

**Name – Majahar Mahamud Kazi**

**Div – B**

**Batch – B2**

**Roll no. – 322036**

**PRN no. – 22110729**

---

## **Assignment 3**

**Aim:** Deploy Web application on AWS Cloud.

### **Theory:**

#### **A. Cloud Computing –**

Cloud Computing means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server. Cloud computing is also referred to as Internet-based computing, it is a technology where the resource is provided as a service through the Internet to the user. The data that is stored can be files, images, documents, or any other storable document.

The following are some of the Operations that can be performed with Cloud Computing

- Storage, backup, and recovery of data
- Delivery of software on demand
- Development of new applications and services
- Streaming videos and audio



## **B. Cloud Service models and Deployment models – Service Models -**

### **1) Infrastructure as a Service ( IaaS ):**

- Flexibility and Control: IaaS comes up with providing virtualized computing resources such as VMs, Storage, and networks facilitating users with control over the Operating system and applications.
- Reducing Expenses of Hardware: IaaS provides business cost savings with the elimination of physical infrastructure investments making it cost-effective.
- Scalability of Resources: The cloud provides in scaling of hardware resources up or down as per demand facilitating optimal performance with cost efficiency.

### **2) Platform as a Service ( PaaS ):**

- Simplifying the Development: Platform as a Service offers application development by keeping the underlying Infrastructure as an Abstraction. It helps the developers to completely focus on application logic ( Code ) and background operations are completely managed by the AWS platform.
- Enhancing Efficiency and Productivity: PaaS lowers the Management of Infrastructure complexity, speeding up the Execution time and bringing the updates quickly to market by streamlining the development process.
- Automation of Scaling: Management of resource scaling, guaranteeing the program's workload efficiency is ensured by PaaS.



### **3) SaaS (software as a service):**

- Collaboration And Accessibility: Software as a Service (SaaS) helps users to easily access applications without having the requirement of local installations. It is fully managed by the AWS Software working as a service over the internet encouraging effortless cooperation and ease of access.
- Automation of Updates: SaaS providers manage the handling of software maintenance with automatic latest updates ensuring users gain experience with the latest features and security patches.
- Cost Efficiency: SaaS acts as a cost-effective solution by reducing the overhead of IT support by eliminating the need for individual software licenses.

### **4) Function as a Service (FaaS):**

- Event-Driven Execution: FaaS helps in the maintenance of servers and infrastructure making users worry about it. FaaS facilitates the developers to run code as a response to the events.
- Cost Efficiency: FaaS facilitates cost efficiency by coming up with the principle “Pay as per you Run” for the computing resources used.
- Scalability and Agility: Serverless Architectures scale effortlessly in handling the workloads promoting agility in development and deployment.



## Cloud Deployment Models -

### 1) Public Cloud:

- The cloud infrastructure is owned and operated by third-party providers.
- Resources are shared among multiple customers.
- Examples include AWS, Azure, and Google Cloud Platform.

### 2) Private Cloud:

- The cloud infrastructure is dedicated to a single organization.
- It can be managed by the organization or a third-party provider and can be on-premises or off-premises.
- Offers greater control and customization.

### 3) Hybrid Cloud:

- Combines public and private cloud environments.
- Allows data and applications to be shared between them.
- Provides greater flexibility, as organizations can use the public cloud for scalable resources and the private cloud for sensitive data.

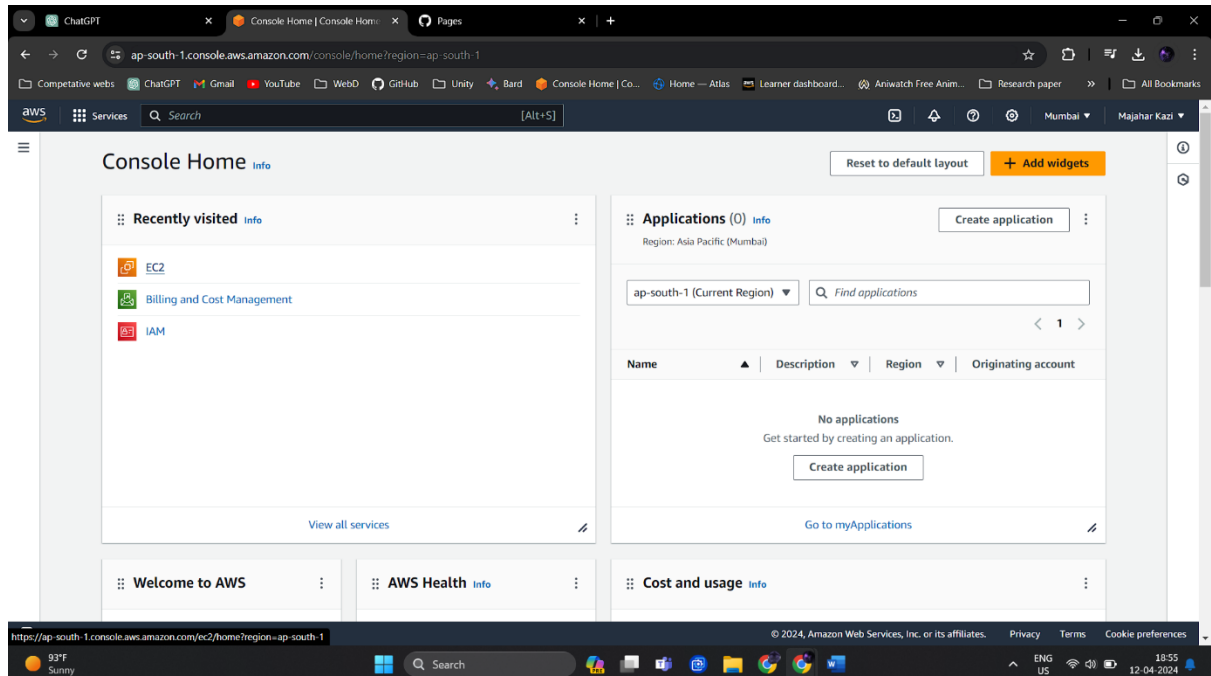
### 4) Community Cloud:

- Shared by several organizations with similar needs and concerns (e.g., security, compliance, jurisdiction).
- Managed by one of the organizations or a third-party provider.
- Offers shared resources and infrastructure tailored to the community's needs.

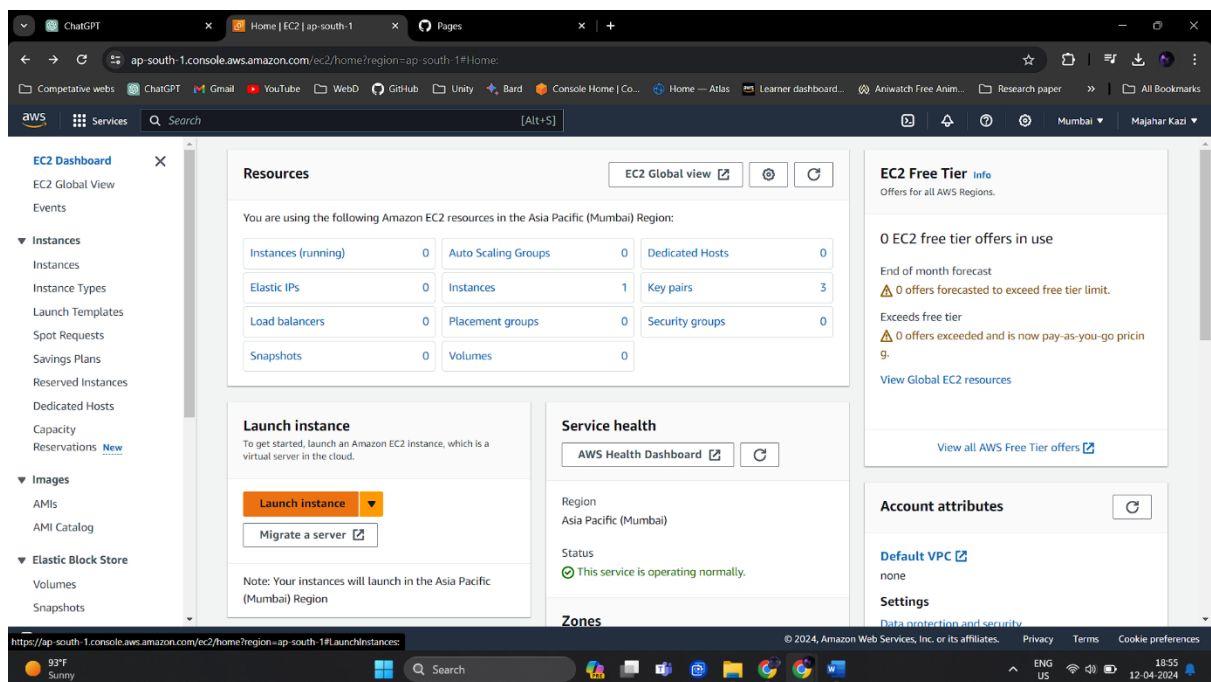


## C. Launching web application on the cloud:

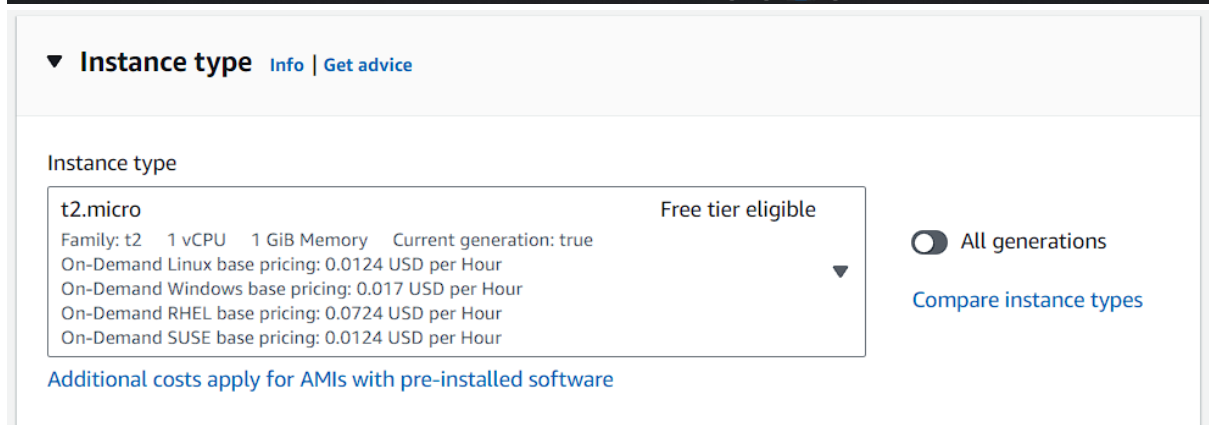
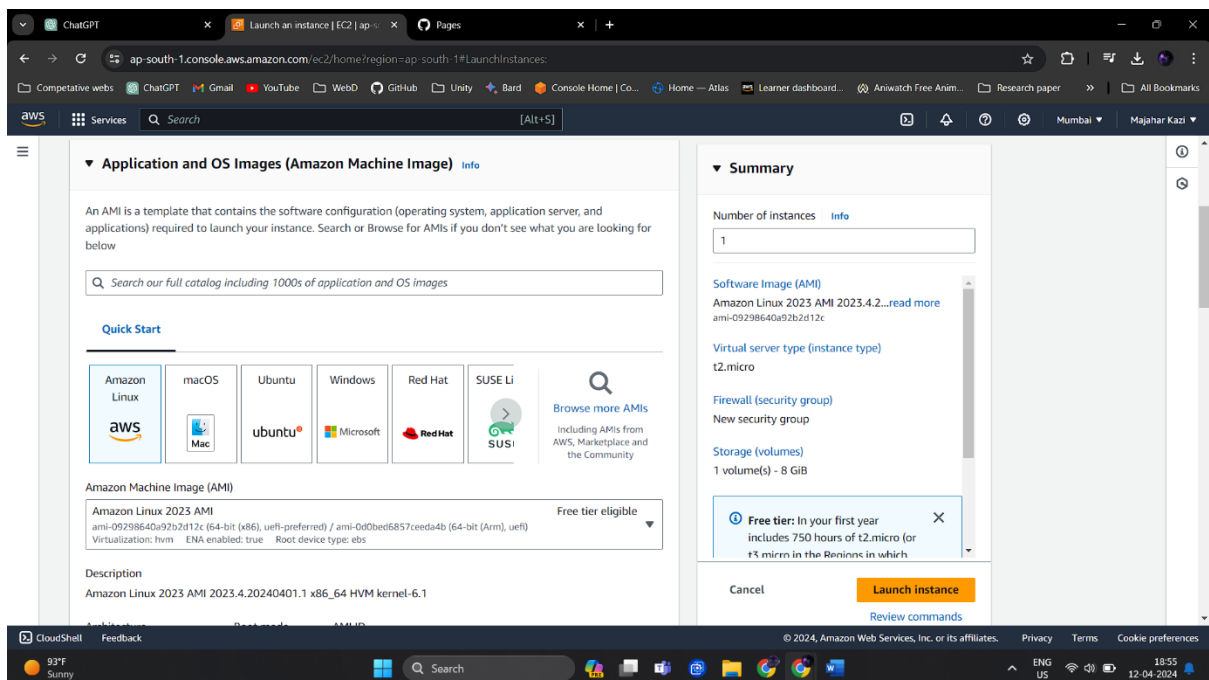
At open aws console and login. After login click onto ec2 instance.



Then click onto launch instance.



Let the parameters for instance and machine be default.



Create a key pair for our instance, with .ppk file format.

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

webDeployment

The name can include up to 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

Private key file format

☐ .pem

For use with OpenSSH

☒ .ppk

For use with PuTTY

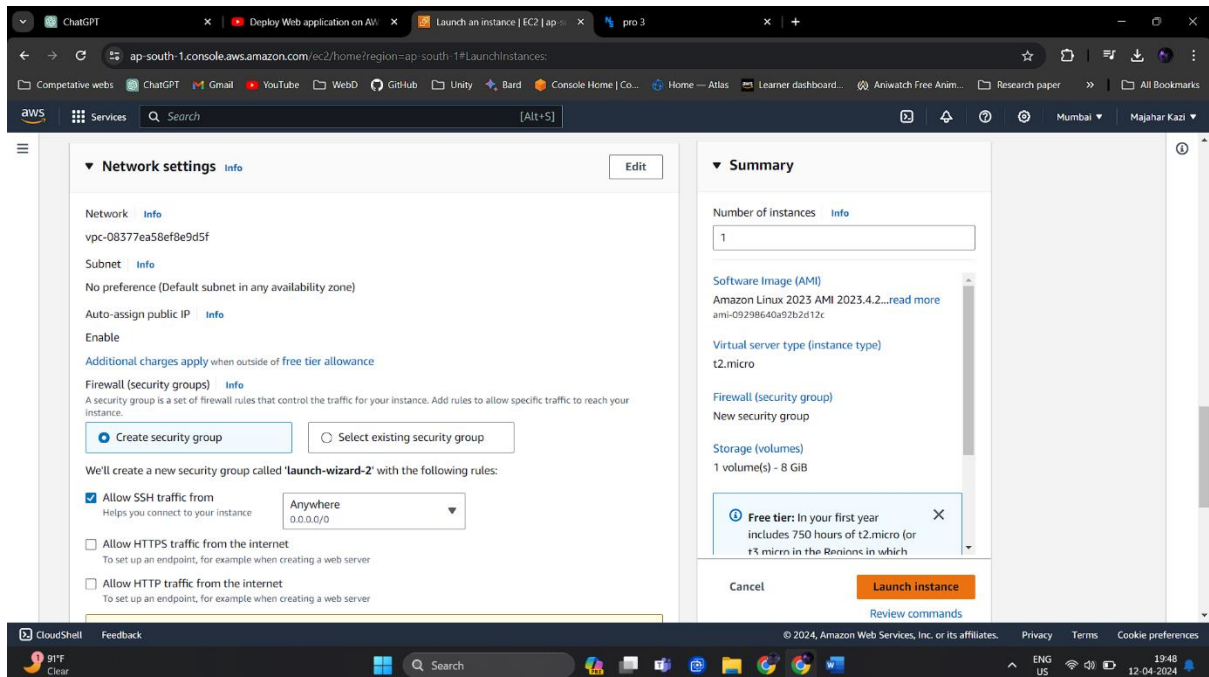
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

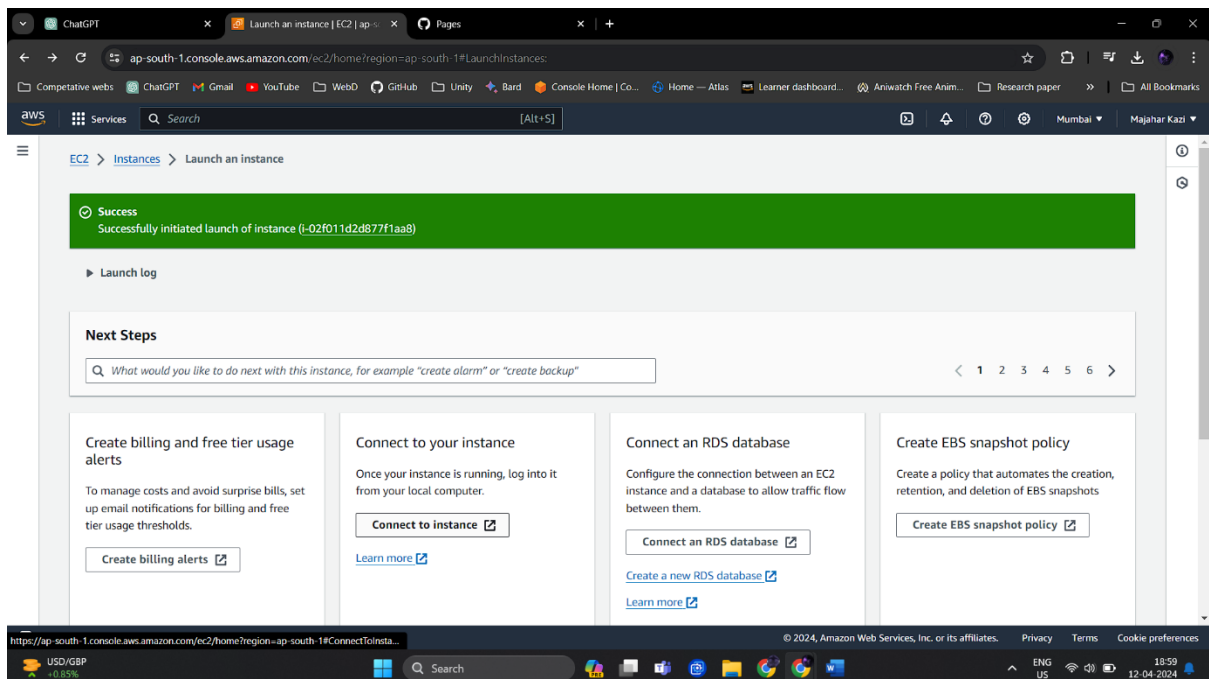
Create key pair



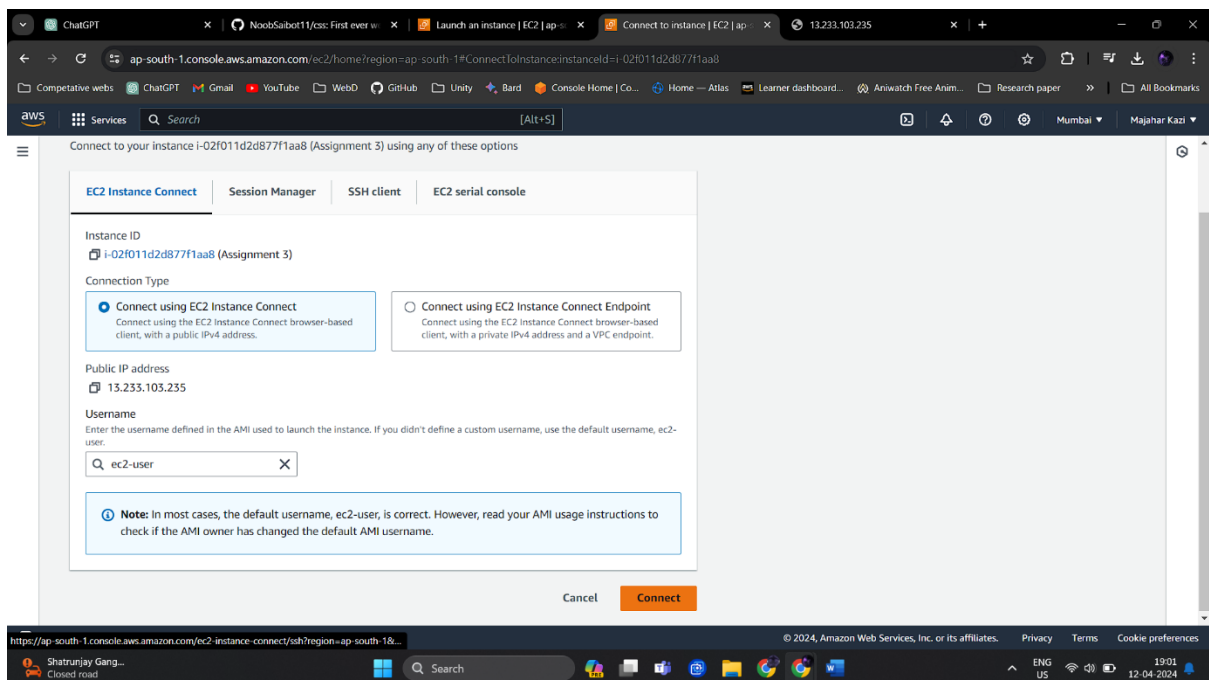
Now all the arrangements for our instance are done, it's time to launch our instance.



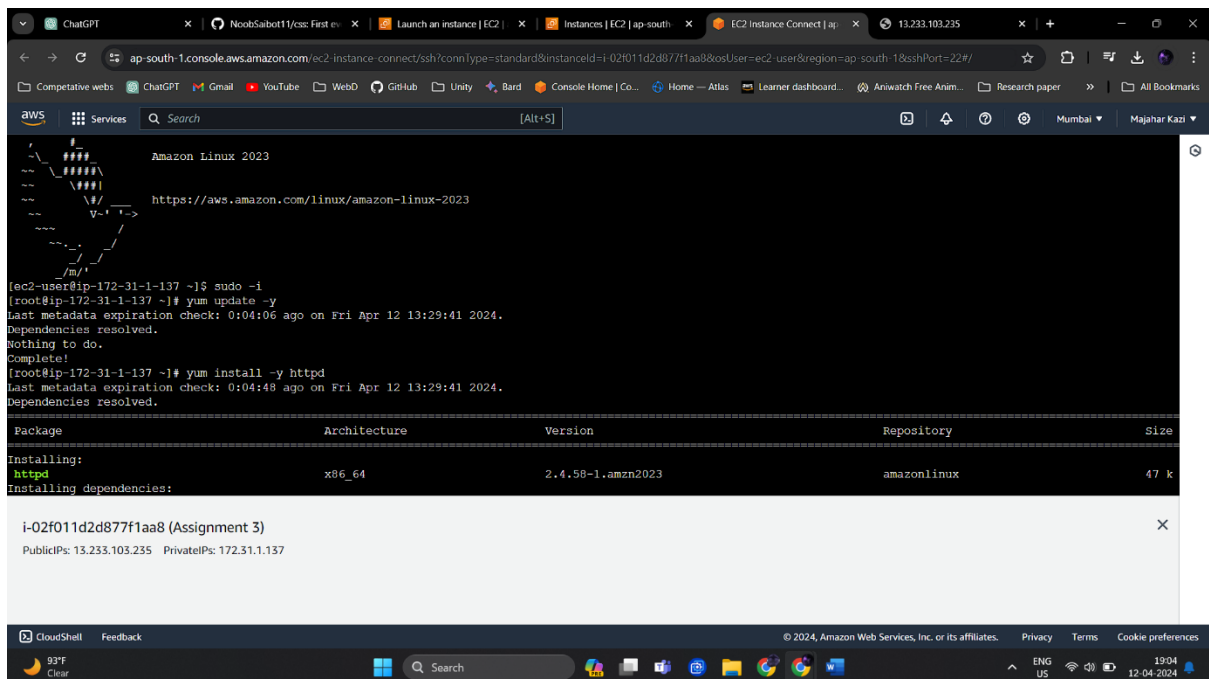
Our instance is created, now we need to connect to our instance by connect your instance option.



