CSE4011- Virtualization

J REPORT

Creating virtual environment for Python in Anaconda in Ubuntu Virtual Machine

By

20BCE1852	DEVARINTI DHAPATLA PUNEETH REDDY
20BCE1855	SANJIL K C
20BCE1892	LENIN VASAN

B.Tech CSE

Submitted to

Dr.ANUSOOYA G,
Assistant Professor Senior,
SCOPE, VIT, Chennai

School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

April 2023

INDEX PAGE

TEAM MEMBERS CONTRIBUTION

Worklet Tasks	*Contributor's Names*
Creating Virtual Machine	Lenin Vasan, Puneeth Reddy
Creating NAT Network	Lenin Vasan, Puneeth Reddy
Transferring files between VM's and Creating Shared Folder and App	Sanjil K C

1.ABSTRACT

This report outlines the process of creating a virtual environment for Python using Anaconda in both Ubuntu and Windows virtual machines using Oracle VM VirtualBox. The report begins by discussing the importance of virtual environments in Python development and the benefits of using a virtual machine. It then provides step-by-step instructions on how to install Oracle VM VirtualBox, download Ubuntu and Windows virtual machine images, and set up the virtual machines. The report also covers how to install Anaconda in both virtual machines, create a virtual environment using the Anaconda command line interface, activate the environment, and install packages within the environment. Additionally, the report discusses how to share files between the virtual machines and the host machine using shared folders and transferring file between Virtual Machines.

2.INTRODUCTION

Python is a popular language for software development, data analysis, and machine learning, among other applications. However, as the number of Python packages and libraries grows, managing dependencies and conflicts between them becomes increasingly complex. A solution to this problem is to use virtual environments, which provide a way to create isolated environments for different projects, each with its own set of dependencies and packages.

In this report, we will explore the process of creating virtual environments for Python using Anaconda in both Ubuntu and Windows virtual machines using Oracle VM VirtualBox.

The report aims to provide a comprehensive guide for developers who want to set up a clean and isolated environment for their Python projects, regardless of the operating system they are using.

The report will begin by discussing the importance of virtual environments in Python development and the benefits they provide. We will then explain how to install Oracle VM VirtualBox, download and set up Ubuntu and Windows virtual machines, and install Anaconda in the virtual machines. We will provide detailed instructions on how to create virtual environments, activate them, and install packages within them using the Anaconda command line interface.

Additionally, we will discuss how to share files between the virtual machines and the host machine using shared folders. This will allow developers to work on projects using their preferred text editors or integrated development environments (IDEs) on the host machine while keeping their virtual environments isolated.

Finally, we will conclude by discussing the advantages of using virtual environments and virtual machines for Python development. We hope that this report will provide a useful guide for developers who want to set up a clean and isolated environment for their Python projects using Anaconda and virtual machines.

3.TECHNOLOGY LEREAND FOR PROJECT

In this project, we not only learned about virtualization and Anaconda but also about networking and file transfer between two virtual machines. In this report, we will discuss the technologies and tools we used to transfer files between two virtual machines, the significance of the NAT network, and how it all fits into our Python development environment.

a) Network Address Translation (NAT):

In Oracle VM VirtualBox, NAT is a networking mode that allows the guest operating system (virtual machine) to access the internet through the host operating system's network connection. NAT provides an isolated network for the virtual machine that is separate from the host network. We used the NAT network mode to allow our Ubuntu and Windows virtual machines to communicate with each other and transfer files.

b) Setting up NAT networking:

To set up NAT networking, we configured the virtual machines' network adapters to use NAT as the network mode. This allowed the virtual machines to obtain an IP address from the virtual network DHCP server and access the internet through the host operating system's network connection. We also ensured that the virtual machines could communicate with each other by assigning them unique IP addresses within the same subnet.

c) Transferring files between two virtual machines:

To transfer files between two virtual machines, we used the NAT network to establish a connection between them. We created a shared folder on one virtual machine and then mounted it as a network drive on the other virtual machine. This allowed us to transfer files between the virtual machines as if they were on the same network.

d) Mounting a network drive:

To mount a network drive, we used the command-line interface on the virtual machine to create a mount point and then used the mount command to connect to the shared folder on the other virtual machine. Once the network drive was mounted, we could access it as if it were a local drive and transfer files between the virtual machines.

In addition to the networking and file transfer technologies discussed in the previous report, we also learned about using the Secure Copy Protocol (SCP) command to transfer files between two virtual machines in our project.

e) SCP command:

SCP is a command-line utility that allows us to securely transfer files between remote hosts using the SSH protocol. SCP uses the same authentication and security mechanisms as SSH, providing secure file transfer over the network.

f) Using SCP to transfer files between virtual machines:

To transfer files between two virtual machines using SCP, we first had to ensure that SSH was enabled on both virtual machines. We then used the SCP command to copy files from one virtual machine to another. The syntax for the SCP command is as follows:

scp [options] [source] [destination]

In our case, the source was the file we wanted to transfer, and the destination was the IP address and path of the target virtual machine. We used the SCP command to transfer files between our Ubuntu and Windows virtual machines using their respective IP addresses.

g) Advantages of using SCP:

Using SCP to transfer files between virtual machines offers several advantages. First, it provides a secure and encrypted connection, ensuring that files are transferred securely over the network. Second, it is a simple and easy-to-use command-line utility that requires minimal configuration. Third, it is a platform-independent utility, which means it can be used to transfer files between different operating systems and platforms.

h) Disadvantages of using SCP:

There are some limitations to using SCP to transfer files between virtual machines. One limitation is that SCP can only transfer files one at a time, which can be time-consuming if we need to transfer multiple files. Another limitation is that SCP does not provide a progress indicator, which means we cannot track the progress of the file transfer.

i) Alternative file transfer methods:

There are several alternative file transfer methods that we can use to transfer files between virtual machines. One such method is using SFTP (Secure File Transfer Protocol), which provides a similar level of security as SCP but with more advanced features, such as multiple file transfers and progress indicators. Another method is using NFS (Network File System), which allows us to mount a remote file system as a local file system, providing seamless file access between virtual machines.
In conclusion, using the SCP command to transfer files between virtual machines provides a simple and secure method for transferring files over the network. It is a useful tool for transferring individual files or small sets of files between virtual machines. However, if we need to transfer large numbers of files or want more advanced features, we may need to consider alternative file transfer methods such as SFTP or NFS. The project provided us with hands-on experience in using these technologies and tools, demonstrating the importance of secure file transfer in Python development.

In addition to the networking and file transfer technologies discussed in the previous report, we also learned about using the Secure Copy Protocol (SCP) command to transfer files between two virtual machines in our project.

j) Using NAT to transfer files between virtual machines:

In our project, we used Network Address Translation (NAT) to enable communication between our Ubuntu and Windows virtual machines. NAT is a technology that allows us to connect multiple virtual machines to the same network using a single IP address. This allows us to create a private network for our virtual machines, allowing them to communicate with each other securely and efficiently.

k) Advantages of using NAT:

Using NAT to connect virtual machines offers several advantages. First, it allows us to create a private network for our virtual machines, providing a secure and isolated environment for our development work. Second, it allows us to conserve IP addresses, as we can connect multiple virtual machines to the same network using a single IP address. Third, it simplifies network configuration, as we do not need to configure complex network settings for each virtual machine.

l) Disadvantages of using NAT:

There are some limitations to using NAT to connect virtual machines. One limitation is that it can be difficult to connect to virtual machines from outside the NAT network, as we need to configure port forwarding to enable external access. Another limitation is that NAT can introduce latency and bandwidth limitations, as all traffic needs to pass through the NAT gateway.

m) Using SCP with NAT:

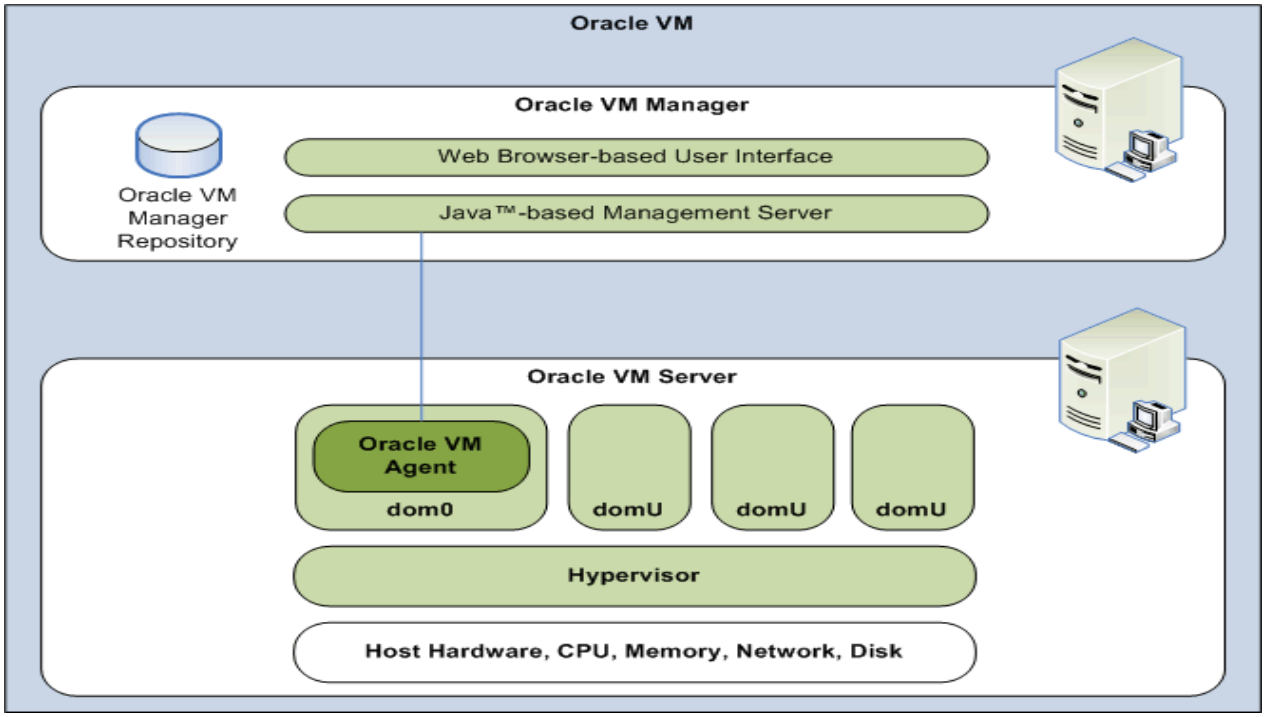
We used the SCP command to transfer files between our Ubuntu and Windows virtual machines using NAT. We first had to identify the IP address of the virtual machine we wanted to transfer files to, which was assigned by the NAT gateway. We then used the

4.PROJECT ARCHITECTURE

Here's a high-level architecture for your project:

1) Host Operating System: The physical machine that runs the virtualization software.

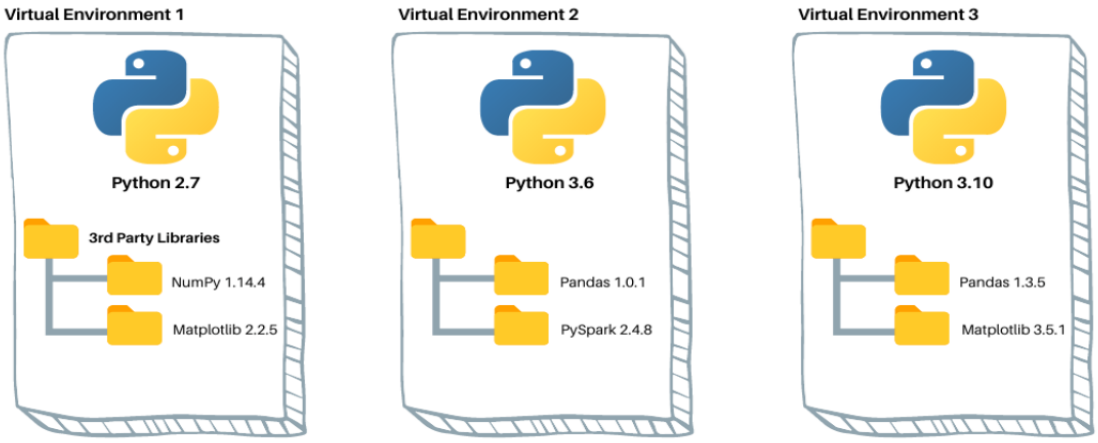
2) Virtualization Software: Oracle VM VirtualBox, which allows you to create and manage virtual machines.



3) Guest Operating Systems: Ubuntu Virtual Machine and Windows Virtual Machine, each with their own virtual environments for Python.

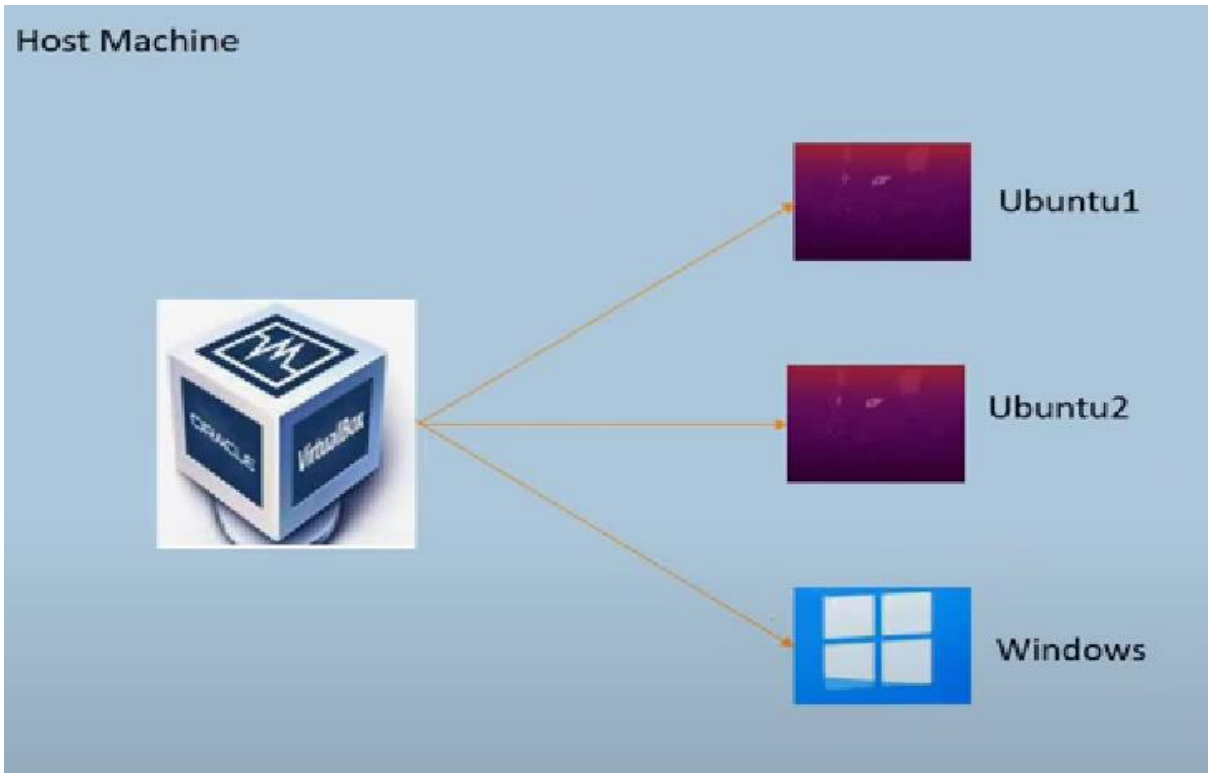
4) Anaconda: A package manager, environment manager, and distribution of Python and R packages that is used to create and manage virtual environments.

5) Virtual Environment: A self-contained environment that includes its own Python interpreter and dependencies, which is created using Anaconda.



6) SCP Command: A secure copy protocol that is used to transfer files between the two virtual machines.

7) Use Shared Folder Options in Oracle VM Box to share folders between host machine and windows machine



5.WORKING MODEL

1. Setting up the Virtual Machines

   Firstly, we need to download and install Oracle VM Box on our system. Once installed, we can create two virtual machines, one for Ubuntu and one for Windows. We will allocate a minimum of 2GB of RAM and 20GB of hard disk space for each VM.

2. Installing Anaconda in the Virtual Machines

   Next, we will download and install Anaconda on both virtual machines. We will use the terminal in Ubuntu and command prompt in Windows to download and install Anaconda. After the installation, we will create a new virtual environment for Python using the conda command.

3. Transferring Files between Virtual Machines

   To transfer files between the two VMs, we will use the SCP command. First, we need to ensure that both VMs are connected to the same network. We will then use the SCP command to transfer files from one VM to another. The syntax for the SCP command is as follows:

   scp /path/to/local/file username@IP:/path/to/remote/file

   Here, we will replace /path/to/local/file with the local path of the file we want to transfer, username with the username of the remote VM, IP with the IP address of the remote VM, and /path/to/remote/file with the path of the file on the remote VM.

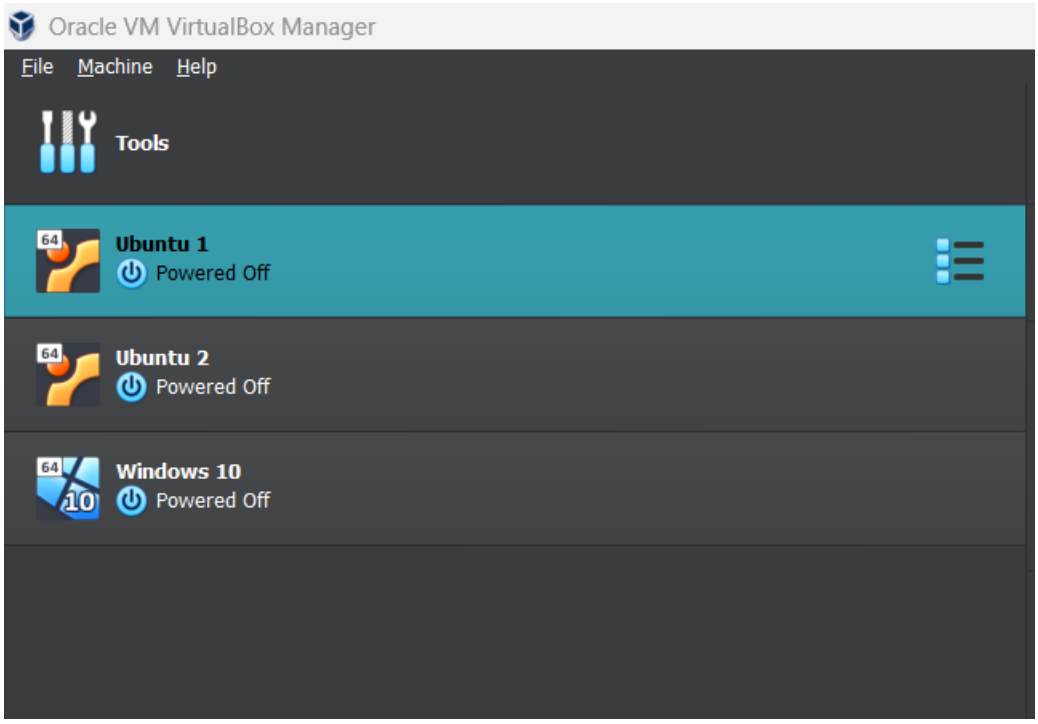
4. Transferring Files between Virtual Machines and Host Machine

   Use Shared Folder Options in Oracle VM Box to share folders between host machine and windows machine.

5. Creating Python Marks Entry app in shared folder

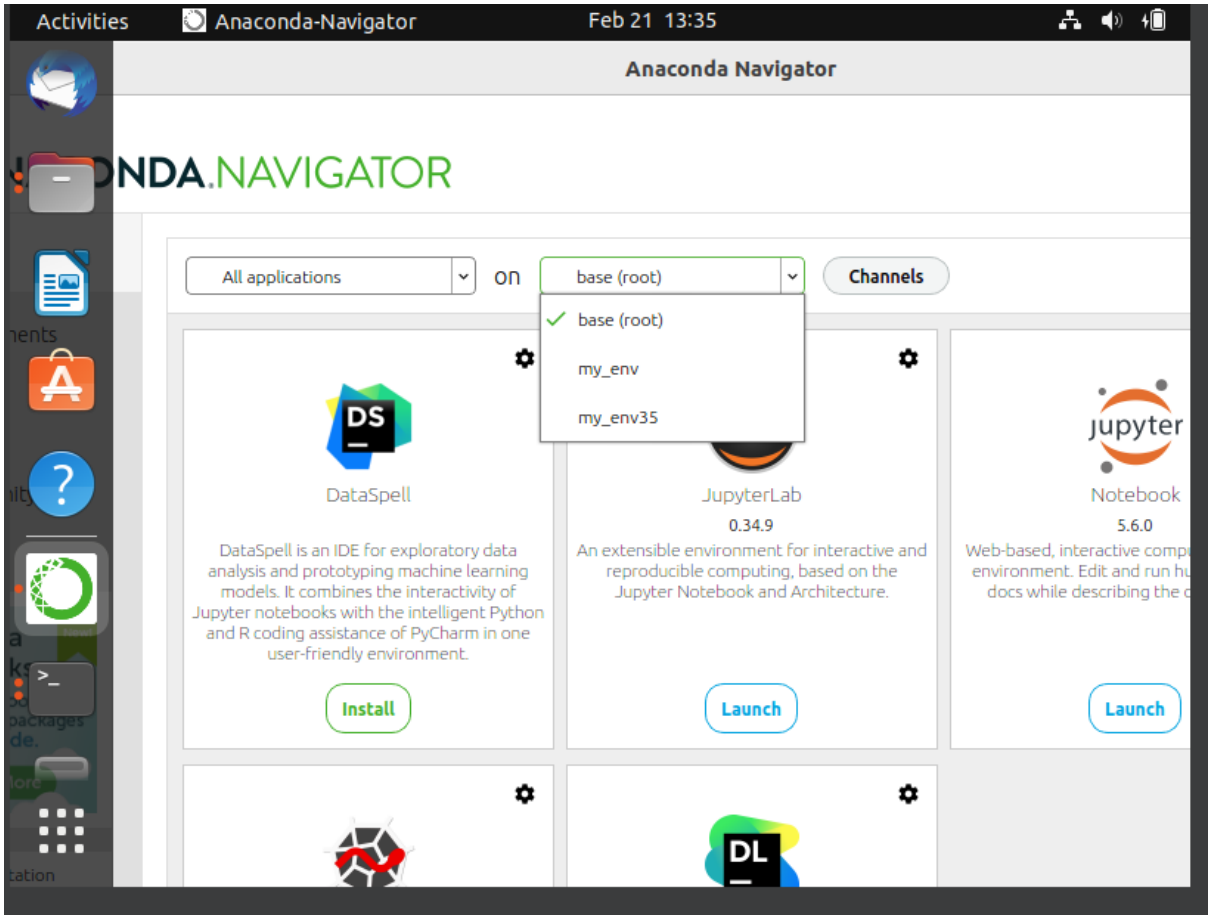
6. EXPLANATION AND SCREENSHOT

1. Setting up the Virtual Machines



2. Installing Anaconda in the Virtual Machines and Creating Virtual Environments

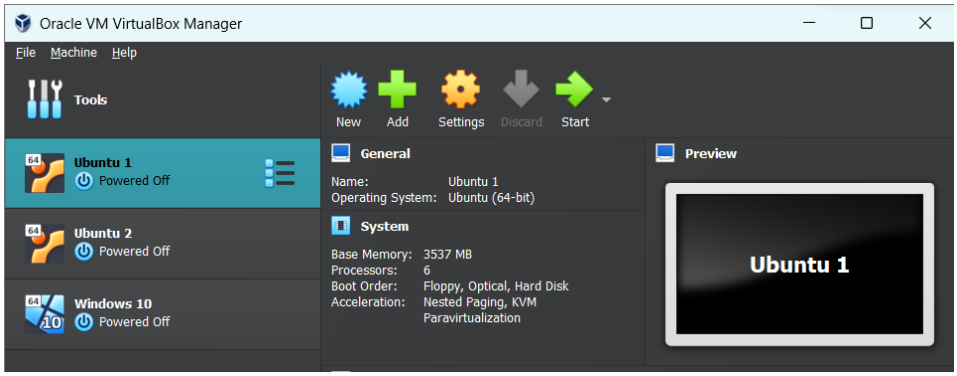
```
(base) sanjil@Sanjil:~$ conda info --envs
# conda environments:
#
base                  *  /home/sanjil/anaconda3
my_env                   /home/sanjil/anaconda3/envs/my_env
my_env35                 /home/sanjil/anaconda3/envs/my_env35
```


Activities
Anaconda-Navigator
Feb 21 13:35
Anaconda Navigator
NDA.NAVIGATOR
All applications
on
base (root)
Channels
base (root)
my_env
my_env35
DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.
Install
JupyterLab
0.34.9
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
Notebook
5.6.0
Launch

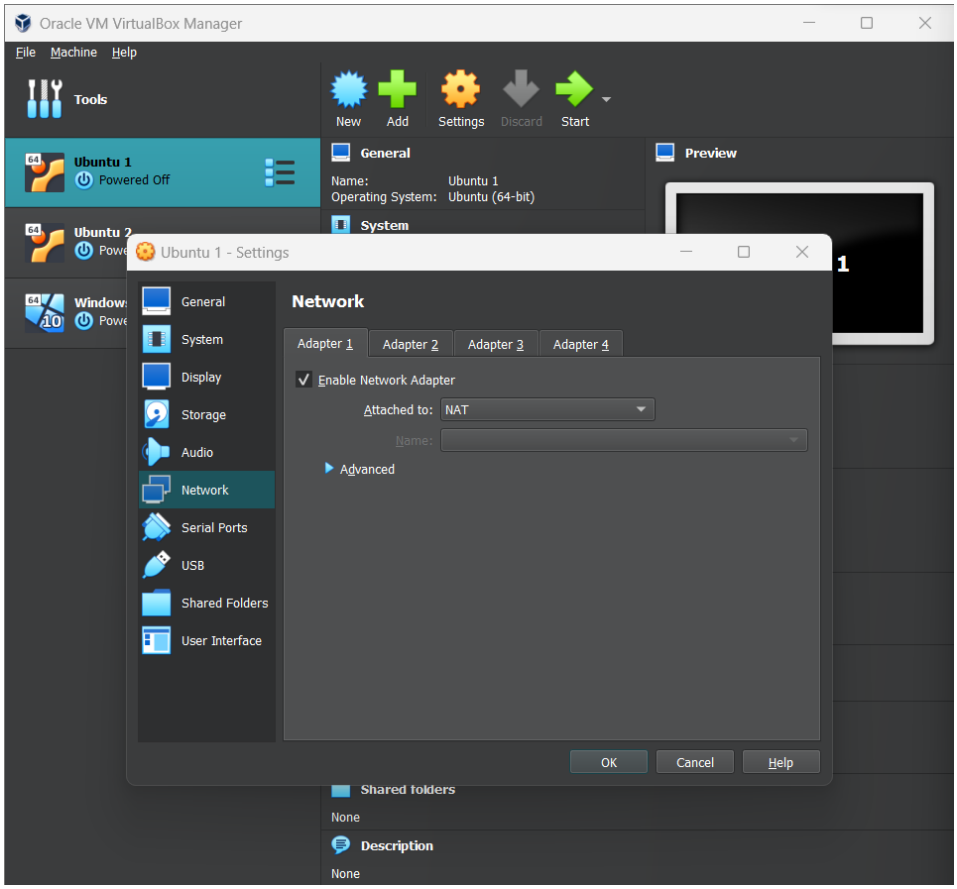
3. Transferring Files between Virtual Machines

a) Creating NAT NETWORK

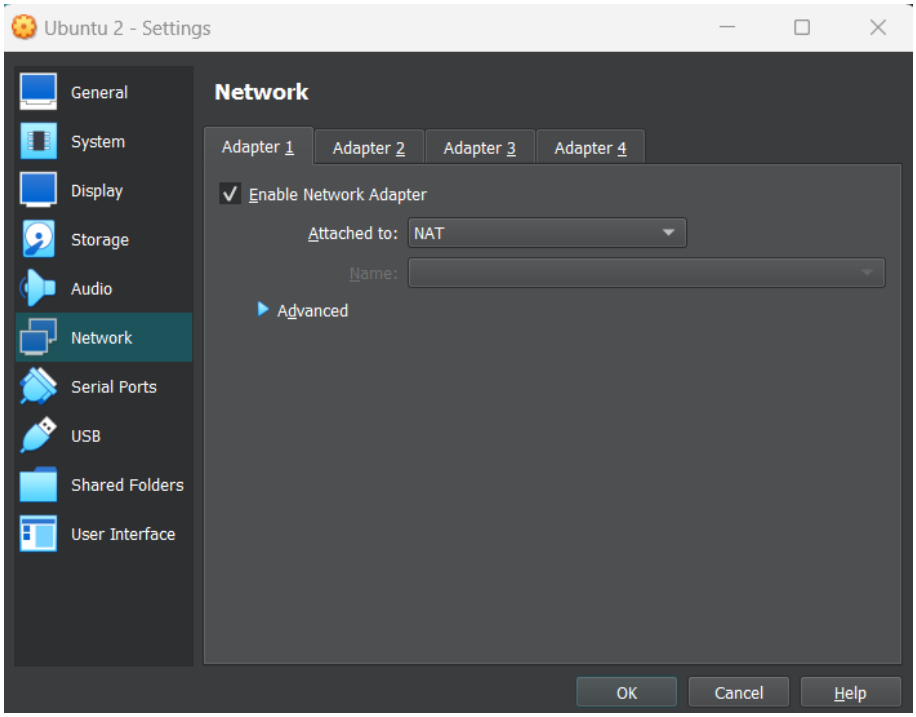
STEP 1 : Check Network of three VM's are attached to NAT



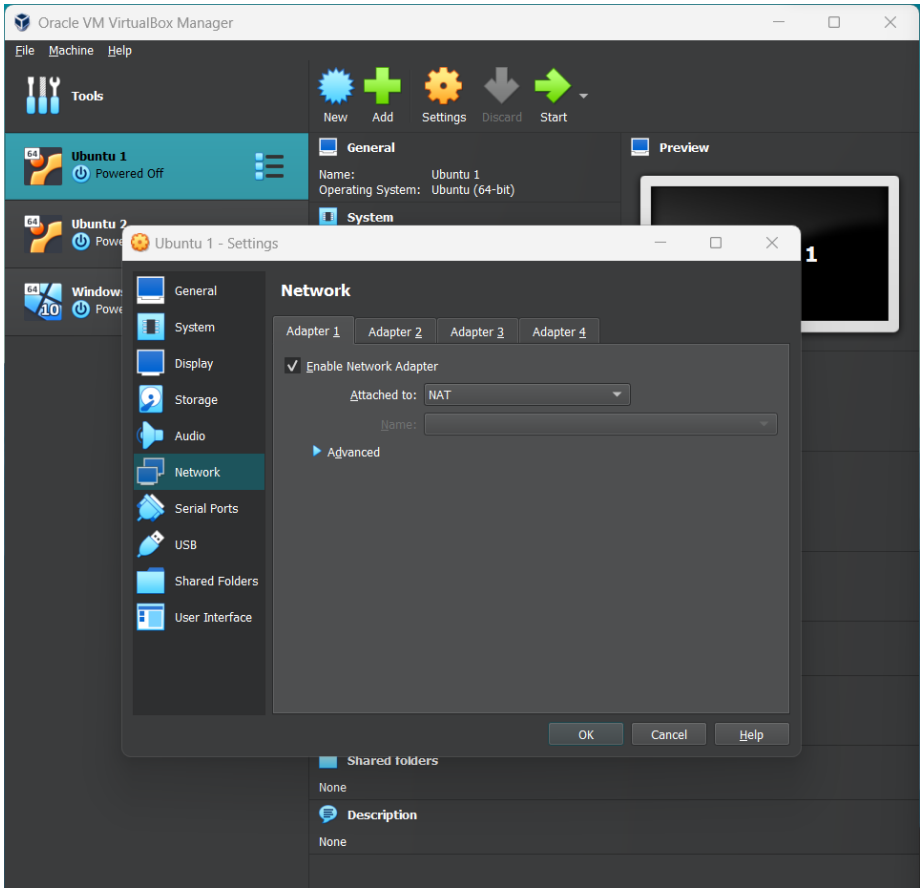
1.1 Ubuntu 1



1.2 Ubuntu 2

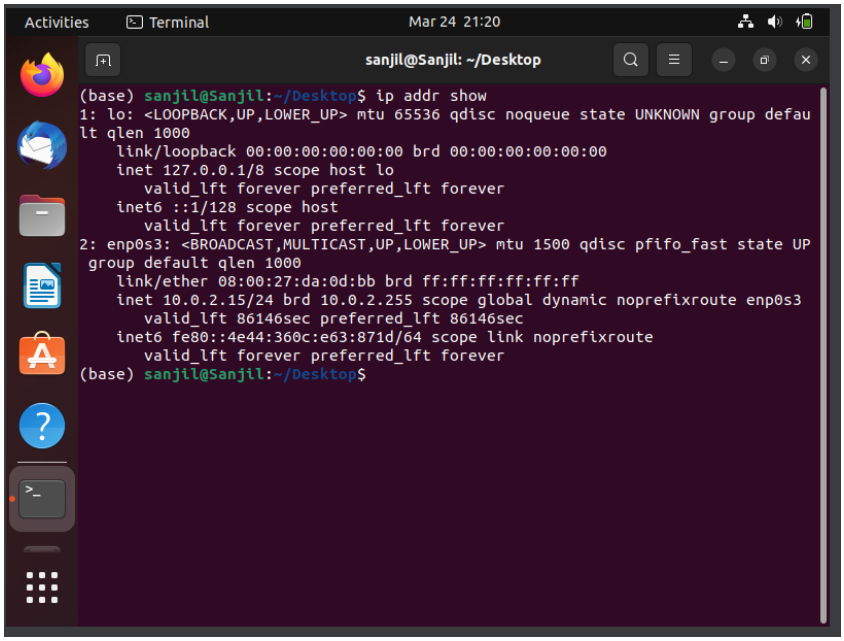


1.3 Windows 10

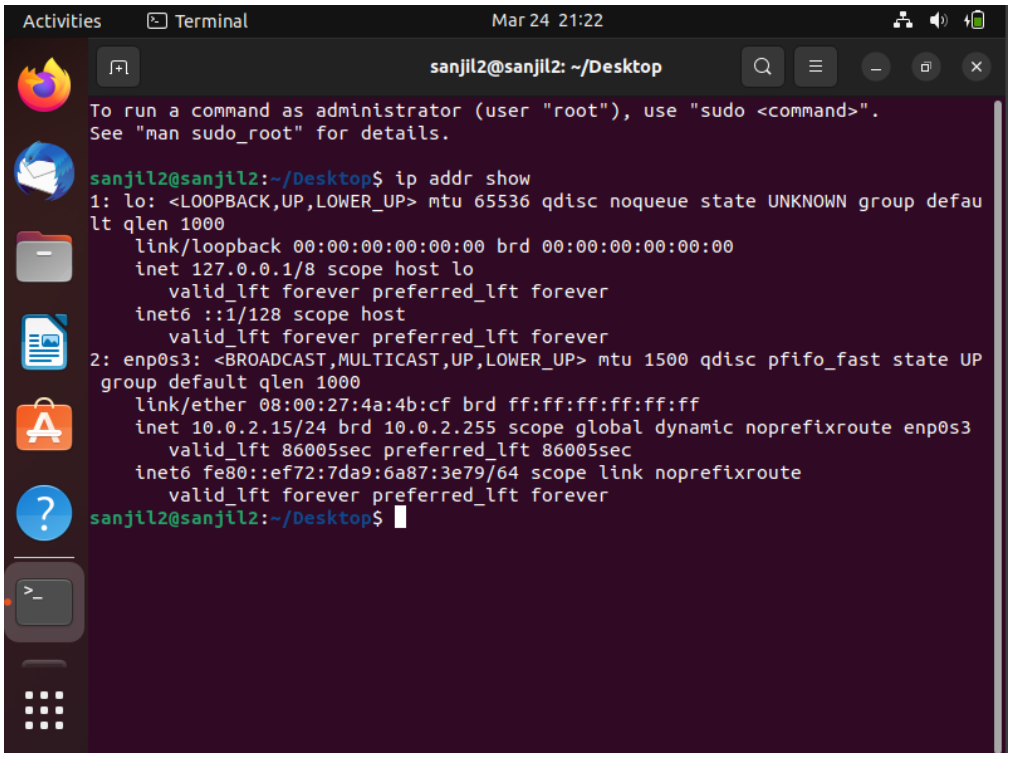


STEP 2 : Check IP address for three virtual Machine

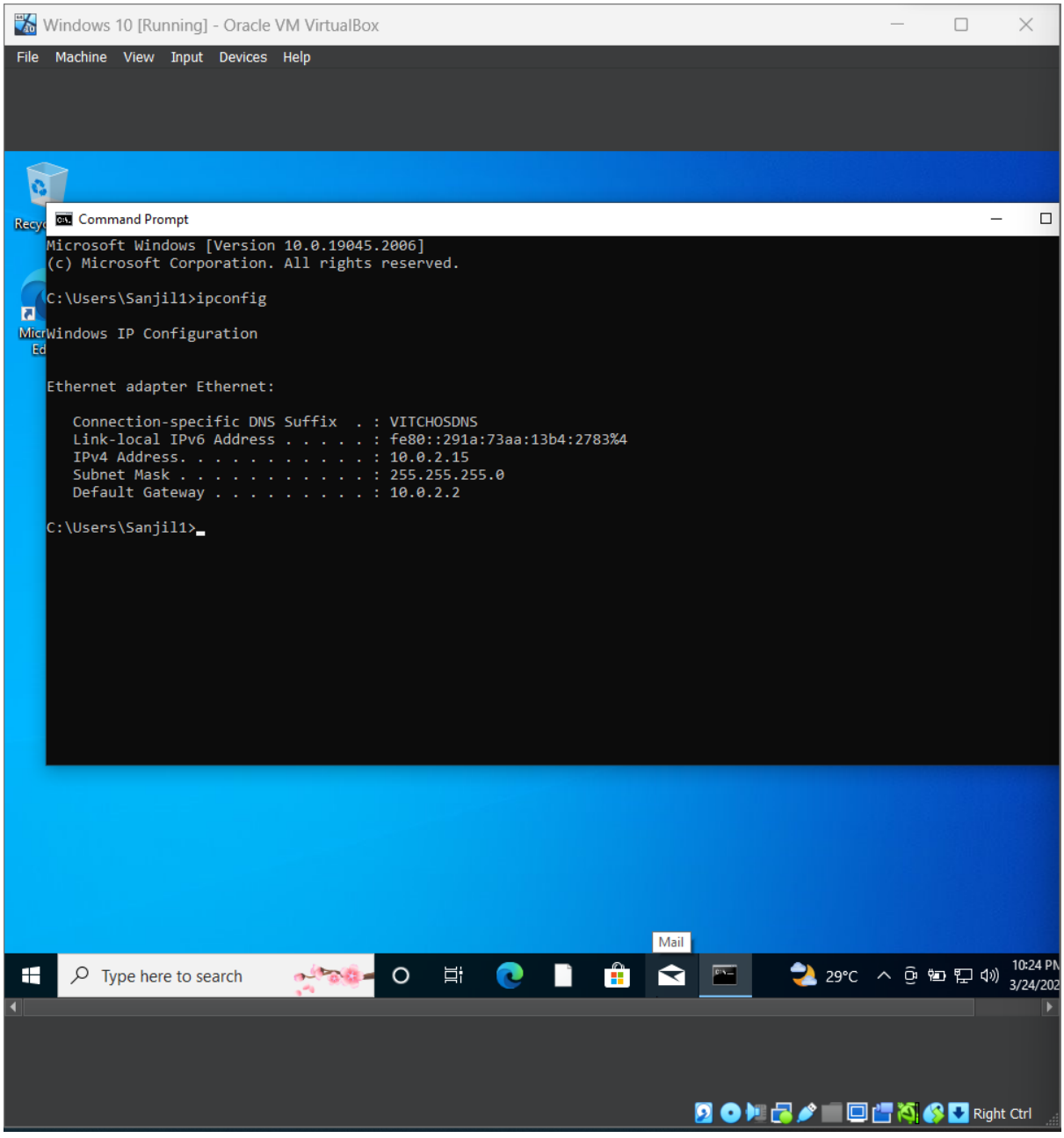
2.1 Ubuntu 1



2.2 Ubuntu 2

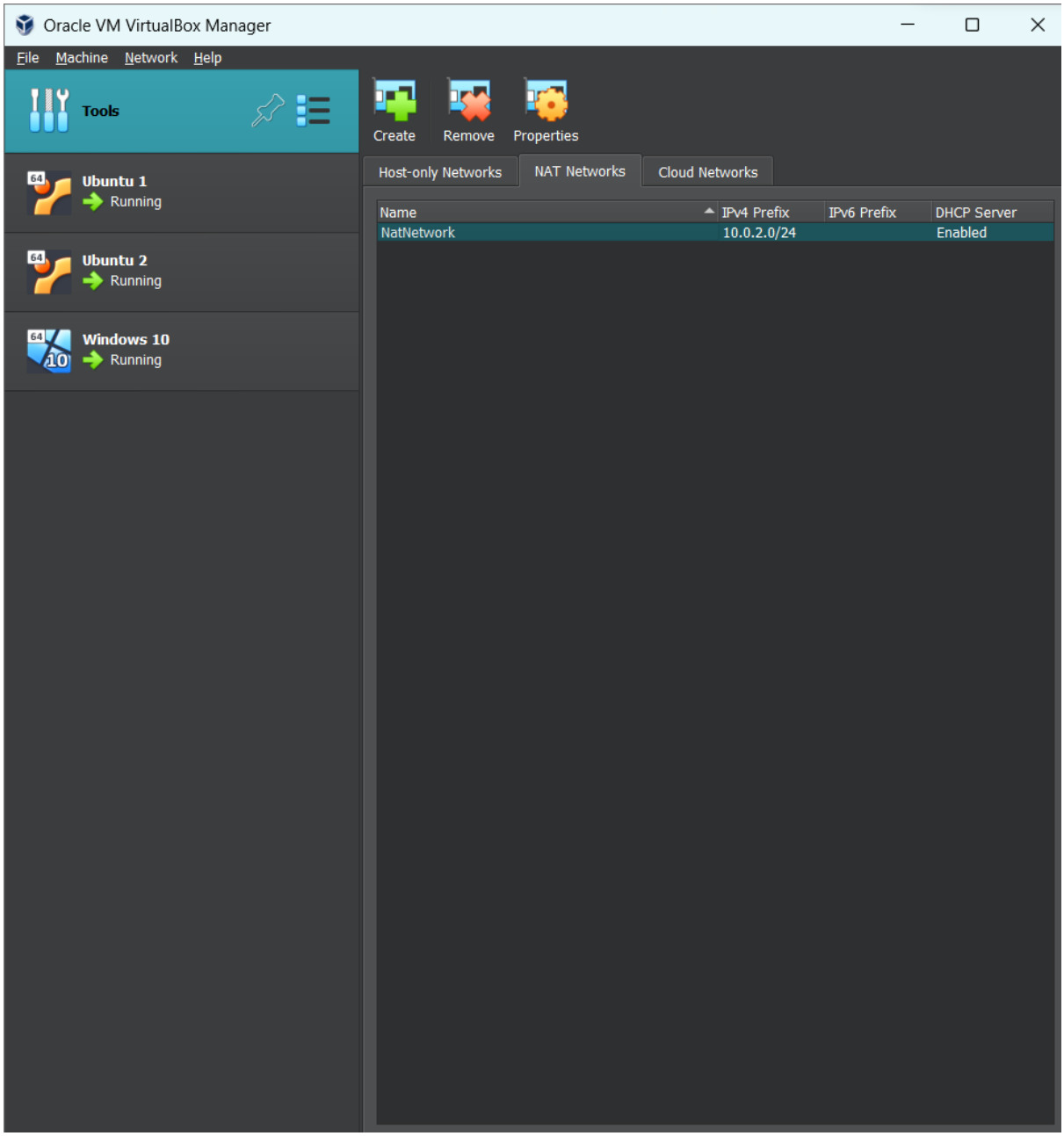


2.3 Windows10

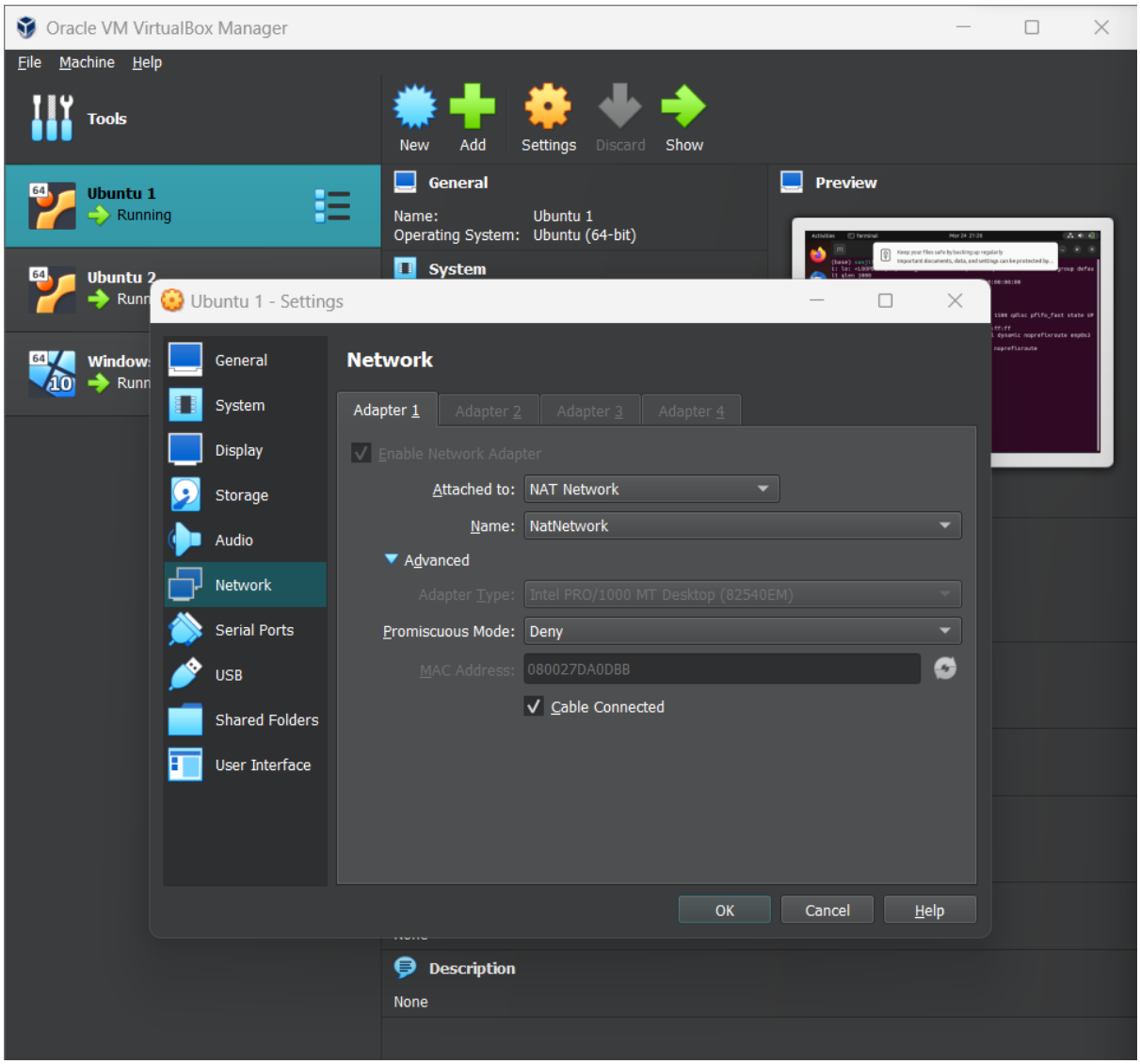


The IP address for three virtual machines is same so create a new NAT Network and assign this NAT network to this three VM's

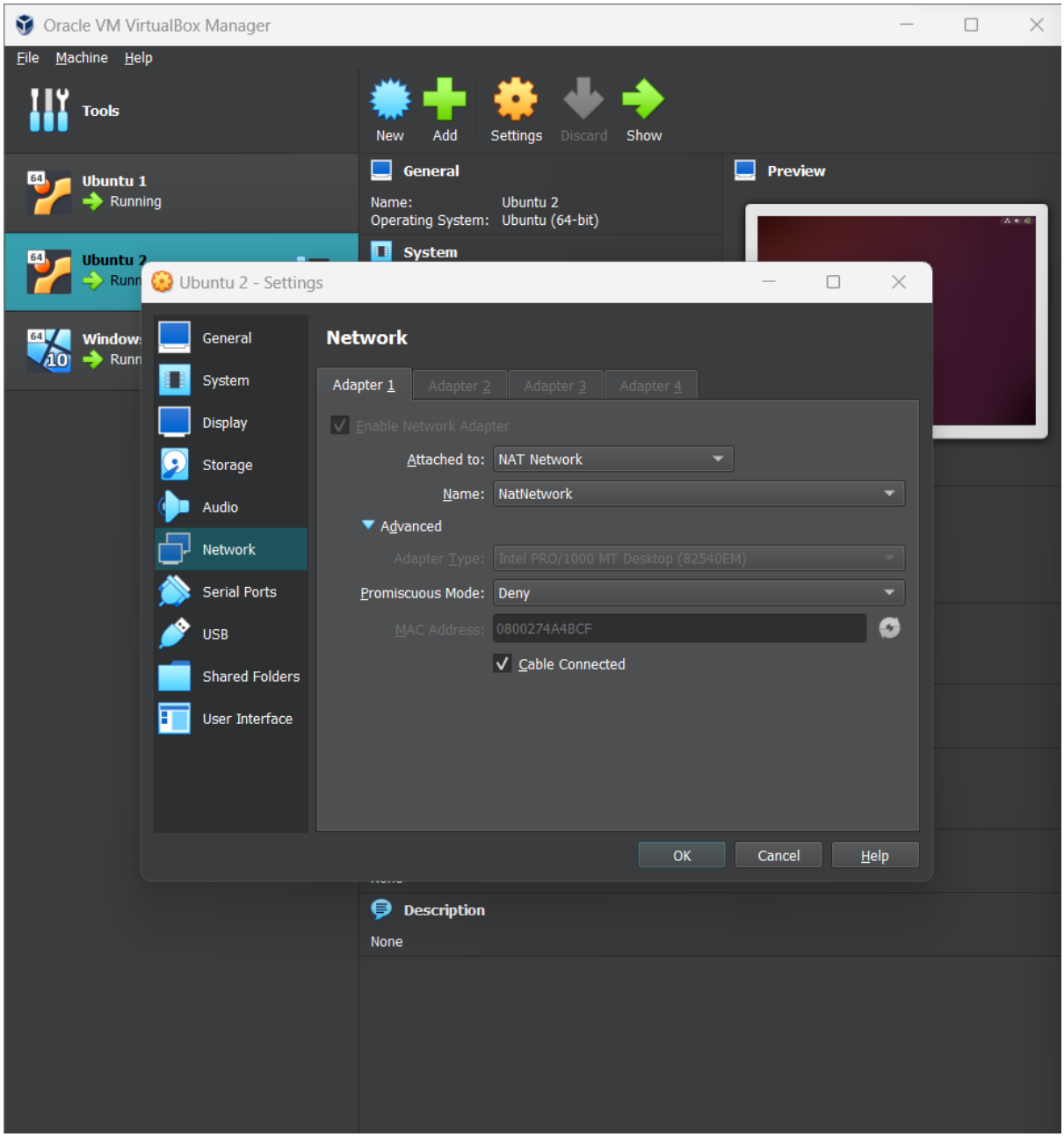
STEP 3 : Create a new NAT Network and assign this NAT network to this three VM's



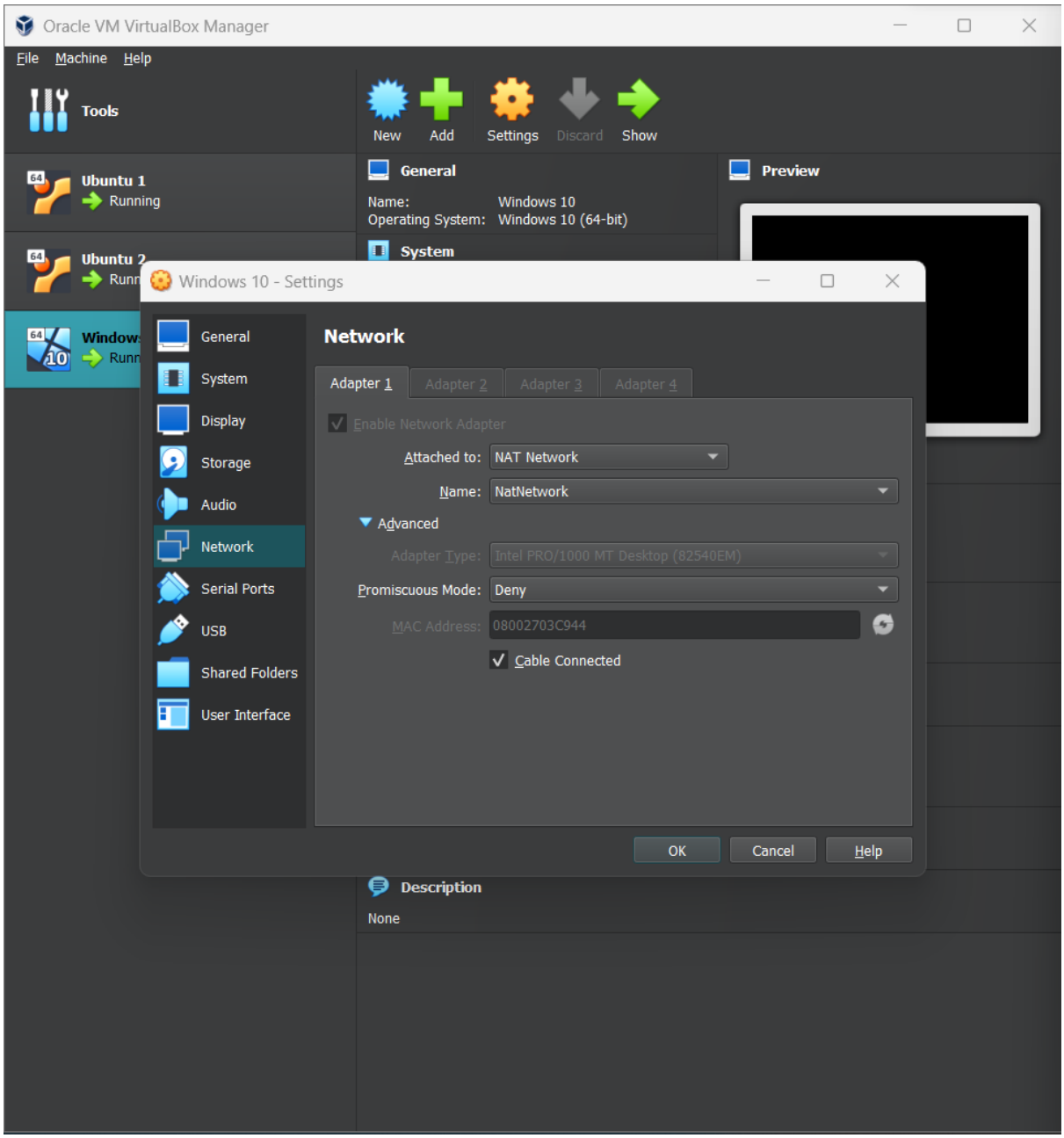
3.1 Ubuntu 1


Oracle VM VirtualBox Manager
File Machine Help
Tools
New Add Settings Discard Show
Ubuntu 1
Running
Ubuntu 2
Windows
General
Name: Ubuntu 1
Operating System: Ubuntu (64-bit)
System
Preview
Ubuntu 1 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Network
Adapter 1
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to: NAT Network
Name: NatNetwork
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 080027DA0DBB
Cable Connected
OK
Cancel
Help
Description
None

3.2 Ubuntu 2

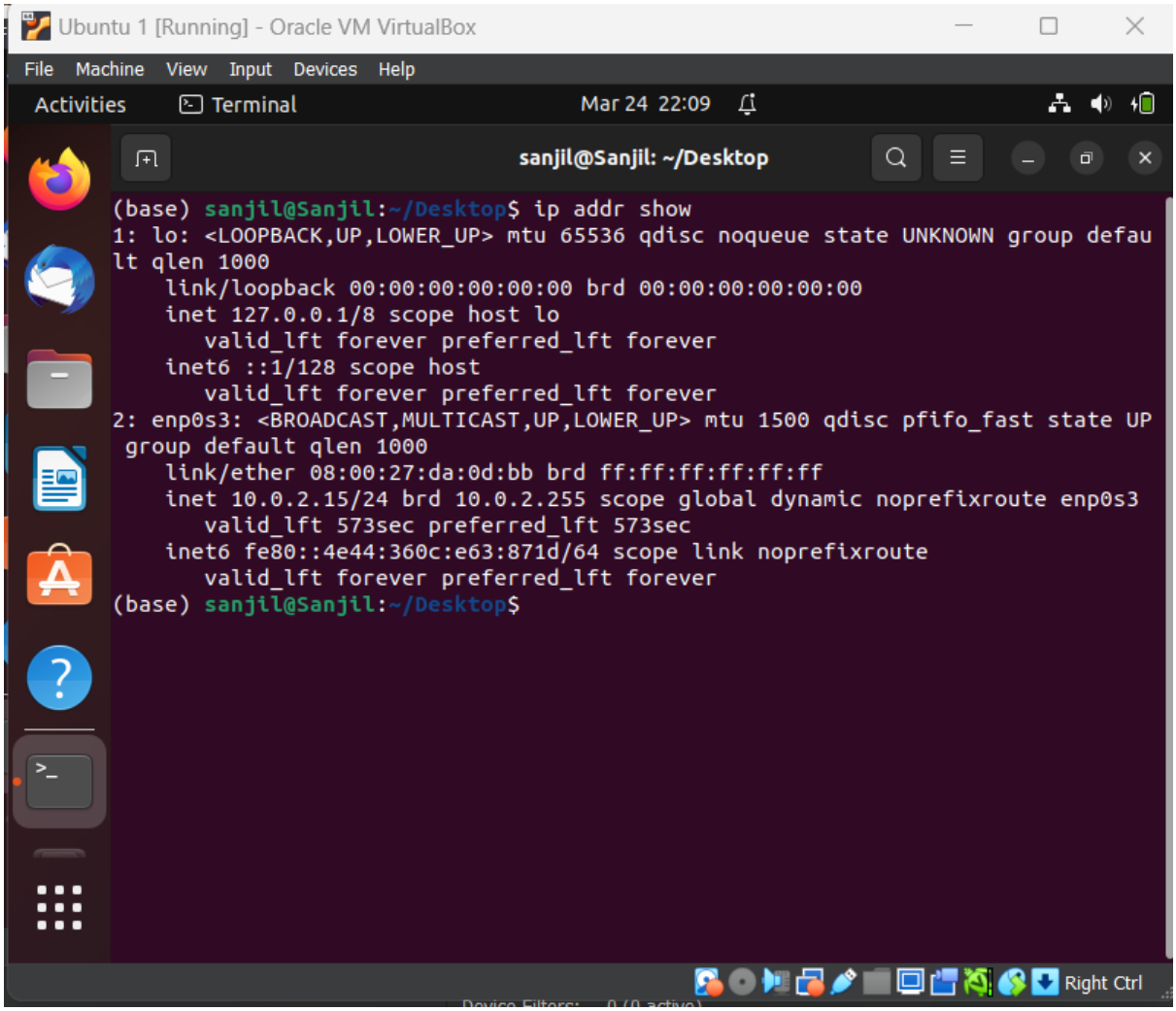

Oracle VM VirtualBox Manager
File Machine Help
Tools
New Add Settings Discard Show
Ubuntu 1
Running
Ubuntu 2
Windows
General
Name: Ubuntu 2
Operating System: Ubuntu (64-bit)
System
Preview
Ubuntu 2 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Network
Adapter 1
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to: NAT Network
Name: NatNetwork
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 0800274A4BCF
Cable Connected
OK
Cancel
Help
Description
None

3.3 Windows10

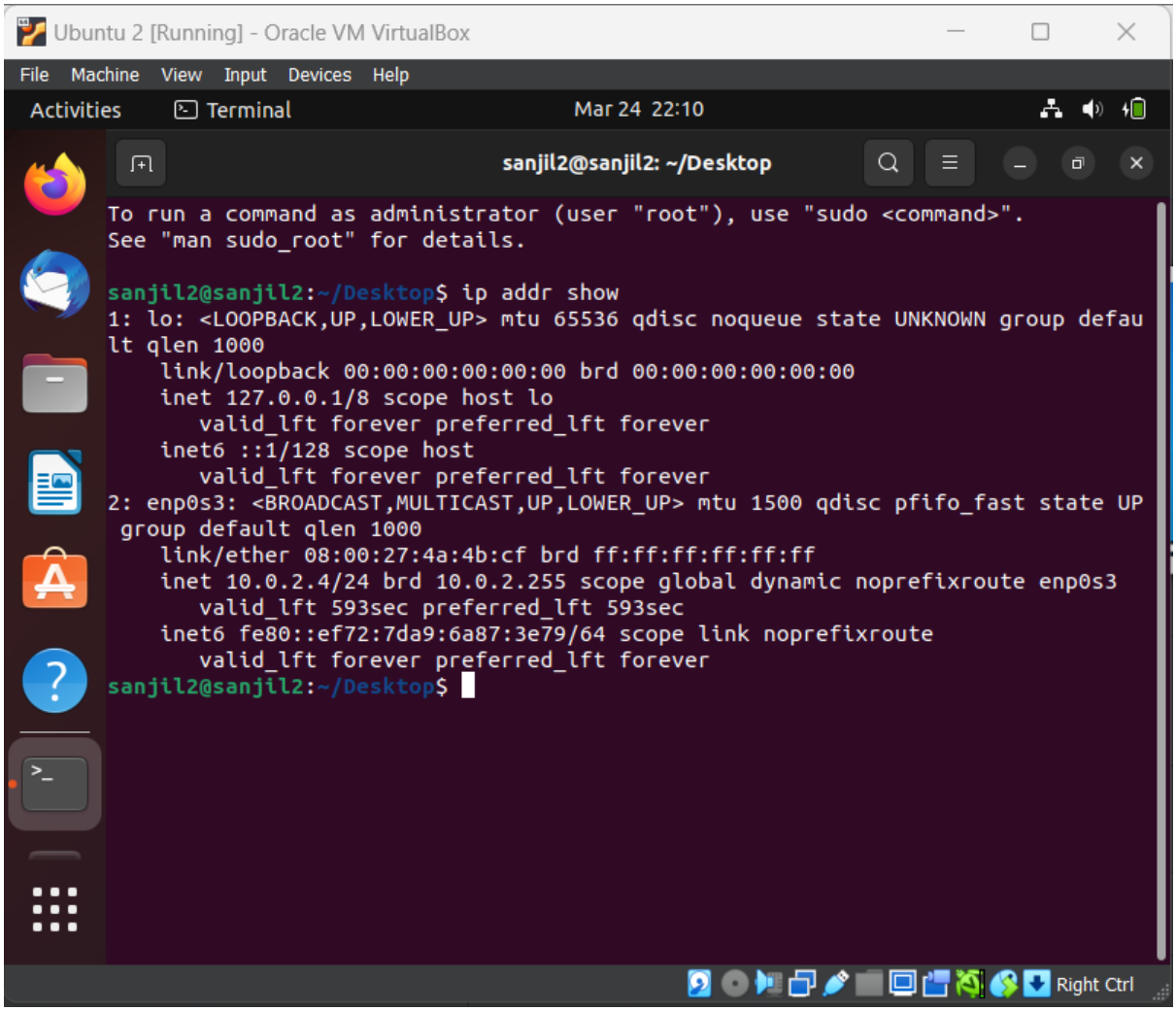


STEP 4 : Now again check IP address for three virtual Machine

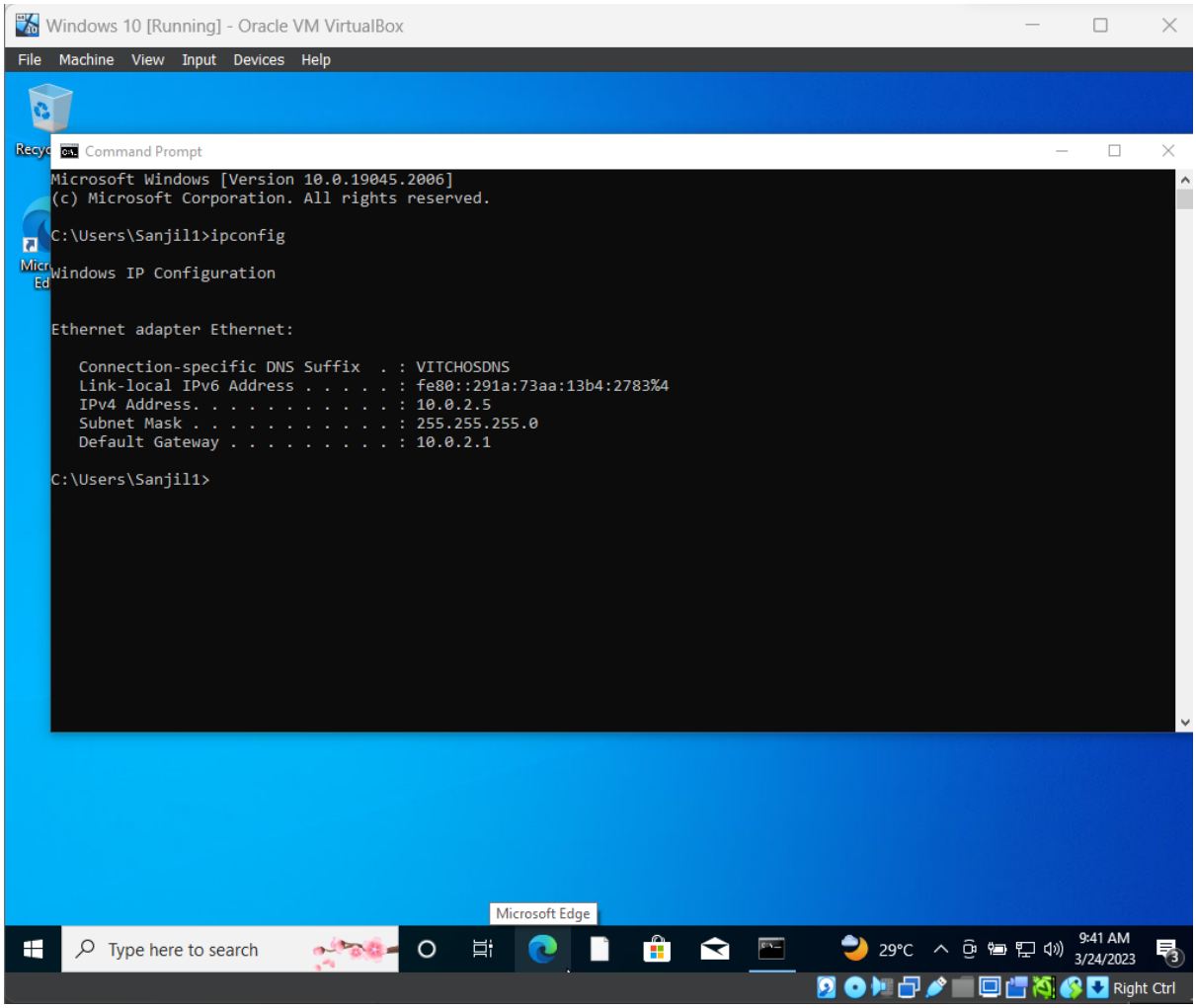
4.1 Ubuntu 1



4.2 Ubuntu 2

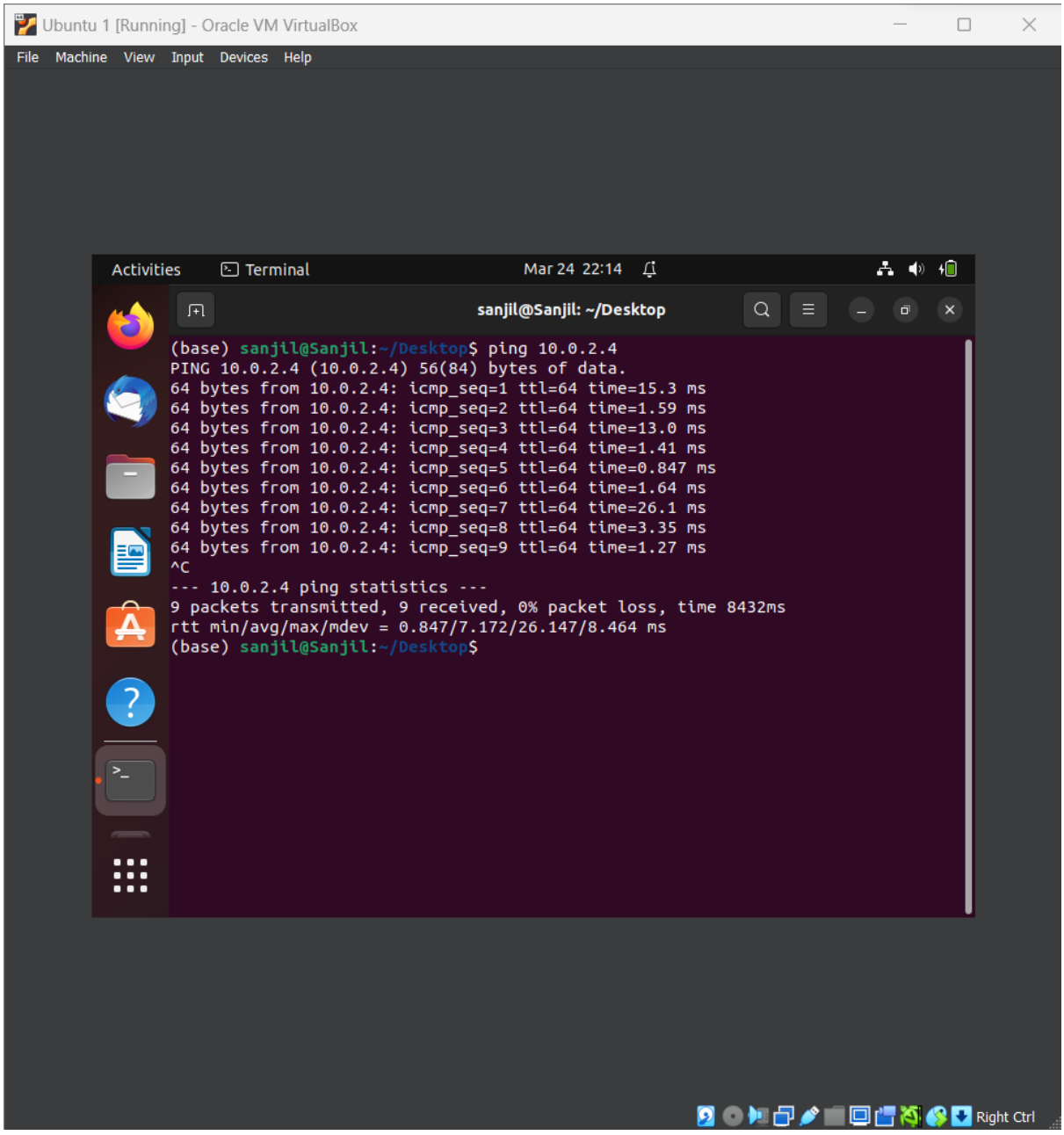

Ubuntu 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Mar 24 22:10
sanjil2@sanjil2: ~/Desktop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
sanjil2@sanjil2:~/Desktop$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
link/ether 08:00:27:4a:4b:cf brd ff:ff:ff:ff:ff:ff
inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
valid_lft 593sec preferred_lft 593sec
inet6 fe80::ef72:7da9:6a87:3e79/64 scope link noprefixroute
valid_lft forever preferred_lft forever
sanjil2@sanjil2:~/Desktop$
Right Ctrl

4.3 Windows10



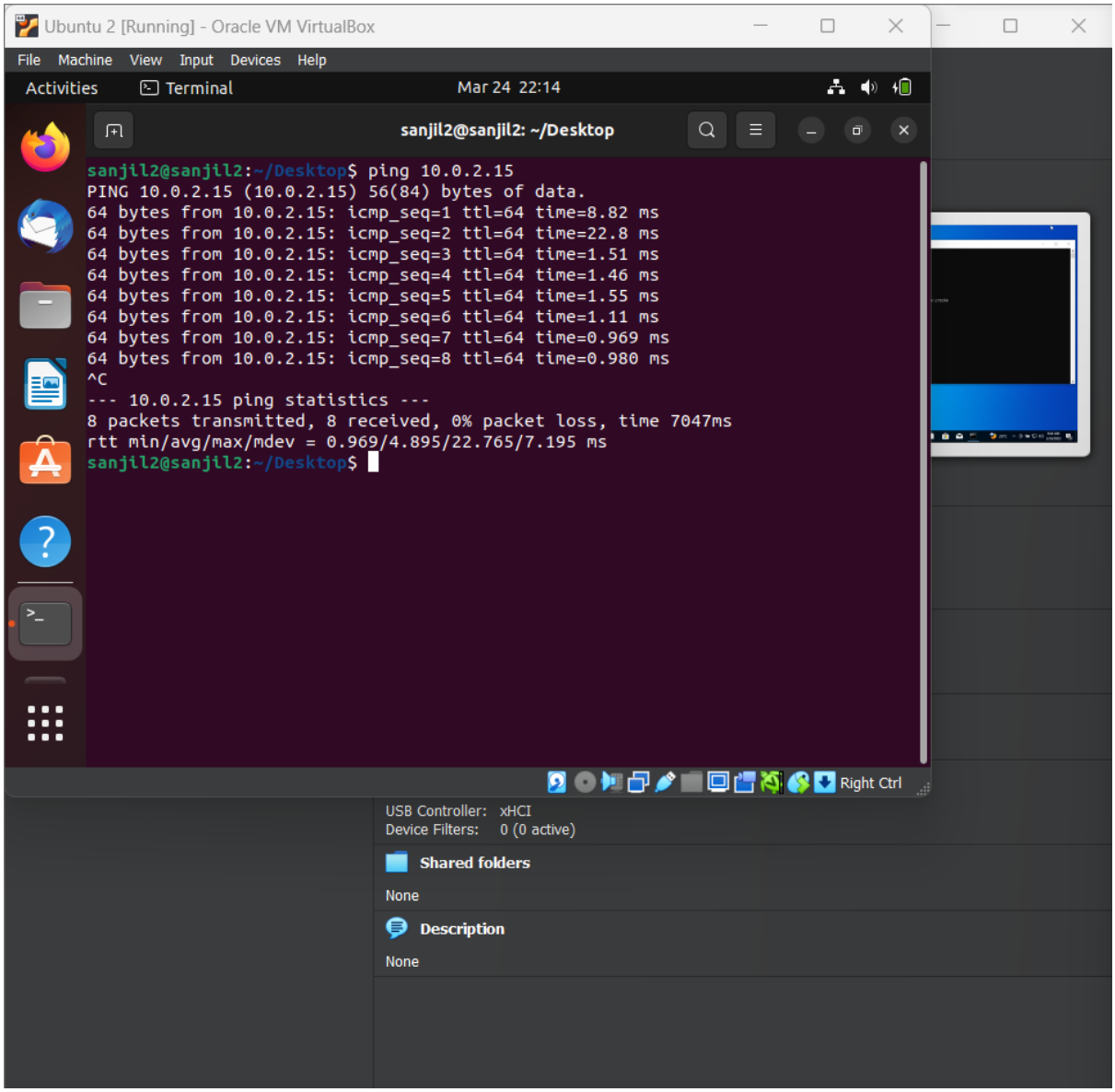
As we can see now there three different IP for three different VM's

STEP 5 : Now try to communicate by pinging Ubuntu 1 to Ubuntu 2



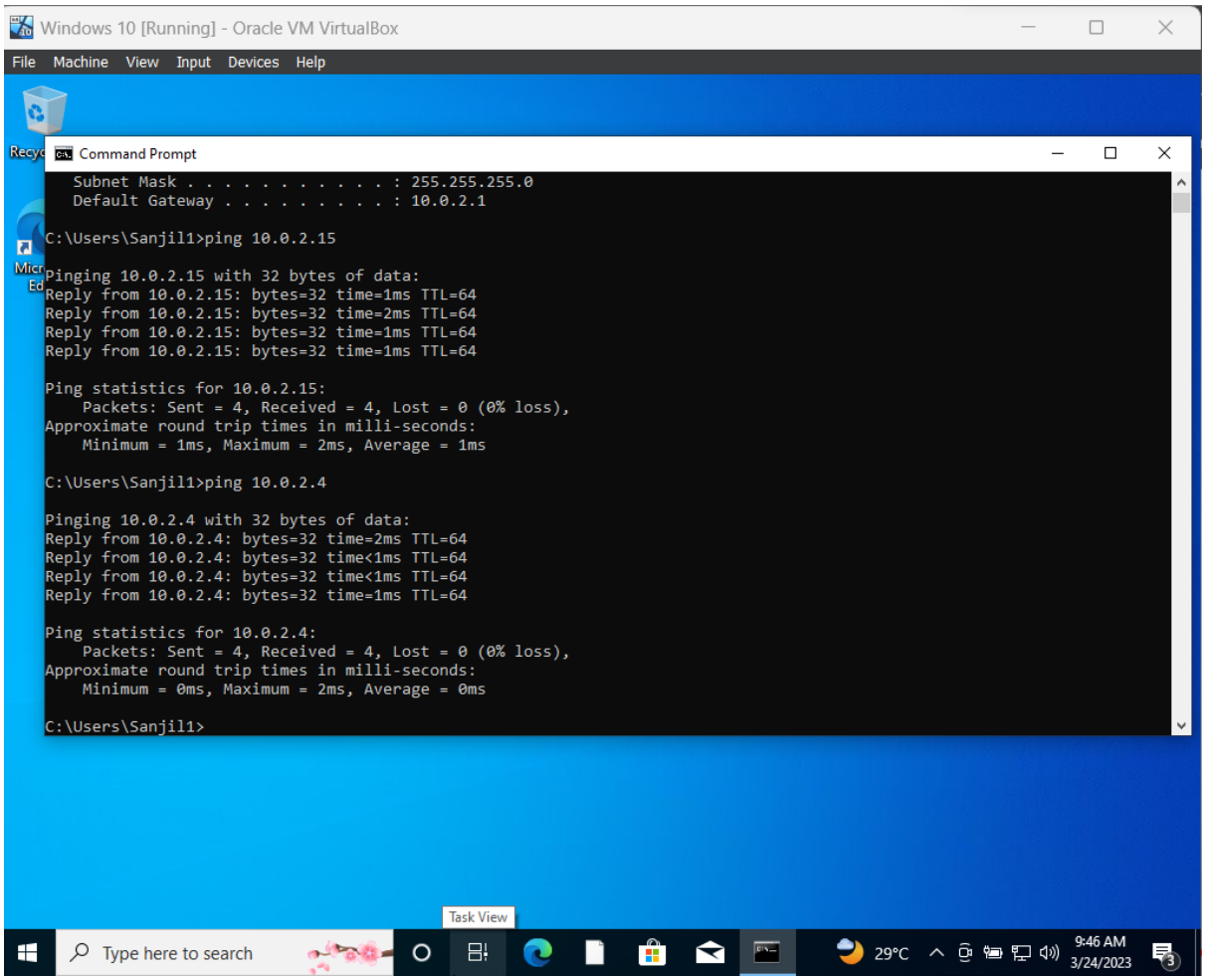
It’s Working We got reply from Ubuntu 2

STEP 6 : Now try to communicate by pinging Ubuntu 2 to Ubuntu 1



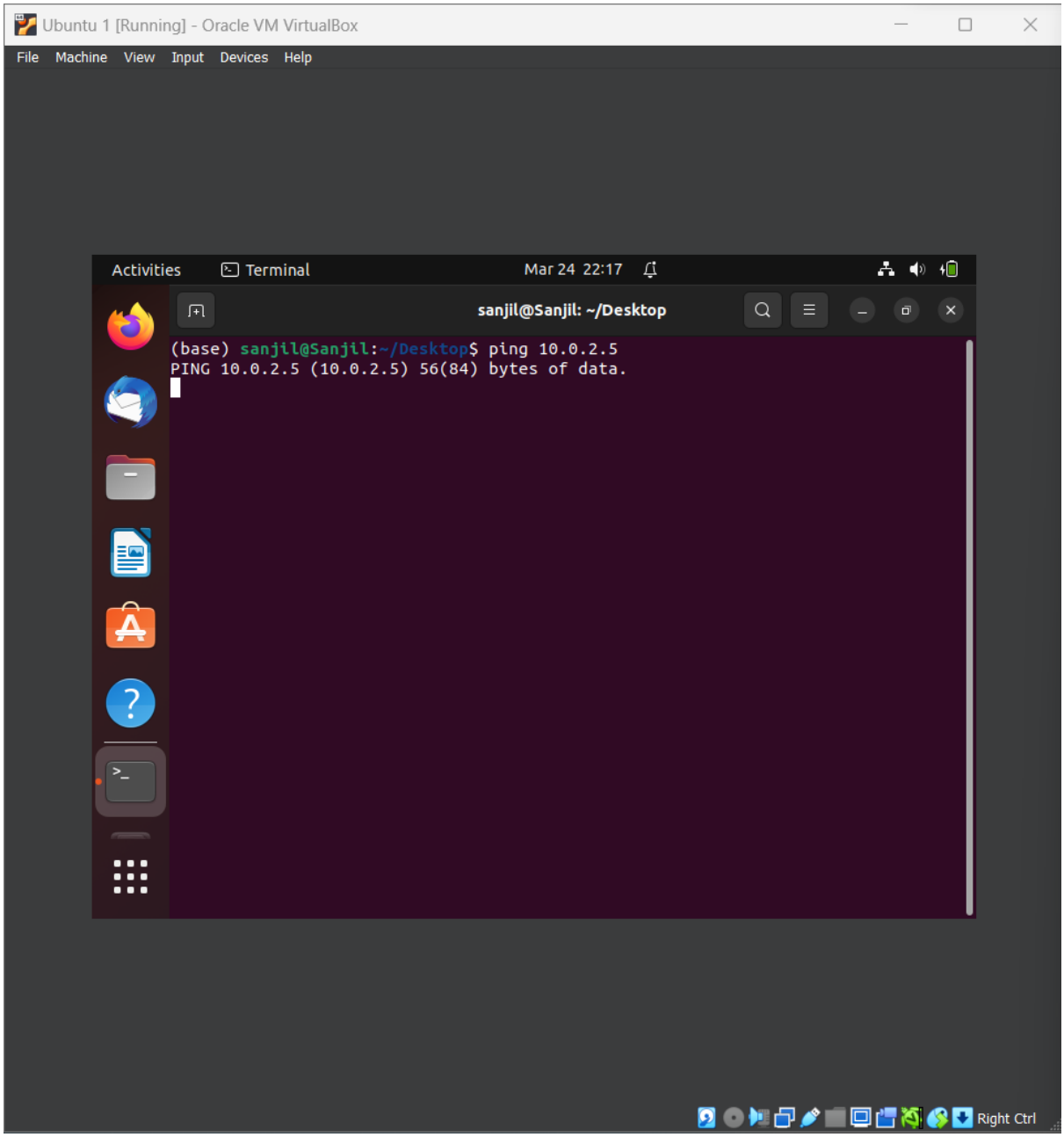
It’s Working We got reply from Ubuntu 1

STEP 7 : Now try to communicate by pinging Windows 10 to Ubuntu 1 and Ubuntu 2



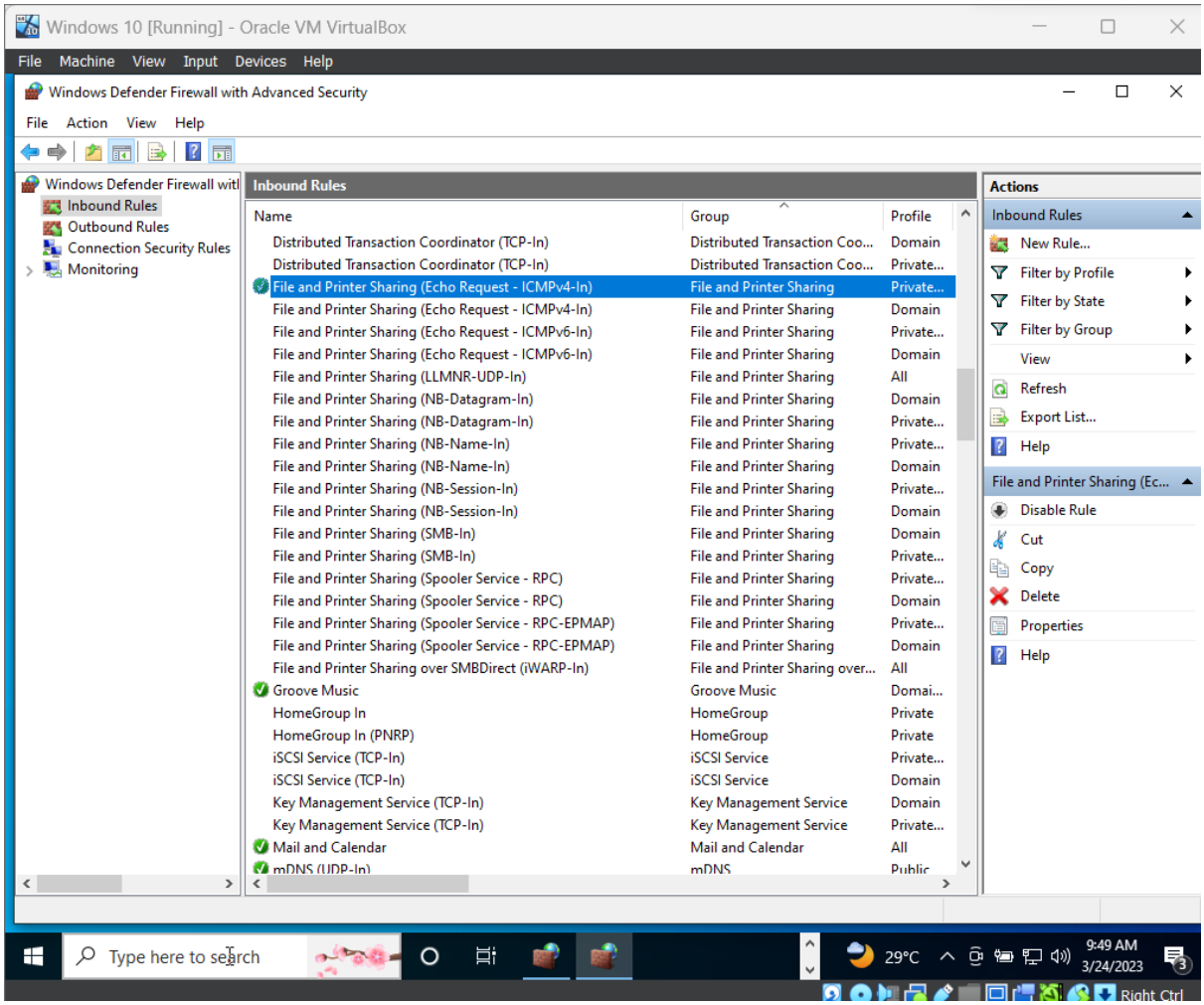
It’s Working We got reply from Ubuntu 1 and Ubuntu 2

STEP 8 : Now try to communicate by pinging Ubuntu 1 to Windows10

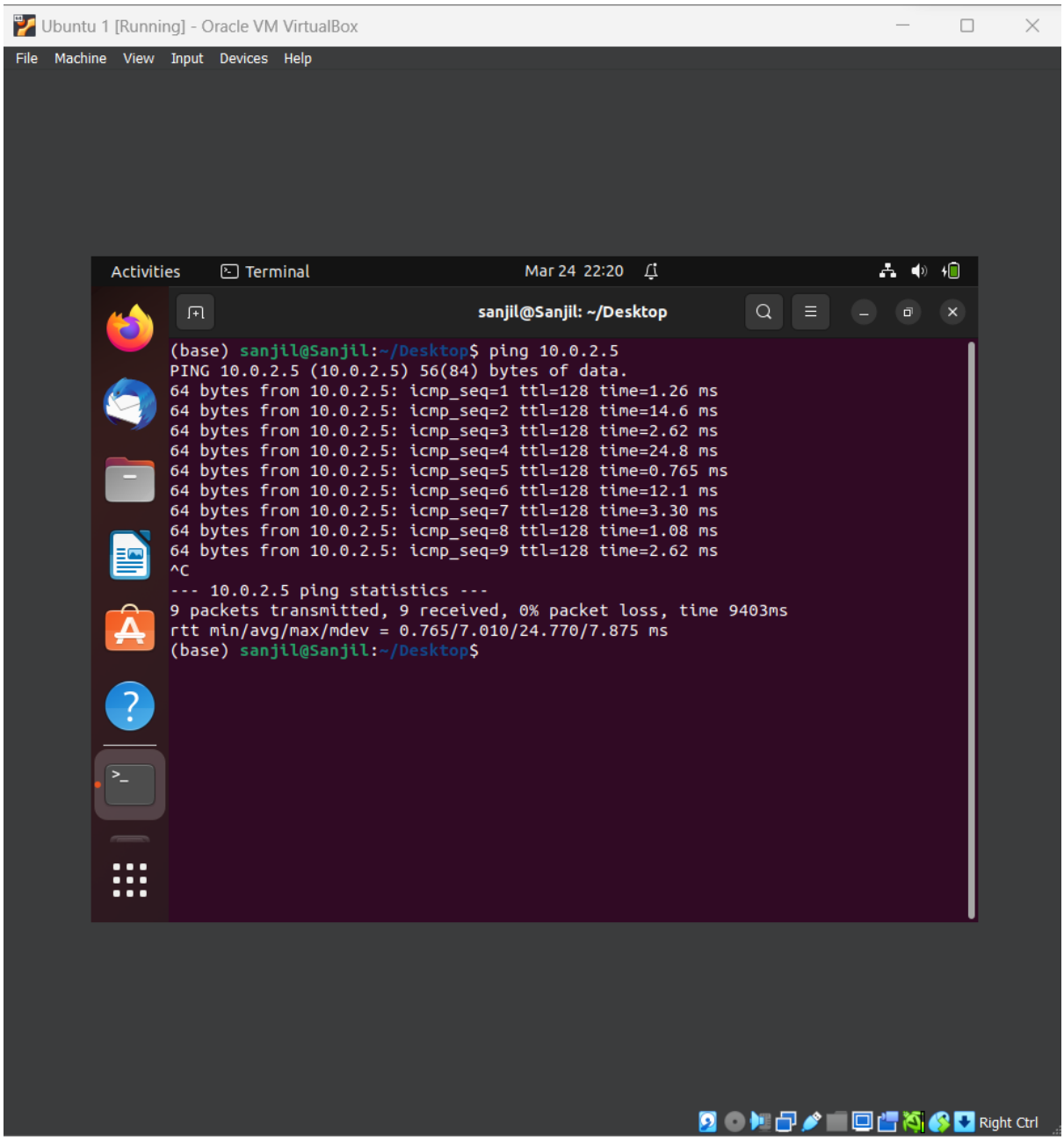


It’s not working because the inbuilt firewall of windows machine.

STEP 9 : Enable the inbound Rules for communication from ubuntu1 to windows10

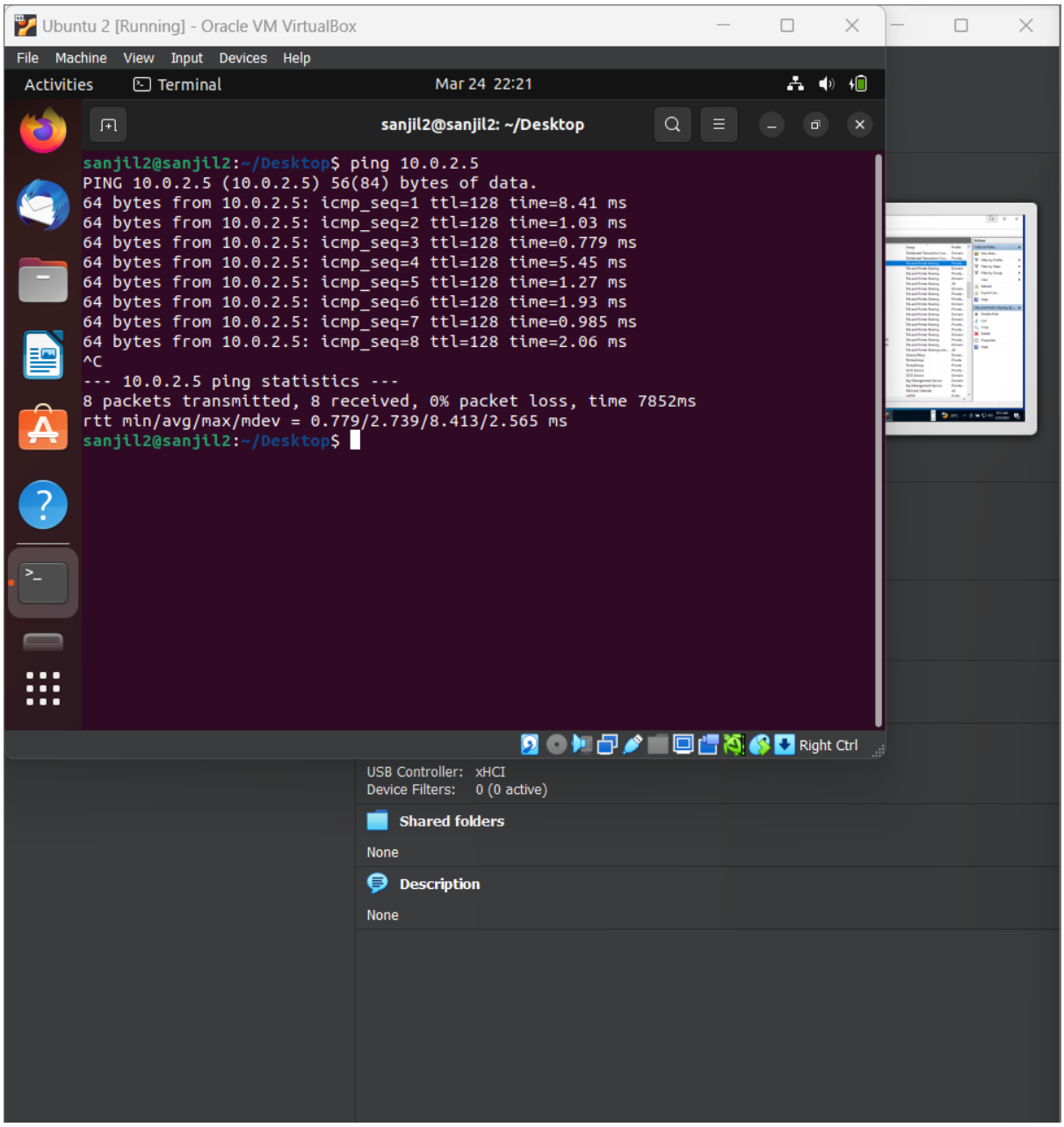


STEP 10 : Now try to communicate by pinging Ubuntu 1 to Windows10



It’s Working We got reply from Windows10

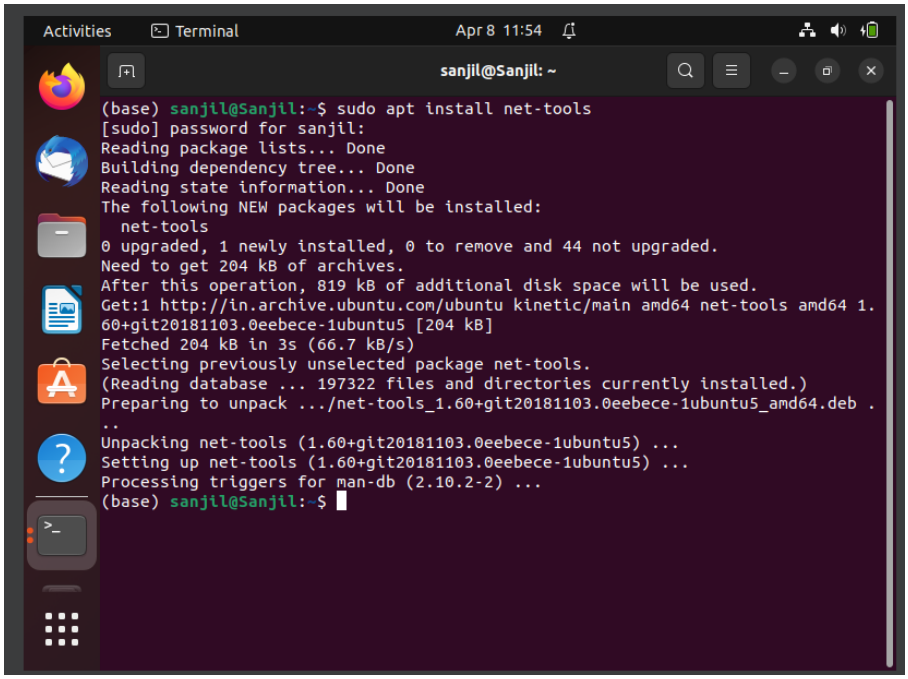
STEP 11 : Now try to communicate by pinging Ubuntu 2 to Windows10



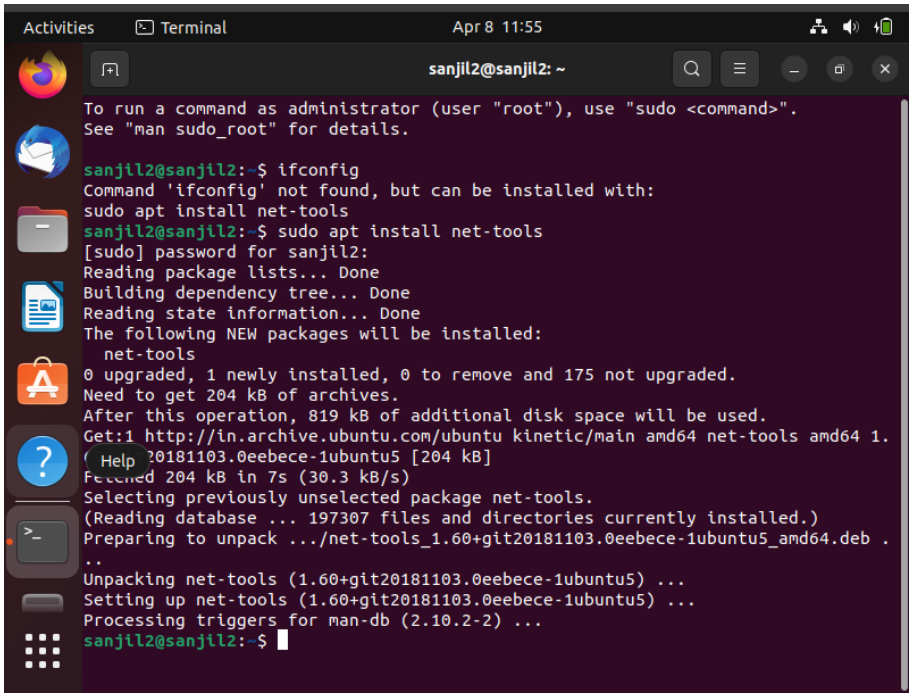
It’s Working We got reply from Windows10

b) Transferring File Between VM's

STEP 1 : Install NAT tools in ubuntu 1

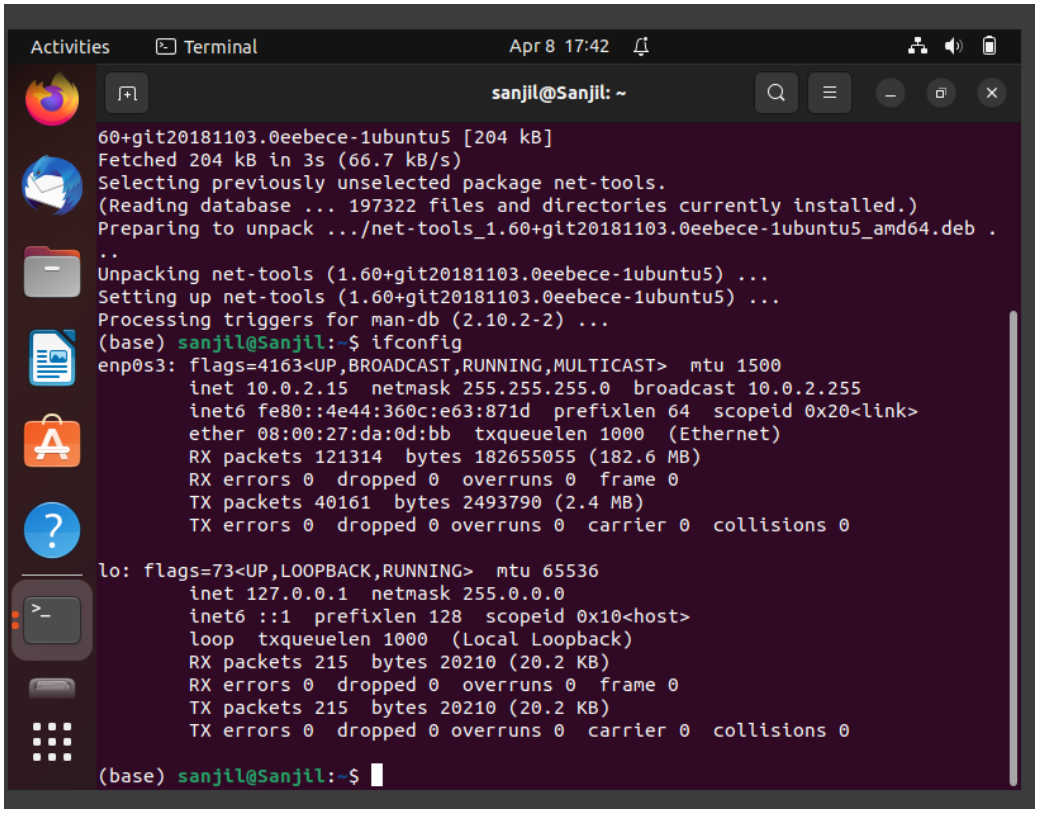


STEP 2 : Install NAT tools in ubuntu 2

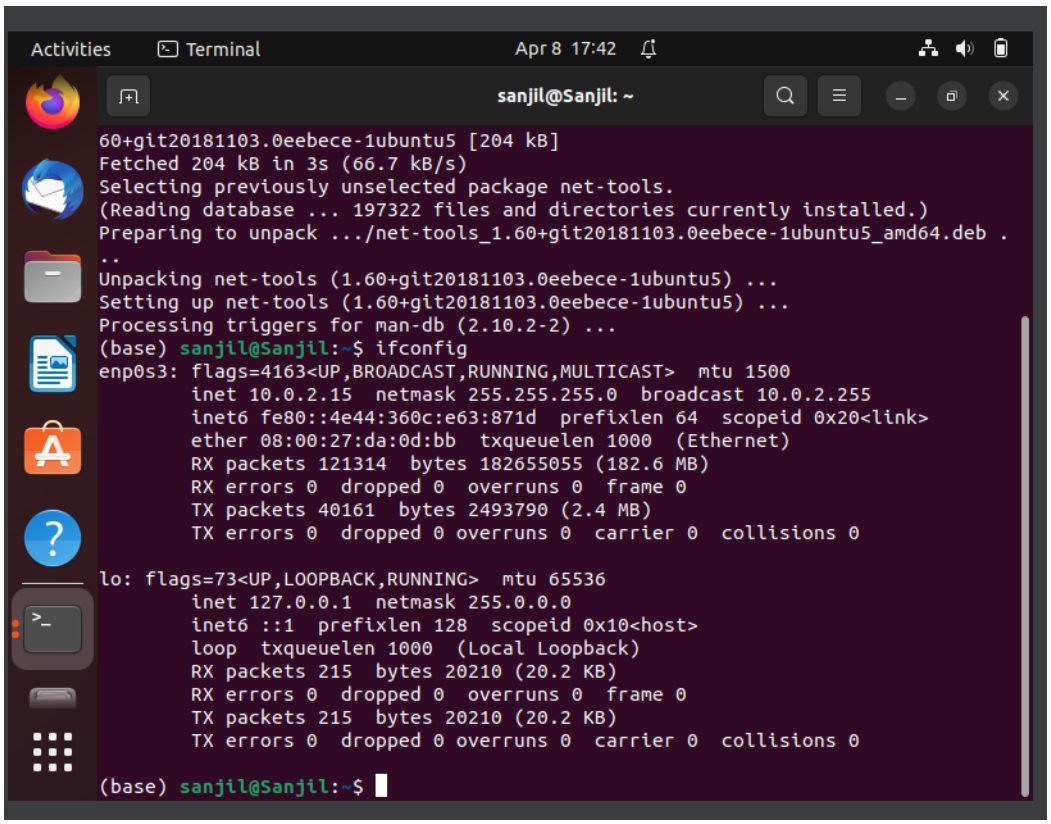


STEP 3: Checking with ping command

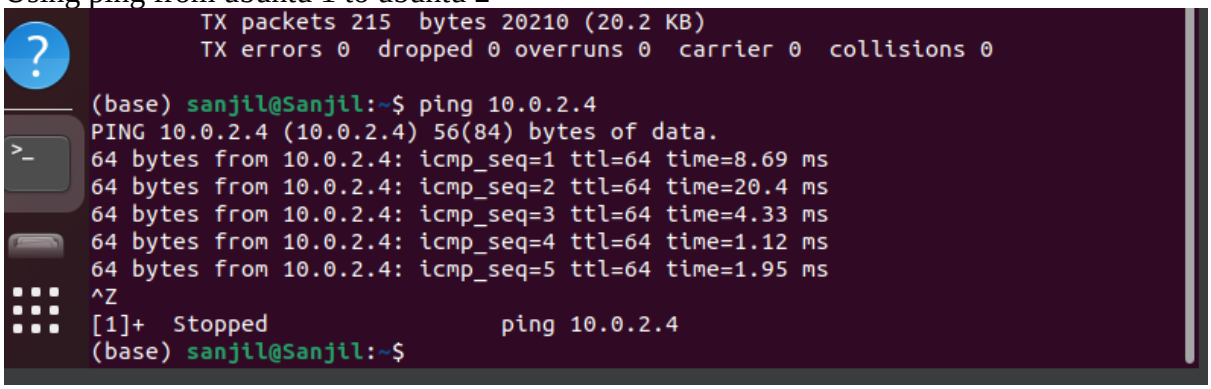
IP address for ubuntu1



IP address for ubuntu2



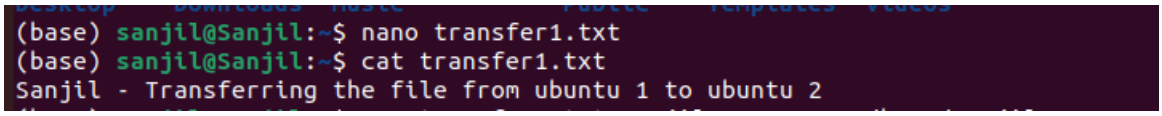
Using ping from ubuntu 1 to ubuntu 2



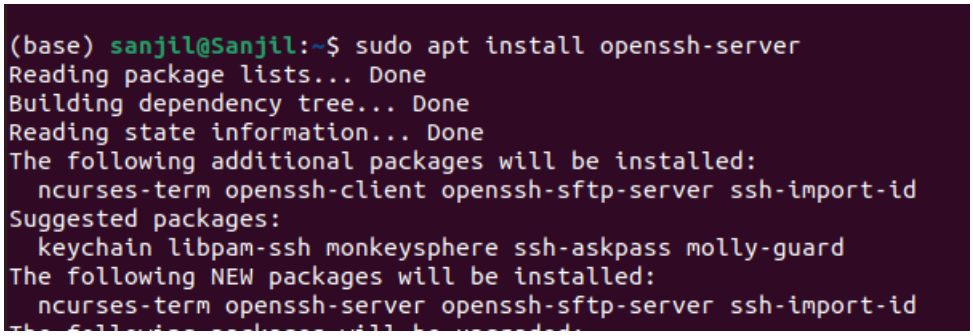
Commands we are going to use

```
ls - list all the files and directories
cat - show the content inside a file
scp - it will help us to copy files from one vm to other
cd - change directory
mkdir - make a new directory
touch - it makes a new file
nano - nano is a editor inside linux os
```

STEP 4 : Creating a file in ubuntu 1 and writing soe content in that file

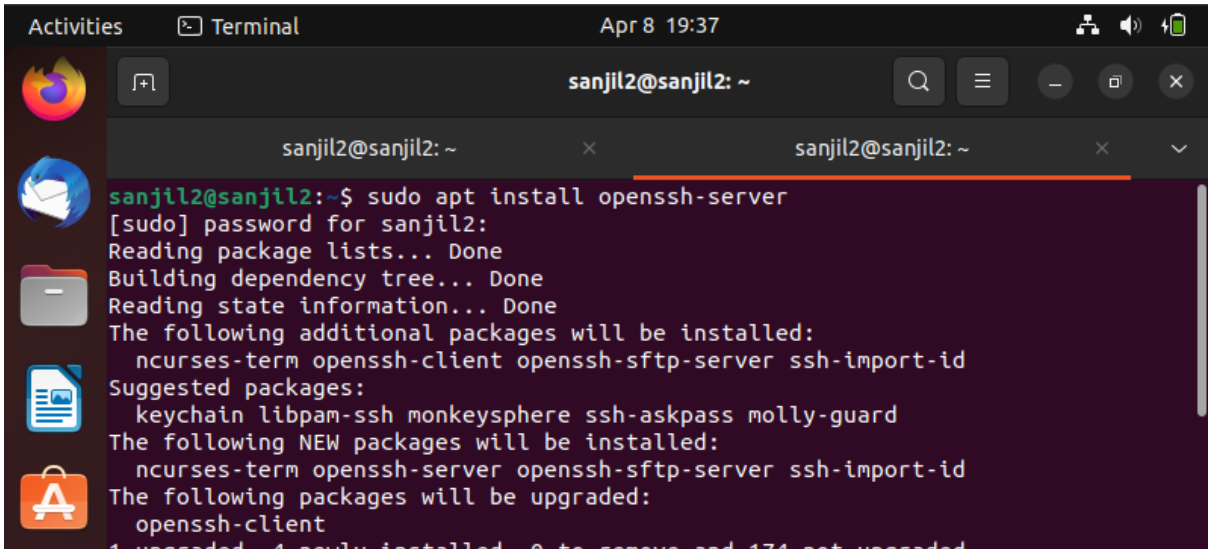


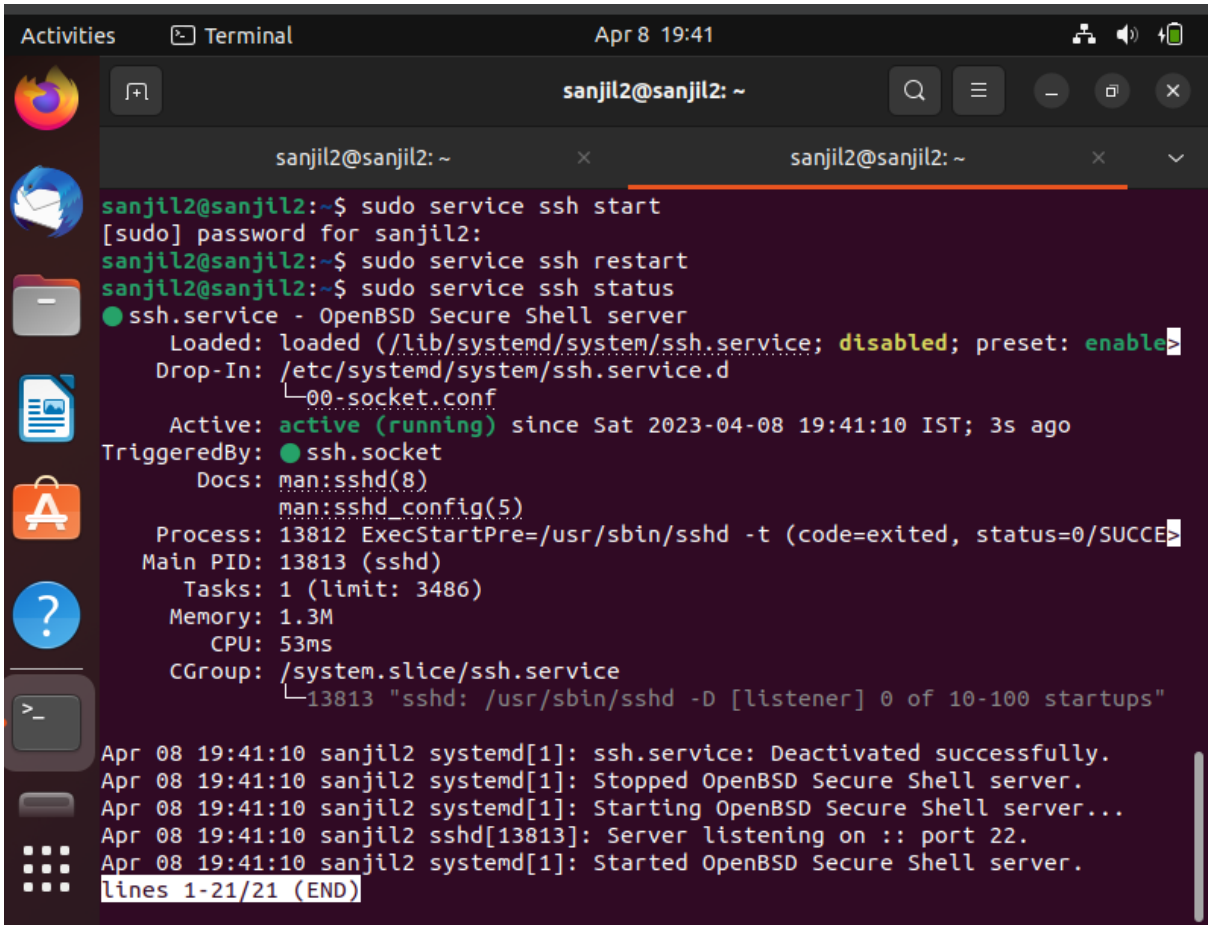
STEP 5 : Activating ssh server in ubuntu 1 and and in ubuntu 2 checking status



```
ERROR: Need 'to' or 'from' clause
(base) sanjil@Sanjil:~$ sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enable>
    Drop-In: /etc/systemd/system/ssh.service.d
             └─00-socket.conf
     Active: active (running) since Sat 2023-04-08 18:12:28 IST; 12min ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 12437 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCE>
   Main PID: 12438 (sshd)
      Tasks: 1 (limit: 4035)
     Memory: 1.4M
        CPU: 104ms
     CGroup: /system.slice/ssh.service
             └─12438 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Activating ssh server in ubuntu 2 and checking status





STEP 6 : Using Scp command to transfer files

```
(base) sanjil@Sanjil:~$ scp transfer1.txt sanjil2@10.0.2.4:/home/sanjil2
The authenticity of host '10.0.2.4 (10.0.2.4)' can't be established.
ED25519 key fingerprint is SHA256:wANhN2LdSkXp204n9wiGtX/9FzGialYYFJqlwpy2UjU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.4' (ED25519) to the list of known hosts.
sanjil2@10.0.2.4's password:
transfer1.txt                                 100%   57     0.6KB/s   00:00
(base) sanjil@Sanjil:~$ scp transfer1.txt Sanjil1@10.0.2.5:/home/Sanjil1
```

Ls command before transferring file

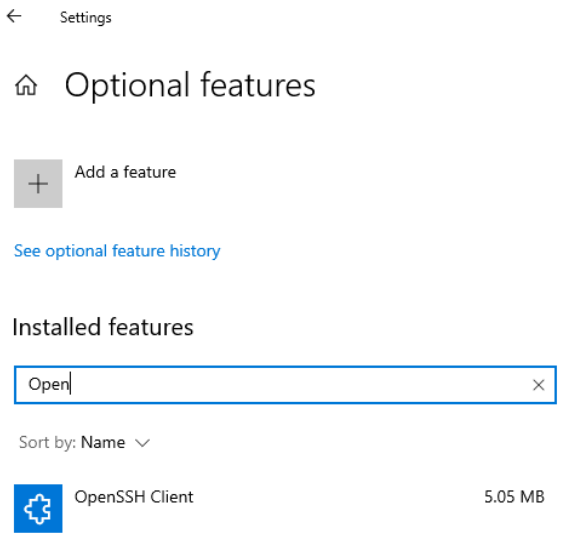
```
sanjil2@sanjil2:~$ ls /home/sanjil2
Desktop    Downloads  Pictures  snap       Videos
Documents  Music      Public    Templates
```

Now checking file transferred successfully in ubuntu2

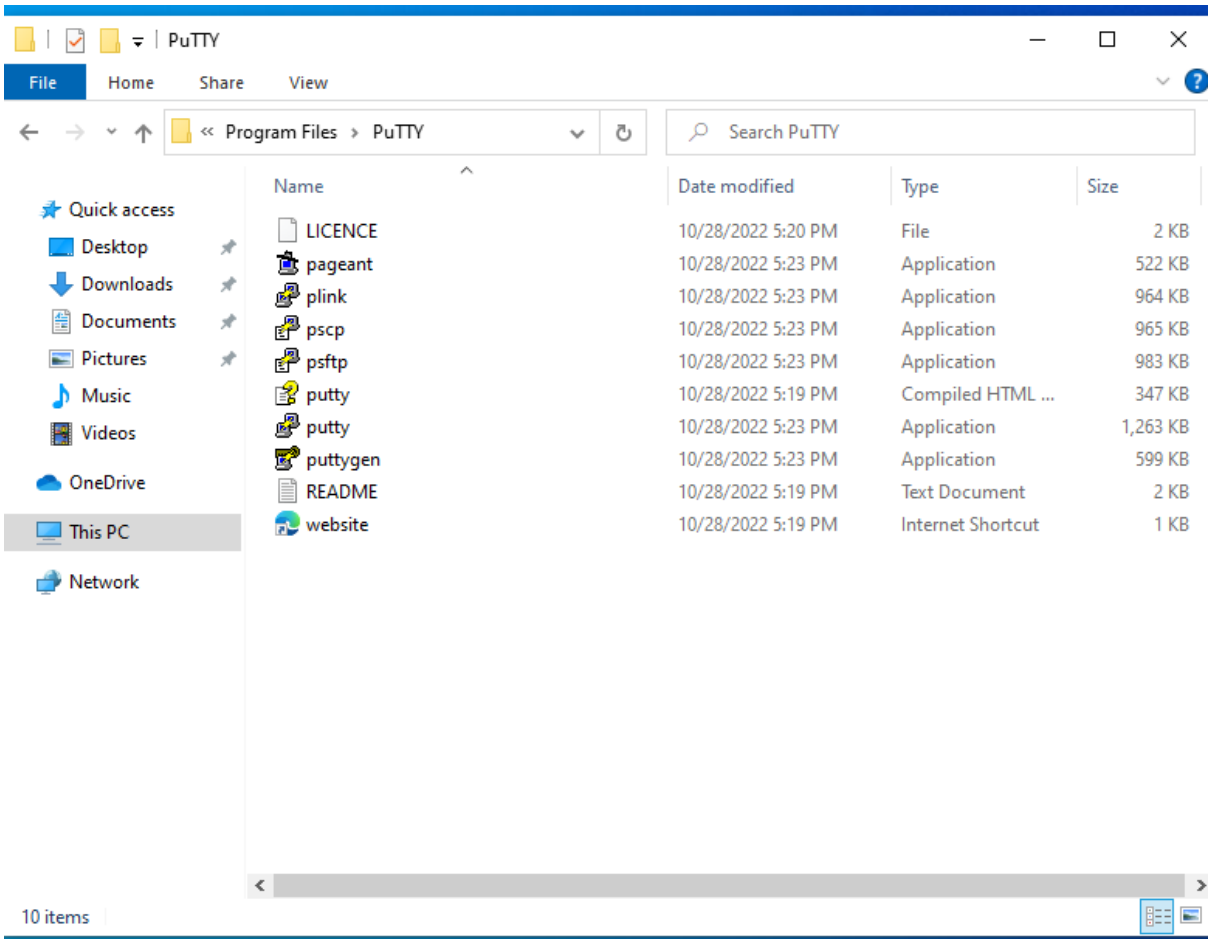
```
sanjil2@sanjil2:~$ ls /home/sanjil2
Desktop    Downloads  Pictures  snap       transfer1.txt
Documents  Music      Public    Templates  Videos
sanjil2@sanjil2:~$ cat transfer1.txt
Sanjil - Transferring the file from ubuntu 1 to ubuntu 2
sanjil2@sanjil2:~$
```

STEP 7 : Now we are going to transfer file from ubuntu1 to windows machine

For that install open shh client



STEP 8 : Next install PuTTY



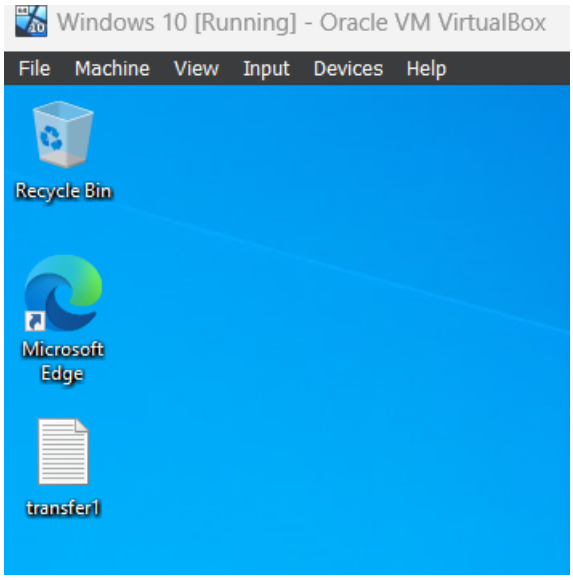
STEP 9 : Next go to command promote and enter the following command

```
C:\Program Files\PuTTY>pscp.exe -P 22 sanjil@10.0.2.15:/home/transfer1.txt C:\Users\Sanjil1\Desktop
The host key is not cached for this server:
  10.0.2.15 (port 22)
You have no guarantee that the server is the computer you
think it is.
The server's ssh-ed25519 key fingerprint is:
  ssh-ed25519 255 SHA256:uMbATocQfTKJQIiNSZWa7MF1zSzn7OKmNi2QcdYOjUs
If you trust this host, enter "y" to add the key to PSCP's
cache and carry on connecting.
If you want to carry on connecting just once, without adding
the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n, Return cancels connection, i for more info) y
sanjil@10.0.2.15's password:
unable to identify /home/transfer1.txt: no such file or directory

C:\Program Files\PuTTY>pscp.exe -P 22 sanjil@10.0.2.15:transfer1.txt C:\Users\Sanjil1\Desktop
sanjil@10.0.2.15's password:
transfer1.txt             | 0 kB |   0.1 kB/s | ETA: 00:00:00 | 100%

C:\Program Files\PuTTY>
```

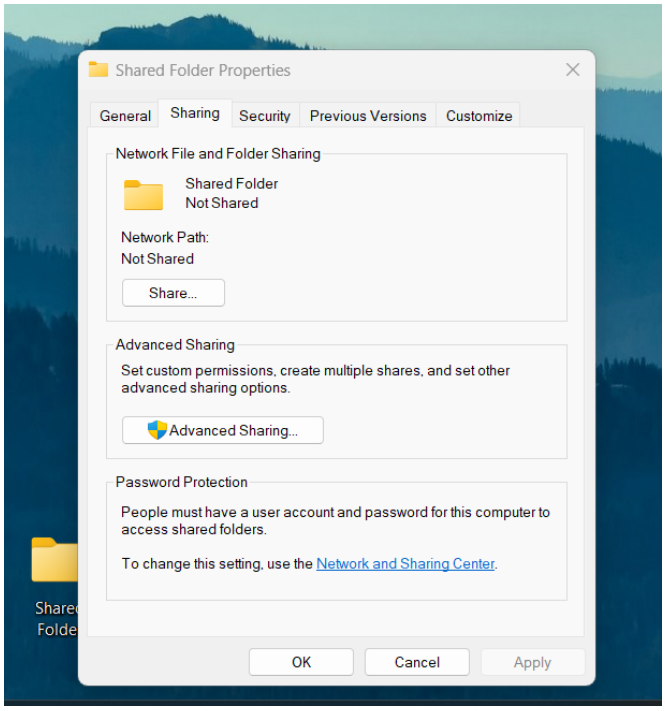
STEP 10 : Now check desktop for the file



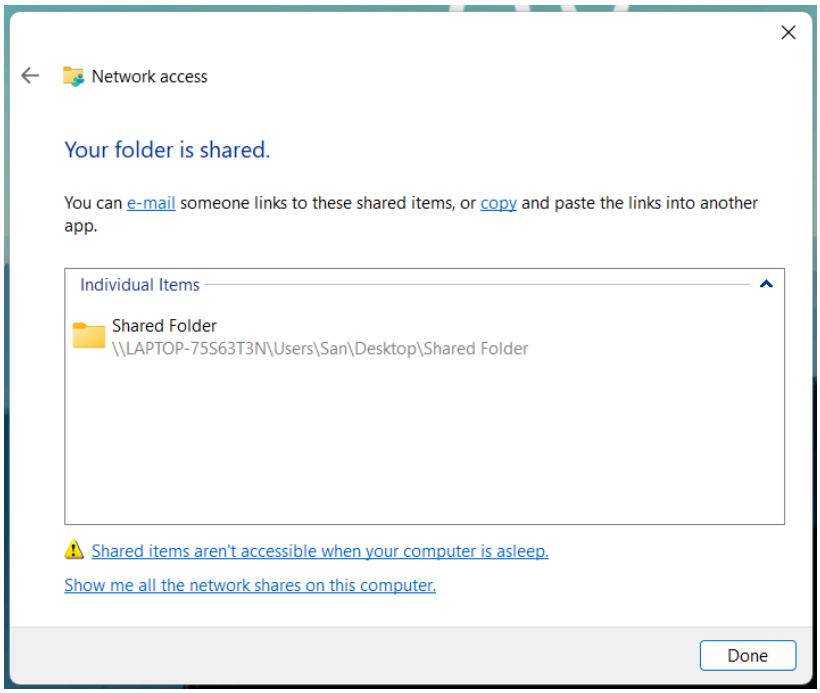
4. Transferring Files between Virtual Machines and Host Machine

a) Sharing Files between vm windows machine and host windows machine

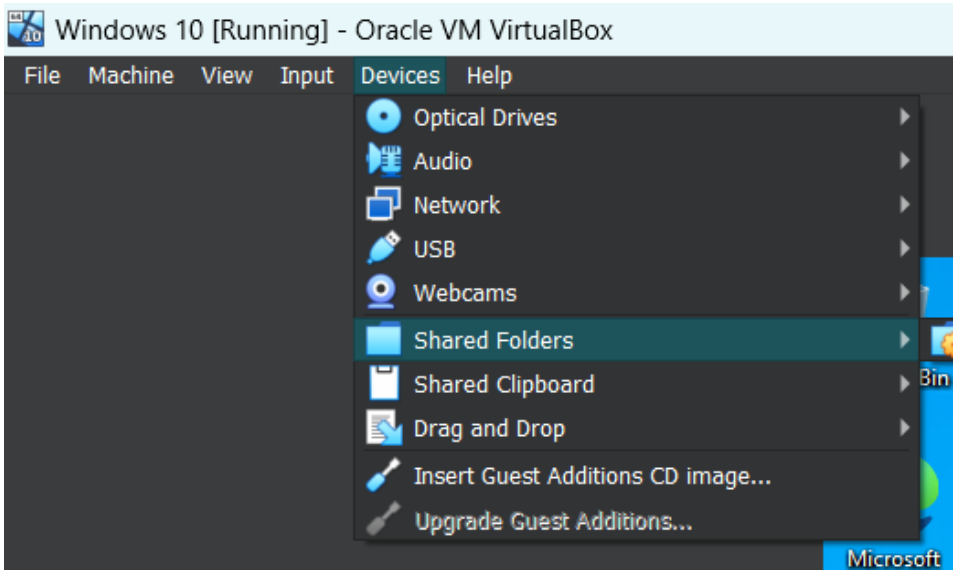
STEP 1 : Create a folder and go to properties and go to sharing



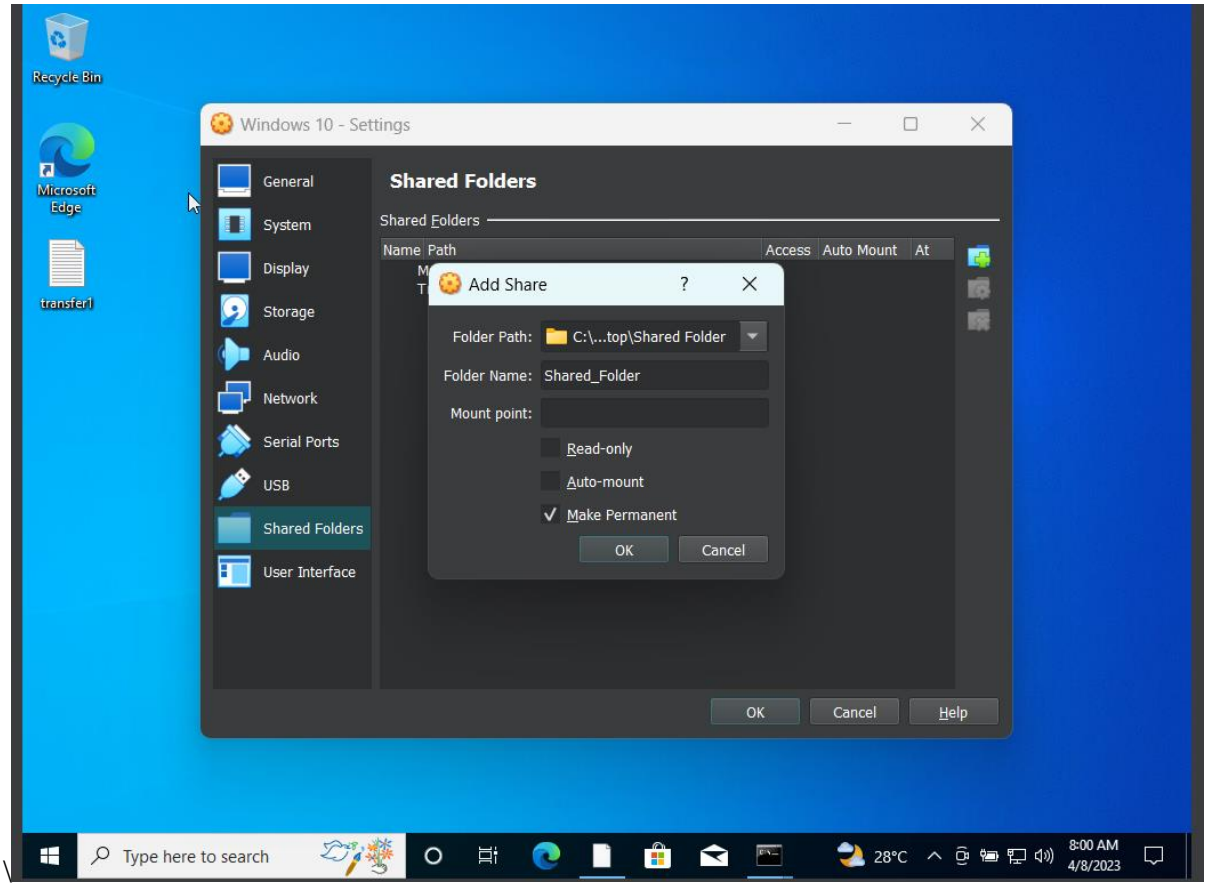
STEP 2 : Give Share permission and read and write permission



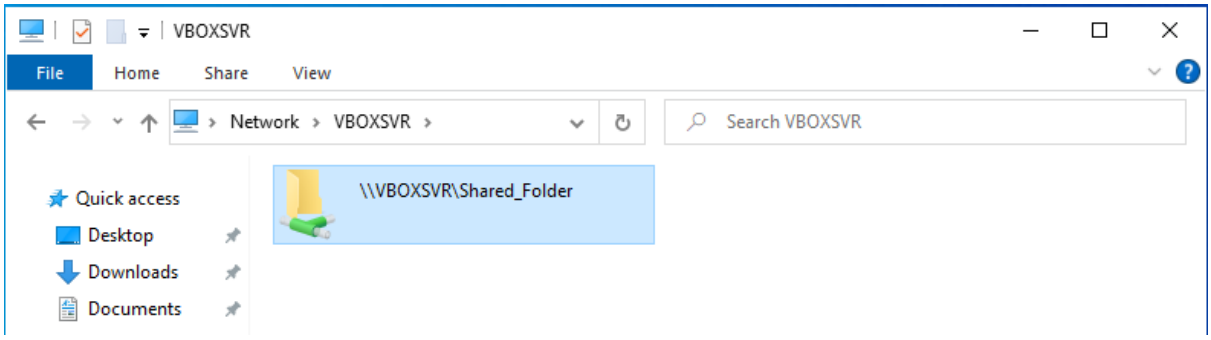
STEP 3 : Go to devices and shared folders

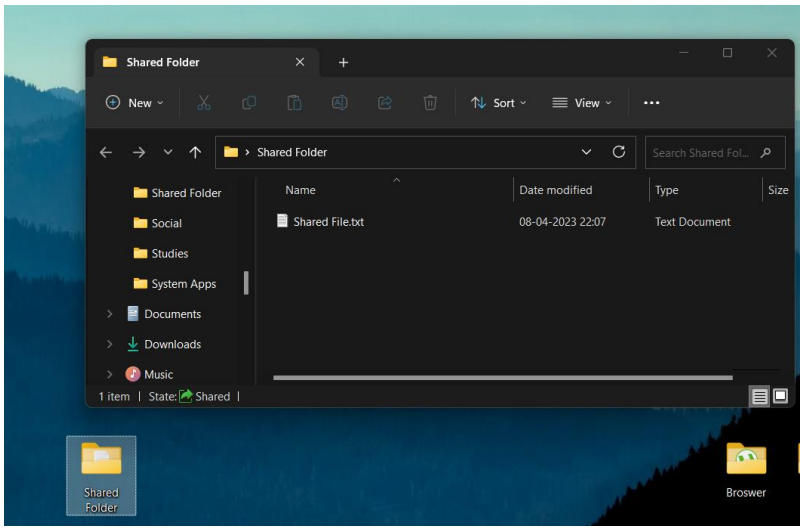


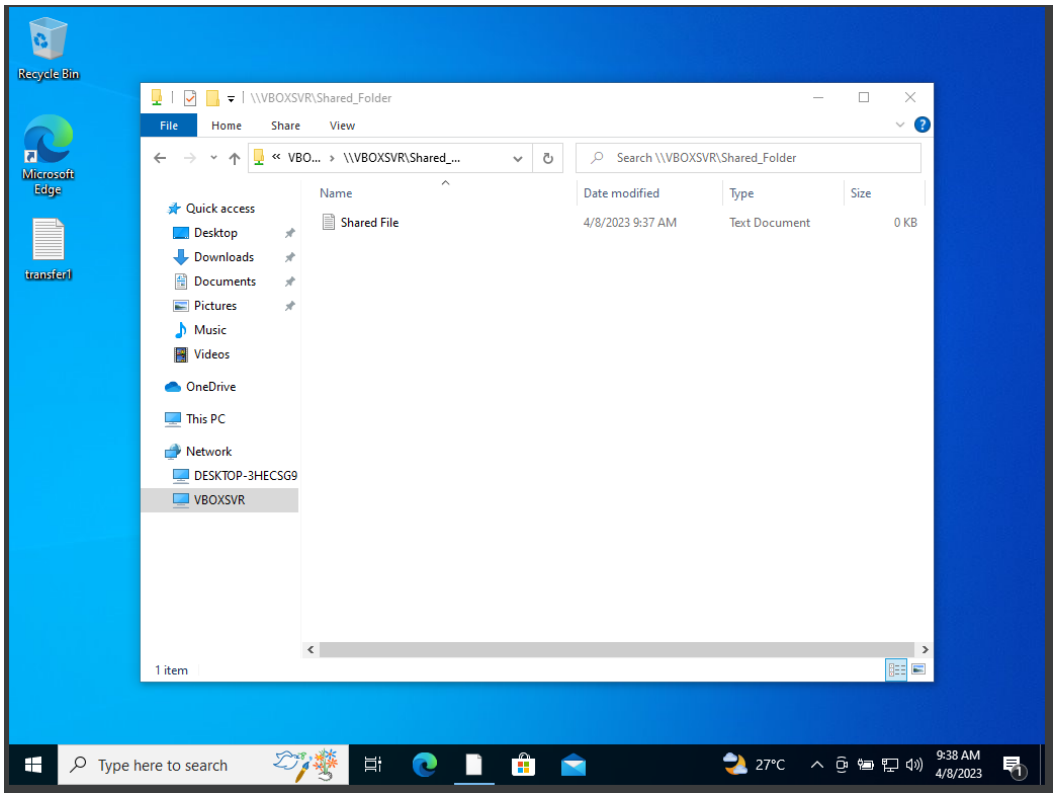
STEP 4 : Select others and select folder that want to be shared



STEP 5 : Under network you can see shared folder

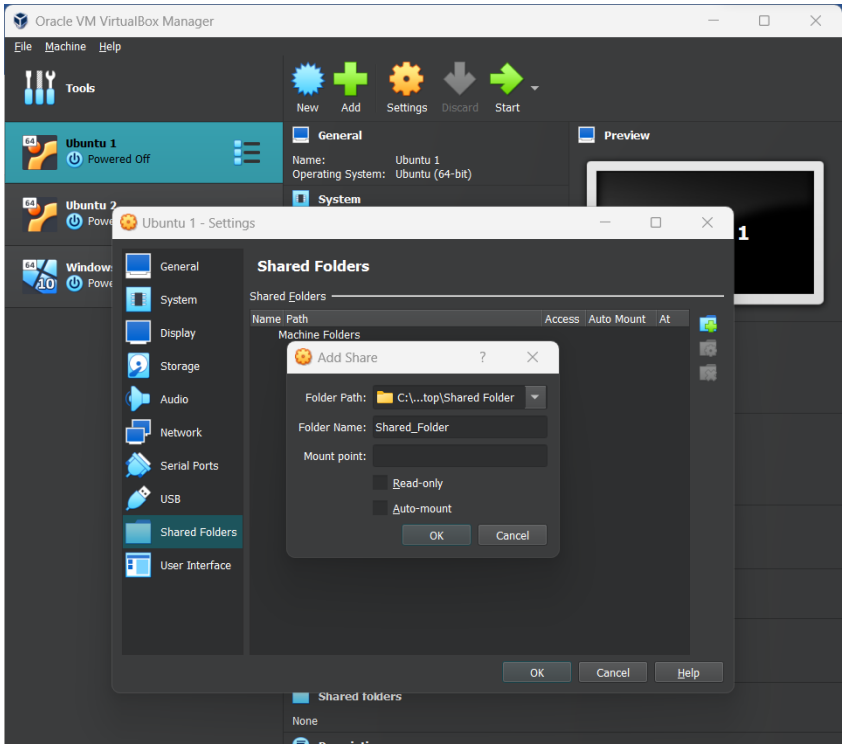




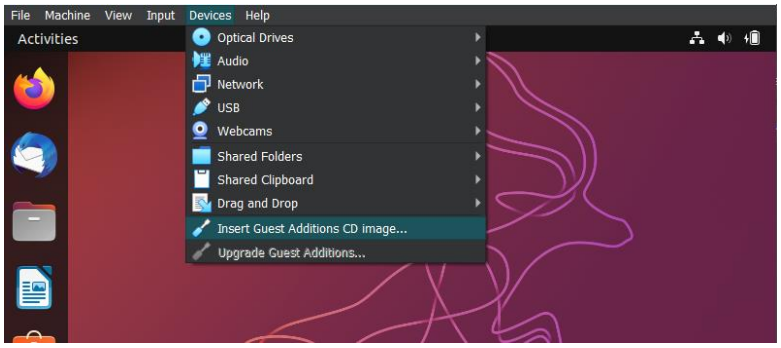


b) Sharing Files between vm Ubuntu machine and host windows machine

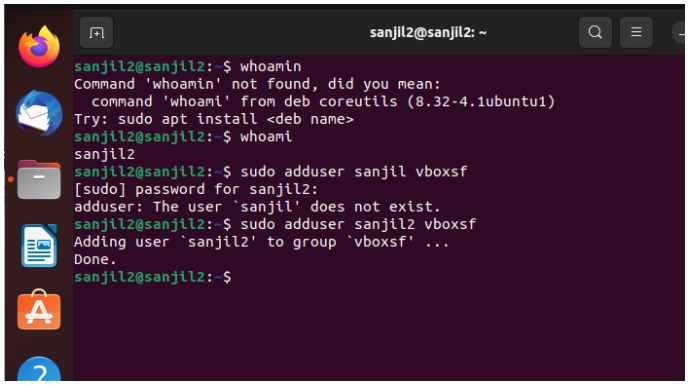
STEP 1 : Select shared folders in settings of ubuntu machine and click which folder that want to shared



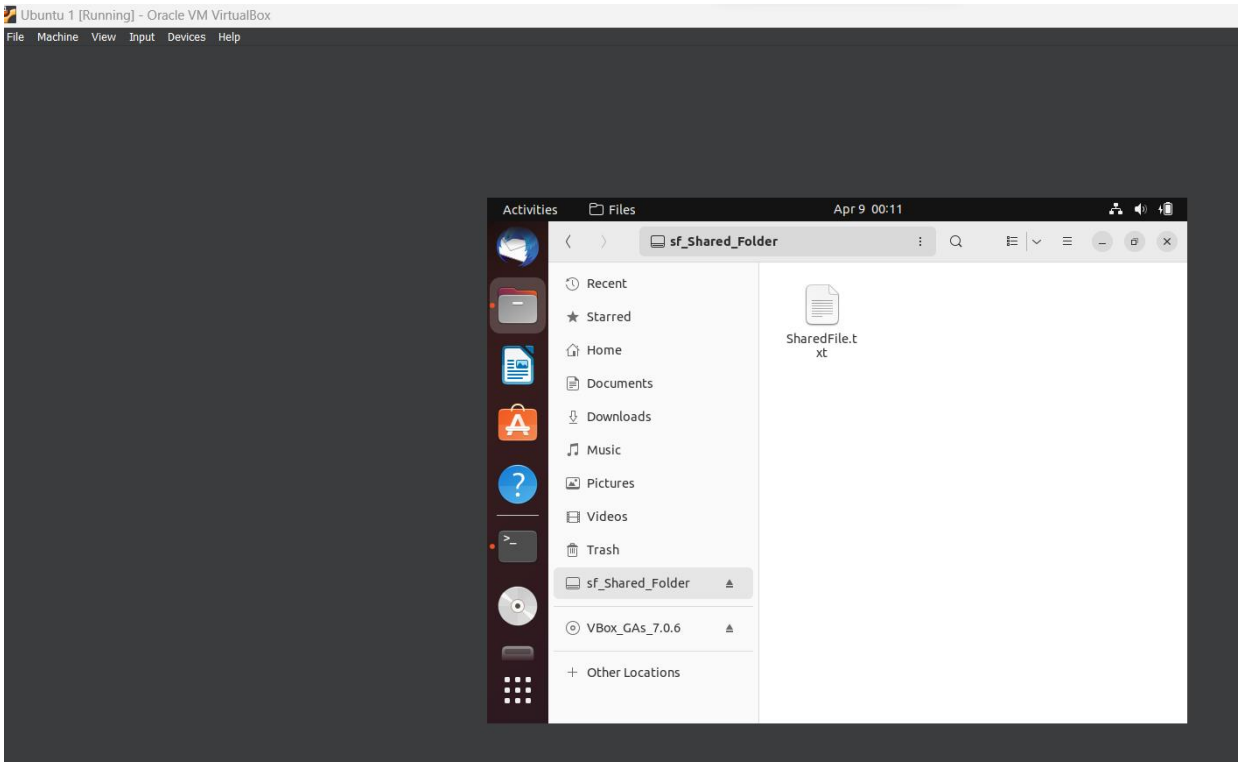
STEP 2 : Install guest addition cd image



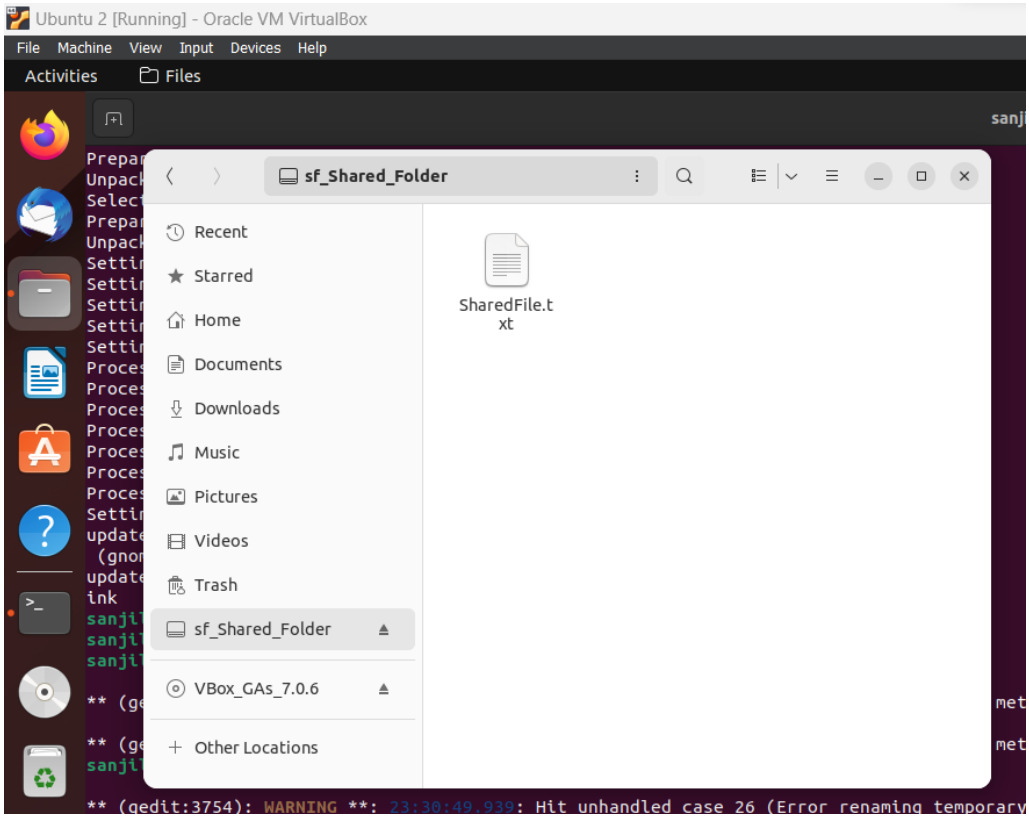
STEP 3 : Add user



STEP 4 : Then you can see share folders in ubuntu 1



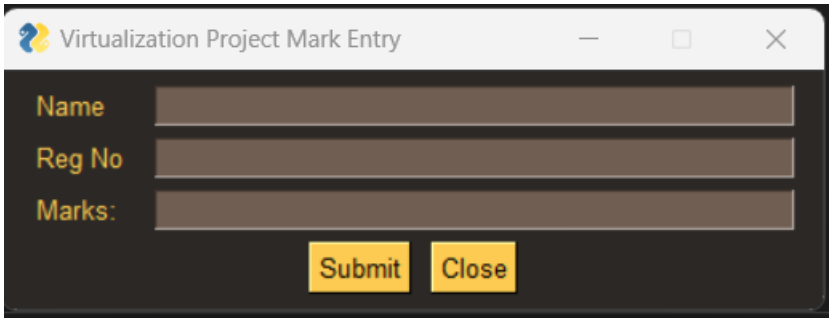
STEP 5 : As well as in ubuntu 2



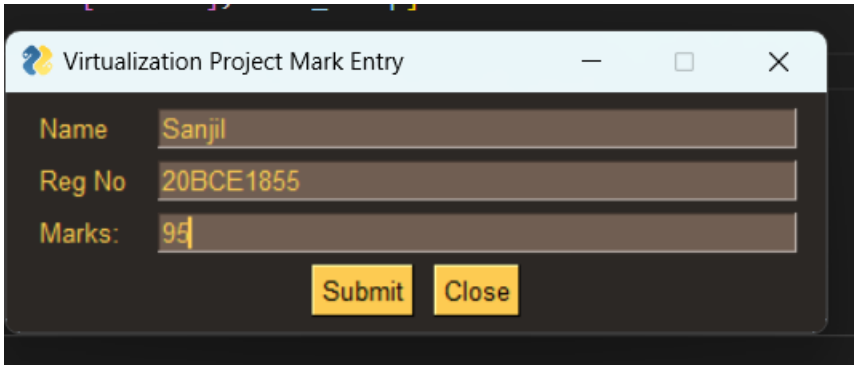
7.OUTPUT

Creating Python Marks Entry app in shared folder

1. APP INTERFACE



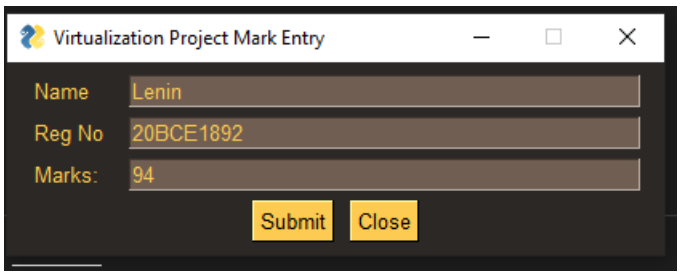
2. Data Entered in Host Machine (Windows)



Entered Data In Excel

	A	B	C	D	E
1	ID	NAME	REG_NO	MARKS	TIME_STAMP
2	2	Sanjil	20BCE1855	95	09/04/2023 16:00:44
3					

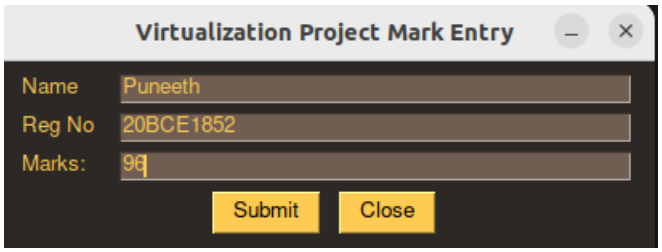
3. Data Entered in Virtual Machine (Windows)



Entered Data In Excel

	A	B	C	D	E
1	ID	NAME	REG_NO	MARKS	TIME_STAMP
2	2	Sanjil	20BCE1855	95	09/04/2023 16:00:44
3	3	Lenin	20BCE1892	94	09/04/2023 03:35:40

4. Data Entered in Virtual Machine (Ubuntu)



Entered Data In Excel

	A	B	C	D	E	F
1	ID	NAME	REG_NO	MARKS	TIME_STAMP	
2	2	Sanjil	20BCE1855	95	09/04/2023 16:00:44	
3	3	Lenin	20BCE1892	94	09/04/2023 03:35:40	
4	4	Puneeth	20BCE1852	96	09/04/2023 16:10:44	
5						

8. CONCLUSION

In conclusion, creating virtual environments for Python in Anaconda on both Ubuntu and Windows Virtual Machines using Oracle VM Box is a simple process that allows for easier management of Python packages and dependencies. Additionally, transferring files between the two VMs using the NAT network configuration can be done easily using the scp command. This can be especially useful for sharing code and data between team members or between different development environments. Overall, using virtual environments and virtual machines can help streamline the development process and improve the reproducibility of research and software projects.

9. REFERENCE

1.Creating Virtual Machines :

https://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/ocis/creating_vm/bmc_vm_tutorial.html

https://www.techbeatly.com/how-to-create-and-use-natnetwork-in-virtualbox/

https://morioh.com/p/ec682c1303ec

2.Creating an Anaconda Environment documentation:

https://docs.anaconda.com/anaconda/user-guide/tasks/creating-environments/

3.Managing environments in the Anaconda documentation:

https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

4.Oracle VM VirtualBox documentation: https://www.virtualbox.org/manual/

5."Using NAT networking" in the VirtualBox documentation:

https://www.virtualbox.org/manual/ch06.html#network_nat

6.SCP command tutorial: https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/

7.How to use SCP command to securely transfer files article: https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/

10. Annexure

```
from openpyxl import load_workbook
import PySimpleGUI as sg
from datetime import datetime


sg.theme('DarkAmber')

layout = [[sg.Text('Name'),sg.Push(), sg.Input(key='NAME')],
          [sg.Text('Reg No'),sg.Push(), sg.Input(key='REG_NO')],
          [sg.Text('Marks:'),sg.Push(), sg.Input(key='MARKS')],
          [sg.Button('Submit'), sg.Button('Close')]]

window = sg.Window('Virtualization Project Mark Entry', layout,
element_justification='center')

while True:
    event, values = window.read()
    if event == sg.WIN_CLOSED or event == 'Close':
        break
    if event == 'Submit':
        try:
            wb = load_workbook('Book.xlsx')
            sheet = wb['Sheet1']
            ID = len(sheet['ID']) + 1
            time_stamp = datetime.now().strftime("%d/%m/%Y %H:%M:%S")

            data = [ID, values['NAME'], values['REG_NO'], values['MARKS'],
time_stamp]

            sheet.append(data)

            wb.save('Book.xlsx')

            window['NAME'].update(value='')
            window['REG_NO'].update(value='')
            window['MARKS'].update(value='')
            window['NAME'].set_focus()

            sg.popup('Success', 'Data Saved')
        except PermissionError:
            sg.popup('File in use', 'File is being used by another
User.\nPlease try again later.')



window.close()
```