

4.1 NextCloud. Tu nube personal

Nextcloud_Logo.svg.png

Imagen obtenida de https://es.m.wikipedia.org/wiki/Archivo:Nextcloud_Logo.svg

Esta herramienta sirve para...

Una vez mas vamos a recurrir a la wikipedia:

“ Nextcloud es una serie de programas cliente-servidor que permiten la creación de servicios de alojamiento de archivos. Su funcionalidad es similar al software Dropbox, aunque Nextcloud es en su totalidad software de código abierto. Nextcloud permite a los usuarios crear servidores privados. Su modelo de desarrollo abierto permite añadir y/o modificar la funcionalidad del software del servidor en forma de aplicaciones.

<https://es.wikipedia.org/wiki/Nextcloud>

Y según lo anterior me podréis decir que en el apartado [3.9 File Browser. Explorador de ficheros en remoto](#) ya nos hemos dotado de una muy buena herramienta web que nos permite gestionar nuestros ficheros en remoto. En esta ocasión vamos a ver una herramienta que mejora las posibilidades (y aumenta la complejidad) de la anterior ya que cuenta entre otras con las siguientes características:

“ • Los archivos Nextcloud son almacenados en estructuras de directorio convencionales y se pueden acceder a través del protocolo WebDAV si

es necesario.

- Los archivos son encriptados en la transmisión y opcionalmente durante el almacenamiento.
- Los usuarios pueden manejar calendarios (CalDAV), contactos (CardDAV), tareas programadas y reproducir contenido multimedia (Ampache).
- Permite la administración de usuarios y grupos de usuarios (vía OpenID o LDAP) y definir permisos de acceso.
- Posibilidad de añadir aplicaciones (de un solo clic) y conexiones con Dropbox, Google Drive y Amazon S3.
- Disponibilidad de acceso a diferentes bases de datos mediante SQLite, MariaDB, MySQL, Oracle Database, y PostgreSQL.
- Disponibilidad de un software llamado Nextcloud box basado en Raspberry Pi que funciona en Ubuntu Core.
- Posibilidad de integrar los editores en línea ONLYOFFICE mediante la aplicación oficial.

<https://es.wikipedia.org/wiki/Nextcloud>

que permiten dar un paso adelante con respecto a la solución que con anterioridad habíamos visto.

Web de proyecto y otros enlaces de interés

Página web obficial del proyecto: <https://nextcloud.com/>

Repositorio de código: <https://github.com/nextcloud/server>

Instrucciones de instalación en un servidor propio: <https://nextcloud.com/install/#instructions-server>

Despliegue

Como en ocasiones anteriores vamos a hacer con docker-compose para ello accedemos al terminal y escribimos

```
cd $HOME
mkdir nextcloud
cd nextcloud
nano docker-compose.yml
```

y dentro del fichero escribiremos el siguiente contenido (adaptado de <https://github.com/nextcloud/all-in-one/blob/main/docker-compose.yml>):

```
version: "3.7"

volumes:
  nextcloud_aio_mastercontainer:
    name: nextcloud_aio_mastercontainer # This line is not allowed to be changed

services:
  nextcloud:
    image: nextcloud/all-in-one:latest
    restart: always
    container_name: nextcloud-aio-mastercontainer # This line is not allowed to be changed
    volumes:
      - nextcloud_aio_mastercontainer:/mnt/docker-aio-config # This line is not allowed to be
changed
      - /var/run/docker.sock:/var/run/docker.sock:ro # May be changed on macOS, Windows or
docker rootless. See the applicable documentation. If adjusting, don't forget to also set
'DOCKER_SOCKET_PATH' !
    ports:
      - 80:80 # Can be removed when running behind a web server or reverse proxy (like Apache,
Nginx and else). See https://github.com/nextcloud/all-in-one/blob/main/reverse-proxy.md
      - 8080:8080
      - 8443:8443 # Can be removed when running behind a web server or reverse proxy (like
Apache, Nginx and else). See https://github.com/nextcloud/all-in-one/blob/main/reverse-
proxy.md
```

como en ocasiones anteriores, para guardar los cambios pulsaremos `control + x` y cuando nos pregunte aceptaremos.

Si algún otro servicio está utilizando los puertos que en este servicio vamos a utilizar se generará un conflicto y puede que ninguno de los servicios funcione o, mas probable, el último que pongamos en marcha.

Para usar un puerto diferente puedes cambiar el valor que aparece ANTES de los : por un valor que no esté en uso. También puedes acceder al directorio dónde se encuentra el otro servicio y ejecutar `docker-compose down`.

Posteriormente ponemos en marcha los contenedores con `docker-compose up -d` (le costará un buen rato).

[nextcloud-deploy-error.png](#)

Elaboración propia

En esta ocasión el despliegue nos ha fallado por lo que comentaba en la advertencia anterior. El mensaje de error nos indica que el puerto 8080 ya está en uso. Con `docker ps` voy a ver qué contenedores están utilizando qué puertos:

[docker-ps.png](#)

Elaboración propia

Aquí veo que el servicio pihole es quién está usando actualmente ese puerto. Posibles soluciones:

- Detener pi-hole: accedo al directorio dónde está pi-hole y ejecuto `docker-compose down`. Esto hará que deje de funcionar.
- Cambiar en el fichero docker-compose.yml de nextcloud la línea que dice 8080:8080 por, por ejemplo, 8081:8080. Nosotros optaremos por esta segunda opción

tomemos la decisión que tomemos deberemos volver a acceder al directorio de nextcloud y ejecutar `docker-compse up -d`. Si ahora va bien veremos algo como:

[nextclod-deploy.png](#)

Elaboración propia

Si accedemos, como en ocasiones anteriores, a `http://IP:PUERTO` siendo en mi caso <http://192.168.0.201:8081> veremos un mensaje de error que nos indica que utilicemos el protocolo https así pues accederemos a `https://IP:PUERTO` que en mi caso es <https://192.168.0.201:8081> es probable que nos aparezca un mensaje de certificado no válido, no hay problema, le indicamos que

continúe y veremos algo como:

[nextcloud-deployed.png](#)

Elaboración propia

Funcionamiento

Ahora la contraseña que genera nextcloud. En mi imagen fíjate que aparece justo encima del botón. Tras acceder llegaremos a una pantalla como la siguiente en la cual deberemos indicar un dominio que apunte a nuestra raspberry pi (podemos utilizar alguno que hayamos configurado con duckdns)

[nextcloud-login.png](#)

Elaboración propia

Podemos trampear este paso si al final de nuestro fichero docker-compose.yml añadimos:

```
environment:  
  SKIP_DOMAIN_VALIDATION: 'true'
```

Para recargar los cambios en el fichero docker-compose.yml deberemos tirar el servicio con `docker-compose down` y volver a levantarlo con `docker-compose up -d`. Si volvemos a acceder a la url anterior veremos algo como:

[nextcloud-settingup.png](#)

Elaboración propia

Si pulsamos en `start containers` (le costará un buen rato) se pondrá en marcha el servicio y ya podremos disfrutar del mismo.

Quizás una Raspberry Pi modelo 4 de 4 GB, como es mi caso, resulta insuficiente para un servicio tan potente como el descrito. Si ejecutamos el comando `htop` veremos que está saturada a mas no poder.



Revision #9

Created 2023-02-04 10:59:23 CET by Pablo Ruiz

Updated 2023-03-22 23:32:36 CET by Pablo Ruiz