SIT323/SIT737- Cloud Native Application Development 10.2D: Continues Delivery Pipeline for Cloud-Native Apps

---

Overview

You are required to develop a continuous delivery pipeline for a cloud-native web application built with Node.js and deployed on Google Cloud Platform (GCP). The pipeline should enable you to automatically build, test, and deploy the application to the GCP environment in a reliable and secure manner.

Part I

The required tools for doing this task are as follows:

- Git (https://github.com)
- Visual Studio (https://code.visualstudio.com/)
- Node.js (https://nodejs.org/en/download/)
- Docker
- Kubernetes // a computing platform to host your microservice
- Kubectl // the command-line tool for interacting with Kubernetes cluster
- MongoDB
- Docker Compose
- GCP access

Instructions

- Build a cloud-native web application using Node.js;
- Containerize the application using Docker and create a Dockerfile for the application;
- Set up a GCP account, create a project, and enable the relevant APIs for your project;
- Use Google Container Registry (GCR) to store the Docker image of your application;
- Configure a Continuous Integration (CI) pipeline using Google Cloud Build to build the Docker image, run unit tests, and perform any other necessary checks;
- Configure a Continuous Deployment (CD) pipeline using Google Kubernetes Engine (GKE) to deploy the application automatically to a Kubernetes cluster on GCP;
- Configure monitoring and logging for your application using GCP tools such as Stackdriver Monitoring and Stackdriver Logging or any other tools in GCP you are confident with.

Part II (Only for SIT737 students)

- By continuing Part I, conduct a thorough review of your CI/CD pipeline, including the tools, processes, and workflows used to build, test, and deploy your application.
- Analyze the data to identify bottlenecks, inefficiencies, and areas for improvement in your pipeline and list your suggestions for any future improvements.

Deliverables

- A cloud-native web application built with Node.js and containerized using Docker.
- A Dockerfile for the application.
- A CI pipeline configured using Google Cloud Build.
- A CD pipeline configured using Google Kubernetes Engine.
- Monitoring and logging configured using GCP tools such as Stackdriver Monitoring and Stackdriver Logging.

Submission Details

- A link to the GitHub repository containing the source code and configuration files for your application, CI pipeline, and CD pipeline.
- A demonstration of your continuous delivery pipeline in action, showing how changes to the application code trigger the pipeline, and how the pipeline builds, tests, and deploys the application automatically to the GCP environment.
- Submit the report documenting the steps taken, tools and configurations used, and any challenges encountered.