

## **Práctica 1:**

### **Presentación de las prácticas y preparación de las herramientas**

#### ***Índice:***

- 1) Introducción*
- 2) Instalación Ubuntu Server 18.04.4*
- 3) Configuración de los adaptadores de red*
- 4) Conexión ssh*
- 5) Conexión curl*
- 6) Pings*

## *1) Introducción*

*Esta práctica se llevará a cabo bajo Linux usando Ubuntu Server. Usaremos máquinas virtuales para disponer en cada ordenador de los recursos necesarios para realizar las prácticas. En este caso, VirtualBox para virtualizar varias máquinas Linux que configuraremos como sigue para esta práctica.*

*Lo primero que deberemos hacer es descargar Oracle Virtual Box para nuestro sistema desde su sitio web oficial: <https://www.virtualbox.org/wiki/Downloads>*

*También debemos descargar la imagen de Ubuntu Server 18.04.4 desde: <https://ubuntu.com/download/server>*

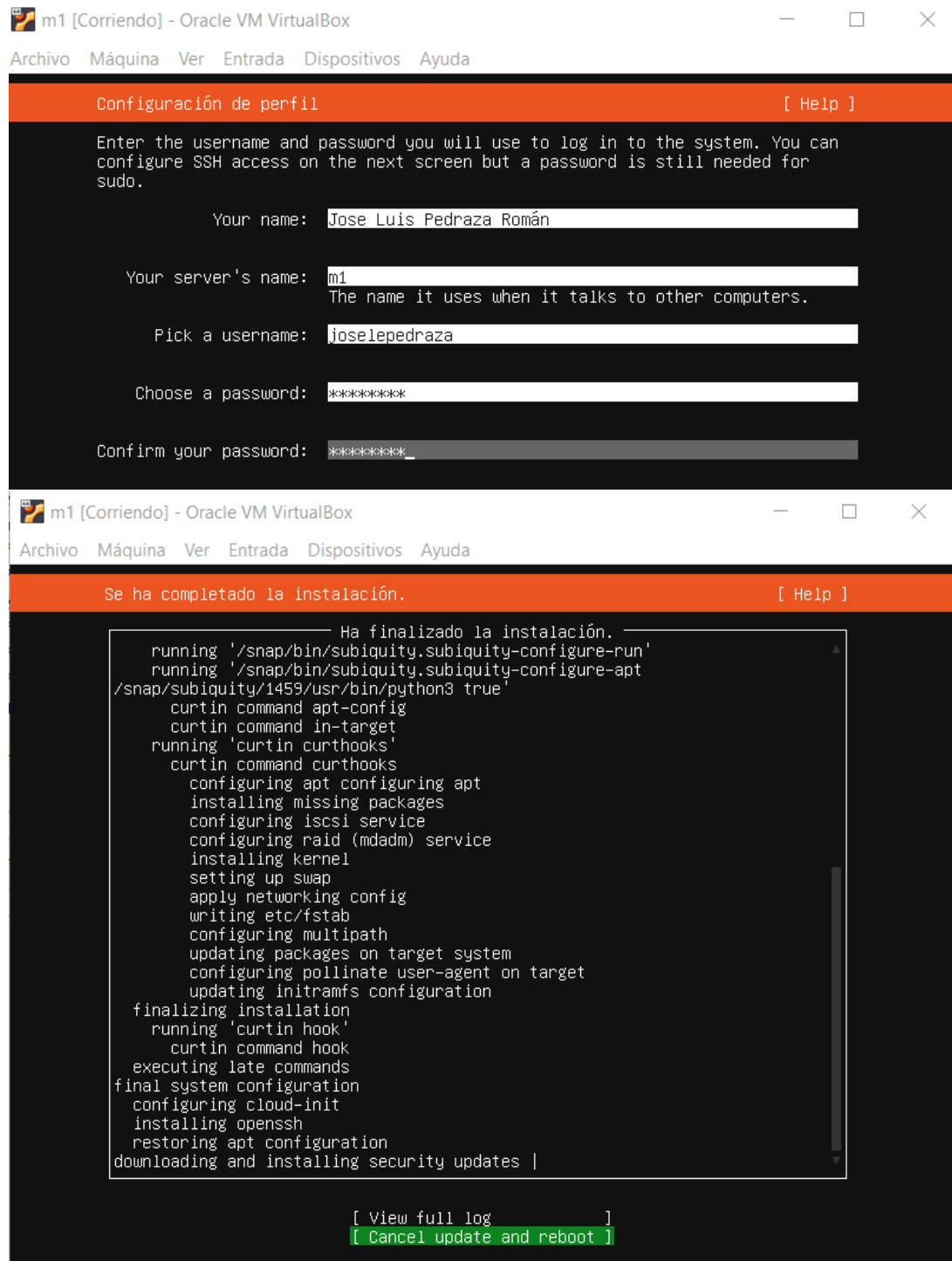
*Debemos crear dos máquinas virtuales M1 y M2 con al menos 512 MB de RAM y 10 GB de disco duro. En cada máquina deben estar disponibles los siguientes servicios: conexión SSH, LAMP server y conexión CURL.*

*Algunos procesos de instalación y configuración se mostrarán una única vez en una sola máquina, ya que para la otra será exactamente igual (se indicará donde sea necesario).*

## 2) Instalación Ubuntu Server 18.04.4

En mi caso, a cada máquina le he asignado 1024 MB de RAM y 10 GB de disco virtual.

Instalación en la máquina m1 (igual para m2):



```

[ OK ] Started Execute cloud user/final scripts.
[ OK ] Reached target Cloud-init target.

Ubuntu 18.04.4 LTS m1 tty1

m1 login: joselepedraza
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 Thu Mar  5 17:41:49 UTC 2020

System load:  0.15           Processes:            87
Usage of /:   38.8% of 9.78GB Users logged in:        0
Memory usage: 15%           IP address for enp0s3: 10.0.2.15
Swap usage:   0%

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

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

joselepedraza@m1:~$

```

*Para configurar LAMP server he instalado de forma manual Apache, PHP y MySQL ejecutando el siguiente comando:*

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

*También podemos seguir el siguiente tutorial: <https://www.unixmen.com/how-to-install-lamp-stack-on-ubuntu-16-04/>*

*Comprobamos la versión y si está en ejecución con los siguientes comandos:*

```
Service apache2 status
```

```
apache2 -v
```

```

Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/ap
ache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/syste
md/system/apache-htcacheclean.service.
Configurando mysql-client (5.7.29-0ubuntu0.18.04.1) ...
Configurando mysql-server-5.7 (5.7.29-0ubuntu0.18.04.1) ...
update-alternatives: utilizando /etc/mysql/mysql.cnf para proveer /etc/mysql/my.cnf (my.cnf) en mod
o automático
Renaming removed key_buffer and myisam-recover options (if present)
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysq
l.service.
Configurando mysql-server (5.7.29-0ubuntu0.18.04.1) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
Procesando disparadores para systemd (237-3ubuntu10.38) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para ufw (0.36-0ubuntu0.18.04.1) ...
Procesando disparadores para ureadahead (0.100.0-21) ...
root@m1:/home/joselepedraza# apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2019-12-03T15:55:03
root@m1:/home/joselepedraza# 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 Thu 2020-03-05 18:41:36 UTC; 3min 49s ago
  Main PID: 1851 (apache2)
  Tasks: 55 (limit: 1108)
  CGroup: /system.slice/apache2.service
          └─1851 /usr/sbin/apache2 -k start
            └─1853 /usr/sbin/apache2 -k start
              └─1854 /usr/sbin/apache2 -k start

mar 05 18:41:36 m1 systemd[1]: Starting The Apache HTTP Server...
mar 05 18:41:36 m1 apachectl[1829]: AH00558: apache2: Could not reliably determine the server's full
mar 05 18:41:36 m1 systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)

```

*Debemos hacer lo mismo para la máquina m2 antes de configurar los adaptadores de red (aunque podríamos haberlos configurado primero).*

### 3) Configuración de los adaptadores de red

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 modo solo-anfitrión para crear una red local entre las máquinas virtuales y el anfitrión.

En la interfaz de VirtualBox nos vamos a “archivo/administrador de red de anfitrión”. Configuramos la red sólo-anfitrión con una dirección 192.168.56.1 con máscara de red /24 y con un servidor DHCP 192.168.56.100 /24 que alcanza direcciones desde la 101 a 254.

-Configuración del adaptador:

Administrador de red anfitrión

Red

Crear Eliminar Propiedades

| Nombre                                | Dirección/máscara IPv4 | Dirección/máscara IPv6 | Servidor DHCP                                 |
|---------------------------------------|------------------------|------------------------|---|
| VirtualBox Host-Only Ethernet Adapter | 192.168.56.1/24        |                        | <input checked="" type="checkbox"/> Habilitar |

Adaptador Servidor DHCP

☐ Configurar adaptador automáticamente

☒ Configurar adaptador manualmente

Dirección IPv4: 192.168.56.1

Máscara de red IPv4: 255.255.255.0

Dirección IPv6: fe80::dd68:26d3:3b39:a869

Longitud de máscara de red IPv6: 64

Restaurar Aplicar Cerrar

-Configuración DHCP:

Administrador de red anfitrión

Red

Crear Eliminar Propiedades

| Nombre                                | Dirección/máscara IPv4 | Dirección/máscara IPv6 | Servidor DHCP                                 |
|---------------------------------------|------------------------|------------------------|---|
| VirtualBox Host-Only Ethernet Adapter | 192.168.56.1/24        |                        | <input checked="" type="checkbox"/> Habilitar |

