ITII DevOps Project

Retwis in production

Introduction

Big news, your team is hired by the Retwis company ! Your mission - if you accept it - will be to bring their famous Retwis software to production. They want to be able to deploy, test new features and get feedback as fast as possible so they can adjust their value proposition according to how their software is used.

As a DevOps team, you must put in production the Retwis software using DevOps methodologies. Get the existing code base (either the PHP code base Retwis, or the Java code base Retwis-J) and build the production ecosystem around it, with version control management, automatic CI/CD pipelines, unit and functional testing, security gates, monitoring and alerting.

Requirements

Features

The features of the software itself are listed on the Retwis page.

Technical

The solution must run in production on a Kubernetes cluster.
A commit on the master branch of the source code management system must end-up with a deployment to production following one of these strategies: rolling update, canary, blue/green.

Project methodology

In order to learn fast and therefore tend to quality software. The DevOps project management will use Agile methodologies.

Follow-up: Every day during the project, the teams will regroup with the professor to talk about their progress. Remember: during these 15 minutes time-boxed meeting, it’s not expected to solve issues but to define a proper way to get to solve issues.

Technical baseline

The technical baseline is the mandatory common elements (to all teams) that are required to validate your project. All teams must provide:

- A continuous integration pipeline building, testing and releasing the software
- A continuous delivery pipeline deploying the software to production
- A triggering mechanism allowing to deploy to production with a commit to the master branch

Proposed bootstrap streamed timeline

This is a suggested bootstrap timeline in order to organize your team to get to achieve the baseline as soon as possible.

Common:

Get the knowledge of the application reading and understanding the documentation.
Define roles and responsibilities among the project.
Prepare a project management tool for progress follow-up.

Stream 1 - Source Code:

Get the source code and make it your own, compile, build, package.
Put the source(s) into a team repository in GitHub
Prepare the containers
Deploy locally and test with Docker
Push the containers to registries to make them available for Kubernetes

Stream 2 - Cloud Environment:

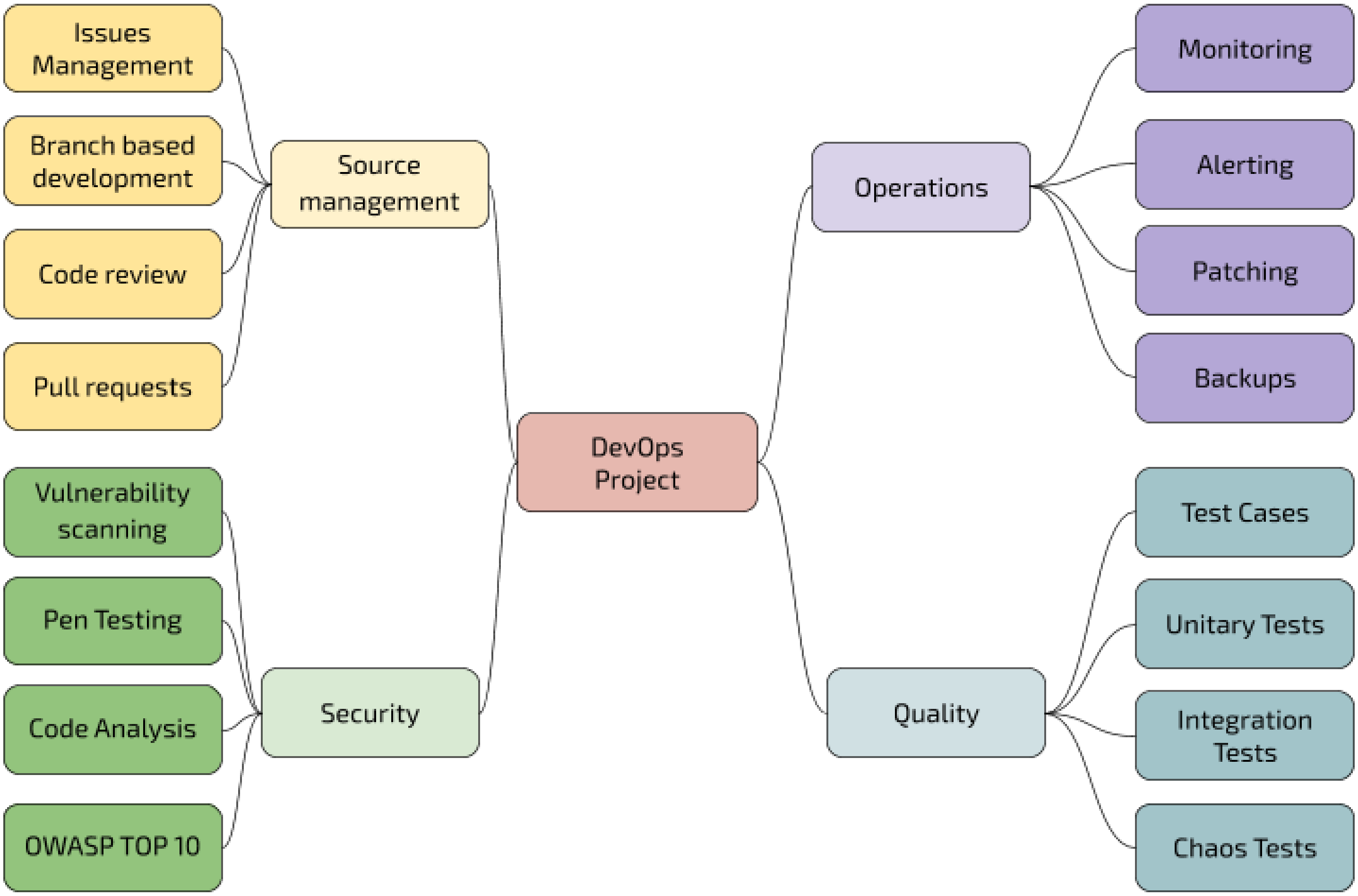
Prepare the cloud environment
Create the Kubernetes cluster
Prepare the image repositories
Prepare the software deployment configurations

Stream 3 - Pipelines:

Prepare the Jenkins server
Test with a hello-world pipeline
Enhance the server image to get the needed tools

Mind map

Once the technical baseline is achieved by the team, it is expected to improve the project with some elements of different categories, here are some of them to take into account in your DevOps journey:



The choice of the technical features to add to the project is up to each team, the only mandatory requirements are: each implemented feature must be explained, demonstrated and integrated into the CI/CD pipelines.

Collaboration

Use of Slack to be able to communicate and exchange pieces of information with the professor.

Evaluation

The DevOps project will be evaluated on the last day of the project (December 7th, 2019). The order of team presentation will be drawn up front. Each team of 4-5 students will have to present their project following the suggested schedule:

1. Present the team roles and responsibilities
2. Demonstrate the baseline requirements with demos
3. Present the achieved steps of the mind map
4. Present the next steps of the software to achieve full production
5. Present a retrospective
    a. Impediments faced during the project
    b. Lessons learned and capitalization