1. Introduction

In this lab, we worked through the process of deploying a Java Spring Boot application with a PostgreSQL database on a Kubernetes cluster. We utilized both Docker and Kubernetes to manage and orchestrate the application, focusing on various Kubernetes concepts such as Pods, Deployments, Services, Secrets, and role-based access control.

2.1 Kubernetes Cluster Setup

The Kubernetes cluster was set up using Minikube or a cloud-based platform. This cluster provided the environment for managing containerized applications.

2.2 Application Deployment with Docker

We containerized the Java Spring Boot app and PostgreSQL using a Dockerfile and Docker Compose. The application was run locally to ensure proper containerization before deploying to Kubernetes.

2.3 Kubernetes Deployment

The Docker Compose setup was transformed into Kubernetes YAML files for the app and database. These configuration files were applied to deploy the application and expose it via Kubernetes services.

2.4 Kubernetes Dashboard

The Kubernetes Dashboard was set up for visual management of the cluster. A service account was created with cluster admin privileges, and the login token was retrieved to access the Dashboard.

3. Key Concepts Explored

- **Pods**: Basic units that encapsulate the application's containers.
- **Deployments**: Managed scaling and rolling updates of the app.
- **Services**: Ensured communication between the app and the PostgreSQL database.
- **Secrets**: Stored sensitive data like database credentials.
- **Role-Based Access Control (RBAC)**: Managed access to Kubernetes resources.

4. Conclusion

This lab demonstrated the full process of deploying a Java application to Kubernetes, converting Docker deployments to Kubernetes, and managing the cluster through both CLI and the Dashboard UI. It emphasized the power of Kubernetes in orchestrating and scaling containerized applications efficiently