



UD4. ADMINISTRACIÓN DE SERVIDORES WEB.

- Introducción. Glosario de términos.
- Configuración avanzada del servidor web (Apache2).
- Hosts virtuales basados en nombres.



4.1. Introducción - Glosario.

Core: Es un software que constituye la parte fundamental de un servicio o sistema operativo y que se ejecuta en modo administrador, también llamado núcleo o kernel.

Directiva: Es una palabra reservada dentro de un fichero de configuración que va acompañada de parámetros para realizar una función específica.

Parámetro: Es una palabra o una opción que personaliza un fichero de configuración de cualquier programa o servicio para realizar una función específica.

Pid (*process identifier*): Identificador de un proceso en un sistema operativo.



4.2. Configuración avanzada del servidor web.

En el primer trimestre se explicó la instalación del servidor **Apache 2.4.** y algunas nociones básicas.

La configuración de **Apache** se realiza a través de ficheros de texto o ficheros.conf, los cuales contienen directivas que se pueden configurar fácilmente. Una vez configurada la directiva correspondiente es necesario reiniciar el servicio de **Apache** y los cambios realizados se llevarán a efecto. Se pueden agrupar en **cinco categorías**, que se muestran a continuación:



4.2. Configuración avanzada del servidor web.

1. **Directivas de control de funcionamiento:** permiten controlar el flujo de trabajo del servidor de forma global.
2. **Parámetros del servidor web:** son parámetros globales para todos los hosts virtuales configurados. Se puede especificar algún parámetro en particular en cada host virtual.
3. **Configuración de hosts virtuales:** permite crear nombres de dominio diferentes para acceder al servidor desde IP diferentes y puertos diferentes.
4. **Instalación y configuración de módulos:** una de las partes más importantes de Apache, que destaca por su modularidad.
5. **Monitorización y rendimiento del servidor:** esta parte se encarga de controlar los accesos al servidor, hosts virtuales, y monitoriza las conexiones y demás problemas o fallos del servidor.



4.2. Configuración avanzada del servidor web.

4.2.1. Directivas de control del servidor Apache.

- ***ServerRoot***. Es la ruta de directorios raíz a partir de la cual cuelga toda la configuración del servidor Apache.
- ***ServerName***. Es el nombre por el cual se identifica Apache, es necesario definirlo en el fichero de hosts para que resuelva la IP o crear un registro DNS en el servicio DNS.
- ***DefaultRuntimeDir*** `${APACHE_RUN_DIR}`. Es el directorio donde el servidor apache crea ficheros en tiempo de ejecución, como bloqueos, memoria compartida, etc. Para ello existe un fichero denominado `/etc/apache2/envvars`, donde se crean todas las variables de entorno que empiezan por \$.



4.2. Configuración avanzada del servidor web.

4.2.1. Directivas de control del servidor Apache.

- **PidFile** `${APACHE_PID_FILE}`. Es la directiva encargada de crear el id del proceso del servidor Apache y apunta a la variable de entorno **APACHE_PID_FILE**.
- **Timeout** `300`. Es el tiempo en segundos que el servidor espera antes de enviar un fallo a una solicitud- Lo ideal es dejarlo como viene configurado, aunque dependiendo de las necesidades se puede aumentar o disminuir.
- **KeepAlive On**. Permite habilitar las conexiones persistentes mediante el protocolo HTTP. Permite enviar varias solicitudes en la misma conexión TCP.
- **MaxKeepAliveRequests** `100`. Es el número de solicitudes que se permiten en una sola conexión.



4.2. Configuración avanzada del servidor web.

4.2.1. Directivas de control del servidor Apache.

- ***KeepAliveTimeout 5.*** Es el número de segundos que espera entre solicitud y solicitud de la misma conexión del mismo cliente.
- ***User \${APACHE_RUN_USER}.*** Es el usuario que ejecutará Apache que debe tener permisos de administrador o root, se basa en el fichero envvars para coger el valor de la variable ***APACHE_RUN_USER***.
- ***Group \${APACHE_RUN_GROUP}.*** Es el grupo al que pertenece el usuario que ejecutará Apache, se basa en el fichero envvars para coger el valor de la variable ***APACHE_RUN_GROUP***.
- ***HostnameLookups Off.*** Esta directiva tiene tres posibles valores: On, Off y Double. Si se configura a On el servidor Apache resuelve automáticamente las direcciones IP para cada conexión. Este proceso sobrecarga el procesamiento del servidor, ya que es necesario consultar un servicio DNS para resolver cada petición. Si está en Double realiza las búsquedas directas e inversas, añade más sobrecarga. Por lo que su valor por defecto e ideal es Off.



