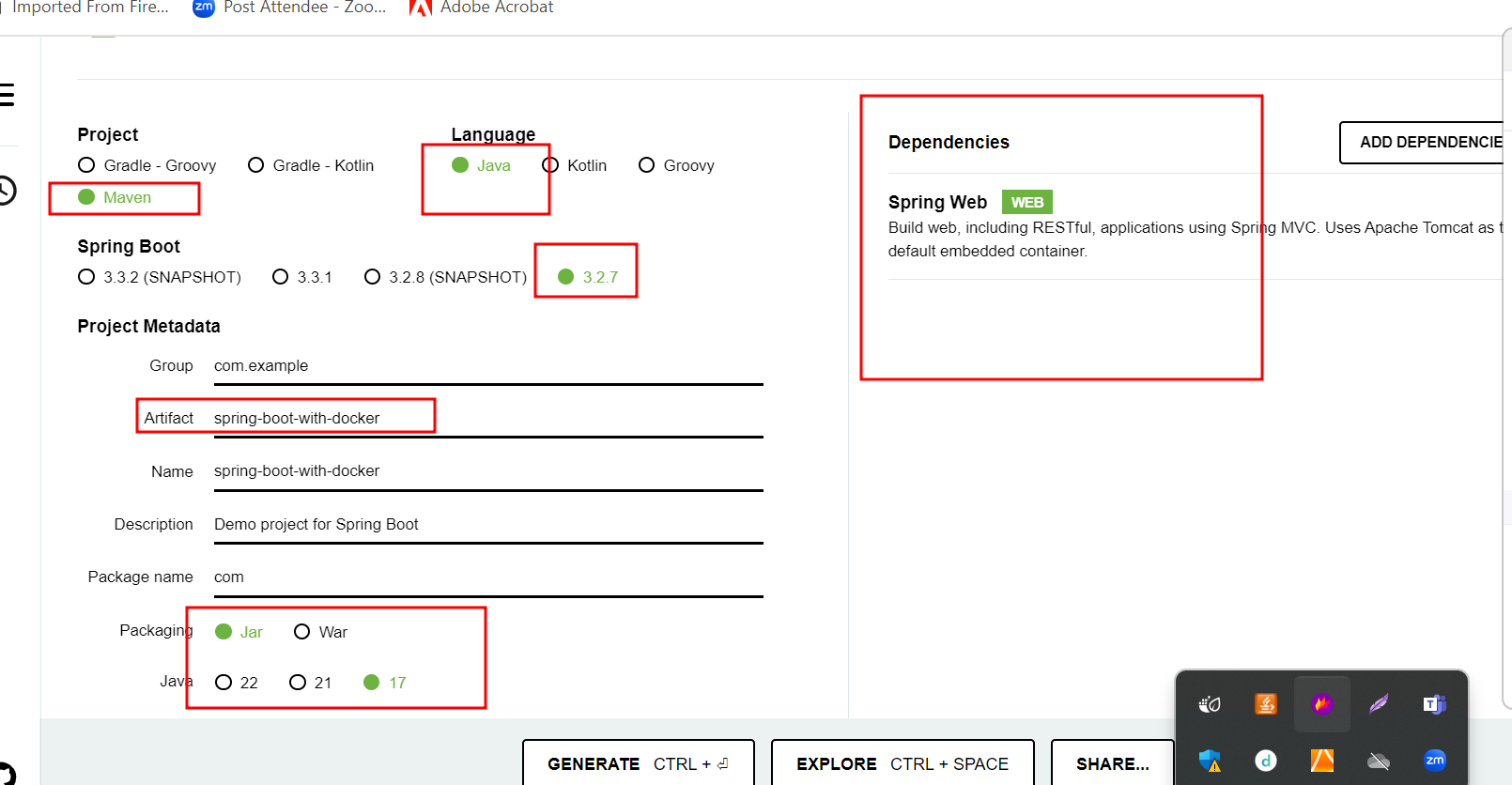
* CB FSD - Integration and Deployment

Day 2 : 9 July 2024

Creating docker image to run spring boot application

Using spring initializer create spring boot project with single starter ie web starter



Create more than one end point and using Eclipse IDE create the jar file with the option run 🡪 maven install.

Then create the Dockerfile

FROM openjdk:17

COPY ./target/spring-boot-with-docker-0.0.1-SNAPSHOT.jar .

CMD ["java","-jar","spring-boot-with-docker-0.0.1-SNAPSHOT.jar"]

Now we need to build the image.

docker build -t my-spring-boot . -f Dockerfile

please verify image created or not using command as

docker images

if image is responsible to run web application we need follow below command to run the image.

docker run -d -p 9090:9090 imageName/imageId

-d : detached mode or background mode

-p : publish port number

9090: right side or red colour is actual application port number.

9090 : left side or green colour is publish port number. It can be same or different.

Now run the image

docker run -d -p 9090:9090 my-spring-boot

docker run -d -p 9091:9090 my-spring-boot

docker run -d -p 9092:9090 my-spring-boot

running the container with user defined name.

docker run --name my-springboot-container-info -d -p 9093:9090 my-spring-boot

then open the browser and check using URL as

<http://localhost:9090>

<http://localhost:9091>

<http://localhost:9092>

container related commands

below command is use to find all running container.

docker ps

or

docker container ls

if we want to view all container ie running or stopped.

docker ps -a

stop container

docker stop containerId/containerName

start container

docker start containerId/containerName

remove container

docker rm containerId/containerName (container not running mode)

or

docker rm containerId/containerName -f

remove the image

docker rmi imageName/imageId

or

docker rmi imageName/imageId -f

docker provided pre defined images for database like mysql or mongo db etc.

docker run --name my-sql-container -e MYSQL\_ROOT\_PASSWORD=root -d -p 3307:3306 mysql:8.0

-e environment variable

-d : detached mode

-p : publish port number

Using docker ps verify mysql container status

docker ps

now we need to open the mysql container os terminal

docker exec -it containerName/containerId bash

docker exec -it my-sql-container bash

then we need to connect that database.

mysql -u root -p

* root

twice exits to come out

creating image for angular application

create folder and insider this folder create the angular project using command as

ng new angular-docker

routing 🡪no

styling 🡪 css

cd angular-docker

open this project in VS code

in app.component.html file do some changes.

app.component.html file

<div>

  <h3>Welcome to Angular application running using Docker!</h3>

</div>

Please run the project using command as ng serve -o

To build the project like creating jar of war file in spring boot

ng build

after build successfully it create dist folder and inside this folder it contains all build files.

Docker provided lot of server images like tomcat, jboss, nginx etc.

Dockerfile

FROM nginx

COPY ./dist/angular-docker/ /usr/share/nginx/html

Create the image

docker build -t my-angular-test . -f Dockerfile

nginx default port number is 80.

docker run -d -p 80:80 my-angular-test

open the browser and hit the URL as <http://localhost:80>

Publish the Image in Docker hub account.

docker login

it may ask you emailid and password and make sure connected successfully.

Before push we need to create the tag for that image

docker tag imageName dockerhubaccountid/imageName:version

version it can be 1,2,3 or a,b,c etc.

docker tag my-angular-test akashkale/my-angular-test:a1

after tag created successfully we can push the image

docker push akashkale/my-angular-test:a1

docker run -d -p 81:80 akashkale/my-angular-test:a1

release new version of an image.

1. Do the changes in your angular project.
2. Re build the project using ng build command
3. Re create the image docker build -t my-angular-test . -f dockerfile
4. Create tag with different version docker tag my-angular-test akashkale/my-angular-test:a2
5. Then push this image in docker hub account.

docker push akashkale/my-angular-test:a2