**NSD CLUSTER DAY03**

1. [案例1：实验环境](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY03/CASE/01/index.html#case1)
2. [案例2：部署ceph集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY03/CASE/01/index.html#case2)
3. [案例3：创建Ceph块存储](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY03/CASE/01/index.html#case3)

**1 案例1：实验环境**

**1.1 问题**

准备四台虚拟机，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。

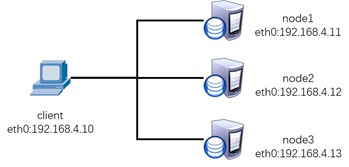
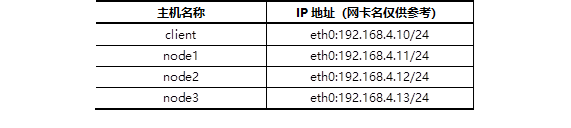


图-1

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



Ceph组件架构如图-2所示。

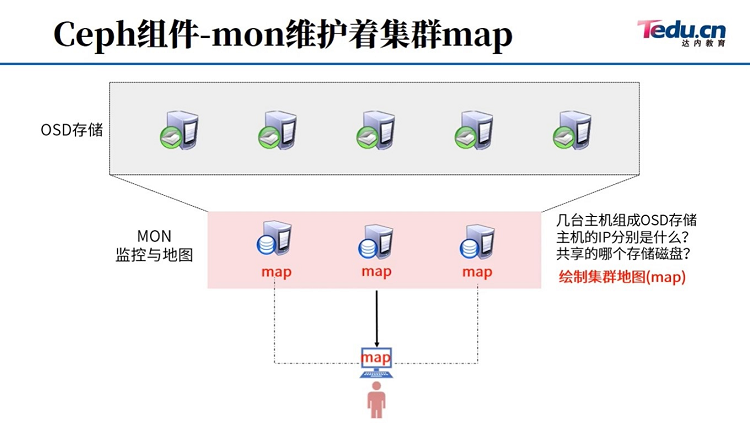


图-2

Ceph会对数据进行切割处理，如图-3所示。

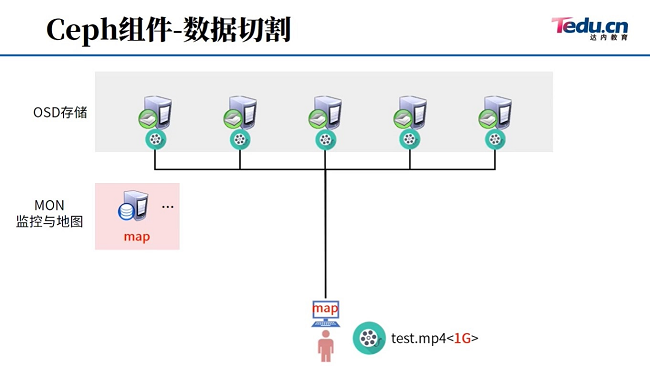


图-3

Ceph随机读写数据的思路，如图-4所示。

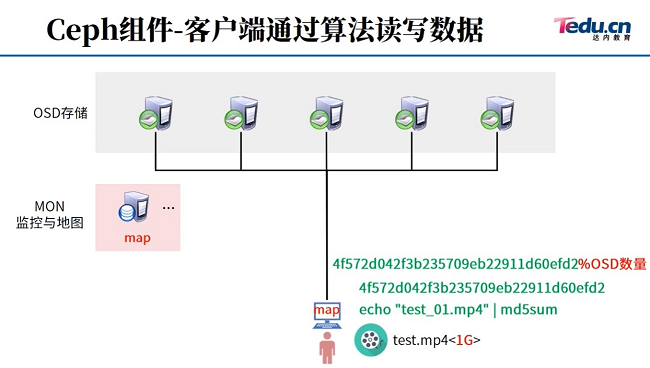


图-4

Ceph集群结构如图-5所示。

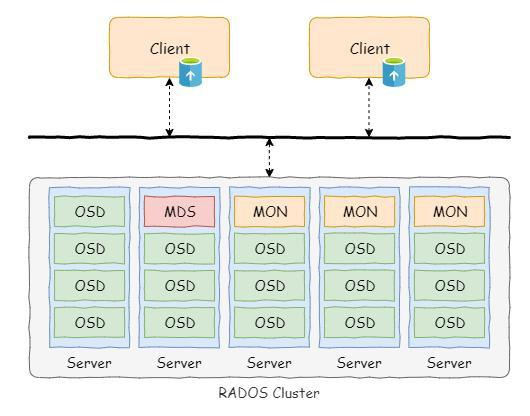


图-5

**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）所有主机设置防火墙和SELinux

1. **[**root@client **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
2. **[**root@client **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
3. **[**root@client **~]**# setenforce **0**
4. **[**root@node1 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
5. **[**root@node1 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
6. **[**root@node1 **~]**# setenforce **0**
7. **[**root@node2 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
8. **[**root@node2 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
9. **[**root@node2 **~]**# setenforce **0**
10. **[**root@node3 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
11. **[**root@node3 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
12. **[**root@node3 **~]**# setenforce **0**

2）为所有节点配置yum源服务器。

把四台虚拟机全部关机；每台虚拟机都添加一个光驱；

做如下相同操作:

右击虚拟机,选【设置】---【添加】---【CD|DVD驱动器】--【完成】；

点击刚刚新建的光盘[CD|DVD],勾选使用ISO映像文件--[浏览]；

找到自己真机的ceph10.iso加载即可。

添加磁盘：

除了客户端，所有3台ceph服务器都添加2块20G磁盘。

启动所有虚拟机后，查看磁盘情况:

1. **[**root@client **~]**# lsblk
2. **[**root@node1 **~]**# lsblk
3. **[**root@node2 **~]**# lsblk
