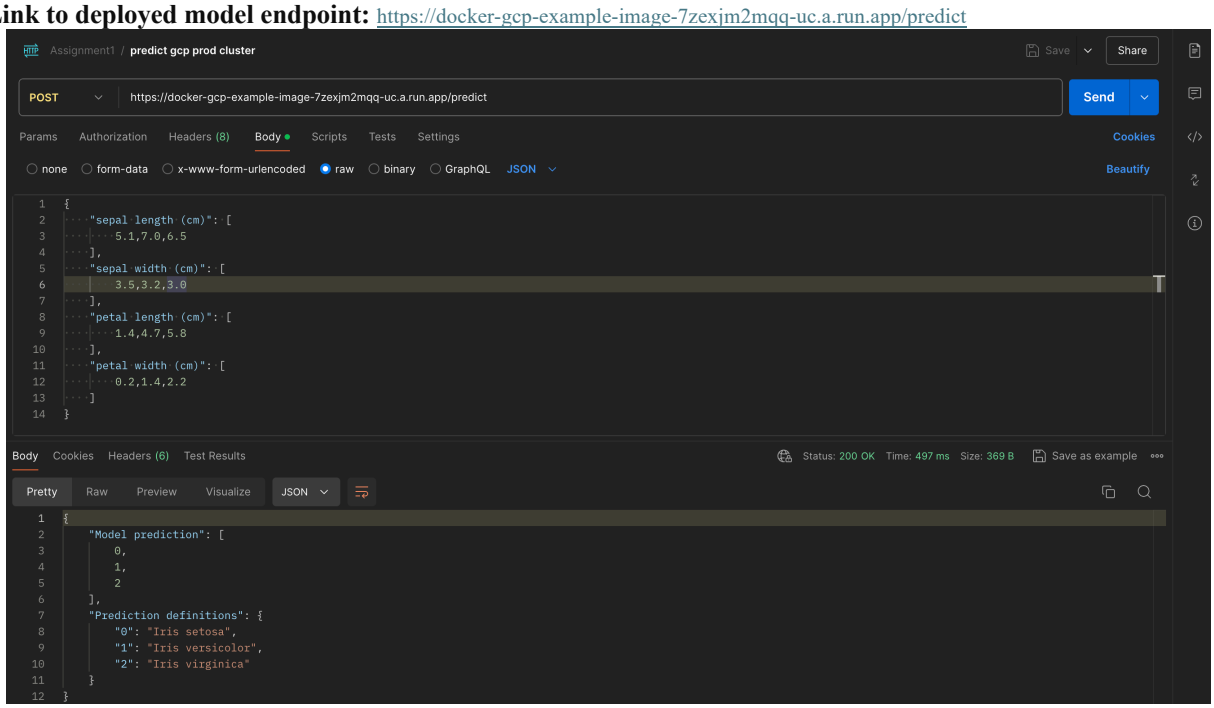
M4 Deliverables:

1. Link to deployed model endpoint: https://docker-gcp-example-image-7zexjm2mqq-uc.a.run.app/predict



2. Link to deployed docker image: https://hub.docker.com/repository/docker/manglamsingh10/iris-pred-flask-app/general

3. A report detailing the deployment and orchestration process:

Step 1: Install and start docker on your machine

Step 2: install minikube and helm:
brew install minikube
brew install helm

Step 3: start minikube cluster with docker driver:
minikube start --driver docker

```
~/B/Semester 3/M/g/ML-Ops_Group35/c/M3 〉 kubernetes-changes ?2 〉 minikube start --driver docker
😄  minikube v1.33.1 on Darwin 14.5 (arm64)
✨  Using the docker driver based on user configuration
📌  Using Docker Desktop driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🔥  Creating docker container (CPUs=2, Memory=7792MB) ...
🐳  Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Step 4: Move to K8 folder and Install helm chart for different environments (namespace) i.e.. dev, qa, prod

Dev:

helm install iris-pred-flask-app-dev-release ./iris-pred-helm-chart --namespace dev --create-namespace -f ./iris-pred-helm-chart/values.yaml -f ./iris-pred-helm-chart/values-dev.yaml

```
~/B/Semester 3/M/g/ML-Ops_Group35/c/M3 〉 kubernetes-changes ?2 〉 cd k8

~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 helm install iris-pred-flask-app-dev-release ./iris-pred-helm-chart --namespace dev --create-namespace -f ./iris-pred-helm-chart/values.yaml -f ./iris-pred-helm-chart/values-dev.yaml
NAME: iris-pred-flask-app-dev-release
LAST DEPLOYED: Mon Aug  5 14:21:15 2024
NAMESPACE: dev
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Congratulations, you have successfully installed the iris-pred-flask-app-dev-release chart!

