# 

# Practical File

# Cloud Computing (23CS009)

## BACHELOR OF ENGINEERING

***in***

## Computer Science and Engineering

#### Submitted by: Submitted to:

Arnav Singh Saini Dr. Sunil Kumar Chawla

2310990042 Designation -Associate Professor

G-21

Sem-5th

Year-3rd

**A black text on a white background

Description automatically generated**

**CHITKARA UNIVERSITY, PUNJAB**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY**

**RAJPURA (PATIALA) PUNJAB-140401 (INDIA)**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Practical No.** | **Practical Name** | **Page No.** |
|  | Sign up for AWS, explore free-tier services, and launch EC2 instances using Ubuntu/Linux and Windows operating systems. | 1 |
|  | Connect to an EC2 instance via EC2 Connect, SSH Client, AWS CLI, and PuTTY SSH Client using a key pair file.  Access and modify Security Group inbound and outbound rules to control traffic for EC2 instances. | 11 |
|  | Set up and configure web servers on EC2 instances, including Apache and Nginxserver. Configure auto-scaling for EC2 instances to handle variable traffic. | 18 |

**Practical No. 1**

**Practical Title: Sign up for AWS, explore free-tier services, and launch EC2 instances using Ubuntu/Linux and Windows operating systems.**

**Objective:**

## To create an AWS account and explore free-tier services.

## To launch an EC2 instance with Ubuntu and Windows OS.

## Step 1: AWS Account Sign In

1. Visit AWS Portal: Open the AWS Official Website <https://awsacademy.instructure.com/login/canvas> and fill the credentials and click on Log In.

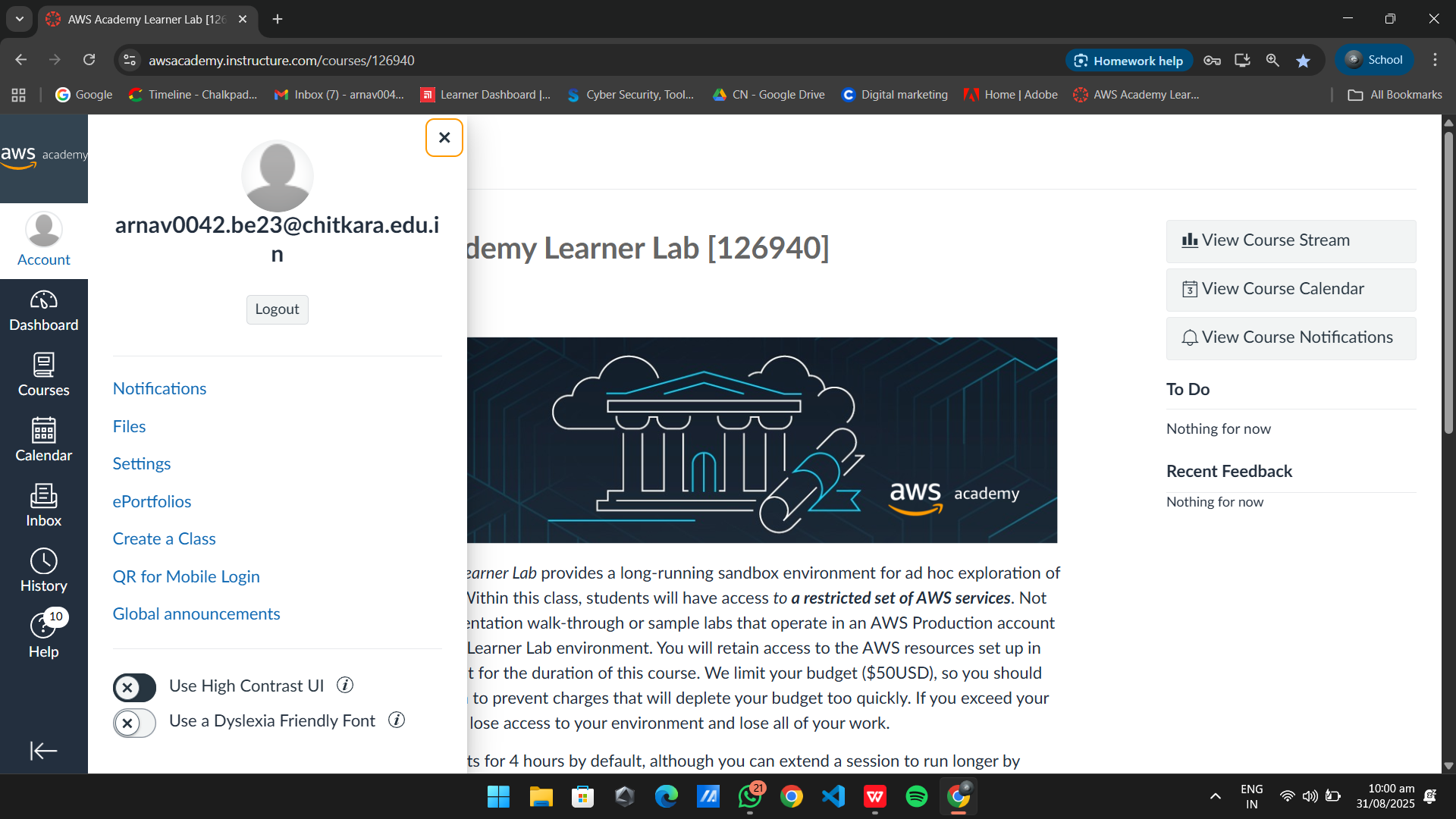


Fig-1.1

1. You will be successfully Logged in to your account and Dashboard will be visible.

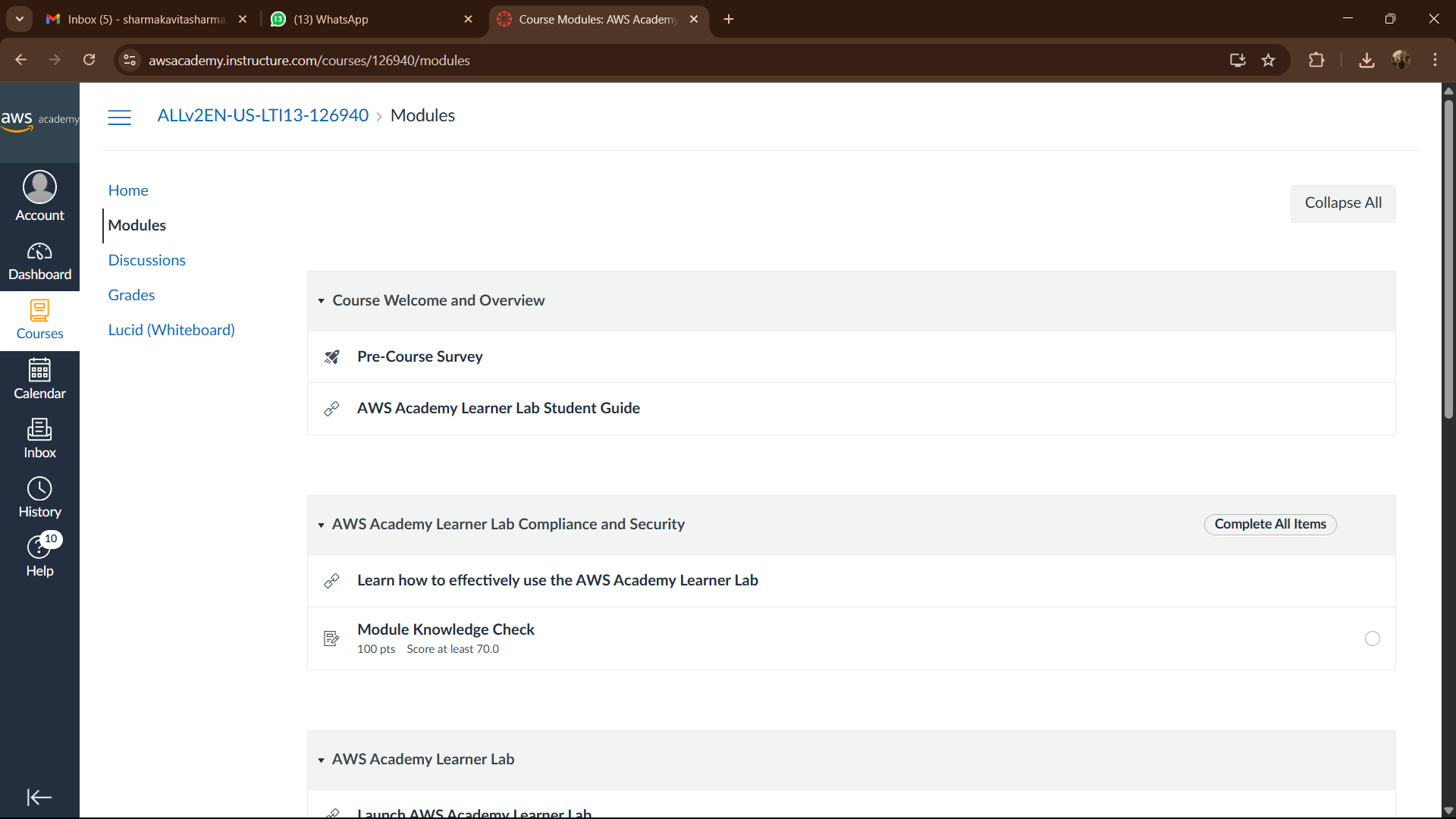
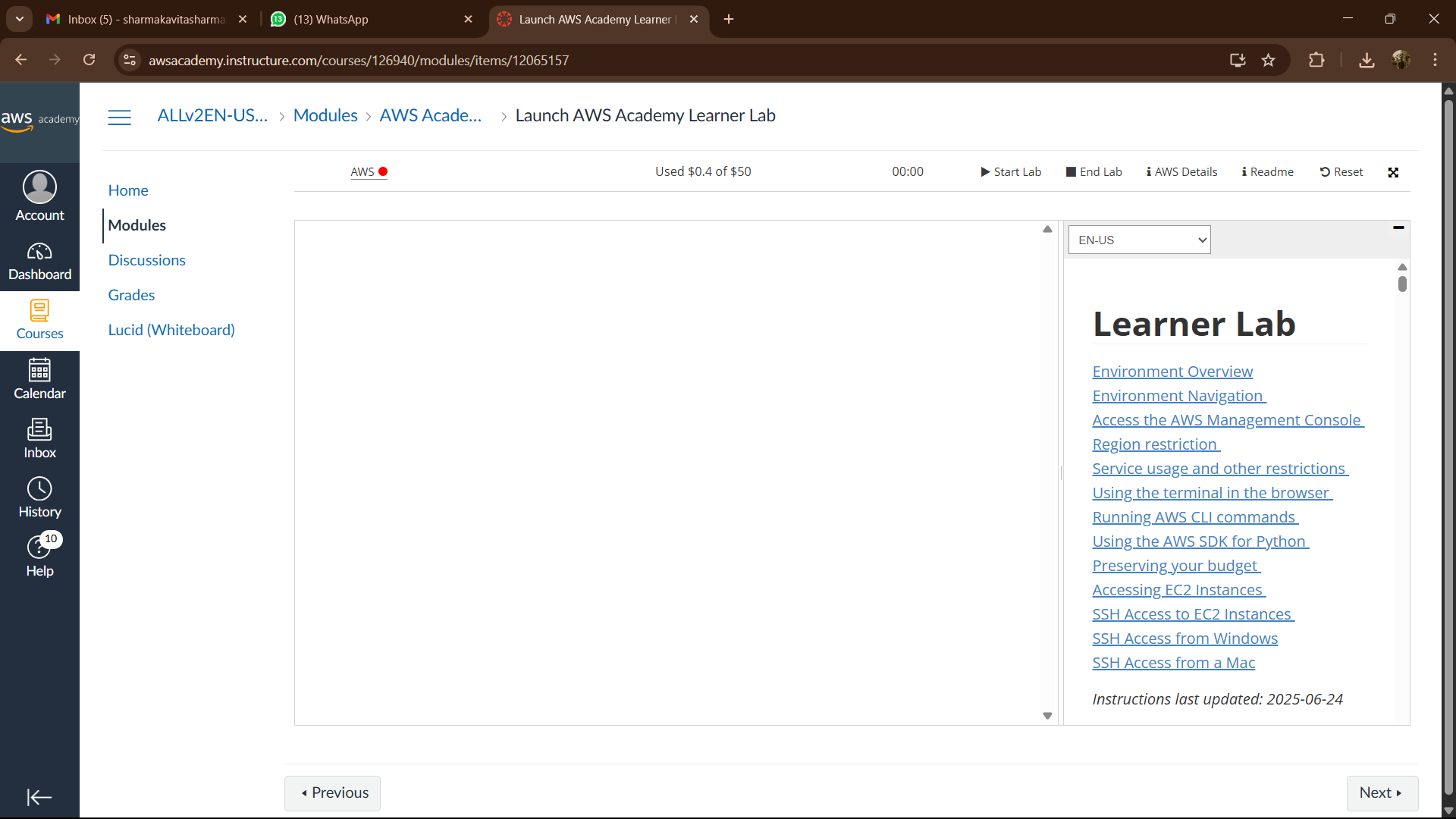


Fig-1.2

1. After navigating to Courses tab navigate to Modules and then click on “Launch AWS Academy Learner Lab”
2. After this click on Start Lab the Lab will be started.

