Day04 Ceph概述 部署Ceph集群 Ceph块存储

1. Ceph概述
2. 基础知识
3. 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

/\*\*

\* 常见分布式网站架构

\* 1.用户访问调度器

\* 2.后台有一组应用服务器，调度器将请求转发给服务器

\* 3.服务器下有各自的页面文件/目录

\* 4.可以在后台做一台共享存储NFS/Sumba，服务器mount此目录

\* 5.当文件过大，NFS无法支持的时候，网站会没法继续做下去

\* 6.还有一台数据库为应用服务器提供数据

\* 7.数据库和共享存储统称为数据端

\*\*/

1. 常用分布式文件系统

· Lustre

· Hadoop

· FastDES

· Ceph

· GlusterFS

1. 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

/\*\*

\* 对象存储，也叫做基于对象的存储，是用来描述解决和处理离散单元的方法的通用术语，这些离散单元被称作为对象。

\* 就像文件一样，对象包含数据，但是和文件不同的是，对象在一个层结构中不会再有层级结构。每个对象都在一个被称作存储池的扁平地址空间的同一级别里，一个对象不会属于另一个对象的下一级。

\* 文件和对象都有与它们所包含的数据相关的元数据，但是对象是以扩展元数据为特征的。每个对象都被分配一个唯一的标识符，允许一个服务器或者最终用户来检索对象，而不必知道数据的物理地址。这种方法对于在云计算环境中自动化和简化数据存储有帮助。

\*\*/

· Ceph可以提供PB级别的存储空间(PB>TB>GB)

- 1024G\*1024G=1048576G

· 软件定义UC年初(Software Defined Storage)作为存储行业的一大发展趋势，已经越来越上受到市场的认可

· 帮助文档：<http://docs.ceph.org/start/intro>

1. Ceph组件

· OSDs

- 存储设备

· Monitors

- 集群监控组件

- 至少三台启动，否则集群报错

· RGW

- 对象存储网关

· 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

iscsi[块]

NFS[文件系统]

· Ceph提供：块，文件系统，对象存储(百度云)

- ceph-osd ceph-mon

\* 必装，用户只能以块的方式访问

- ceph-mds

\* 可以以文件系统方式访问，mount挂载

- ceph-radosgw (RGW，radosgateway)

\* 可以以对象存储的方式访问

/\*\*

\* 分布式存储的算法

\* 以取余算法为例：

\* node1:10T node2:10T node3:10T node4:10T 4台计算机为后端存储OSD，提供真正的存储磁盘

\* Monitor[监控/管理]：算法的执行者，用户访问数据前需要先访问Monitor

\* 用户访问a.txt|md5sum=16进制数%4=1 ==> 存入node2

\* b.txt|md5sum=16进制数%4=0 ==> 存入node1

\* Ceph用的是CRUSH算法

\*\*/

/\*\*

\* 分布式存储的特点

\* 1.数据被分开存储

\* 2.数据的读写是并行的

\* 3.数据会像raid一样，单个文件会被打散

\* 4.Ceph所有数据是3副本

\*\*/

1. 实验环境准备
2. 安装前准备
3. 物理机为所有节点配置yum源服务器

[root@room9pc01 ~]# yum -y install vsftpd

[root@room9pc01 ~]# mkdir /var/ftp/ceph

[root@room9pc01 ~]# mount -o loop \

rhcs2.0-rhosp9-20161113-x86\_64.iso /var/ftp/ceph

[root@room9pc01 ~]# systemctl restart vsftpd

1. 修改所有节点都需要配置YUM源（这里仅以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

1. 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！！！！

[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

1. 配置无密码连接(包括自己远程自己也不需要密码)

[root@node1 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N ''

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

> do

> ssh-copy-id 192.168.4.$i

> done

1. 配置NTP时间同步
2. 真实物理机创建NTP服务器

[root@room9pc01 ~]# 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@room9pc01 ~]# systemctl restart chronyd

如果有防火墙规则，需要清空所有规则

[root@room9pc01 ~]# iptables -F

1. 其他所有节点与NTP服务器同步时间（以node1为例）

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

server 192.168.4.254 iburst

[root@node1 ~]# systemctl restart chronyd

1. 准备存储磁盘
2. 物理机上为每个虚拟机准备3块磁盘。（可以使用命令，也可以使用图形直接添加）

[root@room9pc01 ~]# virt-manager

1. 部署Ceph集群
2. 准备部署环境
3. 安装部署软件

· 使用node1作为部署主机

# yum -y install ceph-deploy

· ceph-deploy命令与子命令都支持--help查看帮助

# ceph-deploy --help

1. 创建目录

· 为部署工具创建目录，存放秘钥与配置文件

# mkdir ceph-cluster

# cd ceph-cluster/

1. 部署存储集群
2. 创建Ceph集群
3. 创建Ceph集群配置(所有节点都为mon)

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

1. 给所有节点安装Ceph软件包

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

1. 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 \*\*/

/\*\*

\* 常见错误和解决方法：

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

\* 先检查自己的命令是否是在ceph-cluster目录下执行的！！！！如果时确认是在该目录下执行的create-initial命令，依然保存，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

\*\*/

1. 初始化完成后，使用ceph -s与systemctl查看状态

# systemctl status ceph-<tab>

[ceph-create-keys@node1.service](mailto:ceph-create-keys@node1.service) ceph-mon.target

ceph-mds.target ceph-osd.target

ceph-mon@node1.service ceph-radosgw.target

1. 创建OSD
2. 给所有节点准备磁盘分区

[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 ~]# lsblk

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

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

[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"

/\*\*

\* 我们都爱for循环

\* for i in node{1..3}; do

ssh $i 'parted /dev/vdb mklabel gpt;

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

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

chown ceph.ceph /dev/vdb1;

chown ceph.ceph /dev/vdb2';

ssh $i '

cat > /etc/udev/rules.d/70-vdb.rules << EOF

ENV{DEVNAME}=="/dev/vdb1",OWNER="ceph",GROUP="ceph"

ENV{DEVNAME}=="/dev/vdb2",OWNER="ceph",GROUP="ceph"

EOF'; done

\*\*/

1. 初始化清空磁盘数据

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

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

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

/\*\*

\* 我们都爱for循环

\* [root@node1 ceph-cluster]# for i in node{1..3};do ceph-deploy disk zap $i:vd{c,d};done

\*\*/

1. 创建OSD存储空间

[root@node1 ceph-cluster]# ceph-deploy osd create \

> node1:vdc:/dev/vdb1 node1:vdd:/dev/vdb2

[root@node1 ceph-cluster]# ceph-deploy osd create \

> node2:vdc:/dev/vdb1 node2:vdd:/dev/vdb2

[root@node1 ceph-cluster]# 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

/\*\*

\* [root@node1 ceph-cluster]# for i in node{1..3};do ceph-deploy osd create $i:vdc:/dev/vdb1 $i:vdd:/dev/vdb2;done

\*\*/

/\*\*

\* vdb1和vdb2用来做存储服务器的日志journal盘

\* vdb1做vdc的缓存，vdb2做vdd缓存

\* 缓存盘对速度的要求较高，因此最好是用固态硬盘，而ceph做在传统磁盘上

\*\*/

1. 验证

· 查看集群状态

[root@node1 ~]# ceph -s

/\*\*

\* 常见错误：

HEALTH\_WARN

clock ...

表示时间不同步，将所有主机时间同步即可

Ceph要求所有主机时间差不能超过0.05s

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

\* 错误2：

权限配置完毕而ceph -s 显示EOR，可能是因为磁盘信息没有刷新

# partprobe

状态根据ceph主机数量和存储容量大小来确定

HEALTH状态会有延迟

\*\*/

1. Ceph块存储
2. 概述
3. 什么是块存储

· 单机块设备

- 光盘

- 磁盘

· 分布式块存储

- Ceph

- Cinder

· Ceph块设备也叫做RADOS块设备

- RADOS block device：RBD

· RDB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

· RBD还支持内存缓存，从而能够大大提高性能

· Linux内核可用直接访问Ceph块存储

· KVM可用借助于librbd访问



图例1

1. 块存储集群
2. 创建镜像

· 查看存储池(默认有一个rbd池)

[root@node1 ~]# ceph osd lspools

· 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

/\*\* image-feature支持哪些功能; layering 支持分层快照功能 \*\*/

/\*\* 帮助文档 rbd help create \*\*/

[root@node1 ~]# rbd list

[root@node1 ~]# rbd info demo-image

1. 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

/\*\*

\* 三台ceph-mon 三台ceph-osd

\* rbd create image 创建镜像

\*\*/

1. 集群内通过KRBD访问

· 集群内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

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

[root@node1 ~]# mount /dev/rbd0 /mnt

1. 客户端通过KRBD访问
2. 客户端需要安装ceph-common软件包
3. 拷贝配置文件(否则不知道集群在哪)
4. 拷贝连接密钥(否则无连接权限)

[root@client ~]# yum install ceph-common -y

[root@node1 ceph]# scp ceph.conf client:/etc/ceph/

[root@node1 ceph]# scp ceph.client.admin.keyring client:/etc/ceph/

1. 将镜像映射为本地磁盘

[root@client ~]# rbd map image

[root@client ~]# lsblk

[root@client ~]# rbd showmapped

id pool image snap device

0 rbd image - /dev/rbd0

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

1. 创建镜像快照

/\*\* 快照 COW(Copy on Write)写时复制 \*\*/

/\*\* 快照使用写时复制技术，对大数据快照速度会很快

\* 原始盘100G

\* a.txt:111--XXX 1G

\* b.txt:222

\* 快照备份50G

\* a.txt:111

\* b.txt:222

\* 10T数据库 先快照 --> 后期再cp和tar

\* Ceph支持快照，但不支持在线快照，在线快照会使Ceph盘崩溃

\*\*/

1. 查看镜像快照

[root@node1 ~]# rbd snap ls image

1. 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

1. 删除客户端写入的测试文件

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

1. 使用快照恢复数据
2. 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

1. 快照克隆

[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

1. 客户端撤销磁盘映射
2. 删除快照与镜像
3. 总结和答疑