

Ex No: 6

Transfer of files from one Virtual Machine to another

04/08/26

Aim:

To transfer files between two virtual machines using : Shared Folder, NFS , SSH and FTP methods.

Theory:**1. Shared Folder**

- A feature in **VirtualBox/VMware** that allows the host and guest VM to access a common directory.
- You configure a folder on the host system, and the VM mounts it as a network/shared directory.
- Good for quick file transfers without networking.

2. NFS (Network File System)

- A **distributed file system protocol** mainly used in Linux/Unix.
- It allows a system to share directories and files with others over a network.
- Files are accessed on a remote machine as if they are on the local machine.

3. SSH (Secure Shell)

- A **secure protocol** for connecting to remote machines.
- File transfer is done using scp (secure copy) or sftp (SSH File Transfer Protocol).
- Provides encrypted communication, so it's more secure than FTP.

4. FTP (File Transfer Protocol)

- A standard protocol for transferring files between client and server over a network.
- Works on ports **20 (data)** and **21 (control)**.
- Can be managed through command-line (ftp>) or GUI clients (like FileZilla).
- Less secure than SSH (unless using FTPS).

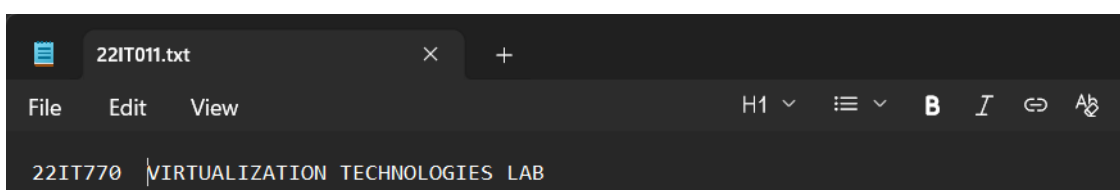
Procedure:**Shared Folder****Step 1: Create a Shared Folder on Your Host**

1. On your **host machine** (Windows/Linux):

- Create a folder, e.g., C:\VMShared



- Add a test file (22IT011.txt) to confirm later.



Step 2: Enable Shared Folder in VirtualBox (for Both VMs)

Repeat this for **both Ubuntu VMs**.

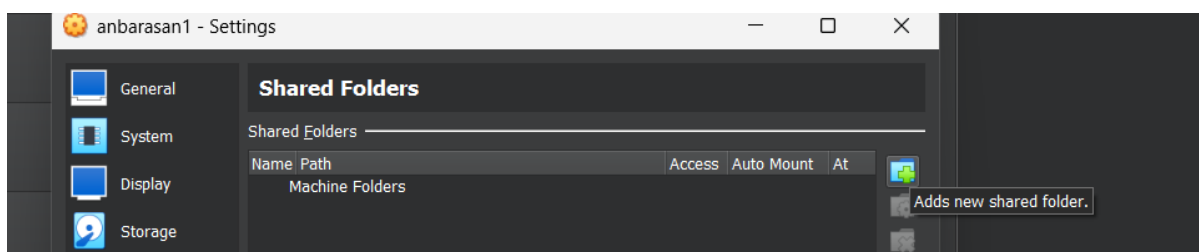
1. **Shut down** the VM if it's running.
2. In **VirtualBox Manager**, select your VM.



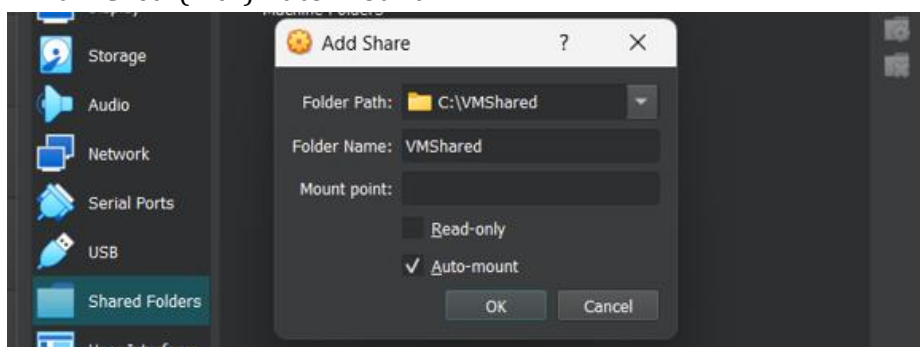
3. Click **Settings** → Go to **Shared Folders** tab.



4. Click **“+”** (Add new shared folder).

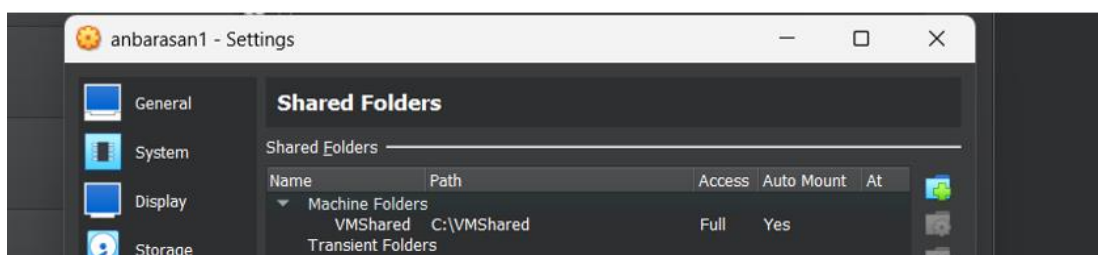


5. Set:
 - **Folder Path:** Select C:\VMShared (or wherever you created it).
 - **Folder Name:** VMShared (updates once path is entered)
 - Check(Tick) **Auto-mount**



Click **OK**.

