GitHub Actions CICD PipeLine

Dockerhub:

userName: [pusamsaikumar302@gmail.com](mailto:pusamsaikumar302@gmail.com)

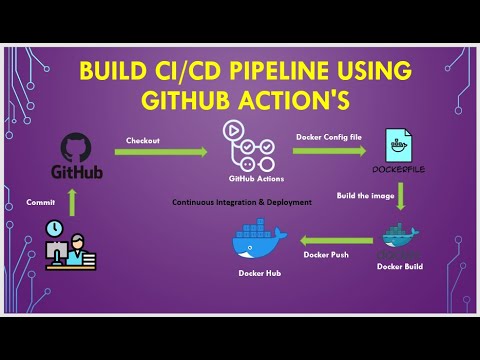
password: VK\_SE5kK\*Gjb2F&

GitHub:

Username: [pusamsaikumar302@gmail.com](mailto:pusamsaikumar302@gmail.com)

Password: saikumarP123

Workflow:



Pre-requisites:

------ create docker hub login  
 ----- create github repositoy

----- Dockerfile

----- application code

Perform some git commands to clone or existing repositories

***ClONE***

Using below commands, we can add a folder to repository.

1. git clone url

2. git add \*

3. git commit -m <commit message>

4. git remote add origin <repository path>

5. git push

***Add existing folder to git repository:***

Using below commands, we can add a folder to repository.

1. git init

2. git add \*

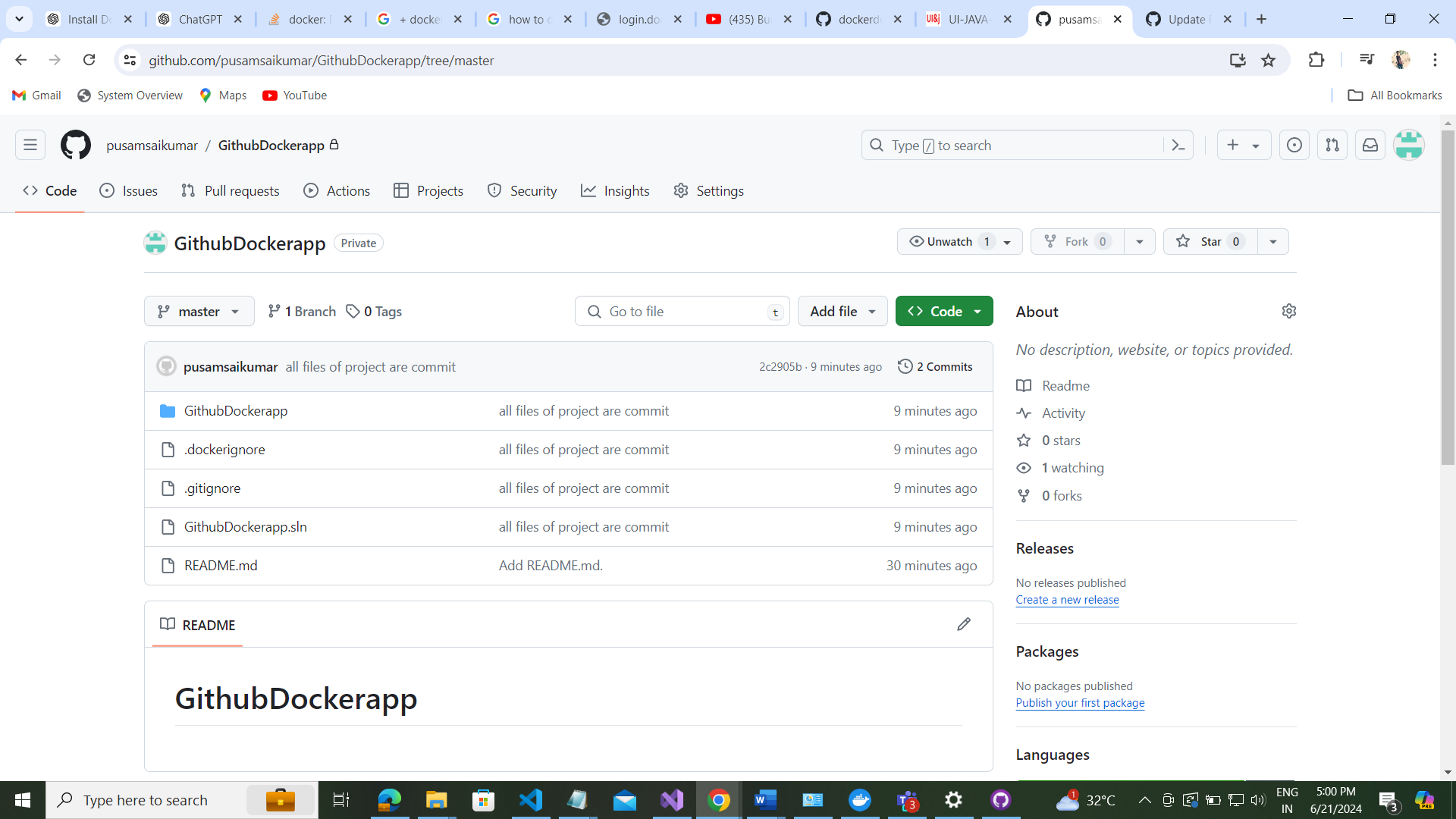
3. git commit -m <commit message>

4. git remote add origin <repository path>

5. git push

push the docker code inside the github repository:

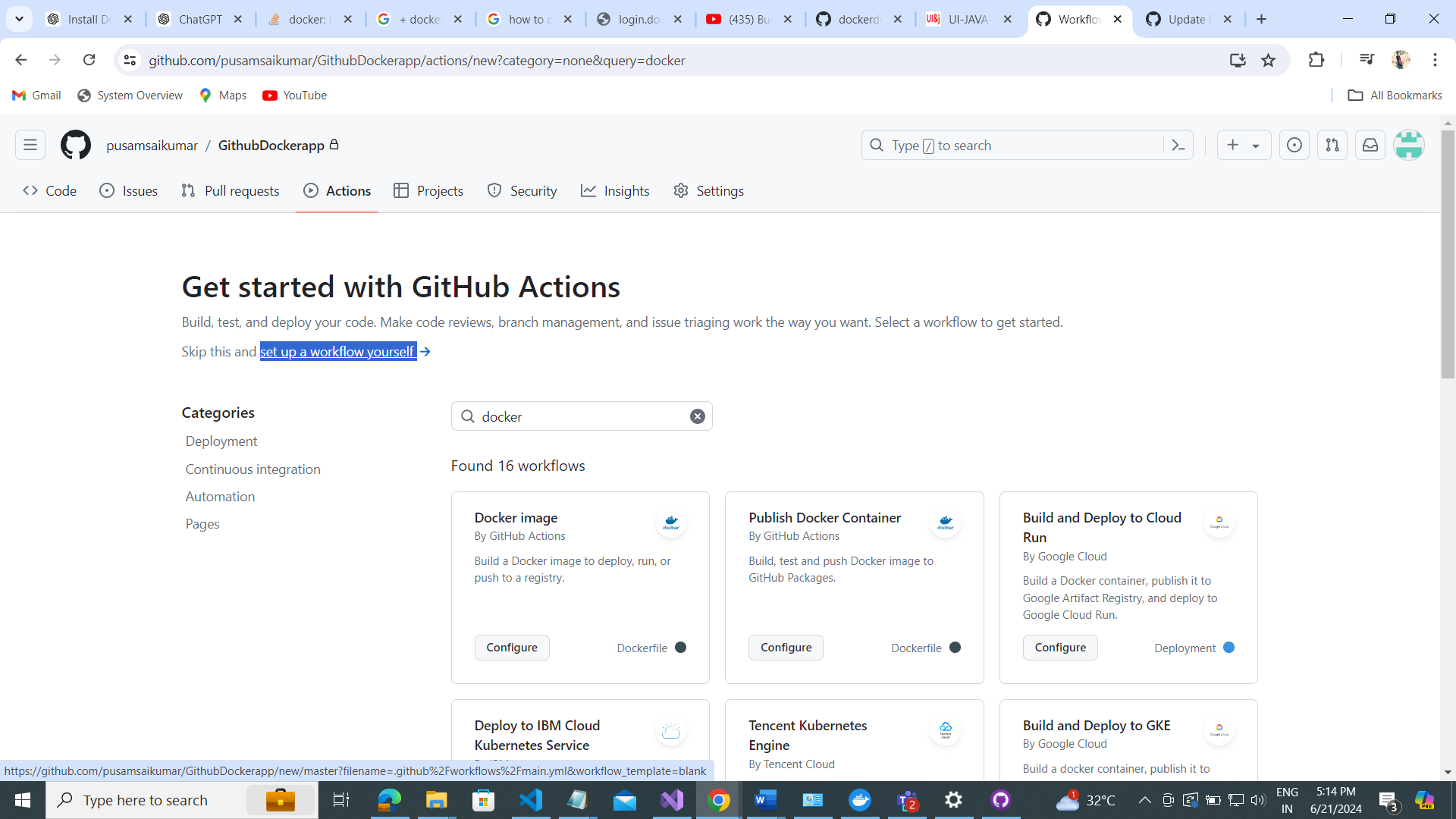
1. Create github repository
2. Push and commit the code into that repository



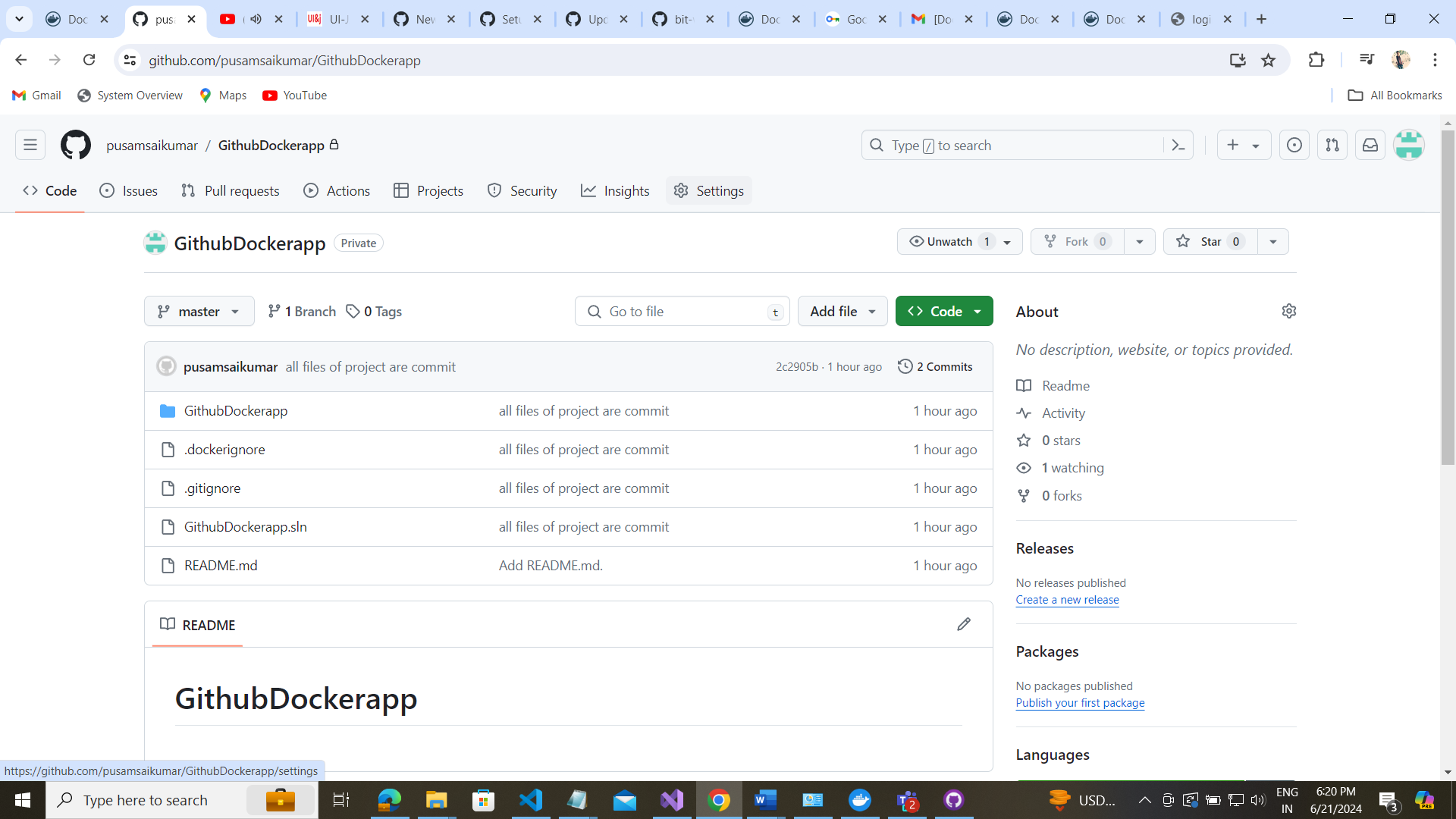
Go to Actions section :

It will you give the options like deploy code with other aws,azure…etc

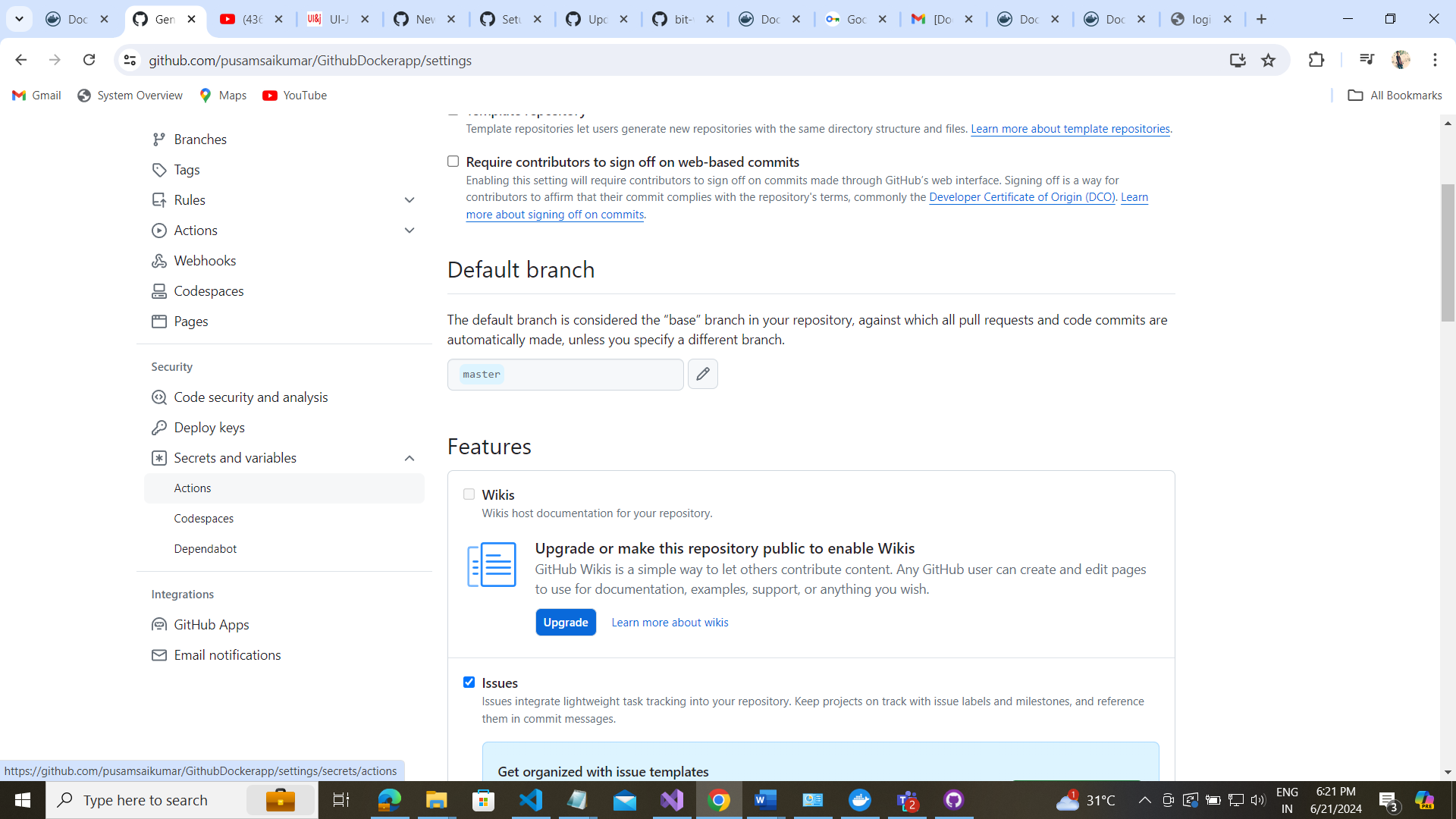
Otherwise if don’t have any workflow . Go with workflow



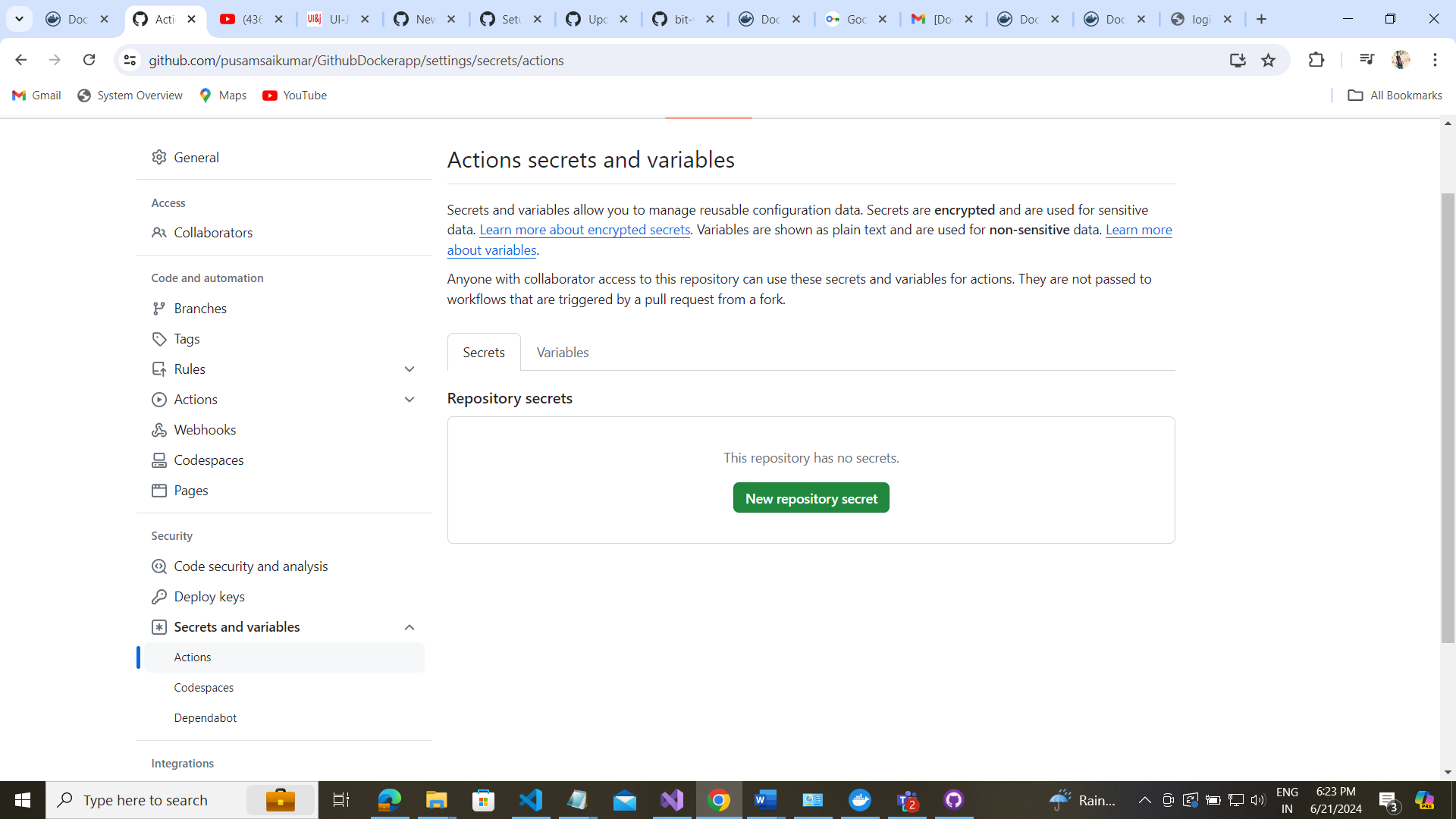
We have to save our docker hub credentials into the secrets of git hub.com  
click on the settings tab



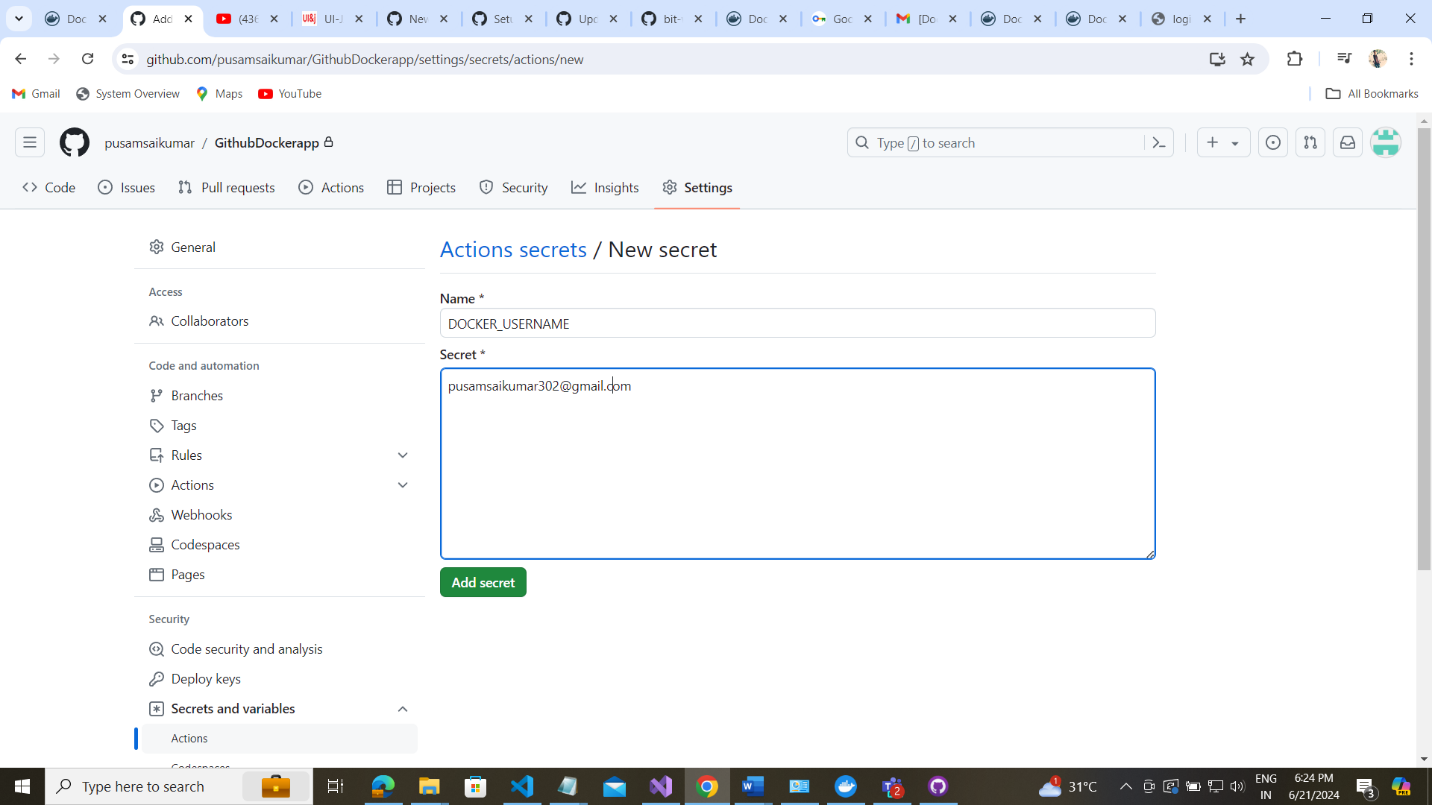
Select secrets and variables: Select Actions



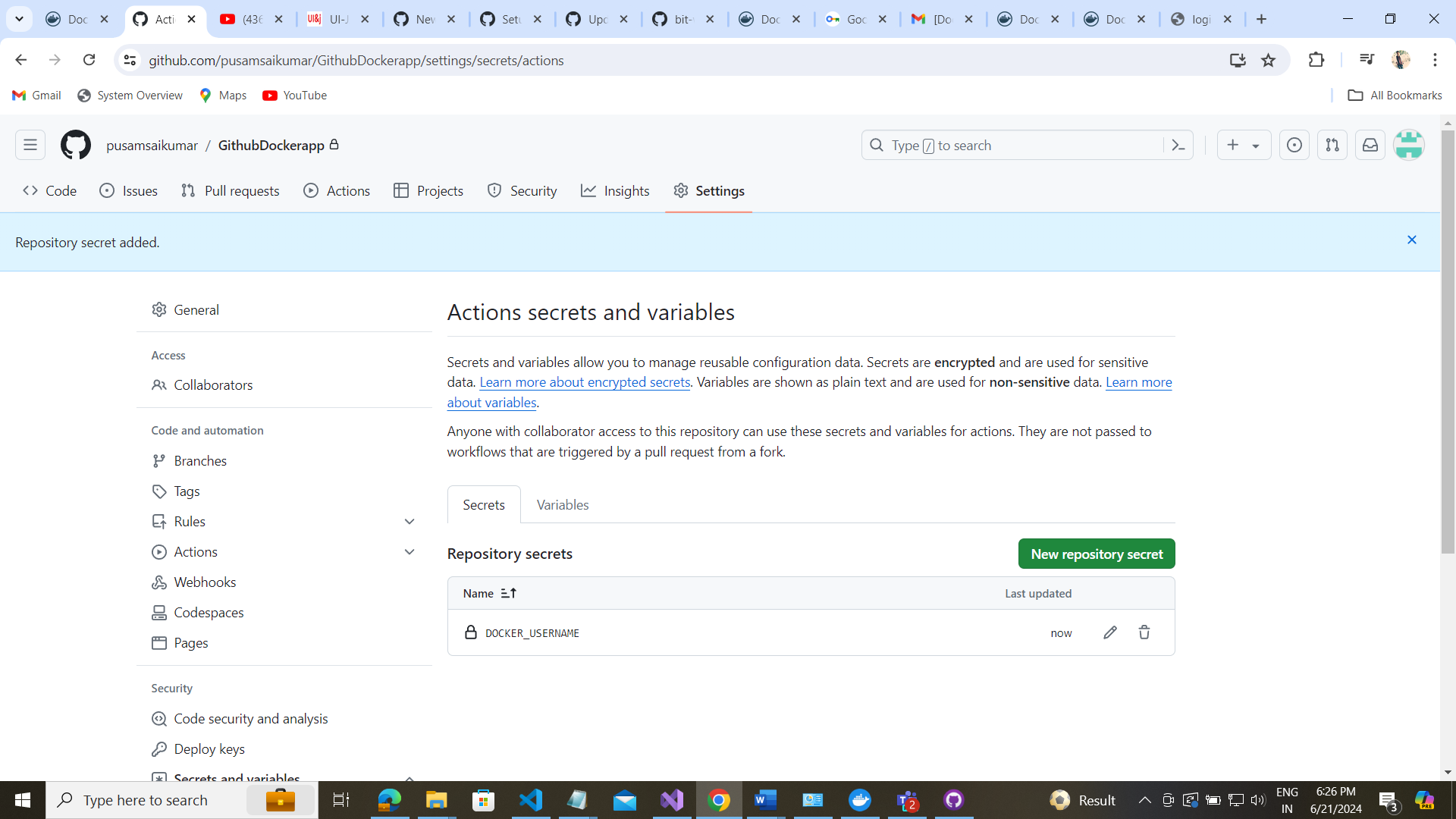
Create new secret repositories

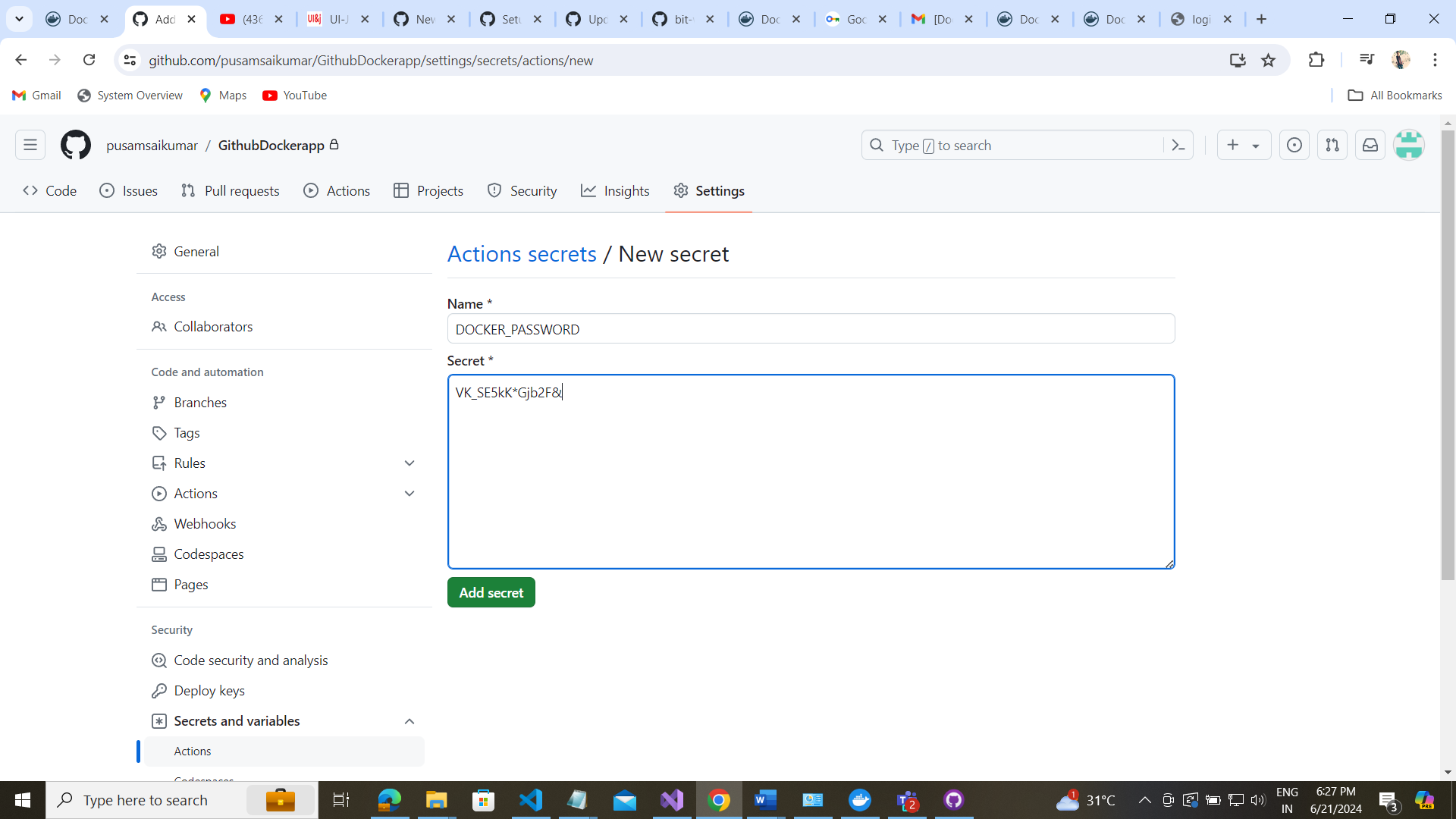


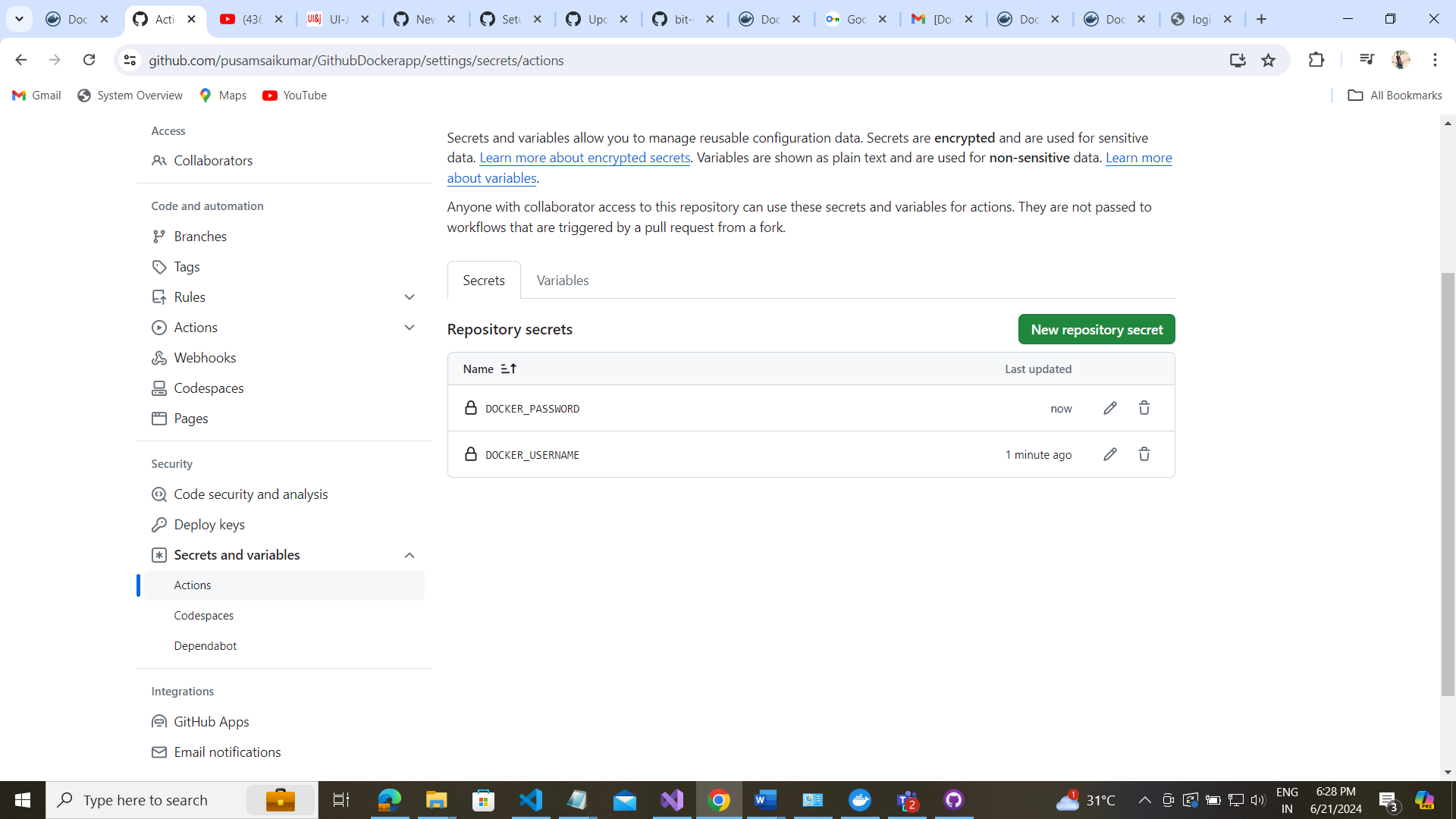
Add Secrets:



Add userpassword to secrets:







Note: while publish docker image into docker hub make sure give it correct docker hub path.

docker build -t pusamsai/testrepo:latest -f GithubDockerapp/Dockerfile .

docker push pusamsai/testrepo:latest

Create main.yml:

name: Publish Docker image to Docker Hub

on:

push:

branches: ['master']

jobs:

publish:

runs-on: windows-latest

steps:

- name: Checkout repository

uses: actions/checkout@v2

- name: Log in to Docker Hub

uses: docker/login-action@v2

with:

username: ${{ secrets.DOCKER\_USERNAME }}

password: ${{ secrets.DOCKER\_PASSWORD }}

- name: Build Docker image

env:

DOCKER\_BUILDKIT: 0 # Disable BuildKit

run: |

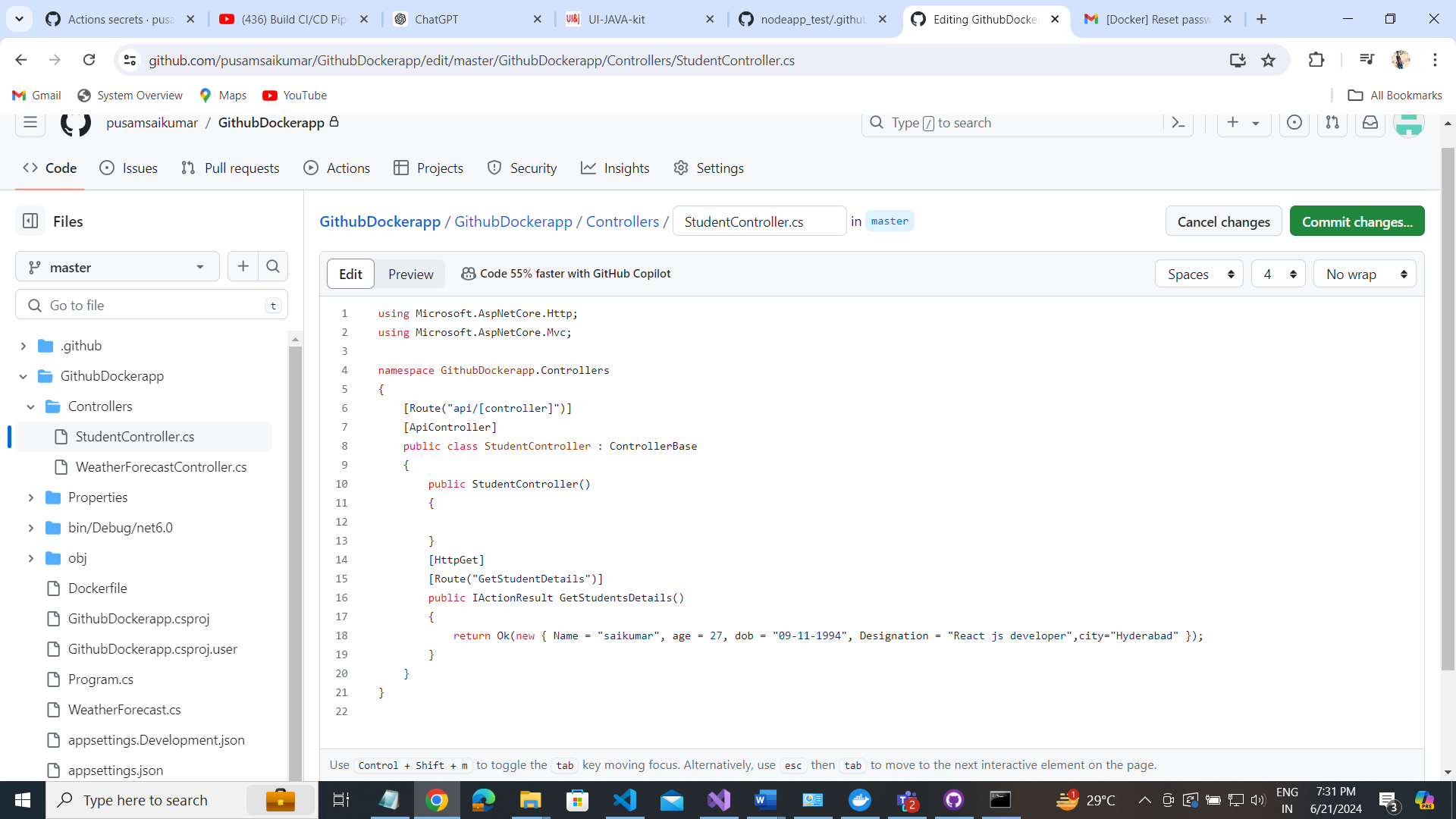
docker build -t pusamsai/testrepo:latest -f GithubDockerapp/Dockerfile .

