

## 4.4. 简洁的客户端 coreutils

在 glusterfs 中，这里提供了一个叫 gfccli 的工具，简单来理解，这可以理解为一个客户端工具，只是能提供一些非常简单的小功能，例如 cat, cp, flock, ls, mkdir, rm 和 stat 等功能，而要使用这个功能，还需要先提前安装 glusterfs-coreutils 包，下面来简单演示一下。

```
1. [root@gfs03 ~]# gfc
2. gfc cat    gfc clear  gfc cli    gfc cp
3.
4. [root@gfs03 ~]# gfc cli
5. //连接 volume
6. gfc cli> connect glfs://localhost/test-afr
7. gfc cli (localhost/test-afr)> ls
8. test-dir.sh test.sh c.txt
9. gfc cli (localhost/test-afr)> rm -fr c.txt
10. rm: failed to remove `c.txt': Not a directory
11. //删除文件
12. gfc cli (localhost/test-afr)> rm c.txt
13. gfc cli (localhost/test-afr)> ls
14. test-dir.sh test.sh
15. //使用 cat 命令查看文件
16. gfc cli (localhost/test-afr)> cat test.sh
17. #!/bin/bash
18.
19.
20. hostNameStr=`sudo hostname -i`
21.
22. for i in $(seq 1 10000)
23. do
24.     dateStr=`sudo date`
25.     sudo echo -e $dateStr > 1.txt
26.     sudo echo -e $dateStr > /tmp/$hostNameStr
27. done
28.
29. //这里使用 stat 命令
```

```

30. gfccli (localhost/test-afr)> stat test.sh
31.   File: `test.sh'
32.   Size: 181          Blocks: 1          IO Block: 4096   r
    egular file
33. Device: b8273e59h/3089579609d   Inode: 116561552966268014
    72 Links: 1
34. Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (
    0/   root)
35. Access: 2021-06-17 14:02:39.782872499 +0800
36. Modify: 2021-06-17 14:02:39.784426977 +0800
37. Change: 2021-06-17 14:02:39.791984784 +0800
38. gfccli (localhost/test-afr)> quit

```

## 4.5. 强大的 webhook event

在 glusterfs 当中,当创建一个 volume 或者做一些操作的时候,想需要知道这些时间 event 的话,glusterfs 官方提供了一个叫 glustereventsds 的服务,需要先启动,下面进行简单的演示。

```

1. root@ubuntu20:~# gluster-eventsapi status
2. Webhooks:
3. http://10.0.12.9:9000/listen
4.
5. +-----+-----+-----+
6. |  NODE   | NODE STATUS | GLUSTEREVENTSD STATUS |
7. +-----+-----+-----+
8. | localhost |          UP |                OK |
9. +-----+-----+-----+

```

这里可以先看看是否有注册的 webhook URL,这里因为之前已经调用了因此这里会显示,当然这里也可以进行删除。

```

1. root@ubuntu20:~# gluster-eventsapi webhook-del http://10.
    0.12.9:9000/listen
2. +-----+-----+-----+
3. |  NODE   | NODE STATUS | SYNC STATUS |
4. +-----+-----+-----+
5. | localhost |          UP |                OK |
6. +-----+-----+-----+

```

```

7.
8. root@ubuntu20:~# gluster-eventsapi status
9. Webhooks: None
10.
11. +-----+-----+-----+
12. |   NODE   | NODE STATUS | GLUSTEREVENTSD STATUS |
13. +-----+-----+-----+
14. | localhost |           UP |                   OK |
15. +-----+-----+-----+

```

这里取消掉了该 webhook url 了，下面重新弄一下，并且启动一个测试代码进行监听测试。

```

1. root@ubuntu20:~# gluster-eventsapi webhook-add http://10.
  0.12.9:9000/listen
2. ...
3.
4. root@ubuntu20:~# cat webhook.py
5. from flask import Flask, request
6.
7. app = Flask(__name__)
8.
9. @app.route("/listen", methods=["POST"])
10. def events_listener():
11.     gluster_event = request.json
12.     if gluster_event is None:
13.         # No event to process, may be test call
14.         return "OK"
15.
16.     # Process gluster_event
17.     # {
18.     #   "nodeid": NODEID,
19.     #   "ts": EVENT_TIMESTAMP,
20.     #   "event": EVENT_TYPE,
21.     #   "message": EVENT_DATA
22.     # }
23.     print (gluster_event)
24.     return "OK"
25.
26. app.run(host="0.0.0.0", port=9000)

```

那么这里有一段 python 代码,这里就是使用了 flask 框架 ,这里的作用就是

监听了该端口，然后使用 python 进行执行，注意这里需要先安装 python-pip，然后使用 pip 进行安装 flask 包，然后执行。

```
1. root@ubuntu20:~# gluster volume create test-event 10.0.12.9:/glusterfs/test-event force
2. volume create: test-event: success: please start the volume to access data
3.
4. ....
```

那么当程序运行的时候，然后创建了一个 volume 的时候，在程序中可以看到下面的信息输出，这里的输出是有规范的，同时这里有很多 events，其中包括 snapshot,quota,brick 和 volume 等 events 的监听，具体的可以查看官方文档<sup>[1]</sup>。

```
1. root@ubuntu20:~# python webhook.py
2. * Serving Flask app 'webhook' (lazy loading)
3. * Environment: production
4.   WARNING: This is a development server. Do not use it in a production deployment.
5.   Use a production WSGI server instead.
6. * Debug mode: off
7. * Running on all addresses.
8.   WARNING: This is a development server. Do not use it in a production deployment.
9. * Running on http://10.0.12.9:9000/ (Press CTRL+C to quit)
10.
11. {'event': 'VOLUME_CREATE', 'message': {'bricks': ' 10.0.12.9:/glusterfs/test-event', 'name': 'test-event'}, 'nodeid': '3ed7d443-1312-4098-bd50-2a9c98ada9fd', 'ts': 1624356529}
12. 10.0.12.9 - - [22/Jun/2021 18:08:49] "POST /listen HTTP/1.1" 200 -
```

附录引用：

[1] <https://gluster.readthedocs.io/en/latest/Administrator-Guide/Events-APIs/>