

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

AREP Arquitectura Empresarial

Paula Andrea Guevara Sánchez¹

¹ Ingeniería de sistemas, Escuela colombiana de ingeniería Julio Garavito, Bogotá, Colombia

Fecha: 11/03/2021

Resumen— Este documento es la presentación del desarrollo de una guía que consiste en crear una aplicación web usando el micro-framework de Spark java, posteriormente se construirá un contenedor para docker donde se despegará el proyecto y se configurará en la máquina local; Finalmente se creará una máquina virtual en AWS donde se despegará lo previamente construido.

Palabras clave— Java, Maven, GitHub, Spark, AWS, Docker

Abstract— This document is the presentation of the development of a guide that consists in creating a web application using the Spark java micro-framework, then a docker container will be built where the project will be deployed and configured in the local machine; finally a virtual machine will be created in AWS where the previously built will be deployed.

Keywords— Java, Maven, GitHub, Spark, AWS, Docker

ARQUITECTURA DESPLEGADA EN AWS

INTRODUCCIÓN

El Taller que se desarrolla durante este documento tiene énfasis en la creación de una aplicación web usando Spark Java a través de contenedores e imágenes diseñadas y desplegadas a través de docker, que posteriormente se desplegarán en una máquina virtual creada en AWS. se crearán tres instancias, la primera ofrece un servicio de base de datos con MongoDB, la segunda ofrecerá un servicio de logService en un servicio REST y en el último se creará una aplicación que implementa un balanceo de carga del algoritmo Round Robin, delegando el procesamiento del mensaje y el retorno de la respuesta a cada instancia del servicio.

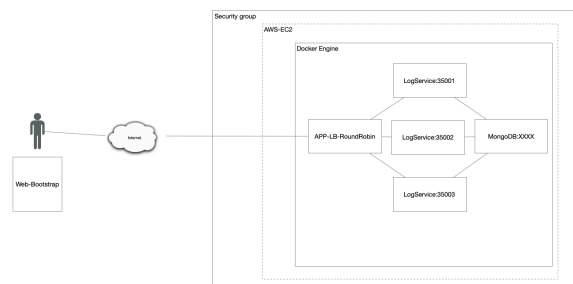


Fig. 1: Arquitectura propuesta y desplegarla en AWS usando EC2 y Docker.

ARQUITECTURA DOCKER

La arquitectura Docker es una arquitectura cliente-servidor, donde el cliente habla con el servidor (que es un proceso daemon) mediante un API para poder gestionar el ciclo de vida de los contenedores y así poder construir, ejecutar y distribuir los contenedores.

El hecho de que el cliente se comunique con el servidor mediante el API hace que el cliente y servidor puedan estar en la misma máquina comunicándose mediante sockets de UNIX o bien en máquinas diferentes comunicándose mediante un end-point en la red.

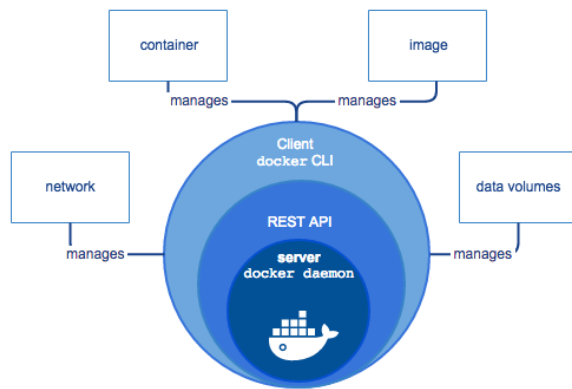


Fig. 2: Arquitectura Docker

DOCKER ENGINE

El Docker Engine es la aplicación cliente-servidor que implementa Docker. Esta aplicación tiene tres componentes:

- Servidor, es el proceso principal de Docker y que funciona como proceso demonio del sistema. Es el encargado de gestionar los objetos que hay en Docker como imágenes, contenedores, redes y volúmenes. Se representa mediante el comando `dockerd`.
- API Rest, es un API Rest que nos permite acceder a las capacidades del servidor y ejecutar comandos sobre él. Podemos utilizar un simple `curl` para acceder a las capacidades del API de Docker
- Cliente, es la línea de comandos representada por el comando `docker`. El cliente habla vía el API Rest para poder ejecutar los comandos. Es lo que utilizaremos para poder ir gestionando el ciclo de vida de nuestras imágenes y contenedores.

Ventajas

- Modularidad Se centra en tomar una parte de la aplicación para actualizarla o repararla, basado en microservicios; Simi-

lar al funcionamiento de la arquitectura orientada al servicio.

- Control de versiones de imágenes y capas Las capas se combinan en una sola imagen; reutiliza las capas para la construcción de nuevos contenedores. Si hay cambios intermedios se comparte entre imágenes, mejorando la velocidad, el tamaño y la eficiencia. El control de versiones es inherente a la creación de capas.
- Restauración Capacidad de restaurar a una versión anterior. Compatible con un enfoque de desarrollo ágil y permite hacer realidad la integración e implementación continuas.

MONGODB

MongoDB es una base de datos de código abierto que utiliza un modelo de datos orientado a documentos y un lenguaje de consulta no estructurado. Es uno de los sistemas y bases de datos NoSQL más potentes que existen en la actualidad. La unidad básica de datos en esta base de datos consiste en un conjunto de pares clave-valor. Permite que los documentos tengan diferentes campos y estructuras. Esta base de datos utiliza un formato de almacenamiento de documentos llamado BSON, que es un estilo binario de JSON. El modelo de datos que sigue MongoDB es altamente elástico que le permite combinar y almacenar datos de tipos multivariados sin tener que comprometer las poderosas opciones de indexación, el acceso a los datos y las reglas de validación.

Arquitectura MongoDB

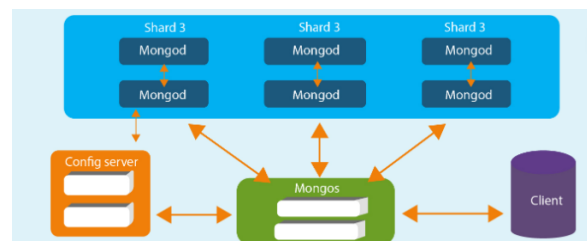


Fig. 3: Arquitectura MongoDB

AMAZON EC2

Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes y administrar el almacenamiento. Amazon EC2 le permite escalar hacia arriba o hacia abajo para controlar cambios en los requisitos o picos de popularidad, con lo que se reduce la necesidad de prever el tráfico.

PRUEBAS

Se realizaron pruebas por medio de JUnit, para asegurar el funcionamiento correcto, para correr las pruebas se realiza la ejecución por medio de consola, haciendo uso de maven con el siguiente comando : mvn test

CONCLUSIÓN

Con los conceptos previos adquiridos, se crea un proyecto maven con java, donde se corre un servicio de una base de datos usando mongoDB que puede ser consultada, una forma de conexión por REST y una pequeña aplicación de balanceo de carga; todo esto previamente cargado en una imagen que se encuentra pública en docker y montadas en una máquina virtual de AWS linux.

REFERENCIAS

- [2] Benavides(2021). INTRODUCTION TO COMPLEX SYSTEMS, JAVA, MVN, AND GIT. Arquitectura empresarial, Bogotá, Colombia.
- [2] Benavides (2010). Servicios Y Virtualización. <https://campusvirtual.escuelaing.edu.co/moodle/pluginfile.php/62258/modresource/content/0/ServiciosYVirtualizacion.pdf>.
- [3] (2008). aws amazon. <https://docs.aws.amazon.com/es/quickstart/latest/mongodb/architecture.html>
- [4] <https://docs.aws.amazon.com/es/AWSEC2/latest/UserGuide/concepts.html>
- [5] <https://intellipaat.com/blog/what-is-mongodb/>
<https://intellipaat.com/blog/what-is-mongodb/>