

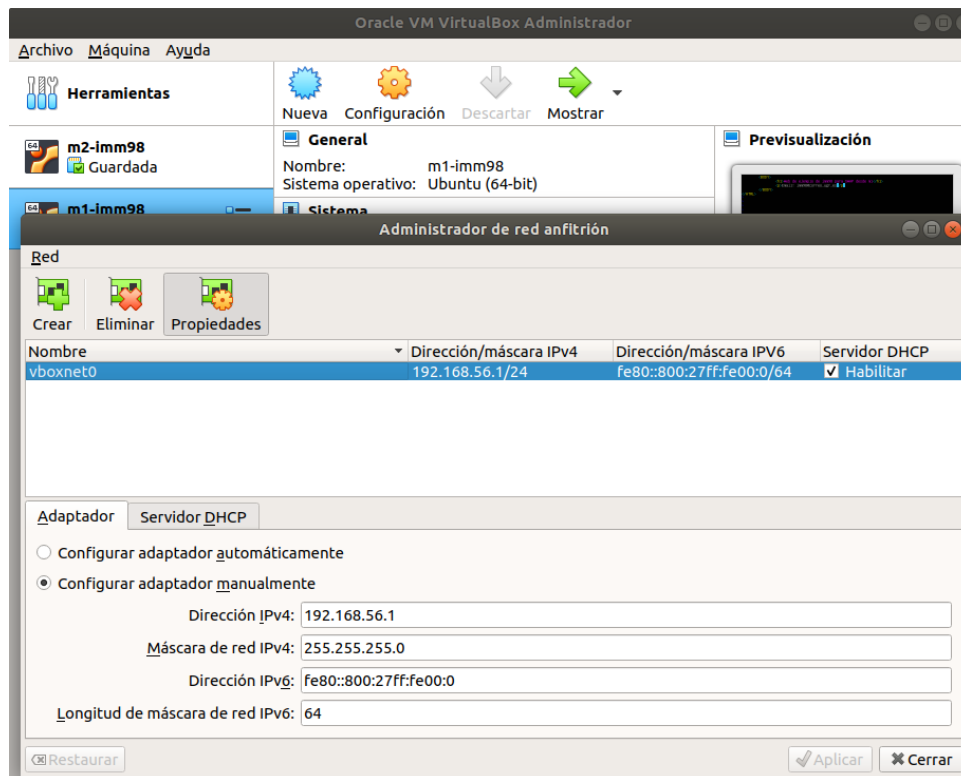
PRÁCTICA 1: PREPARACIÓN DE LAS HERRAMIENTAS

Iñaki Melguizo Marcos

Para la realización de la práctica he utilizado el software de virtualización VirtualBox y he creado dos máquinas virtuales m1-imm98 y m2-imm98 con idénticas características y con UbuntuServer18.04.5

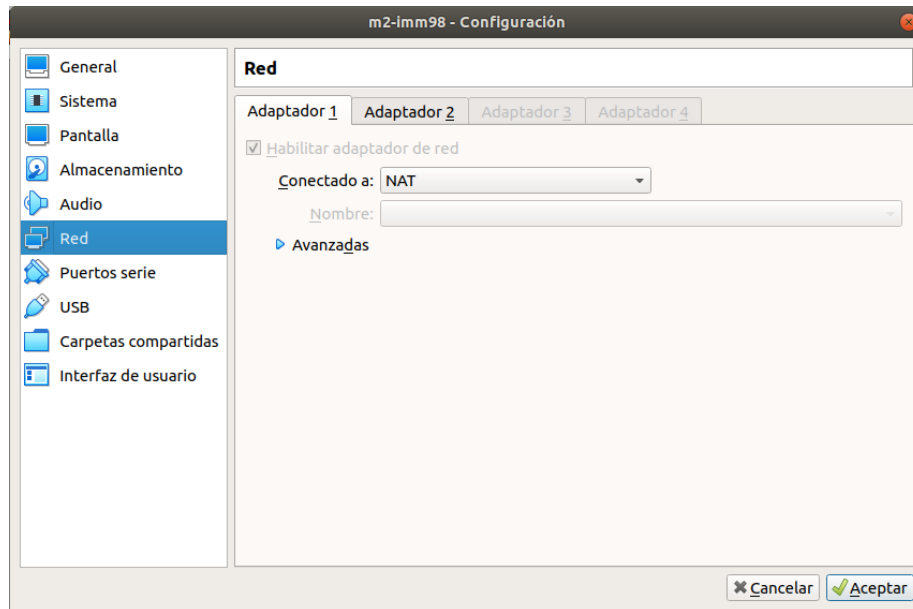


Para la existencia de comunicación y visibilidad entre ambas máquinas virtuales he creado una red sólo anfitrión con la siguiente configuración:

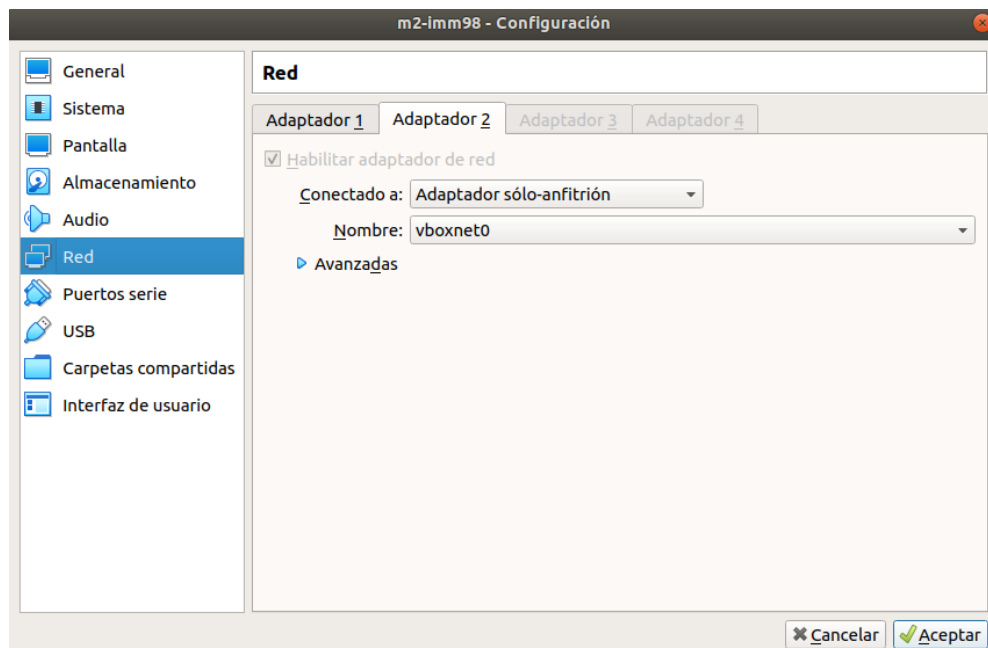


Por lo tanto tendremos dos adaptadores de red:

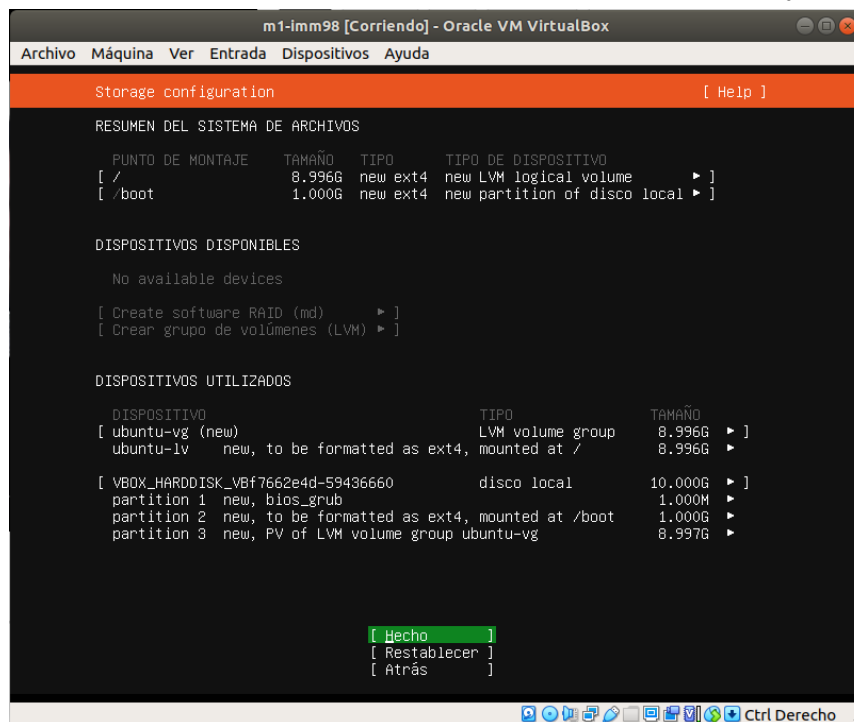
- Un adaptador de tipo **NAT** que permite comunicarnos con todas las máquinas virtuales a Internet.



