DEVOPS PROJECT 3

CI/CD ON CONTAINERS (DOCKER)

Step 1: First login to the AWS account and restart the jenkins server and Tomcat server (Web Server and Jenkins\_Server).

Step 2: Using the IPV4 address of the instances connect them using putty.

Step 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).

Step 4: Create an instance for Docker using Amazon Linux then name it as Docker-Host.

Step 5: Install docker on Docker-Host instance using yum install docker and start services using service docker start.

Step 6: create a new user for Docker management and add him to Docker (default) group

* + useradd dockeradmin
  + passwd dockeradmin
  + For adding the user that is created to the default docker group use the command usermod -aG docker dockeradmin

Step 7: Write a Docker file under /opt/docker

mkdir /opt/docker

### vi Dockerfile

# Pull base image

From tomcat:8-jre8

# Maintainer

MAINTAINER "athullyachandrathil@gmail.com"

#copy war file on to container

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

Step 8: Login to Jenkins console and add Docker server to execute commands from Jenkins. Before that Install the plugin called publish over SSH. Then, go to Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials.

Step 9: Create Jenkins job, name it as Docker\_project-3 and click on maven project

Step 10:

1. Source Code Management  
   Repository: https://github.com/AthulyaG/hello-world.git   
   Branches to build: \*/master
2. Build Root POM: pom.xml  
   Goals and options: clean install package
3. 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

Step 11: To know the status of the configuration we must test the configuration and check if we get the message as SUCCESS. To do so…

Step 12: Open the docker-host server inside that use

vi /etc/ssh/sshd\_config to enable PasswordAuthentication go into it and edit the file with #

Step 13: Then start the SSH service using service sshd restart

Step 14: then check the test configuration again in publish over ssh and check the status if it is showing as SUCCESS or NOT.

Step 15: Now apply and save the job.

Step 16: Now Build the job using Build Now and check the console output whether it is success or not.

Step 17: if the console output is showing as finished: unstable then, give the ownership to dockeradmin user by using the command chown -R dockeradmin:dockeradmin /opt/docker

Step 18: Now again build the job and check whether the output is SUCCESS

Step 19: Then, check the Docker-Host server for images using the command

* docker images (to show the images)
* docker ps (to show the running containers)
* docker ps -a (to show the running containers as well as stopped containers)

Step 20: Now login to the tomcat using the public IPV4 address 52.66.206.40:8090 then check the output using /webapp as 52.66.206.40:8090/webapp.