

---

# **Servidores Web de Altas Prestaciones.**

## **Práctica 1**

Introducción y Preparación de Herramientas.

Ricardo Ruiz Fernández de Alba

28/04/2023

## Índice

|  |           |
|--|-----------|
| <b>Introducción</b>  | <b>2</b>  |
| Software de Virtualización . . . . .                                 | 2         |
| Instalación de Ubuntu Server . . . . .                               | 2         |
| Tareas a realizar . . . . .  | 4         |
| <b>Tarea 1. Acceder por SSH de M1 a M2</b>                           | <b>5</b>  |
| Añadir y Configurar Adaptadores de Red . . . . .                     | 5         |
| Configuración de IP y Puertas de enlace . . . . .                    | 6         |
| Demostración de la tarea . . . . .                                   | 7         |
| Acceso sin Contraseña . . . . .                                      | 9         |
| Demostración acceso sin contraseña . . . . .                         | 11        |
| <b>Tarea 2. Acceder mediante curl de M1 a M2.</b>                    | <b>12</b> |
| Instalación LAMP . . . . .   | 12        |
| Demostración de la tarea. . . . .                                    | 13        |
| Uso avanzado de CURL:  | 15        |
| Cookies . . . . .  | 15        |
| Peticiones POST . . . . .  | 16        |
| Puertos . . . . .  | 16        |
| <b>Tarea 3. Mostrar configuraciones de red y opciones de netplan</b> | <b>16</b> |
| Configuración de la puerta de enlace (gateway): . . . . .            | 16        |
| Configuración de servidores DNS: . . . . .                           | 16        |
| Configuración de máscaras de red: . . . . .                          | 16        |
| <b>Tarea 4. Página web de ejemplo</b>                                | <b>17</b> |
| Creación del archivo en el servidor apache de M2 . . . . .           | 17        |
| Acceso mediante curl desde M1 . . . . .                              | 17        |
| Uso avanzado de Apache . . . . .                                     | 18        |
| Directorios virtuales . . . . .                                      | 18        |
| Redireccionar puertos: . . . . .                                     | 20        |
| <b>Bibliografía</b>  | <b>21</b> |

## Introducción

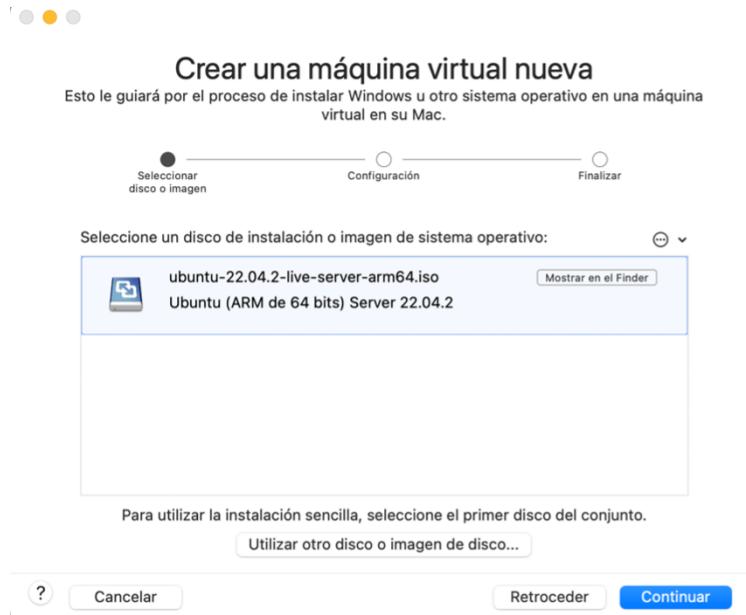
### Software de Virtualización

Utilizaremos VMWare Fusion para virtualizar varias máquinas con Ubuntu 22.04 LTS que configuraremos para ir definiendo la estructura de granja web. Descargamos Ubuntu 22.04 LTS con arquitectura ARM pues trabajamos desde MacOS Ventura con Apple Silicon M2.

La nombramos m1-ricardorui. La máquina se ha instalado con una configuración de 4GB de RAM y 20GB de disco duro. Modificamos el tamaño del disco duro a 10GB mediante

- Máquina Virtual > Disco duro NVMe > Configuración de Disco Duro (NVMe).

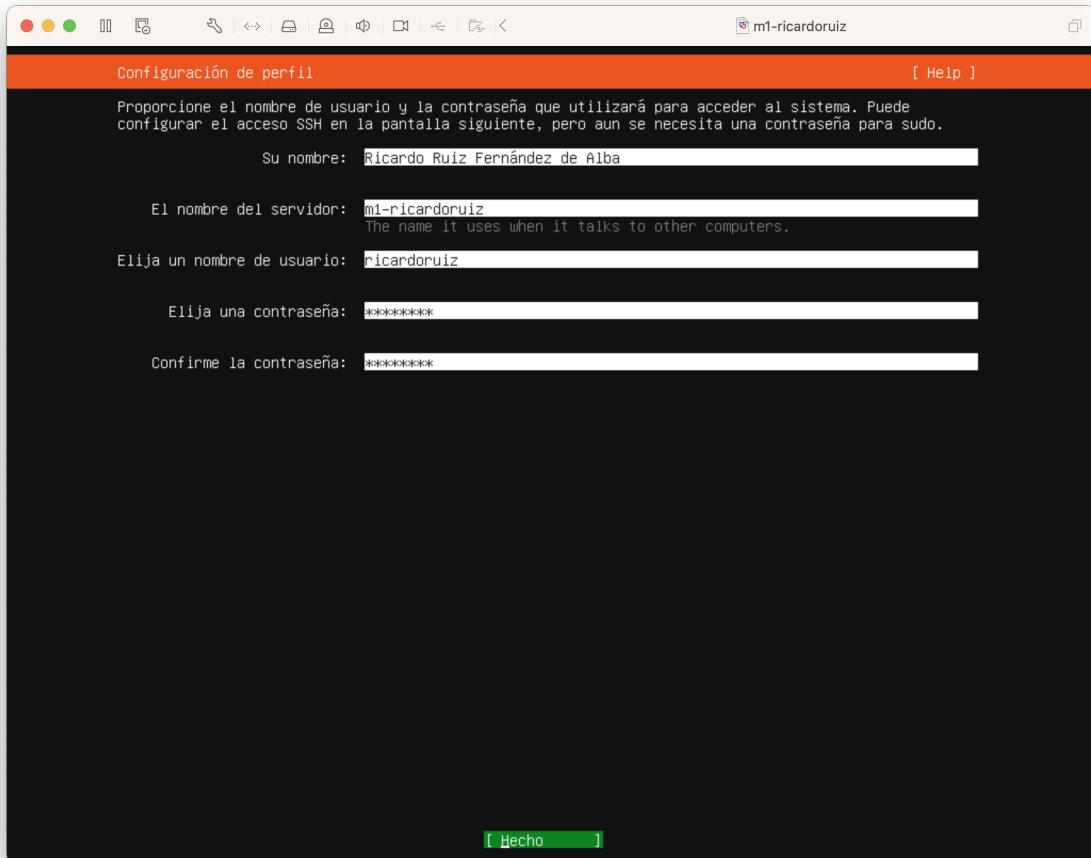
Iniciamos el programa y creamos una nueva máquina virtual con Inicio > Nuevo. Pulsamos en instalar desde disco o imagen y seleccionamos la imagen descargada.



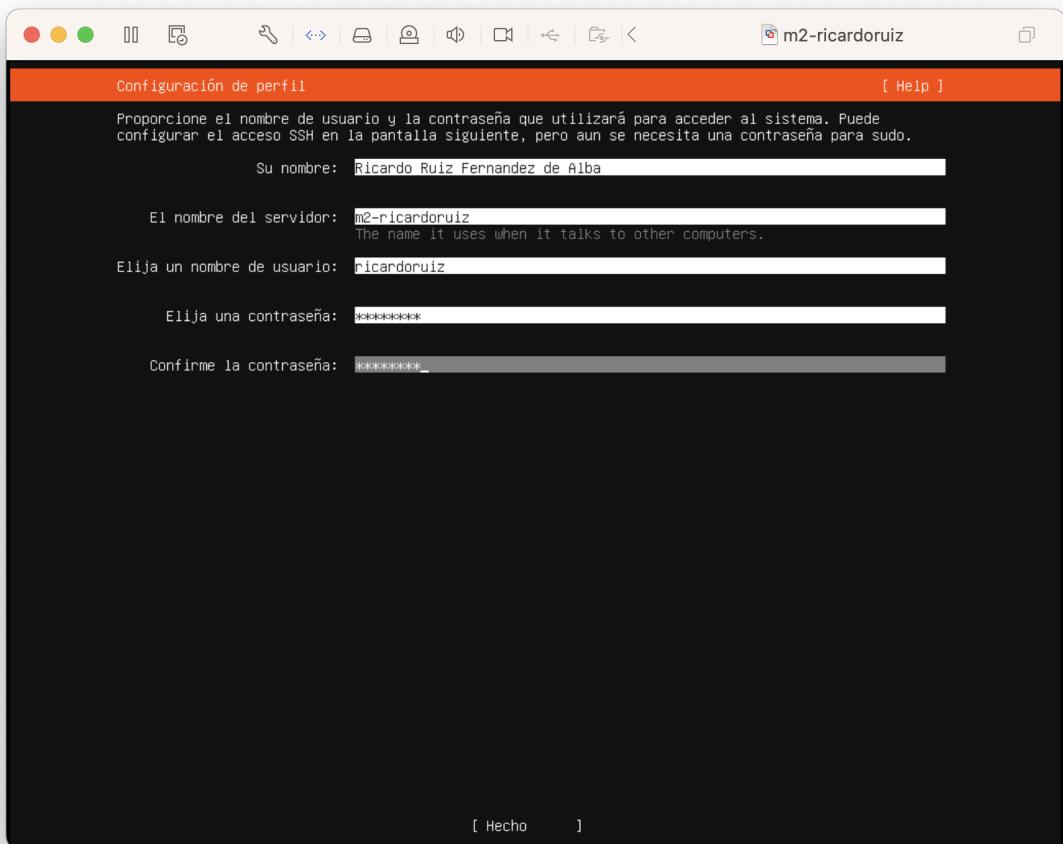
### Instalación de Ubuntu Server

1. Iniciamos la máquina virtual y procedemos a instalar Ubuntu Server.
2. Arrancamos con Install Ubuntu Server e iniciamos en español.
3. En la configuración de perfil añadimos el nombre:
  1. Ricardo Ruiz Fernández de Alba
  2. Nombre de servidor m1-ricardorui

3. Usuario: ricardoruiz
4. Contraseña: Swap1234



Pulsamos la opción de instalar OpenSSH. Repetimos la misma instalación con la máquina m2-ricardoruiz.



Sin hacer ninguna configuración, las direcciones IP de las máquinas son:

- m1-ricardoruez: 172.16.21.132
- m2-ricardoruez: 172.16.21.130

## Tareas a realizar

Necesitamos efectuar las siguientes tareas.

1. Acceder por ssh de una máquina a otra
2. Acceder mediante la herramienta curl desde una máquina a la otra
3. Mostrar configuraciones de red y opciones de netplan
4. Crear web básica (swap.html) y mostrar funcionamiento de las máquinas M1 y M2

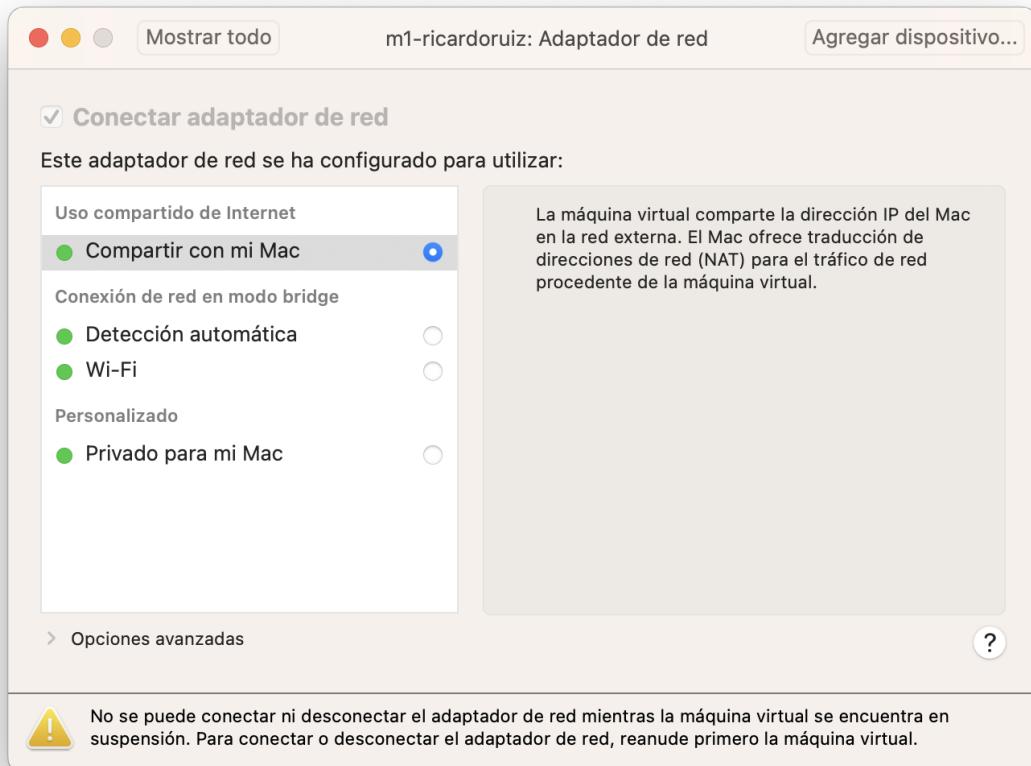
## Tarea 1. Acceder por SSH de M1 a M2

Al instalar las máquinas, ya pulsamos la opción de instalar OpenSSH en ellas.

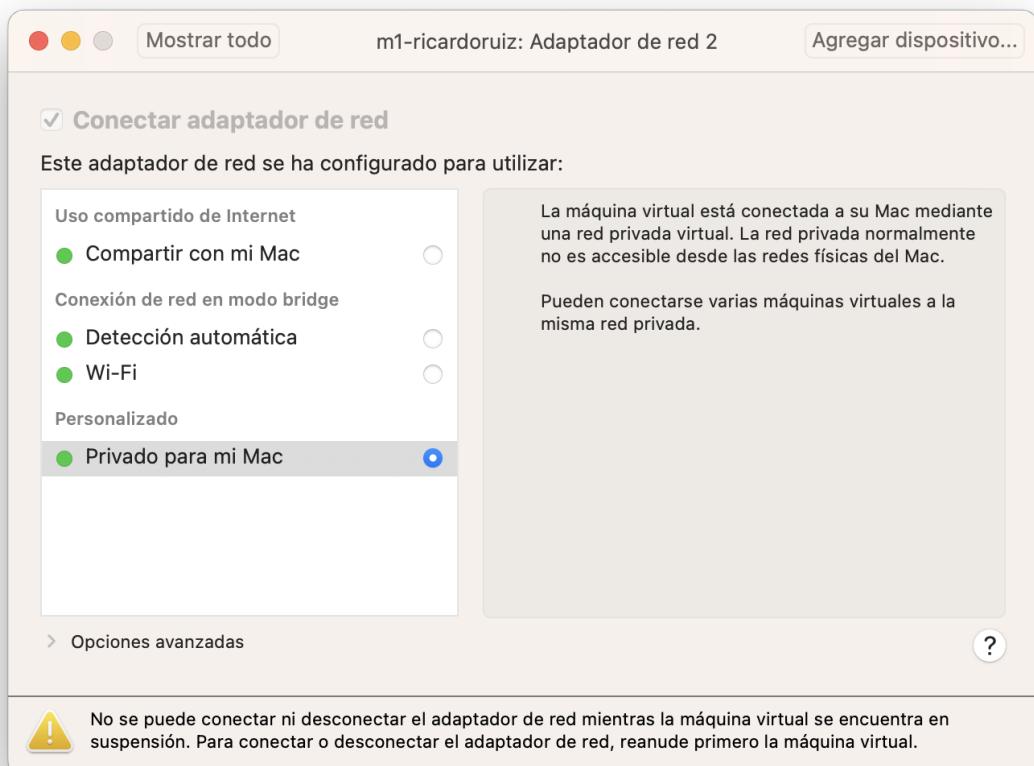
Necesitamos asegurar que ambas máquinas virtuales estén en la misma red “host-only”.

### Añadir y Configurar Adaptadores de Red

Añadimos dos adaptadores de red en VMWare Fusion, uno de tipo **Uso compartido de Internet** que corresponde a modo NAT y otro Personalizado de tipo **Privado para mi mac** que corresponde a modo host-only.



**Figura 1:** Adaptador modo NAT



**Figura 2:** Adaptador host-only

### Configuración de IP y Puertas de enlace

Abrimos el archivo de configuración de Netplan `/etc/netplan/00-installer-config.yaml` y lo editamos con la siguiente configuración:

Para **M1**

```
1  network:
2    version: 2
3    renderer: networkd
4    ethernets:
5      ens160:
6        addresses:
7          - 192.168.1.10/24
8        gateway4: 192.168.1.1
9        nameservers:
```

```

10      addresses: [8.8.8.8, 8.8.4.4]
11      ens256:
12          addresses:
13              - 192.168.2.10/24

```

Para **M2**

```

1  network:
2      version: 2
3      renderer: networkd
4      ethernets:
5          ens160:
6              addresses:
7                  - 192.168.1.20/24
8          gateway4: 192.168.1.1
9          nameservers:
10             addresses: [8.8.8.8, 8.8.4.4]
11         ens256:
12             addresses:
13                 - 192.168.2.20/24

```

Aplicamos la configuración de Netplan con

```

1 $ sudo netplan generate
2 $ sudo netplan apply

```

Y verificamos que las interfaces de red tienen la configuración correcta con `ip addr show`.

