Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058,
India (Autonomous College Affiliated to University of Mumbai)

S.E Lab
Comps(T.E) - Batch- B
Topic-Speedy Typing

Name	Dhruvil Doshi
UID	2021300027

Experiment 07: Study DevOps and implement CI/CD pipeline.

Theory:

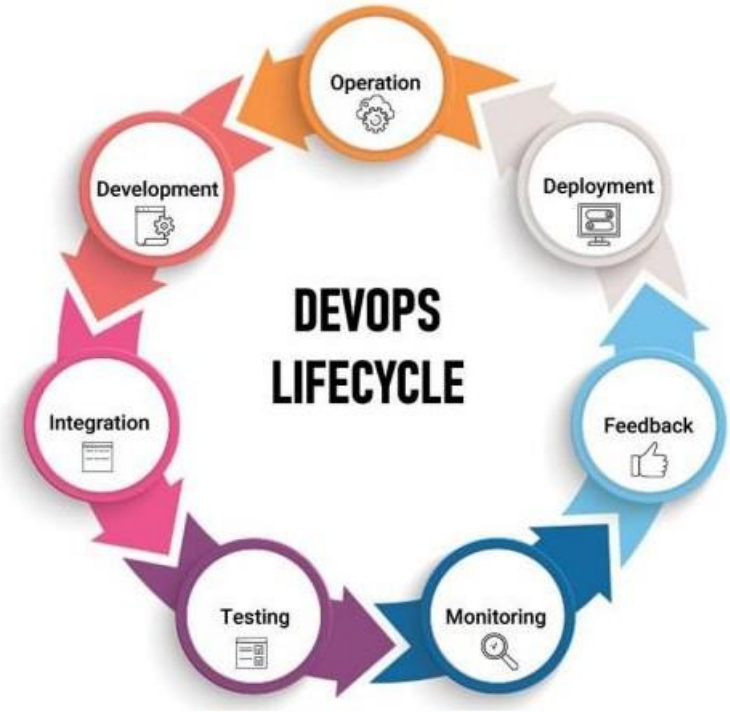
DevOps:
DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

How DevOps Works:

Under a DevOps model, development and operations teams are no longer "siloed." Sometimes, these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function. In some DevOps models, quality assurance and security teams may also become more tightly integrated with development and operations and throughout the application lifecycle. When security is the focus of everyone on a DevOps team, this is sometimes referred to as DevSecOps. These teams use practices to automate processes that historically have been manual and slow. They use a technology stack and tooling which help them operate and evolve applications quickly and reliably. These tools also help engineers independently accomplish tasks (for example, deploying code or provisioning infrastructure) that normally would have required help from other teams, and this further increases a team's velocity.

Benefits of DevOps:

- Speed
- Rapid Delivery
- Reliability
- Scale
- Improved Collaboration
- Security



CI/CD Pipeline:

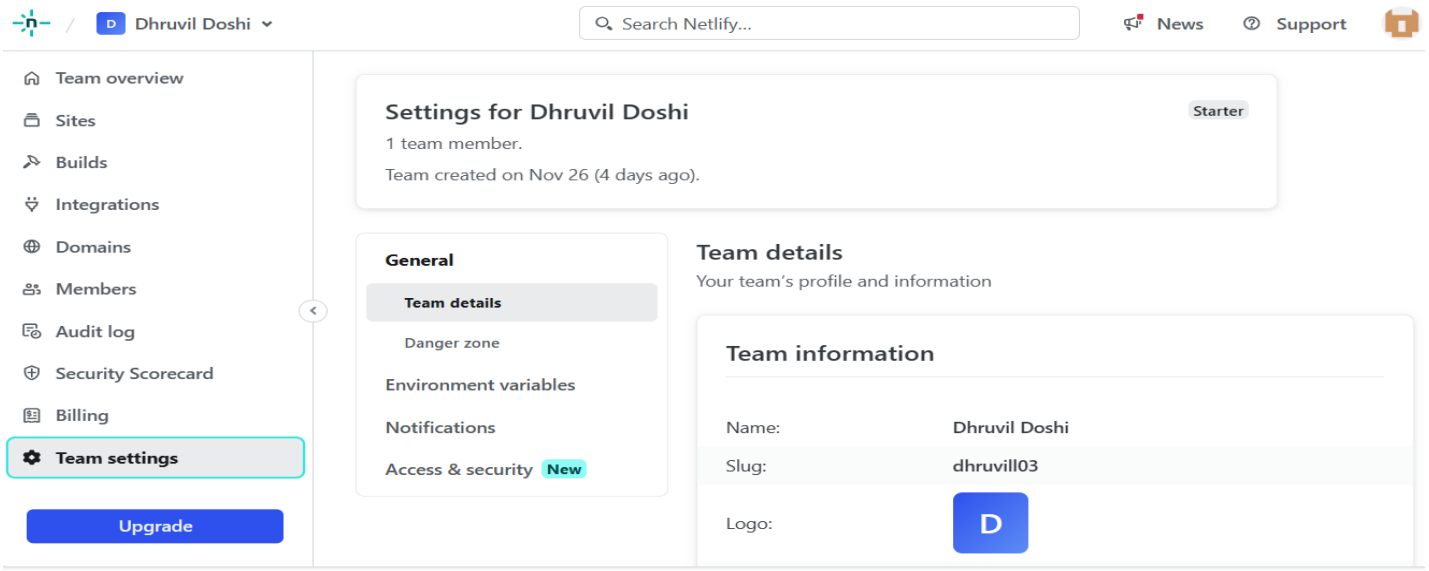
A pipeline is a process that drives software development through a path of building, testing, and deploying code, also known as CI/CD. By automating the process, the objective is to minimize human error and maintain a consistent process for how software is released. Tools that are included in the pipeline could include compiling code, unit tests, code analysis, security, and binaries creation. CI/CD is the backbone of a DevOps methodology, bringing developers and IT operations teams together to deploy software. As custom applications become key to how companies differentiate, the rate at which code can be released has become a competitive differentiator.

Sardar Patel Institute of Technology

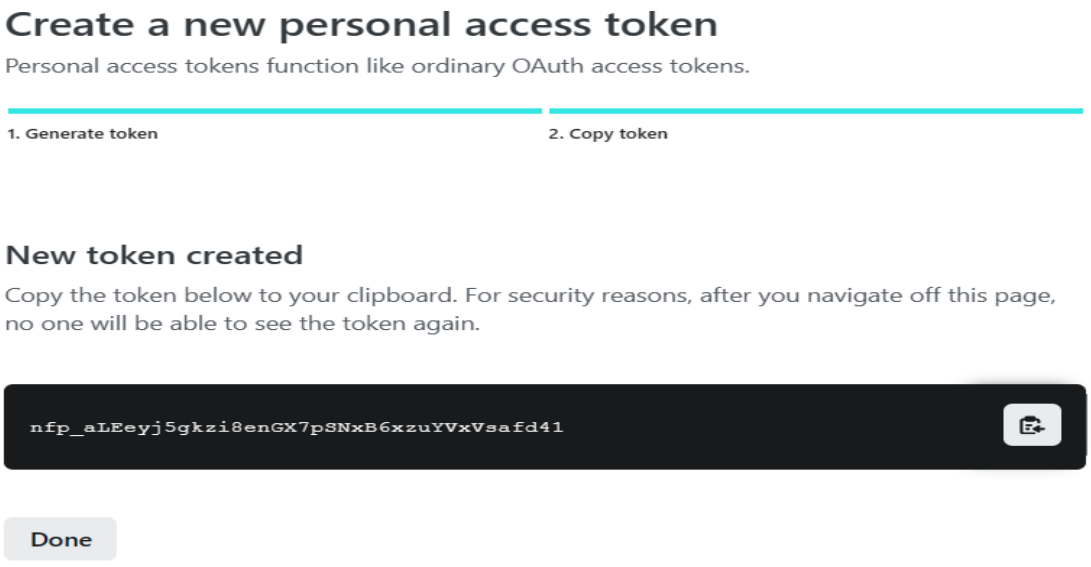
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058,
India (Autonomous College Affiliated to University of Mumbai)

Procedure:

1) Setup Netlify Account



2) Go to Settings => Application => Personal Access Token => Generate CI CD key

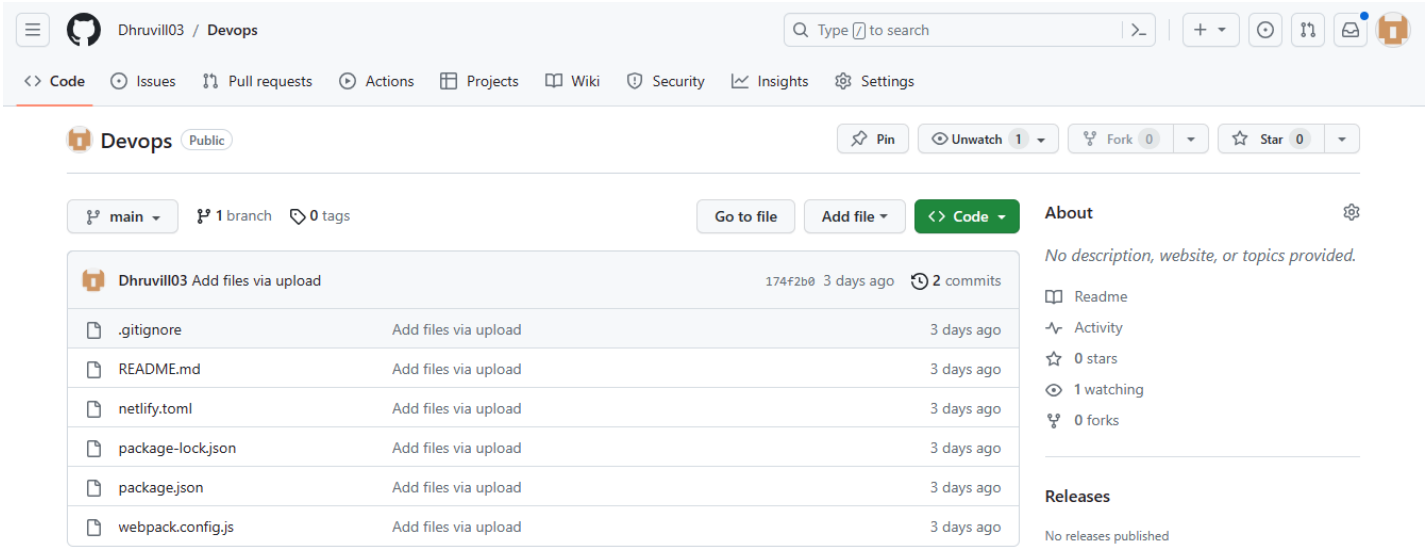


This is the NETLIFY_AUTH_TOKEN

Sardar Patel Institute of Technology

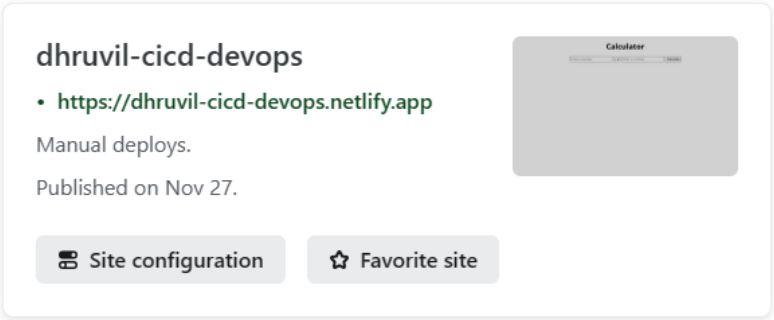
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058,
India (Autonomous College Affiliated to University of Mumbai)

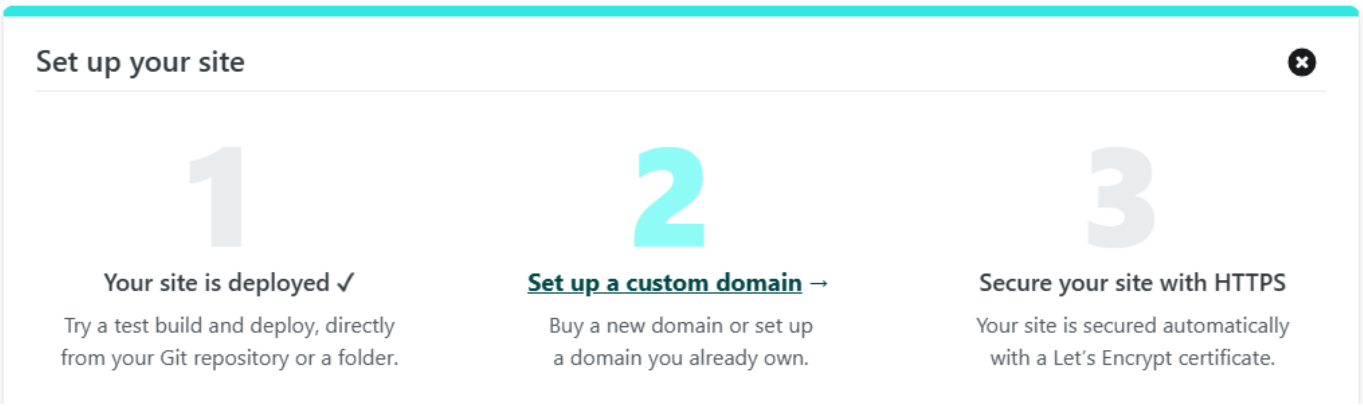
3) Create a new github repository and push your code to github repository.



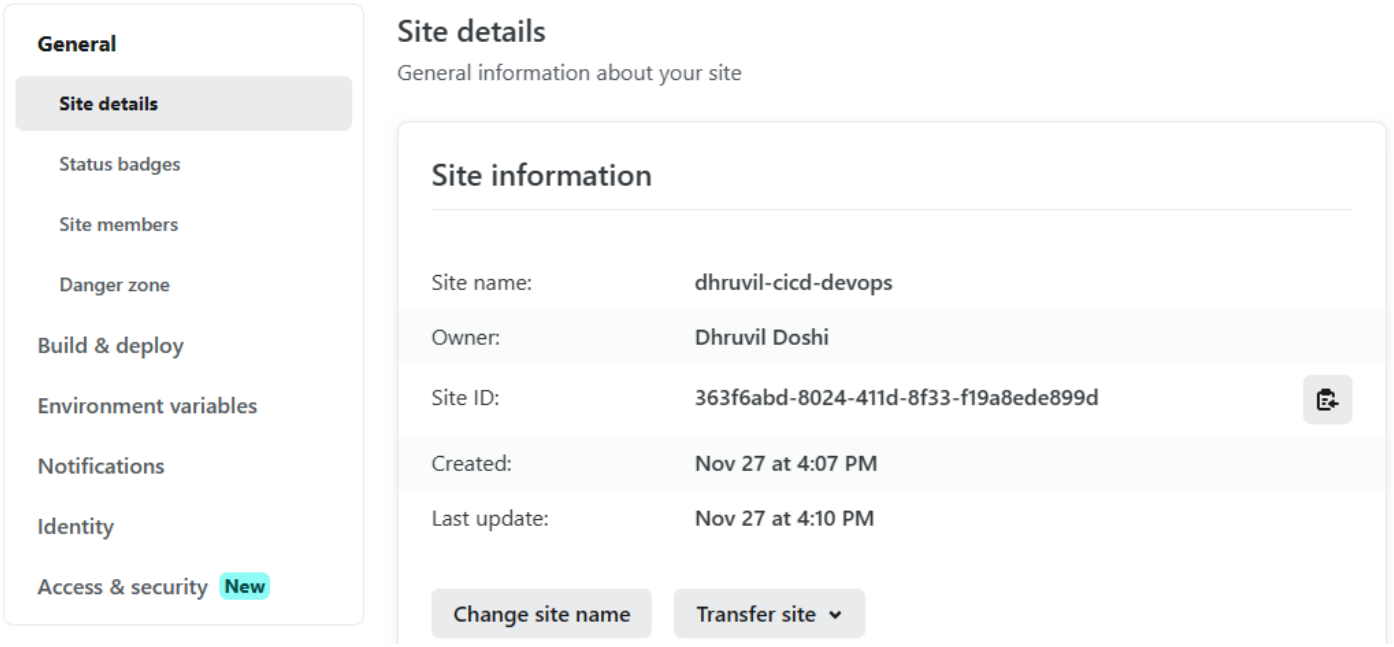
Repository Link: [Dhruvill03/Devops (github.com)]

4) Now drag drop the build folder into netlify to host your project

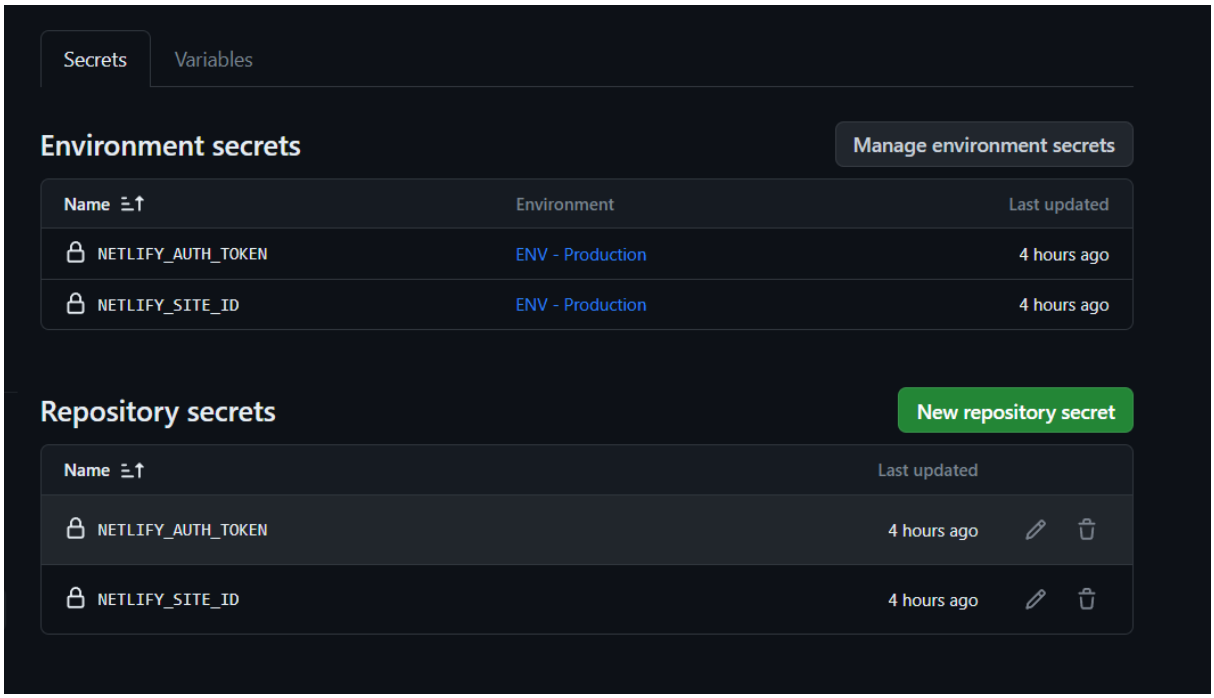




5) Go to site configuration and note down site ID



6) Now go back to the repository and in Repository Settings => Secrets and Variables => Actions, Add NETLIFY_AUTH_ID and NELIFY_SITE_ID

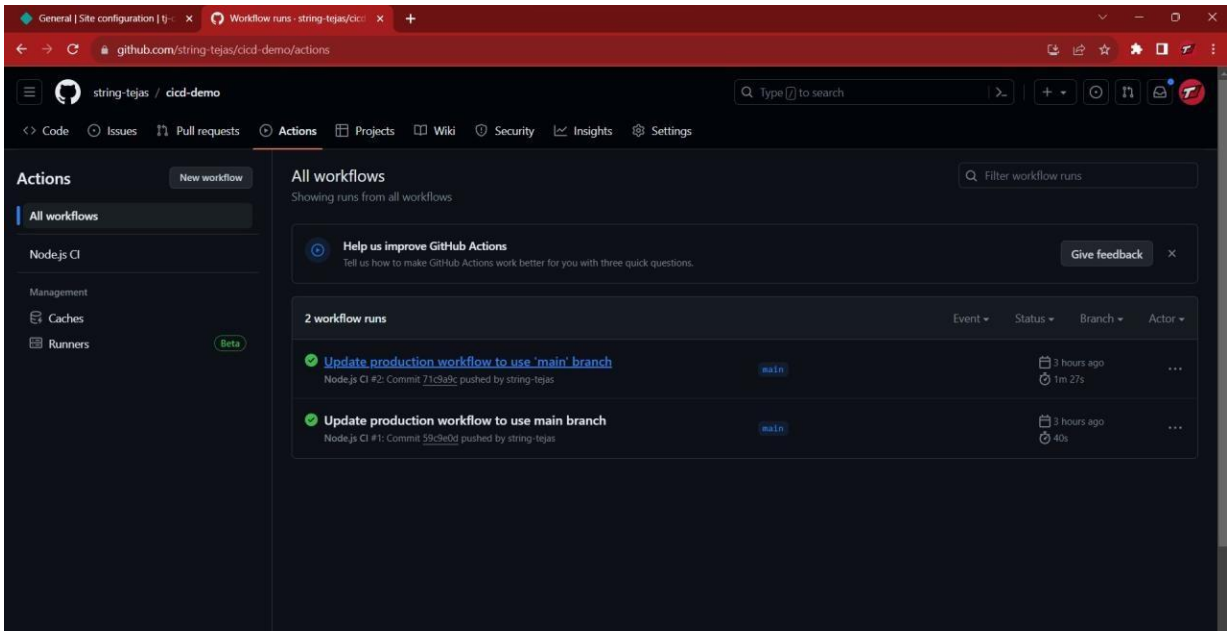


7) In your source code, create folder .github/workspace, in which create fileproduction.yml

```
name: Node.js CI

on
:   push:
        branches: [main]
    pull_request:
        branches: [main]

jobs:
    build:
        runs-on: ubuntu-latest

        steps:
            - uses: actions/checkout@v2
            - name: Setup Node.js
              uses: actions/setup-node@v1
              with:
                  node-version: "16.x"
            - run: npm install
            - run: npm run build
            - uses: actions/upload-artifact@v2
              with:
                  name: build
                  path: build/

    test:
        needs: build
        runs-on: ubuntu-latest

        steps:
            - uses: actions/checkout@v2
            - name: Setup Node.js
              uses: actions/setup-node@v1
              with:
                  node-version: "16.x"
            - run: npm install
            - uses: actions/download-artifact@v2
              with:
                  name: build
                  path: build
            - run: npm run test

    deploy:
```

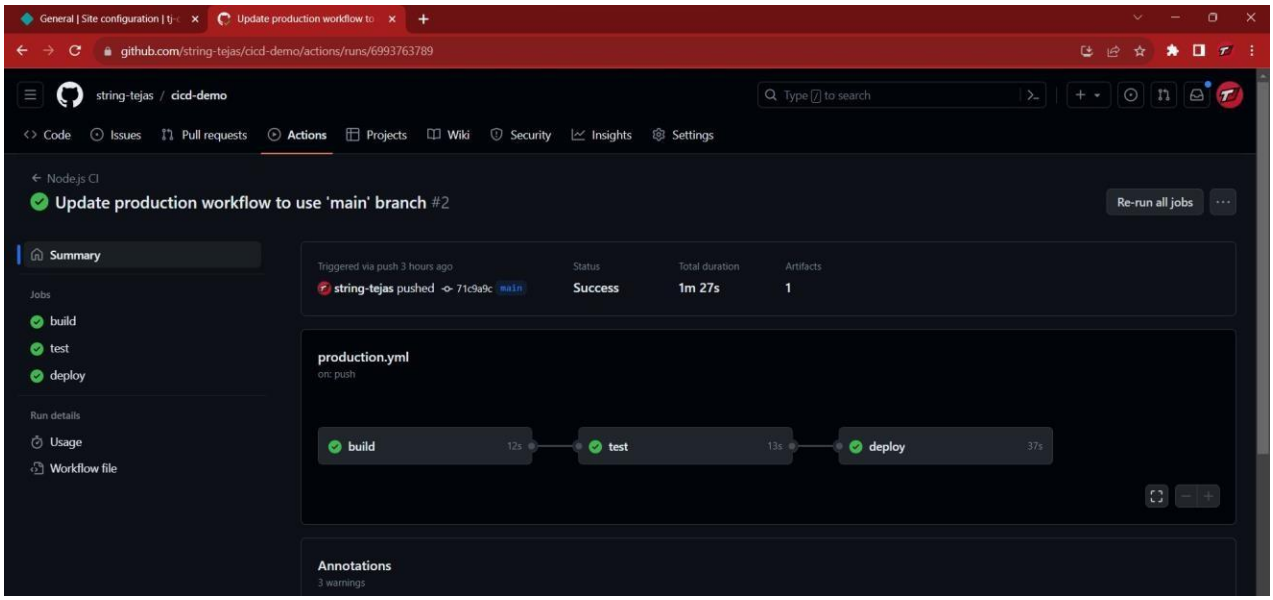
```
        needs: test
        runs-on: ubuntu-latest
        if: github.ref  = 'refs/heads/main'

        steps:
            - name: Checkout Repository
              uses: actions/checkout@v2

            - name: Setup Node.js
              uses: actions/setup-node@v1
              with:
                  node-version: "16.x"

            - name: Download Build Artifacts
              uses: actions/download-artifact@v2
              with:
                  name: build
                  path: build

            - name: Install Netlify CLI
              run: npm install -g netlify-cli

            - name: Deploy to Netlify
                        run:  npx  netlify  deploy   =dir=build   =site  ${{
secrets.NETLIFY_SITE_ID }}  =auth ${{ secrets.NETLIFY_AUTH_TOKEN }}  =prod
```

8) Push your code onto github again, then go to your repository => Actions

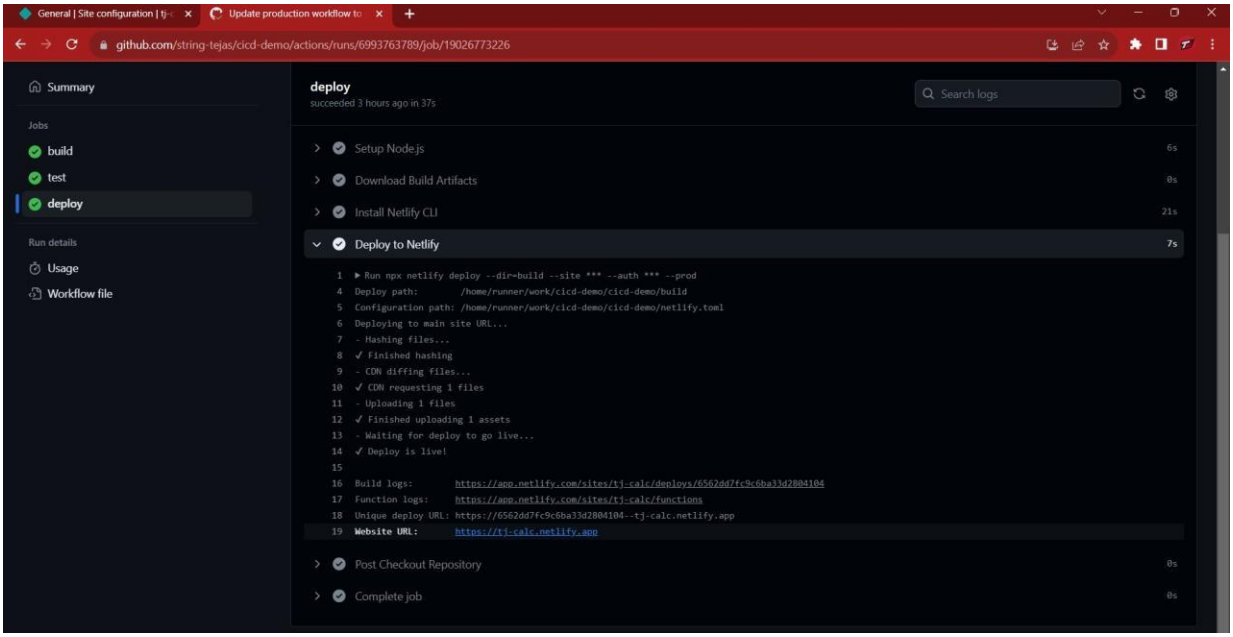


A new workflow is created.

9) All jobs executed correctly

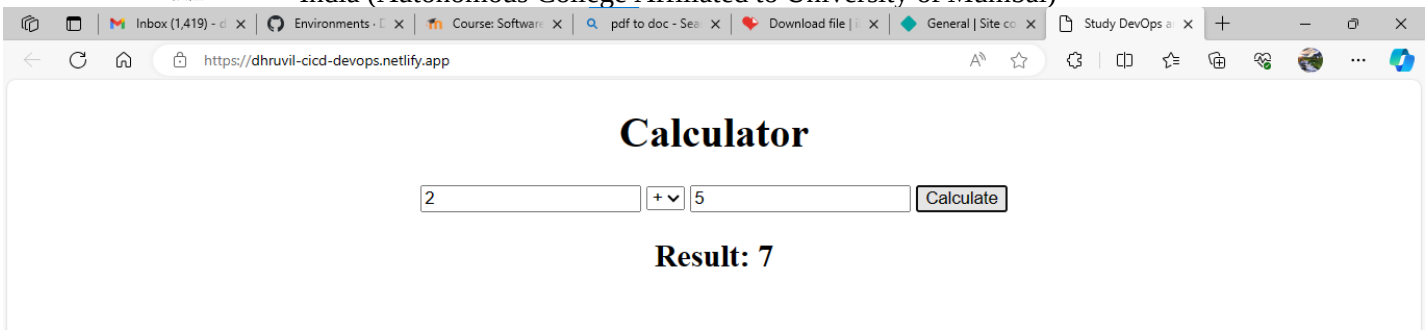


Site has been automatically built, tested and deployed.
Check out the URL



Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058,
India (Autonomous College Affiliated to University of Mumbai)



Site URL: [Study DevOps and implement CI/CD pipeline (dhruvil-cicd-devops.netlify.app)]

Conclusion:

After conducting this experiment, I have learnt to make use of devops to automate the task of building, testing and deploying the application using the CI/CD approach. I have successfully created a CI/CD pipeline using Github Actions.