Adaptador Servidor DHCP

☒ Habilitar servidor

Dirección del servidor: 192.168.56.100

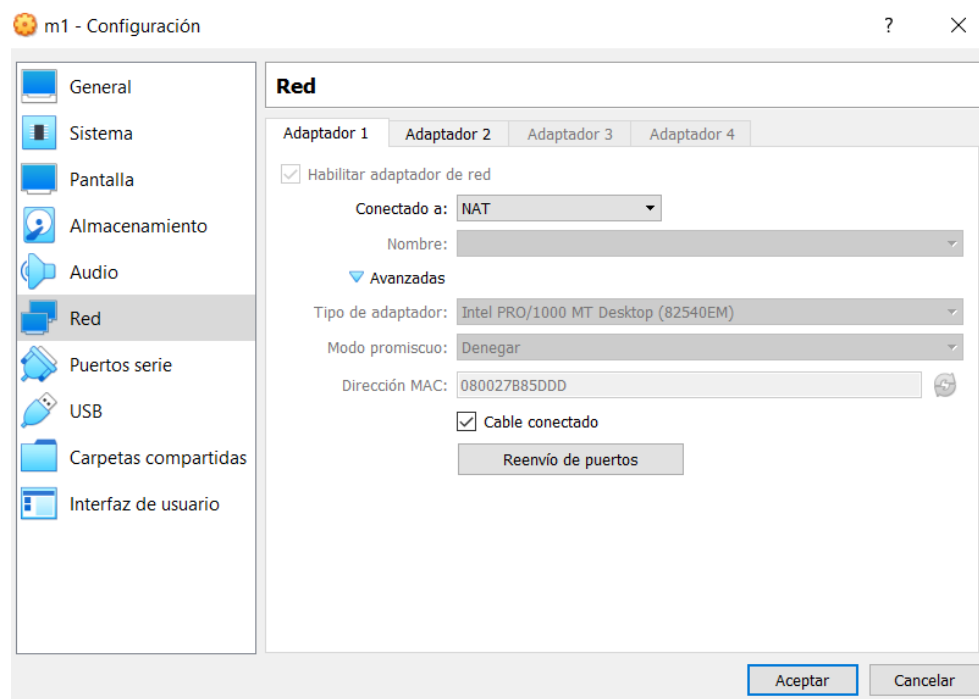
Máscara del servidor: 255.255.255.0

Límite inferior de direcciones: 192.168.56.101

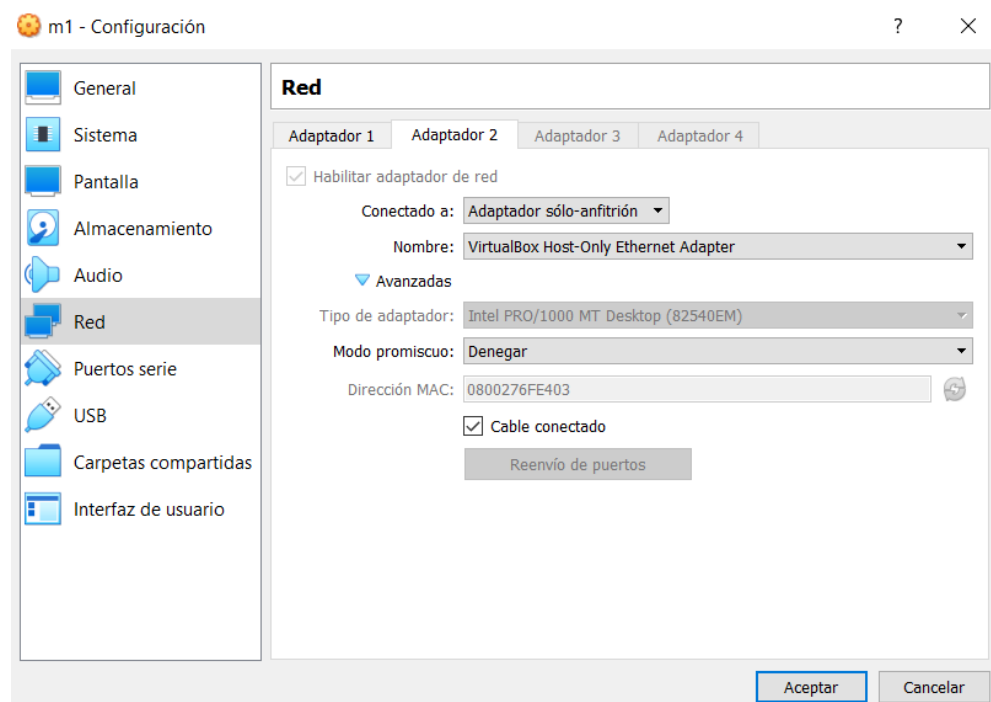
Límite superior de direcciones: 192.168.56.254

Restaurar Aplicar Cerrar

Con las máquinas apagadas, para configurar los adaptadores de red en modo NAT nos vamos a la configuración de la máquina m1, “configuración/red/adaptador1” y seleccionamos “Conectado a NAT”. En la configuración de red de la máquina m1 se ha habilitado el adaptador2 de red de tipo solo-anfitrión y se ha enlazado con la red creada en el paso anterior:



Ahora seleccionamos “adaptador2” y seleccionamos “Conectado a adaptador sólo-anfitrión”:



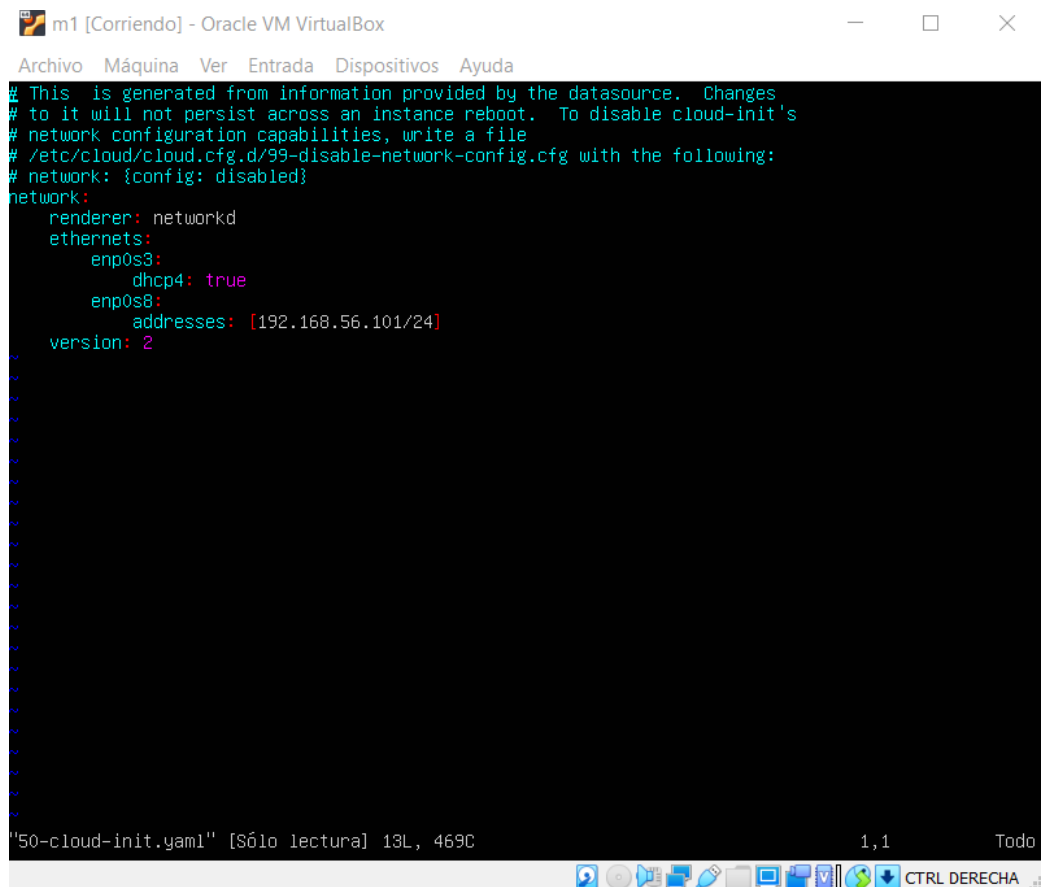
Este mismo proceso lo repetimos para la máquina m2.

*Iniciamos las máquinas virtuales m1 y m2 para configurar la dirección IP y puerta de enlace a cada máquina. Estas configuraciones de red en versiones nuevas de Ubuntu se gestionan con netplan.*

*Debemos acceder al directorio /etc/netplan y antes de modificar el archivo .yaml le crearemos una copia de seguridad .yaml.old como sigue:*

*cp 50-cloud-init.yaml 50-cloud-init.yaml.old*

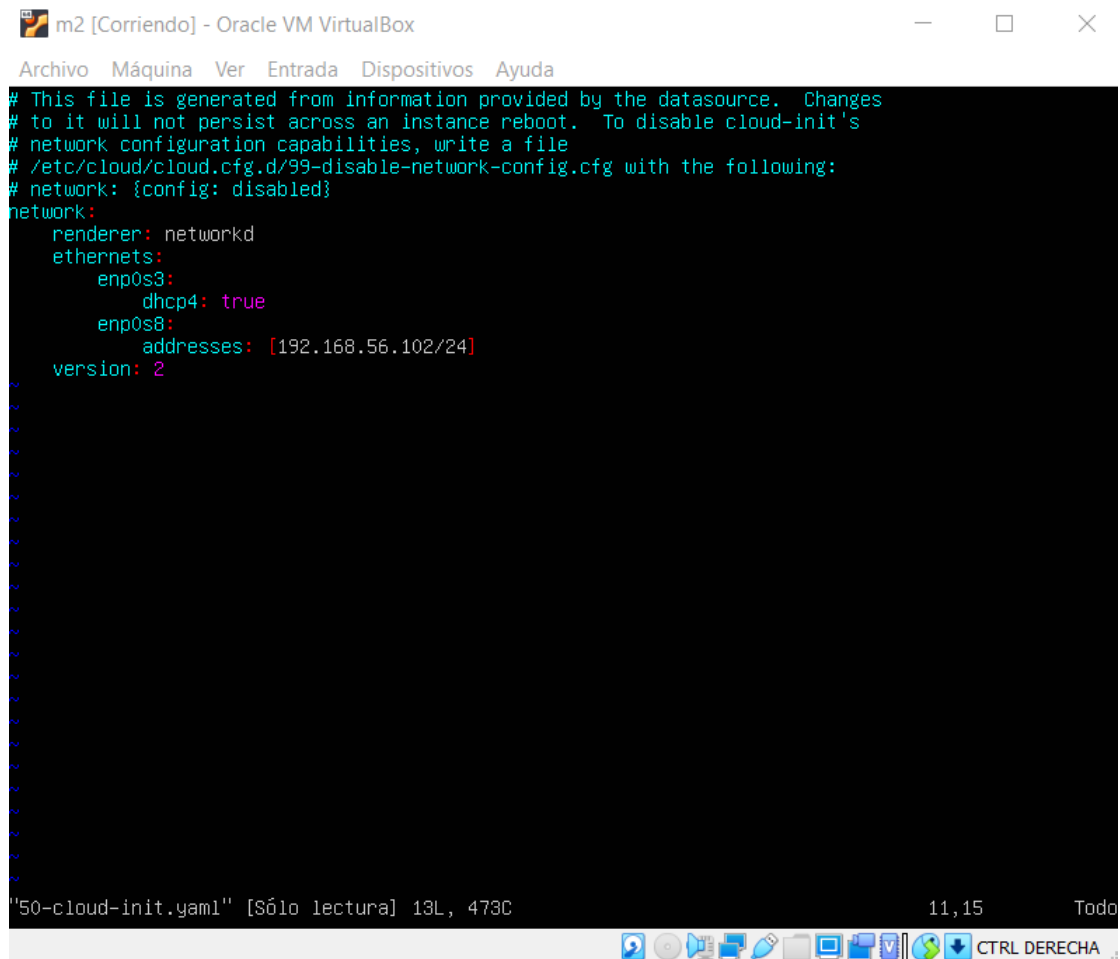
*Una vez hecho esto modificamos con vim (para editar “i” y para salir y guardar “:wq”) dicho archivo con la siguiente configuración para m1 (siempre en modo root), añadiendo una nueva red llamada enp0s8 (usada para la conexión con el anfitrión):*



```
# This 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:
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.56.101/24]
  version: 2
```



*Hacemos lo mismo para la máquina m2 pero con la siguiente configuración (le asignamos otra dirección IP):*



The screenshot shows a terminal window titled "m2 [Corriendo] - Oracle VM VirtualBox". The terminal displays a network configuration file for cloud-init. The configuration includes a comment about the file being generated from datasource information, instructions to write a file to /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg, and a network configuration block for the enp0s8 interface. The interface is configured with DHCP4 set to true and a static IP address of 192.168.56.102/24. The terminal also shows the file path "50-cloud-init.yaml" and the cursor position at line 13, column 4730.

```
# 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:
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.56.102/24]
  version: 2
```

*Es muy importante aplicar esta configuración correctamente, para ello, una vez guardado el archivo .yaml deberemos ejecutar los siguientes comandos (siempre en modo root):*

