9 July – 2024

Day 1

Integration and deployment

Docker

Docker compose

CI and CD using Jenkin and Jenkin pipeline

Cloud computing using AWS : S3 and EC2 instance

Docker : Docker is a software platform which allow use to build, test and deploy application quickly using Docker container.

Using the Docker we can create containerization application.

Docker also known as adv Version of virtualization or VM machine.

If we want to run any application software we need system software ie OS it can be win, linux, mac etc.

One machine with window os and installed required software to run the application

Another machine with mac/linux and installed required software to run the application.

In one machine we can installed multi OS.

But we can run only one OS at time.

VW ware software. This software help us to run multi OS or guest OS on base machine. With help of VM ware software we can run more than one Guest OS is known as Virtualization. VMWare software provide the features as running the more than one Guest OS in the abstraction version of an OS.

Docker is use to create containerization application.

Container : container is known as run time environment or engine.

JRE : Java Run time environment.

Node : Node JS known as JavaScript run time environment.

Web Container : it is a part of web server which is responsible to run the servlet and jsp.

Spring container.

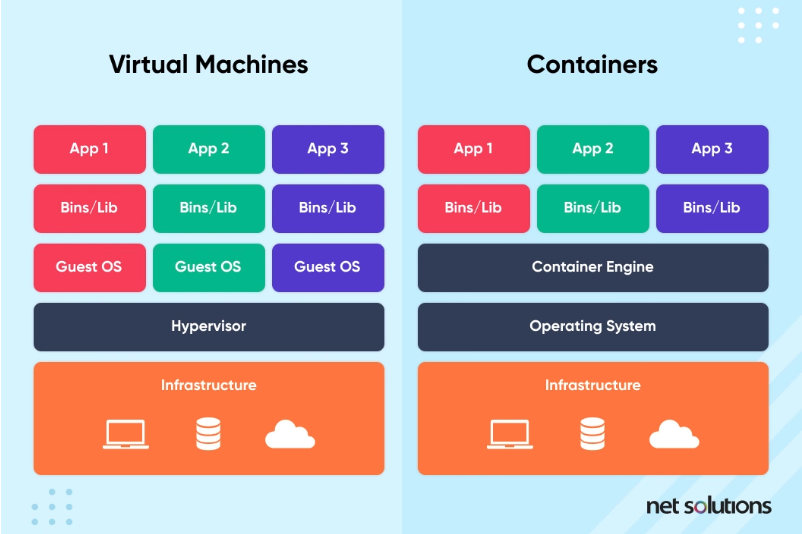
Docker container : it is a type of container which is responsible to more than one application with their dependencies in the form of images.

Docker image : Docker images is a read only template file which contains application details with required dependencies to run the application with help of container.

Containerization : it is an abstract version of an application.

Virtualization :it is an abstraction version of an OS.

Docker engine provide OS to the application using docker images.



In virtualization base machine need to share the resource like RAM and External memory etc.

Open Terminal or CMD in local machine or VM ware lab.

Non window user every command start with pre-fix sudo

docker --version window user

or

sudo docker --version non window user

docker info this command provide the docker details.

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

Docker image : it is a read only template file which is responsible to run the application using docker container.

docker pull imageName

hello-world

docker pull hello-world this command is use to pull the image from docker hub to local machine.

To run the image

docker run imageName/imageId this command is use to run the image.

Please create or signup for Docker hub. Docker hub provide us public as well as private repository which help to publish as well as pull docker images. Those images can be pre-defined for pull purpose as well as for push we can create custom image with help of pre defined images.

**Please pull busybox image and run it.**

docker run -it ubuntu bash this command is use to pull, run and open ubuntu os image terminal

creating custom image

1. Creating image to display welcome message.

**Dockerfile**

FROM busybox

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

docker build -t my-busybox . -f Dockerfile this command to create the

image

docker images

docker run my-busybox

1. Creating image to run Java application (Core Java Program)

Demo.java

public class Demo {

    public static void main(String[] args) {

        System.out.println("Welcome to java program running through docker");

    }

}

Dockerfile

FROM openjdk:11

COPY Demo.java .

RUN javac Demo.java

CMD [ "java","Demo" ]

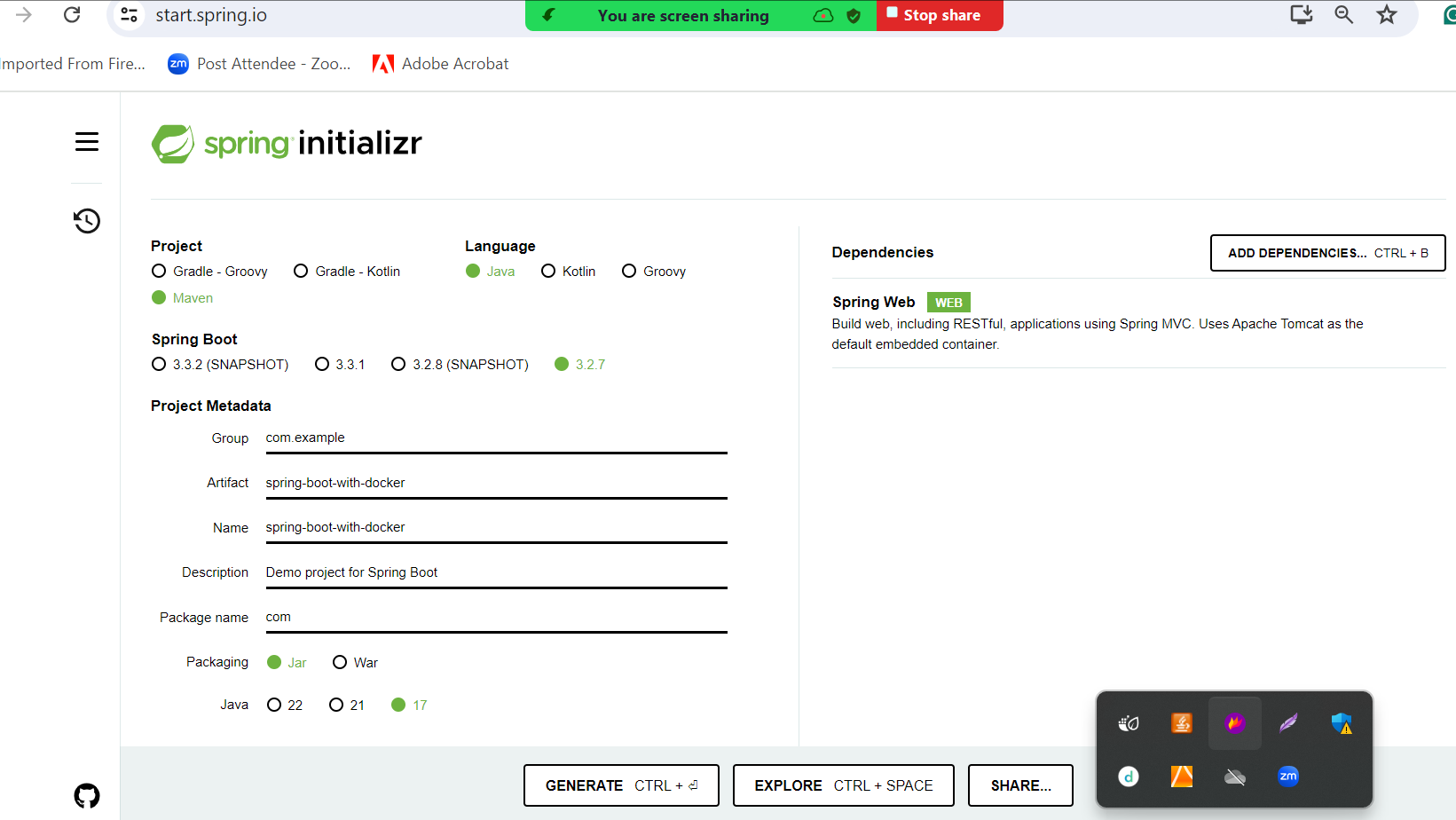
docker build -t my-java . -f Dockerfile

docker images

docker rum my-java

1. Creating image to run spring boot application

Using spring initializer we create simple spring boot application with web starter.



Please create more than one end points

Then create jar file using eclipse IDE with help of maven run 🡪 maven install

Then create 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"]

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