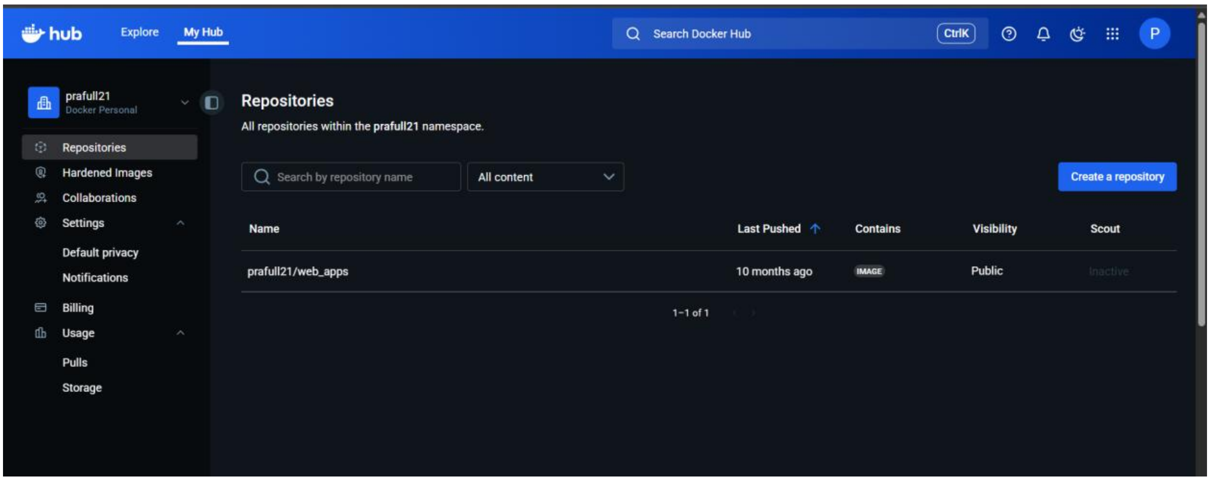
D Y PATIL
INTERNATIONAL
UNIVERSITY
AKURDI PUNE

School of Computer Science, Engineering and Applications(SCSEA)
B.Tech FIY (CCSA)
Subject : Cloud Automation & Devops (P)

Name of the Student: **Pratik.M.Rebari** **PRN: 20220802183**

Title of Practicle : **2. Setting up a Jenkins CI/CD pipeline for a sample project**

Step 1 — Create Docker Hub account.



Repositories

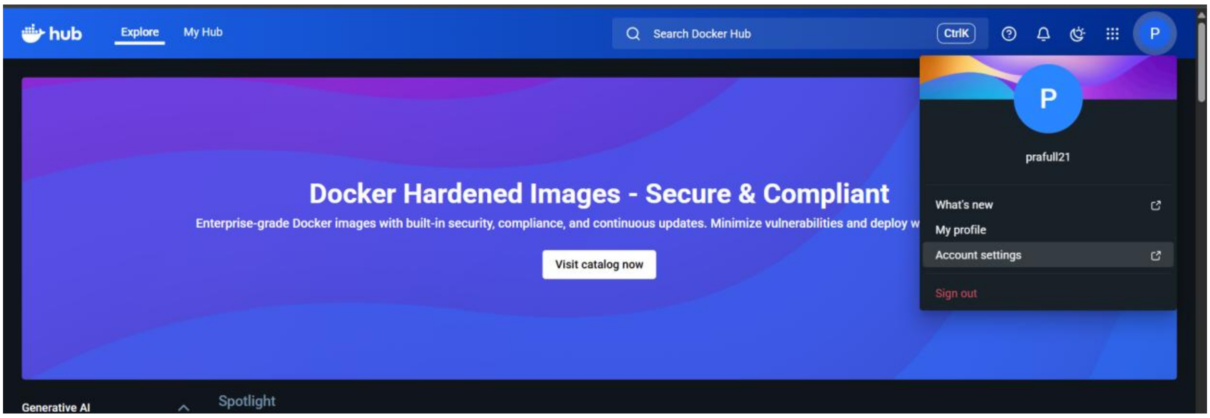
All repositories within the prafull21 namespace.

Name	Last Pushed	Contains	Visibility	Scout
prafull21/web_apps	10 months ago	IMAGE	Public	Inactive

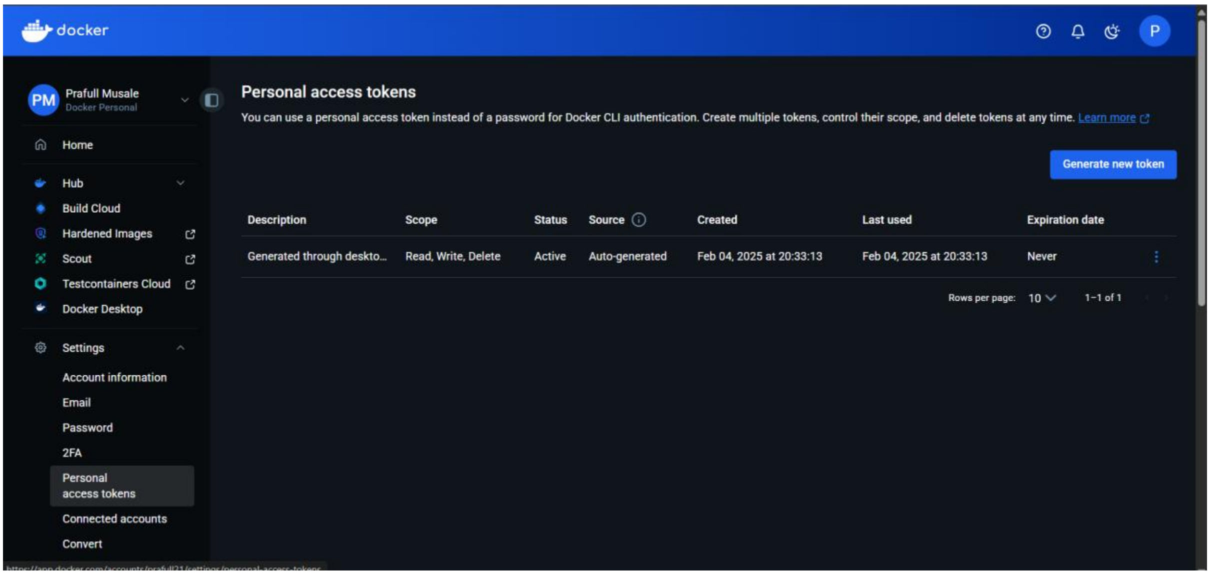
1–1 of 1

Step 2 — Generate Docker Hub Access Token

1. Go to **Account Settings**


hub
Explore
My Hub
Search Docker Hub
CtrlK
P
Docker Hardened Images - Secure & Compliant
Enterprise-grade Docker images with built-in security, compliance, and continuous updates. Minimize vulnerabilities and deploy w
Visit catalog now
prafull21
What's new
My profile
Account settings
Sign out
Generative AI
Spotlight

2. Click **Person Access Token**



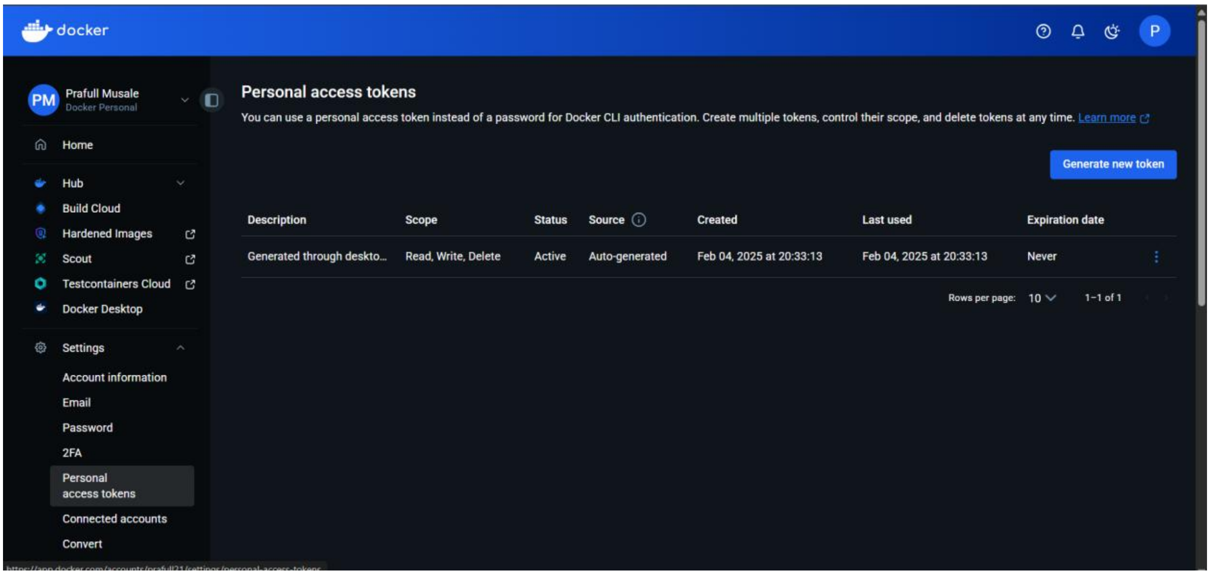
Personal access tokens

You can use a personal access token instead of a password for Docker CLI authentication. Create multiple tokens, control their scope, and delete tokens at any time. Learn more

Description	Scope	Status	Source	Created	Last used	Expiration date
Generated through deskto...	Read, Write, Delete	Active	Auto-generated	Feb 04, 2025 at 20:33:13	Feb 04, 2025 at 20:33:13	Never

Rows per page: 10 1–1 of 1

3. Click **New Access Token**



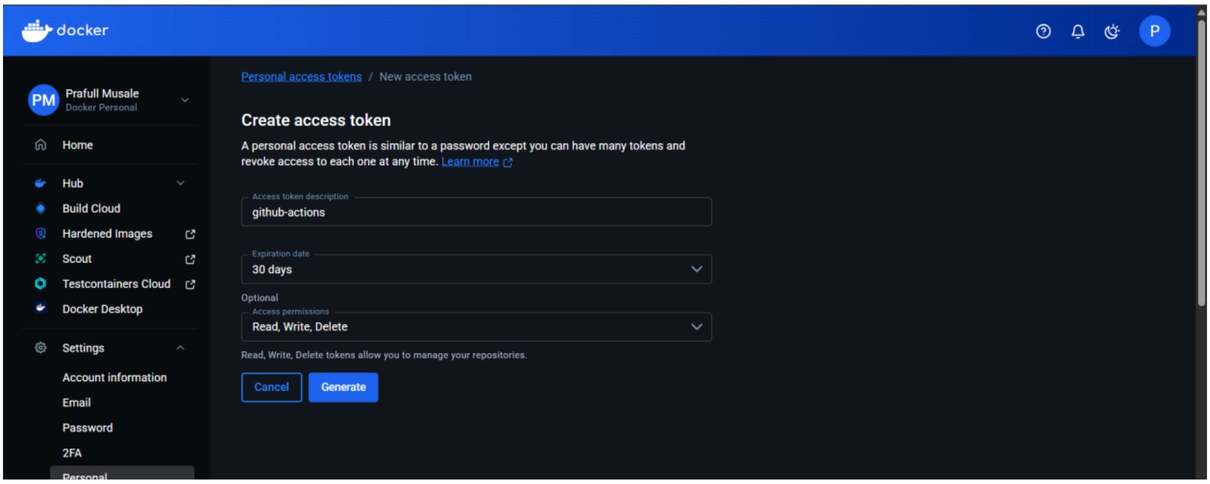
Personal access tokens

You can use a personal access token instead of a password for Docker CLI authentication. Create multiple tokens, control their scope, and delete tokens at any time. Learn more

Description	Scope	Status	Source	Created	Last used	Expiration date
Generated through deskto...	Read, Write, Delete	Active	Auto-generated	Feb 04, 2025 at 20:33:13	Feb 04, 2025 at 20:33:13	Never

Rows per page: 10 1–1 of 1

4. Name: github-actions



Personal access tokens / New access token

Create access token

A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. Learn more

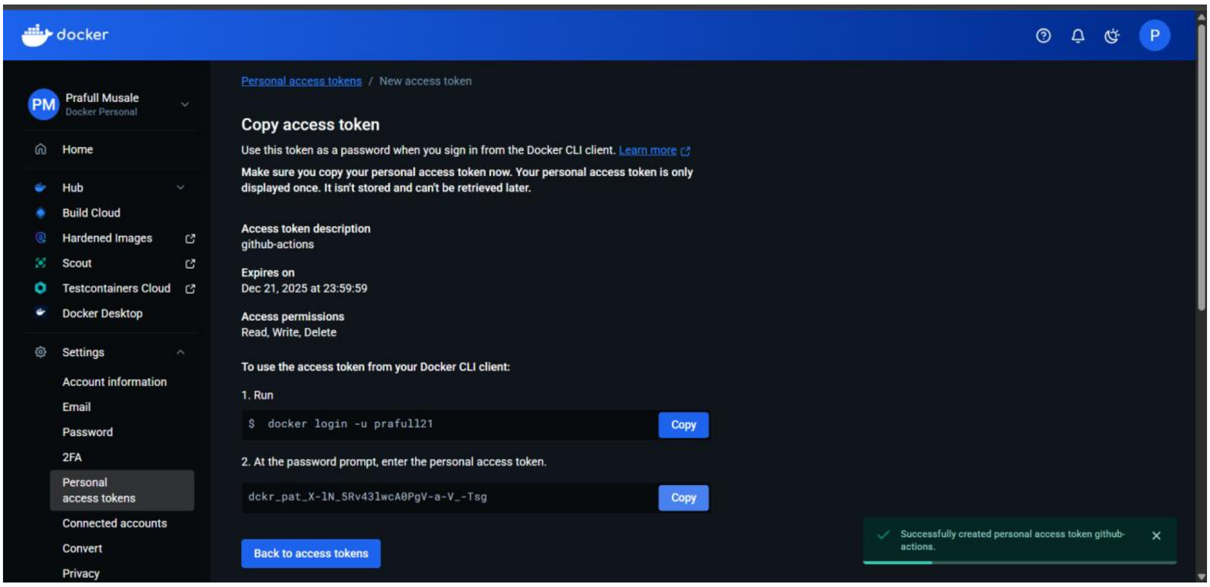
Access token description: github-actions

Expiration date: 30 days

Optional
Access permissions: Read, Write, Delete

Read, Write, Delete tokens allow you to manage your repositories.

5. Copy the token (you will use it soon)



Personal access tokens / New access token

Copy access token

Use this token as a password when you sign in from the Docker CLI client. Learn more

Make sure you copy your personal access token now. Your personal access token is only displayed once. It isn't stored and can't be retrieved later.

Access token description
github-actions

Expires on
Dec 21, 2025 at 23:59:59

Access permissions
Read, Write, Delete

To use the access token from your Docker CLI client:

1. Run

```
$ docker login -u prafull21
```

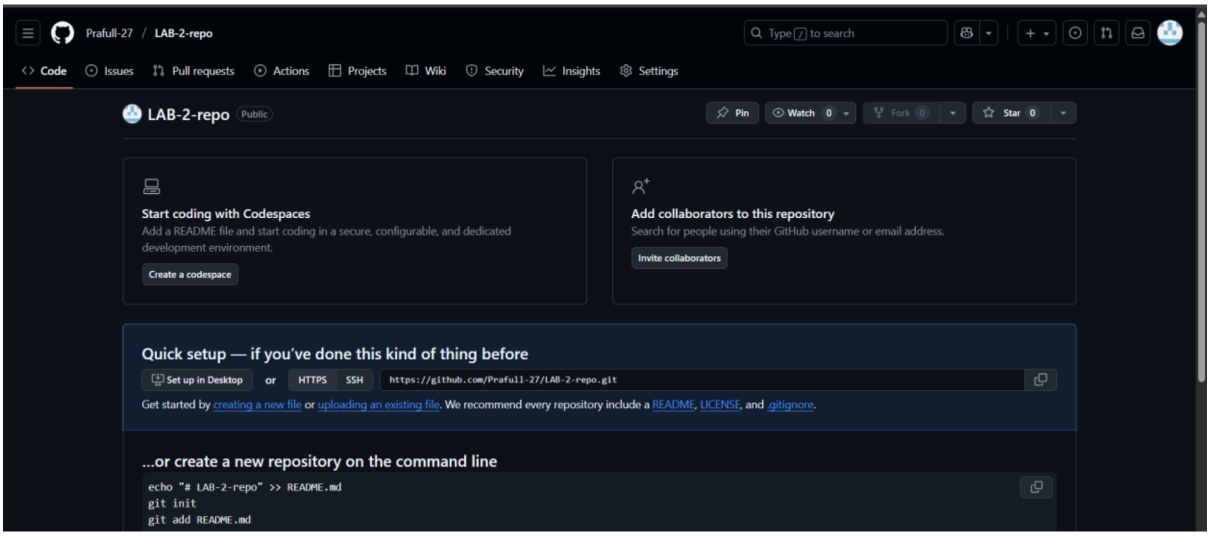
2. At the password prompt, enter the personal access token.

```
dckr_pat_X-1N_5Rv43lwcA0PgV-a-V_-Tsg
```

Successfully created personal access token github-actions.

Step 3 — Add Secrets in GitHub Repository

1. Open your repository



Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.

Add collaborators to this repository
Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

https://github.com/Prafull-27/LAB-2-repo.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# LAB-2-repo" >> README.md
git init
git add README.md
```

2. Go to:
   Settings → Secrets and variables → Actions → New repository secret

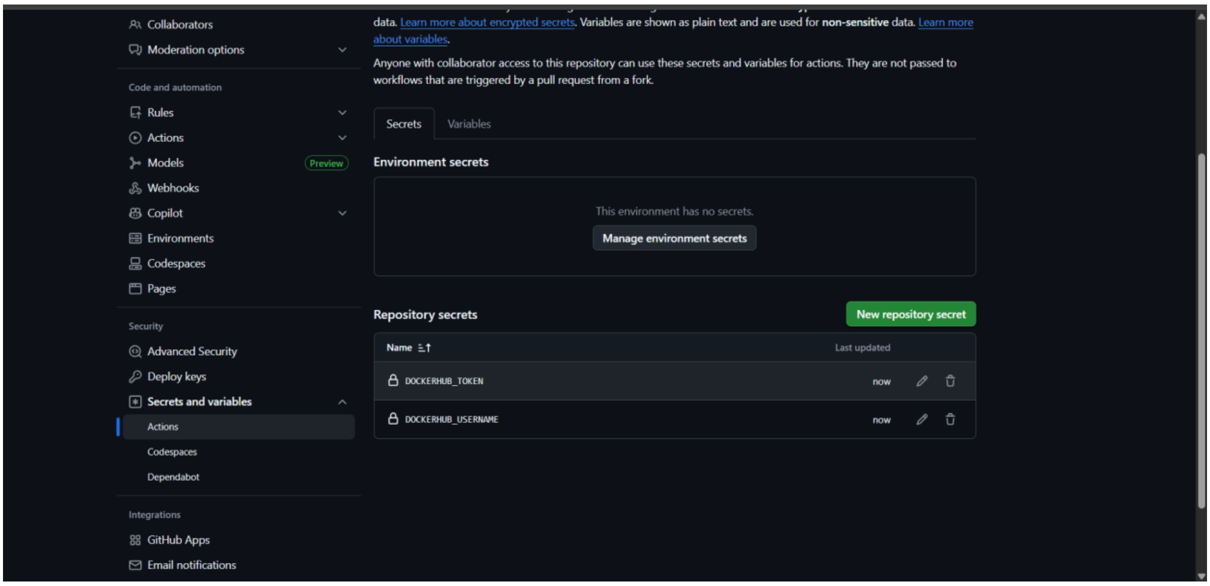
Create two secrets:

Secret 1:

- Name: DOCKERHUB_USERNAME
- Value: your Docker Hub username

Secret 2:

- Name: DOCKERHUB_TOKEN
- Value: the token you generated in Step 2



data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Environment secrets

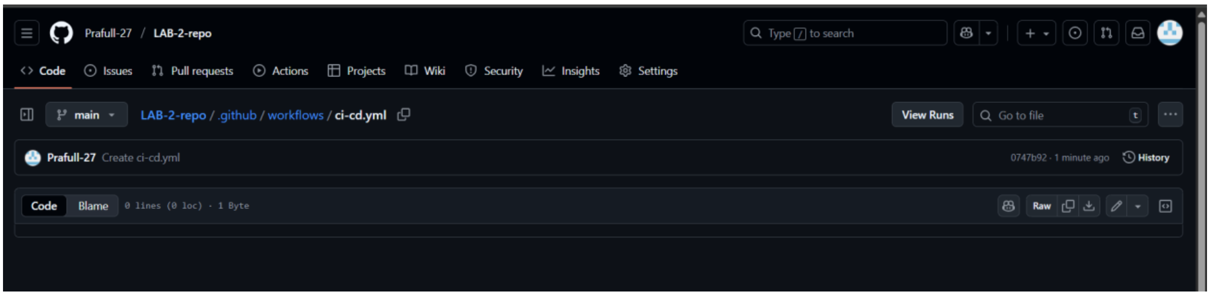
This environment has no secrets.

Repository secrets

Name	Last updated
DOCKERHUB_TOKEN	now
DOCKERHUB_USERNAME	now

Step 4 — Create Workflow Folder

In your repository, create this folder: .github/workflows/



LAB-2-repo / .github / workflows / ci-cd.yml

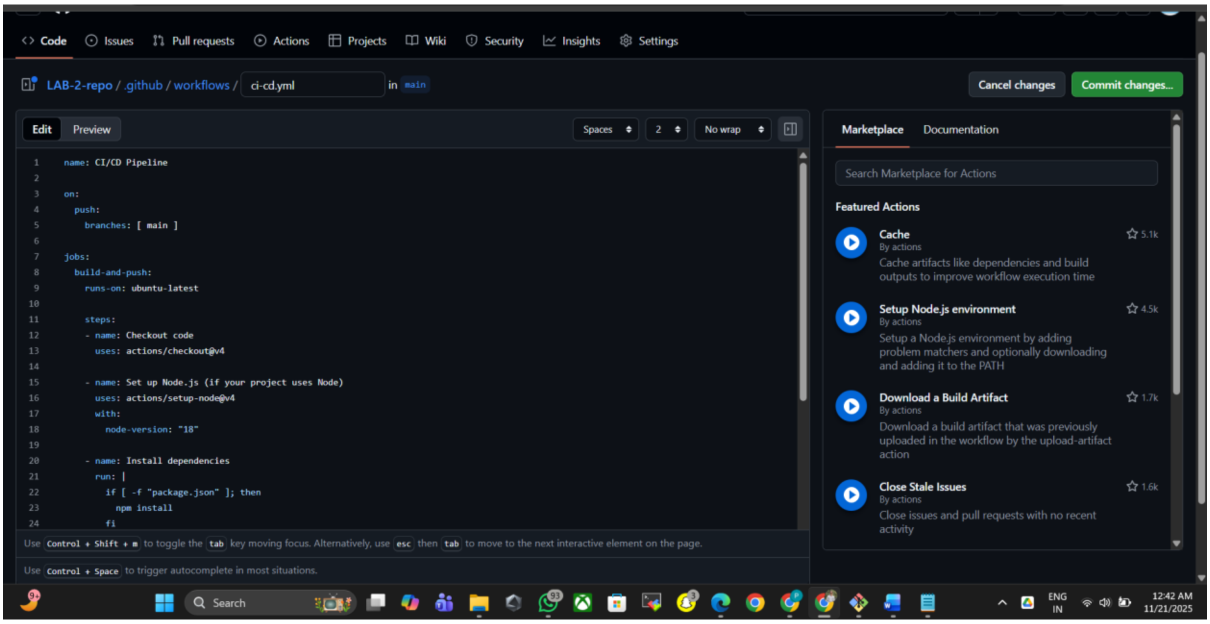
Prafull-27 Create ci-cd.yml 0747b92 · 1 minute ago

0 lines (0 loc) · 1 Byte

Step 5 — Create Workflow File (CI/CD File)

Inside .github/workflows/, create a file

Paste this code inside (simple CI/CD pipeline)



```
name: CI/CD Pipeline

on:
  push:
    branches: [ main ]

jobs:
  build-and-push:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v4

    - name: Set up Node.js (if your project uses Node)
      uses: actions/setup-node@v4
      with:
        node-version: "18"

    - name: Install dependencies
      run: |
        if [ -f "package.json" ]; then
          npm install
        fi
```

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Use Control + Space to trigger autocomplete in most situations.

Featured Actions

Cache — By actions — 5.1k
Cache artifacts like dependencies and build outputs to improve workflow execution time

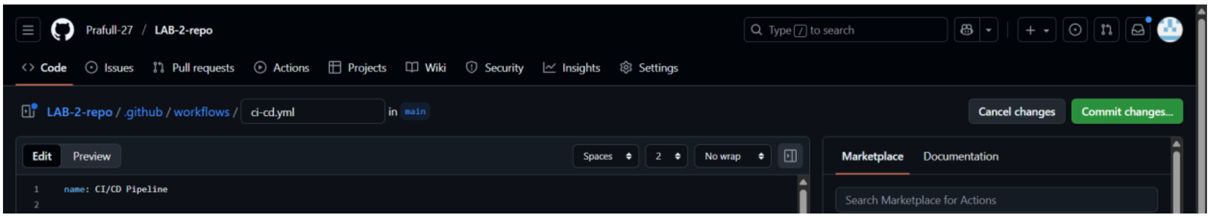
Setup Node.js environment — By actions — 4.5k
Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH

Download a Build Artifact — By actions — 1.7k
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action

Close Stale Issues — By actions — 1.6k
Close issues and pull requests with no recent activity

Step 6 — Commit & Push

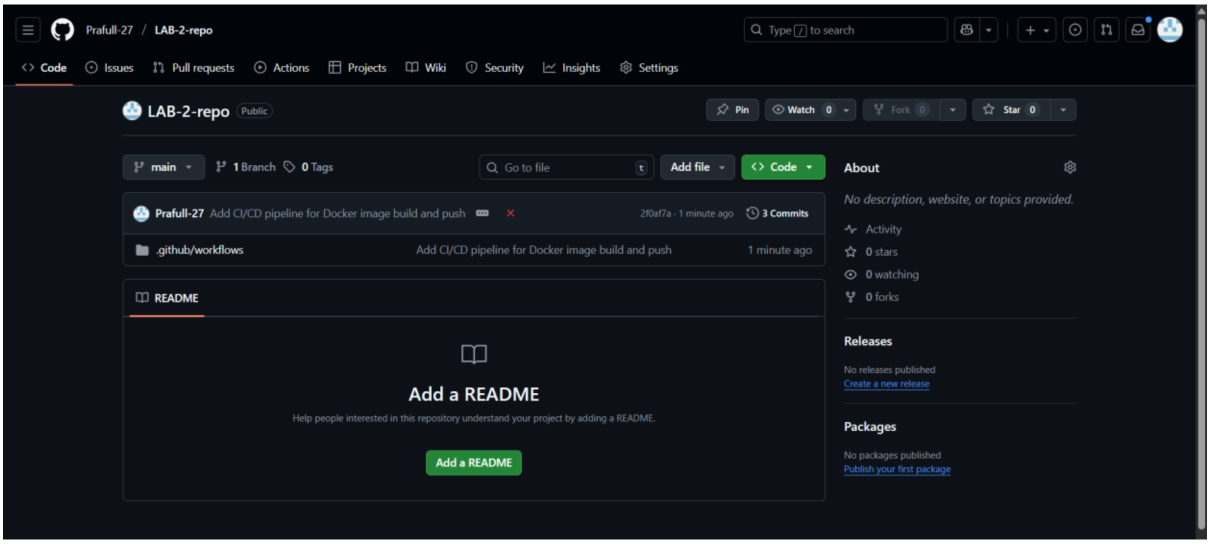
If you push to the **main branch**, the pipeline runs automatically.



```
name: CI/CD Pipeline
```

Step 7 — Check the GitHub Actions Pipeline

1. Go to your repository



Prafull-27 Add CI/CD pipeline for Docker image build and push ✕ 2f0af7a · 1 minute ago 3 Commits

.github/workflows Add CI/CD pipeline for Docker image build and push 1 minute ago

No description, website, or topics provided.

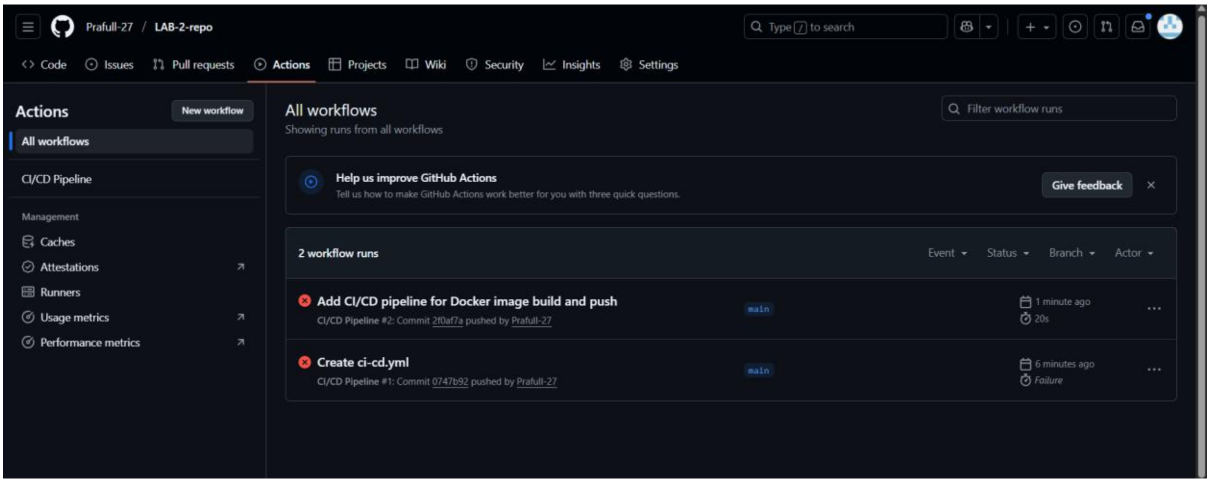
No releases published
Create a new release

No packages published
Publish your first package

Add a README

Help people interested in this repository understand your project by adding a README.

2. Click **Actions**



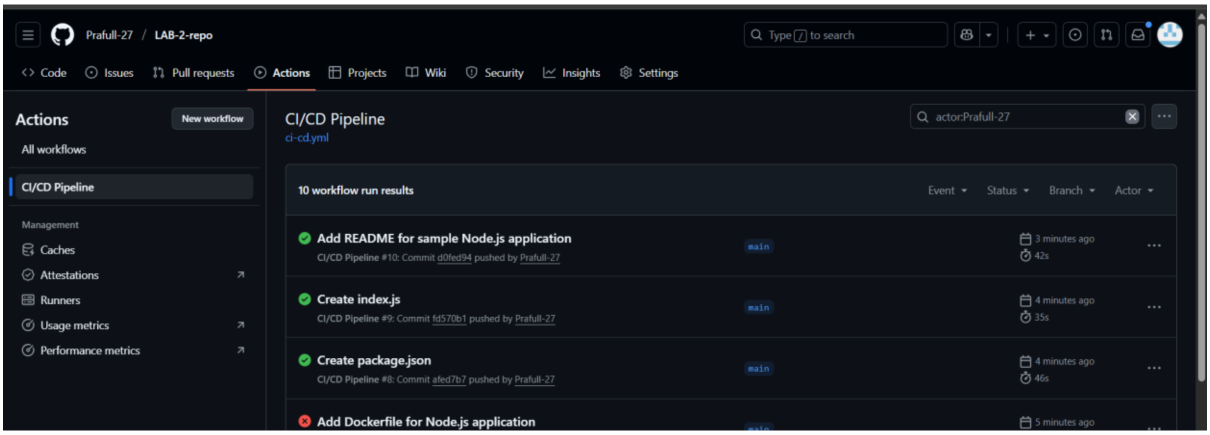
All workflows
Showing runs from all workflows

Help us improve GitHub Actions
Tell us how to make GitHub Actions work better for you with three quick questions.

2 workflow runs

✕ Add CI/CD pipeline for Docker image build and push CI/CD Pipeline #2: Commit 2f0af7a pushed by Prafull-27	main	1 minute ago 20s	…
✕ Create ci-cd.yml CI/CD Pipeline #1: Commit 0747b92 pushed by Prafull-27	main	6 minutes ago Failure	…

3. You will see a new workflow running



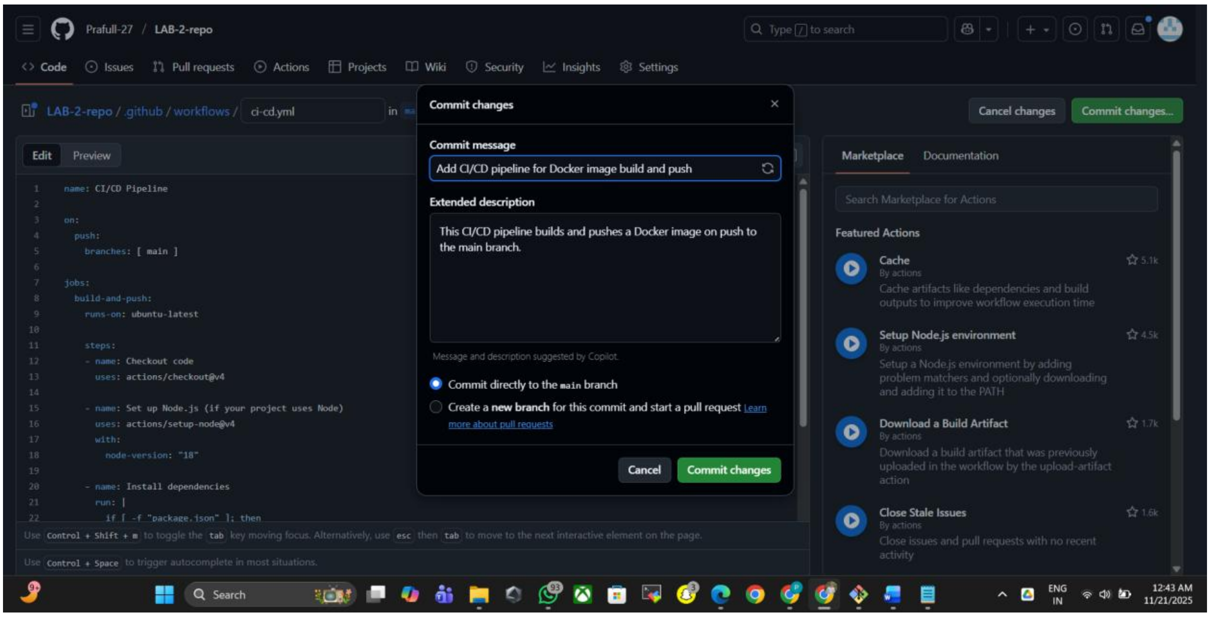
Prafull-27 / LAB-2-repo

Actions — CI/CD Pipeline (ci-cd.yml)

10 workflow run results

- ✓ Add README for sample Node.js application — CI/CD Pipeline #10: Commit d0fed94 pushed by Prafull-27 — main — 3 minutes ago — 42s
- ✓ Create index.js — CI/CD Pipeline #9: Commit fd570b1 pushed by Prafull-27 — main — 4 minutes ago — 35s
- ✓ Create package.json — CI/CD Pipeline #8: Commit afed7b7 pushed by Prafull-27 — main — 4 minutes ago — 46s
- ✗ Add Dockerfile for Node.js application — main — 5 minutes ago

