

Comandos importantes Docker

- Comando de ayuda:

```
docker --help  
docker run --help  
docker network --help  
docker volume --help  
docker ... --help
```

- Descargar imagen del repositorio de Docker Hub:

```
docker pull NOMBRE_IMAGEN
```

- Ver imágenes docker descargadas en el equipo:

```
docker images
```

- Ver las redes creadas en docker:

```
docker network ls
```

- Crear una red:

```
docker network create NOMBRE_RED_NUEVA
```

- Inspeccionar toda la información de

```
docker inspect NOMBRE_CONTENEDOR  
docker inspect NOMBRE_IMAGEN  
docker inspect NOMBRE_RED  
docker inspect ...
```

- Acceder al interior de un contenedor para trabajar en la terminal de él:

```
docker exec -it NOMBRE_CONTENEDOR bash
```

- Copiar un archivo del HOST al contenedor:

```
docker cp RUTA_ARCHIVO NOMBRE_CONTENEDOR:RUTA_DE_COPIADO
```

- Copiar un archivo del contenedor al host:

```
docker cp NOMBRE_CONTENEDOR:RUTA_ARCHIVO RUTA_DE_COPIADO
```

- Salir de un contenedor:

```
exit
```

- Parar/Encender un contenedor:

docker stop/start NOMBRE_CONTENEDOR

- Borrar un contenedor:

docker rm NOMBRE_CONTENEDOR

- Borrar una imagen:

docker rmi NOMBRE_IMAGEN:VERSIÓN

- Crear contenedores:

Opción -p: mapeo de puertos (solo si lo requiere la imagen de la que se va a crear un contenedor). Lo usamos mayormente cuando tenemos que acceder a ese contenedor de manera gráfica. **IMPORTANTE: No puede repetirse el mismo PUERTO HOST más de una vez. Ejemplo de uso: -p PUERTO_HOST:PUERTO_CONTENEDOR.**

Opción --name: para asignar un nombre al contenedor que vamos a crear. Si no se establece el sistema crea un nombre por defecto. **IMPORTANTE: No puede crearse más de un contenedor con el mismo nombre.**

Opción -d: crear contenedor en modo background. Con esto conseguimos que el contenedor creado no nos muestre en pantalla información sobre lo que está ejecutando y podemos seguir trabajando en el terminal sin problema.

Opción -it: se asigna esta opción para que ciertos contenedores, por ejemplo, Ubuntu o Centos, no se cierren por defecto al crear el contenedor. Con esto nos aseguramos que el contenedor esté activo a la espera de que nosotros accedemos al mismo o le indiquemos instrucciones.

Opción --network: asignamos una red creada previamente al contenedor que vamos a crear. Esto se utiliza cuando queremos que los contenedores se puedan ver entre sí para la transmisión de información. Ejemplo: Interfaz gráfica de Wordpress con un contenedor de base de datos.

Opción -e: se utiliza para establecer variables de entorno dentro del contenedor. Las variables de entorno son valores que pueden ser utilizados por los procesos en ejecución dentro del contenedor para acceder a información específica del entorno, como configuraciones, rutas de archivos o credenciales.

Opción -v: mapear volúmenes o carpetas entre la imagen y el contenedor.

Ejemplo crear contenedor:

```
docker run --name NOMBRECONTENEDOR  
  --network NOMBRED  
  -it  
  -d  
  -p PUERTO_HOST:PUERTO_CONTENEDOR  
  -e VARIABLESDEENTORNO  
  -v RUTA-CARPETA ó VOLUMEN  
  NOMBRE_DE_LA_IMAGEN:VERSIÓN
```