Experiment 4

Docker Build and Push using GitHub Actions

Objective: Set up a GitHub Actions workflow to automatically build a Docker image from a Dockerfile in your GitHub repository and push it to a container registry (e.g., Docker Hub).

Prerequisites:

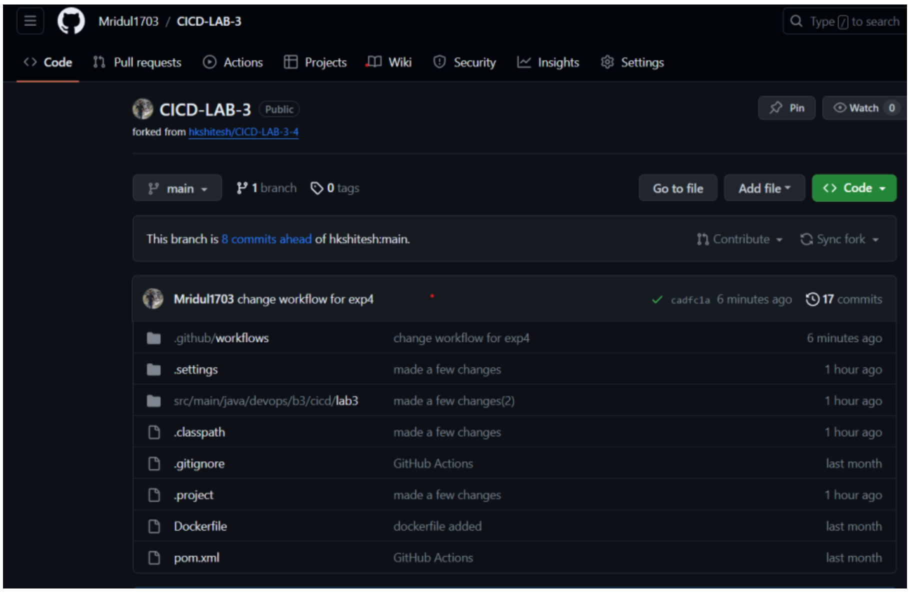
GitHub account

- Docker installed on your local machine
- A Dockerfile in your GitHub repository
- A Docker Hub account (or any other container registry)

Exercise Steps:

Step 1: Fork and Clone the Repository

- Fork a sample GitHub repository containing a Dockerfile or create a new repository and add a Dockerfile to it.
- Clone the forked repository to your local machine.



CICD-LAB-3 Public

forked from hkshitesh/CICD-LAB-3-4

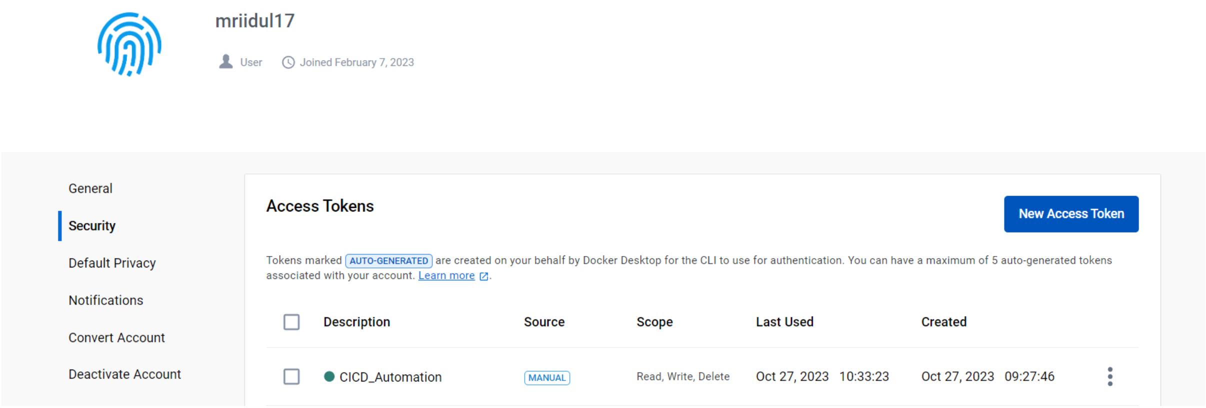
This branch is 8 commits ahead of hkshitesh:main.

Mridul1703 change workflow for exp4 ✓ cadfc1a 6 minutes ago 17 commits

.github/workflows	change workflow for exp4	6 minutes ago
.settings	made a few changes	1 hour ago
src/main/java/devops/b3/cicd/lab3	made a few changes(2)	1 hour ago
.classpath	made a few changes	1 hour ago
.gitignore	GitHub Actions	last month
.project	made a few changes	1 hour ago
Dockerfile	dockerfile added	last month
pom.xml	GitHub Actions	last month

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.



Access Tokens

Tokens marked AUTO-GENERATED are created on your behalf by Docker Desktop for the CLI to use for authentication. You can have a maximum of 5 auto-generated tokens associated with your account. Learn more.

□	Description	Source	Scope	Last Used	Created
□	● CICD_Automation	MANUAL	Read, Write, Delete	Oct 27, 2023 10:33:23	Oct 27, 2023 09:27:46

Step 3: Create a GitHub Actions Workflow

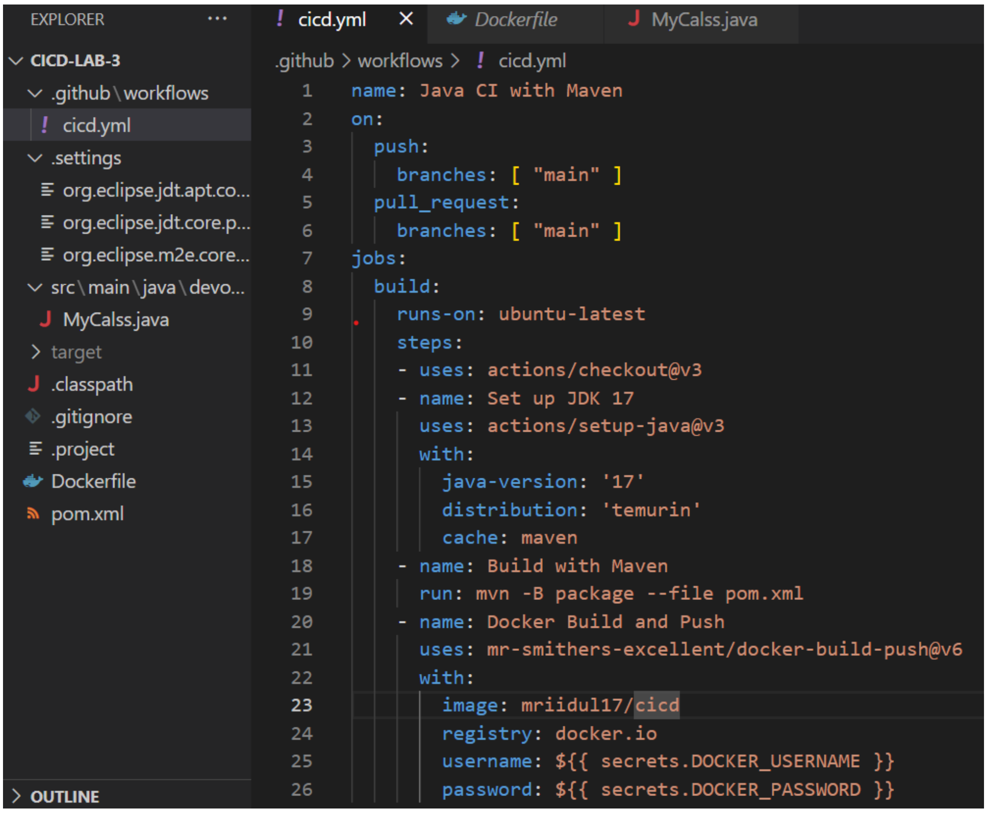
- In your cloned repository, create a directory named .github/workflows if it doesn't exist.

- Inside the .github/workflows directory, create a YAML file (e.g., docker-build-and-push.yml) to define your GitHub Actions workflow. You can use any text editor to create the file.
- Edit docker-build-and-push.yml and add the following content:

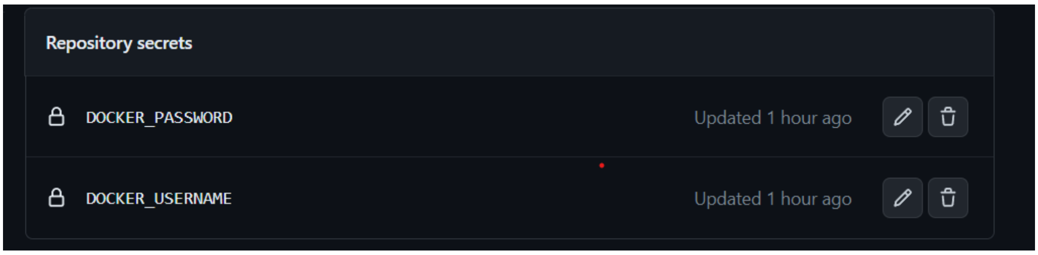
Replace your-dockerhub-username and your-repo-name with your Docker Hub username and repository name.

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.



```yaml
name: Java CI with Maven
on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 17
      uses: actions/setup-java@v3
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml
    - name: Docker Build and Push
      uses: mr-smithers-excellent/docker-build-push@v6
      with:
        image: mriidul17/cicd
        registry: docker.io
        username: ${{ secrets.DOCKER_USERNAME }}
        password: ${{ secrets.DOCKER_PASSWORD }}
```



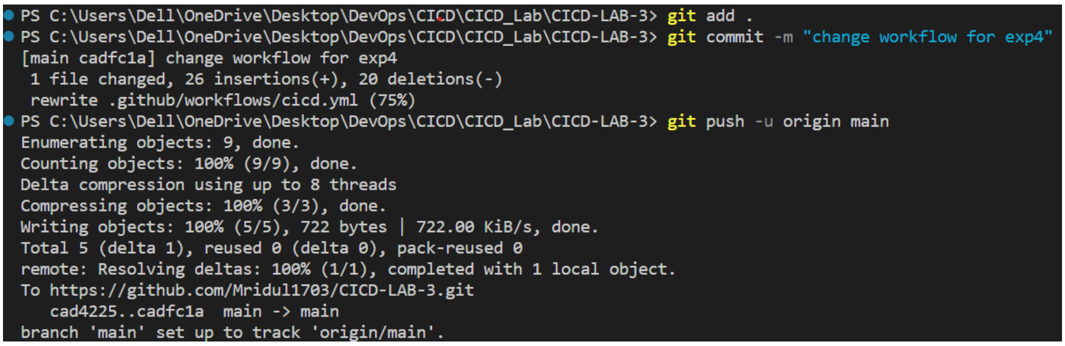
Repository secrets

DOCKER_PASSWORD	Updated 1 hour ago
DOCKER_USERNAME	Updated 1 hour ago

Step 5: Commit and Push Changes

Save the docker-build-and-push.yml file.

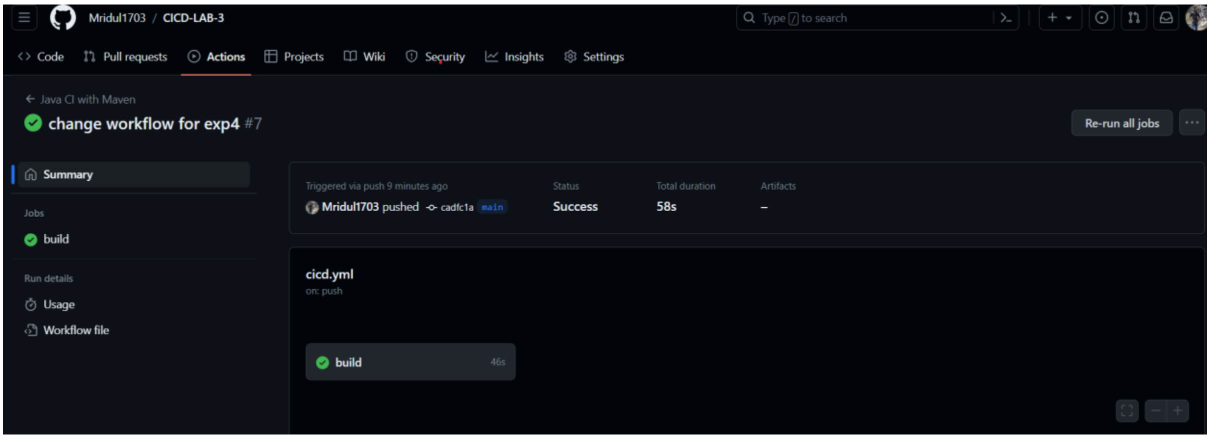
Commit the changes to your local repository:



```
PS C:\Users\Dell\OneDrive\Desktop\DevOps\CICD\CICD_Lab\CICD-LAB-3> git add .
PS C:\Users\Dell\OneDrive\Desktop\DevOps\CICD\CICD_Lab\CICD-LAB-3> git commit -m "change workflow for exp4"
[main cadfc1a] change workflow for exp4
 1 file changed, 26 insertions(+), 20 deletions(-)
 rewrite .github/workflows/cicd.yml (75%)
PS C:\Users\Dell\OneDrive\Desktop\DevOps\CICD\CICD_Lab\CICD-LAB-3> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 722 bytes | 722.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Mridul1703/CICD-LAB-3.git
   cad4225..cadfc1a  main -> main
branch 'main' set up to track 'origin/main'.
```

Step 6: Check the Workflow Status

- Go to your GitHub repository on the GitHub website.
- Click on the "Actions" tab to see the workflow running. You should see a workflow named "Docker Build and Push" or the name you specified in the YAML file.
- Monitor the workflow's progress, and once it completes successfully, you should see a green checkmark indicating a successful build and push of the Docker image to Docker Hub.



← Java CI with Maven

✓ change workflow for exp4 #7

Triggered via push 9 minutes ago: Mridul1703 pushed -o- cadfc1a main

Status: Success

Total duration: 58s

Artifacts: –

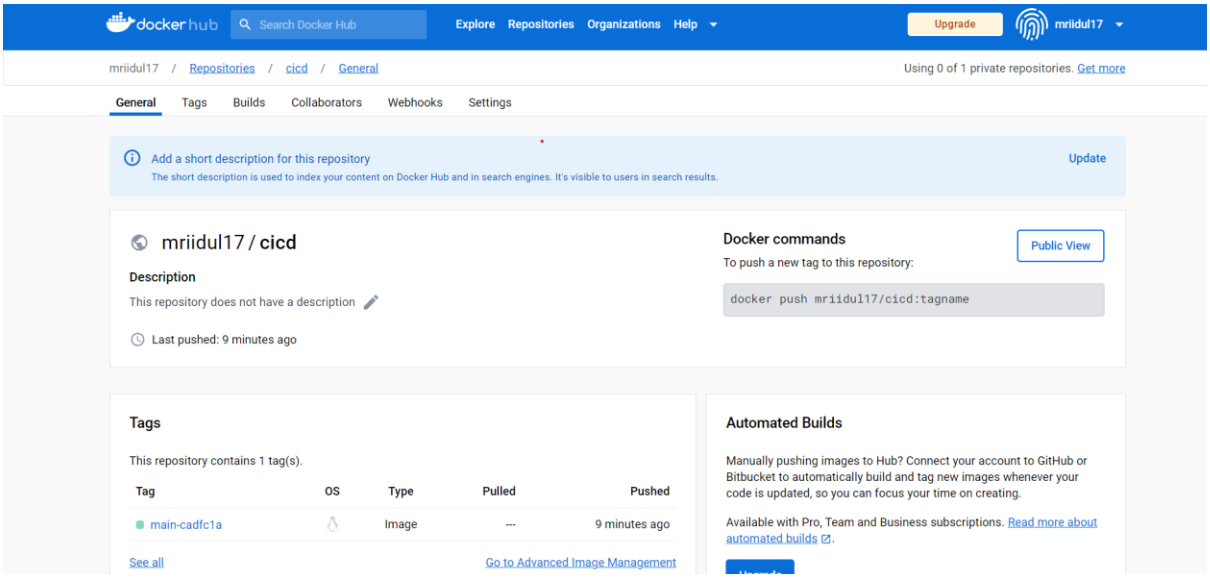
Jobs: ✓ build

cicd.yml
on: push

✓ build 46s

Step 7: Verify the Docker Image on Docker Hub

- Log in to your Docker Hub account.
- Navigate to your Docker Hub repository, and you should see the Docker image you pushed from the GitHub Actions workflow.



mriidul17 / Repositories / cicd / General

Using 0 of 1 private repositories. Get more

Add a short description for this repository
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

mriidul17 / cicd

Description
This repository does not have a description

Last pushed: 9 minutes ago

Docker commands
To push a new tag to this repository:

```
docker push mriidul17/cicd:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
main-cadfc1a		Image	---	9 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.