You are a DevOps engineer at XYZ Ltd. Your company is working on a Java application and wants to automate WAR file artifact deployment so that they don’t have to perform WAR deployment on Tomcat/Jetty web containers. Automate Ansible integration with Jenkins CI server so that we can run and execute playbooks to deploy custom WAR files to a web container and then perform restart for the web container.

**Steps to Perform:**

1. Configure Jenkins server as Ansible provisioning machine
2. Install Ansible plugins in Jenkins CI server
3. Prepare Ansible playbook to run Maven build on Jenkins CI server
4. Prepare Ansible playbook to execute deployment steps on the remote web container with restart of the web container post deployment

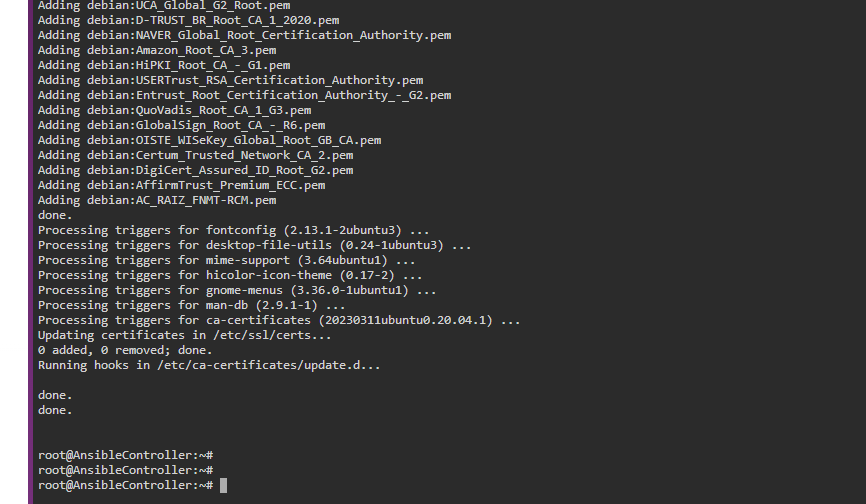
Step 1 : Configure Jenkins server as Ansible provisioning machine

Install java:

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

# sudo su -

# sudo apt install default-jre -y



# java -version

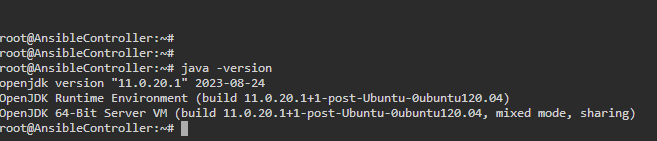
root@AnsibleController:~# java -version

openjdk version "11.0.20.1" 2023-08-24

OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04)

OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)

root@AnsibleController:~#



Install jenkins on Ansible control machine :

https://pkg.jenkins.io/debian-stable/

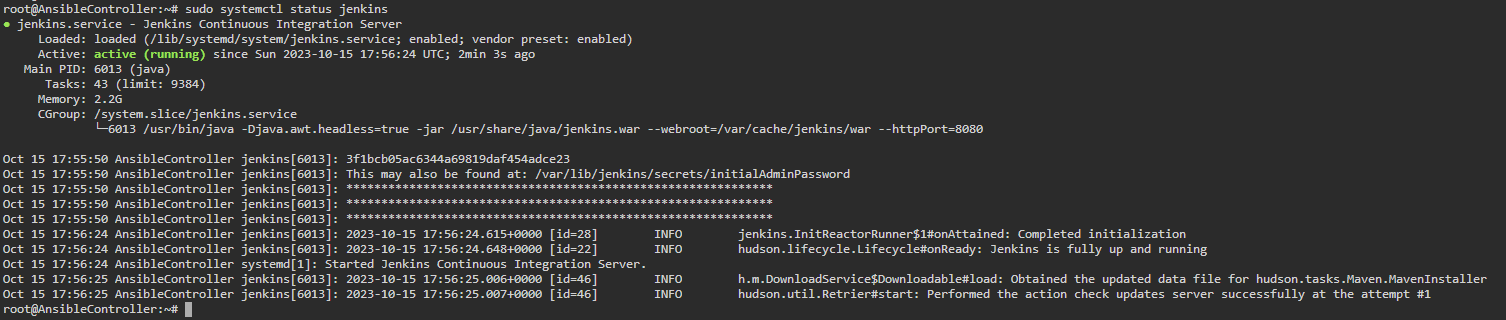
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

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



root@AnsibleController:~# sudo systemctl status jenkins

● jenkins.service - Jenkins Continuous Integration Server

Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)

Active: active (running) since Sun 2023-10-15 17:56:24 UTC; 1min 7s ago

Main PID: 6013 (java)

Tasks: 43 (limit: 9384)

Memory: 2.2G

CGroup: /system.slice/jenkins.service

└─6013 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --h>

Oct 15 17:55:50 AnsibleController jenkins[6013]: 3f1bcb05ac6344a69819daf454adce23

Oct 15 17:55:50 AnsibleController jenkins[6013]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword

Oct 15 17:55:50 AnsibleController jenkins[6013]: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Oct 15 17:55:50 AnsibleController jenkins[6013]: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Oct 15 17:55:50 AnsibleController jenkins[6013]: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Oct 15 17:56:24 AnsibleController jenkins[6013]: 2023-10-15 17:56:24.615+0000 [id=28] INFO jenkins.InitReactorRunn>

Oct 15 17:56:24 AnsibleController jenkins[6013]: 2023-10-15 17:56:24.648+0000 [id=22] INFO hudson.lifecycle.Lifecy>

Oct 15 17:56:24 AnsibleController systemd[1]: Started Jenkins Continuous Integration Server.

Oct 15 17:56:25 AnsibleController jenkins[6013]: 2023-10-15 17:56:25.006+0000 [id=46] INFO h.m.DownloadService$Dow>

Oct 15 17:56:25 AnsibleController jenkins[6013]: 2023-10-15 17:56:25.007+0000 [id=46] INFO hudson.util.Retrier#sta>

Install Ansible plugins in Jenkins CI server

Go to jenkins dashboard and complete the initial set up

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

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.

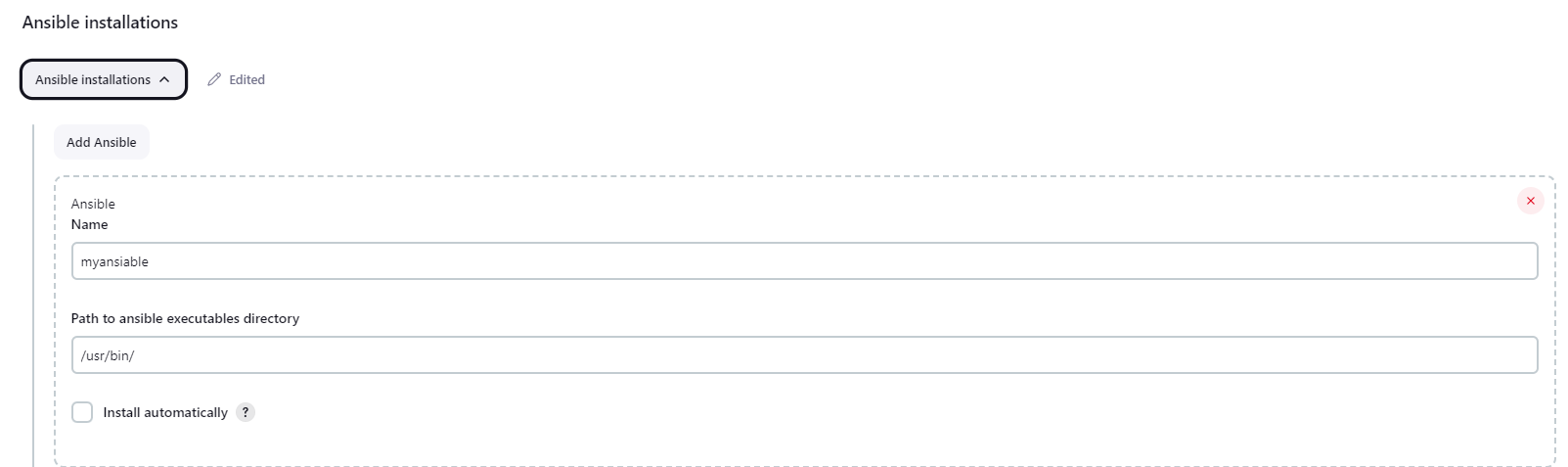
Configure ANsible tool in jenkins

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

Save it.

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

Configure jenkins and loginto jenkins.



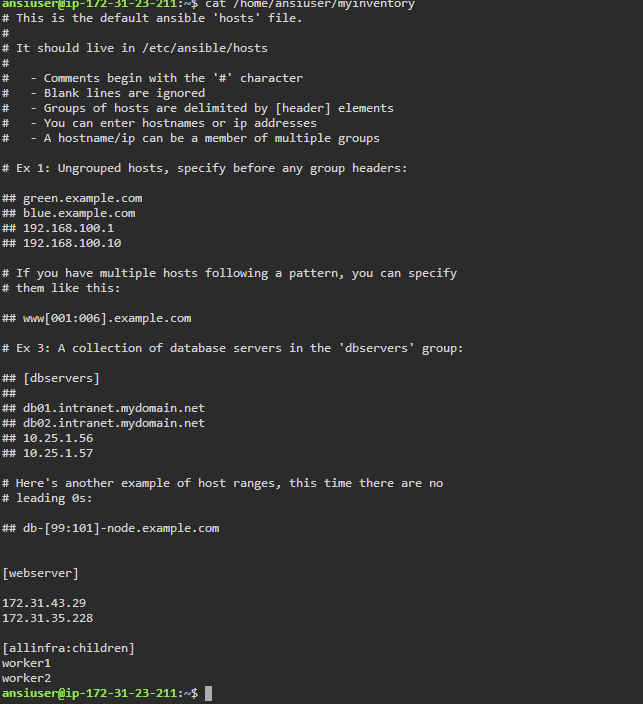
Setting up the inventory

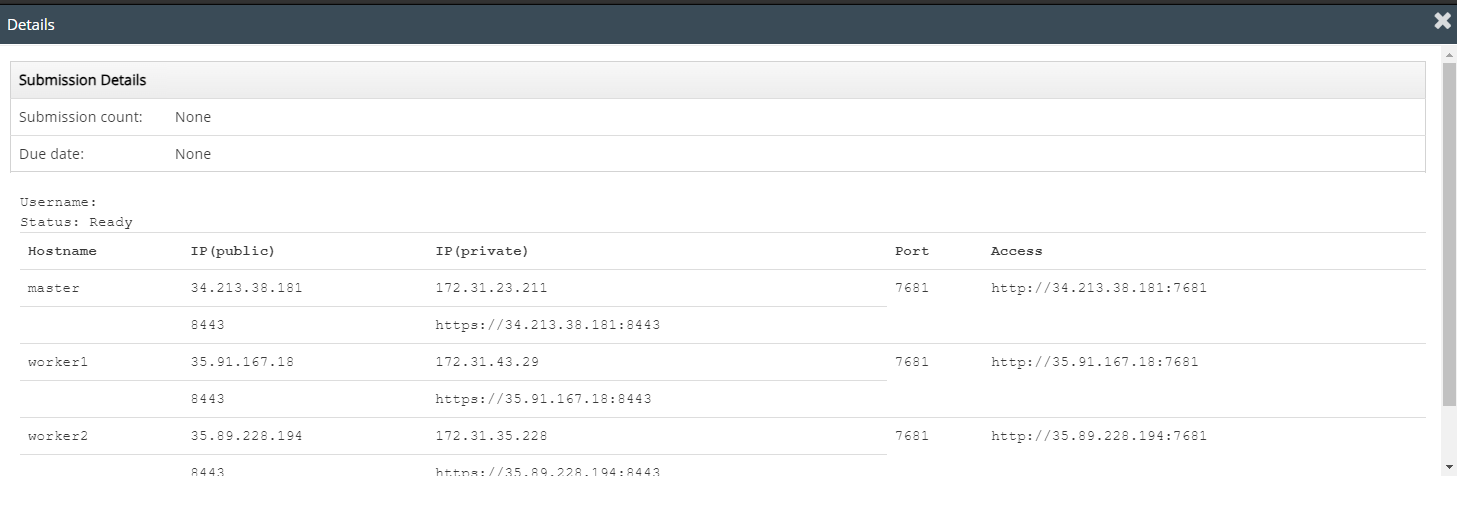
ansiuser@ip-172-31-23-211:~$ cat ansible.cfg

[defaults]

inventory = /home/ansiuser/myinventory

ansiuser@ip-172-31-23-211:~$





**Prepare Ansible playbook to run Maven build on Jenkins CI server**

**=======================================**

ansiuser@ip-172-31-23-211:~$ cat /home/ansiuser/MavenBuildplaybook.yml

- 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

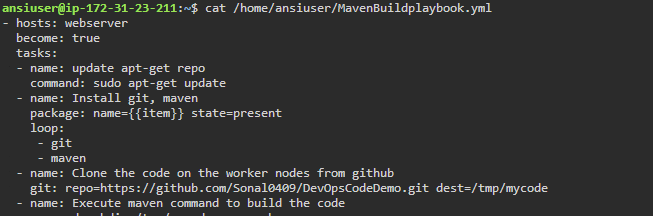
- name: Clone the code on the worker nodes from github

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

ansiuser@ip-172-31-23-211:~$



After Executing the playbook:

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

ansiuser@ip-172-31-23-211:~$ ansible-playbook -i /home/ansiuser/myinventory MavenBuildplaybook.yml

PLAY [webserver] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.35.228]

ok: [172.31.43.29]

TASK [update apt-get repo] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.35.228]

changed: [172.31.43.29]

TASK [Install git, maven] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.35.228] => (item=git)

ok: [172.31.43.29] => (item=git)

changed: [172.31.35.228] => (item=maven)

changed: [172.31.43.29] => (item=maven)

TASK [Clone the code on the worker nodes from github] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.43.29]

changed: [172.31.35.228]

TASK [Execute maven command to build the code] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.35.228]

changed: [172.31.43.29]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.35.228 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

172.31.43.29 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

ansiuser@ip-172-31-23-211:~$



Checking if the build is available on worker nodes

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

ansiuser@ip-172-31-23-211:~$ ansible -i /home/ansiuser/myinventory webserver -m command -a "ls /tmp/mycode/target"

172.31.35.228 | CHANGED | rc=0 >>

addressbook

addressbook.war

classes

generated-sources

generated-test-sources

maven-archiver

maven-status

surefire-reports

test-classes

172.31.43.29 | CHANGED | rc=0 >>

addressbook

addressbook.war

classes

generated-sources

generated-test-sources

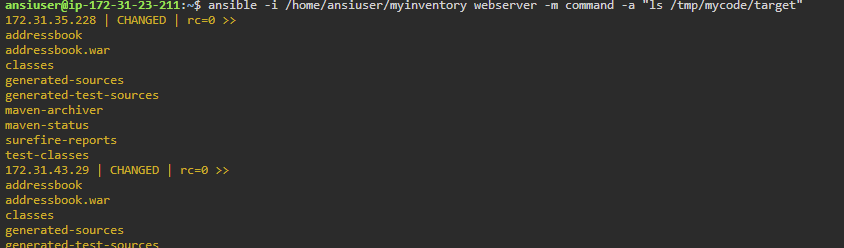
maven-archiver

maven-status

surefire-reports

test-classes

ansiuser@ip-172-31-23-211:~$



**Playbook2 – Deployment**

- hosts: webserver

become: true

tasks:

- name: Start docker service

service: name=docker state=started

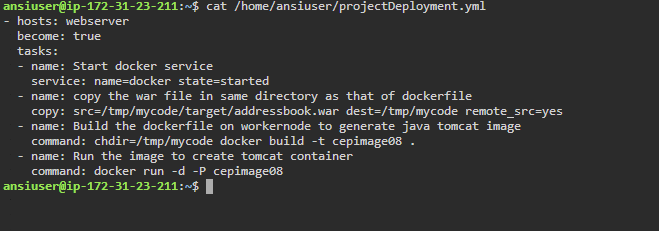
- name: copy the war file in same directory as that of dockerfile

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

ansiuser@ip-172-31-23-211:~$ ansible-playbook -i /home/ansiuser/myinventory projectDeployment.yml

PLAY [webserver] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.35.228]

ok: [172.31.43.29]

TASK [Start docker service] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.35.228]

ok: [172.31.43.29]

TASK [copy the war file in same directory as that of dockerfile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.35.228]

changed: [172.31.43.29]

TASK [Build the dockerfile on workernode to generate java tomcat image] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.35.228]

changed: [172.31.43.29]

TASK [Run the image to create tomcat container] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.35.228]

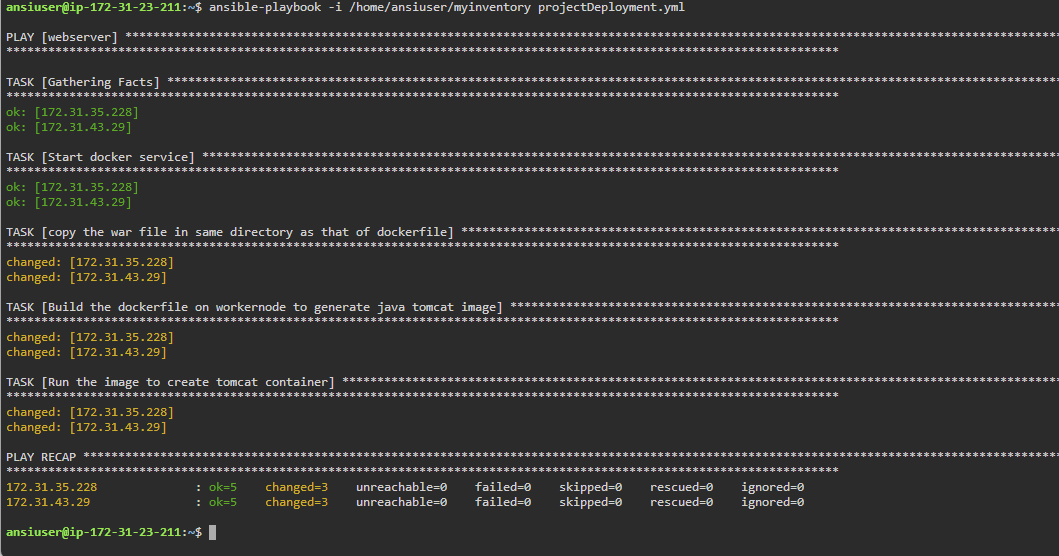
changed: [172.31.43.29]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.35.228 : ok=5 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

172.31.43.29 : ok=5 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

ansiuser@ip-172-31-23-211:~$



Checking the state of container

ansiuser@ip-172-31-23-211:~$ ansible -i /home/ansiuser/myinventory webserver -m command -a "sudo docker ps -a"

172.31.35.228 | CHANGED | rc=0 >>

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

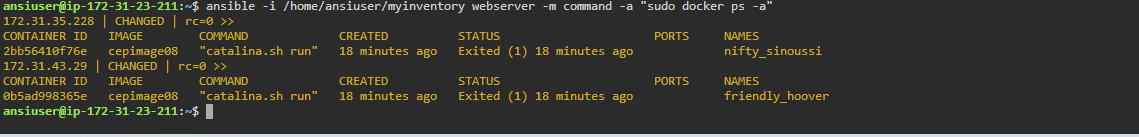
2bb56410f76e cepimage08 "catalina.sh run" 18 minutes ago Exited (1) 18 minutes ago nifty\_sinoussi

172.31.43.29 | CHANGED | rc=0 >>

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0b5ad998365e cepimage08 "catalina.sh run" 18 minutes ago Exited (1) 18 minutes ago friendly\_hoover

ansiuser@ip-172-31-23-211:~$



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

ansiuser@ip-172-31-23-211:~$ ansible -i /home/ansiuser/myinventory webserver -m command -a "sudo docker logs 2bb56410f76e"

172.31.35.228 | CHANGED | rc=0 >>

[0.004s][warning][os,thread] Failed to start thread "GC Thread#0" - pthread\_create failed (EPERM) for attributes: stacksize: 1024k, guardsize: 4k, detached.

[0.008s][error ][gc,task ] Failed to create worker threadNOTE: Picked up JDK\_JAVA\_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED

172.31.43.29 | FAILED | rc=1 >>

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.

Adding SSH key for ansiuser

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

ansiuser@ip-172-31-23-211:~$ ssh-copy-id -i ansiuser@54.226.31.3

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansiuser/.ssh/id\_rsa.pub"

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

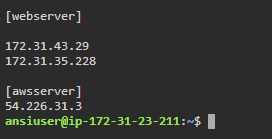
ansiuser@54.226.31.3's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansiuser@54.226.31.3'"

and check to make sure that only the key(s) you wanted were added.

Then write a playbook on ACM



Playbook for deployment on aws server:

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

ansiuser@ip-172-31-23-211:~$ cat /home/ansiuser/awsworkerdockerdeployment.yml

- 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

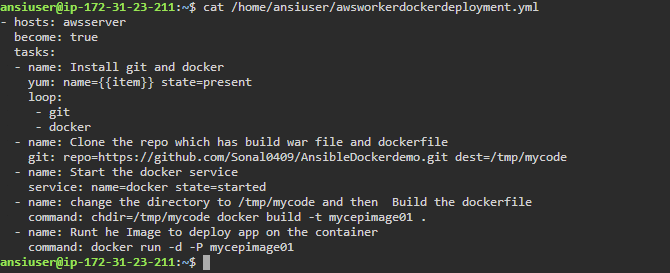
- name: change the directory to /tmp/mycode and then Build the dockerfile

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

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

command: docker run -d -P mycepimage01

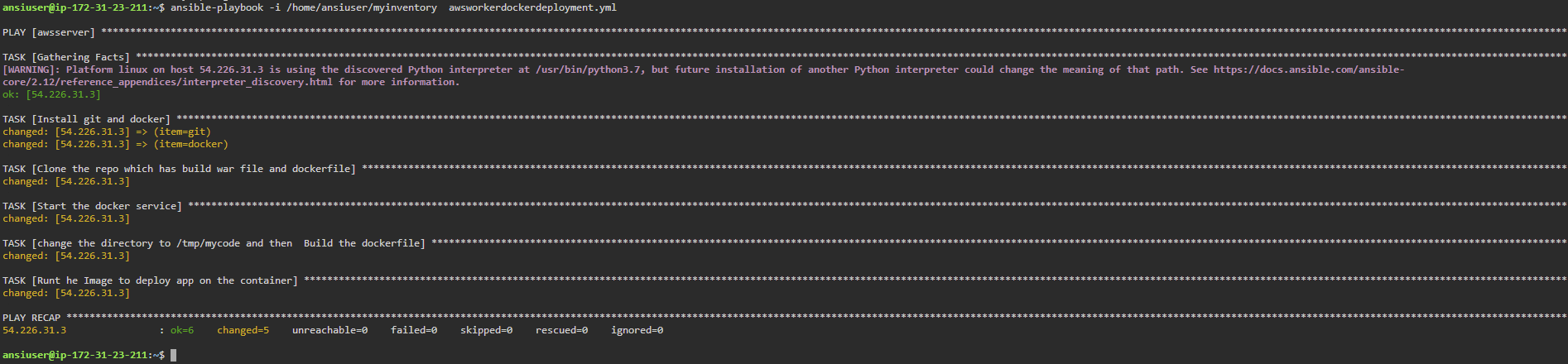
ansiuser@ip-172-31-23-211:~$



Executing the play book :

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

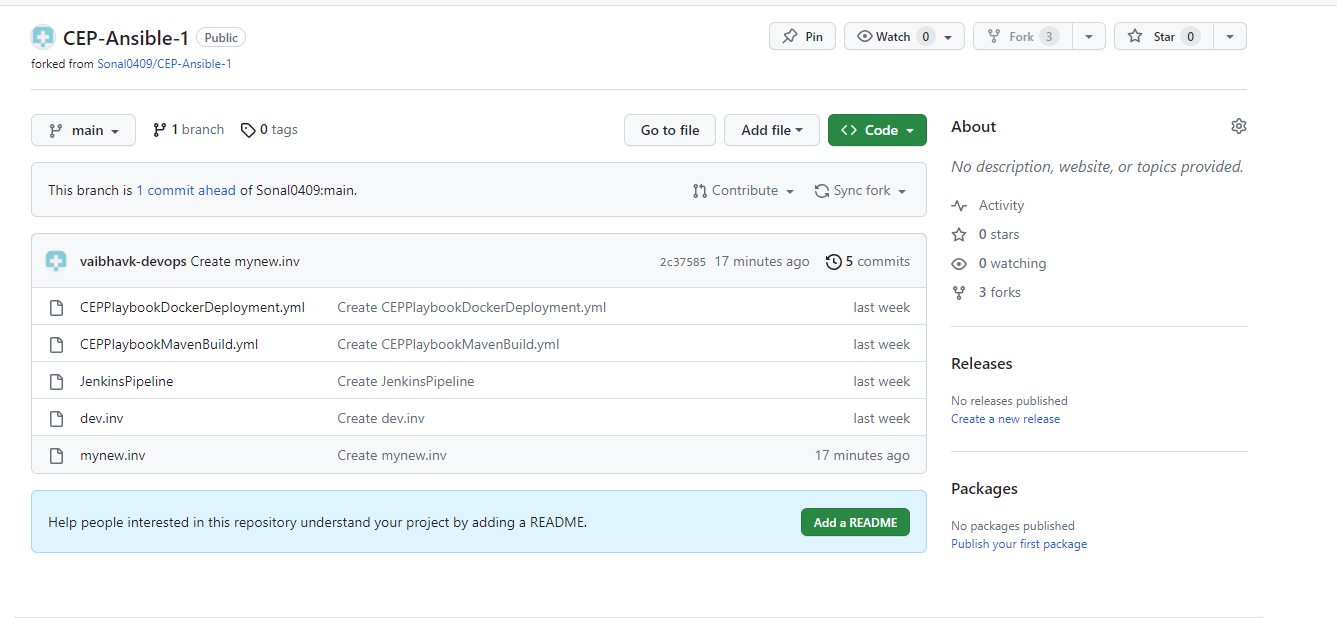
ansible-playbook -i /home/ansiuser/myinventory awsworkerdockerdeployment.yml



Create a github repo and place your playbooks and inventory

Forking <https://github.com/Sonal0409/CEP-Ansible-1.git> to my github

<https://github.com/vaibhavk-devops/CEP-Ansible-1>



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/vaibhavk-devops/CEP-Ansible-1'

}

}

stage('Playbook to Build code') {

steps{

ansiblePlaybook credentialsId: 'awsansiuser', disableHostKeyChecking: true, installation: 'myansible', inventory: 'mynew.inv', playbook: 'CEPPlaybookMavenBuild.yml'

}

}

stage('Playbook to deploy code')

{

steps{

ansiblePlaybook credentialsId: 'awsansiuser', disableHostKeyChecking: true, installation: 'myansible', inventory: 'mynew.inv', playbook: 'CEPPlaybookDockerDeployment.yml'

}

}

}

}

