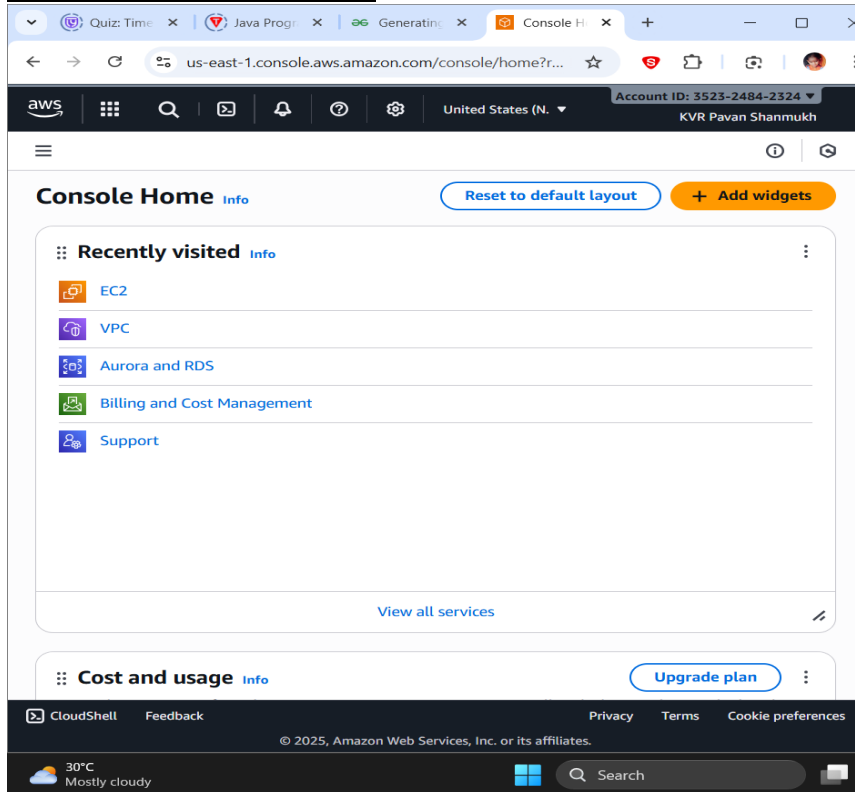


LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker. By 2300090201

Aim: Deploy the Full Stack Project using GitHub Actions with Docker.

Steps:

AWS Management Console:



->Click on VPC:

-->Select security grps under the Security section in the left menu bar.(If VPC is not available directly search in the search bar.)

-->Select the existing security group.

-->Click Edit Inbound Rules and delete the rule.

-->Click on Add Rule and ensure that the type is set to All Traffic and Source Type is set to IPV4.

-->Click on SAVE Rules

In AWS Home Page Click on EC2: (Instance creation process)

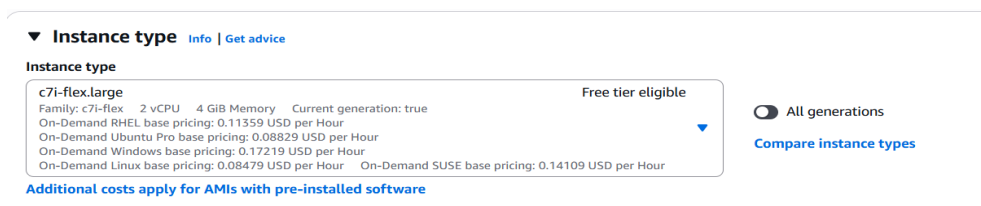
In the left menu bar, click on instances.

click on launch instance

Give name as machine1

Application and OS Images (Amazon Machine Image): Ubuntu

Instance type: t3.medium / t2.medium



LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker. By 2300090201

Key pair (login):

- > create a new pair
- > give name as s111-key
- > select the ".pem" as extension.
- > click on create key pair.

Firewall (security groups): select existing security grps and select the default security group.

Configure storage: 20 GiB

The screenshot shows the AWS Management Console configuration page for an EC2 instance. The 'Subnet' section is set to 'No preference (Default subnet in any availability zone)'. The 'Auto-assign public IP' section is set to 'Enable'. The 'Firewall (security groups)' section has 'Select existing security group' selected, and a dropdown menu shows 'default sg-0b9518d00557198e6' selected. The 'Configure storage' section shows '1x 20 GiB gp3' as the root volume, with a note 'Root volume, 3000 IOPS, Not encrypted'. There is a 'Compare security group rules' button next to the security group selection.

