Lab 1: Ubuntu on VirtualBox

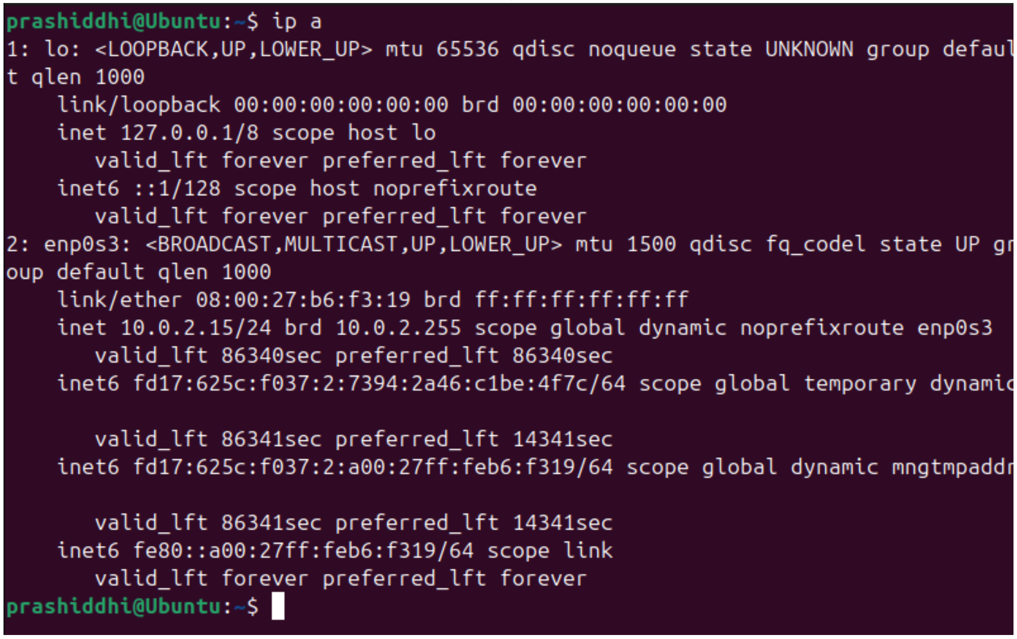
Virtualization is the creation of a software-based version of a physical resource, like an operating system or server. VirtualBox is a popular type-2 hypervisor, which allows a Guest OS (Ubuntu) to run on top of a Host OS. The VM is contained within a file, offering isolation and portability.

Task 1: Different Network Modes

VirtualBox manages the VM's connectivity using specific network modes, which define how the VM's Network Interface Card (NIC) interacts with the physical network:

i) NAT (Network Address Translation)

- The VM uses the host system’s internet connection through network address translation.
- VM gets a private internal IP, not visible on the physical network.
- Internet works without extra configuration.
- Other machines on LAN cannot directly access the VM.

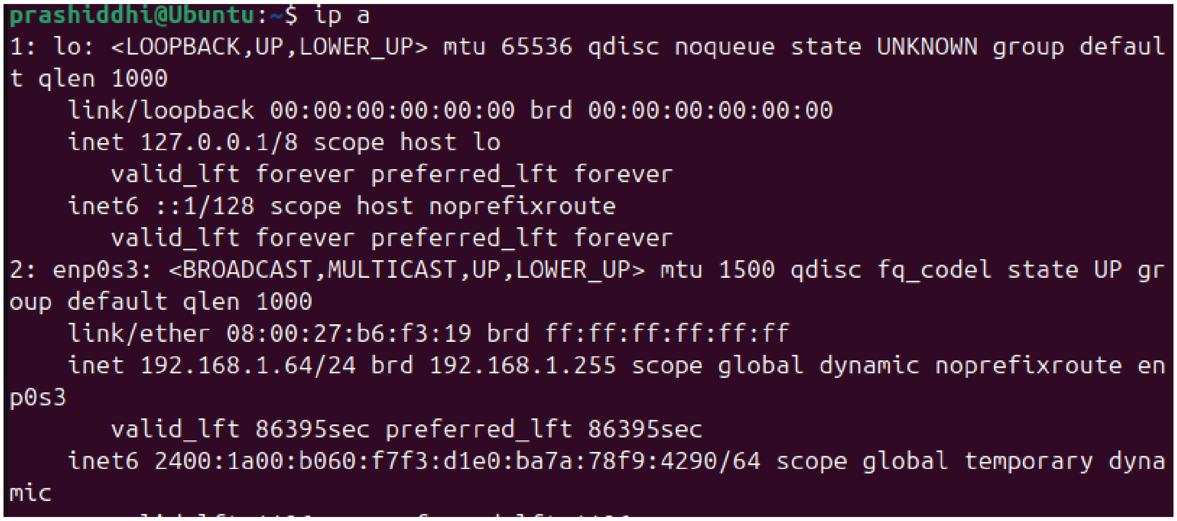


```
prashiddhi@Ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:b6:f3:19 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86340sec preferred_lft 86340sec
    inet6 fd17:625c:f037:2:7394:2a46:c1be:4f7c/64 scope global temporary dynamic

       valid_lft 86341sec preferred_lft 14341sec
    inet6 fd17:625c:f037:2:a00:27ff:feb6:f319/64 scope global dynamic mngtmpaddr

       valid_lft 86341sec preferred_lft 14341sec
    inet6 fe80::a00:27ff:feb6:f319/64 scope link
       valid_lft forever preferred_lft forever
prashiddhi@Ubuntu:~$
```

ii) Bridge mode

- VM becomes a device directly on the physical network.

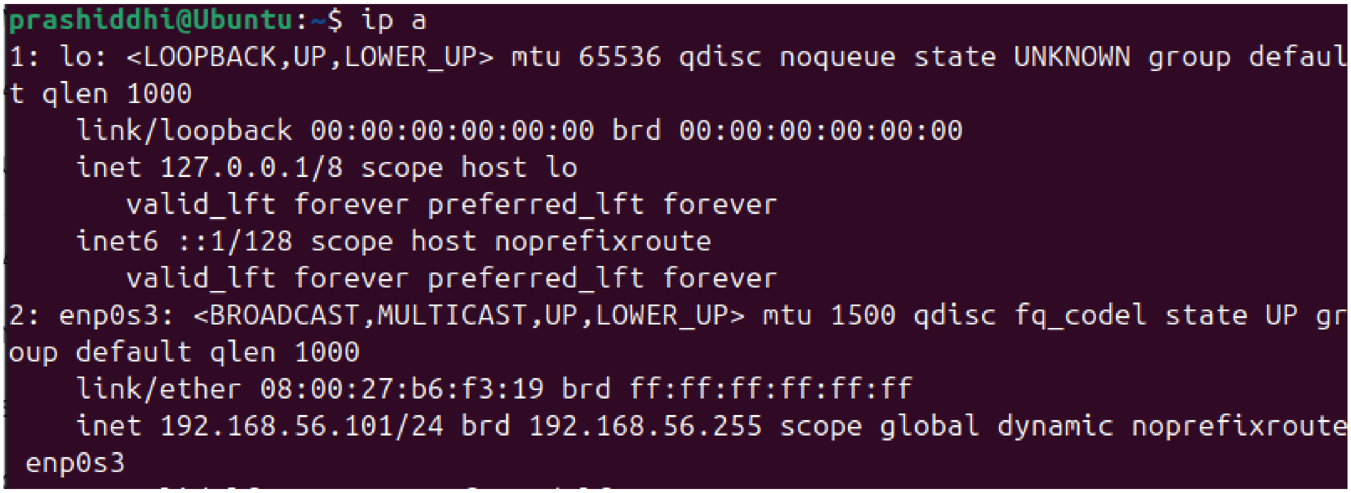
- VM gets IP from the router similar to host machine.
- LAN devices can ping or connect to the VM.
- Suitable for servers or accessing VM from other computers.



```
prashiddhi@Ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:b6:f3:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.64/24 brd 192.168.1.255 scope global dynamic noprefixroute en
p0s3
       valid_lft 86395sec preferred_lft 86395sec
    inet6 2400:1a00:b060:f7f3:d1e0:ba7a:78f9:4290/64 scope global temporary dyna
mic
```

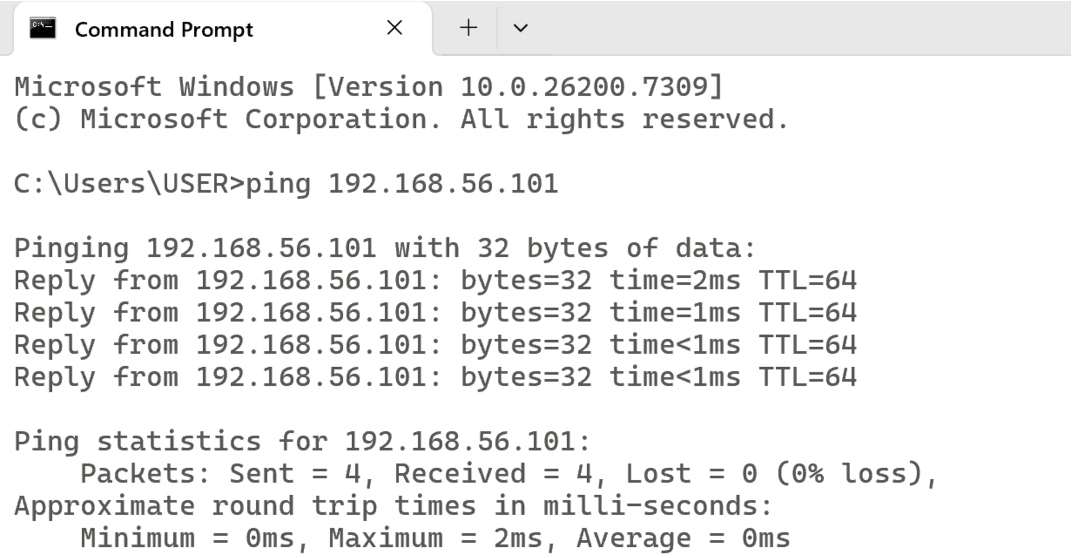
iii) Host Mode

- Creates a private network only between host and VM.
- VM cannot access the internet by default.
- Useful for test environments or secure communication.
- Other physical network devices cannot reach the VM.



```
prashiddhi@Ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:b6:f3:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute
 enp0s3
```

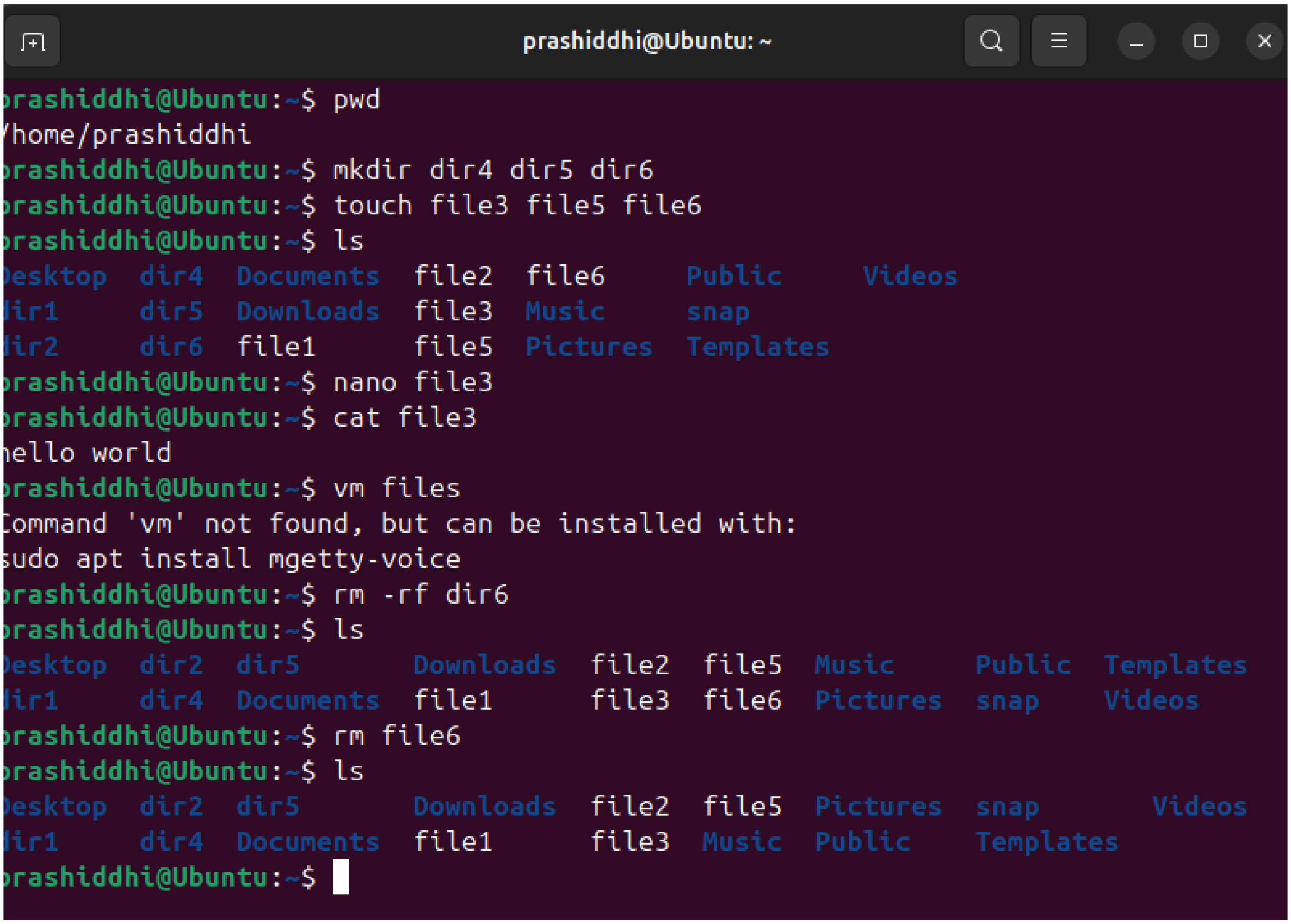
Pinging host from different IP address:



```
Microsoft Windows [Version 10.0.26200.7309]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>ping 192.168.56.101

Pinging 192.168.56.101 with 32 bytes of data:
Reply from 192.168.56.101: bytes=32 time=2ms TTL=64
Reply from 192.168.56.101: bytes=32 time=1ms TTL=64
Reply from 192.168.56.101: bytes=32 time<1ms TTL=64
Reply from 192.168.56.101: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

Task 2: Basic commands

Command	**Description**
cd	Change directory
pwd	Show current working directory
mkdir <directory>	Create a new directory
touch <file>	Create an empty file
ls	List files and directories
nano <file>	Open file in nano editor (exit: Ctrl X)
cat <file>	Display file contents
rm <file>	Delete a file
ls -ltr	List with details sorted by time
rm <directory>	Fails if directory not empty
rm -rf <directory>	Force delete directory
df -h	Show disk/storage usage
free -mh	Show RAM usage
top	Show running processes



```
prashiddhi@Ubuntu:~$ pwd
/home/prashiddhi
prashiddhi@Ubuntu:~$ mkdir dir4 dir5 dir6
prashiddhi@Ubuntu:~$ touch file3 file5 file6
prashiddhi@Ubuntu:~$ ls
Desktop  dir4  Documents  file2  file6     Public     Videos
dir1     dir5  Downloads  file3  Music     snap
dir2     dir6  file1      file5  Pictures  Templates
prashiddhi@Ubuntu:~$ nano file3
prashiddhi@Ubuntu:~$ cat file3
hello world
prashiddhi@Ubuntu:~$ vm files
Command 'vm' not found, but can be installed with:
sudo apt install mgetty-voice
prashiddhi@Ubuntu:~$ rm -rf dir6
prashiddhi@Ubuntu:~$ ls
Desktop  dir2  dir5       Downloads  file2  file5  Music     Public  Templates
dir1     dir4  Documents  file1      file3  file6  Pictures  snap    Videos
prashiddhi@Ubuntu:~$ rm file6
prashiddhi@Ubuntu:~$ ls
Desktop  dir2  dir5       Downloads  file2  file5  Pictures  snap       Videos
dir1     dir4  Documents  file1      file3  Music  Public    Templates
prashiddhi@Ubuntu:~$ 
```

Task 3: Install ngnix

Nginx is a powerful, high-performance web server, and reverse proxy. Unlike traditional web servers, it uses an asynchronous, event-driven architecture, making it highly scalable and memory-efficient.

Commands used:

# Install nginx web server

sudo apt install nginx -y

# Check nginx running processes

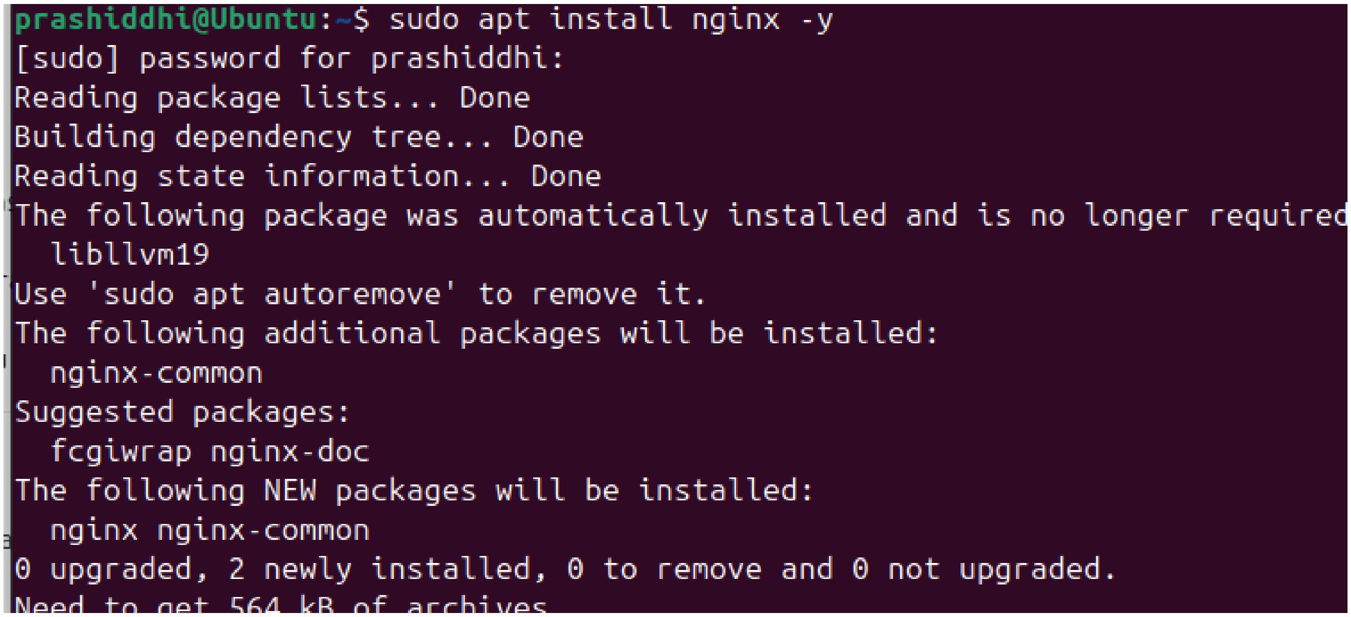
ps -ef | grep nginx

# Install net-tools package (includes netstat, ifconfig etc.)

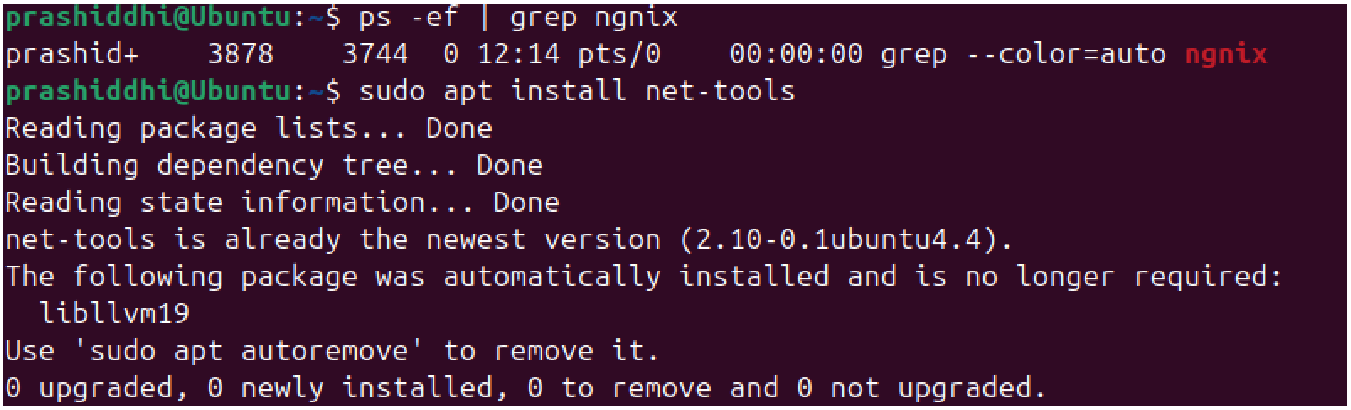
sudo apt install net-tools

# Check which process is listening on port 80 (usually nginx)

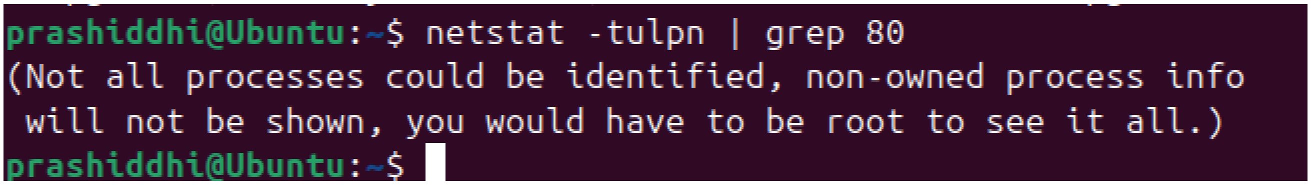
netstat -tupln | grep 80



```
prashiddhi@Ubuntu:~$ sudo apt install nginx -y
[sudo] password for prashiddhi:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 564 kB of archives.
```



```
prashiddhi@Ubuntu:~$ ps -ef | grep ngnix
prashid+    3878    3744  0 12:14 pts/0    00:00:00 grep --color=auto ngnix
prashiddhi@Ubuntu:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (2.10-0.1ubuntu4.4).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```



```
prashiddhi@Ubuntu:~$ netstat -tulpn | grep 80
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
prashiddhi@Ubuntu:~$ 
```

Conclusion

Hence, this lab exercise successfully set up a functional Ubuntu Virtual Machine and allowed exploration of how different VirtualBox network modes behave in practice. Through hands-on testing, the differences between NAT, Bridged, and Host-Only modes became clear—NAT offering internet access with isolation, Bridged enabling full visibility on the external network, and Host-Only restricting communication to the host system. The activity also included installing the Nginx web server and confirming that it was running correctly on Port 80, demonstrating that the VM could serve web content according to the selected network configuration.