# Lab Assignment 01

## Requirement

This assignment requires you to prepare and configure an environment with an automated CI/CD pipeline. It will containerize applications using Jenkins, Docker, and use Helm to deploy on Kubernetes. Every time the content of the app has changed and push to SCM (source code manager), go to Jenkins, trigger the pipeline to deploy the new version of the App. You also need to apply the New Relic for monitoring the infrastructure at the basic level.

In the end, there are some suggested ideas for you. Using them wisely!

## Architecture

Diagram

Description automatically generated

## Infrastructure Deployment requirements

Using Helm chart to deploy services as architecture, include:

1. Deploy Postgres (DB)
2. Deploy NodeJS API Services (BE) connect into DB and the same pod with DB
3. Deploy Angular WebApp (FE)

## Configuration Management requirements

1. Configure Angular WebApp with port 4200 (Container port)
2. Configure NodeJS API Services with port 3000 (Container port)
3. Configure Postgres with port 5432 (Container port)
4. Configure NodeJS API Services connect to Postgres
5. Configure Angular WebApp call API to NodeJS API Services

## CI/CD Configuration requirements

1. At Frontend Gitlab’s repository, add Jenkinsfile file to automate build and deployments. Apply for master branch with released tag.
2. At Backend Gitlab’s repository, add Jenkinsfile to automate build and deployments. Apply for master branch with released tag.

## Instruction

Assumption:

1. You don’t have any information about Internal Gitlab’s server
2. You don’t have any exist Helm’s Charts template
3. You don’t have any idea about product, include FE, BE

You should (not enough):

1. Identify all dependencies (example: Kubernetes, Gitlab’s server) and related information (host, account, database, user/pass …)
2. Create a basic workflow between FE, BE and Postgres (FE call API to BE, BE connect Postgres to get data and return to FE)
3. Etc …

## Prerequisites

|  |  |
| --- | --- |
| **Components** | **Recommended configuration** |
| Kubernestes cluster | • Consider setting up a cluster for your data center • Setup a k8s cluster on your lab VM |
| Docker registry | • Consider a solution for hosting a private Docker registry  • Docker Hub is recommended for this assignment |
| Jenkins | • Consider to setup a Jenkins server in your own lab VM • Jenkins needs to be configured with Docker, Kubectl, Helm |
| SCM | • Consider to setup a Gitlab server in your own lab VM |

## Suggestion

To do this assignment, you have to break down it into small pieces of tasks. Thinking about these ones according to the requirement:

- A Dockerfile for building an image for our App.

- The app is just an HTML file, which is served by the Nginx server.

- Jenkins script, which declare some stage to help Jenkins knows what to do

*Note: \* You can do more than required as well as it optimizes your work. There will be a bonus plus in evaluation*

## Expected Output

- Once you've done this assignment, your output should look like those screenshots below.

- Furthermore, you need to submit A DOCUMENT in which contains a step-to-step guideline and all screenshots from files, scripts, configurations, etc related to the assignment.

- Document needs to be focused, well structured, concise, clear and full of necessary details.

*Note: (You should treat this document as a guide for those who assumed do not know anything.*

*After reading your guide, they could understand the basic concepts and be able to do the same result)*

- Pipeline output example:

Graphical user interface, table

Description automatically generated

- App output example:

Graphical user interface, application, table, Teams

Description automatically generated

### Source code

Source code backend and frontend were got from NamTH22, the idea is just to have BE and FE to run within container following the required architecture. They were made to run with Ingress Nginx, I customized them to run with Istio.

* Backend source code (BE will be expose through port 3000)



* Fontend source code (FE will be expose through port 4200)



Modify Backend URL in file ***src/environments/invironment.ts***. We are going to deploy this container on node which has IP and port BE: IP:3000 and Container FE will be expose on port 4200

**LET’S ENJOY TRAINING COURSE!**