**COMANDOS DOCKER**

* **docker pull nombre\_imagen: version:** descargará desde el repositorio una imagen con la versión indicada o la última versión (latest) si no indicamos versión.
* **docker run:** Tiene como alrededor de 100 opciones, pero los mas importantes son:
  + **-d o --detach** para ejecutar un contenedor (normalmente porque tenga un servicio) en background.
  + **-e o --env** para establecer variables de entorno en la ejecución del contenedor.
  + **-h o --hostname** para establecer el nombre de red parar el contenedor.
  + **--help** para obtener ayuda de las opciones de docker.
  + **--interactive o -i** para mantener la STDIN abierta en el contenedor.
  + **--ip** si quiero darle una ip concreta al contenedor.
  + **--name** para darle nombre al contenedor.
  + **--net o --network** para conectar el contenedor a una red determinada.
  + **-p o --publish** para conectar puertos del contenedor con los de nuestro host.
  + **--restart** que permite reiniciar un contenedor si este se "cae" por cualquier motivo.
  + **--rm** que destruye el contenedor al pararlo.
  + **--tty o -t** para que el contenedor que vamos a ejecutar nos permita un acceso a un terminal para poder ejecutar órdenes en él.
  + **--user o -u** para establecer el usuario con el que vamos a ejecutar el contenedor.
  + **--volume o -v** para montar un bind mount o un volumen en nuestro contenedor.
  + **--wordirk o -w** para establecer el directorio de trabajo en un contenedor.
* **flag -it:** Este -it es la unión del flag -i (--interactive) y el flag -t (--tty),nos va a permitir interactuar con él.
* **flag -d:** Se ejecuta en modo background o dettach.
* **flag -p (puerto\_en\_host:puerto\_en\_contenedor):** Así es accesible desde el exterior.
* **flag -e (nombre\_variable=valor):** Comprobar y definir si es necesario las variables de entorno que puede tener el contenedor.
* **flag –name:** Para usar nombres en los contenedores elegidos por nosotros.
* **docker exec:** Para entrar en el docker y ejecutar órdenes en ella. Tiene algunas opciones como son las siguientes:
  + **-it (-i y -t juntos)** si vamos a querer tener interactividad con el contenedor ejecutando un shell (/bin/bash normamente). Una vez tenemos el terminal ya podremos trabajar desde dentro del propio sistema.
  + **-u o --user** si quiero ejecutar la orden como si fuera un usuario distinto del de root.
  + -**w o --workdir** si quiero ejecutar la orden desde un directorio concreto.
* **docker cp** que me permite mover ficheros desde mi sistema al contenedor y desde el contenedor a mi sistema.
* **docker ps:** Nos da información acerca del contenedor; Estado, imagen,tamaño,nombre,fecha creación, etc. Tiene multitud de opciones:
  + # Mostrar los contenedores que están en ejecución
  + > docker ps
  + # Mostrar todos los contenedores, estén parados o en ejecución (-a o --all)
  + > docker ps -a
  + # Añadir la información del tamaño del contenedor a la información por defecto (-s o --size)
  + > docker ps -a -s
  + # Mostrar información del último contenedor que se ha creado (-l o --latest). Da igual el estado
  + > docker ps -l
  + # Filtar los contenedores de acuerdo a algún criterio usando la opción (-f o --filter)
  + # Filtrado por nombre
  + > docker ps --filter name=servidor\_web
  + # Filtrado por puerto. Contenedores que hacen público el puerto 8080
  + > docker ps --filter publish=8080
* **docker inspect:** Nos da información acerca del contenedor de forma detallada, y se puede seleccionar el contenedor por id o por nombre. También tiene ciertas variaciones como pueden ser:
* **docker inspect --format 'La ip es {{.NetworkSettings.Networks.bridge.IPAddress}}' jenkins:** Nos devuelve la ip.
* **docker inspect --format 'Las redirecciones de puertos son {{.NetworkSettings.Ports}}' jenkins:** Nos devuelve las direcciones de los puertos.
* **docker logs:** Nos da información acerca del contenedor.
* **docker stop** para detener el contenedor, ya sea por nombre o por ID.
  + **docker stop -t 10:** Para detenerlo en un tiempo de 10 sec.
* **docker rm** para borrar el contenedor, ya sea por nombre o por ID.
  + **docker rm -f :** Para detenerlo aunque se encuentre en ejecucción
* **docker start** iniciar un contenedor que estaba parado previamente, ya sea por nombre o por ID.
  + **docker start -i :** Para arrancarlo pero crear una entrada estándar para poder interactuar con él
* **docker restart** para reiniciar un contenedor que previamente ya estaba en ejecución.
  + **docker restart -t 10:** Para reiniciarlo pero esperando 10 sec.

CREACIÓN DE VOLÚMENES

* **docker volumen create:** Crear volúmenes. Tiene varias opciones:
  + **--driver o -d** para especificar el driver elegido para el volumen. Al ser Linux en mi caso ese driver local es overlay2 pero existen otras posibilidades como aufs, btrfs, zfs, devicemapper o vfs.
  + **--label** para especificar los metadatos del volumen mediante parejas clave-valor.
  + **--opt o -o** para especificar opciones relativas al driver elegido.
  + **--name** para especificar un nombre para el volumen.

ELIMINACIÓN DE VOLUMENES

* **docker volumen rm:** para eliminar un volumen en concreto.
* **docker volumen prune** para eliminar los volúmenes que no están siendo usados por ningún contenedor.
* **docker volume rm -f nombre\_volumen:** Forzar la eliminación.
* **docker volume prune -f:** Borrar todos los volúmenes que no tengan contenedores asociados sin pedir confirmación.
* **docker volume prune --filter label=valor:** Borrar todos los volúmenes sin usar que contengan cierto valor de etiqueta

OBTENCIÓN DE INFORMACIÓN DE LOS VOLÚMENES

* **docker volume ls:** nos proporciona una lista de los volúmenes creados y algo de información adicional.
* **docker volume inspect:** nos dará una información mucho más detallada de el volumen que hayamos elegido.
* **--volume o -v.** Este flag lo utilizaremos para establecer bind mounts.
* **--mount.** Este flag nos servirá para establecer bind mounts y para usar volúmenes previamente definidos

IMÁGENES:

* **docker pull mysql:8.0.22:** Es el nombre de la imagen 8.0.22 es la versión o TAG
* **docker run -it -d --name mysql8 -p 3306:3306 mysql:8.0.22:** se descargará la imagen y posteriormente empezará a ejecutarse el contenedor si todos los parámetros están bien.
* **docker images:** Listas imágenes descargas, y nos muestra información como; Repositorios,Tag,Imagen id,Creación,Tamaño.
* **docker pull -a php o docker pull --all-tags php:** permite bajar todas las versiones de una imagen de una sola vez.
* **docker pull -q httpd o docker pull --quiet:** No muestro la información de las capas al descargarse

BORRADO DE IMÁGENES:

* **docker rmi mysql:8.0.22:** Borrado de la imagen mysql:8.0.22
* **docker rmi dd7265748b5d:** Borrado de una imagen usando su IMAGE ID
* **docker rmi mysql:8.0.22 mysql:5.7:** Borrado de dos imágenes (o varias) a la vez. Puedes usar nombre e IMAGE ID
* docker rmi -f httpd: Borra la imagen httpd (Apache latest) aunque hubiera contenedores que estuvieran usando esa imagen, es decir, fuerza el borrado.
* **docker image prune:**  que tiene tres opciones básicas:
  + **-a o --all** para borrar todas las imágenes que no están siendo usadas por contenedores
  + **--filter** para especificar ciertos filtros a las imágenes.
  + **-f o --force** para que no nos solicite confirmación.
* **docker image prune -a:** Borrar todas las imágenes sin usar.
* **docker image prune --filter until="240h":** Borrado de la imágenes creadas hace más de una semana 10 días
* **docker inspect --format '{{.Architecture}} es la arquitectura y el SO es {{.Os}}' mysql:8.0.22:** Mostrar la arquitectura y el sistema
* **docker inspect --format '{{.Config.ExposedPorts}}' mysql:8.0.22:** Mostrar la lista de puertos expuestos
* **docker image build** para construir una imagen desde un fichero Dockerfile (se verá en el apartado 6).
* **docker image history** para que se nos muestre por pantalla la evolución de esa imagen.
* **docker image save / docker image load (o docker save / docker load)** para guardar imágenes en fichero y cargarlas desde fichero (se verá en el apartado 6).
* **docker image tag ( docker tag)** para añadir TAGs (versiones) a las distintas imágenes.