



SERVIDORES WEB DE ALTAS PRESTACIONES

PRÁCTICA 1: Preparación de las herramientas

Autor

Miguel Ángel Pérez Díaz



Escuela Técnica Superior de Ingenierías Informática y
de Telecomunicación

—
Granada, 2020

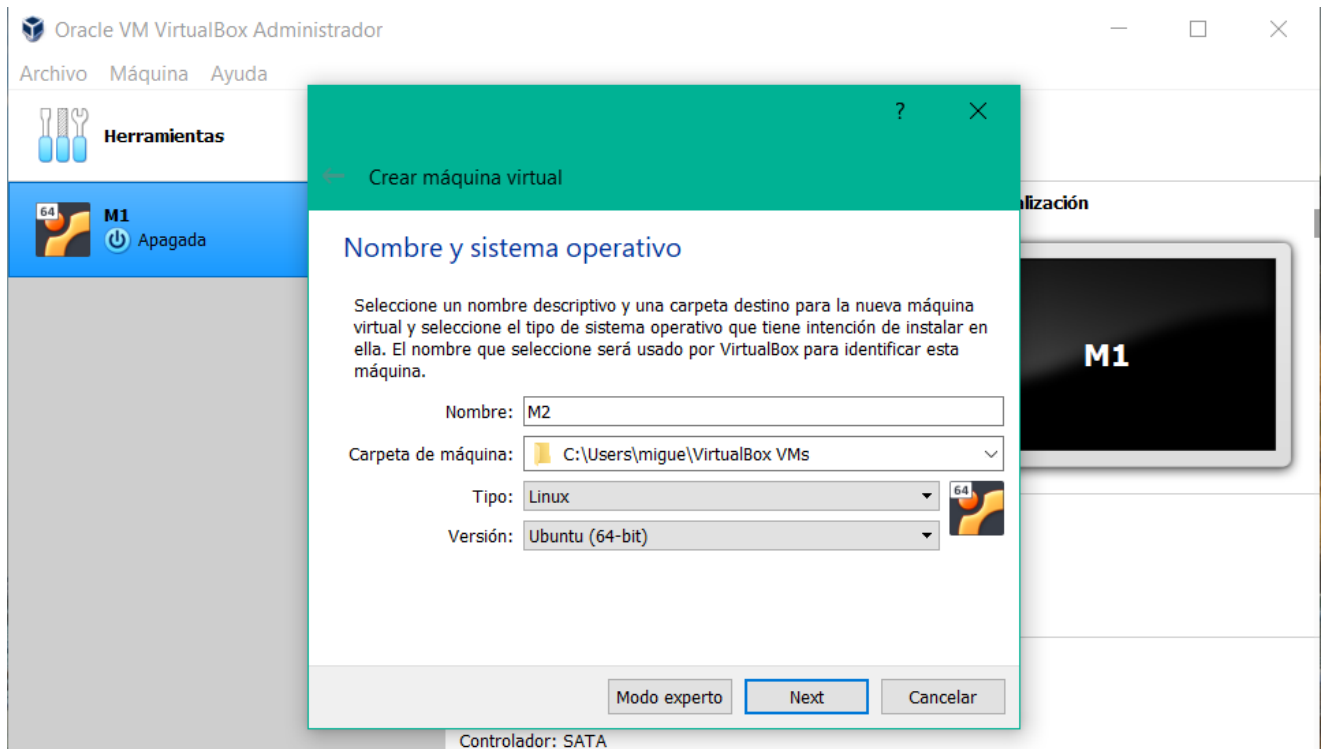
1. Preparación de las máquinas virtuales

Para esta práctica inicialmente debemos crear dos máquinas virtuales llamadas M1 y M2, las cuáles dispondrán de 512 MB de memoria RAM y un disco duro dinámico de 10GB.

La siguiente imagen representa como se ha creado una de las máquinas virtuales pero el proceso se ha realizado sobre ambas máquinas.

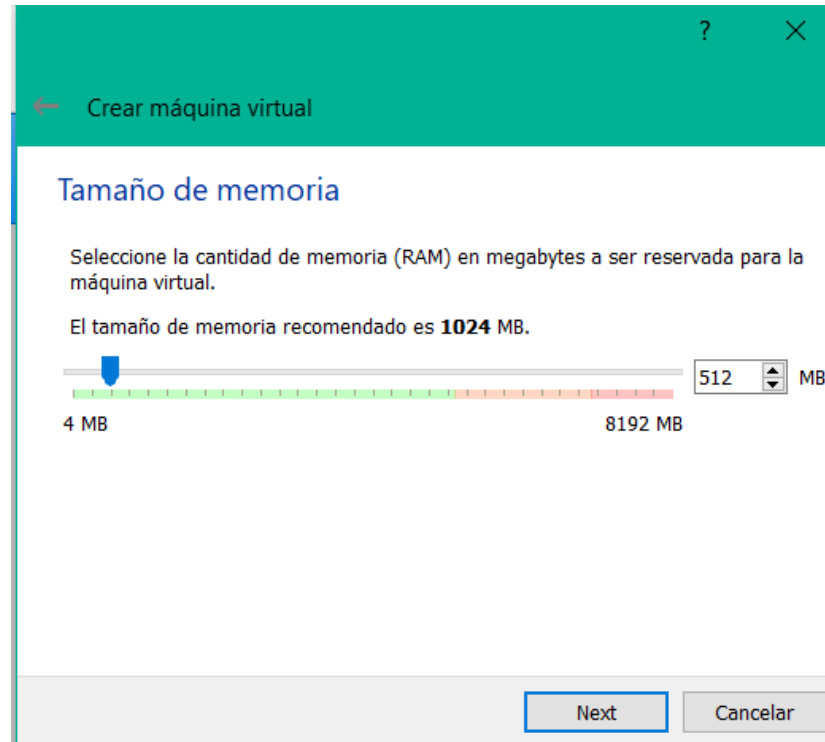
Se nombra a la máquina virtual y se asigna tipo de sistema operativo y versión:

- Nombre de la máquina virtual: M2 (M1 en el otro caso)
- Tipo de sistema operativo: Linux
- Versión: Ubuntu (64bits)

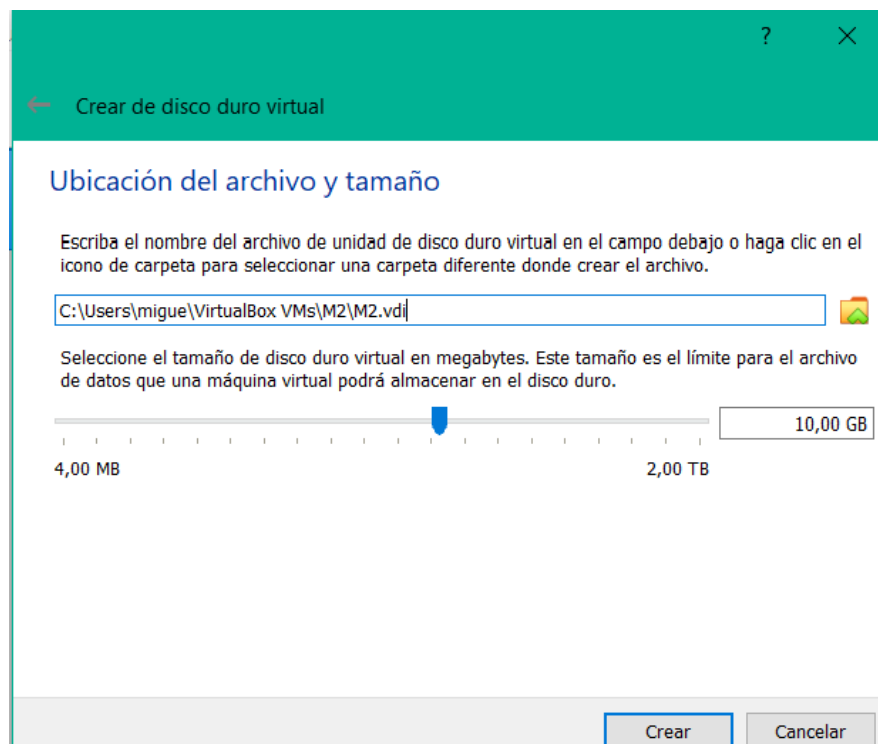


A continuación, vamos a asignar a cada máquina la memoria RAM y el espacio de disco duro mencionados anteriormente.
De esta manera seleccionamos ambos valores:

- Asignación memoria RAM:



- Asignación espacio disco duro:

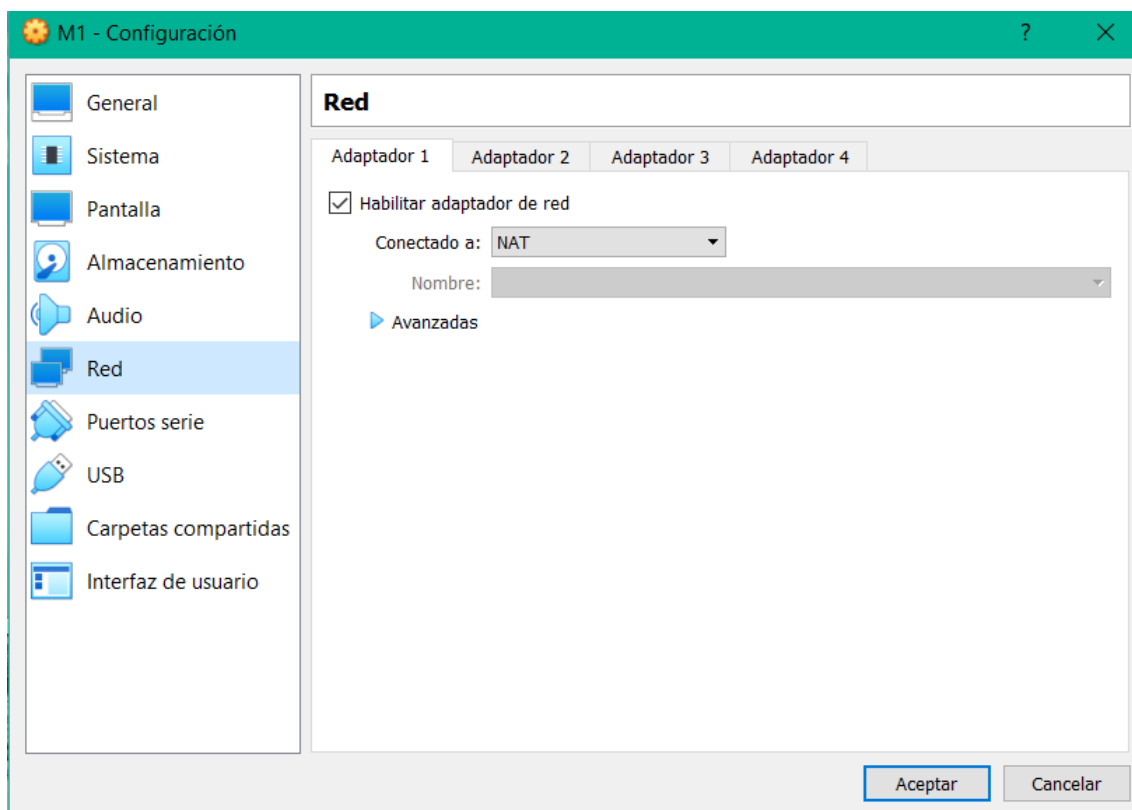


2. Configuración de los adaptadores de red de las máquinas virtuales

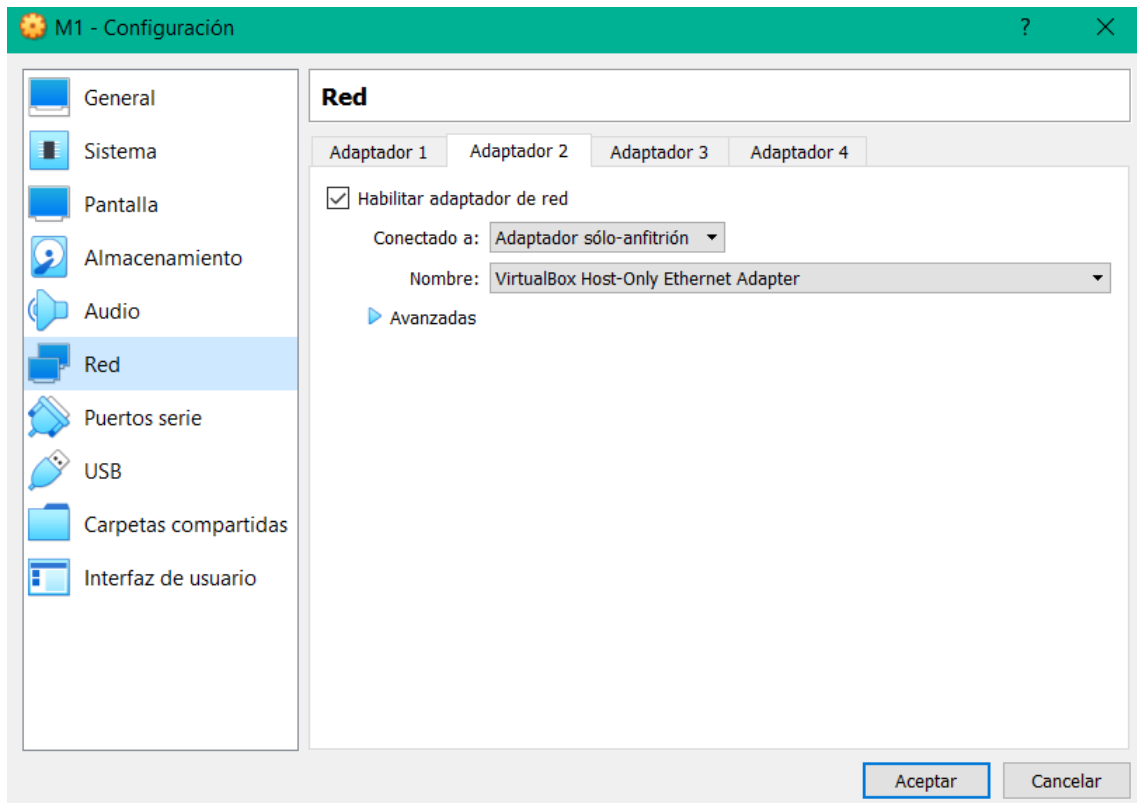
Para poder comunicar máquinas entre un mismo anfitrión y entre ellas, y éstas tener conexión a internet, es necesario añadir a cada máquina dos adaptadores: un adaptador de red en modo NAT y otro adaptador en solo-anfitrión para crear una red local entre las máquinas virtuales y el anfitrión.

A continuación, se va a configurar los adaptadores de red: uno en modo NAT (viene por defecto) y habilitamos el adaptador de host-only:

- Adaptador de red en modo NAT



- Adaptador de red en modo Host-Only

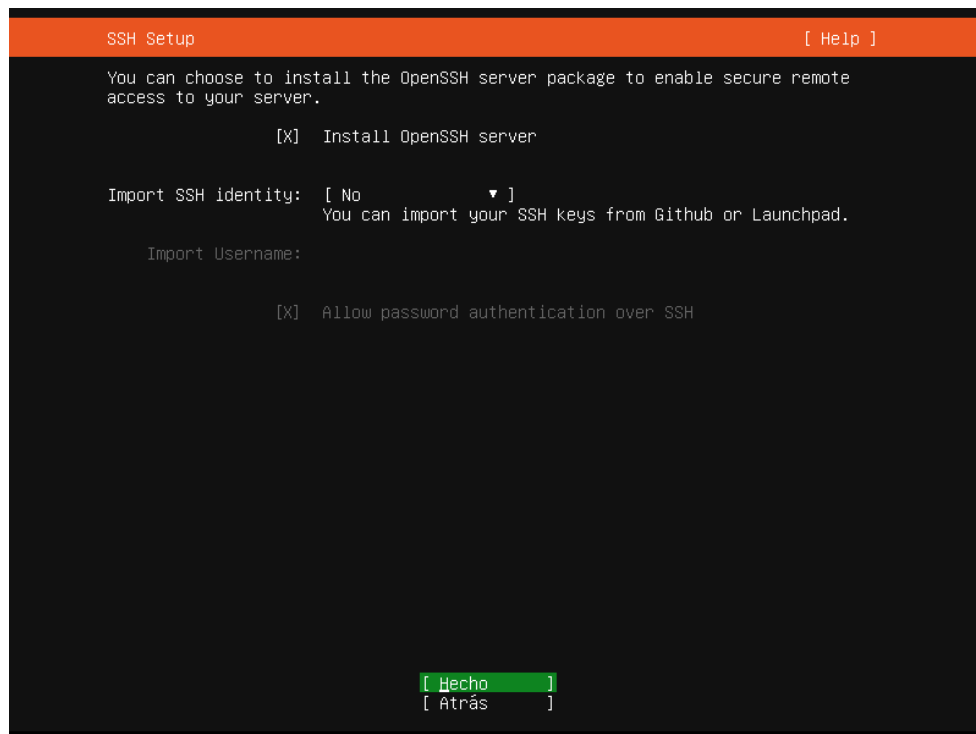


3. Instalación Ubuntu Server 18.04 y puesta a punto

En esta sección se va a completar la instalación de Ubuntu Server para las dos máquinas virtuales, de esta forma obtendremos las máquinas totalmente configuradas para los ejercicios que se piden para esta práctica.

Seleccionamos todas las opciones por defecto y las correspondiente a nuestra franja horaria, y finalmente creamos nuestro usuario. Antes de crear el usuario en cada máquina virtual seleccionamos que se instale el servicio SSH en el proceso de instalación.

- Selección servicio SSH



SSH Setup [Help]

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

[X] Install OpenSSH server

Import SSH identity: [No ▼]
You can import your SSH keys from Github or Launchpad.

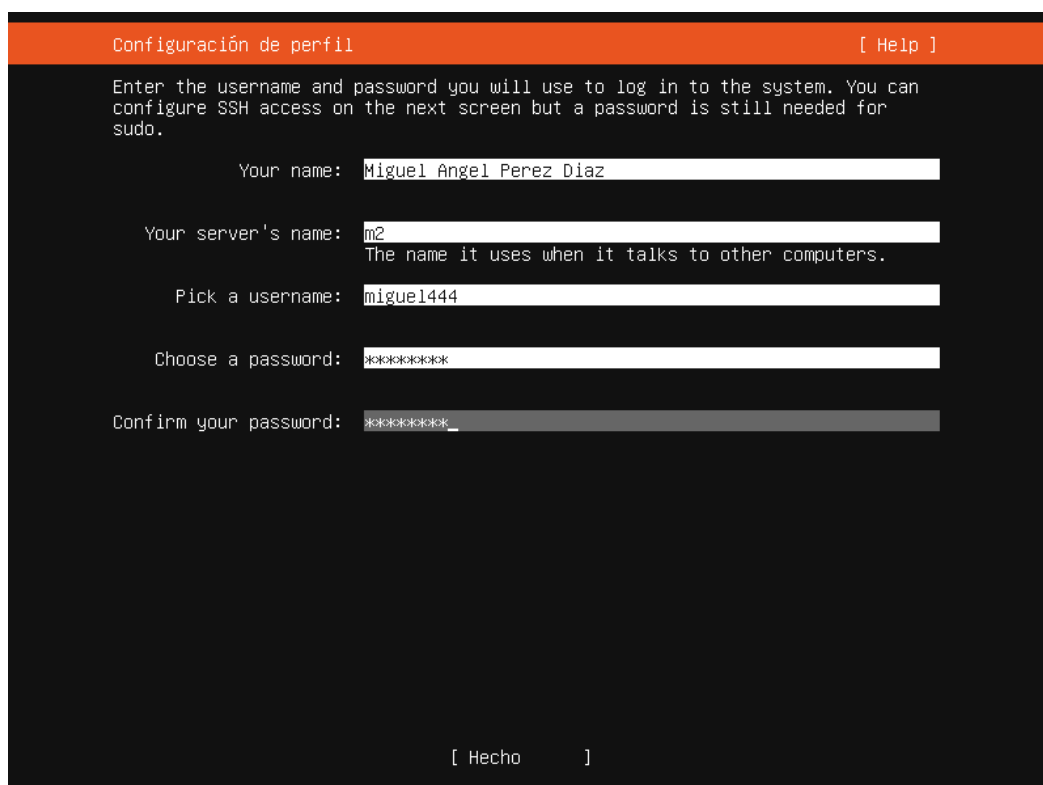
Import Username:

[X] Allow password authentication over SSH

[Hecho]
[Atrás]

- Creación de usuario

- Usuario: miguel444 (usuario de github)
- Contraseña: Swap1234



Configuración de perfil [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: Miguel Angel Perez Diaz

Your server's name: m2
The name it uses when it talks to other computers.

Pick a username: miguel444

Choose a password: ****

Confirm your password: ****_

[Hecho]

Una vez tenemos el sistema operativo en las dos máquinas vamos a instalar el servicio LAMP, ya que no nos ha aparecido la opción durante la instalación vamos a proceder a instalarlo manualmente de la siguiente forma:

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

```
miguel444@miguel444:~$ sudo apt-get install apache2 mysql-server mysql-client_
```

También, y de cara a las siguientes prácticas, podemos activar la cuenta de root. De esta forma, luego podremos acceder como superusuario, copiar contenidos con todos los permisos, etc, sin necesidad de andar usando sudo. Para eso, podéis ejecutar en todas las máquinas el siguiente comando:

- `sudo passwd root`

```
miguel444@m2:~$ sudo passwd root
[sudo] password for miguel444:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
miguel444@m2:~$ _
```

Finalmente, antes de ponerse manos a la obra y realizar las tareas correspondientes a esta práctica, vamos a comprobar que los servicios instalados anteriormente de LAMP y SSH se encuentran activos. De esta forma:

- Para comprobar el estado del servicio de Apache2

- `sudo service apache2 status`
- `apache2 -v`

```
miguel444@miguel444:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2019-12-03T15:55:03
miguel444@miguel444:~$ 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 Mon 2020-03-09 21:53:06 UTC; 2min 9s ago
   Main PID: 2608 (apache2)
   Tasks: 55 (limit: 503)
   CGroup: /system.slice/apache2.service
           └─2608 /usr/sbin/apache2 -k start
             └─2610 /usr/sbin/apache2 -k start
               └─2611 /usr/sbin/apache2 -k start

mar 09 21:53:06 miguel444 systemd[1]: Starting The Apache HTTP Server...
mar 09 21:53:06 miguel444 apachectl[2585]: AH00558: apache2: Could not reliably determine the server
mar 09 21:53:06 miguel444 systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

- Para comprobar el estado del servicio SSH

