School of Computer Science and Engineering (SCOPE)

Fall Semester 2025-26

CBS3005 - Cloud, Microservices and Applications

LAB ASSESSMENT 1

Name: Riya Autade
Registration No.: 22BBS0079

Q1. Create and manage EC2 instances on AWS. Launch an application on an EC2 instance in one region and then migrate that instance to another region. Additionally, set up another EC2 instance to install MySQL, create a database (e.g., student or employee database), and perform basic SQL operations. (5.0 marks)

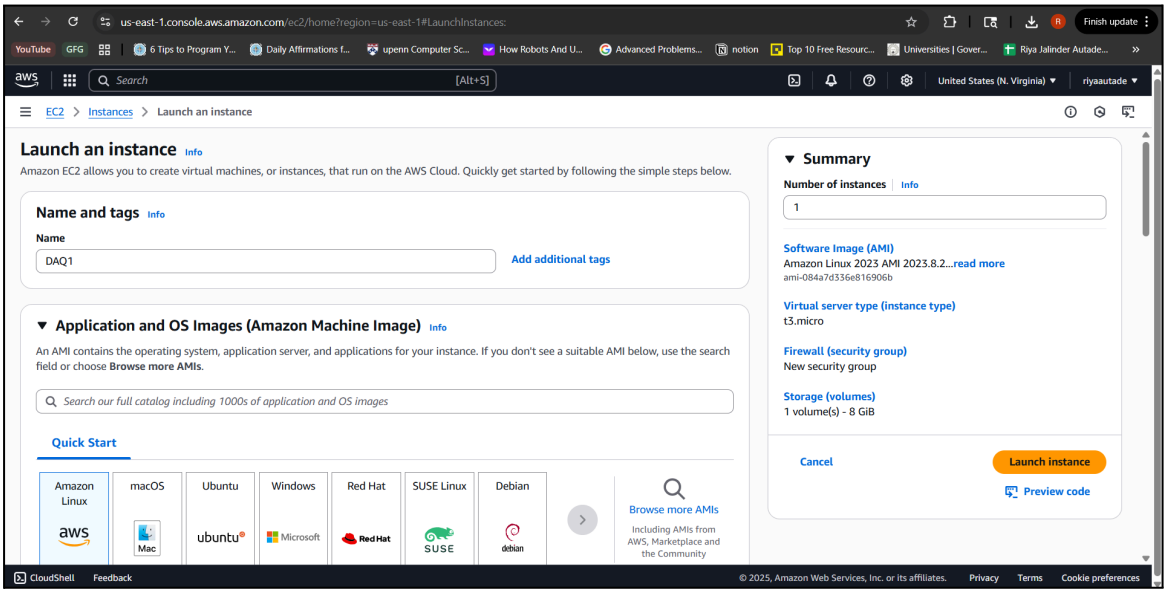
Part A: EC2 Instance & Application Migration

Approach:

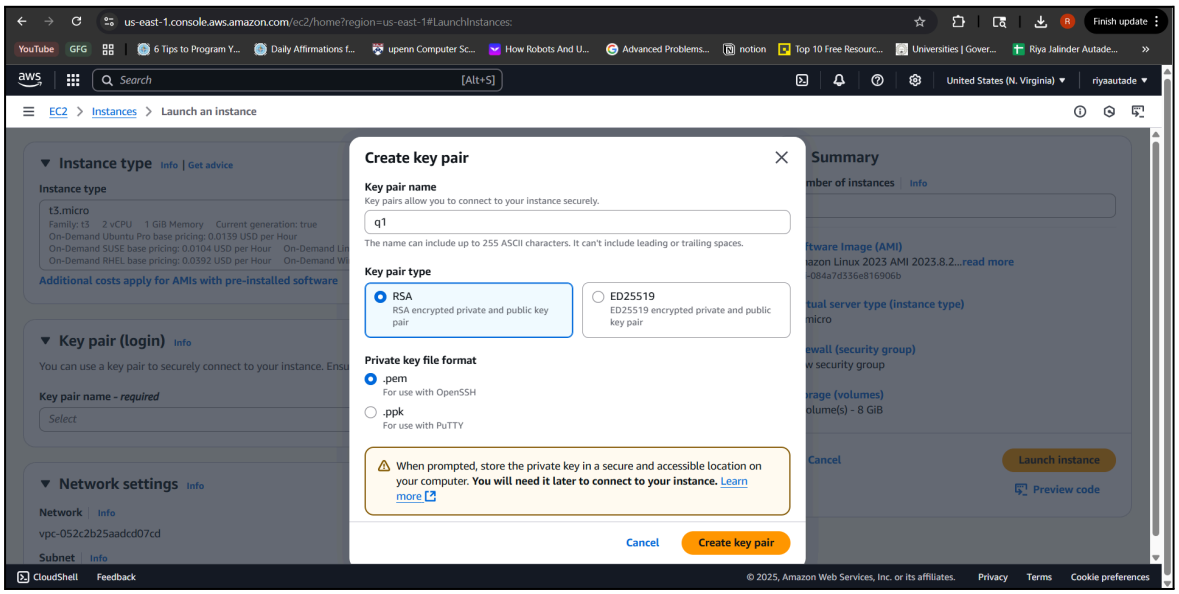
1. Create an EC2 instance.
    - Launch a virtual server (EC2) on AWS.
2. Launch an application on the EC2 instance in one region.
    - Deploy a simple web app (Python Flask app).
3. Migrate that instance to another region.
    - Move this EC2 instance (with the application) from one AWS region (us-east-1) to another (ap-south-1).

STEPS:

1. **Launch an EC2 instance (DAQ1) with Amazon Linux OS and of type t3.micro**

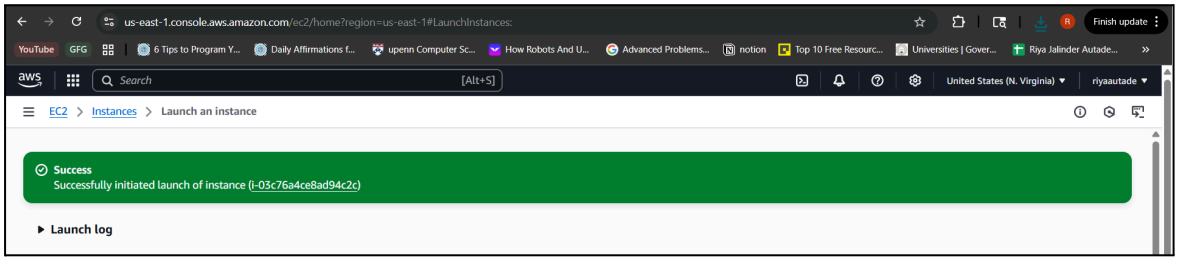


- Create a key pair for SSH (q1.pem) with Network Settings: HTTP, SSH, HTTPS



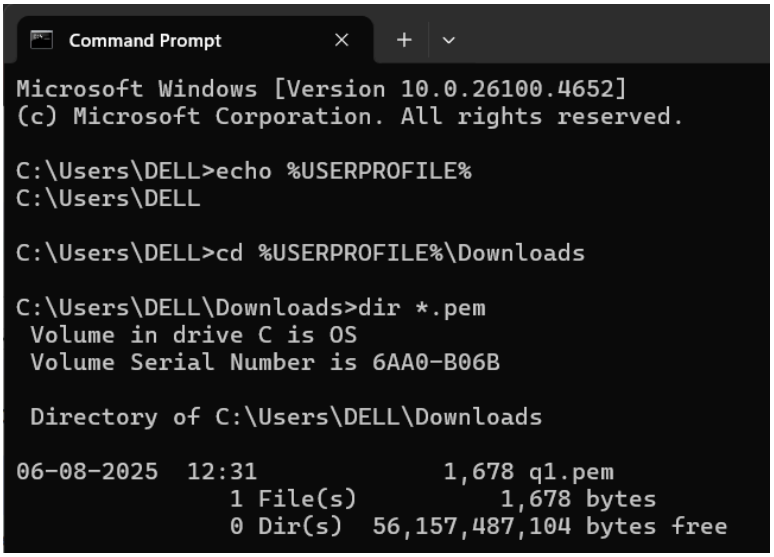
- Keep the rest default and click on Launch Instance.

EC2 instance (DAQ1) launched successfully

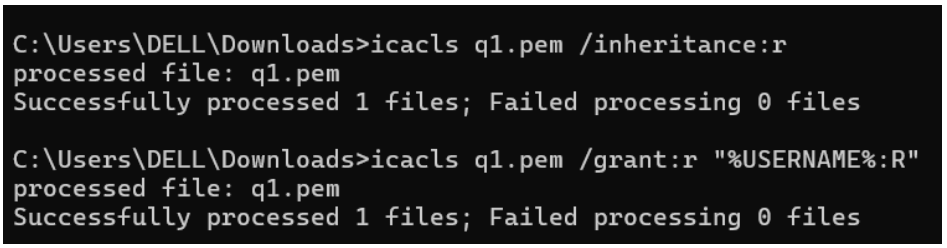


2. **Connect to the EC2 (DAQ1) via SSH using Command Prompt**

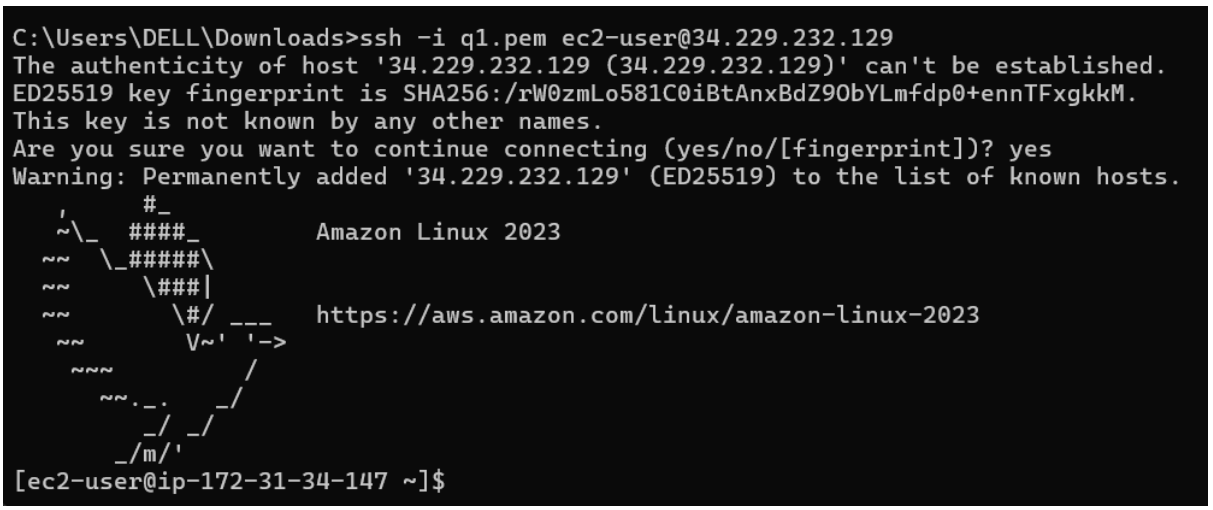
- Navigate to folder where your .pem file is:



- Change file permissions:



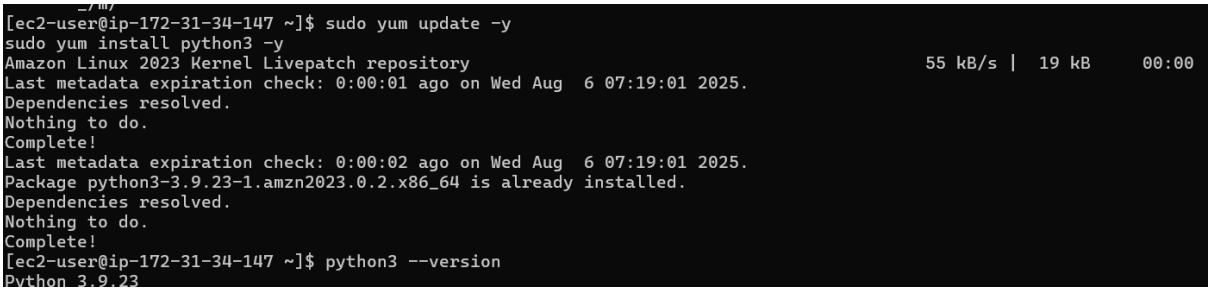
- Connect via SSH with the Public IPv4 of the EC2 instance:



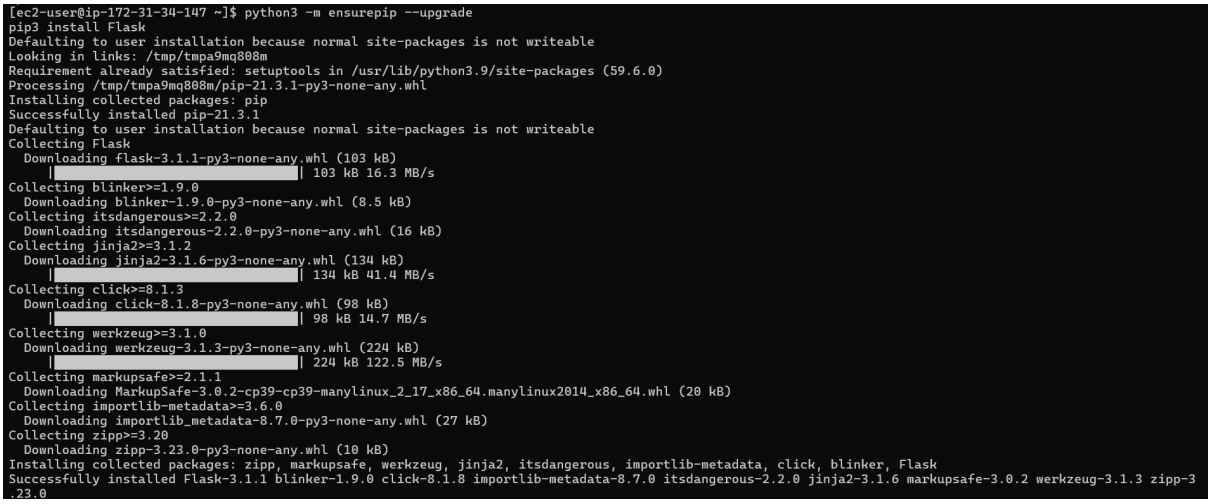
We are in the EC2 terminal now.

3. Flask App + Deployment on EC2

- Install Python on the EC2 instance & check version:



- Install Flask on the EC2 instance



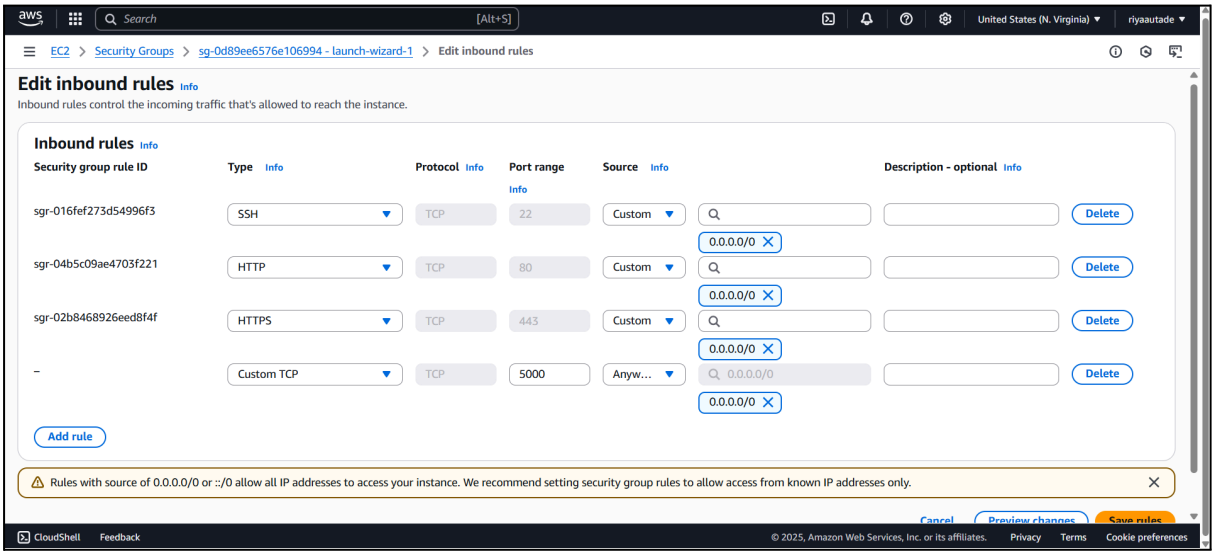
- Move from EC2 terminal to nano editor (command: nano app.py)

- Create the Flask Application in the nano editor & keep port=5000

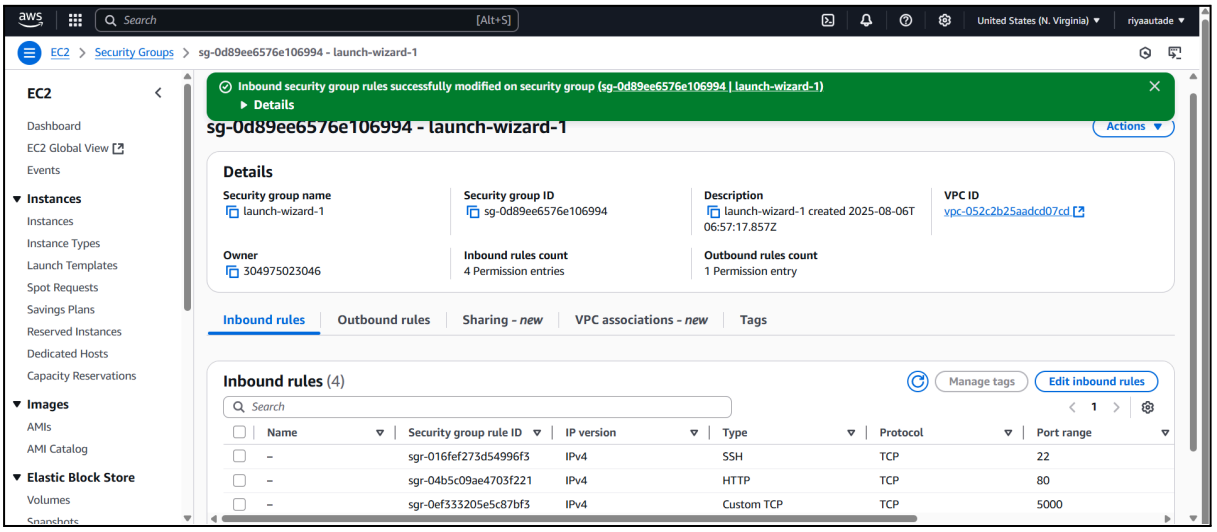
```
ec2-user@ip-172-31-34-147:~
  GNU nano 8.3                                                    app.py
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "<h1>Hello from Python Flask on EC2!</h1><p>This is Riya's live web app.</p>"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

- Press Ctrl + O, then Enter to save & press Ctrl + X to exit the nano editor.

- Open Security Group, Edit the Inbound Rules & Add Rule to include port 5000



- Click on Save rules and add it successfully



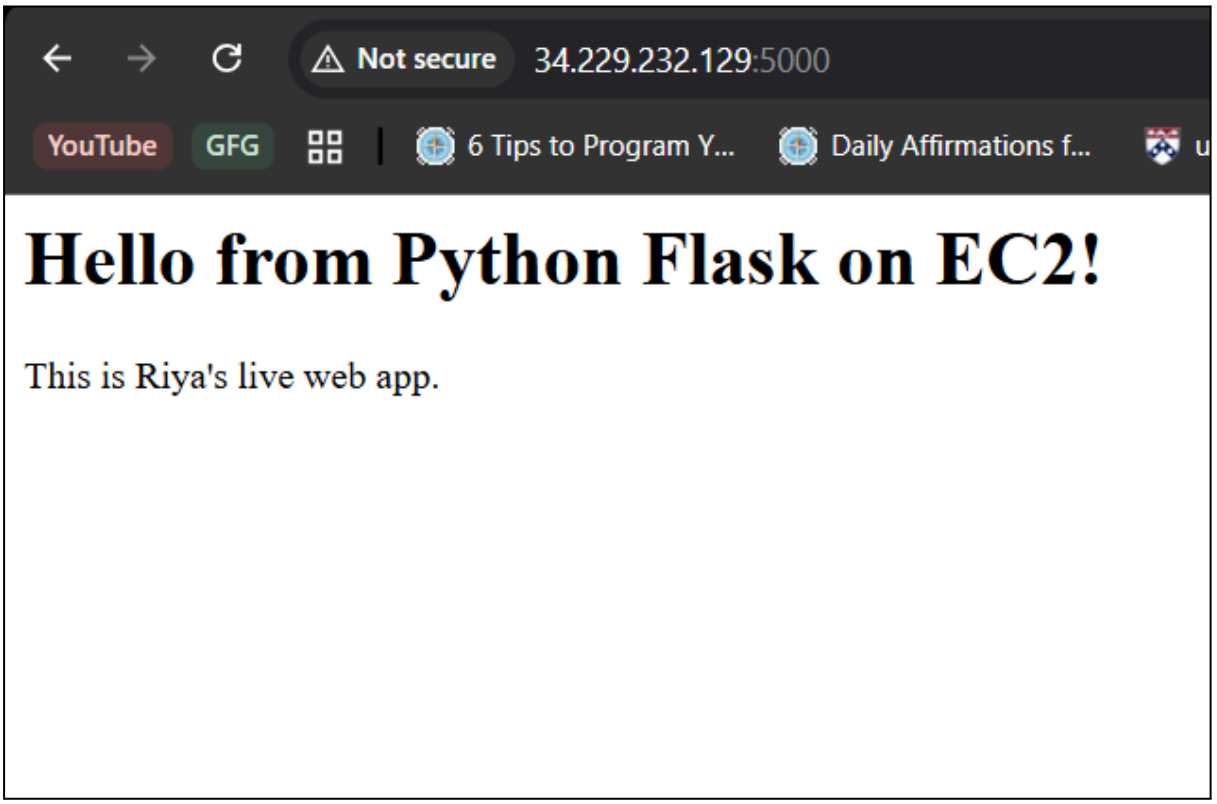
- Run the Flask App

