

Introducción a Docker.

Agustín Gallego



PERCONA

Agenda.

- Contenedores VS Máquinas Virtuales
- Qué es Docker
- Arquitectura
- Notas varias
- Ejemplos de algunos comandos

Contenedores VS Máquinas Virtuales

- Máquinas Virtuales

A **hypervisor** or **virtual machine monitor (VMM)** is a piece of computer software, firmware or hardware that creates and runs [virtual machines](#). A computer on which a hypervisor runs one or more virtual machines is called a *host machine*, and each virtual machine is called a *guest machine*. The hypervisor presents the guest operating systems with a [virtual operating platform](#) and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources: for example, [Linux](#), [Windows](#), and [OS X](#) instances can all run on a single physical [x86](#) machine. This contrasts with [operating-system-level virtualization](#), where all instances must share a single kernel, though the guest operating systems can differ in [user space](#), such as different [Linux distributions](#) with the same kernel.

- Contenedores

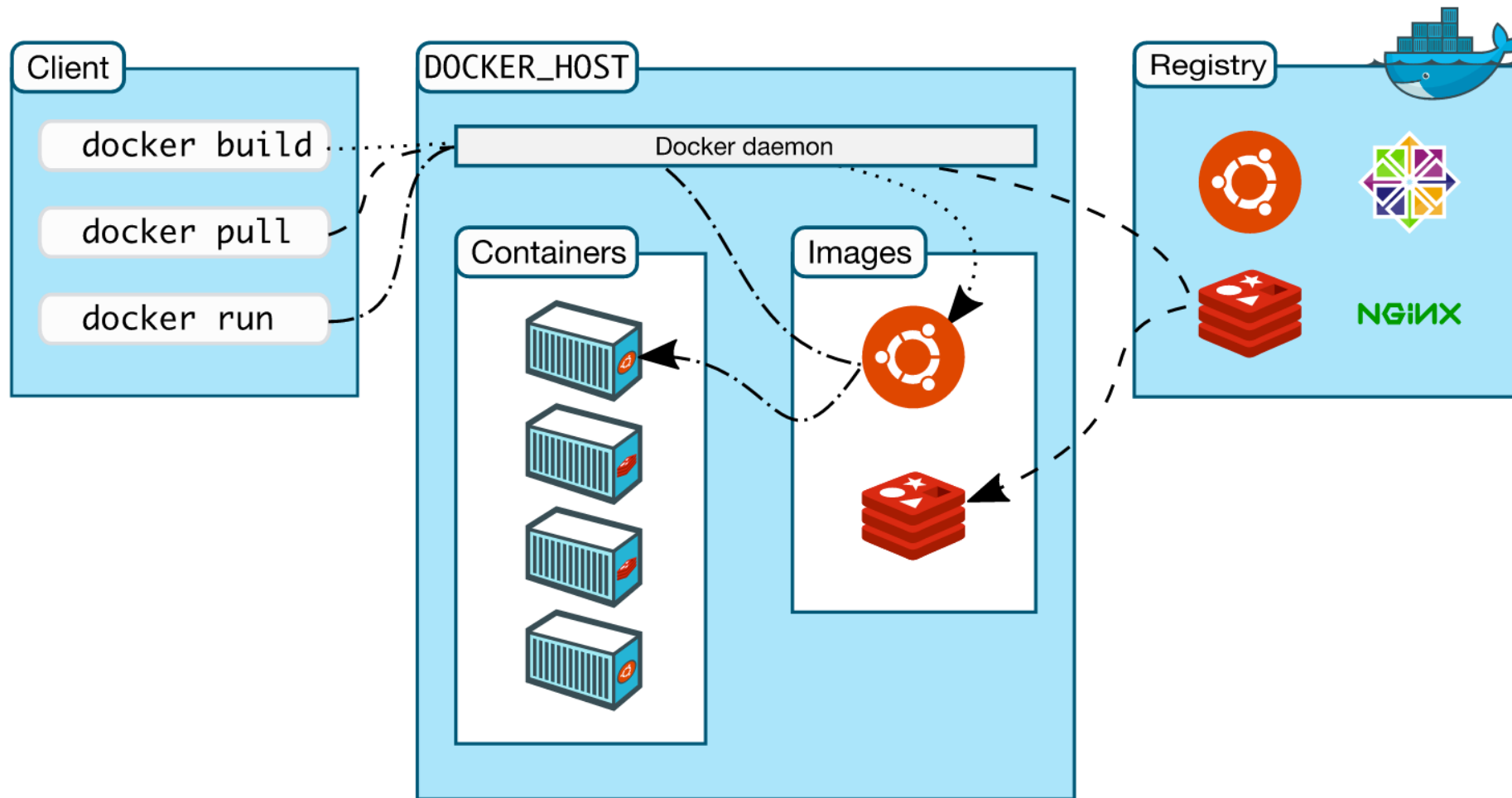
Operating-system-level virtualization is a server [virtualization](#) method in which the [kernel](#) of an [operating system](#) allows the existence of multiple isolated [user-space instances](#), instead of just one. Such instances, which are sometimes called **containers**, **software containers**,^[1] [virtualization engines](#) (VEs) or jails ([FreeBSD jail](#) or [chroot jail](#)), may look and feel like a real server from the point of view of its owners and users.

On [Unix-like](#) operating systems, this technology can be seen as an advanced implementation of the standard [chroot](#) mechanism. In addition to isolation mechanisms, the kernel often provides [resource-management](#) features to limit the impact of one container's activities on other containers.

Qué es Docker

- Engine
 - Implementa contenedores (distinto de LXC) usando: Namespaces + CGroups + UnionFS
 - Aisla aplicaciones en ejecución
- Cloud - <https://cloud.docker.com/>
- Hub - <https://hub.docker.com/>
- Docker Trusted Registry / Universal Control Plane
- Compose
- Swarm Mode (nuevo en v1.12, gestion nativa de clusters)

Arquitectura



Notas varias

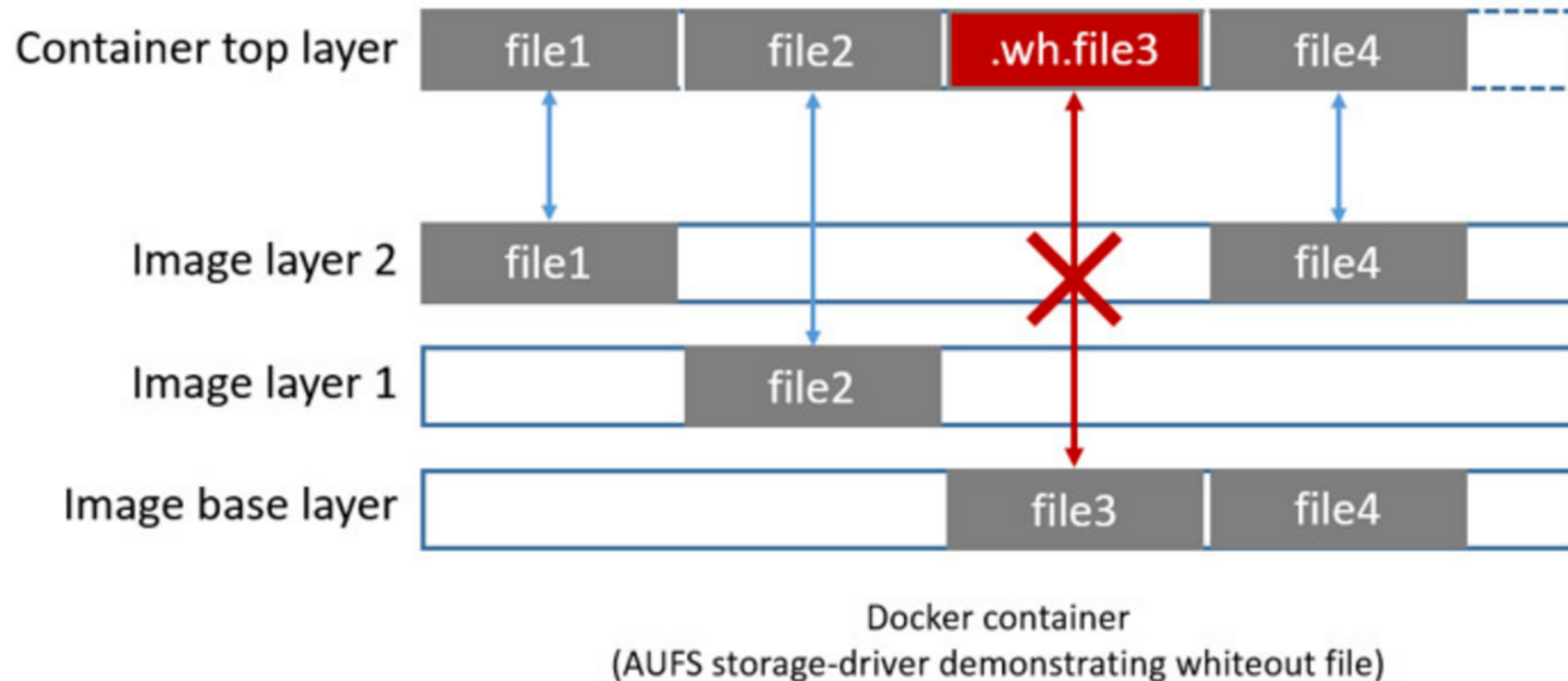
- Los contenedores se organizan en capas
 - Las capas pueden ser reutilizadas por nuevas versiones de la imagen
 - Se puede hacer diff de contenedores y ver que cambios sufrieron.
- Comandos ejecutados por root (o usuarios del grupo docker)
 - Se supone que cambiará en releases futuros
 - <https://docs.docker.com/engine/security/security/#/docker-security>
 - Inconveniente en servidores multi usuario.

Notas varias

- Los contenedores deben tener nombres
 - Si no se indica uno con `--name`, docker elige uno.
- No pueden tener una dirección IP fija
 - A menos que se cree una red propia
- Las capas de imágenes son solo de lectura
- La capa de escritura es efímera
 - A menos que se cree un volumen (mapeado a un directorio en `/var/lib/docker/volumes/`)

Notas varias - Capas de imágenes

- Las capas de imágenes son solo de lectura
- La capa de escritura es efímera (no persistente)



Laboratorio. Curso rápido de docker.

```
# docker --version
```

```
Docker version 1.10.3, build f476348/1.10.3
```



PERCONA

\$ docker search <image-name>;

```
[root@ae-cloud-1 ~]# docker search centos
```

INDEX	NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
docker.io	docker.io/centos	The official build of CentOS.	2544	[OK]	
docker.io	docker.io/ansible/centos7-ansible	Ansible on Centos7	82		[OK]
docker.io	docker.io/jdeathe/centos-ssh	CentOS-6 6.8 x86_64 / CentOS-7 7.2.1511 x8...	27		[OK]
docker.io	docker.io/nimmis/java-centos	This is docker images of CentOS 7 with dif...	14		[OK]
docker.io	docker.io/million12/centos-supervisor	Base CentOS-7 with supervisord launcher, h...	12		[OK]
docker.io	docker.io/gluster/gluster-centos	Official GlusterFS Image [CentOS7 + Glus...	11		[OK]
docker.io	docker.io/torusware/speedus-centos	Always updated official CentOS docker imag...	8		[OK]
docker.io	docker.io/nickistre/centos-lamp	LAMP on centos setup	4		[OK]
docker.io	docker.io/centos/mariadb55-centos7		3		[OK]
docker.io	docker.io/nathonfowlie/centos-jre	Latest CentOS image with the JRE pre-insta...	3		[OK]
docker.io	docker.io/blacklabelops/centos	CentOS Base Image! Built and Updates Daily!	1		[OK]
docker.io	docker.io/darksheer/centos	Base Centos Image -- Updated hourly	1		[OK]
docker.io	docker.io/harisekhon/centos-java	Java on CentOS (OpenJDK, tags jre/jdk7-8)	1		[OK]
docker.io	docker.io/pacur/centos-6	Pacur CentOS 6	1		[OK]
docker.io	docker.io/pacur/centos-7	Pacur CentOS 7	1		[OK]
docker.io	docker.io/timhughes/centos	Centos with systemd installed and running	1		[OK]
docker.io	docker.io/dmglab/centos	CentOS with some extras - This is for the ...	0		[OK]
docker.io	docker.io/grayzone/centos	auto build for centos.	0		[OK]
docker.io	docker.io/grossws/centos	CentOS 6 and 7 base images with gosu and l...	0		[OK]
docker.io	docker.io/harisekhon/centos-scala	Scala + CentOS (OpenJDK tags 2.10-jre7 - 2...	0		[OK]
docker.io	docker.io/januswel/centos	yum update-ed CentOS image	0		[OK]
docker.io	docker.io/kz8s/centos	Official CentOS plus epel-release	0		[OK]
docker.io	docker.io/repositoryjp/centos	Docker Image for CentOS.	0		[OK]
docker.io	docker.io/smartentry/centos	CentOS with smartentry	0		[OK]
docker.io	docker.io/ustclug/centos	USTC centos	0		[OK]

\$ docker pull <image-name>[:<tag>];

```
[root@ae-cloud-1 ~]# docker pull centos:6
Trying to pull repository docker.io/library/centos ...
6: Pulling from docker.io/library/centos
08a7a0bb6122: Pull complete
Digest: sha256:cd6d68000b47a91e7c94b558d7e3e653c3f0eac1a77842d97b0b7ad955cad608
Status: Downloaded newer image for docker.io/centos:6
[root@ae-cloud-1 ~]# docker pull centos
Using default tag: latest
Trying to pull repository docker.io/library/centos ...
latest: Pulling from docker.io/library/centos
3d8673bd162a: Pull complete
Digest: sha256:a66ffcb73930584413de83311ca11a4cb4938c9b2521d331026dad970c19adf4
Status: Downloaded newer image for docker.io/centos:latest
```

- Notar que si no se usa un tag, entonces es 'latest'.
- No hay forma de verificar tags en la CLI, debemos usar Docker hub. En este caso:
 - https://hub.docker.com/_/centos/

\$ docker images [<image-name>];

```
[root@ae-cloud-1 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/fedora	latest	11a5107645d4	30 hours ago	204.4 MB
docker.io/ubuntu	trusty	ff6011336327	5 days ago	187.9 MB
docker.io/centos	6	a3c09d36ab4a	2 weeks ago	194.6 MB
docker.io/centos	latest	970633036444	2 weeks ago	196.7 MB
docker.io/percona/pmm-server	1.0.2	9dfdc021fe53	2 weeks ago	673.5 MB
docker.io/ubuntu	precise	60df6678b255	3 weeks ago	139.3 MB
docker.io/guriandoro/cometcloud	v1	636bea1c791f	7 weeks ago	804.9 MB
docker.io/fedora	22	2d3da2084d08	9 weeks ago	188.7 MB
docker.io/percona/pmm-server	latest	691167a2cff5	9 weeks ago	672.5 MB

```
[root@ae-cloud-1 ~]# docker images centos
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/centos	6	a3c09d36ab4a	2 weeks ago	194.6 MB
docker.io/centos	latest	970633036444	2 weeks ago	196.7 MB

\$ docker create <image-name>[:<tag>];

```
[root@ae-cloud-1 ~]# docker create -ti centos /bin/bash
66d2e331019e4455c5954cdd816f8706f1437784f2ad266dd4ef29050d6a1819
[root@ae-cloud-1 ~]# docker create -ti --name=centos-tag-test centos:6 /bin/bash
8001e88e8f6f676e31c701c3c9271f14e028da0efc11824e7d12102004928153
[root@ae-cloud-1 ~]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8001e88e8f6f	centos:6	"/bin/bash"	4 seconds ago	Created		centos-tag-test
66d2e331019e	centos	"/bin/bash"	42 seconds ago	Created		awesome_poitras

- Útil para crear contenedores con antelación
- Útil para crear contenedores que persistan datos
- Crea el contenedor, pero no ejecuta el comando

\$ docker ps [-a];

```
[root@ae-cloud-1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
47992308b4d7        percona/pmm-server:1.0.2  "supervisord -c /etc/"  2 weeks ago        Up About an hour    0.0.0.0:80->80/tcp  pmm-server
[root@ae-cloud-1 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
8001e88e8f6f        centos:6            "/bin/bash"         7 minutes ago       Created                                 centos-tag-test
66d2e331019e        centos              "/bin/bash"         7 minutes ago       Created                                 awesome_poitras
47992308b4d7        percona/pmm-server:1.0.2  "supervisord -c /etc/"  2 weeks ago        Up About an hour    0.0.0.0:80->80/tcp  pmm-server
ddd7b886b8d5        percona/pmm-server:1.0.2  "/bin/true"         2 weeks ago        Created                                 pmm-data
[root@ae-cloud-1 ~]# docker ps --help

Usage:  docker ps [OPTIONS]

List containers

    -a, --all           Show all containers (default shows just running)
    -f, --filter=[]     Filter output based on conditions provided
    --format            Pretty-print containers using a Go template
    --help              Print usage
    -l, --latest        Show the latest created container (includes all states)
    -n=-1              Show n last created containers (includes all states)
    --no-trunc         Don't truncate output
    -q, --quiet         Only display numeric IDs
    -s, --size          Display total file sizes
```

- -a para listar todos los contenedores (incluidos los detenidos)

\$ docker start [-a -i] <container-ID-or-name>;

```
[root@ae-cloud-1 ~]# docker start -ai awesome_poitras
[root@66d2e331019e /]# cat /etc/redhat-release
CentOS Linux release 7.2.1511 (Core)
[root@66d2e331019e /]# #NOW I WILL DETACH FROM THE CONTAINER WITH CTRL-P-Q.
[root@66d2e331019e /]# #IF YOU EXIT WITHOUT DETACHING, THE MAIN PROCESS
[root@66d2e331019e /]# #WILL FINISH, AND THE CONTAINER WILL STOP.
[root@66d2e331019e /]# [root@ae-cloud-1 ~]#
[root@ae-cloud-1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
66d2e331019e	centos	"/bin/bash"	18 minutes ago	Up 57 seconds		awesome_poitras
47992308b4d7	percona/pmm-server:1.0.2	"supervisord -c /etc/"	2 weeks ago	Up About an hour	0.0.0.0:80->80/tcp	pmm-server

```
[root@ae-cloud-1 ~]# docker start -ai awesome_poitras
#IT IS KIND OF A MISTERY WHY start WORKS WHEN THE CONTAINER IS ALREADY STARTED. BUT I'LL LEAVE THAT FOR NEXT TIME. :)
[root@66d2e331019e /]# exit
exit
[root@ae-cloud-1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
47992308b4d7	percona/pmm-server:1.0.2	"supervisord -c /etc/"	2 weeks ago	Up About an hour	0.0.0.0:80->80/tcp	pmm-server

- De aquí en adelante usaremos <container-name>, pero siempre se puede usar su ID

\$ docker top <container-name> [aux];

```
[root@ae-cloud-1 ~]# docker top awesome_poitras ax
PID          TTY          STAT         TIME         COMMAND
5084          pts/1        Ss           0:00         /bin/bash
5355          pts/1        S+           0:00         /usr/bin/python /usr/bin/yum update
[root@ae-cloud-1 ~]# docker top awesome_poitras ax
PID          TTY          STAT         TIME         COMMAND
5084          pts/1        Ss           0:00         /bin/bash
5364          pts/1        R+           0:01         yes
[root@ae-cloud-1 ~]# docker top awesome_poitras ax
PID          TTY          STAT         TIME         COMMAND
5084          pts/1        Ss           0:00         /bin/bash
5372          pts/1        S+           0:00         vi /test
```

- aux - argumentos que pasaríamos a `ps`

\$ docker logs [-f -t --tail=N] <container-name>;

```
[root@ae-cloud-1 ~]# docker logs --tail=15 pmm-server
2016-08-17 03:41:20,987 INFO supervisord started with pid 1
2016-08-17 03:41:21,991 INFO spawned: 'mysql' with pid 9
2016-08-17 03:41:22,023 INFO spawned: 'consul' with pid 10
2016-08-17 03:41:22,029 INFO spawned: 'grafana' with pid 11
2016-08-17 03:41:22,058 INFO spawned: 'qan-api' with pid 15
2016-08-17 03:41:22,064 INFO spawned: 'nginx' with pid 16
2016-08-17 03:41:22,067 INFO spawned: 'cron' with pid 17
2016-08-17 03:41:22,070 INFO spawned: 'prometheus' with pid 18
2016-08-17 03:41:23,072 INFO success: mysql entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2016-08-17 03:41:23,072 INFO success: consul entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2016-08-17 03:41:23,072 INFO success: grafana entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2016-08-17 03:41:23,072 INFO success: qan-api entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2016-08-17 03:41:23,073 INFO success: nginx entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2016-08-17 03:41:23,073 INFO success: cron entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2016-08-17 03:41:23,073 INFO success: prometheus entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
```

- -f para seguir (“follow”) (como `tail -f`)
- -t para agregar timestamps
- --tail=N para mostrar las últimas 'N' líneas

\$ docker stop [--time=N] <container-name>;

```
[root@ae-cloud-1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1be895a2cfc4	centos	"/bin/bash"	About an hour ago	Up About an hour		awesome_poitras
47992308b4d7	percona/pmm-server:1.0.2	"supervisord -c /etc/"	2 weeks ago	Up 3 hours	0.0.0.0:80->80/tcp	pmm-server

```
[root@ae-cloud-1 ~]# docker stop awesome_poitras
awesome_poitras
[root@ae-cloud-1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
47992308b4d7	percona/pmm-server:1.0.2	"supervisord -c /etc/"	2 weeks ago	Up 3 hours	0.0.0.0:80->80/tcp	pmm-server

- --time=N to esperar 'N' segundos antes de forzar el detenimiento, de ser necesario (por omisión son 10 segundos)

\$ docker attach <container-name>;

```
[root@ae-cloud-1 ~]# docker attach pmm-server
#NO TTY
^P
#CTRL-Q WAS NOT RECOGNIZED. THE ONLY WAY OUT IS TO CLOSE THE TAB.
#CTRL-C WILL BE ROUTED TO THE PROCESS RUNNING, AND WILL MOST LIKELY KILL IT.
```

- Usar con cuidado, podemos quedar bloqueados.
- Para evitarlo, usar `--sig-proxy=false`, mejor aún, usar `docker exec`

```
[root@ae-cloud-1 ~]# docker attach --sig-proxy=false pmm-server
^C[root@ae-cloud-1 ~]#
```

\$ docker exec [-it -d] <container-name> <command>;

```
[root@ae-cloud-1 ~]# docker top awesome_poitras x
PID          TTY          STAT         TIME         COMMAND
15267         pts/2        Ss+          0:00         /bin/bash
15485         pts/3        Ss+          0:00         /bin/bash
[root@ae-cloud-1 ~]#
[root@ae-cloud-1 ~]# docker top awesome_poitras x
PID          TTY          STAT         TIME         COMMAND
15267         pts/2        Ss+          0:00         /bin/bash
15485         pts/3        Ss+          0:00         /bin/bash
15645         pts/6        Ss+          0:00         /bin/bash
[root@ae-cloud-1 ~]#
[root@ae-cloud-1 ~]# docker top awesome_poitras x
PID          TTY          STAT         TIME         COMMAND
15267         pts/2        Ss+          0:00         /bin/bash
15485         pts/3        Ss+          0:00         /bin/bash
15682         pts/6        Ss+          0:00         sleep 10
[root@ae-cloud-1 ~]#
```

```
[root@ae-cloud-1 ~]#
[root@ae-cloud-1 ~]#
[root@ae-cloud-1 ~]#
[root@ae-cloud-1 ~]# docker exec -it awesome_poitras /bin/bash
[root@1be895a2cfc4 /]#
[root@1be895a2cfc4 /]#
[root@1be895a2cfc4 /]#
[root@1be895a2cfc4 /]#
[root@1be895a2cfc4 /]# exit
exit
[root@ae-cloud-1 ~]# docker exec -it awesome_poitras sleep 10
[root@ae-cloud-1 ~]#
```

- Podríamos haber dejado los argumentos -it fuera del segundo comando docker exec

```
[root@ae-cloud-1 ~]# docker exec awesome_poitras whoami
root
[root@ae-cloud-1 ~]# docker exec -d awesome_poitras whoami
[root@ae-cloud-1 ~]#
```

\$ docker run [OPTS] <image-name> <command>;

```
[root@ae-cloud-1 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
[root@ae-cloud-1 ~]# docker run -d --name="detached-sleep" centos sleep 9999999						
9c35eadddc786679c01fc44453d98858708f2065afcdd51a8167ba8c26564635						
[root@ae-cloud-1 ~]# docker run -d --name="detached-bash" centos bash						
3fdd2cd2baa0713058e11a850a44123f44a3bc0756a1a42e209c9ab22d86a317						
[root@ae-cloud-1 ~]# docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9c35eadddc78	centos	"sleep 9999999"	10 seconds ago	Up 7 seconds		detached-sleep
[root@ae-cloud-1 ~]# docker ps -a						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3fdd2cd2baa0	centos	"bash"	12 seconds ago	Exited (0) 9 seconds ago		detached-bash
9c35eadddc78	centos	"sleep 9999999"	17 seconds ago	Up 14 seconds		detached-sleep

- -d ejecuta el contenedor en el background ('detached')
- Cuando el proceso termina, el contenedor es detenido.

\$ docker run [OPTS] <image-name> <command>;

```
[root@ae-cloud-1 ~]# docker run -it -p 6789:9876 --name="interactive-shell" centos /bin/bash
[root@4fa76bf5cc5b /]# yum install -y nc > /dev/null 2>&1
[root@4fa76bf5cc5b /]# nc -l 9876 #nc IS NOW LISTENING ON PORT 9876. I WILL DETACH WITH CTRL-P-Q
[root@ae-cloud-1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
4fa76bf5cc5b        centos              "/bin/bash"        About a minute ago  Up About a minute  0.0.0.0:6789->9876/tcp interactive-shell
[root@ae-cloud-1 ~]# echo "Testing port redirection between host and container." | nc localhost 6789
[root@ae-cloud-1 ~]# docker logs interactive-shell
[root@4fa76bf5cc5b /]# yum install -y nc > /dev/null 2>&1
[root@4fa76bf5cc5b /]# nc -l 9876 #nc IS NOW LISTENING ON PORT 9876. I WILL DETACH WITH CTRL-P-Q
Testing port redirection between host and container.
```

- -it ejecuta el comando interactivamente
- -p redirecciona tráfico del puerto 6789 del huésped al 9876 del contenedor.
- Notar que el contenedor continuará ejecutando, a pesar de que nc finalizó.

\$ docker run [OPTS] <image-name> <command>;

```
[root@ae-cloud-1 ~]# docker run --rm --name="volumes-test" centos ls -l /opt/ /var/lib/mysql
ls: cannot access /var/lib/mysql: No such file or directory
/opt/:
total 0
[root@ae-cloud-1 ~]# docker run --rm --volumes-from pmm-data --name="volumes-test" centos ls -l /opt/ /var/lib/mysql
/opt/:
total 0
drwxr-xr-x 4 root root 58 Jul 31 19:07 consul-data
drwxr-xr-x 3 root root 18 Aug 18 05:07 prometheus

/var/lib/mysql:
total 45064
-rw-r--r-- 1 root root          0 Jul 28 16:37 debian-5.5.flag
-rw-rw---- 1 102 105 5242880 Aug 18 03:42 ib_logfile0
-rw-rw---- 1 102 105 5242880 Aug 18 03:42 ib_logfile1
-rw-rw---- 1 102 105 35651584 Aug 18 03:42 ibdata1
drwx----- 2 102 root    4096 Jul 28 16:39 mysql
drwx----- 2 102 105     4096 Jul 28 16:39 performance_schema
drwx----- 2 102 105      201 Jul 28 16:39 pmm
```

- Agregamos --rm para el que contenedor sea eliminado luego de detenerse.
- pmm-data tiene los siguientes directorios montados: /opt/prometheus/data /opt/consul-data /var/lib/mysql

\$ docker run [OPTS] <image-name> <command>;

```
[root@ae-cloud-1 ~]# docker run --rm --volume=/var/lib/mysql:/var/lib/mysql2/ --name="volumes-test" centos ls -l /var/lib/mysql2
total 293136
-rw-rw---- 1 27 27 36085246 Aug 2 13:01 ae-cloud-1-slow.log
-rw-rw---- 1 27 27 56 Aug 1 01:27 auto.cnf
-rw-rw---- 1 27 27 50331648 Aug 2 13:01 ib_logfile0
-rw-rw---- 1 27 27 50331648 Aug 1 01:59 ib_logfile1
-rw-rw---- 1 27 27 146800640 Aug 2 13:01 ibdata1
drwx----- 2 27 27 4096 Aug 1 01:27 mysql
-rw-r--r-- 1 27 27 7 Aug 1 01:27 mysql_upgrade_info
drwx----- 2 27 27 4096 Aug 1 01:27 performance_schema
drwx----- 2 27 27 20 Aug 1 01:59 test
drwx----- 2 27 27 104 Aug 1 22:32 test2
```


\$ docker run [OPTS] <image-name> <command>;

```
[root@ae-cloud-1 ~]# docker run --rm --volume=volumename-1:/opt/ichi/ --volume=volumename-2:/opt/ni/ --name="volumes-test" centos ls -l /opt/
total 0
drwxr-xr-x 2 root root 6 Aug 18 05:24 ichi
drwxr-xr-x 2 root root 6 Aug 18 05:24 ni
[root@ae-cloud-1 ~]# ls /var/lib/docker/volumes/
0da5ae3b30cfd1d53fdcb446148eb58018e181247a7876f4b391b057d8da02d78  ae4fbe8bb128282591359dcaa96dlb35cde202b7e9455f37c0939b2ea8a3d274
1386bc182bb2ee70eca6908dee9c3f12ac4c91bc31ff5e4bfb0ba93e8642ea08  afc3da6f3ee3557a7519974c440e806605492aa776f3207777d18f947ccab255
14870c9b5e9b2a13b7a5c0a4af7a47311797f1f4a8cf0527c3bcee65115d6313  b53c32004ab83e798fc6a679b92b411287d59b34e9904568bef322582b90ec6c
25b41f3a53bd7a17feb00e79d63cb4fb48d378d9cfef5519b50cb1b4bef4c25  b7a53db5ff526833348e82159016576d9180b56953d17b0ab563a9ae911f0e4c
2d50dc246e58f1668ab25726421993077784b47064da2a45769bfff61b652465  b7dbd6614a57ac0a67d427e286662c4caa12fe727f31cd7776abf75678d65e77
2deca47e718330147dce1a5207703506701f961c05a7395afd0075d55234f154  bc7c5648e88d2141c058860a1cc3c164df7168848faa6873e5bb9187ca34d629
3c886032daf77030478c681aa3e9e9a64f9966f4d82913642cbb2327234a3b3c  bdd35e25189398703eb28d99b65d2d2e965ce7ea3826c30734862917081f0762
4bb325a5d5ec021e4a6c2fee35bb9a31ba9d2b8e8ed8a9045ad70b596c790f5e  bff947e4ae28a85df9c52d6e6b8fc81ba11416904a5b034b37b836247d6c5952
5258ba97cf83ce75786bd696fea3da491c5ffd1dcefc102828c9e41d562af90c  c3f002027f2a491d75350c93e70ee227c6886b92588a817e4c7balac2ba96c15
56e8d9c33151046e27b4003563cac754ec309e83c72f4f2eeaf016e701274851  d48aaf0b3aa39154bd2015f35f86dd29fe5008226b258601279b823e549d1506
5d598a3d53b5db2f09f843a1a9325966d810f328a1c9b72d27f38f85db756660  da28e4bb33a50538d0e0ceee2276be04d120c4242f396c54e9c1b4a1b7d4200e
6db56c287d123d59ecf6a3ef7f0c6c040fc89d0e27267e325fa40cf7585f7c6d  dbead0d5a9d2aa0e34b5ebf243c94b4f85a3eb04cfd79556eed556af69435270
84173f01cbd9407207d0ba4fef7a5c2ecb6faa98b1108c562cb9b56d8dccc072  e04139d48ea68c6a97c0e3966bd220151328d1bca98ea357e57deab17dae3a78
8800d27e0ef072124393eaf98ce6723adead72427f886b4583893ec0384ba20f  ed03186751e2506b94a5623156872a3c8ca23f9ca8cd1e747d798d29c482dd08
90cc54c97e0b3b21bdc64998e8bc463f1fbba81d5c097169c4070603a24dfd7a  mysql
9119060e5302e75aebc3fab5d87db84aab2123505a376b04af21fb0f99895429  new_volume
97814e36070b7f259a6eaa1e22dc055e8901e479cb493a5b99a7eea75b13f034  volumename-1
a15b19c82ff55468796f16a9a58b270492f8ab685a617137d6d3ed3efb03ce53  volumename-2
aa49d249d145683aac72c221ef8140384198ccc392514e05d02ff1c6f0499467
```

\$ docker network create --subnet=<IP_addr/mask> <network-name>;

```
[root@ae-cloud-1 ~]# docker inspect --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' awesome_poitras
172.17.0.3
[root@ae-cloud-1 ~]# docker run -dit --name="network-test-1" --ip=177.17.0.4 centos /bin/bash
1713c56732442043b84cc39ba3c72c52f435cf86a6c82732ec94e822cacc8a6c
docker: Error response from daemon: User specified IP address is supported on user defined networks only.
[root@ae-cloud-1 ~]# docker network create --driver=bridge --subnet=177.0.0.0/16 mycustomnetwork
0ebdc0b7209d04f526baf954cd6ea4c4718b3e7b74ba7e26591b276e2c698618
[root@ae-cloud-1 ~]# docker run -dit --name="network-test-2" --net=mycustomnetwork --ip=177.0.0.2 centos /bin/bash
00720b9fc2f2ad1407e660e4e23450510adf2edd6fb2e927ff9709e1f574fc58
[root@ae-cloud-1 ~]# docker inspect --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' network-test-2
177.0.0.2
```

\$ docker diff <container-name>;

```
[root@ae-cloud-1 ~]# docker diff interactive-shell
C /etc
C /etc/ld.so.cache
C /usr
C /usr/lib64
A /usr/lib64/libpcap.so.1
A /usr/lib64/libpcap.so.1.5.3
C /usr/share
C /usr/share/doc
A /usr/share/doc/nmap-ncat-6.40
A /usr/share/doc/nmap-ncat-6.40/examples
A /usr/share/doc/nmap-ncat-6.40/examples/scripts
C /usr/bin
A /usr/bin/nc
A /usr/bin/ncat
C /root
A /root/.bash_history
C /var
C /var/lib
C /var/lib/yum
C /var/lib/yum/history
```

- En este caso yo había ejecutado `yum install nc`

¿Preguntas?



PERCONA

Laboratorio en grupos

- GRUPO 1
 - Van a trabajar con MySQL 5.5 del repositorio oficial de Docker
- GRUPO 2
 - Van a trabajar con Percona Server 5.6 del repositorio oficial de Docker
- GRUPO 3
 - Van a trabajar con Percona Server 5.7 del repositorio de Percona

Laboratorio en grupos

- GRUPO 1

- `docker pull mysql:5.5`

- GRUPO 2

- `docker pull percona:5.6`

- GRUPO 3

- `docker pull percona/percona-server:5.7`

Laboratorio en grupos

- GRUPO 1
 - `docker images mysql`
- GRUPO 2
 - `docker images percona`
- GRUPO 3
 - `docker images percona/percona-server`

Laboratorio en grupos

- GRUPO 1

- `docker run --name "nombre-grupo-1" -p 9876:3306 \`
`-e MYSQL_ROOT_PASSWORD=grupo1 -d mysql:5.5`

- GRUPO 2

- `docker run --name "nombre-grupo-2" -p 8765:3306 \`
`-e MYSQL_ROOT_PASSWORD=grupo2 -d percona:5.6`

- GRUPO 3

- `docker run --name "nombre-grupo-3" -p 7654:3306 \`
`-e MYSQL_ROOT_PASSWORD=grupo3 -d percona/percona-server:5.7`

Laboratorio en grupos

- GRUPO 1

- `mysql --host=127.0.0.1 --port=9876 --user=root \`
`--password=grupo1`

- GRUPO 2

- `mysql --host=127.0.0.1 --port=8765 --user=root \`
`--password=grupo2`

- GRUPO 3

- `mysql --host=127.0.0.1 --port=7654 --user=root \`
`--password=grupo3`

Laboratorio en grupos

- GRUPO 1

- `mysql> SHOW VARIABLES LIKE '%version%';`

- GRUPO 2

- `mysql> SHOW VARIABLES LIKE '%version%';`

- GRUPO 3

- `mysql> SHOW VARIABLES LIKE '%version%';`

- Para salir del servidor mysql:

- `mysql> QUIT;`

Laboratorio en grupos

- GRUPOS 1, 2 y 3
 - `docker ps`
 - `docker top "nombre-del-contenedor"`
 - `docker logs "nombre-del-contenedor"`
 - `docker stop "nombre-del-contenedor"`
 - `docker ps -a`
 - `docker start "nombre-del-contenedor"`

¡Gracias!



**DATABASE PERFORMANCE
MATTERS**