# GlusterFS Setup

**Version 2.0.2**

### Prepare

GlusterFS is an open source distribute file system which has some very good features, here we prepare real machines to setup a small distribute file system with only two servers.

|  |  |
| --- | --- |
| **Hardware** | **Description** |
| CPU | Pentium ® Dual-Core E5300 @ 2.60GHz |
| RAM | 1 GB |
| Hard Disk | 66 GB |
| Network Card | 100baseTx-FD |
| Switch | D-Link 10/100 Fast Ethernet Switch |

|  |  |
| --- | --- |
| **Software** | **Description** |
| Operating System | Ubuntu 10.04 server |
| Kernel | Linux Ubuntu 2.6.32-28-server |

Connect these two servers to the same trusted sub-network.

|  |  |
| --- | --- |
| **Server** | **IP** |
| Server1(also as client) | 172.17.13.28 (whatever as you like) |
| Server2 | 172.17.13.29 |

Now, start the server, and make sure that the networking connection works fine. And, also, we have to install some software from the mirror, so please make sure all the servers can access the Internet freely.

### Software Install

Now, we begin to install some basic software to support Gluster.

* 1. Install from DEB

We can download the DEB file from: <http://download.gluster.com/pub/gluster/glusterfs/3.1/LATEST/Ubuntu/>

Then we get ***glusterfs\_3.1.7-1\_amd64.deb***, type the command below in Terminal (Need root privilege):

cd /path/to/the/file

dpkg –install glusterfs\_3.1.7-1\_amd64.deb

* 1. Install from mirror

We can also install GlusterFS from mirror. First, add sources which you can access from the server to the APT source list, then run the following command:

apt-get update

apt-get install gluster-server gluster-client

*Note:* Only one of the servers needs to install gluster-client, here is Server1.

### Configuration

We’ll see that we have some new command now if we type in “glu” then press TAB button.

glusterfs glusterfsd

Before we start up gluster client, we have to configure these two gluster file server first.

* 1. Server Configuration

Type the commands below to make 4 directories, which will be used as storage volume.

root@ubuntu:~# cd /root/

root@ubuntu:~# mkdir -p gluster/bricks{0,1,2,3}

root@ubuntu:~# tree

.

└── gluster

├── bricks0

├── bricks1

├── bricks2

└── bricks3

5 directories, 0 files

Go to ***/etc/glusterfs***, we can see two files here: **glusterfs.vol, glusterfsd.vol**.

root@ubuntu:~# cd /etc/glusterfs/

root@ubuntu:/etc/glusterfs# ls

glusterfsd.vol glusterfs.vol

For server, we only have to open **glusterfsd.vol**

root@ubuntu:/etc/glusterfs# vim glusterfsd.vol

edit the content as follows(On ***Server1***):

### file: client-volume.vol.sample

#####################################

### GlusterFS Client Volume File ##

#####################################

#### CONFIG FILE RULES:

### "#" is comment character.

volume brick0

type storage/posix # POSIX FS translator

option directory /root/gluster/bricks0 # Export this directory

end-volume

volume brick1

type storage/posix # POSIX FS translator

option directory /root/gluster/bricks1 # Export this directory

end-volume

volume brick2

type storage/posix # POSIX FS translator

option directory /root/gluster/bricks2 # Export this directory

end-volume

volume brick3

type storage/posix # POSIX FS translator

option directory /root/gluster/bricks3 # Export this directory

end-volume

### Add network serving capability to above brick.

volume server

type protocol/server

option transport-type tcp

# option client-volume-filename /etc/glusterfs/glusterfs-client.vol

subvolumes brick0 brick1 brick2 brick3

# NOTE: Access to any volume through protocol/server is denied by

# default. You need to explicitly grant access through # "auth"

# option.

option auth.addr.brick0.allow 127.0.0.1 # Allow access to "brick" volume

option auth.addr.brick1.allow 127.0.0.1 # Allow access to "brick" volume

option auth.addr.brick2.allow 127.0.0.1 # Allow access to "brick" volume

option auth.addr.brick3.allow 127.0.0.1 # Allow access to "brick" volume

end-volume

And on ***Server2,*** do the same thing but the last few lines have to be modified to:

volume server

type protocol/server

option transport-type tcp

# option client-volume-filename /etc/glusterfs/glusterfs-client.vol

subvolumes brick0 brick1 brick2 brick3

# NOTE: Access to any volume through protocol/server is denied by

# default. You need to explicitly grant access through # "auth"

# option.

option auth.addr.brick0.allow 172.17.13.28 # Allow access to "brick0" volume

option auth.addr.brick1.allow 172.17.13.28 # Allow access to "brick1" volume

option auth.addr.brick2.allow 172.17.13.28 # Allow access to "brick2" volume

option auth.addr.brick3.allow 172.17.13.28 # Allow access to "brick3" volume

end-volume

*Note:* The server only provides disk space to store the files, of course we can let one server has only one storage volume, and configure 8 or more servers. Here is just an example.

