GLUSTERFS

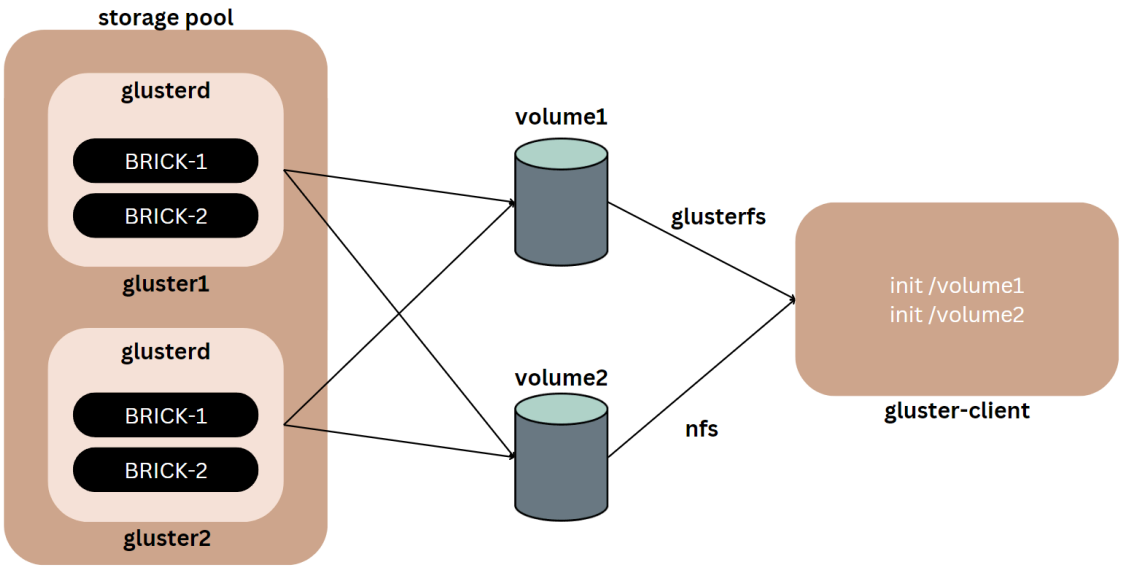
TEAM: 121AD0012, 121AD0014, 121AD0019, 121AD0020

What is GlusterFS and why do we need it?

GlusterFS (GFS) is an open-source distributed file system designed to provide scalable and high-performance storage for various types of data. It is part of the Gluster storage software suite, initially developed by Gluster Inc. and later acquired by Red Hat. GlusterFS is built to run on commodity hardware, allowing users to create a distributed storage infrastructure that can easily scale as their storage needs grow. Main advantages to note are:

1. Distributed architecture
2. Flexibility
3. Scalability
4. High Availability
5. Elasticity

BASIC ARCHITECTURE USED:



This is to be implemented on the Ubuntu systems. We will have three machines, two of which will be servers and one client.

Installing glusterfs on server and client nodes

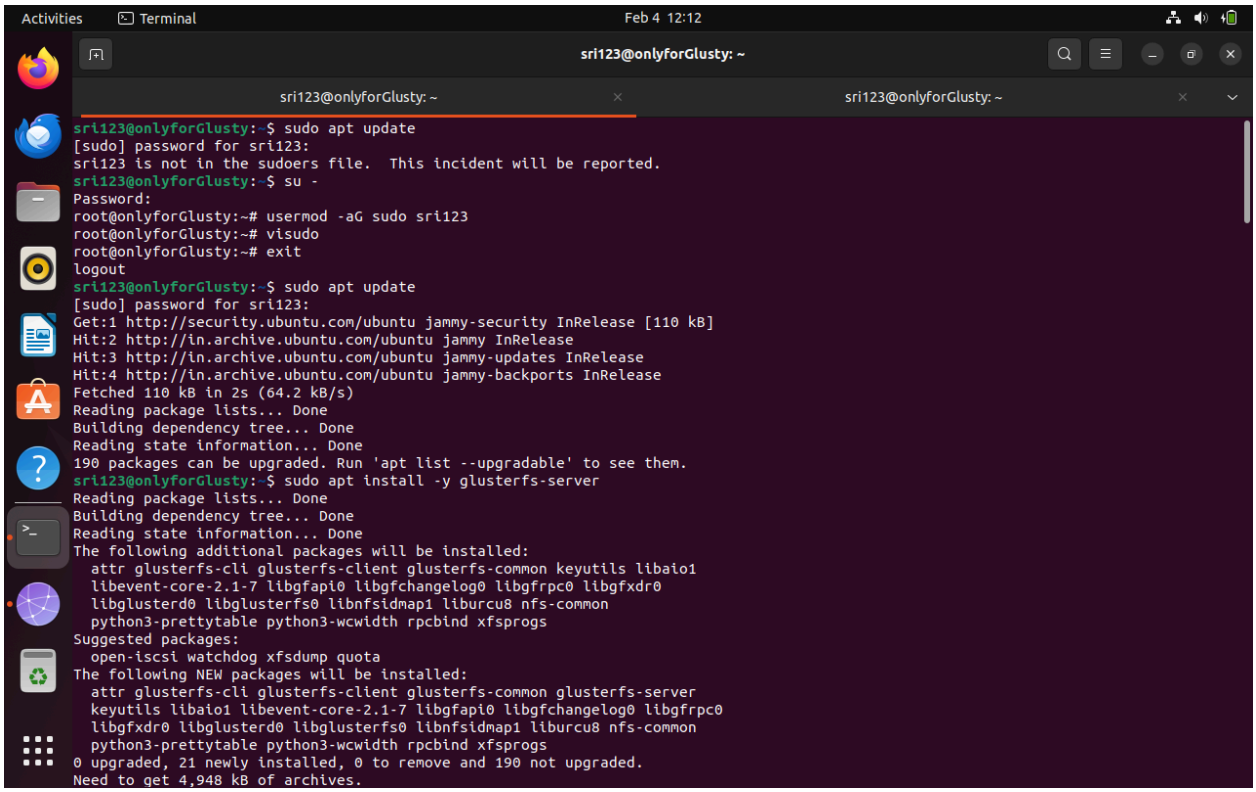
Command: `lsblk` , gives the disks and partitions on the machine.

On both server nodes, first we execute the following commands:

sudo apt update
sudo apt install -y glusterfs-server
sudo systemctl start glusterd
sudo systemctl enable glusterd
sudo systemctl status glusterd

On client node:

sudo apt update
sudo apt install -y glusterfs-client

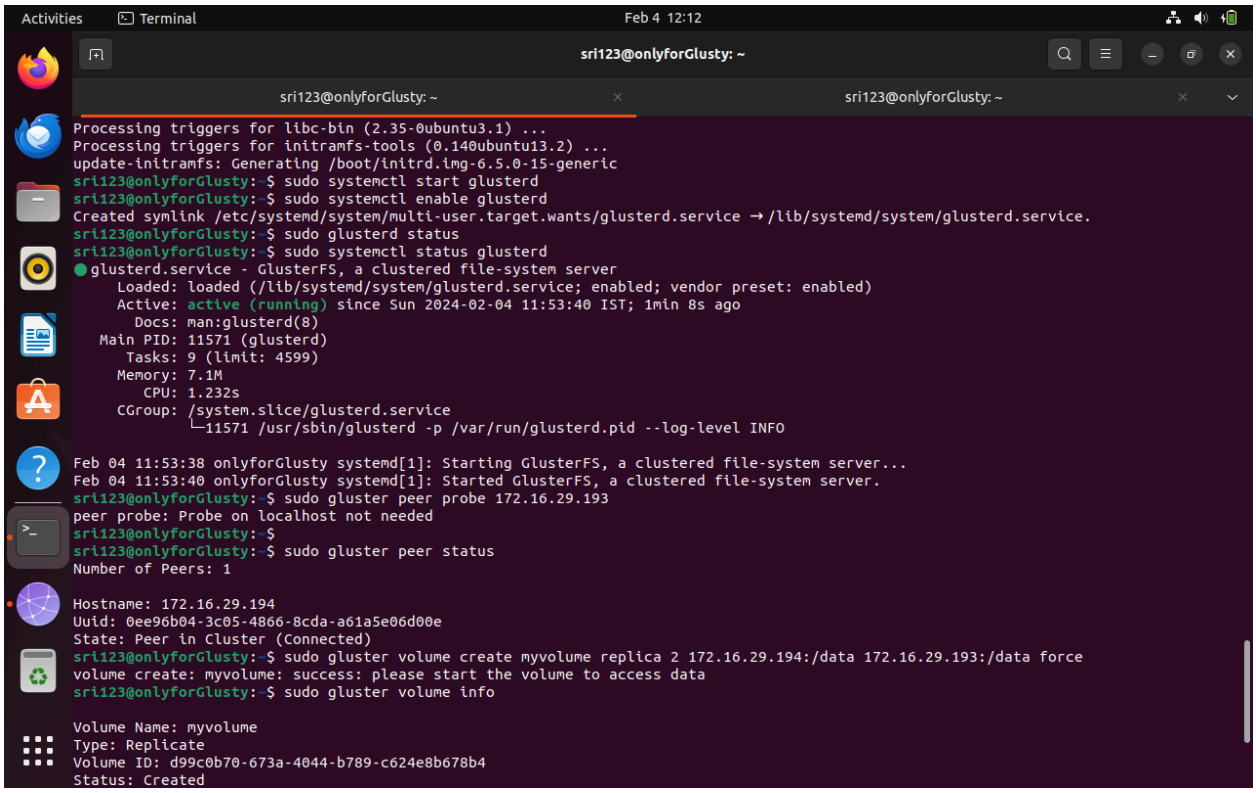


Configuring the servers as a single storage pool

Now on one of the servers, we need to configure them to create a trusted storage pool. In order to do so: *sudo gluster peer probe <server1>* is executed, where the server1 is another machine server along with the one on which this command is being run.

Command to check status:

sudo gluster peer status

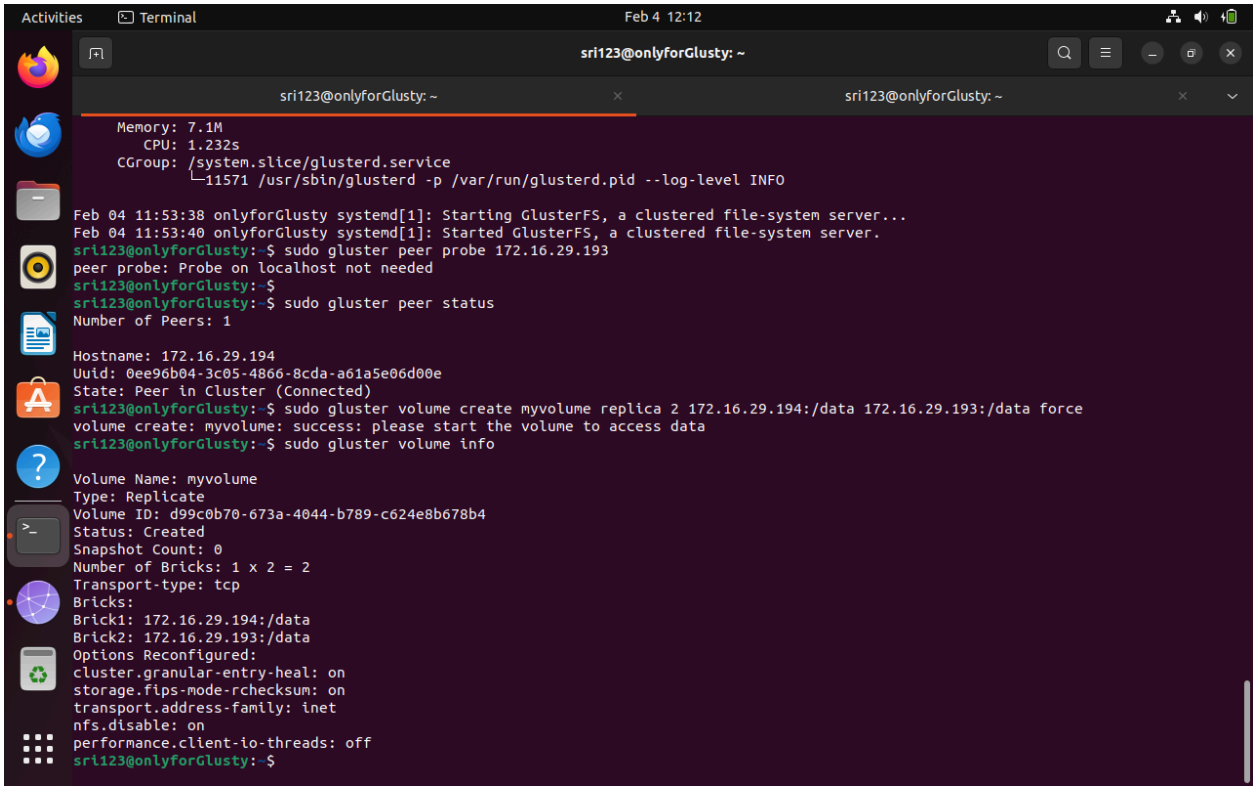


Creating and mounting Volume

sudo gluster volume create ourvolume replica 2 <server1>:/data <server2>:/data force

This command creates a volume or a storage space and its replica on both the servers. This command when run on one of the servers makes changes to both of the server nodes. To check this fact, we can execute:

sudo gluster volume info



Testing the setup

With the above commands we have finished the setting of two server nodes and one client node. Now we need to check whether the server has the client files or not. For this we need to first make the volume active or running:

sudo gluster volume start ourvolume

After executing this command one one of the servers, we may check the volume status on both the servers to be active or running:

sudo gluster volume status

In this the volume should be 'running' in order for the mounting data files by the client.

Now on client node:

sudo mkdir /mnt/glusterfs
sudo mount -t glusterfs <server_ip>:/myvolume /mnt/glusterfs

These commands will create a local directory that will be shared with both the servers. We can verify the mount using command: *df -h*

```
hemanth@hemanth:/data$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           726M  2.3M  724M   1% /run
/dev/nvme0n1p6   96G   18G   74G  19% /
tmpfs           3.6G     0  3.6G   0% /dev/shm
tmpfs           5.0M   20K  5.0M   1% /run/lock
/dev/nvme0n1p1  256M   90M  167M  35% /boot/efi
tmpfs           726M  104K  726M   1% /run/user/1000
```

```
chandu@chandu:/data$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           768M  2.3M  766M   1% /run
/dev/sda5       143G   37G   99G  27% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
/dev/sda4       1.1G   34M  1.1G   4% /boot/efi
tmpfs           768M  112K  768M   1% /run/user/1000
/dev/sdb1        15G  5.1G  9.4G  36% /media/chandu/SONY
chandu@chandu:/data$
```

```
chandu@chandu:~$ sudo gluster volume status
[sudo] password for chandu:
Status of volume: ourvolume
Gluster process                         TCP Port  RDMA Port  Online  Pid
------------------------------------------------------------------------------
Brick 172.16.30.153:/data               55810     0          Y       18642
Brick 172.16.28.242:/data               60320     0          Y       14193
Self-heal Daemon on localhost           N/A       N/A        Y       14210
