配置分布式存储服务

1. Ceph

概述

* 1. 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
* 分布式文件系统的实际基于客户机/服务器模式

b） 常用分布式文件系统

Lustre Hadoop（大数据） FastDFS Ceph GlusterFS

c） 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\*1024G=1048576G

* 软件定义存储（Software Defined Storage）作为存储行业的一大发展趋势，已经越来越收到市场的认可

d) Ceph组件

OSDs —— 存储设备

Monitors —— 集群监控组件

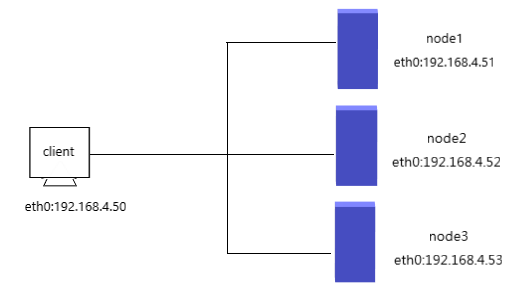
MDSs —— 存放文件系统的元数据（对象存储和块存储不需要改组件）

Client —— ceph客户端

二、实验环境准备（新建4台机器）

实验拓扑图

1台客户端虚拟机

3台存储集群虚拟机

|  |  |
| --- | --- |
| client | eth0:192.168.4.50 |
| node1 | eth0:192.168.4.51 |
| node2 | eth0:192.168.4.52 |
| node3 | eth0:192.168.4.53 |

1. 配置yum源（在宿主机上挂载光盘rhcs2.0-rhosp9-20161113-x86\_64.iso到/var/ftp/ceph）

书写开机自动挂载配置文件：/etc/fstab

#vim /etc/fstab

/var/lib/libvirt/images/iso/rhcs2.0-rhosp9-20161113-x86\_64.iso /var/ftp/ceph iso9660 defaults 0 0

或者

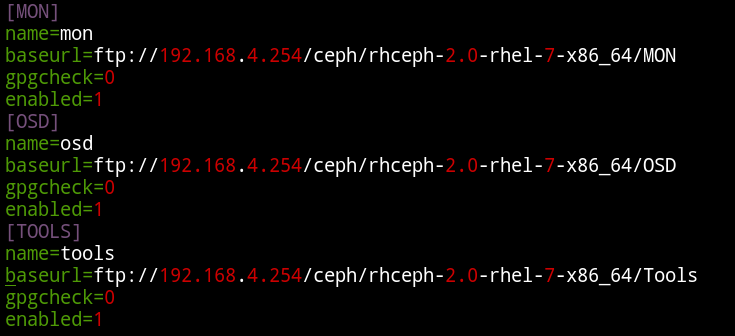
# vim /etc/rc.local（追加下面的内容）

mount -o loop /var/lib/libvirt/images/iso/rhcs2.0-rhosp9-20161113-x86\_64.iso /var/ftp/ceph/

# chmod +x /etc/rc.d/rc.local

1. 书写四台虚拟机的yum配置文件

# vim /etc/yum.repos.d/ceph.repo



# yum clean all

# yum repolist

1. 分别永久设置四台虚拟机的主机名（与之前的实验拓扑中的表格相对应）

# hostnamectl set-hostname client

# hostnamectl set-hostname node1

# hostnamectl set-hostname node2

# hostnamectl set-hostname node3

4. 修改域名解析配置文件

# vim /etc/hosts（追加以下内容）

192.168.4.50 client

192.168.4.51 node1

192.168.4.52 node2

192.168.4.53 node3

5. 配置SSH无密码连接

* + - * 1. 主机51上生成密钥对

# ssh-keygen -f /root/.ssh/id\_rsa -N ''（非交互式生成密钥对）

b) 发布密钥到各个主机（包括自己）

# for i in 50 51 52 53

> do

> ssh-copy-id 192.168.4.$i

> done

6. NTP时间同步

1. 客户端创建NTP服务器（client）

# yum -y install chrony

# vim /etc/chrony.conf

server centos.pool.ntp.org iburst

allow 192.168.4.0/24

local stratum 10

1. 其他所有主机与其同步时间

# vim /etc/chrony.conf

server 192.168.4.50 iburst

1. 所有主机重起chronyd服务

# systemctl restart chronyd

# systemctl enable chronyd

7. 准备存储磁盘

1. 在图形环境中为虚拟机添加磁盘

# virt-manager

1. 宿主机上为每个虚拟机创建3个磁盘（node1-node2）

# cd /var/lib/libvirt/images

# qemu-img create -f qcow2 node1-vdb.vol 10G

................................................................................................

三、部署CEPH集群

A： 创建Ceph集群

node1主机：

1. 使用node1作为部署主机，在node1上安装集群管理软件

# yum -y install ceph-deploy

ceph-deploy命令与子命令都支持--help查看帮助

2. 创建目录（为部署工具创建目录，存放密钥与配置文件）

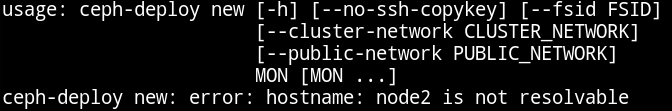
# mkdir /root/ceph-cluster

# cd /root/ceph-cluster/

1. 创建Ceph集群（在/root/ceph-cluster目录下）
2. 创建Ceph集群配置（所有节点都为mon）

# ceph-deploy new node1 node2 node3

若出现以下错误：



则查看/etc/hosts中的地址解析对应的主机名是否与命令行中输入的主机名一致

1. 给所有节点安装Ceph软件包

# ceph-deploy install node1 node2 node3

1. 初始化（启动）所有节点的mon服务（主机名解析必须对）

# 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

B： 创建OSD集群（node1\node2\node3）,以node1为例：

1. 所有节点准备磁盘分区（例如使用vdb来做存储服务器的日志盘）

# parted /dev/vdb mklabel gpt

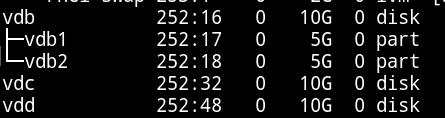
# parted /dev/vdb mkpart primary 1M 50%

# parted /dev/vdb mkpart primary 50% 100%

//有几个存储盘就分几个日志盘

在node2和node3上重复此操作

#lsblk （分区成功后查看输出会包含下面的内容）



如果没有出现则使用partprobe刷新之后再次进行查看

1. 修改/dev/vdb\*的属主与属组（同样是三台存储服务器都要修改），并且使其永久有效

# chown ceph.ceph /dev/vdb\*

# echo "chown ceph.ceph /dev/vdb\*" >> /etc/rc.local

# chmod +x /etc/rc.d/rc.local

1. 初始化存储盘vdc和vdd（在管理主机node1上进行）

# cd /root/ceph-cluster/

# ceph-deploy disk zap node1:vdc node1:vdd

# ceph-deploy disk zap node2:vdc node2:vdd

# ceph-deploy disk zap node3:vdc node3:vdd

1. 创建OSD设备（在管理主机node1上进行）

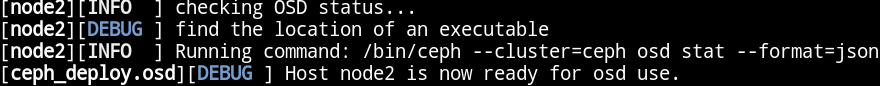
# ceph-deploy osd create node1:vdc:/dev/vdb1 node1:vdd:/dev/vdb2

# ceph-deploy osd create node2:vdc:/dev/vdb1 node2:vdd:/dev/vdb2

# ceph-deploy osd create node3:vdc:/dev/vdb1 node3:vdd:/dev/vdb2

//创建osd存储设备，vdc为集群提供存储空间，vdb1提供日志，一个存储设备对应一个日志设备，日志需要SSD，不需要很大

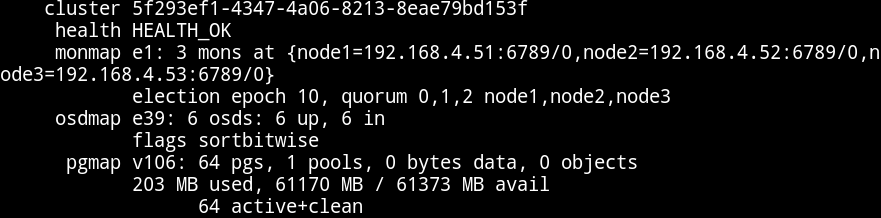
注：在执行过程中只要不出现error就没问题，出现warnin忽略即可，执行完成后出现类似下面的信息即可



不同主机显示的主机名不同

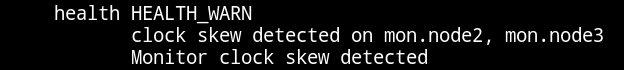
1. 验证
   1. 查看集群状态

# ceph -s



可能出现的错误及解决方法：

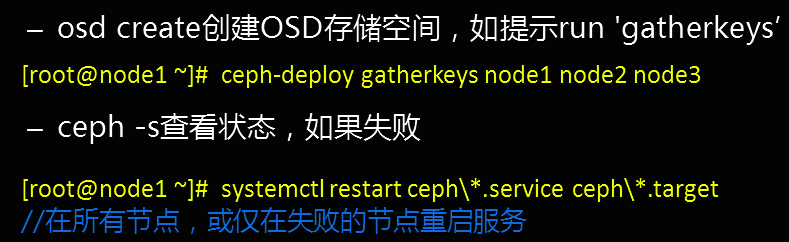
错误1：

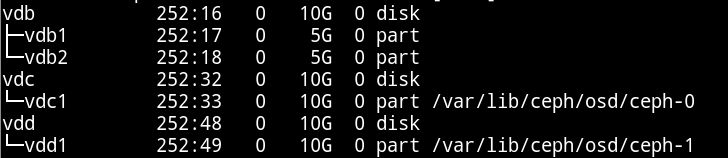


问题 ：——clock skew（时间不同步）

解决方法： 将所有主机的时间都使用NTP时间同步

错误2：



以上操作全部完成并且ceph -s查看状态为OK的时候，使用lsblk命令会包含以下输出： 

三台注机上现实的内容略有不同

1. CSPH块存储

A： 概述：

1. 单机块设备

— 光盘

— 磁盘

1. 分布式块存储

— Ceph

— Cinder

1. Ceph块设备也叫做RADOS块设备

* RADOS block device ：RBD

2． RBD驱动已经很好的集成在了Linux内核中

3. RBD提供了企业功能，如快照、COW克隆等等

4. RBD还支持内存缓存，从而能够大大提高性能

5. Linux内核可以直接访问Ceph块存储

6. KVM可用借助于librbd访问

B： 配置块存储集群（node1）

1. 查看存储池（默认有一个rbd池）

# ceph osd lspools

1. 创建镜像、查看镜像

创建镜像的格式：rbd create 指定位置/镜像名称 --image-feature(创建镜像的方式) 分层 大小

# rbd create demo-image --image-feature layering --size 10G

# rbd create rbd/image --image-feature layering --size 15G

查看镜像列表

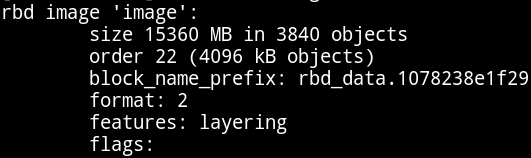
# rbd list

demo-image

image

查看镜像的详细信息

# rbd info demo-image（输出如下图所示）



删除镜像

# rbd remove image

1. 动态调整大小（缩小到/扩大到）
   1. 缩小容量

# rbd resize --size 1G image --allow-shrink

# rbd info image

* 1. 扩容容量

# rbd resize --size 15G image

# rbd info image

C： 使用镜像设备

本机rode1:

1. 将镜像映射为本地磁盘

# rbd map demo-image

1. 查看映射到本机的精细那个设备文件命令

# rbd showmapped

1. 创建挂载目录

# mkdir /notedir

1. 格式化（使用fdisk或者parted）
2. 查看格式化信息（blkid）
3. 挂载

# mount /dev/rbd0 /notedir

1. 存储文件

# echo 123 > /notedir/a.txt

客户端client：通过KRBD访问

1. 安装软件包

# yum -y install ceph-common.x86\_64

# ls /etc/ceph/

rbdmap

2. 从管理主机上拷贝集群配置文件（否则不知道集群在哪）

# scp 192.168.4.51:/etc/ceph/ceph.conf /etc/ceph

3. 从管理主机上拷贝连接集群密钥文件（否则无连接权限）

# scp 192.168.4.51:/etc/ceph/ceph.client.admin.keyring /etc/ceph/

4. 将镜像映射为本地磁盘

# rbd map image

5. 查看映射到本机的精细那个设备文件命令

6. 创建挂载目录

# mkdir /notedir

7. 格式化（使用fdisk或者parted）

1. 查看格式化信息（blkid）
2. 挂载

# mount /dev/rbd0 /studayup

1. 存储文件

# echo 123456 > /studayup/test.txt

D: 在管理主机上给被挂载的镜像文件创建镜像快照（使用COW技术——Copy Online Write）

1. 查看image镜像的快照（node1）

# rbd snap ls image

2. 给image镜像创建快照 名称叫 image-snap1（node1）

# rbd snap create image --snap image-snap1

1. 使用快照恢复误删除的文件（client）

# rm -rf /studayup/test.txt

# cat /studayup/test.txt

cat: /studayup/test.txt: 没有那个文件或目录

1. 在管理主机上使用快照恢复误删除的文件

# rbd snap rollback image --snap image-snap1

1. 客户端卸载当前挂载的image镜像后再次挂载image就可看到恢复的文件

# umount /studayup/

# mount /dev/rbd0 /studayup/

# cat /studayup/test.txt （会有你写入的数据）

注：快照使用COW技术，对大数据快照速度会很快

E： 快照克隆

概述：

— 如果想从快照恢复出来一个新的镜像，则可以使用克隆

— 要注意，在克隆前，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）

— 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

1. 在管理主机（node1）上，对image镜像的快照做克隆 名称叫image-clone

a)对快照image-snap1进行写保护

# rbd snap protect image --snap image-snap1

b)尝试删除镜像会失败

# rbd snap rm image --snap image-snap1

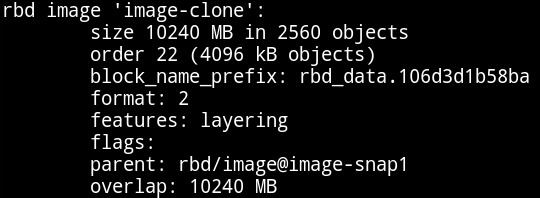
c)对image镜像的快照image-snap1 做克隆 名称叫image-clone

# 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

d)查看文件的信息

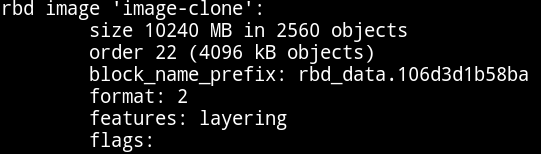
# rbd info image-clone（显示克隆文件对应的镜像信息）



2.使用克隆文件恢复image镜像

# rbd flatten image-clone

# rbd info image-clone（没有克隆文件的对应的镜像信息 ）



F：客户端不使用ceph集群的块设备存储数据的配置步骤

撤销镜像映射（卸载挂载的/dev/rbdx设备）

1. 客户端撤销磁盘映射（client）

# umount /studayup/

# rbd showmapped（有以下输出）

id pool image snap device

0 rbd image - /dev/rbd0

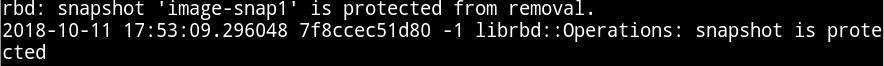
# rbd unmap /dev/rbd/rbd/image

# rbd showmapped（无输出）

1. 删除快照与镜像

在管理主机上删除创建的镜像（如果镜像有快照，要先删除快照文件）

# rbd snap rm image --snap image-snap1



出现以上输出时，是由于对快照image-snap1进行保护了

取消对快照的保护

# rbd snap unprotect image --snap image-snap1

# rbd snap rm image --snap image-snap1 //删除快照

# rbd list //查看镜像列表

# rbd rm image //删除镜像