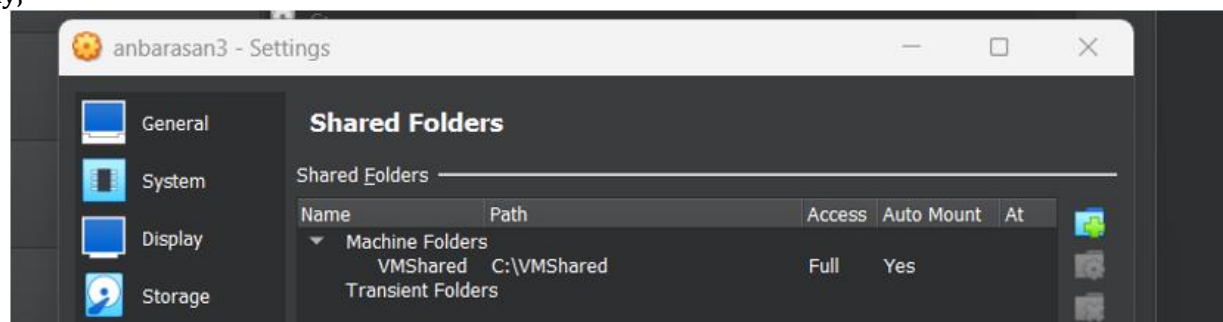
6. Then, again Click **OK** (at bottom) to save.



Do this for both VMs.

VM2:

Finally,



Step 3: Install VirtualBox Guest Additions in Ubuntu (if not already done)

Run the following commands in **each Ubuntu VM**:

sudo apt update

sudo apt install build-essential dkms linux-headers-\$(uname -r)

VM1:

```
tce@tce-VirtualBox:~$ sudo apt update
[sudo] password for tce:
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
```

```
tce@tce-VirtualBox:~$ sudo apt install build-essential dkms linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
build-essential set to manually installed.
linux-headers-5.4.0-150-generic is already the newest version (5.4.0-150.167~18.04.1).
linux-headers-5.4.0-150-generic set to manually installed.
```

VM2:

```
vboxuser@ubuntuos:~$ sudo apt update
[sudo] password for vboxuser:
vboxuser is not in the sudoers file. This incident will be reported.
vboxuser@ubuntuos:~$ su -
Password:
root@ubuntuos:~# sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
```

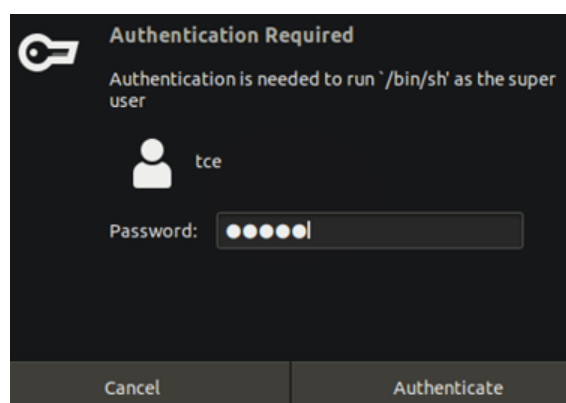
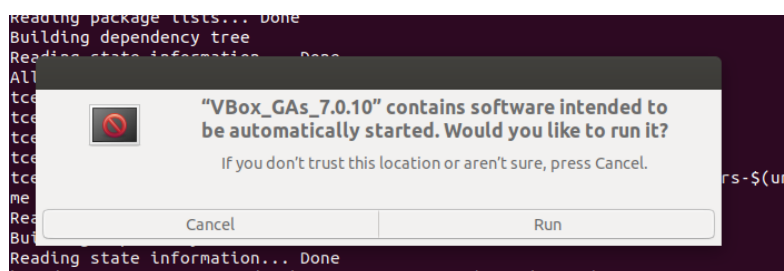
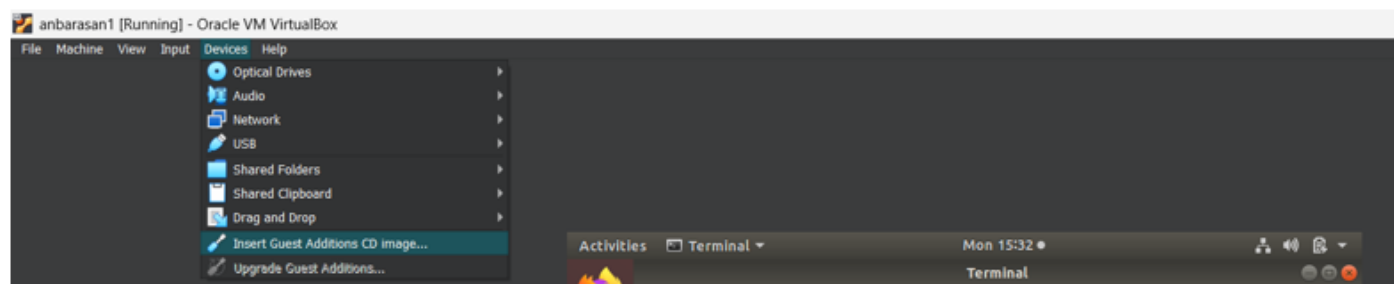
```
root@ubuntuos:~# sudo apt install build-essential dkms linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
linux-headers-6.5.0-26-generic is already the newest version (6.5.0-26.26~22.04.1).
linux-headers-6.5.0-26-generic set to manually installed.
The following additional packages will be installed:
gcc-12-devel-tools gcc-12-libasan8 libgcc-12-dev libsanj
```

Then:

1. In VirtualBox menu (inside VM window), go to:

Devices → Insert Guest Additions CD image

VM1:



```

Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.0.10 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Setting up modules
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.4.0-150-generic.
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted or 'rcvboxadd reload' triggered
VirtualBox Guest Additions: reloading kernel modules and services
VirtualBox Guest Additions: kernel modules and services 7.0.10 r158379 reloaded
VirtualBox Guest Additions: NOTE: you may still consider to re-login if some
user session specific services (Shared Clipboard, Drag and Drop, Seamless or
Guest Screen Resize) were not restarted automatically
Press Return to close this window...

```

Now use the command: `ls /media`

- It lists all the directories (i.e., mounted devices or drives) under the /media directory.

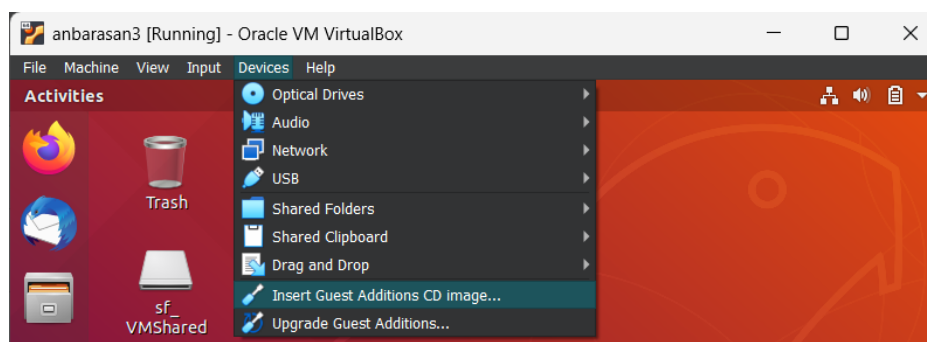
```

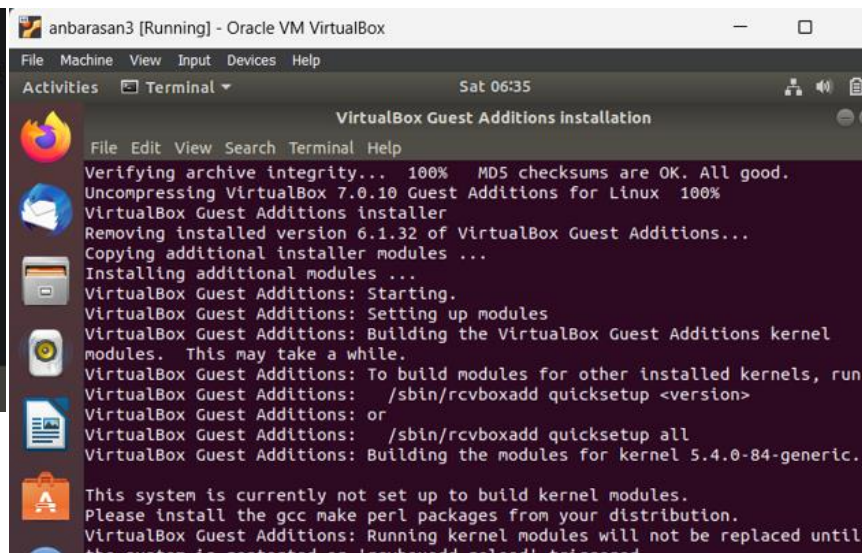
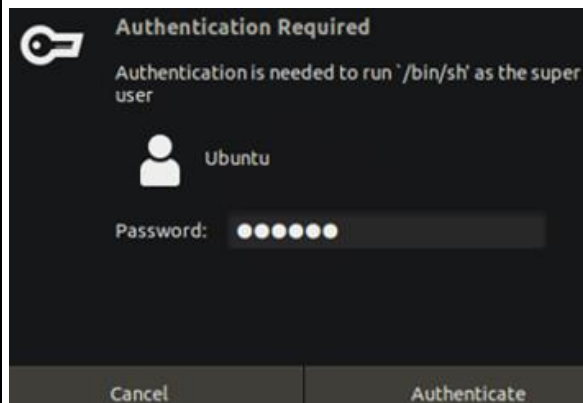
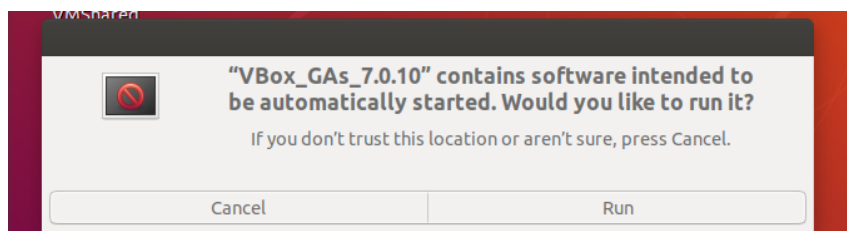
tce@tce-VirtualBox:~$ ls /media
sf_VMShared tce
tce@tce-VirtualBox:~$

```

Therefore, in **VM1** `sf_VMShared` folder is there in `/media` directory.

