

2021

Practica III – Docker Containers

MARKEL ORALLO

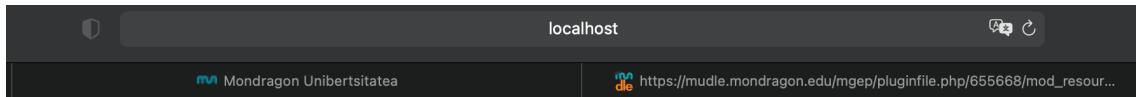
Poner en marcha un contenedor

Poner en marcha un contenedor desde una imagen. La imagen ejecuta un servidor web en el puerto 8080 (ejemplo de clase).

Ejecutando `docker run -name webstatic -p 8888:80 -d dockersamples/static-site` ponemos en marcha el contenedor.

```
markel@MacBook-Pro-de-Markel ~ % docker run --name webstatic -p 8888:80 -d dockersamples/static-site
Unable to find image 'dockersamples/static-site:latest' locally
latest: Pulling from dockersamples/static-site
fdd5d7827f33: Pull complete
a3ed95caeb02: Pull complete
716f7a5f3082: Pull complete
7b10f03a0309: Pull complete
aff3ab7e9c39: Pull complete
Digest: sha256:daa686c61d7d239b7977e72157997489db49f316b9b9af3909d9f10fd28b2dec
Status: Downloaded newer image for dockersamples/static-site:latest
11ae8edec18ce79249ca2e5da6ceaf08d2093e1910ea3e781393ce0c32a02d59
```

Una vez que el contenedor está en marcha y los puertos mapeados conectar mediante un web browser a la dirección IP del Docker host en el puerto 8888. ¿Cuál es la respuesta?



Hello Docker!

This is being served from a **docker** container running Nginx.

Parar y borrar todos los contenedores del sistema.

```
markel@MacBook-Pro-de-Markel ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
11ae8edec18c dockersamples/static-site "/bin/sh -c 'cd /usr..'" About a minute ago Up About a minute 443/tcp, 0.0.0.0:8888->80/tcp, :::8888->80/tcp webstatic
markel@MacBook-Pro-de-Markel ~ % docker stop 11ae8edec18c
11ae8edec18c
markel@MacBook-Pro-de-Markel ~ % docker container rm 11ae8edec18c
11ae8edec18c
markel@MacBook-Pro-de-Markel ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ %
```

Contenedores interactivos

Crear un contenedor utilizando la imagen centos:7 y conectar su bash terminal en modo interactivo

Verificar el sistema de ficheros del contenedor y crea un nuevo fichero.

```
#echo'Hello there...>test.txt
```

Sal del contenedor mediante el comando exit.

```
[markel@MacBook-Pro-de-Markel ~ % docker run -it centos:7 /bin/bash
[[root@ac4f8fa60e53 /]]# ls
anaconda-post.log  dev  home  lib64  mnt  proc  run  srv  tmp  var
bin                etc  lib   media  opt  root  sbin  sys  usr
[[root@ac4f8fa60e53 /]]# echo "Hello there..." > test.txt
[[root@ac4f8fa60e53 /]]# ls
anaconda-post.log  dev  home  lib64  mnt  proc  run  srv  test.txt  usr
bin                etc  lib   media  opt  root  sbin  sys  tmp      var
[[root@ac4f8fa60e53 /]]# exit
exit
markel@MacBook-Pro-de-Markel ~ %
```

Ejecuta el siguiente comando: \$ docker container run -it centos:7 bash

Trata de encontrar el fichero test.txt en este nuevo contenedor. Sal de este contenedor de la misma forma que saliste del anterior.

```
[markel@MacBook-Pro-de-Markel ~ % docker run -it centos:7 /bin/bash
[[root@2e76b57b5d6f /]]# ls
anaconda-post.log  dev  home  lib64  mnt  proc  run  srv  tmp  var
bin                etc  lib   media  opt  root  sbin  sys  usr
[[root@2e76b57b5d6f /]]# exit
exit
markel@MacBook-Pro-de-Markel ~ %
```

Reconectar contenedores

Lista todos los contenedores que tiene el Docker host (también los que están Stopped)

```
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE      COMMAND     CREATED      STATUS          PORTS      NAMES
2e76b57b5d6f  centos:7  "/bin/bash"  2 minutes ago  Exited (0) About a minute ago
ac4f8fa60e53  centos:7  "/bin/bash"  4 minutes ago  Exited (0) 3 minutes ago
cool_lewin
gifted_shamir
```

Reinicia el contenedor que creaste primero utilizando su CONTAINER ID (la que tiene la imagen centos).

Verifica que el contenedor está en marcha

Chequea si existe el fichero test.txt y verifica su contenido.

```
markel@MacBook-Pro-de-Markel ~ % docker start ac4f8fa60e53
ac4f8fa60e53
markel@MacBook-Pro-de-Markel ~ % docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
ac4f8fa60e53        centos:7          "/bin/bash"         5 minutes ago     Up 4 seconds       gifted_shamir
markel@MacBook-Pro-de-Markel ~ % docker exec -it ac4f8fa60e53 /bin/bash
[root@ac4f8fa60e53 ~]# ls
anaconda-post.log  dev  home  lib64  mnt  proc  run  srv  test.txt  usr
bin    etc  lib   media  opt  root  sbin  sys  tmp   var
[root@ac4f8fa60e53 ~]# cat test.txt
Hello there...
[root@ac4f8fa60e53 ~]#
```

Ejecuta: \$ docker container ls -a --no-trunc. ¿Cuál es su output? ¿Qué nos aporta el flag –no-trunc?

```
markel@MacBook-Pro-de-Markel ~ % docker container ls -a --no-trunc
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS
                PORTS               NAMES
2e76b57b5d6fde9b93ac1697799a54c9417626601c2d6fd9776fe69f60af64e1   centos:7      "/bin/bash"         9 minutes ago     Exited (0) 9
minutes ago          cool_lewin
ac4f8fa60e53b07446e001cbb0f6732ee75b0c693341a0f21c6d65fc56e6a02   centos:7      "/bin/bash"         11 minutes ago    Up 6 minutes
gifted_shamir
```

```
markel@MacBook-Pro-de-Markel ~ % docker container ls --help
Usage:  docker container ls [OPTIONS]

List containers

Aliases:
  ls, ps, list

Options:
  -a, --all           Show all containers (default shows just running)
  -f, --filter filter Filter output based on conditions provided
  --format string     Pretty-print containers using a Go template
  -n, --last int      Show n last created containers (includes all states) (default -1)
  -l, --latest        Show the latest created container (includes all states)
  --no-trunc          Don't truncate output
  -q, --quiet         Only display container IDs
  -s, --size          Display total file sizes
markel@MacBook-Pro-de-Markel ~ % docker container ls -a --format {{.ID}}
2e76b57b5d6f
ac4f8fa60e53
markel@MacBook-Pro-de-Markel ~ % docker ps -aq
2e76b57b5d6f
ac4f8fa60e53
```

El flag –no-trunc hace que la salida del comando docker container ls -a no salga cortada. Normalmente el solo se muestran los primeros 12 caracteres del ID y ahora se muestra entero.

Listar los contenedores pero solo mostrando su ID.

```
markel@MacBook-Pro-de-Markel ~ % docker ps -aq
2e76b57b5d6f
ac4f8fa60e53
markel@MacBook-Pro-de-Markel ~ % docker container ls -a --format {{.ID}}
2e76b57b5d6f
ac4f8fa60e53
```

Listar el ultimo contenedor que hemos creado.

```
markel@MacBook-Pro-de-Markel ~ % docker ps --latest
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2e76b57b5d6f centos:7 "/bin/bash" 16 minutes ago Exited (0) 15 minutes ago
markel@MacBook-Pro-de-Markel ~ % docker container ls --latest
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2e76b57b5d6f centos:7 "/bin/bash" 16 minutes ago Exited (0) 16 minutes ago
```

Listar los contenedores que están en estado EXIT mediante el flag -filter.

```
markel@MacBook-Pro-de-Markel ~ % docker ps -a --filter "status=exited"
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2e76b57b5d6f centos:7 "/bin/bash" 18 minutes ago Exited (0) 18 minutes ago
markel@MacBook-Pro-de-Markel ~ % docker container ls -a --filter "status=exited"
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2e76b57b5d6f centos:7 "/bin/bash" 18 minutes ago Exited (0) 18 minutes ago
```

Borra todos los contenedores

```
markel@MacBook-Pro-de-Markel ~ % docker stop $(docker ps -q)
ac4f8fa60e53
markel@MacBook-Pro-de-Markel ~ % docker rm $(docker ps -aq)
ac4f8fa60e53
```

Poner en marcha un nuevo contenedor con la imagen centos:7 y lanza el comando ping al 127.0.0.1

```
markel@MacBook-Pro-de-Markel ~ % docker run -it centos:7 /bin/bash
[root@bcc59bfa205c /]# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.032 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.079 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.206 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.265 ms
```

Inspeccionar los logs generados por este contenedor mediante el comando log

```
markel@MacBook-Pro-de-Markel ~ % docker logs bcc59bfa205c
[root@bcc59bfa205c /]# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.032 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.079 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.206 ms
```

Inspeccionar la información de este contenedor mediante el comando inspect

```
markel@MacBook-Pro-de-Markel ~ % docker inspect bcc59bfa205c
[
  {
    "Id": "bcc59bfa205c13371d94e689aeff525c919d85c8fe66118abc3c522e957c1676",
    "Created": "2021-09-20T19:18:43.9201792Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 127,
      "Error": "",
      "StartedAt": "2021-09-20T19:18:44.1973808Z",
      "FinishedAt": "2021-09-20T19:33:02.909019Z"
    },
    "Image": "sha256:eeb6ee3f44bd0b5103bb561b4c16bcb82328cfef5809ab675bb17ab3a16c517c9",
    "ResolvConfPath": "/var/lib/docker/containers/bcc59bfa205c13371d94e689aeff525c919d85c8fe66118abc3c522e957c1676-resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/bcc59bfa205c13371d94e689aeff525c919d85c8fe66118abc3c522e957c1676-hostname",
    "HostsPath": "/var/lib/docker/containers/bcc59bfa205c13371d94e689aeff525c919d85c8fe66118abc3c522e957c1676-hosts",
    "LogPath": "/var/lib/docker/containers/bcc59bfa205c13371d94e689aeff525c919d85c8fe66118abc3c522e957c1676-json.log",
    "Name": "/lucid_mirzakhani"
  }
]
```

Borrar contenedores

Pon en marcha tres contenedores en modo background.

```
markel@MacBook-Pro-de-Markel ~ % docker run -d nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a330b6cecb98: Pull complete
e0ad2c0621bc: Pull complete
9e56c3e0e6b7: Pull complete
09f31c94adc6: Pull complete
32b26e9cdb83: Pull complete
20ab512bbb07: Pull complete
Digest: sha256:853b221d3341add7aaadf5f81dd088ea943ab9c918766e295321294b035f3f3e
Status: Downloaded newer image for nginx:latest
4c273130cdcc54870123f228a585cd802eefef07f79477cbb4e48288297d11d2df
markel@MacBook-Pro-de-Markel ~ % docker run -d nginx:latest
96256812c9eda837f96208286fd38e501be2dd031c8d741c6b76ac40dd034f1d
markel@MacBook-Pro-de-Markel ~ % docker run -d nginx:latest
d225995f8661aa42979bf1c36e9453c2f0dd807269a505b1c7e28f82d9ca4118
markel@MacBook-Pro-de-Markel ~ % docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS             NAMES
d225995f8661        nginx:latest       "/docker-entrypoint..."   3 seconds ago     Up 2 seconds      80/tcp            sharp_cerf
96256812c9ed        nginx:latest       "/docker-entrypoint..."   4 seconds ago     Up 4 seconds      80/tcp            trusting_swartz
4c273130cdcc5       nginx:latest       "/docker-entrypoint..."   7 seconds ago     Up 6 seconds      80/tcp            bold_kapitsa
```

Lista los contenedores que estan en estado exit (mediante filter)

```
markel@MacBook-Pro-de-Markel ~ % docker ps -a --filter "status=exited"
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS             NAMES
markel@MacBook-Pro-de-Markel ~ % docker run -d centos:7
473def508dc920b8ffe51bc572e938e12941bec4ea2e281a016cb6830d1cef4
markel@MacBook-Pro-de-Markel ~ % docker ps -a --filter "status=exited"
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS             NAMES
473def508dc9        centos:7          "/bin/bash"         2 seconds ago     Exited (0) 1 second ago      infallible_haslett
```

Borra los contenedores que estan parados (exit) y confirma que están borrados

```
markel@MacBook-Pro-de-Markel ~ % docker ps -a --filter "status=exited"
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
473def508dc9 centos:7 "/bin/bash" 2 seconds ago Exited (0) 1 second ago infallible_haslett
markel@MacBook-Pro-de-Markel ~ % docker rm $(docker ps -aq --filter "status=exited")
473def508dc9
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d225995f8661 nginx:latest "/docker-entrypoint..." 4 minutes ago Up 4 minutes 80/tcp sharp_cerf
96256812c9ed nginx:latest "/docker-entrypoint..." 4 minutes ago Up 4 minutes 80/tcp trusting_swartz
4c273130cdc5 nginx:latest "/docker-entrypoint..." 4 minutes ago Up 4 minutes 80/tcp bold_kapitsa
markel@MacBook-Pro-de-Markel ~ %
```

Ahora, haz lo mismo pero con los contenedores que están en marcha. En este caso, existen diferentes opciones para hacerlo. Describe cada una de ellas.

Parar los contenedores que están en marcha para que se encuentren en el estado EXITED y después borrarlos.

```
markel@MacBook-Pro-de-Markel ~ % docker stop $(docker ps -q) && docker rm $(docker ps -aq)
d225995f8661
96256812c9ed
4c273130cdc5
d225995f8661
96256812c9ed
4c273130cdc5
markel@MacBook-Pro-de-Markel ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ %
```

Directamente forzar el borrado de los contenedores que están en marcha. Estos recibirán la señal SIGKILL y se borran.

```
markel@MacBook-Pro-de-Markel ~ % docker run -d nginx:latest
f85ad488490e98a43ef6d7c0904bba53e1a3096d0b390f65b9dcc85524cd65c4
markel@MacBook-Pro-de-Markel ~ % docker run -d nginx:latest
318cd4290ff5b8b9824835cd776829d1ba0cf49e881f5487696d8f3bedaf635d
markel@MacBook-Pro-de-Markel ~ % docker run -d nginx:latest
ca9fb4fac60f7b6bb7f3df1608ec27602f736cda350ea5eebf10a349f6459a2
markel@MacBook-Pro-de-Markel ~ % docker rm --force $(docker ps -q)
ca9fb4fac60f
318cd4290ff5
f85ad488490e
markel@MacBook-Pro-de-Markel ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ %
```

Borrar todos los contenedores del sistema.

```
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
3bd10d4ca251 centos:7 "/bin/bash" 9 seconds ago Exited (0) 8 seconds ago charming_hungle
3dfd59e5f214 centos:7 "/bin/bash" 9 seconds ago Exited (0) 9 seconds ago amazing_bose
06e1a6d98f54 centos:7 "/bin/bash" 11 seconds ago Exited (0) 10 seconds ago dazzling_goldall
9bae641ffff3f nginx:latest "/docker-entrypoint..." 15 seconds ago Up 15 seconds 80/tcp vigilant_wiles
66afe4eef70b nginx:latest "/docker-entrypoint..." 16 seconds ago Up 16 seconds 80/tcp nifty_gates
d8371fc00d1 nginx:latest "/docker-entrypoint..." 18 seconds ago Up 17 seconds 80/tcp amazing_banach
markel@MacBook-Pro-de-Markel ~ % docker rm -f $(docker ps -aq)
3bd10d4ca251
3dfd59e5f214
06e1a6d98f54
9bae641ffff3f
66afe4eef70b
d8371fc00d1
markel@MacBook-Pro-de-Markel ~ % docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
markel@MacBook-Pro-de-Markel ~ %
```