micro HOWTO

Export a block device using NBD

Content

Tested on

Debian (Lenny, Squeeze)
Ubuntu (Lucid)

Objective

To export a block device using the NBD (Network Block Device) protocol

Background

The NBD 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 from which they are hosted.

Scenario

Suppose that you wish to export the logical volume `/dev/vg/foo` to another machine using the NBD protocol on TCP port 9000. The volume should be re-exported in the event of a reboot.

Method

Overview

The method described here has three steps:

1. Install the NBD server.
2. Add the export to the NBD server configuration file.
3. Restart the NBD server.

Install the NBD server

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

```
apt-get install nbd-server
```

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

```
yum install nbd
```

List the export in the NBD server configuration file

The configuration file for the NBD server is usually located at `/etc/nbd-server/config`, however neither Debian nor Red Hat install one by default. A suitable configuration file for the scenario described here would be the following:

```
[generic]

[foo]
    exportname = /dev/vg/foo
    port = 9000
```

The file consists of a sequence of sections, each begun by a name in square brackets. In this case there are two: `[generic]` and `[foo]`. The `[generic]` section is used for global settings that apply to all exports. It must always be included (even if it is empty) and it must occur first in the configuration file.

Each section that follows describes a device to be exported. In this case there is one, named `[foo]`, which exports `/dev/vg/foo` on port 9000. The pathname and port number must always be specified, as in this example. The choice of section name is arbitrary, except that the name of each section must be unique.

Restart the NBD server

The NBD server should now be instructed to reload its configuration file. Do this using the `service` command if this is available on your system:

```
service nbd-server force-reload
```

or alternatively, using the corresponding `init.d` script:

```
/etc/init.d/nbd-server force-reload
```

Unfortunately the only way that the server can do this at present is by killing all instances of the daemon then restarting them. This has the effect of terminating any active connections, which could result in data loss. Running `nbd-client` with the `-persist` option should prevent any adverse effects provided that the export in question reappears after the interruption.

Testing

The exported device can be tested by attempting to mount it using `nbd-client`, either from another machine or via the loopback interface:

```
nbd-client 127.0.0.1 9000 /dev/nbd0
```

The third parameter should be the name of a pre-existing NBD block special device.

Variations

Non-persistent exports

A device can be exported non-persistently by explicitly starting an instance of the NBD server daemon:

```
nbd-server -C /dev/null 9000 /dev/vg/foo
```

The purpose of the `-C` option is to prevent the server daemon from reading its normal configuration file. Without this, the daemon would attempt to serve any exports listed in the configuration file in addition to the one listed on the command line. The server daemon may print a warning when the configuration file contains no exports, for example:

```
** (process:29886): WARNING **: Could not parse config file: Unknown error
```

This should be ignored. Note that the location of the block device must be given as an absolute pathname (as opposed to one that is relative to the current directory).

Read-only exports

The client can be prevented from writing to the block device using the `readonly` setting for persistent exports:

```
[foo]
    exportname = /dev/vg/foo
    port = 9000
    readonly = true
```

or the `-r` option for non-persistent exports:

```
nbd-server -C /dev/null -r 9000 /dev/vg/foo
```

If the device contains a filesystem then the client will need to mount it either using the `ro` (read-only) option, or using a filesystem that is implicitly read-only (such as `iso9660`). Multiple concurrent connections are safe when the device is read-only.

Copy-on-write exports

An alternative to making the device fully read-only is to allow temporary changes which persist only while the client is connected. This can be done using the `copyonwrite` setting for persistent exports:

```
[foo]
    exportname = /dev/vg/foo
    port = 9000
    copyonwrite = true
```

or the `-c` (copy-on-write) option for non-persistent exports:

```
nbd-server -C /dev/null -c 9000 /dev/vg/foo
```

Any changes are written to a temporary file on the server. The device thus appears to be read-only to the server, but can be mounted read-write by the client.

A separate temporary file is created for each connection. Multiple concurrent connections are therefore safe, but changes made by one client are not visible to other clients.

When using copy-on-write, enabling the `sparse_cow` option may improve performance.

Security

Access to an exported device can be restricted several ways:

- by binding the server socket to a particular IP address using the `listenaddr` option,
- by listing IP addresses that can connect in an authorisation file (typically `/etc/nbd-server/allow` by default, or at a pathname specified using the `authfile` option), or
- by using an external mechanism such as `iptables` to block access to the port.

If strong authentication is needed then an effective method might be to block direct external access to the port, but then providing legitimate users with access through an SSH or VPN tunnel.

See also

- Connect to a remote block device using NBD

Further reading

- Network Block Device (project homepage)

Tags: nbd