```
[ec2-user@ip-172-31-34-147 ~]$ nano app.py
[ec2-user@ip-172-31-34-147 ~]$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.34.147:5000
Press CTRL+C to quit
```

Successful. This will start the Flask server on port 5000, accessible by public IP.

View the App in Browser:

- Type in browser the link:- **http://<Your-EC2-Public-IP>:5000**
- Replace it with the public IPv4 of the EC2 instance which you can find it from the EC2 dashboard under "Public IPv4 address".

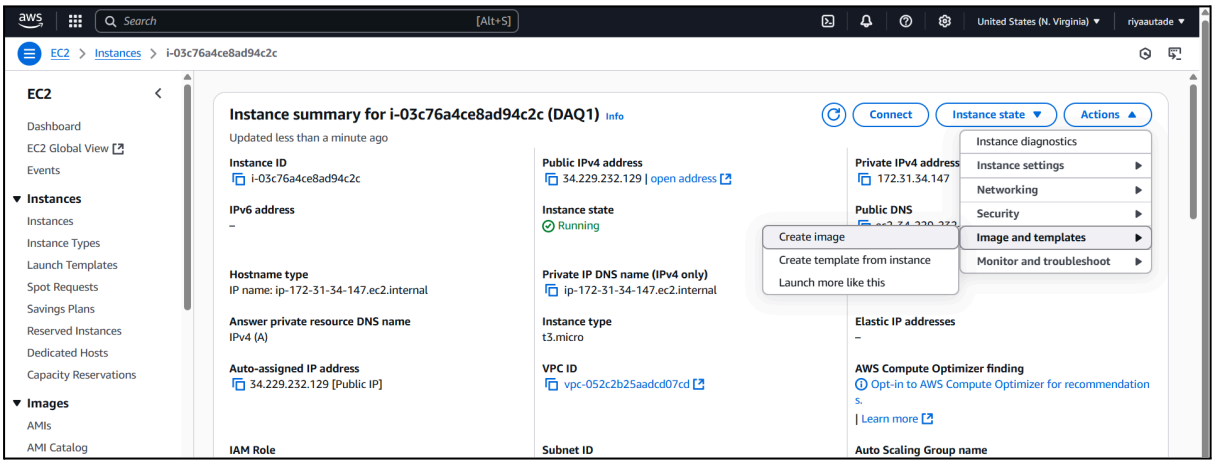


Flask App is successfully hosted on AWS EC2!

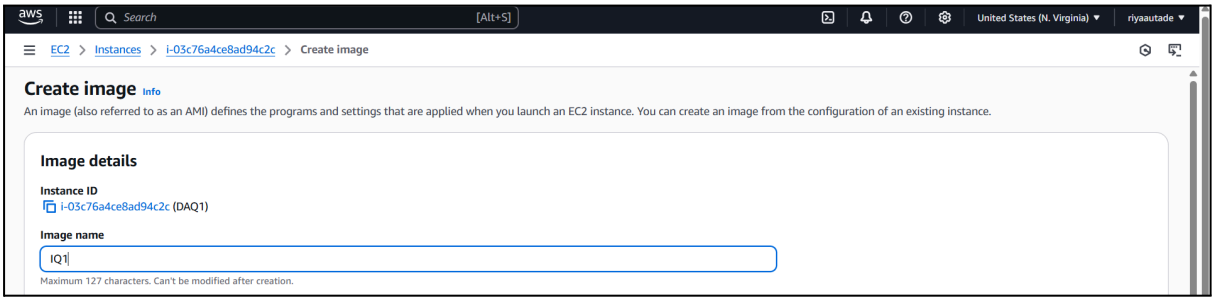
4. EC2 Instance Migration from one Region to another:

Create an image of the EC2 instance in Region A, copy it to Region B, and launch a new EC2 instance from that image. The Flask app and its configuration get copied.

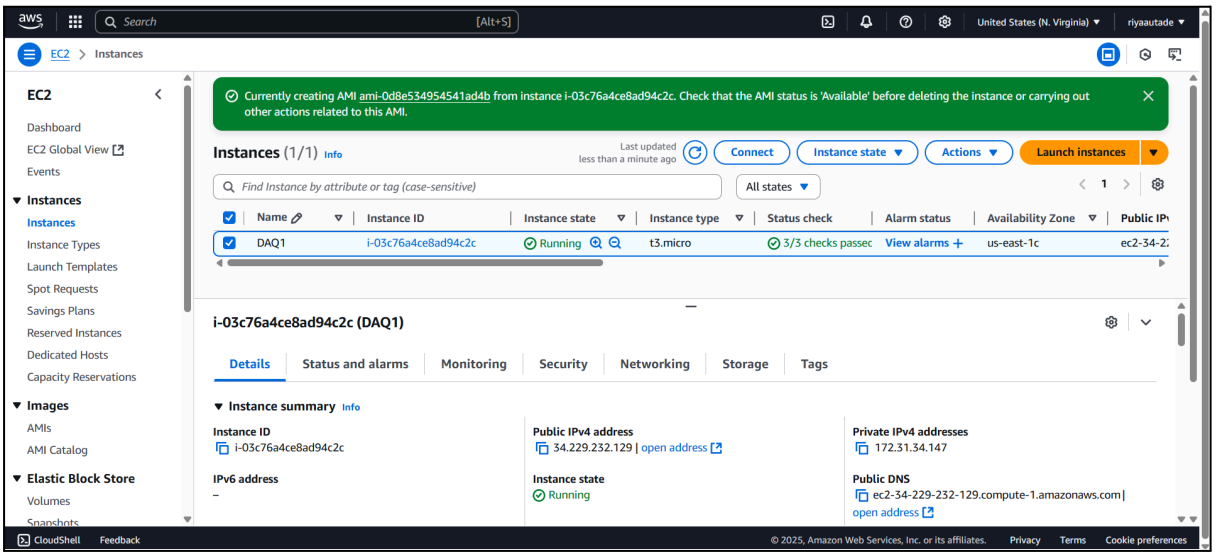
- Create an AMI (Amazon Machine Image)
- EC2 > Instances & at the top, click **Actions > Image and templates > Create image**



- Give the image a name, leave rest default, and click on Create Image

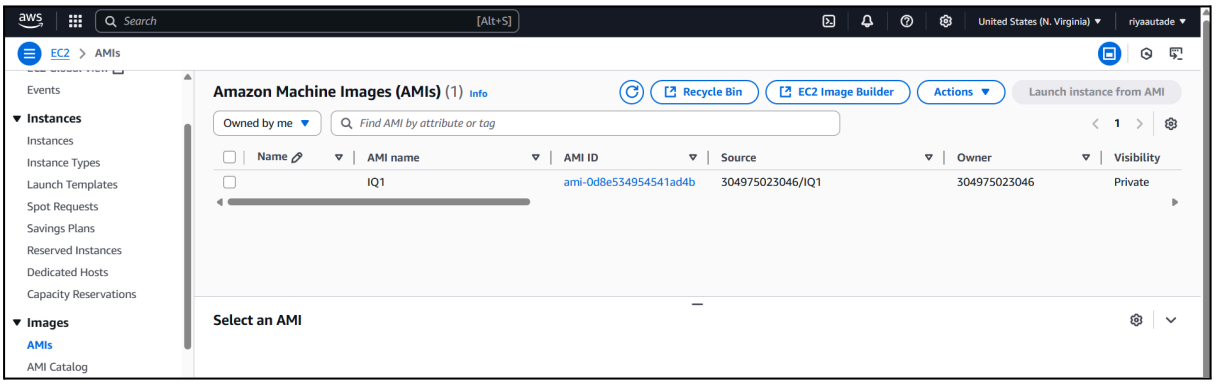


- Wait a few minutes & AMI is created successfully

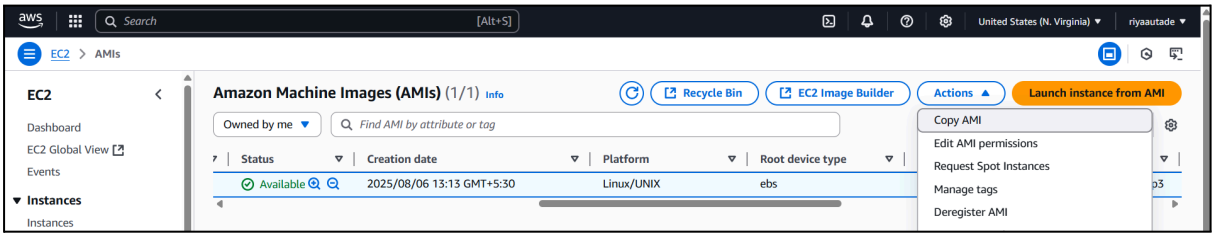


Find the AMI in EC2

- In the left sidebar, click AMIs (under Images)

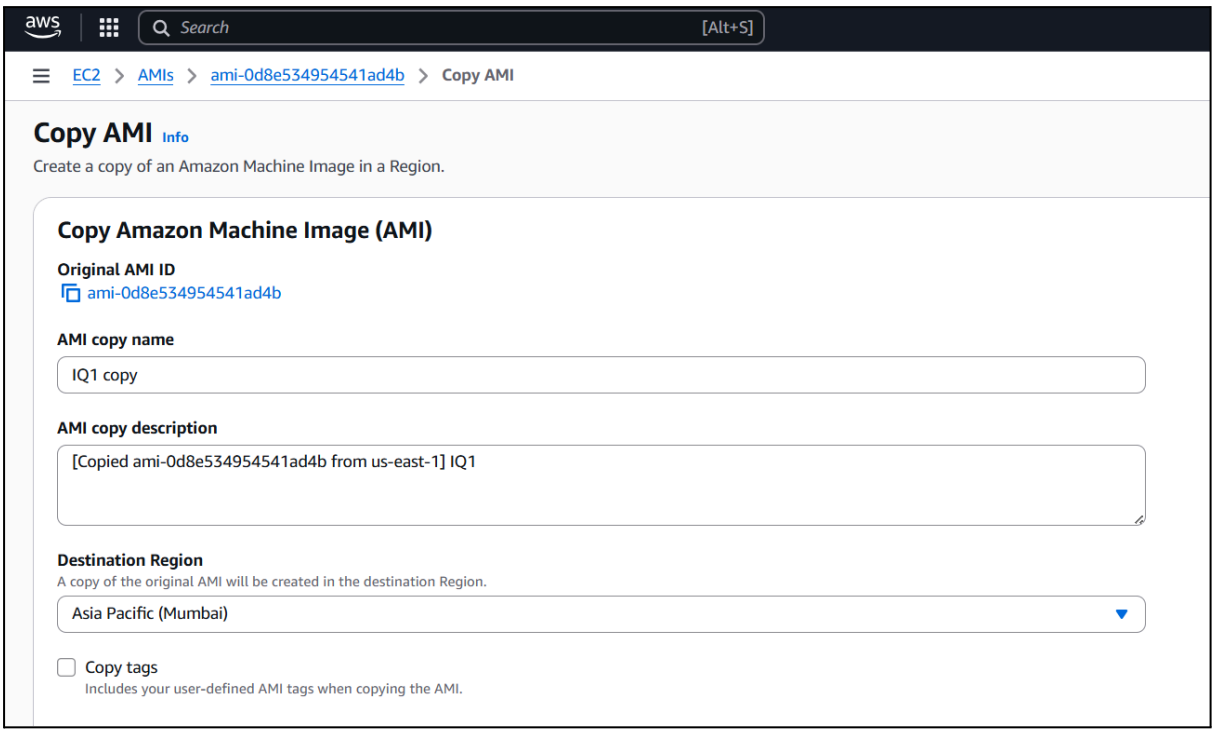


- Once the AMI status shows “available”, select it & then Click Actions > Copy AMI

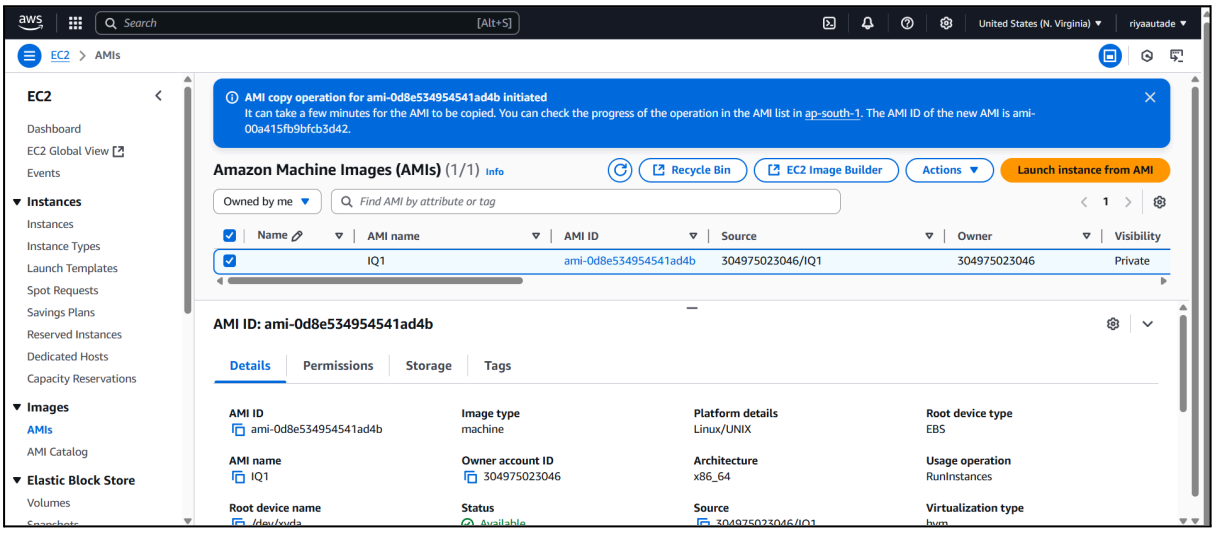


To copy the AMI to a New Region: **ap-south-1:**

- Name AMI Copy, Choose in “Destination Region”: Asia Pacific(Mumbai) & Click Copy



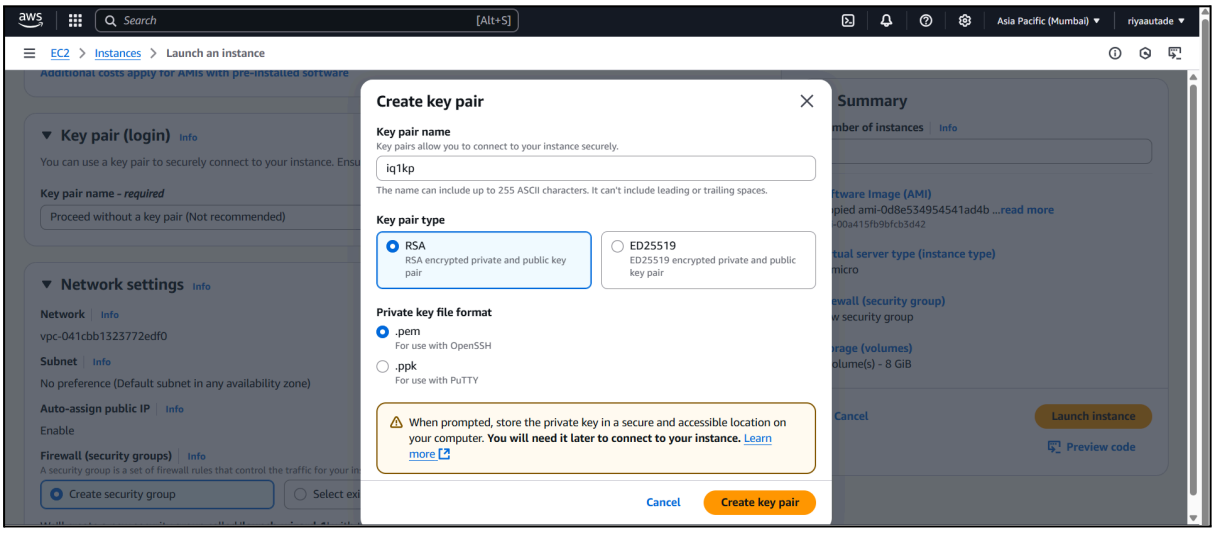
- Wait again & the copied AMI will show up in the new region.



- Switch to target region (ap-south-1) on top-right corner dropdown in AWS Console

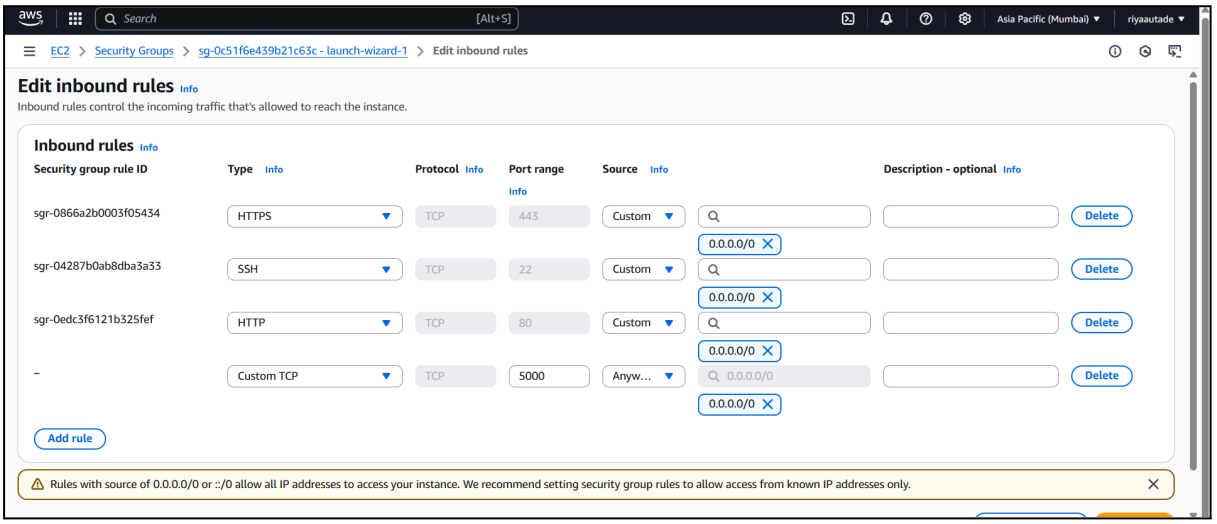
Launch a new EC2 Instance from the copied AMI:

- Go to AMIs — you'll see your copied AMI there
- Select it and click Launch Instance
- Choose instance type (t3.micro) and create a key pair (iq1kp.pem)



AMI Copy EC2 instance is successfully launched:

- Again allow port 5000 for the new region similarly



- SSH into new copied EC2 instance using the new Public IPv4:

```
C:\Users\DELL\Downloads>ssh -i iq1kp.pem ec2-user@13.201.168.14
The authenticity of host '13.201.168.14 (13.201.168.14)' can't be established.
ED25519 key fingerprint is SHA256:OIsnDeG8+X7rNcmJgiLemsT3CdG2V1c+1CYNSvdRKgk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.201.168.14' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Aug  6 07:15:48 2025 from 136.233.9.106
[ec2-user@ip-172-31-15-181 ~]$
```

- Check if Flask is there since we moved to a new region

```
[ec2-user@ip-172-31-15-181 ~]$ pip3 show Flask
Name: Flask
Version: 3.1.1
Summary: A simple framework for building complex web applications.
Home-page:
Author:
Author-email:
License:
Location: /home/ec2-user/.local/lib/python3.9/site-packages
Requires: blinker, click, importlib-metadata, itsdangerous, jinja2, markupsafe, werkzeug
```

- Run the Flask App again

```
[ec2-user@ip-172-31-15-181 ~]$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.15.181:5000
Press CTRL+C to quit
```

Visit the App in Browser with public IPv4 and port 5000:

We've successfully migrated an EC2 app between regions!

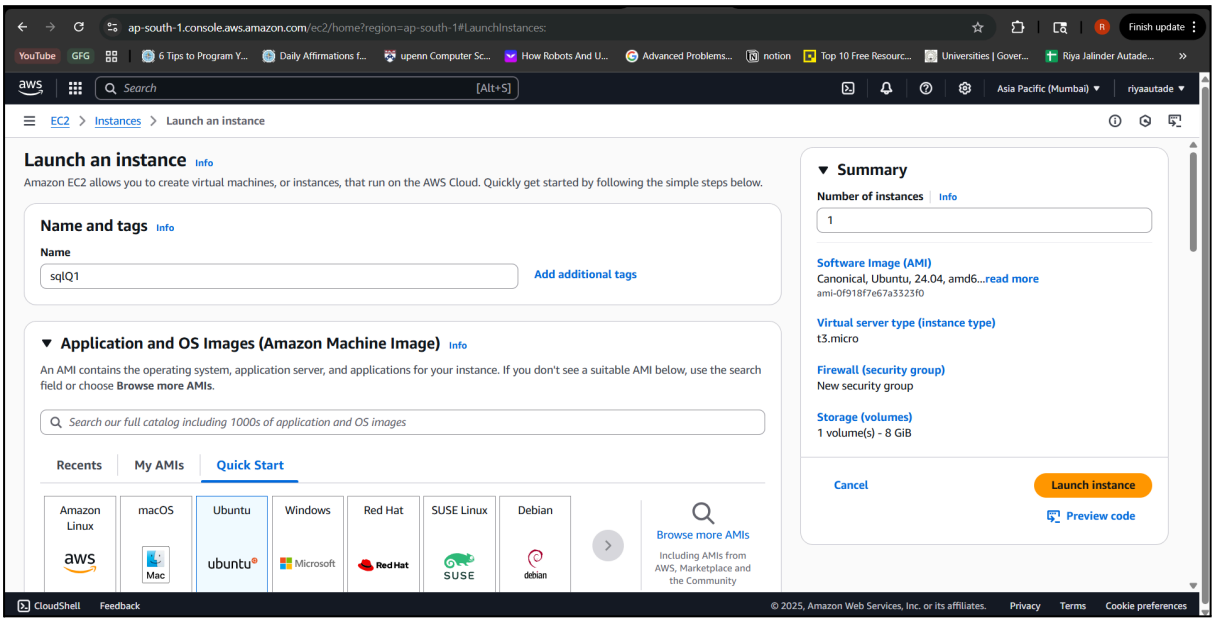
Part B: EC2 + MySQL Setup

Approach:

1. Launch another EC2 instance.
2. Install MySQL server.
3. Create a database.
4. Perform basic SQL operations.
    - E.g., create tables, insert records, and run queries.

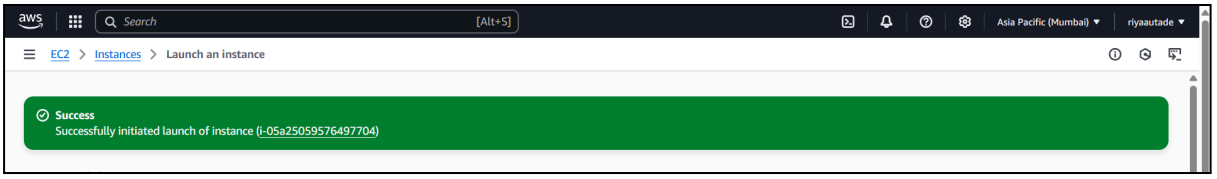
STEPS:

1. Launch a New EC2 Instance with Ubuntu OS



- Choose Instance type: t2.micro and use existing key pair: iq1kp.pem
- Keep the rest default & click on Launch Instance.

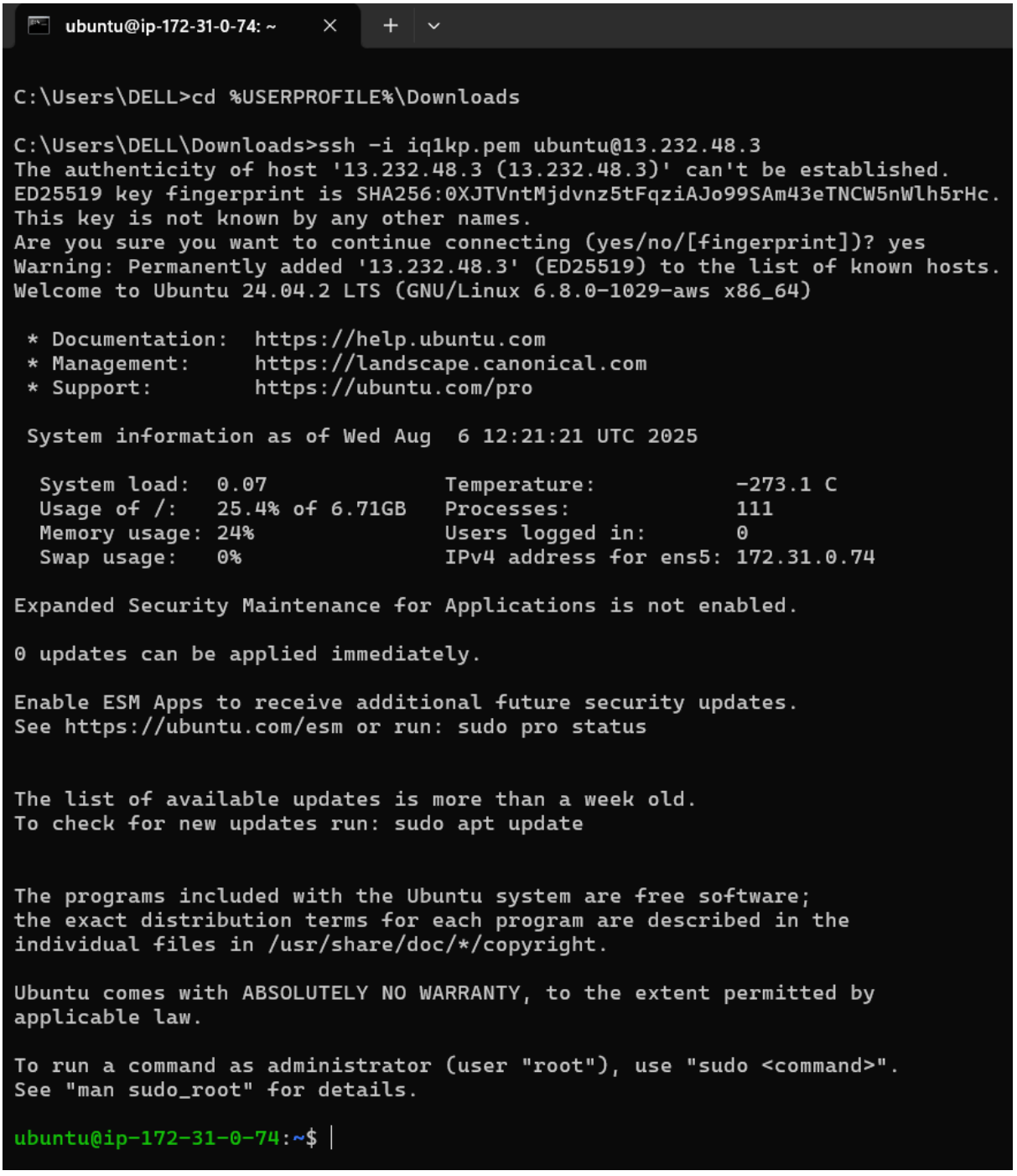
EC2 instance is launched successfully:



Successful.

2. **Connect via SSH with Command Prompt**

- In cmd write: ssh -i iq1kp.pem ubuntu@<New-EC2-Public-IP>
with the new Public IPv4 of the new EC2 ubuntu instance.



Inside the EC2 ubuntu terminal now.

- Update the packages (sudo apt update) (sudo apt upgrade -y)

