NSD CLUSTER DAY03

1. 案例1：实验环境
2. 案例2：部署ceph集群
3. 案例3：创建Ceph块存储

1 案例1：实验环境

1.1 问题

准备四台KVM虚拟机，其三台作为存储集群节点，一台安装为客户端，实现如下功能：

- 创建1台客户端虚拟机
- 创建3台存储集群虚拟机
- 配置主机名、IP地址、YUM源
- 修改所有主机的主机名
- 配置无密码SSH连接
- 配置NTP时间同步
- 创建虚拟机磁盘

1.2 方案

使用4台虚拟机，1台客户端、3台存储集群服务器，拓扑结构如图-1所示。

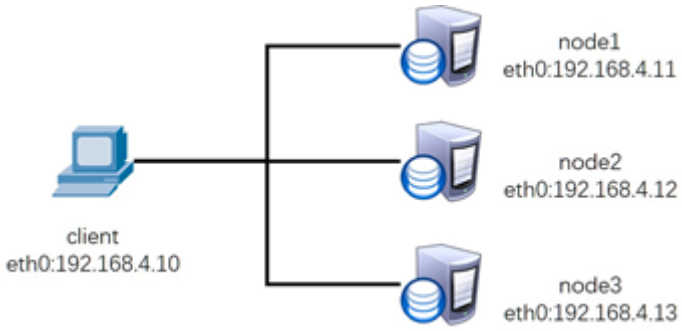

图-1

所有主机的主机名及对应的IP地址如表-1所示。

注意：所有主机基本系统光盘的YUM源必须提前配置好。

表－1 主机名称及对应IP地址表

主机名称	值
client	eth0:192.168.4.10/24
node1	eth0:192.168.4.11/24
node2	eth0:192.168.4.12/24
node3	eth0:192.168.4.13/24

1.3 步骤

实现此案例需要按照如下步骤进行。

Top

步骤一：安装前准备

1）物理机为所有节点配置yum源服务器。

提示：ceph10.iso在/linux-soft/02目录。

```
[root@room9pc01 ~]# mkdir  /var/ftp/ceph
[root@room9pc01 ~]# mount ceph10.iso /var/ftp/ceph/
```

2）配置无密码连接(包括自己远程自己也不需要密码)，在node1操作。

```
[root@node1 ~]# ssh-keygen   -f /root/.ssh/id_rsa    -N ''
[root@node1 ~]# for i in 10  11  12  13
 do
     ssh-copy-id  192.168.4.$i
 done
```

3）修改/etc/hosts并同步到所有主机。

警告：/etc/hosts解析的域名必须与本机主机名一致！！！！

```
 [root@node1 ~]# cat /etc/hosts
... ...
192.168.4.10  client
192.168.4.11     node1
192.168.4.12     node2
192.168.4.13     node3
```

警告：/etc/hosts解析的域名必须与本机主机名一致！！！！

```
[root@node1 ~]# for i in 10  11  12  13
do
scp  /etc/hosts  192.168.4.$i:/etc/
done
```

4）修改所有节点都需要配置YUM源，并同步到所有主机。

Top

```
[root@node1 ~]# cat /etc/yum.repos.d/ceph.repo
[mon]
name=mon
```

```
baseurl=ftp://192.168.4.254/ceph/MON
gpgcheck=0
[osd]
name=osd
baseurl=ftp://192.168.4.254/ceph/OSD
gpgcheck=0
[tools]
name=tools
baseurl=ftp://192.168.4.254/ceph/Tools
gpgcheck=0
[root@node1 ~]# yum repolist              #验证YUM源软件数量
源标识          源名称              状态
Dvd             redhat              9,911
Mon             mon                 41
Osd             osd                 28
Tools           tools               33
repolist: 10,013
[root@node1 ~]# for i in  10  11  12  13
do
scp  /etc/yum.repos.d/ceph.repo  192.168.4.$i:/etc/yum.repos.d/
done
```

5）所有节点主机与真实主机的NTP服务器同步时间。

提示：默认真实物理机已经配置为NTP服务器。

```
[root@node1 ~]# vim /etc/chrony.conf
… …
server 192.168.4.254   iburst


[root@node1 ~]# for i in 10 11 12 13
do
    scp /etc/chrony.conf 192.168.4.$i:/etc/
    ssh 192.168.4.$i "systemctl restart chronyd"
done
```

步骤三：准备存储磁盘

物理机上为每个虚拟机准备3块磁盘（可以使用命令，也可以使用图形直接添加）。

Top

```
01.  [root@room9pc01 ~]# virt-manager
```

2 案例2：部署ceph集群

2.1 问题

沿用练习一，部署Ceph集群服务器，实现以下目标：

- 安装部署工具ceph-deploy
- 创建ceph集群
- 准备日志磁盘分区
- 创建OSD存储空间
- 查看ceph状态，验证

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装部署软件ceph-deploy

1）在node1安装部署工具，学习工具的语法格式。

```
01.  [root@node1 ~]#  yum -y install ceph-deploy
02.  [root@node1 ~]#  ceph-deploy  --help
03.  [root@node1 ~]#  ceph-deploy mon --help
```

