

# WORKSHOP 2

---

## Paso 1: Iniciar el servidor web (Vagrant)

- Ingresamos a la carpeta donde tenemos el webserver configurado.
- Ejecutamos el siguiente comando para iniciar la máquina virtual:

```
vagrant up
```

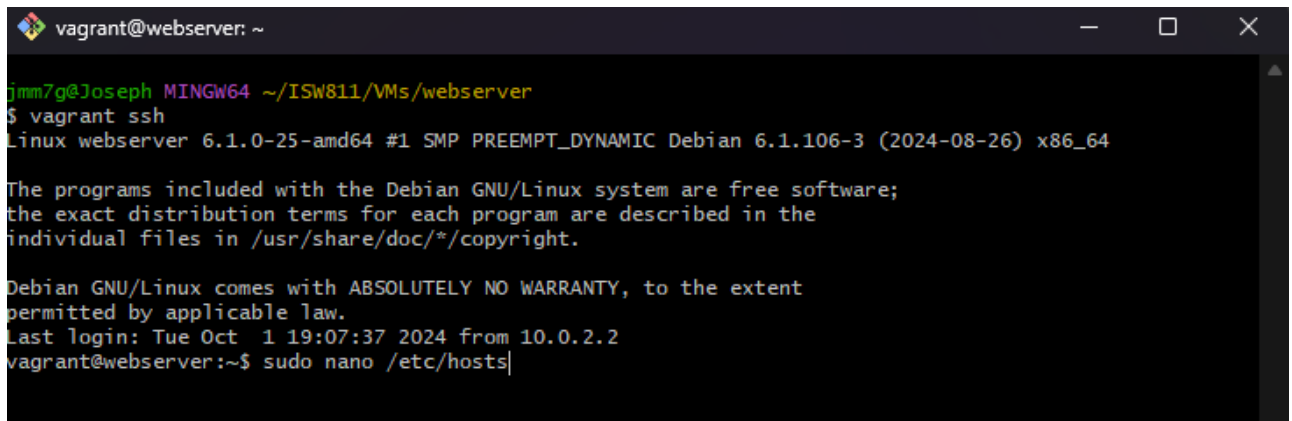
- Luego, accedemos a la terminal de la máquina virtual con:

```
vagrant ssh
```

---

## Paso 2: Cambiar el **hostname** de la máquina virtual

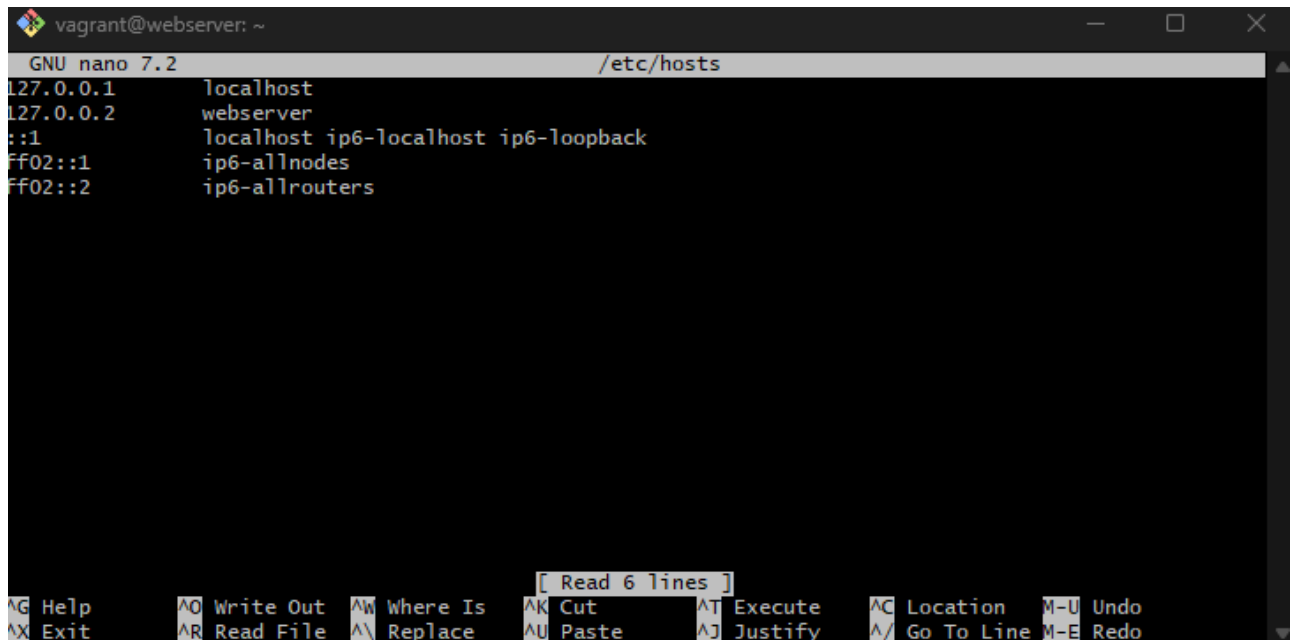
- Cambiamos el hostname usando los comandos correspondientes.
- Para confirmar el cambio, mostramos la siguiente captura:



```
vagrant@webserver: ~  
jmm7g@Joseph MINGW64 ~/ISW811/VMs/webserver  
$ vagrant ssh  
Linux webserver 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Oct 1 19:07:37 2024 from 10.0.2.2  
vagrant@webserver:~$ sudo nano /etc/hosts|
```

