

SERVIDORES WEB Y DE ALTA DISPONIBILIDAD SWAP (2021-2022)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 1

Pedro Antonio Mayorgas Parejo

9 de junio de 2022

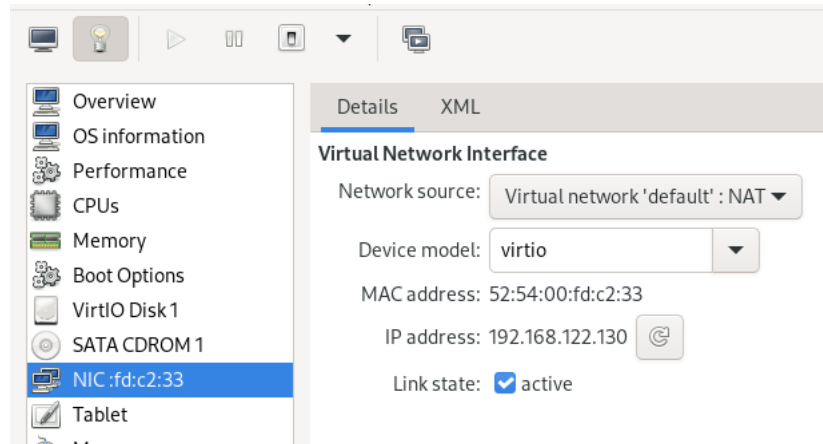
Índice

1	Configuración de las Interfaces en Máquinas Virtuales	3
1.1	Configuración de netplan	3
2	Instalación y configuración de servicios de apache	5
2.1	Configuración de cambio de puerto	5
2.2	Configuración de un VirtualHost por defecto	7
2.2.1	Creando el directorio del VirtualHost	7
2.2.2	Activando y desactivando el VirtualHost	8
3	Instalación y configuración de SSH server y SSH client	12
3.1	Configuración de cambio de puerto	12
3.2	Configuración de acceso por autenticación de par de clave pública/privada	12
3.3	Comandos del cliente de SSH	14
3.3.1	Conexión SSH	14
3.3.2	Creando par de claves y conexión sin contraseña	15
4	Instalación y uso de cURL	17
4.1	Opciones básicas sobre http:	17
4.2	Opciones avanzadas de cURL	17
4.2.1	Cookies	17
4.2.2	Manejo de peticiones GET y POST de HTTP	17
4.2.3	Uso para servidores web en otros puertos	18
5	Bibliografía	19

1. Configuración de las Interfaces en Máquinas Virtuales

El sistema Hypervisor que utilizo, es QEMU/KVM, donde la configuración NAT, al contrario que en la de VirtualBox. Se comporta como la suma del adaptador Host-Only + NAT de VirtualBox.

La interfaz nos ofrece la configuración por DHCP y configurarlo de manera estática en el sistema.



1.1. Configuración de netplan

Tenemos un fichero localizado en `/etc/netplan/config.yaml`, dicho fichero, lo configuramos como viene en la siguiente foto, para poder poner una dirección IP estática.

```
GNU nano 5.4 /etc/netplan/config.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      dhcp4: no
      addresses: [192.168.122.130/24]
      gateway4: 192.168.122.1
      nameservers:
        addresses: [192.168.122.1, 8.8.8.8]
```

Ahora aplicamos la configuración de netplan con el siguiente comando:

```
sudo netplan apply
```

```
pedroamp@m1-pedroamp:~$ sudo nano /etc/netplan/config.yaml
[sudo] password for pedroamp:
pedroamp@m1-pedroamp:~$ sudo netplan apply
pedroamp@m1-pedroamp:~$
```

Con el resultado mostrado en `iproute2`.

```

pedroamp@m1-pedroamp:~$ ip route show
default via 192.168.122.1 dev enpls0 proto static
192.168.122.0/24 dev enpls0 proto kernel scope link src 192.168.122.130
pedroamp@m1-pedroamp:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enpls0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
te UP group default qlen 1000
    link/ether 52:54:00:fd:c2:33 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.130/24 brd 192.168.122.255 scope global dynamic enpls
0
        valid_lft 3348sec preferred_lft 3348sec
    inet6 fe80::5054:ff:fe80:c233/64 scope link
        valid_lft forever preferred_lft forever

```

2. Instalación y configuración de servicios de apache

Para instalar el servicio de apache2 y el SGBD que es Mariadb un fork de MySQL necesitamos solo el siguiente comando.

```
sudo apt-get install apache2 mariadb-server mariadb-client
```

```
pedroamp@m2-pedroamp:~$ sudo apt-get install apache2 mariadb-server mariadb-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils galera-4 gawk libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libcgi-fast-perl libcgi-pm-perl
  libclone-perl libconfig-inifiles-perl libdbd-mariadb-perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblua5.3-0
  liblwp-mediatypes-perl libmariadb3 libmpfr6 libsigsegv2 libsnappy1v5
  libterm-readkey-perl libtimedate-perl liburi-perl mariadb-client-10.5
  mariadb-client-core-10.5 mariadb-common mariadb-server-10.5
  mariadb-server-core-10.5 mysql-common psmisc rsync socat ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  gawk-doc libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx mariadb-test
  netcat-openbsd
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils galera-4 gawk libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcgi-fast-perl
  libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdbd-mariadb-perl
  libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblua5.3-0
  liblwp-mediatypes-perl libmariadb3 libmpfr6 libsigsegv2 libsnappy1v5
  libterm-readkey-perl libtimedate-perl liburi-perl mariadb-client
  mariadb-client-10.5 mariadb-client-core-10.5 mariadb-common
  mariadb-server mariadb-server-10.5 mariadb-server-core-10.5 mysql-common
  psmisc rsync socat ssl-cert
0 upgraded, 47 newly installed, 0 to remove and 0 not upgraded.
Need to get 22.0 MB of archives.
After this operation, 169 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

2.1. Configuración de cambio de puerto

Para poder realizar un cambio de puerto, podemos ver en la ruta `/etc/apache2/ports.conf`. Donde tiene un fichero especialmente dedicado, a la escucha de los puertos.

```

GNU nano 5.4 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

En este caso cambiamos el puerto por defecto del 80 al 8080, para ver cómo le afecta al servicio HTTPd.

```

GNU nano 5.4 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Le indicamos al sistema que el demonio vuelva a leer la información contenida dentro de los .conf. Para que se configure el puerto nuevo, asignado previamente.

```
sudo systemctl reload apache2
```

Creo la página index.html conforme al guión de prácticas.

```

GNU nano 5.4 /var/www/html/index.html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>pedroamp Default Page: It works</title>
  </head>
  <body>
    Web de Ejemplo: pedroamp para SWAP.
    Email: pedroamp@correo.ugr.es
    Default Web working on PORT 8080
  </body>
</html>

```

Modifico el fichero index.html localizado en /var/www/html/index.html, para que muestre mis datos y en el servidor m2-pedroamp, lanzo el comando cURL de manera sencilla para verificar el puerto.

```
pedroamp@m2-pedroamp:~$ curl 192.168.122.130:8080

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>pedroamp Default Page: It works</title>
  </head>
  <body>
    Web de Ejemplo: pedroamp para SWAP.
    Email: pedroamp@correo.ugr.es
    Default Web working on PORT 8080
  </body>
</html>
```

Podemos ver que el puerto ha cambiado. Mostrando que si lo usamos en cualquier otro. No funciona.

```
pedroamp@m2-pedroamp:~$ curl 192.168.122.130
curl: (7) Failed to connect to 192.168.122.130 port 80: Connection refused
pedroamp@m2-pedroamp:~$
```

También se puede, configurar desde el VirtualHost el puerto en el cual se desee escuchar las peticiones, en el caso de que queramos no usar un nombre concreto y queramos usar la IP para direccionar a ese VirtualHost.

2.2. Configuración de un VirtualHost por defecto

Anteriormente hemos configurado el puerto por defecto de Apache2 del index. Ahora vamos a configurar un VirtualHost servido en el host m2-pedroamp, donde configuramos primero su fichero de configuración. Para finalmente en /etc/hosts, poner el nombre de acceso a dicho VirtualHost tanto en m1-pedroamp como en m2-pedroamp

2.2.1. Creando el directorio del VirtualHost

Tenemos que ir a /var/www, que es el directorio desde donde usualmente Apache2 o HTTPd en su defecto lee los ficheros a servir a los clientes que lo soliciten.

El usuario por defecto que viene para Apache2 es www-data:www-data, por lo que cuando accedamos a /var/www/. Dicha carpeta raíz es del usuario root:root. Por lo que tenemos que gastar cuidado al crear cualquier documento, porque tenemos que darle permisos al grupo www-data con el siguiente comando:

```
sudo chown root:www-data -R /var/www/swap/swap.html
```

Ya que de lo contrario, el servidor no puede servir los archivos contenidos dentro del directorio.

Contenido del fichero /var/www/swap/swap.html, que será el documento servido por el servidor m2-pedroamp.

