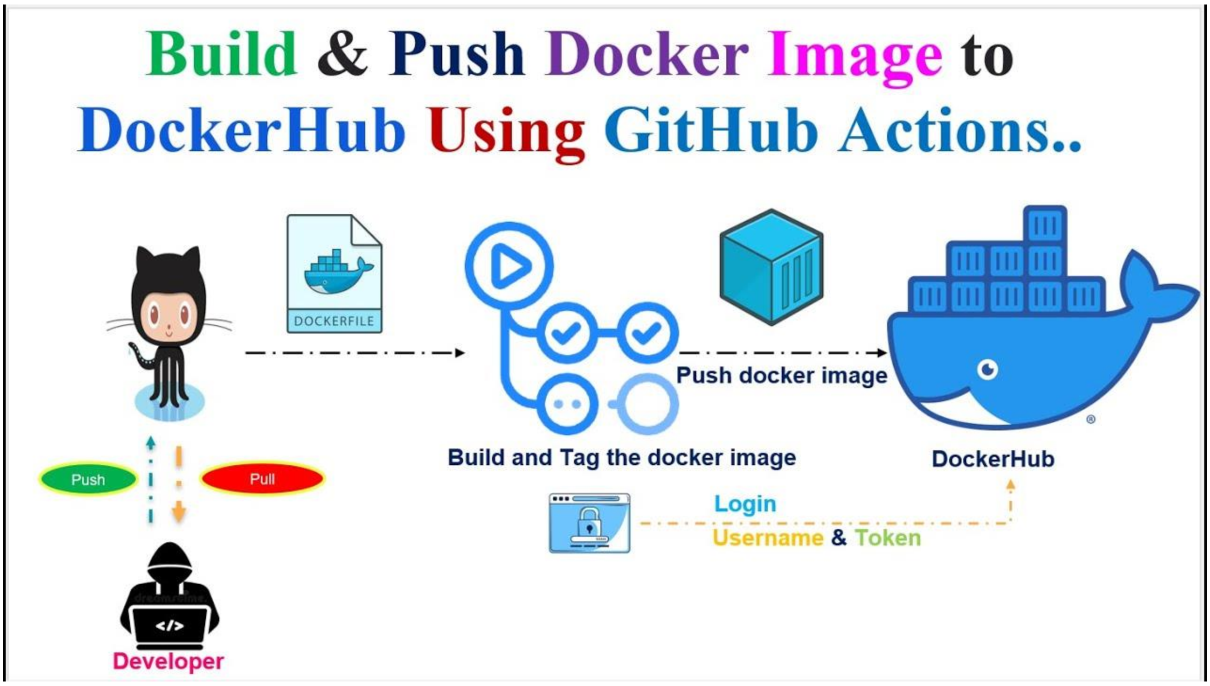
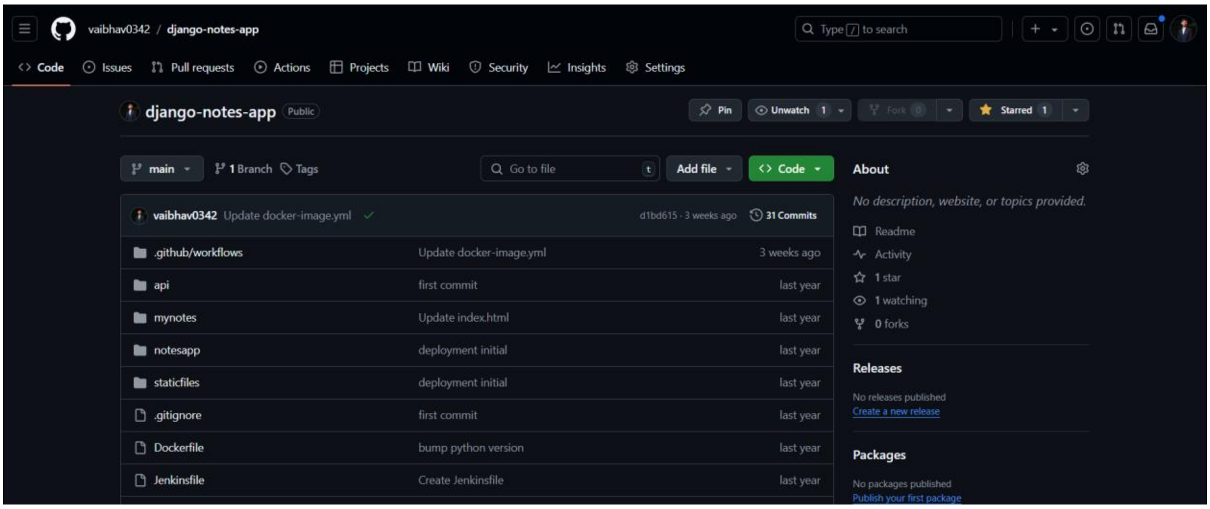


Build & Push Docker Image to DockerHub Using GitHub Actions..

https://github.com/vaibhav0342

Step 1: Repository for GitHub Actions

1. **Open Your GitHub Repository**: Navigate to the repository where your Docker project is stored. Ex: https://github.com/vaibhav0342/django-notes-app



vaibhav0342 Update docker-image.yml ✓ d1bd615 · 3 weeks ago 31 Commits

.github/workflows	Update docker-image.yml	3 weeks ago
api	first commit	last year
mynotes	Update index.html	last year
notesapp	deployment initial	last year
staticfiles	deployment initial	last year
.gitignore	first commit	last year
Dockerfile	bump python version	last year
Jenkinsfile	Create Jenkinsfile	last year

About

No description, website, or topics provided.

- Readme
- Activity
- 1 star
- 1 watching
- 0 forks

Releases

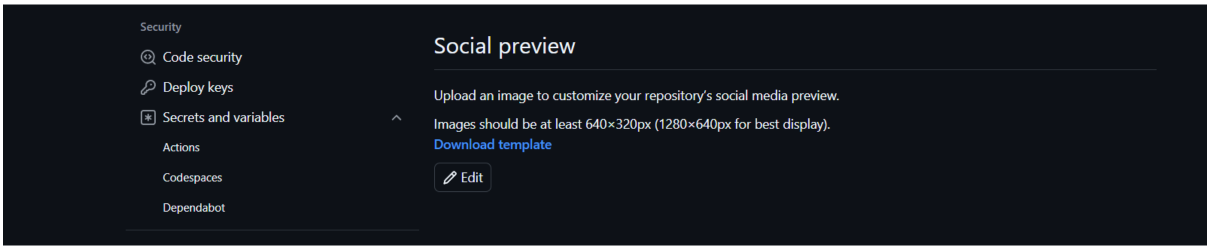
No releases published
Create a new release

Packages

No packages published
Publish your first package

Step 2: Set Up Secrets for Docker Hub

1. **Navigate to Repository Settings**: In your GitHub repository, go to **Settings > Secrets and variables > Actions**.

