DevOps Report

Individual Project: FestivalConnect

Name: Lucas Jacobs
Class: S-A-RB06
PCN: 490692
Student number: 4607368
Technical teachers: Felipe Ebert, Bartosz Paszkowski
Semester coach: Gerard Elbers

Table of Contents

Introduction

Within this document, the development cycle will be automated and made as maintainable as possible. It is important to state how DevSecOps is giving value to this project and how this will be guaranteed.

DevOps

Development and operations (DevOps) is about breaking down the the struggle between software development and IT operation. DevOps is not only around using certain tools or following certain practices but also the mindset and culture shift. With DevOps we instead want teams to collaborate closely with each other by following a software development lifecycle. Furthermore, by then automating tasks and improving communication, DevOps provides a process that is smooth, fast, and efficient. (DevOps, n.d.)

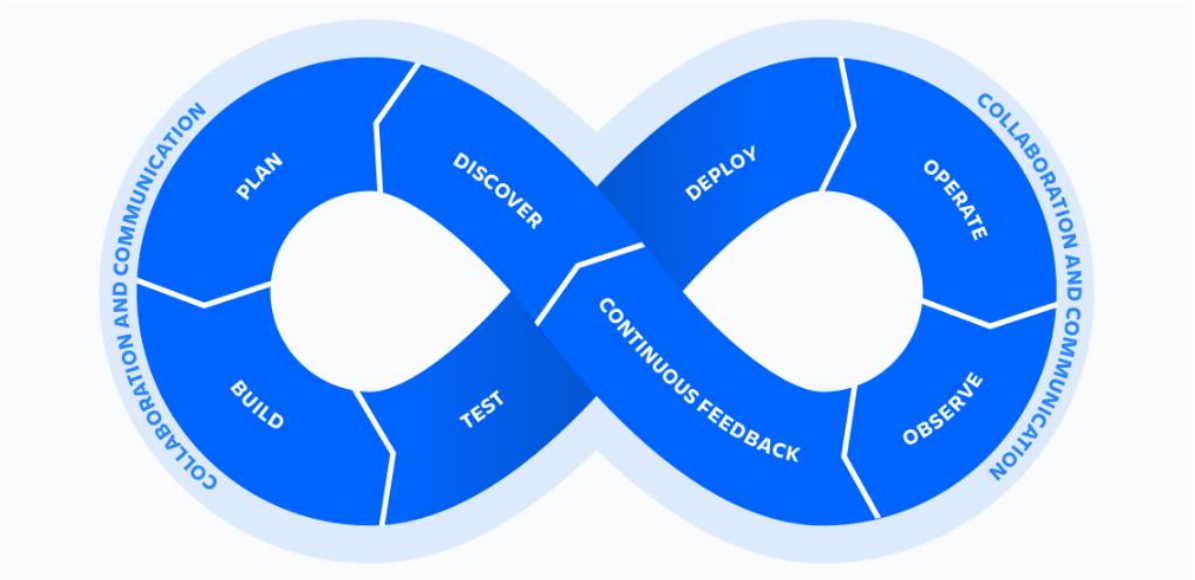

Figure 1: DevOps LifeCycle (DevOps, n.d.)

The DeOps figure represents the need for constant collaboration and an iterative approach in the software development lifecycle. (DevOps, n.d.)

The Benefits of DevOps

Improved Quality

With DevOps, the main focus is to continuously deliver on quality and reliability. With automated testing, where CI/CD is quite important, it will make sure that new code changes are tested and prevent new code bugs. This will lead to a more stable and reliable application.

Reduced Risks

DevOps will make sure that there are smaller releases and that security will be integrated and thought about during the whole process. In case of issues, in a smaller scope when changes are made, it is easier to identify the root cause and make changes.

Monitoring and feedback loops

With the use of DevOps, monitoring and feedback mechanisms become a key point. Developers will get more knowledge about the code, which will lead to optimized performance, reduced downtime, and faster issue resolutions.

(Rini, 2023)

DevOps Best Practices

For DevOps, we can apply several methods to manage the project.

Agile Project Management and Workflow

With the use of methodologies of for example Scrum, the management of projects becomes far easier. The iterative approach with breaking down project tasks, while also having continuous feedback loops to adapt and change requirements and deliver value constantly. Furthermore, the clear workflow with phases of to-do, in-progress, code review, and done, enables the team to track progress effectively and respond to feedback.

CI/CD and Testing Automation

Integration continuous integration and continuous deployment (CD/CD) pipelines to automate several stages such as building, testing, analyzing, and deploying processing. Integrate this with early on testing in the development cycle, to identify and address issues sooner. Moreover, automated tests of different types including unit tests, integration tests, E2E tests, and performance tests will make sure the code is up to quality and reliable.

Tool Selection and Integration

Having tools for each stage in the DevOps lifecycle, with a focus on collaboration, automation, and monitoring capabilities. Selection of tools that are easy to integrate and communicate with between development. Also, version control systems like, Git, platforms like Jenkins, and monitoring tools like Prometheus to improve the smoothness of development and workflows to improve the software quality.

Culture of Collaboration and Continuous Improvement

It is important to have a culture of collaboration that is transparent and is continuously improving across the team. It is important to share knowledge, receive and give feedback, and have overall open communication. To add on, the process is to continuously find improvements and seek a workflow that improves efficiency and effectiveness. This will lead to a more high-quality software product. (HALL, n.d.)

Furthermore, with DevOps, DevSecOps (Development, Security, and Operations) is where we think about security in each stage of the software development process. This is further explained in (Jacobs, 2024, Security Design) to show what stages there are. (What is DevSecOps?, n.d.)

JIRA

To keep track of the tasks that need to be done for each sprint, Jira is being used. The guidelines that are specified in (Jacobs, 2024, User Stories) are how the tasks are divided and how the workflow is being done in the project. This is to keep the agile way of working in a managed place that improves the transparency and insight of the work. It has also some nice features, such as diagrams that can show your progress in each sprint.

Git: Branching Strategy

Repository

FestivalConnect looked into each service that has its repository on GitLab, this brings benefits to the development and makes sure that each microservice is isolated, independent, and it is easier to scale, which is the intention of microservices. When using a mono repository, it simplifies dependency management, streamlines CI/CD, and easier to refactor. This comes with a cost since the complexity grows as the project does, with also the versioning can become hard. Nevertheless, this project is for one semester, and with the trade-offs and being aware of the pros and cons, FestivalConnect will work in one repository. (Sathar, 2023)

Branching Method

In FestivalConnect the usage of the feature branch workflow, means we have a stable main branch, with a development branch where the feature branches will be based.

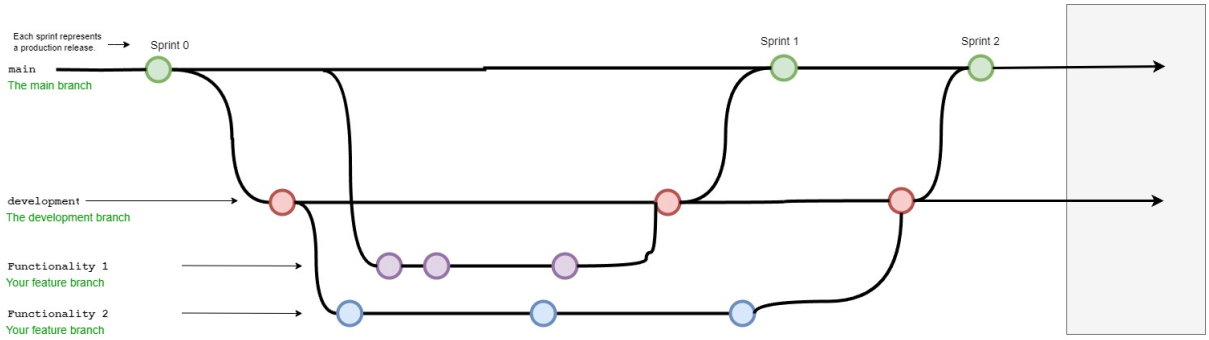

Figure 2: Branching Strategy

CI/CD Pipeline Overview

The DevSecOps problem will be tackled using a CI/CD pipeline, which will make sure the automation of tasks will be defined, within a logical manner. The main flow will include, building, testing, analyzing, and deploying. See the following diagram.

