ORACLE Certified Professional

ORACLE E-Business Suite 11*i* Certified Professional

Author – ***A.Kishore/Sachin***
http://appsdba.info

Step by Step NFS Configuration

Introduction

The Network File System is certainly one of the most widely used network services. Network file system (NFS) is based on the Remote procedure call. It allows the client to auto mount and therefore, transparently access the remote file systems on the network. It is used to share/map disk from one Linux System to other Linux System. NFS was developed to allow the user to access remote directory as a mapped directory. NFS is not a single program. It is a suite of related programs, which work together.

In this document we will export the file system from the linux1.com (IP address 192.168.114.231) host and mount it on linux2.com (IP address 192.168.114.232).

Prerequisites

(1) Verify NFS daemon Service

Here we assume that the NFS service daemon is already installed on our system, including portmap daemon on which NFS setup depends. One more thing, our system needs to support the NFS file system.For this we need to issue the following command:

$ cat /proc/filesystems

Another way to check if NFS is functioning is to use following "rpcinfo" command.

$ rpcinfo -p

We should get a response/output

(2) (a) Server export file

All NFS server exports need to be defined in /etc/exports file.

Most common exports options:

/home/nfs/ 192.168.114.232(rw,sync)	export /home/nfs directory for host with IP 192.168.114.232 with read, write permissions, and synchronized mode
/home/nfs/ 192.168.114.232 (ro,sync)	export /home/nfs directory for network 192.168.114.232 netmask 255.255.255.0 with read only permissions and synchronized mode
/home/nfs/ 192.168.114.232 (rw,sync) 192.168.114.233 (ro,sync)	export /home/nfs directory for host with IP 192.168.114.232 with read, write permissions, synchronized mode, and also export /home/nfs directory for hosts with IP 192.168.114.233 with read only permissions and synchronized mode
/home/nfs/ 192.168.114.232 (rw,sync,no_root_squash)	export /home/nfs directory for host with IP 192.168.114.232 with read, write permissions, synchronized mode and the remote root user will be treated as a root and will be able to change any file and directory.
/home/nfs/ *(ro,sync)	export /home/nfs directory for any host with a read only permission and synchronized mode
/home/nfs/ *.linux1.com(ro,sync)	export /home/nfs directory for any host within linux1.com domain with a read only permission and synchronized mode
/home/nfs/ linux2.com (rw, sync)	export /home/nfs directory for hostname linux2.com with read, write permissions and synchronized mode

(b) Edit exports file

Open up text editor, for example, vi editor and edit /etc/exports file and add line /home/nfs/ *(ro,sync) to export /home/nfs directory for any host with read only permissions.

/home/nfs/ *(rw,sync)
We needto be sure that the directory you export by NFS exists. You can also create a file inside the /home/nfs directory which will help us troubleshoot once we mount this file system remotely.
$ touch /home/nfs/test_file

(c) Restart NFS daemon

Once we edited /etc/exports file, we need to restart NFS service to apply changes in the /etc/exports file. Depending on our Linux distribution, the restarting of NFS may differ.
Debian users:
/etc/init.d/nfs-kernel-server restart
Redhat users
/etc/init.d/nfs restart
If you later decide to add more NFS exports to the /etc/exports file, you will need to either restart NFS service or run command exportfs:
exportfs -ra

3. Mount remote file system on client

First we need to create a mount point:
mkdir /home/nfs_local
If we are sure that the NFS client and mount point are ready, we can run the mount command to mount exported NFS remote file system:
mount 192.168.114.231:/home/nfs /home/nfs_local
here we mounted /home/nfs from 192.168.114.231 to local machine.

In case if we need to specify a type of the filesystem we can do this as shown below:
mount -t nfs 192.168.114.231:/home/nfs /home/nfs_local
We may get error message as shown below:
mount: mount to NFS server failed: timed out (retrying).
This may mean that our server supports higher versions of nfs and therefore we need to pass one extra argument to our nfs client. In this example we use nfs version 3:
mount -t nfs -o nfsvers=3 192.168.114.231:/home/nfs /home/nfs_local

Now we should be able to see that the file system is mounted.

Please Note: The mount command reports that the filesystem is mounted as "read and write", although we can see that it provides a "read only" permission.

4. Configure automount

To make this mount point completely transparent to end users, we can automount the NFS file system every time a user boots a PC, or we can also use PAM modules to mount once a user logs in with a proper username and password. In this situation we need to edit /etc/fstab to mount system automatically during a system boot. We can use vi editor and create new line like this:

192.168.114.231:/home/nfs /home/nfs_local/ nfs defaults 0 0

in /etc/fstab or

echo "192.168.114.231:/home/nfs /home/nfs_local/ nfs defaults 0 0" >> /etc/fstab

Restart NFS daemon with following commands:

$ /etc/init.d/nfs restart

$ /etc/init.d/nfslock restart