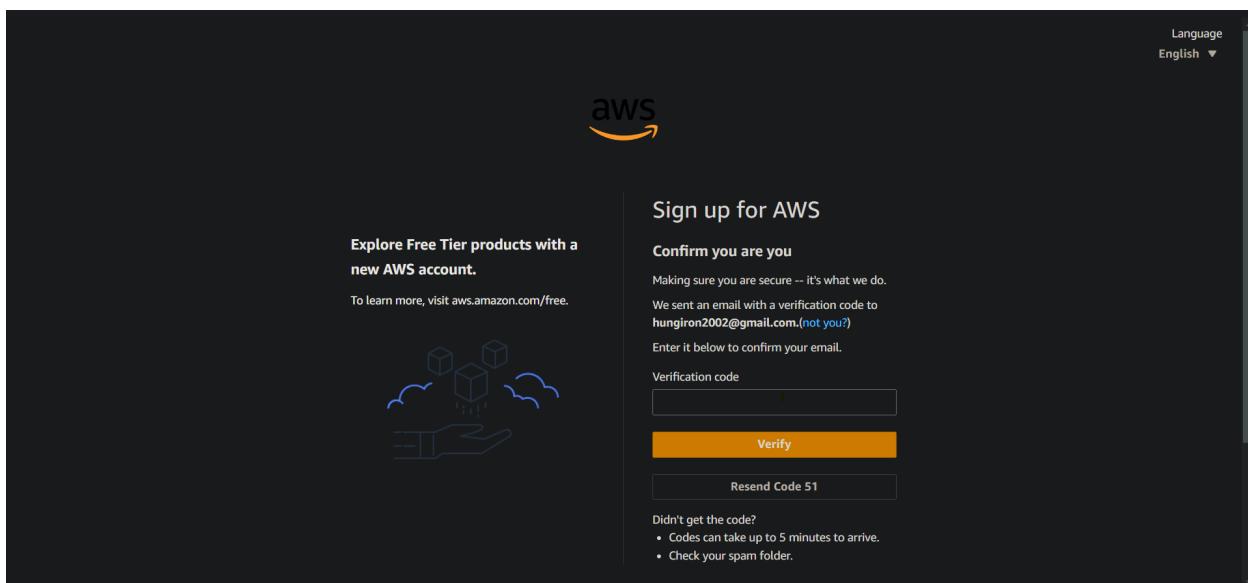


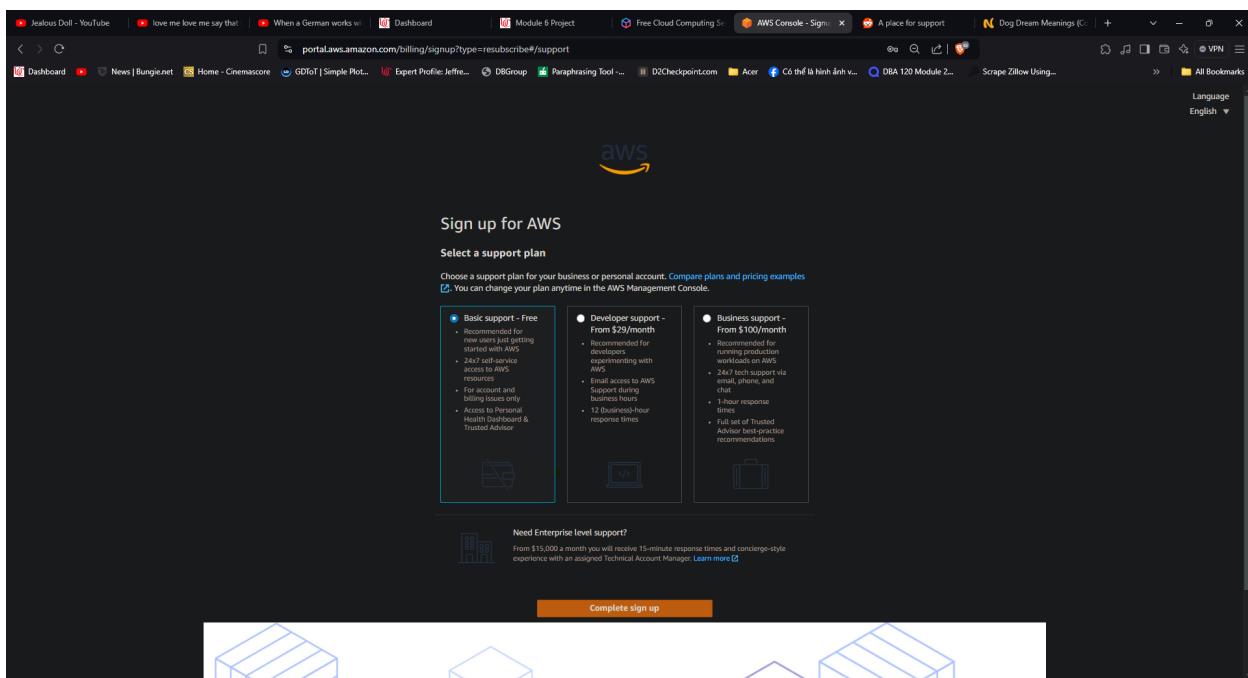
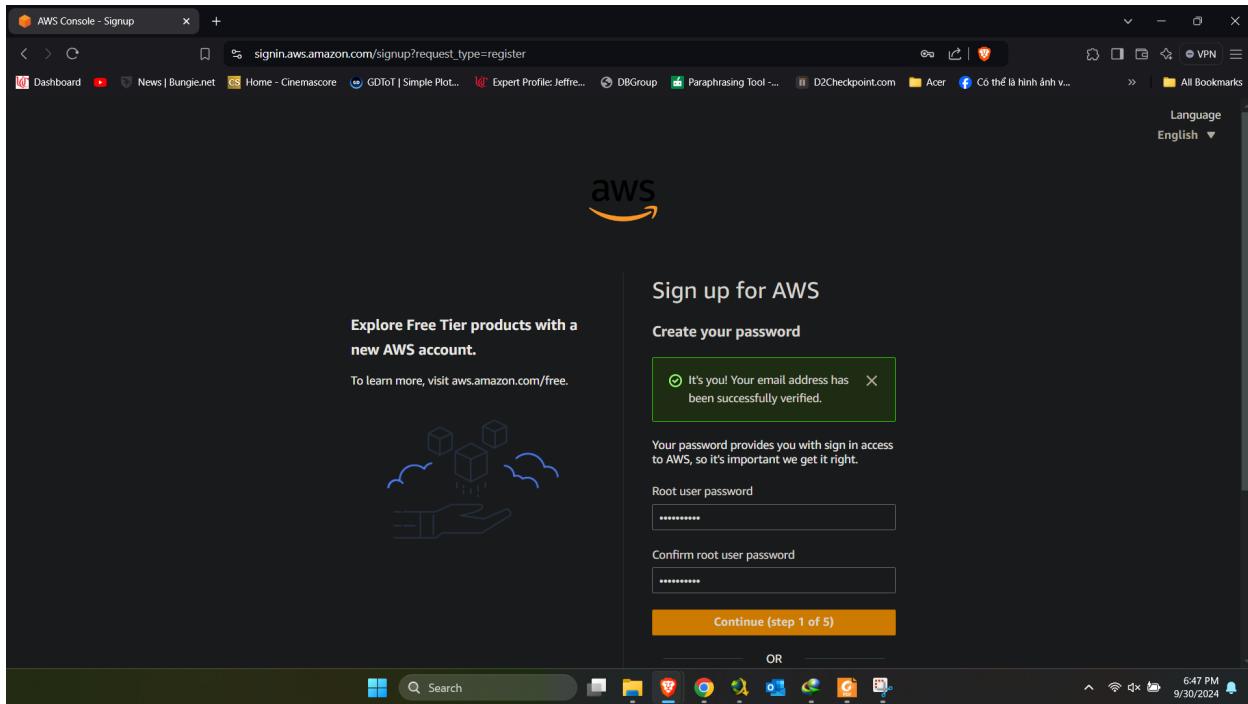
Cloud Computing Project 2

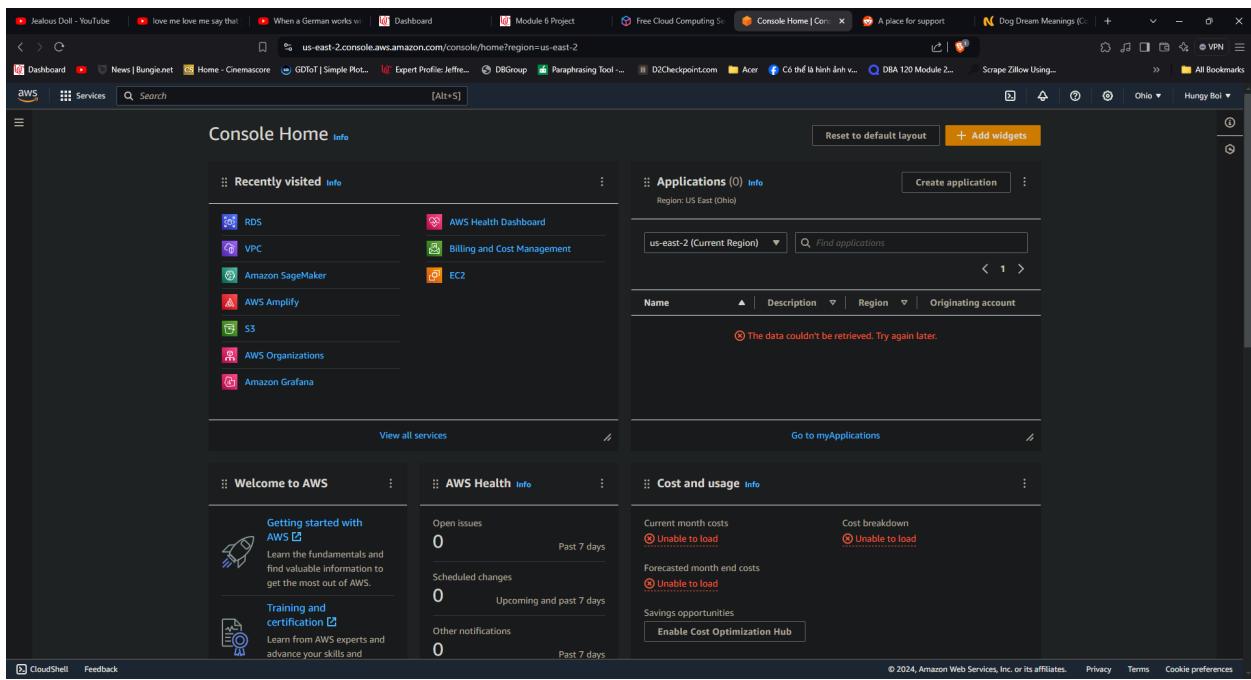
⌚ Created	@October 5, 2024 12:14 PM
👤 Created by	(H) Hungy Boi

Webpage: <http://ec2-3-133-58-209.us-east-2.compute.amazonaws.com/>

1. Launch an **EC2 instance** (any platform accepted but should be accessible through the internet).
 - To create my AWS Free Tier account, I first visit the AWS website and click "Create an AWS Account." I then sign up by entering my email, choosing a strong password, and providing my contact information, including a valid phone number for verification. Once I'm signed up, I'm automatically enrolled in the Free Tier, which gives me 12 months of access to services like EC2, S3, RDS, and Lambda. I'll need to provide a credit card for identity verification.







2. EC2 creation

To launch an EC2 instance on AWS, I first log in to the AWS Management Console and choose Ubuntu as my Amazon Machine Image (AMI). Then, I select the t2.micro instance type, which is free under the Free Tier. Next, I configure the Network Security Group by setting rules to allow SSH (port 22) and HTTP traffic (port 80) from the internet. I also create or select a key pair for login, using a .ppk file for my Windows machine. Finally, I launch the instance

- EC2 Console:

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, a tooltip for the 'Free tier' is displayed, stating: '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'. The 'Launch instance' button is highlighted.

- Amazon Machine Image (AMI) :

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Amazon Machine Image (AMI)' step, a tooltip for the 'Free tier' is displayed, stating: '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'. The 'Launch instance' button is highlighted.

- Key pair (login):

The screenshot shows the 'Launch instances' step of the AWS EC2 wizard. On the left, the 'Instance type' section is open, showing a dropdown menu for 't2.micro' with details like 'Family: t2', '1 vCPU', '1 GiB Memory', and 'Current generation: true'. It also lists base pricing for various operating systems. Below this, there's a note about additional costs for AMIs with pre-installed software. On the right, the 'Summary' section shows 'Number of instances' set to 1, and the 'Software Image (AMI)' selected is 'Canonical, Ubuntu, 24.04, amd64'. Other summary items include 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. At the bottom right are 'Cancel', 'Launch instance', and 'Review commands' buttons.

This screenshot shows the 'Create key pair' dialog box overlaid on the main launch wizard. In the 'Key pair name' field, 'Key4Wir' is entered. Under 'Key pair type', 'RSA' is selected. In the 'Private key file format' section, '.ppk' is selected. A warning message at the bottom states: '⚠ 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](#)'.

• Network settings

The screenshot shows the AWS EC2 'Launch Instances' wizard. On the left, under 'Network settings', there are sections for Network (vpc-0b2da6d41665cdab), Subnet (No preference), Auto-assign public IP (Enabled), and Firewall (security groups). A note about additional charges applies outside the free tier allowance. Under 'Create security group', the 'Create new security group' radio button is selected. Below it, three rules are listed: Allow SSH traffic from Anywhere (0.0.0.0/0), Allow HTTPS traffic from the internet, and Allow HTTP traffic from the internet. A warning message at the bottom states: '⚠ 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.' On the right, the 'Summary' section shows 1 instance, the Canonical Ubuntu 24.04 AMI, and a t2.micro virtual server type. A tooltip for 'Free tier' is displayed, stating: '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'. At the bottom right are 'Cancel', 'Launch instance', and 'Review commands' buttons.

The screenshot shows the AWS EC2 'Launch an instance' success page. A green banner at the top indicates 'Success: Successfully initiated launch of instance (i-0831aa32eb64e8870)'. Below this, the 'Next Steps' section contains four cards:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a 'Create billing alerts' button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a 'Connect to instance' button and a 'Learn more' link.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a 'Connect an RDS database' button and a 'Create a new RDS database' button.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a 'Create EBS snapshot policy' button.

At the bottom right are 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

2. Configure web server and SQLite3 Database table

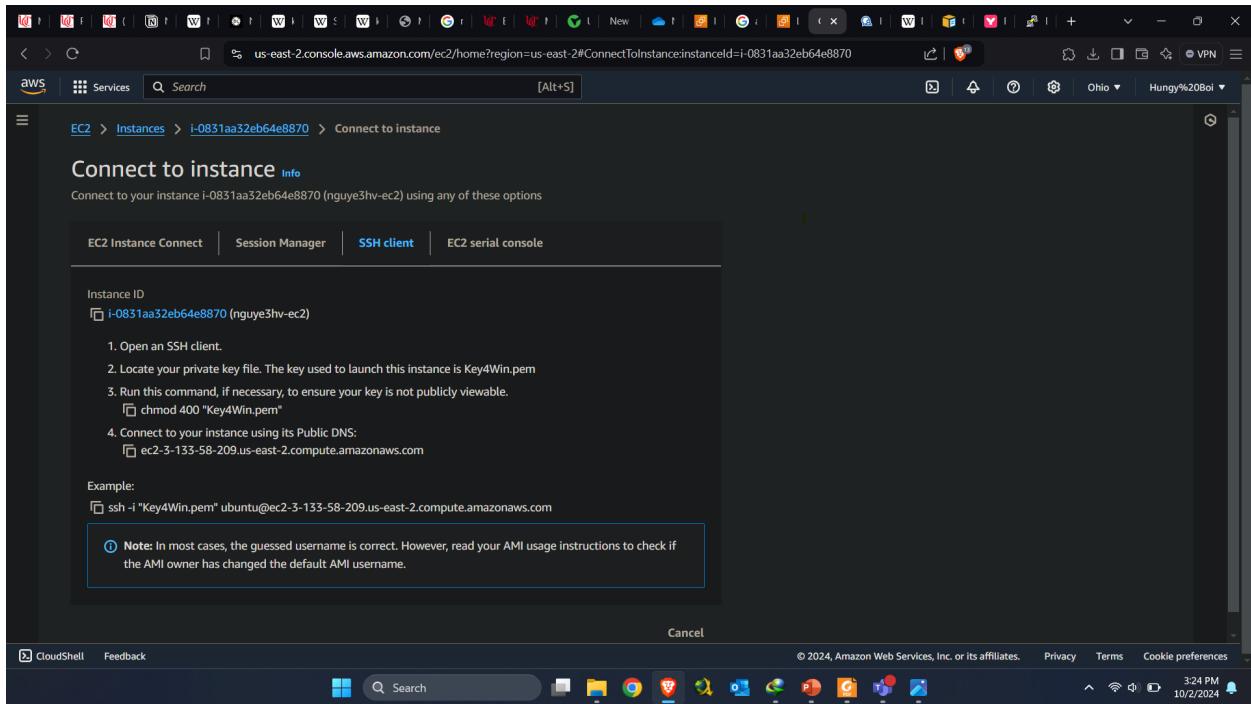
Once my instance is running, I click on the instance ID, go to Connect, and follow the instructions to SSH into my instance.

