Team19: Project – Chicago Business Hub

Read me:

Instructions:

* Create a docker container for the postgresql
  + docker run -d --name postgres\_docker -v dbdata:/var/lib/postgresql/data -p 54320:5432 -e POSTGRES\_PASSWORD=HaveFun postgres:14 (my pgadmin password is HaveFun)
* Run two python files in Python data preprocessing folder to reverse engineer the zipcode and community area from latitude and longitude and store it in tabular form.
* Run the fb prophet python notebook (file name: fb\_traffic.ipynb) to get the future predicted traffic patterns and save the output in the go folder.
* Install and deploy minikube package
  + brew install minikube(for mac)
  + minikube start
  + minikube status
  + eval $(minikube -p minikube docker-env)
* Create a docker container for the go package module.
  + Create a docker file   
      
     FROM golang:1.16-alpine  
     WORKDIR /app  
     COPY go.mod ./  
     COPY go.sum ./  
     RUN go mod download  
     COPY . /app  
     RUN go build -o /app/fetch\_data   
     EXPOSE 8080  
     CMD [ "/app/fetch\_data" ]
  + Build the module   
     docker build . -t ms/retrievedata
  + This will fetch the data from Chicago data portal and will be saved in postgres
* Create a Kubernetes cluster for the dockerized modules
  + Create a microservice-fetch-data.yaml file  
      
     apiVersion: apps/v1

kind: Deployment

metadata:

name: retrievedata

labels:

app: retrievedata

spec:

replicas: 1

selector:

matchLabels:

app: retrievedata

template:

metadata:

labels:

app: retrievedata

spec:

containers:

- name: app

image: ms/retrievedata

imagePullPolicy: Never

* + Delete any existing kubernetes

kubectl delete -f microservice-fetch-data.yaml

* + deploy the cluster using the following command  
     kubectl create -f microservice-fetch-data.yaml
  + Check the status and logs from the cluster using the following command

kubectl get pods

kubectl logs <pod name>. // Our pod name retrievedata-b57899c65-kj4db

* Go the backend folder and start the backend server using Node.js
  + nodemon - -exec npm start app.js
* Start the frontend server using angular
  + ng s –o

Website will be executing in http://localhost:4200/  
Ensure the PostgresQL server is running, to ensure the application and run all the python files.

* Enjoy the features our website has to offer.

2. How many total lines of code written?  
 Appoximately 5k -6k lines of code.

3. Features and functionalities in our project

* + User Account/Profile/Transaction management -> Yes
  + SQL/NoSQL DB -> PostgreSQL
  + Analytics & Visual Reports                               -> Yes
  + Reviews & Trending & MongoDB -> No
  + Auto-Complete Search feature -> No
  + Google MAPS - Near ME search feature -> Yes
  + Knowledge Graph Searches & Neo4J           -> No
  + Recommender -> Yes
  + Twitter matches -> No
  + Any add-on features? --> Docker, Kubernetes, Golang microservices, Time series forecasting, Pop-up Notification Alert , minikube, Angular chart.js, Angular material dashboard and tables, Reverse engineering of location co-ordinates to zipcodes and community area, Live count of Covid positive and death cases in the dashboard, Choropleth map(in progress).
* What are the Assignments features that are NOT implemented?

Since it’s an option B project so we didn’t implement reviews, mongoDB.

* Any data (xml/database/API Keys) files needed to run your project:
  + Google api key
  + <https://data.cityofchicago.org/> - fetch the datasets