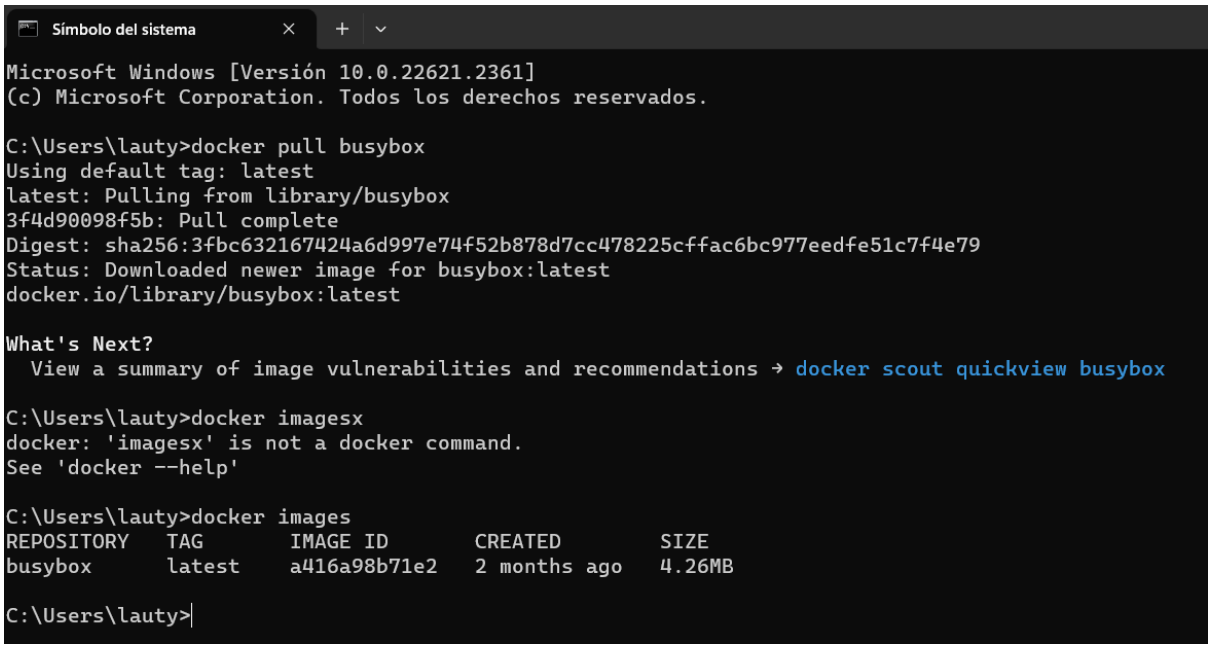
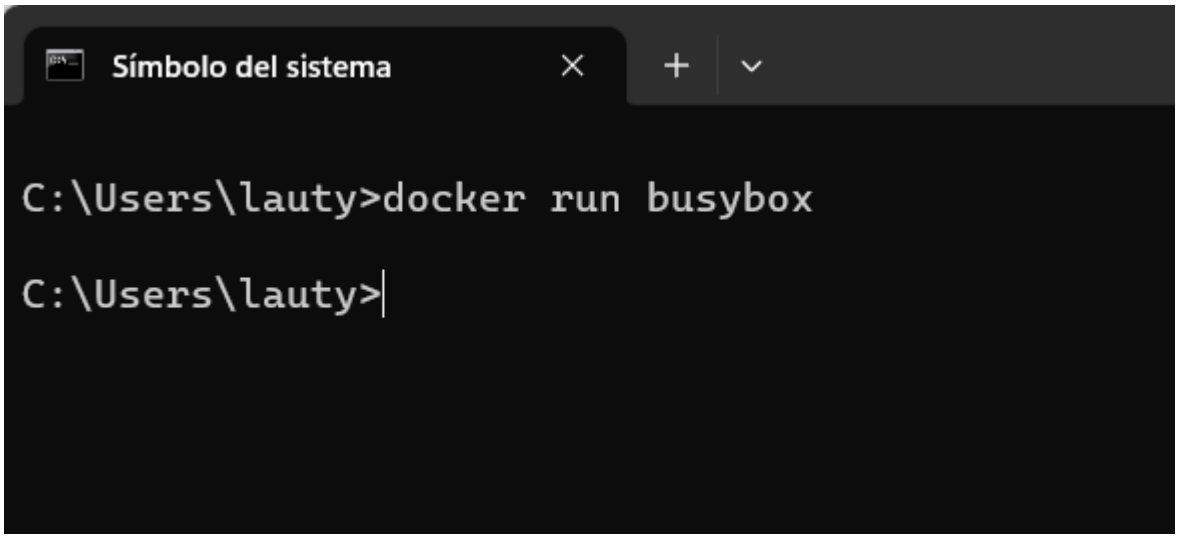
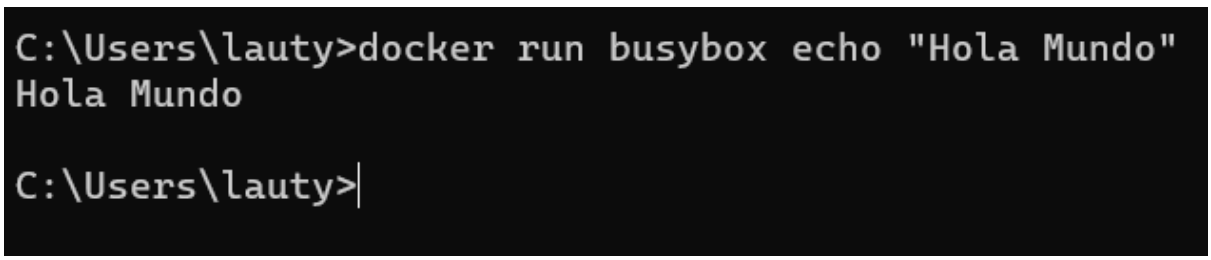
3)

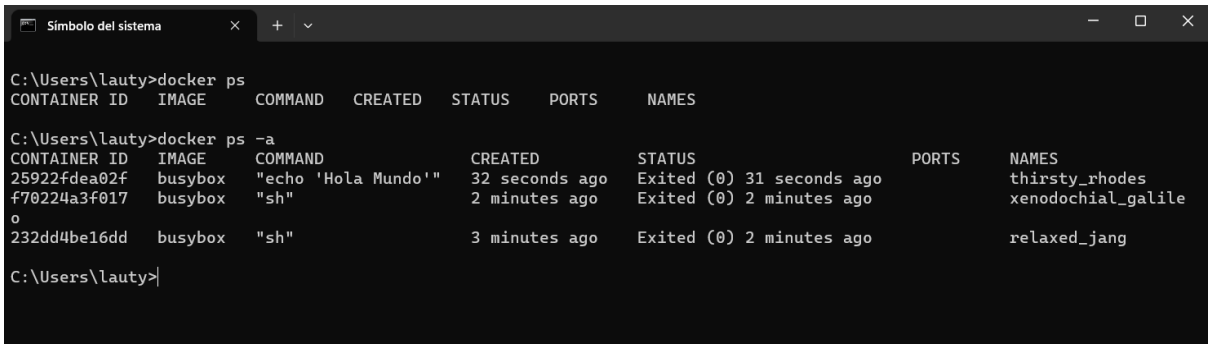


4)



No se obtuvo ningún resultado porque se creó un contenedor y al ser una imagen que no hace nada automáticamente se cerró.





5)

```
C:\Users\lauty>docker run -it busybox sh
/ # ps
PID   USER     TIME  COMMAND
    1 root      0:00 sh
    7 root      0:00 ps
/ # uptime
 22:39:47 up 9 min,  0 users,  load average: 0.01, 0.01, 0.00
/ # free
              total        used        free      shared  buff/cache   available
Mem:       16339432      904824    13195568        1616     2239040    15146012
Swap:       4194304           0     4194304
/ # ls -l /
total 40
drwxr-xr-x    2 root     root         12288 Jul 17 18:30 bin
drwxr-xr-x    5 root     root           360 Oct  4 22:39 dev
drwxr-xr-x    1 root     root          4096 Oct  4 22:39 etc
drwxr-xr-x    2 nobody   nobody        4096 Jul 17 18:30 home
drwxr-xr-x    2 root     root          4096 Jul 17 18:30 lib
lrwxrwxrwx    1 root     root             3 Jul 17 18:30 lib64 -> lib
dr-xr-xr-x  433 root     root             0 Oct  4 22:39 proc
drwx------    1 root     root          4096 Oct  4 22:39 root
dr-xr-xr-x   11 root     root             0 Oct  4 22:39 sys
drwxrwxrwt    2 root     root          4096 Jul 17 18:30 tmp
drwxr-xr-x    4 root     root          4096 Jul 17 18:30 usr
drwxr-xr-x    4 root     root          4096 Jul 17 18:30 var
/ #
```

6)

