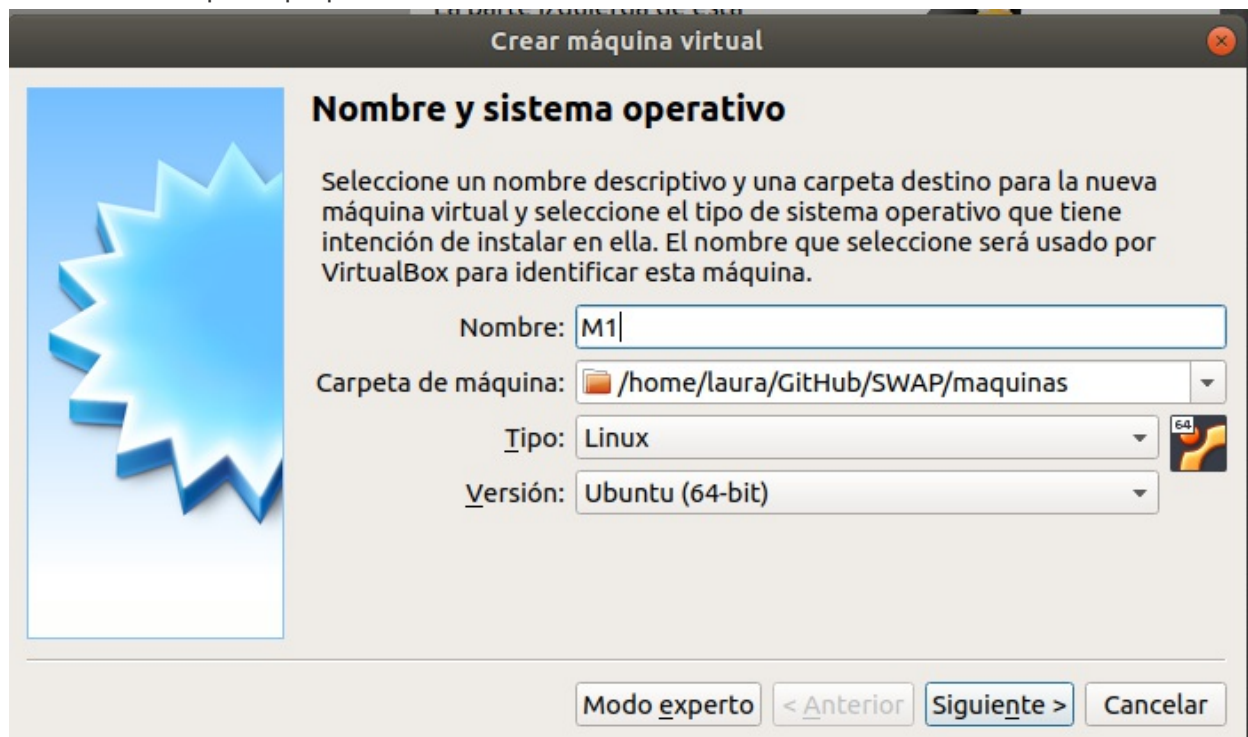


Práctica 1

Comenzaremos descargando VirtualBox 6.1.4 r136177 y Ubuntu Server 18.04.4 en nuestro dispositivo que será un portátil ASUS con Ubuntu 18.04.3 con un procesador Intel® Core™ i7-4510U CPU @ 2.00GHz × 4, 7,7GiB de memoria y 983.4GB de disco. Será, en este dispositivo, donde utilizaremos VirtualBox para virtualizar las máquinas que crearemos para la asignatura.

Creación de las máquinas virtuales M1 y M2

En las imágenes mostraremos el procedimiento seguido tomando como ejemplo la máquina M1. La única diferencia con respecto al procedimiento indicado en el guión, sería que las máquinas han sido creadas con 1024 MB de RAM en lugar de 512 MB, siguiendo con las indicaciones recomendadas por el propio VirtualBox.



Crear máquina virtual

Nombre y sistema operativo

Seleccione un nombre descriptivo y una carpeta destino para la nueva máquina virtual y seleccione el tipo de sistema operativo que tiene intención de instalar en ella. El nombre que seleccione será usado por VirtualBox para identificar esta máquina.

Nombre:

Carpeta de máquina:

Tipo:

Versión:

Modo experto < Anterior Siguiente > Cancelar

Crear máquina virtual



Tamaño de memoria

Seleccione la cantidad de memoria (RAM) en megabytes a ser reservada para la máquina virtual.

El tamaño de memoria recomendado es **1024 MB**.

4 MB

8192 MB

1024 MB

< Anterior

Siguiente >

Cancelar

Crear máquina virtual



Disco duro

Si desea puede añadir un disco duro virtual a la nueva máquina. Puede crear un nuevo archivo de disco duro o seleccionar uno de la lista o de otra ubicación usando el icono de la carpeta.

. Si necesita una configuración de almacenamiento más compleja puede omitir este paso y hacer los cambios a las preferencias de la máquina virtual una vez creada.

El tamaño recomendado del disco duro es **10,00 GB**.

