Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Desarrollo De Aplicaciones Para Sistemas Ubicuos

Docker con MQTT

Estudiante
Juan Sebastián Realpe González

Código
100616021378

Profesor:
Ing. Julian Andrés Bolaños

Popayán, Cauca

2021

Índice

1. Práctica 5 Docker Inicio

Esta práctica tiene como objetivo introducir al estudiante en el manejo de herramientas para el despliegue y empaquetamiento de aplicaciones. La herramienta a trabajar será Docker, lacual se basa en la factorización de software y la optimización del hardware disponible, haciendo una analogía a las máquinas virtuales las cuales dependen de la implementación completa de una instancia de si misma. Por el lado de Docker, se trata de un concepto novedoso al componerse de imágenes que comparten muchas veces el mismo kernel de Linux y en muchos casos hasta el sistema operativo.

1.1. Instalación

Para proceder a la instalación de Docker es necesario contar con el sistema operativo LINUX, en cualquiera de sus distribuciones, ya que el mismo está construido para este tipo de distribuciones. Para sistemas operativos Windows es posible bien sea montar una máquina virtual utilizando VirtualBOX, o manejar el software "Docker Desktop" que permite manejar Docker en WSL2 con una máquina virtual simulada.

- Se descarga el instalador en `https://docs.docker.com/engine/install/`
- Se realiza la instalación de WSL2 para el correcto funcionamiento de Docker Desktop siguiendo los pasos de `https://docs.microsoft.com/en-us/windows/wsl/install`
- En una ventana de comandos o en powershell se escribe el comando `wsl -install`
- Se obtiene la siguiente ventana

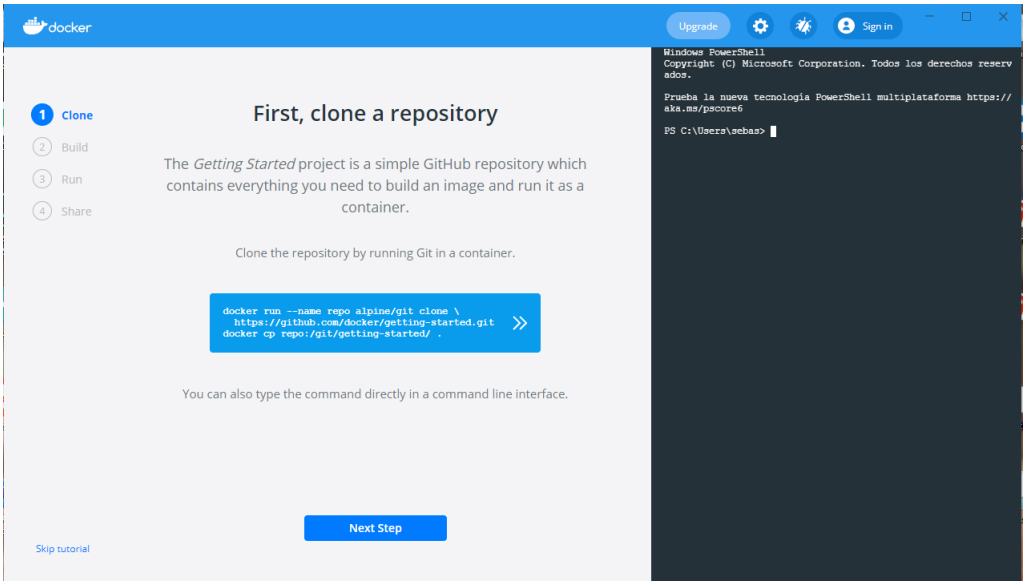

Figura 1: Ventana de Docker

- Para comprobar el funcionamiento de Docker, se corre el siguiente comando `docker run hello-world`

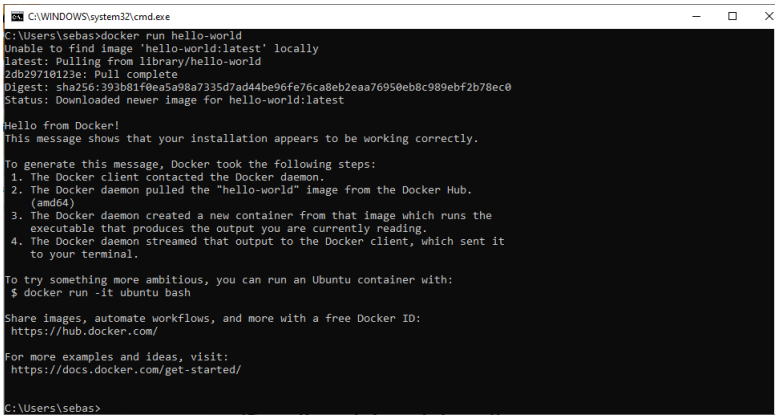

Figura 2: Hola mundo en Docker

2. Práctica 6 Docker Fundamentals

Esta práctica tiene como objetivo introducir al estudiante en el manejo de herramientas para el despliegue y empaquetamiento de aplicaciones. Se revisarán algunos de los aspectos fundamentales para introducir al estudiante en el entendimiento de la herramienta Docker para el manejo de aplicaciones.

2.1. Manejo de imágenes

Docker es una comunidad abierta, y en el Docker hub existen múltiples imágenes que se pueden descargar, usar y modificar.
Para descargar esta y cualquier imagen se debe tener en cuenta el siguiente comando:
`docker pull postgres`

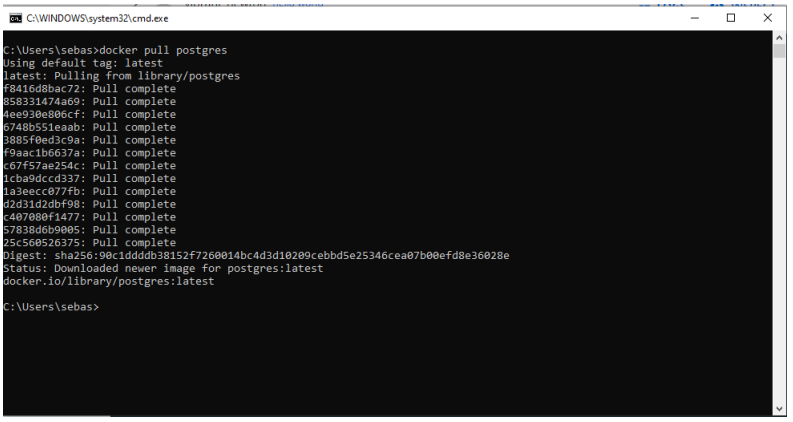

Figura 3: Creación imagen postgres

Para correr la imagen:
`docker run -e POSTGRES_PASSWORD = password postgres`

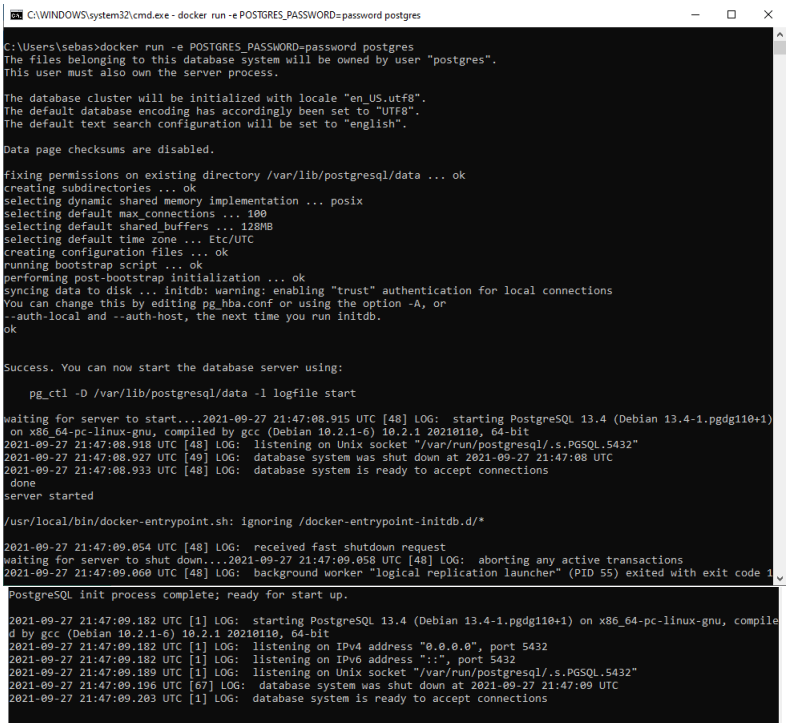

Figura 4: Correr imagen postgres

Con el comando `docker images` se puede ver las imágenes creadas.

