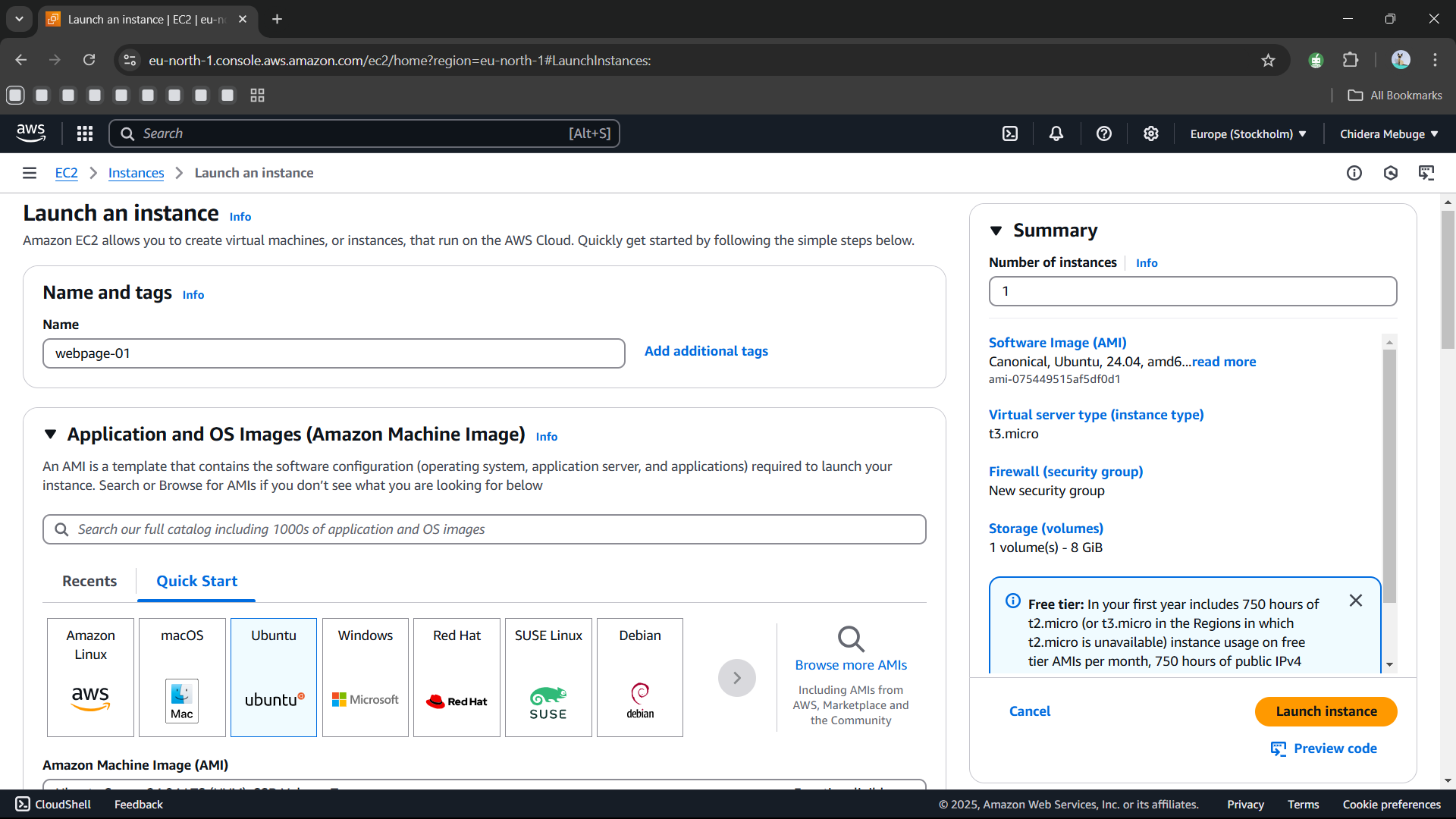
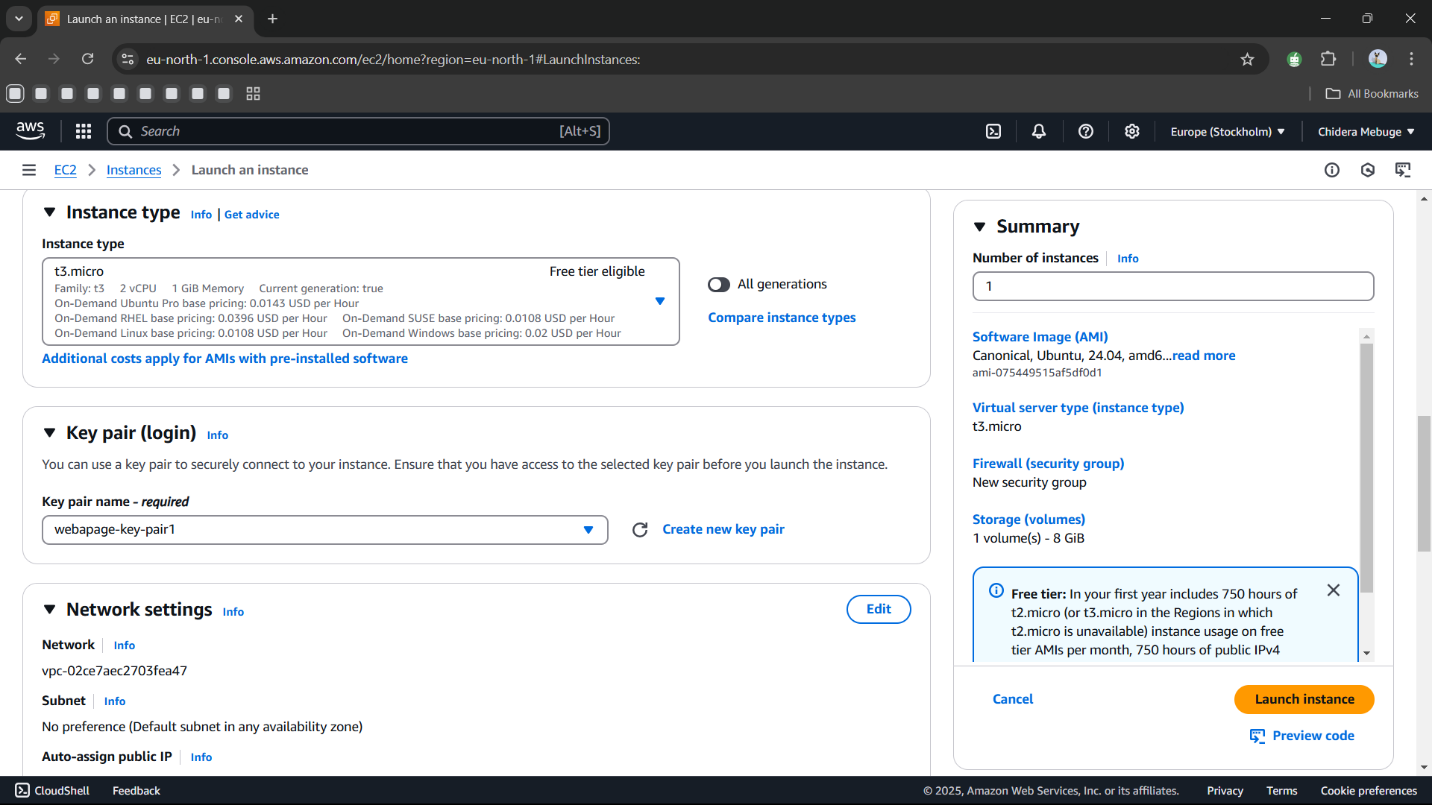
**How to Set Up a Web Server on AWS EC2 Using Ubuntu to Host Web Pages**

**1. Provisioning Your Server**

**Step 1: Launch an EC2 Instance**

1. Sign Up or Log in to your [AWS Management Console](https://aws.amazon.com).
2. Navigate to the **EC2 Dashboard** and click "Launch Instance."



1. Set up your instance:
   * Choose **Ubuntu 22.04 LTS** as your operating system.
   * Select **t2.micro** as the instance type (Free Tier eligible).
   * Set up a key pair to securely connect to your instance.
   * In the **Security Group** settings, make sure you allow the following:
     + **SSH** (port 22) from your IP address.
     + **HTTP** (port 80) from anywhere.
     + **HTTPS** (port 443) from anywhere.
2. A screenshot of a computer

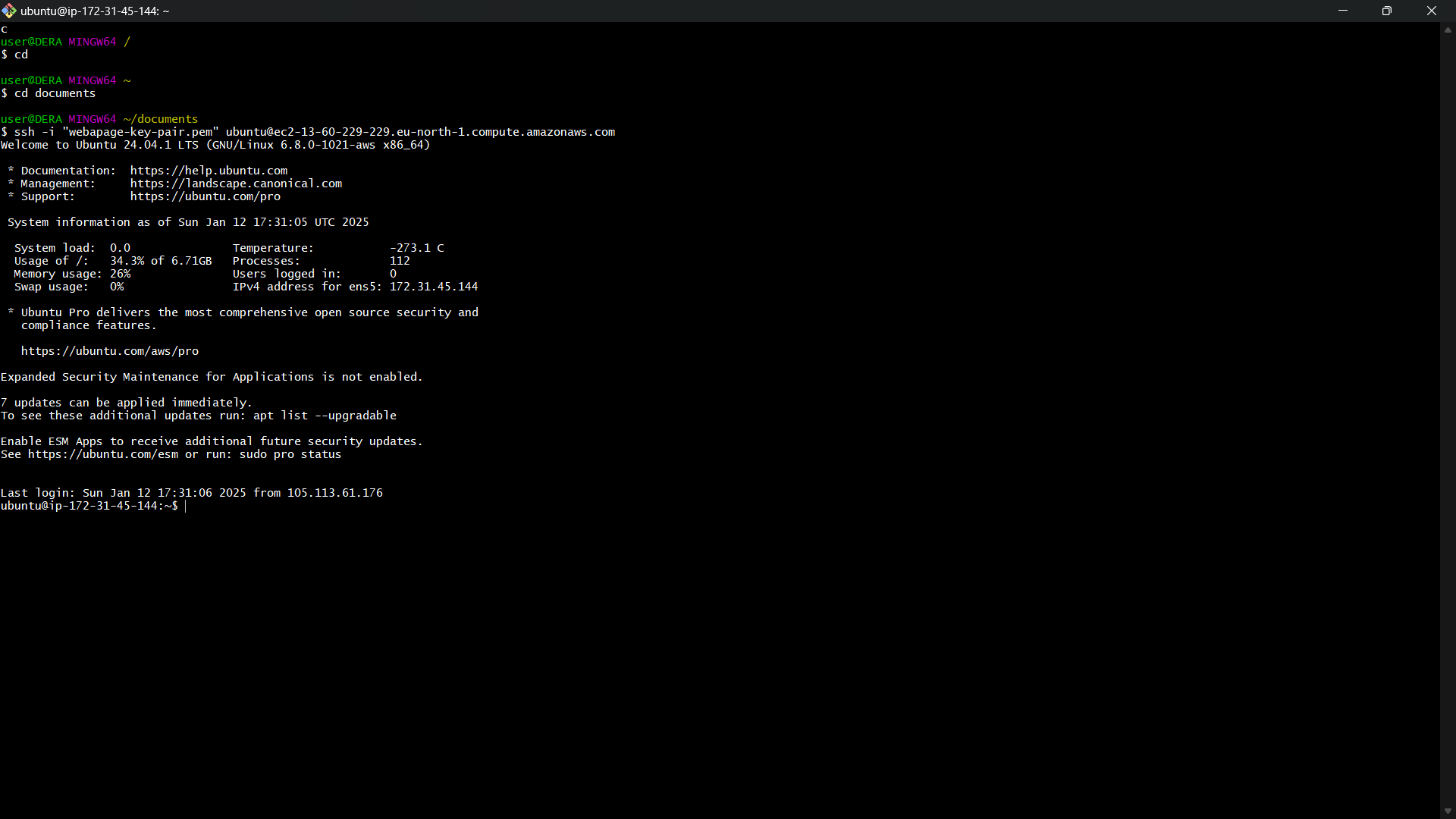
   Description automatically generatedLaunch your instance and note the **Public IPv4 address** for later steps.

**Step 2: Connect to Your Instance**

1. Open a terminal (I personally used GitBash).
2. Run this command to connect via SSH:

**ssh -i your-key.pem ubuntu@<Public-IP>**

Replace your-key.pem with your key file name and <Public-IP> with the public IP address of your instance.



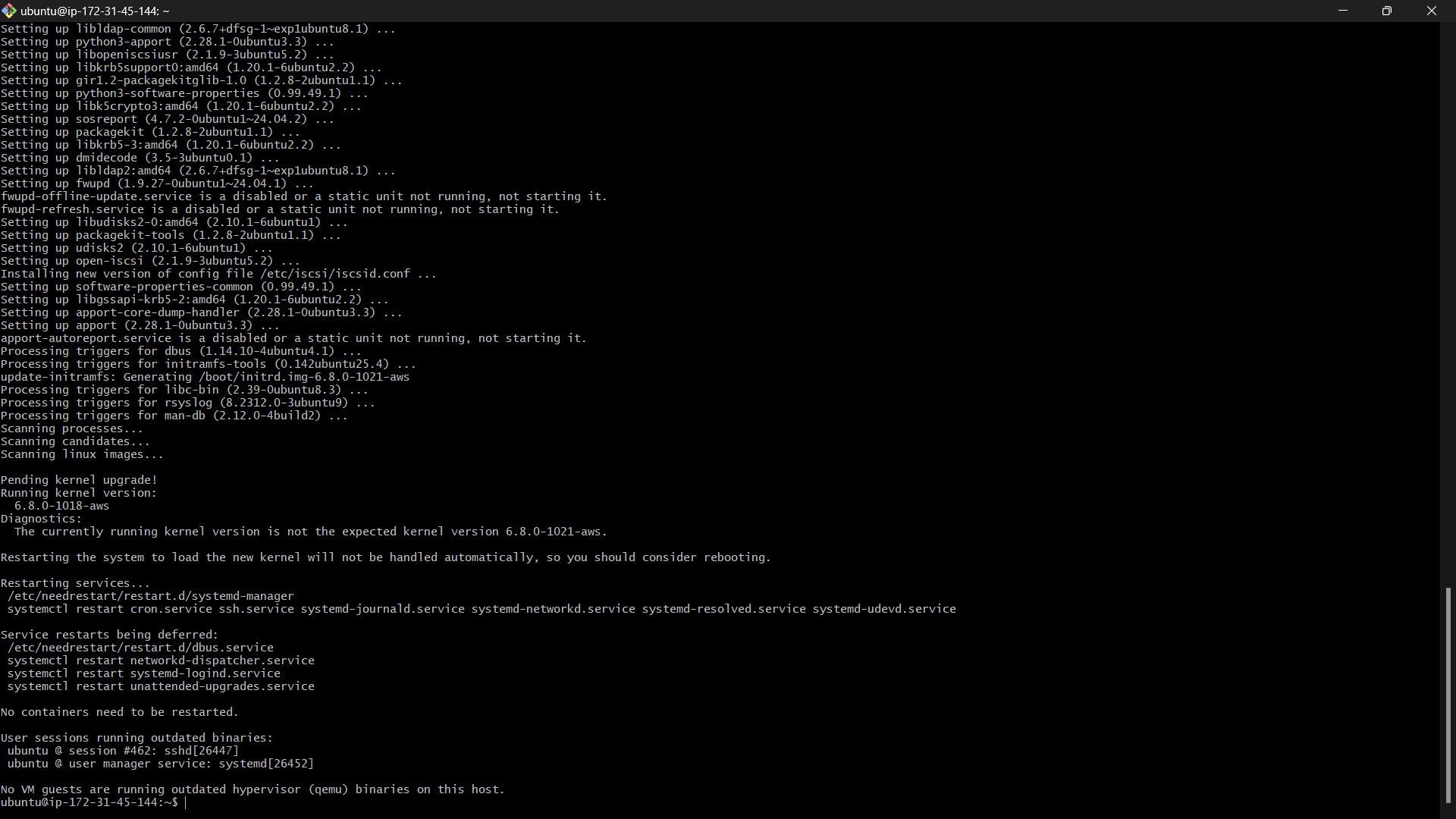
**2. Installing the Web Server**

**Step 1: Update and Install Apache**

1. Update your system packages:

**sudo apt update && sudo apt upgrade -y**

Wait a few minutes while it downloads

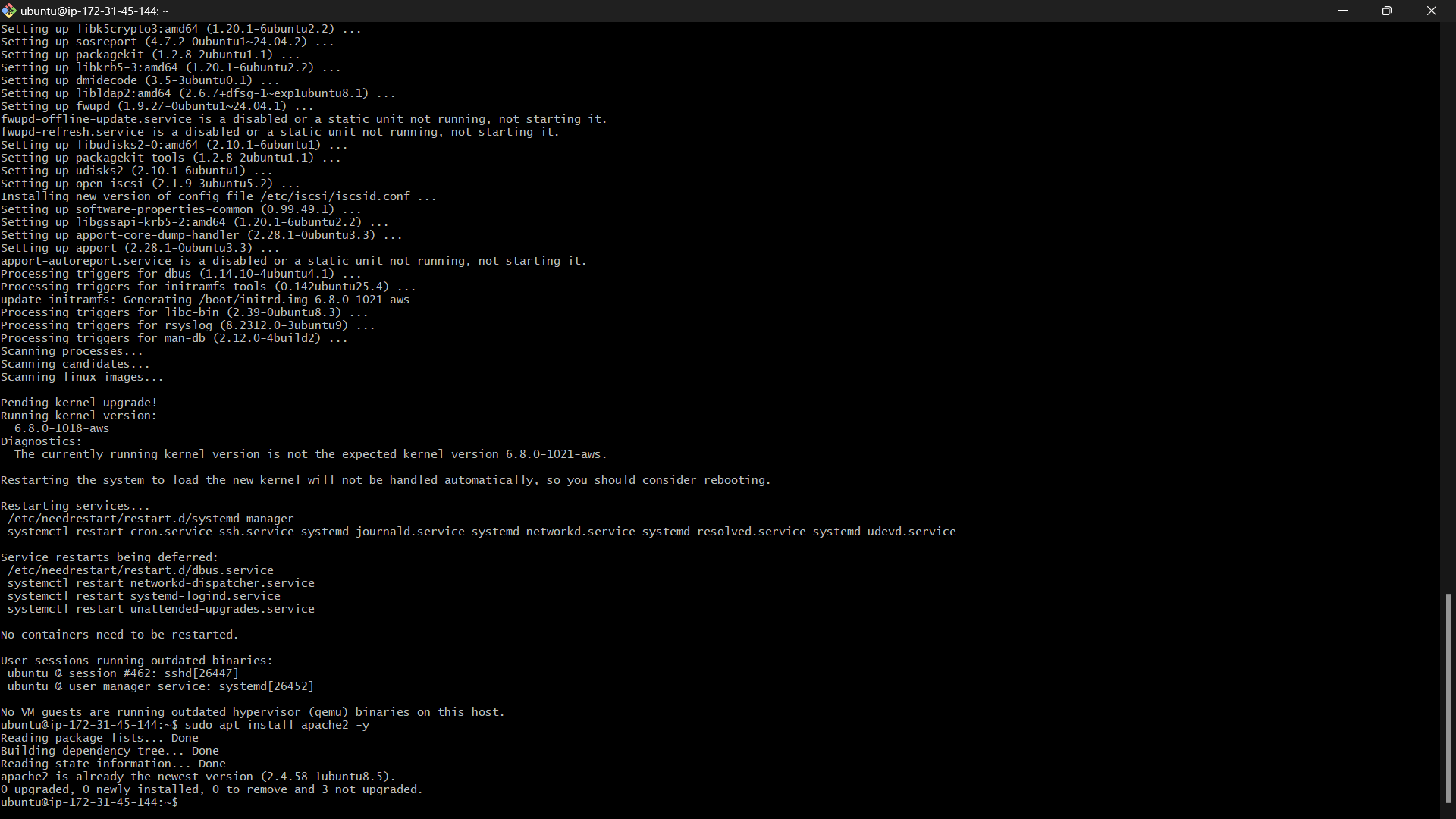


1. Install Apache:

**sudo apt install apache2 -y**

1. Start and enable Apache so it runs on startup:

**sudo systemctl start apache2 && sudo systemctl enable apache2**

****

if Apache is working by visiting http://<Public-IP> in your browser. You should see the Apache default page.

**3. Deploying Your HTML Page**

**Step 1: Set Up the Web Directory**

1. Navigate to the directory for your HTML page:

cd /var/www/html

**Step 2: Create the HTML File**

1. Navigate to the new directory:

cd /var/www/assignment

1. Create an HTML file:

nano index.html

1. Add this content to your file:

<!DOCTYPE html>

<html>

<head>

<title>Welcome to Altschool Assignment</title>

</head>

<body>

<h1>Welcome to Altschool Landing Page</h1>

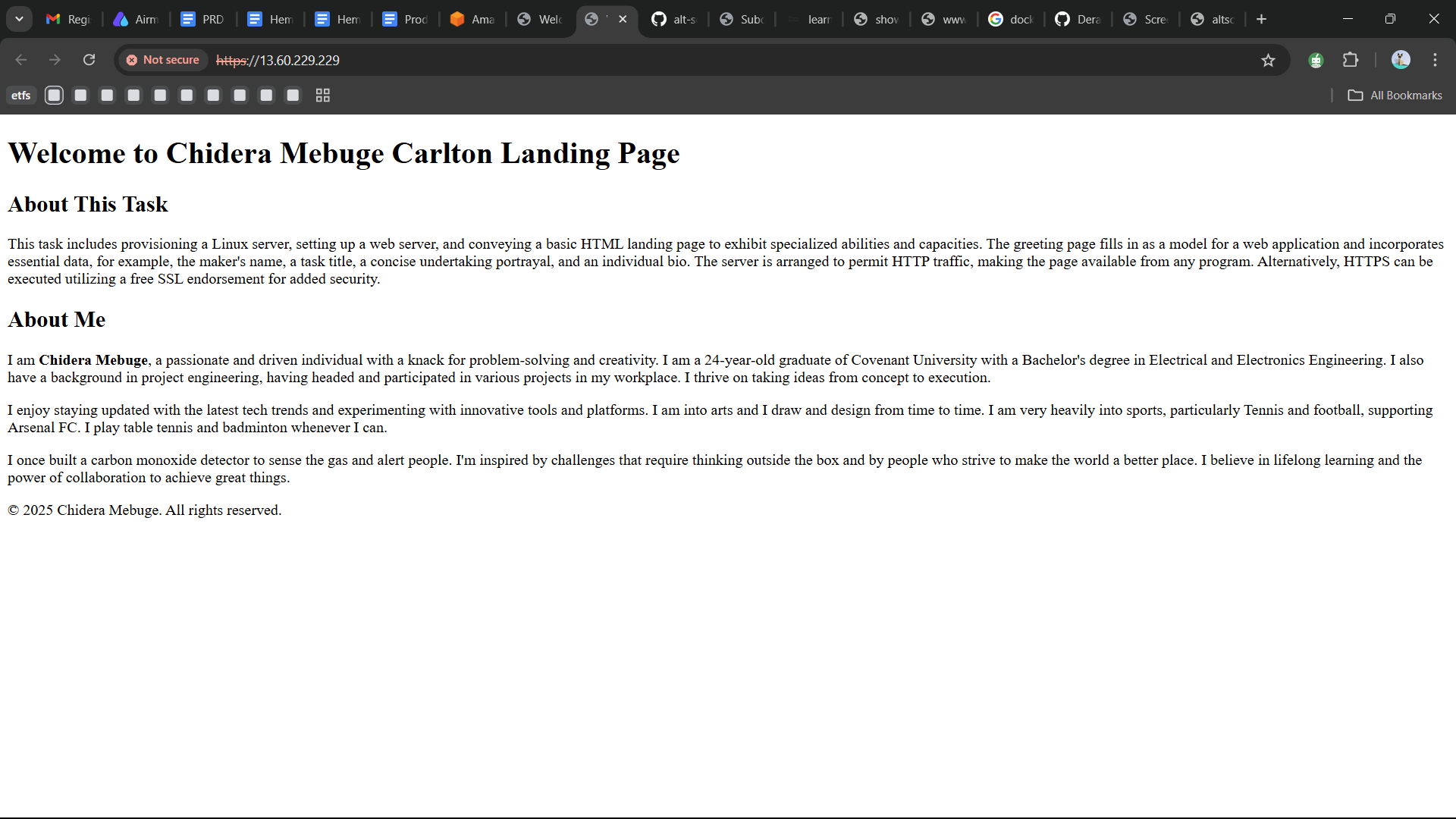
<p>My name is [Your Name].</p>

<p>This is my project for the Altschool Cloud Engineering course.</p>

</body>

</html>

1. Save and exit the file (Ctrl+O, Enter, Ctrl+X).
2. Open the Ip Adress and check for the new webpage



**4. Configuring Networking**

**Step 1: Set Up Security Group Rules**

1. Go to the **EC2 Dashboard** in AWS.
2. Make sure your instance’s security group allows traffic on:
   * Port 22 (SSH).
   * Port 80 (HTTP).
   * Port 443 (HTTPS, if needed).

**Step 2: (Optional) Use a Custom Domain**

1. If you have a domain, update its DNS records:
   * Add an **A Record** pointing the domain to your EC2 public IP.
   * Optionally, add a **CNAME Record** for www.
2. Wait for DNS propagation (this may take a few hours).
3. Test by visiting your domain in a browser.

**5. Bonus: Setting Up HTTPS**

Creating a Domain and Securing Your Web Page with HTTPS

**Step 1: Create a Domain Name**

Start by creating a domain for your webpage. You can use a free DNS provider like <https://freedns.afraid.org/subdomain/> to set up your domain. Link the domain to your instance’s public IP address.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 2: Install Certbot for SSL Configuration**

Securing your website with HTTPS is essential for encrypting data and building trust with your users. Let’s Encrypt offers free SSL certificates, and Certbot simplifies the installation process.

Run this command:

**sudo apt install certbot python3-certbot-apache -y**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 3: Obtain an SSL Certificate

To get an SSL certificate and configure HTTPS for your domain, use the following command:

**sudo certbot --apache -d <your domain name>**

Provide Contact Information: You’ll be prompted to:

Enter an email address for critical notifications (e.g., renewal reminders).

Agree to Let’s Encrypt’s terms of service.

Domain Verification: Certbot verifies that you own the domain by checking the DNS record.

SSL Certificate Installation: After verification, Certbot fetches the SSL certificate and automatically configures Apache to use HTTPS.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 4: Verify HTTPS**

Once the SSL certificate is successfully installed, open your browser and navigate to your domain

What to Look For:

A secure padlock icon in the address bar, indicating that the connection is encrypted.

Your webpage should load without any warnings, confirming that HTTPS is properly set up.