

3.4.2. glusterfs 的几个重要进程

这里想简单分享一下关于 glusterfs 中的几个重要的进程，这一小节的内容也与前面的关系不大，可能日常运维中有些大家也见过了，下面列举几个重要的常见的进程。

1) gluster : 是 cli 命令执行工具，主要功能是解析命令行参数，然后把命令发送给 glusterd 模块执行。

2) glusterd: 是一个管理模块，处理 gluster 发过来的命令，处理集群管理、存储池管理、brick 管理、负载均衡、快照管理等。集群信息、存储池信息和快照信息等都是以配置文件的形式存放在服务器中，当客户端挂载存储时，glusterd 会把存储池的配置文件发送给客户端。

3) glusterfsd : 是服务端模块，存储池中的每个 brick 都会启动一个 glusterfsd 进程。此模块主要是处理客户端的读写请求，从关联的 brick 所在磁盘中读写数据，然后返回给客户端。

4) glusterfs : 是客户端模块，负责通过 mount 挂载集群中某台服务器的存储池，以目录的形式呈现给用户。当用户从此目录读写数据时，客户端根据从 glusterd 模块获取的存储池的配置文件信息，通过 DHT 算法计算文件所在服务器的 brick 位置，然后通过 Infiniband RDMA 或 Tcp/Ip 方式把数据发送给 brick，等 brick 处理完，给用户返回结果。存储池的副本、条带、hash、EC 等逻辑都在客户端处理。

除了这几个核心最基本的进程以外，当然这里还有关于 heal 的 shd 进程，还有和 rebalance(重平衡，也就是当一个 volume 扩容，添加 brick 之后，数据需要重新平衡)相关的进程等等，关于这些进程，在后面提到不同功能的时候

再进行分享。下面先来再感受一下这几个核心进程的关系。

```
1. [root@gfs03 ~]# systemctl status glusterd
2. • glusterd.service - GlusterFS, a clustered file-system s
   erver
3.    Loaded: loaded (/usr/lib/systemd/system/glusterd.servi
   ce; enabled; vendor preset: disabled)
4.    Active: active (running) since Sun 2021-06-13 05:28:42
   EDT; 28min ago
5.    Docs: man:glusterd(8)
6.    Process: 897 ExecStart=/usr/sbin/glusterd -p /var/run/g
   lusterd.pid --log-level $LOG_LEVEL $GLUSTERD_OPTIONS (code
   =exited, status=0/SUCCESS)
7.    Main PID: 902 (glusterd)
8.    CGroup: /system.slice/glusterd.service
9.           └─ 902 /usr/sbin/glusterd -p /var/run/glusterd.
   pid --log-level INFO
10.            └─ 955 /usr/sbin/glusterfsd -s 192.168.0.112 -
   -volfile-id test-arbiter.192.168.0.112.glusterfs-test-arbi
   ter -p /var/run/gluster/vols/test-arbiter/192.168...
11.            └─ 974 /usr/sbin/glusterfsd -s 192.168.0.112 -
   -volfile-id test-disperse.192.168.0.112.glusterfs-test-dis
   perse -p /var/run/gluster/vols/test-disperse/192....
12.            └─ 999 /usr/sbin/glusterfsd -s 192.168.0.112 -
   -volfile-id test-replica.192.168.0.112.glusterfs-test-repl
   ica -p /var/run/gluster/vols/test-replica/192.168...
13.            └─ 1026 /usr/sbin/glusterfs -s localhost --volf
   ile-id shd/test-arbiter -p /var/run/gluster/shd/test-arbit
   er/test-arbiter-shd.pid -l /var/log/glusterfs/glu...
14.
15. Jun 13 05:28:42 gfs03 systemd[1]: Starting GlusterFS, a c
   lustered file-system server...
16. Jun 13 05:28:42 gfs03 systemd[1]: Started GlusterFS, a cl
   ustered file-system server.
```

从这里的进程关系中就比较好理解 glusterfsd 和 glusterd 的关系了 ,另外

这里还可以从/usr/sbin/中看出两者的关系。

```
1. [root@gfs01 ~]# ls -l /usr/sbin/gluster*
2. -rwxr-xr-x. 1 root root 453864 May 10 19:13 /usr/sbin/glu
   ster
3. lrwxrwxrwx. 1 root root      10 May 24 12:45 /usr/sbin/glu
   sterd -> glusterfsd
```

```

4.  lrwxrwxrwx. 1 root root      10 May 24 12:45 /usr/sbin/glu
sterfs -> glusterfsd
5.  -rwxr-xr-x. 1 root root 251384 May 10 19:13 /usr/sbin/glu
sterfsd
6.  -rwxr-xr-x. 1 root root  11232 May 10 19:13 /usr/sbin/glu
ster-setgid2path

