



**Francesc
de Borja Moll**
Centre Integrat de
Formació Professional

Manual d'ús de Docker

Francesc Mut / Albert Perelló
CIFP Francesc de Bora Moll
1r ASIX Dual

ÍNDEX

01. Introducció.....	Pàg. 01
01.1. Què és Docker?	
01.2. Què ens ofereix Docker Hub	
02. Primeres passes: Instal·lació del programari Docker.....	Pàg. 02
01.1. Inici de sessió	
01.2. Creació d'un contenidor	
03. Ús d'una aplicació dockeritzada.....	Pàg. 05
03.1. Pausa o tancament dels contenidors	
03.2. Eliminació dels contenidors	
04. Pull: les nostres imatges a la web.....	Pàg. 06
04.1. Creació d'un repositori	
04.2. Càrrega de contingut al repositori	
05. Bibliografia/Referències.....	Pàg. 08

01. Introducció

La virtualització per contenidors, per a funcionar correctament, necessita un sistema o una aplicació en el propi contenidor, sinó no pot funcionar. Per això, és important instal·lar Docker Engine, i el Docker Hub, per a poder aconseguir aquestes imatges de manera senzilla.

Docker Hub ens dona un seguit d'eines, i ara en veurem les més importants:

Primera eina: **els repositoris**. Els repositoris és una de les eines més crucials de Docker Hub, si no la més important, a part del propi motor de Docker. Un repositori es un espai accessible per al seu autor o la seva comunitat si l'autor el posa públic, on es poden pujar imatges, aplicacions i programes per a un contenidor. L'autor podrà actualitzar la seva imatge amb noves versions, i l'usuari descarregar-la al seu dispositiu i executar-la. Aquesta imatge es guarda als servidors de Docker Hub i estarà sempre accessible per a l'autor i/o els usuaris de Docker Hub.

Segona eina: **les imatges**. Una imatge de contenidor no és més que la còpia d'un contenidor funcional amb una aplicació a sobre d'ell. És una de les grans avantatges de la virtualització per contenidors, la capacitat de treure una còpia del sistema per a poder-la reproduir a una altra màquina en unes altres condicions. Hi ha dues classes d'imatges a destacar:

En primer lloc, estan les *imatges oficials*, que són imatges de contenidors de molta alta qualitat, proveïdes i mantingudes per els mateixos desenvolupadors de Docker. Solen ser imatges base, amb sistemes operatius bàsics com debian o CentOS, que estan pensades per a ser un punt de partida per els desenvolupadors.

En segon lloc, les *imatges de desenvolupador*. Son aquelles imatges proveïdes i mantingudes per el desenvolupador de l'aplicació, no per Docker. Son imatges que a part d'un sistema bàsic, també duen el programari del desenvolupador de l'aplicació, i estan pensades per a ser màquines ja preparades per a funcionar. Com per exemple, la imatge de MySQL, està mantinguda per l'equip de MySQL i està preparada per que comencis a crear una base de dades .

Tercera eina, **la capacitat per muntar equips i organitzacions**. Docker Hub prové una eina per a crear un equip d'usuaris i compartir entre ells certes imatges i repositoris privats. Això ajuda molt en un entorn de desenvolupament d'una aplicació, ja que dona eines a l'equip per a treballar tots en una mateixa imatge. També té integració amb **GitHub** i permet construir imatges des dels repositoris de GitHub.

02. Primeres passes: Instal·lació del programari Docker per començar a aprendre a usar contenidors d'aplicacions.

Explicats els bàsics de funcionament de Docker Hub, anem a començar amb les primeres passes. Per a començar a treballar, necessitarem una conta a Docker Hub. En tenir el compte creat, podrem començar a descarregar l'aplicació Docker Desktop per al nostre ordinador. Com que es una aplicació multiplataforma, serà compatible amb quasi tots els sistemes operatius populars, ja sigui Windows, Mac OS, o distribucions Linux.

Tot seguit hagi acabat l'aplicació de descarregar i instal·lar, executarem unes comprovacions per a saber que tot està correcte. Obrirem la nostra aplicació de terminal preferida, tant sigui PowerShell, CMD, el terminal de Mac OS, urxvt, xterm... i escriurem la comanda següent:

```
docker -v
```

El programa ens haurà de retornar la seva versió i el número de compilació. Si no ens ho retorna, es que hi ha hagut un error en la instal·lació.

També podem comprovar que la instal·lació hagi estat correcta arrancant el contenidor de prova hello-world de docker.

```
PS C:\Users\xescm> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:8c5aeeb6a5f3ba4883347d3747a7249f491766ca1caa47e5da5dfc6b9b717c0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\Users\xescm>
```

Després, començarem iniciant sessió en docker hub amb la comanda

```
docker login
```

Un cop amb la sessió iniciada a Docker Hub, podrem començar a instal·lar imatges de contenidors dels repositoris disponibles. Per exemple, podem descarregar la imatge dockeritzada del sistema Debian teclejant `docker pull debian`.

```
PS C:\Users\xescm> docker pull debian
Using default tag: latest
latest: Pulling from library/debian
e4c3d3e4f7b0: Pull complete
Digest: sha256:8414aa82208bc4c2761dc149df67e25c6b8a9380e5d8c4e7b5c84ca2d04bb244
Status: Downloaded newer image for debian:latest
docker.io/library/debian:latest
PS C:\Users\xescm>
```

Ara anem a crear un contenidor amb Debian dins, i provar d'executar-lo. Per a crear un nou contenidor, escriurem a la línia de comandes el següent:

```
docker run -it debian
```

`docker run` és la comanda que ens crearà el nou contenidor, amb la imatge de Debian. Si afegim la variable `--name` podrem donar un nom al contenidor per a mes facilitat d'ús, si no se li donarà un nom predeterminat.

Per exemple, crearem un contenidor anomenat "debian"

```
PS C:\Users\xescm> docker run --name debian -it debian
root@752df9daa930:/#
```

I com podem comprovar, la nostra terminal està ja dins el directori del nou contenidor. A una altra finestra de terminal, podem executar la comanda `docker ps` que ens donarà la informació necessària sobre l'estat dels nostres contenidors.

```
PS C:\Users\xescm> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
752df9daa930	debian	"bash"	3 minutes ago	Up 3 minutes		debian

03. Ús d'una aplicació dockeritzada: Pausa o tancament dels contenidors, eliminació dels contenidors.

En acabar d'emprar el nostre contenidor, podem tornar al nostre directori escrivint la comanda `exit`.

```
PS C:\Users\xescm> docker run --name debian -it debian
root@752df9daa930:/# exit
exit
PS C:\Users\xescm> 
```

Però el nostre contenidor encara esta en marxa, ho pots comprovar amb la comanda `docker ps`.
Per apagar-ho hauriem d'emprar la comanda:

`docker stop (nom)`

```
PS C:\Users\xescm> docker stop debian
PS C:\Users\xescm> 
```

Si tornem a fer `docker ps`, podem veure que el nostre contenidor ja no es troba en funcionament.

I per últim, podrem eliminar el nostre contenidor amb la comanda: `docker rm (nom)`

```
PS C:\Users\xescm> docker rm debian
PS C:\Users\xescm> 
```

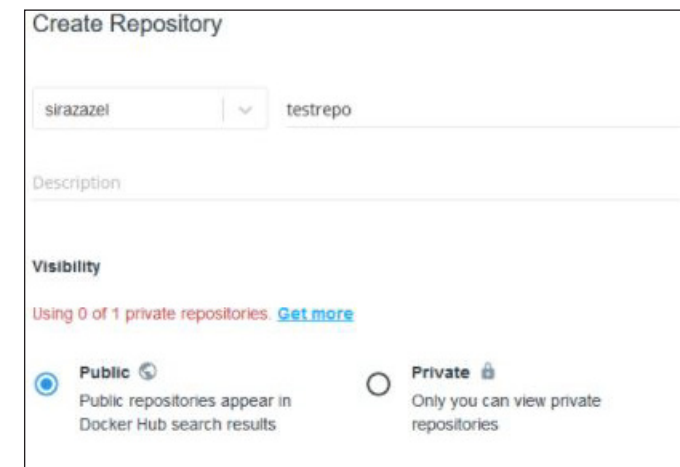
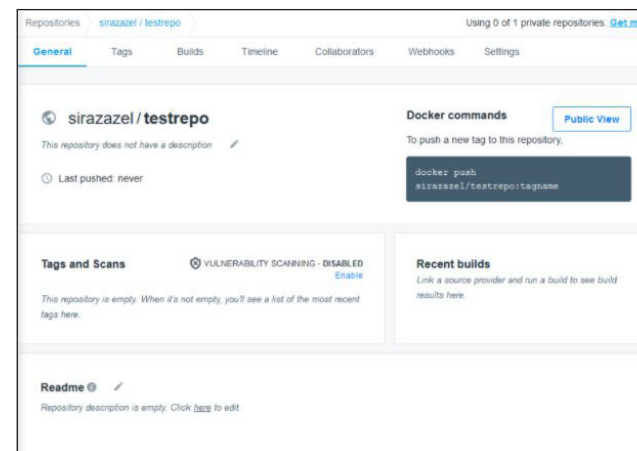
Així, el nostre contenidor deixara de ser accessible, i els seus fitxers, eliminats.

04. Pull: les nostres imatges a la web

Per a compartir les nostres imatges a Docker Hub primer ens familiaritzarem amb els seus repositoris. Començarem creant un nou repositori de proves per a pujar un contenidor creat al nostre ordinador, emprant la web de Docker Hub.

Hem d'iniciar sessió i clicar allà on posa Create New Repository, i després ens sortirà el següent formulari:

Acte seguit ens trobarem a la pàgina principal del nostre primer pepositori:



Ara intentarem fer un push amb la nostra imatge de Debian, emprada en l'exercici anterior.

Primer de tot, hem de mirar quines imatges tenim al nostre ordinador, i seleccionar quina volem pujar. Amb la comanda `docker images`, ens sortiran totes les imatges, els repositoris locals i el seu identificador.

```
PS C:\Users\xescm> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker101tutorial   latest             09930ca33650       3 weeks ago        27.5MB
debian              latest            1510e8501783       3 weeks ago        114MB
alpine/git          latest            94f8849864da       6 weeks ago        28.4MB
hello-world         latest            bf756fb1ae65       10 months ago      13.3kB
```

Volem pujar la segona imatge, que es la que hem creat amb el sistema operatiu de Debian. Per a pujarla correctament, primer l'hem de identificar. La comanda `docker tag` serveix per a identificar les imatges i prepararles per a pujar al nostre repositori.

Necessitarem l'identificador d'imatge, el nostre nom d'usuari, i el nom del nostre repositori a docker hub. L'ús de la comanda és el següent:

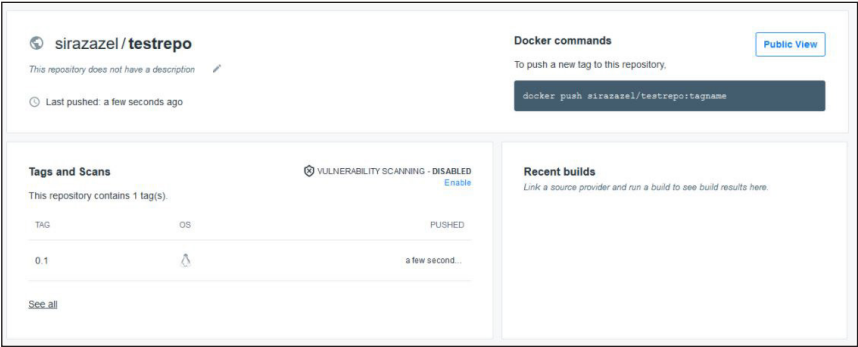
```
docker tag [IMAGE ID] [NOM USUARI]/[NOM REPOSITORI]:IDENTIFICADOR
```

```
docker tag 1510e8501783 sirazazel/testrepo:0.1
```

Aquesta comanda està identificant la imatge de Debian, amb l'identificador 0.1 dins el repositori tes-trepo de la meva carpeta personal.

```
PS C:\Users\xescm> docker tag 1510e8501783 sirazazel/testrepo:0.1
PS C:\Users\xescm> docker push sirazazel/testrepo
The push refers to repository [docker.io/sirazazel/testrepo]
9780f6d83e45: Mounted from library/debian
0.1: digest: sha256:60cb30babcd1740309903c37d3d408407d190cf73015aeddec9086ef3f393a5d size: 529
PS C:\Users\xescm>
```

I quan la comanda push finalitzi, podem anar a la pàgina web de Docker Hub, a la secció dels nostres repositoris, on veurem que recentment s'haurà fet una push al nostre repositori de proves.



Fins aquí les primeres passes de Docker, espero que us hagi ajudat a entendre com funcionen les funcions bàsiques del programa.

05. Bibliografia/Referències

<https://docs.docker.com/docker-hub/>
<https://www.ionos.es/digitalguide/servidores/configuracion/tutorial-docker-instalacion-y-primeros-pasos/>
<https://www.hostinger.es/tutoriales/como-crear-contenedor-docker/>