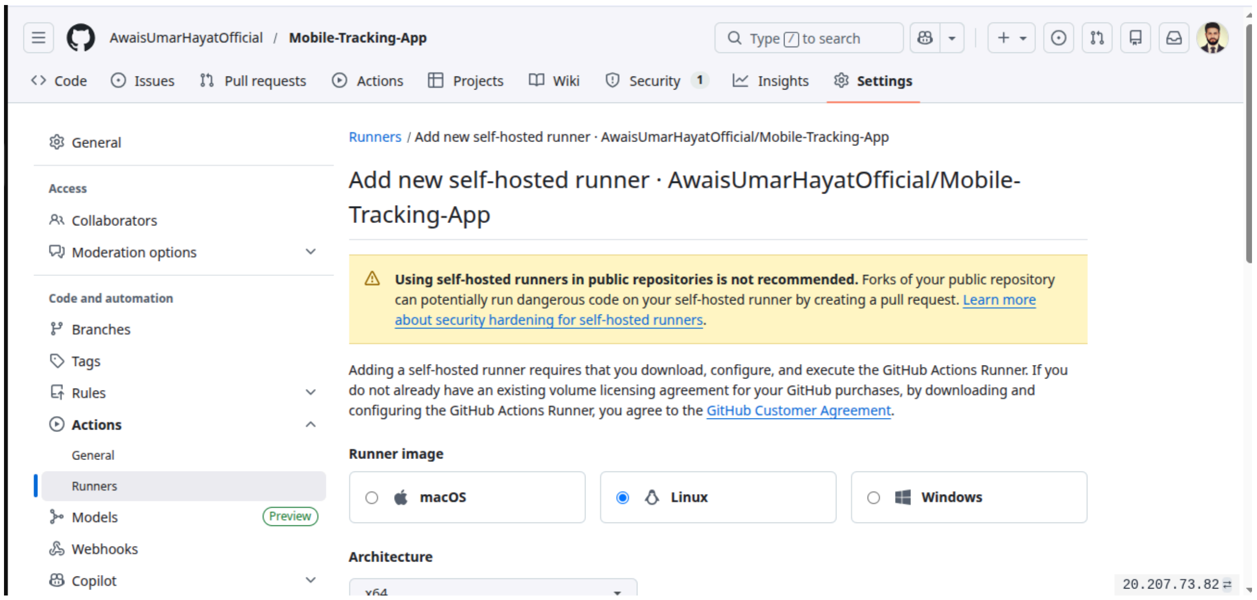
GitHub Actions CICD for Kubernetes AutoDeployment

This guide walks you through setting up a **self-hosted GitHub Actions runner** on Ubuntu and integrating it with **Docker Hub** using a Personal Access Token (PAT) for CI/CD workflows.

Setting up a GitHub Actions Self-Hosted Runner

Step 1: Create a New Self-Hosted Runner in GitHub

1. Go to your repository on GitHub.
2. Navigate to:
   Settings → Actions → Runners → New self-hosted runner
3. Follow the instructions provided for **Ubuntu**.
   You will receive a set of commands to download and configure the runner on your machine.



Runners / Add new self-hosted runner · AwaisUmarHayatOfficial/Mobile-Tracking-App

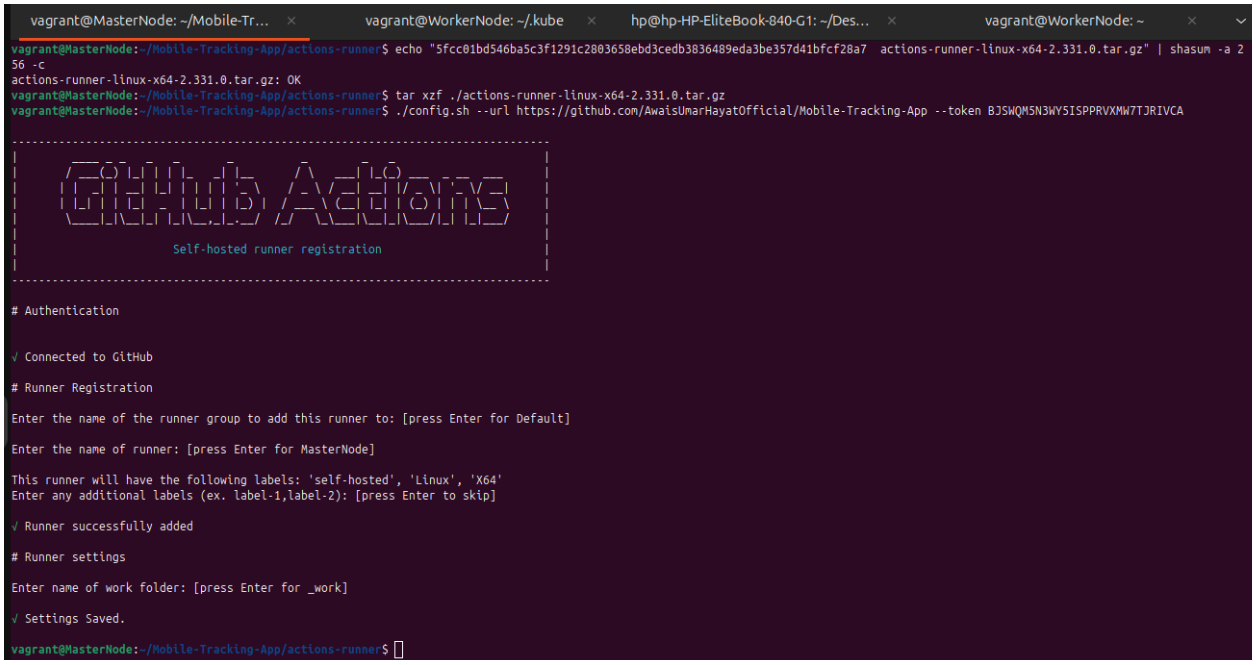
Add new self-hosted runner · AwaisUmarHayatOfficial/Mobile-Tracking-App

Using self-hosted runners in public repositories is not recommended. Forks of your public repository can potentially run dangerous code on your self-hosted runner by creating a pull request. Learn more about security hardening for self-hosted runners.

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. If you do not already have an existing volume licensing agreement for your GitHub purchases, by downloading and configuring the GitHub Actions Runner, you agree to the GitHub Customer Agreement.

Runner image: macOS [✓ Linux] Windows

Architecture: x64

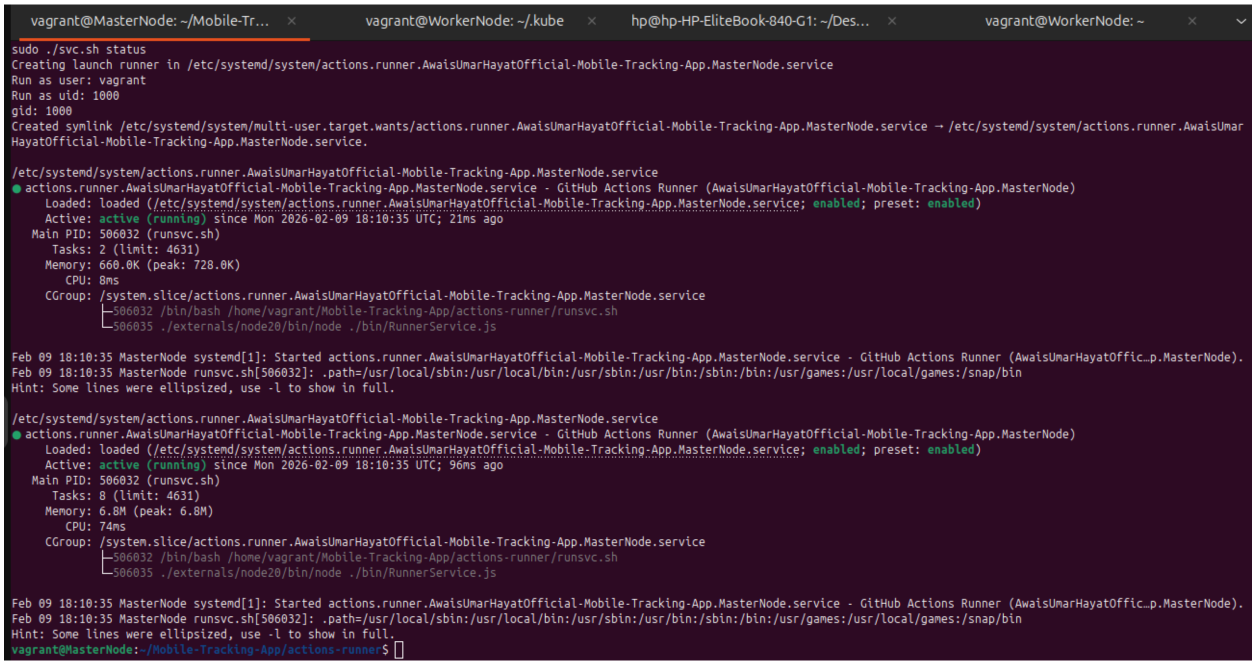

vagrant@MasterNode: ~/Mobile-Tr...
vagrant@WorkerNode: ~/.kube
hp@hp-HP-EliteBook-840-G1: ~/Des...
vagrant@WorkerNode: ~

```
vagrant@MasterNode:~/Mobile-Tracking-App/actions-runner$ echo "5fcc01bd546ba5c3f1291c2803658ebd3cedb3836489eda3be357d41bfcf28a7  actions-runner-linux-x64-2.331.0.tar.gz" | shasum -a 2
56 -c
actions-runner-linux-x64-2.331.0.tar.gz: OK
vagrant@MasterNode:~/Mobile-Tracking-App/actions-runner$ tar xzf ./actions-runner-linux-x64-2.331.0.tar.gz
vagrant@MasterNode:~/Mobile-Tracking-App/actions-runner$ ./config.sh --url https://github.com/AwaisUmarHayatOfficial/Mobile-Tracking-App --token BJSWQM5N3WY5ISPPRVXMW7TJRIVCA

--------------------------------------------------------------------------------
|        ____ _ _   _   _       _          _        _   _                      |
|       / ___(_) |_| | | |_   _| |__      / \   ___| |_(_) ___  _ __  ___      |
|      | |  _| | __| |_| | | | | '_ \    / _ \ / __| __| |/ _ \| '_ \/ __|     |
|      | |_| | | |_|  _  | |_| | |_) |  / ___ \ (__| |_| | (_) | | | \__ \     |
|       \____|_|\__|_| |_|\__,_|_.__/  /_/   \_\___|\__|_|\___/|_| |_|___/     |
|                                                                              |
|                       Self-hosted runner registration                        |
|                                                                              |
--------------------------------------------------------------------------------

# Authentication


√ Connected to GitHub

# Runner Registration

Enter the name of the runner group to add this runner to: [press Enter for Default]

Enter the name of runner: [press Enter for MasterNode]

This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]

√ Runner successfully added

# Runner settings

Enter name of work folder: [press Enter for _work]

√ Settings Saved.

vagrant@MasterNode:~/Mobile-Tracking-App/actions-runner$
```

Step 2: Install Runner on Your Master Node

On your **Master Node** terminal:

```
# Navigate to the runner directory
cd ~/Mobile-Tracking-App/actions-runner

# Install the runner as a background service (so it runs permanently)
sudo ./svc.sh install

# Start the runner service
sudo ./svc.sh start

# Check runner status
sudo ./svc.sh status
```


vagrant@MasterNode: ~/Mobile-Tr...
vagrant@WorkerNode: ~/.kube
hp@hp-HP-EliteBook-840-G1: ~/Des...
vagrant@WorkerNode: ~

```
sudo ./svc.sh status
Creating launch runner in /etc/systemd/system/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service
Run as user: vagrant
Run as uid: 1000
gid: 1000
Created symlink /etc/systemd/system/multi-user.target.wants/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service → /etc/systemd/system/actions.runner.AwaisUmar
HayatOfficial-Mobile-Tracking-App.MasterNode.service.

/etc/systemd/system/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service
● actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service - GitHub Actions Runner (AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode)
     Loaded: loaded (/etc/systemd/system/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service; enabled; preset: enabled)
     Active: active (running) since Mon 2026-02-09 18:10:35 UTC; 21ms ago
   Main PID: 506032 (runsvc.sh)
      Tasks: 2 (limit: 4631)
     Memory: 660.0K (peak: 728.0K)
        CPU: 8ms
     CGroup: /system.slice/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service
             ├─506032 /bin/bash /home/vagrant/Mobile-Tracking-App/actions-runner/runsvc.sh
             └─506035 ./externals/node20/bin/node ./bin/RunnerService.js

Feb 09 18:10:35 MasterNode systemd[1]: Started actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service - GitHub Actions Runner (AwaisUmarHayatOffic…p.MasterNode).
Feb 09 18:10:35 MasterNode runsvc.sh[506032]: .path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
Hint: Some lines were ellipsized, use -l to show in full.

/etc/systemd/system/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service
● actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service - GitHub Actions Runner (AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode)
     Loaded: loaded (/etc/systemd/system/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service; enabled; preset: enabled)
     Active: active (running) since Mon 2026-02-09 18:10:35 UTC; 96ms ago
   Main PID: 506032 (runsvc.sh)
      Tasks: 8 (limit: 4631)
     Memory: 6.8M (peak: 6.8M)
        CPU: 74ms
     CGroup: /system.slice/actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service
             ├─506032 /bin/bash /home/vagrant/Mobile-Tracking-App/actions-runner/runsvc.sh
             └─506035 ./externals/node20/bin/node ./bin/RunnerService.js

Feb 09 18:10:35 MasterNode systemd[1]: Started actions.runner.AwaisUmarHayatOfficial-Mobile-Tracking-App.MasterNode.service - GitHub Actions Runner (AwaisUmarHayatOffic…p.MasterNode).
Feb 09 18:10:35 MasterNode runsvc.sh[506032]: .path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
Hint: Some lines were ellipsized, use -l to show in full.
vagrant@MasterNode:~/Mobile-Tracking-App/actions-runner$
```

Generate a Docker Hub Personal Access Token (PAT)

Docker Hub tokens are used for secure login in CI/CD workflows.

Step 1: Access Docker Hub Security Settings

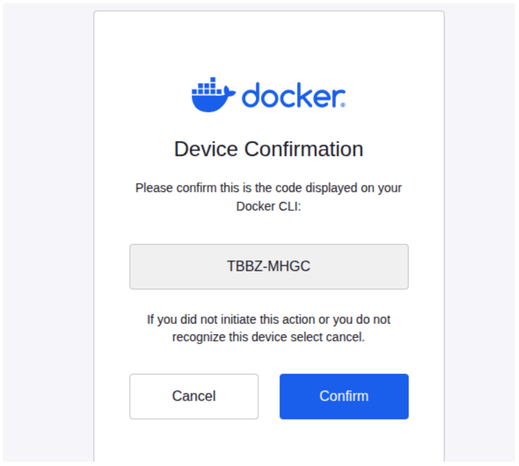
1. Open Docker Hub: https://hub.docker.com/
2. Click **top-right → Account Settings**
3. Go to **Security** on the left sidebar.
4. Click **Create New Access Token.**

Step 2: Create a New Token

1. Give your token a descriptive name (example: `github-actions`).
2. Click **Generate.**
3. Copy the generated token immediately (you won't be able to see it again).

Example token (for demonstration only):

- dckr_pat_hxZKjjcoYSU2sgsbBDUpeDsGuQc

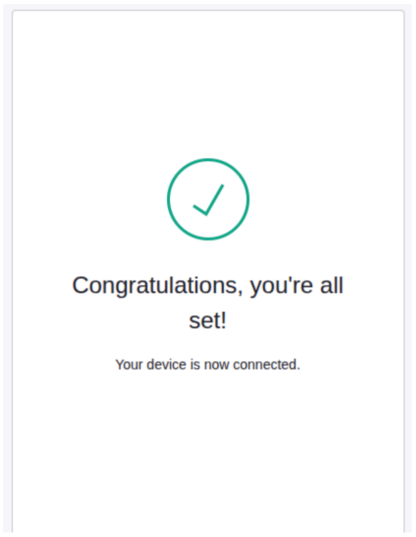


Device Confirmation

Please confirm this is the code displayed on your Docker CLI:

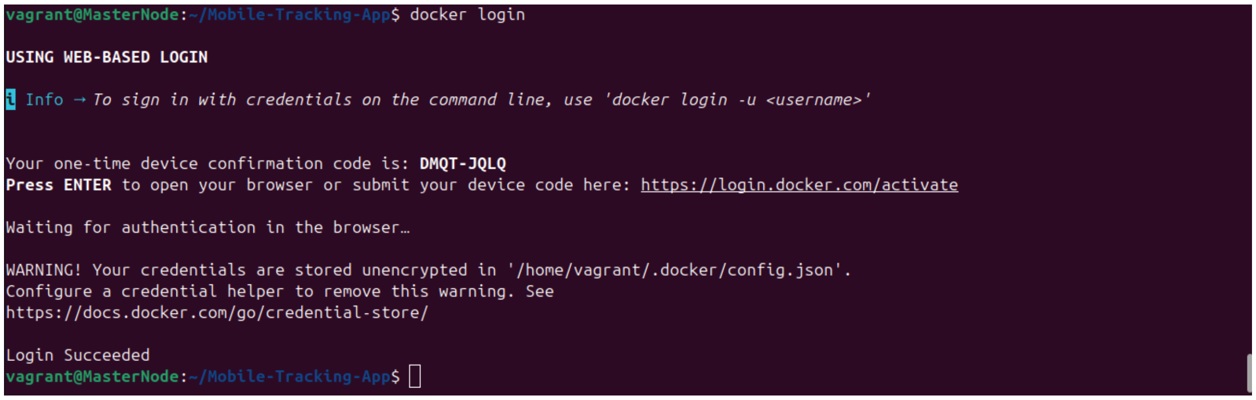
TBBZ-MHGC

If you did not initiate this action or you do not recognize this device select cancel.

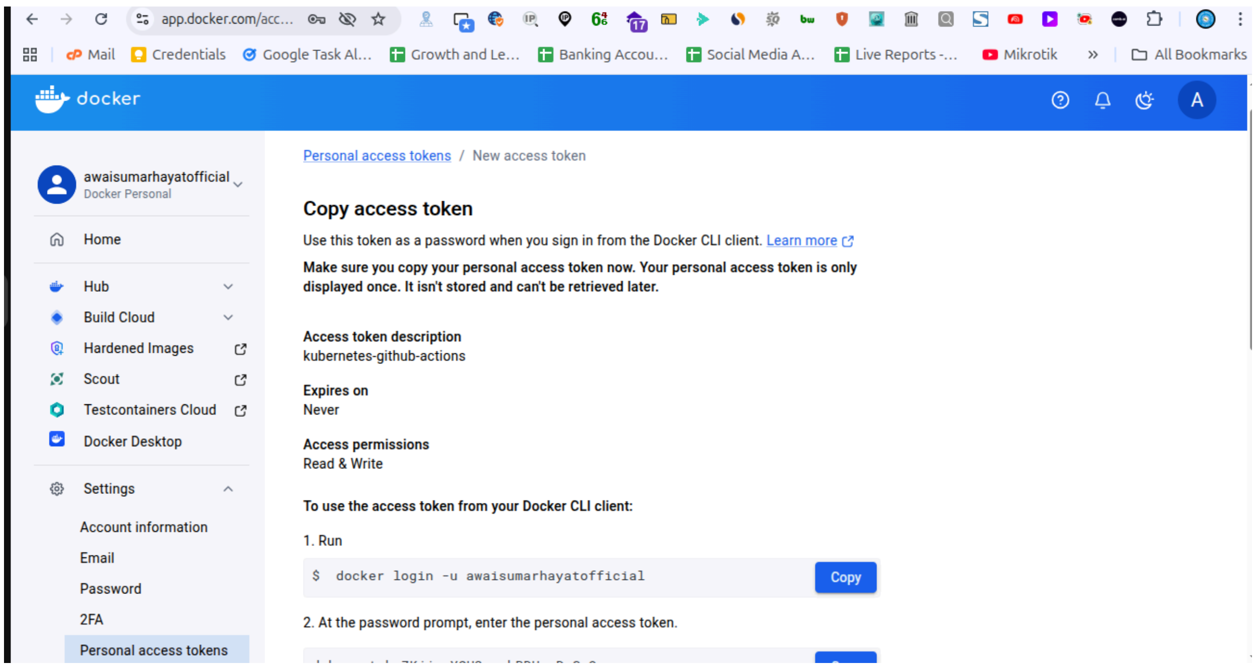


Congratulations, you're all set!

Your device is now connected.



```
vagrant@MasterNode:~/Mobile-Tracking-App$ docker login

USING WEB-BASED LOGIN

i Info → To sign in with credentials on the command line, use 'docker login -u <username>'


Your one-time device confirmation code is: DMQT-JQLQ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser…

WARNING! Your credentials are stored unencrypted in '/home/vagrant/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
vagrant@MasterNode:~/Mobile-Tracking-App$ 
```



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
kubernetes-github-actions

Expires on
Never

Access permissions
Read & Write

To use the access token from your Docker CLI client:

1. Run

```
$ docker login -u awaisumarhayatofficial
```

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

Store Docker Hub Token as a GitHub Secret

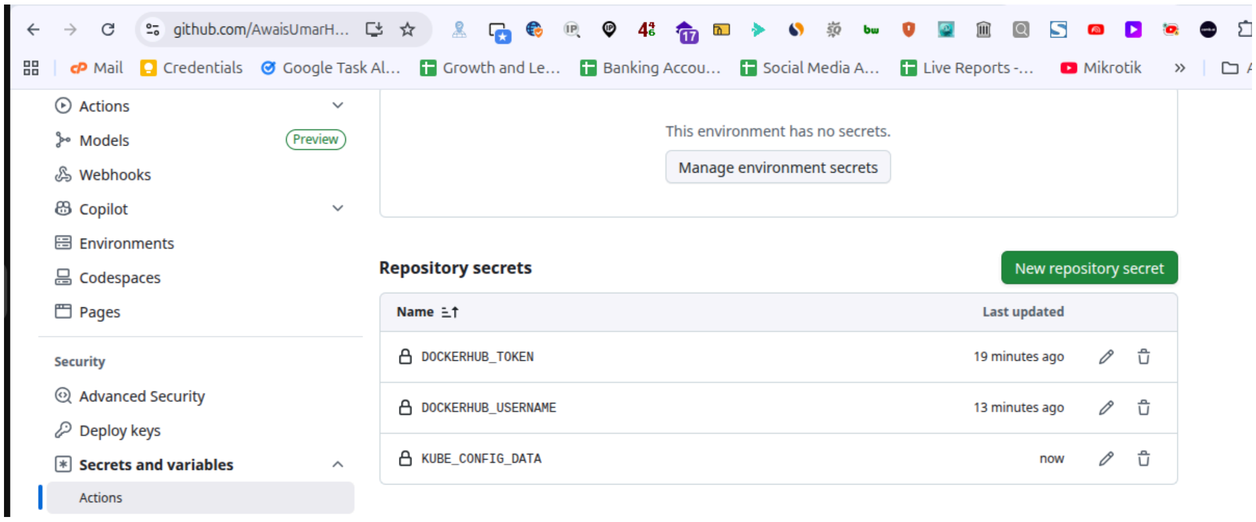
Step 1: Open GitHub Repository Secrets

Go to:

- https://github.com/AwaisUmarHayatOfficial/Mobile-Tracking-App/settings/secrets/actions

Step 2: Add New Repository Secret

1. Click **New repository secret**
2. Add the following:
   - **Name:** DOCKERHUB_TOKEN
   - **Value:** Paste your Docker Hub PAT
3. Click **Save**



This environment has no secrets.

Repository secrets

Name	Last updated
DOCKERHUB_TOKEN	19 minutes ago
DOCKERHUB_USERNAME	13 minutes ago
KUBE_CONFIG_DATA	now

4 Use Docker Hub Token in GitHub Actions Workflow

In your workflow YAML file, you can now use the token securely:

```
- name: Docker Login
  uses: docker/login-action@v2
  with:
    username: awaisumarhayatofficial
    password: ${{ secrets.DOCKERHUB_TOKEN }}
```

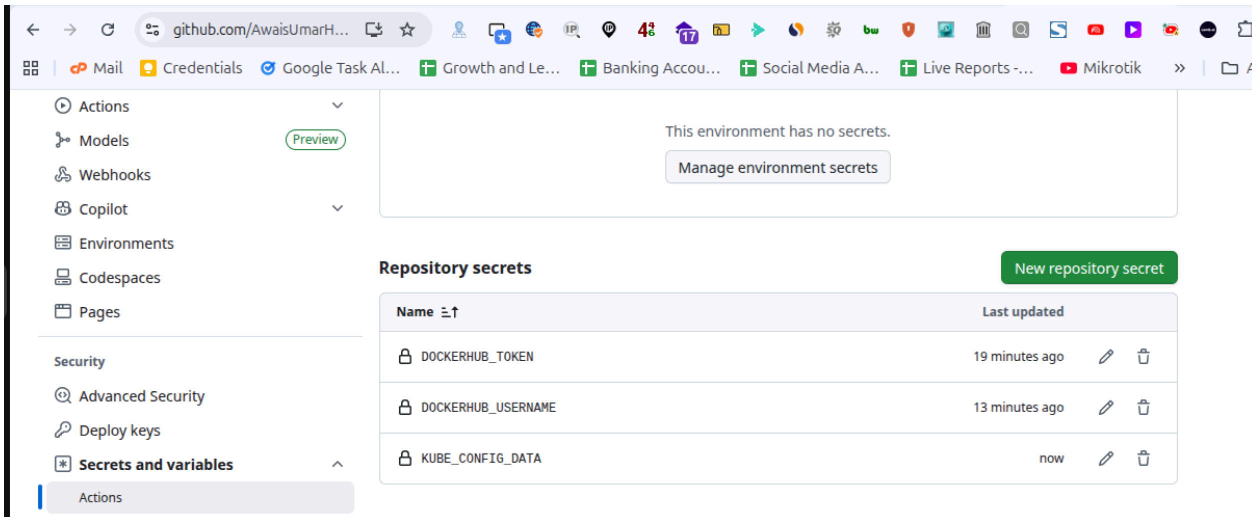
This ensures your workflow can authenticate with Docker Hub without exposing your password.

