NFS:-
-----
config file: /etc/exports
daemon: nfsd
Port Number: 2049

NFS requires portmap(rpcbind) - Remote procedure call, a protocal where as one program request a service
from a program located in the other computer on a network.
Bind RPC daemon: rpcbind
Port Number: 111/udp

Note:-
1.NFS version3 supports synctronous writes and is more robust at error handling than NFSv2.
2.NFS Versison3 supports 64 bit, allows clients to access more than 2GB of data. NFS2 is not supported by RHEL7.
3.NFSv4 no longer requires rpcbind services, supports ACLs and utilizes stateful operations.
4.NFSv2, NFSv3 can use UDP(stateless) or TCP(statefull), whereas NFSv4 works with TCP.
5.The mounting and locking protocols have been incorporated into NFSv4 protocol. NFSv4 also listen at 2049/tcp port.

To check export directory of a server, use below command:
showmount -e servername
The above will list the exported filesystem.

NFS uses client-server architecture. The server portion consists of physical disks containing shared filesystems and several daemons that make shared filesystem visible and available for use by client system on that network. The rpocess is normally referred to as exporting filesystem.

NFS server daemons also provide file locking and quota management on NFS exports.

NFS clients mount the exported filesystem with local mount point called NFS mount. These NFS moutnts appear as if they were on local disk. The biggest advantage of using NFS is centralized administration. However, as it is distributed network based filesystem, NFS is sensitive to network congestion. Heavy network traffic slows down NFS performance.

NFS configuration files and daemons:-

1./etc/exports:
it is a main configuration file of NFS server. This file contains all exported mount npoints or directories.
Format is as below:

```
/shareDir   HostnameOrIp(sync,sec=sys,rw)

examples:-
/sap_Images 192.168.1.95(sync,sec=sys,rw)
/sap_Images db01.alclabs.in(sync,sec=sys,rw,no_root_squash)
/pub        *.alclabs.in(sync,sec=sys,rw,all_sqash)
```

```
/pub        *(ro,insecure,all_squash)
/var/share  webclient01.alclabs.in(ro)
yumser.alclabs.in(rw,no_root_squash)
/home/joe   webser*(rw,all_squash,anonuid=1000,anongid=1000)
/app_build  buidser[0-9].acllabs.in(rw)
```

2./etc/fstab:

To mount NFS persistently, so that the mount is available across reboots, the below is the format in /etc/fstab (at client end)

```
ServIPorSerName:/ExportedFS  /localMountPt    nfs    defaults,OptionsIfAny     0 0
ex:-
sandbox.alclabs.in:/SapImages    /SapInstalls nfs    defaults,rw,hard,acl      0 0
```

3./etc/sysconfig/nfs:-

This configuration files controls NFS on which port the rpc and other services are listening.

To manage NFS services:-

```
service nfs start (systemctl start nfs-server)
service nfs stop (systemctl stop nfs-server)
service nfs restart (systemctl restart nfs-server)
service nfs reload (systemctl reload nfs-server)
service nfs status (systemctl status nfs-server)
Note:- mentioned in brackets is for RHEL7
```

To manage rpcbind daemon:-

```
service rpcbind start[stop restart status reload]
in RHEL
systemctl start/stop/restart/status rpcbind
```

To export NFS mounts in the server:-

```
showmount -e or
shwomount -e ServerNameorIP
```

Note:-

1. make sure iptables(firewalld in RHEL7) are allowing NFS and rpcbind ports (2049 and 111). Else you may need to stop iptables or allow ports in the firewall.

```
To stop iptables(firewall):
service iptables stop
or to stop firewalld
systemctl stop firewalld
Or
You can allow firewall nfs ports as below(IN RHEL7):
Firewall-cmd --add-service=nfs --permanent
Firewall-cmd --add-service=mountd --permanent
Firewall-cmd --add-serivce=rpc-bind --permanent
Firewall-cmd --reload
```

2. Check if selinux is enabled, you may need to disable the same if you dont want to configure the context. To disable:

* getenforce : to see if it is enforcing
* setenforce 0 : to make it permissive
* You can also edit /etc/selinux/config and turn it permissive or disabled. however change will take effect in the next reboot. setenforce command will allow changes immediately.

3. You might need to enabled service in the next reboot automatically to start. To do the same,
*chkconfig nfs on (in RHEL7, systemctl enable nfs-server)
*chkconfig rpcbind on (systemctl enable rpcbind)
*chkconfig iptables off (systemctl disable firewalld)

NFS daemons:-
Server Side Daemons:-
1. nfsd : the nfsd daemon which services the requests from NFS clients. This is main NFS daemon.
2. rpc.mountd: the nfs mount daemon which carries out requests received from nfsd and responsible for nfs mounts.
3. rpc.statd: this daemon interacts with rpc.lockd daemon to provide crash and recovery functions.
4. rpc.lockd: This process locks requirest that are either sent locally by the kernel or remotely by another lock daemon. rpc.lockd daemon interacts with rpc.statd daemon(status monitor) to monitor service. The rpc.statd daemon should always be started before the lockd daemon starts.
5. rpcbind: It is used to convert rpc program numbers into universal addresses.
7. rpc.idmap: used in NFSv4, to translate user and group ids to name and vice-versa.

NFS Client Daemons:-
1. rpcbind
2. rpcstatd
3. rpc.mountd

Note:-
1.make sure there is a proper name resolution exists both client and server side (if you are using /etc/hosts). If name resolution is an issue, the mount might not happen.
2.When /etc/exports file exists, the NFS server daemons will be started automatically at server boot time and exports -a command will be invoked to configure exports.
3.If you make any changes to /etc/exports file, you need to execute exportfs -a command to make those changes in effect.

There are three ways to mount an NFS exported filesystem.
1.Explicit Mount:-
Explicit mounts are usually for the short period using mount command, when there is requirement for occassional unplanned mounts. Explicit mounts does not require updating /etc/fstab file. Explicit mounts are lost when system is rebooted.
Ex:- mount -v nfs -o Options ServerNameOrIP:/ExportedFS /localMntpt
2.Pre-defined Mount:- These are specified in the /etc/fstab file. When the server gets rebooted, the NFS mount will happen automatically.
3.Automount:- Refer to the material Given.

NFS Security:-
By default, when you try to access NFS mount, the 'nobody' user credentials are used. NFS user users certain security modes for authenticating and protecting filesystem access. Below are security methods
sys:- The unix user ids and groups are used to authenticate
dh: diffi Hell Man method
krb5: uses kerboros version 5 authentication
krb5i: users kerboros ver5 with integrity checksum.

Note:-
By default, when files or directories created under nfs share, they are created with nfsnobody user as a owner. This is because by default, the share is exported by default with 'all_squash'

root_squash option:-

This is default option where as when nfs share is exported, in the client, even if root user does any read or write operation, it will have 'nfsnobody' user privileges. However you can undo the squash with no_root_squash option.

Example as below:
/sapImages *.alclabs.in(rw,sync,sec=sys,no_root_squash)
The above entry will not squash the root, and instead of getting nfsnobody, while creating files or directories, we get owner as 'root'.
all_squash: All user privileges are squashed. Entry as below:
/sapImages 192.168.1.0/24(sys,sec=sys,rw,all_squash)

Creating files or directories with particualr user id:-

This is achieved with anonuid and anongid entry and below is the example:
/sapImages 192.168.1.0/24(sync,sec=sys,rw,anonuid=777,anongid=777)
Now reexport the share with exportfs -vra command and try to access the share at client side. Observe that you get specified id when you write any file or Dir.