

Cloud Computing Assignment - 1

Name: T.V.S.S.Sripad

Roll Number: 18CS01008

Q1) Implement virtualization using Oracle's Virtual Box and VMWare:-

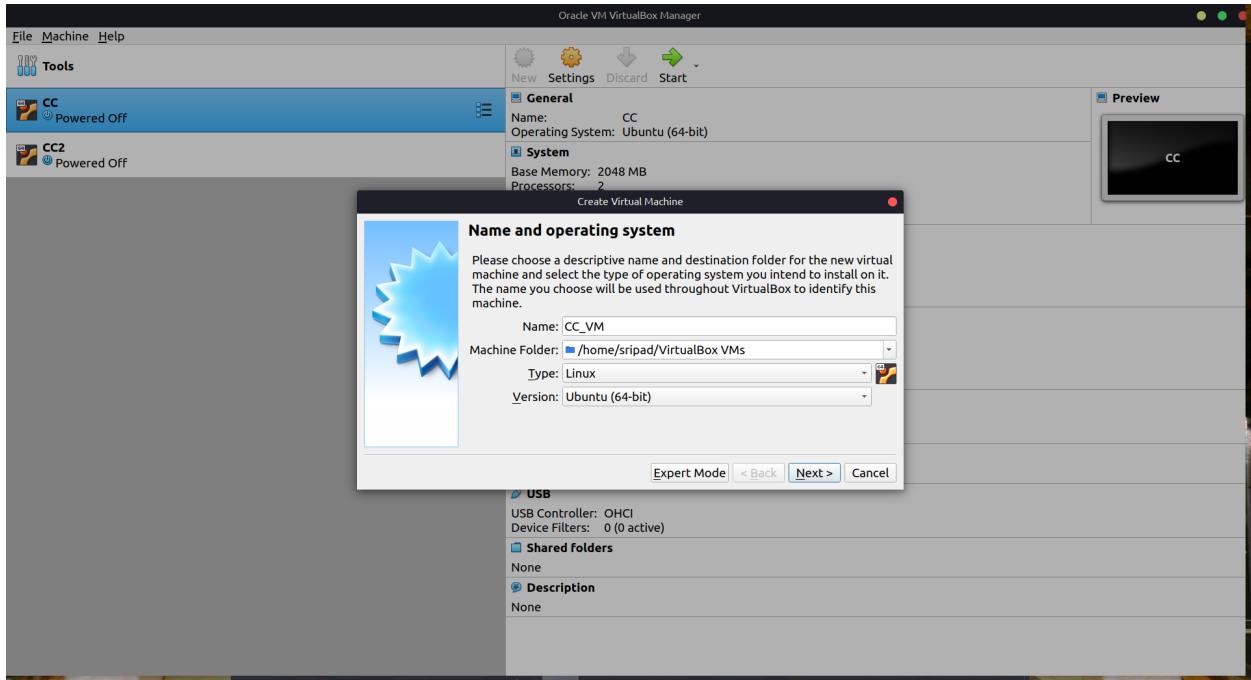
Steps followed for creating VM in Oracle VBox:-

- 1) First I installed Oracle VBox on my system.
- 2) After installation, I clicked on the `New` button and first I chose a name for the VM and I also selected the Operating system of the VM.
- 3) Next, I allocated 2GB ram to the VM and I created a virtual hard-disk of 10GB.
- 4) After this, the VM will be created.
- 5) Next, I assigned 2 processors to the VM in system settings screen of VM. Then I enabled 3D acceleration in display settings.
- 6) Finally, I attached the ISO file of ubuntu (for installing on VM) in the storage settings menu.

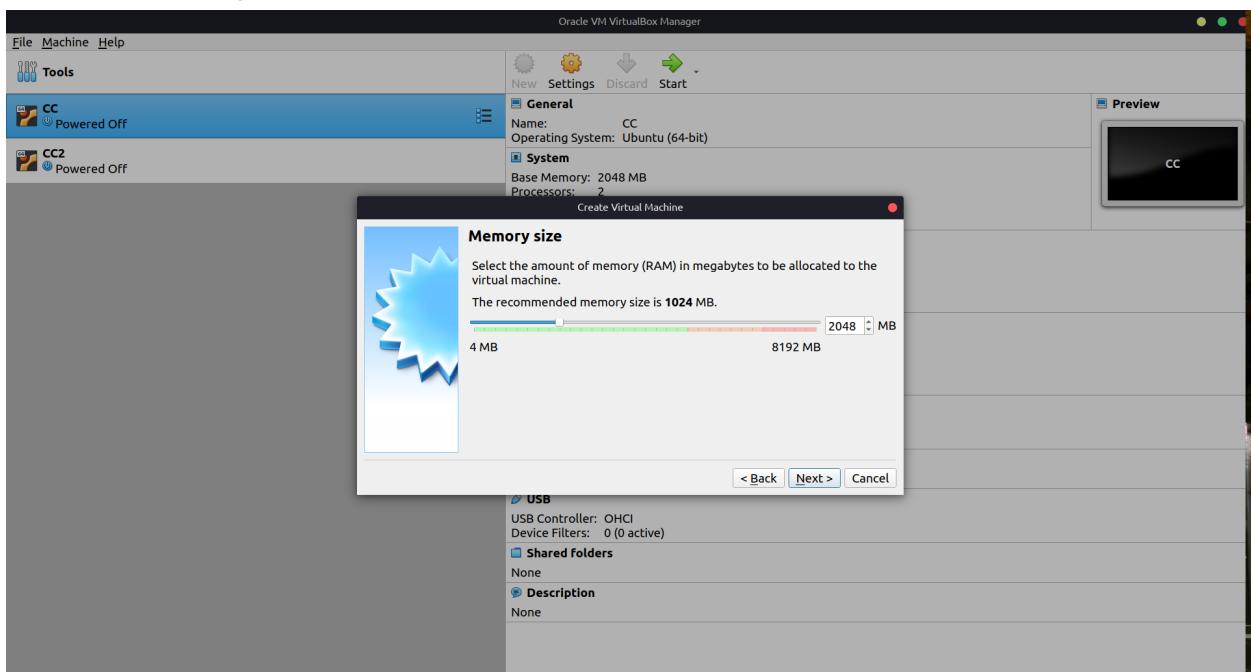
For this assignment, I have created 3 Virtual Machines named as CC, CC2, CC_VM. All the VMs have 2GB ram, 10GB storage and all VMs run Ubuntu-20 OS. All the VMs have 2 processors each. For facilitating the communication between VMs, I chose the network of VMs to be “NAT Network” which creates a private network between all the VMs.

Screenshots of the process of creating Ubuntu-VM on Oracle VBox:-

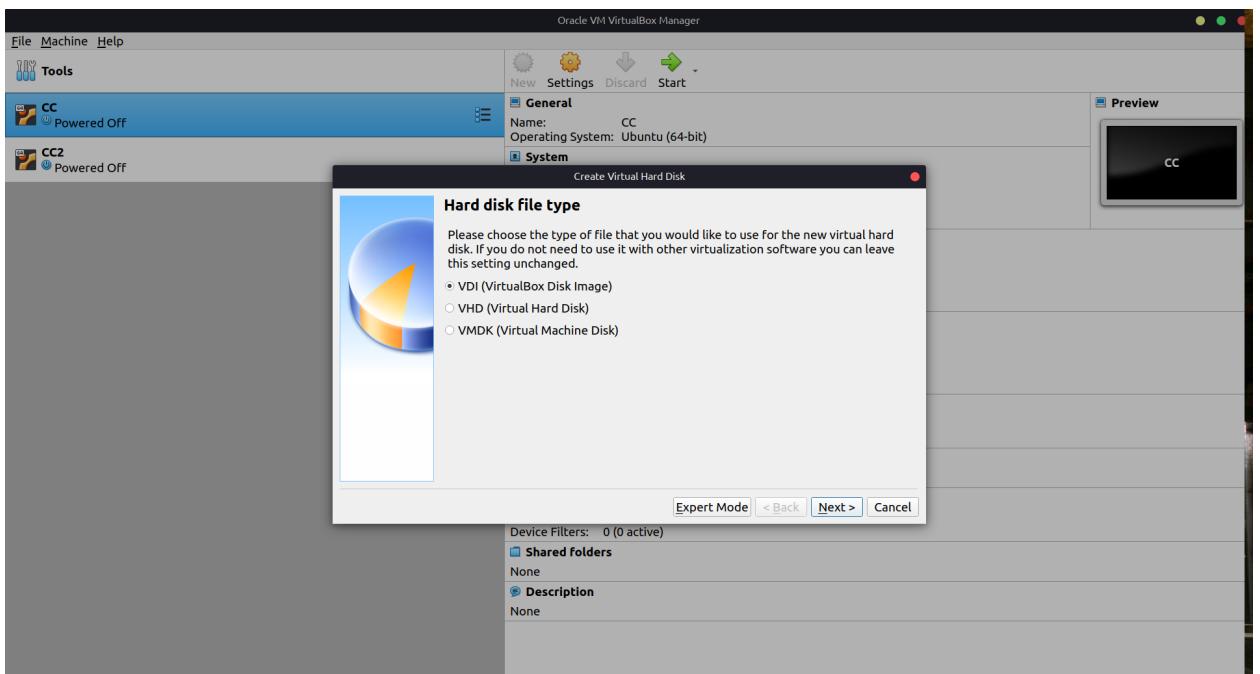
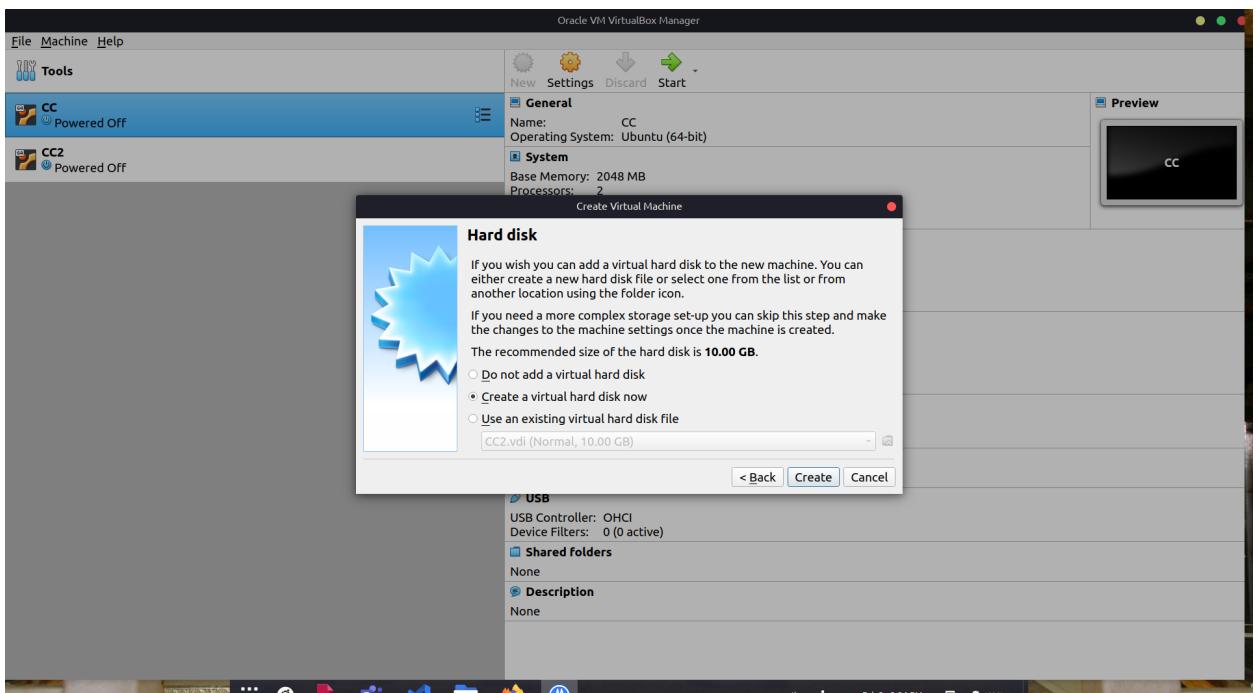
1) Creating VM: Selecting the OS of VM