*netplan generate*

*netplan apply*

*En nuestro caso, las direcciones IP de nuestras máquinas serán:*

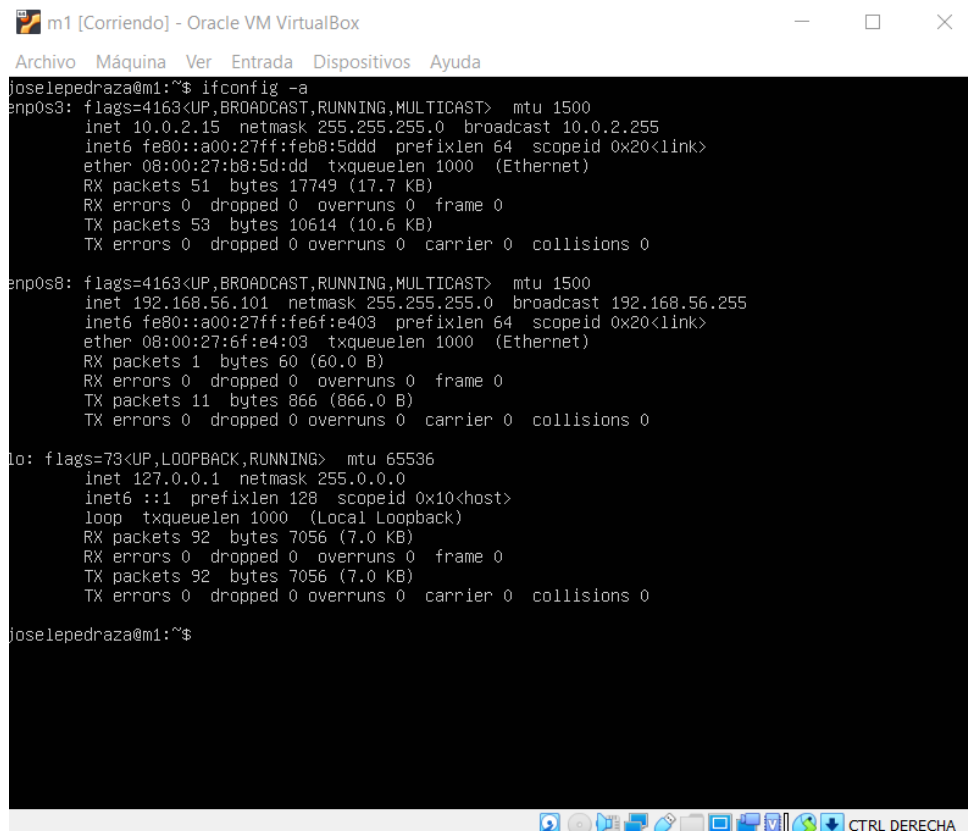
*-m1 = 192.168.56.101*

*-m2 = 192.168.56.102*

*A continuación mostramos la configuración de cada máquina ejecutando el comando siguiente:*

*ifconfig -a*

-m1:



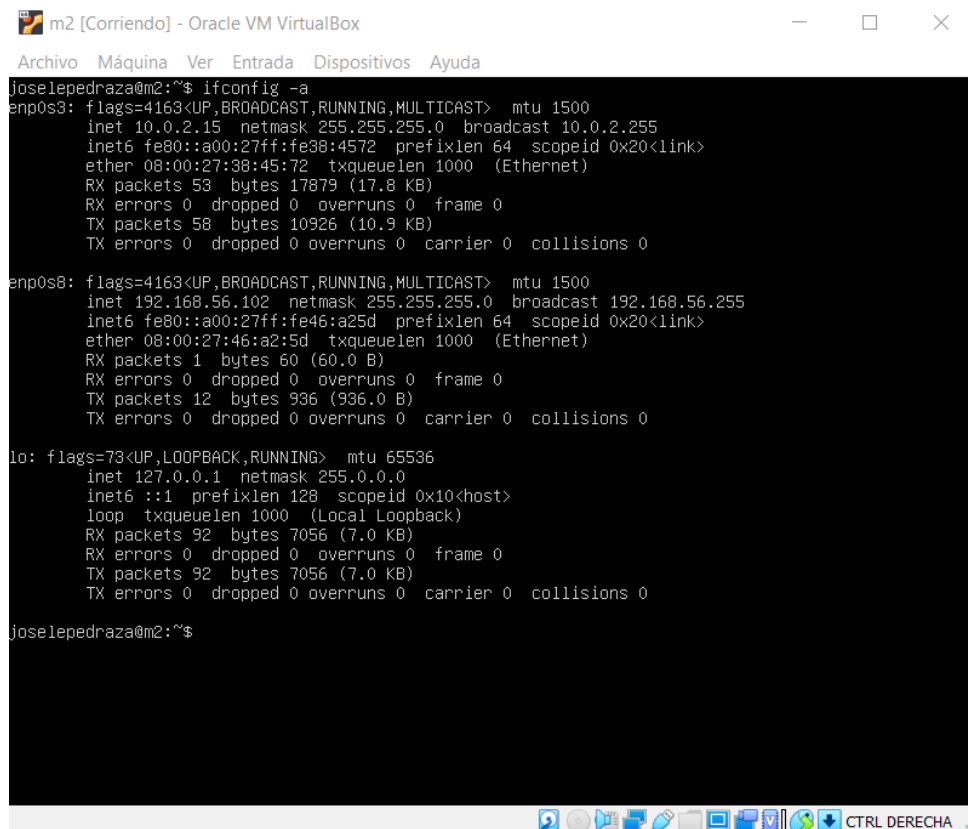
```
joselepedraza@m1:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:feb8:5ddd prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:b8:5d:dd txqueuelen 1000 (Ethernet)
    RX packets 51 bytes 17749 (17.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 53 bytes 10614 (10.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe6f:e403 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:6f:e4:03 txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 60 (60.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 866 (866.0 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 92 bytes 7056 (7.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 92 bytes 7056 (7.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

joselepedraza@m1:~$
```

