

Tópicos	Conceptos
<h2>UF3: Lo del cocker</h2>	
<h3>ÓRDENES DE DOCKER BÁSICAS</h3>	
<p>Estos ejecutan microservicios para con la cerda del pueblo (es decir, tu madre) [la puerca] {la gorrina} /la tocina/ -la maconga- ¡la grasienta; la Barrabás ¿la zarrapastrosa? ¡la borrega! :la genocida de hamburguesas:</p>	
<p>Docker es una herramienta de administración de containers. Estos ejecutan microservicios.</p>	
<p>Partimos de imágenes RO que tienen una capa RW por encima. La suma de estas dos capas junto a los procesos componen un container. Docker es una marca de cosas (lo dijo el Canet en clase).</p>	
<p>Imagen --> container --> imagen muestra --> Docker Hub con imágenes</p>	
Comprobar que el usuario pertenece al grupo <i>docker</i> .	id
Si no es así, hay que usar esto:	usermod -aG docker user
Salir de un container.	exit
Lo típico pa empezar y comprobar que el docker se ha instalado bien.	docker run hello-world
Hacer una imagen de un container.	docker commit -m "server web" f27 myweb
Ver las imágenes.	docker images
Ver el historial de órdenes de un container.	docker history myweb:latest
Listar los containers que se están ejecutando.	docker ps
Es lo mismo, en teoría.	docker container ls
Listar los que se han ejecutado.	docker container ls -a
Inicarlo sin entrar en él.	docker start myhost
Pararlo.	docker stop myhost
Entrar.	docker attach myhost
Listar sus procesos (ps).	docker top myhost
Eliminarlo (es recomendable usar el TAB después de 'rm').	docker rm myhost
Eliminar imágenes.	docker rmi hello-world:latest
FALTA INFO	docker inspect myhost less
<p>Importante: si solo hay una imagen, la borra, pero si existen varias iguales con diferentes etiquetas, primero elimina las etiquetas.</p>	
Examinar el contenido.	docker inspect myhost less
Listar las redes del docker.	docker network ls
Lo mismo pero para examinar las redes de containers.	docker network inspect bridge less
Para hacer una imagen de un container.	docker commit -m "server web" f27 myweb
Fedora32.	docker run -it fedora:32 /bin/bash

Lo crea, hace la orden *cal* y hace *exit* automáticamente.
Se queda en stand-by.

Lo mismo pero autodestruyéndose (--rm).

Descargar una imagen.

```
docker run -it fedora:32 cal
```

```
docker run --rm -it fedora:32 cal
```

```
docker pull fedora:33
```

Nombre que ha de tener una imagen para subirla.
La etiqueta por defecto es 'latest'.

```
usuario/imagen:etiqueta
```

Poner una etiqueta.

```
docker tag
```

Pa iniciar sesión.

```
docker login
```

Ojito que esto crea un .json con la contraseña **SIN ENCRIPtar**.

Pa cerrar sesión.

```
docker logout
```

Esto también elimina dicho .json.

Subir la imagen.

```
docker push usuario/imagen:etiqueta
```

ÓRDENES DE DOCKER PARA CREAR UNA WEB

Crear un servidor web desde Docker y a lo bestia.

```
echo "hola" > /var/www/html/index.html
/etc/init.d/apache2 start
/etc/init.d/apache2 status
nmap localhost
ip a
```

Luego se pone la IP a secas en el navegador web.

Ejecutarlo con identificador (--name myhost),
asignando como host el que se diga (-h myhost) y
de forma interactiva (-it)
con un shell (/bin/bash).

```
docker run --name myhost -h myhost -it debian /bin/bash
```

Una vez que se ejecuta esto, al ver el prompt y usando *id* se puede comprobar
que se está dentro del container.

OPCIONES DE DOCKER RUN

Detach.

```
-d
```

Los containers en modo detach (-d) tienen un proceso que está en foreground.

```
docker --rm -d debian date
```

Host.

```
-h
```

Interactive terminal.

```
-it
```

Se autodestruye al concluir la orden indicada	--rm [opciones] orden
Publish.	-P
Abrir el puerto 8080 de mi PC apuntando al 80. Se comprueba con localhost:8080.	-p 8080:80
Ponerle nombre	--name web
Host.	-h web
Montar un fichero de fuera del container a uno que está dentro.	--mount type=bind,src=/tmp/index.html,dst=/var/www/html/index.html
Volumen.	-v volumen:/route/to/dir
Meter variables en un environment. CREO QUE -e TAMBIÉN HACE EXPORT (REPASAR)	-e DB="training"
	docker run --name f27 -h f27 -it fedora:27 /bin/bash
Dirección IP de la red por defecto de docker: Sus contenedores tienen conectividad entre ellos pero solo vía IP, no por nombre. Solo se puede por nombre si la red es DEFINIDA.	172.17.0.0/16
Dirección IP por defecto del host, que tiene los containers:	172.17.0.1/16
Dirección IP del primer container.	172.17.0.2/16
Se puede hacer ping desde dentro del container hacia las IPs que contiene y hacia otras externas (www.escoladeltreball.org, por ejemplo). También desde el terminal del propio ordenador hacia el container.	
PROPAGAR EL PUERTO DE UN CONTAINER AL HOST	
Primero se crea el container.	docker run --rm --name f27 -h f27 -p 80:80 -it dockerusername/myweb /bin/bash
Iniciar el servicio de Apache.	/sbin/httpd
docker build -t tag .	el . es "directori de contex"
DOCKERFILE	
FROM fedora:27	
LABEL subject="webserver"	
LABEL author="Autor"	
RUN dnf -y install procps iproute iputils nmap httpd	
RUN mkdir /opt/docker	OPCIONAL
COPY * /opt/docker	OPCIONAL
RUN chmod +x /opt/docker/startup.sh	
WORKDIR /opt/docker	
CMD /opt/docker/startup.sh	Usa como comando lo que hay en el archivo startup.sh.
EXPOSE 80	Abre el puerto 80.

UTILIDADES PARA VER QUÉ PASA EN LOS CONTAINERS

Dentro de uno, apt-get update y apt-get install procps iproute2 apache2

procps	Sirve para ver procesos de los containers.
iproute2	Órdenes como ip a.
iputils	(repasar)
nmap	Sirve para ver cosas de IPs (repasar).
httpd	El puto apache. En Debian es apache2 o httpd2 o algo2.

Ver lo que tiene un docker activo.

Copiar algo de un container (este igual habría que comprobarlo).

Copiar algo a container (este funciona 100%).

```
docker exec -it web /bin/bash
docker cp container:/var/www/html/index.html index.html
docker cp index.html container:/var/www/html/index.html
docker exec -it web cat /var/www/html/index.html
```

DOCKER VOLUME

Sus cosas están en formato .json.

Crear volumen.	docker volume create patata
Listarlos.	docker volume ls
Prune es hacer una recopilación y eliminar lo inútil.	docker volume prune
	docker volume inspect

DENTRO DE UN DOCKERFILE

Si se pone ENTRYPOINT ["date"] hace esta orden y ya. Cuando se ejecuta el comando para encender el contenedor junto a otra orden, esta última no se ejecuta.	docker run --rm -it m05entry Fri May 27 06:36:33 UTC 2022 docker run --rm -it m05entry /bin/bash date: invalid date '/bin/bash'
Si se usa CMD ["date"] y al ejecutar el contenedor no se indica otra orden, simplemente esta se ejecuta y ya. Si se indica otra orden, si que funciona.	docker run --rm -it m05cmd Fri May 27 06:36:33 UTC 2022 docker run --rm -it m05cmd cat /etc/os-release (output de os-release)
Esto ocurre porque el ENTRYPOINT es la orden que se tiene que ejecutar POR DEFECTO y el CMD es lo mismo pero se puede sobrescribir. Si hay una orden en el ENTRYPOINT y se indica otra orden, esta última se procesa como argumento, por lo que puede haber errores. Esto se puede explicar mejor.	Para sobrescribir el ENTRYPOINT del Dockerfile se puede usar docker run --rm --entrypoint orden -it m05entry, por ejemplo: docker run --rm --entrypoint /bin/bash -it m05entry

Al ENTRYPOINT se le puede pasar un fichero con un script de este estilo:
Como este script estaba en un contenedor, hace falta usar
docker cp container:/usr/bin/opcions.sh . # Ojo con el punto
vim Dockerfile

```
FROM fedora
RUN dnf -y install util-linux
COPY ./opcions.sh /usr/bin/opcions.sh
RUN chmod +x /usr/bin/opcions.sh
ENTRYPOINT ["opcions.sh"]
```

```
# !/bin/bash

opcio="dia"

if [ $# -eq 1 ]; then
    opcio=$1
fi

case $1 in
    "dia")
        date;;
    "sysinfo")
        cat /etc/os-release;;
    "calendari")
        cal;;
    esac
```

DOCKER REDES

Teniendo la red 172.17.0.0/16, que es la por defecto (**bridge**), crearemos 2 más:
alumnos 172.18.0.0/16 (contenedores identificables por nombre)
profes 172.22.0.0/16

NOTA; Dos redes distintas de containers no se pueden comunicar entre ellas.

```
FROM debian
RUN apt-get update && apt-get -y install openssh-client openssh-server procps
iproute2 iputils-ping nmap telnet
COPY useradd.sh /tmp
RUN bash /tmp/useradd.sh
WORKDIR /tmp
CMD /etc/init.d/ssh start -D # Esto inicia el demonio del ssh en foreground.
```

```
docker network ls
docker network create alumnos
docker network inspect alumnos
docker network rm alumnos
```

```
docker run --rm --name ssh.alumnes.org -h ssh.alumnes.org --network
alumnes -d m05ssh
```