Tutorial: Deploy a Highly Availability GlusterFS Storage Cluster

Link: https://thenewstack.io/tutorial-deploy-a-highly-availability-glusterfs-storage-cluster/

The GlusterFS network file system is perfectly suited for various use cases that require the handling of large amounts (think petabytes) of stored data.

Nov 6th, 2020 10:32am by Jack Wallen
Featued image for: Tutorial: Deploy a Highly Availability GlusterFS Storage Cluster

The GlusterFS network file system is perfectly suited for various use cases that require the handling of large amounts (think petabytes) of stored data. In other words, this could be the ideal storage system for your various cloud or container deployments. With features like sharding, tiering, AFR Statistics, file snapshots, distributed hash tables, nonuniform file access, OVirt and QEMU integration, RDMA connection manager, rebalance, server quorum, distributed geo-replication, and brick failure detection, this file system might be ideal for your needs. Red Hat currently manages this open source network file system.

Of course, how you use GlusterFS with your cloud implementation will depend on which cloud platform you are using. But before you can roll it into your system, you first must get this networkable storage up and running.

I'm going to walk you through the process of deploying a three-node GlusterFS cluster on Ubuntu Server 20.04. To make this work you'll need three instances of Ubuntu. For my purposes those will have the following hostnames and IP addresses:

```
192.168.1.24 gluster1
```

```
192.168.1.25 gluster2
```

```
192.168.1.26 gluster3
```

You will want to change the IP addresses to match your network topography.

TRENDING STORIES

1. The Architect’s Guide to Storage for AI
2. The Architect’s Guide: A Modern Data Lake Reference Architecture
3. How to Work with Containers in TrueNAS
4. How to Create an Object Storage Bucket with MinIO Object Storage
5. Enable End-to-End Encryption Between Nextcloud and Your Desktop Client

First Steps

After spinning up your instances of Ubuntu, the first thing you want to do is update and upgrade each. You can do that (on all three) with the following two commands:

```
sudo apt-get update
```

```
sudo apt-get upgrade -y
```

If the kernel gets upgraded in any of these instances, you’ll want to make sure to reboot the server (so the updates get applied).

After you’ve upgraded, you’ll then want to set the hostname for each. This can be done with a handy command like so:

```
sudo hostnamectl set-hostname NAME
```

Where NAME will be gluster1, gluster2, and gluster3.

Next, we need to map the addresses in the /etc/hosts file. Open that file (on each server) for editing with the command:

```
sudo nano /etc/hosts
```

Map those addresses by adding the following at the bottom of the file:

```
1
2
3
```

```
192.168.1.24 gluster1
192.168.1.25 gluster2
192.168.1.26 gluster3
```

Save and close the file.

Installing GlusterFS

With the release of Ubuntu Server 20.04, GlusterFS is now found in the standard repositories. So to install the software, go back to the terminal window and issue the command:

```
sudo apt-get install glusterfs-server -y
```

Make sure to install GlusterFS on gluster1 and gluster2.

After the installation completes, start and enable GlusterFS on each server with the following two commands:

```
sudo systemctl start glusterd
```

```
sudo systemctl enable glusterd
```

Configuring GlusterFS

Now that your servers are ready and GlusterFS is installed, it’s time to configure gluster. On gluster1, create a trusted pool with the command:

```
sudo gluster peer probe gluster2
```

You should see peer probe: success returned.

Verify the status of the two peers with the command:

```
sudo gluster peer status
```

You should see that gluster2 is connected (Figure 1).

Figure 1: Our gluster1 and gluster2 servers are connected.

Zoom

Figure 1: Our gluster1 and gluster2 servers are connected.

Creating a Distributed Volume

We'll next create a distributed volume. I would highly recommend you create this volume on a partition that isn't within the system directory (aka, not on the same drive that your OS is hosted on). If you create this volume on the same drive as the OS, you could run into sync errors.

Let's create a new directory for GlusterFS (on both gluster1 and gluster2) with the command:

```
sudo mkdir -p /glusterfs/distributed
```

With the directory created, we can now create the volume (named v01) that will replicate on both gluster1 and gluster2. The command for this is:

```
sudo gluster volume create v01 replica 2 transport tcp gluster1:/glusterfs/distributed gluster2:/glusterfs/distributed
```

You will be prompted to okay the creation. Type "y" to allow the creation of the new distributed volume. Once that succeeds, start the volume with the command:

```
sudo gluster volume start v01
```

You can verify the creation with the command:

```
sudo gluster volume info v01
```

Installing the GlusterFS Client and Connecting to the Distributed Volume

It's now time to install the GlusterFS client. We'll do this on gluster3. For this, issue the command:

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

Create a new mount point for GlusterFS on gluster3 with the command:

```
sudo mkdir -p /mnt/glusterfs
```

We can now mount the distributed file system with the command:

```
sudo mount -t glusterfs gluster1:/v01 /mnt/glusterfs/
```

Finally, you'll want to make sure the distributed file system is mounted at boot. To do this, you'll need to edit the fstab file with the command:

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

At the bottom of that file, add the following:

```
gluster1:/v01 /mnt/glusterfs glusterfs defaults,_netdev 0 0
```

Testing the Filesystem

With all of this in place, we can now test the GlusterFS distributed file system. On gluster1 issue the command:

```
sudo mount -t glusterfs gluster1:/v01 /mnt
```

On gluster2 issue the command:

```
sudo mount -t glusterfs gluster2:/v01 /mnt
```

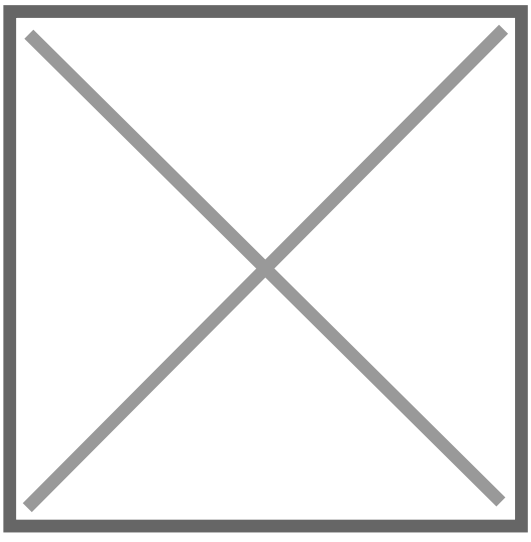
Move over to gluster3 and create a test file with the command:

```
sudo touch /mnt/glusterfs/thenewstack
```

Check to make sure the new file appears on both gluster1 and gluster2 with the command (run on gluster1 and gluster2):

```
ls /mnt
```

You should see thenewstack appear in both directories on gluster1 and gluster2 (Figure 2).



Zoom

Figure 2: The test file has appeared on gluster1.

And there you go, you now have a GlusterFS distributed file system up and running. You should now be able to integrate this into anything that requires a high-volume file system that offers plenty of features that can satisfy many of your cloud and container needs.

More Storage Tutorials

Tutorial: Set up Cloud Storage on a Linux Server, Using Seafile

Tutorial: Create a Docker Swarm with Persistent Storage Using GlusterFS

Tutorial: Dynamic Provisioning of Persistent Storage in Kubernetes with MiniKube

---

Revision #1
Created 2024-07-07 17:32:40 UTC by Administrador
Updated 2024-07-07 17:34:39 UTC by Administrador