```

pedroamp@m2-pedroamp:/var/www/swap$ sudo cat swap.html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www
.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>pedroamp Default Page: It works</title>
  </head>
  <body>
    Web de Ejemplo: pedroamp para SWAP.
    Email: pedroamp@correo.ugr.es
    Default Web working on PORT 8080
  </body>
</html>

pedroamp@m2-pedroamp:/var/www/swap$ █

```

Cambio de permisos para permitir al servidor servir el documento y demostración del cambio realizado.

```

pedroamp@m2-pedroamp:/var/www$ sudo chown -R root:www-data swap/swap.html
pedroamp@m2-pedroamp:/var/www$ ls -la swap/
total 12
drwxr-xr-x 2 root root 4096 Mar  4 16:47 .
drwxr-xr-x 4 root root 4096 Mar  4 16:46 ..
-rw-r--r-- 1 root www-data 368 Mar  4 16:49 swap.html
pedroamp@m2-pedroamp:/var/www$ █

```

Ahora debemos indicarle al VirtualHost la ruta de los ficheros a servir en su fichero de configuración.

2.2.2. Activando y desactivando el VirtualHost

En la información consultada en: `/usr/share/doc/apache2/README.Debian.gz` Tenemos el árbol de apache2 o en su defecto HTTPd. Donde se nos indica que en los siguientes directorios:

- `sites-available/` -> Contiene las directivas de VirtualHost preparadas pero no activadas. Conteniendo un fichero llamado '000-default.conf' que es el VirtualHost por defecto provisto por Debian.
- `sites-enabled/` -> Contiene las mismas directivas de VirtualHost que en el anterior, la diferencia es que es un enlace simbólico donde dicho enlace indica qué VirtualHost quiere activado el administrador.

Por supuesto existe un symlinks (enlace simbólico) del VirtualHost por defecto 000-default.conf.

Example:

- `/etc/apache2/sites-enabled/swap.conf -> /etc/apache2/sites-available/swap.conf`

Las herramientas utilizadas dentro de sites-avaiable *esto es obligatorio, es decir los comandos son:

- a2ensite swap.conf -> Activa el VirtualHost
- a2disite swap.conf -> Desactiva el VirtualHost

Las herramientas anteriores, hacen realmente lo mismo que un `ln -s` para crear un enlace simbólico con algunos extra.

Primero desactivamos el sitio servido por defecto en m2-pedroamp primeramente tenemos que estar en el directorio en `/etc/apache2/sites-available`.

```
sudo a2dissite 000-default.conf
```

Demostración y luego recargamos el servicio de Apache2:

```
pedroamp@m2-pedroamp:/etc/apache2/sites-available$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Creamos el fichero del VirtualHost nuevo que llamaremos: swap.conf. Donde tiene el siguiente contenido:

```
GNU nano 5.4 /etc/apache2/sites-available/swap.conf
<VirtualHost *:8886>
    #Listen 8886 incluido en ports.conf
    ServerAdmin pedroamp@correo.ugr.es
    DocumentRoot /var/www/swap

    ServerName m2-pedroamp.org
    ServerAlias www.m2-pedroamp.org

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- El puerto donde se sirve el VirtualHost es el 8886, es otra forma de servir contenido en puertos distintos usando la misma IP. El puerto debe ser indicado en `ports.conf` o directamente en el VirtualHost
- Si usamos el nombre, tenemos la directiva `ServerName` y `ServerAlias`. Servimos el contenido que se solicite bajo ese nombre.
- El directorio es distinto al indicado por defecto y ya creado en el punto anterior.

Fichero de configuración `ports.conf` en m2-pedroamp.org.

```

GNU nano 5.4 /etc/apache2/ports.conf
# If you just change the port or add more ports here
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8886

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

```

Ahora activamos el VirtualHost, con la herramienta indicada antes y comprobamos el estado del servicio por si hubiera errores.

```

pedroamp@m2-pedroamp:/etc/apache2/sites-available$ sudo a2ensite swap.conf
Enabling site swap.
To activate the new configuration, you need to run:
    systemctl reload apache2
pedroamp@m2-pedroamp:/etc/apache2/sites-available$ sudo systemctl reload a
pache2
pedroamp@m2-pedroamp:/etc/apache2/sites-available$ sudo systemctl status a
pache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor>
   Active: active (running) since Fri 2022-03-04 16:05:03 CET; 1h 54min>
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 3299 ExecReload=/usr/sbin/apachectl graceful (code=exited, s>

```

Finalmente en el servidor y cliente se tienen que modificar el fichero /etc/hosts. Porque no tenemos un servicio de nombres DNS propio que se encargue de ser la autoridad del nombre SOA. Que resuelva el nombre y la IP.

En m1-pedroamp:

```

GNU nano 5.4 /etc/hosts
127.0.0.1 localhost
127.0.1.1 m1-pedroamp

192.168.122.235 m2-pedroamp.org
192.168.122.235 www.m2-pedroamp.org

```

En m2-pedroamp:

```

GNU nano 5.4 /etc/hosts
127.0.0.1 localhost
127.0.1.1 m2-pedroamp

127.0.0.1 m2-pedroamp.org
127.0.0.1 www.m2-pedroamp.org

```

Ahora uso curl en m1-pedroamp, para ver si se sirve el contenido con el nombre y el puerto. Podemos ver que si usamos la dirección por defecto o la IP, nos dará un error ya que el puerto 80 no está siendo usado para escuchas (Listen).

```
pedroamp@m1-pedroamp:~$ curl http://www.m2-pedroamp.org
curl: (7) Failed to connect to www.m2-pedroamp.org port 80: Connection refused
pedroamp@m1-pedroamp:~$ curl http://www.m2-pedroamp.org:8886

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>pedroamp m2-pedroamp.org Default Page: It works</title>
  </head>
  <body>
    Web de Ejemplo: pedroamp para SWAP. VirtualHost m2-pedroamp.org
    Email: pedroamp@correo.ugr.es
    Default Web working on PORT 8080
  </body>
</html>
```

3. Instalación y configuración de SSH server y SSH client

Para la instalación de OpenSSH server necesitamos los siguientes paquetes:

```
apt-get install openssh-server openssh-client
```

Instalación en el nodo ml-pedroamp

```
pedroamp@ml-pedroamp:~$ sudo apt-get install openssh-server openssh-client
[sudo] password for pedroamp:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:8.4p1-5).
The following additional packages will be installed:
  libwrap0 openssh-sftp-server runit-helper
Suggested packages:
  molly-guard monkeysphere ssh-askpass ufw
The following NEW packages will be installed:
  libwrap0 openssh-server openssh-sftp-server runit-helper
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/505 kB of archives.
After this operation, 1,877 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

3.1. Configuración de cambio de puerto

Para la configuración del cambio de puerto, debemos ir a `/etc/ssh/sshd_config`:
Donde cambiamos el puerto por defecto que es el 22 al 4444.

- Port 22 ->Port 4444

```
GNU nano 5.4 /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 4444
```

Finalmente para poder aplicar los cambios necesitamos reiniciar el servicio, que tiene el demonio de SSHd para poder aplicar los cambios.

```
sudo systemctl restart openssh-server
```

3.2. Configuración de acceso por autenticación de par de clave pública/privada

Para permitir el acceso por par de claves pública/privada, necesitamos ir al mismo fichero en `/etc/ssh/sshd_config`. Donde ponemos en yes las siguientes opciones:

- PubkeyAuthentication yes
- PasswordAuthentication yes

¿Por qué también PasswordAuthentication? Porque el reto (challenge), que se hace con ssh-copy-id, para pasar la clave pública para autenticarse. Necesita una autenticación con contraseña de manera que la clave que vaya a recibir sea auténtica y del administrador. Posteriormente una vez terminado el intercambio de par de claves, se puede desactivar la autenticación con contraseña. Para asegurar al 100 %, que solo se permite por par de claves.

```
GNU nano 5.4 /etc/ssh/sshd_config
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
```

Finalmente reiniciamos el demonio SSHd para poder aplicar los cambios.

```
sudo systemctl restart openssh-server
```

