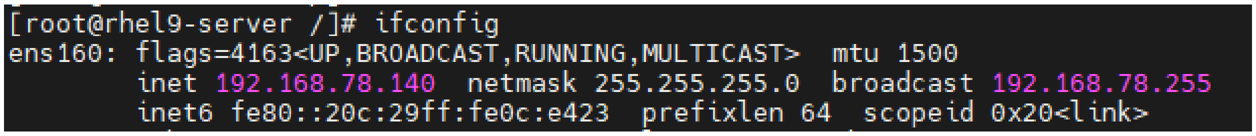
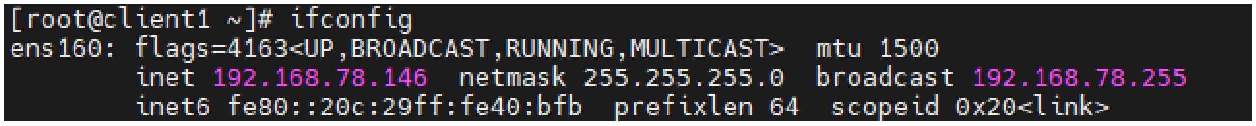
AutoFS NFS

For autofs lab setup, I am taking one server & one client as shown-



```
[root@rhel9-server /]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.78.140  netmask 255.255.255.0  broadcast 192.168.78.255
        inet6 fe80::20c:29ff:fe0c:e423  prefixlen 64  scopeid 0x20<link>
```



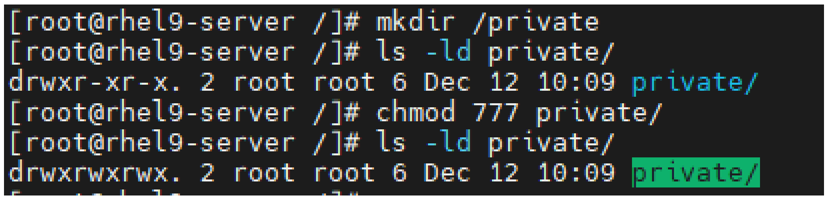
```
[root@client1 ~]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.78.146  netmask 255.255.255.0  broadcast 192.168.78.255
        inet6 fe80::20c:29ff:fe40:bfb  prefixlen 64  scopeid 0x20<link>
```

At Server Side:

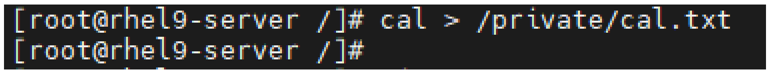
1. Next, we will install nfs packages in server using below command-

```
yum install nfs* -y
```

2. We will create first directory “private” with 777 permission & add some content in it to share with client-

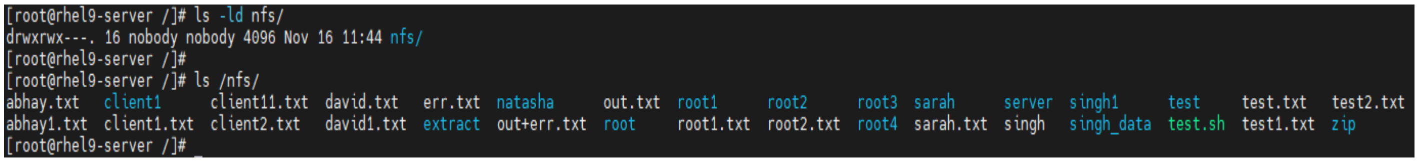


```
[root@rhel9-server /]# mkdir /private
[root@rhel9-server /]# ls -ld private/
drwxr-xr-x. 2 root root 6 Dec 12 10:09 private/
[root@rhel9-server /]# chmod 777 private/
[root@rhel9-server /]# ls -ld private/
drwxrwxrwx. 2 root root 6 Dec 12 10:09 private/
```



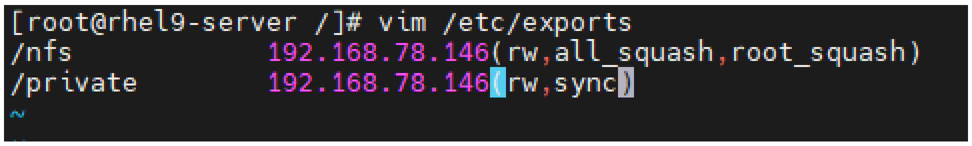
```
[root@rhel9-server /]# cal > /private/cal.txt
[root@rhel9-server /]#
```

3. Second directory “nfs” with 770 permission-



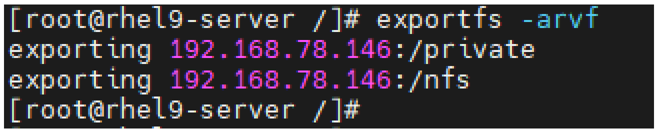
```
[root@rhel9-server /]# ls -ld nfs/
drwxrwx---. 16 nobody nobody 4096 Nov 16 11:44 nfs/
[root@rhel9-server /]#
[root@rhel9-server /]# ls /nfs/
abhay.txt   client1      client11.txt  david.txt   err.txt  natasha      out.txt  root1      root2      root3  sarah      server  singh1      test     test.txt   test2.txt
abhay1.txt  client1.txt  client2.txt   david1.txt  extract  out+err.txt  root     root1.txt  root2.txt  root4  sarah.txt  singh   singh_data  test.sh  test1.txt  zip
[root@rhel9-server /]# _
```

4. Now we will add entries for these two directories in /etc/exports-



```
[root@rhel9-server /]# vim /etc/exports
/nfs            192.168.78.146(rw,all_squash,root_squash)
/private        192.168.78.146(rw,sync)
~
```

5. Then we will exports these two share-



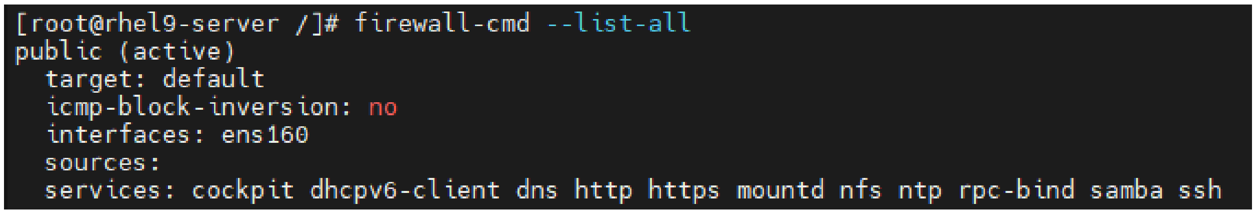
```
[root@rhel9-server /]# exportfs -arvf
exporting 192.168.78.146:/private
exporting 192.168.78.146:/nfs
[root@rhel9-server /]#
```

6. Next, we will add nfs, rpc-bind & mountd service in firewall as –

firewall-cmd --permanent --add-service={nfs,rpc-bind,mountd}

firewall-cmd --reload

7. We can verify these as-

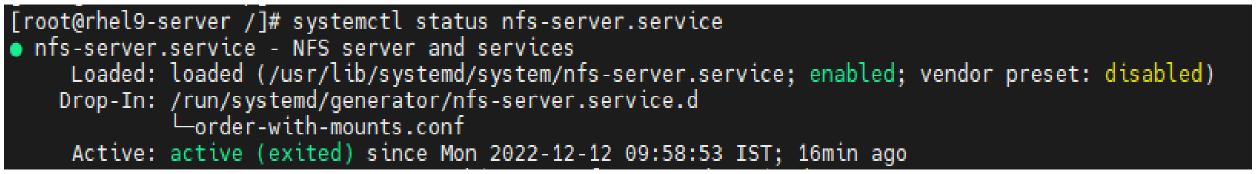


```
[root@rhel9-server /]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client dns http https mountd nfs ntp rpc-bind samba ssh
```

8. Start & enable nfs-server service-

systemctl enable nfs --now

9. Check nfs-server service is up & running-



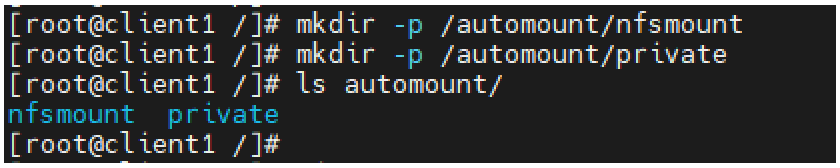
```
[root@rhel9-server /]# systemctl status nfs-server.service
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor preset: disabled)
    Drop-In: /run/systemd/generator/nfs-server.service.d
             └─order-with-mounts.conf
     Active: active (exited) since Mon 2022-12-12 09:58:53 IST; 16min ago
```

At Client Side:

1. First, we will install few packages at client-

yum install nfs-utils nfs4-acl-tools autofs -y

2. Next, we will create two directories inside parent directory as shown-

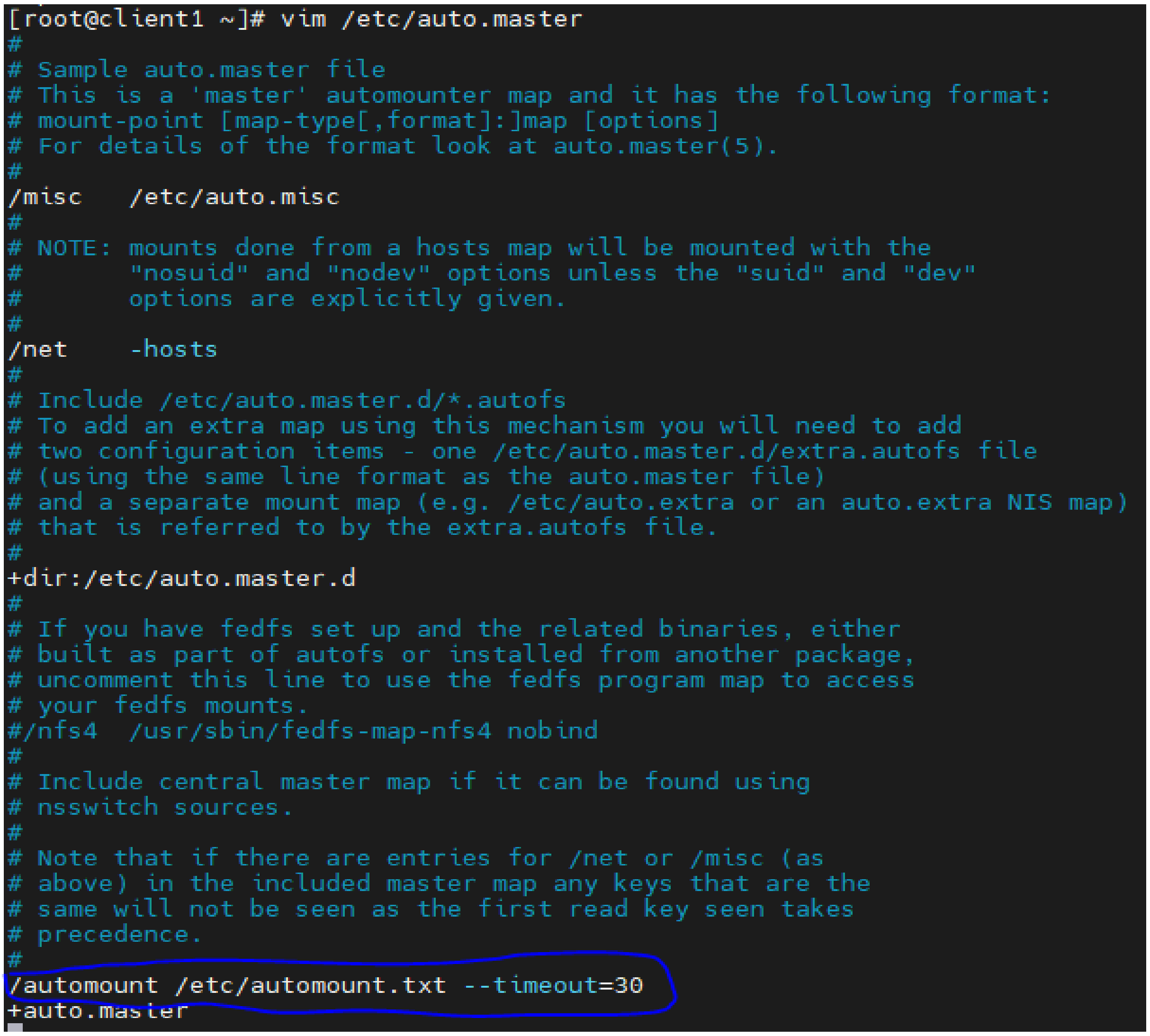


```
[root@client1 /]# mkdir -p /automount/nfsmount
[root@client1 /]# mkdir -p /automount/private
[root@client1 /]# ls automount/
nfsmount  private
[root@client1 /]#
```

Here automount directory (parent) will be used for autofs. Shares will be mounted inside this directory.

nfsmount will be used for nfs share & private will be used for private share.

3. To create autofs, we will edit auto.master file inside /etc as highlighed-

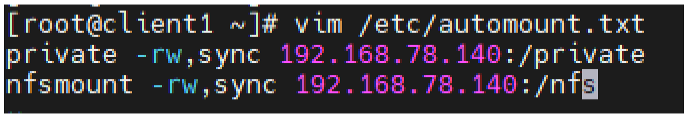


```
[root@client1 ~]# vim /etc/auto.master
#
# Sample auto.master file
# This is a 'master' automounter map and it has the following format:
# mount-point [map-type[,format]:]map [options]
# For details of the format look at auto.master(5).
#
/misc   /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       "nosuid" and "nodev" options unless the "suid" and "dev"
#       options are explicitly given.
#
/net    -hosts
#
# Include /etc/auto.master.d/*.autofs
# To add an extra map using this mechanism you will need to add
# two configuration items - one /etc/auto.master.d/extra.autofs file
# (using the same line format as the auto.master file)
# and a separate mount map (e.g. /etc/auto.extra or an auto.extra NIS map)
# that is referred to by the extra.autofs file.
#
+dir:/etc/auto.master.d
#
# If you have fedfs set up and the related binaries, either
# built as part of autofs or installed from another package,
# uncomment this line to use the fedfs program map to access
# your fedfs mounts.
#/nfs4  /usr/sbin/fedfs-map-nfs4 nobind
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
/automount /etc/automount.txt --timeout=30
+auto.master
```

Here, we will mention that parent directory used for autofs at client side. It will be used to mount shared directories.

Timeout is 30 seconds. After this time duration, shared directory will be unmount automatically if not in use (If we are outside this created child directory at client i.e nfsmount & private).

4. Next, we will create automount.txt in /etc. It will contain detail of exported file system-

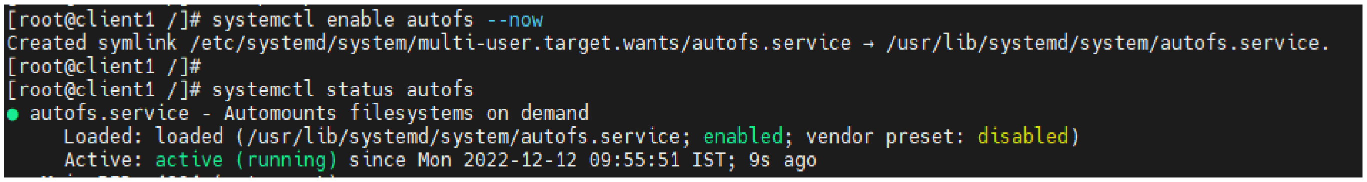


```
[root@client1 ~]# vim /etc/automount.txt
private -rw,sync 192.168.78.140:/private
nfsmount -rw,sync 192.168.78.140:/nfs
```

Here, private & nfsmount are client’s child directories inside parent directory automount on which corresponding server’s share will be mounted.

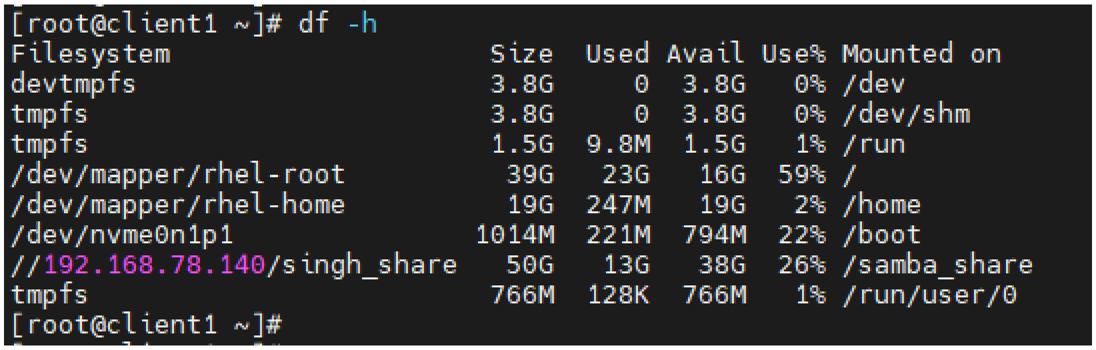
After that, we provide server’s shared directories location.

5. Start & enable autofs service & verify it-



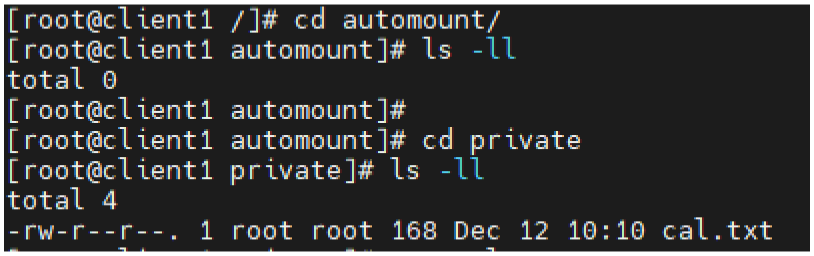
```
[root@client1 /]# systemctl enable autofs --now
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/lib/systemd/system/autofs.service.
[root@client1 /]#
[root@client1 /]# systemctl status autofs
● autofs.service - Automounts filesystems on demand
     Loaded: loaded (/usr/lib/systemd/system/autofs.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2022-12-12 09:55:51 IST; 9s ago
```

6. Now, we will verify for current mounted directory on this client before mounting these new shares-



```
[root@client1 ~]# df -h
Filesystem                    Size  Used Avail Use% Mounted on
devtmpfs                      3.8G     0  3.8G   0% /dev
tmpfs                         3.8G     0  3.8G   0% /dev/shm
tmpfs                         1.5G  9.8M  1.5G   1% /run
/dev/mapper/rhel-root          39G   23G   16G  59% /
/dev/mapper/rhel-home          19G  247M   19G   2% /home
/dev/nvme0n1p1               1014M  221M  794M  22% /boot
//192.168.78.140/singh_share   50G   13G   38G  26% /samba_share
tmpfs                         766M  128K  766M   1% /run/user/0
[root@client1 ~]#
```

