

## A1\_T2. Instal·lació d'un servidor web

La instalación del [servidor web Apache](#) en GNU/Linux es trivial si se realiza desde la herramienta de gestión de paquetes propia de la distribución. Las principales distribuciones incluyen la última versión estable de Apache con sus módulos para ejecutar aplicaciones dinámicas en PHP y acceder a SGBD como MySQL o PostgreSQL entre otros.

En este apartado vamos a ver la configuración típica de un servidor web Apache en una distribución Ubuntu/Debian.

### Requisitos

El sistema de gestión de paquetes de nuestra distribución se ocupará de resolver las dependencias de Apache, que dependerán en gran medida de los módulos que se instalen y el soporte de diferentes lenguajes bases de datos con los que se trabaje.

En general un servidor web consume:

- Ancho de banda para atender los accesos de los clientes
- CPU si tiene que ejecutar aplicaciones dinámicas
- RAM si debe ejecutar aplicaciones dinámicas o mantener un número muy alto de concurrencia
- Espacio de disco para almacenar contenidos estáticos

Evidentemente el consumo de recursos estará relacionado con el nivel de carga y el cuello de botella se producirá en uno u otro lugar dependiendo del uso concreto, pero en un servidor web básico el ancho de banda disponible será el factor crítico. El consumo de memoria principal limitará el número máximo de conexiones concurrentes que se pueden mantener y la cantidad de memoria disponible para los scripts. La presión de memoria nunca debe ser tan grande como para utilizar memoria virtual, si se llega a este extremo se deteriorará el rendimiento.

El espacio de disco consumido dependerá de los contenidos a servir, en un mirror o sitio de réplicas puede ser muy grande. Como regla general no conviene utilizar sistemas de archivos remotos como NFS o Samba.

## Esquema de configuración predeterminada en Ubuntu / Debian

Cuando se instala Apache, la configuración predeterminada deja un servidor funcional que publica la página estática de prueba. Basta abrir un navegador y acceder la cualquier dirección de red local para comprobar el funcionamiento. Por defecto Apache está conectado al puerto 80 y 443 (http y https) de todas las interfaces locales.

http://127.0.0.1/

La página estática está guardada en `/var/www`, que está configurado como raíz de los documentos ([DocumentRoot](#)). Allí se irán a buscar los contenidos especificados por la URL a no ser que una directiva [Alias](#) los dirija a otro lugar.

En el directorio `/var/log/apache2` se encuentran los ficheros de registro que dan cuenta de la actividad del servidor. En **access.log** quedan registrados los accesos de los clientes y en **error.log** los errores, advertencias y notificaciones.

Los ficheros de configuración se encuentran en `/etc/apache2`. El fichero de configuración principal es **apache2.conf**, en **ports.conf** está la configuración sobre los puertos/interfaces de red en los que Apache escucha. Los diferentes módulos instalados y su configuración se pueden encontrar en **mods-available** o **mods-enabled** según se trate de módulos instalados pero que no están en uso o módulos habilitados. También existen los directorios **sites-available** y **sites-enabled** para la configuración de diferentes [VirtualHost](#) que están disponibles/habilitados.

En Debian/Ubuntu los comandos **a2enmod**, **a2ensite**, **a2dismod**, **a2dissite** sirven respectivamente para habilitar/deshabilitar un módulo/sitio web.

Por último los procesos del servidor se ejecutan bajo el UID/GID **www-data**.

## Configuración de aspectos básicos

El fichero de configuración `/etc/apache2/ports.conf` determina en qué puertos e interfaces de red Apache escuchará peticiones. Por defecto lo hace en el puerto 80 (http) y 443 (https) de cualquier interfaz.

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A1: INSTAL·LACIÓ DE SERVIDORS D'APLICACIONS WEB



```
# 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
# This is also true if you have upgraded from before 2.2.9-3 (i.e. from
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.Debian.gz and
# README.Debian.gz

NameVirtualHost *:80

Listen 80

<IfModule mod_ssl.c>

    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.

    Listen 443

</IfModule>

<IfModule mod_gnutls.c>

    Listen 443

</IfModule>
```

En el fichero de configuración principal (**apache2.conf**) se especifican entre otras cuestiones:

**ServerRoot "/etc/apache2"**

El directorio raíz bajo el que están los ficheros de configuración y registro del servidor

**La configuración del módulo de multiproceso (MPM) utilizado**

El servidor web intenta mantener un pool de procesos (workers) para atender las peticiones concurrentes de los usuarios. Este pool se puede gestionar/implementar de diferentes maneras, por eso se debe escoger un módulo MPM concreto para cada instalación.