2）创建目录

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

步骤二：部署Ceph集群

1）创建Ceph集群配置,在ceph-cluster目录下生成Ceph配置文件。

在ceph.conf配置文件中定义monitor主机是谁。

```
01.  [root@node1 ceph-cluster]# ceph-deploy new node1 node2 node3
```

2）给所有节点安装ceph相关软件包。

```
01.  [root@node1 ceph-cluster]# for i in node1 node2 node3
02.  do
03.      ssh  $i "yum -y install ceph-mon ceph-osd ceph-mds ceph-radosgw"
```

Top

```
04.     done
```

3）初始化所有节点的mon服务，也就是启动mon服务（主机名解析必须对）。

```
01.     [root@node1 ceph-cluster]# ceph-deploy mon create-initial
```

常见错误及解决方法（非必要操作，有错误可以参考）：

如果提示如下错误信息：

```
01.     [node1][ERROR ] admin_socket: exception getting command descriptions: [Error 2] No
```

解决方案如下（在node1操作）：

先检查自己的命令是否是在ceph-cluster目录下执行的！！！！如果确认是在该目录下执行的create-initial命令，依然报错，可以使用如下方式修复。

```
01.     [root@node1 ceph-cluster]# vim ceph.conf      #文件最后追加以下内容
02.     public_network = 192.168.4.0/24
```

修改后重新推送配置文件:

```
01.     [root@node1 ceph-cluster]# ceph-deploy --overwrite-conf config push node1 node2 nod
```

步骤三：创建OSD

备注：vdb1和vdb2这两个分区用来做存储服务器的journal缓存盘。

```
01.     [root@node1 ceph-cluster]# for i in node1 node2 node3
02.     do
03.         ssh $i "parted /dev/vdb mklabel gpt"
04.         ssh $i "parted /dev/vdb mkpart primary 1 50%"
05.         ssh $i "parted /dev/vdb mkpart primary 50% 100%"
06.      done
```

Top

2）磁盘分区后的默认权限无法让ceph软件对其进行读写操作，需要修改权限。

node1、node2、node3都需要操作，这里以node1为例。

```
[root@node1 ceph-cluster]# chown  ceph.ceph  /dev/vdb1
[root@node1 ceph-cluster]# chown  ceph.ceph  /dev/vdb2
#上面的权限修改为临时操作，重启计算机后，权限会再次被重置。
#我们还需要将规则写到配置文件实现永久有效。
#规则：如果设备名称为/dev/vdb1则设备文件的所有者和所属组都设置为ceph。
#规则：如果设备名称为/dev/vdb2则设备文件的所有者和所属组都设置为ceph。
[root@node1 ceph-cluster]# vim /etc/udev/rules.d/70-vdb.rules
ENV{DEVNAME}=="/dev/vdb1",OWNER="ceph",GROUP="ceph"
ENV{DEVNAME}=="/dev/vdb2",OWNER="ceph",GROUP="ceph"
```

3）初始化清空磁盘数据（仅node1操作即可）。

```
[root@node1 ceph-cluster]# ceph-deploy disk  zap  node1:vdc   node1:vdd
[root@node1 ceph-cluster]# ceph-deploy disk  zap  node2:vdc   node2:vdd
[root@node1 ceph-cluster]# ceph-deploy disk  zap  node3:vdc   node3:vdd
```

4）创建OSD存储空间（仅node1操作即可）

重要：很多同学在这里会出错！将主机名、设备名称输入错误！！！

```
[root@node1 ceph-cluster]# ceph-deploy osd create \
 node1:vdc:/dev/vdb1 node1:vdd:/dev/vdb2
//创建osd存储设备，vdc为集群提供存储空间，vdb1提供JOURNAL缓存，
//一个存储设备对应一个缓存设备，缓存需要SSD，不需要很大
[root@node1 ceph-cluster]# ceph-deploy osd create \
 node2:vdc:/dev/vdb1 node2:vdd:/dev/vdb2
[root@node1 ceph-cluster]# ceph-deploy osd create \
 node3:vdc:/dev/vdb1 node3:vdd:/dev/vdb2
```

常见错误及解决方法（非必须操作）。

使用osd create创建OSD存储空间时，如提示下面的错误提示：

[ceph_deploy][ERROR] RuntimeError: bootstrap-osd keyring not found; run 'gatherkeys'

可以使用如下命令修复文件，重新配置ceph的密钥文件：

Top

```
[root@node1 ceph-cluster]#  ceph-deploy gatherkeys node1 node2 node3
```

步骤四：验证测试

1) 查看集群状态。

```
01.  [root@node1 ~]#  ceph  -s
```

2）常见错误（非必须操作）。

如果查看状态包含如下信息：

```
01.  health: HEALTH_WARN
02.          clock skew detected on  node2, node3…
```

clock skew表示时间不同步，解决办法：请先将所有主机的时间都使用NTP时间同步！！！

Ceph要求所有主机时差不能超过0.05s，否则就会提示WARN，如果使用NTP还不能精确同步时间，可以手动修改所有主机的ceph.conf，在[MON]下面添加如下一行：

```
01.  mon clock drift allowed = 1
```

如果状态还是失败，可以尝试执行如下命令，重启ceph服务：

```
01.  [root@node1 ~]#  systemctl restart ceph\*.service ceph\*.target
```

3 案例3：创建Ceph块存储

3.1 问题

沿用练习一，使用Ceph集群的块存储功能，实现以下目标：

- 创建块存储镜像
- 客户端映射镜像
- 创建镜像快照
- 使用快照还原数据
- 使用快照克隆镜像
- 删除快照与镜像

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建镜像

Top

1）查看存储池。

```
[root@node1 ~]# ceph osd lspools
0 rbd,
```

2）创建镜像、查看镜像

```
[root@node1 ~]# rbd create demo-image --image-feature  layering --size 10G
[root@node1 ~]# rbd create rbd/image --image-feature  layering --size 10G
```

#这里的demo-image和image为创建的镜像名称，可以为任意字符。
#--image-feature参数指定我们创建的镜像有哪些功能，layering是开启COW功能。
#提示：ceph镜像支持很多功能，但很多是操作系统不支持的，我们只开启layering。

```
[root@node1 ~]# rbd list
[root@node1 ~]# rbd info demo-image
rbd image 'demo-image':
    size 10240 MB in 2560 objects
    order 22 (4096 kB objects)
    block_name_prefix: rbd_data.d3aa2ae8944a
    format: 2
    features: layering
```

步骤二：动态调整

1）缩小容量

```
[root@node1 ~]# rbd resize --size 7G image --allow-shrink
[root@node1 ~]# rbd info image
```

2）扩容容量

```
[root@node1 ~]# rbd resize --size 15G image
[root@node1 ~]# rbd info image
```

步骤三：通过KRBD访问

Top

1）客户端通过KRBD访问

```
#客户端需要安装ceph-common软件包
#拷贝配置文件（否则不知道集群在哪）
#拷贝连接密钥（否则无连接权限）
[root@client ~]# yum -y  install ceph-common
[root@client ~]# scp 192.168.4.11:/etc/ceph/ceph.conf  /etc/ceph/
[root@client ~]# scp 192.168.4.11:/etc/ceph/ceph.client.admin.keyring \
/etc/ceph/
[root@client ~]# rbd map image
[root@client ~]#  lsblk
[root@client ~]# rbd showmapped
id pool image snap device
0  rbd  image -    /dev/rbd0
```

2) 客户端格式化、挂载分区

```
[root@client ~]# mkfs.xfs /dev/rbd0
[root@client ~]# mount /dev/rbd0 /mnt/
[root@client ~]# echo "test" > /mnt/test.txt
```

步骤四：创建镜像快照

1) 查看镜像快照（默认所有镜像都没有快照）。

```
[root@node1 ~]# rbd snap ls image
```

2) 给镜像创建快照。

```
[root@node1 ~]# rbd snap create image --snap image-snap1
#为image镜像创建快照，快照名称为image-snap1
[root@node1 ~]# rbd snap ls image
SNAPID NAME           SIZE
     4 image-snap1 15360 MB
```

3) 删除客户端写入的测试文件

Top

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

4) 还原快照

```
[root@node1 ~]# rbd snap rollback image --snap image-snap1
#客户端重新挂载分区
[root@client ~]# mount /dev/rbd0 /mnt/
[root@client ~]# ls  /mnt
```

步骤四：创建快照克隆

1）克隆快照

```
[root@node1 ~]#  rbd snap protect image --snap image-snap1
[root@node1 ~]#  rbd snap rm image --snap image-snap1    //会失败
[root@node1 ~]# rbd clone \
image --snap image-snap1 image-clone --image-feature layering
//使用image的快照image-snap1克隆一个新的名称为image-clone镜像
```

2）查看克隆镜像与父镜像快照的关系

```
[root@node1 ~]#  rbd info image-clone
rbd image 'image-clone':
    size 15360 MB in 3840 objects
    order 22 (4096 kB objects)
    block_name_prefix: rbd_data.d3f53d1b58ba
    format: 2
    features: layering
    flags:
    parent: rbd/image@image-snap1
#克隆镜像很多数据都来自于快照链
#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
[root@node1 ~]#  rbd flatten image-clone
[root@node1 ~]#  rbd info image-clone
rbd image 'image-clone':
    size 15360 MB in 3840 objects
    order 22 (4096 kB objects)
    block_name_prefix: rbd_data.d3f53d1b58ba                    Top
    format: 2
    features: layering
```

```
        flags:
    #注意，父快照信息没了！
    [root@node1 ~]#  rbd snap unprotect image --snap image-snap1    #取消快照保护
    [root@node1 ~]#  rbd snap rm image --snap image-snap1           #可以删除快照
```

步骤四：其他操作

1） 客户端撤销磁盘映射

```
[root@client ~]# umount /mnt
[root@client ~]# rbd showmapped
id pool image        snap device
0  rbd  image        -    /dev/rbd0
//语法格式:
[root@client ~]# rbd unmap /dev/rbd0
```

Top