DOC1 Course Assignment

Contents

Purpose **1**

The CA's role in the exam 1

The VIA Tabloid Application - VIATAB **2**

Part 0 – Creating a GitHub project for the CA 3

Part 1 – Docker Introduction 3

Part 2 – Running the whole App with Docker Compose 3

Part 3 – Deploy the App on Kubernetes (Locally) 3

Part 4 – Continuous Integration (CI) and Continuous Deployment (CD) Pipeline 3

Purpose

The purpose of the course assignment (CA) is to provide an opportunity for all students to work with the most important tools and methods used in the course. Depending on circumstances, the semester project may not provide everybody with a chance of working with all this stuff. The CA is a small running project each student runs by themselves, but it's okay to work with others as long as everybody works with all the tools and methods.

There are no deadlines and hand ins for the CA, the students are expected to do the CA themselves as a small ongoing activity. But you can obviously get help and support for the CA from the instructor.

The CA's role in the exam

The exam for DOC1 is a 20-minute oral examination with a set of pre-published questions. At the exam you will draw a question and be asked to explain a single core tool or method. After that, and this is the important part, you will be asked to talk about (and show some concrete examples) how you have used the tool/method. If you have used the tool/method in the SEP4 project that's fine, and you can use that to talk about/exemplify the ways in which you have used the tool/method.

But the SEP4 may not provide all team members with equal opportunities to work with all the tools and methods from DOC1. And this is where the CA comes into play – if you have worked with the tool/method in the SEP4 project we will then use the CA as platform for

talking about and see examples of your use of the tool/method from the question you drew to exam.

You should be aware that if you have not used the tool/method from the exam question in the SEP4 project, nor done so in the CA, you risk:

1) Being shown some arbitrary examples and asked to explain them.
2) Being asked about other stuff (tools/methods) in addition to the one from the question you drew.

The reason for that is that the exam provides an appr. 15 minutes of effective QA time for each student, and since explaining a single tool/method on a general level only takes a couple of minutes, we need to find other stuff to talk about for the remainder of the exam session. And if you have some examples to show and talk about (from SEP4 or the CA), that usually suffices, but if you don't, we are forced to go outside the question and talk about some additional stuff which can then be *any* DOC1 related subject.

The VIA Tabloid Application - VIATAB

The main focus of the course assignment is on building a VIA tabloid application that adds sensational stories, read, update and deletes stories, from the different departments here in Horsens following DevOps practices.

In order to make the design and technology decisions in the course more concrete, it is expected that you will implement an example application called the VIA Tabloid application. This application will be implemented incrementally throughout the course.

The VIA-Tabloid application is a web application that displays captions of sensational stories from the different departments (at least 3) in VIA. The department where the respective stories/ideas is displayed could be placed on the page using different components or on different tabs.

The application should have at least three components:

1. Frontend (React, TypeScript)
2. Backend (Java Spring Boot or C# .NET)
3. Database (Postgres or MySQL, MongoDB)

A working frontend is required for this app, and should implement the features for your tabloid app; add story item, delete item, etc. It does not need to be especially pretty but there needs to be a frontend that connects with a backend, server, etc. You can use React or any frontend framework you like, but again it must have a real connection to your backend. We will not be particular on how the frontend looks, rather the use of DevOps throughout the assignment.

Part 0 – Creating a GitHub project for the CA

Each student needs to create a personal GitHub repository for the CA project. Both because that's necessary to work with some tools/methods and because such a repository is a convenient place to maintain the project.

WHAT YOU NEED TO DO:

Configure your GitHub CA repository so that GitHub enforced the use of pull requests to merge any new code into the CA code base.

Part 1 – Docker Introduction

Like all programming problems, learning a new technology is not an exercise in reading but rather an exercise in thinking and coding. This part 1 is designed to give you an opportunity to try hands-on experience in some fundamental skills involved in Docker containerization.

WHAT YOU NEED TO DO:

You will practice creating Docker images and containers

1. Get React frontend working in a Docker Container
2. Get Spring boot backend working in a Docker Container
3. Get PostgreSQL working in a Docker Container

Part 2 – Running the whole App with Docker Compose

With docker-compose, we can describe our build and run instructions in a file and do it for more than one container at a time.

WHAT YOU NEED TO DO:

You will create docker-compose yaml file that will orchestrate the frontend, backend as well as the database using container images.

Part 3 – Deploy the App on Kubernetes (Locally)

Create Kubernetes resources and deploy the whole application in Minikube on your own PC.

WHAT YOU NEED TO DO:

You will create Kubernetes configuration/resource files that will orchestrate the front and backend as well as the database using container images. The whole thing should run on Minikube.

Part 4 – Continuous Integration (CI) and Continuous Deployment (CD) Pipeline

Create CI/CD pipelines for the whole project.

WHAT YOU NEED TO DO:

Configure your GitHub CA repository so that one or more GitHub actions are triggered whenever you add new code to the code base using a pull request. These actions should at a minimum verify that the code can be built and that all tests run successfully.