

2011-
2012

Actividad 7 Administración
servidor Web HTTP (Apache2)
en Ubuntu Server --Módulos --
Directorios personales de
usuarios



José Jiménez Arias
IES Gregorio Prieto
2011-2012

Apache es un servidor modular: el núcleo del servidor tiene las funcionalidades básicas que se pueden ampliar añadiendo módulos adicionales. Cada módulo agrupa un conjunto de funcionalidad y directivas para configurarlas.

Antes de comenzar hemos de recordar que apache consta de 2 tipos de módulos:

- Módulos estáticos que se añaden cuando se compila apache.
- Módulos que se cargan dinámicamente cuando se inicia el servidor.

En primer lugar iniciamos sesión y ejecutamos el siguiente comando para comprobar los módulos estáticos que se han cargado al compilar el servidor.

```
root@ubuntusrv04:/home/josejimenez# apache2ctl -l
Compiled in modules:
  core.c
  mod_log_config.c
  mod_logio.c
  worker.c
  http_core.c
  mod_so.c
```

A continuación comprobamos que los módulos se han cargado dinámicamente al arrancar el servidor consultando el directorio /etc/apache2/mods-enabled.

```
root@ubuntusrv04:/etc/apache2# cd mods-enabled
root@ubuntusrv04:/etc/apache2/mods-enabled# ls
alias.conf          authz_user.load    dir.conf           reqtimeout.conf
alias.load          autoindex.conf    dir.load           reqtimeout.load
auth_basic.load     autoindex.load    env.load          setenvif.conf
authn_file.load     cgid.conf         mime.conf         setenvif.load
authz_default.load  cgid.load         mime.load         status.conf
authz_groupfile.load deflate.conf      negotiation.conf  status.load
authz_host.load     deflate.load      negotiation.load
```

observa que los ficheros que aparecen anteriormente son enlaces simbólicos a ficheros de /etc/apache2/mods-available

```
root@ubuntusrv04:/etc/apache2/mods-enabled# cd /etc/apache2/mods-available
root@ubuntusrv04:/etc/apache2/mods-available# ls
actions.conf        cache.load         filter.load        proxy_http.load
actions.load        cern_meta.load    headers.load       proxy.load
alias.conf          cgid.conf         ident.load         proxy_scgi.load
alias.load          cgid.load         imagemap.load     reqtimeout.conf
asis.load           cgi.load          include.load       reqtimeout.load
auth_basic.load     charset_lite.load info.conf          rewrite.load
auth_digest.load   dav_fs.conf       info.load         setenvif.conf
authn_alias.load   dav_fs.load       ldap.load         setenvif.load
authn_anon.load    dav.load          log_forensic.load spelling.load
authn_dbd.load     dav_lock.load     mem_cache.conf    ssl.conf
authn_dbm.load     dbd.load          mem_cache.load    ssl.load
```

Editamos uno de los ficheros .load, por ejemplo alias.load:

```
GNU nano 2.2.2 Archivo: alias.load
LoadModule alias_module /usr/lib/apache2/modules/mod_alias.so
```

Editamos uno de los .conf, por ejemplo alias.conf:

```
GNU nano 2.2.2 Archivo: alias.conf
<IfModule alias_module>
#
# Aliases: Add here as many aliases as you need (with no limit). The format
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/". If the fakename is slash-terminated, then the
# realname must also be slash terminated, and if the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed directory listings. If
# you do not use FancyIndexing, you may comment this out.
#
Alias /icons/ "/usr/share/apache2/icons/"

<Directory "/usr/share/apache2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

</IfModule>
```

A continuación consultamos el directorio /usr/lib/apache2/modules para observar los módulos que tenemos disponibles para cargar:

```
root@ubuntusrv04:/etc/apache2/mods-enabled# cd /usr/lib/apache2/modules
root@ubuntusrv04:/usr/lib/apache2/modules# ls
httpd.exp          mod_cgi.so        mod_mime_magic.so
mod_actions.so    mod_charset_lite.so  mod_mime.so
mod_alias.so      mod_dav_fs.so      mod_negotiation.so
mod_asis.so       mod_dav_lock.so    mod_proxy_ajp.so
mod_auth_basic.so mod_dav.so         mod_proxy_balancer.so
mod_auth_digest.so mod_dbd.so         mod_proxy_connect.so
mod_authn_alias.so mod_deflate.so     mod_proxy_ftp.so
mod_authn_anon.so mod_dir.so         mod_proxy_http.so
mod_authn_dbd.so  mod_disk_cache.so  mod_proxy_scgi.so
mod_authn_dbm.so  mod_dumpio.so     mod_proxy.so
mod_authn_default.so mod_env.so        mod_reqtimeout.so
mod_authn_file.so mod_expires.so    mod_rewrite.so
mod_authnz_ldap.so mod_ext_filter.so  mod_setenvif.so
```

Posteriormente ejecutamos el siguiente comando para mostrar los paquetes disponibles en los repositorios de Ubuntu que permiten instalar módulos adicionales en Apache:

```
root@ubuntusrv04:/usr/lib/apache2/modules# apt-cache search libapache2-mod

libapache2-mod-geoip - Soporte GeoIP para apache2
libapache2-mod-ldap-userdir - Apache module that provides UserDir lookups via LDAP
libapache2-mod-log-sql - Use SQL to store/write your apache queries logs - Base
libapache2-mod-log-sql-dbi - Use SQL to store/write your apache queries logs - D
BI interface
libapache2-mod-log-sql-mysql - Use SQL to store/write your apache queries logs -
MySQL interface
libapache2-mod-mono - Apache module for running ASP.NET applications on Mono
libapache2-mod-musicindex - Browse, stream, download and search through MP3/Ogg/
FLAC files
```

El siguiente paso es comprobar que el módulo userdir no está habilitado, para ello visitaremos el directorio /etc/apache2/mods-enabled

```
root@ubuntusrv04:/usr/lib/apache2/modules# cd /etc/apache2/mods-enabled
root@ubuntusrv04:/etc/apache2/mods-enabled# ls
alias.conf          authz_user.load    reqtimeout.conf
alias.load          autoindex.conf    reqtimeout.load
auth_basic.load     autoindex.load    setenvif.conf
authn_file.load     cgid.conf         setenvif.load
authz_default.load  cgid.load         status.conf
authz_groupfile.load deflate.conf      negotiation.conf
authz_host.load     deflate.load      negotiation.load  status.load
root@ubuntusrv04:/etc/apache2/mods-enabled#
```

Observamos que userdir no está, debería estar donde se situa el cuadro

Habilitamos el módulo userdir con el siguiente comando:

```
root@ubuntusrv04:/etc/apache2/mods-enabled# a2enmod userdir
Enabling module userdir.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntusrv04:/etc/apache2/mods-enabled#
```

Nota: Para deshabilitar un módulo, por ejemplo el anterior se realiza con la sentencia:

```
root@ubuntusrv04:/etc/apache2/mods-enabled# a2dismod userdir
Module userdir disabled.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntusrv04:/etc/apache2/mods-enabled#
```

Lo volvemos a habilitar con la sentencia "a2enmod userdir" y continuamos con el ejercicio.

Una vez activado el módulo nos situamos en el directorio de los módulos habilitados /etc/apache2/mods-enabled y realizamos ls para comprobar si ha sido creados los ficheros .conf y.load

```
root@ubuntusrv04:/etc/apache2/mods-enabled# ls
alias.conf          autoindex.conf     env.load           setenvif.load
alias.load          autoindex.load     mime.conf         status.conf
auth_basic.load     cgid.conf          mime.load         status.load
authn_file.load     cgid.load          negotiation.conf  userdir.conf
authz_default.load  deflate.conf       negotiation.load  userdir.load
authz_groupfile.load  deflate.load      reqtimeout.conf
authz_host.load     dir.conf           reqtimeout.load
authz_user.load     dir.load           setenvif.conf
```

Reiniciamos el servicio para que los cambios tengan efecto.

Observamos el contenido de ambos ficheros, si observamos detenidamente, podemos evidenciar que userdir.conf habilita el uso de los directorios personales para todos los usuarios excepto root y que public_html es el subdirectorio que pueden crear (no se crea con habilitar el módulo, ha de ser manualmente) para compartir sus páginas.

```
root@ubuntusrv04:/etc/apache2/mods-enabled# cat userdir.load
LoadModule userdir_module /usr/lib/apache2/modules/mod_userdir.so
root@ubuntusrv04:/etc/apache2/mods-enabled# cat userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit Indexes
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        <Limit GET POST OPTIONS>
            Order allow,deny
            Allow from all
        </Limit>
        <LimitExcept GET POST OPTIONS>
            Order deny,allow
            Deny from all
        </LimitExcept>
    </Directory>
</IfModule>
```

A continuación para comprobar la veracidad de la anterior afirmación hemos de crear el directorio public_html y dentro de este un documento de nombre personal04.html

```
jose.jimenez@ubuntusrv04:~$ mkdir public_html
jose.jimenez@ubuntusrv04:~$ cd public_html
jose.jimenez@ubuntusrv04:~/public_html$ nano personal04.html
```

El contenido de la página situada en el directorio public_html es:

```
GNU nano 2.2.2 Archivo: personal04.html
<html>
<body>
<h1>PAGINA PERSONAL DE LOS USUARIOS JUAN EDUARDO Y JOSE</h1>
</body>
</html>
```

Por último nos dirigimos a un cliente y mediante el navegador accedemos al servidor web para comprobar si tenemos acceso al documento creado en el paso anterior:

