HARSH KASHYAP
CSE 4
101917088
hkashyap_be19@thapar.edu

A Practical activity Report submitted
for Practical Computing(UCS311)

PRACTICAL COMPUTING

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering
Patiala Campus
2020

Submitted to
Aashima Sharma

Assignment 4
NFS

Question 1

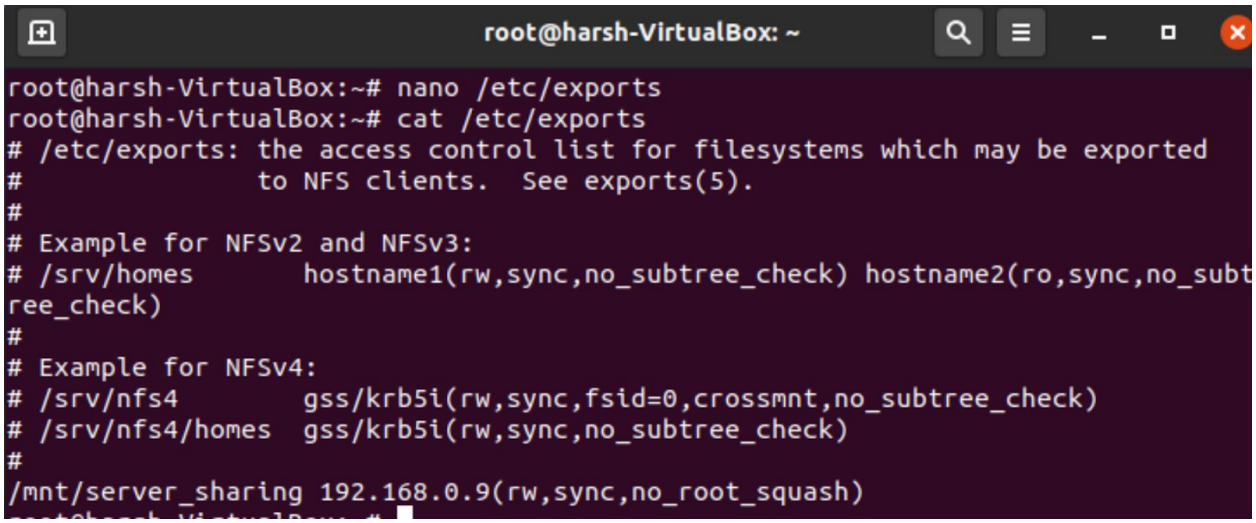
What are the different options used in /etc/exports file?

Solution -

There are many different NFS sharing options, including these:

- **rw:** Share as read-write. Provide both read and write access to client-server.
- **ro**: Share as read-only. Provide read-only permission to the client-server.
- **sync**: File data changes are made to disk immediately, which has an impact on performance but is less likely to result in data loss. On some distributions, this is the default. Sync confirms requests to the shared directory only once the changes have been committed.
- **async**: The opposite of sync; file data changes are made initially to memory. This speeds up performance but is more likely to result in data loss. On some distributions, this is the default.
- **no_subtree_check**: This option prevents the subtree checking. When a shared directory is the subdirectory of a larger file system, nfs performs scans of every directory above it, in order to verify its permissions and details. Disabling the subtree check may increase the reliability of NFS, but reduce security.
- **root_squash**: Map the root user and group account from the NFS client to the anonymous accounts, typically either the nobody account or the nfsnobody account. This phrase allows root to connect to the designated directory.
- **no_root_squash**: Map the root user and group account from the NFS client to the local root and group accounts.

/etc/exports file includes all the directories to be shared over NFS with respective client IP addresses.



Question 2

How to check available NFS share on local & remote machine.

Solution

We use showmount command to see the mount information for an NFS server.
When no options are passed, the showmount command lists the set of clients which are mounting from the host.

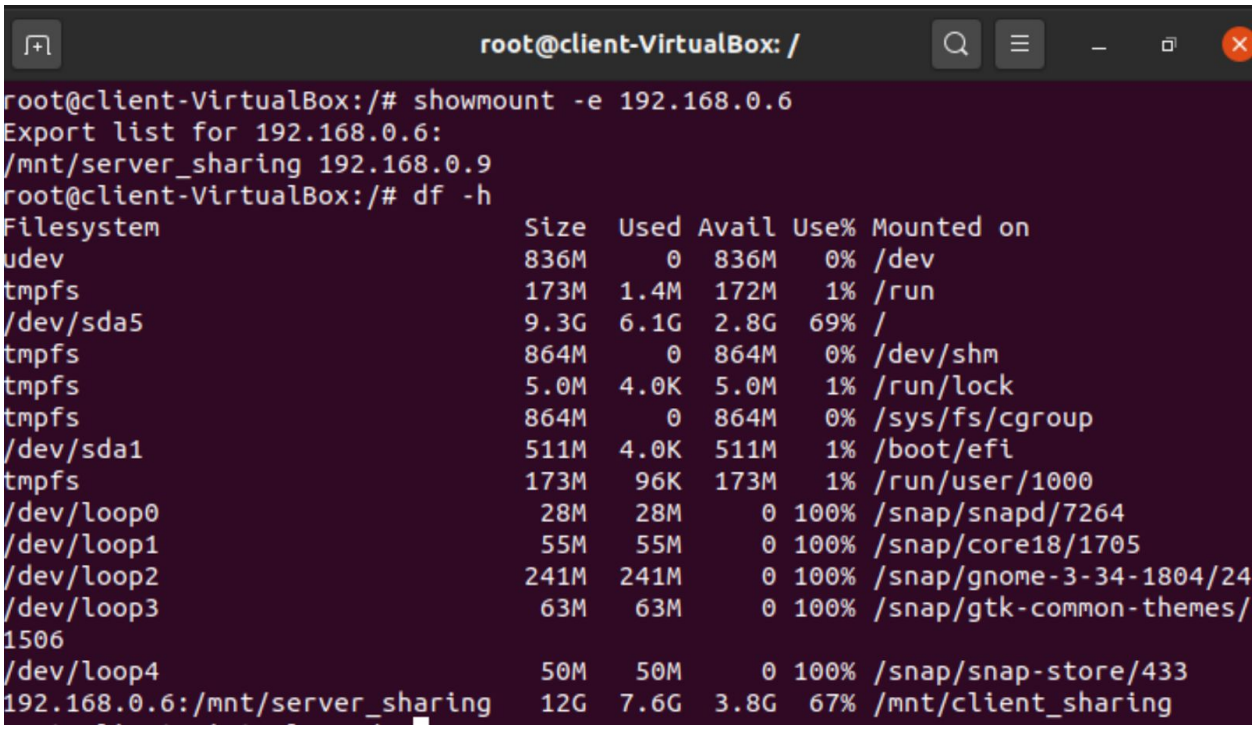
1. *Login as root*
2. *Type the following command:*

~# showmount -e

The given command will print the exported file system from the machine having the given IP address.

There is a handy command called showmount which displays all the active folder exports on an NFS server. This can be handy when trying to connect to a new NFS export from a remote machine as you can see if the export is available in the NFS server.

Run the showmount command with the server name to check which NFS exports are available. In this example, localhost is the server name. showmount -e localhost.
The output shows the available exports and the IP which they are available from.



Question 3

Show all the directories of 'etc/export' file to be reexported.

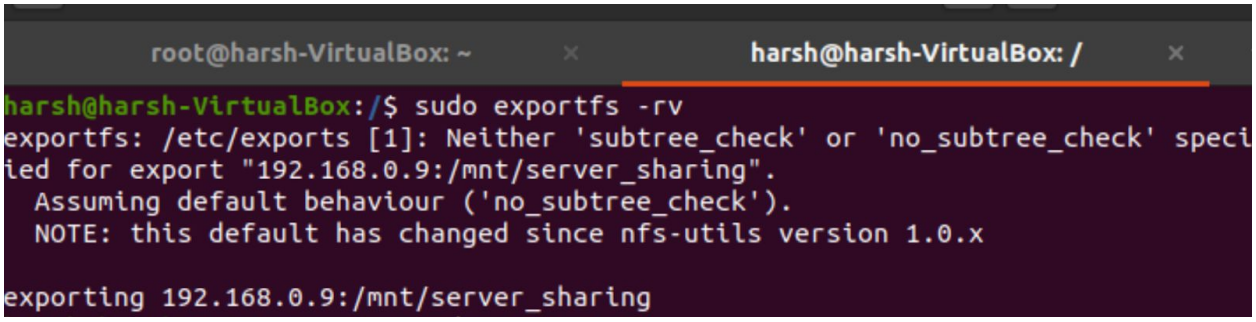
Solution

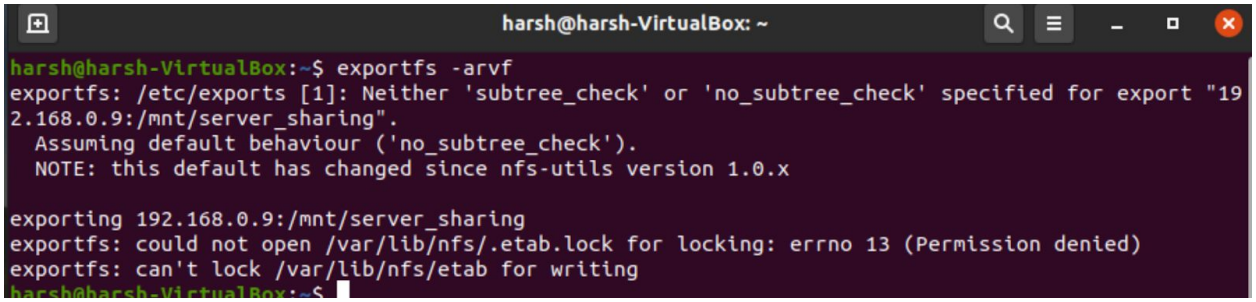
The exportfs command allows the root user to selectively export or unexport directories without restarting the NFS service. When given proper options, the exportfs command writes exported file systems to */var/lib/nfs/xtab*

1) exportfs -rv Causes all directories listed in /etc/exports to be reexported by constructing a new export list in */etc/lib/nfs/xtab*. This option effectively refreshes the export list with any changes that have been made to */etc/exports/*

2) exportfs -a Causes all directories to be exported or unexported, depending on what other options are passed to exportfs. If no other options are specified, exportfs exports all file systems specified in */etc/exports*

The command 'exportfs -r' helps us to show all the directories of 'etc/export' file to be reexported.
Inside /etc/exports file you can configure the directories you want to share with your client server Using command vim /etc/exports Using command exportfs -arvf you can check the directories being exported





Question 4

Find out the difference between Hard Mount and Soft Mount.

Solution -

Soft Mount

Whenever the client tries to access NFS shared file system, nfs-client tries to connect with nfs-server and if nfs-server is down it returns error/timeout.

$ sudo mount -o rw,soft host.nf_server.com/share_name /mnt/nfs_data

Advantage:

- Make client-side fast and responsive.
- If your NFS server is unavailable, the kernel will time out the I/O operation after a pre-configured period of time

Drawback:

- File corruption chances may be high.
- If your NFS driver caches data and the soft mount times out, your application may not know which writes to the NFS volumes were actually committed to disk.

Hard Mount

Whenever the client tries to access the NFS shared file system, nfs-client repeatedly tries to connect with NFS-server until a response is fetched. It is a good idea to use the intr option while specifying Hard Mount to allow KeyBoard Interrupt if by chance NFS server went offline for a very long time. (although after kernel v2.6, intr is default option with Hard Mount).
Hard mount is generally used for block resources like a local disk or SAN. A soft mount is usually used for network file protocols like NFS or CIFS.
$ sudo mount -o rw,hard,intr host.nf_server.com/share_name /mnt/nfs_data

Advantage: Data Integrity is Promised

Drawback: Bad Client Experience in case NFS server crashes.

Advantage of Soft Mount over Hard Mount:
If one's NFS server is unavailable, the kernel, after a pre-configured period of time, can easily time out the execution of a particular operation.

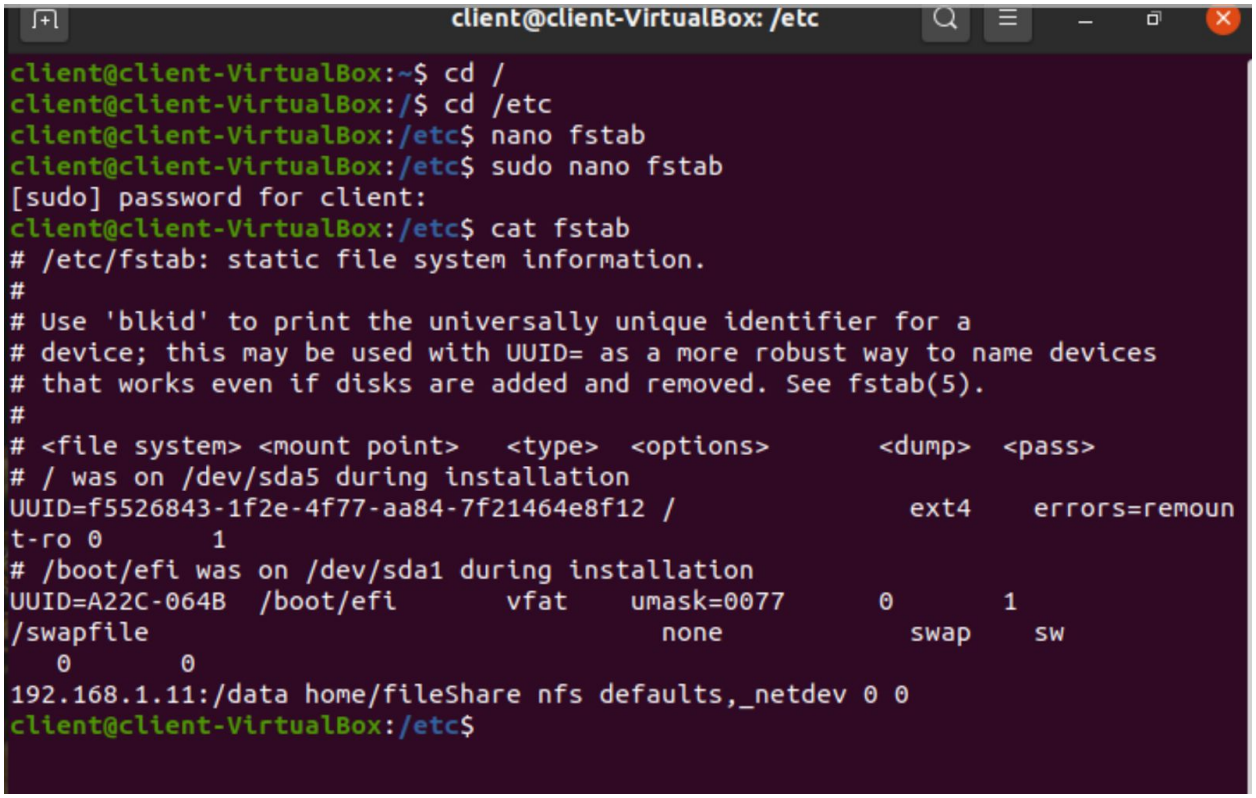
Question 5

110.168.2.10:/data is exported by NFS server and NFS share is to be added to client/etc/fstab/ file. How you will add this entry in etc/fstab file.

