

Workshop 02

¿Qué necesitamos para desplegar una aplicación web?

- Servidor
- Dominio
- IP
- Aplicación (backend, frontend, fullstack)
- Base de datos
- Presupuesto
- Seguridad
 - Firewall
- Seo
 - Analítica

Implementación de un servidor LAMP

1. **Iniciar la maquina:** Es importante que se inicialice donde se ubique el archivo "Vagrantfile".

```
cd ~/ISW811/VMs/webserver  
vagrant up
```

2. **Conectarse por SSH:** Nos conectamos a la maquina vagrant tipo GNU/Linux por medio de ssh.

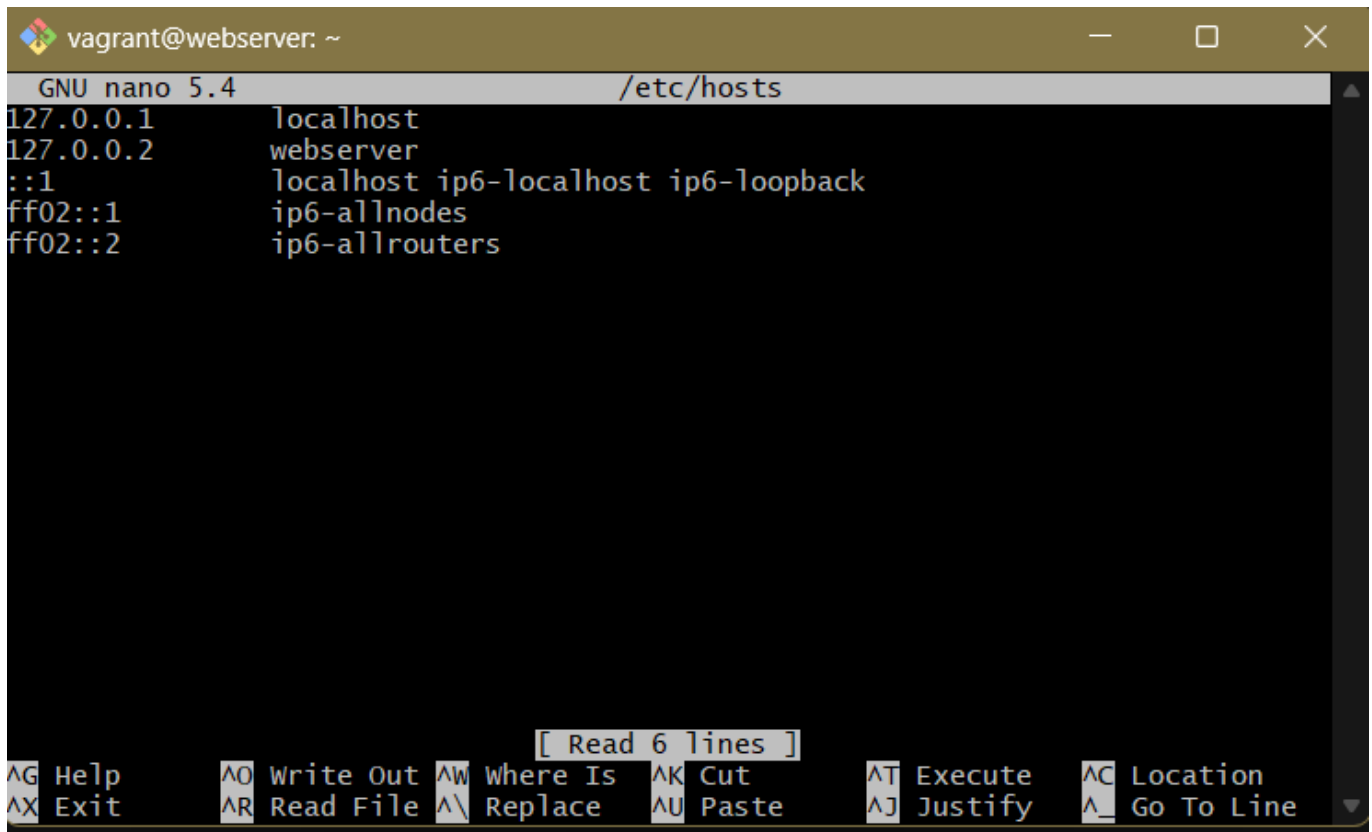
```
cd ~/ISW811/VMs/webserver  
vagrant ssh
```

3. **Cambiar hostname de la maquina virtual:** El siguiente comando se debe ejecutar dentro de la maquina virtual para cambiarle el nombre del host que tenia predeterminado. Para poder ver el cambio se sale de la maquina virtual y se vuelve a ingresar.

```
sudo hostnamectl set-hostname webserver  
exit  
vagrant ssh
```

4. **Actualizar el hostname en el archivo "hosts":** Para completar el cambio hay que actualizar el nombre de la maquina en el archivo hosts

```
sudo nano /etc/hosts
```



```

vagrant@webserver: ~
GNU nano 5.4 /etc/hosts
127.0.0.1    localhost
127.0.0.2    webserver
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters

[ Read 6 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

5. **Actualizar la lista de paquetes elegibles:** Antes de instalar cualquier paquete se debe actualizar la base de datos de paquetes de la maquina virtual.

```
sudo apt-get update
```

6. **Instalar paquetes:** Para instalar paquetes vim, curl, apache2, mysql, y php.

```

sudo apt-get install vim vim-nox \
    curl git apache2 mariadb-server mariadb-client \
    php7.4 php7.4-bcmath php7.4-curl php7.4-json \
    php7.4-mbstring php7.4-mysql php7.4-xml

```

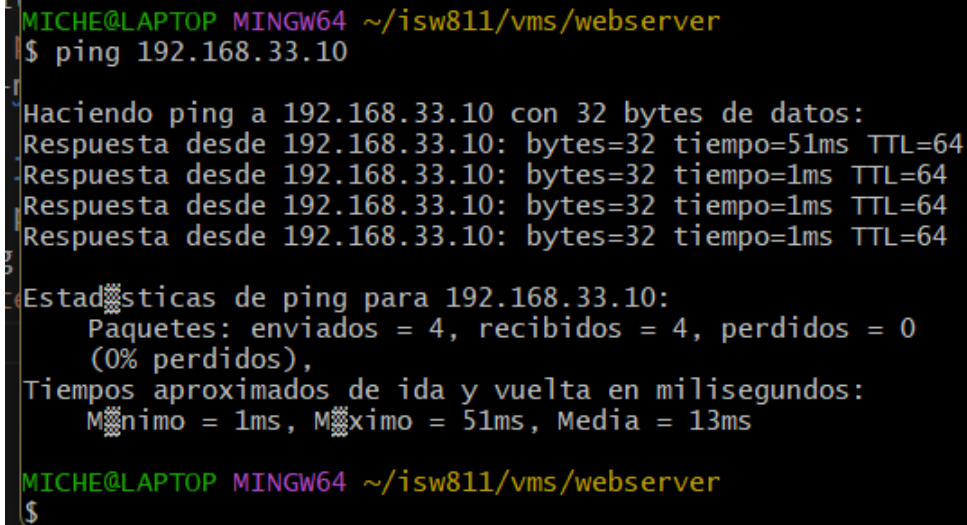
7. **Comprobar la IP del servidor:** desde la maquina anfitriona verificamos la ip definida en el parámetro `private_network`, en este caso es "192.168.33.10", al final le hacemos ping a la dirección ip.

```

29 # within the machine from a port on the host machine and only allow access
30 # via 127.0.0.1 to disable public access
31 # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
32
33 # Create a private network, which allows host-only access to the machine
34 # using a specific IP.
35 config.vm.network "private_network", ip: "192.168.33.10"
36
37 # Create a public network, which generally matched to bridged network.
38 # Bridged networks make the machine appear as another physical device on
39 # your network.
40 # config.vm.network "public_network"
41

```

```
ping 192.168.33.10
```

A terminal window with a dark background. The prompt is 'MICHE@LAPTOP MINGW64 ~/isw811/vms/webserver'. The user enters '\$ ping 192.168.33.10'. The output shows four successful ping responses with 32 bytes, 51ms time, and TTL=64. It also shows statistics: 4 packets sent, 4 received, 0 lost, and average times of 1ms, 51ms, and 13ms.

```
MICHE@LAPTOP MINGW64 ~/isw811/vms/webserver
$ ping 192.168.33.10

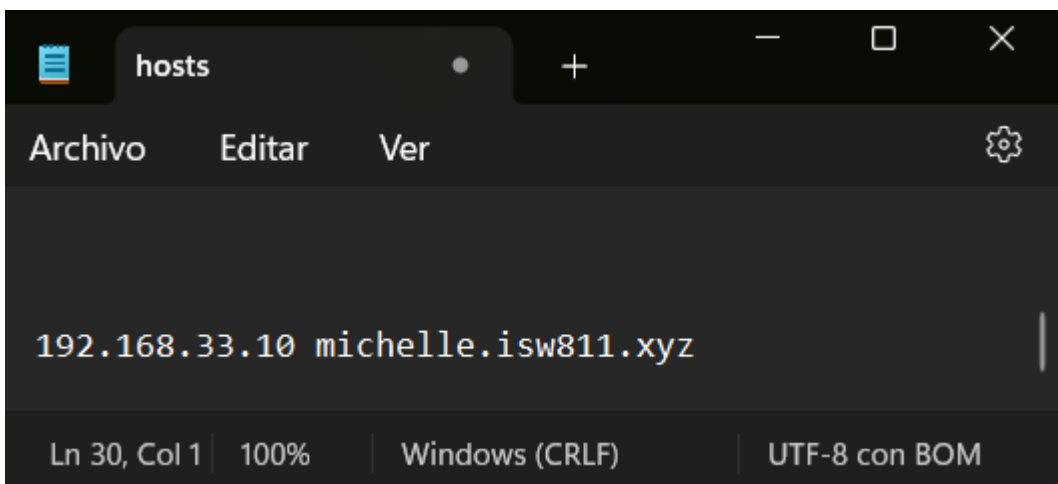
Haciendo ping a 192.168.33.10 con 32 bytes de datos:
Respuesta desde 192.168.33.10: bytes=32 tiempo=51ms TTL=64
Respuesta desde 192.168.33.10: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.33.10: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.33.10: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.33.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 51ms, Media = 13ms

MICHE@LAPTOP MINGW64 ~/isw811/vms/webserver
$
```

8. **Editar el archivo hosts:** desde la maquina anfitriona, ejecutamos la ruta windows/system32/drivers/etc como administrador, en el archivo host insertamos la resolución del dominio deseado y lo abrimos en el navegador para ver que salga la pagina por defecto y todo funcione correctamente.

```
cd \
cd Windows\System32\drivers\etc
notepad hosts
192.168.33.10 michelle.isw811.xyz
```



9. **Habilitar módulos:** Verificamos el estado de apache, y se habilitan modulos para soportar host y certificados SSL para poder publicar el sitio HTTPS, al final reiniciamos apache.

```
sudo systemctl status apache2.service
sudo a2enmod vhost_alias rewrite ssl
sudo systemctl restart apache2
```

10. **Montar carpeta de sitios:** Para mejorar el flujo de trabajo vamos a crear un folder local y lo sincronizamos contra la ruta /home/vagrant/sites de la máquina virtual. Para esto editamos el archivo "VagrantFile"

```
config.vm.synced_folder "sites/", "/home/vagrant/sites", owner: "www-data", group: "www-data"
```

```
44 # the path on the guest to mount the folder. And the optional third
45 # argument is a set of non-required options.
46 # config.vm.synced_folder "../data", "/vagrant_data"
47 config.vm.synced_folder "sites/", "/home/vagrant/sites", owner: "www-data", group: "www-data"
48
49 # Disable the default share of the current code directory. Doing this
50 # provides improved isolation between the vagrant box and your host
51 # by making sure your vagrantfile isn't accessible to the vagrant box
```

11. **Reiniciar maquina:** Luego de modificar el Vagrantfile debemos reiniciar la máquina para que los cambios surtan efectos

```
exit
vagrant halt
vagrant up
vagrant ssh
```

12. **Crear el «conf» para el sitio:** Necesitaremos crear un archivo .conf para cada sitio que deseemos hospedar en el servidor web.

```
cd ~/ISW811/VMs/webserver
mkdir confs
cd confs
touch michelle.isw811.xyz.conf
code michelle.isw811.xyz.conf
```

Una vez creado el archivo .conf le agregamos el siguiente contenido:

```
<VirtualHost *:80>
  ServerAdmin webmaster@michelle.isw811.xyz
  ServerName michelle.isw811.xyz

  # Indexes + Directory Root.
  DirectoryIndex index.php index.html
  DocumentRoot /home/vagrant/sites/michelle.isw811.xyz

  <Directory /home/vagrant/sites/michelle.isw811.xyz>
    DirectoryIndex index.php index.html
    AllowOverride All
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/michelle.isw811.xyz.error.log
  LogLevel warn
```

```
CustomLog ${APACHE_LOG_DIR}/michelle.isw811.xyz.access.log combined
</VirtualHost>
```

15. **Copiar "conf" a "sites-available".** Desde la máquina virtual se va a copiar el archivo.conf a lo que viene siendo la ruta de sitios disponibles de Apache2. Al final reiniciamos Apache.

```
sudo cp /vagrant/confs/michelle.isw811.xyz.conf /etc/apache2/sites-available
sudo apache2ctl -t
```

```
vagrant@webserver:~$ sudo nano /etc/apache2/sites-available/michelle.isw811.xyz.conf
vagrant@webserver:~$ sudo apache2ctl -t
Syntax OK
vagrant@webserver:~$ |
```

17. **Configurar el parámetro "ServerName"** Si al probar la configuración de Apache obtenemos el error Could not reliably determine the server's fully qualified domain name, debemos ejecutar el siguiente comando, para agregar la directiva «ServerName» al archivo de configuración general de Apache, usando el siguiente comando:

```
echo "ServerName webserver" | sudo tee -a /etc/apache2/apache2.conf
```

18. **Habilitar el nuevo sitio:** Si ya no se encuentran errores, habilitamos el sitio y reiniciamos apache nuevamente.

```
sudo a2ensite michelle.isw811.xyz.conf
sudo systemctl restart apache2.service
```

19. **Verificar el nuevo sitio:** Ingresamos la URL <http://michelle.isw811.xyz.conf>

