****

Create a Redundant Storage Pool Using GlusterFS on Ubuntu Servers

****

Redundancy and high availability are necessary for a very wide variety of server activities. Having a single point of failure in terms of data storage is a very dangerous configuration for any critical data.

While many databases and other software allows you to spread data out in the context of a single application, other systems can operate on the filesystem level to ensure that data is copied to another location whenever it is written to disk. A clustered storage solution like **GlusterFS** provides this exact functionality.

A clustered environment allows you to pool resources (generally either computing or storage) in order to allow you to treat various computers as a single, more powerful unit. With GlusterFS, we are able to pool the storage of various VPS instances and access them as if it were a single server.

GlusterFS allows you to create different kinds of storage configurations, many of which are functionally similar to RAID levels. For instance, you can stripe data across different nodes in the cluster, or you can implement redundancy for better data availability.

1) Replicate
2) Distribute

****

server configuration

****

note ::
if it is docker container, then run with –privillaged , -v volume (for mount to container)

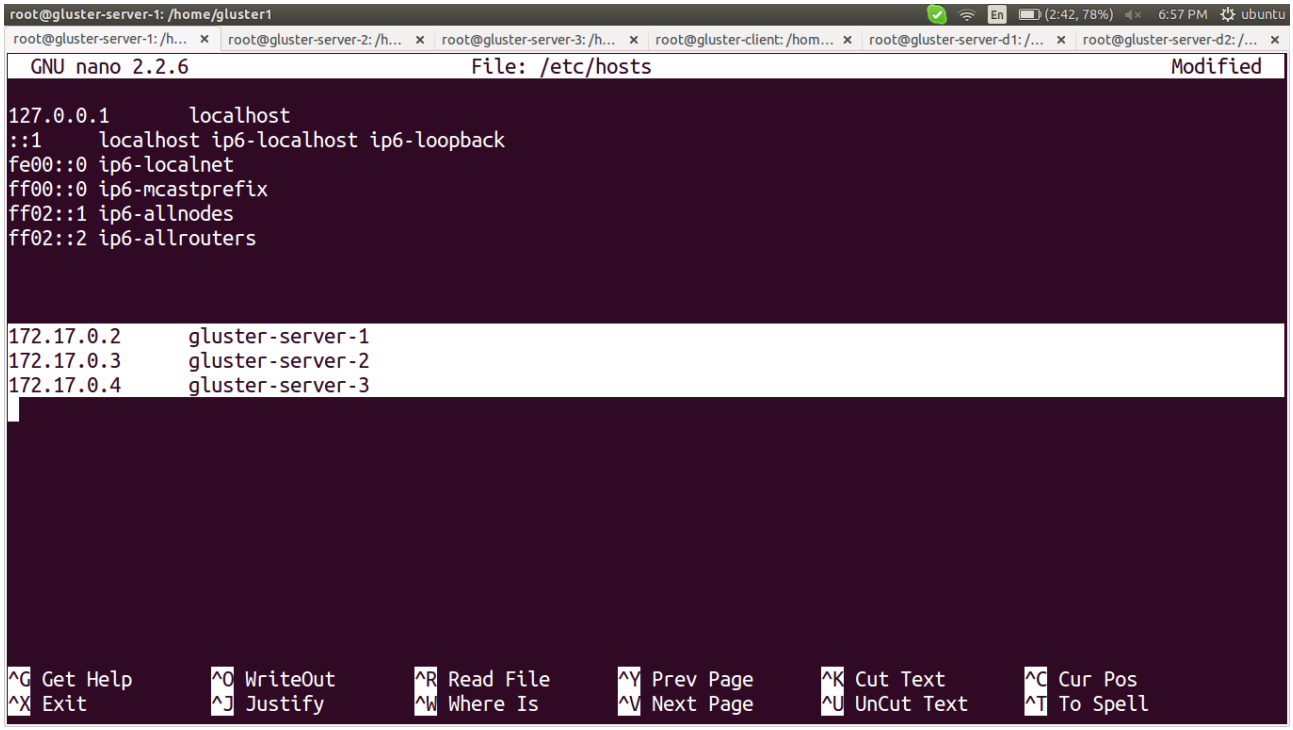
eg :: docker run -it --privileged -v /home/gluster1 --name gluster-server-1 --hostname gluster-server-1 ubuntu:14.04 /bin/bash

give hostname and FQDN

 # nano /etc/hostname
 # nano /etc/hosts

edit /etc/hosts for given all node name

 # nano /etc/hosts



in my case, i created 3 server-nodes,
like this edit /etc/hosts on all remaining nodes

 # apt-get update

 # apt-get install glusterfs-server

 # /etc/init.d/glusterfs-server start

 # glusterfs --version

Once this is installed on both nodes, we can begin to set up our storage volume.

On one of the hosts, we need to peer with the second host. It doesn't matter which server you use, but we will be preforming these commands from our gluster-server-1 server for simplicity:

 # gluster peer probe gluster-server-2

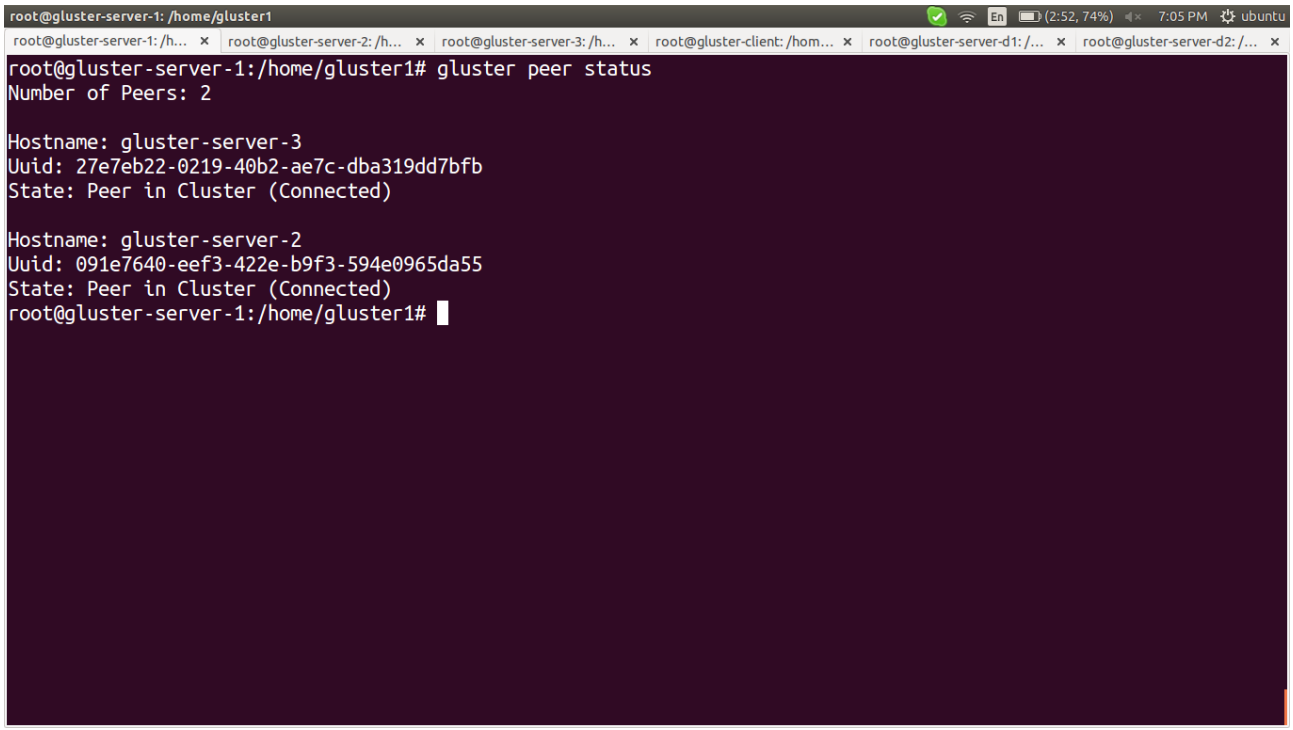
 # gluster peer probe gluster-server-3

 # gluster peer status

do this on each node, need not probe the gluster-server-21 in gluster-server-1 node, similarly in other nodes.

If it is success, it shows some result like follows,

peer probe: success



Create a Storage Volume

Two types::

1) Replicate
2) Distribute

create the directorys first for creating volume

1) Replicate >>

gluster volume create volume_name replica num_of_servers transport tcp domain1.com:/path/to/data/directory domain2.com:/path/to/data/directory ... force

eg ::

 # gluster volume create gluster-volume replica 3 transport tcp gluster-server-1:/home/gluster1 gluster-server-2:/home/gluster2 gluster-server-3:/home/gluster3 force

here “gluster-volume” is my volume's name

2) Distribute >>

for testing i created another 2 server-nodes for demostrate distribute, but i didnt mentioned here, it is easy to create, all steps are same except the below command.

just remove the replica portion from the above command,

 # gluster volume create gluster-volum ed transport tcp gluster-server-d1:/home/glusterd1 gluster-server-d2:/home/glusterd2 force

here “gluster-volumed” is my volume's name

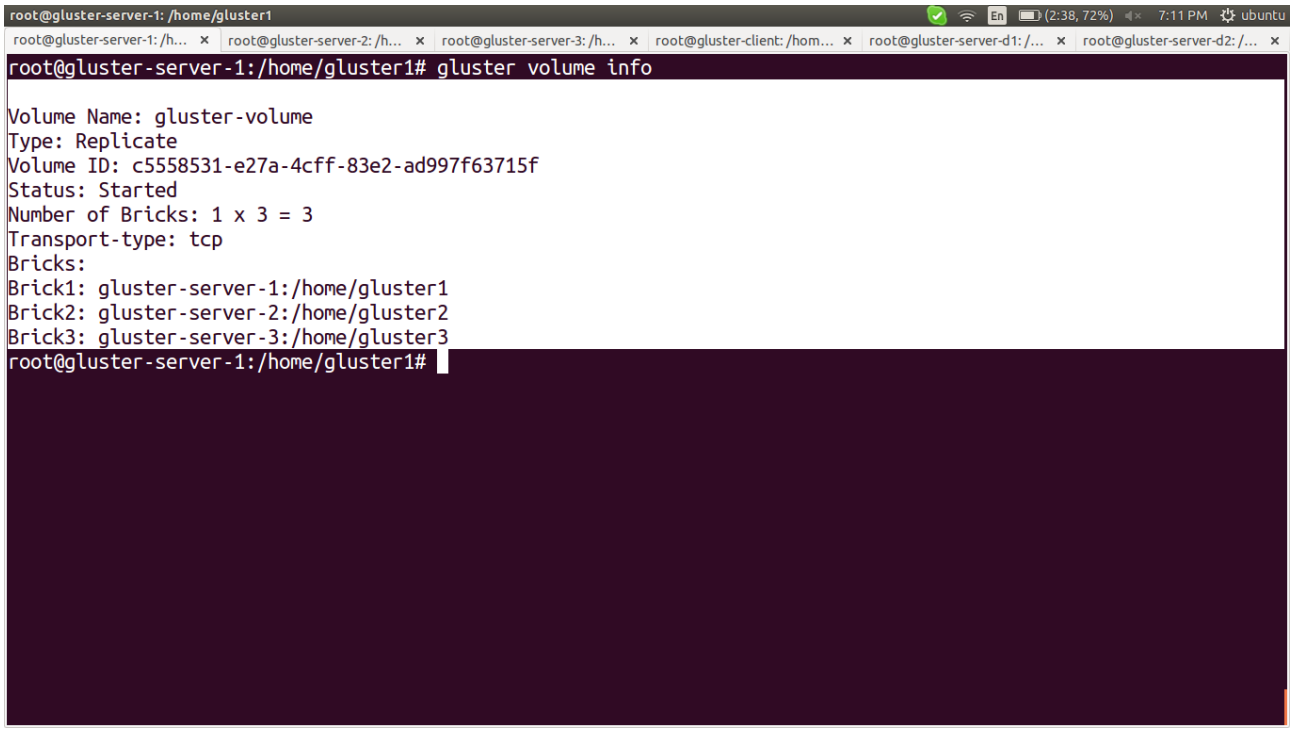
if it is success, then output become

volume create: gluster-volume: success: please start the volume to access data

 # gluster volume start gluster-volume

to see the volume info,

 # gluster volume info



****

Install and Configure the Client Components

****

Now that we have our volume configured, it is available for use by our client machine.

 # apt-get install glusterfs-client

This will install the client application, and also install the necessary fuse filesystem tools necessary to provide filesystem functionality outside of the kernel.

We are going to mount our remote storage volume on our client computer. In order to do that, we need to create a mount point. Traditionally, this is in the /mnt directory, but anywhere convenient can be used.

 # mkdir /home/gluster -client # for Replicate, , no need if you do with distribute only

 # mkdir /home/gluster-clientd # for Distribute, no need if you do with replicate only

then, mount

 # sudo mount -t glusterfs "domain1.com:volume_name" "path_to_mount_point"

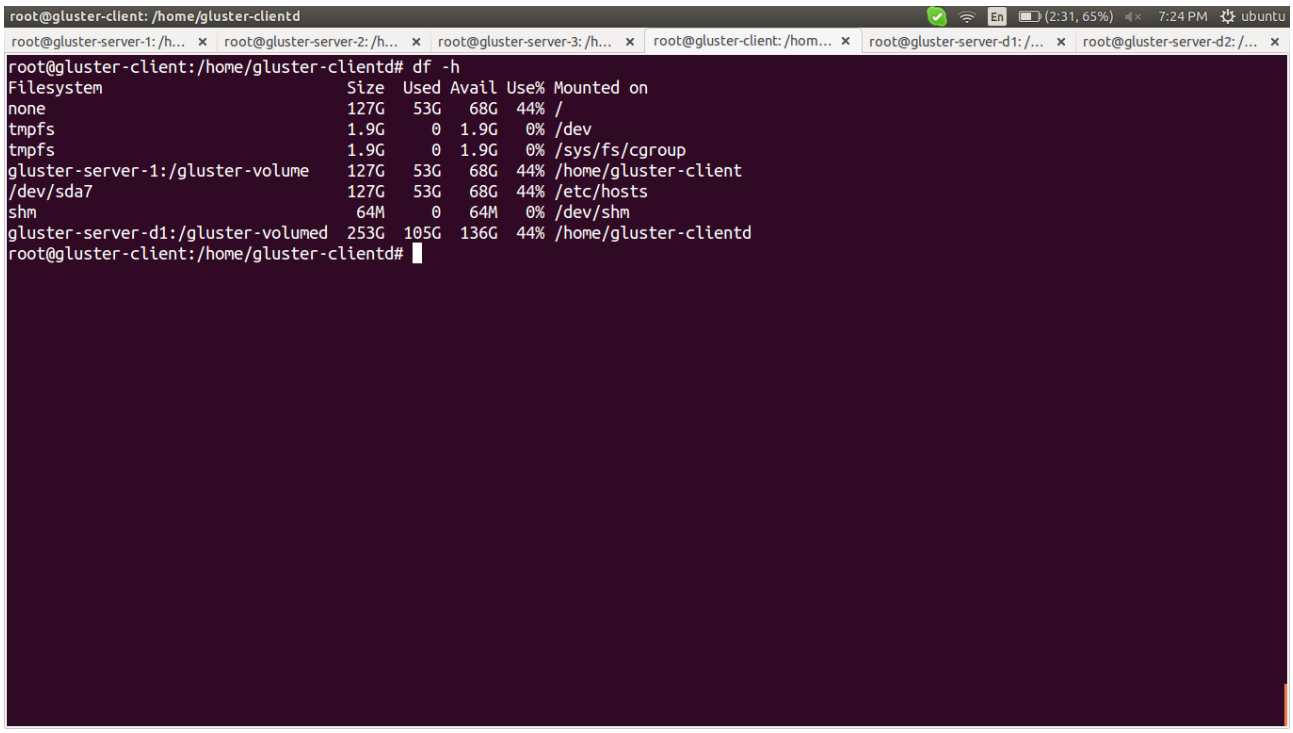
Notice that we are using the volume name in the mount command. GlusterFS abstracts the actual storage directories on each host. We are not looking to mount the /gluster1 directory, but the gluster-volume volume.

 # mount -t glusterfs gluster-server-1:gluster-volume /home/gluster-client # for replicate

 # mount -t glusterfs gluster-server-d1:/gluster-volumed /home/gluster-clientd # for distribute

ensure that mounted by,

 # df -h



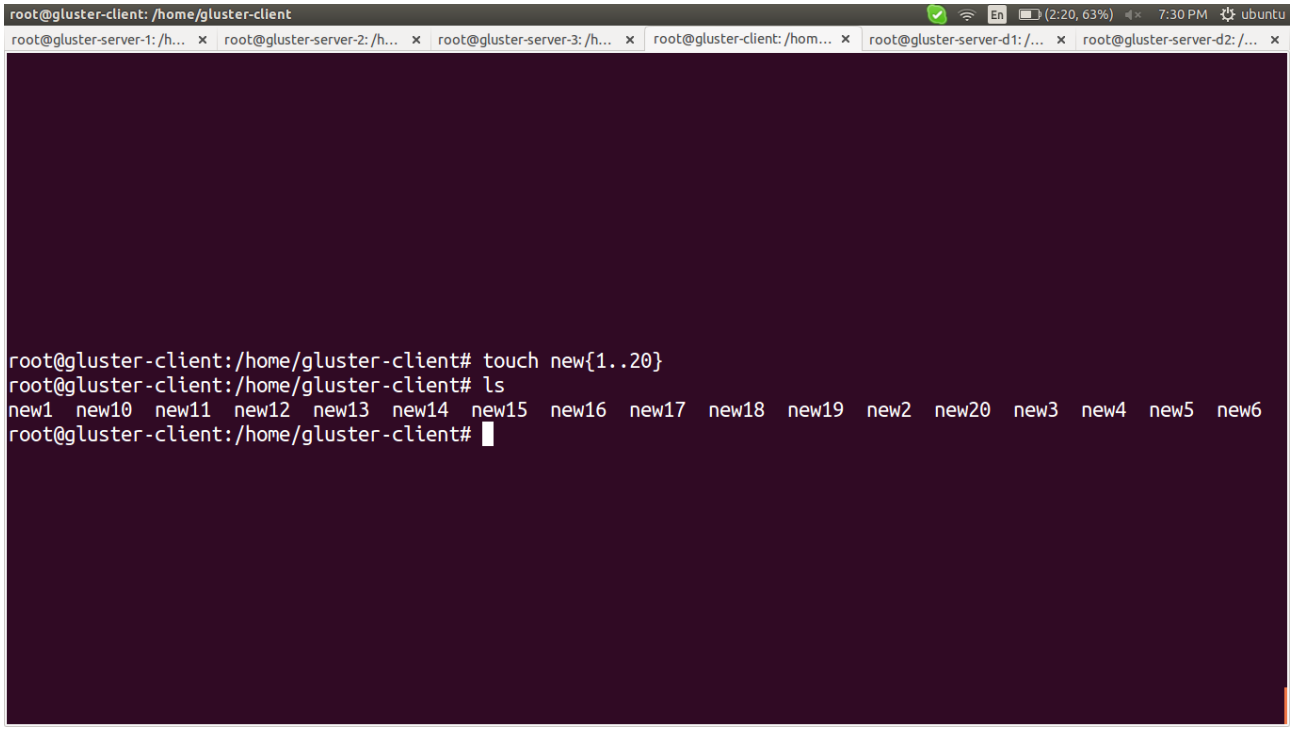
Testing

1) Replicate ::

 # cd /home/gluster-client

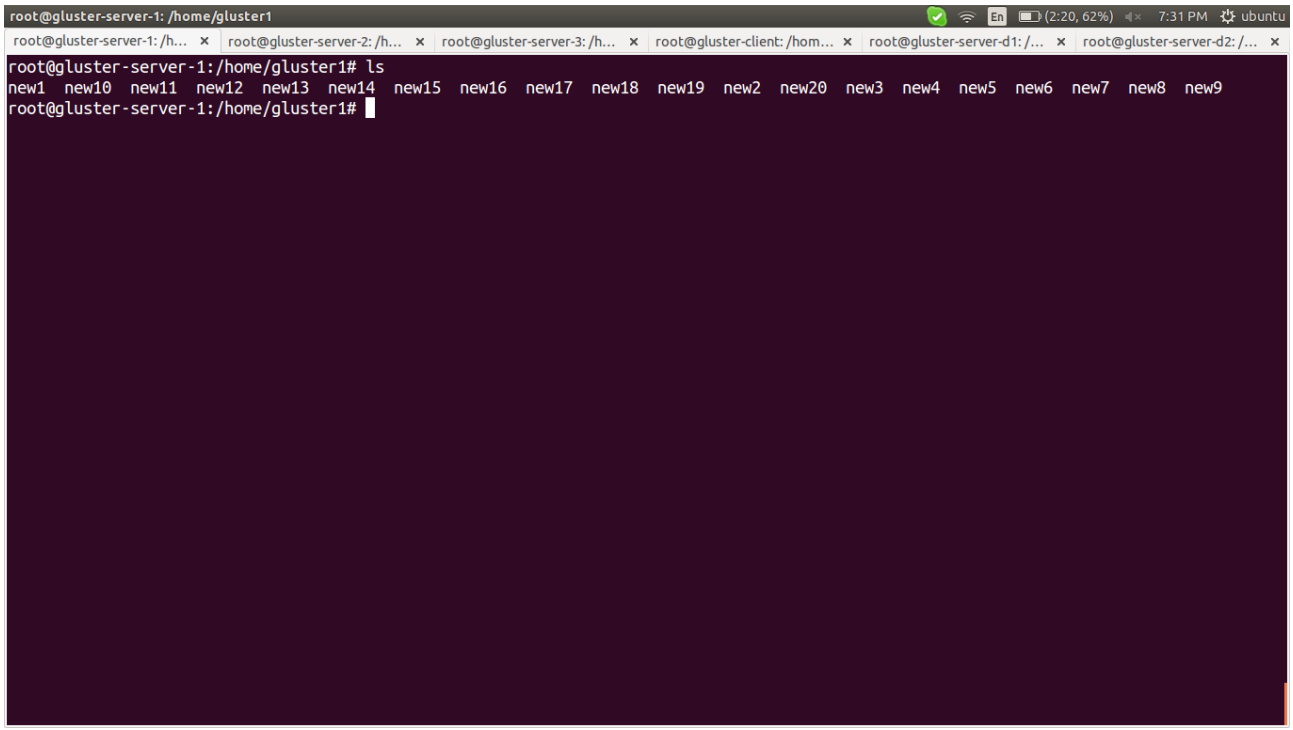
 # touch new{1..20}

 # ls

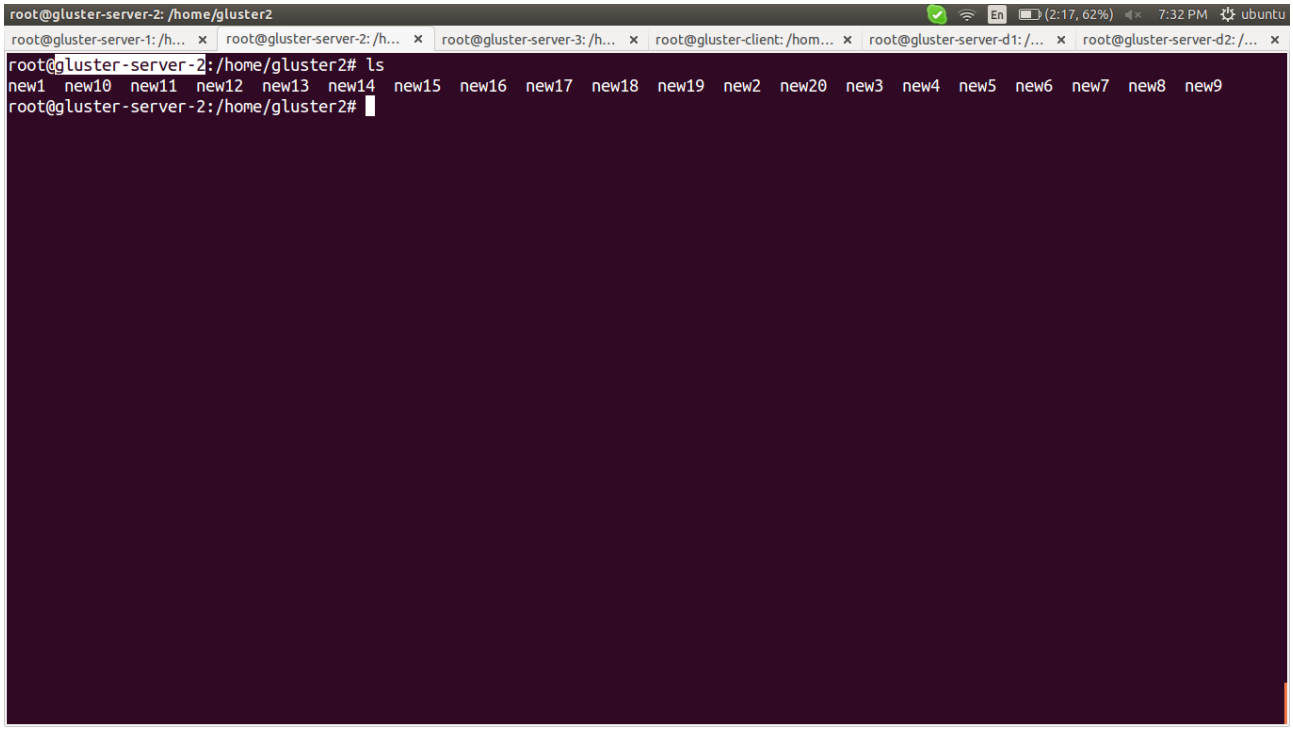


