CSE 3035

PRINCIPLES OF CLOUD COMPUTING

Lab Assessment – 3

L15+L16 | SJT501
Dr. Sivaprakash S

FALL SEMESTER 2022-23

by

SHARADINDU ADHIKARI
19BCE2105

Assessment 3 .

Experiment: 1

Title: X11 Forwarding

Aim:

- To capture the application execution of VM instances in Host OS using X11 traffic forwarding; and
- To test the connection with AWS EC2 VM instance and Local VM instance using X11 traffic forwarding from the Host OS using Xming and Putty tools.

Background Theory:

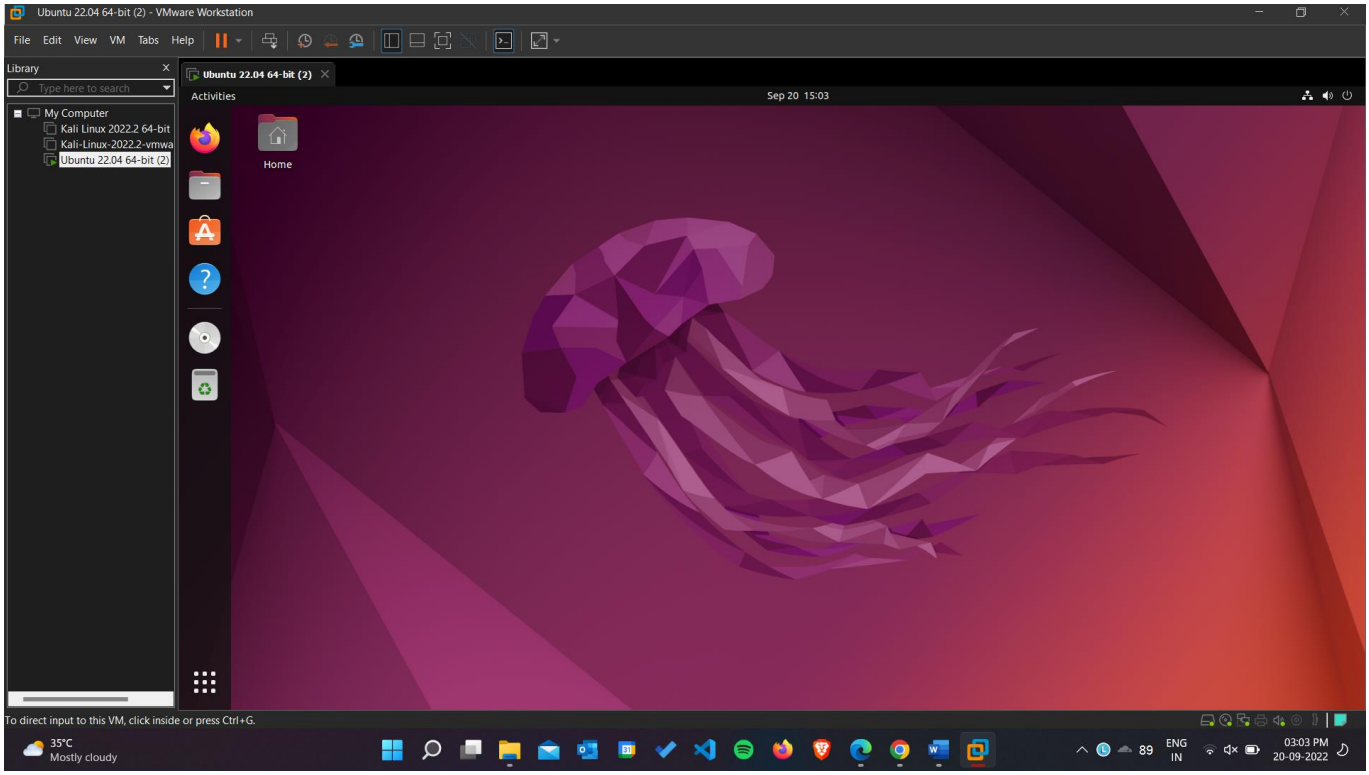
The X Window System (also known as X11, or just X) is a software package and network protocol that lets you interact locally, using your personal computer's display, mouse, and keyboard, with the graphical user interface (GUI) of an application running on a remote networked computer.

Requirements for conducting the experiment:

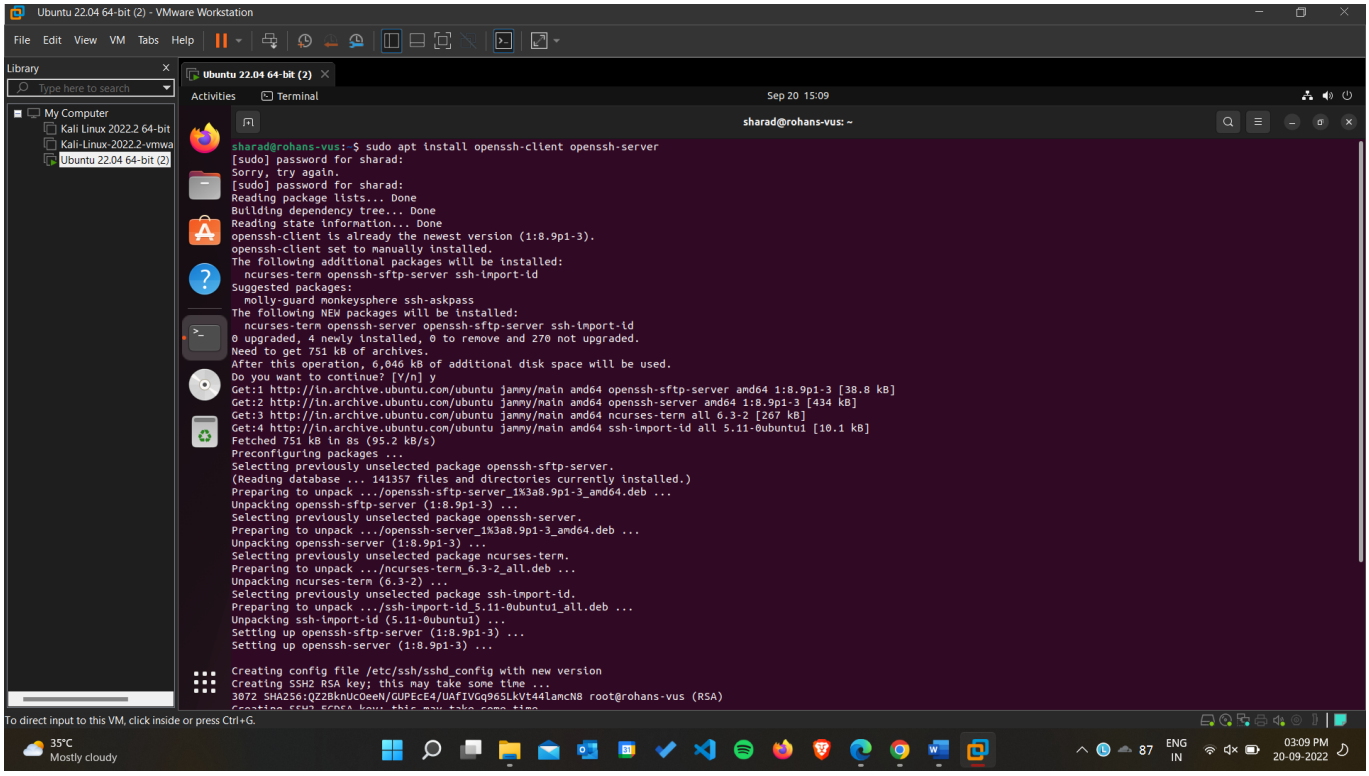
- For X forwarding in SSH to work, the personal computer must be running an X server program. The X server program manages the interaction between the remote application (the X client) and your computer's hardware.
- Most Linux distributions have the X server installed, but if the personal computer is running Windows or macOS, we will most likely need to install and run an X server application.
- Download and install Xming . For X forwarding to work, we'll need to start Xming before connecting to the remote system with your SSH client (for example, PuTTY).

Procedure & Screenshots:

Step 1: Launch local VM Instance (mine is VMWare Workstation Pro®)



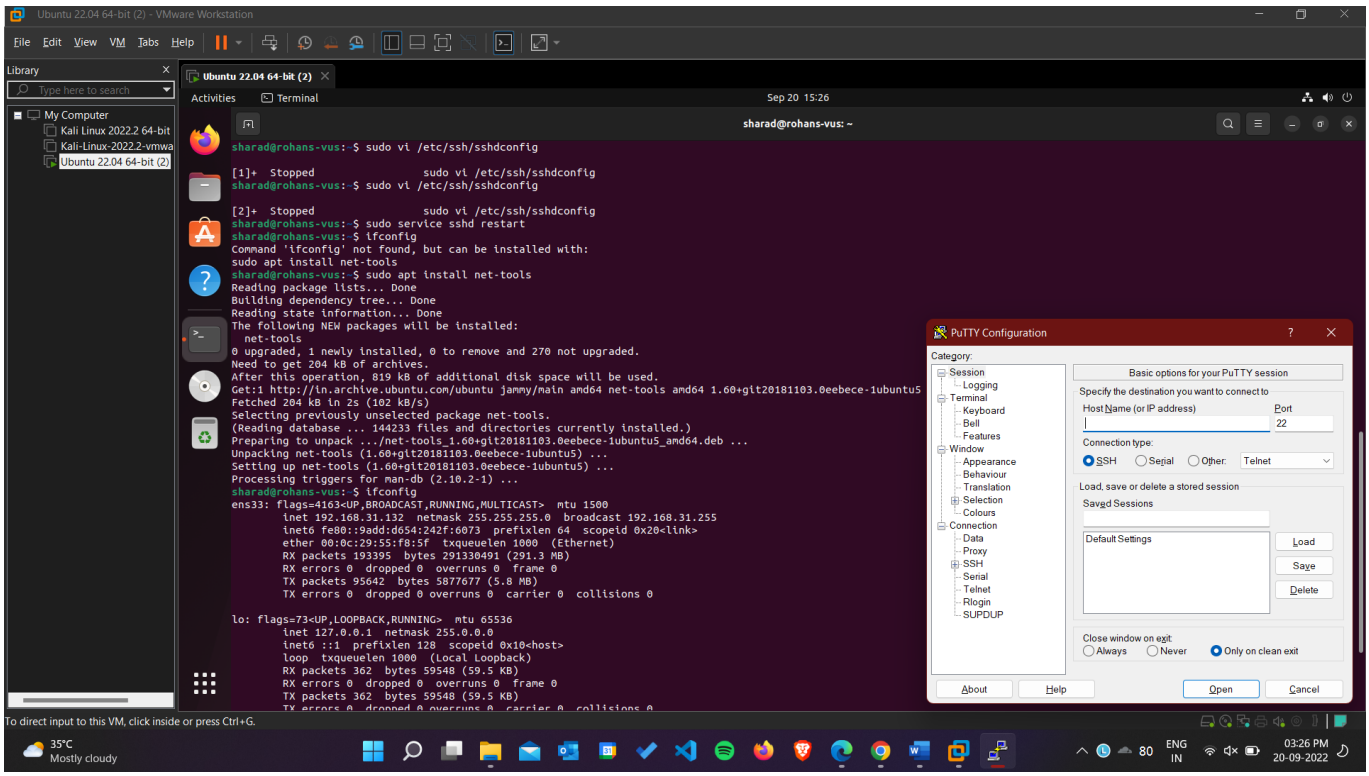
Step 2: Download and run Xming server: sudo apt install openssh-client openssh-server



Step 3: Open the file /etc/ssh/sshdconfig and check if X11Forwarding is enabled or not: sudo vi /etc/ssh/sshdconfig

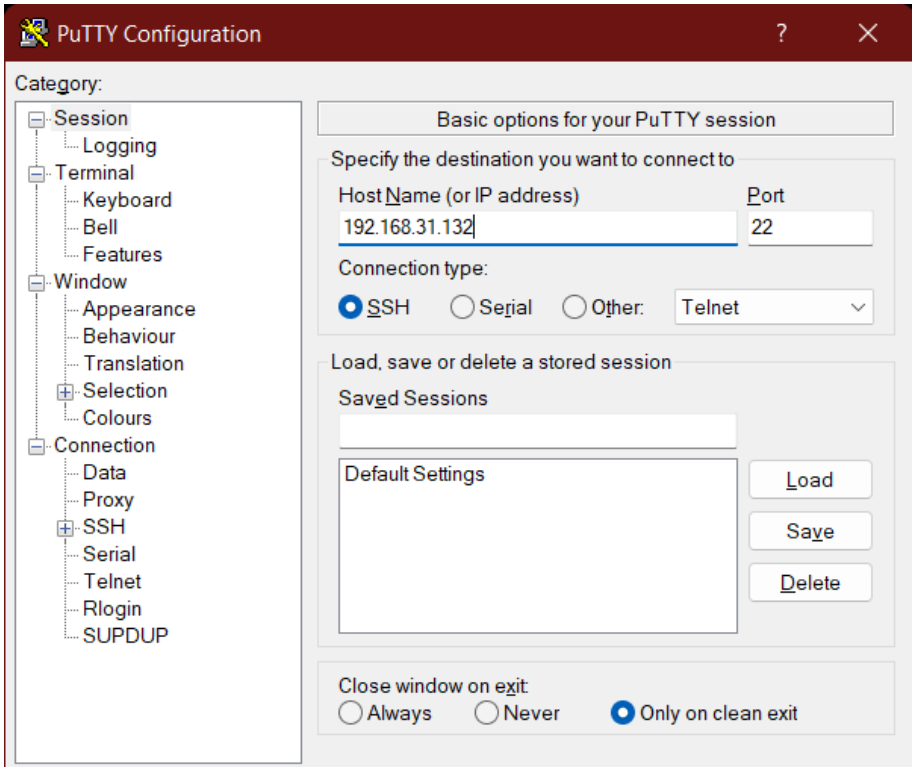
Step 4: Restart sshd service: sudo service sshd restart

Step 5: Find out the local VM instance IP using ifconfig command.

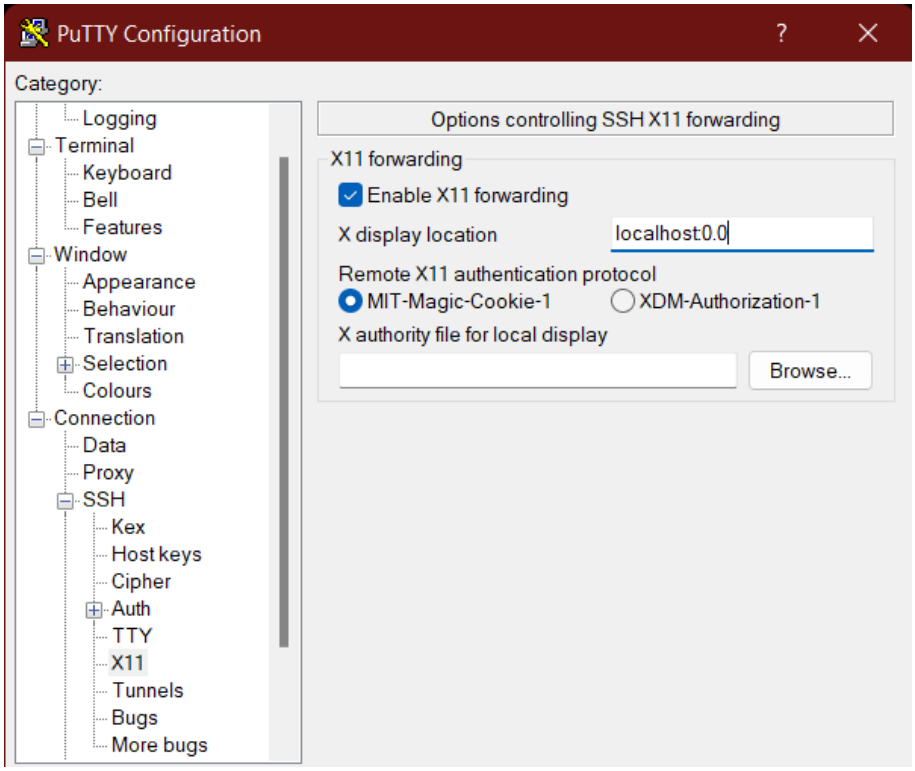


Using ifconfig, I've found my local VM's IP to be: 192.168.31.132

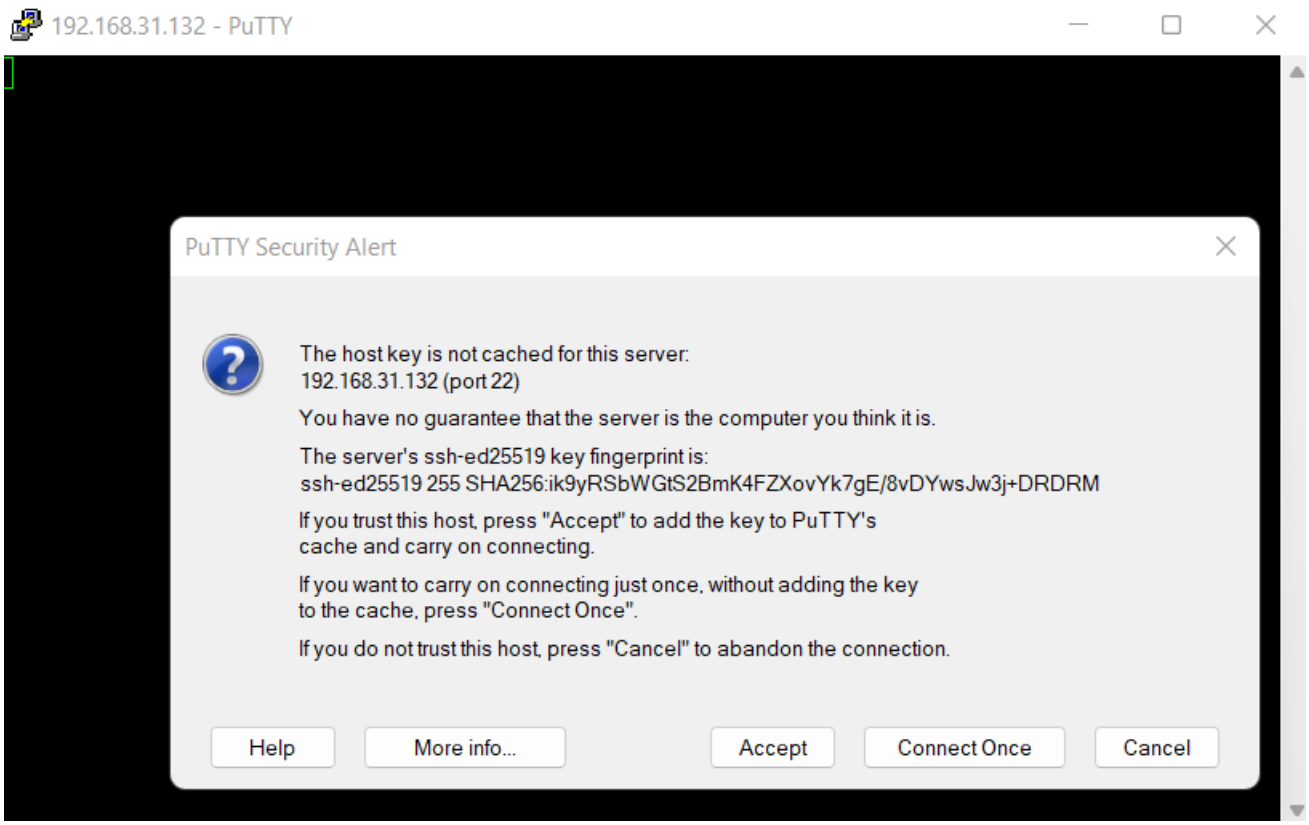
Step 6: Enter this local VM's IP (which is the host here) in PuTTY.



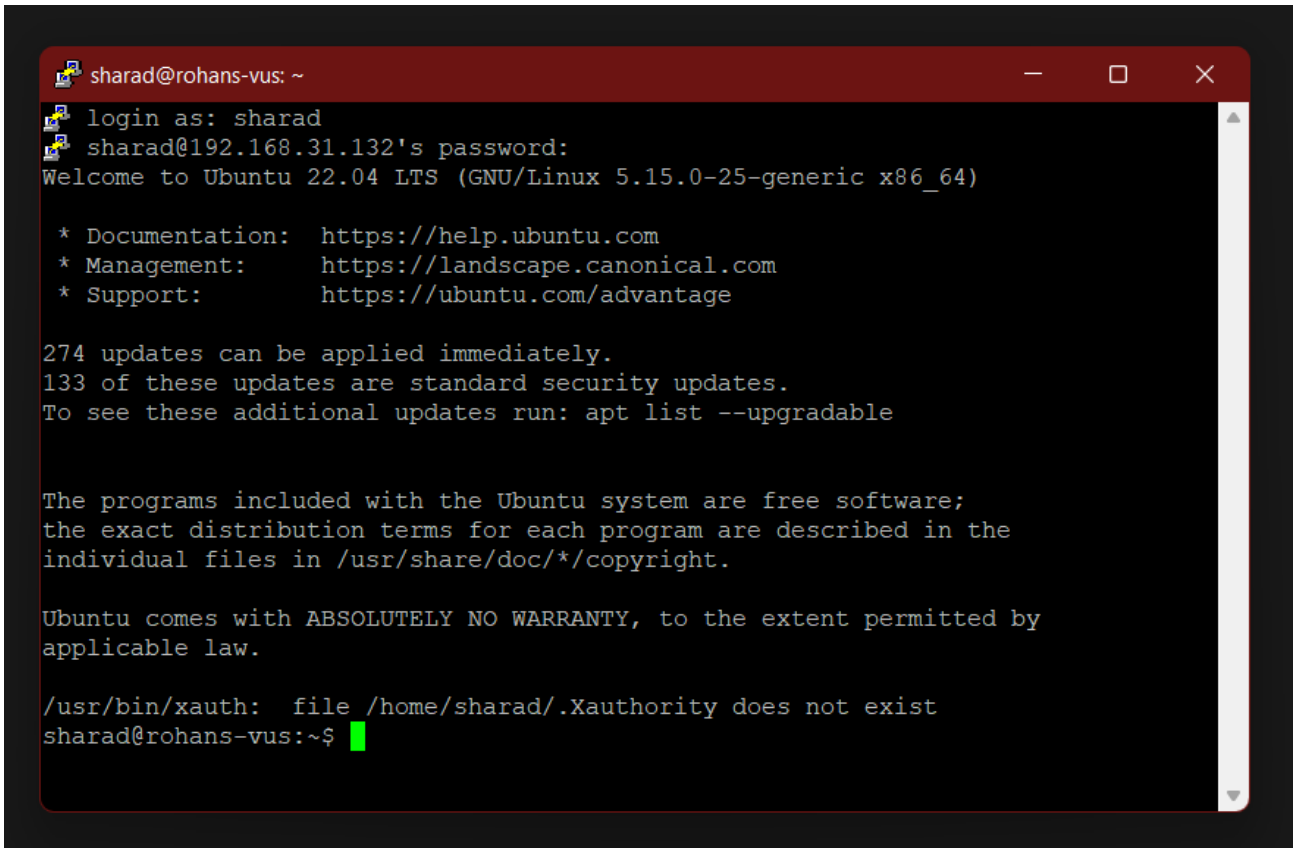
Step 7: Under "Connection→SSH→X11" menu, enable X11 forwarding and enter X display location.



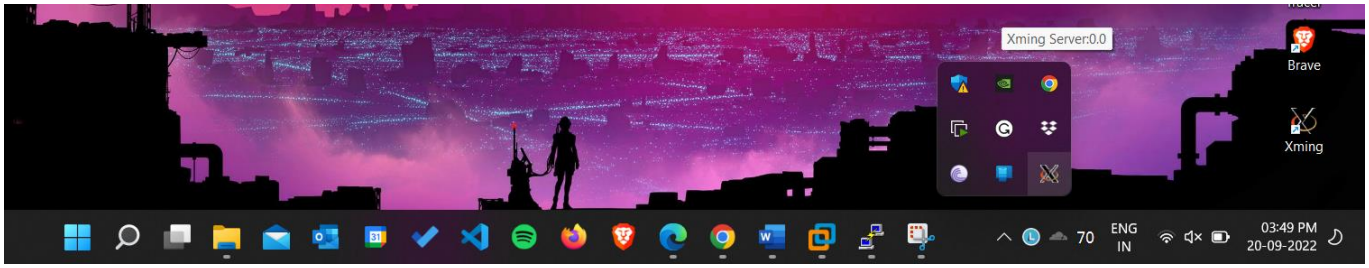
Step 8: Click on Open. Then 'Accept' the warning, since we trust this host.



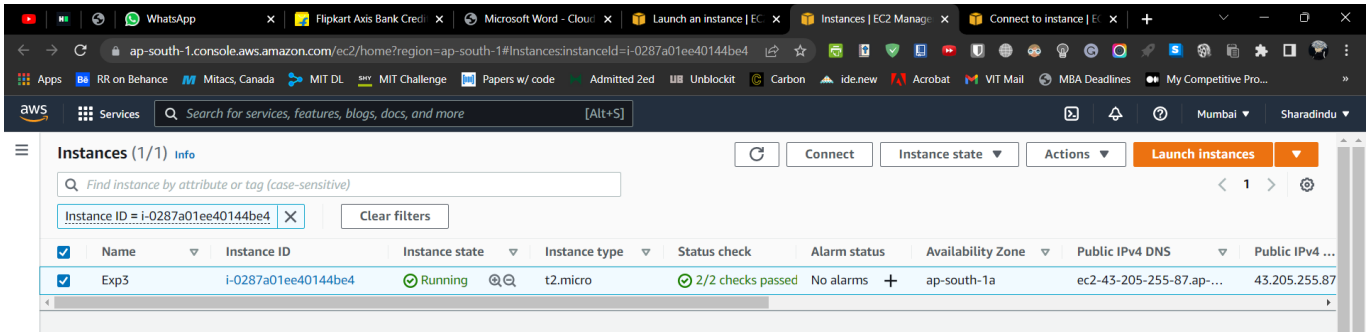
Step 9: Connect to the VM instance with X11 traffic forwarding enabled on the PuTTY. (login username: sharad (from sharad@rohans-vus))



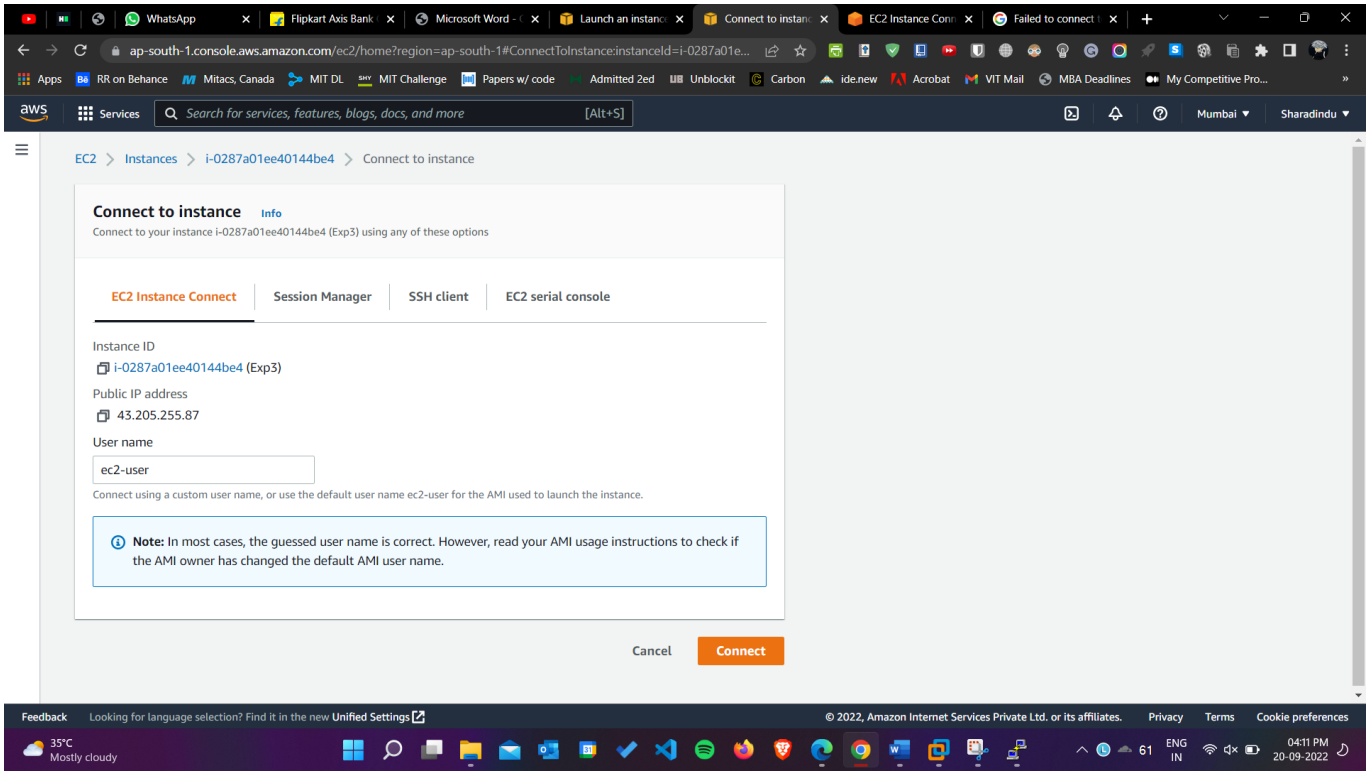
Step 10: Install Xming on local Windows, and have it run alongside PuTTY.



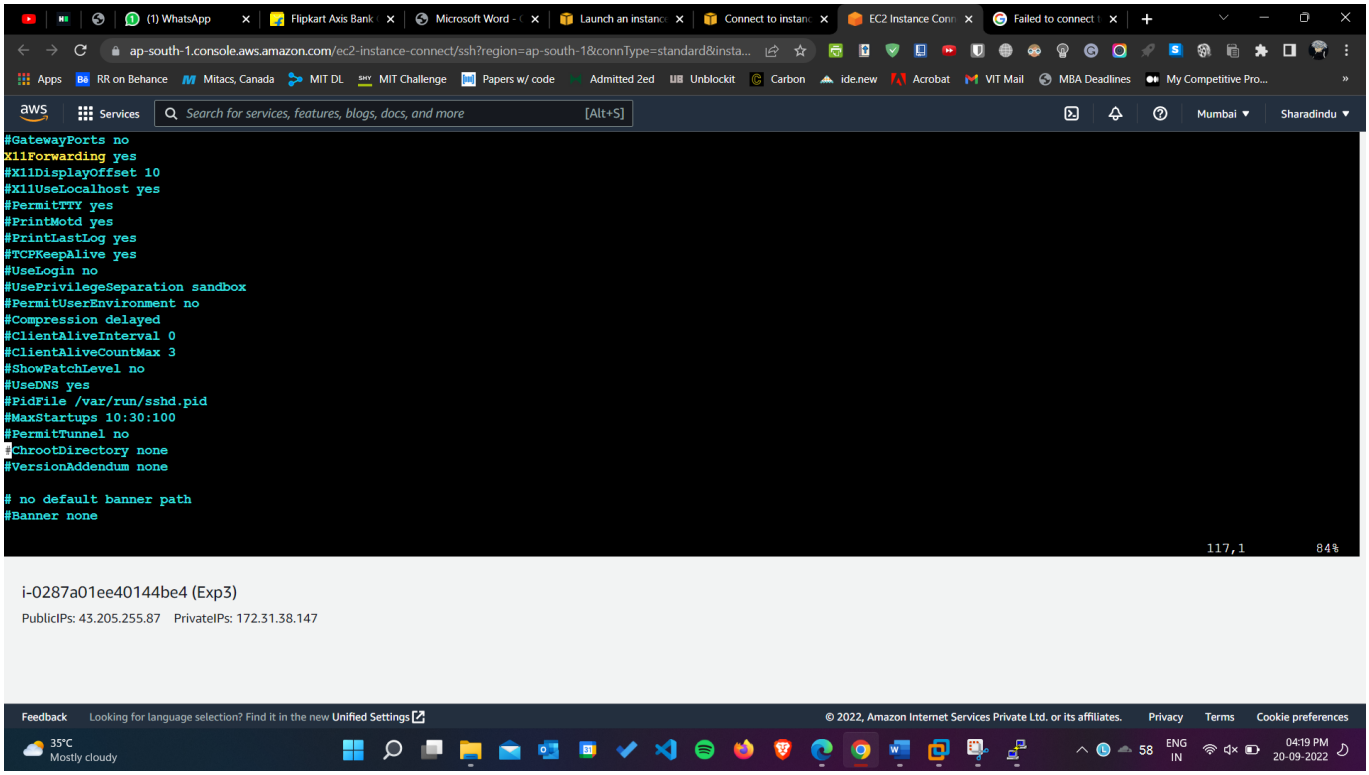
Step 11: Create a new Instance in AWS.



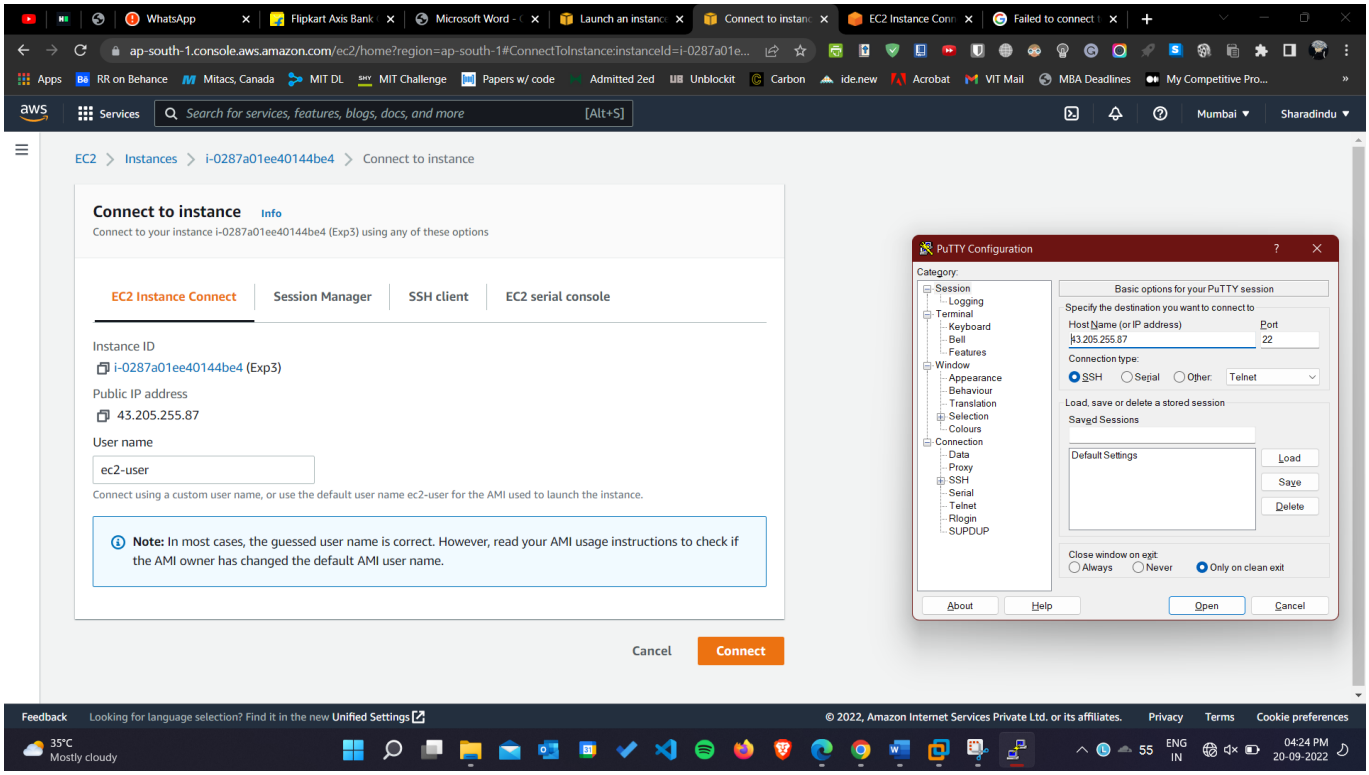
Step 12: Connect to instance. Use the default user name.



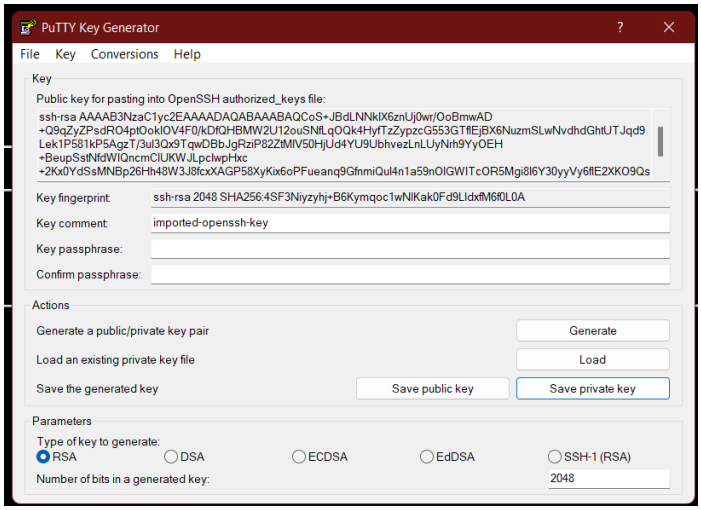
Step 13: In the console which appears thereafter, check whether X11 forwarding is enabled, using: sudo vi /etc/ssh/sshd_config



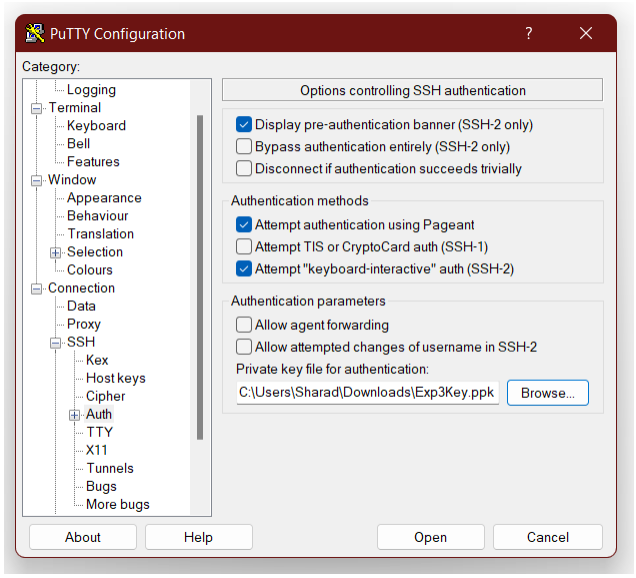
Step 14: Configure PuTTY with the public IP (43.205.255.87) in the EC2 instance.



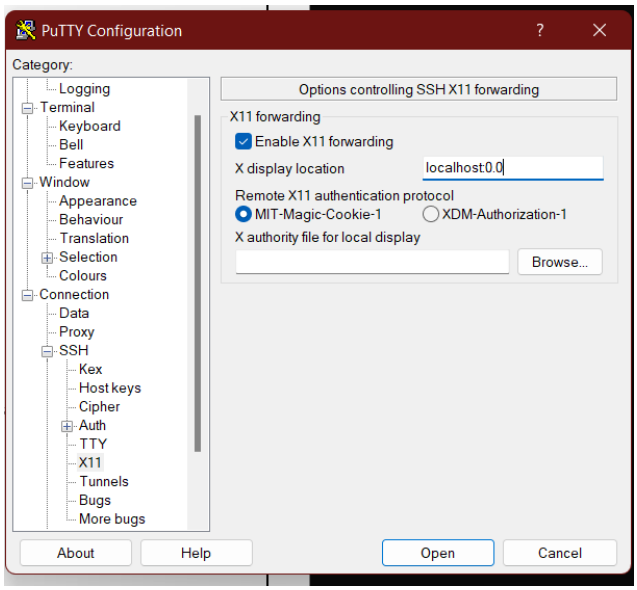
Step 15: Convert the private key from .pem to .ppk using PuTTYgen.



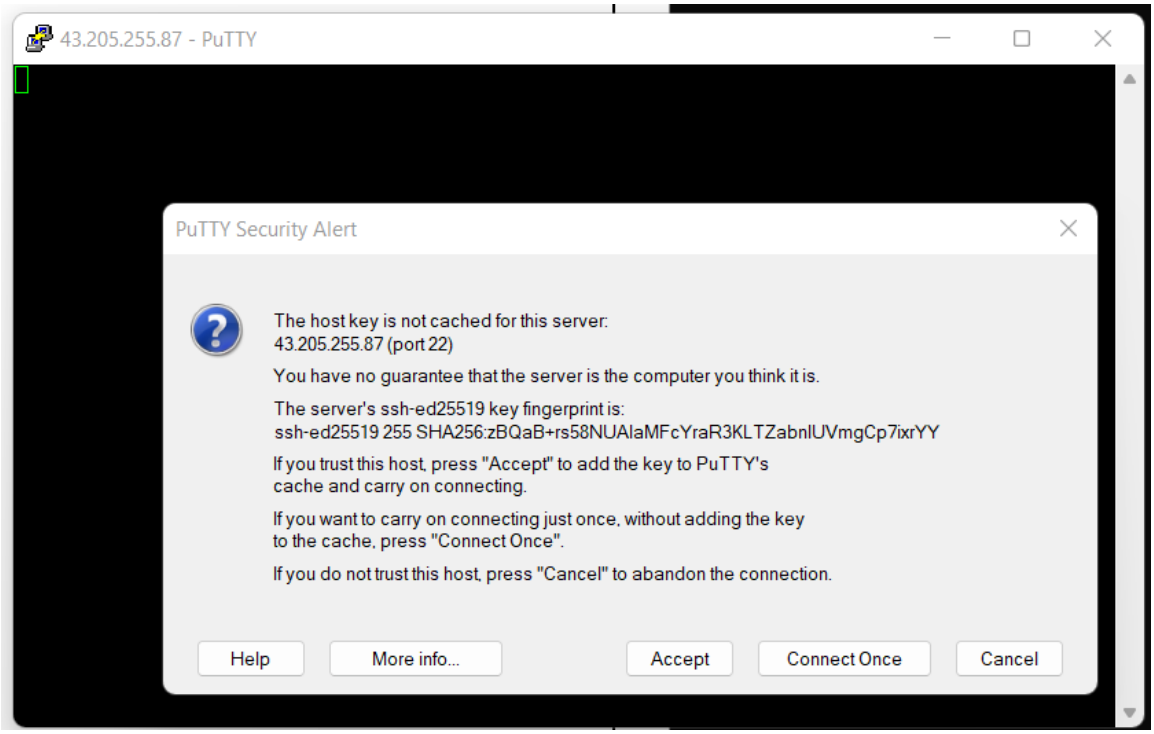
Step 16: Load the private key for the EC2 instance and go to X11 forwarding settings in PuTTY.



Step 17: In the “connection→ssh→auth→x11” menu, enable X11 forwarding and set X display location.

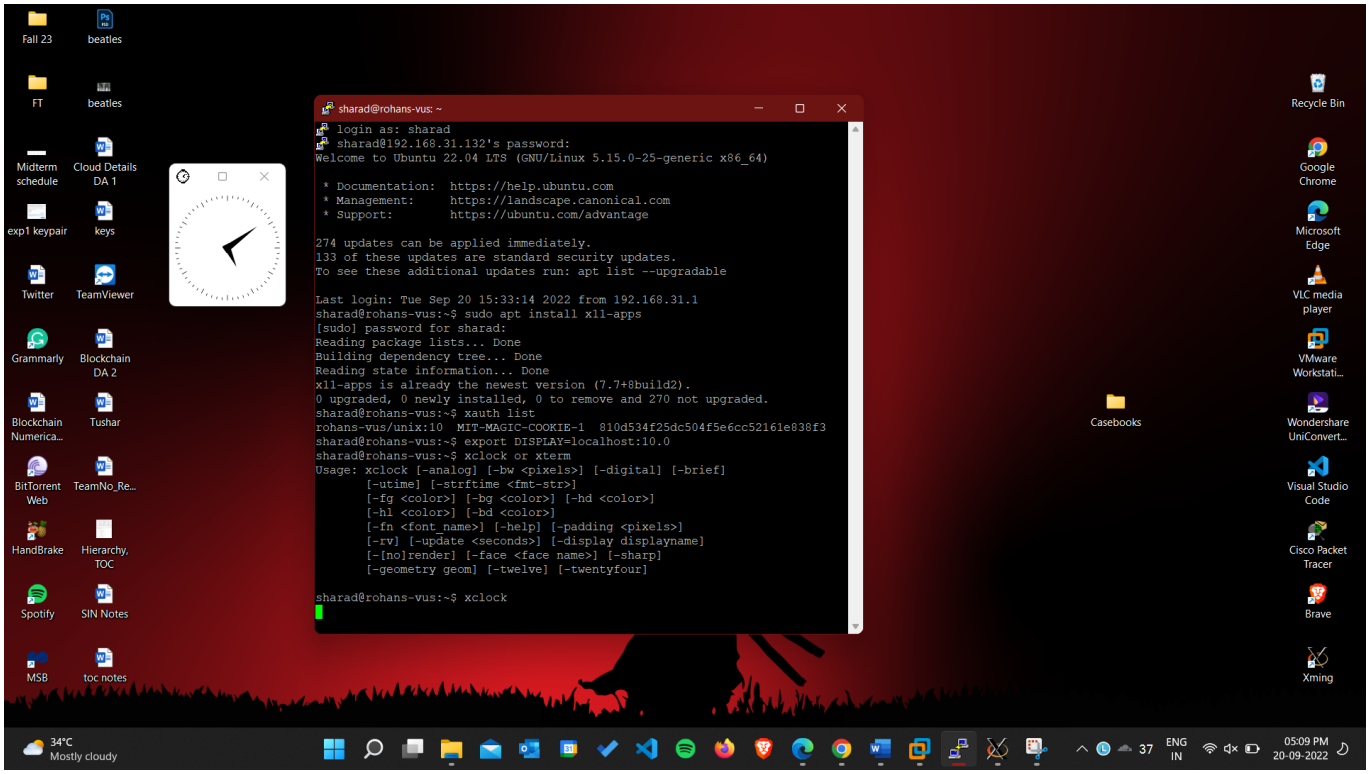


Step 18: Accept the warning, like earlier, since we trust this host.



Step 19: Connect to the EC2 instance and check if the X11 traffic forwarding is working using following commands:

xauth list
export DISPLAY=localhost:10.0
xclock or xterm



Experiment: 2

Title: Hosting Website on AWS

Aim:

- Create an SSH tunnel between your server in the local machine and remote clients in EC2 instances and test the connections with programs using X11 traffic.
- DaaS – Deployment of a basic web app and adding additional Functionality.

Background Theory:

- Amazon Web Services offers cloud web hosting solutions that provide businesses, non-profits, and governmental organizations with low-cost ways to deliver their websites and web applications. Whether you're looking for a marketing, rich-media, or ecommerce website, AWS offers a wide-range of website hosting options, and we'll help you select the one that is right for you.

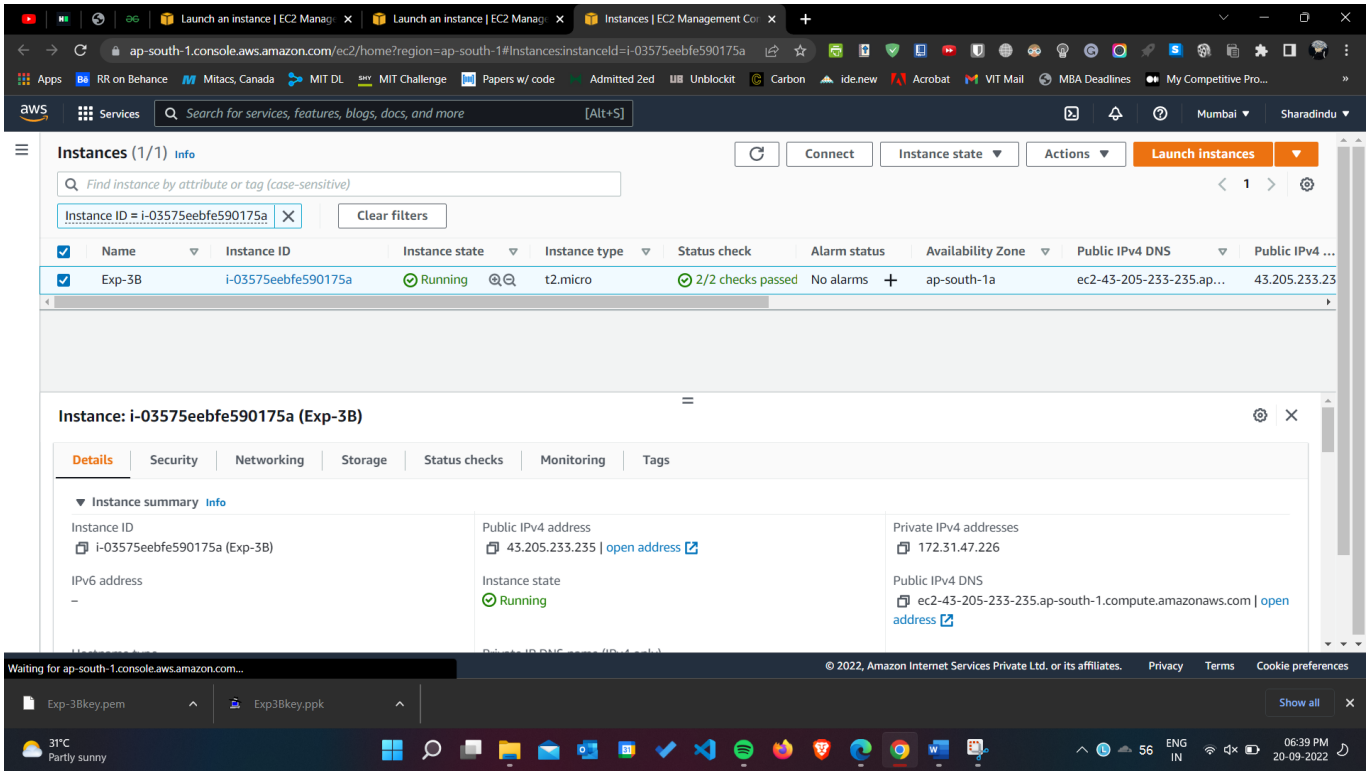
- Resizable compute capacity in the cloud --- EC2 can be applied to host websites that use multiple data centers, and for sites that need to scale using load balancing, autoscaling, or external databases.

Requirements for conducting the experiment:

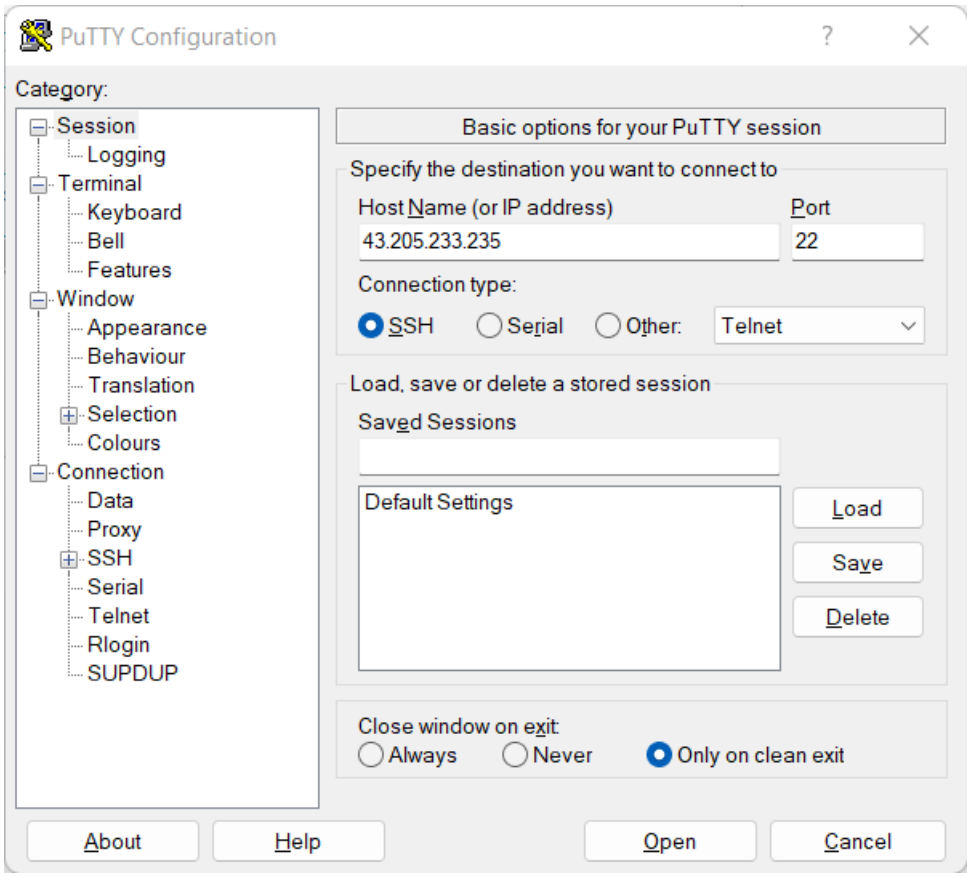
- AWS Licence, PuTTY

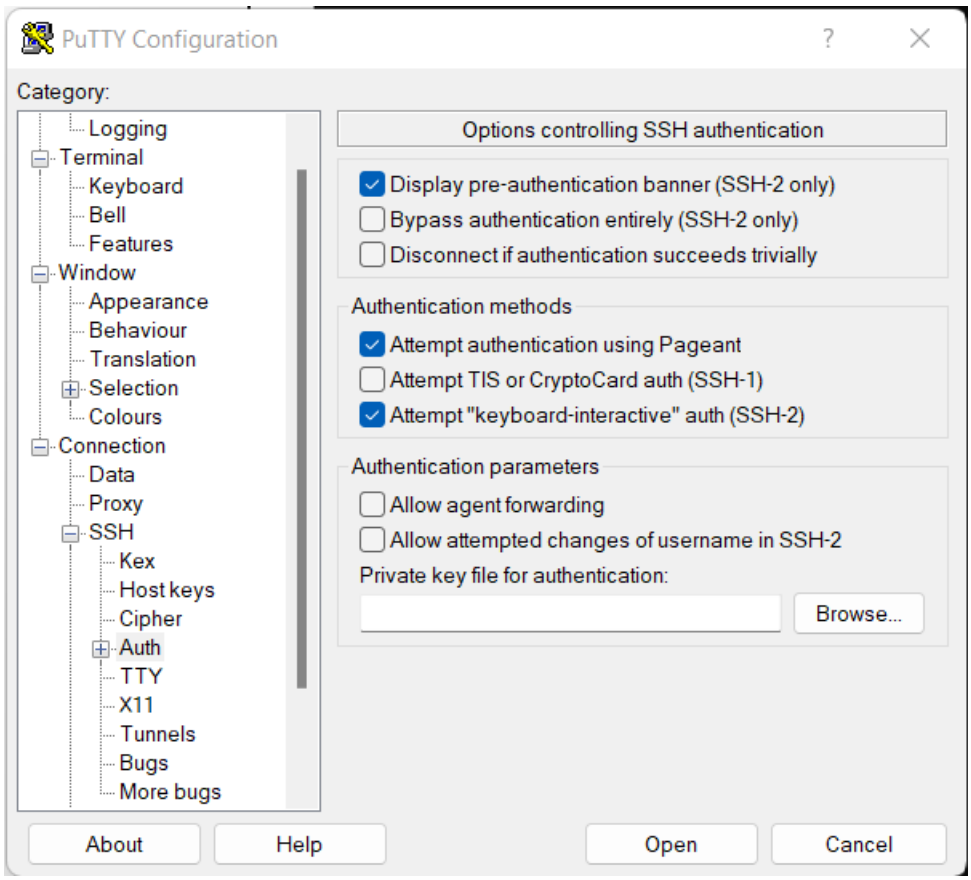
Procedure & Screenshots:

Step 1: Launch local VM Instance (mine is VMWare Workstation Pro®) and have the key pair saved as .ppk (since we're using PuTTY).

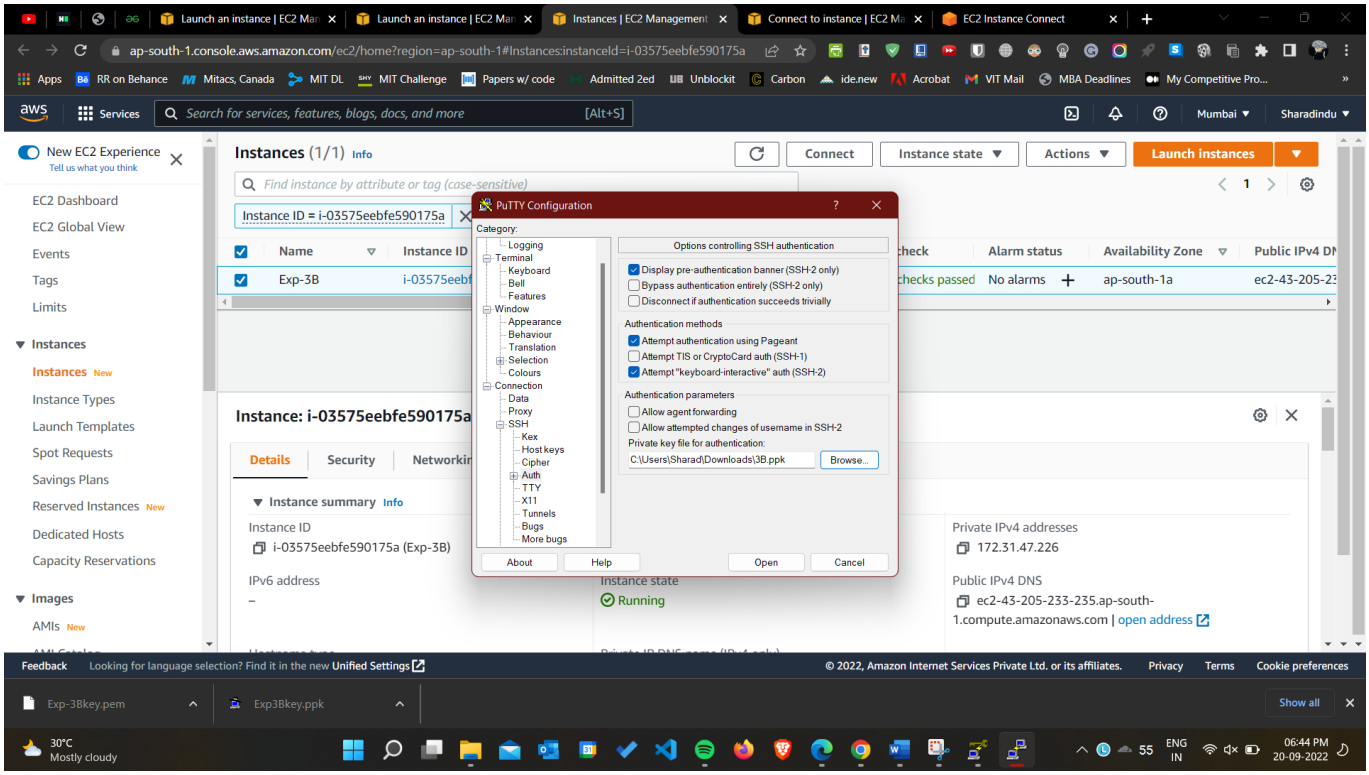


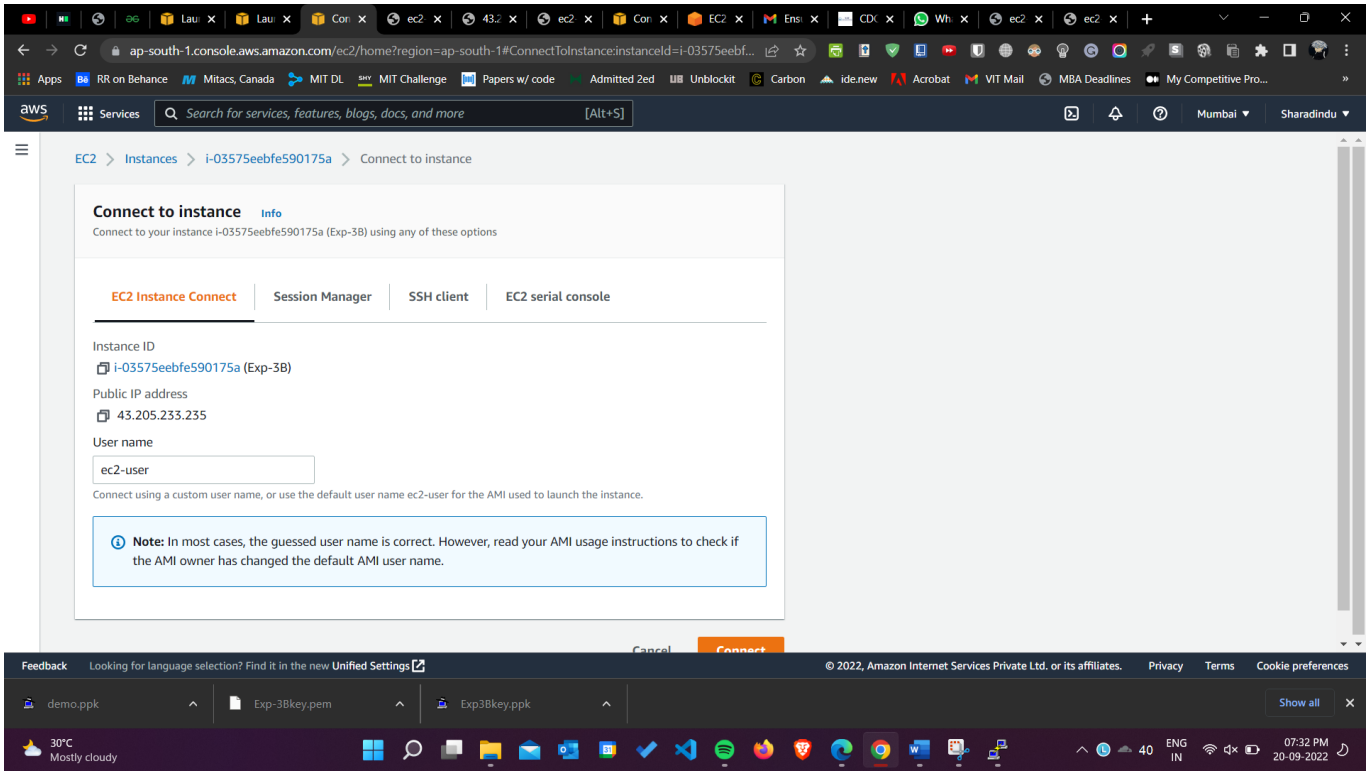
Step 2: Configure the PuTTY: Copy the public IPv4 address from AWS Instance and paste it to the Host Name of PuTTY.





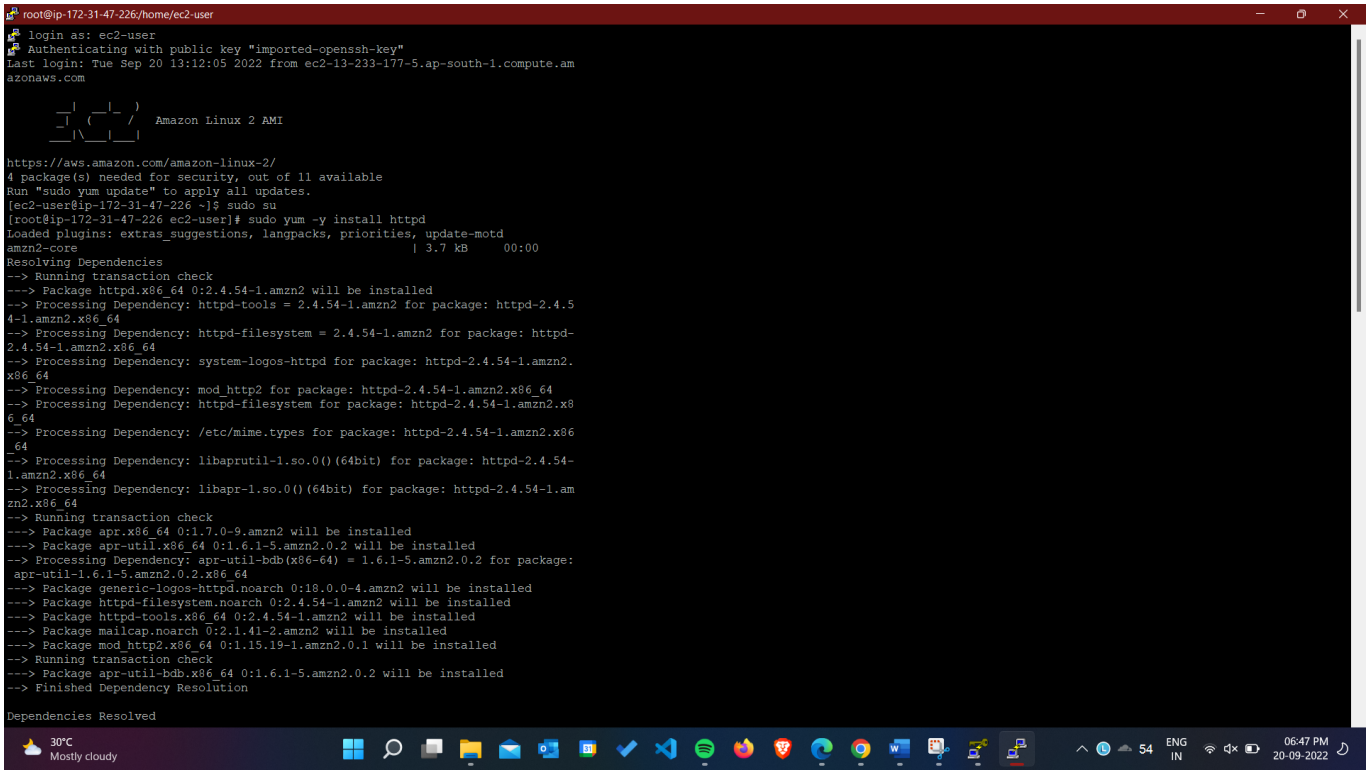
Step 3: Upload the converted RSA key pair (from .pem to .ppk) and upload in the 'Private key file for authentication' (in PuTTY and open it.)



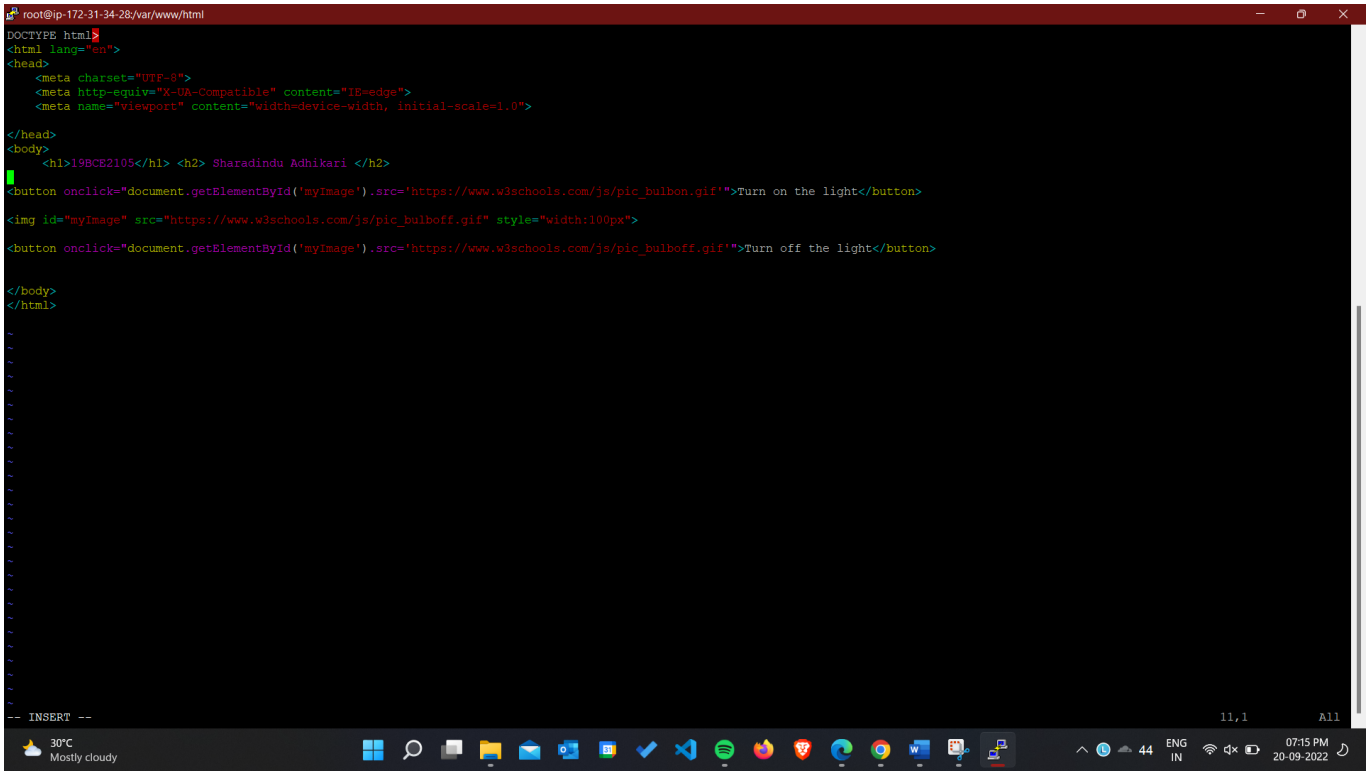


Step 4: Run the following commands in PuTTY:

- Enter command sudo su to gain root access.
- Installing the apache server using the command sudo yum –y install httpd.
- Start the apache server using the command sudo service httpd start.



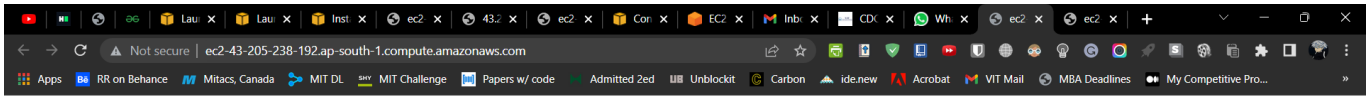
Step 5: Deploy the HTML file.



```
root@ip-172-31-34-28:/var/www/html
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Tue Sep 20 13:32:50 2022 from ec2-13-233-177-0.ap-south-1.compute.am
azonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-34-28 ~]$ sudo su
[root@ip-172-31-34-28 ec2-user]# sudo yum -y install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                               | 3.7 kB     00:00
Package httpd-2.4.54-1.amzn2.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-34-28 ec2-user]# sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-34-28 ec2-user]# sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-34-28 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-34-28 ec2-user]# cd /var/www/html
[root@ip-172-31-34-28 html]# l
bash: l: command not found
[root@ip-172-31-34-28 html]# ls
index.html  quit
[root@ip-172-31-34-28 html]# vi index.html
[root@ip-172-31-34-28 html]# cd /var/www/html
[root@ip-172-31-34-28 html]# vi index.html
[root@ip-172-31-34-28 html]# sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-34-28 html]#
```



DOCTYPE html>

19BCE2105

Sharadindu Adhikari

Turn on the light Turn off the light