RPi 4B iscsi how-to

Inspiration from hiroom2.com. Kudos!

Foreword

These steps are meant to be completed on a Linux computer that serves as a netboot server of the MicroMEC cluster. The steps can of course be adopted for other use cases.

- Our netboot server (ie. iscsi target server) is called `bootserv`.
- The netboot server is available on the LAN and can be pinged using the name `bootserv`.
- We refer to the RPi 4B with an id: `0dc0a15d` and a name: `rpi4-1`.
- We use sudo and start the preparations in /tmp on the netboot server.

Security Note

The setup is meant for a private lab. Security considerations for exposing the rootfs via the LAN are not discussed in this how-to as of now.

Prerequisites

- A computer running Debian Linux. Any Linux flavor can be used, but our instructions refer to Debian.
- 15GB free space on the hard drive.
- Installed packages: tgt, open-iscsi.
- Fast LAN (preferably Gigabit Ethernet).

Steps

1. Download a ready made Raspbian Buster lite image.

The downloaded image has a rootfs which is about 1.5GB large. For MicroMEC we need a rootfs that is 4GB at least. We will need to copy the rootfs from the buster image to a large enough image file.

1. Extract the downloaded Raspbian Buster lite image

```
$ cd /tmp

$ mkdir raspbian_buster

$ unzip 2020-02-13-raspbian-buster-lite.zip
Archive:  2020-02-13-raspbian-buster-lite.zip
  inflating: 2020-02-13-raspbian-buster-lite.img
```

2. Setup a virtual loopback device using the extracted image

```
$ sudo losetup -fP 2020-02-13-raspbian-buster-lite.img

$ lsblk
NAME               MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
loop0                7:0    0   1.7G  0 loop
loop0p1            259:0    0   256M  0 part
loop0p2            259:1    0   1.5G  0 part
...
```

3. Mount the rootfs partition from the extracted image. The rootfs is available via /dev/loop0p2 in our case.

```
$ mkdir raspbian_rootfs

$ sudo mount /dev/loop0p2 raspbian_rootfs
```

4. Create a virtual loop back device that will hold the resized rootfs

```
$ cd /tmp

$ dd if=/dev/zero of=0dc0a15d-raspbian-rootfs.img bs=400M count=10

$ sudo mkfs.ext4 0dc0a15d-raspbian-rootfs.img

$ sudo losetup -fP 0dc0a15d-raspbian-rootfs.img
```

5. Check which loopback devices are allocated by the kernel:

```
$ losetup -a
/dev/loop1: []: (/tmp/0dc0a15d-raspbian-rootfs.img)
/dev/loop0: []: (/tmp/2020-02-13-raspbian-buster-lite.img)
```

6. Mount the new virtual block device

```
$ mkdir 0dc0a15d-rootfs-mount

$ sudo mount -o loop /dev/loop1 0dc0a15d-rootfs-mount
```

7. Copy the content of the Rasbian Buster rootfs to the mounted block device

```
$ sudo cp -R /tmp/raspbian_rootfs/* 0dc0a15d-rootfs-mount/
```

8. Unmount the file and move it to the place where it will be served.

```
$ sudo umount 0dc0a15d-rootfs-mount

$ sudo losetup -D

$ sudo mkdir /srv/iscsi

$ sudo mv 0dc0a15d-raspbian-rootfs.img /srv/iscsi
```

9. Prepare the iscsi target and publish it

```
$ sudo tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.org.micromec:rpi4-1-raspbian-rootfs

$ sudo tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 -b /srv/iscsi/0dc0a15d-raspbian-rootfs.img

$ sudo tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL
```

Note

If your iscsi server has other targets then you will need to pick a different tid.

At this point the rootfs is available on the local network.

10. Save the configuration on the netboot server to remain persistent

```
$ sudo tgt-admin --dump | sudo tee  /etc/tgt/conf.d/micromec-cluster.conf
```

Local Testing

1. Check the iscsi target locally

Discover the iscsi target

```
$ sudo iscsiadm --mode discovery --op update --type sendtargets --portal localhost
127.0.0.1:3260,1 iqn.org.micromec:rpi4-1-raspbian-rootfs
```

Login to the iscsi target

```
$ sudo iscsiadm -m node --targetname iqn.org.micromec:rpi4-1-raspbian-rootfs  -p localhost -l
Logging in to [iface: default, target: iqn.org.micromec:rpi4-1-raspbian-rootfs, portal: 127.0.0.1,3260] (multiple)
Login to [iface: default, target: iqn.org.micromec:rpi4-1-raspbian-rootfs, portal: 127.0.0.1,3260] successful.
```

Check if a new partition appears in the list:

```
$ cat /proc/partitions
major minor  #blocks  name
.....
   8       32    4096000 sdc
```

Mount the partition to a mount point

```
$ mkdir /tmp/test-rootfs

$ sudo mount /dev/sdc /tmp/test-rootfs

$ ls -alrt /tmp/test-rootfs
```

Umount the partition

```
$ sudo umount /tmp/test-rootfs
```

Logout from all iscsi targets

```
$ sudo iscsiadm -m node -U all
```

Remote Testing

1. Check the iscsi target remotely

   Login to an other Linux computer which also has the open-iscsi tools installed.

   Discover the iscsi target

   ```
   $ sudo iscsiadm --mode discovery --op update --type sendtargets --portal bootserv
   192.168.4.1:3260,1 iqn.org.micromec:rpi4-1-raspbian-rootfs
   ```

   Login to the iscsi target

   ```
   $ sudo iscsiadm -m node --targetname iqn.org.micromec:rpi4-1-raspbian-rootfs -p bootserv -l
   Logging in to [iface: default, target: iqn.org.micromec:rpi4-1-raspbian-rootfs, portal: 192.168.4.1,3260]
   Login to [iface: default, target: iqn.org.micromec:rpi4-1-raspbian-rootfs, portal: 192.168.4.1,3260]
successful.
   ```

   Check the available partitions

   ```
   $ cat /proc/partitions
   major minor  #blocks  name
   .....
      8       16    4096000 sdb
   ```

   Mount the partition to a mount point

   ```
   $ mkdir /tmp/test-rootfs

   $ sudo mount /dev/sdb /tmp/test-rootfs

   $ ls -alrt /tmp/test-rootfs
   ```

   Umount the partition

   ```
   $ sudo umount /tmp/test-rootfs
   ```

   Logout from all iscsi targets

   ```
   $ sudo iscsiadm -m node -U all
   ```