CMPE 281 - LAB #8 - Go Gumball + API Backing Services

In this Lab, you will be deploying a multi-tier Cloud Application with Back-end Services deployed in Docker.

1. Docker Images Needed will include:

```
docker pull golang:latest
docker pull mongo:latest
docker pull rabbitmq:3-management
docker pull kong:0.9.9
docker pull cassandra:2.2
```

2. You are to make **minor modifications** to the code and then **deploy into Docker**

3. Docker Containers:

- Go Code will be an API deployed to a Docker Container using golang:latest as Base Image
- MongoDB will be deployed to a Docker Container
- RabbitMQ will be deployed to a Docker Container
- Kong and Cassandra will be deployed to Docker Containers

4. Container Configurations:

- Node.js App will be Frontend (running on localhost) connected to Kong API Gateway Docker Container
- Go API Server will connect to RabbitMQ and MongoDB
- Kong API Gateway will connect to Cassandra Container (as in Starbucks Lab v3)
- Kong API Gateway will be configured to forward API calls to Go API Container as Upstream Server

Lab Files:

- https://github.com/paulnguyen/cmpe281/tree/master/labs/lab8

Lab Steps:

- **Deploy RabbitMQ and MongoDB Docker Containers**
- **Create a Database "cmpe281" and the Collection "gumball" in MongoDB**
- **Then, Install the following Document:**

```
    db.gumball.insert(
        {
          Id: 1,
          CountGumballs: NumberInt(202),
          ModelNumber: 'M102988',
          SerialNumber: '1234998871109'
        }
    ) ;
```

- **Query the to find the document and Take a Screenshot showing the Output of the Query.**

- **Modify the Go API and Node.js Code. The Go API will connect to RabbitMQ and MongoDB. The Node.js App will connect to the Go API.**

- **For this part, you may test your code changes in localhost (i.e. Go API Docker Deployment not required yet in this part).**

- **For portions of your code changes from Node.js and Go API, explain the purpose of each change.**

- **Update the Go API for deployment to Docker. You will have to Build the Go Image to connect to RabbitMQ and MongoDB. Please note that you must use the Docker Internal Link or Host name and not Localhost or Docker Host IP.**

- **Once successfully deployed, include the list of Docker Run Commands (or Docker Compose Yaml File) in the PDF.**

- **Also run the following two Docker Commands and take a Screenshot of their output to be added to PDF**

```
docker-ps:
    docker ps --all --format "table {{.ID}}\t{{.Names}}\t{{.Image}}\t{{.Status}}\t"

docker-ps-ports:
    docker ps --all --format "table {{.Names}}\t{{.Ports}}\t"
```

- **Place a few Gumball Orders on the Node.js App (note, you'll have to insert a quarter for each order)**

- **View the RabbitMQ "gumball" queue showing the count of messages (1 for each Order)**

- **Run the following CURL command from localhost (i.e. outside the Container). Note, you'll have to replace "dockerhost" with your dockerhost ip.**

```
curl -X POST \
  http://dockerhost:3000/orders \
  -H 'content-type: application/json'
```

- **Deploy the Kong API Gateway**

- **Also run the following two Docker Commands**

```
docker-ps:
     docker ps --all --format "table {{.ID}}\t{{.Names}}\t{{.Image}}\t{{.Status}}\t"

docker-ps-ports:
     docker ps --all --format "table {{.Names}}\t{{.Ports}}\t"
```

- **Configure Kong API Gateway to Go API as upstream server**

- **Add API Key Authentication Plugin to Kong to protect the Go API.**
- **REF: https://getkong.org/plugins/key-authentication/**

- **Include all the REST API calls to Kong for your configuration in PDF (Or Screenshots evidence of Kong Configuration if using a Kong UI Admin Tool)**

- **Make changes to Node.js App to to send API request to Kong instead of Go API Server**

- **Include snippets of Node.js changes in the PDF document**

- **Place some orders on the Node.js App (note, you'll have to insert a quarter for each order)**

- **Look the RabbitMQ "gumball" queue showing the count of messages (1 for each Order)**

- **Run the following CURL command from localhost (i.e. outside the Container). Note, you'll have to replace "dockerhost" with your dockerhost ip and apikey with your Kong API key. Take a screenshot of the output of the command.**

```
curl -X POST \
  http://dockerhost:8000/goapi/order \
  -H 'apikey: 94ee60f882cb45e582c7c7670dee61c9' \
  -H 'content-type: application/json'
```