☐ No añadir un disco duro virtual

☒ Crear un disco duro virtual ahora

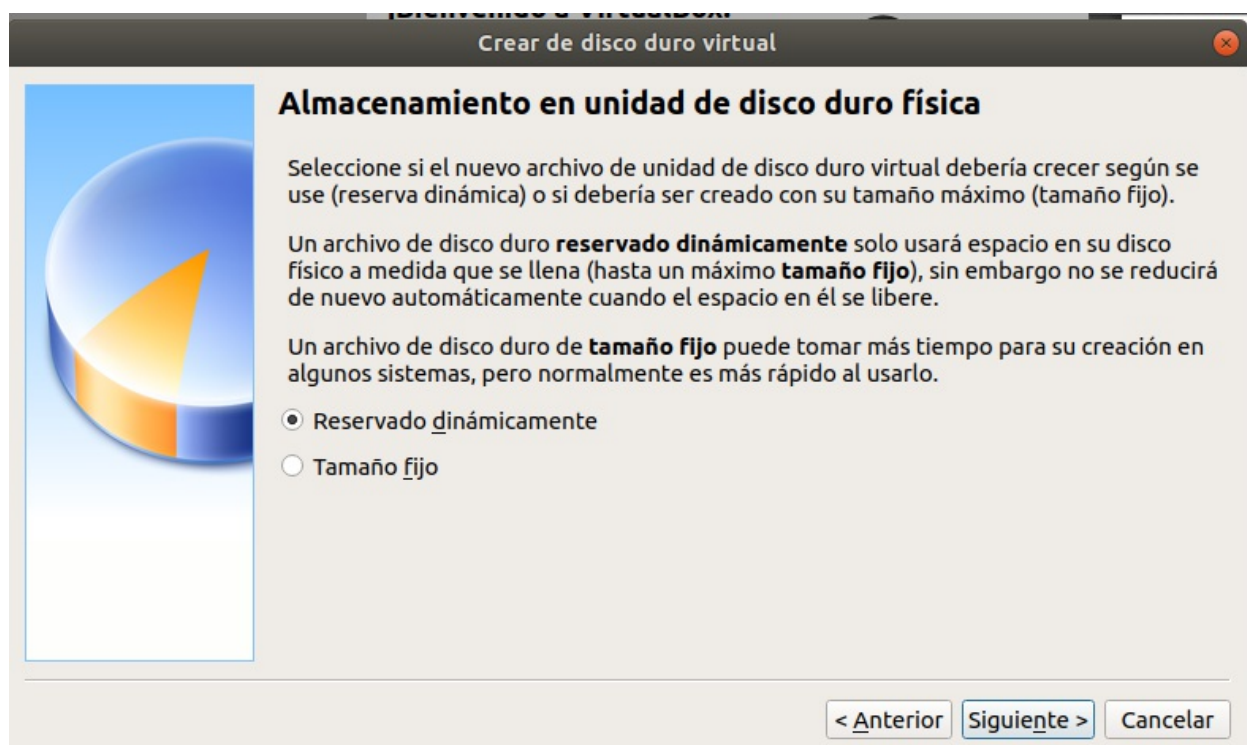
☐ Usar un archivo de disco duro virtual existente

