We will launch one ubuntu ec2 instance (at least t2 medium) 🡪 install docker in it 🡪 create 3 containers (Jenkins, qaserver,prodserver)

1. Create docker file with following content

Vim dockerfile

FROM jenkins/jenkins

USER root

RUN apt-get update -y

RUN apt-get install rsync -y

RUN apt-get install ssh -y

RUN apt-get install vim -y

RUN apt-get install git -y

RUN apt-get install maven -y

1. # docker build -t Jenkins\_image .
2. # vim dockerfile2

FROM tomee

USER root

RUN apt-get update -y

RUN apt-get install rsync -y

RUN apt-get install ssh -y

RUN apt-get install vim -y

1. # docker build -t tomee\_image -f dockerfile2
2. vim docker-compose.yml

---

version: '3.8'

services:

myjenkins:

image: jenkins\_image

ports:

- 5050:8080

container\_name: myjenkins

qaserver:

image: tomee\_image

ports:

- 6060:8080

links:

- myjenkins:jenkins

container\_name: qaserver

prodserver:

image: tomee\_image

ports:

- 7070:8080

links:

- myjenkins:jenkins

container\_name: prodserver

...

1. docker-compose up -d
2. check all containers are accessible from browser by giving particular port
3. # docker exec -it qaserver bash
4. #passwd root
5. # vim /etc/ssh/sshd\_config

Permitrootlogin : yes

1. # service ssh restart
2. Ctr p ctr q
3. Do same steps (8-11) for jenkins and prod container
4. # docker inspect qaserver

# docker inspect prodserver

# docker inspect Jenkins

Take ip’s and keep in notepad

1. Go to Jenkins container

# docker exec -it myjenkins bash

# ssh-keygen

# ssh-copy-id root@private\_ip\_of\_qa

# ssh-copy-id root@private\_ip\_of\_prod

1. Check if pwless connection established or not
2. Do same for Jenkins server and prod server
3. Go to Jenkins webpage
4. Install plugins and fill necessary details
5. Create a job 🡪development🡪 freestyle
6. Sourcecode: git

[https://github.com/intelliqittrainings/maven.git](https://github.com/intelliqittrainings/maven.git" \t "_blank)

build: invoke top level maven targets

build

1. Manage plugins🡪 install 🡪 publish over ssh
2. Manage Jenkins🡪Configure system 🡪 scroll extreme down 🡪 ssh servers

Here add both Jenkins and qaserver

Ip: private ip of container

Username: root

1. Configure development job 🡪 postbuild actions 🡪 send build artifacts over ssh

Exec command:

rsync -avh /var/jenkins\_home/workspace/Devlopment/webapp/target/webapp.war [root@172.18.0.3:/usr/local/tomee/webapps/](mailto:root@172.18.0.3:/usr/local/tomee/webapps/)

1. Build the job and try to open qaservers webpage

Public\_ip\_of\_ec2:6060/webapp

1. Create new job in Jenkins 🡪 testing
2. Git 🡪 [https://github.com/intelliqittrainings/FunctionalTesting.git](https://github.com/intelliqittrainings/FunctionalTesting.git" \t "_blank)

Build the job and copy the workspace path

1. Configure 🡪 build🡪 execute shell

java -jar /var/jenkins\_home/workspace/Testing/testing.jar

1. Linked development and testing jobs in Jenkins
2. Run development job 🡪 testing job will also get execute
3. Install plugin 🡪 copy artifact
4. Development 🡪 postbuild actions 🡪 archive the artifacts

Files to archive: \*\*/\*.war

1. Testing 🡪 build 🡪 copy artifacts from another project 🡪 select ‘development’
2. Apply 🡪 save
3. Go to Testing job -->Post build actions --> send artifacts over ssh --> select jenkins -->
4. Add below command
5. rsync -avh /var/jenkins\_home/workspace/Testing/webapp/target/webapp.war [root@Prod\_server\_ip:usr/local/tomee/webapps/](mailto:root@172.18.0.4:usr/local/tomee/webapps/)
6. Build the Devlopment job
7. App will be deployed to prod server
8. Dockerhost\_public\_ip:7070/webapp