[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY03/CASE/01/index.html" \l "page_top_case)

# NSD CLUSTER DAY03

1. [案例1：实验环境](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY03/CASE/01/index.html" \l "case1)
2. [案例2：部署ceph集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY03/CASE/01/index.html" \l "case2)
3. [案例3：创建Ceph块存储](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/CLUSTER/DAY03/CASE/01/index.html" \l "case3)

## 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。

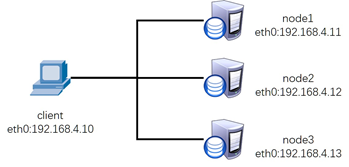
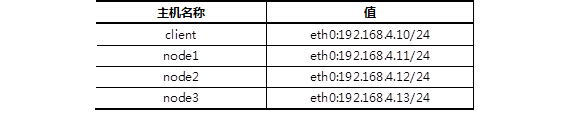


图-1

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



### 1.3 步骤

实现此案例需要按照如下步骤进行。

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。

1. [root@room9pc01 ~]# mkdir /var/ftp/ceph
2. [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。

1. [root@node1 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N ''
2. [root@node1 ~]# for i in 10 11 12 13
3. do
4. ssh-copy-id 192.168.4.$i
5. 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！！！！

1. [root@node1 ~]# cat /etc/hosts
2. ... ...
3. 192.168.4.10 client
4. 192.168.4.11     node1
5. 192.168.4.12     node2
6. 192.168.4.13     node3

警告：/etc/hosts解析的域名必须与本机主机名一致！！！！

1. [root@node1 ~]# for i in client node1 node2 node3
2. do
3. scp /etc/hosts $i:/etc/
4. done

4）修改所有节点都需要配置YUM源，并同步到所有主机。

1. [root@node1 ~]# cat /etc/yum.repos.d/ceph.repo
2. [mon]
3. name=mon
4. baseurl=ftp://192.168.4.254/ceph/MON
5. gpgcheck=0
6. [osd]
7. name=osd
8. baseurl=ftp://192.168.4.254/ceph/OSD
9. gpgcheck=0
10. [tools]
11. name=tools
12. baseurl=ftp://192.168.4.254/ceph/Tools
13. gpgcheck=0
14. [root@node1 ~]# yum repolist                #验证YUM源软件数量
15. 源标识            源名称                    状态
16. Dvd                redhat                    9,911
17. Mon                mon                        41
18. Osd                osd                        28
19. Tools            tools                    33
20. repolist: 10,013
21. [root@node1 ~]# for i in client node1 node2 node3
22. do
23. scp /etc/yum.repos.d/ceph.repo $i:/etc/yum.repos.d/
24. 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。

1. [root@node1 ~]# vim /etc/chrony.conf
2. … …
3. server 192.168.4.254 iburst
4. [root@node1 ~]# for i in client node1 node2 node3
5. do
6. scp /etc/chrony.conf $i:/etc/
7. ssh $i "systemctl restart chronyd"
8. done

步骤三：准备存储磁盘

物理机上为每个虚拟机准备3块20G磁盘（可以使用命令，也可以使用图形直接添加）。

1. [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。

1. [root@node1 ~]# yum -y install ceph-deploy
2. [root@node1 ~]# ceph-deploy --help
3. [root@node1 ~]# ceph-deploy mon --help

2）创建目录

1. [root@node1 ~]# mkdir ceph-cluster
2. [root@node1 ~]# cd ceph-cluster/

步骤二：部署Ceph集群

1）给所有节点安装ceph相关软件包。

1. [root@node1 ceph-cluster]# for i in node1 node2 node3
2. do
3. ssh $i "yum -y install ceph-mon ceph-osd ceph-mds ceph-radosgw"
4. done

2）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。

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

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）。

1. [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：

1. [node1][ERROR ] admin\_socket: exception getting command descriptions: [Error 2] No such file or directory

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。

1. [root@node1 ceph-cluster]# vim ceph.conf #文件最后追加以下内容
2. public\_network = 192.168.4.0/24

修改后重新推送配置文件:

1. [root@node1 ceph-cluster]# ceph-deploy --overwrite-conf config push node1 node2 node3

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。

1. [root@node1 ceph-cluster]# for i in node1 node2 node3
2. do
3. ssh $i "parted /dev/vdb mklabel gpt"
4. ssh $i "parted /dev/vdb mkpart primary 1 50%"
5. ssh $i "parted /dev/vdb mkpart primary 50% 100%"
6. done

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。

1. [root@node1 ceph-cluster]# chown ceph.ceph /dev/vdb1
2. [root@node1 ceph-cluster]# chown ceph.ceph /dev/vdb2
3. #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。
4. #我们还需要将规则写到配置文件实现永久有效。
5. #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。
6. #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。
7. [root@node1 ceph-cluster]# vim /etc/udev/rules.d/70-vdb.rules
8. ENV{DEVNAME}=="/dev/vdb1",OWNER="ceph",GROUP="ceph"
9. ENV{DEVNAME}=="/dev/vdb2",OWNER="ceph",GROUP="ceph"

3）初始化清空磁盘数据（仅node1操作即可）。

1. [root@node1 ceph-cluster]# ceph-deploy disk zap node1:vdc node1:vdd
2. [root@node1 ceph-cluster]# ceph-deploy disk zap node2:vdc node2:vdd
3. [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！！！

