##### brick管理

1、添加brick

gluster volume add-brick volume-name server1:/PATH server2:/PATH

2、移除brick

gluster volume remove-brick VOLNAME BRICK [start | status | commit]

3、替换brick

gluster volume replace-brick VOLNAME BRICKNEW-BRICK [start | pause | sbortstatus | commit]

2、手动同步数据

gluster volume volume-name reblance status

gluster volume volume-name reblance start

————————————————

#### 六、常见故障处理

##### 1. 从一个brick迁移到另外一个brick

gluster volume replace-brick volume source-brick new-brick commit force

new-brick未被其他gluster使用并且和源brick大小一致

eg: gluster volume replace-brick replica2 192.168.189.132:/gluster1/brick1 192.168.189.132:/gluster/brick1 commit force

迁移完成后，源brick就不会再存储新写入的数据

##### 2. 双复制一个节点的磁盘损坏

节点：gluster1、gluster2

gluster1节点的brick硬盘损坏

移除gluster1的brick

1）初始状态

[root@gluster2 ~]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

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

Brick gluster1:/data/brick1 49153 0 Y 3047

Brick gluster2:/data/brick1 49155 0 Y 1617

Self-heal Daemon on localhost N/A N/A Y 1634

Self-heal Daemon on gluster1 N/A N/A Y 3064

Task Status of Volume fuzhi

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

There are no active volume tasks

2）移除brick

[root@gluster2 ~]# gluster volume remove-brick fuzhi replica 1 gluster2:/data/brick1/ force

3）查看当前状态

[root@gluster2 ~]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

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

Brick gluster1:/data/brick1 49153 0 Y 3047

Task Status of Volume fuzhi

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

There are no active volume tasks

4）添加新的brick

[root@gluster2 ~]# gluster volume add-brick fuzhi replica 2 gluster2:/data1/brick1/

Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to avoid this. See: http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.

Do you still want to continue?

(y/n) y

volume add-brick: failed: /data1/brick1 is already part of a volume

brick添加失败，该brick已是卷的一部分，对于硬盘原有做过gluster，该硬盘的brick1下仍存在原有的brick信息，将原有信息删除

5）删除原有brick相关信息

[root@gluster2 ~]# rm -rf /data1/brick1/

6) 重新添加brick

[root@gluster2 data1]# gluster volume add-brick fuzhi replica 2 gluster2:/data1/brick1/

Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to avoid this. See: http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.

Do you still want to continue?

(y/n) y

volume add-brick: success

7）查看当前状态

[root@gluster2 data1]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

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

Brick gluster1:/data/brick1 49153 0 Y 3047

Brick gluster2:/data1/brick1 49155 0 Y 2063

Self-heal Daemon on localhost N/A N/A Y 2080

Self-heal Daemon on gluster1 N/A N/A Y 3493

Task Status of Volume fuzhi

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

There are no active volume tasks

##### 3. 将二复制转为三复制

当前状态

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

节点glusterfs-node2新增一块和已存在的两块brick大小一致的硬盘

Vmware热添加硬盘

echo '- - -' >/sys/class/scsi\_host/host0/scan

echo '- - -' >/sys/class/scsi\_host/host1/scan

echo '- - -' >/sys/class/scsi\_host/host2/scan

lsblk查看，多了一块sdc硬盘

[root@glusterfs-node2 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 20G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

├─centos-root 253:0 0 17G 0 lvm /

└─centos-swap 253:1 0 2G 0 lvm [SWAP]

sdb 8:16 0 5G 0 disk /gluster

sdc 8:32 0 5G 0 disk

sr0 11:0 1 973M 0 rom

进行格式化并挂载

mkfs.xfs /dev/sdc

mkdir /gluster1

mount /dev/sdc /gluster1

mkdir -p /gluster1/brick1

添加brick到原有的二复制卷，并由二复制更改为三复制

gluster v add-brick replica2 replica 3 192.168.189.132:/gluster1/brick1/

卷名 三复制

更改完以后的gluster状态

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Brick3: 192.168.189.132:/gluster1/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

##### 4. 将三复制转为二复制

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Brick3: 192.168.189.132:/gluster1/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

移除一个brick并从三复制更改为二复制

gluster v remove-brick replica2 replica 2 192.168.189.132:/gluster1/brick1 force

卷名 二复制

更改以后的状态

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

##### 5. 更改卷模式（从复制卷更改为分布式卷）

原有模式为二复制

[root@glusterfs-node2 brick1]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: 192.168.189.132:/gluster/brick1

Options Reconfigured:

performance.client-io-threads: off

nfs.disable: on

transport.address-family: inet

删除其中一个brick

gluster volume remove-brick replica2 replica 1 glusterfs-node2:/gluster/brick1 force

停止卷volume

gluster volume stop replica2

gluster volume delete replica2

再创建volume

gluster v create replica2 glusterfs-node1:/gluster/brick1/ glusterfs-node2:/gluster/brick1/ force

更改以后的模式为分布式

[root@glusterfs-node2 brick1]# gluster v info

Volume Name: replica2

Type: Distribute

Volume ID: 003cc5fc-d753-4666-8222-b10d12c865cf

Status: Created

Snapshot Count: 0

Number of Bricks: 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

启动volume

gluster v start replica2

更改为分布式后，所有

##### 8. 系统扩展维护

地域复制（geo-replication）：

# gluster volume geo-replication <MASTER> <SLAVE> start | status | stop

IO信息查看：

# gluster volume profile <VOLNAME> start | info | stop

Top监控：

Top命令允许你查看Brick的性能，例如：read,write, file open calls, file read calls, file write calls, directory opencalls, and directory real calls。所有的查看都可以设置 top数，默认100。

查看打开的 fd：

# gluster volume top <VOLNAME> open[brick <BRICK>] [list-cnt <COUNT>]

其中，open可以替换为read, write, opendir, readdir等。

查看每个 Brick 的读性能：

# gluster volume top <VOLNAME> read-perf [bs <BLOCK-SIZE> count <COUNT>] [brick <BRICK>] [list-cnt <COUNT>]

其中，read-perf可以替换为write-perf等。

#### 八、glusterfs脑裂处理

脑裂分为三种

1. 数据脑裂：文件中的数据在副本组的brick中不同
2. 元数据脑裂：brick中元数据不同
3. GFID脑裂：副本brick上的文件GFID不同，或者副本上的文件类型不同，文件类型不同无法修复，GFID可以修复，**GFID脑裂对外表现为目录脑裂**

列出所有需要修复的文件，glusterfs有自动修复脑裂的机制，列出的文件或目录不一定都需要手动修复

# gluster volume heal test info

Brick \<hostname:brickpath-b1>

<gfid:aaca219f-0e25-4576-8689-3bfd93ca70c2> - Is in split-brain

<gfid:39f301ae-4038-48c2-a889-7dac143e82dd> - Is in split-brain

<gfid:c3c94de2-232d-4083-b534-5da17fc476ac> - Is in split-brain

<gfid:6dc78b20-7eb6-49a3-8edb-087b90142246>

Number of entries: 4

Brick <hostname:brickpath-b2>

/dir/file2

/dir/file1 - Is in split-brain

/dir - Is in split-brain

/dir/file3

/file4 - Is in split-brain

/dir/a

Number of entries: 6

输出分析，

A） brickpath-b1 中，需要修复四个条目：

* 带有 GFID：6dc78b20-7EB6-49A3-8EDB-087B90142246 的文件需要愈合
* “AACA219F-0E25-4576-8689-3BFD93CA70C2”、“39F301AE-4038-48C2-A889-7DAC143E82DD”和“C3C94DE2-232D-4083-B534-5DA17FC476AC”属于裂脑

