***Deploying Spring-boot application using Docker***

1. Create a maven based spring-boot project & make sure you have added below plugin in pom.xml to create executable jar file:

*<build>*

*<plugins>*

*<plugin>*

*<groupId>org.springframework.boot</groupId>*

*<artifactId>spring-boot-maven-plugin</artifactId>*

*<executions>*

*<execution>*

*<goals>*

*<goal>repackage</goal>*

*</goals>*

*<configuration>*

*<mainClass>*

*com.spring.SpringRestDemo.App*

*</mainClass>*

*</configuration>*

*</execution>*

*</executions>*

*</plugin>*

*</plugins>*

*</build>*

Change your <mainClass> value as per requirement.

1. Make sure your have written below Dockerfile at the location same as pom.xml:

FROM openjdk:8

ADD target/SpringRestDemo-0.0.1-SNAPSHOT.jar myapp.jar

ENTRYPOINT ["java", "-jar", "myapp.jar"]

EXPOSE 9999

1. Open console & cd to pom.xml file location.
2. Run “mvn install” command. It will create an executable jar file inside ‘target’ directory.
3. Test this executable jar by running “java –jar SpringRestDemo-0.0.1-SNAPSHOT.jar”. It should start the tomcat container on port 8080 & make sure you get correct response when hit <http://localhost:8080/myapp/order?orderId=25>
4. Make sure you have started Docker Desktop service by running the command “docker version”.
5. Login to docker hub from console.

>docker login

1. Now, we have to build docker image using this executable jar.

>docker build –t anand0204/spring-boot-order-docker .

Here anand0204 is my docker hub login name.

Make sure ‘anand0204/spring-boot-order-docker’ is created by running “docker images –a” command.

1. Now, we need to create a container using this image & run that container.

>docker run –p 9999:8080 anand0204/spring-boot-order-docker

Here, container will be running on port 8080, however it is exposing the port 9999. Hence, while accessing the application, we should use 9999 port i.e. <http://localhost:9999/myapp/order?orderId=25>

1. If application is running fine then you may want to push this image to docker hub.

>docker push anand0204/spring-boot-order-docker

Now, login to <https://hub.docker.com> & make sure the image ‘spring-boot-order-docker’ is seen in the repositories list. It means, anybody can now pull this image & run it.

1. In case, you wish to delete this image from local repository then first stop the container(docker stop %container\_id%), then delete the container (docker rm %container\_id%) & finally delete the image (docker rmi %image\_id%)
2. If you wish to delete this image from docker hub then go to Settings tab & click on ‘Delete’ button.