**What is Autofs?**

Autofs is used to automatically mount any filesystem on demand as and when you access them and not only it will mount automatically but it can automatically unmount any filesystem when not in used for a particular predefined timeout value.

**Why should I use Autofs ?**

I understand a question which generally pops up in people's mind that even putting nfs share entry in /etc/fstab makes it auto mountable so how does it differs from autofs? Well as I defined above autofs is not only/at all about mounting the share on every reboot.

Take a scenario where a nfs share is supposed to be mounted at boot but it is not accessibe due to some reasons which will lead to delayed startup or even there might be a case when your OS will not at all boot up.

**Advantages of AutoFS**

* Shares are accessed automatically and transparently when a user tries to access any files or directories under the designated mount point of the remote filesystem to be mounted.
* Booting time is significantly reduced because no mounting is done at boot time.
* Network access and efficiency are improved by reducing the number of permanently active mount points.
* Failed mount requests can be reduced by designating alternate servers as the source of a filesystem.

In this article I will show you the configuration steps for autofs using NFS share

We we have two servers namely nfsserver and nfsclient. Lets start configuring a NFS share on nfsserver  
  
**On nfsserver**  
/work   nfsclient(ro)  
/myraid nfsclient(ro)

Since this article is all about autofs I am not using much permission on the exports. These are two simple shares which I am sharing with nfsclient

Restart the nfs services or reload the shares  
# /etc/init.d/nfs restart  
Shutting down NFS daemon:                                  [  OK  ]  
Shutting down NFS mountd:                                  [  OK  ]  
Shutting down NFS quotas:                                  [  OK  ]  
Shutting down NFS services:                                [  OK  ]  
Shutting down RPC idmapd:                                  [  OK  ]  
Starting NFS services:                                     [  OK  ]  
Starting NFS quotas:                                       [  OK  ]  
Starting NFS mountd:                                       [  OK  ]  
Starting NFS daemon:                                       [  OK  ]  
Starting RPC idmapd:                                       [  OK  ]  
**NOTE:** Do not run this command on production environment if anyone is accessing any of the share as it will disrupt the traffic for few seconds.  
  
Genreally we reload the nfs share using the below command  
# exportfs -r

The above command reexport all directories, synchronizing /var/lib/nfs/etab with /etc/exports. This option removes entries in var/lib/nfs/etab  which  have  been  deleted from /etc/exports,  and  removes  any  entries from the kernel export table which are no longer valid.

To see all the shares and the applied permissions  
# exportfs -v  
/work           192.168.1.12(ro,wdelay,root\_squash,no\_subtree\_check)  
/myraid         192.168.1.12(ro,wdelay,root\_squash,no\_subtree\_check)  
Well our server side work is done lets roll over to nfsclient  
  
**NOTE:** For this article I have disabled my iptables and selinux.  
  
In case you want to use iptables insert the below rule into your INPUT chain  
# iptables -I INPUT -s 192.168.1.0/24 -p tcp --dport 2049 -j ACCEPT

**On nfsclient**  
This is the place where the magic will happen. But for that we will have to configure autofs.

**auto.master file**

The primary configuration file for the automounter is /etc/auto.master, also referred to as the master map. The master map lists autofs-controlled mount points on the system, and their corresponding configuration files or network sources known as automount maps. The format of the master map is as shown below

*mount-point map-name options*  
For our case  
# vi /etc/auto.master  
(Create a new line in between and add as shown below)  
/mnt    /etc/auto.misc  --timeout=20  
**Description**

Using the above line we are giving the default mount point as /mnt and redirecting to different file i.e. auto.master file to look into /etc/auto.misc for further information about the shares and server details we want to mount on /mnt. And lastly we have provided and extra option providing a timeout value to unmount the share in case or any inactivity for 20seconds.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **mount-point** | The autofs mount point, /mnt, for example. |
| **map-name** | The name of a map source which contains a list of mount points, and the file system location from which those mount points should be mounted. |
| **options** | If supplied, these will apply to all entries in the given map provided they don't themselves have options specified. This behavior is different from autofs version 4 where options were cumulative. This has been changed to implement mixed environment compatibility. |

**auto.misc file**

The general format of maps is similar to the master map, however the "options" appear between the mount point and the location instead of at the end of the entry as in the master map:  
mount-point   [options]   location

|  |  |
| --- | --- |
| **Value** | **Description** |
| **mount-point** | This refers to the autofs mount point. This can be a single directory name for an indirect mount or the full path of the mount point for direct mounts. Each direct and indirect map entry key (mount-point above) may be followed by a space separated list of offset directories (sub directory names each beginning with a "/") making them what is known as a multi-mount entry. |
| **options** | Whenever supplied, these are the mount options for the map entries that do not specify their own options. |
| **location** | This refers to the file system location such as a local file system path (preceded with the Sun map format escape character ":" for map names beginning with "/"), an NFS file system or other valid file system location. |

Enough with the explanation next edit auto.misc file  
# vi /etc/auto.misc  
Add a new line in the last as shown below  
deepak          -fstype=nfs             192.168.1.11:/work  
raid            -fstype=nfs             192.168.1.11:/myraid  
Save and exit the file  
  
**Description**

Here we have defined two different mount points "deepak" and "raid" for their respective shares. Now next time as soon as you try to access these shares using "cd" command the shares will be automatically mounted and when you are done just come out of the share path and after 20 seconds the share would be unmounted. We have specified the filesystem type for both the shares along with the original share location on our NFS server.

Its time to reload our autofs services  
  
Verify if autofs is running  
# /etc/init.d/autofs status  
automount (pid 5362) is running...  
# /etc/init.d/autofs reload  
Reloading maps  
Make sure the NFS shares are not already mounted  
# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup00-LogVol00  
                       11G  2.5G  7.6G  25% /  
/dev/sda1              99M   12M   83M  13% /boot  
tmpfs                 506M     0  506M   0% /dev/shm  
So none of the NFS shares are mounted still lets try to access them  
[root@nfsclient ~]# cd /mnt/deepak  
  
[root@nfsclient deepak]# ls  
deep  myfile.txt

As you see the files under "deepak" share on NFS server are accessible from the nfsclient just by doing "cd" to the share

Similarly you can try to access another share i.e. "raid"

**IMPORTANT NOTE :** If you do not know the names of the share used under auto.misc then you won't be able to access the NFS shares as be defualt /mnt would be shown as empty. The shares are mounted only when you "cd" into the share directory.

You can use the watch command to verify if the shares automatically gets unmounted.  
# watch mount  
Every 2.0s: mount                                       Fri Sep 26 21:16:56 2014  
  
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)  
proc on /proc type proc (rw)  
sysfs on /sys type sysfs (rw)  
devpts on /dev/pts type devpts (rw,gid=5,mode=620)  
/dev/sda1 on /boot type ext3 (rw)  
tmpfs on /dev/shm type tmpfs (rw)  
none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)  
sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)  
192.168.1.11:/work on /mnt/deepak type nfs (rw,addr=192.168.1.11)  
After few seconds the mount point "work" is automatically unmounted as you can see below  
# watch mount  
Every 2.0s: mount                                       Fri Sep 26 21:17:18 2014  
  
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)  
proc on /proc type proc (rw)  
sysfs on /sys type sysfs (rw)  
devpts on /dev/pts type devpts (rw,gid=5,mode=620)  
/dev/sda1 on /boot type ext3 (rw)  
tmpfs on /dev/shm type tmpfs (rw)  
none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)  
sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)