check in all server-node, it will reflect

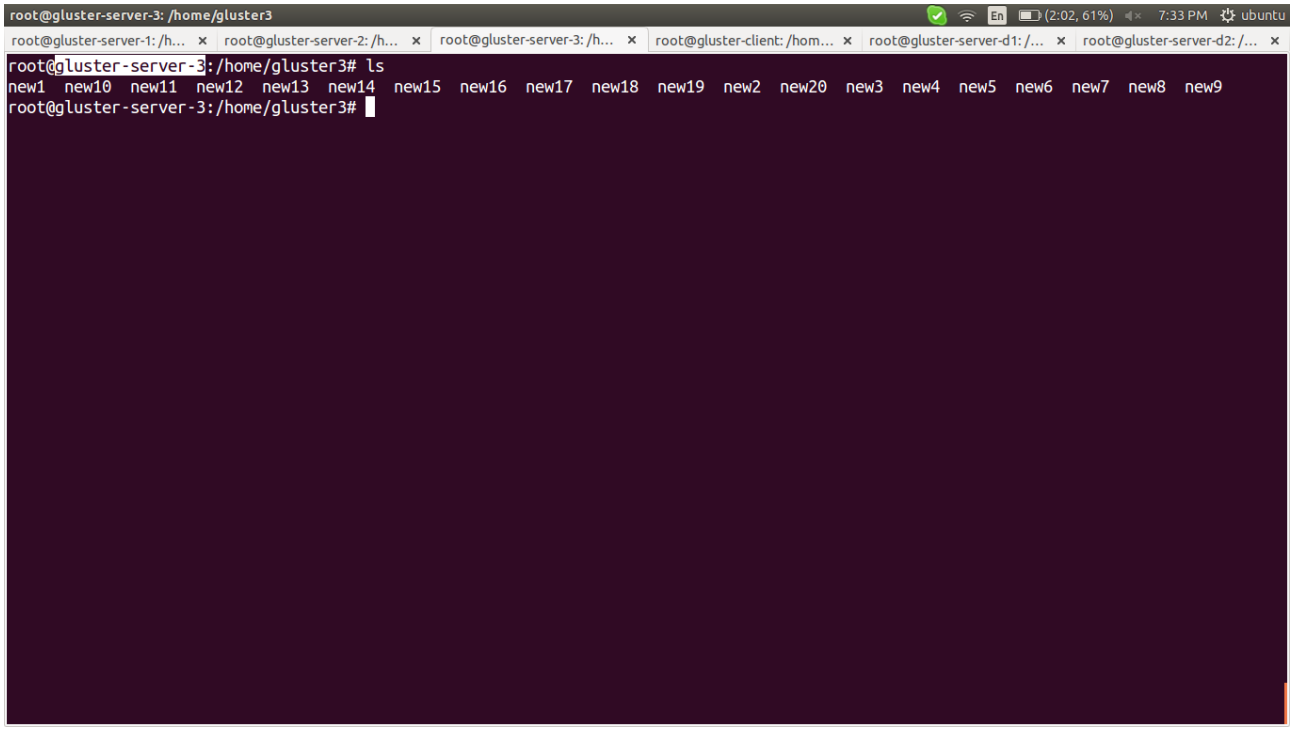
for gluster-server-1 >>