4. **[**root@node3 **~]**# lsblk

3）所有主机挂载ceph光盘和系统光盘

【不能照抄、不能照抄、不能照抄】，需要根据实际情况操作，

案例假设是将系统光盘挂载到/media目录，将ceph光盘挂载到/ceph目录。

1. **[**root@client **~]**# umount **/**dev**/**sr0
2. **[**root@client **~]**# umount **/**dev**/**sr1     #未挂载的话会报错
3. **[**root@client **~]**# mkdir **/**ceph
4. **[**root@client **~]**# vim **/**etc**/**fstab
5. /dev/sr0 **/**ceph iso9660 defaults **0** **0** #需要根据实际情况挂载
6. /dev/sr1 **/**media iso9660 defaults **0** **0** #需要根据实际情况挂载
7. **[**root@client **~]**# mount **-**a
8. **[**root@client **~]**# lsblk
9. **[**root@node1 **~]**# umount **/**dev**/**sr0
10. **[**root@node1 **~]**# umount **/**dev**/**sr1
11. **[**root@node1 **~]**# mkdir **/**ceph
12. **[**root@node1 **~]**# vim **/**etc**/**fstab
13. /dev/sr0 **/**ceph iso9660 defaults **0** **0** #需要根据实际情况挂载
14. /dev/sr1 **/**media iso9660 defaults **0** **0** #需要根据实际情况挂载
15. **[**root@node1 **~]**# mount **-**a
16. **[**root@node2 **~]**# umount **/**dev**/**sr0
17. **[**root@node2 **~]**# umount **/**dev**/**sr1
18. **[**root@node2 **~]**# mkdir **/**ceph
19. **[**root@node2 **~]**# vim **/**etc**/**fstab
20. /dev/sr0 **/**ceph iso9660 defaults **0** **0** #需要根据实际情况挂载
21. /dev/sr1 **/**media iso9660 defaults **0** **0** #需要根据实际情况挂载
22. **[**root@node2 **~]**# mount **-**a
23. **[**root@node3 **~]**# umount **/**dev**/**sr0
24. **[**root@node3 **~]**# umount **/**dev**/**sr1
25. **[**root@node3 **~]**# mkdir **/**ceph
26. **[**root@node3 **~]**# vim **/**etc**/**fstab
27. /dev/sr0 **/**ceph iso9660 defaults **0** **0** #需要根据实际情况挂载
28. /dev/sr1 **/**media iso9660 defaults **0** **0** #需要根据实际情况挂载
29. **[**root@node3 **~]**# mount **-**a

4）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. #**-**f后面跟密钥的文件名称（希望创建密钥到哪个文件）
3. #**-**N ''代表不给密钥配置密钥（不能给密钥配置密码）
4. **[**root@node1 **~]**# **for** i **in** **10** **11** **12** **13**
5. **do**
6. ssh**-**copy**-**id **192.168.4.**$i
7. done
8. #通过ssh**-**copy**-**id将密钥传递给**192.168.4.10**、**192.168.4.11**、**192.168.4.12**、**192.168.4.13**

5）修改/etc/hosts并同步到所有主机。

注意：/etc/hosts解析的域名要与本机主机名一致！！！！

1. **[**root@node1 **~]**# vim **/**etc**/**hosts #修改文件，手动添加如下内容（不要删除文件原有内容）
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

提示：/etc/hosts解析的域名必须与本机主机名一致！！！

将/etc/hosts文件拷贝给所有其他主机（client、node1、node2、node3）

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

6）修改所有节点都需要配置YUM源，并同步到所有主机。

1. **[**root@node1 **~]**# vim **/**etc**/**yum**.**repos**.**d**/**ceph**.**repo #新建YUM源配置文件，内容如下
2. **[**mon**]**
3. name**=**mon
4. baseurl**=**file**:**///ceph/MON
5. gpgcheck**=0**
6. **[**osd**]**
7. name**=**osd
8. baseurl**=**file**:**///ceph/OSD
9. gpgcheck**=0**
10. **[**tools**]**
11. name**=**tools
12. baseurl**=**file**:**///ceph/Tools
13. gpgcheck**=0**
14. **[**root@node1 **~]**# yum clean all #清空缓存
15. **[**root@node1 **~]**# yum repolist                #验证YUM源软件数量
16. 源标识            源名称                    状态
17. Dvd                redhat                    **9,911**
18. Mon                mon                        **41**
19. Osd                osd                        **28**
20. Tools            tools                    **33**
21. repolist**:** **10,013**
22. **[**root@node1 **~]**# **for** i **in** client node1 node2 node3
23. **do**
24. scp **/**etc**/**yum**.**repos**.**d**/**ceph**.**repo $i**:**/etc/yum**.**repos**.**d**/**
25. done

7）Client主机配置NTP服务器。

1. **[**root@client **~]**# yum **-**y install chrony
2. **[**root@client **~]**# vim **/**etc**/**chrony**.**conf
3. allow **192.168.4.0/24**        #修改**26**行
4. local stratum **10**            #修改**29**行**(**去注释即可**)**
5. **[**root@client **~]**# systemctl restart chronyd

8）node1，node2，node3修改NTP客户端配置。

1. **[**root@node1 **~]**# vim **/**etc**/**chrony**.**conf
2. server **192.168.4.10** iburst #配置文件第二行，手动添加一行新内容
3. **[**root@node1 **~]**# systemctl restart chronyd
4. **[**root@node1 **~]**# chronyc sources **-**v #查看同步结果，应该是**^\***
5. **[**root@node2 **~]**# vim **/**etc**/**chrony**.**conf
6. server **192.168.4.10** iburst #配置文件第二行，手动添加一行新内容
7. **[**root@node2 **~]**# systemctl restart chronyd
8. **[**root@node2 **~]**# chronyc sources **-**v            #查看同步结果，应该是**^\***
9. **[**root@node3 **~]**# vim **/**etc**/**chrony**.**conf
10. server **192.168.4.10** iburst #配置文件第二行，手动添加一行新内容
11. **[**root@node3 **~]**# systemctl restart chronyd
12. **[**root@node3 **~]**# chronyc sources **-**v #查看同步结果，应该是**^\***

**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（目录名称可以任意，推荐与案例一致）

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（ceph.conf）。

***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。

拷贝当前目录的配置文件到所有节点的/etc/ceph/目录并启动mon服务。

1. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy mon create**-**initial
2. #配置文件ceph**.**conf中有三个mon的IP，ceph**-**deploy脚本知道自己应该远程谁

4) 在每个node主机查看自己的服务(注意每台主机服务名称不同)

1. **[**root@node1 ceph**-**cluster**]**# systemctl status ceph**-**mon@node1
2. **[**root@node2 **~]**# systemctl status ceph**-**mon@node2
3. **[**root@node3 **~]**# systemctl status ceph**-**mon@node3
4. #备注**:**管理员可以自己启动（start）、重启（restart）、关闭（stop），查看状态（status）**.**
5. #提醒**:**这些服务在**30**分钟只能启动**3**次**,**超过就报错**.**
6. #StartLimitInterval**=**30min
7. #StartLimitBurst**=3**
8. #在这个文件中有定义/usr/lib**/**systemd**/**system**/**ceph**-**mon@**.**service
9. #如果修改该文件，需要执行命令# systemctl daemon**-**reload重新加载配置

5）查看ceph集群状态（现在状态应该是health HEALTH\_ERR）

1. **[**root@node1 ceph**-**cluster**]**# ceph **-**s

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：（如何无法修复说明环境准备有问题，需要重置所有虚拟机）

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

解决方案如下（仅在node1操作）：

1）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**

2）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
2. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy **--**overwrite**-**conf mon create**-**initial

3）如果还出错，可能是准备实验环境时配置的域名解析和主机名不一致！！！

**步骤三：创建OSD**

1) 初始化清空磁盘数据（仅node1操作即可）。

初始化磁盘，将所有磁盘分区格式设置为GPT格式（根据实际情况填写磁盘名称）。

1. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy disk zap node1**:**sdb node1**:**sdc
2. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy disk zap node2**:**sdb node2**:**sdc
3. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy disk zap node3**:**sdb node3**:**sdc
4. #相当于ssh 远程node1，在node1执行parted **/**dev**/**sdb mktable gpt
5. #其他主机都是一样的操作
6. #ceph**-**deploy是个脚本，这个脚本会自动ssh远程自动创建gpt分区

思考题？

1. # vim test**.**sh
2. #**!**/bin/bash
3. **case** $**1** **in**
4. user**)**
5. useradd **-**u **1000** $**2;;**
6. disk**)**
7. partd **/**dev**/**$**2** mktable gpt**;;**
8. esac
9. # chmod **+**x test**.**sh
10. # **./**test**.**sh user jerry
11. # **./**test**.**sh disk sdc

执行上面的脚本没有指定账户UID，为什么会自动创建一个UID为1000的用户？

执行上面的脚本没有指定磁盘分区表类型，为什么创建的分区表类型为gpt类型？

上面的脚本如果执行时不给位置变量的参数为怎么样？

2）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！！！

远程所有node主机，创建分区，格式化磁盘，挂载磁盘，启动osd服务共享磁盘。

1. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy osd create node1**:**sdb node1**:**sdc
2. #每个磁盘都会被自动分成两个分区；一个固定5G大小；一个为剩余所有容量
3. #5G分区为Journal日志缓存；剩余所有空间为数据盘。
4. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy osd create node2**:**sdb node2**:**sdc
5. **[**root@node1 ceph**-**cluster**]**# ceph**-**deploy osd create node3**:**sdb node3**:**sdc

提醒：ceph-deploy是个脚本，脚本会自动创建分区、格式化、挂载！

怎么验证分区了？怎么验证格式化？怎么验证挂载了？

1. **[**root@node1 **~]**# df **-**Th
2. **[**root@node2 **~]**# df **-**Th
3. **[**root@node3 **~]**# df **-**Th

思考题：请问lsblk和df命令的区别？

3）在三台不同的主机查看OSD服务状态，可以开启、关闭、重启服务。

1. **[**root@node1 **~]**# systemctl status ceph**-**osd@**0**
2. **[**root@node2 **~]**# systemctl status ceph**-**osd@**2**
3. **[**root@node3 **~]**# systemctl status ceph**-**osd@**4**
4. #备注**:**管理员可以自己启动（start）、重启（restart）、关闭（stop），查看状态（status）**.**
5. #提醒**:**这些服务在**30**分钟只能启动**3**次**,**超过就报错**.**
6. #StartLimitInterval**=**30min
7. #StartLimitBurst**=3**
8. #在这个文件中有定义/usr/lib**/**systemd**/**system**/**ceph**-**osd@**.**service
9. #如果修改该文件，需要执行命令# systemctl daemon**-**reload重新加载配置

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. **[**root@node1 **~]**# ceph osd tree

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，重启所有ceph服务：

1. **[**root@node1 **~]**# systemctl restart 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
* 删除镜像

**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，默认存储池名称为rbd。

1. **[**root@node1 **~]**# ceph osd lspools
2. **0** rbd**,**
3. #查看结果显示，共享池的名称为rbd，这个共享池的编号为**0**，英语词汇：pool（池塘、水塘）