Press connect to connect with our virtual machine.



Now we are into our new vm, first we need to update it and install httpd in it to launch our website.





After installing httpd, we can see httpd services are disabled.

```
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/12
Verifying : httpd-2.4.58-1.amzn2023.x86_64 5/12
Verifying : httpd-core-2.4.58-1.amzn2023.x86_64 6/12
Verifying : httpd-filesystem-2.4.58-1.amzn2023.noarch 7/12
Verifying : httpd-tools-2.4.58-1.amzn2023.x86_64 8/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : mod_http2-2.0.11-2.amzn2023.x86_64 11/12
Verifying : mod_lua-2.4.58-1.amzn2023.x86_64 12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      httpd-2.4.58-1.amzn2023.x86_64      httpd-core-2.4.58-1.amzn2023.x86_64
httpd-filesystem-2.4.58-1.amzn2023.noarch      httpd-tools-2.4.58-1.amzn2023.x86_64      libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.11-2.amzn2023.x86_64      mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[root@ip-172-31-1-137 ~]# systemctl status httpd
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
   Docs: man:httpd.service(8)
[root@ip-172-31-1-137 ~]#
```

i-02f011d2d877f1aa8 (Assignment 3)  
PublicIPs: 13.233.103.235 PrivateIPs: 172.31.1.137

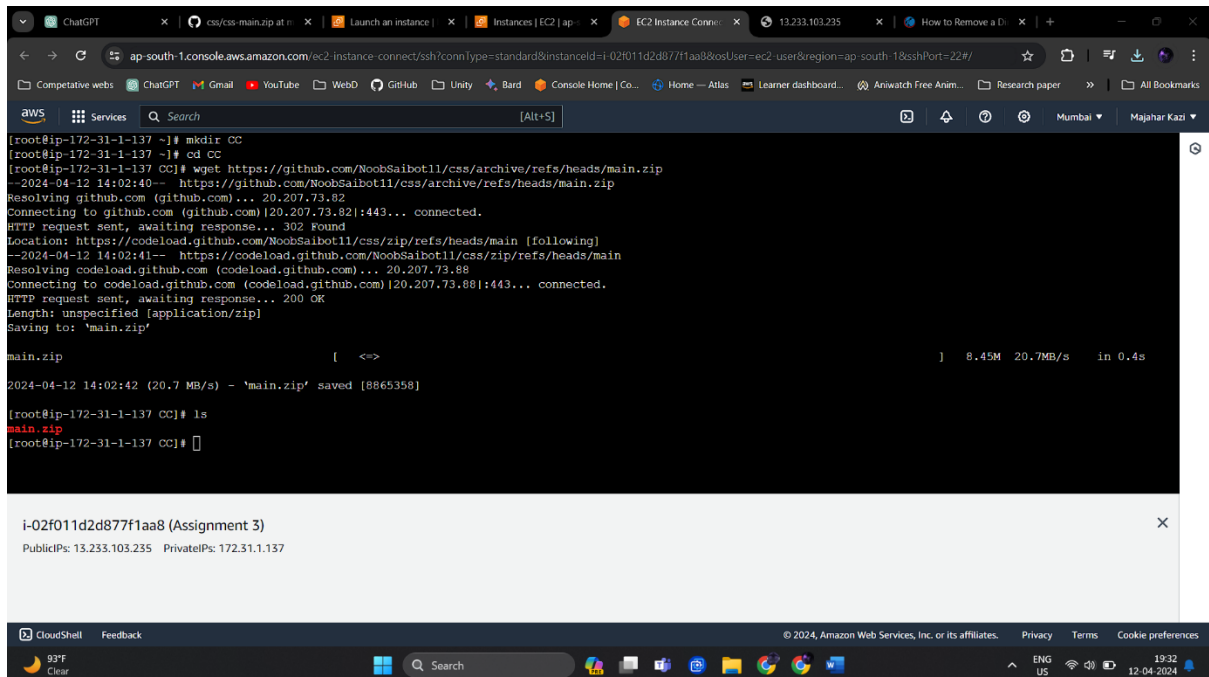
Now let us create a new directory in our vm for storing our website files.

```
[root@ip-172-31-1-137 ~]# mkdir CC
[root@ip-172-31-1-137 ~]# cd CC
[root@ip-172-31-1-137 CC]#
```

i-02f011d2d877f1aa8 (Assignment 3)  
PublicIPs: 13.233.103.235 PrivateIPs: 172.31.1.137