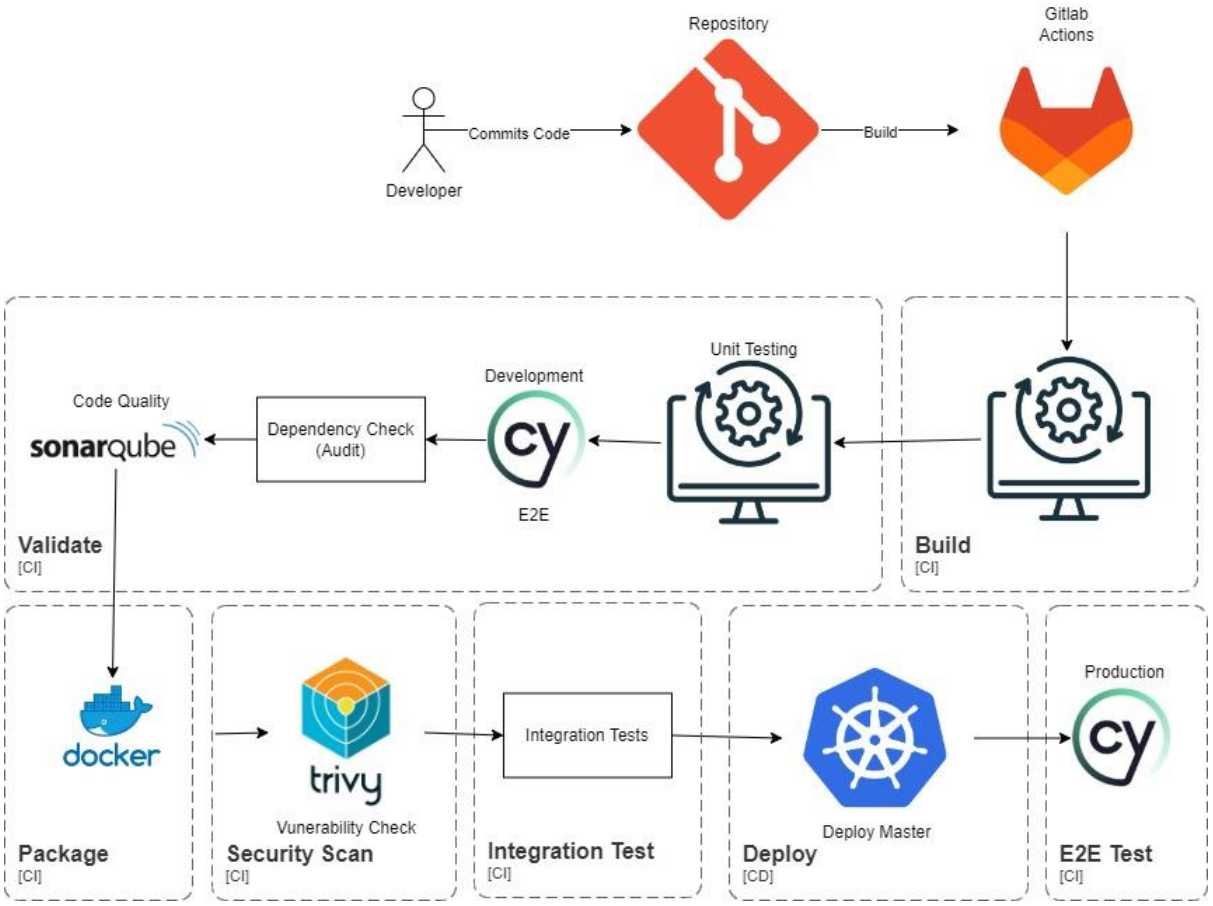

Figure 3: CI/CD Pipeline FestivalConnect

The pipeline can be different depending on each service. Also with the addition of only having one repository, there will be rules for each .yml file that has the configurations for the pipeline that it will only run when there are changes in the service itself. This will make sure that not all the services will be triggered when committing changes to only one service.

References

DevOps. (n.d.). Retrieved from atlassian: https://www.atlassian.com/devops

HALL, T. (n.d.). *DevOps Best Practices*. Retrieved from atlassian: https://www.atlassian.com/devops/what-is-devops/devops-best-practices

Rini, N. (2023, 11 17). *What are the Benefits of DevOps?* Retrieved from techrepublic: https://www.techrepublic.com/article/devops-benefits/

Sathar, A. P. (2023, 10 05). *Choosing Between Mono-repos and Multiple Repositories for Microservices: A Practical Guide*. Retrieved from blog.stackademic: https://blog.stackademic.com/choosing-between-mono-repos-and-multiple-repositories-for-microservices-a-practical-guide-7c90b716e9cb

What is DevSecOps? (n.d.). Retrieved from aws.amazon: https://aws.amazon.com/what-is/devsecops/#:~:text=building%20the%20software.-,What%20does%20DevSecOps%20stand%20for%3F,they%20are%20building%20software%20applications.

Jacobs, L. (2024). Security Design (Unpublished manuscript), FontysICT.
Jacobs, L. (2024). User Stories (Unpublished manuscript), FontysICT.