

NFS Server Cluster

The application was clustered on Centos 4 [Red Hat Enterprise 4 Clone]. Commands used in this guide, other than those related to **rpm/yum**, should be available on all distributions of Linux.

Assumptions

Installation and configuration of Linuxha.net is well documented elsewhere. This guide assumes the software has been installed and is working, even if no applications are currently clustered.

It is also assumed that the machines will be firewalled - always a good practise. Hence the configuration here contains relevant entries for Iptables to ensure NFS can be seen through the firewall.

The following NFS related packages are required - the names may vary with distribution:

- ufs-utils
- quota
- portmap

If the OS installation is classed as a "server" then the packages are already likely to be installed. Otherwise the following command may work:

```
# yum install nfs-utils quota portmap
```

The descriptions in this guide allow different NFS exports to be classed as different packages - which if you need to export more than a single file system is a good idea - for example it allows active/active configurations or allows just certain applications/file systems to be brought down without affecting others.

Tying Down NFS Server Ports

By default, because NFS uses RPC the actual ports that the relevant daemons listen on is not pre-determined - which causes problems with firewall rules.

Hence ensure that NFS server daemons do **not** start automatically on machine boot. This can be achieved by removing the relevant files from /etc/init.d/rc3.d directory, for example:

```
# mv S14nfslock _S14nfslock
```

By using custom scripts to start the NFS services it is possible to tie down the ports - however some changes are necessary to host configuration files.

/etc/modules.conf

The following line should be added to ensure when the kernel module is loaded the port the lock module uses is fixed:

```
options lockd nlm_udpport=4001 nlm_tcpport=4001
```

/etc/rpc

Ensure the following line is present [should be]:

```
rquotad 100011 rquotaprog quota rquota
```

/etc/services

Add or modify entries for "rquotad" to hard code port 4004:

```
rquotad      4003/tcp
rquotad      4003/udp
```

The above changes should be carried out on both servers. After making the changes restart the servers if possible and double check no NFS server daemons are running:

```
# ps -ef | egrep "nfs|rpc"
```

This command should not show the following as running:

```
rpc.statd
rpc.nfsd
rpc.lockd
rpc.mountd
rpc.rquotad
```

NFS Start/Stop Scripts

Once this has been proven then the next step is to create the scripts to stop, start and restart this services.

In this example the three scripts that are put in place are:

```
/usr/local/cluster/nfs_start      - Starts the NFS daemons
/usr/local/cluster/nfs_stop       - Stops the NFS daemons
/usr/local/cluster/nfs_restart    - Restarts the NFS daemons
```

The contents of each are very straightforward:

nfs_start

```
#!/bin/bash

L="$(ps -ef | grep nfsd | grep -v grep)"
if [[ -z "$L" ]]
then
    /sbin/rpc.statd -p 4000
    /usr/sbin/rpc.nfsd 8
    /sbin/rpc.lockd
    /usr/sbin/rpc.mountd -p 4002
    /usr/sbin/rpc.rquotad -p 4003
fi

/usr/sbin/exportfs -o rw */:/nfsdata
exit 0
```

nfs_stop

```
#!/bin/bash

/usr/sbin/exportfs -u */:/nfsdata
exit 0
```

nfs_restart

```
#!/bin/bash
/usr/local/cluster/nfs_stop
/usr/local/cluster/nfs_start
exit 0
```

The above should be installed on both nodes - and made executable, i.e.

```
# chmod +x /usr/local/cluster/nfs_*
```

Firewall Configuration

The next step is to alter the firewall rules. To do this add the following entries to the IP Tables configuration file - typically `/etc/sysconfig/iptables`:

```
# NFS server - start
-A RH-Firewall-1-INPUT -p tcp --dport 111 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 111 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp --dport 2049 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 2049 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp --dport 4000 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 4000 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp --dport 4001 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 4001 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp --dport 4002 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 4002 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp --dport 4003 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 4003 -j ACCEPT
# NFS server - finish
```

Repeat this step on both machines and then restart the firewall to pick up the new rules. This is typically done by issuing:

```
# /etc/init.d/iptables restart
```

Repeat the above command on both hosts.

Linuxha.net Application Definition

All the steps in this section can be completed on a single machine - the steps themselves will copy any files to the other host in the cluster if necessary. Typically the cluster is expected to be running whilst the NFS application is added, though that does not need to be the case.

The first step is to create a directory called `/etc/cluster/nfs` and create an application configuration file in it:

`/etc/cluster/nfs/appconf.xml`

```
<?xml version="1.0"?>
<appconf>
  <global>
    <version>0.3</version>
    <name>nfs</name>
    <takeover>force</takeover>
    <synccrate>5000</synccrate>
    <preferred_node>LEAST_CPU_LOAD</preferred_node>
    <autostart>1</autostart>
  </global>

  <networks>
    <network net="prod" ip="192.168.1.221"/>
  </networks>

  <!-- Currently it is only possible to have a single volume
  group, and the type must be filesystem.
  -->
  <vg>
    <name>nfsvg</name>
    <type>filesystems</type>
  </vg>
```

```

<!-- Now define the scripts that are used to start and
      stop the application in question. Can include arguments
      if necessary.
-->

<application>
  <startscript>/usr/local/cluster/nfs_start</startscript>
  <stopscript>/usr/local/cluster/nfs_stop</stopscript>
  <maxstoptime>10</maxstoptime>
  <maxstarttime>10</maxstarttime>
</application>
</appconf>

```

In the above example the network and IP address are shown in bold and will need to be modified to suit each environment. The IP address is the one clients should use to attach to the NFS file systems the cluster NFS application will make available.

The next file to put in place is the Lems monitoring file, which will appear as follows:

`/etc/cluster/nfs/lems.local.xml`

```

<?xml version="1.0"?>
<lems_config>
  <globals modules="/sbin/cluster/lems/modules"
           programs="/sbin/cluster/lems/programs"
           logs="/var/log/cluster/lems"
  />

  <check>
    <name>flag_check</name>
    <type>internal</type>
    <module>flag_check nfs</module>
    <interval>5</interval>
    <action_list>
      <action rc="0" action="NOP"/>
      <action rc="1" action="%RCDATA%"/>
      <action rc="2" action="ABORT"/>
    </action_list>
  </check>
  <check>
    <name>nfsd</name>
    <type>internal</type>
    <module>procmon /etc/cluster/nfs/nfsd.xml</module>
    <interval>10</interval>
    <action_list>
      <action rc="0" action="NOP"/>
      <action rc="1" action="STOP"/>
      <action rc="2" action="FAILOVER"/>
    </action_list>
  </check>

  <check>
    <name>fsmonitor</name>
    <type>internal</type>
    <module>fsmon nfs</module>
    <interval>10</interval>
    <action_list>
      <action rc="0" action="NOP"/>
      <action rc="1" action="PAUSE 30"/>
      <action rc="2" action="STOP"/>
      <action rc="3" action="FAILOVER"/>
      <action rc="10" action="PAUSE 60"/>
    </action_list>
  </check>

```

```
        </check>
</lems_config>
```

Finally the configuration file for the process monitor should be created:

/etc/cluster/nfs/nfsd.xml

```
<?xml version="1.0"?>
<procmon>
  <global>
    <logdir>/var/log/cluster</logdir>
    <restarts>3</restarts>
    <resetwindow>3600</resetwindow>
    <restartcmd>/usr/local/cluster/nfs_restart</restartcmd>
  </global>
  <process>
    <label>NFS Daemon</label>
    <process_string>nfsd</process_string>
    <min_count>1</min_count>
    <max_count>40</max_count>
  </process>
</procmon>
```

Now that all necessary configuration files are in place, the next step is to create the volume group and file system to make available and then mount it. In this case the volume group will be called "**nfsvg**" with a single file system "/nfsdata".

The file system here is an "**ext3**" file system, though "xfs", "reiserfs" or "jfs" could be used if necessary. In this instance a partition "/dev/sda3" on both machines was used to create the necessary volume group - please change as appropriate in the commands below for your environment.

The following commands were repeated on both machines to create a volume group "nfsvg":

```
# pvcreate /dev/sda3
# vgcreate nfsvg /dev/sda3
# vgscan
# vgchange -a y nfsvg
# lvcreate -L 256 -n lv01 nfsvg
# mkdir /nfsdata
# chmod 775 /nfsdata
```

Once the above have been run on both machines the following commands need only be entered on the machine where the configuration files from above have been installed - this machine will be used to initiate the application.

```
# mkfs -t ext3 /dev/nfsvg/lv01
```

For the example application the file system export is "/nfsdata", so this is mounted, permissions set and the contents to export copied in. Obviously this will differ for each administrators' configuration.

```
# mount /dev/nfsvg/lv01 /nfsdata
# chmod 775 /nfsdata
# cp -R /data/to/export /nfsdata
```

At this point the application can be built and the data synchronised on the remote node:

```
# clbuildapp -A nfs --sync
```

The above command will allocate all required resources and synchronise the application data. Once completed the "/nfsdata" file system will **not** be mounted - it should now only be mounted via the clustering software to ensure that the relevant DRBD meta-data is kept up to date.

To start the application on the current node simply use a command should as:

```
# clstartapp -A nfs
```

If instead the application is to be started by examining the machine load [it was configured with the "LEAST_CPU_LOAD" property], use the high level function instead:

```
# clrunapp -A nfs
```

Client-Side Configuration

Using the configuration from the client-side is very straightforward, a command such as the following [use the sample application IP address], is all that is needed:

```
# mount 192.168.1.221:/nfsdata /tmpmnt
```