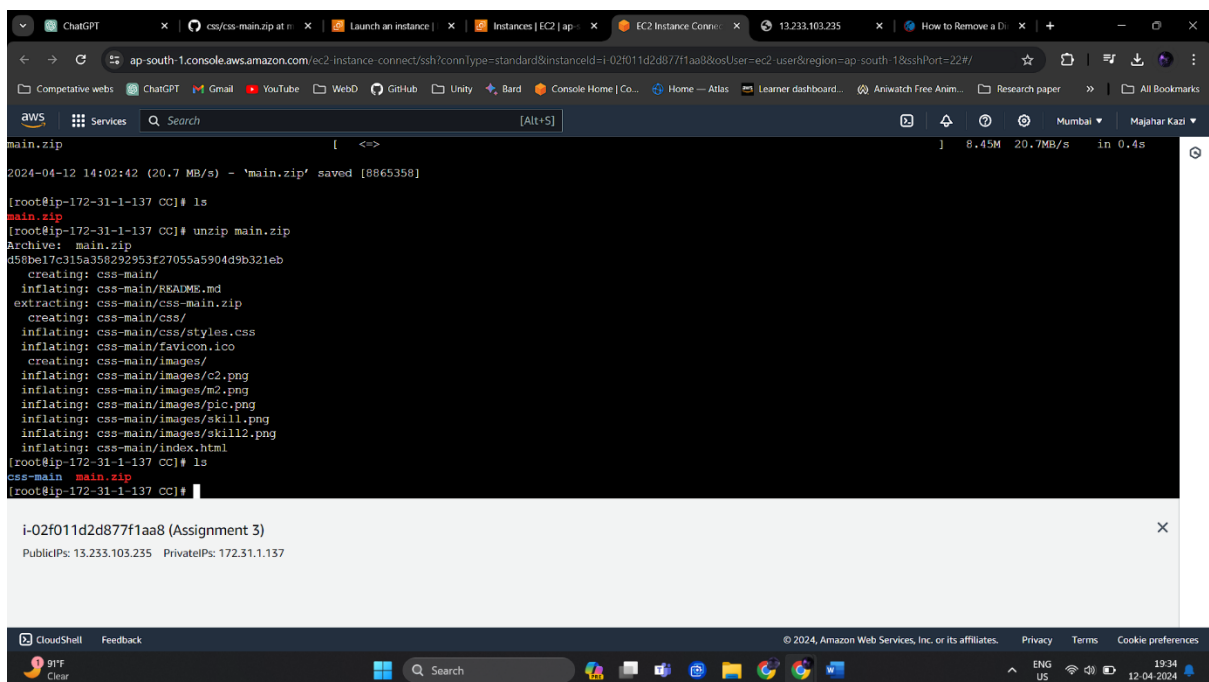


Once we create new directory for our website files, we download those website files from our github.



The screenshot shows an AWS CloudShell terminal window. The user is at a root prompt on an EC2 instance. They run `mkdir CC` to create a new directory. Then, they run `wget https://github.com/NoobSaibot11/css/archive/refs/heads/main.zip` to download a zip file from GitHub. The terminal shows the progress of the download, including the file size (8.45M) and the speed (20.7MB/s). After the download is complete, the user runs `ls` to list the files in the current directory, showing `main.zip`.

We have got the zip file for our files, now let's unzip the main file.



The screenshot shows the same AWS CloudShell terminal window. The user runs `unzip main.zip` to extract the contents of the zip file. The terminal shows the progress of the extraction, including the file size (8.45M) and the speed (20.7MB/s). After the extraction is complete, the user runs `ls` to list the files in the current directory, showing `css-main` and `main.zip`.

We can see all our web-files are extracted at our destination.



After that, we need to make changes in inbound group of our instance to allow traffic from http and https. Open security group of our instance from instances.

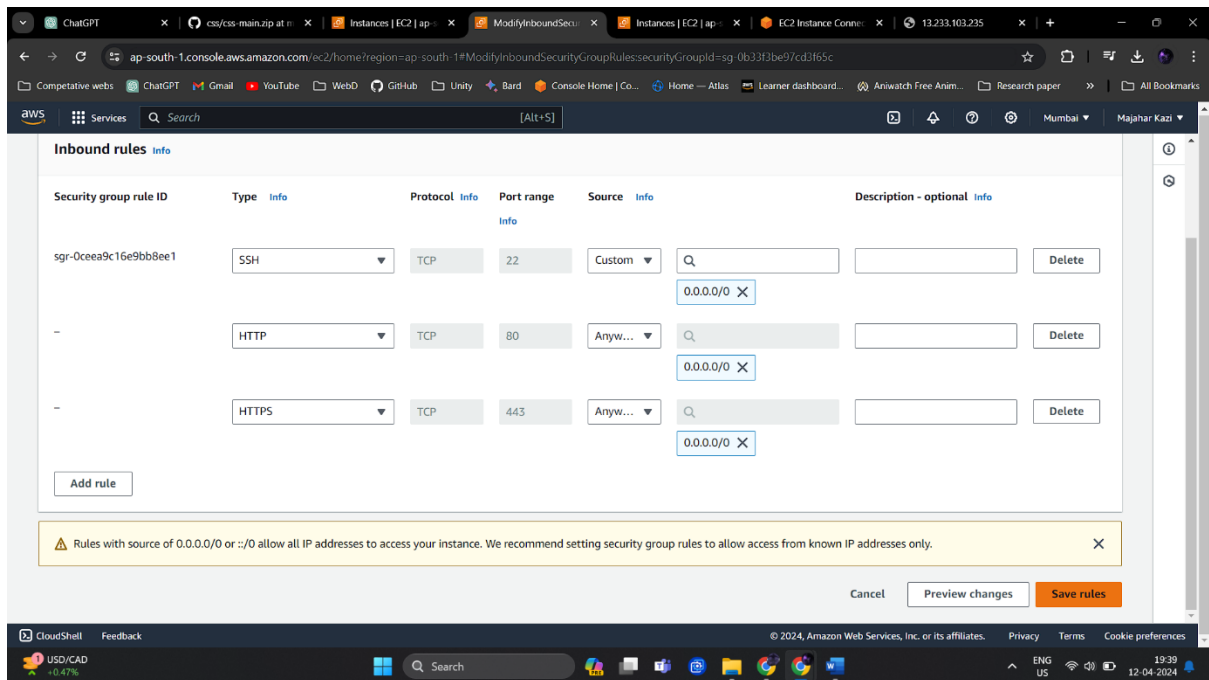
The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, and Snapshots. The main content area displays the 'Instances (1/2) info' page. A table lists two instances: 'Assignment 3' (ID: i-02f011d2d877f1aa8, state: Running) and 'Web deploy...' (ID: i-0ff61f4b985f105f, state: Terminated). Below the table, the details for 'Instance: i-02f011d2d877f1aa8 (Assignment 3)' are shown, including IAM Role, Owner ID, Launch time, Security groups, and Inbound rules. The 'Inbound rules' section shows a single rule with ID 'sgr-0ceea9c16e9bb8ee1' for port 22 (SSH) from source '0.0.0.0/0'.

