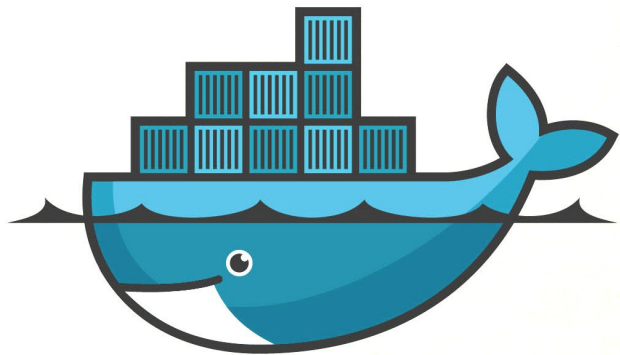
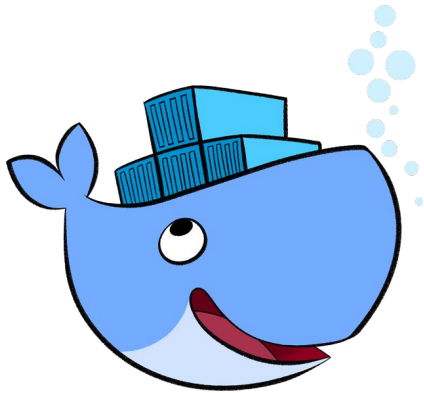


Almacenamiento



docker

Bind mount



Ejercicio 1

Ejercicio 1

1. Crea un contenedor a partir de la imagen oficial de NextCloud usando *bind mount* para persistir los datos en una carpeta llamado “datos”. Selecciona como base de datos SQLite durante la instalación.
2. Sube algún archivo.
3. Comprueba desde el host que el archivo está dentro de la carpeta “datos”.
4. Luego para el contenedor y bórralo (el contenedor, no el archivo).
5. Finalmente crea otro contenedor usando el mismo directorio “datos” que has usado en el primero y comprueba que el archivo que has subido sigue ahí en la interfaz de NextCloud.

Ejercicio 1 - Solución

1. Crea un contenedor a partir de la imagen oficial de NextCloud usando *bind mount* para persistir los datos en una carpeta llamado “datos”. Selecciona como base de datos SQLite durante la instalación.

```
~/Documentos/docker-examples/06-ejer01  
→ docker run -d -p 80:80 -v $(pwd)/datos:/var/www/html --name nextcloud nextcloud  
0af12d95c5ab50abd761725bd58aa5b66e5f5210b767391ebdf503c58a8fdeec
```

Ejercicio 1 - Solución

```
~/Documentos/docker-examples/06-ejer01
→ l
total 12K
drwxrwxr-x 3 pepe      pepe 4,0K abr 11 16:43 .
drwxrwxr-x 9 pepe      pepe 4,0K abr 11 16:28 ..
drwxr-xr-x 4 www-data  root 4,0K abr 11 16:43 datos
```

Se ha creado la carpeta “datos”

Contiene los datos de NextCloud:
configuraciones, archivos subidos,
etc.

```
~/Documentos/docker-examples/06-ejer01
→ l datos
total 168K
drwxr-xr-x 14 www-data  root 4,0K abr 11 16:43 .
drwxrwxr-x  3 pepe      pepe 4,0K abr 11 16:43 ..
drwxr-xr-x 44 www-data  root 4,0K abr 11 16:43 3rdparty
drwxr-xr-x 48 www-data  root 4,0K abr 11 16:43 apps
-rw-r--r--  1 www-data  root 19K abr 11 16:43 AUTHORS
drwxr-xr-x  2 www-data  root 4,0K abr 11 16:43 config
-rw-r--r--  1 www-data  root 3,9K abr 11 16:43 console.php
-rw-r--r--  1 www-data  root 34K abr 11 16:43 COPYING
drwxr-xr-x 22 www-data  root 4,0K abr 11 16:43 core
-rw-r--r--  1 www-data  root 6,0K abr 11 16:43 cron.php
drwxr-xr-x  2 www-data  root 4,0K abr 11 16:43 custom_apps
```

Ejercicio 1 - Solución

1. Vamos a localhost.
2. Asignamos un usuario y contraseña.
3. Seleccionamos SQLite como base de datos.
4. Le damos a instalar.
5. Instalamos las aplicaciones recomendadas.

← → ↻ 🏠 ⓘ localhost

Next.js 12.0.0

Crear una **cuenta de administrador**

pepe

....

