# Assignment: Dockerizing a Java Application and Deploying it on Kubernetes

## Objective

This assignment will test your ability to work with Git, Docker, Kubernetes, and Jenkins to set up a CI/CD pipeline. You will deploy a simple Java application in a Kubernetes cluster, automate the process using Jenkins, and scale the deployment.

## Prerequisites

• Basic knowledge of Git, Docker, Kubernetes, and Jenkins  
• Access to a Linux VM with Docker and Kubernetes (minikube or a managed cluster)  
• A GitHub and DockerHub account

## Test Details

* Task 1, 2 3 & 4 are compulsory tasks that need to be completed by all candidates.
* Task 5 to 7 are bonus tasks. These tasks can be done based on the time available and will be assessed separately.
* The grading for the test will be as per the first 4 tasks. Candidates who complete the bonus tasks will receive added recognition in their overall performance evaluation during the training period.
* Exam duration: **3.5 hours**

## Task 1: Fork a Java Repository

1. Fork the given Java HelloWorld Git repository from GitHub.  
2. Clone it into your local machine.

## Task 2: Define a Dockerfile

Create a `Dockerfile` to containerize the Java application. It should:  
• Use an appropriate base image (e.g., OpenJDK).  
• Copy the Java files into the container.  
• Compile and execute the Java program.

## Task 3: Build and Push Docker Image

1. Build a Docker image from the Dockerfile.

## Task 4: Create a Kubernetes Deployment

Define a Deployment YAML file (`deployment.yaml`) for your application and apply it using `kubectl apply -f deployment.yaml`.

## Task 5: Scale Up and Scale Down

Scale out to 3 replicas:

kubectl scale deployment helloworld-deployment --replicas=3

Scale down to 1 replica:

kubectl scale deployment helloworld-deployment --replicas=1

## Task 6: Expose the Service Using NodePort

Create a `service.yaml` file and expose the service on a NodePort. Then, access it using `curl http://<public-ip>:30007`.

## Task 7: Automate Deployment Using Jenkins

Install Jenkins, create a Freestyle Jenkins Job, and configure GitHub webhooks to automate deployment.

## Submission

Submit the following:  
1. GitHub repo URL of your Java application.  
2. DockerHub repository URL of your image.  
3. Kubernetes YAML files (`deployment.yaml`, `service.yaml`).  
4. Screenshots of running Kubernetes resources and Jenkins pipeline execution.

Best wishes!!