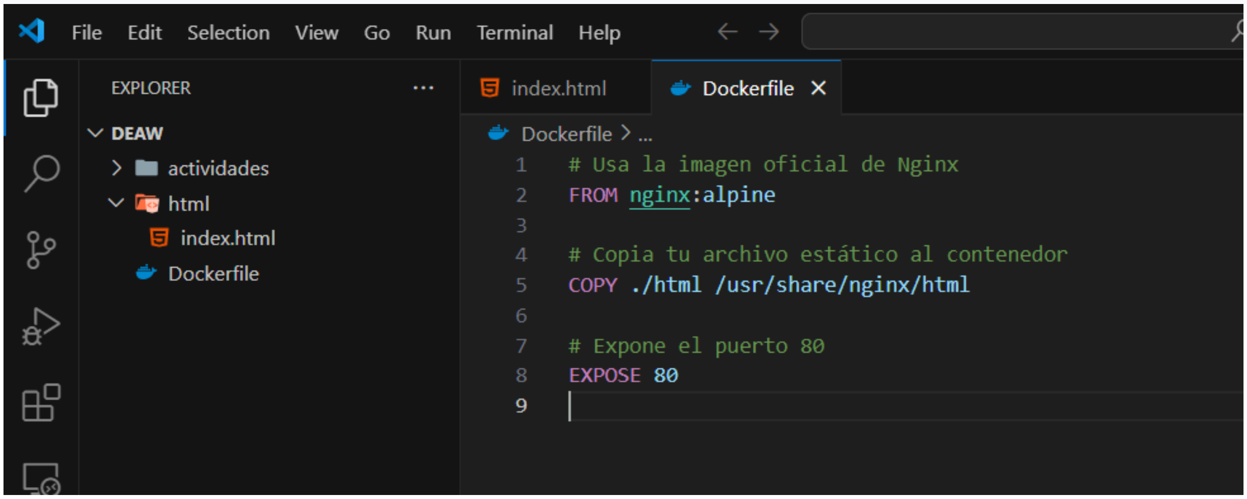
Crear una imagen con un nginx que sirva un documento estático:



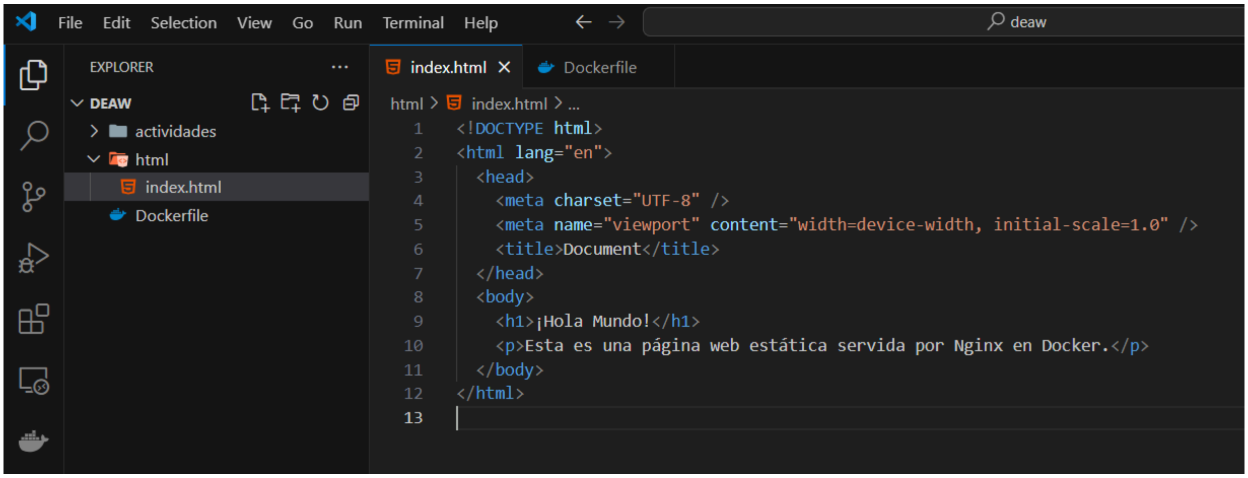
```
# Usa la imagen oficial de Nginx
FROM nginx:alpine

# Copia tu archivo estático al contenedor
COPY ./html /usr/share/nginx/html

# Expone el puerto 80
EXPOSE 80
```

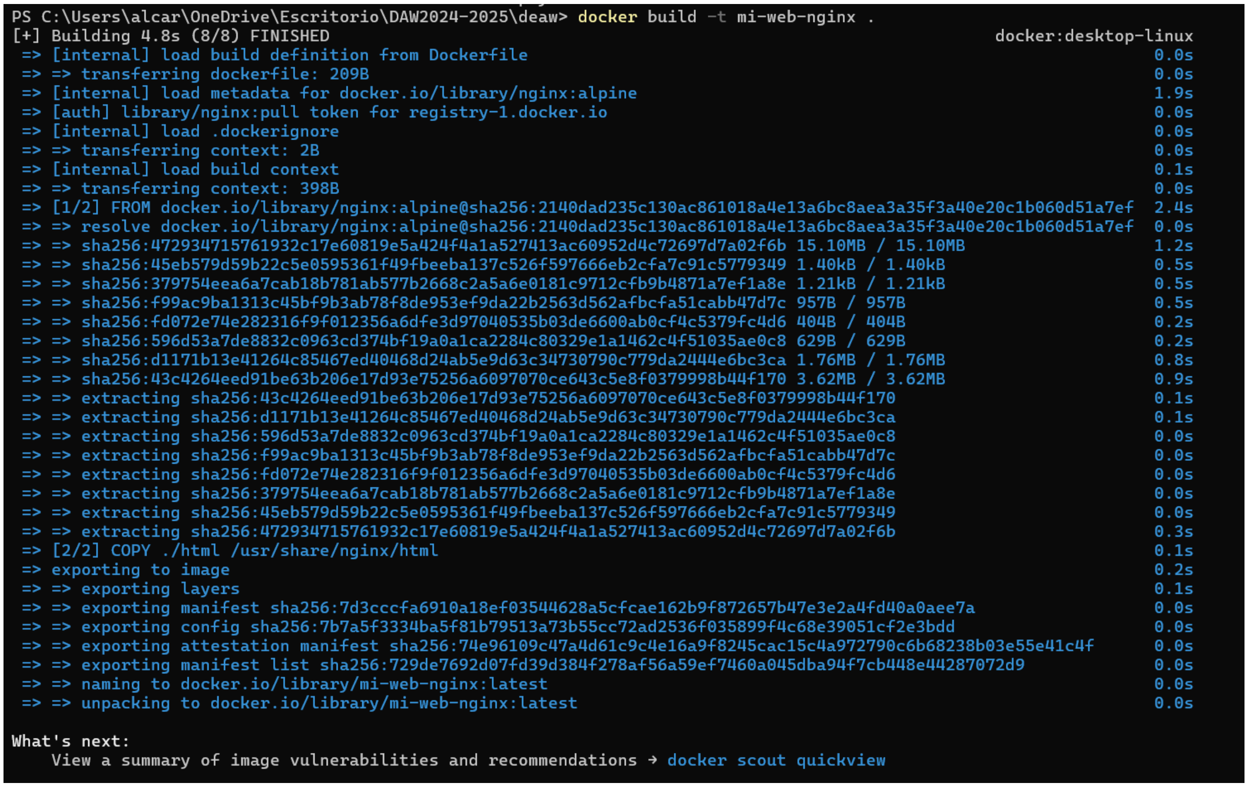
- ○ Utiliza la última versión de la imagen oficial de nginx en Docker Hub.
- ○ Crea una página web que contenga esto:

<h1>¡Hola Mundo!</h1>
<p>Esta es una página web estática servida por Nginx en Docker.</p>



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>¡Hola Mundo!</h1>
    <p>Esta es una página web estática servida por Nginx en Docker.</p>
  </body>
</html>
```

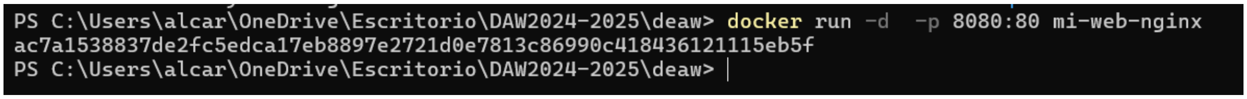
- ○ Utiliza docker build y dale este nombre a la imagen: mi-web-nginx



