CMPE 281 - LAB #7 - Go Gumball + MySQL & Kong

Gumball API Docker Containers

In this Lab challenge, you will be deploying the modified version of the Go Gumball API demonstrated in class to your local Docker Host.

Key Tasks Include:

- Installing Docker Images for MySQL, Cassandra and Kong
- Build Modification of Go Gumball REST API
- Deploy your modified Go REST API
- Deploy and Configure Kong API Gateway
- Test Deployment using Postman or Curl

Lab Files:

- https://github.com/paulnguyen/cmpe281/tree/master/labs/lab7

STEP 1 - INSTALL DOCKER IMAGES & SET UP MYSQL DB

Download the following Docker Images Versions:

- mysql v5.5
- cassandra v2.2
- kong v0.9.9

After MySQL Image is install, startup MySQL Docker Image and create the following data in a Schema: (i.e. Connect to MySQL Server inside Docker Container or outside using MySQL Workbench)

```
CREATE TABLE gumball (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  version bigint(20) NOT NULL,
  count_gumballs int(11) NOT NULL,
  model_number varchar(255) NOT NULL,
  serial_number varchar(255) NOT NULL,
  PRIMARY KEY (id),
  UNIQUE KEY serial_number (serial_number)
) ;

insert into gumball ( id, version, count_gumballs, model_number, serial_number )
values ( 1, 0, 1000, 'M102988', '1234998871109' ) ;
```

Take a screenshot of SQL query Output from "gumball" table.

STEP 2 - Modify and Test Go Gumball API

- Download Source Code: **gumball-go.zip**
- Make changes to the file: **src/gumball/server.go** to connect to your MySQL DB
- Build and Test your modification
- Run your Go Gumball API locally

- **Take Screenshots showing the following (in a PDF file):**
  - The Go Command you used to Build your Program
  - The Go Command you used to Run the Program
  - Curl or Postman API call to /gumball and the output

STEP 3 - Build and Run a Docker Image with your Go Gumball API

- **Note the following:**
  - The contents of your **Dockerfile**
  - The Docker Commands you used to build and run your Container

STEP 4 - Deploy and Configure Kong API Gateway.

- **Deploy Kong API Gateway with your Gumball API Stack with the following configuration:**

  - Gumball API Container connects to MySQL Container
  - Kong API Container connects to Cassandra and Gumball Containers
  - Only the Kong API Container maps ports externally.
  - Gumball, MySQL and Cassandra Containers should not map any exposed ports.

- **Configure Kong API Gateway as follows:**

  - Create an API with request path "/goapi" to route to the Gumball API
  - Add the "**File Log**" Plugin to your Kong API with file path of "**/tmp/kong.log**"

- **Note the the following:**

  - **Docker Run Commands and Curl commands for Kong API setup**

HINT: If you are using Postman, you may use the "generate code" feature in Postman to get the Curl command equivalent.

STEP 5 - Test Your Deployment using Postman or Curl.

- **Perform the following API calls (using Curl or Postman) against your Kong API Endpoint:**

  - Ping the API
  - Get Gumball Inventory
  - Update Gumball Machine Inventory
  - Place Two Gumball Orders
  - Get the Order Status of All Orders
  - Process All Gumball Orders

- **Note: The API Calls should be made against the Go Routes (as noted in server.go)**

```
func initRoutes(mx *mux.Router, formatter *render.Render) {
    mx.HandleFunc("/ping", pingHandler(formatter)).Methods("GET")
    mx.HandleFunc("/gumball", gumballHandler(formatter)).Methods("GET")
    mx.HandleFunc("/gumball", gumballUpdateHandler(formatter)).Methods("PUT")
    mx.HandleFunc("/order", gumballNewOrderHandler(formatter)).Methods("POST")
    mx.HandleFunc("/order/{id}", gumballOrderStatusHandler(formatter)).Methods("GET")
    mx.HandleFunc("/order", gumballOrderStatusHandler(formatter)).Methods("GET")
    mx.HandleFunc("/orders", gumballProcessOrdersHandler(formatter)).Methods("POST")
}
```

- **After sending these API calls, create a PDF document to include the following screenshots:**

  - The output of the following docker commands

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

      - *Containers must be up and running*

    - **docker ps --all --format "table {{.Names}}\t{{.Ports}}\t"**

      - *Must show Ports closed for all containers except for the Kong Gateway ports*

  - The complete dump (full screen view) of the contents from "**/tmp/kong.log**" in your Kong Container