CI-CD Pipe Line Project:

Git Link:

Problem:

Create a complete CI-CD pipeline using bash, python and crontabs. The list of tasks is specified below:

* Task 1: Set Up a Simple HTML Project
  + - Create a simple HTML project and push it to a GitHub repository.
* Task 2: Set Up an AWS EC2/Local Linux Instance with Nginx
* Task 3: Write a Python Script to Check for New Commits
  + - Create a Python script to check for new commits using the GitHub API.
* Task 4: Write a Bash Script to Deploy the Code
  + - Create a bash script to clone the latest code and restart Nginx.
* Task 5: Set Up a Cron Job to Run the Python Script
  + - Create a cron job to run the Python script at regular intervals.
* Task 6: Test the Setup
  + - Make a new commit to the GitHub repository and check that the changes are automatically deployed.
* Task 1: Set Up a Simple HTML Project
  + - Create a simple HTML project and push it to a GitHub repository.

Html code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>HeroVired</title>

</head>

<body>

<header>

<h1>Welcome to HeroVired!</h1>

</header>

<nav>

<ul>

<li><a href="#section1">Batch 1</a></li>

<li><a href="#section2">Batch 2</a></li>

<li><a href="#section3">Batch 3</a></li>

</ul>

</nav>

<section id="section1">

<h2>Batch 1</h2>

<p>This is the content of Batch 1.</p>

</section>

<section id="section2">

<h2>Batch 2</h2>

<p>This is the content of Batch 2.</p>

</section>

<section id="section3">

<h2>Batch 3</h2>

<p>This is the content of Batch 3.</p>

</section>

<footer>

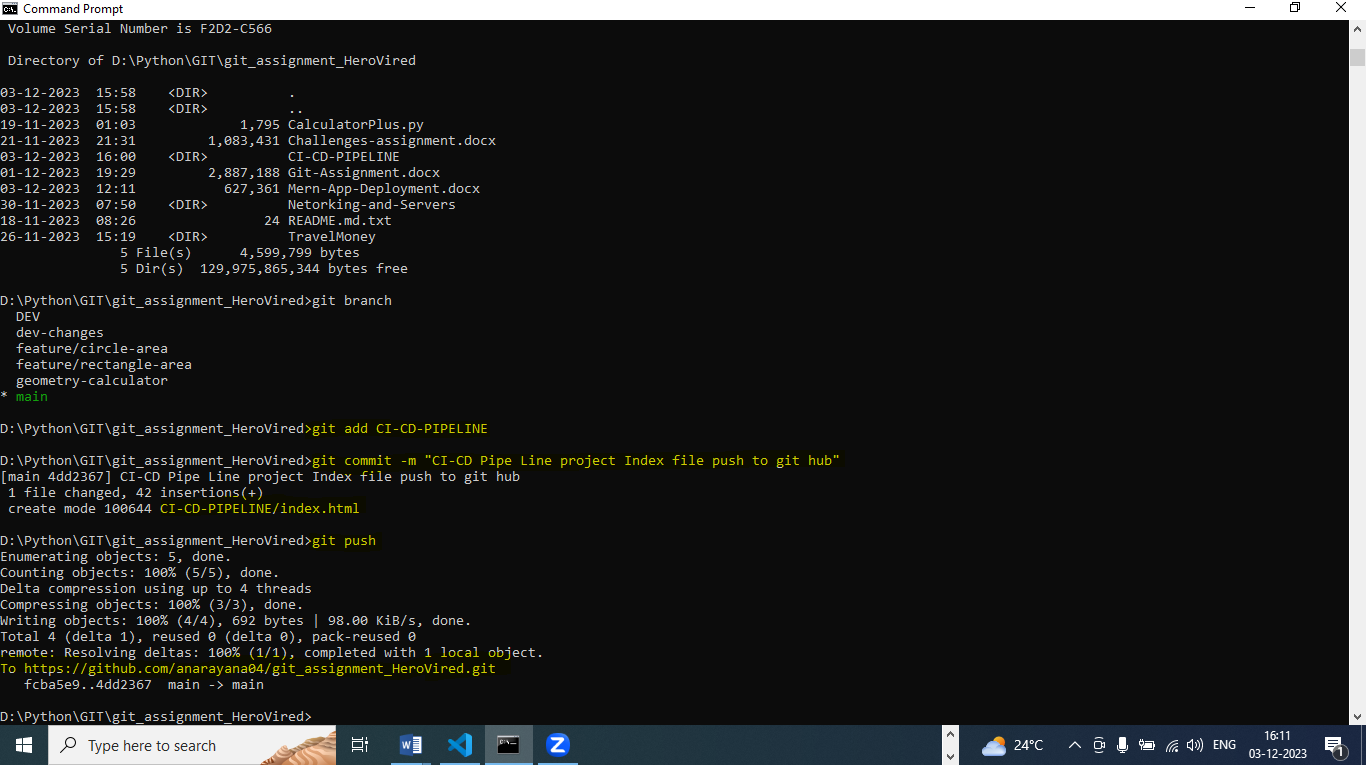
<p>&copy; 2023 HeroVired. All rights reserved.</p>

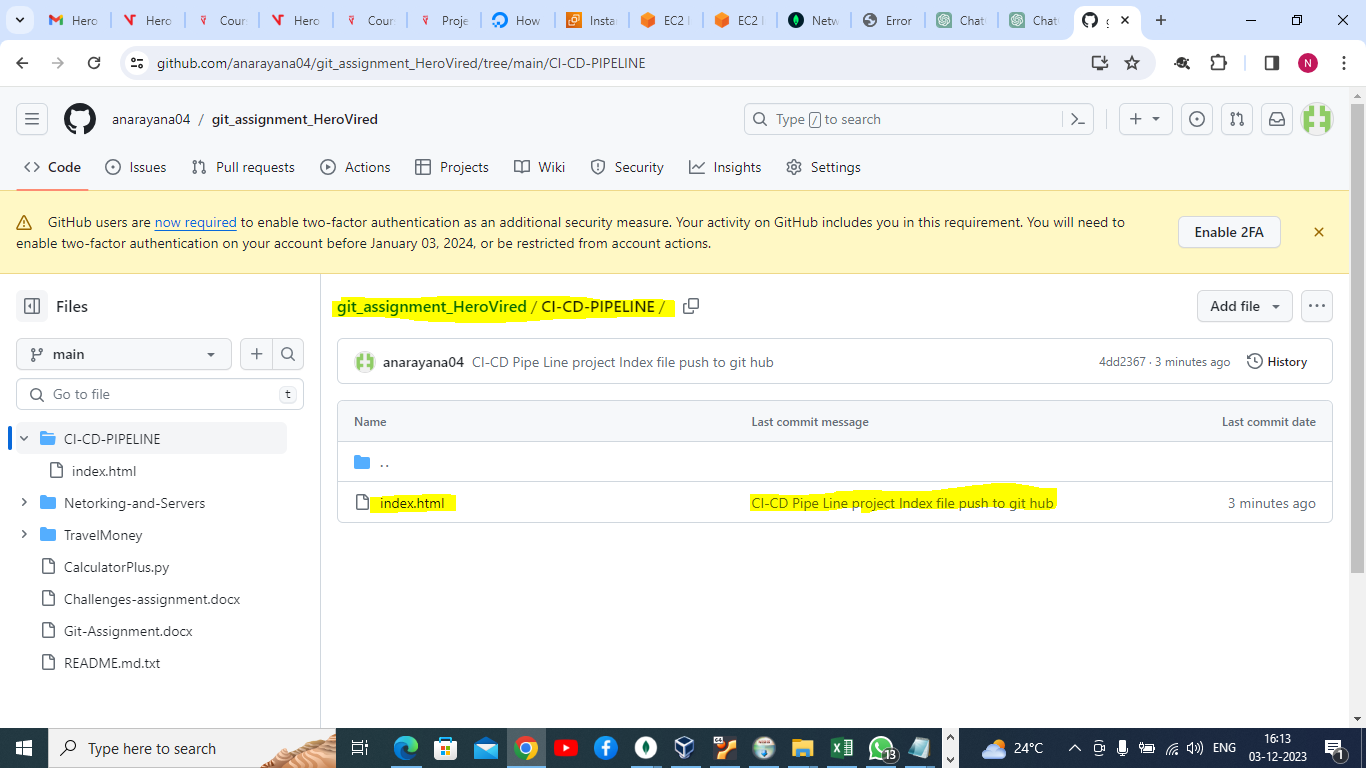
</footer>

</body>

</html>

Going to push to git hub.

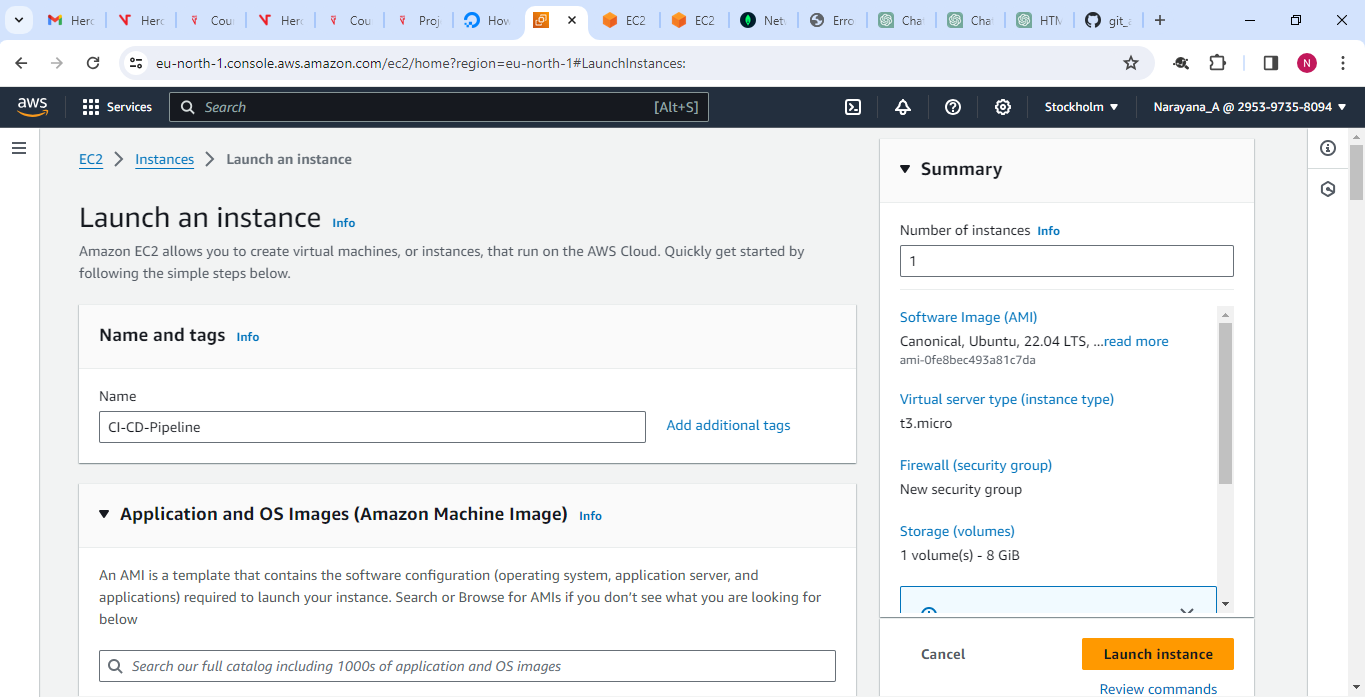




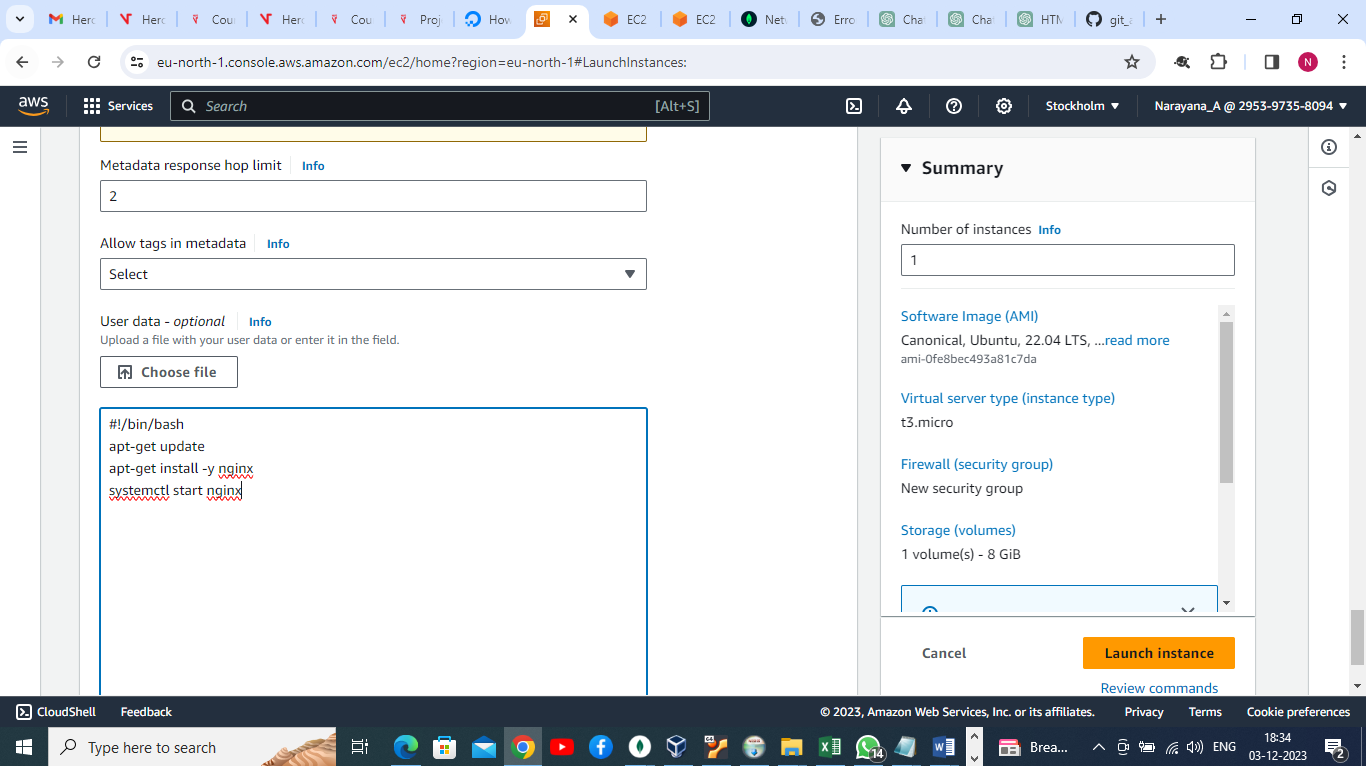
Task1 completed.

Task2:

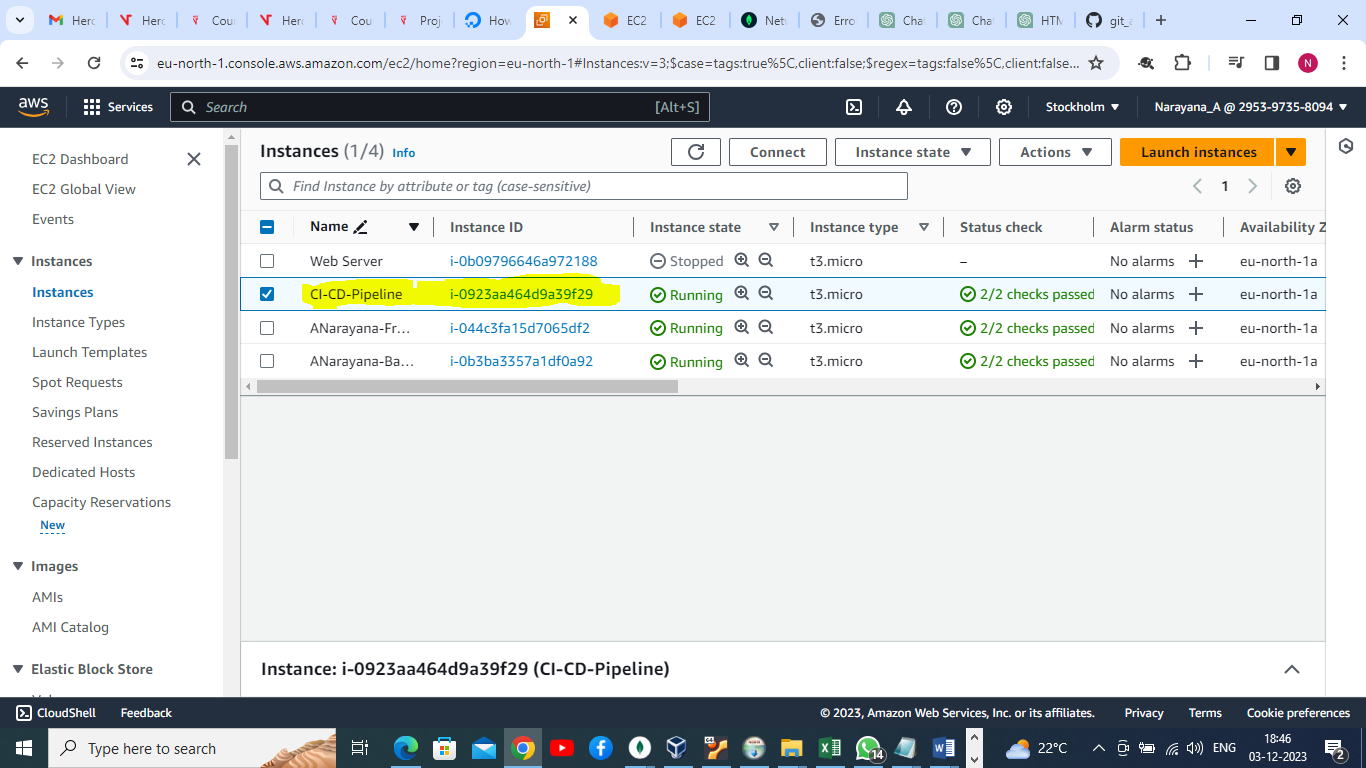
* Task 2: Set Up an AWS EC2/Local Linux Instance with Nginx



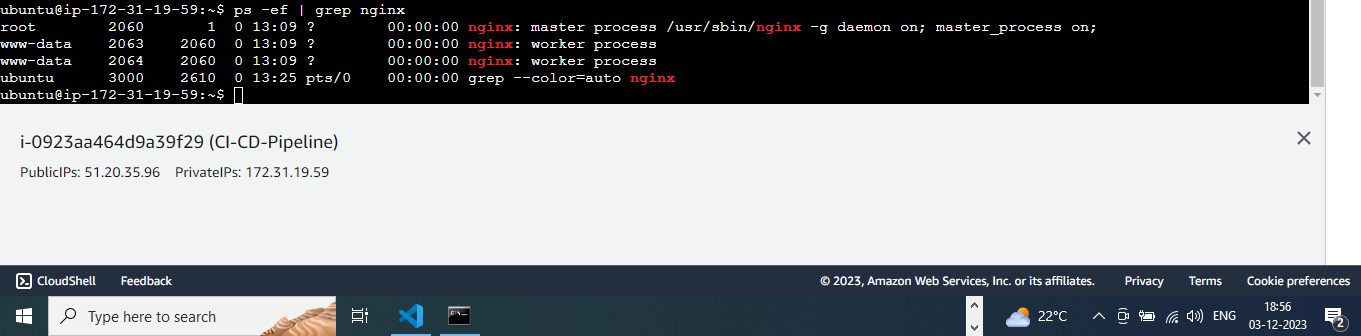
Added shell commands to install nginx and to start nginx service.



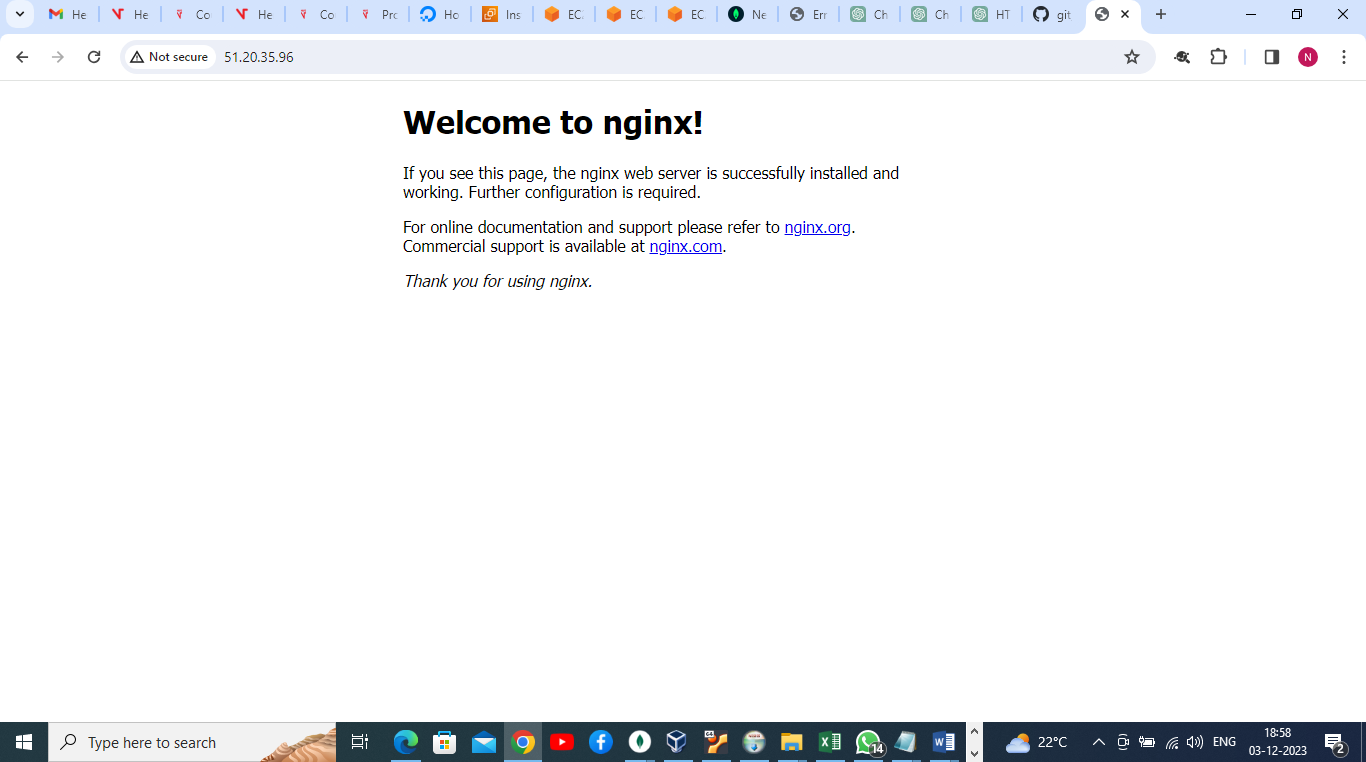
CI-CD-Pipeline instance



NGINX status.



Nginx default web page .



This means nginx successfully installed and running fine.

Task2 completed.

* Task 3: Write a Python Script to Check for New Commits
  + - Create a Python script to check for new commits using the GitHub API.

Python code:

import requests

from datetime import datetime, timedelta

def get\_new\_commits(username, repository, token, since):

    url = f'https://api.github.com/repos/{username}/{repository}/commits'

    headers = {'Authorization': f'token {token}'}

    params = {'since': since}

    response = requests.get(url, headers=headers, params=params)

    if response.status\_code == 200:

        commits = response.json()

        return commits

    else:

        print(f"Failed to fetch commits. Status code: {response.status\_code}")

        return None

def main():

    username = 'anarayana04'

    repository = 'git\_assignment\_HeroVired'

    token = 'github\_pat\_11AK62UOI01Rki6x0nfMy6\_HWU2hxixWUCbPMyKvQ7Id1NWMrL0tM3B9znhEusoRmS6JEIAKC4piohbggm'

    # Set the datetime for the last check (e.g., one day ago)

    since\_datetime = (datetime.now() - timedelta(hours=1)).isoformat()

    commits = get\_new\_commits(username, repository, token, since\_datetime)

    if commits:

        print(f"New commits in {username}/{repository}:")

        for commit in commits:

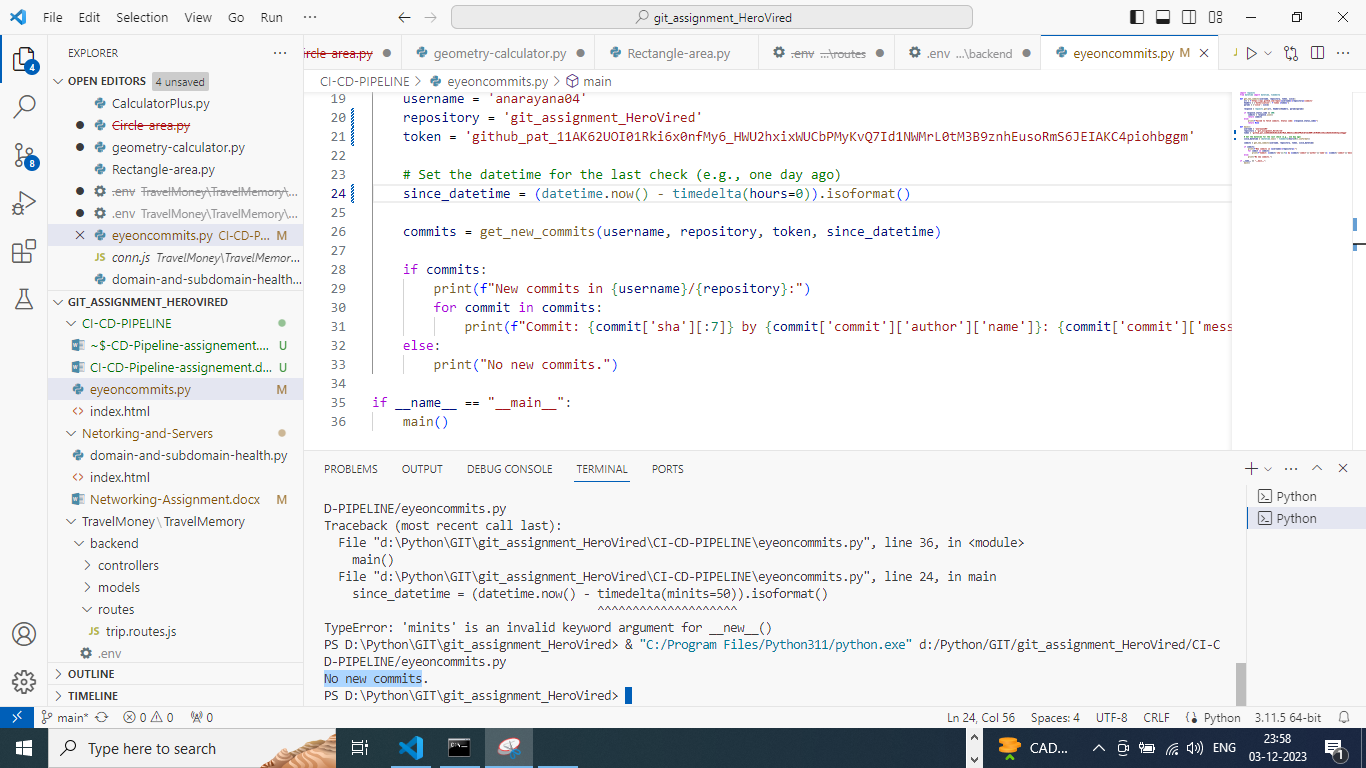
            print(f"Commit: {commit['sha'][:7]} by {commit['commit']['author']['name']}: {commit['commit']['message']}")

    else:

        print("No new commits.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()



After committed new code python script got 200 success status for new updates.

PS D:\Python\GIT\git\_assignment\_HeroVired> & "C:/Program Files/Python311/python.exe" d:/Python/GIT/git\_assignment\_HeroVired/CI-CD-PIPELINE/eyeoncommits.py

No new commits.

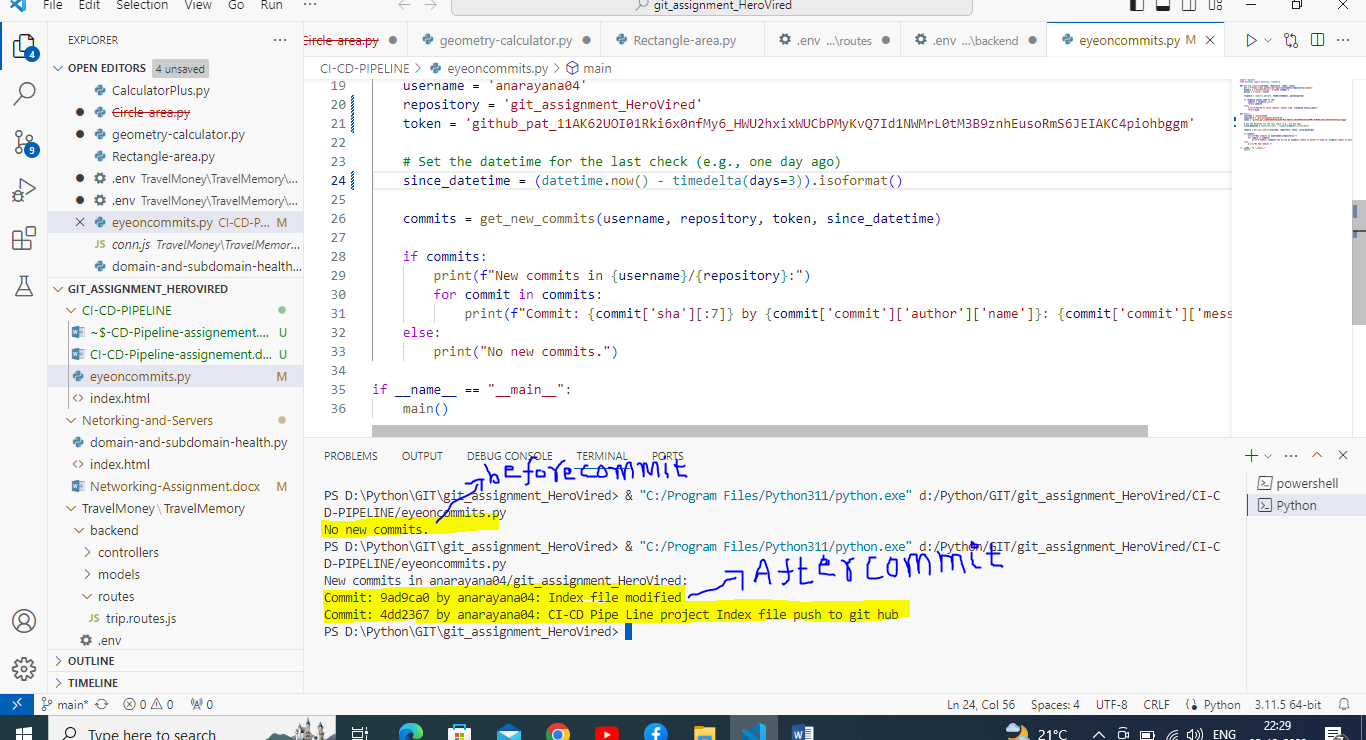
PS D:\Python\GIT\git\_assignment\_HeroVired> & "C:/Program Files/Python311/python.exe" d:/Python/GIT/git\_assignment\_HeroVired/CI-CD-PIPELINE/eyeoncommits.py

New commits in anarayana04/git\_assignment\_HeroVired:

Commit: 9ad9ca0 by anarayana04: Index file modified

Commit: 4dd2367 by anarayana04: CI-CD Pipe Line project Index file push to git hub

PS D:\Python\GIT\git\_assignment\_HeroVired>



Task3 completed.

* Task 4: Write a Bash Script to Deploy the Code
  + - Create a bash script to clone the latest code and restart Nginx.

Shell script code:

#!/bin/bash

# Set your Git repository URL and branch

REPO\_URL="https://github.com/your-username/your-repo.git"

BRANCH="main"

# Set the path to your web application's directory

WEB\_APP\_DIR="/path/to/your/web/app"

# Clone the latest code

echo "Cloning the latest code from $REPO\_URL..."

git clone --branch $BRANCH $REPO\_URL $WEB\_APP\_DIR

# Check if the clone was successful

if [ $? -eq 0 ]; then

echo "Code cloned successfully."

# Restart Nginx

echo "Restarting Nginx..."

sudo systemctl restart nginx

# Check if Nginx restart was successful

if [ $? -eq 0 ]; then

echo "Nginx restarted successfully."

else

echo "Failed to restart Nginx. Please check Nginx configuration."

exit 1

fi

else

echo "Failed to clone the latest code. Please check the Git repository URL and credentials."

exit 1

fi

# Exit with success

exit 0