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A1: INSTAL·LACIÓ DE SERVIDORS D'APLICACIONS WEB



Las opciones más interesantes son:

### prefork

El módulo utilizado por defecto en GNU/Linux. Mantiene un pool de procesos sin hebras (un hilo de ejecución por proceso), es el modelo utilizado en versiones anteriores de Apache y permite compatibilidad con aplicaciones que utilizan componentes que no son thread-safe. Consigue el mayor aislamiento posible entre peticiones y la máxima estabilidad.

### worker

Implementado mediante un modelo híbrido multiproceso multihebra, se lanzan varios procesos worker donde cada uno de ellos tiene varias hebras. Proporciona un mayor rendimiento con un menor consumo de recursos intentando mantener la estabilidad de prefork.

### event

Módulo experimental que intenta mejorar el rendimiento de worker con hebras auxiliares.

En el caso de prefork, la configuración del módulo debe ser tal que cuente con un valor **MaxClients** suficientemente grande como para atender a todas las peticiones simultáneas que se esperen recibir pero suficientemente pequeño como para evitar que se deba consumir memoria virtual.

## **Los VirtualHost**

Aunque Apache cuenta con un buen número de [opciones para realizar mapeados de URLs](#), en algún momento será necesario tener servidores virtuales para publicar sitios web diferenciados.

Apache permite definir sitios web virtuales, cada uno con su propia configuración, basados en IP o nombre. La opción más utilizada es la basada en nombre pues las IPs públicas tienen un coste y sin embargo no hay problema en asignar múltiples nombres DNS a una sola IP.

Por ejemplo:

```
# escoladeltreball.net
```

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A1: INSTAL·LACIÓ DE SERVIDORS D'APLICACIONS WEB



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

```
ServerName escoladeltreball.net
```

```
RewriteEngine On
```

```
RewriteRule ^/(.*) http://192.168.10.14:9080/VirtualHostBase/http/escoladeltreball.net:80/escoladeltreball/Vir  
tualHostRoot/$1 [L,P]
```

```
<Location />
```

```
Order Deny,Allow
```

```
Allow from all
```

```
</Location>
```

```
</VirtualHost>
```

```
# moodle.escoladeltreball.net
```

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

```
ServerName moodle.escoladeltreball.net
```

```
RewriteEngine On
```

```
RewriteRule ^/(.*) http://moodle.escoladeltreball.net/$1 [L,P]
```

```
<Location />
```

```
Order Deny,Allow
```

```
Allow from all
```

```
</Location>
```

```
</VirtualHost>
```

```
# ubuntu.escoladeltreball.net
```

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

```
ServerName ubuntu.escoladeltreball.net
```

```
DocumentRoot /mnt/store/apt-mirror/mirror/archive.ubuntu.com
```

```
<Directory /mnt/store/apt-mirror/mirror/archive.ubuntu.com>
```

```
Options Indexes FollowSymLinks
```

```
Order Deny,Allow
```

```
Allow from all
```

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A1: INSTAL·LACIÓ DE SERVIDORS D'APLICACIONS WEB



```
</Directory>
```

```
</VirtualHost>
```

## Control de acceso

Apache puede controlar el acceso a los contenidos de diferentes maneras, básicamente:

- Puede utilizar las directivas [Order](#), [Allow](#) y [Deny](#) para especificar qué clientes pueden acceder a los contenidos
- Cuenta con un sistema flexible de [módulos que permiten especificar distintos tipos de autenticación y autorización](#)

## Puesta a prueba del servidor web

Una vez instalado el servidor web Apache, la configuración por defecto ya publica una página web estática que permite comprobar su buen funcionamiento. También es posible utilizar la URL **<http://127.0.0.1/server-status>** para obtener una página web, generada por el módulo [mod\\_status](#), en la que se muestra información sobre el funcionamiento del servidor.

Apache también incluye una pequeña herramienta, **ab**, con la que es posible generar peticiones automatizadas al servidor para realizar pruebas de carga.

### Enlaces:

- [Servidor web Apache](#)
- [Documentación oficial de Apache 2.2](#)

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A1: INSTAL·LACIÓ DE SERVIDORS D'APLICACIONS WEB