* 1. Client Configuration

Because GlusterFS requires FUSE module, so first make sure that the FUSE module is installed properly in your Linux kernel. (More information about FUSE, please visit <http://fuse.sourceforge.net/> )

Change to directory ***/etc/glusterfs***, backup the original file **glusterfs.vol** to **glusterfs.vol.bg** , and clear the content.

root@ubuntu:~# cd /etc/glusterfs/

root@ubuntu:/etc/glusterfs# mv glusterfs.vol glusterfs.vol.bg

root@ubuntu:/etc/glusterfs# cat /dev/null > glusterfs.vol

Then add the content below:

### Add client feature and attach to remote subvolume

volume client0-0

type protocol/client

option transport-type tcp

option remote-host 127.0.0.1 # IP address of the remote brick

option remote-subvolume brick0 # name of the remote volume

end-volume

volume client0-1

type protocol/client

option transport-type tcp

option remote-host 172.17.13.29 # IP address of the remote brick

option remote-subvolume brick0 # name of the remote volume

end-volume

volume bricks0

type cluster/replicate # replicate means the two volume

# will write synchronously

subvolumes client0-0 client0-1

end-volume

volume client1-0

type protocol/client

option transport-type tcp

option remote-host 172.17.13.29 # IP address of the remote brick

option remote-subvolume brick1 # name of the remote volume

end-volume

volume client1-1

type protocol/client

option transport-type tcp

option remote-host 172.17.13.29 # IP address of the remote brick

option remote-subvolume brick1 # name of the remote volume

end-volume

volume bricks1

type cluster/replicate

subvolumes client1-0 client1-1

end-volume

volume filesys

type cluster/distribute # distribute means the two volume

# is treated as one volume, it likes

# RAID0, but not totally. One of

# the volume will be chose while

# writing files somehow.

subvolumes bricks0 bricks1

end-volume

### Add readahead feature

#volume readahead

# type performance/read-ahead

# option page-size 1MB # unit in bytes

# option page-count 2 # cache per file = (page-count x page-size)

# subvolumes client

#end-volume

[2011-09-20 01:36:25] N [server-protocol.c:5812:mop\_setvolume] server: accepted client from 127.0.0.1:1021

[2011-09-20 01:36:25] N [server-protocol.c:5812:mop\_setvolume] server: accepted client from 127.0.0.1:1020

### Testing

Now everything is ready, let’s start the services! On ***Server1*** and ***Server2***, type in:

root@ubuntu:~# glusterfsd

and open the log file at ***/var/log/glusterfs/-etc-glusterfs-glustefsd.vol.log***.

root@ubuntu:~# tail /var/log/glusterfs/-etc-glusterfs-glustefsd.vol.log

If you see **“Successfully started”**, then congratulations! If not, according to the logs and figure out what’s wrong.

After both of the servers have successfully started, we can use their disk space by mount it to the client’s local file system:

root@ubuntu:~# mount –t glusterfs /etc/glustefs/glustefs.vol /mnt

root@ubuntu:~# cd /mnt

root@ubuntu:/mnt# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/ubantu-root

71G 1.1G 66G 2% /

none 490M 204K 490M 1% /dev

none 495M 0 495M 0% /dev/shm

none 495M 60K 495M 1% /var/run

none 495M 0 495M 0% /var/lock

none 495M 0 495M 0% /lib/init/rw

/dev/sda1 228M 17M 199M 8% /boot

/etc/glusterfs/glusterfs.vol

7.5G 3.2G 4.0G 45% /mnt

Let’s make some test by using these commands:

root@ubuntu:/mnt# touch test\_file{0,1,2,3}

root@ubuntu:/mnt# mkdir test\_dir{0,1}

root@ubuntu:/mnt# ls

test\_file0 test\_file1 test\_file2 test\_file3 test\_dir0 test\_dir1

And on Server1:

root@ubuntu:~# tree

.

└── gluster

├── bricks0

│   ├── test\_file0

│   ├── test\_file2

│   ├── test\_dir0

│   └── test\_dir1

├── bricks1

│   ├── test\_file1

│   ├── test\_file3

│   ├── test\_dir0

│   └── test\_dir1

├── bricks2

└── bricks3

On Server2:

root@ubuntu:~# tree

.

└── gluster

├── bricks0

│   ├── test\_file0

│   ├── test\_file2

│   ├── test\_dir0

│   └── test\_dir1

├── bricks1

│   ├── test\_file1

│   ├── test\_file3

│   ├── test\_dir0

│   └── test\_dir1

├── bricks2

└── bricks3

We see that Server1 and Server2 have the same content, and if we delete a file from one Server1, it will recover after some time automatically!

The architecture is as follows:

Brick0

Brick0

replicate

replicate

Brick1

Brick1

replicate

replicate

Brick2

Brick2

replicate

BrickN

BrickN

replicate

……

……

**GlusterFS**