Select the launch wizard, our security group.

The screenshot shows the AWS Management Console interface. The left sidebar is the same as the previous screenshot. The main content area displays the 'Security Groups (1/1) info' page. A table lists one security group: 'sg-0b33f3be97cd3f65c - launch-wizard-1'. Below the table, the details for this security group are shown, including the 'Inbound rules' section. The 'Inbound rules' section shows a single rule with ID 'sgr-0ceea9c16e9bb8ee1' for port 22 (SSH) from source '0.0.0.0/0'.

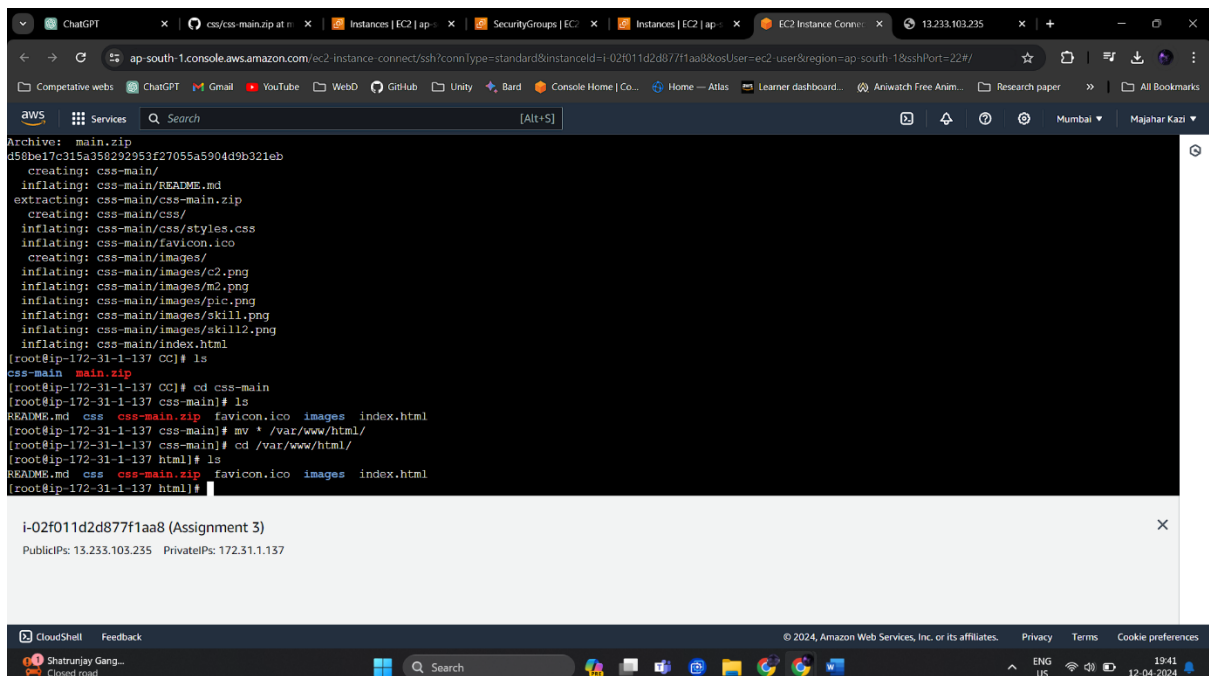


In the inbound rules select 'Edit inbound rules' and add rules as following.

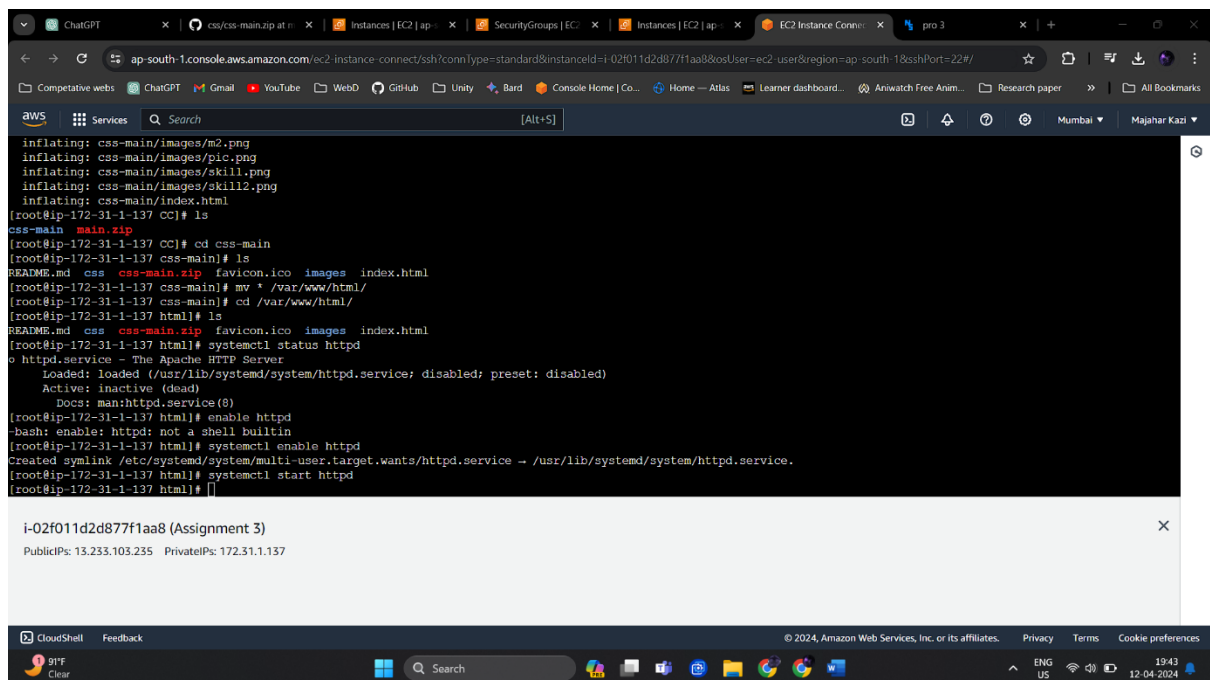


Save these rules and close.

