LAB 6 – DEPLOYING CLUSTERFS IN UBUNTU SERVER

6.1 Introduction to ClusterFS:

6.1.1 The crucial requirments for DFS Replication:

Why Do We Need DFS (Distributed File System) Replication?

In modern IT environments, data is often accessed by many users and applications across different servers or locations. Relying on a single storage server creates problems:

- **Single Point of Failure**: If one server goes down, users lose access to data.
- **Scalability Limits**: A single machine cannot handle very large storage or high I/O workloads efficiently.
- **Disaster Recovery**: Without replication, hardware failures or corruption can result in permanent data loss.
- **Collaboration & Availability**: Multiple users and applications may need real-time access to the same set of files.

DFS services continuously and immediately replicate data across multiple servers, ensuring:

- High availability (if one server fails, others serve the data).
- Continuous synchronization (no need for manual copying).
- Transparent access (users see a single namespace, regardless of where data lives).
- Better load balancing and scalability.

6.1.2 GlusterFS on Ubuntu:

GlusterFS is an open-source, distributed file system that provides functionality very similar to Windows DFS.

- **Key Features**:
  - **Replication**: Keeps identical copies of files across servers.
  - **Scalability**: Add more servers to increase capacity and performance.
  - **Global Namespace**: Clients see one unified folder, even though data is spread across many servers.

    - **Self-Healing**: Automatically syncs files if a node comes back online after downtime.
    - **POSIX-Compliant**: Works like a normal Linux filesystem, so apps and users don’t need special tools.

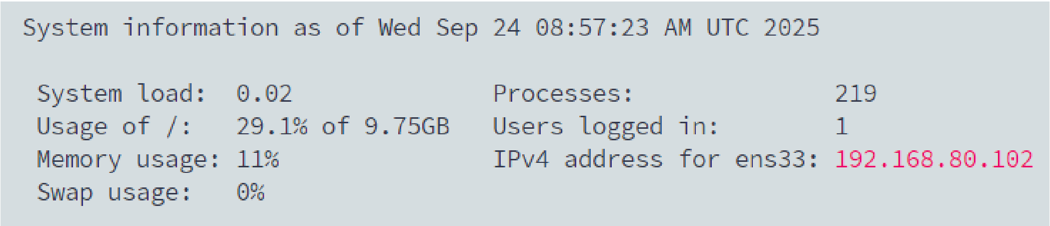
- **How It Works**:
    - You set up storage “bricks” (directories) on multiple servers.
    - These bricks are combined into a **Gluster Volume**.
    - Clients mount this volume just like an NFS or local folder.
    - Files written on one server are replicated and immediately available on all others.
- **Use Cases**:
    - Shared network storage for teams.
    - Backend storage for virtualization (VM disks).
    - Media repositories or file archives.
    - Backup targets with high availability.

6.2 Deploy GlusterFS on Ubuntu:

Step 1: Prepare Your Servers

- At least **2 Ubuntu servers** (can be VMs).
- Make sure they can resolve each other via DNS or /etc/hosts. Example:

  192.168.80.102 filesvr1

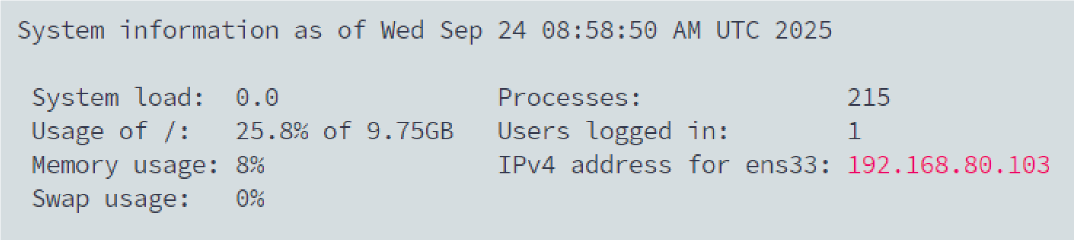


```
System information as of Wed Sep 24 08:57:23 AM UTC 2025

 System load:  0.02              Processes:             219
 Usage of /:   29.1% of 9.75GB   Users logged in:       1
 Memory usage: 11%               IPv4 address for ens33: 192.168.80.102
 Swap usage:   0%
```

Image 6.1: Information of filesvr1 remoted from Terminus tool

192.168.1.103 filesvr2



```
System information as of Wed Sep 24 08:58:50 AM UTC 2025

 System load:  0.0               Processes:               215
 Usage of /:   25.8% of 9.75GB   Users logged in:         1
 Memory usage: 8%                IPv4 address for ens33: 192.168.80.103
 Swap usage:   0%
```

Image 6.2: Information of filesvr2 remoted from Terminus tool

- Ensure ports (especially TCP/24007, TCP+UDP/24008, TCP/49152+) are open between nodes.

