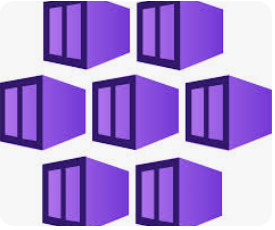
**Design Document (Solution Architecture)**

**Architecture Diagram :**

**Author : Sriram Surendhran**

**Github Link :** [**https://github.com/sriramsurendhran/surge-api-case-study-1.git**](https://github.com/sriramsurendhran/surge-api-case-study-1.git)



**GITUB DOCKER AKS**

1. CI/CD Pipeline: GitHub Actions for automation. The workflow builds, tests, and deploys the Dockerized Flask application to an AKS cluster.

2. Docker: The application is containerized with Docker, enabling easy and consistent deployment across environments.

3. AKS Cluster: A managed Kubernetes cluster provided by Azure is used to host the application, ensuring scalability and availability.

4. Ingress Controller: An NGINX ingress controller manages traffic and exposes the application to the internet

5.SSL Certificate: HTTPS is implemented using a trusted SSL certificate obtained through Cert-Manager (Let's Encrypt).