Team members: 19C010 - Aswath Swasun P

19C092 - Shane Rex S

19C105 - Subash A

19C117 - Vijay G

Gmeet recording: https://drive.google.com/file/d/1BeX-V97SIXiSoaygFVdxLkux4vELooR1/view

Github link: https://github.com/subash2121/cloud_assignment_3

Cloud Assignment 3

a.

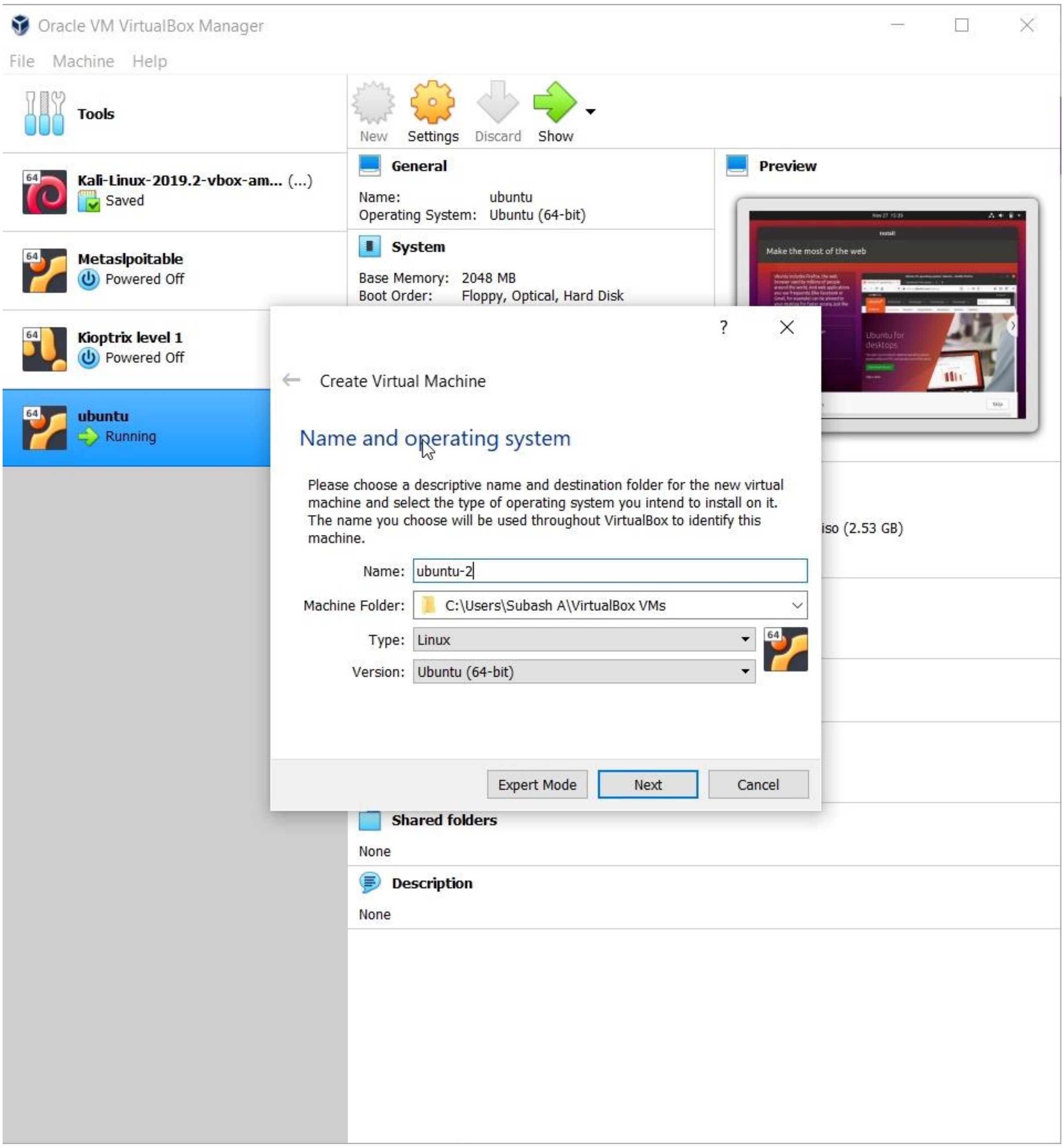
Virtualisation software used: Virtual Box

Virtualisation OS used: Ubuntu

(i) Inter Virtual-machine communication on a single host

Procedure:

1.Create a new Virtual Machine in the Virtual Box



Name: ubuntu
Operating System: Ubuntu (64-bit)

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk

iso (2.53 GB)

Shared folders: None

Description: None

Create Virtual Machine

Name and operating system

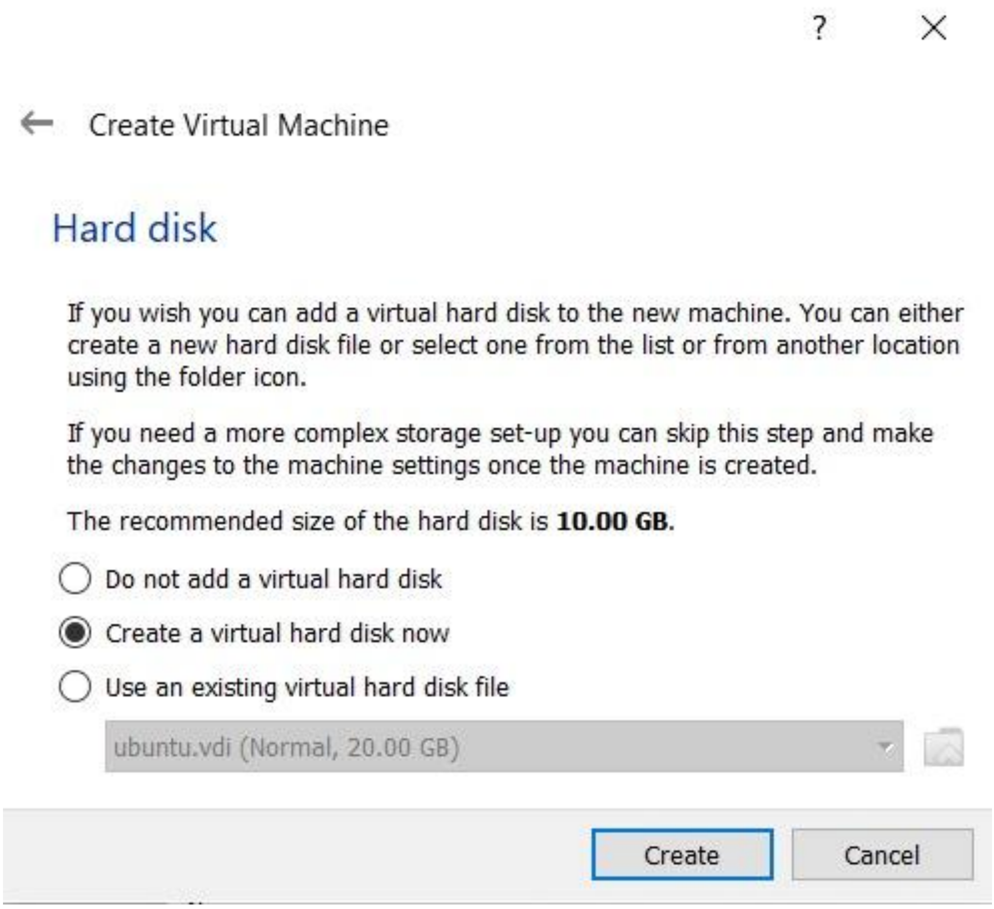
Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: ubuntu-2

Machine Folder: C:\Users\Subash A\VirtualBox VMs

Type: Linux

Version: Ubuntu (64-bit)
]



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

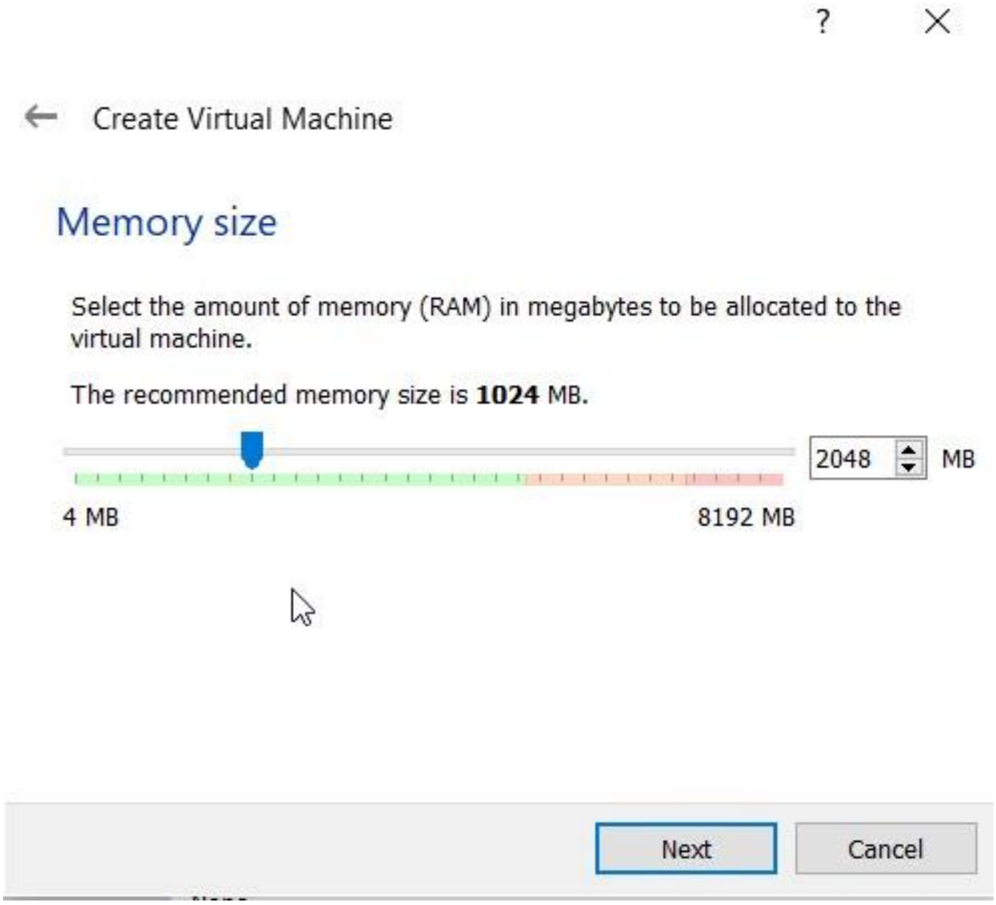
○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

ubuntu.vdi (Normal, 20.00 GB)

2.Allocate the Ram space for the Virtual Machine.



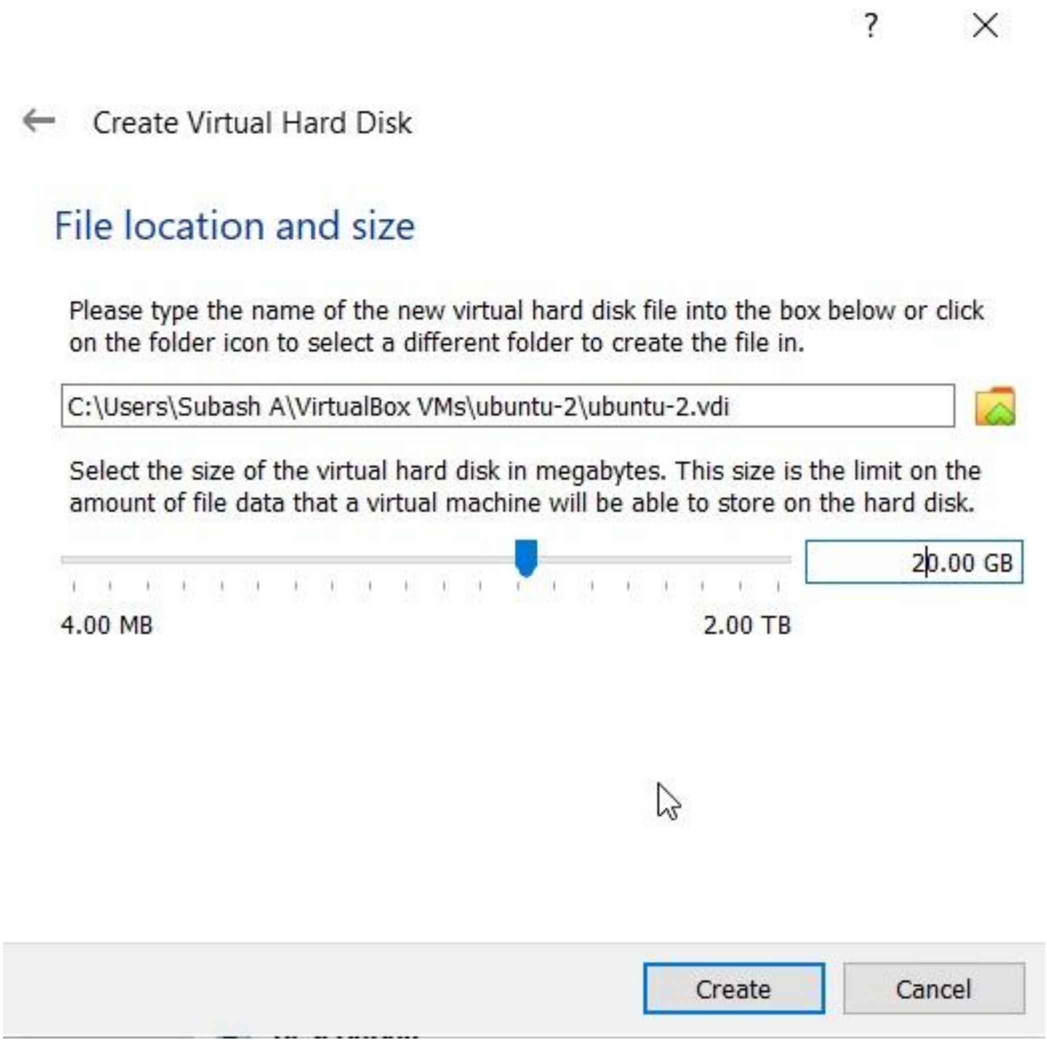
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

2048 MB

3.Select the Folder in which the Virtual Machine is to be created and the space allocated to the Virtual machine



Create Virtual Hard Disk

File location and size

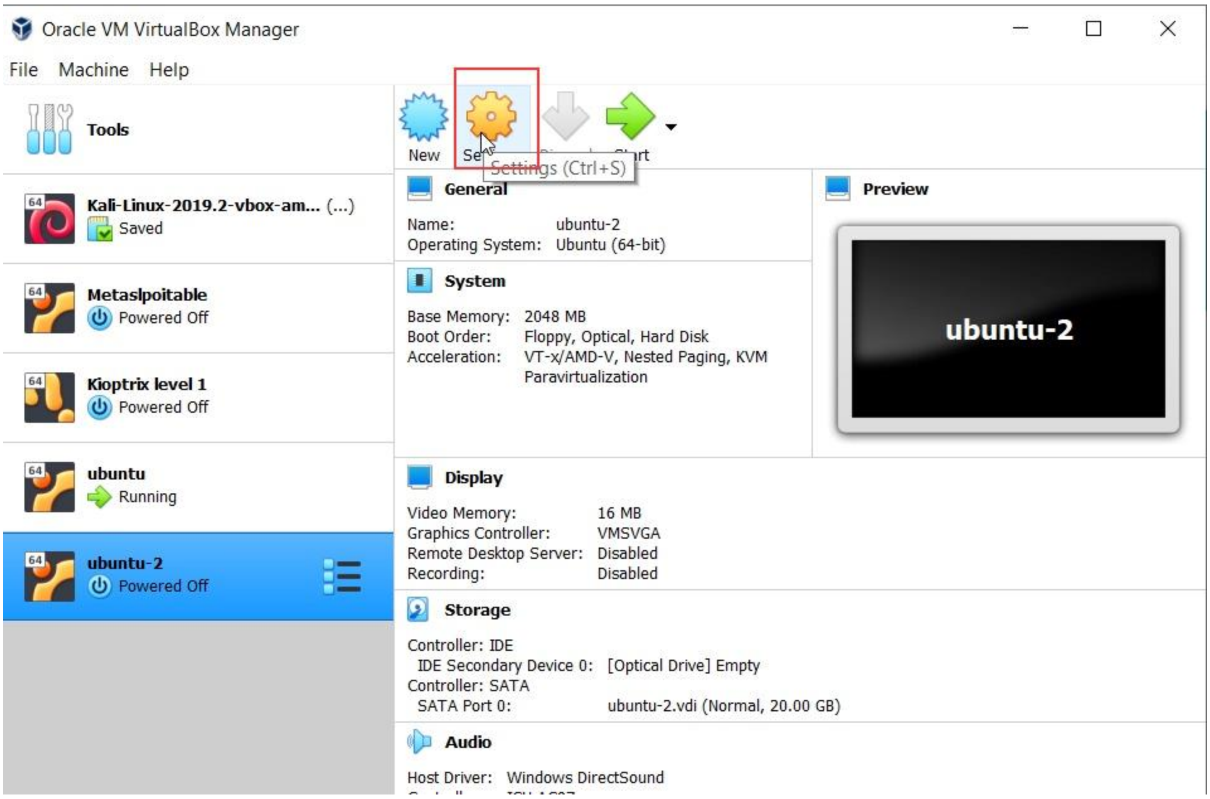
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

C:\Users\Subash A\VirtualBox VMs\ubuntu-2\ubuntu-2.vdi

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

20.00 GB

4.The Virtual Machine is now created and now we load the OS ISO file.