2）创建镜像、查看镜像

1. **[**root@node1 **~]**# rbd create demo**-**image **--**image**-**feature layering **--**size 10G
2. #创建demo**-**image镜像，这里的demo**-**image创建的镜像名称，名称可以为任意字符。
3. #size可以指定镜像大小
4. **[**root@node1 **~]**# rbd create rbd**/**jacob **--**image**-**feature layering **--**size 10G
5. #在rbd池中创建名称为jacob的镜像（rbd**/**jacob），镜像名称可以任意

#--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 #列出所有镜像
2. **[**root@node1 **~]**# rbd info demo**-**image #查看demo**-**image这个镜像的详细信息
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. #调整jacob镜像的大小，jacob是镜像的名称，size指定扩容到15G
3. **[**root@node1 **~]**# rbd info jacob

2）缩小容量

1. **[**root@node1 **~]**# rbd resize **--**size 7G jacob **--**allow**-**shrink
2. #英文词汇：allow（允许），shrink（缩小）
3. **[**root@node1 **~]**# rbd info jacob
4. #查看jacob这个镜像的详细信息（jacob是前面创建的镜像）

**步骤三：通过KRBD访问**

Linux内核可用直接访问Ceph块存储，KVM可用借助于librbd访问Ceph块存储。

客户端访问结构如图-6所示。

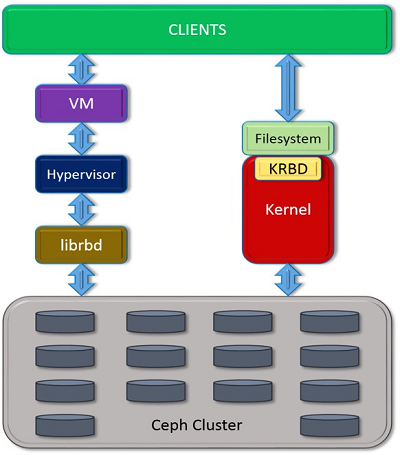


图-6

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 #客户端访问映射服务器的jacob共享镜像
9. **[**root@client **~]**# lsblk #查看结果（会多一块磁盘）
10. **[**root@client **~]**# rbd showmapped #查看磁盘名和共享镜像名称的对应关系
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 #格式化，格式为xfs
2. **[**root@client **~]**# mount **/**dev**/**rbd0 **/**mnt**/** #挂载（可以挂载到任意目录）
3. **[**root@client **~]**# echo "test" **>** /mnt/test**.**txt #写入数据

**步骤四：删除镜像**

1） 客户端撤销磁盘映射

1. **[**root@client **~]**# umount **/**mnt #卸载
2. **[**root@client **~]**# rbd showmapped #查看磁盘名和共享镜像名称的对应关系
3. id pool image snap device
4. **0** rbd jacob **-** /dev/rbd0
5. **[**root@client **~]**# rbd unmap **/**dev**/**rbd0 #撤销磁盘映射

附加信息：Ceph操作思路（知识总结）

一、准备工作：

IP，主机名，hosts解析，ssh密钥，时间同步，yum源，防火墙，selinux

二、部署ceph：

1.安装软件

1. ceph**-**deploy**(**脚本**)**
2. ceph**-**mon ceph**-**osd ceph**-**mds ceph**-**radosgw**(**集群**)**

2.修改配置启动服务mon

1. mkdir 目录；cd 目录
2. ceph**-**deploy **new** node1 node2 node3 **(**生成配置文件**)**
3. ceph**-**deploy mon create**-**initial **(**拷贝配置文件并启动mon服务**)**

3.启动osd服务共享硬盘

1. ceph**-**deploy disk zap 主机名**:**磁盘名 **...** **...**
2. ceph**-**deploy osd create 主机名**:**磁盘 **...** **...**

三、使用Ceph的思路:

1.块共享

1. 服务器**:** rbd create 创建一个共享镜像
2. 客户端**:** 安装cpeh**-**common**;** cp 配置文件和密钥
3. rbd map **|** rbd unmap

附加知识（如何删除某个OSD，下面的假设是删除osd.4）

1. ceph osd tree
2. ceph osd out osd**.4**
3. ceph osd tree
4. ceph **-**s
5. ceph osd crush remove osd**.4**
6. ceph auth del osd**.4**
7. ceph **-**s
8. ceph osd rm osd**.4**
9. 最后要找到对应的主机，umount把osd**.4**对应的磁盘卸载