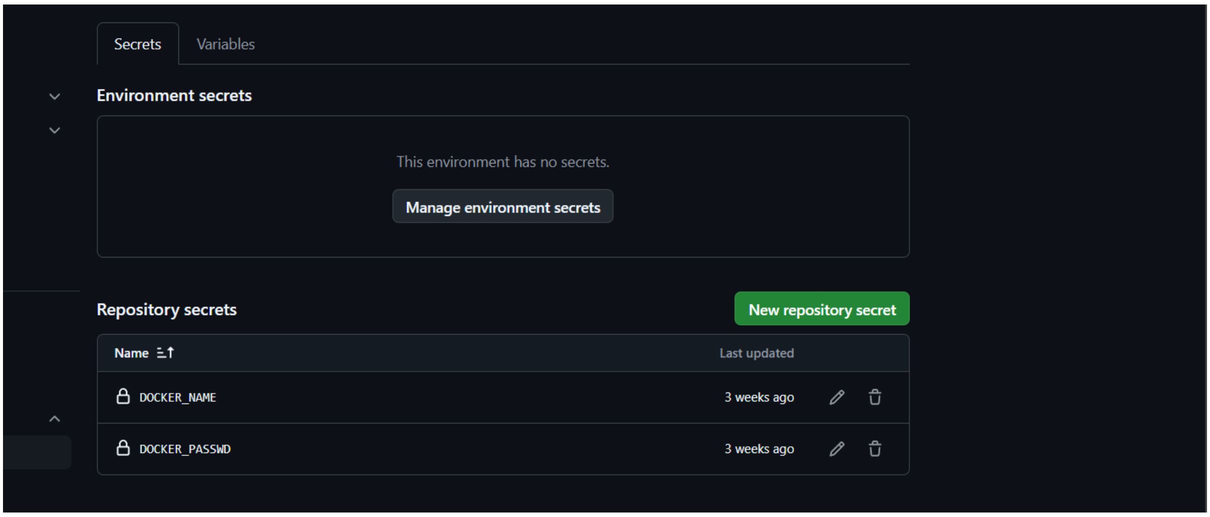


Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

2. **Add Docker Hub Credentials**:
   - Click on **New repository secret**.
   - Add two secrets:
     - `DOCKER_NAME`: Your Docker Hub username.
     - `DOCKER_PASSWD`: Your Docker Hub password or access token.



Environment secrets

This environment has no secrets.

Repository secrets

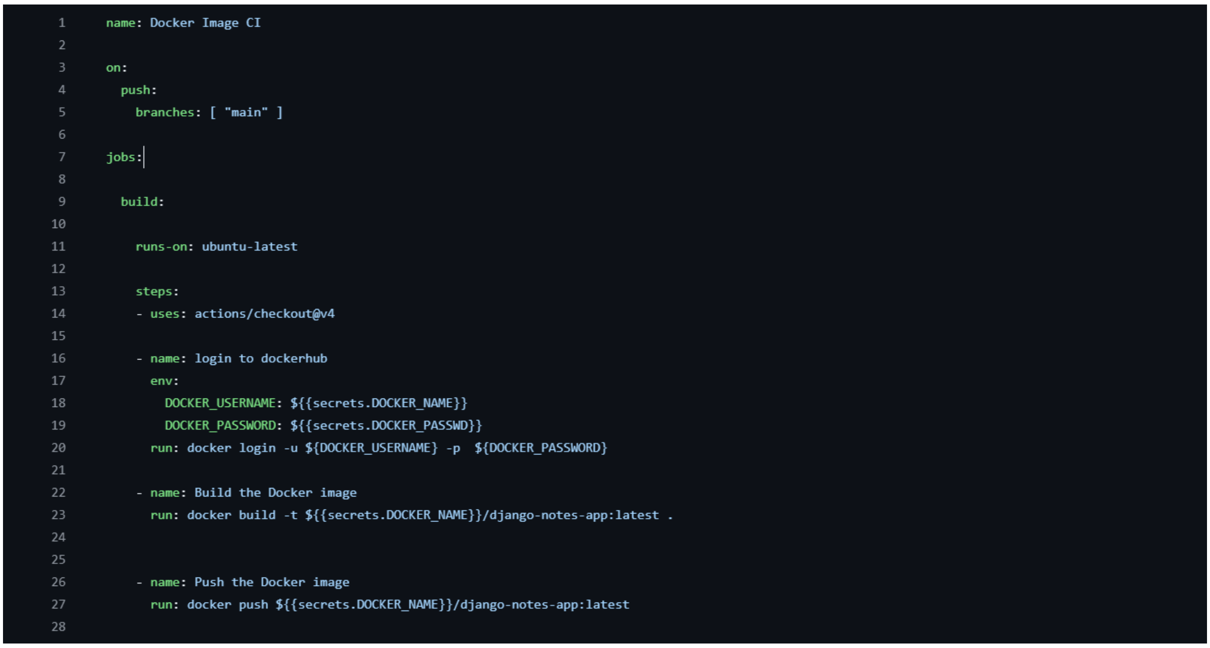
Name	Last updated
DOCKER_NAME	3 weeks ago
DOCKER_PASSWD	3 weeks ago