4. Click it → you can see all logs: build, test, docker push, etc.



LAB-2-repo / .github / workflows / ci-cd.yml

```
name: CI/CD Pipeline

on:
  push:
    branches: [ main ]

jobs:
  build-and-push:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v4

    - name: Set up Node.js (if your project uses Node)
      uses: actions/setup-node@v4
      with:
        node-version: "18"

    - name: Install dependencies
      run: |
        if [ -f "package.json" ]; then
```

Commit changes

Commit message: Add CI/CD pipeline for Docker image build and push

Extended description: This CI/CD pipeline builds and pushes a Docker image on push to the main branch.

Message and description suggested by Copilot.

● Commit directly to the main branch
○ Create a new branch for this commit and start a pull request Learn more about pull requests

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Use Control + Space to trigger autocomplete in most situations.

Edit
Preview
Cancel
Commit changes
Cancel changes
Commit changes...
Marketplace
Documentation
Search Marketplace for Actions
Featured Actions
Cache
Cache artifacts like dependencies and build outputs to improve workflow execution time
Setup Node.js environment
Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH
Download a Build Artifact
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action
Close Stale Issues
Close issues and pull requests with no recent activity
Search
12:43 AM
11/21/2025

Step 8 — Confirm Docker Image in Docker Hub

Go to Docker Hub → Repositories →
You will see:

- sample-app:latest
- sample-app:<build-number>