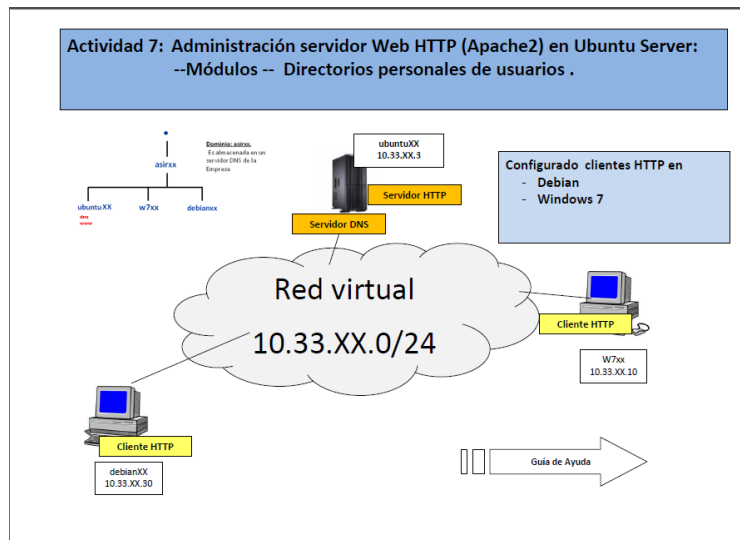


ACTIVIDAD 7- MODULOS-DIRECTORIOS PERSONALES DE USUARIOS – TEMA 4 – HTTP



Lo primero que vamos a hacer es poner el comando para comprobar los módulos estáticos que se han cargado al compilar el servidor, para ello ponemos:

```
apache2ctl -l
```

```
lales@ubuntu10:~$ sudo su
[sudol password for lales:
root@ubuntu10:/home/lales# apache2ctl -l
Compiled in modules:
 core.c
 mod_log_config.c
 mod_logio.c
 worker.c
 http_core.c
 mod_so.c
root@ubuntu10:/home/lales#
```

Ahora vamos a ver los módulos que hay en el directorio /etc/apache2/mods-enabled con un ls

```
root@ubuntu10:/etc/apache2/mods-available# ls
actions.conf          cern_meta.load      ident.load           proxy_http.load
actions.load          cgid.conf           imagenap.load       proxy.load
alias.conf            cgi.load            include.load        proxy_scgi.load
alias.load            cgi.load            info.conf           reqtimeout.conf
asis.load             charset_lite.load   info.load           reqtimeout.load
auth_basic.load       dav_fs.conf         ldap.conf           rewrite.load
auth_digest.load     dav_fs.load         ldap.load           setenvif.conf
authn_alias.load     dav.load            log_forensic.load  setenvif.load
authn_anon.load      dav_lock.load       mem_cache.conf     spelling.load
authn_dbd.load       dbd.load            mem_cache.load     ssl.conf
authn_dbm.load       deflate.conf        mime.conf           ssl.load
authn_default.load   deflate.load        mime.load           status.conf
authn_file.load      dir.conf            mime_magic.conf    status.load
authnz_ldap.load     dir.load            mime_magic.load    substitute.load
authz_dbm.load       disk_cache.conf    negotiation.conf   suexec.load
authz_default.load   disk_cache.load    negotiation.load   unique_id.load
authz_groupfile.load dump_io.load        proxy_ajp.load     userdir.conf
authz_host.load      env.load            proxy_balancer.conf userdir.load
authz_owner.load     expires.load        proxy_balancer.load usertrack.load
authz_user.load      ext_filter.load    proxy.conf          version.load
autoindex.conf       file_cache.load    proxy_connect.load vhost_alias.load
autoindex.load       filter.load         proxy_ftp.conf     proxy_ftp.load
cache.load            headers.load        proxy_ftp.load
```

Ahora vamos a ir a los ficheros alias.load y vemos como se utiliza la directiva LoadModule para cargar el módulo

```
GNU nano 2.2.4 Archivo: alias.load
LoadModule alias_module /usr/lib/apache2/modules/mod_alias.so
```

