PRÁCTICA CREATIVA 2

DESPLIEGUE DE UNA APLICACIÓN ESCALABLE

26/1//2023

Iván Caamaño Castañeda
Mario Fernández Sobrino

Despliegue de la aplicación en máquina virtual pesada

Principalmente, para el primer apartado, decidimos hacer la aplicación usando las máquinas de *Google Cloud*; definiendo una *MV* que permitiera el acceso *http* y *https*, y creando una regla de firewall "*tcp"* para permitir el tráfico pedido en la práctica. Para desplegar la aplicación, usamos un script de python ***"bloq1"*** encontrado en el zip. Para cambiar los valores de ciertos ficheros dentro de **practica_creativa2**, creamos una función **replace_text_in_file,** clonamos el repositorio pedido, declaramos la variable de entorno, y hacemos ciertos cambios en **requirements.txt** debido a incompatibilidad de versiones; posteriormente, se ejecuta la aplicación en el puerto *9080*; para poder cambiar el puerto en el cual se ejecutará, se puede crear un input que pida por pantalla el puerto deseado, y luego cambie el puerto en **requirements.txt**

Despliegue de una aplicación monolítica usando docker

Para esta segunda parte hemos decidido utilizar una máquina virtual facilitada por el profesorado que tiene docker, docker-compose y minikube. Hemos creado un script de python llamado ***"bloq2"*** el cual crea el archivo **Dockerfile**. En este archivo definimos la variable de entorno <GROUP_NUMBER>, descargamos la práctica creativa 2, instalamos las dependencias necesarias y mediante *'pip'* las especificadas en **requirements.txt.** Mediante el comando *'sed'* modificamos el título de **productpage.html.** También habilitamos el puerto 9080. Por último, en el CMD ejecutamos el script ***productpage_monolith.py*** en el puerto 9080 previamente habilitado.
Posteriormente, en el script de python mediante las llamadas a *"docker build -t '43/product-page' ."* para crear la imagen de docker, y a *'docker run --name 43-product-page -p 9080:9080 -e GROUP_NUMBER=43 -d 43/product-page'* para arrancar el contenedor.
Entre los problemas encontrados ha estado que no se pasa correctamente la variable de entorno GROUP_NUMBER. Se han utilizado varias formas de pasar el parámetro en el Dockerfile como : ${GROUP_NUMBER}, $GROUP_NUMBER…

En comparación con el uso de máquinas pesadas cabe destacar la ventaja de la virtualización ligera en cuanto a la cantidad de recursos utilizados/demandados. Así como también encontramos el inconveniente de que la virtualización ligera proporciona una máquina virtual que se basa con un interfaz de sistema operativo, por tanto, no se puede ejecutar un sistema operativo diferente del anfitrión.

Segmentación de una aplicación monolítica en microservicios utilizando dockercompose

En este caso, hemos hecho los ficheros **ProductPage**, **Details**, **Ratings** y **Reviews**, además del fichero orquestador docker, que será el **docker-compose** con toda la información de los microservicios ofrecidos.

Entre las ventajas de utilizar docker-compose con respecto a docker, están:

- Ejecutando docker-compose up se iniciarán todos los contenedores definidos en tu archivo docker-compose en lugar de iniciar manualmente cada contenedor con el comando docker run.
- Permite definir y conectar fácilmente múltiples contenedores para su aplicación, sin necesidad de exponer puertos o enlazar contenedores.
- Permite gestionar fácilmente los recursos para su aplicación, como la red y los volúmenes, en un único archivo.

Y precisamente entre sus desventajas encontramos que solo puede ser utilizado en aplicaciones Docker.