

Handbook de Docker - Guía Rápida

Martín Rodríguez

1 de septiembre de 2025

Índice

1. Introducción	2
2. Conceptos básicos	2
3. Comandos principales	2
3.1. Imágenes	2
3.2. Contenedores	2
3.3. Volúmenes y persistencia	2
4. Dockerfile	2
5. Redes en Docker	3
6. Docker Compose	3
7. Tips útiles	3
8. Conclusión	4

1. Introducción

Docker es una plataforma que permite empaquetar, distribuir y ejecutar aplicaciones dentro de **contenedores**. Los contenedores son entornos ligeros y portables que incluyen todo lo necesario para ejecutar una aplicación (código, dependencias, librerías, configuraciones).

2. Conceptos básicos

- **Imagen:** plantilla inmutable usada para crear contenedores.
- **Contenedor:** instancia en ejecución de una imagen.
- **Dockerfile:** archivo con instrucciones para construir una imagen.
- **Docker Hub:** repositorio público de imágenes.
- **Volumen:** mecanismo de persistencia de datos para contenedores.

3. Comandos principales

3.1. Imágenes

```
1 docker pull ubuntu          # Descargar una imagen
2 docker images               # Listar imágenes locales
3 docker rmi nombre:tag      # Eliminar una imagen
```

3.2. Contenedores

```
1 docker run hello-world      # Ejecutar un contenedor simple
2 docker run -it ubuntu bash  # Contenedor interactivo
3 docker ps                   # Listar contenedores en ejecución
4 docker ps -a                # Listar todos los contenedores
5 docker stop <id>            # Detener contenedor
6 docker rm <id>              # Eliminar contenedor
```

3.3. Volúmenes y persistencia

```
1 docker volume create datos
2 docker run -v datos:/app/data ubuntu
3 docker volume ls
```

4. Dockerfile

Ejemplo básico de Dockerfile:

```
1 # Imagen base
2 FROM php:8.2-cli
3
4 # Directorio de trabajo
5 WORKDIR /app
6
7 # Copiar archivos
8 COPY . /app
9
10 # Comando por defecto
11 CMD ["php", "index.php"]
```

Construcción y ejecución:

```
1 docker build -t miapp .
2 docker run miapp
```

5. Redes en Docker

```
1 docker network ls          # Listar redes
2 docker network create mi-red # Crear red
3 docker run --network mi-red ... # Ejecutar en una red
```

6. Docker Compose

Permite definir y administrar múltiples contenedores. Ejemplo con PHP + MySQL:

```
1 version: "3.8"
2 services:
3   web:
4     image: php:8.2-apache
5     ports:
6       - "8080:80"
7     volumes:
8       - ./var/www/html
9   db:
10    image: mysql:8.0
11    environment:
12      MYSQL_ROOT_PASSWORD: root
13      MYSQL_DATABASE: ejemplo
```

Ejecución:

```
1 docker-compose up -d    # Levantar en segundo plano
2 docker-compose down     # Detener y eliminar contenedores
```

7. Tips útiles

- Usa `docker exec -it <id>bash` para entrar a un contenedor en ejecución.

- Usa `docker logs <id>` para ver los logs.
- Usa `docker system prune` para limpiar recursos no usados.

8. Conclusión

Docker simplifica la creación y despliegue de aplicaciones, garantizando que funcionen igual en cualquier entorno. Este handbook resume los comandos y conceptos más importantes para comenzar a trabajar con contenedores.