BUYOGO ASSIGNMENT

PART 2:

* **Step 1: Dockerfile for the Python Project to create the Docker image.**1) Go to the root folder and create a Dockerfile with following content before that configure the app.env with your username and password:

FROM python:3.11

WORKDIR /app

COPY . /app/

RUN pip install -r /app/requirements.txt

EXPOSE 5000

ENV FLASK\_APP=app.py

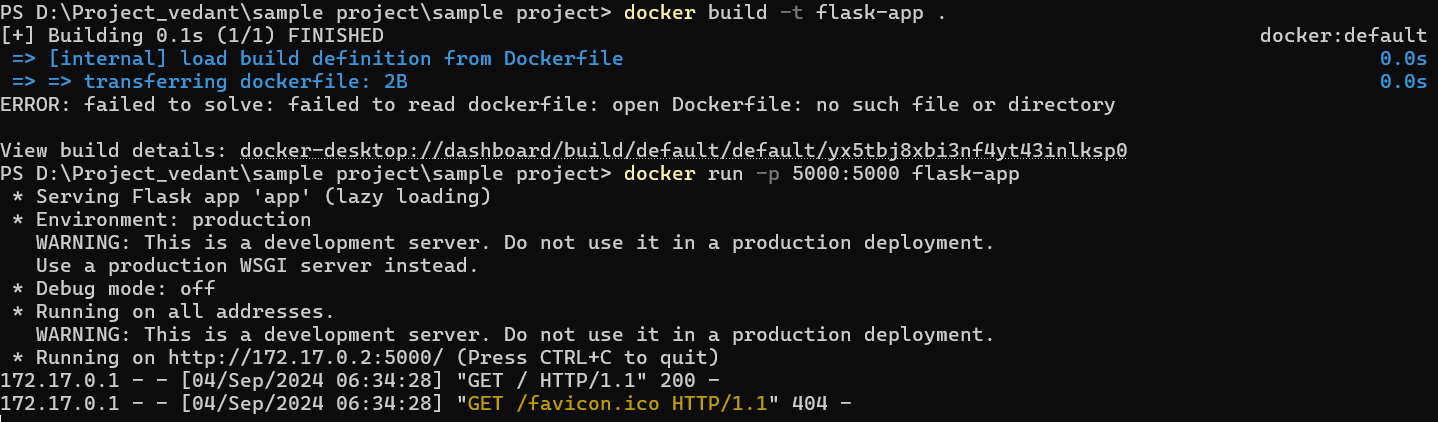
ENV FLASK\_ENV=deployment

ENV $(cat .env | xargs)

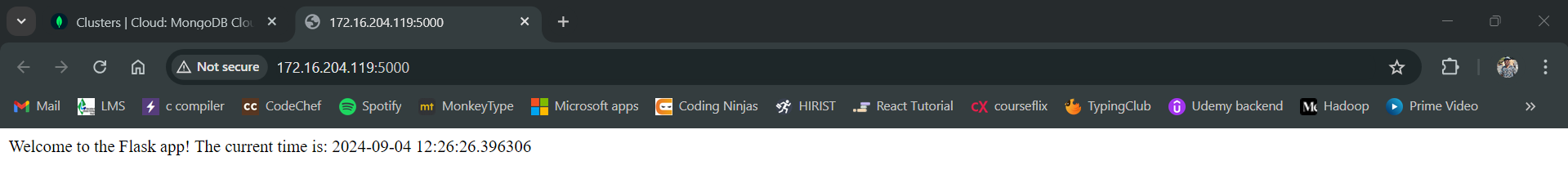
CMD ["python","app.py"]

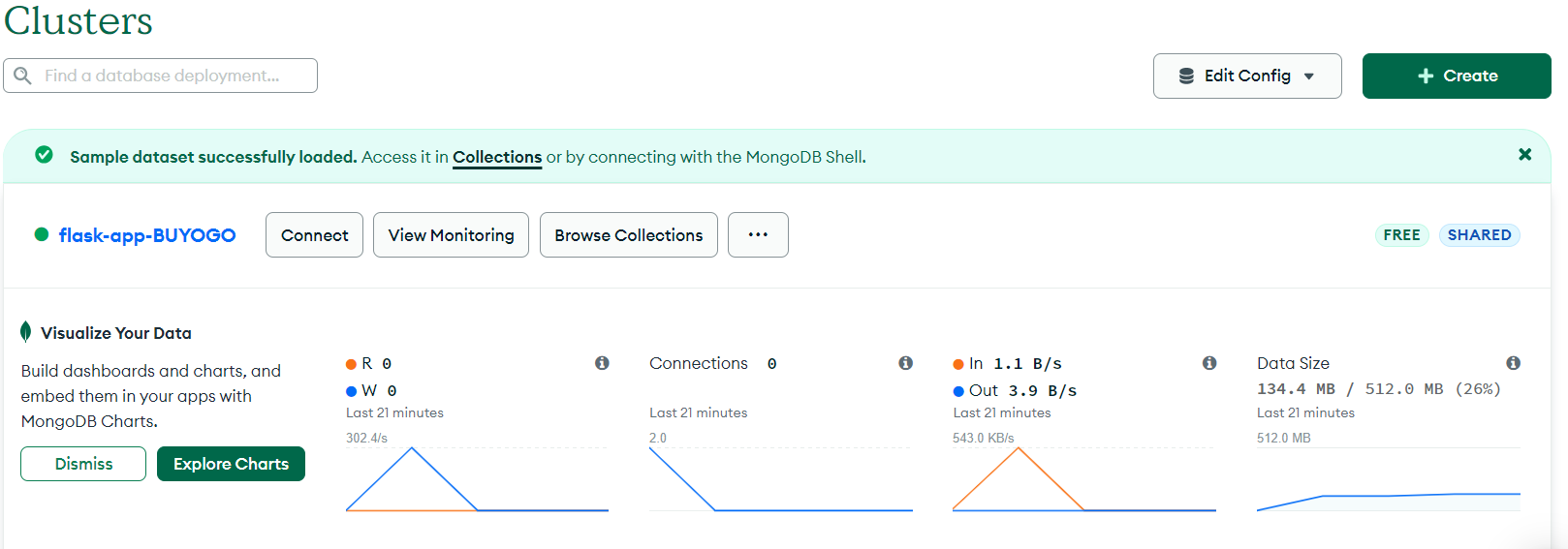
-Then run:

* docker build -t flask-app .
* docker run -p 5000:5000 flask-app
* docker tag flask-app abhishekporwal836/flask-app-buyogo:latest
* docker push abhishekporwal836/flask-app-buyogo:latest



-Application is up and running:

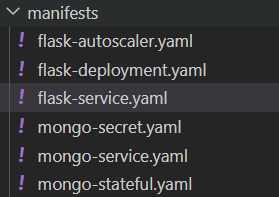


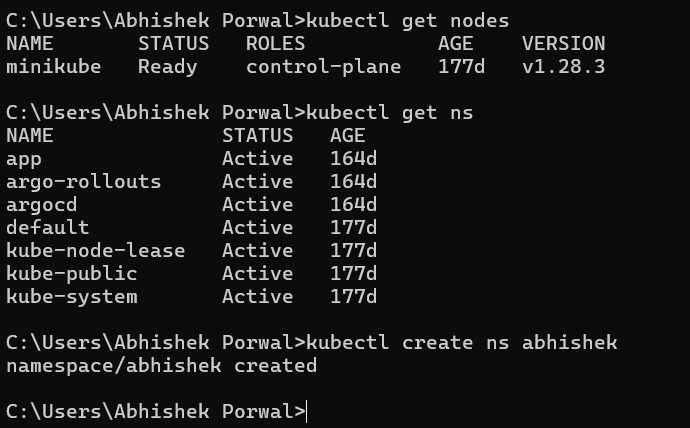


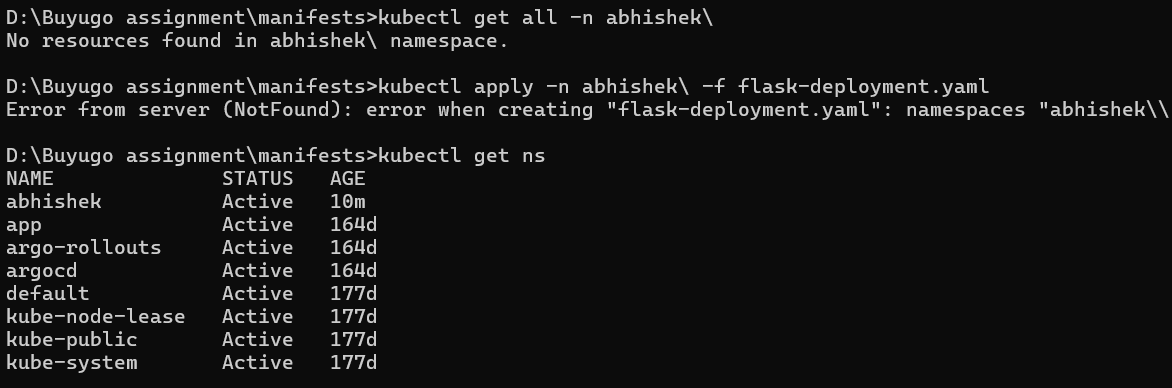
* **Step 2: Manifests for Kubernetes/Minikube:**

Since we are running single node K8’s cluster we will with Minikube on our local system.

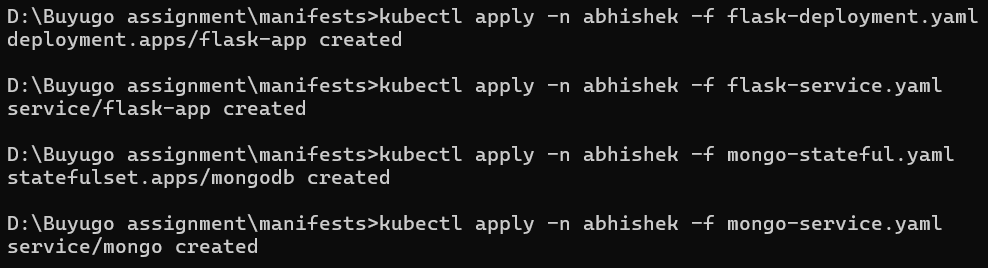
We will configure the files which are there on GitHub :



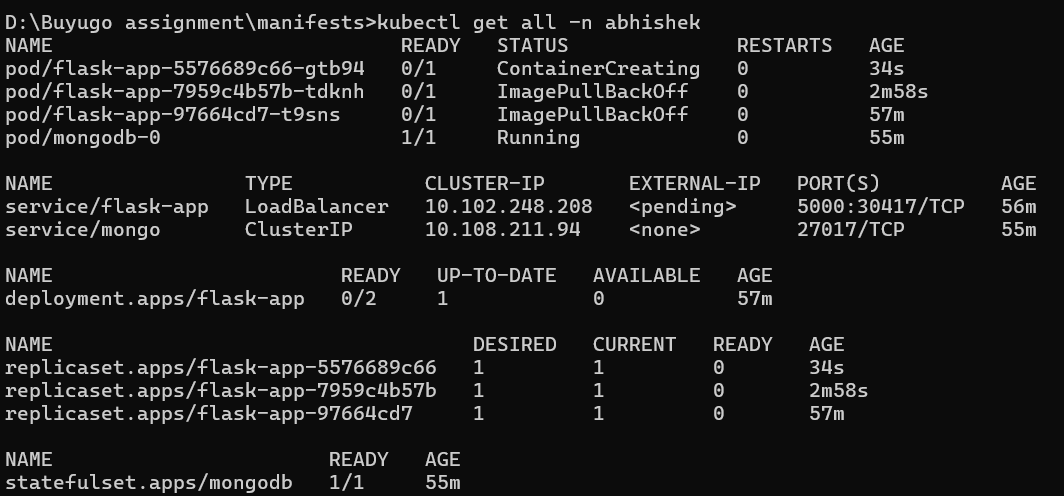
* minikube start
* kubectl get ns
* kubectl create ns abhishek
* 
* kubectl create ns abhishek
* kubectl get ns