4.2. Configuración avanzada del servidor web.

4.2.1. Directivas de control del servidor Apache.

- **ErrorLog** *\${APACHE_LOG_DIR}/error.log*. Es el registro de log de los errores del servidor. Por defecto, la ruta donde se almacena en */var/log/apache2*, se puede modificar en el fichero de variables de entorno */etc/apache2/envvars*.
- **LogLevel** *warn*. Establece los detalles que tendrá el log de errores del servidor. Los diferentes valores están ordenados por nivel de detalle desde el menor detalle hasta los más detallados. El orden es *emerg*, *alert*, *crit*, *error*, *warn*, *notice*, *info* o *debug*. Al menos es necesario configurar el valor de *crit*, de ahí para arriba.
- **IncludeOptional** *mods-enabled/*.load*. Incluye el fichero de módulos de configuración activados.



4.2. Configuración avanzada del servidor web.

4.2.1. Directivas de control del servidor Apache.

- ***IncludeOptional mods-enabled/*.conf***. Incluye el fichero de módulos de configuración habilitados.
- ***Include ports.conf***. Incluye el fichero de configuración de los puertos de escucha de los hosts virtuales y del servidor.
- ***AccessFileName .htaccess***. Directiva que permite dar acceso a un directorio configurando tal fichero.
- ***IncludeOptional conf-enabled/*.conf***. Incluye el fichero de configuraciones habilitadas para el servidor Apache.
- ***IncludeOptional sites-enabled/*.conf***. Incluye el fichero de sitios o hosts virtuales habilitados para el servidor Apache.



4.2. Configuración avanzada del servidor web.

4.2.2. Parámetros del servidor.

El objetivo de esta sección es configurar los valores del servidor principal, que permitirán a los hosts virtuales poseer una configuración por defecto. Las directivas son las siguientes:

- **ServerAdmin.** Este parámetro sirve para suministrar una dirección de email del administrador del sistema al usuario que se conecta mediante un host virtual y será mostrada en las páginas de error generadas. Permitirá ponerse en contacto con el administrador del sistema.
- **ServerName.** Es el nombre del servidor o una dirección IP válida y un puerto opcional. Esta directiva se puede usar tanto en el fichero principal como en los hosts virtuales. La nomenclatura sería *ServerName domain-name | IP[:port]*.



4.2. Configuración avanzada del servidor web.

4.2.2. Parámetros del servidor.

- ***ServerAlias***. Esta directiva permite establecer nombres adicionales y complementarios al host declarado en la directiva anterior. Su nomenclatura sería *ServerAlias hostname [hostname]...*
- ***DocumentRoot***. Esta directiva establece el directorio a partir del cual es visible la ruta desde el servidor web. La nomenclatura es *DocumentRoot directory-path*.
- ***DirectoryIndex***. Es la lista de recursos que se van a buscar cuando un cliente solicite un directorio. La nomenclatura es *DirectoryIndex disabled | local-url [local-url]...* Permite declarar varios tipos de ficheros como *index.html*, *index.jsp*, etc.



4.2. Configuración avanzada del servidor web.

4.2.2. Parámetros del servidor.

- **ErrorDcoument.** Es la directiva que permite establecer una página de error para que la visualice el cliente en caso de no poder atender la petición por la causa que sea. La sintaxis sería `ErrorDocument error-code document`.
- **AccesFileName.** Es la directiva que permite el acceso limitado a determinados directorios.



REALIZAR TAREAS 4.2 y 4.3.

A partir de las diapositivas anteriores, realizar las **tareas 4.2. y 4.3.** propuestas en el classroom.



4.2. Configuración avanzada del servidor web.

Debemos de tener claro que el archivo más importante y principal para configurar el servidor **Apache** es el *“**apache2.conf**”* y se encuentra en la ruta *“**etc/apache2**”*.

