Modularización con virtualización e Introducción a Docker y a AWS

Carlos Amorocho

Marzo 2021

1 Introduction

En el siguiente laboratorio vamos a realizar un montaje de un servicio con ayuda de dockers, mediante docker hub y una máquina virtual de AWS, crearemos 3 dockers para el servicio de log service y nuestro round robin.

2 Objetivo

Implementar un programa con ayuda de docker para el log service y ayuda de AWS para tener un servicio montado.

3 Marco Teórico

3.1 AWS Educate

AWS Educate es una iniciativa global de Amazon cuyo objetivo es proveer a los estudiantes recursos integrales para desarrollar habilidades vinculadas con la nube. Es un programa sin costo mediante el cual se ofrece acceso a contenido, formaciones, itinerarios y servicios de AWS. [3]

3.2 Circle CI

Es una plataforma de integración continua en la nube que nos facilita realizar pruebas untarias para el correcto funcionamiento sobre la aplicación. [4]

3.3 Integración continua

La integración continua ayuda a que los desarrolladores fusionen los cambios que introducen en el código para incorporarlos a una división compartida con más frecuencia. Una vez que se fusionan los cambios implementados por un desarrollador en una aplicación, se validan con el desarrollo automático de la aplicación y la ejecución de distintos niveles de pruebas automatizadas para verificar que los cambios no hayan dañado la aplicación. [2]

3.4 Servidor HTTP

En cuanto a hardware, un servidor web es una computadora que almacena los archivos que componen un sitio web (ej. documentos HTML , imágenes, hojas de estilos CSS y archivo JavaScript) y los entrega al dispositivo del usuario final. Un servidor HTTP es una pieza de software que comprende URLs (direcciones web) y HTTP (el protocolo que tu navegador usa para ver las páginas web). [1]

4 Descripción

Para este laboratorio vamos a hacer uso de docker, y aws educate. Primero realizaremos la implementación de nuestro Round Robin y el log service, haremos uso de una base de datos Mongo, para esto nos ayudaremos con el servicio de clusters de Compass MongoDB. Luego de crear nuestro programa, en la carpeta raíz del proyecto crearemos un archivo llamado "Dockerfile" sin extensión, contendrá lo siguiente:

FROM openjdk:8
WORKDIR /usrapp/bin
ENV PORT 6000
COPY /target/classes /usrapp/bin/classes
COPY /target/dependency /usrapp/bin/dependency
CMD ["java","-cp","./classes:./dependency/*","edu.escuelaing.arep .RoundRobin.SparkWebServer"]

Luego de esto crearemos nuestra imagen docker con el siguiente comando:
docker build –tag dockerspark .

Por consiguiente procederemos a crear tres instancias del docker que serán para el log service:
docker run -d -p 34001:6000 –name primerdockercontainer dockersparkr

docker run -d -p 34002:6000 –name segundodockercontainer dockerspark

docker run -d -p 34003:6000 –name tercerdockercontainer dockerspark

Con esto nuestro servicio ya esta corriendo en los tres dockers y con esos puertos, si desea probar debe ingresar a http://localhost:34001/ o cualquiera de los tres y verá su respuesta. A continuación vamos a finalizar la preparación de nuestros dockers, creamos un archivo en el directorio raíz llamado "docker-compose.yml" que tendrá el siguiente contenido para poder configurar la base de datos mongo y el servicio web:
version: '2'
services:

web:
build:
context: .
dockerfile: Dockerfile
container$_name$: web
$ports$:
− "8087 : 6000"
db :
$image$: $mongo$: 3.6.1
$container_name$: db
$volumes$:
− $mongodb$: $/data/db$
− $mongodb_config$: $/data/configdb$
$ports$:
− 27017 : 27017
$command$: $mongod$
$volumes$:
$mongodb$:
$mongodb_config$:

Con esto hemos terminado la preparacion de los docker. Ahora debemos ingresar a docker hub y crear un repositorio para nuestros dockers, después de haberlo creado ejecutamos este comando en la consola "docker tag dockersparkprimer usuario/nombreRepositorio" y habremos creado una imagen con el nombre dle repositorio, por último ejecutamos "docker login" para acceder al repositorio y un push "docker push usuario/nombreRepositorio:latest" para actualizar nuestro repositorio.

Por último queda acceder a AWS Educate y crear una máquina virtual donde subiremos nuestro servicio. Al tener la máquina virtual creada debemos ingresar a ella y ejecutar los siguientes comandos:
sudo yum update -y
sudo yum install docker
sudo service docker start
sudo usermod -a -G docker ec2-user
docker run -d -p 8087:6000 –name firstdockerimageaws usuario/nombreRepositorio

Con esto finalizamos y nuestro servicio queda montado en la plataforma de aws.

5 Conclusiones

Hacer uso de dockers y AWS, nos permite tener un ambiente de funcionamiento más eficiente de nuestros servicios web, haciendo uso

de implementaciones de log service podemos distribuir la ejecución de nuestro programa. Estos ambientes web favorecen tener un mejor funcionamiento de nuestras aplicaciones.

References

[1] **MDN contributors.** ***Que es un servidor WEB?*** **Developer Mozilla, 2020.**

[2] **Red Hat.** ***¿Qué son la integración/distribución continuas (CI/CD)?*** **Red Hat.**

[3] **Amazon Web Service.** ***¿Qué es AWS Educate?*** **Amazon Web Services.**

[4] **Windoctor7.** ***CircleCI Integración continua.*** **Windoctor 7, 2015.**