

Escuela Colombiana de Ingeniería Julio Garavito

TALLER DE DE MODULARIZACIÓN CON VIRTUALIZACIÓN E INTRODUCCIÓN A DOCKER Y A AWS

David Andres Herrera
Profesor: Luis Daniel Benavides Navarro,
Ph.D.

Ingeniería de Sistemas
Bogotá
Marzo del 2021

List of Contents

1	Introducción	II
2	Marco teórico	II
3	Arquitectura	II
4	Conclusiones	III
5	Referencias	III

1 Introducción

Este proyecto trata de una implementación de un servicio para obtener datos y mostrar datos de una base de datos montada en dockers en una máquina de AWS, el objetivo principal de este proyecto es aprender a manejar dockers e imágenes de este servicio, subiendo estas imágenes a dockers que tenemos en máquinas de AWS.

En este documento lo que se quiere lograr es que el lector se de cuenta de la importancia y la utilidad que tiene los contenedores como docker, para poder subir múltiples imágenes de un mismo servicio.

2 Marco teórico

- **Servidor Web:** Un servidor web es una pieza de software para la comunicación y la pieza fundamental para establecer una comunicación entre el servidor donde están alojados los datos y el computador del cliente, esto permite una conexión de tipo direccional o bidireccional y síncrona o asíncrona. Este servicio ayuda a que lo que está en el servidor de datos se pueda mostrar y generar de manera correcta para que sea más fácil la lectura en el computador del cliente.
- **Docker:** es una tecnología de creación de contenedores que permite la creación y el uso de contenedores de Linux. Además Docker tiene la capacidad de ejecutar varios procesos y aplicaciones por separado para hacer un mejor uso de su infraestructura y, al mismo tiempo, conservar la seguridad que tendría con sistemas separados.

Una imagen de docker es una plantilla que se sube al contenedor y se puede correr de manera independiente del resto.

Es un archivo de configuración que se utiliza para crear imágenes. En dicho archivo indicamos qué es lo que queremos que tenga la imagen, y los distintos comandos para instalar las herramientas.

- **AWS:** Un POJO es el nombre que se utiliza para enfatizar en el uso de clases que no dependen de un framework.

3 Arquitectura

La arquitectura de esta solución está compuesta por tres directorios: demo, httpserver y miniSpring.

- El directorio demo cuenta con la clase que se encarga de las anotaciones y devolver la consulta que tenga cada anotación.
- El directorio httpserver cuenta con dos clases que se encargan de crear el servidor http que se va a usar para el miniSpringboot.
- El directorio miniSpringBoot cuenta con las clases necesarias que reciben la solicitud del servidor http y llama a la clase de las anotaciones para consultar que es lo que tiene la anotación, si la anotación no se encuentra esta clase manda un error diciendo que la anotación que se busca no está.

4 Conclusiones

La practica del laboratorio ayuda a obtener un conocimiento basico de como utilizar Heroku y como desplegar la aplicacion web en heroku de manera local y remota, ademas de esto se aprende a hacer la integracion continua con CircleCi

Al consttuir nuestro miniSpring nos damos cuenta de como es el funcionamiento basico de este framWork y de la ayuda que este frameWork le da al desarroyador para facilitar la creacion de la aplicacion web.

5 Referencias

- Caules, C., 2021. ¿Qué es Spring Boot?. [online] Arquitectura Java. Available at: <<https://www.arquitecturajava.es-spring-boot/>> [Accessed 26 February 2021].
- Mirandas. 2021. Spring vs Spring Boot: Cual es la diferencia?. [online] Available at: <<https://www.mirandas.work/es/posts/spring-vs-spring-boot-cual-es-la-diferencia/>> [Accessed 26 February 2021].