The screenshot shows the AWS EC2 Dashboard for the US East (Ohio) Region. On the left, a sidebar lists various EC2-related services like Instances, Images, and Network & Security. The main area displays a summary of resources: 0 running instances, 0 auto scaling groups, 0 capacity reservations, 0 dedicated hosts, 0 elastic IPs, 0 instances, 0 key pairs, 0 load balancers, 0 placement groups, 1 security group, 0 snapshots, and 0 volumes. Below this, there's a "Launch instance" section with a "Launch instance" button and a "Migrate a server" option. A note says instances will launch in the US East (Ohio) Region. To the right, there's a "Service health" section showing "AWS Health Dashboard" and a status message: "This service is operating normally." Further right is an "EC2 Free Tier" section with information about free tier offers and account attributes like the default VPC (vpc-0b2da6d41665cdab7). The bottom right corner includes copyright and legal links.

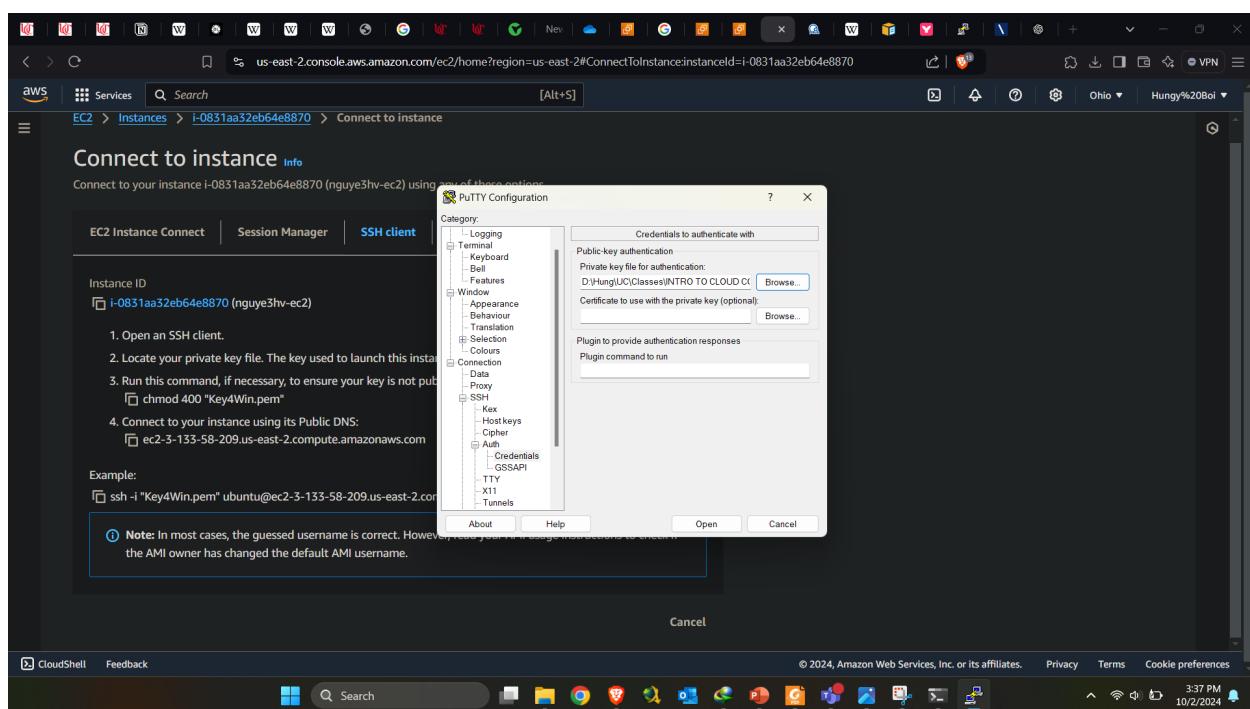
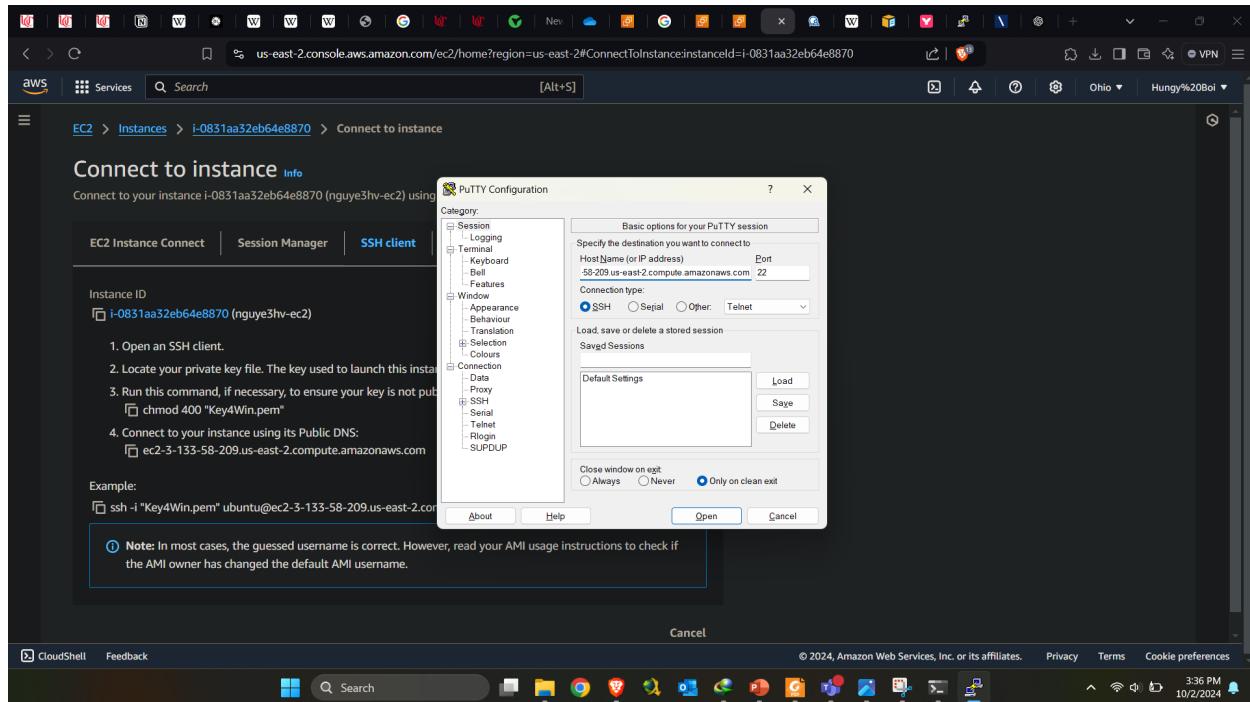
This screenshot shows the detailed view for a specific EC2 instance, i-0831aa32eb64e8870. The top navigation bar shows the instance ID. The main content area is titled "Instance summary for i-0831aa32eb64e8870 (nguye3hv-ec2)". It provides a table of instance details:

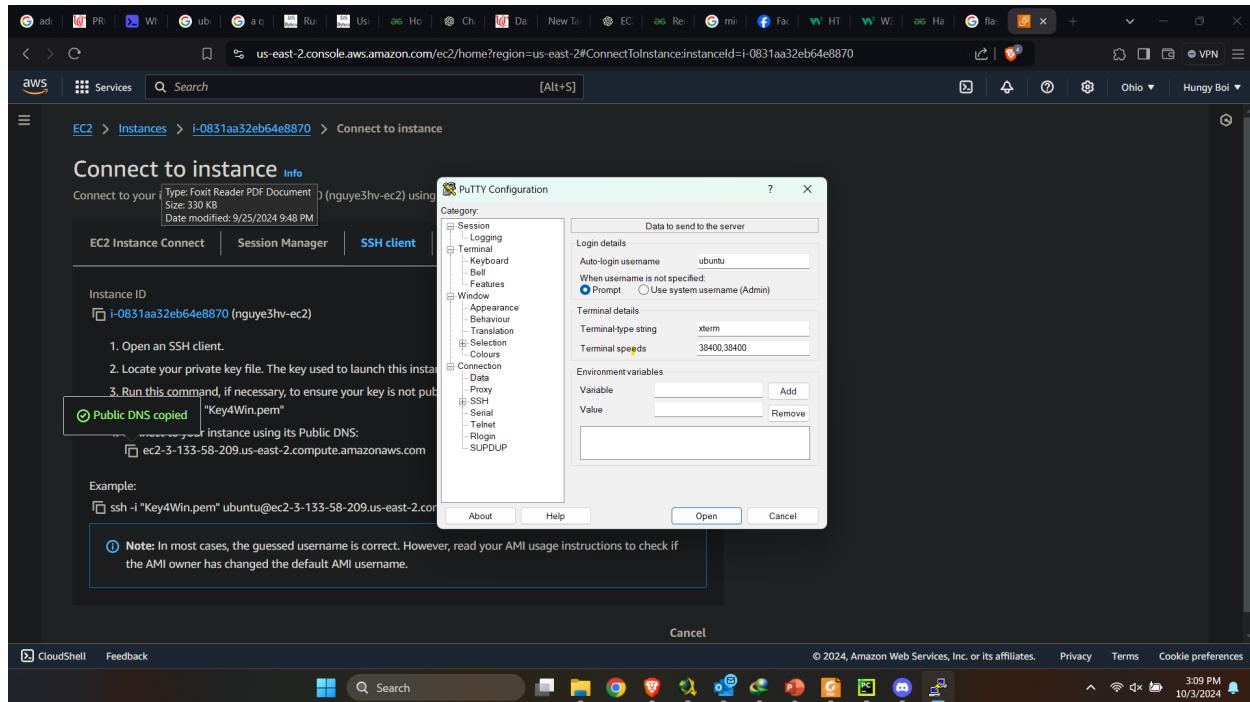
Attribute	Value
Instance ID	i-0831aa32eb64e8870 (nguye3hv-ec2)
Public IPv4 address	3.133.58.209 open address
IPv6 address	-
Instance state	Running
Private IP DNS name (IPv4 only)	ip-172-31-35-115.us-east-2.compute.internal
Instance type	t2.micro
VPC ID	vpc-0b2da6d41665cdab7
Subnet ID	subnet-0c989de601ec0331e
Instance ARN	arn:aws:ec2:us-east-2:615299755888:instance/i-0831aa32eb64e8870
IAM Role	-
Auto-assigned IP address	3.133.58.209 [Public IP]
IMDSv2 Required	Required

On the right side, there are additional sections: "Private IPv4 addresses" (172.31.35.115), "Public IPv4 DNS" (ec2-3-133-58-209.us-east-2.compute.amazonaws.com), "Elastic IP addresses" (-), "AWS Compute Optimizer finding" (Opt-in to AWS Compute Optimizer for recommendations), and "Auto Scaling Group name" (-). The bottom right corner includes copyright and legal links.



To connect to the EC2 instance using PuTTY, I set the hostname as my instance's DNS. Then I go to SSH, Auth, then Credentials then I select my download .ppk file as the key. Finally, I click on the Data button on the side tab and put in my auto-login username as *ubuntu*.





```

$ ssh -i "Key4Win.pem" ubuntu@ec2-3-133-58-209.us-east-2.compute.amazonaws.com
Last login: Thu Oct  3 03:17:29 UTC 2024
[ubuntu@ip-172-31-35-115 ~]
$ Using username "ubuntu".
$ Authenticating with public key "Key4Win"
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.6.0-1016-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/parts

System information as of Thu Oct  3 03:17:29 UTC 2024

System load: 0.0      Processes:          104
Usage of /: 23.0% of 6.71GB  Users logged in:   0
Memory usage: 20%
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

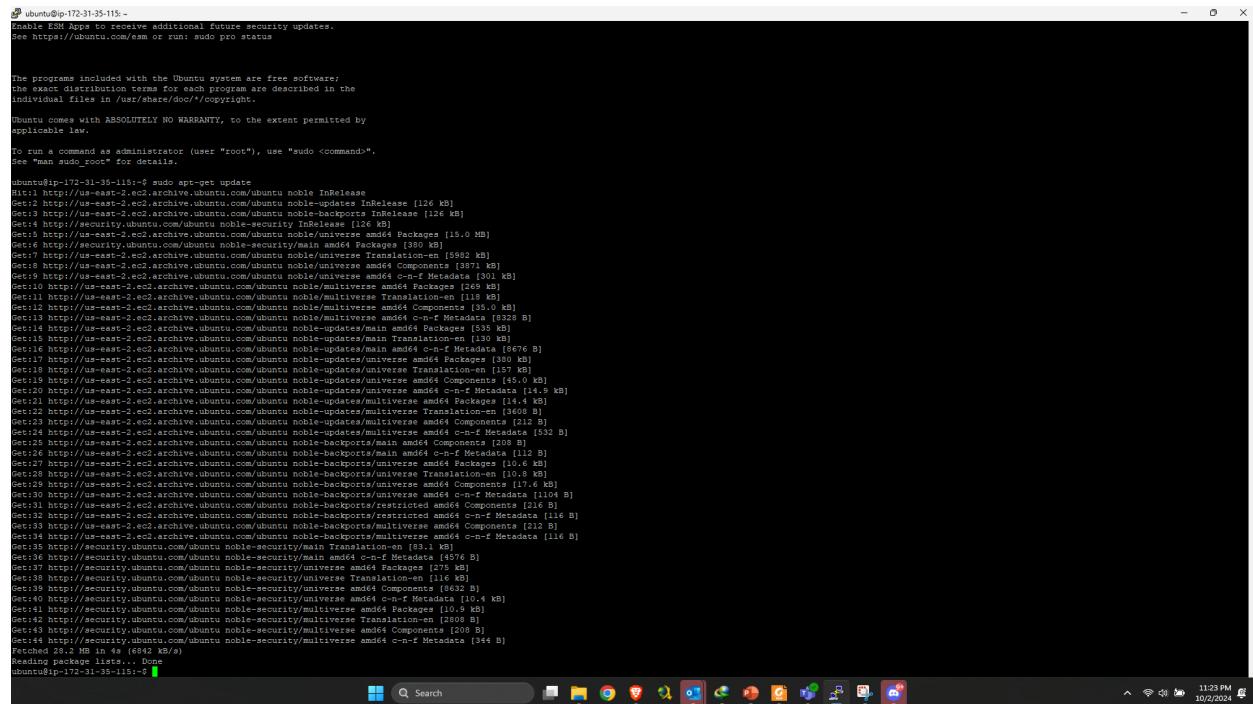
ubuntu@ip-172-31-35-115:~$ 

```

To install the Apache web server and mod_wsgi on my EC2 instance, I updated the package list with `sudo apt-get update`. Next, I install Apache using `sudo apt-get install apache2` and mod_wsgi with `sudo apt install libapache2-mod-wsgi-py3`. I then install Flask

using the pip tool by running `sudo apt install python3-pip` followed by `sudo apt install python3-flask`. I also change the permissions of the `/home/ubuntu/` directory using `chmod 755 /home/ubuntu/`. Then, I install the ACL package by typing `sudo apt install acl`. Finally, to test the configuration, I check the Apache error log with `tail /var/log/apache2/error.log`.

- Update all packages:



```
ubuntu@ip-172-31-35-115:~$ Enable ESM Apps to receive additional future security updates.
see https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-35-115:~$ sudo apt-get update
Hit:1 http://us-east-2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.archive.ubuntu.com/ubuntu noble/universe Translation-en Packages [360 kB]
Get:6 http://us-east-2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [360 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3971 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Metadata [265 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [265 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [39.0 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Metadata [1028 B]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [6676 B]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [157 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [157 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Metadata [14.9 kB]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3600 B]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Metadata [112 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.6 kB]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [17.4 kB]
Get:30 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 kB]
Get:31 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:32 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Metadata [112 B]
Get:33 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [212 B]
Get:34 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [116 B]
Get:35 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.1 kB]
Get:36 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [176 B]
Get:37 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [275 kB]
Get:38 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 kB]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Metadata [10.4 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2008 B]
Get:43 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [200 B]
Get:44 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 28.2 MB in 4s (6942 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-115:~$
```

- Install Apache 2 package

```
lubuntu@lubuntu-173-31-35-115: ~
$ apt-get update
$ apt-get upgrade
$ apt-get install apache2
$ curl http://173.31.35.115/

```

- Install mod_wsgi package

```
ubuntu@ip-172-31-35-115: ~
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libapache2-mod-wsgi-py3
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
ubuntu@ip-172-31-35-115: ~$ sudo apt-get install libapache2-mod-wsgi
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package libapache2-mod-wsgi is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'libapache2-mod-wsgi' has no installation candidate
ubuntu@ip-172-31-35-115: ~$ apt-cache search libapache2-mod-wsgi
libapache2-mod-wsgi - a Python WSGI module for Apache
ubuntu@ip-172-31-35-115: ~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
ubuntu@ip-172-31-35-115: ~$ sudo apt-get install libapache2-mod-wsgi
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package libapache2-mod-wsgi is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'libapache2-mod-wsgi' has no installation candidate
ubuntu@ip-172-31-35-115: ~$ sudo apt-get install libapache2-mod-wsgi-py3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-wsgi-py3 is already the newest version.
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 103 kB in 0 s (6019 kB/s).
Fetched 103 kB in 0 s (6019 kB/s)
Selecting previously unselected package libapache2-mod-wsgi-py3.
Preparing to unpack .../libapache2-mod-wsgi-py3_5.0.0-1build2_amd64.deb ...
Unpacking libapache2-mod-wsgi-py3 (5.0.0-1build2) ...
Setting up libapache2-mod-wsgi-py3 (5.0.0-1build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VNC guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-35-115: ~
```

- Install python pip package

- Install the python flask package

```
ubuntu@ip-172-31-35-115:~$ sudo apt install python3-flask
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
python3-asigref python3-itdangerous python3-pyasnycore python3-pyinotify python3-simplejson python3-werkzeug
Suggested packages:
python3-docs python3-flask-doc python3-pyinotify-doc ipython3 python3-werkzeug-doc python3-lxml python3-watchdog
The following NEW packages will be installed:
python3-asigref python3-flask python3-itdangerous python3-pyasnycore python3-pyinotify python3-simplejson python3-werkzeug
Upgraded: 7 newly installed, 0 to remove and 6 not upgraded.
Not all packages could be upgraded due to dependencies.
This operation will use 581 KB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-asigref all 3.7.2-1 [24.8 kB]
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-itdangerous all 3.1.2-4 [11.6 kB]
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-werkzeug all 3.0.1-3ubuntu0.1 [170 kB]
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-flask all 3.0.2-lubuntu [82.9 kB]
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-pyasnycore all 3.0.2-2 [10.1 kB]
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-pyinotify all 0.9.6-2ubuntu1 [25.0 kB]
Get: http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-simplejson all 3.19.2-build2 [54.5 kB]
Fetched 385 kB in 0s (12.6 MB/s)
Selecting previously unselected package python3-asigref.
Reading package lists... Found 4000 files and directories currently installed.
Preparing to unpack .../0-python3-asigref_3.7.2-1_all.deb...
Unpacking python3-asigref (3.7.2-1) ...
Selecting previously unselected package python3-itdangerous.
Reading package lists... Found 4000 files and directories currently installed.
Preparing to unpack .../1-python3-itdangerous_3.1.2-4_all.deb...
Unpacking python3-itdangerous (3.1.2-4) ...
Selecting previously unselected package python3-werkzeug.
Preparing to unpack .../2-python3-werkzeug_3.0.1-3ubuntu0.1_all.deb...
Unpacking python3-werkzeug (3.0.1-3ubuntu0.1) ...
Selecting previously unselected package python3-flask.
Preparing to unpack .../3-python3-flask_3.0.2-lubuntu_all.deb...
Unpacking python3-flask (3.0.2-lubuntu) ...
Selecting previously unselected package python3-pyasnycore.
Preparing to unpack .../4-python3-pyasnycore_1.0.2-2_all.deb...
Unpacking python3-pyasnycore (1.0.2-2) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../5-python3-pyinotify_0.9.6-2ubuntu1_all.deb...
Unpacking python3-pyinotify (0.9.6-2ubuntu1) ...
Selecting previously unselected package python3-simplejson.
Preparing to unpack .../6-python3-simplejson_3.19.2-build2_amd64.deb...
Unpacking python3-simplejson (3.19.2-build2) ...
Setting up python3-itdangerous (3.1.2-4) ...
Setting up python3-simplejson (3.19.2-build2) ...
Setting up python3-pyasnycore (1.0.2-2) ...
Setting up python3-pyinotify (0.9.6-2ubuntu1) ...
Setting up python3-asigref (3.7.2-1) ...
Setting up python3-flask (3.0.2-lubuntu) ...
Setting up python3-pyinotify (0.9.6-2ubuntu1) ...
Scanning processes...
Scanning Linux images...

Running kernel seems to be up-to-date.

0 services need to be restarted.

0 containers need to be restarted.

0 user sessions are running outdated binaries.

0 VM guests are running outdated hypervisor (qemu) binaries on this host.
```

- Install Sqlite package

```
ubuntu@ip-172-31-35-115:~$ sudo apt install sqlite3
Selecting previously unselected package python3-flask.
Preparing to unpack .../0-python3-flask_3.0.2-lubuntu1_all.deb...
Unpacking python3-flask (3.0.2-lubuntu1) ...
Selecting previously unselected package python3-pyasnycore.
Preparing to unpack .../1-python3-pyasnycore_1.0.2-2_all.deb...
Unpacking python3-pyasnycore (1.0.2-2) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../2-python3-pyinotify_0.9.6-2ubuntu1_all.deb...
Unpacking python3-pyinotify (0.9.6-2ubuntu1) ...
Selecting previously unselected package python3-simplejson.
Preparing to unpack .../3-python3-simplejson_3.19.2-build2_amd64.deb...
Unpacking python3-simplejson (3.19.2-build2) ...
Intalling python3-simplejson (3.19.2-build2) ...
Setting up python3-simplejson (3.19.2-build2) ...
Setting up python3-werkzeug (3.0.1-3ubuntu0.1) ...
Setting up python3-itdangerous (3.1.2-4) ...
Setting up python3-asigref (3.7.2-1) ...
Setting up python3-flask (3.0.2-lubuntu1) ...
Setting up python3-pyinotify (0.9.6-2ubuntu1) ...
Scanning processes...
Scanning Linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

0 VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-35-115:~$ sudo apt install sqlite3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
No packages to install or upgrade.
0 packages to download.
0 packages to upgrade.
0 packages to remove.
0 packages not fully installed or removed.
Selecting previously unselected package sqlite3.
(Reading database... 76264 files and directories currently installed.)
Preparing to unpack .../0-sqlite3_3.45.1-1ubuntu2_amd64.deb...
Unpacking sqlite3 (3.45.1-1ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning Linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

0 VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-35-115:~$
```

- Install ACL package

The screenshot shows a Windows desktop environment with several open windows:

- File Explorer:** Shows a folder structure under "Assignments" containing "Project 1" and "Project 2". "Project 2" contains files like "flaskapp.py", "index.html", "KeyWin.ppk", "Limerick-1.txt", "Limerick-1 (1).txt", "Limerick-1 (2).txt", "Limerick-1 (3).txt", and "pixelsx-pixabay-268533.jpg". It also lists "users.db" and "WAC1 Launching_EC2_Instance_on_AWS_Web Server_4.pdf".
- Terminal Window:** Titled "flaskapp.py", it runs the command "python flaskapp.py" and displays the output of the application. The application is a simple web server serving "index.html".
- Taskbar:** Shows icons for File Explorer, Task View, Start, Task Manager, and several pinned applications.

