# GlusterFS中文文档

# 一、简介

### 1.1 什么是Gluster？

Gluster是一个可扩展的分布式文件系统，可将来自多个服务器的磁盘存储资源聚合到一个全局命名空间中。可根据您的存储消耗需求快速配置额外的存储。它将自动故障转移作为主要功能。

### 1.2 优点

1. 可扩展到几千兆字节
2. 处理成千上万的客户
3. POSIX兼容
4. 使用商用硬件
5. 可以使用任何支持扩展属性的ondisk文件系统
6. 可使用NFS和SMB等行业标准协议访问
7. 提供复制，配额，地理复制，快照和位置检测
8. 允许针对不同工作负载的优化
9. 开源

企业可以根据需求扩展规模、容量、性能，没有供应商锁定、跨企业预置型、公共

云和混合环境。Gluster用于数千家企业的生产，涵盖媒体，医疗保健，政府，教育，Web 2.0和金融服务。

# 二、安装GlusterFS - 快速入门指南

本文件的目的

本文档旨在提供首次设置GlusterFS的分步指南。出于本指南的目的，需要使用Fedora 26（或更高版本）虚拟机实例。

如果您想要更详细的演练，其中包含使用不同方法（在本地虚拟机，EC2和裸设备中）和不同发行版进行安装的说明，请查看安装指南。

使用Ansible部署和管理GlusterFS

如果您已经是Ansible用户，并且更习惯使用Ansible设置分布式系统，我们建议您跳过所有这些并转移到gluster-ansible存储库，它提供了大部分细节以使系统运行得更快。

## 2.1 部署GlusterFS

#### 第1步 - 至少有三个节点

1）Fedora 26（或更高版本）上名为“server1”，“server2”和“server3”的3个节点；

2）一个有效的网络连接；

3）每个虚拟机上至少有两个虚拟磁盘，一个用于操作系统安装，另一个用于服务GlusterFS存储（sdb）。这将模拟真实世界的部署，您可能希望将GlusterFS存储与操作系统安装分开。

4）在每台服务器上设置NTP，以便在文件系统之上正常运行许多应用程序。

注意：GlusterFS将其动态生成的配置文件存储在/var/lib/glusterd。如果在任何时候GlusterFS无法写入这些文件（例如，当后备文件系统已满时），它将至少导致系统的不稳定行为; 或者更糟糕的是，让您的系统完全脱机。建议为目录创建单独的分区，/var/log以减少发生这种情况的可能性。

#### 第2步 - 格式化并安装块

在所有节点上执行此步骤，“server {1,2,3}”

注意：我们将使用XFS文件系统作为后端块。但Gluster旨在处理任何支持扩展属性的文件系统。

以下示例假定brick将驻留在/ dev / sdb1上。

mkfs.xfs -i size=512 /dev/sdb1

mkdir -p /data/brick1

echo '/dev/sdb1 /data/brick1 xfs defaults 1 2' >> /etc/fstab

mount -a && mount

您现在应该看到sdb1挂载在/ data / brick1

#### 第3步 - 安装GlusterFS

安装软件

yum install glusterfs-server

启动GlusterFS管理守护程序：

service glusterd start

service glusterd status

glusterd.service - LSB: glusterfs server

Loaded: loaded (/etc/rc.d/init.d/glusterd)

Active: active (running) since Mon, 13 Aug 2012 13:02:11 -0700; 2s ago

Process: 19254 ExecStart=/etc/rc.d/init.d/glusterd start (code=exited, status=0/SUCCESS)

CGroup: name=systemd:/system/glusterd.service

├ 19260 /usr/sbin/glusterd -p /run/glusterd.pid

├ 19304 /usr/sbin/glusterfsd --xlator-option georep-server.listen-port=24009 -s localhost...

└ 19309 /usr/sbin/glusterfs -f /var/lib/glusterd/nfs/nfs-server.vol -p /var/lib/glusterd/...

#### 第4步 - 配置防火墙

节点上的gluster进程需要能够相互通信。要简化此设置，请在每个节点上配置防火墙以接受来自其他节点的所有流量。

iptables -I INPUT -p all -s <ip-address> -j ACCEPT

其中ip-address是另一个节点的地址。

#### 第5步 - 配置信任池

来自“server1”

gluster peer probe server2

gluster peer probe server3

注意：使用主机名时，需要从另一台服务器探测第一 台服务器以设置其主机名。

来自“server2”

gluster peer probe server1

注意：建立此池后，只有受信任的成员才能将新服务器探测到池中。新服务器无法探测池，必须从池中进行探测。

检查server1上的对等体状态

gluster peer status

你应该看到这样的东西（UUID会有所不同）

Number of Peers: 2

Hostname: server2

Uuid: f0e7b138-4874-4bc0-ab91-54f20c7068b4

State: Peer in Cluster (Connected)

Hostname: server3

Uuid: f0e7b138-4532-4bc0-ab91-54f20c701241

State: Peer in Cluster (Connected)

#### 第6步 - 设置GlusterFS卷

在所有服务器上：

mkdir -p /data/brick1/gv0

从任何单一服务器：

gluster volume create gv0 replica 3 server1:/data/brick1/gv0 server2:/data/brick1/gv0 server3:/data/brick1/gv0

gluster volume start gv0

确认卷显示“已启动”：

gluster volume info

您应该看到类似的内容（卷ID会有所不同）：

Volume Name: gv0

Type: Replicate

Volume ID: f25cc3d8-631f-41bd-96e1-3e22a4c6f71f

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: server1:/data/brick1/gv0

Brick2: server2:/data/brick1/gv0

Brick3: server3:/data/brick1/gv0

Options Reconfigured:

transport.address-family: inet

注意：如果卷未显示“已启动”，/var/log/glusterfs/glusterd.log则应检查下面的文件 以便调试和诊断情况。可以在配置的一个或所有服务器上查看这些日志。

#### 第7步 - 测试GlusterFS卷

对于此步骤，我们将使用其中一个服务器来装入卷。通常，您可以从外部计算机（称为“客户端”）执行此操作。由于使用此方法需要在客户端计算机上安装其他软件包，因此我们将使用其中一个服务器作为首先进行测试的简单位置，就像它是“客户端”一样。

mount -t glusterfs server1:/gv0 /mnt

for i in `seq -w 1 100`; do cp -rp /var/log/messages /mnt/copy-test-$i; done

首先，检查客户端挂载点：

ls -lA /mnt/copy\* | wc -l

您应该看到返回100个文件。接下来，检查每台服务器上的GlusterFS块安装点：

ls -lA /data/brick1/gv0/copy\*

您应该使用我们在此处列出的方法在每台服务器上看到100个文件。如果没有复制，在仅分发卷（此处未详述）中，您应该在每个卷上看到大约33个文件。

## 2.2 安装Gluster

对于基于RPM的分发，如果您将使用InfiniBand，请将glusterfs RDMA包添加到安装中。对于基于RPM的系统，yum / dnf用作安装方法以满足外部依赖性，例如compat-readline5

#### 对于Debian

将GPG密钥添加到apt：

wget -O - http://download.gluster.org/pub/gluster/glusterfs/LATEST/rsa.pub | apt-key add -

如果rsa.pub在上述位置不可用，请在此处查看https://download.gluster.org/pub/gluster/glusterfs/3.12/rsa.pub并将GPG密钥添加到apt：

wget -O - https://download.gluster.org/pub/gluster/glusterfs/3.12/rsa.pub | apt-key add -

添加来源：

DEBID=$(**grep** 'VERSION\_ID=' /etc/os-release | cut -d '=' -f 2 | **tr** -d '"')

DEBVER=$(**grep** 'VERSION=' /etc/os-release | **grep** -Eo '[a-z]+')

DEBARCH=$(dpkg --**print**-architecture)

echo deb https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/${DEBID}/${DEBARCH}/apt ${DEBVER} main > /etc/apt/sources.list.d/gluster.list

更新包列表：

apt-**get** update

安装：

apt-get **install** glusterfs-**server**

#### 对于Ubuntu

安装软件 - 属性 - 常见：

sudo apt-**get** install software-properties-common

然后添加社区GlusterFS PPA：

sudo **add**-apt-repository ppa:gluster/glusterfs-3.8

sudo apt-**get** update

最后，安装包：

sudo apt-get **install** glusterfs-**server**

注意：Ubuntu 12.04 LTS，12.10,13.10和14.04 LTS存在包

#### 对于Red Hat / CentOS

CentOS和其他RHEL克隆的RPM可从CentOS Storage SIG镜像获得。

有关更多安装详细信息，请参阅CentOS Storage SIG的[Gluster快速入门指南](https://wiki.centos.org/SpecialInterestGroup/Storage/gluster-Quickstart)。

#### 对于Fedora

安装Gluster包：

dnf **install** glusterfs-**server**

完成安装后，您可以继续进行[配置](https://docs.gluster.org/en/latest/Install-Guide/Configure/)部分。

#### 对于Arch Linux

安装Gluster包：

pacman -S glusterfs

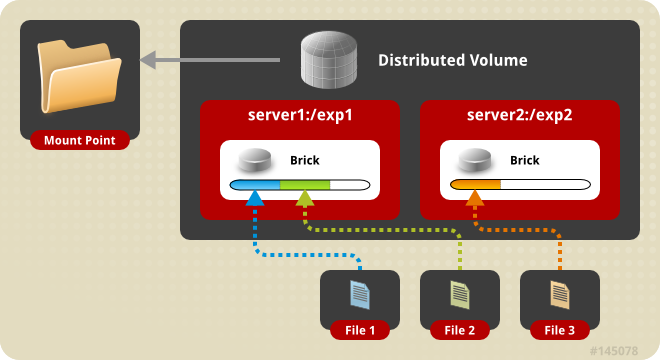
# 三、架构

### 3.1、卷的类型

卷是块的集合，大多数gluster文件系统操作都发生在卷上。Gluster文件系统根据要求支持不同类型的卷。有些卷适用于扩展存储大小，有些用于提高性能，有些用于提高性能。

#### 1.1**分布式Glusterfs卷**

这是默认的glusterfs卷，即，如果未指定卷的类型，则在创建卷时，默认选项是创建分布式卷。这里，文件分布在卷中的各个块中。因此file1可能只存储在brick1或brick2中，但不能存储在两者中。因此没有数据冗余。这种存储卷的优点是扩展容易和廉价。然而，这也意味着砖块故障将导致数据完全丢失，并且必须依靠底层硬件来保护数据丢失。