## Step 2: Launching EC2 Instances (Ubuntu & Windows)

**(a) Launching an Ubuntu EC2 Instance and creating Web Server**

1. Click on EC2 on the console page.



Fig-1.3

2. Open EC2 Service: In the AWS Management Console, navigate to the EC2 service and click on Launch Instance.

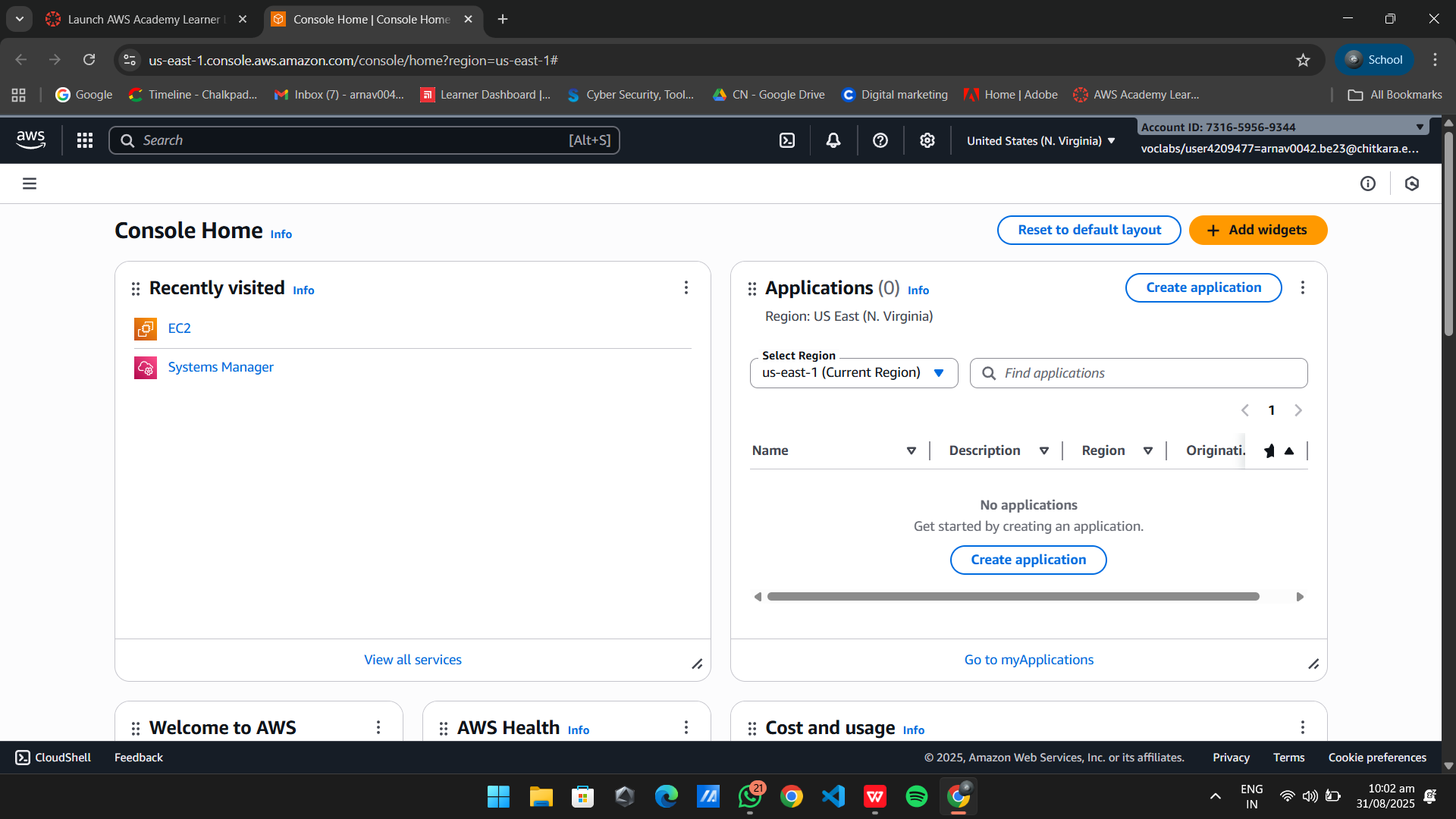


Fig-1.4

3. Choose Amazon Machine Image (AMI): Select Ubuntu 22.04 LTS and enter instance name.

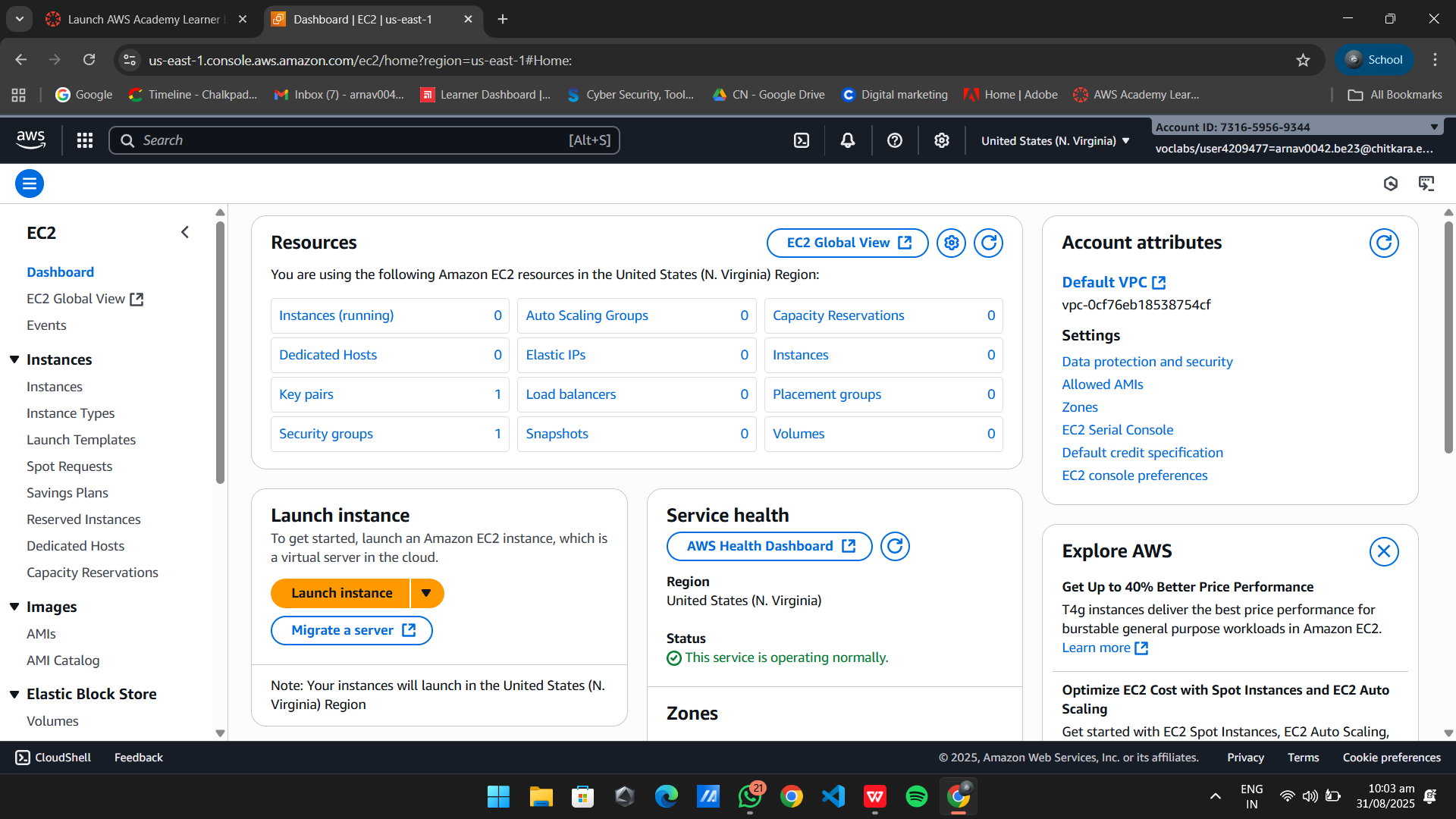


Fig-1.5

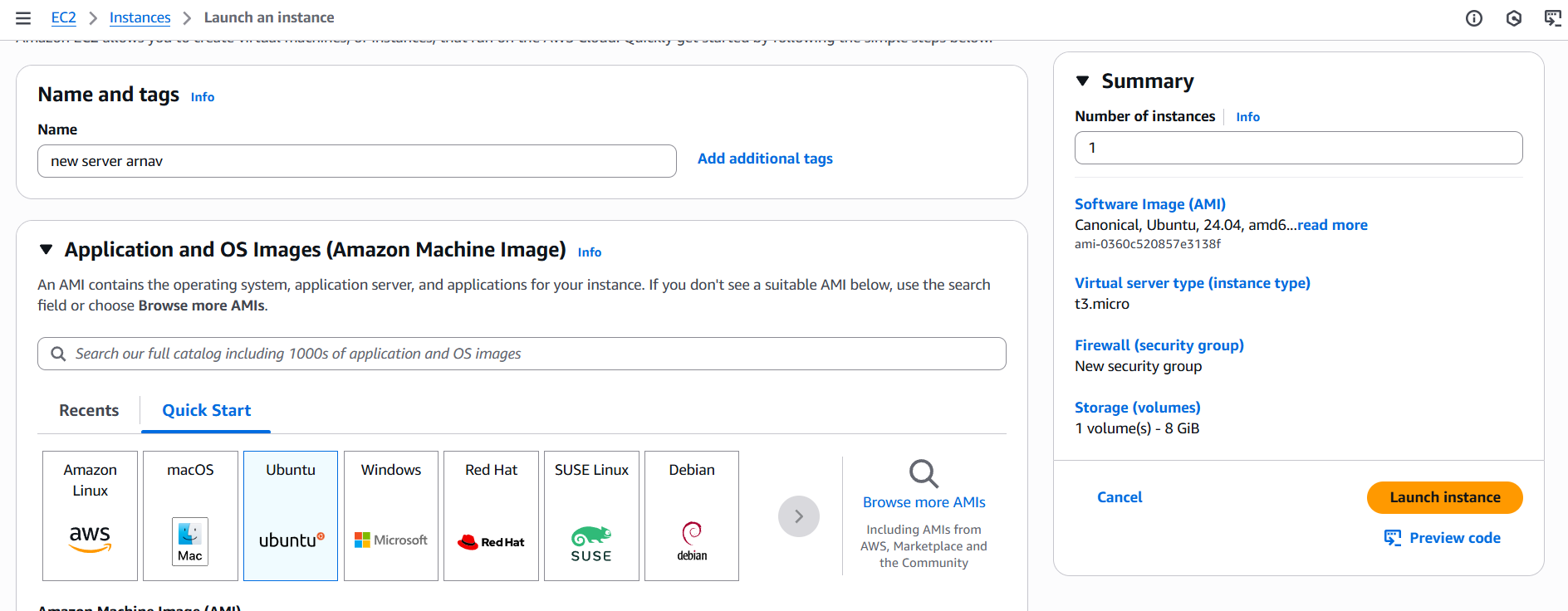
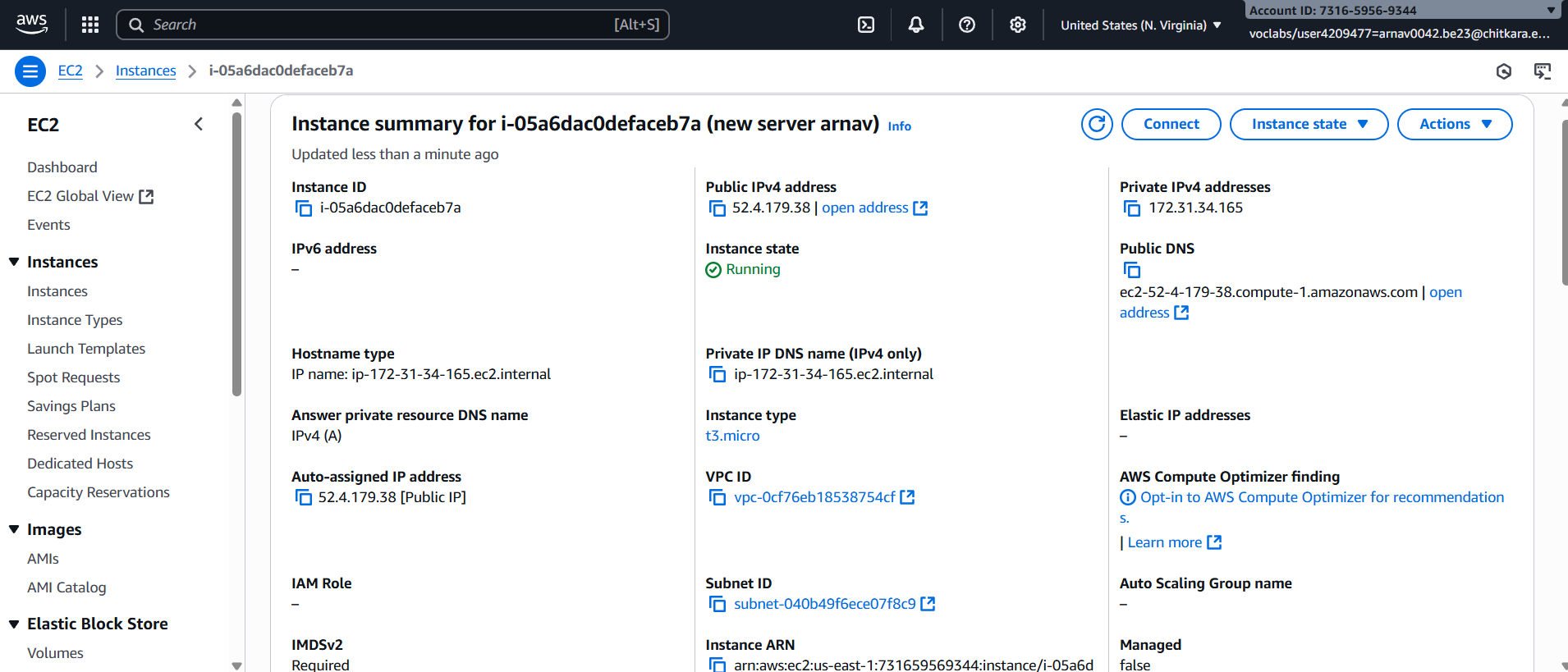


Fig-1.6

1. Configure Instance: You can keep the default settings for this step and then click on Launch Instance.



1. **==**Connect To Instance: After Launching the click on connect on the top right corner.

Fig-1.7

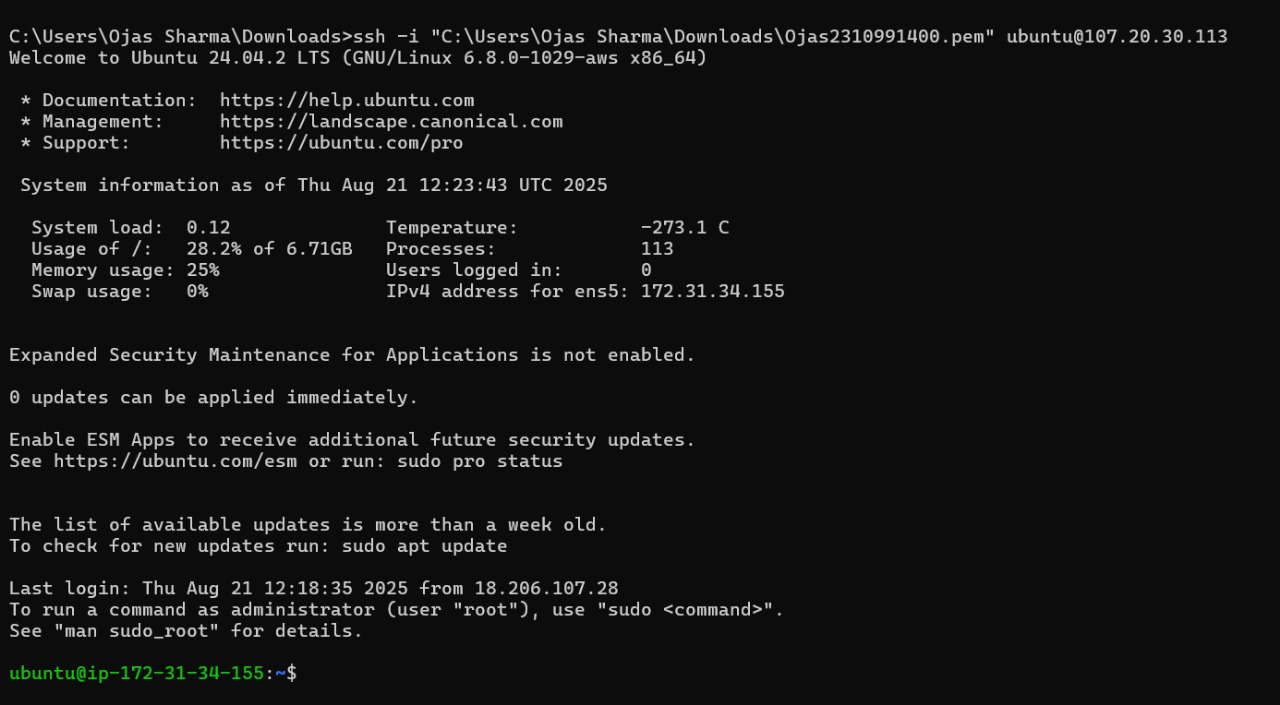
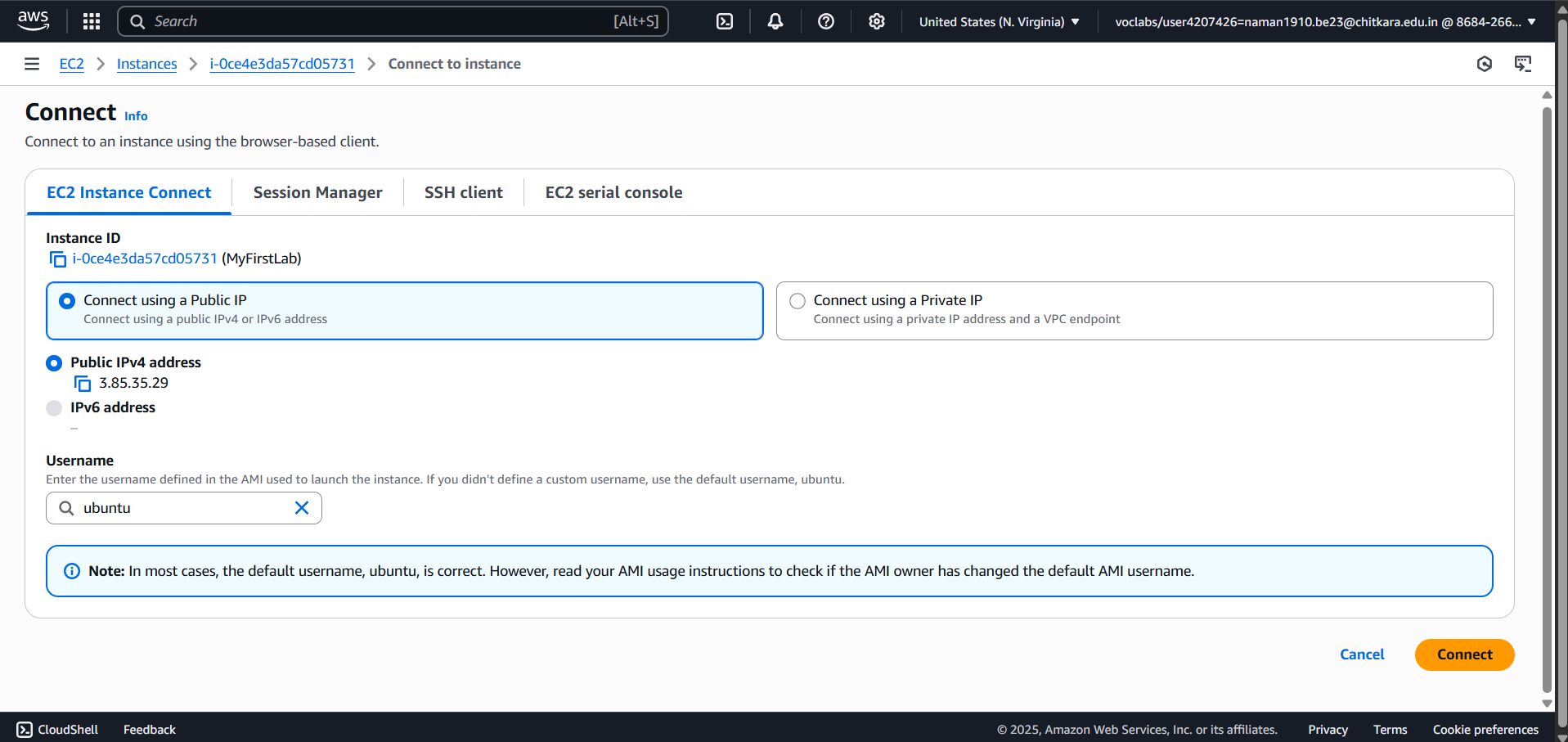


Fig-1.8

1. Click on Connect button
2. We have successfully connected to the Ubuntu instance

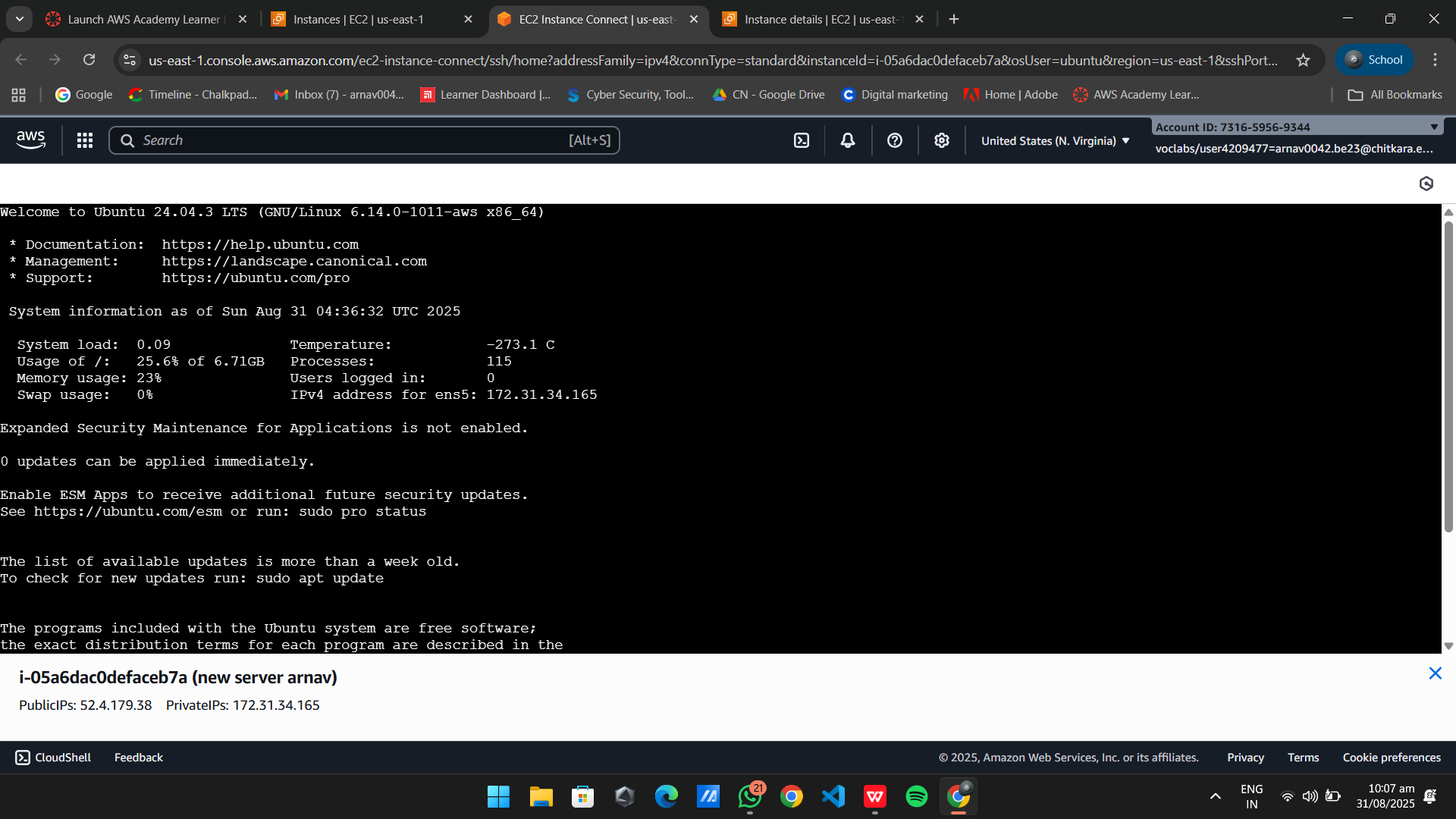


Fig-1.9.1

**(b) Launching a Windows EC2 Instance.**

1. Repeat Steps 1-3 from the Ubuntu launch: This time, choose the Windows Server 2025 AMI.

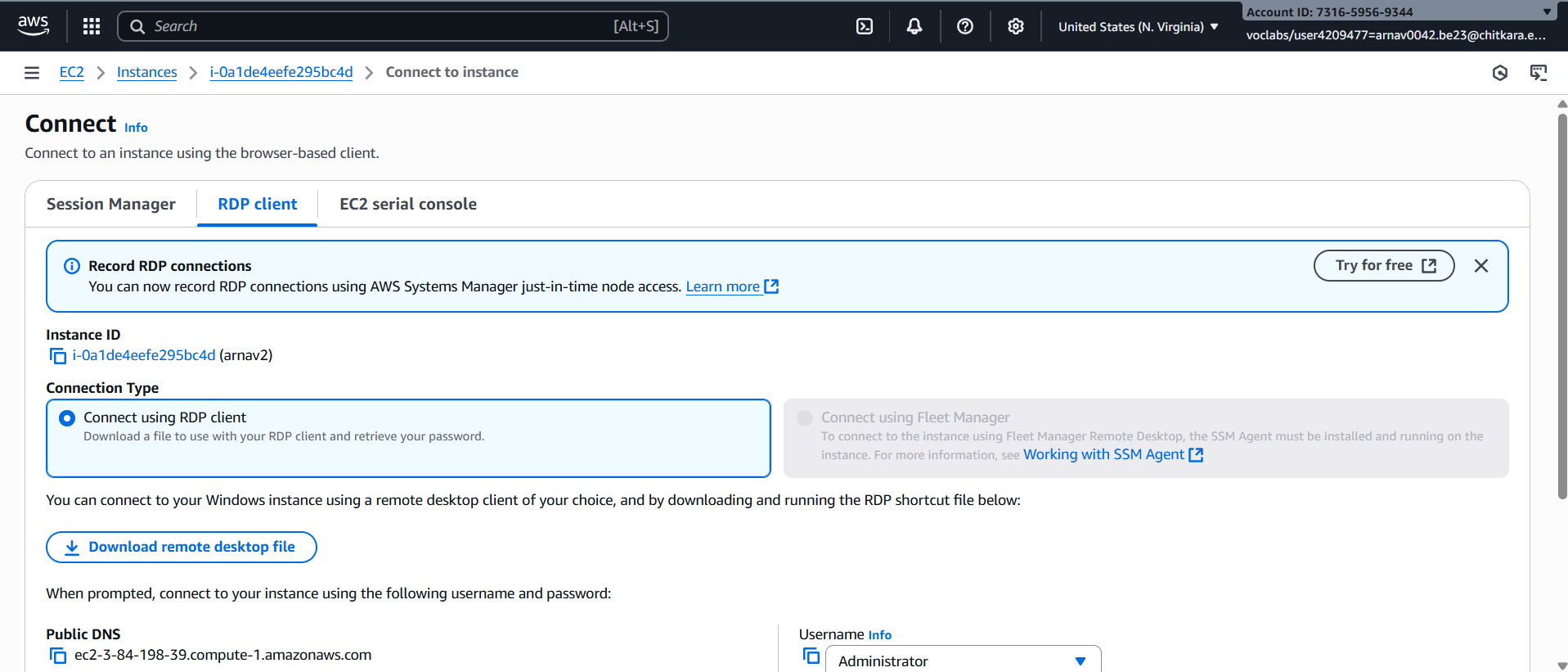


Fig-1.9.2

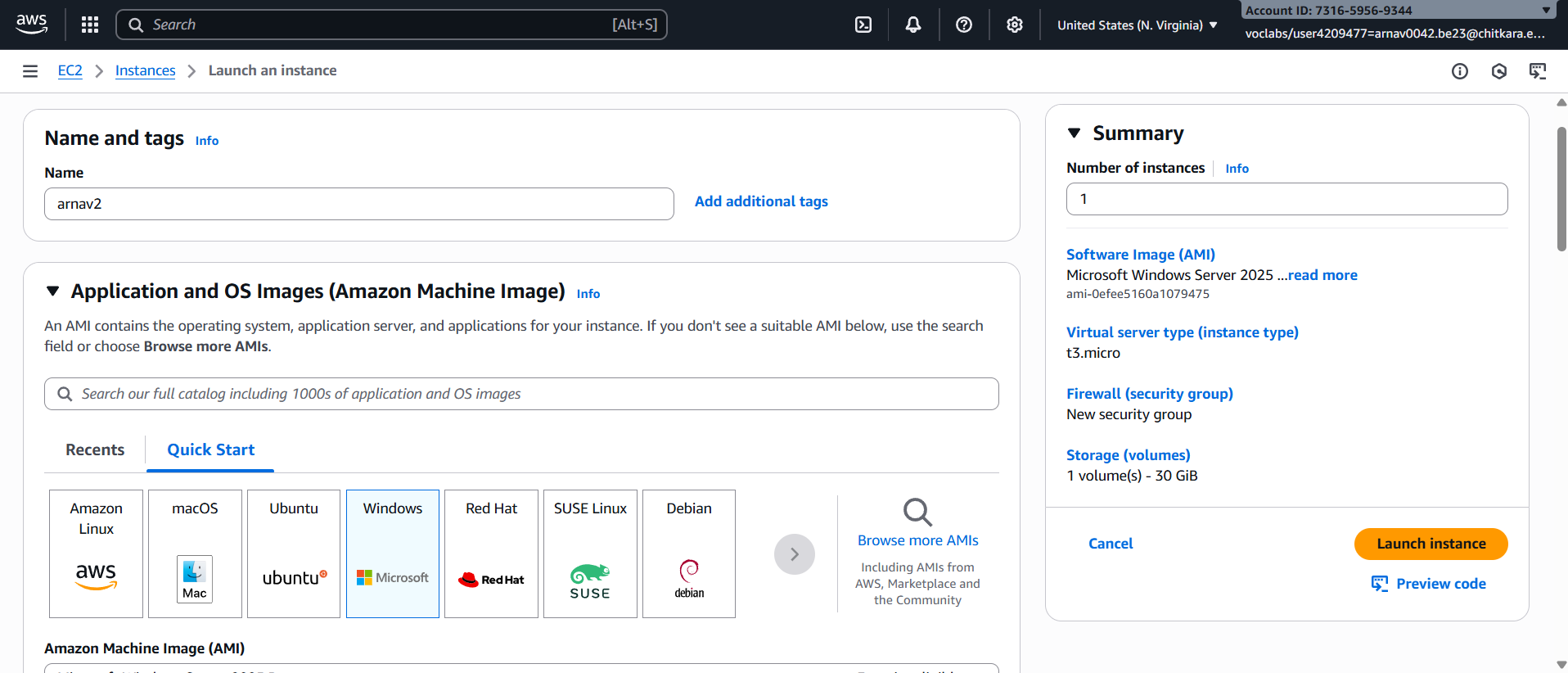
1. Configure Security Group: Create a new security group that allows RDP (Port 3389) traffic for remote desktop access and click on launch

Fig-1.9.3

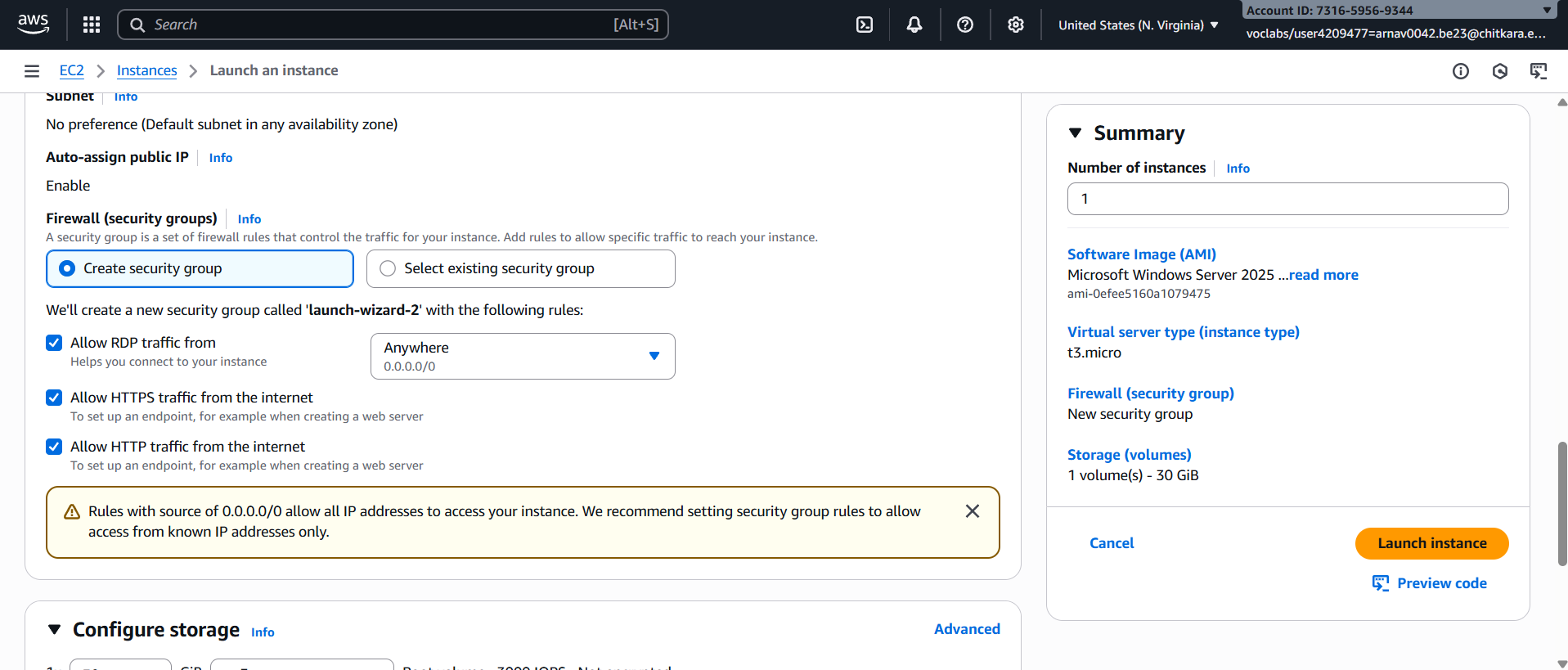


Fig-1.9.4

1. Connect using RDP client: Download the remote desktop file and then click on Get Password.
2. Generate Password: Upload the private key file and then click on decrypt password.
3. Copy the Password and connecting RDP client: Copy the password and then launch the downloaded RDP file.

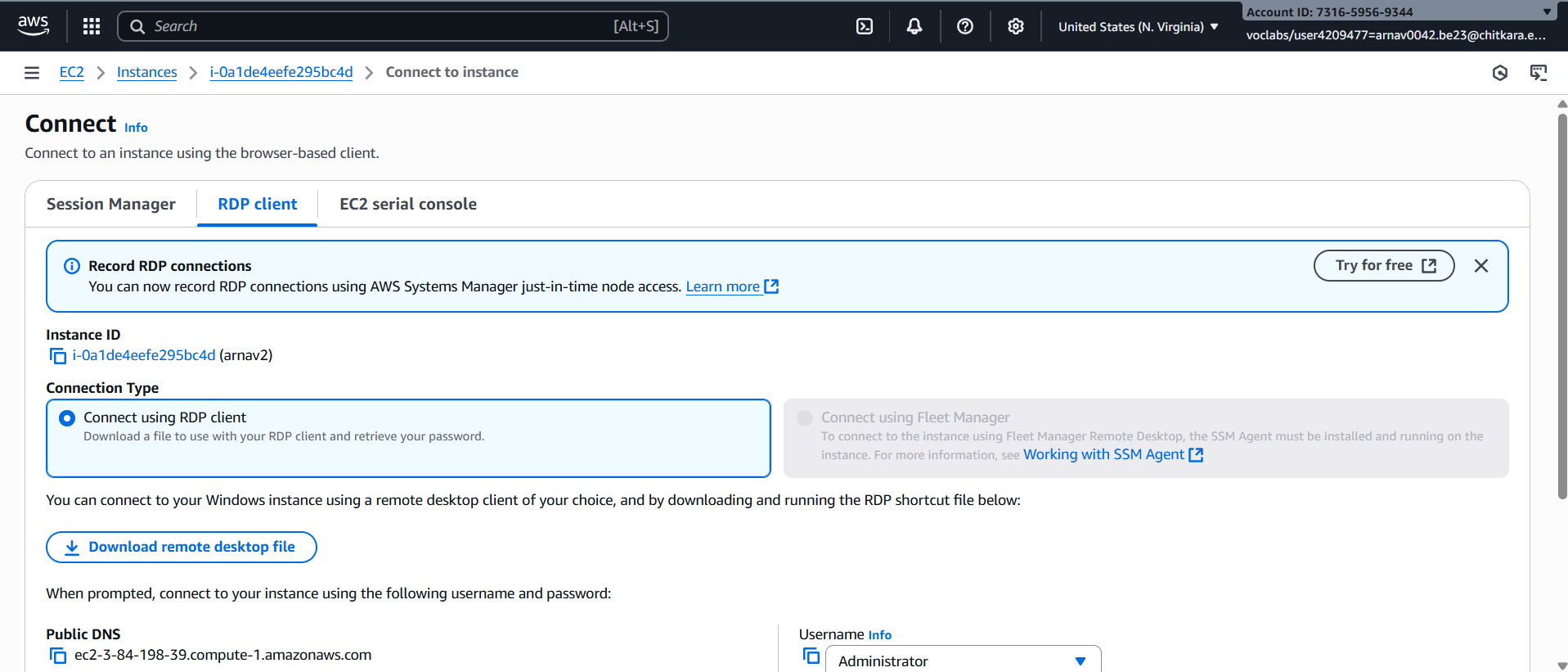


Fig-1.9.5

1. Enter Password: Enter the copied password and then Click OK.

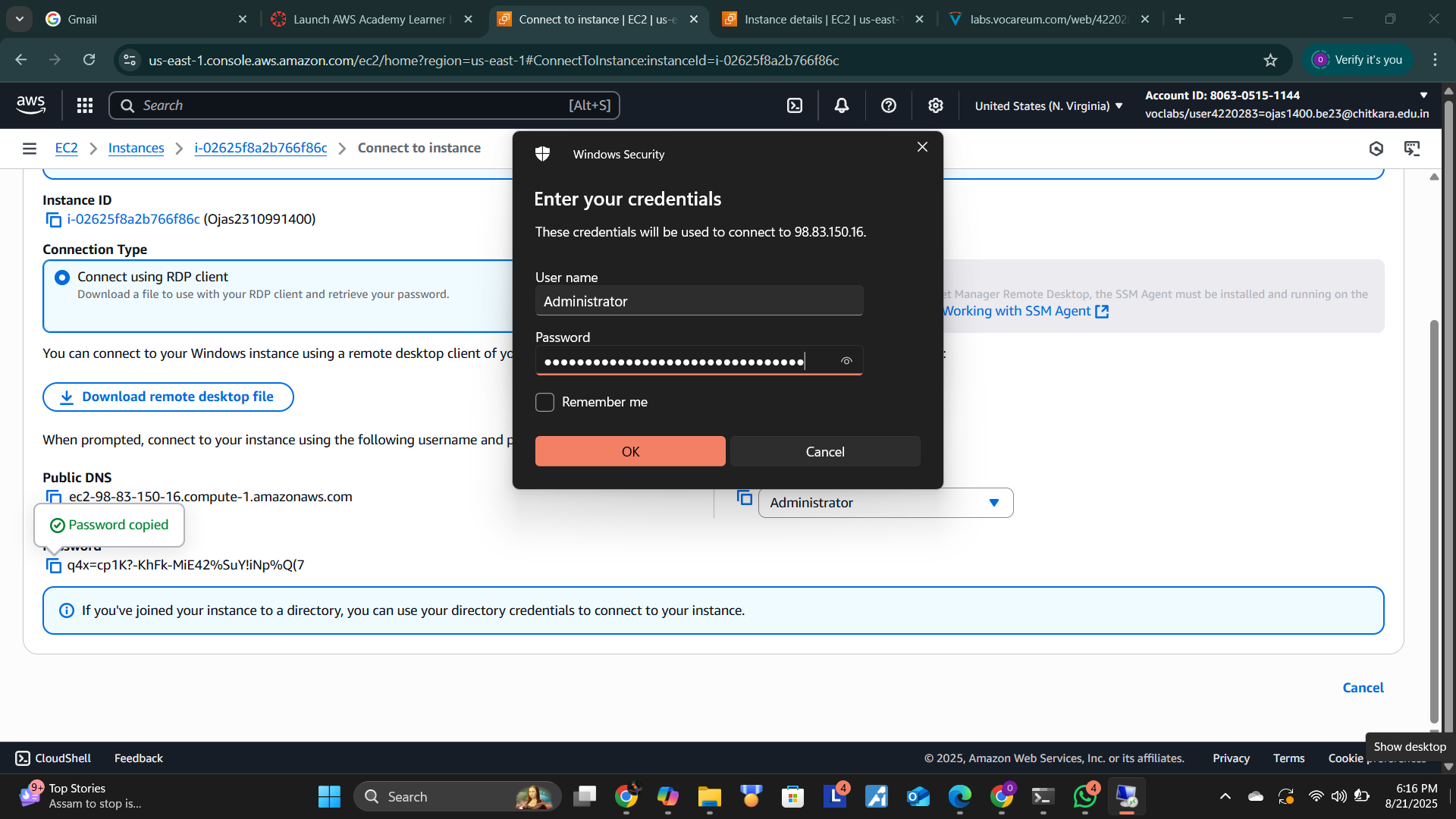
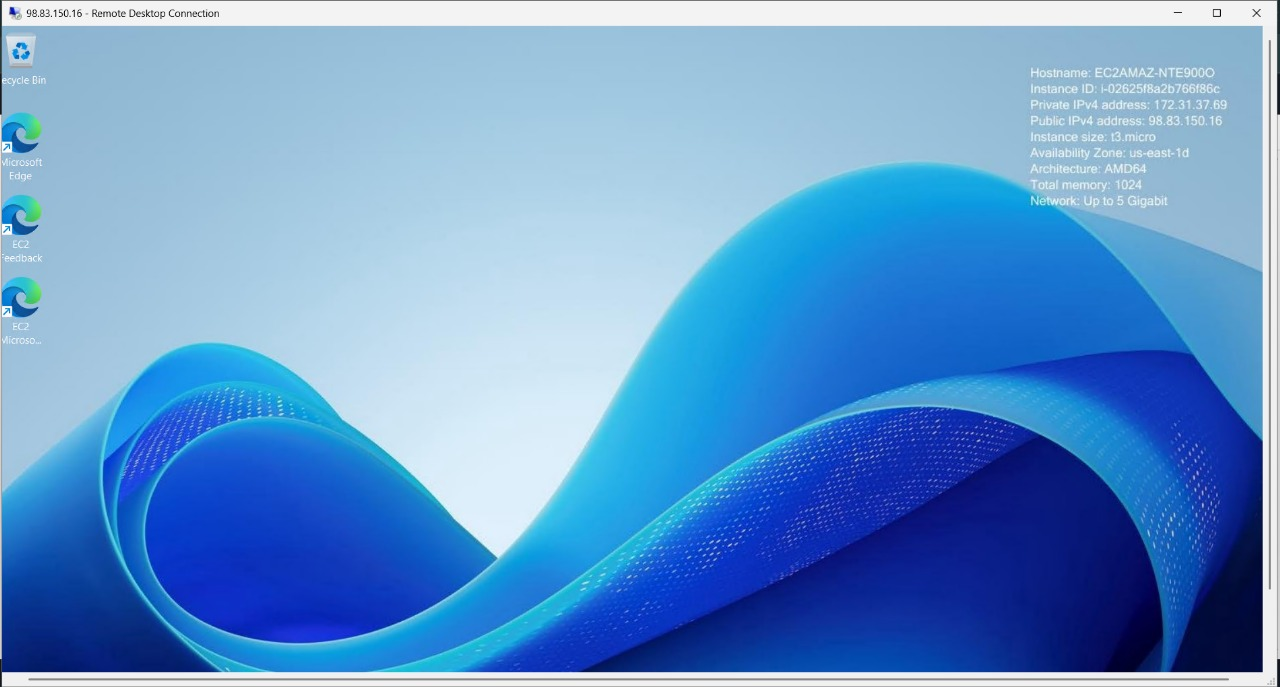


Fig-1.9.6



1. RDP connection is successful and we can see the Remote Desk.

**Learning Outcomes:**

* I successfully signed up for an AWS account and learned how to navigate and explore the free-tier services.
* I can now launch and connect to virtual machine instances using both Ubuntu (Linux) and Windows operating systems on AWS.
* I understand that AWS provides a free-tier environment which is useful for learning and experimentation in cloud computing.
* I can create EC2 instances and access them remotely using appropriate clients like SSH for Ubuntu and RDP for Windows.
* This experiment helped me understand the fundamental steps of launching and managing virtual machines in the AWS cloud.