1. [root@node1 ceph-cluster]# ceph-deploy osd create \
2. node1:vdc:/dev/vdb1 node1:vdd:/dev/vdb2
3. //创建osd存储设备，vdc为集群提供存储空间，vdb1提供JOURNAL缓存，
4. //每个存储设备对应一个缓存设备，缓存需要SSD，不需要很大
5. [root@node1 ceph-cluster]# ceph-deploy osd create \
6. node2:vdc:/dev/vdb1 node2:vdd:/dev/vdb2
7. [root@node1 ceph-cluster]# ceph-deploy osd create \
8. 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：

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

步骤四：验证测试

1) 查看集群状态。

1. [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：

1. health: HEALTH\_WARN
2. 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。

如果状态还是失败，可以尝试执行如下命令，重启ceph服务：

1. [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

1）查看存储池。

1. [root@node1 ~]# ceph osd lspools
2. 0 rbd,

2）创建镜像、查看镜像

1. [root@node1 ~]# rbd create demo-image --image-feature layering --size 10G
2. [root@node1 ~]# rbd create rbd/jacob --image-feature layering --size 10G

#这里的demo-image和jacob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。

1. [root@node1 ~]# rbd list
2. [root@node1 ~]# rbd info demo-image
3. rbd image 'demo-image':
4. size 10240 MB in 2560 objects
5. order 22 (4096 kB objects)
6. block\_name\_prefix: rbd\_data.d3aa2ae8944a
7. format: 2
8. features: layering

步骤二：动态调整

1）扩容容量

1. [root@node1 ~]# rbd resize --size 15G jacob
2. [root@node1 ~]# rbd info jacob

2）缩小容量

1. [root@node1 ~]# rbd resize --size 7G jacob --allow-shrink
2. [root@node1 ~]# rbd info image

步骤三：通过KRBD访问

1）客户端通过KRBD访问

1. #客户端需要安装ceph-common软件包
2. #拷贝配置文件（否则不知道集群在哪）
3. #拷贝连接密钥（否则无连接权限）
4. [root@client ~]# yum -y install ceph-common
5. [root@client ~]# scp 192.168.4.11:/etc/ceph/ceph.conf /etc/ceph/
6. [root@client ~]# scp 192.168.4.11:/etc/ceph/ceph.client.admin.keyring \
7. /etc/ceph/
8. [root@client ~]# rbd map jacob
9. [root@client ~]# lsblk
10. [root@client ~]# rbd showmapped
11. id pool image snap device
12. 0 rbd jacob - /dev/rbd0

2) 客户端格式化、挂载分区

1. [root@client ~]# mkfs.xfs /dev/rbd0
2. [root@client ~]# mount /dev/rbd0 /mnt/
3. [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）。

1. [root@node1 ~]# rbd snap ls jacob

2) 给镜像创建快照。

1. [root@node1 ~]# rbd snap create jacob --snap jacob-snap1
2. #为jacob镜像创建快照，快照名称为jacob-snap1
3. [root@node1 ~]# rbd snap ls jacob
4. SNAPID NAME SIZE
5. 4 jacob-snap1 15360 MB

3) 删除客户端写入的测试文件

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

4) 还原快照

1. [root@node1 ~]# rbd snap rollback jacob --snap jacob-snap1
2. #客户端重新挂载分区
3. [root@client ~]# mount /dev/rbd0 /mnt/
4. [root@client ~]# ls /mnt

步骤五：创建快照克隆

1）克隆快照

1. [root@node1 ~]# rbd snap protect jacob --snap jacob-snap1
2. [root@node1 ~]# rbd snap rm jacob --snap jacob-snap1 //会失败
3. [root@node1 ~]# rbd clone \
4. jacob --snap jacob-snap1 jacob-clone --image-feature layering
5. //使用jacob镜像的快照jacob-snap1克隆一个新的名称为jacob-clone的镜像

2）查看克隆镜像与父镜像快照的关系

1. [root@node1 ~]# rbd info jacob-clone
2. rbd image 'jacob-clone':
3. size 15360 MB in 3840 objects
4. order 22 (4096 kB objects)
5. block\_name\_prefix: rbd\_data.d3f53d1b58ba
6. format: 2
7. features: layering
8. flags:
9. parent: rbd/jacob@jacob-snap1
10. #克隆镜像很多数据都来自于快照链
11. #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，但比较耗时！！！
12. [root@node1 ~]# rbd flatten jacob-clone
13. [root@node1 ~]# rbd info jacob-clone
14. rbd image 'jadob-clone':
15. size 15360 MB in 3840 objects
16. order 22 (4096 kB objects)
17. block\_name\_prefix: rbd\_data.d3f53d1b58ba
18. format: 2
19. features: layering
20. flags:
21. #注意，父快照信息没了！
22. [root@node1 ~]# rbd snap unprotect jacob --snap jacob-snap1 #取消快照保护
23. [root@node1 ~]# rbd snap rm jacob --snap jacob-snap1         #可以删除快照

步骤六：其他操作

1） 客户端撤销磁盘映射

1. [root@client ~]# umount /mnt
2. [root@client ~]# rbd showmapped
3. id pool image snap device
4. 0 rbd jacob - /dev/rbd0
5. //语法格式:
6. [root@client ~]# rbd unmap /dev/rbd0