1.DAS:直连存储(direct attach storage)

接口:IDE(现很少使用),SATA(家用电脑),SAS(服务器)

2.NAS:网络附加存储

例:nfs,http,Samba,ftp,共享的是文件系统,客户端可以直接使用

3.SAN:存储区域网络

例:iscsi,共享的是块设备,客户端需要格式化文件系统才能使用

4.分布式文件系统:文件系统管理的物理存储资源是通过计算机网络与节点相连

DAS扩展性差,空间很容易用完, 而NAS和SAN提供的空间是独立的,这时就要用到**分布式存储**,不仅容量得到提升,性能也有很大的提升

常用的分布式文件系统

Hadoop 大数据使用较多

FastDFS 国产

**Ceph 最佳,标杆**

GlusterFS

Ceph

分布式文件系统

高扩展,高可用,高性能

可以提供对象存储,块存储,文件系统存储

可以提供EB级别的存储空间(EB>PB>TB>GB)1024G\*1024G\*1024G

阿里2011年去IOE活动,IBM(做服务器) Oralce(做数据库) EMC(做存储)

Ceph组件(默认块存储)

OSDs :存储设备

Monitors:集群监控组件(中心节点),至少3台起步,确保高可用

Ceph默认3副本(至少3台OSD),大文件自动切割小文件,小文件并发写入,提高效率,任何一个OSD损坏可以无影响

MDS :提供文件系统

RGW: 提供对象存储

一 安装前准备

1.所有虚拟机磁盘扩容

2.Node1操作,为所有节点配置无密码连接

3.所有节点修改/etc/hosts,解析域名

4.所有节点配置yum源

Baseurl=ftp://192.168.4.254/ceph/MON,OSD,Tools

1. 所有节点主机与真机的NTP服务器同步时间
2. 每个虚拟机准备3块磁盘

二 部署ceph集群

步骤一：安装部署软件ceph-deploy

1. 在node1安装部署工具

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

2）创建目录

[root@node1 ~]# mkdir ceph-cluster

[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。

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

2）给所有节点安装ceph相关软件包。

[root@node1 ceph-cluster]# for i in node1 node2 node3

do

ssh $i "yum -y install ceph-mon ceph-osd ceph-mds ceph-radosgw"

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）。

[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：

[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。

[root@node1 ceph-cluster]# vim ceph.conf #文件最后追加以下内容

public\_network = 192.168.4.0/24

修改后重新推送配置文件:

[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。

[root@node1 ceph-cluster]# for i in node1 node2 node3

do

ssh $i "parted /dev/vdb mklabel gpt"

ssh $i "parted /dev/vdb mkpart primary 1 50%"

ssh $i "parted /dev/vdb mkpart primary 50% 100%"

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。

[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]# for i in node1 node2 node3

Do

ceph-deploy disk zap $i:vdc $i:vdd

Done

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]# for i in node1 node2 node3

Do

ceph-deploy osd create $i:vdc:/dev/vdb1 $i:vdd:/dev/vdb2

Done

//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

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：

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

步骤四：验证测试

1) 查看集群状态。

[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：

health: HEALTH\_WARN

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：

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：

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

三 创建ceph块存储

步骤一:创建镜像

[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。

[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

[root@node1 ~]# rbd resize --size 7G image --allow-shrink #缩小容量

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

步骤三:通过KRBD访问

[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 #访问image镜像

[root@client ~]# lsblk

[root@client ~]# rbd showmapped

id pool image snap device

0 rbd image - /dev/rbd0

[root@client ~]# mkfs.xfs /dev/rbd0 #只需要格式化一次

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

[root@client ~]# echo "test" > /mnt/test.txt #测试

步骤四:创建镜像快照

[root@node1 ~]# rbd snap ls image #查看是否有快照(镜像默认没有快照)

[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

[root@client ~]# rm -rf /mnt/test.txt #删除客户端写入的测试文件

[root@client ~]# umount /mnt

[root@node1 ~]# rbd snap rollback image --snap image-snap1 #客户端重新挂载分区

[root@client ~]# mount /dev/rbd0 /mnt/ #还原快照

[root@client ~]# ls /mnt

步骤四:创建快照克隆

[root@node1 ~]# rbd snap protect image --snap image-sanp1

[root@node1 ~]# rbd clone image --snap image-sanp1 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

[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！

[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

步骤五:其他操作

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