd put /var/log/messages s3:#my_bucket/log/


client ~]# s3cmd ls

2018-05-09 08:14 s3:#my_bucket

client ~]# s3cmd ls s3:#my_bucket

DIR s3:#my_bucket/log/
```

```
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

```
client ~]# s3cmd get s3:#my_bucket/log/messages /tmp/
```

6）测试删除功能

```
client ~]# s3cmd del s3:#my_bucket/log/messages
```