```
ubuntu@ip-172-31-0-74:~$ sudo apt update
sudo apt upgrade -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1314 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [264 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [164 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1120 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [287 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [26.0 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1645 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [359 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [33.2 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [6772 B]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [39.9 kB]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main Translation-en [9152 B]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7068 B]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [272 B]
Get:31 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [28.0 kB]
Get:32 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.1 kB]
Get:33 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [28.4 kB]
Get:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1304 B]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:38 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1054 kB]
Get:40 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [183 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [878 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [194 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1560 kB]
```

3. **Install MySQL Server** (sudo apt install mysql-server -y)

```
ubuntu@ip-172-31-0-74: ~

ubuntu@ip-172-31-0-74:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64 libtimedate-perl liburi-perl mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64 libtimedate-perl liburi-perl mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 4 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1 [6746 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.42-0ubuntu0.24.04.2 [2728 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.42-0ubuntu0.24.04.2 [22.4 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libevent-pthreads-2.1-7t64 amd64 2.1.12-stable-9ubuntu2 [7982 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libmecab2 amd64 0.996-14ubuntu4 [201 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libprotobuf-lite32t64 amd64 3.21.12-8.2ubuntu0.2 [238 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server-core-8.0 amd64 8.0.42-0ubuntu0.24.04.2 [17.5 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server-8.0 amd64 8.0.42-0ubuntu0.24.04.2 [1438 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libhtml-tagset-perl all 3.20-6 [11.3 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liburi-perl all 5.27-1 [88.0 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libhtml-parser-perl amd64 3.81-1build3 [85.8 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libcgi-pm-perl all 4.63-1 [185 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libfcgi0t64 amd64 2.4.2-2.1ubuntu0.24.04.1 [27.0 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libfcgi-perl amd64 0.82+ds-3build2 [21.7 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libcgi-fast-perl all 1:2.17-1 [10.3 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libclone-perl amd64 0.46-1build3 [10.7 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libencode-locale-perl all 1.05-3 [11.6 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libfcgi-bin amd64 2.4.2-2.1ubuntu0.24.04.1 [11.2 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libhtml-template-perl all 2.97-2 [60.2 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libtimedate-perl all 2.3300-2 [34.0 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libhttp-date-perl all 6.06-1 [10.2 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libio-html-perl all 1.004-3 [15.9 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblwp-mediatypes-perl all 6.04-2 [20.1 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libhttp-message-perl all 6.45-1ubuntu1 [78.2 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mecab-utils amd64 0.996-14ubuntu4 [4804 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mecab-ipadic all 2.7.0-20070801+main-3 [6718 kB]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mecab-ipadic-utf8 all 2.7.0-20070801+main-3 [4384 B]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server all 8.0.42-0ubuntu0.24.04.2 [9518 B]
Fetched 29.6 MB in 1s (35.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package mysql-common.
(Reading database ... 102312 files and directories currently installed.)
Preparing to unpack .../0-mysql-common_5.8+1.1.0build1_all.deb ...
```

- Start MySQL (sudo systemctl start mysql)
- Enable on boot(sudo systemctl enable mysql)

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-0-74:~$ sudo systemctl start mysql
ubuntu@ip-172-31-0-74:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
ubuntu@ip-172-31-0-74:~$
```

- Enter the MySQL Shell (sudo mysql)

```
ubuntu@ip-172-31-0-74:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.42-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

We're now inside the MySQL CLI.

4. Performing basic SQL operations

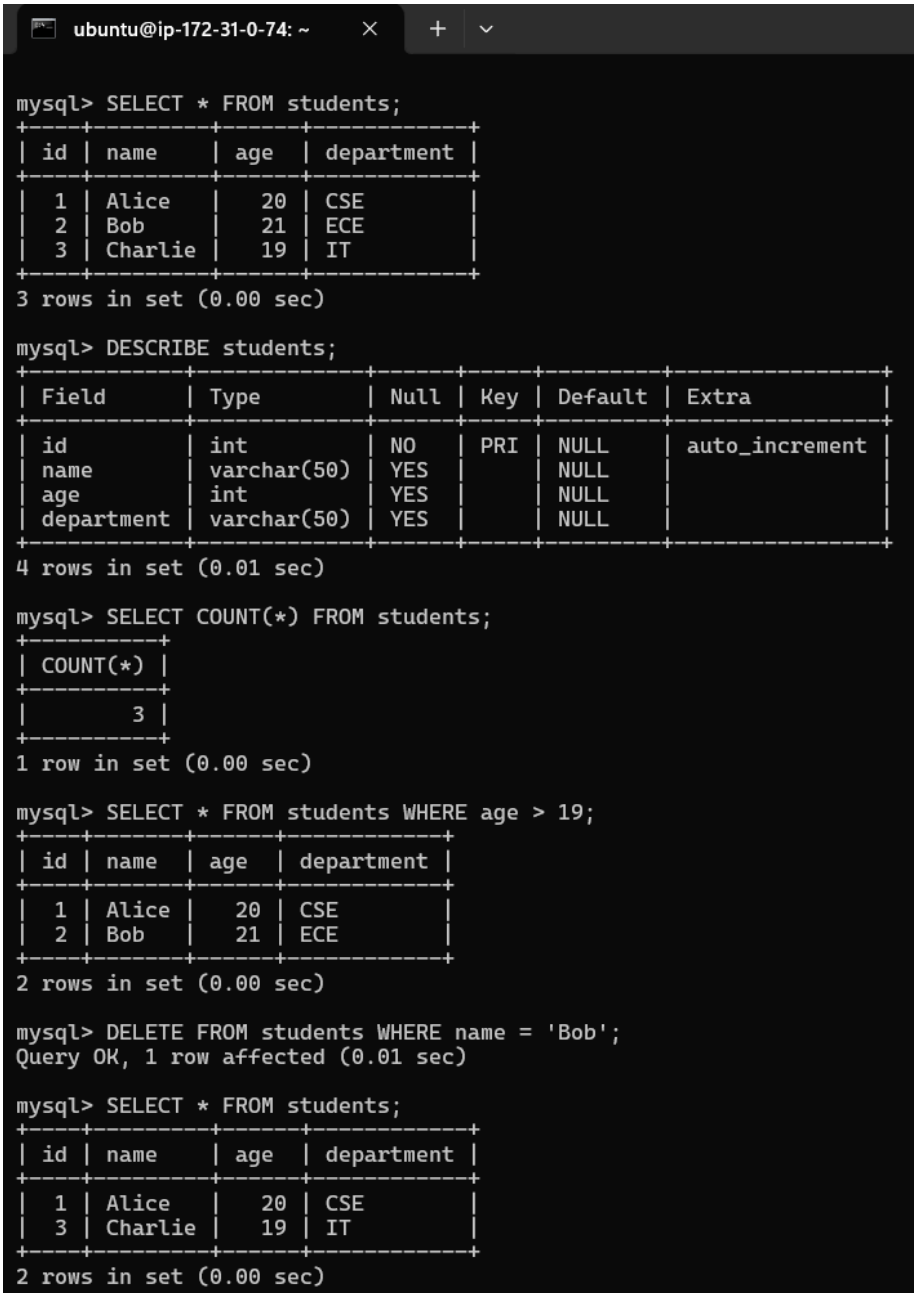
- Create Database & Table

```
mysql> CREATE DATABASE student;
 KEY,
    name VARCHAR(50),
    age INT,
    department VARCHAR(50)
);
Query OK, 1 row affected (0.04 sec)

mysql> USE student;
Database changed
mysql>
mysql> CREATE TABLE students (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     age INT,
    ->     department VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql>
```

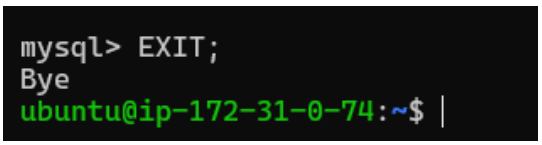
- Insert Records in table

```
mysql> INSERT INTO students (name, age, department)
    -> VALUES
    -> ('Alice', 20, 'CSE'),
    -> ('Bob', 21, 'ECE'),
    -> ('Charlie', 19, 'IT');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
```

- Run basic queries for selection, deletion and more:



- Exit MySQL & go back to the Ubuntu EC2 shell.



We've successfully run MySQL queries in an EC2 instance!

Q2. Set up an S3 bucket in AWS, ensuring the name is globally unique. Upload your static website files (HTML, CSS, JavaScript, etc.) to this bucket. Enable static website hosting in the S3 bucket properties and configure the documents. Apply the necessary bucket policy to make your site publicly accessible. Also, enable versioning on the bucket to maintain previous versions of your website files. (5.0 marks)

Approach:

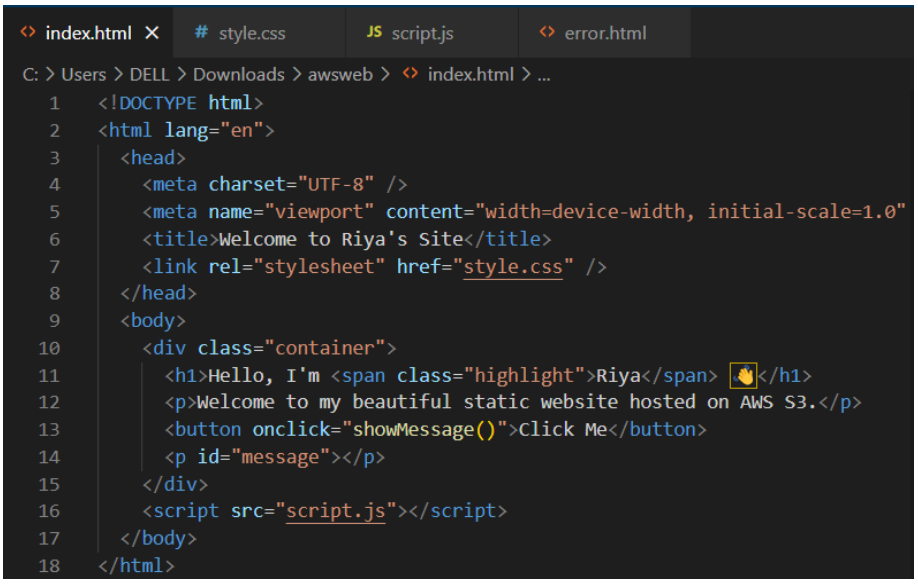
1. Create an S3 bucket
2. Upload your static site files (HTML, CSS, JS)
3. Enable static website hosting
4. Set an index and error page
5. Make the site publicly accessible
6. Enable versioning on the bucket

STEPS:

1. **Create the HTML, CSS, JS files of the application and save them in one Folder.**

Folder Name: awsweb

Index.html



Script.js

Style.css

index.html | style.css | script.js | error.html

C: > Users > DELL > Downloads > awsweb > style.css > ...

```css
body {
  margin: 0;
  padding: 0;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background: linear-gradient(135deg, #fceabb, #f8b500);
  color: #333;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  text-align: center;
  background: white;
  padding: 40px;
  border-radius: 20px;
  box-shadow: 0 10px 25px rgba(0,0,0,0.1);
}

.highlight {
  color: #f39c12;
  font-weight: bold;
}

button {
  margin-top: 20px;
  padding: 10px 20px;
  font-size: 16px;
  border: none;
  background-color: #f39c12;
  color: white;
  border-radius: 5px;
  cursor: pointer;
}
```

Error.html

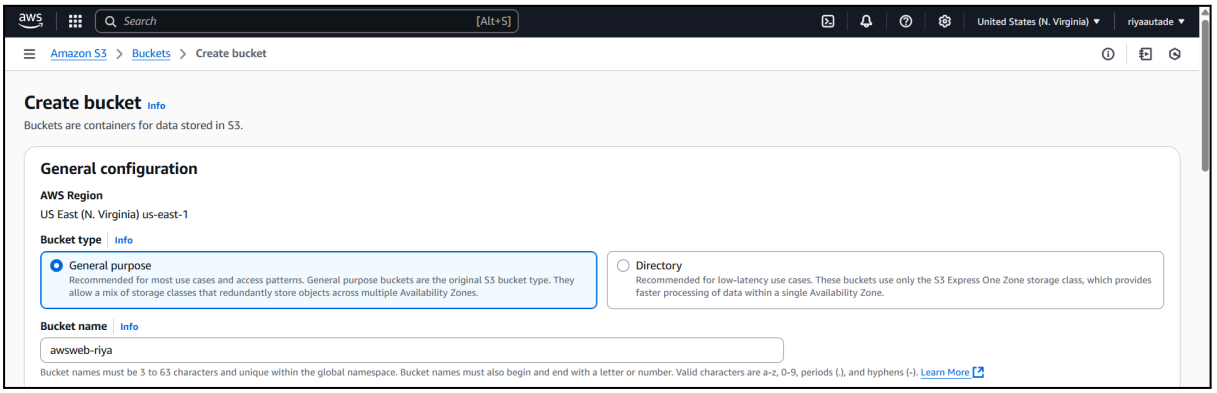
index.html | style.css | script.js | error.html

C: > Users > DELL > Downloads > awsweb > error.html > ...

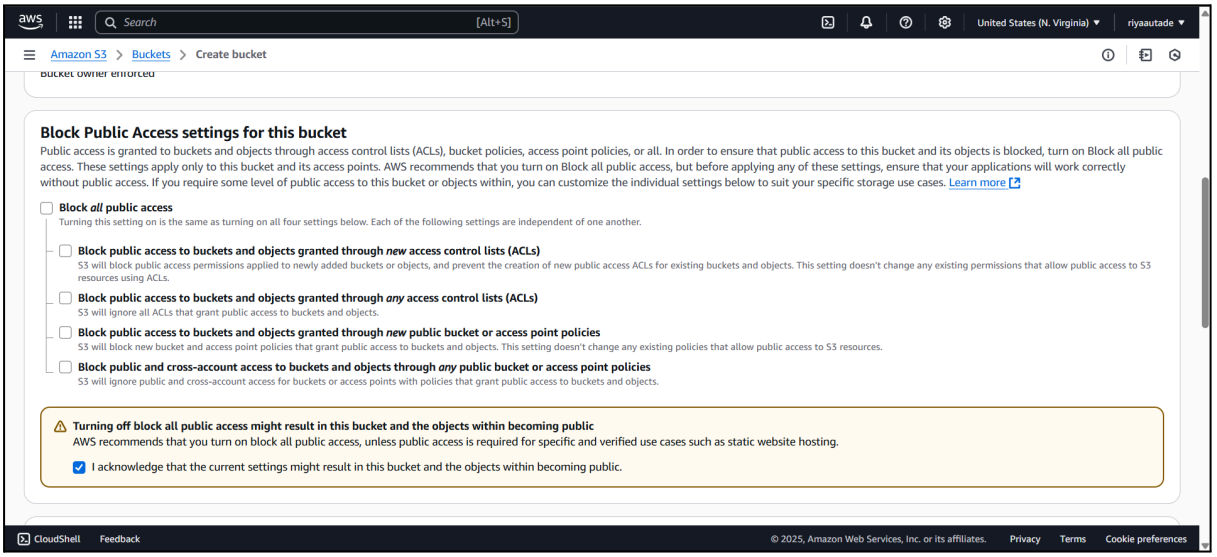
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Oops! Page Not Found</title>
    <style>
      body {
        background-color: #fef6f6;
        font-family: Arial, sans-serif;
        text-align: center;
        padding: 100px;
      }
      h1 {
        font-size: 50px;
        color: #c0392b;
      }
      p {
        font-size: 20px;
      }
    </style>
  </head>
  <body>
    <h1>404 - Page Not Found</h1>
    <p>Sorry, couldn't find the page you're looking for.</p>
  </body>
</html>
```

2. Create a unique S3 Bucket

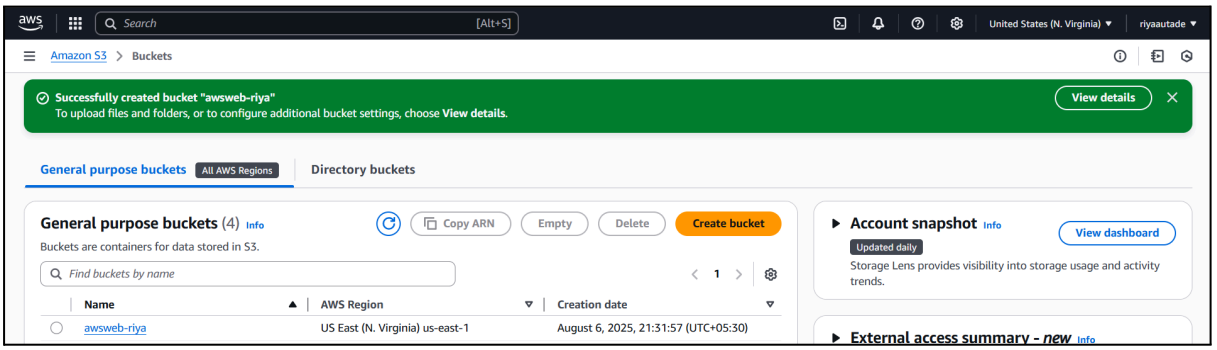
- Go to the S3 Console and click "Create bucket"
- Give it a globally unique Bucket name (awsweb-riya)



- Scroll to Block Public Access settings
- Uncheck “Block all public access” & Check the acknowledgment checkbox
- Leave all other settings as default & Click Create bucket

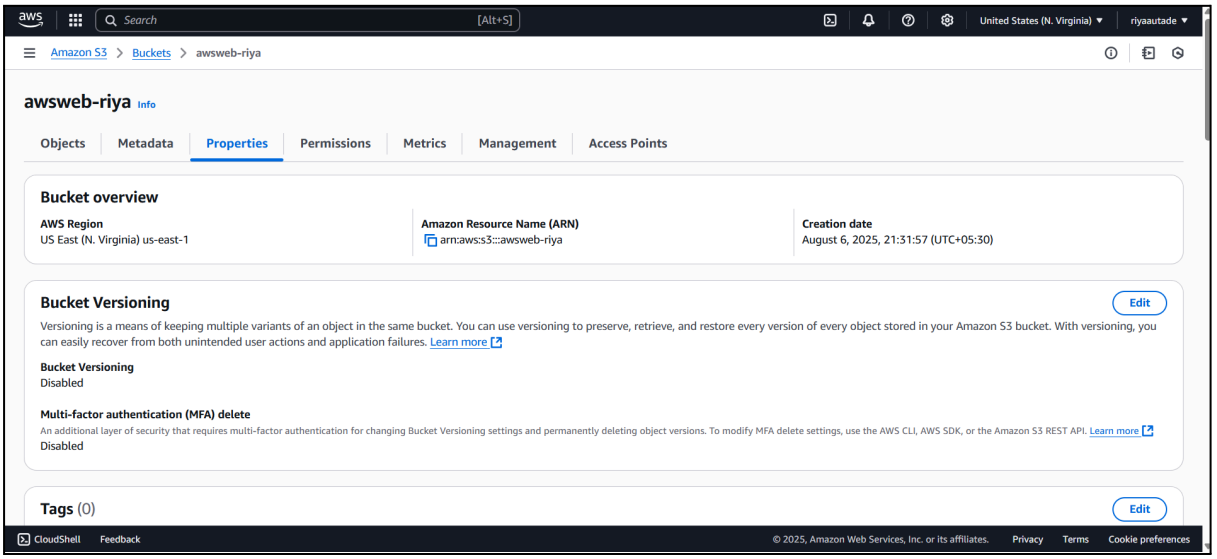


S3 Bucket is created successfully:

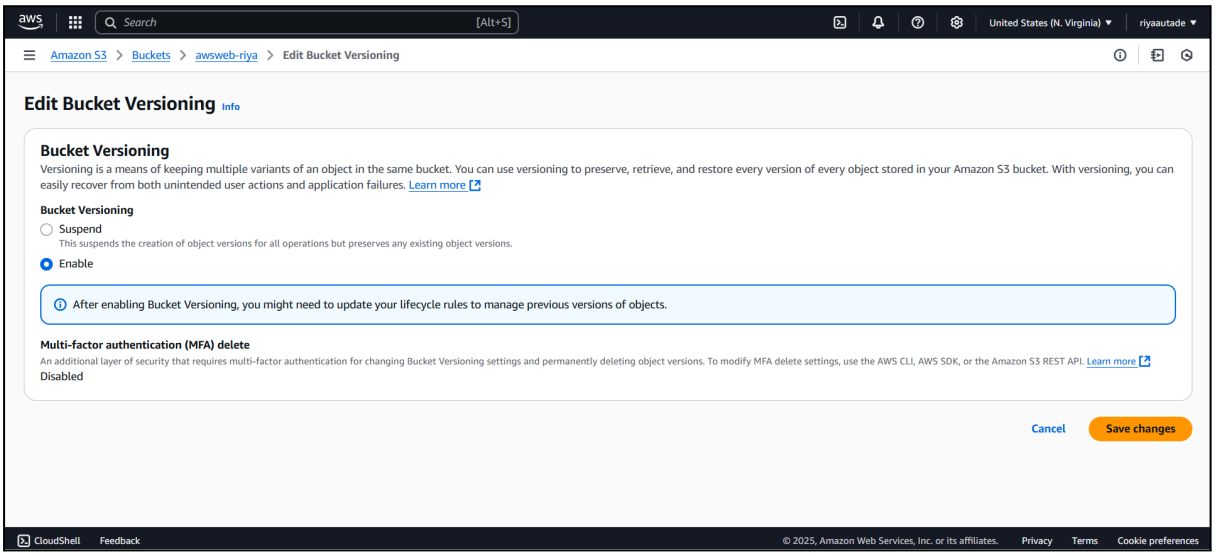


3. Enable Versioning for S3 Bucket

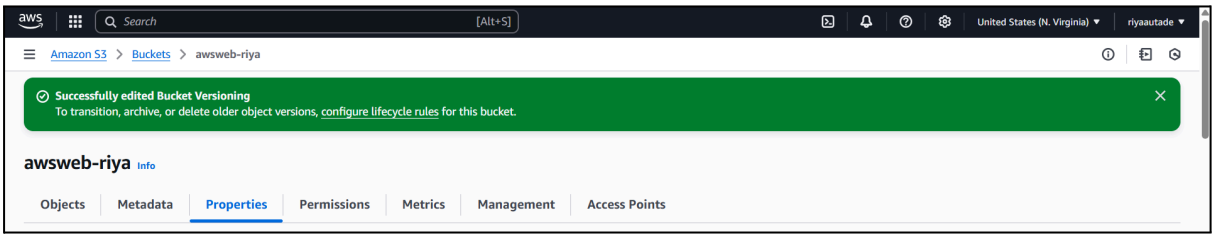
- After bucket is created, go to the Properties tab & Scroll down to Bucket Versioning
- Click Edit



- Choose Enable under Bucket Versioning
- Click Save changes

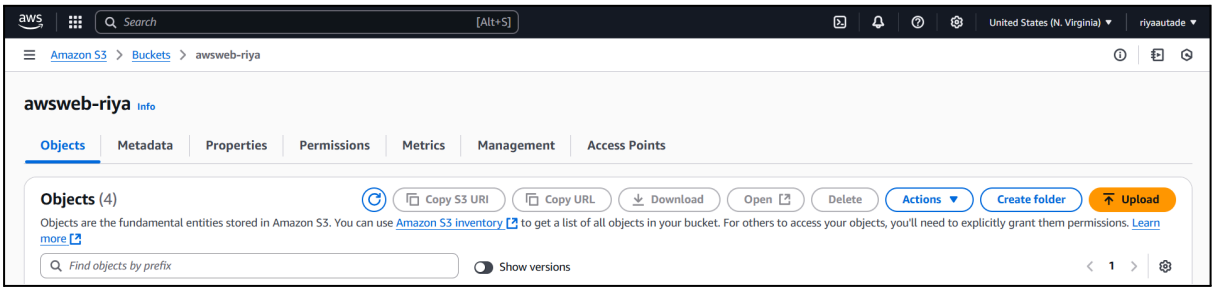


Bucket Versioning is successfully enabled:

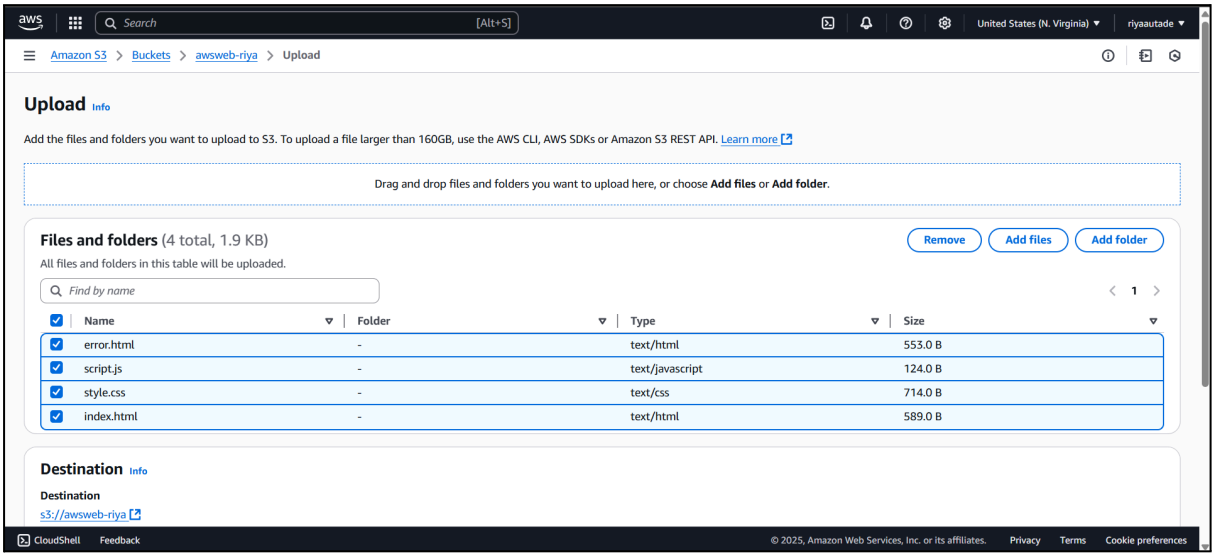


4. Upload Your Website Files

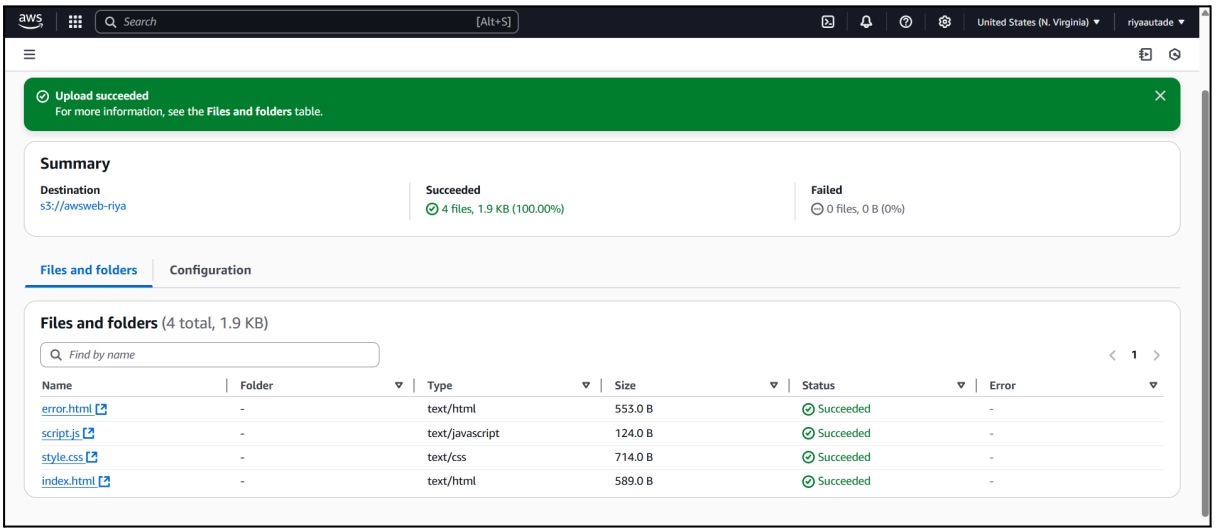
- Go to the Objects tab of the new S3 bucket
- Click Upload



- Upload all 4 files: index.html, style.css, script.js, error.html
- Click Upload

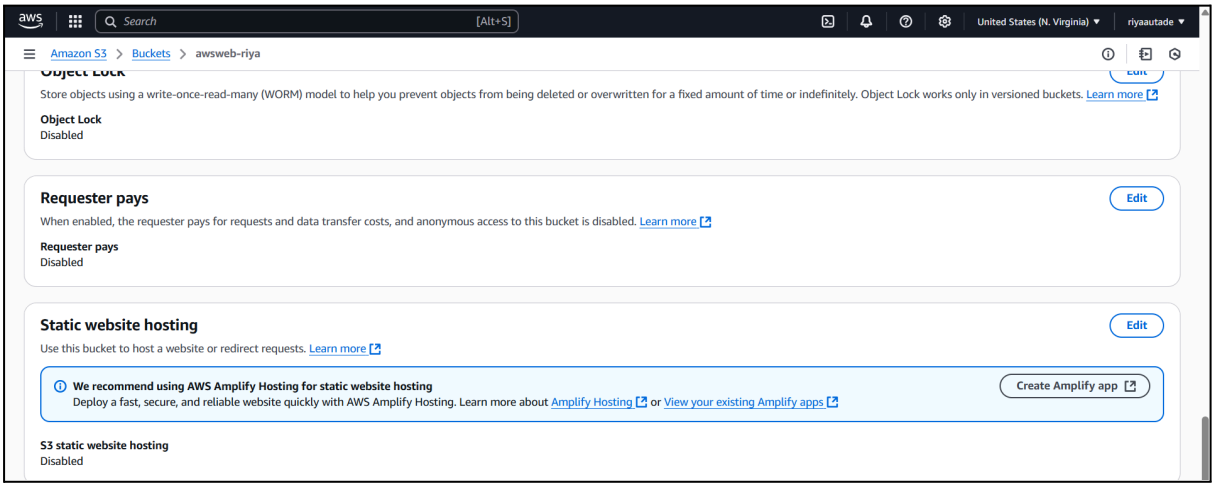


Website files are successfully uploaded in S3:

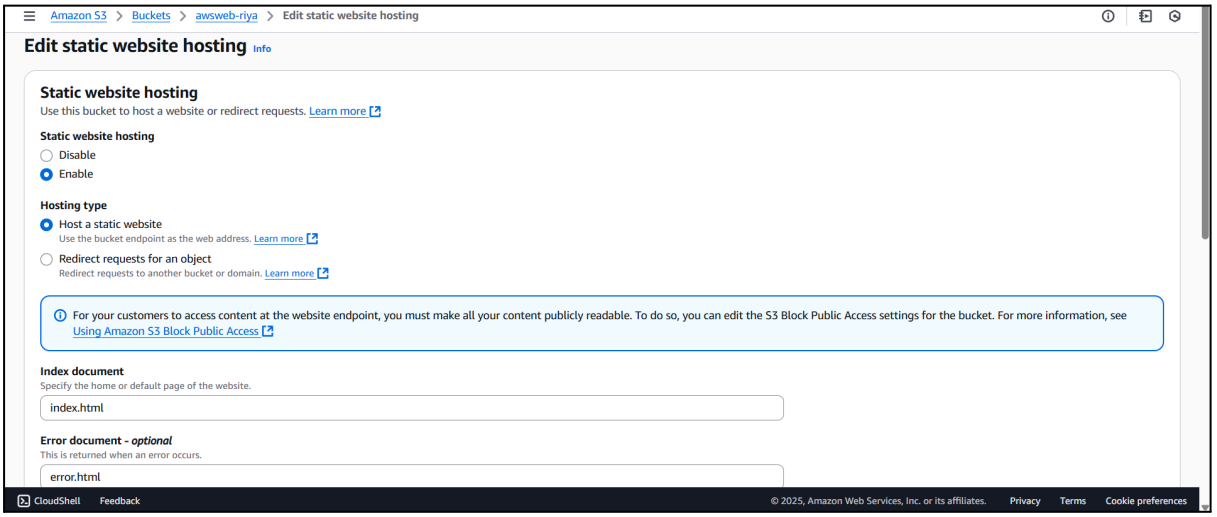


5. Enable Static Website Hosting

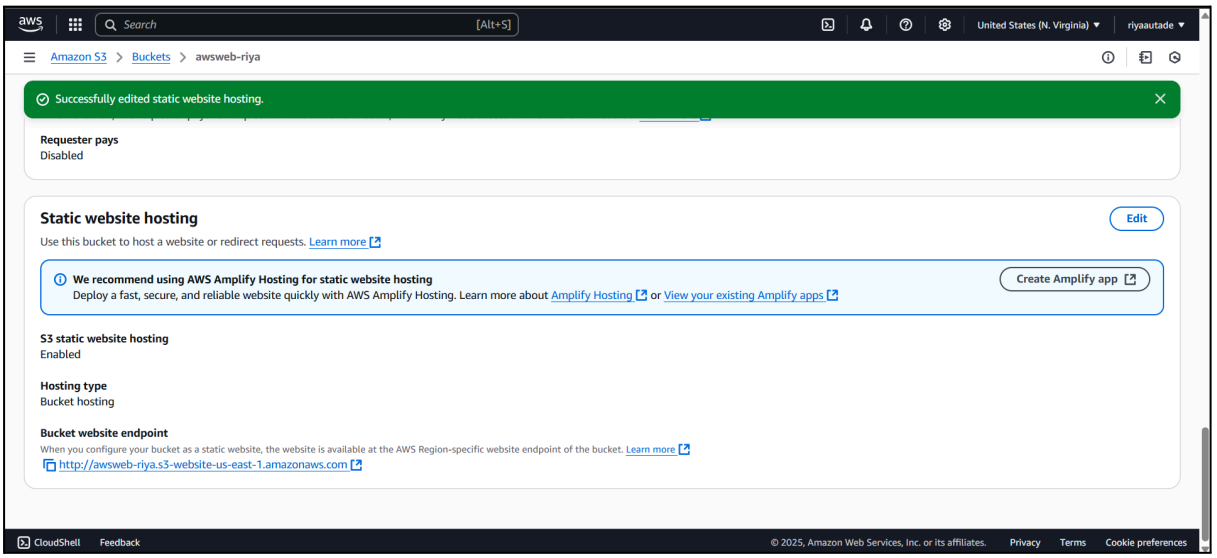
- Go to Properties tab of S3 bucket & scroll down to Static website hosting & Click Edit



- Click on Enable & in Index & Error document put the names of the respective files



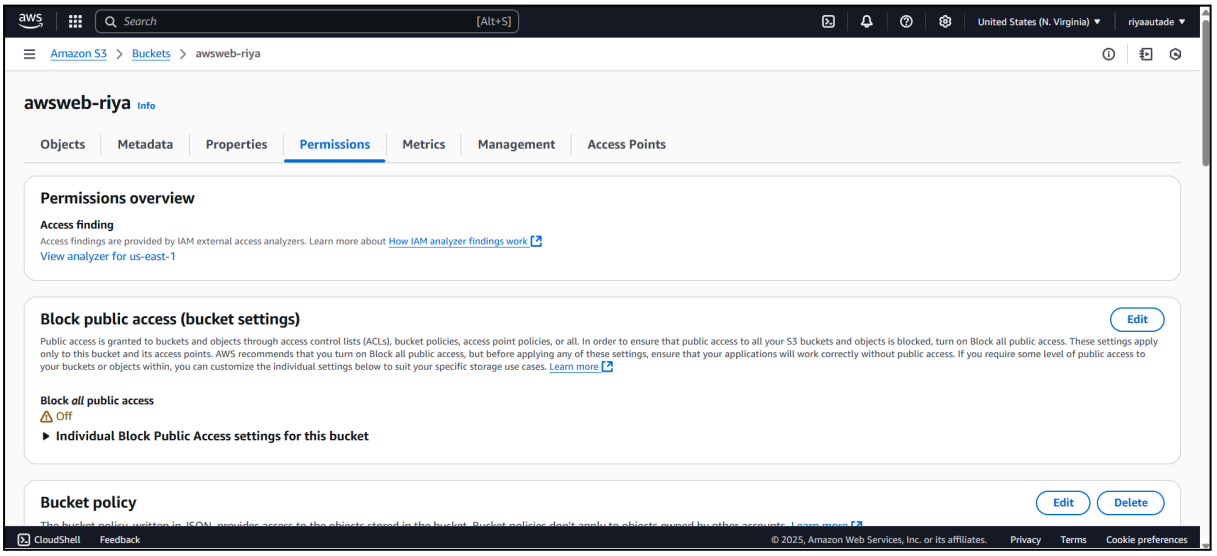
- Click Save changes. Static website hosting is successfully enabled.



Copy the Website Endpoint URL shown here — this is the live site!

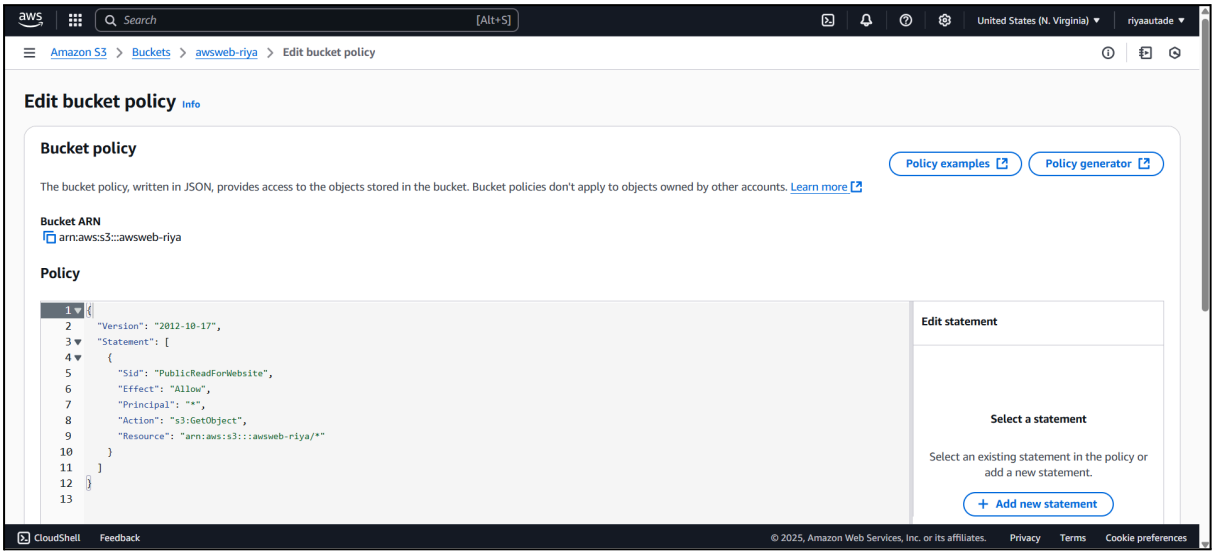
6. Add Bucket Policy (Make Site Public)

- Go to Permissions tab, Scroll down to Bucket policy & Click Edit

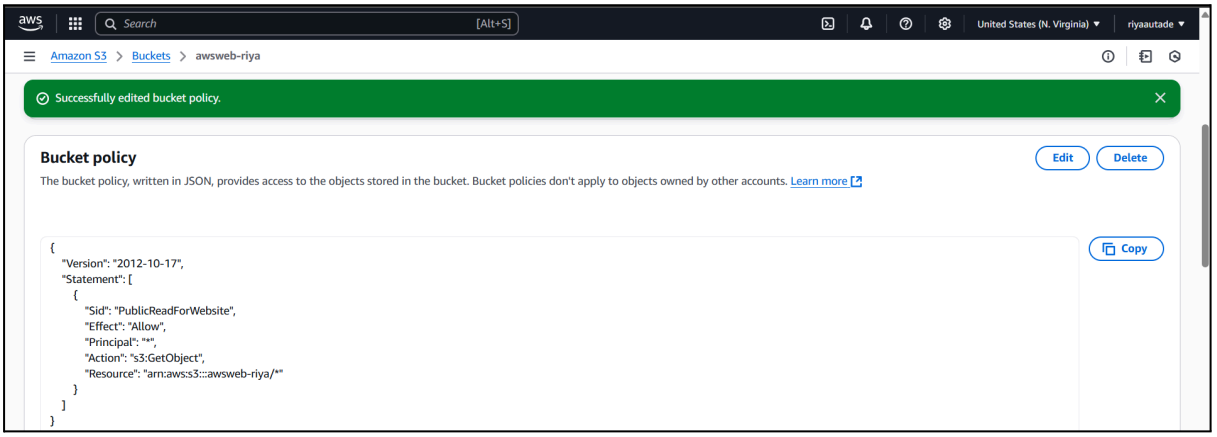


- Paste this policy below with the correct Bucket ARN in the editor:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadForWebsite",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::awsweb-riya/*"
    }
  ]
}
```

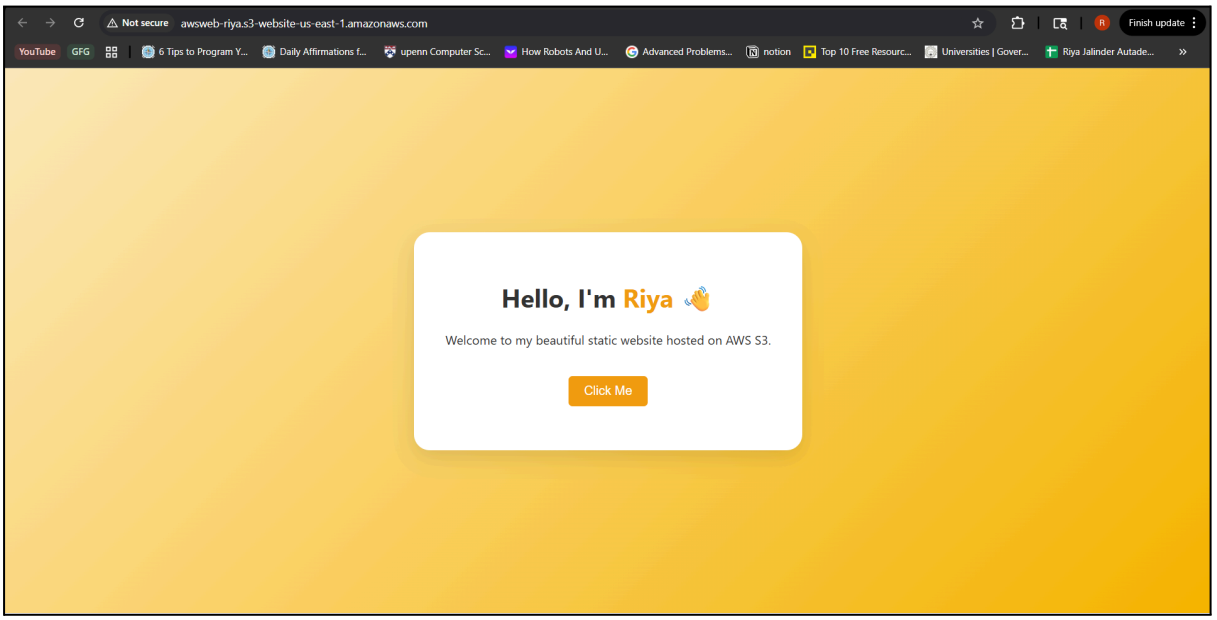


Bucket Policy is successfully edited:



7. Test the Website

- Go back to Properties → Static website hosting
- Copy the Endpoint URL: http://awsweb-riya.s3-website-us-east-1.amazonaws.com
- Paste it into the browser



The website is successfully hosted and publicly accessible now!