Step 3: Define the GitHub Actions Workflow

1. **Create a `.github/workflows` Directory**: Inside your repository, create a folder named `.github` and then a subfolder named `workflows`.

2. **Create a Workflow YAML File**: Inside the `workflows` folder, create a YAML file like `docker-image.yml`. This file will hold the configuration for your GitHub Actions workflow.

In the `docker-image.yml` file, you'll write the YAML code that defines your workflow:



```yaml
name: Docker Image CI

on:
  push:
    branches: [ "main" ]

jobs:

  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v4

    - name: login to dockerhub
      env:
        DOCKER_USERNAME: ${{secrets.DOCKER_NAME}}
        DOCKER_PASSWORD: ${{secrets.DOCKER_PASSWD}}
      run: docker login -u ${DOCKER_USERNAME} -p  ${DOCKER_PASSWORD}

    - name: Build the Docker image
      run: docker build -t ${{secrets.DOCKER_NAME}}/django-notes-app:latest .


    - name: Push the Docker image
      run: docker push ${{secrets.DOCKER_NAME}}/django-notes-app:latest

```

Step 4: Explanation of Each Step in the Workflow

- **Trigger**: The workflow runs on pushes to the `main` branch (`on: push`), ensuring only main branch changes trigger the automation.

- **Checkout Code**: The `actions/checkout@v4` action pulls the latest code from the repository to the GitHub runner, preparing it for the build.

- **Login to Docker Hub**:

  - Uses environment variables (`DOCKER_NAME` and `DOCKER_PASSWD`), stored as secrets, to securely log into Docker Hub.

- **Build Docker Image**:

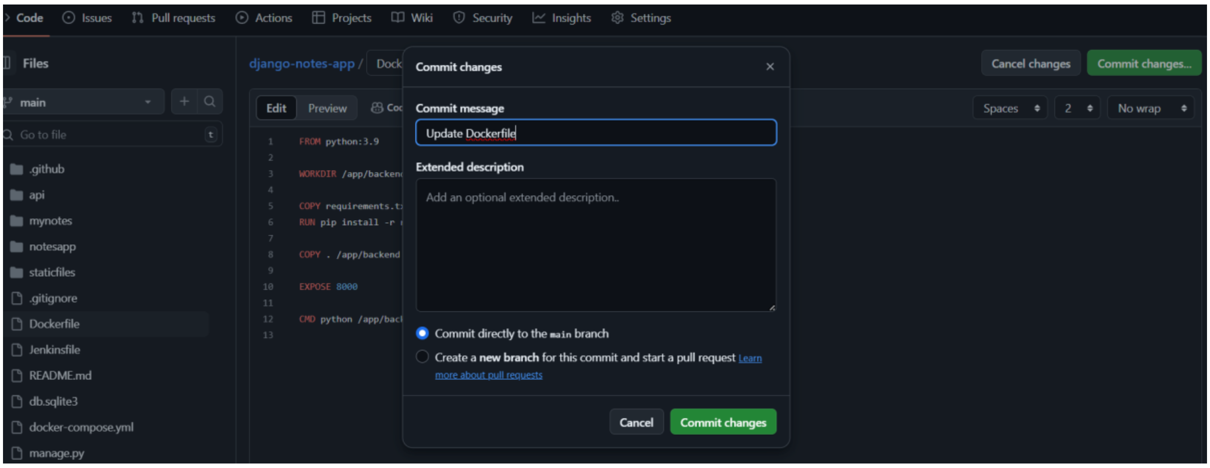
  - Builds a Docker image with the tag `DOCKER_NAME/django-notes-app:latest`, naming it based on the Docker Hub username and project.