-m2:



```
joselepedraza@m2:~$ ifconfig -a
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:fe38:4572 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:38:45:72 txqueuelen 1000 (Ethernet)
    RX packets 53 bytes 17879 (17.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 58 bytes 10926 (10.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe46:a25d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:46:a2:5d txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 60 (60.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

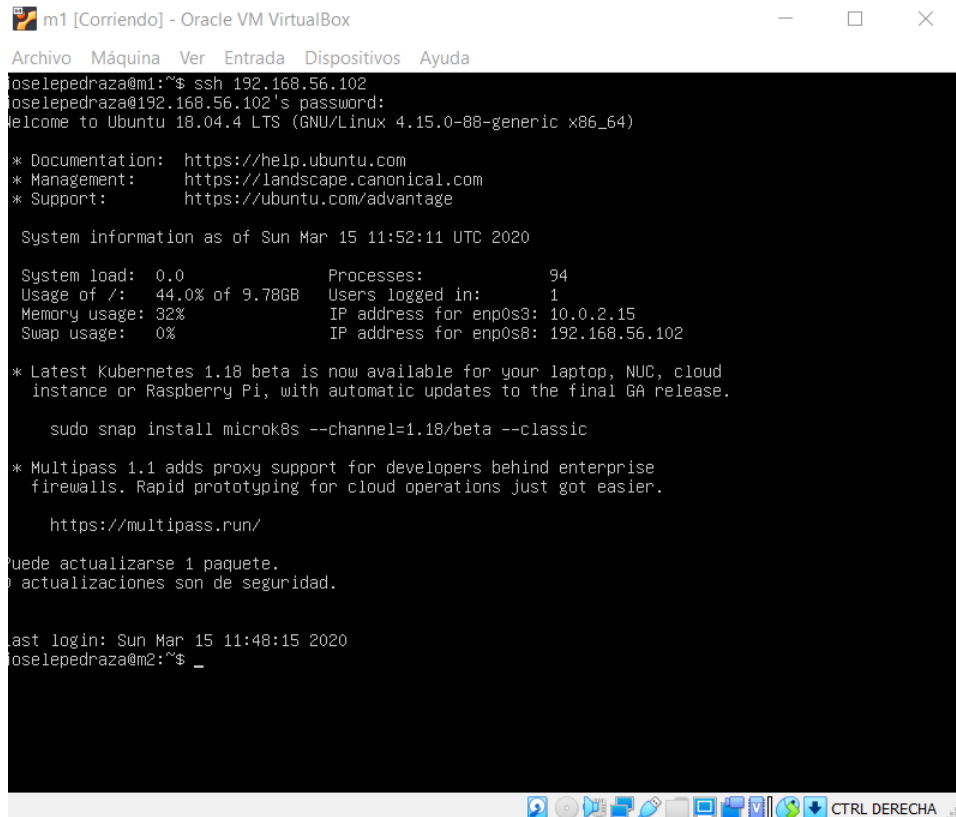
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 92 bytes 7056 (7.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 92 bytes 7056 (7.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

joselepedraza@m2:~$
```

#### 4) Conexión ssh:

*Comprobamos que podemos establecer la conexión ssh entre máquinas.*

*-m1: ssh 192.168.56.102*



```
m1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
joselepedraza@m1:~$ ssh 192.168.56.102
joselepedraza@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 Sun Mar 15 11:52:11 UTC 2020

System load:  0.0          Processes:      94
Usage of /:   44.0% of 9.78GB  Users logged in: 1
Memory usage: 32%          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

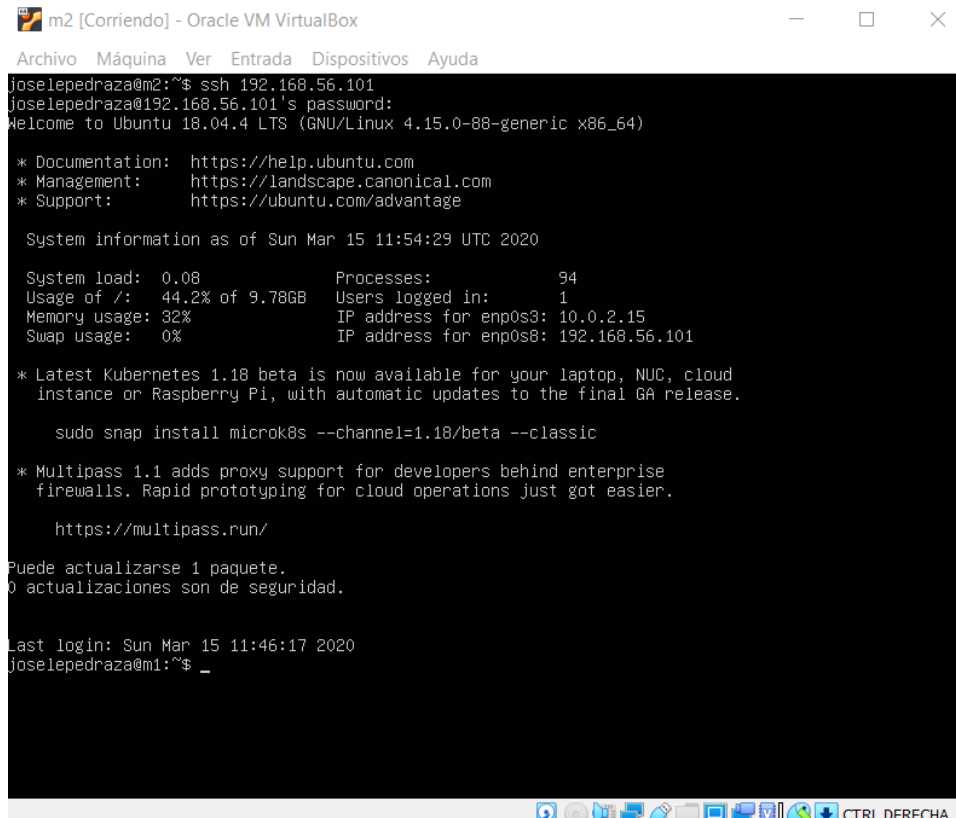
 * Multipass 1.1 adds proxy support for developers behind enterprise
   firewalls. Rapid prototyping for cloud operations just got easier.

   https://multipass.run/

Puede actualizarse 1 paquete.
0 actualizaciones son de seguridad.

Last login: Sun Mar 15 11:48:15 2020
joselepedraza@m2:~$ _
```

