



UNIVERSIDAD
DE MÁLAGA



TEMA 3.2: APACHE, MYSQL Y PHP EN LINUX

LAMP

Servidor Web: Apache

- **El protocolo HTTP es el que se usa para ver en un navegador web (Mozilla, Internet Explorer, Chrome...) una serie de contenidos en HTML que se encuentran en una computadora remota, donde un demonio (programa que se ejecuta de forma continua) es el que se encarga de localizar y enviar dicho contenido.**
- **Estamos ante una arquitectura CLIENTE/SERVIDOR**
- **Un único SERVIDOR está atento a las solicitudes de muchos CLIENTES, de forma que son los CLIENTES quienes inicien el proceso de comunicación mediante una petición al SERVIDOR, la cual es atendida y procesada para generar una respuesta, que se envía al CLIENTE que inició el proceso.**

Servidor Web: Apache

- En entornos GNU Linux, la implementación de servidor HTTP más usada es Apache 2
- El servidor Apache está atento a cualquier paquete que llega al puerto 80 por el protocolo TCP (existe una versión segura de HTTP llamada HTTPS que opera a través del puerto 443 de TCP)
- Cuando se le pide un fichero HTML, el servidor lo recupera del disco duro y se lo envía por la red al cliente HTTP (navegador Web) que hizo la solicitud inicial.
- A veces el fichero HTML no está en el disco duro, sino que se genera dinámicamente (justo antes de enviarlo) siguiendo las instrucciones de un código (posiblemente en el lenguaje interpretado PHP) y recuperando información de algún sistema gestor de bases de datos

Servidor Web: Apache

- En el navegador se escribe una URL (Uniform Resource Locator) , que especifica una forma estandarizada la ubicación (localización) de un recurso en la red Internet.
- Tiene las siguientes componentes:
 - Servicio ofrecido = Protocolo (http, ftp, samba ...)
 - Nombre de la máquina en Internet: Hostname
 - Puerto por donde se atienden peticiones (opcional)
 - Directorio donde está el recurso (opcional)
 - Nombre del fichero/recurso (opcional)
 - Usuario / password (opcionales):

p.e. `ftp://michel:cl@v3@ftpserver.uma.es/fichero.txt`

PROTOCOLOS

file – acceso a fichero local	http – mostar ficheros remotos
ftp – subida/bajada ficheros	https – versión segura de http
smb – compartir impresoras, ficheros ... en una red Windows	

Servidor Web: Apache

- HTTP es un protocolo cliente/servidor “SIN ESTADO”:
 - El servidor no necesita guardar el estado ni información del cliente de una petición a otra
 - No existen mecanismos de inicio y/o fin de sesión
 - Cada petición es independiente de cualquier otra
 - En cada petición se envía toda la información necesaria
- HTTP es muy simple:
Ejecutamos # telnet
Enviamos comandos HTTP: GET – para pedir un fichero
EJEMPLO: # telnet 150.214.57.133 80
telnet> GET /webiv/index.html
- CGI (Common Gateway Interface) no es un lenguaje sino una forma de intercambio de información entre el servidor Apache y el navegador: el contenido html que se envía al cliente se genera en el servidor mediante un código en un lenguaje determinado interpretado: PHP, Perl, Python ...

Servidor Web: Apache

- A veces es necesario recurrir a otros lenguajes más eficientes y versátiles como Java
- Esto se hace a través de un SERVLET: son programas en hechos en Java que, una vez compilados, se ejecutan en el servidor para generar de forma más eficiente (en lugar de usar lenguajes interpretados como PHP) el contenido dinámico que luego se envía al cliente.
- Para ejecutar un SERVLET es necesario instalar (además de la JRE) el servidor TOMCAT que permite la ejecución de los programas Java.
- Con el PHPJava Bridge podremos incrustar código Java dentro de nuestro código PHP y llamar a funciones compiladas hechas en Java

Servidor Web: Apache

- Existen algunos mecanismos de optimización del acceso a servidores web. Por ejemplo el **BALANCEO DE CARGA**, que implica:
 - Redundancia de datos/servicios
 - Eficiencia: repartimos la carga
- Métodos de balanceo de carga:
 - Round Robin DNS – En el servidor DNS asignamos a un mismo nombre de máquina distintas direcciones IP:
 - www.uma.es. IN CNAME www1.uma.es
 - www.uma.es. IN CNAME www2.uma.es
 - www1.uma.es. IN A 150.214.40.101
 - www2.uma.es. IN A 150.214.40.102(CNAME significa Canonical NAME, y es un tipo de registro DNS para definir un alias)
 - Hardware de balanceo de carga (dentro del router)
 - Software de balanceo de carga para Linux