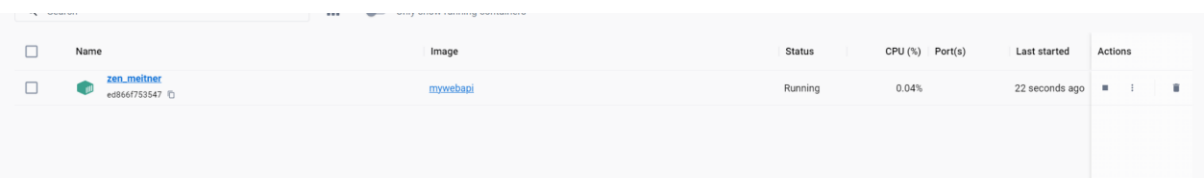
```
C:\Users\lauty>docker ps -a
CONTAINER ID   IMAGE     COMMAND               CREATED              STATUS                          PORTS     NAMES
e82d42b5af07   busybox   "sh"                  About a minute ago   Exited (127) 52 seconds ago               keen_swanson
25922fdea02f   busybox   "echo 'Hola Mundo'"   3 minutes ago        Exited (0) 3 minutes ago                  thirsty_rhod
es
f70224a3f017   busybox   "sh"                  4 minutes ago        Exited (0) 4 minutes ago                  xenodochial_
galileo
232dd4be16dd   busybox   "sh"                  5 minutes ago        Exited (0) 5 minutes ago                  relaxed_jang

C:\Users\lauty>docker rm keen_swanson
keen_swanson

C:\Users\lauty>docker ps -a
CONTAINER ID   IMAGE     COMMAND               CREATED         STATUS                     PORTS     NAMES
25922fdea02f   busybox   "echo 'Hola Mundo'"   3 minutes ago   Exited (0) 3 minutes ago             thirsty_rhodes
f70224a3f017   busybox   "sh"                  5 minutes ago   Exited (0) 4 minutes ago             xenodochial_galileo
232dd4be16dd   busybox   "sh"                  5 minutes ago   Exited (0) 5 minutes ago             relaxed_jang

C:\Users\lauty>
```

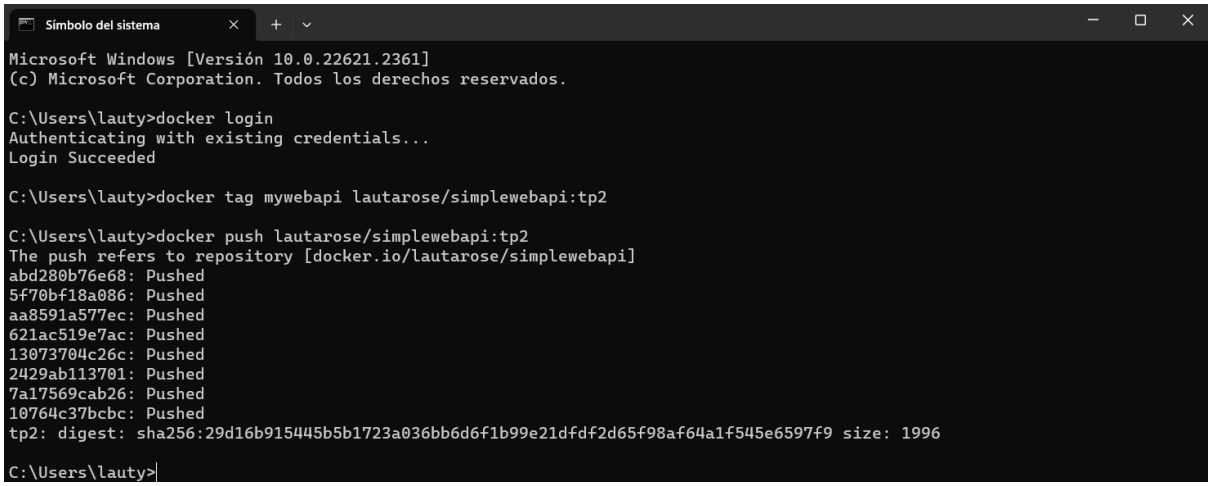
```
C:\Users\lauty>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
25922fdea02ff475716c8f5954bf139abe81d78c084480ae58e45929fb05864d
f70224a3f017259ac493c775201395720d45d434d6209f97513120a6042831d8
232dd4be16dda4429333faf5e83f15c691f2a0ef0eb2962d2d9e9706471386b5

Total reclaimed space: 0B

C:\Users\lauty>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

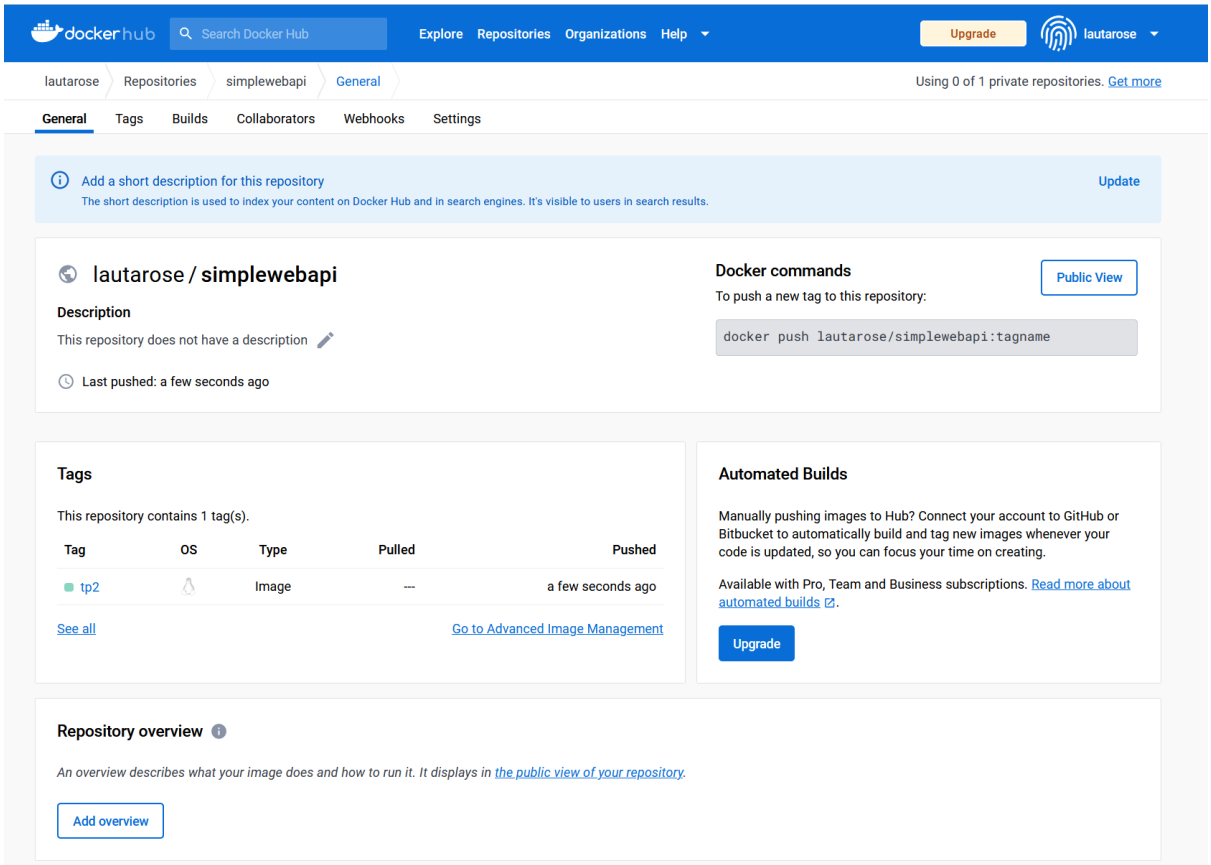
7)

