

Workshop 02/LAMP

Que necesitamos para desplegar una aplicación web?

- Servidor web
- Almacenamiento
- Dominio
- Una IP
- Una App (backend, frontend, fullstack)
- Base de datos
- Presupuesto
- Seguridad
 - Firewall
- SEO
 - Analytics

Se realizará un servidor con LAMP (Linux, Apache, MySQL y PHP)

Para iniciar la máquina vagrant en el anfitrión

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

Para iniciar la máquina vagrant con ssh

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

Para cambiar el nombre de host se debe ejecutar este comando, al final va el nombre del server, en este caso es webserver, luego salimos y volvemos a ingresar

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

Antes de continuar se deben actualizar la lista de paquetes elegibles, con el siguiente comando se descargan los paquetes disponibles. Seguidamente con las últimas 4 líneas se instalan los paquetes necesarios

```
sudo apt-get update
```

```
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
```

```
vagrant@webserver:~$ sudo apt-get update
Get:1 https://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 https://security.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 https://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 https://deb.debian.org/debian bullseye-backports InRelease [49.0 kB]
Get:5 https://deb.debian.org/debian bullseye/main Sources [8637 kB]
Get:6 https://security.debian.org/debian-security bullseye-security/main Sources [201 kB]
Get:7 https://security.debian.org/debian-security bullseye-security/main amd64 Packages [245 kB]
Get:8 https://security.debian.org/debian-security bullseye-security/main Translation-en [161 kB]
Get:9 https://deb.debian.org/debian bullseye/main amd64 Packages [8183 kB]
Get:10 https://deb.debian.org/debian bullseye/main Translation-en [6240 kB]
Get:11 https://deb.debian.org/debian bullseye-updates/main Sources [5084 B]
Get:12 https://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.8 kB]
Get:13 https://deb.debian.org/debian bullseye-updates/main Translation-en [8129 B]
Get:14 https://deb.debian.org/debian bullseye-backports/main Sources [430 kB]
Get:15 https://deb.debian.org/debian bullseye-backports/main amd64 Packages [420 kB]
Get:16 https://deb.debian.org/debian bullseye-backports/main Translation-en [351 kB]
Fetched 25.2 MB in 2min 48s (150 kB/s)
Reading package lists... Done
```

Ahora se comprobará la dirección de IP desde la máquina anfitriona, se encuentra en el archivo Vagrantfile con el parámetro `private_network` y se le hace ping

```
code Vagrantfile
ping 192.168.33.10
```

Se modifica el archivo `hosts` de la máquina anfitriona desde un cmd y utilizando los siguientes comandos.

```
cd \
cd Windows\System32\drivers\etc
notepad hosts
```

dentro del archivo `hosts` se agrega la línea con la IP y el nombre del dominio, en este caso es `192.168.56.10 javier.isw811.xyz`

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1            localhost

127.0.0.1 license.sublimehq.com # SublimeText
127.0.0.1 www.sublimetext.com # SublimeText
|
192.168.33.10 javier.isw811.xyz
```

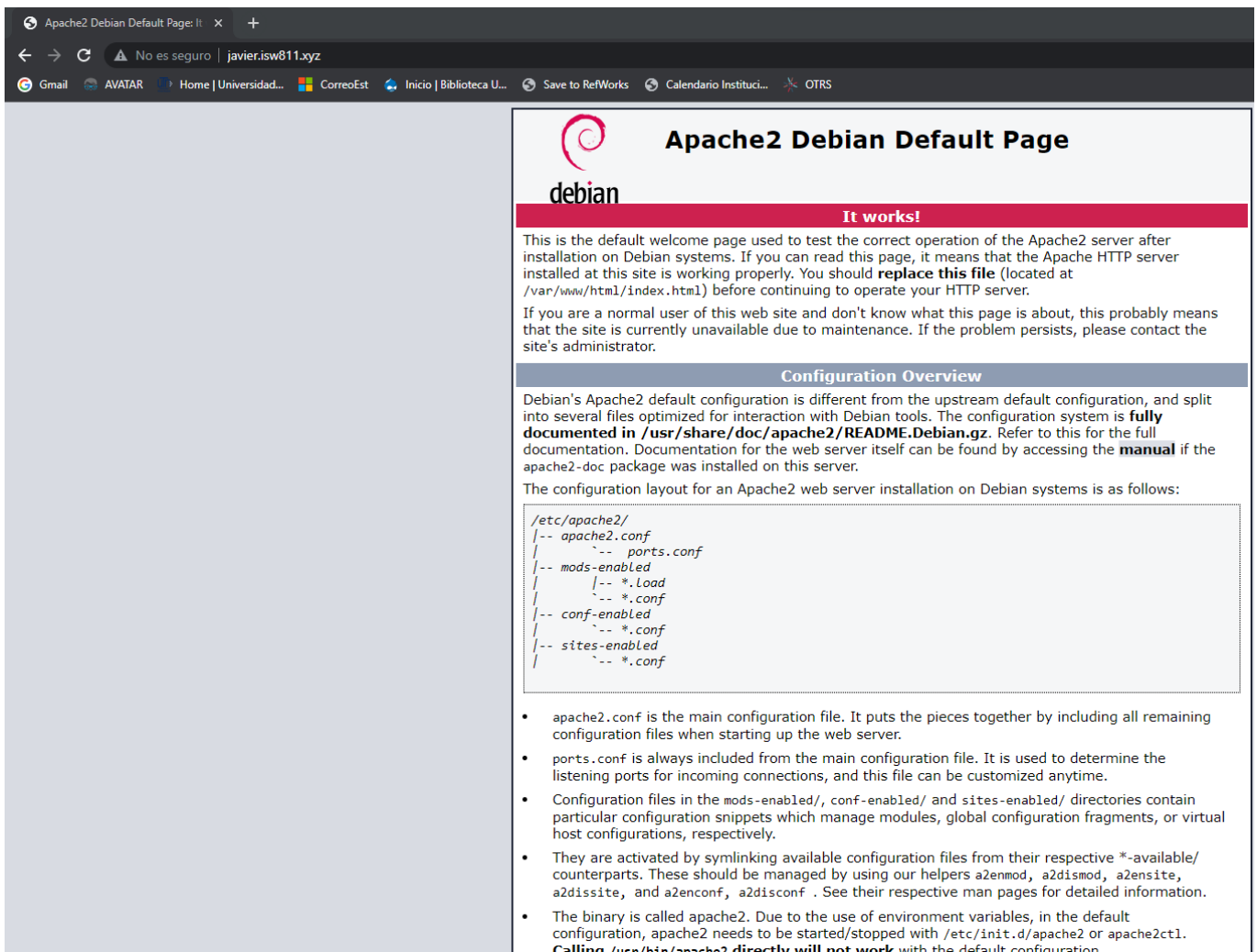
Ya guardado se hace ping para verificar que todo funcione y luego se puede acceder a la página desde un navegador ingresando a `javier.isw811.xyz`

