**PROJECT 1: DEPLOY A MULTI-TIER WEB APPLICATION ON KUBERNETES**

This project will guide you through deploying a **Flask (backend) + MySQL (database) + Nginx (frontend)** web application on Kubernetes step by step.

**1️. Setup Kubernetes Cluster**

If you don't have a Kubernetes cluster, set up Minikube:

**minikube start**

**kubectl cluster-info**

**kubectl get nodes**

**2️. Deploy MySQL Database (StatefulSet)**

**Step 1: Create a Persistent Volume for MySQL**

**# mysql-pv.yaml**

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: mysql-pv**

**spec:**

**capacity:**

**storage: 1Gi**

**accessModes:**

**- ReadWriteOnce**

**hostPath:**

**path: "/mnt/data"**

**kubectl apply -f mysql-pv.yaml**

**Step 2: Create MySQL Secret for Password Storage**

**# mysql-secret.yaml**

**apiVersion: v1**

**kind: Secret**

**metadata:**

**name: mysql-secret**

**type: Opaque**

**data:**

**mysql-root-password: cGFzc3dvcmQ= # Base64 encoded password (e.g., "password")**

**kubectl apply -f mysql-secret.yaml**

**Step 3: Deploy MySQL as a StatefulSet**

**# mysql-deployment.yaml**

**apiVersion: apps/v1**

**kind: StatefulSet**

**metadata:**

**name: mysql**

**spec:**

**serviceName: mysql**

**replicas: 1**

**selector:**

**matchLabels:**

**app: mysql**

**template:**

**metadata:**

**labels:**

**app: mysql**

**spec:**

**containers:**

**- name: mysql**

**image: mysql:5.7**

**env:**

**- name: MYSQL\_ROOT\_PASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mysql-secret**

**key: mysql-root-password**

**ports:**

**- containerPort: 3306**

**name: mysql**

**kubectl apply -f mysql-deployment.yaml**

Verify MySQL deployment:

**kubectl get pods**

**3️. Deploy Flask Backend**

**Step 1: Create a Flask App (Dockerized)**

**# app.py**

**from flask import Flask**

**import mysql.connector**

**app = Flask(\_\_name\_\_)**

**@app.route("/")**

**def hello():**

**return "Hello from Flask!"**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(host="0.0.0.0", port=5000)**

**Step 2: Create a Dockerfile**

**# Dockerfile**

**FROM python:3.8**

**WORKDIR /app**

**COPY app.py .**

**RUN pip install flask mysql-connector-python**

**CMD ["python", "app.py"]**

Build & push Docker image:

**docker build -t your-dockerhub/flask-app .**

**docker push your-dockerhub/flask-app**

**Step 3: Deploy Flask App on Kubernetes**

**# flask-deployment.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: flask-app**

**spec:**

**replicas: 2**

**selector:**

**matchLabels:**

**app: flask**

**template:**

**metadata:**

**labels:**

**app: flask**

**spec:**

**containers:**

**- name: flask**

**image: your-dockerhub/flask-app**

**ports:**

**- containerPort: 5000**

**kubectl apply -f flask-deployment.yaml**

Verify Flask deployment:

**kubectl get pods**

**4️. Deploy Nginx as Frontend**

**# nginx-deployment.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: nginx**

**template:**

**metadata:**

**labels:**

**app: nginx**

**spec:**

**containers:**

**- name: nginx**

**image: nginx**

**ports:**

**- containerPort: 80**

**kubectl apply -f nginx-deployment.yaml**

**Expose Nginx via NodePort:**

**# nginx-service.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: nginx-service**

**spec:**

**selector:**

**app: nginx**

**ports:**

**- protocol: TCP**

**port: 80**

**targetPort: 80**

**nodePort: 30007**

**type: NodePort**

**kubectl apply -f nginx-service.yaml**

**5️. Verify and Test Application**

1. **Check Running Pods**

**kubectl get pods**

1. **Check Running Services**

**kubectl get svc**

1. **Access Flask Application**

**minikube service nginx-service --url**

**Conclusion**

We successfully deployed a **Flask + MySQL + Nginx** web application on Kubernetes.  
Used **StatefulSet, Deployments, Services, Secrets, and Persistent Volumes**.  
Implemented **scalability** by using multiple replicas for the backend.