

Memoria del Trabajo de Fundamentos de Redes

Sergio Quijano Rey, Daniel González

Motivación

La motivación que hay detrás de este trabajo es la de estudiar cómo funcionan los productos de almacenamiento en la nube, a través de la creación de un servidor bastante básico de almacenamiento en red sobre nuestro propio servidor.

Para ello hemos usado una *Raspberry Pi 3* en la que hemos instalado el programa de software libre *Nextcloud*. Sobre este servidor muy básico hemos tenido que realizar ciertas labores de administración, más propias de una asignatura como *Ingeniería de Servidores*, y otras labores centradas en aspectos de redes.

Esto nos ha permitido conocer y aprender hasta cierto punto los aspectos a tener en cuenta para crear infraestructuras de este tipo, centrándonos especialmente en aspectos específicos de las redes.

Especificación del hardware usado

El hardware que hemos usado es el siguiente:

- ▶ Raspberry Pi 3 Model B:
 - ▶ CPU Quad Core 1,2GHz Broadcom BCM2837 64bit
 - ▶ 1GB RAM
 - ▶ Tarjeta SD *Samsung Evo*, 64 GB (actúa como disco duro del ordenador)
- ▶ Cargador de móvil *MicroUSB*: dará corriente a la *Raspberry Pi 3*

Raspberry Pi 3 Model B

Instalación básica del servidor

Instalación del sistema operativo

- ▶ Ubuntu Server: sin interfaz gráfica para ahorrar recursos
- ▶ Descargamos el .img del siguiente link

- ▶ Vemos dónde está localizada la tarjeta SD con `lsblk`
- ▶ Quemamos la imagen con el comando: `sudo dd`
`if="ubuntu-raspberry.img" of="/dev/sdb" bs=4M`
`status=progress`
 - ▶ `if`: archivo de entrada
 - ▶ `of`: archivo de salida
 - ▶ `bs`: tamaño del bloque
 - ▶ `status=progress`: para ver la barra de progreso
- ▶ Se podría usar una herramienta con GUI como `etcher`

Configuración inicial del sistema

Ahora introducimos la tarjeta SD en la *Raspberry Pi* y conectamos el cargador, tanto a la *Raspberry Pi* como a la corriente. También conectamos un cable ethernet desde el router hasta la *Raspberry Pi*, para tener conexión a internet

Gracias a que estamos usando *Ubuntu Server*, por defecto tiene el servicio ssh habilitado por defecto. Así que entramos al administrador del router escribiendo en la barra de nuestro navegador la dirección 192.168.1.1. Tras esto podemos localizar su dirección local, como se muestra en la **Imagen 1**:

Imagen 1

Hacemos ssh a nuestro servidor con la dirección que acabamos de obtener: `ssh ubuntu@192.168.1.8`. `ubuntu` es el usuario por defecto que está creado, además de `root`. Este usuario está en la lista de `sudoers`, por lo que tenemos privilegios de administración a través de él. Pero para facilitar las cosas, nos registramos como `root` usando el comando `sudo su -`.

El primer paso es hacer una primera actualización del sistema con `apt update`; `apt upgrade`, e instalar algunos paquetes básicos para trabajar, como pueden ser `vim` o `make`

Aspectos de Redes en la instalación del servidor

Protección básica del servidor y SSH

En lo que sigue, vamos a dar conexión a nuestro servidor para que sea alcanzable desde fuera de la red local. Por ello, hay que dar una seguridad básica para evitar ciertos ataques que comprometan nuestro servidor. Ahora mismo, el mayor problema que tenemos es que se pueden hacer ataques de fuerza bruta en `ssh` para logearse, probando todas las posibles contraseñas. Y esto es aún más peligroso teniendo en cuenta de que se puede hacer directamente sobre el usuario `root`. Aunque ahora esto último no se puede hacer, porque `root` no tiene contraseña, si en algún momento se le asigna una contraseña, esto sería posible. Por tanto, estos son los problemas más básicos (que ni remotamente son los únicos) que vamos a tratar de solventar.

Para ello vamos a usar `ssh`. `ssh` puede usar claves de cifrado asimétrico. Para generar estas claves, desde nuestros ordenadores con los que accedemos al servidor, lanzamos el comando `ssh-keygen`, que genera los archivos `~/.ssh/id_rsa.pub` (clave pública) y `~/.ssh/id_rsa` (clave privada). La encriptación que se

Instalación de Nextcloud

Uso básico de Nextcloud

Análisis del sistema

Referencias