B） brickpath-b2中，需要修复六个条目：

* “a”、“file2”和“file3”需要修复
* “file1”、“file4”和“/dir”在裂脑中

因为/dir/a文件GFID脑裂，导致/dir处于脑裂状态

列出需要手动修复的文件和目录

# gluster volume heal test info split-brain

Brick <hostname:brickpath-b1>

<gfid:aaca219f-0e25-4576-8689-3bfd93ca70c2>

<gfid:39f301ae-4038-48c2-a889-7dac143e82dd>

<gfid:c3c94de2-232d-4083-b534-5da17fc476ac>

Number of entries in split-brain: 3

Brick <hostname:brickpath-b2>

/dir/file1

/dir

/file4

Number of entries in split-brain: 3

##### 1、使用gluster cli修复数据/元数据脑裂(server)

###### 选择较大的文件作为源

gluster volume heal <VOLNAME> split-brain bigger-file <FILE>

###### 选择最新的mtime作为源

gluster volume heal <VOLNAME> split-brain latest-mtime <FILE>

###### 选择副本中的一个brick作为特定文件的源

gluster volume heal <VOLNAME> split-brain source-brick <HOSTNAME:BRICKNAME> <FILE>

<FILE>可以以指定gfid的方式标识

gluster volume heal test split-brain source-brick test-host:/test/b1 gfid:c3c94de2-232d-4083-b534-5da17fc476ac

###### 选择副本中的一个brick作为所有文件的源

gluster volume heal <VOLNAME> split-brain source-brick <HOSTNAME:BRICKNAME>

##### 2、使用gluster cli修复GFID脑裂(server)

列出脑裂，该文件的父目录处于脑裂中，说明就是GFID脑裂

# gluster volume heal testvol info

Brick 10.70.47.45:/bricks/brick2/b0

/f5

/ - Is in split-brain

Status: Connected

Number of entries: 2

Brick 10.70.47.144:/bricks/brick2/b1

/f5

/ - Is in split-brain

Status: Connected

Number of entries: 2

对于GFID脑裂，它是父目录，显示处于脑裂中

使用如下命令获取文件是否处于GFID脑裂

# getfattr -d -e hex -m. <path-to-file>

在brick b0上

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

getfattr: Removing leading '/' from absolute path names

file: bricks/brick2/b0/f5

security.selinux=0x73797374656d5f753a6f626a6563745f723a676c7573746572645f627269636b5f743a733000

trusted.afr.testvol-client-1=0x000000020000000100000000

trusted.afr.dirty=0x000000000000000000000000

**trusted.gfid**=0xce0a9956928e40afb78e95f78defd64f

trusted.gfid2path.9cde09916eabc845=0x30303030303030302d303030302d303030302d303030302d3030303030303030303030312f6635

在brick b1上

# getfattr -d -m . -e hex /bricks/brick2/b1/f5

getfattr: Removing leading '/' from absolute path names

file: bricks/brick2/b1/f5

security.selinux=0x73797374656d5f753a6f626a6563745f723a676c7573746572645f627269636b5f743a733000

trusted.afr.testvol-client-0=0x000000020000000100000000

trusted.afr.dirty=0x000000000000000000000000

**trusted.gfid**=0x9563544118653550e888ab38c232e0c

trusted.gfid2path.9cde09916eabc845=0x30303030303030302d303030302d303030302d303030302d3030303030303030303030312f6635

可以看到两个brick中gfid不同

###### 选择较大的文件作为源

# gluster volume heal VOLNAME split-brain bigger-file FILE

先使用stat <file>查看文件大小

然后再选择较大的文件作为源修复文件的GFID脑裂

# gluster volume heal testvol split-brain bigger-file /f5

GFID split-brain resolved for file /f5

修复完成后，两个brick上的GFID必须与较大尺寸的GFID相同，修复完成后可以使用如下命令查看GFID使用相同

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

###### 选择最新的mtime作为源

# gluster volume heal VOLNAME split-brain latest-mtime FILE

修复完成后，两个brick上的GFID必须与较大尺寸的GFID相同，修复完成后可以使用如下命令查看GFID使用相同

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

###### 选择副本中的一个brick作为特定文件的源

# gluster volume heal VOLNAME split-brain source-brick HOSTNAME:export-directory-absolute-path FILE

修复完成后，两个brick上的GFID必须与较大尺寸的GFID相同，修复完成后可以使用如下命令查看GFID使用相同

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

###### 注意

* 不能将文件的 GFID 用作任何 CLI 选项的参数来解决 GFID 裂脑问题。它应该是被视为源的文件的绝对路径。
* 使用brick作为源选项，无法一次性解决所有 GFID 裂脑，因为在解析数据或元数据裂脑时，无需在 CLI 中指定任何文件路径。对于 GFID 裂脑中的每个文件，使用要使用的策略运行 CLI。
* 使用带有“分布式复制”卷中的“brick块”选项的 CLI 解析目录 GFID 裂脑需要在处于此状态的所有子卷上显式完成。由于目录会在所有brick上创建，因此使用一个特定的brick作为目录 GFID 裂脑的源可以修复该特定子卷的目录。源brick的选择方式应使修复后所有子卷的所有brick都具有相同的 GFID。
* 如前所述，无法使用CLI解决文件系统类型不匹配的问题

##### 3、从挂载点（client）修复脑裂

提供了一组 getfattr 和 setfattr 命令来检测文件的数据和元数据裂脑状态，并从挂载点解析裂脑（如果有）。

# gluster v heal test info split-brain

Brick test-host:/test/b0/

/file100

/dir

Number of entries in split-brain: 2

Brick test-host:/test/b1/

/file100

/dir

Number of entries in split-brain: 2

Brick test-host:/test/b2/

/file99

<gfid:5399a8d1-aee9-4653-bb7f-606df02b3696>

Number of entries in split-brain: 2

Brick test-host:/test/b3/

<gfid:05c4b283-af58-48ed-999e-4d706c7b97d5>

<gfid:5399a8d1-aee9-4653-bb7f-606df02b3696>

Number of entries in split-brain: 2

使用如下命令了解处于哪种脑裂状态

getfattr -n replica.split-brain-status <path-to-file>

eg:

处于元数据脑裂

# getfattr -n replica.split-brain-status file100

file: file100

replica.split-brain-status="data-split-brain:no metadata-split-brain:yes Choices:test-client-0,test-client-1"

处于数据脑裂

# getfattr -n replica.split-brain-status file99

file: file99

replica.split-brain-status="data-split-brain:yes metadata-split-brain:yes Choices:test-client-2,test-client-3"

元数据和数据都处于脑裂中

# getfattr -n replica.split-brain-status file99

file: file99

replica.split-brain-status="data-split-brain:yes metadata-split-brain:yes Choices:test-client-2,test-client-3"

dir不在数据或者元数据脑裂下，

# getfattr -n replica.split-brain-status dir

file: dir

replica.split-brain-status="The file is not under data or metadata split-brain"

###### 从client解决数据和元数据脑裂

尝试在挂载点上对脑裂的文件进行操作（例如 cat、getfattr 等），会出现输入/输出错误。为了使用户能够分析此类文件，提供了一个 setfattr 命令。

# setfattr -n replica.split-brain-choice -v "choiceX" <path-to-file>

使用此命令，可以选择一个特定的块来访问裂脑中的文件。

eg：

1、“file1”位于数据脑裂中。尝试从文件中读取会产生输入/输出错误。

# cat file1

cat: file1: Input/output error

为file1提供的服务端选择有test-cilent-1和test-client-2

将test-client-2设置为file1的脑裂选择源，将从test-client-2的brick中读取文件

# setfattr -n replica.split-brain-choice -v test-client-2 file1

然后再cat文件

cat file1

xyz123

要撤销已设置的脑裂选择，可以使用“none”作为setfattr的扩展属性的值

# setfattr -n replica.split-brain-choice -v none file1

再cat文件，就会和以前一样，输出Input/output error

# cat file

cat: file1: Input/output error

如果想解决脑裂问题，应该设置源brick

# setfattr -n replica.split-brain-heal-finalize -v <heal-choice> <path-to-file>

eg：

# setfattr -n replica.split-brain-heal-finalize -v test-client-2 file1

上述过程可以解决所有文件的数据/元数据脑裂

###### 注意：

1、如果禁用了“fopen-keep-cache”保险丝装载选项，则每次在选择新副本之前都需要使 inode 失效。拆分大脑选择检查文件。这可以通过使用来完成：

# sefattr -n inode-invalidate -v 0 <path-to-file>

2、上面从client修复脑裂的过程不适合nfs挂载，因为nfs不提供xattrs支持

##### 4、设置glustefs，让gluseterfs自动修复脑裂

基于CLI和client的方法需要人工干预，有一个卷设置，当设置为各种可用策略之一时，无需用户干预即可自动恢复脑裂，默认被禁用。

cluster.favorite-child-policy

查看help

[root@glusterfs-node2 brick]# gluster v set help | grep -A3 cluster.favorite-child-policy

Option: cluster.favorite-child-policy

Default Value: none

Description: This option can be used to automatically resolve split-brains using various policies without user intervention.

"size" picks the file with the biggest size as the source. "ctime" and "mtime" pick the file with the latest ctime and mtime respectively as the source. "majority" picks a file with identical mtime and size in more than half the number of bricks in the replica.a

设置glusterfs以最新mtime自动修复脑裂

[root@glusterfs-node2 ~]# gluster v set replica2 cluster.favorite-child-policy mtime

volume set: success

[root@dockernode1 /]# gluster v info

Options Reconfigured:

nfs.disable: on #关闭nfs挂载

storage.fips-mode-rchecksum: on

transport.address-family: inet

diagnostics.latency-measurement: on

diagnostics.count-fop-hits: on

##### 5、快速入门：

1. 获取裂脑中文件的路径：

它可以通过  
a） 命令获得。  
b） 确定从客户端执行的文件操作不断失败并出现输入/输出错误的文件。gluster volume heal info split-brain

1. 关闭从装入点打开此文件的应用程序。 对于虚拟机，需要关闭它们的电源。
2. 确定正确的副本：

这是通过观察文件的 afr 更改日志扩展属性来完成的 使用 getfattr 命令的砖块;然后确定裂脑的类型 （数据裂脑、元数据裂脑、条目裂脑或裂脑由于 GFID-不匹配）;最后确定哪个砖块包含“好副本” 的文件。  
.  
也可能一个砖可能包含正确的数据，而 其他可能包含正确的元数据。getfattr -d -m . -e hex <file-path-on-brick>

1. 重置包含 使用 setfattr 命令的文件数据/元数据的“错误副本”。

setfattr -n <attribute-name> -v <attribute-value> <file-path-on-brick>

1. 通过从客户端执行查找来触发对文件的自我修复：

ls -l <file-path-on-gluster-mount>

#### 九、深入探究glusterfs

|  |  |  |
| --- | --- | --- |
| glusterfs-node1 |  |  |
| glusterfs-node2 |  |  |
| glusterfs-client |  |  |

##### 双复制集群单一节点故障分析

双复制机器，当client和server端已建立连接后

任一节点或者双节点glusterd服务stop(glusterd被kill掉)不影响client端

任一节点的glusterfsd进程被kill掉，有一半几率会出问题

[root@glusterfs-node1 brick]# netstat -tunlp | grep "gluster"

tcp 0 0 0.0.0.0:49152 0.0.0.0:\* LISTEN 1101/glusterfsd

tcp 0 0 0.0.0.0:49153 0.0.0.0:\* LISTEN 4480/glusterfsd

tcp 0 0 0.0.0.0:49154 0.0.0.0:\* LISTEN 1116/glusterfsd

tcp 0 0 0.0.0.0:24007 0.0.0.0:\* LISTEN 4468/glusterd

[root@glusterfs-node2 ~]# netstat -tunlp | grep "gluster"

tcp 0 0 0.0.0.0:49152 0.0.0.0:\* LISTEN 1001/glusterfsd

tcp 0 0 0.0.0.0:49153 0.0.0.0:\* LISTEN 4228/glusterfsd

tcp 0 0 0.0.0.0:49154 0.0.0.0:\* LISTEN 1034/glusterfsd

tcp 2 0 0.0.0.0:24007 0.0.0.0:\* LISTEN 4216/glusterd

[root@ceph-node1 ~]# netstat -tunp | grep "gluster"

tcp 0 0 192.168.189.150:49147 192.168.189.131:24007 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49129 192.168.189.132:24007 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49151 192.168.189.131:24007 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49142 192.168.189.131:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49137 192.168.189.132:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49138 192.168.189.131:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49144 192.168.189.131:49153 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49143 192.168.189.132:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49130 192.168.189.132:49153 ESTABLISHED 1904/glusterfs