**Practical No. 2**

**Practical Title: Connect to an EC2 instance via EC2 Connect, SSH Client, AWS CLI, and PuTTY SSH Client using a key pair file.**

**Access and modify Security Group inbound and outbound rules to control traffic for EC2 instances.**

**Objective:**

* **Securely connect to a Linux-based EC2 instance using multiple methods: EC2 Instance Connect, a standard SSH client, the AWS CLI, and the PuTTY SSH client on Windows.**
* **Understand and manage EC2 Security Groups to control inbound and outbound network traffic to the instance.**

## Step 1: Launching an EC2 Instance and Creating a Key Pair

1. Log in to your AWS Management Console. In the services search bar, type "EC2" and select it.

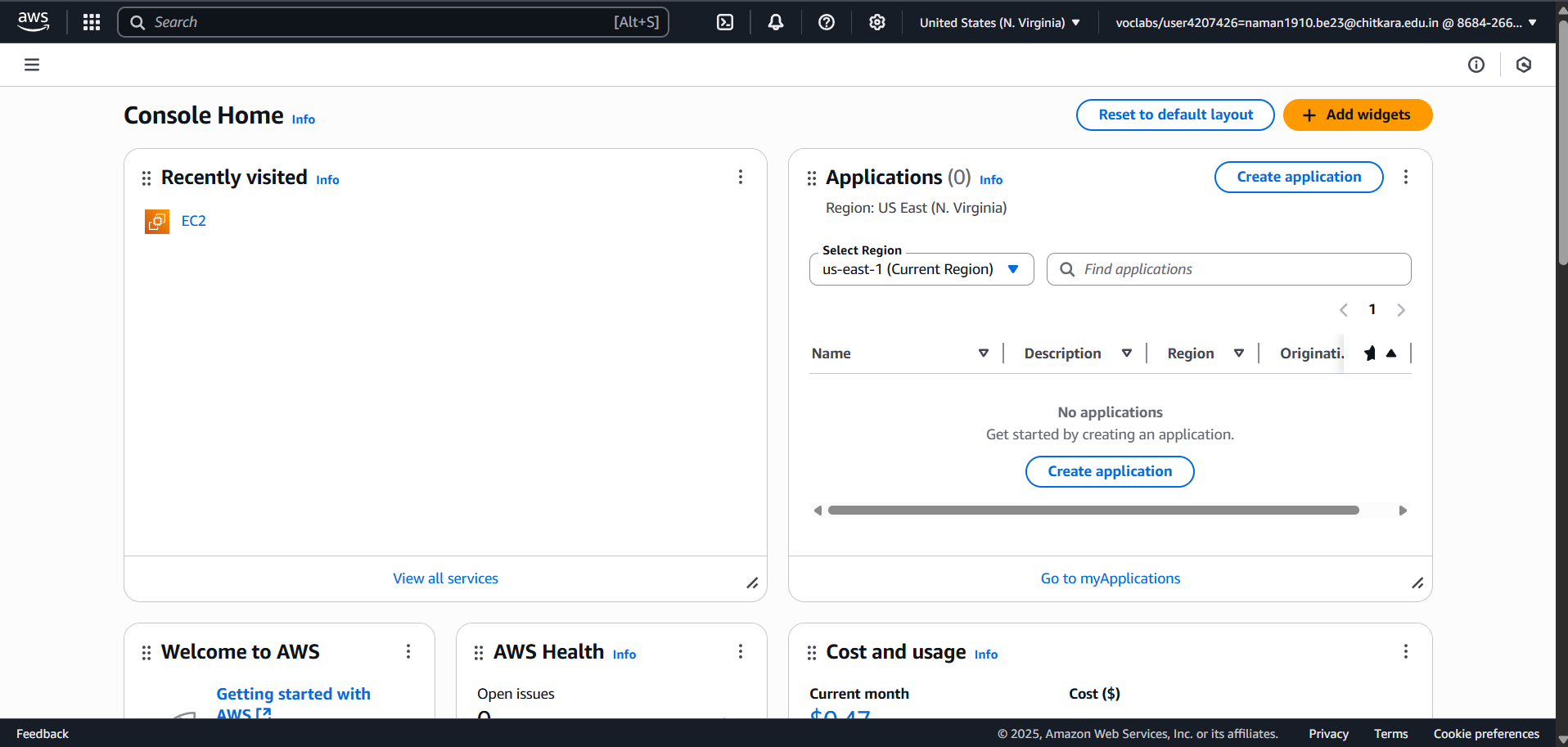
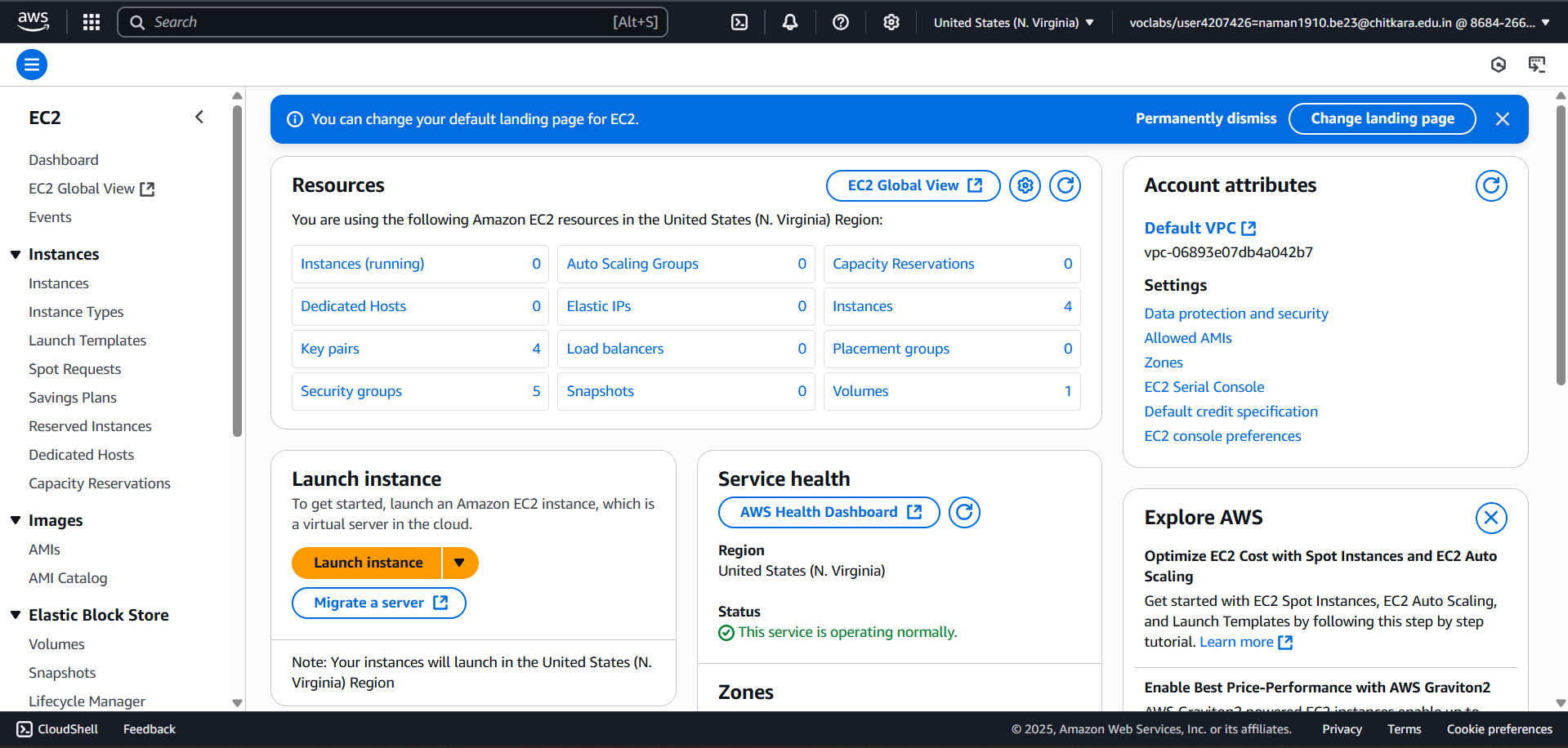
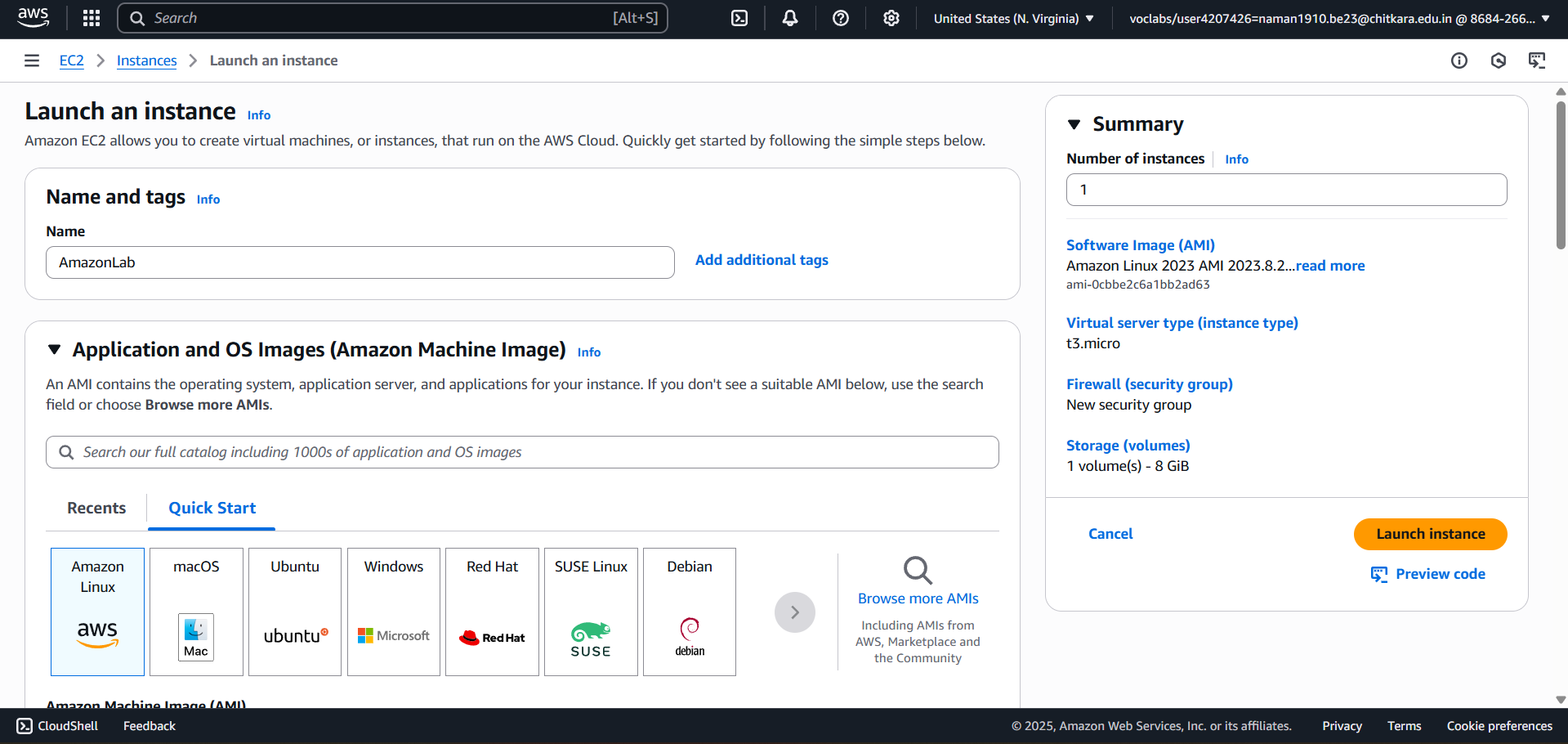


Fig 2.1

1. From the Dashboard, click the “Launch instance” button.

Fig 2.2

1. Give your instance a descriptive name. Select "Amazon Linux 2023 kernel-6.1 AMI". Choose t3.micro as instance type.



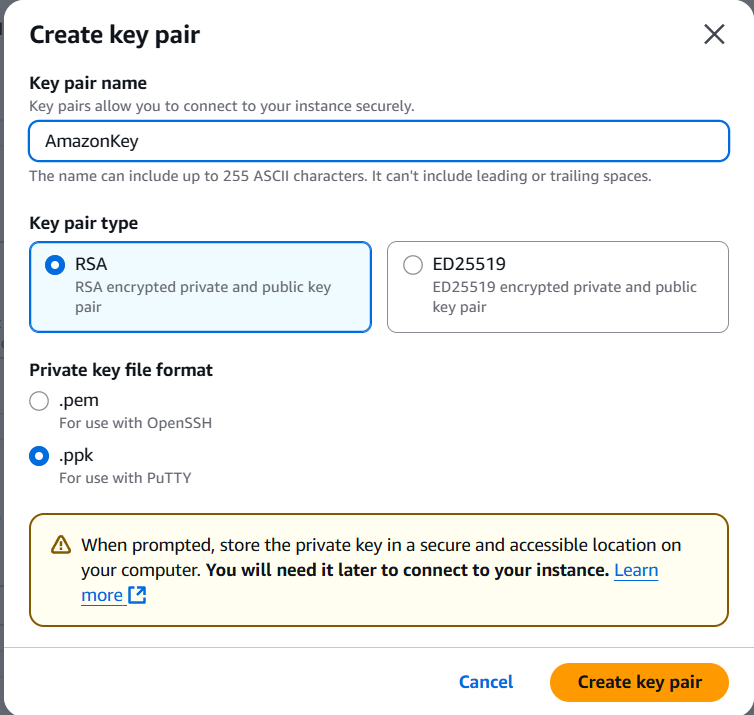


Fig-2.3

1. In the "Key pair (login)" section, click "Create new key pair". Enter a name for key-pair. Key pair type: RSA. Select .ppk if you are using PuTTY on Windows. Click "Create key pair". Your browser will download the private key file. Store this file securely; you cannot download it again.
2. Review the summary and click "Launch instance". It will take a few minutes for the instance to launch and pass its status checks. You can view its status on the "Instances" screen.

## Step 2: Modifying Security Group Rules

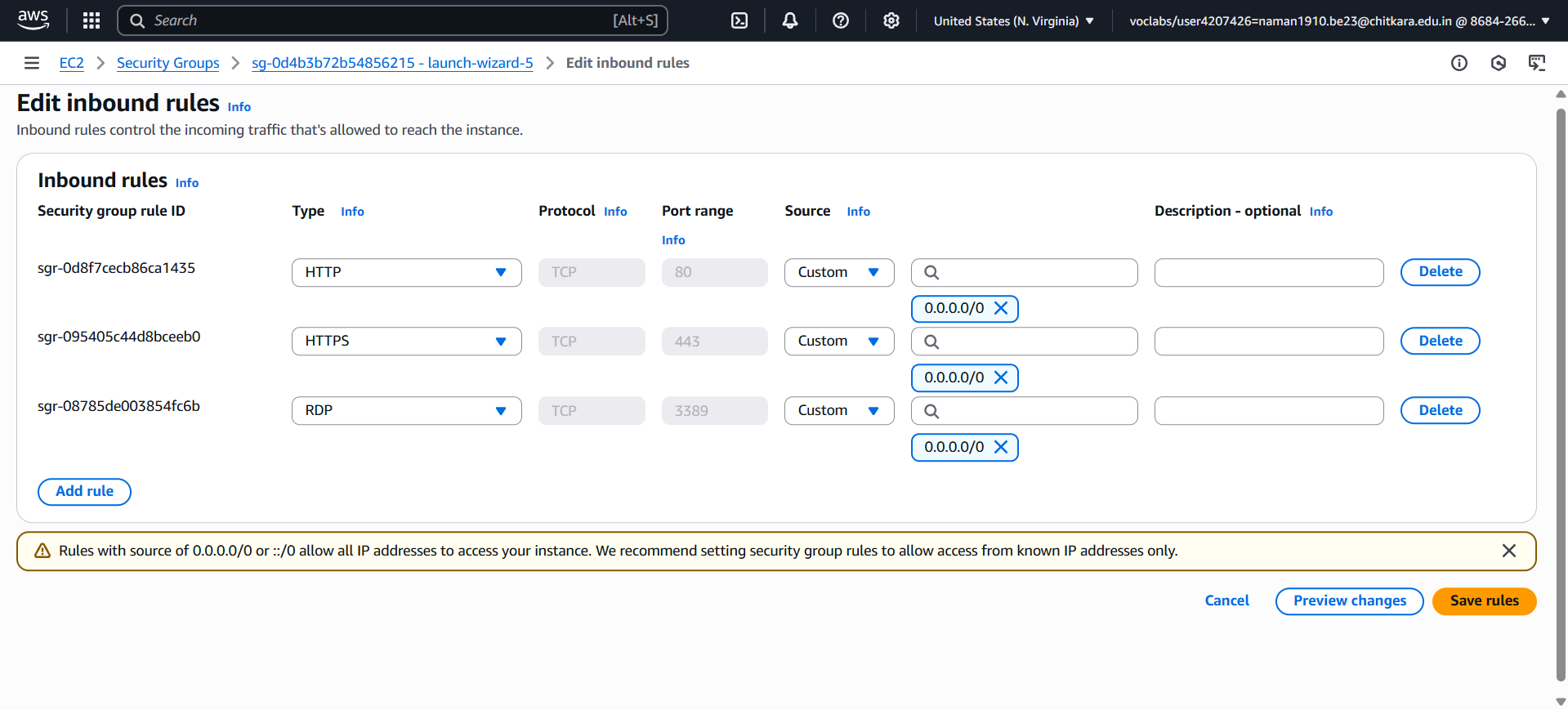
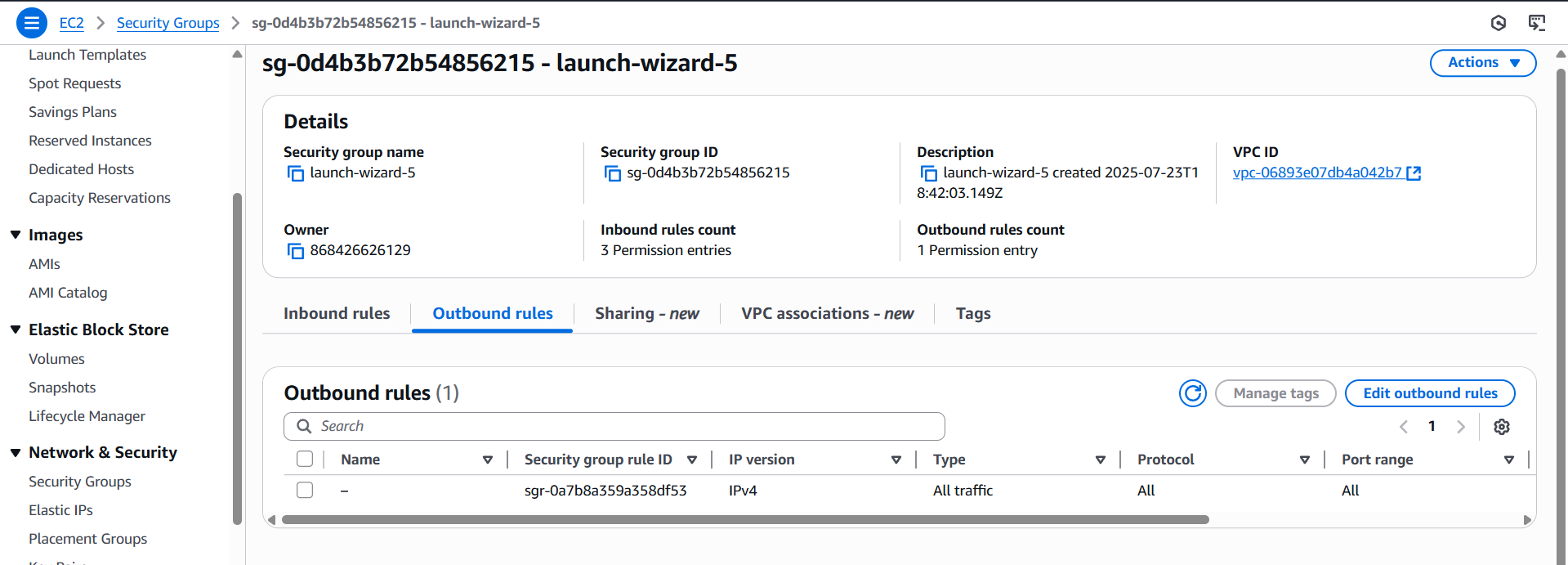
1. In the left-hand navigation pane of the EC2 dashboard, under "Network & Security," click "Security Groups".
2. Find and select the security group associated with your instance.
3. Select the "Inbound rules" tab at the bottom and click "Edit inbound rules". You should see your SSH rule on port 22. Click "Add rule". Select HTTP. Select Anywhere-IPv4 (0.0.0.0/0). This allows anyone to access a potential web server on your instance. Click "Save rules".

Fig-2.4

1. Select the "Outbound rules" tab. By default, all outbound traffic is allowed. This is a standard configuration that allows your instance to download updates and connect to
2. other services. No changes are needed here.

## 

## Step 3: Connecting to the EC2 Instance

**Method 1: EC2 Instance Connect**

1. Select your running instance from the list.
2. Click the "Connect" button at the top of the panel.
3. On the "Connect to instance" page, select the "EC2 Instance Connect" tab.
4. The User name (ec2-user for Amazon Linux) will be pre-filled.
5. Click "Connect". A new browser tab will open with a terminal session to your instance.
6. 

Fig-2.5

**Method 2: SSH Client**

1. Open PowerShell or Command Prompt.
2. Navigate to the directory where you saved your .pem file.
3. Use the following SSH command format. “ssh -i "my\_aws\_key.pem" [ec2-user@ec2-100-24-73-145.compute-1.amazonaws.com](mailto:ec2-user@ec2-100-24-73-145.compute-1.amazonaws.com)”
4. The first time you connect, you will be asked to verify the host's authenticity. Type yes and press Enter.

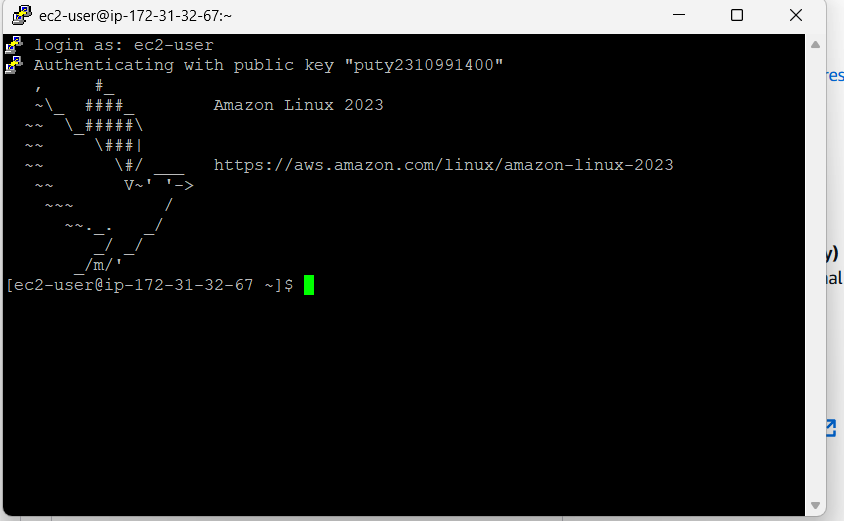


Fig-2.6

**Method 3: PuTTY SSH Client**

1. Open PuTTY. Session -> Host Name (or IP address): Enter your instance's Public IPv4 address.
2. Connection -> SSH -> Auth -> Credentials: Click "Browse..." and select your .ppk file.
3. Go back to the "Session" screen. Enter a name under "Saved Sessions" and click "Save".
4. Click "Open" to start the connection.

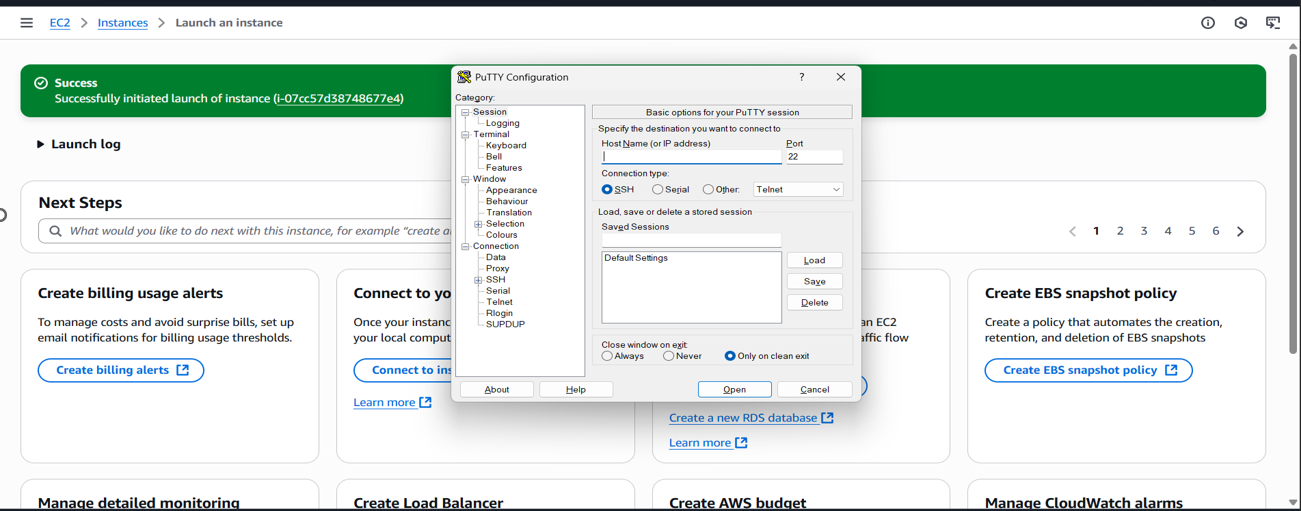


Fig-2.7

1. A terminal window will open. When prompted for "login as:", enter ec2-user.

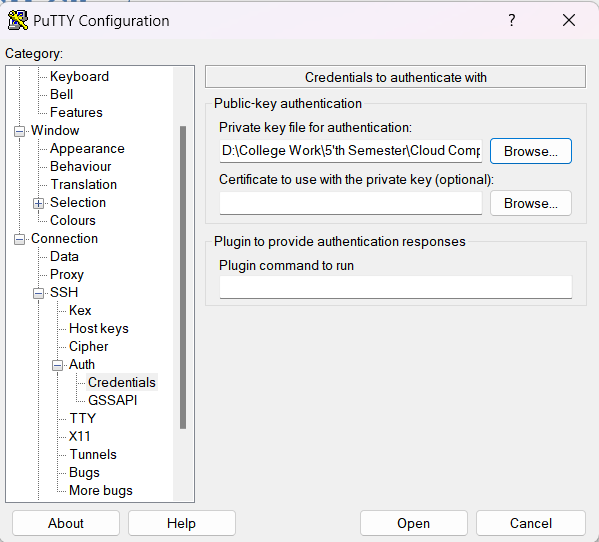
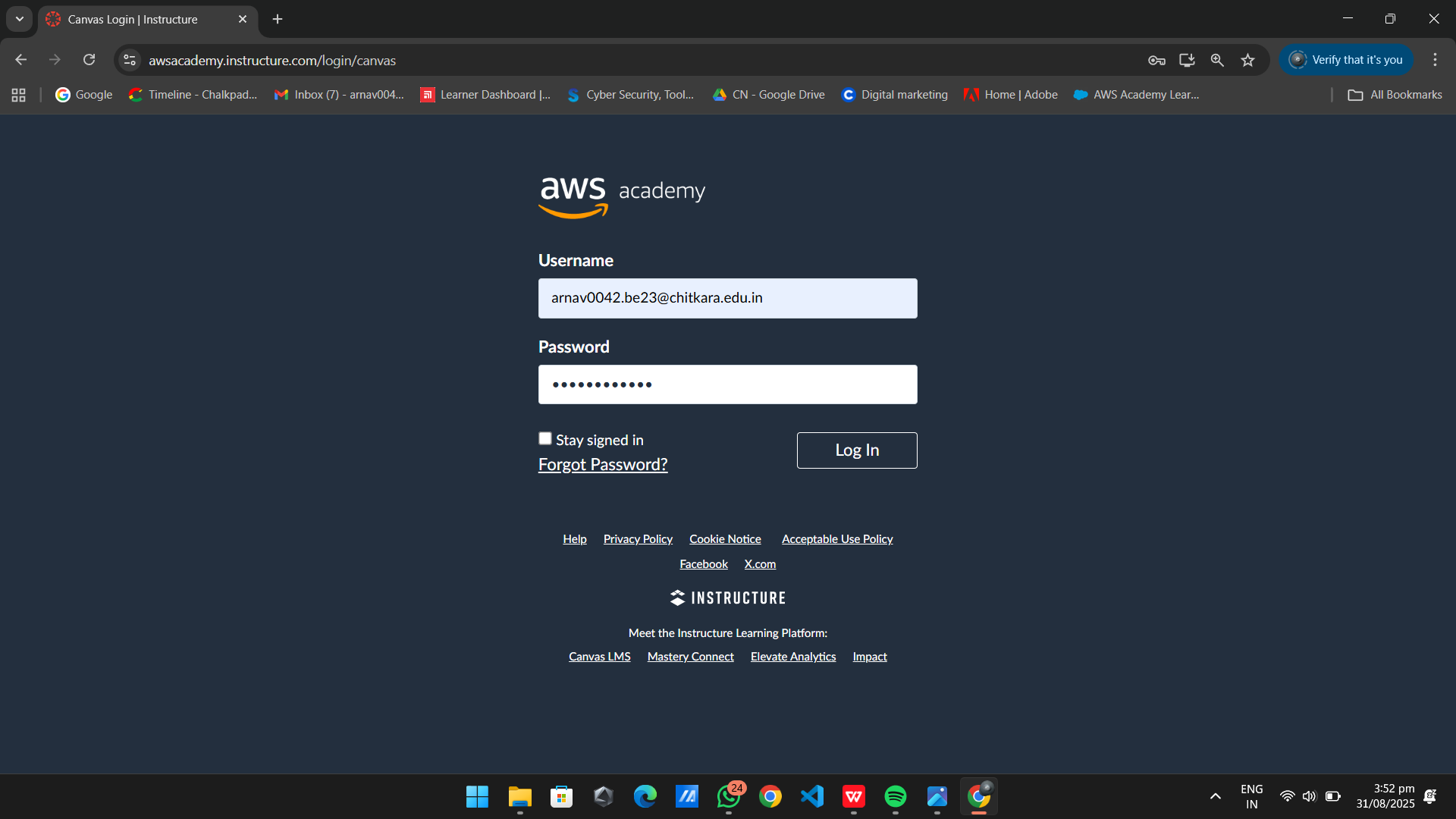
****