```
Start a build
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI> docker build -t mywebapi .
[+] Building 27.3s (18/18) FINISHED                                                                                 docker:default
 => [internal] load build definition from Dockerfile                                                                         0.0s
 => => transferring dockerfile: 810B                                                                                         0.0s
 => [internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                              0.0s
 => [internal] load metadata for mcr.microsoft.com/dotnet/sdk:7.0                                                            1.7s
 => [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:7.0                                                         1.9s
 => [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:7.0@sha256:2dd6fa19392967b26d59228af0ec481c652b98346ced56a4db1c664  16.8s
 => => resolve mcr.microsoft.com/dotnet/sdk:7.0@sha256:2dd6fa19392967b26d59228af0ec481c652b98346ced56a4db1c66416b4c94  0.0s
 => => sha256:2dd6fa19392967b26d59228af0ec481c652b98346ced56a4db1c66416b4c947f 1.79kB / 1.79kB                               0.0s
 => => sha256:4da9dff9f77f86ae7380a41332b838118f25b96eb27c4ac776325b285a331be0 2.01kB / 2.01kB                               0.0s
 => => sha256:7dbc1adf280e1aa588c033eaa746aa6db327ee16be705740f81741f5e6945c86 31.42MB / 31.42MB                             3.0s
 => => sha256:969d48310aaa1579d8f4f4016de95508e456e5ad40a6ea7317ece466ada626e0 14.97MB / 14.97MB                             2.0s
 => => sha256:1c5a65799405c54dbf710180bd7381f9012b952434130af1ccb14f7a36674581 5.28kB / 5.28kB                               0.0s
 => => sha256:2194c6af68612e5f92a605b2ff1f203b3d8998cb67144a4bff374050854daf38 32.45MB / 32.45MB                             2.3s
 => => sha256:98003354b5affced784d6e1ea8130fc9a6417e4c7745be98a95f908fde955a2a 153B / 153B                                   2.2s
 => => sha256:a5db88be08caa99f41a71144018a814ad2b336546c3d145204f52ae93f95b6e7 10.12MB / 10.12MB                             3.2s
 => => sha256:24ab60b48e57071e6e0782f2b63cbd8ee23520592bcec121000d3a0de6c9abc3 25.37MB / 25.37MB                             3.9s
 => => extracting sha256:7dbc1adf280e1aa588c033eaa746aa6db327ee16be705740f81741f5e6945c86                                    1.4s
 => => sha256:0f0942e0643ed250f1b14a818ff8980a05b2557cd20e72a559cacc482dc37025 180.95MB / 180.95MB                           13.6s
 => => sha256:01dbb2a1b9268bdf7c8a0280e5919b0e5ea27dc85aece94c9f962ae317810919 13.90MB / 13.90MB                             4.8s
 => => extracting sha256:969d48310aaa1579d8f4f4016de95508e456e5ad40a6ea7317ece466ada626e0                                    0.2s
 => => extracting sha256:2194c6af68612e5f92a605b2ff1f203b3d8998cb67144a4bff374050854daf38                                    0.4s
 => => extracting sha256:a5db88be08caa99f41a71144018a814ad2b336546c3d145204f52ae93f95b6e7                                   20.3s
 => => extracting sha256:24ab60b48e57071e6e0782f2b63cbd8ee23520592bcec121000d3a0de6c9abc3                                    0.9s
 => => extracting sha256:0f0942e0643ed250f1b14a818ff8980a05b2557cd20e72a559cacc482dc37025                                    2.8s
 => => extracting sha256:01dbb2a1b9268bdf7c8a0280e5919b0e5ea27dc85aece94c9f962ae317810919                                    0.3s
 => [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:7.0@sha256:933ae169296fe093776749cc47fd60e0a6e85bde607651a3fc6cd9  5.3s
 => => resolve mcr.microsoft.com/dotnet/aspnet:7.0@sha256:933ae169296fe093776749cc47fd60e0a6e85bde607651a3fc6cd97bb5b  0.0s
 => => sha256:969d48310aaa1579d8f4f4016de95508e456e5ad40a6ea7317ece466ada626e0 14.97MB / 14.97MB                             2.0s
 => => sha256:2194c6af68612e5f92a605b2ff1f203b3d8998cb67144a4bff374050854daf38 32.45MB / 32.45MB                             2.4s
 => => sha256:933ae169296fe093776749cc47fd60e0a6e85bde607651a3fc6cd97bb5bc1ce0 1.79kB / 1.79kB                               0.0s
 => => sha256:ddd5b0406dfefbac2b6a382fab3f3b100fa12ff799b398c0209e954ae71c8fbd 1.37kB / 1.37kB                               0.0s
 => => sha256:a4a9087cf895281bc3a723d0c9953228b144bf71ca62696d5d018a39ddcb40b8 2.36kB / 2.36kB                               0.0s
 => => sha256:7dbc1adf280e1aa588c033eaa746aa6db327ee16be705740f81741f5e6945c86 31.42MB / 31.42MB                             3.0s
 => => sha256:98003354b5affced784d6e1ea8130fc9a6417e4c7745be98a95f908fde955a2a 153B / 153B                                   2.2s
 => => sha256:a5db88be08caa99f41a71144018a814ad2b336546c3d145204f52ae93f95b6e7 10.12MB / 10.12MB                             3.2s
 => => extracting sha256:7dbc1adf280e1aa588c033eaa746aa6db327ee16be705740f81741f5e6945c86                                    1.4s
 => => extracting sha256:969d48310aaa1579d8f4f4016de95508e456e5ad40a6ea7317ece466ada626e0                                    0.2s
 => => extracting sha256:2194c6af68612e5f92a605b2ff1f203b3d8998cb67144a4bff374050854daf38                                    0.4s
 => => extracting sha256:98003354b5affced784d6e1ea8130fc9a6417e4c7745be98a95f908fde955a2a                                    0.0s
 => => extracting sha256:a5db88be08caa99f41a71144018a814ad2b336546c3d145204f52ae93f95b6e7                                    0.1s
 => [internal] load build context                                                                                            0.0s
 => => transferring context: 83.83kB                                                                                         0.0s
 => [base 2/2] WORKDIR /app                                                                                                  0.3s
 => [final 1/2] WORKDIR /app                                                                                                 0.1s
 => [build 2/7] WORKDIR /src                                                                                                 0.5s
 => [build 3/7] COPY [SimpleWebAPI/SimpleWebAPI.csproj, SimpleWebAPI/]                                                       0.0s
 => [build 4/7] RUN dotnet restore "SimpleWebAPI/SimpleWebAPI.csproj"                                                        3.5s
 => [build 5/7] COPY . .                                                                                                     0.0s
 => [build 6/7] WORKDIR /src/SimpleWebAPI                                                                                    0.0s
 => [build 7/7] RUN dotnet build "SimpleWebAPI.csproj" -c Release -o /app/build                                              2.8s
 => [publish 1/1] RUN dotnet publish "SimpleWebAPI.csproj" -c Release -o /app/publish /p:UseAppHost=false                    1.6s
```

```
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI> docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
mywebapi     latest    95846c1c66fd   55 seconds ago   216MB
busybox      latest    a416a98b71e2   2 months ago     4.26MB
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI> docker run mywebapi
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app
```




Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.2361]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\lauty>docker login
Authenticating with existing credentials...
Login Succeeded
C:\Users\lauty>docker tag mywebapi lautarose/simplewebapi:tp2
C:\Users\lauty>docker push lautarose/simplewebapi:tp2
The push refers to repository [docker.io/lautarose/simplewebapi]
abd280b76e68: Pushed
5f70bf18a086: Pushed
aa8591a577ec: Pushed
621ac519e7ac: Pushed
13073704c26c: Pushed
2429ab113701: Pushed
7a17569cab26: Pushed
10764c37bcbc: Pushed
tp2: digest: sha256:29d16b915445b5b1723a036bb6d6f1b99e21dfdf2d65f98af64a1f545e6597f9 size: 1996
C:\Users\lauty>


docker hub
Search Docker Hub
Explore
Repositories
Organizations
Help
Upgrade
lautarose
lautarose
Repositories
simplewebapi
General
Using 0 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings
Add a short description for this repository
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.
Update
lautarose / simplewebapi
Description
This repository does not have a description
Last pushed: a few seconds ago
Docker commands
Public View
To push a new tag to this repository:
docker push lautarose/simplewebapi:tagname
Tags
This repository contains 1 tag(s).
Tag
OS
Type
Pulled
Pushed
tp2
Image
---
a few seconds ago
See all
Go to Advanced Image Management
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds.
Upgrade
Repository overview
An overview describes what your image does and how to run it. It displays in the public view of your repository.
Add overview