```

从这里可以知道，其实 glusterfs 和 glusterd 都是来自 glusterfsd 的。当然这里就可以使用这些命令来手动启动一些进程了，下面进行一下小实验，当然在做这个实验之前，先看看测试的 volume 情况。

```

1.  [root@gfs02 ~]# gluster volume info test-replica
2.
3.  Volume Name: test-replica
4.  Type: Replicate
5.  Volume ID: 4568c063-5b75-4304-98f6-21f3955cc138
6.  Status: Started
7.  Snapshot Count: 0
8.  Number of Bricks: 1 x 3 = 3
9.  Transport-type: tcp
10. Bricks:
11. Brick1: 192.168.0.110:/glusterfs/test-replica
12. Brick2: 192.168.0.111:/glusterfs/test-replica
13. Brick3: 192.168.0.112:/glusterfs/test-replica
14. Options Reconfigured:
15. diagnostics.client-log-level: DEBUG
16. diagnostics.brick-log-level: DEBUG
17. cluster.granular-entry-heal: on
18. storage.fips-mode-rchecksum: on
19. transport.address-family: inet
20. nfs.disable: on
21. performance.client-io-threads: off

```

这里是一个很简单的复制卷，同时有三个 brick，下面进行测试。

```

1.  [root@gfs01 ~]# ps aux |grep glusterfs
2.  root      1402  0.0  0.3 875904 14328 ?        Ssl  05:28
    0:00 /usr/sbin/glusterfsd -s 192.168.0.110 --volfile-id
    test-arbiter.192.168.0.110.glusterfs-test-arbiter -p /var
    /run/gluster/vols/test-arbiter/192.168.0.110-glusterfs-tes
    t-arbiter.pid -S /var/run/gluster/fdd86b6311f78aa9.socket
    --brick-name /glusterfs/test-arbiter -l /var/log/glusterfs

```

```

/bricks/glusterfs-test-arbiter.log --xlator-option *-posix.
glusterd-uuid=99827066-72c7-484f-b1e8-0a9c4bc46b54 --proce
ss-name brick --brick-port 49152 --xlator-option test-arbi
ter-server.listen-port=49152
3. root      1413  0.0  0.3 875904 15324 ?          Ssl  05:28
   0:00 /usr/sbin/glusterfsd -s 192.168.0.110 --volfile-id
   test-disperse.192.168.0.110.glusterfs-test-disperse -p /v
   ar/run/gluster/vols/test-disperse/192.168.0.110-glusterfs-
   test-disperse.pid -S /var/run/gluster/fa9f90c834b1bc98.soc
   ket --brick-name /glusterfs/test-disperse -l /var/log/glus
   terfs/bricks/glusterfs-test-disperse.log --xlator-option *
   -posix.glusterd-uuid=99827066-72c7-484f-b1e8-0a9c4bc46b54
   --process-name brick --brick-port 49153 --xlator-option te
   st-disperse-server.listen-port=49153
4. ....
5.
6.
7. [root@gfs01 ~]# ps aux |grep glusterd
8. root      904  0.0  0.2 524836  9452 ?          Ssl  05:28
   0:00 /usr/sbin/glusterd -p /var/run/glusterd.pid --log-
   level INFO
9. root      1402  0.0  0.3 875904 14328 ?          Ssl  05:28
   0:00 /usr/sbin/glusterfsd -s 192.168.0.110 --volfile-id
   test-arbiter.192.168.0.110.glusterfs-test-arbiter -p /var
   /run/gluster/vols/test-arbiter/192.168.0.110-glusterfs-tes
   t-arbiter.pid -S /var/run/gluster/fdd86b6311f78aa9.socket
   --brick-name /glusterfs/test-arbiter -l /var/log/glusterfs
   /bricks/glusterfs-test-arbiter.log --xlator-option *-posix.
   glusterd-uuid=99827066-72c7-484f-b1e8-0a9c4bc46b54 --proce
   ss-name brick --brick-port 49152 --xlator-option test-arbi
   ter-server.listen-port=49152
10. root      1413  0.0  0.3 875904 15324 ?          Ssl  05:28
   0:00 /usr/sbin/glusterfsd -s 192.168.0.110 --volfile-id
   test-disperse.192.168.0.110.glusterfs-test-disperse -p /v
   ar/run/gluster/vols/test-disperse/192.168.0.110-glusterfs-
   test-disperse.pid -S /var/run/gluster/fa9f90c834b1bc98.soc
   ket --brick-name /glusterfs/test-disperse -l /var/log/glus
   terfs/bricks/glusterfs-test-disperse.log --xlator-option *
   -posix.glusterd-uuid=99827066-72c7-484f-b1e8-0a9c4bc46b54
   --process-name brick --brick-port 49153 --xlator-option te
   st-disperse-server.listen-port=49153
11. ....
12.
13.

