SE489 DevOps Engineering

Course Project

**SE489: DevOps Engineering – Course Project**

The goal of this project is to apply all the principles, techniques, and best practices of DevOps in a real project. You will demonstrate your DevOps engineer skills to release cycles and high-quality products. You will deliver value continuously by building an automated CI/CD pipeline. Those new skills will also be put to the test as you set up automated monitoring and alerting to ensure the delivered value stays valuable.

In this project, you will continue your work on operationalizing microservices by deploying an elastic and fault-tolerant software application using Kubernetes. You will configure this microservice to be highly available by using Kubernetes best practices. You will validate your design by load testing the service and verifying the application architecture performs as designed.

The purpose of this project is to give you a chance to combine what you have learned throughout the course. Each team will be given a chance to choose their project. You will build a CI/CD pipeline for a microservices application for different deployment strategies. Students define the scope of the project and select the right deployment strategy based on different business requirements.

**Suggested Tools:**

* Git – For version control for tracking changes in the code files
* Maven – For software packaging
* Jenkins – For continuous integration and continuous deployment
* Docker – For container image which is a lightweight, executable package of the software which includes everything needed to run the image (eg. code, libraries, etc)
* Puppet - Open-source software configuration management tool
* Nagios - Application monitoring tool

**Suggested Hints:**

The course project is going to focus on open-source tools. For a quality course project, you need to ensure the following:

* The project strictly follows the DRY principle: everything that must appear in multiple places in the project has a single source.
* Version control should be used for every part of the project, including the source code, build scripts, graphics, documentation, and everything else.
* Make sure to test changes locally before committing.
* The project should have automated testing with over 50% test coverage, and, if web-based, has Selenium tests.
* All of your code should pass a static analysis tool for the language you are using.
* There is an automated build process for any built project components.
* Use a Docker container for development, and deploys can be done via a container. Furthermore, scripts should build the containers automatically.
* The project should be deployed to the cloud with monitoring capabilities.

**GitHub Links for similar Projects:**

<https://github.com/PacktPublishing/hello-world>

<https://github.com/yankils/Simple-DevOps-Project>

<https://github.com/ksemaev/project_template>

<https://github.com/Abhinav-26/DevOps-Projects>

<https://github.com/johnbedeir/Devops-Tools-Documentation>

<https://github.com/icetlab/devopscourse>

<https://github.com/awsmug/wp-devops>

**Project Phases:**

Phase 1

* + Set up a Slack channel for communications within your team
  + Set up a GitHub repository for your project
  + Set up a Kanban board for managing workflow within your team using GitHub “Projects” feature
* Formulate your project proposal
* Setup a running version of your application

Phase 2

* + Run tests locally, as well as push code into production
  + Keep developing your application
  + Testing: you should be writing tests as you write your code. Proper code styling should be enforced
  + Set up a CI/CD pipeline. Use it to push your code to the cloud.
  + Create a Docker container for your application. Deploy it to a cloud service. Create monitoring.

Phase 3

* + Final project report and presentations.