# Kubernetes (K8S) YAML Exercise: Full-Stack Application Deployment on Minikube

This exercise will guide you through deploying a simple full-stack application using Kubernetes (K8S) on Minikube. The application will include a database, a backend server, and a frontend server. You will learn to work with deployments, pods, replicas, namespaces, resource quotas, services, ConfigMaps, and Secrets.

## Prerequisites

- Minikube installed and running on your local machine  
- Basic knowledge of Kubernetes and YAML syntax

## Exercise Overview

1. Create a Namespace  
2. Set a Resource Quota  
3. Deploy a MySQL Database  
4. Deploy a Backend Server  
5. Deploy a Frontend Server  
6. Configure Services (ClusterIP, NodePort, ExternalName)  
7. Use ConfigMaps for Configuration  
8. Use Secrets for Sensitive Data

## Instructions

### 1. Create a Namespace

Create a namespace called `fullstack-app` to isolate the resources.

### 2. Set a Resource Quota

Set a resource quota for the namespace to limit resource usage.

### 3. Deploy a MySQL Database

Create a Secret for the MySQL root password and a ConfigMap for database configuration.

- Create a Secret for the MySQL root password.  
- Create a ConfigMap for MySQL database configuration.  
- Deploy MySQL with the created Secret and ConfigMap.

### 4. Deploy a Backend Server

- Deploy the backend server with environment variables configured to connect to the MySQL database.  
- Ensure the backend server has three replicas.

### 5. Deploy a Frontend Server

- Deploy the frontend server with environment variables configured to connect to the backend server.  
- Ensure the frontend server has three replicas.

### 6. Configure Services

- Create a ClusterIP service for the MySQL database.  
- Create a ClusterIP service for the backend server.  
- Create a NodePort service for the frontend server.  
- Create an ExternalName service to map an external DNS name.

### 7. Use ConfigMaps for Configuration

Create a ConfigMap for the backend server configuration.

### 8. Use Secrets for Sensitive Data

Create a Secret for storing API keys used by the backend server.

## Testing the Setup

Once everything is deployed, you can test the setup using `curl` commands.

1. \*\*Check the Frontend Service:\*\*  
```sh  
curl http://<minikube-ip>:30001  
```

2. \*\*Access the Backend Service from the Frontend Pod:\*\*  
```sh  
kubectl exec -it <frontend-pod> -n fullstack-app -- curl http://backend:8080  
```

3. \*\*Verify Database Connection from the Backend Pod:\*\*  
```sh  
kubectl exec -it <backend-pod> -n fullstack-app -- curl http://mysql:3306  
```

## Submission

Please submit the following YAML files:  
1. Namespace definition  
2. Resource Quota definition  
3. Secret for MySQL root password  
4. ConfigMap for MySQL configuration  
5. MySQL Deployment and Service  
6. Backend Deployment and Service  
7. Frontend Deployment and Service  
8. ConfigMap for Backend configuration  
9. Secret for API keys  
10. ExternalName Service definition