VM2:





Run the command: `ls /media`

```
ubuntu@ubuntu1804:~$ ls /media
sf_VMShared  ubuntu
ubuntu@ubuntu1804:~$
```

Therefore, in **VM2** also `sf_VMShared` folder is there in `/media` directory.

Step 4: Access the Shared Folder

After rebooting each VM, the shared folder should be mounted automatically at: `/media/sf_VMShared`

To check, run the command : `ls /media`

VM1 :

```
tce@tce-VirtualBox:~$ ls /media/
sf_VMShared  tce
tce@tce-VirtualBox:~$
```

VM2 :

```
ubuntu@ubuntu1804:~$ ls /media
sf_VMShared  ubuntu
ubuntu@ubuntu1804:~$
```

Run the command : `ls /media/sf_VMShared`

VM1 :

```
tce@tce-VirtualBox:~$ ls /media/sf_VMShared
ls: cannot open directory '/media/sf_VMShared': Permission denied
tce@tce-VirtualBox:~$
```

VM2 :

```
ubuntu@ubuntu1804:~$ ls /media/sf_VMShared
ls: cannot open directory '/media/sf_VMShared': Permission denied
```

You should see the **22IT011.txt** file from your host.

But in **VM1** and **VM2** , it asks for permission to list the files of `sf_VMShared` folder.

Step 5: Fix Permissions (if Permission Denied)

You need to add your user to the vboxsf group:

```
sudo usermod -aG vboxsf $USER
```

```
tce@tce-VirtualBox:~$ sudo usermod -aG vboxsf $USER
[sudo] password for tce:
tce@tce-VirtualBox:~$
```

```
ubuntu@ubuntu1804:~$ sudo usermod -aG vboxsf $USER
```

Then reboot again:

```
sudo reboot
```

Now try again:

```
ls /media/sf_VMShared
```

```
tce@tce-VirtualBox:~$ ls /media/sf_VMShared
22IT011.txt
tce@tce-VirtualBox:~$
```

```
ubuntu@ubuntu1804:~$ ls /media/sf_VMShared
22IT011.txt
```

You should now have full access.

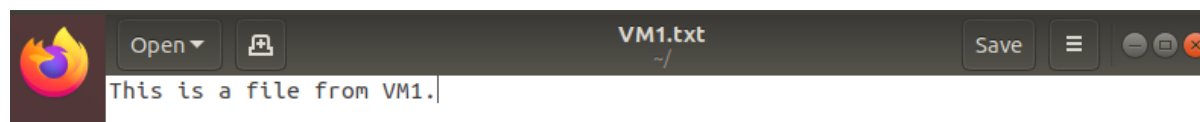
Step 6: Transfer Files Between VMs

- Trying to print the content of `22IT011.txt` file from **VM1**,

```
tce@tce-VirtualBox:~$ cat /media/sf_VMShared/22IT011.txt
22IT770 VIRTUALIZATION TECHNOLOGIES LAB tce@tce-VirtualBox:~$
```

- Now in **VM1**, copy a file to `/media/sf_VMShared` folder,

```
tce@tce-VirtualBox:~$ touch VM1.txt
tce@tce-VirtualBox:~$ gedit VM1.txt
```



```
cp VM1.txt /media/sf_VMShared/
```

```
tce@tce-VirtualBox:~$ cp VM1.txt /media/sf_VMShared
tce@tce-VirtualBox:~$ ls /media/sf_VMShared
22IT011.txt VM1.txt
tce@tce-VirtualBox:~$
```

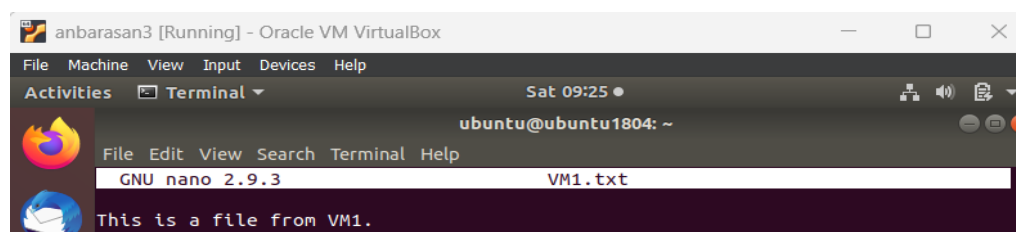
- Switch to **VM2**, and check the shared folder:

```
ls /media/sf_VMShared
```

```
ubuntu@ubuntu1804:~$ ls /media/sf_VMShared
22IT011.txt VM1.txt
```

