

Despliegue de Nextcloud con Docker Compose

Álvaro González Sotillo

17 de febrero de 2025

Índice

1. Nextcloud	1
2. Docker compose	1
3. Wordpress	2
4. Ejecutar <i>scripts</i>	3
5. Secretos	3
6. Instalar Nextcloud	3
7. Referencias	5

1. **Nextcloud**

- Es un *fork* de *owncloud*
- Paquete ofimático (a través de **Collabora**)
- Almacenamiento y compartición de ficheros
- Cliente de correo, *chat*

2. **Docker compose**

- Facilita la creación de *containers* interconectados (*stacks*)
- Facilita la creación de redes
- Usa ficheros *yaml*

2.1. **Nivel superior**

- `name`: Nombre del *stack* a crear
- `services`: Lista de *containers* a crear
- `networks`: Lista de redes que usarán los *containers*
- `volumes`: Lista de volúmenes, generalmente compartidos entre los *containers*
- Otros (`secrets`, `configs`)

2.2. Servicios

- Nombre del servicio
 - image: Imagen Docker
 - volumes: Volúmenes montados en este *container*
 - ports: mapeo de puertos TCP/UDP

2.3. Ejemplo

- Conectar desde el host con
 - `mysql --host localhost --port 1234 -u root -pcosmicsecret`

```
name: mariadb-stack
services:
  db:
    container_name: mariadb
    image: mariadb
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: asdasd
    ports:
      - 1234:3306
    volumes:
      - mariadb_data:/var/lib/mysql
volumes:
  mariadb_data:
```

2.4. Redes

- Todos los servicios de un *stack* comparten una red *bridge* propia
 - No hay comunicación entre redes distintas
 - Cada red simula estar detrás de un NAT, siendo el *host* el router
- Pueden definirse redes *bridge* adicionales
- DNS
 - Se añaden los nombres del *stack* a la resolución de DNS

3. Wordpress

```
name: wordpress-stack
services:
  wordpress:
    container_name: servidor_wp
    image: wordpress
    restart: always
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: user_wp
      WORDPRESS_DB_PASSWORD: asdasd
      WORDPRESS_DB_NAME: bd_wp
    ports:
      - 9999:80
    volumes:
      - wordpress_data:/var/www/html
  db:
    container_name: servidor_mysql
    image: mariadb
    restart: always
    environment:
      MARIADB_DATABASE: bd_wp
      MARIADB_USER: user_wp
      MARIADB_PASSWORD: asdasd
      MARIADB_ROOT_PASSWORD: asdasd
    volumes:
      - mariadb_data:/var/lib/mysql
```

```
volumes:
  wordpress_data:
  mariadb_data:
```

4. Ejecutar *scripts*

- Se pueden ejecutar *scripts* dentro un *container*
 - `post_start`
 - `pre_stop`
- Opciones:
 - `command`: El comando del *container* a ejecutar
 - `user`: Usuario con el que ejecutar el comando
 - `working_dir`: Directorio actual del comando
- Se pueden copiar comandos dentro del container con un volumen. Por ejemplo:
 - `volume: ./init.sh:/init.sh`

5. Secretos

- Variables de entorno
- Ficheros de entorno (`.env`)
- `secrets` (las imágenes de docker deben soportarlos)

5.1. Variables de entorno

- Se utilizan como en bash: `${VARIABLE}`
- Pueden tener un valor por defecto `${VARIABLE:-pordefecto}`
- El fichero `.env` se carga antes de ejecutar el fichero YAML

6. Instalar Nextcloud

- Containers:
 - Nexcloud:
 - Base de datos Mariadb
 - Redis: cache en memoria para rendimiento (opcional)
 - Opcional: proxy inverso apache/nginx para mejorar la seguridad
- **Ejemplos**

```
name: nextcloud-stack

services:
  db:
    container_name: nextcloud-db
    image: mariadb:10.11
    restart: always
    command: --transaction-isolation=READ-COMMITTED
    volumes:
      - db:/var/lib/mysql
    environment:
      - MARIADB_ROOT_PASSWORD=${MARIADB_ROOT_PASSWORD:-bydefault}
      - MARIADB_PASSWORD=${MARIADB_PASSWORD:-pordefecto}
      - MARIADB_DATABASE=nextcloud
      - MARIADB_USER=nextcloud
```

```

redis:
  container_name: nextcloud-redis
  image: redis:alpine
  restart: always

app:
  container_name: nextcloud-app
  image: nextcloud
  restart: always
  ports:
    - 8888:80

  depends_on:
    - redis
    - db

  volumes:
    - ./nextcloud-base:/var/www/html

  environment:
    - MYSQL_PASSWORD=${MARIADB_PASSWORD:-pordefecto}
    - MYSQL_DATABASE=nextcloud
    - MYSQL_USER=nextcloud
    - MYSQL_HOST=db
    - REDIS_HOST=redis

volumes:
  nextcloud:
  db:

```

6.1. Dirección de acceso

- Para evitar problemas, usaremos una IP estática en el host
- Usaremos esa IP para acceder al nextcloud
 - config/config.php , cambiar trusted_domains

```

'trusted_domains' =>
array (
  0 => 'localhost:8888',
  1 => '192.168.1.120:8888',
  3 => 'localhost:80'
),

```

6.2. Instalar Collabora

- App: nextcloud office
 - Puede conectarse a un servidor de documentos y edición
- APP: Collabora Online - Built-in CODE Server
 - Es un servidor de edición de documentos integrado
 - Pueden instalarse otros por separado
- Settings: Nextcloud Office
 - Usar "Built in CODE"

6.3. WOPI

Could not establish connection to the Collabora Online server.

Failed to connect to the remote server: cURL error 7: Failed to connect to localhost port 8888

You have not configured the allow-list for WOPI requests. Without this setting users may down

- Hay que poner, o la IPdelhost:puerto_mapeado

7. Referencias

- Formatos:
 - [Transparencias](#)
 - [PDF](#)
 - [Página web](#)
 - [EPUB](#)
- Creado con:
 - [Emacs](#)
 - [org-re-reveal](#)
 - [Latex](#)
- Alojado en [Github](#)

<https://plataforma.josedomingo.org/pledin/cursos/docker2024/>