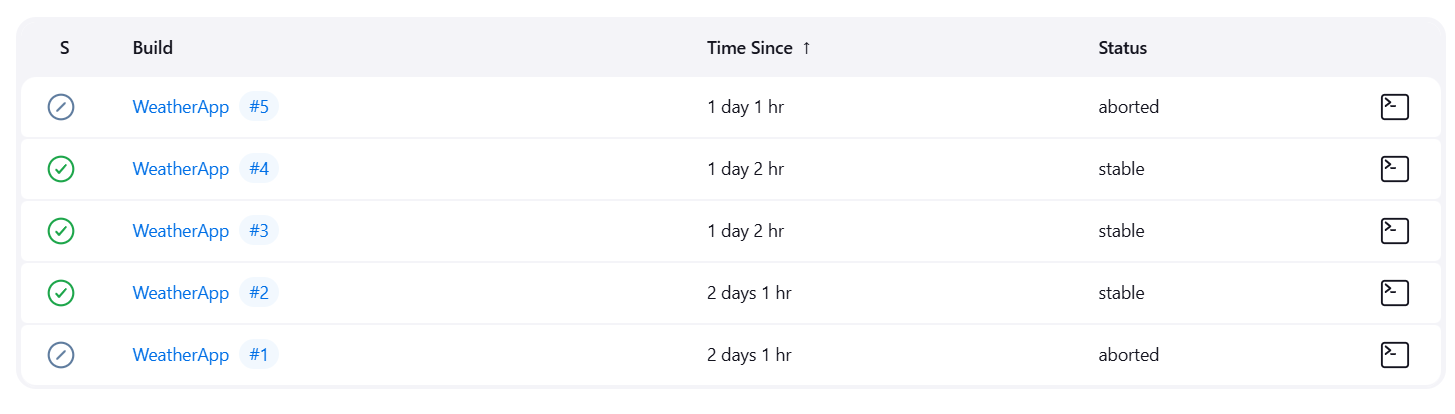
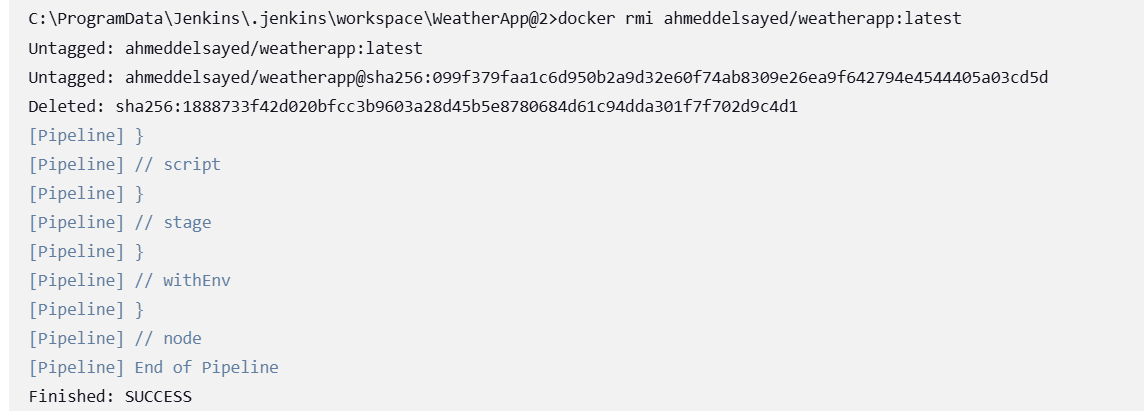
What is This Project About?

The goal of this project is to automate the process of building, testing, and deploying a web application. Here’s what I did:

1. Pushed Python code to a private GitHub repository.
2. Created a Dockerfile to containerize the application.
3. Set up two virtual machines using WSL (Windows Subsystem for Linux) and Hyper-V instead of Vagrant.
4. Configured a Jenkins pipeline to automate the entire process:
   * Pulling code from GitHub.
   * Building a Docker image.
   * Pushing the Docker image to Docker Hub.
   * Using Ansible to:
     + Install Docker on the virtual machines.
     + Pull the Docker image from Docker Hub.
     + Run the Docker container on the virtual machines.

1. Jenkins Pipeline Output

This is what the Jenkins pipeline looks like when it runs successfully. It shows each step of the process, from pulling the code to deploying the application.

**And this is the pipeline script :**

**pipeline {**

**agent any**

**environment {**

**GITHUB\_REPO = "github.com/ahmed-elsayed690/weatherapp.git"**

**DOCKER\_IMAGE = "ahmeddelsayed/weatherapp:latest"**

**DOCKER\_CREDENTIALS = "docker-hub-credentials" // Fixed typo in variable name**

**GITHUB\_CREDENTIALS = "github-token"**

**}**

**stages {**

**stage('Checkout Code') {**

**steps {**

**script {**

**withCredentials([string(credentialsId: GITHUB\_CREDENTIALS, variable: 'GITHUB\_TOKEN')]) {**

**bat '''**

**if exist repo rmdir /s /q repo**

**git clone https://%GITHUB\_TOKEN%@%GITHUB\_REPO% repo**

**'''**

**}**

**}**

**}**

**}**

**stage('Build Docker Image') {**

**steps {**

**script {**

**bat "docker build -t ${DOCKER\_IMAGE} ./repo"**

**} } }**

**stage('Login to Docker Hub') {**

**steps {**

**script {**

**withCredentials([usernamePassword(credentialsId: DOCKER\_CREDENTIALS, usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {**

**bat '''**

**echo %DOCKER\_PASS% | docker login -u %DOCKER\_USER% --password-stdin**

**'''**

**}**

**}**

**} }**

**stage('Push to Docker Hub') {**

**steps {**

**script {**

**bat "docker push ${DOCKER\_IMAGE}"**

**}**

**} }**

**stage('Cleanup') {**

**steps {**

**script {**

**bat "rmdir /s /q repo" // Use rmdir instead of rm for Windows**

**bat "docker rmi ${DOCKER\_IMAGE}"**

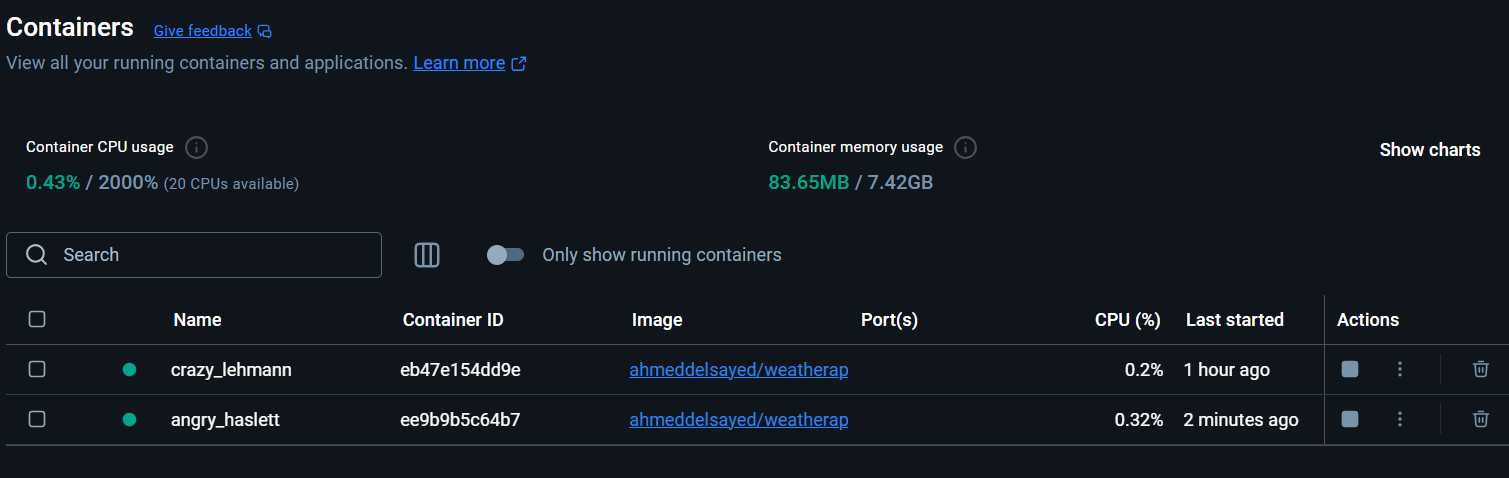
**}**

**} }**

-------------------------------------------------------------------------------------------------------------------------------

**Docker Image in Docker Hub**

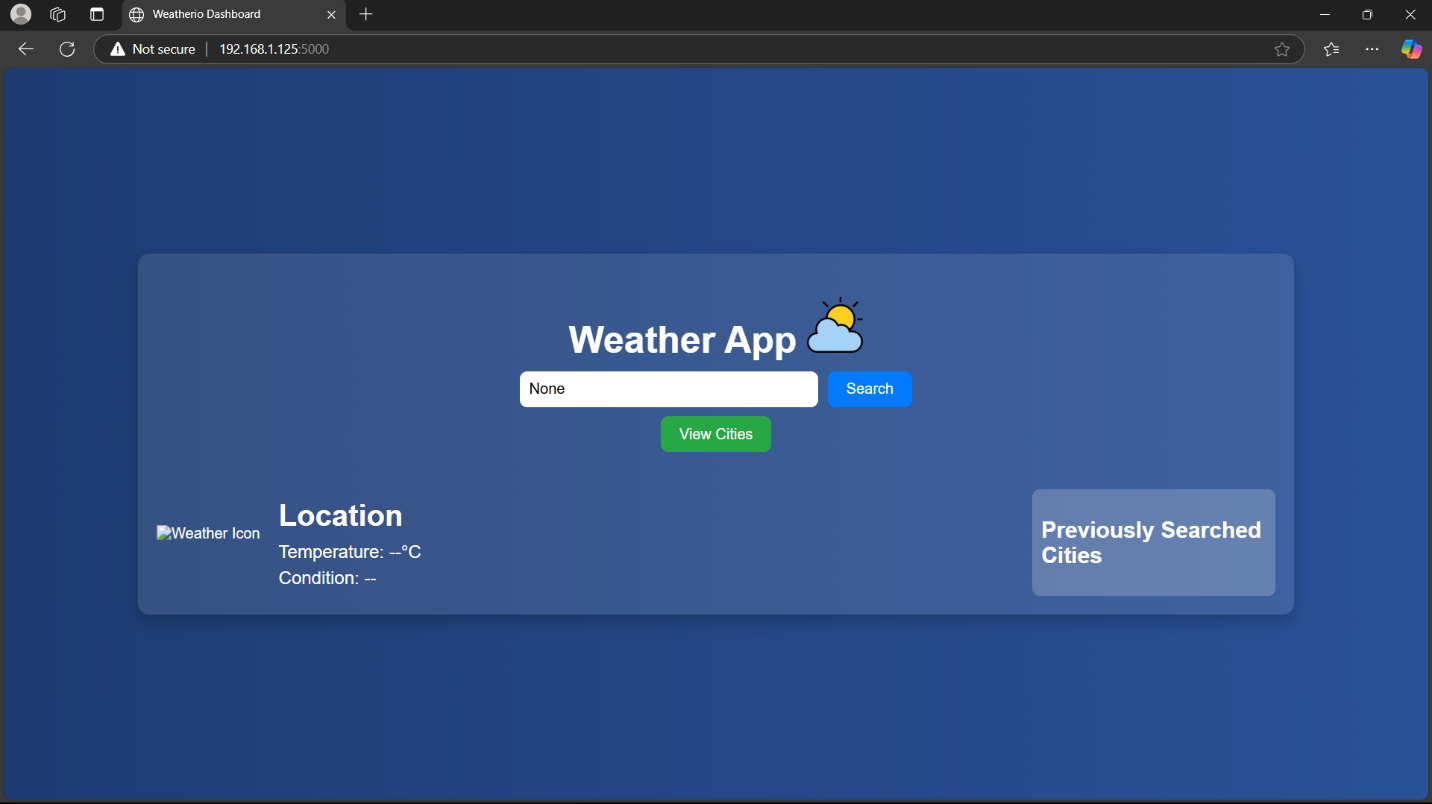
After the Jenkins pipeline builds the Docker image, it gets pushed to Docker Hub. Here’s a screenshot of the image in my Docker Hub repository.



The docker output from ansible in wsl :



On the browser :



**How I did it step by step :**

1. **Pushed Python Code to GitHub**:
   * I started by pushing the provided Python code to my private GitHub repository. This is where Jenkins will pull the code from.
2. **Created a Dockerfile**:
   * I wrote a Dockerfile to containerize the application. This file tells Docker how to build the application into an image.
3. **Set Up Virtual Machines with WSL and Hyper-V**:
   * Instead of using Vagrant, I set up two virtual machines using **WSL (Windows Subsystem for Linux)** and **Hyper-V**. These machines act as the "target" servers where the application will be deployed.
4. **Configured Jenkins Pipeline**:
   * I set up a Jenkins pipeline to automate everything. Here’s what the pipeline does:
     + **Pulls the code** from GitHub (I had to use a credentials token for this).
     + **Builds the Docker image** using the Dockerfile.
     + **Pushes the Docker image** to Docker Hub.
     + **Runs an Ansible playbook** to:
       - Install Docker on the virtual machines.
       - Pull the Docker image from Docker Hub.
       - Run the Docker container on the virtual machines.