.htaccess es un archivo de texto oculto, que permite personalizar el funcionamiento del servidor Apache, sin necesidad de tocar el archivo principal (**apache2.conf**).

- Impedir el listado de los archivos de un directorio.
- Redireccionar el tráfico web.
- Personalizar las páginas de error.
- Restringir el acceso a determinados archivos.
- Impedir el acceso a determinadas IPs o rangos de IP.



4.2.1. Estructura Archivos “/etc/apache2”

- > *apache2.conf*: es el fichero de configuración principal del servidor **apache2**. Contiene las variables globales. Cualquier cambio en este fichero implicaría reiniciar el servicio.
- > *conf-available*: este directorio contiene configuraciones adicionales que están asociadas a un módulo en particular. Las configuraciones no están activas.
- > *conf-enabled*: este directorio contiene enlaces al directorio anterior para poder activar las configuraciones que contiene.
- > *envvars*: es un fichero en el que se definen variables de entorno como `APACHE_RUN_USER`, `APACHE_RUN_GROUP`, etc., que en principio no es necesario modificar.



4.2.1. Estructura Archivos “/etc/apache2”

-> *magic*: es un fichero de configuración de tipo MIME, que permite configurar el tipo de medio del contenido que va a visualizar en el servidor web, puede ser del correo electrónico o de los servidores webs.

-> *mods-avaliabile*, *mods-enabled*: estos directorios son similares a los *sites*, pero para módulos que se pueden acoplar al servidor web.

-> *ports.conf*: este fichero permite configurar los puertos IP por los que escucha el servidor. El puerto por defecto es el 80.

-> *sites-avaliabile*: este directorio contiene los ficheros de los hosts virtuales definidos en el servidor, que pueden ser diferentes. Estos sitios están disponibles pero no activos.



4.2.1. Estructura Archivos “/etc/apache2”

-> *sites-enabled*: este directorio es similar al anterior pero las definiciones de hosts virtuales que se están usando, normalmente son enlaces simbólicos a los ficheros que se encuentran en *sites-available*. Por defecto, tiene el directorio */var/www/html*, que es donde se almacena la página principal de Apache.

La configuración de los ficheros *000-default.conf* y *apache2.conf*, son los que van a permitir el acceso de un usuario desde el exterior y siempre que esté autorizado en caso de que existiera un firewall.

Vamos a comentar el fichero *000-default.conf*, que es el fichero de *virtual host* por defecto:



4.2.2. Fichero “*000-default-conf*”

Este fichero se encuentra en la ruta: */etc/apache2/sites-available/*

Este fichero es un sitio virtual, que se configura en el servidor web, donde posee varias directivas pero la más importante es **DocumentRoot**.

Esta directiva permite configurar una ruta donde se ubicará el fichero *index.html*. Para ello se ha creado un host virtual que escucha por el puerto 80 con la directiva **<VirtualHost *:80>**.

Para habilitar el sitio es necesario usar el comando *a2ensite* y para deshabilitar es el comando *a2dissite*.

<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>



4.2.2. Directivas del fichero “*apache2.conf*”

Este fichero se encuentra en la ruta:
/etc/apache2/apache2.conf

El primer bloque (`<Directory/>`) deniega todos los accesos, esto se realiza como método de seguridad para que nadie acceda al sistema de ficheros del sistema operativo. Se queda por defecta esta parte, que tiene las directivas:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```



4.2.2. Directivas del fichero “*apache2.conf*”

- > *Options FollowSymLinks*: que permite los enlaces simbólicos a otros directorios.
- > *AllowOverride None*: ignora el fichero .htaccess para que no existan demasiadas llamadas a este fichero.
- > *Require all denied*: deniega el permiso a todo el sistema de ficheros.

Y en el bloque `<Directory /var/www>`, que tiene las siguientes directivas:

- > *Options Indexes FollowSymLinks*: Indexes permite visualización del directorio en caso de que no exista en el directorio el fichero index.html o los patrones declarados en el servidor Apache. FollowSymLinks que permite los enlaces simbólicos a otros directorios.



