EXPERIMENT III

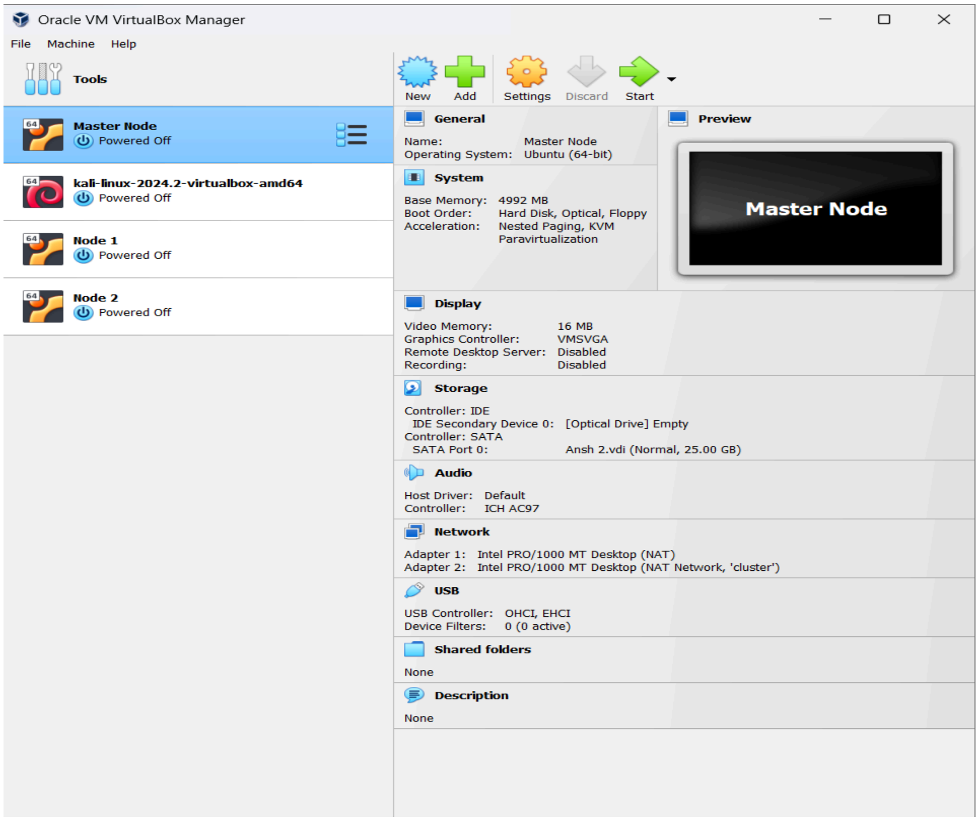
VIRTUAL BOX

AIM **Create the VM cluster where two nodes serve as a worker node and one machine works as a maser node. Master node must be able to ssh the worker node through ssh.**

1 Create two work node (server/desktop) and assign resources.

1.)Set Up Two Machines (Physical or Virtual using VirtualBox/VMware).
2.)Install Ubuntu/Debian/CentOS on both machines.
3.) Assign Resources:

CPU: 2 cores
RAM: 4GB
Storage: 40GB



Master Node
Powered Off

kali-linux-2024.2-virtualbox-amd64
Powered Off

Node 1
Powered Off

Node 2
Powered Off

General
Name: Master Node
Operating System: Ubuntu (64-bit)

System
Base Memory: 4992 MB
Boot Order: Hard Disk, Optical, Floppy
Acceleration: Nested Paging, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Ansh 2.vdi (Normal, 25.00 GB)

Audio
Host Driver: Default
Controller: ICH AC97

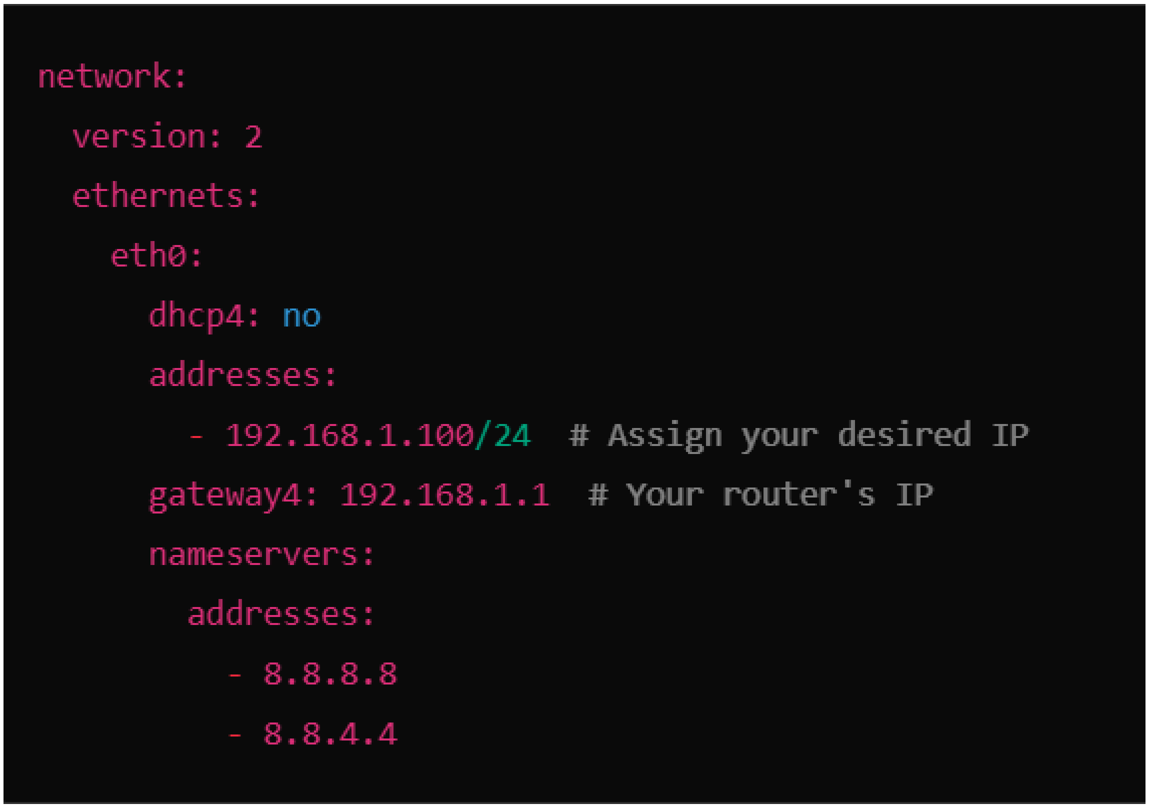
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (NAT Network, 'cluster')

USB
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders
None

Description
None

2	**Create one master node with desktop machine.**
	Set up a desktop machine with Ubuntu/Debian/CentOS. 2.) Assign resources: CPU: 4 cores RAM: 8GB Storage: 100GB 3.) Connect to the network and ensure communication with worker nodes. 4.) Enable remote access for managing worker nodes. 5.) Monitor system performance to ensure smooth operation.
3	**Assign the static IP address for all the nodes.**
	Run the following commands: 1.)Find Your Network Interface Name: ip a 2.)Edit Network Configuration File: sudo nano /etc/netplan/01-netcfg.yaml # For Ubuntu/Debian sudo nano /etc/sysconfig/network-scripts/ifcfg-eth0 # For CentOS

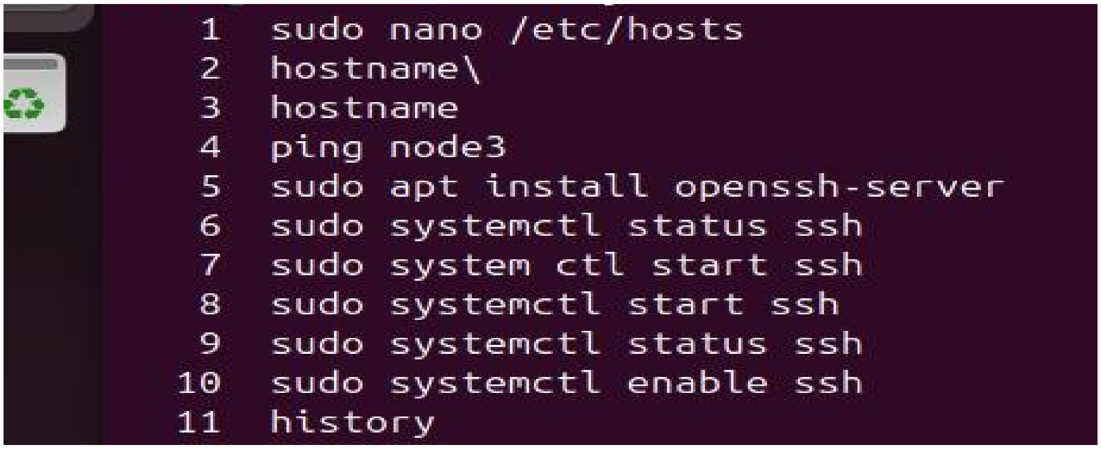


```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.1.100/24  # Assign your desired IP
      gateway4: 192.168.1.1  # Your router's IP
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

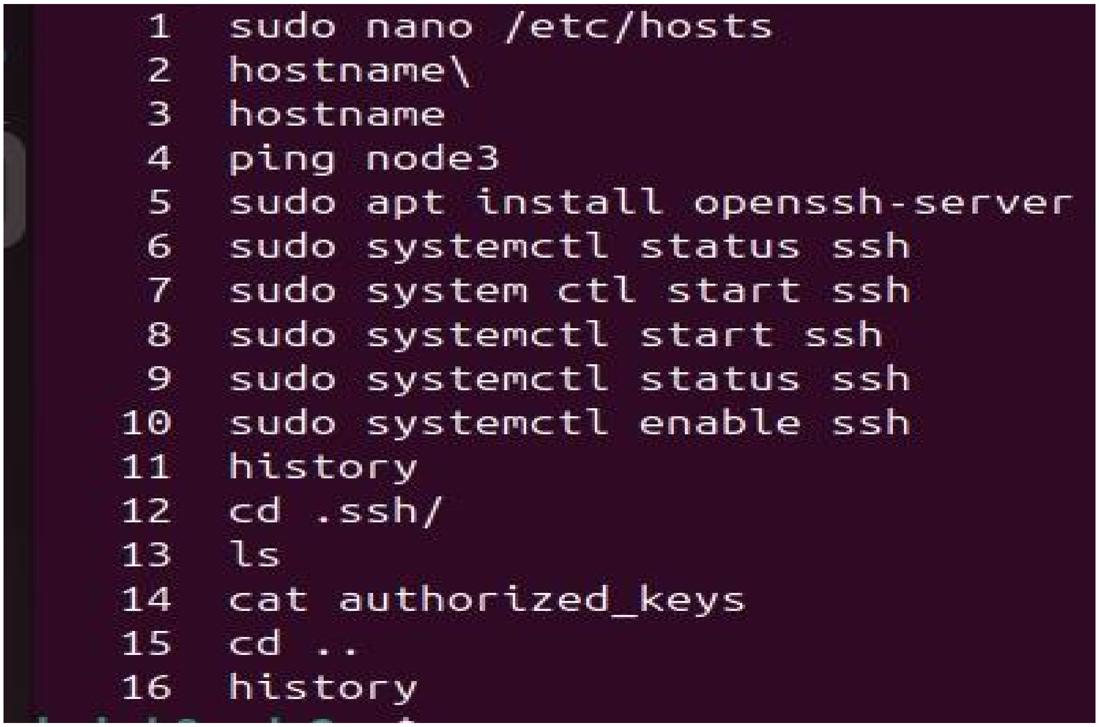
4	**Change the host name to node1, node2 and master_node.**
	Open the Hostname Configuration File E.g: sudo nano /etc/hostname Modify the Hostname: For Master Node, set it to: master_node Apply Changes: sudo hostnamectl set-hostname master_node # Change "master_node" to "node1" or "node2" accordingly.

5	**ssh node from the master node.**
	Ensure SSH is Installed on All Nodes On master node and worker nodes, install SSH: sudo apt install openssh-server -y # For Ubuntu/Debian sudo yum install openssh-server -y # For CentOS **Start & enable SSH:** sudo systemctl enable ssh sudo systemctl start ssh **Check Node Connectivity:** ping node1 ping node2
6	**Run any program on the node from the master node.**
	Ensure SSH is Set Up Between Master and Nodes. Ensure SSH is Set Up Make sure the master node can connect to the worker nodes via SSH. If required, set up passwordless SSH to avoid entering passwords repeatedly. 2.) Prepare the Program or Script Write the program or script you want to execute. Save it on the master node. 3.)Transfer the Program to the Node Copy the program or script from the master node to the target node. 4.) Execute the Program on the Node From the master node, remotely run the program on the worker node using SSH. 5.)Verify Execution Check the output or logs on the worker node to ensure the program ran successfully. Now, your master node can run programs on worker nodes remotely!

HISTORY OF COMMANDS PERFORMED:



```
    1  sudo nano /etc/hosts
    2  hostname\
    3  hostname
    4  ping node3
    5  sudo apt install openssh-server
    6  sudo systemctl status ssh
    7  sudo system ctl start ssh
    8  sudo systemctl start ssh
    9  sudo systemctl status ssh
   10  sudo systemctl enable ssh
   11  history
```



```
    1  sudo nano /etc/hosts
    2  hostname\
    3  hostname
    4  ping node3
    5  sudo apt install openssh-server
    6  sudo systemctl status ssh
    7  sudo system ctl start ssh
    8  sudo systemctl start ssh
    9  sudo systemctl status ssh
   10  sudo systemctl enable ssh
   11  history
   12  cd .ssh/
   13  ls
   14  cat authorized_keys
   15  cd ..
   16  history
```