Contraseña muy débil

Almacenamiento y base de datos

Carpeta de datos

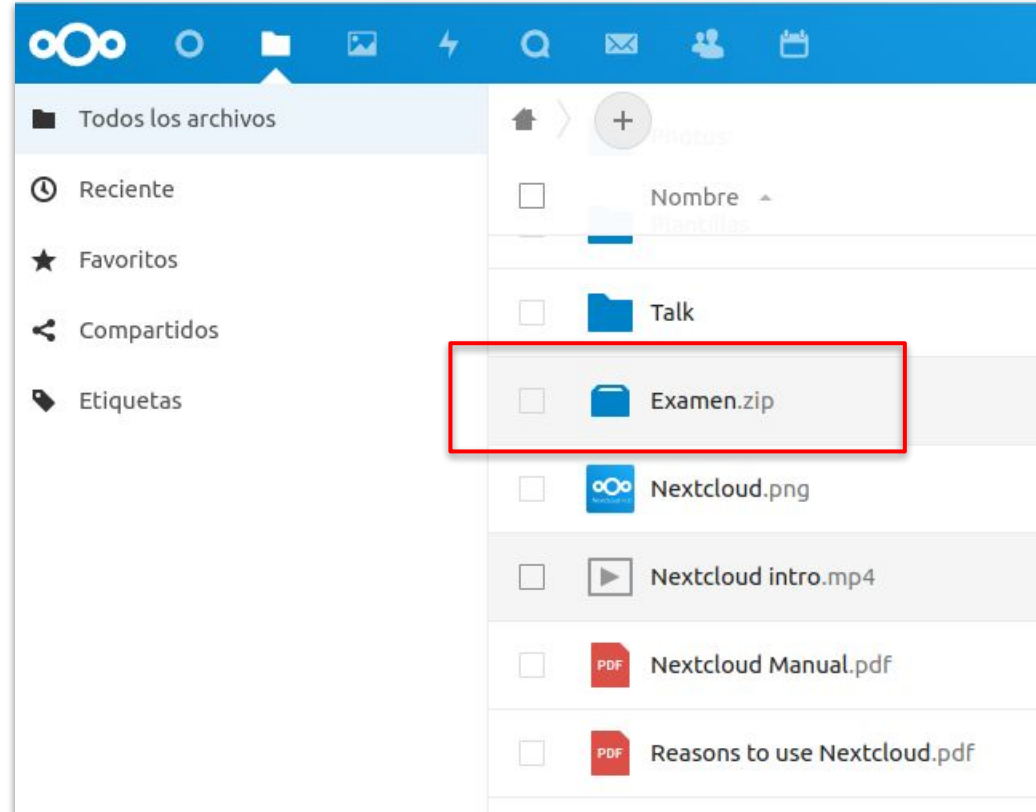
/var/www/html/data

Configurar la base de datos

SQLite MySQL/MariaDB PostgreSQL

Ejercicio 1 - Solución

2. Vamos a archivos y subimos uno. En este caso “Examen.zip”



Ejercicio 1 - Solución

3. Comprobamos que podemos acceder desde el host, y está en el directorio “datos”.

```
~/Documentos/docker-examples/06-ejer01  
→ sudo ls -l --color datos/data/pepe/files  
total 17808  
drwxr-xr-x 2 www-data www-data    4096 abr 11 16:53 Documents  
→ -rw-r--r-- 1 www-data www-data 512374 feb 24 18:23 Examen.zip  
-rw-r--r-- 1 www-data www-data 3963036 abr 11 16:53 'Nextcloud intro.mp4'  
-rw-r--r-- 1 www-data www-data 12704899 abr 11 16:53 'Nextcloud Manual.pdf'  
-rw-r--r-- 1 www-data www-data    50598 abr 11 16:53 Nextcloud.png  
drwxr-xr-x 2 www-data www-data    4096 abr 11 16:53 Photos  
drwxr-xr-x 2 www-data www-data    4096 abr 11 16:53 Plantillas  
-rw-r--r-- 1 www-data www-data    976625 abr 11 16:53 'Reasons to use Nextcloud.pdf'  
drwxr-xr-x 2 www-data www-data    4096 abr 11 16:54 Talk
```

Ejercicio 1 - Solución

4. Paramos y borramos el contenedor.

```
~/Documentos/docker-examples/06-ejer01  
→ docker stop nextcloud && docker rm nextcloud  
nextcloud  
nextcloud
```

5. Lanzamos otro contenedor haciendo bind mount del mismo directorio “datos”.

```
~/Documentos/docker-examples/06-ejer01  
→ docker run -d -p 80:80 -v $(pwd)/datos:/var/www/html --name nextcloud2 nextcloud  
745d853b598530580a28fadb133b05762f6b6317d22d8c77646a392b5d0feaa8
```

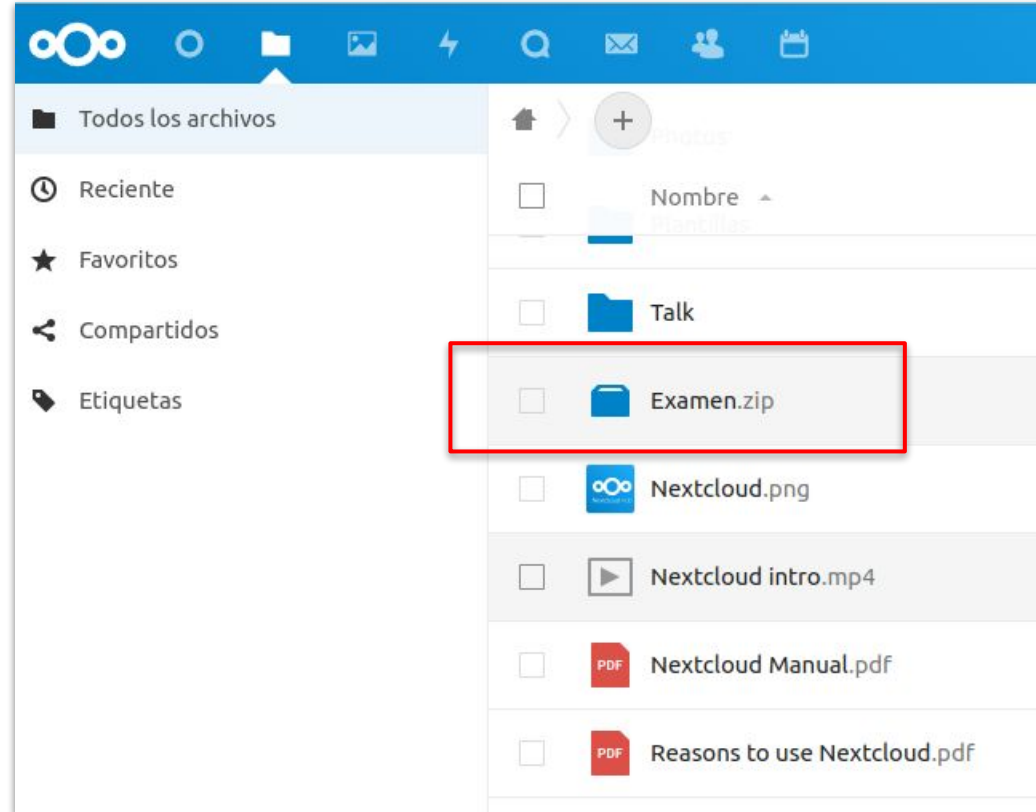
Ejercicio 1 - Solución

Reconoce que hay una instalación y nos pide directamente iniciar sesión.



Ejercicio 1 - Solución

Vamos a archivos
comprobamos que tenemos
nuestro “Examen.zip”



Ejercicio 2

Ejercicio 2

1. Crea un contenedor a partir de la imagen oficial de MySQL usando *bind mount* para persistir las bases de datos en una carpeta llamado “db”.
2. Lanza un contenedor phpmyadmin que se conecte al MySQL y crea una base de datos.
3. Para y elimina los contenedores.
4. Vuele a hacer los pasos 1 y 2 y comprueba que la base de datos ha persistido.

Ejercicio 2 - Solución

1. Crea un contenedor a partir de la imagen oficial de MySQL usando *bind mount* para persistir las bases de datos en una carpeta llamado "db".

```
→ docker run -d --name mysql -v $(pwd)/db:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=root mysql  
b30f2a92d9f740dbbdf0b11b41785172bdd17cc9d526bf194a7dfc63a36abfce
```

2. Lanza un contenedor phpmyadmin que se conecte al MySQL y crea una base de datos.

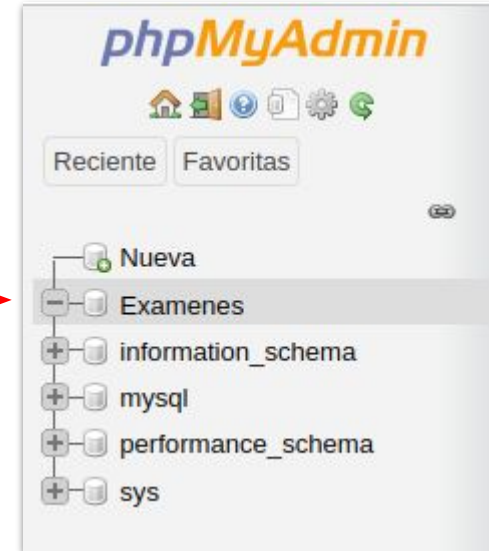
```
→ docker run --name dbadmin -d --link mysql:db -p 8080:80 phpmyadmin  
3a9073dfe901dc638ef31a56c5dd4544b34a5d3c472081d187ab88c4a63d26a4
```

Ejercicio 2 - Solución

Iniciamos sesión y creamos la base de datos “Exámenes”



The image shows the phpMyAdmin login interface. At the top, there is a logo with a sailboat and the text "phpMyAdmin". Below the logo, it says "Bienvenido a phpMyAdmin". There is a section for "Idioma - Language" with a dropdown menu set to "Español - Spanish". Below that is a section for "Iniciar sesión" (Login) with a button icon. It contains two input fields: "Usuario:" (Username) with the value "root" and "Contraseña:" (Password) with masked characters "....". At the bottom right of the login section is a "Continuar" (Continue) button.



Ejercicio 2 - Solución

Podemos verla ya creada también en el sistema de archivos:

```
→ l db
total 194M
drwxr-xr-x 7 systemd-coredump root          4,0K abr 11 17:26 .
drwxrwxr-x 3 pepe          pepe          4,0K abr 11 17:23 ..
-rw-r----- 1 systemd-coredump systemd-coredump 56 abr 11 17:23 auto.cnf
-rw-r----- 1 systemd-coredump systemd-coredump 3,0M abr 11 17:24 binlog.000001
-rw-r----- 1 systemd-coredump systemd-coredump 407 abr 11 17:26 binlog.000002
-rw-r----- 1 systemd-coredump systemd-coredump 32 abr 11 17:24 binlog.index
-rw-r----- 1 systemd-coredump systemd-coredump 1,7K abr 11 17:24 ca-key.pem
-rw-r--r-- 1 systemd-coredump systemd-coredump 1,1K abr 11 17:24 ca.pem
-rw-r--r-- 1 systemd-coredump systemd-coredump 1,1K abr 11 17:24 client-cert.pem
-rw-r----- 1 systemd-coredump systemd-coredump 1,7K abr 11 17:24 client-key.pem
drwxr-x--- 2 systemd-coredump systemd-coredump 4,0K abr 11 17:26 Examenes
-rw-r----- 1 systemd-coredump systemd-coredump 192K abr 11 17:26 '#ib_16384_0.dblwr'
-rw-r----- 1 systemd-coredump systemd-coredump 8,2M abr 11 17:23 '#ib_16384_1.dblwr'
```

Ejercicio 2 - Solución

3. Paramos y eliminamos los contenedores.

```
~/Documentos/docker-examples/06-ejer02
```

```
→ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3a9073dfe901	phpmyadmin	"/docker-entrypoint..."	6 minutes ago	Up 6 minutes	0.0.0.0:8080->80/tcp, :::8080->80/tcp	dbadmin
b30f2a92d9f7	mysql	"docker-entrypoint.s..."	6 minutes ago	Up 6 minutes	3306/tcp, 33060/tcp	mysql

```
~/Documentos/docker-examples/06-ejer02
```

```
→ docker stop 3a9073dfe901 b30f2a92d9f7 && docker rm 3a9073dfe901 b30f2a92d9f7
```

```
3a9073dfe901
```

```
b30f2a92d9f7
```

```
3a9073dfe901
```

```
b30f2a92d9f7
```

```
~/Documentos/docker-examples/06-ejer02
```

```
→ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Ejercicio 2 - Solución

4. Volvemos a crear los contenedores.

```
~/Documentos/docker-examples/06-ejer02
```

```
→ docker run -d --name mysql -v $(pwd)/db:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=root mysql  
12ca70c726fa93a93571ba2d3c4280e8be4fcef3bb9276d0496d57947ec1199a
```

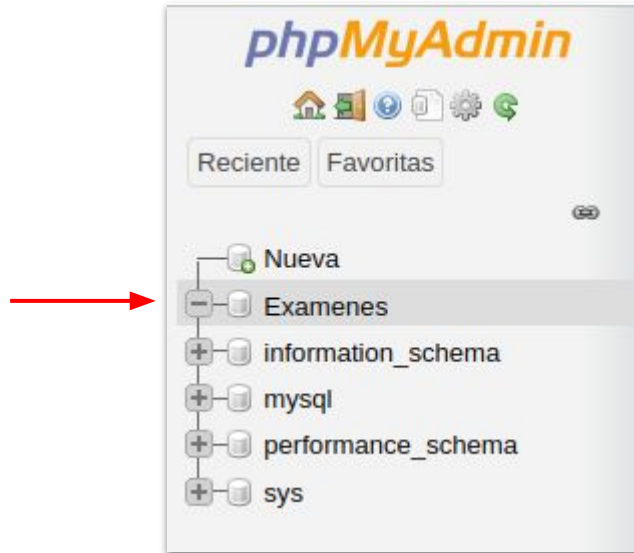
```
~/Documentos/docker-examples/06-ejer02
```

```
→ docker run --name dbadmin -d --link mysql:db -p 8080:80 phpmyadmin
```

```
457be790fdcb01c888a9865bf48a3faf8495f0de2e672fe9a593da66e172b041
```

Ejercicio 2 - Solución

Y comprobamos que la base de datos “Exámenes” ha persistido.



Volúmenes



Ejercicio 1

Ejercicio 1

Haz el ejercicio 1 de bind mount usando ahora volúmenes.

Ejercicio 1 - Solución

La única diferencia es el comando para lanzar el contenedor nextcloud:

```
~/Documentos/docker-examples/06-ejer01  
→ docker run -d -p 80:80 -v datos:/var/www/html --name nextcloud nextcloud  
2ea87f42f551f3853e0f71ef331ef3129c8dd698123b2d610162680838cca2ee
```

```
~/Documentos/docker-examples/06-ejer01  
→ docker volume ls  
DRIVER      VOLUME NAME  
local      datos
```


Ejercicio 2

Ejercicio 2

Haz el ejercicio 2 de bind mount usando ahora volúmenes.

Ejercicio 2 - Solución

La única diferencia es el comando para lanzar el contenedor mysql:

```
→ docker run -d --name mysql -v db:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=root mysql  
3e712558fc17fd9a64b4abafc2efd6f6abbfc96945639c8ab4f76df42e3db0ac
```

```
→ docker volume ls  
DRIVER      VOLUME NAME  
local      db
```

Ejercicio 3

Ejercicio 3

A partir del ejercicio anterior, en que la persistencia de la base de datos mysql se está haciendo en un volumen, se pide crear un contenedor que haga un archivo tar.gz con el contenido del volumen en un directorio del host llamado backup.

El contenedor tienen que borrarse después de hacer la copia.

Ejercicio 3 - Solución

```
~/Documentos/docker-examples/06-ejer03
```

```
→ docker run --rm -v db:/database -v $(pwd)/backup:/backup bash bash -c "cd database && tar czvf ../backup/db.tar.gz ."
```

```
./
./public_key.pem
./ca-key.pem
./ib_buffer_pool
./undo_002
./ib_logfile0
./mysql.ibd
./performance_schema/
./performance_schema/replication_appl_170.sdi
./performance_schema/events_transacti_131.sdi
./performance_schema/events_errors_su_139.sdi
./performance_schema/replication_appl_165.sdi
./performance_schema/users_144.sdi
./performance_schema/table_handles_158.sdi
./performance_schema/session_account__151.sdi
./performance_schema/replication_conn_164.sdi
./performance_schema/file_instances_93.sdi
./performance_schema/setup_objects_105.sdi
./performance_schema/events_stages_su_115.sdi
./performance_schema/keyring_keys_152.sdi
```

Ejercicio 3 - Solución

```
~/Documentos/docker-examples/06-ejer03
→ tree
.
├── backup
│   └── db.tar.gz
└──
```

1 directory, 1 file

```
~/Documentos/docker-examples/06-ejer03
→ ll backup
total 11M
-rw-r--r-- 1 root root 11M abr 11 20:34 db.tar.gz
```

```
~/Documentos/docker-examples/06-ejer03
→ tar --list -f backup/db.tar.gz
./
./public_key.pem
./ca-key.pem
./ib_buffer_pool
./undo_002
./ib_logfile0
./mysql.ibd
./performance_schema/
./performance_schema/replication_appl_170.sdi
```

El contenedor se ha borrado gracias a la opción `--rm`:

```
~/Documentos/docker-examples/06-ejer03
→ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0828bd301e20	phpmyadmin	"/docker-entrypoint...."	15 minutes ago	Up 15 minutes	0.0.0.0:8080->80/tcp, :::8080->80/tcp	dbadmin
3e712558fc17	mysql	"docker-entrypoint.s..."	23 minutes ago	Up 23 minutes	3306/tcp, 33060/tcp	mysql