In our vm, select all the files from our web-files directory and move them into '/var/www/html/'.



Now we need to enable httpd services for our vm.



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output displays the following commands and their results:

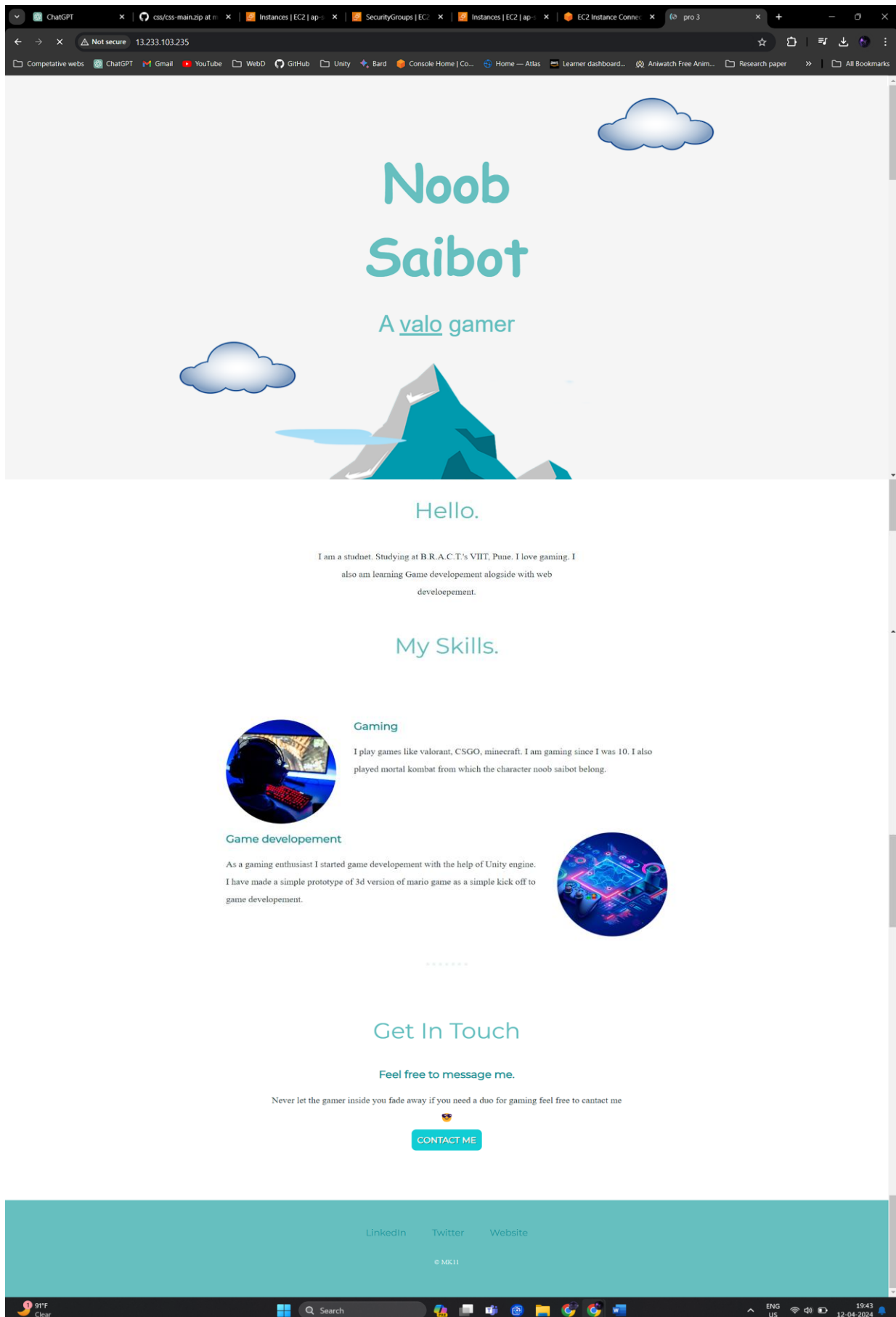
```
inflating: css-main/images/m2.png
inflating: css-main/images/pic.png
inflating: css-main/images/skill1.png
inflating: css-main/images/skill2.png
inflating: css-main/index.html
[root@ip-172-31-1-137 CC]# ls
css-main  main.zip
[root@ip-172-31-1-137 CC]# cd css-main
[root@ip-172-31-1-137 css-main]# ls
README.md  css  css-main.zip  favicon.ico  images  index.html
[root@ip-172-31-1-137 css-main]# mv * /var/www/html/
[root@ip-172-31-1-137 css-main]# cd /var/www/html/
[root@ip-172-31-1-137 html]# ls
README.md  css  css-main.zip  favicon.ico  images  index.html
[root@ip-172-31-1-137 html]# systemctl status httpd
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
         Docs: man:httpd.service(8)
[root@ip-172-31-1-137 html]# enable httpd
-bash: enable: httpd: not a shell builtin
[root@ip-172-31-1-137 html]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service -> /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-1-137 html]# systemctl start httpd
[root@ip-172-31-1-137 html]# []
```

Below the terminal output, a box displays the instance ID: `i-02f011d2d877f1aa8 (Assignment 3)` and its IP addresses: `PublicIPs: 13.233.103.235` and `PrivateIPs: 172.31.1.137`.

Now our website is successfully launched with the help of AWS EC2 instance.

**Result:**





---

Author – Majahar Kazi

Friday 12 April 2024 08:26:34 PM IST

