

Exercise 21.3: Configure and test NBD

Overview

In this lab we are going to configure and test **NBD**, communicating via the **loopback** device. Both client and server will be running on the same system.

For the **NBD** export, an empty file will be used.

The `/opt` directory is to be used for the exported device and configuration files. Production environments may prefer different locations.



CentOS-stream-8 issue

At creation time **CentOS-8-Stream** is not working correctly.

Please use **CentOS-Stream-9** or **Fedora**. Waiting for resolution.

1. First step:



On Red Hat, Centos, or Fedora

Prepare a directory to house the source code downloaded with **git**. Clone, build the **NBD** programs and install.



On Debian, Ubuntu, or Linux Mint

Install the **NBD** client and server packages.

2. Create a `nbd-server.conf` file in the `/opt` directory. A minimal, no comments file is sufficient. (A more interesting configuration file can be found in the SOLUTIONS directory.
3. Create two files for exporting and set the ownership to `student`.
4. Start the **nbd-server**.
5. Verify the nbd server responds to a query from the client.
6. Install the **nbd** kernel module, verify the block devices are present.
7. Connect the server supplied image, using the ip address, port to the local block device.
8. Wipe out any existing information on the **NBD**.
9. Using the **NBD**, create a **GPT**.
10. Create a partition on the **NBD**.
11. Add an **ext3** filesystem to the **NBD** and test.

✓ Solution 21.3

1. First step:



On Red Hat, Centos, or Fedora

```
$ mkdir ~/src
$ cd ~/src
$ git clone https://github.com/NetworkBlockDevice/nbd.git
$ cd nbd/
```

The packages `docbook-utils` and `autoconf-archive` may not be installed on your system, install them now:

```
$ sudo dnf install docbook-utils
$ sudo dnf install autoconf-archive
```

Build the programs:

```
$ ./autogen.sh
$ ./configure
$ make
$ sudo make install
```



On Debian, Ubuntu, or Linux Mint

The **Debian**, **Ubuntu**, or **Linux Mint** have deb packages available, install the **NBD** utilities.

```
$ sudo apt install nbd-client nbd-server
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nbd-client nbd-server
0 upgraded, 2 newly installed, 0 to remove and 43 not upgraded.
\ldots output truncated
```

2. Create the **nbd-server** configuration file in the `/opt` directory. **sudo** will be required to edit files in the `/opt` directory.

```
$ sudo vim /opt/nbd-server.conf
```



/opt/nbd-server.conf

```
[generic]
  allowlist = 1
  listenaddr = 0.0.0.0
  port = 10042
[foo]
  exportname = /opt/dsk1
  readonly = false
[bar]
  exportname = /opt/dsk2
```

3. `sudo dd if=/dev/zero of=/opt/dsk1 status=progress bs=100M count=5`
`sudo dd if=/dev/zero of=/opt/dsk2 status=progress bs=100M count=5`

```
sudo chmod 777 /opt
sudo chown student.student /opt/*
```

4. `$ sudo nbd-server -C /opt/nbd-server.conf`

5. `$ sudo nbd-client -l 127.0.0.1 -p 10042`

```
Negotiation: ..
foo
bar
```

6. `$ sudo modprobe -i nbd`
`$ ls /dev/nbd*`

```
/dev/nbd0 /dev/nbd11 /dev/nbd14 /dev/nbd3 /dev/nbd6 /dev/nbd9
/dev/nbd1 /dev/nbd12 /dev/nbd15 /dev/nbd4 /dev/nbd7
/dev/nbd10 /dev/nbd13 /dev/nbd2 /dev/nbd5 /dev/nbd8
```

7. `$ sudo nbd-client -N foo 127.0.0.1 10042 /dev/nbd0`

8. Wipe out the NBD:

```
$ sudo dd if=/dev/zero of=/dev/nbd0 bs=1M count=5
```

```
5+0 records in
5+0 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.00161564 s, 3.2 GB/s
```

9. Create a GPT label on the NBD:

```
$ sudo su -c "echo 'label: gpt' | sfdisk /dev/nbd0"
```

```
Checking that no-one is using this disk right now ... OK

Disk /dev/nbd0: 500 MiB, 524288000 bytes, 1024000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

>>> Script header accepted.
>>> Done.
Created a new GPT disklabel (GUID: 000C20C1-6929-424F-93E1-28319DE1FF1F).

New situation:
Disklabel type: gpt
Disk identifier: 000C20C1-6929-424F-93E1-28319DE1FF1F

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

10. Add a partition to the NBD:

```
$ sudo su -c "echo ';' | sfdisk /dev/nbd0"
```

```

Checking that no-one is using this disk right now ... OK

Disk /dev/nbd0: 500 MiB, 524288000 bytes, 1024000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 000C20C1-6929-424F-93E1-28319DE1FF1F

Old situation:

>>> Created a new GPT disklabel (GUID: EFFC087B-A15C-7D42-809C-4BDED58EE238).
/dev/nbd0p1: Created a new partition 1 of type 'Linux filesystem' and of size 499 MiB.
/dev/nbd0p2: Done.

New situation:
Disklabel type: gpt
Disk identifier: EFFC087B-A15C-7D42-809C-4BDED58EE238

Device      Start      End Sectors Size Type
/dev/nbd0p1  2048 1023966 1021919 499M Linux filesystem

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```

11. Add an **ext3** filesystem to the **NBD** and test:

```
$ sudo mkfs.ext3 -L nbd-foo /dev/nbd0
```

```

mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 128000 4k blocks and 128000 inodes
Filesystem UUID: 6a01ff5a-9ad5-4f43-8050-9f2b8df26c14
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

```

```

$ sudo mount LABEL=nbd-foo /mnt
$ sudo touch /mnt/file1
$ ls -l /mnt

```

```

total 16
-rw-r--r-- 1 root root    0 Jun  9 10:58 file1
drwx----- 2 root root 16384 Jun  9 10:56 lost+found

```