
Preparación de las herramientas de SWAP

Andrés Millán Muñoz (amilmun@correo.ugr.es)

March 12, 2022

Contents

1	Objetivos	1
2	Arrancando la máquina virtual	2
3	Programas básicos	9
3.1	LAMP	9
3.2	cURL	9
3.2.1	Parámetros de cURL	10
3.3	Apache	11
3.3.1	Puertos de Apache	12
3.3.2	Hosts virtuales	13
3.3.3	Redirección de puertos	15
3.4	SSH	15
3.4.1	Conexión sin contraseña	16
4	Configurando la interfaz de red	18
5	Uso de SSH	22

1 Objetivos

En esta práctica instalaremos y configuraremos dos máquinas virtuales de Ubuntu Server mediante VMWare. En particular, pondremos a punto una instalación de Apache, PHP, MySQL, SSH; así como la interfaz de red necesaria para comunicar ambas máquinas con el host y entre ellas mismas.

Para ver que todo funciona correctamente, usaremos la herramienta `curl` con el fin de solicitar una pequeña página web específica de cada VM. De esa forma, comprobaremos que Apache está propiamente configurado.

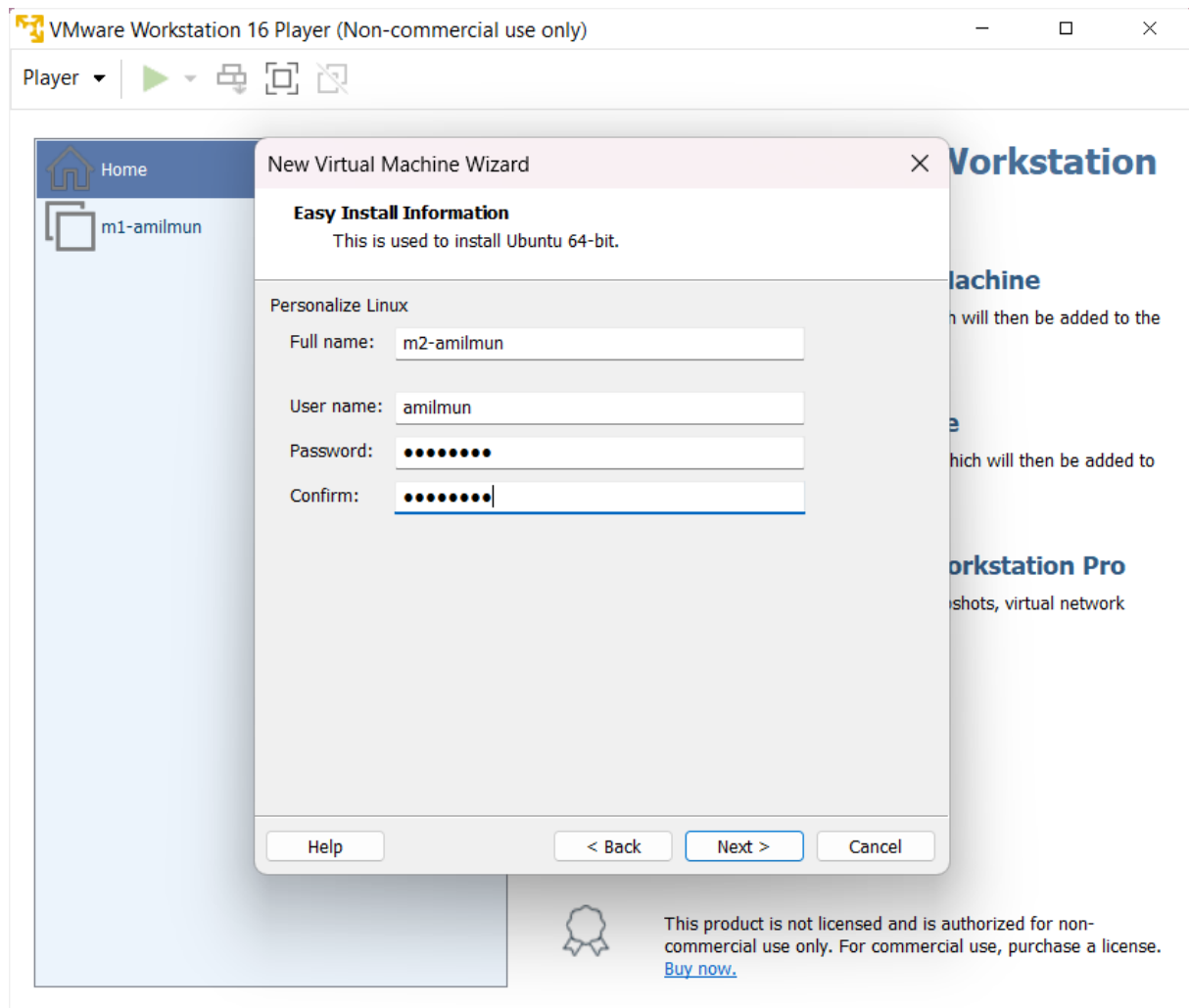
Esta documentación se encuentra disponible en [Github](#)

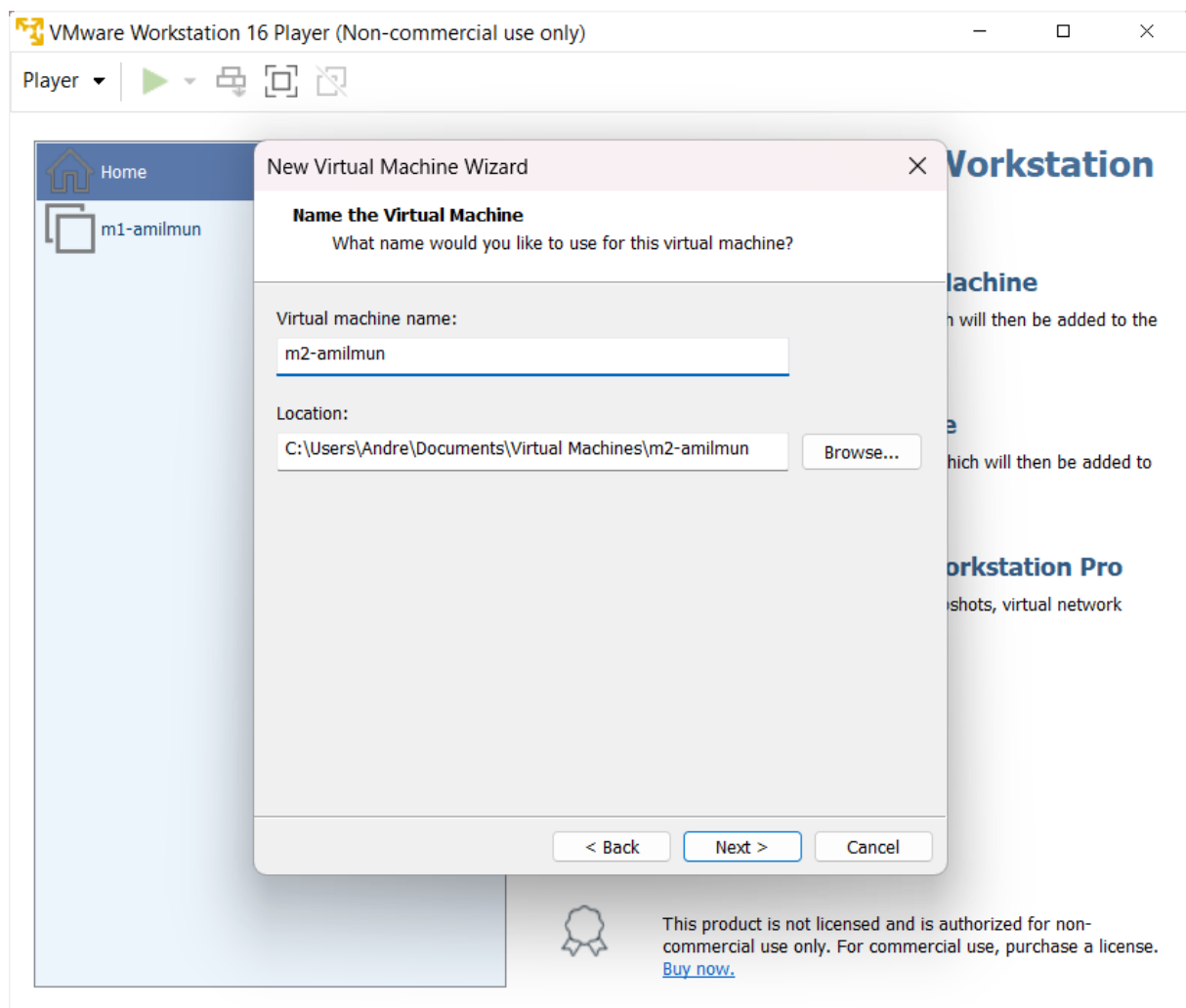
2 Arrancando la máquina virtual

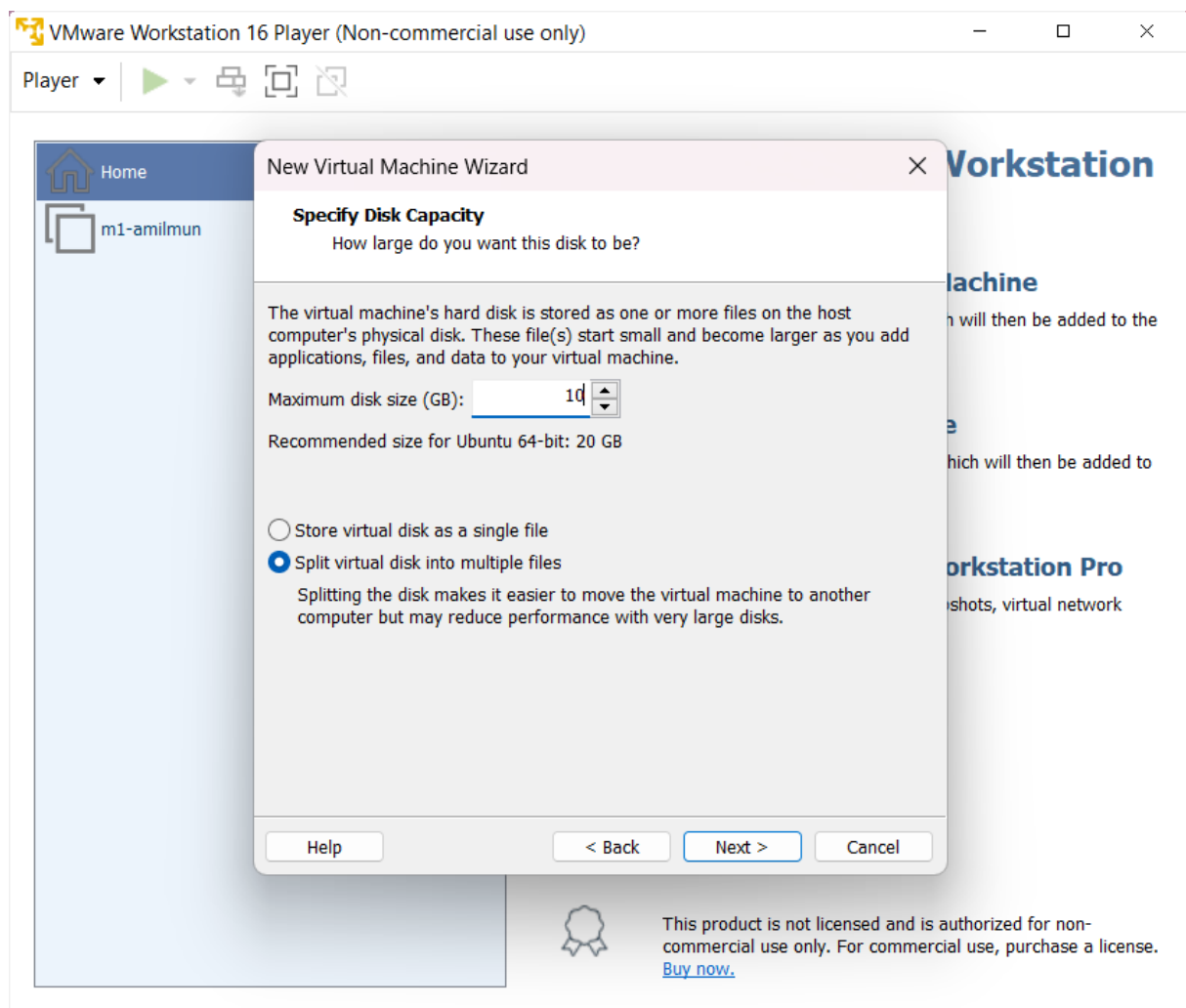
Antes de comenzar, debemos instalar ambos sistemas en **VMWare**. Se intentó hacerlo en Virtual-Box, pero un `Kernel panic - not syncing: Attempted to kill the idle task` no me permitía iniciarlo. ¿Quizás sea debido a que estoy en Windows 11 insiders, y no se lleva bien con VirtualBox? Como alternativa probé con Hyper-V, pero los 8GB de RAM de mi portátil no permitía iniciar la máquina.