Self-heal Daemon on 172.16.30.153       N/A       N/A        Y       18659

Task Status of Volume ourvolume
------------------------------------------------------------------------------
There are no active volume tasks

chandu@chandu:~$
```

```
hemanth@hemanth:~$ sudo gluster volume status
[sudo] password for hemanth:
Status of volume: ourvolume
Gluster process                         TCP Port  RDMA Port  Online  Pid
------------------------------------------------------------------------------
Brick 172.16.30.153:/data               55810     0          Y       18642
Brick 172.16.28.242:/data               60320     0          Y       14193
Self-heal Daemon on localhost           N/A       N/A        Y       18659
Self-heal Daemon on 172.16.28.242       N/A       N/A        Y       14210

Task Status of Volume ourvolume
------------------------------------------------------------------------------
There are no active volume tasks
```

```
chandu@chandu:~$ sudo gluster volume info

Volume Name: ourvolume
Type: Replicate
Volume ID: 640bfd62-1d57-448b-904f-0df66861c763
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: 172.16.30.153:/data
Brick2: 172.16.28.242:/data
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
chandu@chandu:~$
```

```
hemanth@hemanth:~$ sudo gluster volume info

Volume Name: ourvolume
Type: Replicate
Volume ID: 640bfd62-1d57-448b-904f-0df66861c763
Status: Created
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: 172.16.30.153:/data
Brick2: 172.16.28.242:/data
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

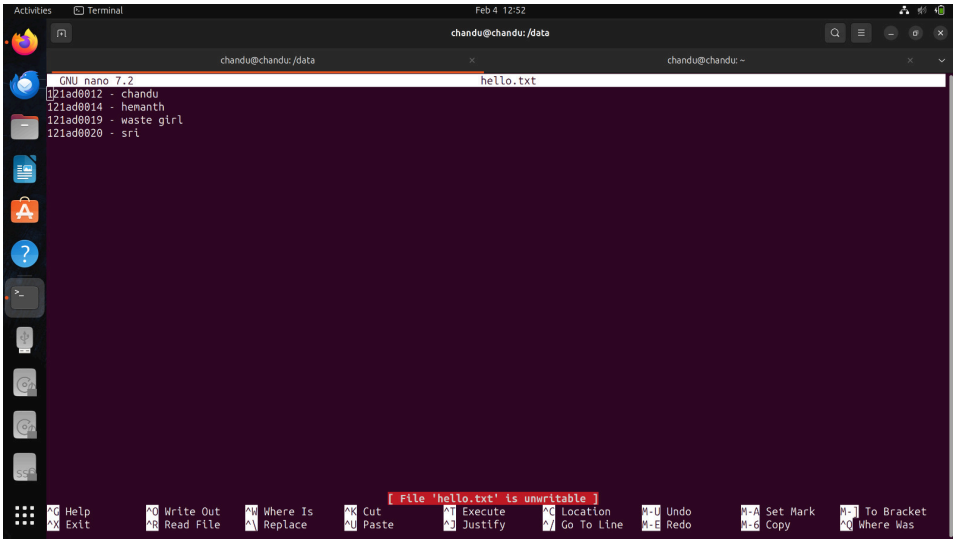
Now we create a file in client and save it to see whether the file is present in servers:

echo "Hello, chandu, hemanth!" | sudo tee /mnt/glusterfs/testfile.txt

This creates a text file in the /mnt/glusterfs directory, and on going to /data it shows the file in servers as well.

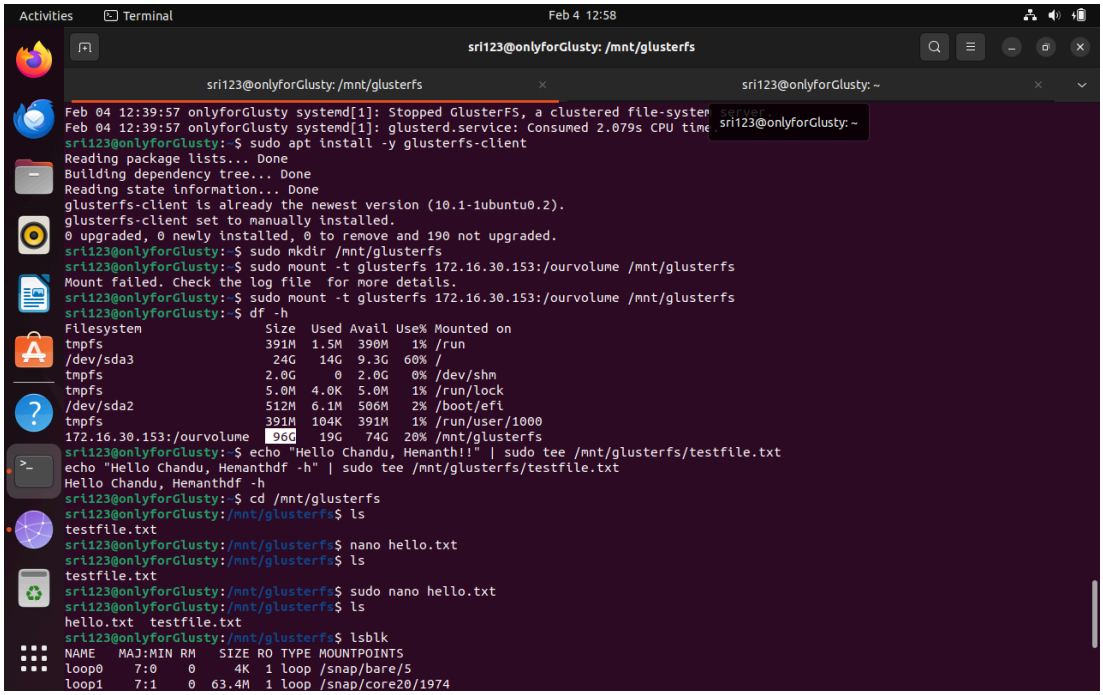
```
hemanth@hemanth:~$ ls /data
testfile.txt
```

We notice that on accessing the file, we are only able to see and not to edit any of its contents on the servers.



Similarly on writing another file in the same directory, it is also showing in both the servers.

Client:



Servers: