

APACHE

TEMA 6 - GESTIÓN DE UN SERVIDOR WEB

APACHE WEB SERVER

Software libre (Apache License)

- Desarrollado y mantenido por la comunidad
- Supervisado por Apache Software Foundation

Disponible para múltiples plataformas

- Unix, Windows, MacOS, FreeBSD, Solaris, etc
- El servidor más popular desde 1996, actualmente rivaliza con *nginx*

Arquitectura modular:

- El core solo contiene la funcionalidad básica
- Se pueden instalar módulos que amplían la funcionalidad



Instalación en Ubuntu (Debian)

- Instalación

```
sudo apt install apache2
```

- Status

```
sudo systemctl status apache2
```

- Iniciar

```
sudo systemctl start apache2
```

- Para

```
sudo systemctl stop apache2
```

- Reiniciar

```
sudo systemctl restart apache2
```

Contenidos del directorio /etc/apache2

Una vez instalado, en el directorio /etc/apache2 tendremos:

```
ubuntu@ip-172-31-86-22:/etc/apache2$ ls -la
total 88
drwxr-xr-x  8 root root 4096 Nov 29 21:08 .
drwxr-xr-x 99 root root 4096 Nov 29 21:08 ..
-rw-r--r--  1 root root 7224 Jun 14 12:30 apache2.conf
drwxr-xr-x  2 root root 4096 Nov 29 21:08 conf-available
drwxr-xr-x  2 root root 4096 Nov 29 21:08 conf-enabled
-rw-r--r--  1 root root 1782 Mar 23  2022 envvars
-rw-r--r--  1 root root 31063 Mar 23  2022 magic
drwxr-xr-x  2 root root 12288 Nov 29 21:08 mods-available
drwxr-xr-x  2 root root 4096 Nov 29 21:08 mods-enabled
-rw-r--r--  1 root root  320 Mar 23  2022 ports.conf
drwxr-xr-x  2 root root 4096 Nov 29 21:08 sites-available
drwxr-xr-x  2 root root 4096 Nov 29 21:08 sites-enabled
```

Archivos del directorio /etc/apache2

- **apache2.conf**: archivo de configuración principal
- **envvars**: archivo de configuración de variables de entorno
- **magic**: contiene instrucciones para determinar el tipo MIME de un archivo leyendo sólo sus primeros bytes.
- **ports.conf**: contiene directivas que determinan los puertos TCP que Apache2 está escuchando

Carpetas del directorio `/etc/apache2`

- **`conf-available`**: contiene archivos de configuración adicionales
- **`conf-enabled`**: enlaza los archivos de configuración adicional del directorio *`conf-available`* que están activados
- **`mods-available`**: contiene archivos de configuración para cargar y configurar módulos
- **`mods-enabled`**: enlaza la configuración de los módulos activados. Los módulos se activan tras reiniciar Apache2
- **`sites-available`**: contiene archivos de configuración para Virtual Hosts, que permite alojar múltiples sitios web con diferentes configuraciones
- **`sites-enabled`**: enlaza la configuración de Virtual Hosts que están activados

Configurando Apache2

- Apache2 se ejecuta como usuario `www-data` y grupo `www-data`
- El directorio raíz del servidor web es `/var/www/html`
- Apache2 se configura colocando directivas en archivos de configuración de texto sin formato

Ejercicio

1. Crea una EC2 con **UBUNTU** e instala ***apache2***
2. Verifica la versión y comprueba que el servidor está encendido
3. Comprueba que se ha creado el usuario **`www-data`** y que existe el directorio **`/etc/apache2`** con los directorios que hemos visto
4. Mira los módulos disponibles y los que están activados. ¿Están todos los módulos activados?
5. ¿Cuántos sitios web están activados?
6. Visita la página http://<ip_pública_de_EC2>

Ejercicio

7. Antes de tocar nada, haz una copia de los siguientes archivos:
 - `/etc/apache2/apache2.conf`
 - `/etc/apache2/ports.conf`
 - `/etc/apache2/sites-available/000-default.conf`
 - `/etc/apache2/sites-available/default-ssl.conf`
8. Mira el contenido del archivo `apache2.conf`
 - ¿Qué ficheros se incluyen al final?
 - Este fichero se organiza en base a directivas. ¿Cuál es el valor y significado de las siguientes? Visita esta [página](#):
 - `KeepAlive`
 - `KeepAliveTimeout`
 - `MaxKeepAliveRequest`
 - `Include`

