

SWAP: Preparación de las herramientas

David Cabezas Berrido

dxabezas@correo.ugr.es

6 de marzo de 2021

Índice

1. Objetivos	2
2. Creación de las máquinas e instalación del sistema	2
3. SSH	3
4. Configuración de red	4
5. Servicios Apache	5

1. Objetivos

El objetivo de esta primera práctica es la puesta en marcha de dos máquinas virtuales idénticas que puedan conectarse a internet y entre sí. Así como la instalación y configuración de ciertos servicios y herramientas como son Apache, PHP, MySQL, SSH o CURL.

2. Creación de las máquinas e instalación del sistema

Comenzamos creando dos máquinas virtuales idénticas, seleccionamos Ubuntu-64b y la configuración recomendada (1 core, 1GB de RAM y 10GB de disco dinámicos).

Ambas máquinas vienen con un adaptador de red NAT por defecto, añadimos un segundo adaptador de red Host-only (Settings -> Network -> Adapter 2 -> Host-only Adapter) para que las máquinas puedan comunicarse entre sí. Si no tenemos ninguno, lo podemos crear en File -> Host Network Manager -> Create.

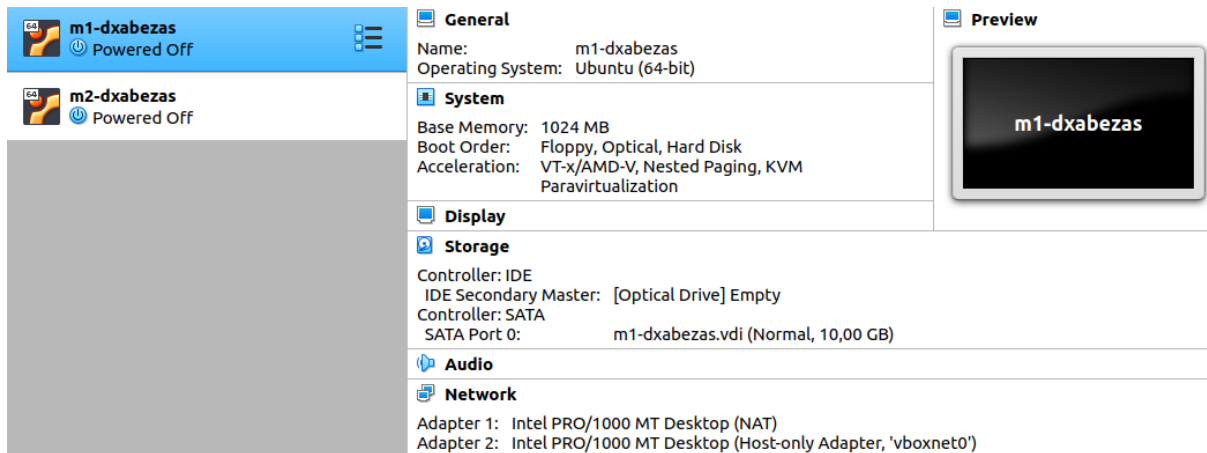


Figura 1: Resumen de la configuración de la máquina M1. Observamos que tiene los dos adaptadores de red que hemos comentado.

Seguidamente, instalamos Ubuntu Server 18.04 LTS en ambas máquinas. Podemos descargar la ISO de la página oficial. Creamos en las dos máquinas perfiles idénticos, con usuario **dxabezas** y contraseña **Swap1234**.

