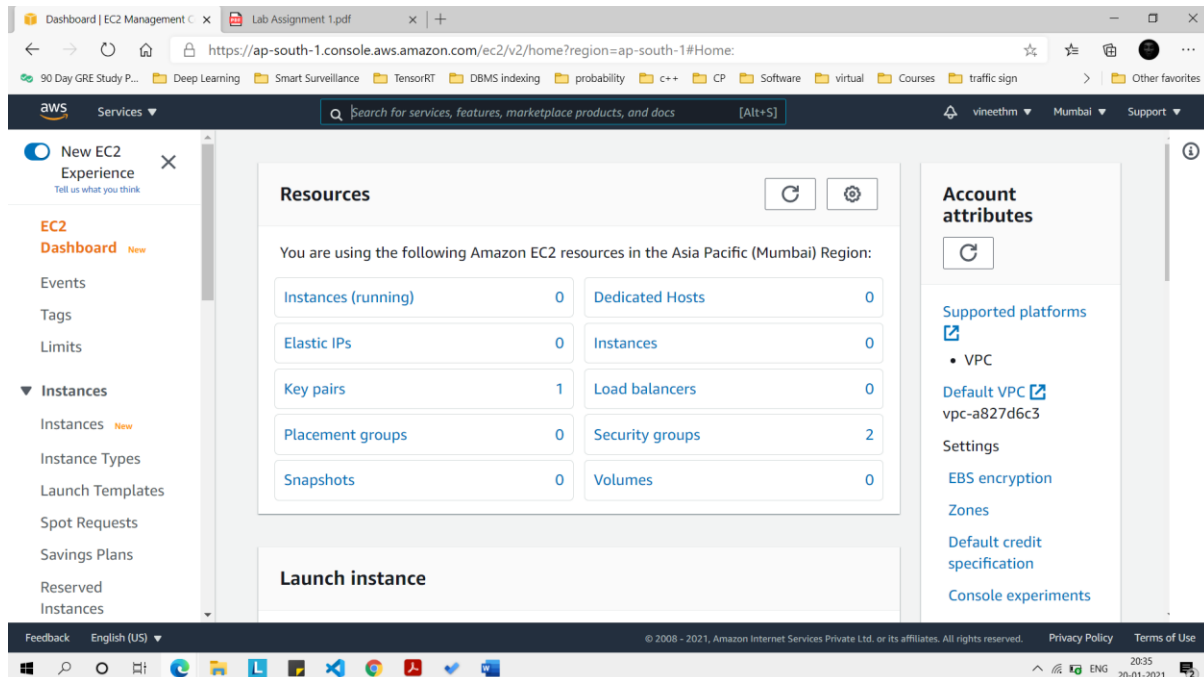


## Lab Assignment – 01

Task 1: Create AWS Free Tier personal account.



Task 2: Review the AWS services and Account, Billing, cost dashboard etc.

Explored all the above mentioned activities and spent 30 mins exploring them.

Task 3: Launch, connect and install EC2 IIS webserver and host a static HTML web page.

Reference link: <http://fredericpaladin.com/kb/aws-ec2-instance-with-windows-server-iis-web-server/>

Lab Assignment 1.pdf x AWS EC2 Instance with Windows x Launch instance wizard | EC2 M...

https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

90 Day GRE Study P... Deep Learning Smart Surveillance TensorRT DBMS indexing probability c++ CP Software virtual Courses traffic sign Other favorites

aws Services Search for services, features, marketplace products, and docs [Alt+S] vineethm Mumbai Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group  
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop

Add Rule

**Warning**

Cancel Previous **Review and Launch**

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Lab Assignment 1.pdf x AWS EC2 Instance with Windows x Launch instance wizard | EC2 M...

https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

90 Day GRE Study P... Deep Learning Smart Surveillance TensorRT DBMS indexing probability c++ CP Software virtual Courses traffic sign Other favorites

aws Services Search for services, features, marketplace products, and docs [Alt+S] vineethm Mumbai Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Warning** Improve your instances' security. Your security group, launch-wizard-2, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

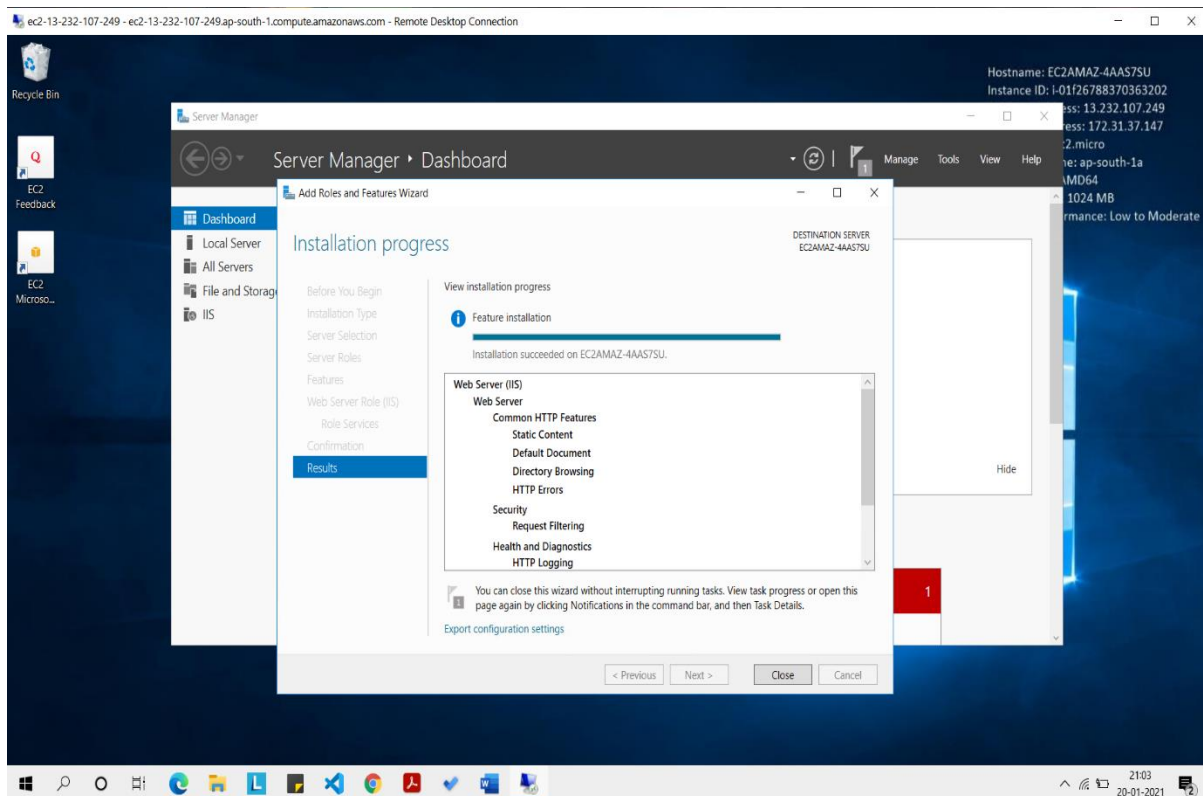
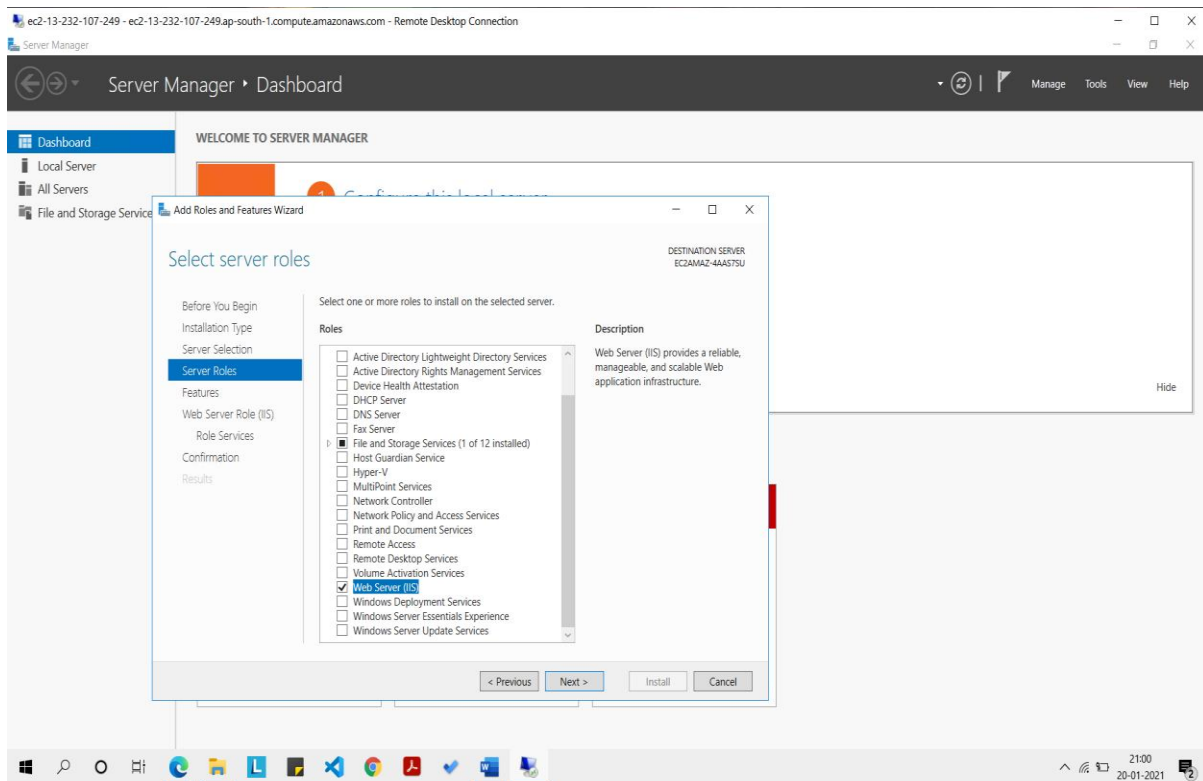
**Microsoft Windows Server 2016 Base - ami-0eb094e431a3874cc**  
 Free tier eligible Microsoft Windows 2016 Datacenter edition. [English]  
 Root Device Type: ebs Virtualization type: hvm  
 If you plan to use this AMI for an application that benefits from Microsoft License Mobility, fill out the [License Mobility Form](#). Don't show me this again

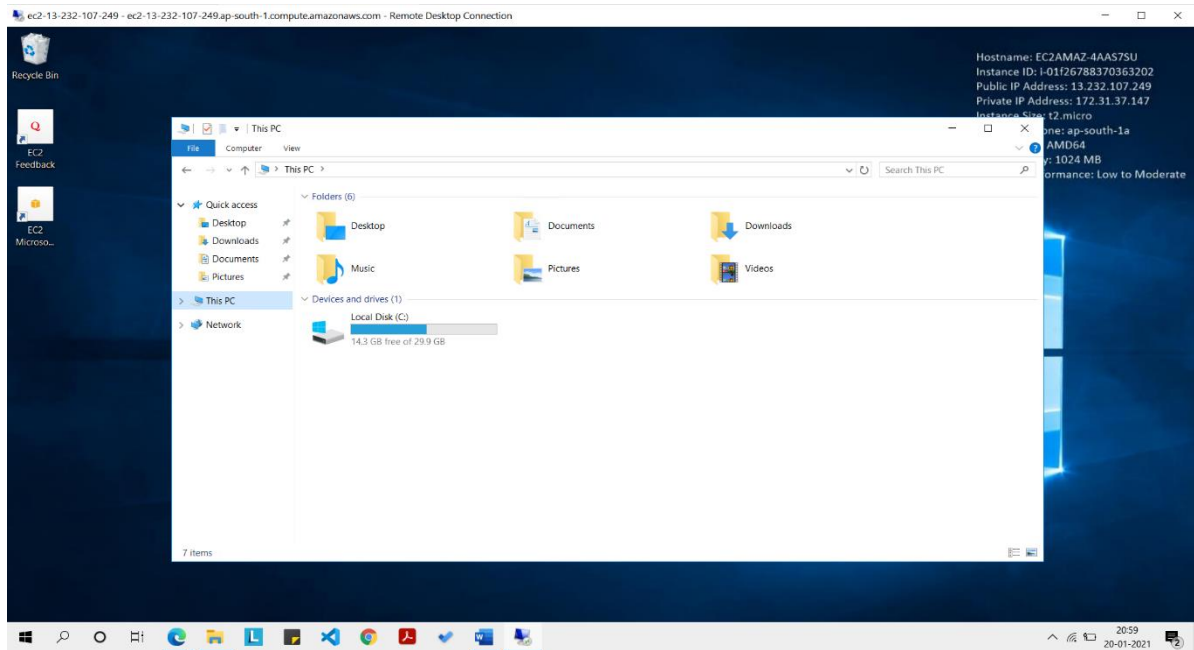
Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

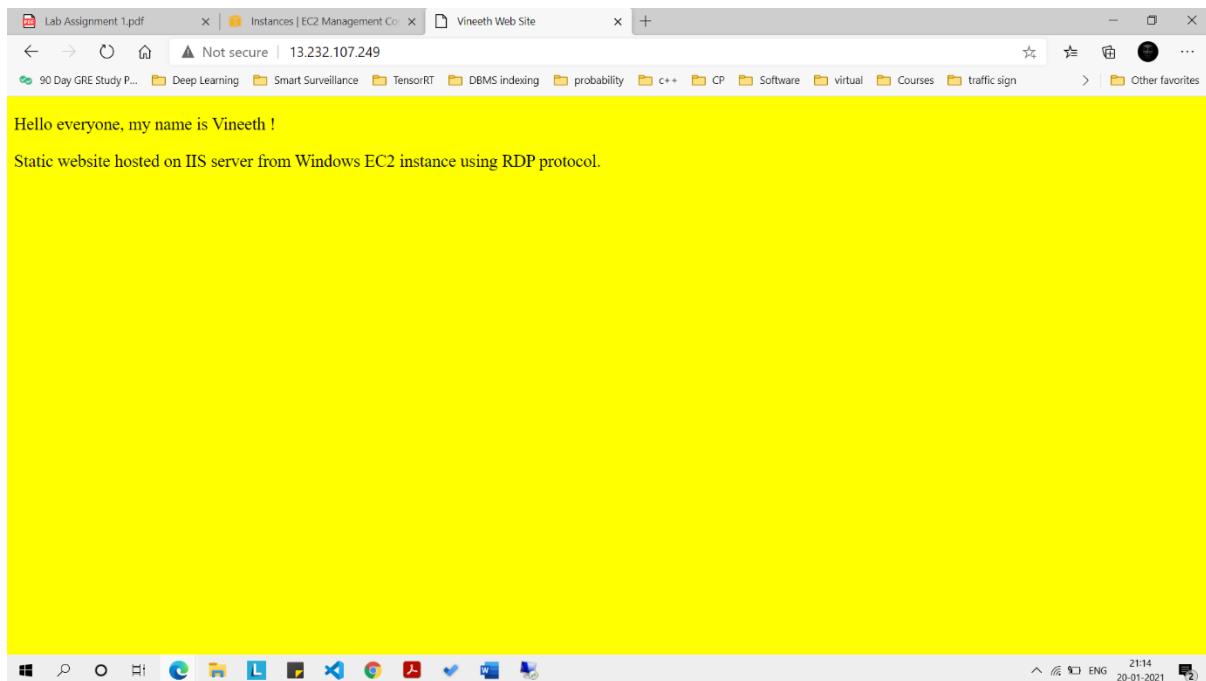
Cancel Previous **Launch**

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

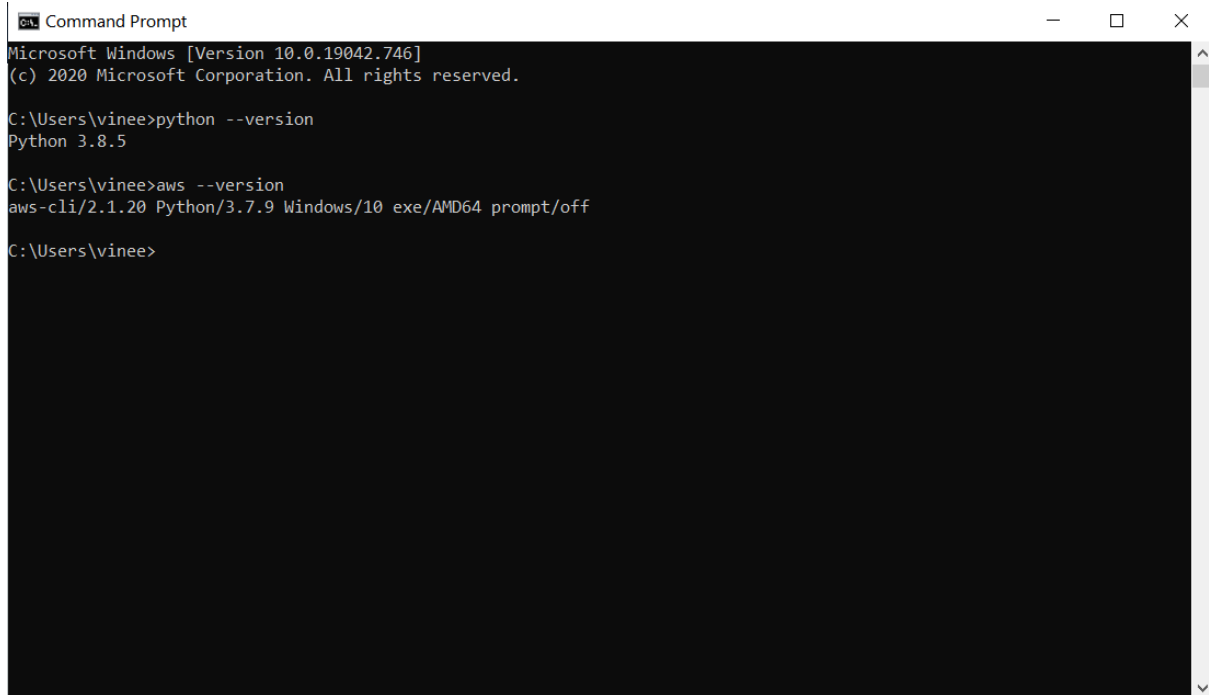




Hosting of the static webiste in the IIS server and accessing it via the public ipv4 address.



## Task 4: Working with AWS command line and perform start, stop, and terminate EC2 instance using the CLI.

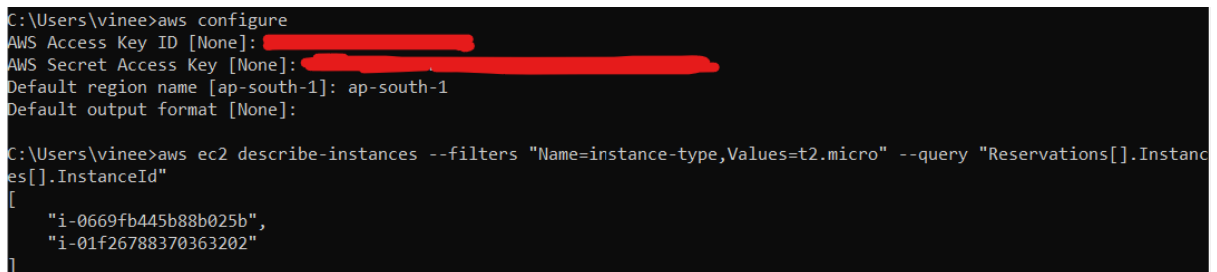


```
Command Prompt
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\vinee>python --version
Python 3.8.5

C:\Users\vinee>aws --version
aws-cli/2.1.20 Python/3.7.9 Windows/10 exe/AMD64 prompt/off

C:\Users\vinee>
```



```
C:\Users\vinee>aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [ap-south-1]: ap-south-1
Default output format [None]: 

C:\Users\vinee>aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query "Reservations[].Instances[].InstanceId"
[
  "i-0669fb445b88b025b",
  "i-01f26788370363202"
]
```

## Terminating the Windows EC2 instance using AWS CLI.



```
Command Prompt
C:\Users\vinee>aws ec2 terminate-instances --instance-ids i-01f26788370363202
{
  "TerminatingInstances": [
    {
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "InstanceId": "i-01f26788370363202",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

## Confirmation.

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check
<input type="checkbox"/>	Windows...	i-0669fb445b88b025b	Termina... 🔍	t2.micro	—
<input type="checkbox"/>	Windows	i-01f26788370363202	Termina... 🔍	t2.micro	—

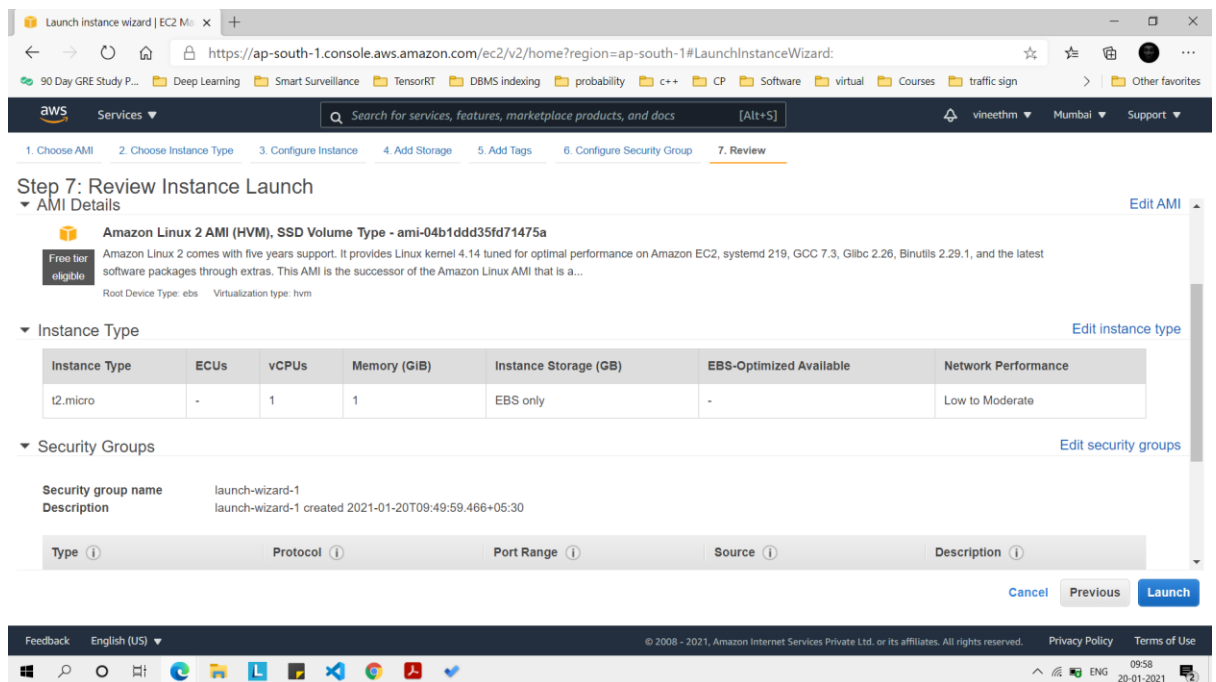
Task 6: Launch the AWS Linux instance and connect it using PuTTY.

Reference article:

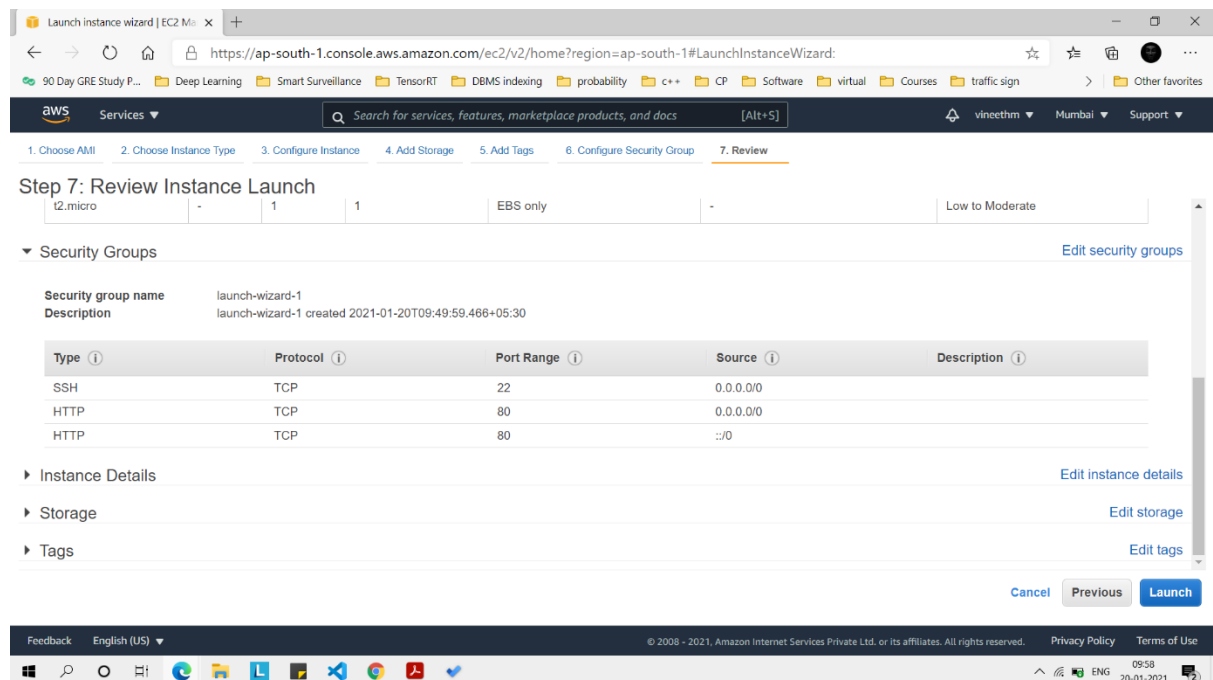
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

Screenshots:

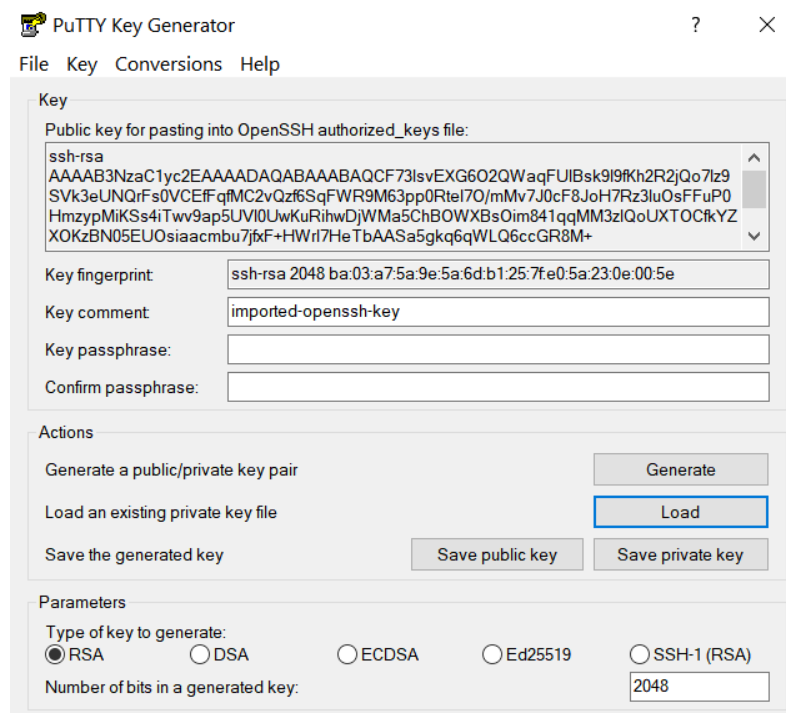
## 1) Creation of Amazon EC2 Linux instance.



## 2) Add the HTTP protocol to access the webpage (apache server welcome page)

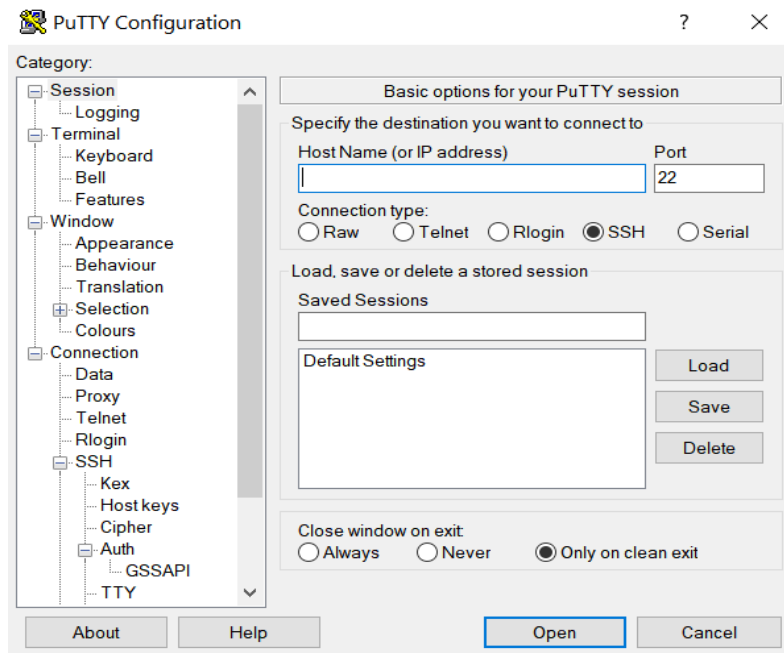


## 3) Convert the private key file (.pem) to (.ppk) using PuTTYgen.

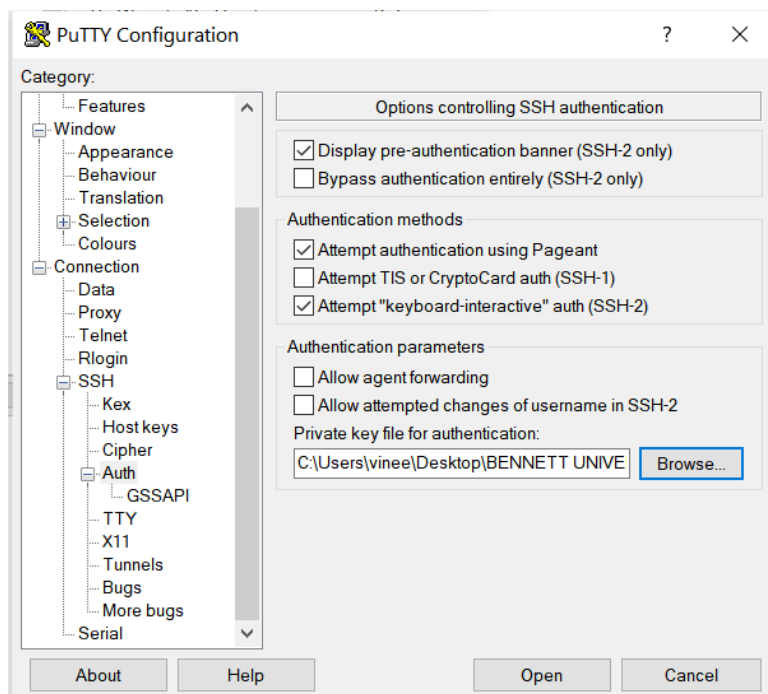




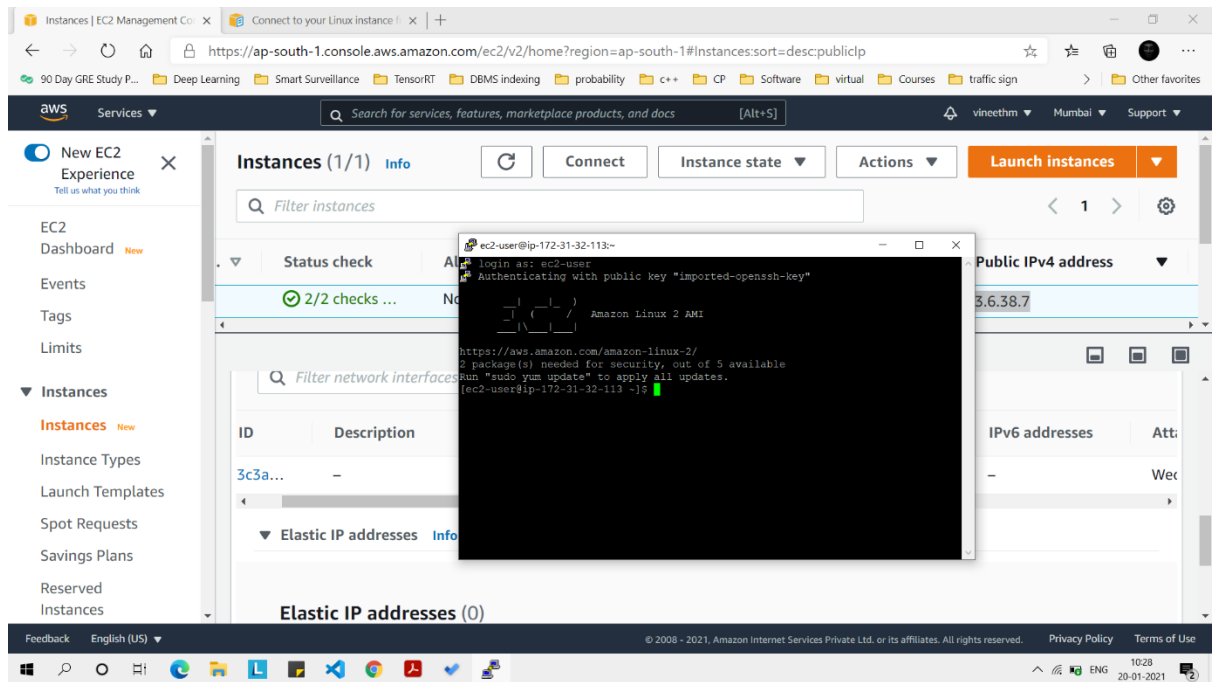
#### 4) Open the PuTTY and copy the host address (public ipv4)



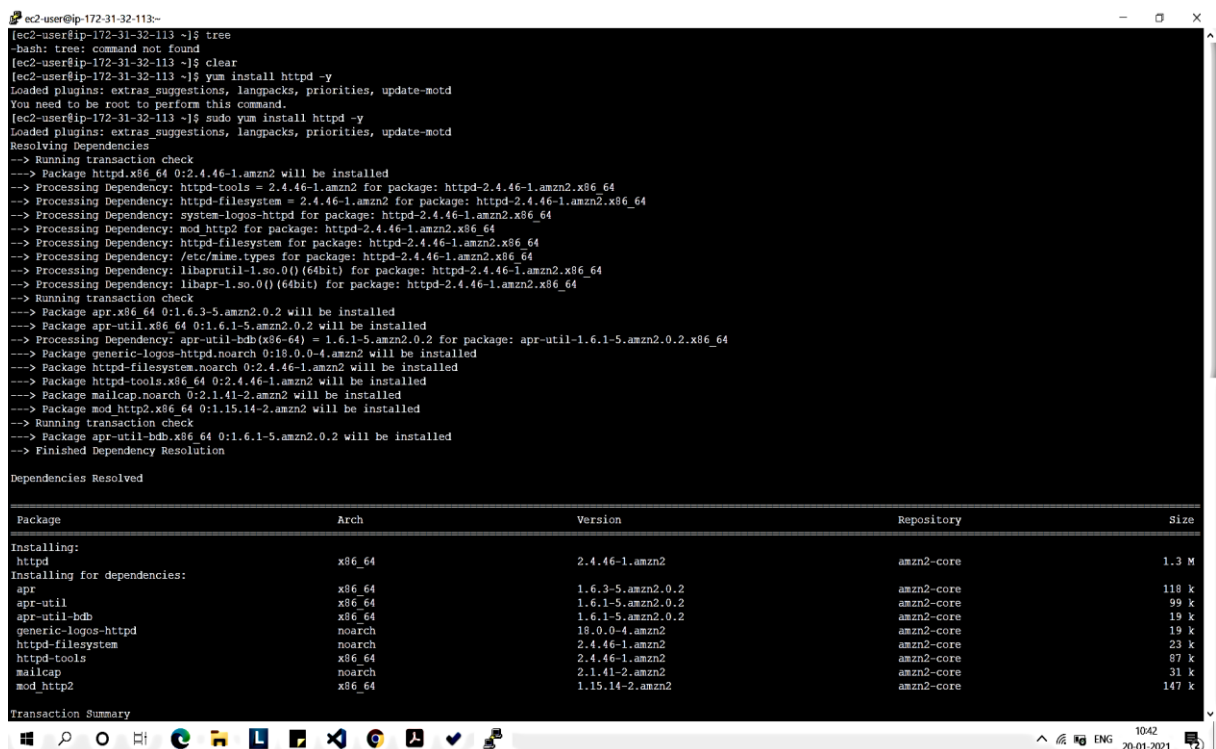
#### 5) In the SSH, under Auth load the key(.ppk file)



6) Now the connection to the Linux server is established.



7) Run some commands on the linux terminal.



```
ec2-user@ip-172-31-32-113:~  
[ec2-user@ip-172-31-32-113 ~]$ iostat  
Linux 4.14.209-160.339.amzn2.x86_64 (ip-172-31-32-113.ap-south-1.compute.internal) 01/20/2021 _x86_64_ (1 CPU)  
  
avg-cpu:  %user   %nice %system %iowait  %steal   %idle  
           1.09    0.00    0.22    0.05    0.37   98.26  
  
Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn  
xvda                 4.46         90.19        126.55      194156      272418  
[ec2-user@ip-172-31-32-113 ~]$
```

```
ec2-user@ip-172-31-32-113:~  
[ec2-user@ip-172-31-32-113 ~]$ uptime  
05:07:10 up 34 min,  1 user,  load average: 0.16, 0.06, 0.02  
[ec2-user@ip-172-31-32-113 ~]$
```

```

ec2-user@ip-172-31-32-113:~
[ec2-user@ip-172-31-32-113 ~]$ sudo apt-get update
sudo: apt-get: command not found
[ec2-user@ip-172-31-32-113 ~]$ clear
[ec2-user@ip-172-31-32-113 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                               | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package chrony.x86_64 0:3.5.1-1.amzn2.0.1 will be updated
--> Package chrony.x86_64 0:3.5.1-1.amzn2.0.1 will be an update
--> Processing Dependency: libnettle.so.4()(64bit) for package: chrony-3.5.1-1.amzn2.0.1.x86_64
--> Package cloud-init.noarch 0:19.3-4.amzn2 will be updated
--> Package cloud-init.noarch 0:19.3-4.amzn2 will be an update
--> Package p11-kit.x86_64 0:0.23.22-1.amzn2.0.1 will be updated
--> Package p11-kit.x86_64 0:0.23.22-1.amzn2.0.1 will be an update
--> Package p11-kit-trust.x86_64 0:0.23.22-1.amzn2.0.1 will be updated
--> Package p11-kit-trust.x86_64 0:0.23.22-1.amzn2.0.1 will be an update
--> Package tzdata.noarch 0:2020a-1.amzn2 will be updated
--> Package tzdata.noarch 0:2020a-1.amzn2 will be an update
--> Running transaction check
--> Package nettle.x86_64 0:2.7.1-8.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package Arch Version Repository Size
-----
Updating:
chrony x86_64 3.5.1-1.amzn2.0.1 amzn2-core 258 k
cloud-init noarch 19.3-4.amzn2 amzn2-core 924 k
p11-kit x86_64 0.23.22-1.amzn2.0.1 amzn2-core 321 k
p11-kit-trust x86_64 0.23.22-1.amzn2.0.1 amzn2-core 130 k
tzdata noarch 2020a-1.amzn2 amzn2-core 481 k
Installing for dependencies:
nettle x86_64 2.7.1-8.amzn2.0.2 amzn2-core 329 k

Transaction Summary
-----
Install ( 1 Dependent package)
Upgrade 5 Packages

Total download size: 2.4 M
Is this ok [y/d/n]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/6): chrony-3.5.1-1.amzn2.0.1.x86_64.rpm | 258 kB 00:00:00
(2/6): cloud-init-19.3-4.amzn2.noarch.rpm | 924 kB 00:00:00
(3/6): nettle-2.7.1-8.amzn2.0.2.x86_64.rpm | 329 kB 00:00:00
(4/6): p11-kit-0.23.22-1.amzn2.0.1.x86_64.rpm | 321 kB 00:00:00

```

```

ec2-user@ip-172-31-32-113:~
* Authenticating with public key "imported-openssh-key"

  ____  _
 / ___|| | | |
| |___| | | |
 \___ \| | | |
      | | | |
      |_|_|_|

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 5 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-32-113 ~]$ ls
[ec2-user@ip-172-31-32-113 ~]$ sudo ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.5 125504 5380 ?        Ss   04:32   0:01 /usr/lib/systemd/systemd --switched-root --system --deserialize 21
root         2  0.0  0.0  0 0 ?        S   04:32   0:00 [kthreadd]
root         4  0.0  0.0  0 0 ?        I<  04:32   0:00 [kworker/0:0H]
root         5  0.0  0.0  0 0 ?        I   04:32   0:00 [kworker/u30:0]
root         6  0.0  0.0  0 0 ?        I<  04:32   0:00 [mm_percpu_wq]
root         7  0.0  0.0  0 0 ?        S   04:32   0:00 [ksftirqd/0]
root         8  0.0  0.0  0 0 ?        I   04:32   0:00 [rcu_sched]
root         9  0.0  0.0  0 0 ?        I   04:32   0:00 [rcu_bh]
root        10  0.0  0.0  0 0 ?        S   04:32   0:00 [migration/0]
root        11  0.0  0.0  0 0 ?        S   04:32   0:00 [watchdog/0]
root        12  0.0  0.0  0 0 ?        S   04:32   0:00 [cpuhp/0]
root        14  0.0  0.0  0 0 ?        S   04:32   0:00 [kdevtmpfs]
root        15  0.0  0.0  0 0 ?        I<  04:32   0:00 [netns]
root        21  0.0  0.0  0 0 ?        S   04:32   0:00 [xenbus]
root        22  0.0  0.0  0 0 ?        S   04:32   0:00 [xennetctl]
root        176  0.0  0.0  0 0 ?        S   04:32   0:00 [khungtaskd]
root        177  0.0  0.0  0 0 ?        S   04:32   0:00 [oom_reaper]
root        178  0.0  0.0  0 0 ?        I<  04:32   0:00 [writeback]
root        180  0.0  0.0  0 0 ?        S   04:32   0:00 [kcompactd0]
root        181  0.0  0.0  0 0 ?        SN   04:32   0:00 [kamd]
root        182  0.0  0.0  0 0 ?        SN   04:32   0:00 [khugepaged]
root        183  0.0  0.0  0 0 ?        I<  04:32   0:00 [crypto]
root        184  0.0  0.0  0 0 ?        I<  04:32   0:00 [kintegrityd]
root        186  0.0  0.0  0 0 ?        I<  04:32   0:00 [khlockd]
root        524  0.0  0.0  0 0 ?        I<  04:32   0:00 [md]
root        535  0.0  0.0  0 0 ?        I   04:32   0:00 [kworker/u30:2]
root        543  0.0  0.0  0 0 ?        I<  04:32   0:00 [edac-poller]
root        548  0.0  0.0  0 0 ?        I<  04:32   0:00 [watchdogd]
root        689  0.0  0.0  0 0 ?        S   04:32   0:00 [ksuited]
root        695  0.0  0.0  0 0 ?        S   04:32   0:00 [kavapd0]
root        827  0.0  0.0  0 0 ?        I<  04:32   0:00 [kthrotld]
root        877  0.0  0.0  0 0 ?        I<  04:32   0:00 [katrip]
root        905  0.0  0.0  0 0 ?        I<  04:32   0:00 [ip6v_addrconf]
root       1346  0.0  0.0  0 0 ?        I<  04:32   0:00 [ata Sff]
root       1552  0.0  0.0  0 0 ?        S   04:32   0:00 [scsi_eh_0]
root       1562  0.0  0.0  0 0 ?        I<  04:32   0:00 [scsi_tmf_0]
root       1565  0.0  0.0  0 0 ?        S   04:32   0:00 [scsi_eh_1]

```

Run the below command to start the server and see the apache server welcome page.

```

root@ip-172-31-32-113/home/ec2-user
[ec2-user@ip-172-31-32-113 ~]$ sudo su
[root@ip-172-31-32-113 ec2-user]# systemctl start httpd
[root@ip-172-31-32-113 ec2-user]#

```

