



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)

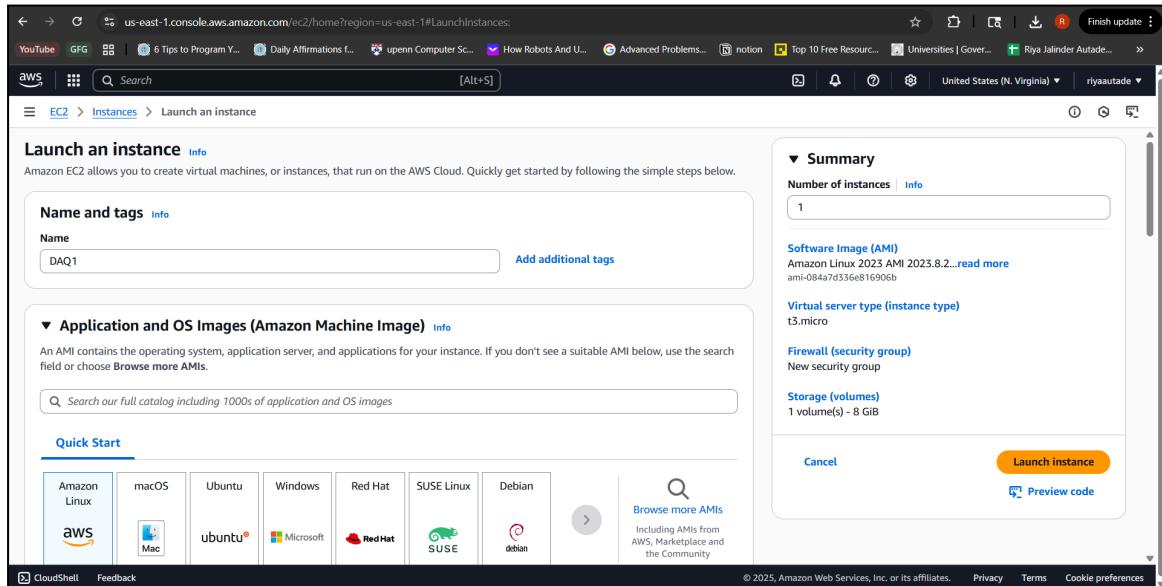
Part A: EC2 Instance & Application Migration

Approach:

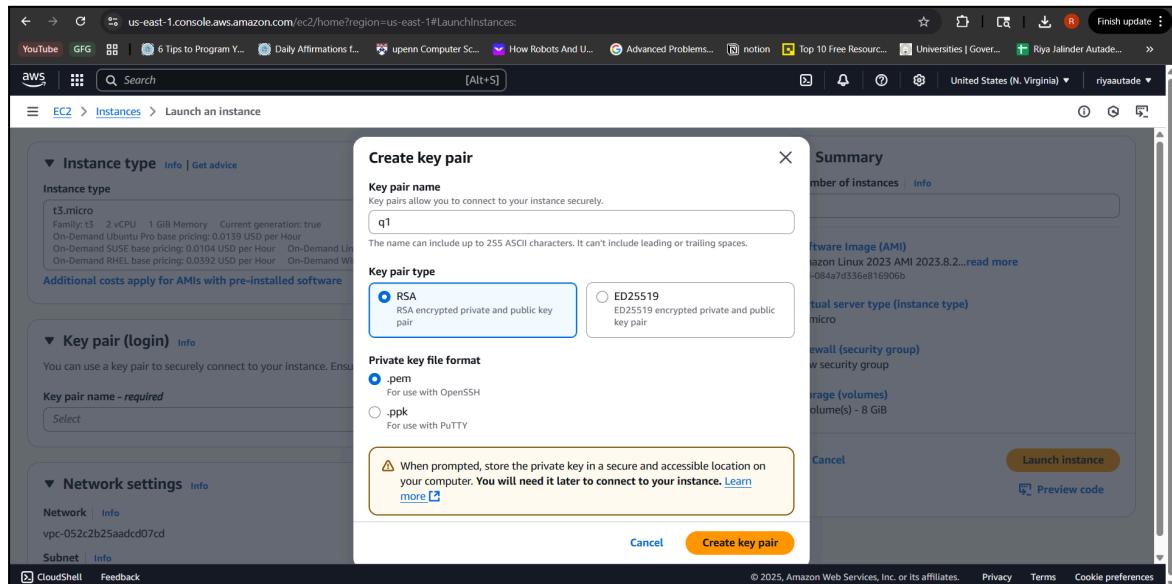
1. Create an EC2 instance.
 - Launch a virtual server (EC2) on AWS.
2. Launch an application on the EC2 instance in one region.
 - Deploy a simple web app (Python Flask app).
3. Migrate that instance to another region.
 - Move this EC2 instance (with the application) from one AWS region (us-east-1) to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

STEPS:

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

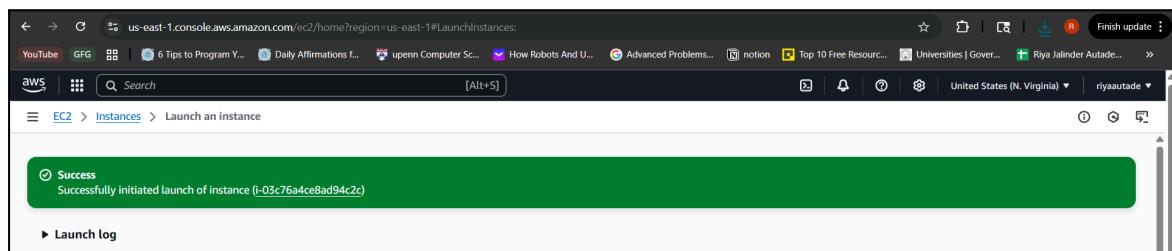


- Create a key pair for SSH (q1.pem) with Network Settings: HTTP, SSH, HTTPS



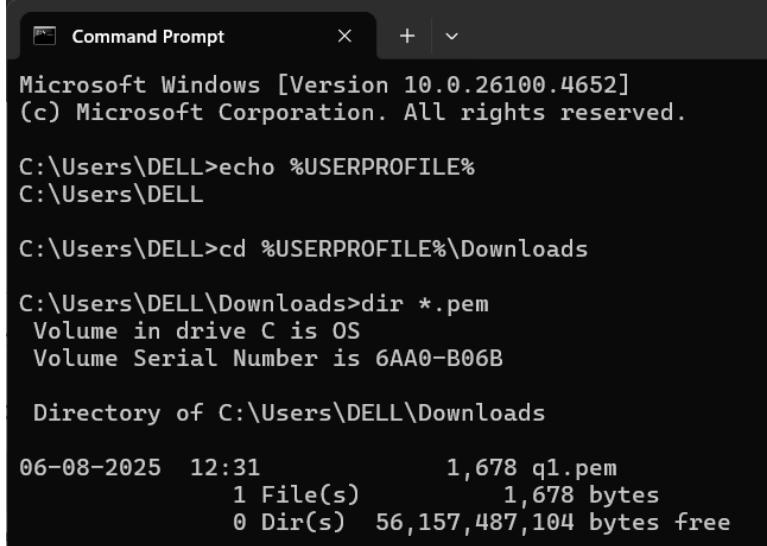
- Keep the rest default and click on Launch Instance.

EC2 instance (DAQ1) launched successfully



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

- Navigate to folder where your .pem file is:



```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>echo %USERPROFILE%
C:\Users\DELL

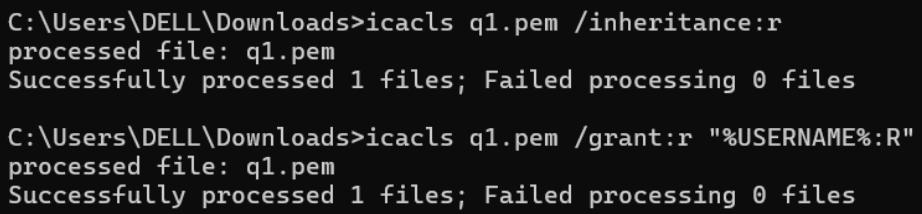
C:\Users\DELL>cd %USERPROFILE%\Downloads

C:\Users\DELL\Downloads>dir *.pem
Volume in drive C is OS
Volume Serial Number is 6AA0-B06B

Directory of C:\Users\DELL\Downloads

06-08-2025  12:31           1,678 q1.pem
               1 File(s)      1,678 bytes
               0 Dir(s)  56,157,487,104 bytes free
```

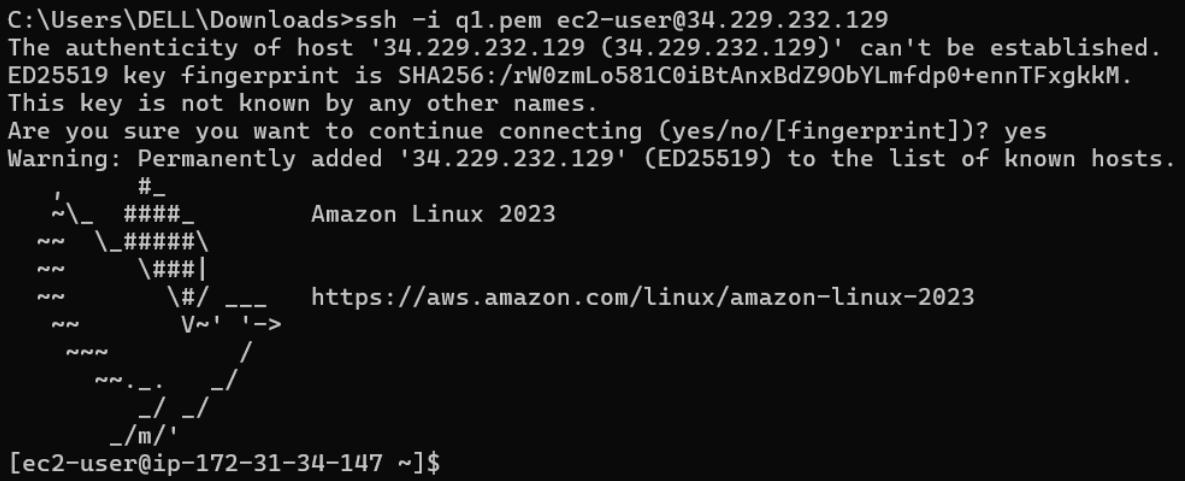
- Change file permissions:



```
C:\Users\DELL\Downloads>icacls q1.pem /inheritance:r
processed file: q1.pem
Successfully processed 1 files; Failed processing 0 files

C:\Users\DELL\Downloads>icacls q1.pem /grant:r "%USERNAME%:R"
processed file: q1.pem
Successfully processed 1 files; Failed processing 0 files
```

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



```
C:\Users\DELL\Downloads>ssh -i q1.pem ec2-user@34.229.232.129
The authenticity of host '34.229.232.129 (34.229.232.129)' can't be established.
ED25519 key fingerprint is SHA256:/rW0zmLo581C0iBtAnxBdZ90bYLmfdp0+ennTFxgkkM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.229.232.129' (ED25519) to the list of known hosts.

' _#
~\_ #####_      Amazon Linux 2023
~~ \_#####\
~~ \###|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-->
~~~ /
~~..-/-
~/m/'

[ec2-user@ip-172-31-34-147 ~]$
```

We are in the EC2 terminal now.

3. Flask App + Deployment on EC2

- Install Python on the EC2 instance & check version:

```
[ec2-user@ip-172-31-34-147 ~]$ sudo yum update -y
sudo yum install python3 -y
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Wed Aug  6 07:19:01 2025.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:02 ago on Wed Aug  6 07:19:01 2025.
Package python3-3.9.23-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-34-147 ~]$ python3 --version
Python 3.9.23
```

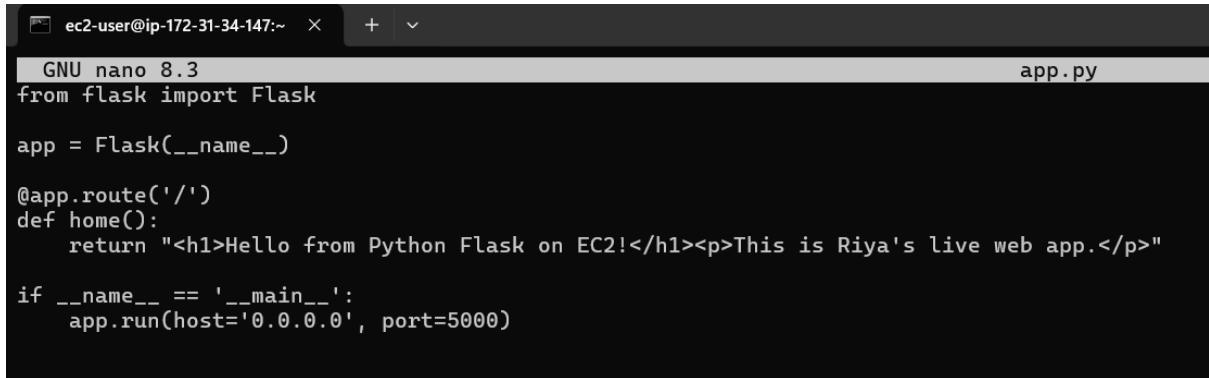
- Install Flask on the EC2 instance

```
[ec2-user@ip-172-31-34-147 ~]$ python3 -m ensurepip --upgrade
pip3 install Flask
Defaulting to user installation because normal site-packages is not writeable
Looking in links: /tmp/tmpa9mq808m
Requirement already satisfied: setuptools in /usr/lib/python3.9/site-packages (59.6.0)
Processing /tmp/tmpa9mq808m/pip-21.3.1-py3-none-any.whl
Installing collected packages: pip
Successfully installed pip-21.3.1
Defaulting to user installation because normal site-packages is not writeable
Collecting Flask
  Downloading Flask-3.1.1-py3-none-any.whl (103 kB)
    |██████████| 103 kB 16.3 MB/s
Collecting blinker>=1.9.0
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting itsdangerous>=2.2.0
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting jinja2>=3.1.2
  Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
    |██████████| 134 kB 41.4 MB/s
Collecting click>=8.1.3
  Downloading click-8.1.8-py3-none-any.whl (98 kB)
    |██████████| 98 kB 14.7 MB/s
Collecting werkzeug>=3.1.0
  Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
    |██████████| 224 kB 122.5 MB/s
Collecting markupsafe>=2.1.1
  Downloading MarkupSafe-3.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
Collecting importlib-metadata>=3.6.0
  Downloading importlib_metadata-8.7.0-py3-none-any.whl (27 kB)
Collecting zip>=3.20
  Downloading zipp-3.23.0-py3-none-any.whl (10 kB)
Installing collected packages: zip, markupsafe, werkzeug, jinja2, itsdangerous, importlib-metadata, click, blinker, Flask
Successfully installed Flask-3.1.1 blinker-1.9.0 click-8.1.8 importlib-metadata-8.7.0 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.2 werkzeug-3.1.3 zipp-3.23.0
```

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



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



