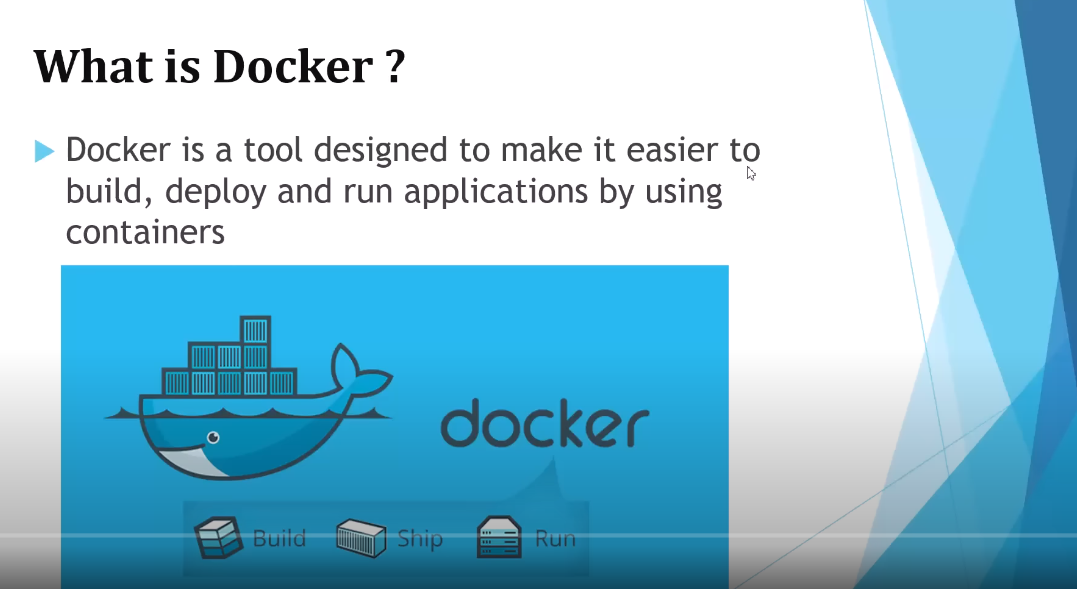
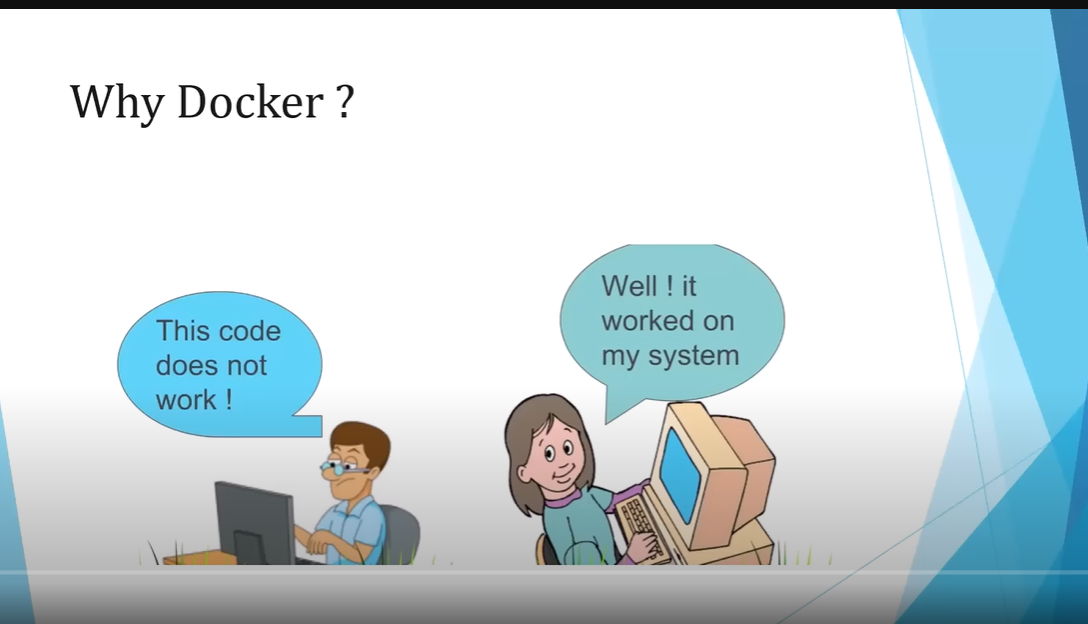
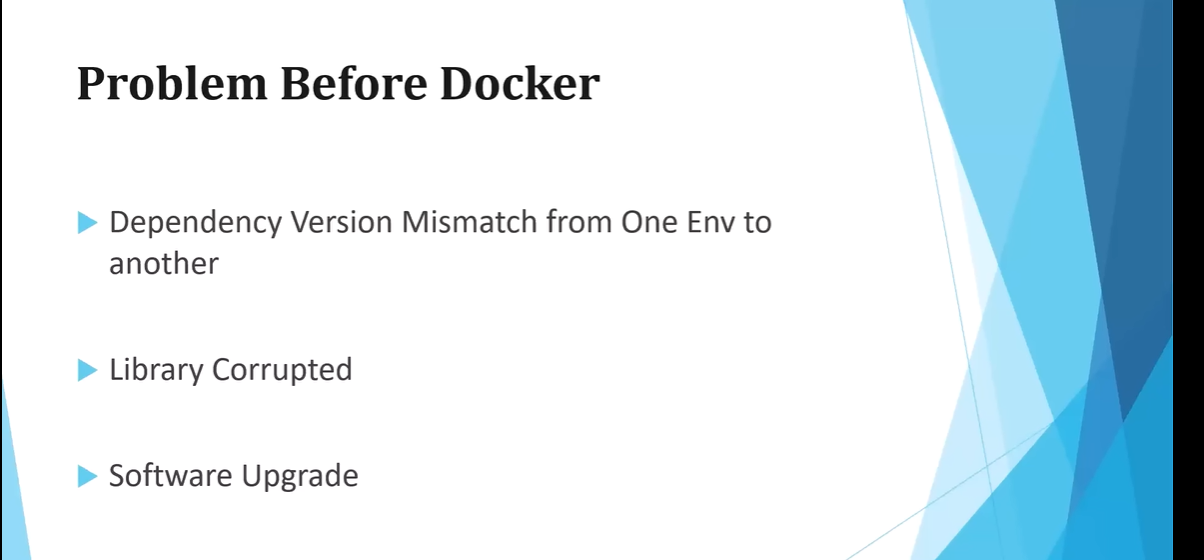
**Docker basics and deploying springboot application in docker**

**Docker:**



****

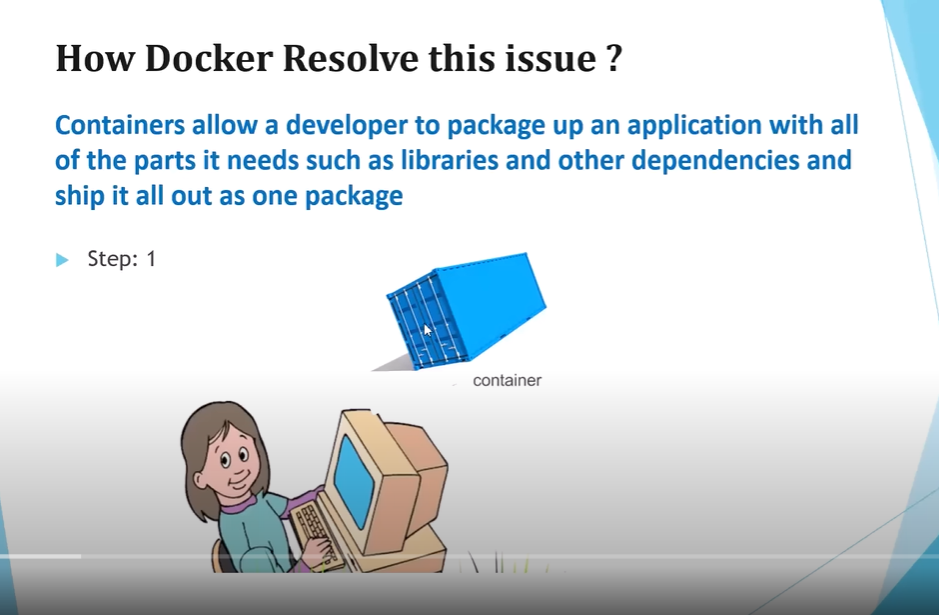


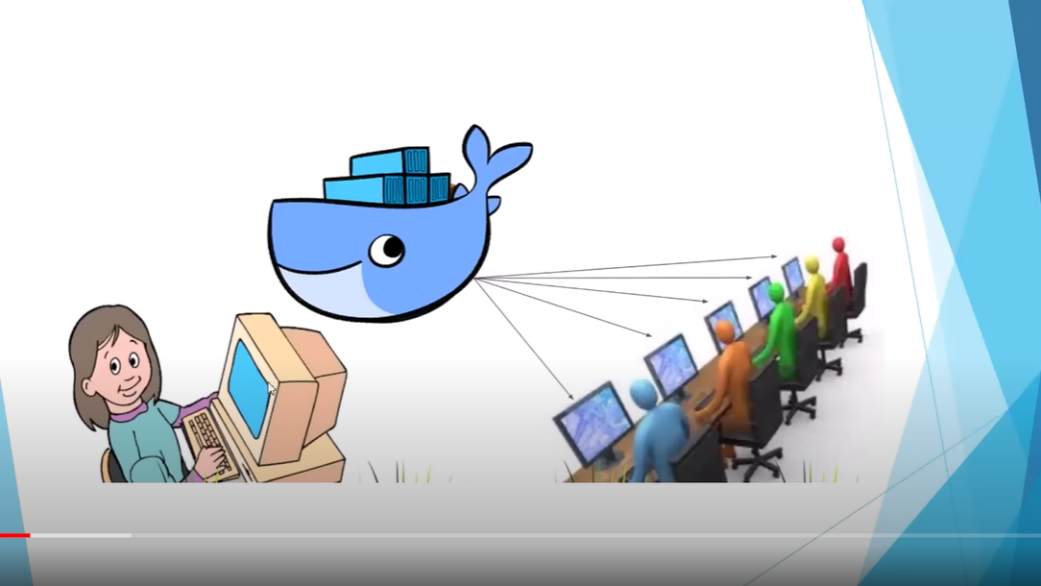
**Problems like:**

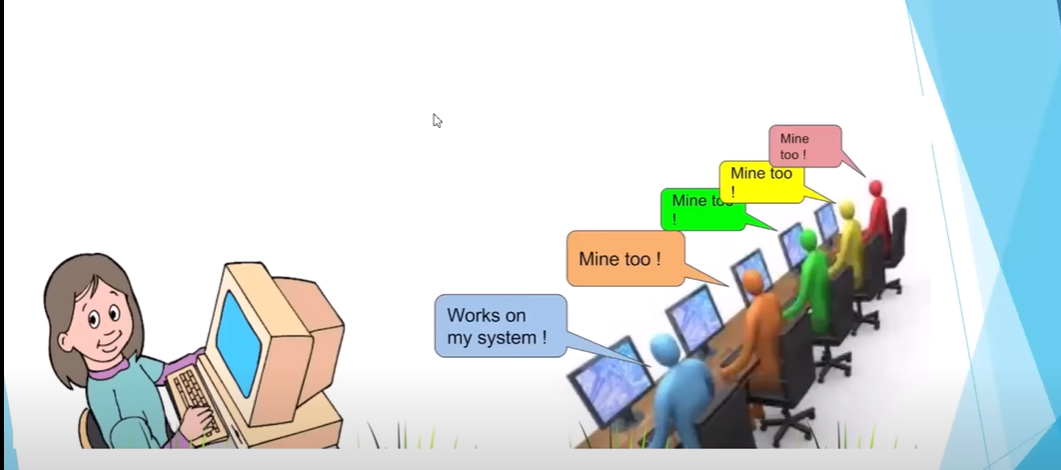
In dev im using spring 2 but in qa im using spring 3.

In dev tomcat jar is available, but  in qa the tomcat jar is corrupted.

In dev im using jdk 8 but in qa it is jdk 7.

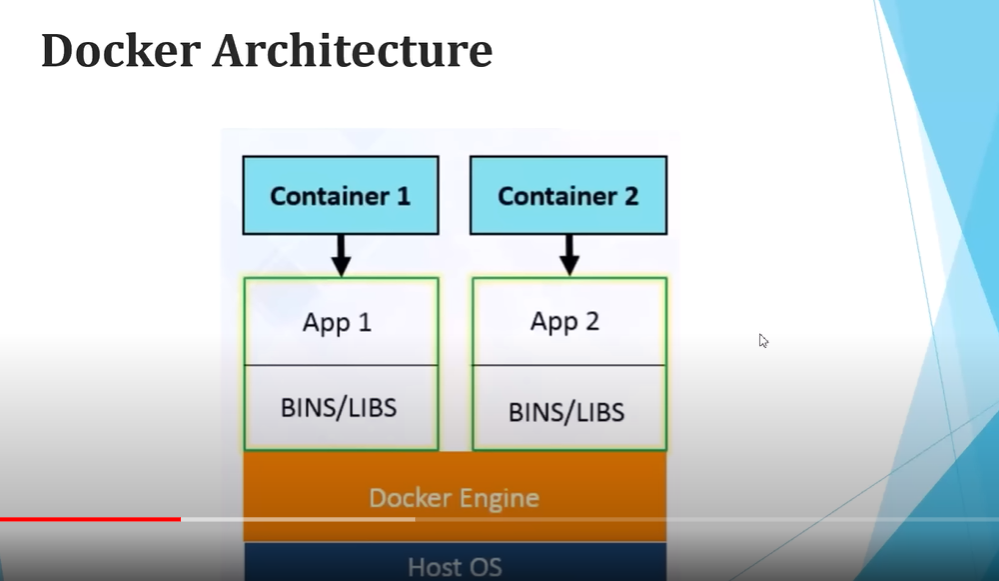






* Developer adds all the required libraries/ software with the code to the container, with the help of docker engine, we can ship(run) the container to(in) each environment like dev, qa,....
* In the above example all the envs point to the same container, the code works in all the envs.

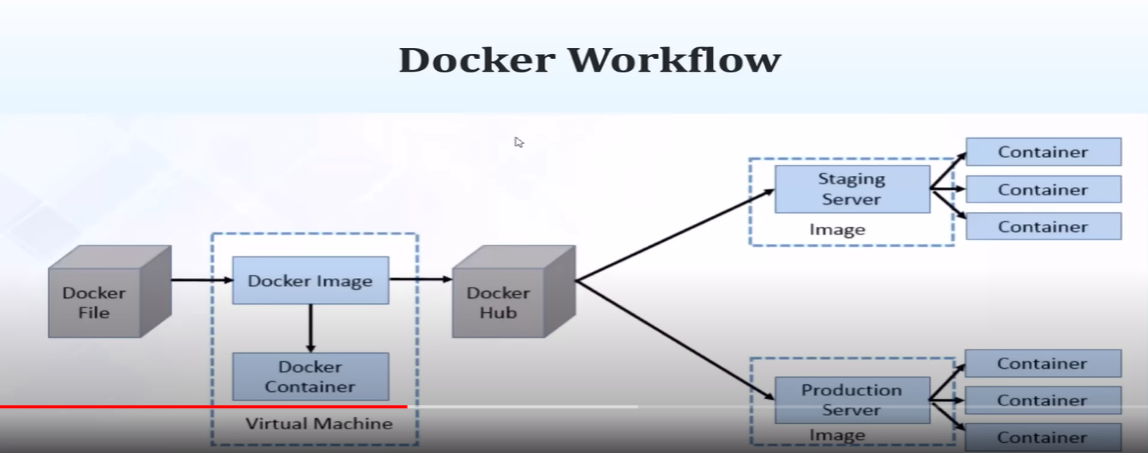
**Docker architecture:**



* Docker engine runs on host os.
* Docker engine consists of n applications.