- Change permission of the home directory and check for error logs

3. Design an interactive web page

First, I create a flaskapp directory in the home directory with `mkdir flaskapp` and set the default and directory permission to allow all users to read, write and execute files with `chmod 777 flaskapp` and `setfacl -d -m "o::rwx" flaskapp/` then change the working directory to flaskapp using `cd flaskapp`.

The screenshot shows a terminal window with several tabs open. The current tab contains Python code for a Flask application named `flaskapp.py`. The code includes database queries to select from a `users` table, handle file uploads, and run the application. Below the code, the terminal shows the command `python flaskapp.py` being run, and the output indicates the application is running on port 5000.

```
rows = execute_query('***SELECT * FROM users***')
return '<br>'.join(str(row) for row in rows)

# load user file and overwrite old file
.route('/upload/<userid>', methods=['POST'])
.upload_file(userid):
    f = request.files['file']
    filepath = os.path.join(project_root, f.filename)
    if os.path.exists(filepath):
        os.remove(filepath)
    f.save(filepath)
    insert_query(query='***UPDATE users SET filename = ? WHERE id = ?***', args=(f.filename, userid))
    return redirect('/user//'.format(userid))

# download user file
.route('/download/<filename>')
.download_file(filename):
    return send_from_directory(project_root, filename, as_attachment=True)

if __name__ == '__main__':
    app.run()

C:\Users\Admin\AppData\Local\Programs\Python\Python311\python.exe "D:\Hung\UC\Classes\INTRO TO CLOUD COMPUTING\Assignments\Project 2\flaskapp.py"
```

The screenshot shows a PyCharm IDE interface with the following details:

- Project:** The project structure is visible on the left, showing a folder named "Assignments" containing "Project 1", "Project 2", and "index.html".
- Terminal:** A terminal window titled "flaskapp.py" is open, displaying the code for a Flask application. The code includes a route for "/user" that executes a database query to select all users.
- Output:** The terminal output shows the execution of the command "python flaskapp.py" and the resulting log from the Flask application, which includes a warning about outdated binaries and a successful connection to port 5000.
- Run Tab:** The "Run" tab at the bottom shows the command "flaskapp" and the path "C:\Users\Admin\AppData\Local\Programs\Python\Python311\python.exe" followed by the file path "D:\Hung\UC\Classes\INTRO TO CLOUD COMPUTING\Assignments\Project 2\flaskapp.py".
- Status Bar:** The status bar at the bottom right shows the current file as "Current File", the file size as "121:30 (3984 chars, 133 line breaks)", the encoding as "CRLF", the character set as "UTF-8", the number of spaces as "4 spaces", and the Python version as "Python 3.11 (stimpl-main)".

```
ubuntu@ip-172-31-35-115:~  
total 4  
drwxrwxrwx 2 ubuntu ubuntu 4096 Oct 4 23:03 flaskapp  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
ubuntu@ip-172-31-35-115:~/flaskapp$ nano flaskapp.py  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
ubuntu@ip-172-31-35-115:~/flaskapp$ nano flaskapp.py  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
ubuntu@ip-172-31-35-115:~/flaskapp$ nano flaskapp.py  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
[Fri Oct 04 23:19:18.862202 2024] [wsgi:error] [pid 24716:tid 127973411784384] [client 129.137.96.15:36257] File "/usr/lib/python3/dist-packages/flask/app.py", line 870, in full_dispatch_request  
[Fri Oct 04 23:19:18.862203 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257) rv = self.dispatch_request()  
[Fri Oct 04 23:19:18.862205 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257) ^^^^^^^^^^  
[Fri Oct 04 23:19:18.862207 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257) File "/usr/lib/python3/dist-packages/flask/app.py", line 855, in dispatch_request  
[Fri Oct 04 23:19:18.862210 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257) return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]  
[Fri Oct 04 23:19:18.862212 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257)  
[Fri Oct 04 23:19:18.862214 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257)  
[Fri Oct 04 23:19:18.862216 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257)  
[Fri Oct 04 23:19:18.862218 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257)  
[Fri Oct 04 23:19:18.862221 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257) conn = sql.connect("users.db")  
[Fri Oct 04 23:19:18.862223 2024] [wsgi:error] [pid 24716:tid 127973411784384] (client 129.137.96.15:36257) sqlite3.OperationalError: unable to open database file  
 ``C  
ubuntu@ip-172-31-35-115:~/flaskapp$ nano flaskapp.py  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo tail -f /var/log/apache2/error.log  
[Fri Oct 04 23:20:23.051230 2024] [wsgi:error] [pid 24793:tid 127973411784384] [client 129.137.96.15:7235] File "/usr/lib/python3/dist-packages/flask/app.py", line 870, in full_dispatch_request  
[Fri Oct 04 23:20:23.051231 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) rv = self.dispatch_request()  
[Fri Oct 04 23:20:23.051235 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) ^^^^^^^^^^  
[Fri Oct 04 23:20:23.051237 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) File "/usr/lib/python3/dist-packages/flask/app.py", line 855, in dispatch_request  
[Fri Oct 04 23:20:23.051240 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]  
[Fri Oct 04 23:20:23.051242 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235)  
[Fri Oct 04 23:20:23.051244 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) File "/var/www/html/flaskapp/flaskapp.py", line 89, in viewdb  
[Fri Oct 04 23:20:23.051247 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) c.execute_query("""SELECT * FROM users""")  
[Fri Oct 04 23:20:23.051249 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) ^^^^^^^^^^  
[Fri Oct 04 23:20:23.051251 2024] [wsgi:error] [pid 24793:tid 127973411784384] (client 129.137.96.15:7235) AttributeError: 'sqlite3.Cursor' object has no attribute 'execute_query'  
 ``C  
ubuntu@ip-172-31-35-115:~/flaskapp$ nano flaskapp.py  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
ubuntu@ip-172-31-35-115:~/flaskapp$ nano flaskapp.py  
ubuntu@ip-172-31-35-115:~/flaskapp$ sudo apachectl restart  
ubuntu@ip-172-31-35-115:~/flaskapp$ chmod 777 users.db  
ubuntu@ip-172-31-35-115:~/flaskapp$ cd  
ubuntu@ip-172-31-35-115:~/flaskapp$ chmod 777 flaskapp/  
ubuntu@ip-172-31-35-115:~$
```

To create the web app, I first create a user database with `sqlite3 users.db` and change database file permission to allow all users to all users to read, write and execute the file using `chmod 777 users.db`. Then I add index.html, user.html, flaskapp.wsgi and `flaskapp.py` in the current directory. Then, I modify the apache configuration file located at `/etc/apache2/sites-enabled/000-default.conf` using `nano /etc/apache2/sites-enabled/000-default.conf`. Finally, I restart the server with the new configuration.

- Add index.html

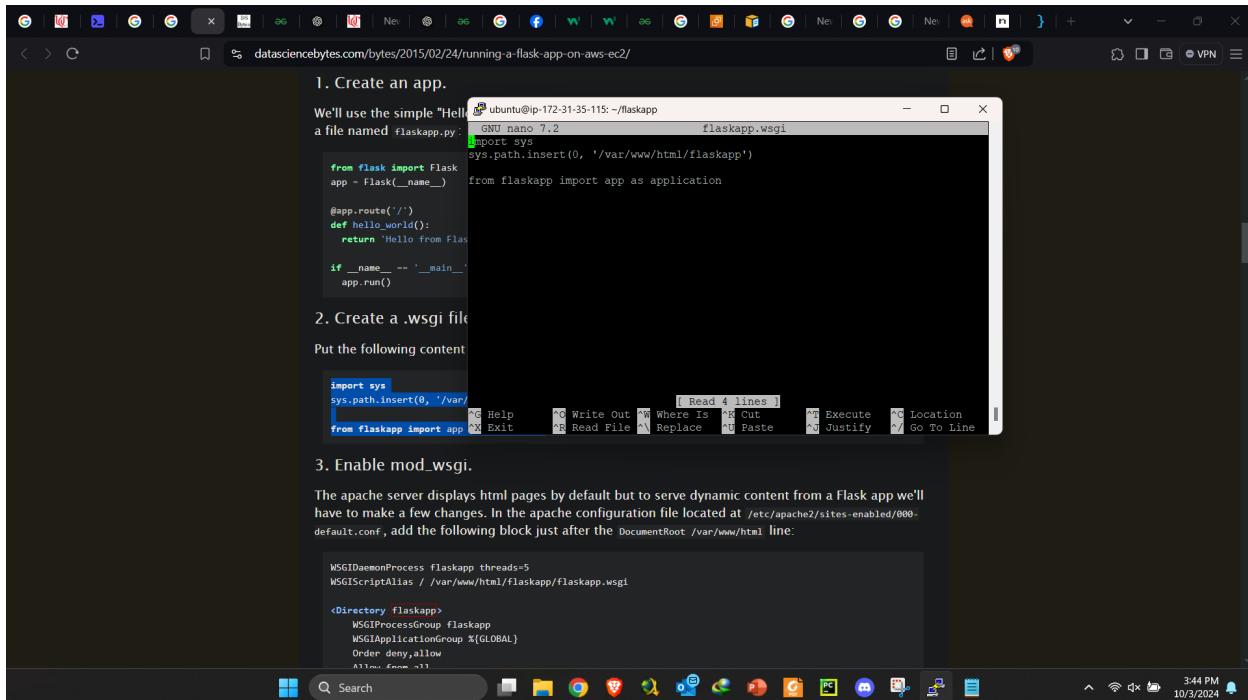
```
ubuntu@ip-172-31-35-115:~/flaskapp
$ my_name_12
$ !DOCTYPE html>
<html>
  <head>
    <title>Webpage</title>
  <body>
    <h1>Hung Nguyen EC2 Webpage</h1>
    <form method="post" >
      <label for="uname">User name:</label><br>
      <input type="text" id="uname" name="uname"><br>
      <label for="pwd">Password:</label><br>
      <input type="password" id="pwd" name="pwd"><br>
      <label for="fname">First Name (Register Only)</label><br>
      <input type="text" id="fname" name="fname"><br>
      <label for="lname">Last Name (Register Only)</label><br>
      <input type="text" id="lname" name="lname"><br>
      <label for="email">Email (Register Only)</label><br>
      <input type="text" id="email" name="email"><br>
      <input type="submit" formaction="/login" value="Login">
      <input type="submit" formaction="/register" value="Register">
    </form>
  </body>
</html>
```

- Add user.html

```
ubuntu@ip-172-31-115-:~/flaskapp
GNU nano 7.2
<html>
<head>
<title>User info</title>
</head>
<body>
<h1>User info</h1>
<div class="info">
<strong>Username:</strong> {{ user[1] }}<br>
<strong>First Name:</strong> {{ user[2] }}<br>
<strong>Last Name:</strong> {{ user[3] }}<br>
<strong>Email:</strong> {{ user[5] }}<br>
</div>
<form method="POST" action={{ url_for('upload') }} enctype="multipart/form-data">
<strong>Upload File:</strong> <br>
<input type="file" name="file"> <br>
<input type="submit" value="Upload"> <br>
</form>
<strong>Download File:</strong> {{ file[0] }} <br>
<strong>File Count:</strong> {{ file[1] }} <br>
<a href={{ url_for('download') }}>Download</a>
</div>
</body>

```

- Add flaskapp.wsgi



- Modify the apache configuration file located at `/etc/apache2/sites-enabled/000-default.conf`

The screenshot shows a desktop environment with several windows open:

- A browser window displaying a guide on running a Flask app on AWS EC2.
- A terminal window showing the content of `app.py`, which contains a simple Flask application.
- A terminal window showing the configuration of an Apache server via `nano`. It includes the creation of a `.wsgi` file and modifications to the `000-default.conf` file.
- A file dialog window titled "File Name to Write" showing the path `/etc/apache2/sites-enabled/000-default.conf`.
- A status bar at the bottom showing system icons and the date/time: 10/3/2024, 3:55 PM.

- Add flaskapp.py

The screenshot shows a terminal window displaying the content of `flaskapp.py`. The code defines a Flask application that connects to a SQLite database and handles user login requests. It includes functions for connecting to the database, executing queries, and rendering templates.

```

#!/usr/bin/env python3
# coding: utf-8
from flask import Flask, render_template, g, request, redirect, send_from_directory
import sqlite3 as sql
import os

project_root = os.path.dirname(__file__)
template_path = os.path.join(project_root, './')
DATABASE = os.path.join(project_root, 'users.db')
PDFFILE = ('', 0)

app = Flask(__name__, template_folder=template_path)
app.config.from_object(__name__)

# Connect to database
def connect_to_database():
    return sql.connect(app.config['DATABASE'])

# Get database object
def get_db():
    db = getattr(g, 'db', None)
    if db is None:
        db = g.db = connect_to_database()
    return db

# Disconnect from database
@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, 'db', None)
    if db is not None:
        db.close()

# Run Select queries
def execute_query(query, args=()):
    db = get_db()
    rows = db.execute(query, args).fetchall()
    db.commit()
    return rows

# Run other queries
def insert_query(query, args=()):
    db = get_db()
    db.execute(query, args)
    db.commit()
    return

# Create table if not exist
def create_tb():
    users_query="""CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY AUTOINCREMENT, uname TEXT, pword TEXT, fname TEXT, lname TEXT, email TEXT, filename TEXT DEFAULT '')"""
    return

# First webpage
@app.route('/')
def index():
    create_tb()
    return render_template('index.html')

# Get user id from username and password
def users_id(uname, password):
    rows = execute_query("SELECT id FROM users WHERE uname = ? AND pword = ?",
    (uname, password))
    return rows[0].id

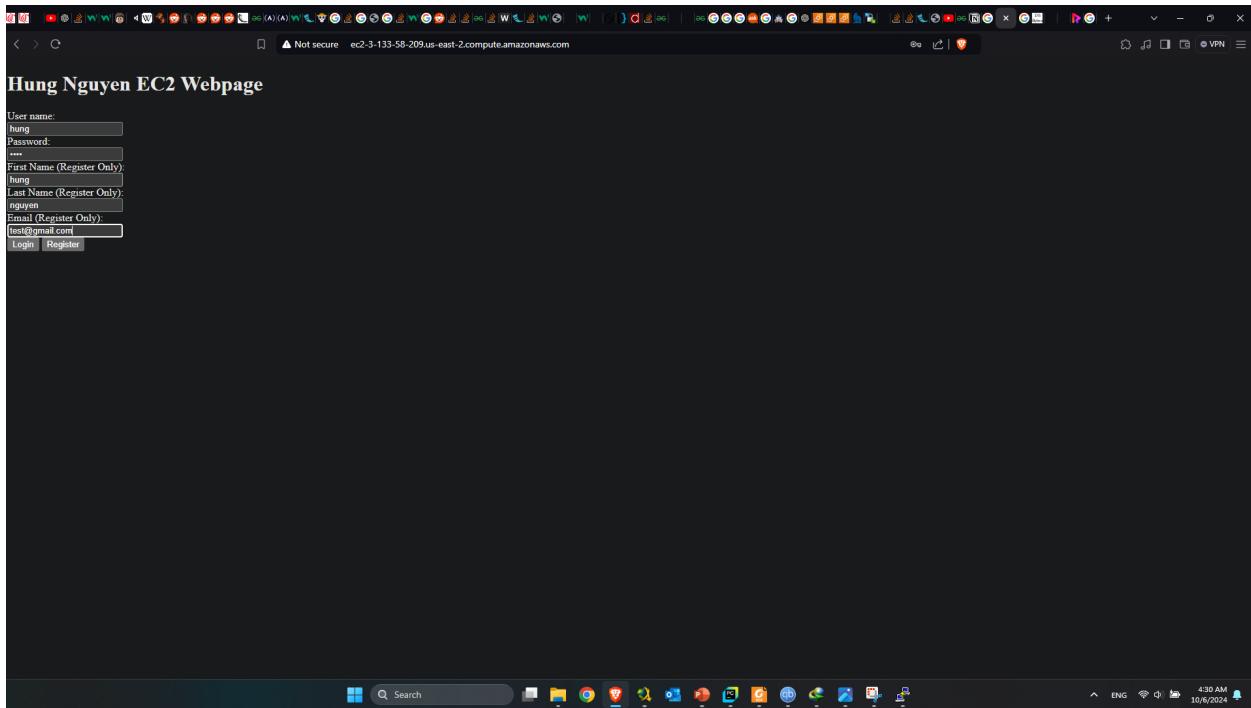
```

- Restart the server with the new configuration

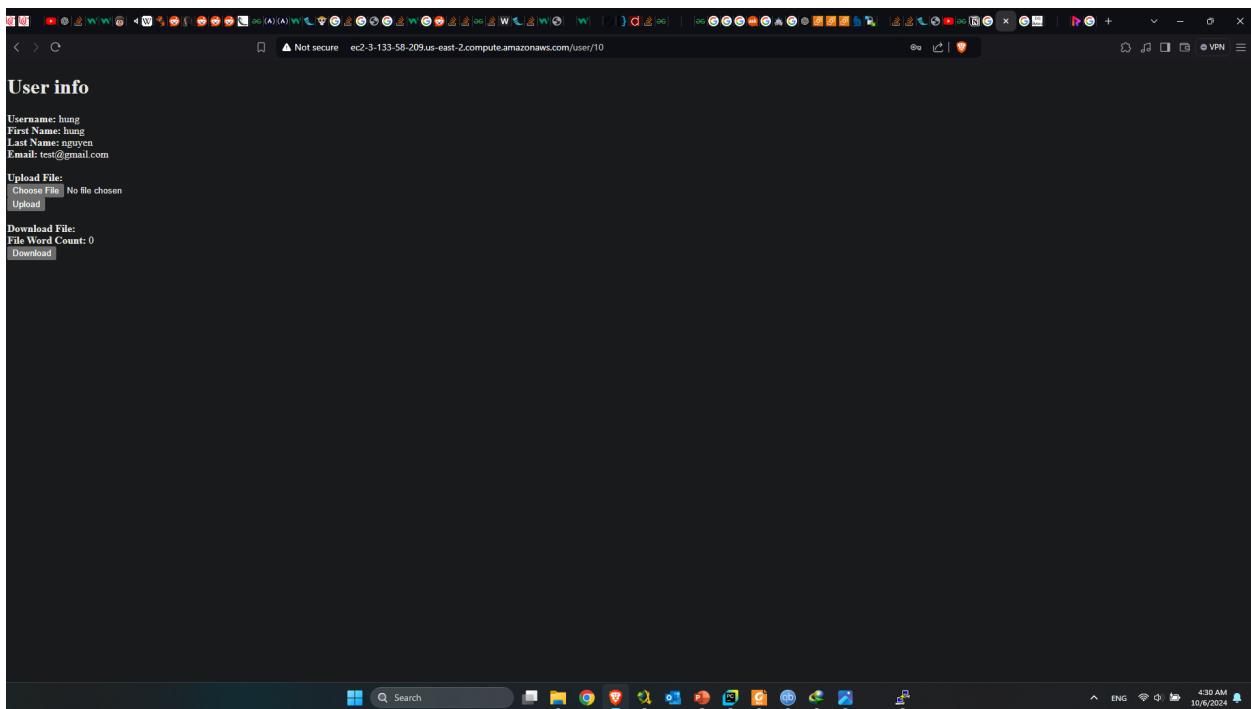
4. Webpage: <http://ec2-3-133-58-209.us-east-2.compute.amazonaws.com/>

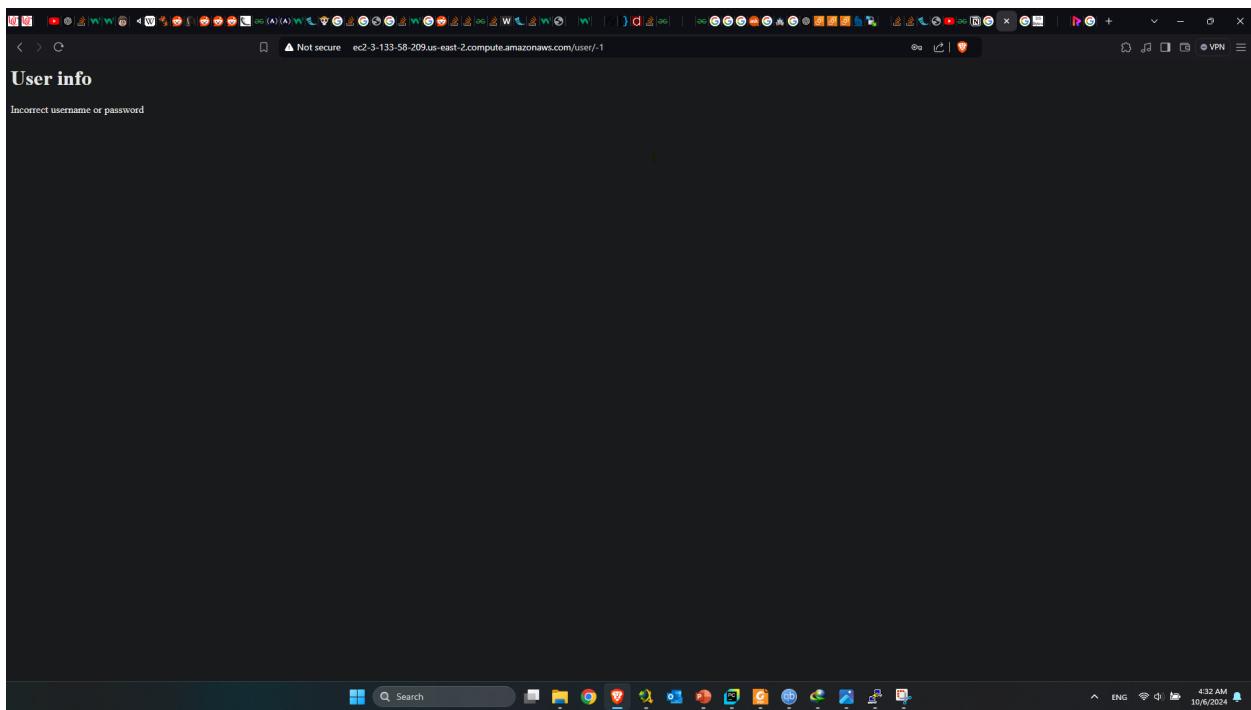
To sign in, I would put in my username and password then press on the login button. To register, I would put in my username, password, first name, last name, and email then press the Register button. Once I have logged in successfully or registered, the User info page is displayed with my username, first name, last name, and email. On the other hand, if I enter an incorrect password or username, the webpage will say "Incorrect username or password".

- Login page



- User info page





3. Extra credit

For uploading files, I would press on the Choose file button then select the file I want to upload. Then, I click on the upload button to upload the file to the server and show the word count and the download link. For downloading files, I would click on the Download button when there is a file name shown in the Download File section.

- User info page upload file and download file