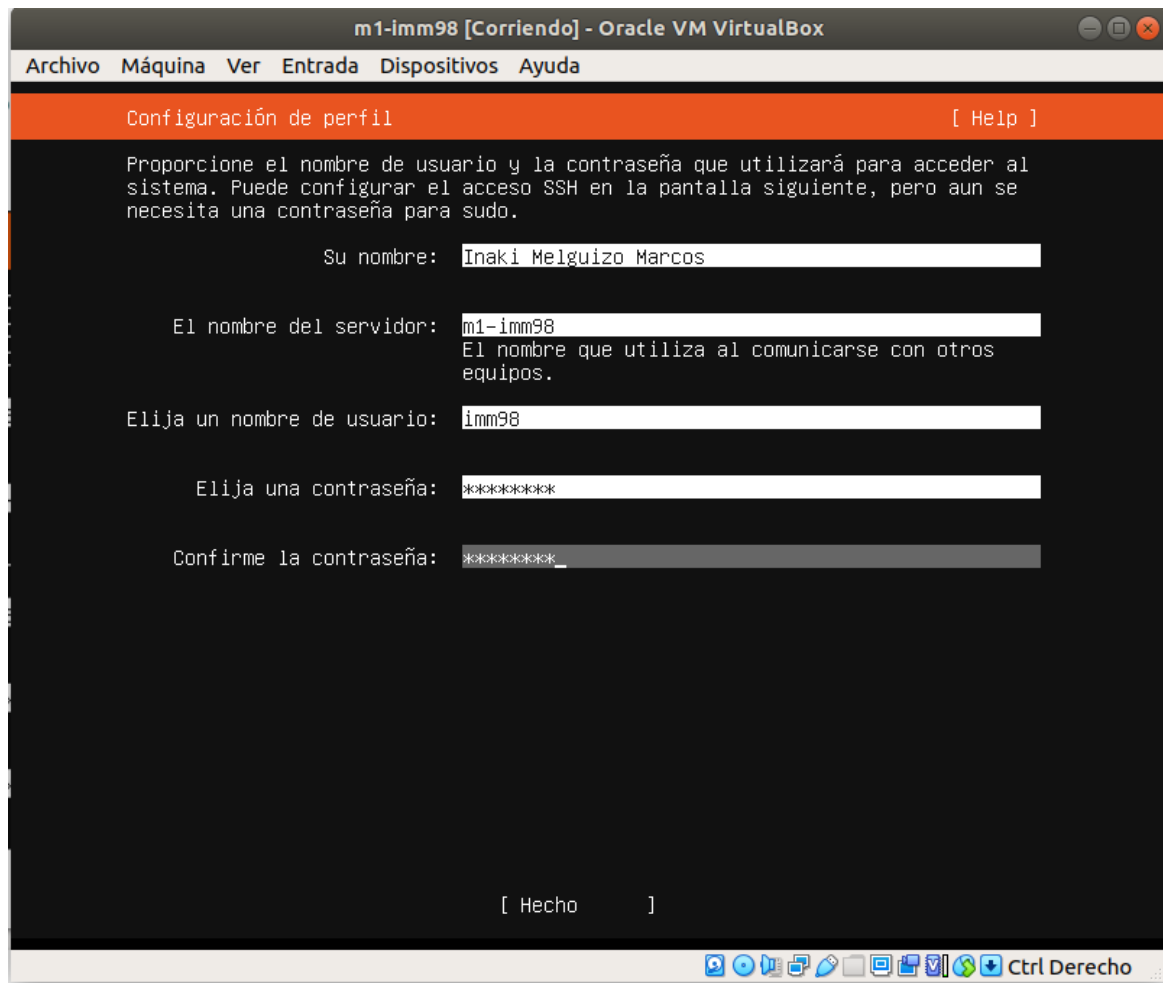
- Un adaptador de tipo **sólo-anfitrión** para poder establecer una comunicación entre las diferentes máquinas virtuales y entre las máquinas virtuales y el host como veremos más adelante.



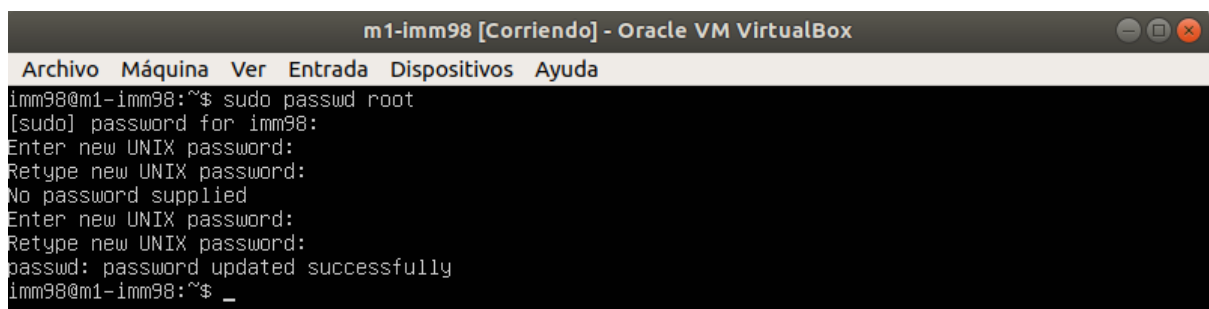
Ya podemos por tanto proceder a iniciar ambas máquinas virtuales y a instalar US en estas:



Introduzco como se indica en el guión de prácticas como usuario mi usuario de la UGR y como contraseña *Swap1234*



Una vez hemos hecho un *reboot* procedemos a la **activación de la cuenta de root** con la orden
sudo passwd root



Durante el proceso de instalación no aparecía la opción “LAMP” por lo que para instalar Apache, PHP y MySQL ejecuto el siguiente comando:
sudo apt-get install apache2 mysql-server mysql-client

```
m1-imm98 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
imm98@m1-imm98:~$ sudo apt-get install apache2 mysql-server mysql-client
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 53 not upgraded.
Need to get 21.5 MB of archives.
After this operation, 164 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

y tras la instalación vemos la versión del servidor:

apache2 -v

```
m1-imm98 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
imm98@m1-imm98:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built:   2020-08-12T21:33:25
imm98@m1-imm98:~$ _
```

y vemos si está en ejecución:

ps aux | grep apache

o

sudo service apache2 status

```

imm98@m1-imm98:~$ ps aux | grep apache
root      2340  0.0  0.4  73960  4740 ?        Ss   10:18   0:00 /usr/sbin/apache2 -k start
www-data  2343  0.0  0.4  826252  4664 ?        Sl   10:18   0:00 /usr/sbin/apache2 -k start
www-data  2344  0.0  0.4  826252  4664 ?        Sl   10:18   0:00 /usr/sbin/apache2 -k start
imm98     3455  0.0  0.1  13212  1060 tty1    S+   10:21   0:00 grep --color=auto apache
imm98@m1-imm98:~$ 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 2021-03-03 10:18:43 UTC; 2min 50s ago
 Main PID: 2340 (apache2)
   Tasks: 55 (limit: 1107)
    CGroup: /system.slice/apache2.service
            └─2340 /usr/sbin/apache2 -k start
              └─2343 /usr/sbin/apache2 -k start
                └─2344 /usr/sbin/apache2 -k start

Mar 03 10:18:43 m1-imm98 systemd[1]: Starting The Apache HTTP Server...
Mar 03 10:18:43 m1-imm98 apachectl[2317]: AH00558: apache2: Could not reliably determine the server's
Mar 03 10:18:43 m1-imm98 systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)

```

y vemos que efectivamente está en ejecución

IP DE LAS DOS MÁQUINAS

Vamos primeramente a configurar las interfaces de red, vamos a configurar una

- IP estática para la interfaz de red Ethernet enp0s8
- una IP dinámica para la interfaz de red Ethernet enp0s3

Para ello tenemos que modificar el archivo `/etc/netplan/00-installer-config.yaml`. Primero vamos a configurar dicho fichero de la máquina virtual m1 y le vamos a asociar a la interfaz de red enp0s8 la IP 192.168.56.105:

```
m1-imm98 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
imm98@m1-imm98:~$ cd /etc/netplan/
imm98@m1-imm98:/etc/netplan$ ls
00-installer-config.yaml
imm98@m1-imm98:/etc/netplan$ sudo vi 00-installer-config.yaml _

# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.56.105/24, ]
      gateway4: 192.168.56.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2
```

En la segunda máquina volvemos a hacer lo mismo pero asociándole a la interfaz de red enp0s8 la IP 192.168.56.104:

```
imm98@m2-imm98: /etc/netplan
Archivo Editar Ver Buscar Terminal Ayuda
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.56.104/24, ]
      gateway4: 192.168.56.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2
```

Ejecutamos la orden `netplan apply` para aplicar los cambios que hemos hecho en las interfaces de red y mediante el comando `ifconfig` vamos a ver la IP de las máquinas `m1-imm98`

```

imm98@m1-imm98: /etc/netplan
Archivo Editar Ver Buscar Terminal Ayuda
imm98@m1-imm98: /etc/netplan$ sudo netplan apply
imm98@m1-imm98: /etc/netplan$ 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:fe8d:8a0f prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:8d:8a:0f txqueuelen 1000 (Ethernet)
    RX packets 36 bytes 8296 (8.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 50 bytes 5673 (5.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.105 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe5:5a4d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:f5:5a:4d txqueuelen 1000 (Ethernet)
    RX packets 597 bytes 54749 (54.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 364 bytes 52213 (52.2 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 88 bytes 6700 (6.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 88 bytes 6700 (6.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

imm98@m1-imm98: /etc/netplan$

```

que es la **192.168.56.105**
y de `m2-imm98`

```

imm98@m2-imm98: /etc/netplan$ sudo netplan apply
imm98@m2-imm98: /etc/netplan$ 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:fe98:5d02 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:98:5d:02 txqueuelen 1000 (Ethernet)
    RX packets 23 bytes 3940 (3.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 40 bytes 4138 (4.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.104 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe21:5e93 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:5e:93 txqueuelen 1000 (Ethernet)
    RX packets 918 bytes 80969 (80.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 578 bytes 79791 (79.7 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 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

```

que es la **192.168.56.104**, es decir:

m1-imm98	192.168.56.105
m2-imm98	192.168.56.104

Probamos a hacer **ping** para comprobar que ambas máquinas se “ven”:

- ping de m1 a m2