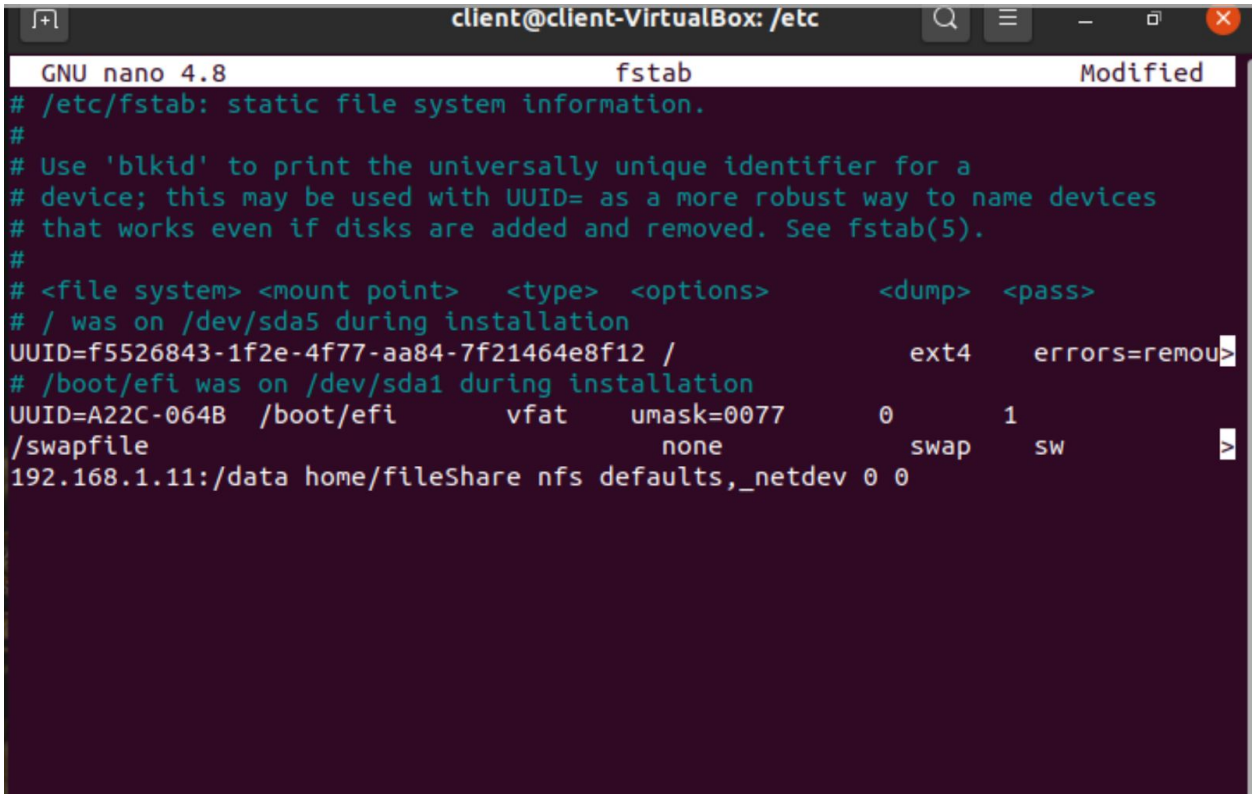
Solution -

1. First of all, we will mount the shared directory to in/mnt/nfsshare on the client server.
2. The above mount command has mounted the nfs shared directory on the nfs client temporarily.
3. To mount an NFS directory permanently on the system we will made entry in /etc/fstab.
   Syntax:
   - # mount -t nfs 110.168.2.10:/nfsshare/mnt/nfsshare
   - # vi /etc/fstab
   - 110.168.2.10:/nfsshare/mnt nfs defaults 0 0

Inside etc/fstab



In simple words, you can use the mount command to add the NFS Share exported file by the NFS server on the client machine.
The syntax of mount command is as follows-
 # mount [option..] : [exported_directory] MOUNT_POINT
 # mount | grep nfs
This command temporarily mounts the nfs shared directory on to nfs client temporarily.
To mount an NFS directory permanently on the client across reboots, go to the */etc/fstab* file
~# vi /etc/fstab
Add the following entry in the file:
<NFS_Server_IP>:[Exported_Directory_Name] MOUNT_POINT nfs defaults 0 0
For the above given data, the commands will be executed in the following order
~# mount -t 110.168.2.10:/data /mnt/data
~# mount | grep nfs ~# vi /etc/fstab
110.168.2.10:/data /mnt nfs defaults 0 0



Question 6

How can we check whether 'Portmap' service is on or not?

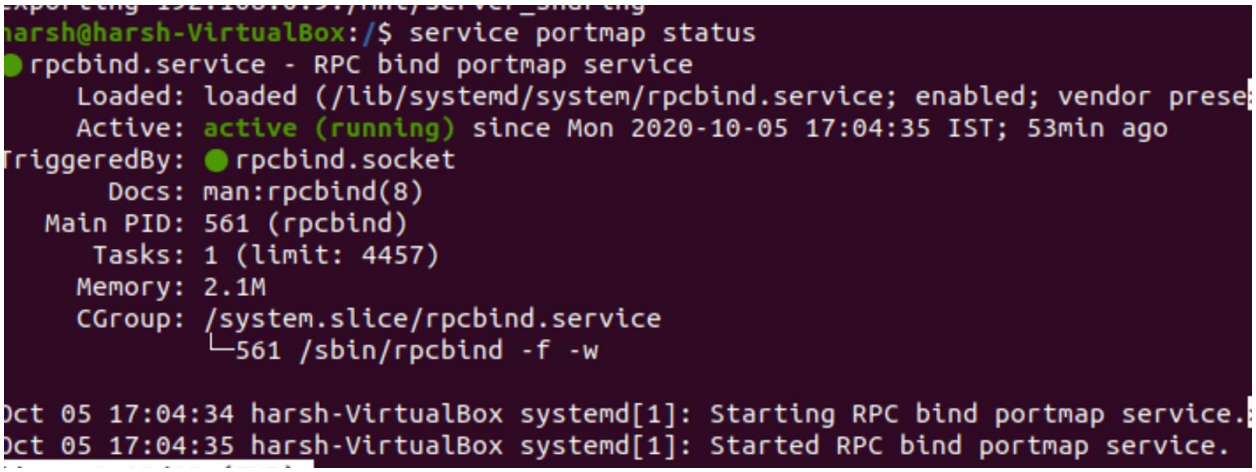
Solution -

Portmap is a server that converts RPC program numbers into DARPA protocol port numbers. It must be running in order to make RPC calls. When a client wishes to make an RPC call to a given program number, it will first contact portmap on the server machine to determine the port number where RPC packets should be sent.

Portmap services map RPC requests to the correct services. RPC services notify Portmap when they start, revealing the port number they are monitoring.

