

How to mount NFS network filesystem to remote server via /etc/fstab on Linux

Author : admin



If you have a server topology part of a project where 3 (A, B, C) servers need to be used to deliver a service (one with **application server** such as Jboss / Tomcat / Apache, second just as a **Storage Server** holding a dozens of LVM-ed SSD hard drives and an **Oracle database backend** to provide data about the project) and you need to access server A (application server) to server B (the Storage "monster") one common solution is to use **NFS (Network FileSystem) Mount**.

NFS mount is considered already a bit of obsoleted technology as it is generally considered unsecure, however if SSHFS mount is not required due to initial design decision or because *both servers A and B are staying in a serious firewalled (DMZ) dedicated network* then NFS should be a good choice.

Of course to use NFS mount should always be **a carefully selected Environment Architect decision so remote NFS mount**, imply that both servers are connected via a *high-speed gigabyte network*, e.g. *network performance is calculated to be enough for application A to network storage B* two sides communication not to cause delays for systems end Users.

To test whether the *NFS server B mount* is possible on the application server A, type something like:

```
mount -t nfs -o soft,timeo=900,retrans=3,vers=3, proto=tcp remotenfssserver-host:/home/nfs-mount-data /mnt/nfs-mount-point
```

If the mount is fine **to make the mount permanent on application server host A** (in case of server

reboot), add to **/etc/fstab** end of file, following:

1.2.3.4:/application/local-application-dir-to-mount /application/remote-application-dir-to-mount nfs rw,bg,nolock,vers=3,tcp,timeo=600,rsz=32768,wsz=32768,hard,intr 1 2

If the NTFS server has a hostname you can also type hostname instead of above example sample IP 1.2.3.4, this is however not recommended as this might cause in case of DNS or Domain problems. If you want to mount with hostname (in case if storage server IP is being commonly changed due to auto-selection from a DHCP server):

server-hostA:/application/local-application-dir-to-mount /application/remote-application-dir-to-mount nfs rw,bg,nolock,vers=3,tcp,timeo=600,rsz=32768,wsz=32768,hard,intr 1 2

In above example you need to have the **/application/local-application-dir-to-mount** (dir where remote NFS folder will be mounted on server A) as well as the **/application/remote-application-dir-to-mount** Also on server Storage B server, you have to have running NFS server with firewall accessibility from server A working.

The timeou=600 (is defined in) order to make the timeout for remote NFS accessibility 1 hour in order to escape mount failures if there is some minutes network failure between server A and server B, the **rsz** and **wsz** should be fine tuned according to the files that are being red from remote NFS server and the network speed between the two in the example are due to environment architecture (e.g. to reflect the type of files that are being transferred by the 2) and the remote NFS server running version and the Linux kernel versions, these settings are for Linux **kernel branch 2.6.18.x** which as of time of writting this article is obsolete, so if you want to use the settings check for your kernel version and NTFS and google and experiment.

Anyways, if you're not sure about wsz and and rise, its perfectly safe to omit these 2 values if you're not familiar to it.

To finally check the NFS mount is fine, grep it:

```
# mount|grep -i nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
server-hostA:/application/remote-application-dir-to-mount
```

on **/application/remote-application-dir-to-mount** type nfs

(rw,bg,nolock,nfsvers=3,tcp,timeo=600,rsz=32768,wsz=32768,hard,intr,addr=1.2.3.4)

That's all enjoy :)