## gluster操作维护手册

目录

[gluster操作维护手册 1](#_Toc30915)

[1、安装gluster 1](#_Toc3744)

[2、部署gluster集群准备工作 1](#_Toc5568)

[（1）编辑每个节点的/etc/hosts文件（客户端与服务端） 1](#_Toc6951)

[（2）对每个节点进行时间同步,关闭防火墙,关掉selinux 2](#_Toc7120)

[3、创建GlusterFS集群 2](#_Toc943)

[（1）创建集群 2](#_Toc31023)

[（2） 移除集群 2](#_Toc4419)

[（3） 挂载分区 2](#_Toc8065)

[4、 创建GlusterFS卷 3](#_Toc28875)

[（1）卷模式说明（数据冗余） 3](#_Toc19085)

[（2） 创建卷，启动卷 3](#_Toc20305)

[（3） 在两个节点上分别mount以上创建的卷 4](#_Toc23063)

[（4） 扩展卷 4](#_Toc16640)

[（5） 收缩卷 4](#_Toc16198)

[（6） Brick替换 5](#_Toc4450)

[（7） 卷平衡 5](#_Toc17013)

[（8） 停止卷 6](#_Toc20089)

[（9） 删除卷 6](#_Toc19558)

[5、 GlusterFS卷功能与常见问题 6](#_Toc11678)

[（1） 存储集群状态 6](#_Toc30254)

[（2） 存储池（卷）状态 7](#_Toc8262)

[（3） 卷自修复 7](#_Toc32081)

[（4） shard功能 8](#_Toc24618)

[（5） client无法挂载 8](#_Toc26083)

[（6） 自修复服务显示N，未上线 9](#_Toc13661)

#### 1、安装gluster（ubuntu在/etc/rc.local中添加开机启动glusterd）

#### 2、部署gluster集群准备工作

##### （1）编辑每个节点的/etc/hosts文件（客户端与服务端）

# vi /etc/hosts

添加集群中每个节点的主机名和主机的IP地址的对应关系，如下：

192.168.15.130 node1

192.168.15.129 node2

...

127.0.0.1 localhost

##### （2）对每个节点进行时间同步,关闭防火墙,关掉selinux

#### 3、创建GlusterFS集群

##### （1）创建集群

# gluster peer probe node2

Probe successful

##### 移除集群

# gluster peer detach node3

Probe successful

##### ****挂载分区****

格式化磁盘（建议为mkfs.xfs -f -i size=2048），创建挂载点，名称建议为/brick{1..n}

#mkdir /brick1

使用#lsblk命令可以清晰的获取全局的块设备布局，一般服务器都有多个硬盘分区，在重启后，这些分区的逻辑位置加载时可能会发生变动，如果使用传统的设备名称(例如：/dev/sda)方式挂载磁盘，就可能因为磁盘顺序变化而造成混乱。Linux环境中每个Block Device都有一个全局唯一的UUID，可以标识这个设备，#blkid命令可以获取设备的UUID，然后复制记下设备的UUID。例如：

#mount UUID="b510d40d-c28a-49c2-a6fc-db8c97adecd8" /brick1 -o noatime

设置开机启动自动挂载，# vi /etc/fstab，在最后加上一行，然后:wq保存退出。

UUID=b510d40d-c28a-49c2-a6fc-db8c97adecd /brick1 xfs defaults,noatime 0 0

不能直接使用挂载目录/brick1直接来作为gluster的卷参数，挂载好后需要增加一层：

#mkdir /brick1/data

#### 创建GlusterFS卷

##### （1）卷模式说明（数据冗余）

* 复制卷 - 在高可用性和高可靠性环境中使用复制卷。副本数越多，数据可用性越好，可靠性也越高，但也意味着更低的空间利用率以及更高的成本。
* 纠删卷 - 纠删码机制具有更高的存储效率，在提供相同存储可靠性的条件下，可以最小化冗余存储开销，但是相对于复制卷其IOPS性能下降较多。