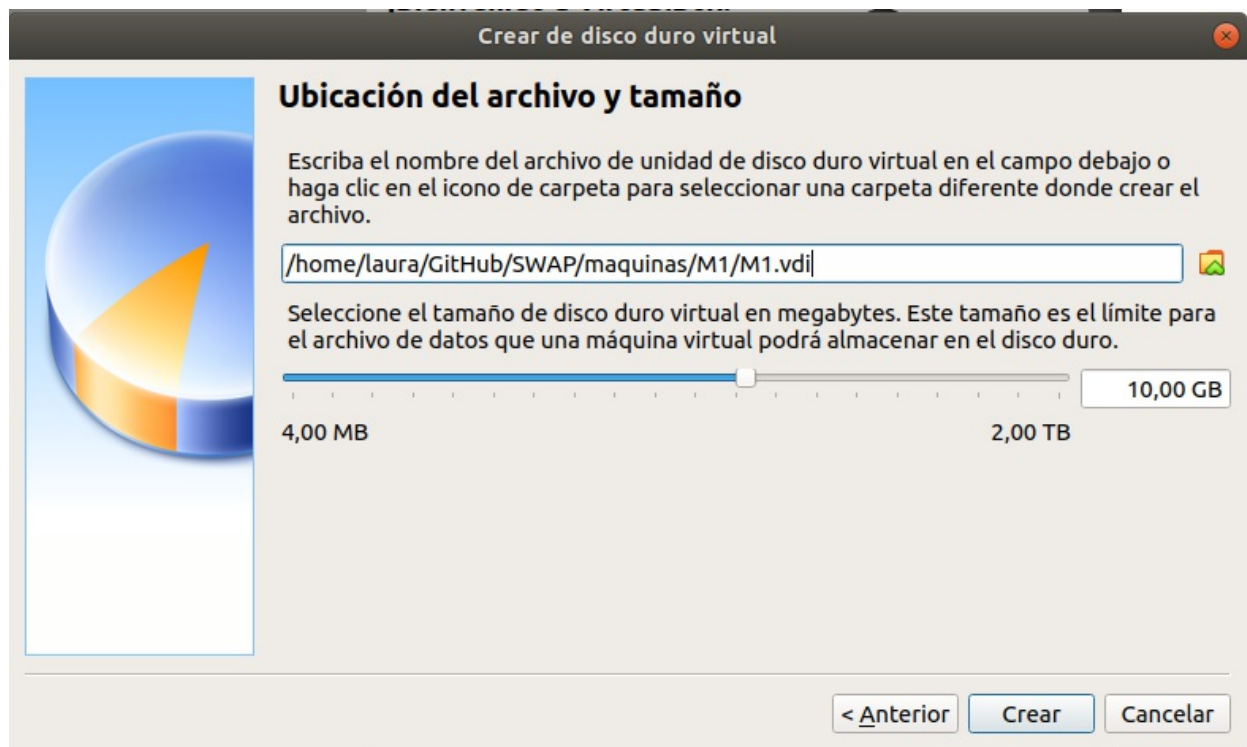
Vacío

< Anterior

Crear

Cancelar





Instalación de Ubuntu Server

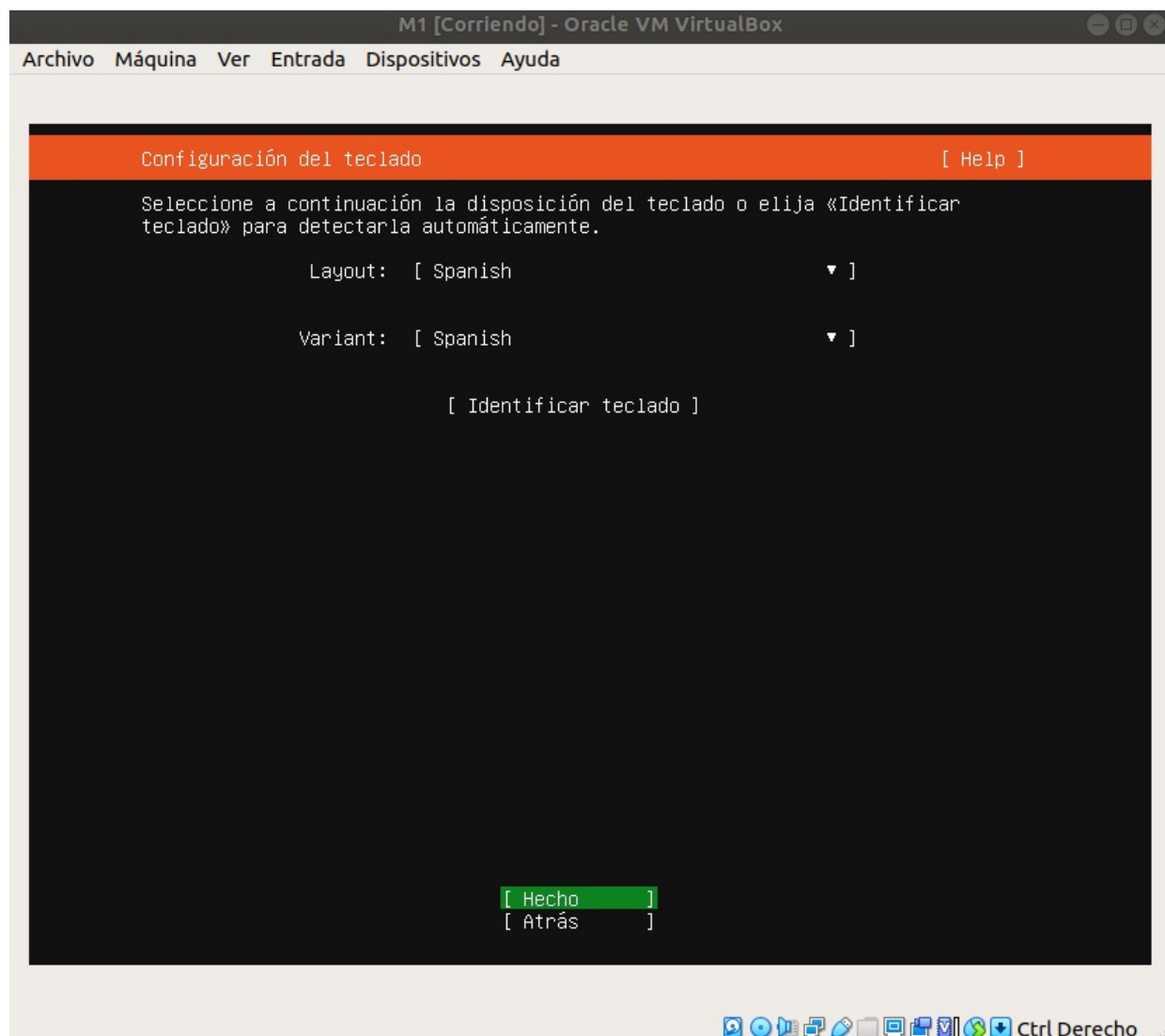
A continuación, mostramos el procedimiento seguido para instalar Ubuntu Server. Nótese que SSH es instalado mediante este procedimiento.

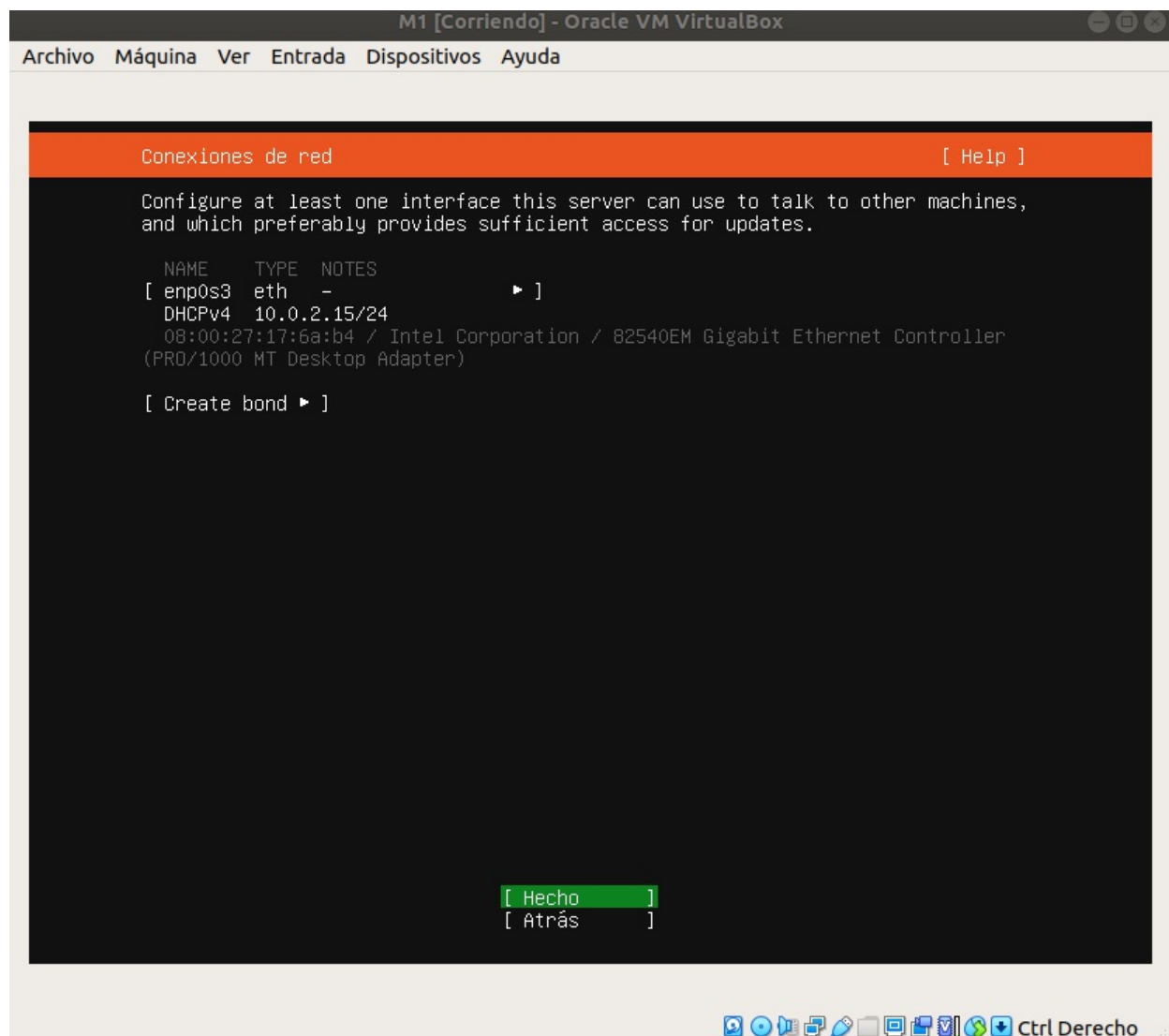
Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom!

[Help]

Use UP, DOWN and ENTER keys to select your language.

[English	▶]
[Asturianu	▶]
[Català	▶]
[Hrvatski	▶]
[Nederlands	▶]
[Suomi	▶]
[Français	▶]
[Deutsch	▶]
[Ελληνικά	▶]
[Magyar	▶]
[Latviešu	▶]
[Norsk bokmål	▶]
[Polski	▶]
[Русский	▶]
[Español	▶]
[Українська	▶]





If this system requires a proxy to connect to the internet, enter its details here.

Proxy address:

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user] [:pass]@]host[:port]/".

[Hecho]
[Atrás]

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:

<http://es.archive.ubuntu.com/ubuntu>

You may provide an archive mirror that will be used instead of the default.

[Hecho]
[Atrás]

Configure a guided storage layout, or create a custom one:

☒ Use an entire disk

[VBOX_HARDDISK_VB836e568e-03202ec1 disco local 10.000G ▼]

☐ Set up this disk as an LVM group

☐ Encrypt the LVM group with LUKS

Passphrase:

Confirm passphrase:

☐ Custom storage layout

[Hecho]
[Atrás]

RESUMEN DEL SISTEMA DE ARCHIVOS

PUNTO DE MONTAJE	TAMAÑO	TIPO	TIPO DE DISPOSITIVO
[/	9.997G	new ext4	new partición de disco local ▶]

DISPOSITIVOS DISPONIBLES

No available devices

[Create software RAID (md) ▶]
[Crear grupo de volúmenes (LVM) ▶]

DISPOSITIVOS UTILIZADOS

DISPOSITIVO	TIPO	TAMAÑO
[VBOX_HARDDISK_VB836e568e-03202ec1	disco local	10.000G ▶]
partición 1 new, bios_grub		1.000M ▶]
partición 2 new, to be formatted as ext4, mounted at /		9.997G ▶]

[Hecho]
[Restablecer]
[Atrás]

Confirmar acción destructiva

Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the installation has started.

Are you sure you want to continue?

[No]
[Continuar]

Proporcione el nombre de usuario y la contraseña que utilizará para acceder al sistema. Puede configurar el acceso SSH en la pantalla siguiente, pero aun se necesita una contraseña para sudo.

Your name:

Your server's name:
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

[Hecho]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☒ Install OpenSSH server

Import SSH identity:
You can import your SSH keys from Github or Launchpad.

Importar nombre de
usuario:

☒ Allow password authentication over SSH

[Hecho]
[Atrás]

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of the package, publisher and versions available.

```
[ ] microk8s           Kubernetes for workstations and appliances ▶
[ ] nextcloud          Nextcloud Server - A safe home for all your data ▶
[ ] wekan              Open-Source kanban ▶
[ ] kata-containers    Lightweight virtual machines that seamlessly plug into ▶
[ ] docker             Docker container runtime ▶
[ ] canonical-livepatch Canonical Livepatch Client ▶
[ ] rocketchat-server  Group chat server for 100s, installed in seconds. ▶
[ ] mosquitto          Eclipse Mosquitto MQTT broker ▶
[ ] etcd               Resilient key-value store by CoreOS ▶
[ ] powershell         PowerShell for every system! ▶
[ ] stress-ng          A tool to load, stress test and benchmark a computer ▶
[ ] sabnzbd            SABnzbd ▶
[ ] wormhole           get things from one computer to another, safely ▶
[ ] aws-cli            Universal Command Line Interface for Amazon Web Services ▶
[ ] google-cloud-sdk   Command-line interface for Google Cloud Platform products ▶
[ ] slcli              Python based SoftLayer API Tool. ▶
[ ] doctl              DigitalOcean command line tool ▶
[ ] conjure-up          Package runtime for conjure-up spells ▶
[ ] minidlna-escoand    server software with the aim of being fully compliant ▶
[ ] postgresql10       PostgreSQL is a powerful, open source object-relational database ▶
[ ] heroku             CLI client for Heroku ▶
[ ] keepalived          High availability VRRP/BFD and load-balancing for Linux ▶
[ ] prometheus          The Prometheus monitoring system and time series data ▶
[ ] juju               Simple, secure and stable devops. Juju keeps complexity ▶
```

[Hecho]
[Atrás]

```

      _____ Ha finalizado la instalación. _____
installing system
  curtin command install
    preparing for installation
    configuring storage
      running 'curtin block-meta simple'
      curtin command block-meta
        removing previous storage devices
        configuring disk: disk-sda
        configuring partition: partition-0
        configuring partition: partition-1
        configuring format: format-0
        configuring mount: mount-0
    configuring network
      running 'curtin net-meta auto'
      curtin command net-meta
    writing install sources to disk
      running 'curtin extract'
      curtin command extract
        acquiring and extracting image from cp:///media/filesystem
    configuring installed system
      running '/snap/bin/subiquity.subiquity-configure-run'
      running '/snap/bin/subiquity.subiquity-configure-apt'
/snap/subiquity/1561/usr/bin/python3 true'
    curtin command apt-config
    curtin command in-target
    running 'curtin curthooks'
    curtin command curthooks
    configuring apt configuring apt

```

[View full log]
[Reiniciar]

Se ha completado la instalación.

[Help]

```
Ha finalizado la instalación.
running '/snap/bin/subiquity.subiquity-configure-apt
/snap/subiquity/1561/usr/bin/python3 true'
  curtin command apt-config
  curtin command in-target
running 'curtin curthooks'
  curtin command curthooks
    configuring apt configuring apt
    installing missing packages
    configuring iscsi service
    configuring raid (mdadm) service
    installing kernel
    setting up swap
    apply networking config
    writing etc/fstab
    configuring multipath
    updating packages on target system
    configuring pollinate user-agent on target
    updating initramfs configuration
finalizing installation
  running 'curtin hook'
  curtin command hook
executing late commands
final system configuration
  configuring cloud-init
  installing openssh
  restoring apt configuration
  downloading and installing security updates
  copying logs to installed system
```

[View full log]

[Reiniciar]

Instalación de programas.

Dicho esto, instalamos apache, php, mysql y curl. Además de asegurarnos de que estos funcionan correctamente.

```
lauragg@m1:~$ sudo apt-get install apache2 mysql-server mysql-client
[sudo] password for lauragg:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libaio1 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libfcgi-fast-perl libfcgi-pm-perl libencode-locale-perl libevent-core-2.1-6
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblua5.2-0 liblwp-mediatypes-perl libtimedate-perl
  liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server-5.7
  mysql-server-core-5.7 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom libdata-dump-perl
  libipc-sharedcache-perl libwww-perl mailx tinycat openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libaio1 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libfcgi-fast-perl libfcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblua5.2-0 liblwp-mediatypes-perl
  libtimedate-perl liburi-perl mysql-client mysql-client-5.7 mysql-client-core-5.7 mysql-common
  mysql-server mysql-server-5.7 mysql-server-core-5.7 ssl-cert
0 upgraded, 32 newly installed, 0 to remove and 11 not upgraded.
Need to get 21.4 MB of archives.
After this operation, 163 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

```

lauragg@m1:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2019-12-03T15:55:03
lauragg@m1:~$ ps aux | grep apache
root      2617  0.0  0.4 73960 4500 ?        Ss   14:44   0:00 /usr/sbin/apache2 -k start
www-data  2619  0.0  0.4 826256 4596 ?        Sl   14:44   0:00 /usr/sbin/apache2 -k start
www-data  2620  0.0  0.4 826256 4596 ?        Sl   14:44   0:00 /usr/sbin/apache2 -k start
lauragg@m1:~$ sudo service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Wed 2020-03-18 14:44:24 UTC; 4min 4s ago
   Main PID: 2617 (apache2)
   Tasks: 55 (limit: 1108)
   CGroup: /system.slice/apache2.service
           └─2617 /usr/sbin/apache2 -k start
             └─2619 /usr/sbin/apache2 -k start
               └─2620 /usr/sbin/apache2 -k start

Mar 18 14:44:24 m1 systemd[1]: Starting The Apache HTTP Server...
Mar 18 14:44:24 m1 apachectl[2593]: AH00558: apache2: Could not reliably determine the server's full
Mar 18 14:44:24 m1 systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
lauragg@m1:~$ _

```

```

lauragg@m1:~$ sudo apt-get install php
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli php7.2-common php7.2-json
  php7.2-opcache php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php7.2 libsodium23 php php-common php7.2 php7.2-cli php7.2-common php7.2-json
  php7.2-opcache php7.2-readline
0 upgraded, 10 newly installed, 0 to remove and 15 not upgraded.
Need to get 4011 kB of archives.
After this operation, 17.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] _

```

```

lauragg@m1:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
lauragg@m1:~$ sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web
server.

Ports:
  80,443/tcp
lauragg@m1:~$ sudo ufw allow in "Apache Full"
Rules updated
Rules updated (v6)
lauragg@m1:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
lauragg@m1:~$ curl http://icanhazip.com
84.76.19.65
lauragg@m1:~$ _

```

Aquí obtenemos una imagen sencilla de una web externa, utilizando curl.