创建分布式卷

**gluster volume create NEW-VOLNAME [transport [tcp | rdma | tcp，rdma]] NEW-BRICK ......**

**例如**，使用TCP创建具有四个存储服务器的分布式卷。

gluster volume create test-volume server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4

Creation of test-volume has been successful

Please start the volume to access data

显示卷信息

#gluster volume info

Volume Name: test-volume

Type: Distribute

Status: Created

Number of Bricks: 4

Transport-type: tcp

Bricks:

Brick1: server1:/exp1

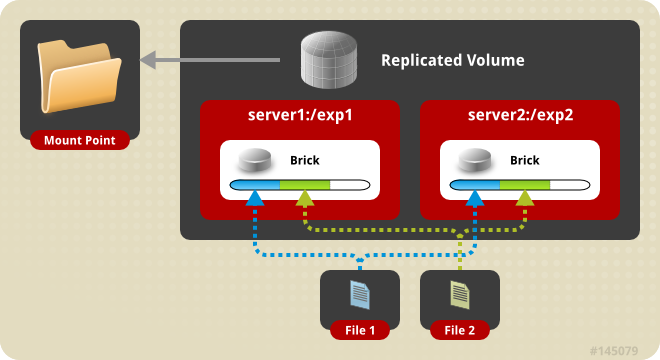
Brick2: server2:/exp2

Brick3: server3:/exp3

Brick4: server4:/exp4

#### 1.2 **复制Glusterfs卷**

在本卷中，我们克服了分布式卷中面临的数据丢失问题。在这里，所有砖块都保留了精确的数据副本。在创建卷时，客户端可以决定卷中的副本数。所以我们需要至少有两块块来创建一个包含2个复制品或至少3个块的卷来创建3个复制品。这种卷的一个主要优点是，即使一块砖出现故障，仍然可以从其复制的块访问数据。这样的卷的优点是可靠性和数据冗余。



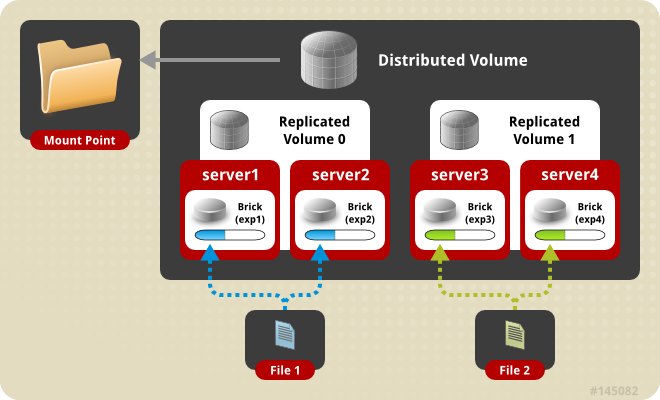
创建复制卷

**gluster volume create NEW-VOLNAME [replica COUNT] [transport [tcp | rdma | tcp，rdma]] NEW-BRICK ......**

例如，要创建具有两个存储服务器的复制卷：gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2

#### 1.3 **分布式复制Glusterfs卷**

此卷文件分布在复制的块集中。块数量必须是副本数量的倍数。此外，我们指定砖的顺序很重要，因为相邻的砖成为彼此的复制品。当需要由于冗余和扩展存储而导致的数据高可用性时，使用这种类型的卷。因此，如果有8个块和副本计数2，则前两个块将成为彼此的复制品，然后是接下来的两个块，依此类推。该体积表示为4×2。同样，如果有8块和4块复制品，则4块成为彼此的复制品，我们将该体积表示为2x4体积。



创建分布式复制卷：

**#gluster volume create NEW-VOLNAME [replica COUNT] [transport [tcp | rdma | tcp，rdma]] NEW-BRICK ......**

**例如**，带有双向镜像的四个节点分布式（复制）卷：

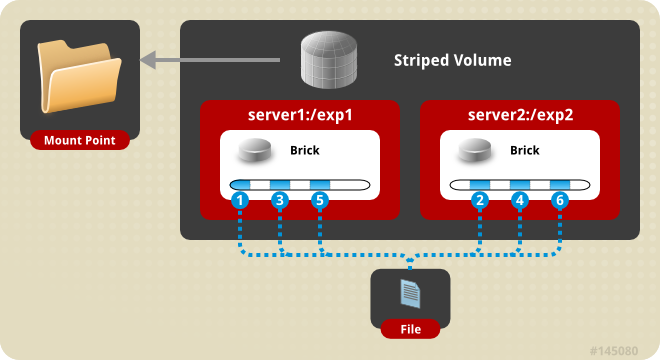
# gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4

Creation of test-volume has been successful

Please start the volume to access data

#### 1.4 **条纹Glusterfs卷**

考虑将大型文件存储在砖块中，许多客户端经常同时访问该块。这将导致单块砖上的负载过大，从而降低性能。在条带化体积中，数据在将其分成不同的条带后存储在砖块中。因此，大文件将被分成较小的块（等于卷中的块数），每个块存储在一块砖中。现在可以分配负载，并且可以更快地获取文件，但不提供数据冗余。



创建条带卷

gluster volume create NEW-VOLNAME [stripe COUNT] [transport [tcp | dma | tcp,rdma]] NEW-BRICK...

**例如**，要跨两个存储服务器创建条带卷：

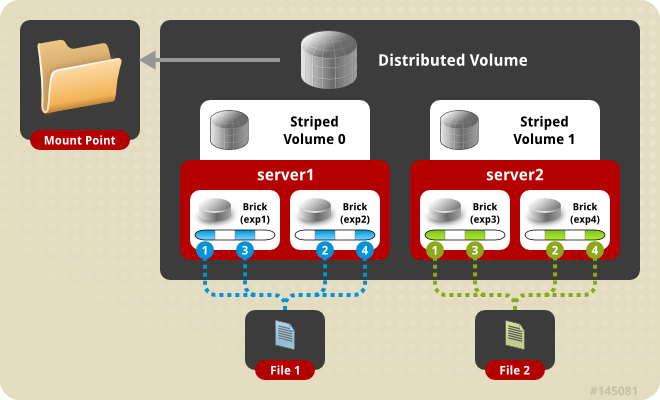
# gluster volume create test-volume stripe 2 transport tcp server1:/exp1 server2:/exp2

Creation of test-volume has been successful

Please start the volume to access data

#### 1.5 **分布式条纹Glusterfs卷**

类似于条纹Glusterfs卷，除了条纹现在可以分布在更多数量的砖块上。但是块的数量必须是条纹数量的倍数。因此，如果我们想要增加卷大小，我们必须在多个条带数中添加块。



创建分布式条带卷：

**gluster volume create NEW-VOLNAME [stripe COUNT] [transport [tcp | rdma | tcp，rdma]] NEW-BRICK ......**

例如，要在八个存储服务器上创建分布式条带卷：

# gluster volume create test-volume stripe 4 transport tcp

 server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4 server5:/exp5 server6:/exp6 server7:/exp7 server8:/exp8

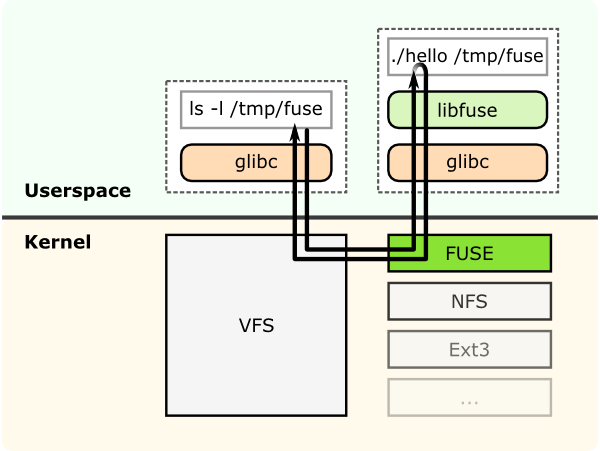
Creation of test-volume has been successful

Please start the volume to access data.

### 3.2、FUSE

GlusterFS是一个用户空间文件系统。这是GlusterFS开发人员最初做出的决定，因为将模块放入Linux内核是一个非常漫长而艰难的过程。

作为用户空间文件系统，GlusterFS与内核VFS交互，利用FUSE（用户空间中的文件系统）。很长一段时间，用户空间文件系统的实现被认为是不可能的。FUSE是为此而开发的解决方案。FUSE是一个内核模块，支持内核VFS和非特权用户应用程序之间的交互，它有一个可以从用户空间访问的API。使用此API，可以使用您喜欢的几乎任何语言编写任何类型的文件系统，因为FUSE和其他语言之间存在许多绑定。



FUSE的结构图

这显示了一个文件系统“hello world”，它被编译为创建二进制“hello”。它使用文件系统安装点/ tmp / fuse执行。然后用户在挂载点/ tmp / fuse上发出命令ls -l。此命令通过glibc到达VFS，并且由于mount / tmp / fuse对应于基于FUSE的文件系统，因此VFS将其传递给FUSE模块。FUSE内核模块在通过用户空间（libfuse）中的glibc和FUSE库后联系实际的文件系统二进制文件“hello”。结果由“hello”通过相同的路径返回并到达ls -l命令。

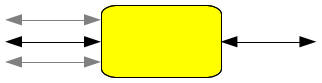
FUSE内核模块和FUSE库（libfuse）之间的通信是通过打开/ dev / fuse获得的特殊文件描述符。可以多次打开此文件，并将获取的文件描述符传递给mount syscall，以使描述符与挂载的文件系统相匹配。

