ORACLE

Deploying a BeeGFS filesystem on OCI – step by step manual

Ainura Djumagulov
EMEA Cloud Specialist Engineer - Compute

Table of Contents

ORACLE

NB: Avoid directly doing copy/paste from this document since it could include hidden characters resulting into command lines failures.

1. Introduction

BeeGFS is a popular parallel file system, known for its performance, ease of use and simple installation. BeeGFS is frequently used with HPC workloads but can be used in any scenario that requires a file system. It is available free of charge for end users, however professional commercial support is also available, please visit beegfs.io for more details.

This manual demonstrates step by step instructions to deploy a small BeeGFS system for training, demo or testing purpose.

2. Infrastructure setup

The following components were used during the demo:

- BeeGFS v7.3.4
- VM.Standard.E4.Flex shapes for all servers
- All VMs runs Ubuntu 18.04 Platform image provided by OCI
- Block Volumes with Balanced Performance for Storage
- VCN with Public and Private Subnet and all required components were already in place. You can use "Wizard" to create a new vcn with internet connectivity, public and private subnets, NAT gateway and security lists.
- Pair of Private-Public key, which will be used to connect to all VMs.

The architecture of BeeGFS can be found here: https://doc.beegfs.io/7.3.4/architecture/overview.html#overview

The following picture shows the OCI architecture used in this demo:

ORACLE
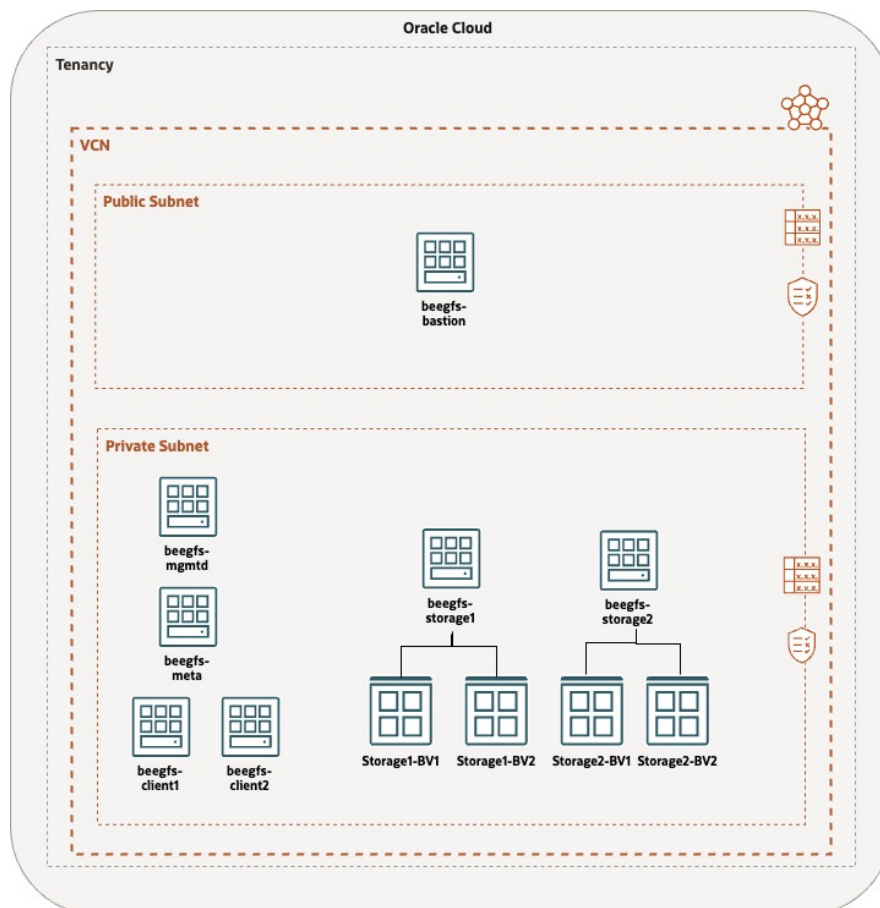
Please note that for a better performance OCI BM instances with RDMA support can be used instead of VM instances.

This demo is focused mainly on BeeGFS deployment and does not cover details about OCI Networking, Compartments, IAM etc. Please refer to corresponding documentation for details.

3. Installation

The official BeeGFS documentation is followed during this demo:

https://doc.beegfs.io/7.3.4/overview/overview.html#key-benefits

https://doc.beegfs.io/7.3.4/advanced_topics/manual_installation.html

3.1. VM Creation

The following 7 VMS are created (please see the architecture diagram):
- beegfs-bastion
- beegfs-meta
- beegfs-mgmtd
- beegfs-storage1
- beegfs-storage2
- beegfs-client1
- beegfs-client2

All VMs are Standard.E4.Flex VMs with 1 OCPU and 8 GB RAM. You can choose other shape and CPU/Memory count as per your requirements.

Note: beegfs-bastion is created in Public subnet and will be used as a jump server to connect to the environment. All other servers are created in Private subnet.

beegfs-bastion:

ORACLE
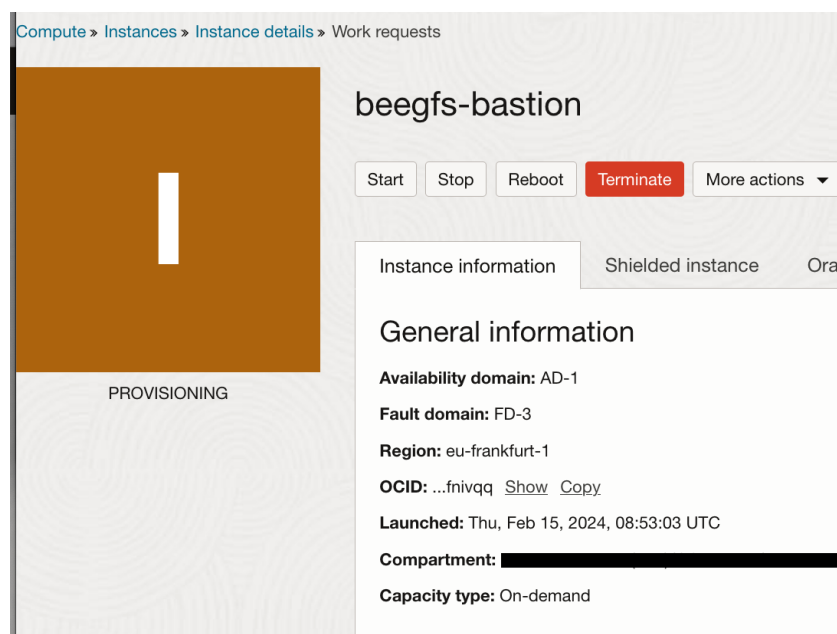
Instance access

The instance must be running before you can connect to it.

Public IP address: ███████████ Copy

Username: ubuntu

Primary VNIC

Public IPv4 address: ███████████

Private IPv4 address: 10.0.0.195

Network security groups: None Edit ⓘ

Subnet: Public Subnet-FRAVCN

Private DNS record: Enable

Hostname: beegfs-bastion

Internal FQDN: beegfs-bastion... Show Copy

Instance details

Virtual cloud network: FRAVCN

Maintenance reboot: -

Image: Canonical-Ubuntu-18.04-2023.05.10-0

Launch mode: PARAVIRTUALIZED

Instance metadata service: Versions 1 and 2 Edit ⓘ

Live migration: ✅ Enabled Change

Maintenance recovery action: Restore instance

Shape configuration

Shape: VM.Standard.E4.Flex

OCPU count: 1

Network bandwidth (Gbps): 1

Memory (GB): 16

Local disk: Block storage only

Any other VM in private subnet, exp:

beegfs-meta

Start Stop Reboot **Terminate** More actions ▼

PROVISIONING

Instance information   Shielded instance   Oracl

General information

Availability domain: AD-1

Fault domain: FD-3

Region: eu-frankfurt-1

OCID: ...ulw5ma Show Copy

Launched: Thu, Feb 15, 2024, 09:16:33 UTC

Compartment: ███████████████████

Capacity type: On-demand

Instance details

Virtual cloud network: FRAVCN

Maintenance reboot: -

Image: Canonical-Ubuntu-18.04-2023.05.10-0

Launch mode: PARAVIRTUALIZED

Instance metadata service: Versions 1 and 2 Edit ⓘ

Live migration: ✅ Enabled Change

Maintenance recovery action: Restore instance

Shape configuration
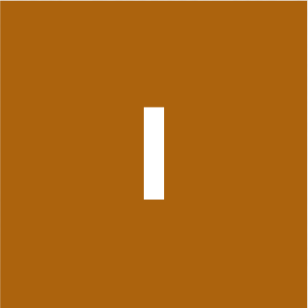
Shape: VM.Standard.E4.Flex

OCPU count: 1

Network bandwidth (Gbps): 1

Memory (GB): 16

Local disk: Block storage only

Instance access

The instance must be running before you can connect to it.

Primary VNIC

Public IPv4 address:

Private IPv4 address: 10.0.1.211

Network security groups: None Edit ⓘ

Subnet: Private Subnet-FRAVCN

Private DNS record: Enable

Hostname: beegfs-meta

Internal FQDN: beegfs-meta... Show Copy

ORACLE

All 7 VMs:

beegfs-client2	● Running	-	10.0.1.107	Thu, Feb 15, 2024, 09:25:51 UTC	VM.Standard.E4.Flex	1	16
beegfs-client1	● Running	-	10.0.1.216	Thu, Feb 15, 2024, 09:25:17 UTC	VM.Standard.E4.Flex	1	16
beegfs-storage2	● Running	-	10.0.1.31	Thu, Feb 15, 2024, 09:24:35 UTC	VM.Standard.E4.Flex	1	16
beegfs-storage1	● Running	-	10.0.1.214	Thu, Feb 15, 2024, 09:23:57 UTC	VM.Standard.E4.Flex	1	16
beegfs-mgmtd	● Running	-	10.0.1.76	Thu, Feb 15, 2024, 09:23:09 UTC	VM.Standard.E4.Flex	1	16
beegfs-meta	● Running	-	10.0.1.211	Thu, Feb 15, 2024, 09:16:33 UTC	VM.Standard.E4.Flex	1	16
beegfs-bastion	● Running	██████████	10.0.0.195	Thu, Feb 15, 2024, 08:53:03 UTC	VM.Standard.E4.Flex	1	16

3.2. BeeGFS Deployment

Copy the private key to beegfs-bastion:

```
scp –i mykey mykey ubuntu@public–ip:/home/ubuntu
```

Connect to beegfs-bastion VM and add the following lines to /etc/hosts

```
ubuntu@beegfs–bastion:~$ sudo vi /etc/hosts


10.0.1.211     beegfs-meta
10.0.1.76      beegfs-mgmtd
10.0.0.195     beegfs-bastion
10.0.1.31      beegfs-storage2
10.0.1.214     beegfs-storage1
10.0.1.216     beegfs-client1
10.0.1.107     beegfs-client2
```

Connect to all VMs (beegfs-meta, beegfs-mgmtd, beegfs-storage1, beegfs-storage2, beegfs-client1, beegfs-client2) via beegfs-bastion, exp:

```
ubuntu@beegfs–bastion:~$ ssh –i mykey ubuntu@beegfs–
storage1
```

ORACLE

On all VMs do the following:

1. Modify /etc/hosts and add the private ip address of VMs.

```
sudo vi /etc/hosts


10.0.1.211        beegfs-meta
10.0.1.76         beegfs-mgmtd
10.0.0.195        beegfs-bastion
10.0.1.31         beegfs-storage2
10.0.1.214        beegfs-storage1
10.0.1.216        beegfs-client1
10.0.1.107        beegfs-client2
```

2. Prepare for installation, download the requiored packages and keys.

```
sudo apt upgrade

sudo su -
cd /etc/apt/sources.list.d/
wget https://www.beegfs.io/release/beegfs_7.3.4/dists/beegfs-bionic.list
wget -q https://www.beegfs.io/release/beegfs_7.3.4/gpg/GPG-KEY-beegfs -O- | apt-key add -
apt update
```

