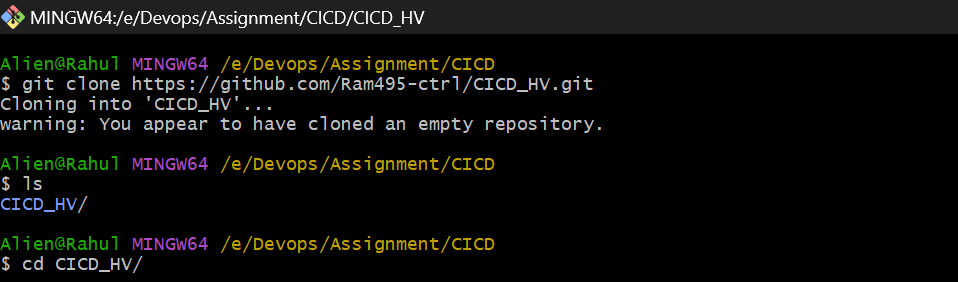
* **Task 1:** Set Up a Simple HTML Project
  + - Create a simple HTML project and push it to a GitHub repository.
* **Task 2:** Set Up an AWS EC2/Local Linux Instance with Nginx
* **Task 3:** Write a Python Script to Check for New Commits
  + - Create a Python script to check for new commits using the 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.

**Github URL: https://github.com/Ram495-ctrl/CICD\_HV.git**

# TASK1

**Created a new repository in GITHUB and clone**

****

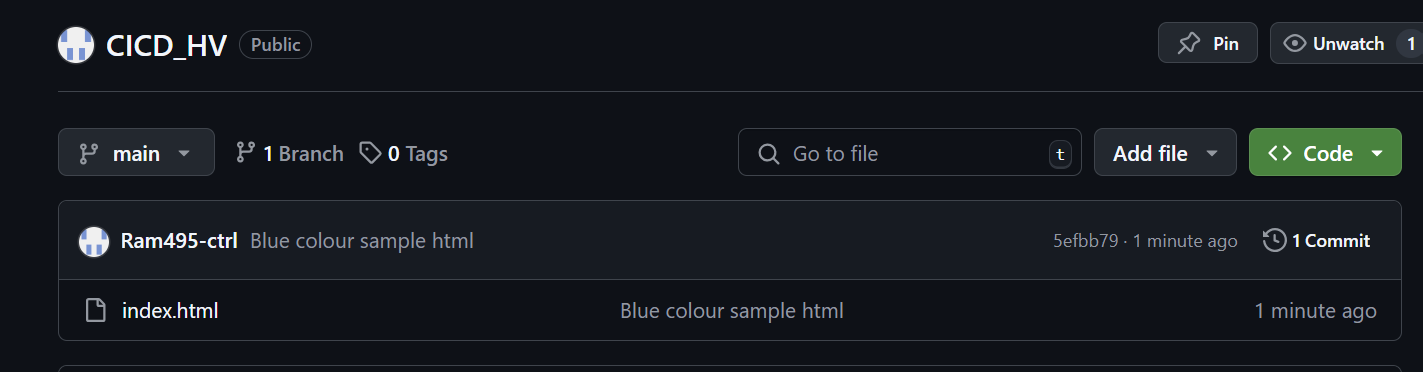
Create index.html file



**Commit and push to main**



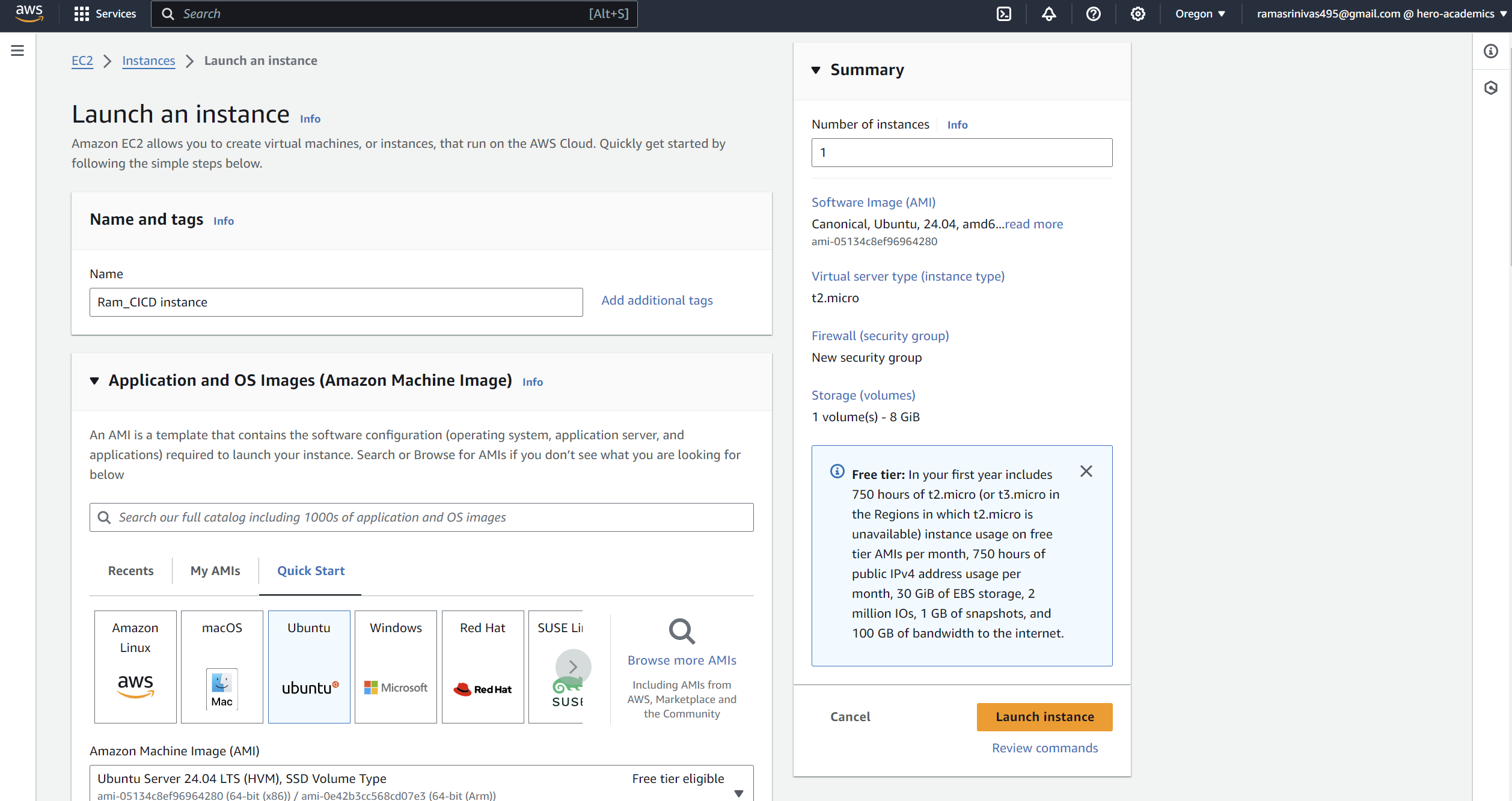
**Can see in Git remote**

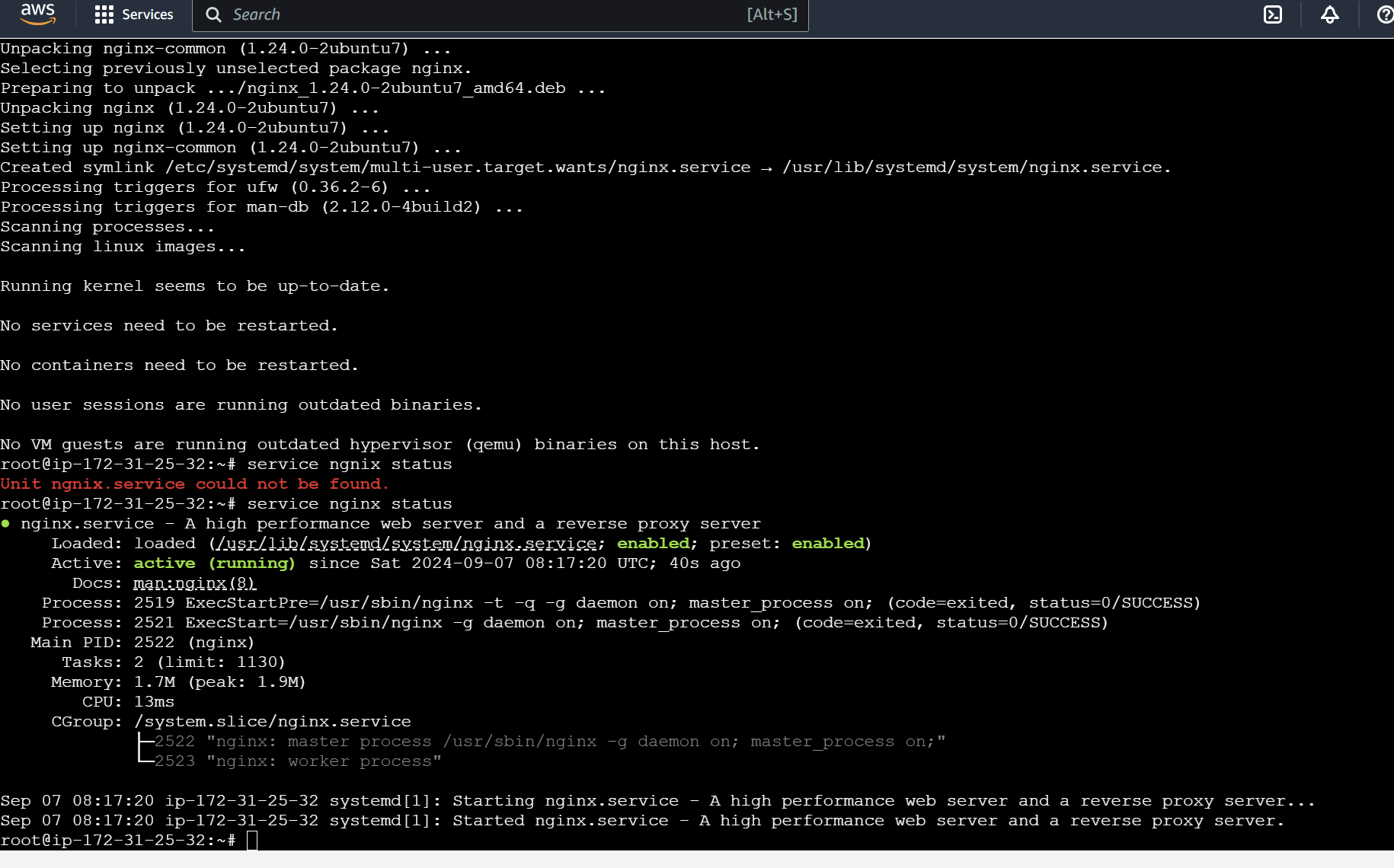


Task 2**:** Set Up an AWS EC2/Local Linux Instance with Nginx deploying the app1

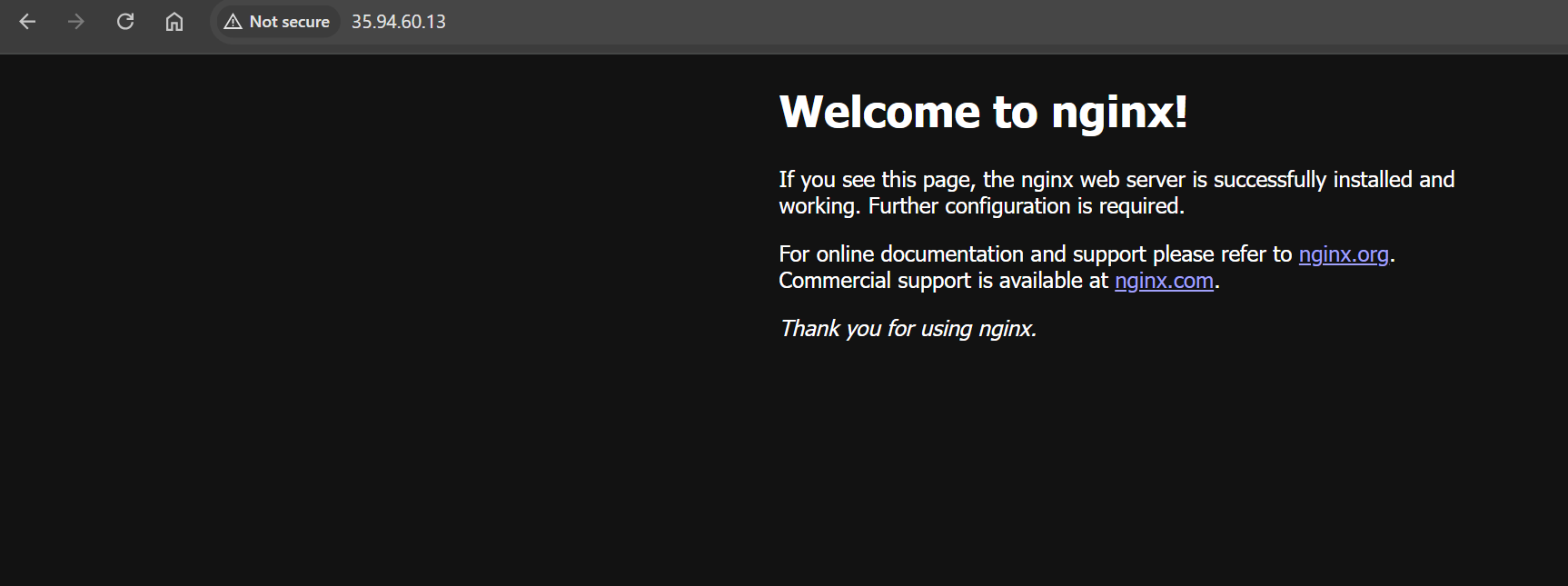
**Create an Ec2 Instance running and hosting our simple html in app1**