4.2.2. Directivas del fichero “*apache2.conf*”

- > *AllowOverride None*: ignora el fichero .htaccess para que no existan demasiadas llamadas a éste.
- > *Require all granted*: da permiso a todo el mundo y desde cualquier IP a directorio /var/www.

4.2.3. Práctica guiada.



Vamos a configurar un sitio distinto que se llamará *sitio.conf*.

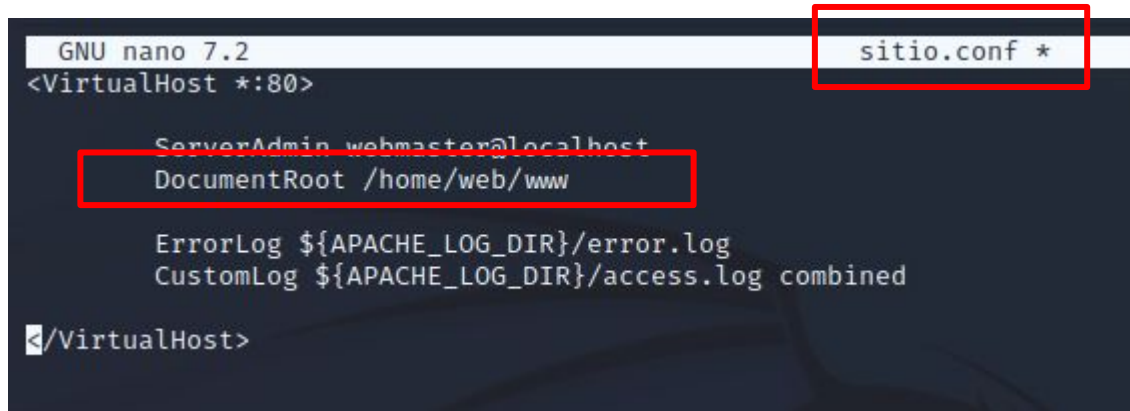
Este fichero va a tener una ruta distinta para ubicar el fichero *index.html* que será: */home/web/www*.

En primer lugar, deberemos crear la carpeta *web* y *www*, que será donde ubicaremos el fichero *index.html*.

El fichero *index.html*, podemos copiarlo del que tenemos en la ruta */var/www*, o hacer uno nuevo según nuestras necesidades.

4.2.3. Creación y configuración fichero “*sitio.conf*”.

Dentro de la ruta */etc/apache2/sites-available*, creamos el fichero *sitio.conf*.



```
GNU nano 7.2      sitio.conf *
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /home/web/www

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

</VirtualHost>
```


4.2.3. Edición del archivo “*apache2.conf*”.

Se puede comentar con # por si queremos volver al original.

Editamos el archivo y al final del último bloque
`<Directory /var/www/>` añadimos el nuevo
`<Directory /home/web/www/>`

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

#<Directory /var/www/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>

<Directory /home/web/www>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```



4.2.3. Deshabilitar y habilitar nuevo sitio.

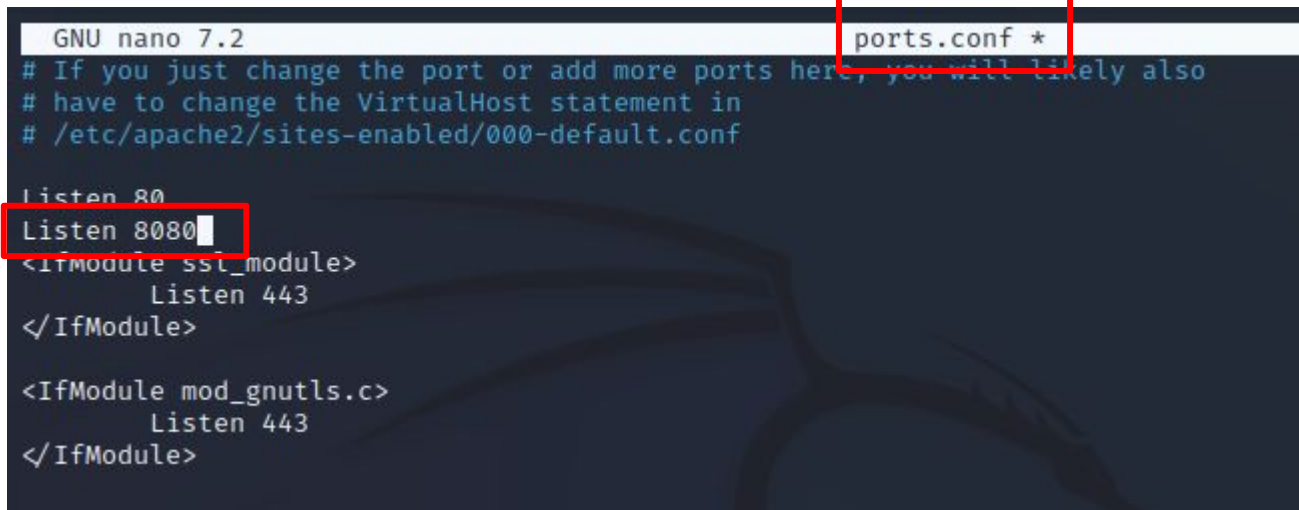
Una vez realizadas las modificaciones debemos recargar el servidor Apache, y deshabilitar el sitio que teníamos por defecto (*000-default.conf*) y habilitar el nuevo (*sitio.conf*). La secuencia de comandos sería la siguiente:

1. `systemctl reload apache2`
2. `a2dissite 000-default` (con esto deshabilitamos).
3. `systemctl reload apache2`
4. `a2ensite sitio` (con esto habilitamos).

Por último, vamos al navegador y probamos si todo funciona correctamente, escribiendo en el mismo: *http://localhost*.

4.2.3. Configuración puerto distinto.

Ahora vamos a configurar que el servidor escuche por otro puerto. Para ellos se configurará en el fichero *ports.conf*, por ejemplo, el puerto 8080.



```
GNU nano 7.2 ports.conf *
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

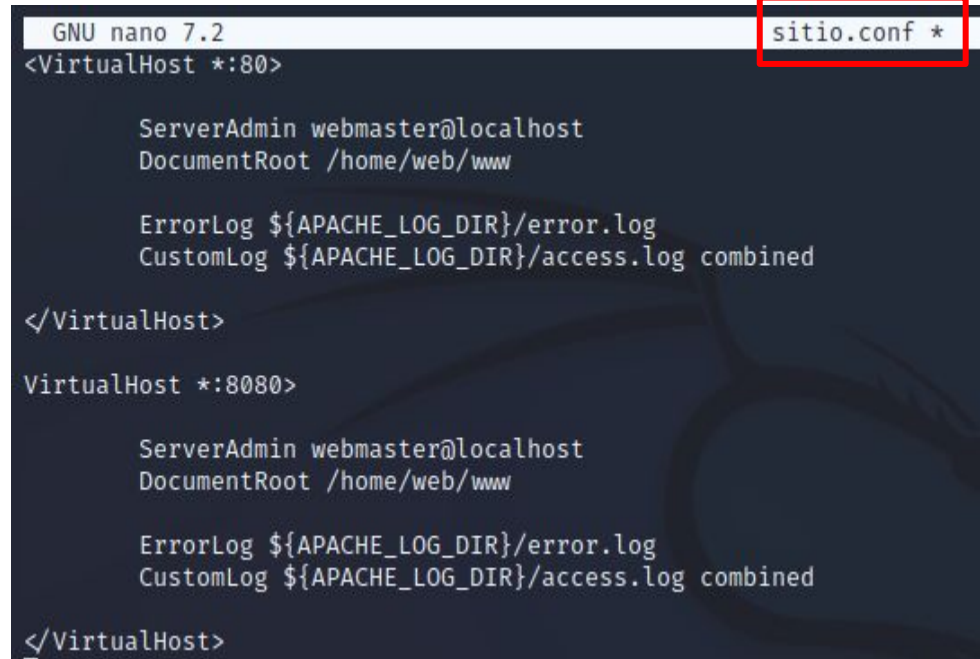
Listen 80
Listen 8080
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

4.2.3. Configuración puerto distinto.

Ahora deberemos modificar el fichero de *sitio.conf* para indicarle que escuche también por el **puerto 8080**, de lo contrario el servidor web daría un error de acceso prohibido. De esta forma, el **servidor** escucharía por los **dos puertos**.

Se copia el virtual host del puerto 80 y solo se cambia el puerto a 8080



```
GNU nano 7.2 sitio.conf *
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /home/web/www

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

</VirtualHost>

VirtualHost *:8080>

    ServerAdmin webmaster@localhost
    DocumentRoot /home/web/www

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

</VirtualHost>
```

TAREA 4.3. Configuración avanzada servidor Apache2.

Ahora ya puedes hacer la tarea 4.3 del classroom.



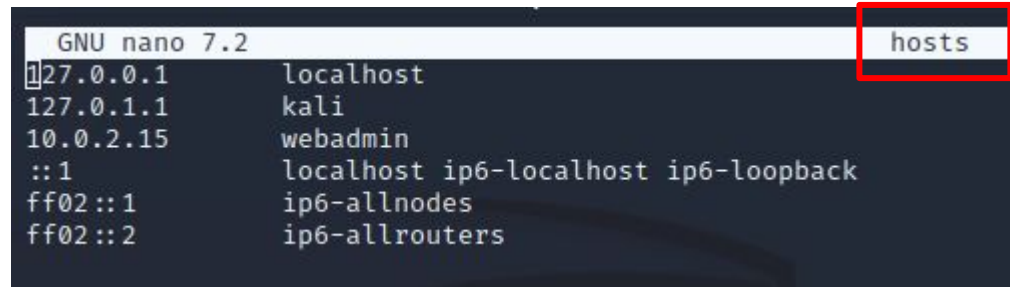


4.3. Hosts virtuales. Creación, configuración y utilización.

Hasta ahora se ha visto una conexión mediante un sitio o host virtual por el **puerto 80** o **8080**. En esta sección se va a comentar cómo definir **varios nombres de dominio distintos**, que pueden ubicarse en **directorios diferentes** y en **puertos distintos** mediante la declaración de hosts virtuales basados en IP o en nombres o en servicios principales.

Para declarar estos hosts virtuales sería necesario modificar el fichero host para poder declarar el nombre en cuestión, o declararlo en el servidor DNS. En el fichero **/etc/hosts** sería de la siguiente forma:

4.3. Hosts virtuales. Creación, configuración y utilización.



A screenshot of a terminal window showing the /etc/hosts file being edited with the GNU nano 7.2 editor. The file contains several entries mapping IP addresses to hostnames. The word 'hosts' in the top right corner of the editor window is highlighted with a red rectangle.

```
GNU nano 7.2 hosts
127.0.0.1    localhost
127.0.1.1    kali
10.0.2.15    webadmin
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```



4.3. Hosts virtuales. Creación, configuración y utilización.

Por lo tanto, si se necesita que un **servidor web** atienda **peticiones de diferentes dominios**, la solución es el uso de hosts virtuales que puedan estar asociados a dominios diferentes. Con esta solución se abre un abanico de posibilidades, las más usadas son las siguientes:

- **Host virtual basado en nombre:** un servidor web que tiene una sola IP, y esta IP tiene asociados diferentes dominios. El contenido es el mismo, a nivel de páginas web, sin embargo, son nombres de acceso diferentes.
- **Host virtual basado en IP:** otra opción sería un servidor web que tiene diferentes IP públicas, ya sean físicas o virtuales. Cada dirección web está asociada a una IP.
- **Host virtual mixto:** esta es menos usual pero también se puede dar, es una mezcla de las anteriores.



4.3. Hosts virtuales basados en nombre.

- Crear un directorio que se llame *empresa-daw* en la ruta */var/www*.
- Este directorio debe permitir el acceso a las URLs: *www.empresa-daw.com* y *www.empresa-daw.local*.
- Los logs se deben configurar de acceso y de error, que se denominan *empresa-daw-access.log* y *empresa-daw-error.log*, respectivamente. El log debe ser *combined*.

