

# Referencia Rápida Docker

frjaraur/DockerQuickReference

## Instalación de productos

### Instalación docker community.

```
$ curl -sSL https://get.docker.com/ | sh
- Para versiones experimentales o de test (RCs)
$ curl -sSL https://[test|experimental].docker.com/ | sh
```

**Instalación docker commercially supported (la guía está actualizada a Diciembre de 2016, consultar <https://docs.docker.com/cs-engine/install/> para una versión actualizada).**

```
- RedHat
$ sudo rpm --import "https://sks-keyservers.net/pks/lookup?
op=get&search=0xee6d536cf7dc86e2d7d56f59a178ac6c6238f52e"
$ sudo yum install -y yum-utils
$ sudo yum-config-manager --add-repo
https://packages.docker.com/1.12/yum/repo/main/centos/7
$ sudo yum install docker-engine

- Ubuntu
$ curl -s 'https://sks-keyservers.net/pks/lookup?
op=get&search=0xee6d536cf7dc86e2d7d56f59a178ac6c6238f52e' | sudo apt-key add
--import
$ sudo apt-get update && sudo apt-get install apt-transport-https
$ sudo apt-get install -y linux-image-extra-$(uname -r) linux-image-extra-virtual
$ echo "deb https://packages.docker.com/1.12/apt/repo ubuntu-trusty main" |
sudo tee /etc/apt/sources.list.d/docker.list
$ sudo apt-get update && sudo apt-get install docker-engine
```

### **Instalación docker-compose (para versión 1.9, Diciembre 2016)**

```
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.9.0/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

## Gestión de Docker Engine

```
$ sudo systemctl start|stop|restart docker
```

### Configuración de Docker Engine

Copiar el fichero de configuración /lib/systemd/system/docker.service que viene por defecto durante la instalación del software en /etc/systemd/system/docker.service para modificar un fichero customizado en lugar del anterior.

Todos los cambios realizados sobre los ficheros unit de systemd tienen que ser seguidos de “sudo systemctl daemon-reload” para que tengan efecto.

#### NOTA:

Podemos proporcionar acceso local al socket de docker agregando usuarios al grupo docker.

Para realizar cambios sobre el demonio docker modificaremos el arranque del mismo (ExecStart).

```
# dockerd
--host fd:// <--
--host unix:///var/run/docker.sock
--host tcp://[IP]:[Puerto]
--debug
--log-level=info|warning|error
```

### Gestión de versiones de cliente y Docker Engine:

```
$ sudo docker info
```

```
$ sudo docker version
```

## Gestión de Imágenes

### Creación de Imágenes:

```
$ sudo docker build [--tag etiqueta] --file [fichero_dockerfile]  
[directorio_de_trabajo]
```

#### NOTA:

Para referenciar una imagen podemos usar el formato completo [registro]:[puerto]/[usuario]/[nombre\_imagen:etiqueta] o bien su identificador.

### Revisión de creación de imagen:

```
$ sudo docker images
```

```
$ sudo docker history [nombre_imagen:etiqueta]
```

```
$ sudo docker inspect [nombre_imagen:etiqueta]
```

### Borrado de imágenes:

```
$ sudo docker rmi imagen
```

#### NOTA:

Para conectar a un registro de imágenes es necesario validarse usando

```
"docker login [registro]:[puerto]"
```

### Descarga de imágenes:

```
$ sudo docker pull [registro]:[puerto]/[usuario]/  
[nombre_imagen:etiqueta]
```

### Subida de imágenes:

```
$ sudo docker push [registro]:[puerto]/[usuario]/[nombre_imagen:etiqueta]
```

#### NOTA:

Si se omite la etiqueta al usar imágenes, se usará por defecto "latest"

### Gestión de Etiquetas:

```
$ sudo docker tag ORIGEN_[registro]:[puerto]/[usuario]/  
[nombre_imagen:etiqueta] DESTINO_[registro]:[puerto]/[usuario]/  
[nombre_imagen:etiqueta]
```

### Guardar/Cargar Imágenes:

```
$ sudo docker save [-o fichero_tar] imagen:etiqueta
```

```
$ sudo load < fichero_tar_de_imagen
```

### Búsqueda de imágenes en DockerHub:

```
$ sudo docker search patrón_a_buscar
```

## Gestión de Contenedores

## **Creación de contenedores:**

```
$ sudo docker run
    [--tty]
    [--interactive]
    [--name "nombre_contenedor"]
    [--detach]
    [--env variable=valor]
    [--publish ip_host:puerto_host:puerto_contenedor]
    [--publish-all]
    [--rm]
    [--network=red_especifica]
    [--restart always|unless-stopped]
    [--volume NOMBRE_VOLUMEN:/_path_punto_montaje_contenedor]
    [--volume
/_path_origen_datos_host:/_path_punto_montaje_contenedor]
    [--volume /_path_punto_montaje_contenedor]
    [registro]:[puerto]/[usuario]/nombre_imagen:etiqueta
    [comando y argumentos]
```

## **Parada y arranque de contenedores existentes:**

```
$ sudo docker start contenedor
```

```
$ sudo docker stop contenedor
```

```
$ sudo docker restart contenedor
```

```
$ sudo docker kill contenedor
```

## **Revisión de estado de contenedores:**

```
$ sudo docker ps [--all] [--quiet] [--filter nombre=valor] [--no-trunc]
```

```
$ sudo docker inspect [--format formato_salida] contenedor
```

```
$ sudo docker logs [--follow] [--tail número_líneas] contenedor
```

## **Conectarse a un contenedor en ejecución:**

```
$ sudo docker attach contenedor
```

```
$ sudo docker exec [--tty] [--interactive] contenedor
[comando_con_argumentos]
```

## **Borrar contenedores existentes:**

```
$ sudo docker rm [--force] [--volumes]
```

## **Revisar cambios realizados en el contenedor respecto de su imagen origen:**

```
$ sudo docker diff contenedor
```

## **Actualizar cambios realizados en contenedor en una imagen:**

```
$ sudo docker commit contenedor [registro]:[puerto]/
[usuario]/nombre_imagen:etiqueta
```

NOTA:  
Para referenciar un contenedor podemos usar su nombre o bien su identificador.

## **Gestión de Volúmenes:**

### **Crear Volumen:**

```
$ sudo docker volume create [--driver driver] [--opt valores] [--name nombre_volumen]
```

### **Revisar volúmenes:**

```
$ sudo docker volume ls
```

```
$ sudo docker volume inspect nombre_volumen
```

### **Borrar volúmenes:**

```
$ sudo docker volume rm nombre_volumen
```

## **Gestión de Comunicaciones:**

### **Creación de redes:**

```
$ sudo docker network create [--driver driver] [--opt valores] [--subnet subred] [--gateway gateway] nombre_de_red
```

### **Conectarse a una red existente:**

```
$ sudo docker network connect [--ip IP] red contenedor
```

```
$ sudo docker network disconnect [--force] red contenedor
```

### **Revisión de redes existentes:**

```
$ sudo docker network ls
```

```
$ sudo docker network inspect red
```

## **Clusterización (swarm mode)**

### **Gestión cluster:**

```
$ sudo docker swarm init
```

```
$ sudo docker swarm join-token [worker|manager]
```

```
$ sudo docker swarm leave [--force]
```

### **Revisión de nodos asociados al cluster**

```
$ sudo docker node ls
```

```
$ sudo docker node inspect nodo
```

```
$ sudo docker node promote|demote nodo
```

### **Revisión de tareas desplegadas en un nodo**

```
$ sudo docker node ps
```

### **Borrado de nodos del cluster**

```
$ sudo docker node rm
```

## **Gestión de servicios (swarm mode)**

### **Creación de servicios**

```
$ sudo docker service create
```

```

[--tty]
[--interactive]
[--name "nombre_contenedor"]
[--mode replicated|global]
[--env variable=valor]
[--publish ip_host:puerto_host:puerto_contenedor]
[--replicas número_de_réplicas]
[--rm]
[--network red_específica]
[--restart-condition none|on-failure|any]
[--mount
type=volume,source=volumen_origen,destination=/_path_interno_contenedor]
[registro]:[puerto]/[usuario]/nombre_imagen:etiqueta
[comando y argumentos]

```

### **Revisión de servicios existentes y tareas asociadas**

```
$ sudo docker service ls
```

```
$ sudo docker service inspect servicio
```

```
$ sudo docker service ps servicio
```

### **Escalado de tareas asociadas a servicios**

```
$ sudo docker service scale servicio=número_de_tareas
```

NOTA:  
“docker service update” permite actualizar parámetros (e incluso la imagen asociada para la creación de tareas) relativos al servicio.

### **Borrar servicio creado**

```
$ sudo docker service rm servicio
```

## **Aplicaciones multicontenedor**

### **NOTA:**

swarm mode no soporta docker-compose V1 y V2 (la aplicación se despliega en un único host). En docker 1.13 se añade la capacidad de carga de ficheros V3.

No se muestran en esta guía entradas para application bundle debido a su estado experimental y que no soporta volúmenes (Diciembre 2016).

### **Despliegue de stack completo de servicios para una aplicación multicontenedor en swarm mode**

```
$ sudo docker stack deploy --compose-file fichero_compose_V3.yml
nombre_stack
```

### **NOTA:**

En caso de no usar swarm mode, docker-compose puede usar versiones anteriores a V3.

### **Despliegue aplicación multicontenedor (V1, V2)**

```
$ sudo docker-compose [-p nombre_proyecto] [-f nombre_fichero] up [-d]
```

### **Borrado de aplicación multicontenedor (V1, V2)**

```
$ sudo docker-compose [-p nombre_proyecto] [-f nombre_fichero] down
```

### **Arranque/Parada de aplicación ya desplegada (V1, V2)**

*\$ sudo docker-compose [-p nombre\_proyecto] [-f nombre\_fichero] start|stop|restart*

### **Revisión de logs de aplicativo**

*\$ sudo docker-compose [-p nombre\_proyecto] [-f nombre\_fichero] logs*

### **Revisión de contenedores/servicios de aplicativo**

*\$ sudo docker-compose [-p nombre\_proyecto] [-f nombre\_fichero] ps*

*\$ sudo docker-compose [-p nombre\_proyecto] [-f nombre\_fichero] config [--services]*

### **Escalado de servicios asociados a una aplicación**

*\$ sudo docker-compose [-p nombre\_proyecto] [-f nombre\_fichero] scale nombre\_servicio=número\_de\_contenedores\_a\_levantar*

## **Referencia rápida Dockerfile**

**FROM** [registro]:[puerto]/[usuario]/nombre\_imagen:etiqueta

**RUN** comandos\_a\_ejecutar\_para\_completar\_la\_imagen

**ADD** origen(puede ser una\_url\_con\_ficheros\_empaquetados) destino

**COPY** origen(path\_relativo) destino

**ENV** variable valor

**ENTRYPOINT** path\_completo\_al\_ejecutable\_

**CMD** comandos\_o\_argumentos\_para\_entrypoint

**EXPOSE** puerto\_de\_aplicativo\_a\_exponer\_al\_mundo

**VOLUME** path\_de\_contendio\_de\_datos\_que\_se\_usará\_como\_volumen

**WORKDIR** path\_de\_trabajo\_para\_las\_siguientes\_sentencias\_en\_la\_imagen

**USER** usuario\_que\_ejecutará\_las\_siguientes\_sentencias

## **Referencia Rápida docker-compose.yml**

**version** version\_entre\_comillas (1,2 o 3)

**services** comienza\_la\_definición\_de\_servicios

**name** nombre\_del\_servicio\_(si no se especifica nada, tomará el nombre de la definición)

**build** directorio\_de\_datos\_para\_creación\_de\_imagen\_de\_servicio

**image** imagen\_de\_referencia\_para\_el\_servicio

**depends\_on** nombre\_del\_servicio\_del\_que\_depende

**command** argumento\_o\_comandos\_a\_pasar\_al\_entrypoint\_de\_la\_imagen

**environment** lista\_con\_variables\_definidas\_para\_el\_servicio

**labels** lista\_con\_etiquetas\_para\_el\_servicio

**networks** lista\_con\_las\_redes\_disponibles\_para\_el\_servicio

**ports** lista\_con\_los\_puertos\_a\_exponer\_al\_mundo

**volumes** lista\_con\_los\_volumenes\_disponibles\_para\_el\_servicio\_y\_opciones