**Im running all instructions as part of the start up script for ease and the startup script is shown below**





PublicIP: 35.94.60.13



**Python script to check new commits:**

import os, logging

import subprocess

from github import Github

from github import GithubException

#Configuration

GITHUB\_TOKEN = "ghp\_aq7BI4kHlP3wmPZzAwyuDrKwwXHpcH2PFejw"

GITHUB\_REPO = "Ram495-ctrl/CICD\_HV"

LOCAL\_REPO\_PATH = "/home/ubuntu/CICD\_HV"

# Initialize GitHub API client

try:

    g = Github(GITHUB\_TOKEN)

    repo = g.get\_repo(GITHUB\_REPO)

except GithubException:

    logging.exception("An exception was thrown!")

    exit(1)

def get\_latest\_github\_commit():

    try:

        commits = repo.get\_commits()

        latest\_commit = commits[0].sha

        return latest\_commit

    except GithubException as e:

        print(f"Error fetching latest commit from GitHub: {e}")

        exit(1)

def get\_latest\_local\_commit():

    try:

        os.chdir(LOCAL\_REPO\_PATH)

        result = subprocess.run(["git", "rev-parse", "HEAD"], capture\_output=True, text=True)

        return result.stdout.strip()

    except subprocess.CalledProcessError as e:

        print(f"Error getting latest local commit: {e}")

        exit(1)

def pull\_latest\_changes():

    try:

        os.chdir(LOCAL\_REPO\_PATH)

        subprocess.run(["git", "pull"], check=True)

    except subprocess.CalledProcessError as e:

        print(f"Error pulling latest changes: {e}")

        exit(1)

def main():

    try:

        latest\_github\_commit = get\_latest\_github\_commit()

        latest\_local\_commit = get\_latest\_local\_commit()

        print(f"Latest GitHub commit ID: {latest\_github\_commit}")

        print(f"Latest local commit ID: {latest\_local\_commit}")

        if latest\_github\_commit != latest\_local\_commit:

            print("New commit detected. Pulling latest changes...")

            pull\_latest\_changes()

        else:

            print("Local repository is up to date.")

    except Exception as e:

        print(f"An unexpected error occurred: {e}")

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

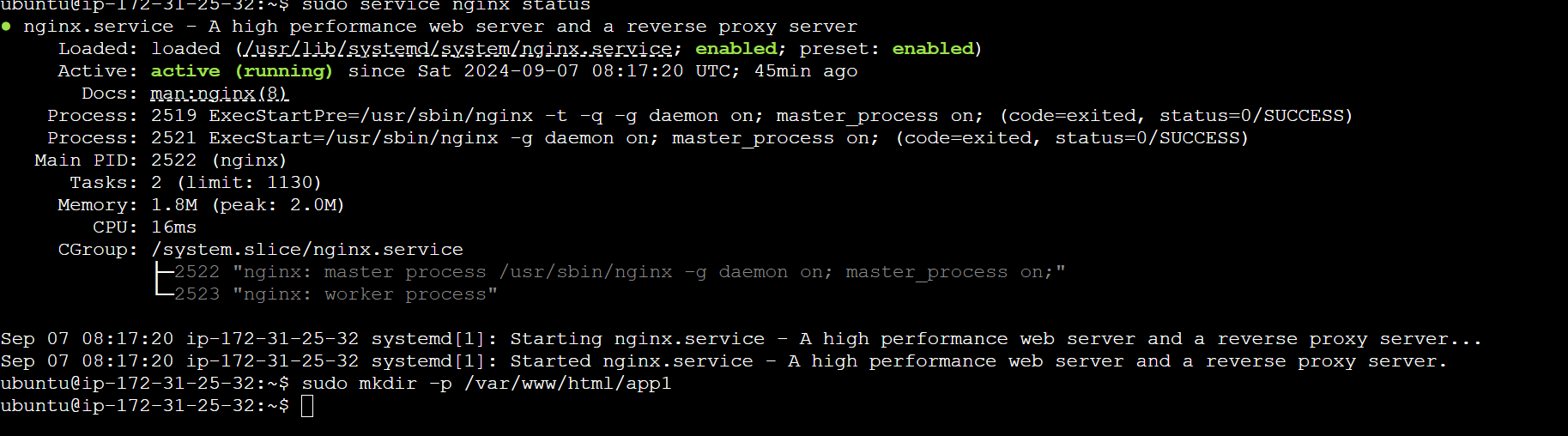
    print("main")

    main()

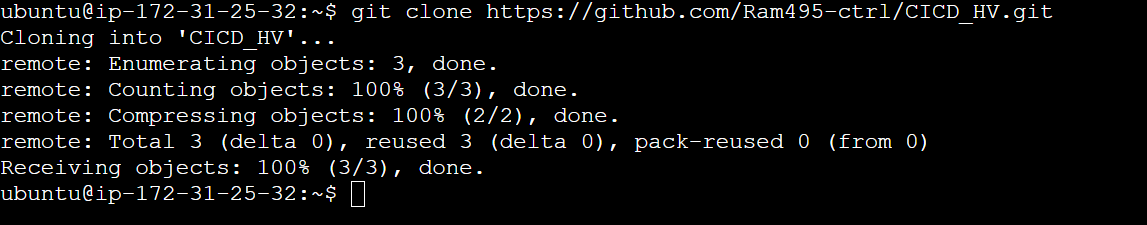
# Create the /var/www/html/app1 directory if it doesn't exist

# Give necessary permissions to the /var/www/html/app1 directory

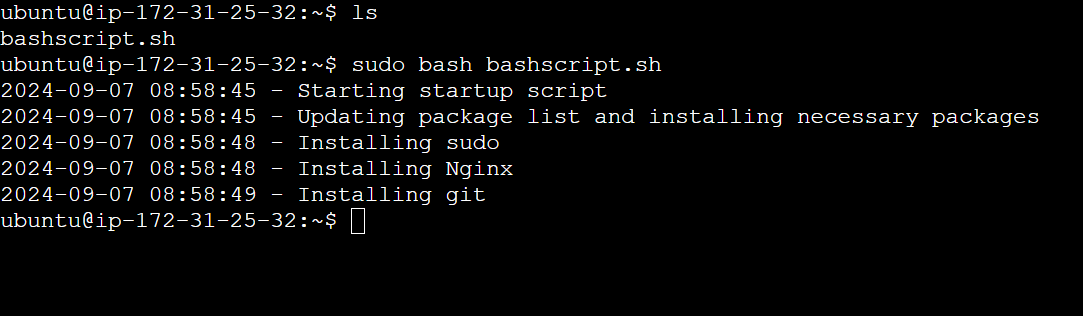
# sudo chmod -R 755 /var/www/html/app1

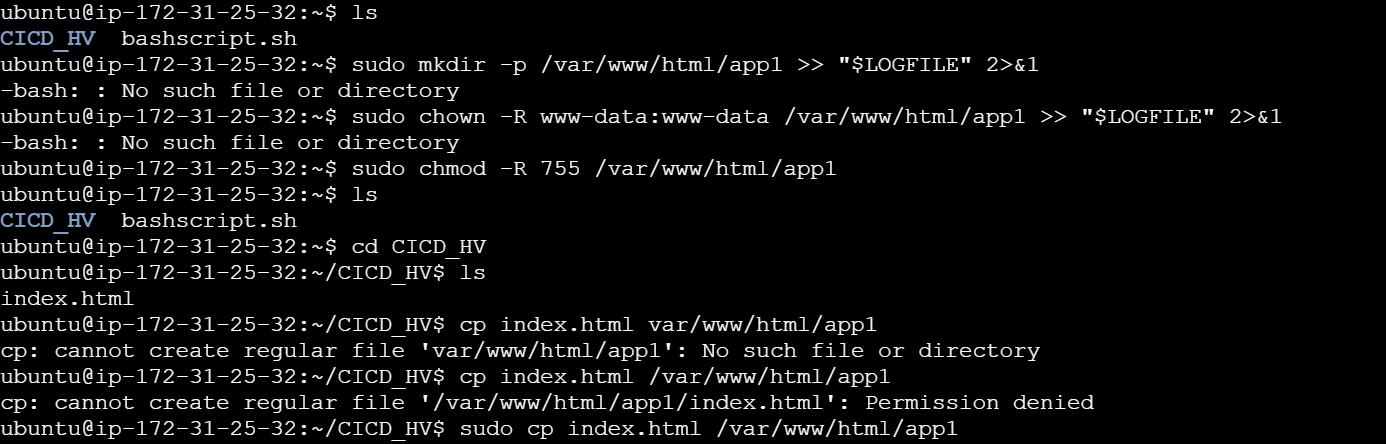
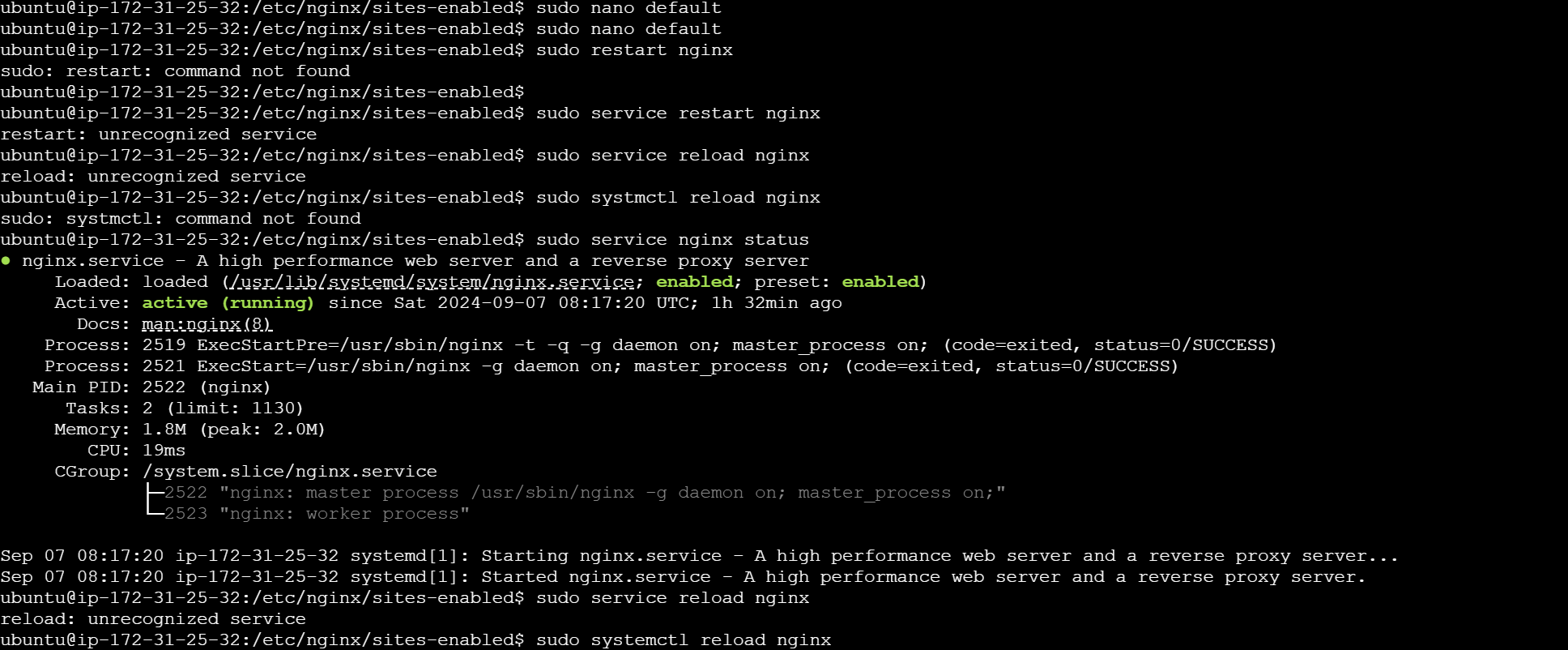


**Do Git clone in EC2:**



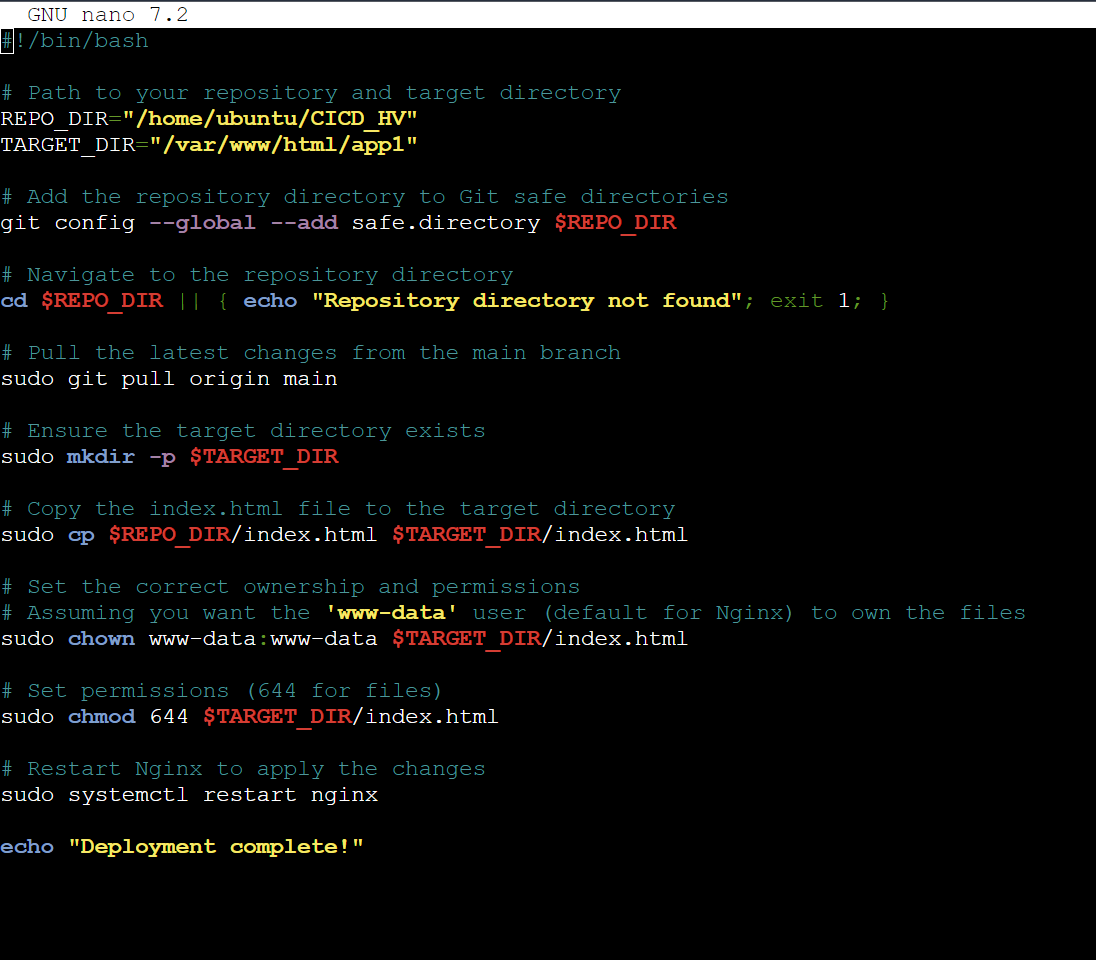
**Create bashscript.sh file**





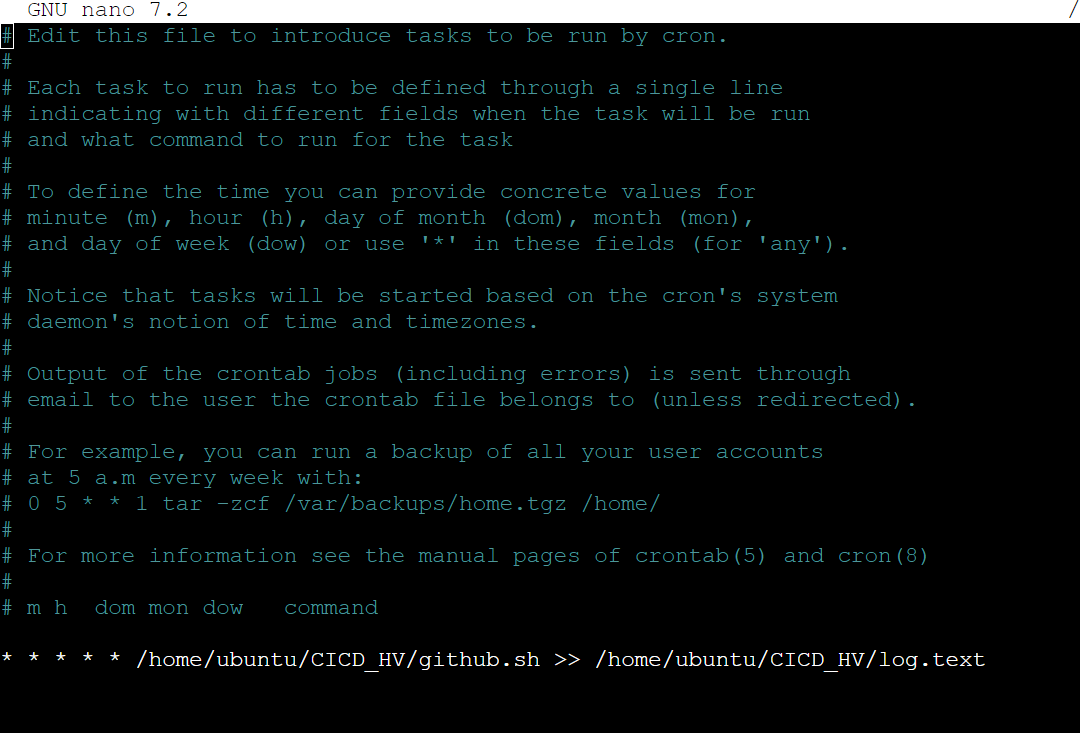
**Create Bash file**



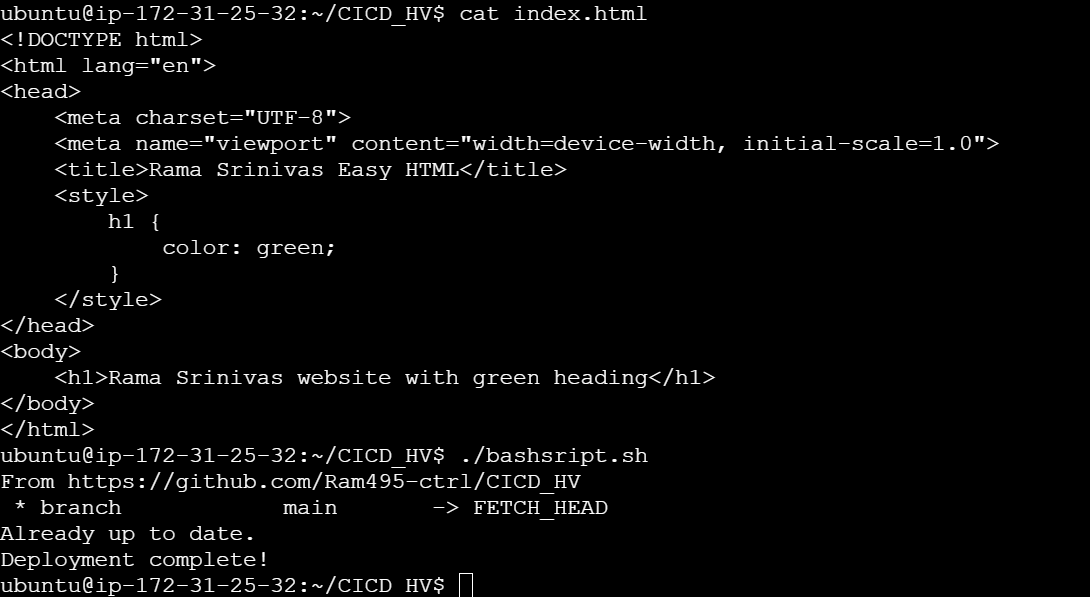
**Changing the html and pushing the Green colour code to GITHUB which should be picked up automatically now by the cronjob**



**Cronjob to run every minute to check for commits**

****

**Test now changing the html and pushing the Green colour code to GITHUB which should be picked up automatically now by the cronjob**



**Finally colour is changed to Green automatically once the shell script is executed**