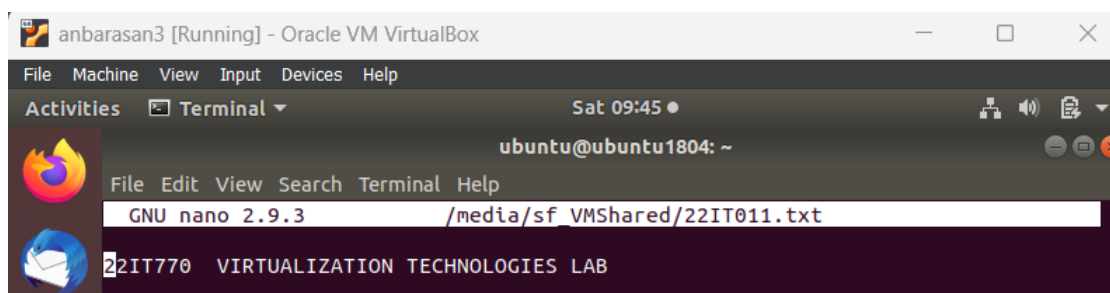
The **VM1.txt** file will appear — you can copy it out to any other location and also you can edit files in shared folder location itself. Any changes done in shared folder using one VM will also be reflected to the other VM.

```
ubuntu@ubuntu1804:~$ cp /media/sf_VMShared/VM1.txt /home/ubuntu
ubuntu@ubuntu1804:~$ nano VM1.txt
```



Also 22IT011.txt file is accessible in VM2,

```
ubuntu@ubuntu1804:~$ nano /media/sf_VMShared/22IT011.txt
```



NFS (Network File System)

Step 1: Install NFS Packages

On **both VMs**, update system packages first:

sudo apt update

```
tce@tce-VirtualBox:~$ sudo apt update
[sudo] password for tce:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [102 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://dl.google.com/linux/chrome/deb stable InRelease [1,025 B]
```

```
ubuntu@ubuntu1804:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
```

On **VM1 (Server):**

sudo apt install nfs-kernel-server -y

```
tce@tce-VirtualBox:~$ sudo apt install nfs-kernel-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
gir1.2-goa-1.0 gir1.2-spand-1
```

On **VM2 (Client):**

sudo apt install nfs-common -y

```
ubuntu@ubuntu1804:~$ sudo apt install nfs-common -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
keyutils libnfsidmap2 libtirpc1 rpcbind
```

Step 2: Create a Shared Directory on Server

On **VM1 (Server):**

sudo mkdir -p /mnt/sharedfolder

sudo chmod 777 /mnt/sharedfolder

```
tce@tce-VirtualBox:~$ sudo mkdir -p /mnt/sharedfolder
tce@tce-VirtualBox:~$ sudo chmod 777 /mnt/sharedfolder
```

This will be the folder that VM2 can access.

Step 3: Configure NFS Exports (done in VM1)

Edit the **exports** file:

```
sudo nano /etc/exports
```

Add the following line (replace 192.168.56.0/24 with your VM network range):

```
/mnt/sharedfolder 192.168.56.0/24(rw,sync,no_subtree_check)
```

```

Terminal
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/exports Modified
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_sub$
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/mnt/sharedfolder 192.168.56.0/24(rw,sync,no_subtree_check)

```

Save and exit.

Apply the export rules:

```
sudo exportfs -a
```

```
tce@tce-VirtualBox:~$ sudo exportfs -a
```

Restart NFS service:

```
sudo systemctl restart nfs-kernel-server
```

```
tce@tce-VirtualBox:~$ sudo systemctl restart nfs-kernel-server
```

Step 4: Get Server's IP Address

On VM1:

```
ip a
```

```

inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s3
    valid_lft 364sec preferred_lft 364sec

```

Allow firewall settings for NFS to avoid Connection timed out error.

```
sudo ufw allow from 192.168.56.0/24 to any port 2049 # Allow NFS main port
```

```
sudo ufw allow from 192.168.56.0/24 to any port 111 # Allow portmapper (used by NFS)
```

```
sudo ufw reload # Reload firewall
```

```

tce@tce-VirtualBox:~$ sudo ufw allow from 192.168.56.0/24 to any port 2049
Rule added
tce@tce-VirtualBox:~$ sudo ufw allow from 192.168.56.0/24 to any port 111
Rule added
tce@tce-VirtualBox:~$ sudo ufw reload
Firewall reloaded

```


Step 5: Mount NFS Share on Client

On VM2 (Client):

```
sudo mkdir -p /mnt/nfsshare
```

```
ubuntu@ubuntu1804:~$ sudo mkdir -p /mnt/nfsshare
```

```
sudo mount 192.168.1.10:/mnt/sharedfolder /mnt/nfsshare
```

```
ubuntu@ubuntu1804:~$ sudo mount 192.168.56.103:/mnt/sharedfolder /mnt/nfsshare
```

Step 6: Verify File Transfer

On VM2:

```
cd /mnt/nfsshare
```

```
ubuntu@ubuntu1804:~$ cd /mnt/nfsshare
```

```
echo "22IT770 - VIRTUALIZATION TECHNOLOGIES LAB" > file.txt
```

```
ubuntu@ubuntu1804:/mnt/nfsshare$ echo "22IT770 - VIRTUALIZATION TECHNOLOGIES LAB" > file.txt
```

