# Kubernetes (K8S) YAML Exercise: Multi-Tier Web Application Deployment on Minikube

This exercise will guide you through deploying a multi-tier web application using Kubernetes (K8S) on Minikube. The application will include a Redis cache, 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 Redis Cache  
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 `multi-tier-app` to isolate the resources.

### 2. Set a Resource Quota

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

### 3. Deploy a Redis Cache

Create a ConfigMap for Redis configuration.

- Create a ConfigMap for Redis configuration.  
- Deploy Redis with the created ConfigMap.

### 4. Deploy a Backend Server

- Deploy the backend server with environment variables configured to connect to the Redis cache.  
- Ensure the backend server has two 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 two replicas.

### 6. Configure Services

- Create a ClusterIP service for the Redis cache.  
- 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 sensitive information 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>:30002  
```

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

3. \*\*Verify Cache Connection from the Backend Pod:\*\*  
```sh  
kubectl exec -it <backend-pod> -n multi-tier-app -- curl http://redis:6379  
```

## Submission

Please submit the following YAML files:  
1. Namespace definition  
2. Resource Quota definition  
3. ConfigMap for Redis configuration  
4. Redis Deployment and Service  
5. Backend Deployment and Service  
6. Frontend Deployment and Service  
7. ConfigMap for Backend configuration  
8. Secret for sensitive information  
9. ExternalName Service definition