```
imm98@m1-imm98:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=0.926 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=1.00 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=0.945 ms
^C
--- 192.168.56.104 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.926/0.958/1.005/0.049 ms
imm98@m1-imm98:~$ _
```

- ping de m2 a m1

```
imm98@m2-imm98:~$ ping 192.168.56.105
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=64 time=1.65 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=64 time=0.958 ms
64 bytes from 192.168.56.105: icmp_seq=3 ttl=64 time=0.930 ms
64 bytes from 192.168.56.105: icmp_seq=4 ttl=64 time=0.858 ms
^C
--- 192.168.56.105 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.858/1.099/1.652/0.322 ms
```

Una vez que ya hemos visto que ambas máquinas se pueden comunicar mediante ping vamos a proceder a modificar el puerto de escucha (por defecto el 80) de apache en una de las dos máquinas (m1) por el 8080. Para ello modificamos el archivo

/etc/apache2/ports.conf

```
imm98@m1-imm98:/etc/apache2$ sudo vi 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 8080_

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Como hemos modificado el archivo de configuración reiniciamos el servicio de apache con **systemctl restart apache2**

```
imm98@m1-imm98:/etc/apache2$ sudo systemctl restart apache2
```

Vemos que está activo el servicio de apache:

```
imm98@m1-imm98:/etc/apache2$ sudo systemctl status apache2.service
● 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 Tue 2021-03-09 16:56:21 UTC; 2min 37s ago
     Process: 2113 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
     Process: 2118 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 2145 (apache2)
      Tasks: 55 (limit: 1107)
   CGroup: /system.slice/apache2.service
           └─2145 /usr/sbin/apache2 -k start
             └─2146 /usr/sbin/apache2 -k start
               └─2147 /usr/sbin/apache2 -k start

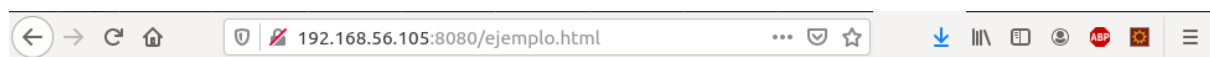
Mar 09 16:56:21 m1-imm98 systemd[1]: Stopped The Apache HTTP Server.
Mar 09 16:56:21 m1-imm98 systemd[1]: Starting The Apache HTTP Server...
Mar 09 16:56:21 m1-imm98 apachectl[2118]: AH00558: apache2: Could not reliably determine the server's
Mar 09 16:56:21 m1-imm98 systemd[1]: Started The Apache HTTP Server.
time: 1.18/1.18 (END)
```

Vemos que está escuchando por el puerto 8080 con la opción
sudo netstat -anp | grep apache2

```
imm98@m1-imm98:/etc/apache2$ sudo netstat -anp | grep apache2
tcp6      0      0 :::8080          :::*              LISTEN     2145/apache2
```

y tras crear un archivo html en el directorio **/var/www/html** llamado ejemplo.html veo que puedo acceder a él:

```
imm98@m1-imm98:/etc/apache2$ cd /var/www/html/
imm98@m1-imm98:/var/www/html$ ls
ejemplo.html  index.html
```



Web de ejemplo de imm98 para SWAP desde m1

Email: imm98@correo.ugr.es

Ahora voy a proceder a crear un **directorio virtual**. Voy a crear un directorio llamado web/ y dentro de ese directorio voy a crear un fichero html

```
imm98@m1-imm98:/var/www/html$ cd /
imm98@m1-imm98:/$ sudo mkdir web
imm98@m1-imm98:/$ ls
bin      dev      initrd.img      lib64      mnt      root      snap      sys      var      web
boot    etc      initrd.img.old  lost+found  opt      run      srv      tmp      vmlinuz
cdrom   home    lib             media      proc     sbin     swap.img  usr      vmlinuz.old
imm98@m1-imm98:/$ cd web/
imm98@m1-imm98:/web$ vi index.html_
```