```

GNU nano 8.3
app.py

from flask import Flask

app = Flask(__name__)

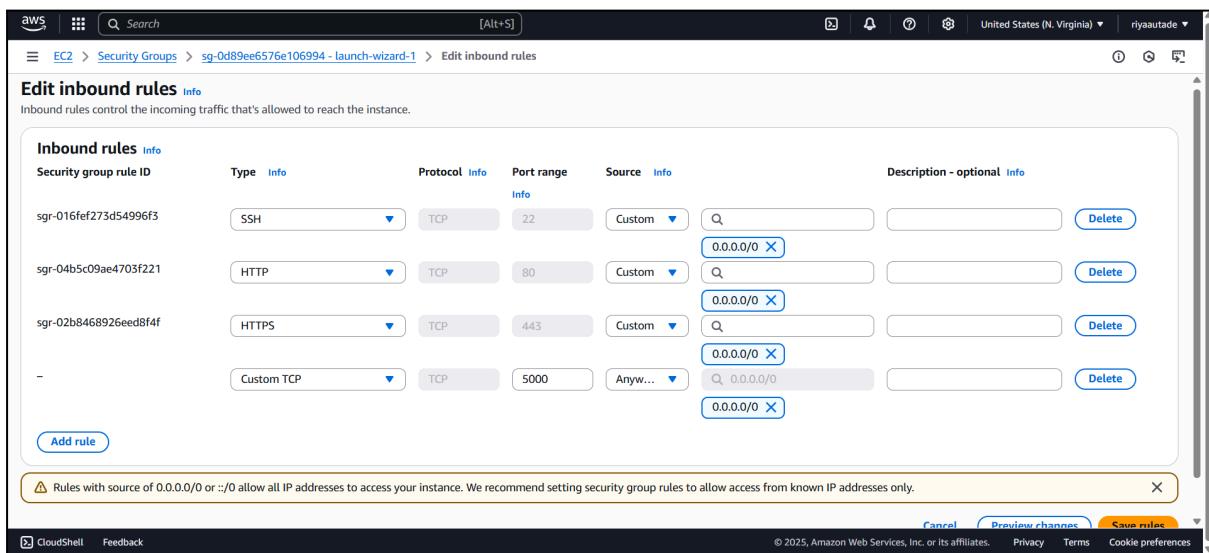
@app.route('/')
def home():
    return "<h1>Hello from Python Flask on EC2!</h1><p>This is Riya's live web app.</p>"

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



Edit inbound rules Info

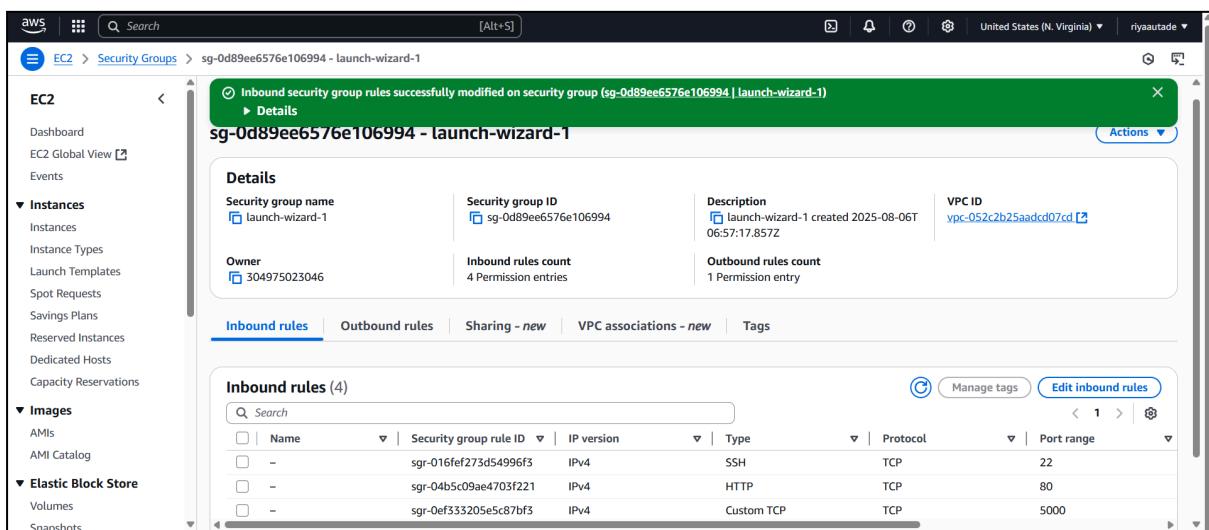
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules	Type	Protocol	Port range	Source	Description - optional
sgr-016fef273d54996f3	SSH	TCP	22	Custom	0.0.0.0/0
sgr-04b5c09ae4703f221	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-02b8468926eed8f4f	HTTPS	TCP	443	Custom	0.0.0.0/0
-	Custom TCP	TCP	5000	Anyw...	0.0.0.0/0

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- Click on Save rules and add it successfully



Inbound security group rules successfully modified on security group (sg-0d89ee6576e106994 | launch-wizard-1)

sg-0d89ee6576e106994 - launch-wizard-1

Details

Security group name	sg-0d89ee6576e106994	Description	launch-wizard-1 created 2025-08-06T06:57:17.857Z
Owner	304975023046	Inbound rules count	4 Permission entries
		Outbound rules count	1 Permission entry

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-016fef273d54996f3	IPv4	SSH	TCP	22
-	sgr-04b5c09ae4703f221	IPv4	HTTP	TCP	80
-	sgr-0ef333205e5c87bf3	IPv4	Custom TCP	TCP	5000

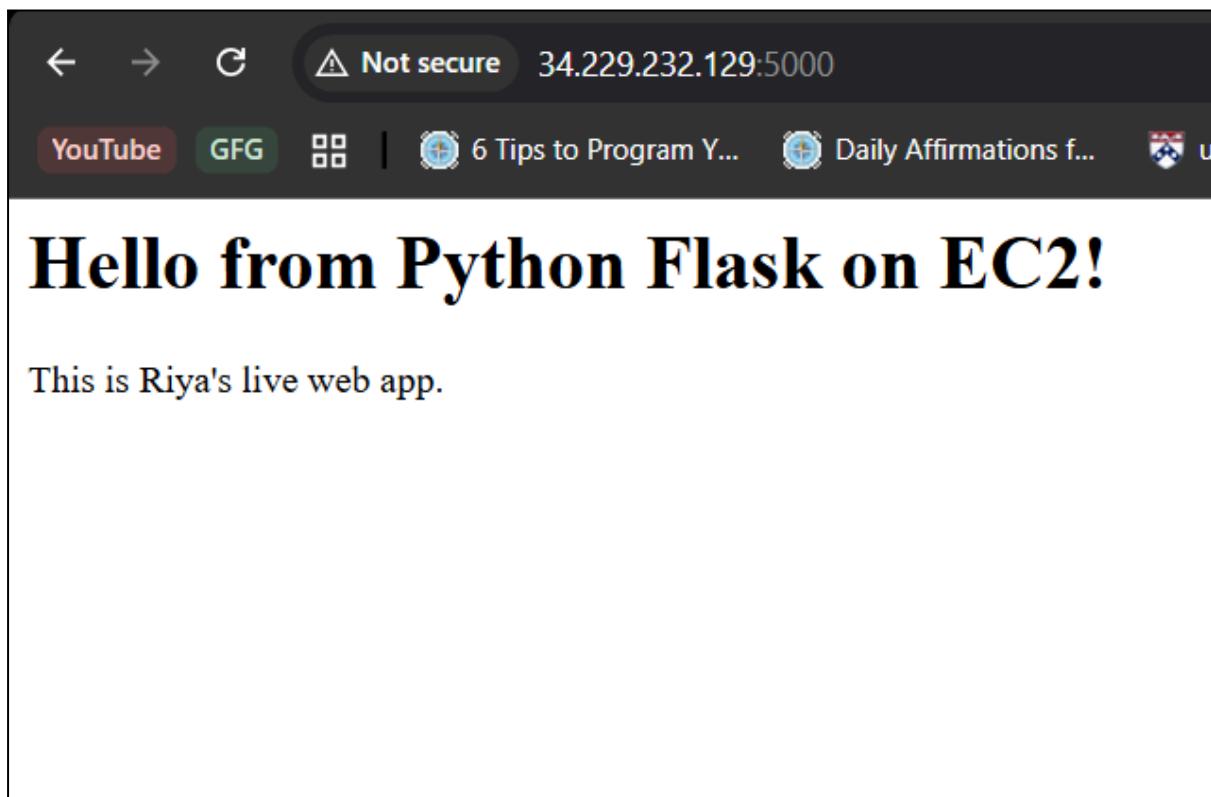
- Run the Flask App

```
[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

The screenshot shows the AWS EC2 Instances page for an instance named 'i-03c76a4ce8ad94c2c (DAQ1)'. The 'Actions' menu is open, and the 'Image and templates' option is selected. The 'Create image' button is highlighted. Other options like 'Create template from instance' and 'Launch more like this' are also visible.

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

The screenshot shows the 'Create image' wizard. In the 'Image details' step, the 'Instance ID' is set to 'i-03c76a4ce8ad94c2c (DAQ1)' and the 'Image name' is 'IQ1'. A note at the bottom states: 'Maximum 127 characters. Can't be modified after creation.'

- Wait a few minutes & AMI is created successfully

The screenshot shows the AWS EC2 Instances page. A green notification bar at the top says: 'Currently creating AMI ami-0d8e534954541ad4b from instance i-03c76a4ce8ad94c2c. Check that the AMI status is "Available" before deleting the instance or carrying out other actions related to this AMI.' Below this, the 'Instances (1/1)' table shows a single instance named 'DAQ1' with the AMI ID 'ami-0d8e534954541ad4b'. At the bottom, the instance details for 'i-03c76a4ce8ad94c2c (DAQ1)' are shown, including its public IP address (34.229.232.129), private IP address (172.31.34.147), and state (Running).

Find the AMI in EC2

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

The screenshot shows the AWS EC2 console with the 'AMIs' section selected under 'Images'. A single AMI named 'IQ1' is listed in the table. The table columns include Name, AMI name, AMI ID, Source, Owner, and Visibility. The AMI ID is 'ami-0d8e534954541ad4b', the Source is '304975023046/IQ1', the Owner is '304975023046', and the Visibility is 'Private'.

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

The screenshot shows the AWS EC2 console with the 'AMIs' section selected under 'Images'. The table now shows the AMI 'IQ1' with a status of 'Available'. The 'Actions' dropdown menu is open, showing options like 'Copy AMI', 'Edit AMI permissions', 'Request Spot Instances', 'Manage tags', and 'Deregister AMI'. The 'Copy AMI' option is highlighted.

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

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

The screenshot shows the 'Copy AMI' configuration page. The 'Original AMI ID' is set to 'ami-0d8e534954541ad4b'. The 'AMI copy name' is 'IQ1 copy'. The 'AMI copy description' field contains '[Copied ami-0d8e534954541ad4b from us-east-1] IQ1'. The 'Destination Region' dropdown is set to 'Asia Pacific (Mumbai)'. The 'Copy tags' checkbox is unchecked.

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

The screenshot shows the AWS EC2 AMIs page. A blue banner at the top indicates an 'AMI copy operation' has been initiated for AMI ID 'ami-0d8e534954541ad4b'. Below this, a table lists the copied AMI 'IQ1' with details: AMI ID 'ami-0d8e534954541ad4b', Owner account ID '304975023046', Source '304975023046/IQ1', and other metadata like Platform details (Linux/UNIX), Root device type (EBS), and Usage operation (RunInstances). The left sidebar shows navigation options for EC2, Instances, Images, and Elastic Block Store.

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

The screenshot shows the AWS EC2 AMIs page in the 'Asia Pacific (Mumbai)' region. The top right corner dropdown has been changed to the target region, ap-south-1. The rest of the interface is identical to the previous screenshot.

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)

The screenshot shows the 'Create key pair' dialog box overlaid on the EC2 Instances launch screen. In the 'Key pair name' field, 'iq1kp' is entered. The 'Key pair type' section shows 'RSA' selected. The 'Private key file format' section shows '.pem' 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.' A 'Create key pair' button is at the bottom right.

AMI Copy EC2 instance is successfully launched:

The screenshot shows the EC2 Instances launch screen with a green success message at the bottom: 'Success Successfully initiated launch of instance (i-0bcb41c16d5424484)'. The top right corner dropdown still shows the target region, ap-south-1.

- 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!

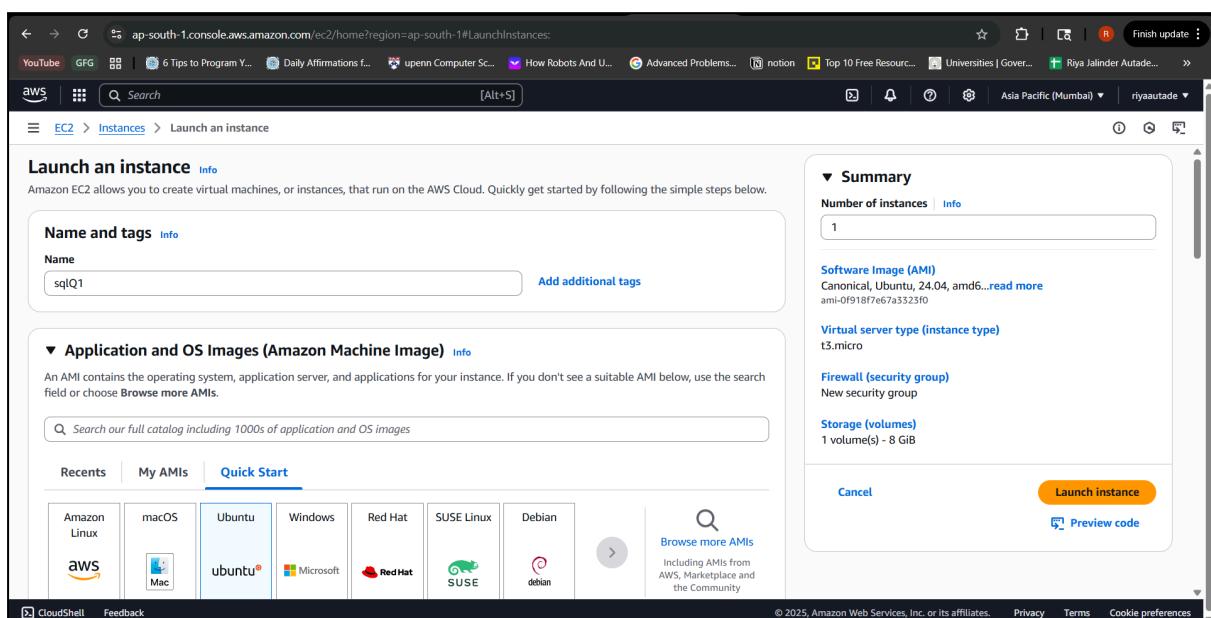
Part B: EC2 + MySQL Setup

Approach:

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

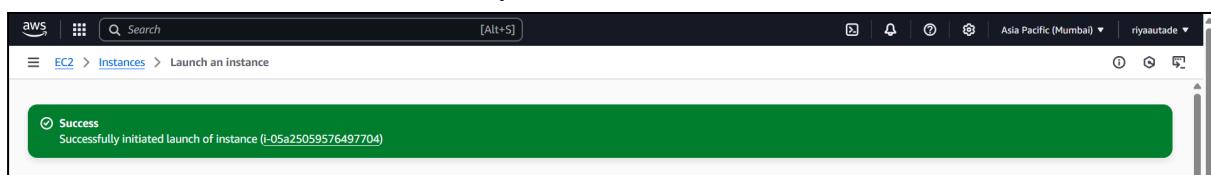
STEPS:

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.

