Day 11 – 10 Apr 2024

Docker Intro

Docker is an open-source platform is used to containerized our application or software, using which we can easily build our application and package them with required dependencies to run the application into container and these containers are easily shipped to run on other machine without run time environment.

Docker is an advanced version of virtualization.

Container : run time environment or engine which provide service to run the application.

Docker images : docker image is read only template which is responsible to runt the application.

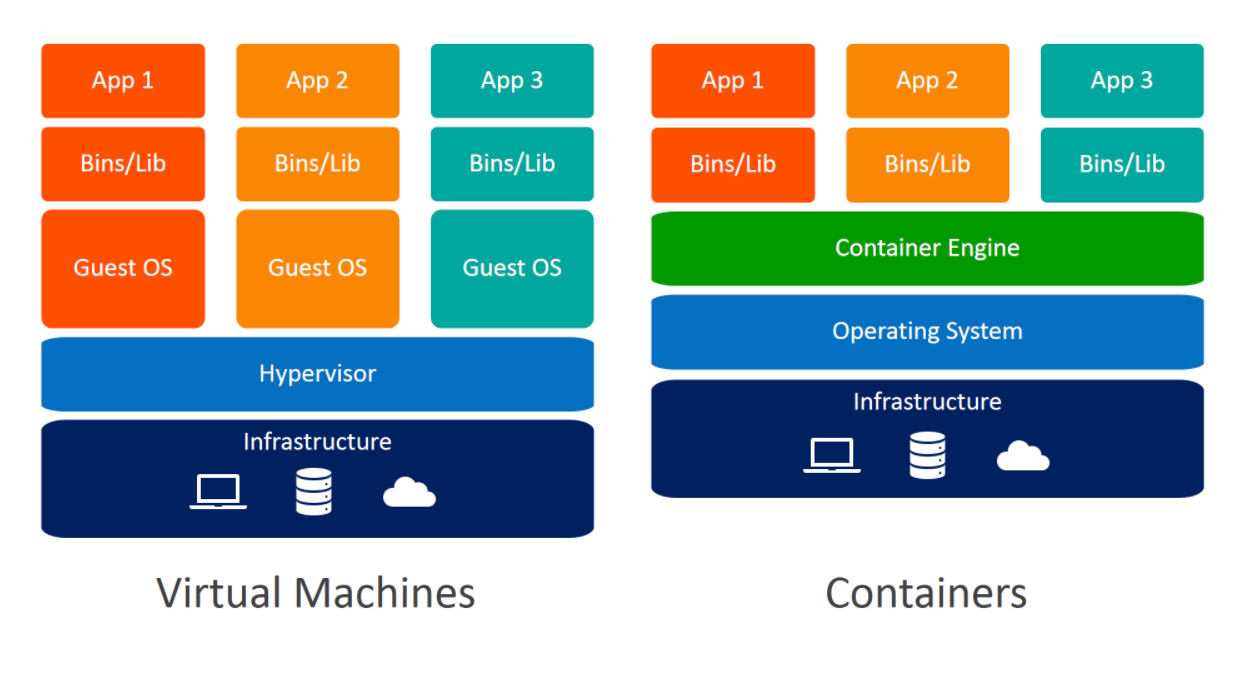
Virtualization Vs Containerization

VMWare software

Base machine 16 RAM and 1 TB hard disk.

Guest OS -🡪 4GB ram 50 GB memory

10 Guest OS.



Using Virtualization we can create abstraction version of an OS.

Using Containerization we can create abstract version of an application.

**docker --version**

**docker info**

**docker images**

**docker pull imageName/imageId**

**docker run imageName/imageId**

**docker file : docker file contains set of rules which help to create the image. It is source code for the images.**

**Creating image to display message**

**Dockerfile**

FROM busybox

CMD ["echo","Welcome to Docker image created by akash kale!"]

**docker build -t my-busybox . -f Dockerfile**

**docker images**

**docker run my-busybox**

**creating image to run java program**

**Demo.java**

class Demo {

    public static void main(String[] args) {

        System.out.println("running java program using Docker!");

    }

}

**Dockerfile**

FROM openjdk:11

COPY Demo.java .

RUN javac Demo.java

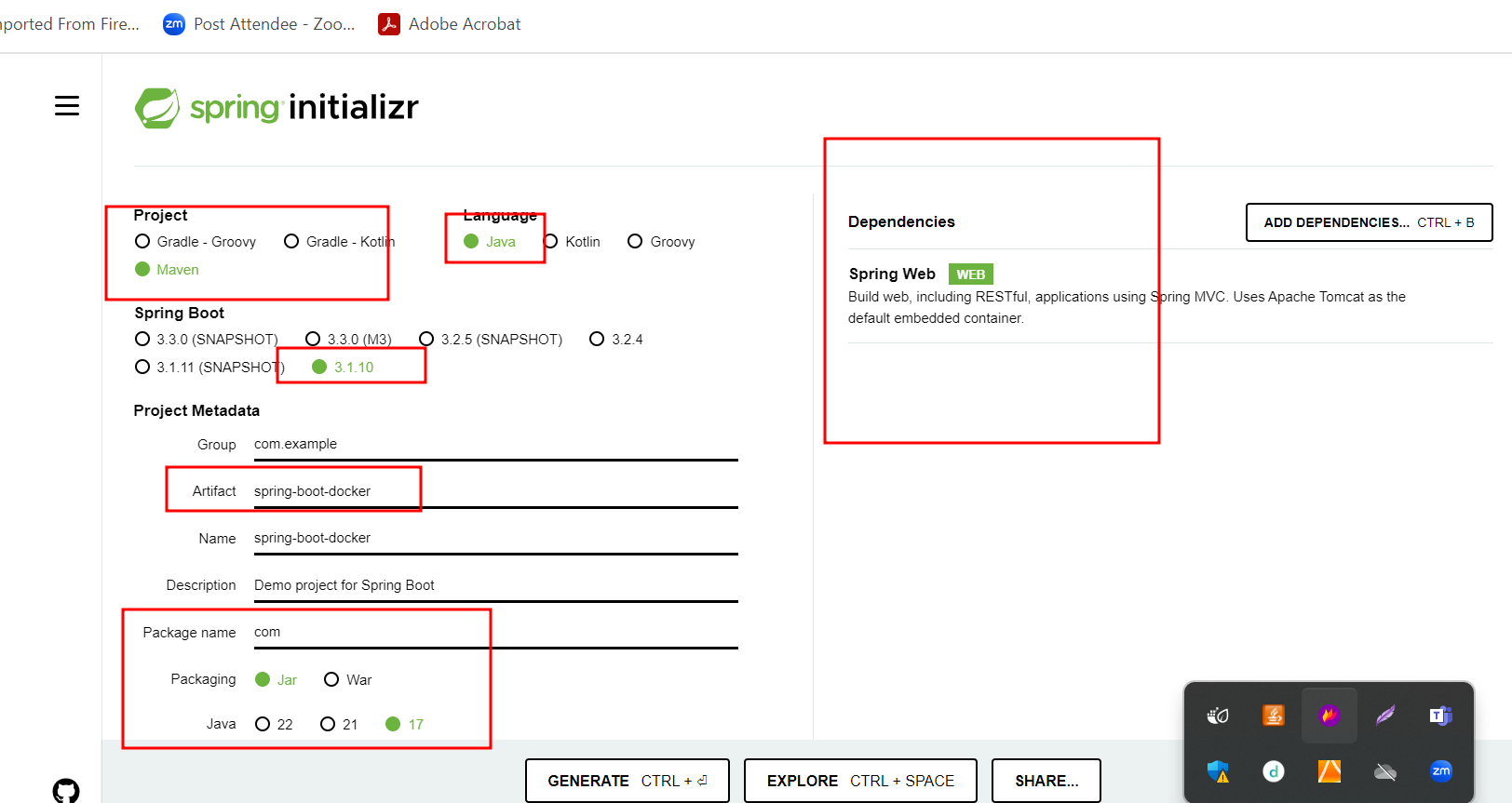
CMD ["java","Demo"]

**docker build -t my-java . -f Dockerfile**

**docker run my-java**

**Creating the image to run the spring boot project.**

**Create spring boot project with web starter**



**Create simple rest api**

**package** com.controller;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

**public** **class** SimpleController {

@GetMapping(value = "/")

**public** String sayHello() {

**return** "Welcome to Spring boot with Docker created by Akash Kale";

}

}

**Build the project or create the jar file**

**Then Dockerfile**

FROM openjdk:17

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

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

**docker build -t my-spring-app . -f Dockerfile**

**if image is responsible to run the web application then we need to follow below commands**

**docker run -d -p 8080:8080 my-spring-app**

**-d : ditched mode**

**-p : publish port number**

**Red colour port number actual port number and green colour port expose port number can be same or different**

**docker ps**

**if you want to push or publish your image in one of the remote repository**

**docker login (by default link with dockerhub account).**

**docker tag imageName dockerhubaccoutid/imageName:version**

**docker push dockerhubaccoutid/imageName:version**

**docker run -d -p 9191:8080** akashkale/my-spring-app:vv1