## Paso 3: Editar el archivo **hosts**

- Editamos el archivo **hosts** en la máquina virtual para asignarle el nombre **webserver**.
- Puedes ver el proceso en esta captura:

A screenshot of a terminal window titled 'vagrant@webserver: ~'. The terminal shows the contents of the /etc/hosts file being edited with nano 7.2. The file content is: 127.0.0.1 localhost, 127.0.0.2 webserver, ::1 localhost ip6-localhost ip6-loopback, ff02::1 ip6-allnodes, and ff02::2 ip6-allrouters. The nano editor's status bar at the bottom shows '[ Read 6 lines ]' and various keyboard shortcuts like ^G Help, ^O Write Out, etc.

```
vagrant@webserver: ~  
GNU nano 7.2 /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  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

---

## Paso 4: Actualizar la lista de paquetes

- Actualizamos la lista de paquetes ejecutando el siguiente comando:

```
sudo apt update
```

```
vagrant@webserver:~$ sudo apt-get update
```

---

## Paso 5: Instalar Vim, cURL, Apache2, MySQL y PHP

- Instalamos todo el software esencial con un solo comando:

```
sudo apt install vim curl apache2 mysql-server php libapache2-mod-php php-mysql
```

```
vagrant@webserver:~$ sudo apt-get install vim vim-nox curl \  
apache2 mariadb-server mariadb-client \  
php8.2 php8.2-curl php8.2-bcmath php8.2-mysql \  
php8.2-mcrypt php8.2-xml php8.2-zip php8.2-mbstring
```

---

## Paso 6: Editar el archivo **hosts** en Windows

- Abrimos la terminal **cmd** en modo administrador para tener permisos.
- Editamos el archivo **hosts** en Windows con el siguiente contenido:

```
192.168.56.10 joseph.isw811.xyz
```

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>cd drivers
C:\Windows\System32\drivers>cd etc
C:\Windows\System32\drivers\etc>notepad hosts
```

---

## Paso 7: Acceder al dominio en el navegador

- Abrimos un navegador en **modo incógnito** y accedemos al siguiente URL:

```
http://joseph.isw811.xyz
```

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1            localhost
127.0.0.1       iswutn.com
192.168.56.10   joseph.isw811.xyz
```

---

## Paso 8: Habilitar módulos de Apache para hosts virtuales

- Habilitamos los módulos necesarios para soportar **hosts virtuales**:

```
sudo a2enmod rewrite
```

```
vagrant@webserver:~$ sudo a2enmod vhost_alias rewrite ssl
vagrant@webserver:~$ sudo systemctl restart apache2
```

---

## Paso 9: Sincronizar una carpeta local

- Creamos una carpeta local en nuestra máquina y la sincronizamos con la carpeta `/home/vagrant/sites` en la máquina virtual.

```
45 # argument is a set of non-required options.
46 config.vm.synced_folder "sites/", "/home/vagrant/sites", owner: "www-data", group: "www-data"
47
```

---

## Paso 10: Reiniciar la máquina virtual

- Luego de realizar los cambios necesarios, reiniciamos la máquina virtual con:

```
vagrant reload
```

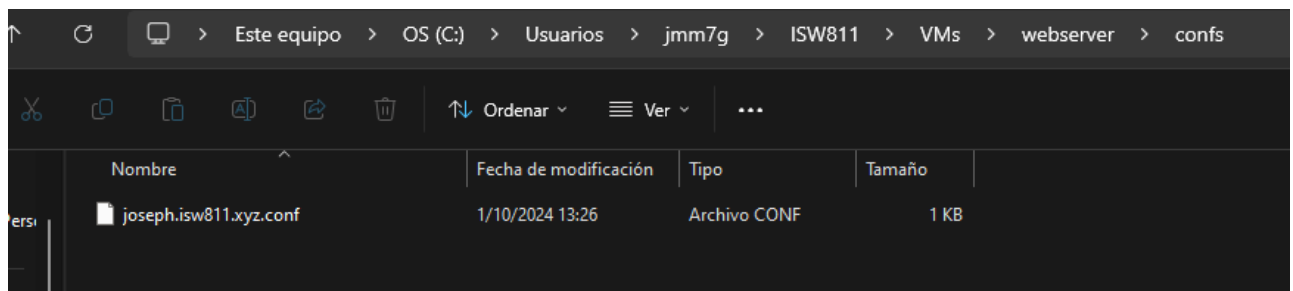
```
jmm7g@Joseph MINGW64 ~/ISW811/VMs/webserver
$ vagrant halt
```

---

## Paso 11: Crear un archivo `.conf` para el sitio web

- Creamos un archivo de configuración `.conf` para el sitio `joseph.isw811.xyz`:

```
<VirtualHost *:80>
    ServerName joseph.isw811.xyz
    DocumentRoot /home/vagrant/sites/joseph.isw811.xyz
</VirtualHost>
```



---

## Paso 12: Copiar el archivo `.conf` a `sites-available`

- Copiamos el archivo de configuración a la carpeta `sites-available` de Apache:

```
vagrant@webserver:~$ sudo cp joseph.isw811.xyz.conf /etc/apache2/sites-available/
```

---

## Paso 13: Verificar la configuración

- Comprobamos que todo esté bien y que la salida sea `Syntax OK`:

```
vagrant@webserver:~$ sudo apache2ctl -t
```

- Si aparece la advertencia: **Could not reliably determine the server's fully qualified domain name**, ejecutamos el siguiente comando para solucionarlo:

```
vagrant@webserver:~$ echo "ServerName webserver" | sudo tee -a /etc/apache2/apache2.conf
```

- Ejecutamos el comando hasta que solo salga `Syntax OK`:

```
vagrant@webserver:~$ sudo a2ensite joseph.isw811.xyz.conf
```

---

## Paso 14: Reiniciar Apache

- Finalmente, reiniciamos Apache para que los cambios tomen efecto:

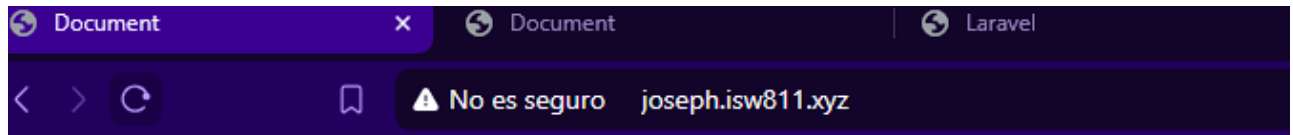
```
sudo systemctl restart apache2
```

```
vagrant@webserver:~$ sudo systemctl restart apache2.service
```

---

## Paso 15: Confirmación final

- Verificamos que todo funcione correctamente accediendo de nuevo al sitio:



## La pagina de Josepo