Fig-2.8

1. Click "Open" to start the connection.
2. A terminal window will open. When prompted for "login as:", enter ec2-user.

**Learning Outcomes:**

* EC2 Instance Management: You can now launch a basic EC2 instance and understand the core configuration options like AMI and instance type.
* Key Pair Management: You understand the role of key pairs in securing SSH access and can create and use them in .pem and .ppk formats.
* Network Security: You are able to define and modify Security Group rules to control inbound
* **Practical No. 3**
* **Practical Title:**
* **Set up and Configure Web Servers on EC2 Instances (Apache & Nginx) with Auto Scaling**
* **Objective:**
* **To set up and configure Apache and Nginx web servers on AWS EC2 instances.**
* **To configure Auto Scaling for EC2 instances to handle variable traffic efficiently.**
* **Step-by-Step Procedure with Screenshots:**
* **Step 1: Login to AWS Console**
* **Visit AWS Management Console and sign in with your credentials.**
* **Navigate to the EC2 service from the dashboard.**



**Fig-3.1.1**

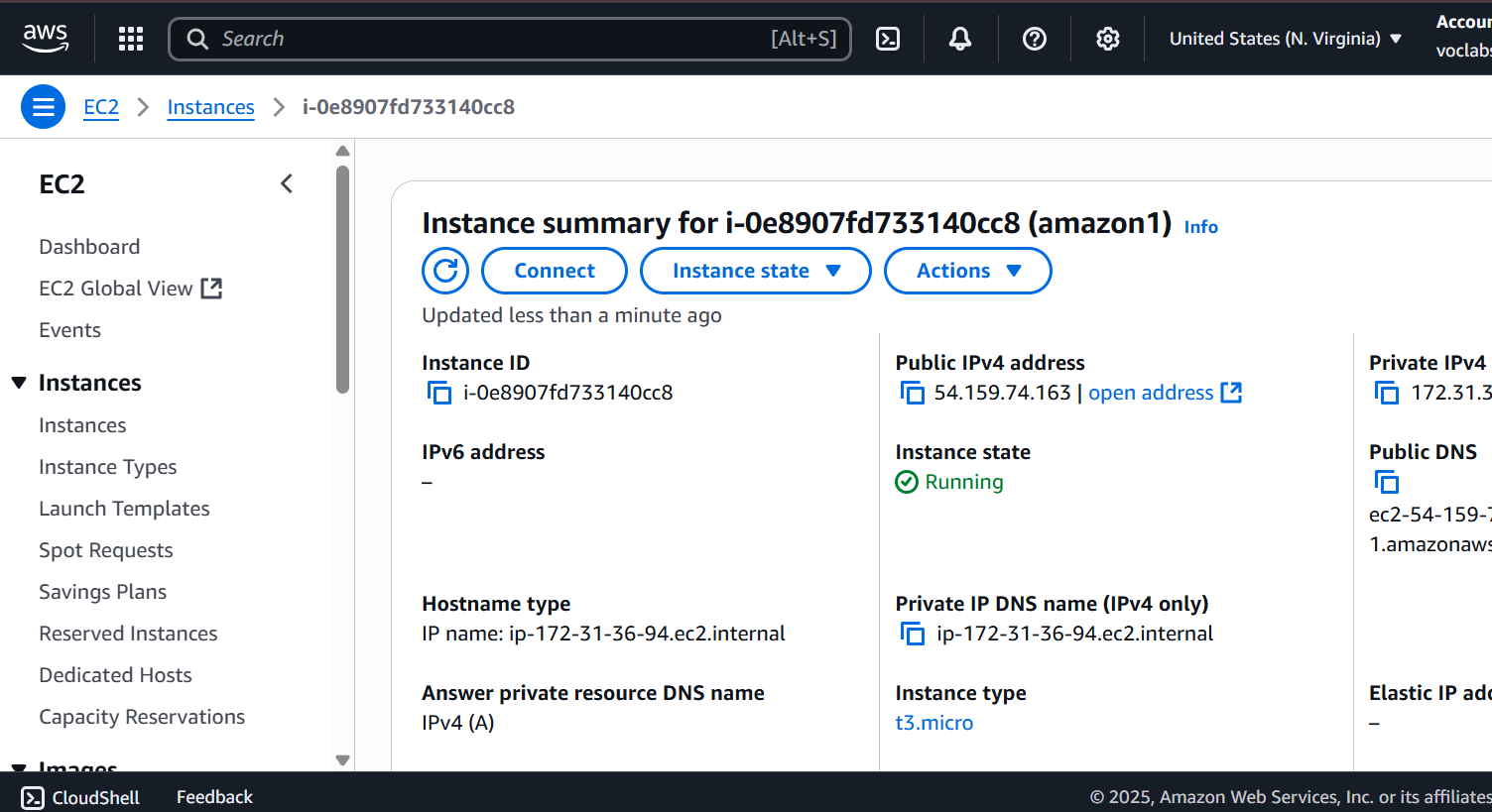
* **Step 2: Launch EC2 Instance**
* **Go to EC2 → Launch Instance.**
* **Choose Amazon Linux 2 AMI / Ubuntu 22.04.**
* **Select t2.micro (Free Tier).**
* **Create or select a Key Pair for SSH access.**
* **Configure Security Group:**
* **Allow Port 22 (SSH)**
* **Allow Port 80 (HTTP)**
* **Allow Port 443 (HTTPS)**
* **Launch the instance.**
* ****

Fig-3.1.2

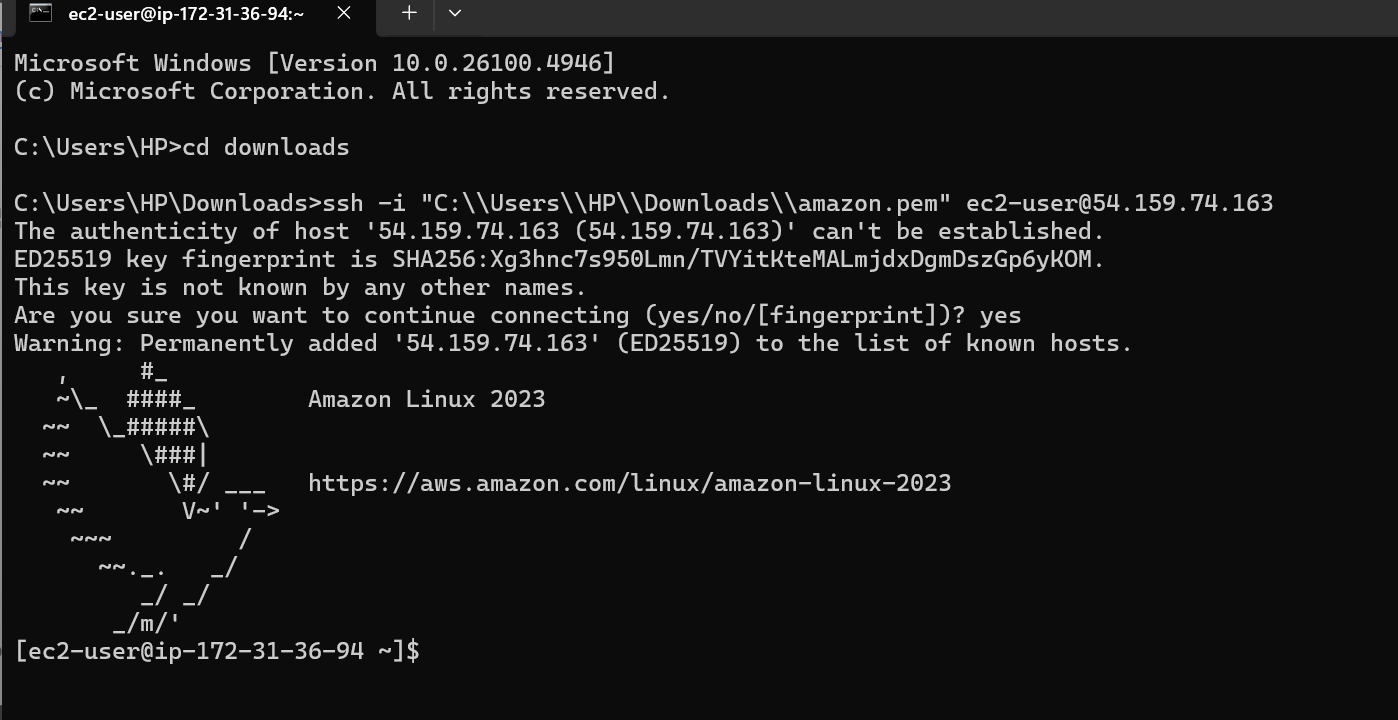
* **Step 3: Connect to EC2 Instance**
* **Using EC2 Instance Connect:**
* **ssh -i key.pem ec2-user@<Public-IP>**
* **Confirm connection by checking system details.**
* ****