```
sudo systemctl status ssh
```

```
root@m1:/var/www/html# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-03-10 15:56:55 UTC; 1h 28min ago
     Main PID: 842 (sshd)
        Tasks: 1 (limit: 503)
       CGroup: /system.slice/ssh.service
               └─842 /usr/sbin/sshd -D

mar 10 15:56:55 m1 sshd[842]: Server listening on 0.0.0.0 port 22.
mar 10 15:56:55 m1 sshd[842]: Server listening on :: port 22.
mar 10 15:56:55 m1 systemd[1]: Started OpenBSD Secure Shell server.
mar 10 15:58:13 m1 sshd[1812]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tt
mar 10 15:58:15 m1 sshd[1812]: Failed password for root from 192.168.56.1 port 49213 ssh2
mar 10 15:58:24 m1 sshd[1812]: Failed password for root from 192.168.56.1 port 49213 ssh2
mar 10 15:58:25 m1 sshd[1812]: Connection closed by authenticating user root 192.168.56.1 port 49213
mar 10 15:58:25 m1 sshd[1812]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruse
mar 10 15:58:38 m1 sshd[1821]: Accepted password for miguel444 from 192.168.56.1 port 49214 ssh2
mar 10 15:58:38 m1 sshd[1821]: pam_unix(sshd:session): session opened for user miguel444 by (uid=0)
lines 1-18/18 (END)
```

Antes de finalizar el proceso de puesta a punto para nuestras máquinas virtuales es necesario configurar los adaptadores de red Host-Only. Para ello vamos a utilizar **netplan**, es importante configurar la dirección IP y puerta de enlace a cada máquina. Las configuraciones de red en versiones nuevas de Ubuntu se gestionan con netplan.

Para configurar nuestras máquinas es necesario acceder al fichero `/etc/netplan/50-cloud-init.yaml` y añadir lo siguiente relativo a la IP y la puerta de enlace que vamos a asignar a las máquinas:

- A M1 le vamos a asignar la IP 192.168.56.12

```
GNU nano 2.9.3 /etc/netplan/50-cloud-init.yaml

# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.56.12/24
      gateway4: 192.168.56.1
  version: 2
  renderer: networkd

[ Read 16 lines ]
^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  ^N Replace    ^U Uncut Text ^T To Spell  ^_ Go To Line  M-E Redo
```


- A M2 le vamos a asignar la IP 192.168.56.102

```
GNU nano 2.9.3 /etc/netplan/50-cloud-init.yaml

# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.56.102/24
      gateway4: 192.168.56.1
  version: 2

[ Read 15 lines ]
^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
```

A las dos máquinas hemos modificado los siguientes parámetros:

- DHCP: no (para que la IP de nuestro adaptador de red sea estática)
- ADDRESSES: IP ESTÁTICA DE LA MÁQUINA
- GATEWAY4: IP DE LA PUERTA DE ENLACE (192.168.56.1)

Una vez sobrescrito el fichero correspondiente procedemos a aplicar los cambios mediante el comando **netplan apply**.

4. Cuestiones a resolver

En esta práctica el objetivo es configurar las máquinas virtuales (al menos dos) para trabajar en prácticas posteriores, asegurando la conectividad entre dichas máquinas

Específicamente, hay que llevar a cabo las siguientes tareas:

- Acceder por SSH de una máquina a otra.
- Acceder mediante la herramienta curl desde una máquina a la otra.

Antes de realizar cualquiera de las tareas vamos a comprobar si una vez configurados los adaptadores de red obtenemos conectividad entre ambas máquinas. De este modo vamos a realizar **ping** de una máquina a otra y viceversa.

- Ping de M1 a M2

```
root@m1:/var/www/html# ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.435 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.469 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.495 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.615 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.439 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.513 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.508 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.447 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.452 ms
64 bytes from 192.168.56.102: icmp_seq=10 ttl=64 time=0.417 ms
```

- Ping de M2 a M1

```
miguel444@m2:~$ ping 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=0.484 ms
64 bytes from 192.168.56.12: icmp_seq=2 ttl=64 time=0.490 ms
64 bytes from 192.168.56.12: icmp_seq=3 ttl=64 time=0.427 ms
64 bytes from 192.168.56.12: icmp_seq=4 ttl=64 time=0.436 ms
64 bytes from 192.168.56.12: icmp_seq=5 ttl=64 time=0.451 ms
```

Una vez comprobada la conectividad vamos a proceder a crear el fichero *ejemplo.html* en la ruta *var/www/html* en M1:

```
GNU nano 2.9.3 ejemplo.html

<HTML>
<BODY>
  Web de ejemplo de miguel444 para SWAP
</BODY>
</HTML>

[ Read 5 lines ]
^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
```

De esta forma vamos a ejecutar el siguiente comando en M2: ***curl http://192.168.56.12/ejemplo.html*** para obtener y mostrar el fichero creado anteriormente en M1. Obteniendo así la siguiente respuesta:

```
miguel444@m2:~$ curl http://192.168.56.12/ejemplo.html
<HTML>
<BODY>
    Web de ejemplo de miguel444 para SWAP
</BODY>
</HTML>
miguel444@m2:~$ _
```

Como última tarea vamos a proceder a realizar SSH desde la máquina M2 a M1, para ello debemos ejecutar lo siguiente: ***ssh 192.168.56.12*** (siendo ésta la dirección IP de M1).

Una vez establecida la conexión nos pedirá confirmación y la contraseña de la dirección destino, en este caso la contraseña de M1 que como sabemos sería Swap1234.

```
miguel444@m2:~$ ssh 192.168.56.12
The authenticity of host '192.168.56.12 (192.168.56.12)' can't be established.
ECDSA key fingerprint is SHA256:jciONzH/CrBpd2dWCrvAufK1q7WRYEEhNkGjWq6MAjs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.12' (ECDSA) to the list of known hosts.
miguel444@192.168.56.12's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information as of Tue Mar 10 17:27:00 UTC 2020

System load:  0.08               Processes:            94
Usage of /:   33.1% of 9.78GB    Users logged in:     1
Memory usage: 65%               IP address for enp0s3: 10.0.2.15
Swap usage:   1%                IP address for enp0s8: 192.168.56.12

 * 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/

Pueden actualizarse 18 paquetes.
4 actualizaciones son de seguridad.

Last login: Tue Mar 10 15:58:39 2020 from 192.168.56.1
miguel444@m1:~$
```

De esta forma vamos a ejecutar lo anterior, pero en este caso vamos a realizar la conexión de M1 a M2. Al igual que anteriormente ejecutamos el siguiente comando en M1: **curl http://192.168.56.102/ejemplo.html** para obtener y mostrar el fichero creado anteriormente en M2. Obteniendo así la siguiente respuesta:

```
miguel444@m1:~$ curl http://192.168.56.102/ejemplo2.html
<HTML>
<BODY>
    Web de ejemplo de miguel444 para SWAP
</BODY>
</HTML>

miguel444@m1:~$ _
```

Como última tarea vamos a proceder a realizar SSH desde la máquina M1 a M2, para ello debemos ejecutar lo siguiente: **ssh 192.168.56.102** (siendo ésta la dirección IP de M2).

Una vez establecida la conexión nos pedirá confirmación y la contraseña de la dirección destino, en este caso la contraseña de M2 que como sabemos sería Swap1234.

```
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:eK91k4unzUuxjxt03gvVMjWI821bpewHlrWmEu5DdyA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
miguel444@192.168.56.102's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

System information as of Wed Mar 11 20:23:21 UTC 2020

System load:  0.0               Processes:    92
Usage of /:   33.3% of 9.78GB   Users logged in: 1
Memory usage: 65%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.102

* 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/

Pueden actualizarse 14 paquetes.
0 actualizaciones son de seguridad.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection
or proxy settings

Last login: Wed Mar 11 20:13:59 2020
miguel444@m2:~$ _
```