**CASE STUDY :- 3**

**Docker Installation on AWS Instance :-**

1.Update the packages on your instance

#sudo yum update -y

2. Install Docker

# sudo yum install docker -y

3.Start the Docker Service

# sudo service docker start

4. Add the ec2-user to the docker group so you can execute Docker commands without using sudo.

# sudo usermod -a -G docker ec2-user

**Creating a Dockerfile under /opt/docker :-**

1.mkdir /opt/docker

2. vi Dockerfile

# Pull base image

From tomcat:8-jre8

# Maintainer

MAINTAINER "Abhinav Tarhekar"

# copy war file on to container

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

3.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

4.Create Jenkins job

A) Source Code Management  
Repository : <https://github.com/adt77/Case_Study_3.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[s] :

docker stop abhinav\_demo;

docker rm -f abhinav\_demo;

docker image rm -f abhinav\_demo;

cd /opt/docker;

docker build -t abhinav\_demo .

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

5.Execute Jenkins job

6.check images and containers again on Docker host. This time an image and container get creates through Jenkins job

7.Access web application from browser which is running on container

<docker\_host\_Public\_IP>:8090

