DevOps projectreport

Group J - Pythonkindergarten

Joachim Køcher Kelsen (jokk@itu.dk)
Victor Nordestgaard (vino@itu.dk)
Isabella Drest Rasmussen (iras@itu.dk)
Anne Siemkowicz (asie@itu.dk)
Bjørnar Haugstad Jåtten(bjja@itu.dk)

19 May 2021

Contents

1 Introduction

2 Lessons Learned Perspective

Describe the biggest issues, how you solved them, and which are major lessons learned with regards to:

- Evolution and refactoring
- Operation
- Maintenance
- Of your ITU-MiniTwit systems. Link back to respective commit messages, issues, tickets, etc. to illustrate these.
- Also reflect and describe what was the "DevOps" style of your work. For example, what did you do differently to previous development projects and how did it work?

3 Process' perspective

Interactions as developers
How do you interact as developers?

Team organization
How is the team organized?

- A complete description of stages and tools included in the CI/CD chains.(That is, including deployment and release of your systems.)
- Organization of your repositor(ies).(That is, either the structure of of mono-repository or organization of artifacts across repositories. −¿ In essence, it has to be be clear what is stored where and why.)
- Applied branching strategy.
- Applied development process and tools supporting it (For example, how did you use issues, Kanban boards, etc. to organize open tasks)
- How do you monitor your systems and what precisely do you monitor?
- What do you log in your systems and how do you aggregate logs?
- Brief results of the security assessment.
- Applied strategy for scaling and load balancing.
- In essence it has to be clear how code or other artifacts come from idea into the running system and everything that happens on the way.

4 System's Perspective

- Design of your ITU-MiniTwit systems.
- Architecture of your ITU-MiniTwit systems.
- All dependencies of your ITU-MiniTwit systems on all levels of abstraction and development stages. (That is, list and briefly describe all technologies and tools you applied and depend on.)
- Important interactions of subsystems.
- Describe the current state of your systems, for example using results of static analysis and quality assessment systems. We use two different code analysis tools. They are both run in our CI pipeline.
- Finally, describe briefly, if the license that you have chosen for your project is actually compatible with the licenses of all your direct dependencies.