

# **INFORME SOBRE DOCKER**

---



Universidad  
del Cauca

## **PRESENTADO POR:**

Angee Daniela Quintero Gutierrez  
Manuel Alejandro Macías

## **ASIGNATURA:**

Laboratorio II de Sistemas Telemáticos

## **PRESENTADO A:**

Fulvio Yesid Vivas

UNIVERSIDAD DEL CAUCA

Facultad de Ingeniería Electrónica y Telecomunicaciones

PROGRAMA: Tecnología en Telemática

Popayán

15/05/2023

## **Contenido**

**1-Introducción**

**1.1- Objetivo**

**2-¿Qué es un Docker?**

**3-¿Para qué se usa un Docker?**

**4-importancia del Docker**

**5-Proceso de Docker**

**6-Observaciones del trabajo**

**7-Conclusión**

**8-Referencias Bibliográficas**

## **1-INTRODUCCIÓN**

En el entorno de desarrollo y despliegue de aplicaciones modernas, la tecnología de contenedores se ha convertido en un elemento fundamental. Los contenedores permiten la creación de entornos de aplicación aislados y portátiles que facilitan el desarrollo, la prueba y la implementación de aplicaciones en diferentes plataformas y entornos sin preocuparse por las diferencias de configuración. En este informe, se presenta el proceso de creación de un contenedor Docker en un entorno de VirtualBox, que es una solución de virtualización de código abierto ampliamente utilizada.

## **1.1-OBJETIVO**

Este informe tiene como objetivo proporcionar una descripción detallada del proceso de creación de un contenedor Docker en una máquina virtual VirtualBox, incluyendo los pasos suministrados en la guía.

## 2-¿Qué es un Docker?

Docker es una plataforma de código abierto que se utiliza para desarrollar, empaquetar y desplegar aplicaciones en contenedores. Los contenedores son entornos ligeros y portátiles que contienen todo lo necesario para que una aplicación se ejecute, incluyendo el código, las bibliotecas y las dependencias. Docker utiliza tecnologías de contenedorización para crear estos entornos, lo que facilita la creación, distribución y ejecución de aplicaciones en cualquier entorno compatible con Docker.

### **Características:**

**Contenedorización:** Docker utiliza contenedores para encapsular aplicaciones y sus dependencias. Los contenedores son unidades aisladas que se ejecutan en el sistema operativo del anfitrión y comparten el mismo kernel del sistema operativo. Esto los hace ligeros y eficientes en términos de recursos.

**Imagen de Docker:** Una imagen de Docker es un paquete de solo lectura que contiene todo lo necesario para ejecutar una aplicación, incluyendo el código, las bibliotecas y las configuraciones. Las imágenes de Docker se utilizan como plantillas para crear contenedores en tiempo de ejecución.

**Dockerfile:** Un Dockerfile es un archivo de texto que contiene instrucciones para construir una imagen de Docker. Permite definir cómo se debe configurar el entorno del contenedor, qué aplicaciones deben instalarse y cómo se deben configurar.

**Registro de Docker:** Docker proporciona un servicio llamado Docker Hub, que es un registro de imágenes de Docker públicas y privadas. Los desarrolladores pueden compartir sus imágenes en Docker Hub o en registros privados para facilitar la distribución de aplicaciones.

**Portabilidad:** Una de las principales ventajas de Docker es la portabilidad de las aplicaciones en contenedores. Los contenedores se ejecutan de la misma manera en cualquier entorno que admita Docker, lo que facilita el desarrollo en una máquina local y el despliegue en servidores de producción o en la nube.

**Seguridad:** Docker proporciona características de aislamiento que ayudan a mantener la seguridad de las aplicaciones en contenedores, aunque se ejecuten en el mismo sistema.

## 3¿Para qué se utiliza un Docker?

Docker se utiliza principalmente para simplificar y mejorar el proceso de desarrollo, implementación y administración de aplicaciones.

**Aislamiento de aplicaciones:** Docker permite encapsular aplicaciones y sus dependencias en contenedores, lo que proporciona un alto grado de aislamiento. Esto significa que las aplicaciones pueden ejecutarse de manera independiente en su propio entorno, sin interferir con otras aplicaciones en el mismo sistema.

**Desarrollo y pruebas consistentes:** Docker permite a los desarrolladores crear entornos de desarrollo y pruebas idénticos a los entornos de producción. Esto garantiza que las pruebas y la depuración se realicen en entornos que sean lo más similares posible al entorno de producción, lo que reduce los errores y problemas relacionados con las diferencias de configuración.

**Escalabilidad:** Docker facilita la creación y el escalado de aplicaciones en contenedores de manera eficiente.

**Implementación rápida:** Docker simplifica la implementación de aplicaciones. Los contenedores se pueden crear y desplegar en cuestión de segundos, en lugar de minutos u horas que pueden llevar otras formas de implementación.

**Administración simplificada:** Docker ofrece una forma estandarizada de administrar aplicaciones y sus dependencias. Las actualizaciones y correcciones se pueden aplicar fácilmente a través de nuevas imágenes de contenedor, lo que simplifica la gestión de parches y actualizaciones.

**Colaboración en equipos:** Docker facilita la colaboración entre equipos de desarrollo y operaciones.

**Reutilización de componentes:** Docker fomenta la reutilización de componentes y servicios. Los contenedores se pueden usar como bloques de construcción para crear aplicaciones más complejas y se pueden compartir a través de repositorios de imágenes.

#### **4-Importancia de un Docker.**

La importancia de Docker empieza por la capacidad para revolucionar la forma en que desarrollamos, implementamos y gestionamos aplicaciones.

Algunas de las principales razones que hacen que Docker sea una tecnología muy importante son:

**Aislamiento y consistencia:** Docker proporciona un alto grado de aislamiento, lo que significa que las aplicaciones y sus dependencias se ejecutan de manera independiente en contenedores.

**Eficiencia de recursos:** Los contenedores son ligeros y comparten el kernel del sistema operativo del anfitrión, lo que los hace eficientes en términos de recursos. O sea se puede ejecutar múltiples contenedores en una única máquina física sin sacrificar el rendimiento.

**Mantenimiento y actualizaciones simplificadas:** Las actualizaciones y correcciones se pueden aplicar de manera más sencilla a través de nuevas imágenes de contenedor.

**Seguridad mejorada:** Aunque los contenedores comparten el mismo kernel del sistema operativo, Docker proporciona mecanismos de seguridad para aislar y limitar los recursos y las capacidades de los contenedores, lo que ayuda a proteger las aplicaciones y la infraestructura oculta,

**Reducción de costos:** Al optimizar el uso de recursos y simplificar la administración de aplicaciones, Docker puede ayudar a reducir los costos operativos y de infraestructura.

#### 4-Proceso del Docker:

Para iniciar con el proceso de Docker, debemos instalarlo en una maquina virtual Box, en este caso instalamos la siguiente maquina: [Máquina virtual VirtualBox \(OVA\) de Ubuntu 22.04.2, actualizada al 30 de marzo de 2023 \(1,77 GB\)](#) y procedo a instalar el docker manualmente en la máquina Virtual.

Iniciamos Actualizando la distribución y ejecutamos los siguientes comandos en la terminal. Los comandos son:

**-sudo apt update:** se utiliza para mantener actualizada la lista de paquetes disponibles en el sistema basado en Debian, lo que facilita la gestión de actualizaciones y la instalación de software en el sistema.

**-sudo apt full-upgrade:** Se utiliza para actualizar todos los paquetes instalados en el sistemas.

**-sudo apt install ca-certificates curl gnupg** Este comando se utiliza para instalar los siguientes paquetes:

**ca-certificates:** Este paquete contiene certificados de autoridades de certificación (CAs) confiables que se utilizan para verificar la autenticidad de sitios web y servicios en línea mediante HTTPS. Es importante tener estos certificados instalados en el sistema para garantizar conexiones seguras.

**Curl:** Curl es una herramienta de línea de comandos que se utiliza para transferir datos con URLs. Es una herramienta muy versátil que admite varios protocolos de transferencia, como HTTP, HTTPS, FTP, SCP y más. Se usa comúnmente para descargar archivos y realizar solicitudes HTTP desde la línea de comandos.

**gnupg:** Gnupg, Se utiliza para gestionar claves criptográficas, firmar y verificar archivos, y realizar otras operaciones relacionadas con la seguridad.

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

 Este comando se utiliza para agregar la clave GPG (GnuPG) del repositorio de Docker a la lista de claves de confianza en el sistema. Esta clave se utiliza para verificar la autenticidad y la integridad de los paquetes de Docker que se descargaron e instalaron en el sistema Ubuntu.

```
-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
```

 Este comando se utiliza para agregar una entrada al archivo `/etc/apt/sources.list.d/docker.list` en el sistema Ubuntu. Esta entrada se utiliza para configurar el repositorio de Docker en el sistema y habilitar la instalación de Docker desde ese repositorio.

**-apt policy docker-ce** Este comando se utiliza para verificar y mostrar información sobre la disponibilidad de paquetes de Docker en el sistema. Específicamente, proporciona información sobre las versiones disponibles, el estado de instalación y de dónde provienen los paquetes.

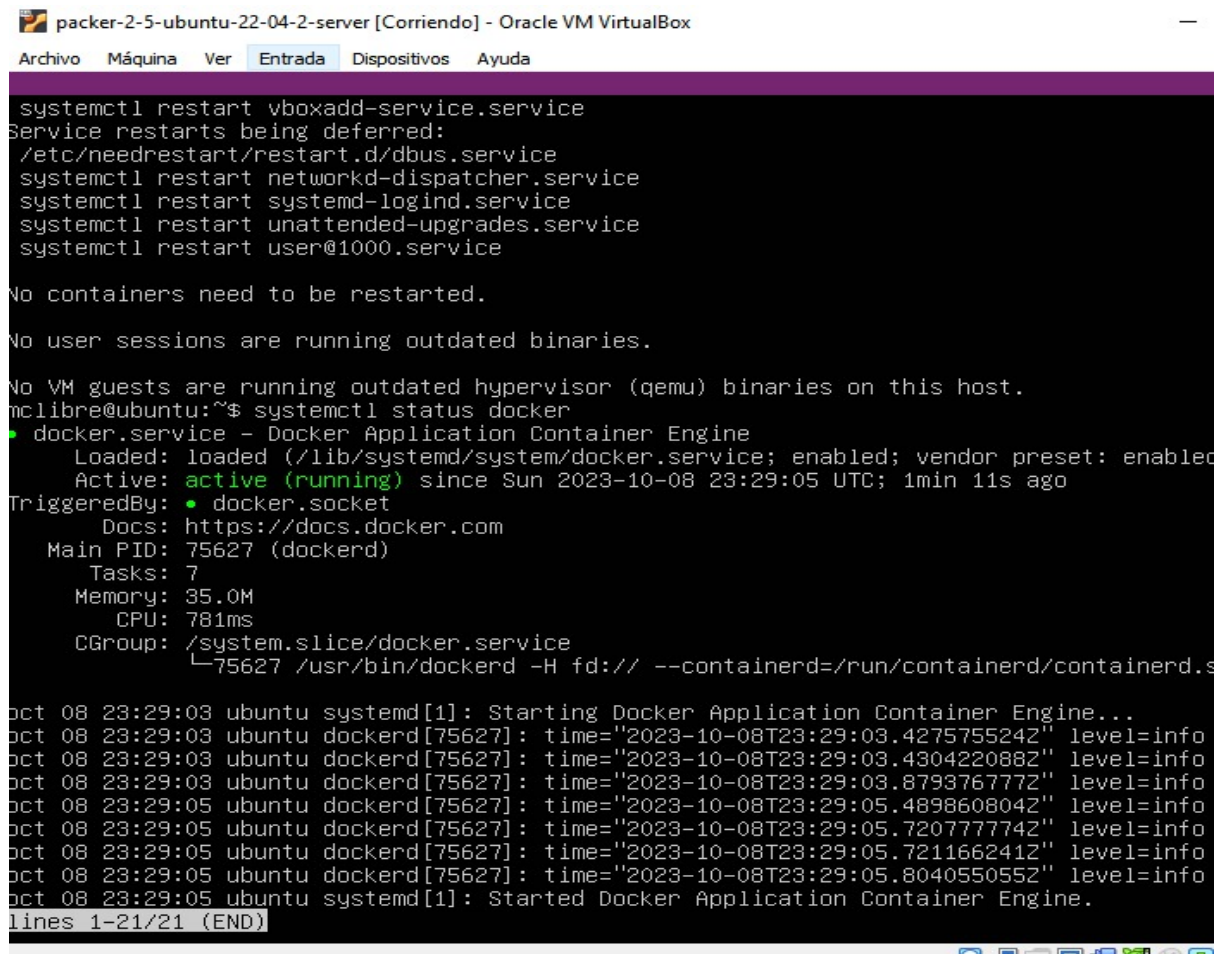
```
-sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

 Este comando se utiliza para instalar varios componentes relacionados con Docker.

**-systemctl status docker** Este comando se utiliza para verificar el estado del servicio de Docker en un sistema Linux. Proporciona información sobre si el servicio de Docker está en funcionamiento, si ha encontrado algún error o si se ha detenido. (EVIDENCIA EN LA IMAGEN1)



## imagen1



```
packer-2-5-ubuntu-22-04-2-server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

systemctl restart vboxadd-service.service
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
mcilibre@ubuntu:~$ systemctl status docker
• docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-10-08 23:29:05 UTC; 1min 11s ago
   TriggeredBy: • docker.socket
   Docs: https://docs.docker.com
   Main PID: 75627 (dockerd)
   Tasks: 7
   Memory: 35.0M
   CPU: 781ms
   CGroup: /system.slice/docker.service
           └─75627 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.s

oct 08 23:29:03 ubuntu systemd[1]: Starting Docker Application Container Engine...
oct 08 23:29:03 ubuntu dockerd[75627]: time="2023-10-08T23:29:03.427575524Z" level=info
oct 08 23:29:03 ubuntu dockerd[75627]: time="2023-10-08T23:29:03.430422088Z" level=info
oct 08 23:29:03 ubuntu dockerd[75627]: time="2023-10-08T23:29:03.879376777Z" level=info
oct 08 23:29:05 ubuntu dockerd[75627]: time="2023-10-08T23:29:05.489860804Z" level=info
oct 08 23:29:05 ubuntu dockerd[75627]: time="2023-10-08T23:29:05.720777774Z" level=info
oct 08 23:29:05 ubuntu dockerd[75627]: time="2023-10-08T23:29:05.721166241Z" level=info
oct 08 23:29:05 ubuntu dockerd[75627]: time="2023-10-08T23:29:05.804055055Z" level=info
oct 08 23:29:05 ubuntu systemd[1]: Started Docker Application Container Engine.
lines 1-21/21 (END)
```

-**sudo docker ps -a** se utiliza para listar todos los contenedores de Docker en el sistema, incluyendo aquellos que están en ejecución y aquellos que están detenidos.

-**sudo docker image ls** Se utiliza para listar las imágenes del contenedor Docker que están almacenadas en el sistema

-**sudo docker run IMAGEN** Se utiliza para crear y ejecutar un nuevo contenedor Docker basado en una imagen específica.

-**sudo docker run hello-world** : Se utiliza para ejecutar un Docker llamado hello-world (COMO EN LA IMAGEN 2 y 3)

## imagen2

```
--volumes-from list      Mount volumes from the specified container(s)
-w, --workdir string      Working directory inside the container
mclibre@ubuntu:~$ sudo docker run IMAGEN
docker: invalid reference format: repository name must be lowercase.
See 'docker run --help'.
mclibre@ubuntu:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23db
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/

mclibre@ubuntu:~$ sudo docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    latest    9c7a54a9a43c   5 months ago   13.3kB
mclibre@ubuntu:~$ _
```

## imagen3

packer-2-5-ubuntu-22-04-2-server [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
See 'docker run --help'.
mclibre@ubuntu:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23db
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

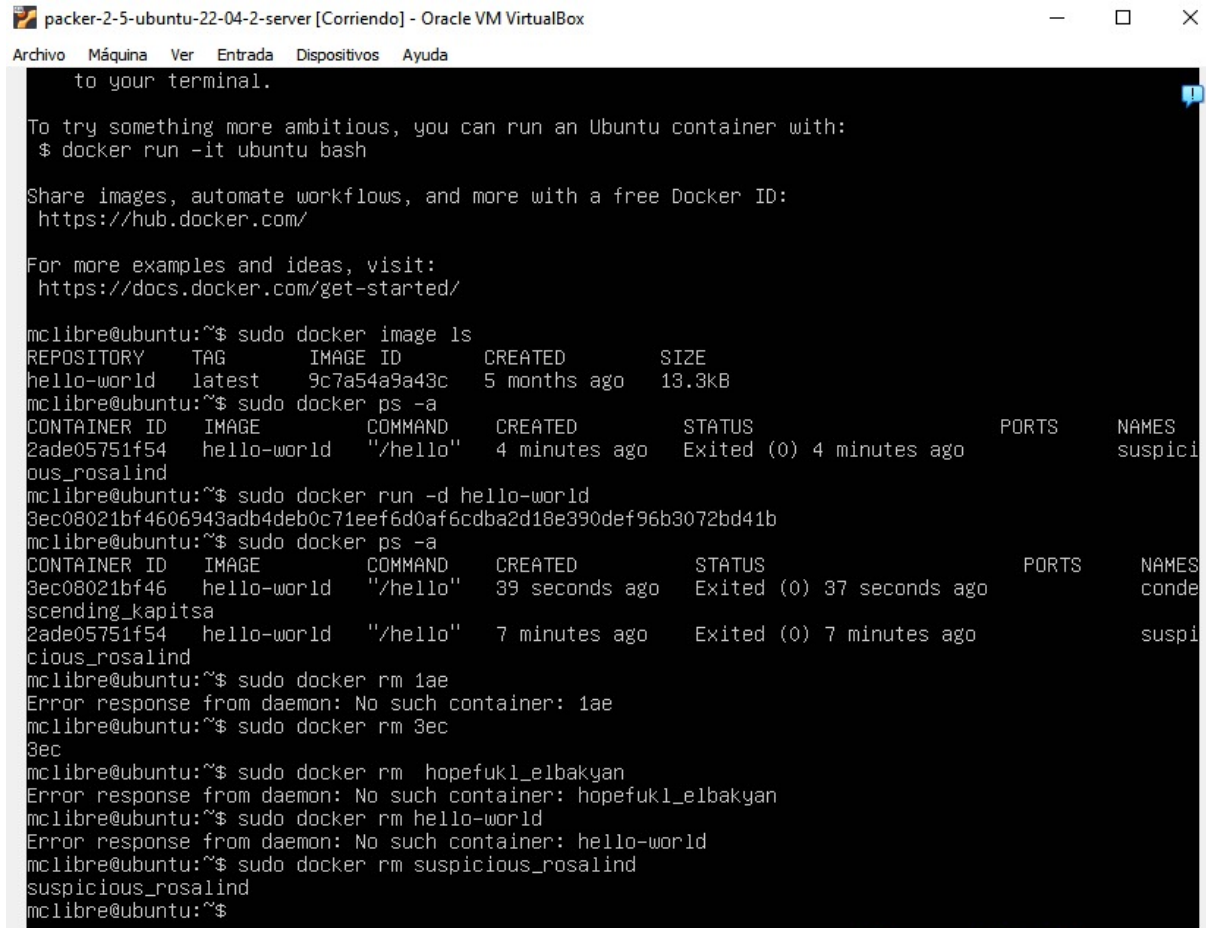
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

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

mclibre@ubuntu:~$ sudo docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    latest    9c7a54a9a43c   5 months ago   13.3kB
mclibre@ubuntu:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
2ade05751f54   hello-world  "/hello"   4 minutes ago   Exited (0) 4 minutes ago           suspicious_rous_rosalind
mclibre@ubuntu:~$ _
```

**-sudo docker rm** Se utilizó para eliminar los contenedores que ya no son necesarios en este caso eliminamos el docker 1ae y el Docker hpefuk\_elbakyan

#### *imagen4*



```
packer-2-5-ubuntu-22-04-2-server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

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/

mclibre@ubuntu:~$ sudo docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    9c7a54a9a43c   5 months ago  13.3kB
mclibre@ubuntu:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS          PORTS          NAMES
2ade05751f54   hello-world  "/hello"            4 minutes ago  Exited (0) 4 minutes ago          suspicious_rosalind
mclibre@ubuntu:~$ sudo docker run -d hello-world
3ec08021bf4606943adb4deb0c71eef6d0af6cdba2d18e390def96b3072bd41b
mclibre@ubuntu:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS          PORTS          NAMES
3ec08021bf46   hello-world  "/hello"            39 seconds ago  Exited (0) 37 seconds ago          condescending_kapitsa
2ade05751f54   hello-world  "/hello"            7 minutes ago  Exited (0) 7 minutes ago          suspicious_rosalind
mclibre@ubuntu:~$ sudo docker rm 1ae
Error response from daemon: No such container: 1ae
mclibre@ubuntu:~$ sudo docker rm 3ec
3ec
mclibre@ubuntu:~$ sudo docker rm hopefuk1_elbakyan
Error response from daemon: No such container: hopefuk1_elbakyan
mclibre@ubuntu:~$ sudo docker rm hello-world
Error response from daemon: No such container: hello-world
mclibre@ubuntu:~$ sudo docker rm suspicious_rosalind
suspicious_rosalind
mclibre@ubuntu:~$
```

**-sudo docker run -d --name=hola-1 hello-world** Se utiliza para crear y ejecutar un contenedor Docker basado en la imagen "hello-world" en segundo plano (en modo daemon).

**-sudo docker exec -it apache-2 /bin/bash** Se utiliza para ejecutar un shell interactivo dentro de un contenedor Docker llamado "apache-2".

## imagen5

```
packer-2-5-ubuntu-22-04-2-server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

--platform string      Set platform if server is multi-platform capable
--privileged           Give extended privileges to this container
-p, --publish list     Publish a container's port(s) to the host
-P, --publish-all     Publish all exposed ports to random ports
--pull string          Pull image before running ("always", "missing",
                      "never") (default "missing")
-q, --quiet            Suppress the pull output
--read-only            Mount the container's root filesystem as read only
--restart string       Restart policy to apply when a container exits
                      (default "no")
--rm                  Automatically remove the container when it exits
--runtime string       Runtime to use for this container
--security-opt list    Security Options
--shm-size bytes      Size of /dev/shm
--sig-proxy            Proxy received signals to the process (default true)
--stop-signal string   Signal to stop the container
--stop-timeout int     Timeout (in seconds) to stop a container
--storage-opt list     Storage driver options for the container
--sysctl map           Sysctl options (default map[])
--tmpfs list           Mount a tmpfs directory
-t, --tty              Allocate a pseudo-TTY
--ulimit ulimit        Ulimit options (default [])
-u, --user string      Username or UID (format: <name|uid[:<group|gid>])
--userns string        User namespace to use
--uts string           UTS namespace to use
-v, --volume list      Bind mount a volume
--volume-driver string Optional volume driver for the container
--volumes-from list    Mount volumes from the specified container(s)
-w, --workdir string   Working directory inside the container

mcilibre@ubuntu:~$ sudo docker run -d -P --name=apache-1 bitnami/apache
Unable to find image 'bitnami/apache:latest' locally
latest: Pulling from bitnami/apache
8e5cf0abe9fb: Pull complete
Digest: sha256:19632a24f13ea4bc3ee551489f32c96c2fb752b177e15793a2214e124bf35210
Status: Downloaded newer image for bitnami/apache:latest
9486a8df4988e2e21b79c7bbf24cd7cb8ee5bdf6a74050eb465bfbf43a4d90d2
mcilibre@ubuntu:~$ _
```

## imagen6

```
packer-2-5-ubuntu-22-04-2-server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Apache en Docker</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>
  <h1>¡Hola, mundo!</h1>
</body>
</html>

mcilibre@ubuntu:~$ sudo docker exec -it apache-2 /bin/bash
I have no name!@d7eac147c9dd:/app$ _
```

### **Observaciones del trabajo.**

Al realizar la práctica orientada por el profesor, desafortunadamente tuvimos muchos inconvenientes ya que algunos comandos no nos funcionaron para poder continuar y finalizar la práctica, sin embargo nos hemos informado sobre cada uno de los comandos y hemos investigado sobre cada una de sus funciones para poder desenvolvernos con más facilidad en un próximo intento.

## **CONCLUSIONES**

Docker es una herramienta esencial en la informática contemporánea que simplifica y mejora la forma en que desarrollamos y desplegamos aplicaciones. Su enfoque en la contenedorización ofrece una mayor eficiencia, portabilidad y escalabilidad, lo que lo convierte en una tecnología fundamental en la gestión de aplicaciones en una variedad de entornos. El conocimiento y la comprensión de Docker son valiosos para cualquier persona involucrada en el desarrollo y la administración de software en la actualidad.

## **REFERENCIAS BIBLIOGRÁFICAS**

[https://www.youtube.com/watch?v=pn0eeYr\\_8U8](https://www.youtube.com/watch?v=pn0eeYr_8U8)

<https://www.hostinger.co/tutoriales/que-es-docker>

<https://www.ionos.es/digitalguide/servidores/know-how/que-es-docker/>

<https://openwebinars.net/blog/docker-que-es-sus-principales-caracteristicas/>