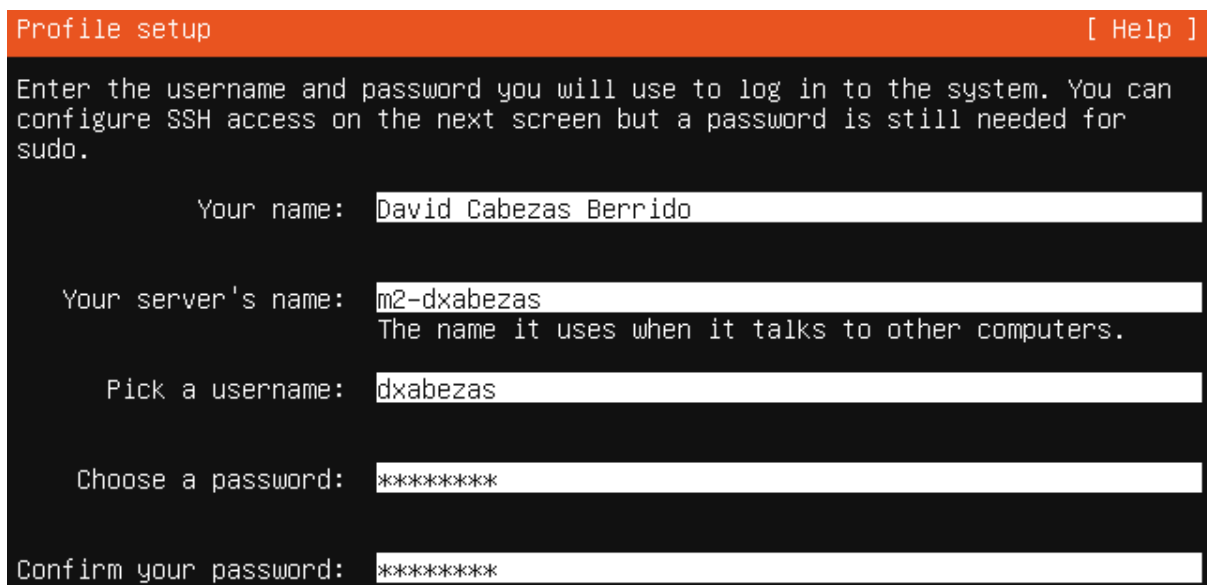


Figura 2: Creando el perfil en la máquina M2.

Durante la instalación escogemos siempre las opciones recomendadas, con la excepción de instalar OpenSSH.

3. SSH

Ya tenemos OpenSSH instalado en ambas máquinas. Si durante la instalación no lo hubiésemos marcado, tendríamos que ejecutar:

```
sudo apt-get install openssh-client
sudo apt-get install openssh-server
```

Con `ifconfig`, podemos ver la dirección IP de cada máquina:

```
dxabezas@m1-dxabezas:~$ 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:fe59:eedf prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:59:ee:fd txqueuelen 1000 (Ethernet)
    RX packets 7 bytes 1680 (1.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 1788 (1.7 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:fe99:111a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:99:11:1a txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 3042 (3.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 1408 (1.4 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 164 bytes 12244 (12.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 164 bytes 12244 (12.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

(a) Máquina 1.

```
dxabezas@m2-dxabezas:~$ 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:fea3:214f prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a3:21:4f txqueuelen 1000 (Ethernet)
    RX packets 21 bytes 6126 (6.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29 bytes 3564 (3.5 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:febb:4fc3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:bb:4f:c3 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 3042 (3.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 1408 (1.4 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
```

(b) Máquina 2.

Figura 3: Direcciones IP de ambas máquinas.

Nos conectamos de una máquina a otra, ejecutando `ssh user@ip`.

```
dxabezas@m1-dxabezas:~$ ssh dxabezas@192.168.56.101
dxabezas@192.168.56.101'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 16:45:54 UTC 2021

System load:  0.06          Processes:           92
Usage of /:   40.0% 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: 192.168.56.101

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

   https://microk8s.io/high-availability

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 16:22:57 2021
dxabezas@m2-dxabezas:~$
```

(a) M1 se conecta a M2.