```

14. //下面删掉所有相关的进程

```
15. [root@gfs01 ~]# killall glusterd glusterfs glusterfsd
```

```
16. [root@gfs01 ~]# ps aux |grep glusterd
```

```
17. root      1964  0.0  0.0 112812   976 pts/0    R+   06:30
    0:00 grep --color=auto glusterd
```

```
18. [root@gfs01 ~]# ps aux |grep glusterfs
```

```
19. root      1966  0.0  0.0 112812   976 pts/0    R+   06:30
    0:00 grep --color=auto glusterfs
```

这里首先使用 killall 干掉所有相关的进程，主要就是 glusterd 和 glusterfsd。然后下面启动 glusterfs 进程，如下所示。

```
1. [root@gfs01 ~]# glusterfs -f /var/lib/glusterd/vols/test-
   replica/trusted-test-replica.tcp-fuse.vol /mnt/test-replic
   a
```

```
2. [root@gfs01 ~]# df -h /mnt/test-replica
```

```
3. Filesystem
```

```
Size Used Avail Use% Mounted on
```

```
4. /var/lib/glusterd/vols/test-replica/trusted-test-replica.
   tcp-fuse.vol 50G 8.7G 42G 18% /mnt/test-replica
```

```
5. [root@gfs01 ~]# ps aux |grep glusterfs
```

```
6. root      1984  0.3  0.2 629228 8304 ?        Ssl  06:48
    0:02 glusterfs -f /var/lib/glusterd/vols/test-replica/t
   rusted-test-replica.tcp-fuse.vol /mnt/test-replica
```

```
7.
```

8. //这里在其他节点上查看 volume status

```
9. [root@gfs02 ~]# gluster volume status test-replica
```

```
10. Status of volume: test-replica
```

```
11. Gluster process                                TCP Port  RDM
    A Port  Online  Pid
```

```
12. -----
    -----
```

```
13. Brick 192.168.0.111:/glusterfs/test-replica 49154    0
    Y      1423
```

```
14. Brick 192.168.0.112:/glusterfs/test-replica 49154    0
    Y      999
```

```
15. Self-heal Daemon on localhost                N/A        N/A
    Y      1436
```

```
16. Self-heal Daemon on gfs03                     N/A        N/A
    Y      1026
```

```

17.
18. Task Status of Volume test-replica
19. -----
    -----
20. There are no active volume tasks

```

这里可以看到，虽然挂载的进程启动了，但是 brick 相关进程是没有的，同时在其他节点上，使用 volume status 也是看不到相关信息的。另外为了测试到底这个挂载进行是否生效的，可以在挂载目录下创建一个文件，然后在其他节点下的 brick 目录查看这个文件内容，这里的测试结果如下所示。

```

1. [root@gfs01 ~]# touch /mnt/test-replica/210613.txt
2. [root@gfs01 ~]# date >> /mnt/test-replica/210613.txt
3.
4. [root@gfs02 ~]# cat /glusterfs/test-replica/210613.txt
5. Sun Jun 13 07:08:14 EDT 2021

```

另外这里如果想手动启动 glusterfsd 进程，可以根据 brick 中的日志内容，找到之前启动的一些类似的命令进行重启，但是往往这里还要先判断端口是否存在占用的情况。这里的日志类似如下所示。

```

1. [2021-05-25 16:11:42.442930 +0000] I [MSGID: 100030] [glusterfsd.c:2687:main] 0-/usr/sbin/glusterfsd: Started running version [{arg=/usr/sbin/glusterfsd}, {version=9.2}, {cmdlinestr=/usr/sbin/glusterfsd -s 192.168.0.110 --volfile-id test-replica.192.168.0.110.glusterfs-test-replica -p /var/run/gluster/vols/test-replica/192.168.0.110-glusterfs-test-replica.pid -S /var/run/gluster/d403ef3161bd809c.socket --brick-name /glusterfs/test-replica -l /var/log/glusterfs/bricks/glusterfs-test-replica.log --xlator-option *-posix.glusterd-uuid=99827066-72c7-484f-b1e8-0a9c4bc46b54 --process-name brick --brick-port 49152 --xlator-option test-replica-server.listen-port=49152}]

```