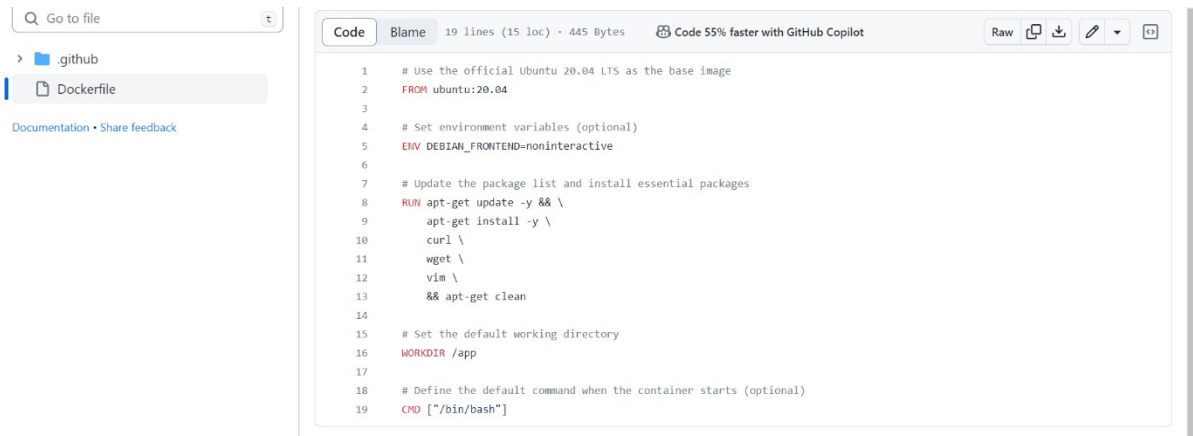


Lab-4

Docker Build and Push Using GitHub Actions

Step 1 → Fork and clone the GitHub repository



The screenshot shows a GitHub repository interface. On the left, there's a sidebar with a search bar and a file explorer showing a folder named '.github' and a file named 'Dockerfile'. The main area displays the content of the 'Dockerfile' file, which is 19 lines long. The code is as follows:

```
1 # Use the official Ubuntu 20.04 LTS as the base image
2 FROM ubuntu:20.04
3
4 # Set environment variables (optional)
5 ENV DEBIAN_FRONTEND=noninteractive
6
7 # Update the package list and install essential packages
8 RUN apt-get update -y && \
9     apt-get install -y \
10         curl \
11         wget \
12         vim \
13         && apt-get clean
14
15 # Set the default working directory
16 WORKDIR /app
17
18 # Define the default command when the container starts (optional)
19 CMD ["/bin/bash"]
```

Step 2 → Create Docker Hub Access Token

- Log in to your Docker Hub account.
- Go to your account settings and click on the “Security” tab
- Under "Access Tokens," click "New Access Token." Give it a name, select the required permissions (e.g., "Write" for pushing Docker images), and click "Create."
- Copy the generated access token. You will need it to authenticate with Docker Hub in your GitHub Actions workflow.

Step 3 → Create GitHub Action workflow

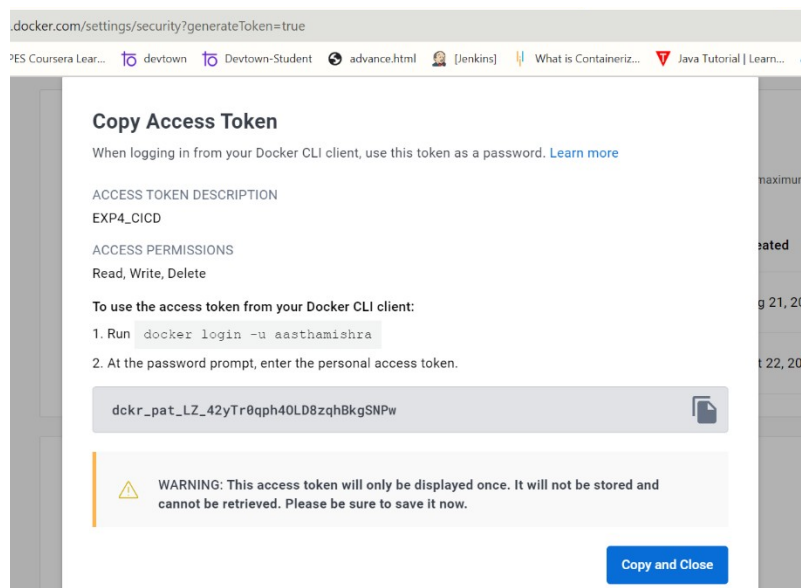


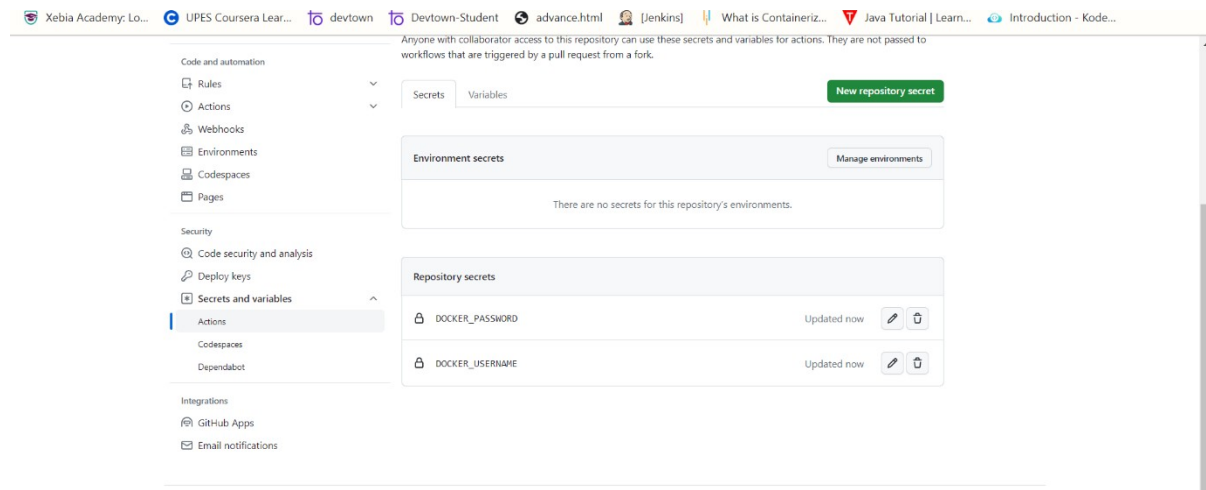
The screenshot shows a GitHub repository interface. On the left, the 'Files' sidebar shows a directory structure with 'github/workflows' expanded, listing 'Docker-Build-and-Push.yml' and 'Dockerfile'. The main area displays the content of 'Docker-Build-and-Push.yml', which is a GitHub Actions workflow. The workflow is named 'Docker Build and Push' and is triggered on a 'push' to the 'main' branch. It runs on 'ubuntu-latest' and consists of two jobs: 'build-and-push' and 'build-and-push'. The 'build-and-push' job has two steps: 'Checkout code' and 'Login to Docker Hub'. The 'Login to Docker Hub' step uses the 'actions/docker/login' action and sets environment variables for 'DOCKER_USERNAME' and 'DOCKER_PASSWORD'. The 'build-and-push' job also has a step named 'Build and Push Docker Image' which runs 'docker build -t 16chirag/exp_4:latest .' and 'docker push 16chirag/exp_4:latest'.

```
1 name: Docker Build and Push
2 on:
3   push:
4     branches:
5       - main # change this to your main branch name
6
7 jobs:
8   build-and-push:
9     runs-on: ubuntu-latest
10
11     steps:
12       - name: Checkout code
13         uses: actions/checkout@v2
14
15       - name: Login to Docker Hub
16         run: docker login -u ${ secrets.DOCKER_USERNAME } -p ${ secrets.DOCKER_PASSWORD }
17         env:
18           DOCKER_USERNAME: ${ secrets.DOCKER_USERNAME }
19           DOCKER_PASSWORD: ${ secrets.DOCKER_PASSWORD }
20
21       - name: Build and Push Docker Image
22         run: |
23           docker build -t 16chirag/exp_4:latest .
24           docker push 16chirag/exp_4:latest
```

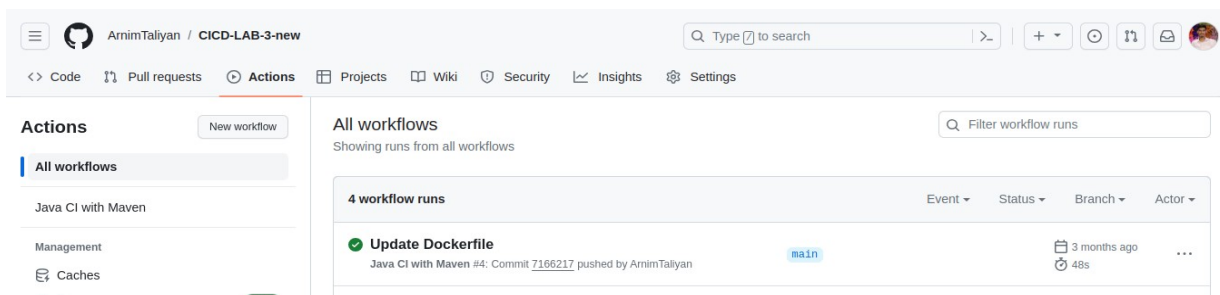
Step 4 → Add Docker Hub Credentials to GitHub Secrets

- Go to your GitHub repository on the GitHub website.
- Click on "Settings" and then "Secrets" in the left sidebar.
- Click on "New repository secret" and add two secrets:
- **DOCKER_USERNAME**: Set this to your Docker Hub username.
- **DOCKER_PASSWORD**: Set this to the Docker Hub access token you generated earlier.





Step 6 → Check workflow status



Step 7 → Verify the Docker Image on Docker Hub

