Name :- Anirudha Kurhade

Prn :- 22120150

Assignment 3

Aim: - Deploy Web application on AWS Cloud (or any cloud)(PHP/Python/Node js any application).

Description :-

1. Cloud Computing Definition:

Cloud Computing refers to the delivery of computing services over the Internet. It enables organizations to access and utilize computing resources, such as servers, storage, databases, software, and analytics, without having to own, manage, and maintain the underlying infrastructure. Cloud computing enables organizations to access these resources on-demand, as they need them, providing flexible and scalable computing solutions that can be easily adapted to meet changing business needs. Additionally, cloud computing provides several benefits, such as reduced costs, improved agility, and increased innovation.

2. Cloud Service Models:

Infrastructure as a Service (IaaS): IaaS is a cloud service model that provides virtualized computing resources, including storage, computing power, and networking, over the Internet. With IaaS, organizations can rent these resources, such as virtual machines and storage, as needed, and only pay for what they use. This eliminates the need for organizations to invest in and maintain their own infrastructure, allowing them to focus on their core business activities.

1. Platform as a Service (PaaS): PaaS is a cloud service model that provides a platform and environment for organizations to develop, run, and manage applications and services. This model typically includes the underlying infrastructure, such as servers and storage, as well as a development environment and tools to support the creation and deployment of applications. This allows organizations to focus on the development of their applications and services, without having to worry about the underlying infrastructure.

2. Software as a Service (SaaS): SaaS is a cloud service model that provides access to software applications over the Internet. With SaaS, organizations can subscribe to and use

software applications, such as email, customer relationship management, and project management, without having to install, configure, and maintain the software themselves.

This provides organizations with a cost-effective and convenient way to access and use software applications.

Cloud Deployment Models:

1. Public Cloud: Public Cloud is a cloud deployment model where the cloud infrastructure is owned and operated by a third-party service provider and is made available to the general public over the Internet. Public Clouds are designed to provide a cost-effective and scalable solution for organizations that need to access and utilize computing resources. Public Clouds are typically operated by large companies, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform.

2. Private Cloud: Private Cloud is a cloud deployment model where the cloud infrastructure is owned, operated, and only used by a single organization. Private Clouds provide organizations with more control and customization options compared to public cloud services. They are typically used by organizations that have specific security, privacy, and compliance requirements.

3. Hybrid Cloud: Hybrid Cloud is a cloud deployment model that combines the benefits of both public and private clouds. This model allows organizations to run their most critical applications and data in a private cloud environment, while leveraging the cost-effectiveness and scalability of public clouds for less critical workloads. Hybrid Clouds are often used to provide organizations with the flexibility and scalability they need to meet changing business needs.

4. Community Cloud: Community Cloud is a cloud deployment model that is shared by several organizations and supports a specific community with shared concerns, such as security, compliance, jurisdiction, and data sovereignty. Community Clouds provide organizations with a way to collaborate and share resources, while also ensuring that their data and applications are protected.

Steps Output :-

→ Login To Aws

STEP 2

→ Go to E2C Instance via compute or access it directly from from frequently used section

STEP 3

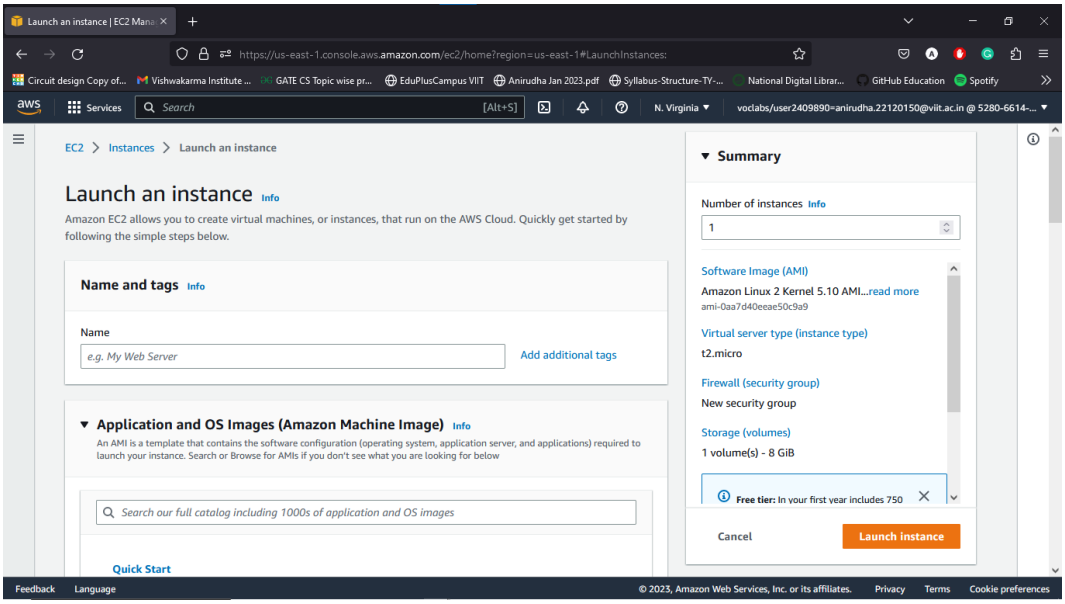
→ Configure the Instance

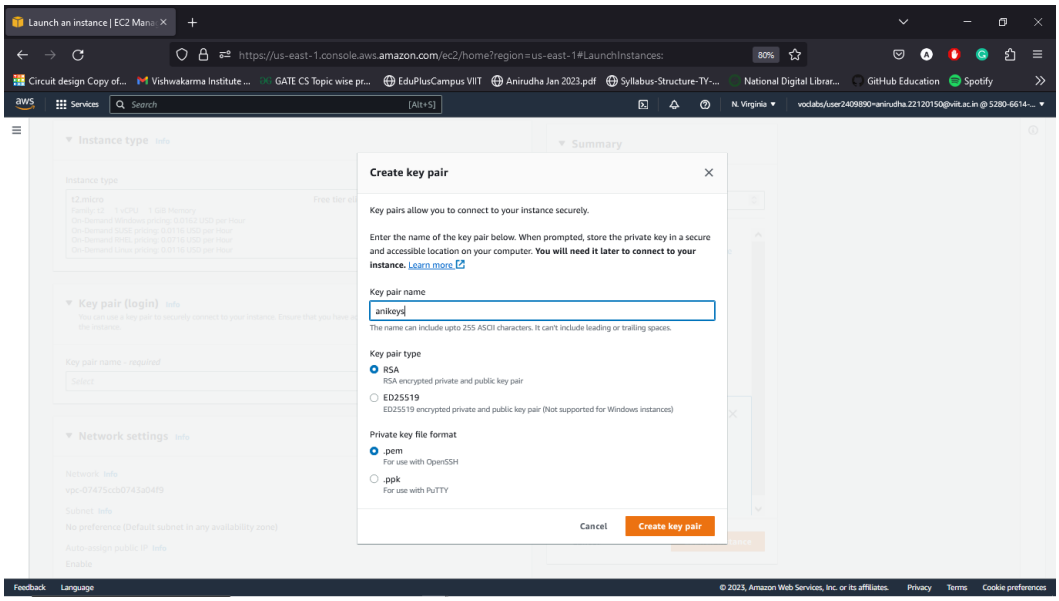
Name → Any name you like

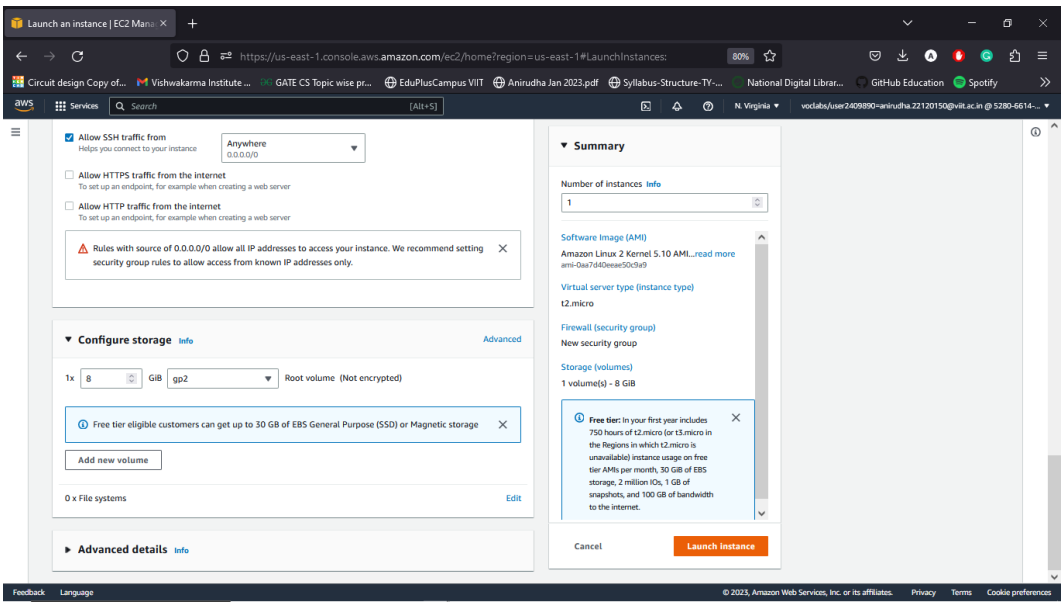
OS → aws linux

Architecture → 64 bit

Create keys (Give any name to them and download it)


Launch an instance | EC2 Mana
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Services
Search
[Alt+S]
N. Virginia
voclabs/user2409890=anirudha.22120150@viit.ac.in @ 5280-6614-...
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
e.g. My Web Server
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Quick Start
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0aa7d40eeae50c9a9
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750
Cancel
Launch instance
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

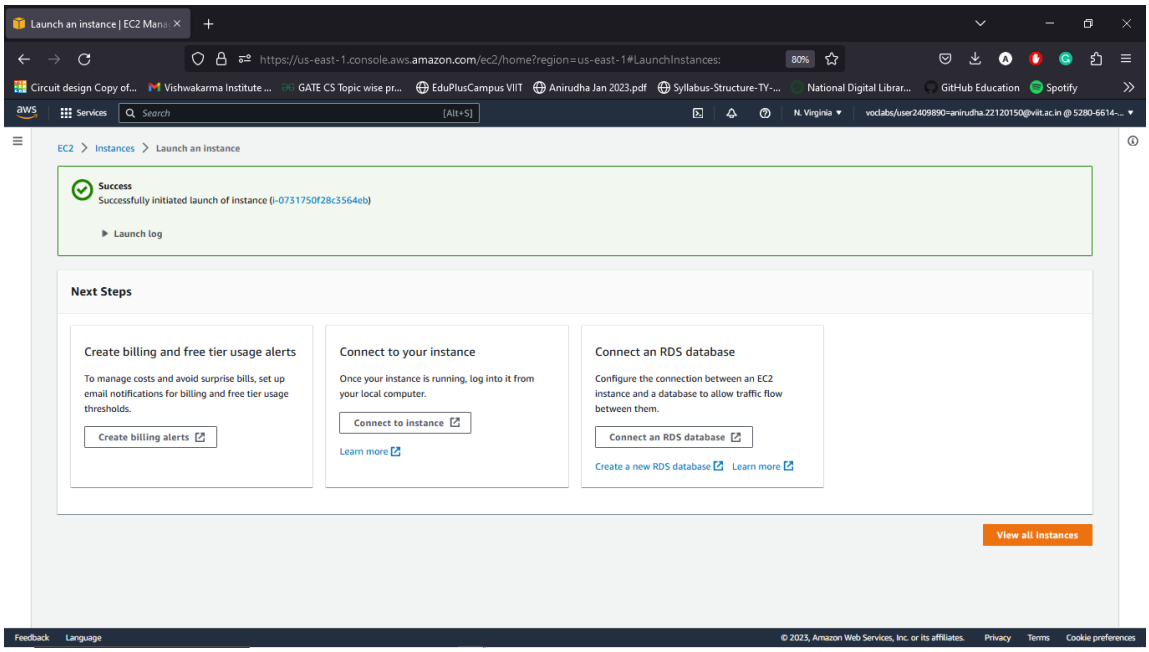

Create key pair
Key pairs allow you to connect to your instance securely.
Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
Key pair name
anikeys
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
Cancel
Create key pair


Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Configure storage Info
Advanced
1x
8
GiB
gp2
Root volume (Not encrypted)
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage
Add new volume
0 x File systems
Edit
Advanced details Info
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0aa7d40eeae50c9a9
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance

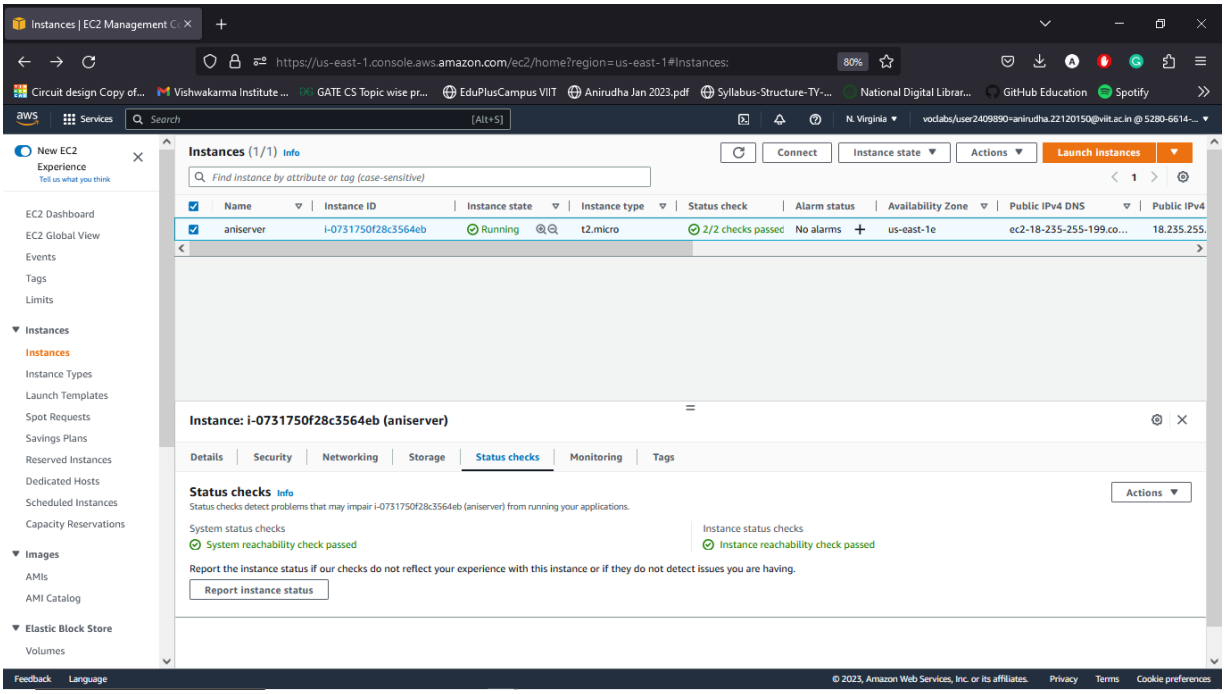
STEP 4

→ Launch the instance

→ After clicking at right bottom launch instance button it show look like this

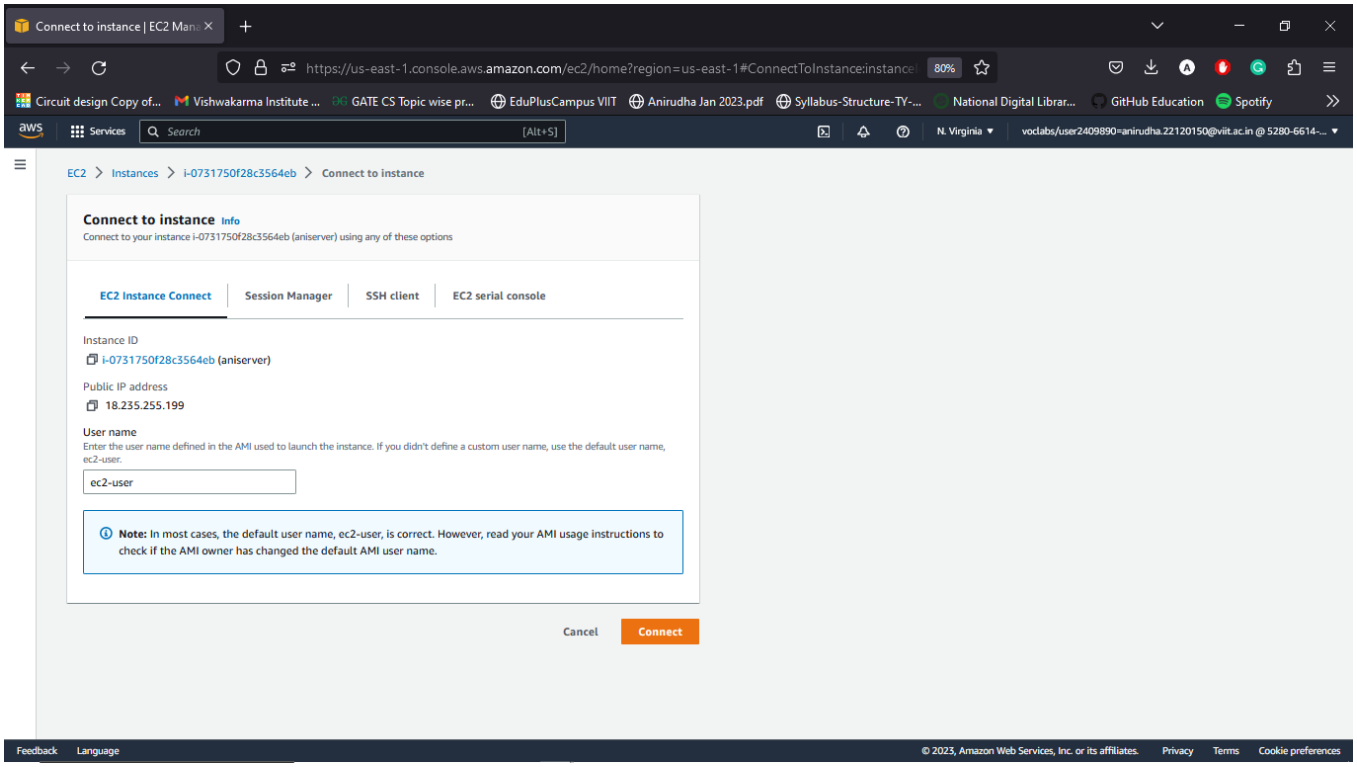


We will begin launching the instance after 2/2 status checks

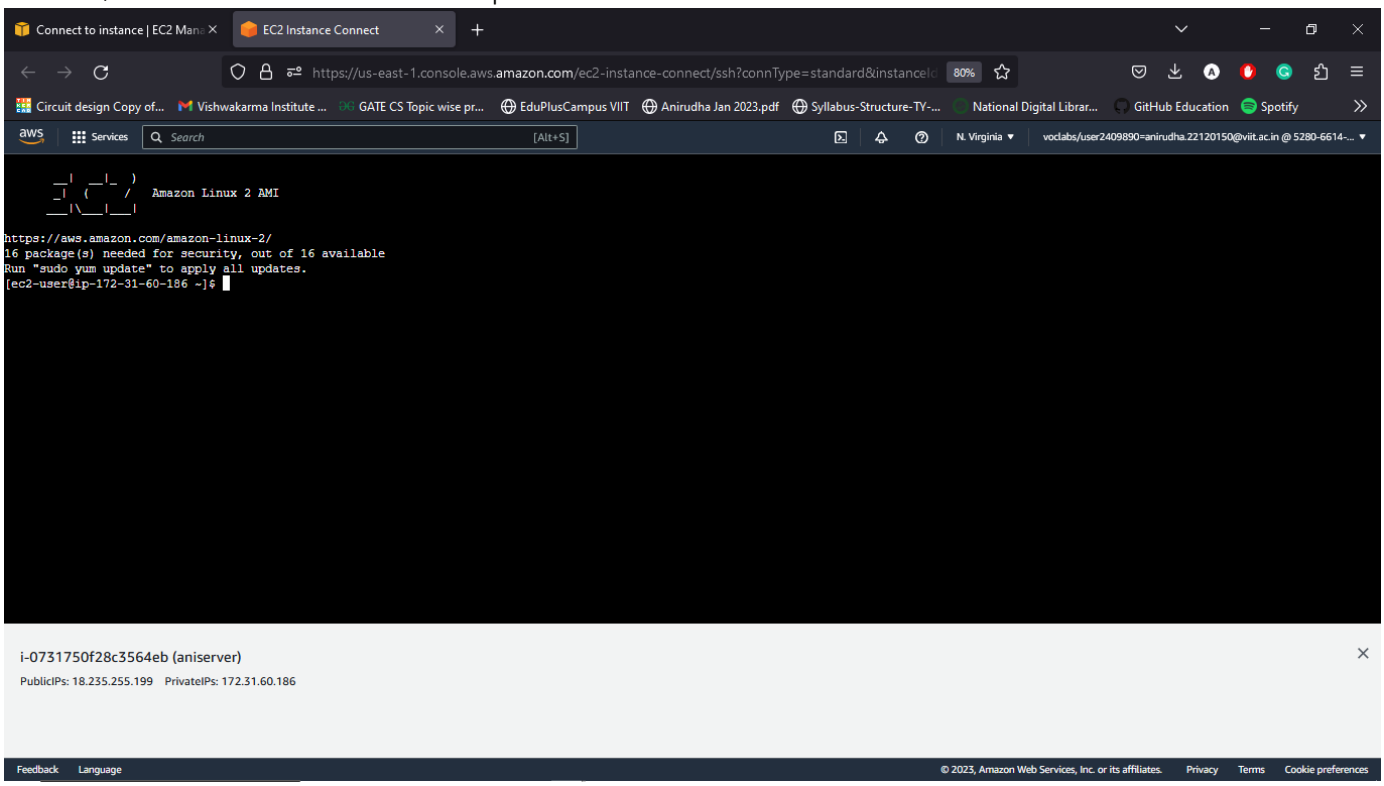


STEP 5

→ Launch the instance (Connect to the instance)

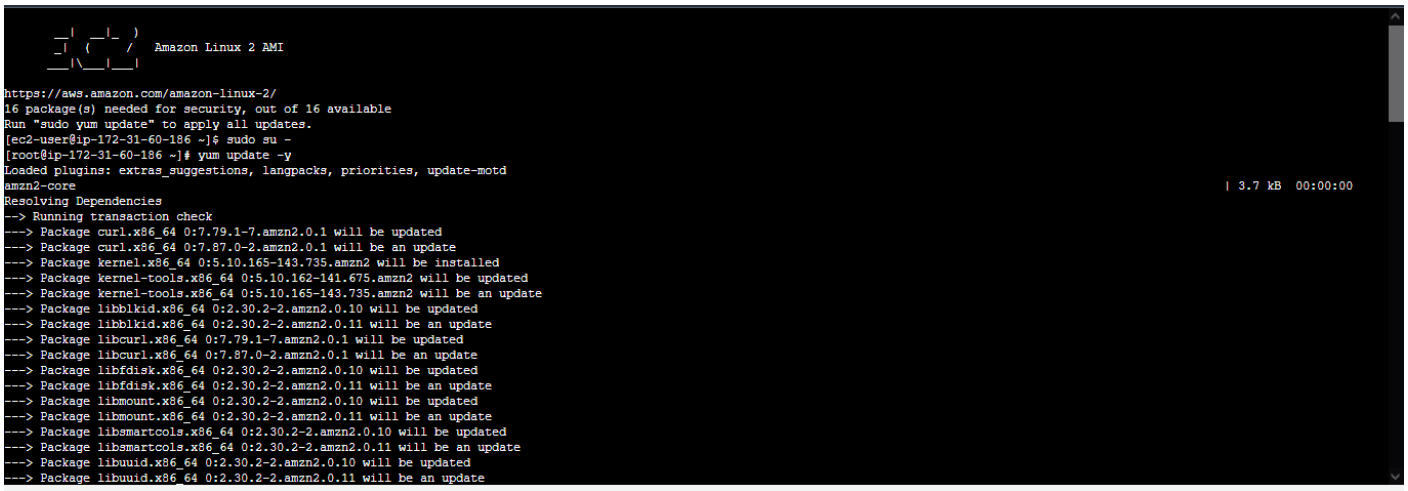


→ This terminal window shall open

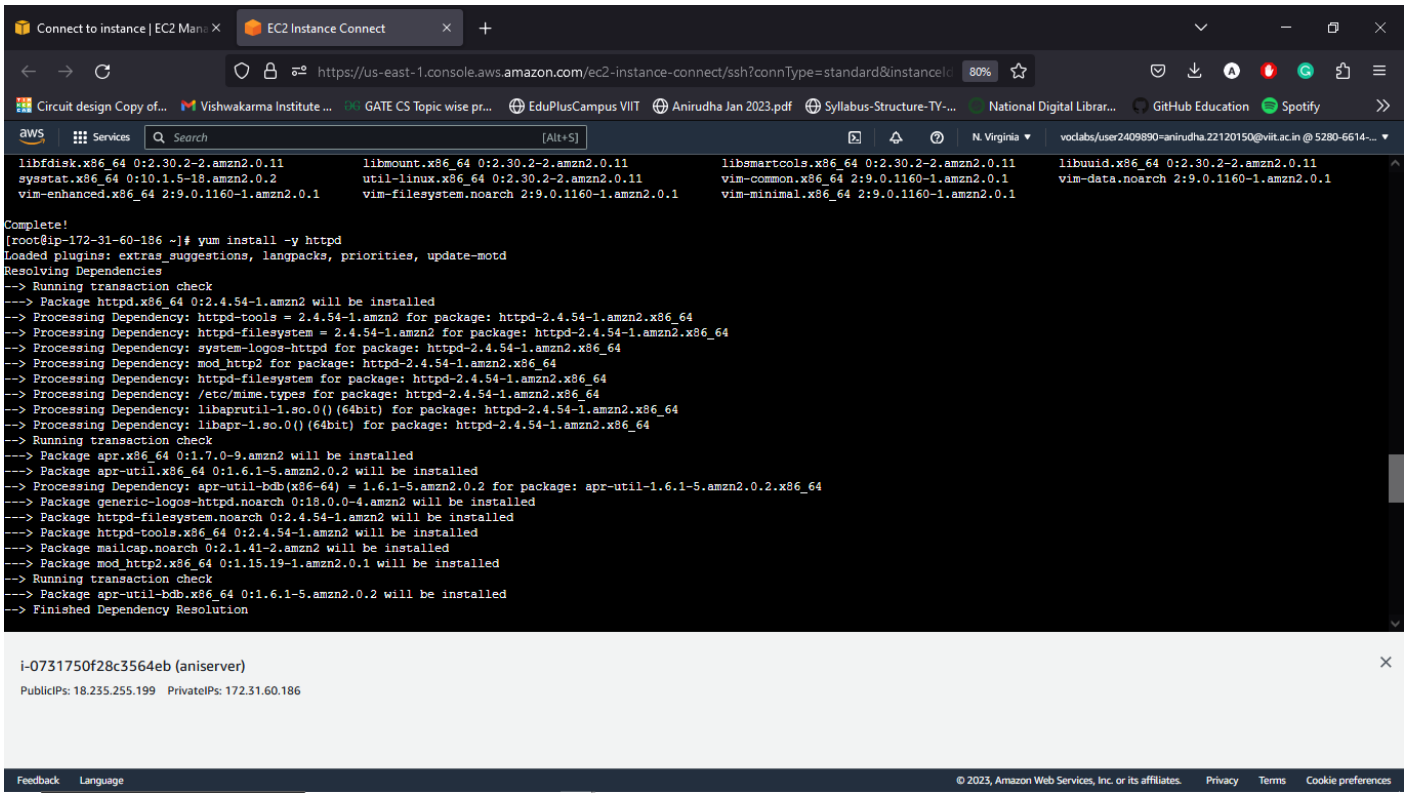


→ Put this commands in order

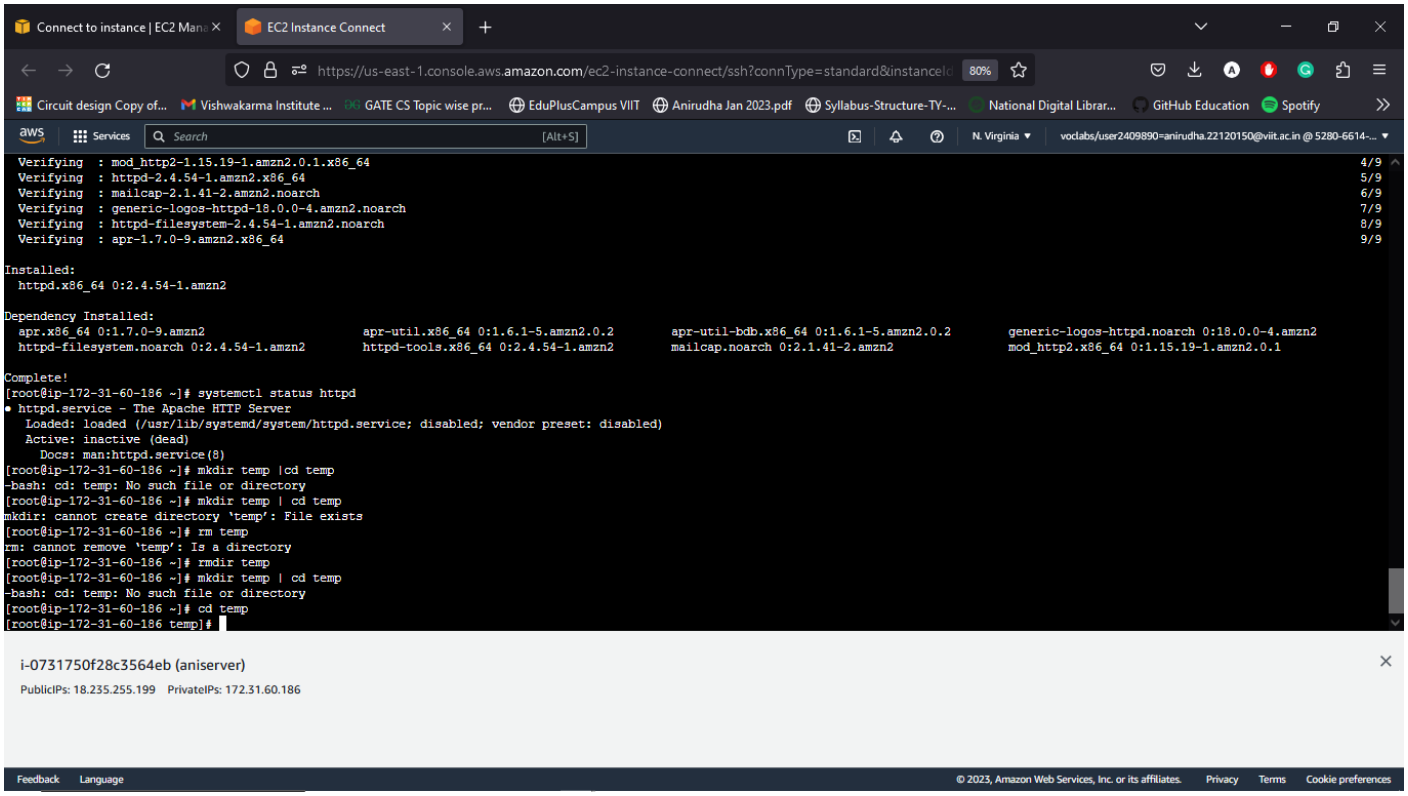
a. sudo su -
b. yum update -y



c. yum install -y httpd

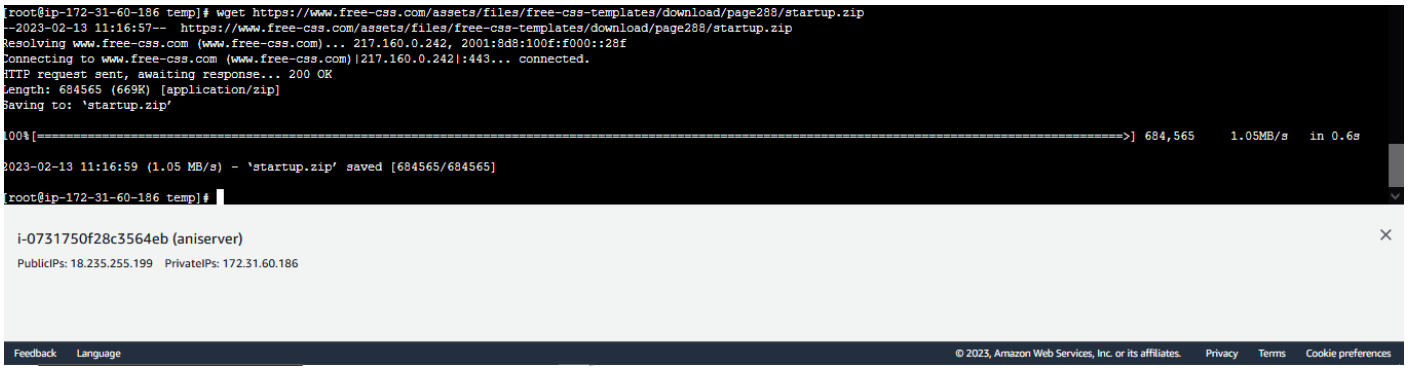


d. systemctl status httpd
e. mkdir temp | cd temp

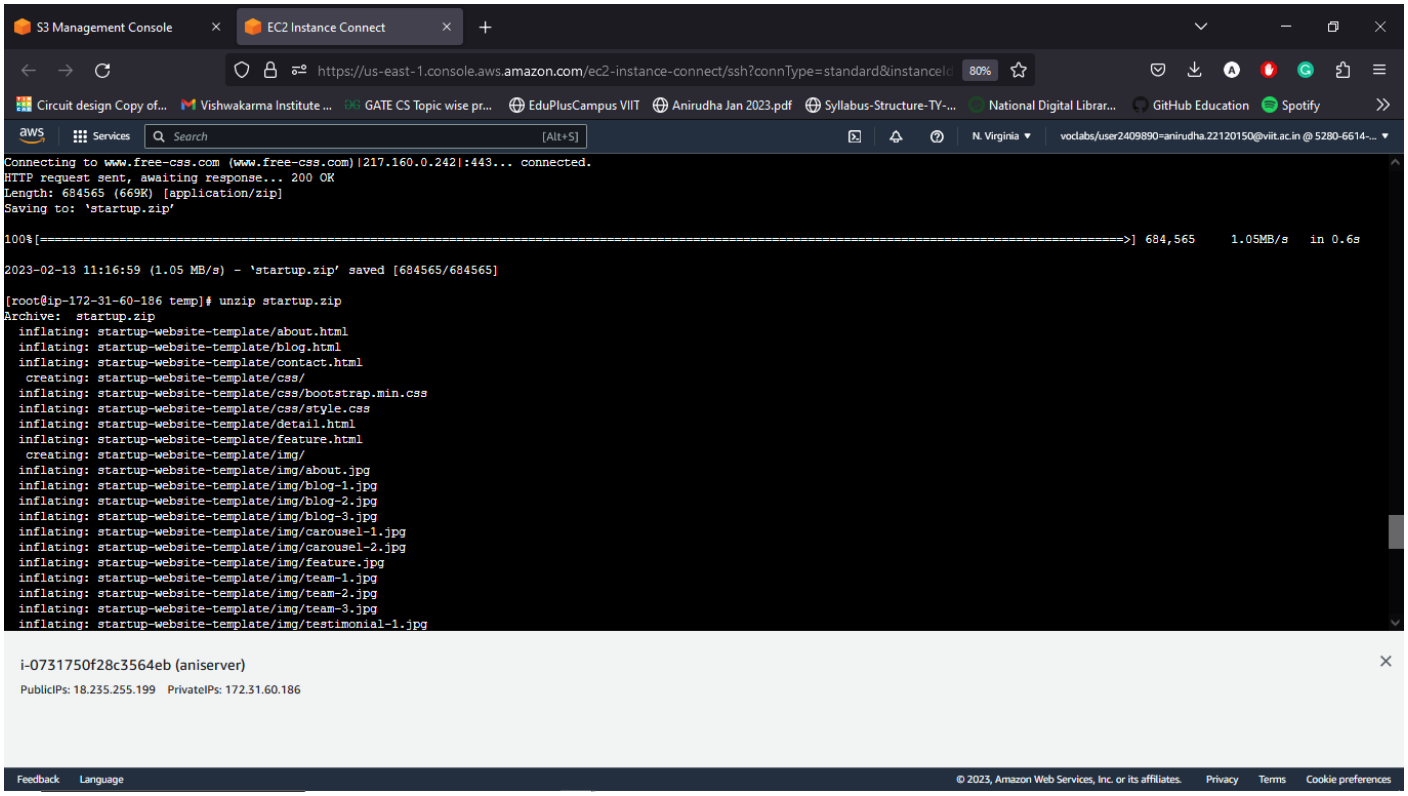


f. wget https://www.free-css.com/assets/files/free-css-templates/download/page288/startup.zip

1. As I’m no web developer so I took a template from https://www.free-css.com/free-css-templates remember to place the download link in front of wget



g. unzip startup.zip



h. cd startup-website-template/

```
[root@ip-172-31-9-124 temp]# cd startup-website-template/
[root@ip-172-31-9-124 startup-website-template]# ls-lrt
-bash: ls-lrt: command not found
[root@ip-172-31-9-124 startup-website-template]# ls -lrt
total 400
-rw-r--r-- 1 root root 89601 Jul 27  2021 startup-website-template.jpg
-rw-r--r-- 1 root root   538 Aug 11  2021 README-ME.txt
drwxr-xr-x 8 root root    99 Aug 11  2021 lib
drwxr-xr-x 2 root root    21 Aug 11  2021 js
drwxr-xr-x 2 root root  4096 Aug 11  2021 img
```

i. mv * /var/www/html/

```
[root@ip-172-31-9-124 startup-website-template]# mv * /var/www/html/
```

j. cd /var/www/html/

```
[root@ip-172-31-9-124 startup-website-template]# cd /var/www/html/
[root@ip-172-31-9-124 html]# ls -lrt
total 400
-rw-r--r-- 1 root root 89601 Jul 27  2021 startup-website-template.jpg
-rw-r--r-- 1 root root   538 Aug 11  2021 README-ME.txt
drwxr-xr-x 8 root root    99 Aug 11  2021 lib
drwxr-xr-x 2 root root    21 Aug 11  2021 js
drwxr-xr-x 2 root root  4096 Aug 11  2021 img
drwxr-xr-x 2 root root    48 Aug 11  2021 css
-rw-r--r-- 1 root root  1456 Aug 16  2021 LICENSE.txt
-rw-r--r-- 1 root root 22799 Oct 19  2021 about.html
-rw-r--r-- 1 root root 38423 Oct 19  2021 blog.html
-rw-r--r-- 1 root root 20237 Oct 19  2021 contact.html
-rw-r--r-- 1 root root 30401 Oct 19  2021 detail.html
-rw-r--r-- 1 root root 19294 Oct 19  2021 testimonial.html
```

k. ls -lrt

```
[root@ip-172-31-9-124 html]# ls -lrt
total 400
-rw-r--r-- 1 root root 89601 Jul 27  2021 startup-website-template.jpg
-rw-r--r-- 1 root root   538 Aug 11  2021 READ-ME.txt
drwxr-xr-x 8 root root    99 Aug 11  2021 lib
drwxr-xr-x 2 root root    21 Aug 11  2021 js
drwxr-xr-x 2 root root  4096 Aug 11  2021 img
drwxr-xr-x 2 root root    48 Aug 11  2021 css
-rw-r--r-- 1 root root  1456 Aug 16  2021 LICENSE.txt
-rw-r--r-- 1 root root 22799 Oct 19  2021 about.html
-rw-r--r-- 1 root root 38423 Oct 19  2021 blog.html
-rw-r--r-- 1 root root 20237 Oct 19  2021 contact.html
-rw-r--r-- 1 root root 30401 Oct 19  2021 detail.html
-rw-r--r-- 1 root root 19294 Oct 19  2021 testimonial.html
-rw-r--r-- 1 root root 19029 Oct 19  2021 feature.html
-rw-r--r-- 1 root root 52988 Oct 19  2021 index.html
-rw-r--r-- 1 root root 24975 Oct 19  2021 price.html
-rw-r--r-- 1 root root 19602 Oct 19  2021 quote.html
-rw-r--r-- 1 root root 24329 Oct 19  2021 service.html
-rw-r--r-- 1 root root 20123 Oct 19  2021 team.html
```

l. systemctl enable httpd | systemctl start httpd

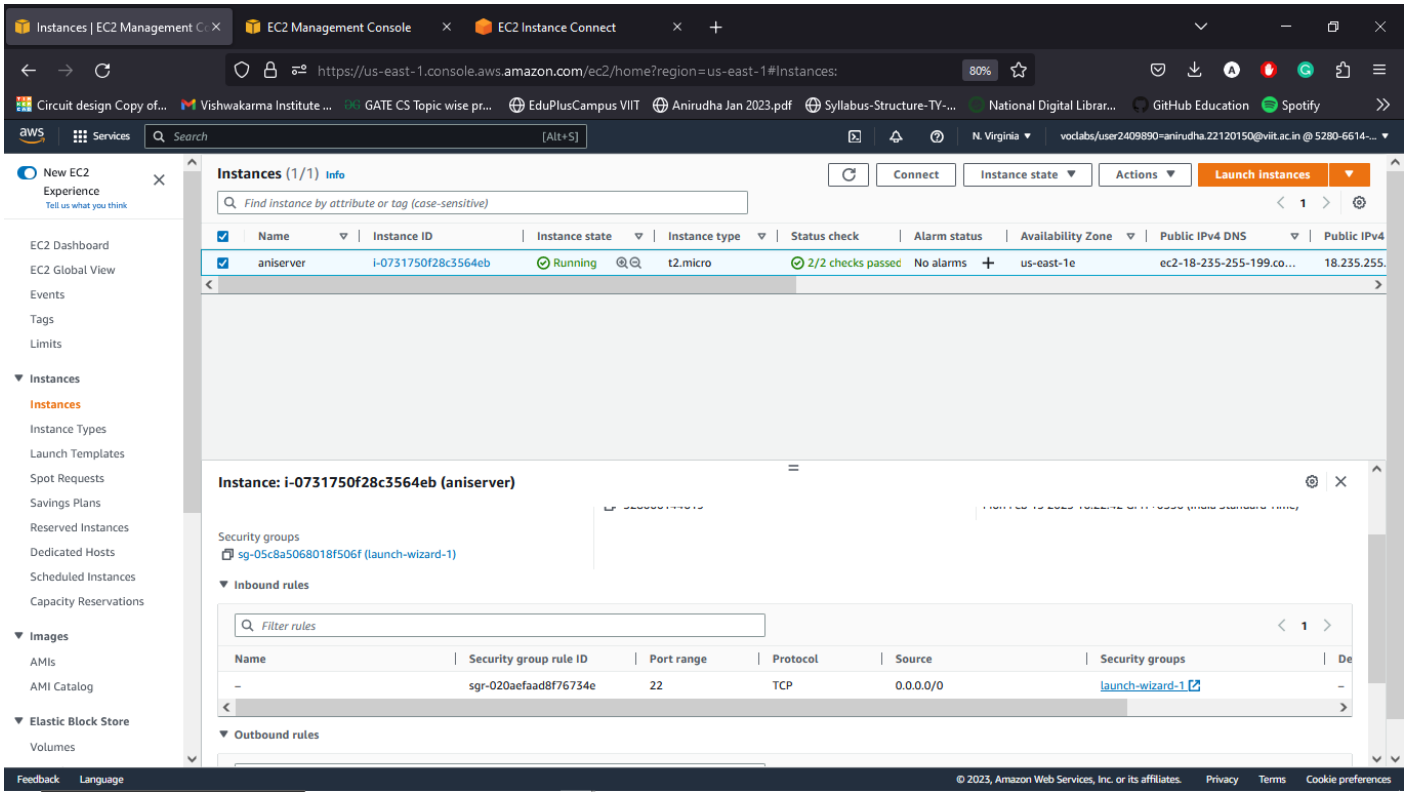
```
[root@ip-172-31-9-124 html]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-9-124 html]# system start httpd
-bash: system: command not found
[root@ip-172-31-9-124 html]# sysytemctl start httpd
-bash: sysytemctl: command not found
[root@ip-172-31-9-124 html]# sytemctl start httpd
-bash: sytemctl: command not found
[root@ip-172-31-9-124 html]# systemctl start httpd
```

