

1. Instale los certificados y herramientas para trabajar con https:

```
usuario@usuario-VirtualBox:~$ sudo apt update
[sudo] password for usuario:
Hit:1 https://es.archive.ubuntu.com/ubuntu jammy InRelease
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
usuario@usuario-VirtualBox:~$ sudo apt install ca-certificates curl gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

2. Añada la clave GPG oficial de Docker:

```
usuario@usuario-VirtualBox:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

3. Añada el repositorio correspondiente a su versión de Ubuntu (escriba todo en una sola línea):

```
usuario@usuario-VirtualBox:~$ sudo 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
```

4. Actualice APT:

```
usuario@usuario-VirtualBox:~$ sudo apt update
```

5. Compruebe que APT se conecta con el repositorio adecuado:

```
usuario@usuario-VirtualBox:~$ apt policy docker-ce
```

En la pantalla debe mostrarse el repositorio de la versión de Docker disponible para la versión de Ubuntu indicada (en el ejemplo se muestran la versión de Docker 23.0.2 y la versión de Ubuntu jammy, es decir, Ubuntu 22.04):

```
usuario@usuario-VirtualBox:~$ apt policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:24.0.7-1~ubuntu.22.04~jammy
  Version table:
     5:24.0.7-1~ubuntu.22.04~jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable
amd64 Packages
```

Instale Docker

1. Actualice el administrador de paquetes:

```
usuario@usuario-VirtualBox:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
```

2. Instale la última versión de Docker

```
usuario@usuario-VirtualBox:~$ sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

3. Compruebe que el servicio está activo:
Se mostrará un mensaje similar a este (pulse Ctrl+C para salir del texto):

```
usuario@usuario-VirtualBox:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service;
   Active: active (running) since Thu 2023-12-14 11:45:
 TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 5819 (dockerd)
```

4. Para arrancar el servicio docker (si no estuviera activo):

```
usuario@usuario-VirtualBox:~$ sudo systemctl start docker
```

Docker – Images Hello-world

1. Compruebe que inicialmente no hay ningún contenedor creado (la opción -a hace que se muestren también los contenedores detenidos, sin ella se muestran sólo los contenedor que estén en marcha):

```
usuario@usuario-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS
NAMES
usuario@usuario-VirtualBox:~$
```

2. Compruebe que inicialmente tampoco disponemos de ninguna imagen

```
usuario@usuario-VirtualBox:~$ sudo docker image ls
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
usuario@usuario-VirtualBox:~$
```

3. Docker crea los contenedores a partir de imágenes locales (ya descargadas), pero si al crear el contenedor no se dispone de la imagen local, Docker descarga la imagen de su repositorio.

Como no tenemos todavía la imagen en nuestro ordenador, Docker descarga la imagen, crea el contenedor y lo pone en marcha. En este caso, la aplicación que contiene el contenedor hello-world simplemente escribe un mensaje de salida al arrancar e inmediatamente se detiene el contenedor. La respuesta será similar a esta:

```
usuario@usuario-VirtualBox:~$ sudo docker run hello-world
[sudo] password for usuario:
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:3155e04f30ad5e4629fac67d6789f8809d74fea22d4e9a82f757d28cee79e0c5
Status: Downloaded newer image for hello-world:latest
```

4. Si listamos ahora las imágenes existentes

```
usuario@usuario-VirtualBox:~$ sudo docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    9c7a54a9a43c   7 months ago   13.3kB
usuario@usuario-VirtualBox:~$
```

5. Si listamos ahora los contenedores existentes ..

```
usuario@usuario-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
9f8338606e45   hello-world  "/hello"                3 minutes ago   Exited (0)
) 3 minutes ago   keen_euler
```

6. Podemos crear tantos contenedores como queramos a partir de una imagen. Una vez la imagen está disponible localmente, Docker no necesita descargarla y el proceso de creación del contenedor es inmediato (aunque en el caso de hello-world la descarga es rápida, con imágenes más grandes la descarga inicial puede tardar un rato)
7. Si listamos los contenedores existentes

```
usuario@usuario-VirtualBox:~$ sudo docker run -d hello-world
564f080812518a83280b090229e6f938e62dc22a14ec2793e6d65ce9deb7d0db
```

8. Los contenedores se pueden destruir mediante el comando `rm`, haciendo referencia a ellos mediante su nombre o su id. No es necesario indicar el id completo, basta con escribir los primeros caracteres (de manera que no haya ambigüedades). Borre los dos contenedores existentes:

```
usuario@usuario-VirtualBox:~$ sudo docker rm 564
564
```

9. Podemos dar nombre a los contenedores al crearlos:

```
usuario@usuario-VirtualBox:~$ sudo docker run -d --name=hola-1 hello-world
3da3637ebbeb118cecf33c6c1549646a178fd719ec7faf8bcd77dc73767bff68
usuario@usuario-VirtualBox:~$
```

10. Si listamos los contenedores existentes ..

```
usuario@usuario-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE      PORTS          COMMAND          CREATED          STATUS
3da3637ebbeb   hello-world  "/hello"       hola-1           20 seconds ago   Exited (0) 19 seconds ago
e179025c947b   hello-world  "/hello"       intelligent_stonebraker 3 minutes ago    Exited (0) 3 minutes ago
9f8338606e45   hello-world  "/hello"       keen_euler       7 minutes ago    Exited (0) 7 minutes ago
usuario@usuario-VirtualBox:~$
```

11. Si intentamos crear un segundo contenedor con un nombre ya utilizado ...

```
usuario@usuario-VirtualBox:~$ sudo docker run -d --name=hola-1 hello-world
docker: Error response from daemon: Conflict. The container name "/hola-1" is already in use by container "3da3637ebbeb118cecf33c6c1549646a178fd719ec7faf8bcd77dc73767bff68". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
usuario@usuario-VirtualBox:~$
```