

集群与存储

NSD CLUSTER

DAY04

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	ceph概述
	10:30 ~ 11:20	
	11:30 ~ 12:20	部署Ceph集群
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	Ceph块存储
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



ceph概述

ceph概述

基础知识

什么是分布式文件系统

常用分布式文件系统

什么是ceph

Ceph组件

实验环境准备

实验拓扑图

配置YUM

附加组件

RHCS运行原理

基础知识

什么是分布式文件系统

- 分布式文件系统 (Distributed File System) 是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连
- 分布式文件系统的设计基于客户机/服务器模式



常用分布式文件系统

- Lustre
- Hadoop
- FastDFS
- Ceph
- GlusterFS



什么是ceph

- ceph是一个分布式文件系统
- 具有高扩展、高可用、高性能的特点
- ceph可以提供对象存储、块存储、文件系统存储
- ceph可以提供PB级别的存储空间(PB→TB→GB)
 - $1024G \times 1024G = 1048576G$
- 软件定义存储(Software Defined Storage)作为存储行业的一大发展趋势，已经越来越受到市场的认可



ceph组件

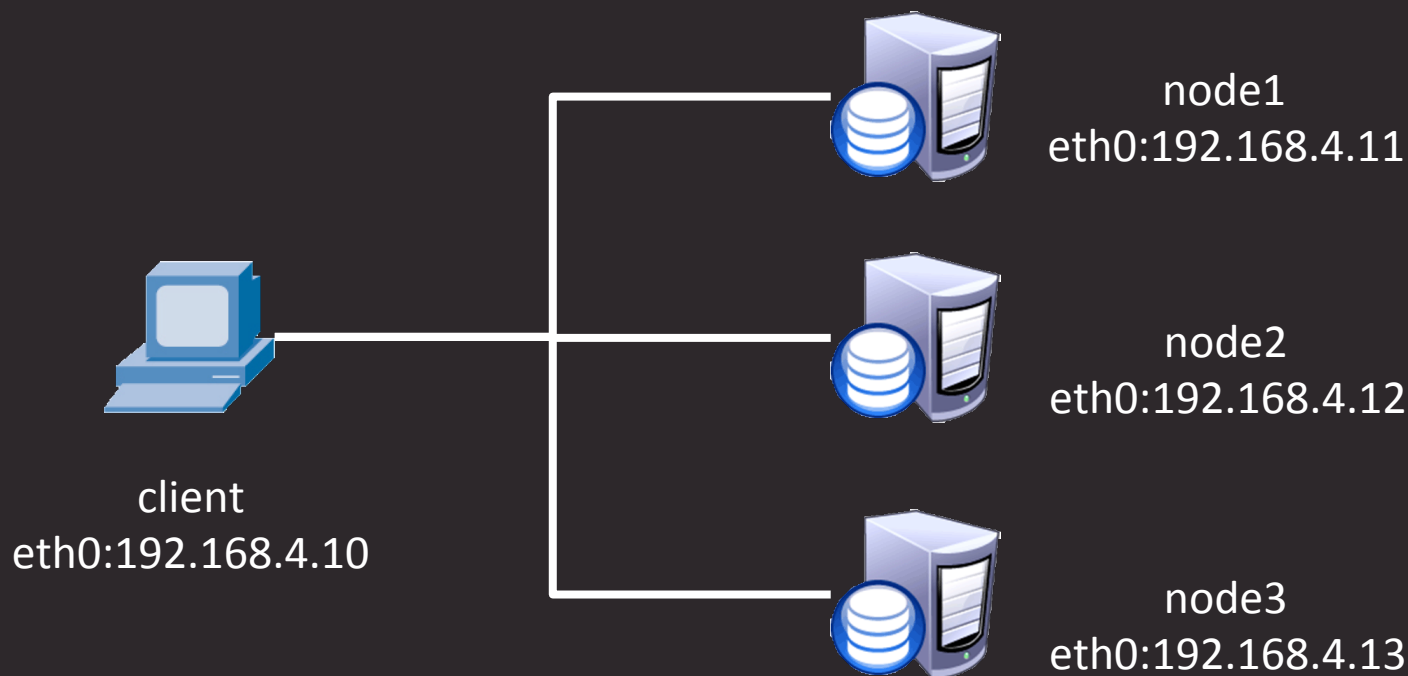
- OSDs
 - 存储设备
- Monitors
 - 集群监控组件
- MDSs
 - 存放文件系统的元数据（对象存储和块存储不需要该组件）
- Client
 - ceph客户端



实验环境准备

实验拓扑图

- 1台客户端虚拟机
- 3台存储集群虚拟机



配置YUM

- 物理机创建网络yum源服务器

```
[root@root9pc01 ~]# yum -y install vsftpd  
[root@root9pc01 ~]# mkdir /var/ftp/ceph  
[root@root9pc01 ~]# mount -o loop \  
rhcs2.0-rhosp9-20161113-x86_64.iso /var/ftp/ceph  
[root@root9pc01 ~]# systemctl restart vsftpd
```



配置YUM（续1）

- 虚拟机调用YUM源（下面以node1为例）

```
[root@node1 ~]# cat /etc/yum.repos.d/ceph.repo
[mon]
name=mon
baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86_64/MON
gpgcheck=0
[osd]
name=osd
baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86_64/OSD
gpgcheck=0
[tools]
name=tools
baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86_64/Tools
gpgcheck=0
```



配置SSH无密钥连接

- 修改主机名

```
[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
```

```
[root@node1 ~]# for i in 10 11 12 13
```

```
> do
```

```
> scp /etc/hosts 192.168.2.$i:/etc/
```

```
> done
```



配置SSH无密钥连接（续1）

- 非交互生成密钥对

```
[root@node1 ~]# ssh-keygen -f /root/.ssh/id_rsa -N "
```

- 发布密钥到各个主机（包括自己）

```
[root@node1 ~]# for i in 10 11 12 13
> do
> ssh-copy-id 192.168.4.$i
> done
```



NTP时间同步

- 客户端创建NTP服务器

```
[root@client ~]# yum -y install chrony
```

```
[root@client ~]# cat /etc/chrony.conf
```

```
server 0.centos.pool.ntp.org iburst
```

```
allow 192.168.4.0/24
```

```
local stratum 10
```

```
[root@client ~]# systemctl restart chronyd
```

- 其他所有主机与其同步时间（下面以node1为例）

```
[root@node1 ~]# cat /etc/chrony.conf
```

```
server 192.168.4.10 iburst
```

```
[root@node1 ~]# systemctl restart chronyd
```



准备存储磁盘

- 物理机上为每个虚拟机创建3个磁盘

```
[root@root9pc01 ~]# cd /var/lib/libvirt/images
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node1-vdb.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node1-vdc.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node1-vdd.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node2-vdb.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node2-vdc.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node2-vdd.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node3-vdb.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node3-vdc.vol 10G
```

```
[root@root9pc01 ~]# qemu-img create -f qcow2 node3-vdd.vol 10G
```

- 在图形环境中为虚拟机添加磁盘

```
[root@root9pc01 ~]# virt-manager
```

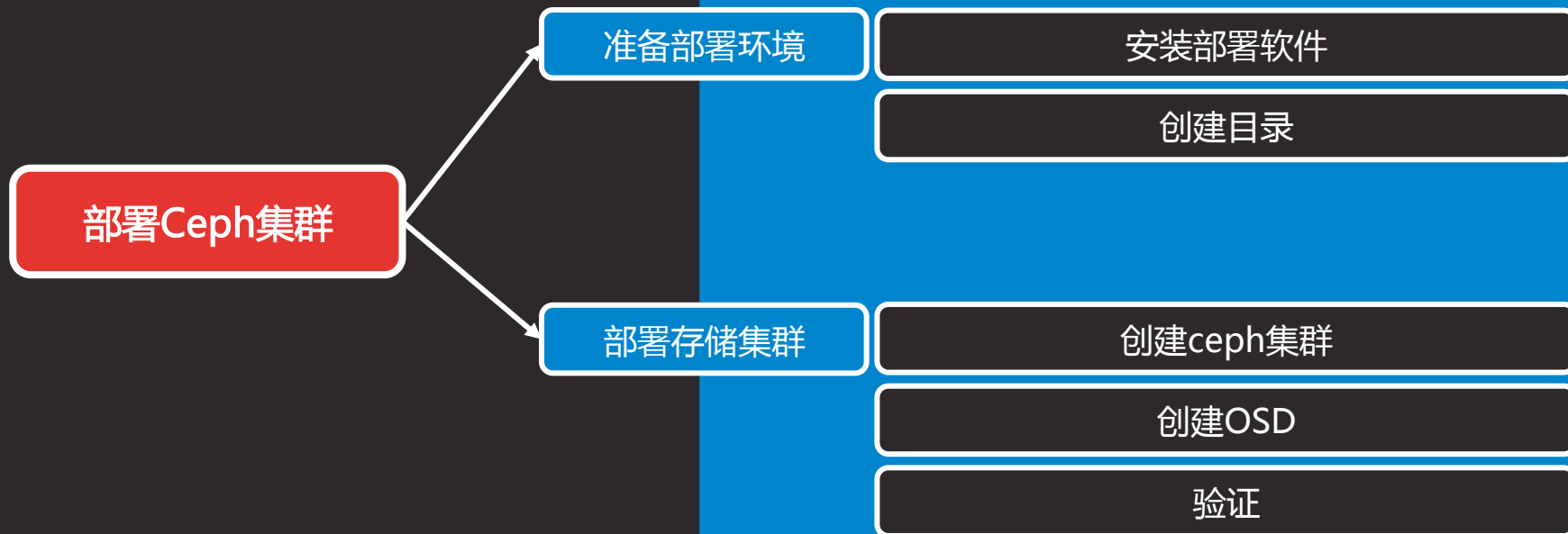


案例1：实验环境

- 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



部署Ceph集群



准备部署环境

安装部署软件

- 使用node1作为部署主机

```
[root@node1 ~]# yum -y install ceph-deploy
```

- ceph-deploy命令与子命令都支持--help查看帮助

```
[root@node1 ~]# ceph-deploy --help
```



创建目录

- 为部署工具创建目录，存放密钥与配置文件

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



部署存储集群

创建Ceph集群

- 创建ceph集群配置（所有节点都为mon）

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

- 给所有节点安装ceph软件包

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

- 初始化所有节点的mon服务（主机名解析必须对）

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

//这里没有指定主机，是因为第一步创建的配置文件中已经有了，//所以要求主机名解析必须对，否则连接不到对应的主机



创建OSD

- 所有节点准备磁盘分区（下面以node1为例）

```
[root@node1 ~]# parted /dev/vdb mklabel gpt
```

```
[root@node1 ~]# parted /dev/vdb mkpart primary 1M 50%
```

```
[root@node1 ~]# parted /dev/vdb mkpart primary 50% 100%
```

```
[root@node1 ~]# chown ceph.ceph /dev/vdb1
```

```
[root@node1 ~]# chown ceph.ceph /dev/vdb2
```

//这两个分区用来做存储服务器的日志journal盘



创建OSD (续1)

- 初始化清空磁盘数据 (仅node1操作即可)

```
[root@node1 ~]# ceph-deploy disk zap node1:vdc node1:vdd
```

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

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

- 创建OSD存储空间 (仅node1操作即可)

```
[root@node1 ~]# 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日志，一个存储设备对应一个日志设备，日志需要SSD，不需要很大

```
[root@node1 ~]# ceph-deploy osd create node2:vdc:/dev/vdb1 node2:vdd:/dev/vdb2
```

```
[root@node1 ~]# ceph-deploy osd create node3:vdc:/dev/vdb1 node3:vdd:/dev/vdb2
```



验证

- 查看集群状态

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

- 可能出现的错误

- osd create创建OSD存储空间，如提示run
'gatherkeys'

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

- ceph -s查看状态，如果失败

```
[root@node1 ~]# systemctl restart ceph\*.service ceph\*.target  
//在所有节点，或仅在失败的节点重启服务
```

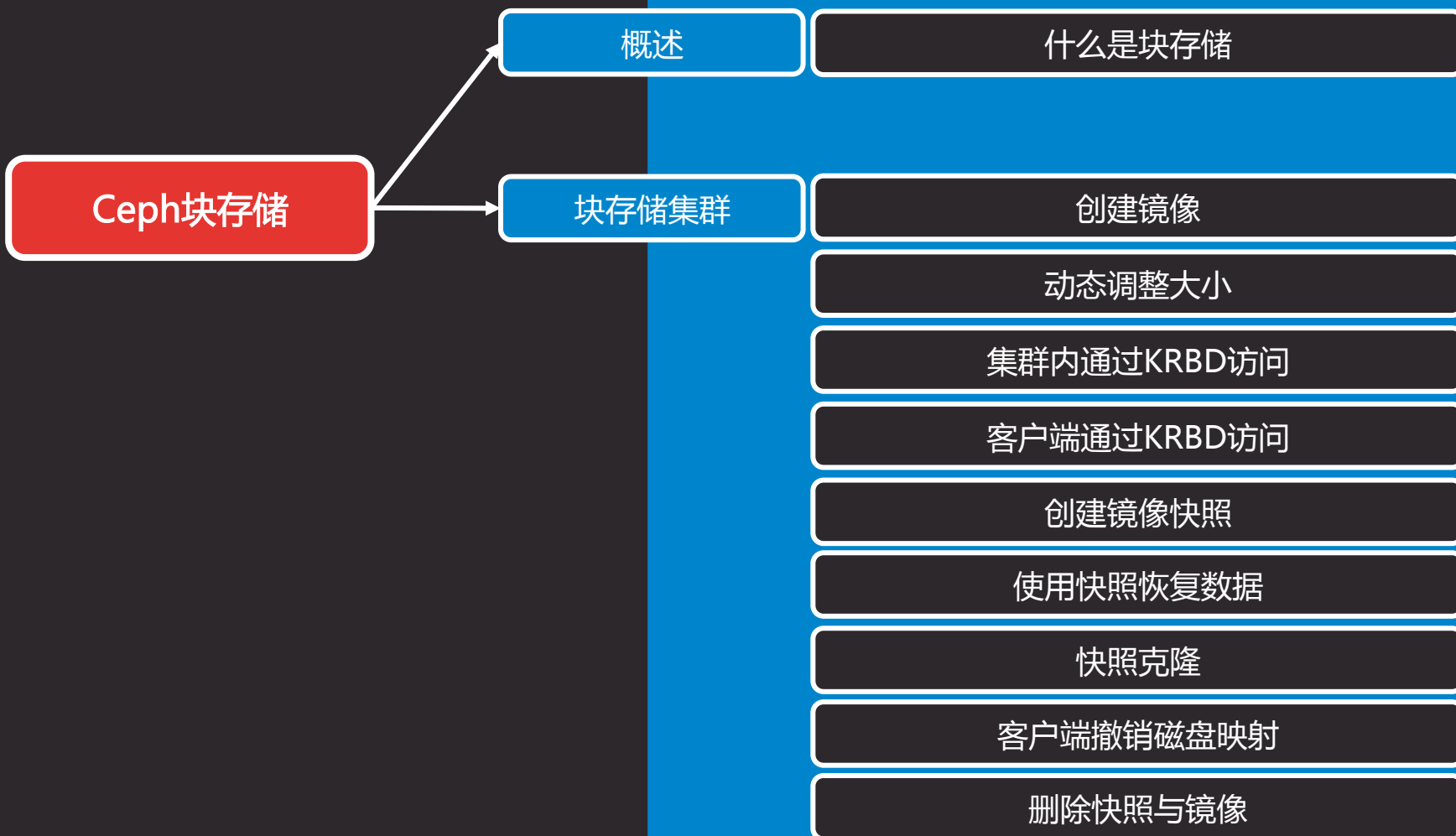


案例2：部署ceph集群

- 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



Ceph块存储



概述

什么是块存储

- 单机块设备
 - 光盘
 - 磁盘
- 分布式块存储
 - Ceph
 - Cinder



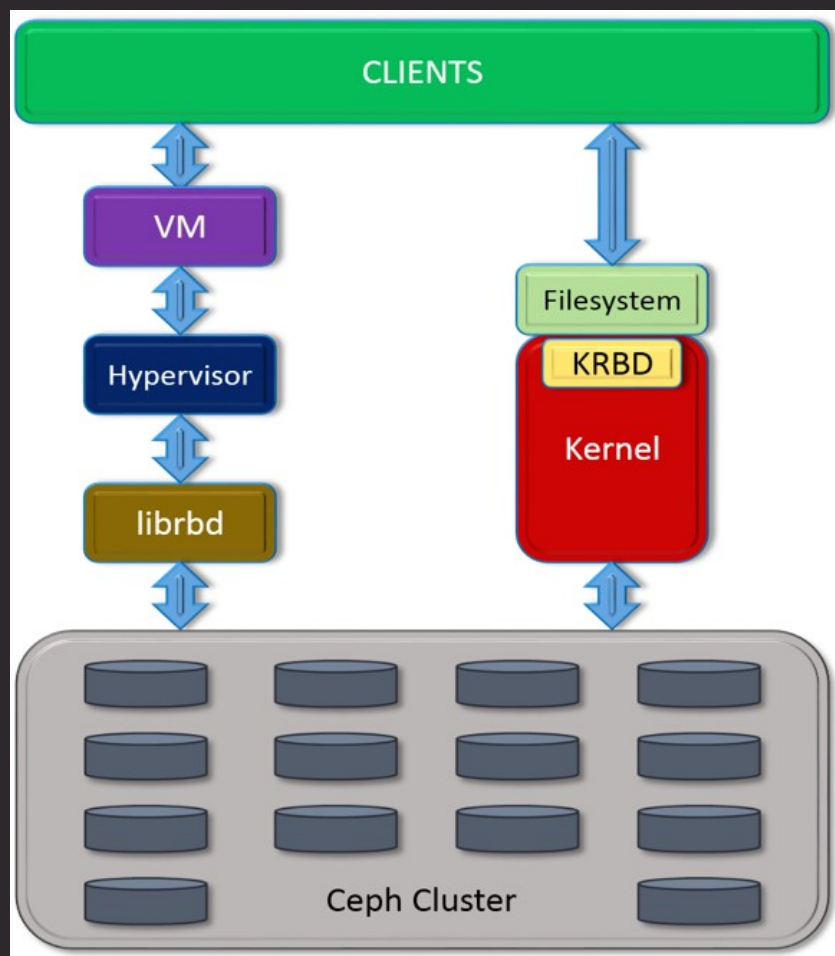
什么是块存储（续1）

- Ceph块设备也叫做RADOS块设备
 - RADOS block device : **RBD**
- RBD驱动已经很好的集成在了Linux内核中
- RBD提供了企业功能，如快照、COW克隆等等
- RBD还支持内存缓存，从而能够大大提高性能



什么是块存储（续2）

- Linux内核可用直接访问Ceph块存储
- KVM可用借助于librbid访问



块存储集群



创建镜像

- 查看存储池（默认有一个rbd池）

```
[root@node1 ~]# ceph osd lspools
```

```
0 rbd,
```

- 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
```

```
[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
```



动态调整大小

- 缩小容量

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

- 扩容容量

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



集群内通过KRBD访问

- 将镜像映射为本地磁盘

```
[root@node1 ~]# rbd map demo-image
/dev/rbd0
[root@node1 ~]# lsblk
... ..
rbd0      251:0   0  10G  0 disk
```

- 接下来，格式化了！

```
[root@node1 ~]# mkfs.xfs /dev/rbd0
[root@node1 ~]# mount /dev/rbd0 /mnt
```



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/
```

- 映射镜像到本地磁盘

```
[root@client ~]# rbd map image
```

```
[root@client ~]# lsblk
```

```
[root@client ~]# rbd showmapped
```

```
id pool image snap device
```

```
0 rbd image - /dev/rbd0
```

客户端通过KRBD访问（续1）

- 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
```



创建镜像快照

- 查看镜像快照

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

- 创建镜像快照

```
[root@node1 ~]# rbd snap create image --snap image-snap1
```

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

```
SNAPID NAME      SIZE
4 image-snap1 15360 MB
```

- 注意：快照使用COW技术，对大数据快照速度会很快！



使用快照恢复数据

- 删除客户端写入的测试文件

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

- 还原快照

```
[root@node1 ~]# rbd snap rollback image --snap image-snap1
```

- 客户端重新挂载分区

```
[root@client ~]# umount /mnt
```

```
[root@client ~]# mount /dev/rbd0 /mnt/
```

```
[root@client ~]# ls /mnt
```



快照克隆

- 如果想从快照恢复出来一个新的镜像，则可以使用克隆
- 注意，克隆前，需要对快照进行<保护>操作
- 被保护的快照无法删除，取消保护(unprotect)

```
[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image-clone镜像



快照克隆（续1）

- 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
```



快照克隆（续2）

- 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，但比较耗时！！！！

```
[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
    format: 2
    features: layering
    flags:
```

//注意，父快照信息没了！



客户端撤销磁盘映射

- umount挂载点

```
[root@client ~]# umount /mnt
```

- 取消RBD磁盘映射

```
[root@client ~]# rbd showmapped
```

```
id pool image      snap device
0 rbd image      - /dev/rbd0
```

//语法格式:

```
[root@client ~]# rbd unmap /dev/rbd/{poolname}/{imagename}
```

```
[root@client ~]# rbd unmap /dev/rbd/rbd/image
```



删除快照与镜像

- 删除快照（确保快照未被保护）

```
[root@node1 ~]# rbd snap rm image --snap image-snap
```

- 删除镜像

```
[root@node1 ~]# rbd list
```

```
[root@node1 ~]# rbd rm image
```



案例3：创建Ceph块存储

- 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



总结和答疑
