

ARQUITECTURAS DE SERVIDORES DE APLICACIONES

Nikolai Bermudez Vega



Taller de de modularización con virtualización e Introducción a Docker y a AWSAssignment

Escuela Colombiana de Ingeniería Julio Garavito
Luis Daniel Benavides Navarro
03/12/2021

1. Introducción

El objetivo de este laboratorio es implementar un aplicativo web con el framework de spark para java, el cual enlazaremos con un contenedor docker. Luego crearemos un repositorio en DockerHub y subiremos dicha imagen al repositorio. Finalmente con una maquina AWS, instalaremos Docker y desplegaremos el contenedor creado anteriormente para poder utilizarlo.

2. Definiciones

2.1. Socket

Un socket es conocido como un tipo de software que actua como un punto final que funciona estableciendo un enlace de comunicacion de red bidireccional entre el extremo del servidor y el programa receptor del cliente..[1].

2.2. Servidor Web

El servidor web es el software que se encarga de despachar el contenido de un sitio web al usuario [2].

2.3. Get

El metodo GET solicita un recurso del servidor indicado en el campo URI. Si la URI apunta a una base de datos de produccion de recursos como un servlet, los datos seran devueltos dentro del mensaje de respuesta [3].

2.4. Post

El metodo HTTP POST envia datos al servidor. El tipo del cuerpo de la solicitud es indicada por la cabecera Content-Type [4].

2.5. Docker

El software de TI, es una tecnologia de creacion de contenedores que permite la creacion y el uso de contenedores de Linux.

2.6. MongoDB

Es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexacion avanzado

3. Diseño y Arquitectura

El aplicativo web cuenta con un balanceador de carga y tres nodos, el balanceador se encargara de enviar las peticiones a los distintos nodos mediante el algoritmo de balanceo de cargas de Round Robin. Estos nodos se conectaran a una base de datos mongo para poder obtener los datos y retornarlos. Cabe destacar que toda esta arquitectura esta en contenedores de docker y se comunican con la base de datos mongo que fue creada a partir de una imagen base de docker.

Para una mejor implementacion, se desplego el aplicativo en una maquina virtual EC2 en AWS, utilizando como referencias las imagenes de los repositorios creados en DockerHub.

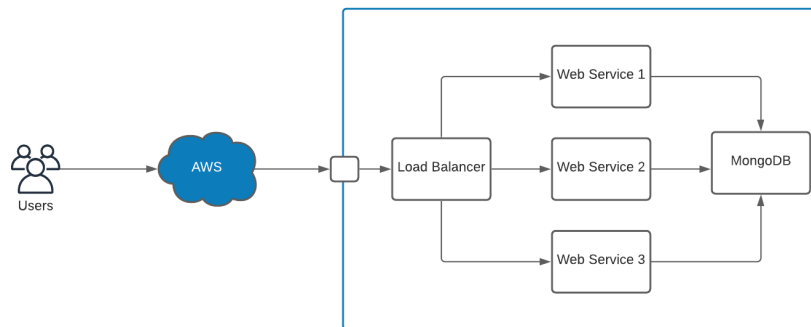


Figura 1: Representacion de la arquitectura

4. Pruebas

Verificando que el aplicativo web funciona correctamente, procedemos a mostrar algunos archivos principales como el index.html para probar que la solicitud get funciona correctamente al consultar en la base de datos y añadir un nuevo dato para probar que la petición post añade correctamente los datos.

4.1. Localmente

Se procedera a mostrar las pruebas locales que se hicieron al aplicativo web. En ellas encontraremos distintas imágenes con la respectiva descripción de cada prueba y el resultado obtenido. En la figura 2 se puede evidenciar el aplicativo funcionando correctamente sobre el puerto 4568 y el contenido de la base de datos

Agrega Mensajes

MENSAJE:

Mensaje	Fecha
Suescaniano	Tue Mar 16 21:07:11 COT 2021
Europa	Tue Mar 16 21:06:07 COT 2021
Prueba post	Tue Mar 16 20:58:31 COT 2021
Hola es una prueba	3/16/2021

Figura 2: Aplicativo web funcionando correctamente de forma local

4.2. AWS

Se procedera a mostrar las pruebas que se hicieron al aplicativo web sobre AWS. En ellas encontraremos distintas imágenes con la respectiva descripción de cada prueba y el resultado obtenido. En la figura 6 se puede evidenciar el aplicativo funcionando correctamente sobre el puerto 10000 y el contenido de la base de datos.

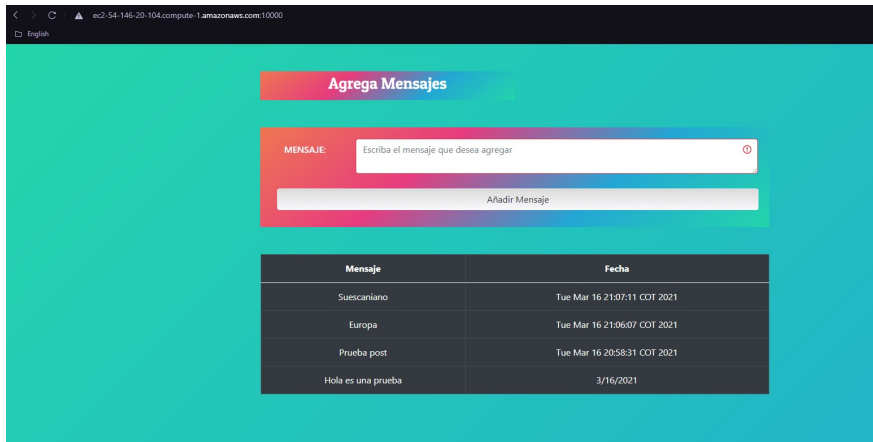


Figura 3: Aplicativo web funcionando correctamente en AWS puerto 10000



Figura 4: Aplicativo web funcionando correctamente en AWS puerto 10000 /mensajes

5. Conclusiones

Con las herramientas que nos ofrece el lenguaje de programación Java, es posible implementar un servidor web con un balanceador que se encargue de enviar las respectivas solicitudes a los nodos. Docker es una herramienta mediante la cual podemos configurar proyectos de manera local muy fácil y luego utilizarlos en cualquier máquina que tenga incorporada esta tecnología.

Referencias

- [1] Socket, <https://www.speedcheck.org/es/wiki/socket/fn1>, Accessed on 2020-09-03.
- [2] Servidor, <https://blog.infranetworking.com/servidor-web/>
- [3] GET, <https://sites.google.com/site/conceptoprogramacion/>
- [4] POST, <https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST>