软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作系统 | IP地址 | 节点 | 磁盘 | 内存 |
| CentOS 7.1.1503 | 124.251.47.223 | 计算节点 | 300G×2 | 32G |
| CentOS 7.1.1503 | 124.251.47.224 | 计算节点 | 300G×2 | 32G |
| CentOS 7.1.1503 | 124.251.47.218 | 控制节点 | 300G | 32G |
| CentOS 7.1.1503 | 124.251.47.219 | 控制节点 | 300G | 32G |
| CentOS 7.1.1503 | 124.251.47.221(50.2) | 网络节点 | 300G | 32G |
| CentOS 7.1.1503 | 124.251.47.222 | 网络节点 | 300G | 32G |

实施

1. 系统环境初始化
2. 系统网络环境初始化
3. GlusterFS 配置
4. Openstack 配置
   1. 初始环境配置
   2. OPENSTACK KEYSTONE 认证配置
   3. OPENSTACK GLANCE镜像配置
   4. OPENSTACK COMPUTE 计算配置
   5. OPENSTACK NEUTRON网络配置
   6. OPENSTACK DASHBOARD面板配置
   7. OPENSTACK BLOCK STORAGE块存储配置
   8. OPENSTACK QA
5. Openstack 块迁移
6. OPENSTACK + GlusterFS配置
7. 系统环境初始化

省略…

1. 系统网络环境初始化

省略…

1. GlusterFS配置

GlusterFS集群

GlusterFS服务器

SERVER01：124.251.47.223

SERVER02：124.251.47.224

GlusterFS客户端

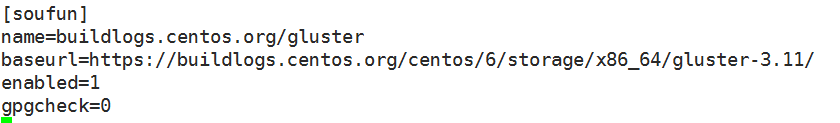
CLIENT01：124.251.47.223

CLIENT02：124.251.47.224

安装部署：

得到源

##wget -P /etc/yum.repos.d <http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo>



安装软件包

yum install -y glusterfs glusterfs-server glusterfs-fuse

# 设置开机启动

systemctl enable glusterd.service

# 启动服务

systemctl start glusterd.service

配置server端集群

# gluster peer probe zookeeper03 # 本机显示信息如下

peer probe: success. Probe on localhost not needed

# gluster peer probe zookeeper02

# gluster peer probe zookeeper01 # 其它机器信息如下

peer probe: success.

# 查看集群状态信息

# gluster peer status

Number of Peers: 2

Hostname: zookeeper02

Uuid: dc73bb27-9472-4be9-8d30-018b3240bfb6

State: Peer in Cluster (Connected)

Hostname: zookeeper01

Uuid: cc89969d-0646-4f4e-a375-8145802a911e

State: Peer in Cluster (Connected)

# 创建glusterfs目录，每台都需要创建

mkdir /www/{glusterdata,glusterdata1}

# 创建卷，并设置保存三份，名字为glusterdata,指向路径为/www/glusterdata

# gluster volume create glusterdata replica 2 transport tcp zookeeper01:/www/glusterdata

zookeeper02:/www/glusterdata zookeeper03:/www/glusterdata

volume create: glusterdata: success: please start the volume to access data

# 参考命令 gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2

# 启动磁盘

# gluster volume start glusterdata

volume start: glusterdata: success

# 查看磁盘块信息

# gluster volume info

Volume Name: glusterdata

Type: Replicate

Volume ID: 82c74e12-b4fb-4a47-94b0-babb7c01443c

Status: Started

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: zookeeper01:/www/glusterdata

Brick2: zookeeper02:/www/glusterdata

Brick3: zookeeper03:/www/glusterdata

Options Reconfigured:

performance.readdir-ahead: on

【挂载之后，修改修改挂载点权限为777，否则其他账户无法使用】

############################## gluster 客户端配置 #################################

# 安装软件

yum install -y glusterfs glusterfs-fuse

# 创建挂载点

# mkdir /glusterdata

# 挂载，只需要指定server端一台机器即可

# 挂载，并设置为只读，不可写

mount -t glusterfs -o ro zookeeper01:glusterdata /glusterdata/

# 挂载，可读写

mount -t glusterfs zookeeper01:glusterdata /glusterdata/

# 高可用挂载，避免其中挂载点宕机的情况，客户端可以挂载多个节点，实现高可用。

mount -t glusterfs -o backup-volfile-servers=zookeeper02:zookeeper03 zookeeper01:glusterdata /glusterdata/

mount -t glusterfs -o backup-volfile-servers=idc1-server2:idc1-server3:idc1-server4,ro idc1-server1:datavolume2 /mnt/datavolume2/

## 高可用挂载测试结果：当zookeeper01这台机器宕了之后，不能创建磁盘，df -Th不能查看信息。但是原先已经挂载好的磁盘信息，或者已经创建好的磁盘，不会受到影响，一样可以挂载，并在挂载好的磁盘上读写操作

#######待测试

# 删除GlusterFS磁盘：

gluster volume stop datavolume1

gluster volume delete datavolume1

# 卸载GlusterFS磁盘：

gluster peer detach idc1-server4

# 访问控制：

gluster volume set datavolume1 auth.allow 192.168.242.\*,192.168.241.\*

# 添加GlusterFS节点：

gluster peer probe c6

gluster peer probe c7

gluster volume add-brick datavolume1 c6:/usr/local/share/datavolume1 c7:/usr/local/share/datavolume1

# 迁移GlusterFS磁盘数据：

gluster volume remove-brick datavolume1 c1:/usr/local/share/datavolume1 c6:/usr/local/share/datavolume1 start

gluster volume remove-brick datavolume1 c1:/usr/local/share/datavolume1 c6:/usr/local/share/datavolume1 status

gluster volume remove-brick datavolume1 c1:/usr/local/share/datavolume1 c6:/usr/local/share/datavolume1 commit

# 数据重新分配：

gluster volume rebalance datavolume1 start

gluster volume rebalance datavolume1 status

gluster volume rebalance datavolume1 stop

# 修复GlusterFS磁盘数据（例如在c1宕机的情况下）：

gluster volume replace-brick datavolume1 c1:/usr/local/share/datavolume1 c6:/usr/local/share/datavolume1 commit -force

gluster volume heal datavolume1 full

################################## gluster + openstack 结合使用 ####################################

vim /etc/cinder/cinder.conf

# openstack-config --set /etc/cinder/cinder.conf DEFAULT volume\_driver cinder.volume.drivers.glusterfs.GlusterfsDriver

# openstack-config --set /etc/cinder/cinder.conf DEFAULT glusterfs\_shares\_config /etc/cinder/shares.conf

# openstack-config --set /etc/cinder/cinder.conf DEFAULT glusterfs\_mount\_point\_base /var/lib/cinder/volumes

然后

vim /etc/cinder/shares.conf

指定gluster客户端和挂载点

zookeeper01:/glusterdata # 注意，必须是绝对路径

GLUSTERHOST:VOLUME

GLUSTERHOST:NEXTVOLUME

GLUSTERHOST2:SOMEOTHERVOLUME

# 重启cinder相应服务

$ for i in api scheduler volume; do sudo service openstack-cinder-${i} start; done

# 观察日志，是否有错误

sudo tail -50 /var/log/cinder/volume.log

######################################## 验证是否成功 ############################

# 在控制节点上

# 创建逻辑卷

cinder create --display\_name myvol 10

# 查看是否创建成功

cinder list

# 然后到GlusterFS服务器上查看是否存在

$ sudo ls -lah /var/lib/cinder/volumes/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/

### 测试

高可用测试

挂载

mount -t glusterfs -o backup-volfile-servers=sjhl-w-vm118:glusterdatat sjhl-w-vm120:glusterdata /glusterdata/

Sjhl-w-vm120磁盘损坏，系统删除

Sjhl-w-wm118上挂载

mount -t glusterfs -o backup-volfile-servers=sjhl-w-vm120,rw sjhl-w-vm118:glusterdata /glusterdata/

疑惑：gluster集群如何恢复节点和添加新节点

### 基础知识

<https://my.oschina.net/hncscwc/blog/210072>

[gluster卷种类](http://blog.chinaunix.net/uid-11026451-id-4983080.html)

[gluster配置安装](http://cmdschool.blog.51cto.com/2420395/1828450)

（1）brick：brick是被标记为卷的文件系统

（2）subvolume：一个经过至少一个translator处理的brick

（3）volume：经过所有translator之后的最终共享

（4）translator：一个translator连接一个或多个subvolumes，做一些处理，并且提供一个subvolume连接

distributed volumes

replicated volumes

striped volumes

distributed replicated volumes

distributed striped volumes