Project Proposal - Online course material CI/CD tool for A+: Godfather

1. Background

A+ is interoperable and extendable Learning Management System that uses different services to provide unlimited selection of interactive e-learning material types. Today, several web services and build tools are included in the A+ software family. A+ is extensively used in Aalto University and Tampere University of Technology. The software is used by thousands of students and by around thirty courses yearly. Most of the courses are from computer science, but there is one from the language center too.

A+ environment has two connected sides. First, the course material creation, management and production. Second, the learner side, where course material is read and exercises are completed. The course material is typically written using rich text (RST) and versioned using git. Material is automatically compiled, when it changes, and rendered version is pushed to production for learners to use it.

2. The Problem

To build a fully featured course, multiple backend services are needed. All of these services require course specific configuration, which needs to be updated when the course material changes. Thus, the system requires constant upkeep to stay consistent and error free. Furthermore, current tools don’t provide information about the state of services to the course staff.

2. Project goals

The goal of the project is to build a production ready web service, that handles course material building, deployment and backend service configuration. In addition, there should be sufficient views for course staff to check the state of the deployment. The role of this new service can be described as continuous integration, continuous deployment and configuration management for a course material. The name for this new service is godfather, which should describe its role within A+ software family pretty well.
Currently, we have prototype that does a bit of the course build automation. However, it does not do much. It works as a good proof of concept, but isn’t a production capable tool.

3. Technologies

The service will be built using Django Web Framework and celery as an IPC (inter-process communication). Communication to other services will be done over REST APIs and the network filesystem. Course building, local development and future deployment do use docker containers for a build environment definition and containment. We already have a partially complete library for course material building, which will require some work to fulfill the project goal.

The selection of tools is based on service performance requirements and existing libraries. In addition, most other services share the technology and thus future developers can work on all of them with ease.

4. Requirements for the students

The students have to be familiar with Python. The group should have at least a few members that have completed a web software development course already, but rest could do it during this project for example.

Expertise or familiarity with the django framework, inter-process communication, concurrent programming, databases, websockets, devops or docker containers would help, but is not mandatory. After the project, you will be familiar with some of them.
This project requires a bit of ambition to learn new things and to work with new libraries and software patterns. The group will get to do more challenging tasks, if they have a bit more prior knowledge (see above). But if the group doesn’t, then they will learn a lot and the project will be heavier on understanding than implementing. On the other hand, the architecture, overall design and tools are already planned and you will be working together with the A+ lead developer so you will not get stuck on how to solve problems.

5. Legal Issues

All project code for the service and related tools are published in github.com under MIT license. In other words, all your work will be available for your CV.

6. Client

The client is Aalto Learning + Technology Research Group as it is the current director of A+ development. The product owner and technical specialist is available at the CS building, so it’s easy to organize meetings and ask for help.

Client representative:
- Product Owner and technical specialist
- A+ lead developer
- Jaakko Kantojärvi
- jaakko.kantojarvi@aalto.fi

7. Additional information

The project group will join A+’s development Slack for communication. Public documentation should be written in English (github repository).