11 July – 2024

Day 3

Integration and deployment

Docker

Docker compose : Docker compose provide tool kit which help to run more than one containers. These container running independently or they are communicating with each other’s to do some task. To write all configuration details to run more than one container we need to use docker-compose.yml or docker-compose.yaml

Yaml : Yet Another markup language.

docker-compose --version

Download the current stable release of Docker Compose

sudo curl -L https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

Give the permission

sudo chmod +x /usr/local/bin/docker-compose

docker-compose version

docker-compose.yml

version: '3.6'

services:

  akash-image:

    image: akashkale/my-reactjs:a2

    container\_name: akash-reactjs-container

    ports:

      - 81:80

  student1:

    image: buschfunk/react-docker-app:latest

    ports:

      - 82:80

  student2:

    image: dragosmn/my-reactjs:a2

    ports:

      - 83:80

  student3:

    image: paulodm/my-reactjsgood:v2

    ports:

      - 84:80

Create separate folder as : docker compose project or files

Student multiple container

docker-compose up : it pull all images and start all service or container

docker-compose up -d : it start in background or detached mode

docker ps : check all running containers.

docker-compose down : to stop all container details part of that file.

docker-compose start

docker-compose stop

docker system prune -a : it is use to remove all images, stopped container, cache as well as network.

Different OS Different OS Different OS

Front end technologies backend technologies

React js container spring boot container mysql container

http call TCP/IP connection

using axios

or fetch function

Capstone Project

Frontend folder -🡪 React JS

Backend folder -🡪 Spring boot with MySQL Database

SignIn and SignUp

Admin account

User Account

Using docker compose we run spring boot container and mysql container to make backend end point ready.

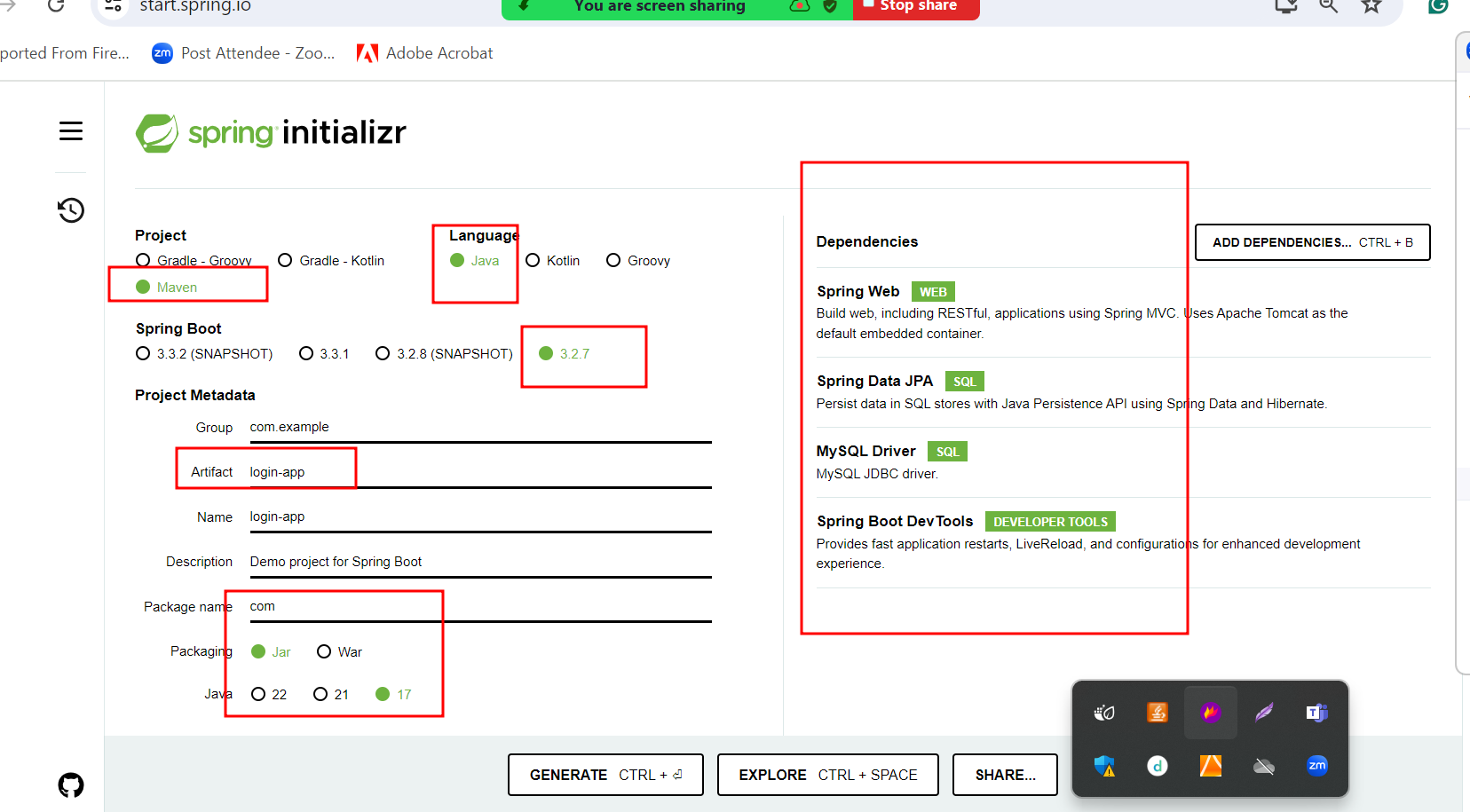
Spring boot project ie login-app

Starter -🡪 Web Starter

Starter 🡪JPA Starter

Starter 🡪Devtools

Dependencies 🡪 My SQL



Create end point and create the jar file

version: '3.8'

services:

  mysql-service:

    image: mysql:8.0

    container\_name: my\_sql\_container

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

      - MYSQL\_DATABASE=mydb

    ports:

      - 3307:3306

    restart: always

  spring-boot-service:

    build: ./backend-app/login-app/login-app/

    container\_name: spring\_boot\_container

    ports:

      - 9090:9090

    depends\_on:

      - mysql-service

    restart: always

Now we need to run docker compose file

docker-compose up –-build -d

using docker ps please check container running or not. If you get any error please debug.

Then test those end point using post man client.

Now we connect mysql terminal OS using below commands.

docker exec -it my\_sql\_container bash

after connected successfully now you can connect mysql terminal of docker container

mysql -u root -p

password : root

show databases

use mydb;

select \* from login;

please create admin account using sql query

insert into login values('admin@gmail.com','admin@123','admin');