The application is available at:
  http://localhost:5050

Make sure to tunnel the external request to minikube cluster
    minikube tunnel

To get more details about the deployment, run:
  kubectl get all -n dev
To get the service port
    kubectl get svc -n dev
```

QA:

helm install iris-pred-flask-app-dev-release ./iris-pred-helm-chart --namespace qa --create-namespace -f ./iris-pred-helm-chart/values.yaml -f ./iris-pred-helm-chart/values-qa.yaml

```
~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 helm install iris-pred-flask-app-dev-release ./iris-pred-helm-chart --namespace qa --create-namespace -f ./iris-pred-helm-chart/values.yaml -f ./iris-pred-helm-chart/values-qa.yaml
NAME: iris-pred-flask-app-dev-release
LAST DEPLOYED: Mon Aug  5 14:22:00 2024
NAMESPACE: qa
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Congratulations, you have successfully installed the iris-pred-flask-app-dev-release chart!

The application is available at:
  http://localhost:5051

Make sure to tunnel the external request to minikube cluster
    minikube tunnel

To get more details about the deployment, run:
  kubectl get all -n qa
To get the service port
    kubectl get svc -n qa
```

Prod:

helm install iris-pred-flask-app-dev-release ./iris-pred-helm-chart --namespace prod --create-namespace -f ./iris-pred-helm-chart/values.yaml -f ./iris-pred-helm-chart/values-prod.yaml

```
~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 helm install iris-pred-flask-app-dev-release ./iris-pred-helm-chart --namespace prod --create-namespace -f ./iris-pred-helm-chart/values.yaml -f ./iris-pred-helm-chart/values-prod.yaml
NAME: iris-pred-flask-app-dev-release
LAST DEPLOYED: Mon Aug  5 14:23:05 2024
NAMESPACE: prod
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Congratulations, you have successfully installed the iris-pred-flask-app-dev-release chart!

The application is available at:
  http://localhost:5052

Make sure to tunnel the external request to minikube cluster
    minikube tunnel

To get more details about the deployment, run:
  kubectl get all -n prod
To get the service port
    kubectl get svc -n prod
```

Step 5: Verify pods in all the environments (**1 pod is configured for dev, 2 pod for QA and 3 pod for prod environment**)

kubectl get pods -n dev
kubectl get pods -n qa
kubectl get pods -n prod

```
~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 kubectl get pods -n dev

NAME                                              READY   STATUS    RESTARTS   AGE
iris-pred-flask-app-dev-deployment-cf5bf498c-s5p6c   1/1     Running   0          6m32s

~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 kubectl get pods -n qa

NAME                                              READY   STATUS    RESTARTS   AGE
iris-pred-flask-app-qa-deployment-5f4946c556-9lxbd   1/1     Running   0          5m54s
iris-pred-flask-app-qa-deployment-5f4946c556-n9q95   1/1     Running   0          5m54s

~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 kubectl get pods -n prod

NAME                                               READY   STATUS    RESTARTS   AGE
iris-pred-flask-app-prod-deployment-5d48657549-6knwl   1/1     Running   0          4m53s
iris-pred-flask-app-prod-deployment-5d48657549-fckhd   1/1     Running   0          4m53s
iris-pred-flask-app-prod-deployment-5d48657549-xvkm7   1/1     Running   0          4m53s
```

Step 6: tunnel external request to minikube cluster minikube tunnel:

minikube tunnel

```
~/B/Semester 3/M/g/ML-Ops_Group35/c/M/k8 〉 kubernetes-changes ?2 〉 minikube tunnel
✅  Tunnel successfully started

📌  NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

🏃  Starting tunnel for service iris-pred-flask-app-dev-service.
🏃  Starting tunnel for service iris-pred-flask-app-prod-service.
🏃  Starting tunnel for service iris-pred-flask-app-qa-service.
```

Step 7: Get the service port for all namespaces:

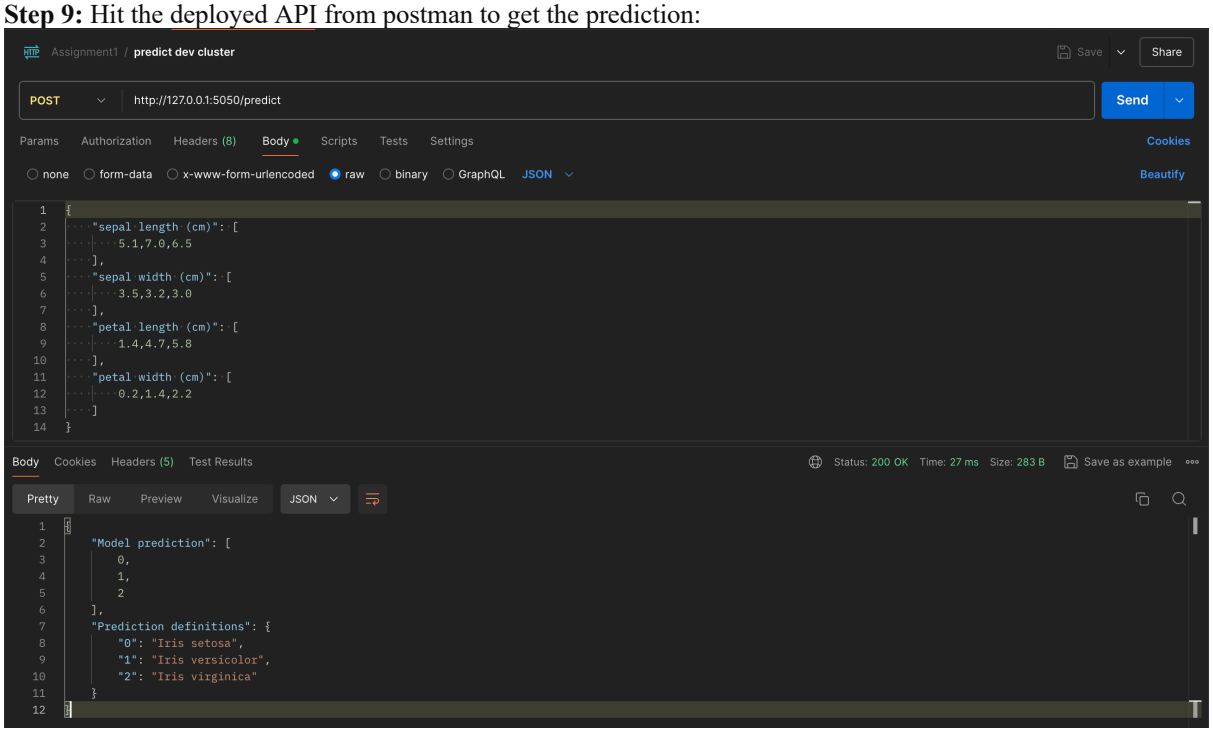
kubectl get svc -n dev
kubectl get svc -n qa
kubectl get svc -n prod

```
~ kubectl get svc -n dev
NAME                            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
iris-pred-flask-app-dev-service   LoadBalancer   10.100.213.87   127.0.0.1     5050:30423/TCP   14m

~ kubectl get svc -n qa
NAME                            TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
iris-pred-flask-app-qa-service   LoadBalancer   10.97.246.65   127.0.0.1     5051:32178/TCP   13m

~ kubectl get svc -n prod
NAME                             TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
iris-pred-flask-app-prod-service   LoadBalancer   10.104.106.59   127.0.0.1     5052:32279/TCP   12m
```

Step 9: Hit the deployed API from postman to get the prediction:



Step 8: Try getting response from QA and Prod namespace as well
QA API url: http://127.0.0.1:5051/predict
Prod API url: http://127.0.0.1:5052/predict