

Docker Ejercicios Repaso

1. Instala docker en una máquina y configúralo para que se pueda usar con un usuario sinprivilegios.

Estoy utilizando la MV de clase, pero los comandos para conseguirlo serian:

instalamos docker

```
apt install docker.io
```

Para utilizar docker

```
usermod -aG docker usuario
```

2. Ejecuta un contenedor a partir de la imagen hello-word. Comprueba que nos devuelve la salida adecuada. Comprueba que no se está ejecutando. Lista los contenedores que están parado. Borra el contenedor.

```
ago ntrty_rermt
daw@daw-docker:~$ docker run hello-world

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/

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

```
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
2d54b80b5a84   hello-world  "/hello"                About a minute ago  Exited (0) About a minute ago
goofy_aryabhata
aadfed744026   mariadb     "docker-entrypoint.s..." 33 hours ago    Exited (255) 33 hours ago    0.0.0.0:3306->3306/tcp, :::3306->3
306/tcp some-mariadb
a865274aa83f   httpd:2.4   "httpd-foreground"       3 days ago      Exited (0) 34 hours ago
my-apache-app
1f9954f88b7a   hello-world  "/hello"                3 days ago      Exited (0) 3 days ago
nifty_ferri
daw@daw-docker:~$ docker rm hello-world
Error: No such container: hello-world
daw@daw-docker:~$ docker rm goofy_aryabhata
goofy_aryabhata
daw@daw-docker:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
aadfed744026   mariadb     "docker-entrypoint.s..." 33 hours ago    Exited (255) 33 hours ago    0.0.0.0:3306->3306/tcp, :::3306->3306/tcp
some-mariadb
a865274aa83f   httpd:2.4   "httpd-foreground"       3 days ago      Exited (0) 34 hours ago
my-apache-app
1f9954f88b7a   hello-world  "/hello"                3 days ago      Exited (0) 3 days ago
nifty_ferri
daw@daw-docker:~$
```

3. Crea un contenedor interactivo desde una imagen debian. Instala un paquete (por ejemplonano). Sal de la terminal, ¿sigue el contenedor corriendo? ¿Por qué?. Vuelve a iniciar el contenedor y accede de nuevo a él de forma interactiva. ¿Sigue instalado el nano?. Sal del contenedor, y bórralo. Crea un nuevo contenedor interactivo desde la misma imagen. ¿Tiene el nano instalado?

```
daw@daw-docker:~$ docker run -it debian
root@15cf7696ad6b:/# nano prueba
root@15cf7696ad6b:/# cat prueba
Despliegue
root@15cf7696ad6b:/# apt install nano
```

Si abrimos otra terminal vemos que seguira activo

```
daw@daw-docker:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
15cf7696ad6b   debian     "bash"                  6 minutes ago  Up 6 minutes  affectionate_hopper
daw@daw-docker:~$
```

Si nos salimos del contenedor:

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

Lo volvemos a ejecutar y vemos que seguimos teniendo nano

```

affectionate_hopper
daw@daw-docker:~$ docker start -i affectionate_hopper
root@15cf7696ad6b:/# nano prueba 2
root@15cf7696ad6b:/# cat prueba 2
Despliegue Nuevo

cat: 2: No such file or directory
root@15cf7696ad6b:/# cat 2
cat: 2: No such file or directory
root@15cf7696ad6b:/# cat prueba
Despliegue Nuevo

```

Borramos y creamos otro contenedor con la misma imagen y obviamente veremos que no tenemos el nano

```

daw@daw-docker:~$ docker rm affectionate_hopper
affectionate_hopper
daw@daw-docker:~$ docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|---|--------------|--------------------------|--------------|---------------------------|
| aadfed744026 | mariadb | "docker-entrypoint.s..." | 34 hours ago | Exited (255) 33 hours ago |
| 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp | some-mariadb | | | |
| a865274aa83f | httpd:2.4 | "httpd-foreground" | 3 days ago | Exited (0) 34 hours ago |
| 1f9954f88b7a | hello-world | "/hello" | 3 days ago | Exited (0) 3 days ago |
| | nifty_fermi | | | |

```

daw@daw-docker:~$

```

```

nifty_fermi
daw@daw-docker:~$ docker run -it --name contenedordebian debian
root@d1d9e27e90eb:/# nano prueba
bash: nano: command not found
root@d1d9e27e90eb:/#

```

4. Crea un contenedor demonio con un servidor nginx, usando la imagen oficial de nginx. Al crear el contenedor, ¿has tenido que indicar algún comando para que lo ejecute? Accede al navegador web y comprueba que el servidor esta funcionando. Muestra los logs del contenedor.

Creamos el contenedor con la imagen oficial de nginx, no hemos tenido que indicar ningun comando para que se ejecute ya que le hemos metido modificadores para que lo haga directamente

```

daw@daw-docker:~$ docker run -it -d -p 8080:80 --name contenedornginx nginx
0a7146ae6a9fe0aeb2f4ec3314c3f889b35c39c708f0ff584c094eda7db10532
daw@daw-docker:~$ docker ps

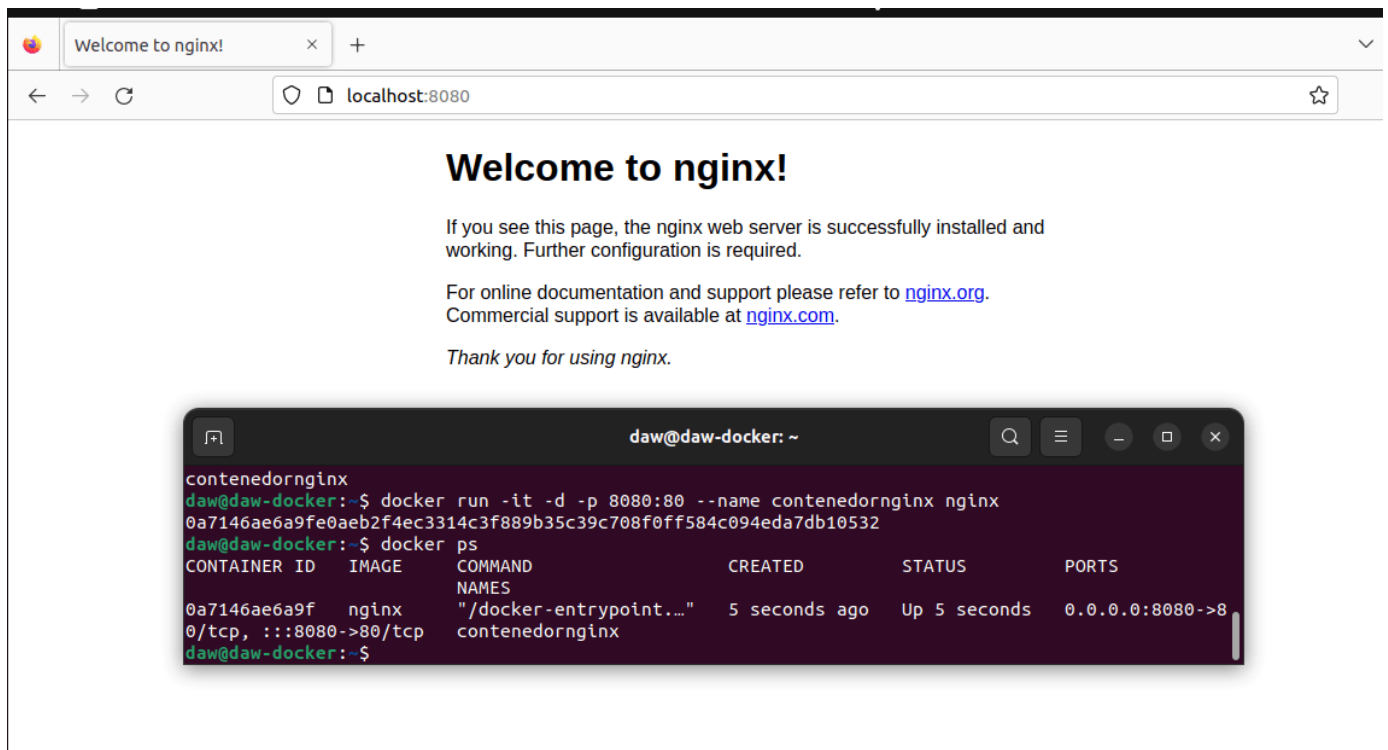
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS |
|--------------|-----------------|--------------------------|---------------|--------------|---------------------------------------|
| 0a7146ae6a9f | nginx | "/docker-entrypoint...." | 5 seconds ago | Up 5 seconds | 0.0.0.0:8080->80/tcp, :::8080->80/tcp |
| | contenedornginx | | | | |

```

daw@daw-docker:~$

```



```
daw@daw-docker:~$ docker logs contenedornginx
/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
2023/01/14 18:57:22 [notice] 1#1: using the "epoll" event method
2023/01/14 18:57:22 [notice] 1#1: nginx/1.23.3
2023/01/14 18:57:22 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/14 18:57:22 [notice] 1#1: OS: Linux 5.15.0-57-generic
2023/01/14 18:57:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/14 18:57:22 [notice] 1#1: start worker processes
2023/01/14 18:57:22 [notice] 1#1: start worker process 29
2023/01/14 18:57:22 [notice] 1#1: start worker process 30
2023/01/14 18:57:22 [notice] 1#1: start worker process 31
2023/01/14 18:57:22 [notice] 1#1: start worker process 32
2023/01/14 18:57:22 [notice] 1#1: start worker process 33
2023/01/14 18:57:22 [notice] 1#1: start worker process 34
2023/01/14 18:57:22 [notice] 1#1: start worker process 35
2023/01/14 18:57:22 [notice] 1#1: start worker process 36
2023/01/14 18:57:22 [notice] 1#1: start worker process 37
172.17.0.1 - - [14/Jan/2023:18:58:23 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0" "-"
2023/01/14 18:58:23 [error] 29#29: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8080", referrer: "http://localhost:8080/"
```

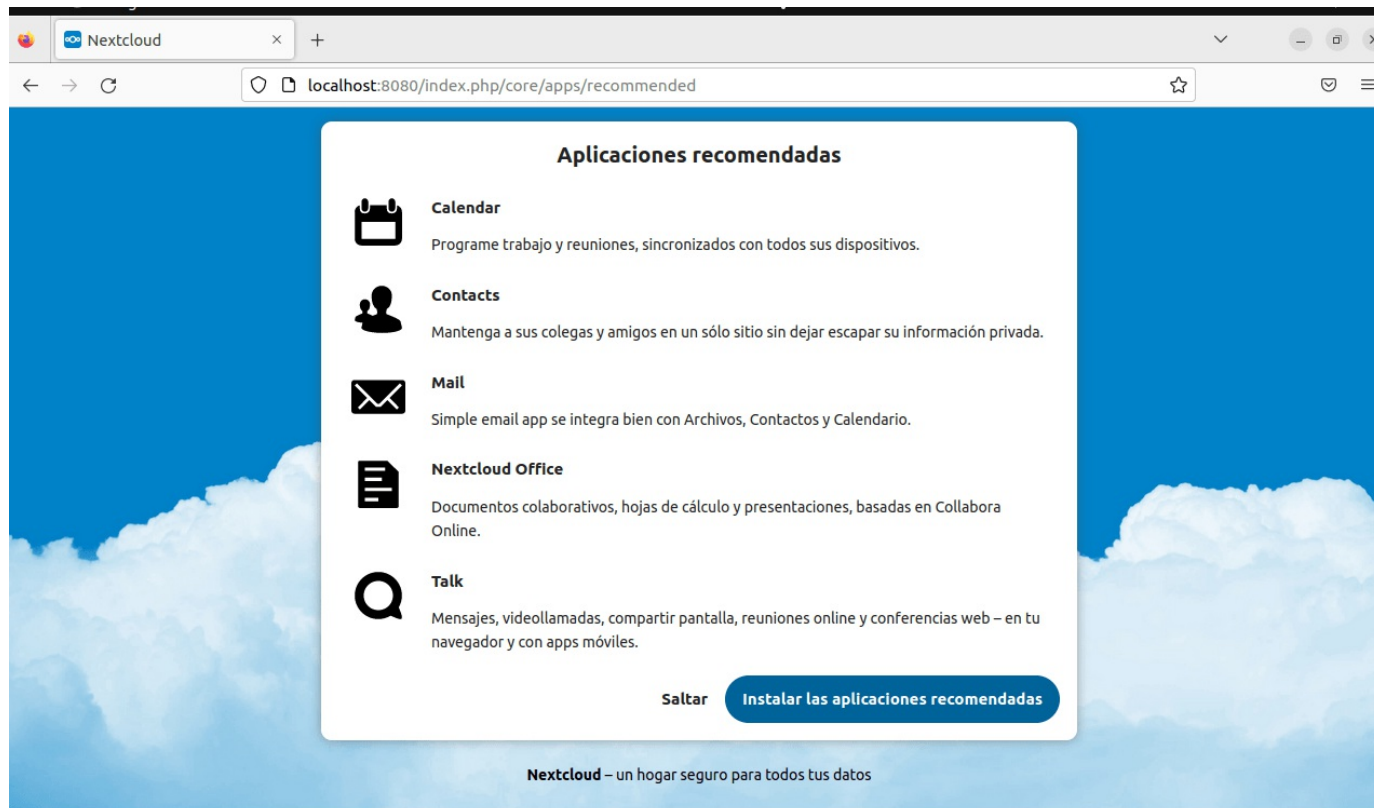
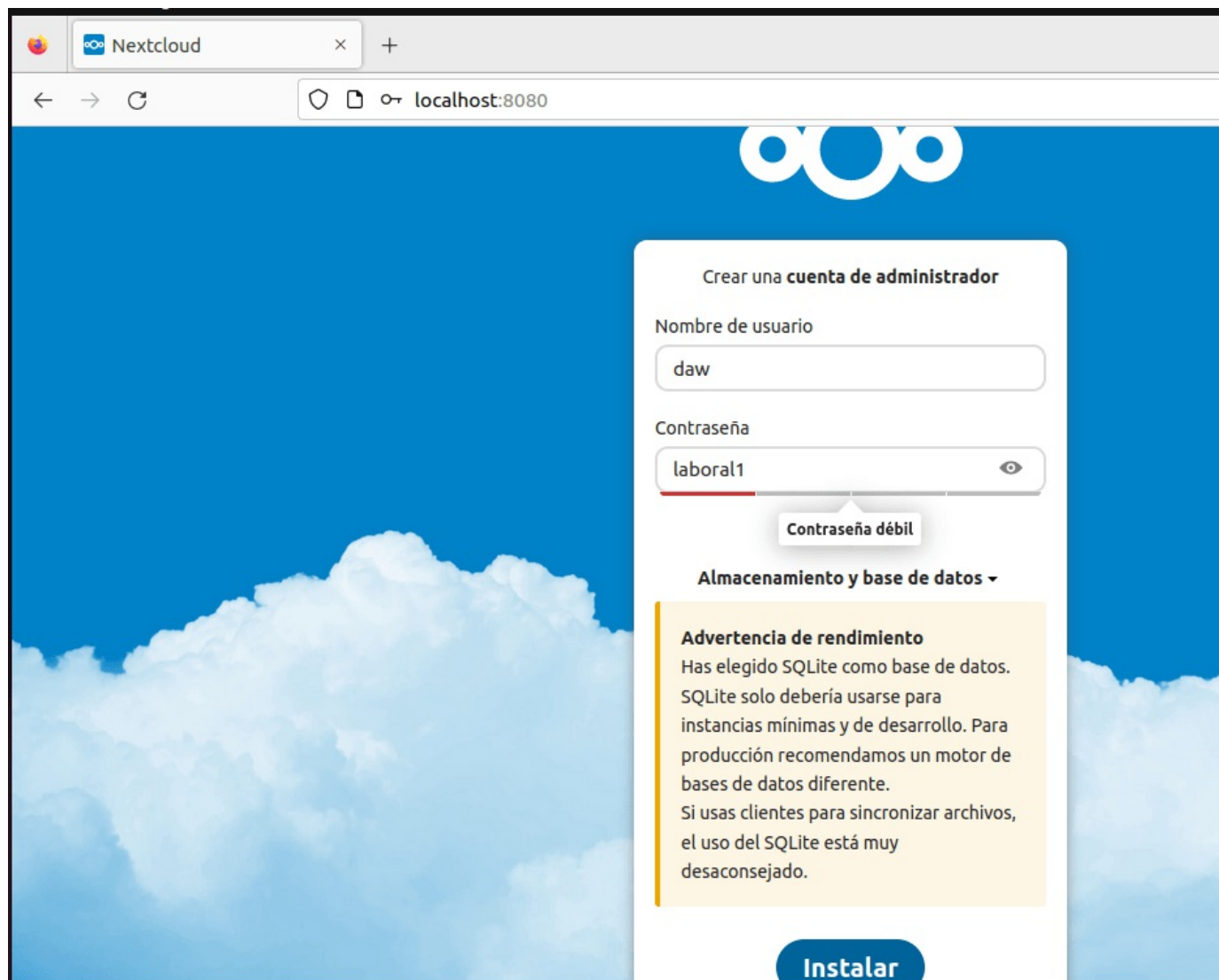
5. Crea un contenedor con la aplicación Nextcloud, mirando la documentación en docker Hub, para personalizar el nombre de la base de datos sqlite que va a utilizar.

Utilizaremos esta [pagina de referencia](#)

Creamos el contenedor, utilizando el puerto 8080 con la imagen de nextcloud

```
connected to the container
daw@daw-docker:~$ docker run -d -p 8080:80 nextcloud
Unable to find image 'nextcloud:latest' locally
latest: Pulling from library/nextcloud
8740c948ffd4: Already exists
1873be858264: Pull complete
7ce6a163d8c1: Pull complete
008a172010ba: Pull complete
d15353ae3d77: Pull complete
223eb1888c0f: Pull complete
83374c2a967a: Pull complete
8fdc86711b26: Pull complete
23c0224c39b8: Pull complete
915d82c7f5c5: Pull complete
dc037a9c9035: Pull complete
768542e0b637: Pull complete
d7ade602d94f: Pull complete
7361225e9a5d: Pull complete
fdc75c5d6478: Pull complete
6e598d96642e: Pull complete
2183a95f6531: Pull complete
e4d461a63b9a: Pull complete
e50f69db5ce5: Pull complete
4e2f130d99f4: Pull complete
Digest: sha256:01435cc1463c01a8a611fd387dc588ccbd4abe9bf290d668c7d80eac160ab4ef
Status: Downloaded newer image for nextcloud:latest
```

Cuando acaba podemos ir a localhost para crear una cuenta de administrador y realizar unas ultimas configuraciones



Y ya lo tendríamos todo disponible:

