Laboratorio Docker y AWS

Daniel Felipe Gomez Suarez

September 2020

1 Introduccion

Este laboratorio se enfoca en la arquitectura de un sistema montado en la nube, para ello requerimos conocimientos en AWS y Docker dentro del servicio en la nube, El framework fue spring.

La aplicación consiste en un sistema de envió de mensajes, donde se manejan balanceadores de carga para una mayor eficiencia de la aplicación. estos destalles se describirán de forma más específica en la sección de Diseño

1.1 Amazon Web Service

Figure 1: aws

Hoy en día, Amazon Web Services proporciona una plataforma de infraestructura escalable, de confianza y de bajo costo en la nube que impulsa cientos de miles de negocios de 190 países de todo el mundo. Con centros de datos en Estados Unidos, Europa, Brasil, Singapur, Japón y Australia, tenemos clientes de todos los sectores que disfrutan de los siguientes beneficios

1.1.1 Contenedor EC2

Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable en la nube de Amazon Web Services (AWS). 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.

Dentro de este contenedor tenemos una máquina virtual LINUX la cual nos permite tener un ambiente limpio donde allí utilizamos Docker para el despliegue del proyecto.

La conexión a este contendor lo hacemos de forma remota por SSH con una "clave" o un archivo .Perm que nos permite realizar la conexión de forma remota al servicio por el puerto 22 el cual es el único que tiene habilitado inicialmente

1.2 Docker

Con DOCKER, puede usar los contenedores como máquinas virtuales extremadamente livianas y modulares. Además, obtiene flexibilidad con estos contenedores: puede crearlos, implementarlos, copiarlos y moverlos de un entorno a otro, lo cual le permite optimizar sus aplicaciones para la nube.

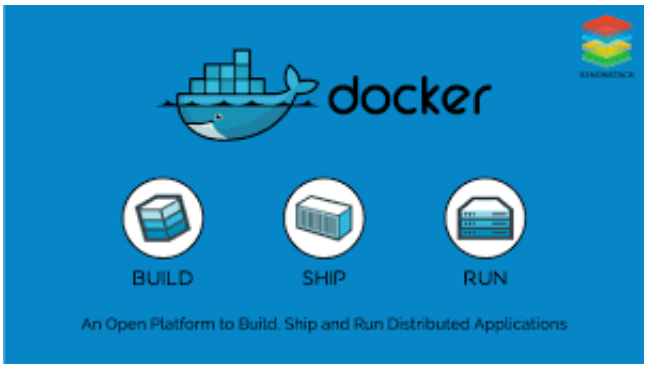

Figure 2: DOCKER

Los contenedores que usamos en docker:

- WEB
- BALANCEADOR DE CARGA
- PERSINSTENCIA

2 Diseño

Este modelo es el propuesto para el trabajo, donde todos los servicios están desplegados en AWS en un contenedor EC2 como se mencionó anteriormente, para esto lo primero que tenemos que hacer es tener la instancia del EC2 habilitada y los puertos que se requieren para el despliegue, es decir los puertos mapeados en el Docker, estos puertos serán los usados para acceder a cada uno de los microservicios.

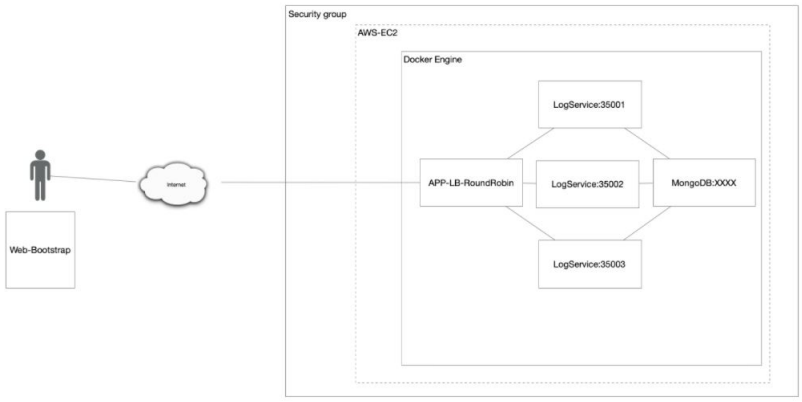

Figure 3: Modelo propuesto

- APP-LB (p 8091): este item hace referencia a la aplicación la cual el usuario interactúa, este servicio es un servicio web el cual únicamente se compone de un framework SPARK. en este servicio el usuario enviará un mensaje el cual será recibido posteriormente por los balanceadores de carga, los cuales hacen referencia a los servicios log que se ven en el diagrama. la comunicación de estos servicios es bidireccional ya que los servicios log también le envían mensajes a ESTE SERVICIO el cual contiene los mensajes que han intercambiado y se han almacenado en la Base de Datos.

- BALANCEADOR DE CARGA (p 8087-8089): Este Servio recibe la información de acuerdo con el puerto en el que se encuentre, esto funciona mediante una función la cual cambia el puerto cada que finaliza una acción. con la información recibida la envía a la base de datos cuando agrega un mensaje, y envía los datos con los mensajes una vez el servicio web haga el REQUEST.

- PERSINSTENCIA (p 27017): Este servicio es un servicio de base de datos

no-SQL, es una base de datos MONGO. Allí se almacena toda la información que recibe del servicio Log o de los balanceadores de carga.

3 Conclusion

Durante el desarrollo del laboratorio nos encontramos con diferentes inconvenientes los cuales partían desde el manejo de los puertos al despliegue en aws principalmente. en algunas ocasiones los puertos parecieran no estar funcionando a pesar de estar bien configurados en el archivo DOCKER-COMPOSE el cual contiene los puertos que se van a mapear con su respectivo puerto del Docker.

una vez solucionados los problemas de los puertos fue más sencillo subir las imágenes a AWS ya que contando con una buena configuración local, nos garantizaba que con las imágenes en dockerHUb de cada servicio, un despliegue más sencillo en AWS ya que era solo poner a correr los servicios en los respectivos puertos y habilitar estos puertos, los cuales no están por defecto en las reglas de salida de AWS.

Podemos concluir que con un adecuado manejo de los puertos y de los servicios el despliegue se hace sencillo, contando con un buen modelo de los servicios que se van a consumir. Amazon Web Service y su contenedor EC2 el cual es económico desde un punto de vista económico, nos permite con Docker maneja diferentes servicios en un ambiente Linux hacer el despliegue de una aplicación con varios microservicios de forma Estructurada

References

[1] definicion AWS
https://aws.amazon.com/es/ 2020.

[2] contenedor EC2
https://docs.aws.amazon.com/eses/AWSEC2/latest/UserGuide/concepts.html
2020.

[3] QUE ES Mongo
https://www.mongodb.com/es/what-is-mongodb
2020.

[4] que es docker
https://www.redhat.com/es/topics/containers/what-is-docker
2020.

[5] Especificaciones
http://campusvirtual.escuelaing.edu.co/moodle/mod/assign/view.php?id=37113
2020.