for gluster-server-2 >>



for gluster-server-3 >>

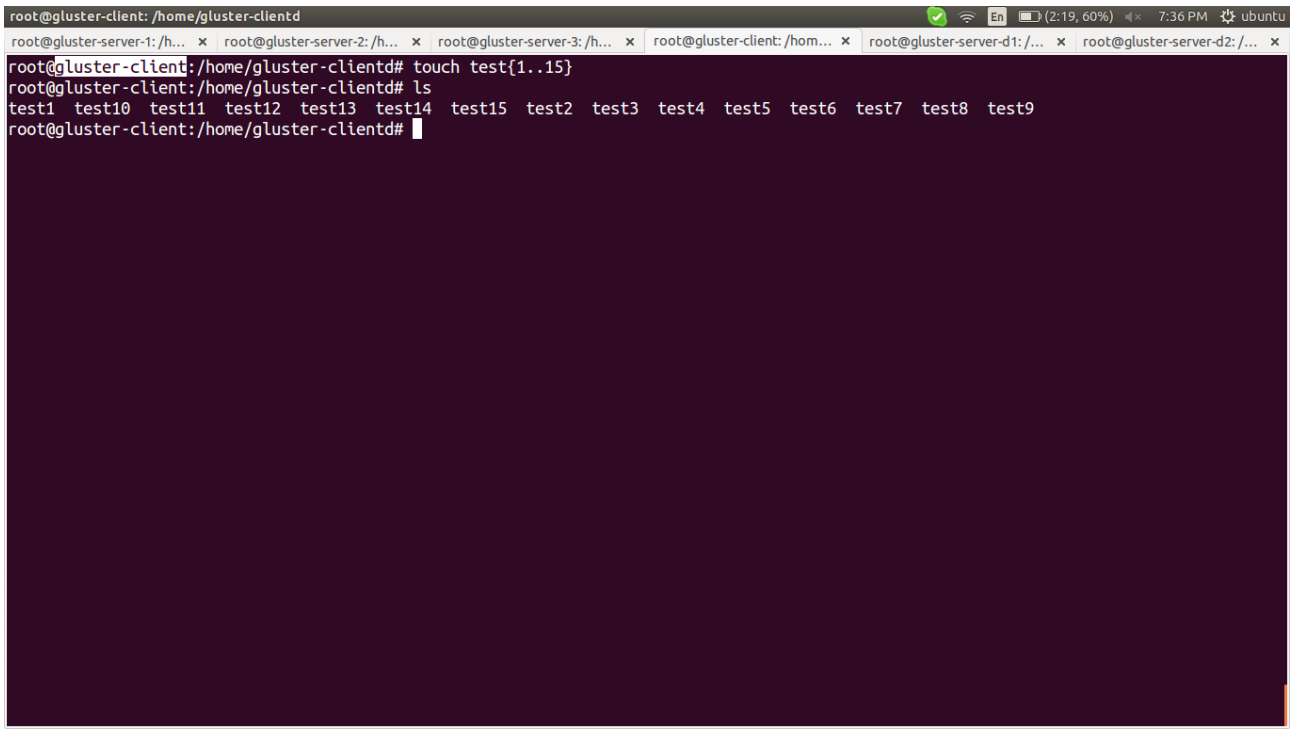


2) for Distribute ::