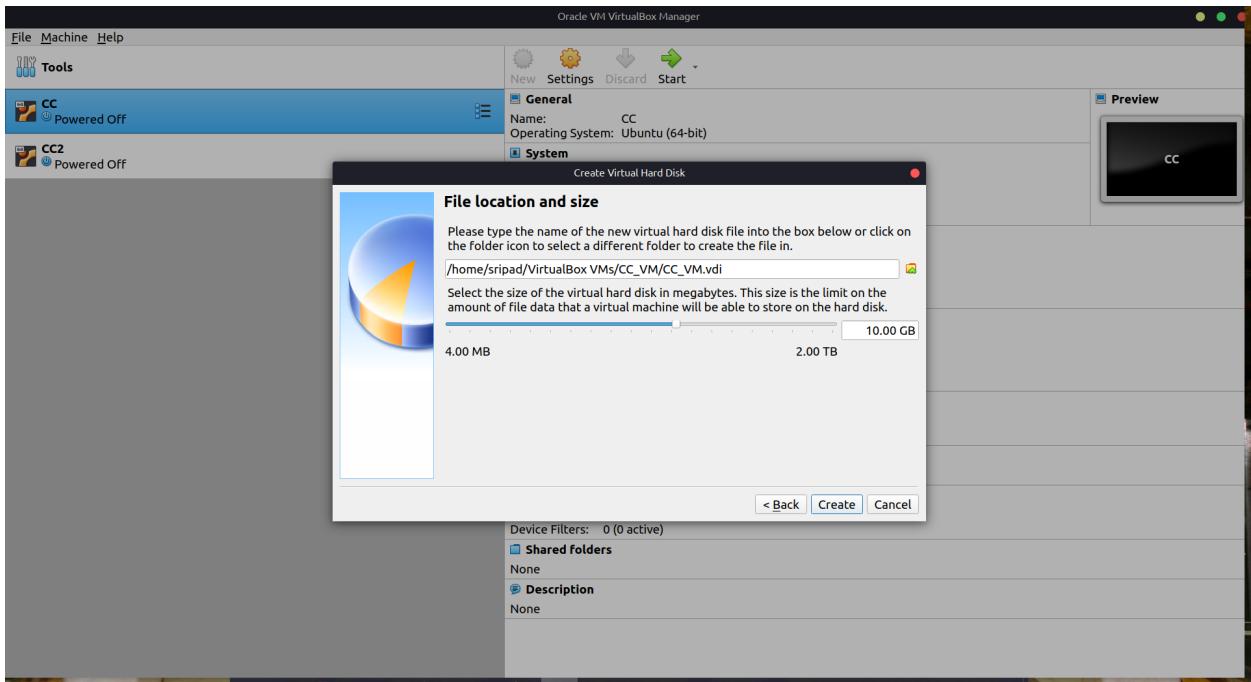


2) Selecting RAM:-



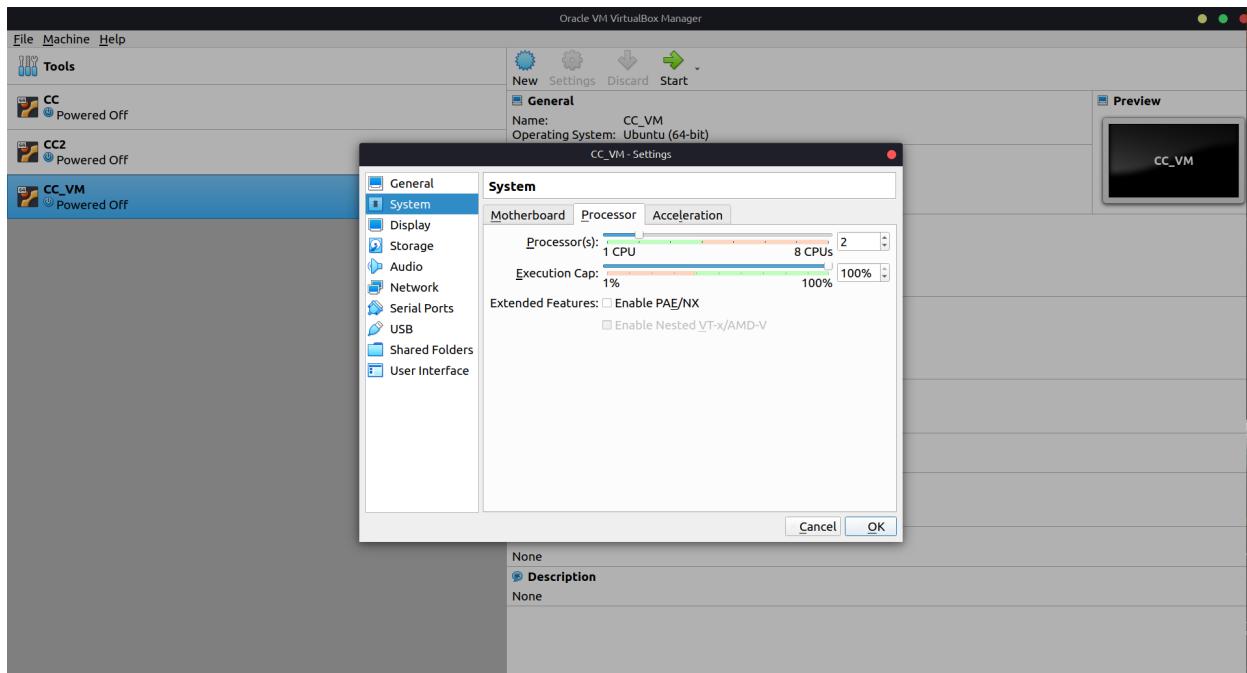
3) Selecting Hard-disk type:-



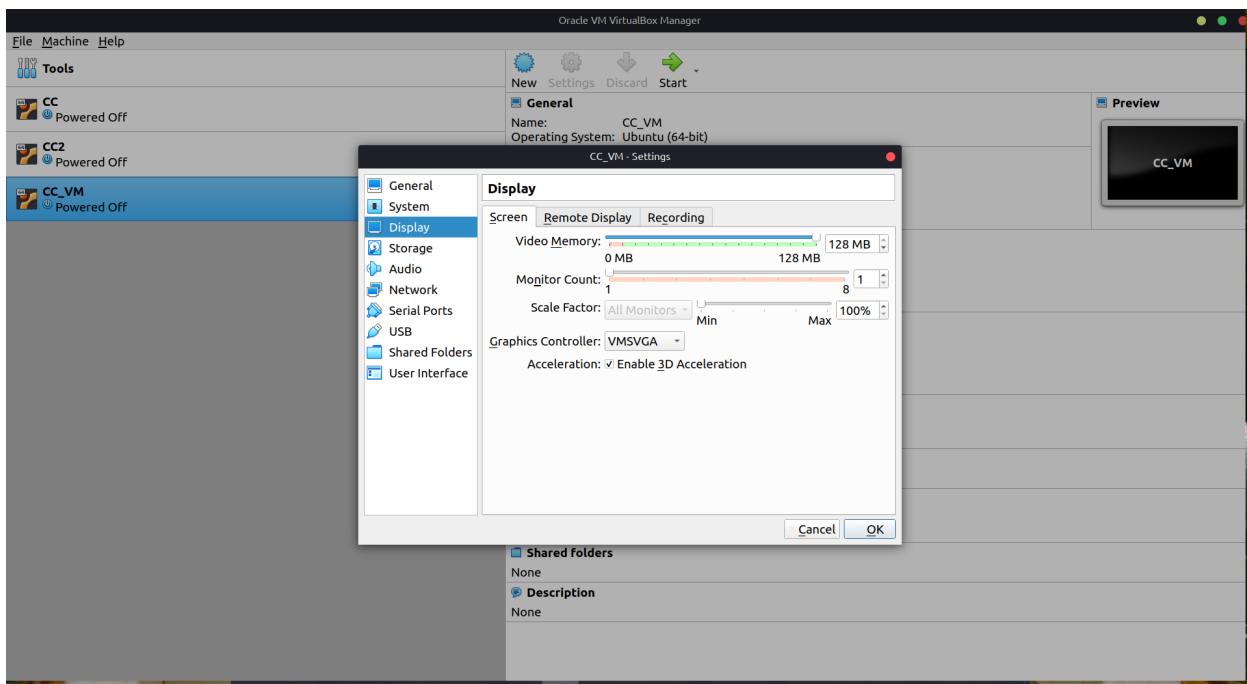


After this VM is created. Next, I change the number of processors, display settings and I attach ISO file for installation of os on VM.

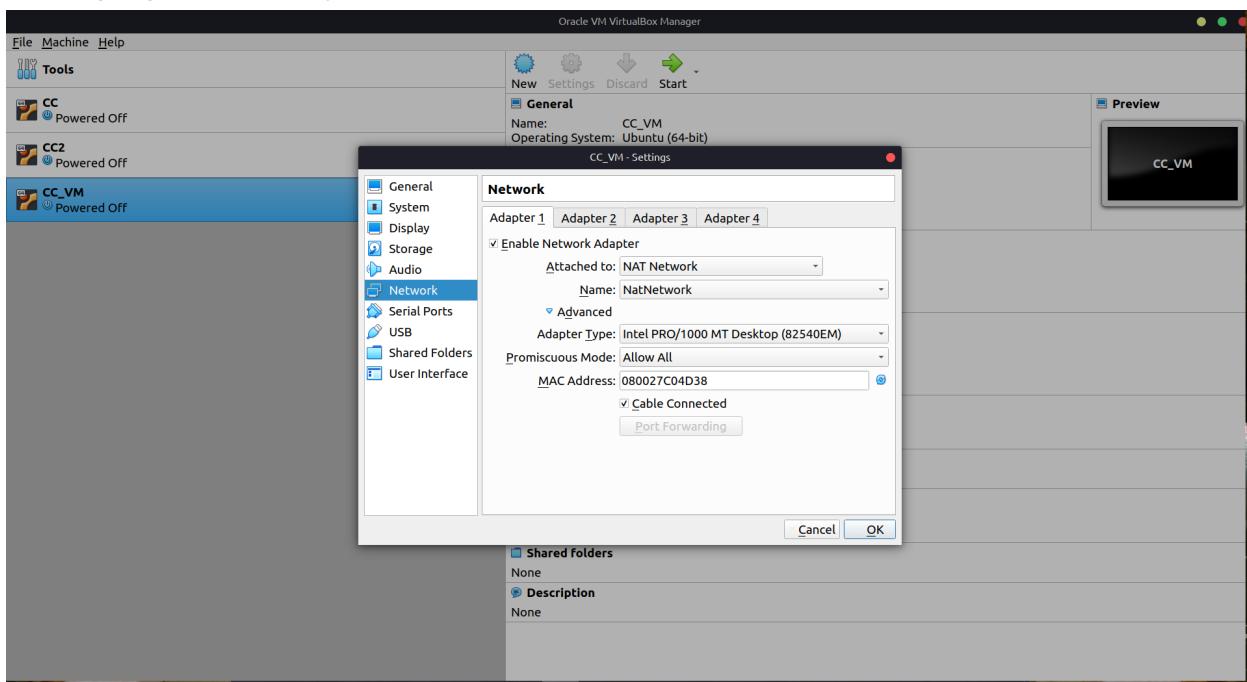
Changing number of processors:-



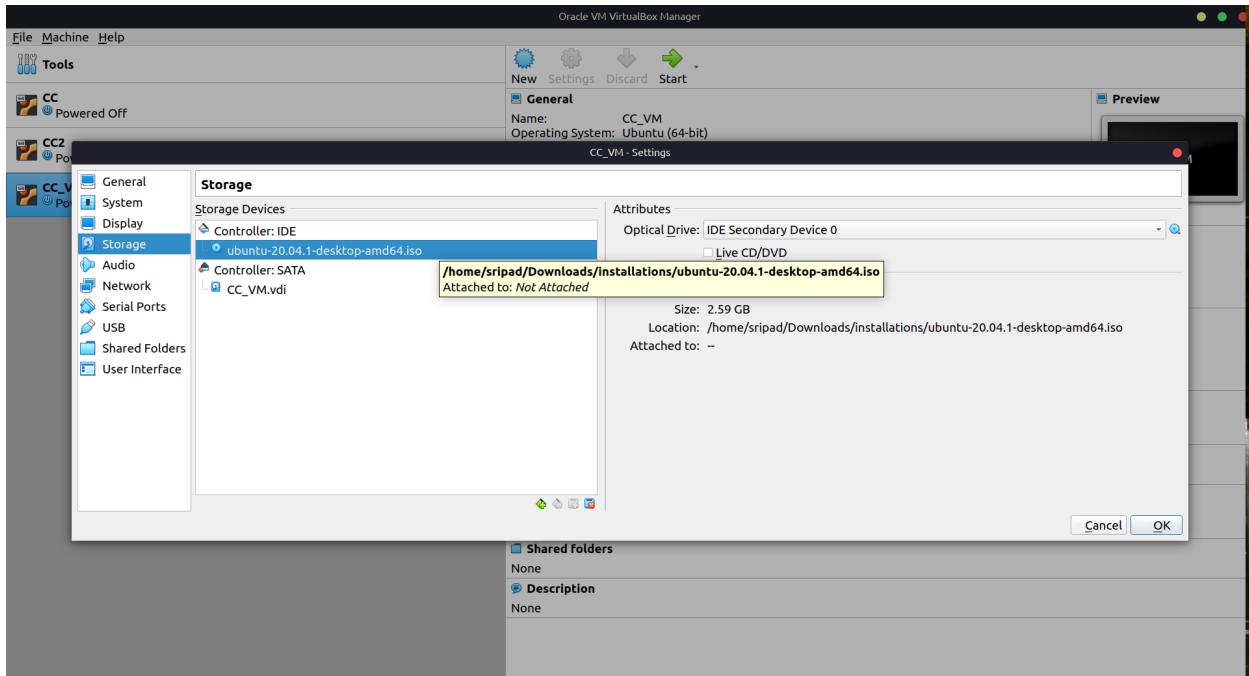
Changing Display settings:-



Changing network type to NAT Network:-



Attaching ISO file :-



After this process, the VM is created successfully in Oracle VBox. When running the VM for the first time, I was prompted to install the Ubuntu OS. After the installation of the OS, I was able to use the VM as a normal machine.

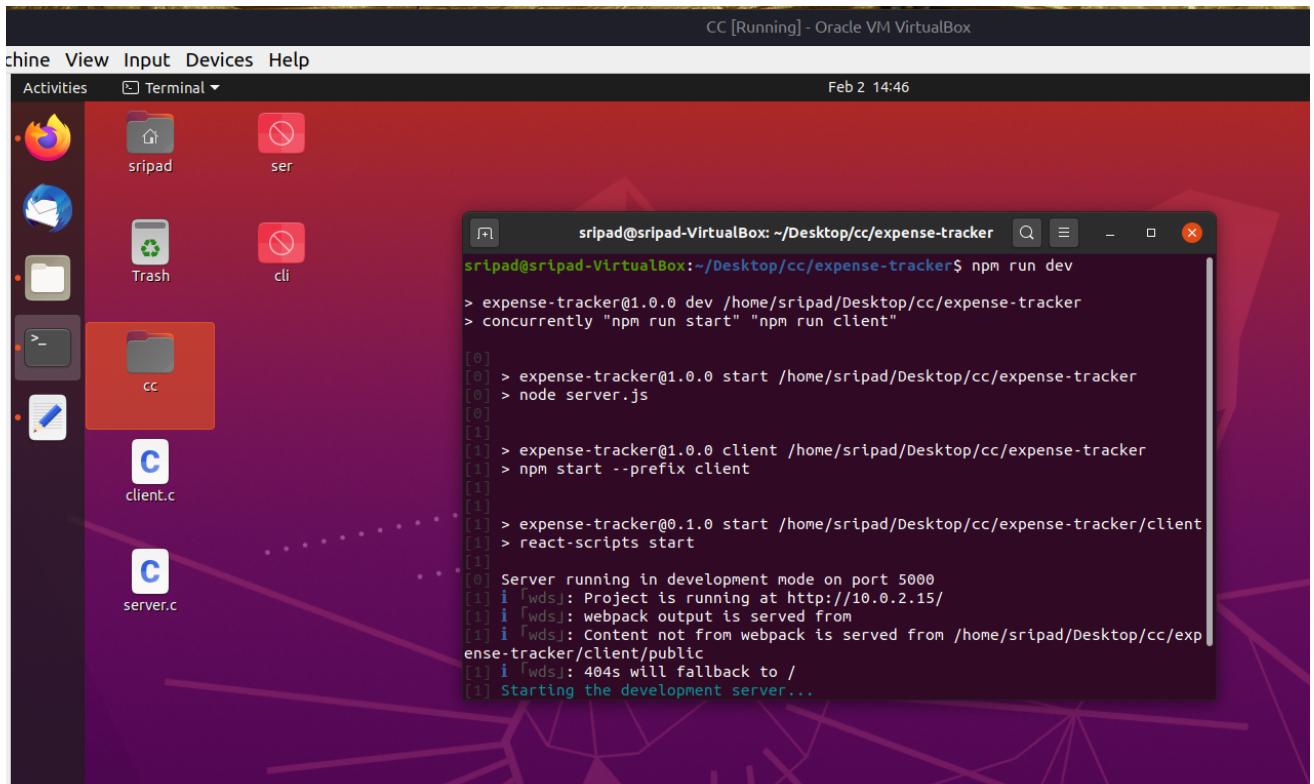
Q2) Run webserver in a created VM, which stores, retrieves and delivers objects in webpages to the clients/users based on requests.

For this question, I have developed a web server using NodeJs (javascript). The application uses MongoDB as database and also uses Google OAuth service for authentication of users. Based on the emailId of the signed-in user, content is delivered. The main aim of the web application is to store and retrieve expenses of the user (which are stored in MongoDB database). I developed the frontend of application using ReactJs (javascript-based library).

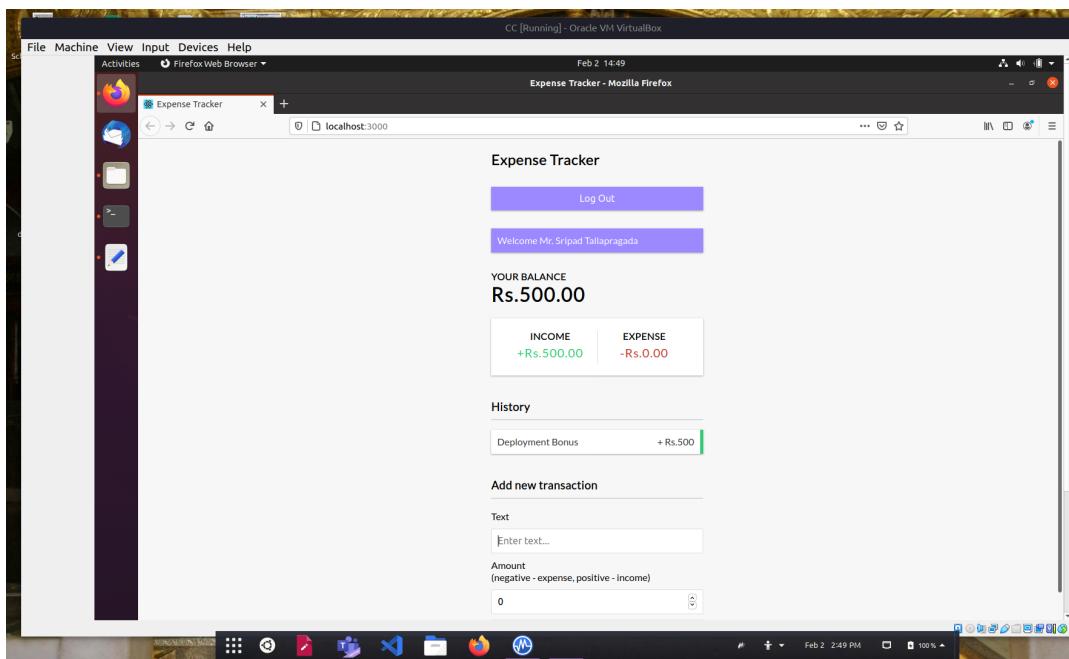
Running server:-

I used the command: "npm run start" to start my web application on PORT 5000 of my VM. I am using a hosted instance of MongoDB as database.

Screenshot of running the application:-



Home page of website:-



Q3) Create two VMs in VBox and establish communication between the created VMs. Use pipe/socket

For this question, I created 2 VMs named CC and CC2 and I established communication between them using Sockets. I wrote 2 programs called “server.c” and “client.c” which take IP address as a parameter and establish Socket communication.

Working of server.c:-

- 1) The server.c file takes IP_Address as parameter and creates a socket connection on PORT 5000.
- 2) Then it keeps listening for connection on the PORT

Working of client.c:-

- 1) The client.c file also takes IP_Address as parameter and tries to connect to PORT 5000 on the given IP.

Once a connection is established, the server will be prompted to send a message to the client. Then the client can respond to the server.

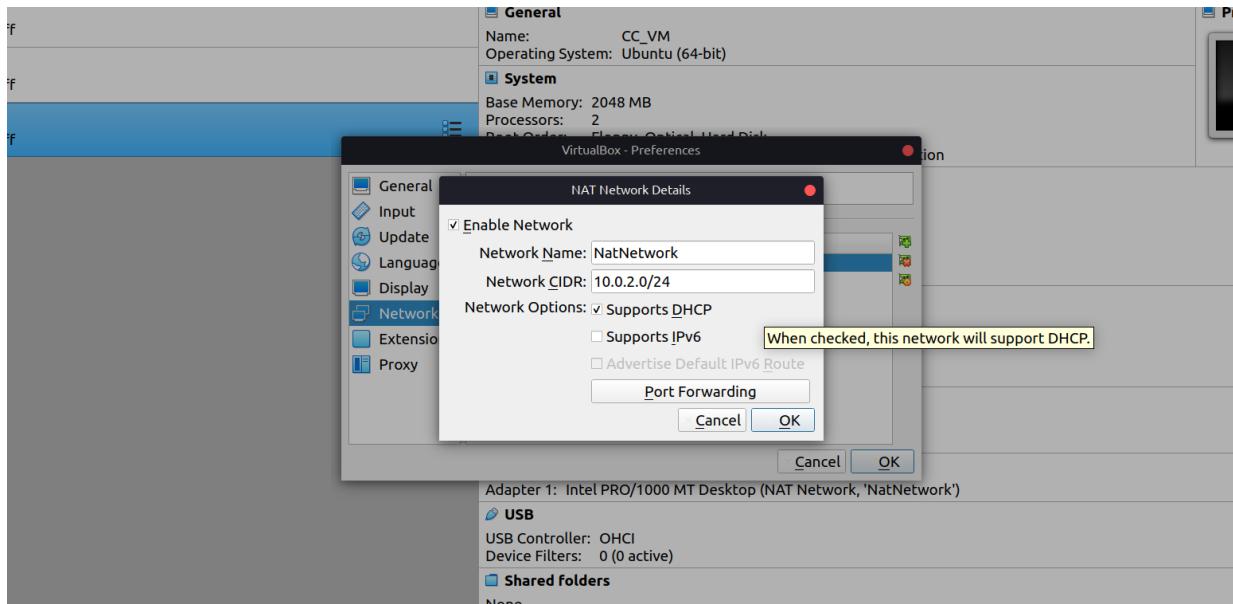
For running these files, first we need to find the IP_address of VMs.

Steps to connect VMs to custom NAT Network:-

A NAT Network is a network in which all the VMs running on a physical machine are connected. All VMs connected to a NAT Network can also access the internet through the network of PM.

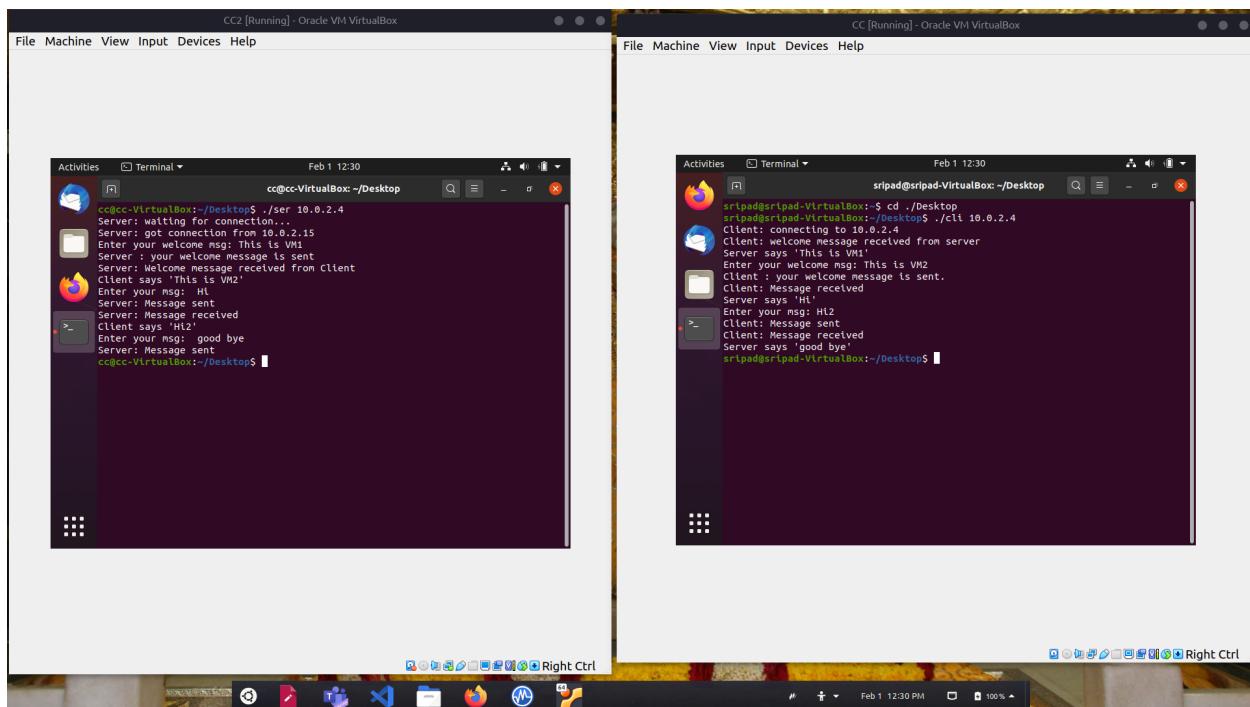
Before VMs were created, I created a custom NAT_Network in Oracle VBox. Then I changed the network settings of VMs so that they will be connected to the custom network. In this network, each VM is assigned a private IP address.

Screenshot of creation of NAT Network:-



Screenshots of Chat between 2 VMs:-

Here the VM on left side is Server (having IP address of 10.0.2.4) and VM on right side is client which is connected to the server (on left VM) using sockets



Finding IP Address of VM1 using `ifconfig` command:-

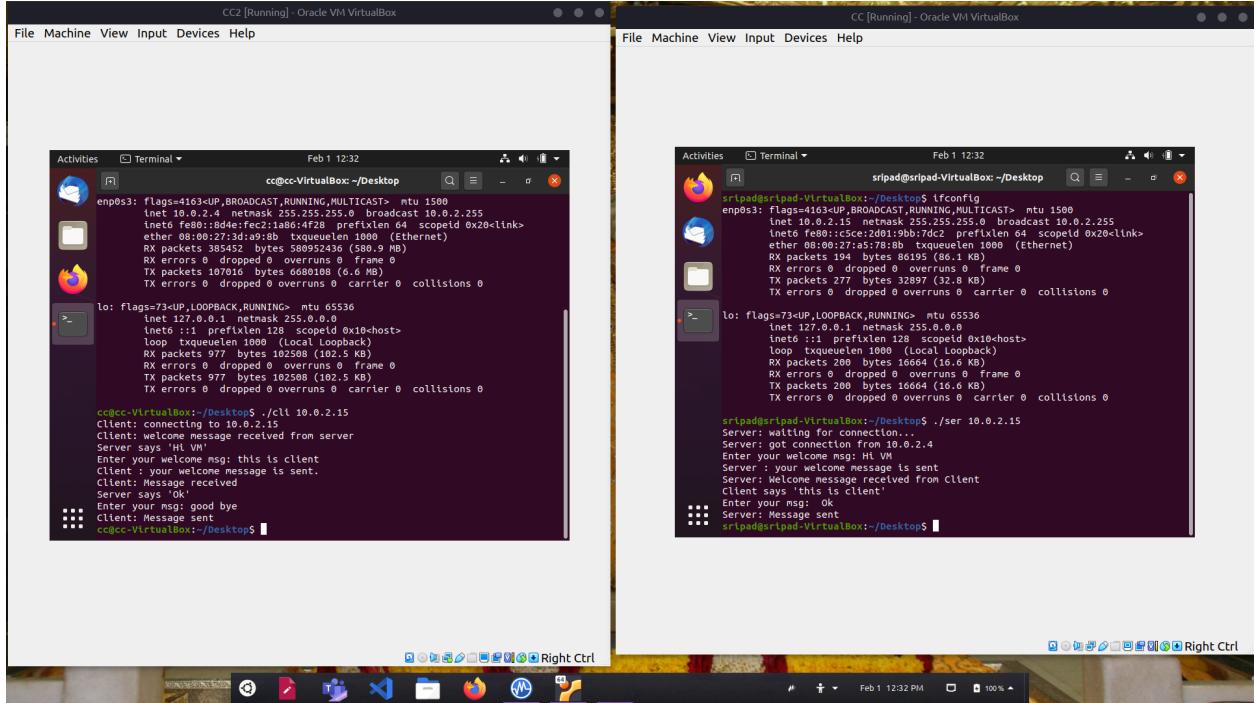
The screenshot shows a Linux desktop environment with a terminal window open in a window titled "CC2 [Running] - Oracle VM VirtualBox". The terminal window has a title bar "File Machine View Input Devices Help" and a status bar "Activities Terminal Feb 1 12:30". Inside the terminal, the following text is displayed:

```
cc@cc-VirtualBox:~/Desktop$ ./ser 10.0.2.4
Server: waiting for connection...
Server: got connection from 10.0.2.15
Enter your welcome msg: This is VM1
Server : your welcome message is sent
Server: Welcome message received from Client
Client says 'This is VM2'
Enter your msg: Hi
Server: Message sent
Server: Message received
Client says 'Hi2'
Enter your msg: good bye
Server: Message sent
cc@cc-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::8d4:fec2:1a86:4f28 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:3d:a9:8b txqueuelen 1000 (Ethernet)
            RX packets 385452 bytes 580952436 (580.9 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 107016 bytes 6680108 (6.6 MB)
            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 977 bytes 102508 (102.5 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 977 bytes 102508 (102.5 KB)
```

The desktop environment includes a dock at the bottom with icons for various applications like a terminal, file manager, browser, and code editor.

Screenshot when VM on right side is Server (IP address: 10.0.2.15) and VM on left side is client:-



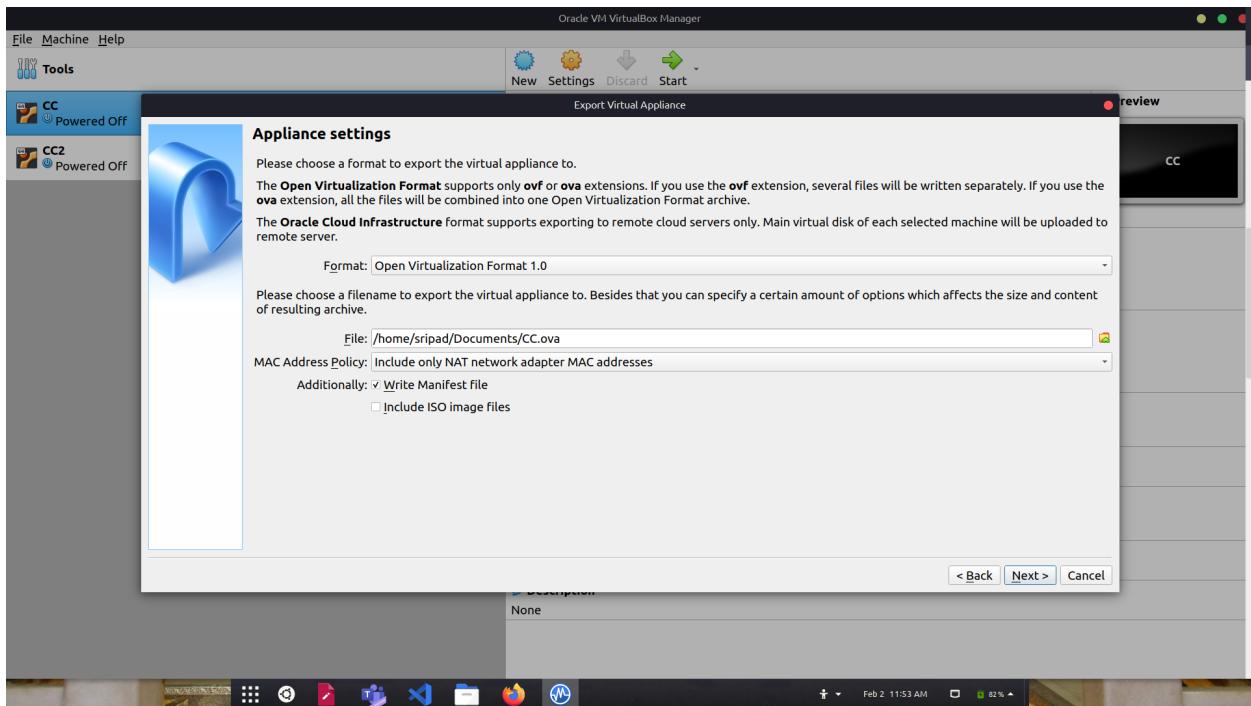
Q4) Test migration service in VBox, VMWare:-

For this question, I migrated my VM from Oracle VBox to VMWare.

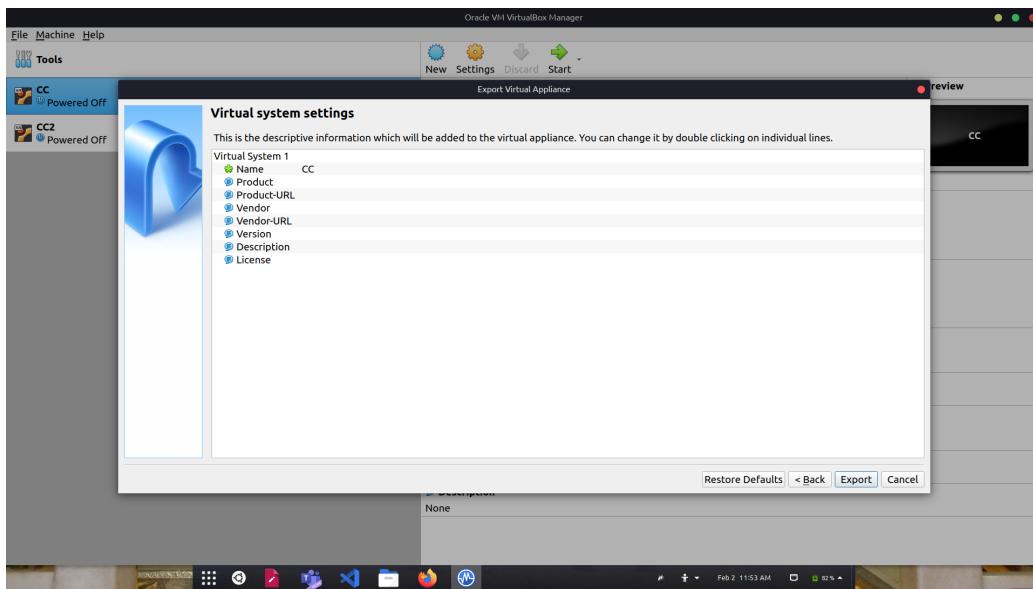
Steps followed for exporting from Oracle VBox:-

1. First I selected 'Export Appliance' from the 'File' option in VBox.
2. Then I selected the VM to export.
3. Next, I selected the format & provided a location to store the .ova file.
4. Finally, I clicked on Export to begin the export process.

Screenshots:-



Final Settings of VM to be exported:-

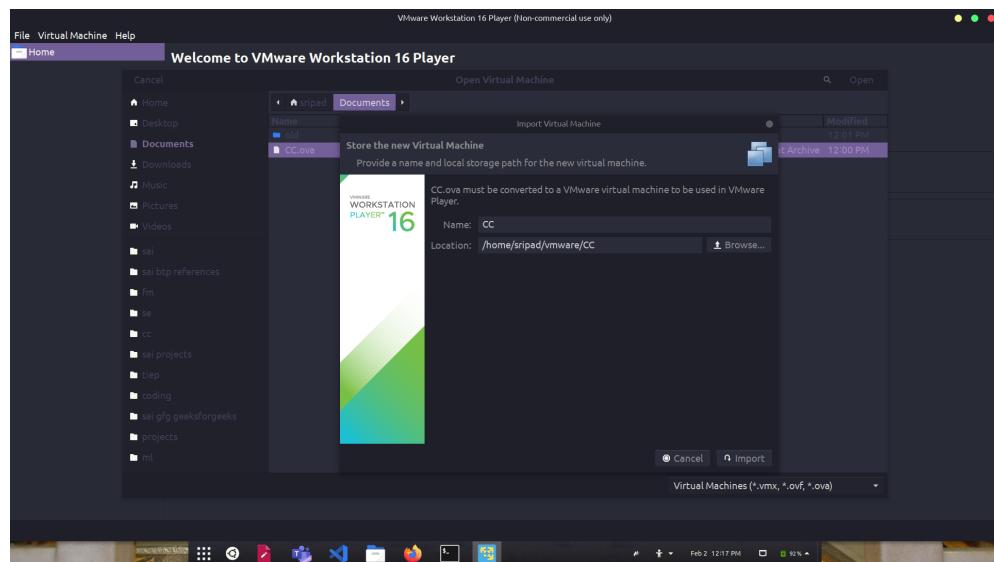


After the exporting process was completed, a new file called as “CC.ova” was created. (CC is the name of exported VM)

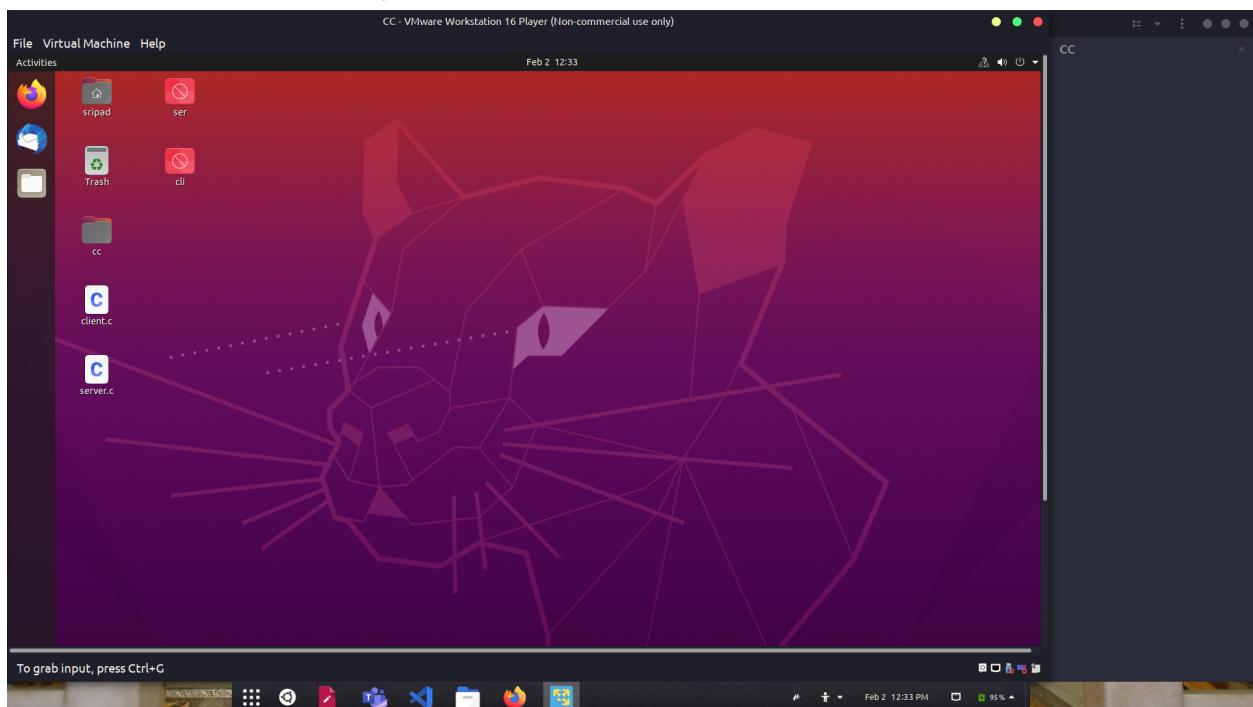
Steps followed for importing the VM in VMWare:-

1. In the VMWare Workstation, I selected the `Open` option from the `File` menu.
2. Then I selected the .ova file which was previously created and finally clicked on Import.
3. Then a VM is created for us to use directly.

Importing VM on VMWare:-



Final Screenshot of Migrated VM in VMWare:-



Here, we can see from the screenshot that the migration process is successfully completed. All the data and files on the VM before migration are also present on the VM after migration. We can even see some files (like server.c, client.c) on the desktop which have been successfully migrated.

Q5) Create an application using Hadoop Map/Reduce to count frequency of each word, number of 1-letter words, 2-letter words,...k-letter words in a file

The files for question are attached in the Q5 folder. The compiled jar file is also attached in the folder (named as “cc.jar”).

The src folder consists of 3 files.

They are:-

- 1) WordCountDriver.java
- 2) WordCountMapper.java
- 3) WordCountReducer.java

The WordCountMapper class has the logic for inserting word, count pairs which are then sorted and combined based on the ‘key’. The WordCountDriver class combines the WordCountMapper class and WordCountReducer class.

Command used to run the generated jar file:-

hadoop jar /home/sripad/Desktop/cc.jar WordCountDriver /test.txt /WordCountResult

Here, cc.jar is the name of jar file which has the hadoop code, test.txt contains the input text and output is written.

The generated output is present in the WordCountResult folder.

Sample text file contents:-

“I am I CC CCCC CCCC”

For this input, output will be:-

1 letter word 2

2 letter word 2

4 letter word 2

CC 1

CCCC 2

12

am 1

Screenshots of execution:-