Docker

**Tabla de Contenido**

[1. Instalación 2](#_Toc83217246)

[2. Marco teórico 2](#_Toc83217247)

[3. Comandos básicos 3](#_Toc83217248)

# Instalación

* Para ejecutar en Windows es necesario activar la virtualización de desde la BIOS.
* La instalación en UBUNTU requiere el siguiente [[PROCESO](https://docs.docker.com/engine/install/ubuntu/)]

# Marco teórico

* Imágenes: Serie de instrucciones o tecnologías que permiten construir un contenedor. Es similar a un témplate único (librerías, herramientas, dependencias y códigos).
* Container: Es el instanciamiento de una imagen de Docker. es un entorno de ejecución virtualizado que se utiliza en el desarrollo de aplicaciones. Se utiliza para crear, ejecutar e implementar aplicaciones que están aisladas del hardware y entre ellos.
* Volumes: Un volumen es un directorio o un fichero en el docker engine que se monta directamente en el contenedor. Podemos montar varios volúmenes en un contenedor y en varios contenedores podemos montar un mismo volumen.
* Tags: Versiones de una misma imagen
* Network: Redes existentes entre los contenedores con su propia Gateway.

[[REFERENCIA1](https://searchitoperations.techtarget.com/definition/Docker-image)]

# Comandos

|  |  |
| --- | --- |
| **Comandos** | **Descripción** |
| docker pull <IMAGE>  docker image pull <IMAGE> | Descarga una imagen |
| docker image rm <IMAGE>  docker image rm <IMAGE> -f  docker image rm <IMAGE> --force | Eliminar una imagen. |
| docker run <IMAGE> | Descarga y ejecuta una imagen |
| docker run -it <IMAGE> | Descarga y ejecuta una imagen de docker de manera iterativa |
| docker run <IMAGE> <ARGS> | Descargar y ejecutar una imagen junto a args |
| docker run -p <PUERTO>:<PUERTOINTERNO> -d <IMAGEPUERTO> | Descargar y ejecutar una imagen que contenga que utilice puertos. -p Permite conectar puertos internos con externos. -d Ejecuta el proceso en segundo plano |
| docker run –-name <NOMBRE> | Correr un contenedor con un nombre especifico |
| docker run –e <VARIABLE\_ENTORNO> | Correr un contenedor con una o varias variable de entorno. |
| Docker run -v <DIRECCION>:<VOLUME>:ro | Agregar un volumes a una ejecución de un contenedor. ro: ReadOnly |
| Docker run –volumes-from  <OTROCONTENEDOR> | Se le indica que utilice los mismos volúmenes que otro contenedor existente |
| docker run –-link <OTROCONTENEDOR>:<alias> | Agregar una conexión directa con u host de otro contenedor |
| docker run –-netwok <RED> | Agregar la red en la que se montara el contenedor. |
| docker images | Visualizar las imágenes instaladas |
| docker search <IMAGE> | Buscar una imagen de docker en DockerHub |
| docker ps | Imágenes instaladas (Contenedores) |
| docker ps -a | Listado o Log de los contenedores |
| docker ps -q | Listado o Log de los contenedores (Solo ids) |
| docker ps –-format=<FORMATO> | Se puede establecer el formato en el que se quiere que se muestren los contenedores  [[FORMATO1](https://docs.docker.com/config/formatting/)][[FORMATO2](https://docs.docker.com/engine/reference/commandline/ps/)] |
| docker rm <ID|NOMBRE CONTENEDOR> | Eliminar un contenedor |
| $(docker ps -aq) | Solicitar subcomandos |
| docker start <ID|NOMBRE CONTENEDOR> | Inicializar un contenedor “dormido” |
| docker stop <ID|NOMBRE CONTENEDOR> | Detener la ejecución de un contenedor |
| docker exec -it <ID|NOMBRE> bash | Abrir una consola para ingresar al contenedor |
| docker build -t <NOMBRE> <DIRECCION> | Generar una imagen a partir de un Dockerfile |
| docker build -t <USUARIO DOCKERHUB>/<DIRECCION> <DIRECCION> | Generar una imagen a partir de un Dockerfile y subirlo a DockerHub |
| docker build -t <NOMBRE> <DIRECCION> --build-arg <VARIABLE> | Permite establecer argumentos para el build. |
| docker commit -t <NOMBRE> <CONTAINER> | Generar una imagine a partir de un Container. |
| docker login | Hacer login en dockerhub |
| docker inspect <ID|NOMBRE> | grep IPAdd | Observar la IP donde se encuentra el contenedor |
| docker network –-help | Mirar todas las funciones que se tienen para manejar una red |
| docker volume –-help | Mirar todas las funciones que se tienen para manejar un volumen |

# DockerFile

**FROM <IMAGEN>:<TAG>**

//Imagen que utiliza de base

**WORKDIR <DIRECCION INTERNA>**

//Directorio donde operara COPY O ADD

**COPY <VOLUME> <DIRECCION INTERNA | .>**

//Copia el contenido local al de la imagen

**RUN <COMANDO>**

// Comandos que se ejecutaran previamente

**CMD [String[]]**

//Comando que se ejecutara cuando se genera un contenedor

# Preguntas

¿Cuál es la ventaja principal de utilizar una red bridge propia en lugar de la que viene por defecto?