Connect to a remote block device using NBD

Content

Tested on

Debian (Lenny, Squeeze)
Ubuntu (Lucid)

Objective

To connect to a block device that has been exported using the NBD (Network Block Device) protocol

Background

The Network Block Device protocol allows a block device (or an image of a block device) to be exported from one machine to another. Typically the device would then be mounted as a filesystem or used as a swap area. Exported devices are identified by the IP address and TCP port number at which they are hosted.

Scenario

Suppose that you wish to use a block device that has been exported using NBD by the host `server.example.com` on TCP port 9000.

Method

Overview

The method described here has three steps:

1. Install the NBD client.
2. Load the NBD kernel module (if necessary).
3. Start an instance of the NBD client.

Debian-based systems provide for an alternative, fully persistent method that is controlled by a configuration file. This will be the subject of a future microHOWTO.

Install the NBD client

First install the NBD client if it is not already present. On Debian-based systems this is provided by the `nbd-client` package:

```
apt-get install nbd-client
```

On Red Hat-based systems that support NBD, both client and server are provided by the `nbd` package:

```
yum install nbd
```

Load the NBD kernel module

On Debian-based systems the NBD kernel module is loaded automatically if the NBD client is installed. On Red Hat-based systems this is not the case and it must be loaded explicitly:

```
modprobe nbd
```

You can check whether the module is loaded using the `lsmod` command:

```
lsmod | grep nbd
```

Provided that you are running `udev` or an equivalent, loading the NBD kernel module should have the effect of creating a set of device nodes for use by NBD, the first of which is named `/dev/nbd0`.

Start an instance of the NBD client

An instance of the NBD client must be started for each block device that is to be imported. It is necessary to state the hostname or address of the remote server, the port number, and the name of the local device to be used to present the remote device:

```
nbd-client server.example.com 9000 /dev/nbd0 -persist
```

The local block device must be an NBD block special device and must not already have been used. If you wish to present multiple NBD devices then you will needed to keep track of which device nodes are available and which are in use.

The `-persist` option causes the NBD client to reconnect if the connection is dropped for any reason. It does *not* allow the connection to survive a reboot of the client machine.

The client daemonises itself by default. Before it does so it displays the total size, block size and block count of the remote device, for example:

```
Negotiation: ..size = 1048576KB
bs=1024, sz=1048576
```

To disconnect from the remote device you should execute the `nbd-client` command again with the `-d` option:

```
nbd-client -d /dev/nbd0
```

Testing

One way to check whether an NBD device is connected is to inspect its size using the `blockdev` command:

```
blockdev --getsize64 /dev/nbd0
```

If the local device is connected then the above command should display the size of the remote device in bytes. If it is not connected then it should display a size of zero.

Variations

Swap over NBD

If the block device is to be used as a swap area then you should additionally use the `-swap` option to reduce the risk of deadlocks:

```
nbd-client server.example.com 9000 /dev/nbd0 -swap -persist
```

Note that the meaning of the `-swap` option changed in version 2.9.12 of NBD (corresponding to Debian Squeeze or Ubuntu Lucid). Prior to this version it depended on a very old, unmaintained kernel patch. Use of version 2.9.12 or later is strongly recommended if wish to swap over NBD; even then it is not considered to be completely safe, however its successful use by LTSP indicates a reasonable level of usability in practice.

A set of kernel patches have been proposed that would further improve the safety of swap over NBD, but at the time of writing they had not been incorporated into the mainline kernel.

See also

- Export a block device using NBD

Further reading

- Network Block Device (project homepage)
- Mel Gorman, Swap-over-NBD without deadlocking V7, Linux Kernel Mailing List, October 2011
- Wouter Verhelst, comment to Moving from a Laptop to a Cloud Lifestyle, September 2011

Tags: nbd