- **Push Docker Image**:

  - Pushes the built image to Docker Hub, making the updated version available for use in deployments.

Step 5: Commit and Push the Dockerfile.

1. Save your changes to `Dockerfile`.
2. Commit and push this file to the `main` branch of your repository.



```
FROM python:3.9

WORKDIR /app/backend

COPY requirements.t
RUN pip install -r 

COPY . /app/backend

EXPOSE 8000

CMD python /app/bac
```

Commit changes

Commit message: Update Dockerfile

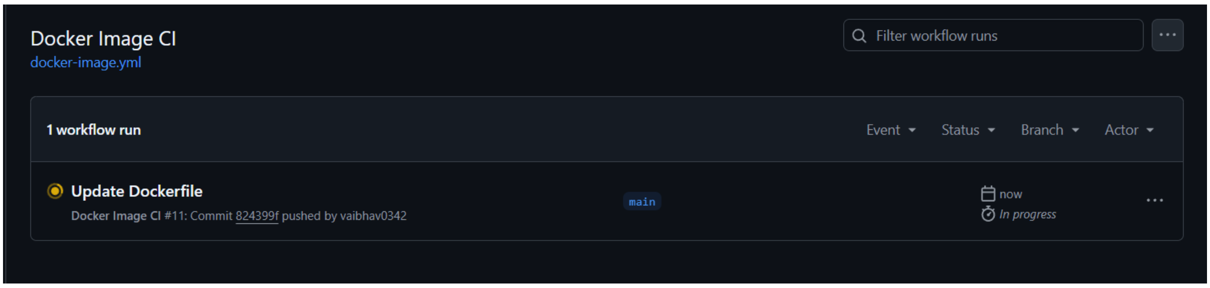
Extended description: Add an optional extended description..

● Commit directly to the `main` branch

○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

Step 6: Monitor Your Workflow

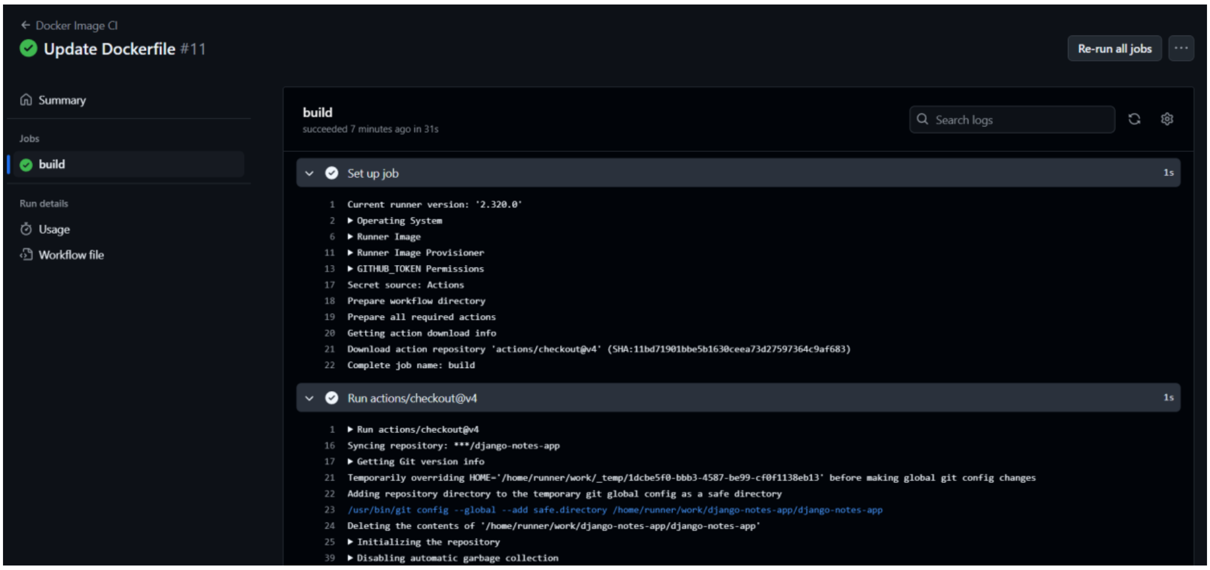
1. Go to the **Actions** tab in your repository.