Kali-Linux-2019.2-vbox-am... (...) Saved
Metaslpoitable Powered Off
Kioptrix level 1 Powered Off
ubuntu Running
ubuntu-2 Powered Off

General

```
Name:               ubuntu-2
Operating System:   Ubuntu (64-bit)
```

System

```
Base Memory:   2048 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  VT-x/AMD-V, Nested Paging, KVM
               Paravirtualization
```

Display

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

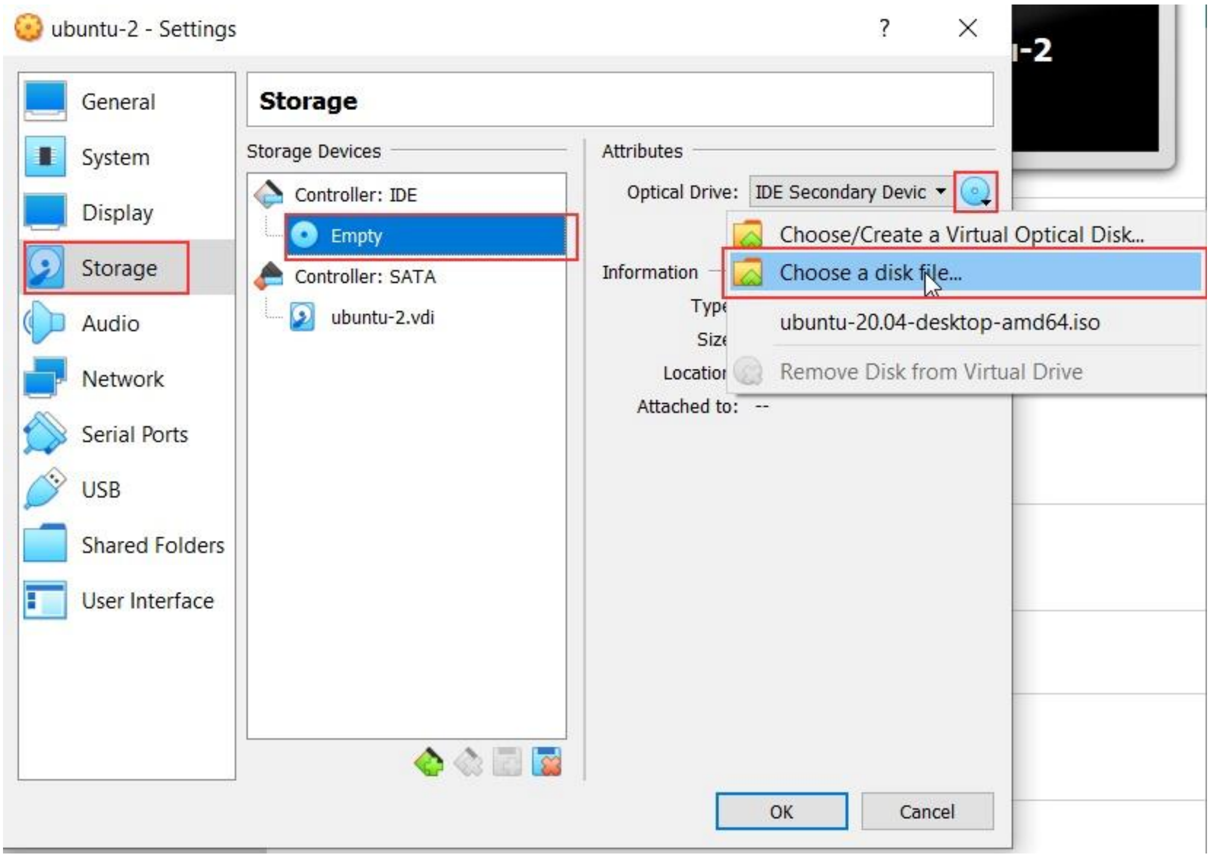
Storage

```
Controller: IDE
  IDE Secondary Device 0:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:             ubuntu-2.vdi (Normal, 20.00 GB)
```

Audio

```
Host Driver:  Windows DirectSound
```

5.Navigate and Select the Ubuntu ISO file.



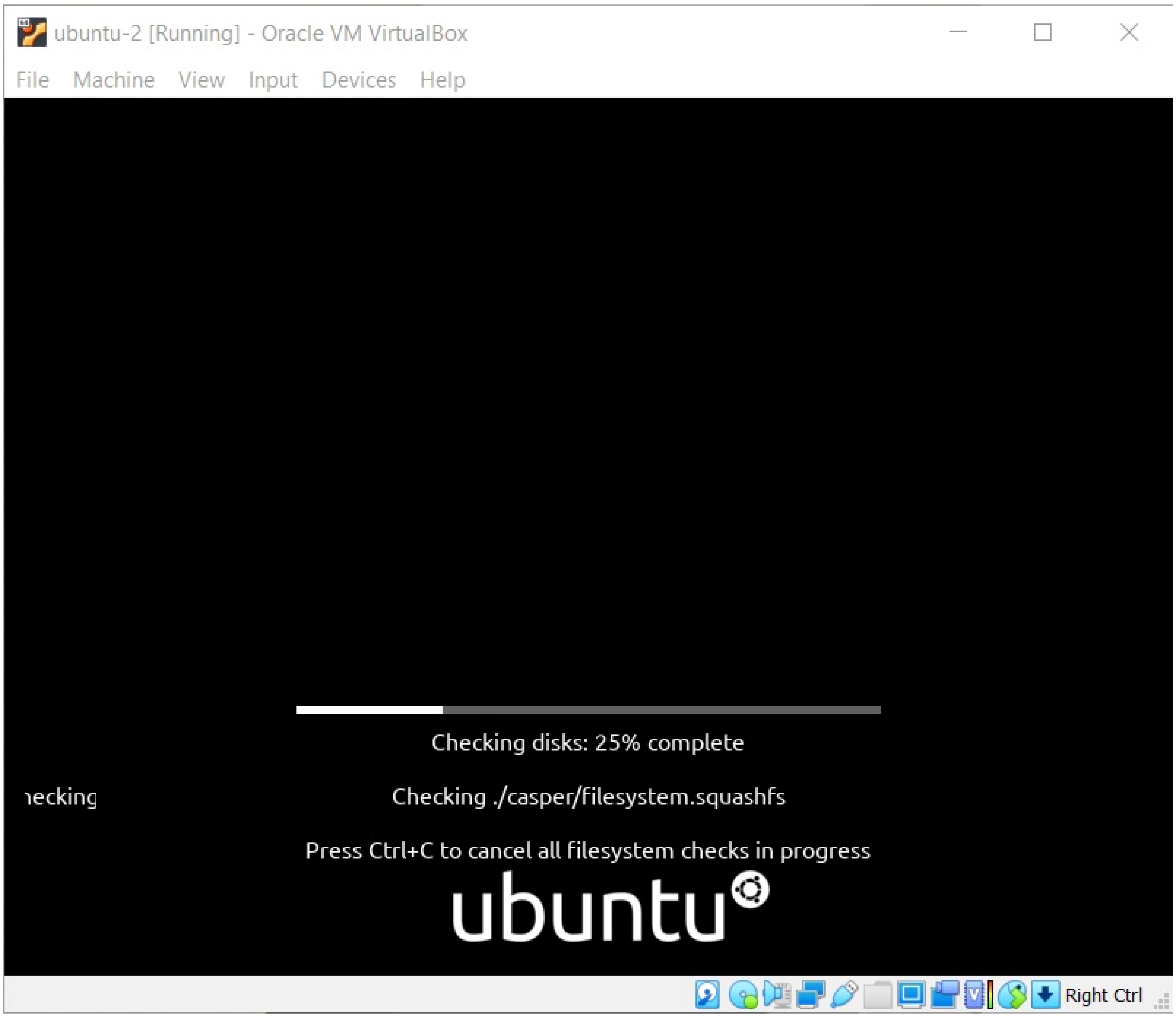
Controller: IDE
Empty
Controller: SATA
ubuntu-2.vdi

Optical Drive: IDE Secondary Devic

Choose/Create a Virtual Optical Disk...
Choose a disk file...
ubuntu-20.04-desktop-amd64.iso
Remove Disk from Virtual Drive

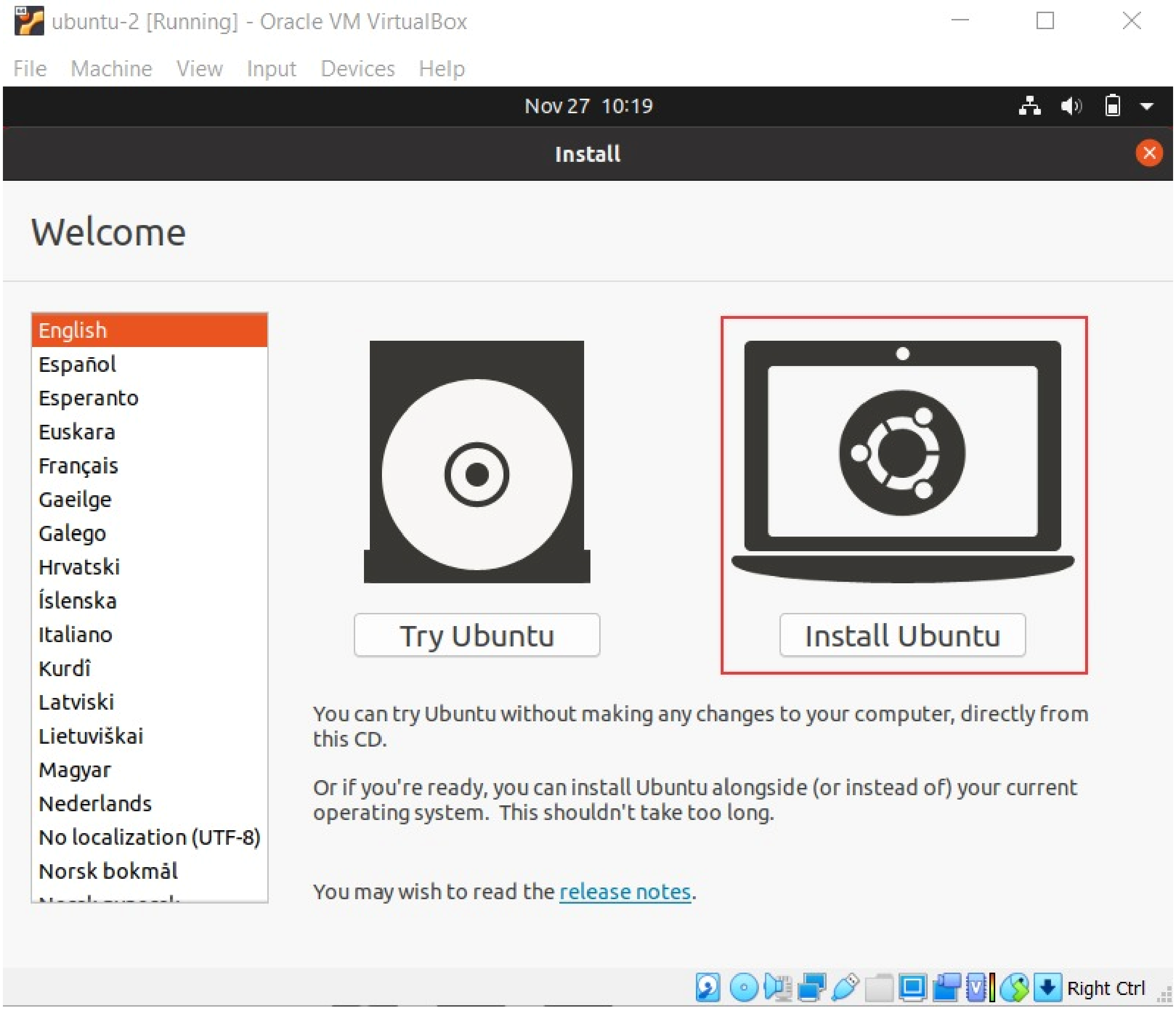
Type
Size
Location
Attached to: --

6.Wait for the machine to boot.



```
Checking disks: 25% complete

ıecking                Checking ./casper/filesystem.squashfs

Press Ctrl+C to cancel all filesystem checks in progress
```

7.Select the way in which you want to run the OS

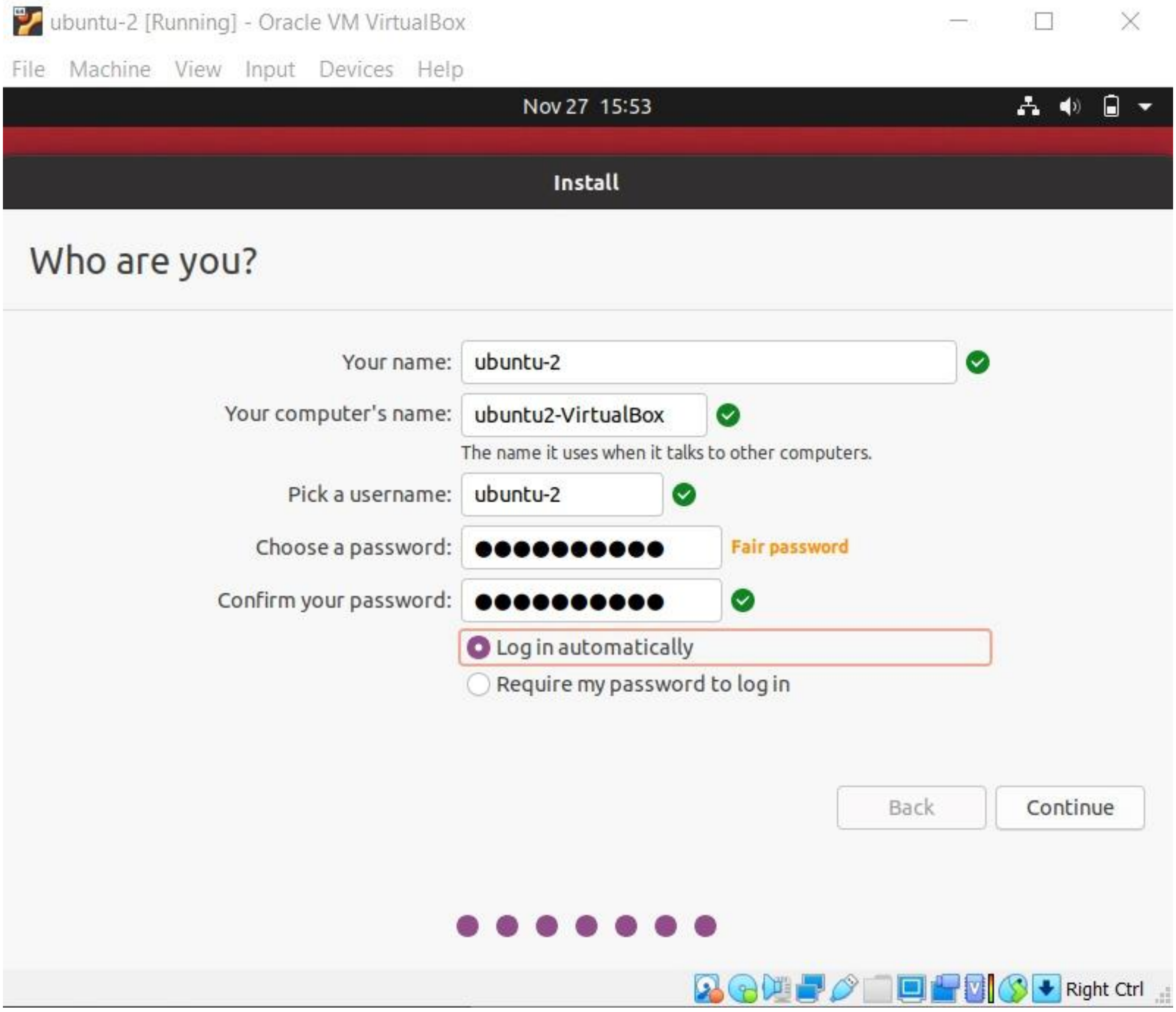


You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

8.Go on with the instructions



ubuntu-2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Nov 27 15:53
Install

Who are you?

Your name: ubuntu-2

Your computer's name: ubuntu2-VirtualBox

The name it uses when it talks to other computers.

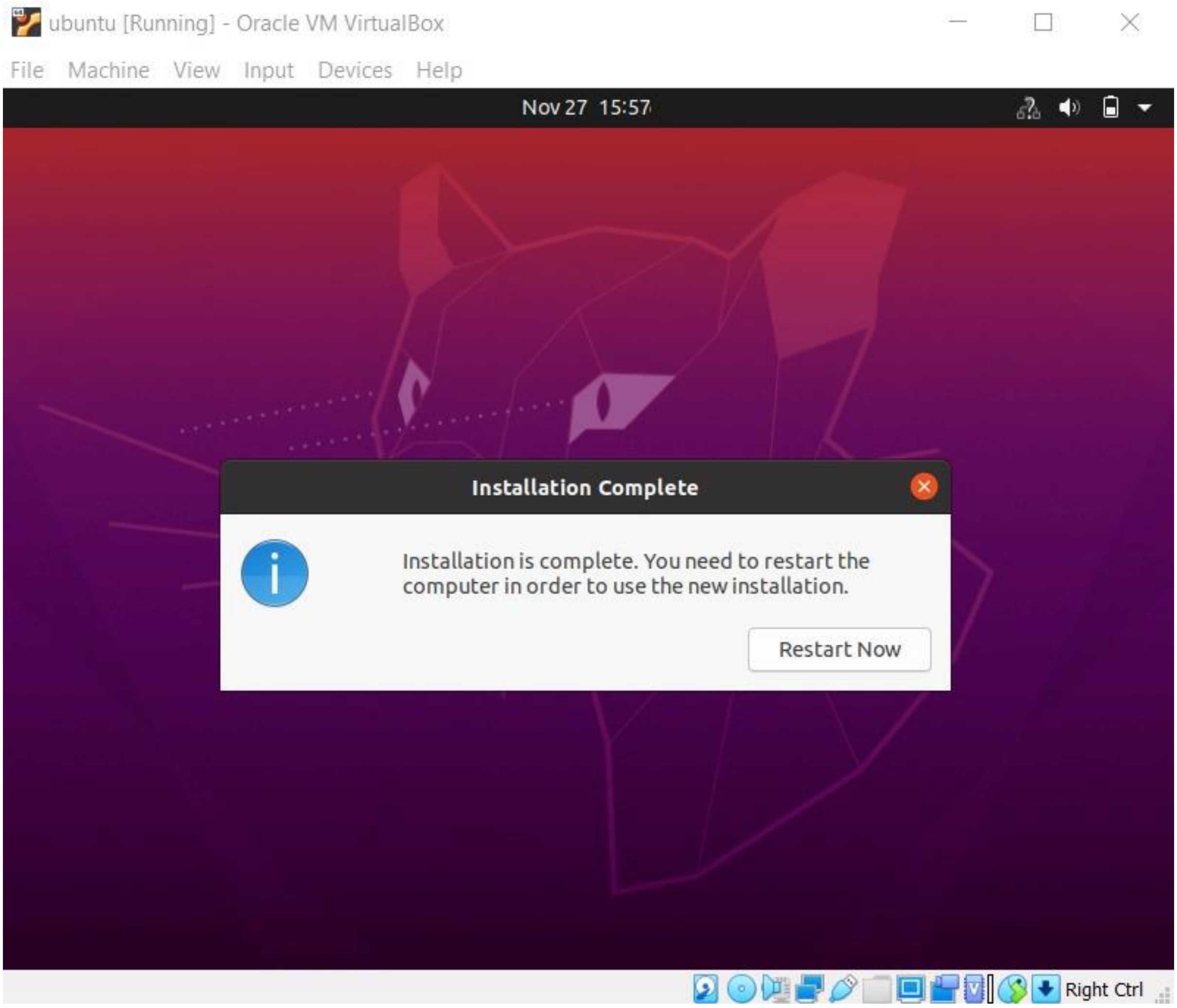
Pick a username: ubuntu-2

Choose a password: ●●●●●●●●●●● Fair password

Confirm your password: ●●●●●●●●●●●

● Log in automatically

○ Require my password to log in



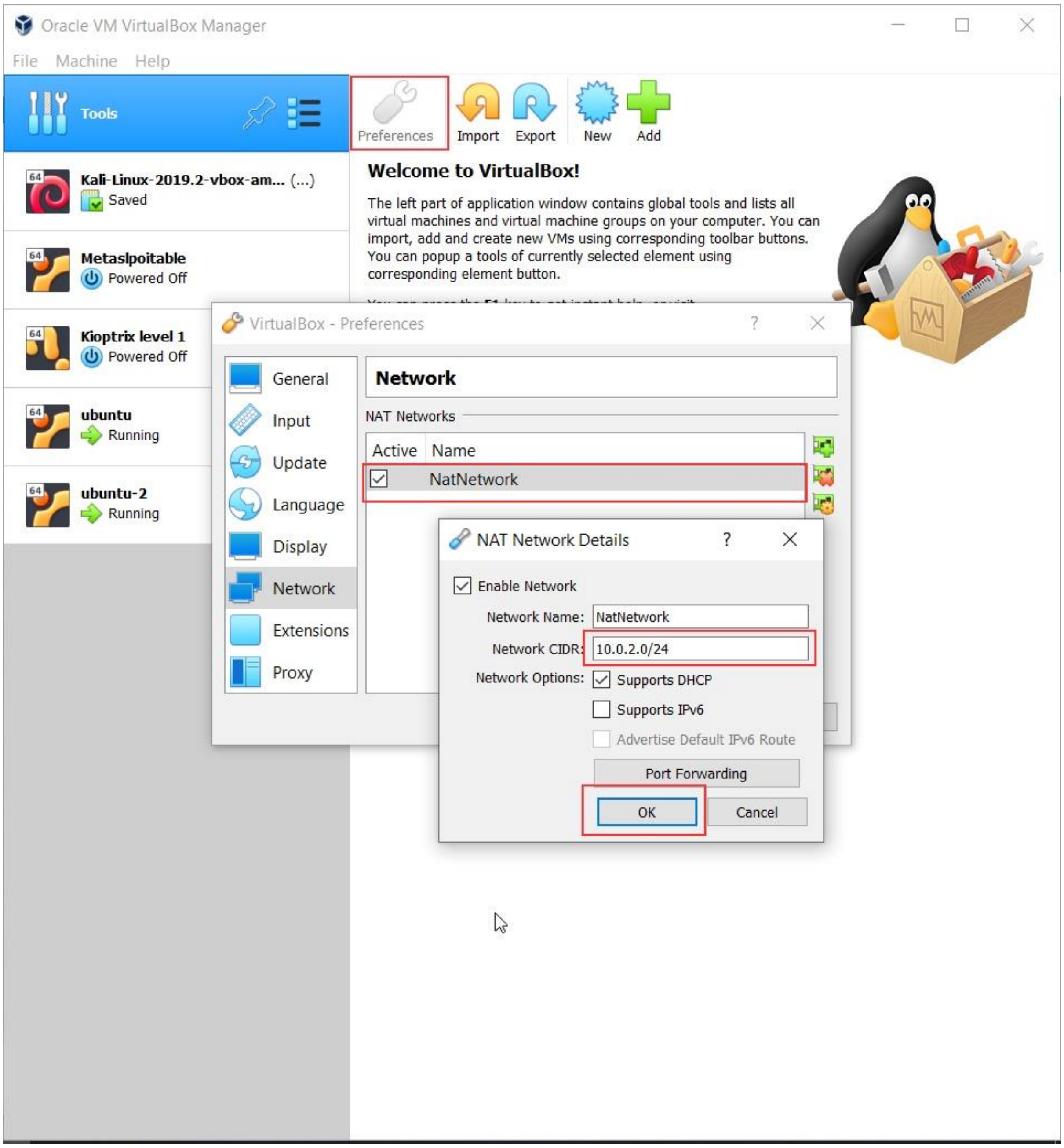
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Nov 27 15:57
Right Ctrl

Installation Complete

Installation is complete. You need to restart the computer in order to use the new installation.

Perform the steps to create a virtual machine inorder to create another machine to perform the Inter Virtual Machine communication.

9.Configure the network inside the vitual box by creating a custom NAT network.



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

Network

NAT Networks

Active	Name
☑	NatNetwork

☑ Enable Network

Network Name: NatNetwork

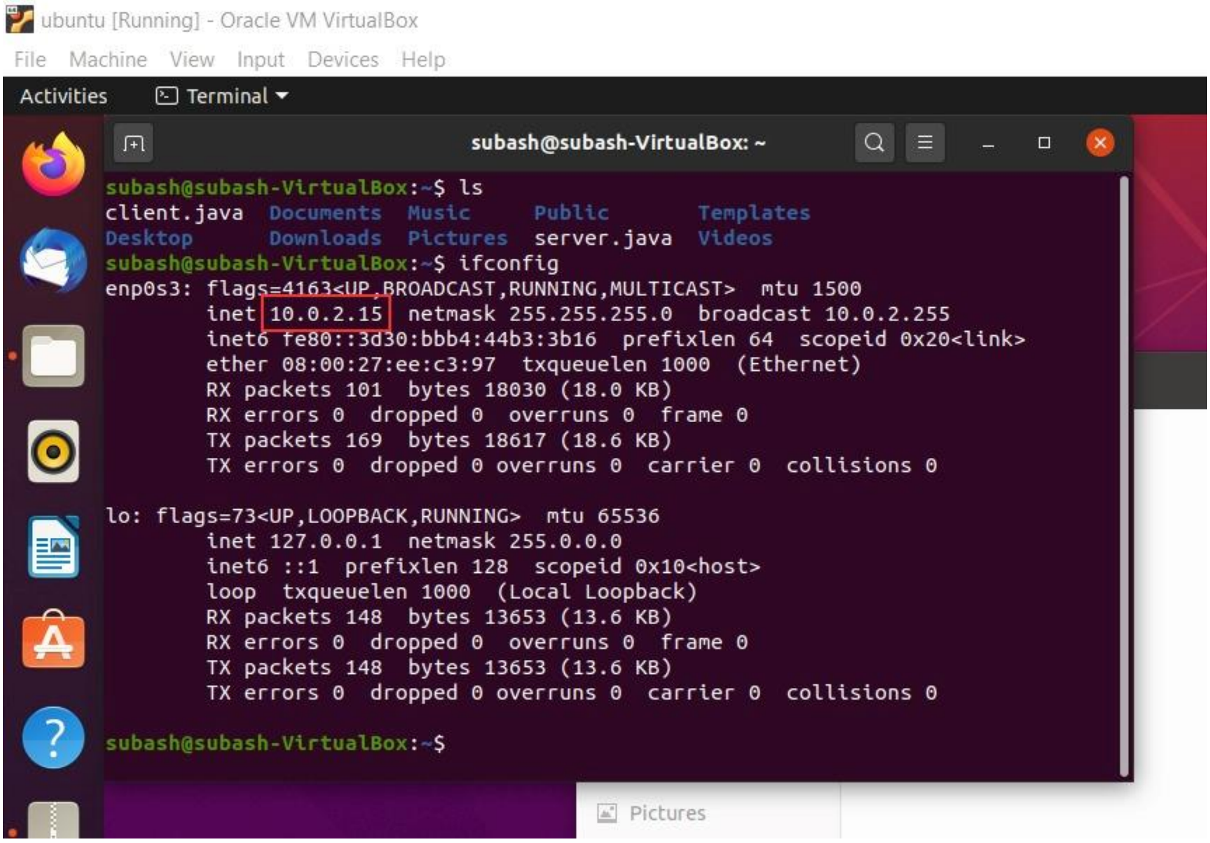
Network CIDR: 10.0.2.0/24

Network Options: ☑ Supports DHCP

□ Supports IPv6

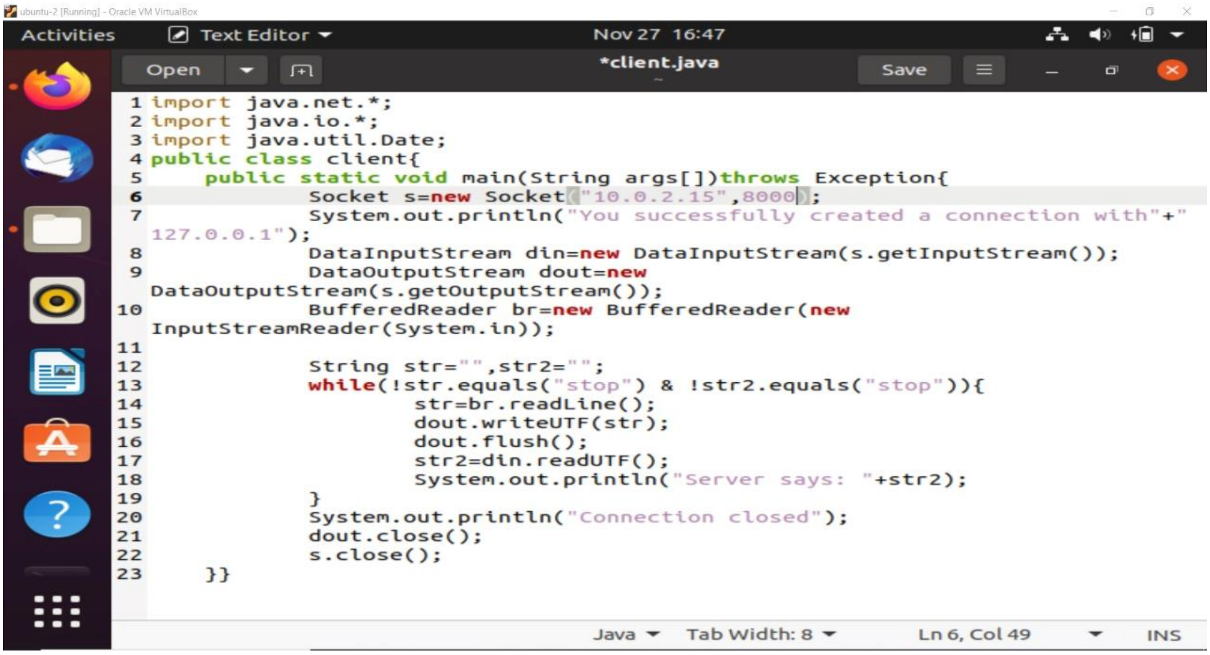
□ Advertise Default IPv6 Route

10.Find the IP address of the server machine


ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
subash@subash-VirtualBox: ~
Pictures

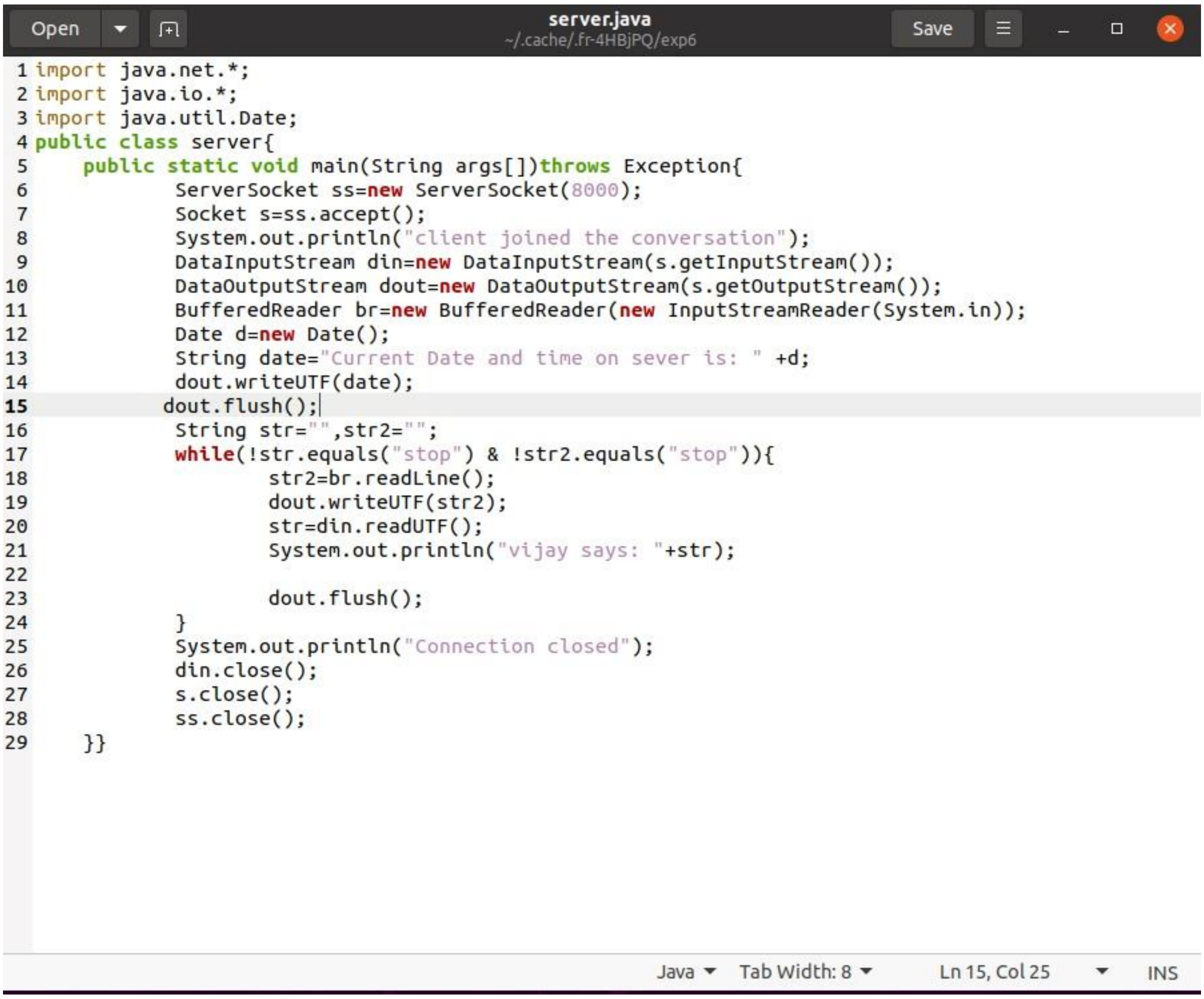
```
subash@subash-VirtualBox:~$ ls
client.java  Documents  Music     Public       Templates
Desktop      Downloads  Pictures  server.java  Videos
subash@subash-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::3d30:bbb4:44b3:3b16  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ee:c3:97  txqueuelen 1000  (Ethernet)
        RX packets 101  bytes 18030 (18.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 169  bytes 18617 (18.6 KB)
        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 148  bytes 13653 (13.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 148  bytes 13653 (13.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

subash@subash-VirtualBox:~$
```

11.Use the IP address of the server machine in client program which is run in the client Machine



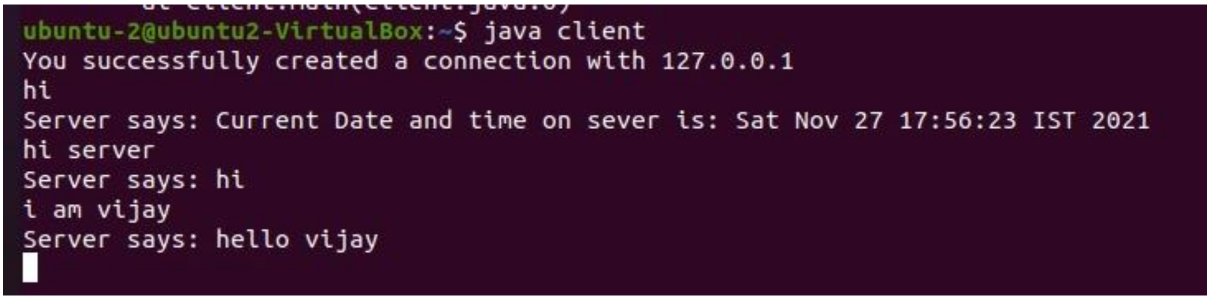
```
import java.net.*;
import java.io.*;
import java.util.Date;
public class client{
    public static void main(String args[])throws Exception{
            Socket s=new Socket("10.0.2.15",8000);
            System.out.println("You successfully created a connection with"+"
127.0.0.1");
            DataInputStream din=new DataInputStream(s.getInputStream());
            DataOutputStream dout=new
DataOutputStream(s.getOutputStream());
            BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

            String str="",str2="";
            while(!str.equals("stop") & !str2.equals("stop")){
                    str=br.readLine();
                    dout.writeUTF(str);
                    dout.flush();
                    str2=din.readUTF();
                    System.out.println("Server says: "+str2);
            }
            System.out.println("Connection closed");
            dout.close();
            s.close();
    }}
```

12.Configure the server program in the server machine by creating a server socket

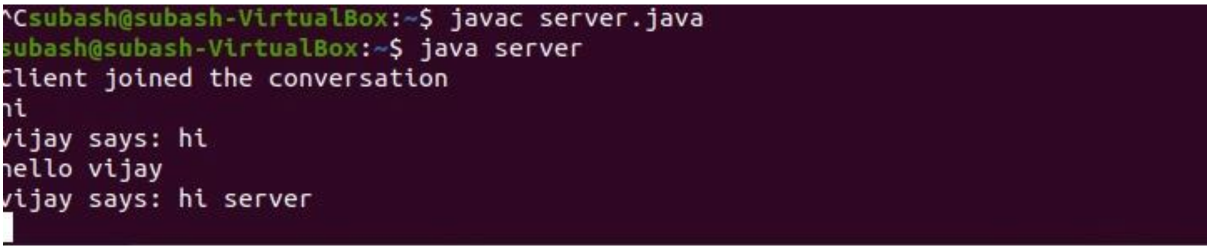


```
import java.net.*;
import java.io.*;
import java.util.Date;
public class server{
    public static void main(String args[])throws Exception{
            ServerSocket ss=new ServerSocket(8000);
            Socket s=ss.accept();
            System.out.println("client joined the conversation");
            DataInputStream din=new DataInputStream(s.getInputStream());
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            Date d=new Date();
            String date="Current Date and time on sever is: " +d;
            dout.writeUTF(date);
           dout.flush();
            String str="",str2="";
            while(!str.equals("stop") & !str2.equals("stop")){
                    str2=br.readLine();
                    dout.writeUTF(str2);
                    str=din.readUTF();
                    System.out.println("vijay says: "+str);

                    dout.flush();
            }
            System.out.println("Connection closed");
            din.close();
            s.close();
            ss.close();
    }}
```

13.Compile and run the code in both machines for communication between both machines



```
      at client.main(client.java:6)
ubuntu-2@ubuntu2-VirtualBox:~$ java client
You successfully created a connection with 127.0.0.1
hi
Server says: Current Date and time on sever is: Sat Nov 27 17:56:23 IST 2021
hi server
Server says: hi
i am vijay
Server says: hello vijay
```

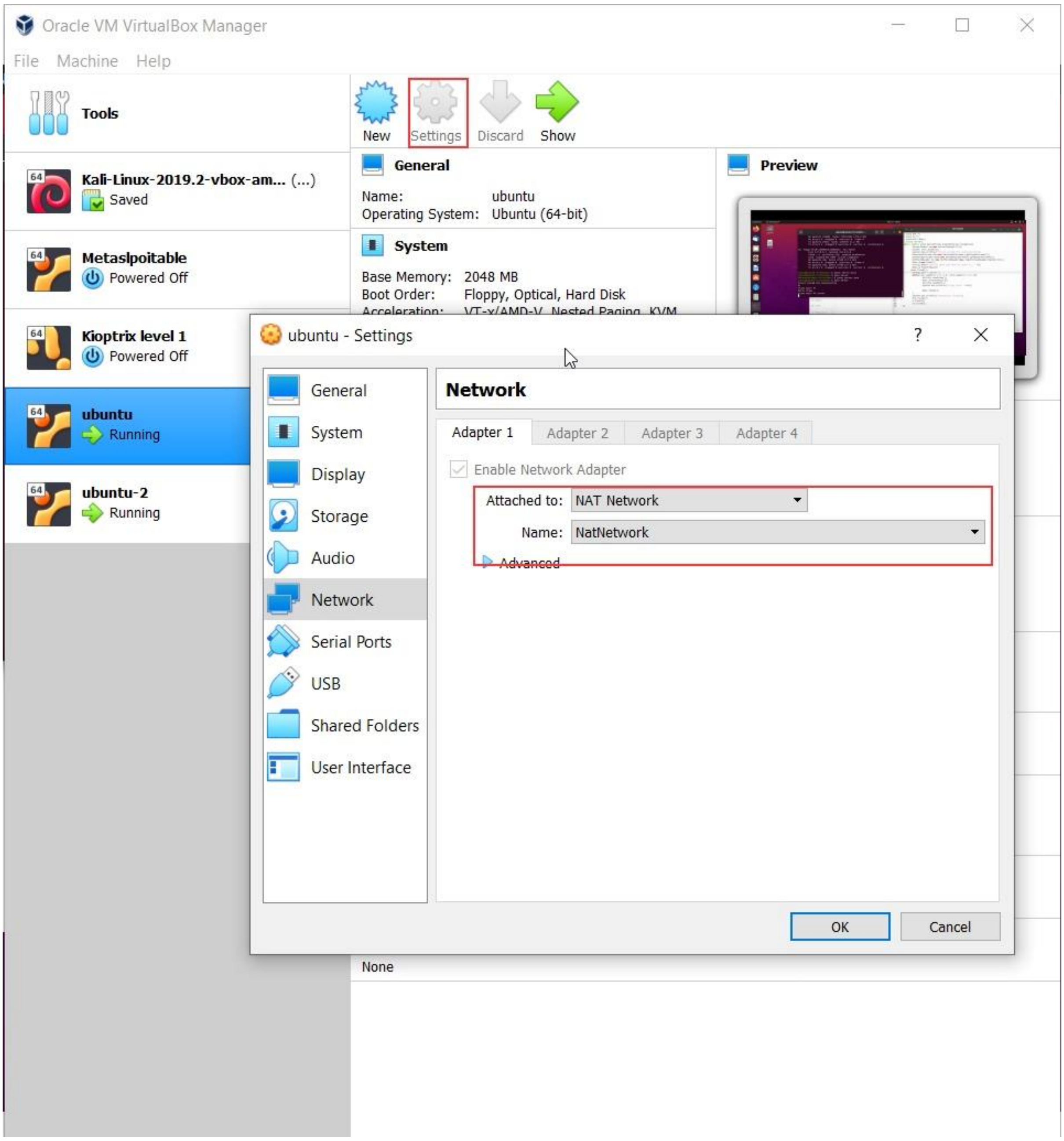


```
^Csubash@subash-VirtualBox:~$ javac server.java
subash@subash-VirtualBox:~$ java server
client joined the conversation
hi
vijay says: hi
hello vijay
vijay says: hi server
```

Experiment 2

Communication between the host machine and the virtual machine in the same computer

1.Configure the Nat Network


Oracle VM VirtualBox Manager
File Machine Help
Tools
New Settings Discard Show
Kali-Linux-2019.2-vbox-am... (...) Saved
Metaslpoitable Powered Off
Kioptrix level 1 Powered Off
ubuntu Running
ubuntu-2 Running
Preview
ubuntu - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Adapter 1 Adapter 2 Adapter 3 Adapter 4
Advanced
OK Cancel
None

General

Name: ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM

Network

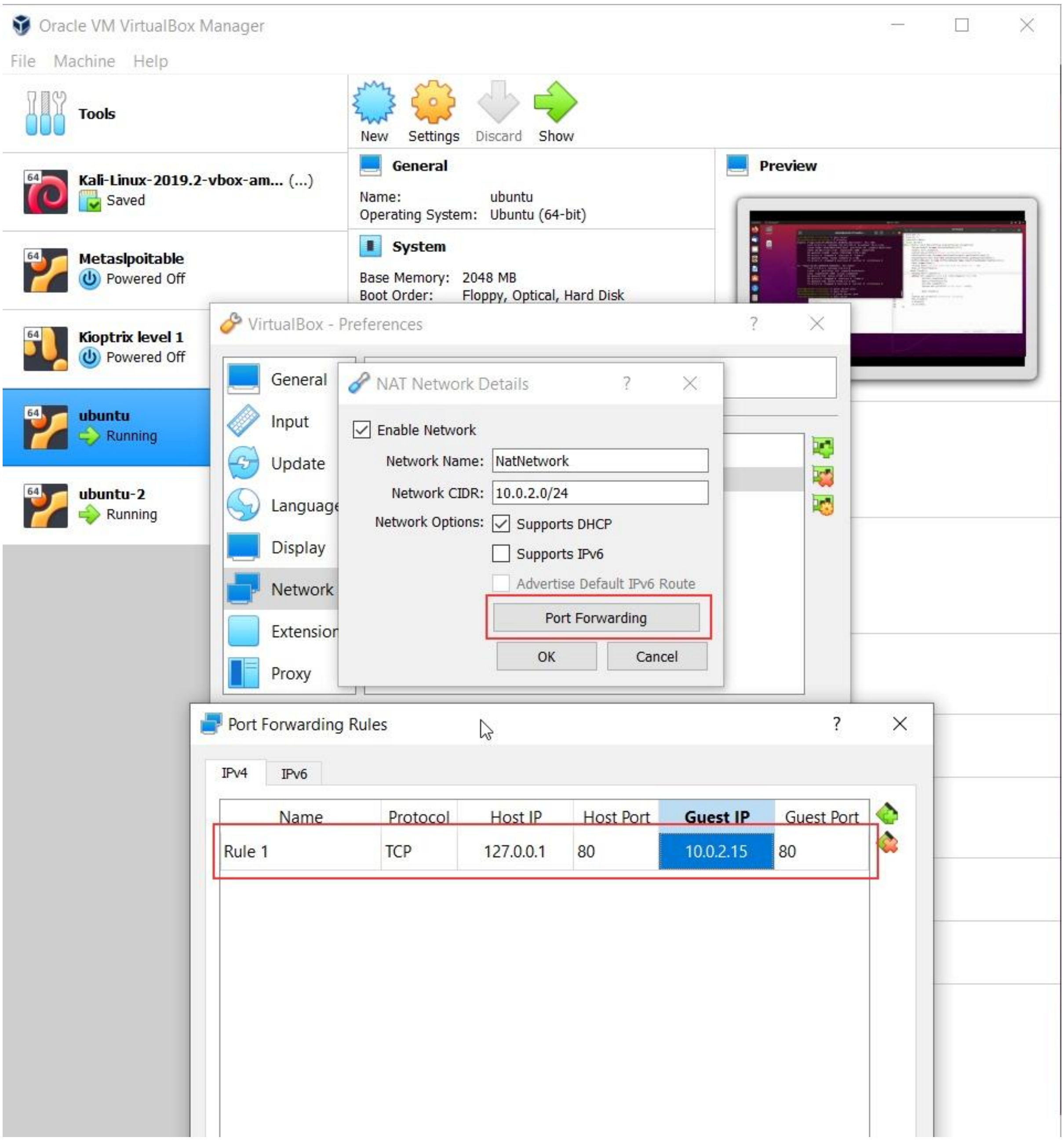
Adapter 1

☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

2.Do Port Forwarding to enable the communication between the machines



Kali-Linux-2019.2-vbox-am... (...) Saved

Metaslpoitable Powered Off

Kioptrix level 1 Powered Off

ubuntu Running

ubuntu-2 Running

General

Name: ubuntu

Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB

Boot Order: Floppy, Optical, Hard Disk

☑ Enable Network

Network Name: NatNetwork

Network CIDR: 10.0.2.0/24

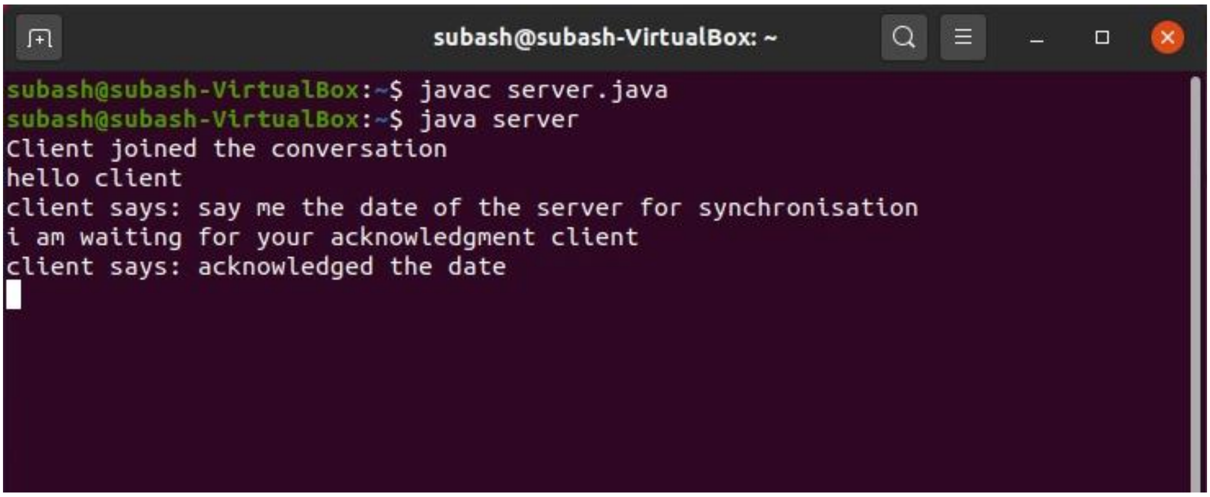
Network Options: ☑ Supports DHCP

□ Supports IPv6

□ Advertise Default IPv6 Route

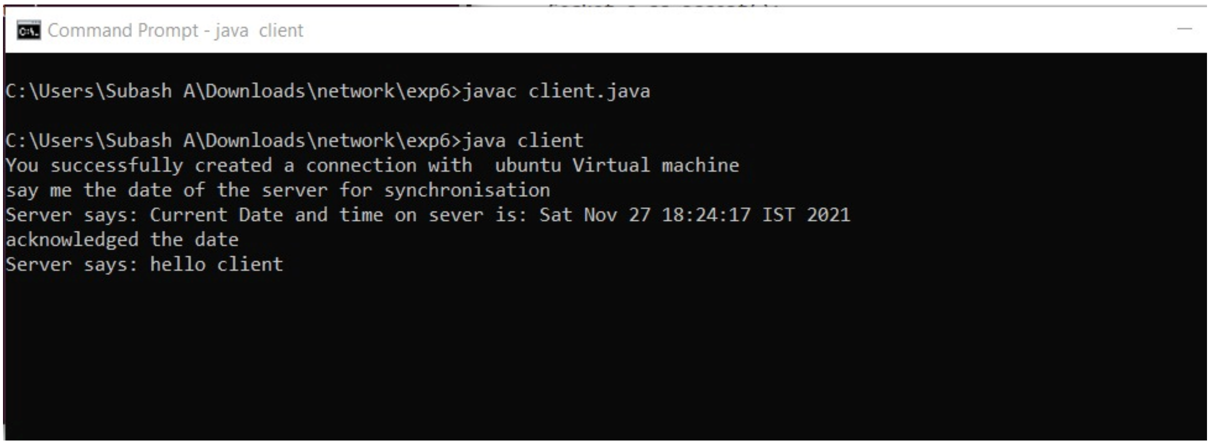
Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
Rule 1	TCP	127.0.0.1	80	10.0.2.15	80

3.Create the server socket and run the server in the virtual machine



```
subash@subash-VirtualBox:~$ javac server.java
subash@subash-VirtualBox:~$ java server
Client joined the conversation
hello client
client says: say me the date of the server for synchronisation
i am waiting for your acknowledgment client
client says: acknowledged the date
```

4.Open terminal in the host machine and connect the host with the virtual machine using the server machine's IP address and run the the client program.After the connection has been established you can perform the chat



```
C:\Users\Subash A\Downloads\network\exp6>javac client.java

C:\Users\Subash A\Downloads\network\exp6>java client
You successfully created a connection with  ubuntu Virtual machine
say me the date of the server for synchronisation
Server says: Current Date and time on sever is: Sat Nov 27 18:24:17 IST 2021
acknowledged the date
Server says: hello client
```

Experiment 3:

Communication between two virtual machines on different Hosts:

Proecdure:

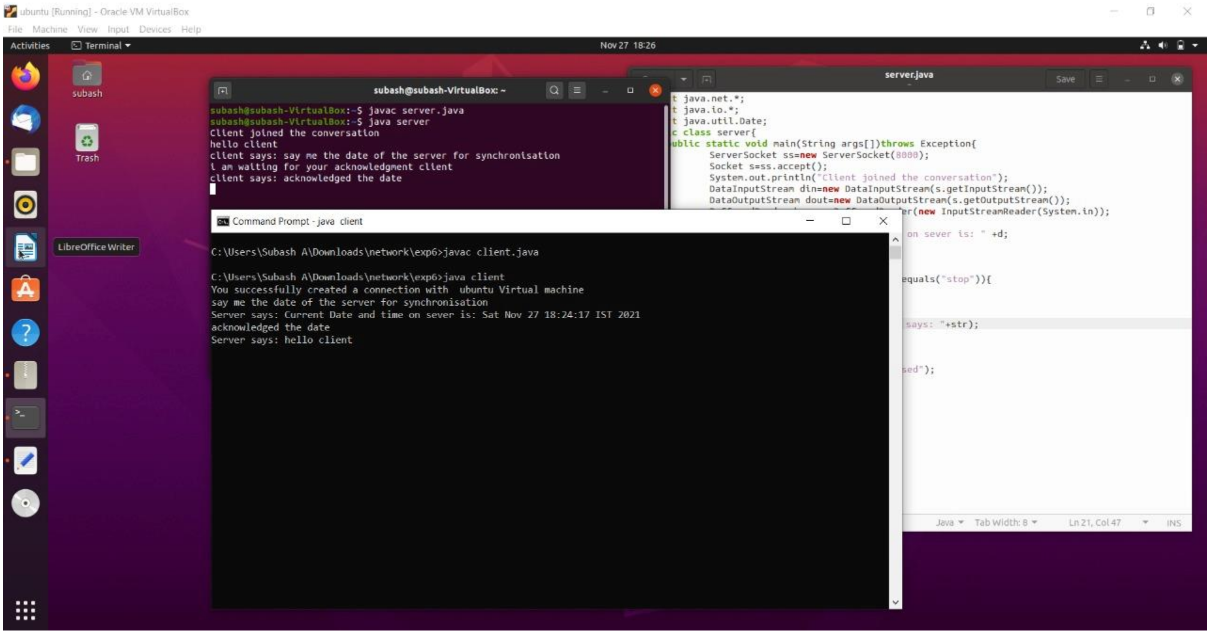
1.Create two virtual machines on different hosts following the already mentioned steps

2.Comfigure the NAT Network and perform Port forwarding to connect both the machines

3.To perform this communication both the mamchines must be connected to the same network

4.Create Server Socket on one system and run the server program.

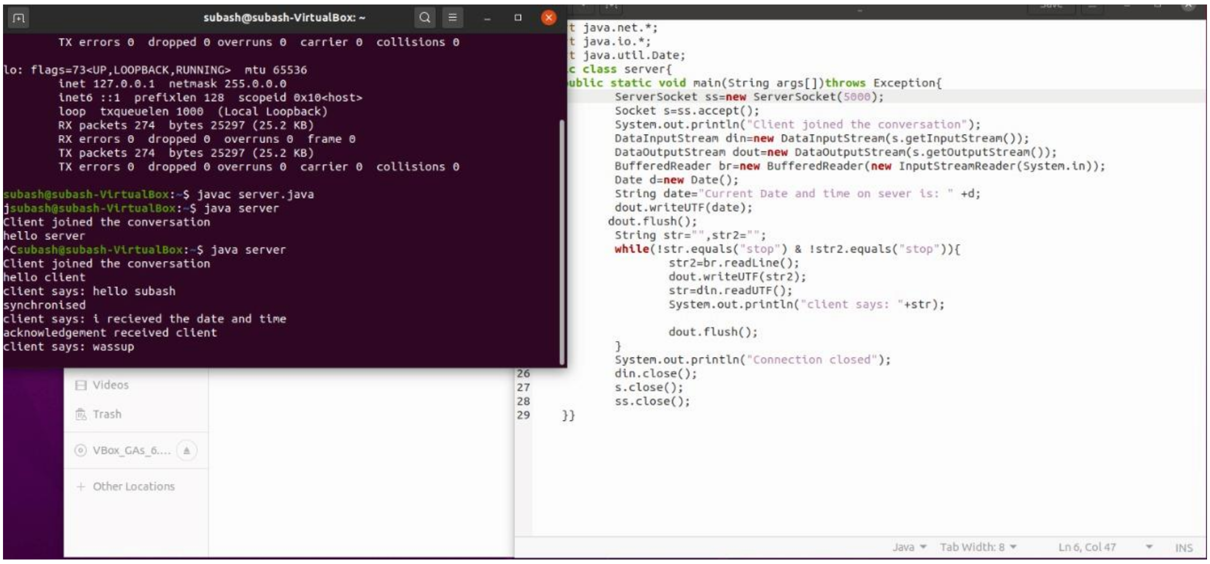
5.Run the client program on the other machine the servers port number and IP address.Thus connection would have established.Now you can perform the communication between both the machines


ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal ▾
Nov 27 18:26
subash
Trash
LibreOffice Writer
subash@subash-VirtualBox: ~
Command Prompt - java client
server.java
Save
Java ▾
Tab Width: 8 ▾
Ln 21, Col 47
INS

```
subash@subash-VirtualBox:~$ javac server.java
subash@subash-VirtualBox:~$ java server
Client joined the conversation
hello client
client says: say me the date of the server for synchronisation
i am waiting for your acknowledgment client
client says: acknowledged the date
```

```
C:\Users\Subash A\Downloads\network\exp6>javac client.java

C:\Users\Subash A\Downloads\network\exp6>java client
You successfully created a connection with  ubuntu Virtual machine
say me the date of the server for synchronisation
Server says: Current Date and time on sever is: Sat Nov 27 18:24:17 IST 2021
acknowledged the date
Server says: hello client
```

```
t java.net.*;
t java.io.*;
t java.util.Date;
c class server{
ublic static void main(String args[])throws Exception{
        ServerSocket ss=new ServerSocket(8000);
        Socket s=ss.accept();
        System.out.println("Client joined the conversation");
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
                                      er(new InputStreamReader(System.in));

                                       on sever is: " +d;

                                      equals("stop")){

                                       says: "+str);

                                      sed");
```



```
        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 274  bytes 25297 (25.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 274  bytes 25297 (25.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

subash@subash-VirtualBox:~$ javac server.java
jsubash@subash-VirtualBox:~$ java server
Client joined the conversation
hello server
^Csubash@subash-VirtualBox:~$ java server
Client joined the conversation
hello client
client says: hello subash
synchronised
client says: i recieved the date and time
acknowledgement received client
client says: wassup
```

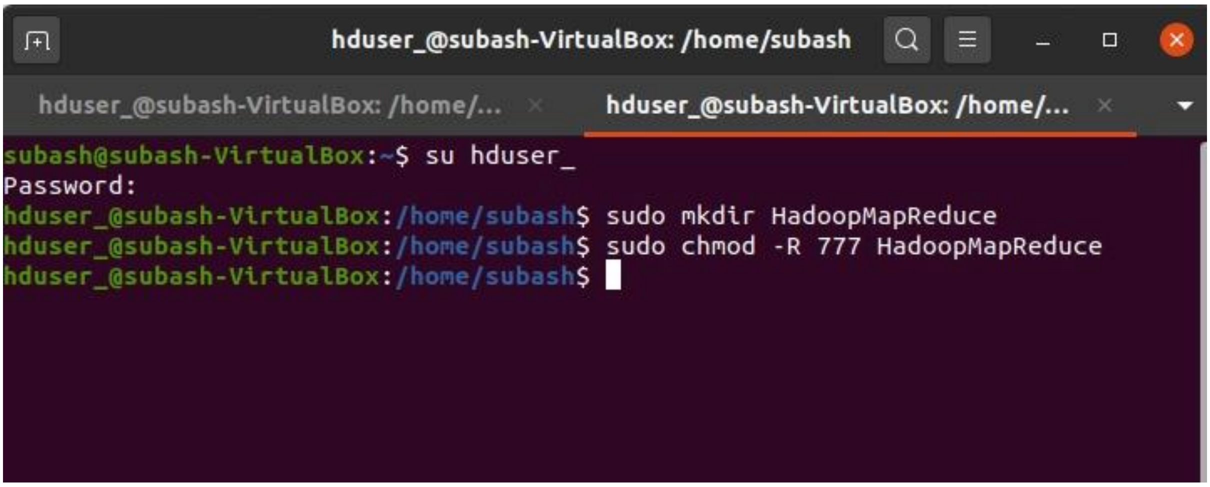
```
t java.net.*;
t java.io.*;
t java.util.Date;
c class server{
ublic static void main(String args[])throws Exception{
        ServerSocket ss=new ServerSocket(5000);
        Socket s=ss.accept();
        System.out.println("Client joined the conversation");
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        Date d=new Date();
        String date="Current Date and time on sever is: " +d;
        dout.writeUTF(date);
       dout.flush();
        String str="",str2="";
        while(!str.equals("stop") & !str2.equals("stop")){
                str2=br.readLine();
                dout.writeUTF(str2);
                str=din.readUTF();
                System.out.println("client says: "+str);

                dout.flush();
        }
        System.out.println("Connection closed");
        din.close();
        s.close();
        ss.close();
}}
```

Experiment4.

Map reduce:

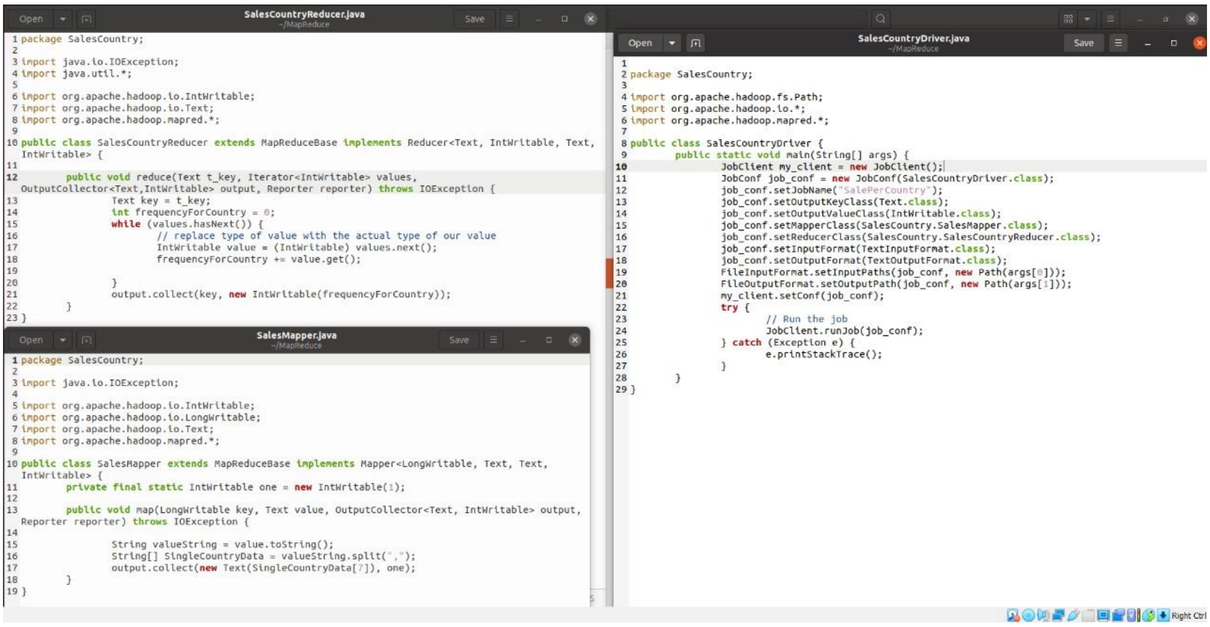
Procedure:

1.Create new User.


hduser_@subash-VirtualBox: /home/subash
hduser_@subash-VirtualBox: /home/...
hduser_@subash-VirtualBox: /home/...

```
subash@subash-VirtualBox:~$ su hduser_
Password:
hduser_@subash-VirtualBox:/home/subash$ sudo mkdir HadoopMapReduce
hduser_@subash-VirtualBox:/home/subash$ sudo chmod -R 777 HadoopMapReduce
hduser_@subash-VirtualBox:/home/subash$ 
```

2.We write the programs for the mapping,shuffling and reducing.The programs are shown below.

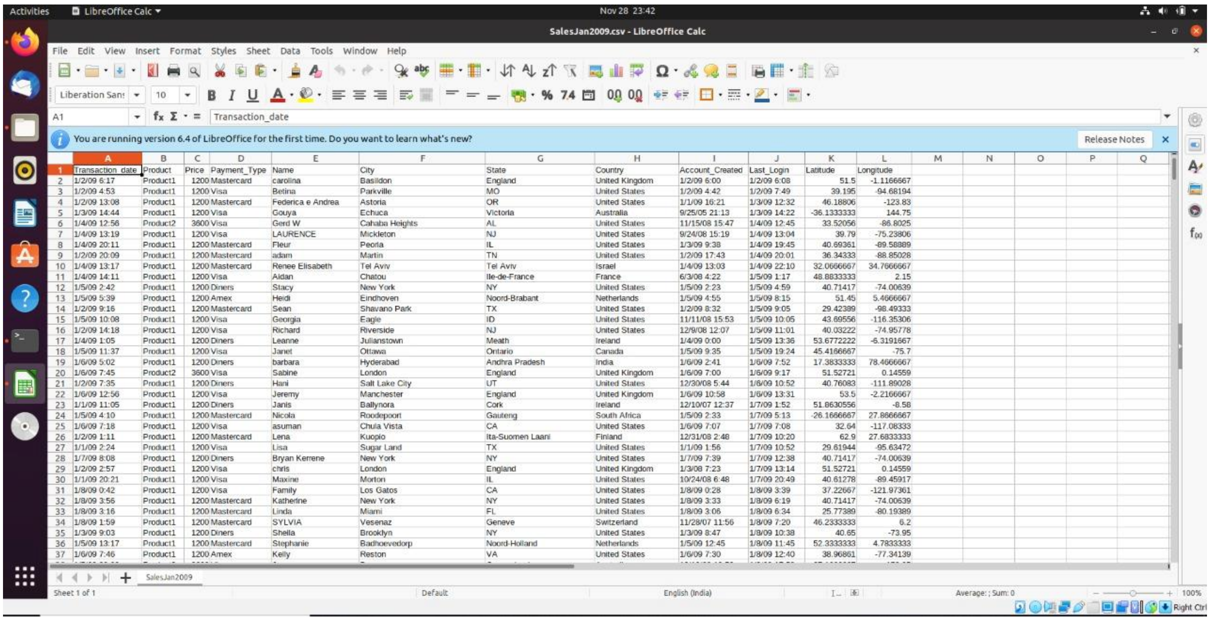


```java
package SalesCountry;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class SalesCountryReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

        public void reduce(Text t_key, Iterator<IntWritable> values, OutputCollector<Text,IntWritable> output, Reporter reporter) throws IOException {
                Text key = t_key;
                int frequencyForCountry = 0;
                while (values.hasNext()) {
                        // replace type of value with the actual type of our value
                        IntWritable value = (IntWritable) values.next();
                        frequencyForCountry += value.get();

                }
                output.collect(key, new IntWritable(frequencyForCountry));
        }
}
```

```java
package SalesCountry;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class SalesMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);

        public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {

                String valueString = value.toString();
                String[] SingleCountryData = valueString.split(",");
                output.collect(new Text(SingleCountryData[7]), one);
        }
}
```

```java

package SalesCountry;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class SalesCountryDriver {
        public static void main(String[] args) {
                JobClient my_client = new JobClient();
                JobConf job_conf = new JobConf(SalesCountryDriver.class);
                job_conf.setJobName("SalePerCountry");
                job_conf.setOutputKeyClass(Text.class);
                job_conf.setOutputValueClass(IntWritable.class);
                job_conf.setMapperClass(SalesCountry.SalesMapper.class);
                job_conf.setReducerClass(SalesCountry.SalesCountryReducer.class);
                job_conf.setInputFormat(TextInputFormat.class);
                job_conf.setOutputFormat(TextOutputFormat.class);
                FileInputFormat.setInputPaths(job_conf, new Path(args[0]));
                FileOutputFormat.setOutputPath(job_conf, new Path(args[1]));
                my_client.setConf(job_conf);
                try {
                        // Run the job
                        JobClient.runJob(job_conf);
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

3.Below is the csv File which needs to be mapreduced.

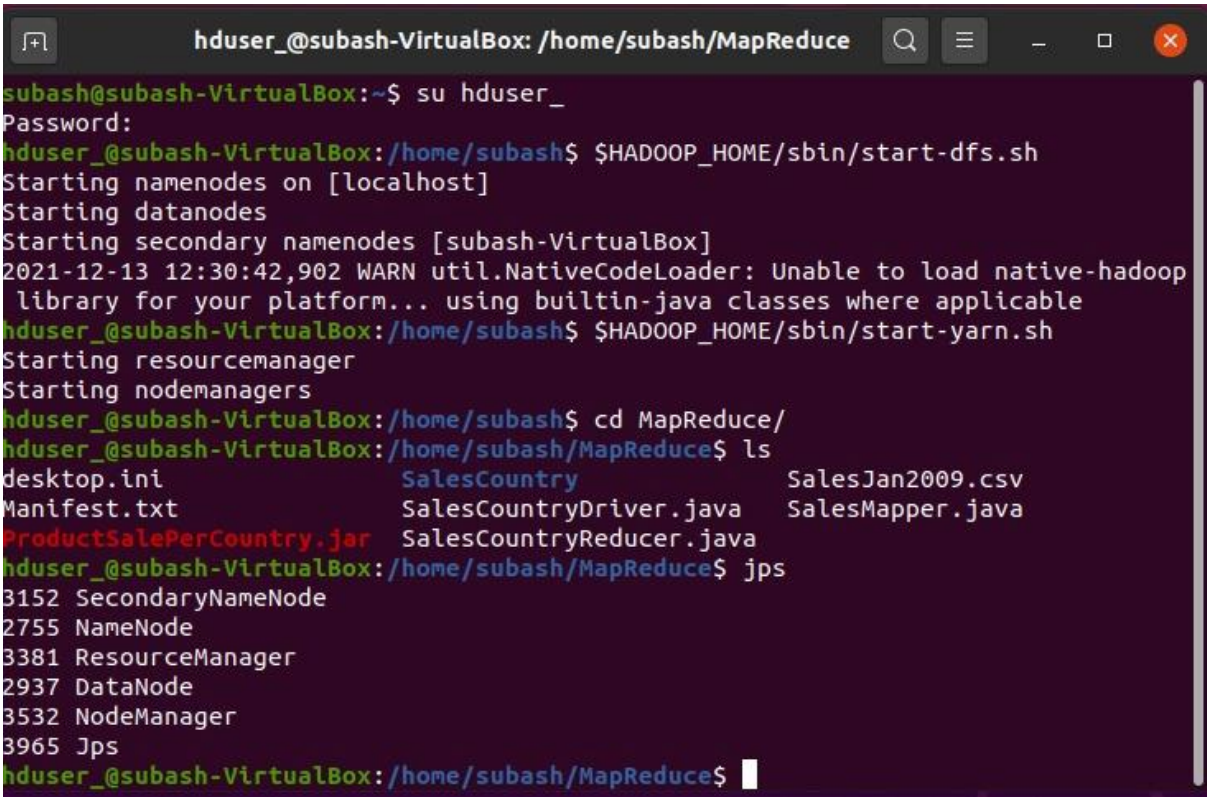


A1 Transaction_date

You are running version 6.4 of LibreOffice for the first time. Do you want to learn what's new?

Transaction_date	Product	Price	Payment_Type	Name	City	State	Country	Account_Created	Last_Login	Latitude	Longitude
1/2/09 6:17	Product1	1200	Mastercard	carolina	Basildon	England	United Kingdom	1/2/09 6:00	1/2/09 6:08	51.5	-1.1166667
1/2/09 4:53	Product1	1200	Visa	Betina	Parkville	MO	United States	1/2/09 4:42	1/2/09 7:49	39.195	-94.68194
1/2/09 13:08	Product1	1200	Mastercard	Federica e Andrea	Astoria	OR	United States	1/1/09 16:21	1/3/09 12:32	46.18806	-123.83
1/3/09 14:44	Product1	1200	Visa	Gouya	Echuca	Victoria	Australia	9/25/05 21:13	1/3/09 14:22	-36.1333333	144.75
1/4/09 12:56	Product2	3600	Visa	Gerd W	Cahaba Heights	AL	United States	11/15/08 15:47	1/4/09 12:45	33.52056	-86.8025
1/4/09 13:19	Product1	1200	Visa	LAURENCE	Mickleton	NJ	United States	9/24/08 15:19	1/4/09 13:04	39.79	-75.23806
1/4/09 20:11	Product1	1200	Mastercard	Fleur	Peoria	IL	United States	1/3/09 9:38	1/4/09 19:45	40.69361	-89.58889
1/2/09 20:09	Product1	1200	Mastercard	adam	Martin	TN	United States	1/2/09 17:43	1/4/09 20:01	36.34333	-88.85028
1/4/09 13:17	Product1	1200	Mastercard	Renee Elisabeth	Tel Aviv	Tel Aviv	Israel	1/4/09 13:03	1/4/09 22:10	32.0666667	34.7666667
1/4/09 14:11	Product1	1200	Visa	Aidan	Chatou	Ile-de-France	France	6/3/08 4:22	1/5/09 1:17	48.8833333	2.15
1/5/09 2:42	Product1	1200	Diners	Stacy	New York	NY	United States	1/5/09 2:23	1/5/09 4:59	40.71417	-74.00639
1/5/09 5:39	Product1	1200	Amex	Heidi	Eindhoven	Noord-Brabant	Netherlands	1/5/09 4:55	1/5/09 8:15	51.45	5.4666667
1/2/09 9:16	Product1	1200	Mastercard	Sean	Shavano Park	TX	United States	1/2/09 8:32	1/5/09 9:05	29.42389	-98.49333
1/5/09 10:08	Product1	1200	Visa	Georgia	Eagle	ID	United States	11/11/08 15:53	1/5/09 10:05	43.69556	-116.35306
1/2/09 14:18	Product1	1200	Visa	Richard	Riverside	NJ	United States	12/9/08 12:07	1/5/09 11:01	40.03222	-74.95778
1/4/09 1:05	Product1	1200	Diners	Leanne	Julianstown	Meath	Ireland	1/4/09 0:00	1/5/09 13:36	53.6772222	-6.3191667
1/5/09 11:37	Product1	1200	Visa	Janet	Ottawa	Ontario	Canada	1/5/09 9:35	1/5/09 19:24	45.4166667	-75.7
1/6/09 5:02	Product1	1200	Diners	barbara	Hyderabad	Andhra Pradesh	India	1/6/09 2:41	1/6/09 7:52	17.3833333	78.4666667
1/6/09 7:45	Product2	3600	Visa	Sabine	London	England	United Kingdom	1/6/09 7:00	1/6/09 9:17	51.52721	0.14559
1/2/09 7:35	Product1	1200	Diners	Hani	Salt Lake City	UT	United States	12/30/08 5:44	1/6/09 10:52	40.76083	-111.89028
1/6/09 12:56	Product1	1200	Visa	Jeremy	Manchester	England	United Kingdom	1/6/09 10:58	1/6/09 13:31	53.5	-2.2166667
1/1/09 11:05	Product1	1200	Diners	Janis	Ballynora	Cork	Ireland	12/10/07 12:37	1/7/09 1:52	51.8630556	-8.58
1/5/09 4:10	Product1	1200	Mastercard	Nicola	Roodepoort	Gauteng	South Africa	1/5/09 2:33	1/7/09 5:13	-26.1666667	27.8666667
1/6/09 7:18	Product1	1200	Visa	asuman	Chula Vista	CA	United States	1/6/09 7:07	1/7/09 7:08	32.64	-117.08333
1/2/09 1:11	Product1	1200	Mastercard	Lena	Kuopio	Ita-Suomen Laani	Finland	12/31/08 2:48	1/7/09 10:20	62.9	27.6833333
1/1/09 2:24	Product1	1200	Visa	Lisa	Sugar Land	TX	United States	1/1/09 1:56	1/7/09 10:52	29.61944	-95.63472
1/7/09 8:08	Product1	1200	Diners	Bryan Kerrene	New York	NY	United States	1/7/09 7:39	1/7/09 12:38	40.71417	-74.00639
1/2/09 2:57	Product1	1200	Visa	chris	London	England	United Kingdom	1/3/08 7:23	1/7/09 13:14	51.52721	0.14559
1/1/09 20:21	Product1	1200	Visa	Maxine	Morton	IL	United States	10/24/08 6:48	1/7/09 20:49	40.61278	-89.45917
1/8/09 0:42	Product1	1200	Visa	Family	Los Gatos	CA	United States	1/8/09 0:28	1/8/09 3:39	37.22667	-121.97361
1/8/09 3:56	Product1	1200	Mastercard	Katherine	New York	NY	United States	1/8/09 3:33	1/8/09 6:19	40.71417	-74.00639
1/8/09 3:16	Product1	1200	Mastercard	Linda	Miami	FL	United States	1/8/09 3:06	1/8/09 6:34	25.77389	-80.19389
1/8/09 1:59	Product1	1200	Mastercard	SYLVIA	Vesenaz	Geneve	Switzerland	11/28/07 11:56	1/8/09 7:20	46.2333333	6.2
1/3/09 9:03	Product1	1200	Diners	Sheila	Brooklyn	NY	United States	1/3/09 8:47	1/8/09 10:38	40.65	-73.95
1/5/09 13:17	Product1	1200	Mastercard	Stephanie	Badhoevedorp	Noord-Holland	Netherlands	1/5/09 12:45	1/8/09 11:45	52.3333333	4.7833333
1/6/09 7:46	Product1	1200	Amex	Kelly	Reston	VA	United States	1/6/09 7:30	1/8/09 12:40	38.96861	-77.34139

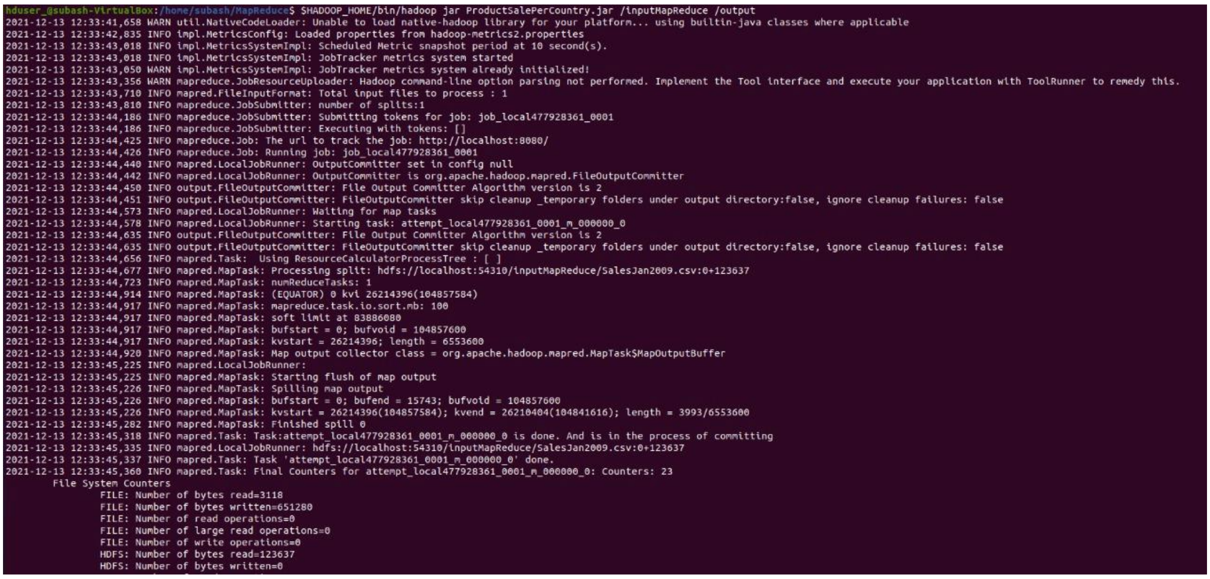
4.In the command Prompt we gert Hadoop started.



hduser_@subash-VirtualBox: /home/subash/MapReduce

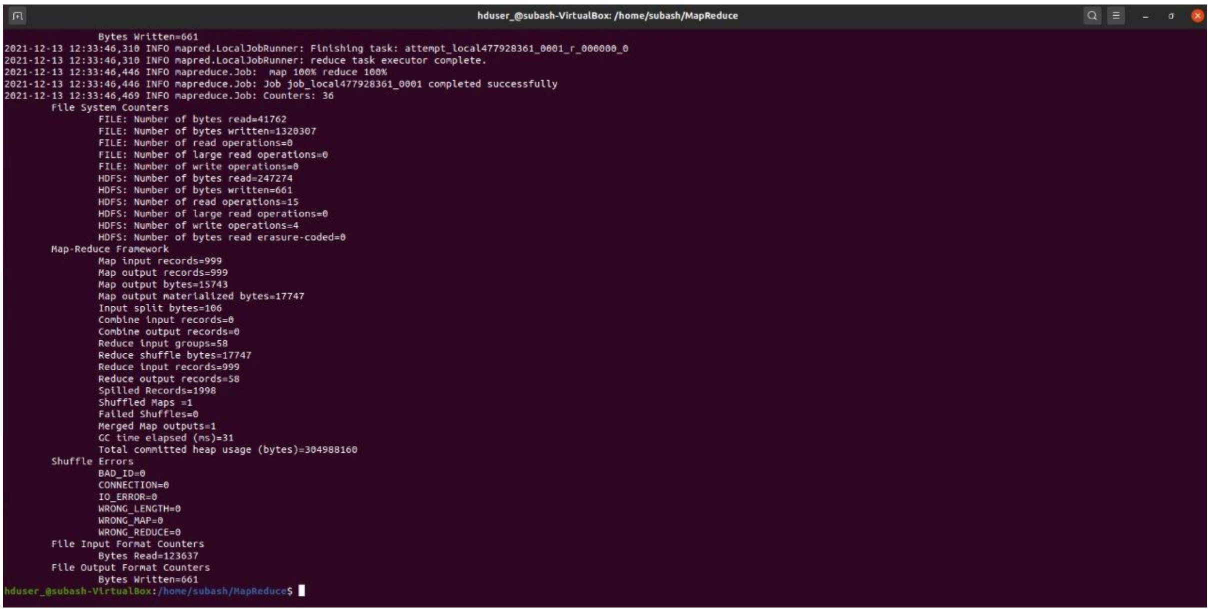
```
subash@subash-VirtualBox:~$ su hduser_
Password:
hduser_@subash-VirtualBox:/home/subash$ $HADOOP_HOME/sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [subash-VirtualBox]
2021-12-13 12:30:42,902 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
hduser_@subash-VirtualBox:/home/subash$ $HADOOP_HOME/sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser_@subash-VirtualBox:/home/subash$ cd MapReduce/
hduser_@subash-VirtualBox:/home/subash/MapReduce$ ls
desktop.ini                SalesCountry               SalesJan2009.csv
Manifest.txt               SalesCountryDriver.java    SalesMapper.java
ProductSalePerCountry.jar  SalesCountryReducer.java
hduser_@subash-VirtualBox:/home/subash/MapReduce$ jps
3152 SecondaryNameNode
2755 NameNode
3381 ResourceManager
2937 DataNode
3532 NodeManager
3965 Jps
hduser_@subash-VirtualBox:/home/subash/MapReduce$ 
```

5.Then we run the jar file which has the compilation of the programs required for map reduce.



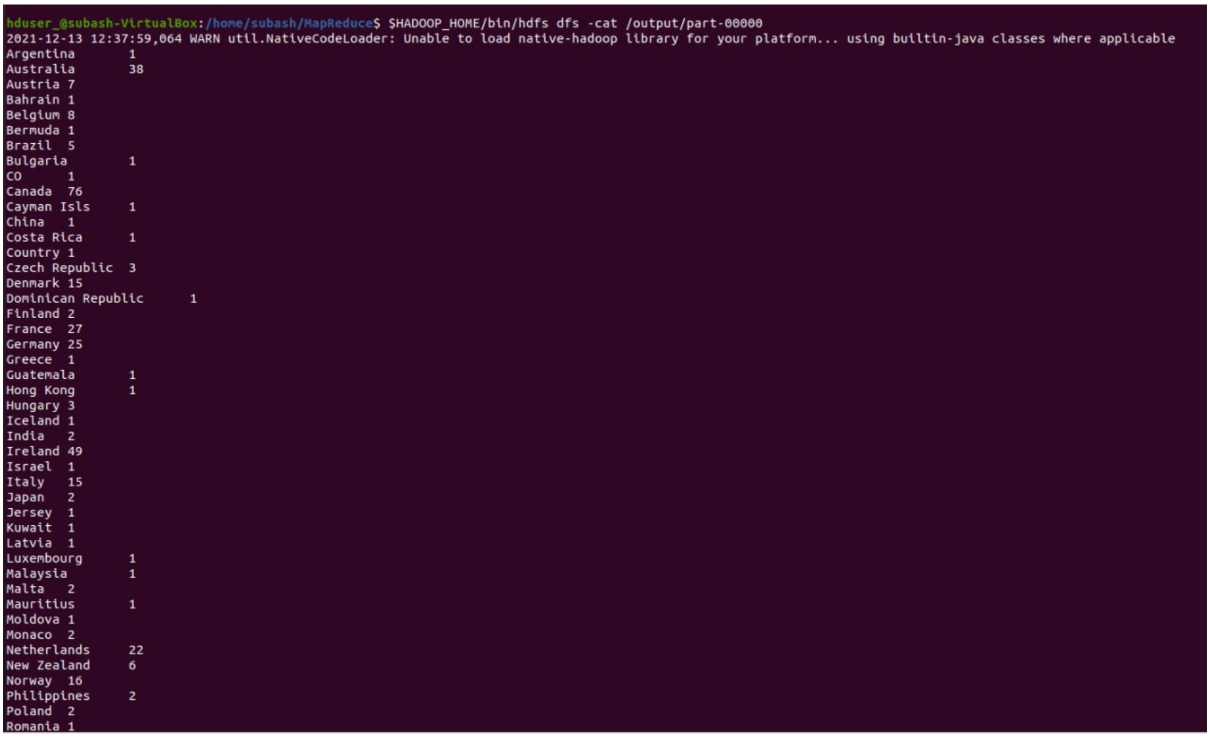
```
hduser_@subash-VirtualBox:/home/subash/MapReduce$ $HADOOP_HOME/bin/hadoop jar ProductSalePerCountry.jar /inputMapReduce /output
2021-12-13 12:33:41,658 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2021-12-13 12:33:42,835 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2021-12-13 12:33:43,018 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-12-13 12:33:43,018 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-12-13 12:33:43,050 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2021-12-13 12:33:43,356 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-12-13 12:33:43,710 INFO mapred.FileInputFormat: Total input files to process : 1
2021-12-13 12:33:43,810 INFO mapreduce.JobSubmitter: number of splits:1
2021-12-13 12:33:44,186 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local477928361_0001
2021-12-13 12:33:44,186 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-12-13 12:33:44,425 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-12-13 12:33:44,426 INFO mapreduce.Job: Running job: job_local477928361_0001
2021-12-13 12:33:44,440 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-12-13 12:33:44,442 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2021-12-13 12:33:44,450 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-12-13 12:33:44,451 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2021-12-13 12:33:44,573 INFO mapred.LocalJobRunner: Waiting for map tasks
2021-12-13 12:33:44,578 INFO mapred.LocalJobRunner: Starting task: attempt_local477928361_0001_m_000000_0
2021-12-13 12:33:44,635 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-12-13 12:33:44,635 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2021-12-13 12:33:44,656 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2021-12-13 12:33:44,677 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/inputMapReduce/SalesJan2009.csv:0+123637
2021-12-13 12:33:44,723 INFO mapred.MapTask: numReduceTasks: 1
2021-12-13 12:33:44,914 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2021-12-13 12:33:44,917 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2021-12-13 12:33:44,917 INFO mapred.MapTask: soft limit at 83886080
2021-12-13 12:33:44,917 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2021-12-13 12:33:44,917 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2021-12-13 12:33:44,920 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2021-12-13 12:33:45,225 INFO mapred.LocalJobRunner:
2021-12-13 12:33:45,225 INFO mapred.MapTask: Starting flush of map output
2021-12-13 12:33:45,226 INFO mapred.MapTask: Spilling map output
2021-12-13 12:33:45,226 INFO mapred.MapTask: bufstart = 0; bufend = 15743; bufvoid = 104857600
2021-12-13 12:33:45,226 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26210404(104841616); length = 3993/6553600
2021-12-13 12:33:45,282 INFO mapred.MapTask: Finished spill 0
2021-12-13 12:33:45,318 INFO mapred.Task: Task:attempt_local477928361_0001_m_000000_0 is done. And is in the process of committing
2021-12-13 12:33:45,335 INFO mapred.LocalJobRunner: hdfs://localhost:54310/inputMapReduce/SalesJan2009.csv:0+123637
2021-12-13 12:33:45,337 INFO mapred.Task: Task 'attempt_local477928361_0001_m_000000_0' done.
2021-12-13 12:33:45,360 INFO mapred.Task: Final Counters for attempt_local477928361_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=3118
                FILE: Number of bytes written=651280
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=123637
                HDFS: Number of bytes written=0
```

6. We can see the process happening



```
                Bytes Written=661
2021-12-13 12:33:46,310 INFO mapred.LocalJobRunner: Finishing task: attempt_local477928361_0001_r_000000_0
2021-12-13 12:33:46,310 INFO mapred.LocalJobRunner: reduce task executor complete.
2021-12-13 12:33:46,446 INFO mapreduce.Job:  map 100% reduce 100%
2021-12-13 12:33:46,446 INFO mapreduce.Job: Job job_local477928361_0001 completed successfully
2021-12-13 12:33:46,469 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=41762
                FILE: Number of bytes written=1320307
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=247274
                HDFS: Number of bytes written=661
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=999
                Map output records=999
                Map output bytes=15743
                Map output materialized bytes=17747
                Input split bytes=106
                Combine input records=0
                Combine output records=0
                Reduce input groups=58
                Reduce shuffle bytes=17747
                Reduce input records=999
                Reduce output records=58
                Spilled Records=1998
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=31
                Total committed heap usage (bytes)=304988160
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=123637
        File Output Format Counters
                Bytes Written=661
hduser_@subash-VirtualBox:/home/subash/MapReduce$ 
```

7.From the directory we specified for the output,we could access the final output.We are left with the reduced data.



```
hduser_@subash-VirtualBox:/home/subash/MapReduce$ $HADOOP_HOME/bin/hdfs dfs -cat /output/part-00000
2021-12-13 12:37:59,064 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Argentina       1
Australia       38
Austria 7
Bahrain 1
Belgium 8
Bermuda 1
Brazil  5
Bulgaria        1
CO      1
Canada  76
Cayman Isls     1
China   1
Costa Rica      1
Country 1
Czech Republic  3
Denmark 15
Dominican Republic      1
Finland 2
France  27
Germany 25
Greece  1
Guatemala       1
Hong Kong       1
Hungary 3
Iceland 1
India   2
Ireland 49
Israel  1
Italy   15
Japan   2
Jersey  1
Kuwait  1
Latvia  1
Luxembourg      1
Malaysia        1
Malta   2
Mauritius       1
Moldova 1
Monaco  2
Netherlands     22
New Zealand     6
Norway  16
Philippines     2
Poland  2
Romania 1
```

Concurrent Programming

About concurrency:

Concurrency is the ability of a processor to execute multiple instruction sequences to run on a single/multiple processor, thus using the full processing capability of a processor. It allows for parallel execution of concurrent units which can significantly improve the overall speed of execution in multi-processor and multi-core system. Although concurrency allows parallel execution, it is slightly different from parallelism. Concurrency is about dealing with lot of things at once whereas parallelism is about doing a lot of things at once. Basically concurrency is about and structure whereas parallelism is about execution.

Concurrency is achieved in Java by using Threads and the Runnable interface. Multiple threads are spawned to execute the multiple instruction sequences and each thread's execution is controlled by the Runnable interface. Synchronization is achieved in Java using the synchronized keyword which allows the function(shared resource) to be executable only by one thread at a time.

Multiple clients – server chat in Java using concurrency:

Server is used here to wrap each client with a thread using a ClientHandler class and show how many clients are connected at that time of instance. ClientHandler class is used to get the incoming messages from all clients and display it on the current client's terminal. The Client class is used to get inputs from each client terminal and send it to the ClientHandler class which would broadcast the message to other clients. Each client is identified by their username so that we can tell the other clients who is sending the particular message.

The code goes as follows:

Server.java

```java
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
    private ServerSocket ss;

    public Server(ServerSocket ss) {
        this.ss = ss;
    }

    public void startServer() {
        int count = 0;
        try {
            while(!ss.isClosed()) {
                Socket s = ss.accept();
                System.out.println(++count +" client(s) connected");
                ClientHandler clientHandler = new ClientHandler(s); // ClientHandler implements Runnable which will spawn a new thread for each client that is connected with the server

                Thread thread = new Thread(clientHandler); // Thread created to invoke the run() in ClientHandler
                thread.start();
            }
        } catch (IOException e) {
            closeServerSocket();
        }
    }

    public void closeServerSocket() { // function to avoid nested try catches in startServer
        try {
            if(ss != null) {
                ss.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(6767);
        Server server = new Server(serverSocket);
        server.startServer();
    }
}
```

Client.java

```
import java.io.*;
import java.net.Socket;
import java.util.Scanner;
```

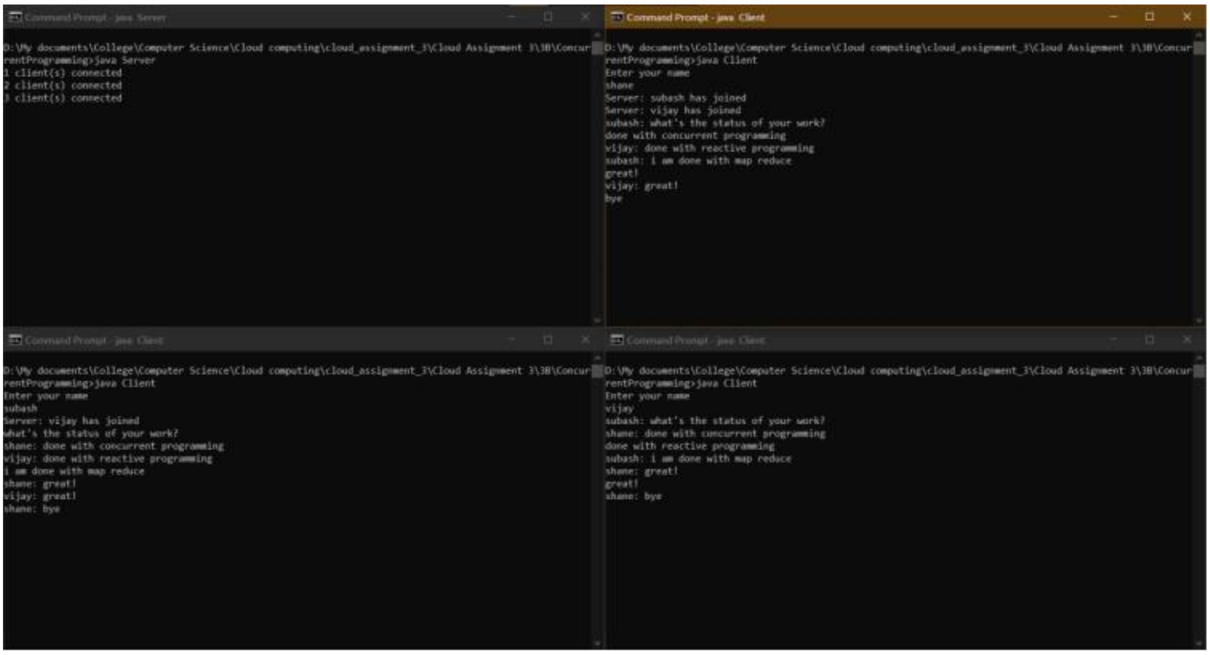
```
public class Client {
    private Socket s;
    private BufferedReader in;
    private BufferedWriter out;
    private String clientName;

    public Client(Socket s, String clientName) {
        try {
            this.s = s;
            this.in = new BufferedReader(new
InputStreamReader(s.getInputStream()));
            this.out = new BufferedWriter(new
OutputStreamWriter(s.getOutputStream()));
            this.clientName = clientName;
        } catch(IOException e) {
            closeSocketAndStreams(s, in, out);
        }
    }

    public void sendMessage() {
        try {
            out.write(clientName);
            out.newLine();
            out.flush();

            Scanner scanner = new Scanner(System.in);
            while(s.isConnected()) {
                String messageToSend = scanner.nextLine();
                out.write(clientName+": "+messageToSend);
                out.newLine();
                out.flush();
            }
        } catch (IOException e) {
            closeSocketAndStreams(s, in, out);
        }
    }

    public void listenForMessages() {
        new Thread(new Runnable() { // listening to messages is a blocking operation. If thread is not used here then we may end up waiting for messages from other users and not be able to send messages
            @Override
            public void run() {
                String messageFromClients;

                while(s.isConnected()) {
                    try {
                        messageFromClients = in.readLine();
                        System.out.println(messageFromClients);
                    } catch(IOException e) {
                        closeSocketAndStreams(s, in, out);
                    }
                }
            }
        }).start();
    }
    public void closeSocketAndStreams(Socket s, BufferedReader in, BufferedWriter out) {

        try {
            if(in != null) {
                in.close();
            }

            if(out != null) {
                out.close();
            }

            if(s != null) {
                s.close();
            }
        } catch(IOException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws IOException {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your name");
        String clientName = scanner.nextLine();
        Socket s = new Socket("localhost", 6767);
        Client client = new Client(s, clientName);
        client.listenForMessages();
        client.sendMessage();
    }
```

```
}
```

ClientHandler.java

```java
import java.io.*;
import java.net.Socket;
import java.util.ArrayList;

public class ClientHandler implements Runnable{

    public static ArrayList<ClientHandler> clientHandlers = new ArrayList<>();
    private Socket s;
    private BufferedReader in;
    private BufferedWriter out;
    private String clientName;

    public ClientHandler(Socket s) {
        try {
            this.s = s;
            this.out = new BufferedWriter(new OutputStreamWriter(s.getOutputStream()));
            this.in = new BufferedReader(new InputStreamReader(s.getInputStream()));
            this.clientName = in.readLine();//get the client's name using BufferedReader
            clientHandlers.add(this);
            broadcastMessage("Server: "+clientName+" has joined");
        } catch (IOException e) {
            closeSocketAndStreams(s, in, out);
        }
    }


    @Override
    public void run() {
        String messageFromClient;
        while(s.isConnected()) {
            try {
                messageFromClient = in.readLine(); // reading a message will block the client thread. Therefore it must be run on seperate threads
                if(messageFromClient.equals("stop")) {
                    throw new IOException(clientName+" has left");
                } else {
                    broadcastMessage(messageFromClient);
                }

            } catch (IOException e) {
                closeSocketAndStreams(s, in, out);
                break;
            }
        }
    }

    public void broadcastMessage(String messageToSend) {
        for(ClientHandler clientHandler: clientHandlers) {
            try {
                if(!clientHandler.clientName.equals(clientName)) {
                    clientHandler.out.write(messageToSend);
                    clientHandler.out.newLine();
                    clientHandler.out.flush();
                }
            } catch(IOException e) {
                closeSocketAndStreams(s, in, out);
            }
        }
    }

    public void removeClientHandler() {
        clientHandlers.remove(this);
        broadcastMessage("Server: "+clientName+" has left");
    }

    public void closeSocketAndStreams(Socket s, BufferedReader in,
BufferedWriter out) {
        removeClientHandler(); // if some error happens with a client we are
removing the client from the chat
        try {
            if(in != null) {
                in.close();
            }

            if(out != null) {
                out.close();
            }

            if(s != null) {
                s.close();
            }
        } catch(IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

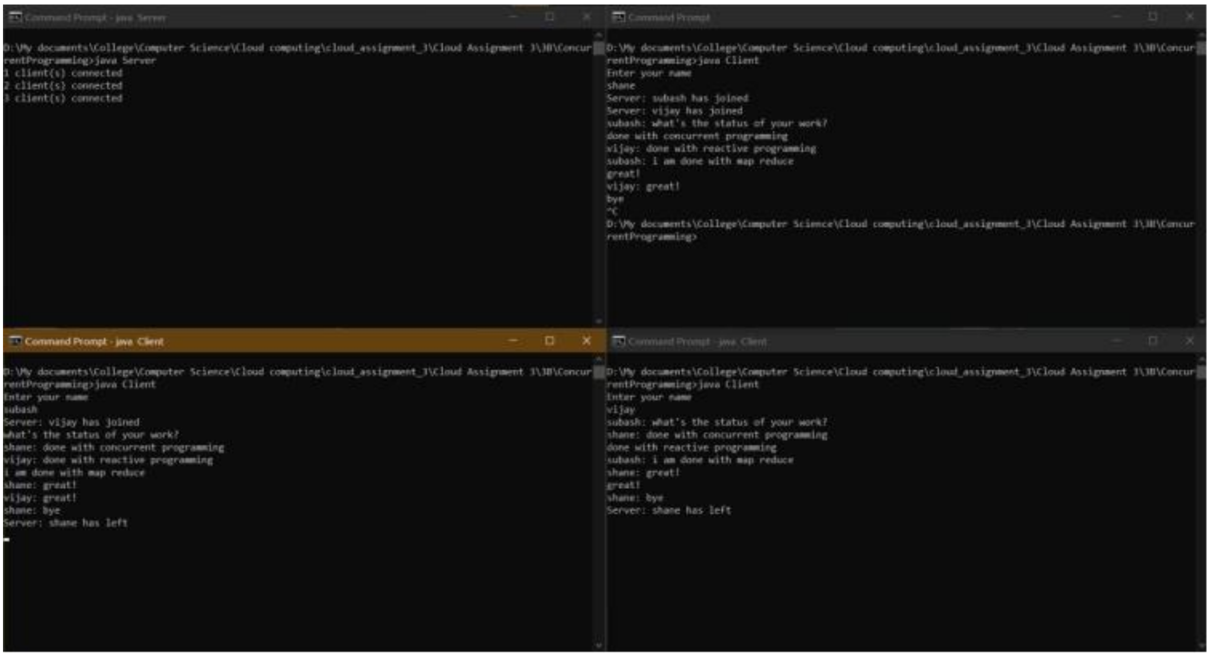

Command Prompt - java Server
Command Prompt - java Client
Command Prompt - java Client
Command Prompt - java Client

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Server
1 client(s) connected
2 client(s) connected
3 client(s) connected
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
shane
Server: subash has joined
Server: vijay has joined
subash: what's the status of your work?
done with concurrent programming
vijay: done with reactive programming
subash: i am done with map reduce
great!
vijay: great!
bye
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
subash
Server: vijay has joined
what's the status of your work?
shane: done with concurrent programming
vijay: done with reactive programming
i am done with map reduce
shane: great!
vijay: great!
shane: bye
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
vijay
subash: what's the status of your work?
shane: done with concurrent programming
done with reactive programming
subash: i am done with map reduce
shane: great!
great!
shane: bye
```

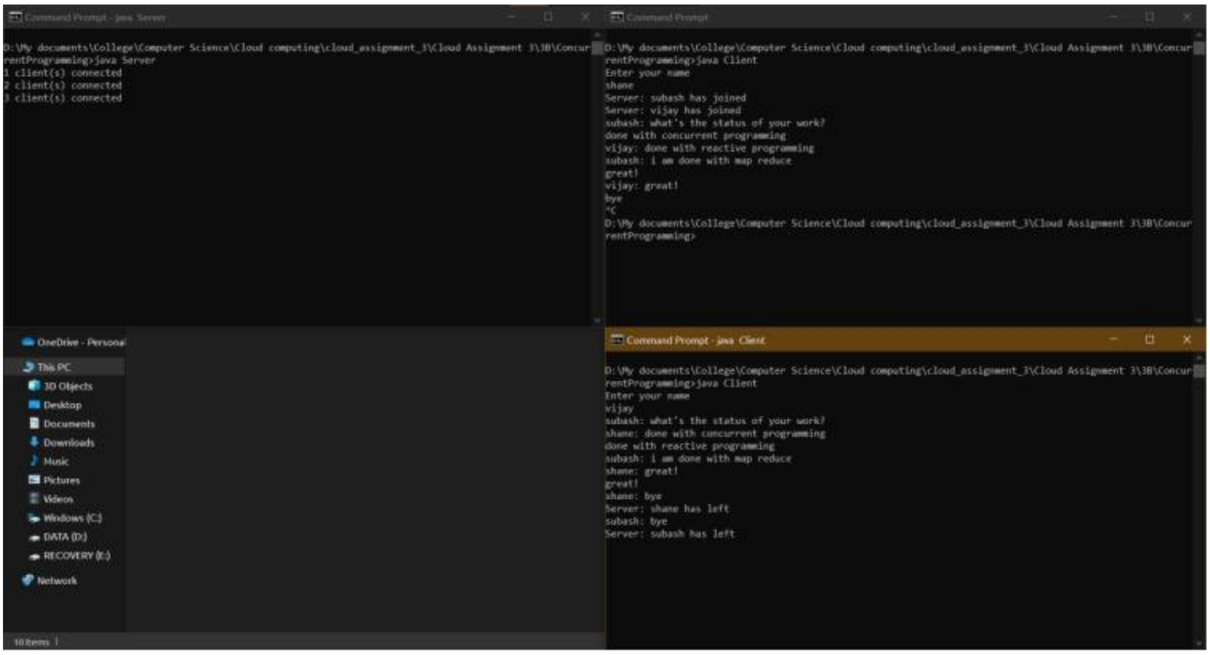


```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Server
1 client(s) connected
2 client(s) connected
3 client(s) connected
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
shane
Server: subash has joined
Server: vijay has joined
subash: what's the status of your work?
done with concurrent programming
vijay: done with reactive programming
subash: i am done with map reduce
great!
vijay: great!
bye
^C
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
subash
Server: vijay has joined
what's the status of your work?
shane: done with concurrent programming
vijay: done with reactive programming
i am done with map reduce
shane: great!
vijay: great!
shane: bye
Server: shane has left
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
vijay
subash: what's the status of your work?
shane: done with concurrent programming
done with reactive programming
subash: i am done with map reduce
shane: great!
great!
shane: bye
Server: shane has left
```



```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Server
1 client(s) connected
2 client(s) connected
3 client(s) connected
```

```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
shane
Server: subash has joined
Server: vijay has joined
subash: what's the status of your work?
done with concurrent programming
vijay: done with reactive programming
subash: i am done with map reduce
great!
vijay: great!
bye
^C
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>
```

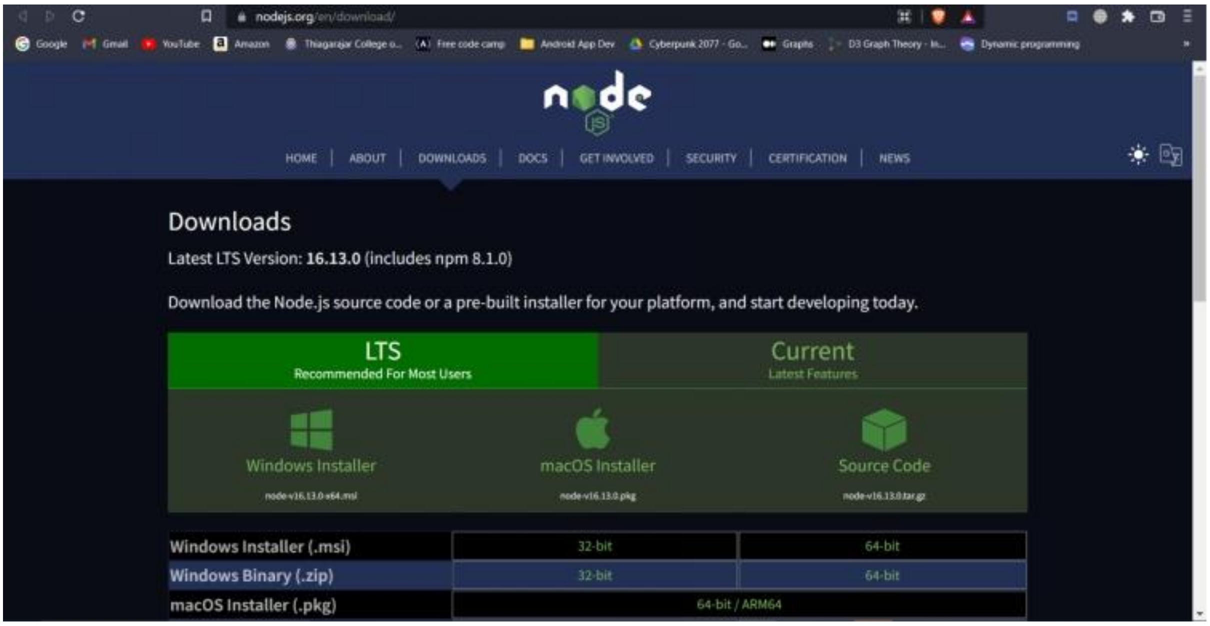
```
D:\My documents\College\Computer Science\Cloud computing\cloud_assignment_3\Cloud Assignment 3\3B\Concur
rentProgramming>java Client
Enter your name
vijay
subash: what's the status of your work?
shane: done with concurrent programming
done with reactive programming
subash: i am done with map reduce
shane: great!
great!
shane: bye
Server: shane has left
subash: bye
Server: subash has left
```

Thus, concurrent programming model has been used to perform multiple clients server chat successfully.

Reactive Programming using RxJS

Step 1: To work with RxJS, we need the following things: NodeJS, npm and RxJS package installation

Step 2: Download NodeJS from the website https://nodejs.org/en/download/. Installation of NodeJS is similar to other free and open source software. This would install npm as well



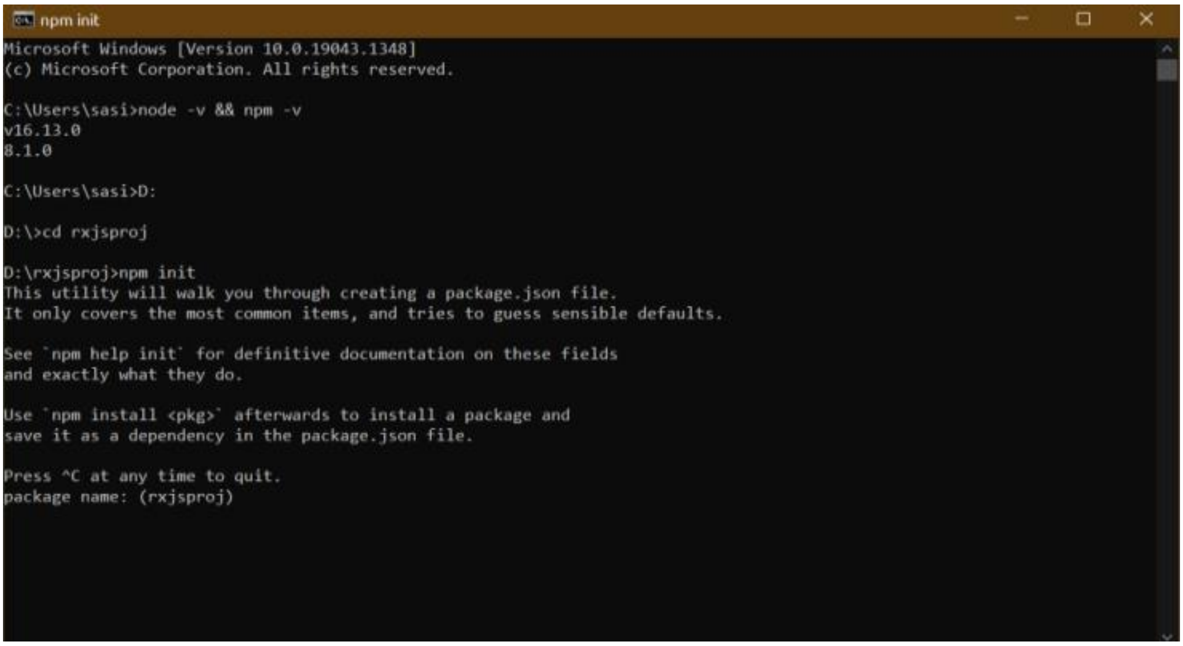
Downloads

Latest LTS Version: **16.13.0** (includes npm 8.1.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

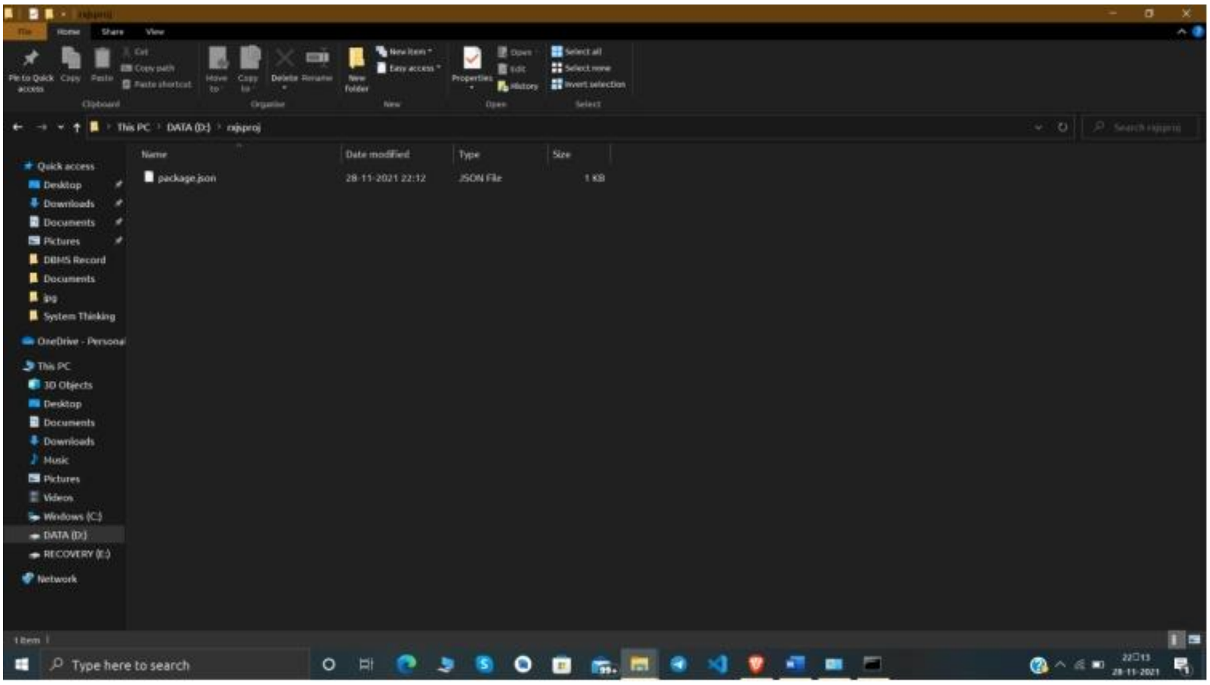
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	

Step 3: Create a folder named rxjsproj to store the example program and navigate to that directory in cmd. Then type npm init to create a package.json for project setup



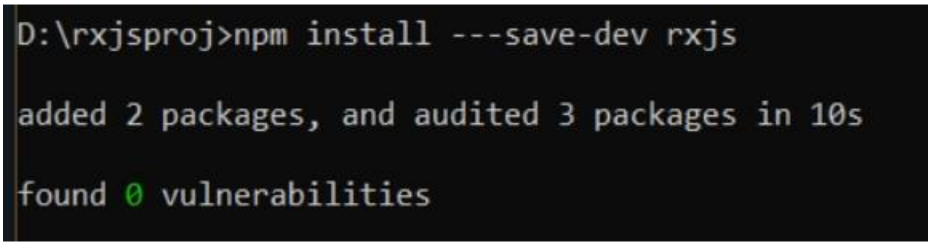
```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sasi>node -v && npm -v
v16.13.0
8.1.0

C:\Users\sasi>D:

D:\>cd rxjsproj

D:\rxjsproj>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (rxjsproj)
```

Step 4: After this process you will see a package.json file in the rxjsproj folder



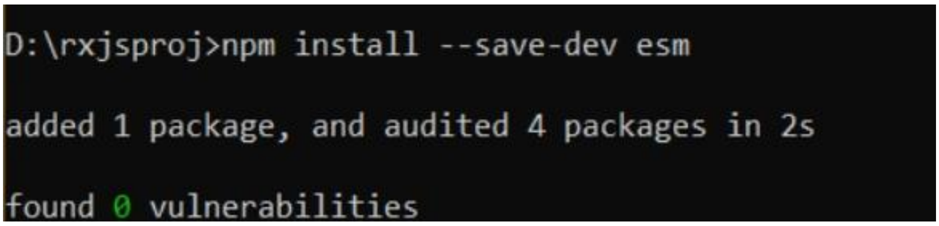
Name	Date modified	Type	Size
package.json	28-11-2021 22:12	JSON File	1 KB

Step 5: Install rxjs using the following command: npm install ---save-dev rxjs



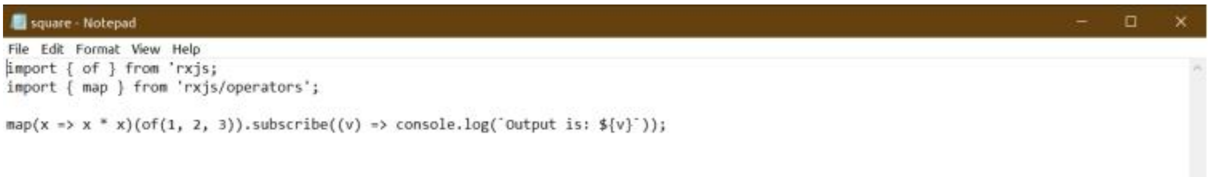
```
D:\rxjsproj>npm install ---save-dev rxjs

added 2 packages, and audited 3 packages in 10s

found 0 vulnerabilities
```

Step 6: Install ep6 modules to avoid any problems with importing packages which node requires using the following command: npm install --save-dev esm



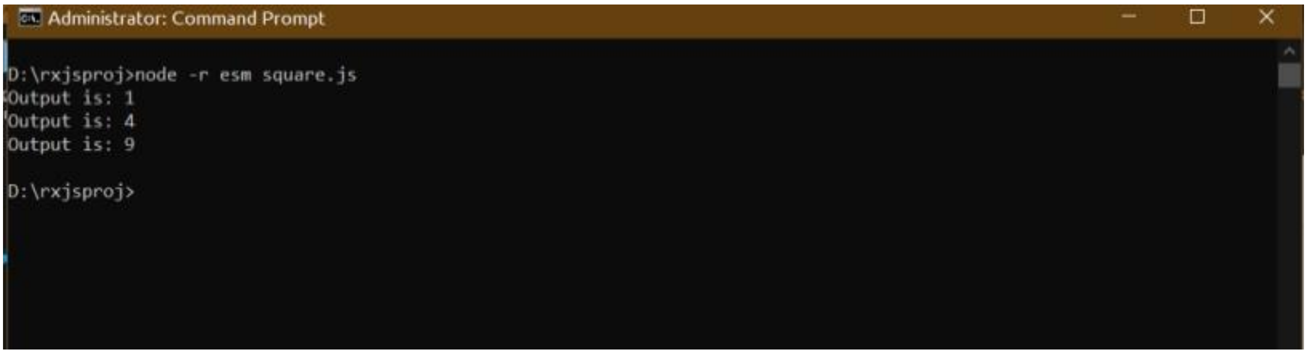
```
D:\rxjsproj>npm install --save-dev esm

added 1 package, and audited 4 packages in 2s

found 0 vulnerabilities
```

Step 7: Create a javascript file named square.js to demonstrate reactive programming



```
import { of } from 'rxjs;
import { map } from 'rxjs/operators';

map(x => x * x)(of(1, 2, 3)).subscribe((v) => console.log(`Output is: ${v}`));
```

Step 8: Run the program using the following command: node -r esm square.js



```
D:\rxjsproj>node -r esm square.js
Output is: 1
Output is: 4
Output is: 9

D:\rxjsproj>
```

And we get the desired output after execution