A continuación, se van a describir los pasos para llevar a cabo la configuración de un host virtual:

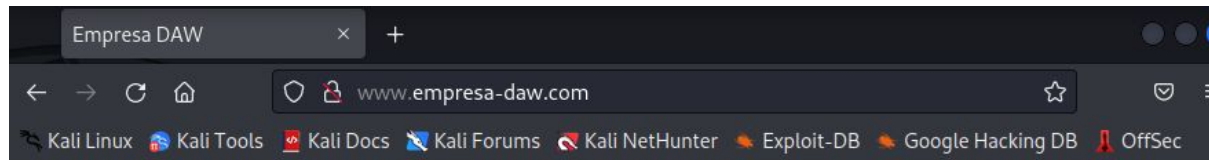
4.3. Hosts virtuales basados en nombre.

1.- Al principio se creará el directorio que pone en la ruta y se le dará permisos de lectura y ejecución, tal como se muestra en la imagen.

```
(root@kali)-[/var/www]
# ls -l
total 8
drwxr-xr-x 2 root root 4096 feb 17 07:48 empresa-daw
drwxr-xr-x 3 root root 4096 feb 15 05:09 html
```

4.3. Hosts virtuales basados en nombre.

2.- Seguidamente se creará el fichero index.html para identificar que está funcionando el **host virtual** en la ruta anteriormente creada. La página de inicio tendrá el siguiente aspecto:



SITIO WEB CREADO EN `/var/www/empresa-daw`

Autor: Gustavo Bustos

4.3. Hosts virtuales basados en nombre.

3.- Por otro lado, es necesario crear el fichero de configuración *empresa-daw.conf*:

```
GNU nano 7.2      empresa-daw.conf *
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/empresa-daw
    ServerName www.empresa-daw.com
    ServerAlias empresa-daw.local

    LogFormat "%h %l %u %t \ %r\" %>S " combined
    ErrorLog ${APACHE_LOG_DIR}/empresa-daw-error.log
    CustomLog ${APACHE_LOG_DIR}/empresa-daw-access.log combined

</VirtualHost>
```



4.3. Hosts virtuales basados en nombre.

La nomenclatura de VirtualHost es `<VirtualHost IP[:puerto] [IP][:puerto]...>` hay que poner mínimo una IP, que puede ser asterisco, que significa todas las IP. El puerto es opcional y se pueden poner más IP y puertos.

Es necesario, **recargar el servidor** después de cada **cambio en los archivos**, y después **habilitar** este sitio con el comando `#a2ensite empresa-daw`. Y **deshabilitar** mediante el comando `#a2dissite` el que tuviéramos antes. Ésto se puede comprobar en la carpeta de sitios habilitados (`/etc/apache2/sites-enabled`), aquí nos aparecen los sitios que tenemos habilitados actualmente.



4.3. Hosts virtuales basados en nombre.

Las directivas del archivo *empresa-daw.conf*, son las siguientes:

- **ServerAdmin**: email del administrador del servidor web.
- **DocumentRoot**: ruta donde se ubicará el fichero index.html.
- **ServerName**: nombre del sitio web que albergará la información.
- **ServerAlias**: alias del nombre del sitio web.
- **LogFormat**: formato del fichero de log.
- **ErrorLog**: log de errores del sitio web.
- **CustomLog**: log de acceso al sitio web.



4.3. Hosts virtuales basados en nombre.

4.- Es necesario también que en el fichero de configuración ***apache2.conf***, el servidor web dé permiso para leer en el directorio ***/var/www/empresa-daw/***. Esto se realiza con las líneas que se muestran:

4.3. Hosts virtuales basados en nombre.

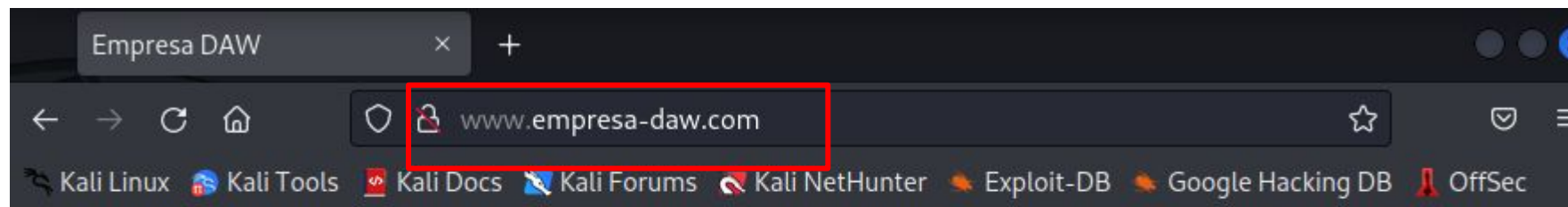
Debe coincidir con lo
indicado en el
DocumentRoot del archivo
empresa-daw.conf

