LS lab 3: CI/CD Infrastructure

In this lab, we will focus on how to set up CI/CD infrastructure by deploying Gitlab on-premise:

- Self-managed Gitlab server
- Gitlab Runner
- Deployment server
- CI/CD pipeline
- Web application with test case

Task 1: Infra Deployment

1. Deploy three VMs that you will be using as Gitlab Server, Gitlab Runner, and the deployment server. Make sure VMs can reach each other.
2. Set up **Gitlab Server** (VM1), create a docker-compose file with below configs:
    a. Pull the Gitlab EE or CE edition
    b. Name the running container as <stx>-gitlab
    c. Map container ports 80 and 22 to host machine
    d. Expose the Gitlab server as <stx>.sne.com. (Hint: find the right env variable to pass to the container to update the configs of Gitlab. Update the *hosts* file to resolve the mentioned DNS record)
    e. Bind the necessary directories of the Gitlab server container to host machine (e.g. logs, app data, configs…)
    f. Run the docker-compose file and make sure the configs are working.
    g. Access the Gitlab server and log in, create a project name it as <stx>-repo.
    Note: you can also check https://docs.gitlab.com/ee/install/docker.html

3. Set up the **Gitlab Runner** (VM2), don't use the docker approach this time.
    a. Install and configure shared Gitlab Runner
    b. Explain what is the Gitlab runner executor and set the executor type to *shell*
    c. Set Gitlab Runner tag to <stx>-runner. (You will be using this tag in the pipeline in the coming task)
    d. Authenticate your Gitlab runner with Gitlab server, and validate.
    Note: you can also check https://docs.gitlab.com/runner/install/

4. Set up the **Deployment Server** (VM3).
    a. Set up SSH connection such that your Gitlab runner should be able to run scripts on the deployment server.

Task 2: Create CI/CD Pipeline

1. Clone the project you have created in step **1.2.g**.
2. Write a simple web application in any programming language. (E.g. Random text or Addition of two numbers)
3. Create CI/CD pipeline (.gitlab-ci.yml)
    a. CI stages of the pipeline should:
        i. Build the application

ii. Run test (to check the application works ok)
iii. Build docker image (Note: you need Dockerfile)
iv. Push to your docker hub account.

b. CD stages of the pipeline should:

i. Pull the docker image and deploy it on deployment server

4. Validate that the deployment is successful by accessing the web app via the browser on deployment server side.

Task 3: (Bonus)

5. Update the CD stages to be able to deploy the web applicate using Ansible.
6. Update the pipeline to support multi-branch (e.g. master and develop) and jobs should be triggered based on the specific target branch.
7. Update keywords such as cache, artifact, needs, and decencies to have more control of pipeline execution.