**Docker workflow:**

* We need to create docker file.
* Using docker file we need to generate docker image(skeleton of our app)
* Docker container is nothing but the running instance of our dockerimage.
* Once we create docker image we need to keep in docker hub.
* Docker hub is the repository for docker. Eg. like maven central repository for maven from where we are getting dependencies for maven. Similarly we can pull docker image from docker hub.



**Steps:**

* Install docker desktop
* docker --version in cmd

**dockerize application just by defining docker file:**

* As a developer we need to create docker file.
* With the docker command we need to create docker image
* There is a docker container which is the running instance of docker image.
* On Docker container the instance of docker image will be running.
* We will create a spring boot application and create a docker file then with the help of docker command we will generate docker image, and we will be deploying it in docker container.

**Steps:**

* Create docker file

#explanation

FROM openjdk:11        #it pulls jdk 11 from docker hub

EXPOSE 8080 #make this port available outside the docker

ADD target/docker-one.jar docker-one.jar                      #jar file path

ENTRYPOINT [ "java","-jar","/docker-one.jar" ] # command to execute jarfile

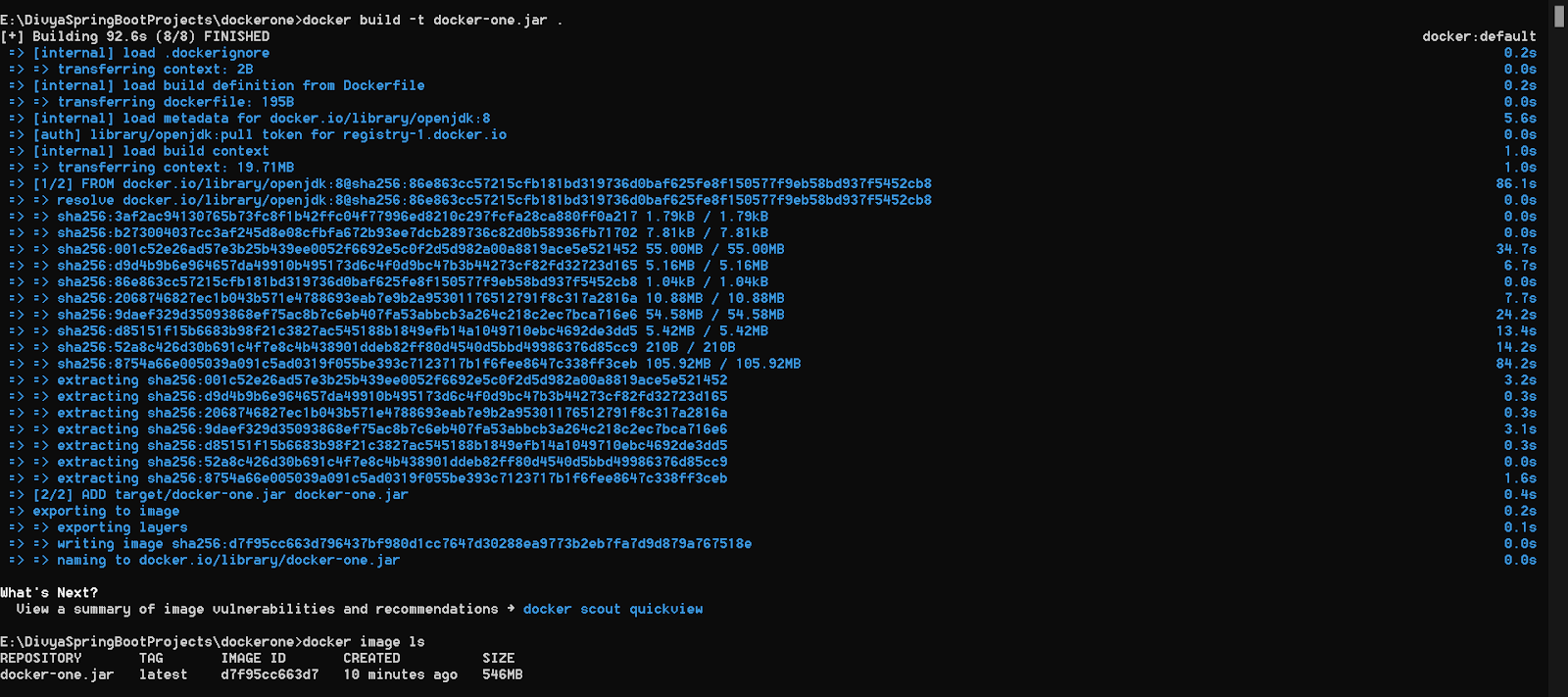
* Maven→update project
* Run as maven clean
* Run as maven install

* Generate docker image using docker command

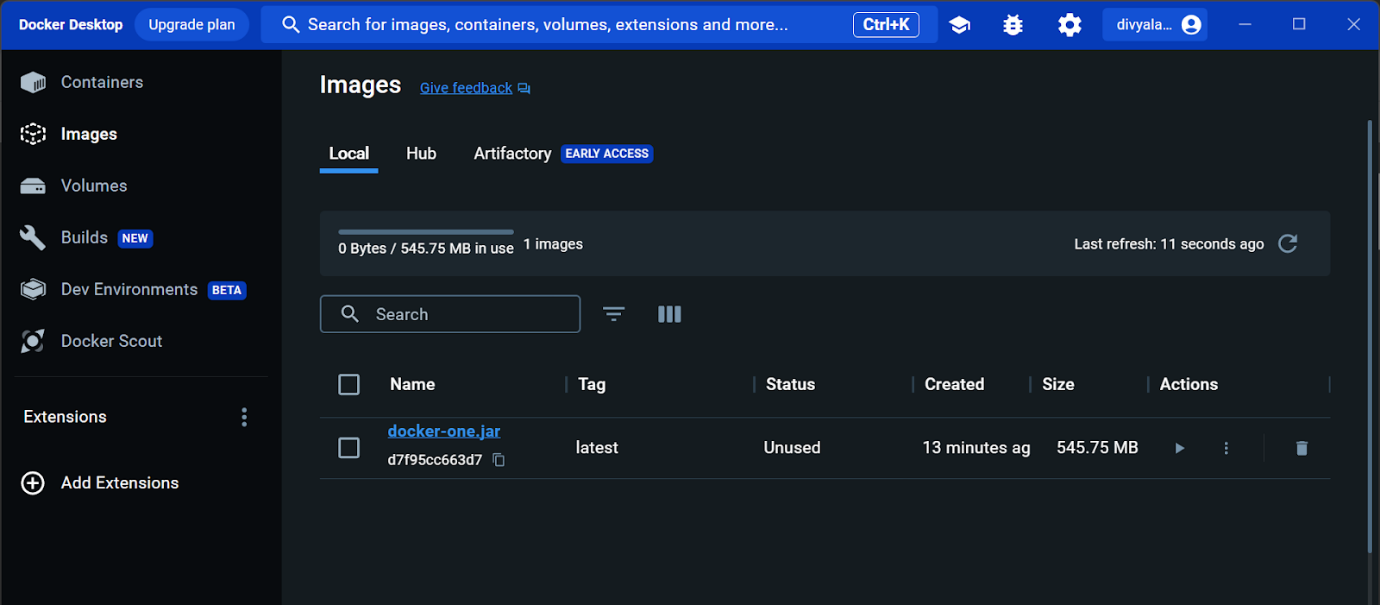
**E:\DivyaSpringBootProjects\dockerone>docker build -t docker-one.jar .**

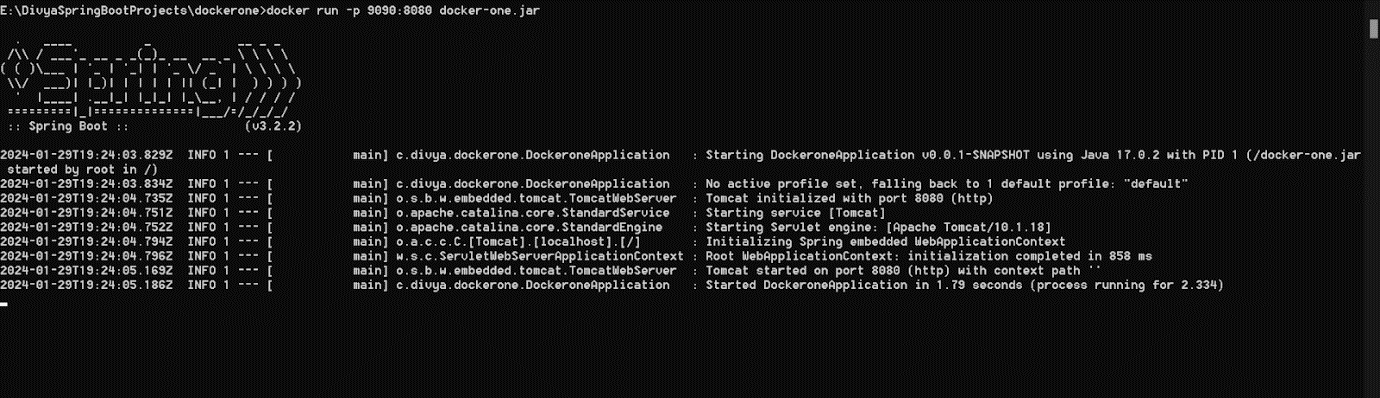
 .  -->root directory

* Display docker images **E:\DivyaSpringBootProjects\dockerone>docker image ls**

****

* We can run the docker image in docker desktop or by using command

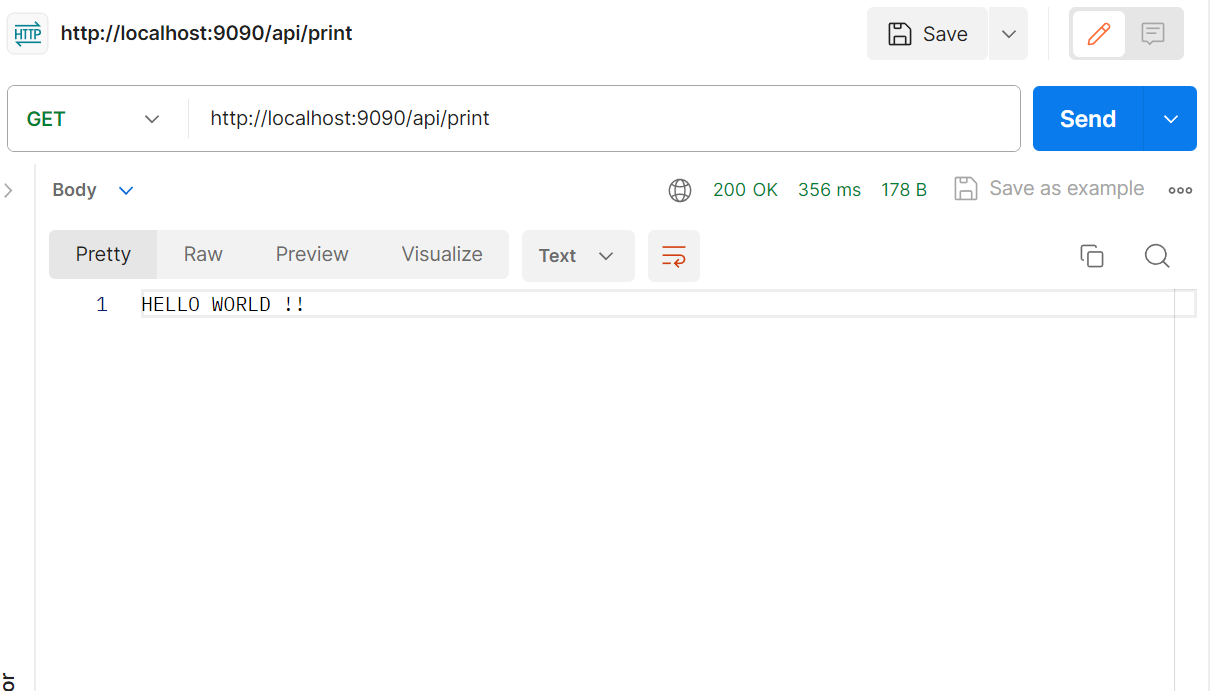
****



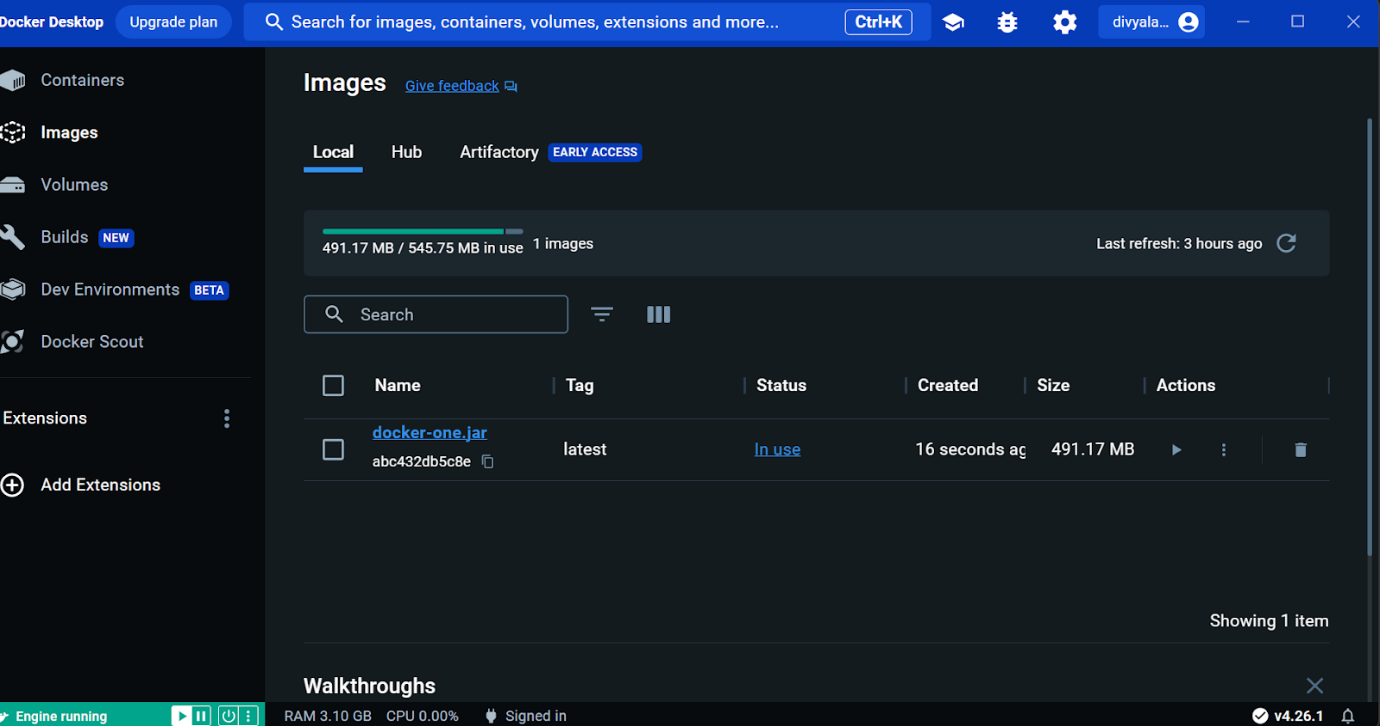
 the port 8080 on the container should be mapped to the port 9090 on the Host OS.

**E:\DivyaSpringBootProjects\dockerone>docker run -p 9090:8080 docker-one.jar**

**In postman:**



**Docker image:**



**Docker container:**

