Lab 10: Fault Tolerance

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of some of its components. Fault tolerance is especially demanded in systems with high availability or in critical conditions.

Spare components address the fundamental characteristic of fault tolerance in three ways:
- Replication
- Redundancy
- Diversity

In this Lab, you will know how to set up a high availability storage server with GlusterFS[1] on Ubuntu 18.04 LTS (Bionic Beaver). You will use 3 ubuntu servers, 1 server as a client, and 2 others as a storage. Each storage server will be a mirror of the other, and files will be replicated across both storage servers.

Prerequisites
- 3 Ubuntu 18.04 Servers
- 10.0.15.10 - gfs01
- 10.0.15.11 - gfs02
- 10.0.15.100 - client01
- Root Privileges

Exercises
1. Install and configure servers on AWS
2. GlusterFS Pre-Installation
3. Install GlusterFS Server
4. Configure GlusterFS Servers
5. Setup GlusterFS Client
6. Testing Replicate/Mirroring

Exercise 1. Install and configure servers on AWS
Before launching the servers let’s configure the network for them.

Exercise 1.1 VPC configuration
Amazon’s Virtual Private Cloud (VPC) service provides the networking layer of EC2.
1. Login to your AWS account and go to the Your VPC tab in the VPC Dashboard
2. Create a VPC with the IPv4 CIDR block **10.0.0.0/16** and with the Name tag **GlusterFS Network**.
3. Enable DNS hostnames to resolve the instance name. Select GlusterFS Network and click Actions. Choose Edit DNS hostnames. Select **enable** item and click the Save button.
4. Create a subnet in the *GlusterFS Network* you created earlier. Choose any availability zone, assign the IPv4 CIDR block **10.0.15.0/24** and with the Name tag **GlusterFS subnet-1**.
5. To enable Internet access for your instances, you must create a default route pointing to the Internet gateway. Create an Internet gateway and attach it to the GlusterFS Network:

[1] GlusterFS is an open source and distributed file system that sets disk storage resources from multiple servers into a single namespace. It's suitable for data-intensive tasks such as cloud storage and data media streaming.

   a. In the VPC Dashboard, click Internet Gateways.
   b. Click the Create Internet Gateway button.
   c. Specify the Name tag as **GlusterFS internet gateway** and click Create.
   d. Select the Internet gateway you just created, and under the Actions menu, click Attach To VPC.
   e. Select the VPC GlusterFS Network and then click Attach.
   f. To create a default route in the main route table, in the VPC Dashboard, click Route Tables.
   g. Select the route table for the GlusterFS Network.
   h. Click the Routes tab, and click Edit.
   i. Click the Add route button.
   j. 10. For the destination, enter 0.0.0.0/0. For the target, enter the identifier of the GlusterFS internet gateway.
   k. Click Save routes.
6. Create a new Security group to connect to our servers using ssh and to check their availability using ping:
   a. In the VPC Dashboard, click Security Groups
   b. Click the Create security group button.
   c. Specify the Security group name as **gfs-security-group** and the Description as **Security group for GlusterFS network**.
   d. Select the VPC GlusterFS Network
   e. Click Create and then Close.
   f. Select the security group you just created.
   g. Click the Inbound Rules tab, and click Edit rules.
   h. Click the Add Rule button.
   i. For the Type, select SSH. For the Source, select Anywhere.
   j. Click the Add Rule button.
   k. For the Type, select All traffic. For the Source, leave Custom and enter the network's IP address **10.0.15.0/24**.
   l. Click Save rules

Exercise 1.2 Servers launch

1. From the EC2 Dashboard, click to launch a new instance and select a **Ubuntu Server 18.04 LTS** and instance type **t2.micro**.
2. On the Configure Instance page choose VPC *GlusterFS Network* in the Network filed, **enable** Auto-assign Public IP, and assign primary IP **10.0.15.10** in Network interfaces tab.
3. On the Add Storage page click Add New Volume. Select **/dev/sdf** in the Device, type **1** in the Size (GiB), and check the box Delete on Termination.
4. On the Add Tags page add the key **Name** with the **gfs01** value.
5. On the Configure Security Group page, Select an existing security group you created before **gfs-security-group**.
6. Launch the instance.
7. Before letting you launch the instance, AWS will require you to select—or create—a key pair. Follow the instructions.
8. Once the instance is launched, you can return to the Instances Dashboard to wait until everything is running properly.
9. Click the Connect button and follow the instructions to connect to the instance from your local machine using ssh.
10. Check the instance's IP address:

```
$ hostname -I
10.0.15.10
```

11. Change the instance's hostname:

```
$ sudo hostnamectl set-hostname gfs01
```

12. Reboot the system to changes take effect:

```
$ sudo reboot
```

13. Connect to the instance using *ssh* and verify the name has been changed:

```
$ hostnamectl
Static hostname: gfs01
...
```

14. Repeat exercise 1.2 for the remaining two servers, **gfs02** and **client01**, with the primary IP address configuration described in the prerequisites.
15. Check the servers' availability from the client machine with `ping` command:

```
$ ping -c3 10.0.15.10
PING 10.0.15.10 (10.0.15.10) 56(84) bytes of data.
64 bytes from 10.0.15.10: icmp_seq=1 ttl=64 time=0.421 ms
64 bytes from 10.0.15.10: icmp_seq=2 ttl=64 time=0.396 ms
64 bytes from 10.0.15.10: icmp_seq=3 ttl=64 time=0.439 ms

--- 10.0.15.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss,time 3011ms
rtt min/avg/max/mdev = 0.396/0.435/0.485/0.035 ms

$ ping -c3 10.0.15.11
PING 10.0.15.11 (10.0.15.11) 56(84) bytes of data.
64 bytes from 10.0.15.11: icmp_seq=1 ttl=64 time=0.773 ms
64 bytes from 10.0.15.11: icmp_seq=2 ttl=64 time=0.451 ms
64 bytes from 10.0.15.11: icmp_seq=3 ttl=64 time=0.472 ms

--- 10.0.15.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss,time 3063ms
rtt min/avg/max/mdev = 0.451/0.550/0.773/0.132 ms
```

Exercise 2. GlusterFS Pre-Installation

The first step we need to do before installing glusterfs on all servers is configuring the hosts' file and add GlusterFS repository to each server.

1. Configure Hosts File:

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

2. Paste hosts configuration below.

```
10.0.15.10 gfs01
10.0.15.11 gfs02
10.0.15.100 client01
```

3. Save and exit.

4. Now ping each server using the hostname as below.

   ```
   $ ping -c 3 gfs01
   $ ping -c 3 gfs02
   $ ping -c 3 client01
   ```

   Each hostname will resolve to each server's IP address.
5. Add GlusterFS Repository
   Install the software-properties-common package to the system.

   ```
   $ sudo apt install software-properties-common -y
   ```

6. Add the glusterfs repository by running commands below.

   ```
   $ sudo add-apt-repository ppa:gluster/glusterfs-7
   ```

7. The command will update all repositories. And we've already added the glusterfs repository to all systems.

Exercise 3. Install GlusterFS Server

In this step, we will install the glusterfs server on **gfs01** and **gfs02** servers.

1. Install glusterfs-server using the apt command.

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

2. Now start the *glusterd* service and enable it to launch every time at system boot.

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

   Glusterfs server is now up and running on the **gfs01** and **gfs02** servers.

3. Check the services and the installed software version.

   ```
   $ systemctl status glusterd
   $ glusterfsd --version
   ```

Exercise 4. Configure GlusterFS Servers

Let's consider some basic terms that we will use throughout this lab.

Distributed File System - A file system that allows multiple clients to access data which is spread across cluster peer. The servers allow the client to share and store data just like they are working on locally.

Glusterd - The GlusterFS daemon/service process that need to be run all members of trusted storage pool to manage volumes and cluster membership.

Cluster - a group of peer computer that works together.

Trusted Storage Pool - A storage pool is a trusted network of storage servers.

FUSE - File system in User space is a loadable kernel module that lets non-privileged users create files without editing kernel codes. FUSE module only provides a bridge to access to actual kernel interface.

Brick - basic unit of storage in GlusterFS . Represented by an export directory on a server in trusted storage pool.

Volume - logical collection of bricks.

In Figure 1 you can see an example of the GlusterFS architecture.

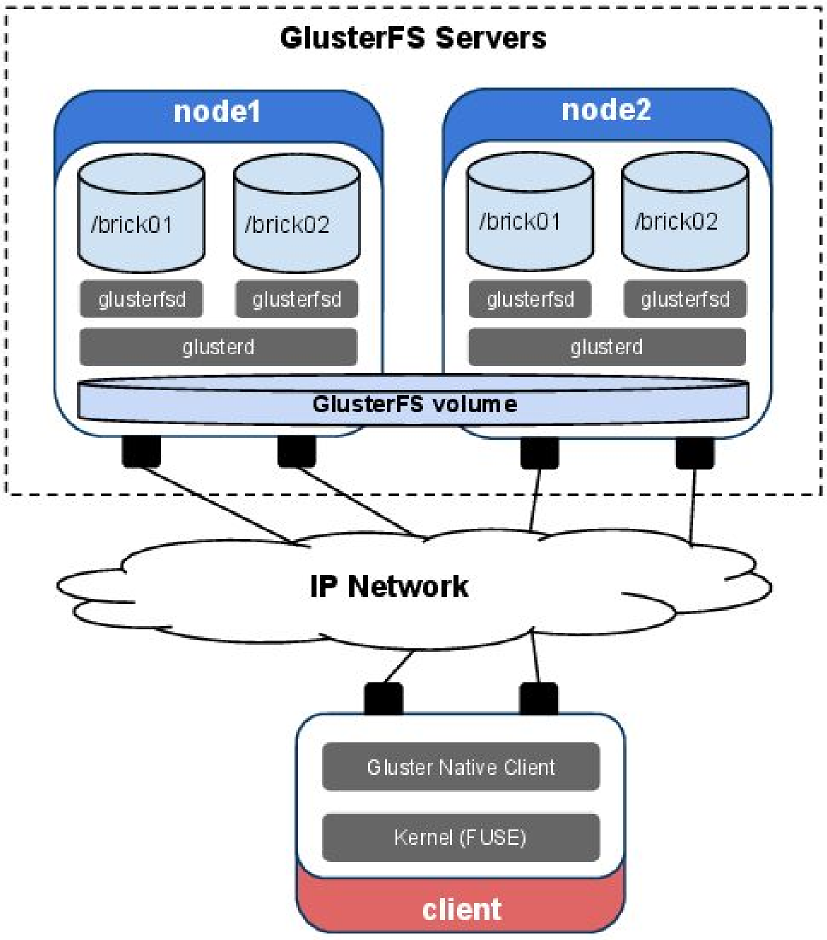

Figure 1. An example of the GlusterFS architecture

Glusterd services are now up and running, and the next step we will do is to configure those servers by creating a trusted storage pool and creating the distributed glusterfs volume.

1. Create a Trusted Storage Pool
   From the **gfs01** server, we need to add the **gfs02** server to the glusterfs storage pool. Run the command below.

   ```
   $ sudo gluster peer probe gfs02
   ```

   Now we will see the result '`peer probe: success`', and we've added the **gfs02** server to the storage trusted pool.

2. Check the storage pool status and list using commands below.

   ```
   $ sudo gluster peer status
   $ sudo gluster pool list
   ```

And you will see the **gfs02** server is connected to the peer cluster, and it's on the pool list.

3. Setup Distributed GlusterFS Volume
   After creating the trusted storage pool, we will create a new distributed glusterfs volume. For the server production, it's recommended to create the glusterfs volume using the different partition, not using a system directory.

   We will create the new glusterfs volume using the different partition.
   a. Check for a volume attached to **gfs01**:

      ```
      $ sudo fdisk -l
      ...
      Disk /dev/xvdf: 1 GiB, 1073741824 bytes, 2097152
      sectors
      ...
      ```

   b. Repeat the above step for **gfs02**

   c. Create a new partition with the XFS filesystem for the backend bricks and mount it.
      Perform this step on **gfs01** and **gfs02**:

      ```
      $ sudo fdisk /dev/xvdf

      Welcome to fdisk (util-linux 2.31.1).
      Changes will remain in memory only, until you decide to
      write them.
      Be careful before using the write command.

      Device does not contain a recognized partition table.
      Created a new DOS disklabel with disk identifier
      0xcbe7b411.

      Command (m for help): n
      Partition type
         p   primary (0 primary, 0 extended, 4 free)
         e   extended (container for logical partitions)
      Select (default p): p
      Partition number (1-4, default 1): 1
      First sector (2048-2097151, default 2048): press <Enter>
      Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151,
      default 2097151): press <Enter>

      Created a new partition 1 of type 'Linux' and of size
      1023 MiB.

      Command (m for help): w
      The partition table has been altered.
      Calling ioctl() to re-read partition table.
      Syncing disks.
      ```

The following examples assume that the brick will be residing on **/dev/xvdf1**.

```
$ sudo mkfs.xfs -i size=512 /dev/xvdf1
$ sudo mkdir -p /data/brick1
$  sudo  bash  -c  'echo  "/dev/xvdf1  /data/brick1  xfs
defaults 1 2" >> /etc/fstab'
$ sudo mount -a && mount
```

You should now see **xvdf1** mounted at **/data/brick1**

```
$ mount | grep xvdf1
/dev/xvdf1      on      /data/brick1      type      xfs
(rw,relatime,attr2,inode64,noquota)
```

d. Create a new directory `/data/brick1/vol01` on each **gfs01** and **gfs02** servers.

```
$ sudo mkdir -p /data/brick1/vol01
```

e. From the **gfs01** server, create the distributed glusterfs volume named **vol01** with 2 replicas **gfs01** and **gfs02**.

```
$ sudo gluster volume create vol01 replica 2 transport
tcp gfs01:/data/brick1/vol01 gfs02:/data/brick1/vol01
force

volume create: vol01: success: please start the volume
to access data
```

f. Now we've created the distributed volume **vol01** - start the **vol01** and check the volume info.

```
$ sudo gluster volume start vol01
volume start: vol01: success

$ sudo gluster volume info vol01

Volume Name: vol01
Type: Replicate
Volume ID: 09330619-0645-481c-a64d-2ac1e23cd274
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: gfs01:/data/brick1/vol01
Brick2: gfs02:/data/brick1/vol01
Options Reconfigured:
transport.address-family: inet
storage.fips-mode-rchecksum: on
nfs.disable: on
performance.client-io-threads: off
```

Take a screenshot of the output of the `sudo gluster volume info vol01` command and paste it into the report.

At this stage, we created the **vol01** volume with the type **Replicate** and 2 bricks on **gfs01** and **gfs02** server. All data will be distributed automatically to each replica server, and we're ready to mount the volume.

Exercise 5. Setup GlusterFS Client

In this step, we will mount the glusterfs volume **vol01** to the Ubuntu *client*, and we need to install the glusterfs-client to the client server.

1. Install glusterfs-client to the **client01** instance using the apt command.

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

2. Create a new directory `/mnt/glusterfs` when the glusterfs-client installation is complete.

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

3. Mount the distributed glusterfs volume **vol01** to the `/mnt/glusterfs` directory.

   ```
   $ sudo mount -t glusterfs gfs01:/vol01 /mnt/glusterfs
   ```

4. Now check the available volume on the system.

   ```
   $ df -h /mnt/glusterfs
   ```

5. And we will get the glusterfs volume mounted to the `/mnt/glusterfs` directory.
6. Mount glusterfs permanently to the Ubuntu client system:

   ```
   $ sudo bash -c 'echo "gfs01:/vol01 /mnt/glusterfs glusterfs
   defaults,_netdev 0 0" >> /etc/fstab'
   ```

7. Now we will get the glusterfs volume **vol01** mounted automatically through the fstab.
8. Change the owner of the directory so that the user has full access to resources:

   ```
   $ sudo chown -R $USER:$USER /mnt/glusterfs
   ```

9. Now reboot the server and when it's online, we will get the glusterfs volume 'vol01' mounted automatically through the fstab.
10.

Exercise 6. Testing

In this step, we will test the data mirroring on each server node.

1. Go to the `/mnt/glusterfs` directory on the **client01**

   ```
   cd /mnt/glusterfs
   ```

2. Create some files using touch command.

   ```
   $ touch file{01,02,03,04,05}
   ```

3. Test replication
   Now check on each - **gfs01** and **gfs02** - server, and we will get all the files that we've created from the client machine.

   ```
   $ ls /data/brick1/vol01/
   file01  file02  file03  file04  file05
   ```

   All files that we created from the client machine will be distributed to all the glusterfs volume node servers.

Take a screenshot of the output of the GlusterFS' directory content on each server and paste it into the report.

4. Stop the **gfs02** instance.
5. Create directory `dir1` and remove `file5` in the `/mnt/glusterfs` directory on the **client01**:

   ```
   $ mkdir dir1
   $ rm file5
   ```

6. Start the **gfs02** instance and check the content of the `/data/brick1/vol01/` directory on the **gfs02**:

   ```
   $ ls /data/brick1/vol01/
   dir1  file01  file02  file03  file04
   ```

Take a screenshot of the output of the GlusterFS' directory content on each server and paste it into the report.

Assignment

Suppose our storage server **gfs02** is down. The **gfs01** server is still able to store data and process requests from the client. But to ensure fault tolerance, we must provide redundancy for the core component. You need to add the new **gfs03** storage server to the existing replicated volume.
Hint: here is a link for help.
Take a screenshot of the output of the `sudo gluster volume info vol01` command after adding the new gfs03 server and paste it into the report.

In Moodle, submit the report in **pdf** format and include all the necessary screenshots.

Do not forget to clean up AWS test resources when you're done using them.