Task 3: Docker

Docs:

1. Read documentation about docker (https://docs.docker.com/)

Tasks:

1. Install docker. (Hint: please use VMs or Clouds for this.)
   EXTRA 1.1. Write bash script for installing Docker.

2. Find, download and run any docker container "hello world". (Learn commands and parameters to create/run docker containers.
   EXTRA 2.1. Use image with html page, edit html page and paste text: <Username> 2024

3.1. Create your Dockerfile for building a docker image. Your docker image should run a Spring Boot web application with few simple GET endpoints. Web application should be located inside the docker image.
   EXTRA 3.1.1. For creating the docker image use clear basic images (ubuntu, centos, alpine, etc.)

3.2. Add an environment variable "DEVOPS=<username> to your docker image
   Print this environment variable's value in one of your GET endpoints

4. Push your docker image to docker hub (https://hub.docker.com/). Create any description for your Docker image.
   EXTRA 4.1. Integrate your docker image and your github repository. Create an automatic deployment for each push. (The Deployment can be in the "Pending" status for 10-20 minutes. This is normal).

5. Create docker-compose file. Deploy a few docker containers via one docker-compose file.

- first image - docker image from step 2. 5 nodes of the first image should be run;
- second image - your Spring Boot application;
- last image - any database image (mysql, postgresql, mongo or etc.). The database should contain a simple table with some sample data.
  Second container should be run right after a successful run of a database container.
  EXTRA 5.1. One of the endpoints of the second container should retrieve data from the DB table.
  EXTRA 5.2. Use env files to configure each service.

The task results are the dockerfile/docker-compose files in your GitHub repository. Also please put your Spring Boot app source code to the same repo.