m. systemctl status httpd

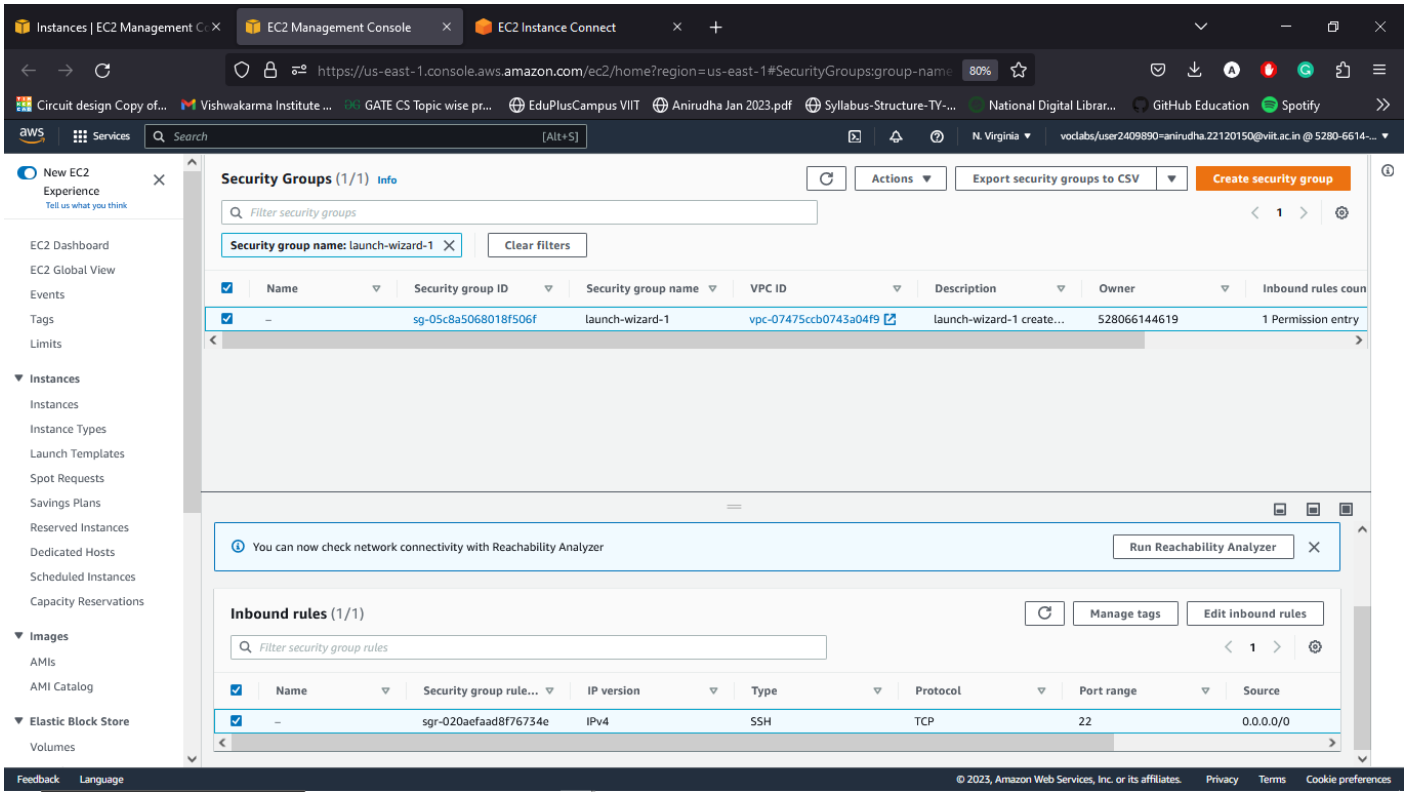
```
[root@ip-172-31-9-124 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-02-05 12:37:24 UTC; 22s ago
     Docs: man:httpd.service(8)
 Main PID: 4034 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─4034 /usr/sbin/httpd -DFOREGROUND
           ├─4035 /usr/sbin/httpd -DFOREGROUND
           ├─4036 /usr/sbin/httpd -DFOREGROUND
           ├─4037 /usr/sbin/httpd -DFOREGROUND
           ├─4038 /usr/sbin/httpd -DFOREGROUND
           └─4039 /usr/sbin/httpd -DFOREGROUND

Feb 05 12:37:24 ip-172-31-9-124.ap-northeast-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Feb 05 12:37:24 ip-172-31-9-124.ap-northeast-1.compute.internal systemd[1]: Started The Apache HTTP Server.
```

STEP 6

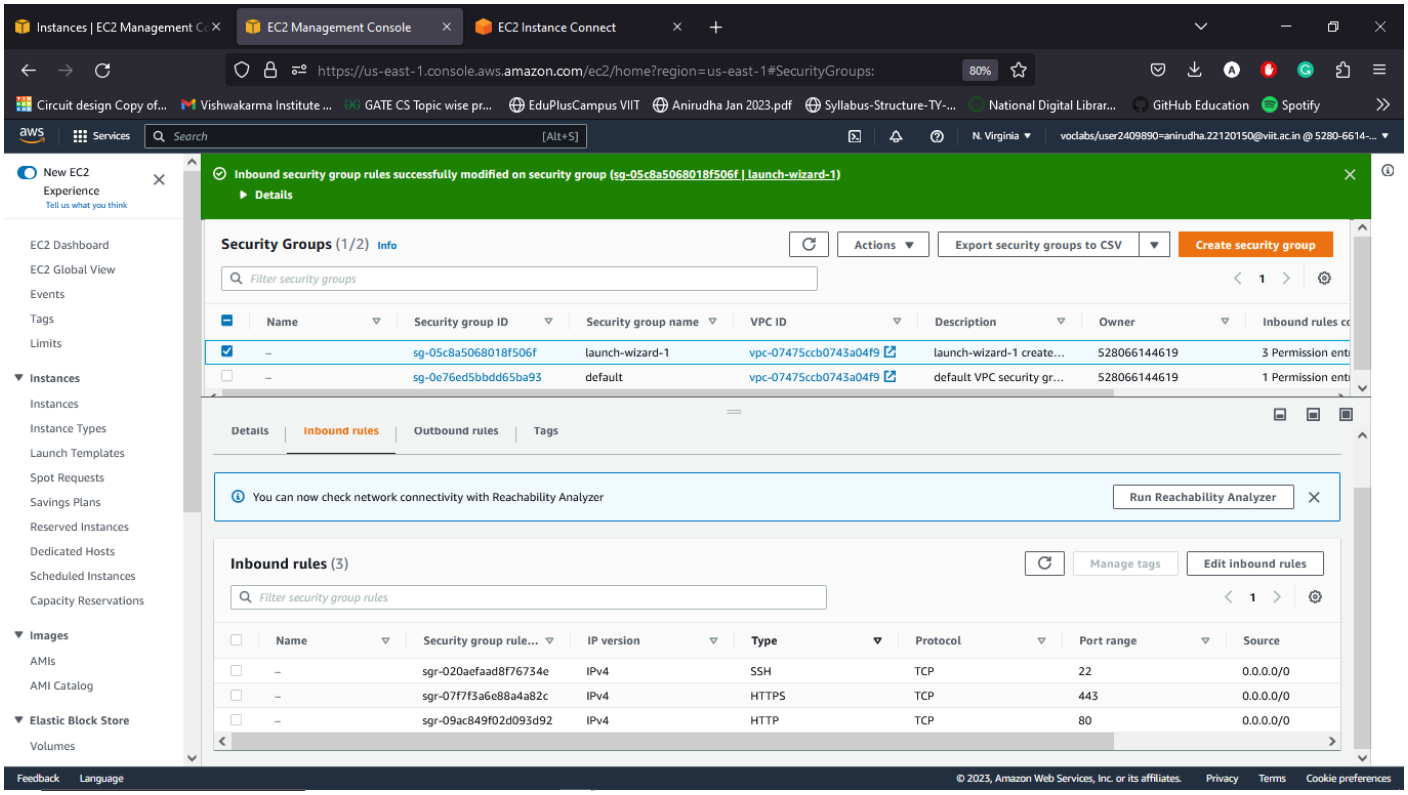
→ Change the inbound rules from security by simply clicking on launch wizard first



→ Click on edit inbound rules



→ Add HTTP and HTTPS like the configuration given below



STEP 7

→ Copy and paste the public IP to the browser to see the web template hosted

STEP 8

→ Stop the instance & terminate it

Conclusion :-

Thus, We Have Studied And Deployed Web application on AWS Cloud.