西北工業大學

2021

Parallel Programming
Assignment #1
Linux Cluster

Team Members

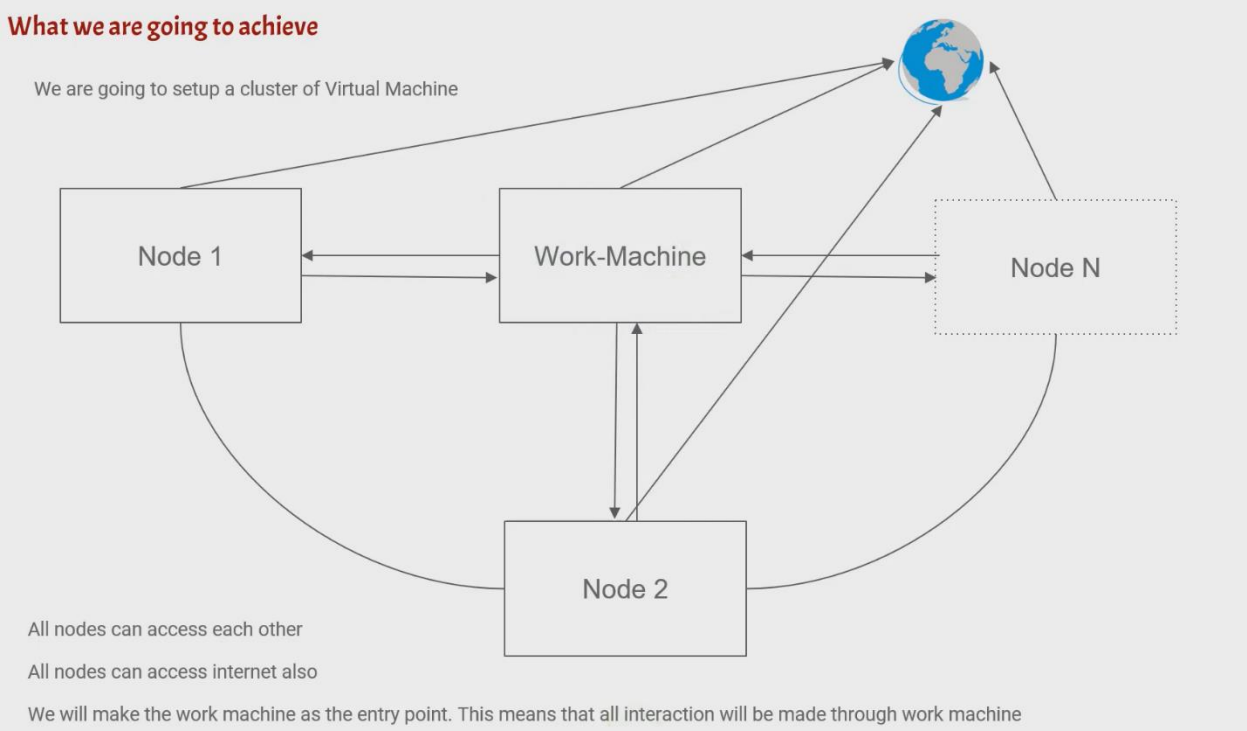
Name: Khan Md Shahedul Islam
Student ID: 2018380130

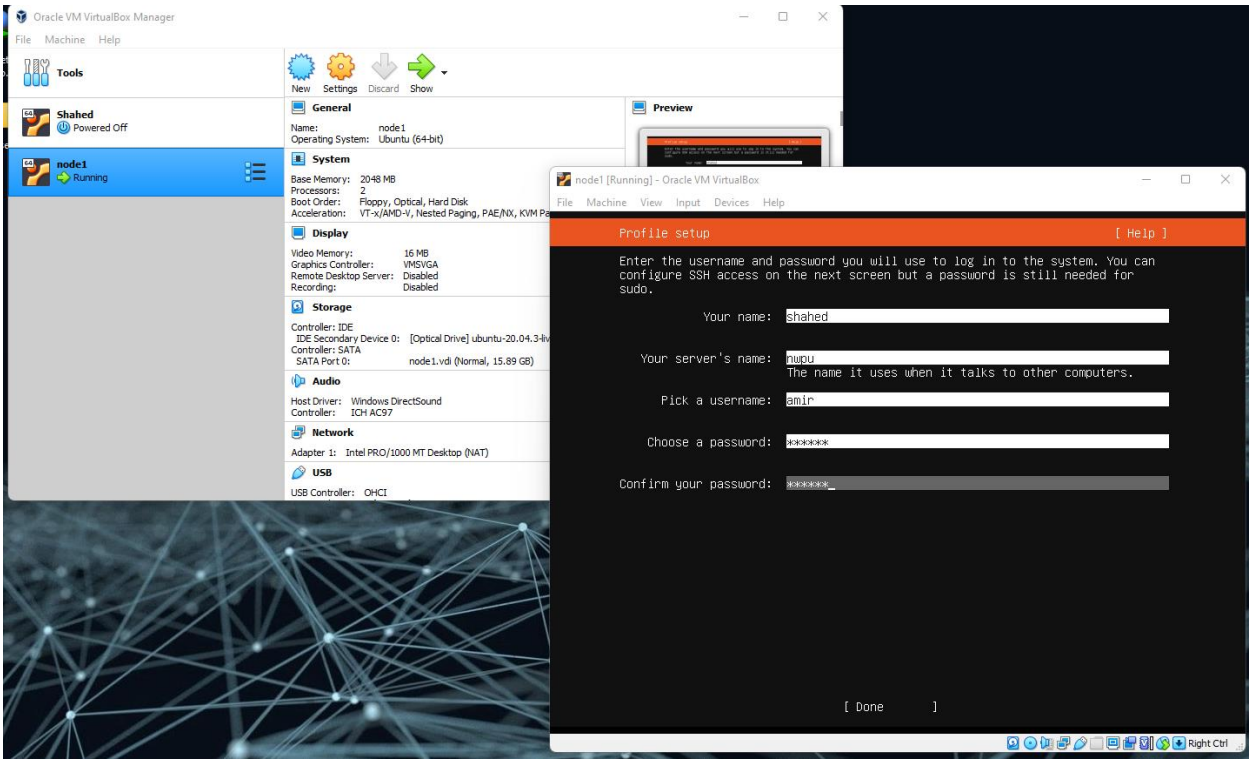
Name: Amirbek Raimov
Student ID: 2018380038

Building linux Cluster

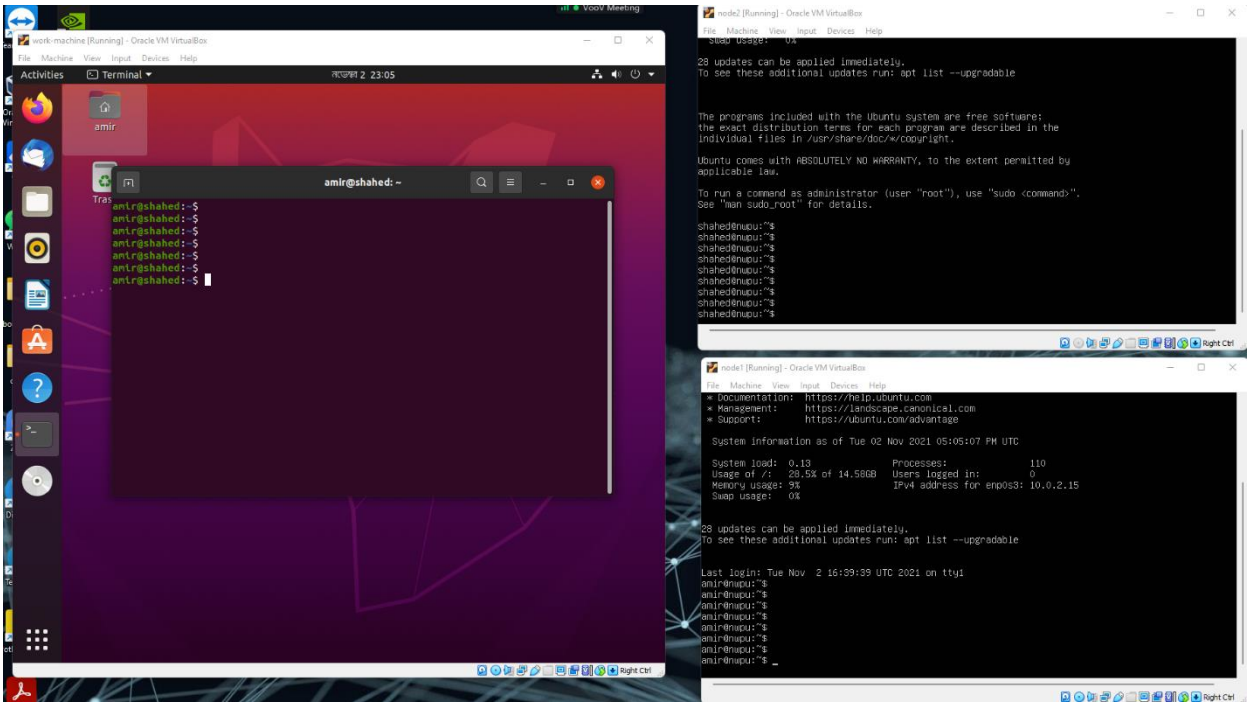
#Structure:

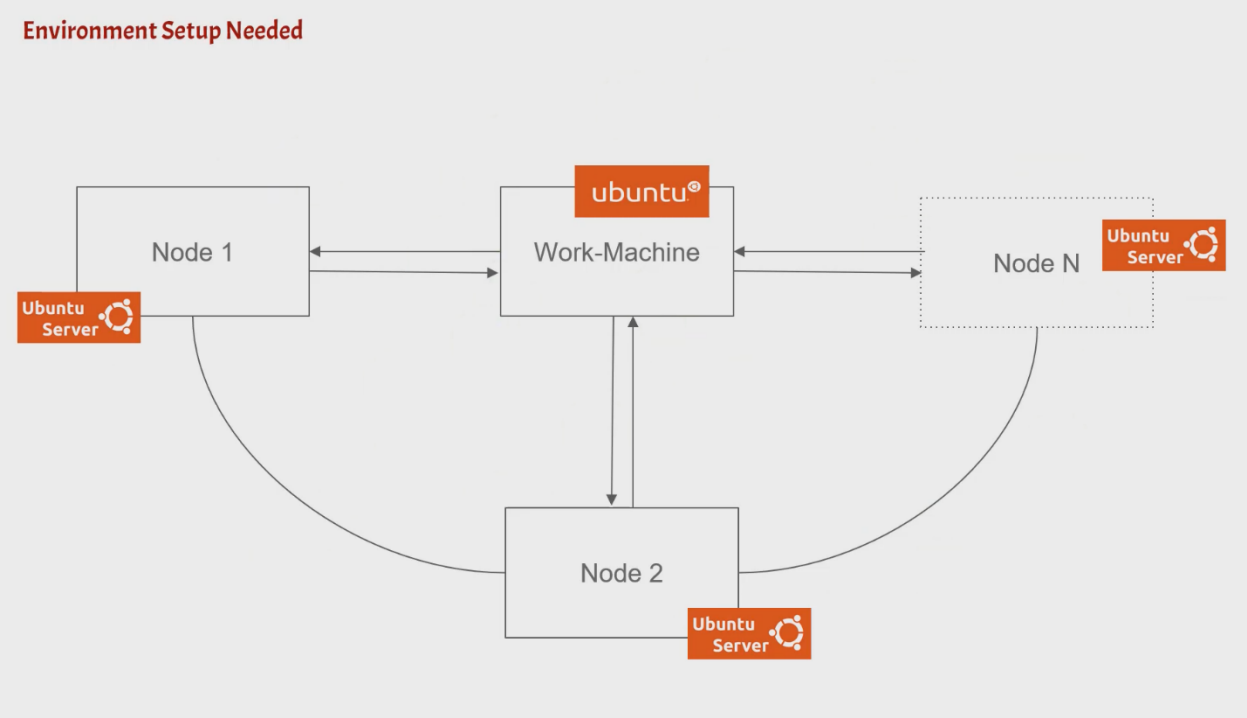
This is how our cluster will run. The workmachine is our main machine(client) and can talk to both nodes while the nodes can talk to the workmachine as well. Alongside the nodes will be able to contact each other and every machine can individually connect to internet.

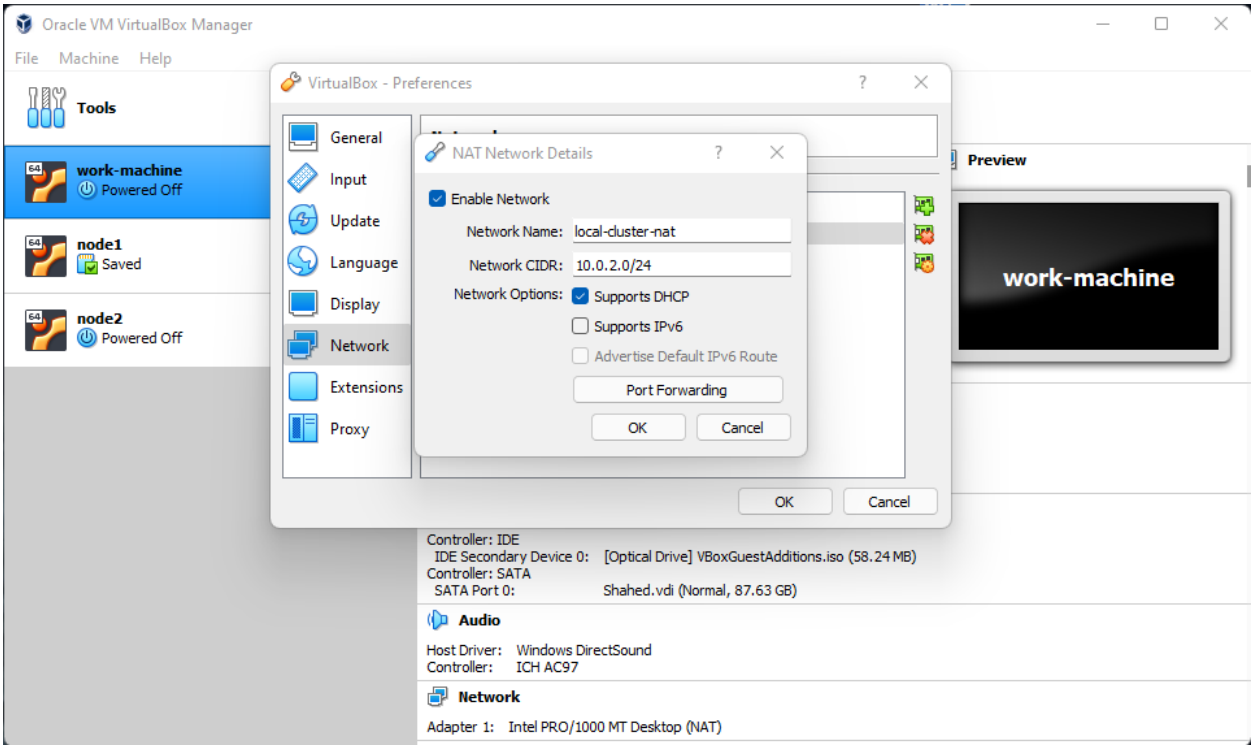




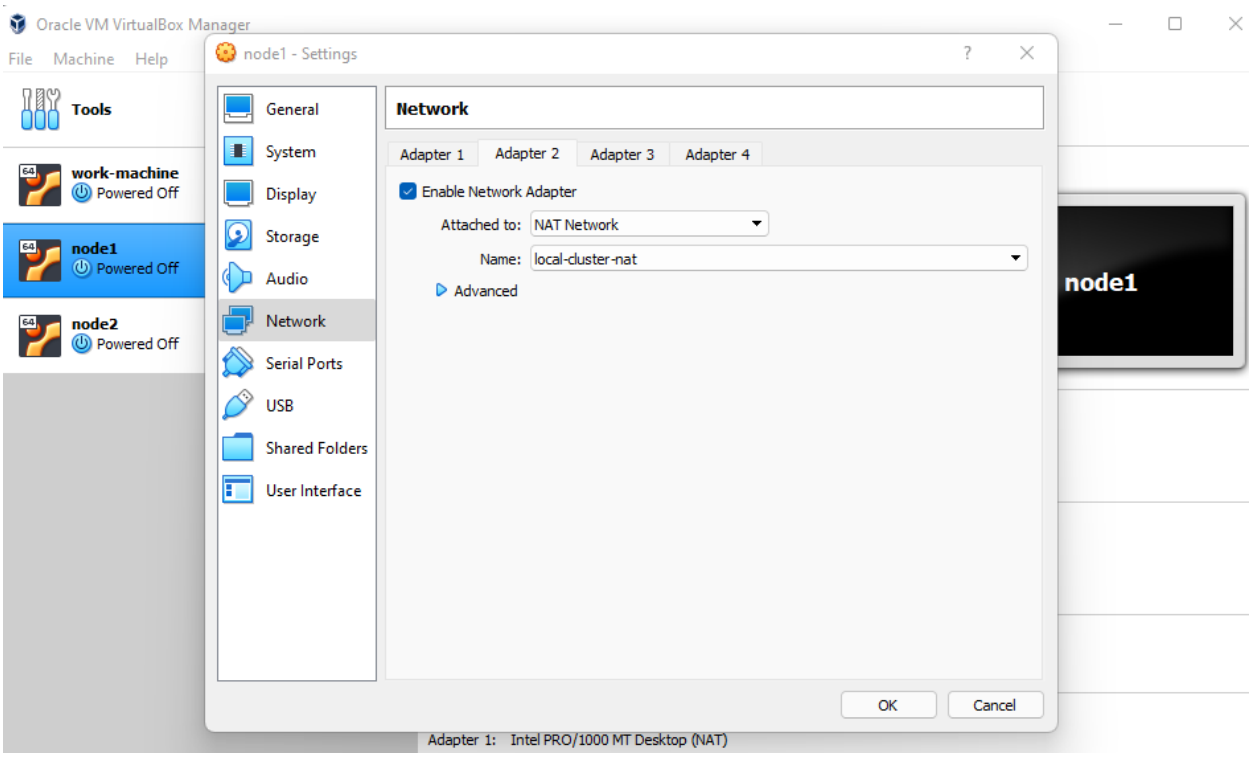
Here we are creating our server machine.

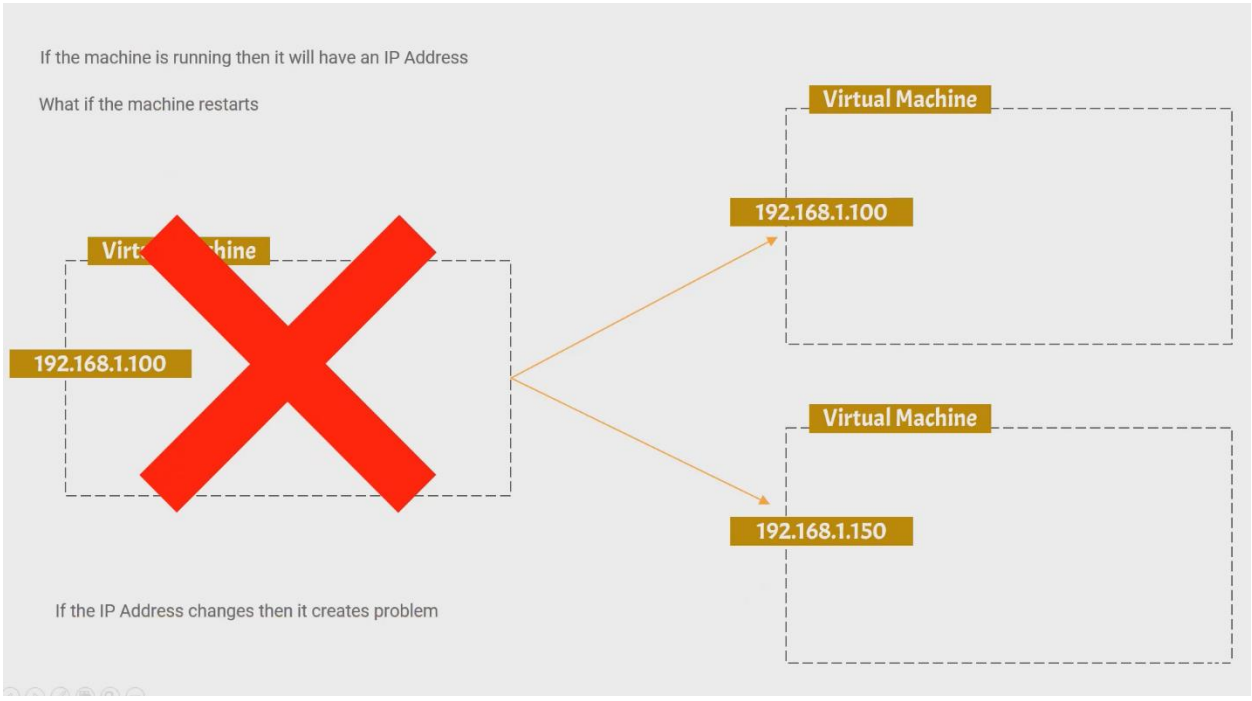






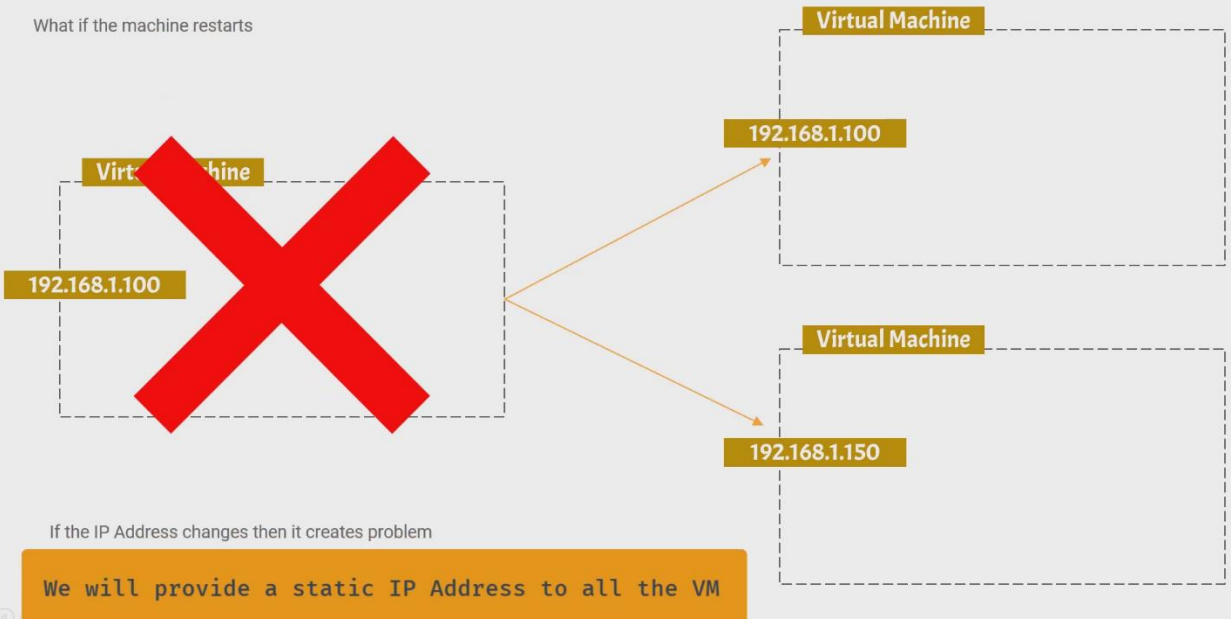
#Creating Nat-network


Oracle VM VirtualBox Manager
File Machine Help
Tools
work-machine
Powered Off
node1
Powered Off
node2
Powered Off
node1 - 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: local-cluster-nat
Advanced
OK
Cancel
node1
Adapter 1: Intel PRO/1000 MT Desktop (NAT)


If the machine is running then it will have an IP Address
What if the machine restarts
Virtual Machine
192.168.1.100
192.168.1.100
Virtual Machine
192.168.1.150
If the IP Address changes then it creates problem

If the machine is running then it will have an IP Address

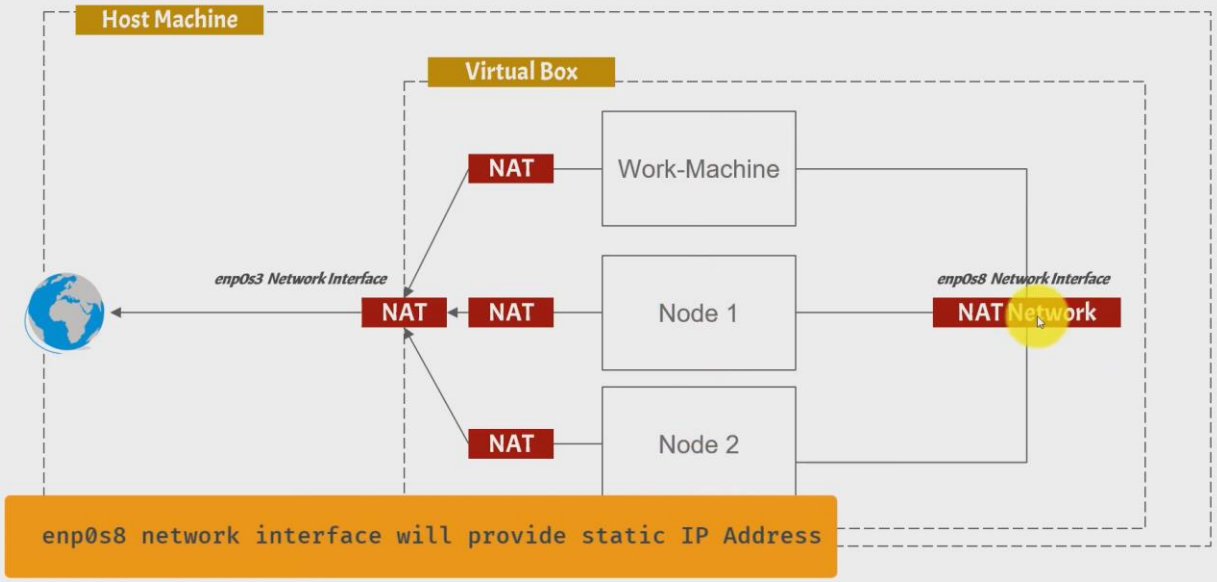
What if the machine restarts



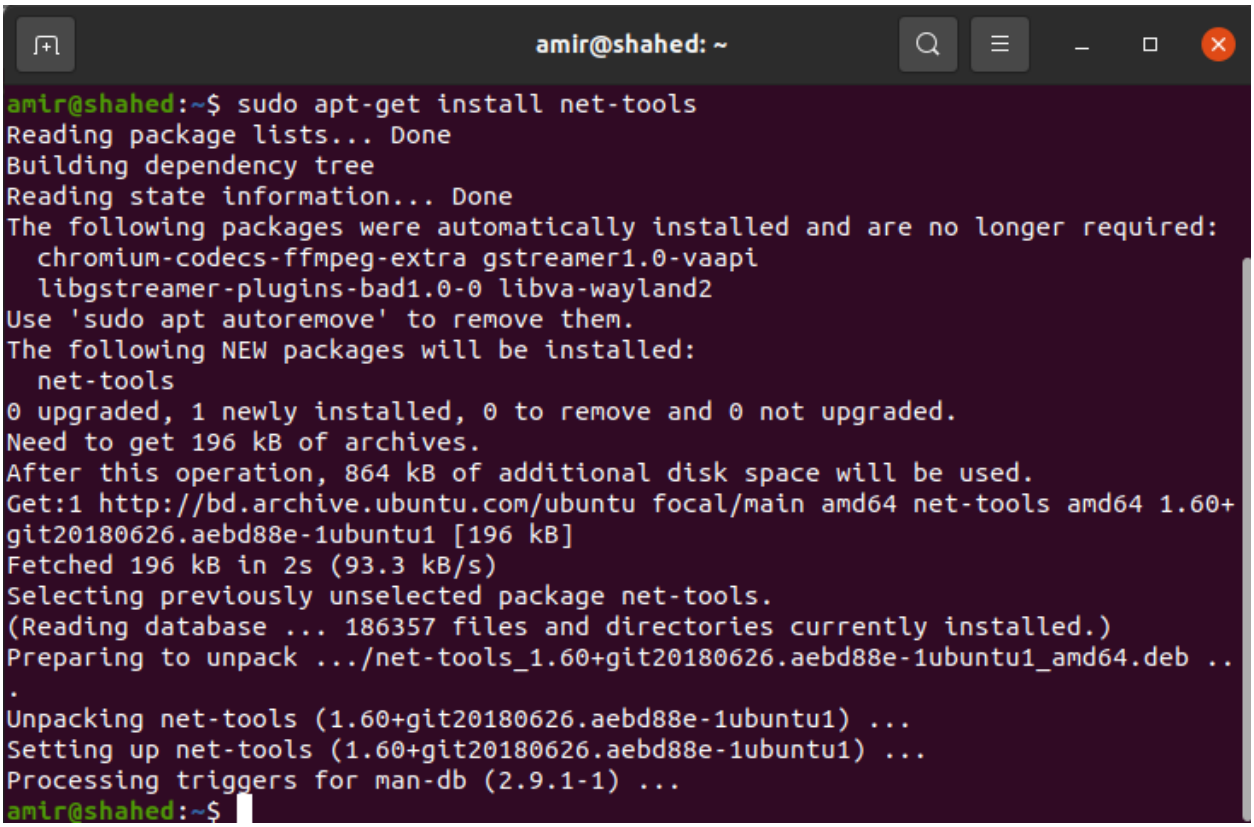
If the IP Address changes then it creates problem

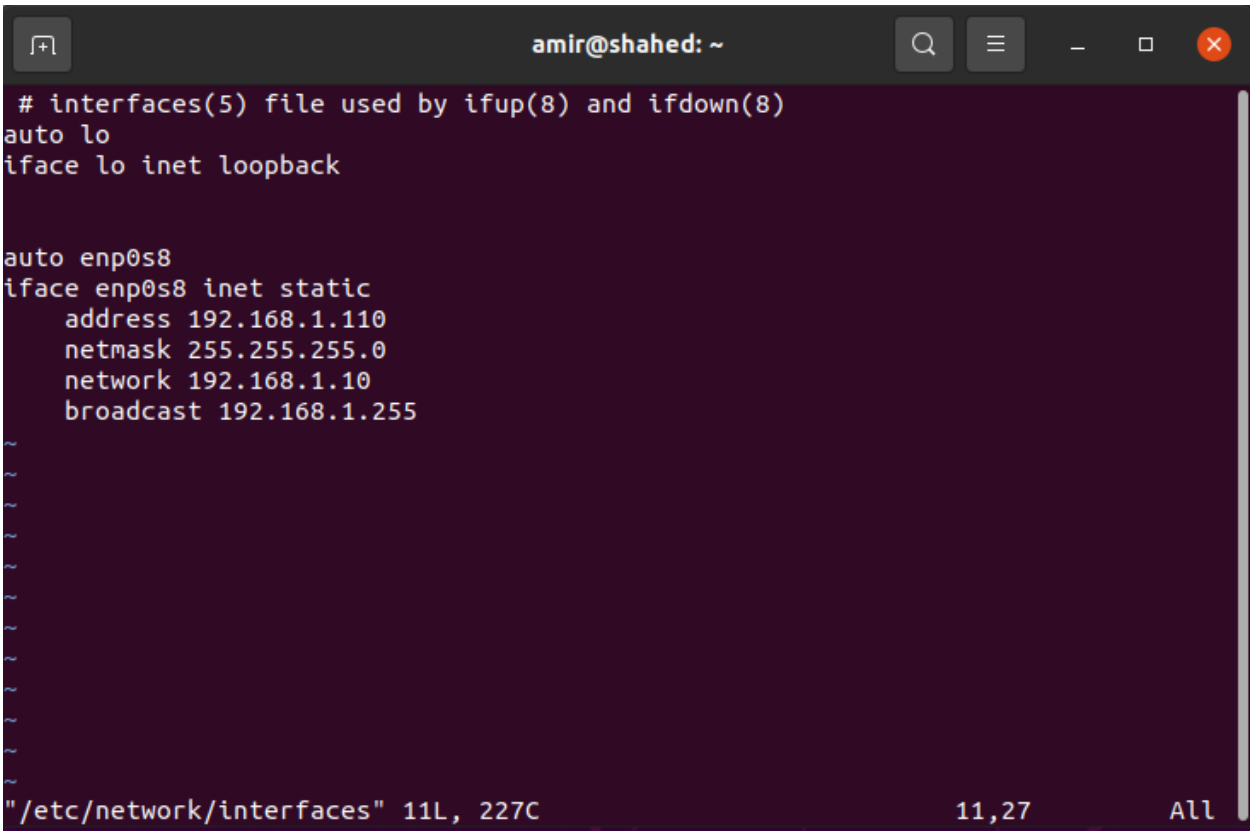
We will provide a static IP Address to all the VM

Network Setup



enp0s8 network interface will provide static IP Address


amir@shahed: ~
amir@shahed:~$ sudo apt-get install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://bd.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+
git20180626.aebd88e-1ubuntu1 [196 kB]
Fetched 196 kB in 2s (93.3 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 186357 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ..
.
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
amir@shahed:~$


amir@shahed: ~
interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto enp0s8
iface enp0s8 inet static
address 192.168.1.110
netmask 255.255.255.0
network 192.168.1.10
broadcast 192.168.1.255
"/etc/network/interfaces" 11L, 227C
11,27
All

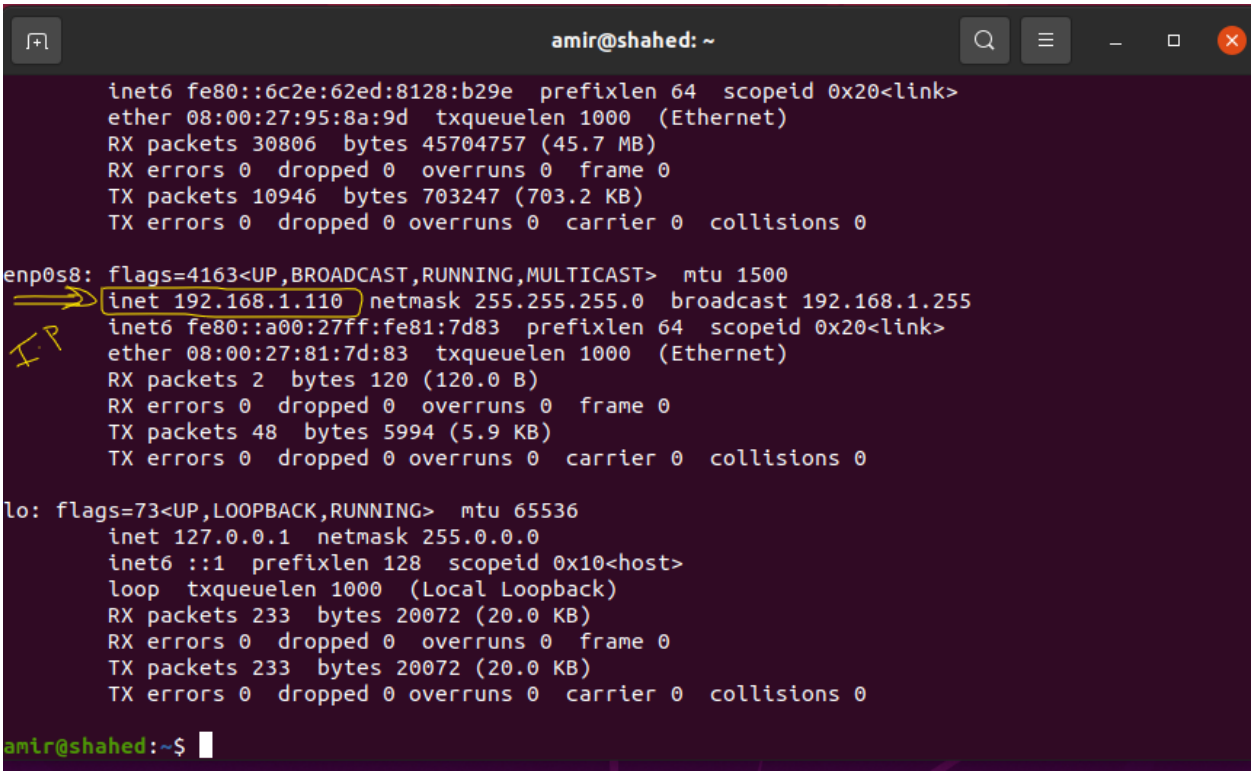
We will create our own NAT Network interface in virtualbox

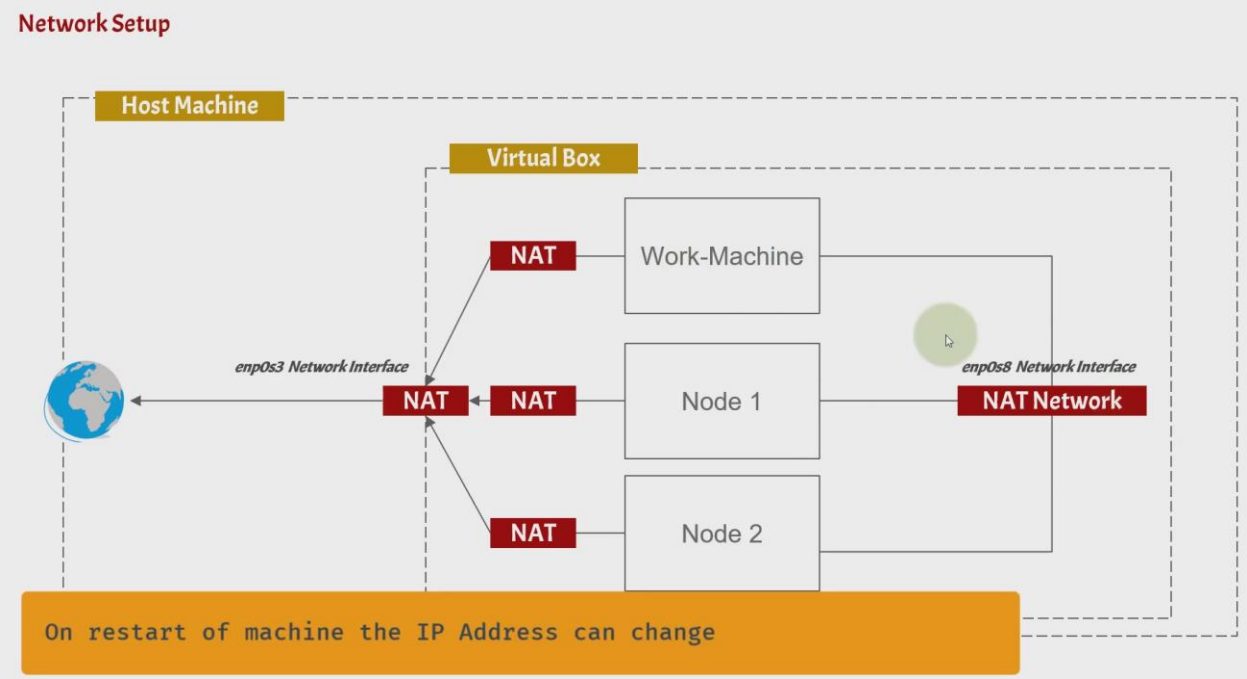
The name of the network interface in Ubuntu is called

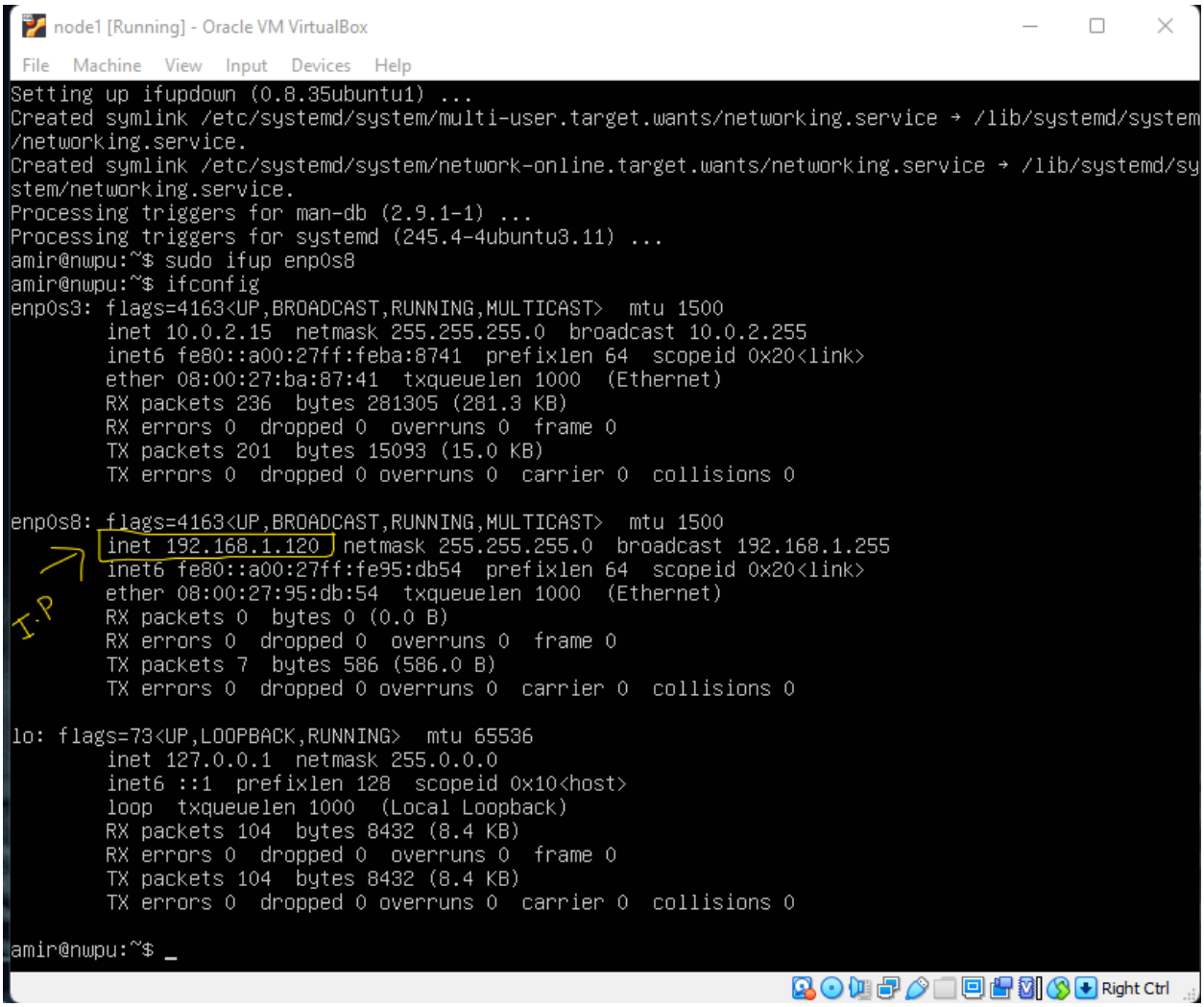
enp0s8

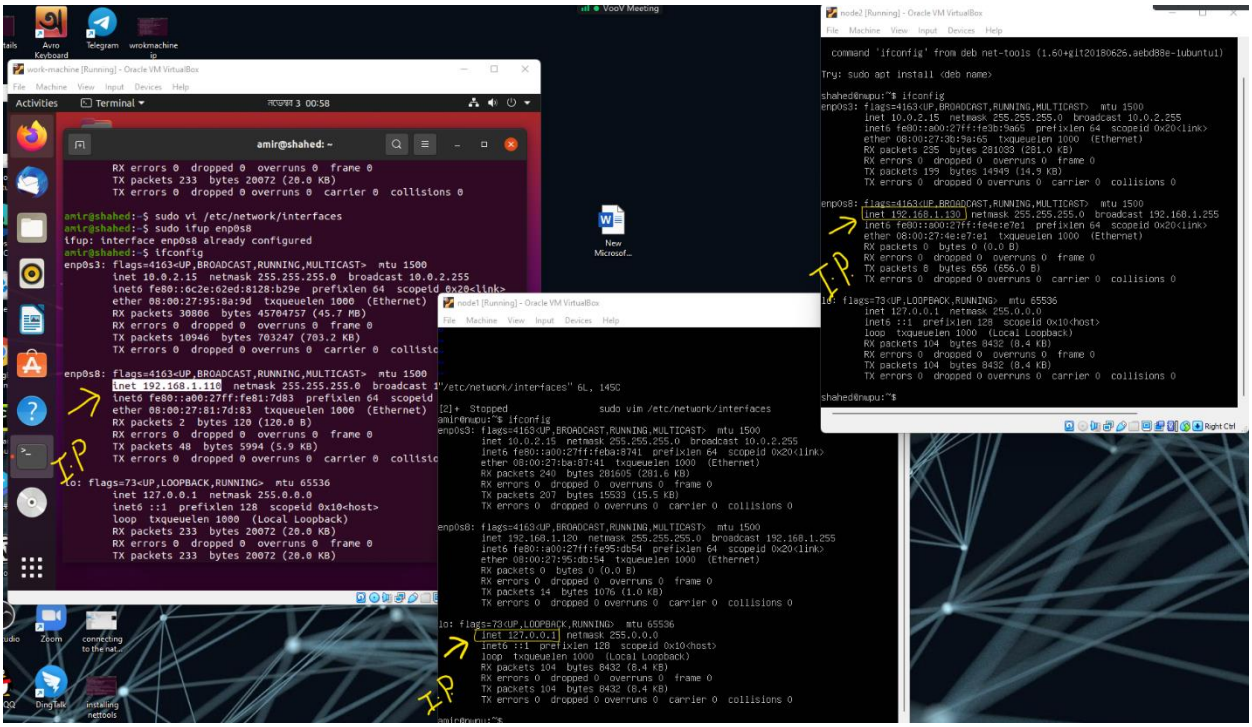
Then we will configure a static IP Address to each Virtual Machine

```
auto enp0s8
iface enp0s8 inet static
   address 192.168.1.100
   netmask 255.255.255.0
   network 192.168.1.0
   broadcast 192.168.1.255
```




Network Setup
Host Machine
Virtual Box
NAT
Work-Machine
enp0s3 Network Interface
enp0s8 Network Interface
NAT
NAT
Node 1
NAT Network
NAT
Node 2
On restart of machine the IP Address can change


node1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Setting up ifupdown (0.8.35ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/networking.service → /lib/systemd/system/networking.service.
Created symlink /etc/systemd/system/network-online.target.wants/networking.service → /lib/systemd/system/networking.service.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.11) ...
amir@nwpu:~$ sudo ifup enp0s8
amir@nwpu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::a00:27ff:feba:8741 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:ba:87:41 txqueuelen 1000 (Ethernet)
RX packets 236 bytes 281305 (281.3 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 201 bytes 15093 (15.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.120 netmask 255.255.255.0 broadcast 192.168.1.255
I.P
inet6 fe80::a00:27ff:fe95:db54 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:95:db:54 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7 bytes 586 (586.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 104 bytes 8432 (8.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 104 bytes 8432 (8.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
amir@nwpu:~$
Right Ctrl



We will create our own NAT Network interface in virtualbox

The name of the network interface in Ubuntu is called

enp0s8

Then we will configure a static IP Address to each Virtual Machine

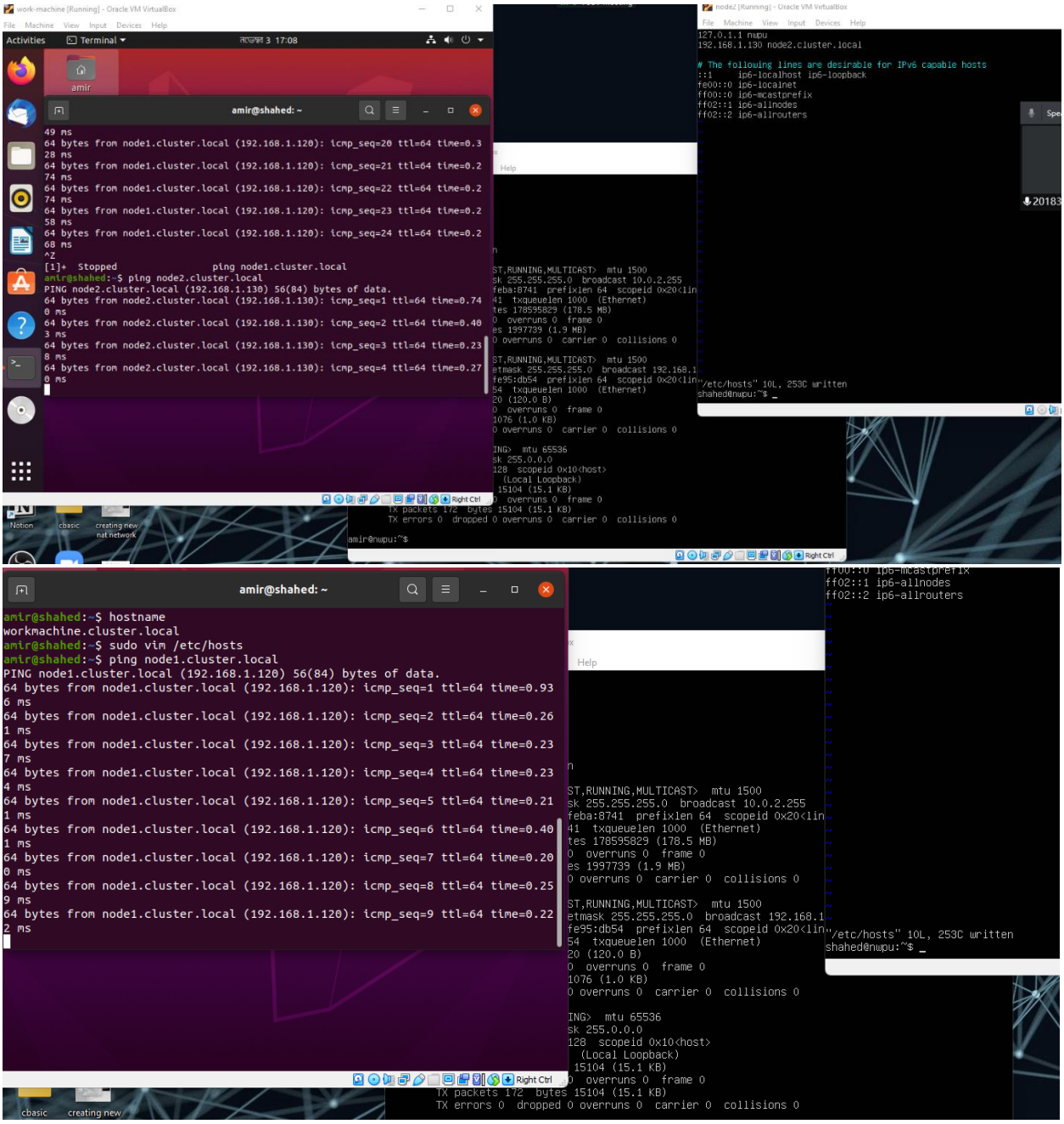
```
auto enp0s8
iface enp0s8 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
```

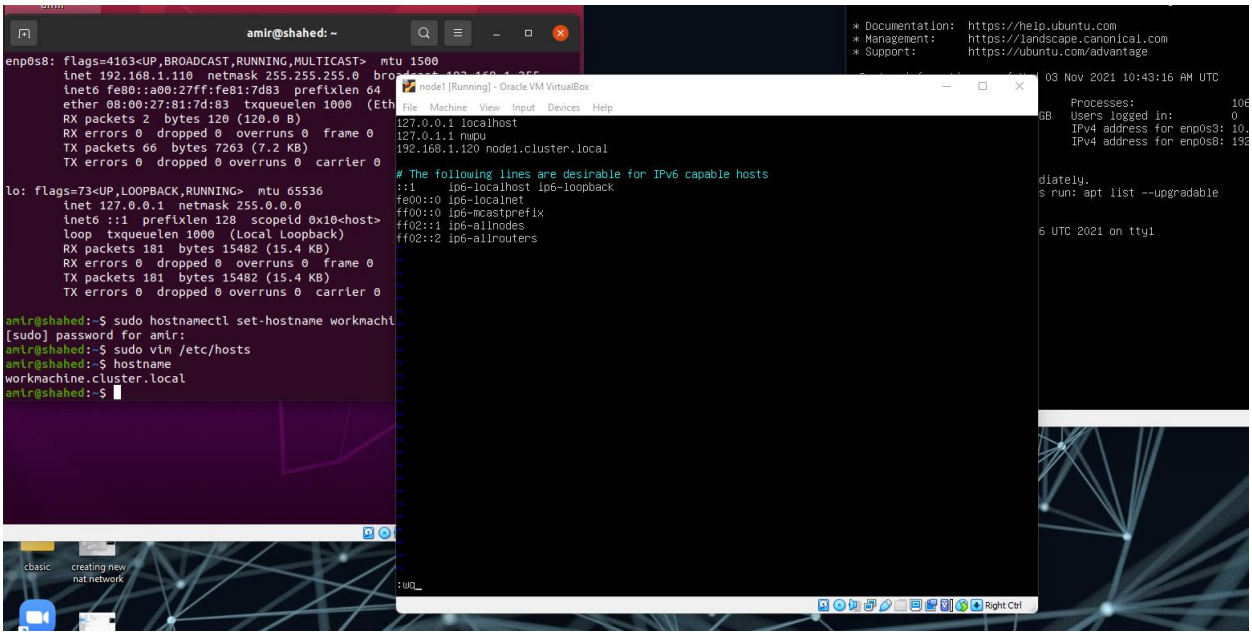
#SSH:

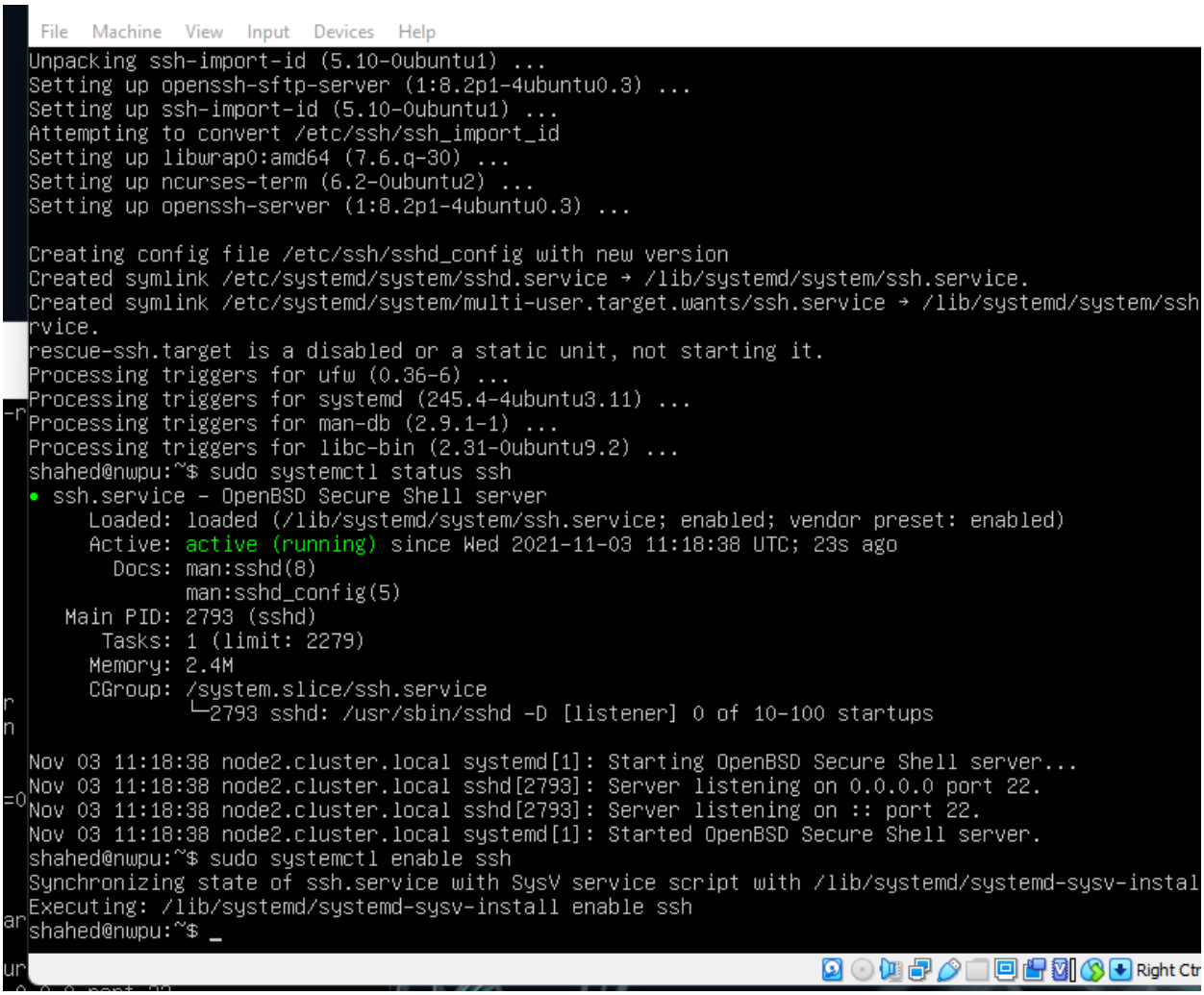
Cluster SSH (cssh) is a utility that allows us to manage multiple servers over SSH from a single administration console. It was originally designed to work with multiple

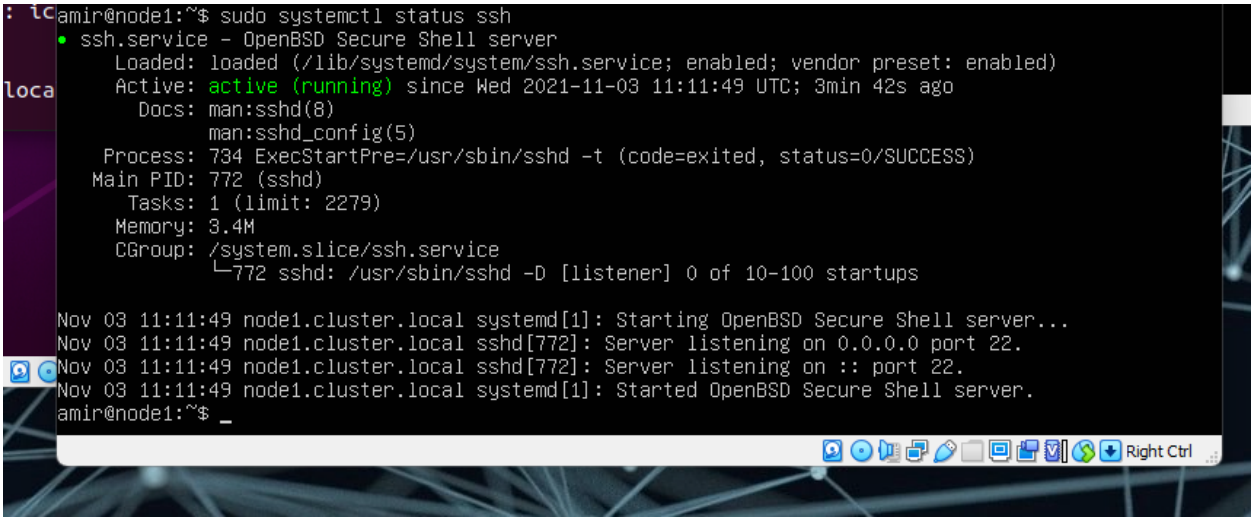
nodes that make up a HPC (High Performance Computing) cluster. These nodes are usually configured identically, therefore requiring the same administrative command to be run on each node. Using Cluster SSH allows an administrator to type a command into a single console and have it replicated over many systems.

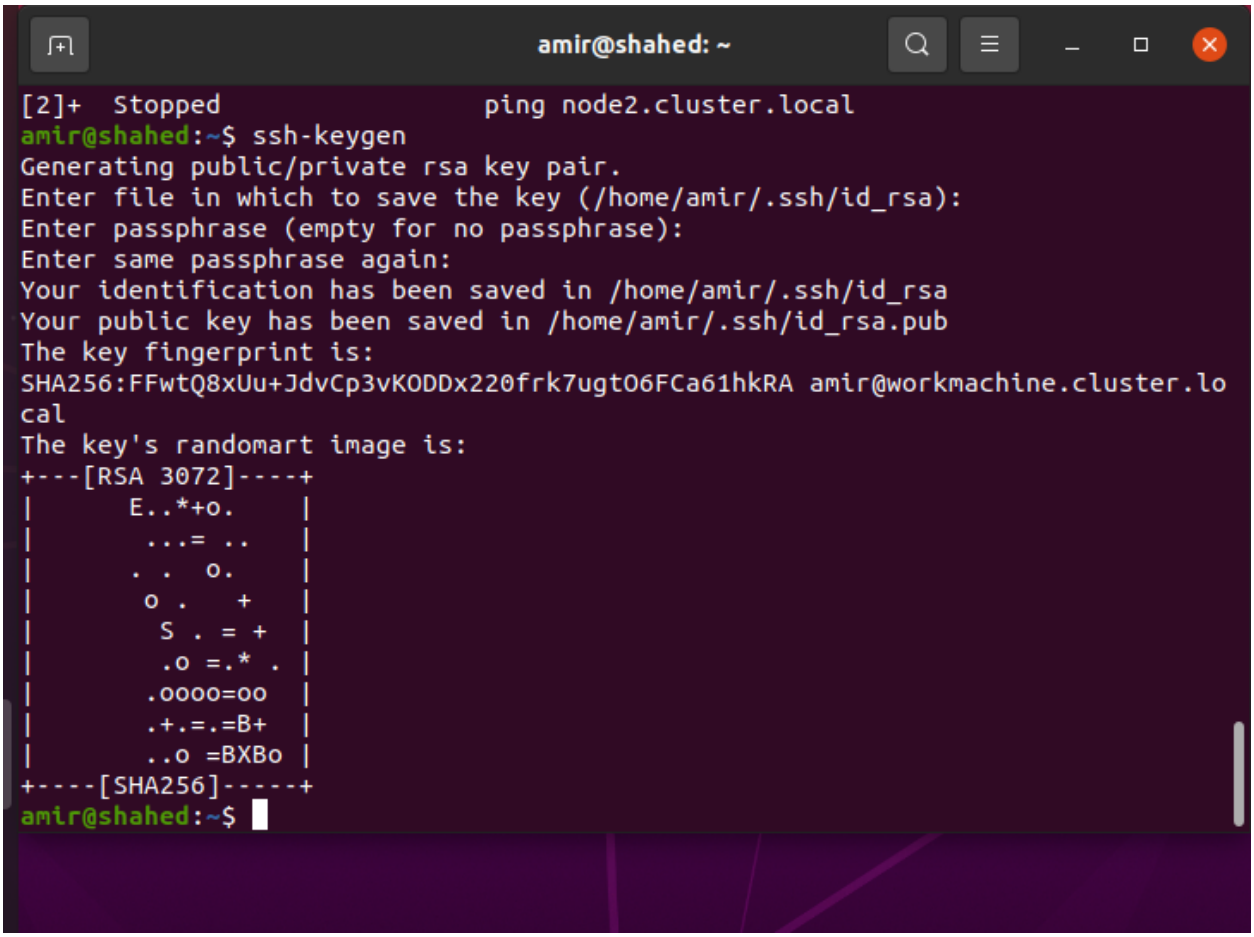
As a SysAdmin this tool can be a huge time saver.


work-machine [Running] - Oracle VM VirtualBox
Activities
Terminal
amir@shahed: ~
49 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=20 ttl=64 time=0.3
28 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=21 ttl=64 time=0.2
74 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=22 ttl=64 time=0.2
74 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=23 ttl=64 time=0.2
58 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=24 ttl=64 time=0.2
68 ms
^Z
[1]+ Stopped ping node1.cluster.local
amir@shahed:~$ ping node2.cluster.local
PING node2.cluster.local (192.168.1.130) 56(84) bytes of data.
64 bytes from node2.cluster.local (192.168.1.130): icmp_seq=1 ttl=64 time=0.74
0 ms
64 bytes from node2.cluster.local (192.168.1.130): icmp_seq=2 ttl=64 time=0.40
3 ms
64 bytes from node2.cluster.local (192.168.1.130): icmp_seq=3 ttl=64 time=0.23
8 ms
64 bytes from node2.cluster.local (192.168.1.130): icmp_seq=4 ttl=64 time=0.27
0 ms
node2 [Running] - Oracle VM VirtualBox
127.0.1.1 nupu
192.168.1.130 node2.cluster.local
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
"/etc/hosts" 10L, 253C written
shahed@nupu:~$
amir@nupu:~$
Notion
cbasic
creating new nat network
amir@shahed: ~
amir@shahed:~$ hostname
workmachine.cluster.local
amir@shahed:~$ sudo vim /etc/hosts
amir@shahed:~$ ping node1.cluster.local
PING node1.cluster.local (192.168.1.120) 56(84) bytes of data.
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=1 ttl=64 time=0.93
6 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=2 ttl=64 time=0.26
1 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=3 ttl=64 time=0.23
7 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=4 ttl=64 time=0.23
4 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=5 ttl=64 time=0.21
1 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=6 ttl=64 time=0.40
1 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=7 ttl=64 time=0.20
0 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=8 ttl=64 time=0.25
9 ms
64 bytes from node1.cluster.local (192.168.1.120): icmp_seq=9 ttl=64 time=0.22
2 ms
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
"/etc/hosts" 10L, 253C written
shahed@nupu:~$
cbasic
creating new


amir@shahed: ~
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.110 netmask 255.255.255.0
inet6 fe80::a00:27ff:fe81:7d83 prefixlen 64
ether 08:00:27:81:7d:83 txqueuelen 1000
RX packets 2 bytes 120 (120.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 66 bytes 7263 (7.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 181 bytes 15482 (15.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 181 bytes 15482 (15.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0
amir@shahed:~$ sudo hostnamectl set-hostname workmachi
[sudo] password for amir:
amir@shahed:~$ sudo vim /etc/hosts
amir@shahed:~$ hostname
workmachine.cluster.local
amir@shahed:~$
node1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
127.0.0.1 localhost
127.0.1.1 nwpu
192.168.1.120 node1.cluster.local
The following lines are desirable for IPv6 capable hosts
::1   ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
:wq_
Right Ctrl
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
03 Nov 2021 10:43:16 AM UTC
Processes:
Users logged in:
IPv4 address for enp0s3:
IPv4 address for enp0s8:
diately.
s run: apt list --upgradable
6 UTC 2021 on tty1
cbasic
creating new nat network

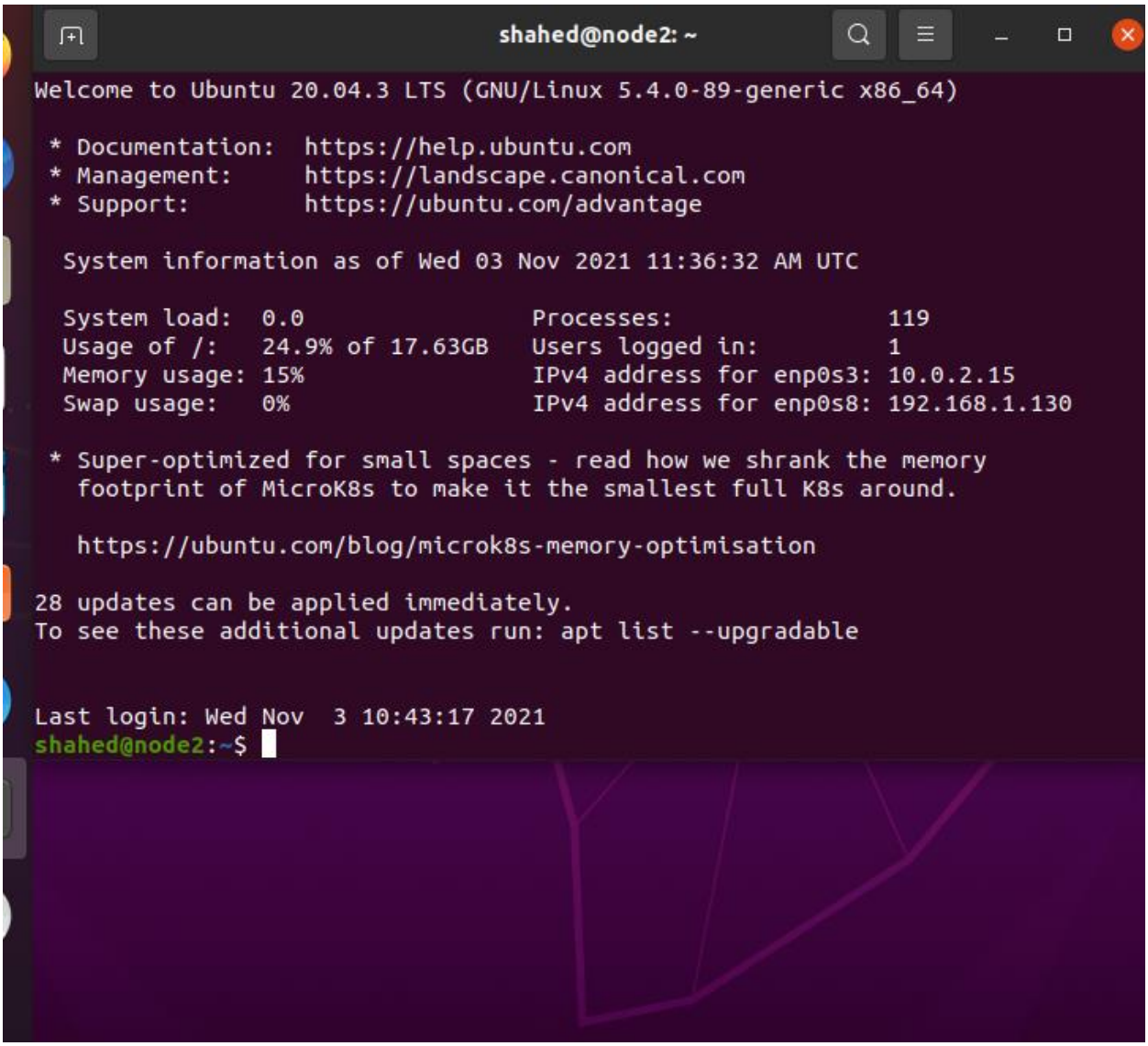

File Machine View Input Devices Help
Unpacking ssh-import-id (5.10-0ubuntu1) ...
Setting up openssh-sftp-server (1:8.2p1-4ubuntu0.3) ...
Setting up ssh-import-id (5.10-0ubuntu1) ...
Attempting to convert /etc/ssh/ssh_import_id
Setting up libwrap0:amd64 (7.6.q-30) ...
Setting up ncurses-term (6.2-0ubuntu2) ...
Setting up openssh-server (1:8.2p1-4ubuntu0.3) ...
Creating config file /etc/ssh/sshd_config with new version
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh
rvice.
rescue-ssh.target is a disabled or a static unit, not starting it.
Processing triggers for ufw (0.36-6) ...
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
shahed@nwpu:~$ 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 Wed 2021-11-03 11:18:38 UTC; 23s ago
Docs: man:sshd(8)
man:sshd_config(5)
Main PID: 2793 (sshd)
Tasks: 1 (limit: 2279)
Memory: 2.4M
CGroup: /system.slice/ssh.service
└─2793 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
Nov 03 11:18:38 node2.cluster.local systemd[1]: Starting OpenBSD Secure Shell server...
Nov 03 11:18:38 node2.cluster.local sshd[2793]: Server listening on 0.0.0.0 port 22.
Nov 03 11:18:38 node2.cluster.local sshd[2793]: Server listening on :: port 22.
Nov 03 11:18:38 node2.cluster.local systemd[1]: Started OpenBSD Secure Shell server.
shahed@nwpu:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-instal
Executing: /lib/systemd/systemd-sysv-install enable ssh
shahed@nwpu:~$ _
Right Ctr

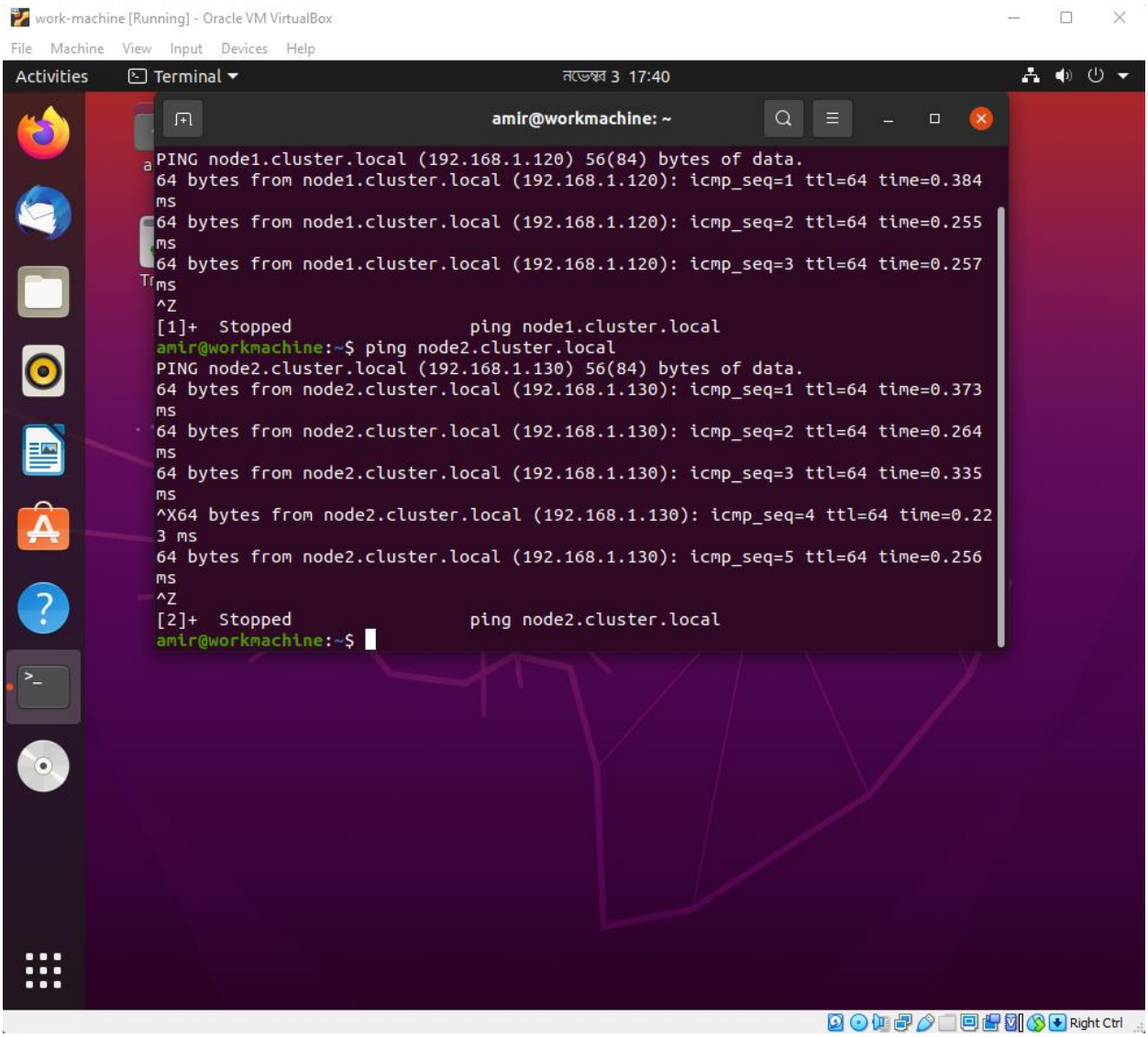

amir@node1:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2021-11-03 11:11:49 UTC; 3min 42s ago
Docs: man:sshd(8)
man:sshd_config(5)
Process: 734 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
Main PID: 772 (sshd)
Tasks: 1 (limit: 2279)
Memory: 3.4M
CGroup: /system.slice/ssh.service
└─772 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
Nov 03 11:11:49 node1.cluster.local systemd[1]: Starting OpenBSD Secure Shell server...
Nov 03 11:11:49 node1.cluster.local sshd[772]: Server listening on 0.0.0.0 port 22.
Nov 03 11:11:49 node1.cluster.local sshd[772]: Server listening on :: port 22.
Nov 03 11:11:49 node1.cluster.local systemd[1]: Started OpenBSD Secure Shell server.
amir@node1:~$ _
Right Ctrl


amir@shahed: ~
[2]+ Stopped ping node2.cluster.local
amir@shahed:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/amir/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/amir/.ssh/id_rsa
Your public key has been saved in /home/amir/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:FFwtQ8xUu+JdvCp3vKODDx220frk7ugtO6FCa61hkRA amir@workmachine.cluster.lo
cal
The key's randomart image is:
+---[RSA 3072]----+
| E..*+o. |
| ...= .. |
| . . o. |
| o . + |
| S . = + |
| .o =.* . |
| .oooo=oo |
| .+.=.=B+ |
| ..o =BXBo |
+----[SHA256]-----+
amir@shahed:~$

```
amir@shahed:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/amir/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/amir/.ssh/id_rsa
Your public key has been saved in /home/amir/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:FFwtQ8xUu+JdvCp3vKODDx220frk7ugtO6FCa61hkRA amir@workmachine.cluster.local
The key's randomart image is:
+---[RSA 3072]----+
|      E..*+o.    |
|       ...= ..   |
|       . .  o.   |
|        o .   +  |
|         S . = + |
|          .o =.* |
|        .oooo=oo |
|        .+.=.=B+ |
|        ..o =BXBo|
+----[SHA256]-----+
amir@shahed:~$ cd .ssh/
amir@shahed:~/.ssh$ ls
id_rsa  id_rsa.pub
amir@shahed:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub amir@node
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/amir/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: ERROR: ssh: Could not resolve hostname node: Temporary failure in name resolution

amir@shahed:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub amir@node1.cluster.local
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/amir/.ssh/id_rsa.pub"
The authenticity of host 'node1.cluster.local (192.168.1.120)' can't be established.
ECDSA key fingerprint is SHA256:6vqFOc9DVzCBu/oIXvmbGuIcECbgM3PKRvYXhKvekhc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
amir@node1.cluster.local's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'amir@node1.cluster.local'"
and check to make sure that only the key(s) you wanted were added.
```

```
amir@shahed:~/.ssh$ ssh node1.cluster.local
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic x86_64)

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

  System information as of Wed 03 Nov 2021 11:27:15 AM UTC

  System load:  0.0               Processes:               117
  Usage of /:   30.2% of 14.58GB  Users logged in:         1
  Memory usage: 10%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.1.120


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


Last login: Wed Nov  3 11:12:14 2021
amir@node1:~$ exit
logout
Connection to node1.cluster.local closed.
amir@shahed:~/.ssh$
```


shahed@node2: ~
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Wed 03 Nov 2021 11:36:32 AM UTC
System load: 0.0
Usage of /: 24.9% of 17.63GB
Memory usage: 15%
Swap usage: 0%
Processes: 119
Users logged in: 1
IPv4 address for enp0s3: 10.0.2.15
IPv4 address for enp0s8: 192.168.1.130
* Super-optimized for small spaces - read how we shrank the memory
footprint of MicroK8s to make it the smallest full K8s around.
https://ubuntu.com/blog/microk8s-memory-optimisation
28 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Last login: Wed Nov 3 10:43:17 2021
shahed@node2:~$



#NIS

The Network Information Service, or NIS (initially called YP or yellow pages), is a mainframe-client index service convention for circulating server configuration information, for example, client and host names between PCs on a PC network. Sun Microsystems built up the NIS; the innovation is authorized to essentially all other Unix merchants. Since British Telecom PLC claimed the name “Yellow Pages” as an enlisted brand name in the United Kingdom for its paper-based, business phone catalog,

Sun changed the name of its framework to NIS, however, all the orders capacities actually start with “yp”. A NIS/YP framework keeps up and disseminates a focal index of the client and gathering information, hostnames, email pseudonyms, and other content-based tables of information in a PC network. For instance, in a typical UNIX climate, the rundown of clients for ID is put in/and so forth/passwd and mystery verification hashes in/and so on/shadow. NIS includes another “worldwide” client list which is utilized for recognizing clients on any customer of the NIS area. Administrators can arrange NIS to serve secret key information to outside cycles to verify clients utilizing different variants of the Unix crypt(3) hash calculations. In any case, in such cases, any NIS(0307) customer can recover the whole secret phrase information base for disconnected investigation. Kerberos was intended to deal with confirmation in a safer way.

We utilize the Linux NIS mainframe (Network Information Service) for sharing basic information put away in level documents between frameworks on a network. It is frequently ideal to have a mutual archive, (for example, NIS) for putting away clients and gatherings information as opposed to putting away them in level documents like/and so forth/passwd. So what is the advantage of that? By making such documents accessible through the NIS worker, that would permit any distant NIS customer machine to access or question the information in these

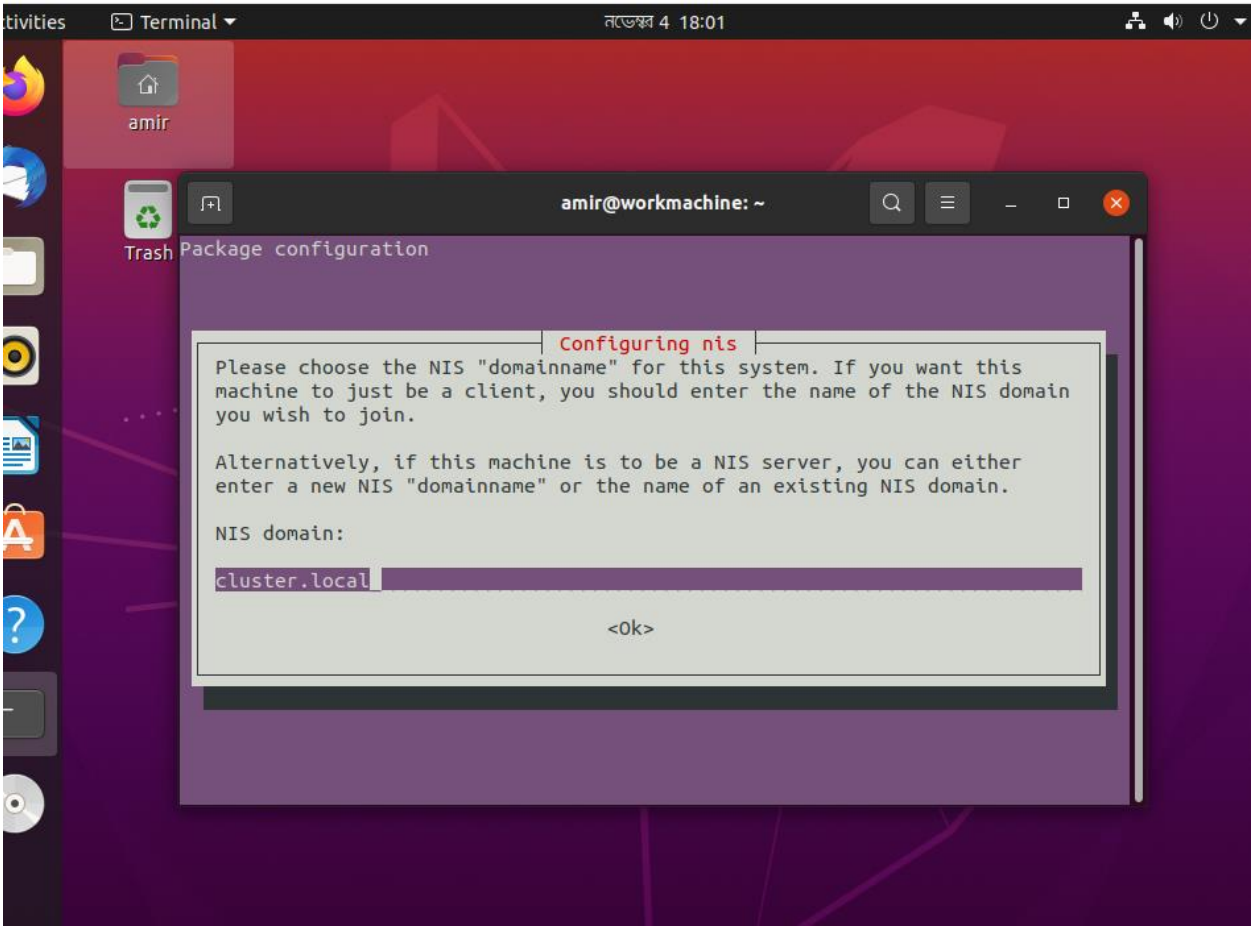
mutual records and use them as expansions to the nearby forms.

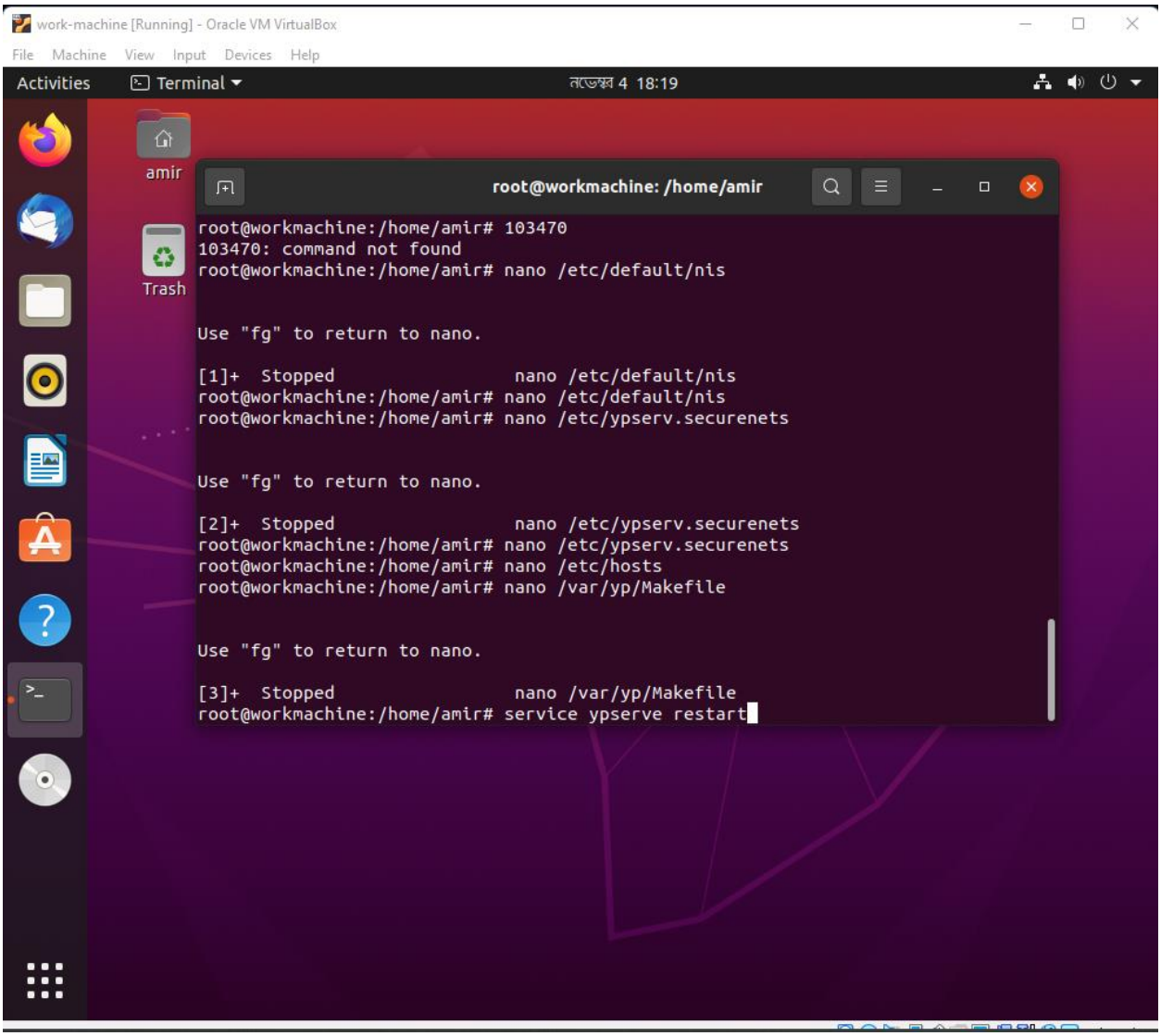
NIS isn’t for sharing records. We can share any even document which in any event has one section with an extraordinary worth by means of NIS like/and so on/services record. The primary advantage of utilizing the NIS worker is that you keep your information and records, and spread any updates to all clients. A few clients, particularly Windows clients, may think this is a kind of Active Directory like service. The Linux NIS worker is more established than Active Directory and not a reproduce for it.

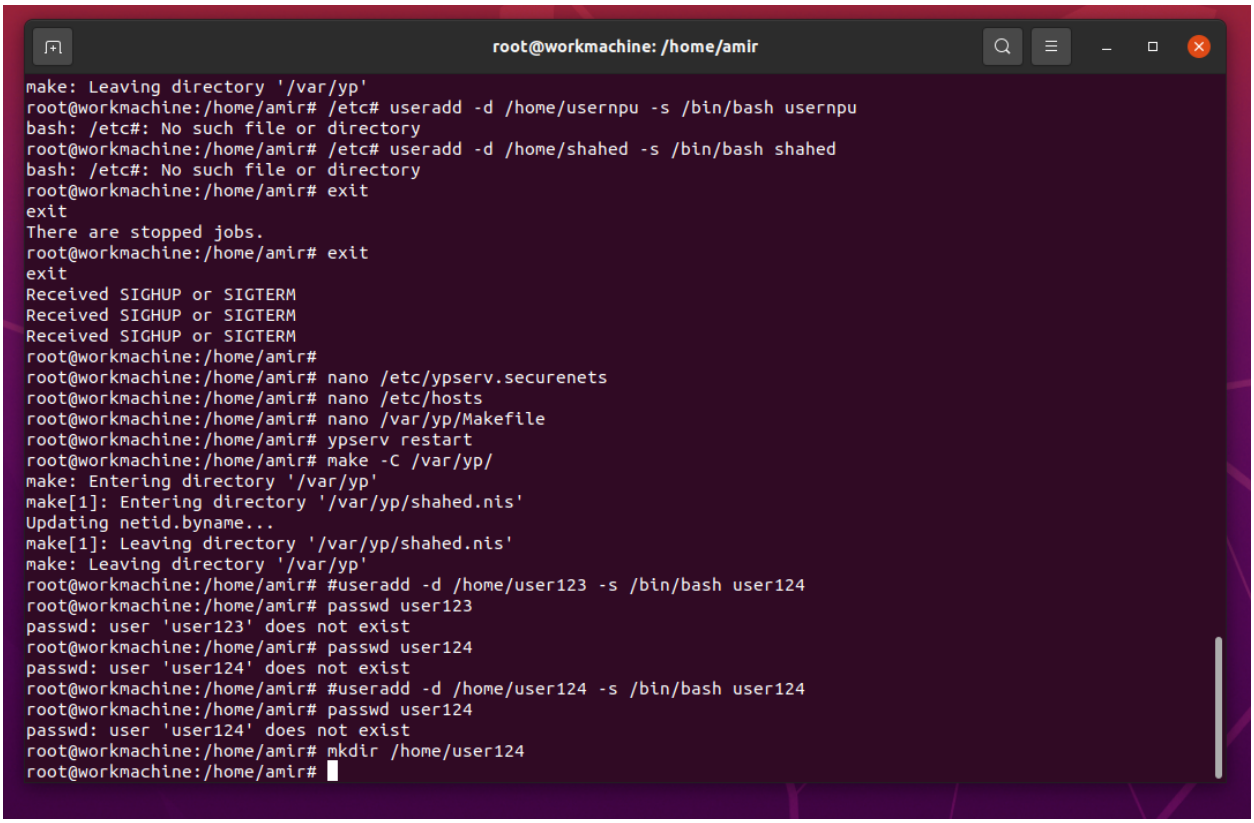
By running NIS, the framework administrator can disperse administrative information bases, called maps, among an assortment of mainframes (ace and slaves). The administrator can refresh those information bases from a brought together area in a programmed and solid design to guarantee that all customers share a similar naming service information predictably all through the network. NIS was grown freely of DNS and has a marginally extraordinary core interest. While DNS centers around making correspondence less difficult by utilizing machine names rather than mathematical IP addresses, NIS centers around making network administration more sensible by giving unified command over an assortment of network information. NIS stores information about machine names and addresses, yet additionally about

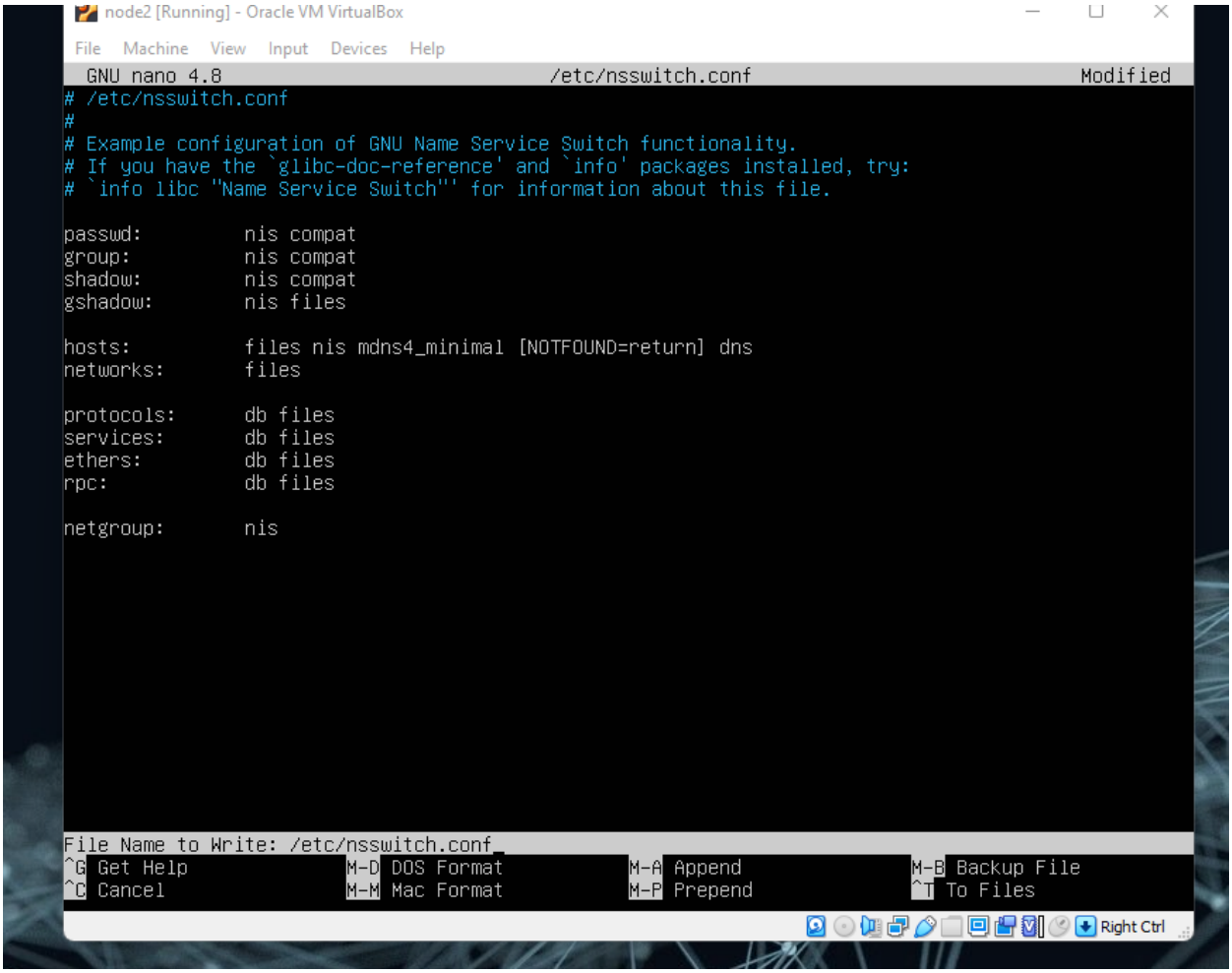
clients, the network itself, and network services. This assortment of network information is alluded to as the NIS namespace.

Configuration:


tivities
Terminal
নভেম্বর 4 18:01
amir
Trash
amir@workmachine: ~
Package configuration
Configuring nis
Please choose the NIS "domainname" for this system. If you want this machine to just be a client, you should enter the name of the NIS domain you wish to join.
Alternatively, if this machine is to be a NIS server, you can either enter a new NIS "domainname" or the name of an existing NIS domain.
NIS domain:
cluster.local
<Ok>


work-machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
নভেম্বর 4 18:19
amir
Trash
root@workmachine: /home/amir
root@workmachine:/home/amir# 103470
103470: command not found
root@workmachine:/home/amir# nano /etc/default/nis
Use "fg" to return to nano.
[1]+ Stopped nano /etc/default/nis
root@workmachine:/home/amir# nano /etc/default/nis
root@workmachine:/home/amir# nano /etc/ypserv.securenets
Use "fg" to return to nano.
[2]+ Stopped nano /etc/ypserv.securenets
root@workmachine:/home/amir# nano /etc/ypserv.securenets
root@workmachine:/home/amir# nano /etc/hosts
root@workmachine:/home/amir# nano /var/yp/Makefile
Use "fg" to return to nano.
[3]+ Stopped nano /var/yp/Makefile
root@workmachine:/home/amir# service ypserve restart


root@workmachine: /home/amir
make: Leaving directory '/var/yp'
root@workmachine:/home/amir# /etc# useradd -d /home/usernpu -s /bin/bash usernpu
bash: /etc#: No such file or directory
root@workmachine:/home/amir# /etc# useradd -d /home/shahed -s /bin/bash shahed
bash: /etc#: No such file or directory
root@workmachine:/home/amir# exit
exit
There are stopped jobs.
root@workmachine:/home/amir# exit
exit
Received SIGHUP or SIGTERM
Received SIGHUP or SIGTERM
Received SIGHUP or SIGTERM
root@workmachine:/home/amir#
root@workmachine:/home/amir# nano /etc/ypserv.securenets
root@workmachine:/home/amir# nano /etc/hosts
root@workmachine:/home/amir# nano /var/yp/Makefile
root@workmachine:/home/amir# ypserv restart
root@workmachine:/home/amir# make -C /var/yp/
make: Entering directory '/var/yp'
make[1]: Entering directory '/var/yp/shahed.nis'
Updating netid.byname...
make[1]: Leaving directory '/var/yp/shahed.nis'
make: Leaving directory '/var/yp'
root@workmachine:/home/amir# #useradd -d /home/user123 -s /bin/bash user124
root@workmachine:/home/amir# passwd user123
passwd: user 'user123' does not exist
root@workmachine:/home/amir# passwd user124
passwd: user 'user124' does not exist
root@workmachine:/home/amir# #useradd -d /home/user124 -s /bin/bash user124
root@workmachine:/home/amir# passwd user124
passwd: user 'user124' does not exist
root@workmachine:/home/amir# mkdir /home/user124
root@workmachine:/home/amir#



#NFS:

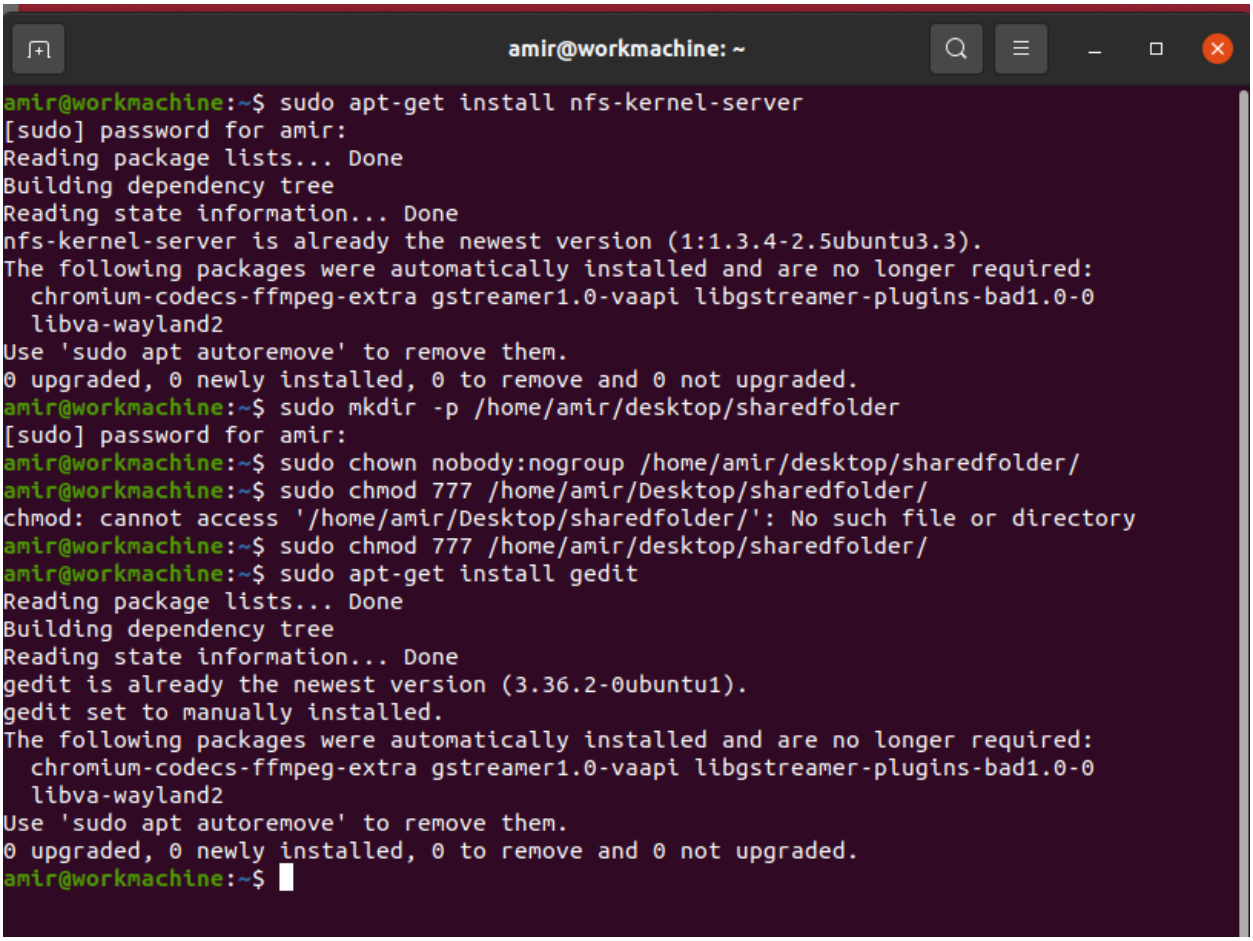
The advent of distributed computing was marked by the introduction of distributed file systems. Such systems involved multiple client machines and one or a few servers. The server stores data on its disks and the clients may request data through some protocol messages.

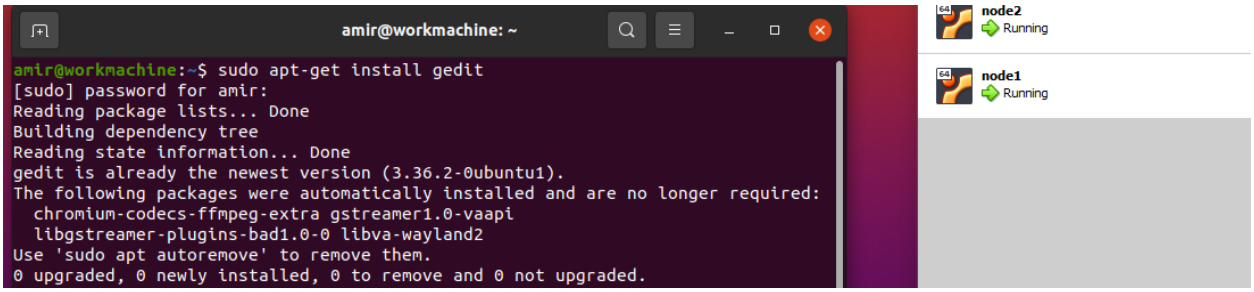
Even a simple client/server architecture involves more components than the physical file systems discussed previously in OS. The architecture consists of a client-side file system and a server-side file system. A client application issues a system call (e.g. read(), write(),

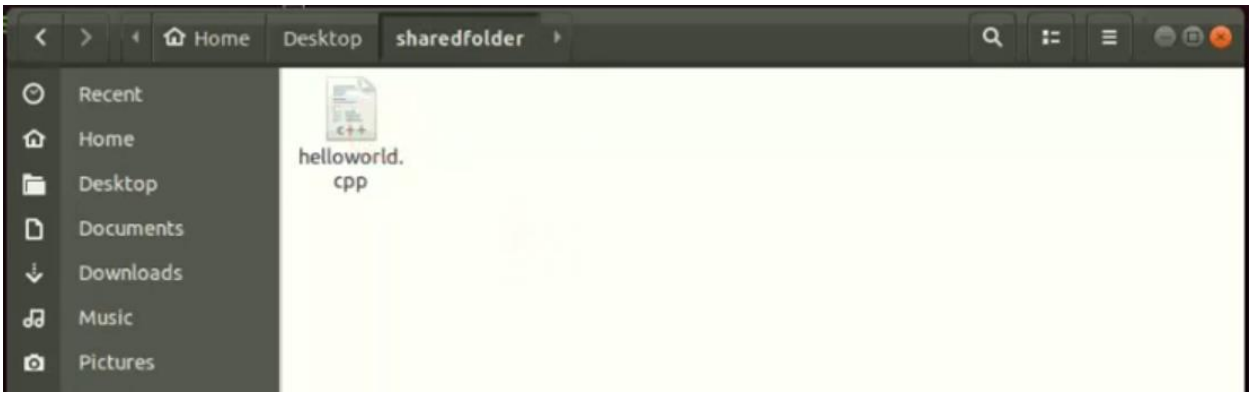
open(), close() etc.) to access files on the client-side file system, which in turn retrieves files from the server. It is interesting to note that to a client application, the process seems no different than requesting data from a physical disk, since there is no special API required to do so. This phenomenon is known as transparency in terms of file access. It is the client-side file system that executes commands to service these system calls.

For instance, assume that a client application issues the read() system call. The client-side file system then messages the server-side file system to read a block from the server's disk and return the data back to the client. Finally, it buffers this data into the read() buffer and completes the system call.

The server-side file system is also simply called the file server.


amir@workmachine: ~
amir@workmachine:~$ sudo apt-get install nfs-kernel-server
[sudo] password for amir:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-kernel-server is already the newest version (1:1.3.4-2.5ubuntu3.3).
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
amir@workmachine:~$ sudo mkdir -p /home/amir/desktop/sharedfolder
[sudo] password for amir:
amir@workmachine:~$ sudo chown nobody:nogroup /home/amir/desktop/sharedfolder/
amir@workmachine:~$ sudo chmod 777 /home/amir/Desktop/sharedfolder/
chmod: cannot access '/home/amir/Desktop/sharedfolder/': No such file or directory
amir@workmachine:~$ sudo chmod 777 /home/amir/desktop/sharedfolder/
amir@workmachine:~$ sudo apt-get install gedit
Reading package lists... Done
Building dependency tree
Reading state information... Done
gedit is already the newest version (3.36.2-0ubuntu1).
gedit set to manually installed.
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
amir@workmachine:~$


amir@workmachine: ~
amir@workmachine:~$ sudo apt-get install gedit
[sudo] password for amir:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gedit is already the newest version (3.36.2-0ubuntu1).
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
node2
Running
node1
Running


Home
Desktop
sharedfolder
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
helloworld.
cpp

MPI

MPI is a standard application programming interface for creating and executing parallel programs. MPI was originally written for C, C++ and Fortran code but implementations have since become available for a variety of other languages, including Python.

MPI programs are started as a bunch of instances (often called “processes” or “tasks”) of an executable, which run concurrently.

As each process runs, the program may need to exchange data (“messages” - hence the name) with other processes. An example of data exchanges is point-to-point communication where one process sends data to another process. In other cases data may be “gathered” from multiple processes at once and sent to a root process. Inversely, data can be “scattered” from the root process to multiple other processes in one step.

OpenMP (Open Multi-Processing) is an application programming interface (API) for shared memory multiprocessing programming in C, C++ and Fortran. An OpenMP-parallelised application starts as a serial application that runs on a single compute core. When instructed by the programmer, the application spawns a

number of threads, which can run concurrently on separate cores. Thus, work can be distributed to leverage more resources.

Note that the OpenMP standard was recently extended to enable offloading computations to GPUs and other accelerators. However, not all compilers support this feature yet and there is a similar, competing standard called OpenACC that addresses the same use case. We will limit this lesson to multicore computing without offloading.

```
amir@workmachine:~$ sudo apt-get install  openmpi-bin libopenmpi-dev
[sudo] password for amir:
Reading package lists... Done
Building dependency tree
Reading state information... Done
libopenmpi-dev is already the newest version (4.0.3-0ubuntu1).
openmpi-bin is already the newest version (4.0.3-0ubuntu1).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
amir@workmachine:~$ cd /home/amir/Desktop
amir@workmachine:~/Desktop$ cd /home/amir/Desktop
amir@workmachine:~/Desktop$ cd openmpi-4.1.1/
amir@workmachine:~/Desktop/openmpi-4.1.1$ ./configure --prefix="/home/amir/.openmpi"
checking for perl... perl

==========================================================================
== Configuring Open MPI
==========================================================================

*** Startup tests
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
```

```
amir@workmachine: ~/sharedfolder
amir@workmachine:~/sharedfolder$ mpicc MPIserver.c -o MPIserver
amir@workmachine:~/sharedfolder$ mpirun -np 1 ./MPIserver
Server available at port: 2168455169.0:2560756268
Enter the string :
AVINASH

Reversed string is : HSANIVA
```

```
amir@workmachine: ~/sharedfolder
amir@workmachine:~/sharedfolder$ mpicc MPIclient.c -o MPIclient
amir@workmachine:~/sharedfolder$ mpirun -np 1 ./MPIclient 2166161409.0:1627668610
```

Team Contribution:

2018380130(Khan Md Shahedul Islam)
2018380038(Amirbek Raimov)

- ⇨ We have created virtual cluster in both of our computer and continued through the processes simultaneously. Being completely new to parallel computing, while creating virtual cluster we might run into complexities in the later stages that might require to start all over again. To eliminate the risk we have done the complete processes in both of computers and successfully completed all the tasks. Hence, our contribution was equal.

Acknowledgement: During the process we have faced many obstacles and problems that took countless hours of searching and trials but at the end we are happy to learn the vital skills of parallel computing and we are thankful to our teacher who have taught us really well and provided us with the best possible resources.

The end