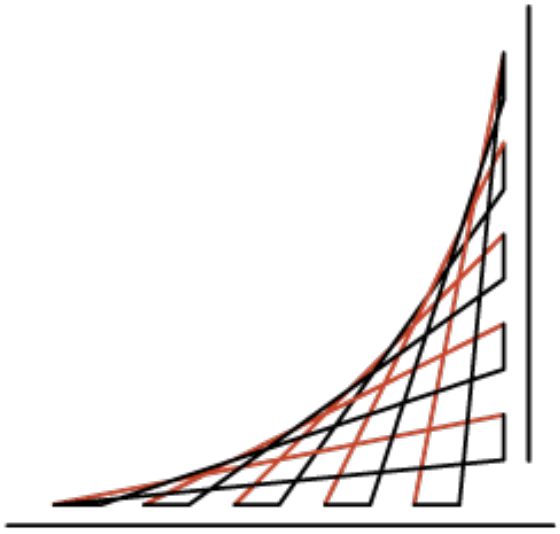

ESCUELA COLOMBIANA DE INGENIERÍA
JULIO GARAVITO

VIGILADA MINEDUCACIÓN

SYSTEMS ENGINEERING

Arquitecturas Empresariales

Laboratory 5

Luis Daniel Benavides Navarro

Author:
Nicolás Aguilera Contreras

Contents

1 Prerequisites **2**

2 Introduction **2**

3 Architecture **2**
3.1 RoundRobinApp 3
3.2 LogService 3
3.3 MongoDB Database 3

4 Results **3**

1 Prerequisites

These are the necessary installations to run the application on your computer:

Docker - Container Manager
Maven - Dependency Management
Java 8 - Development Environment
Git - Version Control System
Spark - Micro framework for creating web applications in Java 8

2 Introduction

The workshop consists of creating a small web application using the Spark java micro-framework (http://sparkjava.com/). Once we have this application we will proceed to build a container for docker for the application and we will deploy and configure them on our local machine. Then, we will close a repository on DockerHub and upload the image to the repository. Finally, we will create a virtual machine on AWS, install Docker, and deploy the container we just created.

3 Architecture

In figure 1 you can see the architecture of the system, first the user communicates with the instance that implements the Round Robin algorithm, said component is in charge of directing the requests and distributing it among the 3 instances of the Log Service, then the component the Log Service that received the task communicates with the database

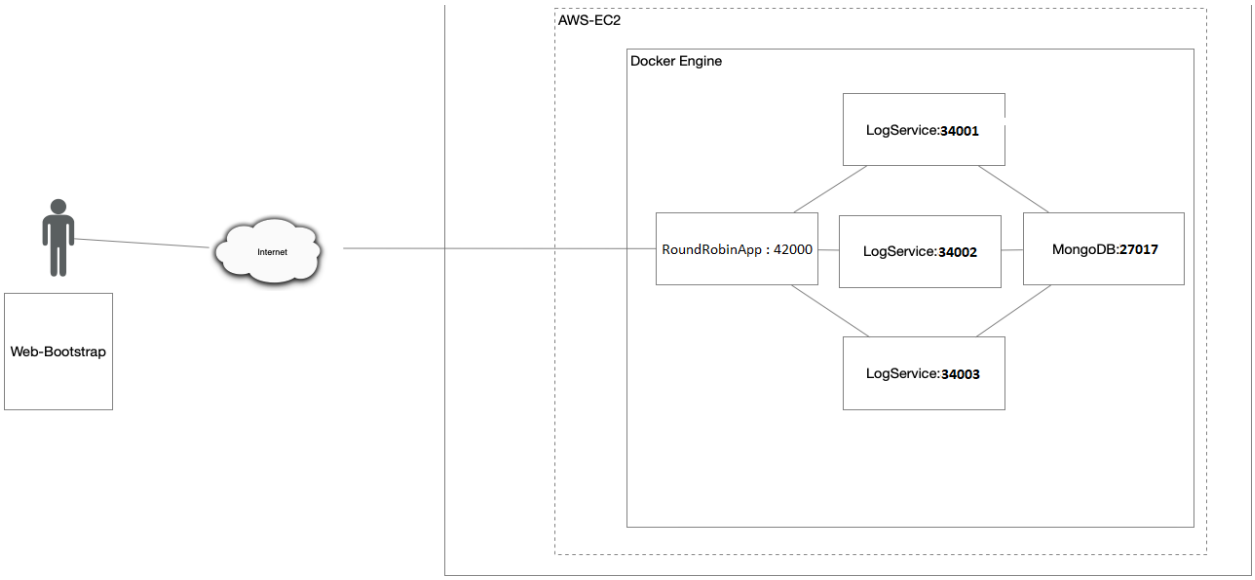

Figure 1: Class diagram

3.1 RoundRobinApp

This component is accessible through port 42000, it offers the GET / lastlogs and POST / addlog endpoits, which redirect the request to the chosen Log Service component.

3.2 LogService

These components are built using the Spark framework, each one offers the endpoits GET / getlogs and POST / insertlog which allow to send a log and get the last 10 stored logs, each component uses a layer of services and persistence to achieve such behavior. The principle of investment of dependencies to implement the communication of the different layers.

3.3 MongoDB Database

This component represents the system database, this being a database MongoDB, said database was built on a docker image where the respective database was created and the collection representing the messages.

4 Results

At the end of this workshop, a system capable of publishing messages was built, the architecture of said system consists of a series of modularized components using Docker, the components of the logic are assigned through a spark web application that simulates a load balancer using the Round Robin algorithm