Task 2: GitHub Actions CI/CD Pipeline Flask App

Objective:

Implement a CI/CD workflow using GitHub Actions for a Python application.

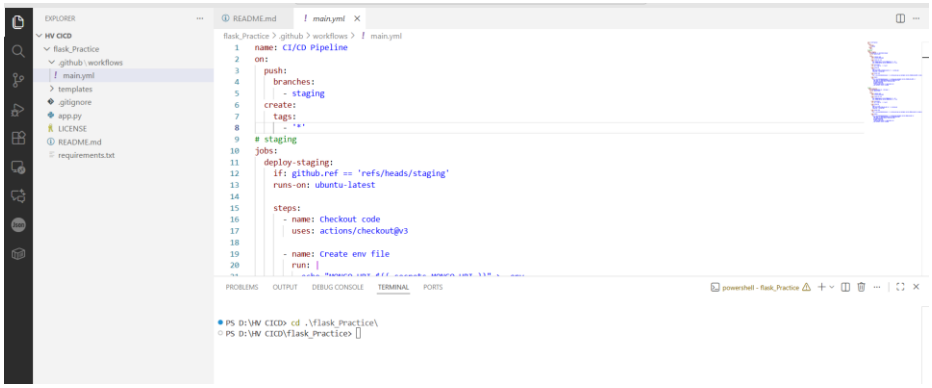
Github Link: https://github.com/jatinggg/flask_Practice.git

Steps:

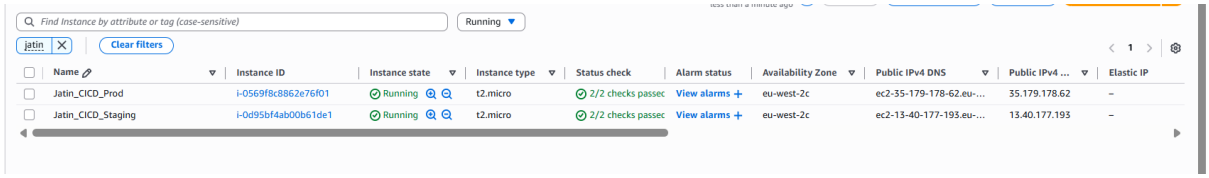
1. Fork the provided repo in your github account.

2. Create a staging branch in your forked repo.
3. Clone the github repository in your local system.

4. Now create a new directory .github/workflows in this and add main.yml file with following code in both main and staging branch:



5. Launch 2 ec2 instances (for prod and staging) with inbound rules allowed at port 8000 from anywhere ipv4



6. SSH into ec2 instance and create a flaskapp.service under the sysemd directory (configure for both the environments)
Run below commands:

```
sudo apt get update

sudo apt install -y python3-pip python3-venv unzip

sudo mkdir -p /home/ubuntu/app

sudo chown ubuntu:ubunut /home/ubuntu/app
```

```
sudo nano /etc/systemd/system/flaskapp.service
```

flaskapp.service file :

```
[Unit]
Description=Gunicorn instance to serve Flask App
After=network.target

[Service]
User=ubuntu
Group=ubuntu
WorkingDirectory=/home/ubuntu/app
Environment="PATH=/home/ubuntu/app/venv/bin:/usr/local/bin:/usr/bin"
EnvironmentFile=/home/ubuntu/app/.env

# FIX IS HERE:
# 1. Point to the 'venv' gunicorn
# 2. Bind to 0.0.0.0:5000 (Required for browser access)
ExecStart=/home/ubuntu/app/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

Restart=always

[Install]
WantedBy=multi-user.target
```

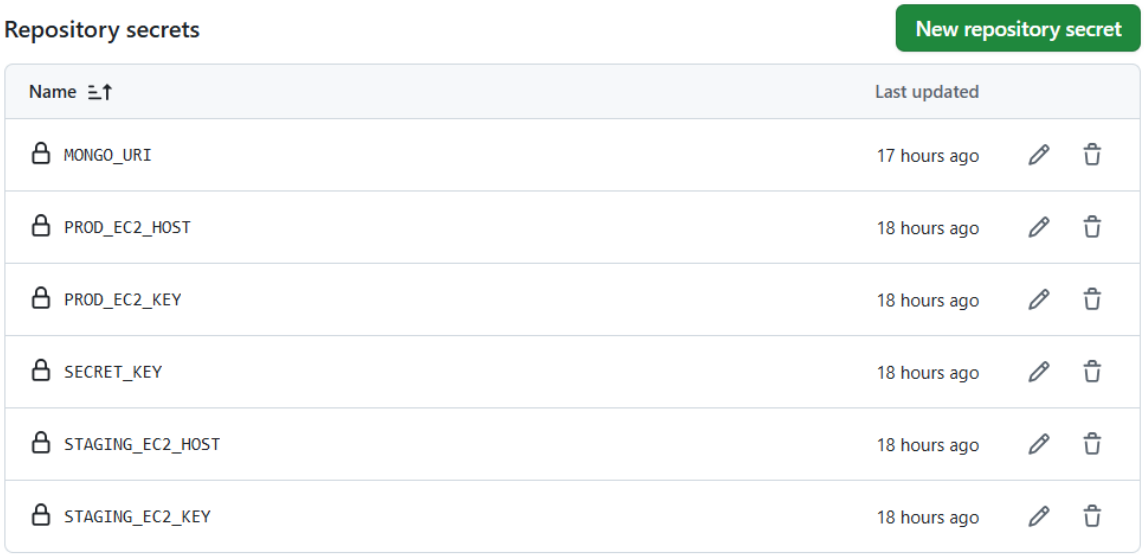
now run below commands:

```
sudo systemctl daemon-reload
sudo systemctl enable flaskapp.service
sudo systemctl start flaskapp.service
```

7. Update the required dependencies in the requirements.txt file on the visual studio

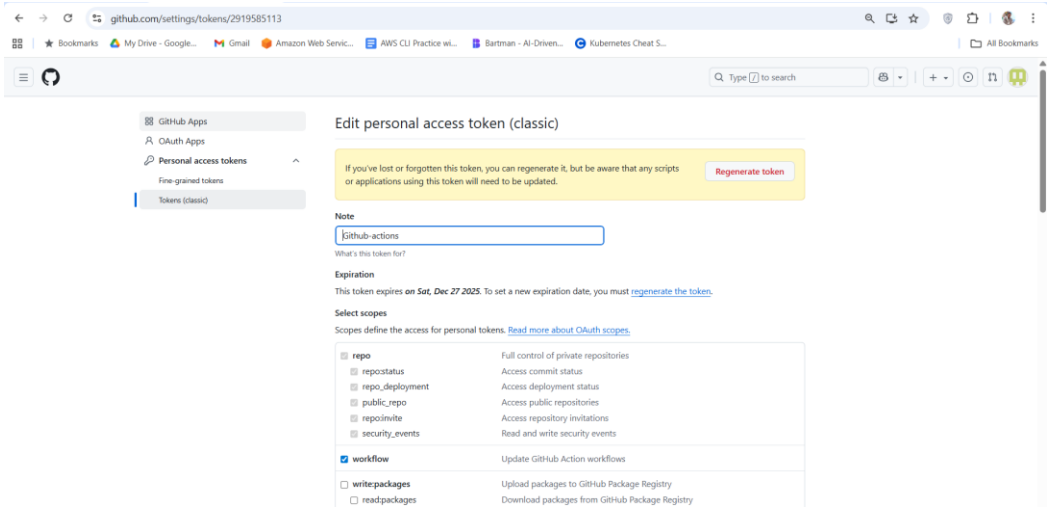
8. Create a MongoDB cluster with traffic allowed from all ip addresses.

