Ex:No:5	Testing and Deploying SpringBoot Application
Date:	

Aim

- To perform Junit test for REST API and to deploy springboot application in docker.

Procedure

1. **Perform Junit testing in springboot application based on controller/service class**

Unit Testing is a one of the testing done by the developers to make sure individual unit or component functionalities are working fine.

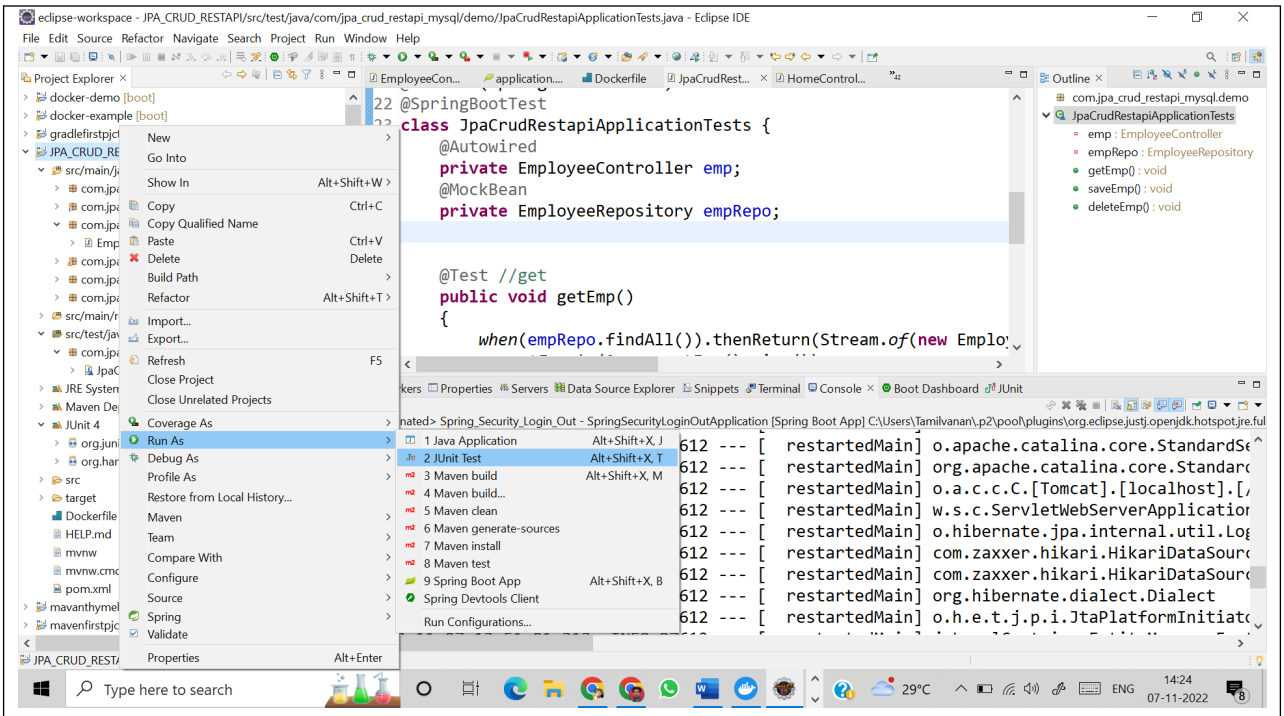
Add the following code in src/test/java application file

```
@RunWith(SpringRunner.class)
@SpringBootTest
class JpaCrudRestapiApplicationTests {
	@Autowired
	private EmployeeController emp;
	@MockBean
	private EmployeeRepository empRepo;

	@Test //get
	public void getEmp()
	{
		when(empRepo.findAll()).thenReturn(Stream.of(new
Employee(10,"xxx",50,50000),new Employee(11,"yyy",40,35000)).collect(Collectors.toList()));
		assertEquals(2,emp.getEmp().size());
	}
	@Test //Insert
	public void saveEmp()
	{
		Employee empl=new Employee(20,"UUU",19,10000);
		when(empRepo.save(empl)).thenReturn(empl);
		assertEquals(empl,emp.createEmp(empl));
	}

	@Test //Delete
	public void deleteEmp()
	{
		int id=10;
		emp.deleteEmp(id);
		verify(empRepo,times(1)).deleteById(id);
	}


}
```