Now check on **VM1 (Server)** inside `/mnt/sharedfolder/` — the file should be there.

```
tce@tce-VirtualBox:~$ cd /mnt/sharedfolder/
tce@tce-VirtualBox:/mnt/sharedfolder$ ls -l
total 4
-rw-rw-r-- 1 tce tce 42 Aug 22 20:53 file.txt
tce@tce-VirtualBox:/mnt/sharedfolder$ cat file.txt
22IT770 - VIRTUALIZATION TECHNOLOGIES LAB
tce@tce-VirtualBox:/mnt/sharedfolder$
```

SSH (Secure Shell)**Step 1: Install SSH**

On both VMs (VM1 & VM2):

```
sudo apt update
```

```
sudo apt install openssh-server -y
```

```
tce@tce-VirtualBox:~$ sudo apt install openssh-server -y
[sudo] password for tce:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.7).
The following packages were automatically installed and are no longer required:
```

```
ubuntu@ubuntu1804:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openssh-client openssh-sftp-server
```

Check SSH status:

```
sudo systemctl status ssh
```

```
tce@tce-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Fri 2025-08-22 19:11:46 IST; 2h 12min ago
   Main PID: 779 (sshd)
```

```
ubuntu@ubuntu1804:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-08-22 12:10:18 EDT; 8min ago
     Main PID: 2715 (sshd)
```

Step 2: Get Server's IP

On VM1 (Server, username = tce):

```
ip a
inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s3
    valid_lft 364sec preferred_lft 364sec
```

Allow firewall settings for SSH to avoid Connection timed out error.

```
sudo ufw allow 22/tcp
sudo ufw reload
```

```
tce@tce-VirtualBox:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
tce@tce-VirtualBox:~$ sudo ufw reload
Firewall reloaded
tce@tce-VirtualBox:~$
```

Step 3: Connect from VM2 to VM1

On VM2 (Client, username = ubuntu):

```
ssh tce@192.168.56.103
```

```
ubuntu@ubuntu1804:~$ ssh tce@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ECDSA key fingerprint is SHA256:ImaQYLVZCqfB/6sZmSQmIJta7VCyk+4M5Sfz0gEL2V4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.103' (ECDSA) to the list of known hosts.
tce@192.168.56.103's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Infrastructure is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

289 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

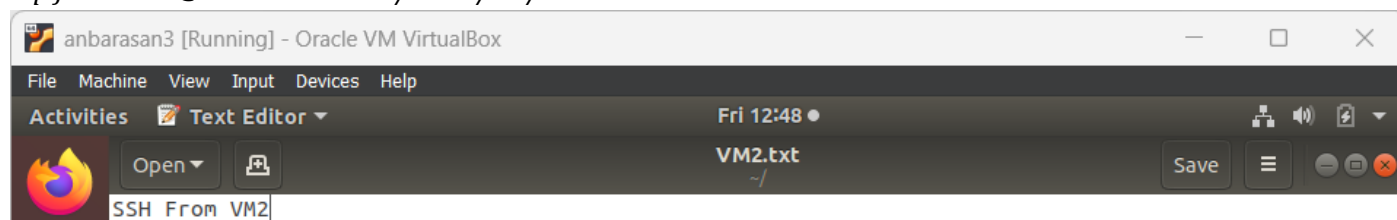
Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Sat Oct  5 19:15:29 2024
tce@tce-VirtualBox:~$
```

Enter tce's password → now you are logged into VM1.

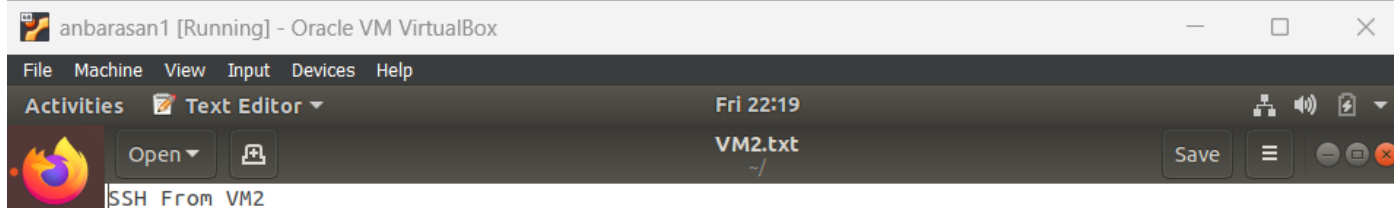
Step 4: Transfer Files with SCP (From VM2)