9. On the GitHub repository add below secrets from settings > secrets and variables > actions > New Repository Secret (for SRCRET_KEY use any random string, I used 123456)



10. Now in the visual studio make any changes in any file (make sure you have switched to the staging branch using “git switch staging”).

11. You will need to create a Personal Access Token from GitHub with permission to workflow and repository and configure it on your local machine.



git remote set-url origin https://<your-PAT>@github.com/jatinggg/flask_Practice.git

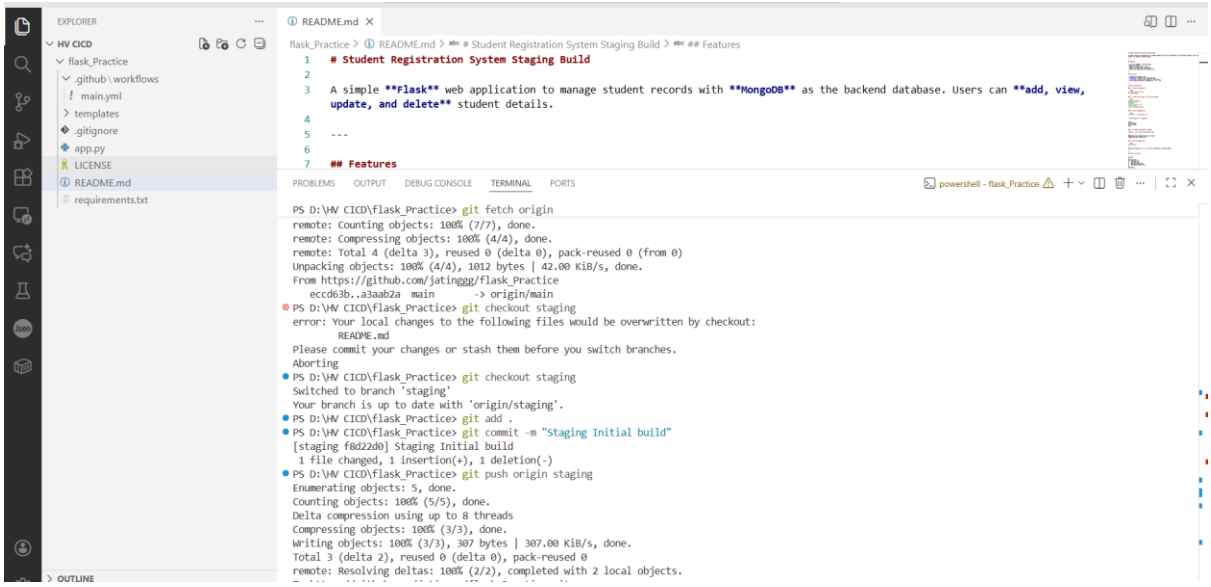
12. Now run below commands to trigger the CI/CD pipeline build

