**Deploying a Flask Application on Kubernetes with Auto-Scaling & Load Testing**

**1️. Building & Containerizing the Flask Application**

Flask Application (app.py)

from flask import Flask, jsonify

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

@app.route('/')

def home():

return jsonify(message="Hello, World! This is a Flask app running in Docker.")

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

**Issue: Flask bound to 127.0.0.1 won't be accessible. Fix: Use app.run(host="0.0.0.0", port=5000).**

**Dockerfile**

FROM python:3.11

WORKDIR /app

COPY . /app

RUN pip install flask

EXPOSE 5000

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

**Build & Push Image**

docker build -t kpkm25/flask-kube .

docker push kpkm25/flask-kube

**2️. Deploying Flask App on Kubernetes**

**Deployment & Service YAML (deployment-service.yaml)**

apiVersion: apps/v1

kind: Deployment

metadata:

name: flask-app

spec:

replicas: 3

selector:

matchLabels:

app: flask-app

template:

metadata:

labels:

app: flask-app

spec:

containers:

- name: flask-container

image: kpkm25/flask-kube:latest

ports:

- containerPort: 5000

resources:

requests:

cpu: "100m"

limits:

cpu: "250m"

imagePullSecrets:

- name: docker-secret

---

apiVersion: v1

kind: Service

metadata:

name: flask-service

spec:

selector:

app: flask-app

ports:

- protocol: TCP

port: 80

targetPort: 5000

type: NodePort

**Issue: ErrImagePull due to unauthenticated Docker pulls. Fix: Authenticate Kubernetes with Docker Hub.**

**Apply Deployment**

kubectl apply -f deployment-service.yaml

**3. Enabling HPA (Horizontal Pod Autoscaler)**

kubectl autoscale deployment flask-app --cpu-percent=50 --min=3 --max=10

kubectl get hpa

**4. Load Testing & Debugging NodePort Issues**

**Testing Service Internally**

kubectl run -it --rm busybox --image=busybox -- /bin/sh

wget -q -O- http://10.97.210.48:80

**Finding NodePort & Testing External Access**

kubectl get svc flask-service

**7️. Simulating Load for HPA**

kubectl run -it --rm load-generator --image=busybox -- /bin/sh

while true; do wget -q -O- http:// 192.168.49.2:31607; done

**Check Scaling**

kubectl get hpa

kubectl get pods

Output:









