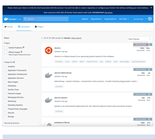
Ejemplo de imagenes y contenedores.

Este PDF nos muestra todos crear en nuestra máquina local un servidor de ubuntu con mysql usando la tecnología de contenedores.

El primer paso es ingresar a Docker hub para ver si existen imágenes que se pueden reutilizar para el montaje del contenedor, en este caso el procedimiento a realizar será montar un contenedor con una imagen del sistema operativo ubuntu, luego ejecutaremos este contenedor para instalar sobre el contenedor mysql y realizaremos unas pruebas. Desde la página de docker hub podremos usar el buscador para listar todas las imágenes que se han compartido de forma pública y que podrían reutilizarse para el montaje de nuestros propios contenedores.



Como se puede notar en las imágenes, en el proceso de búsqueda pueden aparecer imágenes compartidas de forma oficial por los fabricantes originales.

Al ingresar al detalle de cada una de las imágenes se puede observar las diferentes versiones disponibles y la forma en la que se pueden obtener desde la plataforma Docker instalada en mi máquina local.

También se puede realizar el proceso de búsqueda directamente desde una terminal del equipo donde está instalado docker con la ejecución del comando **docker search nombre**, donde **nombre** corresponde al mismo texto de búsqueda usado en Docker hub, en este caso ubuntu.

Para descargar la imagen utilizamos el comando **docker pull nombre** en la máquina host, donde **nombre** corresponde al nombre de la imagen a descargar, en este ejemplo ubuntu. Recuerde que estos procesos requieren ser ejecutados con un usuario administrador en la máquina host.

Las imágenes pueden tener disponibles varias versiones compartidas, a esto se le conoce con el nombre de tag, sino se especifica un tag el sistema asume que se desea descargar la última versión disponible de la imagen.

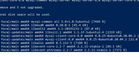
Para asegurarse de que la imagen realmente está disponible en nuestro sistema local puede ejecutar el comando **docker images**

Para crear un contenedor a partir de la imagen descargada se utiliza el comando **docker run -ti nombre**, donde el parámetro **i** indica una ejecución iterativa, es decir se puede modificar el contenedor generado, el comando **t** indica terminal y permite ingresar al contenedor una vez este se ejecuta, y nombre corresponde al **nombre** de la imagen en el equipo local, en este caso ubuntu.

Tenga en cuenta que bajo el comando anterior una vez se crea el contenedor se ingresa a él por lo que deberá ejecutar el comando **exit** para salir del contenedor

Para listar los contenedores disponibles en la máquina se ejecuta el comando **docker ps -a**, nota que al listar los contenedores se despliega el estado actual del mismo y el identificador asignado a cada contenedor, lo cual es muy importante para varios de los comandos sobre los contenedores.

Para ejecutar un contenedor detenido, se debe utilizar el comando **docker start id_contenedor**, donde **id_contenedor** corresponde al identificador de un contenedor en mi maquina local, también se pueden utilizar las iniciales del identificador si con este se puede diferenciar claramente entre los contenedores existentes.



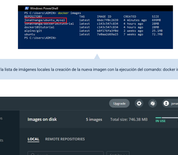
Para ingresar nuevamente a un contenedor que se está ejecutando, se utiliza el comando **docker attach id_contenedor**, donde el **id_contenedor** corresponde a un identificador válido de un contenedor en nuestra máquina local.

Una vez dentro del contenedor en ejecución, podemos interactuar sobre el como si fuera una máquina virtual ya que en este caso el contenedor está ejecutando el sistema operativo ubuntu. Por defecto se ingresa como usuario root por lo que se pueden realizar las modificaciones requeridas en el sistema operativo instalado en el contenedor.

En primer lugar actualizaremos la lista de paquetes disponibles y versiones del sistema operativo con el comando **apt update**

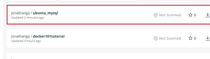
A continuación, instalamos MySQL en el contenedor al que ya ingresamos por medio del comando: **apt install mysql-server**

Luego de la instalación se va a ejecutar el servicio de MySQL para realizar pruebas. para iniciar el servicio se ejecuta el comando **service mysql start**



Para asumentar la seguridad del servicio de MySQL en el contenedor se ejecuta el siguiente script:
mysql_secure_installation, el cual realiza algunas configuraciones de seguridad a partir de un conjunto de preguntas que se responden como si o no para configurar el nivel de seguridad del password de root, asignación de password de root, eliminar usuarios anónimos, conexiones remotas, eliminación de bases de datos de prueba y recargar privilegios de usuarios.

Con esto finaliza la instalación de un servidor mysql sobre el contenedor, si desea verificar el estado del servicio puede ejecutar el comando **service mysql status**. Si desea ingresar al servidor para hacer algún tipo de acción adicional puede ejecutar **mysql** e ingresará automáticamente con el usuario root.



Teniendo en cuenta los pasos anteriores se puede instalar todos los componentes adicionales de trabajo que necesitemos en el contenedor, ahora como puede ser posible que queramos que ese contenedor quede disponible para compartir y poder generar nuevas imágenes del mismo o simplemente hacer un backup se puede realizar una derivación del contenedor con el siguiente comando:

docker commit -m "nombre del cambio" **-a** "autor" **id_contenedor usuario_dockerHub/nombreImagen**

Donde **id_contenedor** corresponde al identificador del contenedor a crear, usuario_dockerHub es opcional y si se usa debe reflejar un **usuario registrado en Docker hub y nombreImagen** el nombre con el que se crear la imagen a partir del contenedor.

Se puede verificar en la lista de imágenes locales la creación de la nueva imagen con la ejecución del comando: docker images.

Con esta nueva imagen podemos crear nuevos contenedores ejecutando la misma dinámica vista anteriormente con el comando **docker run -ti nombreImagen**

Si desea compartir la imagen creada en un repositorio personal de Docker Hub debemos identificarnos con su usuario y contraseña, usando el siguiente comando: **docker login -u nombreUsuario**, donde **nombreusuario** corresponde a un nombre de usuario registrado previamente en la plataforma docker Hub.

También se puede usar en el caso de windows la aplicación Docker Desktop para explorar las imágenes creadas localmente y subirlas a Docker Hub. El proceso de carga dependerá del tamaño de la imagen y la velocidad de subida de internet de la que se disponga.