



Práctica 3: Docker sobre EC2.

Albañiles Digitales: Cloud (Amazon Web Services)

Descripción:

En esta práctica, vamos a explorar una de las formas más simples de desplegar un contenedor Docker en la nube: utilizando el daemon de Docker sobre una instancia EC2.

Objetivos

Descargar y arrancar el servicio Docker en una instancia EC2

- Utilizar `yum` para instalar Docker
- Ejecutar el daemon para poder usar la CLI (línea de comandos) de Docker en la instancia

Descargar una imagen Docker en la instancia

- Explorar la página Docker Hub
- Descargar una imagen pública del Hub a la instancia
- Listar la imagen utilizando la CLI

Levantar un contenedor Docker

- Utilizar la imagen descargada como base para ejecutar varios contenedores Docker en la instancia, y realizar una serie de comandos
- Listar los contenedores Docker

Generar nuestra propia imagen Docker

- Generar una imagen Docker a partir de un contenedor en ejecución
- Utilizar esa imagen para levantar un nuevo contenedor

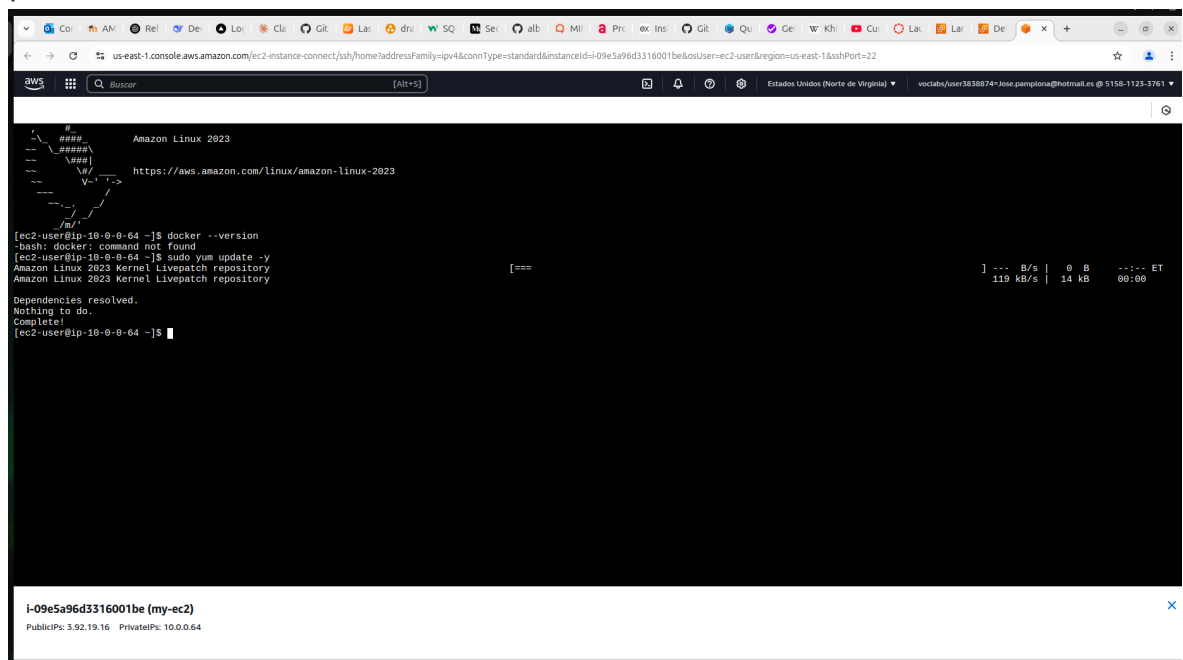
Limpiar la máquina

- Usando la CLI, parar y eliminar los contenedores Docker de la instancia

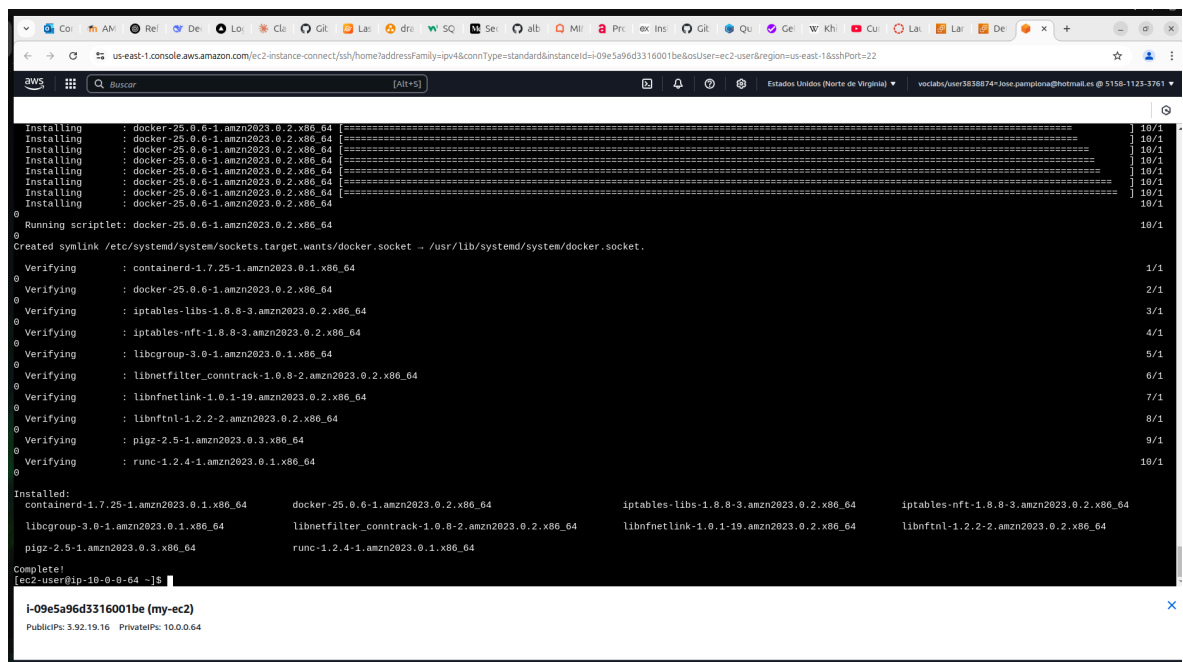
Pasos:

Descargar y arrancar el servicio Docker

1. Entra a una máquina de EC2. Puedes usar la que creaste en la práctica número 2.
2. Usa el comando: `sudo yum update -y` para actualizar las librerías que se pueden instalar



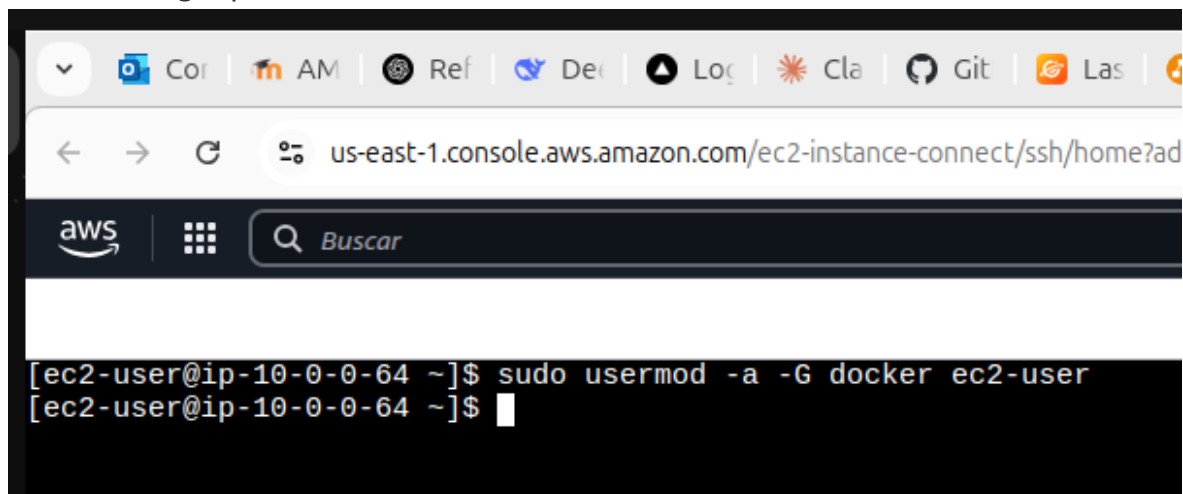
3. Usa el comando: `sudo yum install -y docker` para instalar Docker



```
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 [=====] 10/1
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 [=====] 10/1
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 [=====] 10/1
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 [=====] 10/1
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 [=====] 10/1
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 [=====] 10/1
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/1
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Verifying : containerd-1.7.25-1.amzn2023.0.1.x86_64 1/1
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64 2/1
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/1
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/1
Verifying : libcgrou-3.0-1.amzn2023.0.1.x86_64 5/1
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/1
Verifying : libnetfilterlink-1.0.1-19.amzn2023.0.2.x86_64 7/1
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/1
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/1
Verifying : runc-1.2.4-1.amzn2023.0.1.x86_64 10/1
Installed:
containerd-1.7.25-1.amzn2023.0.1.x86_64 docker-25.0.6-1.amzn2023.0.2.x86_64 iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
libcgrou-3.0-1.amzn2023.0.1.x86_64 libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 libnetfilterlink-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64
pigz-2.5-1.amzn2023.0.3.x86_64 runc-1.2.4-1.amzn2023.0.1.x86_64
Complete!
[ec2-user@ip-10-0-0-64 ~]$
```

i-09e5a96d3316001be (my-ec2)
PublicIP: 3.92.19.16 PrivateIP: 10.0.0.64

4. Usa el comando: `sudo usermod -a -G docker ec2-user` para añadir tu usuario al grupo Docker



```
[ec2-user@ip-10-0-0-64 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-10-0-0-64 ~]$
```

5. Cierra la pestaña de la instancia y vuelve a abrirla, para que se haga login otra vez y adquieras los permisos del grupo docker

EC2 > Instancias > i-09e5a96d3316001be > Conectarse a la instancia

Conectarse a la instancia Información

Conéctese a la instancia i-09e5a96d3316001be (my-ec2) mediante cualquiera de estas opciones

Conexión de la instancia EC2

Administrador de sesiones

Cliente SSH

Consola de serie de EC2

ID de la instancia

i-09e5a96d3316001be (my-ec2)

Tipo de conexión

☒ Conectarse mediante la Conexión de la instancia EC2

Conect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Conectarse mediante punto de conexión de EC2 Instance Connect

Conéctese mediante el cliente basado en navegador de EC2 Instance Connect, con una dirección IPv4 privada y un punto de conexión de VPC.

☒ Dirección IPv4 pública

3.92.19.16

☐ Dirección IPv6

—

Nombre de usuario

Escriba el nombre de usuario definido en la AMI utilizada para lanzar la instancia. Si no definió un nombre de usuario personalizado, utilice el nombre de usuario predeterminado, ec2-user.

Nota: En la mayoría de los casos, el nombre de usuario predeterminado, ec2-user, es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado.

Cancelar

Conectar

6. Inicia el servicio de Docker con `sudo service docker start`

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-09e5a96d3316001be&osUser=ec2-user&sshPort=22&addressFamily=ipv4

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Feb 19 11:18:09 2025 from 18.206.107.27
[ec2-user@ip-10-0-0-64 ~]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-10-0-0-64 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-10-0-0-64 ~]$
```

7. Usa el comando: `docker --version` para ver la versión de Docker. Si se muestra una línea con la versión, es que Docker está funcionando correctamente.

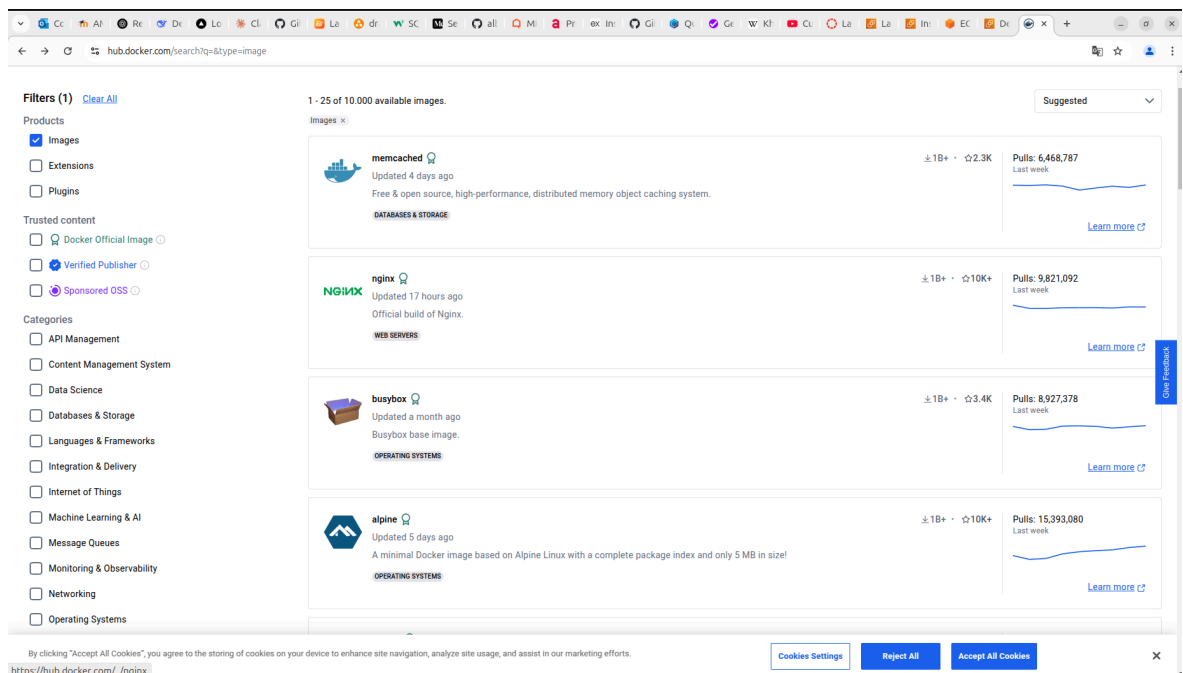
```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-09e5a96d3316001be&osUser=ec2-user&sshPort=22&addressFamily=ipv4

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

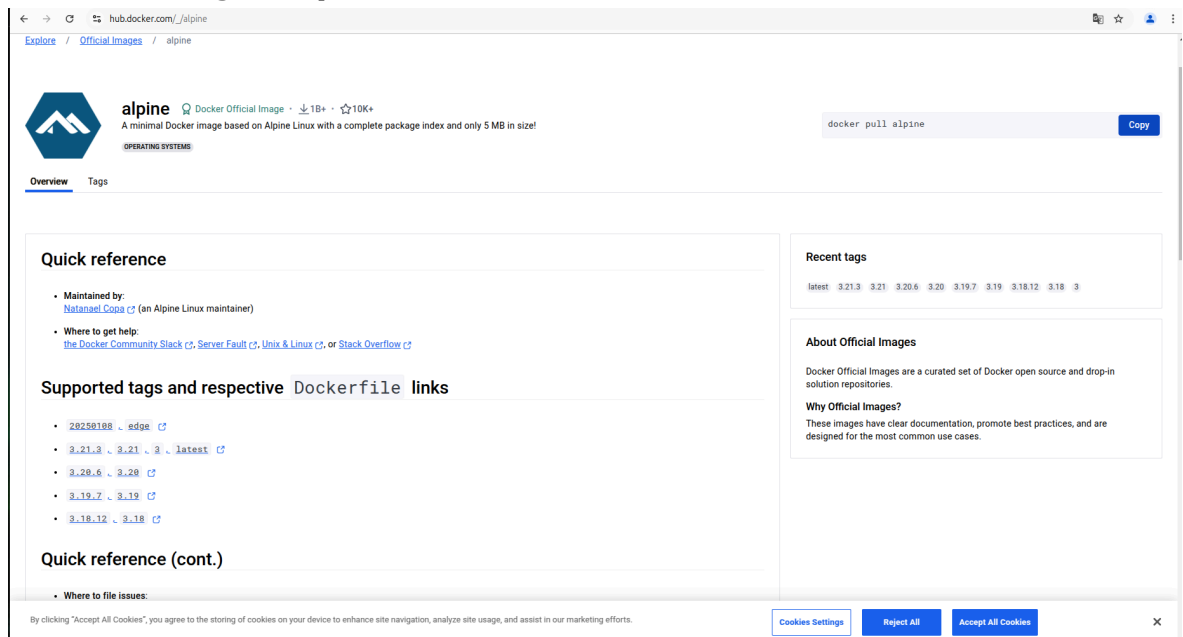
Last login: Wed Feb 19 11:18:09 2025 from 18.206.107.27
[ec2-user@ip-10-0-0-64 ~]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-10-0-0-64 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-10-0-0-64 ~]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-10-0-0-64 ~]$
```

Descargar una imagen Docker en la instancia

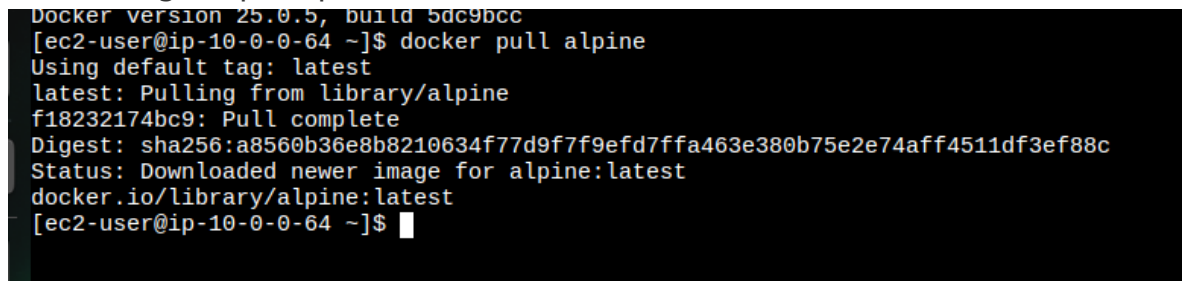
1. Explora Docker Hub: <https://hub.docker.com/search?q=>



2. Busca la imagen **alpine**



3. Mira su página, la documentación, y copia el comando para hacer pull de la imagen que aparece arriba a la derecha



4. Lista las imágenes Docker descargadas en la instancia con el comando:

`docker images` . Si en la lista aparece la imagen **alpine**, significa que se ha descargado correctamente.

```
[ec2-user@ip-10-0-0-64 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
alpine        latest    aded1e1a5b37   5 days ago    7.83MB
[ec2-user@ip-10-0-0-64 ~]$
```

Levantar un contenedor Docker

1. Utiliza el comando: `docker run -it alpine` para levantar un contenedor de forma interactiva. Se abrirá una sesión con la que puedes lanzar distintos comandos, al igual que una terminal normal.

```
[ec2-user@ip-10-0-0-64 ~]$ docker run -it alpine
/ #
```

2. Cuando termines de trastear con el contenedor, usa el comando: `exit`

```
[ec2-user@ip-10-0-0-64 ~]$ docker run -it alpine
/ # exit
[ec2-user@ip-10-0-0-64 ~]$
```

3. Usa el comando: `docker ps -a` para listar los contenedores. Deberías ver un contenedor con el STATUS: Exited (0).

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS                    PORTS          NAMES
4130b13cac66   alpine    "/bin/sh"               5 minutes ago   Exited (0) About a minute ago           laughing_tharp
[ec2-user@ip-10-0-0-64 ~]$
```

Generar nuestra propia imagen Docker

1. Utiliza el comando: `docker run -itd alpine` para levantar un contenedor de forma interactiva en segundo plano (gracias al flag `-d` indicado en el comando).

```
[ec2-user@ip-10-0-0-64 ~]$ docker run -itd alpine
0dd8ee70b0dc8af93f386db2000c6d248150a5b1acbf9dc819032381acf89f6b
[ec2-user@ip-10-0-0-64 ~]$
```

2. Utiliza de nuevo el comando: `docker ps -a` para listar los contenedores. Ahora, además del contenedor con STATUS: Exited (0), deberías ver un contenedor con STATUS: Up. Quédate con el id del contenedor corriendo para usarlo en el siguiente comando.

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0dd8ee70b0dc	alpine	"/bin/sh"	2 minutes ago	Up 2 minutes		pedantic_moser
4130b13cac66	alpine	"/bin/sh"	19 minutes ago	Exited (0) 14 minutes ago		laughing_tharp

```
[ec2-user@ip-10-0-0-64 ~]$
```

3. Usa el comando: `docker commit {id del contenedor} nueva-imagen` para generar una imagen con el nombre **nueva-imagen** a partir del contenedor que está corriendo.

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0dd8ee70b0dc	alpine	"/bin/sh"	2 minutes ago	Up 2 minutes		pedantic_moser
4130b13cac66	alpine	"/bin/sh"	19 minutes ago	Exited (0) 14 minutes ago		laughing_tharp

```
[ec2-user@ip-10-0-0-64 ~]$ docker commit 0dd8ee70b0dc nueva-imagen
sha256:3474cac4e34063a564497941ac4f906b11a5c75d616e4cbcff5b59560c1ac73e
[ec2-user@ip-10-0-0-64 ~]$
```

4. Usa el comando: `docker images` para listar las imágenes. Ahora, además de la imagen **alpine**, se debería poder ver en la lista nuestra imagen creada **nueva-imagen**.

```
[ec2-user@ip-10-0-0-64 ~]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nueva-imagen	latest	3474cac4e340	3 minutes ago	7.83MB
alpine	latest	aded1e1a5b37	5 days ago	7.83MB

```
[ec2-user@ip-10-0-0-64 ~]$
```

Limpiar la máquina

1. Utiliza el comando: `docker ps -a` para ver nuestros contenedores.

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0dd8ee70b0dc	alpine	"/bin/sh"	11 minutes ago	Up 11 minutes		pedantic_moser
4130b13cac66	alpine	"/bin/sh"	29 minutes ago	Exited (0) 24 minutes ago		laughing_tharp

```
[ec2-user@ip-10-0-0-64 ~]$
```

2. Utiliza el comando: `docker stop {id del contenedor corriendo}` para parar el contenedor que está ejecutándose.

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0dd8ee70b0dc	alpine	"/bin/sh"	13 minutes ago	Up 13 minutes		pedantic_moser
4130b13cac66	alpine	"/bin/sh"	30 minutes ago	Exited (0) 25 minutes ago		laughing_tharp

```
[ec2-user@ip-10-0-0-64 ~]$ docker stop 0dd8ee70b0dc
```

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0dd8ee70b0dc	alpine	"/bin/sh"	15 minutes ago	Exited (137) About a minute ago		pedantic_moser
4130b13cac66	alpine	"/bin/sh"	32 minutes ago	Exited (0) 28 minutes ago		laughing_tharp

```
[ec2-user@ip-10-0-0-64 ~]$
```

3. Utiliza el comando: `docker rm {id del contenedor}` para ir borrando uno

a uno todos los contenedores.

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
0dd8ee70b0dc   alpine    "/bin/sh" 15 minutes ago   Exited (137) About a minute ago   pedantic_moser
4130b13cac66   alpine    "/bin/sh" 32 minutes ago   Exited (0) 28 minutes ago         laughing_tharp
[ec2-user@ip-10-0-0-64 ~]$ docker rm 0dd8ee70b0dc
0dd8ee70b0dc
[ec2-user@ip-10-0-0-64 ~]$ docker rm 4130b13cac66
4130b13cac66
[ec2-user@ip-10-0-0-64 ~]$
```

4. Cuando estén todos los contenedores borrados, usa de nuevo el comando: `docker ps -a` para demostrar que no quedan contenedores en la instancia.

```
[ec2-user@ip-10-0-0-64 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
[ec2-user@ip-10-0-0-64 ~]$
```

Evaluación

Esta práctica se evaluará en base a los pasos que logréis avanzar. Para ello, será necesario enviar un PDF con los pantallazos de las diferentes etapas de esta práctica, mostrando el trabajo realizado:

- Descargar Docker: pantallazo del resultado del comando `docker --version`
- Descargar imagen Docker: pantallazo del resultado del comando `docker images`
- Levantar un contenedor Docker: pantallazo del resultado del comando `docker ps -a`
- Generar nuestra propia imagen Docker: pantallazo del resultado del comando `docker images` tras la generación de la nueva imagen
- Limpiar la máquina: pantallazo del resultado del comando `docker ps -a`

Todas estas capturas de pantalla están a lo largo de la práctica.

Fin de práctica.