Part 3: Kubernetes (25 points)

Kubernetes is a powerful tool for managing containerized applications at scale. In this section, you'll explore the basics of deployments, services, and scaling in Kubernetes.

Theory Questions (10 points)

a) **Kubernetes Architecture**

- What are the core components of a Kubernetes cluster (e.g., master, node, etcd, kube-apiserver)? Briefly explain their roles.
- What is a pod in Kubernetes, and how does it differ from a Docker container?

b) **Deployments and Services**

- Explain the purpose of a Kubernetes deployment. How do deployments ensure high availability of applications?
- What are the different types of services in Kubernetes (e.g., ClusterIP, NodePort, LoadBalancer)? When would you use each type?

c) **Scaling and Autoscaling**

- How does Kubernetes handle scaling? Explain the concept of *Horizontal Pod Autoscaler* and how it responds to workload changes.

Practical Task (15 points)

a) **Create a Deployment**

- Create a Kubernetes deployment that runs 3 replicas of the web server container from **Assignment 2**.
- Ensure that all replicas are load-balanced across the cluster using a ClusterIP service.
- Describe how you would test the load balancing functionality.

b) **Service Exposure**

- Expose your deployment to the outside world using a NodePort service. Map the external port to 80 on the Kubernetes cluster.
- Verify the service is reachable by accessing the external IP and port from your browser.

c) **Scaling with Autoscaling**

- Set up the Kubernetes Horizontal Pod Autoscaler (HPA) to automatically scale the web server deployment up to 10 replicas when CPU utilization exceeds 70%.
- Simulate high CPU usage using kubectl or a stress test tool, and observe how Kubernetes scales the pods.