##### 创建卷，启动卷

**创建纠删卷**，Disperse卷的创建与节点个数无关（节点个数大于等于1），只与bricks(B)、冗余度redunancy(R)相关；其中bricks(B)必须大于等于3，disperse-data 的个数必须要大于等于2，redunancy(R)的值最小为1，最大为(B-1)/2、必须小于bricks(B)的一半且值是不能改变的.基于Glusterfs搭建的集群中，创建EC卷推荐以下几种配置：

a. 冗余度为1时，推荐创建配置为(2+1)的EC卷；

b. 冗余度为2时，推荐创建配置为(4 +2)的EC卷；

c. 冗余度为3时，推荐创建配置为(8 +3)的EC卷；

d. 冗余度为4时，推荐创建配置为(8 + 4)的EC卷。

由于在相同的EC配置下，Distributed Disperse卷比Disperse卷具有更好的IO性能，所以推荐在硬盘数量足够的情况下创建Distributed Disperse卷。在底层的配置中，推荐逻辑盘（如/dev/sda）不分区直接格式化为块大小512B的XFS文件系统，且逻辑盘与brick是一一对应的关系；推荐有n个节点，B就等于n，即同一组的Disperse卷配置中，一个brick对应一个节点。比如，三个节点的gluster集群就推荐创建配为（B=3，R=1）的EC卷，每组Disperse卷配置中，一个brick对应一个节点。

# gluster volume create test-volume disperse 3 redundancy 1 node1:/brick1/data node2:/brick1/data node3:/brick1/data node1:/brick2/data node2:/brick2/data node3:/brick2/data

Creation of test-volume has been successful

Please start the volume to access data.

# gluster volume start test-volume

**创建复制卷**,最少配置复制卷数为2。如果考虑可能出现的脑裂问题（主要由于客户端与brick程序断连导致），可以参考多配置一个仲裁盘arbiter 1或者配置为3副本；建议直接配置为3副本最好。

# gluster volume create test-volume replica 3 node1:/brick1/data node2:/brick1/data node3:/brick1/data

Creation of test-volume has been successful

Please start the volume to access data.

# gluster volume start test-volume

##### 在客户端节点上mount以上创建的卷

#mount -t glusterfs node1:test-volume /mntpoint

##### 扩展卷

当需要容量扩展时，对于副本卷和纠删卷需要添加其倍数。添加后需要平衡（见（7））

#gluster volume add-brick test-volume replica 2 node1:/brick2/data node2:/brick2/data

##### 收缩卷

当因为磁盘原因需要收缩容量时，对于副本和纠删需要删除其倍数，并且是同一组的需要全部删除；收缩卷需要先进行数据迁移，等待完成之后再确认删除。

#gluster volume remove-brick test-volume replica 2 node1:/brick2/data node2:/brick2/data start

查看状态（此时brick依然存在于卷中）

#gluster volume remove-brick test-volume replica 2 node1:/brick2/data node2:/brick2/data start

确认状态完成之后再commit

#gluster volume remove-brick test-volume replica 2 node1:/brick2/data node2:/brick2/data commit

Removing brick(s) can result in data loss. Do you want to Continue? (y/n) y

volume remove-brick commit: success

Check the removed bricks to ensure all files are migrated.

If files with data are found on the brick path, copy them via a gluster mount point before re-purposing the removed brick.

##### Brick替换

场景发生在当磁盘故障时需要替换磁盘，在原盘已经故障的情况下（数据已丢失），重新换上新盘存储池并不能识别，因为数据已丢失所以不存在等待迁移数据问题，直接强制提交即可。同时新换上的盘挂载目录名不能与故障目录同名。替换成功之后不需要其他动作冗余卷自身的恢复功能会自动恢复数据。

#gluster volume replace-brick test-volume node1:/brick1/data node1:/brick1/data-new commit force

##### 卷平衡

扩大卷（使用add-brick命令）后，需要重新平衡服务器之间的数据。扩大后创建的新目录将自动均匀分布。对于所有现有的目录中在平衡可以修改分布式出现的不均衡。分为修复布局和迁移数据两步：

修复布局：gluster文件定位是按照哈希来确定文件位置，brick记录了哈希的范围，不管是add-brick还是remove-brick原有的哈希分布布局都不变，这样即使新写的文件也无法定位到新brick上去，这时就需要修复布局

#gluster volume rebalance test-volume fix-layout start

volume rebalance: rep: success: Rebalance on rep has been started successfully. Use rebalance status command to check status of the rebalance process.

数据迁移：布局修复完成之后，对原来集群中的数据需要进行迁移，避免分布不均

#gluster volume rebalance test-volume start

volume rebalance: rep: success: Rebalance on rep has been started successfully. Use rebalance status command to check status of the rebalance process.

查看平衡状态：

#gluster volume rebalance test-volume status

##### 停止卷

#gluster volume stop test-volume

Stopping volume will make its data inaccessible. Do you want to continue? (y/n) y

volume stop: rep: success

##### 删除卷

#gluster volume delete test-volume

Deleting volume will erase all information about the volume. Do you want to continue? (y/n) y

volume delete: rep: success

#### GlusterFS卷功能与常见问题

##### 存储集群状态

1. 集群状态主要看各节点链接状态，正常状态为connected，异常状态为disconnected; 对应节点看hostname

# gluster pool list



表 1 正常状态

1. Node2掉线或者glusterd管理服务挂死时，node2的状态为disconnected，但是只能在正常状态的节点上查看，如果node2是glusterd管理服务挂死，那么集群状态和存储池状态（见（2））都查询不到，会显示“Connection failed. Please check if gluster daemon is operational”，此时要显示状态信息就需要ssh登陆其他节点来查看状态（优先配置好ssh无密码访问）：



表 2 异常状态

##### 存储池（卷）状态

Gluster存储池的状态并不像ceph有一个healthy状态，所以需要具体的来判断。分基础状态和功能状态，Y为上线，N或者不存在为下线：

1. 基础状态：所有的brick全部上线online-Y，这个判断是保证gluster存储池读写数据的基本状态要求，brick掉线或者down掉则显示online-N
2. 功能状态：

B1. 类似纠删卷和副本卷如果开启修复功能均有self-heal服务处于上线online-Y；如果关闭不显示该服务。（数据恢复功能默认是启动的）

# gluster volume status test-volume

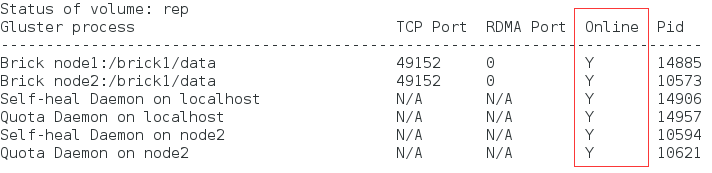


表 3 启动恢复，brick全部上线

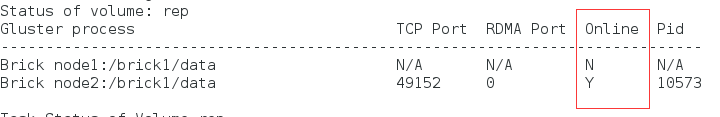


表 4 关闭恢复，brick1掉线

##### 卷自修复

副本卷和纠删卷都有数据修复服务self-heal，默认都是开启的，可手动关闭或开启

# gluster volume heal test-volume enable/disable

卷自修复分为两种：index-heal、full-heal

1. Index-heal：触发条件有3种（每隔10分钟，通过gluster volume heal rep显示触发，当一个brick掉线又重新上线时）

# gluster volume heal test-volume

1. Full-heal: 完全修复，只能是手动触发，一般情况不建议使用；场场景一般是副本模式下出现脑裂时，手动处理掉异常文件后使用full来修复。与heal不同的是full-heal只在子卷的一个节点上运行，并且是uuid最大的一个节点。与index不同的是从卷的根开始修复。

# gluster volume heal test-volume full

1. 查看修复的条目

# gluster volume heal test-volume info

1. 脑裂split-brain:在2副本的场景下可能会出现脑裂现象（互相认为对方有错，可以通过三副本或者仲裁来规避）再已经出现这种问题的情况下，修复heal是无法自动解决的，需要人为干预，首先查看是否脑裂信息，然后我们自己判断哪个文件异常，手动将其删除，然后执行heal来修复。

# gluster volume heal test-volume info split-brain

##### shard功能

Gluster对非常大的文件（超过一个brick）在某些使用场景有限制，因此提出一个新的解决方案shard，文件的整个大小记录在第一个块的xattr中，这样做大文件可以正常分布，同时自修复时可以分布跨越更多服务器。再已经写入的文件如果调整切片大小，那么原来文件的属性不变，如果不想分片，建议创建新卷将其拷贝过去。

# gluster volume set test-volume features.shard on //开启

# gluster volume set test-volume features.shard-block-size xxMB //默认是64MB，不建议修改

##### client无法挂载

检查/etc/hosts中需要配置整个集群所有节点的名称，确保不论服务端还是客户端都要添加所有。例如初始只有3个节点，创建了一个2：1卷test，此时集群新添加了2个节点，这时如果不更新所有节点的/etc/hosts那么原来的test卷就无法挂载

##### 自修复服务显示N，未上线

明明已经开启了self-heal功能，但gluster volume status看到的状态还是N，则检查/etc/hosts中是否缺少 127.0.0.1 localhost，添加上之后重启glusterd服务。