

# comment-installer-glusterfs-sur-debian-12

GlusterFS or Gluster File System, is a free and open-source distributed file system developed by RedHat. GlusterFS is a scalable file system formed from several servers into one entity file system that allows users to connect and mount the GlusterFS volume.

GlusterFS is a modern file system that can handle petabytes of data. Also, it's easy to install and maintain, and also easy to scale the file system.

In the following guide, I'll take you through the installation of GlusterFS on Debian 12 servers. We will create a high-available and replicated storage system with GlusterFS using multiple Debian servers. You will also learn how to mount GlusterFS to a Debian client machine and set up auto-mount of GlusterFS volume via `/etc/fstab` file.

## Prerequisites

Before you proceed with this guide, confirm that you have the following:

- Multiple Debian 12 servers - This example uses 3 Debian servers for creating the GlusterFS Cluster.
- A non-root user with administrator privileges.
- A Debian client machine.

## Partitioning Disk with Parted

Before you get started, you will create a new partition that will be used for GlusterFS on each server. In this example, you will create a partition from disk `/dev/sdb` with a Parted partition manager.

Install parted to your Debian servers using the following apt command.

```
sudo apt install parted -y
```

```
root@server1:~#  
root@server1:~# sudo apt install parted -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  libparted2  
Suggested packages:  
  libparted-dev libparted-i18n parted-doc  
The following NEW packages will be installed:  
  libparted2 parted  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 333 kB of archives.  
After this operation, 662 kB of additional disk space will be used.  
Get:1 http://httpredir.debian.org/debian bookworm/main amd64 libparted2 amd64 3.5-3 [294 kB]  
81% [Working]
```

Execute the following parted command to start partitioning the `/dev/sdb` disk.

```
sudo parted /dev/sdb
```

After executing the parted command, you should be in the parted environment.

Run the following command to create a new partition table for disk `/dev/sdb` to `msdos`.

```
mklabel msdos
```

Now run the command below to create a new primary partition `/dev/sdb1` with format `ext4` and size `5 GB`.

```
mkpart primary ext4 1MB 5369MB
```

Once the partition is created, type `quit` to exit from Parted.

Next, run the command below to format the `/dev/sdb1` partition to `ext4`.

```
sudo mkfs -t ext4 /dev/sdb1
```

After the `/dev/sdb1` is formatted, the following output will be shown:

```
root@server1:~#
root@server1:~# sudo parted /dev/sdb
GNU Parted 3.5
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
(parted) mklabel msdos

(parted)
(parted) mkpart primary ext4 1MB 5369MB
(parted)
(parted) quit
Information: You may need to update /etc/fstab.

root@server1:~# sudo mkfs -t ext4 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 1310464 4k blocks and 327680 inodes
Filesystem UUID: 8a4d9f63-5599-4a14-8178-f3911d8e0e1c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@server1:~#
```

Now run the following command to create a new target mount directory `/gluster`. Then, mount the `/dev/sdb1` partition to the `/gluster` directory.

```
mkdir -p /gluster
sudo mount /dev/sdb1 /gluster
```

Lastly, create a new directory `/gluster/brick0` that will be used as the data directory for each GlusterFS server.

```
sudo mkdir -p /gluster/brick0
```

## Installing GlusterFS Server

After you've partitioned your disk on each server, you will add the GlusterFS repository and install the `glusterfs-server` package to each server.

Execute the following command to add the GPG key of the GlusterFS repository.

```
curl https://download.gluster.org/pub/gluster/glusterfs/11/rsa.pub | gpg --dearmor > /usr/share/keyrings/glusterfs-archive-keyring.gpg
```

Then, run the command below to add the GlusterFS repository to your Debian servers.

```
DEBID=$(grep 'VERSION ID=' /etc/os-release | cut -d '=' -f 2 | tr -d '"')
DEBVER=$(grep 'VERSION=' /etc/os-release | grep -Eo '[a-z]+')
DEBARCH=$(dpkg --print-architecture)
```

```
echo "deb [signed-by=/usr/share/keyrings/glusterfs-archive-keyring.gpg]
https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/${DEBID}/${DEBARCH}/apt ${DEBVER} main" | sudo tee
/etc/apt/sources.list.d/gluster.list
```

```

root@server1:~#
root@server1:~# curl https://download.gluster.org/pub/gluster/glusterfs/11/rsa.pub | gpg --dearmor > /usr/share/keyrings/glusterfs-arc
hive-keyring.gpg
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 1713 100 1713 0 0 1330 0 0:00:01 0:00:01 --:--:-- 1331
root@server1:~#
root@server1:~# DEBID=$(grep 'VERSION_ID=' /etc/os-release | cut -d '=' -f 2 | tr -d "'")
DEBVER=$(grep 'VERSION=' /etc/os-release | grep -Eo '[a-z]+')
DEBARCH=$(dpkg --print-architecture)
root@server1:~#
root@server1:~# echo "deb [signed-by=/usr/share/keyrings/glusterfs-archive-keyring.gpg] https://download.gluster.org/pub/gluster/glust
erfs/LATEST/Debian/${DEBID}/${DEBARCH}/apt ${DEBVER} main" | sudo tee /etc/apt/sources.list.d/gluster.list
deb [signed-by=/usr/share/keyrings/glusterfs-archive-keyring.gpg] https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/12/
amd64/apt bookworm main
root@server1:~#

```

After adding the GlusterFS repository and GPG key, execute the following apt command to update and refresh your Debian package index.

```
sudo apt update
```

```

root@server1:~#
root@server1:~# sudo apt update
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://httpredir.debian.org/debian bookworm InRelease
Hit:3 http://httpredir.debian.org/debian bookworm-updates InRelease
Get:4 https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/12/amd64/apt bookworm InRelease [2,103 B]
Get:5 https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/12/amd64/apt bookworm/main amd64 Packages [5,653 B]
Fetched 7,756 B in 2s (4,078 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

```

Then execute the apt install command below to install the glusterfs-server package.

```
sudo apt install glusterfs-server
```

Input y to proceed with the installation.

```

root@server1:~#
root@server1:~# sudo apt install glusterfs-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  attr fuse3 glusterfs-client glusterfs-common libfuse3-3 libgfapi0 libgfchange
  libgoogle-perftools4 libinih1 libtcmalloc-minimal4 libunwind8 liburcu8 liburi
  python3-jwt python3-prettytable python3-wcwidth xfsprogs
Suggested packages:
  python-cryptography-doc python3-cryptography-vectors python3-crypto xfsdump a
The following packages will be REMOVED:
  fuse
The following NEW packages will be installed:
  attr fuse3 glusterfs-client glusterfs-common glusterfs-server libfuse3-3 libg
  libglusterfs0 libgoogle-perftools4 libinih1 libtcmalloc-minimal4 libunwind8 l
  python3-cryptography python3-jwt python3-prettytable python3-wcwidth xfsprogs
0 upgraded, 24 newly installed, 1 to remove and 34 not upgraded.
Need to get 31.1 MB of archives.
After this operation, 52.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

Once the glusterfs-server is installed, run the systemctl command below to start and enable the glusterd service.

```
sudo systemctl start glusterd
sudo systemctl enable glusterd
```

Lastly, verify the glusterd service to ensure that the service is running and enabled.

```
sudo systemctl status glusterd
```

The following output reveals that the glusterd service is running and enabled.

```

root@server1:~#
root@server1:~# sudo systemctl start glusterd
root@server1:~# sudo systemctl enable glusterd
Created symlink /etc/systemd/system/multi-user.target.wants/glusterd.service → /lib/systemd/
root@server1:~#
root@server1:~# sudo systemctl status glusterd
• glusterd.service - GlusterFS, a clustered file-system server
  Loaded: loaded (/lib/systemd/system/glusterd.service; enabled; preset: enabled)
  Active: active (running) since
  Docs: man:glusterd(8)
  Main PID: 6897 (glusterd)
  Tasks: 24 (limit: 2304)
  Memory: 20.6M
  CPU: 2.299s

```

## Initializing GlusterFS Cluster

With the glusterfs-server package installed, the next step is to initialize the GlusterFS Cluster. In this example, you will initialize the GLusterFS Cluster from server1, then you will add both server2 and server3 to the cluster.

From the server1 terminal, execute the gluster command below to initialize the GlusterFS Cluster with members of **server2** on IP address **192.168.10.21** and **server3** with IP address **192.168.10.22**.

```

sudo gluster peer probe 192.168.10.21
sudo gluster peer probe 192.168.10.22

```

If the initialization is successful, the output message "**peer probe: success**" will be displayed.

```

root@server1:~#
root@server1:~# sudo gluster peer probe 192.168.10.21
peer probe: success
root@server1:~#
root@server1:~# sudo gluster peer probe 192.168.10.22
peer probe: success
root@server1:~#

```

Now move to **server2** or **server3** and run the gluster command below to verify the GlusterFS Cluster status.

```

sudo gluster peer status

```

If everything goes well, two peers will be available on your GlusterFS Cluster.

```

root@server1:~#
root@server1:~# sudo gluster peer status
Number of Peers: 2

Hostname: 192.168.10.21
Uuid: aa667cbb-fd83-44c4-ae4c-5da17e4198ce
State: Peer in Cluster (Connected)

Hostname: 192.168.10.22
Uuid: 831fcc37-139c-42f9-8a13-76bb0371eb0d
State: Peer in Cluster (Connected)
root@server1:~#

```

On **server2** - you should see the **server1** and **server3**.

```

root@server2:~#
root@server2:~# sudo gluster peer status
Number of Peers: 2

Hostname: 192.168.10.20
Uuid: d9a0bf16-ad6f-409c-bae3-d45188812331
State: Peer in Cluster (Connected)

Hostname: 192.168.10.22
Uuid: 831fcc37-139c-42f9-8a13-76bb0371eb0d
State: Peer in Cluster (Connected)
root@server2:~#

```

On **server3**, you should see the **server1** and **server2**.

```

root@server3:~#
root@server3:~# sudo gluster peer status
Number of Peers: 2

Hostname: 192.168.10.20
Uuid: d9a0bf16-ad6f-409c-bae3-d45188812331
State: Peer in Cluster (Connected)

Hostname: 192.168.10.21
Uuid: aa667cbb-fd83-44c4-aefc-5da17e4198ce
State: Peer in Cluster (Connected)
root@server3:~#

```

Lastly, you can also verify the list of available peers on your GlusterFS Cluster using the gluster command below.

```
sudo gluster pool list
```

The following output shows that the GlusterFS Cluster has three different servers.

```

root@server1:~#
root@server1:~# sudo gluster pool list
UUID                               Hostname      State
aa667cbb-fd83-44c4-aefc-5da17e4198ce 192.168.10.21 Connected
831fcc37-139c-42f9-8a13-76bb0371eb0d 192.168.10.22 Connected
d9a0bf16-ad6f-409c-bae3-d45188812331 localhost     Connected
root@server1:~#
root@server1:~#

```

## Creating GlusterFS Volume

At this point, you've initialized the GlusterFS Cluster. To make the GlusterFS available for clients, you must create the volume on your GlusterFS cluster.

Execute the following command to create a new volume **volume1** with **3 replicas**, **server1**, **server2**, and **server3**.

```
sudo gluster volume create volume1 replica 3 192.168.10.20:/gluster/brick0 192.168.10.21:/gluster/brick0 192.168.10.22:/gluster/brick0
```

Once volume **volume1** is created, start it using the gluster command below.

```
sudo gluster volume start volume1
```

The output "**volume start: volume1: success**" indicates that the **volume1** is started.

```

root@server1:~#
root@server1:~# sudo gluster volume create volume1 replica 3 192.168.10.20:/gluster/brick0 192.168.10.21:/gluster/brick0 192.168.10.22:/gluster/brick0
volume create: volume1: success: please start the volume to access data
root@server1:~#
root@server1:~# sudo gluster volume start volume1
volume start: volume1: success
root@server1:~#

```

Now run the following command to check the list of available volumes on your GlusterFS Cluster.

```
sudo gluster volume info
```

The following output reveals that **volume1** is created with type **Replicate** and 3 servers, such as **server1**, **server2**, and **server3**.

```

root@server1:~#
root@server1:~# sudo gluster volume info

Volume Name: volume1
Type: Distributed-Replicate
Volume ID: 9c801727-d0e7-4310-a702-b06c97a28fa3
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 3 = 3
Transport-type: tcp
Bricks:
Brick1: 192.168.10.20:/gluster/brick0
Brick2: 192.168.10.21:/gluster/brick0
Brick3: 192.168.10.22:/gluster/brick0
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
root@server1:~#

```

## Mounting GlusterFS Volume

In the following section, you will mount the GlusterFS volume **volume1** that you've created to a Debian Client machine. To achieve that, you must add the GlusterFS repository and install the `glusterfs-client` package to your Debian client machine.

Download the GPG key for the GLusterFS repository to your Debian client machine.

```
curl https://download.gluster.org/pub/gluster/glusterfs/11/rsa.pub | gpg --dearmor > /usr/share/keyrings/glusterfs-archive-keyring.gpg
```

Add the GLusterFS repository by executing the following command on your client.

```
DEBID=$(grep 'VERSION ID=' /etc/os-release | cut -d '=' -f 2 | tr -d "'")
DEBVER=$(grep 'VERSION=' /etc/os-release | grep -Eo '[a-z]+')
DEBARCH=$(dpkg --print-architecture)
```

```
echo "deb [signed-by=/usr/share/keyrings/glusterfs-archive-keyring.gpg]
https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/${DEBID}/${DEBARCH}/apt ${DEBVER} main" | sudo tee
/etc/apt/sources.list.d/gluster.list
```

Now refresh your Debian client repository and install the `glusterfs-client` package via the following apt command.

```
sudo apt update
sudo apt install glusterfs-client
```

Type `y` to proceed with the installation. Once the `glusterfs-client` is installed, you can now mount the GlusterFS volume to your Debian client machine.

```

root@client:~#
root@client:~# sudo apt install glusterfs-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
glusterfs-client is already the newest version (11.0-2).
glusterfs-client set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 34 not upgraded.
root@client:~#
root@client:~#

```

Next, execute the following command to create a new target mount directory `/mnt/data`.

```
mkdir -p /mnt/data
```

Then, run the following command to mount the `volume1` to the `/mnt/data` directory. You can change the GlusterFS server IP address with **server1**, **server2**, or **server3**.

```
sudo mount.glusterfs 192.168.10.20:/volume1 /mnt/data
```

Once volume1 is mounted, verify it using the following command.

```
sudo df -h
```

If everything goes well, you should see that volume1 is mounted to the `/mnt/data` directory.

```
root@client:~#  
root@client:~# mkdir -p /mnt/data  
root@client:~#  
root@client:~# sudo mount.glusterfs 192.168.10.20:/volume1 /mnt/data  
root@client:~#  
root@client:~# sudo df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	961M	0	961M	0%	/dev
tmpfs	197M	544K	197M	1%	/run
/dev/mapper/debian--12--vg-root	62G	1.4G	57G	3%	/
tmpfs	984M	0	984M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda1	455M	58M	373M	14%	/boot
tmpfs	197M	0	197M	0%	/run/user/1000
192.168.10.20:/volume1	62G	2.0G	57G	4%	/mnt/data

```
root@client:~#
```

Now that the GlusterFS volume is mounted, you will verify the write access of the target mount directory and verify the data replication from the client machine to servers on the GlusterFS Cluster.

Next, move to the `/mnt/data` directory and create new files using the following command. This will create multiple files in that directory and ensure that the volume is writable.

```
cd /mnt/data  
touch file{1..15}.md
```

Check the list of available files using the `ls` command below.

```
ls -ah
```

```
root@client:~#  
root@client:~# cd /mnt/data  
root@client:/mnt/data#  
root@client:/mnt/data# touch file{1..15}.md  
root@client:/mnt/data#  
root@client:/mnt/data# ls  
file10.md file12.md file14.md file1.md file3.md file5.md file7.md file9.md  
file11.md file13.md file15.md file2.md file4.md file6.md file8.md  
root@client:/mnt/data#  
root@client:/mnt/data#
```

Now move to the GlusterFS servers, **server1**, **server2**, or **server3**, and run the `ls` command below to check the list of available files.

```
ls /gluster/brick0
```

When replication is working, you should see files on all those GlusterFS servers created from the Debian client machine.

Below data is replicated to **server1**.

```
root@server1:~#  
root@server1:~# ls /gluster/brick0  
file10.md file12.md file14.md file1.md file3.md file5.md file7.md file9.md  
file11.md file13.md file15.md file2.md file4.md file6.md file8.md  
root@server1:~#  
root@server1:~#
```

Below data is replicated to **server2**

```
root@server2:~#  
root@server2:~# ls /gluster/brick0  
file10.md file12.md file14.md file1.md file3.md file5.md file7.md file9.md  
file11.md file13.md file15.md file2.md file4.md file6.md file8.md  
root@server2:~#  
root@server2:~#
```

Below data replicated to **server3**.

```
root@server3:~#  
root@server3:~# ls /gluster/brick0  
file10.md file12.md file14.md file1.md file3.md file5.md file7.md file9.md  
file11.md file13.md file15.md file2.md file4.md file6.md file8.md  
root@server3:~#  
root@server3:~#
```

## Setting Up Auto-Mount GlusterFS Volume

In the following section, you will learn how to set up auto-mount of GlusterFS volume via `/etc/fstab` file. This allows you to mount the GlusterFS volume automatically at boot.

Open the `/etc/fstab` file using the following nano editor command.

```
sudo nano /etc/fstab
```

Insert the following configuration to mount the volume1 to the target directory `/mnt/data`.

```
192.168.10.20:/volume1 /mnt/data glusterfs defaults,_netdev 0 0
```

Save and exit the file when finished.

Now run the following command to reload systemd manager and verify the `/etc/fstab` configuration. When no output message, it confirms that you've proper `/etc/fstab` configuration.

```
sudo systemctl daemon-reload  
sudo mount -a
```

## Conclusion

Great job! You've successfully installed the GlusterFS on Debian 12 servers and created the GlusterFS Cluster with 3 servers. You've also created and mounted a GlusterFS volume to the Debian client machine. Furthermore, you've also configured the auto-mount of GlusterFS volume using the `/etc/fstab` file.

---