Base64 Encode Kubeconfig

If you want to use your kubeconfig in GitHub Actions (for Kubernetes deployments):

- cat $HOME/.kube/config | base64 -w 0

This produces a base64 string suitable for storing as a GitHub secret.



This environment has no secrets.

Repository secrets

Name	Last updated
DOCKERHUB_TOKEN	19 minutes ago
DOCKERHUB_USERNAME	13 minutes ago
KUBE_CONFIG_DATA	now

- Self-hosted runner is installed and running as a background service.
- Docker Hub PAT is securely stored as a GitHub secret.
- GitHub Actions workflow can log in to Docker Hub using the token.
- Optional: Kubernetes config can be safely encoded for CI/CD.

ci-cd.yaml

```
name: CI/CD Pipeline

on:
  push:
    branches:
      - main

jobs:
  build-test:
    runs-on: self-hosted
    steps:
      - name: Checkout Code
        uses: actions/checkout@v3

#      - name: Set up Node.js
#        uses: actions/setup-node@v3
#        with:
#          node-version: '20'

#      - name: Install Dependencies & Run Tests
#        run: |
#          cd frontend && npm install && npm test
#          cd ../backend && npm install && npm test

  sonar-scan:
    runs-on: self-hosted
```

```
   needs: build-test
   steps:
     - name: Checkout Code
       uses: actions/checkout@v3

#     - name: SonarQube Scan
#       uses: sonarsource/sonarcloud-github-action@v2
#       with:
#         projectKey: 'your_project_key'
#         organization: 'your_org'
#         token: ${{ secrets.SONAR_TOKEN }}

  trivy-security-scan:
   runs-on: self-hosted
   needs: build-test
   steps:
     - name: Checkout Code
       uses: actions/checkout@v3

 #    - name: Build Docker Images for Scan
 #    run: |
 #        docker build -t frontend:scan ./frontend
 #       docker build -t backend:scan ./backend

 #    - name: Trivy Scan - Frontend
 #      run: trivy image frontend:scan

 #    - name: Trivy Scan - Backend
 #      run: trivy image backend:scan

  docker-build-image:
     runs-on: self-hosted
     needs: [build-test, trivy-security-scan]
     steps:
     - name: Build Docker Images
       run: |
         docker build -t awaisumarhayatofficial/frontend:latest ./frontend
         docker build -t awaisumarhayatofficial/backend:latest ./backend

  docker-push-image:
   runs-on: self-hosted
   needs: [build-test, trivy-security-scan]
   steps:
     - name: Checkout Code
       uses: actions/checkout@v3

     - name: Log in to Docker Hub
       uses: docker/login-action@v2
       with:
         username: ${{ secrets.DOCKERHUB_USERNAME }}
         password: ${{ secrets.DOCKERHUB_TOKEN }}

     - name: Push Docker Images
       run: |
         docker push awaisumarhayatofficial/frontend:latest
         docker push awaisumarhayatofficial/backend:latest

  deploy:
   runs-on: self-hosted
   needs: docker-push-image
```

```
steps:
  - name: Checkout Code
    uses: actions/checkout@v3

  - name: Set up kubectl
    run: |
      mkdir -p $HOME/.kube
      echo "${{ secrets.KUBE_CONFIG_DATA }}" | base64 --decode > $HOME/.kube/config

  - name: Deploy to Kubernetes
    run: |
      kubectl apply -f frontend-deployment.yaml
      kubectl apply -f frontend-service.yaml
      kubectl apply -f backend-deployment.yaml
      kubectl apply -f backend-service.yaml
      kubectl apply -f redis-deployment.yaml
      kubectl apply -f redis-service.yaml
```