```
dxabezas@m2-dxabezas:~$ ssh dxabezas@192.168.56.102
dxabezas@192.168.56.102'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 16:46:03 UTC 2021

System load:  0.06          Processes:           91
Usage of /:   40.1% 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: 192.168.56.102

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 16:22:44 2021
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dxabezas@m1-dxabezas:~$
```

(b) M2 se conecta a M1.

Figura 4: Cada máquina se conecta a la otra por SSH con el usuario que creamos (dxabezas, Swap1234).

NOTA: En lo que viene a continuación, la dirección IP de M1 es la que termina en 1 y la de M2 es la que termina

en 2. En la siguiente sección explico cómo las intercambio.

Ahora vamos a configurar SSH, para ello modificamos el fichero `/etc/ssh/sshd_config`, donde se encuentra la configuración del servicio `ssh-server`. `/etc/ssh/ssh_config` es para la configuración de `ssh-client`. Por ejemplo, podemos modificar el puerto del 22 (por defecto) al 2222 con la línea `Port 2222`. Para hacer efectivos los cambios, guardamos el archivo y reestauramos el servicio con

```
sudo service ssh restart
```

Ahora desde la otra máquina nos conectamos, pero debemos indicar el puerto explícitamente (con la opción `-p`) al no tratarse del por defecto.

```
dxabezas@m2-dxabezas:/etc/netplan$ ssh dxabezas@192.168.56.101
ssh: connect to host 192.168.56.101 port 22: Connection refused
dxabezas@m2-dxabezas:/etc/netplan$ ssh -p 2222 dxabezas@192.168.56.101
dxabezas@192.168.56.101's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)
```

Figura 5: La máquina 2 se intenta conectar a la máquina 1. El primer intento fracasa porque intenta conectarse al puerto por defecto.

Una buena práctica desde el punto de vista de la seguridad es deshabilitar el root login (`PermitRootLogin no`). De esta manera, nadie se puede conectar como root por SSH, tendrá que obtener permisos de superusuario una vez logueado en el sistema mediante `sudo`.

Finalmente, configuraremos el acceso sin contraseña mediante clave pública. Desde la máquina M2 generamos un par de claves con `ssh-keygen -t rsa`, nos da la opción de poner una passphrase por si tememos que alguien pueda acceder a nuestro dispositivo.

```
dxabezas@m2-dxabezas:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dxabezas/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dxabezas/.ssh/id_rsa.
Your public key has been saved in /home/dxabezas/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:6qNcC9g560/Zo2Iaz1COMH0bi+wg4Mq4pRPITb0unyw dxabezas@m2-dxabezas
```

Figura 6: La máquina 2 genera un par de claves (pública y privada).

Ahora copiamos la clave pública en M1 ejecutando (desde M2)

```
ssh-copy-id -p 2222 dxabezas@192.168.56.101
```

nos pide la contraseña (Swap1234) y nos dice que se ha añadido 1 clave. Ahora nos podemos conectar con SSH sin poner sin que nos pida la contraseña, nos pedirá la passphrase en caso de haber puesto una.

Hemos elegido crear la pareja de claves en los archivos por defecto (`~/.ssh/id_rsa` y `~/.ssh/id_rsa.pub`). De haber seleccionado otros archivos, tendríamos que indicar con la opción `-i` el archivo de clave a copiar (pública) y a utilizar para autenticarse (privada).

4. Configuración de red

Ya tenemos los dos adaptadores de red (NAT y Host-only) creados. Cuando instalamos el sistema se configuran por defecto, pero podemos usar Netplan para realizar los cambios que deseemos.

Para ello, modificamos el fichero `00-installer-config.yaml` en la carpeta `/etc/netplan`, donde ya encontramos la configuración por defecto del adaptador NAT (interfaz `enp0s3`) y del adaptador Host-only (interfaz `enp0s8`), que reciben direcciones IP via DHCP. Podemos consultar las direcciones asignadas con `ifconfig` (Figura 3).