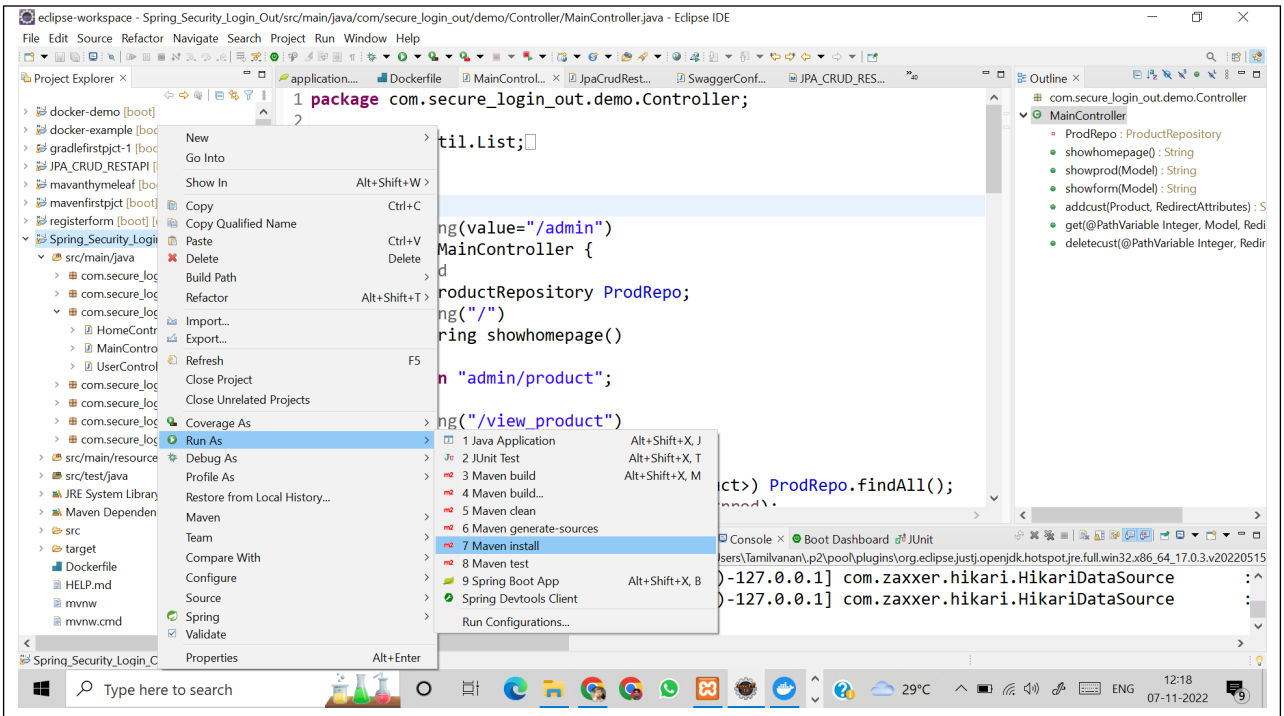


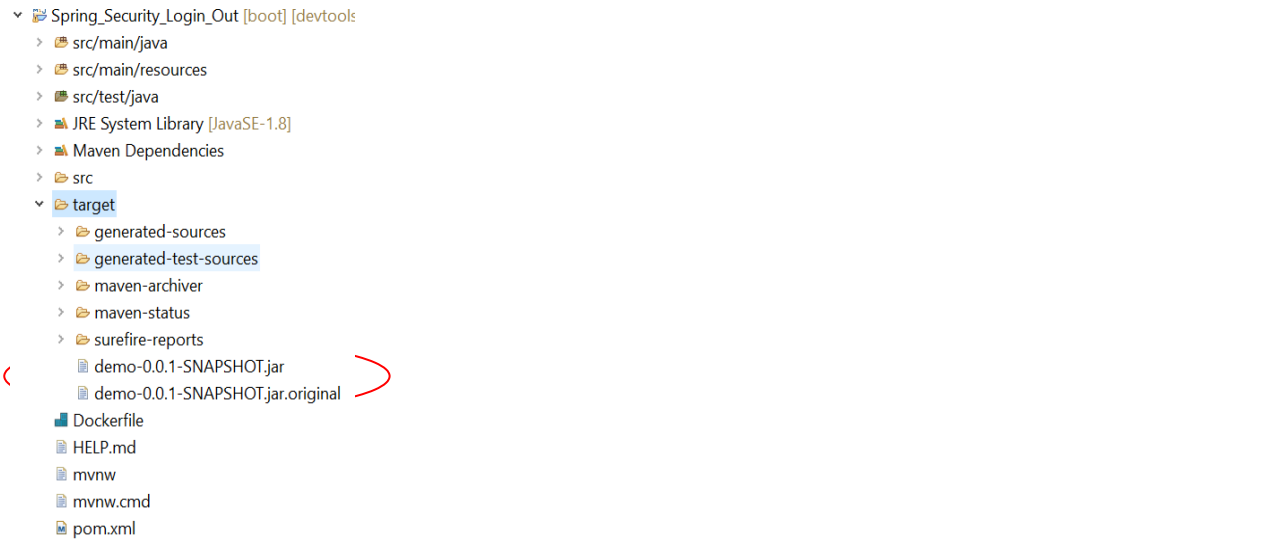
2. Deploy Locally a Spring Boot Application in Docker

Prerequisites

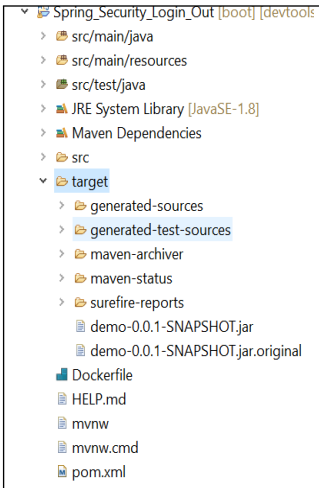
- Docker environment installed and configured..
- Have a Docker Hub account. Register for a free account.

Step: 1 create the .jar file for your SpringBoot Application





Step: 2 create a Docker file



Dockerfile contains the commands to create the image

```
From openjdk:8
ADD target/*.jar secapp.jar
ENTRYPOINT ["java","-jar","secapp.jar"]
```

FROM - Must be the first non-comment instruction in the Dockerfile. This command creates a layer from the Docker image. In our case, we have used java:8 which means this application will run on Java 8.

ADD - This command helps to take a source and destination. Normally, the source is your local copy. The COPY command also does same thing, but there is a small difference between the COPY and ADD commands.

ENTRYPOINT - It's similar to CMD, where our command/jar file will be executed.

Step 3: Run the command to build the image and deploy it to Docker

Create a Docker image file.

Open Command Prompt and follow the below steps…

The below command is used to create the image file

>> **docker build -t secapp .**

check the created image by using the "**docker images**" command

Create an image for mysql

>> docker pull mysql

Load mysql image into container

>>docker run -p 3308:3306 --name mysqlsec -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=User_Mgmt -e MYSQL_USER=user -e MYSQL_PASSWORD=password -d mysql

Create a network to connect the mysql with springboot application

>>docker create network spring-net

Check the created network is added in docker

>>docker network ls

Connect the created network with mysql

>>docker network connect spring-net mysqlsec

Check the connection is established with mysqlsec

>> docker container inspect mysqlsec

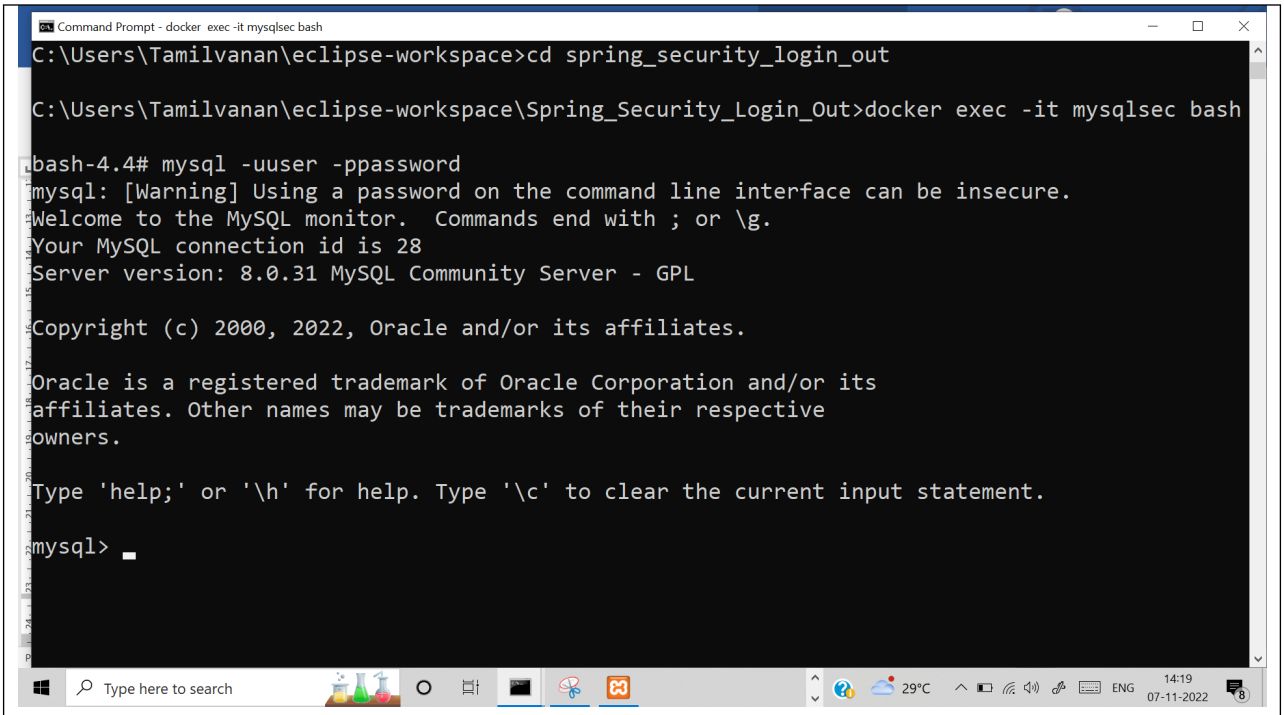
Load your springboot application in docker container

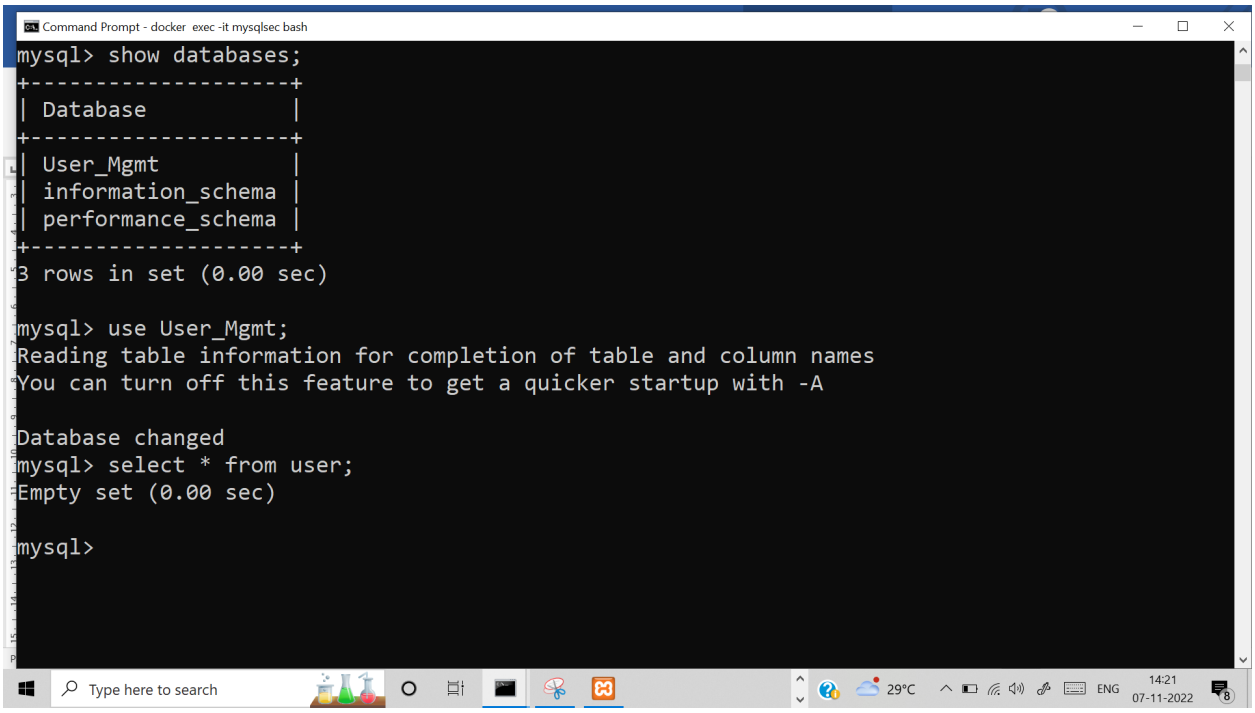
>>docker run -p 9090:8086 --name secapp --net spring-net -e MYSQL_USER=user -e MYSQL_PASSWORD=password -e MYSQL_HOST=mysqlsec -e MYSQL_PORT=3306 secapp

Your application starts running now and Open your browser and type

localhost:9090

To access mysql with docker

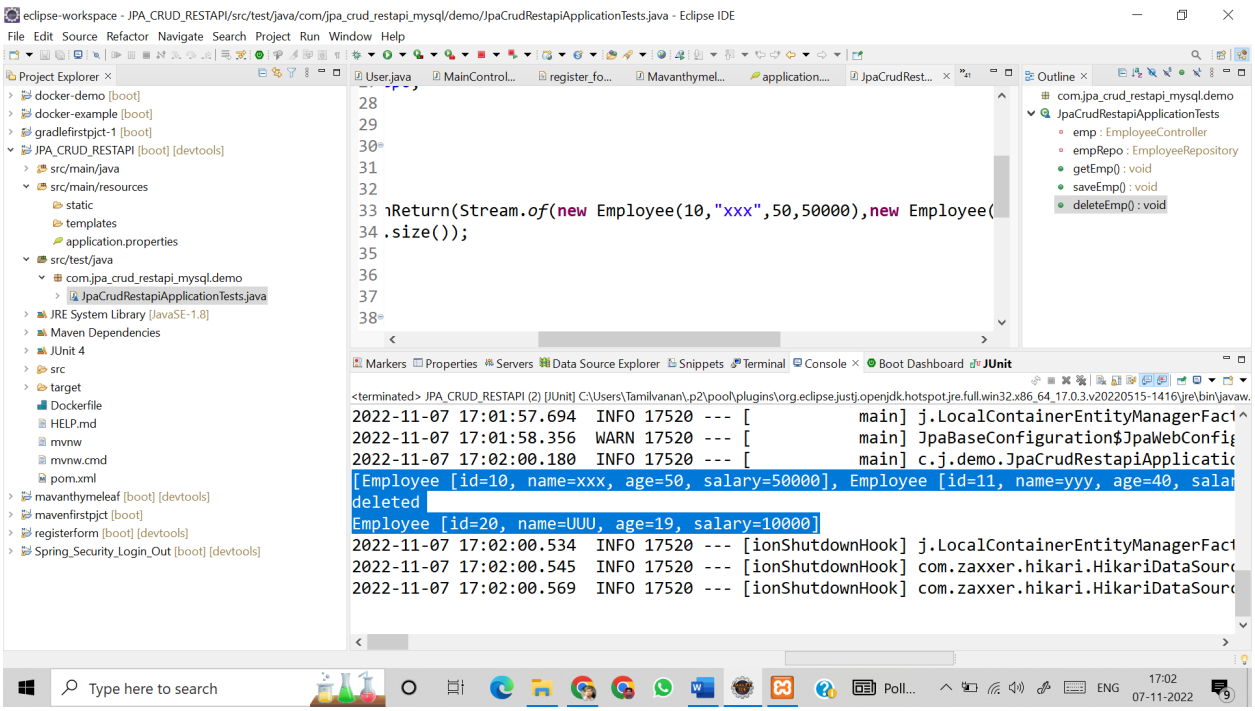

Command Prompt - docker exec -it mysqlsec bash
C:\Users\Tamilvanan\eclipse-workspace>cd spring_security_login_out
C:\Users\Tamilvanan\eclipse-workspace\Spring_Security_Login_Out>docker exec -it mysqlsec bash
bash-4.4# mysql -uuser -ppassword
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.31 MySQL Community Server - GPL
Copyright (c) 2000, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
Type here to search
29°C
ENG
14:19
07-11-2022


Command Prompt - docker exec -it mysqlsec bash
mysql> show databases;
+--------------------+
| Database |
+--------------------+
| User_Mgmt |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)
mysql> use User_Mgmt;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> select * from user;
Empty set (0.00 sec)
mysql>
Type here to search
29°C
ENG
14:21
07-11-2022

SAMPLE PROGRAM

OUTPUT

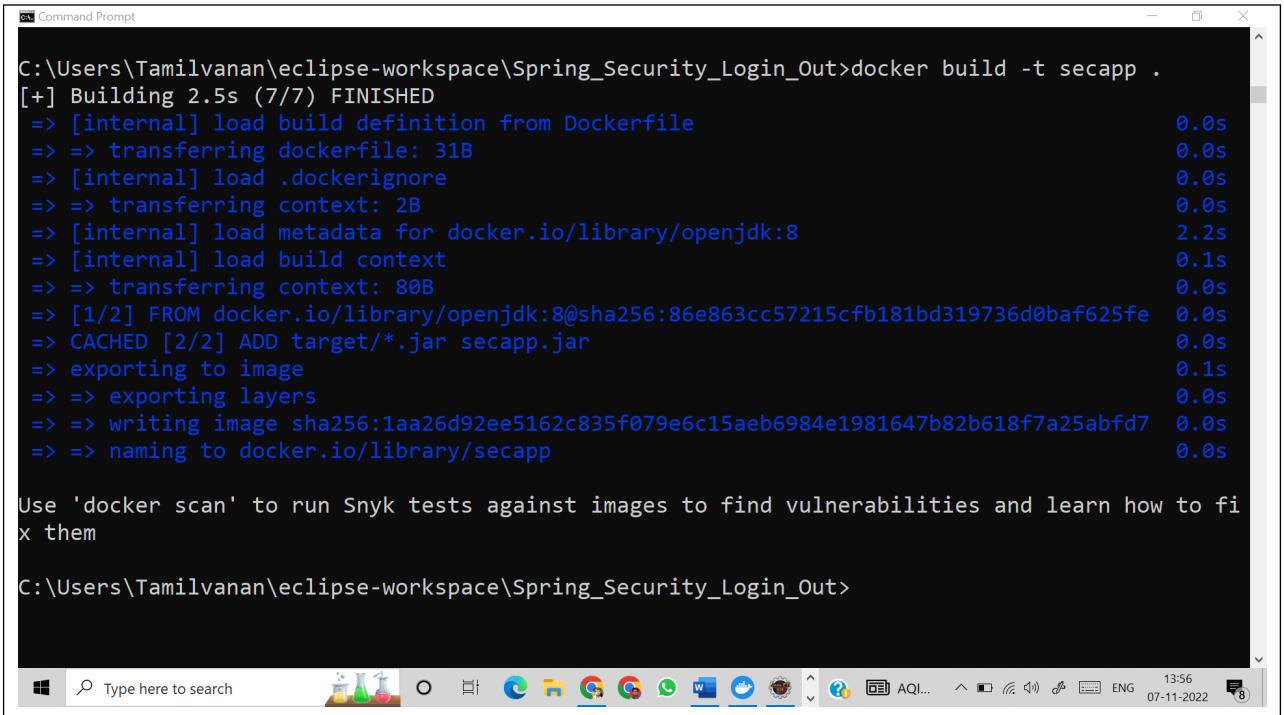
Junit Test :



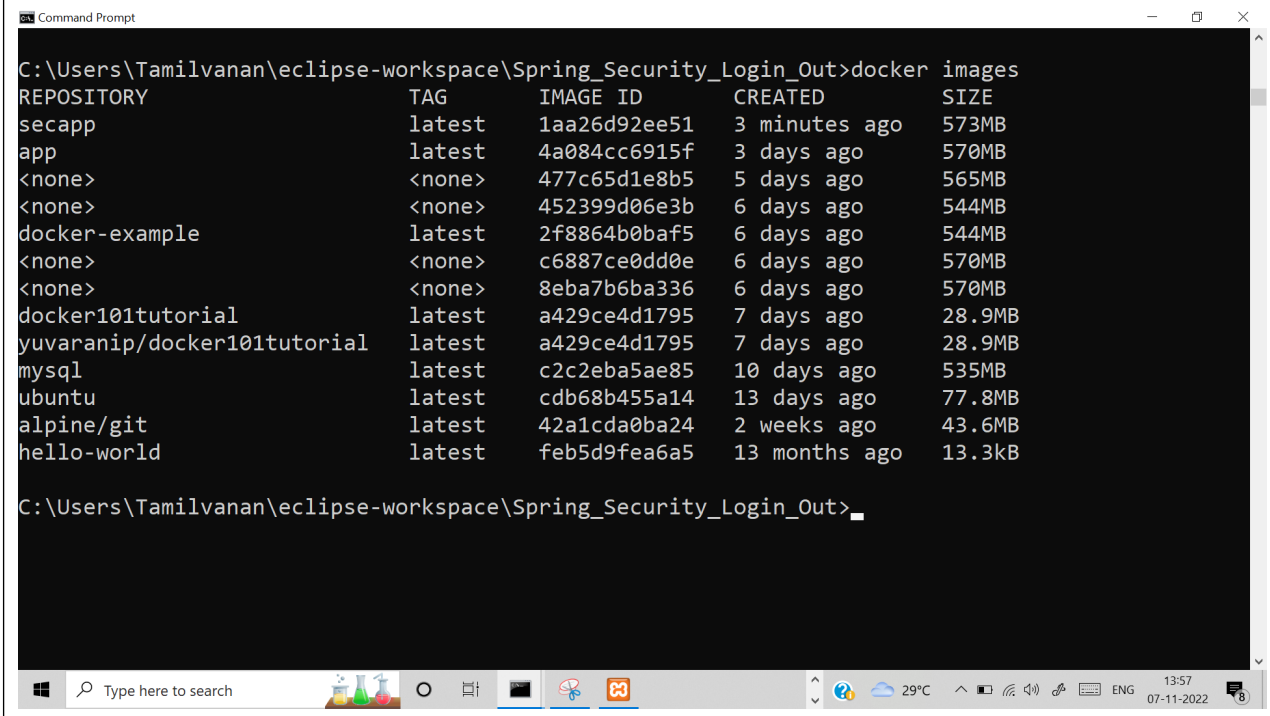
Create a Docker image file for your application and mysql.

>>docker pull mysql

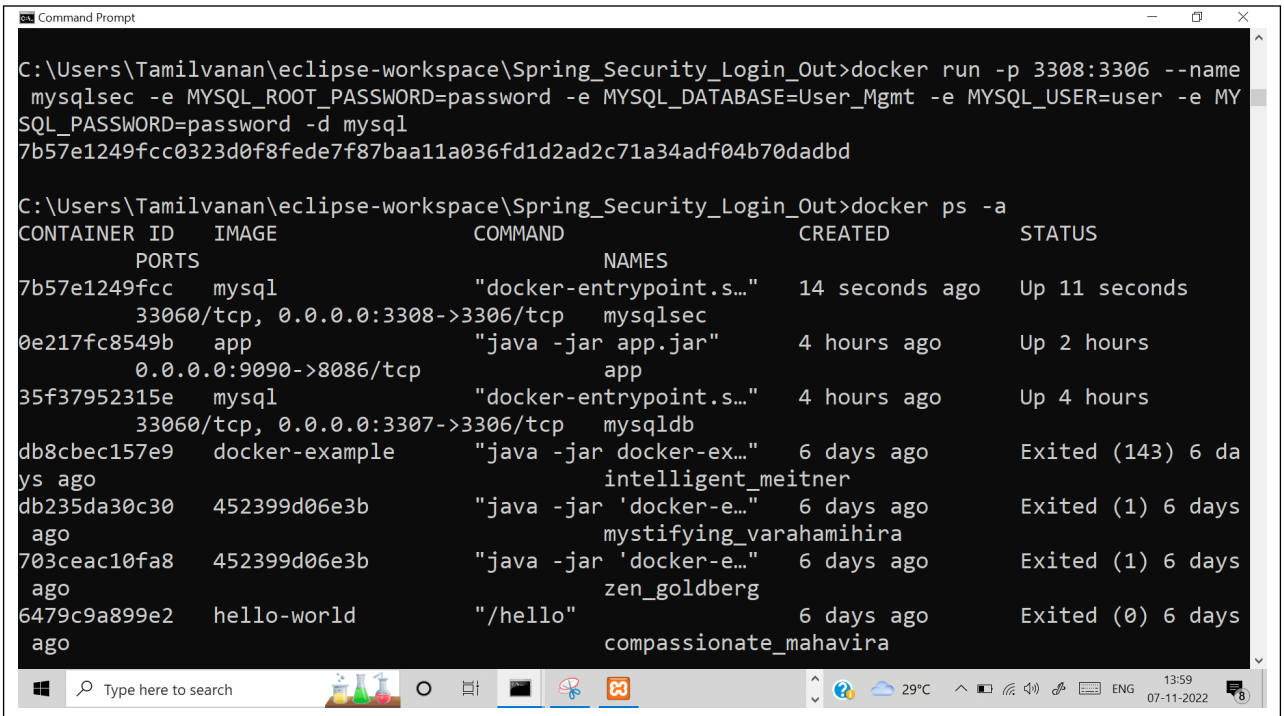
and



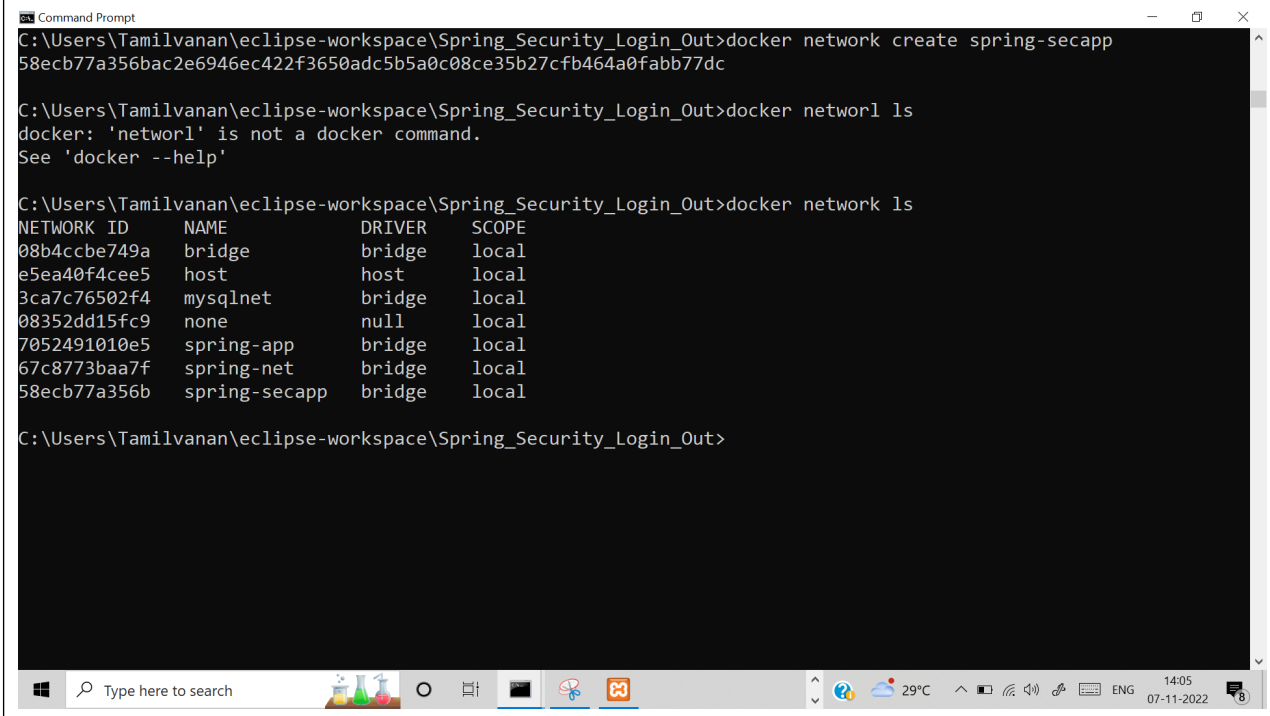
check the created image by using the "**docker images**" command

