ACIT 4850 – Enterprise Systems Integration – Assignment 2

Instructor	Mike Mulder (mmulder10@bcit.ca or Discord)
Total Marks	20
Due Dates	March 24th at midnight

<u>This assignment is to be completed and submitted with your partner (i.e., Group submission) if applicable.</u>

<u>Overview</u>

This time you already have a solution for the tool you are evaluating in your Enterprise Development Environment: Jenkins for CI/CD. But your software development team is looking at alternative new options – because developers like trying out new tools – for Continuous Integration.

Jenkins vs. GitLab CI

We already have Jenkins in our Enterprise Development Environment as of Lab 4 for Continuous Integration. And in Lab 8 you prototyped the use of GitLab CI for our Point Python Pipeline using either a Shell or Docker Runner. Now you are going to prototype GitLab CI for our Java Sample Pipeline.

Part A – Practical (10 marks)

- You must configure GitLab CI on your Cloud VM (you cannot use the online version of GitLab CI).
- Reproduce your Java Sample Pipeline using GitLab CI with a **Docker Runner**. You can use the Maven docker image which has Java and Maven both installed by default (https://hub.docker.com/_/maven)
- The stages you must reproduce are:
  - Build
  - Test
  - Deliver

  Note: Do not have to reproduce the SonarQube integration
- The Build stage builds the Java JAR file that is then run in the Deliver stage. However, in GitLab CI build outputs aren't carried over from stage to stage when using a Docker runner. **You must investigate and use <u>caching</u> in GitLab CI to deal with this issue.**

Submission:

- (2 marks) Screenshot of your Docker runner, proving that it is a Docker running on your GitLab VM.

- (2 marks) Screenshot of your pipeline running successfully with the above stages. It must show the URL of your GitLab installation (with your Group #).
- (2 marks) Screenshot of the output of your Deliver stage showing the output of the built Java program. It must show the URL of your GitLab installation (with your Group #).
- (4 marks) Your .gitlab-ci.yml file with the stages above and leveraging the Maven docker image and caching. Make sure to include a comment in your gitlab-ci.yml file with your Group # and names.

Part B – Written

The answers to the following questions should reflect your hands-on experience with both Jenkins and GitLab CI and less on online research (though you may have to do some research and experimentation):

Answer the following questions comparing Jenkins and GitLab CI:

1. (3 marks) Identify three similarities between Jenkins and GitLab CI
2. (3 marks) Identify three differences between Jenkins and GitLab CI
3. (4 marks) Define two scenarios – i.e., companies with software teams – that would use a CI/CD tool. For one scenario, recommend and justify Jenkins as the preferred choice. For the other scenario, recommend and justify GitLab CI as the preferred choice.

Submission

Submit your screenshots, gitlab-ci.yml file and PDF with your answers to the written portion to the Assignment 2 dropbox by March 24th at midnight.