```
ping javier.isw811.xyz
```

```
Estudiante@LAPSSC2357 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<1m TTL=64
Respuesta desde 192.168.33.10: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.33.10: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.33.10: bytes=32 tiempo=2ms 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 = 0ms, Máximo = 2ms, Media = 1ms
```



Apache2 Debian Default Page: It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly will not work with the default configuration.

Se activan los módulos para habilitar host virtuales y certificados ssl dentro de la máquina virtual

```
sudo a2enmod vhost_alias rewrite ssl
sudo systemctl restart apache2
```

Se crea una nueva carpeta en la máquina anfitriona para guardar el archivo html que usará nuestro sitio y se agrega el archivo html que podremos modificar a nuestro gusto, en mi caso usaré code para crear el archivo html y modificarlo, dentro de la carpeta sites

```
cd ISW811/VMs
mkdir sites
```

Ahora modificamos con code el archivo Vagrantfile y agregamos lo siguiente en la línea 47

```
cd cd ISW811/VMs/webserver
code Vagrantfile

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

Luego se reinicia la máquina vagrant

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

Para cada sitio que se de desea alojar se debe tener un archivo conf por lo que se va a crear una carpeta donde guardarlo y se procede a crearlo, luego lo abrimos con code y agregamos el texto que se encuentra abajo

```
mkdir confs
cd confs
touch javier.isw811.xyz.conf
code javier.isw811.xyz.conf
```

```
<VirtualHost *:80>
ServerAdmin webmaster@javier.isw811.xyz
ServerName javier.isw811.xyz
DirectoryIndex index.php index.html
DocumentRoot /home/vagrant/sites/javier.isw811.xyz
<Directory /home/vagrant/sites/javier.isw811.xyz>
DirectoryIndex index.php index.html
AllowOverride All
Require all granted
</Directory>
```

```
ErrorLog ${APACHE_LOG_DIR}/javier.isw811.xyz.error.log
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/javier.isw811.xyz.access.log combined
</VirtualHost>
```

Ahora en la máquina virtual se copia ese archivo conf recién creado y se pegará en la carpeta sites-available

```
sudo cp /vagrant/confs/mizaq.isw811.xyz.conf /etc/apache2/sites-available
```

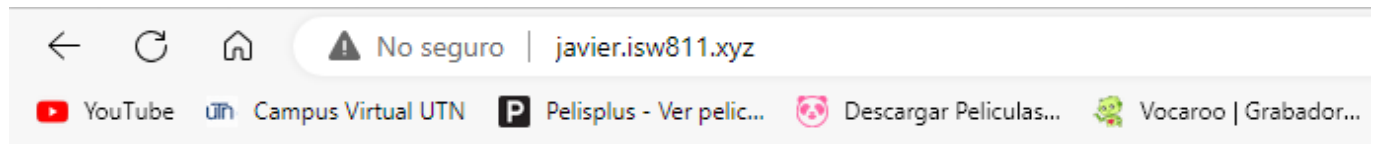
Ahora se debe reiniciar apache porque se agregó un nuevo archivo conf y tendremos una respuesta "syntax ok"

```
sudo apache2ctl -t
```

```
vagrant@webserver:~$ sudo apache2ctl -t
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.2.
Please set the 'ServerName' directive globally to suppress this message
Syntax OK
vagrant@webserver:~$ |
```

Finalmente habilitamos el nuevo sitio, reiniciamos apache y podemos proceder a probarlo ingresando desde el navegador a nuestro sitio y deberíamos ver el html ya aplicado

```
sudo apache2ctl -t
sudo a2ensite mizaq.isw811.xyz.conf
sudo systemctl restart apache2.service
```



Sito web de Javier

Arriba el comunismo!!!!



Para finalizar se apaga la máquina y se procede a preparar el archivo .tar.gz para entregar