https://docs.docker.com/engine/reference/commandline/container/

`ATTACH` ✓

`COMMIT` ✓

`CP` ✓

`CREATE` ✓

`DIFF` ✓

`EXEC` ✓

`EXPORT`

`INSPECT` ✓

`KILL` ✓

`LOGS` ✓

`LS` ✓

`PAUSE` ✓

`PORT` ✓

`PRUNE` ✓

`RENAME` ✓

`RESTART` ✓

`RM`

`RUN` ✓

`START` ✓

`STATS` ✓

`STOP` ✓

`TOP`

`UNPAUSE` ✓

`UPDATE`

`WAIT`

`SAVE` ✓

`IMPORT` ✓

https://docs.docker.com/reference/cli/docker/image/

Un poco de repaso

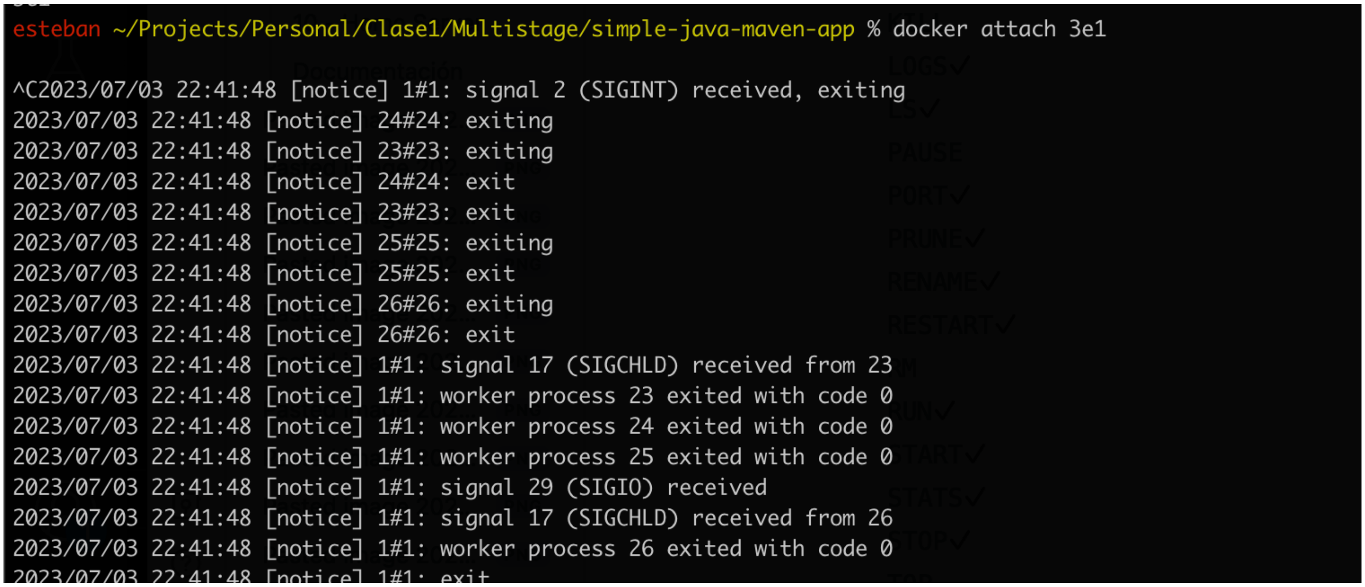
- Ejecutar un contenedor
- Ejecutar un contenedor en background
- Ejecutar un contenedor con puerto expuesto al host
- Deneter un contenedor
- Encende un contenedor

- Reiniciar un contenedor
- Ingresar dentro de un contenedor
    - como ingresar como otro usuario diferente al por defecto

```
        docker exec -u www-data -it 926  bash
```

Ademas de los comando basicos existen otros comandos útiles que nos sirven para manejar contenedores.

