

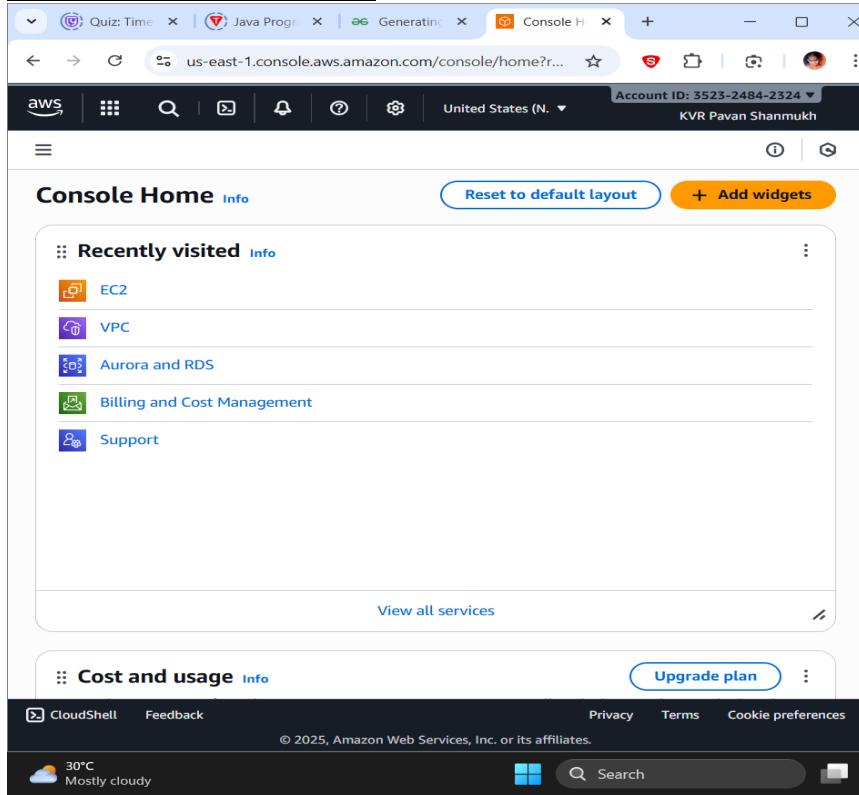
LAB 6- Deploy the Full Stack Project using GitHub Actions with Docker.

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)

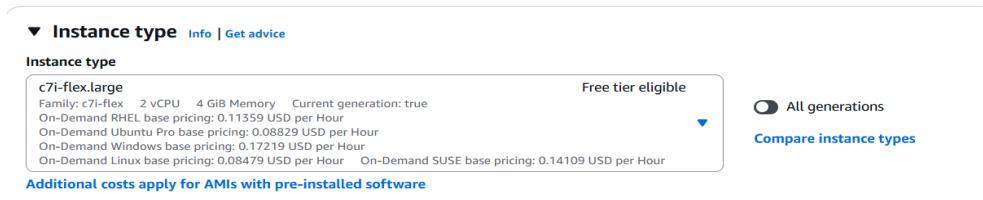
In the left menu bar, click on instances.

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 CloudFormation configuration interface. At the top, there are sections for 'Subnet' and 'Auto-assign public IP'. Under 'Firewall (security groups)', the 'Select existing security group' option is selected, and a dropdown menu shows 'default sg-0b9518d00557198e6' with a VPC listed below it. In the 'Common security groups' section, a dropdown menu shows 'Select security groups'. Below these, a section for 'Configure storage' is shown, indicating 1x 20 GiB gp3 volume. An 'Advanced' link is visible at the top right of this section.

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.

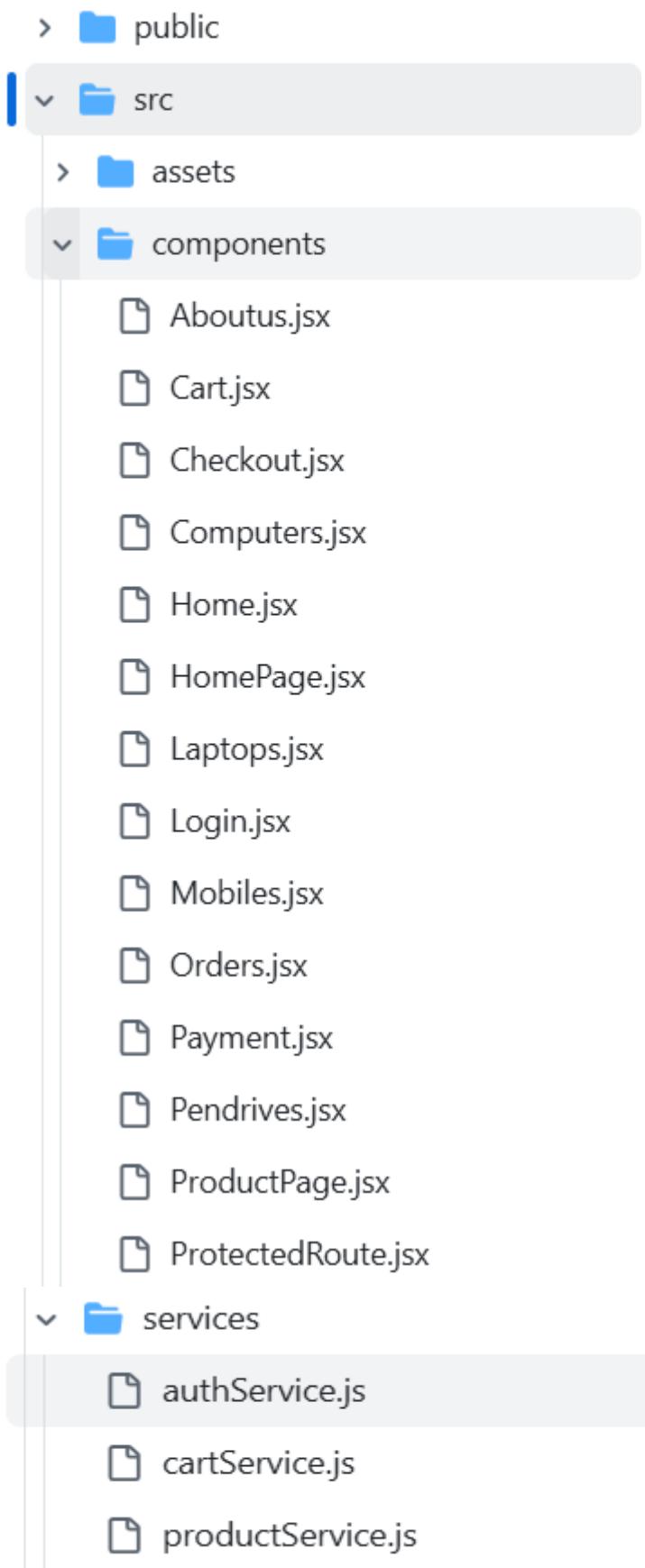
In the Skill11 repo-

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 page. The left sidebar has a tree view with General, Access, Collaborators, Moderation options, Code and automation (Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, Pages), Security (Advanced Security, Deploy keys, Secrets and variables), Actions, Codespaces, and Dependabot. The 'Actions' section is currently selected. The main content area has tabs for 'Secrets' (selected) and 'Variables'. Under 'Environment secrets', it says 'This environment has no secrets.' and has a 'Manage environment secrets' button. Under 'Repository secrets', there are two entries:

Name	Last updated	Actions	
EC2_HOST	5 minutes ago		
EC2_SSH_KEY	4 minutes ago		

A green 'New repository secret' button is located at the top right of the 'Repository secrets' table.

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 screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main area displays an 'Instance summary' for an instance named 'machine1' with the ID i-04d6f3ae535763537. Key details include:

- Public IPv4 address:** 3.87.205.75 | [open address](#)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-89-224.ec2.internal
- Instance type:** c7i-flex.large
- VPC ID:** vpc-0cfa4956ce9bcd699
- Subnet ID:** subnet-0f9c6e5147e04c9f5
- Instance ARN:** arn:aws:ec2:us-east-1:352324842324:instance/i-04d6f3ae535763537

At the bottom, there are buttons for CloudShell and Feedback, along with a system tray showing the date and time (12-09-2025).

The Ubuntu environment will be open.
Run the commands one after another:

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

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

System information as of Thu Sep 18 06:03:07 UTC 2025

System load: 0.0          Processes:           117
Usage of /: 9.4% of 18.33GB  Users logged in: 0
Memory usage: 5%           IPv4 address for enX0: 172.31.41.48
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

34 updates can be applied immediately.
24 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

ubuntu@ip-172-31-41-48:~$ sudo -i
root@ip-172-31-41-48:~# ls
snap
root@ip-172-31-41-48:~# cd ..
root@ip-172-31-41-48:~/#
root@ip-172-31-41-48:~/bin lib lib64 media opt root sbin snap sys usr
root@ip-172-31-41-48:~/bin dev home lib usr-is-merged lost+found mnt proc run sbin usr-is-merged srv var
root@ip-172-31-41-48:~/home# cd ubuntu
root@ip-172-31-41-48:~/home/ubuntu# ls
ecommercefullstack
root@ip-172-31-41-48:~/home/ubuntu# cd ecommercefullstack
root@ip-172-31-41-48:~/home/ubuntu/ecommercefullstack# ls
Dockerfile.backend Dockerfile.frontend docker-compose.yml
root@ip-172-31-41-48:~/home/ubuntu/ecommercefullstack# 
```

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:

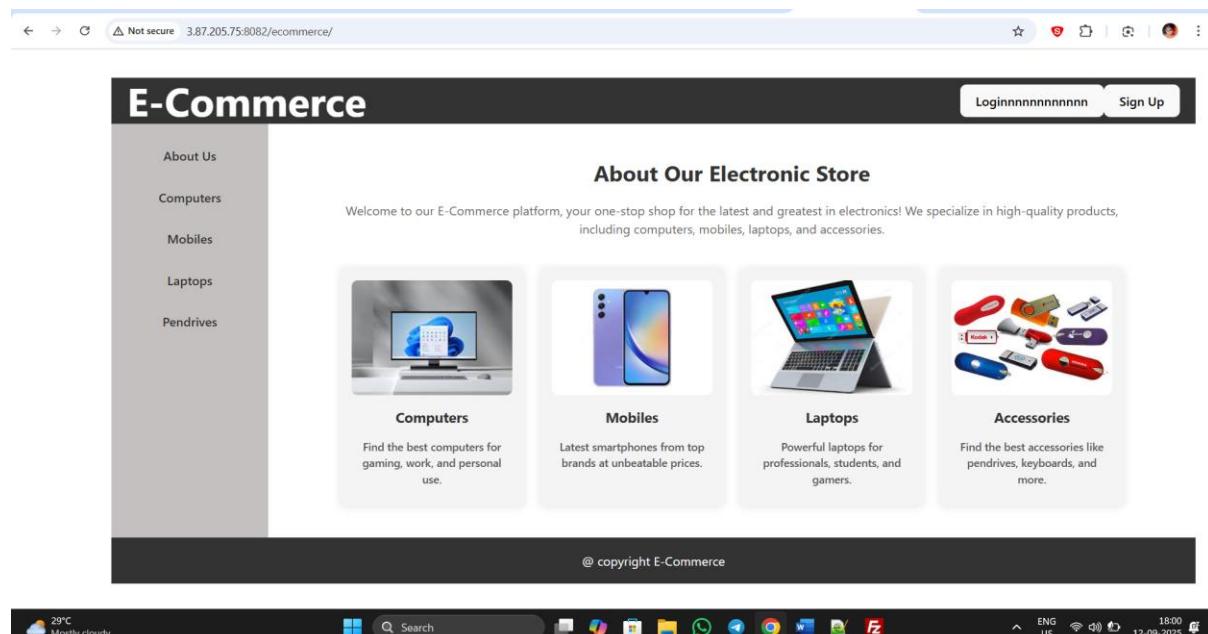
Place the ec2ip like this to see the output:

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).