

Red Hat OpenShift GlusterFS

Red Hat Gluster Storage ist eine SDS-Lösung zur Speicherung von persistenten Daten welche in Pods generiert werden. Im Gegensatz zu herkömmlichen Lösungen lässt sich Red Hat Gluster Storage über virtuelle, Bare Metal-, Container- und Cloud-Umgebungen hinweg skalieren und ist dazu noch äusserst flexibel und kostengünstig.



GlusterFS (with Heketi)

Bricks von einem PV anzeigen

Um von einem PV, welches einem Projekt angehängt ist, die einzelnen Bricks anzuzeigen, wird folgendermassen vorgegangen:

1. Nach dem entsprechenden PV "grep" und die ID kopieren:

```
$ oc get pv | grep <pvc_name>
```

2. Anschliessend die kopierte ID verwenden, um die Details des PV's zu betrachten. Die Volume ID muss nun kopiert werden. Diese wird unter "Path" oder "gluster.kubernetes.io/heketi-volume-id=" zu finden sein:

```
$ oc describe pv <pvc_id>
```

3. Nun muss via Konsole auf einen der entsprechenden Gluster Nodes gewechselt werden um darauf die Infos zum entsprechenden Volumen anzuzeigen. Aus diesem Grund wurde auch die Volume ID kopiert. Um nun den entsprechenden Brick anzusprechen, muss die Brick ID (*alles nach brick_*), welche auf dem aktuell verbundenen Gluster Node läuft kopiert werden (*hierzu die IP Adresse des Hosts vergleichen*):

```
$ ssh gluster_node
$ sudo -i
$ ip a
$ gluster volume info <volume_id>
```

4. Zum Schluss kann via Brick ID mittels `df -h` auf den entsprechenden Brick zugegriffen werden:

```
$ df -h | grep <brick_id>
```

Link: <https://access.redhat.com/solutions/3164841>

Troubleshooting Heketi

Alle Volumes welche Heketi bekannt sind anzeigen (Achtung: Um `heketi-cli` Operationen durchzuführen wird immer das `heketi Secret` gebraucht!):
Beispiel Secret : `p/J+OhZgDjwuMy/vZBoD74hP4CWmjWWE32ye6czr2aU=`

```
$ heketi-cli volume list --user admin --secret XXXXXXXXXXXXXXXXXXXXXXXX
```

Gluster Block Storage

Wenn die Pods welche Block Storage verwenden folgender oder einen ähnlichen Error erhalten, dann funktioniert der Block Service von Gluster nicht mehr richtig. **MountVolume.WaitForAttach failed for volume "pvc-cf639834-3353-11e9-91f6-005056a90a9e" : read /sys/class/iscsi_host/host39/device/session37/connection37:0/iscsi_connection/connection37:0/address: transport endpoint is not connected**

Als erstes muss überprüft werden, ob der entsprechende Block Service am Laufen ist:

```
$ systemctl status gluster-blockd.service
$ systemctl status gluster-block-target.service
```

Falls dieser nicht läuft (was anzunehmen ist), kann versucht werden diesen zu starten / stoppen:

```
$ systemctl stop gluster-blockd.service
$ systemctl start gluster-blockd.service
$
$ systemctl stop gluster-block-target.service
$ systemctl start gluster-block-target.service
```

Wenn er erfolgreich wieder startet kann überprüft werden, ob die Pods nun wieder Block Volume mounten können. Wenn ein Error beim Starten der Service erscheint, muss im Journald nachgeschaut werden, was schiefgegangen ist:

```
$ journalctl -xe
```

Ein möglicher Weg, wie der Block Storage wieder zum Laufen gebracht werden kann ist, die `saveconfig.json` Datei zu restaurieren. Hierbei muss darauf geachtet werden, dass die Datei `/var/lib/gluster-block/gb_upgrade.status` existiert. Falls diese nicht existiert muss diese erstellt werden. Wenn die Datei nicht existieren würde, würde eine neue `savecong.json` erstellt werden.

```
$ cd /etc/target/
$ cp "saveconfig.json-bak-yyyy-mm-dd-HH:MM:SS" saveconfig.json
$ touch /var/lib/gluster-block/gb_upgrade.status
$ ls -l /var/lib/gluster-block/gb_upgrade.status
```

```
-rw-r--r--. 1 root root 0 Feb 21 10:01 /var/lib/gluster-  
block/gb_upgrade.status
```

Anschliessend kann der Blockd Service gestoppt / gestartet werden. Der Block Service sollte nun wieder laufen (natürlich nur, wenn er vor dem Update bereits funktionierte):

```
$ systemctl stop gluster-blockd.service  
$ systemctl start gluster-blockd.service  
$  
$ systemctl stop gluster-block-target.service  
$ systemctl start gluster-block-target.service
```

GlusterFS (without Heketi)

Achtung: Manuelle Gluster PV Administration ist obsolete, wird neu mit heketi gemacht!

Gluster PVs erstellen (obsolete)

Um ein neues PV zu erstellen, muss als erstes berücksichtigt werden, ob genügend Platz zur Verfügung steht.

Folgender Ansible Befehl kann hierfür verwendet werden:

```
# ansible gluster -m shell -a "vgs | grep vg_gluster"
```

Wenn genügend Platz zur Verfügung steht, muss das Hostfile der jeweiligen Plattform angepasst werden:

```
# cd /root/openshift-hostfiles/  
# vim hosts_os***.yaml
```

Anschliessend im Hostfile das neue PV am Ende des Abschnittes "gluster_pvs" einfügen:

```
...  
gluster_pvs:  
  - name: gluster-pv1  
    options:  
      performance.write-behind-window-size: 32MB  
      size: 100  
  - name: gluster-pv2  
    options:  
      cluster.consistent-metadata: 'on'  
      cluster.post-op-delay-secs: '0'  
      size: 50  
  ...  
  - name: gluster-pv42
```

```
    size: 40
  - name: gluster-pv43
    size: 10
  - name: NEW GLUSTER PV
    size: NEW PV SIZE
  ...
```

Nun kann das Playbook main.yml im Verzeichnis /root/gluster-pvs/ im Check Modus ausgeführt werden. Damit kann überprüft werden, was das Playbook tatsächlich machen würde, ohne etwas zu ändern (dry run)

```
# cd /root/gluster-pvs/
# ansible-playbook --check main.yml
```

Nachdem der Check durchgelaufen ist (mit Fehler, da die Änderungen nicht gemacht wurden und so das PV so nicht gefunden wurde) kann das Playbook ohne das -check ausgeführt werden:

```
# cd /root/gluster-pvs/
# ansible-playbook main.yml
```

Zum Schluss muss noch überprüft werden, ob das PV tatsächlich angelegt wurde:

```
# oc get pv | grep CREATED GLUSTER-PV
```

Gluster PVs vergrößern (obsolete)

An OpenShift GlusterFS PV can be online resized, increased in size. This needs to be done on the OpenShift master node.

These are the steps:

Edit the Ansible Inventory file and set the new size of the pv. Push the changes also to gitit.pnet.ch

```
# cd /root/openshift-hostfiles/
# vim hosts_osit2.yaml
```

```
...
- name: gluster-pv118
  size: 5
...
```

```
# git status
```

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#       modified:   hosts_osit2.yaml
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
# git add hosts_osit2.yaml
# git commit -m "Resize pv118 to 5Gi"
# git push
```

Run `/root/gluster-pvs/ansible-playbook -check main.yml`. This is a dry run only to check what the changes will be

```
# cd /root/gluster-pvs/
ansible-playbook --check main.yml
```

After successful check run `/root/gluster-pvs/ansible-playbook main.yml`

```
# cd /root/gluster-pvs/
# ansible-playbook main.yml
```

Check the size of the gluster-pv on the local node. Should now show 1/3rd of the size in mounted position, because GlusterFS is replicated on the node instances.

```
# df -h | grep gluster-pv118
```

Run `oc edit pv <pv-name>` to have the metadata of the pv volume in consistent state on the master node

```
# oc edit pv gluster-pv118
```

```
...
  capacity:
    storage: 5Gi
  claimRef:
  ...
```

```
# oc get pv | grep pv118
```

CAUTION: Do not continue if there are errors to see in a step. If there are errors in the Task “[Configure Gluster management firewall

rules]“, when running `ansible-playbook -check main.yml`, note that during container restart, there is a firewall reconfiguration ongoing, which may disturb running `ansible-playbook`. Try this check again some moments later.

Gluster PV's löschen (obsolete)

Auf dem ersten Master das PV unmounten:

```
# umount /gluster/gluster-pvXX
# df | grep endpoint
```

Das PV im OpenShift entfernen:

```
# oc delete pv gluster-pvXX
```

Das gelöschte PV anschliessend auf einem Gluster Node Stopen und löschen:

```
# gluster volume stop gluster-pvXX
```

```
Stopping volume will make its data inaccessible. Do you want to continue?
(y/n) y
volume stop: gluster-pv11: success
```

```
# gluster volume delete gluster-pvXX
```

```
Deleting volume will erase all information about the volume. Do you want to
continue? (y/n) y
volume delete: gluster-pv11: success
```

Zum schluss folgende Ansible Befehle auf dem Master ausführen:

```
# ansible gluster -m lineinfile -a "path=/etc/fstab state=absent
regexp='^/dev/vg_gluster/gluster-pv11' backup=yes"

# ansible gluster -a "umount /dev/vg_gluster/gluster-pv11"

# ansible gluster -a "wipefs -a /dev/vg_gluster/gluster-pv11"      #(2x
ausführen zum checken)

# ansible gluster -a "lvremove /dev/vg_gluster/gluster-pv11 -y"
```

Performance I/O Test mit FIO

Um die I/O Performance von Gluster zu messen, kann das Programm FIO verwendet werden. Um einen solchen Test starten zu können, muss auf dem System, auf welchem der Test gestartet wird das Paket **glusterfs** installiert sein.

```
# yum install glusterfs fio -y
```

Nun muss das Volume ausfindig und auf dem auszuführenden Host gemountet werden:

```
$ oc get pv | grep <project_name>
$ oc describe pv <pv_name> | grep Path | awk '{print $2}'

$ mount -t glusterfs <random_gluster_node_ip>:<volume_name> /mnt/<path>
```

Zum Schluss muss noch die FIO Datei erstellt werden, in welcher definiert ist, was genau getestet wird.

```
$ cat /root/fio.txt
```

```
; -- start job file --
[global]
directory=/mnt/${target}
ioengine=libaio
# Linux native asynchronous I/O.
[first-run]
rw=randrw
# Random mixed reads and writes.
nrfiles=10000
# Number of files to use for this job.
blocksize=1k
# The block size in bytes used for I/O units. Default: 4096. A single value
applies to reads, writes, and trims.
direct=0
# If value is true, use non-buffered I/O.
size=1g
# The total size of file I/O for each thread of this job. Fio will run until
this many bytes has been transferred, unless runtime is limited by other
options
numjobs=8
# Create the specified number of clones of this job. Each clone of job is
spawned as an independent thread or process.
iodepth=8
# Number of I/O units to keep in flight against the file.
```

```
; -- end job file --
```

Schlussendlich FIO mit der erstellen Datei ausführen.

```
$ fio fio.txt
```

Last update: **2019/04/04 15:11**