```
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: true
```

En la Figura 3 también observamos que las direcciones obtenidas están “cambiadas”, la máquina 1 recibió 192.168.56.102 y la máquina 2 la 192.168.56.101. Esto se debe a que realizamos primero la instalación de la M2, pero ahora podemos cambiarlas manualmente para tener direcciones más intuitivas. También en la Figura 3 observamos que se utiliza la máscara de red 255.255.255.0 (24 bits a 1 y 8 bits a 0), la mantenemos añadiendo /24 tras la dirección. Editamos el archivo con `sudo nano 00-installer-config.yaml`. En la máquina 1 escribimos:

```
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses: [192.168.56.101/24]
```

y en la M2 cambiamos a 192.168.56.102/24. Para aplicar los cambios ejecutamos `sudo netplan apply`, podemos comprobarlos con `ifconfig` y conectarnos por ssh desde la máquina anfitriona o la otra máquina para comprobar que funciona. Si ya nos conectamos desde la máquina anfitriona antes de este cambio, habrá que borrar la fingerprint de known hosts primero.

```
dxabezas@m1-dxabezas:/etc/netplan$ 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:fe59:ee:fd prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:59:ee:fd txqueuelen 1000 (Ethernet)
    RX packets 6379 bytes 7823098 (7.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1303 bytes 89806 (89.8 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:fe99:111a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:99:11:1a txqueuelen 1000 (Ethernet)
    RX packets 211 bytes 29141 (29.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 200 bytes 29563 (29.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

(a) La IP de M1 ha cambiado.

```
dxabezas@Lenovo:~$ ssh dxabezas@192.168.56.101
dxabezas@192.168.56.101'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 Fri Mar  5 17:40:29 UTC 2021

System load:  0.03          Processes:      93
Usage of /:   40.2% of 8.79GB Users logged in:   1
Memory usage: 15%          IP address for enp0s3: 10.0.2.15
Swap usage:  0%            IP address for enp0s8: 192.168.56.101
```

(b) Nos conectamos a M1 desde el anfitrión con la nueva IP.

Figura 7: Conexión por SSH desde el anfitrión a M1 con la nueva IP.

5. Servicios Apache

Puesto que no marcamos los servicios LAMP durante la instalación, debemos instalarlos manualmente. Empezamos con Apache2, en ambas máquinas ejecutamos:

```
sudo apt install -y apache2
```

Para comprobar la versión que hemos instalado usamos `apache2 -v`, y para comprobar que esté en ejecución,

```
sudo service apache2 status
```

```

dxabezas@m1-dxabezas:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2020-08-12T21:33:25
dxabezas@m1-dxabezas:~$ 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 16:57:18 UTC; 55min ago
     Main PID: 2251 (apache2)
       Tasks: 55 (limit: 1107)
    CGroup: /system.slice/apache2.service
            └─2251 /usr/sbin/apache2 -k start
               2253 /usr/sbin/apache2 -k start
               2254 /usr/sbin/apache2 -k start

```

(a) Máquina 1.

```

dxabezas@m2-dxabezas:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2020-08-12T21:33:25
dxabezas@m2-dxabezas:~$ 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 17:00:15 UTC; 52min ago
     Main PID: 2242 (apache2)
       Tasks: 55 (limit: 1107)
    CGroup: /system.slice/apache2.service
            └─2242 /usr/sbin/apache2 -k start
               2244 /usr/sbin/apache2 -k start
               2245 /usr/sbin/apache2 -k start

```

(b) Máquina 2.

Figura 8: Comprobamos la versión de Apache2 y que esté activo.

En la máquina 1 cambiamos el puerto de escucha al 8000 escribiendo `Listen 8000` en el fichero `/etc/apache2/ports.conf`. En el fichero avisa de que modifiquemos también `/etc/apache2/sites-enabled/000-default.conf`, donde cambiamos la línea

```
<VirtualHost *:80>
```

por `<VirtualHost *:8000>`.

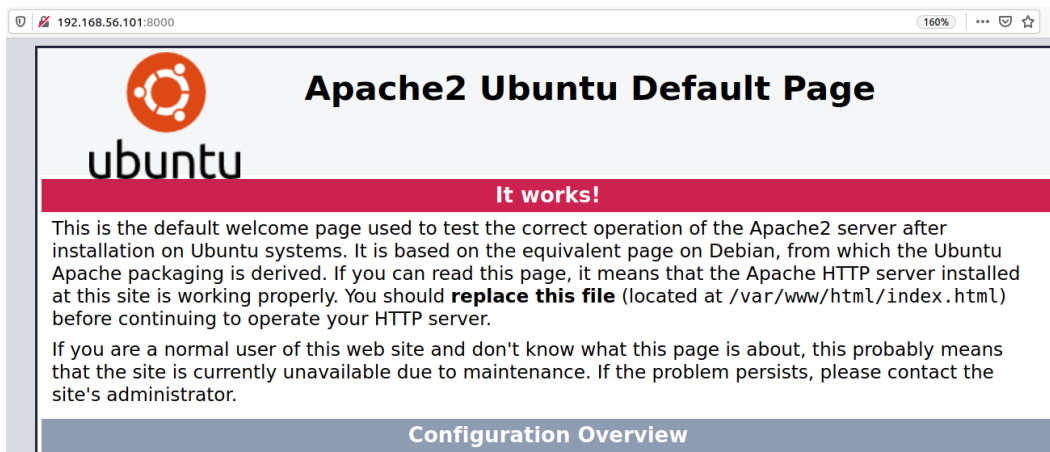
Ejecutamos `sudo apache2ctl configtest` para comprobar que está todo bien. Nos da un warning diciendo que le pongamos un nombre de dominio al servidor, ya que actualmente se está usando (127.0.1.1). Lo cambiamos añadiendo la línea

```
ServerName 192.168.56.101
```

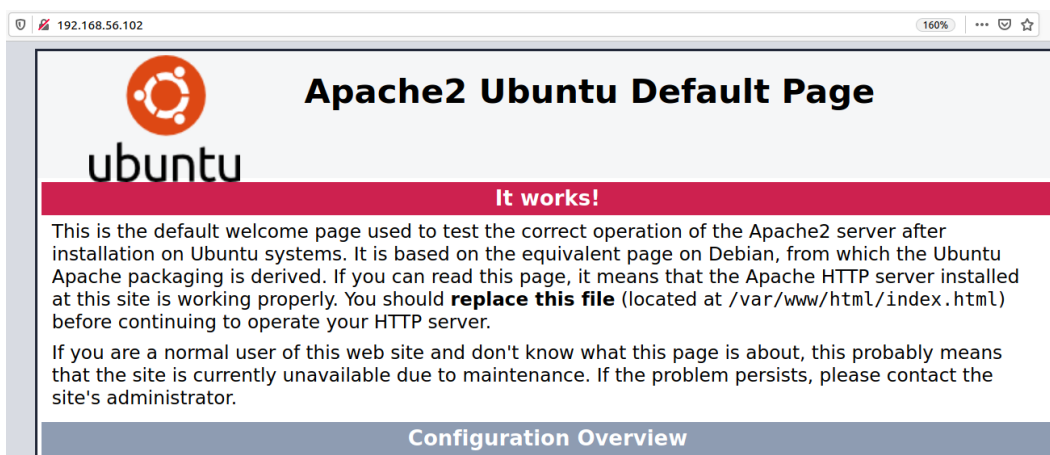
al fichero `/etc/apache2/apache2.conf`. También cambiamos el nombre a 192.168.56.102 en la M2. Ahora la comprobación sólo nos devuelve Syntax OK. Reestablecemos el servicio para que se apliquen los cambios.

```
sudo systemctl restart apache2
```

Desde la máquina anfitriona podemos visitar las direcciones 192.168.56.101:8000 (M1, hemos cambiado el puerto de escucha) y 192.168.56.102 (M2, puerto 80 por defecto). Comprobamos que Apache funciona correctamente.



(a) M1, al puerto 8000.



(b) M2, al puerto por efecto (80).

Figura 9: El servicio Apache funciona correctamente, hemos cambiado el puerto de escucha en M1.

Ahora instalamos cURL:

```
sudo apt install curl
```

y nos dice que ya está instalado y en su última versión.

Finalmente, instalaremos PHP: