



Proyecto AS - Servidores FTP y Samba

Administrador de Servidores

Grupo G10

Antonio Rodríguez Palacio

Iván Guillén Márquez

Alejandro Parada Espinosa

CURSO 2024/25

Capítulo 1. Servidores Samba y FTP

1.1. Servidor Samba

Para montar nuestro servidor Samba hemos utilizado la imagen de Docker `dperson/samba`, la imagen más utilizada de la plataforma Docker Hub para Samba.

Esta imagen permite configurar el contenedor tanto mediante variables de entorno como a través de los parámetros del script incluido.

El contenedor forma parte de la red de producción, cuenta con un volumen para cada directorio que utiliza.

En una primera iteración, utilizamos el apartado `environment` de nuestro fichero Docker `compose` para crear los empleados con sus respectivas contraseñas.

Realizamos el resto de la configuración de nuestro servidor mediante parámetros del script (`init_samba.sh`), con los que ajustábamos de cada directorio compartido su nombre y los permisos del mismo.

Sin embargo al probar esta versión teníamos problemas para acceder que, al indagar, descubríamos que estaban relacionados a la forma en la que creábamos los usuarios, ya que si reducíamos los permisos ganábamos acceso a pesar de que el usuario en cuestión debería tener los necesarios. Todo resulto ser un problema de las IDs que se asignaban a los usuarios en el momento de creación, que no se correspondían el que `chmod` asignaba como propietario del archivo.

Al final solucionamos el problema moviendo la declaración de los usuarios de la sección `environment` de nuestro `compose` a los parámetros del script y añadiendo el parámetro `"-p"`, que encarga al script del trabajo de la gestión de permisos en los shares.

```
samba:
  image: dperson/samba
  container_name: samba_services
  volumes:
    - ./vol/samba/desarrollo:/mnt/desarrollo
    - ./vol/samba/revision:/mnt/revision
    - ./vol/samba/publico:/mnt/publico
    - ./vol/samba/init_samba.sh:/init_samba.sh
  networks:
    development_net:
      ipv4_address: 172.40.0.30
  dns: 172.20.0.10
  command: bash /init_samba.sh
```

Listing 1.1: Contenedor Samba

```
#!/bin/bash
for i in 1 2 3 4 5; do
    mkdir -p /mnt/desarrollo/SW$i
             /mnt/revision/SW$i
             /mnt/publico/SW$i
done
exec samba.sh \
  -p \
  -u 'empleado1;password1' \
  -u 'empleado2;password2' \
  -u 'empleado3;password3' \
  -u 'empleado4;password4' \
  -u 'empleado5;password5' \
  -u 'revisor;revisor' \
  -s 'SW1;/mnt/desarrollo/SW1;yes;no;no;empleado1' \
  -s 'SW2;/mnt/desarrollo/SW2;yes;no;no;empleado2' \
  -s 'SW3;/mnt/desarrollo/SW3;yes;no;no;empleado3' \
  -s 'SW4;/mnt/desarrollo/SW4;yes;no;no;empleado4' \
  -s 'SW5;/mnt/desarrollo/SW5;yes;no;no;empleado5' \
  -s 'rev1;/mnt/revision/SW1;yes;no;no;empleado1,revisor' \
  -s 'rev2;/mnt/revision/SW2;yes;no;no;empleado2,revisor' \
  -s 'rev3;/mnt/revision/SW3;yes;no;no;empleado3,revisor' \
  -s 'rev4;/mnt/revision/SW4;yes;no;no;empleado4,revisor' \
  -s 'rev5;/mnt/revision/SW5;yes;no;no;empleado5,revisor' \
  -s 'publico;/mnt/publico;yes;yes;no;;;revisor' \
```

Listing 1.2: init_samba.sh

```
root@e769fd80f403:/var/www/html# smbclient //172.40.0.30/rev2 -U revisor
Enter WORKGROUP\revisor's password:
Try "help" to get a list of possible commands.
smb: \> smbclient //172.40.0.30/pub -U revisor
root@e769fd80f403:/var/www/html# smbclient //172.40.0.30/rev2 -U revisor
Enter WORKGROUP\revisor's password:
root@e769fd80f403:/var/www/html# smbclient //172.40.0.30/rev2 -U revisor
Enter WORKGROUP\revisor's password:
Try "help" to get a list of possible commands.
smb: \> !smbclient //172.40.0.30/publico -U revisor
Enter WORKGROUP\revisor's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Wed May  7 19:12:04 2025
..               D           0   Wed May  7 19:25:25 2025
SW1              D           0   Wed May  7 19:12:04 2025
SW2              D           0   Wed May  7 19:35:34 2025
SW3              D           0   Wed May  7 19:12:04 2025
SW4              D           0   Wed May  7 19:12:04 2025
SW5              D           0   Wed May  7 19:12:04 2025

487094816 blocks of size 1024. 193494380 blocks available
smb: \> cd SW2
smb: \SW2\> █
```

Figura 1.1: Captura entrando como revisor en samba

```
root@8e25a66744ed:/var/www/html# smbclient //172.40.0.30/SW3 -U empleado3
Enter WORKGROUP\empleado3's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Tue May 13 20:08:14 2025
..               D           0   Wed May  7 19:12:04 2025
ejemplo.tex      N           0   Tue May 13 20:08:05 2025

487094816 blocks of size 1024. 184956348 blocks available
smb: \> █
```

Figura 1.2: Captura entrando como el empleado 3

1.2. Servidor FTP

De la misma manera que con el servidor de Samba, comenzamos buscando imágenes disponibles en docker hub ya configuradas. Encontramos bogem/ftp, pero no permitía tener usuario anónimo.

Tras de investigar esta imagen, descubrimos que estaba basada en otra imagen fauria/vsftpd, pero descubrimos que esta tampoco nos permitía el uso de usuario anónimo.

Para no desperdiciar nuestra búsqueda intentamos sobrescribir el archivo de configuración de `vsftpd.conf` de la imagen, pero no funcionó.

En ultima instancia, terminamos construyendo nuestra propia imagen a partir de la imagen base de Debian. La imagen esta definida en el siguiente dockerfile:

```
FROM debian:latest

RUN apt-get update && \
    apt-get install -y vsftpd

RUN mkdir -p /var/run/vsftpd/empty && \
    mkdir -p /home/publico && \

    chown ftp:ftp /home/publico && \
    chmod 755 /home/publico

COPY vol/samba/vsftpd.conf /etc/vsftpd/vsftpd.conf

EXPOSE 21

CMD ["/usr/sbin/vsftpd", "/etc/vsftpd/vsftpd.conf"]
```

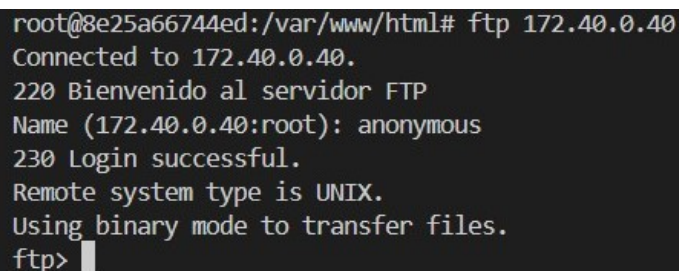
Listing 1.3: ftp.Dockerfile

En el archivo `vsftpd.conf` establecemos el directorio raíz en `/home/publico`, la opción de acceder con usuario anónimo sin contraseña, y vetamos a los usuarios moverse más allá de sus directorios locales.

```
listen=YES
listen_ipv6=NO
anonymous_enable=YES
no_anon_password=YES
anon_root=/home/publico
local_enable=YES
write_enable=YES
local_umask=022
anon_upload_enable=NO
anon_mkdir_write_enable=NO
chroot_local_user=YES
allow_writeable_chroot=YES
user_sub_token=$USER
local_root=/home/$USER/ftp
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=40005
ftpd_banner=Bienvenido al servidor FTP
```

Listing 1.4: `vsftpd.conf`

```
ftp:
  build:
    dockerfile: ftp.Dockerfile
  container_name: ftp_publico
  volumes:
    - ./vol/samba/publico:/home/publico:ro
  networks:
    development_net:
      ipv4_address: 172.40.0.40
```

Listing 1.5: `ftp_público`

```
root@8e25a66744ed:/var/www/html# ftp 172.40.0.40
Connected to 172.40.0.40.
220 Bienvenido al servidor FTP
Name (172.40.0.40:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

Figura 1.3: Captura entrando en ftp como anónimo

1.3. Acceso seguro a los servidores

En la primera practica, donde montamos la arquitectura base que estamos expandiendo poco a poco, añadimos la posibilidad a los usuarios de nuestro sistema a conectarse a través de una VPN a las redes internas de production, development y service en base a sus permisos.

A traves de esta VPN los usuarios son capaces de acceder de forma facil y segura a los servidores de esta practica, sin añadir nada a la infraestructura que ya hemos montado.