```
FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443
EXPOSE 5254

FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
WORKDIR /src
COPY ["SimpleWebAPI/SimpleWebAPI.csproj", "SimpleWebAPI/"]
RUN dotnet restore "SimpleWebAPI/SimpleWebAPI.csproj"
COPY . .
WORKDIR "/src/SimpleWebAPI"
RUN dotnet build "SimpleWebAPI.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "SimpleWebAPI.csproj" -c Release -o /app/publish
/p:UseAppHost=false

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "SimpleWebAPI.dll"]
#CMD ["/bin/bash"]
```

La primera línea “FROM” indica cual es la imagen base de nuestro dockerfile.

“AS” se utiliza para nombrar esa imagen base.

“WORKDIR” establece el directorio de trabajo dentro del contenedor en este caso es /app. Todas las instrucciones siguientes se ejecutarán en este directorio.

“EXPOSE” se utiliza para exponer los puertos, en este caso los puertos 80, 443 y 5254.

“FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build” esta línea establece una nueva etapa de construcción (build) y utiliza la imagen “mcr.microsoft.com/dotnet/sdk:7.0” como base.

“ COPY ["SimpleWebAPI/SimpleWebAPI.csproj", "SimpleWebAPI/"] ” Copia el archivo de proyecto (SimpleWebAPI.csproj) al directorio SimpleWebAPI/ en el contenedor. Esto se hace antes de restaurar las dependencias para aprovechar la caché de Docker y evitar la reinstalación de las dependencias si no han cambiado.

“ RUN dotnet restore "SimpleWebAPI/SimpleWebAPI.csproj" “ Restaura las dependencias del proyecto usando dotnet restore.

“ COPY . . “ Copia todos los archivos del contexto de construcción (el directorio donde resides cuando ejecutas el comando docker build) al directorio de trabajo actual en el contenedor (/src).

“ RUN dotnet build "SimpleWebAPI.csproj" -c Release -o /app/build “ Compila la aplicación en modo Release y la salida se coloca en el directorio /app/build dentro del contenedor.

“ FROM build AS publish “ Crea una nueva etapa llamada publish basada en la etapa de construcción anterior (build).