Step 2: Install GlusterFS

On **all nodes**:

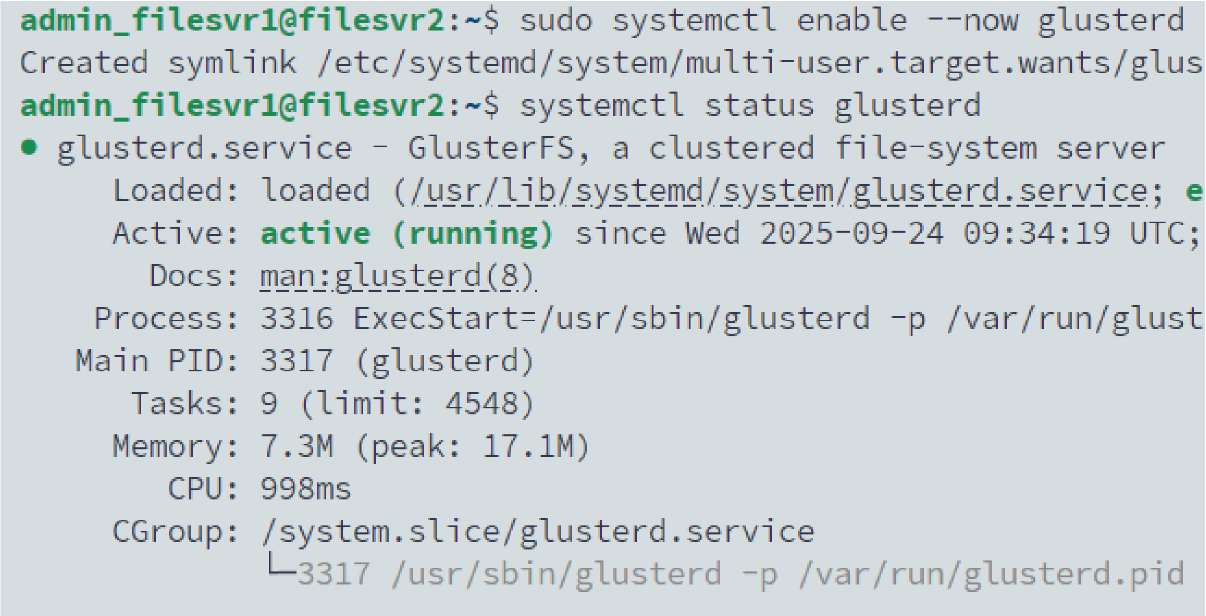
sudo apt update

sudo apt install glusterfs-server -y

sudo systemctl enable --now glusterd

Check service:

systemctl status glusterd



```
admin_filesvr1@filesvr2:~$ sudo systemctl enable --now glusterd
Created symlink /etc/systemd/system/multi-user.target.wants/glus
admin_filesvr1@filesvr2:~$ systemctl status glusterd
● glusterd.service - GlusterFS, a clustered file-system server
     Loaded: loaded (/usr/lib/systemd/system/glusterd.service; e
     Active: active (running) since Wed 2025-09-24 09:34:19 UTC;
       Docs: man:glusterd(8)
    Process: 3316 ExecStart=/usr/sbin/glusterd -p /var/run/glust
   Main PID: 3317 (glusterd)
      Tasks: 9 (limit: 4548)
     Memory: 7.3M (peak: 17.1M)
        CPU: 998ms
     CGroup: /system.slice/glusterd.service
             └─3317 /usr/sbin/glusterd -p /var/run/glusterd.pid
```

