



2023  
2024



# Módulo: Despliegue de Aplicaciones Web



## Unidad de Trabajo: 1-2

### Primeros pasos con Docker



# Contenido de la unidad

<b>1</b>	<b><i>Introducción</i></b>	<b>3</b>
<b>2</b>	<b><i>Instalación</i></b>	<b>3</b>
2.1	Métodos de instalación	3
2.2	Instalación desde el propio repositorio de Docker	3
<b>3</b>	<b><i>Lanzando tu primer contenedor</i></b>	<b>4</b>

# 1 Introducción

En esta parte del tema vamos a ver como instalar Docker en un Ubuntu 22.04 y posteriormente veremos como arrancar un contenedor con una imagen del repositorio, así como algunas de las operaciones básicas que vamos a poder hacer con los contenedores.



## 2 Instalación

Para realizar la instalación de Docker, lo mejor es acudir a la web oficial (<https://docs.docker.com/get-docker/>). Ahí encontrarás toda la información detallada del proceso de instalación para los distintos sistemas operativos. Para nuestro ejemplo, vamos a realizar la instalación en Ubuntu 22.04.

**Nota:** Como siempre, antes de instalar cualquier nuevo software, tendremos que:

- Consultar los requisitos hardware/software necesarios para su instalación, ya que de no cumplirlos, podríamos tener problemas en la instalación y/o uso del mismo.
- Desinstalar cualquier versión previa del software

En nuestro caso, cumplimos con todos los requisitos necesarios y además al ser una máquina limpia, no tenemos instalado Docker, con lo que directamente procederemos a su instalación

### 2.1 Métodos de instalación

Existen distintos métodos de instalación de Docker:

- Instalación desde el propio repositorio de Docker
- Instalación y gestión de la actualización de manera manual
- Utilizar un script que se puede descargar desde la web oficial. Sólo recomendado para entornos de prueba y desarrollo.

Para esta guía, vamos a centrarnos en la primera de las opciones planteadas.

### 2.2 Instalación desde el propio repositorio de Docker

Antes de instalar Docker por primera vez en una nueva máquina, tendremos que añadir al apt-list el repositorio de Docker. Una vez añadido, podremos instalar y actualizar el software usando “apt”.

#### Paso 1 - Actualizar el sistema:

Antes de instalar cualquier software, es recomendable actualizar el sistema.

Para ello abriremos la terminal y ejecutaremos lo siguientes comandos:

```
$ sudo apt update
$ sudo apt upgrade
```

#### Paso 2 - Instalar paquetes pre-requisitos:

Instalar los paquetes necesarios para permitir la instalación de software a través de HTTPS.

Para conseguirlo, ejecutamos el siguiente comando desde la terminal:

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

#### Paso 3 - Agregar el repositorio de Docker:

Añadir la clave GPG oficial de Docker: necesario para el cifrado y transferencia segura

```
$ sudo install -m 0755 -d /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
```

Agregar el repositorio estable de Docker:

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

**Paso 4 - Instala Docker:**

Actualizar la base de datos de paquetes e instala Docker:

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

**Paso 5 – Iniciar y habilitar el servicio Docker:**

Iniciar el servicio Docker y habilitarlo para que se inicie automáticamente al arrancar:

```
$ sudo systemctl start docker
$ sudo systemctl enable docker
```

**Paso 6 – Verificar la instalación:**

Verificar que Docker se haya instalado correctamente y que está dando servicio ejecutando:

```
$ docker --version
$ sudo systemctl status docker
```

**Paso 7 – Permisos (opcional)**

Una vez ya tenemos instalado docker, conviene realizar algunos pasos adicionales para nuestra propia comodidad.

Lo primero es, que para evitar tener que ejecutar docker con derechos de administrador (sudo), tienes que añadir a tu usuario al grupo docker. Para ello, simplemente ejecuta la siguiente instrucción:

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

Una vez realizado esto, tienes que cerrar sesión y volver a entrar

## 3 Lanzando tu primer contenedor

Ahora es necesario, iniciar tu primer contenedor y ver que todo funciona de manera correcta. No te preocupes ahora mismo por las instrucciones. Simplemente queremos saber si Docker está instalado correctamente. Ya verás con detalle las instrucciones y el funcionamiento de Docker. Para comprobar que todo funciona correctamente, vas a arrancar tu primer contenedor. Tan solo tienes que ejecutar la siguiente instrucción:

```
$ docker run hello-world
```

En pocas palabras, lo que hace esta instrucción es ejecutar un contenedor que te va a mostrar **Hello from Docker!**. Lo primero que hace es comprobar si tienes la imagen hello-world localmente en tu equipo. Una vez comprobada que no la tienes, la descargará del repositorio de imágenes, y la ejecutará.

Si quieres, todavía puedes dar un paso más. Puedes meterte en el interior de un contenedor, y echar un vistazo. De esta manera puedes saber cómo es un contenedor por dentro. Verás que no hay nada extraño, que todo es más o menos igual. Para meterte en el interior de un contenedor ejecuta la siguiente instrucción:

```
$ docker run -it ubuntu bash
```

Donde:

- i: Permite la entrada estándar (STDIN) interactiva. Esto permite que interactúes con el contenedor a través del terminal.
- t: Asigna una TTY (terminal) al contenedor. Esto habilita una interacción más natural con el terminal dentro del contenedor.
- ubuntu: Es el nombre de la imagen de Docker que se utilizará para crear el contenedor. En este caso, estás utilizando la imagen de Ubuntu.
- bash: Este es el comando que se ejecutará dentro del contenedor después de iniciarse. En este caso, estás iniciando el intérprete de comandos "bash", lo que te permitirá interactuar con el contenedor como si estuvieras usando una terminal.

```
magarcia@ubuntu-desktop:~$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
b237fe92c417: Pull complete
Digest: sha256:ec050c32e4a6085b423d36ecd025c0d3ff00c38ab93a3d71a460ff1c44fa6d77
Status: Downloaded newer image for ubuntu:latest
root@3bce6c7e6f1:/#
```

```
magarcia@ubuntu-desktop:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:dcba6daec718f547568c562956fa47e1b03673dd010fe6ee58ca806767031d1c
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

magarcia@ubuntu-desktop:~$ █
```