(a) Copy file from VM2 → VM1

```
scp file.txt tce@192.168.56.103:/home/tce/
```

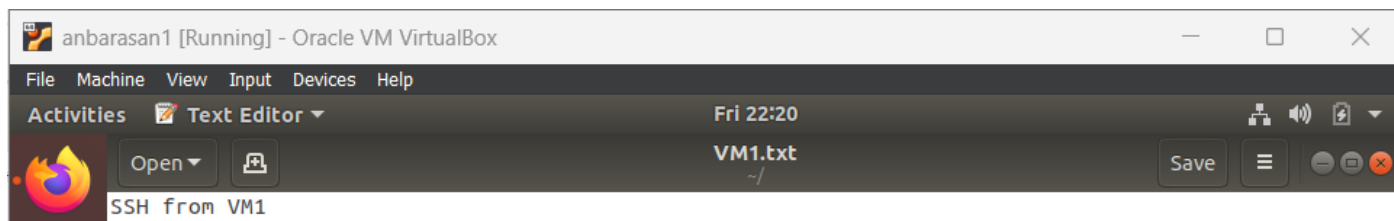


```
ubuntu@ubuntu1804:~$ scp VM2.txt tce@192.168.56.103:/home/tce/
tce@192.168.56.103's password:
VM2.txt                                100% 13    4.3KB/s   00:00
ubuntu@ubuntu1804:~$
```

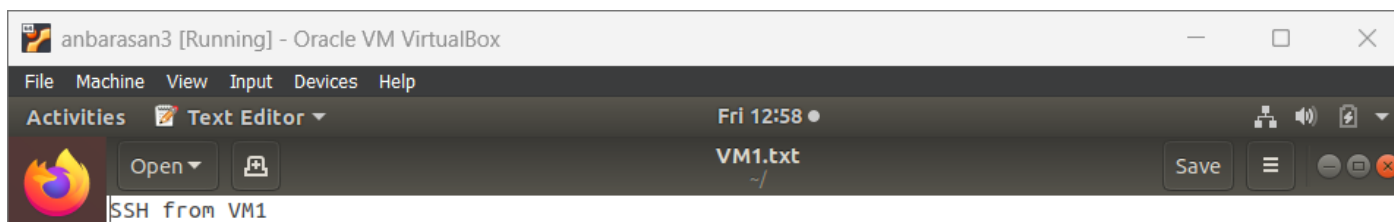


(b) Copy file from VM1 → VM2

scp tce@192.168.56.103:/home/tce/serverfile.txt /home/ubuntu/



```
ubuntu@ubuntu1804:~$ scp tce@192.168.56.103:/home/tce/VM1.txt /home/ubuntu/
tce@192.168.56.103's password:
VM1.txt                                100% 13    5.5KB/s   00:00
```



FTP (File Transfer Protocol)

Step 1: Install FTP Server on VM1

(VM1 = server, VM2 = client)

On VM1:

sudo apt update

sudo apt install vsftpd -y

```
tce@tce-VirtualBox:~$ sudo apt install vsftpd -y
[sudo] password for tce:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
gir1.2-goa-1.0 gir1.2-spand-1 libmemcached11 libmemcachedutil2 proftpd-doc
```

Enable and start the service:

sudo systemctl enable vsftpd

sudo systemctl start vsftpd

```
tce@tce-VirtualBox:~$ sudo systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
tce@tce-VirtualBox:~$ sudo systemctl start vsftpd
tce@tce-VirtualBox:~$ sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-08-22 23:00:53 IST; 9min ago
     Main PID: 5257 (vsftpd)
```

Step 2: Configure vsftpd

Edit the config file:

```
sudo nano /etc/vsftpd.conf
```

```
tce@tce-VirtualBox:~$ sudo nano /etc/vsftpd.conf
[sudo] password for tce:
```

Make sure these lines are set (uncomment or add if missing):

```
local_enable=YES
```

```
write_enable=YES
```

```
chroot_local_user=YES
```

```
allow_writeable_chroot=YES
```

Save (CTRL+S, CTRL+X)

```
tce@tce-VirtualBox:~$ grep -v "^#" /etc/vsftpd.conf
listen=NO
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
chroot_local_user=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

allow_writeable_chroot=YES
```

Restart the ftp server:

```
sudo systemctl restart vsftpd
```

```
tce@tce-VirtualBox:~$ sudo systemctl restart vsftpd
```

Step 3: Check VM1 IP Address

On VM1, run:

```
ip a
inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s3
    valid_lft 364sec preferred_lft 364sec
```

Allow firewall settings for FTP to avoid Connection timed out error.

```
sudo ufw allow 21/tcp
```

```
sudo ufw allow 20/tcp # (for data transfer)
```

```
sudo ufw status
```

```
tce@tce-VirtualBox:~$ sudo ufw allow 21/tcp
Rule added
Rule added (v6)
tce@tce-VirtualBox:~$ sudo ufw allow 20/tcp
Rule added
Rule added (v6)
tce@tce-VirtualBox:~$ sudo ufw reload
Firewall reloaded
```

Step 4: Connect from VM2 (FTP client)

On VM2, install ftp client:

sudo apt install ftp -y

```
ubuntu@ubuntu1804:~$ sudo apt install ftp -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Now connect:

ftp 192.168.56.103

Enter VM1's username (tce) and password when prompted.

