Faculty of Informatics
Masaryk University

PA181
Services - Systems, Modeling and Execution

Term Project Documentation

Adrián Tóth (491322)

Jiří Čechák (445717)

Jan Ondruch (433341)

Tadeáš Pavlík (487555)

Václav Stehlík (487580)

April 24, 2019

Contents

1 About

Term project for course *PA181 Services - Systems, Modeling and Execution*[1] in year 2019. Within the project, we had to create a fully functional application using the *IBM Cloud*[2] technology including a detailed documentation and a presentation. *doc. Mouzhi Ge, Ph.D.*[3] is the project supervisor.

2 Idea

The core idea was to create and develop a useful and practical application. The application provide services for testing the users in a form of questions and answers. Users are able to test themselves via these questions by selecting the correct answers. There are severals tests in three different types of language (Czech, Slovak and English).

3 Used Technologies

The following technologies were integrated and used during the development process:

- cloud based platform
    - IBM Cloud
- version control system (VCS)
    - GitHub[4]
- continuous integration (CI)
    - Travis CI[5]

Besides the used technologies we were using additional available tools such as:

- IBM Cloud DevOps Toolchain
    - Set or combination of tools that build and deploy the software in a repeatable way with minimal human intervention.
- React
    - A JavaScript library for building user interfaces.
- ASP.NET
    - A framework for building web apps and services with .NET and C#.

---

[1]is.muni.cz/predmet/fi/jaro2019/PA181
[2]cloud.ibm.com
[3]is.muni.cz/auth/osoba/239833
[4]github.com
[5]travis-ci.org

4 Work Division

Our team consisted of 5 members: Adrián Tóth, Jan Ondruch, Jiří Čechák, Tadeáš Pavlík and Václav Stehlík.

Everyone from us was in charge of a certain part of the project. The work was divided as the following:

- Adrián Tóth
  - project initialization
  - VCS initialization
  - creation of project skeleton
  - Travis CI integration and configuration
  - IBM DevOps toolchain configuration
  - creation of continuous delivery pipeline
  - project deployment
  - bug fixing
  - documentation

- Jan Ondruch
  - user research
  - specifications
  - application design (idea)
  - testing (user interface)
  - testing (usability)

- Jiří Čechák
  - application design (user interface)
  - application user interface initialization (framework, layout, views)
  - application frontend
  - testing (user interface)
  - bug fixing

- Tadeáš Pavlík
  - analysis
  - application design (concept)
  - application design (user interface)
  - application backend **TODO**
  - testing (blackbox)

- Václav Stehlík
  - application design (concept)
  - application backend
  - backend and frontend interconnection
  - troubleshooting
  - bug fixing

5 References

Domain names of the deployed application are:

- pa181.eu-de.mybluemix.net

- pa181.eu-de.cf.appdomain.cloud

Source code of the application can be found at:

- github.com/europ/MUNI-FI-PA181/tree/master/src

Setup guide for the application can be found at:

- github.com/europ/MUNI-FI-PA181/wiki/Setup

Documentation of the application can be found at:

- github.com/europ/MUNI-FI-PA181/blob/master/doc/doc.pdf

Presentation of the application can be found at:

- github.com/europ/MUNI-FI-PA181/blob/master/pres/pres.pdf

6 Application initialization

Firstly, we have chosen a stable, reliable and safe platform supporting team project development - GitHub. Furthermore, GitHub provides a version control system management and the integration with IBM Cloud is supported. Subsequently, after the repository was configured properly, we had to choose the technologies. We decided to use #C (general-purpose, multi-paradigm and object oriented programming language) and React (library for building user interfaces) for the project. Based on the above, we have initialized the project skeleton - a *'Hello, World!'* application.

7 Application implementation

The project was implemented via *ASP.NET* framework using a Model–view–controller as an architectural pattern. The core UI was implemented in JavaScript using React library. This UI was later interconnected with backend. Our application supports an application log too to find easier the fatal failures. Our application was divided into few parts that communicate together and process requests - API, entities, repositories and services. The requests are firstly processed in the API, more precisely endpoint in controller. To process this request, the required actions are delegated to services (the business logic specific for our application is implemented here) that delegates the related subrequests to repository that provides the database communication using wrappers. **TODO**

8 Application deployment

The deployment was configured before the core implementation itself. Deployment was initialized and configured on the already set *'Hello, World!'* application. We wanted to create a fully automated delivery with zero intervention required, which allowed us to concentrate and focus on code only instead of problematic deployment. We were using a continuous delivery approach - automated build and deploy method via IBM's delivery pipeline. The delivery pipeline was configured to automatically install all required dependencies, build and deploy the application. The development process required to just commit the changes into the version control repository.

Whole deployment is based on one own properly configured DevOps toolchain. The toolchain is shown in Figure 1, which includes GitHub[6], Travis CI[7] (as other tool) and Delivery Pipeline[8].
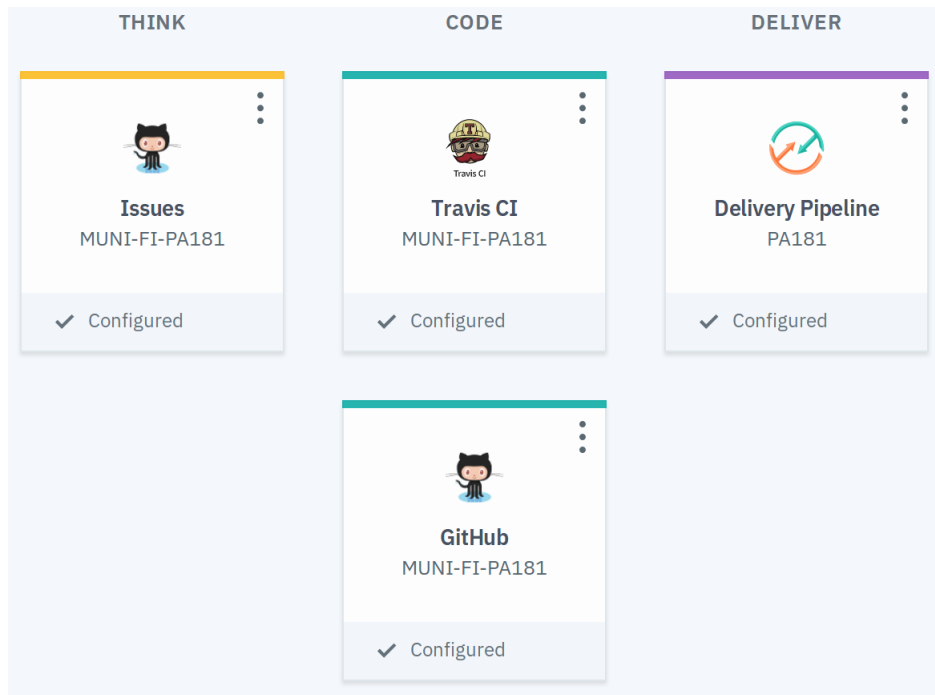

Figure 1: Toolchain

The toolchain providing a continuous delivery service allows us to focus on code only. After the continuous delivery toolchain was set properly, we just had to commit the changes to the repository and the application was immediately built and deployed.

9 Application instructions

TODO: how to use, tutorial, examples

10 Screenshots

TODO: images (add it to 'doc/img' folder - vector would be the best or raster with high resolution)

11 Development issues

During the project development we have faced a few problems that have been reported:

- github.com/IBM-Bluemix-Docs/ContinuousDelivery/issues/13

- github.com/IBM-Cloud/aspnet-core-helloworld/issues/39

---

[6]console.bluemix.net/docs/services/ContinuousDelivery/toolchains_integrations.html#github
[7]console.bluemix.net/docs/services/ContinuousDelivery/toolchains_integrations.html#othertool
[8]console.bluemix.net/docs/services/ContinuousDelivery/toolchains_integrations.html#deliverypipeline