To check whether the portmap service is running or not, you need to type the following command (login as root) ~# service portmap status And then you will see a message in the nextline whether the portmap service is running currently or not.

Syntax for Checking Status:
service portmap status



Question 7

Show how automount works on NFS.

Solution -

automount uses the same kernel table (/etc/mnttab) as the conventional NFS mounting approach in which to store a record of all active mounts. When automount creates a mount, it adds a record of the mount to /etc/mnttab. When it unmounts a filesystem, it removes the record of that mount from /etc/mnttab.

We can use automount using autofs. Install it.

automount is a daemon that automatically and transparently mounts an NFS file system as needed. It monitors attempts to access directories that are associated with an automount map, along with any directories or files that reside under them. When a file is to be accessed, the daemon mounts the appropriate NFS file system. You can assign a map to a directory using an entry in a direct automount map, or by specifying an indirect map on the command line.

Syntax:
automount [-m] [-n] [-v] [-t duration] [-i interval] [-f file] [-s timeout] [-D name=value] ... [-d value]

automount starts when the command automount(NADM) is executed on a host that will operate as an NFS client. Execution typically takes place at boot time from the command /etc/nfs, but automount can also be executed from the command line.

autofs is a service in Linux like operating system which automatically mounts the file system and remote shares when it is accessed.

Step 1. install the autofs package using
~# apt-get install autofs
Step 2. Open the /etc/auto.master file
~# vi /etc/auto.master
Step 3. Add the following line in this file
/nfsshare /etc/auto.nfsdb --timeout=180
Step 4. Create the /etc/auto.nfsdb file
~# vi /etc/auto.nfsdb
Step 5. Add the following line
nfsshare -fstype=nfs,rw,soft,intr /nfsshare
Step 6. Start the autofs service
~# systemctl start autofs.service
Now you can try to access the mount point.

Question 8

Which are the important configuration files for DNS server?

Solution -

Syntax:

- $ sudo nano /etc/auto.master /nfs /etc/auto.nfs
- $ sudo nano /etc/auto.nfs
- $ server -fstype= nfs4 server:

The client needs the same changes to '/etc/default/nfs-common' to connect to an NFSv4 server.

DNS on a name server consists of a configuration file called "/etc/named.conf.

There are three main client configuration files associated with DNS:

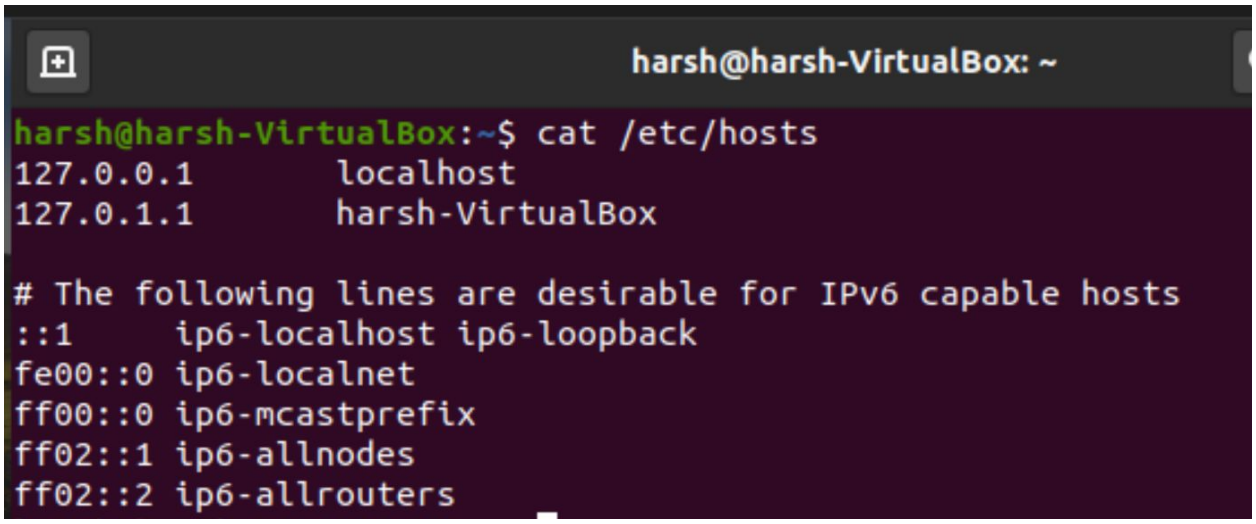
- /etc/hosts,

- /etc/nsswitch.conf ,and
- /etc/resolv.conf.

In addition to the in.named daemon, DNS on a name server consists of a configuration file called named.conf, a resolver file named resolv.conf, and four types of zone data files.

Some of the important configuration files for DNS Server are->

1. /etc/named.conf
   The configuration file specifies the type of server it is running on and the zones that it serves as 'Master', 'Slave', or 'Stub'. It also defines security, logging, and a finer granularity of options applied to zones.
2. /etc/resolv.conf
   This file resides on every DNS client (including DNS servers) and designates the servers that the client queries for DNS information.
3. root.cache
   This file establishes the names of root servers and lists their addresses.



Question 9

What is the role of "sync" option for NFS server?

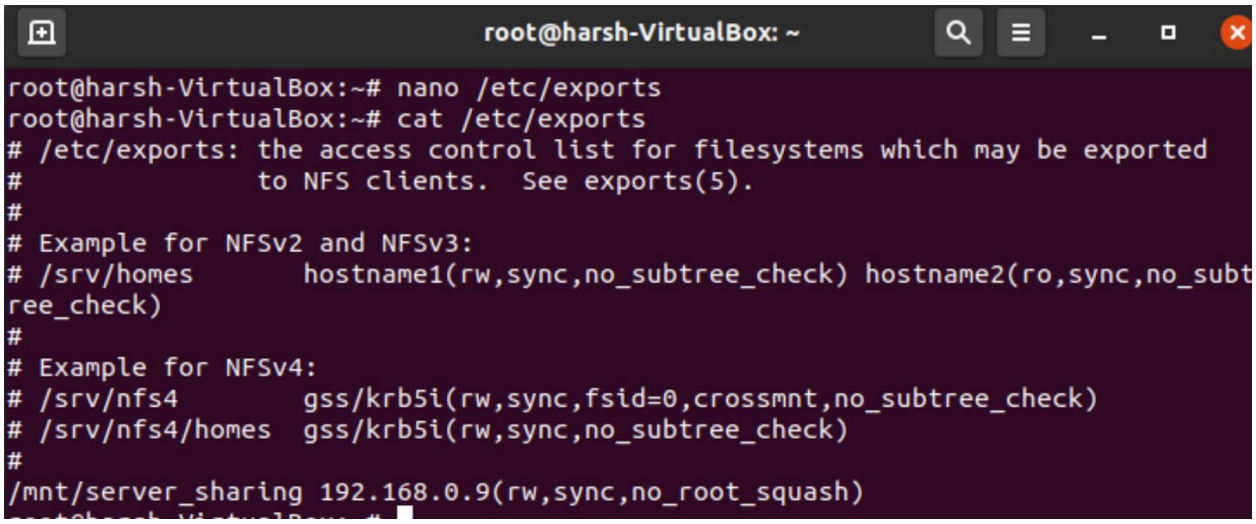
Solution -

Most of us use the synchronous option on the NFS server.
For synchronous writers, the server replies to NFS clients only when the data has been written to stable storage.

Many people prefer this option because they have little chance of losing data if the NFS server goes down or network connectivity is lost.
The synchronous or asynchronous mode can be set when the filesystem is mounted on the clients by simply putting sync or async on the mount command line or in the file /etc/fstab for the NFS filesystem.
If you don't have copies of the data cannot be easily or quickly reproduced, then perhaps synchronous mode is the better option.

Sync: Whenever there are write modifications to NFS file, the write operations are transferred immediately to NFS Server. It has a performance penalty when using Mechanical Drives. Sync confirms the request to shared directory only once changes have been committed.

If sync is specified, the server waits until the request is written to disk before responding to the client. The sync option is recommended because it follows the NFS protocol. File data changes are made to the disk immediately, which has an impact on performance but is less likely to result in a data loss.



Question 10

You are unable to mount an NFS share. How will you trace out the reason?

Solution -

- On the client, check that the NFS server is reachable.
- If the command reports that the server is alive, remotely check the NFS server.

- If the server is not reachable from the client, ensure that the local name service is running on the client.
- If the host information is correct but the server is not reachable from the client, run the ping command from another client.
- If the command run from a second client fails, check whether the NFS service is enabled on the server.
- If the server is reachable from the second client, use ping to check the connectivity of the first client to other systems on the local network.
- If the software is correct, check the networking hardware.