1 workflow run

Update Dockerfile
Docker Image CI #11: Commit 824399f pushed by vaibhav0342
main
now
In progress

2. Find and select the `Docker Image CI/CD` workflow.
3. Observe the progress, logs, and ensure all steps complete successfully.



build
succeeded 7 minutes ago in 31s

Set up job 1s

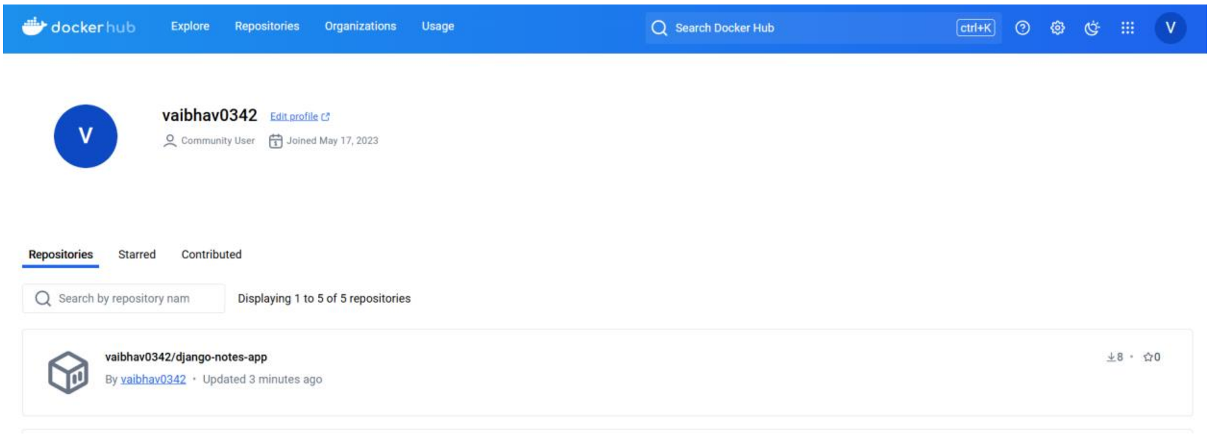
```
Current runner version: '2.320.0'
▶ Operating System
▶ Runner Image
▶ Runner Image Provisioner
▶ GITHUB_TOKEN Permissions
Secret source: Actions
Prepare workflow directory
Prepare all required actions
Getting action download info
Download action repository 'actions/checkout@v4' (SHA:11bd71901bbe5b1630ceea73d27597364c9af683)
Complete job name: build
```

Run actions/checkout@v4 1s

```
▶ Run actions/checkout@v4
Syncing repository: ***/django-notes-app
▶ Getting Git version info
Temporarily overriding HOME='/home/runner/work/_temp/1dcbe5f0-bbb3-4587-be99-cf0f1138eb13' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/django-notes-app/django-notes-app
Deleting the contents of '/home/runner/work/django-notes-app/django-notes-app'
▶ Initializing the repository
▶ Disabling automatic garbage collection
```

Step 7: Check Docker Image:

Pushes the built image to Docker Hub, making the updated version available



vaibhav0342

Community User · Joined May 17, 2023

Displaying 1 to 5 of 5 repositories

vaibhav0342/django-notes-app

By vaibhav0342 · Updated 3 minutes ago

8 · 0