`ATTACH`



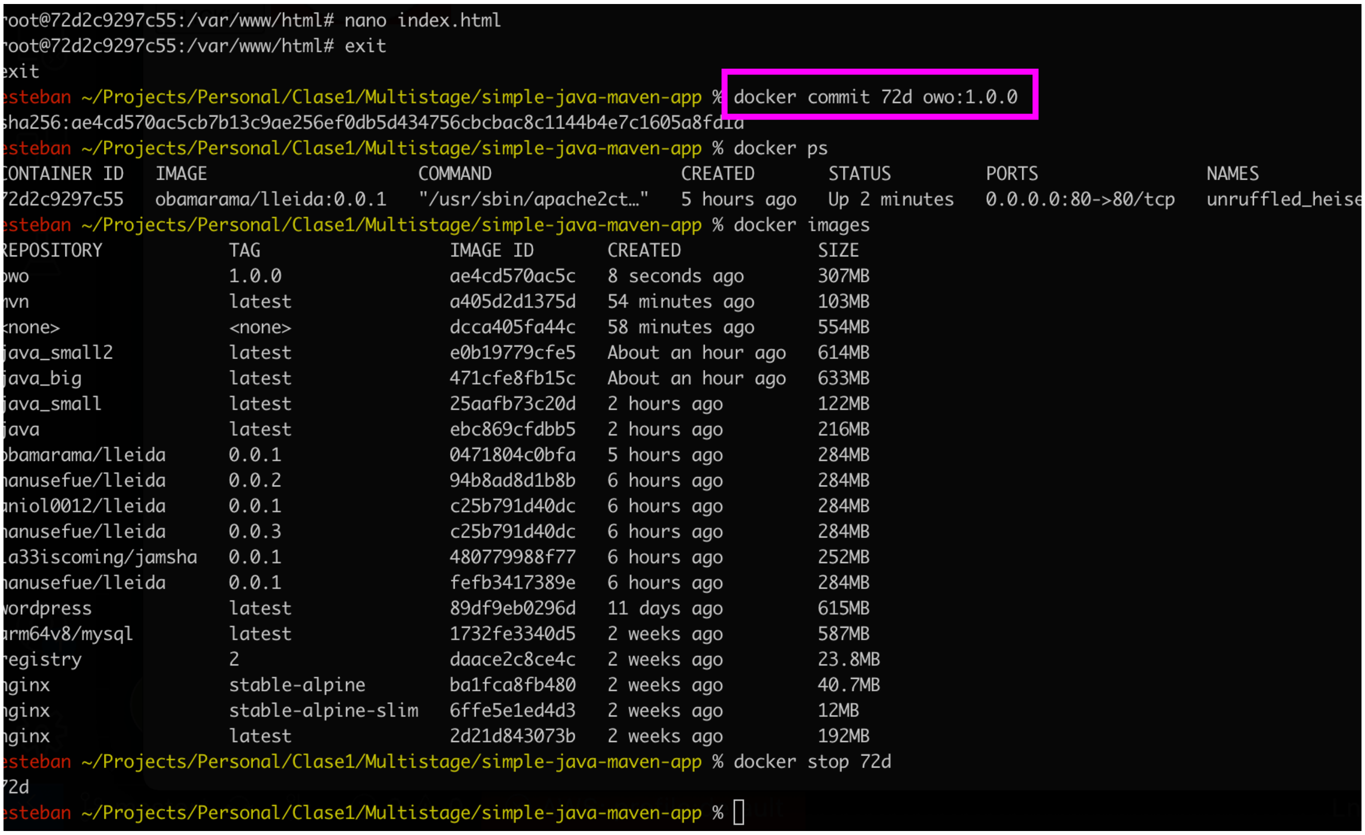
```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker attach 3e1

^C2023/07/03 22:41:48 [notice] 1#1: signal 2 (SIGINT) received, exiting
2023/07/03 22:41:48 [notice] 24#24: exiting
2023/07/03 22:41:48 [notice] 23#23: exiting
2023/07/03 22:41:48 [notice] 24#24: exit
2023/07/03 22:41:48 [notice] 23#23: exit
2023/07/03 22:41:48 [notice] 25#25: exiting
2023/07/03 22:41:48 [notice] 25#25: exit
2023/07/03 22:41:48 [notice] 26#26: exiting
2023/07/03 22:41:48 [notice] 26#26: exit
2023/07/03 22:41:48 [notice] 1#1: signal 17 (SIGCHLD) received from 23
2023/07/03 22:41:48 [notice] 1#1: worker process 23 exited with code 0
2023/07/03 22:41:48 [notice] 1#1: worker process 24 exited with code 0
2023/07/03 22:41:48 [notice] 1#1: worker process 25 exited with code 0
2023/07/03 22:41:48 [notice] 1#1: signal 29 (SIGIO) received
2023/07/03 22:41:48 [notice] 1#1: signal 17 (SIGCHLD) received from 26
2023/07/03 22:41:48 [notice] 1#1: worker process 26 exited with code 0
2023/07/03 22:41:48 [notice] 1#1: exit
```

`COMMIT`

Puede ser útil confirmar los cambios o la configuración del archivo de un contenedor en una nueva imagen. Esto le permite depurar un contenedor ejecutando un shell interactivo o exportar un conjunto de datos de trabajo a otro servidor. Generalmente, es mejor usar Dockerfiles para administrar sus imágenes de una manera documentada y mantenible. Obtenga más información sobre nombres y etiquetas de imágenes válidos.

https://docs.docker.com/engine/reference/commandline/commit/

```
docker commit c3f279d17e0a  svendowideit/testimage:version3
```



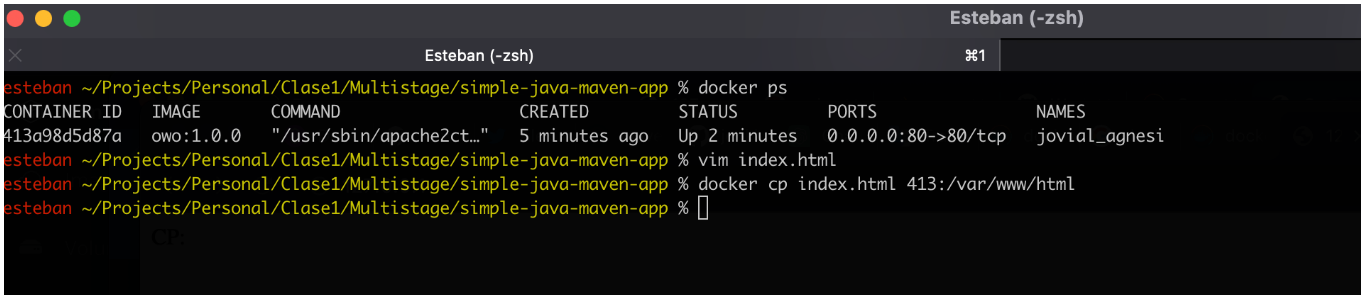
```
root@72d2c9297c55:/var/www/html# nano index.html
root@72d2c9297c55:/var/www/html# exit
exit
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker commit 72d owo:1.0.0
sha256:ae4cd570ac5cb7b13c9ae256ef0db5d434756cbcbac8c1144b4e7c1605a8fd1a
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED       STATUS         PORTS                NAMES
72d2c9297c55   obamarama/lleida:0.0.1 "/usr/sbin/apache2ct…"   5 hours ago   Up 2 minutes   0.0.0.0:80->80/tcp   unruffled_heise
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker images
REPOSITORY          TAG                 IMAGE ID       CREATED              SIZE
owo                 1.0.0               ae4cd570ac5c   8 seconds ago        307MB
mvn                 latest              a405d2d1375d   54 minutes ago       103MB
<none>              <none>              dcca405fa44c   58 minutes ago       554MB
java_small2         latest              e0b19779cfe5   About an hour ago    614MB
java_big            latest              471cfe8fb15c   About an hour ago    633MB
java_small          latest              25aafb73c20d   2 hours ago          122MB
java                latest              ebc869cfdbb5   2 hours ago          216MB
obamarama/lleida    0.0.1               0471804c0bfa   5 hours ago          284MB
manusefue/lleida    0.0.2               94b8ad8d1b8b   6 hours ago          284MB
aniol0012/lleida    0.0.1               c25b791d40dc   6 hours ago          284MB
manusefue/lleida    0.0.3               c25b791d40dc   6 hours ago          284MB
la33iscoming/jamsha 0.0.1               480779988f77   6 hours ago          252MB
manusefue/lleida    0.0.1               fefb3417389e   6 hours ago          284MB
wordpress           latest              89df9eb0296d   11 days ago          615MB
arm64v8/mysql       latest              1732fe3340d5   2 weeks ago          587MB
registry            2                   daace2c8ce4c   2 weeks ago          23.8MB
nginx               stable-alpine       ba1fca8fb480   2 weeks ago          40.7MB
nginx               stable-alpine-slim  6ffe5e1ed4d3   2 weeks ago          12MB
nginx               latest              2d21d843073b   2 weeks ago          192MB
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker stop 72d
72d
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app %
```

`––change`

`CP`

https://docs.docker.com/engine/reference/commandline/cp/

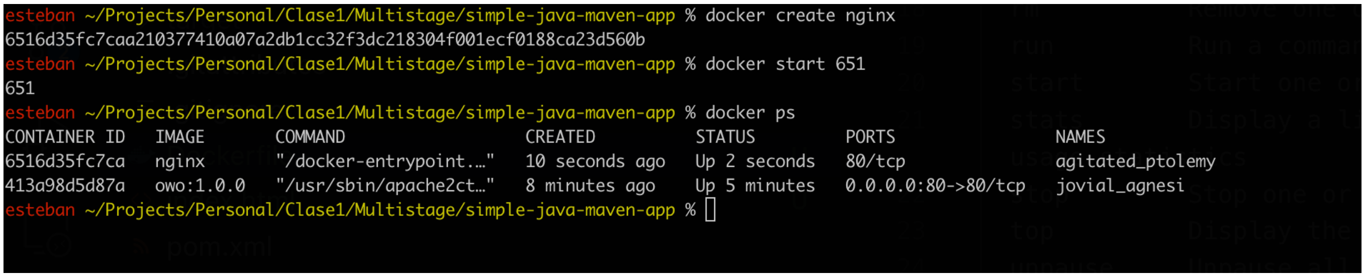
Permite copiar de el Host al contenedor o desde el contenedor al Host



```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS         PORTS                NAMES
413a98d5d87a   owo:1.0.0   "/usr/sbin/apache2ct…"   5 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp   jovial_agnesi
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % vim index.html
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker cp index.html 413:/var/www/html
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app %
```

`CREATE`

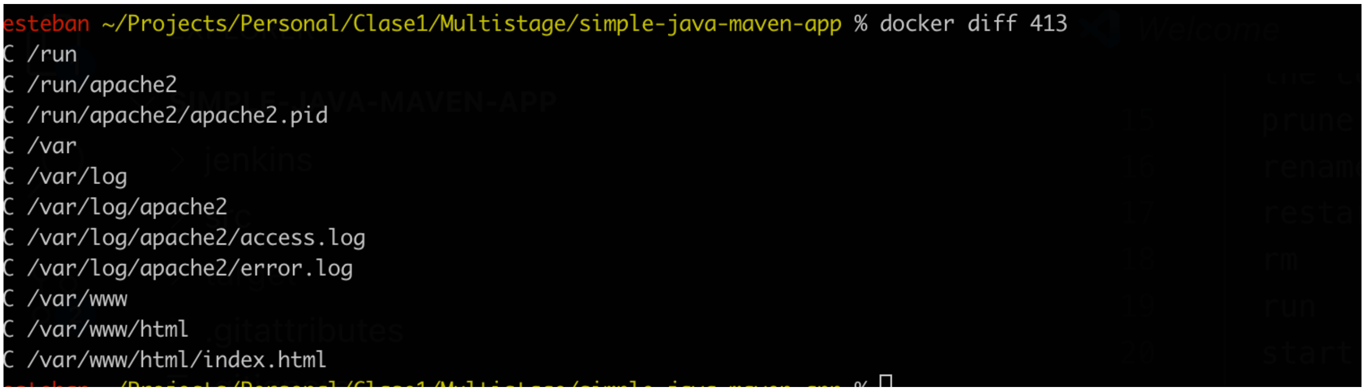
Es identico al `Docker run` pero no inicializa el contenedor, para eso es necesario hacer un `docker star ID`



```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker create nginx
6516d35fc7caa210377410a07a2db1cc32f3dc218304f001ecf0188ca23d560b
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker start 651
651
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED          STATUS         PORTS                NAMES
6516d35fc7ca   nginx       "/docker-entrypoint.…"   10 seconds ago   Up 2 seconds   80/tcp               agitated_ptolemy
413a98d5d87a   owo:1.0.0   "/usr/sbin/apache2ct…"   8 minutes ago    Up 5 minutes   0.0.0.0:80->80/tcp   jovial_agnesi
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app %
```

DIFF

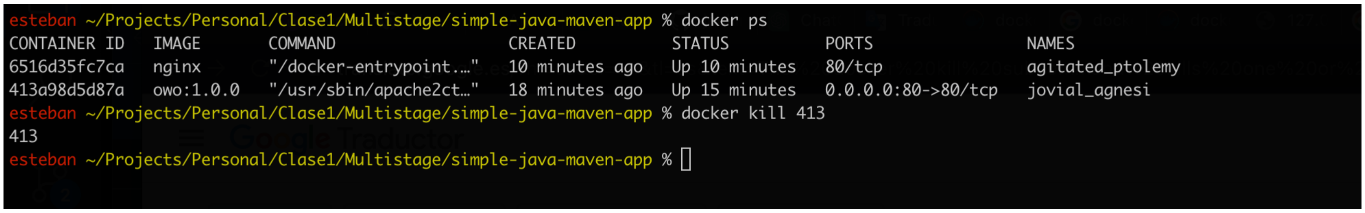
Enumera los archivos y directorios modificados en el sistema de archivos de un contenedor desde que se creó el contenedor. Se rastrean tres tipos diferentes de cambio:



```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker diff 413
C /run
C /run/apache2
C /run/apache2/apache2.pid
C /var
C /var/log
C /var/log/apache2
C /var/log/apache2/access.log
C /var/log/apache2/error.log
C /var/www
C /var/www/html
C /var/www/html/index.html
```

KILL

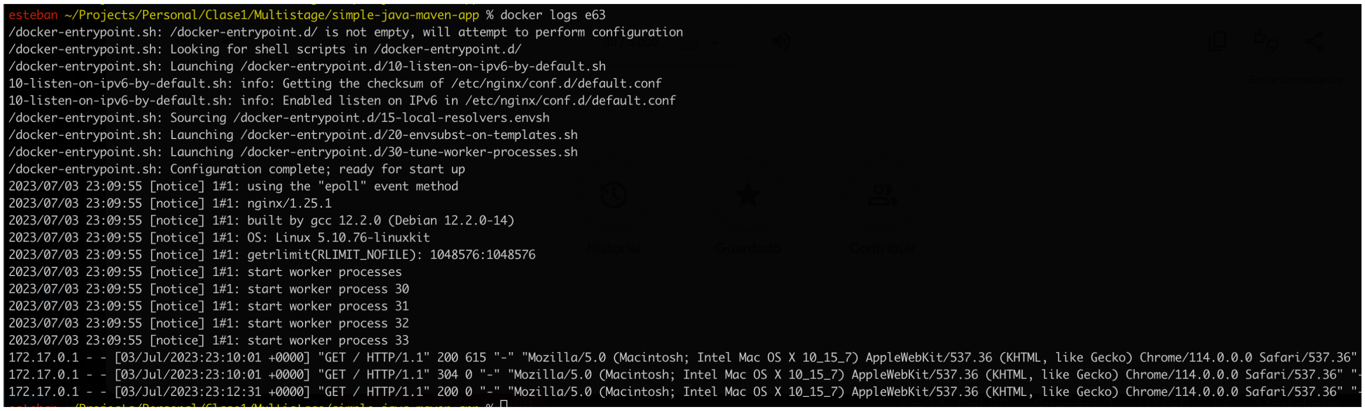
El subcomando docker kill elimina uno o más contenedores. Al proceso principal dentro del contenedor se le envía la señal SIGKILL (predeterminada), o la señal que se especifica con la opción --signal. Puede hacer referencia a un contenedor por su ID, prefijo de ID o nombre.



```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED          STATUS          PORTS                NAMES
6516d35fc7ca   nginx       "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   80/tcp               agitated_ptolemy
413a98d5d87a   owo:1.0.0   "/usr/sbin/apache2ct…"   18 minutes ago   Up 15 minutes   0.0.0.0:80->80/tcp   jovial_agnesi
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker kill 413
413
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % 
```

LOGS

El comando docker logs recupera por lotes los registros presentes en el momento de la ejecución.



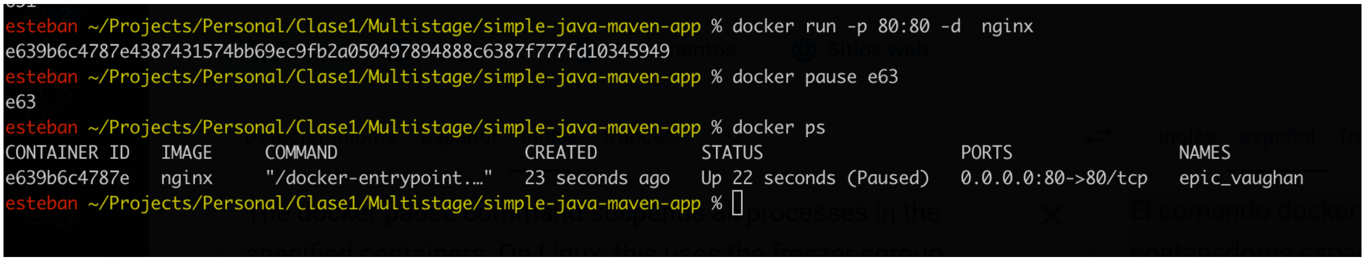
```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker logs e63
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/07/03 23:09:55 [notice] 1#1: using the "epoll" event method
2023/07/03 23:09:55 [notice] 1#1: nginx/1.25.1
2023/07/03 23:09:55 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/07/03 23:09:55 [notice] 1#1: OS: Linux 5.10.76-linuxkit
2023/07/03 23:09:55 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/07/03 23:09:55 [notice] 1#1: start worker processes
2023/07/03 23:09:55 [notice] 1#1: start worker process 30
2023/07/03 23:09:55 [notice] 1#1: start worker process 31
2023/07/03 23:09:55 [notice] 1#1: start worker process 32
2023/07/03 23:09:55 [notice] 1#1: start worker process 33
172.17.0.1 - - [03/Jul/2023:23:10:01 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36"
172.17.0.1 - - [03/Jul/2023:23:10:01 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36" "-
172.17.0.1 - - [03/Jul/2023:23:12:31 +0000] "GET / HTTP/1.1" 200 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36" "-
```

PAUSE

El comando docker pause suspende todos los procesos en los contenedores especificados. En Linux, esto usa el congelador cgroup. Tradicionalmente, al suspender un proceso se utiliza la señal SIGSTOP, que es observable por el proceso que se suspende. Con el cgroup del congelador, el proceso no sabe, y no puede capturar, que se suspende y luego se reanuda.

```
docker pause contenedor
```

```
docker ps
```

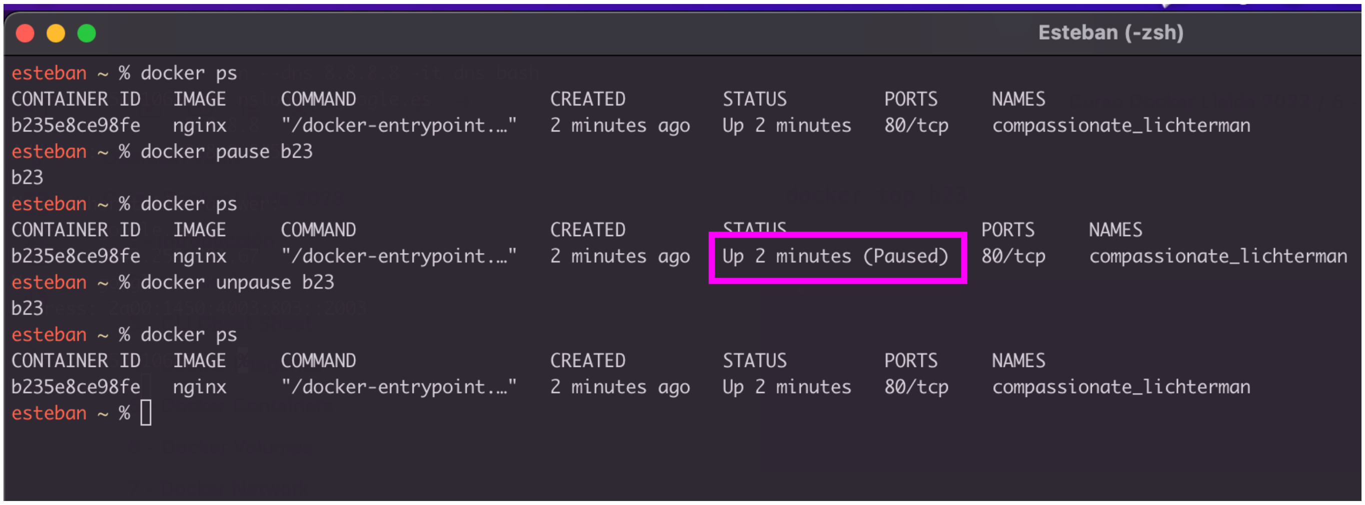


```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker run -p 80:80 -d  nginx
e639b6c4787e4387431574bb69ec9fb2a050497894888c6387f777fd10345949
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker pause e63
e63
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                   PORTS                NAMES
e639b6c4787e   nginx     "/docker-entrypoint.…"   23 seconds ago   Up 22 seconds (Paused)   0.0.0.0:80->80/tcp   epic_vaughan
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % 
```

`UNPAUSE` ✓

https://docs.docker.com/engine/reference/commandline/unpause/

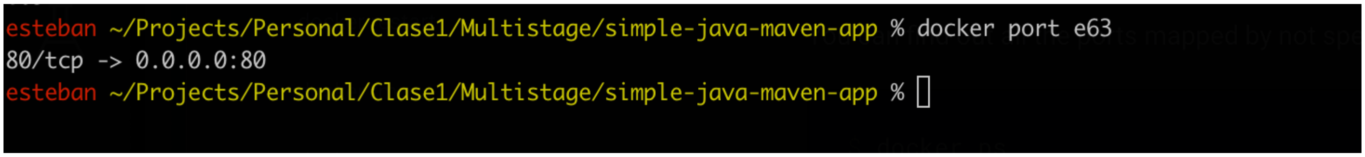
```
docker unpause container
```



```
esteban ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
b235e8ce98fe   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   80/tcp    compassionate_lichterman
esteban ~ % docker pause b23
b23
esteban ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS                  PORTS     NAMES
b235e8ce98fe   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes (Paused)   80/tcp    compassionate_lichterman
esteban ~ % docker unpause b23
b23
esteban ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
b235e8ce98fe   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   80/tcp    compassionate_lichterman
esteban ~ % 
```

`PORT`

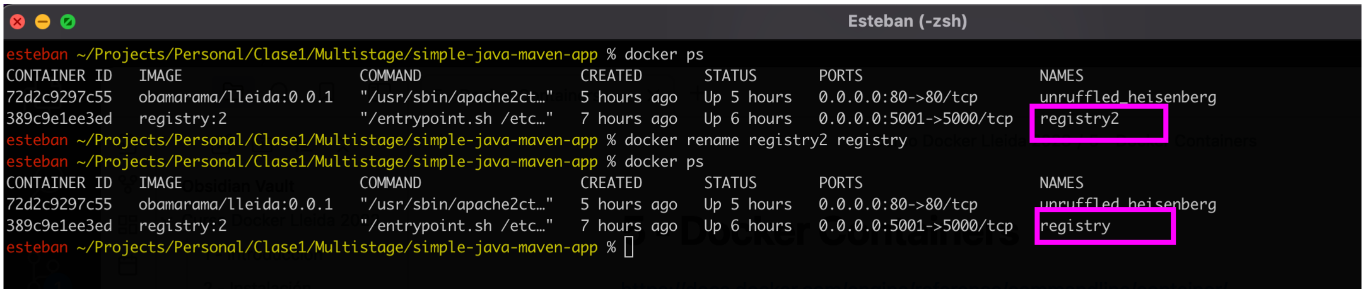
Muestra todos los puertos mapeados de un contenedor



```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker port e63
80/tcp -> 0.0.0.0:80
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % 
```

`RENAME`

```
docker rename registry2 registry
```



```
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED       STATUS       PORTS                    NAMES
72d2c9297c55   obamarama/lleida:0.0.1   "/usr/sbin/apache2ct…"   5 hours ago   Up 5 hours   0.0.0.0:80->80/tcp       unruffled_heisenberg
389c9e1ee3ed   registry:2               "/entrypoint.sh /etc…"   7 hours ago   Up 6 hours   0.0.0.0:5001->5000/tcp   registry2
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker rename registry2 registry
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED       STATUS       PORTS                    NAMES
72d2c9297c55   obamarama/lleida:0.0.1   "/usr/sbin/apache2ct…"   5 hours ago   Up 5 hours   0.0.0.0:80->80/tcp       unruffled_heisenberg
389c9e1ee3ed   registry:2               "/entrypoint.sh /etc…"   7 hours ago   Up 6 hours   0.0.0.0:5001->5000/tcp   registry
esteban ~/Projects/Personal/Clase1/Multistage/simple-java-maven-app % 
```

`RESTART` ✓

https://docs.docker.com/engine/reference/commandline/restart/

Reiniciar uno o más contenedores

```
docker restart my_container
```

`RM`

https://docs.docker.com/engine/reference/commandline/rm/

Eliminar uno o mas contenedores

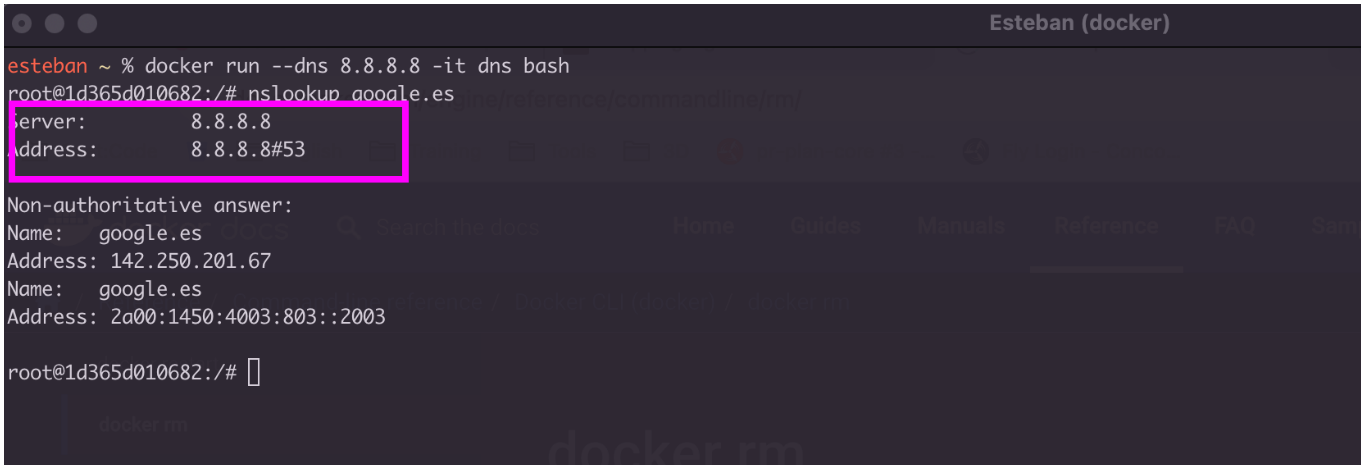
```
docker rm containerID
```

`RUN` ✓

https://docs.docker.com/engine/reference/commandline/run/

Crear y ejecutar un nuevo contenedor a partir de una imagen

```
docker run ——dns 8.8.8.8 —it dns
```



```
esteban ~ % docker run --dns 8.8.8.8 -it dns bash
root@1d365d010682:/# nslookup google.es
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   google.es
Address: 142.250.201.67
Name:   google.es
Address: 2a00:1450:4003:803::2003

root@1d365d010682:/# 
```

`START` ✓

https://docs.docker.com/engine/reference/commandline/start/

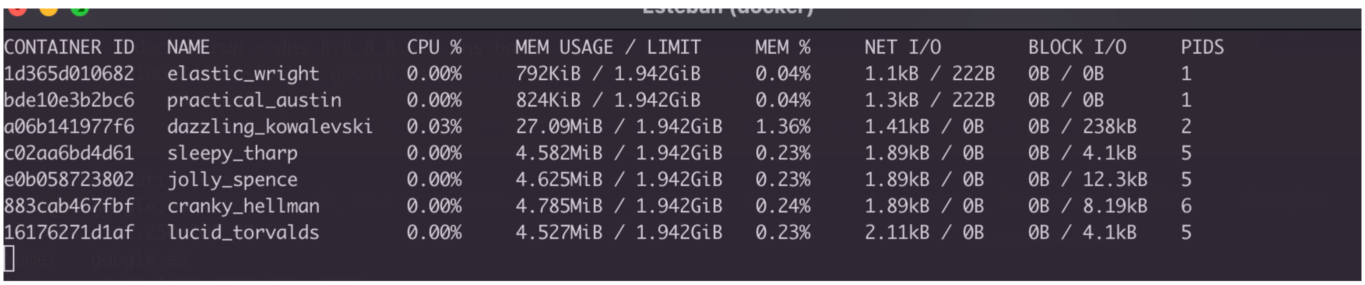
Iniciar un contenedor o mas que están parados

```
docker start contenedor
```

STATS ✓

Muestra los recursos utilizados por un contenedor/s en tiempo real

```
docker stats
```



