

Practicando con Docker

Obtener la lista de imágenes disponibles

Una vez hemos bajado todas las imágenes podemos listar las imágenes disponibles mediante la orden:

[docker](#) images

Si lo ejecutamos deberemos obtener una imagen similar a la siguiente:

```
root@usuario-VirtualBox: /home/usuario
root@usuario-VirtualBox:/home/usuario# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
mysql               latest             5fac85ee2c68       15 hours ago
408.2 MB
httpd               latest             c24f66af34b4       4 days ago
177.3 MB
ubuntu              latest             747cb2d60bbe       6 days ago
122 MB
wordpress           latest             d3f0cddf9493       6 days ago
408.4 MB
php                 latest             c342f917459a       7 days ago
370.7 MB
debian              latest             874e27b628fd       7 days ago
100.1 MB
ubuntu              14.04             dea1945146b9       4 weeks ago
188 MB
root@usuario-VirtualBox:/home/usuario#
```

Creamos un contenedor partiendo de una imagen (la imagen ubuntu:14.04). Podéis probar con otras imágenes.

[docker](#) run -it --name usuario ubuntu:14.04 /bin/bash

```
root@8d58327aa7fd: /
root@usuario-VirtualBox:/home/usuario# docker run -it --name usuario ubuntu:14.04 /bin/bash
root@8d58327aa7fd:/#
```

Al arrancar el contenedor obtendremos un prompt de administrador para un sistema Ubuntu 14.04.

Ejecutar las siguientes ordenes dentro de ese contenedor:

- `uname -a`

```
root@8d58327aa7fd:/# uname -a
Linux 8d58327aa7fd 4.10.0-35-generic #39~16.04.1-Ubuntu SMP Wed Sep 13 09:02:42 UTC 2017 x86_64 x86_64
x86_64 GNU/Linux
root@8d58327aa7fd:/#
```

- `sudo apt-get update`

```
root@8d58327aa7fd:/# sudo apt-get update
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://archive.ubuntu.com trusty InRelease
Get:2 http://archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:3 http://archive.ubuntu.com trusty-backports InRelease [65.9 kB]
Get:4 http://archive.ubuntu.com trusty Release.gpg [933 B]
Get:5 http://archive.ubuntu.com trusty Release [58.5 kB]
Get:6 http://security.ubuntu.com trusty-security/universe Sources [77.2 kB]
Get:7 http://security.ubuntu.com trusty-security/main amd64 Packages [842 kB]
Get:8 http://archive.ubuntu.com trusty-updates/universe Sources [242 kB]
Get:9 http://archive.ubuntu.com trusty-updates/main amd64 Packages [1283 kB]
Get:10 http://security.ubuntu.com trusty-security/restricted amd64 Packages [17.8 kB]
Get:11 http://security.ubuntu.com trusty-security/universe amd64 Packages [243 kB]
Get:12 http://archive.ubuntu.com trusty-updates/restricted amd64 Packages [21.1 kB]
Get:13 http://security.ubuntu.com trusty-security/multiverse amd64 Packages [3999 B]
Get:14 http://archive.ubuntu.com trusty-updates/universe amd64 Packages [555 kB]
Get:15 http://archive.ubuntu.com trusty-updates/multiverse amd64 Packages [15.5 kB]
Get:16 http://archive.ubuntu.com trusty-backports/main amd64 Packages [14.8 kB]
Get:17 http://archive.ubuntu.com trusty-backports/restricted amd64 Packages [40 B]
Get:18 http://archive.ubuntu.com trusty-backports/universe amd64 Packages [52.6 kB]
Get:19 http://archive.ubuntu.com trusty-backports/multiverse amd64 Packages [1396 B]
Get:20 http://archive.ubuntu.com trusty/universe Sources [7926 kB]
Get:21 http://archive.ubuntu.com trusty/main amd64 Packages [1743 kB]
Get:22 http://archive.ubuntu.com trusty/restricted amd64 Packages [16.0 kB]
Get:23 http://archive.ubuntu.com trusty/universe amd64 Packages [7589 kB]
Get:24 http://archive.ubuntu.com trusty/multiverse amd64 Packages [169 kB]
Fetched 21.1 MB in 14s (1485 kB/s)
Reading package lists... Done
root@8d58327aa7fd:/#
```

