University of Sunderland School of Computer Science

CETM67 - Software Platforms, Infrastructure and Deployment

Assignment 2 of 2 - 40% of the summative value of the module

Extract from the module descriptor - The apprentice will implement a fully automated build and release pipeline for a full-stack web application that includes an element of monitoring and analysis. The assessment will include a demonstration and an upload of the source code/configuration files to a stated repository such as Github.

Assignment 2 of 2: CI/CD pipeline and demo

The assessment covers learning outcomes **1** and **2**:

1: Achieve advanced knowledge of DevOps principles including the design, implementation and management of CI/CD (continuous integration and continuous delivery) frameworks, automation of software deployment, configuration management, automated testing and monitoring/analysis.

2: Develop a fully automated build and release pipeline for a full-stack web application that includes an element of monitoring and analysis.

The detail in the learning outcomes should be reflected in the deliverable for the assignment.

Scenario

You have already created a full-stack web application (CETM65) and Microservices in (CETM67) now is the time to automate the build and deployment of one of these artefacts by implementing a Lambda Function into a fully automated CI/CD pipeline.

You are to build the pipeline around an existing Lambda Function (or a new Lambda Function) and use this to fully automate your build (including test) and artefact delivery process (package to S3). The resulting Lambda Function will then be monitored and analysed through the use of an interactive dashboard. This dashboard will enable you to identify problems with the Lambda through the monitoring of key performance metrics such as HTTP errors.

Finally, you will be required to demonstrate the creation and use of both the pipeline and the interactive dashboard and discuss how this combination will enable effective DevOps practice.

Important Information

You are required to submit your work within the bounds of the University Infringement of Assessment Regulations (see your Programme Guide). Plagiarism, paraphrasing and downloading large amounts of information from external sources, will not be tolerated and will be dealt with severely. Although you should make full use of any source material, which would normally be an occasional sentence and/or paragraph (referenced) followed by your own critical analysis/evaluation. You will receive no marks for work that is not your own. Your work may be subject to checks for originality which can include use of an electronic plagiarism detection service. Where you are asked to submit an individual piece of work, the work must be entirely your own. The safety of your assessments is your responsibility.

You must not permit another student access to your work. Where referencing is required, unless otherwise stated, the Harvard referencing style is to be used.

Maximum hours it should take to complete: 48hrs
Assignment Specification

A fully automated CI/CD pipeline that includes an element of monitoring and analysis.

The assessment will include a demonstration and an upload of the source code to a stated repository such as Github.

Deliverables:

The following elements constitute the deliverables in this assignment:

1. A **Lambda Function** (for this you can use one of the Lambda Functions you created in CETM67: Assignment 1)
2. A **CI/CD pipeline** that includes:
    - GitHub integration
    - Unit Tests
    - Linting
    - Package creation (.zip)
    - Package delivery to S3

3. **Monitoring and analysis**:
    - An interactive **Cloudwatch** dashboard that includes the following:
      - **Metrics**
        - Application usage and traffic e.g.
          - HTTP request volume
          - HTTP Response Types e.g. Failed requests
          - Lambda Invocations
          - Misc e.g. Number of Objects in S3, Total size of objects in S3, DynamoDB metrics etc..
      - **Alarms**
        - Create a Cloudwatch Alarm to inform when:
          - A certain threshold of HTTP errors is breached

4. **Presentation/demo** (**20 minutes**) that evidences:
    - Reasons for using CI/CD practices to enable DevOps
    - Explanation of individual technology components used in pipeline e.g. GitHub Actions, Workflows, yaml files and advantages/disadvantages etc
    - Pipeline in action
    - Dashboard in action
    - Lessons learned

Hand-in date and feedback due date:

The hand-in date for this assignment is **23:59PM on 23/07/2021**

Assessment Criteria

Criteria for Assessing software system – these indicate the level of academic work that will be assessed in the relevant part of the assessment requirements.

Criteria	Fail (up to10)	Fail (up to 39)	Pass (up to 50)	Pass (up to 60)	Merit (up to 69)	Distinction (up to 79)	Distinction (up to 89)	Distinction (up to 100)
CI/CD pipeline and dashboard functionality. (70%)	No CI/CD pipeline submitted to GitHub and/or Canvas. Or, irrelevant files submitted, and pipeline is not functional.	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests An interactive dashboard that includes: Minimum of 1 metrics	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests An interactive dashboard that includes: Minimum of 2 metrics	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests An interactive dashboard that includes: Minimum of 2 metrics Minimum of 1 Alarm	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests, Static Analysis and Linting checks An interactive dashboard that includes: Minimum of 3 metrics Minimum of 2 Alarm	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests, Static Analysis and Linting checks A comprehensive interactive dashboard that includes: Minimum of 5 metrics Minimum of 3 Alarms	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests, Static Analysis and Linting checks A comprehensive interactive dashboard that includes: Minimum of 5 metrics Minimum of 4 Alarms	A fully automated CI/CD pipeline that includes: Element of a full-stack web app e.g. Lambda Function. GitHub Integration Unit Tests, Static Analysis and Linting checks A comprehensive interactive dashboard that includes: Minimum of 6 metrics Minimum of 4 Alarms
CI/CD pipeline demonstration (30%)	No demonstration of the CI/CD pipeline.	A demonstration of the CI/CD pipeline that provides no answers to reviewer's questions.	A demonstration of the CI/CD pipeline that struggles to answer most questions.	A demonstration of the CI/CD pipeline that addresses one or two questions of reviewer in full.	A reasonably presented demonstration of the CI/CD pipeline that addresses a few questions of reviewer in full.	A coherent and logically presented demonstration of the CI/CD pipeline that addresses some questions of reviewer in full.	A coherent and logically presented demonstration of the CI/CD pipeline that addresses most questions of reviewer in full.	A coherent and logically presented demonstration of the CI/CD pipeline that addresses all questions of reviewer in full.