Computer and Communication Networks

INSTRUCTOR: Dr. Hassaan Khaliq Qureshi
COURSE CODE: EE357
SEMESTER: 6th
Credit Hours: 3 (Theory)+1 (Lab)

The semester project is targeted to help students identify and solve complex engineering problems. Following characteristics of complex engineering problem are targeted in this assignment:

WP 1 Depth of Knowledge Required
WP 2 Range of Conflicting Requirements
WP 3 Depth of Analysis Required

CEP Statement:

The IT industry is swiftly moving towards a containerized architecture where services are used to communicate between different applications instead of a monolithic architecture. While doing this implementation you will use concepts learned in the labs till now.

Problem definition:

Create a containerized weather application available as a web front end using Minikube.

Analysis of the problem:

This is just one way of doing things, you can use any language/framework that you want.

1. Create a python application that uses OpenWeatherAPI to get the weather data.
2. Use HTML and a python server framework (Flask being the simplest)
3. Use docker to containerize this application.
4. Create a pod in Minikube using this container/image
5. Expose this pod using a service on a port.
6. Observer and interact with this application on your host machine (figure out how to make your application available on your host machine)
7. Finally deploy the Minikube dashboard to observe statistics related to your cluster.

Evaluation:

Students will be evaluated on the following criteria:

- Implementation: 70%
- Output/result achieved: 30%
- Extra credit for the following:
    - Proper File Structure format for the python program
    - Your Program returns the weather of the next 7 days upon going to the website.
    - Divide the web application into 2 parts: front-end and back-end, the back end will get exposed as a service, and you will use the concept of post and get to communicate between them.

Summary:

Following are salient outcomes of this assignment in terms of complex engineering problem:

- Learning architecture and use cases of containerized applications which are now extensively used in computer networks/virtualized network functions.
- The implementation gives them a chance to use their engineering knowledge to solve a real world problem in an effective and structured way.

Miscellaneous

- Any Linux Distribution you want (CentOS and Ubuntu are recommended) as long as it can run docker and Minikube *AND* you are using the CLI version of the distro.

Quality of Life things

MobaXTerm: Allows you to connect to your virtual machine and provides a much better interface than the virtual machine.

Some Helpful Tips:

- Try to test things very early: When you create the applications, test them locally before you containerize them, test the docker file by running it on a port locally before you create a pod.
- Make sure that all ports and service names are correct.