

JenKins PipeLine Demo Documentation

* Machines Requirement:
* Three AWS Ec2 Instance T2.micro
* Any SSH Client Software
* Software or Cloud Based Sites
* Git-GitHub
* For Image Registry Server: DockerHub

Steps:

1] Launch the linux instance from Ec2, access them through ssh client

2] In this demo I used root user to do all task.

3] Do the key base authentication on all machines for file transfer.

4] By Default on cloud instance PasswordAuthentication is No. , First edit the ssh file,

#vim /etc/ssh/sshd\_config

Search For: PasswordAuthentication (yes) change to yes

:wq

#set root password

5] Now from first machine which is Jenkins-Instance

Create ssh key

#ssh-keygen

#ssh-copy-id root@ip\_Ansible\_Second\_Instance

Enter password

6] Same Step As (5) On Ansible Machine Ans DockerHost Machine Which is 3rd

7] After key Base Authentication Done, Go to /etc/ssh/sshd\_config file and No the

PasswordAuthentication

For test direct ssh root@ip

Machine Installation For Jenkins, Ansible, Docker

1] Jenkins: <https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos>

Go to above link and follow the steps as given.

2] After successful Installation Open Chrome paste the IP of jenkins machine and start

using Jenkins

3] Ansible : # yum install<https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

# yum -y install ansible

Check Ansible Version # ansible --version

4] Go to /etc/ansible/hosts

Add

[Docker]

Docker\_machine\_IP

:wq

# ansible -m ping all

Result should be in green text.

#docker login

5] Docker:

Open - <https://linuxconfig.org/how-to-install-docker-in-rhel-8>

#dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

#dnf repolist -v

#dnf install docker-ce

#docker –version

Jenkins Configuration

1] Login to Jenkins

Go to Manage Jenkins

* Configure plugins, Install Publish Over ssh (Install without restart)
* Configure System- Go to ssh server ADD
* Enter name Of server (jenkins), root PASSWD-Hostname, and in advance option add password of that machine

2] Same steps for Ansible In publish over ssh (ADD SSH SERVER)

3] After adding do test configuration also.

4] Login to GitHub account- create new repo- go to settings- webhooks- add the jenkins machine url/github/webhook/

SAVE and wait for green tick

5] Go to Dashboard- add new item- name it- select freestyle project- search for SCM

GIT-

Enter the github repo URL- check the branch name-

Save And Apply

6] Again Configure-

In Build Step- Select Jenkins machine- And in execute tab-

**rsync -avh /var/lib/jenkins/workspace/demo-06/Dockerfile**

**root@172.31.20.87:/project6**

**root@172.31.20.87 (ansible ip)**

Add above command as per changes.

Save And Apply

7] Configure-build step

On ansible-machine

Exec command-

For Docker Build,tag,push image

**cd /project6**

**docker build -t $JOB\_NAME:v$BUILD\_ID .**

**docker tag $JOB\_NAME:v$BUILD\_ID dockerhub-name/$JOB\_NAME:v$BUILD\_ID**

**docker tag $JOB\_NAME:v$BUILD\_ID dockerhub-name/$JOB\_NAME:latest**

**docker push dockerhub-name/$JOB\_NAME:v$BUILD\_ID**

**docker push dockerhub-name/$JOB\_NAME:latest**

**docker rmi -f $JOB\_NAME:v$BUILD\_ID dockerhub-name/$JOB\_NAME:v$BUILD\_ID dockerhub-name/$JOB\_NAME:latest**

Save And Apply

**8**] **Creation of Ansible Playbook on Ansible Machine**

#vim pro.yml

**---**

**- name: my playbook for jenkins demo to create container**

**hosts: all**

**tasks:**

**- name: stop container**

**shell: docker stop con**

**- name: remove container**

**shell: docker rm con**

**- name: delete image**

**shell: docker rmi -f dockerHubName/yash-project**

**- name: create container**

**shell: docker run -itd -p 80:80 --name=con dockerHubName/demo-06**

9] Configure- Add post build action

**Select Send Build Artifacts over SSH**

Select ssh server Ansible

Exec Command

**cd /project6**

**ansible-playbook pro-demo.yml**

Save And Apply

10]GitHub DockerFile :

Search on google free template copy particular path and paste into the DockerFile

**FROM centos**

**RUN sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-\***

**RUN sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-\***

**RUN yum install httpd zip unzip vim -y**

**ADD https://www.free-css.com/assets/files/free-css-templates/download/page284/mical.zip /var/www/html**

**WORKDIR /var/www/html**

**RUN unzip mical.zip**

**RUN rm -rf mical.zip &&\**

**cp -rf mical-html/\* . &&\**

**rm -rf mical-html**

**RUN echo "Hello World xyz" > /samplefile.txt**

**EXPOSE 80**

**CMD ["/usr/sbin/httpd" , "-D" , "FOREGROUND"]**

11] After all these steps, check them carefully And build the job.

12] If the build result is success then, Put the docker host machine IP in Browser,

ENTER.