- `sudo apt-get install nano`

```
root@8d58327aa7fd:/# sudo apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 194 kB of archives.
After this operation, 614 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu/ trusty/main nano amd64 2.2.6-1ubuntu1 [194 kB]
Fetched 194 kB in 0s (591 kB/s)
Selecting previously unselected package nano.
(Reading database ... 11569 files and directories currently installed.)
Preparing to unpack .../nano_2.2.6-1ubuntu1_amd64.deb ...
Unpacking nano (2.2.6-1ubuntu1) ...
Setting up nano (2.2.6-1ubuntu1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
root@8d58327aa7fd:/#
```

- ifconfig (comprobar el tipo de ip que se le ha concedido al contenedor)

```
root@8d58327aa7fd:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5627 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5072 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21649346 (21.6 MB)  TX bytes:371235 (371.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@8d58327aa7fd:/#
```

- ifconfig en otra terminal del host [docker](#) (fuera del contenedor, en nuestro equipo)

```
root@usuario-VirtualBox:/home/usuario# ifconfig
docker0    Link encap:Ethernet  direcciónHW 02:42:cf:3a:b8:8f
          Direc. inet:172.17.0.1  Difus.:0.0.0.0  Másc:255.255.0.0
          Dirección inet6: fe80::42:cfff:fe3a:b88f/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:5072 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:5600 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:300227 (300.2 KB)  TX bytes:21645348 (21.6 MB)

enp0s3     Link encap:Ethernet  direcciónHW 08:00:27:2f:66:3f
          Direc. inet:192.168.1.55  Difus.:192.168.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::6eec:2820:d47b:fbab/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:398309 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:106999 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:593950022 (593.9 MB)  TX bytes:7991095 (7.9 MB)

lo         Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO  MTU:65536  Métrica:1
          Paquetes RX:1198 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:1198 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:171442 (171.4 KB)  TX bytes:171442 (171.4 KB)

veth6bd427c Link encap:Ethernet  direcciónHW a2:8e:e7:5a:3c:ea
          Dirección inet6: fe80::a08e:e7ff:fe5a:3cea/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:5072 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:5627 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:371235 (371.2 KB)  TX bytes:21649346 (21.6 MB)

root@usuario-VirtualBox:/home/usuario#
```

- ping 8.8.8.8 desde el contenedor (podrás comprobar que funciona)

```
root@8d58327aa7fd:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=14.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=14.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=14.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=14.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=55 time=17.5 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=55 time=15.2 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=55 time=14.7 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=55 time=13.7 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=55 time=15.1 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=55 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=55 time=14.6 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=55 time=14.8 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=55 time=14.9 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=55 time=15.2 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=55 time=14.1 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=55 time=14.4 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=55 time=12.7 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=55 time=14.7 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=55 time=15.0 ms
^C
--- 8.8.8.8 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18031ms
rtt min/avg/max/mdev = 10.872/14.589/17.529/1.237 ms
root@8d58327aa7fd:/#
```

- ping a la ip de tu host [docker](#)

```
root@8d58327aa7fd:/# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=63 time=0.755 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=63 time=0.121 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=63 time=0.203 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=63 time=0.132 ms
^C
--- 192.168.1.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.121/0.302/0.755/0.263 ms
root@8d58327aa7fd:/#
```

- exit

```
root@8d58327aa7fd:/# exit
exit
root@usuario-VirtualBox:/home/usuario#
```


Al hacer exit hemos salido del contenedor y el contenedor, por cómo está definido y por como lo hemos arrancado dejará de ejecutarse.

Listar los contenedor de están en ejecución

Para obtener la lista de contenedor en ejecución ejecutaremos la orden:

`docker ps`

Si queremos ver todos los contenedores incluso aquellos que se han parado ejecutaremos la siguiente orden:

`docker ps -a`

Podemos ver el resultado de estas dos órdenes en el siguiente pantallazo.

```
root@usuario-VirtualBox:/home/usuario# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
root@usuario-VirtualBox:/home/usuario# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
ES            8d58327aa7fd  ubuntu:14.04          "/bin/bash"      7 minutes ago Exited (0)    About a minute ago    usu
ario
root@usuario-VirtualBox:/home/usuario#
```

Podemos ver que el estado de nuestro contenedor es Exited, es decir está parado.

Rearrancar y parar contenedores.

La forma de arrancar un contenedor parado para poder interactuar con él va a depender de cómo se ha creado y de como se ha arrancado usando run pero de manera general usaremos la orden:

`docker start usuario`

Al hacer esto rearrancamos el contenedor pero no podremos interactuar con él. Si queremos comprobar que el contenedor está funcionando ejecutaremos **`docker ps`**

Para parar un contenedor

`docker stop usuario`

```
root@usuario-VirtualBox:/home/usuario# docker start usuario
usuario
root@usuario-VirtualBox:/home/usuario# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
8d58327aa7fd  ubuntu:14.04  "/bin/bash"            8 minutes ago Up 15 seconds        usuario
root@usuario-VirtualBox:/home/usuario# docker stop usuario
usuario
root@usuario-VirtualBox:/home/usuario# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
root@usuario-VirtualBox:/home/usuario#
```

Vemos que el contenedor está UP, es decir funcionando, pero no se nos ha mostrado un shell. Si hubieramos querido un shell bash al arrancar el contedor deberíamos haber usado el flag -ai. Para el contenedor y ejecuta.

- **`docker start -ai usuario`**

```
root@usuario-VirtualBox:/home/usuario# docker start -ai usuario
root@8d58327aa7fd:/#
```

7. Ejecutar órdenes en un contenedor que está ejecutándose pero si necesidad de tener que ejecutar un shell

Si el contenedor está funcionando (comprueba que está UP con [docker ps](#)) para ejecutar una orden en ese contenedor usaremos la orden:

[docker](#) exec usuario ls

```
root@usuario-VirtualBox:/home/usuario# docker start usuario
usuario
root@usuario-VirtualBox:/home/usuario# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
8d58327aa7fd   ubuntu:14.04   "/bin/bash"             12 minutes ago Up 3 seconds   usuario
root@usuario-VirtualBox:/home/usuario# docker exec usuario ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

[docker](#) exec usuario sudo apt-get update

```
root@usuario-VirtualBox:/home/usuario# docker exec usuario sudo apt-get update
Hit http://security.ubuntu.com trusty-security InRelease
Ign http://archive.ubuntu.com trusty InRelease
Hit http://archive.ubuntu.com trusty-updates InRelease
Hit http://archive.ubuntu.com trusty-backports InRelease
Hit http://archive.ubuntu.com trusty Release.gpg
Hit http://archive.ubuntu.com trusty Release
Hit http://security.ubuntu.com trusty-security/universe Sources
Hit http://security.ubuntu.com trusty-security/main amd64 Packages
Hit http://archive.ubuntu.com trusty-updates/universe Sources
Hit http://security.ubuntu.com trusty-security/restricted amd64 Packages
Hit http://archive.ubuntu.com trusty-updates/main amd64 Packages
Hit http://security.ubuntu.com trusty-security/universe amd64 Packages
Hit http://archive.ubuntu.com trusty-updates/restricted amd64 Packages
Hit http://security.ubuntu.com trusty-security/multiverse amd64 Packages
Hit http://archive.ubuntu.com trusty-updates/universe amd64 Packages
Hit http://archive.ubuntu.com trusty-updates/multiverse amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/main amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://archive.ubuntu.com trusty/universe Sources
Hit http://archive.ubuntu.com trusty/main amd64 Packages
Hit http://archive.ubuntu.com trusty/restricted amd64 Packages
Hit http://archive.ubuntu.com trusty/universe amd64 Packages
Hit http://archive.ubuntu.com trusty/multiverse amd64 Packages
Reading package lists...
```

[docker](#) exec ping 8.8.8.8

```
root@usuario-VirtualBox:/home/usuario# docker exec usuario ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=15.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=18.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=14.0 ms
^C
root@usuario-VirtualBox:/home/usuario#
```

Y si queremos lanzar un bash en un contenedor funcionado:

[docker](#) exec -it usuario /bin/bash

```
root@usuario-VirtualBox:/home/usuario# docker exec -it usuario /bin/bash
root@8d58327aa7fd:/#
```

Arrancando contenedor con servicios

Arrancar contenedores con servicios

Para explicar como arrancar contenedores con servicios veremos un ejemplo de contenedor que ejecute un servidor apache:

[docker](#) run -d -p 8080:80 --name usuario httpd

```
root@usuario-VirtualBox:/home/usuario# docker run -d -p 8080:80 --name usuario2 httpd
a6196d15d71c74476b048d3170debf66c0c7b42e6226ae0d0cb736119aacdc5a
root@usuario-VirtualBox:/home/usuario#
```

Una vez arrancado podemos ver que el contenedor está funcionando con **[docker ps](#)**

```
root@usuario-VirtualBox:/home/usuario# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a6196d15d71c	httpd	"httpd-foreground"	About a minute ago	Up About a minute	0.0.0.0:8080->80/tcp	usuario2
8d58327aa7fd	ubuntu:14.04	"/bin/bash"	32 minutes ago	Up 20 minutes		usuario

```
root@usuario-VirtualBox:/home/usuario#
```

Y como está funcionando si queremos conectarnos al contenedor para interactuar con el podremos ejecutar:

[docker](#) exec -it a1 /bin/bash

```
root@usuario-VirtualBox:/home/usuario# docker exec -it usuario2 /bin/bash
root@a6196d15d71c:/usr/local/apache2#
```

Arrancar un contenedor de la imagen de MySQL

[docker](#) run -d --name m1 -e MYSQL_ROOT_PASSWORD=root -p 3316:3306 mysql

```
root@usuario-VirtualBox:/home/usuario# docker run -d --name m1 -e MYSQL_ROOT_PASSWORD=root -p 3316:3306 mysql
9a9d07fed3decd2576a718503e28777a85e9ebb75c6ee84a0fcbfae69b5aa25c
root@usuario-VirtualBox:/home/usuario#
```

Aquí hemos fijado la variable de entorno (-e) **MYSQL_ROOT_PASSWORD** al arrancar el contenedor.

Si queremos arrancar un cliente mysql en el contenedor

[docker](#) exec -it m1 mysql -u root -p

```
root@usuario-VirtualBox:/home/usuario# docker exec -it m1 mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

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> █
```

Y como he redireccionado los puertos nos pedirá la contraseña que hayamos puesto más arriba (en nuestro caso root)

Y si queremos usar el cliente local tenemos que indicar tanto el host como el nuevo puerto.

mysql -u root -h 192.168.1.55 -P 3316 -p

```
mysql> mysql -u root -h 192.168.1.55 -P 3316 -p
-> █
```

Compartiendo directorios entre el host y el contenedor

Para esto utilizaremos el flag -v de [docker](#) run

Por ejemplo

[docker](#) run -it -v /home/usuario :/datos --name d3 debian /bin/bash

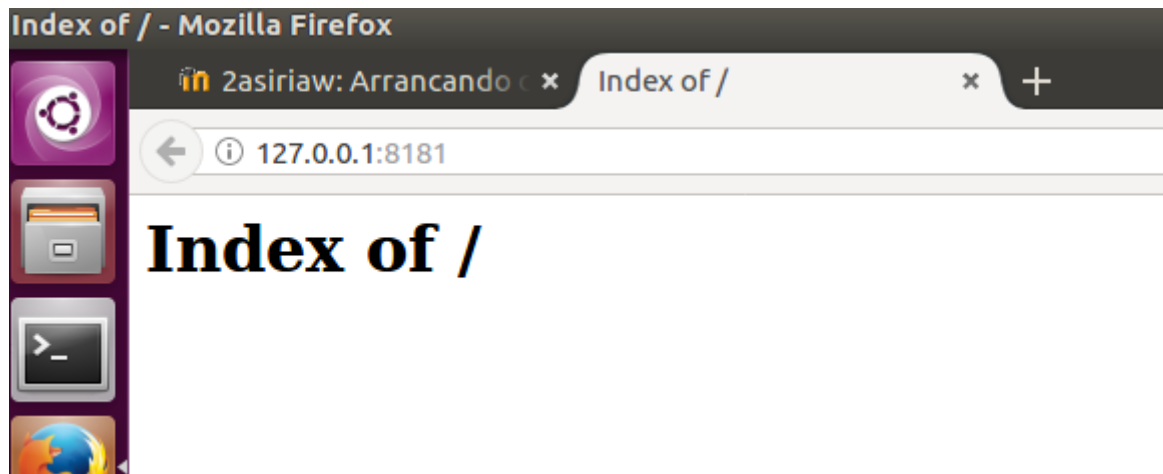
```
root@usuario-VirtualBox:/home/usuario# docker run -it -v /home/usuario:/datos --name d3 debian /bin/bash
root@bc9bb143c55b:/# █
```

Usando una carpeta propia como directorio raíz un servidor web.

Usaremos la siguiente orden

[docker](#) run -d --name a2 -v /home/usuario/data:/usr/local/apache2/htdocs -p 8181:80 httpd

```
root@usuario-VirtualBox:/home/usuario# docker run -d --name a2 -v /home/usuario/data:/usr/local/apache2/htdocs -p 8181:80 httpd
8d7215e3db02087a9cf015ed535ec41bc5610799ca4fb56ba3f02d4d40171c61
root@usuario-VirtualBox:/home/usuario# █
```



Viendo los puertos abiertos y compartidos por cada contenedor

docker port nombre_contenedor (o id)

```
root@usuario-VirtualBox:/home/usuario# docker port a2
80/tcp -> 0.0.0.0:8181
root@usuario-VirtualBox:/home/usuario#
```

Gestión de imágenes y contenedores

Para borrar contenedores

[docker](#) rm nombre (o id)

De manera previa deberemos haber parado el contenedor si era un contenedor ejecutando servicios.

[docker](#) stop nombre (o id)

```
root@usuario-VirtualBox:/home/usuario# docker stop usuario
usuario
root@usuario-VirtualBox:/home/usuario# docker rm usuario
usuario
root@usuario-VirtualBox:/home/usuario# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8d7215e3db02	httpd	"httpd-foreground"	12 minutes ago	Up 12 minutes	0.0.0.0:8181->80/tcp	a2
9a9d07fed3de	mysql	"docker-entrypoint.sh"	25 minutes ago	Up 25 minutes	0.0.0.0:3316->3306/tcp	m1
a6196d15d71c	httpd	"httpd-foreground"	31 minutes ago	Up 31 minutes	0.0.0.0:8080->80/tcp	usuar

```
io2
```

Para borrar imágenes

[docker](#) rmi nombre (o id)

De manera previa deberemos haber borrados todos los conenedores que se hayan creado a partir de esa imagen.

```
root@usuario-VirtualBox:/home/usuario# docker rmi ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:506e2d5852de1d7c90d538c5332bd3cc33b9cbd26f6ca653875899c505c82687
Deleted: sha256:747cb2d60bbebda48aff14a8be5c8b913ca69318a6067e57c697f8a78dda06e
Deleted: sha256:ec1fd849ff0a8f0aa2fd1acc29ad5dabb979b89f63b74a4f54e31a7b0a100aa1
Deleted: sha256:e3f6dffa20cf36460d23bfb22e17be6e5339891f8537f32db79887caf832048b
Deleted: sha256:c213ffdc9f7032702de5a8e9045fcce2353b7221ef6bf4509e02005cfc858f58
Deleted: sha256:3fddf55a451aa43707518f2d8788c12ee5eb1f1e3075433f5bcf4d445d5c275d
Deleted: sha256:0f5ff0cf6a1c53f94b15f03536c490040f233bc455f1232f54cc8eb344a3a368
root@usuario-VirtualBox:/home/usuario# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	latest	5fac85ee2c68	16 hours ago	408.2 MB
httpd	latest	c24f66af34b4	4 days ago	177.3 MB
wordpress	latest	d3f0cddf9493	7 days ago	408.4 MB
php	latest	c342f917459a	7 days ago	370.7 MB
debian	latest	874e27b628fd	7 days ago	100.1 MB
ubuntu	14.04	dea1945146b9	4 weeks ago	188 MB

```
root@usuario-VirtualBox:/home/usuario# █
```

Creando una nueva imagen a partir de un contenedor

Los contenedores normalmente vienen con lo mínimo pero nosotros poderemos arrancarlos instalar lo que necesitemos y posteriormente crear nuevas imágenes que tenga ya todo lo que hemos instalado nuevo y que podemos compartir.

Creemos un contenedor debian:

[docker](#) run -it --name d1 debian /bin/bash

```
root@usuario-VirtualBox:/home/usuario# docker run -it --name d1 debian /bin/bash
root@ff5003081943:/# █
```

En ese contenedor vamos a ejecutar la siguiente órdenes para instalar ciertos programas para la gestión de red que por defecto no vienen instalador en la imagen de debian:latest

apt-get update

```
root@ff5003081943:/# apt-get update
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://security.debian.org stretch/updates InRelease [62.9 kB]
Get:3 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://deb.debian.org/debian stretch Release [118 kB]
Get:5 http://deb.debian.org/debian stretch Release.gpg [2479 B]
Get:6 http://deb.debian.org/debian stretch-updates/main amd64 Packages [5841 B]
Get:7 http://security.debian.org stretch/updates/main amd64 Packages [222 kB]
Get:8 http://deb.debian.org/debian stretch/main amd64 Packages [9500 kB]
Fetched 10.0 MB in 2s (3848 kB/s)
Reading package lists... Done
root@ff5003081943:/#
```

apt-get install nano

```
root@ff5003081943:/# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 485 kB of archives.
After this operation, 2092 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 nano amd64 2.7.4-1 [485 kB]
Fetched 485 kB in 0s (1717 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 6490 files and directories currently installed.)
Preparing to unpack .../nano_2.7.4-1_amd64.deb ...
Unpacking nano (2.7.4-1) ...
Setting up nano (2.7.4-1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
root@ff5003081943:/#
```

apt-get install net-tools

```
root@ff5003081943:/# apt-get install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 248 kB of archives.
After this operation, 963 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1 [248 kB]
Fetched 248 kB in 0s (1398 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package net-tools.
(Reading database ... 6590 files and directories currently installed.)
Preparing to unpack ../net-tools_1.60+git20161116.90da8a0-1_amd64.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1) ...
root@ff5003081943:/#
```

apt-get install dnsutils

```
root@ff5003081943:/# apt-get install dnsutils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bind9-host geoip-database krb5-locales libbind9-140 libcap2 libdns162 libgeoip1 libgssapi-krb5-2 libicu57 libisc160 libisccc140
  libiscfg140 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblwres141 libssl1.0.2 libxml2 sgml-base xml-core
Suggested packages:
  rblcheck geoip-bin krb5-doc krb5-user sgml-base-doc debhelper
The following NEW packages will be installed:
  bind9-host dnsutils geoip-database krb5-locales libbind9-140 libcap2 libdns162 libgeoip1 libgssapi-krb5-2 libicu57 libisc160
  libisccc140 libiscfg140 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblwres141 libssl1.0.2 libxml2 sgml-base xml-core
0 upgraded, 22 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.8 MB of archives.
After this operation, 53.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian stretch/main amd64 sgml-base all 1.29 [14.8 kB]
Get:2 http://deb.debian.org/debian stretch/main amd64 libssl1.0.2 amd64 1.0.2l-2 [1294 kB]
Get:3 http://deb.debian.org/debian stretch/main amd64 libcap2 amd64 1:2.25-1 [16.8 kB]
Get:4 http://deb.debian.org/debian stretch/main amd64 libgeoip1 amd64 1.6.9-4 [90.5 kB]
Get:5 http://deb.debian.org/debian stretch/main amd64 libkeyutils1 amd64 1.5.9-9 [12.4 kB]
Get:6 http://deb.debian.org/debian stretch/main amd64 libkrb5support0 amd64 1.15-1+deb9u1 [61.9 kB]
Get:7 http://deb.debian.org/debian stretch/main amd64 libk5crypto3 amd64 1.15-1+deb9u1 [119 kB]
Get:8 http://deb.debian.org/debian stretch/main amd64 libkrb5-3 amd64 1.15-1+deb9u1 [311 kB]
Get:9 http://deb.debian.org/debian stretch/main amd64 libgssapi-krb5-2 amd64 1.15-1+deb9u1 [155 kB]
Get:10 http://deb.debian.org/debian stretch/main amd64 libicu57 amd64 57.1-6 [7701 kB]
Get:11 http://deb.debian.org/debian stretch/main amd64 libxml2 amd64 2.9.4+dfsg1-2.2+deb9u1 [920 kB]
Get:12 http://deb.debian.org/debian stretch/main amd64 libisc160 amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [397 kB]
Get:13 http://deb.debian.org/debian stretch/main amd64 libdns162 amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [1077 kB]
Get:14 http://deb.debian.org/debian stretch/main amd64 libisccc140 amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [198 kB]
Get:15 http://deb.debian.org/debian stretch/main amd64 libiscfg140 amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [222 kB]
Get:16 http://deb.debian.org/debian stretch/main amd64 libbind9-140 amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [206 kB]
Get:17 http://deb.debian.org/debian stretch/main amd64 liblwres141 amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [214 kB]
Get:18 http://deb.debian.org/debian stretch/main amd64 bind9-host amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [231 kB]
Get:19 http://deb.debian.org/debian stretch/main amd64 krb5-locales all 1.15-1+deb9u1 [93.8 kB]
Get:20 http://deb.debian.org/debian stretch/main amd64 dnsutils amd64 1:9.10.3.dfsg.P4-12.3+deb9u3 [283 kB]
Get:21 http://deb.debian.org/debian stretch/main amd64 geoip-database all 20170512-1 [2112 kB]
Get:22 http://deb.debian.org/debian stretch/main amd64 xml-core all 0.17 [23.2 kB]
Fetched 15.8 MB in 1s (10.3 MB/s)
```


apt-get install nmap

```
root@ff5003081943:/# apt-get install nmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bzip2 file libblas-common libblas3 libbexpat1 libffi6 libgfortran3 liblinear3 liblua5.3-0 libmagic-mgc libmagic1 libpcap0.8
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libquadmath0 libreadline7 libsqlite3-0 libssl1.1 libxslt1.1 mime-support
  ndiff python python-bs4 python-chardet python-html5lib python-lxml python-minimal python-pkg-resources python-six
  python-webencodings python2.7 python2.7-minimal readline-common xz-utils
Suggested packages:
  bzip2-doc liblinear-tools liblinear-dev python-doc python-tk python-genshi python-lxml-dbg python-lxml-doc python-setuptools
  python2.7-doc binutils binfmt-support readline-doc
The following NEW packages will be installed:
  bzip2 file libblas-common libblas3 libbexpat1 libffi6 libgfortran3 liblinear3 liblua5.3-0 libmagic-mgc libmagic1 libpcap0.8
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libquadmath0 libreadline7 libsqlite3-0 libssl1.1 libxslt1.1 mime-support
  ndiff nmap python python-bs4 python-chardet python-html5lib python-lxml python-minimal python-pkg-resources python-six
  python-webencodings python2.7 python2.7-minimal readline-common xz-utils
0 upgraded, 36 newly installed, 0 to remove and 0 not upgraded.
Need to get 15.1 MB of archives.
After this operation, 60.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian stretch/main amd64 libpython2.7-minimal amd64 2.7.13-2 [389 kB]
Get:2 http://deb.debian.org/debian stretch/main amd64 python2.7-minimal amd64 2.7.13-2 [1383 kB]
Get:3 http://deb.debian.org/debian stretch/main amd64 python-minimal amd64 2.7.13-2 [40.5 kB]
Get:4 http://deb.debian.org/debian stretch/main amd64 mime-support all 3.60 [36.7 kB]
Get:5 http://deb.debian.org/debian stretch/main amd64 libbexpat1 amd64 2.2.0-2+deb9u1 [83.4 kB]
Get:6 http://deb.debian.org/debian stretch/main amd64 libffi6 amd64 3.2.1-6 [20.4 kB]
Get:7 http://deb.debian.org/debian stretch/main amd64 readline-common all 7.0-3 [70.4 kB]
Get:8 http://deb.debian.org/debian stretch/main amd64 libreadline7 amd64 7.0-3 [151 kB]
Get:9 http://deb.debian.org/debian stretch/main amd64 libsqlite3-0 amd64 3.16.2-5 [572 kB]
Get:10 http://deb.debian.org/debian stretch/main amd64 libssl1.1 amd64 1.1.0f-3 [1342 kB]
Get:11 http://deb.debian.org/debian stretch/main amd64 libpython2.7-stdlib amd64 2.7.13-2 [1895 kB]
Get:12 http://deb.debian.org/debian stretch/main amd64 python2.7 amd64 2.7.13-2 [285 kB]
Get:13 http://deb.debian.org/debian stretch/main amd64 libpython-stdlib amd64 2.7.13-2 [20.0 kB]
Get:14 http://deb.debian.org/debian stretch/main amd64 python amd64 2.7.13-2 [154 kB]
Get:15 http://deb.debian.org/debian stretch/main amd64 bzip2 amd64 1.0.6-8.1 [47.5 kB]
Get:16 http://deb.debian.org/debian stretch/main amd64 libmagic-mgc amd64 1:5.30-1+deb9u1 [222 kB]
Get:17 http://deb.debian.org/debian stretch/main amd64 libmagic1 amd64 1:5.30-1+deb9u1 [111 kB]
Get:18 http://deb.debian.org/debian stretch/main amd64 file amd64 1:5.30-1+deb9u1 [64.1 kB]
Get:19 http://deb.debian.org/debian stretch/main amd64 xz-utils amd64 5.2.2-1.2+b1 [266 kB]
```

Salgo del contenedor y desde la línea de comandos del host ejecuto esta orden para crear mi nueva imagen

