**Day 35:**

DevOps : Development and Operation

Docker is an open source platform that helps to developer as well admin to build, package(jar or ware file) and run the application in light weighted and portable container environment.

Virtualization : VMWare software which help to run the virtual OS on base machine. Using virtualization we run or create abstract version of an OS.

Containerization : using containerization we can run abstract version of an application.



Container : A light weighted standalone and executable package that contains all the code and need external dependencies to run the application or software.

Docker Image: it is a ready only template file which is responsible to run the application with help of container.

Docker File: it is a text file which contains set of instruction to create the image.

Docker hub: It is like a git hub which help to push or publish as well as pull user defined as well as pre defined images.

Docker engine : it is an run time environment for Docker containers.

Create the Docker hub account and login through docker desktop

Open the command prompt

docker --version

docker info

docker images this command is use to display all images present in local machine.

docker pull imagename this command is use to pull the image

docker pull hello-world hello-world is one of the pre defined image. It pull in local machine.

To run the image

docker run imagename/imageId this command is use to run the image

**busybox : os image**

**creating custom**

1. Display simple welcome message using user defined images.

Create the file with name as Dockerfile

FROM busybox

CMD ["echo","Welcome to Docker Image created by Akash"]

To create the image

docker build -t my-busybox . -f Dockerfile

docker run my-busybox

1. creating image to run the Java application

**Demo.java**

public class Demo {

    public static void main(String[] args) {

     System.out.println("Welcome to Java Program running using Docker");

    }

}

**Dockerfile**

FROM openjdk:8

COPY . /usr/src/myapp

WORKDIR /usr/src/myapp

RUN javac Demo.java

CMD ["java", "Demo"]

docker build -t my-java . -f Dockerfile

docker run my-java

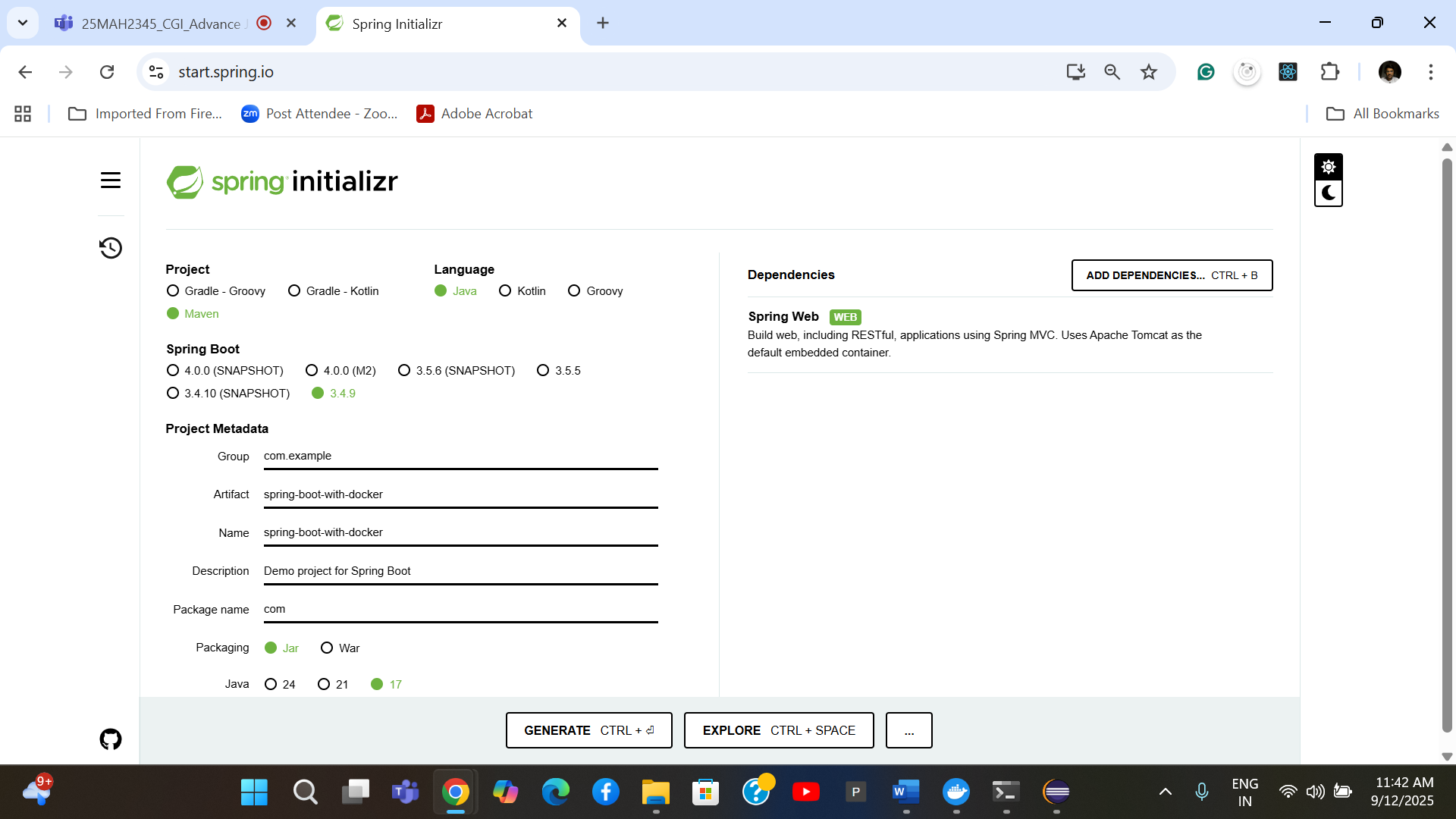
1. creating image to run spring boot application

create simple spring boot with rest api or thymleaf etc.

after development you need to create jar or war.

We can create using mvn command or using eclipse ide.

Then we need create docker file which is responsible to run the spring boot application



Create the one or many rest api

Then you need to build the project

Right click on project and docker build and goal must be package.

Then create the Docker file

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"]

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

if image is responsible to run web application

docker run -d -p 8080:8080 imageName/imageId

-d detach mode or background

-p public port number

8080: left side or orange colour expose port number can be same or different

8080: right side or red colour actual port number.

docker run -d -p 8080:8080 my-spring-boot

docker run -d -p 8081:8080 my-spring-boot

docker run -d -p 8082:8080 my-spring-boot

to view running container

docker ps

or

docker container ls

To publish the image on docker hub registry

before push we need create the tag for that image

docker tag imagename dockerhubaccountid/imageName:tag

docker tag my-spring-boot akashkale/my-spring-boot:cgi1.0.0

Now you need to push the image

Docker push dockerhubaccountId/imagName:tag

docker tag imagename dockerhubaccountid/imageName:tag

docker push akashkale/akashkale/my-spring-boot:cgi1.0.0