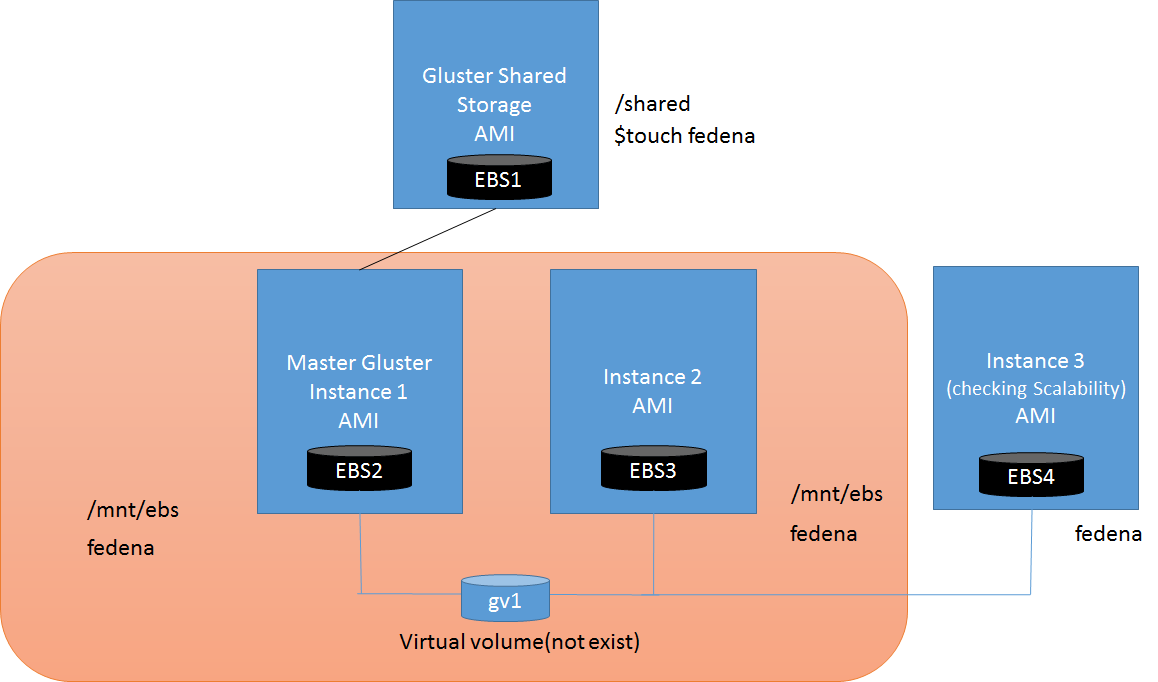
**Shared Memory on Multiple EC2 Instances via GlusterFS**

[Gluster](http://www.gluster.org/) is a high-performance shared file system that can be deployed in Amazon's EC2 cloud instances.

Following is the architecture which shows the example of using 2 instances with the shared memory and addition of 3rd instances (for the possibility in autoscaling)



The above architecture shows that There is one Shared Gluster storage which acts as shared memory. And two instance in which one instance acts as Master Gluster which connects with Shared Gluster storage. The Master Gluster Instance 1 and Instance 2 will create a virtual volume(gv1) whose data is shared with both of each other. If anything got written on the Shared Gluster Storage then automatically that change will reflect in gv1 means that data will reflect in both Master Gluster Instance 1 and Instance 2.

For the scalability purpose if we have Instance 3 then we have to connect it with the gv1 so that the same data will reflect on the Instance 3 also

**Steps to implement this approach is given below**

**STEP 1.**

**Create a Configured Gluster shared storage AMI**

1. Launch a t2.micro instance running any AMI (preferable RHEL7/CentOS7) and install Gluster on it with a 2GB EBS volume attached as /dev/sdb set to not delete on termination (the root volume on /dev/sda1 should be deleted on termination). Instead of just 2GB you can choose a larger size as appropriate for your needs (it should be big enough to hold your software and data).
2. SSH to it and set the Gluster daemon to start at boot time.

Installation of Gluster on RHEL7/CentOS7

# yum update –y

#yum install wget –y

#wget <http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm>

#rpm -ivh epel-release-7-5.noarch.rpm

#wget -P /etc/yum.repos.d <http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo>

#yum install glusterfs\* -y –skip-broken

#systemctl enable glusterd

#systemctl start glusterd

#chkconfig glusterd on

#fdisk /dev/xvdb

and answer n, p, 1, enter, enter, w to create a single full-size primary partition

#partprobe

#mkfs.xfs -f /dev/xvdb

Configure the volume to be mounted at boot time:

#mkdir -p /mnt/ebs

#chown ec2-user:ec2-user /mnt/ebs

#echo '/dev/xvdb /mnt/ebs xfs rw,user,auto 0 0' >> /etc/fstab

# mount –a

**Create an AMI**

If this AMI will be made public, clean things up so it is secure:

# find /root/.\*history /home/\*/.\*history -exec rm -f {} \;

# find / -name "authorized\_keys" -exec rm -f {} \;

# exit

Go to the instance pane of your EC2 management console, select your instance, and click the Actions button to open its menu, then click “Create Image (EBS AMI).

In the Create Image wizard, provide a name ***gluster\_share\_storage*** and description and click "Yes, Create". (Doing this will log you out of your instance, after which you won't be able to log back in if you deleted authorized\_keys - you have been warned.)

**Terminate the Instance**

**STEP 2.**

# Create a Mountable Gluster Volume (gv1) with the help of 2 instance

Independently launch 2 new t1.micros running the new AMI, each with a unique manually chosen ip address in the subnet your configured previous instances will be in. The security group and availability zone should also match your previous instances. For example, the eu-west-1a subnet is 172.31.16.0/20 so you might chose 172.31.16.100 for the first, and 172.31.16.101 for the second. These instances should end up with their own independent copies of the formatted EBS volume, mounted as /mnt/ebs

Chose one of the instances and consider it as the 'Master Gluster Instance 1' and SSH to it

and probe the other one (instance 2) to form the network of Gluster instances:

Lets say the public IP of 'Master Gluster Instance 1' is 54.169.142.128 and Instance 2 is 54.169.142.129

#ssh -i “key.pem” ec2-user@ 54.169.142.128

Parallely ssh to another instance 2.

#ssh -i “key.pem” ec2-user@ 54.169.142.129

Perform the below operation on both the instances.

#yum install wget –y

#wget <http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm>

#rpm -ivh epel-release-7-5.noarch.rpm

#wget -P /etc/yum.repos.d <http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo>

#yum install glusterfs\* -y –skip-broken

#systemctl enable glusterd

#systemctl start glusterd

#chkconfig glusterd on

# mkdir /mnt/ebs

Perform the below operation in Master Gluster Instance only

Now Peer the another Instance 2 by probing its IP (let’s say the Private IP of Instance 2 is 172.31.8.25)

# gluster peer probe 172.31.8.25

# gluster peer status

Shows it worked

Create a Gluster volume that replicates the data from the master to the other instance:

# gluster volume create gv1 replica 2 172.31.8.26:/mnt/ebs 172.31.8.25:/mnt/ebs

# gluster volume start gv1

# Mount the Gluster Volume gv1 with the EBS of Shared Gluster Storage Instance

Now that a Gluster volume gv1 is available, we'll bring up a new normal ***gluster\_share\_storage*** instance and mount it.

Launch that gluster\_share\_storage AMI by all the same network configuration.

SSH to it its public IP and enter into that instance

Mount the Gluster volume:

# mkdir /shared

#chown ec2-user:ec2-user /shared  
  
#echo '172.31.8.26:/gv1 /shared glusterfs defaults 0 0' >> /etc/fstab => IP is of Master Instance 1

#mount /shared

If everythings goes well then go to /shared directory by

# cd /shared

And create files

#touch fedena{1..10}

Go to Master instance 1 and navigate to /mnt/ebs by

# cd /mnt/ebs

#ls

(the output will be same as you saw in the /shared directory of Shared Gluster Storage Instance)

Go to Instance 2 and navigate to /mnt/ebsby

# cd /mnt/ebs

#ls

(the output will be same as you saw in the /shared directory of Shared Gluster Storage Instance)

**Case: Adding the third Instance for scalability**

The third instance is added to check whether the shared memory will work under the case of scalability.

Launch the instance with in same network. And provide the 2 GB EBS and SSH to it.

Let’s say the public IP of third Instance is 54.169.172.27

Then do the following operation on the Terminal

#ssh –i “key.pem” [ec2-user@54.169.172.27](mailto:ec2-user@54.169.172.27)

# yum update –y

#yum install wget –y

#wget <http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm>

#rpm -ivh epel-release-7-5.noarch.rpm

#wget -P /etc/yum.repos.d <http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo>

#yum install glusterfs\* -y –skip-broken

#systemctl enable glusterd

#systemctl start glusterd

#chkconfig glusterd on

Configure the volume to be mounted at boot time:

#mkdir -p /mnt/ebs

GO into the Master Gluster Instance and add the third instance to its gluster list by probing the ip of third instance.

# gluster peer probe 172.31.9.58 => Private ip of the third instance.

# gluster volume add-brick gv1 replica 3 172.31.9.58:/mnt/ebs force => Adding the third instance mount point to gv1

Now go to third Instance and Navigate to /mnt/ebs and do ls then you will find every data which was in Master Gluster Instance and Instance 2 will be also updated to Instance 3.