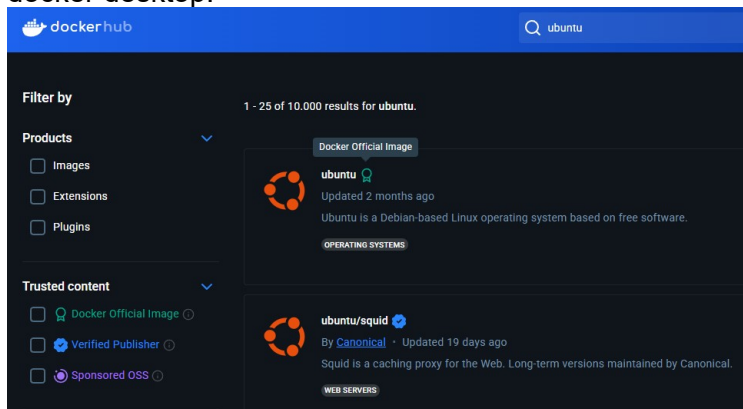


Docker desktop

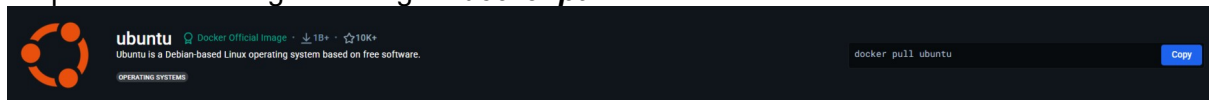
Trabaja a partir de una imagen de un programa (tomcat, ubuntu, node...).

Todas las imágenes oficiales se encuentran en <https://hub.docker.com> o podemos usar docker desktop.



La imagen contendrá lo estrictamente necesario para funcionar (no tiene interfaz gráfica ni editores de texto, etc)

Si queremos descargar la imagen: ***docker pull***



Podemos ver un listado de las imágenes locales (descargadas) haciendo: ***docker images***.

```
C:\Users\ftthog>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
node	latest	e4f23baa3e59	5 weeks ago	1.12GB
tomcat	9.0.100-jdk17-corretto	a2f7c8af5f54	6 weeks ago	501MB
ubuntu	latest	a04dc4851cbc	2 months ago	78.1MB
openjdk	17-ea-slim	1873bf63282d	3 years ago	400MB

A partir de la imagen:

- **Crear un contenedor (o varios):**

docker run:

en 2º plano(imagen ejecutada pero sin interacción):

`docker run -td --name <nombreparaelcontenedor> <imagenocker>`

docker run -td --name ubuntufer ubuntu

para interactuar(en este caso con bash):

`docker run -it <nombre o id del contenedor> bash`

docker run -it ubuntufer bash

- **Borrar un contenedor(el contenedor debe estar parado):**

docker rm:

`docker rm <Id o nombre del contenedor>`

por nombre: ***docker rm ubuntufer***

por Id: ***docker rm f18vjkebn*** (el id es inventado)

- **Parar un contenedor(lo pausa, no lo desactiva):**

docker stop:

`docker stop <Id o nombre del contenedor>`

por nombre: ***docker stop ubuntufer***

por Id: ***docker stop f18vjkebn*** (el id es inventado)

- **volver a activar contenedor:**
docker start:
 docker start <Id o nombre del contenedor>
 por nombre: **docker start ubuntufer**
 por Id: **docker start f18vjkebn** (el id es inventado)
- **borrar una imagen de programa (las imágenes dejan de estar en el host):**
docker rmi:
 docker rmi <Id o nombre del contenedor>
 por nombre: **docker rmi ubuntu**
 por Id: **docker rmi 34srthjkebn** (el id es inventado)
- **Ejecutar un programa en un contenedor :**
docker exec:
 en 2º plano(imagen ejecutada pero sin interacción):
 docker exec -td <nombre del contenedor>
docker exec -it ubuntufer bash (ejecuta bash en el contenedor)
- **Redireccionar puerto (-p) :**
 por defecto el puerto del ordenador para pruebas es el 80
ejemplo::
docker run -d --name tomcatpruebas -p 8080:8080
- **crear una nueva imagen a partir de la que tenemos:**
 docker commit <Id o nombre del contenedor> <nombre nueva imagen>
docker commit ubuntufer nuevoubuntu
- **crear un comprimido de la imagen:**
 docker save -o fichero.tar imagen:tag
- **Compartir una carpeta en docker(situándose en la carpeta que se quiere compartir**
 (por eso el . antes de los :):
docker run -td --name nombrealias -v ./home/node nombreimagen

RESUMEN:

docker run:

- -i (interactive)
- -t (terminal)
- -d (segundo plano)
- -p (redirige el puerto)
- --name(nombre del contenedor)

docker exec:

- -i (interactive)
- -t (terminal)