

Instalación de Docker en Ubuntu 22.04 LTS (WSL 2)

Repositorio Completo

Este documento explica cómo instalar **Docker Engine** y configurar **Docker en WSL 2** usando **Ubuntu 22.04 LTS**, tal como lo hemos instalado previamente.

La instalación sigue las recomendaciones oficiales de Docker para entornos basados en Linux bajo WSL.

1. Requisitos previos

Antes de comenzar, asegúrate de que:

- Estás usando **WSL 2**

```
wsl --status
```

- Tu distribución activa es **Ubuntu 22.04 LTS**

```
lsb_release -a
```

- Tu sistema está actualizado:

```
sudo apt update && sudo apt upgrade -y
```

[!WARNING] Si no cumples los requisitos ve a esta guía: [instalación WSL con Ubuntu 22.04 LTS](#)

2. Desinstalar cualquier versión antigua de Docker

Si tienes versiones previas, elimínalas:

```
sudo apt remove -y docker docker-engine docker.io containerd runc
```

Esto no borra datos ni configuraciones, solo desinstala paquetes antiguos.

3. Instalar dependencias necesarias

Docker requiere varios paquetes del sistema:

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

4. Añadir la clave GPG oficial de Docker

```
sudo mkdir -p /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

5. Añadir el repositorio oficial de Docker

```
echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy  
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Actualiza repositorios:

```
sudo apt update
```

6. Instalar Docker Engine, CLI y containerd

```
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

Esto instala:

- **docker** (cliente)
- **dockerd** (demonio)
- **docker compose** (nuevo plugin)
- **containerd** como runtime

7. Habilitar Docker para ejecutarse sin sudo

Agrega tu usuario al grupo **docker**:

```
sudo usermod -aG docker $USER
```

Aplica los cambios cerrando y abriendo de nuevo WSL, o ejecutando:

```
newgrp docker
```

8. Probar que Docker funciona

```
docker run hello-world
```

Si funciona, verás un mensaje de éxito indicando que Docker está correctamente instalado.

9. Habilitar systemd en WSL (para que Docker funcione siempre)

Docker requiere **systemd** para ejecutar servicios en segundo plano.

Edita el archivo de configuración de WSL:

```
sudo nano /etc/wsl.conf
```

Añade:

```
[boot]
systemd=true
```

Guarda (CTRL+O) y cierra (CTRL+X).

Luego reinicia WSL desde Windows:

```
wsl --shutdown
```

Y vuelve a abrir Ubuntu.

10. Comprobar que el servicio Docker está activo

```
systemctl status docker
```

Deberías ver:

```
Active: active (running)
```

12. (Opcional) Probar un stack básico de Docker Compose

Crea un archivo `docker-compose.yml`:

```
services:
  web:
    image: nginx
    ports:
      - "8080:80"
```

Ejecuta:

```
docker compose up -d
```

Abre en navegador:

```
http://localhost:8080
```