```
ubuntu@ip-172-31-0-74: ~      X + ^

C:\Users\DELL>cd %USERPROFILE%\Downloads

C:\Users\DELL\Downloads>ssh -i iq1kp.pem ubuntu@13.232.48.3
The authenticity of host '13.232.48.3 (13.232.48.3)' can't be established.
ED25519 key fingerprint is SHA256:0XJTVntMjdvnz5tFqziAJo99SAM43eTNCW5nWlh5rHc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.232.48.3' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

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

System information as of Wed Aug  6 12:21:21 UTC 2025

System load: 0.07          Temperature:           -273.1 C
Usage of /:   25.4% of 6.71GB Processes:            111
Memory usage: 24%          Users logged in:        0
Swap usage:   0%           IPv4 address for ens5: 172.31.0.74

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 list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-31-0-74:~$ |
```

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 kB]
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 kB]
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 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 kB]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [592 kB]
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 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7668 kB]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [272 kB]
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 kB]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 kB]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 kB]
Get:38 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 kB]
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:~$ 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:
  libbcgi-perl libbcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-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-common mysql-server-8.0 mysql-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:
  libbcgi-fast-perl libbcgi-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-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 kB]
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/noinc/main amd64 libevent-pthreads-2.1-7t64 amd64 2.1.12-stable-0ubuntu2 [7982 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noinc/main amd64 libmecab2 amd64 0.996-14ubuntu0 [281 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 kB]
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 libltdl-tagset-perl all 3.20-6 [11.3 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl tagset-perl all 5.27-1 [88.0 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl-tagset-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 2.1.2-2-lubuntu0.24.04.1 [27.0 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 libfcgi-perl amd64 2.1.2-2-lubuntu0.24.04.1 [27.0 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 libfcgi0t64 amd64 2.1.2-2-lubuntu0.24.04.1 [1.3 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libclone-perl amd64 0.46-1build3 [18.7 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-lubuntu0.24.04.1 [11.2 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libhtml-template-perl all 2.97-2 [68.2 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl-tagset-perl all 2.3300-2 [34.0 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl-date-perl all 6.06-1 [18.2 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libhttp-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 [10.1 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 [4884 kB]
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 kB]
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 kB]
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-Ubuntu0.24.04.2 (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:

```
ubuntu@ip-172-31-0-74: ~      X + ▾

mysql> SELECT * FROM students;
+---+-----+-----+-----+
| id | name  | age   | department |
+---+-----+-----+-----+
| 1  | Alice | 20    | CSE      |
| 2  | Bob   | 21    | ECE      |
| 3  | Charlie | 19    | IT       |
+---+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESCRIBE students;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int   | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50) | YES  |     | NULL    |                |
| age   | int   | YES  |     | NULL    |                |
| department | varchar(50) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> SELECT COUNT(*) FROM students;
+-----+
| COUNT(*) |
+-----+
|      3   |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM students WHERE age > 19;
+---+-----+-----+-----+
| id | name  | age   | department |
+---+-----+-----+-----+
| 1  | Alice | 20    | CSE      |
| 2  | Bob   | 21    | ECE      |
+---+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> DELETE FROM students WHERE name = 'Bob';
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM students;
+---+-----+-----+-----+
| id | name  | age   | department |
+---+-----+-----+-----+
| 1  | Alice | 20    | CSE      |
| 3  | Charlie | 19    | IT       |
+---+-----+-----+-----+
2 rows in set (0.00 sec)
```

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

```
mysql> EXIT;
Bye
ubuntu@ip-172-31-0-74:~$ |
```

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

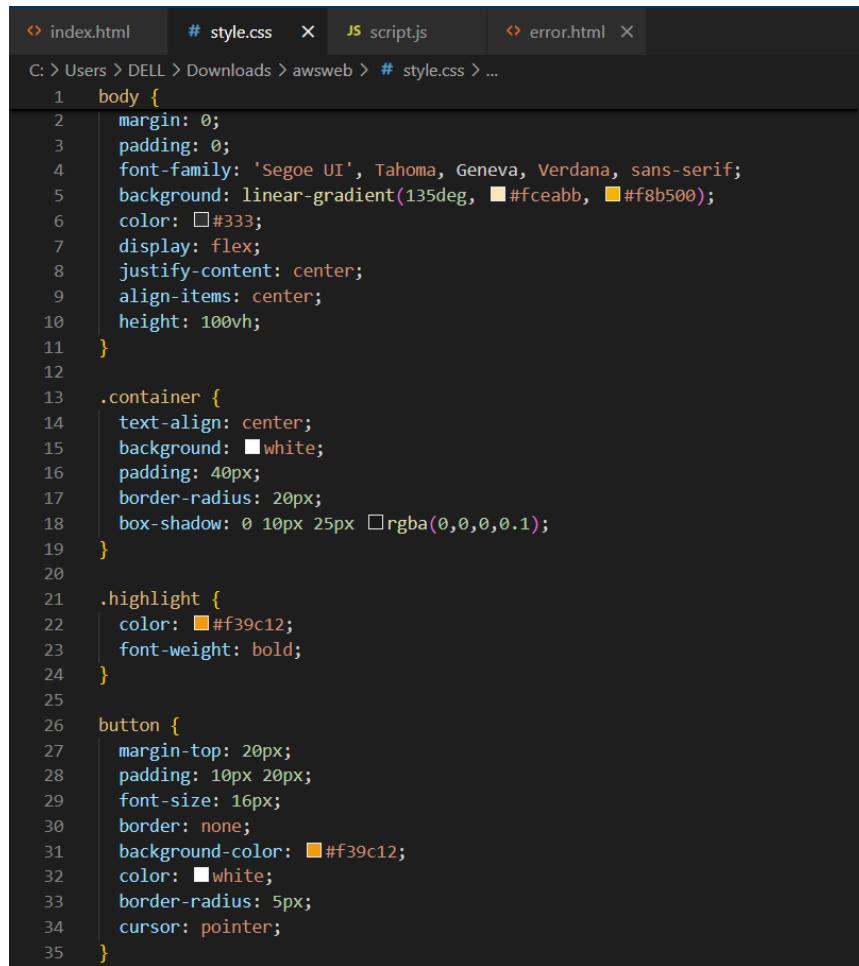
Index.html

