

Assignment: Application Deployment with Docker, Kubernetes, and CI/CD

Objective

This assignment evaluates your ability to work with a generic application stack using Docker, Kubernetes, and Jenkins to build, containerize, and deploy applications in a CI/CD pipeline.

Prerequisites

- Familiarity with Git, Docker, Kubernetes, and Jenkins
- Access to a Linux VM with Docker, Kubernetes (minikube or managed cluster), and Jenkins
- GitHub and DockerHub accounts

Test Details

- All the questions are compulsory and need to be given equal weightage
- If the files and screenshots are not uploaded, there will be a negative grading of 10 marks.
- Exam duration: **3 hours**

Question 1: Containerize and Push the Application

- Fork and clone a sample application repository (Java, Node.js, or Python preferred).
- Package the application as required (e.g., JAR, ZIP, etc.).
- Write a Dockerfile to containerize the application.
- Build the Docker image and push it to DockerHub.

Question 2: Kubernetes Deployment and Service Exposure

- Create Kubernetes `deployment.yaml` to deploy your Docker image.
- Create `service.yaml` to expose it using NodePort.
- Apply both YAMLs using `kubectl`, and verify deployment.
- Use `curl` with public IP and node port to access the service.

Question 3: Scale the Deployment

- Scale the application to 3 replicas using `kubectl scale`.
- Validate scaling with `kubectl get pods`.
- Scale it back down to 1 replica.

Question 4: Automate Deployment with Jenkins

- Install Jenkins and create a Freestyle Job that:
 - Pulls code from GitHub
 - Builds and packages the application
 - Builds and pushes the Docker image

- Applies Kubernetes deployment and service YAMLs
- Configure GitHub webhook to trigger Jenkins job on push.

Submission

Submit the following:

- GitHub repo URL
- DockerHub image link
- YAML files used (deployment and service)
- Screenshots: image pushed to DockerHub, running Kubernetes pods and services, Jenkins build and deployment logs

Best wishes!!