De manera que

- La interfaz `ens160` correspondiente al adaptador en **modo NAT** se configura con la dirección ip estática `192.168.1.10/24` en M1 y `192.168.1.20/24` en M2, la puerta de enlace (`192.168.1.1`) y servidores DNS.
- La interfaz `ens256` correspondiente al adaptador en **modo host-only** se configura con una dirección IP estática `192.168.2.10/24` en la máquina M1 y `192.168.2.20/24` en la máquina M2.

## Demostración de la tarea

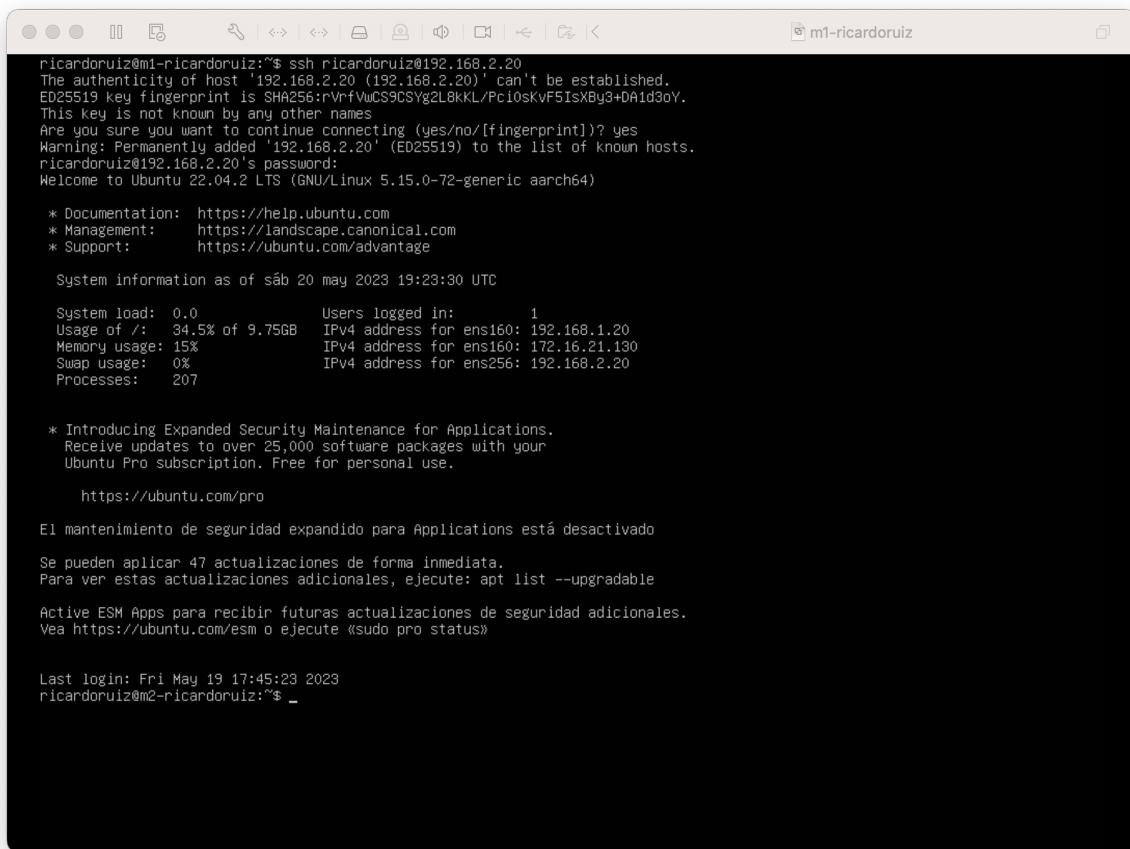
Podemos realizar entonces la conexión SSH, desde la M1 (máquina de origen) a M2 (la máquina destino)

Desde M1:

```

1 $ ssh ricardoruiz@192.168.2.20

```



```
ricardoruiz@m1-ricardoruiz:~$ ssh ricardoruiz@192.168.2.20
The authenticity of host '192.168.2.20 (192.168.2.20)' can't be established.
ED25519 key fingerprint is SHA256:rVrfVuCS9CSYg2L8KKL/PcioSkvF5IsXBy3+DA1d3oY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.20' (ED25519) to the list of known hosts.
ricardoruiz@192.168.2.20's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of sáb 20 may 2023 19:23:30 UTC

System load: 0.0          Users logged in:      1
Usage of /: 34.5% of 9.75GB  IPv4 address for ens160: 192.168.1.20
Memory usage: 15%          IPv4 address for ens160: 172.16.21.130
Swap usage:  0%          IPv4 address for ens256: 192.168.2.20
Processes:   207

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 47 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

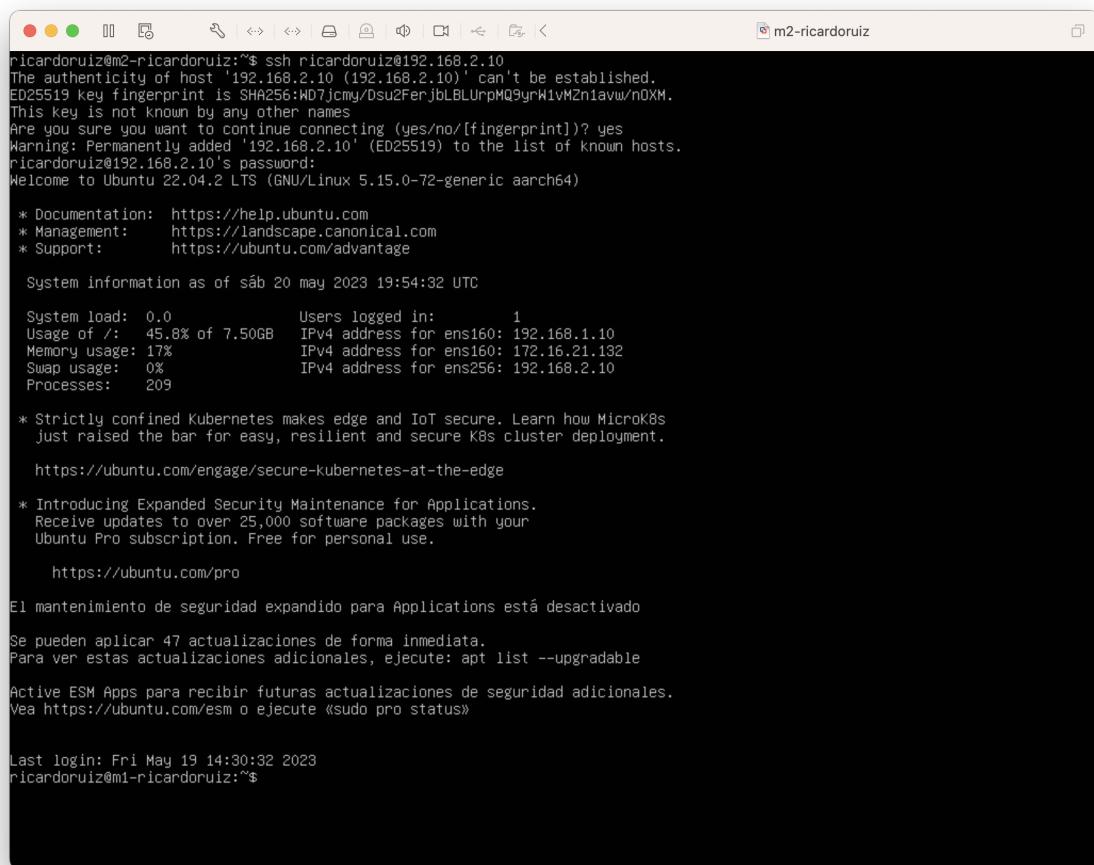
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Fri May 19 17:45:23 2023
ricardoruiz@m2-ricardoruiz:~$ _
```

