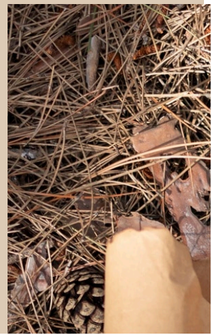
Jenkins Master slave Configurations

PROJECT-4

Presented By

Pravalika Amgoth



Workflow Overview

Method-1

1.Installation of Nginx

2.Installation of HTTPD

3.Installation of Sonarqube

4.Installation of Tomcat

5.Static application deployment

Method-2

1.python application deployment-manual method

2.Using Bash script

3.Using Terraform

4.Java application deployment-manual method

5.Using Bashscript

What is Jenkins master-slave configuration

TheJenkins Masteracts as the central server that manages the build pipelines, schedules jobs, and coordinates tasks. TheSlaves(agents) are distributed nodes that execute the tasks assigned by the master, enabling parallelism and reducing load on the master.

-- Master schedules and monitors builds.

-- Slaves handle heavy workloads by running jobs, allowing scalability and efficient resource utilization.

Method-1

1.Nginx Installation

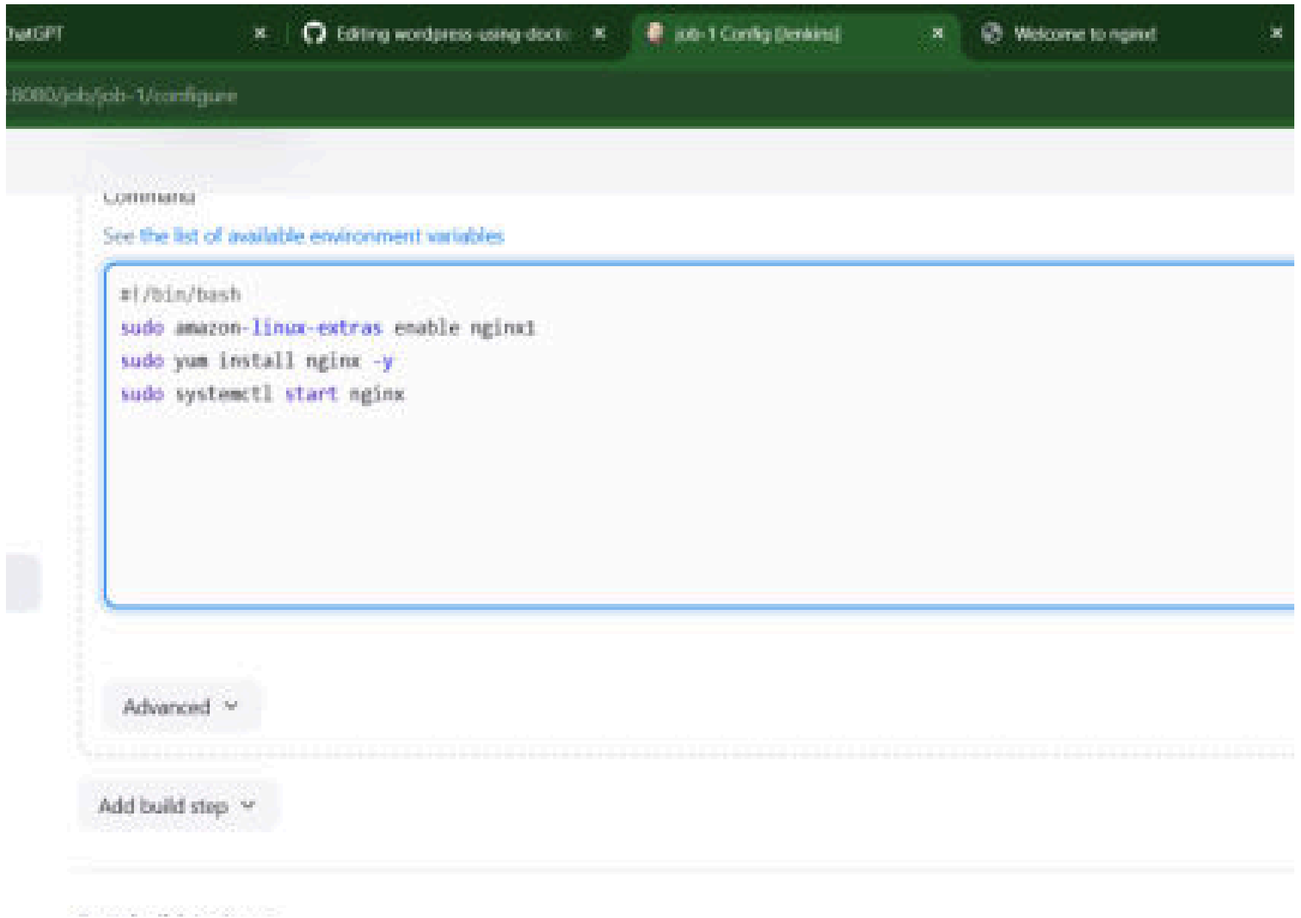
Step 1: Launch Master and Slave Instances

Step 2: Install Jenkins on Master

Step 3: Configure Jenkins and Add Nodes

Step 4: Create a Job in Jenkins

Step 5: Configure Shell Commands to Install Nginx



step 6: Run the Job

Step 7: Access Nginx on the Slave Instance

2.HTTPD Installation

Step 1: Launch Master and Slave Instances

Step 2: Install Jenkins on Master

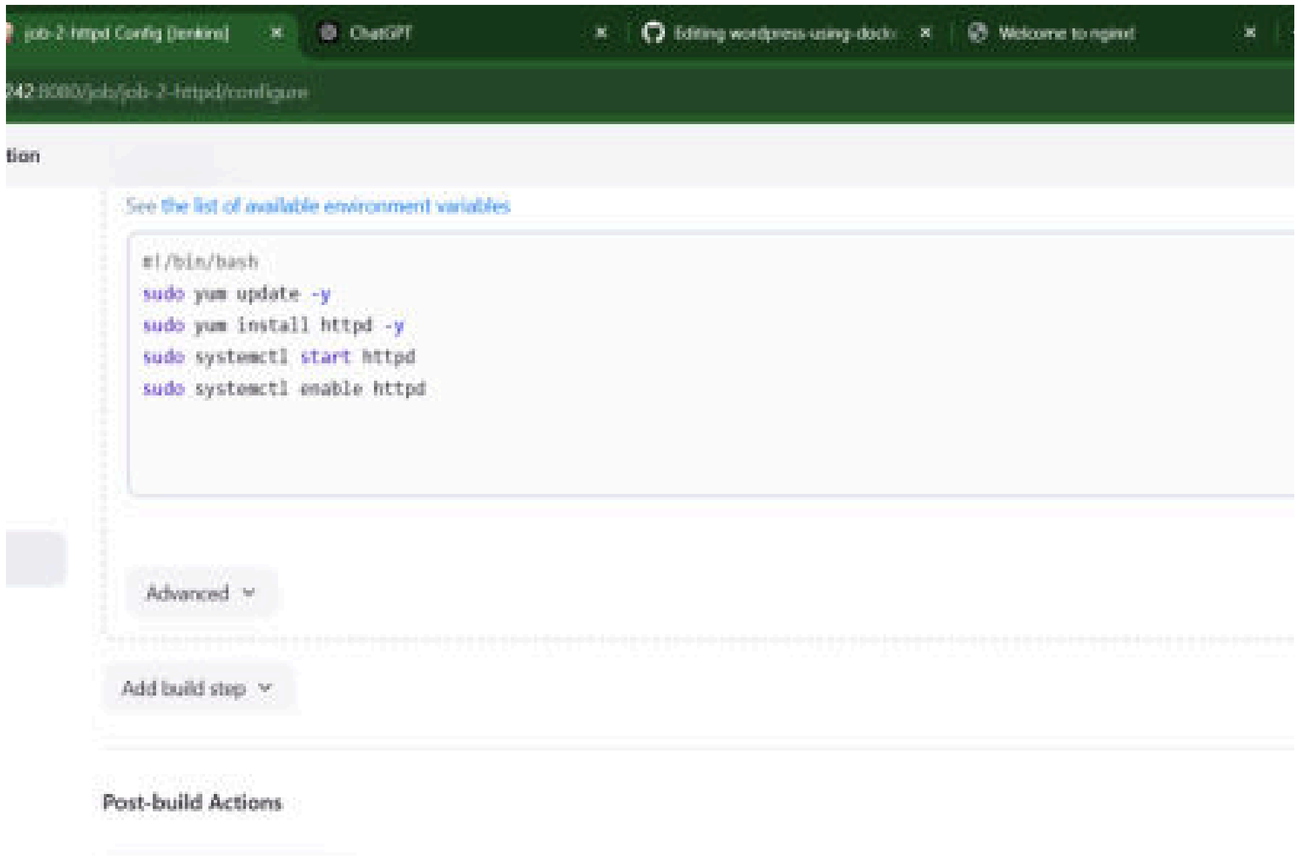
Step 3: Configure Jenkins and Add Nodes

Step 4: Create a Job in Jenkins

Step 5: Configure Shell Commands to Install HTTPD

Step 6: Run the Job

Step 7: Access HTTPD on the Slave Instance



3.Sonarqube Installation

Step 1: Launch Master and Slave Instances-t2.medium

Step 2: Install Jenkins on Master Instance

Step 3: Configure Jenkins and Add Slave Node

Step 4: Create a Jenkins Job

Step 5: Configure Shell Commands to Install sonarqube

```
wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.9.6.50800.zip
unzip sonarqube-8.9.6.50800.zip
mv sonarqube-8.9.6.50800 sonar
cd sonar/bin/linux-x86-64
chmod +x sonar.sh
sh sonar.sh start
tail -f /home/ec2-user/workspace/slave3-job/sonar/logs/sonar.20250107.log
```

Step 6: Access SonarQube on Slave Instance:http://<Slave_IP>:9000

4.Tomcat Installation

Step 1: Launch Master and Slave Instances

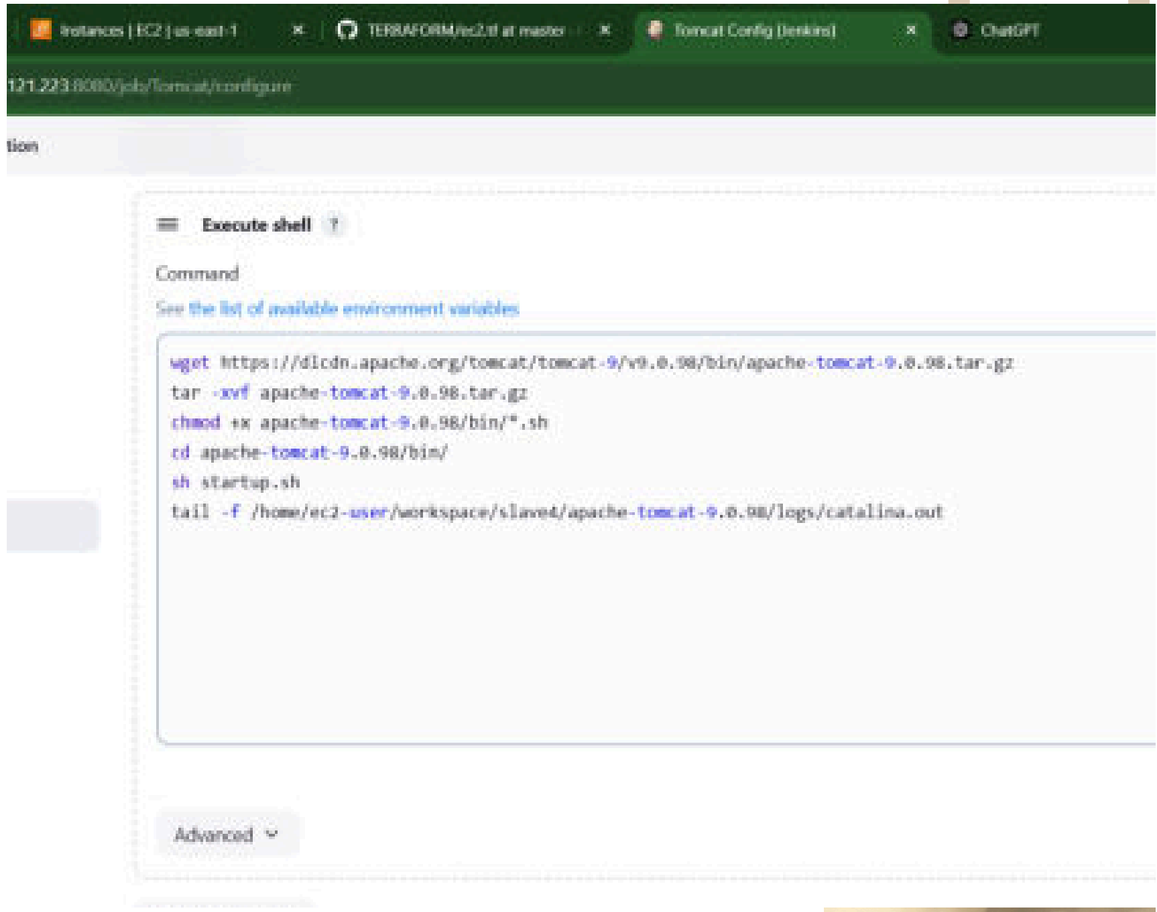
Step 2: Install Jenkins on Master Instance

Step 3: Configure Jenkins and Add Slave Node

Step 4: Create a Jenkins Job

Step 5: Configure Shell Commands to Install tomcat

Step 6: Access tomcat on Slave Instance:http://:8080



5.Static Application Deployment

Step 1: Launch Master and Slave Instances

Step 2: Install Jenkins on Master Instance

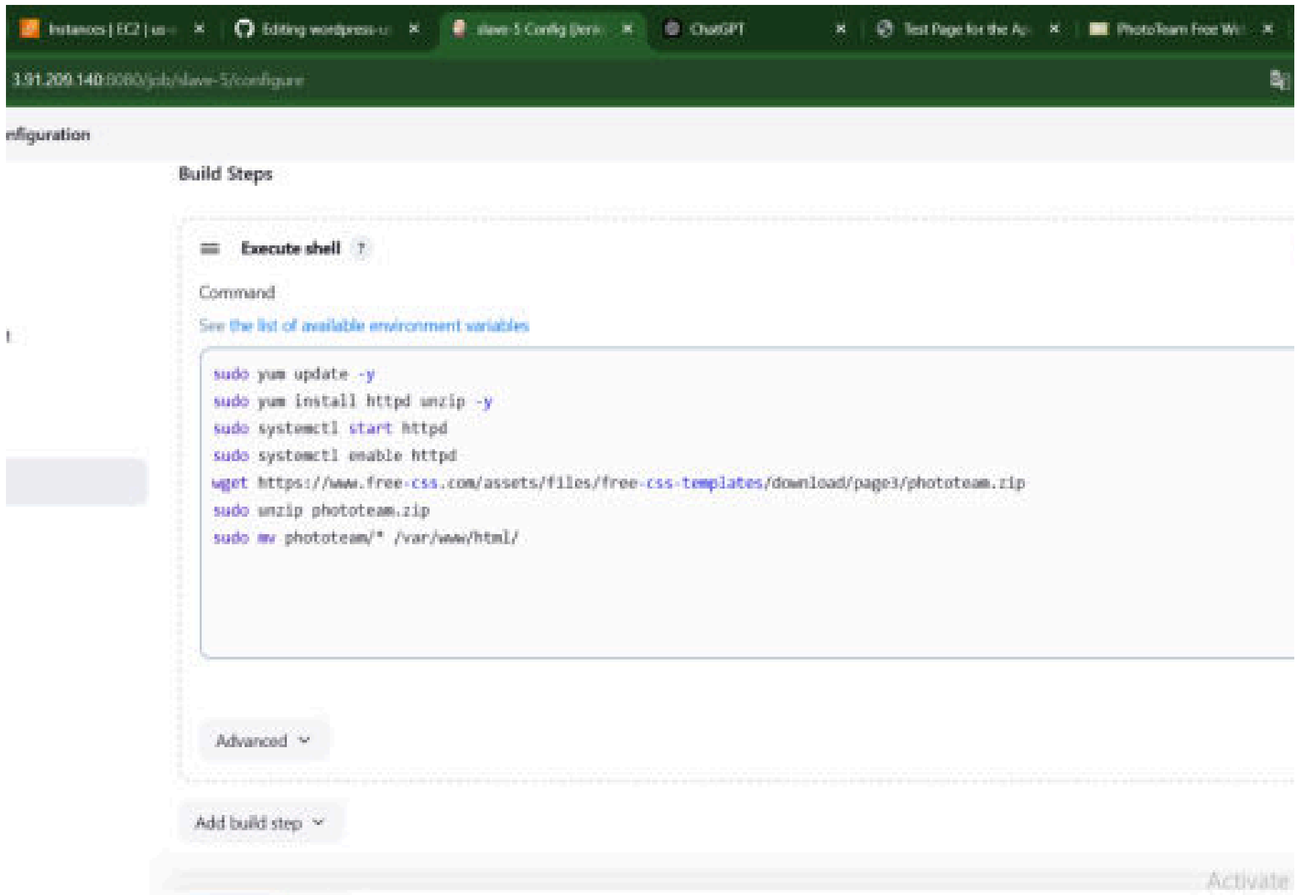
Step 3: Configure Jenkins and Add Slave Node

Step 4: Create a Jenkins Job

Step 5: take any static website from free css templates

Step 6: Configure Shell Commands to Install sonarqube

step 7:access the static application by using slave public ip address



Method-2

1.Python web application -manual process

Step 1: Launch Master and Slave Instances

step 2:Install jenkins in master and python3-pip in both master and slave

step 3:Setup jenkins and add the slavenode

step 4:Create a jenkins job

step 5:Place the github url which consist of the python application code

step 6: In execute shell :pip3 install -r requirements.txt,python app.py

step 7:Access the application using slave public ip:and the node port which is present in the code

Using Bash Script

Step 1: Launch Master and Slave Instances

Step 2: Install Jenkins on Master Instance

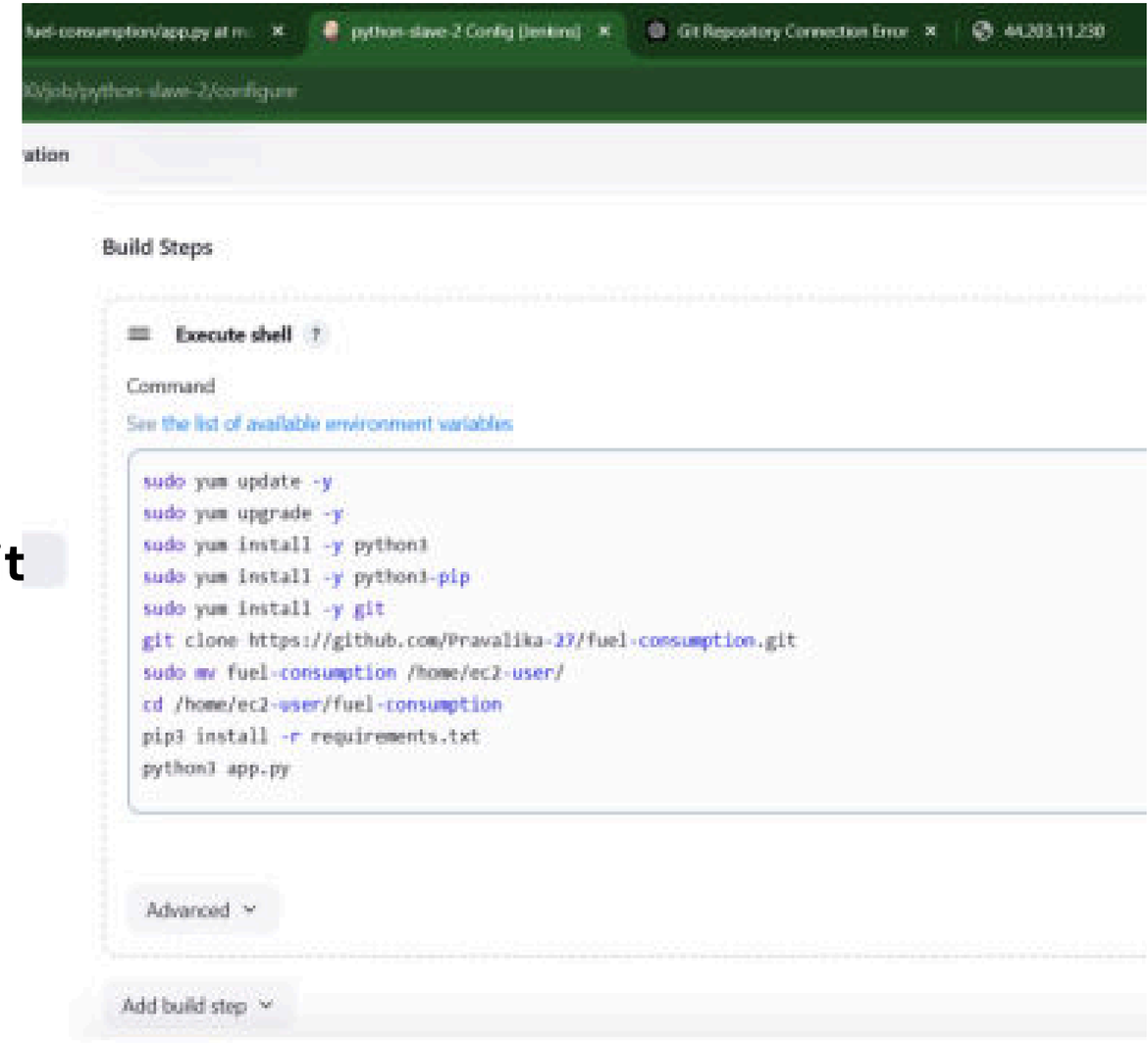
Step 3: Configure Jenkins and Add Slave Node

Step 4: Create a Jenkins Job

Step 5: Configure Shell Commands :add python,git installation and clone the repository

step 6:Build the job

step 7: Access the application using slave ip and the node port



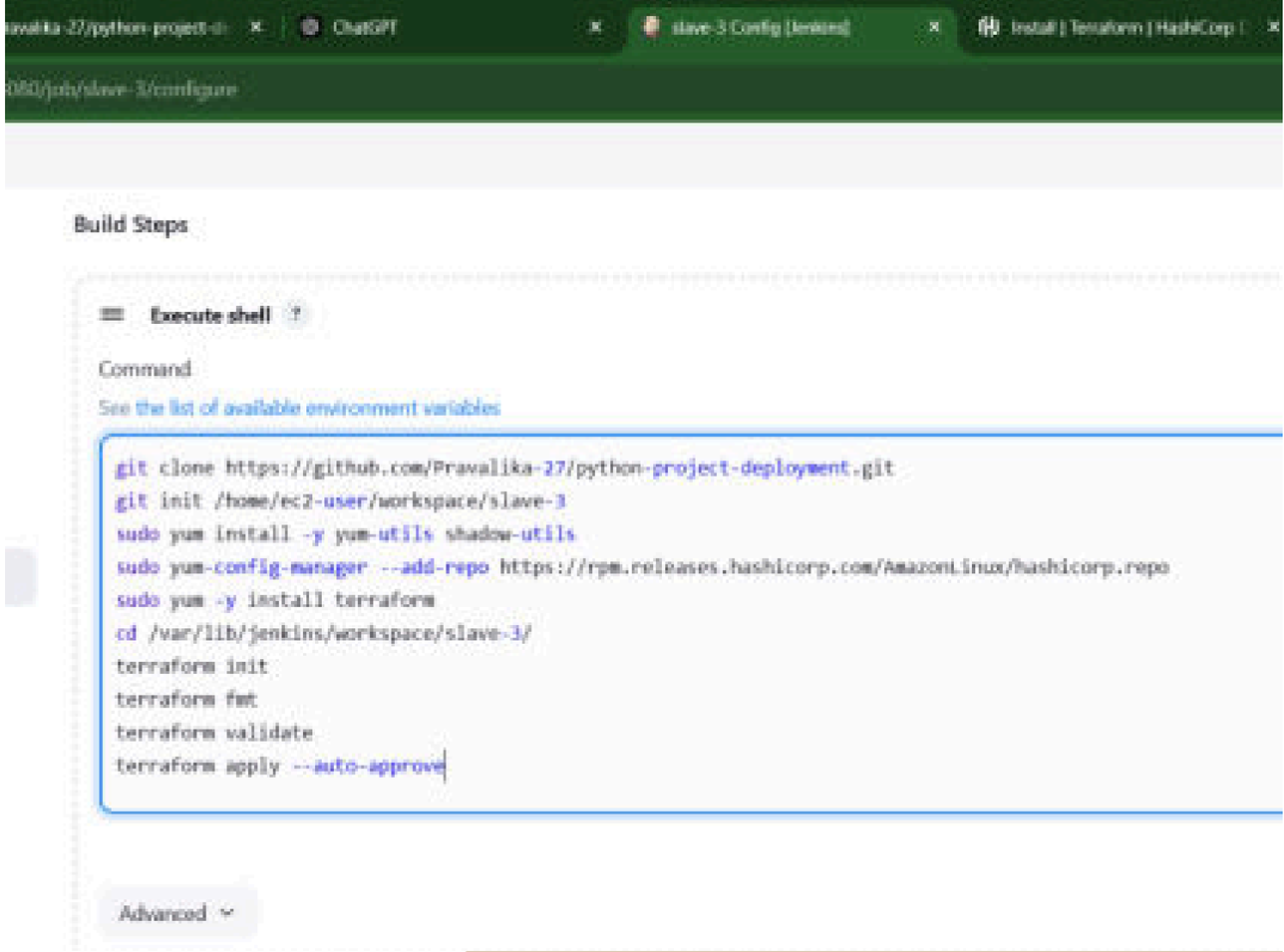
Using Terraform

step 1:launch master and slave instance

step 2:set up jenkins and add node

step 3:install aws credential plugin and add the credentials

step 4:create a job



step 5: add script in execute shell ,I have cloned the repository which consist of the python code

step 6:Build the job and access the application by using newly created instance public ip

Java application deployment

step 1:launch master and slave instance

step 2:Install maven and jenkins in master,tomcat in slave server

step 3: install deploy to the container plugin

step 4: create a job and add the git url which consist of java application

step 5:near build steps -->Goals:add clean package

step 6: post build actions:select deploy war/ear to the container-->give file path and context path-->select containers-->tomcat 9x remote-->add tomcat url and credentials

step 7:Build the job then we can see the application in the tomcat page their we can access the application

Using Bash Script

step 1:launch master and slave instance

Step 2: Install Jenkins and tomcat in slave server

Step 3: Configure Jenkins and Add Slave Node

Step 4: Configure the execute shell

step 5: post build actions:select deploy war/ear to the container-->give file path and context path-->select containers-->tomcat 9x remote-->add tomcat url and credentials

step 6:build the job the application can be seen in the tomcat page


Thank
You