Ahora modifico el archivo de configuración que hay en el directorio **/etc/apache2/sites-enabled** y añado el directorio virtual que he creado asociándole un alias :

```
imm98@m1-imm98:/web$ cd /etc/apache2/sites-enabled/  
imm98@m1-imm98:/etc/apache2/sites-enabled$ sudo vi 000-default.conf
```

```
Alias /web /web  
  
<Directory /web>  
AllowOverride None  
Require all granted  
</Directory>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
  
"000-default.conf" 40L, 1421C written  
imm98@m1-imm98:/etc/apache2/sites-enabled$ sudo systemctl restart apache2
```

Voy a probar si efectivamente puedo acceder al archivo index.html que he creado en este directorio:



- ssh de m1 a m2:

```
m1-imm98 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
imm98@m1-imm98:~$ ssh imm98@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ECDSA key fingerprint is SHA256:00QSKie5N0dv5zZ7zqzEEdw1J8kVzCCTPPQ2NCh0JLA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.104' (ECDSA) to the list of known hosts.
imm98@192.168.56.104's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

System information as of Wed Mar  3 14:53:16 UTC 2021

System load:  0.0               Processes:            94
Usage of /:   43.5% of 8.79GB   Users logged in:     1
Memory usage: 31%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.104

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

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Mar  3 14:47:28 2021
imm98@m2-imm98:~$ ls
arch2.txt
imm98@m2-imm98:~$
```

- ssh de m2 a m1:

```
m2-imm98 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
imm98@m2-imm98:~$ ssh imm98@192.168.56.105
The authenticity of host '192.168.56.105 (192.168.56.105)' can't be established.
ECDSA key fingerprint is SHA256:KNhKE/23SQicMt2baEMHXo5XRRZE1WJvCcdZ8ZzeI70.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.105' (ECDSA) to the list of known hosts.
imm98@192.168.56.105's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

System information as of Wed Mar  3 14:51:31 UTC 2021

System load:  0.0               Processes:            90
Usage of /:   39.9% of 8.79GB   Users logged in:     1
Memory usage: 14%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.105

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

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Mar  3 14:44:57 2021
imm98@m1-imm98:~$ ls
arch1.txt
imm98@m1-imm98:~$ _
```

y veo que claramente estoy conectado a la otra máquina por lo que cerramos la conexión para terminar lo que queda de la práctica

```
imm98@m2-imm98:~$ exit
logout
Connection to 192.168.56.104 closed.
```

Ahora vamos a cambiar el puerto de ssh. Para ello tendremos que modificar el archivo de configuración `/etc/ssh/sshd_config` y vamos a poner por ejemplo el puerto de ssh al 2222:

```
imm98@m1-imm98:/etc/ssh$ sudo vi sshd_config
```

```
Port 2222
#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
```

y probamos a conectarnos desde la otra máquina indicando con la opción **-p** el nuevo puerto (después de reiniciar el servicio sshd):

```
imm98@m2-imm98:~$ ssh -p 2222 imm98@192.168.56.105
imm98@192.168.56.105's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

System information as of Tue Mar  9 17:56:02 UTC 2021

System load:  0.0               Processes:           94
Usage of /:   43.8% of 8.79GB    Users logged in:    1
Memory usage: 32%               IP address for enp0s3: 10.0.2.15
Swap usage:   0%                IP address for enp0s8: 192.168.56.105

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

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

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Mar  9 17:46:13 2021 from 192.168.56.104
```

Para conectarnos mediante clave pública tenemos que generar una clave pública por lo que ejecutaremos el comando ssh-keygen:

```
imm98@m2-imm98:~$ ssh-keygen
ssh-keygen  ssh-keyscan
imm98@m2-imm98:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/imm98/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/imm98/.ssh/id_rsa.
Your public key has been saved in /home/imm98/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:NqETxu8V4Db1/TYRQoJWwhLpIYXfh14mQ+cGz7SerU imm98@m2-imm98
The key's randomart image is:
+---[RSA 2048]-----+
|
|  .+*=0000.  ..
|  0.=X=.+0.  ..
|  .+=0B+..=..
|  0..*B+0+ 0
|    0=0S0
|    . .+E.
|    .
|-----[SHA256]-----+
```

que nos crea una clave pública y una privada y nos las mete en los archivos `~/.ssh/id_rsa.pub` y `~/.ssh/id_rsa` respectivamente. Ahora tendremos que pasarle al servidor donde deseamos conectarnos dicha clave pública y esto se hace con el comando `ssh-copy-id` que nos guarda dicha clave pública en un archivo llamado `~/.ssh/authorized_keys`

```
imm98@m1-imm98:~/.ssh$ ssh-copy-id imm98@192.168.56.104
```

Una vez que ya tenga guardada esa clave en `authorized_keys` ya me podré conectar sin contraseña de m1 a m2:

```
id_rsa id_rsa.pub known_hosts
imm98@m1-imm98:~/.ssh$ ssh imm98@192.168.56.104
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

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

System information as of Wed Mar 10 10:19:46 UTC 2021

System load:  0.0               Processes:    95
Usage of /:   43.8% of 8.79GB    Users logged in: 1
Memory usage: 32%              IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for enp0s8: 192.168.56.104

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.
   https://microk8s.io/high-availability

imm98@m2-imm98:~$ cd ~/.ssh/
imm98@m2-imm98:~/.ssh$ ls
authorized_keys id_rsa id_rsa.pub known_hosts
imm98@m2-imm98:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQGf2pJdhcSvvsajvvU1zWo6x55c23QH2MJ2VjQUXWPFH/XL4y2BY3hp2YoDEH
saSIy19gx3MnCGdxj1HKSWGXTSB5PsR6ZBLkcV6Rt2V31Ybn1HgHj1Sd4qIezCrMvV0ozaXRR6m4Xnh820Fc/oAiQEHmtmGt107F
FEPDB6QbnJyM1Wn8c000CME5mnCqQ88vshS0YweCnY0UuPiFCv0Lxp+accbMb28dt6KIb1xd0bNnKF7nYhjIhw9mU141h2TKJ9a
RU99bod3C2T/uws8tmVnPy2L1BWL3161TdGgz5XChKnxyDUBG04/Gu818YQCXWfY1QyX5ShaMYjJFzmj imm98@m1-imm98
```

Ahora vamos a comprobar que apache está funcionando para ello creamos dos archivos HTML en el directorio `/var/www/html/` en cuyo contenido viene en que máquina (m1 o m2) están localizados y vamos a acceder a ellos desde la otra máquina usando cURL. Para ello utilizo el comando :

curl direcciónIP/ejemplo.html

- m1 a m2

```
imm98@m1-imm98:/etc/ssh$ curl 192.168.56.104/ejemplo.html
<HTML>
  <BODY>
    Web de ejemplo de imm98 para SWAP en m2
    Email: imm98@correo.ugr.es
  </BODY>
</HTML>
```

- m2 a m1

```
imm98@m2-imm98:~/.ssh$ curl 192.168.56.105:8080/ejemplo.html
<HTML>
  <BODY>
    <h1>Web de ejemplo de imm98 para SWAP desde m1</h1>
    <p>Email: imm98@correo.ugr.es</p>
  </BODY>
</HTML>
```

Por último vamos a proceder a descargar un archivo, en lugar de que lo muestre por la salida estándar. Para ello utilizo la orden:

curl -o imagen.png <https://www.google.es/images/srpr/logo3w.png>

```
imm98@m1-imm98:~$ sudo 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  76681      0 --:--:-- --:--:-- --:--:-- 76681
imm98@m1-imm98:~$ ls
imagen.png
```

La opción curl -O nos descarga un archivo y le pone el nombre del archivo que estamos descargando:

curl -O <https://www.google.es/images/srpr/logo3w.png>

```
imm98@m1-imm98: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
imm98@m1-imm98:~$ curl -O 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  65514      0 --:--:-- --:--:-- --:--:-- 65514
imm98@m1-imm98:~$ ls
arch1.txt  logo3w.png
```

Para el tema de las cookies tenemos dos opciones principales `curl -b` y `curl -c`:
curl -c se utiliza para guardar las cookies de respuesta en un determinado fichero. En nuestro ejemplo vamos a guardarlas en el fichero `cookies.txt`. La opción **curl -b** set a cookie with the values written in a text file in a GET request.

```
imm98@m1-imm98: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
imm98@m1-imm98:~$ curl -c cookies.txt https://google.com  
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">  
<TITLE>301 Moved</TITLE></HEAD><BODY>  
<H1>301 Moved</H1>  
The document has moved  
<A HREF="https://www.google.com/">here</A>.  
</BODY></HTML>  
imm98@m1-imm98:~$ cat cookies.txt  
# Netscape HTTP Cookie File  
# https://curl.haxx.se/docs/http-cookies.html  
# This file was generated by libcurl! Edit at your own risk.  
  
.google.com TRUE / FALSE 2145916800 CONSENT PENDING+480  
imm98@m1-imm98:~$ curl -b cookies.txt https://google.com  
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">  
<TITLE>301 Moved</TITLE></HEAD><BODY>  
<H1>301 Moved</H1>  
The document has moved  
<A HREF="https://www.google.com/">here</A>.  
</BODY></HTML>  
imm98@m1-imm98:~$ ls  
arch1.txt cookies.txt logo3w.png
```

Para realizar una petición POST se hace con la opción **curl -X POST -F** junto con el archivo que desees enviarle al servidor:

```
imm98@m1-imm98: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
imm98@m1-imm98:~$ curl -X POST -F 'image=@/home/imm98/logo3w.png' http://www.google.es  
<!DOCTYPE html>  
<html lang=en>  
  <meta charset=utf-8>  
  <meta name=viewport content="initial-scale=1, minimum-scale=1, width=device-width">  
  <title>Error 413 (Request Entity Too Large)!!1</title>  
  <style>  
    *{margin:0;padding:0}html,code{font:15px/22px arial,sans-serif}html{background:#fff;color:#222;padding:15px}body{margin:7% auto 0;max-width:390px;min-height:180px;padding:30px 0 15px}* > body{background:url(//www.google.com/images/errors/robot.png) 100% 5px no-repeat;padding-right:205px}p{margin:11px 0 22px;overflow:hidden}ins{color:#777;text-decoration:none}a img{border:0}@media screen and (max-width:772px){body{background:none;margin-top:0;max-width:none;padding-right:0}}#logo{background:url(//www.google.com/images/branding/googlelogo/1x/googlelogo_color_150x54dp.png) no-repeat;margin-left:-5px}@media only screen and (min-resolution:192dpi){#logo{background:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat 0% 0%/100% 100%;-moz-border-image:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) 0}}@media only screen and (-webkit-min-device-pixel-ratio:2){#logo{background:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat;-webkit-background-size:100% 100%}}#logo{display:inline-block;height:54px;width:150px}  
  </style>  
  <a href=//www.google.com/><span id=logo aria-label=Google></span></a>  
  <p><b>413.</b> <ins>That's an error.</ins>  
  <p>Your client issued a request that was too large.  
<script>  
(function() { /*
```


Para realizar una petición GET simplemente hay que hacer **curl Dirección IP**

```
imm98@m1-imm98: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
imm98@m1-imm98:~$ curl http://www.google.es  
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="es"><head><meta  
content="Google.es permite acceder a la información mundial en castellano, catalán, galle  
go, euskara e inglés." name="description"><meta content="noodp" name="robots"><meta content  
="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googl  
eg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce  
="4azwTBXuIpxR5HiC04x3fw==">(function(){window.google={kEI:'5bJIYKS8DI2eUNL-p9AJ',kEXPI:'0,  
1302433,7,56969,954,5104,207,4804,2316,383,246,5,306,1047,5252,1504,4881,1116130,1232,11964  
73,577,328985,51224,16114,17444,11240,9188,8384,4859,1361,9290,3026,4742,12841,4020,978,132  
28,2054,920,873,4192,6430,14527,4518,2777,919,2277,9,2795,1593,1279,1042,1170,531,148,1103,  
840,517,1466,56,4224,34,1449,2,2063,606,2023,1777,520,4269,328,1284,8789,3228,1026,1818,7,4  
773,7581,5096,7877,4928,108,3407,908,2,941,2614,2397,7470,3275,3,346,230,1835,4625,148,5990  
,6324,1661,4,498,1030,2304,1236,1145,4081,577,1791,127,4,2762,459,1555,4067,7060,3041,2454,  
1753,2658,4243,518,912,564,464,656,30,3854,4275,3283,3,2213,2305,638,1494,596,2,1493,3495,9  
206,587,11,731,665,2145,377,3293,2542,479,511,2,1962,1140,20,47,99,2972,6,908,3,3541,1,4174  
,1843,60,922,2295,362,180,109,368,132,1497,2768,1187,627,38,245,709,3637,1170,770,618,1260,  
239,955,2,1056,2017,1225,32,369,31,2859,423,287,4,73,121,18,1555,2,77,1317,549,867,143,365,  
192,223,2,465,845,493,7,370,2,1459,1598,554,782,4518,467,7,333,323,345,338,555,4,106,391,2,  
2,810,2,1316,108,39,3,100,967,761,2,61,582,36,253,277,729,5,1,1,1,253,287,134,320,106,924,7  
34,727,664,5,5679267,3505,204,161,35,204,5997100,2799384,1323,549,333,444,1,2,80,1,1796,1,9  
,2,2551,1,748,141,795,563,1,4265,1,1,2,1331,3299,843,1,2608,155,17,13,72,174,9,155,13,16,46  
,5,37,11,45,43,18,1,1,4,4,5,4,1,80,23956723,150,2790459,1219664,267,723,26807',kBL:'4Bw4'};  
google.sn='webhp';google.kHL='es';})();(function(){  
var f=[];google.getEI=function(a){for(var b;a&&(!a.getAttribute)||!(b=a.getAttribute("eid"))  
);)a=a.parentNode;return b||google.kEI};google.getLEI=function(a){for(var b=null;a&&(!a.get  
Attribute)||!(b=a.getAttribute("leid"));)a=a.parentNode;return b||google.ml=function(){retu
```