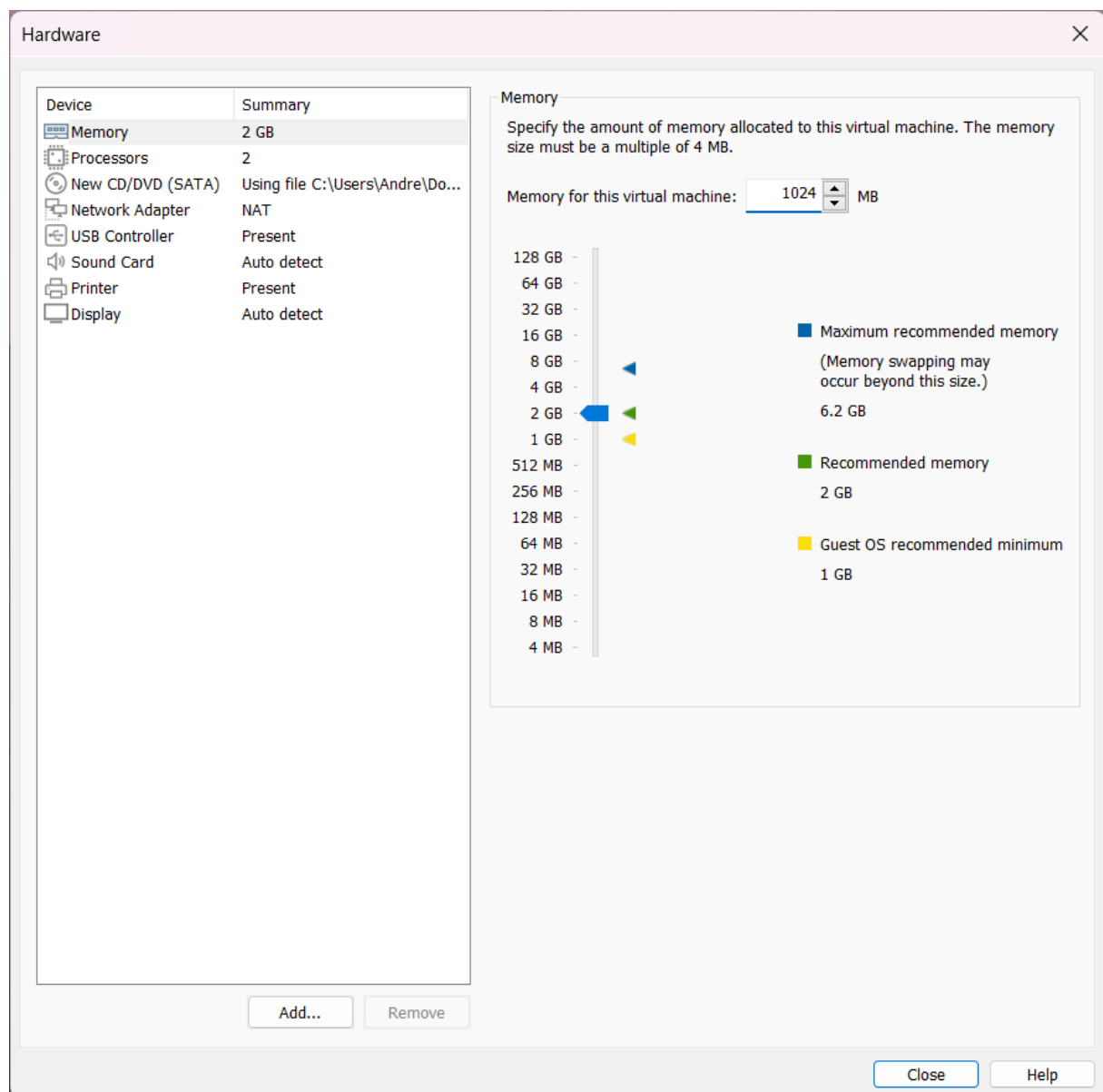
Empecemos con la creación de las máquinas. Como el proceso es análogo, mostraremos únicamente fotos de la segunda máquina, la correspondiente a `m2-amilmun`.

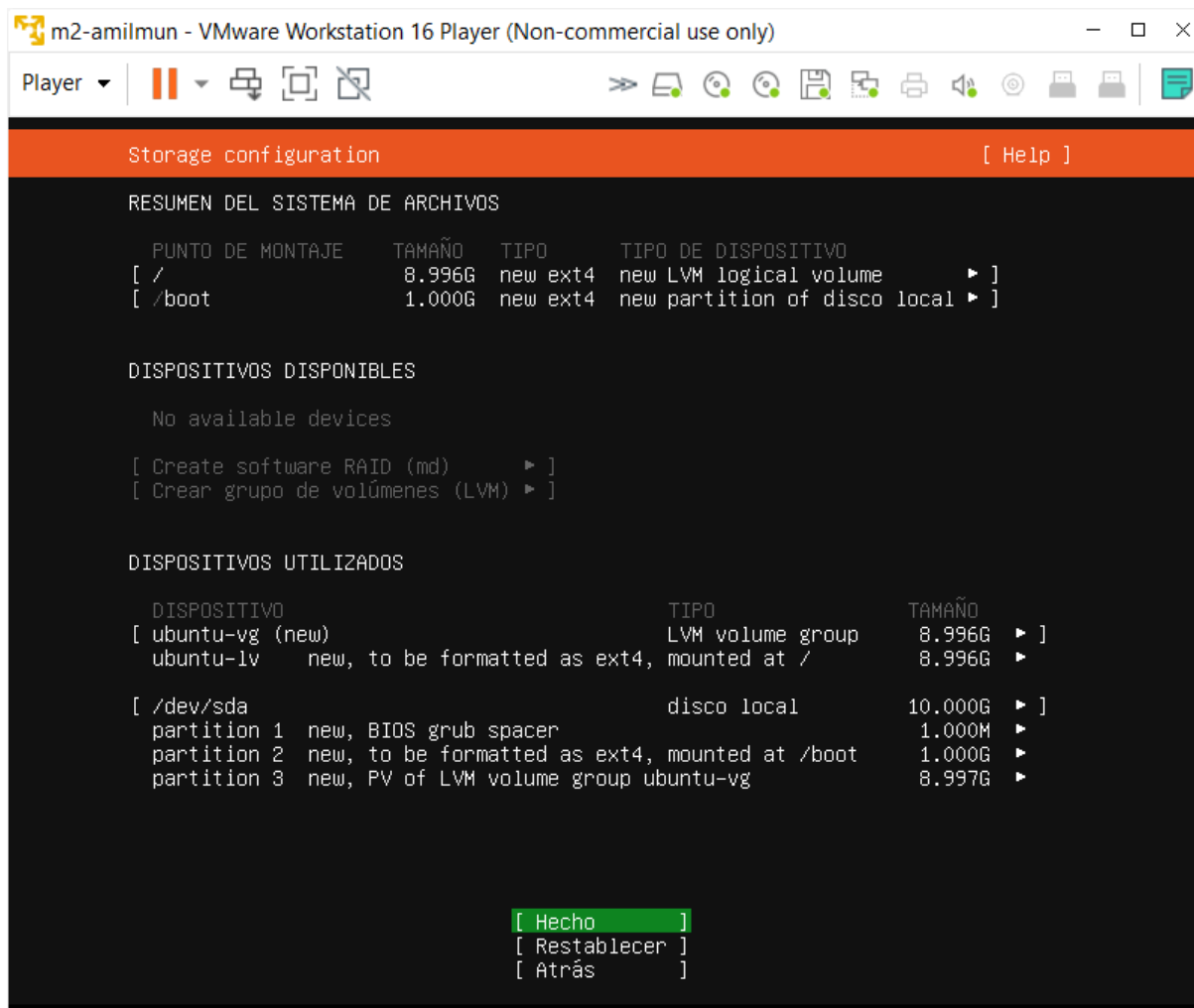
El usuario será `amilmun`, y la contraseña será `Swap1234`, como se indica en el guion. Aparte de la distribución de teclado, puesto que utilizo ANSI y debo configurarla a *US international con teclas muertas*, no se cambia ningún parámetro por defecto.

