

Práctica apache 2.4. Primeros pasos con Apache.

Ya tenemos instalado el apache ahora vamos a comprobar que funcione para ello podemos hacerlo de muchas formas en un primer lugar usando el comando `systemctl status apache2`, también podemos comprobar que se haya creado un fichero `index.html` en la siguiente ruta `/var/www/html/` para comprobar los puertos por los que está escuchando podemos hacer lo siguiente usando el comando `netstat` con la sentencia `ltn`. Para ello escribimos en la terminal `netstat -ltn`, si no lo tenemos instalado lo instalamos poniendo `sudo apt install net-tools`

```
jesusub@jesusub-VirtualBox:/var/www/html$ netstat -ltn
Conexiones activas de Internet (solo servidores)
Proto Recib Envíad Dirección local Dirección remota Estado
tcp 0 0 127.0.0.1:631 0.0.0.0:* ESCUCHAR
tcp 0 0 0.0.0.0:22 0.0.0.0:* ESCUCHAR
tcp 0 0 127.0.0.53:53 0.0.0.0:* ESCUCHAR
tcp6 0 0 :::631 :::* ESCUCHAR
tcp6 0 0 :::82 :::* ESCUCHAR
tcp6 0 0 :::22 :::* ESCUCHAR
```

631 es la dns y el 53 indica la dirección http en mi caso como había modificado el servidor mi apache está configurado escuchando por el puerto 82 normalmente os deberá salir a vosotros el puerto 80.

Nota: para cambiar el puerto de escucha sólo tenemos que usar un editor y modificar el fichero `ports.conf` de `apache2` con el comando por ejemplo `"nano /etc/apache2/ports.conf"` y cambiar donde ponga `NameVirtualHost *:80` `Listen 80` por el valor 82.

Una vez hecho esto vamos a configurar el cortafuegos por defecto de apache 2.4. Ufw es el cortafuego y si ponemos el siguiente comando nos pondrá una lista con la opciones posibles, (dependiendo distribución pueden variar los perfiles posibles).

sudo ufw app list

Da tres perfiles:

apache (puerto 80) tráfico sin cifrar

apache full (puerto 80 y el 443) el 443 es el tráfico cifrado

apache secure (puerto 443) abre sólo el cifrado para el tráfico

cups (un sistema de impresión en red) <https://ubuntu.com/server/docs/service-cups>

En esta práctica nos vamos a centrar en la primera opción y vamos a usar el comando **sudo ufw allow apache**. Una vez ejecutado vamos a comprobar su estado para ver si está bien con el comando **sudo ufw status** y si todo está bien debería salir "estado activo" si queremos ver lo que está actualmente bloqueado o permitido, podemos usar el verboso parámetro al correr `ufw estado` como se muestra a continuación **sudo ufw status verbose**. Otros comandos que debemos saber son:

- `sudo ufw disable` desactiva ufw
- `sudo ufw enable` activa ufw

Ahora nos permite el tráfico por el puerto 80 `sudo`

vamos a ver el nombre de nuestro
puerto host usamos `hostname -I`

```
jesusub@jesusub-VirtualBox:~$ hostname -I
192.168.1.133
```

Os dejo este enlace con otras funciones que podemos hacer con el cortafuegos ufw

<https://www.linuxparty.es/57-seguridad/10410-como-configurar-el-firewall-ufw-en-ubuntu-y-debian.html>

Recordad que el apache se activa nada más arrancar la máquina para ver su estado ponemos `systemctl status apache2`, para pararlo `sudo systemctl stop apache2` y luego lo comprobamos con `status` y para activarlo `sudo systemctl start apache2` para actualizar y registrar los cambios tendremos que usar el comando `restart` quedando así el siguiente comando `sudo systemctl restart apache2`.

Si queremos que la próxima vez que arranquemos la máquina no se arranque de forma automática nuestro servidor web usamos el comando `sudo systemctl disable apache2` y para volver a que funcione como siempre y que se active cuando se inicie la máquina virtual usamos el `sudo systemctl enable apache2`

Ahora vamos a instalar java para ello primero vamos a ver si lo tenemos instalado

`java -version` (indica la versión de java) si lo teneis instalado os saldrá vuestra versión instalada

```
jesusub@jesusub-VirtualBox:~$ java -version
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

`sudo apt-get update` se actualiza las cabeceras de los repositorios

`sudo apt-get install default-jdk`

```
jesusub@jesusub-VirtualBox:~$ sudo apt-get install default-jdk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
default-jdk ya está en su versión más reciente (2:1.11-72build2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
```

Ahora vamos a crear una variable de entorno para hacer que la ruta de java siempre sea la misma por si me muevo de ordenador o cambio de servidor no vamos a dar la ruta absoluta sino una relativa.cd ..

Para lo cual escribimos `JAVA_HOME=/usr/lib/jvm/default-java` y ahora para comprobar que esté bien hecha ponemos `cd /usr/lib/jvm/default-java` en caso que no os salga ponemos el siguiente comando que nos tiene que dar la ruta echo `$JAVA_HOME` y si hacemos un listado dentro de la ruta de default-java con el comando `ls` nos debería de salir algo así