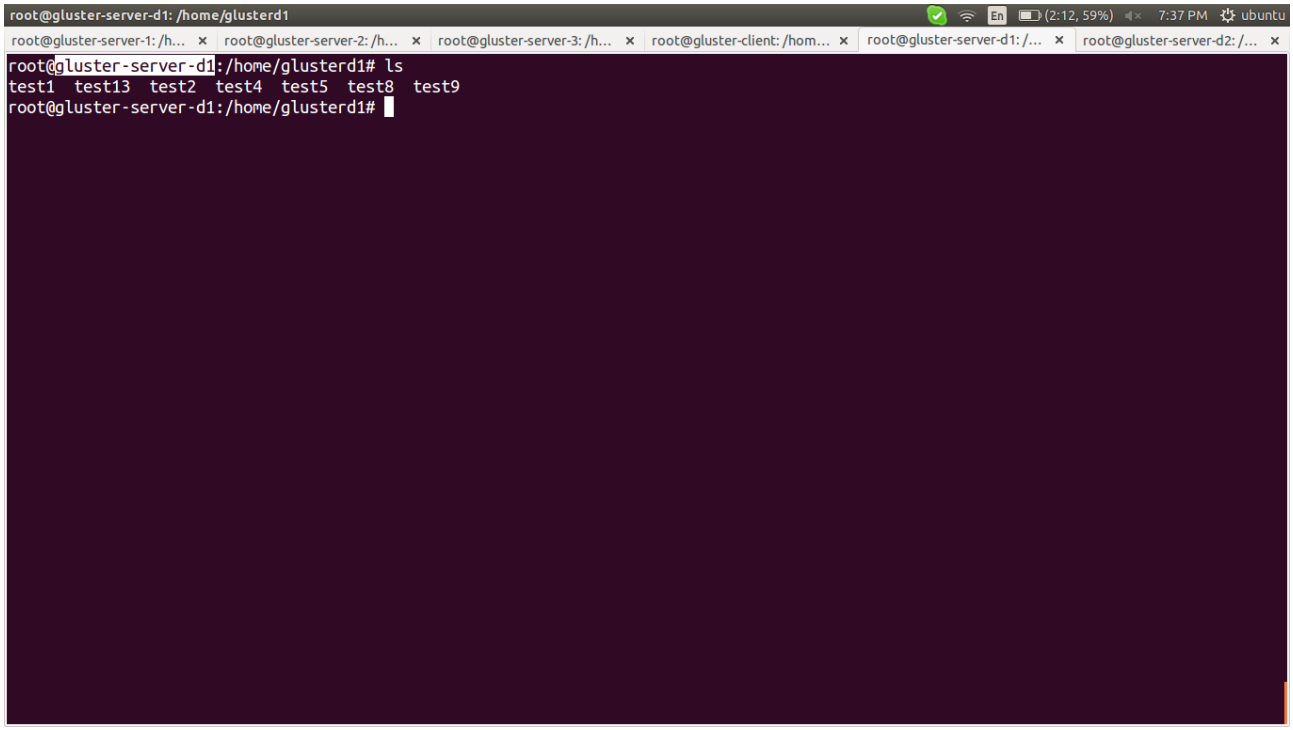
 # cd /home/gluster-clientd

 # touch test{1..15}

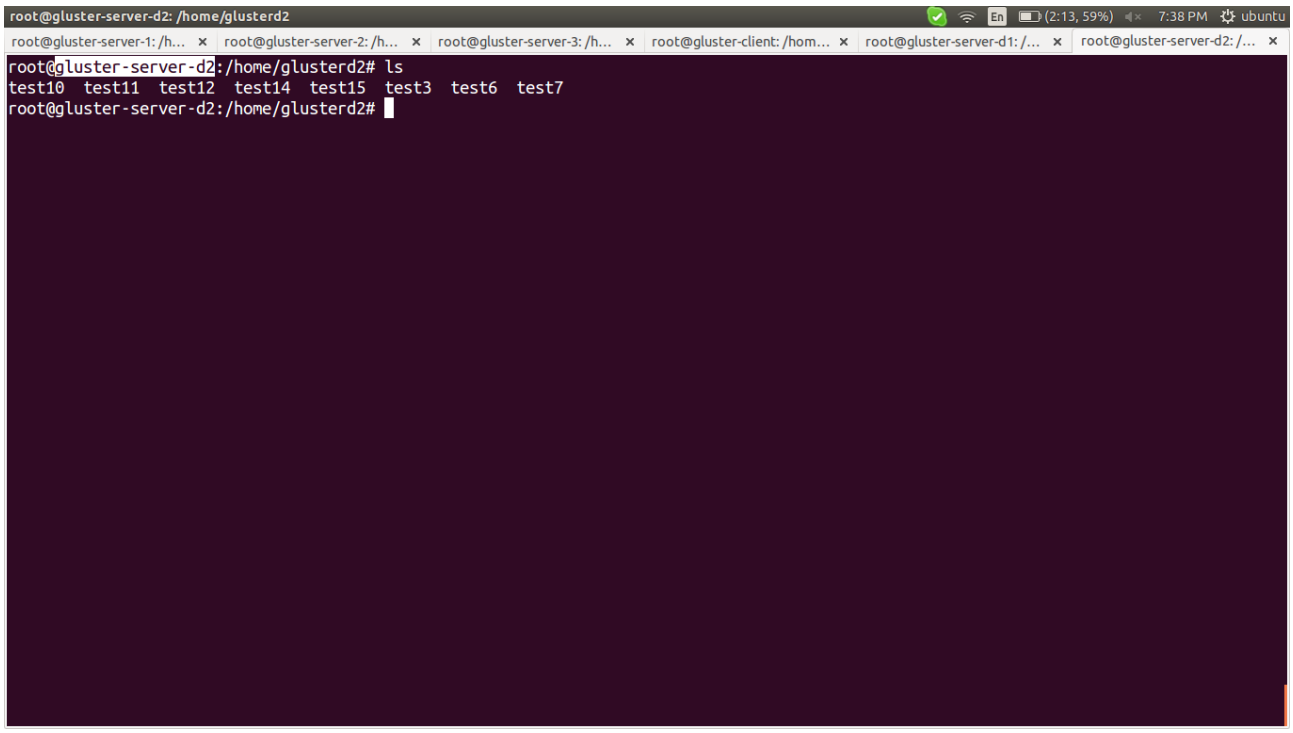
 # ls



for gluster-server-d1 >>



for gluster-server-d2 >>



for more options ::
https://www.digitalocean.com/community/tutorials/how-to-create-a-redundant-storage-pool-using-glusterfs-on-ubuntu-servers

****

links::
https://www.digitalocean.com/community/tutorials/how-to-create-a-redundant-storage-pool-using-glusterfs-on-ubuntu-servers

https://www.server-world.info/en/note?os=Ubuntu_14.04&p=glusterfs

https://www.youtube.com/watch?v=D76miE2HaK4

for docker:: error solution
http://neependra.net/?p=1048