

1. **Crea una nueva red que se llame “adaitsnetwork”**

Para crear la red en Docker, vamos a usar el comando “docker network create ‘nombre de la red’”.



1. **Crea un nuevo contenedor: Servidor web NGINX (última**

**versión) que por defecto inicie con una página html (web.html) que**

**contendrá un botón que hará peticiones a una API con el contenido**

**indicado en la imagen del anexo.**

**1. Crea una carpeta que contenga:**

**- web.html**

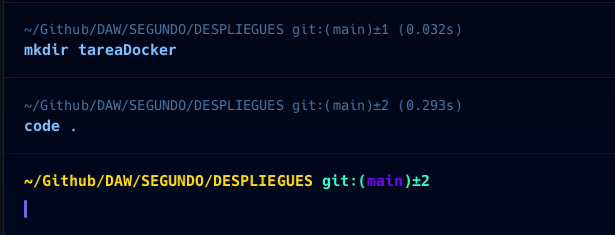
**- DockerFile**

**2. Construir la imagen nginx-daw (docker build)**

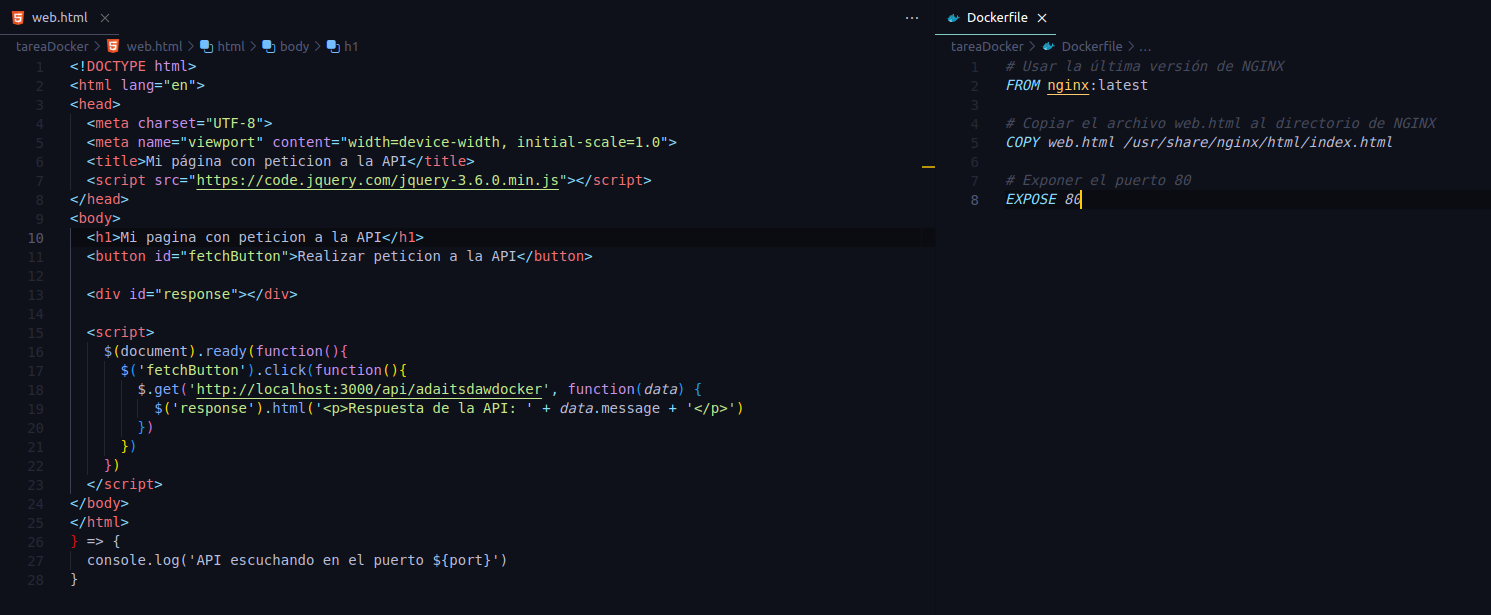
**3. Ejecutar contenedor con la imagen nginx-daw y en la red**

