ACIT 4850 – Enterprise Systems Integration – Assignment 3

Total Marks: 20
Due Date: April 8th for Set C, April 9th for Set B, April 11th for Set A

This assignment is to be completed and submitted individually.

Overview

For this assignment you are going to create CI/CD pipelines for each of your 3855 backend services.

If you are not currently in 3855, please let me know to arrange an alternate assignment.

CI/CD Pipelines for 3855 Microservices

This assignment will require completion before the end of the last class (see above) with a brief demo.

Using a single shared library and a Jenkinsfile per service, create a CI/CD pipeline for each of your Python based 3855 microservices (Receiver, Storage, Processing) in your Jenkins installation. The pipeline can pull the code for your services from your existing Git repo on GitLab.com or GitHub.com, it doesn't need to be put in to your own GitLab installation.

The CI/CD pipeline must have at minimum the following four stages:

- (1 mark) Lint – Run pylint on your code and enforce a minimum score of 5 out of 10. You may have to fix some of the warnings in your code to get to pass this stage.
- (2 marks) <Your Security Stage> - A stages that performs some security assessment of your Python code, Python dependencies or Docker image. For instance, scan the Docker image for vulnerabilities or scan your Python dependencies for vulnerabilities. You will need to research free or open source tools you can use to do these scans. ***You must be able to provide a reason for your choice of security scan – i.e., what does the tool scan for and why is that important***
- (1 mark) Package – Builds the docker image and pushes it to a repository in your Dockerhub account
- (3 marks) Deploy – A stage that remotely pulls your updated image on your 3855 VM and redeploys via a docker-compose up -d command (you should NOT have to bring all the services down, it should recreate the container with the updated image). You will need to create a credential in Jenkins and make a remote calls to your 3855 VM from your Jenkins VM.

A change to the code in any of the Receiver, Storage or Processing services must automatically trigger a build of your corresponding CI/CD pipeline (2 marks).

Clarification:

If you are using GitHub with a single repo for all four services:

- *It is okay if any change to that repo trigger Jenkins builds to all four services.*
- *But you then cannot have an automatically run Deploy stage in the CI pipeline for the services.*
- *Instead, have a separate manually initiated Deploy stage that pulls the Docker image from DockerHub to your 3855 VM and runs docker-compose up -d*

Your docker-compose.yml file must be updated to use the images from your Dockerhub account (2 marks).

Your demonstration must include:

- (3 marks) Running of the CI/CD pipeline of each service and showing the updated image in Dockerhub
- (6 marks) Proving and showing that Docker Compose is using your images on Dockerhub
    - Show the updates you made to the file
    - Show a change made to one of your services getting built by your CI/CD pipeline and then automatically redeployed via a docker-compose up –d command
- Justification for your choice of tool for your Security Stage.

The submission to the dropbox on D2L must include the following to receive your marks:

- Your Groovy file with your Shared Pipeline Definition
- Updated docker-compose.yml file
- Screenshot of the pipeline stages in Jenkins for one of your services (successfully built)
- Screenshots of your Dockerhub showing the repositories for the services recently updated