Image 6.3: Show that glusterd service is running

Step 3: Form a Trusted Storage Pool

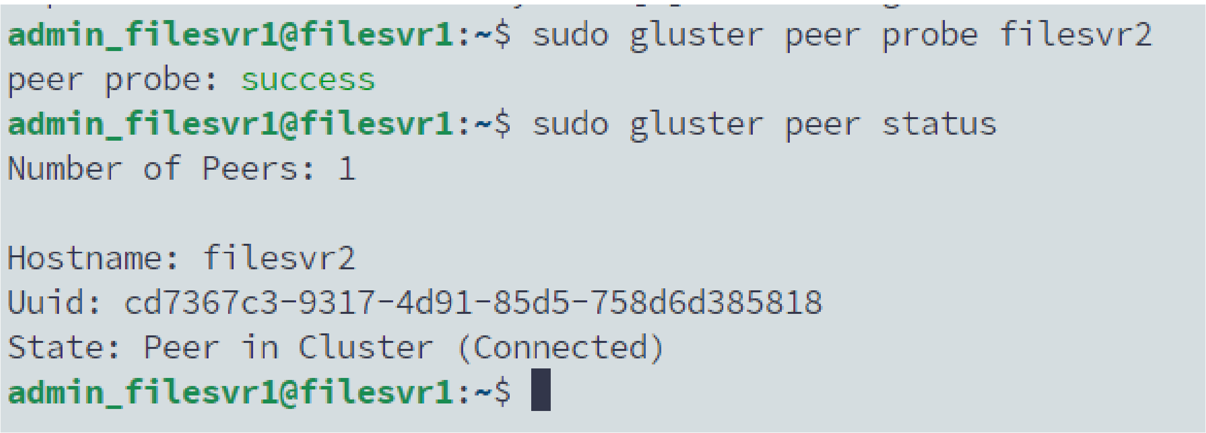
From **filesvr1**, probe filesvr2:

sudo gluster peer probe filesvr2

Verify:

sudo gluster peer status

You should see gluster2 as a peer.



```
admin_filesvr1@filesvr1:~$ sudo gluster peer probe filesvr2
peer probe: success
admin_filesvr1@filesvr1:~$ sudo gluster peer status
Number of Peers: 1

Hostname: filesvr2
Uuid: cd7367c3-9317-4d91-85d5-758d6d385818
State: Peer in Cluster (Connected)
admin_filesvr1@filesvr1:~$
```

Image 6.4: Storage pool includes peers

Step 4: Create a Brick (Storage Path)

On each node, prepare a directory:

sudo mkdir -p /data/gluster/brick1

Step 5: Create a Replicated Volume

Run this on **gluster1**:

sudo gluster volume create gv0 replica 2 \

gluster1:/data/gluster/brick1 \

gluster2:/data/gluster/brick1 force

- gv0 = volume name.

- replica 2 = replicate across 2 nodes.

Start the volume:

```
sudo gluster volume start gv0
```

Step 6: Mount the Volume

On a client (or one of the nodes), install:

```
sudo apt install glusterfs-client -y
```

Mount the volume:

```
sudo mkdir -p /mnt/glusterfs
sudo mount -t glusterfs gluster1:/gv0 /mnt/glusterfs
```

Make it persistent by adding to /etc/fstab:

```
filesvr1:/gv0 /mnt/glusterfs glusterfs defaults,_netdev 0 0
```

Step 7: Test Replication

- Create a file on /mnt/glusterfs from one server:

  ```
  echo "Hello GlusterFS" | sudo tee /mnt/glusterfs/testfile.txt
  ```

- Check on the other node — the file should be replicated automatically.

6.3 Mapping network drive on Windows Client:

6.3.1 Export GlusterFS Volume over SMB for Windows:

1. Install Samba on Gluster node(s)

On each GlusterFS server:

```
sudo apt update
sudo apt install samba
```

2) Configure Samba share (no vfs objects = glusterfs)

Edit /etc/samba/smb.conf and add:

```
[gv0]

  comment = Gluster volume gv0

  path = /mnt/glusterfs

  read only = no

  browseable = yes

  guest ok = no

  create mask = 0664

  directory mask = 0775

  force user = root
```

(Adjust permissions/ownership to your needs; often you'll chown -R someuser:somegroup /srv/gluster/gv0 and set force user = someuser.)

Reload Samba:

```
sudo systemctl restart smbd

sudo systemctl status smbd --no-pager
```

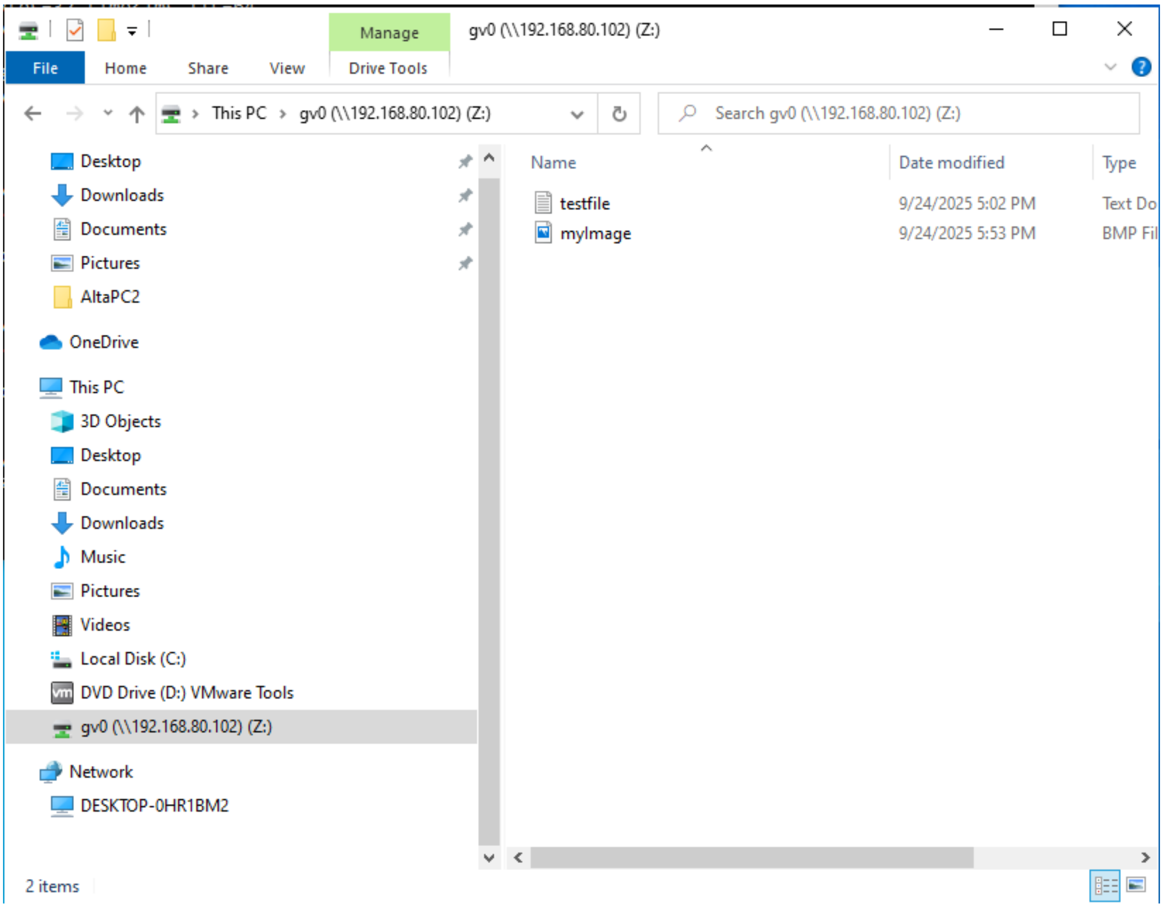
4) Create a Samba user (if not using guest)

```
sudo smbpasswd -a yourusername
```

5) From Windows, map the share

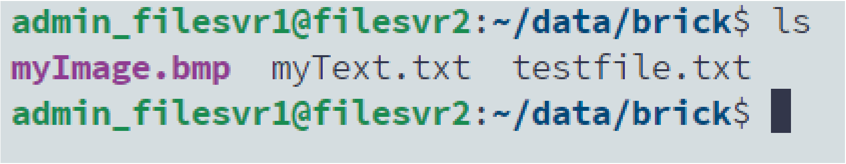
- File Explorer: \\filesvr1\gv0
- Or PowerShell/CMD:

```
net use Z: \\filesvr1\gv0 /user:yourusername
```

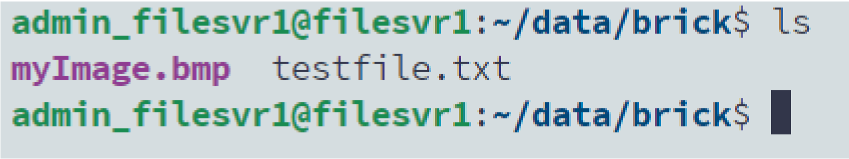


Name	Date modified	Type
testfile	9/24/2025 5:02 PM	Text Do
myImage	9/24/2025 5:53 PM	BMP Fil

Image 6.5 Volume gv0 of Ubuntu filesvr1 was mapped to Windows 10 drive Z



```
admin_filesvr1@filesvr2:~/data/brick$ ls
myImage.bmp  myText.txt  testfile.txt
admin_filesvr1@filesvr2:~/data/brick$ 
```



```
admin_filesvr1@filesvr1:~/data/brick$ ls
myImage.bmp  testfile.txt
admin_filesvr1@filesvr1:~/data/brick$ 
```

Image 6.6 myImage.bmp file was distributed to both filesvr2 and filesvr1