```
ubuntu@ubuntu1804:~$ ftp 192.168.56.103
Connected to 192.168.56.103.
220 (vsFTPd 3.0.3)
Name (192.168.56.103:ubuntu): tce
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Step 5: FTP Commands

Once connected, you can use:

- *ls* → list files on VM1

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  3 1000    1000    4096 Oct 05  2024 Desktop
drwxr-xr-x  6 1000    1000    4096 Aug 23 00:21 Documents
drwxr-xr-x  3 1000    1000    4096 Aug 10 22:20 Downloads
drwxrwxr-x  2 1000    1000    4096 Aug 23 00:20 FTP1
drwxrwxr-x  2 1000    1000    4096 Aug 23 00:20 FTP2
drwxr-xr-x  2 1000    1000    4096 May 12  2022 Public
drwxr-xr-x  2 1000    1000    4096 May 12  2022 Templates
-rw-rw-r--  1 1000    1000      13 Aug 22 22:20 VM1.txt
-rw-rw-r--  1 1000    1000      13 Aug 22 22:17 VM2.txt
-rw-rw-r--  1 1000    1000    1301 Jul 19 22:19 anbarasan.class
-rw-rw-r--  1 1000    1000     625 Jul 19 22:18 anbarasan.java
drwxrwxr-x  2 1000    1000    4096 Aug 23 00:19 ubuntu images
226 Directory send OK.
```

- *cd folder* → change directory on VM1

```
ftp> cd FTP1
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r--  1 1000    1000      17 Aug 23 00:18 ftp_vm1.txt
226 Directory send OK.
```


- *get filename* → download file from VM1 → VM2

```
ftp> get ftp_vm1.txt
local: ftp_vm1.txt remote: ftp_vm1.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ftp_vm1.txt (17 bytes).
226 Transfer complete.
17 bytes received in 0.02 secs (0.7959 kB/s)
```

- *put filename* → upload file from VM2 → VM1

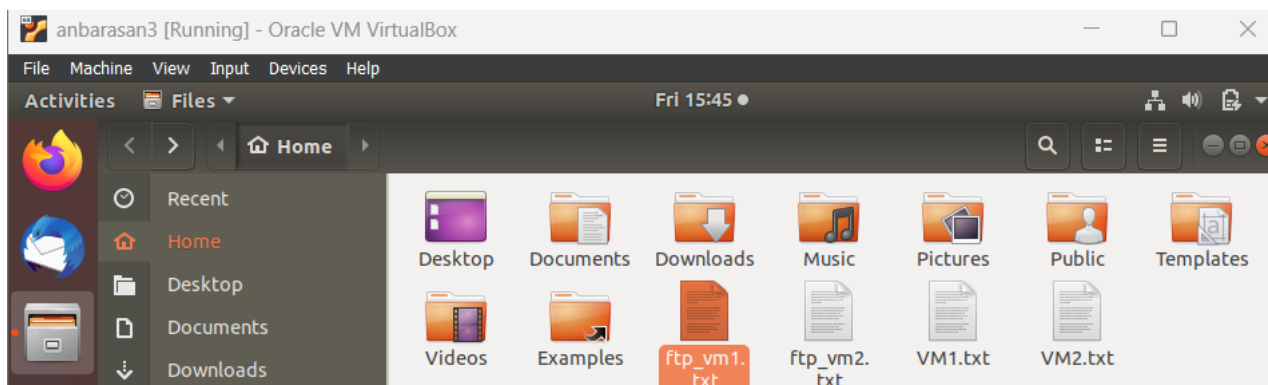
```
ftp> put ftp_vm2.txt
local: ftp_vm2.txt remote: ftp_vm2.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
17 bytes sent in 0.00 secs (55.7099 kB/s)
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r-- 1 1000 1000 17 Aug 23 00:18 ftp_vm1.txt
-rw----- 1 1000 1000 17 Aug 23 00:26 ftp_vm2.txt
226 Directory send OK.
```

- *bye* → exit

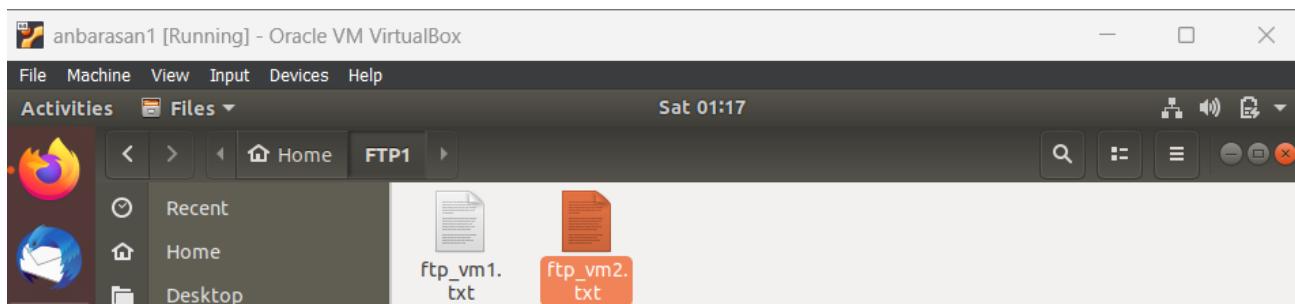
```
ftp> bye
221 Goodbye.
ubuntu@ubuntu1804:~$
```

Step 6: Verify

- ftp_vm1.txt should now be in VM2's current directory.



- ftp_vm2.txt should now be uploaded to VM1's */home/FTP1/* directory.



Result:

Thus, the Files were shared and accessed between the two virtual machines using Shared Folders, NFS, SSH and FTP methods successfully.