Connect the instance.

JUST HOLD THE PROCESS AND OPEN THE GITHUB.

Open GitHub:

Create a new repository, do not push any files initially.

Go to Repo settings,



Actions → General → Scroll down and select Read & Write permissions. Save.

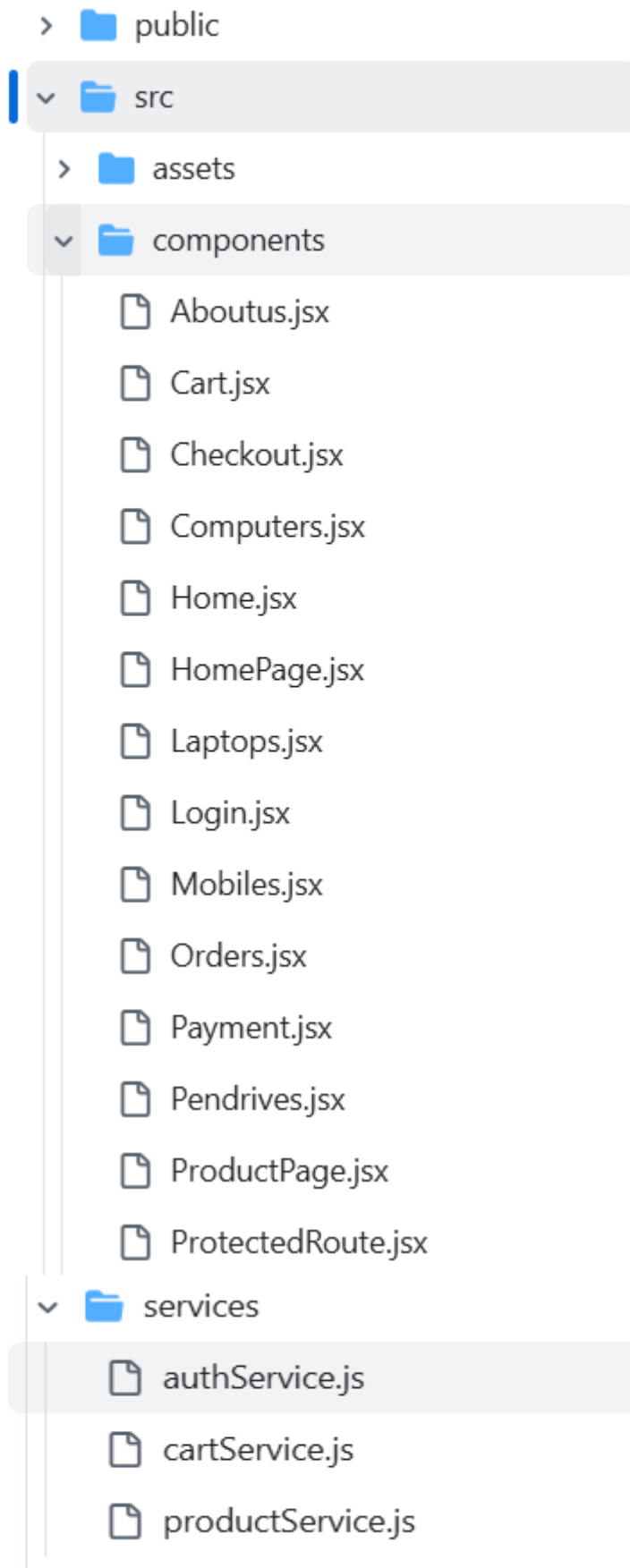
In the Skill11 repo-

In the frontend files: check each and every file and replace the ip address like:

For frontend only.

<http://localhost:8083/ecommerce> → <http://<ec2-ip>:8083/ecommerce>

LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker.
By 2300090201



LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker.

By 2300090201

Workflow permissions

Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. [Learn more about managing permissions.](#)

☒ **Read and write permissions**

Workflows have read and write permissions in the repository for all scopes.

☐ **Read repository contents and packages permissions**

Workflows have read permissions in the repository for the contents and packages scopes only.

Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.

☐ **Allow GitHub Actions to create and approve pull requests**

Save

Secrets & Variables → Actions → Create repo secret →

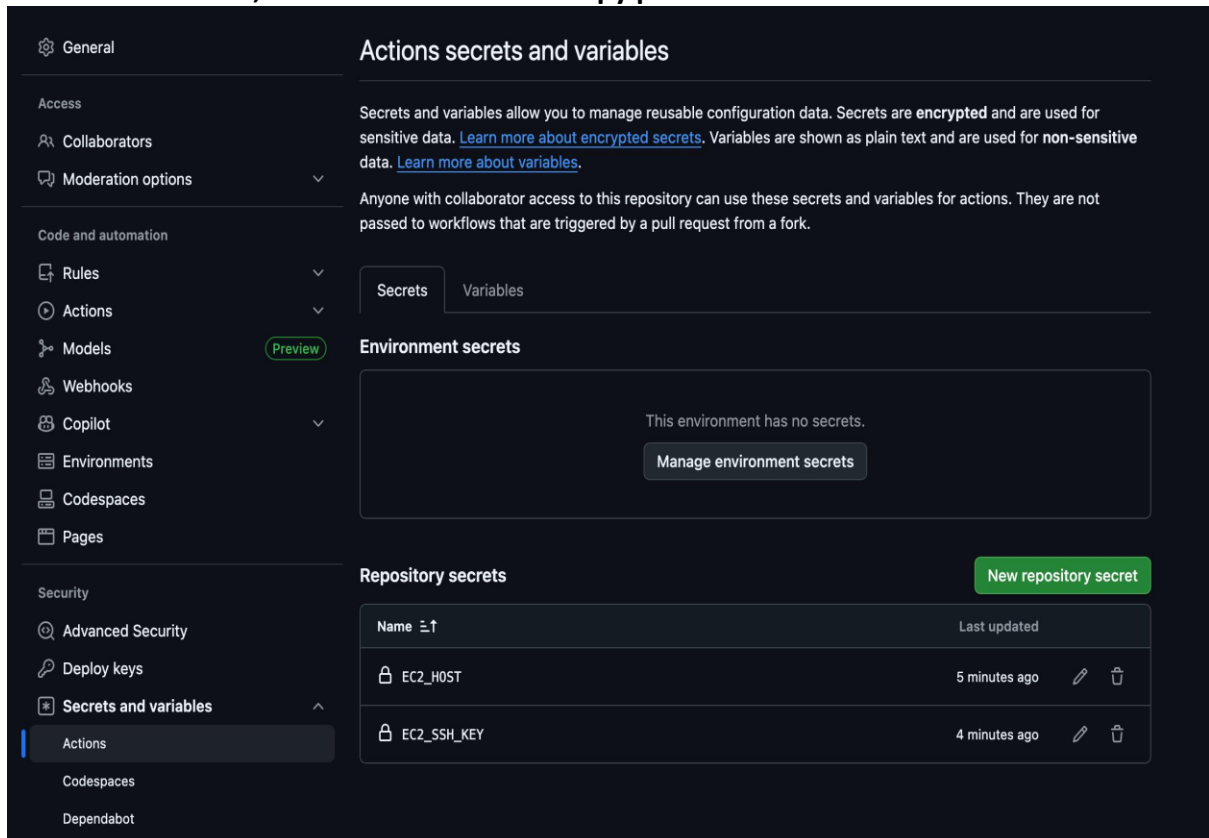
1st: Name:EC2_HOST

Value: your EC2_IP

2nd : Name : EC2_SSH_KEY

Value: Copy paste the content in .pem key pair file.

For the MAC Users, use cat command and copy paste.



The screenshot shows the GitHub Actions secrets and variables interface. On the left is a sidebar with navigation options: General, Access, Collaborators, Moderation options, Code and automation (Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, Pages), and Security (Advanced Security, Deploy keys, Secrets and variables). The 'Secrets and variables' section is expanded, showing 'Actions' selected. The main content area is titled 'Actions secrets and variables' and contains three sections: 'Environment secrets' (with a 'Manage environment secrets' button), 'Repository secrets' (with a 'New repository secret' button and a table of secrets), and 'Variables' (currently inactive). The 'Repository secrets' table lists two secrets: 'EC2_HOST' and 'EC2_SSH_KEY', both created 5 and 4 minutes ago respectively.

Name	Last updated
EC2_HOST	5 minutes ago
EC2_SSH_KEY	4 minutes ago

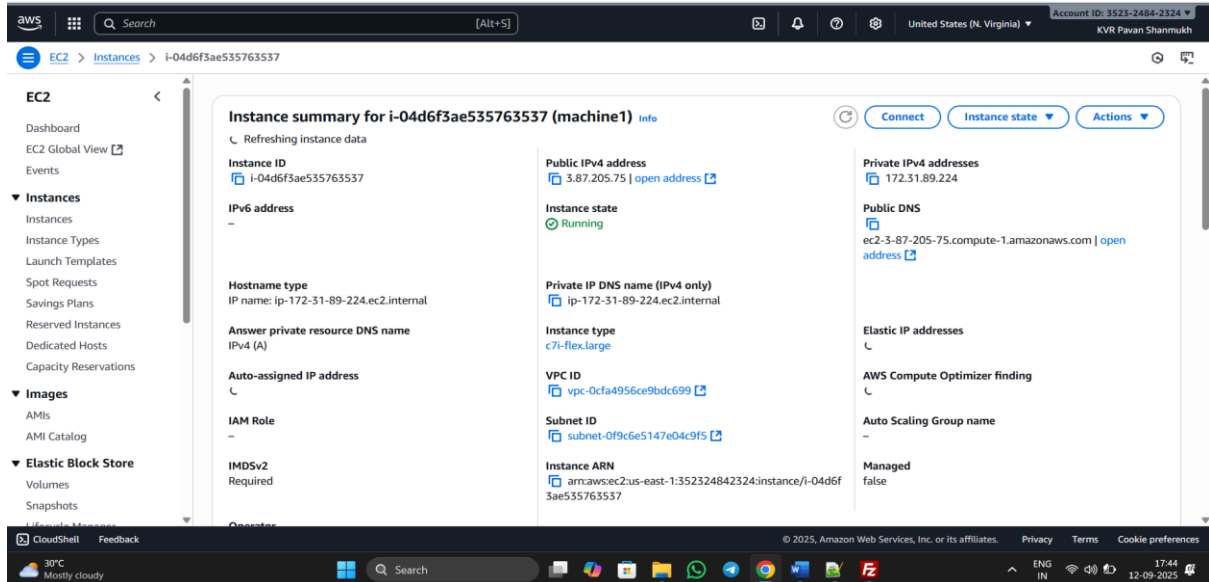
LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker.

By 2300090201

Now in AWS Console:

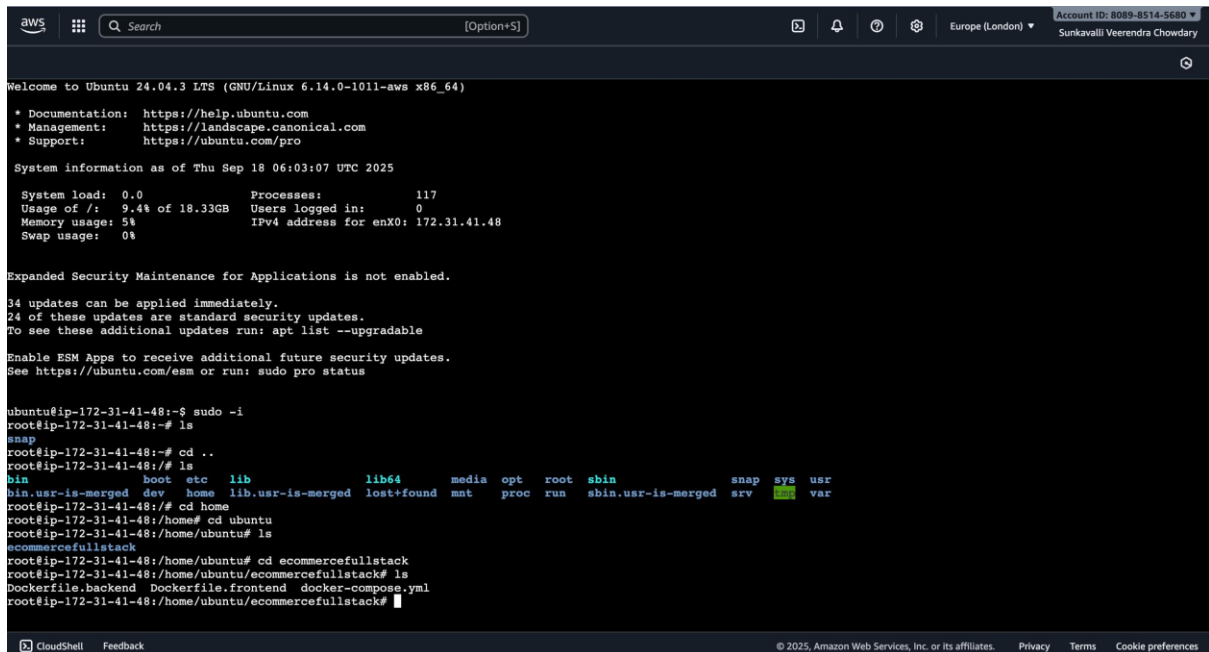
Scroll down and launch instance

-->Click Connect and then connect.



The Ubuntu environment will be open.

Run the commands one after another:



Run the commands:

docker exec -it ecommerce-db1 /bin/bash

mysql -u root -p

root

LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker. By 2300090201

```
select * from ecommerce.users;
```

Output Verification:

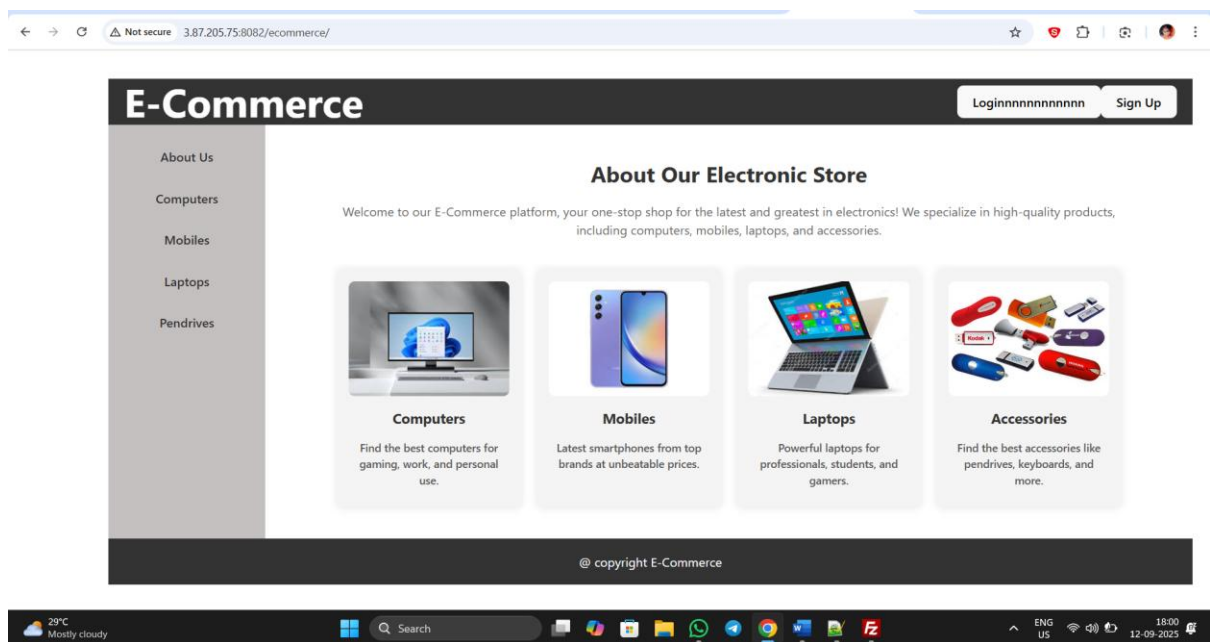
Place the ec2ip like this to see the output:

Frontend:

<http://<your ec2 ip>:8082/ecommerce/>

Backend:

<http://<your ec2 ip>:8083/back1/>



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Thu Sep 18 06:10:29 UTC 2025

There was an unexpected error (type=Forbidden, status=403).