**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

IMG_256

**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。**

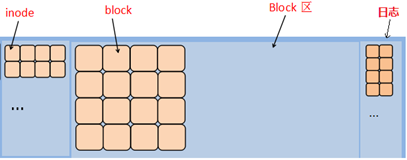


图-1

**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>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

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