**Figura 3:** Conexion ssh de M1 a M2

Análogamente, de M2 a M1:

```
1 $ ssh ricardoruiz@192.168.2.10
```



```

ricardoruiz@m2-ricardoruiz:~$ ssh ricardoruiz@m1-ricardoruiz
The authenticity of host '192.168.2.10 (192.168.2.10)' can't be established.
ED25519 key fingerprint is SHA256:W07jcmv/Dsu2FerjbLBLUrpMQ9yrH1vM2h1avw/n0XM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.10' (ED25519) to the list of known hosts.
ricardoruiz@m1-ricardoruiz:~$ password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of sáb 20 may 2023 19:54:32 UTC

 System load:  0.0          Users logged in:      1
 Usage of /:   45.8% of  7.50GB   IPv4 address for ens160: 192.168.1.10
 Memory usage: 17%
 Swap usage:  0%          IPv4 address for ens160: 172.16.21.132
 Processes:   209          IPv4 address for ens256: 192.168.2.10

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.
 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
 Receive updates to over 25,000 software packages with your
 Ubuntu Pro subscription. Free for personal use.
 https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 47 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Fri May 19 14:30:32 2023
ricardoruiz@m1-ricardoruiz:~$
```

**Figura 4:** Conexión ssh de M2 a M1

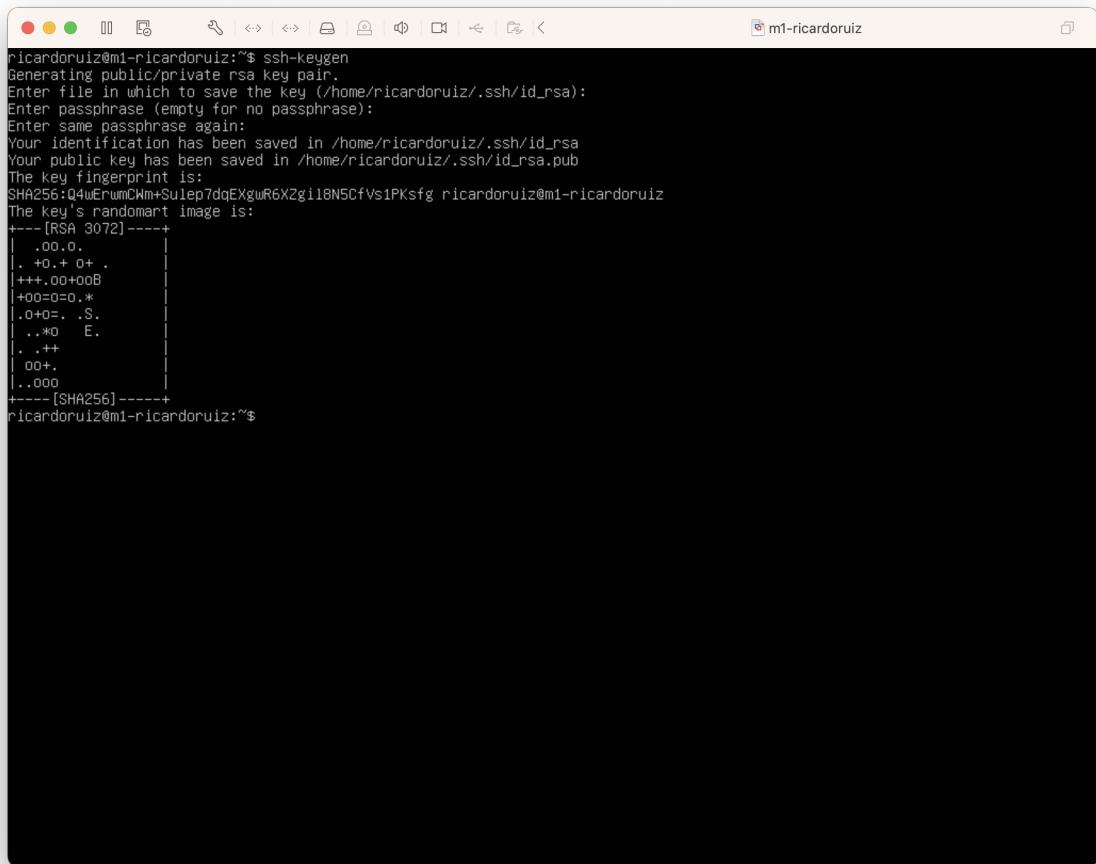
### Acceso sin Contraseña

En la anterior demostración, se nos pide la contraseña de usuario para acceder a la máquina destino. Para evitar esto, podemos configurar la autenticación mediante clave pública y privada.

### Generar claves pública y privada

1. En M1, generamos un par de claves pública y privada utilizando el comando `ssh-keygen`.

```
1 ricardoruiz@m1-ricardoruiz $ ssh-keygen
```

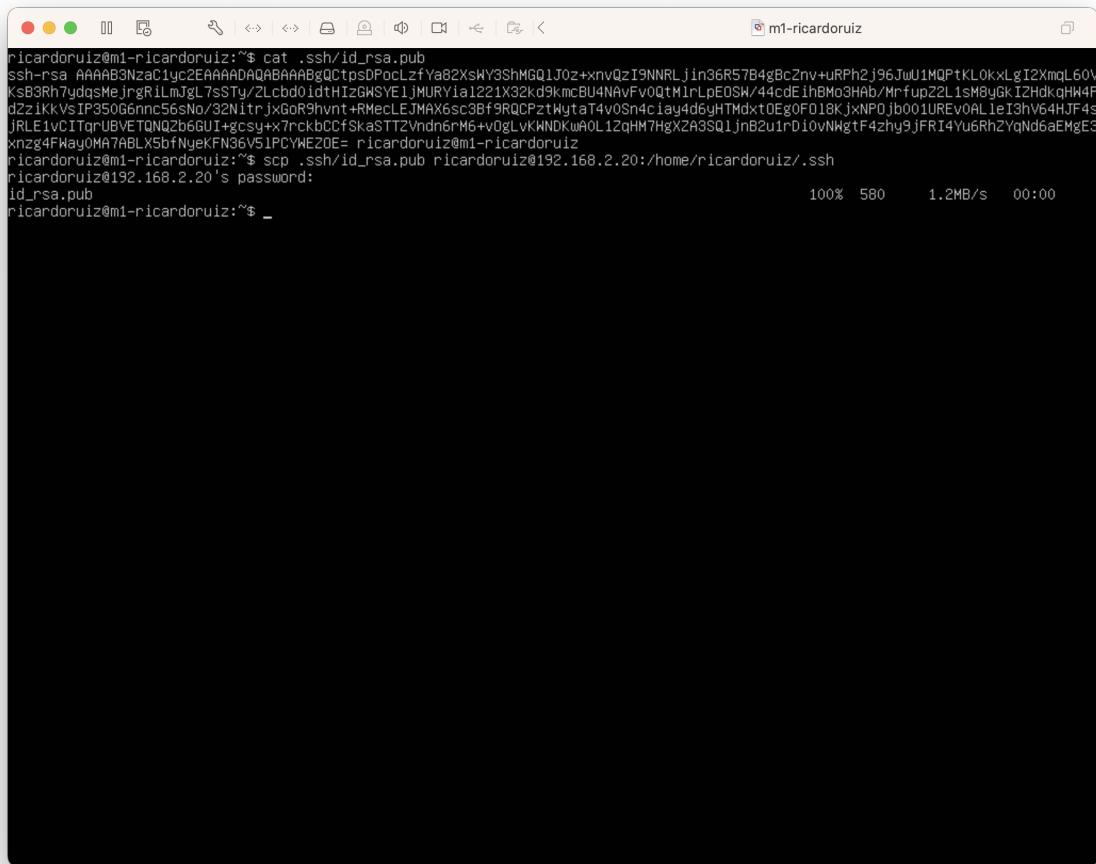


```
ricardorui@m1-ricardorui:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ricardorui/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ricardorui/.ssh/id_rsa
Your public key has been saved in /home/ricardorui/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Q4uErnunCKm+nSulep7dqExgwR6XZgi18NS5CfVs1PKsfg ricardorui@m1-ricardorui
The key's randomart image is:
+---[RSA 3072]----+
| ..0.0.
| .+0.+ 0+ .
| ++++.00+00B
| +00=0=0.*
| .0+=. .S.
| ..*o E.
| ..++
| 00+.
| ..000
+---[SHA256]----+
ricardorui@m1-ricardorui:~$
```

