Wilson wilson Follow
sysadmin interested in linux , privacy, docker, CI/CD, PAAS, dynamic storage,Dynamic routing...
Apr 3 · 5 min read

Install Heketi and GlusterFS with Openshift to allow dynamic Persistent-Volume management

In this tutorial we will see how to install glusterfs on a 6nodes cluster (with LVM) and how to manage it with heketi, then how to expose heketi's API to an Openshift cluster for dynanic volume management.

If you want to know why and where does the lvm volumes come from, what is openshift origin and how we installed it, this tutorial follows these 2 ones:

1. Change OVH kernel to enable SELinux on CentOS7
2. Install Openshift on OVH dedicated server

State of my LVM install (which is the same on the 6 dedicated servers)

We have a volume group named docker for docker usages
Here is my `vgdisplay`

```
— — Volume group — -
VG Name               docker
System ID
Format                lvm2
Metadata Areas        1
Metadata Sequence No  14
VG Access             read/write
VG Status             resizable
MAX LV                0
Cur LV                1
Open LV               1
Max PV                0
Cur PV                1
Act PV                1
VG Size               1,52 TiB
PE Size               4,00 MiB
Total PE              399043
Alloc PE / Size       103200 / 403,12 GiB
Free PE / Size        295843 / 1,13 TiB
VG UUID               DhPgBy-PctL-skSh-neV9-eeBe-rHYK-9X3Gvk
```

We have a lv named docker-pool on vg docker, it is used by the devicemapper for a 'direct-lvm' mode to run containers.
Here is my `lvdisplay`

```
 — — Logical volume — -
 LV Name                docker-pool
 VG Name                docker
 LV UUID                n7xORF-x1kc-Y5Xd-1TON-XMGu-syrE-
gZopee
 LV Write Access        read/write
 LV Creation host,      time $MASTER_1_HOSTNAME,
$date_creation
 LV Pool metadata       docker-pool_tmeta
 LV Pool data           docker-pool_tdata
 LV Status              available
 # open 25
 LV Size                400,00 GiB
 Allocated pool data    2,07%
 Allocated metadata     0,19%
 Current LE             102400
 Segments               1
 Allocation             inherit
 Read ahead sectors     auto
 — currently set to     8192
 Block device           253:2
```

Install GlusterFS

1. First we have to create a new LV named gluster in docker VG

```
# lvcreate -l 20 -n gluster docker
```

2. Set the size you want for your LV (i'll set 1TiB)

```
# lvextend -L1000000 /dev/docker/gluster
```

3. Search for the LTS glusterfs-server version and install it

```
# yum search centos-release-gluster
# yum install centos-release-gluster310
# yum install glusterfs-server
```

4. start and enable the service

```
# systemctl start glusterd
# systemctl enable glusterd
```

5. Manage firewalld acces to enable glusterfs

```
# firewall-cmd --add-service=glusterfs --permanent &&\
  firewall-cmd --reload
```

6. Allow write on mounted GlusterFS volume with SELinux

```
# setsebool -P virt_sandbox_use_fusefs on
```

All the previous steps must be done on all servers, you do it with following ssh loop

```
for host in $MASTER_1_IP \
    $MASTER_2_IP \
    $MASTER_3_IP \
    $NODE_3_IP \
    $NODE_2_IP \
    $NODE_3_IP;
    do ssh $host lvcreate -l 1000000 -n gluster docker ; \
                 yum -y install centos-release-gluster310;\
                 yum -y install glusterfs-server;\
                 systemctl start glusterd; \
                 systemctl enable glusterd; \
                 firewall-cmd --add-service=glusterfs \
                 --permanent; \
                 firewall-cmd --reload; \
                 setbool -P virt_sandbox_use_fusefs on; \
    done
```

Install and configure Heketi

We will install Heketi on the MASTER_1 server.

1. Add epel release repository and install heketi

```
# yum install -y epel-release
# yum -y --enablerepo=epel install heketi heketi-client
```

2. Copy the ssh key created for ansible install to /etc/heketi/heket_key

```
# cp /root/.ssh/id_rsa /etc/heketi/heketi_key
# chown heketi: /etc/heketi/heketi_key
```

You can also create a new one and propagate it on servers

3. Edit /etc/heketi/heketi.json config file

```
{
  "_port_comment": "Heketi Server Port Number",
  "port": "8080",
  "_use_auth": "Enable JWT authorization. Please enable for
deployment",
  "use_auth": true,
  "_jwt": "Private keys for access",
  "jwt": {
   "_admin": "HeketiAdmin",
   "admin": {
    "key": "$PASSWORD_ADMIN"
   },
   "_user": "HeketiUser",
   "user": {
    "key": "$PASSWORD_USER"
   }
  },
  "_glusterfs_comment": "GlusterFS Configuration",
  "glusterfs": {
   "executor": "ssh",
   "sshexec": {
     "keyfile": "/etc/heketi/heketi_key",
     "user": "root",
     "port": "22",
     "fstab": "/etc/fstab"
   }
  },
  "_db_comment": "Database file name",
  "db": "/var/lib/heketi/heketi.db",
  "loglevel" : "debug"
  }
}
```

4. Create the GlusterFS Cluster topology in file /usr/share/heketi/toppology-sample.json

```
{
    "clusters": [
        {
            "nodes": [
                {
                    "node": {
                        "hostnames": {
                            "manage": [
                                "$MASTER_1_IP"
                            ],
                            "storage": [
                                "$MASTER_1_IP"
                            ]
                        },
                        "zone": 1
                    },
                    "devices": [
                        "/dev/docker/gluster"
                    ]
                },
                {
                    "node": {
                        "hostnames": {
                            "manage": [
                                "$MASTER_2_IP"
                            ],
                            "storage": [
                                "$MASTER_2_IP"
                            ]
                        },
                        "zone": 2
                    },
                    "devices": [
                        "/dev/docker/gluster"
                    ]
                },
                {
                    "node": {
                        "hostnames": {
                            "manage": [
                                "$MASTER_3_IP"
                            ],
                            "storage": [
                                "$MASTER_3_IP"
                            ]
                        },
                        "zone": 1
                    },
                    "devices": [
                        "/dev/docker/gluster"
                    ]
                },
                {
                    "node": {
                        "hostnames": {
                            "manage": [
                                "$NODE_1_IP"
                            ],
                            "storage": [
                                "$NODE_1_IP"
                            ]
                        },
                        "zone": 1
                    },
                    "devices": [
                        "/dev/docker/gluster"
                    ]
                },
                {
                    "node": {
                        "hostnames": {
                            "manage": [
                                "$NODE_2_IP"
                            ],
                            "storage": [
                                "$NODE_2_IP"
                            ]
                        },
                        "zone": 1
                    },
                    "devices": [
                        "/dev/docker/gluster"
                    ]
                },
                {
                    "node": {
                        "hostnames": {
                            "manage": [
                                 "$NODE_3_IP"
                            ],
                            "storage": [
                                 "$NODE_3_IP"
                             ]
                         },
                        "zone": 2
                    },
                     "devices": [
                        "/dev/docker/gluster"
                    ]
                 }
            ]
        }
    ]
}
```

Configure Firewall access

1. Create /etc/firewalld/services/heketi.xml file for firewalld heketi service

```
<?xml version="1.0" encoding="utf-8"?>
<service>
 <short>Heketi</short>
 <description>Heketi glusterfs REST API</description>
 <port protocol="tcp" port="8080"/>
</service>
```

2. Set proper right on the file

```
# restorecon /etc/firewalld/services/heketi.xml
# chmod 640 /etc/firewalld/services/heketi.xml
```

3. Add Heketi service into internal firewalld zone

```
# firewall-cmd --zone=internal --add-service=heketi --
permanent
```

4. Add an access to that zone for every node in the cluster

```
for host in $MASTER_1_IP \
    $MASTER_2_IP \
    $MASTER_3_IP \
    $NODE_3_IP \
    $NODE_2_IP \
    $NODE_3_IP;
    do firewall-cmd --zone=internal\
                    --add-source=$host/32 --permanent; \
    done
```

5. Reload firewalld

```
# firewall-cmd --reload
```

Run Heketi

1. Start Heketi server

```
# systemctl start heketi
```

2. Load topology json file

```
# heketi-cli --server http://$MASTER_1_IP:8080 \
             --user admin --secret $PASSWORD_ADMIN \
             topology load \
             --json=/usr/share/heketi/topology-sample.json
```

3. Enable and restart Heketi

```
# systemctl enable heketi
# systemctl restart heketi
```

4. Check if Heketi is responding

```
# curl http://$MASTER_1_IP:8080/hello
Hello from Heketi
```

Connect Openshift with Heketi

1. Create a StorageClass SC_Heketi.yml

```
apiVersion: storage.k8s.io/v1beta1
kind: StorageClass
metadata:
  name: heketi
provisioner: kubernetes.io/glusterfs
parameters:
  resturl: "http://$MASTER_1_IP:8080"
  restuser: "admin"
  restuserkey: "$PASSWORD_ADMIN"
```

2. Create a PVC_Heketi.yml using the StorageCLass

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: heketi-pvc
 annotations:
   volume.beta.kubernetes.io/storage-class: heketi
spec:
 accessModes:
  - ReadWriteMany
 resources:
   requests:
     storage: 4Gi
```

Notice: This is by the annotation that you tell the PVC which storageclass to use

3. Send objects to openshift with oc create -f

```
# oc create -f SC_Heketi.yml -n default
# oc create -f PVC_Heketi.yml -n default
```

You are done here, we will now verify

Check if PVC is created running `oc get pvc`

```
NAME       STATUS VOLUME CAPACITY ACCESSMODES AGE
heketi-pvc Pending                            1s
```

You can see that the PVC is created and is now pending.
Just wait a few seconds and run again the command : `oc get pvc`

```
NAME       STATUS    VOLUME    CAPACITY ACCESSMODES AGE
heketi-pvc Bound     pvc-...   4Gi      RWX         16s
```

Now you can see that a Persistent volume is Binded to your PVC, let's check how is glusterfs now :

```
$ gluster volume list
vol_054068b3e1a656c57e0a6bac6462d4c8
```

List the PV with `oc get pv` you should see the newly created PV

```
NAME      CAP ACCESSMODES RECLAIMPOLICY  STATUS  CLAIM
pvc-[...] 4Gi RWX         Delete         Bound
heketi/heketi-pvc
```

Inspect the glusterfs volume :

```
$ gluster volume info vol_054068b3e1a656c57e0a6bac6462d4c8
```

```
Volume Name: vol_054068b3e1a656c57e0a6bac6462d4c8
Type: Replicate
Volume ID: 0bc87ad7-159b-4d7b-a368-30e9369ee3bf
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 3 = 3
Transport-type: tcp
Bricks:
```

```
Brick1:
$MASTER_1_IP:/var/lib/heketi/mounts/vg_7e804a6438f4cb40e96e3
e8546d14a1c/brick_4bc68c89029e76f170cd5c43526d48a5/brick
```

```
Brick2:
$MASTER_3_IP:/var/lib/heketi/mounts/vg_14b88aefc944a985e1d3c
2a03d17fe29/brick_810ae4dd513b7694cdd1bc4e2c51d885/brick
```

```
Brick3:
$NODE_2_IP:/var/lib/heketi/mounts/vg_75b6b7d29c202e5d9fc5beb
be68414d1/brick_4eea4728498c4e0ac9d85b9ea7480357/brick


Options Reconfigured:
transport.address-family: inet
nfs.disable: on
```

We can see that the volume is created and replicated 3 times on master1,master3 and node2.

Now if we delete the PVC, the PV should disappear

```
$ oc delete pvc heketi-pvc
persistentvolumeclaim "heketi-pvc" deleted
```

If we list again the PV, we can see that it has been deleted too

```
NAME CAPACITY ACCESSMODES RECLAIMPOLICY STATUS CLAIM REASON
AGE
```