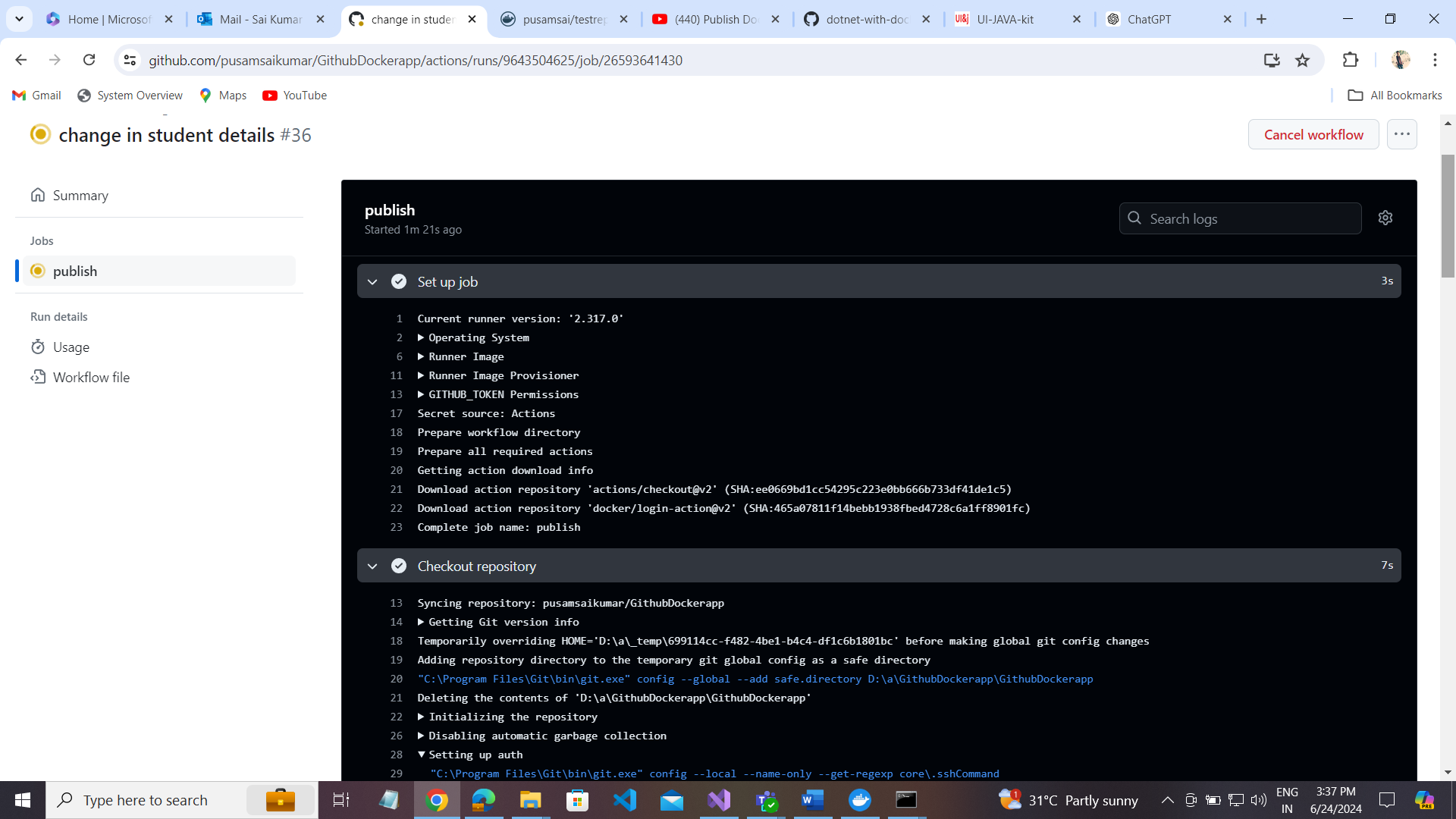
docker push pusamsai/testrepo:latest

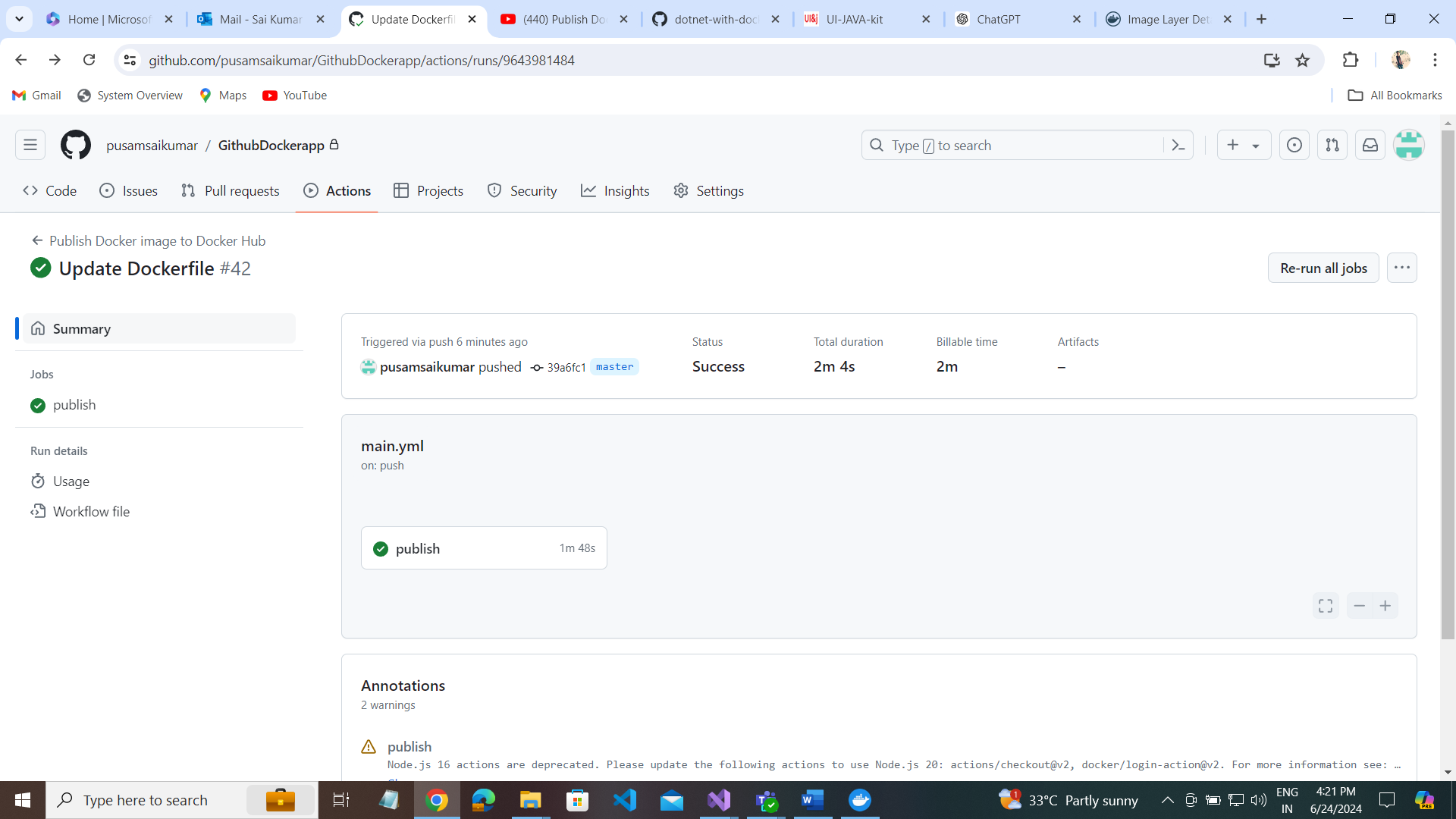
- name: Display Docker version

run: docker --version

Add some chages to any in the code. Then commit that changes

  
  
publish the image with Actions





GitHub Actions is a powerful automation tool provided by GitHub that allows you to automate

various tasks directly in your GitHub repositories. When combined with Docker, GitHub Actions can be used to build, test, and deploy Docker container-based applications efficiently. Here’s how GitHub Actions can be used with Docker:

Typical Use Cases

Continuous Integration (CI):

Build Docker Images: Automatically build Docker images whenever code is pushed to your repository.

Run Tests: Use GitHub Actions to run tests inside Docker containers to ensure your application behaves as expected.

Linting and Code Quality Checks: Run linters or code quality tools inside Docker containers to maintain code standards.

Continuous Deployment (CD):

Deploy to Environments: Use GitHub Actions to deploy Docker containers to various environments (e.g., staging, production) after successful tests or approvals.

Release Management: Automatically tag Docker images and push them to Docker registries (e.g., Docker Hub) as part of your release process.

Scheduled Tasks:

Automated Backups: Schedule GitHub Actions to run Docker containers that perform automated backups of databases or files.

Scheduled Jobs: Run periodic jobs inside Docker containers, such as data aggregation, cleanup tasks, or generating reports.

Complex Workflows:

Multi-Service Testing: Orchestrate testing workflows with multiple Docker containers to simulate complex environments.

Integration Testing: Run integration tests that require multiple services or dependencies using Docker-compose within GitHub Actions.

Example:

name: CI/CD with Docker

on:

push:

branches:

- main

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v2

- name: Set up Docker Buildx

uses: docker/setup-buildx-action@v1

- name: Login to Docker Hub

uses: docker/login-action@v2

with:

username: ${{ secrets.DOCKER\_USERNAME }}

password: ${{ secrets.DOCKER\_PASSWORD }}

- name: Build and push Docker image

id: docker\_build

uses: docker/build-push-action@v2

with:

context: .

file: ./Dockerfile

push: true

tags: |

pusamsai/testrepo:latest

pusamsai/testrepo:${{ github.sha }}

- name: Run tests

run: |

docker run --rm pusamsai/testrepo:${{ github.sha }} echo "Running tests..."

- name: Deploy to staging

if: github.event\_name == 'push' && github.ref == 'refs/heads/main'

run: |

echo "Deploying to staging environment..."

# Add deployment steps here