

Introducción a Docker y Kubernetes

UD 02. Instalación de Docker



Autor: Sergi García Barea

Actualizado Marzo 2025

Licencia



Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

ÍNDICE DE CONTENIDO

1. Introducción	3
2. Instalación de Docker en sistemas Linux (Ubuntu)	3
2.1 Instalación desde el repositorio oficial de Ubuntu (No recomendado)	3
2.2 Instalación desde el repositorio de Docker-CE (Recomendado)	3
2.2.1 Paso 1: Eliminando versiones antiguas de Docker Engine	3
2.2.2 Paso 2: Incluyendo el repositorio de Docker CE	3
2.2.3 Paso 3: Instalando Docker Engine CE	4
2.3 Post instalación	5
2.3.1 Permitir administrar Docker con usuarios sin privilegios	5
2.3.2 Activar/desactivar arranque al inicio	6
2.4 Desinstalando Docker en Ubuntu	7
3. Instalación de Docker en sistemas Windows	7
4. Instalación de Docker en sistemas MacOS	8
5. Playgrounds de Docker	9
6. Conclusión	9
7. Bibliografía	9

UD02. INSTALACIÓN DE DOCKER

1. INTRODUCCIÓN

En esta unidad explicaremos diversos itinerarios para la instalación de Docker, realizando nuestras recomendaciones. Tras ello, describiremos opciones de post instalación y comprobaremos que todo funciona correctamente. Aunque el curso debería poder realizarse en sistemas Windows y MacOS, **recomendamos siempre que sea posible realizar el curso (y en general, usar Docker) en sistemas Linux**, ya que su implementación a día de hoy es más robusta y puede evitarnos muchos quebraderos de cabeza.

2. INSTALACIÓN DE DOCKER EN SISTEMAS LINUX (UBUNTU)

En este apartado vamos a hablar de la instalación de Docker Engine CE (Community Edition) en sistemas Linux en distribuciones basadas en **Ubuntu** (Ubuntu, KUbuntu, Linux Mint, etc.).

El procedimiento para instalar Docker Engine CE en otras distribuciones es similar y aquí dejo algunos enlaces con instrucciones para instalarlo en algunas de las más populares:

- **CentOS:** <https://docs.docker.com/engine/install/centos/>
- **Debian:** <https://docs.docker.com/engine/install/debian/>
- **Fedora:** <https://docs.docker.com/engine/install/fedora/>

Asimismo, también es posible la realización de una instalación desde los binarios, siguiendo las instrucciones de <https://docs.docker.com/engine/install/binaries/>

2.1 Instalación desde el repositorio oficial de Ubuntu (No recomendado)

Es posible instalar Docker Engine desde el repositorio oficial de Ubuntu, pero no está recomendado, ya que instala versiones antiguas. **En este curso, daremos por supuesto que no se utilizaran estas versiones e instalaremos Docker CE desde la fuente oficial.**

2.2 Instalación desde el repositorio de Docker-CE (Recomendado)

A continuación detallaremos los pasos para instalar Docker Engine CE en Ubuntu desde el repositorio oficial de Docker CE. Seguiremos los pasos detallados en <https://docs.docker.com/engine/install/ubuntu/>

2.2.1 Paso 1: Eliminando versiones antiguas de Docker Engine

En primer lugar, deberemos eliminar otras versiones de Docker, por si estuvieran instaladas (por ejemplo, una versión del repositorio oficial de Ubuntu) y pudieran provocar conflictos.

Para eliminar las versiones antiguas puede bastarnos con la orden:

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2  
podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

2.2.2 Paso 2: Incluyendo el repositorio de Docker CE

Para poner en marcha el repositorio, lo primero que haremos será actualizar nuestro índice de paquetes y tras ello, instalar los paquetes necesarios (si no lo estaban ya) para que se puedan utilizar repositorios con HTTPS.

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

Una vez realizado este paso, descargamos la clave GPG del repositorio de Docker CE y la incluiremos. Podemos hacer todo con las siguientes líneas:

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Ahora, solo nos queda añadir el repositorio de Docker CE como fuente para instalación de paquetes. **MUCHO OJO en este paso en distribuciones derivadas, como Linux Mint.** El motivo es el siguiente. Al configurar la fuente de paquetes indicamos la versión de Ubuntu. El comando que utilizamos para obtener la versión de Ubuntu es el siguiente:

```
lsb_release -cs
```

Este comando nos dirá qué distribución tenemos. Por ejemplo, si tenemos “Ubuntu Noble 24.04 (LTS)”, este comando imprimirá por pantalla “Noble”.

En algunas versiones derivadas de Ubuntu, como Linux Mint 22, aunque la distribución esté basada en Ubuntu 24.04 (Noble), no devolverá el texto “Noble”, sino otro diferente. Si estáis en este caso, deberéis introducir a mano la versión de Ubuntu en que se basa vuestra distribución (sustituyendo el comando de “*lsb_release -cs*” de la siguiente línea)

Aclarado esto, con el siguiente comando podéis añadir el repositorio:

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

O en el caso de que tengáis una distribución basada en Ubuntu con el problema comentado anteriormente, poner el nombre de la versión de Ubuntu a mano. En este ejemplo, si la base es Ubuntu 24.04 (Noble) quedaría de una forma similar a:

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

2.2.3 Paso 3: Instalando Docker Engine CE

Por último, ya con el repositorio oficial de Docker en nuestro sistema, solo nos queda actualizar el índice de paquetes e instalar la última versión de Docker Engine CE de la siguiente forma:

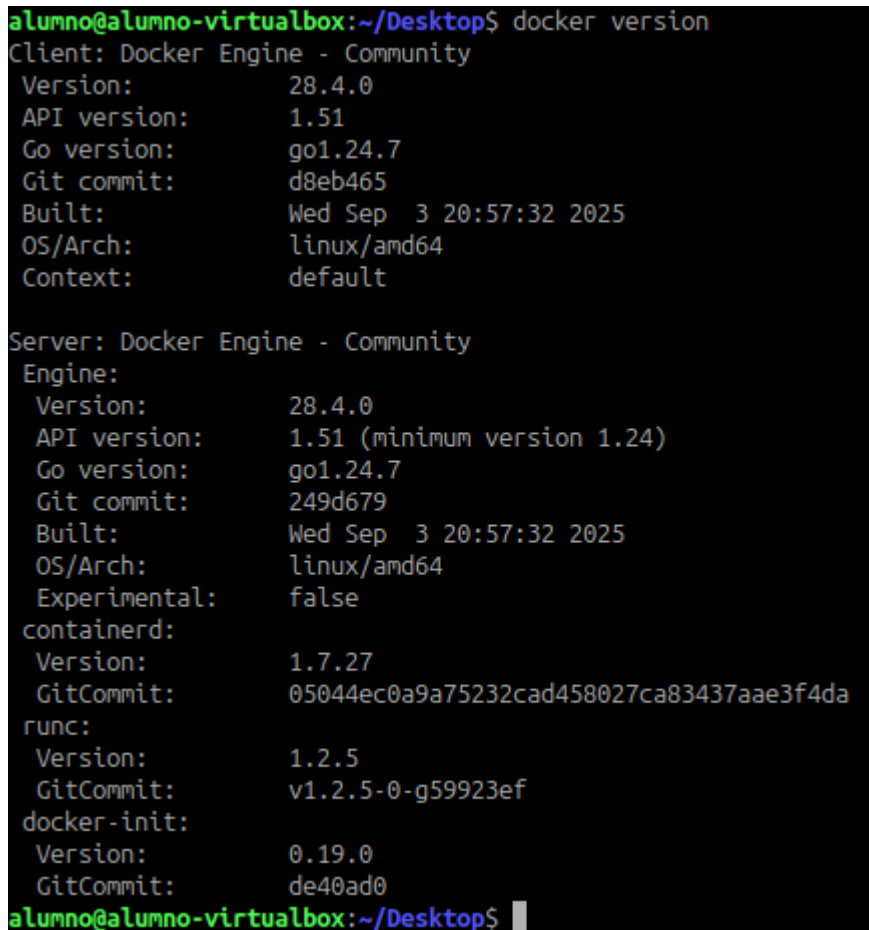
```
sudo apt-get update
```

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

Finalmente, comprobaremos que Docker Engine CE se ha instalado correctamente ejecutando:

```
sudo docker version
```

y obteniendo un resultado similar al siguiente:



```
alumno@alumno-virtualbox:~/Desktop$ docker version  
Client: Docker Engine - Community  
Version: 28.4.0  
API version: 1.51  
Go version: go1.24.7  
Git commit: d8eb465  
Built: Wed Sep 3 20:57:32 2025  
OS/Arch: linux/amd64  
Context: default  
  
Server: Docker Engine - Community  
Engine:  
Version: 28.4.0  
API version: 1.51 (minimum version 1.24)  
Go version: go1.24.7  
Git commit: 249d679  
Built: Wed Sep 3 20:57:32 2025  
OS/Arch: linux/amd64  
Experimental: false  
containerd:  
Version: 1.7.27  
GitCommit: 05044ec0a9a75232cad458027ca83437aae3f4da  
runc:  
Version: 1.2.5  
GitCommit: v1.2.5-0-g59923ef  
docker-init:  
Version: 0.19.0  
GitCommit: de40ad0  
alumno@alumno-virtualbox:~/Desktop$
```

Para más información sobre este comando podéis visitar

<https://docs.docker.com/engine/reference/commandline/version/>

2.3 Post instalación

En la documentación de Docker, nos proponen algunos pasos de post instalación. Los podéis consultar en <https://docs.docker.com/engine/install/linux-postinstall/>

En esta sección vamos a comentar dos de ellos: administrar Docker con usuarios sin privilegios (no root ni sudoers) y arrancar Docker desde al inicio.

2.3.1 Permitir administrar Docker con usuarios sin privilegios

Docker utiliza sockets Unix. Para la creación y reserva de un socket Unix, es necesario tener permisos de root, por lo cual Docker Engine necesita permisos de root para ejecutarse.

A veces, en algunos contextos, puede sernos útil que Docker se ejecute por usuarios sin permisos de root. Pongamos un contexto de un aula de ordenadores, que será utilizada por alumnos con los que queremos realizar actividades que utilizan Docker.

Podemos configurar el aula para que dichos alumnos puedan utilizar Docker sin necesidad de proporcionarles una cuenta con permisos de root.

Para ello, en primer lugar crearemos un grupo llamado “docker”.

```
sudo groupadd docker
```

Tras ello añadiremos a los usuarios que queremos que usen Docker sin permisos de root al grupo con un comando similar al siguiente, donde \$USER es el nombre de usuario:

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

Para probar que esto funcione, será necesario que el/los usuarios afectados cierren sesión completamente (si la tenían abierta) y la vuelvan a abrir, para que se reevalúe su pertenencia al grupo “docker”. Con esto podrán utilizar comandos Docker sin necesidad de permisos de root.

MUCHO OJO: si lanzaste comandos de Docker usando sudo con uno de estos usuarios antes de esta operación, es posible que al lanzar Docker te aparezca un mensaje similar a este:

```
WARNING: Error loading config file: /home/user/.docker/config.json -  
stat /home/user/.docker/config.json: permission denied
```

Para arreglar este error, las opciones son eliminar el directorio “.docker” con

```
sudo rm -rf ~/.docker/
```

o cambiar el propietario y permisos del directorio, usando

```
sudo chown "$USER":"$USER" /home/"$USER"/.docker -R
```

```
sudo chmod g+rwX "$HOME/.docker" -R
```

2.3.2 Activar/desactivar arranque al inicio

Para indicar que el servicio de Docker se inicie al arrancar la máquina, podemos indicarlo mediante los siguientes comandos:

```
sudo systemctl enable docker.service
```

```
sudo systemctl enable containerd.service
```

Si lo que queremos es deshabilitar este arranque automático, podemos usar:

```
sudo systemctl disable docker.service
```

```
sudo systemctl disable containerd.service
```

Para iniciar/parar/reiniciar los servicios manualmente, podemos usar:

```
sudo systemctl start/stop/restart docker.service
```

```
sudo systemctl start/stop/restart containerd.service
```

2.4 Desinstalando Docker en Ubuntu

Si en algún momento queremos desinstalar completamente Docker en Ubuntu, podemos usar el siguiente conjunto de comandos. El primer comando eliminará Docker Engine, pero no eliminará contenedores e imágenes presentes en el sistema.

```
sudo apt-get purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin docker-ce-rootless-extras
```

Si queremos eliminar estos datos, deberemos realizar el borrado manual de las mismas. Aquí el conjunto de comandos para el borrado manual:

```
sudo rm -rf /var/lib/docker
sudo rm -rf /var/lib/containerd
sudo rm /etc/apt/sources.list.d/docker.list
sudo rm /etc/apt/keyrings/docker.asc
```

3. INSTALACIÓN DE DOCKER EN SISTEMAS WINDOWS

En este apartado veremos cómo instalar Docker Desktop en sistemas Windows. Básicamente, para la instalación en sistemas Windows 10, 11 o Windows Server, seguiremos los pasos de esta guía

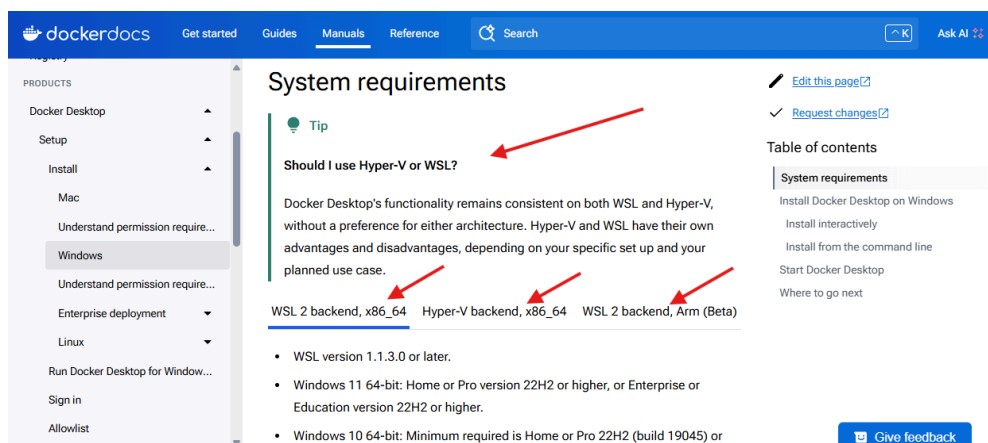
- Guía para instalar Docker en sistemas Windows:
 - <https://docs.docker.com/docker-for-windows/install/>
- Guía para instalar WSL2 en sistemas Windows:
 - <https://learn.microsoft.com/es-es/windows/wsl/install>

Antes de nada, indicar que he preparado un video de creación propia donde podéis ver como realizo una instalación completa de Docker Desktop y WSL 2 en un sistema Windows 11.

El video dura unos 13 minutos (es largo porque no tiene cortes) y lo tenéis disponible en <https://youtu.be/Od8LX5Gs1RE>

En la instalación, se distingue entre la posibilidad de simular Linux internamente vía Hyper-V o vía WSL2. Cabe destacar que en cualquiera de las dos opciones, se está virtualizando un sistema Linux, por lo cual el rendimiento es menor que si lanzaremos Docker desde un sistema Linux.

A la hora de seguir las instrucciones veremos un recuadro donde reflexiona si usar Hyper-V (no disponible en Windows 10 Home) o WSL2, seguido de 3 pestañas cada una con las instrucciones de instalación de su caso, tal como se ve en la siguiente imagen:



Para usuarios no experimentados, puede ser una opción más sencilla usar WSL2.

Aun así, por los problemas de rendimiento comentados anteriormente y por una mejor estabilidad en general, recomendamos encarecidamente durante el curso, **si es posible, usar Docker en sistemas Linux, donde Docker funciona de manera más robusta.**

Elijamos la opción que elijamos, se instalará el software “Docker Desktop” que nos permitirá utilizar Docker en sistemas Windows.

Aprovecho para recordar que he realizado un video de creación propia donde podéis ver como realizo una instalación completa de Docker Desktop y WSL 2 en un sistema Windows 11.

El video dura unos 13 minutos (es largo porque no tiene cortes) y lo tenéis disponible en <https://youtu.be/Od8LX5Gs1RE>

Finalmente, con **Docker Desktop instalado y lanzado**, comprobaremos que Docker Engine CE se ha instalado correctamente ejecutando:

```
docker version
```

obteniendo un resultado similar al siguiente:

```
Microsoft Windows [Versión 10.0.19042.804]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergi>docker version
Client: Docker Engine - Community
 Cloud integration: 1.0.7
 Version:           20.10.2
 API version:       1.41
 Go version:        go1.13.15
 Git commit:        2291f61
 Built:             Mon Dec 28 16:14:16 2020
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:           20.10.2
  API version:       1.41 (minimum version 1.12)
  Go version:        go1.13.15
  Git commit:        8891c58
  Built:             Mon Dec 28 16:15:28 2020
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:           1.4.3
  GitCommit:        269548fa27e0089a8b8278fc4fc781d7f65a939b
 runc:
  Version:           1.0.0-rc92
  GitCommit:        ff819c7e9184c13b7c2607fe6c30ae19403a7aff
 docker-init:
  Version:           0.19.0
  GitCommit:        de40ad0
```

Para más información sobre este comando podéis visitar

<https://docs.docker.com/engine/reference/commandline/version/>

4. INSTALACIÓN DE DOCKER EN SISTEMAS MacOS

Las instrucciones para la instalación de Docker Desktop en MacOS están descritas en <https://docs.docker.com/docker-for-mac/install/>.

Para realizar esta instalación, básicamente debe descargarse el paquete “.dmg” de <https://hub.docker.com/editions/community/docker-ce-desktop-mac/> y seguir las instrucciones de instalación en pantalla.

5. PLAYGROUNDS DE DOCKER

En Internet existen varios sitios que nos permiten utilizar un “playground” de distintas herramientas, para que juguemos con ellas online, sin necesidad de instalar ni configurar nada (y sin el riesgo de romper cosas).

El sitio web “Play with Docker” <https://labs.play-with-docker.com/> posee un “playground” de Docker donde se pueden realizar pruebas, pequeños despliegues, etc.

Os animo a utilizarlo si queréis hacer alguna prueba, trastear en un lugar donde no tenéis Docker instalado o para enseñar Docker en contextos donde no se pueda instalar de forma nativa.

6. CONCLUSIÓN

En esta unidad hemos visto los pasos básicos para instalar Docker en distintos sistemas operativos. **No obstante, continuamos con nuestra recomendación de que si es posible, para minimizar problemas, utilicemos Docker en sistemas Linux.**

7. BIBLIOGRAFÍA

[1] Docker Docs <https://docs.docker.com/>