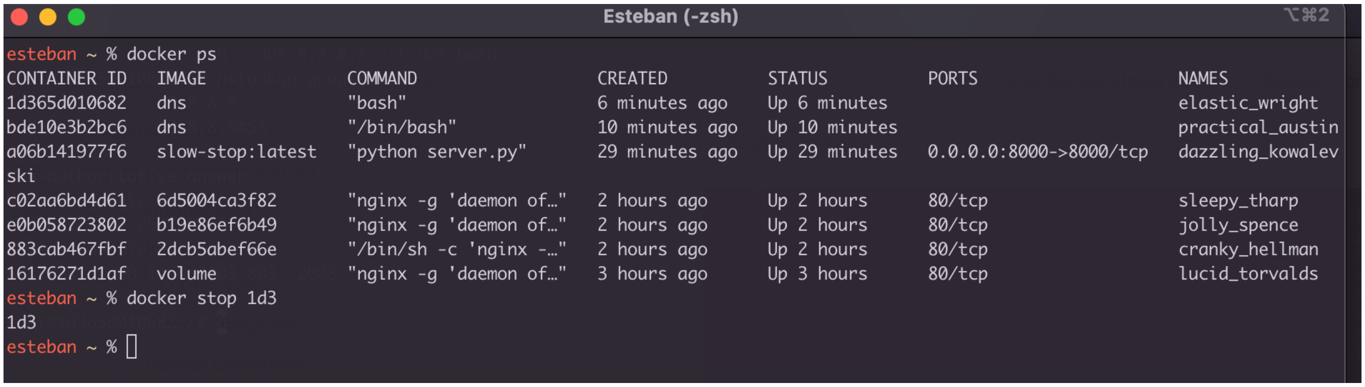
Esteban (docker)

```
CONTAINER ID   NAME                CPU %     MEM USAGE / LIMIT     MEM %     NET I/O          BLOCK I/O     PIDS
1d365d010682   elastic_wright      0.00%     792KiB / 1.942GiB     0.04%     1.1kB / 222B     0B / 0B       1
bde10e3b2bc6   practical_austin    0.00%     824KiB / 1.942GiB     0.04%     1.3kB / 222B     0B / 0B       1
a06b141977f6   dazzling_kowalevski 0.03%     27.09MiB / 1.942GiB   1.36%     1.41kB / 0B      0B / 238kB    2
c02aa6bd4d61   sleepy_tharp        0.00%     4.582MiB / 1.942GiB   0.23%     1.89kB / 0B      0B / 4.1kB    5
e0b058723802   jolly_spence        0.00%     4.625MiB / 1.942GiB   0.23%     1.89kB / 0B      0B / 12.3kB   5
883cab467fbf   cranky_hellman      0.00%     4.785MiB / 1.942GiB   0.24%     1.89kB / 0B      0B / 8.19kB   6
16176271d1af   lucid_torvalds      0.00%     4.527MiB / 1.942GiB   0.23%     2.11kB / 0B      0B / 4.1kB    5
```

STOP ✓

parar uno o más contenedores.

```
docker stop my_container
```

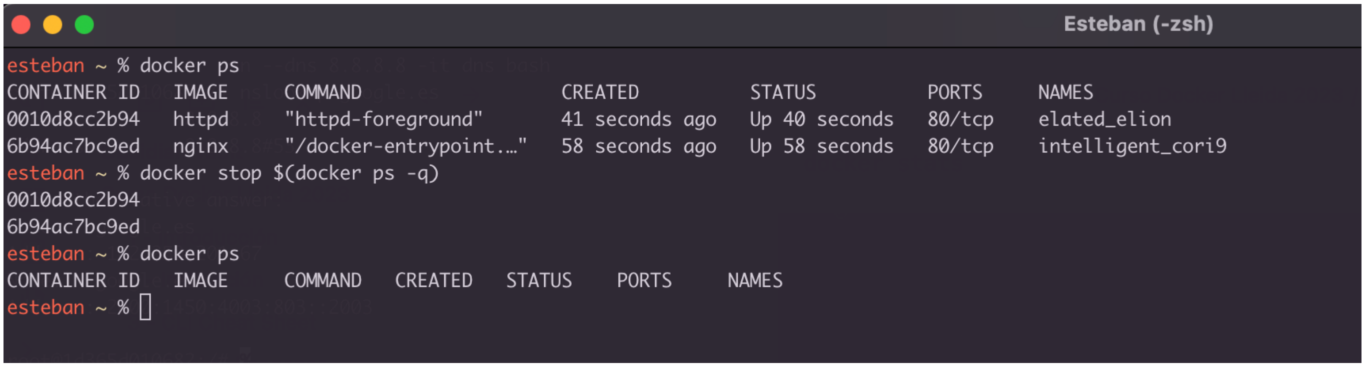


Esteban (-zsh)

```
esteban ~ % docker ps
CONTAINER ID   IMAGE               COMMAND                   CREATED          STATUS          PORTS                    NAMES
1d365d010682   dns                 "bash"                    6 minutes ago    Up 6 minutes                             elastic_wright
bde10e3b2bc6   dns                 "/bin/bash"               10 minutes ago   Up 10 minutes                            practical_austin
a06b141977f6   slow-stop:latest    "python server.py"        29 minutes ago   Up 29 minutes   0.0.0.0:8000->8000/tcp   dazzling_kowalev
ski
c02aa6bd4d61   6d5004ca3f82        "nginx -g 'daemon of…"    2 hours ago      Up 2 hours      80/tcp                   sleepy_tharp
e0b058723802   b19e86ef6b49        "nginx -g 'daemon of…"    2 hours ago      Up 2 hours      80/tcp                   jolly_spence
883cab467fbf   2dcb5abef66e        "/bin/sh -c 'nginx -…"    2 hours ago      Up 2 hours      80/tcp                   cranky_hellman
16176271d1af   volume              "nginx -g 'daemon of…"    3 hours ago      Up 3 hours      80/tcp                   lucid_torvalds
esteban ~ % docker stop 1d3
1d3
esteban ~ %
```

como parar todos los contenedores

```
docker stop $(docker ps -q)
```



Esteban (-zsh)

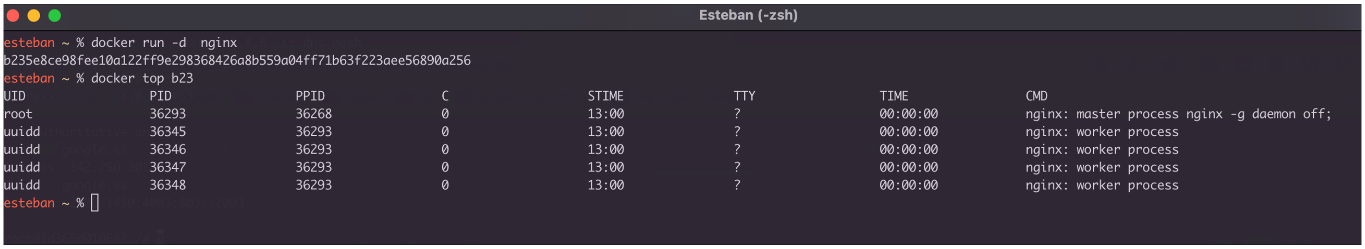
```
esteban ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                   CREATED          STATUS          PORTS     NAMES
0010d8cc2b94   httpd     "httpd-foreground"        41 seconds ago   Up 40 seconds   80/tcp    elated_elion
6b94ac7bc9ed   nginx     "/docker-entrypoint.…"    58 seconds ago   Up 58 seconds   80/tcp    intelligent_cori9
esteban ~ % docker stop $(docker ps -q)
0010d8cc2b94
6b94ac7bc9ed
esteban ~ % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
esteban ~ %
```

`TOP`

https://docs.docker.com/engine/reference/commandline/top/

Mostrar los procesos en ejecución de un contenedor

```
docker top b23
```



Esteban (-zsh)

```
esteban ~ % docker run -d  nginx
b235e8ce98fee10a122ff9e298368426a8b559a04ff71b63f223aee56890a256
esteban ~ % docker top b23
UID                 PID                 PPID                C                   STIME               TTY                 TIME                CMD
root                36293               36268               0                   13:00               ?                   00:00:00            nginx: master process nginx -g daemon off;
uuidd               36345               36293               0                   13:00               ?                   00:00:00            nginx: worker process
uuidd               36346               36293               0                   13:00               ?                   00:00:00            nginx: worker process
uuidd               36347               36293               0                   13:00               ?                   00:00:00            nginx: worker process
uuidd               36348               36293               0                   13:00               ?                   00:00:00            nginx: worker process
esteban ~ % 
```

`UPDATE`

https://docs.docker.com/engine/reference/commandline/update/

Actualizar la configuración de uno o más contenedores

`WAIT`

https://docs.docker.com/engine/reference/commandline/wait/

Bloquee hasta que uno o más contenedores se detengan, luego imprima sus códigos de salida.

```
#!/bin/sh

#below command will run a container named ‘my-con’ in detach mode and output ‘Hello’ on STDOUT #five times and remove it once container is stopped.

docker container run --rm -d --name my-con alpine ash -c "for i in 1 2 3 4 5; do echo 'Hello'; done"

#we have named the container ‘my-con’ in the above command so we are using that name in below #command, however, we can use contaienr ID but we need to get the id that required to write some #additional lines of code

exit_code="$(docker wait my-con)"

#Compare the status code

if [ $exit_code -eq "0" ]

then
```

```
echo “Job has been completed successfully.”

else

echo “Job has encountered some error.”

fi
```

`EXPORT`
https://docs.docker.com/reference/cli/docker/container/export/
Exporta el contenedor, pero no sus volumenes asociados.

```
docker export --output="latest.tar" red_panda
```

Repaso
Imagenes

1- Crear una imagen con python (Dockerfile) que se ejecute un pequeño script y muestre un "Hola Mundo" por el navegador. https://flask.palletsprojects.com/en/3.0.x/quickstart/#a-minimal-application

- Utilizar una imagen base de Ubuntu
- Utilizar un requirements.txt para instalar los módulos de python necesarios.
- Utilizar el .dockerignore
- Subir la imagen creada a un repositorio public y privado(local) registry2
- Crear una version ligera
- Crear tags
- Compartir con los compañeros y profesor

2- Modificar el Dockerfile anterior para reducir/optimzar aun más el tamaño de nuestra imagen (cambiando la imagen Base).

- Al finalizar el ejercicio miraremos que imagen pesa menos.

Contenedores

Ejercicio 1
Revisar cada uno de los comando explicados e intentar utilizarlos.
Crear un contenedor con nginx y probar los comando que hemos visto anteriormente
`ATTACH` , `COMMIT` , `CP` , `CREATE` , `DIFF` , `EXEC` , `EXPORT` , `INSPECT` , `KILL` , `LOGS`

, `LS` , `PAUSE` , `PORT` , `PRUNE` , `RENAME` , `RESTART` , `RM` , `RUN` , `START` , `STATS` , `STOP` , `TOP` , `UNPAUSE` , `UPDATE` , `WAIT` , `SAVE` , `IMPORT`

Ejercico 2

Crear un contenedor con nginx
Copiar un nuevo index con el comando CP desde la terminal
Y crear un commit de este nuevo contenedor
Limitar la memoria con un UPDATE a ~4 GB
Hacer un export y despues un Import

```
docker create -p 80:80 nginx
docker start containerID
touch index.html
vim index.html
docker cp index.html contenedorID:/usr/share/nginx/html/index.html
docker commit containerID nginx:v2
docker update --memory 4GB --memory-swap 5GB containerID
docker stats
docker export containerID -o backup.tar
docker import backup.tar nginx:v3
docker run -d -p 80:80 nginx:v3 nginx -g "daemon off;"
```