



T3A2 - VOLÚMENES, REDES E IMÁGENES PERSONALIZADAS

1. VOLÚMENES DE DOCKER (2.5 puntos)

1.- Crear los siguientes volúmenes con la orden docker volume:

- volumen_datos
- volumen_web

Para crear estos volúmenes ejecutamos el comando **docker volumen create + nombre del volumen**

```
root@ubuntu: /home/eric/Desktop
root@ubuntu:/home/eric/Desktop# docker volume create volumen_datos
volumen_datos
root@ubuntu:/home/eric/Desktop# docker volume create volumen_web
volumen_web
root@ubuntu:/home/eric/Desktop#
```

2.- Una vez creados estos volúmenes:

- Arrancar un contenedor llamado **c1** sobre la imagen **php:7.4-apache** que monte el volumen **volumen_web** en la ruta **/var/www/html**.

Para arrancar en segundo plano un contenedor llamado **c1** sobre la imagen **php:7.4-apache** y montar el volumen **volumen_web** en la ruta **/var/www/html**, utilizaremos el siguiente comando

```
root@ubuntu:/home/eric/Desktop# docker run -d --name c1 -v volumen_web:/var/www/html php:7.4-apache
Unable to find image 'php:7.4-apache' locally
7.4-apache: Pulling from library/php
a603fa5e3b41: Pull complete
c428f1a49423: Pull complete
156740b07ef8: Pull complete
fb5a4c8af82f: Pull complete
25f85b498fd5: Pull complete
9b233e420ac7: Pull complete
fe42347c4ecf: Pull complete
d14eb2ed1e17: Pull complete
66d98f73acb6: Pull complete
d2c43c5efbc8: Pull complete
ab590b48ea47: Pull complete
80692ae2d067: Pull complete
05e465aaa99a: Pull complete
Digest: sha256:c9d7e608f73832673479770d66aacc810001ec751d1905ff63fae3fe2e0ca6d
Status: Downloaded newer image for php:7.4-apache
11be1027fefba81933a4f4583ff19bf7e0b63775148f3f424c79e17883ef0876
root@ubuntu:/home/eric/Desktop#
```

- **Arrancar un contenedor llamado c2 sobre la imagen mariadb que monte el volumen_datos en la ruta /var/lib/mysql y cuya contraseña de root sea admin.**

A continuación arrancaremos en segundo plano un contenedor llamado **c2** sobre la imagen **mariadb**, y montaremos el volumen **volumen_datos** en la ruta **/var/lib/mysql** con la contraseña de **root** establecida como **"admin"** usando el siguiente comando

```
root@ubuntu:/home/eric/Desktop# docker run -d --name c2 -e MYSQL_ROOT_PASSWORD=admin -v volumen_datos:/var/lib/mysql mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
bccd10f490ab: Pull complete
d9d8e1823c6f: Pull complete
4b658f15686b: Pull complete
153080ffcddf: Pull complete
fc35f7aae1e5: Pull complete
59efd043a883: Pull complete
676a7ad9f737: Pull complete
335ef3100b9e: Pull complete
Digest: sha256:b5e508abc5d889425e90212541e30d29279b7ed34dd74bab5bb715b2f2aeeb7b
Status: Downloaded newer image for mariadb:latest
ec5571f76b01c7de524562db4a4ac6a5a1b5b164e4c0e571e9d2bd7df7a51e2b
root@ubuntu:/home/eric/Desktop#
```

- **Entra en la base de datos del contenedor c2 de la manera que prefieras (entrando en el contenedor o con un cliente remoto) y crea una nueva base de datos añadiendo posteriormente alguna tabla con info.**

Entraremos al contenedor **c2** mediante el comando **docker exec** y el usuario creado

```
root@ubuntu:/home/eric/Desktop# docker exec -it c2 mariadb --user root -padmin
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.3.2-MariaDB-1:11.3.2+maria-ubu2204 mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

Y creamos una nueva base de datos con alguna tabla con información

```
MariaDB [(none)]> CREATE DATABASE mi_bbdd;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> USE mi_bbdd;
Database changed
MariaDB [mi_bbdd]>
MariaDB [mi_bbdd]> CREATE TABLE nombres (
->   id INT AUTO_INCREMENT PRIMARY KEY,
->   nombre VARCHAR(255),
->   edad INT
-> );
Query OK, 0 rows affected (0.005 sec)

MariaDB [mi_bbdd]>
MariaDB [mi_bbdd]> INSERT INTO nombres (nombre, edad) VALUES ('Eric', 21), ('Mj', 25);
Query OK, 2 rows affected (0.002 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [mi_bbdd]>
```

Podemos ejecutar un **select** para probar que los datos se han metido correctamente

```
MariaDB [mi_bbdd]> USE mi_bbdd;
Database changed
MariaDB [mi_bbdd]> SELECT * FROM nombres;
+----+-----+-----+
| id | nombre | edad |
+----+-----+-----+
| 1  | Eric   | 21   |
| 2  | Mj     | 25   |
+----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [mi_bbdd]>
```

3.- Para y elimina el contenedor c2.

Para parar y eliminar dicho contenedor ejecutaremos un **docker stop** seguido de un **docker rm**, además ejecutaremos un **docker ps -a** para verificar que hemos borrado los contenedores

```
root@ubuntu:/home/eric/Desktop# docker stop c2
c2
root@ubuntu:/home/eric/Desktop# docker rm c2
c2
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
11be1027febf   php:7.4-apache "docker-php-entrypoi..." 27 minutes ago Up 27 minutes 80/tcp       c1
root@ubuntu:/home/eric/Desktop#
```

4.- Arrancarlo de nuevo usando el volumen_datos y comprueba que la base de datos que creaste tiene los datos anteriormente introducidos.

Arrancamos de nuevo el contenedor eliminado (creándolo en el momento) y volvemos a acceder con nuestro usuario para ejecutar la consulta en la base de datos creada comprobando que los datos anteriormente añadidos siguen dentro de la base de datos

```
root@ubuntu:/home/eric/Desktop# docker run -d --name c2 -e MYSQL_ROOT_PASSWORD=admin -v volumen_datos:/var/lib/mysql mariadb
e72792dd59193e4e1bca03c7fc56737299ac424962848c9dccc0ebdca3de1299f
root@ubuntu:/home/eric/Desktop# docker exec -it c2 mariadb --user root -padmin
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.3.2-MariaDB-1:11.3.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE mi_bbdd
Reading table information for completion of table and column names
You can turn off this feature with -A

Database changed
MariaDB [mi_bbdd]> SELECT * FROM nombres
-> ^C
MariaDB [mi_bbdd]> SELECT * FROM nombres;
+----+-----+-----+
| id | nombre | edad |
+----+-----+-----+
| 1  | Eric   | 21   |
| 2  | Mj     | 25   |
+----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [mi_bbdd]>
```

5.- Parar y borrar c1 y c2 y tras ello eliminar los dos volúmenes.

Por último, volvemos a ejecutar los comandos **docker stop** y **docker rm** para parar y eliminar los contenedores c1 y c2 y, para eliminar los volúmenes, ejecutamos un **docker volume rm** y posteriormente un **docker ps -a** para confirmar que los contenedores se han eliminado

```

root@ubuntu:/home/eric/Desktop# docker stop c1 c2
c1
c2
root@ubuntu:/home/eric/Desktop# docker rm c1 c2
c1
c2
root@ubuntu:/home/eric/Desktop# docker volume rm volumen_datos volumen_web
volumen_datos
volumen_web
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS       NAMES

```

También podemos ejecutar un **docker volume ls** para corroborar que los volúmenes también se han borrado

```

root@ubuntu:/home/eric/Desktop# docker volume ls
DRIVER      VOLUME NAME
local      1a9a588cda2502507d79f59e1f7b2fab3aa1858be5d182a61ef5ba72fd0ce003
root@ubuntu:/home/eric/Desktop#

```

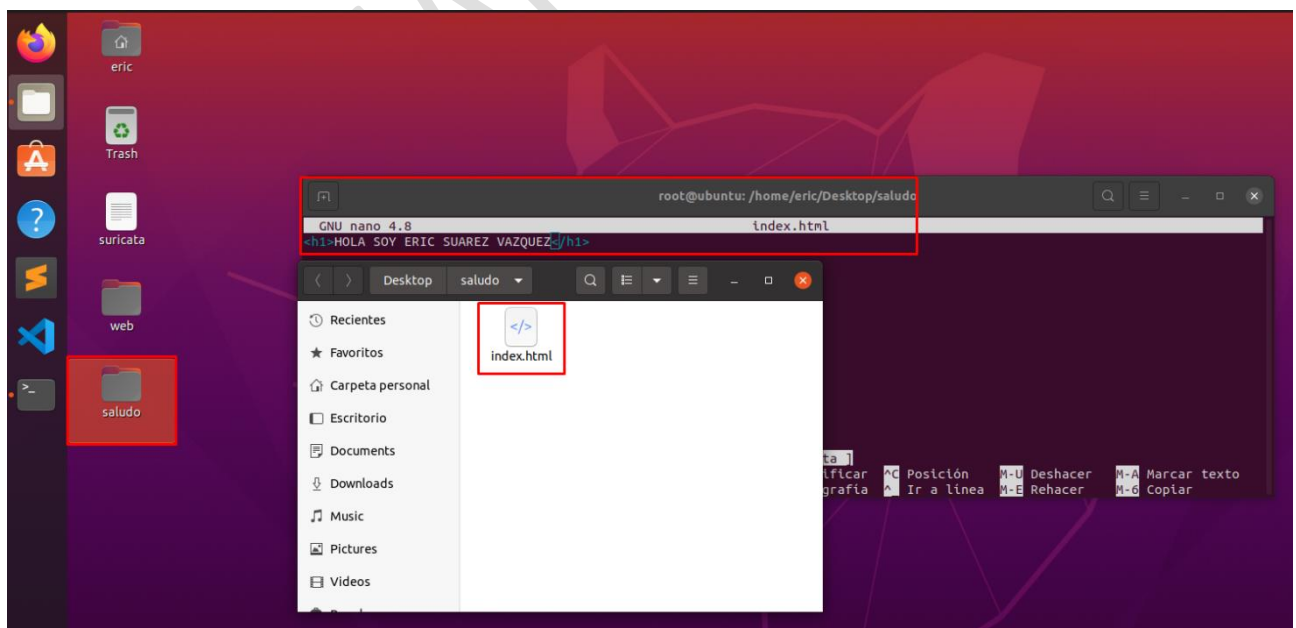
2. VOLÚMENES DE TIPO BIND MOUNT (2.5 puntos)

1.- Crea una carpeta llamada **saludo** y dentro de ella crea un fichero llamado **index.html** con el siguiente contenido:

`<h1>HOLA SOY X</h1>`

(Deberás sustituir esa X por tu nombre)

Crearemos una carpeta en el escritorio y dentro de esta un **index.html** con el contenido requerido



2.- Una vez hecho esto arrancar dos contenedores basados en la imagen php:7.4-apache que hagan un bind mount de la carpeta saludo en la carpeta /var/www/html del contenedor.

3.- Uno de ellos deberá redireccionar su puerto 80 al 8181 y el otro al 8282. Y sus nombres serán c1 y c2.

El apartado dos y tres los haremos juntos en dos comandos para levantar los contenedores, el c1 con el puerto 80 al 8181 y el otro al 8282

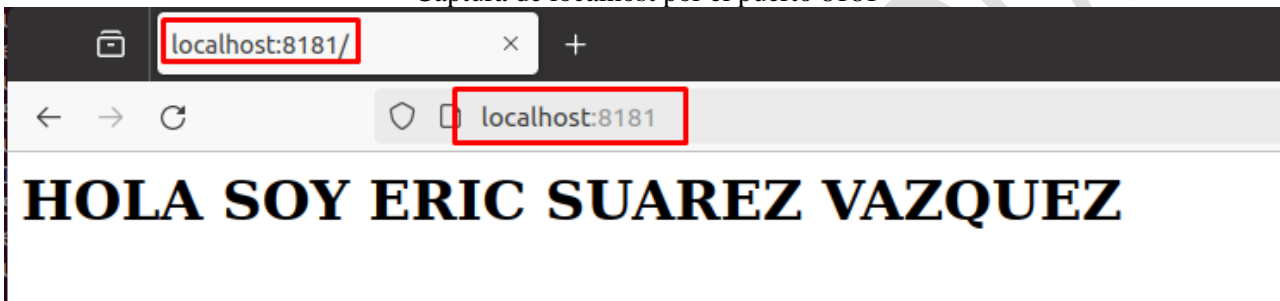
```
root@ubuntu:/home/eric/Desktop# docker run -d --name c1 -p 8181:80 -v $(pwd)/saludo:/var/www/html php:7.4-apache
af3e8e0dbad287f89fb62bb74c9cb3b2c30e0010b87940d48db02800941130a
root@ubuntu:/home/eric/Desktop# docker run -d --name c2 -p 8282:80 -v $(pwd)/saludo:/var/www/html php:7.4-apache
df20a5e3af1c6985ea3e04db27e68c808c2c1db1d7abbce77a3c3a16dbce463
root@ubuntu:/home/eric/Desktop# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
df20a5e3af1c	php:7.4-apache	"docker-php-entrypoi..."	7 seconds ago	Up 6 seconds	0.0.0.0:8282->80/tcp, :::8282->80/tcp	c2
af3e8e0dbad2	php:7.4-apache	"docker-php-entrypoi..."	19 seconds ago	Up 18 seconds	0.0.0.0:8181->80/tcp, :::8181->80/tcp	c1

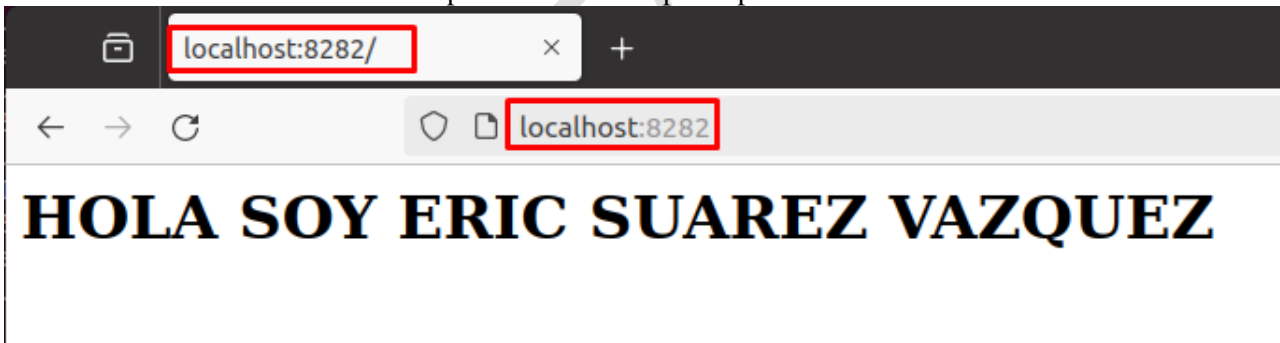
```
root@ubuntu:/home/eric/Desktop#
```

4.- Guardar captura de pantalla del escritorio del host con las dos páginas abiertas mostrando el index.html del c1 y del c2.

Captura de localhost por el puerto 8181



Captura de localhost por el puerto 8282



3. REDES (3 puntos)

1.- Crear dos redes de tipo (BRIDGE) con los siguientes datos:

RED1	RED2
Nombre: red1 Dirección de red: 172.28.0.0 Máscara de red: 255.255.0.0 Gateway: 172.28.0.1	Nombre: red2 El resto de los datos serán proporcionados automáticamente por Docker.

Para crear las dos redes necesarias, debemos ejecutar los siguientes comandos docker

```
root@ubuntu: /home/eric
root@ubuntu: /home/eric# docker network create --subnet=172.28.0.0/16 --gateway=172.28.0.1 red1
fa4b66f553f495184e2d476856e028dbc3a7e4bf53bd1fe5b2b20f909f1ecaae
root@ubuntu: /home/eric# docker network create red2
09ed7a1fd2f5805d7a52be59b51da2c82e8f5aea13a66244da0c96ea84a16f0c
root@ubuntu: /home/eric#
```

Los datos de la red1 lo implementamos nosotros, la red2 será proporcionada por Docker

2.- Poner en ejecución un contenedor de la imagen ubuntu:20.04 que tenga como hostname host1, como IP 172.28.0.10 y que esté conectado a la red1. Lo llamaremos c1. Utilizar un solo comando.

Entrar en ese contenedor e instalar la aplicación ping (apt update && apt install inetutils-ping) y el paquete iproute2.

Entramos a ese contenedor con el comando **docker exec** y descargamos la aplicación **ping** y el paquete **iproute2**

Para ejecutar un contenedor con todos esos datos en un solo comando, usaremos el siguiente:

```
root@ubuntu: /home/eric/Desktop# docker run -it --name c1 --hostname host1 --network red1 --ip 172.28.0.10 ubuntu:20.04 /bin/bash -c "apt-get update && apt-get install -y iputils-ping iproute2 && bash"
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3542 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [29.8 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1194 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [3476 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1489 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3634 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.5 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [4021 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
Fetched 30.9 MB in 16s (1926 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libatm1 libbsd0 libcapp2 libcapp2-bin libelf1 libmnl0 libpam-cap libxtables12
Suggested packages:
  iproute2-doc
The following NEW packages will be installed:
  iproute2 iputils-ping libatm1 libbsd0 libcapp2 libcapp2-bin libelf1 libmnl0 libpam-cap libxtables12
0 upgraded, 10 newly installed, 0 to remove and 9 not upgraded.
Need to get 1101 kB of archives.
After this operation, 3791 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libbsd0 amd64 0.10.0-1 [45.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcapp2 amd64 1:2.32-1ubuntu0.1 [15.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libelf1 amd64 0.176-1.1ubuntu0.1 [44.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 libmnl0 amd64 1.0.4-2 [12.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libxtables12 amd64 1.8.4-3ubuntu2.1 [28.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcapp2-bin amd64 1:2.32-1ubuntu0.1 [26.2 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/main amd64 iproute2 amd64 5.5.0-1ubuntu1 [858 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 iputils-ping amd64 3:20190709-3ubuntu1 [40.0 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/main amd64 libatm1 amd64 1:2.5.1-4 [21.8 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libpam-cap amd64 1:2.32-1ubuntu0.1 [8364 B]
Fetched 1101 kB in 1s (939 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libbsd0:amd64.
(Reading database ... 4124 files and directories currently installed.)
Preparing to unpack .../0-libbsd0_0.10.0-1_amd64.deb ...
Unpacking libbsd0:amd64 (0.10.0-1) ...
```

3.- Poner en ejecución un contenedor de la imagen ubuntu:20.04 que tenga como hostname host2 y que esté conectado a la red2. En este caso será docker el que le de una IP correspondiente a esa red. Lo llamaremos c2. Utilizar un solo comando.

Como hemos hecho anteriormente, ejecutaremos un contenedor con los datos pedidos en un solo comando

```
root@ubuntu:/home/eric# docker run -d --name c2 --hostname host2 --network red2 ubuntu:20.04
c2a19b150fed9a106872f236d5272fc729ac9fb3c6dcb62f1fc555415f4e1c87
root@ubuntu:/home/eric#
```

Entrar en ese contenedor e instalar la aplicación ping (apt update && apt install inetutils-ping) y el paquete iproute2.

Volveremos a hacer lo mismo que hemos hecho con el primer contenedor pero cambiando la red

```
root@ubuntu:/home/eric/Desktop# docker run -it --name c2 --hostname host2 --network red2 ubuntu:20.04 /bin/bash -c 'apt-get update && apt-get install -y iputils-ping iproute2'
&& bash
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [3476 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:9 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1489 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3634 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.5 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [4021 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3542 kB]
Get:15 http://security.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
Get:17 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [29.8 kB]
Get:18 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1194 kB]
Fetched 38.9 MB in 6s (5147 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libatm1 libbsd0 libcap2 libcap2-bin libelf1 libnsl0 libpam-cap libxtables12
Suggested packages:
  iproute2-doc
The following NEW packages will be installed:
  iproute2 iputils-ping libatm1 libbsd0 libcap2 libcap2-bin libelf1 libnsl0 libpam-cap libxtables12
0 upgraded, 10 newly installed, 0 to remove and 9 not upgraded.
Need to get 1101 kB of archives.
After this operation, 3791 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libbsd0 amd64 0.10.0-1 [45.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcap2 amd64 1:2.32-1ubuntu0.1 [15.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libelf1 amd64 0.176-1.1ubuntu0.1 [44.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 libnsl0 amd64 1.0.0-2 [12.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libxtables12 amd64 1.8.4-3ubuntu2.1 [28.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcap2-bin amd64 1:2.32-1ubuntu0.1 [26.2 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/main amd64 iproute2 amd64 5.5.0-1ubuntu1 [858 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 iputils-ping amd64 3:20190709-3ubuntu1 [40.0 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/main amd64 libatm1 amd64 1:2.5.1-4 [21.8 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libpam-cap amd64 1:2.32-1ubuntu0.1 [8364 B]
Fetched 1101 kB in 0s (2307 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libbsd0:amd64.
(Reading database ... 4124 files and directories currently installed.)
Preparing to unpack .../0-libbsd0_0.10.0-1_amd64.deb ...
```

4.- Comprueba la conectividad haciendo ping entre contenedores ¿Qué ocurre?

Con ambos contenedores levantados, entraremos en uno de ellos e intentaremos ejecutar un ping entre ellos

```
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
12572af9ac74   ubuntu:20.04   "/bin/bash -c 'apt-g..." About a minute ago    Up 4 seconds                c2
20db89f56e81   ubuntu:20.04   "/bin/bash -c 'apt-g..." 6 minutes ago       Up 6 minutes                c1
root@ubuntu:/home/eric/Desktop#
```

```
root@ubuntu:/home/eric/Desktop# docker exec -it c1 ping 172.28.0.1
PING 172.28.0.1 (172.28.0.1) 56(84) bytes of data.
64 bytes from 172.28.0.1: icmp_seq=1 ttl=64 time=0.165 ms
64 bytes from 172.28.0.1: icmp_seq=2 ttl=64 time=0.082 ms
64 bytes from 172.28.0.1: icmp_seq=3 ttl=64 time=0.106 ms
64 bytes from 172.28.0.1: icmp_seq=4 ttl=64 time=0.078 ms
64 bytes from 172.28.0.1: icmp_seq=5 ttl=64 time=0.099 ms
64 bytes from 172.28.0.1: icmp_seq=6 ttl=64 time=0.080 ms
64 bytes from 172.28.0.1: icmp_seq=7 ttl=64 time=0.069 ms
^C
--- 172.28.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6131ms
rtt min/avg/max/mdev = 0.069/0.097/0.165/0.030 ms
root@ubuntu:/home/eric/Desktop#
```

El ping entre ambas máquinas funciona perfectamente

5.- Aplica el cambio de configuración que estimes oportuno y explicarlo para que los contenedores puedan tener conectividad entre ellos.

Para que los contenedores puedan tener conectividad entre sí, necesitamos asegurarnos de que estén en la misma red y que las direcciones **IP** y las **configuraciones de red** sean correctas.

En este caso, hemos creado dos redes (**red1** y **red2**), cada una con su propia subred y puerta de enlace.

Luego, hemos conectado los contenedores **c1** y **c2** a estas redes respectivas. Al asignar direcciones **IP** específicas a los contenedores (**c1** con **172.28.0.10** en **red1**), **Docker** manejará automáticamente las rutas y la conectividad entre los contenedores dentro de la misma red.

Además, al instalar las herramientas de red necesarias en cada contenedor (**ping** e **iproute2**), permitimos que los contenedores puedan comunicarse entre sí mediante la resolución de nombres y la resolución de direcciones **IP**.

En mi caso **docker** me ha puesto la red2 en la misma que red1, si esto no fuera así, habría que conectar la red1 al contenedor c2 con el siguiente comando:

docker network connect red1 c2

Para así conectar la red al contenedor

4. SUBIDA DE IMAGENES PROPIAS A DOCKERHUB (2 puntos)

1.- A partir de la imagen base de ubuntu o debian que prefieras añadirle el paquete **iproute2**, el editor **nano** y el servicio **SSH** aparte de actualizar previamente los repositorios del sistema.

Cambia también el hostname de la máquina por uno que contenga tu nombre.

Vamos a ejecutar un **docker pull** a una imagen Ubuntu 18.04 al que añadiremos las herramientas que nos piden, como ya ejecuté un docker pull de un Ubuntu 18.04, no es necesario que haga otro, podemos ver que imágenes tenemos ejecutando un **docker images**

```
root@ubuntu:/home/eric/Desktop# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
mariadb             latest          bc6434c28e9a   6 weeks ago    405MB
ubuntu              20.04          3cff1c6ff37e   7 weeks ago    72.8MB
ubuntu              18.04          f9a80a55f492   10 months ago  63.2MB
php                 7.4-apache     20a3732f422b   17 months ago  453MB
danielkraic/asciitquarium latest          7bab964067d2   5 years ago    309MB
root@ubuntu:/home/eric/Desktop#
```

Con dicha imagen obtenida ejecutamos un **docker run** para crear un contenedor con el nombre que escojamos

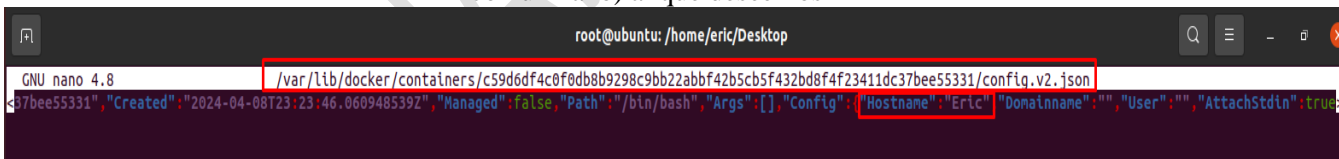
```
root@ubuntu:/home/eric/Desktop# docker run -it --name ubuntu_eric_container ubuntu:18.04 /bin/bash
root@c59d6df4c0f0:/#
```


Una vez dentro ejecutaremos un update e instalaremos el paquete `iproute2`, `nano` y el servicio SSH

```
root@c59d6df4c0f0:/home# apt update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [23.8 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1637 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [3373 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [1728 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [3786 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2411 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [30.8 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [20.6 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [64.0 kB]
Get:18 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1688 kB]
Fetched 28.1 MB in 5s (5290 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@c59d6df4c0f0:/home# apt install -y iproute2 nano openssl-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates dbus dmsetup file gir1.2-glib-2.0 krb5-locales libapparmor1 libargon2-0 libatm1 libbsd0 libcap2 libcryptsetup12 libdbus-1-3 libdevmapper1.02.1 libedit2
  libelf1 libexpat1 libgirepository-1.0-1 libglib2.0-0 libgssapi-krb5-2 libcub0 libidn11 libip4tc0 libjansson-2.0 libk5crypto3 libkeyutils1 libkmod2 libkrb5-3
  libkrb5support0 libmagic-mgc libmagic1 libmnl0 libmpdec2 libnss-systemd libpam-systemd libpsl5 libpython3-stdlib libpython3.6-minimal libpython3.6-stdlib libreadline7
  libsqlite3-0 libssl1.0.0 libssl1.1 libwrap0 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxml2 libxmuu1 libxtables12 mime-support multiarch-support
  ncurses-term networkd-dispatcher openssl-client openssl-sftp-server openssl-publicsuffix python3 python3-certifi python3-chardet python3-dbus python3-gi python3-idna
  python3-minimal python3-pkg-resources python3-requests python3-six python3-urllib3 python3.6 python3.6-minimal readline-common shared-mime-info ssh-import-id systemd
  systemd-sysv ucf wget xauth xdg-user-dirs xz-utils
Suggested packages:
  default-dbus-session-bus | dbus-session-bus iproute2-doc krb5-doc krb5-user spell iw | wireless-tools keychain libpam-ssh monkeysphere ssh-askpass molly-guard rssh ufw
  python3-doc python3-tk python3-venv python-dbus-doc python3-dbus-dbg python3-setuputils python3-cryptography python3-openssl python3-socks python3.6-venv python3.6-doc
  binutils binfmt-support readline-doc systemd-container policykit-1
The following NEW packages will be installed:
  ca-certificates dbus dmsetup file gir1.2-glib-2.0 iproute2 krb5-locales libapparmor1 libargon2-0 libatm1 libbsd0 libcap2 libcryptsetup12 libdbus-1-3 libdevmapper1.02.1
  libedit2 libelf1 libexpat1 libgirepository-1.0-1 libglib2.0-0 libgssapi-krb5-2 libidn11 libip4tc0 libjansson-2.0 libk5crypto3 libkeyutils1 libkmod2
  libkrb5-3 libkrb5support0 libmagic-mgc libmagic1 libmnl0 libmpdec2 libnss-systemd libpam-systemd libpsl5 libpython3-stdlib libpython3.6-minimal libpython3.6-stdlib
  libreadline7 libsqlite3-0 libssl1.0.0 libssl1.1 libwrap0 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxml2 libxmuu1 libxtables12 mime-support
  multiarch-support nano ncurses-term networkd-dispatcher openssl-client openssl-server openssl-sftp-server openssl-publicsuffix python3 python3-certifi python3-chardet
  python3-dbus python3-gi python3-idna python3-minimal python3-pkg-resources python3-requests python3-six python3-urllib3 python3.6 python3.6-minimal readline-common
  shared-mime-info ssh-import-id systemd systemd-sysv ucf wget xauth xdg-user-dirs xz-utils
0 upgraded, 88 newly installed, 0 to remove and 1 not upgraded.
```

Esto puede tardar unos minutos

Cambiar el hostname puede ser lioso, así que vamos poco a poco, primero debemos parar el docker con **service docker stop** ya que ahora nos dirigiremos a la configuración JSON del nuestro contenedor diferenciado por su **ID** y lo abriremos, buscamos el hostname en el archivo **JSON** y lo cambiamos (entrando con un `nano`) al que deseemos



Tras ello, volvemos a activar docker, levantamos el contenedor y entramos en él y vemos nuestro hostname de nuestro contenedor

Una vez terminemos, ejecutamos un `docker commit` para crear una imagen de Ubuntu 18.04 con nuestros cambios diseñados en nuestro contenedor

```
root@ubuntu:/home/eric/Desktop# docker commit ubuntu_eric_container eric/ubuntu-ssh:18.04
sha256:c7147760e232a226c8a8be8712d0697776a1be5f03151f3d3c307fec218a0c551
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
c59d6df4c0f0   ubuntu:18.04   "/bin/bash"             46 minutes ago   Up 25 minutes               ubuntu_eric_container
12572af9ac74   ubuntu:20.04   "/bin/bash -c 'apt-g..."   About an hour ago   Exited (0) 30 minutes ago   c2
20db89f56e81   ubuntu:20.04   "/bin/bash -c 'apt-g..."   About an hour ago   Exited (0) 30 minutes ago   c1
root@ubuntu:/home/eric/Desktop# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
eric/ubuntu-ssh   18.04     c7147760e232   15 seconds ago   226MB
marladb        latest    bc6434c28e9a   6 weeks ago     405MB
ubuntu         20.04     3cfff1c6ff37e   7 weeks ago     72.8MB
ubuntu         18.04     f9a80a55f492   10 months ago   63.2MB
php            7.4-apache  20a3732f422b   17 months ago   453MB
danielkratc/asciitquarium   latest    7bab964067d2   5 years ago     309MB
root@ubuntu:/home/eric/Desktop#
```

2.- Comprueba que puedes acceder al contenedor mediante SSH sin redigir los puertos al host. Muestra el comando docker con el que lanzas el contenedor. Realiza la conexión mediante algún cliente SSH de tu elección o desde la terminal directamente.

Crearemos un nuevo contenedor con la nuestra imagen anteriormente creada

```
root@ubuntu:/home/eric/Desktop# docker run -d --name ericssh eric/ubuntu-ssh:18.04
7936079f07fbf16365147c6f642f97feb51c8f600228556c78867df9173be
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
7936079f07fb   eric/ubuntu-ssh:18.04   "/bin/bash"             57 seconds ago   Exited (0) 56 seconds ago           ericssh
c59d6df4c0f0   ubuntu:18.04         "/bin/bash"             50 minutes ago   Up 28 minutes                       ubuntu_eric_container
12572af9ac74   ubuntu:20.04         "/bin/bash -c 'apt-g..."   About an hour ago   Exited (0) 34 minutes ago           c2
20db89f56e81   ubuntu:20.04         "/bin/bash -c 'apt-g..."   About an hour ago   Exited (0) 34 minutes ago           c1
root@ubuntu:/home/eric/Desktop#
```

Levantamos un contenedor al que buscaremos su IP mediante el siguiente comando

```
root@be4a496f4ec4:/# docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' ericssh
172.17.0.3
root@ubuntu:/home/eric#
```

Ya conociendo la IP y nuestro usuario/hostname que es Eric, podemos conectarnos mediante SSH a nuestro contenedor encendiendo el servicio SSH

```
root@be4a496f4ec4:/# service ssh status
* sshd is running
root@be4a496f4ec4:/#
eric@be4a496f4ec4: ~
root@ubuntu:/home/eric# ssh eric@172.17.0.3
eric@172.17.0.3's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Tue Apr  9 00:23:23 2024 from 172.17.0.1
eric@be4a496f4ec4:~$
```

3.- Guarda la imagen y subela a tu dockerhub personal con los comandos pertinentes.

Para guardar nuestra imagen “eric/ubuntu-ssh:18.04” debemos usar el comando **docker tag** primero para poner el tag de la imagen a latest

```
root@ubuntu:/home/eric/Desktop# docker tag eric/ubuntu-ssh:18.04 eric/ubuntu-ssh:latest
root@ubuntu:/home/eric/Desktop# docker images
REPOSITORY    TAG        IMAGE ID        CREATED        SIZE
eric/ubuntu-ssh   18.04     c7147760e232   21 minutes ago   226MB
eric/ubuntu-ssh   latest     c7147760e232   21 minutes ago   226MB
mariadb        latest     bc6434c28e9a   6 weeks ago     405MB
ubuntu         20.04     3cff1c6ff37e   7 weeks ago     72.8MB
ubuntu         18.04     f9a80a55f492   10 months ago   63.2MB
php            7.4-apache 20a3732f422b   17 months ago   453MB
danielkraic/asciiquarium   latest     7bab964067d2   5 years ago     309MB
root@ubuntu:/home/eric/Desktop#
```

Ahora debemos ejecutar un docker login e iniciar sesión con nuestras credenciales

```
root@ubuntu:/home/eric/Desktop# docker login -u ericsuarezv
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ubuntu:/home/eric/Desktop#
```

ATENCIÓN, EL NOMBRE DE LA IMAGEN DEBE SER IGUAL A TU NOMBRE DE DOCKER HUB, SI NO, NO SE PODRÁ SUBIR

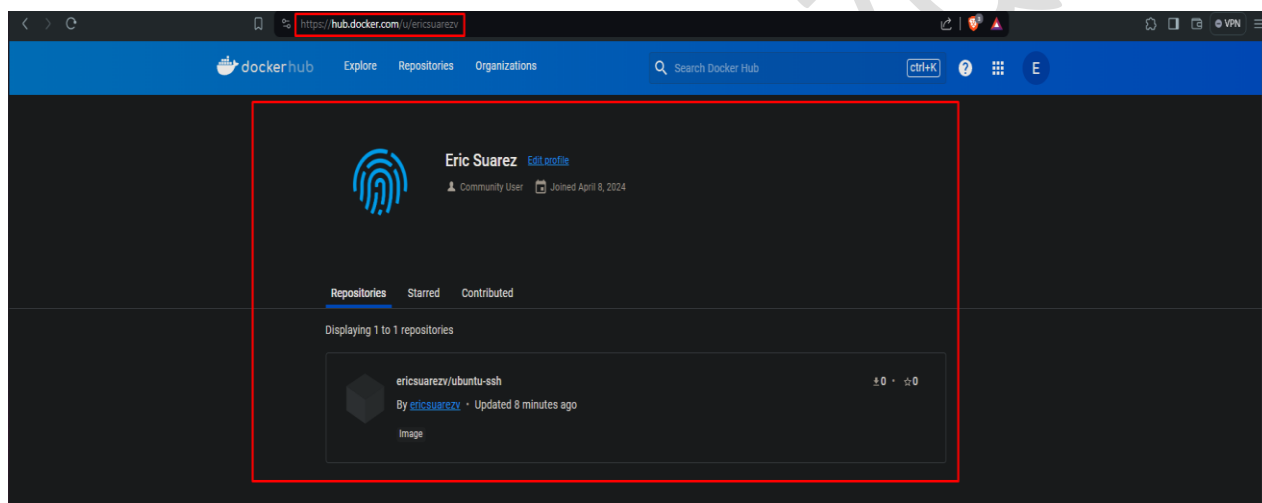
EJEMPLO: docker push usuarioDockerHub/nombreDeLaImagen:Version

En mi caso yo tuve que ejecutar otro docker tag para poner mi usuario de docker hub

```

root@ubuntu:/home/eric/Desktop# docker tag eric/ubuntu-ssh:latest ericsuarezv/ubuntu-ssh:latest
root@ubuntu:/home/eric/Desktop# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
eric/ubuntu-ssh     18.04              c7147760e232       31 minutes ago     226MB
eric/ubuntu-ssh     latest             c7147760e232       31 minutes ago     226MB
ericsuarezv/ubuntu-ssh latest             c7147760e232       31 minutes ago     226MB
mariadb             latest             bc6434c28e9a       6 weeks ago        405MB
ubuntu              20.04             3cfff1c6ff37e       7 weeks ago        72.8MB
ubuntu              18.04             f9a80a55f492       10 months ago      63.2MB
php                 7.4-apache        20a3732f422b       17 months ago      453MB
danielkraic/asciiaquarium latest             7bab964067d2       5 years ago        309MB
root@ubuntu:/home/eric/Desktop# docker push ericsuarezv/ubuntu-ssh:latest
The push refers to repository [docker.io/ericsuarezv/ubuntu-ssh]
90730c03adc7: Pushed
548a79621a42: Pushed
latest: digest: sha256:345cdde47c52c3e603a56d6cb5f5ba56de3f6f5e389812cf85e2834b2946db38 size: 741
root@ubuntu:/home/eric/Desktop#
  
```

Comprobamos en la página web que se ha pusheado con éxito



4.- Elimina el contenedor y la imagen y vuelve a poner el marcha el contenedor tirando de la imagen personalizada de espacio en dockerhub con un solo comando.

Esto en un solo comando no se puede hacer, o como mucho, se podría concatenando comando, cosa que es una muy mala práctica a la hora de borrar y crear archivos (o contenedores e imágenes) con el mismo nombre.

Primero lo haremos con comandos separados, borraremos los contenedores

```
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
c59d6df4c0f0   ubuntu:18.04   "/bin/bash"             2 hours ago   Exited (0)    23 seconds ago          ubuntu_eric_container
12572af9ac74   ubuntu:20.04   "/bin/bash -c 'apt-g..." 2 hours ago   Exited (0)    About an hour ago       c2
20db89f56e81   ubuntu:20.04   "/bin/bash -c 'apt-g..." 2 hours ago   Exited (0)    About an hour ago       c1
root@ubuntu:/home/eric/Desktop# docker rm ubuntu_eric_container
root@ubuntu:/home/eric/Desktop# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
12572af9ac74   ubuntu:20.04   "/bin/bash -c 'apt-g..." 2 hours ago   Exited (0)    About an hour ago       c2
2 Ayuda 56e81   ubuntu:20.04   "/bin/bash -c 'apt-g..." 2 hours ago   Exited (0)    About an hour ago       c1
root@ubuntu:/home/eric/Desktop#
```

Posteriormente borramos las imágenes

```
root@ubuntu:/home/eric/Desktop# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
eric/ubuntu-ssh 18.04     c7147760e232  46 minutes ago 226MB
eric/ubuntu-ssh latest     c7147760e232  46 minutes ago 226MB
ericsuarezv/ubuntu-ssh latest     c7147760e232  46 minutes ago 226MB
mariadb        latest     bc6434c28e9a  6 weeks ago    405MB
ubuntu         20.04     3cfff1c6ff37e 7 weeks ago    72.8MB
ubuntu         18.04     f9a80a55f492 10 months ago 63.2MB
php            7.4-apache 20a3732f422b 17 months ago 453MB
danielkraic/asciiquarium latest     7bah964067d2 5 years ago    309MB
root@ubuntu:/home/eric/Desktop# docker rmi eric/ubuntu-ssh:18.04 eric/ubuntu-ssh:latest ericsuarezv/ubuntu-ssh:latest
Untagged: eric/ubuntu-ssh:18.04
Untagged: eric/ubuntu-ssh:latest
Untagged: ericsuarezv/ubuntu-ssh:latest
Deleted: sha256:c7147760e232a226c8a8be8712d069776a1be5f03151f3dc307fec218a0c551
Deleted: sha256:611df43dad44f45d39037abf68543c5742ec26c5e6e81316a714b24b370ce15
root@ubuntu:/home/eric/Desktop# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mariadb        latest     bc6434c28e9a  6 weeks ago    405MB
ubuntu         20.04     3cfff1c6ff37e 7 weeks ago    72.8MB
ubuntu         18.04     f9a80a55f492 10 months ago 63.2MB
php            7.4-apache 20a3732f422b 17 months ago 453MB
danielkraic/asciiquarium latest     7bah964067d2 5 years ago    309MB
root@ubuntu:/home/eric/Desktop#
```

Y ahora pones en marcha la descarga del contenedor con la imagen propia de tu cuenta de **Docker Hub**

```
root@ubuntu:/home/eric/Desktop# docker run -d --name ericssh ericsuarezv/ubuntu-ssh
Unable to find image 'ericsuarezv/ubuntu-ssh:latest' locally
latest: Pulling from ericsuarezv/ubuntu-ssh
41af1b5f0f51: Already exists
0e02a4991962: Pull complete
Digest: sha256:345cdde47c52c3e603a56d6cb5f5ba56de3f6f5e389812cf85e2834b2946db38
Status: Downloaded newer image for ericsuarezv/ubuntu-ssh:latest
a367f0e1a353314fb9ea4c83a0c53048e1d1a47bae84fa4dc4a11e5cb321b78c
root@ubuntu:/home/eric/Desktop#
```

Como he dicho antes, para ejecutar todo de un solo comando, habría que concatenarlos de esta manera:

```
docker rm ubuntu_eric_container ||
docker rmi eric/ubuntu-ssh:18.04 eric/ubuntu-ssh:latest ericsuarezv/ubuntu-ssh:latest ||
docker run -d --name ericssh ericsuarezv/ubuntu-ssh
```

5.- Comprueba que efectivamente el contenedor tiene el software instalado iniciando una nueva sesión por SSH a la máquina.

Volvemos a intentar conectarnos por SSH, ya que además a la hora de guardar la imagen lo dejamos encendido, y probamos con el usuario que creamos anteriormente, que era Eric, y como podemos observar, hemos entrado correctamente

```
root@ubuntu:/home/eric# ssh eric@172.17.0.2
eric@172.17.0.2's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

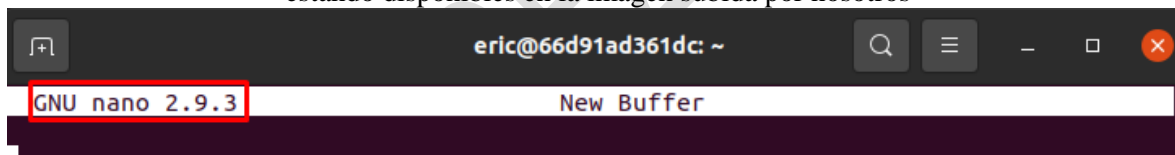
To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

eric@66d91ad361dc:~$
```

Además podemos ejecutar un nano para corroborar que los cambios y herramientas que añadimos siguen estando disponibles en la imagen subida por nosotros

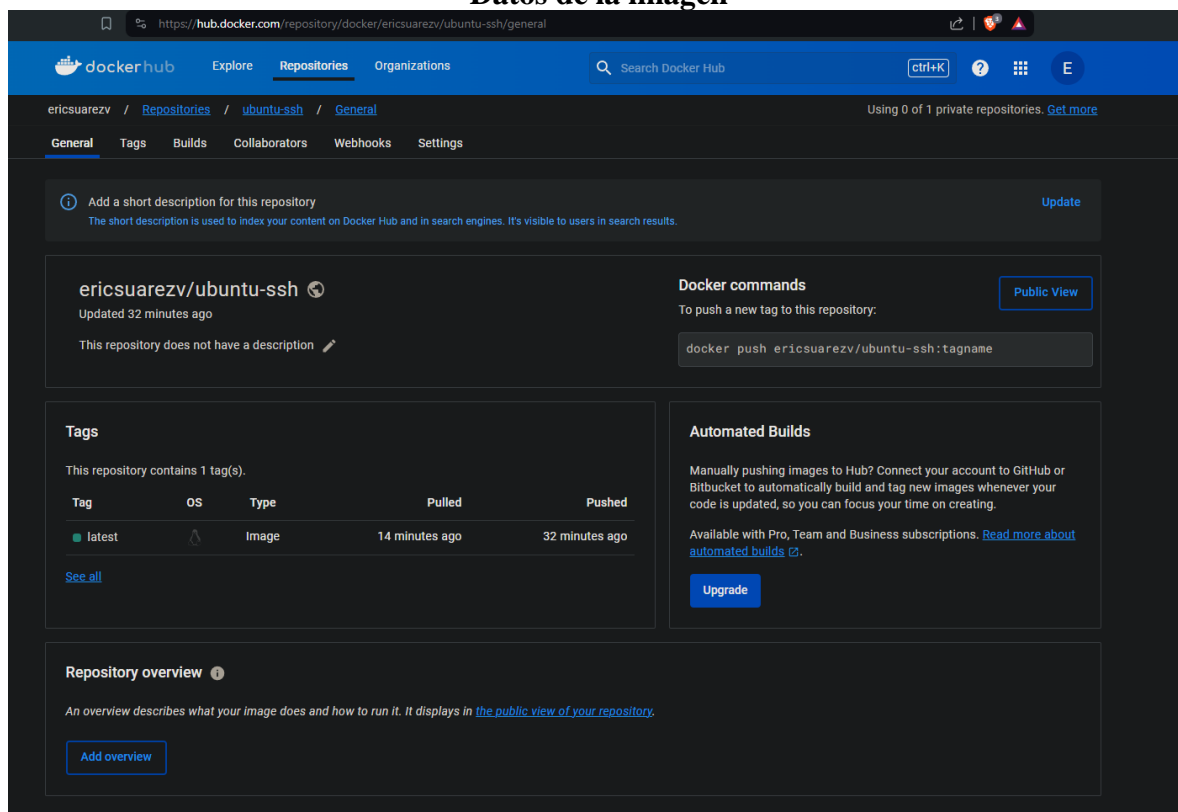


Explica los pasos con capturas de pantalla y además adjunta tu usuario y el nombre de la imagen.

Usuario de DockerHub: <https://hub.docker.com/u/ericsuarezv>

Nombre de la imagen: <https://hub.docker.com/r/ericsuarezv/ubuntu-ssh>

Datos de la imagen



The screenshot shows the Docker Hub repository page for `ericsuarezv/ubuntu-ssh`. The page is in the "General" tab. It includes a description field, a "Docker commands" section with a public view button, a "Tags" section with a table of tags, an "Automated Builds" section, and a "Repository overview" section.

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	linux	Image	14 minutes ago	32 minutes ago

[See all](#)

Repository overview

An overview describes what your image does and how to run it. It displays in [the public view of your repository](#).

[Add overview](#)