

Assignment 1

Objective: In this assignment, you will create a simple GitHub Actions workflow that builds a custom Docker image with a basic NGINX configuration, deploys it to an Amazon EC2 instance, and ensures that the container remains running even after an EC2 instance restart.

Evaluation Criteria:

Your assignment will be evaluated based on the following criteria:

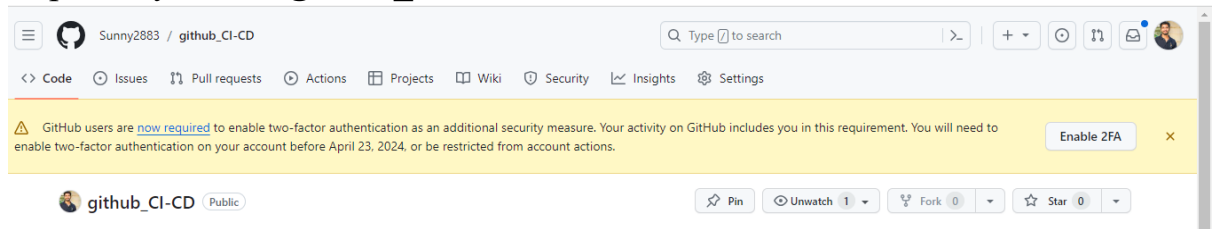
- Successful setup of GitHub Actions workflow. - 10 points
- Secure handling of sensitive information using GitHub Secrets. - 15 points
- Successful building and deploying of the custom Docker image to your EC2 instance. - 20 points
- Proper handling of the Docker container to ensure it remains running after an EC2 instance restart with documentation. -15 points
- Create another branch and push it. Pipeline should not trigger from the any other branch instead of dev. -10 points.

"Automated Deployment of NGINX with Terraform and GitHub Actions"

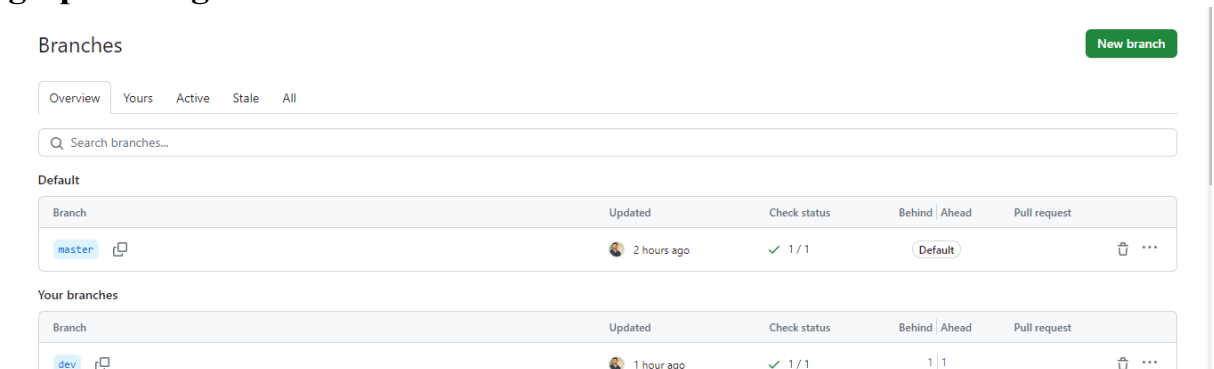
This assignment involves creating a Dockerfile for an NGINX image with basic configuration, utilizing Terraform to provision AWS resources, and integrating Terraform Cloud with GitHub Actions for automated deployment. GitHub Actions workflows are configured to build the NGINX image, deploy it to EC2 instances provisioned by Terraform, ensuring seamless container availability.

Step1: GitHub Repository Setup:

- Create a new GitHub repository for your project.
- Repository Name: **github_CI-CD**



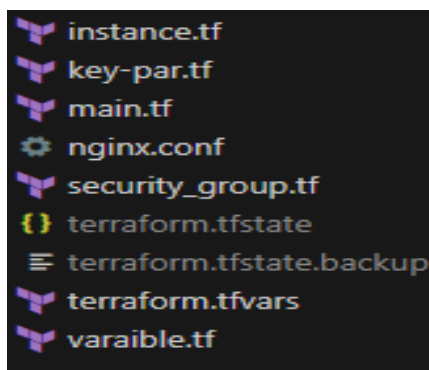
- Create two branches, dev and master, using the following commands:
- **git checkout -b dev**
- **git push origin dev**
- **git checkout -b master**
- **git push origin master**



**** Clone the repository to your local machine. ****

Step 2: Terraform Configuration:

- Write Terraform configuration files (.tf) to define your infrastructure resources, such as EC2 instances, security groups, Variables etc.



Step 3: Terraform Cloud Setup:

- Sign up for Terraform cloud and Create a new workspace in Terraform Cloud for your project.
- Link your GitHub repository to the Terraform Cloud workspace.
- Workspace name: **githubCICDForAssignment1**
- Repository: **Sunny2883/github_CI-CD**

Workspace Name ↓	Run Status	Repo	Latest Change ↕
githubCICDForAssignment1 Default Project	✓ Applied	Sunny2883/github_CI-CD	22 minutes ago

- Set up environment variables or sensitive values in Terraform Cloud.

Key	Value	Category
access_key SENSITIVE	Sensitive - write only	terraform
secret_key SENSITIVE	Sensitive - write only	terraform

Step 4: Docker Image Creation:

- Write a Dockerfile with a basic NGINX configuration.

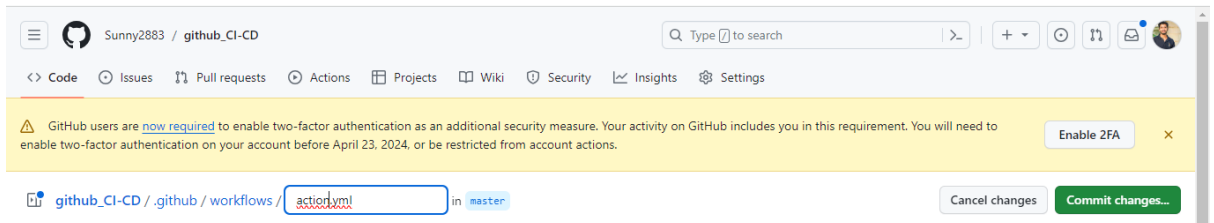
```
Dockerfile > ...
1 FROM nginx:latest
2 COPY ./index.html /usr/share/nginx/html/index.html
3 EXPOSE 80
4 CMD ["nginx", "-g", "daemon off;"]
```

- Build the Docker image locally and test it to ensure it functions as expected.

















REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
gitaction	latest	321f0a722286	3 days ago	187MB

Step 5: GitHub Actions Workflow Setup:

- Create a `.github/workflows` directory in your project repository and Inside this directory, create a YAML file (action.yml) to define your GitHub Actions workflow.



- Securely handle sensitive information using GitHub Secrets and Terraform Cloud variables.

Name 	Last updated
 AWS_ACCESS_KEY_ID	3 days ago  
 AWS_SECRET_ACCESS_KEY	3 days ago  
 DOCKER_PASSWORD	3 days ago  
 DOCKER_USERNAME	3 days ago  
 EC2_USERNAME	3 days ago  
 SSH_PORT	3 days ago  
 SSH_PRIVATE_KEY	3 days ago  

Step 6: Define the workflow steps to build the Docker image, push it to a Docker registry (e.g., Docker Hub), deploy it to your EC2 instance, and apply Terraform configuration using Terraform Cloud.

```
.github > workflows > Ⓞ action.yml
```

```
1  name: Terraform EC2 Deployment
2
3  on:
4    push:
5      branches: ["dev"]
6    pull_request:
7      branches: ["dev"]
8    workflow_dispatch:
9
10 jobs:
11   deploy:
12     runs-on: ubuntu-latest
13
14     steps:
15       - name: Checkout code
16         uses: actions/checkout@v3
17
18
19
20       - name: Build and push Docker image
21         env:
22           DOCKER_USERNAME: ${ secrets.DOCKER_USERNAME }
23           DOCKER_PASSWORD: ${ secrets.DOCKER_PASSWORD }
24         run: |
25
26           docker build -t mynginx .
27
28           docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
29
30           docker tag mynginx $DOCKER_USERNAME/mynginx:latest
31
32           docker push $DOCKER_USERNAME/mynginx:latest
```

```
.github > workflows > Ⓞ action.yml
```

```
10 jobs:
11   deploy:
12     steps:
13
14       - name: Build and push Docker image
15         run: |
16
17           docker tag mynginx $DOCKER_USERNAME/mynginx:latest
18
19           docker push $DOCKER_USERNAME/mynginx:latest
20
21
22       - name: AWS login
23         uses: aws-actions/configure-aws-credentials@v4
24         with:
25           aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
26           aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
27           aws-region: us-east-2
28
29       - name: SSH into EC2 instance and deploy container
30         uses: appleboy/ssh-action@master
31         with:
32           host: "44.212.58.158"
33           username: ${ secrets.EC2_USERNAME }
34           key: ${ secrets.SSH_PRIVATE_KEY }
35           port: ${ secrets.SSH_PORT }
36           script: |
37             sudo docker pull "${ secrets.DOCKER_USERNAME }/mynginx:latest"
38             sudo docker run -d --restart=always -p 80:80 "${ secrets.DOCKER_USERNAME }/mynginx:latest"
```

Step 7: Deployment and Testing:

- Trigger the command `git push origin master` and Verify the Changes on terraform cloud.

```
386b6c3...cb9c336 master -> master
PS C:\Users\promact\Desktop\Github_CI-CD> git add .
PS C:\Users\promact\Desktop\Github_CI-CD> git commit -m "Added more content on html file"
[master 7c1aa06] Added more content on html file
1 file changed, 86 insertions(+), 2 deletions(-)
PS C:\Users\promact\Desktop\Github_CI-CD> git push origin master
```

Plan finished 2 hours ago Resources: 3 to add, 0 to change, 0 to destroy

Started 2 hours ago > Finished 2 hours ago

+ 3 to create

Filter by action Terraform 1.7.4 [Download raw log](#)

> aws_instance.git_instance[0]

> aws_key_pair.key-tf

> aws_security_group.SecurityGroupforgit

[Download Sentinel mocks](#) ⓘ Sentinel mocks can be used for [testing your Sentinel policies](#)

- Confirm and Apply:

Apply finished 2 hours ago Resources: 3 added, 0 changed, 0 destroyed

Started 2 hours ago > Finished 2 hours ago

+ 3 created

Filter by action Terraform 1.7.4 [Download raw log](#)

> aws_instance.git_instance[0] ✓ Created id=i-04355c8ac1a7a8900

> aws_key_pair.key-tf ✓ Created id=my-newkeypair

> aws_security_group.SecurityGroupforgit ✓ Created id=sg-0b35c3b0ec330920d

State versions created:
[Sunny2883/githubCICDForAssignment1#sv-RdsW4RqRhTKqQiz1](#) (Mar 11, 2024 14:44:39 pm)

Resources Created on AWS:

- Instance name: gitactionInstance
- Instance ip: 44.202.92.11 (Changes Everytime)
- Instance id: i-04355c8ac1a7a8900
- Instance type: t2.micro
- Key-pair: my-newkeypair

- Security Group: SecurityGroupgit
- Security group id: sg-0b35c3b0ec330920d

Step 8: Test the GitHub Actions workflow by pushing changes to the `dev` branch and observing the deployment process.

← Terraform EC2 Deployment

✓ Changes ip address of instance #24 Re-run all jobs ...

Summary

Jobs

- ✓ deploy

Run details

Usage

Workflow file

deploy

succeeded now in 19s Beta Give feedback Search logs Refresh Settings

- Set up job 1s
- Build appleboy/ssh-action@master 3s
- Checkout code 0s
- Build and push Docker image 7s
- AWS login 0s
- SSH into EC2 instance and deploy container 7s
- Post AWS login 0s
- Post Checkout code 0s
- Complete job 0s

← Terraform EC2 Deployment

✓ Changes ip address of instance #24 Re-run all jobs ...

Summary

Jobs

- ✓ deploy

Run details

Usage

Workflow file

Triggered via push 1 minute ago

Sunny2883 pushed • 446b836 dev

Status: Success

Total duration: 29s

Artifacts: —

action.yml

on: push

✓ deploy 19s

Annotations

1 warning

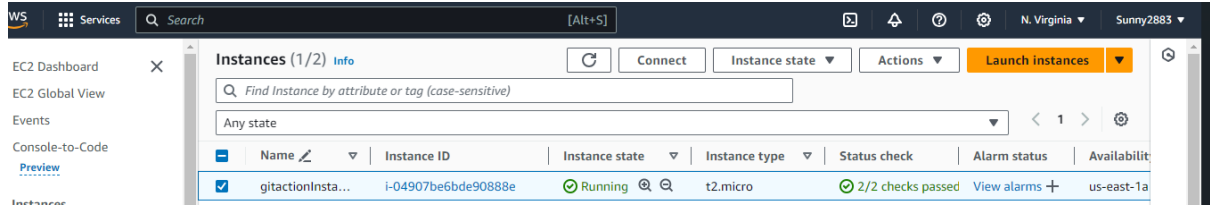
⚠ deploy

Node.js 16 actions are deprecated. Please update the following actions to use Node.js 20: actions/checkout@v3. For more information see...

https://github.com/Sunny2883/github_CI-CD/actions/runs/8230334267

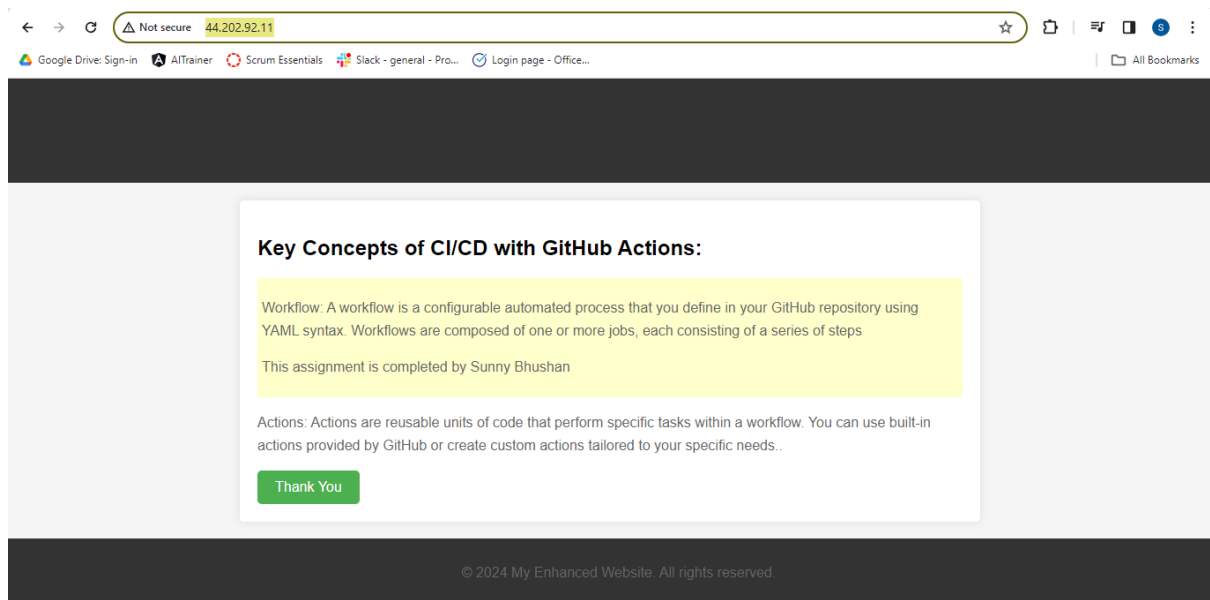
Step 9: Veirfy The Application on Port <http://44.202.92.11/>

**** I did not use Elastic ip address so everytime ports changes ****



Final Output:

Home Page:



Thank You