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

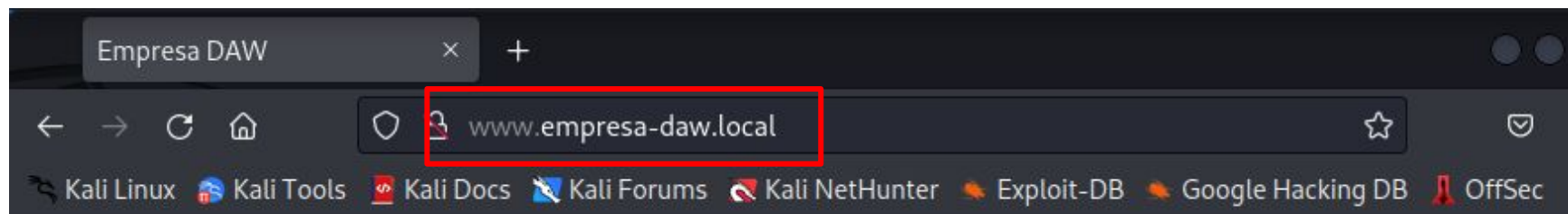
<Directory /var/www/empresa-daw/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Archivo “*apache2.conf*”.



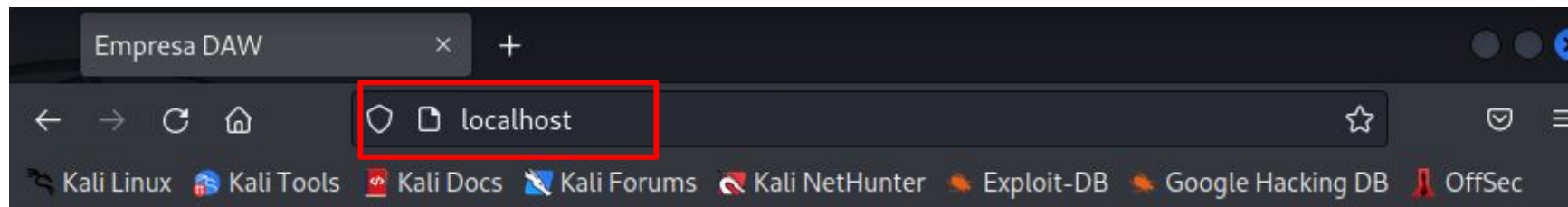
SITIO WEB CREADO EN `/var/www/empresa-daw`

Autor: Gustavo Bustos

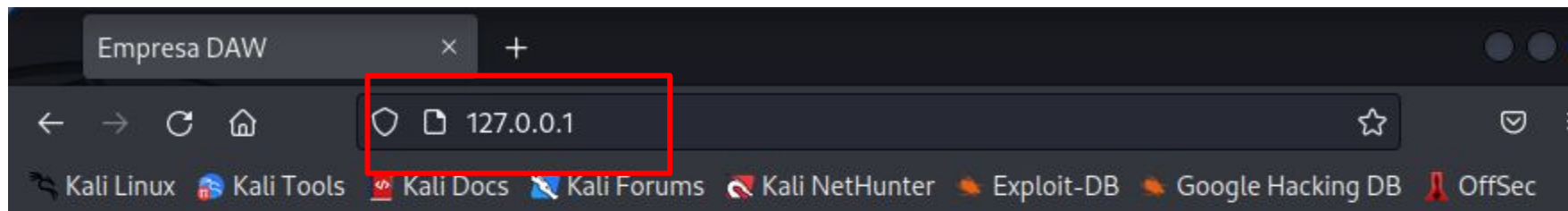


SITIO WEB CREADO EN /var/www/empresa-daw

Autor: Gustavo Bustos



SITIO WEB CREADO EN `/var/www/empresa-daw`



SITIO WEB CREADO EN `/var/www/empresa-daw`

Autor: Gustavo Bustos