Colombian School of Engineering Julio Garavito
Enterprise Architectures

Laboratory 5

Modularization Workshop with Virtualization and Introduction to Docker and AWS

Author: Angi Paola Jimenez Pira
Teacher: Luis Daniel Benavides Navarro

Date: Friday 12th March, 2021
Bogota, Colombia

1 Introduction

The purpose of this workshop is to learn about virtualization using tools as Docker and Services of AWS, developing an architecture composed by containers with instances of differents services.

In this document you will find the explanation of challenge purposed in the laboratory 5, the architecture implemented, the tools used during development of this project and other sections that are important to understand how the project works (how install it, how run it, etc.) and if you want extend the code and implement new functionalities.

2 Objectives

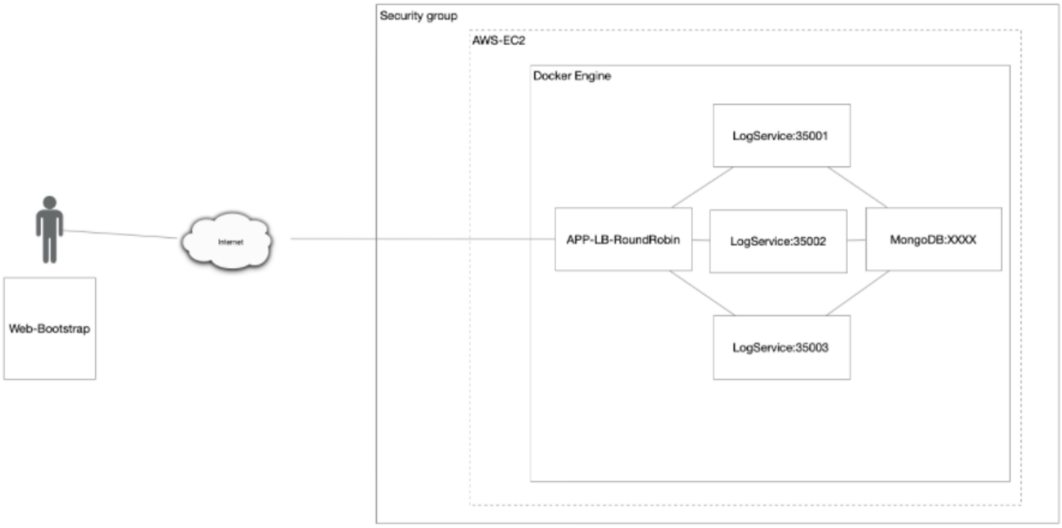
- To develop an architecture using virtualization tools.
- To learn about use of EC2 services of AWS.
- To learn about use of Docker.
- To use known tools like Circleci, Maven, Java, Git and GitHub to develop a web app.

3 Glossary

- **Docker:** Docker is an open source software platform to create, deploy and manage virtualized application containers on a common operating system (OS), with an ecosystem of allied tools.
- **AWS:** Amazon web service is an online platform that provides scalable and cost-effective cloud computing solutions.
- **Virtualization:** Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware. Most commonly, it refers to running multiple operating systems on a computer system simultaneously.
- **EC2:** Amazon Elastic Compute Cloud (Amazon EC2) is a service that provides scalable computing capacity in the Amazon Web Services (AWS) Cloud.
- **Circleci:** CircleCI is a modern continuous integration and continuous delivery (CI/CD) platform.
- **Git:** Git is an open-source version control system that was started by Linus Torvalds—the same person who created Linux.
- **Github:** GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

4 Challenge description

To build an application with the proposed architecture and deploy it in AWS using EC2 and Docker.



- **MongoDB** is an instance of MongoDB running in a docker container. LogService is a REST service that receives a message, save it in the data base and return a JSON object with the 10 last messages saved.

- **LogService** is a REST service that receives a message, save it in the data base and return a JSON object with the 10 last messages saved.

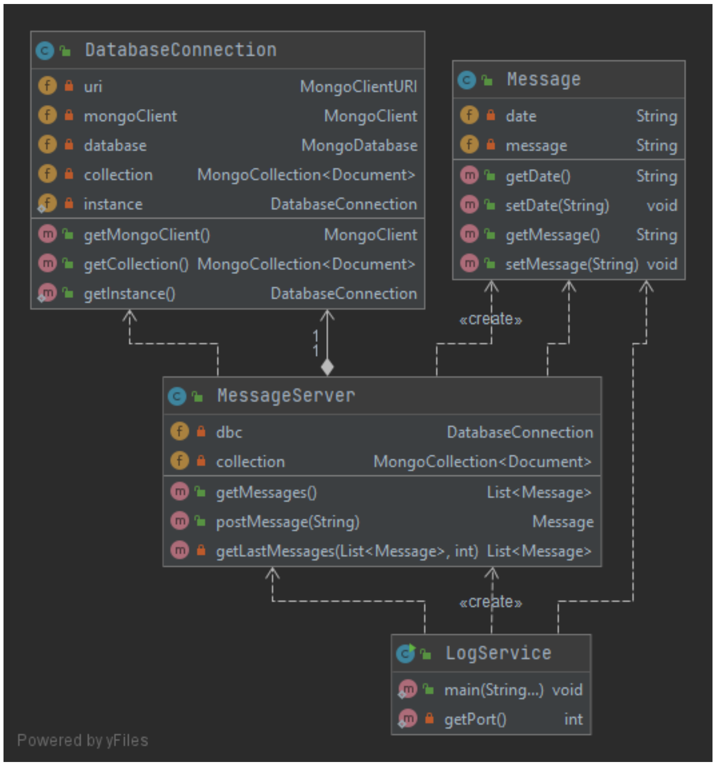
- **APP-LB-RoundRobin** is a application composed by a web client and at least a service REST, the web client contains a field and a button. The user sends a message, it is sent to the REST service and updates the screen with the information that it returns in JSON format. The REST service receives the chain and implements a Round Robin load balancing algorithm, delegating the processing of the message and the return of the response to each of the three instances of the LogService service.

5 Architecture of program

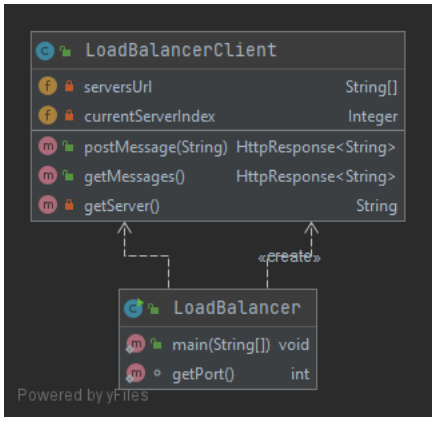
5.1 Class Diagram

The next image shows the class Diagrams of the solutions implemented:

- **LogService:**



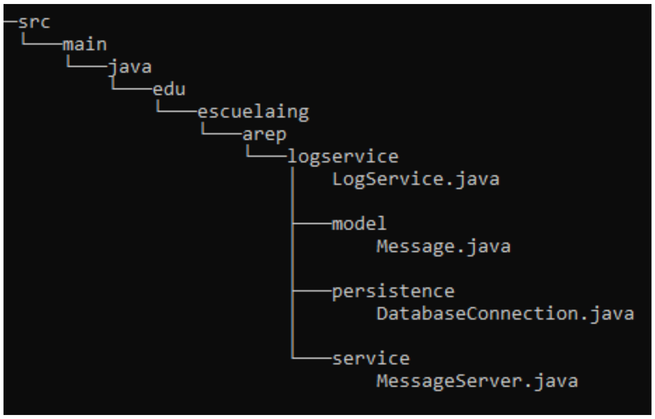
- **Load Balancer:**



5.2 Project Tree

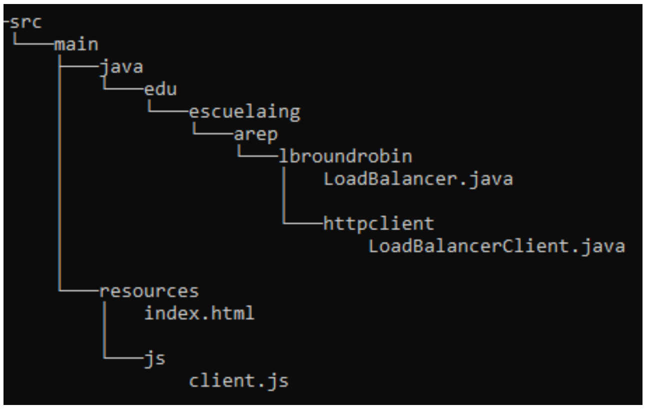
The next image shows the structure of the projects generated by the Maven dependency manager, with the objective of giving more information for the reader about how the solutions were implemented:

- **LogService:**



```
─src
 └───main
     └───java
         └───edu
             └───escuelaing
                 └───arep
                     └───logservice
                         │   LogService.java
                         │
                         ├───model
                         │       Message.java
                         │
                         ├───persistence
                         │       DatabaseConnection.java
                         │
                         └───service
                                 MessageServer.java
```

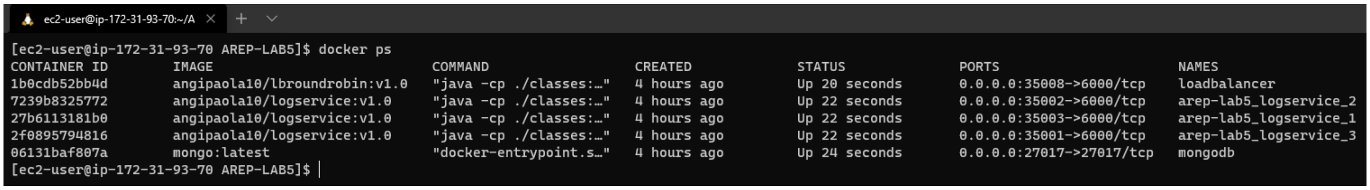
- **Load Balancer:**



```
─src
 └───main
     ├───java
     │   └───edu
     │       └───escuelaing
     │           └───arep
     │               └───lbroundrobin
     │                   │   LoadBalancer.java
     │                   │
     │                   └───httpclient
     │                           LoadBalancerClient.java
     │
     └───resources
         │   index.html
         │
         └───js
                 client.js
```

5.3 Containers

The next image shows the containers created after to run the docker-compose.yml file using the docker-compose command to download images of a DockerHub repository and create the necessary configurations of the containers:



```
[ec2-user@ip-172-31-93-70 AREP-LAB5]$ docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED       STATUS          PORTS                        NAMES
1b0cdb52bb4d   angipaola10/lbroundrobin:v1.0    "java -cp ./classes:…"   4 hours ago   Up 20 seconds   0.0.0.0:35008->6000/tcp      loadbalancer
7239b8325772   angipaola10/logservice:v1.0      "java -cp ./classes:…"   4 hours ago   Up 22 seconds   0.0.0.0:35002->6000/tcp      arep-lab5_logservice_2
27b6113181b0   angipaola10/logservice:v1.0      "java -cp ./classes:…"   4 hours ago   Up 22 seconds   0.0.0.0:35003->6000/tcp      arep-lab5_logservice_1
2f0895794816   angipaola10/logservice:v1.0      "java -cp ./classes:…"   4 hours ago   Up 22 seconds   0.0.0.0:35001->6000/tcp      arep-lab5_logservice_3
06131baf807a   mongo:latest                     "docker-entrypoint.s…"   4 hours ago   Up 24 seconds   0.0.0.0:27017->27017/tcp     mongodb
[ec2-user@ip-172-31-93-70 AREP-LAB5]$ |
```

5.4 Technology Stack

- **Development environment:** Java Development Kit 8, IntelliJ IDEA
- **Dependency Manager:** Maven
- **Version Control:** Git, Github
- **Tests:** JUnit4
- **Deploy and build:** EC2, Circleci, Docker

5.5 Documentation

The documentation and instructions for use are in a git repository: https://github.com/angipaola10/AREP-LAB5.

6 Conclusions

- The use of tools like Circleci to build a web app is very important because this tools help us to check the code and test developed in the program. Also, with this tools we can found bugs, duplicated code and other factors that make our project less efficient.
- In the actually the virtualization is very important because the differents services, components and applications are implemented using those technologies. Docker is a very important tool to start to learn about vitualization and AWS is a great supplier of differents components that allow us build efficient architectures using virtualization.

References

[1] TechTarget. https://searchitoperations.techtarget.com/definition/Docker

[2] SimpliLearn. https://www.simplilearn.com/tutorials/aws-tutorial/what-is-aws

[3] Circleci. https://circleci.com/docs/enterprise/overview/

[4] OpenSource. https://opensource.com/resources/virtualization

[5] OpenSource. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html