Introduccion Docker y AWS

Johan Nicolas Cortes Torres
Escuela Colombiana de Ingenieria Julio Garivito
AREP

August 2020

Abstract

En el presente documento se encontrará información acerca de la elaboración de un laboratorio que se basaba en la introduccion al uso de Docker, AWS y MongoDB. El laboratorio consiste en crear una maquina virtual en AWS que ejecute contenedores basados en imagenes de Docker, donde una de esos contenedores debe tener una base de datos en MongoDB

1 Introduction

El laboratorio se divide en tres partes fundamentales. La primera parte trata de la implementacion de una base de datos en MongoDB. La segunda parte trata de crear un servicio REST que concte con la base de datos alojada en mongo y permita añadir y obtener os daos alojada en ella. La tercera parte se base en crear un balanceador de carga que reparte entre tre imagenes de la parte dos, ademas de esto la elaboracion de un cliente que va arealizar las peticiones a las imagenes de la parte 2 y finalmente una pagina web que maneje el Reuqest inicial del cleinte pata despues mostrar y agregar dato a la base de datos.

Un servidor web es un programa que utiliza HTTP (Hypertext Transfer Protocol) para mostrar los archivos como paginas web, en respuesta a las diferentes solicitudes que hace un usuario.

Spark es un simple y expresivo marco web de Java/Kotlin DSL construido para un desarrollo rápido. La intención de Sparks es proporcionar una alternativa para los desarrolladores de Java/Kotlin que quieran desarrollar sus aplicaciones web de la forma más expresiva posible y con un mínimo de complejidad.

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto, que en lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

Amazon Web Services (AWS) es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com.

2 Arquitectura

La arquiectura se basa en la existencia de dos clases principales y dos clases secundarias. Las clases pricipales son: SparkWebServer y Balancer, ambos son servicios REST. Las clases secundarias son: JavaCliente y MongoDb, ambas hechas para realizar petciones a otros servicios. Ademas de estos, el proyecto se basa en archivos docker-compose y Dockerfile, los cuales permite crear imagenes y contenedores en Docker

2.1 Balancedor de carga y serevicio REST

Para la implementacion del servicio REST se uso la clase SparkWebServer, la cual se basaba en Spark para construir el servicio correctamente. Esta clase contiene un cliente llamado JavaClient el cual crea las petiiones al servicio web que esta conectado con la base de datos, y retorna esta inormacion a SparkWebServer para que la muestre al usuario. El balanceador de carga se implemento con una pequeña funcion en el cliente la cual permite de manera lineal asignarle la peticion al servicio REST.

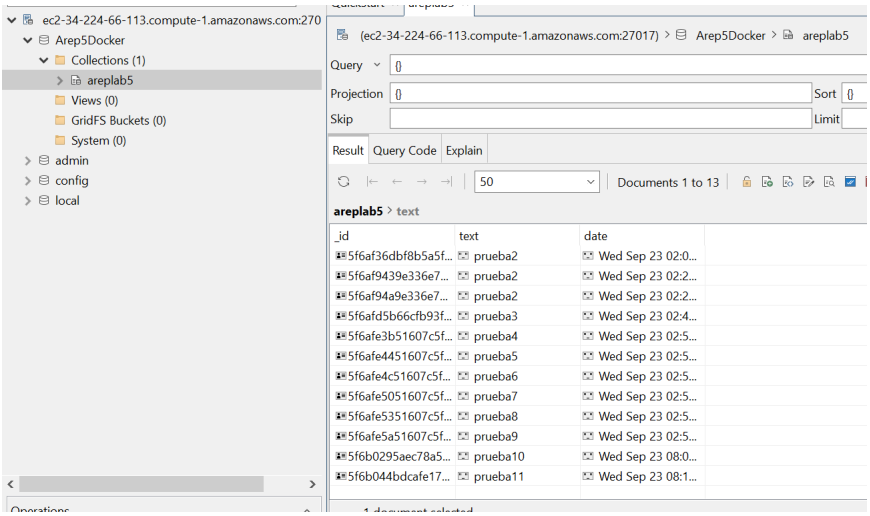
```
public void roundRobin(){
    if (this.actualPort == 2){
        this.actualPort = 0;
    }
    else{
        this.actualPort = this.actualPort+1;
    }
}
```

2.2 Servicio REST hacia la base de datos

El servicio REST que conecta con la base de datos es la clase Balancer, la cual mediante request de un un cliente realiza una peticion a la base de datos alojada

en MongoDB. Para la implementación de a conexion se creo la clase MongoDB, en esta clase se creo una conxion a la base de datos que se va a alojar en link porvisto por AWS.
Los metodos que se implementaron para lograr una conexion exitosa, fueron:

- MongoDB(): Costructor de la clase, que permita crear la conexion a la base de datos.
- add(): Metodo que agrega documentos a la base de datos mongo.
- select(): Metodo que retorna los documentos que existen en a base de datos



2.3 Configuracion Docker

Para la configuracion en Docker se usaron dos archivos esenciales:

- docker-compose.yml: Este archivo permite crear varias imagenes alojadas en un contenedor de Docker, este permitia asignar a cada imagen su servicio, donde se encontraba su contexto y los puertos por los que se iba a ejecutar en Docker.
- Dockerfile: Este archivo permite crear una imagen de una clase especifica, es decir, mediante un clase alojada en el proyecto un puerto y el comando java -cp, nos crea la imagen de Docker para cada clase.

3 Conclusion

Concluyendo se realizó correctamente la implementación completa del laboratorio propuesto por la clase de AREP, se logró realizar la conexion a la base de datos MongoDB, se logro crear un servcio REST que conectra con la base de datos, se logro crear un cleinte que manejara los request al servicio REST anterior mencionado, una pagina web y servicio REST que manipulara la interaccion del usuario y finalemente un balanceador de carga que distribuyea de manera lineal las request del usuario hacia la base de datos.

References

[1] Laboratorio
http://campusvirtual.escuelaing.edu.co/moodle/mod/assign/view.php?id=37113
2020.

[2] Repositorio
https://github.com/jnicolasct/AREP-LAB5
2020.