```
jesusub@jesusub-VirtualBox:/usr/lib/jvm/default-java$ ls
bin  conf  docs  include  jmods  legal  lib  man  release
jesusub@jesusub-VirtualBox:/usr/lib/jvm/default-java$
```

Configurar un Host Virtual

Ahora vamos a configurar un host virtual. La idea es que montéis un host para subir una página nos vamos al directorio raíz ponemos el comando `ls` y nos metemos en la siguiente ruta `cd /var/www/html/` y hacemos un `ls` y nos debería salir un fichero llamado `index.html`

Vamos a crear una nueva carpeta `public_html` para que agregue mis nuevos archivos para cada nuevo sitios que creamos debo crear esa carpeta para ello usamos el siguiente comando **`sudo mkdir -p /var/www/despliegue.com/public_html`** pongo el `-p` para que cree también la carpeta `despliegue.com`

Ahora vamos a conceder permisos para que lo que hagamos sea propiedad de nuestro usuario. Para lo que vamos a hacer es usar el siguiente comando que me cambia los permisos al usuario actual `sudo chown -R $USER:$USER /var/www/despliegue.com/public_html`

vamos a modificar nuestros permisos para garantizar que se permita la lectura al directorio web general y eso lo vamos a hacer usando el comando `sudo chmod -R 755 /var/www` y vamos a comprobando los permisos haciendo un listado con el siguiente comando `ls -l`

```
jesusub@jesusub-VirtualBox:/var/www$ ls -l
total 8
drwxr-xr-x 3 root root 4096 oct 20 01:24 despliegue.com
drwxr-xr-x 2 root root 4096 oct 9 12:17 html
```

Ahora nuestro usuario puede crear contenido en dichas carpetas. Ahora vamos a crear una página (yo he puesto una muy simple podéis meter la vuestra si tenéis alguna creada)

`nano /var/www/despliegue.com/public_html/index.html`

```
GNU nano 6.2 /var/www/despliegue.com/public_html/index.html *
<html>
  <head>
    <title>Bienvenidos a Despliegue de Aplicaciones Web!</title>
  </head>
  <body>
    <h1>¡Funciona! ¡El host virtual despliegue.com funciona!</h1>
  </body>
</html>
```

Salimos y guardamos y volvemos a entrar para verificar que se haya modificado. Ahora vamos a configurar este host virtual. Apache por defecto tiene un archivo creado 000-default.conf. Vamos a aprovecharlo lo vamos a coger de base y lo vamos a modificarlo

Por lo tanto lo primero que vamos a hacer es copiar un fichero que está creado por defecto en apache y después lo vamos a modificar con los siguientes comandos

`sudo cp /etc/apache2/site-available/000-default.conf /etc/apache2/site-available/despliegue.com.conf`

`sudo nano /etc/apache2/site-available/despliegue.com.conf`

Cuando lo abramos nos va a salir lo siguiente:

```
GNU nano 6.2 /etc/apache2/sites-available/despliegue.com.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Y ahora tenemos que modificar unas cuantas cosas, en nuestro caso algunas más ya que el dominio que hemos puesto no es nuestro y no lo tenemos comprado por lo que vamos a tener que añadir las siguientes cosas:

```
ServerAdmin admin@despliegue.com
ServerName despliegue.com
ServerAlias www.despliegue.com
DocumentRoot /var/www/despliegue.com/public_html
```


- En primer lugar tiene que llegarle los mensaje al email de mi administrador por eso modificamos esa primera linea
- En segundo lugar tenemos que pensar que ese dominio no es nuestros por lo que tenemos que poner un ServerName (dominio base) que tiene que coincidir con el nombre que hemos puesto a nuestro host y el Alias es para que todo los que nos llegue nombres adicionales deberian de coincidir con nuestro dominio base
- Y por último tenemos que cambiar nuestro documentRoot ya que el nuestro es distinto

Guardamos las modificaciones y ahora vamos habilitar los sitios web disponibles con el siguiente comando y después vamos a recargar apache para que coja los cambios

sudo a2ensite despliegue.com.conf

```
jesusub@jesusub-VirtualBox:/var/www$ sudo a2ensite despliegue.com.conf
[sudo] contraseña para jesusub:
Enabling site despliegue.com.
To activate the new configuration, you need to run:
systemctl reload apache2
```

systemctl reload apache2

Ahora tenemos que desactivar el otro sitio que teníamos por defecto con el siguiente comando para evitar error usando el comando **sudo a2dissite 000-default.conf**

Ahora como no es mi sitio comprado lo que voy a tener que hacer es definir un host virtual para que mi equipo al menos lo reconozca. En caso que fuera mio ya no tendría que hacer nada mas. Por lo que voy a tener que crear un fichero para que si yo pongo www.despliegue.com me entre en esa página mía y no la busque en internet por lo que usando un editor voy a editar nuestro host con el siguiente comando **sudo nano /etc/hosts** y vamos a añadir 192.168.1.133 despliegue.com

```

┌─[+]─┐ jesusub@jesusub-VirtualBox: /var/www
└─┬───┘ GNU nano 6.2 /etc/hosts *
  S127.0.0.1    localhost
  127.0.1.1    jesusub-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

```

y se quedaría asi

```

┌─┬───┘ GNU nano 6.2 /etc/hosts
  S127.0.0.1    localhost
  127.0.1.1    jesusub-VirtualBox
  192.168.1.133 despliegue.com

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

```

Y si ponemos en el navegador despliegue.com nos debe salir la siguiente página