3. Disable iptables, reboot and reconnect to all VMs.

```
(sudo su - )

iptables -L -v
iptables -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
systemctl disable netfilter-persistent
reboot
```

4. On beegfs-meta:

```
(sudo su - )

apt-get -y install beegfs-mgmtd

mkdir /beegfs_mgmtd

/opt/beegfs/sbin/beegfs-setup-mgmtd -p /beegfs_mgmtd
```

disable authentification checks:

```
vi /etc/beegfs/beegfs-meta.conf

connDisableAuthentication          = true
```

ORACLE

5. On beegfs-mgmtd:

```
(sudo su - )
apt-get -y install beegfs-mgmtd
mkdir /beegfs_mgmtd
/opt/beegfs/sbin/beegfs-setup-mgmtd -p /beegfs_mgmtd
```

disable authentification checks:

```
vi /etc/beegfs/beegfs-mgmtd.conf
connDisableAuthentication          = true
```

6. On Storage servers:

a. Create 4 Block Volumes:

Create Block Volume

Name	State	Size	Default performance ⓘ
beegfs-storage2-bv2	● Available	1024 GB	VPU:10
beegfs-storage2-bv1	● Available	1024 GB	VPU:10
beegfs-storage1-bv2	● Available	1024 GB	VPU:10
beegfs-storage1-bv1	● Available	1024 GB	VPU:10

Please note that AD's of block volumes should match corresponding storage VMs.

b. Attach block volumes to storage VMs (you can leave all parameters as default while attaching):

Attached block volumes

Block volumes provide high-performance network storage to support a broad range of I/O intensive workloads.

Attach block volume

Name	State	Volume type	Device path	Type	Access	Size	VPU	Multipath
beegfs-storage1-bv1	● Attached	Block volume	-	paravirtualized	Read/write	1 TB	10	No
beegfs-storage1-bv2	● Attached	Block volume	-	paravirtualized	Read/write	1 TB	10	No

Attached block volumes

Block volumes provide high-performance network storage to support a broad range of I/O intensive workloads.

Attach block volume

Name	State	Volume type	Device path	Type	Access	Size	VPU	Multipath
beegfs-storage2-bv1	● Attached	Block volume	-	paravirtualized	Read/write	1 TB	10	No
beegfs-storage2-bv2	● Attached	Block volume	-	paravirtualized	Read/write	1 TB	10	No

ORACLE

c. Check device paths, create a file system and mount block volumes (on both storage VMs):

```
(sudo su - )
root@beegfs-storage1:~# ls /dev/sd*
mkfs -t xfs /dev/sdb
mkfs -t xfs /dev/sdc
mkdir /disk1
mkdir /disk2
install beegfs software:
apt-get -y install beegfs-storage
```

d. Install BeeGFS on Storage VMs:

on beegfs-storage1:

```
(sudo su - )
/opt/beegfs/sbin/beegfs-setup-storage -p /disk1/beegfs_storage -s 3 -i 301 -m beegfs-mgmtd
/opt/beegfs/sbin/beegfs-setup-storage -p /disk2/beegfs_storage -s 3 -i 302
```

on beegfs-storage2:

```
(sudo su - )
/opt/beegfs/sbin/beegfs-setup-storage -p /disk1/beegfs_storage -s 4 -i 401 -m beegfs-mgmtd
/opt/beegfs/sbin/beegfs-setup-storage -p /disk2/beegfs_storage -s 4 -i 402
```

e. Disable authentification checks (on both VMs):

```
(sudo su - )
vi /etc/beegfs/beegfs-storage.conf
connDisableAuthentication    = true
```

7. On Client VMs

```
apt-get -y install beegfs-client beegfs-helperd beegfs-utils
/opt/beegfs/sbin/beegfs-setup-client -m beegfs-mgmtd
```

disable authentification checks (on both VMs):

```
(sudo su - )
vi /etc/beegfs/beegfs-client.conf
connDisableAuthentication    = true
vi /etc/beegfs/beegfs-helperd.conf
connDisableAuthentication    = true
```

ORACLE

8. Start BeeGFS on all VMs

  beegfs-mgmtd:

```
(sudo su - )

systemctl start beegfs-mgmtd
```

  beegfs-meta:

```
(sudo su - )

systemctl start beegfs-meta
```

  beegfs-storage1 and beegfs-storage2:

```
(sudo su - )

systemctl start beegfs-storage
```

  beegfs-client1 and beegfs-client2:

```
(sudo su - )

systemctl start beegfs-helperd

systemctl start beegfs-client
```

9. If beegfs-client started without errors then you file system is up and running; you can check the mount directory.

```
[root@beegfs-client1:/var/log# systemctl start beegfs-client
[root@beegfs-client1:/var/log# cd /mnt/beegfs/
[root@beegfs-client1:/mnt/beegfs# ls
[root@beegfs-client1:/mnt/beegfs# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.8G     0  7.8G   0% /dev
tmpfs           1.6G  992K  1.6G   1% /run
/dev/sda1        45G  2.7G   43G   6% /
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/loop0       56M   56M     0 100% /snap/core18/2745
/dev/loop1       54M   54M     0 100% /snap/snapd/19122
/dev/loop2       34M   34M     0 100% /snap/oracle-cloud-agent/54
/dev/sda15      105M  5.3M  100M   5% /boot/efi
tmpfs           1.6G     0  1.6G   0% /run/user/1001
beegfs_nodev    4.0T  4.3G  4.0T   1% /mnt/beegfs
root@beegfs-client1:/mnt/beegfs#
```

ORACLE

```
[root@beegfs-client2:~# systemctl start beegfs-client
[root@beegfs-client2:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.8G     0  7.8G   0% /dev
tmpfs           1.6G  992K  1.6G   1% /run
/dev/sda1        45G  2.6G   43G   6% /
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/loop0       56M   56M     0 100% /snap/core18/2745
/dev/loop1       34M   34M     0 100% /snap/oracle-cloud-agent/54
/dev/loop2       54M   54M     0 100% /snap/snapd/19122
/dev/sda15      105M  5.3M  100M   5% /boot/efi
tmpfs           1.6G     0  1.6G   0% /run/user/1001
beegfs_nodev    4.0T  4.3G  4.0T   1% /mnt/beegfs
root@beegfs-client2:~#
```

```
[root@beegfs-client1:/mnt/beegfs# touch client1.txt
[root@beegfs-client1:/mnt/beegfs# ls -lrt
total 0
-rw-r--r-- 1 root root 0 Feb 15 11:33 client1.txt
root@beegfs-client1:/mnt/beegfs#
```

```
[root@beegfs-client2:/mnt/beegfs# touch client2.txt
[root@beegfs-client2:/mnt/beegfs# ls -lrt
total 0
-rw-r--r-- 1 root root 0 Feb 15 11:33 client1.txt
-rw-r--r-- 1 root root 0 Feb 15 11:35 client2.txt
```

At this point the BeeGFS system is successfully deployed.

4. Some notes for troubleshooting

Generally, if there is a problem in setup, "systemctl start beegfs-client" will fail.

Check the network communication between nodes:

(sudo su -)

beegfs-ctl --listnodes --nodetype=management --details

successful communication:

```
root@beegfs-client1:/mnt/beegfs# beegfs-ctl --listnodes --nodetype=management --details
beegfs-mgmtd [ID: 1]
   Ports: UDP: 8008; TCP: 8008
   Interfaces: ens3(TCP)
```

unsuccessful communication:

```
root@beegfs-client1:~# beegfs-ctl --listnodes --nodetype=metadata --details
Management node communication failed: beegfs-mgmtd
root@beegfs-client1:~# cd /var/log/
```

ORACLE

Possible solutions:

- Check the security list for Private subnet to allow UDP ports
- Check the iptables configuration
- Check whether all nodes are created in the same subnet.
- The log file can be found under /var/log/beegfs-xxx.log on all nodes.
- You can increase the log level to 5 (default is 3) in conf file to see more details about errors, for exp:

  vi /etc/beegfs/beegfs-client.conf

  logLevel   = 5

ORACLE