### 3.3、转换器

#### **3.1 什么是“转换器”？**

1）转换器将来自用户的请求转换为存储请求。

\*一对一，一对多，一对零（例如缓存）



2）转换器可以通过以下方式修改请求：

将一种请求类型转换为另一种请求类型（在转换器之间的请求传输期间） 修改路径，标志，甚至数据（例如加密）

3）转换器可以拦截或阻止请求。（例如访问控制）

1. 或者产生新的请求（例如预取）

#### Translators**如何运作？**

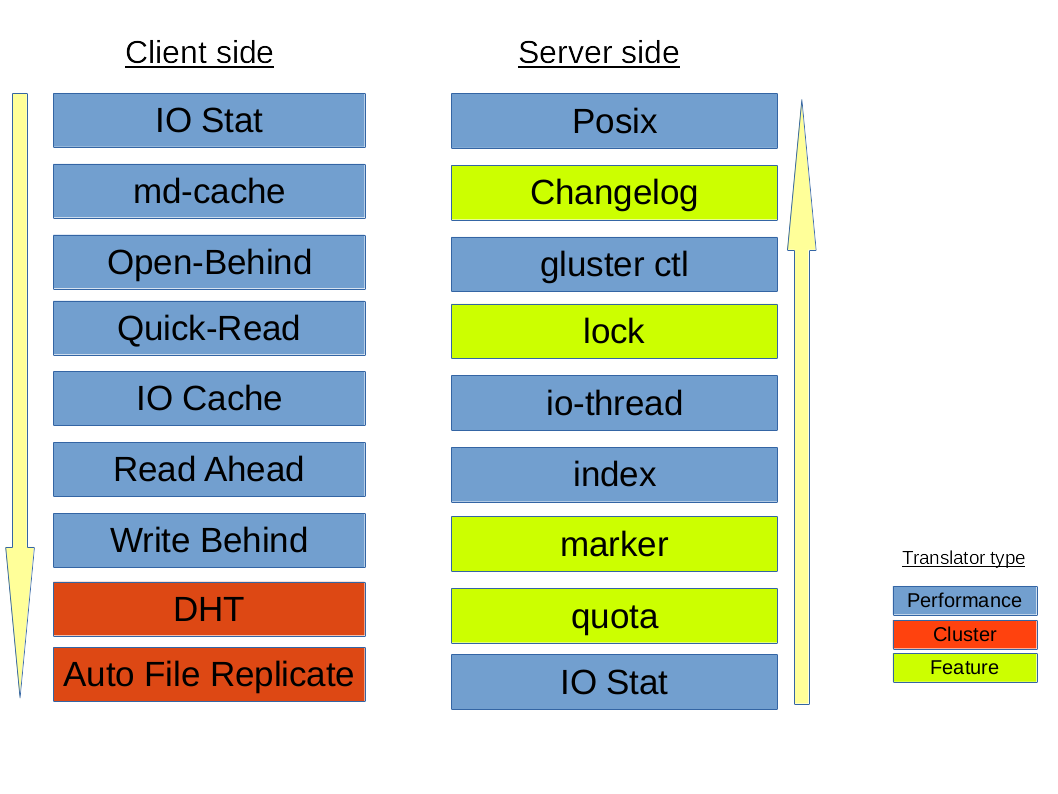
1. 共享对象；
2. 根据'volfile'动态加载；dlopen / dlsync 设置指向parent / children的指针 调用init（构造 函数）通过​​fops调用IO函数。
3. 验证/通过选项等的约定；
4. 转换器的配置（由于GlusterFS 3.1）是通过gluster命令行界面（cli）管理的，因此您无需知道将翻译器组合在一起的顺序。

#### 3.3 Translator的类型

已知的**Translators**及其当前状态。

| **Translator类型** | **功能** |
| --- | --- |
| 存储 | 最低级别的转换器，存储和访问本地文件系统中的数据。 |
| 调试 | 提供错误和调试的接口和统计信息。 |
| 簇 | 处理与砖块和节点的写入和读取相关的数据分发和复制。 |
| 加密 | 扩展转换器，用于存储数据的即时加密/解密。 |
| 协议 | 用于客户端/服务器通信协议的扩展转换器。 |
| 性能 | 调整转换器以调整工作负载和I / O配置文件。 |
| 绑定 | 添加可扩展性，例如Jeff Darcy编写的Python接口，用于扩展与GlusterFS的API交互。 |
| 系统 | 系统访问转换器，例如与文件系统访问控制接口。 |
| 调度 | I / O调度程序，用于确定如何跨集群系统分发新的写入操作。 |
| 特征 | 添加其他功能，例如配额，过滤器，锁等。 |

vol文件中转换器的默认/一般层次结构：



所有连接在一起执行功能的转换器称为图形。左侧转换器包括**客户端栈**。右侧转换器包括**服务器栈**。

**glusterfs Translator可以细分为多个类别，但两个重要的类别是：Cluster和Performance Translator：**

数据/请求必须经历的最重要的一个和第一个**Translator**是**FuseTranslator**，属于**Mount Translators**类别 。

1. **Cluster Translators**：

\* DHT（分布式哈希表）—— Distributed Hash Table

\* AFR（自动文件复制）—— Automatic File Replication

1. **Performance Translators**：

\* io-cache

\* io-threads

\* md-cache

\* OB（后读取）

\* QR（快速读取）

\* ra（预读）

\* wb（后写）

其他**Feature Translators**包括：

\*更改日志

\* locks ：GlusterFS具有锁定转换器，它提供以下内部锁定操作  inodelk，  entrylk被afr用来实现对彼此冲突的文件或目录的操作同步。

\*标记

\*配额

**Debug Translators**

\* trace - To trace the error logs generated during the communication amongst the translators.

\* io-stats

#### 3.4 DHT（分布式哈希表）Translator

##### **3.4.1 什么是DHT？**

DHT（Distributed Hash Table）是GlusterFS如何在多个服务器之间聚合容量和性能的真正核心。它的职责是将每个文件放在其子卷中的一个上，不像复制（将副本放在所有子卷上）或条带化（将块放在其所有子卷上）。这是一个路由功能，而不是分割或复制。

##### **3.4.2 DHT如何运作**？

DHT中使用的基本方法是一致哈希。每个子卷（块）在32位散列空间内分配一个范围，整个范围没有孔或重叠。然后通过对其名称进行散列，在同一空间中为每个文件分配一个值。每一个砖将具有指定的范围，包括文件的哈希值，因此文件“应该”在相应的块上。但是，在许多情况下情况并非如此，例如，当块集发生更改在创建文件后，或当块接近满时。DHT的大部分复杂性涉及这些特殊情况，我们将在稍后讨论。

当您打开文件时，distribute translator会提供一条信息来查找您的文件，即文件名。要确定该文件的位置，translator通过散列算法运行文件名，以便将该文件名转换为数字。

**DHT哈希值赋值的一些观察**：

1. 散列范围到砖块的分配由存储在目录上的扩展属性确定，因此分布是特定于目录的。
2. 一致性哈希通常被认为是围绕圆圈的散列，但在GlusterFS中它更线性。没有必要在零处“环绕”，因为在零处总是有一个中断（在一个块的范围和另一个块的范围之间）。
3. 如果缺少块，则哈希空间中会有一个洞。更糟糕的是，如果在块处于脱机状态时重新分配哈希范围，则某些新范围可能会与存储在该块上的（现在已过时）范围重叠，从而对文件的位置产生一些混淆。

#### 3.5 AFR（自动文件复制）转换器

GlusterFS中的自动文件复制（AFR—Automatic File Replication）Translator利用扩展属性来跟踪文件操作。它负责跨块复制数据。

##### AFR的作用

包括以下内容：

1. 保持复制一致性（即两块上的数据应该相同，即使在多个应用程序/挂载点并行运行同一文件/目录上的操作的情况下，只要副本集中的所有块都已启动）。
2. 只要存在至少一个具有正确数据的块，就提供一种在出现故障时恢复数据的方法。
3. 为read / stat / readdir等提供新数据。

### 3.4、地理复制

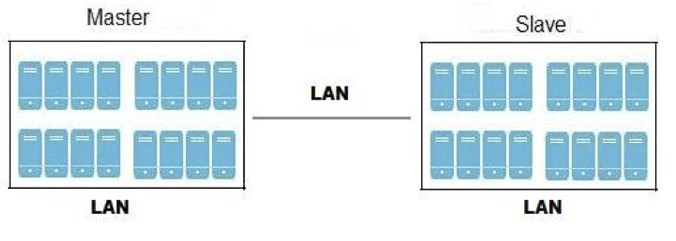
地理复制提供跨地理位置不同位置的异步数据复制，并在Glusterfs 3.2中引入。它主要在WAN上工作，用于复制整个卷，而不像AFR那样是集群内复制。这主要用于备份整个数据以进行灾难恢复。

地理复制使用主从模型，从而在**Master**（GlusterFS卷）和**Slave**（可以是本地目录或GlusterFS卷）之间进行复制。Slave使用SSH隧道访问本地目录或卷。

地理复制通过局域网（LAN），广域网（WAN）和Internet提供增量复制服务。

**1）通过LAN进行地理复制**

您可以配置地理复制以通过局域网镜像数据。



**2）通过WAN进行地理复制**

您可以配置地理复制以通过广域网复制数据。



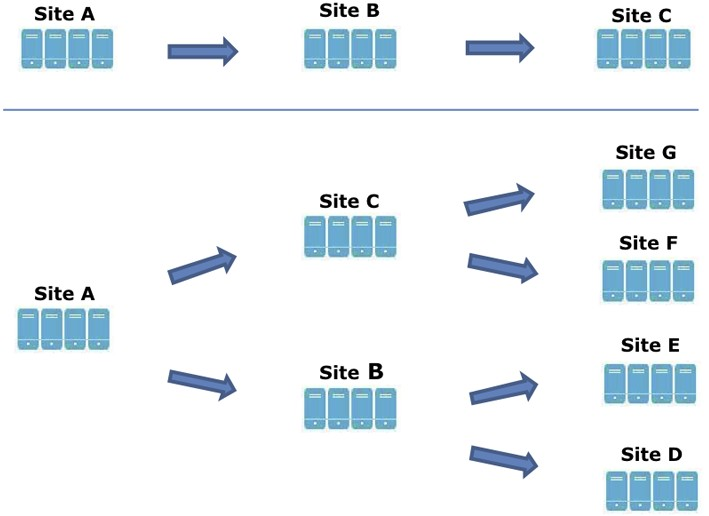
**3）通过Internet进行地理复制**

您可以配置地理复制以通过Internet镜像数据。



**4）多站点级联地理复制**

您可以配置地理复制，以跨多个站点以级联方式镜像数据。



异步复制数据主要有两个方面：

1）**更改检测** - 这些包括文件操作必需的详细信息。有两种方法可以同步检测到的更改：

a）更改日志 ： 更改日志是一个translator，它记录发生fops的必要的详细信息。这些更改可以以二进制格式或ASCII格式编写。每个类别有三个类别，由特定的更改日志格式表示。所有三种类别都记录在一个changelog文件中。

**Entry** ： create（），mkdir（），mknod（），symlink（），link（），rename（），unlink（），rmdir（）

**数据** ：write（），writev（），truncate（），ftruncate（）

**Meta** ： setattr（），fsetattr（），setxattr（），fsetxattr（），removexattr（），fremovexattr（）

为了记录操作类型和接受的实体，使用类型标识符。通常，执行操作的实体将由路径名标识，但我们选择使用GlusterFS内部文件标识符（GFID）（因为GlusterFS支持基于GFID的后端，并且路径名字段可能并不总是有效以及其他原因，超出了本文档的范围）。因此，三种操作的记录格式可归纳如下：

Entry ： GFID + FOP +模式+ UID + GID + PARGFID / BNAME [PARGFID / BNAME]

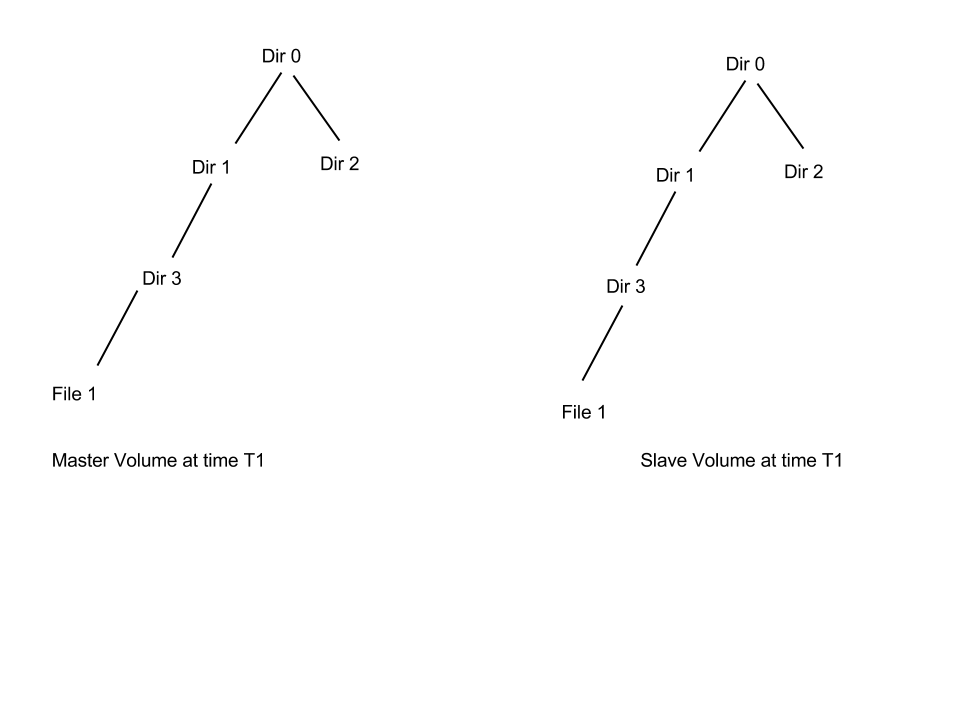
Meta ：文件的GFID

Data ：文件的GFID

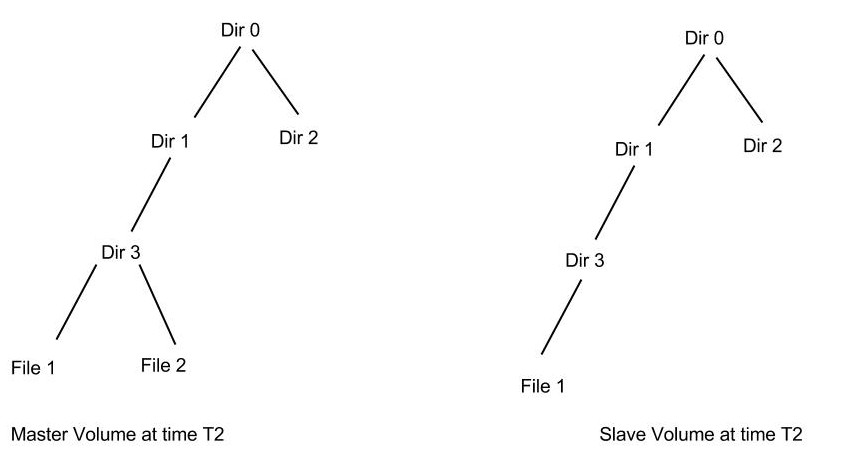
GFID类似于inodes。Data和Meta fops记录执行操作实体的GFID，从而记录inode上的数据/元数据更改。Entry fops至少记录一组六或七条记录（取决于操作类型），足以识别实体所经历的操作类型。通常，此记录包括实体的GFID，文件操作的类型（它是一个整数[在Glusterfs中使用的枚举值]）以及父GFID和basename（类似于父inode和basename）。

更改日志文件在特定时间间隔后翻转。然后，我们对文件执行处理操作，例如将其转换为可理解/可读的格式，保留更改日志的私有副本等。然后，库将使用这些日志并提供应用程序请求。

b）Xsync ：Marker Translator为每个文件和目录维护扩展属性“xtime”。每当发生任何更新时，它都会更新该文件及其所有祖先的xtime属性。因此，更改会从节点（发生更改的位置）一直传播到根节点。



考虑上面的目录树结构。在时间T1，主设备和从设备彼此同步。



在时间T2，创建了新文件File2。这将触发从File2到root的xtime标记（其中xtime是当前时间戳），即File2，Dir3，Dir1和最后Dir0的xtime都将被更新。

地理复制守护程序根据xtime（master）> xtime（slave）的条件对文件系统进行爬取。因此，在我们的示例中，它将仅爬取目录结构的左侧部分，因为目录结构的右侧部分仍具有相同的时间戳。虽然爬取算法很快，但我们仍然需要抓取目录结构的很大一部分。

2）**复制** - 我们使用rsync进行数据复制。Rsync是一个外部实用程序，它将计算两个文件的差异，并将此差异从源发送到同步。

### 3.5、GlusterFS的整体工作

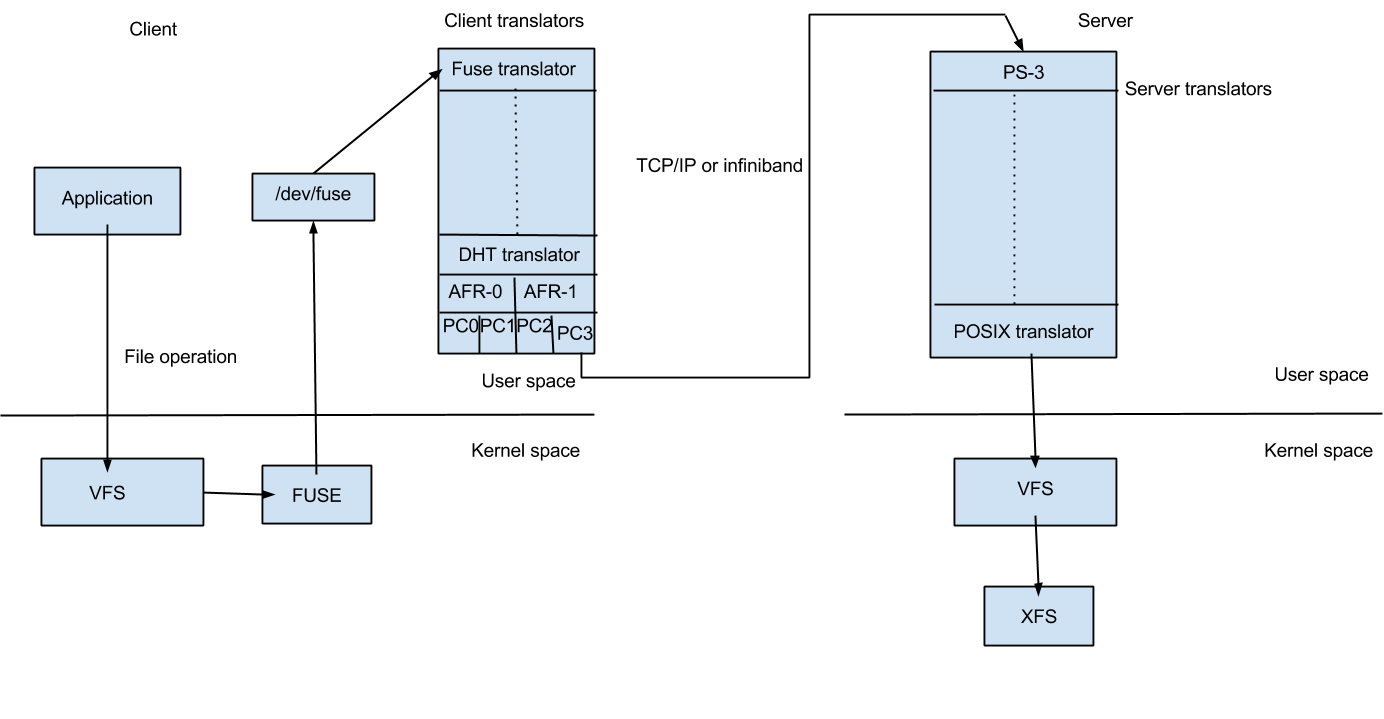
只要GlusterFS安装在服务器节点中，就会创建一个gluster管理守护程序（glusterd）二进制文件。此守护程序应该在群集中的所有参与节点中运行。启动glusterd后，可以创建一个包含所有存储服务器节点的可信服务器池（TSP）（TSP甚至可以包含单个节点）。现在可以将作为基本存储单元的块创建为这些服务器中的导出目录。来自这个TSP的任何数量的块都可以组合在一起形成一个卷。

创建卷后，glusterfsd进程将在每个参与的块中开始运行。除此之外，还将在/ var / lib / glusterd / vols /中生成称为vol文件的配置文件。卷中的每个块都有对应的配置文件。这将包含有关该特定块的所有详细信息。还将包含创建客户端进程所需的配置文件。现在我们的文件系统已经可以使用了。我们可以非常轻松地在客户端计算机上安装此卷，如下所示，并像使用本地存储一样使用它：

mount.glusterfs `<IP or hostname>`:`<volume\_name>` `<mount\_point>

IP或主机名可以是可信服务器池中创建所需卷的任何节点的IP或主机名。

当我们在客户端中安装卷时，客户端glusterfs进程与服务器的glusterd进程通信。服务器glusterd进程发送包含客户端转换器列表的配置文件（vol文件），另一个包含卷中每个块的信息，客户端glusterfs进程现在可以直接与每个块的glusterfsd进程通信。设置现已完成，现在该卷已准备好供客户端服务。



当客户端在挂载的文件系统中发出系统调用（文件操作或Fop）时，VFS（标识文件系统的类型为glusterfs）将把请求发送到FUSE内核模块。FUSE内核模块将通过/ dev / fuse将其发送到客户机节点的用户空间中的GlusterFS（这已在FUSE部分中描述）。客户端上的GlusterFS进程由一组称为client translator的translator组成，这些translator在存储服务器glusterd进程发送的配置文件（vol文件）中定义。这些translators中的第一个是FUSE translator，它由FUSE库（libfuse）组成。每个translator都具有与glusterfs支持的每个文件操作或fop相对应的功能。该请求将在每个translator中达到相应的功能。

1. FUSE translator
2. DHT translator – DHT translator将请求映射到包含所需文件或目录的正确块。
3. AFR translator - 它接收来自前一个translator的请求，如果卷类型是复制的，它会复制请求并将其传递给副本的protocol client translator。
4. Protocol Client translator - Protocol Client translator是client translator堆栈中的最后一个。该translator分为多个线程，每个线程对应一个卷。这将直接与每块的glusterfsd通信。

在包含需要的块的存储服务器节点中，请求再次通过一系列称为服务器转换器的转换器，主要是：

1. 协议服务器转换器
2. POSIX翻译

请求最终将到达VFS，然后将与底层本机文件系统进行通信。响应将回溯相同的路径。

# 四、管理

## 管理集群

### 4.1.1 管理Gluster服务

安装GlusterFS后，必须启动glusterd服务。glusterd服务充当Gluster弹性卷管理

器，监督glusterfs进程，协调动态卷操作，例如无中断地跨多个存储服务器添加和删除卷。

手动启动和停止glusterd

要手动启动glusterd，请输入以下命令：

# /etc/init.d/glusterd start

要手动停止glusterd，请输入以下命令：

# /etc/init.d/glusterd stop

自动启动glusterd

Red Hat和Fedora发行版

# chkconfig glusterd on

Debian和Ubuntu等衍生品

# update-rc.d glusterd defaults

Red Hat和Debian以外的系统

# echo "glusterd" >> /etc/rc.local

### 4.1.2 管理受信任的存储池

可信存储池（TSP）是存储服务器的可信网络。在配置GlusterFS卷之前，必须创建存储服务器的可信存储池，该存储服务器将通过对等方探测服务器为卷提供块。TSP中的服务器是彼此的对等体。

在服务器上安装Gluster之后，在创建可信存储池之前，每个服务器都属于仅包含该服务器的存储池。

用于创建存储池的服务器必须可通过主机名解析。

glusterd守护程序必须在要添加到存储池的所有存储服务器上运行。

必须将服务器上的防火墙配置为允许访问端口24007。

以下命令在由3个服务器组成的TSP上运行 - server1，server2和server3。

#### 4.1.2.1 添加服务器

要将服务器添加到TSP，请从池中已有的服务器进行对等探测。

# gluster peer probe <server>

例如，要将新服务器4添加到上述群集，请从其他服务器之一进行探测：

server1# gluster peer probe server4

Probe successful

验证第一台服务器（server1）的对等体状态：

server1# gluster peer status

Number of Peers: 3

Hostname: server2

Uuid: 5e987bda-16dd-43c2-835b-08b7d55e94e5

State: Peer in Cluster (Connected)

Hostname: server3

Uuid: 1e0ca3aa-9ef7-4f66-8f15-cbc348f29ff7

State: Peer in Cluster (Connected)

Hostname: server4

Uuid: 3e0cabaa-9df7-4f66-8e5d-cbc348f29ff7

State: Peer in Cluster (Connected)

#### 4.1.2.2 列表服务器

列出TSP中的所有节点：

server1# gluster pool list

UUID Hostname State

d18d36c5-533a-4541-ac92-c471241d5418 localhost Connected

5e987bda-16dd-43c2-835b-08b7d55e94e5 server2 Connected

1e0ca3aa-9ef7-4f66-8f15-cbc348f29ff7 server3 Connected

3e0cabaa-9df7-4f66-8e5d-cbc348f29ff7 server4 Connected

#### 4.1.2.3 查看对等方状态

要查看TSP中对等方的状态：

server1# gluster peer status

Number of Peers: 3

Hostname: server2

Uuid: 5e987bda-16dd-43c2-835b-08b7d55e94e5

State: Peer in Cluster (Connected)

Hostname: server3

Uuid: 1e0ca3aa-9ef7-4f66-8f15-cbc348f29ff7

State: Peer in Cluster (Connected)

Hostname: server4

Uuid: 3e0cabaa-9df7-4f66-8e5d-cbc348f29ff7

State: Peer in Cluster (Connected)

#### 4.1.2.4 删除服务器

要从TSP中删除服务器，请从池中的另一台服务器运行以下命令：

# gluster peer detach <server>

例如，要从受信任存储池中删除server4：

server1# gluster peer detach server4

Detach successful

验证对等体状态：

server1# gluster peer status

Number of Peers: 2

Hostname: server2

Uuid: 5e987bda-16dd-43c2-835b-08b7d55e94e5

State: Peer in Cluster (Connected)

Hostname: server3

Uuid: 1e0ca3aa-9ef7-4f66-8f15-cbc348f29ff7

State: Peer in Cluster (Connected)

## 4.2设置存储

卷是块的逻辑集合，其中每个块是可信存储池中服务器上的导出目录。在创建卷之前，您需要设置将形成卷的块。

### [块命名约定](https://docs.gluster.org/en/latest/Administrator%20Guide/Brick%20Naming%20Conventions/)

FHS-2.3并不完全清楚服务器共享的数据应驻留在何处。它确实声明“ / srv包含由此系统提供的特定于站点的数据 ”，但GlusterFS数据是否特定于站点？

共识似乎倾向于使用/data。放置块的良好分层方法是：

/data/glusterfs/<volume>/<brick>/brick

在此示例中，<brick>是已挂载的文件系统。

示例：每个服务器一个块

物理磁盘/ dev / sdb将用作您要创建名为myvol1的卷的块存储。您已经在4台服务器的每一台上使用XFS对/ dev / sdb1进行了分区和格式化。

在所有4台服务器上：

mkdir -p /data/glusterfs/myvol1/brick1

mount /dev/sdb1 /data/glusterfs/myvol1/brick1

我们将brick在该文件系统的目录中定义实际的块。如果未安装XFS文件系统，则会导致块无法启动。

仅在一台服务器上：

gluster volume create myvol1 replica 2 server{1..4}:/data/glusterfs/myvol1/brick1/brick

这将创建卷myvol1，它使用/data/glusterfs/myvol1/brick1/brick所有4台服务器上的目录。

示例：每个服务器两个砖

两个物理磁盘/ dev / sdb和/ dev / sdc将用作您要创建的名为myvol2的卷的块存储。您已经在4台服务器上分别格式化了/ dev / sdb1和/ dev / sdc1以及XFS。

在所有4台服务器上：

mkdir -p /data/glusterfs/myvol2/brick{1,2}

mount /dev/sdb1 /data/glusterfs/myvol2/brick1

mount /dev/sdc1 /data/glusterfs/myvol2/brick2

我们将再次brick在这些文件系统的目录中定义实际的块。

仅在一台服务器上：

gluster volume create myvol2 replica 2 \

server{1..4}:/data/glusterfs/myvol2/brick1/brick \

server{1..4}:/data/glusterfs/myvol2/brick2/brick

注意：gluster volume create myvol2 replica 2 server{1..4}:/data/glusterfs/myvol2/brick{1,2}/brick， Bash会扩展最后{}一个，所以最终会在每个服务器上的两块之间复制，而不是跨服务器复制。

### 4.2.2格式化和安装块

要创建精简配置的逻辑卷，请继续执行以下步骤：

使用pvcreate命令创建物理卷（PV）。例如：

# pvcreate --dataalignment 1280K /dev/sdb

这里，/ dev / sdb是一个存储设备。根据您的设备使用正确的数据对齐选项。

注意：设备名称和对齐值将根据您使用的设备而有所不同。

使用vgcreate命令从PV创建卷组（VG）：例如：

# vgcreate --physicalextentsize 128K gfs\_vg /dev/sdb

建议只能从一个存储设备创建一个VG。

使用以下命令创建精简池：

使用以下命令创建LV以充当元数据设备：

# lvcreate -L metadev\_sz --name metadata\_device\_name VOLGROUP

例如：

# lvcreate -L 16776960K --name gfs\_pool\_meta gfs\_vg

使用以下命令创建LV作为数据设备：

# lvcreate -L datadev\_sz --name thin\_pool VOLGROUP

例如：

# lvcreate -L 536870400K --name gfs\_pool gfs\_vg

使用以下命令从数据LV和元数据LV创建精简池：

# lvconvert --chunksize STRIPE\_WIDTH --thinpool VOLGROUP/thin\_pool --poolmetadata VOLGROUP/metadata\_device\_name

例如：

# lvconvert --chunksize 1280K --thinpool gfs\_vg/gfs\_pool --poolmetadata gfs\_vg/gfs\_pool\_meta

注意：默认情况下，精简池中新配置的块被清零，以防止不同块设备之间的数据泄漏。

# lvchange --zero n VOLGROUP/thin\_pool

例如：

# lvchange --zero n gfs\_vg/gfs\_pool

使用lvcreate命令从先前创建的池创建精简配置卷：

例如：

# lvcreate -V 1G -T gfs\_vg/gfs\_pool -n gfs\_lv

建议在精简池中只创建一个LV。

使用支持的XFS配置格式化砖块，安装块，并验证砖块是否正确安装。

运行# mkfs.xfs -f -i size=512 -n size=8192 -d su=128k,sw=10 DEVICE以将块格式化为支持的XFS文件系统格式。这里，DEVICE是瘦LV。inode大小设置为512字节，以适应GlusterFS使用的扩展属性。

运行# mkdir /mountpoint以创建将块链接到的目录。

在/ etc / fstab中添加一个条目：

/dev/gfs\_vg/gfs\_lv /mountpoint xfs rw,inode64,noatime,nouuid 1 2

运行# mount /mountpoint以安装砖。

运行该df -h命令以验证是否已成功装入块：

# df -h

/dev/gfs\_vg/gfs\_lv 16G 1.2G 15G 7% /exp1

### [POSIX访问控制列表](https://docs.gluster.org/en/latest/Administrator%20Guide/Access%20Control%20Lists/)

POSIX访问控制列表（ACL）允许您为不同的用户或组分配不同的权限，即使它们与原始所有者或拥有组不对应。

例如：用户john创建一个文件，但不希望任何人对此文件执行任何操作，但另一个用户antony除外（即使有其他用户属于该组john）。

这意味着，除文件所有者，文件组和其他人外，还可以使用POSIX ACL授予或拒绝其他用户和组访问权限。

1）激活POSIX ACL支持

要将POSIX ACL用于文件或目录，必须使用POSIX ACL支持挂载文件或目录的分区。

2）在服务器上激活POSIX ACL支持

要为POSIX ACL支持安装后端导出目录，请使用以下命令：

# mount -o acl

例如： # mount -o acl /dev/sda1 /export1

或者，如果分区在/ etc / fstab文件中列出，请为分区添加以下条目以包括POSIX ACLs选项：

LABEL=/work /export1 ext3 rw, acl 14

3）在客户端上激活POSIX ACL支持

要为POSIX ACL支持安装glusterfs卷，请使用以下命令：

# mount –t glusterfs -o acl

例如：

# mount -t glusterfs -o acl 198.192.198.234:glustervolume /mnt/gluster

4）设置POSIX ACL

您可以设置两种类型的POSIX ACL，即访问ACL和默认ACL。您可以使用访问ACL为特定文件或目录授予权限。您只能在目录上使用默认ACL，但如果该目录中的文件没有ACL，则它将继承该目录的默认ACL的权限。

您可以为每个用户，每个组，不在文件的用户组中的用户以及有效的右掩码设置ACL。

5）设置Access ACL

您可以应用Access ACL以授予文件和目录的权限。

设置或修改访问ACL

您可以使用以下命令设置或修改访问ACL：

# setfacl –m file

ACL条目类型是所有者，组和其他的POSIX ACL表示。

权限必须是字符r（读取），w （写入）和x（执行）的组合。您必须以下列格式指定ACL条目，并可以指定以逗号分隔的多个条目类型。

ACL输入 描述：

u：UID：\ < permission> 设置用户的访问ACL。您可以指定用户名或UID

g：GID：\ < permission> 设置组的访问ACL。您可以指定组名称或GID。

m：\ < permission> 设置有效权限掩码。掩码是拥有组的所有访问权限与所有用户和组条目的组合。

o：\ < permission> 为文件组中的用户以外的用户设置访问ACL。

如果文件或目录已具有POSIX ACL，并且使用了setfacl命令，则会将其他权限添加到现有POSIX ACL，或者修改现有规则。

例如，要为用户antony提供读写权限：

# setfacl -m u:antony:rw /mnt/gluster/data/testfile

6）设置默认ACL

您只能将默认ACL应用于目录。它们确定在创建时从其父目录继承的文件系统对象的权限。

您可以使用以下命令为文件和目录设置默认ACL：

# setfacl –m –-set

权限必须是字符r（读取），w（写入）和x（执行）的组合。如下所述指定ACL entry\_type，用逗号分隔多个条目类型。

u：user\_name：permissions 设置用户的访问ACL。指定用户名或UID。

g：group\_name：permissions 设置组的访问ACL。指定组名称或GID。

m：permission 设置有效权限掩码。掩码是拥有组的所有访问权限以及所有用户和组条目的组合。

o：permissions 为文件组中的用户以外的用户设置访问ACL。

例如，要为不在用户组中的用户设置/ data目录的默认ACL：

# setfacl –m --set o::r /mnt/gluster/data

注意：为单个文件设置的访问ACL可以覆盖默认ACL权限。

默认ACL的影响

以下是将目录的默认ACL的权限传递给其中的文件和子目录的方式：

a）子目录继承父目录的默认ACL作为其默认ACL和访问ACL。文件继承默认ACL作为其访问ACL。

b）检索POSIX ACL

您可以查看文件或目录的现有POSIX ACL。

7）查看现有POSIX ACL

使用以下命令查看文件的现有访问ACL：

# getfacl

例如，要查看sample.jpg的现有POSIX ACL

# getfacl /mnt/gluster/data/test/sample.jpg

# owner: antony

# group: antony

user::rw-

group::rw-

other::r--

使用以下命令查看目录的默认ACL：

# getfacl

例如，要查看/ data / doc的现有ACL

# getfacl /mnt/gluster/data/doc

# owner: antony

# group: antony

user::rw-

user:john:r--

group::r--

mask::r--

other::r--

default:user::rwx

default:user:antony:rwx

default:group::r-x

default:mask::rwx

default:other::r-x

8）删除POSIX ACL

要删除用户，组或其他人的所有权限，请使用以下命令：

# setfacl -x

setfaclentry\_type选项

ACL entry\_type转换为所有者，组和其他的POSIX ACL表示。

权限必须是字符r（读取），w（写入）和x（执行）的组合。如下所述指定ACL entry\_type，用逗号分隔多个条目类型。

u：user\_name 设置用户的访问ACL。指定用户名或UID。

g：group\_name 设置组的访问ACL。指定组名称或GID。

m：permission 设置有效权限掩码。掩码是拥有组的所有访问权限以及所有用户和组条目的组合。

o：permissions 为文件组中的用户以外的用户设置访问ACL。

例如，要从用户antony中删除所有权限：

# setfacl -x u:antony /mnt/gluster/data/test-file

9）Samba和ACL

如果您使用Samba访问GlusterFS FUSE安装，则默认情况下会启用POSIX ACL。Samba已使用该--with-acl-support选项进行编译 ，因此在访问或装载Samba共享时不需要特殊标志。

10）NFS和ACL

目前GlusterFS通过NFS挂载支持POSIX ACL配置，即setfacl和getfacl命令通过NFS挂载工作。

## 4.3[设置客户端](https://docs.gluster.org/en/latest/Administrator%20Guide/Setting%20Up%20Clients/)

* 1. [处理属于多个组的用户](https://docs.gluster.org/en/latest/Administrator%20Guide/Handling-of-users-with-many-groups/)

## 4.4卷

### 4.4.1 调整卷选项

您可以根据需要在群集联机且可用时调整卷选项。

注意：如果每个卷中有太多块，或者有太多服务已经使用了系统中的所有特权端口，建议您将server.allow-insecure选项设置为ON。打开此选项允许端口接受/拒绝来自不安全端口的消息。因此，仅在部署需要时才使用此选项。

使用以下命令调整卷选项：

# gluster volume set <VOLNAME>

例如，要为test-volume指定性能缓存大小：

# gluster volume set test-volume performance.cache-size 256MB

Set volume successful

### 4.4.2 配置卷的传输类型

卷可以支持一种或多种传输类型，以便在客户端和块进程之间进行通信。支持的传输有三种类型，分别是tcp，rdma和tcp，rdma。

要更改卷的支持传输类型，请按照以下过程操作：

1）使用以下命令卸载所有客户端上的卷：

# umount mount-point

2）使用以下命令停止卷：

# gluster volume stop <VOLNAME>

3）更改运输类型。例如，要启用tcp和rdma，请执行followimg命令：

# gluster volume set test-volume config.transport tcp,rdma OR tcp OR rdma

4）在所有客户端上安装卷。例如，要使用rdma transport进行安装，请使用以下命令：

# mount -t glusterfs -o transport=rdma server1:/test-volume /mnt/glusterfs

### 4.4.3 扩展卷

您可以根据需要在群集联机且可用时扩展卷。例如，您可能希望向分布式卷添加块，从而增加分布并增加GlusterFS卷的容量。

同样，您可能希望将一组块添加到分布式复制卷，从而增加GlusterFS卷的容量。

注意：扩展分布式复制和分布式分散卷时，需要添加多个块，这些块是副本或分散计数的倍数。例如，要展开副本计数为2的分布式复制卷，您需要以2的倍数（例如4,6,8等）添加砖块。

扩展一个卷

1、如果它们还不是TSP的一部分，请使用以下命令探测包含要添加到卷的块的服务器：

# gluster peer probe <SERVERNAME>

例如：

# gluster peer probe server4

Probe successful

2、使用以下命令添加块：

# gluster volume add-brick <VOLNAME> <NEW-BRICK>

例如：

# gluster volume add-brick test-volume server4:/exp4

Add Brick successful

3、使用以下命令检查卷信息：

# gluster volume info <VOLNAME>

该命令显示类似于以下内容的信息：

Volume Name: test-volume

Type: Distribute

Status: Started

Number of Bricks: 4

Bricks:

Brick1: server1:/exp1

Brick2: server2:/exp2

Brick3: server3:/exp3

Brick4: server4:/exp4

4、重新平衡卷以确保将文件分发到新块。

您可以使用重新平衡卷中所述的rebalance命令

收缩量

您可以根据需要在群集联机且可用时缩小卷。例如，您可能需要删除由于硬件或网络故障而在分布式卷中无法访问的块。

注意

驻留在您要移除的砖块上的数据将无法再在Gluster安装点访问。但请注意，仅删除配置信息 - 您可以根据需要继续直接从块中访问数据。

缩小分布式复制和分布式分散卷时，需要删除多个副本或条带计数的块。例如，要缩小副本计数为2的分布式复制卷，您需要删除2的倍数的砖（例如4,6,8等）。此外，您要删除的砖块必须来自相同的子卷（相同的副本或分散集）。

使用start选项运行remove-brick 将自动触发重新平衡操作，以将数据从已移除的块迁移到卷的其余部分。

### 4.4.4 缩小卷

您可以根据需要在群集联机且可用时缩小卷。例如，您可能需要删除由于硬件或网络故障而在分布式卷中无法访问的块。

注意：驻留在您要移除的砖块上的数据将无法再在Gluster安装点访问。但请注意，仅删除配置信息 - 您可以根据需要继续直接从块中访问数据。

缩小分布式复制和分布式分散卷时，需要删除多个副本或条带计数的块。例如，要缩小副本计数为2的分布式复制卷，您需要删除2的倍数的砖（例如4,6,8等）。此外，您要删除的砖块必须来自相同的子卷（相同的副本或分散集）。

使用start选项运行remove-brick 将自动触发重新平衡操作，以将数据从已移除的块迁移到卷的其余部分。

1、使用以下命令删除块：

# gluster volume remove-brick <VOLNAME> <BRICKNAME> start

例如，要删除server2：/ exp2：

# gluster volume remove-brick test-volume server2:/exp2 start

volume remove-brick start: success

2、使用以下命令查看remove brick操作的状态：

# gluster volume remove-brick <VOLNAME> <BRICKNAME> status

例如，要查看server2上的remove brick操作的状态：/ exp2 brick：

# gluster volume remove-brick test-volume server2:/exp2 status

Node Rebalanced-files size scanned status

--------- ---------------- ---- ------- -----------

617c923e-6450-4065-8e33-865e28d9428f 34 340 162 in progress

3、状态显示“已完成”后，提交remove-brick操作

# gluster volume remove-brick <VOLNAME> <BRICKNAME> commit

在这个例子中：

# gluster volume remove-brick test-volume server2:/exp2 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

4、使用以下命令检查卷信息：

# gluster volume info

该命令显示类似于以下内容的信息：

# gluster volume info

Volume Name: test-volume

Type: Distribute

Status: Started

Number of Bricks: 3

Bricks:

Brick1: server1:/exp1

Brick3: server3:/exp3

Brick4: server4:/exp4

### 4.4.5 更换有缺陷的砖块

在纯分布卷中替换块

要替换仅分发卷上的块，请添加新块，然后删除要替换的块。这将触发重新平衡操作，该操作将从移除的砖块移动数据。

注意：仅对分布式复制卷或纯复制卷支持在gluster中使用'replace-brick'命令替换块。

删除brick Server1：/ home / gfs / r2\_1并添加Server1：/ home / gfs / r2\_2的步骤：

1、这是初始卷配置：

Volume Name: r2

Type: Distribute

Volume ID: 25b4e313-7b36-445d-b524-c3daebb91188

Status: Started

Number of Bricks: 2

Transport-type: tcp

Bricks:

Brick1: Server1:/home/gfs/r2\_0

Brick2: Server1:/home/gfs/r2\_1

2、以下是mount上存在的文件：

# ls

1. 10 2 3 4 5 6 7 8 9

3、现在添加新砖 - Server1：/ home / gfs / r2\_2：

# gluster volume add-brick r2 Server1:/home/gfs/r2\_2

volume add-brick: success

4、使用以下命令启动remove-brick：

# gluster volume remove-brick r2 Server1:/home/gfs/r2\_1 start

volume remove-brick start: success

ID: fba0a488-21a4-42b7-8a41-b27ebaa8e5f4

5、等到remove-brick状态表明它已完成。

# gluster volume remove-brick r2 Server1:/home/gfs/r2\_1 status

Node Rebalanced-files size scanned failures skipped status run time in secs

--------- ----------- ----------- ----------- ----------- ----------- ------------ --------------

localhost 5 20Bytes 15 0 0 completed 0.00

6、现在我们可以安全地删除旧砖，因此提交更改：

# gluster volume remove-brick r2 Server1:/home/gfs/r2\_1 commit

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

volume remove-brick commit: success

7、这是新的卷配置。

Volume Name: r2

Type: Distribute

Volume ID: 25b4e313-7b36-445d-b524-c3daebb91188

Status: Started

Number of Bricks: 2

Transport-type: tcp

Bricks:

Brick1: Server1:/home/gfs/r2\_0

Brick2: Server1:/home/gfs/r2\_2

8、检查mount的内容：

# ls

1. 10 2 3 4 5 6 7 8 9

替换复制/分布式复制卷中的块

本文档的这一部分描述了brick：如何Server1:/home/gfs/r2\_0用砖代替：具有副本计数的Server1:/home/gfs/r2\_5卷。r22

Volume Name: r2

Type: Distributed-Replicate

Volume ID: 24a0437a-daa0-4044-8acf-7aa82efd76fd

Status: Started

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: Server1:/home/gfs/r2\_0

Brick2: Server2:/home/gfs/r2\_1

Brick3: Server1:/home/gfs/r2\_2

Brick4: Server2:/home/gfs/r2\_3

步骤：

1、确保新砖Server1中没有数据：/ home / gfs / r2\_5

2、检查所有砖块是否正在运行。如果要更换的砖块已经关闭，那也没关系。

3、如果还没有，那么将要更换的砖块。

a）通过执行'gluster volume'获取砖块的pid 状态'

# gluster volume status

Status of volume: r2

Gluster process Port Online Pid

------------------------------------------------------------------------------

Brick Server1:/home/gfs/r2\_0 49152 Y 5342

Brick Server2:/home/gfs/r2\_1 49153 Y 5354

Brick Server1:/home/gfs/r2\_2 49154 Y 5365

Brick Server2:/home/gfs/r2\_3 49155 Y 5376

b）登录到运行砖块的机器并杀死砖块。

# kill -15 5342

c）确认砖块不再运行，其他砖块运行正常。

# gluster volume status

Status of volume: r2

Gluster process Port Online Pid

------------------------------------------------------------------------------

Brick Server1:/home/gfs/r2\_0 N/A N 5342 <<---- brick is not running, others are running fine.

Brick Server2:/home/gfs/r2\_1 49153 Y 5354

Brick Server1:/home/gfs/r2\_2 49154 Y 5365

Brick Server2:/home/gfs/r2\_3 49155 Y 5376

4、使用该Gluster volume fuse mount（在这个例子中：/mnt/r2）设置元数据，以便数据将被同步到新块（在这种情况下，它是从Server1:/home/gfs/r2\_1至Server1:/home/gfs/r2\_5）

在挂载点上创建一个尚不存在的目录。然后删除该目录，通过执行setfattr对元数据更改日志执行相同操作。此操作标志着未决的changelog它会告诉自愈达蒙/坐骑执行从自我修复/home/gfs/r2\_1来/home/gfs/r2\_5。

mkdir /mnt/r2/<name-of-nonexistent-dir>

rmdir /mnt/r2/<name-of-nonexistent-dir>

setfattr -n trusted.non-existent-key -v abc /mnt/r2

setfattr -x trusted.non-existent-key /mnt/r2

检查正在替换的砖的副本上是否有待处理的xattrs：

getfattr -d -m. -e hex /home/gfs/r2\_1

# file: home/gfs/r2\_1

security.selinux=0x756e636f6e66696e65645f753a6f626a6563745f723a66696c655f743a733000

trusted.afr.r2-client-0=0x000000000000000300000002 <<---- xattrs are marked from source brick Server2:/home/gfs/r2\_1

trusted.afr.r2-client-1=0x000000000000000000000000

trusted.gfid=0x00000000000000000000000000000001

trusted.glusterfs.dht=0x0000000100000000000000007ffffffe

trusted.glusterfs.volume-id=0xde822e25ebd049ea83bfaa3c4be2b440

5、卷修复信息将显示'/'。（可能有更多的条目基于工作负荷。但'/'必须存在）

# gluster volume heal r2 info

Brick Server1:/home/gfs/r2\_0

Status: Transport endpoint is not connected

Brick Server2:/home/gfs/r2\_1

/

Number of entries: 1

Brick Server1:/home/gfs/r2\_2

Number of entries: 0

Brick Server2:/home/gfs/r2\_3

Number of entries: 0

6、用'commit force'选项替换砖。请注意，不支持replace-brick命令的其他变体。

执行replace-brick命令

# gluster volume replace-brick r2 Server1:/home/gfs/r2\_0 Server1:/home/gfs/r2\_5 commit force

volume replace-brick: success: replace-brick commit successful

检查新块现在是否在线

# gluster volume status

Status of volume: r2

Gluster process Port Online Pid

------------------------------------------------------------------------------

Brick Server1:/home/gfs/r2\_5 49156 Y 5731 <<<---- new brick is online

Brick Server2:/home/gfs/r2\_1 49153 Y 5354

Brick Server1:/home/gfs/r2\_2 49154 Y 5365

Brick Server2:/home/gfs/r2\_3 49155 Y 5376

用户可以使用以下方式跟踪修复的进度：gluster volume heal [volname] info。一旦自我修复完成，将删除更改日志。

# getfattr -d -m. -e hex /home/gfs/r2\_1

getfattr: Removing leading '/' from absolute path names

# file: home/gfs/r2\_1

security.selinux=0x756e636f6e66696e65645f753a6f626a6563745f723a66696c655f743a733000

trusted.afr.r2-client-0=0x000000000000000000000000 <<---- Pending changelogs are cleared.

trusted.afr.r2-client-1=0x000000000000000000000000

trusted.gfid=0x00000000000000000000000000000001

trusted.glusterfs.dht=0x0000000100000000000000007ffffffe

trusted.glusterfs.volume-id=0xde822e25ebd049ea83bfaa3c4be2b440

# gluster volume heal <VOLNAME> info 将表明不需要修复。

# gluster volume heal r2 info

Brick Server1:/home/gfs/r2\_5

Number of entries: 0

Brick Server2:/home/gfs/r2\_1

Number of entries: 0

Brick Server1:/home/gfs/r2\_2

Number of entries: 0

Brick Server2:/home/gfs/r2\_3

Number of entries: 0

### 4.4.6 重新平衡卷

使用add-brick命令扩展卷后，您可能需要在服务器之间重新平衡数据。扩展或缩小卷后创建的新目录将自动均匀分布。对于所有现有目录，可以通过重新平衡布局和/或数据来修复分发。

本节介绍如何使用以下常见方案在存储环境中重新平衡GlusterFS卷：

修复布局 - 修复布局以使用新的卷拓扑，以便可以将文件分发到新添加的节点。

修复布局和迁移数据 - 通过修复布局以使用新卷拓扑并迁移现有数据来重新平衡卷。

**重新平衡卷以修复布局更改**

修复布局是必要的，因为布局结构对于给定目录是静态的。即使在将新砖添加到卷中之后，现有目录中新创建的文件仍将仅在原始块中分发。该命令gluster volume rebalance <volname> fix-layout start将修复布局信息，以便可以在新添加的砖块上创建文件。发出此命令后，将重新验证已缓存的所有文件统计信息。

从GlusterFS 3.6开始，将文件分配给砖块将考虑砖块的大小。例如，20TB砖的分配数量是10TB砖的两倍。在3.6之前的版本中，无论大小如何，两块砖被视为相等，并且将被分配相等份额的文件。

修复布局重新平衡仅修复布局更改，不会迁移数据。如果要迁移现有数据，请使用gluster volume rebalance start命令在服务器之间重新平衡数据。

重新平衡卷以修复布局

使用以下命令在任何Gluster服务器上启动重新平衡操作：

# gluster volume rebalance <VOLNAME> fix-layout start

例如：

# gluster volume rebalance test-volume fix-layout start

Starting rebalance on volume test-volume has been successful

**重新平衡卷以修复布局和迁移数据**

分别使用add-brick扩展卷后，需要在服务器之间重新平衡数据。remove-brick命令将自动触发重新平衡。

重新平衡卷以修复布局并迁移现有数据

1、使用以下命令在任一服务器上启动重新平衡操作：

# gluster volume rebalance <VOLNAME> start

例如：

# gluster volume rebalance test-volume start

Starting rebalancing on volume test-volume has been successful

2、使用以下命令在任何一台服务器上强制启动迁移操作：

# gluster volume rebalance <VOLNAME> start force

例如：

# gluster volume rebalance test-volume start force

Starting rebalancing on volume test-volume has been successful

**显示再平衡操作的状态**

您可以根据需要显示有关重新平衡卷操作的状态信息。

1、使用以下命令检查重新平衡操作的状态：

# gluster volume rebalance <VOLNAME> status

例如：

# gluster volume rebalance test-volume status

Node Rebalanced-files size scanned status

--------- ---------------- ---- ------- -----------

617c923e-6450-4065-8e33-865e28d9428f 416 1463 312 in progress

完成重新平衡操作的时间取决于卷上的文件数以及相应的文件大小。继续检查重新平衡状态，验证重新平衡的文件数或扫描的总文件数不断增加。

例如，再次运行status命令可能会显示类似于以下内容的结果：

# gluster volume rebalance test-volume status

Node Rebalanced-files size scanned status

--------- ---------------- ---- ------- -----------

617c923e-6450-4065-8e33-865e28d9428f 498 1783 378 in progress

重新平衡完成后，重新平衡状态显示以下内容：

# gluster volume rebalance test-volume status

Node Rebalanced-files size scanned status

--------- ---------------- ---- ------- -----------

617c923e-6450-4065-8e33-865e28d9428f 502 1873 334 completed

**停止正在进行的再平衡操作**

如果需要，您可以停止重新平衡操作。

1、使用以下命令停止重新平衡操作：

# gluster volume rebalance <VOLNAME> stop

例如：

# gluster volume rebalance test-volume stop

Node Rebalanced-files size scanned status

--------- ---------------- ---- ------- -----------

617c923e-6450-4065-8e33-865e28d9428f 59 590 244 stopped

Stopped rebalance process on volume test-volume

### 4.4.7 停止卷

使用以下命令停止卷：

# gluster volume stop <VOLNAME>

例如，要停止测试卷：

# gluster volume stop test-volume

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

输入y以确认操作。该命令的输出显示以下内容：

Stopping volume test-volume has been successful

### 4.4.8 删除卷

使用以下命令删除卷：

# gluster volume delete <VOLNAME>

例如，要删除测试卷：

# gluster volume delete test-volume

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

输入y以确认操作。该命令显示以下内容：

Deleting volume test-volume has been successful

### 4.4.9 在复制时触发自我修复

在复制模块中，以前您必须在砖块脱机并重新联机时手动触发自我修复，以使所有副本同步。现在，主动式自我修复守护程序在后台运行，诊断问题并自动启动每10分钟自我修复需要修复的文件。

您可以查看需要的文件列表愈合，这是目前/先前的文件列表中医治这是在脑裂状态，文件的列表，你可以手动触发整个卷上或仅在文件自我修复哪个需要治愈。

1、仅在需要治疗的文件上触发自我修复：

# gluster volume heal <VOLNAME>

例如，要触发需要修复 测试卷的文件的自我修复：

# gluster volume heal test-volume

Heal operation on volume test-volume has been successful

2、触发卷上所有文件的自我修复：

# gluster volume heal <VOLNAME> full

例如，要在test-volume的所有文件上触发自我修复：

# gluster volume heal test-volume full

Heal operation on volume test-volume has been successful

3、查看需要修复的文件列表：

# gluster volume heal <VOLNAME> info

例如，要查看需要修复的测试卷上的文件列表 ：

# gluster volume heal test-volume info

Brick server1:/gfs/test-volume\_0

Number of entries: 0

Brick server2:/gfs/test-volume\_1

Number of entries: 101

/95.txt

/32.txt

/66.txt

/35.txt

/18.txt

/26.txt

/47.txt

/55.txt

/85.txt

...

4、查看自我修复的文件列表：

# gluster volume heal <VOLNAME> info healed

例如，要查看自我修复的测试卷上的文件列表：

# gluster volume heal test-volume info healed

Brick Server1:/gfs/test-volume\_0

Number of entries: 0

Brick Server2:/gfs/test-volume\_1

Number of entries: 69

/99.txt

/93.txt

/76.txt

/11.txt

/27.txt

/64.txt

/80.txt

/19.txt

/41.txt

/29.txt

/37.txt

/46.txt

...

5、查看自愈失败的特定卷的文件列表：

# gluster volume heal <VOLNAME> info failed

例如，要查看未自我修复的测试卷文件列表：

# gluster volume heal test-volume info failed

Brick Server1:/gfs/test-volume\_0

Number of entries: 0

Brick Server2:/gfs/test-volume\_3

Number of entries: 72

/90.txt

/95.txt

/77.txt

/71.txt

/87.txt

/24.txt

...

6、查看处于裂脑状态的特定卷的文件列表：

# gluster volume heal <VOLNAME> info split-brain

例如，要查看处于裂脑状态的测试卷文件列表：

# gluster volume heal test-volume info split-brain

Brick Server1:/gfs/test-volume\_2

Number of entries: 12

/83.txt

/28.txt

/69.txt

...

Brick Server2:/gfs/test-volume\_3

Number of entries: 12

/83.txt

/28.txt

/69.txt

...

## [配置NFS-Ganesha](https://docs.gluster.org/en/latest/Administrator%20Guide/NFS-Ganesha%20GlusterFS%20Integration/)

## 4.6 特征

[地理复制](https://docs.gluster.org/en/latest/Administrator%20Guide/Geo%20Replication/)

[配额](https://docs.gluster.org/en/latest/Administrator%20Guide/Directory%20Quota/)

[快照](https://docs.gluster.org/en/latest/Administrator%20Guide/Managing%20Snapshots/)

[垃圾](https://docs.gluster.org/en/latest/Administrator%20Guide/Trash/)

## 4.7 使用其他接口进行数据访问

[管理对象库](https://docs.gluster.org/en/latest/Administrator%20Guide/Object%20Storage/)

[使用Cinder主机访问GlusterFS](https://docs.gluster.org/en/latest/Administrator%20Guide/GlusterFS%20Cinder/)

[带Keystone的GlusterFS](https://docs.gluster.org/en/latest/Administrator%20Guide/GlusterFS%20Keystone%20Quickstart/)

[在ZFS之上安装Gluster](https://docs.gluster.org/en/latest/Administrator%20Guide/Gluster%20On%20ZFS/)

[配置Bareos以在Gluster上存储备份](https://docs.gluster.org/en/latest/Administrator%20Guide/Bareos/)

## 4.8 [GlusterFS服务日志和位置](https://docs.gluster.org/en/latest/Administrator%20Guide/Logging/)

## 4.9 [监控工作量](https://docs.gluster.org/en/latest/Administrator%20Guide/Monitoring%20Workload/)

## 4.10 [使用SSL保护GlusterFS通信](https://docs.gluster.org/en/latest/Administrator%20Guide/SSL/)

## 4.11 [Puppet Gluster](https://docs.gluster.org/en/latest/Administrator%20Guide/Puppet/)

## 4.12 [RDMA运输](https://docs.gluster.org/en/latest/Administrator%20Guide/RDMA%20Transport/)

## 4.13 [GlusterFS iSCSI](https://docs.gluster.org/en/latest/Administrator%20Guide/GlusterFS%20iSCSI/)

## 4.14 [Linux内核调优](https://docs.gluster.org/en/latest/Administrator%20Guide/Linux%20Kernel%20Tuning/)

## 4.15[导出和网络组身份验证](https://docs.gluster.org/en/latest/Administrator%20Guide/Export%20And%20Netgroup%20Authentication/)

## 4.16 [瘦仲裁卷](https://docs.gluster.org/en/latest/Administrator%20Guide/Thin-Arbiter-Volumes/)

## 4.17 [GlusterFS的垃圾桶](https://docs.gluster.org/en/latest/Administrator%20Guide/Trash/)

## 4.18 [分裂大脑和处理它的方法](https://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/)

## 4.19 [仲裁卷和仲裁选项](https://docs.gluster.org/en/latest/Administrator%20Guide/arbiter-volumes-and-quorum/)

## 4.20 [强制锁定](https://docs.gluster.org/en/latest/Administrator%20Guide/Mandatory%20Locks/)

## 4.21 [GlusterFS coreutilities](https://docs.gluster.org/en/latest/Administrator%20Guide/GlusterFS%20Coreutils/)

## 4.22 [事件API](https://docs.gluster.org/en/latest/Administrator%20Guide/Events%20APIs/)

## 4.23 [用基于Debian的系统的gfapi构建QEMU](https://docs.gluster.org/en/latest/Administrator%20Guide/Building%20QEMU%20With%20gfapi%20For%20Debian%20Based%20Systems/)