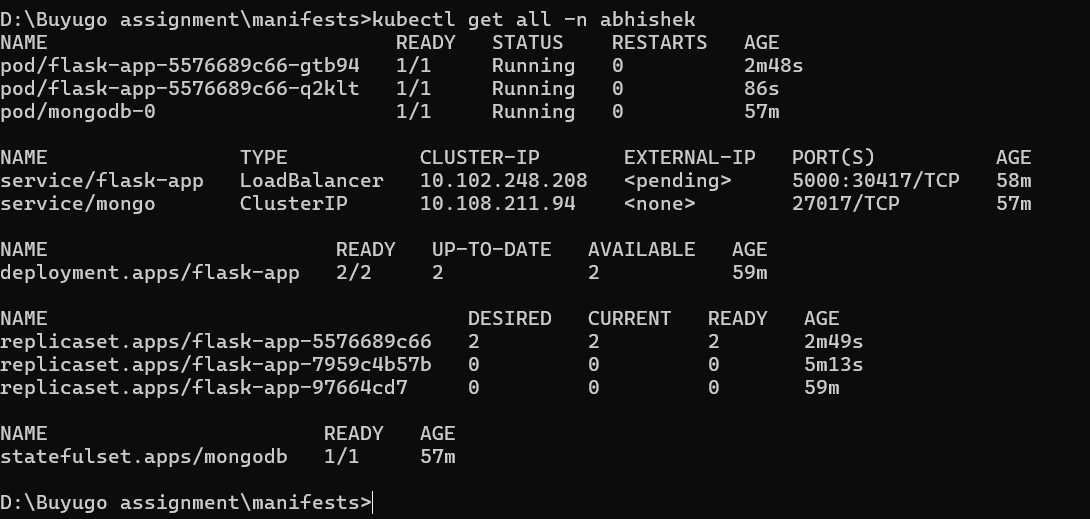


* kubectl apply -n abhishek -f flask-deployment.yaml
* kubectl apply -n abhishek -f flask-service.yaml
* kubectl apply -n abhishek -f mongo-stateful.yaml
* kubectl apply -n abhishek -f mongo-service.yaml



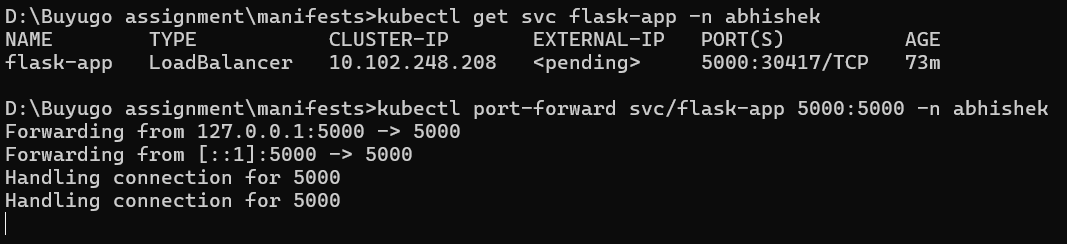
* kubectl apply -n abhishek -f mongo-secret.yaml  
  
* kubectl get all -n abhishek

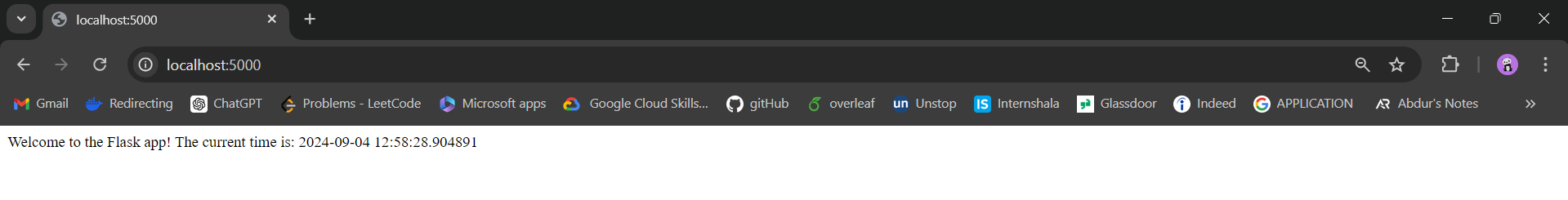


Now all the pods are up and running.  


* Accessing the application:  
  kubectl port-forward svc/flask-app 5000:5000 -n abhishek

Now access on the localhost:5000





Alternatively we can use NodePort. We can connect the external ip to it and access the application from external link.

* AutoScalers:  
  kubectl apply -f flask-autoscaler.yaml