Ahora editamos el fichero alias.conf para ver como se añaden las directivas dentro de una declaración

```
GNU nano 2.2.4 Archivo: alias.conf
<IfModule alias_module>
#
# Aliases: Add here as many aliases as you need (with no limit). The format is
# 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
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía
```

Ahora nos vamos al directorio /usr/lib/apache2/modules y hacemos un ls para ver los módulos disponibles

```
root@ubuntu10:/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
mod_authz_dbm.so mod_file_cache.so mod_speling.so
mod_authz_default.so mod_filter.so mod_ssl.so
mod_authz_groupfile.so mod_headers.so mod_status.so
mod_authz_host.so mod_ident.so mod_substitute.so
mod_authz_owner.so mod_imagemap.so mod_suexec.so
mod_authz_user.so mod_include.so mod_unique_id.so
mod_autoindex.so mod_info.so mod_userdir.so
mod_cache.so mod_ldap.so mod_usertrack.so
mod_cern_meta.so mod_log_forensic.so mod_version.so
mod_cgid.so mod_mem_cache.so mod_who_who_alias.so
root@ubuntu10:/usr/lib/apache2/modules# _
```

Ahora vamos a mostrar los paquetes disponibles en los repositorios de Ubuntu que permiten instalar módulos adicionales en Apache, para ellos ponemos el siguiente comando

```
sudo apt-cache search libapache2-mod
```

```
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
libapache2-mod-neko - Apache module for running server-side Neko programs
libapache2-mod-removeip - Module to remove IP from apache2's logs
libapache2-mod-suphp - Módulo Apache2 para correr guiones php con permisos de pr
opietario
mod-musicindex-common - Archivos comunes para mod-musicindex
libapache2-mod-php5 - server-side, HTML-embedded scripting language (Apache 2 mo
dule)
php5-cgi - server-side, HTML-embedded scripting language (CGI binary)
libapache2-mod-apparmor - changehat AppArmor library as an Apache module
libapache2-mod-authnz-external - authenticate Apache against external authentica
tion services
libapache2-mod-fcgid - an alternative module compat with mod_fastcgi
libapache2-mod-fcgid-dbg - debugging symbols for mod_fcgid
libapache2-mod-php5filter - server-side, HTML-embedded scripting language (apach
e 2 filter module)
php5-fpm - server-side, HTML-embedded scripting language (FPM-CGI binary)
root@ubuntu10:/usr/lib/apache2/modules# _
```

Ahora vamos a ver en el directorio /etc/apache2/mods-enabled con un ls que el módulo userdir no está habilitado, ya que no aparece en la lista de módulos

```
root@ubuntu10:/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
root@ubuntu10:/etc/apache2/mods-enabled# _
```

Para habilitarlo lo que hacemos es poner sudo a2enmod userdir y vemos como haciendo un ls nos aparece ya el módulo junto con los otros

```
root@ubuntu10:/etc/apache2/mods-enabled# sudo a2enmod userdir
Enabling module userdir.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@ubuntu10:/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
root@ubuntu10:/etc/apache2/mods-enabled# _
```

Ahora vamos a ir al fichero userdir.conf y vemos como no public_html es el nombre del subdirectorio donde podemos crear las páginas personales

```
GNU nano 2.2.4 Archivo: 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>
```

Ahora creamos un directorio llamado public_html

```
root@ubuntu10:/etc/apache2/mods-enabled# cd ..
root@ubuntu10:/etc/apache2# cd ..
root@ubuntu10:/etc# cd ..
root@ubuntu10:/# cd /home
root@ubuntu10:/home# cd lales
root@ubuntu10:/home/lales# mkdir public_html
root@ubuntu10:/home/lales#
```

Y dentro un fichero que se llame personal1.html donde vamos a poner un texto

```
GNU nano 2.2.4 Archivo: personal1.html Modificado
<html>
<body>
<h1> PAGINA PERSONAL DEL USUARIO LALES </h1>
</body>
</html>
```

A continuación accedemos en el navegador de Debian con 10.33.10.3/~lales/personal1.html y vemos el texto que hemos escrito en el fichero personal1.html

