



## Gestión de imágenes

```
docker images
```

- Información de imágenes locales disponibles.

```
docker search ubuntu
```

- Busca la imagen “ubuntu” en el repositorio remoto (por defecto Docker Hub).

```
docker pull alpine
```

- Descarga localmente imagen “alpine”.

```
docker history alpine
```

- Muestra la historia de creación de la imagen “alpine”.

```
docker rmi ubuntu:14.04
```

- Elimina localmente la imagen “ubuntu” con tag “14.04”.

```
docker rmi $(docker images -q)
```

- Borra toda imagen local que no esté siendo usada por un contenedor.

```
docker rm IDCONTENEDOR
```

- Borra un contenedor con IDCONTENEDOR.

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

- Para todos los contenedores del sistema.

```
docker rm $(docker ps -a -q)
```

- Borra todos los contenedores parados del sistema.

```
docker system prune -a
```

- Borra todas las imágenes y contenedores parados del sistema.



## Creación de imágenes a partir de contenedores

```
docker commit -m “comentario” IDCONTENEDOR usuario/imagen:version
```

- Hace commit de un contenedor existente a una imagen local.

```
docker save -o copiaSeguridad.tar imagenA
```

- Guarda una copia de seguridad de una imagen en fichero “.tar”.

```
docker load -i copiaSeguridad.tar
```

- Restaura una copia de seguridad de una imagen en fichero “.tar”.



Unidad 04 - Autor: Sergi García Barea

## Docker Hub

---

```
docker login
```

- Permite introducir credenciales del registro (por defecto “Docker Hub”).

```
docker push usuario/imagen:version
```

- Permite subir al repositorio una imagen mediante “push”.

## Ejemplo de Dockerfile

---

```
FROM alpine
LABEL maintainer="email@gmail.com"
#Actualizamos e instalamos paquetes con APK para Alpine
RUN apk update && apk add apache2 php php-apache2 openrc tar
#Copiamos script para lanzar Apache 2
ADD ./start.sh /start.sh
#Descargamos un ejemplo de <?php phpinfo(); ?> por enseñar como bajar algo de Internet
#Podría haber sido simplemente
#RUN echo "<?php phpinfo(); ?>" > /var/www/localhost/htdocs/index.php
ADD https://gist.githubusercontent.com/SyntaxC4/5648247/raw/94277156638f9c309f2e36e19bff378ba7364907/info.php
/var/www/localhost/htdocs/index.php
# Si quisiéramos algo como Wordpress haríamos
#ADD http://wordpress.org/latest.tar.gz /var/www/localhost/htdocs/wordpress.tar.gz
#RUN tar xvzf /var/www/localhost/htdocs/wordpress.tar.gz && rm -rf /var/www/localhost/htdocs/wordpress.tar.gz

# Usamos usuario y grupo www-data. El grupo lo crea Apache, pero si quisiéramos crear grupo
# Grupo www-data RUN set -x && addgroup -g 82 -S www-data
# Creamos usuario www-data y lo añadimos a ese grupo
RUN adduser -u 82 -D -S -G www-data www-data
# Hacemos todos los ficheros de /var/www propiedad de www-data
# Y damos permisos a esos ficheros y a start.sh
RUN chown -R www-data:www-data /var/www/ && chmod -R 775 /var/www/ && chmod 755 /start.sh
#Indicamos puerto a exponer (para otros contenedores) 80
EXPOSE 80
#Comando lanzado por defecto al instalar el contenedor
CMD /start.sh
```

- Ejemplo de fichero “Dockerfile”.