Ejercicio

9. Mira el contenido del archivo `ports.conf`. ¿Qué puertos se están escuchando?
10. Mira el contenido el archivo `sites-enabled/000-default.conf`. ¿En qué ruta se encuentran los archivos del sitio web?
 - Ve a esa ruta y comprueba su contenido
 - Modifica el contenido del fichero y comprueba que la web ha cambiado

Directivas

La configuración de Apache2 funciona en base a directivas:

- Una por línea: **nombre_directiva valor**
- El nombre es insensible a mayúsculas, aunque los valores pueden no serlo
- # inicia línea de comentario. No debe ponerse en una línea después de una directiva
- Líneas en blanco y espacios antes de la directiva se ignoran

Cada directiva puede afectar a uno o más de estos ámbitos:

- Servidor
- Host
- Directorios
- Archivos
- URLs

Contenedores de Directivas

Contenedores: tipo especial de directiva para agrupar directivas

- Contenedores evaluados en cada petición
 - <Directory>, <DirectoryMatch>, <Files>, <FilesMatch>, <Location>, <LocationMatch>, <Proxy>, <ProxyMatch>, <VirtualHost>, etc
- Contenedores evaluados solo al inicio del servidor
 - <IfDefine>, <IfModule>, <IfVersion>, etc

Ejemplo Contenedores

```
<Directory />
```

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
Require all denied
```

```
</Directory>
```

Directivas que afectan al directorio raíz

```
<Directory /var/www/>
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride None
```

```
Require all granted
```

```
</Directory>
```

Directivas que afectan al directorio /var/www

Ejemplo Contenedores

```
<Directory />
```

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
Require all denied
```

```
</Directory>
```

```
<Directory /var/www/>
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride None
```

```
Require all granted
```

```
</Directory>
```

Options

Configura una lista de características para el directorio

AllowOverride

Indica qué directivas pueden sobrescribir otros archivos de configuración

Require

Para conceder permisos sobre el directorio

<Directory>

Sintaxis: `<Directory path> ... </Directorio>`

- Donde `path` es la ruta completa del directorio

Las directivas incluidas se aplican:

- Al directorio referenciado por `path`
- A todos sus subdirectorios
- A todos los archivos contenidos en ellos

<Files>

Sintaxis: `<Files name_or_pattern> ... </Files>`

- Donde `name_or_pattern` es un nombre de fichero o un patrón con:
 - `?`: un solo carácter
 - `*`: varios caracteres
- Las directivas se aplican a todos los ficheros que se ajusten al nombre o patrón
- `<Files>` se puede anidar dentro de un `<Directory>` para acotar la zona de aplicación
- Ejemplo:
 - `<Files "imagen.jpg"> ... </Files>`
 - `<Files "*.jpg"> ... </Files>`

Otros contenedores

<Location */docs/es*>

- Afecta al ámbito de la URL indicada, sin incluir el dominio, query, etc
- No confundir con la ubicación en el sistema de archivos

<VirtualHost *addr[:port]*>

- Afecta a un sitio web
 - *Addr*: se refiere al nombre de dominio o ip
 - *[:port]*: opcionalmente se puede indicar el puerto

Orden de aplicación de directivas

Si varias secciones aplican a una petición, puede haber conflictos

Las directivas se aplican de más general a más particular

1. <Directory>, se aplican primero los directorios más cortos
2. <Files>
3. <Location>
4. En caso de empate, prevalecen las últimas secciones definidas

Las directivas dentro de <VirtualHost> se aplican después de las que están fuera de <VirtualHost>

- Esto permite a un virtual host sobrescribir la configuración general del servidor

Orden de aplicación. Ejemplo

```
<Location /> D </Location>

<Files f.html> C </Files>

<VirtualHost *>
  <Directory /a/b> B </Directory>
</VirtualHost>

<Directory /a/b> A </Directory>
```

Orden de aplicación:

A > B > C > D

```
<Location />
  Order deny,allow
  Allow from all
</Location>

<Directory />
  Order allow,deny
  Allow from all
  Deny from badguy.example.com
</Directory>
```

Location se evalúa al final y deja sin efecto lo que se defina en Directory