Servidor Web: Apache

- **Servidores WEB más populares:**

- **Apache (ha bajado mucho últimamente, 50% servidores)**
- **Nginix (20% y subiendo con fuerza, puede trabajar con miles de conexiones simultáneas) usado por grandes corporaciones: Facebook, WordPress...**
- **IIS (Internet Information Services, de Microsoft): web, ftp...**
- **GWS (Google Web Server, casi un 10% de los sitios web)**

Instalación de “Apache 2”:

```
# sudo -i ; apt-get update
```

```
# apt-get install apache2
```

```
...
```

```
Processing triggers for ufw ...
```

```
Enabling module mpm_event
```

```
Enabling module authz_core
```

```
...
```

Servidor Web: Apache

• Cortafuegos de Linux:

ufw: en realidad es un script incluido en Ubuntu para simplificar el uso/gestión de netfilter y iptables

- **Por defecto está instalado (si no, hacemos apt-get install ufw)**
... pero suele estar **inactivo**. Para ver el **estado actual** hacemos:
ufw status

Status: **inactive** / Status: **active**

Para habilitar o deshabilitar:

ufw enable / # ufw disable

Firewall is active/stopped and enabled/disabled on startup

- **Ufw tiene distintos perfiles para cada aplicación instalada:**

ufw app list ... y podemos ver qué puertos toca cada perfil

Apache

ufw app info 'Apache Full'

Apache Full

80, 443 /tcp

Apache Secure

... para habilitar un perfil y abrir los puertos asociados

Bind9

ufw allow in 'Apache Full' (abre 80 y 443 de tcp)

CUPS

... para deshabilitar un perfil y cerrar sus puertos

Dovecot IMAP

ufw deny in 'Apache Full' (cierra 80 y 443 /tcp)

Dovecot POP3

Servidor Web: Apache

- Directorio de configuración: /etc/apache2
- Fichero de configuración básico:
 - /etc/apache2/httpd.conf (RedHat, CentOS, Mandrake...)
 - /etc/apache2/apache2.conf (Debian, Ubuntu, Linux Mint...)
- Comando de control:

```
# apache2ctl start|restart|stop|configtest|status
```
- Añadir nombre/IP del servidor en el fichero de configuración:

```
# cat "ServerName 127.0.0.1" >> /etc/apache2/apache2.conf
```
- ...y reiniciamos apache:

```
# apache2ctl restart
```
- Variables de entorno usadas por Apache en fichero llamado:

```
/etc/apache2/envvars
```
- Definición de los puertos TCP que usará Apache:

```
/etc/apache2/ports.conf
```

Por defecto está configurado para usar el puerto TCP 80
Si el módulo SSL está activo también tendrá el TCP 443.

Servidor Web: Apache

• Directorio **etc/apache2/mods-available/**

Contiene un par de archivos .load y .conf por cada modulo

***.load contiene información para cargar el módulo en Apache**

***.conf contiene la configuración para usar con el módulo cargado**

• Directorio **etc/apache2/mods-enable/**

Contiene enlaces simbólicos a los archivos .load y .conf activados

ACTIVACIÓN/DESACTIVACIÓN DE UN MÓDULO

Método A a.1) **Hacer los links simbólicos en mods-enable para activar**

apache2> ln -s mods-available/cgi.load mods-enable/

apache2> ln -s mods-available/cgi.conf mods-enable/

apache2> apache2ctl restart

a.2) **Borrar los enlaces para desactivar**

apache2> rm mods-available/cgi.*

b.1) **Usar el comando a2enmod para activar**

apache2> a2enmod cgi

b.1) **Usar el comando a2desmod para desactivar**

apache2> a2desmod cgi

Servidor Web: Apache

- Directorio de bitácora/registro de incidencias

`/var/log/apache2/access.log`

`/var/log/apache2/error.log`

- Detención normal de Apache

`#apache2ctl stop | restart` (dan problemas de cierre de ficheros)

`# service apache2 stop | restart`

`# /etc/init.d/apache2 stop | restart`

- Detención suave de Apache

`#apache2ctl graceful-stop | graceful`

- Comprobación de la configuración de Apache 2

`# apache2ctl configtest`

- Información del estado de Apache 2

`# apache2ctl status`

`# apache2ctl fullstatus`

- Documentación de Apache 2

`# apt-get install apache2-doc`

Dominios virtuales

- Vamos a crear los siguientes dominios virtuales:

`www.<apellido1>.<apellido2>.es`

`www.<apellido1>.<apellido2>.com`

- Creamos los directorios y páginas de prueba

`cd /var/www/<apellido1>.<apellido2>.es/html`

`cd /var/www/<apellido1>.<apellido2>.com/html`

- Modificamos el archivo `/etc/hosts`

`127.0.0.1 <apellido1>.<apellido2>.es`

`127.0.0.1 <apellido1>.<apellido2>.com`

- Creamos los archivos de host virtual

`cd /etc/apache2/sites-available /`

`cp 000-default.conf piliougue.rocha.es.conf`

`cp 000-default.conf piliougue.rocha.com.conf`

`...y los editamos`

Dominios virtuales

- **Dominio virtual COM**

```
<VirtualHost *:80>
```

```
    ServerName piliougue.rocha.com
```

```
    ServerAlias www.piliougue.rocha.com
```

```
    ServerAdmin michel@uma.es
```

```
    documentRoot /var/www/piliougue.rocha.com/html
```

- **Dominio virtual ES**

```
<VirtualHost *:80>
```

```
    ServerName piliougue.rocha.es
```

```
    ServerAlias www.piliougue.rocha.es
```

```
    ServerAdmin michel@uma.es
```

```
    documentRoot /var/www/piliougue.rocha.es/html
```

- **Activar los virtual hosts (y deshabilitar el host por defecto)**

```
# a2ensite piliougue.rocha.com.conf
```

```
# a2ensite piliougue.rocha.es.conf
```

```
# a2dissite 000-default.conf
```

```
# service apache2 reload
```

Intérprete de PHP

- **Instalación:**

```
# apt-get install php  
# apt-get install libapache2-mod-php  
# apt-get install php-mysql  
# apt-get install php-mcrypt
```

...

```
# apt-cache search php- | more  
(miramos módulos interesantes de php)  
# apt-cache show php-cgi
```

- **Modificamos otro fichero de configuración de Apache 2:**

/etc/apache2/mods-enabled/dir.conf

- **Reiniciamos de nuevo Apache 2:**

```
# apache2ctl restart
```

...y comprobamos el estado del demonio de Apache ...

```
# systemctl status apache2
```

Servidor de bases de datos: MySQL

- **Instalación:**

```
# apt-get install mysql-server
```

- **Script para aumentar la seguridad:**

```
# mysql_secure_installation
```

- **Inicio del cliente texto:**

```
# mysql -u root -p
```

- **Instalamos el plugin de seguridad de claves**

```
mysql> install plugin validate_password soname  
      'validate_password.so';
```

- **Niveles de la política de seguridad de claves**

```
mysql> set global validate_password_policy='LOW';
```

```
mysql> set global validate_password_policy='MEDIUM';
```

```
mysql> set global validate_password_policy='STRONG';
```

- **Longitud mínima de la clave**

```
mysql> set global validate_password_length=8
```

Servidor de bases de datos: MySQL

- Directorio de configuración de MySQL /etc/mysql
- Fichero de inclusión de directorios de configuración: my.cnf
- Directorio/fichero verdadero con configuración mysql.conf.d/ mysqld.cnf
- Otros fichero de configuración de MySQL: debian-start
debian.cnf
- Demonio para iniciar / parar /reiniciar MySQL:
`# /etc/init.d/mysql start|stop|restart| status`
- Bitácora de eventos
 - `# /var/log/mysql.log` (en general)
 - `# /var/log/syslog` (en ubuntu)
- Bitácora de errores
 - `# /var/log/error.log` (en ubuntu)
- Localización real de ficheros de la base de datos
 - `# /var/lib/mysql`
- Herramienta mysqladmin:
`# mysqladmin`

Servidor de bases de datos: MySQL

Creamos una base de datos y alguna tablas ...

```
mysql> show databases;  
mysql> create database prueba;  
mysql> use prueba  
mysql> create table alumno (  
        id int not null auto_increment,  
        nombre varchar(100) not null,  
        apellido varchar(100) not null,  
        primary key(id)  
);  
mysql> insert into alumno(nombre,apellido) values  
        ('Luis','Sanchez'),  
        ('Aurelio','Casanova');
```

Intérprete de PHP + MySQL

- **Modificaciones configuración** /etc/php/7.0/apache2/php.ini

Buscamos “error_reporting”. Lo ponemos a “E_ALL”

Buscamos “display_errors”. Lo ponemos a “On”

Buscamos “mysql” y activamos la extensión

Uso de MySQL en PHP

- **Conexión**

```
$enlace=mysqli_connect('127.0.0.1','root','servidor');
```

- **Selección de base de datos**

```
mysqli_select_db($enlace,'prueba');
```

- **Consulta base de datos**

```
$consulta='select id,nombre,apellido from alumno';
```

```
$tabla=mysqli_query($enlace,$consulta);
```

- **Recuperación de registros**

```
while ($row=mysqli_fetch_array($tabla)) { ... }
```

- **Lectura de atributos/campos:**

```
echo "Nombre: ".$row['nombre']. "<br>\n";
```

```
echo "Apellido: ".$row['apellido']. "<br>\n";
```



MUCHAS GRACIAS POR SU ATENCIÓN