**adaitsnetwork**

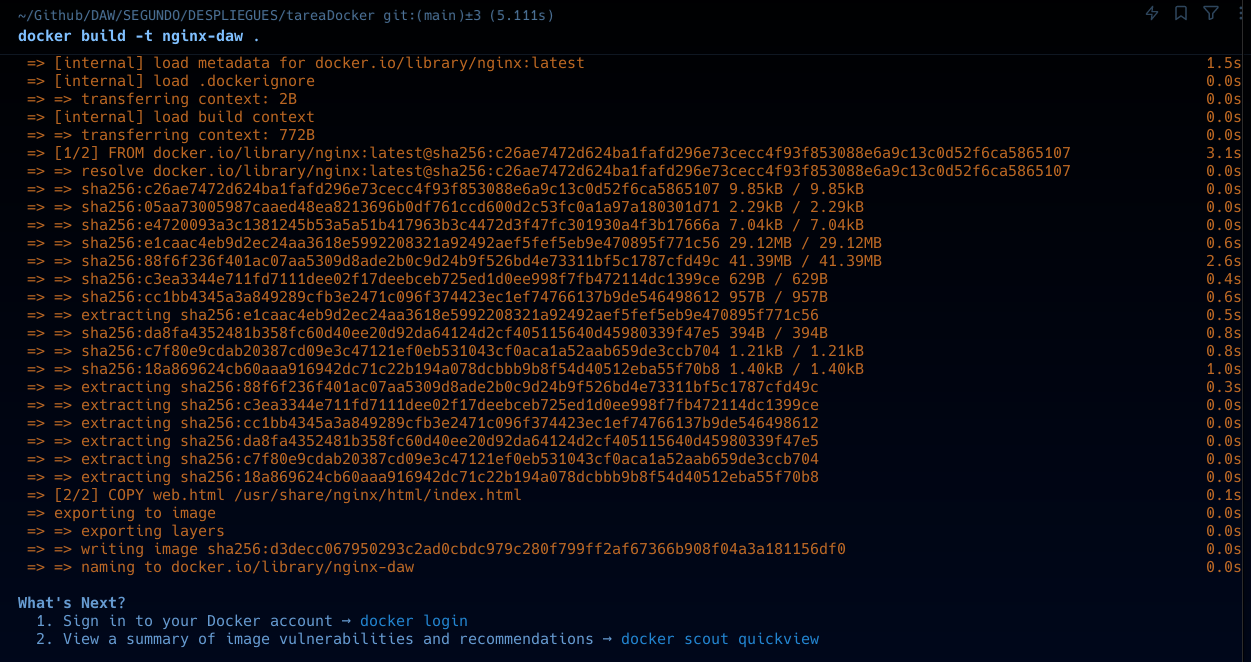
Primero de todo, vamos a crear una carpeta donde vamos a tener nuestro web.html y el archivo Dockerfile.



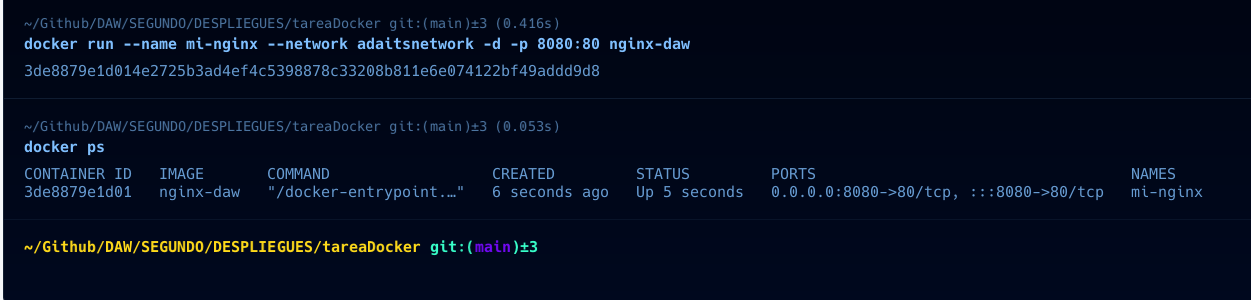
Con el comando “code .” abrimos VSCode en el directorio donde nos encontramos. Dentro del directorio “tareaDocker” vamos a crear ambos archivos.



Ahora, vamos a navegar hasta la carpeta que contiene los archivos y vamos a ejecutar el comando “docker build -t nginx-daw .”.



Por último, vamos a ejecutar el contenedor utilizando la red que hemos creado previamente.



1. **Crea una API REST en Node.js que contenga un server.js con el contenido indicado en la imagen del anexo.**

**1. Crea una carpeta que contenga:**

**- server.js**

**- DockerFile**

1. **Construir la imagen api-rest-daw (docker build)**

**3. Ejecutar el contenedor con la imagen api-rest-daw y en la red**

**adaitsnetwork**

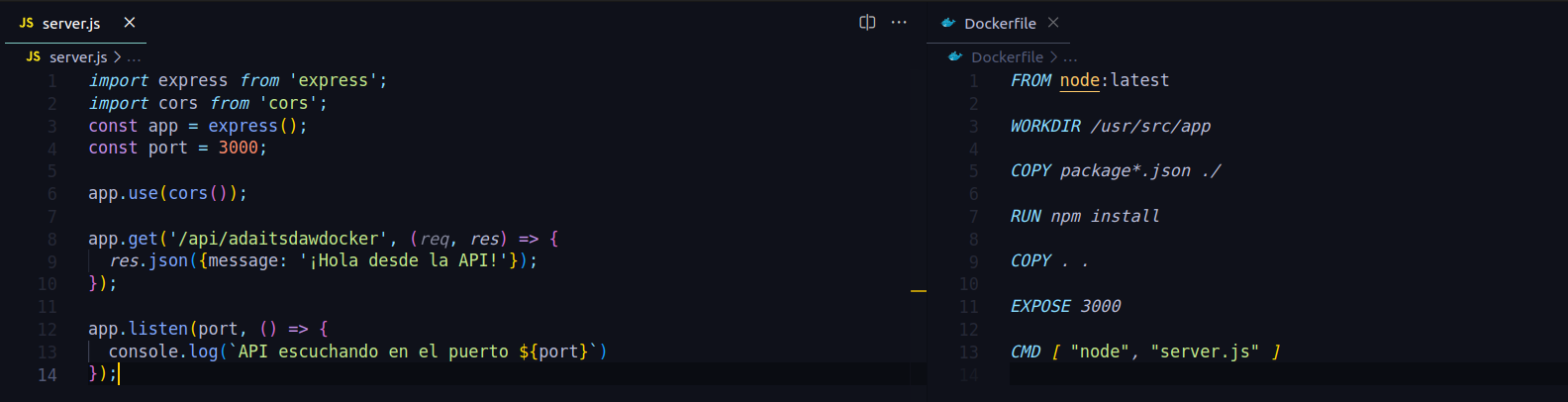
**Para facilitar la creación de los contenedores, se debe realizar un docker-**

**compose (.yml), el cual facilitará crear los servicios a la vez:**

**- NGINX**

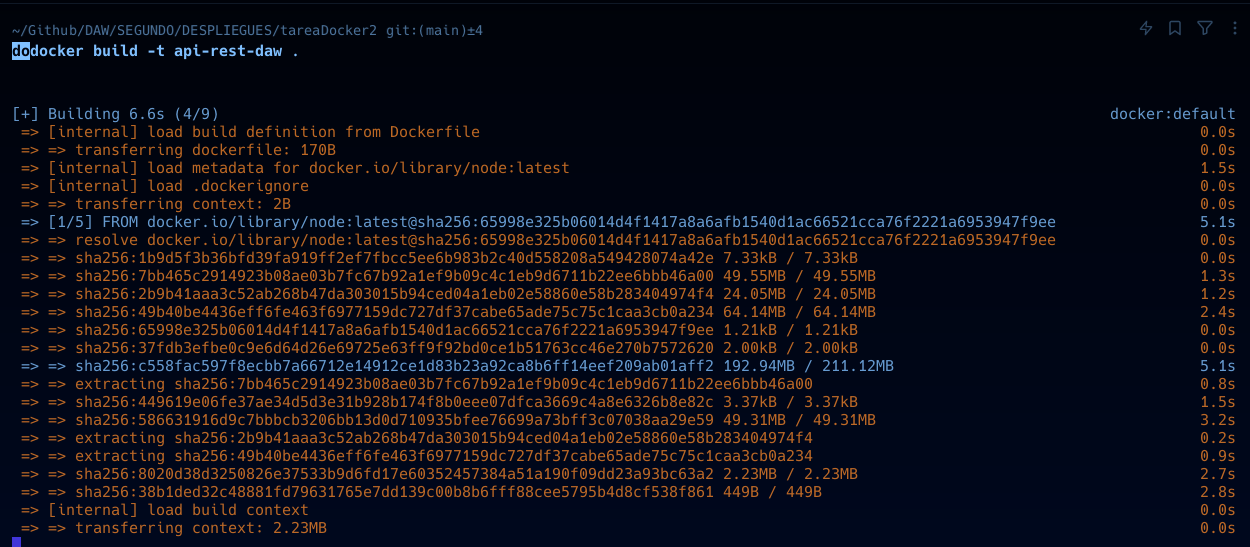
**- API**

Para crear la API REST, he creado un nuevo directorio llamado “TareaDocker2”, sobre el que vamos a trabajar. Vamos a empezar creando “server.js” y Dockerfile.



Es importante asegurarnos de preparar correctamente nuestro “package.json” para que las dependencias se instalen correctamente al hacer nuestra build.

Ahora, vamos a construir la imagen de docker de nuestra API REST.



Una vez creada la imagen, vamos a ejecutar el contenedor en nuestra red “adaitsnetwork”.



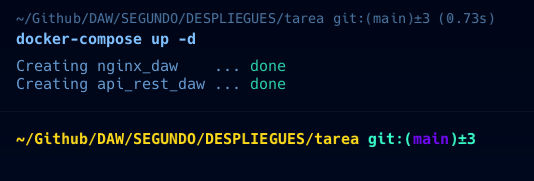
Para hacer el despliegue de ambos contenedores usando docker compose, he reestructurado los directorios donde tenia ambos contenedores para facilitarme el proceso. Han quedado tal que así:



Ahora, vamos a definir el contenido del archivo “docker-compose.yaml”.



Una vez tenemos configurado nuestro docker compose, ejecutamos el comando “docker-compose up -d” para proceder a crear y ejecutar nuestros contenedores.



Nuestros contenedores están corriendo correctamente. Vamos a acceder a la web para probar que todo funciona como debería.