*-m2: ssh 192.168.56.101*



```
m2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
joselepedraza@m2:~$ ssh 192.168.56.101
joselepedraza@192.168.56.101'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 Sun Mar 15 11:54:29 UTC 2020

System load:  0.08         Processes:      94
Usage of /:   44.2% of 9.78GB  Users logged in: 1
Memory usage: 32%          IP address for enp0s3: 10.0.2.15
Swap usage:   0%            IP address for enp0s8: 192.168.56.101

 * 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.

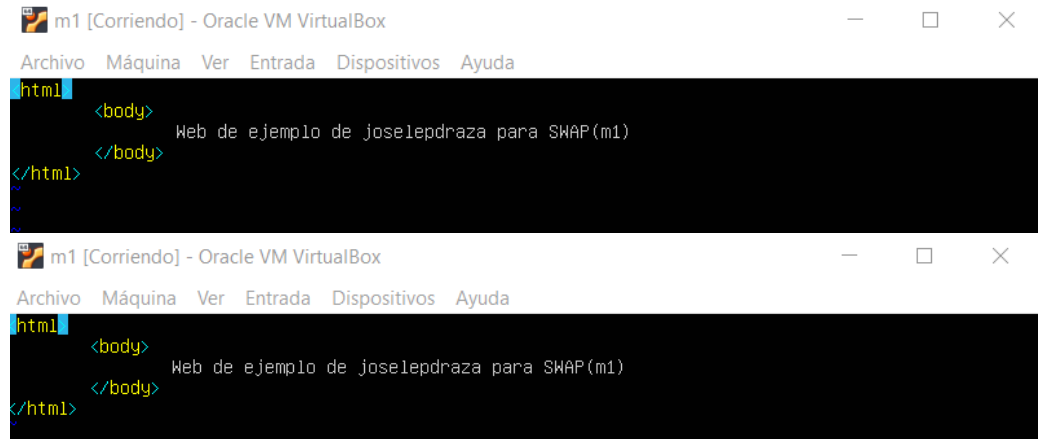
   https://multipass.run/

Puede actualizarse 1 paquete.
0 actualizaciones son de seguridad.

Last login: Sun Mar 15 11:46:17 2020
joselepedraza@m1:~$ _
```

## 5) Conexión curl:

Debemos crear un archivo llamado `ejemplo.html` en el directorio `/var/www/html/` de las dos máquinas `m1` y `m2`, y comprobar que podemos acceder a él desde la otra máquina y desde el anfitrión, indicando la dirección IP de nuestra máquina servidora virtualizada.



The first screenshot shows a terminal window for machine `m1` with the command `cat > ejemplo.html` and the following content:

```
<body>
    Web de ejemplo de joselepdraza para SWAP(m1)
</body>
</html>
```

The second screenshot shows a similar terminal window for machine `m2` with the same content.

Comprobamos que funciona `curl` entre máquinas con el siguiente comando:  
`curl http://direccionIPdelservidor/ejemplo.html`



The first screenshot shows a terminal window for machine `m1` with the command `curl http://192.168.56.102/ejemplo.html` and the output:

```
<html>
  <body>
    Web de ejemplo de joselepdraza para SWAP(m2)
  </body>
</html>
```

The second screenshot shows a terminal window for machine `m2` with the command `curl http://192.168.56.101/ejemplo.html` and the output:

```
<html>
  <body>
    Web de ejemplo de joselepdraza para SWAP(m1)
  </body>
</html>
```

Comprobamos desde un navegador de la máquina anfitrión que se puede acceder a dicho contenido de las máquinas servidoras:



Web de ejemplo de joselepedraza para SWAP(m1)



Web de ejemplo de joselepedraza para SWAP(m2)

## 6) Pings:

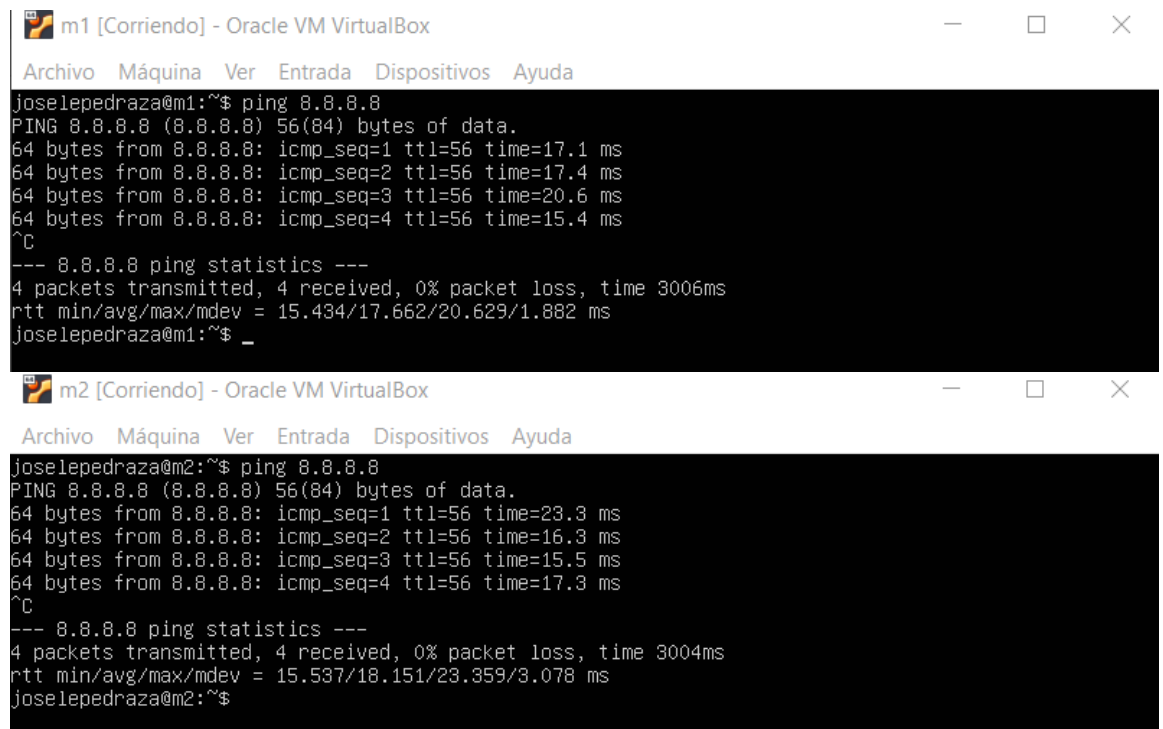
*Ping de m1 a m2:*

```
m1 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
joselepedraza@m1:~$ 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=0.583 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.601 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.801 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.825 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.834 ms
^C
--- 192.168.56.102 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4089ms
rtt min/avg/max/mdev = 0.583/0.728/0.834/0.117 ms
joselepedraza@m1:~$
```

*Ping de m2 a m1:*

```
m2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
joselepedraza@m2:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.684 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.896 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.789 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.631 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.818 ms
^C
--- 192.168.56.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4048ms
rtt min/avg/max/mdev = 0.631/0.763/0.896/0.099 ms
joselepedraza@m2:~$ _
```

### *Ping a Google:*



The image shows two screenshots of Oracle VM VirtualBox terminal windows. The first window, titled 'm1 [Corriendo] - Oracle VM VirtualBox', shows a terminal session where the user 'joselepedraza@m1' runs the command 'ping 8.8.8.8'. The output shows four successful ping requests with times ranging from 15.4 ms to 20.6 ms, and a summary indicating 0% packet loss and an average time of 17.662 ms. The second window, titled 'm2 [Corriendo] - Oracle VM VirtualBox', shows a similar terminal session for user 'joselepedraza@m2'. The output shows four successful ping requests with times ranging from 15.5 ms to 23.3 ms, and a summary indicating 0% packet loss and an average time of 18.151 ms.

```
joselepedraza@m1:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=17.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=17.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=20.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=15.4 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 15.434/17.662/20.629/1.882 ms
joselepedraza@m1:~$ _

joselepedraza@m2:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=23.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=16.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=15.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=17.3 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 15.537/18.151/23.359/3.078 ms
joselepedraza@m2:~$
```

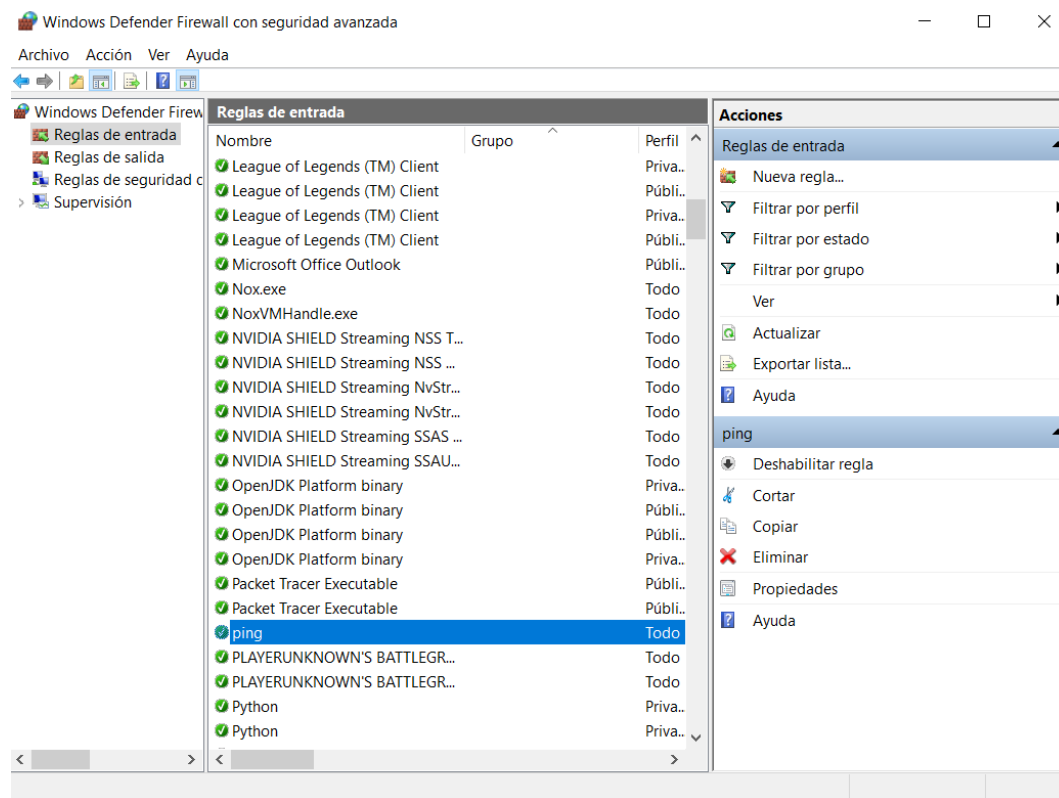
### *Ping a anfitrión:*

*Debemos asegurarnos previamente de que, tanto los firewall de las máquinas virtuales como de la máquina anfitrión estén desactivados.*

*Para ello, en las máquinas virtuales ejecutamos el siguiente comando:*

*ufw status*

*Y en la máquina anfitrión, en mi caso con Windows, deberemos acceder a la configuración de firewall y añadirle una excepción (o regla de entrada) tal que así:*



*Una vez hecho esto, probamos a hacer ping desde m1 y m2 a la dirección de la máquina anfitrión, 192.168.56.1:*

