**CI/CD on Containers (DOCKER)**

1. First login to the AWS account and restart the **Jenkins server** and **Tomcat server** (Web-Server and Jenkins-Server).
2. Using the IPV4 address of the instances connect them using putty.
3. Using the localhost addresses 52.66.18.47(Jenkins-Server) and 52.66.206.40 (Web - Server) and **start the Jenkins and tomcat server** inside the server like **tomcatup and systemctl** start Jenkins. Then, check if the server is running or not by using ps -ef | grep tomcat and systemctl status Jenkins (**Localhost address may vary**).
4. Now creating an **EC2** **instance** named **Docker-Host** with **Amazon linux.**
5. Install docker on EC2 instance and start services using

**yum install docker**

and to start the docker use the command

**service docker start**

1. Create a new user for Docker management and add him to Docker (default) group

**useradd dockeradmin**

**passwd dockeradmin**

For adding the created user to the default docker group use this command

**usermod -aG docker dockeradmin**

1. Write a Docker file under /opt/docker

**mkdir /opt/docker**

**### vi Dockerfile**

**# Pull base image**

From tomcat:8-jre8

**# Maintainer**

MAINTAINER "ramprakashsaravanan98@gmail.com"

**# copy war file on to container**

COPY ./webapp.war /usr/local/tomcat/webapps

1. Login to Jenkins console and add Docker server to execute commands from Jenkins

**Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials**

1. Create **Jenkins job** named **Docker Job**

A) Source Code Management  
Repository: <https://github.com/ramprakash04/hello-world.git>  
Branches to build: **\*/master**

B) Build Root POM: pom.xml  
Goals and options: **clean install package**

C) Send files or execute commands over SSH Name: docker\_host  
Source files: **webapp/target/\*.war**  
Remove prefix: **webapp/target**  
Remote directory: **//opt//docker**Exec command: **docker stop dock\_demo; docker rm -f** **dock\_demo; docker image rm -f dock\_demo; cd /opt/docker; docker build -t dock\_demo.**

D) send files or execute commands over SSH  
Name: **docker\_host**  
Exec command: **docker run -d --name dock\_demo -p 8090:8080 dock\_demo**

1. Open the docker-host server inside that use

**vi /etc/ssh/sshd\_config**

to enable **PasswordAuthentication** go into it and edit the file with **#**

1. Then start the SSH service using

**service sshd restart**

then check the test configuration again in publish over **ssh** and check the status if it is showing as **SUCCESS.**

1. Now **apply** and **save** the Docker Job.
2. Execute the jenkins jobs using the **Build now** option and then check the console output whether it is success or not**.**
3. If it the console output is **FINISHED: UNSTABLE,** then give the ownership to dockeradmin user by using the command

**chown -R dockeradmin:dockeradmin /opt/docker**

1. Now again execute the jenkins job using Build now option and check the console output as **FINISHED: SUCCESS**
2. Use commands

**docker images (**To check whether the images are available)

**docker ps (**Tocheck the running containers in the server**)**

**docker ps -a (**Tocheck both running and stopped containers**)**

Now got to the browser and check whether the tomcat server is running or not.

1. Finally give **43.204.24.61:8090/webapp** to check the desired output.