当kill掉glusterfs-node2的49153端口的进程时，查看glusterfs-client

[root@ceph-node1 ~]# !netstat

netstat -tunp

Active Internet connections (w/o servers)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 68 192.168.189.150:22 192.168.189.1:64938 ESTABLISHED 1195/sshd: root@pts

tcp 0 0 192.168.189.150:49147 192.168.189.131:24007 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49150 192.168.189.132:24007 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49151 192.168.189.131:24007 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49142 192.168.189.131:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49137 192.168.189.132:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49138 192.168.189.131:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49144 192.168.189.131:49153 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49143 192.168.189.132:49154 ESTABLISHED 1350/glusterfs

此时客户端还可以继续使用49153端口的replica2 volume

当kill掉glusterfs-node1的49153端口的进程时，查看glusterfs-client

[root@ceph-node1 ~]# netstat -tunp | grep "gluster"

tcp 0 0 192.168.189.150:49147 192.168.189.131:24007 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49129 192.168.189.132:24007 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49151 192.168.189.131:24007 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49142 192.168.189.131:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49137 192.168.189.132:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49138 192.168.189.131:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49143 192.168.189.132:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49130 192.168.189.132:49153 ESTABLISHED 1904/glusterfs

此时glusterfs-client访问replica2 volume的挂载目录

[root@ceph-node1 ~]# cd /replica

-bash: cd: /replica: Transport endpoint is not connected

当server端为双复制的时候，且cluster.quorum-type: auto时，一个节点down机，有一半的可能导致client无法使用，与挂载时写的哪个serverIP无关系

当glusterfs-node1节点恢复正常后，client会自动恢复正常，无需其他操作

##### 2、glusterfs是否支持多副本？

[root@glusterfs-node1 brick]# gluster v create test-replica4 replica 4 glusterfs-node1:/test-replica1/brick/ glusterfs-node1:/test-replica2/brick/ glusterfs-node1:/test-replica3/brick/ glusterfs-node1:/test-replica4/brick/ force

volume create: test-replica4: success: please start the volume to access data

[root@glusterfs-node1 brick]# gluster v status test-replica4

Volume test-replica4 is not started

启动test-replica4 volume

[root@glusterfs-node1 brick]# gluster v start test-replica4

volume start: test-replica4: success

[root@glusterfs-node1 brick]# gluster v status test-replica4

Status of volume: test-replica4

Gluster process TCP Port RDMA Port Online Pid

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

Brick glusterfs-node1:/test-replica1/brick 49155 0 Y 5202

Brick glusterfs-node1:/test-replica2/brick 49156 0 Y 5222

Brick glusterfs-node1:/test-replica3/brick 49157 0 Y 5242

Brick glusterfs-node1:/test-replica4/brick 49158 0 Y 5262

Self-heal Daemon on localhost N/A N/A Y 5283

Self-heal Daemon on glusterfs-node2 N/A N/A Y 4466

Task Status of Volume test-replica4

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

There are no active volume tasks

[root@glusterfs-node1 brick]# gluster v info test-replica4

Volume Name: test-replica4

Type: Replicate

Volume ID: dd9cc607-8ef9-4c6b-ba63-492fadadd084

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 4 = 4

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/test-replica1/brick

Brick2: glusterfs-node1:/test-replica2/brick

Brick3: glusterfs-node1:/test-replica3/brick

Brick4: glusterfs-node1:/test-replica4/brick

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

可以看到glusterfs支持多副本（3副本以上）