**Figura 5:** Clave Pública y privada

2. Copiamos la clave pública generada (generalmente se encuentra en el archivo `~/.ssh/id_rsa.pub` o `~/.ssh/id_dsa.pub`) al archivo `authorized_keys` en la máquina remota. Para ello, podemos utilizar el comando `scp`

```
1 ricardorui@m1-ricardorui $ scp .ssh/id_rsa.pub ricardorui@192
.168.2.20:/home/ricardo/.ssh
```



```
ricardorui@m1-ricardorui:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABAAAABgCtpsdPocLzfyYa82XsWY3ShMGQ1J0z+xnvQzI9NNRLjin36R57B4gBc2nv+uRPh2J96JuU1MQPtKL0kxLgI2XmqL60V
KsB3R7yqdsMeJrgILmJg17sSTy/ZLcb0idtH1zGKSYE1jMURY1a1221X32kd9kmcBU4NavyfvoQTM1rLpE0SW/44cdE1hBMo3HAb/Mrfup22L1sMByGkI2HdkqHM4F
dZz1kVsIP350G6nncc56sNo/32NitrJxGo9hvnt+RMecLEJMAX6sc3Bf9RQCpztWutaT4v0Sn4ciay4d6yHTMdx1DEg0F018KJxNP0jb001UREvoAL1eI3hV64HJF4s
JRL1v0ITqrUBVETQNQ2b6U1+gcsy+X7rckbCfksaSTT2Vndn6rM6+v0gLvKNDKwAO112qHM7hgX2aS01JnB2u1rD10vNWgtF4zhg9JFR14Yu6Rh2Yqld6aEMgE3
xnzg4FMay0MA7aBLX5bfNyekFN36V51PCyME20E= ricardorui@m1-ricardorui
ricardorui@m1-ricardorui:~$ scp .ssh/id_rsa.pub ricardorui@192.168.2.20:/home/ricardorui/.ssh
ricardorui@m1-ricardorui:~$ _

id_rsa.pub
ricardorui@m1-ricardorui:~$ _
```

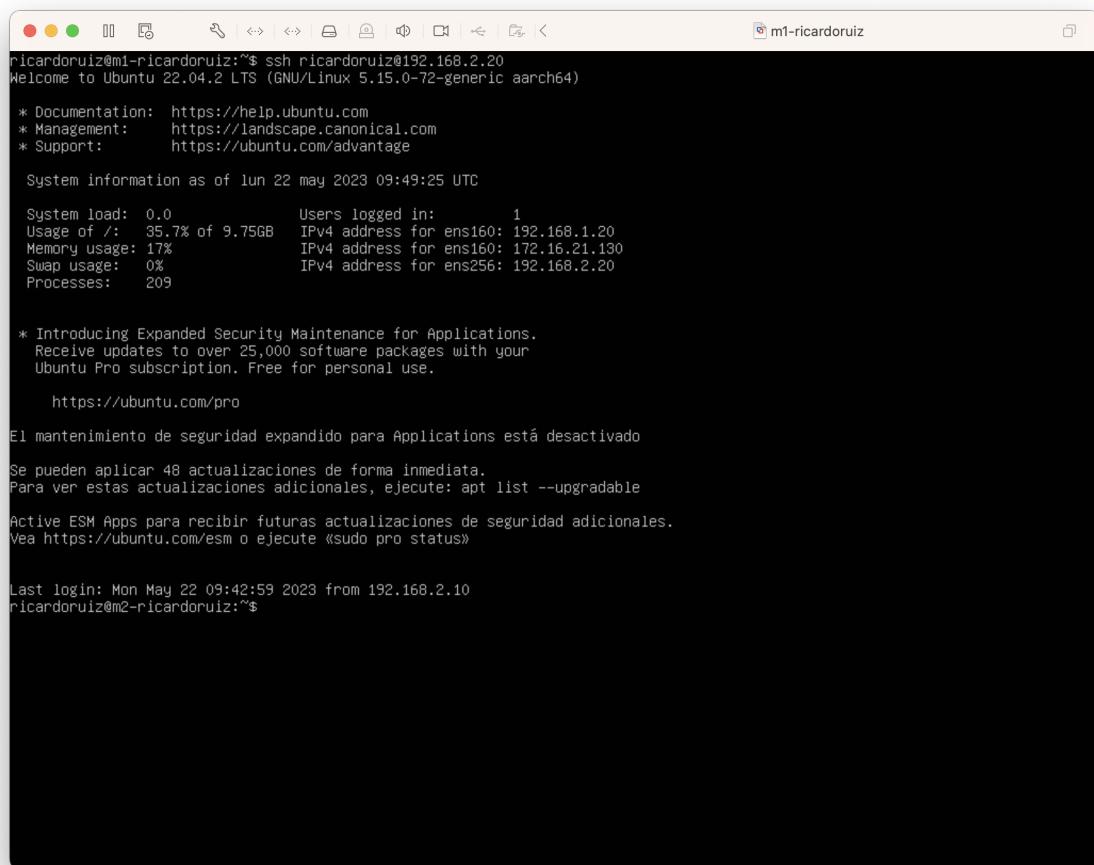
**Figura 6:** Enviar clave publica

Que debe tener permisos de lectura y escritura:

```
1 ricardorui@m2-ricardorui $ mv .ssh/id_rsa.pub .ssh/authorized_keys
2 ricardorui@m2-ricardorui $ chmod 600 .ssh/authorized_keys
```

### Demostración acceso sin contraseña

En efecto, ahora ya no es necesario incluir la contraseña cuando realizamos la conexión:



```
ricardoruiz@m1-ricardoruiz:~$ ssh ricardoruiz@192.168.2.20
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of lun 22 may 2023 09:49:25 UTC

 System load: 0.0          Users logged in:      1
 Usage of /: 35.7% of 9.75GB  IPv4 address for ens160: 192.168.1.20
 Memory usage: 17%          IPv4 address for ens160: 172.16.21.130
 Swap usage:  0%          IPv4 address for ens256: 192.168.2.20
 Processes:   209

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 48 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Mon May 22 09:42:59 2023 from 192.168.2.10
ricardoruiz@m2-ricardoruiz:~$
```

**Figura 7:** Conexion ssh de M1 a M2 sin contraseña

## Tarea 2. Acceder mediante curl de M1 a M2.

Necesitamos realizar una instalación de Apache+MySQL (LAMP) para poder acceder mediante [curl](#).

### Instalación LAMP

```
1 $ sudo apt install apache2 mysql-server mysql-client
```

Comprobamos la versión:

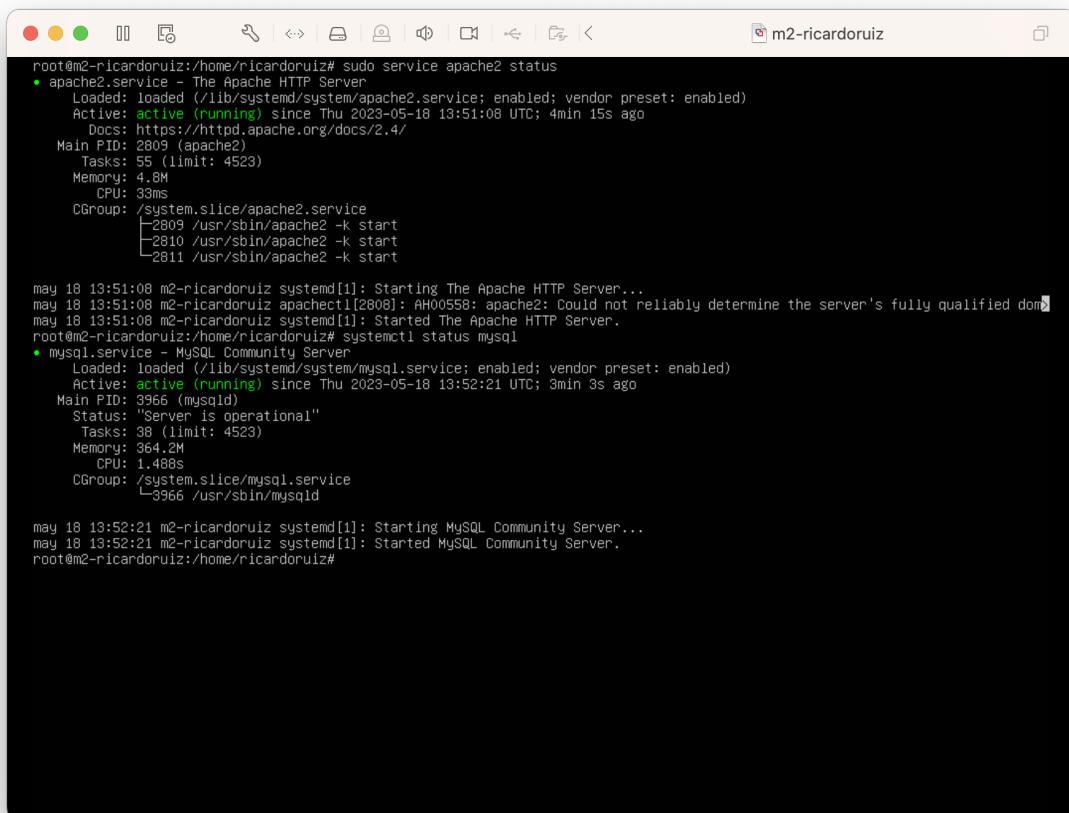
```
1 $ apache2 -v
2   Server version: Apache/2.4.52 (Ubuntu)
```

```
3 Server built: 2023-03-01T22:43:55
```

Y lo iniciamos mediante

```
1 $ sudo systemctl enable apache2
2 $ sudo systemctl start apache2
```

Y comprobamos que los dos servicios están activos:



```
root@m2-ricardoruez:/home/ricardoruez# sudo service apache2 status
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-05-18 13:51:08 UTC; 4min 15s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2809 (apache2)
      Tasks: 55 (limit: 4523)
     Memory: 4.8M
        CPU: 39ms
       CGroup: /system.slice/apache2.service
           └─2809 /usr/sbin/apache2 -k start
             ├─2810 /usr/sbin/apache2 -k start
             ├─2811 /usr/sbin/apache2 -k start
             └─2812 /usr/sbin/apache2 -k start

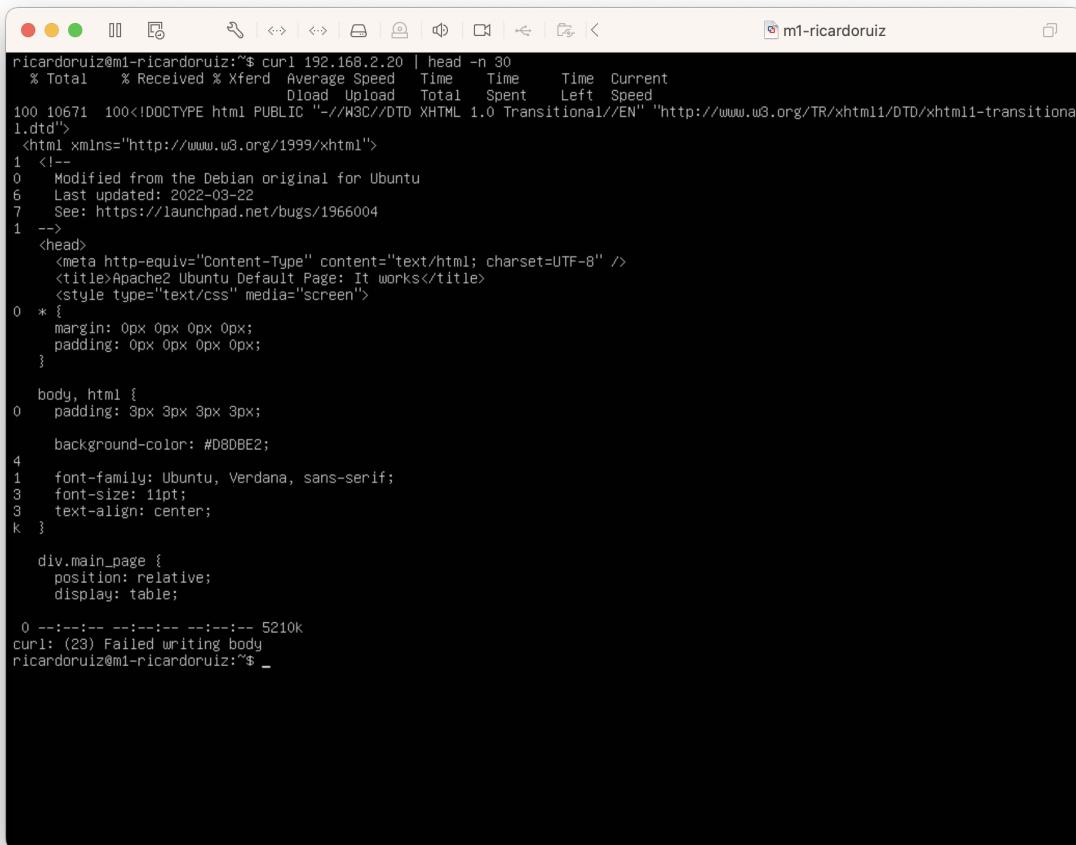
may 18 13:51:08 m2-ricardoruez systemd[1]: Starting The Apache HTTP Server...
may 18 13:51:08 m2-ricardoruez apache2[2809]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using m2-ricardoruez. You probably need to update your host file.
may 18 13:51:08 m2-ricardoruez systemd[1]: Started The Apache HTTP Server.
root@m2-ricardoruez:/home/ricardoruez# systemctl status mysql
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-05-18 13:52:21 UTC; 3min 3s ago
    Main PID: 3966 (mysqld)
   Status: "Server is operational"
      Tasks: 38 (limit: 4523)
     Memory: 364.2M
        CPU: 1.498s
       CGroup: /system.slice/mysql.service
           └─3966 /usr/sbin/mysqld

may 18 13:52:21 m2-ricardoruez systemd[1]: Starting MySQL Community Server...
may 18 13:52:21 m2-ricardoruez systemd[1]: Started MySQL Community Server.
root@m2-ricardoruez:/home/ricardoruez#
```

## Demostración de la tarea.

Tras instalar curl con `sudo apt-get install curl`, realizamos

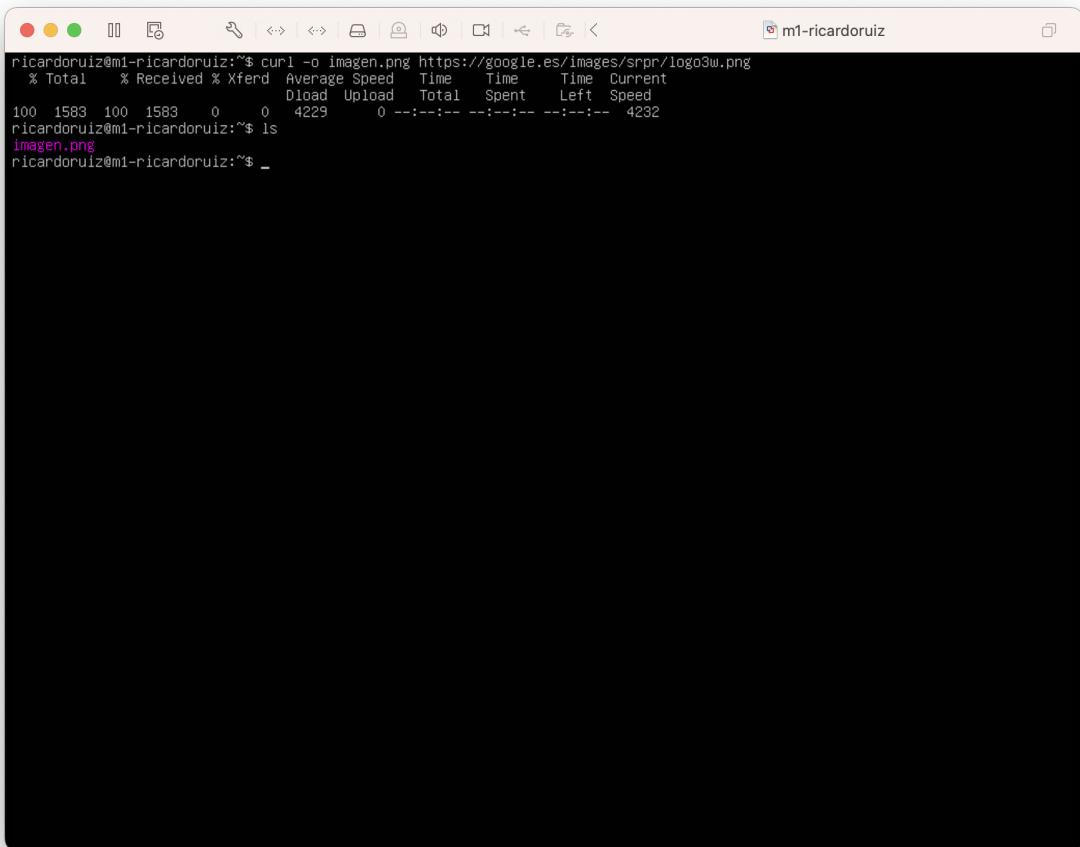
```
1 $ curl 192.168.2.20/index.html
```



A screenshot of a terminal window titled 'm1-ricardoruz'. The window shows the output of a curl command. The output is a CSS file (Apache2 Ubuntu Default Page: It works) with line numbers and a timestamp. The terminal window has a dark background and light text. The title bar shows the terminal title and the host name.

```
ricardoruz@m1-ricardoruz:~$ curl 192.168.2.20 | head -n 30
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
          0          0          0      0  0:00:00  0:00:00  0:00:00
  100 10671  100<!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">
1 <!--
0   Modified from the Debian original for Ubuntu
1   Last updated: 2022-03-22
2   See: https://launchpad.net/bugs/1966004
3 -->
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6     <title>Apache2 Ubuntu Default Page: It works</title>
7     <style type="text/css" media="screen">
8       *
9       {
10         margin: 0px 0px 0px 0px;
11         padding: 0px 0px 0px 0px;
12       }
13
14   body, html {
15     padding: 3px 3px 3px 3px;
16     background-color: #D8DBE2;
17   }
18   font-family: Ubuntu, Verdana, sans-serif;
19   font-size: 11pt;
20   text-align: center;
21 }
22
23   div.main_page {
24     position: relative;
25     display: table;
26
27 ---:--- ---:--- ---:--- 5210k
28 curl: (23) Failed writing body
ricardoruz@m1-ricardoruz:~$ _
```

```
1 curl -o imagen.png https://www.google.es/images/srpr/logo3w.png
```



```
ricardoruiz@m1-ricardoruiz:~$ curl -o imagen.png https://google.es/images/srpr/logo3w.png
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
     0     0     0      0      0      0      0      0 --:--:--:--:--:--:--:-- 4232
  100 1583  100 1583    0     0  4229      0 --:--:--:--:--:--:--:-- 4232
ricardoruiz@m1-ricardoruiz:~$ ls
imagen.png
ricardoruiz@m1-ricardoruiz:~$ _
```

## Uso avanzado de CURL:

Curl también proporciona opciones avanzadas para trabajar con HTTP, como cookies, peticiones GET/POST personalizadas y especificación de puertos. A continuación, se muestran algunas de las opciones más comunes:

### Cookies

Usando la opción `-b` cookies

```
1 $ curl -b "cookie1=value1; cookie2=value2" http://example.com
```

### Peticiones POST

Mediante `-X` o `--request` y `d`

```
1 $ curl -X POST -d "username=admin&password=12345" http://example.com
```

### Puertos

Se pueden especificar con `-po` `--port`:

```
1 $ curl -p 8080 http://example.com
```

## Tarea 3. Mostrar configuraciones de red y opciones de netplan

En efecto, ya configuramos Netplan durante la tarea 1 para disponer de IPs fáciles de recordar.

Netplan ofrece varias opciones avanzadas para configurar la puerta de enlace (gateway), servidores DNS y máscaras de red.

### Configuración de la puerta de enlace (gateway):

El apartado

```
1 - to:  
2   via: <IP puerta de enlace>
```

de la configuración yaml, permite modificar la puerta de enlace (gateway)

### Configuración de servidores DNS:

El apartado

```
1 nameservers:  
2   addresses: [<DNS1>, <DNS2>]
```

de la configuración yaml, permite modificar los servidores DNS.

### Configuración de máscaras de red:

El apartado

```
1 [...]  
2 addresses:  
3   - <IP>/<prefijo de red>
```

permite modificar la máscara de red. El prefijo de red indica el número de bits y define el tamaño de la red. En nuestro caso, hemos utilizado una máscara de red de 24 bits.

## Tarea 4. Página web de ejemplo

### Creación del archivo en el servidor apache de M2

Creamos el archivo `swap.html` en M2 en `/var/www/html`.

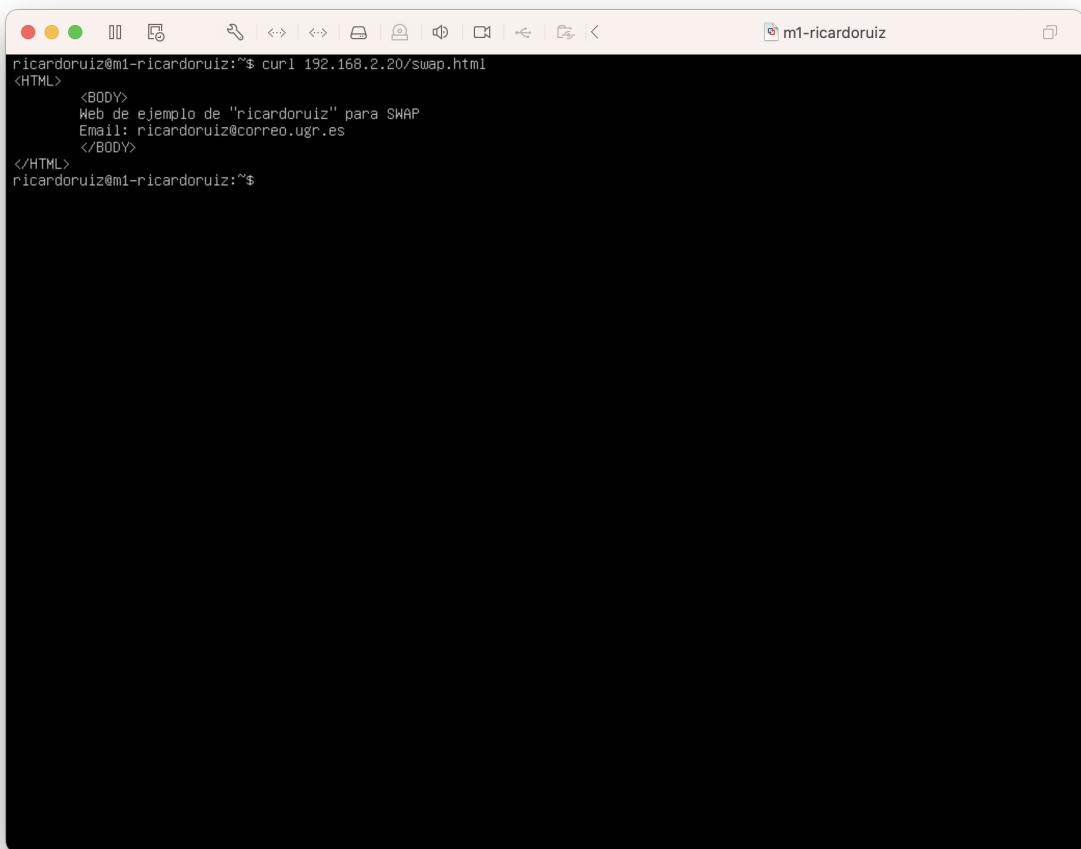
#### `swap.html`

```
1 <HTML>  
2   <BODY>  
3     Web de ejemplo de "ricardoruiz" para SWAP  
4     Email: ricardoruiz@correo.ugr.es  
5   </BODY>  
6 </HTML>
```

### Acceso mediante curl desde M1

```
1 $ curl 192.168.2.20/swap.html
```

obteniendo



```
ricardoruez@m1-ricardoruez:~$ curl 192.168.2.20/swap.html
<HTML>
  <BODY>
    Web de ejemplo de "ricardoruez" para SWAP
    Email: ricardoruez@correo.ugr.es
  </BODY>
</HTML>
ricardoruez@m1-ricardoruez:~$
```

## Uso avanzado de Apache

### Directorios virtuales

Para crear directorios virtuales en Apache, creamos archivos `.conf` en `/etc/apache2/sites-available` y realizamos un enlace simbolico a estos archivos en `/etc/apache2/sites-enabled`.

Crearemos dos directorios virtuales, uno en el puerto 80 que apuntará al directorio `/var/www/html` y otro en el puerto 8000 que apuntará al directorio `/var/www/virtual`. Este tendrá un archivo `index.html` que indique que accedemos al directorio virtual.

`index.html`

```
1 <HTML>
2   <BODY>
```

```

3     Directorio virtual de "ricardoruiz" para SWAP
4     Email: ricardoruiz@correo.ugr.es
5     </BODY>
6 </HTML>

```

### default-html.conf

```

1 <VirtualHost *:80>
2   ServerName 192.168.2.20
3   DocumentRoot /var/www/html
4
5   <Directory /var/www/html>
6     Options Indexes FollowSymLinks
7     AllowOverride All
8     Require all granted
9   </Directory>
10 </VirtualHost>

```

### virtual.conf

```

1 <VirtualHost *:8000>
2   ServerName 192.168.2.20
3   DocumentRoot /var/www/virtual
4
5   <Directory /var/www/virtual>
6     Options Indexes FollowSymLinks
7     AllowOverride All
8     Require all granted
9   </Directory>
10 </VirtualHost>

```

Enlazamos los archivos .conf en /etc/apache2/sites-enabled:

```

1 $ sudo ln -s /etc/apache2/sites-available/default-html.conf /etc/
      apache2/sites-enabled/default-html.conf
2 $ sudo ln -s /etc/apache2/sites-available/virtual.conf /etc/apache2/
      sites-enabled/virtual.conf

```

Necesitamos escuchar el puerto 8000, por lo que modificamos el archivo /etc/apache2/ports.conf, añadiendo `Listen 8000`

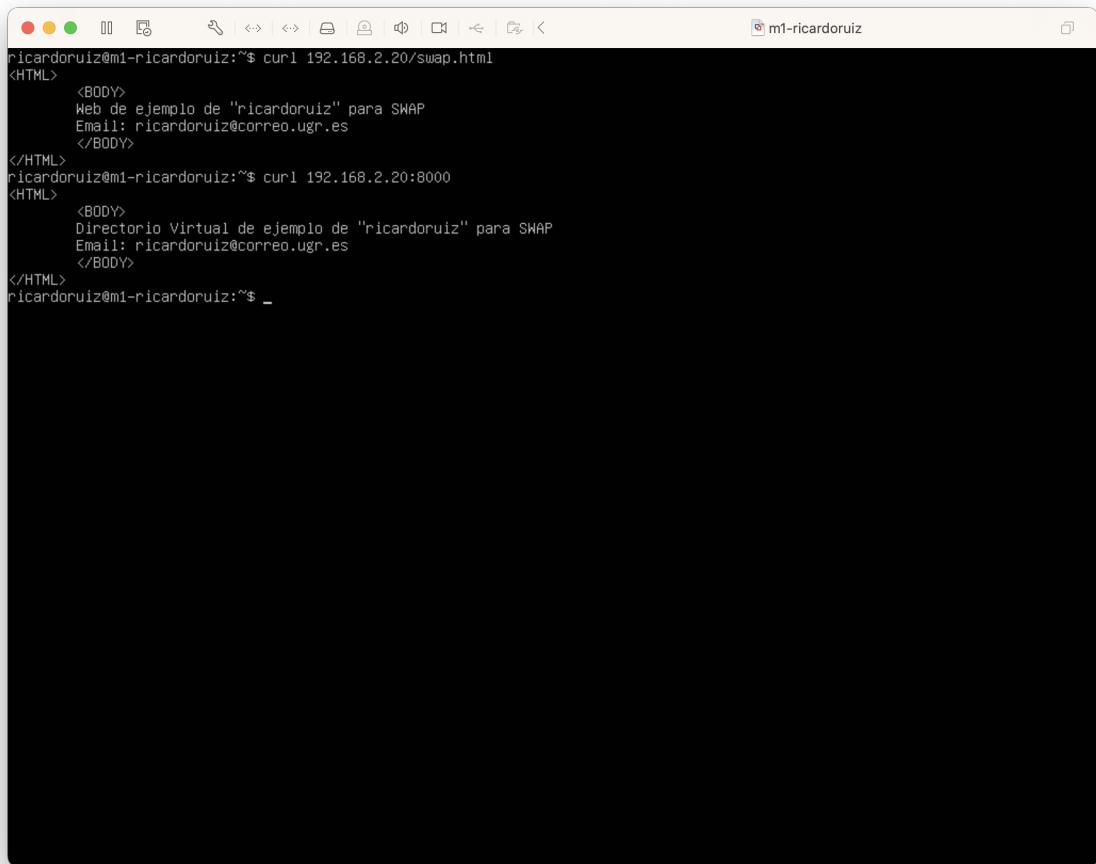
Y reiniciamos el servidor de Apache:

```

1 $ sudo systemctl restart apache2

```

Podemos comprobar accediendo con curl:

A screenshot of a macOS terminal window titled 'm1-ricardoruez'. The window shows the command 'curl 192.168.2.20/swap.html' being run, followed by the HTML content of the page. The content includes a body with the text 'Web de ejemplo de "ricardoruez" para SWAP' and 'Email: ricardoruez@correo.ugr.es'. The command 'curl 192.168.2.20:8000' is also shown, with its output being the same HTML content. The terminal prompt 'ricardoruez@m1-ricardoruez:~\$' is at the bottom.

```
ricardoruez@m1-ricardoruez:~$ curl 192.168.2.20/swap.html
<HTML>
<BODY>
Web de ejemplo de "ricardoruez" para SWAP
Email: ricardoruez@correo.ugr.es
</BODY>
</HTML>
ricardoruez@m1-ricardoruez:~$ curl 192.168.2.20:8000
<HTML>
<BODY>
Directorio Virtual de ejemplo de "ricardoruez" para SWAP
Email: ricardoruez@correo.ugr.es
</BODY>
</HTML>
ricardoruez@m1-ricardoruez:~$ _
```

### Redireccionar puertos:

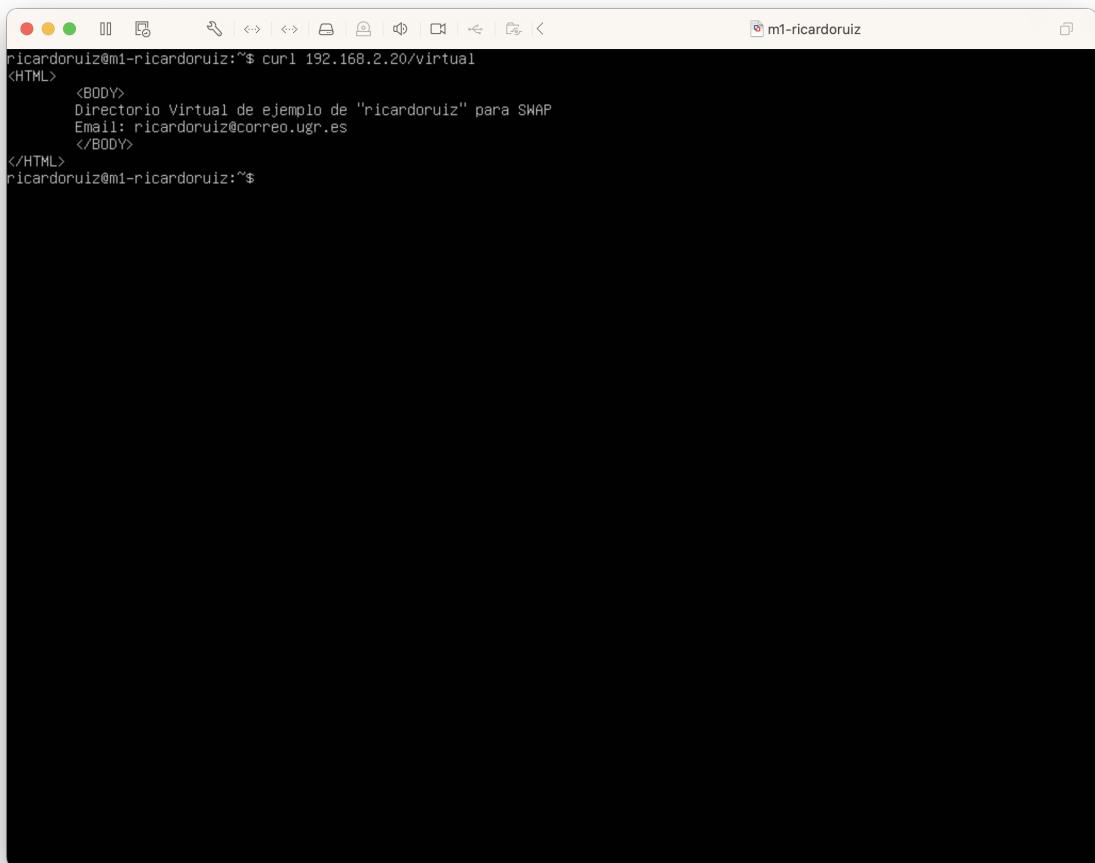
Para redireccionar puertos en Apache, se utilizan las directivas `ProxyPass` y `ProxyPassReverse`. Haremos una redirección del puerto 80 al 8000. Para ello añadimos el apartado `VirtualHost` en `default-ssl.conf` lo siguiente:

```
1 [...]
2 ProxyPass /virtual http://192.168.2.20:8000/
3 ProxyPassReverse /virtual http://192.168.2.20:8000/
4 </VirtualHost>
```

Activamo el módulo de proxy de Apache y reiniciamos el servidor:

```
1 $ sudo a2enmod proxy
2 $ sudo a2enmod proxy_http
3 $ sudo systemctl restart apache2
```

De manera que podemos acceder a nuestro directorio virtual del puerto 8000 desde el puerto 80:



```
ricardoruiz@m1-ricardoruiz:~$ curl 192.168.2.20/virtual
<HTML>
  <BODY>
    Directorio Virtual de ejemplo de "ricardoruiz" para SWAP
    Email: ricardoruiz@correo.ugr.es
  </BODY>
</HTML>
ricardoruiz@m1-ricardoruiz:~$
```

**Figura 8:** Redireccionamiento de puertos

## Bibliografía

- **VMware Fusion.** (s.f.). Recuperado de <https://www.vmware.com/products/fusion.html>
- **Ubuntu.** (s.f.). Ubuntu Server Documentation. Recuperado de <https://ubuntu.com/server/docs>
- **Netplan.** (s.f.). Recuperado de <https://netplan.io/>
  - Ejemplos de configuración: Netplan. (s.f.). Recuperado de <https://netplan.io/examples>
- **Apache.** (s.f.). Apache HTTP Server Documentation. Recuperado de <https://httpd.apache.org/docs/>

- Guía de configuración de Apache: Apache. (s.f.). Apache HTTP Server Configuration.
- **cURL.** (s.f.). Recuperado de <https://curl.se/docs/>
  - Comandos y opciones de cURL: cURL. (s.f.). cURL Manual. Recuperado de <https://curl.se/docs/manpage.html>