Kubernetes Deployment Task

Objective: Deploy a Flask-based web application with a PostgreSQL database on a Minikube-managed Kubernetes cluster.

Task Steps:

1. **Setup Minikube** – Start a Minikube cluster if not already running.
2. **Deploy PostgreSQL Database** – Create a PostgreSQL deployment with a persistent volume for data storage.
3. **Deploy Flask Application** – Containerize and deploy a Flask-based application that connects to the PostgreSQL database.
4. **Service Configuration** – Expose both the database and Flask application within the cluster using Kubernetes services.
5. **Testing & Verification** – Access the Flask application and verify database connectivity.

You are required to fill in the following git template structure for submission:

k8s-flask-app/

├── manifests/

│ ├── deployment/

│ │ ├── flask-deployment.yaml

│ │ ├── postgres-deployment.yaml

│ ├── service/

│ │ ├── flask-service.yaml

│ │ ├── postgres-service.yaml

│ ├── configmap/

│ │ ├── postgres-configmap.yaml

│ ├── secret/

│ │ ├── postgres-secret.yaml

├── app/

│ ├── Dockerfile

│ ├── requirements.txt

│ │— app.py

│— README.md

Further Submission:

In the repository create a “submission” folder and submit the following:

- Snapshot of the internal deployment
- Snap of the ‘kubectl get all’
- Test the effect of the scaling up and down the replica set
- Investigate the min and max replicas count in the deployment file.