“RUN dotnet publish "SimpleWebAPI.csproj" -c Release -o /app/publish/p:UseAppHost=false” Publica la aplicación en el directorio /app/publish dentro del contenedor. La opción /p:UseAppHost=false se usa para aplicaciones autónomas (self-contained) y significa que no se utilizará un ejecutable de host para la publicación.

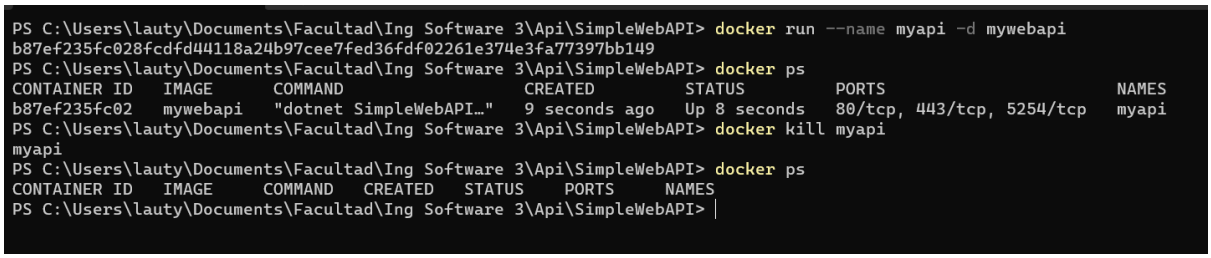
“ FROM base AS final “ Crea una última etapa llamada final basada en la etapa base inicial (base).

“ COPY --from=publish /app/publish . “ Copia los archivos publicados desde la etapa publish al directorio de trabajo actual en la etapa final.

“ ENTRYPOINT ["dotnet", "SimpleWebAPI.dll"] “ Establece el punto de entrada para el contenedor. Cuando el contenedor se inicie, ejecutará el archivo SimpleWebAPI.dll usando dotnet.

“ #CMD ["/bin/bash"] “ Esta línea está comentada (#) y no se utilizará. Si se necesita utilizar una shell para depurar se puede descomentar.

8)



https://localhost/WeatherForecast

No se puede conectar

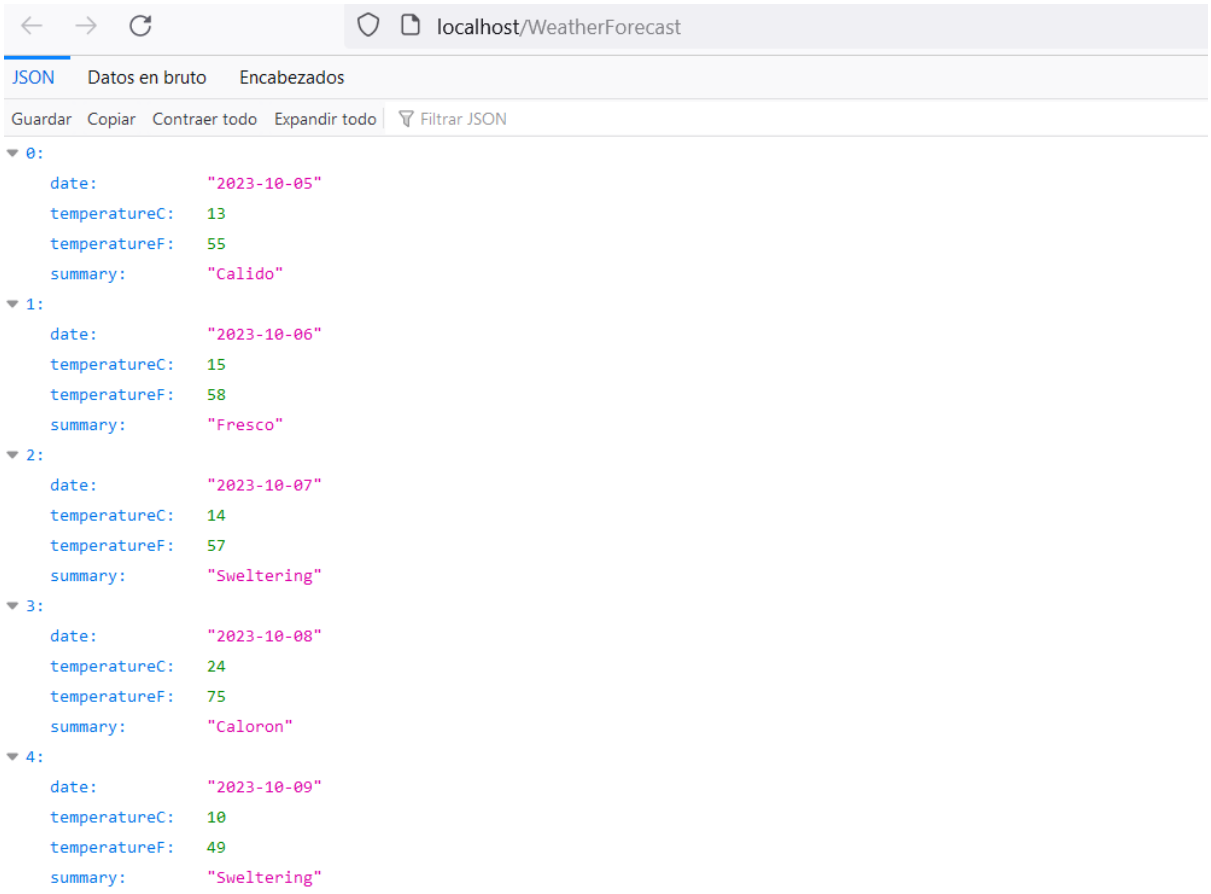
Ocurrió un error al conectarse a localhost.

- El sitio puede no estar disponible temporariamente o estar sobrecargado. Intente nuevamente en unos momentos.
- Si no puede cargar ninguna página, verifique la conexión de su computadora a la red.
- Si su computadora o red están protegidas por un firewall o proxy, asegúrese que Firefox tiene permiso para acceder a la web.

Intente nuevamente

Windows PowerShell

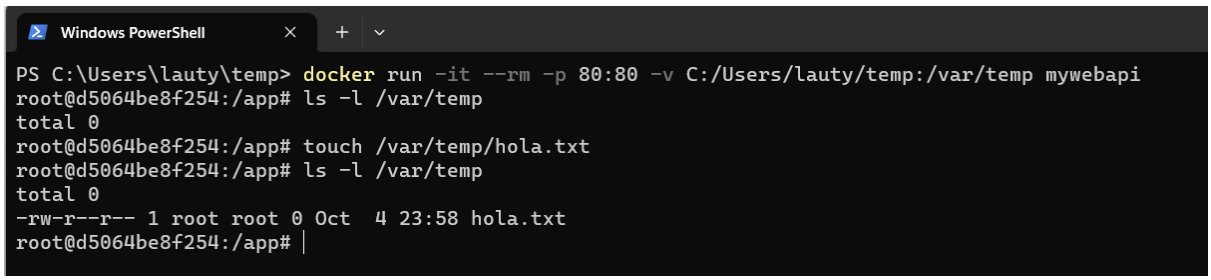
```
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI> docker run --name myapi -d -p 80:80 -p 5254:5254 myweb
api
e452e1b859e83e00f6470008006078b1cd9cac8634c3a0f508f2faa4fe513f2d
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS
           NAMES
e452e1b859e8   mywebapi   "dotnet SimpleWebAPI…"   4 seconds ago   Up 3 seconds   0.0.0.0:80->80/tcp, 0.0.0.0:5254->5254/tcp
, 443/tcp   myapi
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI>
```

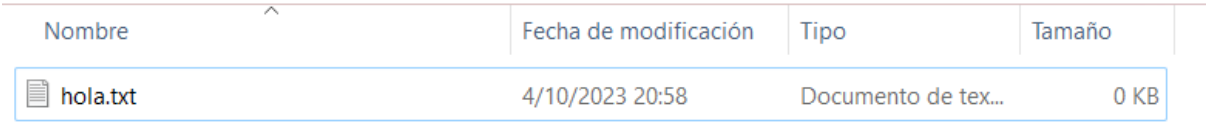


En este caso lo que ocurre es que con el comando “-p” exponemos los puertos del contenedor hacia los puertos del host. Por lo tanto lo que se encuentra por ejemplo en el puerto 80 del contenedor se va a redireccionar al puerto 80 de nuestra pc.

9)

```
PS C:\Users\lauty\Documents\Facultad\Ing Software 3\Api\SimpleWebAPI> docker run -it --rm -p 80:80 mywebapi
root@e6cbb8b77fd3:/app# dotnet SimpleWebAPI.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```





Nombre	Fecha de modificación	Tipo	Tamaño
hola.txt	4/10/2023 20:58	Documento de tex...	0 KB

11)

```
psql (9.4.26)
Type "help" for help.

postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
(3 rows)

postgres=# create database test;
CREATE DATABASE
postgres=# \connect test
You are now connected to database "test" as user "postgres".
test=# create table tabla_a (mensaje varchar(50));
CREATE TABLE
test=# insert into tabla_a (mensaje) values('Hola mundo!');
INSERT 0 1
test=# select * from tabla_a
test-# ;
   mensaje
-------------
 Hola mundo!
(1 row)
```

“ docker run --name my-postgres -e POSTGRES_PASSWORD=mysecretpassword -v $HOME/.postgres:/var/lib/postgresql/data -p 5432:5432 -d postgres:9.4 “ Con este comando, se crea un contenedor de PostgreSQL.

“docker exec -it my-postgres /bin/bash” Con este comando, se ingresa en una sesión interactiva dentro del contenedor de PostgreSQL, lo que te permite ejecutar comandos y realizar operaciones dentro del entorno del contenedor.

12)

```
PS C:\Users\lauty> docker run --name=mysql1 -e MYSQL_ROOT_PASSWORD=password -v $HOME/.sqlserver:/var/lib/mysql -d mysql/
mysql-server
cc33784643686628ea51f5c14d43bd464a50b3d50487d56fe90c7ddfcbbc1469
PS C:\Users\lauty> docker exec -it mysql1 mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database test;
Query OK, 1 row affected (0.01 sec)

mysql> use test
Database changed
mysql> create table tabla_a (mensaje varchar(50));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into tabla_a (mensaje) values('Hola mundo!');
Query OK, 1 row affected (0.00 sec)

mysql> select * from tabla_a
    -> ;
+-------------+
| mensaje     |
+-------------+
| Hola mundo! |
+-------------+
1 row in set (0.00 sec)
```