Gluster常见操作及问题总结

GlusterFS是一个具有高扩展性、高性能、高可用性、可横向扩展的弹性分布式文件系统，本文主要对Gluster的一些常见命令用法及常见问题处理，包括后续测试办法做了整理。

1.虚拟机配置IP关键参数（/etc/sysconfig/network-scripts/ifcfg-eth0）

Bootproto=static

ONBOOT=yes

IPADDR=20.0.0.12

PREFIXO=24

GATEWAY=20.0.0.1

DNS1=114.114.114.114

2.安装glusterfs –server及配置集群  
A. yum install centos-release-gluster  
yum install -y glusterfs glusterfs-server glusterfs-fuse glusterfs-rdma

启动 glusterFS   
systemctl start glusterd.service  
systemctl enable glusterd.service

GlusterFS无法自动启动原因及处理方案:

glusterfs配置文件(/etc/glusterfs/glusterd.vol)出现问题:

#working-directory的位置为/var/lib/glusterfsd  
option working-directory /var/lib/glusterfsd

将working-directory的位置修改为配置文件所在位置（即为/etc/glusterfs)

B.配置集群

首先在/etc/hosts中要配置集群中ip与hostname的对应关系，后续配置集群均使用hostname

gluster peer probe node1

gluster peer probe node2

……

3.删除集群中的节点

Gluster peer detach node01 (force)

如果失败，采用下面的方法：

systemctl stop glusterd

cd /var/lib/glusterd

cp glusterd.info ../

rm –rf \*

cp ../glusterd.info .

systemctl restart glusterd

4.创建卷volume

例1(分布式复制卷):

gluster volume create models replica 2 node01:/usr1/gluster/data/ node02:/usr1/gluster/data/ node03:/usr1/gluster/data/ node01:/usr2/gluster/data/ node02:/usr2/gluster/data/ node03:/usr2/gluster/data/

例2（条带复制卷）：

gluster volume create models stripe 2 replica 2 node01:/usr1/gluster/data/ node03:/usr1/gluster/data/ node01:/usr2/gluster/data/ node03:/usr2/gluster/data/

5.删除volume

Gluster volume stop models

Gluster volume delete models

6．新增bricks

gluster volume add-brick models replica 2 node01:/usr1/gluster/data/ node02:/usr1/gluster/data/

7.删除bricks

a. gluster volume remove-brick models  replica 2 node01:/usr1/gluster/data/ node02:/usr1/gluster/data/ start 先进行数据迁移

b. gluster volume remove-brick models  replica 2 node01:/usr1/gluster/data/ node02:/usr1/gluster/data/ commit 删除brick（如果没有进行第一步，数据不会迁移）

备注：如果删除了重新添加提示$path is already part of a volume

1. systemctl stop glusterd

setfattr -x trusted.glusterfs.volume-id $path

setfattr -x trusted.gfid $path

rm -rf $path/.glusterfs/

systemctl restart glusterd

ssh node02 'setfattr -x trusted.glusterfs.volume-id /usr1/gluster/data/ && setfattr -x trusted.gfid /usr1/gluster/data/ && rm -rf /usr1/gluster/data/.glusterfs'

setfattr -x trusted.glusterfs.volume-id /usr2/gluster/data/ && setfattr -x trusted.gfid /usr2/gluster/data/ && rm -rf /usr2/gluster/data/.glusterfs

补充：目前来看再删除节点之后，此操作不一定能使该节点重新加入到其他volume中，需要将node集群释放才可以（参照方法3），推测node服务器记录了原brick的信息。

B. gluster volume create models replica 2 node01:/usr1/gluster/data/ node02:/usr1/gluster/data/ force目前此方法测试可行。

8.新增或删除bricks之后需要平衡数据分布

gluster volume rebalance <VOLNAME> [fix-layout|migrate-data] {start开始|stop停止|status查看状态}，利用这个功能可以对已经存在的目录文件进行Rebalance，使得早先创建的老目录可以在新增存储节点上分布，并可对现有文件数据进行迁移实现容量负载均衡。为了便于控制管理，rebalance操作分为两个阶段进行实际执行，即fix layout和migrate data。gluster volume rebalance目前支持以下三种应用场景：  
a.Fix Layout 修复layout以使得新旧目录下新建文件可以在新增节点上分布上。  
b.Migrate Data 新增或缩减节点后，在卷下所有节点上进行容量负载平滑。为了提高rebalance效率，通常在执行此操作前先执行Fix Layout。  
c.Fix Layout and Migrate Data 用法：gluster volume rebalance <VOLNAME> {start|stop|status}，同时执行以上两个阶段操作，先Fix Layout再Migrate Data。

9.客户端安装及挂载

yum install -y glusterfs glusterfs-fuse

mkdir -p /opt/gfsmnt

mount –t glusterfs node1:models /opt/gfsmnt/

10.  迁移命令:  
gluster volume replace-brick models node01:/usr1/gluster/data/ node02:/usr1/gluster/data/ commit –force

11.地域复制(geo-replication)地域复制是系统提供的灾备功能，能够将系统的全部数据进行异步的增量备份到另外的磁盘中。

#gluster volume geo-replication models 20.0.0.15:/usr1 start

//如上，开始执行将models卷的所有内容备份到15下的/usr1，需要注意的是，这个备份目标不能是系统中的Brick。

12.测试命令

a.安装

yum install libaio-devel（fio的依赖包）

yum install fio-2.1.2-1.el6.rf.x86\_64.rpm

b.直接使用dd命令

dd if=/dev/zero of=hello bs=1000MB count=1

c.使用fio工具

顺序写fio -filename=/opt/gfsmnt/1:/opt/gfsmnt/2:/opt/gfsmnt/4:/opt/gfsmnt/7 -direct=1 -iodepth=1 -thread -rw=write -ioengine=libaio -bs=1M -size=10G -numjobs=4 -group\_reporting -name=mytest

随机写

fio -filename=/opt/gfsmnt1/3:/opt/gfsmnt1/5:/opt/gfsmnt1/6:/opt/gfsmnt1/8 -direct=1 -iodepth=1 -thread -rw=randwrite -ioengine=libaio -bs=1M -size=10G -numjobs=4 -group\_reporting -name=mytest

fio -filename=/opt/gfsmnt2/9:/opt/gfsmnt2/10:/opt/gfsmnt2/11:/opt/gfsmnt2/12 -direct=1 -iodepth=1 -thread -rw=write -ioengine=libaio -bs=1M -size=10G -numjobs=4 -group\_reporting -name=mytest

随机读写

fio -filename=/opt/gfsmnt2/a:/opt/gfsmnt2/b:/opt/gfsmnt2/c:/opt/gfsmnt2/d -direct=1 -iodepth=1 -thread -rw=randrw -rwmixwrite=70 -ioengine=libaio -bs=1M -size=10G -numjobs=4 -group\_reporting -name=mytest