

SYSTEMS ENGINEERING

Arquitecturas Empresariales

Workshop 5

Luis Daniel Benavides Navarro

Author:
Juan Camilo Angel Hernandez

Contents

1	Introducción	2
2	Arquitectura	2
2.1	Balanceador de carga	3
2.2	Componente logica de negocio	3
2.3	Base de datos MongoDB	3
3	Resultados	3

1 Introducción

En la actualidad las infraestructuras TI basadas en la nube están presentes en una gran variedad de sistemas a lo largo del mundo, esto se debe principalmente al ofrecer grandes capacidades de computo a precios accesibles. Sin embargo cualquier sistema tiende a aumentar su complejidad con el tiempo lo que impacta en la administración de dichos sistemas, una solución frente a este comportamiento es la modularización y virtualización.

Usando contenedores mediante Docker estaríamos virtualizando la infraestructura mediante el uso de máquinas virtuales.

En este taller se construirá una aplicación para el envío de mensajes usando un balanceador de carga, 3 aplicaciones web replicadas las cuales implementan la lógica de negocio, y una base de datos MongoDB, cada componente mencionado anteriormente representa todo el sistema el cual estará desplegado en una instancia EC2 de AWS, dichos componentes corresponden a un contenedor Docker.

2 Arquitectura

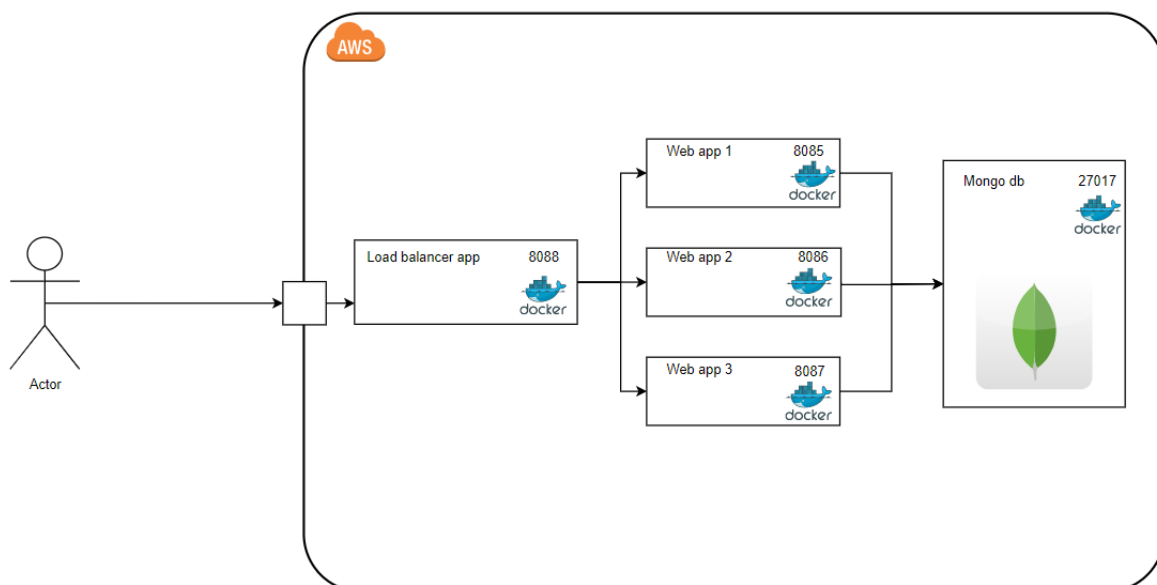


Figure 1: Arquitectura del sistema

En la figura 1 se puede observar la arquitectura del sistema, primeramente el usuario se comunica con la instancia del balanceador de carga, dicho componente es el encargado de dirigir las peticiones hacia los componentes que implementan la lógica, este comportamiento se realiza según el algoritmo round robin, posteriormente el componente de la lógica se comunica con la base de datos.

2.1 Balanceador de carga

Este componente es accesible mediante el puerto 8088, ofrece los endpoints GET /mensajes y POST /mensajes, los cuales redirigen la petición al componente de la lógica correspondiente.

2.2 Componente lógica de negocio

Estos componentes están contruidos usando el framework Spark, cada uno ofrece los endpoints GET /mensajes y POST /mensajes los cuales permiten enviar un mensaje y obtener los mensajes almacenados respectivamente, cada componente utiliza una capa de servicios y persistencia para lograr dicho comportamiento. Se utilizó el principio de inversión de dependencias para implementar la comunicación de las distintas capas.

2.3 Base de datos MongoDB

Este componente representa la base de datos del sistema, siendo esta una base de datos MongoDB, dicha base de datos fue construida sobre una imagen de docker donde se creo la base de datos respectiva y la colección que representa los mensajes.

3 Resultados

Al final de este taller se construyó un sistema capaz de publicar mensajes, la arquitectura de dicho sistema consiste en una serie de componentes modularizados mediante Docker, los componentes de la lógica son asignados mediante un componente de balanceador de carga.