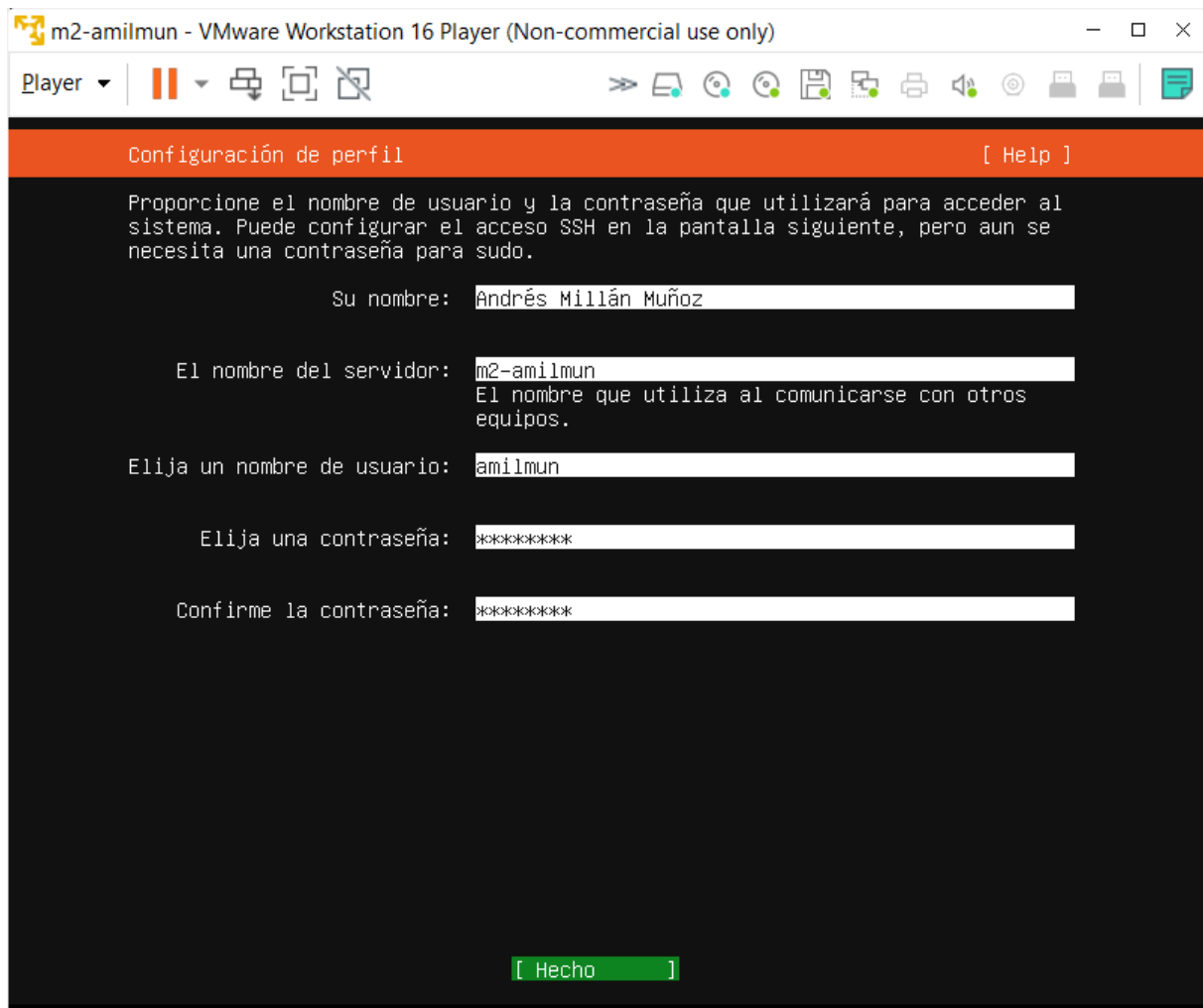












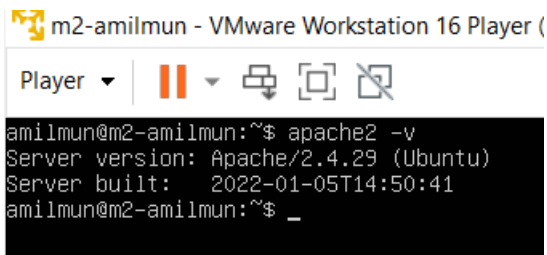
3 Programas básicos

3.1 LAMP

En el instalador no figuraba la opción para instalar LAMP, así que usaremos `tasksel` para hacerlo ahora:

```
1 sudo apt-get install tasksel
2 sudo tasksel install lamp-server
```

Si hacemos `apache2 -v`, vemos que aparece la versión:



```
m2-amilmun - VMware Workstation 16 Player (
Player | || | |
amilmun@m2-amilmun:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2022-01-05T14:50:41
amilmun@m2-amilmun:~$ _
```

Podemos comprobar que se está ejecutando con `ps aux | grep apache`:

```
amilmun@m2-amilmun:~$ ps aux | grep apache
root      10285  0.0  1.7 333736 16924 ?        Ss   16:44   0:00 /usr/sbin/apache2 -k start
www-data  10291  0.0  0.9 338136  9204 ?        S    16:44   0:00 /usr/sbin/apache2 -k start
www-data  10292  0.0  0.9 338136  9204 ?        S    16:44   0:00 /usr/sbin/apache2 -k start
www-data  10293  0.0  0.9 338136  9204 ?        S    16:44   0:00 /usr/sbin/apache2 -k start
www-data  10294  0.0  0.9 338136  9080 ?        S    16:44   0:00 /usr/sbin/apache2 -k start
www-data  10297  0.0  0.9 338136  9068 ?        S    16:44   0:00 /usr/sbin/apache2 -k start
amilmun   10923  0.0  0.1 13144  1040 tty1    S+   16:46   0:00 grep --color=auto apache
```

3.2 cURL

cURL está instalado por defecto, así que no será necesario ponerlo a mano.

```
amilmun@m2-amilmun:~$ curl --version
curl 7.58.0 (x86_64-pc-linux-gnu) libcurl/7.58.0 OpenSSL/1.1.1 zlib/1.2.11 libidn2/2.0.4 libpsl/0.19.1 (+libidn2/2.0.4) nghttp2/1.30.0 librtmp/2.3
Release-Date: 2018-01-24
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp smb smbs smtp smtps telnet tftp
Features: AsynchDNS IDN IPv6 Largefile GSS-API Kerberos SPNEGO NTLM NTLM_WB SSL libz TLS-SRP HTTP2 UnixSockets HTTPS-proxy PSL
amilmun@m2-amilmun:~$ _
```

3.2.1 Parámetros de cURL

3.2.1.1 Métodos de petición

Podemos especificar el tipo de petición que estamos haciendo con cURL. Por defecto, se utiliza `GET`. Si queremos utilizar otro tipo, podemos usar `-X` después del comando `curl`. Por ejemplo: `curl -X POST {url}`. Se puede mandar información gracias a `-d`. Al usarlo, no será necesario poner `-X POST`, puesto que se infiere.

3.2.1.2 Especificar la versión de http

Usando la opción `-0` o `--http1.0`, le decimos a cURL que queremos utilizar la versión 1.0 de http. Alternativamente, se puede usar `--http1.1` o `--http2`.

3.2.1.3 Redireccionando la salida

Por defecto, cURL escribe los contenidos a `stdout`. Este comportamiento se puede modificar usando `-o {archivo}`. Por ejemplo, si quisiéramos guardar [el resultado de la salida] al conectarnos a nuestras máquinas, podemos escribir

```
1 curl http://192.168.49.128/swap.html -o log_curl.txt
```

3.2.1.4 Especificar el puerto

Para especificar el puerto que se utilizará en la conexión, podemos usar `curl http://example.com:puerto`.

3.2.1.5 JSON

Una de las utilidades principales de cURL es la posibilidad de [enviar y recibir datos](#) en formato JSON.

Para recuperar un JSON, podemos usar el comando `curl -X GET -H "Content-Type: application/json"`.

Mientras que para mandarlo, utilizamos `-d` seguido del JSON que queramos adjuntar.

3.2.1.6 Cookies

Esta herramienta permite [añadir cookies](#) a las peticiones que hagamos. Para ello, basta con utilizar el parámetro `-b` o `--cookie`

```
1 curl -b '{nombre_cookie}={valor}' {url}
```

Por ejemplo, si queremos mandarle la sesión activa a la dirección: `curl -b 'session=amilmun' https://localhost/...`

Para añadir varias cookies en la misma orden, basta con poner `-b cookie` todas las veces que necesitemos.

Se pueden guardar las cookies en disco con la opción `-c nombre_archivo.txt`. De esta forma, podremos reutilizarlas usando `-b nombre_archivo.txt`

3.3 Apache

La carpeta de [configuración para Apache](#) se encuentra en `/etc/apache2`, siendo el archivo principal de configuración `/etc/apache2/apache2.conf`.

```

GNU nano 2.9.3 /etc/apache2/apache2.conf

# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
#
# Summary of how the Apache 2 configuration works in Debian:
# The Apache 2 web server configuration in Debian is quite different to
# upstream's suggested way to configure the web server. This is because Debian's
# default Apache2 installation attempts to make adding and removing modules,
# virtual hosts, and extra configuration directives as flexible as possible, in
# order to make automating the changes and administering the server as easy as
# possible.
#
# It is split into several files forming the configuration hierarchy outlined
# below, all located in the /etc/apache2/ directory:
#
#      /etc/apache2/
#      |-- apache2.conf
#      |   |-- ports.conf
#      |   |-- mods-enabled
#      |       |-- *.load
#      |       |-- *.conf
#      |   |-- conf-enabled
#      |       |-- *.conf
#      |   |-- sites-enabled
#      |       |-- *.conf
#
#
# * apache2.conf is the main configuration file (this file). It puts the pieces
# together by including all remaining configuration files when starting up the
# server.

[ File '/etc/apache2/apache2.conf' is unwritable ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos  M-U Undo
^X Exit      ^R Read File ^_ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line M-E Redo

```

En este archivo, podemos [cambiar algunos parámetros](#) como el archivo de configuración de puertos (por defecto, `Include ports.conf`), el nivel de logs, el timeout, `ServerRoot`...

Para alojar los archivos `.html`, se utiliza por defecto el directorio `/var/www/html`. Más adelante mostraremos un ejemplo de uso de esta carpeta.

```

amilmun@m1-amilmun:~$ cd /var/www/html/
amilmun@m1-amilmun:/var/www/html$ ls
index.html  swap.html
amilmun@m1-amilmun:/var/www/html$ _

```

3.3.1 Puertos de Apache

Si quisiéramos cambiar los puertos de Apache, debemos irnos al archivo `/etc/apache2/ports.conf` y cambiar el valor de `Listen` por el que queremos.

```
GNU nano 2.9.3 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

[Cancelled]
```

Get Help	Write Out	Where Is	Cut Text	Justify	Cur Pos	M-U Undo
Exit	Read File	Replace	Uncut Text	To Spell	Go To Line	M-E Redo

3.3.2 Hosts virtuales

Los hosts virtuales **nos permiten** alojar múltiples páginas web en un único servidor.

```

amilmun@m1-amilmun:/etc/apache2/sites-available$ ls
000-default.conf default-ssl.conf
amilmun@m1-amilmun:/etc/apache2/sites-available$ cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
amilmun@m1-amilmun:/etc/apache2/sites-available$

```

Figure 3.1: Configuración del host por defecto

Para configurar un host virtual, debemos crear un archivo en la carpeta `/etc/apache2/sites-available/`. Este archivo debe tener el mismo nombre que el host virtual que queremos crear. Por ejemplo, si queremos crear un host virtual para la página web `example.com`, debemos crear el archivo `example.com.conf` en la carpeta `/etc/apache2/sites-available/`. Este archivo debe ser similar al ejemplo del blog enlazado al principio de esta sección:

```

1 <VirtualHost *:80>
2     ServerAdmin email@gmail.com
3     ServerName example.com
4     ServerAlias www.example.com
5     DocumentRoot /var/www/example.com/public_html
6     <Directory /var/www/example.com/public_html>
7         Options Indexes FollowSymLinks
8         AllowOverride All
9         Require all granted
10    </Directory>
11    ErrorLog ${APACHE_LOG_DIR}/error.log

```



```
12 CustomLog ${APACHE_LOG_DIR}/access.log combined
13 <IfModule mod_dir.c>
14     DirectoryIndex index.php index.pl index.cgi index.html index.
        xhtml index.htm
15 </IfModule>
16 </VirtualHost>
```

Para habilitarlo, se usará el comando `sudo a2ensite example.com.conf`. Si queremos activar los certificados SSL, usamos `sudo certbot --apache -d example.com -d www.example.com`.

Finalmente, se aplica la configuración reiniciando el servicio de Apache con `sudo systemctl restart apache2`. Con esto, debería estar listo.

3.3.3 Redirección de puertos

Si se quiere redireccionar el puerto 80 al 8080, [hacemos lo siguiente](#):

1. Crear el archivo `/etc/apache2/other/port8080-redirect.conf`.
2. Usar la siguiente configuración:

```
1 <VirtualHost _default_:80>
2 DocumentRoot /ruta/a/la/pagina/web
3 RewriteEngine On
4 # Redirect all requests to the local Apache server to port 8080
5 RewriteRule ^.*$ http://%{HTTP_HOST}:8080%{REQUEST_URI}
6 </VirtualHost>
```

3. Reiniciar el servicio.

3.4 SSH

Hay una opción para poner SSH durante la instalación inicial de Ubuntu Server, así que se ha marcado.

El archivo de configuración de SSH se encuentra en `/etc/ssh/sshd_config`:

```
GNU nano 2.9.3 /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.101 2017/03/14 07:19:07 djm 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.
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#SyslogFacility AUTH
#LogLevel INFO
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
Search (to replace):
^G Get Help      M-C Case Sens  M-B Backwards  ^Y First Line   ^P PrevHstory
^C Cancel        M-R Regexp     ^R No Replace   ^V Last Line    ^N NextHstory
```

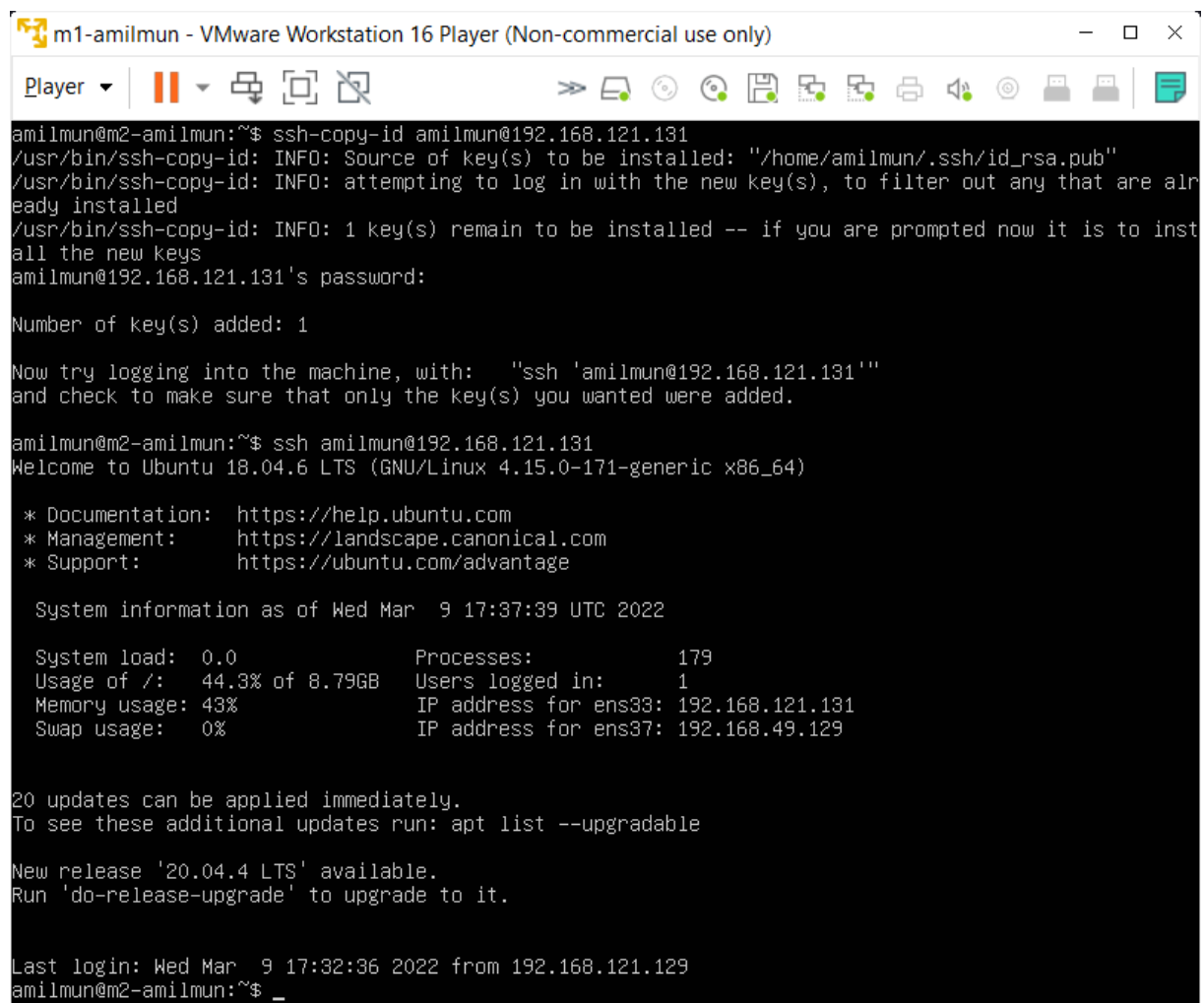
Si quisiéramos cambiar el puerto, simplemente cambiamos el parámetro `Port 22` al que queramos. De momento, lo dejaremos en el que viene por defecto.

Podemos ver un ejemplo de cómo utilizar este programa en la sección [Uso de SSH](#)

3.4.1 Conexión sin contraseña

Si queremos logearnos en las máquinas sin necesidad de poner la contraseña, tenemos que hacer lo siguiente:

1. Generar un nuevo par de claves SSH con `ssh-keygen`.
2. Modificar el archivo `/etc/ssh/sshd_config` poniendo `PermitRootLogin yes`.
3. Copiar la clave a la otra máquina con `ssh-copy-id`. En mi caso, para mandarla de M1 a M2, ponemos `ssh-copy-id amilmun@192.168.121.131`.



```
m1-amilmun - VMware Workstation 16 Player (Non-commercial use only)
Player ▾ | [Icons]
amilmun@m2-amilmun:~$ ssh-copy-id amilmun@192.168.121.131
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/amilmun/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter 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 all the new keys
amilmun@192.168.121.131's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'amilmun@192.168.121.131'"
and check to make sure that only the key(s) you wanted were added.
amilmun@m2-amilmun:~$ ssh amilmun@192.168.121.131
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-171-generic x86_64)

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

System information as of Wed Mar  9 17:37:39 UTC 2022

System load:  0.0                Processes:            179
Usage of /:   44.3% of 8.79GB     Users logged in:     1
Memory usage: 43%                IP address for ens33: 192.168.121.131
Swap usage:   0%                 IP address for ens37: 192.168.49.129

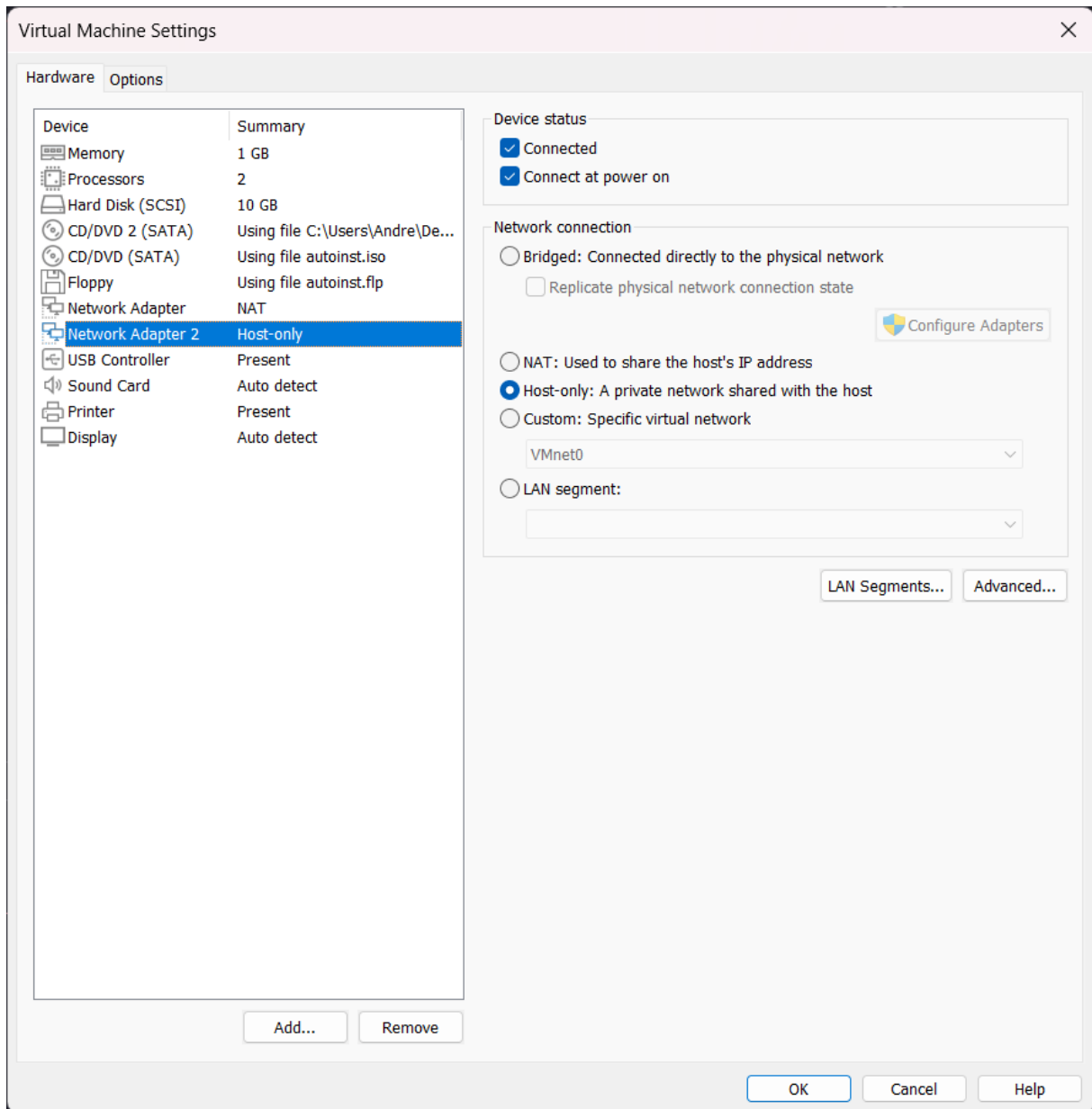
20 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

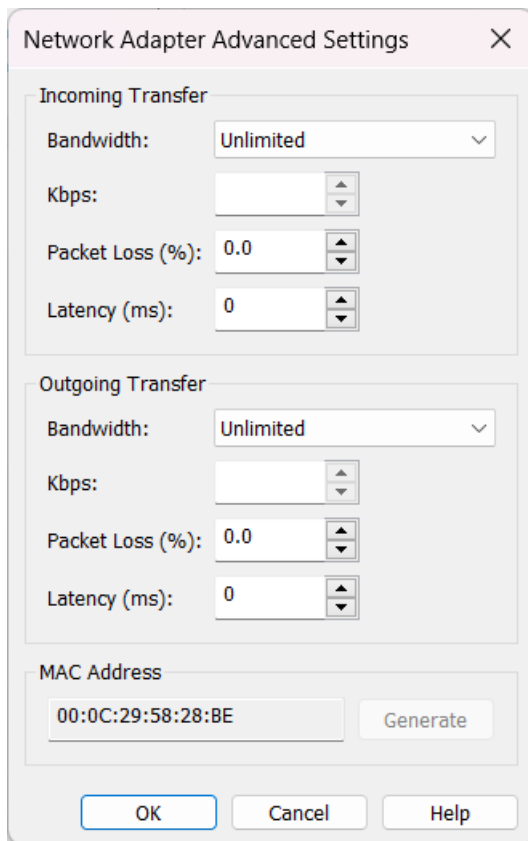
Last login: Wed Mar  9 17:32:36 2022 from 192.168.121.129
amilmun@m2-amilmun:~$ _
```

4 Configurando la interfaz de red

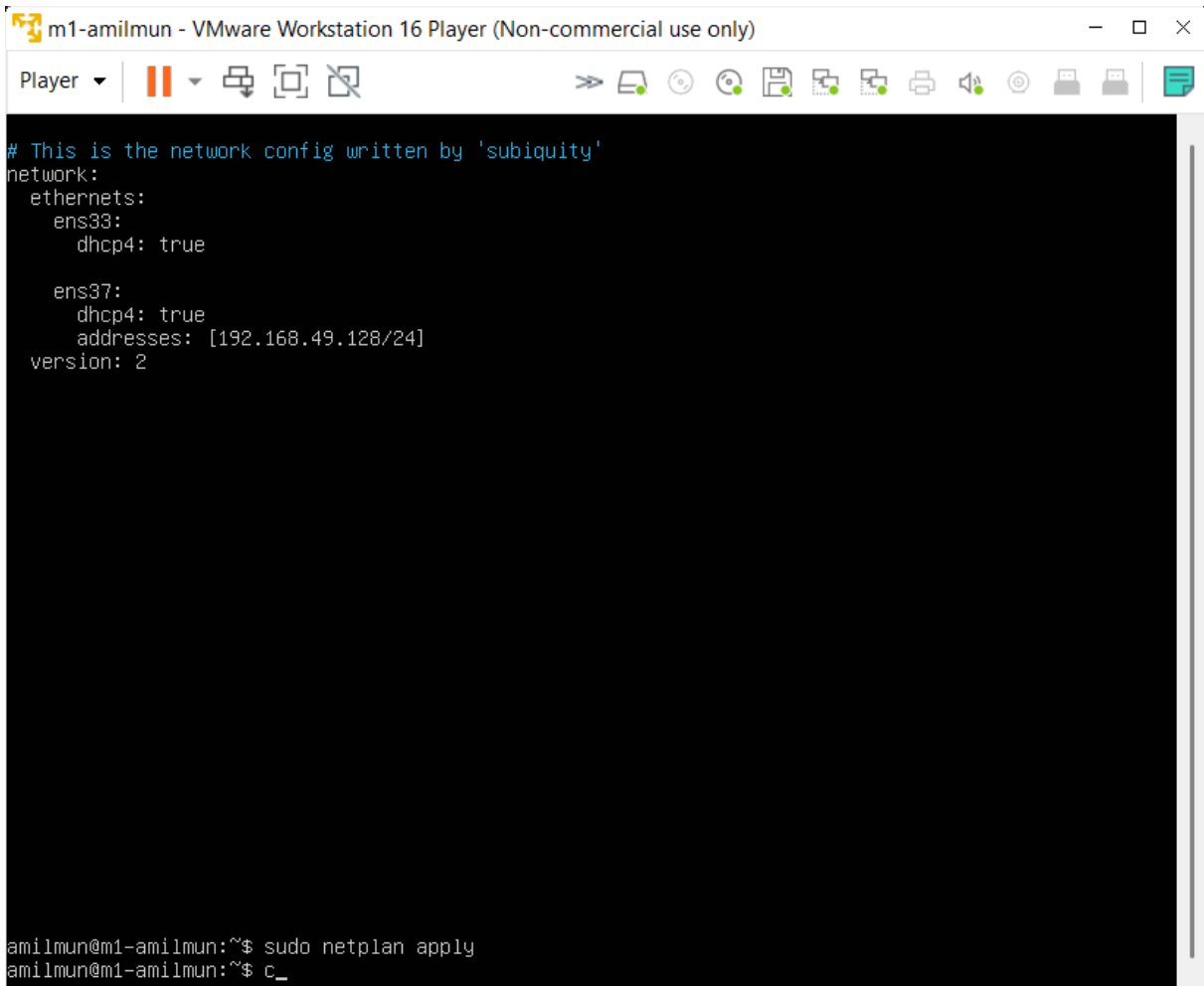
Añadiremos un nuevo adaptador de red desde VMWare del tipo *host only*:



Podríamos configurar algunas opciones avanzadas del adaptador, como la MAC o simular un throttling de la red. No nos será necesario, así que no lo haremos



Los planes de red se encuentran almacenados en `/etc/netplan`. Vamos a añadir un nuevo adaptador `host-only` y configurarlo para fijar las IPs. En `m1`, será `192.168.49.128`, mientras que en `m2` `192.168.49.129`. Para ello, [ponemos lo siguiente](#):



```
m1-amilmun - VMware Workstation 16 Player (Non-commercial use only)
Player
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      dhcp4: true

    ens37:
      dhcp4: true
      addresses: [192.168.49.128/24]
  version: 2

amilmun@m1-amilmun:~$ sudo netplan apply
amilmun@m1-amilmun:~$ c_
```

Vamos a crear dos páginas sencillas en las máquinas virtuales. Ponemos los siguientes archivos en `/var/www/html/swap.html`:

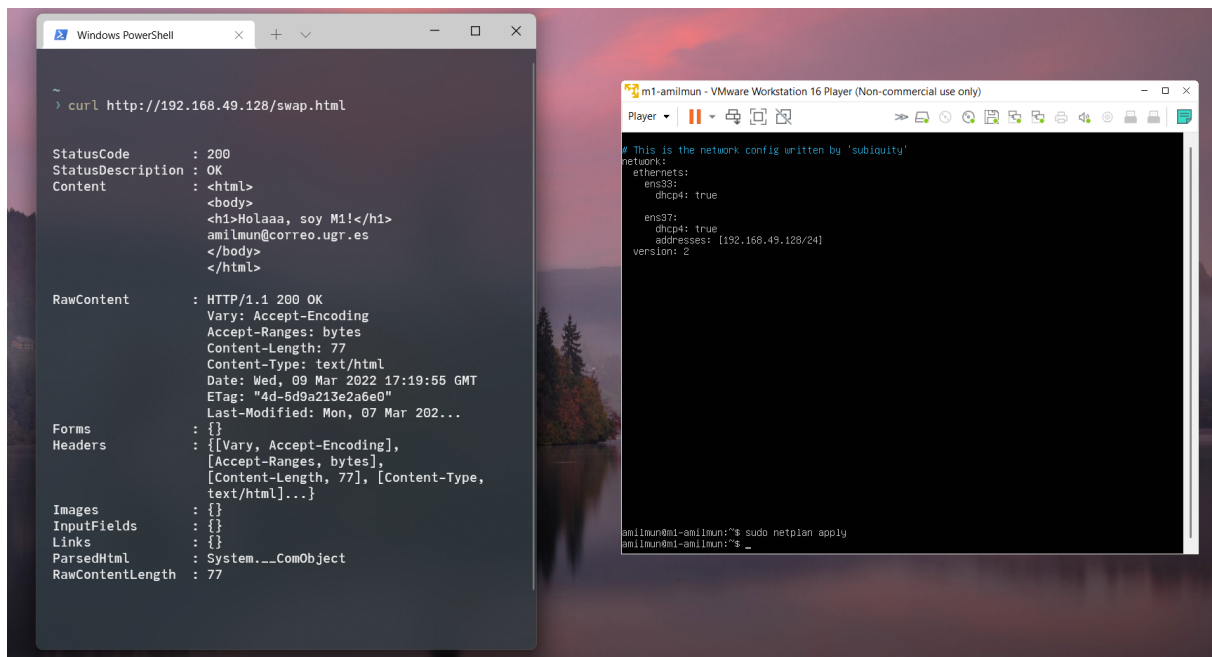
En m1:

```
1 <html>
2 <body>
3 <h1>Holaaa, soy M1!</h1>
4 </body>
5 </html>
```

Y en m2:

```
1 <html>
2 <body>
3 <h1>Soy M2 o</h1>
4 </body>
5 </html>
```

Si desde el host hacemos `curl http://192.168.49.128/swap.html`, obtenemos

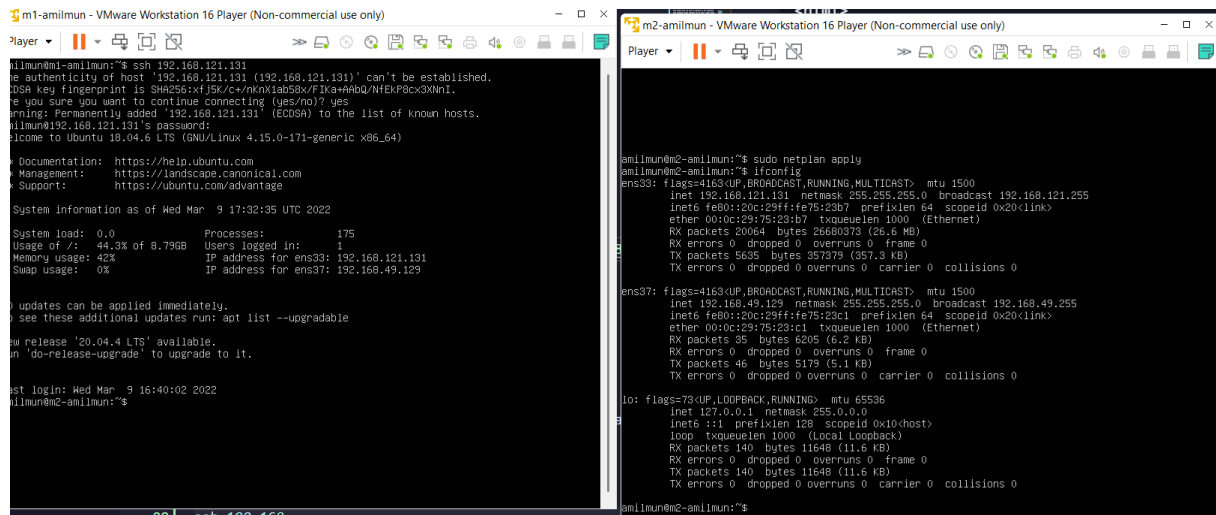


5 Uso de SSH

Vamos a conectar las dos máquinas mediante SSH. Para acceder desde M1 a M2, ponemos

```
1 ssh 192.168.121.131
```

Y obtenemos



The image shows two terminal windows from VMware Workstation 16 Player. The left window, titled 'm1-amilmun - VMware Workstation 16 Player (Non-commercial use only)', shows an SSH session from m1 to m2. The user 'amilmun' on m1 connects to '192.168.121.131'. The terminal output shows the SSH handshake, including the host's SSH version, the user's login attempt, and the system information of m2 (Ubuntu 18.04.6 LTS). The right window, titled 'm2-amilmun - VMware Workstation 16 Player (Non-commercial use only)', shows the user 'amilmun' on m2 running 'sudo netplan apply' and 'ifconfig'. The output of 'ifconfig' shows the configuration for 'ens33' and 'lo', including IP addresses, netmasks, and broadcast addresses.

```
m1mun@m1-amilmun:~$ ssh 192.168.121.131
Warning: Permanently added '192.168.121.131' (ECDSA) to the list of known hosts.
m1mun@192.168.121.131's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-171-generic x86_64)

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

System information as of Wed Mar  9 17:32:35 UTC 2022

System load:  0.0          Processes:      175
Usage of /:   44.3% of 8.79GB Users logged in:    1
Memory usage: 42%         IP address for ens33: 192.168.121.131
Swap usage:   0%          IP address for ens37: 192.168.49.129

Updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Mar  9 16:40:02 2022
m1mun@m2-amilmun:~$
```

```
m2mun@m2-amilmun:~$ sudo netplan apply
m2mun@m2-amilmun:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.121.131 netmask 255.255.255.0 broadcast 192.168.121.255
    inet6 fe80::20c:29ff:fe75:23b7 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:75:23:b7 txqueuelen 1000 (Ethernet)
    RX packets 20064 bytes 26680373 (26.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5639 bytes 857379 (857.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.49.129 netmask 255.255.255.0 broadcast 192.168.49.255
    inet6 fe80::20c:29ff:fe75:23c1 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:75:23:c1 txqueuelen 1000 (Ethernet)
    RX packets 35 bytes 6205 (6.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 46 bytes 5179 (5.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 140 bytes 11648 (11.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 140 bytes 11648 (11.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

m2mun@m2-amilmun:~$
```