**LAB- Creating Custom Docker Image**

# In this lab you will be understanding how to Create your own docker Image

**STEP 1 – Create Image for a Java Application**

Inside 02-creating-image-forjavaapp folder of docker-labs observe that we have given you a jar file with name ProductService-0.0.1-SNAPSHOT.jar

We want to build our own docker image with ubuntu os as base image.

Then Install java inside it. Copy the .jar file into the image and execute it.

Open Dockerfile given to you inside 02-creating-image-forjavaapp folder of docker-labs and observe how we have done it.

From terminal, cd to 02-creating-image-forjavaapp folder.

To build the imageand tag it with a name productservice, run the below command:

**docker build –t productservice .** (Here “.” represents the folder which contains Dockerfile)

After execution, execute **docker history productservice** to see the layers of this image

Observe the layer where java is downloaded. What is the size of that layer??

This image is exposing port 8080

Firstly, we need to get the ip of the docker machine(linux) in which our containers are started.

So, execute docker-machine ip and note the ip . Mostly it should be 192.168.99.100 by default.

Now run the container using below command:

docker run –p 30080:8080 productservice

Observe the logs . Once the application started, we want to invoke the rest api exposed by that application at /rest/products

Open your browser and give a request to <http://192.168.99.100:30080/rest/products> . You should see list of products in json format.

If you observe the logs, you will understand that it is using an inmemory database HSQL.

Press CTRL+C to get the terminal. U can kill this container using **docker stop <<containerid>>**

Now we want to start a mysql container and make my productservice container to use this mysql db

Start a mysql container and name it as “mysql” using below command:

docker run -d --name mysql -e MYSQL\_ROOT\_PASSWORD=root mysql

For product service which is written in spring if we pass an environment variable **spring.profiles.active=mysql**, this application will connect to mysql database with host name “mysql”

Now we want to run productservice by linking this mysql container so that product service container can use “mysql” as host name to connect to mysql container.

Use the below command:

docker run –p 30080:8080 –e spring.profiles.active=mysql --link mysql:mysql productservice