**German International University**
**Faculty of Engineering**
**Department of Media Engineering and Technology**

**Architecture of Massively Scalable Applications**, Spring 2025
**Lab Manual 2**

Docker and Docker Compose

In this Lab, we will be working with docker and docker compose to dockerize the application we built last lab and scale it by a factor of 3.

## 1 Docker Prerequisites

To be able to follow with this lab, you need to have docker installed, please follow the document on CMS to install docker on your device. When running docker, you need to have docker desktop open (unless running on linux)

a) Make sure docker is running by trying the following command

```
docker --version
```

this should return your docker version

b) Pull docker image using

```
docker pull openjdk:25-ea-4-jdk-oraclelinux9
```

## 2 CRUD Project

If you have the last lab code on your laptop, you can ignore step a.

a) Clone the project from github repo: `git clone https://github.com/Scalable2025/Lab1`

b) Run the project in IntelliJ to make sure everything is running

c) Visit `http://localhost:8080/users/cf82b98f-3438-4696-8316-c8f6e4010f36` and make sure that the use *Martha Downing* is the user returned

d) Run the test case created last week to make sure everything is working fine.

## 3 Environment Variable

a) Create a new controller called `WelcomeController` in the controllers folder

b) You need to be able to access this controller via `/welcome` route

c) Create a method called welcome and let it return a string saying **Hello <your_name>**, your name will be added as an environment variable called **NAME**

## 4 Prepare your project

a) Before building the project we need to make a slight change in the code to make work correctly after building, we need to change the `findAll()` method in the repository class to be the following:

```
public List<User> findAll() {
    try {
        ObjectMapper objectMapper = new ObjectMapper();
        ClassPathResource resource = new ClassPathResource("users.json");
        InputStream inputStream = resource.getInputStream();
        List<User> users = objectMapper.readValue(inputStream,
        new TypeReference<List<User>>() {});
        return users;
    } catch (IOException e) {
        throw new ResponseStatusException(HttpStatus.NOT_FOUND,
        "File not found: " + e.getMessage());
    }
}
```

this is to make spring boot find the users.json file correctly after building.

b) Ensure all packages are installed using: `mvn clean install`

c) Build your project using `mvn package -DskipTests`

d) Open a terminal/cmd session inside the project root folder and run
`java -jar -DName=your_name target/<your_build_name>.jar`

e) Visit localhost:8080/users to verify all is working as it should be.

f) Visit localhost:8080/welcome to verify that your name is displayed correctly

## 5 Dockerfile

a) Create `Dockerfile` in the root folder of your project with the necessary commands to perform the following:

1. Base Image (FROM)
2. Define working directory (WORKDIR)
3. Copy the build file from the target folder (COPY)
4. Make sure also to add all the content of the target folder to ensure `users.json` is copied as well, you can use this command `COPY target/ target/`
5. Add ENV variable called NAME and set it to you **Docker_<your_name>**
6. Expose the necessary port (EXPOSE)
7. Add the command to run the project (now as target/app.jar) (ENTRYPOINT)

b) Build your Dockerfile to be an image called `lab2-img`

c) Run your image to be a container in detached called `lab2` and map the running port to port 9090

d) Verify your app is running by visiting `localhost:9090/users`

## 6 Docker Compose

a) Create `docker-compose.yaml` next to the Dockerfile created in the previous step.

b) Add your application as a service to the docker-compose file with the following features:

   1. 10 Ports: 4040-4049 mapped to the inside container 8080
   2. Reassign the env variable NAME to: Docker Compose Your_Name, and check which env will be used

c) Run your docker-compose and scale your app to have 5 instances

d) Verify your app is running by visiting `localhost:4040/welcome`

e) Verify all instances are running correctly by visiting all 5 instances' welcome URLs

## Solution

You can find the full solution on `https://github.com/Scalable2025/Lab2`, all commands used will be in the README section.