Configuring Linux server to work with the BeaST storage system over iSCSI protocol

Mikhail Zakharov zmey20000@yahoo.com
Version 1.1 2018.10.26

Description

This document describes basic steps only for configuring Linux (Centos/Red Hat Enterprise Linux 7.5) based server to work with the BeaST storage system over iSCSI protocol. Some essential parts like security restrictions and server-side volume management are not considered in this document.

All commands are executed with root privileges.

The BeaST storage system configuration:

- the system configured to work in ALUA mode (`kern.cam.ctl.ha_mode=2` in `/boot/loader.conf`)
- two LUNs (`device-id a0v0 and device-id b0v0`) are defined in the BeaST /etc/ctl.conf on both controllers.
- ISCSI target definition in /etc/ctl.conf: `iqn.2016-01.local.beast:target0`

IP-addresses configuration:

Controller 1	10.10.10.211
Controller 2	10.10.10.212

Procedure

1. Install and enable dm-multipath and iSCSI initiator

```
# yum install -y device-mapper-multipath iscsi-initiator-utils
# systemctl enable iscsi iscsid
```

2. Connect with both controllers of the BeaST storage system

```
# iscsiadm -m discovery -t sendtargets -p 10.10.10.211
10.10.10.211:3260,257 iqn.2016-01.local.beast:target0
# iscsiadm -m node -T iqn.2016-01.local.beast:target0 -p 10.10.10.211:3260 --login
Logging in to [iface: default, target: iqn.2016-01.local.beast:target0, portal: 10.10.10.211,3260]
(multiple)

# iscsiadm -m discovery -t sendtargets -p 10.10.10.212
10.10.10.212:3260,257 iqn.2016-01.local.beast:target0
# iscsiadm -m node -T iqn.2016-01.local.beast:target0 -p 10.10.10.212:3260 --login
Logging in to [iface: default, target: iqn.2016-01.local.beast:target0, portal: 10.10.10.212,3260]
(multiple)
Login to [iface: default, target: iqn.2016-01.local.beast:target0, portal: 10.10.10.212,3260]
successful.
```

3. Check iSCSI connections

```
# iscsiadm -m session -P3 | less
```

4. Configure dm-multipath to work with the BeaST storage system

Multipath configuration of the Linux server depends on the BeaST storage system configuration. In the example below:

- the BeaST storage system configured to work in ALUA mode (`kern.cam.ctl.ha_mode=2` in `/boot/loader.conf`)
- two LUNs (`device-id a0v0 and device-id b0v0`) are defined in the BeaST /etc/ctl.conf on both controllers.

Edit `/etc/multipath.conf` on the Linux server to enable multipathing for the BeaST storage system:

```
defaults {
      user_friendly_names yes
      find_multipaths yes

      polling_interval 10
}

multipaths {
      multipath {
            wwid                    1FREEBSD_a0v0
            alias                   beast_l0
            path_grouping_policy    failover
            path_selector           "round-robin 0"
            rr_weight               priorities
            prio                    alua
            failback                10
            no_path_retry           30
      }
      multipath {
            wwid                    1FREEBSD_b0v0
            alias                   beast_l1
            path_grouping_policy    failover
            path_selector           "round-robin 0"
            rr_weight               priorities
            prio                    alua
            failback                16
            no_path_retry           30
      }
}
```

Make sure to specify `wwid` parameters according to the `device-id` definitions on the BeaST storage system.

5. Load dm-multipath kernel module and start the service

```
# modprobe dm-multipath
# systemctl start multipathd
# systemctl enable multipathd
```

6. Activate multipath configuration

Flush all unused multipath device maps to clean out old multipathing records:

```
# multipath -F
```

Create new multipathing:

```
# multipath -v2
```

```
beast_l1 (1FREEBSD_b0v0) dm-1 FREEBSD ,CTLDISK
size=256G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| `- 9:0:0:1 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  `- 8:0:0:1 sdd 8:48 active ready running
beast_l0 (1FREEBSD_a0v0) dm-0 FREEBSD ,CTLDISK
size=256G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| `- 8:0:0:0 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  `- 9:0:0:0 sde 8:64 active ready running
```

Configuration could be checked with:

```
# multipath -ll
```

and

```
# lsblk
...
sdc          8:32   0   256G  0 disk
└─beast_l0 253:0    0   256G  0 mpath /beast/l0
sdd          8:48   0   256G  0 disk
└─beast_l1 253:1    0   256G  0 mpath /beast/l1
sde          8:64   0   256G  0 disk
└─beast_l0 253:0    0   256G  0 mpath /beast/l0
sdf          8:80   0   256G  0 disk
└─beast_l1 253:1    0   256G  0 mpath /beast/l1
```

and

```
# ls /dev/mapper/ -rlta
total 0
crw-------  1 root root 10, 236 Jun 10 10:42 control
drwxr-xr-x 18 root root    3540 Jun 11 13:07 ..
drwxr-xr-x  2 root root     100 Jun 11 13:07 .
lrwxrwxrwx  1 root root       7 Jun 11 13:27 beast_l0 -> ../dm-0
lrwxrwxrwx  1 root root       7 Jun 11 13:27 beast_l1 -> ../dm-1
```

After that point, depending on the local policies and practices, remote volumes are ready to be used as normal raw drives on the Linux server: create a filesystem on them, mount or put them under volume manager control in advance.

Detaching from iSCSI drives

- Umount the filesystem
- Remove drives from the volume manager
- Logout and delete iscsi-initiator:

```
# iscsiadm -m node -T iqn.2016-01.local.beast:target0 -p 10.10.10.211:3260 --logout
# iscsiadm -m node -T iqn.2016-01.local.beast:target0 -p 10.10.10.211:3260 -o delete

# iscsiadm -m node -T iqn.2016-01.local.beast:target0 -p 10.10.10.211:3260 --logout
# iscsiadm -m node -T iqn.2016-01.local.beast:target0 -p 10.10.10.211:3260 -o delete
```

- Clean out multipathing records:

```
# multipath -F
```