Fig-3.1.3

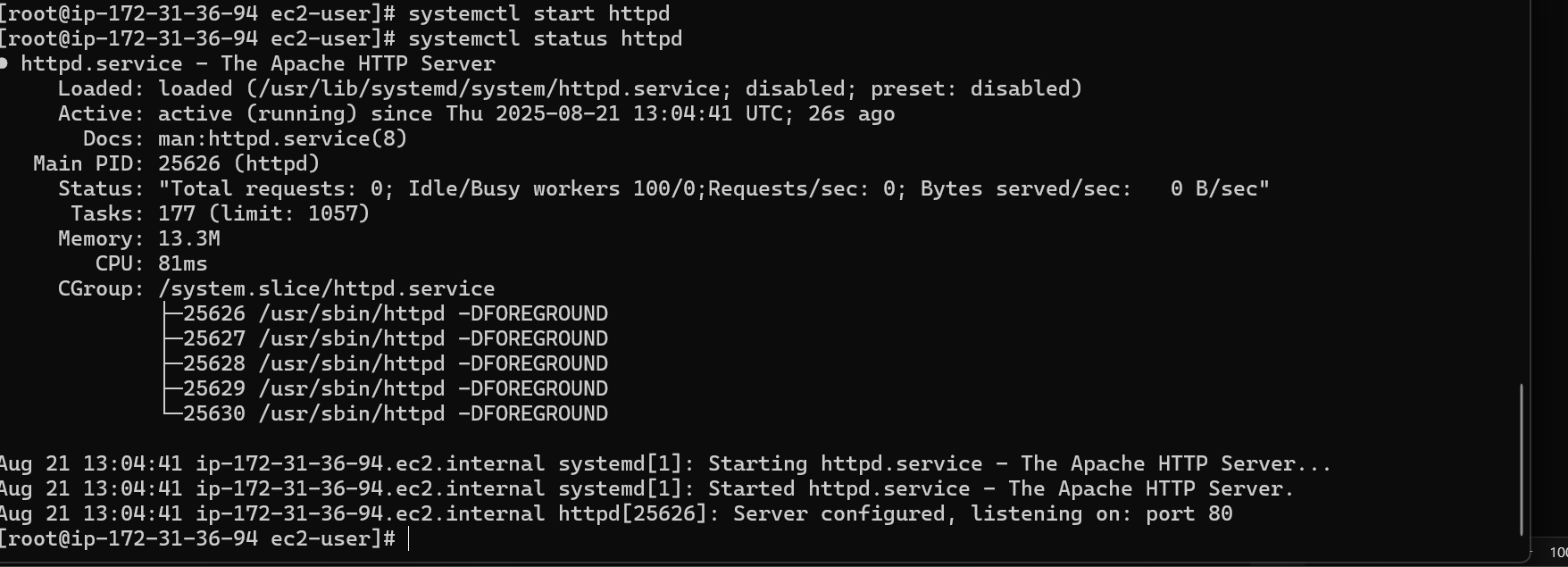
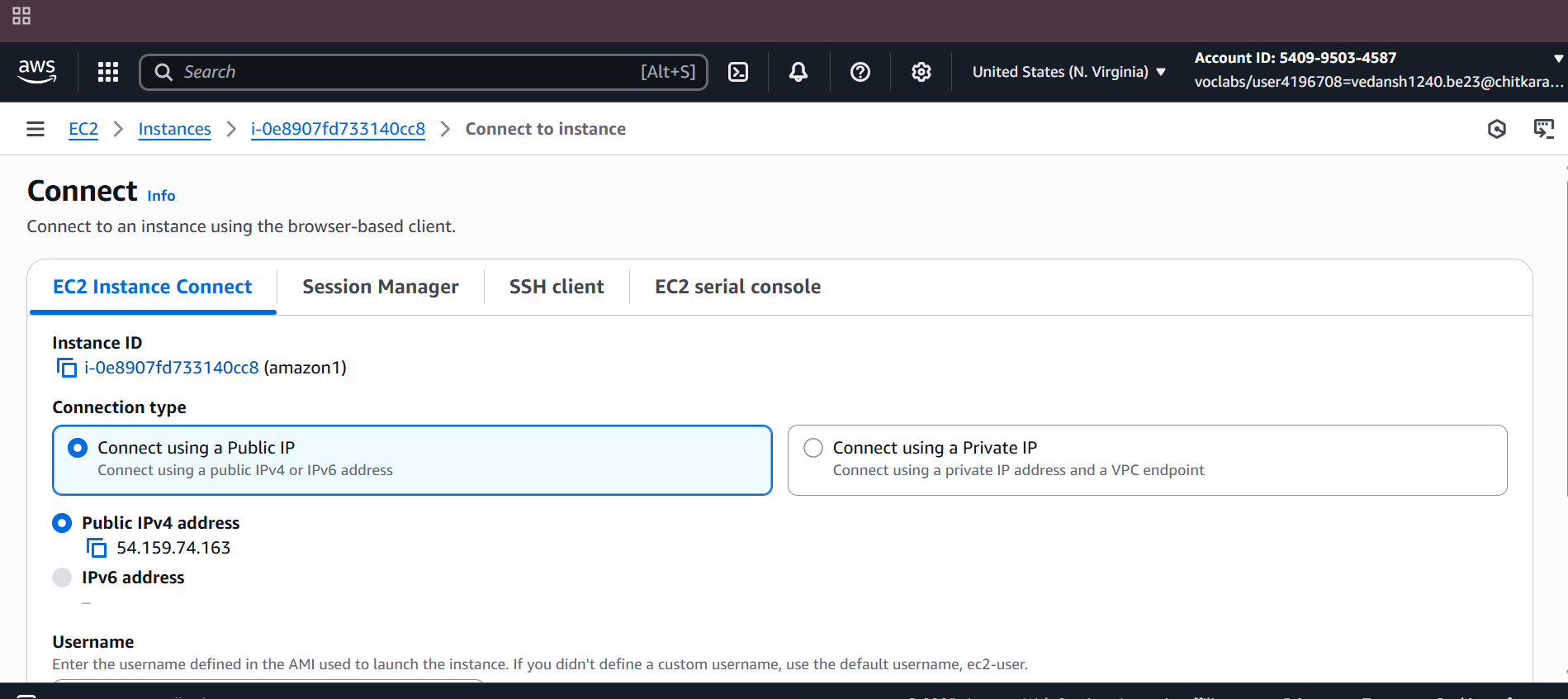
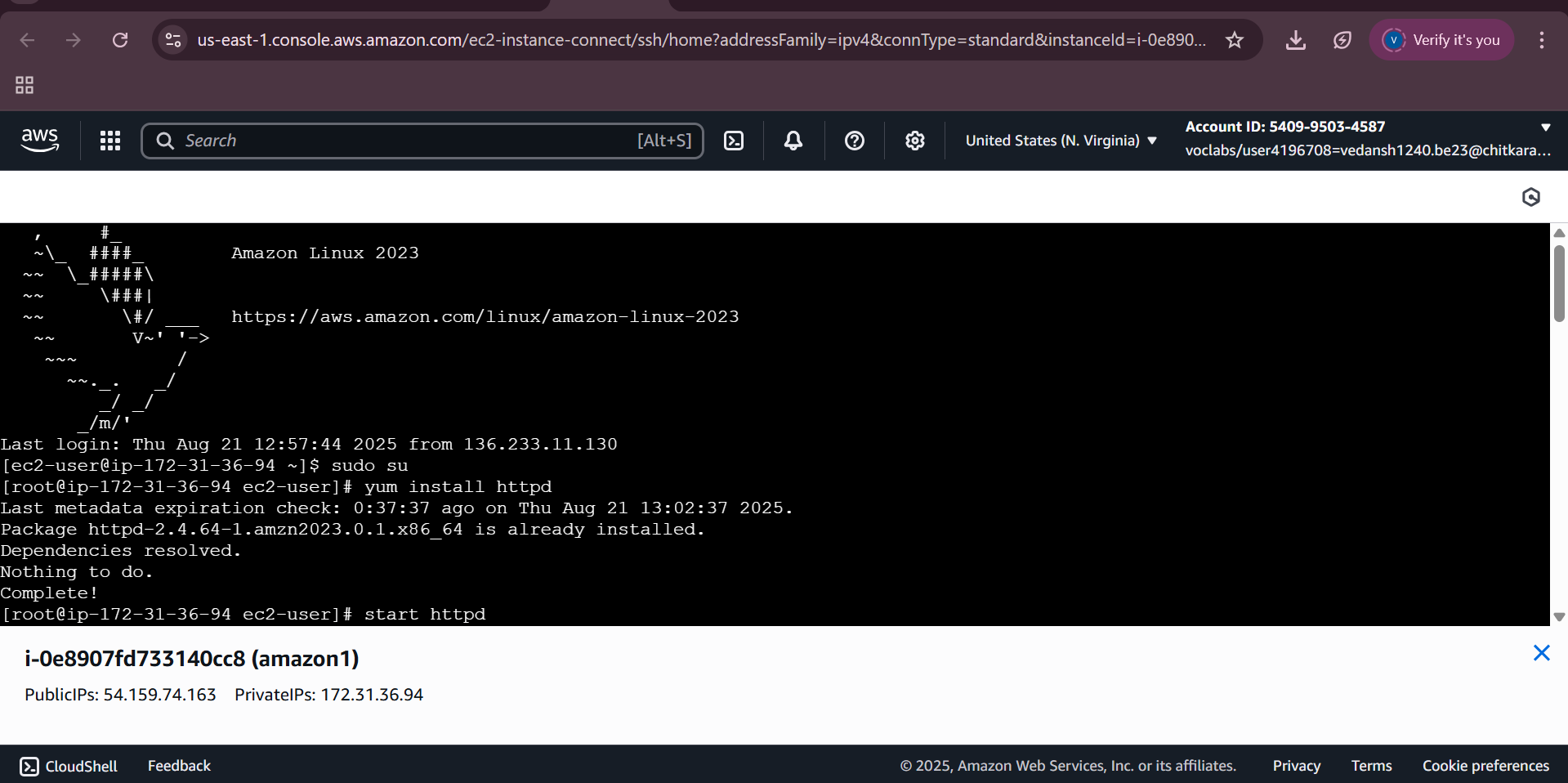
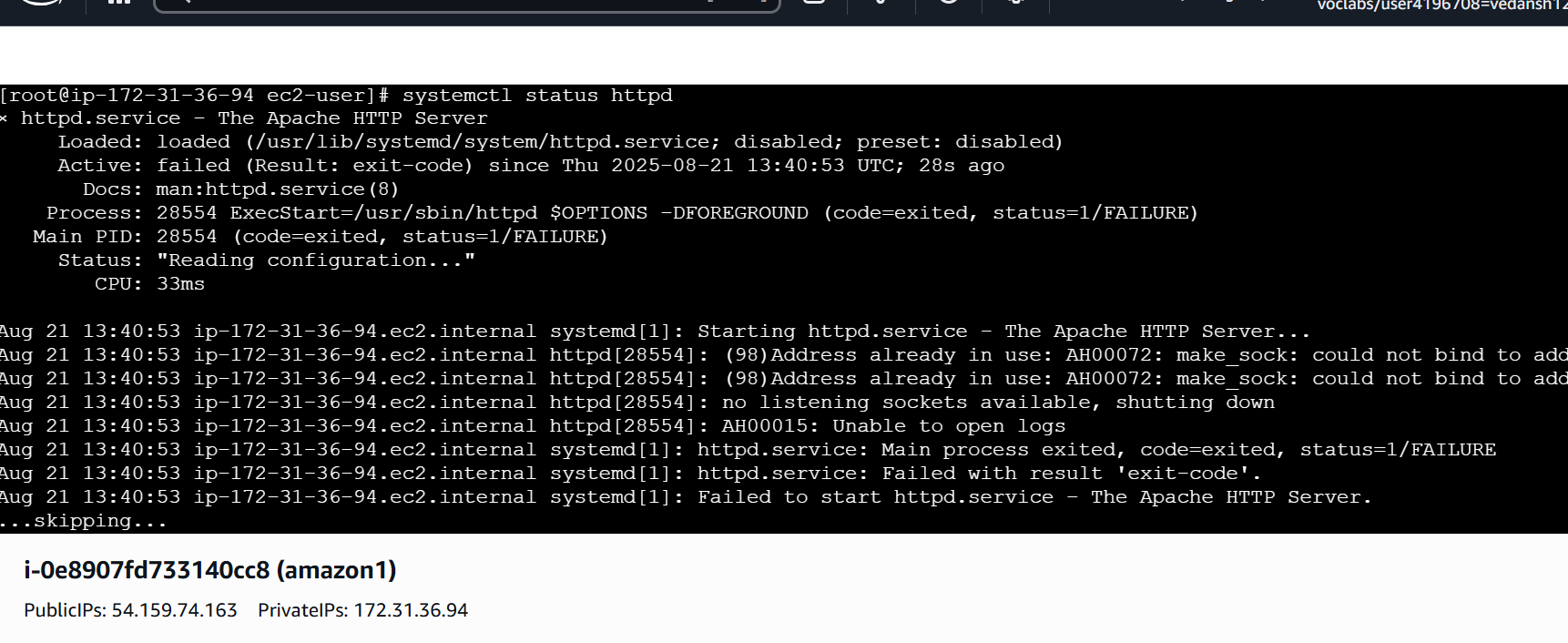
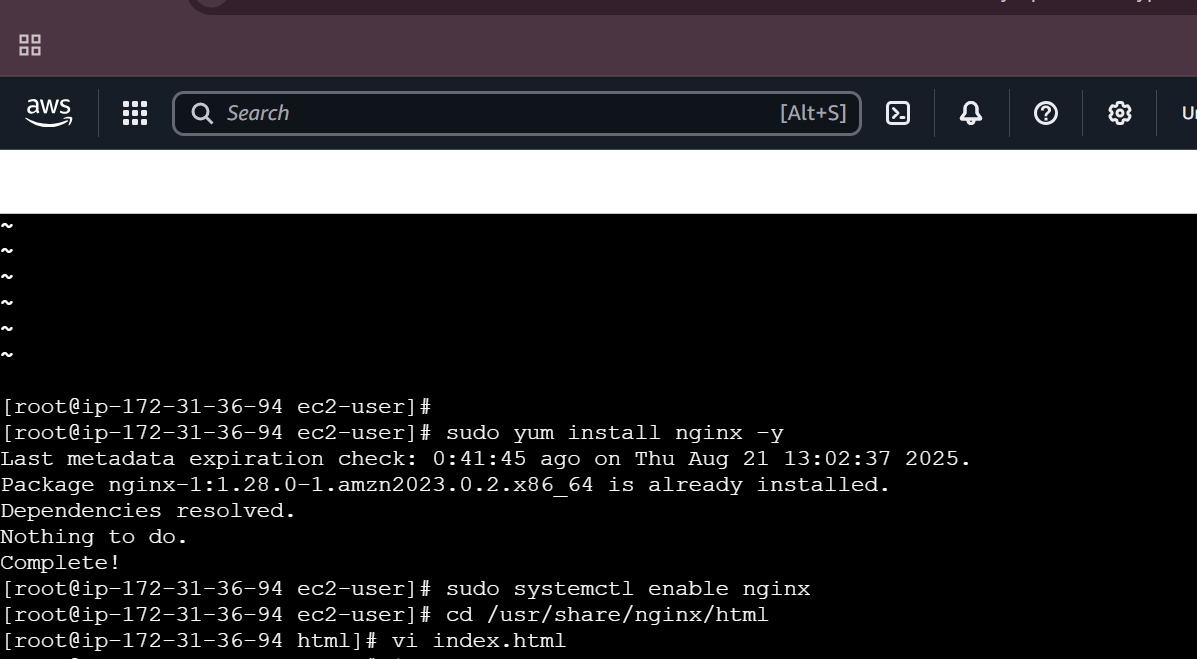
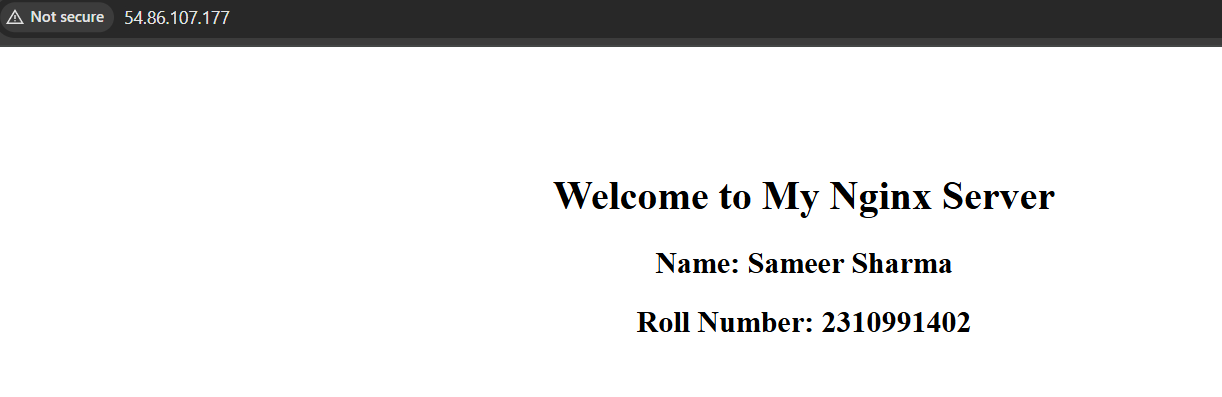
* **Step 4: Install and Configure Apache Web Server  
  For Amazon Linux Apache :**
* **Sudo su**
* **Yum install httpd**
* **Systemctl start httpd**
* **Systemctl status httpd**
* **It works! Is showing on web browser.**
* ****

Fig-3.1.4

* ****

**Fig-3.1.5**

* **Step 5: Install and Configure Nginx Web Server**
* **Sudo su**
* **Yum install httpd**
* **Systemctl Start httpd**
* **Systemctl status httpd**
* **Sudo yum install nginx -y**
* **Sudo systemctl enable nginx**
* **Cd/usr/share/nginx/html**
* **Vi index.html**
* **Now open the public address**
* **Open browser → http://<Public-IP> to verify Nginx.**
* ****
* **Fig-3.1.6**
* ****
* **Fig-3.1.7**
* ****
* **Fig-3.1.8**
* ****
* **Fig-3.1.9**
* ****

**Fig-3.2**

* **Learning Outcomes:**
* **Successfully deployed Apache and Nginx web servers on AWS EC2 instances.**
* **Configured an Auto Scaling Group with Load Balancer to manage varying traffic loads.**
* **Learned how scalability in cloud computing ensures high availability and performance during peak loads.**