git checkout staging

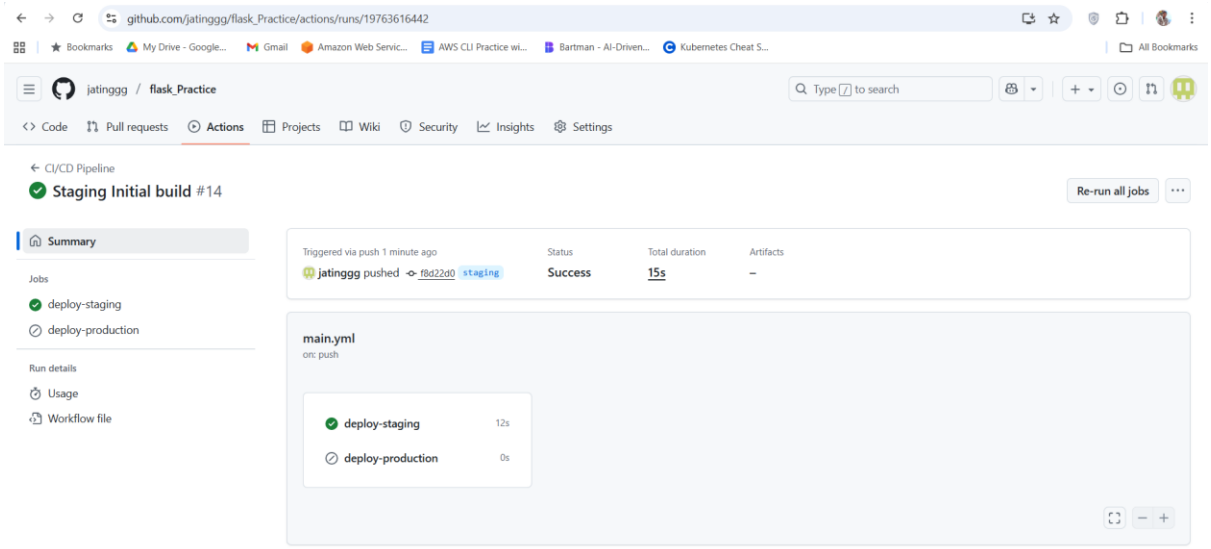
```
git add .
```

```
git commit -m "Staging Initial build"
```

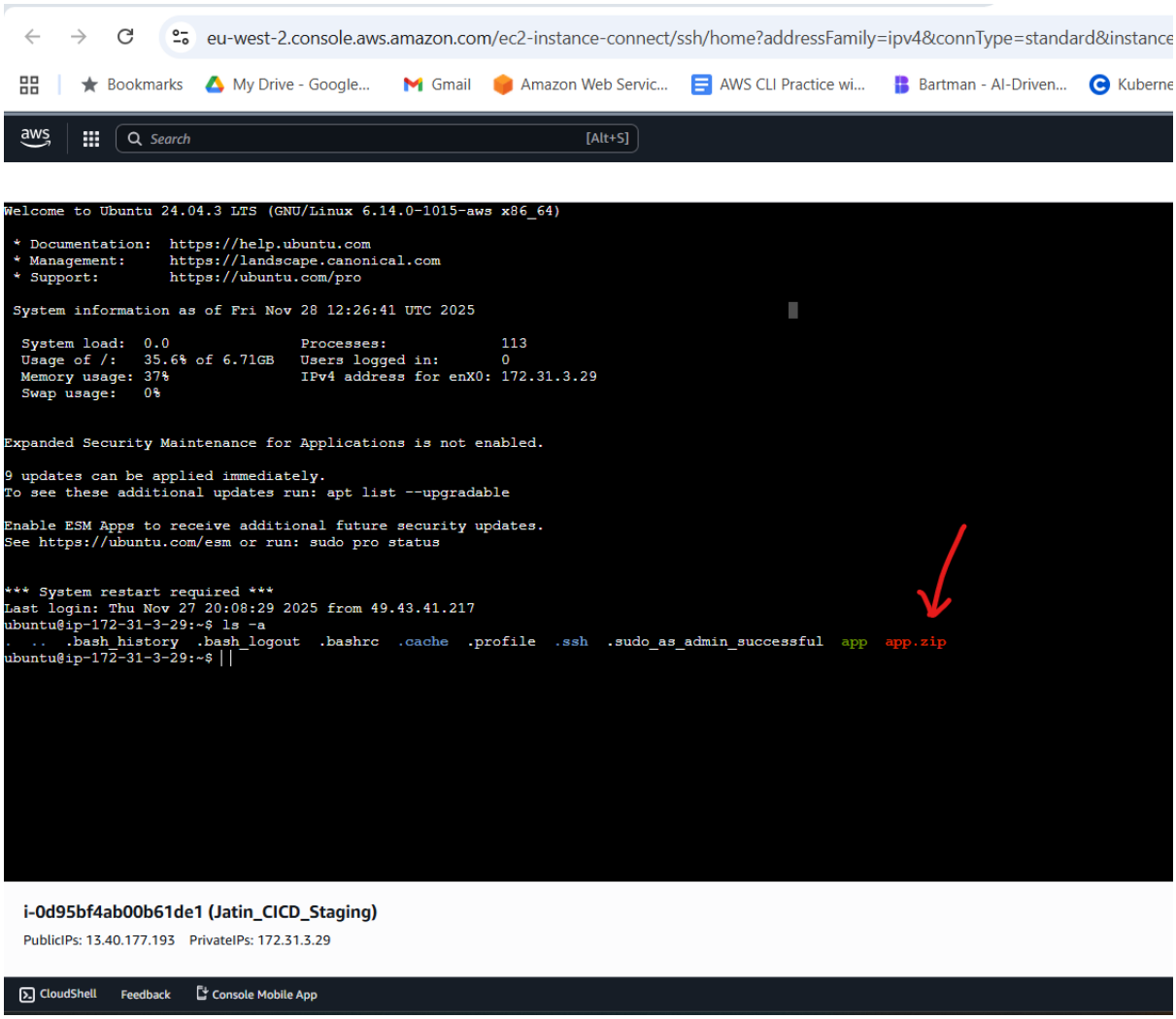
```
git push origin staging
```



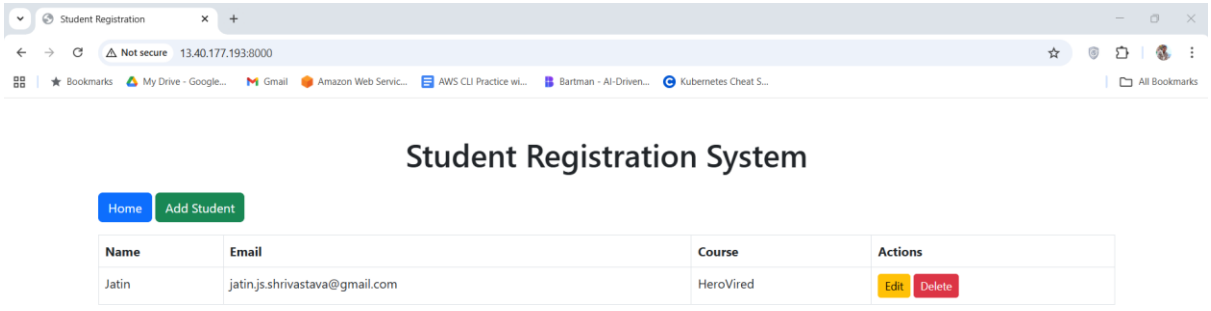
13. Verify the build from the Github Actions console



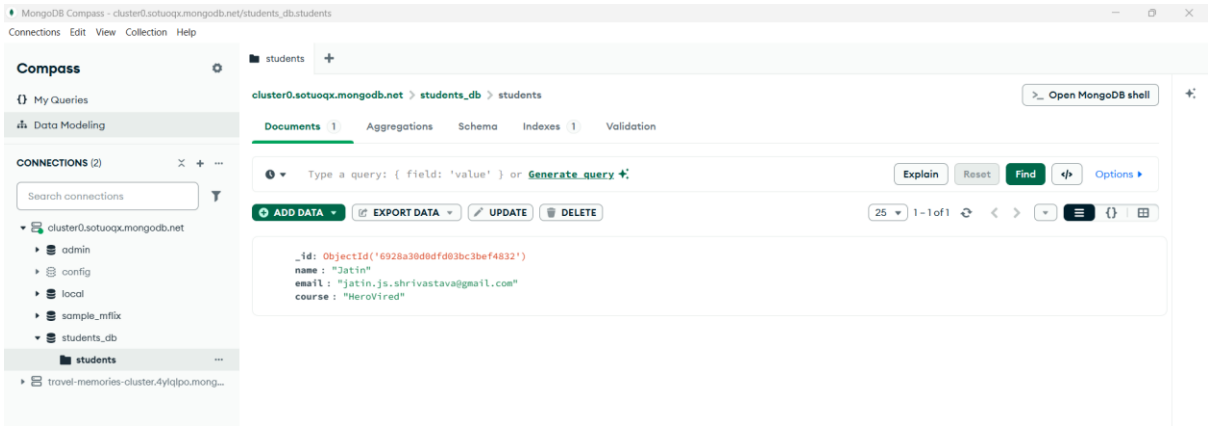
14. SSH into the Staging server and look under the root directory you can find the app.zip artifact and this file will be used to push the changes to the prod server once it has been tested.



15. You can now access the application at the staging server at port 8000

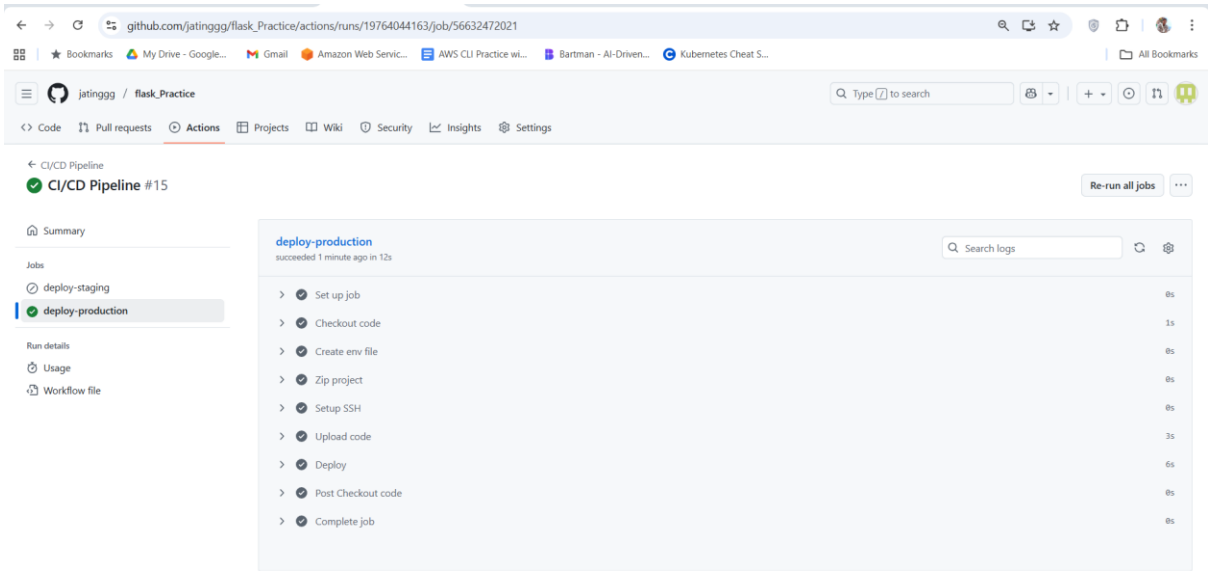


you can also verify this data in the mongodb compass



16. Now that we have verified the application and it has passed all the test cases on the staging server, we can push the code to the prod server.
In the visual studio switch to the main branch and run following commands:

```
git fetch origin

git pull origin main

git checkout main
git commit -m “push to prod”

git merge origin/staging

git tag v1.0.1

git push origin v1.0.1
```



17. You can now access the application on the Prod server at port 8000.

Student Registration System Prod

Home | Add Student

Name	Email	Course	Actions
Jatin	jatin.js.shrivastava@gmail.com	HeroVired	Edit Delete
Jatin-Prod	jatin.js.shrivastava-Prod@gmail.com	HeroVired-prod	Edit Delete