En la imagen de abajo podemos ver que se han cambiado los puertos por defecto al comprobarlo con el status de systemctl. Que indica, el estado en el que se encuentra el servicio/demonio.

```
sudo systemctl status openssh-server
```

```

pedroamp@m1-pedroamp:~$ sudo systemctl restart sshd.service
pedroamp@m1-pedroamp:~$ sudo systemctl status sshd.service
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-03-01 18:24:08 CET; 4s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 1183 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1184 (sshd)
    Tasks: 1 (limit: 2312)
   Memory: 1.1M
      CPU: 21ms
   CGroup: /system.slice/ssh.service
           └─1184 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Mar 01 18:24:08 m1-pedroamp systemd[1]: Stopping OpenBSD Secure Shell server...
Mar 01 18:24:08 m1-pedroamp sshd[1184]: Server listening on 0.0.0.0 port 4444.
Mar 01 18:24:08 m1-pedroamp systemd[1]: ssh.service: Succeeded.
Mar 01 18:24:08 m1-pedroamp sshd[1184]: Server listening on :: port 4444.
Mar 01 18:24:08 m1-pedroamp systemd[1]: Stopped OpenBSD Secure Shell server.
Mar 01 18:24:08 m1-pedroamp systemd[1]: Starting OpenBSD Secure Shell server...
Mar 01 18:24:08 m1-pedroamp systemd[1]: Started OpenBSD Secure Shell server.
pedroamp@m1-pedroamp:~$ q_

```

3.3. Comandos del cliente de SSH

3.3.1. Conexión SSH

En la configuración anterior, conocemos que nuestro servidor tiene el puerto 4444. Pero el comando por defecto intenta conectarse al puerto 22.

Para ello tenemos que especificar como argumento el puerto concreto, en el cual el servidor SSH se encuentra escuchando peticiones.

```
ssh pedroamp@192.168.122.130 -p 4444
```

Nota: Si no indicamos el usuario en el comando, se coge el \$USER de tu sistema.

```

pedroamp@m2-pedroamp:~$ ssh 192.168.122.130 -p 4444
The authenticity of host '[192.168.122.130]:4444 ([192.168.122.130]:4444)'
can't be established.
ECDSA key fingerprint is SHA256:qx+10G0ETLLICvTfeMXjBOWEuc4iF+0b7K3WzuaFSZ
g.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.122.130]:4444' (ECDSA) to the list of
known hosts.
pedroamp@192.168.122.130's password:
Linux m1-pedroamp 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86
_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar  2 15:33:24 2022 from 192.168.122.1
pedroamp@m1-pedroamp:~$ sudo reboot

```

Si es la primera vez que accedemos a dicho servidor SSH, se nos dará una función HASH SHA256 como firma de que es el servidor. La aceptamos, y a continuación se almacena en ~/.ssh/known_hosts

3.3.2. Creando par de claves y conexión sin contraseña

Ahora usamos el siguiente comando del cliente de SSH para poder generar un par de claves pública/privada, que nos permita el acceso remoto, sin tener que autenticarse con contraseña.

```
ssh-keygen
```

Podemos poner una contraseña para evitar el uso inadecuado o por parte de terceros que no sean los creadores de dicho par de claves. Como medida de seguridad.

```
pedroamp@m2-pedroamp:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pedroamp/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pedroamp/.ssh/id_rsa
Your public key has been saved in /home/pedroamp/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:HDAR0efTsosK2oLr8u5GgGHxrNBLGphQaHJimuKT7fw pedroamp@m2-pedroamp
The key's randomart image is:
+---[RSA 3072]-----+
|. +o   *=          |
|B*+    o. .        |
|%=00   .o .        |
|B+..   . .+ .      |
|+O+    S  +        |
| +..          .     |
| o+ .         .     |
|o o= . . . .      |
|+B=.oE..         |
+----[SHA256]-----+
pedroamp@m2-pedroamp:~$
```

Ahora tenemos que pasar la clave pública al servidor que nos queramos conectar. Para ello el cliente tiene el siguiente comando:

Nota: En esta práctica mi servidor SSHd está escuchando en el puerto 4444.

```
ssh-copy-id -p 4444 pedroamp@192.168.122.130
```

Ahora nos indica que la clave pública se instala en el servidor, pero primero necesita autenticarse por contraseña para permitir eso.

```

pedroamp@m2-pedroamp:~$ ssh-copy-id -p 4444 pedroamp@192.168.122.130
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/pedro
amp/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
ilter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
pedroamp@192.168.122.130's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -p '4444' 'pedroamp@192.168
.122.130'"
and check to make sure that only the key(s) you wanted were added.

pedroamp@m2-pedroamp:~$ ssh -p 4444 pedroamp@192.168.122.130
Linux m1-pedroamp 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86
_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar  2 17:38:21 2022 from 192.168.122.235
pedroamp@m1-pedroamp:~$ █

```

Con esto hemos terminado la parte de autenticarse con ssh con clave pública y nos podemos autenticar sin contraseña.

Podemos verificar que efectivamente se ha instalado la clave pública en \$HOME/.ssh/authorized_keys. Verificando que se pide el usuario pedroamp@m2-pedroamp como host.

```

pedroamp@m1-pedroamp:~/.ssh$ ls authorized_keys
authorized_keys
pedroamp@m1-pedroamp:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCZRJVr/lg0GFPd0T/Lf5+6+erHT5FXCwXre7
TS0RdLukury2F6FnK6gankYp40N0hkVDFs4Ze7XV00fPrC59+/jrbYJsr5zdo/Zrk1d/9osRhA
ZMEgrqsQm3jYD7Cj0/IE5gDEZ8ssIP6ITxIq0VEtnAb9VuHgBljkc3yeyHrNKCrtS5gq+rzYaI
U7xov0pmmMtsJJhGawQ0HRbuf2Py0sVZ9/PCU6Bb1xmsclLQqxEY+tUBLHqL/pA4qzpUNl3lgW
8UeI573Lnrh8qL1mZVlerrgY6LyFIRZ6E9k4ZPZ6kuj8AkZqQ+KMb1ToAKwHy/1AUBczvum/2C
LpVuD/UCgFF+aXK1goWdGrXS5U85z1s2E4nAsG9Z3fPuWZ5ciFp8yfYTvvyeyI3eJSanDJTAVr
KQokLQ4eC2JgGbn5acu/MgzTI02a3iARr5wWPwKtuVBAEqtg+RvQWi91D/xaXgRhBo2ssovJ5Z
dwJaFc91ulyduqZajfTDc3xiYtx3VVBZk= pedroamp@m2-pedroamp
pedroamp@m1-pedroamp:~/.ssh$ █

```


4. Instalación y uso de cURL

Para instalar cURL:

```
sudo apt-get install curl
```

```
pedroamp@m2-pedroamp:~$ sudo apt-get install curl
[sudo] password for pedroamp:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 608 kB of archives.
After this operation, 1,186 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

4.1. Opciones básicas sobre http:

Se ha usado en la sección de apache2, para demostrar el correcto funcionamiento de las páginas básicas.

4.2. Opciones avanzadas de cURL

4.2.1. Cookies

```
curl -c cookies.txt http://google.es
```

Donde generamos un fichero con una cookie que indica de permisos, por la protección de datos y que tenemos que dar nuestro consentimiento en un navegador web normal.

```
pedroamp@m2-pedroamp:~$ cat cookies.txt
# Netscape HTTP Cookie File
# https://curl.se/docs/http-cookies.html
# This file was generated by libcurl! Edit at your own risk.

.google.es    TRUE    /          TRUE    1709487257    CONSENT PENDING+283
pedroamp@m2-pedroamp:~$ curl -c cookies.txt https://google.es/
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.es/">here</A>.
</BODY></HTML>
```

4.2.2. Manejo de peticiones GET y POST de HTTP

Una petición GET es como la siguiente, es la petición estándar, por defecto todo lo que usemos en curl es un GET:

```
curl https://google.es
```

Una petición POST es como la siguiente, necesita subir una serie de datos como fichero json que para el servidor signifique algo.

```
curl -X POST https://reqbin.com/echo/post/json  
-H "Content-Type: application/json"  
-d '{"productId": 123456, "quantity": 100}'
```

```
pedroamp@m2-pedroamp:~$ curl -X POST https://reqbin.com/echo/post/json -H "Content-Type: applic  
ation/json" -d '{"productId": 123456, "quantity": 100}'  
{"success": "true"}
```

4.2.3. Uso para servidores web en otros puertos

Un uso del puerto adecuado es usando los ":", donde indicamos el puerto a la derecha. Tal y como se especifica en el estándar de TCP/IP.

```
curl https://m2-pedroamp.org:8886
```

5. Bibliografía

- ¹ <https://cwiki.apache.org/confluence/display/HTTPD/Home>
- ² <https://httpd.apache.org/docs/>
- ³ <https://cwiki.apache.org/confluence/display/HTTPD/DistrosDefaultLayout>
- ⁴ <https://httpd.apache.org/docs/2.4/vhosts/examples.html>
- ⁵ <https://everything.curl.dev/http/cookies>
- ⁶ <https://reqbin.com/req/c-1n4ljxb9/curl-get-request-example>

¹Wiki de Apache2 - Para localizar ficheros de configuración

²Documentación de Apache2

³Página con el árbol de directorios de configuración de Apache2

⁴Ejemplos de VirtualHost de Apache2

⁵Uso de cookies cURL

⁶Página con peticiones GET y POST para pruebas y ejemplos