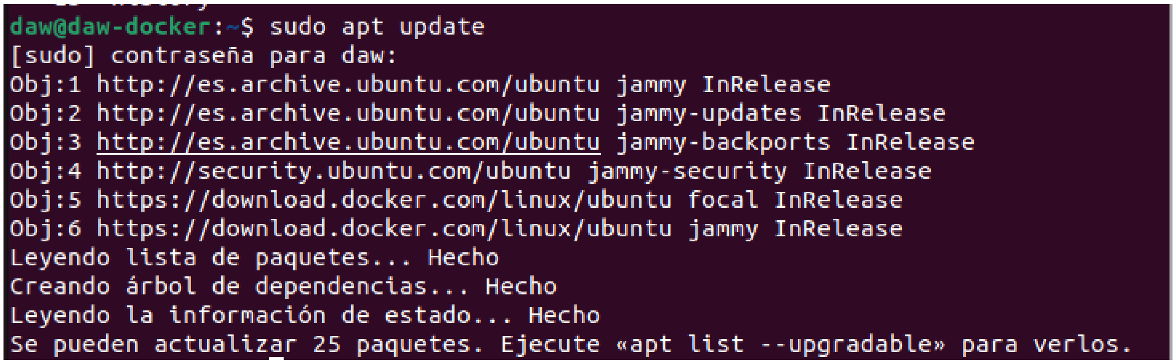
Tarea Docker 1

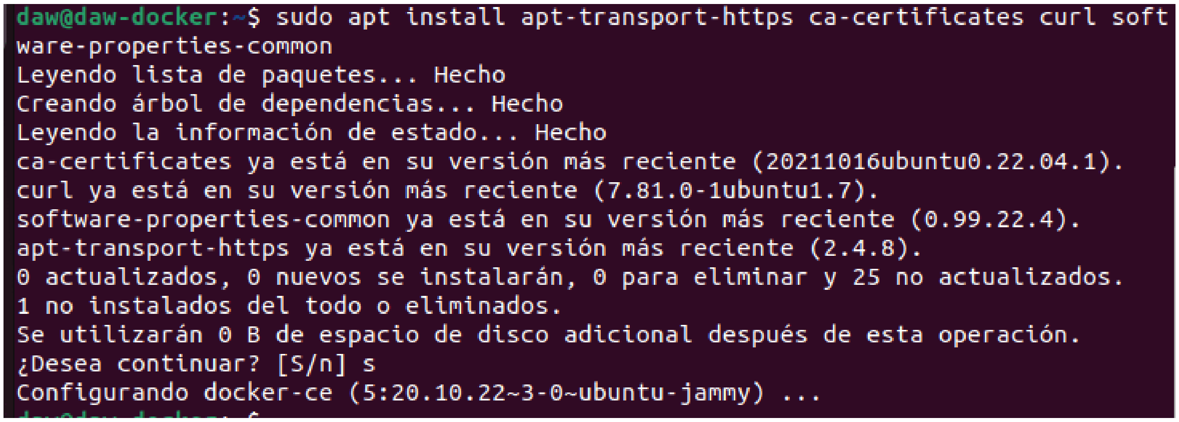
1. Instalamos docker, a través del tutorial DigitalOcean, siguiendo los pasos para la correcta instalación

```
sudo apt update
```



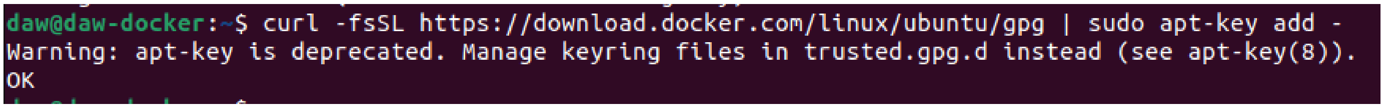
```
daw@daw-docker:~$ sudo apt update
[sudo] contraseña para daw:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:5 https://download.docker.com/linux/ubuntu focal InRelease
Obj:6 https://download.docker.com/linux/ubuntu jammy InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 25 paquetes. Ejecute «apt list --upgradable» para verlos.
```

```
sudo apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```



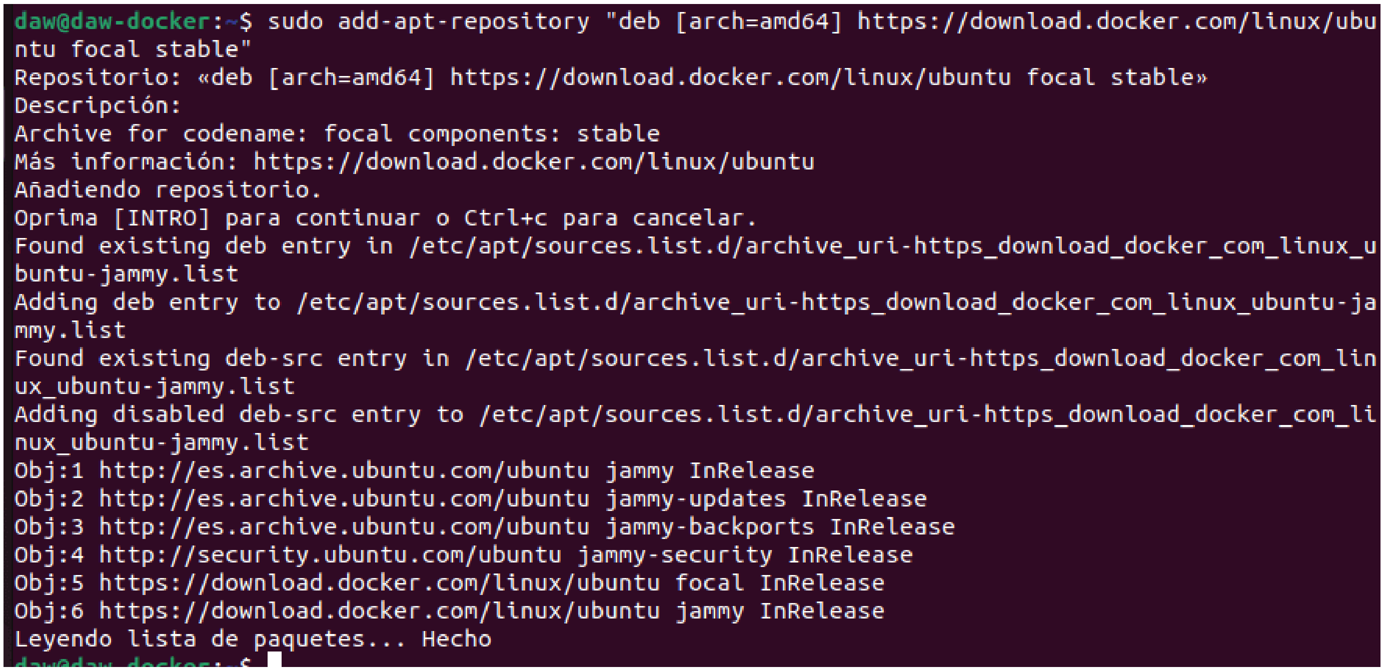
```
daw@daw-docker:~$ sudo apt install apt-transport-https ca-certificates curl soft
ware-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20211016ubuntu0.22.04.1).
curl ya está en su versión más reciente (7.81.0-1ubuntu1.7).
software-properties-common ya está en su versión más reciente (0.99.22.4).
apt-transport-https ya está en su versión más reciente (2.4.8).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 25 no actualizados.
1 no instalados del todo o eliminados.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Configurando docker-ce (5:20.10.22~3-0~ubuntu-jammy) ...
```

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```



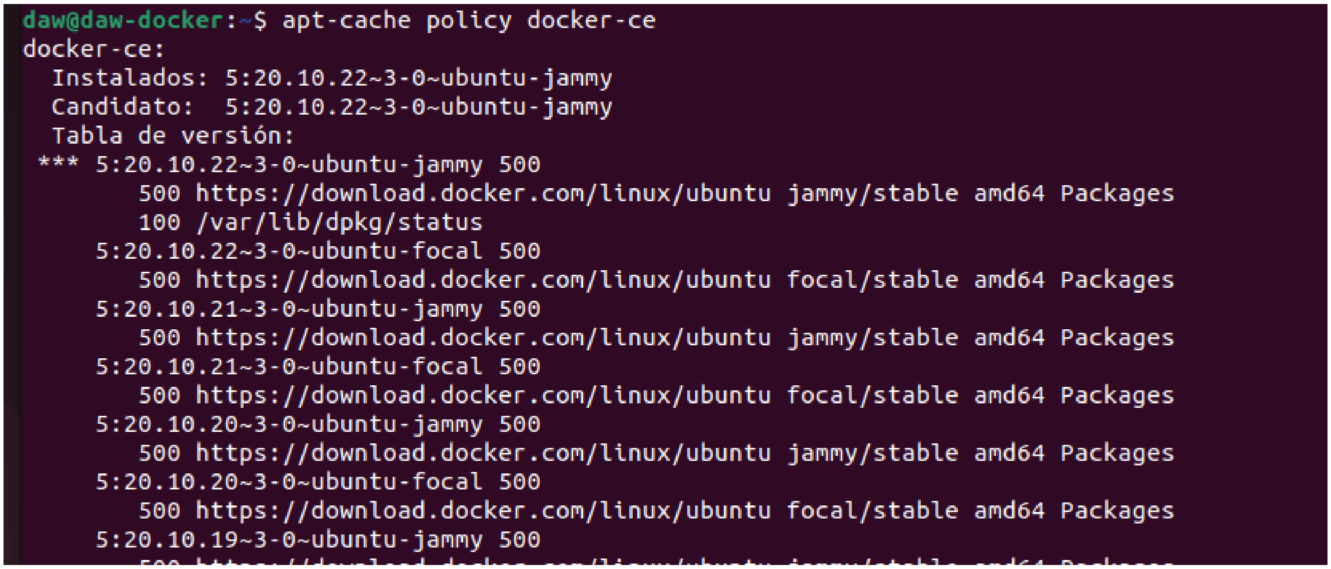
```
daw@daw-docker:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian
$(lsb_release -cs) stable"
```



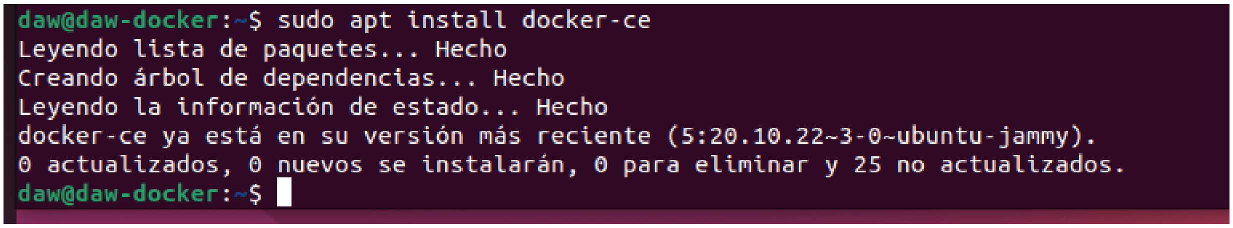
```
daw@daw-docker:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubu
ntu focal stable"
Repositorio: «deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable»
Descripción:
Archive for codename: focal components: stable
Más información: https://download.docker.com/linux/ubuntu
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Found existing deb entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_u
buntu-jammy.list
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-ja
mmy.list
Found existing deb-src entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_lin
ux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_li
nux_ubuntu-jammy.list
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:5 https://download.docker.com/linux/ubuntu focal InRelease
Obj:6 https://download.docker.com/linux/ubuntu jammy InRelease
Leyendo lista de paquetes... Hecho
daw@daw-docker:~$
```

```
apt-cache policy docker-ce
```



```
daw@daw-docker:~$ apt-cache policy docker-ce
docker-ce:
  Instalados: 5:20.10.22~3-0~ubuntu-jammy
  Candidato:  5:20.10.22~3-0~ubuntu-jammy
  Tabla de versión:
 *** 5:20.10.22~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
        100 /var/lib/dpkg/status
     5:20.10.22~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.21~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.21~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.20~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
     5:20.10.20~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
     5:20.10.19~3-0~ubuntu-jammy 500
        500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
```

```
sudo apt install docker-ce
```



```
daw@daw-docker:~$ sudo apt install docker-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
docker-ce ya está en su versión más reciente (5:20.10.22~3-0~ubuntu-jammy).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 25 no actualizados.
daw@daw-docker:~$
```

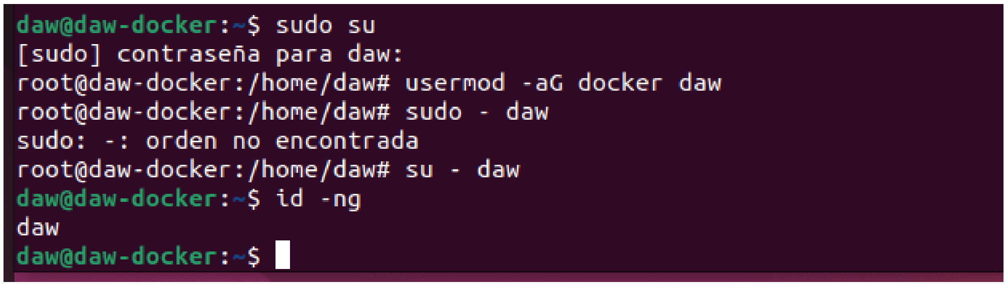
Creamos un usuario sin privilegios

```
sudo usermod -aG docker daw
```

```
su - daw
```

Comprobamos que se añadio el usurio al grupo docker

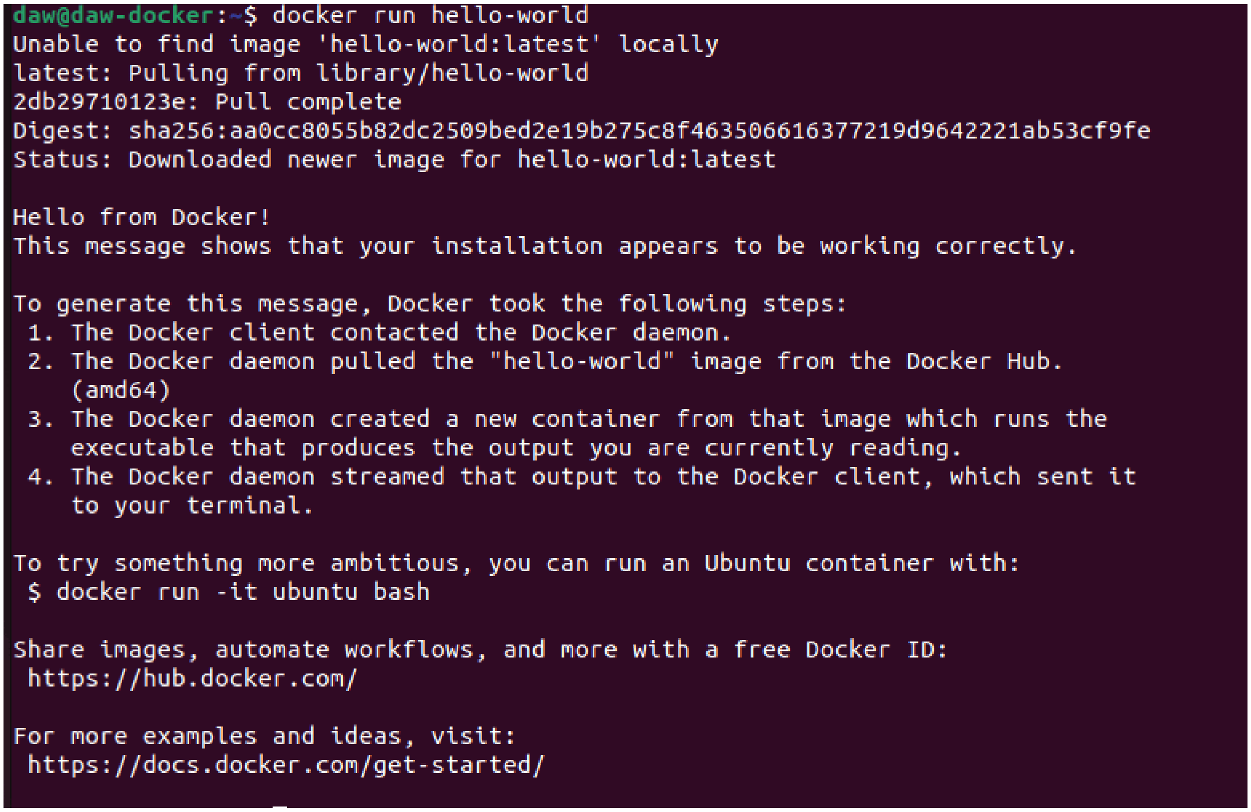
```
id -nG
```



```
daw@daw-docker:~$ sudo su
[sudo] contraseña para daw:
root@daw-docker:/home/daw# usermod -aG docker daw
root@daw-docker:/home/daw# sudo - daw
sudo: -: orden no encontrada
root@daw-docker:/home/daw# su - daw
daw@daw-docker:~$ id -ng
daw
daw@daw-docker:~$ 
```

2. Ejecutamos el contenedor desde la imagen hello-world

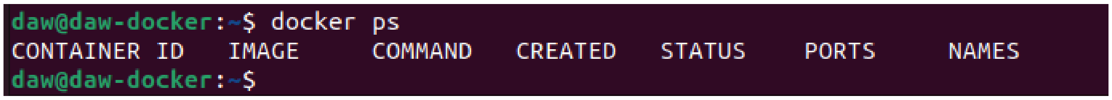
```
docker run hello-world
```



```
daw@daw-docker:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:aa0cc8055b82dc2509bed2e19b275c8f463506616377219d9642221ab53cf9fe
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/
```

Los contenedores en ejecución

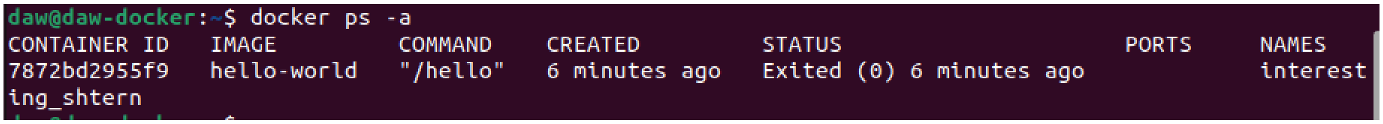
```
docker ps
```



```
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
daw@daw-docker:~$
```

Muestra todos los contenedores, tanto los que se están ejecutando como los que están parados.

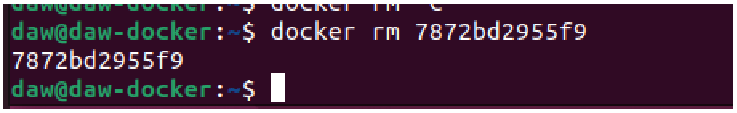
```
docker ps -a
```



```
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAMES
7872bd2955f9   hello-world   "/hello"   6 minutes ago   Exited (0) 6 minutes ago             interest
ing_shtern
```

Borramos el contenedor a través de su id

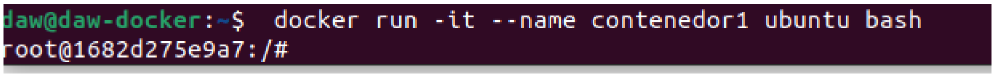
```
docker rm 7872bd2955f9
```



```
daw@daw-docker:~$ docker rm 7872bd2955f9
7872bd2955f9
daw@daw-docker:~$
```

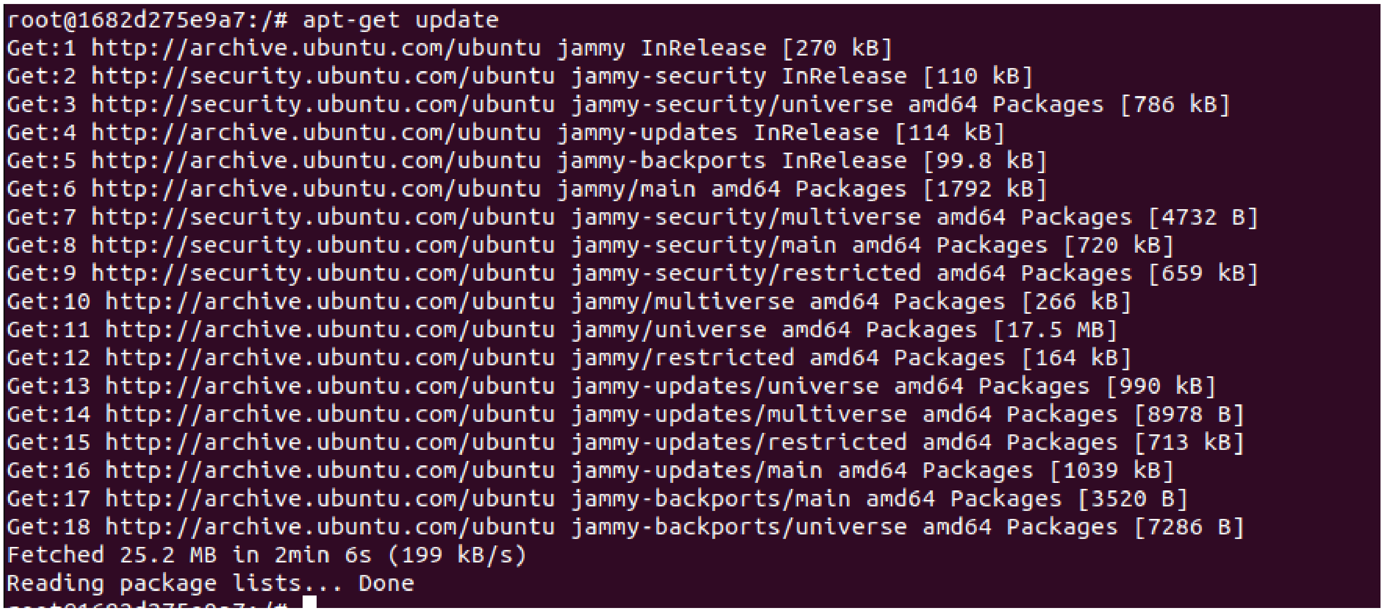
3. Creamos un contenedor a partir de la imagen, ubuntu. Actualizamos el contenedor e instalamos el editor de texto nano, ya que no viene instalado.

```
docker run -it --name contenedor1 ubuntu bash
```



```
daw@daw-docker:~$  docker run -it --name contenedor1 ubuntu bash
root@1682d275e9a7:/#
```

```
docker apt-get update
```

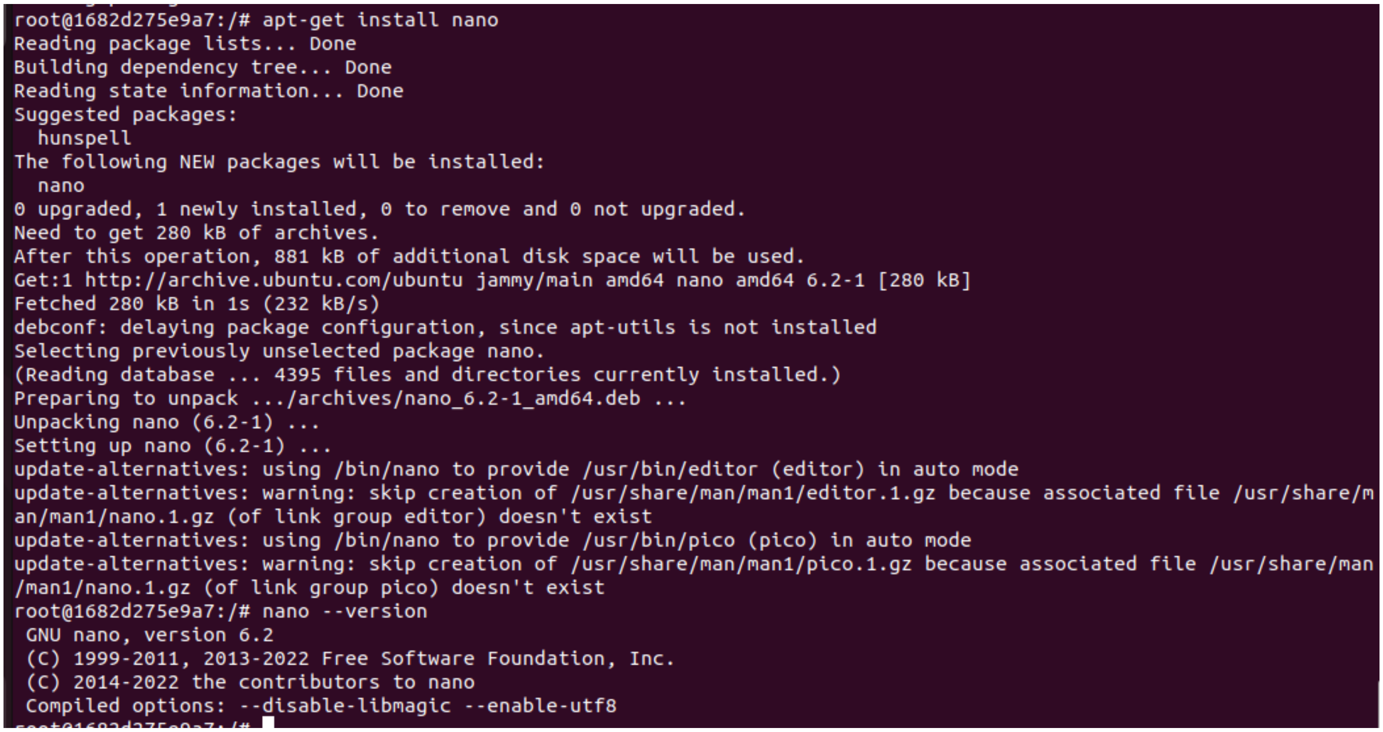


```
root@1682d275e9a7:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [786 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [720 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [659 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [990 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8978 B]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [713 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1039 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7286 B]
Fetched 25.2 MB in 2min 6s (199 kB/s)
Reading package lists... Done
```

```
docker apt-get install nano
```

Comprobamos que se instaló y la version

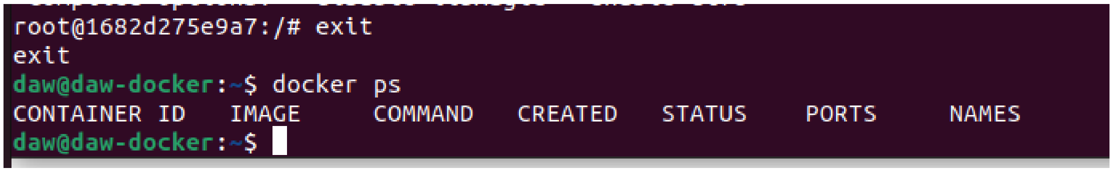
```
docker nano --version
```



```
root@1682d275e9a7:/# apt-get install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 280 kB of archives.
After this operation, 881 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 nano amd64 6.2-1 [280 kB]
Fetched 280 kB in 1s (232 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 4395 files and directories currently installed.)
Preparing to unpack .../archives/nano_6.2-1_amd64.deb ...
Unpacking nano (6.2-1) ...
Setting up nano (6.2-1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/m
an/man1/nano.1.gz (of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man
/man1/nano.1.gz (of link group pico) doesn't exist
root@1682d275e9a7:/# nano --version
 GNU nano, version 6.2
 (C) 1999-2011, 2013-2022 Free Software Foundation, Inc.
 (C) 2014-2022 the contributors to nano
 Compiled options: --disable-libmagic --enable-utf8
```

Salimos y comprobamos los contenedores que están en marcha, que en este caso es ninguno ya que acabamos de salir del contenedor y este se para automaticamente cuando salimos de él.

```
docker ps
```



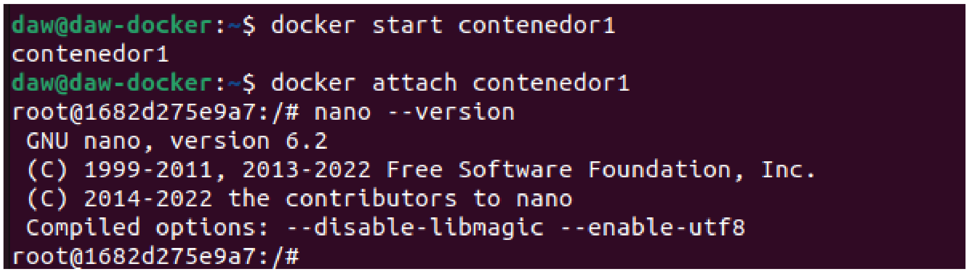
```
root@1682d275e9a7:/# exit
exit
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
daw@daw-docker:~$ 
```

Volvemos a acceder al contenedor, primero tenemos que inicializarlos y luego acceder, y comprobamos y confirmamos que el editor de texto, nano, sigue instalado.

```
docker start contenedor1
```

```
docker attach contenedor1
```

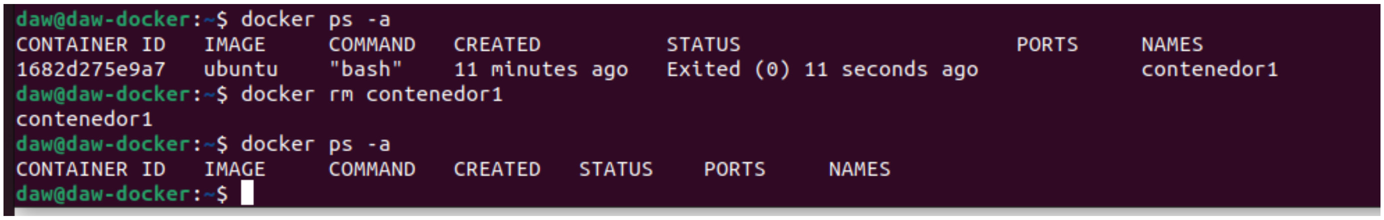
```
docker nano --version
```



```
daw@daw-docker:~$ docker start contenedor1
contenedor1
daw@daw-docker:~$ docker attach contenedor1
root@1682d275e9a7:/# nano --version
 GNU nano, version 6.2
 (C) 1999-2011, 2013-2022 Free Software Foundation, Inc.
 (C) 2014-2022 the contributors to nano
 Compiled options: --disable-libmagic --enable-utf8
root@1682d275e9a7:/#
```

Salimos del contenedor y eliminamos el contenedor. Se eliminar a traves de su ID o del nombre que le hayamos dado.

```
docker rm contenedor1
```

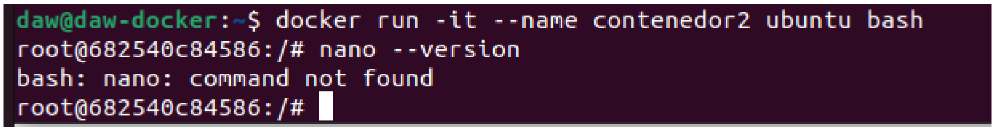


```
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED          STATUS                      PORTS     NAMES
1682d275e9a7   ubuntu    "bash"    11 minutes ago   Exited (0) 11 seconds ago             contenedor1
daw@daw-docker:~$ docker rm contenedor1
contenedor1
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
daw@daw-docker:~$ 
```

Creamos otro contenedor interactivo desde la misma imagen, ubuntu. Le llamaremos contenedor2. Y comprobamos que no tiene editor de texto, ya que la imagen no viene con ningún editor de texto instalado.

```
docker run -it --name contenedor2 ubuntu bash
```

```
docker nano --version
```

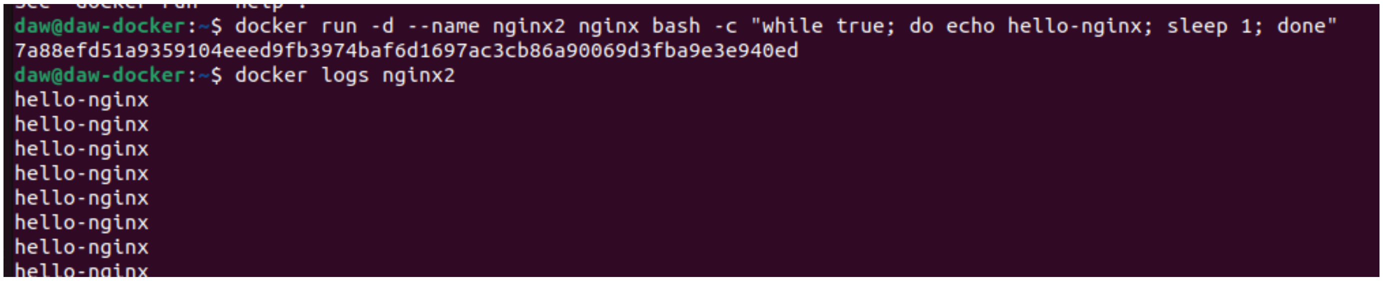


```
daw@daw-docker:~$ docker run -it --name contenedor2 ubuntu bash
root@682540c84586:/# nano --version
bash: nano: command not found
root@682540c84586:/# 
```

4. Creamos un contenedor demonio que haremos en segundo plano '-d' lo que hara que este en funcionamiento siempre a no ser que lo paremos y con un comando de ejecución 'bash -c' Nginx2 y mostramos los logs

```
docker run -d--name nginx2 nginx bash-c"while true; do echo hello-nginx; sleep 1; done
```

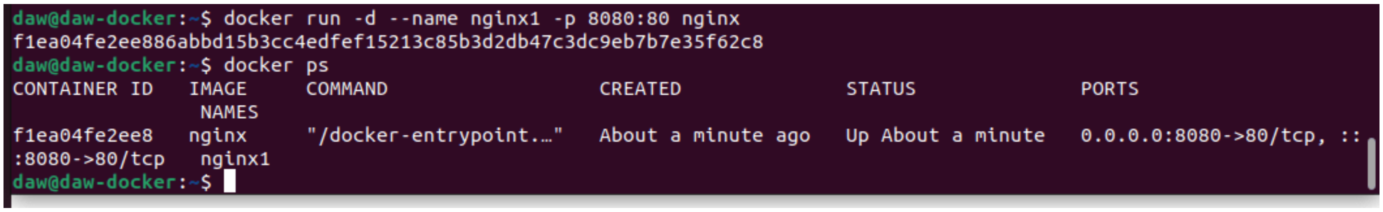
```
docker logs nginx2
```



```
daw@daw-docker:~$ docker run -d --name nginx2 nginx bash -c "while true; do echo hello-nginx; sleep 1; done"
7a88efd51a9359104eeed9fb3974baf6d1697ac3cb86a90069d3fba9e3e940ed
daw@daw-docker:~$ docker logs nginx2
hello-nginx
hello-nginx
hello-nginx
hello-nginx
hello-nginx
hello-nginx
hello-nginx
hello-nginx
```

Como no tenía muy claro el enunciado, he creado otro contenedor con un servidor nginx. nginx1.

```
docker run -d --name nginx1 -p 8080:80 nginx
```

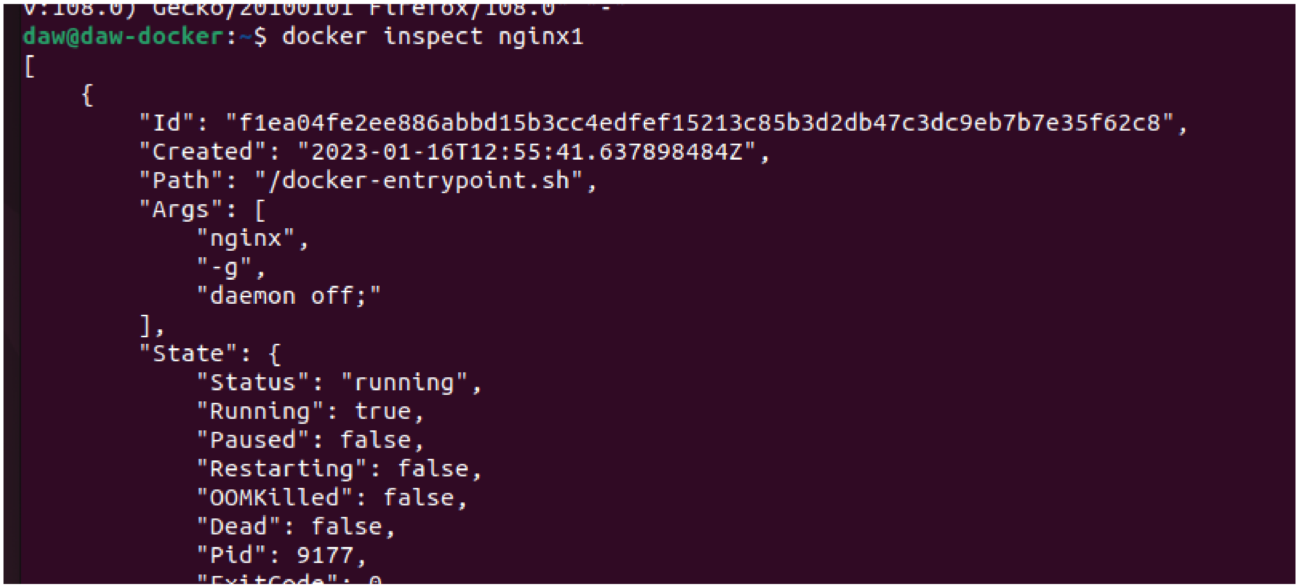


```
daw@daw-docker:~$ docker run -d --name nginx1 -p 8080:80 nginx
f1ea04fe2ee886abbd15b3cc4edfef15213c85b3d2db47c3dc9eb7b7e35f62c8
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS                                   
        NAMES
f1ea04fe2ee8   nginx     "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp, :
:8080->80/tcp   nginx1
daw@daw-docker:~$ 
```

Se accede siempre desde la ip, no desde el nombre.

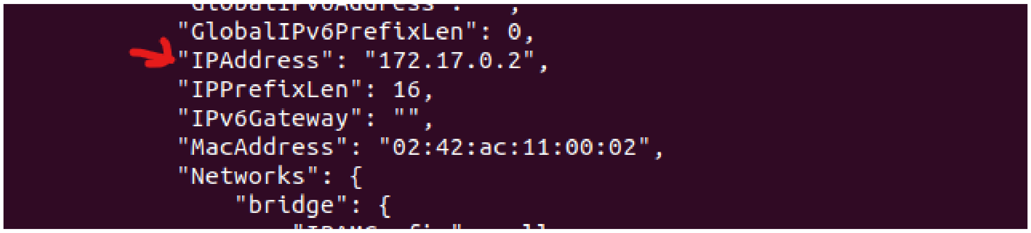
para saber la ip del contenedor:

```
docker inspect nginx1
```

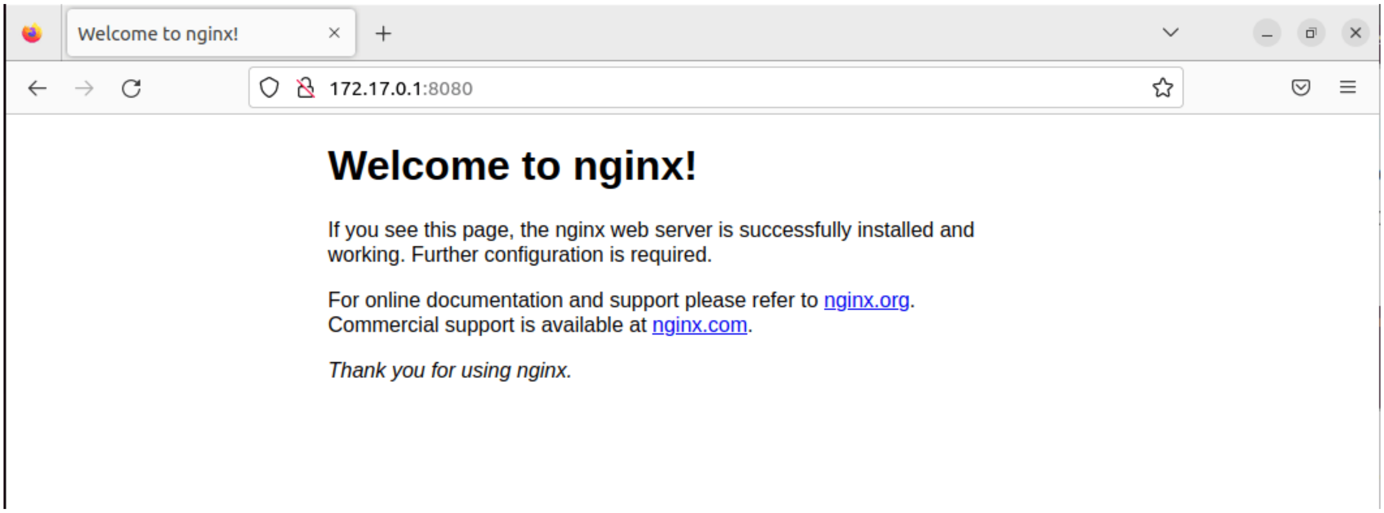


```
v:108.0) Gecko/20100101 Firefox/108.0  -
daw@daw-docker:~$ docker inspect nginx1
[
    {
        "Id": "f1ea04fe2ee886abbd15b3cc4edfef15213c85b3d2db47c3dc9eb7b7e35f62c8",
        "Created": "2023-01-16T12:55:41.637898484Z",
        "Path": "/docker-entrypoint.sh",
        "Args": [
            "nginx",
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 9177,
            "ExitCode": 0
```

y en el apartado de "networks"



```
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "172.17.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "MacAddress": "02:42:ac:11:00:02",
            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
```



Welcome to nginx!

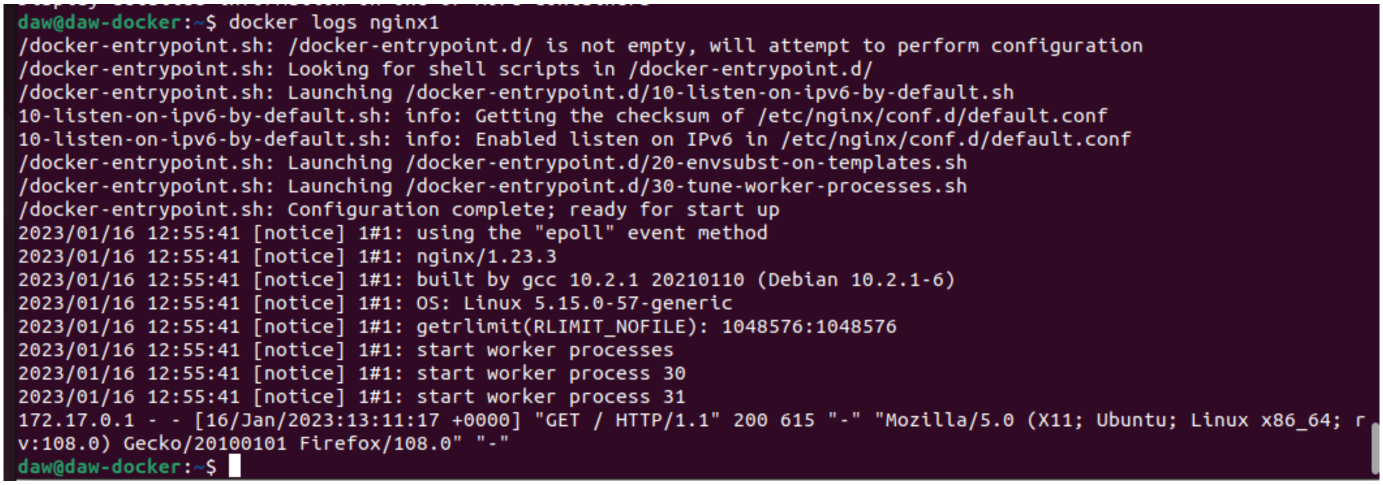
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Mostramos los logs de nginx1.

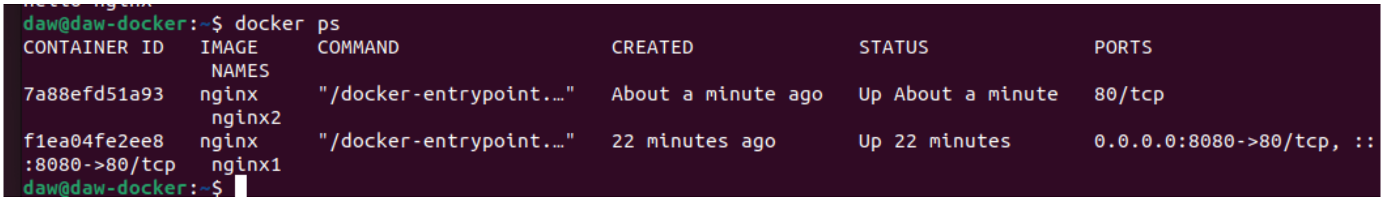
```
docker logs nginx1
```



```
daw@daw-docker:~$ docker logs nginx1
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/01/16 12:55:41 [notice] 1#1: using the "epoll" event method
2023/01/16 12:55:41 [notice] 1#1: nginx/1.23.3
2023/01/16 12:55:41 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/16 12:55:41 [notice] 1#1: OS: Linux 5.15.0-57-generic
2023/01/16 12:55:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/16 12:55:41 [notice] 1#1: start worker processes
2023/01/16 12:55:41 [notice] 1#1: start worker process 30
2023/01/16 12:55:41 [notice] 1#1: start worker process 31
172.17.0.1 - - [16/Jan/2023:13:11:17 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; r
v:108.0) Gecko/20100101 Firefox/108.0" "-"
daw@daw-docker:~$ 
```

Comprobamos que están activos los dos contenedores, ya que están funcionando en segundo plano.

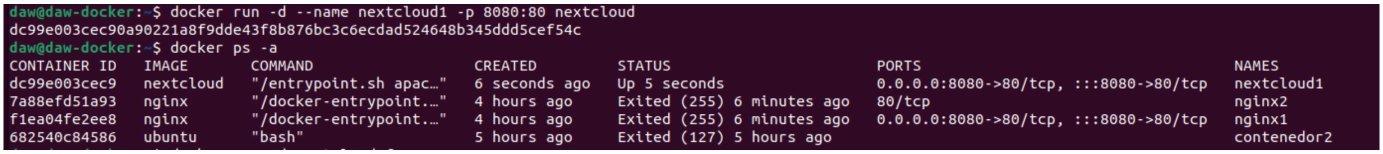
```
docker ps
```



```
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS
                NAMES
7a88efd51a93   nginx     "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp
                nginx2
f1ea04fe2ee8   nginx     "/docker-entrypoint.…"   22 minutes ago       Up 22 minutes       0.0.0.0:8080->80/tcp, ::
:8080->80/tcp   nginx1
daw@daw-docker:~$ 
```

5. Instalamos un contenedor nextcloud:

```
docker run -d -p 8080:80 nextcloud
```



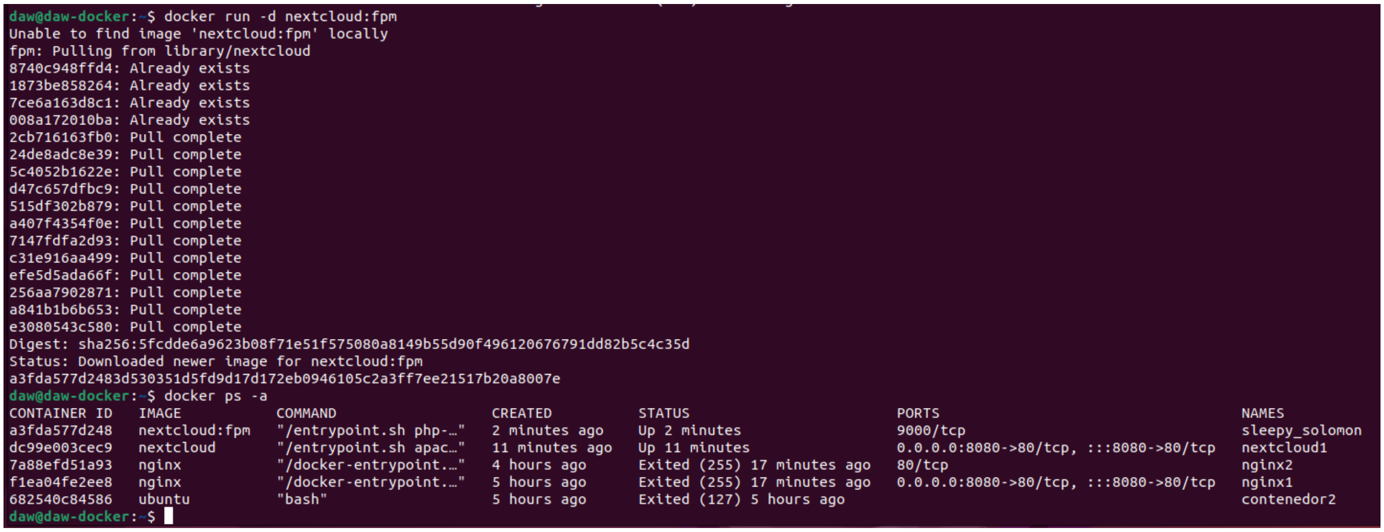
```
daw@daw-docker:~$ docker run -d --name nextcloud1 -p 8080:80 nextcloud
dc99e003cec90a90221a8f9dde43f8b876bc3c6ecdad524648b345ddd5cef54c
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE       COMMAND                  CREATED          STATUS                      PORTS                                   NAMES
dc99e003cec9   nextcloud   "/entrypoint.sh apac…"   6 seconds ago    Up 5 seconds                0.0.0.0:8080->80/tcp, :::8080->80/tcp   nextcloud1
7a88efd51a93   nginx       "/docker-entrypoint.…"   4 hours ago      Exited (255) 6 minutes ago  80/tcp                                  nginx2
f1ea04fe2ee8   nginx       "/docker-entrypoint.…"   4 hours ago      Exited (255) 6 minutes ago  0.0.0.0:8080->80/tcp, :::8080->80/tcp   nginx1
682540c84586   ubuntu      "bash"                   5 hours ago      Exited (127) 5 hours ago                                            contenedor2
```

Para poder usar la imagen fpm necesitamos instalar un servidor web adicional como nginx, cuyo puerto será 9000 por defecto. Para poder instalar esa imagen usaremos el siguiente comando:

```
docker run -d nextcloud:fpm
```

Y comprobamos que ambos existen:

```
docker ps -a
```



```
daw@daw-docker:~$ docker run -d nextcloud:fpm
Unable to find image 'nextcloud:fpm' locally
fpm: Pulling from library/nextcloud
8740c948ffd4: Already exists
1873be858264: Already exists
7ce6a163d8c1: Already exists
008a172010ba: Already exists
2cb716163fb0: Pull complete
24de8adc8e39: Pull complete
5c4052b1622e: Pull complete
d47c657dfbc9: Pull complete
515df302b879: Pull complete
a407f4354f0e: Pull complete
7147fdfa2d93: Pull complete
c31e916aa499: Pull complete
efe5d5ada66f: Pull complete
256aa7902871: Pull complete
a841b1b6b653: Pull complete
e3080543c580: Pull complete
Digest: sha256:5fcdde6a9623b08f71e51f575080a8149b55d90f496120676791dd82b5c4c35d
Status: Downloaded newer image for nextcloud:fpm
a3fda577d2483d530351d5fd9d17d172eb0946105c2a3ff7ee21517b20a8007e
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE           COMMAND                  CREATED          STATUS                       PORTS                                     NAMES
a3fda577d248   nextcloud:fpm   "/entrypoint.sh php-…"   2 minutes ago    Up 2 minutes                 9000/tcp                                  sleepy_solomon
dc99e003cec9   nextcloud       "/entrypoint.sh apac…"   11 minutes ago   Up 11 minutes                0.0.0.0:8080->80/tcp, :::8080->80/tcp     nextcloud1
7a88efd51a93   nginx           "/docker-entrypoint.…"   4 hours ago      Exited (255) 17 minutes ago  80/tcp                                    nginx2
f1ea04fe2ee8   nginx           "/docker-entrypoint.…"   5 hours ago      Exited (255) 17 minutes ago  0.0.0.0:8080->80/tcp, :::8080->80/tcp     nginx1
682540c84586   ubuntu          "bash"                   5 hours ago      Exited (127) 5 hours ago                                               contenedor2
daw@daw-docker:~$
```

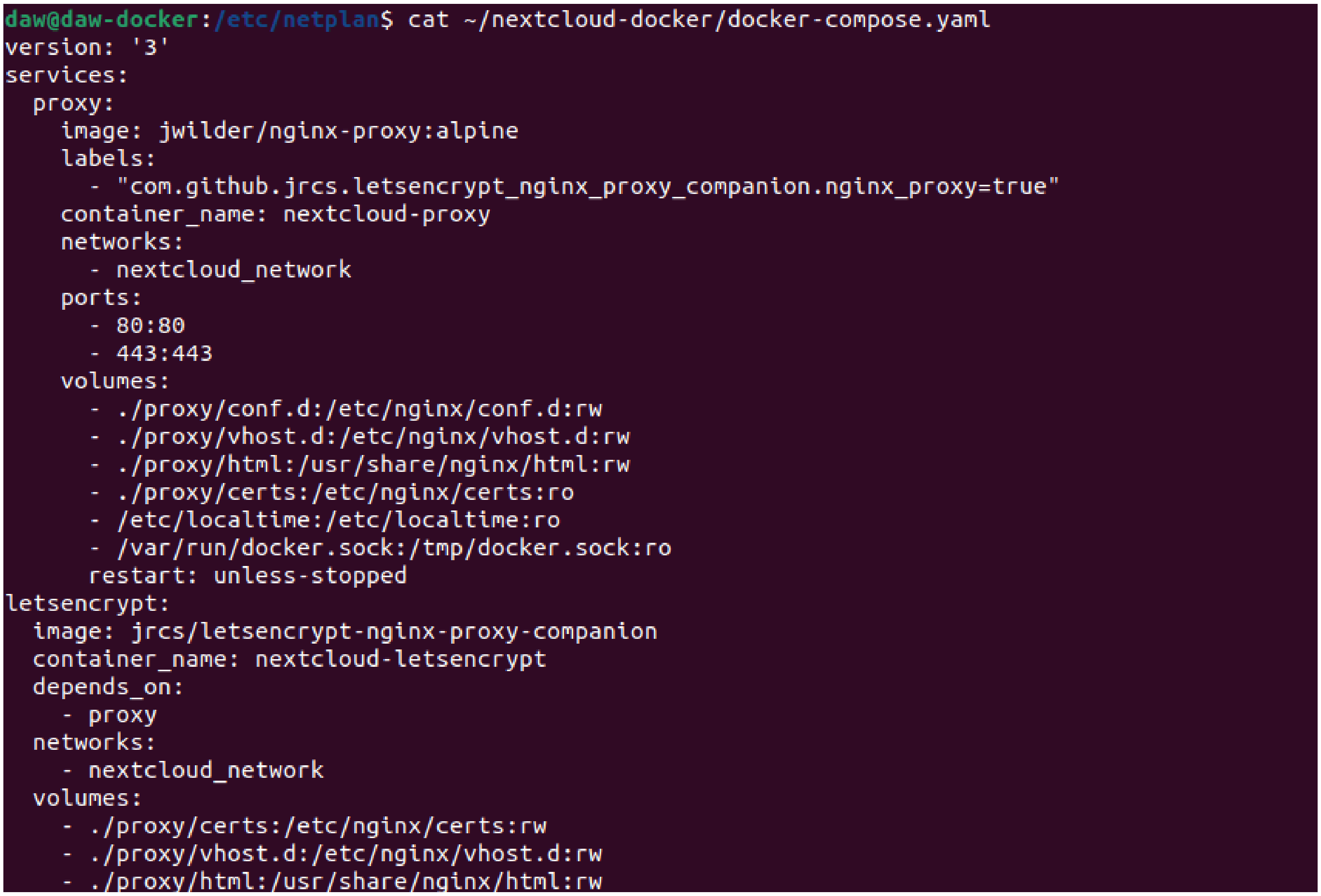
Para poder modificar el nombre de la base de datos de SQLITE tiene que especificarse el comando, ya que nextcloud viene con configuración automática.

Tenemos que crear el archivo en el que se pueda modicar el nombre Primero creamos la carpeta en la que irán los archivos.

```
mkdir ~/nextcloud-docker/
touch ~/nextcloud-docker/docker-compose.yaml
touch ~/nextcloud-docker/.env
```

Luego crearemos dos archivos un en el ymal, que ira la configuración completa del servidor Nextcloud: proxy, base de datos...

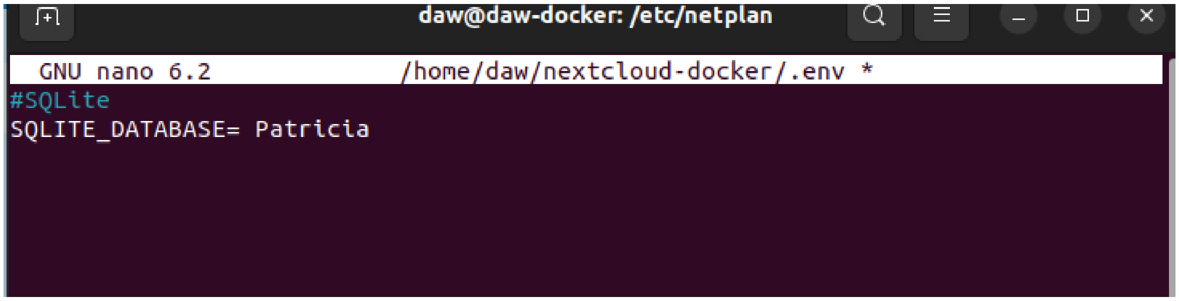
```
nano ~/nextcloud-docker/docker-compose.yaml
```



```
daw@daw-docker:/etc/netplan$ cat ~/nextcloud-docker/docker-compose.yaml
version: '3'
services:
  proxy:
    image: jwilder/nginx-proxy:alpine
    labels:
      - "com.github.jrcs.letsencrypt_nginx_proxy_companion.nginx_proxy=true"
    container_name: nextcloud-proxy
    networks:
      - nextcloud_network
    ports:
      - 80:80
      - 443:443
    volumes:
      - ./proxy/conf.d:/etc/nginx/conf.d:rw
      - ./proxy/vhost.d:/etc/nginx/vhost.d:rw
      - ./proxy/html:/usr/share/nginx/html:rw
      - ./proxy/certs:/etc/nginx/certs:ro
      - /etc/localtime:/etc/localtime:ro
      - /var/run/docker.sock:/tmp/docker.sock:ro
      restart: unless-stopped
letsencrypt:
  image: jrcs/letsencrypt-nginx-proxy-companion
  container_name: nextcloud-letsencrypt
  depends_on:
    - proxy
  networks:
    - nextcloud_network
  volumes:
    - ./proxy/certs:/etc/nginx/certs:rw
    - ./proxy/vhost.d:/etc/nginx/vhost.d:rw
    - ./proxy/html:/usr/share/nginx/html:rw
```

Y luego el archivo .env en el que irán los datos de la base de datos.

```
nano ~/nextcloud-docker/.env
```



```
  GNU nano 6.2              /home/daw/nextcloud-docker/.env *
#SQLite
SQLITE_DATABASE= Patricia
```

SQLITE_DATABASE Nombre de la base de datos usando sqlite

Bibliografía:

- https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-es
- https://www.ionos.mx/digitalguide/servidores/configuracion/instalacion-de-nextcloud-con-docker/
- https://hub.docker.com/_/nextcloud#docker=secrets