**Creating CI-CD Pipeline Project which will monitor all the code changes made in an HTML website, codes committed on GitHub using python bash sh script, python code script, and cron job**

* 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: Simple HTML File index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Important HTML Project</title>

</head>

<body>

<h1>Hello, GitHub!</h1>

<p>This web page is used for demo purpose to check CI-CD pipeline works fine or not</p>

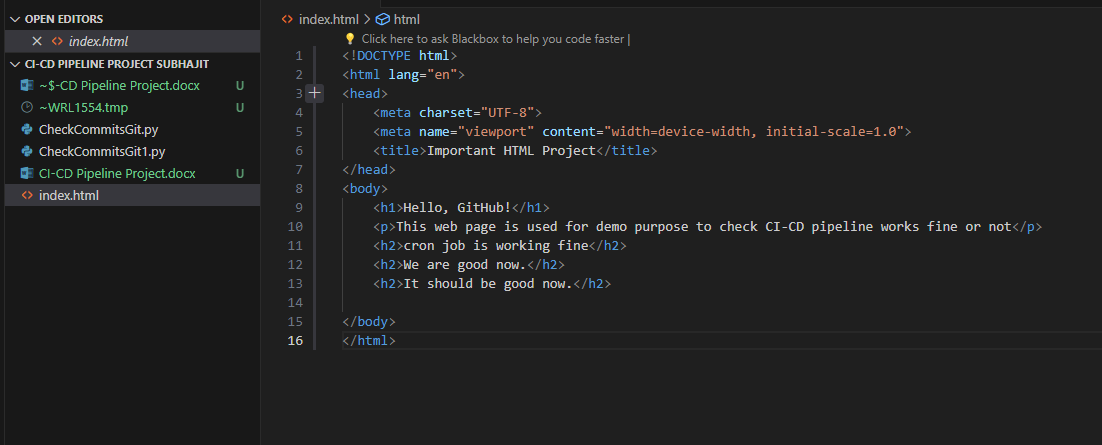
<h2>cron job is working fine</h2>

<h2>We are good now.</h2>

<h2>It should be good now.</h2>

</body>

</html>



Push that file to GitHub Repository - <https://github.com/my-git-subhajit/CI-CD-Pipeline-Project>

A screenshot of a computer

Description automatically generated

Write a Python script to connect with GitHub API and track the new changes committed to GitHub.

It includes code changes in HTML file and push the new code or updated HTML file to GitHub.

!! Make sure your GitHub repository is public !!

Python Script – CheckCommitsGit.py

A screen shot of a computer program

Description automatically generated

Final python code deployed on AWS EC2 instance

import requests

import json

import subprocess

def get\_commits(repo\_url):

response = requests.get(repo\_url)

#check the commit

if response.status\_code == 200:

commits = response.json()

return commits

else:

raise Exception("Could not get commits")

def main():

repo\_url = "https://api.github.com/repos/my-git-subhajit/CI-CD-Pipeline-Project/commits"

# Get the latest commits.

commits = get\_commits(repo\_url)

if commits:

sha\_latest = str(commits[0]["sha"])

# sha1\_latest= str(commits[1]["sha"])

if sha\_latest:

subprocess.call(['sh', '/home/ubuntu/deploy.sh'])

#print (sha\_latest)

#print(sha\_remote)

else:

print("No Commits found.")

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

main()

