**SIMPLE DEVOPS PROJECT 3**

**CI/CD THROUGH CONTAINER (DOCKER)**

1. Create a new instance in aws as **Docker Host**
2. Copy the IPV4 address and go and open the putty and configure the ip address with the respective **.ppk** file.
3. Then login it with the **username: ec2-user.**
4. Then move to the root directory by giving the command **sudo su**
5. Then install the docker on ec2 instance using the command **yum install docker**
6. Start the docker services using the command **service docker start**
7. Then check the status of the docker using the **service docker status**
8. Create a new user for docker management using the command **useradd dockeradmin.**
9. Now set password for the username using **passwd dockeradmin**
10. Then add the user to the default docker group using the command **usermod -aG docker dockeradmin**
11. Create a file under the opt folder as docker using the command **mkdir docker**
12. Then go to that docker folder using the command **cd docker**
13. Write the Dockerfile inside the docker folder using the command **vi Dockerfile**

**CODE :**

# Pull base image

From tomcat:8-jre8

# Maintainer

MAINTAINER "bharathrajan77@gmail.com"

# copy war file on to container

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

1. Login to Jenkin console and add Docker server to execute the commands from Jenkins
2. After that install the plugin **Publish over SSH**
3. Then go to **Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials**
4. Now create a Jenkin job in the name of **DEVOPS CS 003** and select the maven project
5. Now configure the job,

A) **Source Code Management**  
 **Repository** :  https://github.com/Bharath-77/CASE-STUDY.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 doc\_demo; docker rm -f doc\_demo; docker image rm - f doc\_demo; cd /opt/docker; docker build -t doc\_demo .

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

19. Now to know whether the configuration is success give the **test configuration** option

20. If the test configuration is not success, then edit the /etc/ssh/sshd\_config using the command **vi /etc/ssh/sshd\_config** and edit the password authentication with **#**

21. Now check the test configuration whether it is **success** or not.

22. Now apply and save the **DEVOPS CS 003** Job.

23. Execute the jenkins jobs using the **Build now** option and then check the console output whether it is success or not.

24. 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**

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

26. Now in the docker server check whether the images are available using the command **docker images**

26. Then check the running containers in the server using the command **docker ps**

27. Now check both running and stopped containers using the command **docker ps -a**

28. Now got to the browser and give the address as **65.2.9.133:8090** and check whether the tomcat server is running or not.

29. Finally give **65.2.9.133:8090/webapp** to check the **desired output**