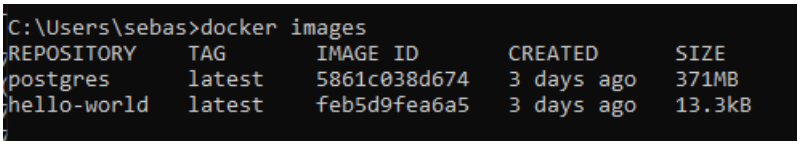

Figura 5: Imágenes creadas en Docker

Con el comando `docker ps` se puede ver los containers creados

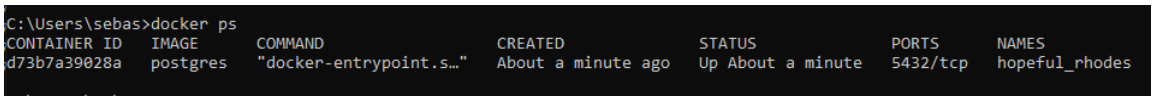

Figura 6: Containers creados en Docker

2.2. Dockerfile

El uso de dockerfile puede ser adecuado para lleva a cabo implementaciones de nuestra aplicación en tiempo real, tomando el ejemplo de la guía se obtiene en `localhost:8060` lo siguiente:

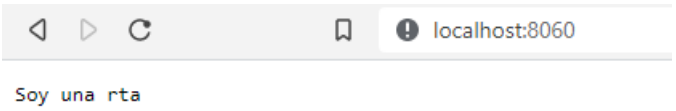

Figura 7: Salida localhost:8060, ejemplo 1

Se corre el comando `docker build -t imagenode .` para crear una imagen con el tag imagenode que podremos correr posteriormente con el comando `docker run imagenode` como se muestra en las siguientes figuras.

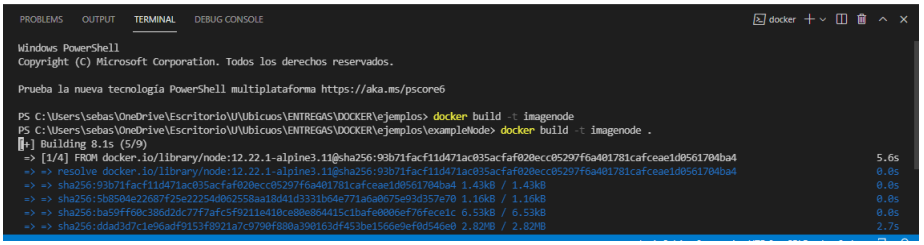

Figura 8: Crear imagen con etiqueta

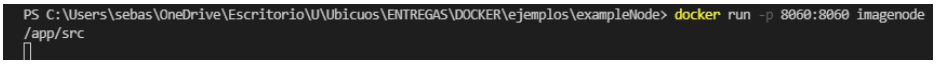

Figura 9: Correr imagen con etiqueta

En la ventana de docker se muestra lo siguiente:

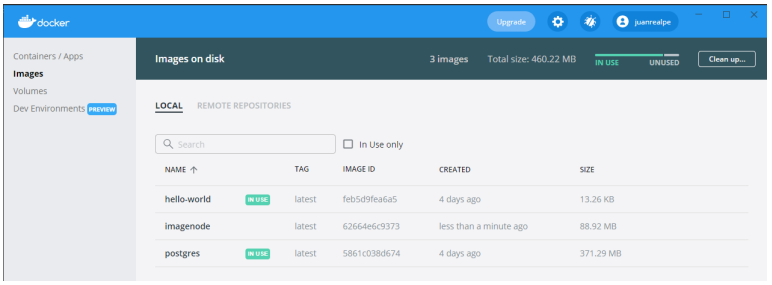

Figura 10: Ventana docker

En `localhost:8060`, se muestra el siguiente mensaje:

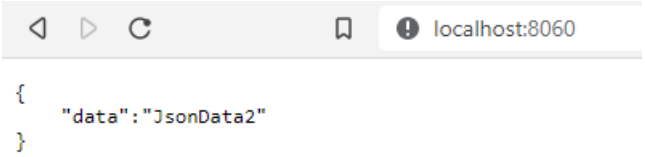

Figura 11: Salida localhost:8060, ejemplo 2

Para que sea posible observar desde el contenedor un archivo cualquiera o un directorio es necesario el uso de volúmenes, para este caso se creó una nueva imagen llamada *imagenode-tres*, y se probó de la siguiente forma:

Figura 12: Uso de volúmenes

En `localhost:8060`, se muestra el siguiente mensaje:

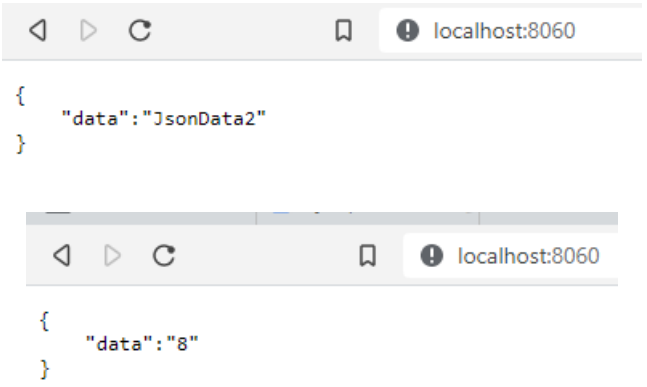

Figura 13: Salida localhost:8060, ejemplo 3

3. Tareas

3.1. MQTT pub-sub

Se usa la imagen ruimarinho-mosquitto para implementar un nuevo ejemplo, con el siguiente comando
`docker pull postgres` :

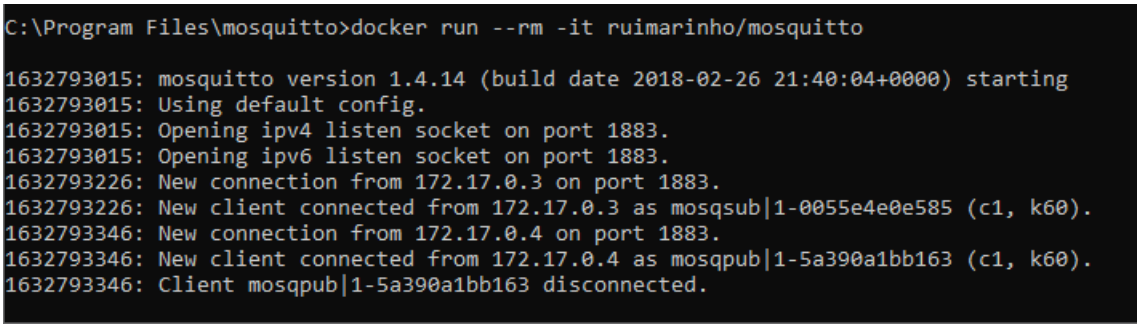

Figura 14: Imagen ruimarinho-mosquitto

Luego, se procede a crear probar el método subscribe y publish, para este caso, se usaron los siguientes comandos:

- Para subscribe

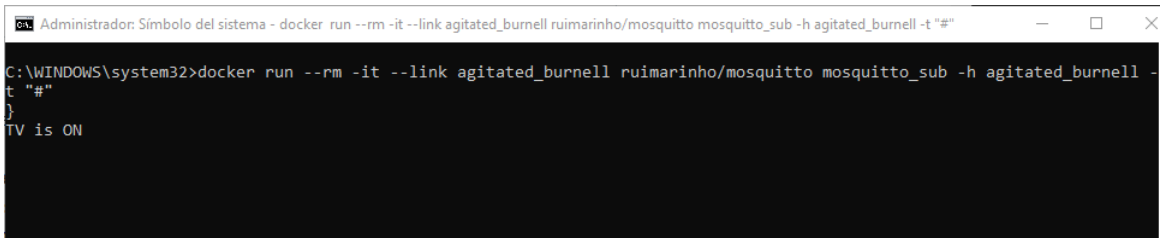

Figura 15: Comando subscribe

- Para publish

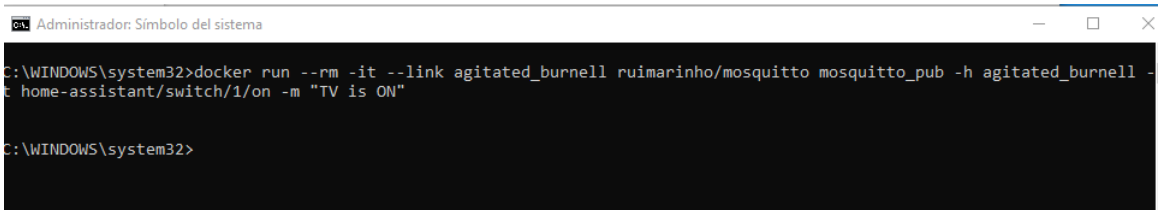

Figura 16: Comando publish

Se debe tener en cuenta que el nombre de la imagen corresponda a la imagen creada en Docker.

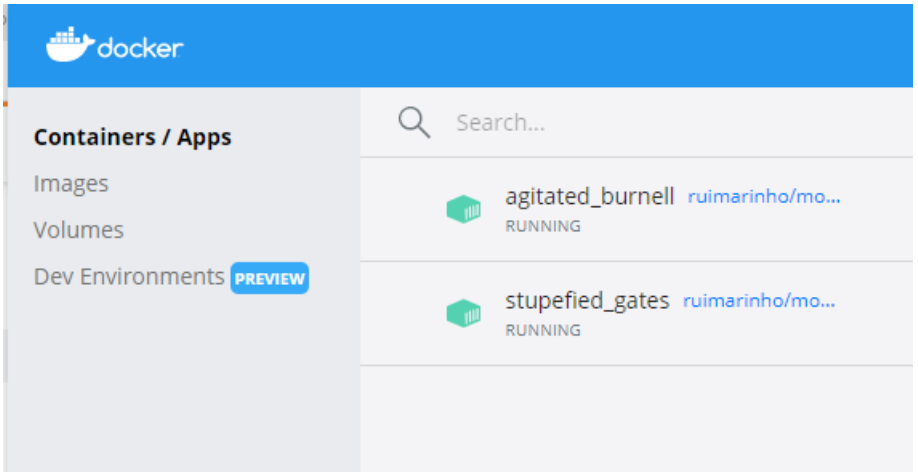

Figura 17: Ventana Docker

Y dentro de la otra imagen, debe mostrarse el mensaje publicado.

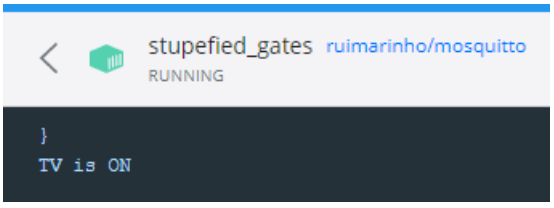

Figura 18: Imagen "stupefied_gates"

3.2. MYSQL

Para MYSQL, se hace uso del archivo docker-compose.yml, y se hace la respectivas correcciones, para posteriormente probarlo.

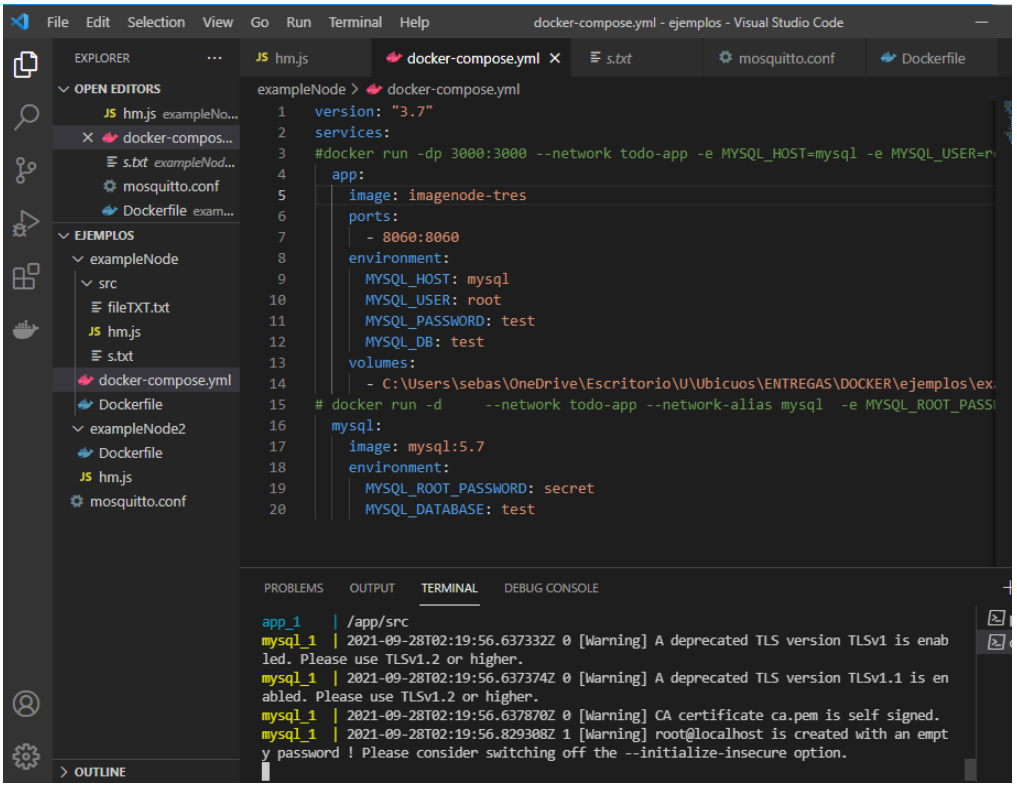

Figura 19: Archivo docker-compose.yml

Y se puede observar en `localhost:8060` el mensaje que se uso en los anteriores puntos, gracias a la imagen *imagenode-tres*.

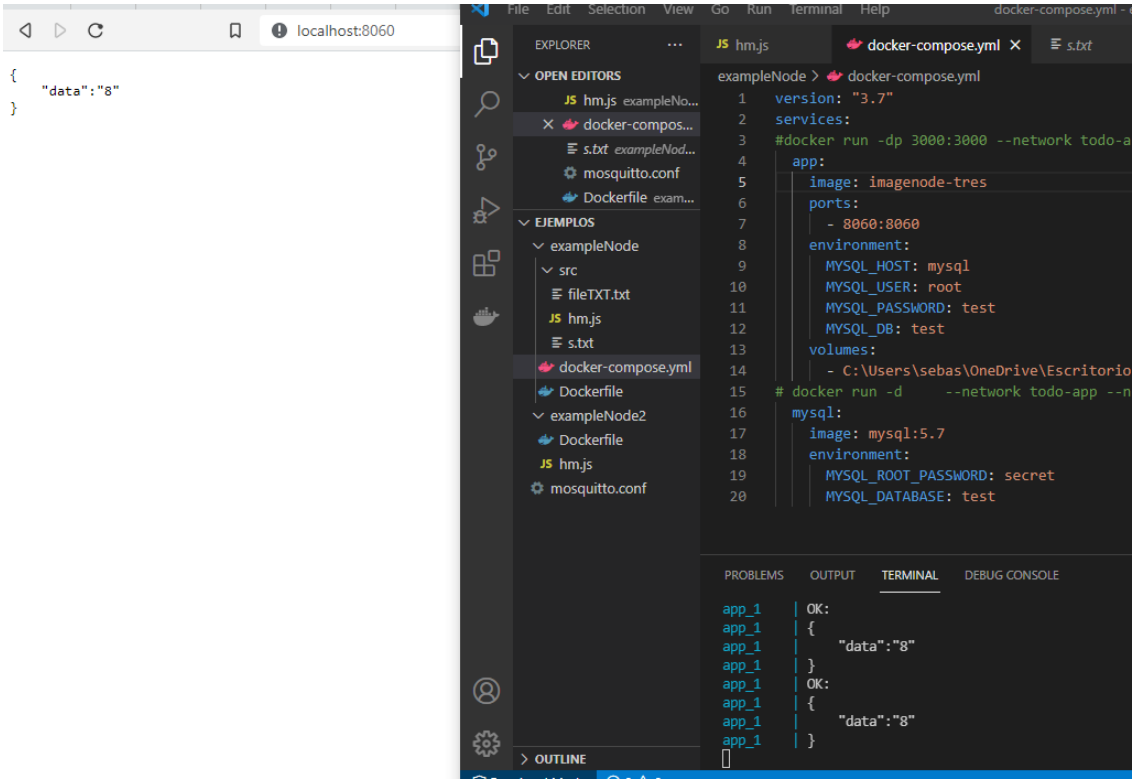

Figura 20: Salida localhost:8060 Tarea