

Laboratorio 5. Volumenes. Operaciones con la app NODE

Paso 1. Creamos una carpeta en el directorio le ponemos un nombre proyecto 2Node-Volumenes. Descargamos la carpeta adjunta. Abrimos la carpeta en Visual estudio Code.

```
PS C:\Users\carlo\OneDrive\Escritorio\2Node-Volumenes>
```

Paso 2 . Creamos un Dockerfile

```
1 FROM node:14
2
3 WORKDIR /app
4
5 COPY package.json .
6
7 RUN npm install
8
9 COPY . .
10
11 EXPOSE 80
12
13 CMD ["node", "server.js"]
```

Paso 3. Construimos la imagen.

A continuación, podemos ver el resultado (después de unos minutos).

```
PS C:\Users\carlo\OneDrive\Escritorio\2Node-Volumenes> docker build -t feedback-node .
[+] Building 78.4s (18/18) FINISHED
--> [Internal] load .dockerignore
--> [Internal] load build definition from Dockerfile
--> [Internal] load metadata for docker.io/library/node:14
--> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4c3fa813fabc8c590b0fba060e015ad4e59bce5744d2f6f8461aa
--> resolve docker.io/library/node:14@sha256:a158d3b9b4c3fa813fabc8c590b0fba060e015ad4e59bce5744d2f6f8461aa
--> sha256:2ff1d7c41c74a2525bfa0f08da0ba727f84518f55f65ca845bc747978c408 50.49MB / 50.49MB
--> sha256:2c4af31b86029e4720af590e27ee557a2da07010132201779959927f 2.21MB / 2.21MB
--> sha256:1d12470fa602a5c50378dc0ea228c1735747db4180b6efb0e20914405452 7.51MB / 7.51MB
--> sha256:b253aeafaa7e0671b600080f81de101a3ba045ff7bc656a30dfbf7c705cca5 7.80MB / 7.80MB
--> sha256:3d2201bd995ccc12851a50820a0e3d34a17011dc69ac9f0f3a9c952cb131 10.00MB / 10.00MB
--> sha256:a158d3b9b4c3fa813fabc8c590b0fba060e015ad4e59bce5744d2f6f8461aa 77MB / 77MB
--> sha256:1d676e208b10d9fa090e0f77951ff540912b63429c34f5d0adfa09f5f9ee2 51.80MB / 51.80MB
--> sha256:d9a0df5894511ce28a5e2925a7508a0a034c39ad734fd0fb0e23d1b1569 191.85MB / 191.85MB
--> sha256:0f11e0945a04099890e01b0e00e0f20ebf1a1140b3f613a0c0e0408 4.10MB / 4.10MB
--> sha256:5f32ed3c3f278dda4fc571c8806277355a2a08f52b52cdf865f868378a590 35.24MB / 35.24MB
--> sha256:0c8cc2f24a4dc64e02e086fc9440ba541eac0ba72d2e00df3ba22f158b3 2.29MB / 2.29MB
--> sha256:0d27a0e0132900754c0766fba06d08c20d2de96a073de352081f22c4ceb 450B / 450B
--> extracting sha256:3f41c7c41c74a2525bfa0f08da0ba727f84518f55f65ca845bc747978c408 1.75
--> extracting sha256:b253aeafaa7e0671b600080f81de101a3ba045ff7bc656a30dfbf7c705cca5 0.25
--> extracting sha256:3d2201bd995ccc12851a50820a0e3d34a17011dc69ac9f0f3a9c952cb131 1.85
--> extracting sha256:1d676e208b10d9fa090e0f77951ff540912b63429c34f5d0adfa09f5f9ee2 5.25
--> extracting sha256:d9a0df5894511ce28a5e2925a7508a0a034c39ad734fd0fb0e23d1b1569 1.55
--> extracting sha256:0f11e0945a04099890e01b0e00e0f20ebf1a1140b3f613a0c0e0408 0.05
--> extracting sha256:5f32ed3c3f278dda4fc571c8806277355a2a08f52b52cdf865f868378a590 0.05
--> extracting sha256:0c8cc2f24a4dc64e02e086fc9440ba541eac0ba72d2e00df3ba22f158b3 0.05
--> [Internal] load build context
--> transferring context: 1.00kB
--> [2/5] WORKDIR /app
--> [3/5] COPY package.json .
--> [4/5] RUN npm install
--> [5/5] COPY . .
--> exporting layers
--> writing image sha256:3f4a3caf743c31b0e0ef09ca0a844046c683d874b01523447a04d96207a1
--> naming to docker.io/library/feedback-node
What's Next?
View a summary of image vulnerabilities and recommendations + docker scout quickview
PS C:\Users\carlo\OneDrive\Escritorio\2Node-Volumenes>
```

Paso 4. Iniciar un proceso, es decir un contenedor basado en esa imagen.

Configuramos unas cuantas cosas para ello:

En el puerto local por ejemplo 3100, **-d** segundo plano, **--name** le damos un nombre de **feedback-app** y después le adjuntamos una condición de que se elimine el contenedor (**feedback-node**) una vez se pare, es decir con **--rm**

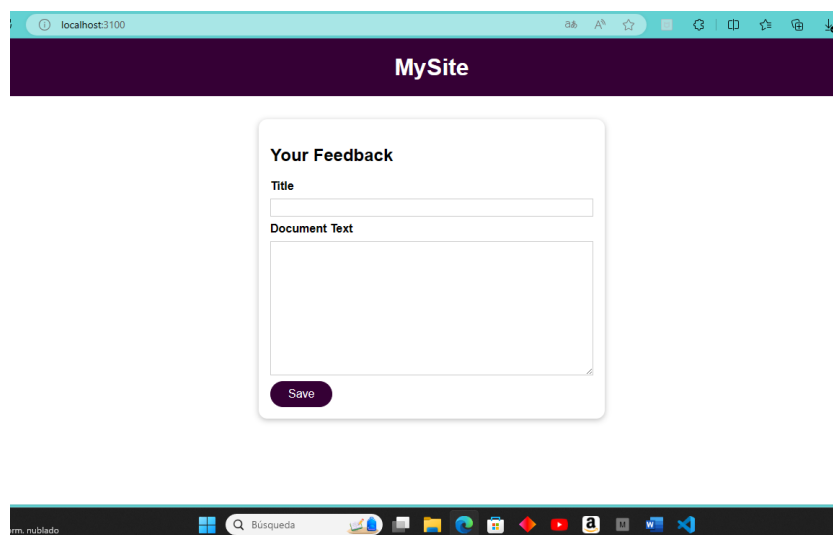
```
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
feedback-node  latest    671d2656bcee   12 seconds ago 916MB

PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> docker run -p 3100:80 -d --name feedback-app --rm feedback-node
bdc2a8787f87a3d045b92a4e4eee10998c06ccdd5873dff592e67e8cb67175e
```

Paso 5. Comprobamos el resultado con `docker ps` y verificamos en localhost que lo hemos indicado, (en mi caso es 3100).

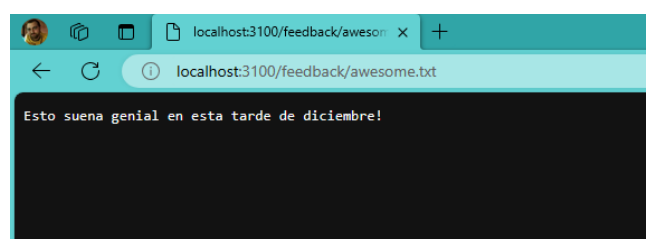
```
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
bdc2a8787f8   feedback-node  "docker-entrypoint.s..." 5 seconds ago  Up 4 seconds  0.0.0.0:3100->80/tcp    feedback-app
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup>
```

Paso 6. Comprobar el localhost:



Se puede ver este archivo almacenado usando en localhost la siguiente: Esto suena genial en esta tarde de diciembre !

/ feedback / awesome.txt para ver este archivo almacenado.



Paso 7. Detengamos este contenedor:

docker stop feedback-app

sí nos devuelve el nombre significa que ha parado.

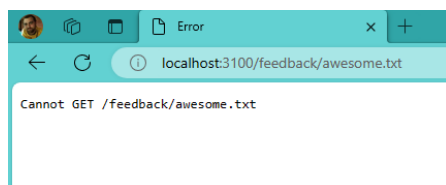
```
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> docker stop feedback-app
feedback-app
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> |
```

Paso 8. Una vez que esté detenido, reiniciémoslo, pero sin el ***-rm***, o sea ahora al pararlo no se va eliminar

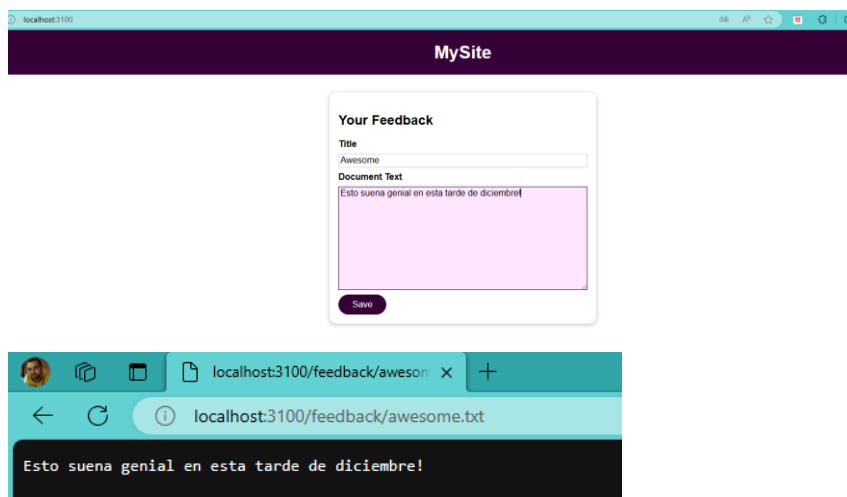
Es decir, con el siguiente comando: *docker run -p 3100:80 -d --name feedback-app feedback-node*

```
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> docker run -p 3100:80 -d --name feedback-app feedback-node
b2ae28f6980b3a562b43efeddb650f89b05c515c264bce13c5ac2263ef12e6c8
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> |
```

Paso 9. Iniciamos un nuevo contenedor con el mismo nombre, comprobamos en local host, vemos que funciona, pero el archivo *awesome.txt* me va dar un error, no existe este archivo, hace unos minutos funcionaba, pero ahora a dejado de ir.



Paso 10. Volvemos a rescribir en localhost



Paso 11. Sí ahora detenemos el contenedor .

```
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> docker stop feedback-app
feedback-app
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\Laboratorio 5 (Volumens)\2Node-Volumens\data-volumes-01-starting-setup> |
```

Paso 12. Volvemos a iniciar el contenedor otra vez:

```
PS C:\Users\carlo\OneDrive\Escritorio\DM\ENTORNIOS\PRACTICAS_DOCKER\Laboratorio 5 (Voluemenes)\2Node-Volumenes\data-volumes-01-starting-setup> docker start feedback-app  
PS C:\Users\carlo\OneDrive\Escritorio\DM\ENTORNIOS\PRACTICAS_DOCKER\Laboratorio 5 (Voluemenes)\2Node-Volumenes\data-volumes-01-starting-setup> |
```

Visitamos el local host veremos que el archivo sigue allí,