```
lauragg@m1:~$ curl -o imagen.png https://www.google.es/images/srpr/logo3w.png
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100 6748  100 6748    0     0  12007      0  --:--:-- --:--:-- --:--:-- 12007
lauragg@m1:~$ ls
imagen.png
lauragg@m1:~$
```

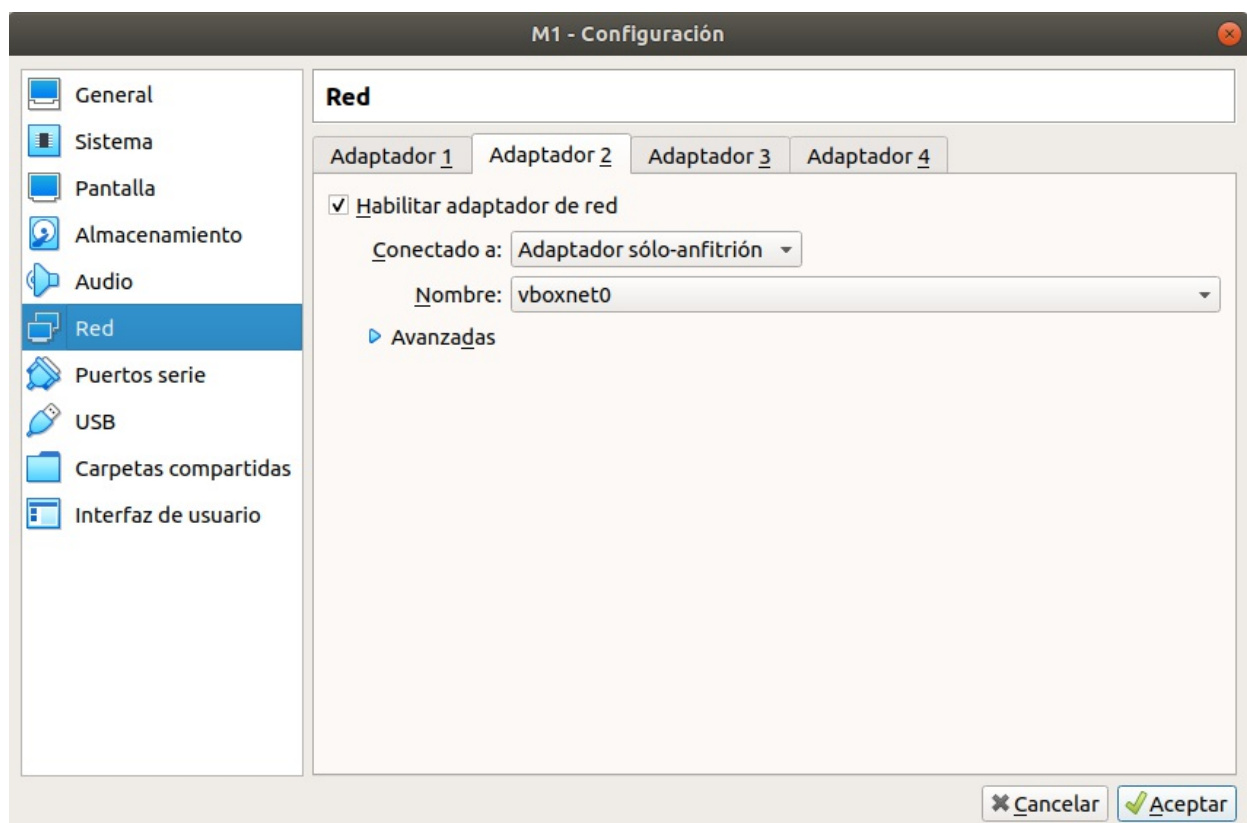
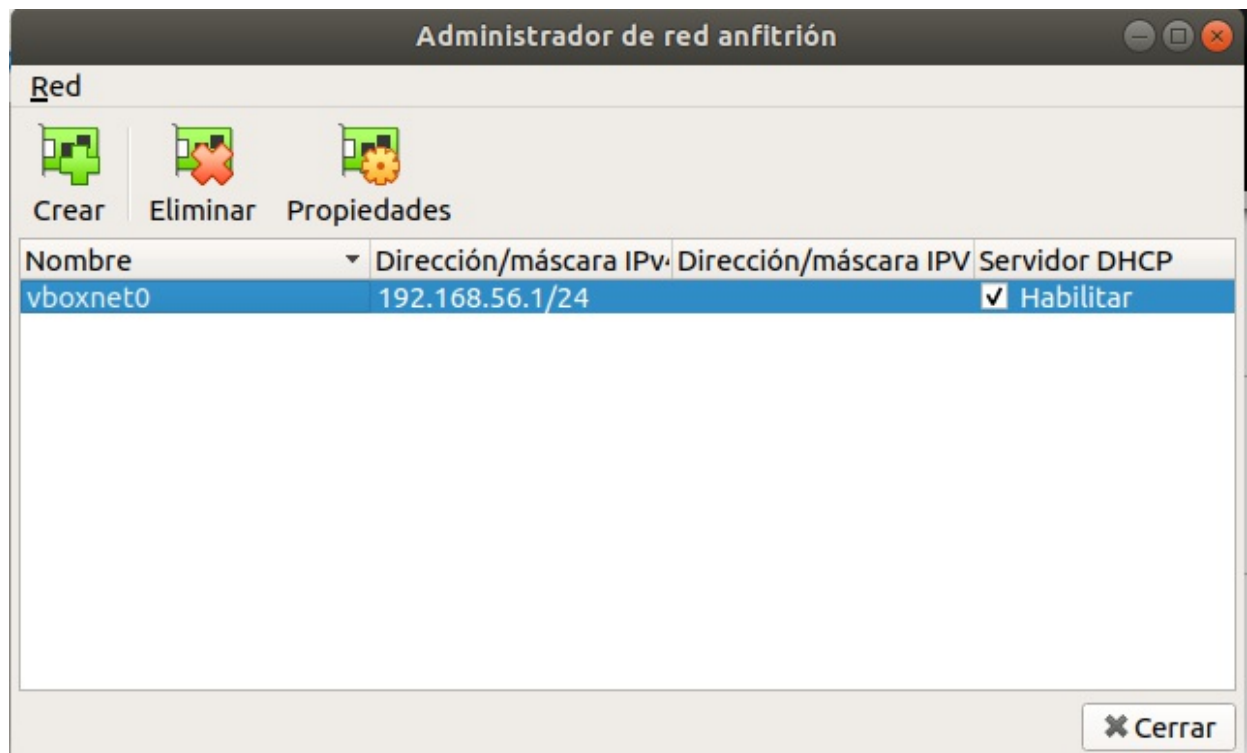
Acceso curl entre máquinas

Sin embargo, si intentamos hacer curl desde m2 hacia m1 para obtener el archivo ejemplo.html veremos que no se realiza la descarga. Esto es por varios motivos, por ejemplo, en este caso podríamos plantearnos hacer curl a través de la dirección **84.76.19.65** la cual, si nos fijamos, es compartida entre nuestras dos máquinas y el host. Si hacemos eso, estaríamos accediendo a nuestro router, pero ahí no tenemos ningún archivo que coincida con el nombre solicitado.

```
lauragg@m1:/var/www/html$ ls
ejemplo.html  index.html
lauragg@m1:/var/www/html$ cat ejemplo.html
<HTML>
    <BODY>
        Web de ejemplo de Lauragg para SWAP
    </BODY>
</HTML>

lauragg@m1:/var/www/html$ curl http://www.icanhazip.com
^C
lauragg@m1:/var/www/html$ curl http://icanhazip.com
84.76.19.65
```

Primero de todo, debemos de configurar la red que comunica m1, m2 y host de acuerdo al esquema del guión de prácticas. De esta forma, podremos realizar la petición curl utilizando este camino que estamos creando. Mostramos cómo sería con m1, sabiendo que con m2 se haría de forma similar.




```
# ifupdown has been replaced by netplan(5) on this system. See
# /etc/netplan for current configuration.
# To re-enable ifupdown on this system, you can run:
#     sudo apt install ifupdown
```

```
source /etc/network/interfaces.d/*
```

```
auto lo
iface lo inet loopback
```

```
auto enp0s3
iface enp0s3 inet dhcp
```

```
auto enp0s8
iface enp0s8 inet static
address 192.168.56.200
netmask 255.255.255.0
```

[illegible]

```
"/etc/network/interfaces" 17L, 378C
```

15,2

```
source /etc/network/interfaces.d/*
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto enp0s3
```

```
iface enps3 inet dhcp
```

```
auto enp0s8
```

```
iface enp0s8 inet static
```

```
address 192.168.56.100
```

```
netmask 255.255.255.0
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"/etc/network/interfaces" 16L, 375C written
```

```
lauragg@m1:~$ ifup enp0s8
```

```
ifup: failed to open lockfile /run/network/ifstate.enp0s8: Permission denied
```

```
lauragg@m1:~$ sudo ifup enp0s8
```

```
lauragg@m1:~$
```

```

lauragg@m1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe17:6ab4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:17:6a:b4 txqueuelen 1000 (Ethernet)
    RX packets 12 bytes 2070 (2.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23 bytes 2300 (2.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.100 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe27:bca7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:27:bc:a7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 656 (656.0 B)
    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 84 bytes 6324 (6.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 6324 (6.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lauragg@m1:~$

```

```

lauragg@m1:~$ ping 192.168.56.200
PING 192.168.56.200 (192.168.56.200) 56(84) bytes of data.
64 bytes from 192.168.56.200: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 192.168.56.200: icmp_seq=2 ttl=64 time=0.866 ms
64 bytes from 192.168.56.200: icmp_seq=3 ttl=64 time=0.782 ms
^C
--- 192.168.56.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.782/0.889/1.021/0.104 ms
lauragg@m1:~$

```

Una vez hecho esto, ¡funciona! Ya podemos proseguir con el acceso SSH.

```

lauragg@m2:~$ curl 192.168.56.100/ejemplo.html
<HTML>
    <BODY>
        Web de ejemplo de Lauragg para SWAP
    </BODY>
</HTML>

lauragg@m2:~$ _

```

Acceso SSH entre máquinas

Nótese, que estamos utilizando el mismo usuario para ambas máquinas. Si quisiéramos acceder

con otro usuario, tendríamos que configurar ssh para admitirle el acceso.

```
lauragg@m1:~$ ssh 192.168.56.200
lauragg@192.168.56.200's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

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

System information as of Sun Mar 22 02:12:48 UTC 2020

System load:  0.07               Processes:            101
Usage of /:   43.5% of 9.78GB    Users logged in:     1
Memory usage: 34%               IP address for enp0s3: 10.0.2.15
Swap usage:   0%                IP address for enp0s8: 192.168.56.200

 * Latest Kubernetes 1.18 beta is now available for your laptop, NUC, cloud
   instance or Raspberry Pi, with automatic updates to the final GA release.

   sudo snap install microk8s --channel=1.18/beta --classic

 * Multipass 1.1 adds proxy support for developers behind enterprise
   firewalls. Rapid prototyping for cloud operations just got easier.

   https://multipass.run/

15 packages can be updated.
0 updates are security updates.

Last login: Sun Mar 22 02:09:49 2020 from 192.168.56.100
lauragg@m2:~$
```

```
lauragg@m2:~$ ssh 192.168.56.100
lauragg@192.168.56.100's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

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

System information as of Sun Mar 22 02:14:25 UTC 2020

System load: 0.01          Processes:              98
Usage of /:  43.3% of 9.78GB Users logged in:             1
Memory usage: 33%          IP address for enp0s3: 10.0.2.15
Swap usage:  0%            IP address for enp0s8: 192.168.56.100

 * Latest Kubernetes 1.18 beta is now available for your laptop, NUC, cloud
   instance or Raspberry Pi, with automatic updates to the final GA release.

   sudo snap install microk8s --channel=1.18/beta --classic

 * Multipass 1.1 adds proxy support for developers behind enterprise
   firewalls. Rapid prototyping for cloud operations just got easier.

   https://multipass.run/

15 packages can be updated.
0 updates are security updates.

Last login: Sun Mar 22 02:14:08 2020
lauragg@m1:~$
```