

TASK-1. Automate Code Deployment Using CI/CD Pipeline (GitHub Actions)

Objective:

Set up GitHub Actions to:

1. Run tests
2. Build a Docker image
3. Push it to DockerHub

Tools Involved:

1. GitHub Actions: Automate your build, test, and deployment steps.
2. Docker: Package your app into a container.
3. Docker Hub: Store and share your container images.
4. Node.js: (app.js) – the application being deployed.

What Happens in the Pipeline? Trigger:

The pipeline starts automatically on a git push to the main branch.

1. Checkout Code: Pulls your source code into the GitHub Actions runner.
2. Install Dependencies: Runs npm install to set up the app.
3. Run Tests: Runs npm test to ensure everything is working.
4. Build Docker Image: Builds a Docker image from the Dockerfile.
5. Login to DockerHub: Uses your DockerHub credentials stored in GitHub Secrets.
6. Push Image to DockerHub: Pushes the image to your DockerHub repository for deployment or sharing.

Prerequisites:

Before you begin:

1. Have a GitHub account
2. Have a DockerHub account
3. Create a GitHub repo for your Node.js app
4. Generate a DockerHub access token: <https://hub.docker.com/settings/security>

Project Structure (Sample Node.js App)

```
bash

nodejs-demo-app/
├── app.js
├── Dockerfile
├── package.json
├── package-lock.json
└── .github/
    └── workflows/
        └── main.yml
```

GitHub Secrets Required

Go to your repo → Settings → Secrets and variables → Actions:

- DOCKER_USERNAME — your DockerHub username # give here docker hub username (imp: same name DOCKER_USERNAME given in inside of main.yml .github/workflows/main.yml)
- DOCKER_PASSWORD — your DockerHub Access Token # give here docker hub PAT(personal access token) (imp: same name DOCKER_PASSWORD given in inside of main.yml .github/workflows/main.yml)

GitHub Actions CI/CD: .github/workflows/main.yml

name: CI/CD Pipeline

on:

push:

 branches:

 - main

 workflow_dispatch: # this is use for enable run pipeline Button

jobs:

 build-test-push:

 name: Build, Test & Push Docker Image

 runs-on: ubuntu-latest

steps:

- name:  Checkout Code

uses: actions/checkout@v3

- name:  Set up Node.js

uses: actions/setup-node@v4

with:

node-version: '18'

- name:  Install Dependencies

run: npm install

- name:  Run Tests

run: npm test

- name:  Log in to DockerHub

uses: docker/login-action@v3

with:

username: \${{ secrets.DOCKER_USERNAME }}

password: \${{ secrets.DOCKER_PASSWORD }}

- name:  Build Docker Image

run: |

```
docker build -t ${{ secrets.DOCKER_USERNAME }}/nodejs-demo-app:latest .
```

- name:  Push Docker Image to DockerHub

run: |

```
docker push ${ secrets.DOCKER_USERNAME }}/nodejs-demo-app:latest
```

After code is ready in locally. Execute so cmd:

- Git init
- git config --global user.name ""
- git config --global user.email ""
- git status
- git add .
- git commit -m "message"
- git branch -M main # change to main branch
- git remote add origin <git repo URL>
- git push -u origin main

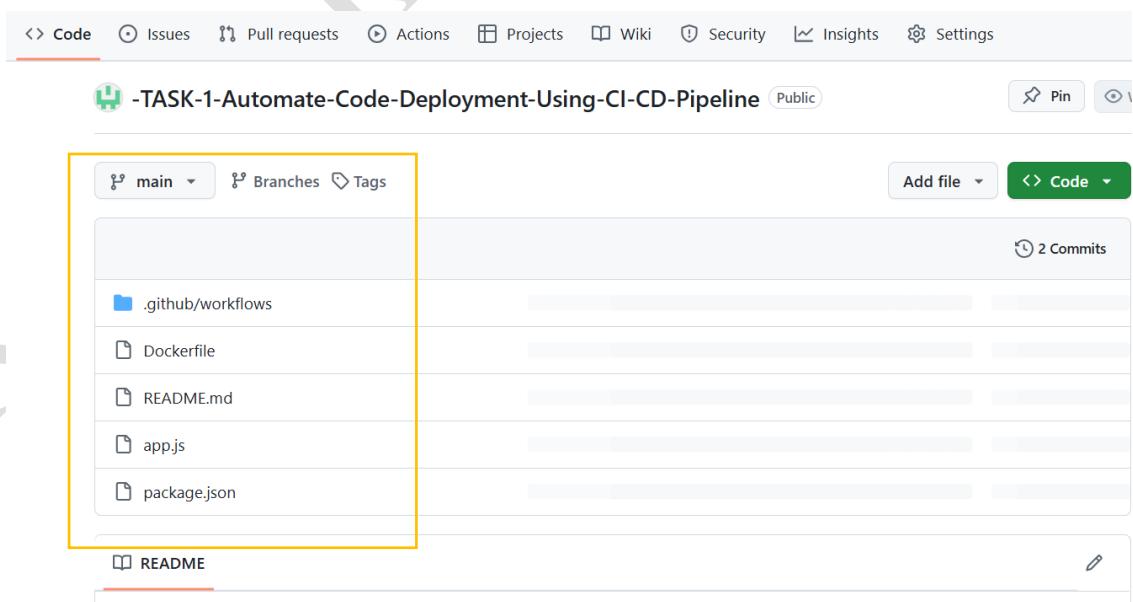
#check remote repo the local files are reached or not

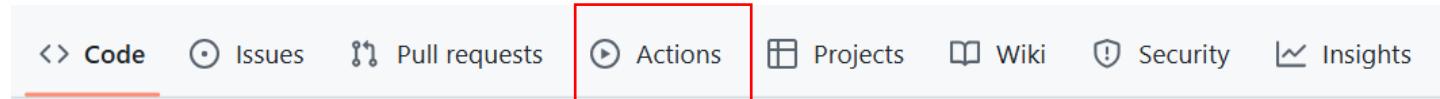
Go to your repo → Settings → Secrets and variables → Actions ---> Repository secrets

Add:

DOCKER_USERNAME = your DockerHub username # give here docker hub username (imp: same name DOCKER_USERNAME given in inside of main.yml .github/workflows/main.yml)

DOCKER_PASSWORD = your DockerHub access token # give here docker hub PAT(personal access token) (imp: same name DOCKER_PASSWORD given in inside of main.yml .github/workflows/main.yml)





-TASK-1-Automate-Code-Deployment-Using-CI-CD-Pipeline Public

CI/CD Pipeline
main.yml

1 workflow run

This workflow has a `workflow_dispatch` event trigger.

sample-nodejs-app

CI/CD Pipeline #1: Commit [8560f33](#) pushed by Ramakrishnaragi

main 16 minutes ago 29s

← CI/CD Pipeline **sample-nodejs-app #1** Re-run all jobs

Summary

Jobs **build-test-push**

Run details

Usage

Workflow file

build-test-push

succeeded 6 minutes ago in 22s

Search logs

Set up job

Checkout code

Set up Node.js

Install dependencies

Run tests

Log in to DockerHub

Build Docker image

Push Docker image to DockerHub

Post Log in to DockerHub

This code is successfully executed.

#Go to docker hub, check the docker hub your image is present in

Docker Personal

Repositories

All repositories within th namespace.

Search by repository name All content

Create a repository

Name	Last Pushed	Contains	Visibility	Scout
/nodejs-demo-app	11 minutes ago	IMAGE	Public	Inactive

go to docker desktop

Ask Gordon BETA

Containers

Images

Volumes

Builds

Docker Hub

Docker Scout

Extensions

Images

Local Docker Hub repositories

Search

Tags	OS	Vulnerabilities	Last pushed	Size
latest			18 minutes ago	45.62 MB

click pull the image

move to Local in image section

Run a new container

/nodejs-demo-app:latest

Optional settings

Container name: nodejs-demo-app

Ports

Host port: 80 Container port: :3000/tcp

Volumes

Environment variables

Name: nodejs-demo-app

Cancel Run

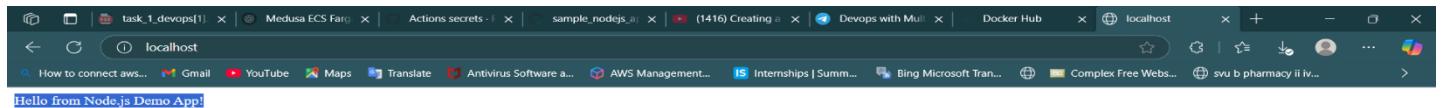
#After give run the container create, and copy the url <lo

Logs Inspect Bind mounts Exec Files Stats

```
2025-04-07 21:23:44
2025-04-07 21:23:44 > nodejs-demo-app@1.0.0 start
2025-04-07 21:23:44 > node app.js
2025-04-07 21:23:44
2025-04-07 21:23:45 App listening at http://localhost:3000
```

#finally access the application

#give like this http://localhost:80



Goals: Automatically build, test, and deploy your application every time you push code to your repository.