Also, carry out the following steps if there is some problem in mounting the NFS share:

1. Check if there is some problem in the network connection
2. Check whether the portman biod daemon is running on the client or not.
3. Check if the server is running or not.
4. Check for the /etc/exports file on the server list name of the file system that the client wants to mount and that the file system is exported or not.
5. Check that you have permissions to mount nfs share or not. Check /etc/exports file. Secondly you can get RPC error: Program Not Registered (or another "RPC" error) For this check your NFS server and portmap service running or not by "rpcinfo -p"

Troubleshooting NFS is important as sometimes you might not be able to mount the file properly. Some of the common errors encountered are

1. **Server Not Responding**

   The NFS Client and server communicate using Remote Procedure Call (RPC). Both the host- >client and client->host paths must be functional.

Solution

Use common tools such as ping, traceroute or tracepath to verify that the client and server machines can reach each other. If not, then check the Network Interface Card (NIC) settings.

2. **No route to host**

   This error can be reported when the client attempts to mount an NFS file system, even if the client can successfully ping the server. This can be caused by the RPC messages being filtered by the host firewall, or a network switch.

Solution

Check firewall status, if enabled, then disable firewall

~# ufw status ~# service iptables stop

3. **mount clntudp_create**:

   RPC: Port mapper failure RPC: Unable to receive The Linux NFS implementation requires that both the NFS service and the portmapper (RPC) service be running on both the client and the server.

Solution

First of all check if the portmap service is running

~# service portmap status

If not, then restart the service with the help of the following command-

~# chkconfig portmap on

~# service portmap start

4. **NFS Stale File Handle**

Unlike traditional Linux file systems that allow an application to access an open file even if the file has been deleted using unlink or rm, NFS does not support this feature. An NFS file is deleted immediately. Any program which attempts to do further I/O on the deleted file will receive the "NFS Stale File Handle" error. For example, if your current working directory is an NFS directory and is deleted, you will see this error at the next shell prompt.

Solution

To refresh the client's state with that of the server you may forcibly unmount the mount point using the command-

~# umount -f /mnt/mount_point

5. **Access Denied or Permission Denied**

Solution

Check the export permissions for the NFS file system.

You can do this

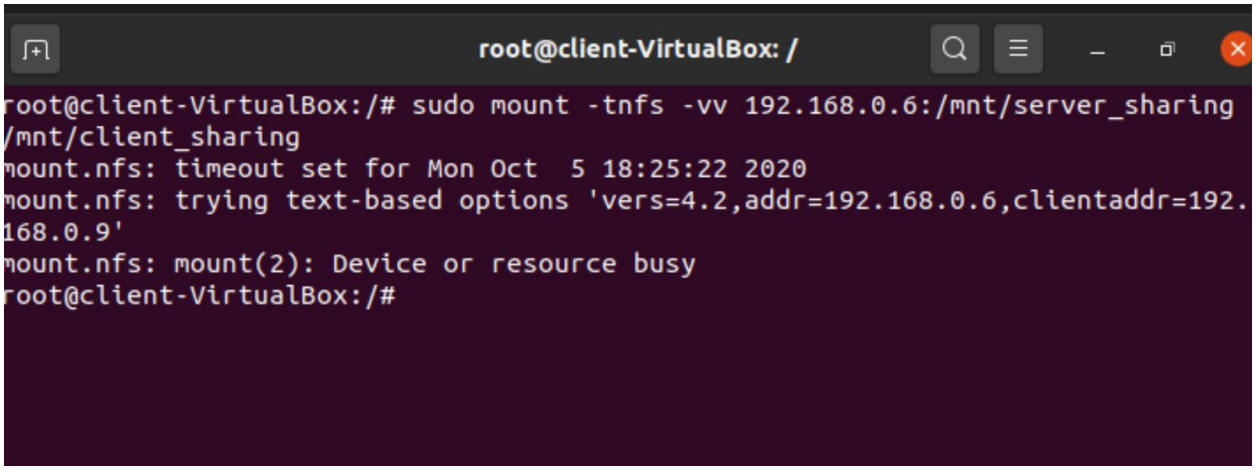
1} From client:

~# showmount -e

2} From server:

~# exportfs -a

And change the permissions



```
harsh@harsh-VirtualBox:~$ rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp  36141  mountd
    100005    1   tcp  35235  mountd
    100005    2   udp  44808  mountd
    100005    2   tcp  39193  mountd
    100005    3   udp  37126  mountd
    100005    3   tcp  46537  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049
    100003    3   udp   2049  nfs
    100227    3   udp   2049
    100021    1   udp  48774  nlockmgr
    100021    3   udp  48774  nlockmgr
    100021    4   udp  48774  nlockmgr
    100021    1   tcp  42443  nlockmgr
    100021    3   tcp  42443  nlockmgr
    100021    4   tcp  42443  nlockmgr
```