```
index.html # style.css JS script.js error.html  
C: > Users > DELL > Downloads > awsweb > index.html > ...  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4    <meta charset="UTF-8" />  
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
6    <title>Welcome to Riya's Site</title>  
7    <link rel="stylesheet" href="style.css" />  
8  </head>  
9  <body>  
10   <div class="container">  
11     <h1>Hello, I'm <span class="highlight">Riya</span> 12     <p>Welcome to my beautiful static website hosted on AWS S3.</p>  
13     <button onclick="showMessage()">Click Me</button>  
14     <p id="message"></p>  
15   </div>  
16   <script src="script.js"></script>  
17 </body>  
18 </html>
```

Script.js

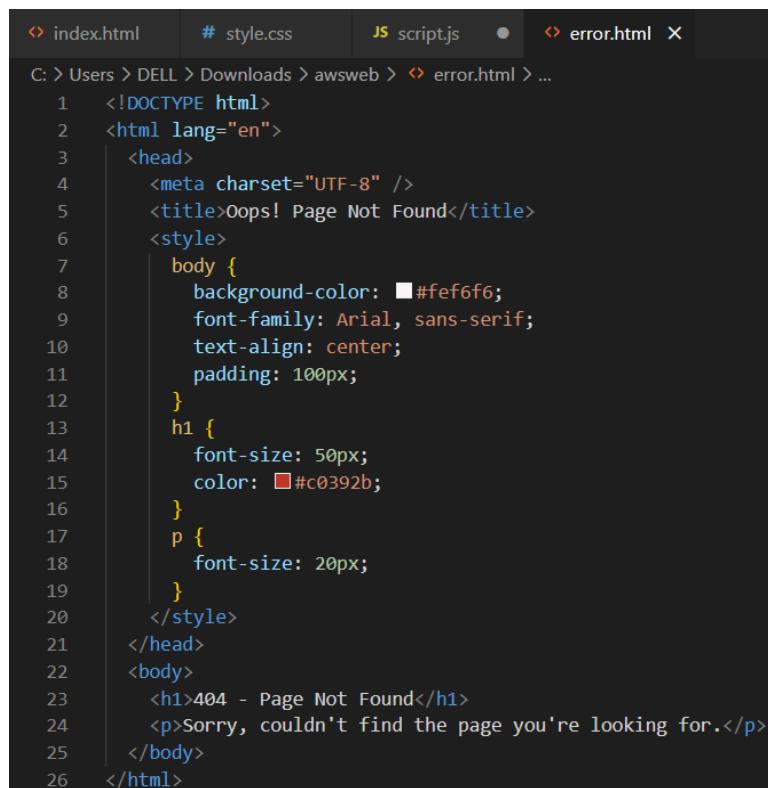
```
index.html # style.css JS script.js ● error.html  
C: > Users > DELL > Downloads > awsweb > script.js > ...  
1  function showMessage() {  
2    document.getElementById("message").innerText = "Thanks for visiting, I appreciate it!";  
3  }
```

Style.css



```
C: > Users > DELL > Downloads > awsweb > # style.css > ...
1 body {
2   margin: 0;
3   padding: 0;
4   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
5   background: linear-gradient(135deg, #fceabb, #f8b500);
6   color: #333;
7   display: flex;
8   justify-content: center;
9   align-items: center;
10  height: 100vh;
11 }
12
13 .container {
14   text-align: center;
15   background: white;
16   padding: 40px;
17   border-radius: 20px;
18   box-shadow: 0 10px 25px rgba(0,0,0,0.1);
19 }
20
21 .highlight {
22   color: #f39c12;
23   font-weight: bold;
24 }
25
26 button {
27   margin-top: 20px;
28   padding: 10px 20px;
29   font-size: 16px;
30   border: none;
31   background-color: #f39c12;
32   color: white;
33   border-radius: 5px;
34   cursor: pointer;
35 }
```

Error.html



```
C: > Users > DELL > Downloads > awsweb > error.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Oops! Page Not Found</title>
6     <style>
7       body {
8         background-color: #fef6f6;
9         font-family: Arial, sans-serif;
10        text-align: center;
11        padding: 100px;
12      }
13      h1 {
14        font-size: 50px;
15        color: #c0392b;
16      }
17      p {
18        font-size: 20px;
19      }
20    </style>
21  </head>
22  <body>
23    <h1>404 - Page Not Found</h1>
24    <p>Sorry, couldn't find the page you're looking for.</p>
25  </body>
26 </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)

Create bucket [Info](#)
Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type [Info](#)

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)
awsweb-riya

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

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

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through new access control lists (ACLS)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through any access control lists (ACLS)
S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

S3 Bucket is created successfully:

Buckets

Successfully created bucket "awsweb-riya"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

General purpose buckets [All AWS Regions](#) | **Directory buckets**

General purpose buckets (4) [Info](#)

Buckets are containers for data stored in S3.

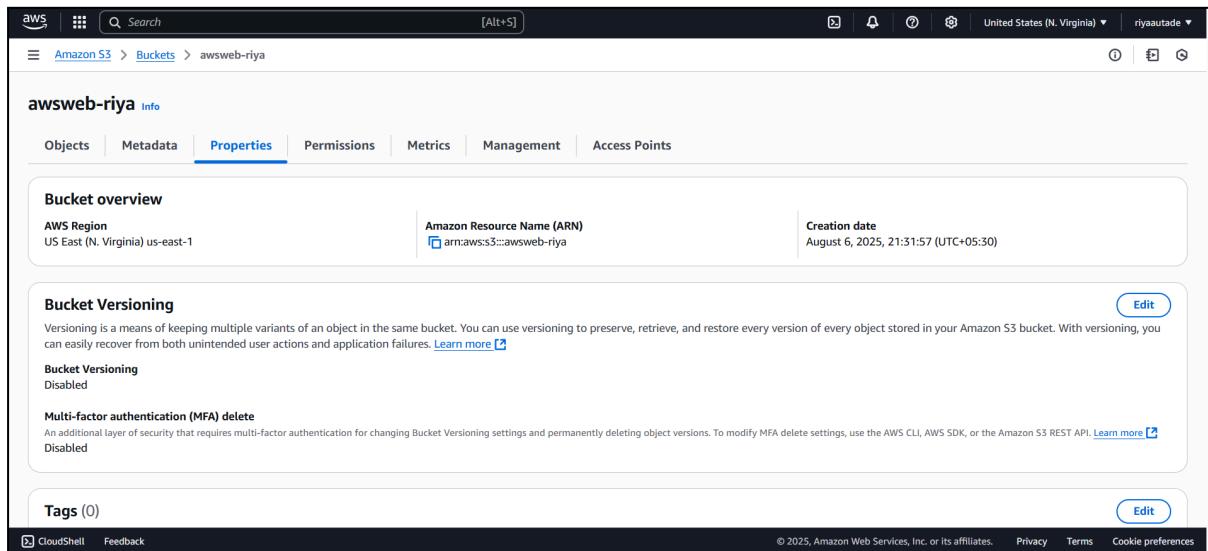
Name	AWS Region	Creation date
awsweb-riya	US East (N. Virginia) us-east-1	August 6, 2025, 21:31:57 (UTC+05:30)

Account snapshot [Info](#) [View dashboard](#)
Updated daily
Storage Lens provides visibility into storage usage and activity trends.