[docker](#) commit d1 debian_net

Y al hacer [docker](#) images tendré esa nueva imagen en nuestra lista de imágenes.

```
root@ff5003081943:/# exit
exit
root@usuario-VirtualBox:/home/usuario# docker commit d1 debian_net
sha256:77b198015ab34b46f8ea37d7638d8b741901a0a422f8e0e2a2d671426f3b465b
root@usuario-VirtualBox:/home/usuario# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
debian_net          latest             77b198015ab3       15 seconds ago     243 MB
mysql               latest            5fac85ee2c68       16 hours ago       408.2 MB
httpd               latest            c24f66af34b4       4 days ago         177.3 MB
wordpress           latest            d3f0cddf9493       7 days ago         408.4 MB
php                 latest            c342f917459a       7 days ago         370.7 MB
debian              latest            874e27b628fd       7 days ago         100.1 MB
ubuntu              14.04             dea1945146b9       4 weeks ago        188 MB
root@usuario-VirtualBox:/home/usuario#
```

Para compartirla puedo hacer dos cosas:

a) Guardarla con [docker](#) save , compartir el fichero y cargarla con [docker](#) load

[docker](#) save --output debian_net.tar debian_net

```
root@usuario-VirtualBox:/home/usuario# docker save --output debian_net.tar debian_net
root@usuario-VirtualBox:/home/usuario#
```

Y podemos pasarla de cualquier forma para distribuirla. Tened en cuenta que es un fichero de bastante tamaño. Una vez el destinatario recibe el fichero puede cargar o recuperar dicha imagen mediante la siguiente orden:

[docker](#) load --input debian_net.tar

```
root@usuario-VirtualBox:/home/usuario# docker load --input debian_net.tar
Loaded image: debian_net:latest
root@usuario-VirtualBox:/home/usuario#
```

Para probar juntos en pareja y crear dos imágenes distintas y compartirlas con el otro.

b) Subirla a [DockerHub](#)

Para esto tenemos que seguir los siguientes pasos:

1. Entrar en sesión: **[docker](#) login** (para asegurarnos de que podemos conectarnos al repositorio de [DockerHub](#))
2. Etiquetar la imagen: **[docker](#) tag nombre_imagen (o id) usuario_[docker](#)_hub/nombre_imagen**
3. Hacer push a [DockerHub](#): **[docker](#) push nombre_usuario/nombre_imagen**

```
root@usuario-VirtualBox:/home/usuario# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username (eduardogomez): eduardogomez
Password:
Login Succeeded
root@usuario-VirtualBox:/home/usuario# docker tag debian_net eduardogomez/debian_net
root@usuario-VirtualBox:/home/usuario# docker push eduardogomez/debian_net
The push refers to a repository [docker.io/eduardogomez/debian_net]
486d5ee7f1c7: Pushed
a75caa09eb1f: Mounted from library/debian
latest: digest: sha256:3c263194e14fe7db1810b409275e6f3a87f68c8524813cf7d4c5dd37cab986ce size: 741
root@usuario-VirtualBox:/home/usuario#
```

Y podremos comprobar que esa imagen ya está pública a través de nuestra cuenta de [DockerHub](#).

The screenshot shows a web browser window with two tabs: 'eduardogomez/debian' and 'ies triana moodle - Bu...'. The address bar shows the URL 'https://hub.docker.com/r/eduardogomez/debian_net/'. The DockerHub interface is displayed, featuring a dark blue header with a search bar, navigation links (Dashboard, Explore, Organizations, Create), and a user profile for 'eduardogomez'. The main content area is titled 'PUBLIC REPOSITORY' and shows the repository 'eduardogomez/debian_net' with a star icon. Below the repository name, it says 'Last pushed: a few seconds ago'. A horizontal menu contains 'Repo Info' (selected), 'Tags', 'Collaborators', 'Webhooks', and 'Settings'. The repository details are organized into two columns. The left column has 'Short Description' and 'Full Description' sections, both with edit icons and placeholder text: 'Short description is empty for this repo.' and 'Full description is empty for this repo.' respectively. The right column has a 'Docker Pull Command' section with a copy icon and the command 'docker pull eduardogomez/debian_', and an 'Owner' section showing a user icon and the name 'eduardogomez'.

eduardogomez/debian_net ☆

Last pushed: a few seconds ago

Repo Info Tags Collaborators Webhooks Settings

Short Description

Short description is empty for this repo.

Full Description

Full description is empty for this repo.

Docker Pull Command

```
docker pull eduardogomez/debian_
```

Owner

eduardogomez