Step1: On the MAster node:

Install java:

=============================

# sudo su -

# **sudo apt install default-jre -y**

**# java -version**

Install jenkins on ACM:

============================

**curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null**

**echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null**

**sudo apt-get update**

**sudo apt-get install jenkins -y**

**sudo systemctl start jenkins**

==============================

Go to jenkins dashboard and complete the initial set up

Configure jenkins and loginto jenkins.

====================================

Step 2: Install Ansible plugins in Jenkins CI server

Manage Jenkins--> plugins --> available plugins --> Search for plugin-> Ansible>> click on install button⇒ plugin will installed successfully.

Configure ANsible tool in jenkins

Manage Jenkins--> Configure Tools --> Scroll down to find Ansible --> Add ansible--> give name as myansible-->give path as /usr/bin

Save it.

===================================

Playbook1:

- hosts: webserver

become: true

tasks:

- name: update apt-get repo

command: sudo apt-get update

- name: Install git, maven

package: name={{item}} state=present

loop:

- git

- maven

- name: Clone the code on the worker nodes from github

git: repo=https://github.com/Sonal0409/DevOpsCodeDemo.git dest=/tmp/mycode

- name: Execute maven command to build the code

command: chdir=/tmp/mycode mvn package

Execute the playbook:

ansible-playbook -i /home/ansiuser/myinventory CEPPlaybookMavenBuild.yml

Check if the build is available on worker nodes

ansible -i /home/ansiuser/myinventory webserver -m command -a "ls /tmp/mycode/target"



Playbook2 – Deployment

- hosts: webserver

become: true

tasks:

- name: Start docker service

service: name=docker state=started

- name: copy the war file in same directory as that of dockerfile

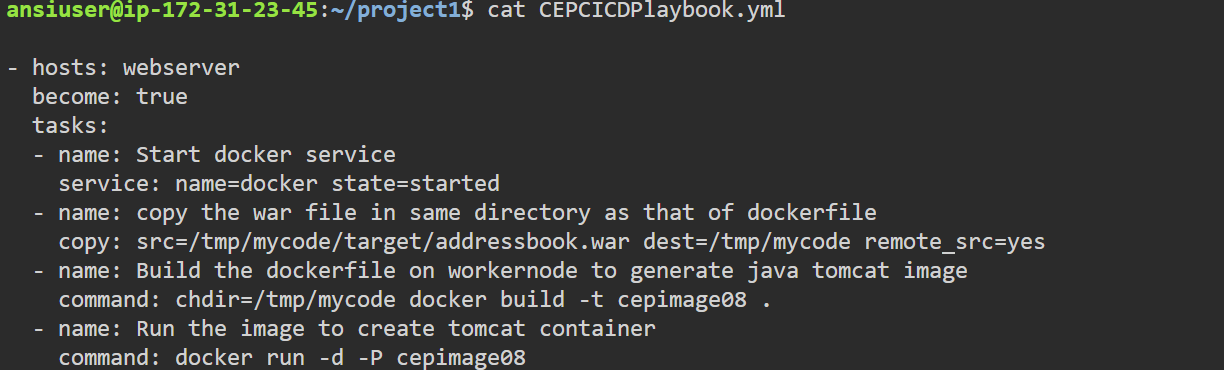
copy: src=/tmp/mycode/target/addressbook.war dest=/tmp/mycode remote\_src=yes

- name: Build the dockerfile on workernode to generate java tomcat image

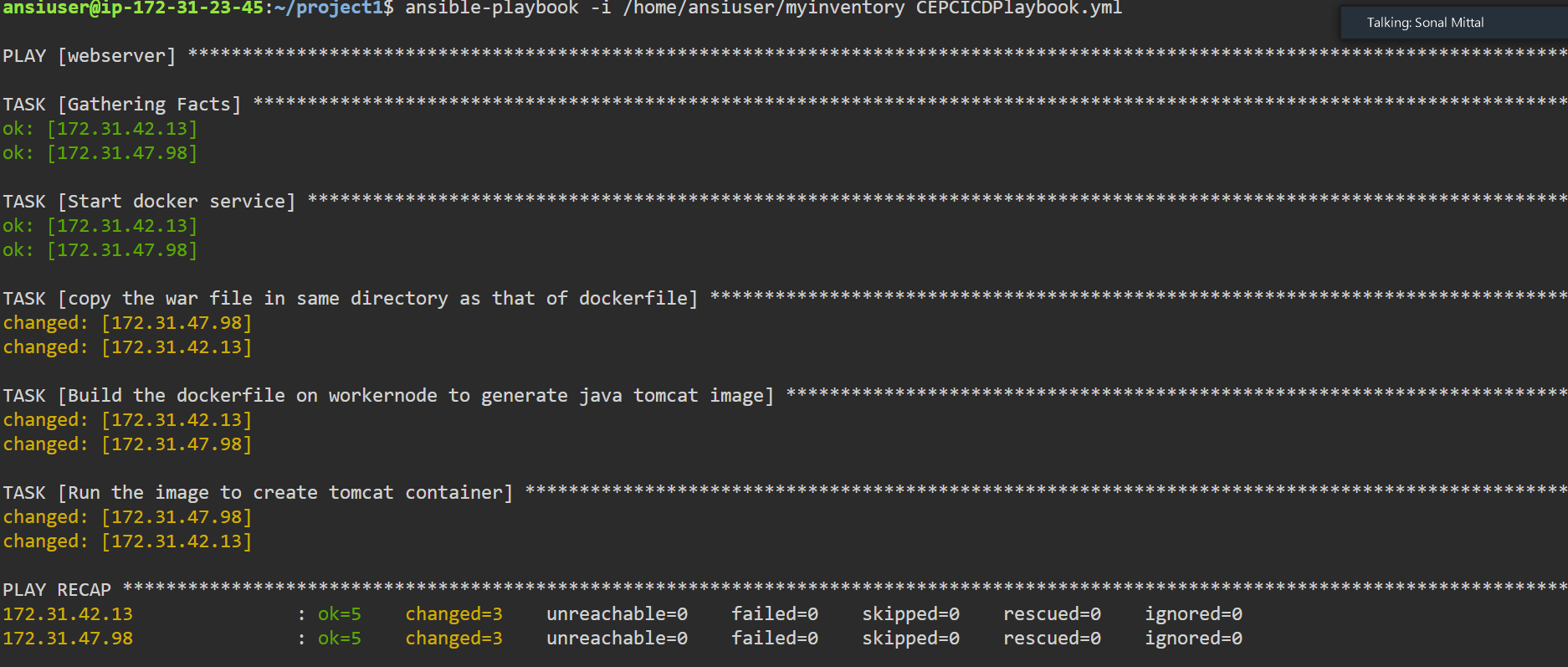
command: chdir=/tmp/mycode docker build -t cepimage08 .

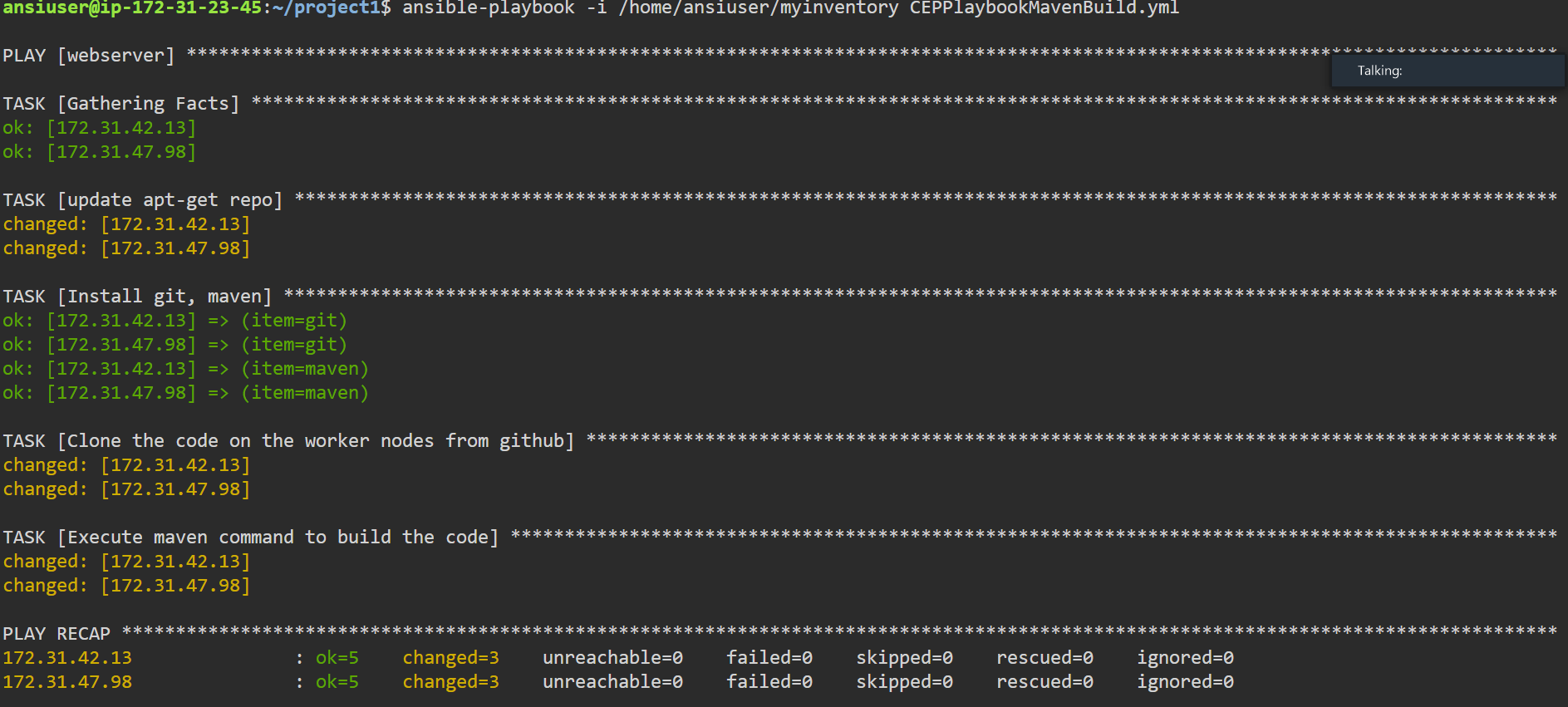
- name: Run the image to create tomcat container

command: docker run -d -P cepimage08



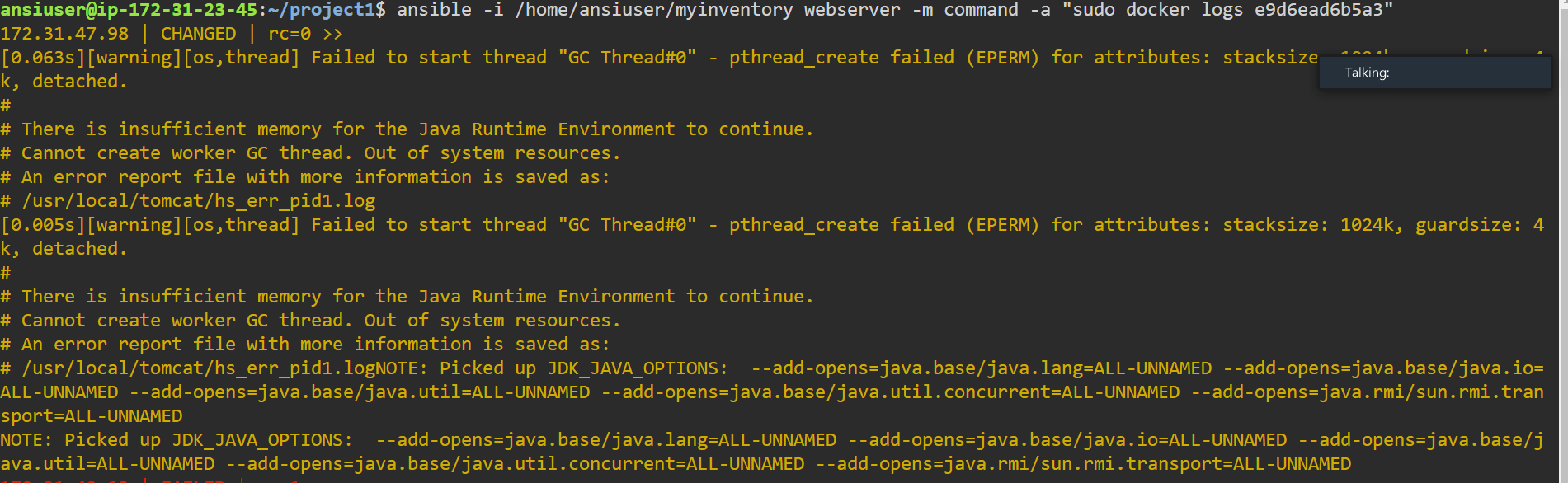
Executed plabook2





Since the container got created on the Simplilearn CM lab worker nodes

But the container was in Exited state. We tried several solutions but container was not coming up due to the below error



SO we decided to create a new worker node of OS ubuntu on AWS lab

In this worker node:

* Create ansiuser
* Copy SSH key of controller on this worker node
* Then write a playbook on ACM

1. Install git and docker on new worker node
2. Build the dockerfile into an Image
3. Run the Image

This time the container will be up and running.

Playbook for deployment on aws server:

- hosts: awsserver

become: true

tasks:

- name: Install git and docker

yum: name={{item}} state=present

loop:

- git

- docker

- name: Clone the repo which has build war file and dockerfile

git: repo=https://github.com/Sonal0409/AnsibleDockerdemo.git dest=/tmp/mycode

- name: Start the docker service

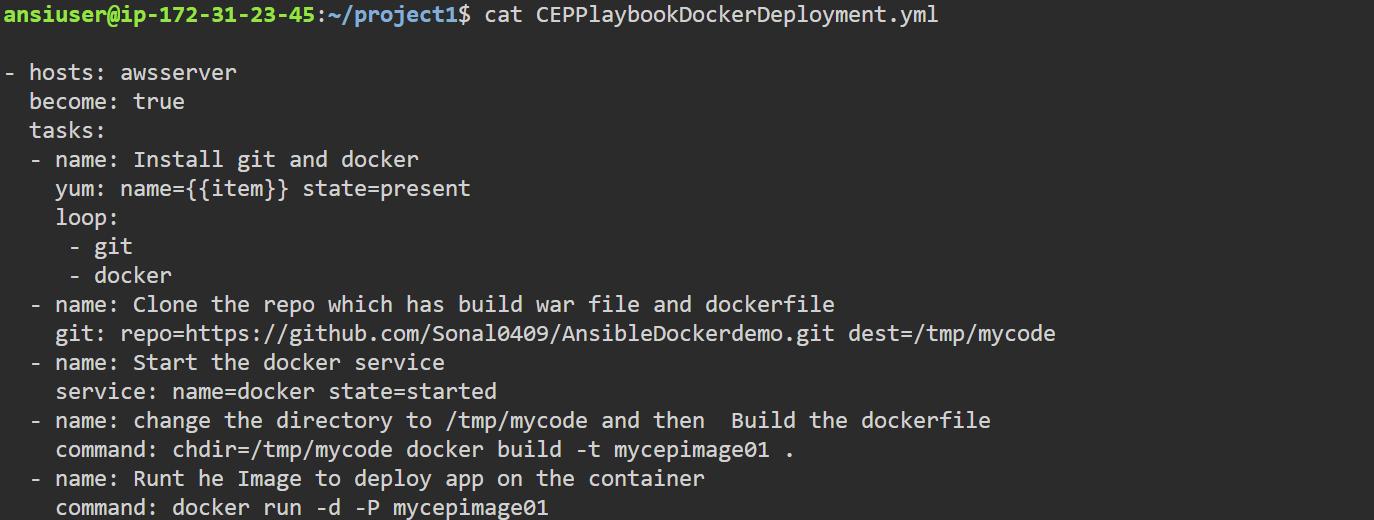
service: name=docker state=started

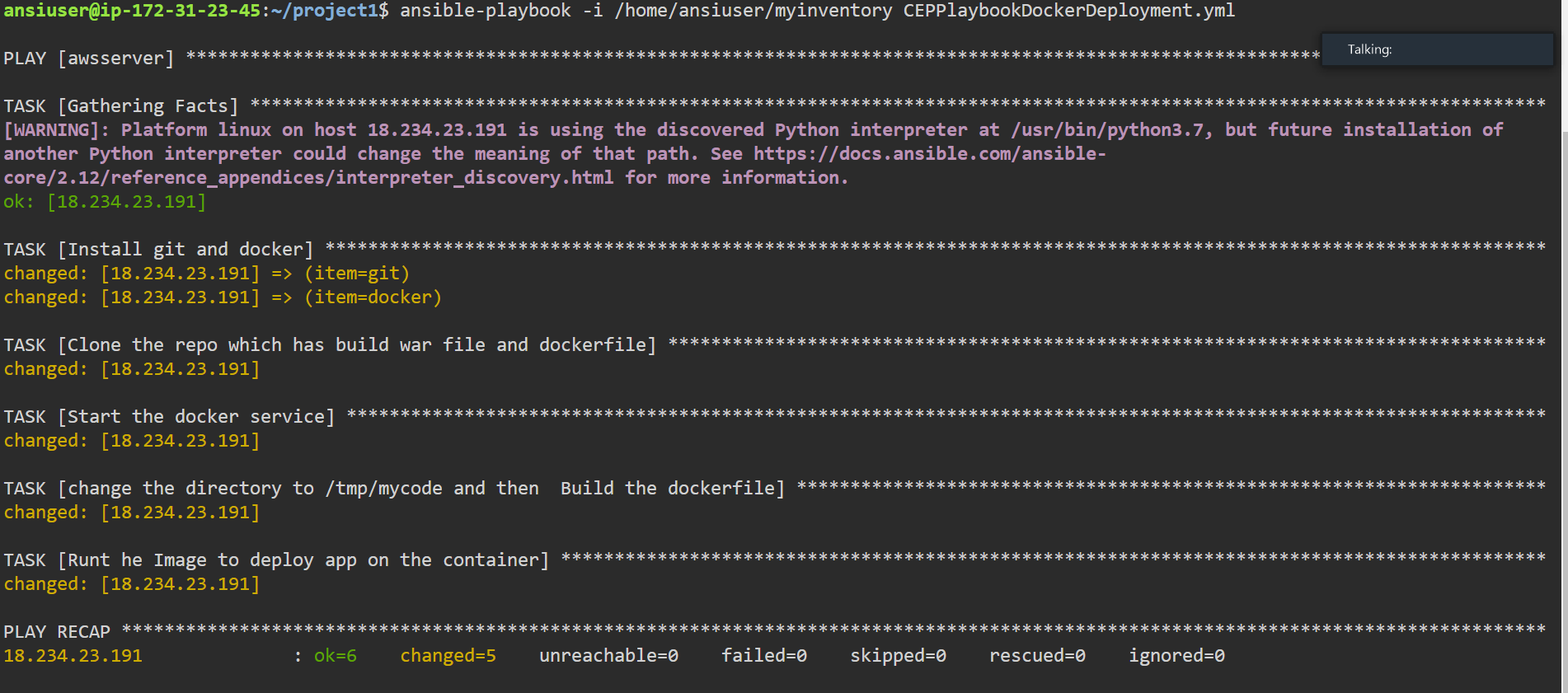
- name: change the directory to /tmp/mycode and then Build the dockerfile

command: chdir=/tmp/mycode docker build -t mycepimage01 .

- name: Runt he Image to deploy app on the container

command: docker run -d -P mycepimage01

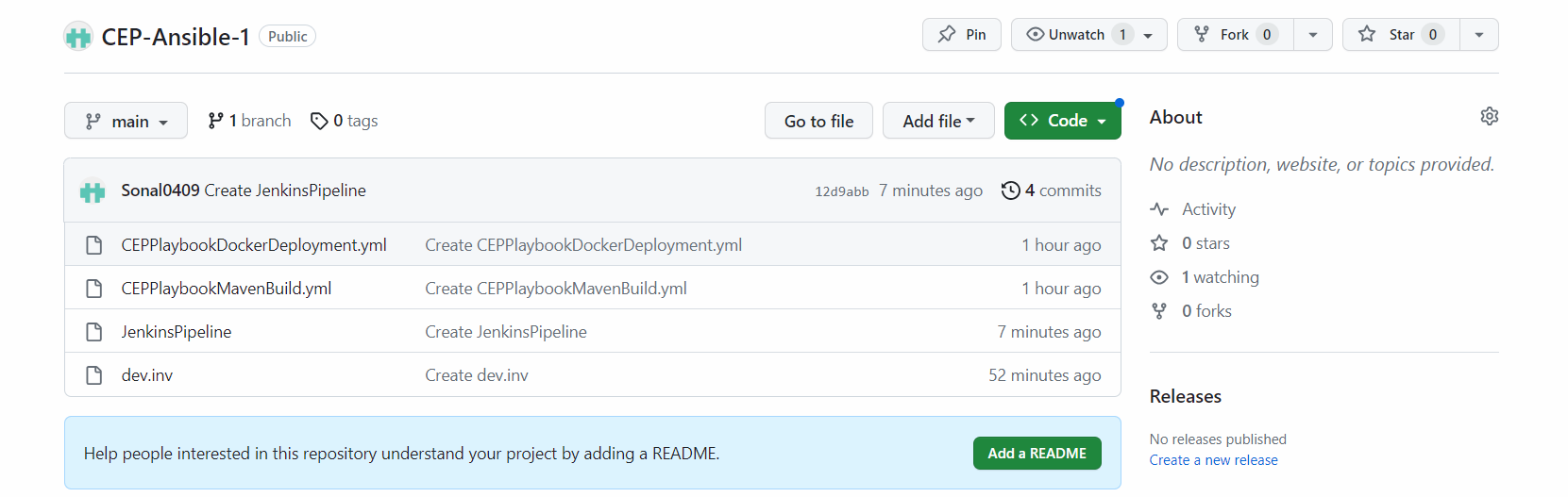


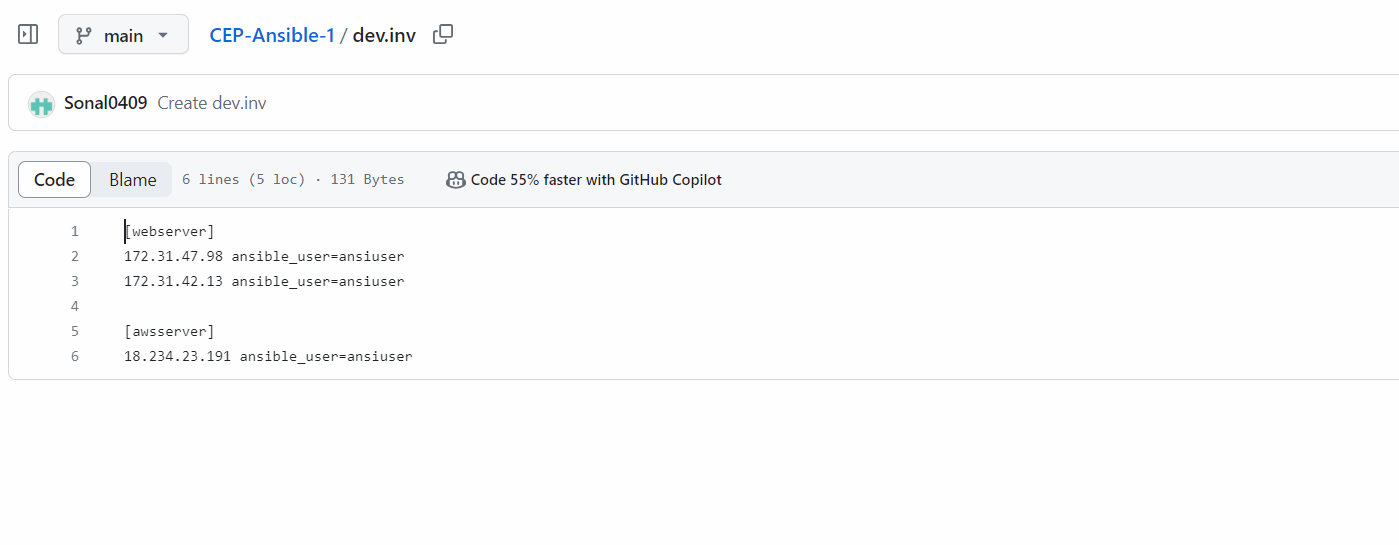


Create a github repo and place your playbooks and inventory

You can fork my repo and change the inventory file with your server ip address

https://github.com/Sonal0409/CEP-Ansible-1.git





Create a new job in Jenkins and write the pipeline code

pipeline{

agent any

stages{

stage('Clone the playbook repo')

{

steps{

git branch: 'main', url: 'https://github.com/Sonal0409/CEP-Ansible-1.git'

}

}

stage('Playbook to Build code')

{

steps{

ansiblePlaybook credentialsId: 'ansiblecredentials', disableHostKeyChecking: true, installation: 'myansible', inventory: 'dev.inv', playbook: 'CEPPlaybookMavenBuild.yml'

}

}

stage('Playbook to deploy code')

{

steps{

ansiblePlaybook credentialsId: 'ansiblecredentials', disableHostKeyChecking: true, installation: 'myansible', inventory: 'dev.inv', playbook: 'CEPPlaybookDockerDeployment.yml'

}

}

}

}