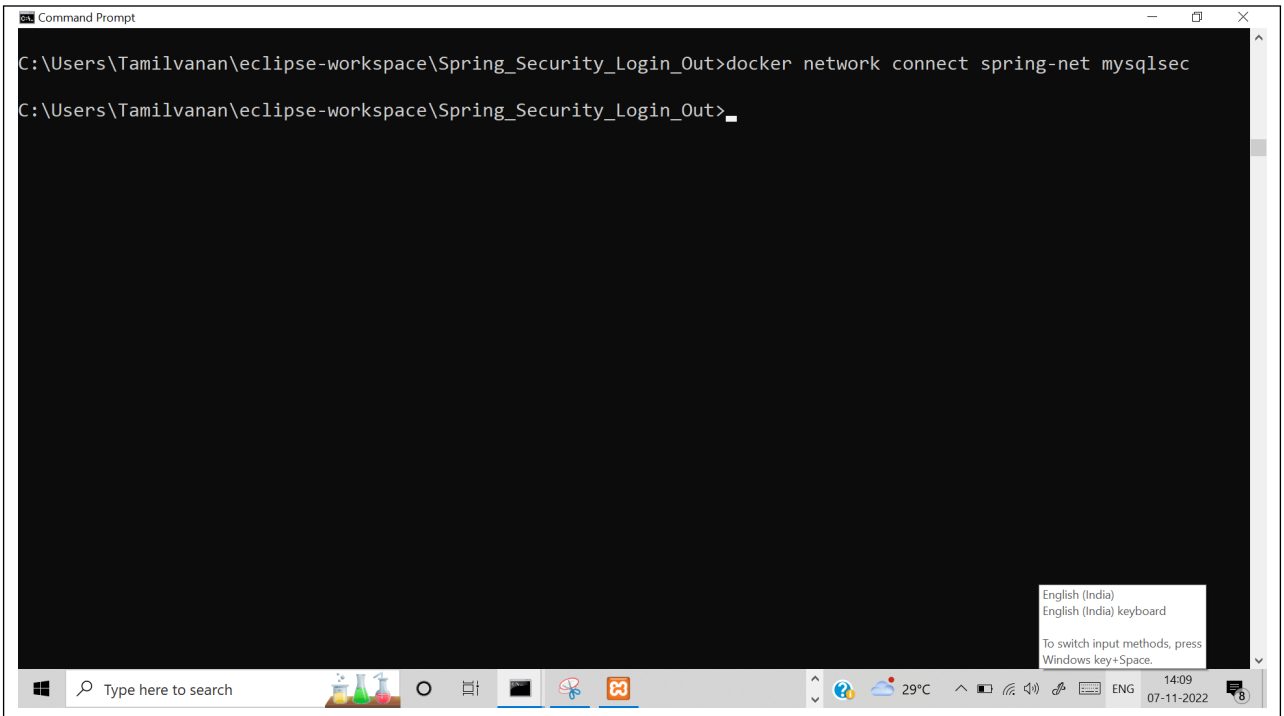


Load mysql image into container

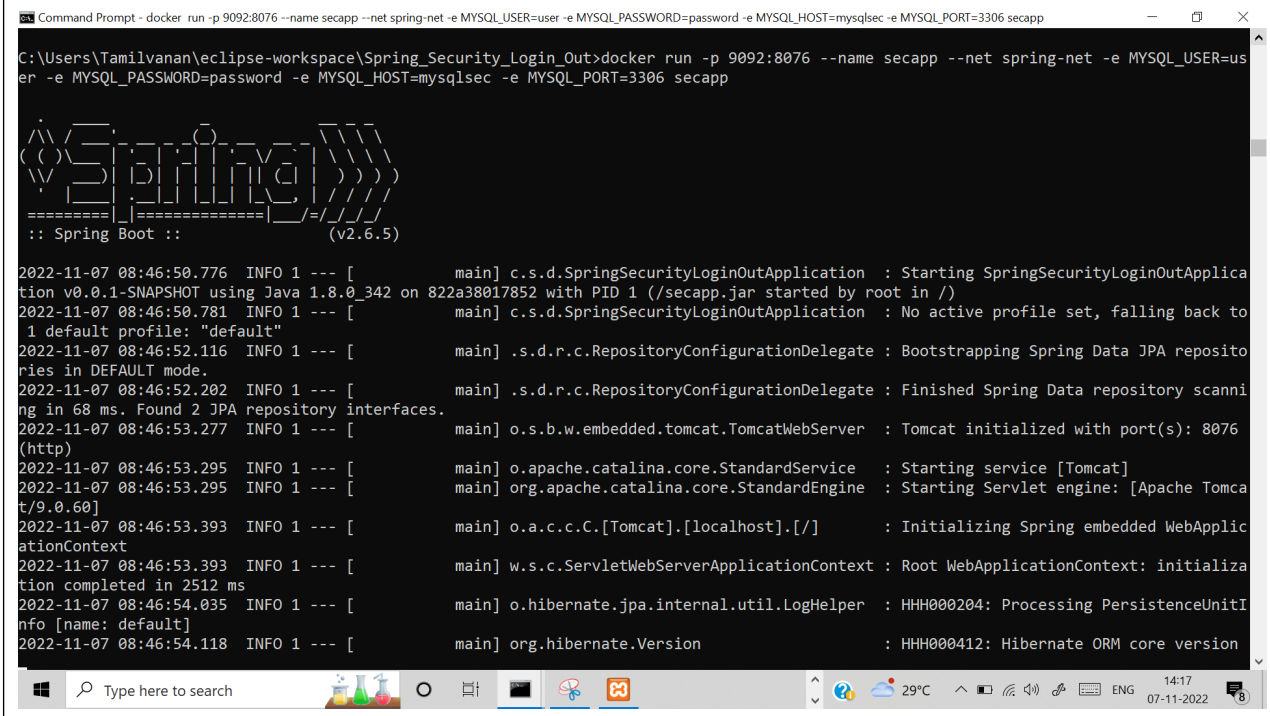


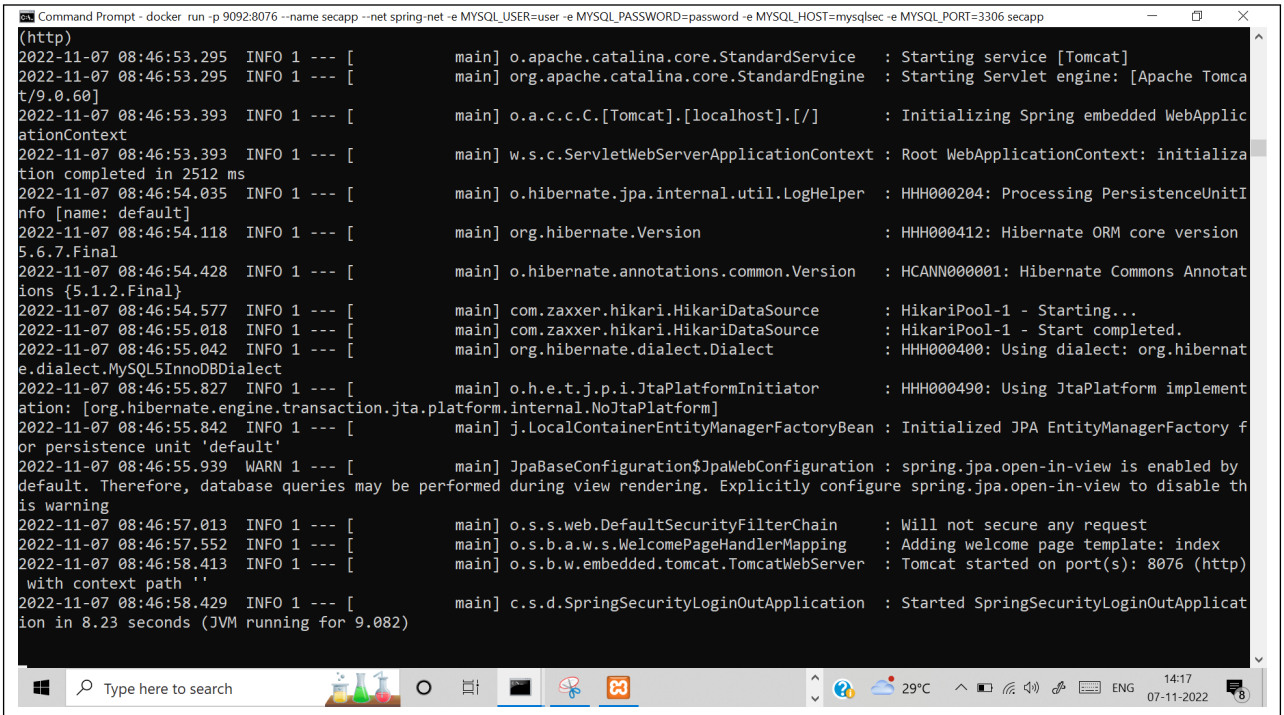
Create a network to connect the mysql with springboot application





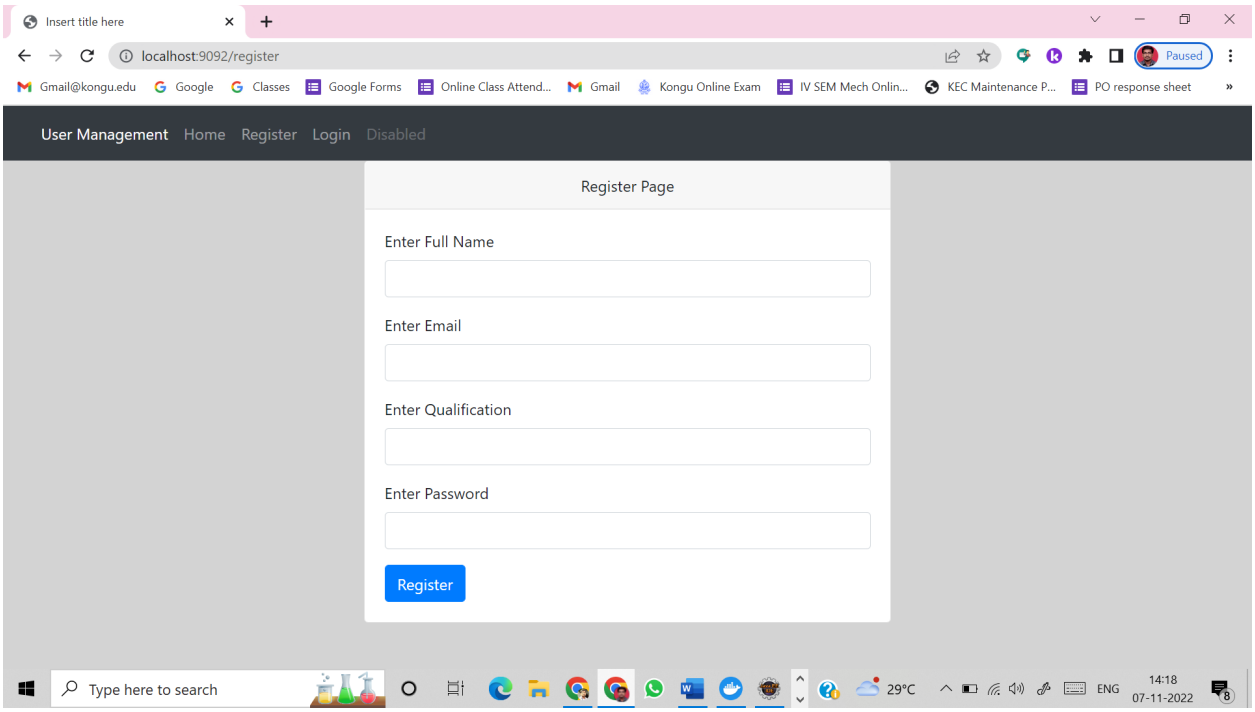
Load your springboot application in docker container





Your application starts running now from docker and Open your browser and type

localhost:9092



RESULT