Docker y AWS

Daniel Felipe Rincón Muñoz Escuela Colombiana de Ingeniería Julio Garavito Facultad de Ingeniería de Sistemas daniel.rincon-m@mail.escuelaing.edu.co

1 Introducción

En este laboratorio desarrollaremos un proyecto web simple, en en cual cada uno de los módulos que lo componen estarán corriendo en contenedores de docker independientes en una máquina virtual AWS, sin embargo, estos módulos se podrán comunicar entre ellos a través de la red local, adicionalmente uno de ellos llamado RoundRobin estará escuchando en la red pública a la espera de peticiones web GET y POST.

2 Diseño

2.1 MongoDB

El contenedor elegido para almacenar la base de datos es la implementación en Docker de la base de datos MongoDB, esta base de datos fué publicada en el puerto 27017 dentro de la red local de los contenedores y es utilizada por las tres instancias de LogService.

2.2 LogService

Se creó un contenedor que corre un programa escrito en java que publica una API REST a la cual se le puede realizar una petición de tipo PUT para agregar un nuevo registro de log a la base de datos, se corrieron tres instancias de este contenedor que estaban escuchando en los puertos 35001, 35002 y 35003 dentro de la red local de los contenedores, estas instancias serán utilizadas por el contenedor que ejecuta el programa RoundRobin, haciendo uso de su API REST.

2.3 RoundRobin

Este módulo contiene un programa escrito en Java que publica una API REST en el puerto 8080, la cual contiene un formulario con un campo, al llenar el formulario y oprimir el botón de enviar, haremos una petición POST al servicio el cual a su vez redirigirá la petición a uno de los servicios de log, y nos retornará los diez últimos logs registrados, incluido el recién enviado, información que a su vez es la respuesta brindada por el LogService.

El algoritmo de RoundRobin simplemente trata de ordenar los servicios de log y elegir uno consecutivamente por cada petición.

A continuación podremos observar un diagrama con la arquitectura de la solución:

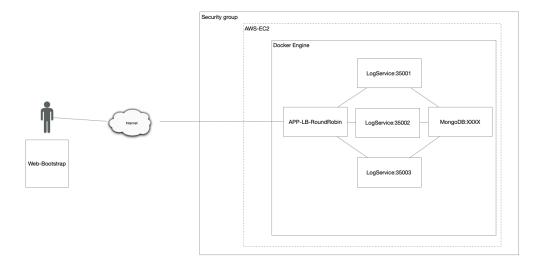


Fig. 1: Arquitectura de la aplicación

3 Conclusiones

- ♦ La creación de contenedores docker con nuestras aplicaciones nos permite ejecutarlas en un entorno controlado que nunca va a cambiar.
- ♦ Los contenedores docker nos permiten la fácil instalación, ejecución y distribución de aplicaciones de todo tipo, además de su intercomunicación gracias a su característica de redes internas.