External access summary - new [Info](#)

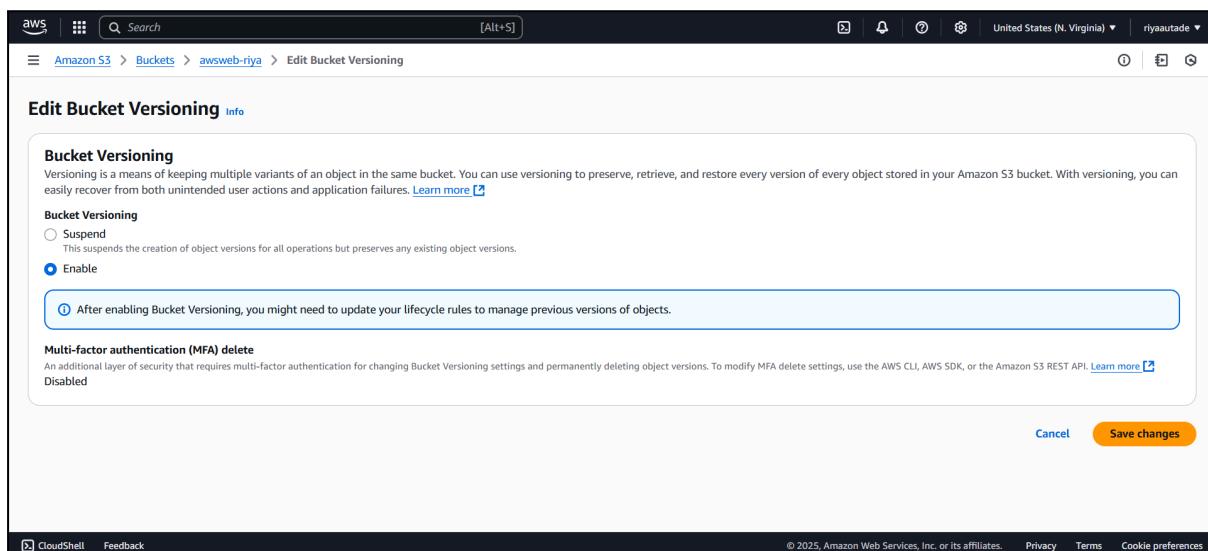
3. Enable Versioning for S3 Bucket

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



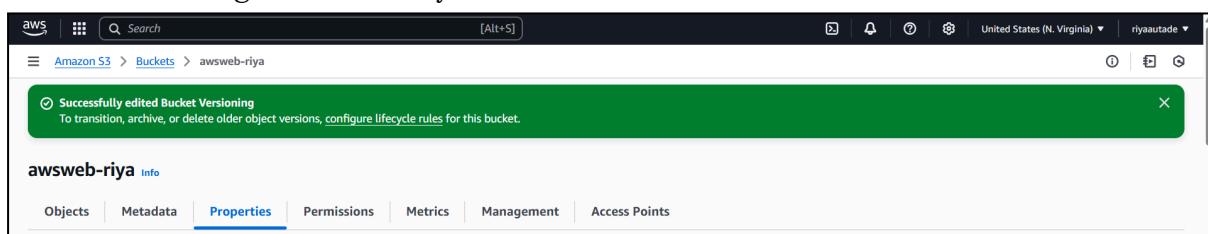
The screenshot shows the AWS S3 console. The top navigation bar includes the AWS logo, search bar, and United States (N. Virginia) region. The main content area is titled 'awsweb-riya' with a 'Properties' tab selected. In the 'Bucket overview' section, the ARN is shown as arn:aws:s3:::awsweb-riya. The 'Bucket Versioning' section indicates it is disabled. A note about MFA delete is present. The bottom of the page includes a 'Tags (0)' section and standard footer links.

- Choose Enable under Bucket Versioning
- Click Save changes



The screenshot shows the 'Edit Bucket Versioning' dialog box. It contains the 'Bucket Versioning' section with a note about enabling versioning. The 'Enable' option is selected. A note below says 'After enabling Bucket Versioning, you might need to update your lifecycle rules to manage previous versions of objects.' At the bottom are 'Cancel' and 'Save changes' buttons.

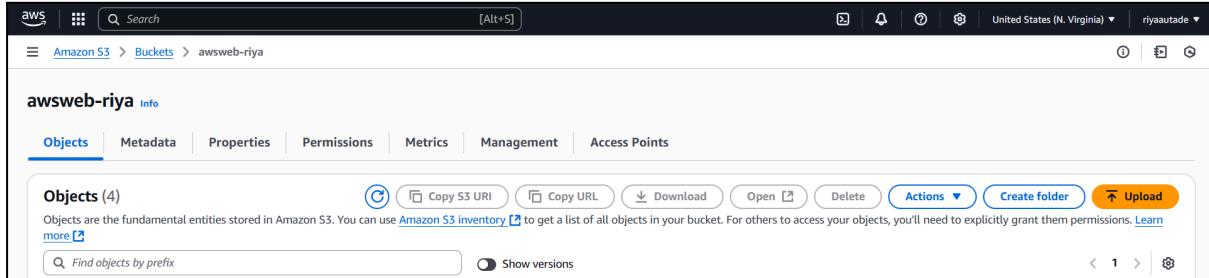
Bucket Versioning is successfully enabled:



The screenshot shows the AWS S3 console again. A green success message at the top reads 'Successfully edited Bucket Versioning' and 'To transition, archive, or delete older object versions, configure lifecycle rules for this bucket.' The 'Properties' tab is selected in the navigation bar.

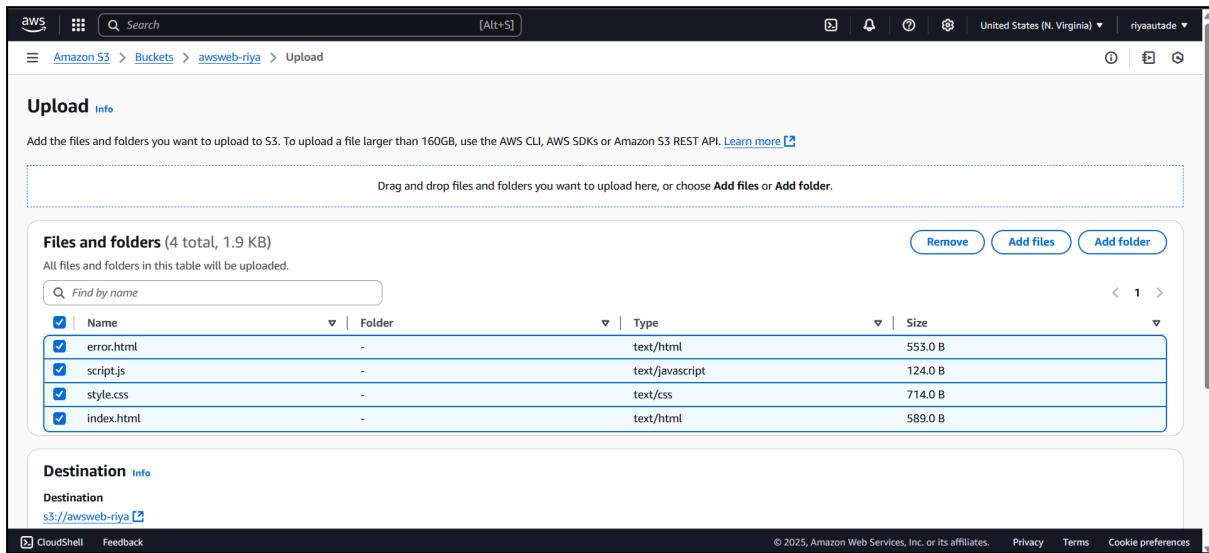
4. Upload Your Website Files

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



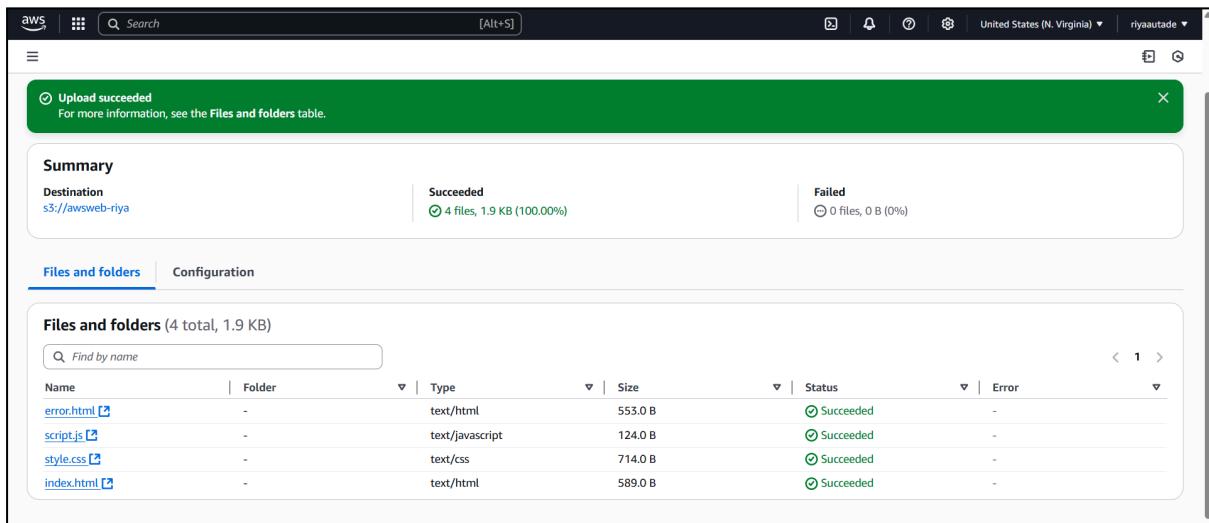
The screenshot shows the AWS S3 console with the 'Objects' tab selected for the 'awsweb-riya' bucket. There are four objects listed: 'index.html', 'style.css', 'script.js', and 'error.html'. The 'Actions' dropdown menu is open, showing options like Copy S3 URI, Copy URL, Download, Open, Delete, Actions (selected), Create folder, and Upload.

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



The screenshot shows the AWS S3 console on the 'Upload' page for the 'awsweb-riya' bucket. It shows a file selection area with a placeholder 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' and a table of files to be uploaded. The table includes columns for Name, Type, and Size. Below the table is a 'Destination' section with the URL 's3://awsweb-riya'.

Website files are successfully uploaded in S3:



The screenshot shows the AWS S3 console displaying the results of a recent upload. A green banner at the top indicates 'Upload succeeded'. Below it, a summary table shows 'Succeeded' (4 files, 1.9 KB (100.00%)) and 'Failed' (0 files, 0 B (0%)). At the bottom, a detailed 'Files and folders' table lists the four uploaded files: 'error.html', 'script.js', 'style.css', and 'index.html', each with its respective type, size, status (all marked as 'Succeeded'), and error status.

5. Enable Static Website Hosting

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

The screenshot shows the AWS S3 Bucket properties page for 'awsweb-riya'. In the 'Static website hosting' section, there is a note: 'We recommend using AWS Amplify Hosting for static website hosting'. A 'Create Amplify app' button is visible. The 'S3 static website hosting' status is set to 'Disabled'.

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

The screenshot shows the 'Edit static website hosting' configuration page. Under 'Hosting type', 'Host a static website' is selected. In the 'Index document' field, 'index.html' is specified. In the 'Error document - optional' field, 'error.html' is specified. A note states: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access.'

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

The screenshot shows the AWS S3 Bucket properties page again. A green success message bar at the top says: 'Successfully edited static website hosting.' The 'S3 static website hosting' status is now set to 'Enabled'. The 'Bucket website endpoint' is shown as <http://awsweb-riya.s3-website-us-east-1.amazonaws.com>.

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

The screenshot shows the AWS S3 console for the bucket 'awsweb-riya'. The 'Permissions' tab is selected. In the 'Block public access (bucket settings)' section, 'Block all public access' is set to 'Off'. In the 'Bucket policy' section, there is a JSON policy provided by AWS. The policy allows public read access to objects in the bucket.

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "PublicReadForWebsite",  
6             "Effect": "Allow",  
7             "Principal": "*",  
8             "Action": "s3:GetObject",  
9             "Resource": "arn:aws:s3:::awsweb-riya/*"  
10        }  
11    ]  
12}  
13
```

- 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/*"  
    }  
]
```

}

The screenshot shows the 'Edit bucket policy' page for the 'awsweb-riya' bucket. The 'Policy' section contains the JSON policy from above. The 'Edit statement' and 'Select a statement' sections are visible on the right, along with a button to add a new statement.

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "PublicReadForWebsite",  
6             "Effect": "Allow",  
7             "Principal": "*",  
8             "Action": "s3:GetObject",  
9             "Resource": "arn:aws:s3:::awsweb-riya/*"  
10        }  
11    ]  
12}  
13
```

Bucket Policy is successfully edited:

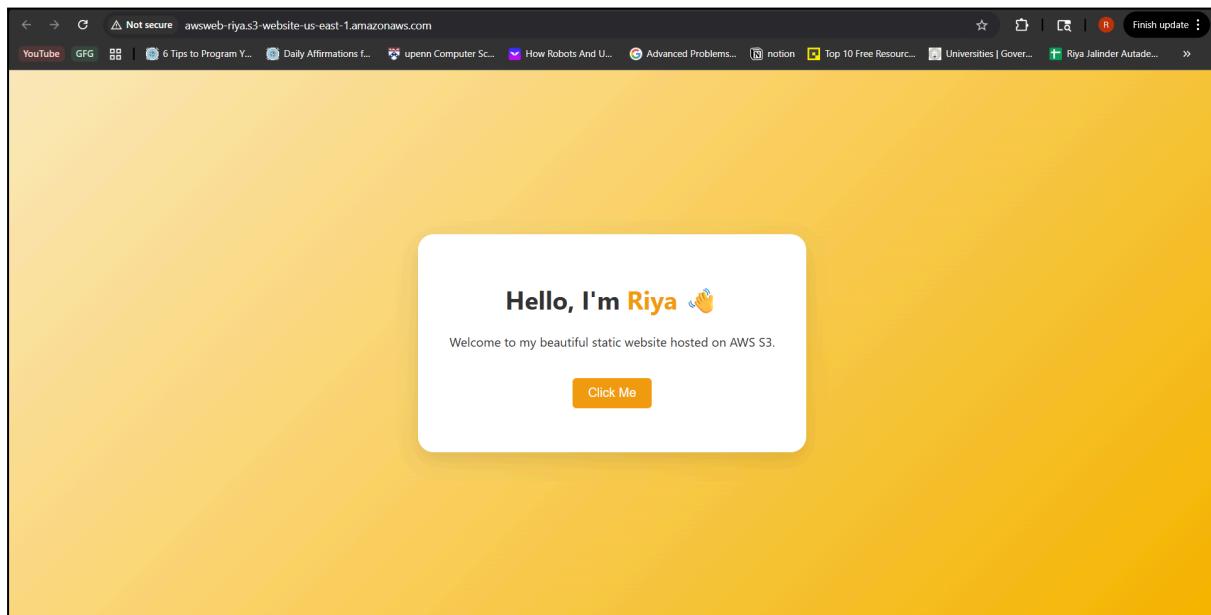
The screenshot shows the AWS S3 Bucket Policy editor. At the top, there's a green success message: "Successfully edited bucket policy." Below it, the title "Bucket policy" is displayed, along with a note: "The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts." There are "Edit" and "Delete" buttons. The main area contains the JSON policy code:

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

A "Copy" button is located in the top right corner of the JSON editor.

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!