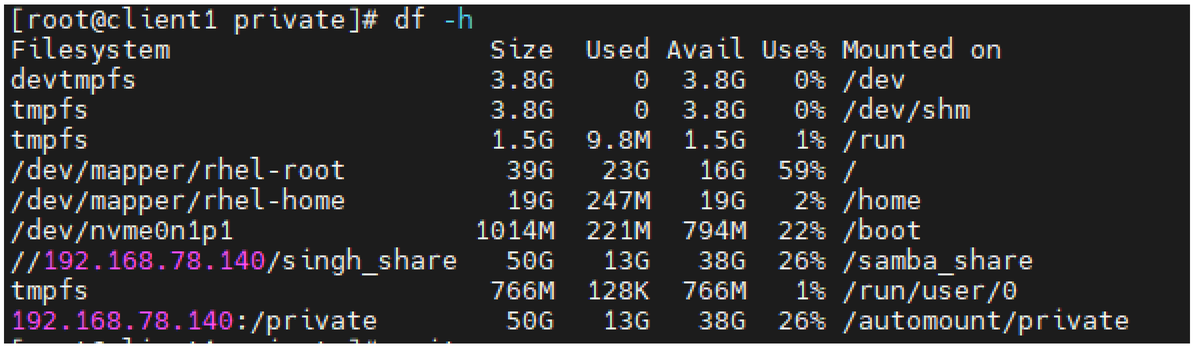
7. Next, go to parent directory automount & go inside child directories to see whether we are able to see shared files/directories by server-



```
[root@client1 /]# cd automount/
[root@client1 automount]# ls -ll
total 0
[root@client1 automount]#
[root@client1 automount]# cd private
[root@client1 private]# ls -ll
total 4
-rw-r--r--. 1 root root 168 Dec 12 10:10 cal.txt
```

Here, cal.txt is visible which was created at server side inside their private directory.

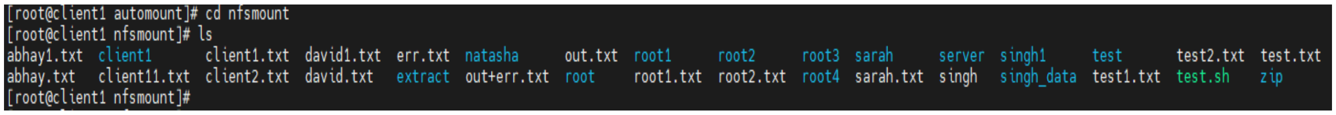
8. Now, we will verify this mount point at client-



```
[root@client1 private]# df -h
Filesystem                    Size  Used Avail Use% Mounted on
devtmpfs                      3.8G     0  3.8G   0% /dev
tmpfs                         3.8G     0  3.8G   0% /dev/shm
tmpfs                         1.5G  9.8M  1.5G   1% /run
/dev/mapper/rhel-root          39G   23G   16G  59% /
/dev/mapper/rhel-home          19G  247M   19G   2% /home
/dev/nvme0n1p1               1014M  221M  794M  22% /boot
//192.168.78.140/singh_share   50G   13G   38G  26% /samba_share
tmpfs                         766M  128K  766M   1% /run/user/0
192.168.78.140:/private        50G   13G   38G  26% /automount/private
```

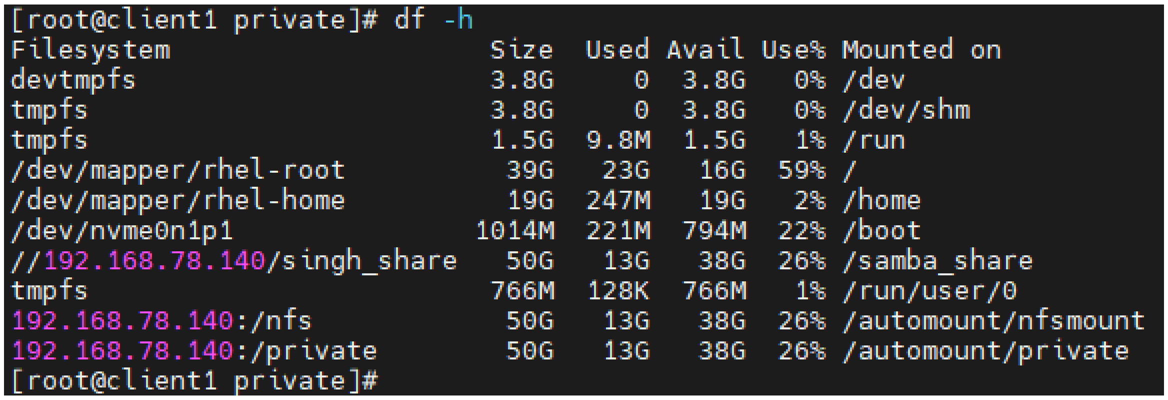
Here it is showing as mounted in last line.

9. In the same way, we will check for another share-



```
[root@client1 automount]# cd nfsmount
[root@client1 nfsmount]# ls
abhay1.txt  client1      client1.txt  david1.txt  err.txt  natasha      out.txt  root1      root2      root3  sarah      server  singh1      test       test2.txt  test.txt
abhay.txt   client11.txt client2.txt  david.txt   extract  out+err.txt  root     root1.txt  root2.txt  root4  sarah.txt  singh   singh_data  test1.txt  test.sh    zip
[root@client1 nfsmount]#
```

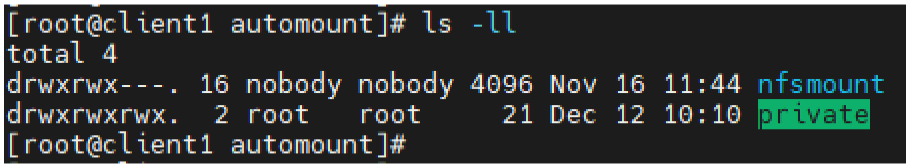
10. Now verify all new mount points-



```
[root@client1 private]# df -h
Filesystem                     Size  Used Avail Use% Mounted on
devtmpfs                       3.8G     0  3.8G   0% /dev
tmpfs                          3.8G     0  3.8G   0% /dev/shm
tmpfs                          1.5G  9.8M  1.5G   1% /run
/dev/mapper/rhel-root           39G   23G   16G  59% /
/dev/mapper/rhel-home           19G  247M   19G   2% /home
/dev/nvme0n1p1                1014M  221M  794M  22% /boot
//192.168.78.140/singh_share    50G   13G   38G  26% /samba_share
tmpfs                          766M  128K  766M   1% /run/user/0
192.168.78.140:/nfs             50G   13G   38G  26% /automount/nfsmount
192.168.78.140:/private         50G   13G   38G  26% /automount/private
[root@client1 private]#
```

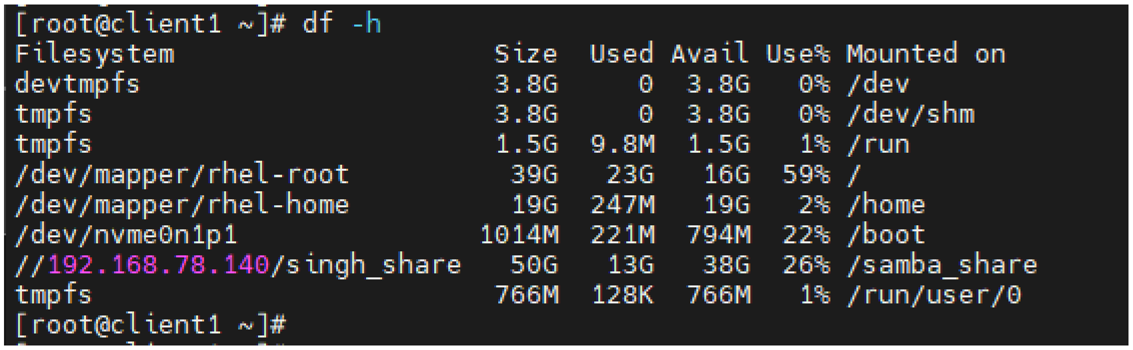
Last two lines shows our required mount point.

11. If we long list parent automount directory, we will see both child directories which with server's shared directory permission-



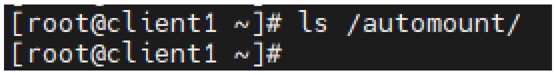
```
[root@client1 automount]# ls -ll
total 4
drwxrwx---. 16 nobody nobody 4096 Nov 16 11:44 nfsmount
drwxrwxrwx.  2 root   root     21 Dec 12 10:10 private
[root@client1 automount]#
```

12. After 30 seconds of timeout duration, these mounted directories will be unmounted if not in use-



```
[root@client1 ~]# df -h
Filesystem                     Size  Used Avail Use% Mounted on
devtmpfs                       3.8G     0  3.8G   0% /dev
tmpfs                          3.8G     0  3.8G   0% /dev/shm
tmpfs                          1.5G  9.8M  1.5G   1% /run
/dev/mapper/rhel-root           39G   23G   16G  59% /
/dev/mapper/rhel-home           19G  247M   19G   2% /home
/dev/nvme0n1p1                1014M  221M  794M  22% /boot
//192.168.78.140/singh_share    50G   13G   38G  26% /samba_share
tmpfs                          766M  128K  766M   1% /run/user/0
[root@client1 ~]#
```

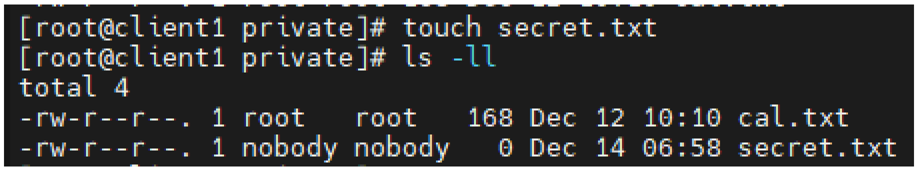
13. We can verify the same using long listing parent automount directory-



```
[root@client1 ~]# ls /automount/
[root@client1 ~]#
```

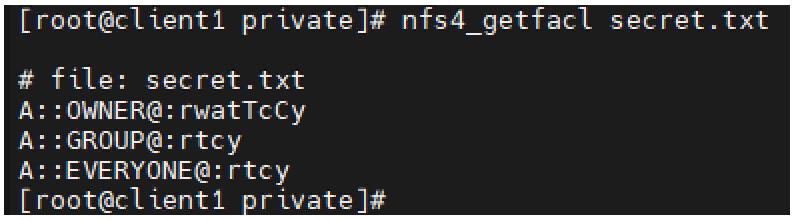
ACL on NFS-Share:

1. First create any file/dir on which we will play ACL-



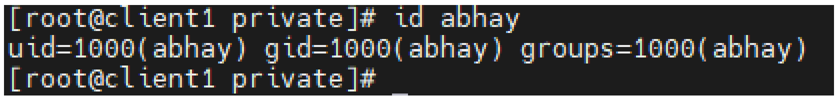
```
[root@client1 private]# touch secret.txt
[root@client1 private]# ls -ll
total 4
-rw-r--r--. 1 root   root   168 Dec 12 10:10 cal.txt
-rw-r--r--. 1 nobody nobody   0 Dec 14 06:58 secret.txt
```

2. Check current ACL applied on this secret.txt file-



```
[root@client1 private]# nfs4_getfacl secret.txt

# file: secret.txt
A::OWNER@:rwatTcCy
A::GROUP@:rtcy
A::EVERYONE@:rtcy
[root@client1 private]#
```

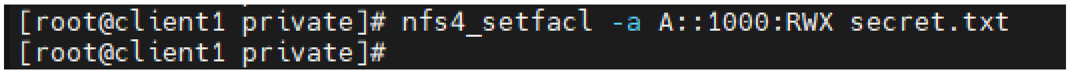
3. We want to give read, write & execute access to abhay user at this file. So first check its user id-



```
[root@client1 private]# id abhay
uid=1000(abhay) gid=1000(abhay) groups=1000(abhay)
[root@client1 private]# _
```

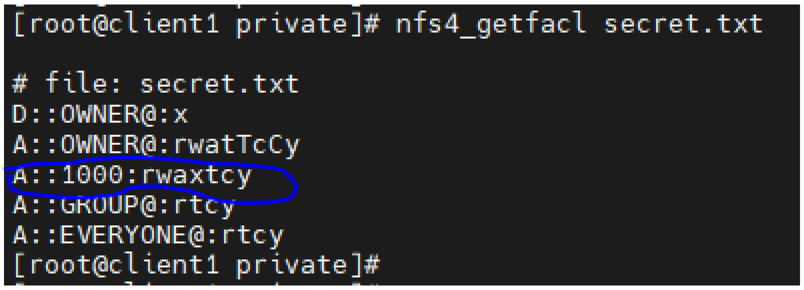
Initially, we will not be able to edit this file by abhay user as it has read-only access.

4. Next, set ACL permission for abhay user-

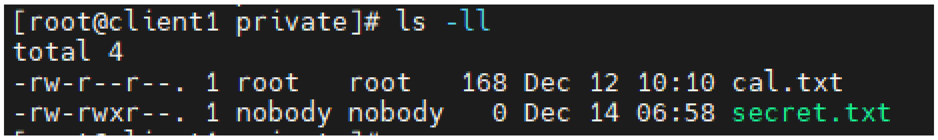


```
[root@client1 private]# nfs4_setfacl -a A::1000:RWX secret.txt
[root@client1 private]#
```

5. Verify this ACL-



```
[root@client1 private]# nfs4_getfacl secret.txt

# file: secret.txt
D::OWNER@:x
A::OWNER@:rwatTcCy
A::1000:rwaxtcy
A::GROUP@:rtcy
A::EVERYONE@:rtcy
[root@client1 private]#
```

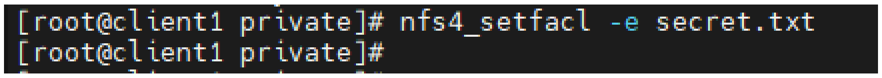


```
[root@client1 private]# ls -ll
total 4
-rw-r--r--. 1 root   root   168 Dec 12 10:10 cal.txt
-rw-rwxr--. 1 nobody nobody   0 Dec 14 06:58 secret.txt
```

Now Abhay will be able to do anything with this file.

6. To remove ACL, we will use either of two ways-

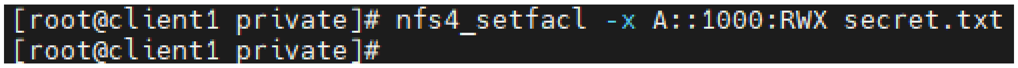
(i)



```
[root@client1 private]# nfs4_setfacl -e secret.txt
[root@client1 private]#
```

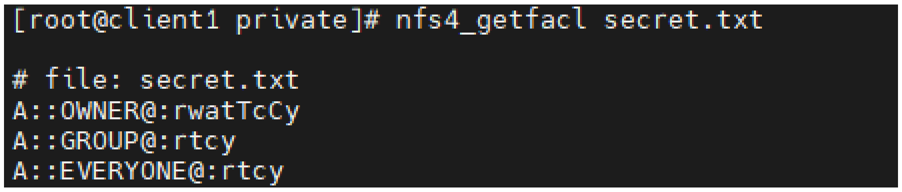
It will get open in vim editor. We will simply delete that line.

(ii)



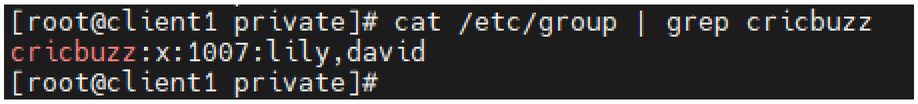
```
[root@client1 private]# nfs4_setfacl -x A::1000:RWX secret.txt
[root@client1 private]#
```

7. Verify the same-



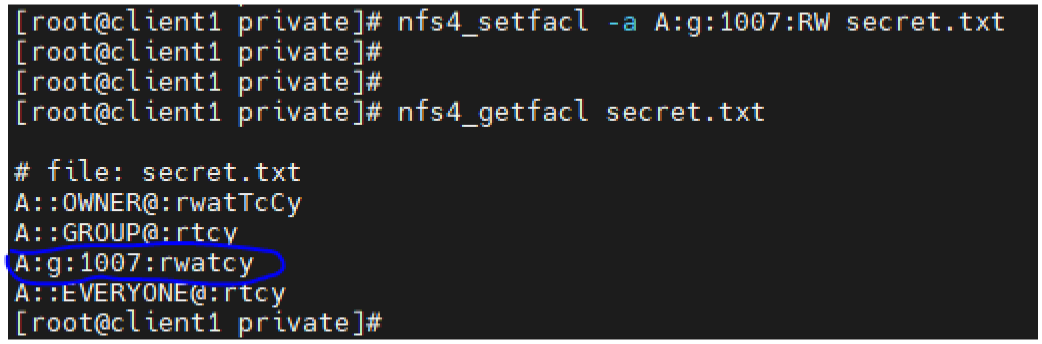
```
[root@client1 private]# nfs4_getfacl secret.txt

# file: secret.txt
A::OWNER@:rwatTcCy
A::GROUP@:rtcy
A::EVERYONE@:rtcy
```

8. If we want to give read, write access to cricbuzz group, then first check group id-



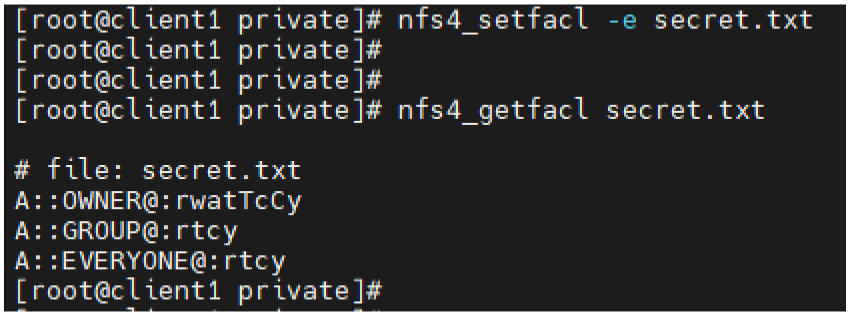
```
[root@client1 private]# cat /etc/group | grep cricbuzz
cricbuzz:x:1007:lily,david
[root@client1 private]#
```

9. Next, add ACL for this group & verify-



```
[root@client1 private]# nfs4_setfacl -a A:g:1007:RW secret.txt
[root@client1 private]#
[root@client1 private]#
[root@client1 private]# nfs4_getfacl secret.txt

# file: secret.txt
A::OWNER@:rwatTcCy
A::GROUP@:rtcy
A:g:1007:rwatcy
A::EVERYONE@:rtcy
[root@client1 private]#
```

10. Now remove ACL permission from group in any of the two ways-



```
[root@client1 private]# nfs4_setfacl -e secret.txt
[root@client1 private]#
[root@client1 private]#
[root@client1 private]# nfs4_getfacl secret.txt

# file: secret.txt
A::OWNER@:rwatTcCy
A::GROUP@:rtcy
A::EVERYONE@:rtcy
[root@client1 private]#
```