

Introducción a Docker

UD 02. Instalación de Docker



Fons Social Europeu

L'FSE inverteix en el teu futur

Autor: Sergi García Barea

Actualizado Marzo 2022

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

| | |
|--|-----------|
| Introducción | 3 |
| Instalación de Docker en sistemas Linux (ubuntu) | 3 |
| Instalación desde el repositorio oficial de Ubuntu (No recomendado) | 3 |
| Instalación desde el repositorio de Docker-CE (Recomendado) | 3 |
| Paso 1: Eliminando versiones antiguas de Docker Engine | 3 |
| Paso 2: Incluyendo el repositorio de Docker CE | 4 |
| Paso 3: Instalando Docker Engine CE | 5 |
| Post instalación | 5 |
| Permitir administrar Docker con usuarios sin privilegios | 6 |
| Activar/desactivar arranque al inicio | 6 |
| Desinstalando Docker en Ubuntu | 7 |
| Instalación de Docker en sistemas Windows | 7 |
| Pasos previos Windows 10 Pro y Windows Server: activando Hyper-v | 7 |
| Pasos previos Windows 10 Home: Instalando WSL2 | 8 |
| Instalación de Docker Desktop | 8 |
| Resolviendo problemas en la instalación de Docker Desktop | 9 |
| Instalación de Docker en sistemas MacOS | 10 |
| Playgrounds de Docker | 10 |
| Conclusión | 10 |
| Bibliografía | 10 |

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 utilizarán 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. Las versiones de Ubuntu soportadas (todas de 64 bits) son:

- Ubuntu Groovy 20.10.
- Ubuntu Focal 20.04 (LTS).
- Ubuntu Bionic 18.04 (LTS).
- Ubuntu Xenial 16.04 (LTS).

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:

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

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 apt-transport-https ca-certificates curl  
gnupg-agent software-properties-common
```

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

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg  
--dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

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 Bionic 18.04 (LTS)”, este comando imprimirá por pantalla “bionic”.

En algunas versiones derivadas de Ubuntu, como Linux Mint, aunque la distribución esté basada en Ubuntu Bionic, no devolverá el texto “bionic”, sino otro diferente. Si estáis en este caso, deberéis introducir a mano la versión de Ubuntu en que se basa vuestra distribución.

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

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

O en el caso que tengáis una distribución basada en Ubuntu con el problema comentado anteriormente, sustituir “\$(lsb_release -cs)” a mano por el nombre, de una forma similar a:

```
echo \  
"deb [arch=$(dpkg --print-architecture)  
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu \  
stable
```

```
focal stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

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
```

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

```
sudo docker version
```

y obteniendo un resultado similar al siguiente:

```
sergi@ubuntu:~$ sudo docker version
Client: Docker Engine - Community
 Version:      20.10.4
 API version:  1.41
 Go version:   go1.13.15
 Git commit:   d3cb89e
 Built:        Thu Feb 25 07:05:27 2021
 OS/Arch:      linux/amd64
 Context:      default
 Experimental:  true

Server: Docker Engine - Community
 Engine:
  Version:      20.10.4
  API version:  1.41 (minimum version 1.12)
  Go version:   go1.13.15
  Git commit:   363e9a8
  Built:        Thu Feb 25 07:03:23 2021
  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
sergi@ubuntu:~$
```

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 ser 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+rxw "$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 Docker en Ubuntu, podemos usar el comando

```
sudo apt-get purge docker-ce docker-ce-cli containerd.io
```

Este comando eliminará Docker Engine, pero no eliminará contenedores e imágenes presentes en el sistema. Si queremos eliminar estos datos, tenemos dos opciones.

La primera consiste en, previamente a la eliminación de Docker Engine, ejecutar el comando:

```
sudo docker system prune -a
```

La segunda opción, se debe realizar tras eliminar Docker y consiste en el borrado manual de las mismas, con el comando:

```
sudo rm -rf /var/lib/docker
```

3. INSTALACIÓN DE DOCKER EN SISTEMAS WINDOWS

En este apartado veremos cómo instalar Docker Desktop en sistemas Windows.

Docker posee dos guías diferenciadas de instalación en sistemas Windows:

- Guía para Windows 10 Pro y Windows Server
 - <https://docs.docker.com/docker-for-windows/install/>
- Guía para Windows 10 Home
 - <https://docs.docker.com/docker-for-windows/install-windows-home/>

La principal diferencia entre ellas, es que el primer grupo requiere que se activen las características de Windows para Hyper-V, mientras que la segunda guía requiere la activación de WSL2 (Windows Subsystem for Linux 2).

3.1 Pasos previos Windows 10 Pro y Windows Server: activando Hyper-v

En este enlace se explica cómo habilitar Hyper-V en:

- Windows 10 Pro
 - <https://docs.microsoft.com/es-es/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>
- Windows Server

- <https://docs.microsoft.com/es-es/windows-server/virtualization/hyper-v/get-started/install-the-hyper-v-role-on-windows-server>

3.2 Pasos previos Windows 10 Home: Instalando WSL2

Antes de comenzar, nuestro sistema debe tener instalado WSL2. La guía para instalarlo se encuentra aquí <https://docs.microsoft.com/en-us/windows/wsl/install-win10>

A continuación un resumen de los pasos a realizar.

En primer lugar, utilizando una consola powershell **con permisos de administrador**, debemos habilitar WSL. Lo podemos hacer con el comando.

```
dism.exe /online /enable-feature  
/featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

Una vez habilitado, debemos asegurarnos que tenemos nuestro Windows 10 Home actualizado, al menos hasta la versión “Versión 1903, Build 18362”. Puedes comprobar tu versión ejecutando el comando “winver”.

Una vez actualizado y reiniciado el sistema, debemos habilitar la característica de “Virtual Machine”, lo cual podemos hacerlo con una consola powershell **con permisos de administrador**:

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform  
/all /norestart
```

Tras ello, **deberás reiniciar tu máquina antes del próximo paso.**

Una vez completado el reinicio, deberás instalar la última versión del Kernel de Linux para WSL2 aquí https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi

Finalizada la instalación, con una powershell **con permisos de administrador**, deberás establecer WSL2 como la versión por defecto al instalar una distribución de Linux.

```
wsl --set-default-version 2
```

Con esto ya tienes listo WSL2 y una versión de Ubuntu instalada en tu sistema Windows.

OPCIONAL: una vez instalado WSL2, puedes instalar distribuciones de Linux mediante la tienda Microsoft Store <https://aka.ms/wslstore> (por ejemplo Ubuntu 20). **Una vez instalada, es obligatorio que la inicies al menos una vez** para que descomprima parte del sistema y después te pida establecer un usuario administrador de la distribución Linux virtualizada.

3.3 Instalación de Docker Desktop

La instalación de Docker Desktop, que es la versión de Docker CE para sistemas Windows, es sencilla y básicamente consiste en descargar el instalador desde Docker Hub, en el siguiente enlace <https://hub.docker.com/editions/community/docker-ce-desktop-windows/> y seguir las instrucciones de instalación en pantalla.

En este enlace se puede ver un video en el que se realiza la instalación <https://www.youtube.com/watch?v=9AWYlt86B8>

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/>

3.4 Resolviendo problemas en la instalación de Docker Desktop

En mi experiencia con Docker Desktop, he sufrido problemas, incluso simplemente instalando actualizaciones. Hay cantidad de bugs típicos como los que os enlazo aquí:

- <https://github.com/docker/for-win/issues/7629>
- <https://forums.docker.com/t/docker-starts-but-trying-to-do-anything-results-in-error-during-connect/49007/4>

Este es el motivo de que, si os es posible, os recomiendo utilizar Docker en un sistema Linux, donde su implementación es más robusta.

En cualquier caso, en estos problemas de instalación en sistemas Windows, una solución que siempre me ha funcionado es hacer un “borrado completo” de Docker Desktop. Las instrucciones son las siguientes:

- Desinstala Docker Desktop.
- Ve a tu directorio de usuario (algo como C:\Users\tuUsuario) y elimina todo el contenido de la carpeta “.docker”, si es que está.
- Elimina todas las “variables de entorno” del sistema relacionadas con Docker. Suelen empezar con “DOCKER_”. Algunas de ellas pueden ser DOCKER_TLS_VERIFY, DOCKER_CERT_PATH o DOCKER_HOST.

- Aquí puedes ver como eliminar las variables de entorno de Windows 10 https://answers.microsoft.com/es-es/windows/forum/windows_10-other_settings/windows-10-variables-de-entorno-windows-10-version/703ea5fa-1db4-46da-8ff7-6261140bf58b
- Una vez hecho esto, reinicia el sistema.
- Una vez reiniciado, instala de nuevo “Docker Desktop”.

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 Katacoda <https://katacoda.com/> posee un “playground” de Docker disponible aquí: <https://www.katacoda.com/courses/docker/playground>

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/>