



Docker run

```
docker run -d --name=cont1 ubuntu tail -f /dev/null
```

- Crea un contenedor con la imagen “ubuntu” (al no especificar, toma versión “latest”), le establece un nombre “cont1” y lanza el comando “tail -f /dev/null”, que es un comando cuya ejecución no acaba nunca.

```
docker run -d -p 1200:80 nginx
```

- Crea un contenedor con la versión “latest” de la imagen “nginx” y lo lanza en “background”, exponiendo el puerto 80 del contenedor en el puerto 1200 de la máquina anfitrión.

```
docker run -it -e MENSAJE=HOLA ubuntu:24.04 bash
```

- Crea un contenedor con la imagen “ubuntu”, versión “24.04” y establece la variable de entorno “MENSAJE”.

Docker ps

```
docker ps
```

- Muestra información de los contenedores en ejecución.

```
docker ps -a
```

- Muestra información de todos los contenedores, tanto parados como en ejecución.

Docker start/stop/restart

```
docker start micontenedor
```

- Arranca el contenedor con nombre “mi contenedor”.

```
docker start -ai micontenedor
```

- Arranca el contenedor con nombre “mi contenedor”, enlazando el comando ejecutado al arranque a la entrada y salida estándar de la terminal del anfitrión.

Docker exec

```
docker exec -it idcont bash
```

- Lanza en el contenedor “idcont” (que debe estar arrancado) el comando “bash”, enlazando la ejecución de forma interactiva a la entrada y salida estándar del anfitrión en la terminal.

```
docker exec -it -e FICHERO=prueba cont bash
```

- Lanza en el contenedor “cont” (que debe estar arrancado) el comando “bash”, estableciendo la variable de entorno “FICHERO” y enlazando la ejecución de forma interactiva a la entrada y salida estándar del anfitrión.



Unidad 03 - Autor: Sergi García Barea

```
docker exec -d cont touch /tmp/prueba
```

- Lanza en el contenedor “cont” (que debe estar arrancado) el comando “touch /tmp/prueba”. Este comando se ejecuta en segundo plano, generando el fichero “/tmp/prueba”.



Docker attach

```
docker attach idcontainer
```

- Enlaza nuestra terminal la entrada/salida de nuestra al proceso en segundo plano del contenedor “idcontainer”.



Docker logs

```
docker logs -n 10 idcontainer
```

- Muestra las 10 últimas líneas de la salida estándar producida por el proceso en ejecución en el contenedor.



Docker cp

```
docker cp idcontainer:/tmp/prueba ./
```

- Copia el fichero “/tmp/prueba” del contenedor “idcontainer” al directorio actual del anfitrión.

```
docker cp ./miFichero idcontainer:/tmp
```

- Copia el fichero “miFichero” del directorio actual del anfitrión a la carpeta “/tmp” del contenedor.