```
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker build -t mi-web-nginx .
[+] Building 4.8s (8/8) FINISHED                                                                       docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                   0.0s
 => => transferring dockerfile: 209B                                                                                   0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                                        1.9s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                           0.0s
 => [internal] load .dockerignore                                                                                      0.0s
 => => transferring context: 2B                                                                                        0.0s
 => [internal] load build context                                                                                      0.1s
 => => transferring context: 398B                                                                                      0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:2140dad235c130ac861018a4e13a6bc8aea3a35f3a40e20c1b060d51a7ef  2.4s
 => => resolve docker.io/library/nginx:alpine@sha256:2140dad235c130ac861018a4e13a6bc8aea3a35f3a40e20c1b060d51a7ef  0.0s
 => => sha256:472934715761932c17e60819e5a424f4a1a527413ac60952d4c72697d7a02f6b 15.10MB / 15.10MB                       1.2s
 => => sha256:45eb579d59b22c5e0595361f49fbeeba137c526f597666eb2cfa7c91c5779349 1.40kB / 1.40kB                         0.5s
 => => sha256:379754eea6a7cab18b781ab577b2668c2a5a6e0181c9712cfb9b4871a7ef1a8e 1.21kB / 1.21kB                         0.5s
 => => sha256:f99ac9ba1313c45bf9b3ab78f8de953ef9da22b2563d562afbcfa51cabb47d7c 957B / 957B                             0.5s
 => => sha256:fd072e74e282316f9f012356a6dfe3d97040535b03de6600ab0cf4c5379fc4d6 404B / 404B                             0.2s
 => => sha256:596d53a7de8832c0963cd374bf19a0a1ca2284c80329e1a1462c4f51035ae0c8 629B / 629B                             0.2s
 => => sha256:d1171b13e41264c85467ed40468d24ab5e9d63c34730790c779da2444e6bc3ca 1.76MB / 1.76MB                         0.8s
 => => sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170 3.62MB / 3.62MB                         0.9s
 => => extracting sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170                              0.1s
 => => extracting sha256:d1171b13e41264c85467ed40468d24ab5e9d63c34730790c779da2444e6bc3ca                              0.1s
 => => extracting sha256:596d53a7de8832c0963cd374bf19a0a1ca2284c80329e1a1462c4f51035ae0c8                              0.0s
 => => extracting sha256:f99ac9ba1313c45bf9b3ab78f8de953ef9da22b2563d562afbcfa51cabb47d7c                              0.0s
 => => extracting sha256:fd072e74e282316f9f012356a6dfe3d97040535b03de6600ab0cf4c5379fc4d6                              0.0s
 => => extracting sha256:379754eea6a7cab18b781ab577b2668c2a5a6e0181c9712cfb9b4871a7ef1a8e                              0.0s
 => => extracting sha256:45eb579d59b22c5e0595361f49fbeeba137c526f597666eb2cfa7c91c5779349                              0.0s
 => => extracting sha256:472934715761932c17e60819e5a424f4a1a527413ac60952d4c72697d7a02f6b                              0.3s
 => [2/2] COPY ./html /usr/share/nginx/html                                                                            0.1s
 => exporting to image                                                                                                 0.2s
 => => exporting layers                                                                                                0.1s
 => => exporting manifest sha256:7d3cccfa6910a18ef03544628a5cfcae162b9f872657b47e3e2a4fd40a0aee7a                      0.0s
 => => exporting config sha256:7b7a5f3334ba5f81b79513a73b55cc72ad2536f035899f4c68e39051cf2e3bdd                        0.0s
 => => exporting attestation manifest sha256:74e96109c47a4d61c9c4e16a9f8245cac15c4a972790c6b68238b03e55e41c4f          0.0s
 => => exporting manifest list sha256:729de7692d07fd39d384f278af56a59ef7460a045dba94f7cb448e44287072d9                 0.0s
 => => naming to docker.io/library/mi-web-nginx:latest                                                                 0.0s
 => => unpacking to docker.io/library/mi-web-nginx:latest                                                              0.0s

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
```

- ○ Ejecuta un contenedor con Docker run mapeando el puerto 80 del contenedor al 8080 del host.
- ○ Verifica que el servidor nginx esté funcionando accediendo desde un navegador a http://localhost:8080



```
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker run -d  -p 8080:80 mi-web-nginx
ac7a1538837de2fc5edca17eb8897e2721d0e7813c86990c418436121115eb5f
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> 
```