

Capítol 5

El Servei HTTP
Configuració del servei
Opcions del servidor Apache2

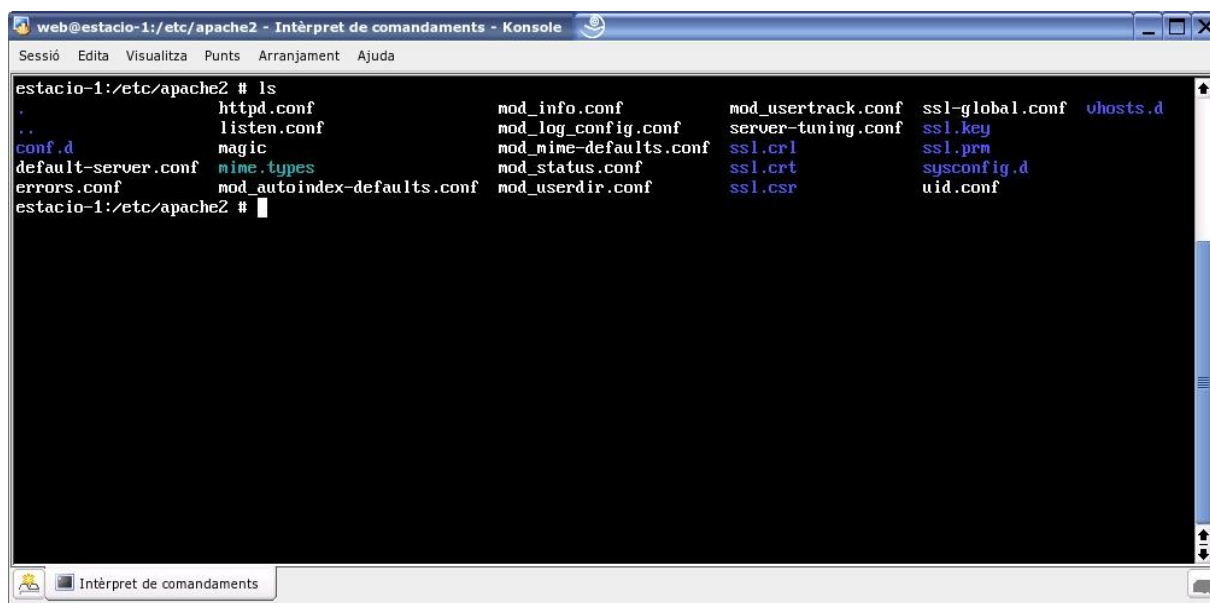
5.1 Servei HTTP: servidor WEB Apache

Sense cap mena de dubte el servei més “rellevant” que pot oferir el nostre servidor de cara als usuaris finals és el servei de pàgines WEB. Des de la invenció de l'estàndard HTTP Internet s'ha convertit en un mitjà de comunicació d'àmbit mundial amb el servei WEB com a fil conductor de la informació. Podem definir un servidor WEB com l'aplicació informàtica que gestiona les peticions d'ordinadors clients (per mig d'un navegador) per accedir a informació que hi ha emmagatzemada en el disc dur local del servidor en format HTML o altres formats comprensibles per el servidor WEB o el client que fa la petició. En realitat, el servei WEB (el servidor) pot fer moltes altres coses, com ara dur un registre estricte del tràfic HTTP, permetre autenticacions, crear diferents àmbits de servei per un mateix servidor, etc.

Hi ha un cert nombre de servidors WEB que compleixen els estàndards establerts internacionalment per HTTP 1.1. Entre tots destaca Apache [1] com el servidor WEB més utilitzat arreu del món. L'estadística publicada al [Netcraft Web Server Survey](#) [2] en febrer del 2005 apunta que més d'un 64 % dels servidors WEB que estan actualment actius a la xarxa són del tipus Apache en les seves diferents versions (versions de programari 1.3 o 2, versions per sistemes operatius UNIX/Linux, Netware, OSX, Windows).

A la pàgina del Projecte Apache es defineix el servidor seu servidor HTTP com una aplicació de codi obert desenvolupada per sistemes operatius moderns (de tipus UNIX i Windows) amb les característiques de ser segur, eficient i extensible però assegurant l'estricta compliment dels estàndards. L'actual versió estable del servidor Apache és la 2.0.53. De la versió Apache 1.3 a la versió 2.0 hi ha canvis significatius que requereixen reconfigurar el servidor si es vol fer una migració d'una versió a l'altra. Ens centrarem en la versió 2.0 però cal considerar que la versió 1.3 està molt estesa i es encara utilitzada per molts servidors WEB [3].

En la versió 2.0 d'Apache els fitxers de configuració es troben sota el directori `/etc/apache2`. A continuació es pot observar un llistat d'aquest directori:



```

web@estacio-1:/etc/apache2 - Intèrpret de comandaments - Konsole
Sessió Edita Visualitza Punts Arranjament Ajuda

estacio-1:/etc/apache2 # ls
.                  mod_info.conf      mod_usertrack.conf  ssl-global.conf    vhosts.d
..                 listen.conf         server-tuning.conf  ssl.key
conf.d             magic              ssl.crl             ssl.prm
default-server.conf  mime.types         ssl.crt             sysconfig.d
errors.conf         mod_autoindex-defaults.conf  mod_status.conf    uid.conf
mod_log_config.conf  mod_mime-defaults.conf  mod_userdir.conf    ssl.csr
  
```

Per configurar Apache hi ha diverses opcions: línia de comandes, YAST per als usuaris de SuSE i WEBMIN. En aquest cas WEBMIN ens ofereix una molt interessant possibilitat per la facilitat que ens ofereix en el seu entorn. De totes formes, utilitzarem l'edició directa dels fitxers de configuració per fer-ho ja que ens interessa una posta en marxa senzilla. Apache bé ja configurat en les seves opcions “normals” i es pot executar una vegada instal·lat amb les opcions per defecte.

Per engegar i para el servei cal utilitzar el shell-script `/etc/rc.d/apache2`. Les opcions de l'script són:

<i>Opció</i>	<i>Funcionalitat</i>
start	Engega Apache
startsl	Engega amb suport DSSL
stop	Para Apache
try-start	Para i torna a engegar Apache si ha tingut èxit
status	Verifica quin serveis HTTPD hi ha en marxa
restart	Para Apache i el torna a engegar
restart-hup	Atura i torna a engegar (SIGHUP) Apache o el fica en marxa si està aturat
reload graceful	Atura i torna a engegar (SIGUSR1) Apache o el fica en marxa si està aturat
configtest	Fa un test sobre la sintaxi de la configuració
extreme-configtest	Engega Apache amb usuari nobody (detecta més errors)
probe	Comprova què necessita tornar a carregar per el bon funcionament del servei. Retorna com argument allò que cal fer.
full-server-status	Ens mostra en pantalla (de text) l'estat complert del servidor (ens cal lynx o w3m i mod_status habilitat)
server-status	Ens mostra en pantalla (de text) l'estat resumit del servidor (ens cal lynx o w3m i mod_status habilitat)
help	Mostra una pantalla d'ajuda amb aquestes opcions

Una vegada instal·lat i ficat en marxa el servei, podem comprovar el seu funcionament utilitzant un navegador i fent una connexió com la següent: <http://localhost>. Ens apareix una pantalla de benvinguda del servidor WEB que ens informa que és un test del servidor WEB Apache. En aquesta pàgina veurem dos enllaços: un ens porta a la web del projecte Apache i l'altre a la documentació d'Apache que tenim en el nostre servidor.

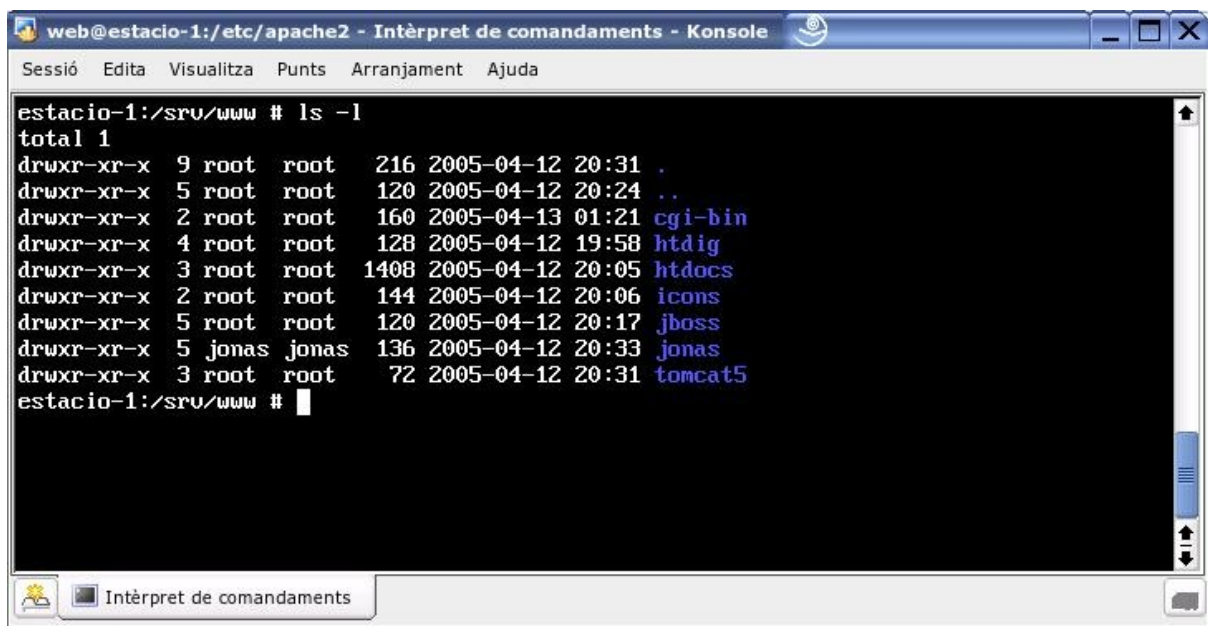
Les opcions de configuració del servei Apache2 són moltes i molt extenses. Aquí faré una descripció de les més usals i en quin fitxer de configuració les podem trobar i modificar. En cap cas intento explicar configuracions “complexes” com podrien ser la creació de servidors segurs (SSL). Tota la documentació i els exemples que s'adjunten a la aquesta són prou explícits com per qualsevol administrador d'un sistema que conegui les bases de funcionament del servidor pugui realitzar les manipulacions corresponent.

5.2 Servidor per defecte i servidors virtuals

El servidor Apache ens permet realitzar moltes de les feines que esperem que ens permeti fer un servidor web comercial. Entre altres possibilitats ens permet fer el següent:

- Servei HTTP estàndard
- Servei HTTP amb negociació del contingut segons estàndard HTTP 1.1.
- Servei HTTP amb certificats de seguretat i xifrat (SSL/TLS)
- Servei HTTP amb execució de CGIs i suport per múltiples llenguatges de programació (fins i tot amb llenguatges anomenats “embedded” inclosos al codi html): PERL, PHP, Python, ASP, ...
- Servei HTTP amb publicació de pàgines per usuari (`public_html`)
- Servei HTTP per a múltiples servidors (servidors virtuals) per redirecció de port o per nom del servidor.
- Servei HTTP amb definició de restriccions d'accés a zones del servidor (usuari/password)
- Servei HTTP amb diferents nivells de control (en forma de fitxers de text) on es guarda la informació de les connexions realitzades al servidor (o servidors) en harmonia a la normativa de protecció de dades vigent.

La configuració que mostra el servidor Apache2 “de sèrie” defineix el lloc on es troba l'arrel del nostre servidor web al fitxer `/etc/apache2/default-server.conf`. Editant aquest fitxer podrem veure que ens descriu detalladament cadascuna de les opcions que ens mostra. En el cas de les distribucions SuSE (això canvia d'una distribució a altra) l'arrel per defecte del servidor web es troba a `/srv/www` on hi ha els següent directoris:



```

web@estacio-1:/etc/apache2 - Intèrpret de comandaments - Konsole
Sessió  Edita  Visualitza  Punts  Arranjament  Ajuda

estacio-1:/srv/www # ls -l
total 1
drwxr-xr-x  9 root  root   216 2005-04-12 20:31 .
drwxr-xr-x  5 root  root   120 2005-04-12 20:24 ..
drwxr-xr-x  2 root  root   160 2005-04-13 01:21 cgi-bin
drwxr-xr-x  4 root  root   128 2005-04-12 19:58 htdig
drwxr-xr-x  3 root  root  1408 2005-04-12 20:05 htdocs
drwxr-xr-x  2 root  root   144 2005-04-12 20:06 icons
drwxr-xr-x  5 root  root   120 2005-04-12 20:17 jboss
drwxr-xr-x  5 jonas jonas  136 2005-04-12 20:33 jonas
drwxr-xr-x  3 root  root    72 2005-04-12 20:31 tomcat5
estacio-1:/srv/www #

```

Els documents html s'han de publicar sota el directori `htdocs`, mentre que cal localitzar els executables CGI sota el directori `cgi-bin` (tot això es pot canviar com veurem més tard). `htdig` correspon a un buscador que s'ha instal·lat en el sistema i `jboss`, `jonas` (implementació Open Source del [J2EE](#)) i `tomcat5` són aplicacions diverses de suport JAVA al servidor WEB.

Els usuaris del sistema poden publicar tots els documents que es deixin sota el seu directori personal dins la carpeta `public_html` sempre que es compleixin els requisits dels permisos de fitxers i directoris corresponents.

5.3 Personalitzem el servei

Podem personalitzar el servei WEB, per exemple, canviant el directori per defecte del servidor. Mirem com quedaria el fitxer *default-server.conf* amb aquestes modificacions (s'han esborrat els comentaris):

```
DocumentRoot "/home/web/htdocs"
<Directory "/home/web/htdocs">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
Alias /icons/ "/usr/share/apache2/icons/"
<Directory "/usr/share/apache2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
ScriptAlias /cgi-bin/ "/home/web/cgi-bin/"
<Directory "/home/web/cgi-bin">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
</Directory>
<IfModule mod_userdir.c>
    UserDir public_html
    Include /etc/apache2/mod_userdir.conf
</IfModule>
Include /etc/apache2/conf.d/*.conf
Include /etc/apache2/conf.d/apache2-manual.conf
```

Com es pot veure, s'ha canviat el lloc on es publiquen els documents a la WEB. Els motius per fer aquest canvi són bàsicament dos:

- ✓ Combinar el servei de publicació WEB amb la possibilitat de definir un usuari (no administrador) que pugui publicar i confeccionar pàgines web utilitzant per exemple el servei SAMBA. Si definim un usuari de xarxa anomenat web amb un directori de treball /home/web, tot el que es troba sota aquest directori serà propietat de l'usuari i estarà assignat al seu perfil mòbil en la xarxa. Això vol dir que podrà fer la publicació de la pàgina web sense necessitat de ser un usuari administrador.
- ✓ Facilitar la realització de les còpies de seguretat de la pàgina web. Normalment, una de les necessitats més importants es tenir una còpia sempre actualitzada de la publicació web. Tenir l'arrel dels documents sota /home ens facilitarà la feina a l'hora de fer còpies de seguretat com veurem al capítol 9.

Al fitxer de configuració anterior crida l'atenció la definició *Alias*. Aquesta paraula clau ens permet canviar la forma de connectar-se via web a un recurs del servidor que hauria de tenir un altre nom. Així, podem donar noms concrets a recursos que en realitat tenen un nom diferent al nostre sistema

sense haver de canviar ni la ubicació ni la seva localització. Mirem l'exemple concret següent:

El departament de química d'una universitat vol publicar la seva pròpia pàgina web al servidor central. Evidentment, el responsable del disseny de la pàgina no pot tenir accés d'administració.

Una possible solució és crear un usuari (de nom *quimica*) perquè el responsable de la pàgina web es pugui connectar al servidor i publicar el contingut de la pàgina del departament sota el directori *public_html* corresponent. Ara, per visitar la pàgina web del departament de química caldrà ficar la següent adreça: <http://localhost/~quimica/>.

Però podem evitar que els usuaris hagin d'escriure el signe ~ (AltGr 4 en Linux i ALT+126 en Windows) per poder-se connectar a aquesta pàgina si definim un *Alias*.

Aquest seria:

```
Alias /dquimica/ "/home/quimica/public_html/"
<Directory "/home/quimica/public_html/">
Options Indexes MultiViews
Order allow,deny
Allow from all
AllowOverride AuthConfig
</Directory>
```

Ara caldrà escriure únicament <http://localhost/quimica/>.

Sota la definició de l'*Alias* podem veure l'opció pel directori definit, anomenada *Directory*. La sintaxi està formada per una definició inicial d'un directori del sistema (accessible pel servidor web) una sèrie d'opcions i el tancament de la sentència. Hi ha la definició de les opcions per aquest directori amb la paraula clau *Options* i després la definició de l'accés (seguretat) al mateix directori amb les paraules *Order*, *Allow* i *AllowOverride*.

La primera permet definir quin és l'ordre d'actuació de les restriccions imposades (permetre i denegar), la segona especifica que el directori és accessible a tothom i la última informa al servidor que les opcions particulars “per directori” que es troben sota el mateix directori són les que s'apliquen finalment per l'autenticació d'usuaris. Això significa que el propietari de la pàgina web del departament de química pot imposar quins usuaris poden o no accedir a la pàgina demanant una autenticació (identificació amb usuari/password).

Les restriccions d'accés s'apliquen definint un fitxer *.htaccess* sota el directori on es vol aplicar la restricció. Mirem què conté el fitxer definit pel nostre suposat responsable de la pàgina web del departament de química:

```
AuthType Basic
AuthName "Departament de Química"
AuthUserFile /home/quimica/usuari/Auth
<files *>
require valid-user
</files>
```

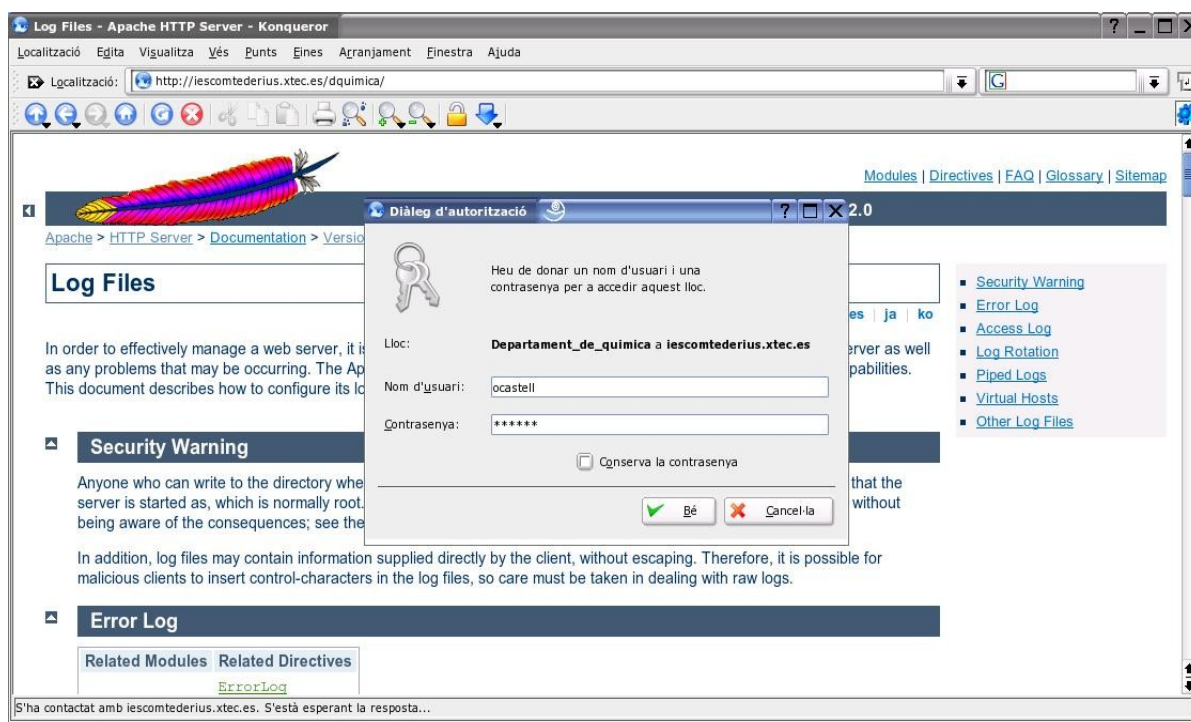
Està indicant al servidor WEB que per accedir als fitxers (<files *>) del departament cal una autenticació senzilla (amb usuari/paraula clau) per part del client i que els usuaris i paraula clau autoritzats estan dins d'un fitxer anomenat */home/quimica/usuari/Auth*.

Per crear els usuaris i les paraules claus corresponents cal utilitzar la comanda *htpasswd2*. Mirem com seria aquesta comanda:

```
xie45:/home/enxarxa # htpasswd2 -c /home/quimica/usuari/Auth ocastell
New password:
Re-type new password:
Adding password for user ocastell
xie45:/home/enxarxa #
```

El contingut del fitxer */home/quimica/usuari/Auth* seria: *ocastell:wNIOzxEZNKopA*. Això correspon al nom de l'usuari i al password (amb xifrat) que s'ha assignat.

El resultat per un usuari que es vulgui connectar a la pàgina web del departament és el següent:



5.4 Execució de programes en el servidor

Una de les possibilitats més interessants del servidor Apache és que permet definir permisos per executar comandes de la part del servidor.

Normalment, el servidor web únicament “serveix” informació en forma d'hipertext i imatges als clients que es volen connectar als seus recursos. Però es pot anar més enllà: sols cal donar una ullada al que pot fer WEBMIN per veure les possibilitats que tenim. El fet de crear pàgines interactives, a les que es pot donar una informació determinada, i que elles realitzen una tasca o altra segons les dades que s'han introduït, implica l'execució d'un programa en el servidor, apart del mateix servidor Apache.

La implementació d'un comptador de visites, d'un pàgina dinàmica, d'un formulari, impliquen l'activació dels CGI (Common Gateway Interface o interfície de passarel·la comú) al servidor. Això significa que, quan un client demana un d'aquest fitxers no es serveixen “tal qual”, sinó que s'executen al servidor per realitzar una funció que pot donar com a resultat una sortida en format HTML o no. Evidentment el servidor cal que tingui constància de quins són aquest fitxers i en quins directoris està permès això.

Per defecte és el directori *cgi-bin* de l'arrel del servidor web l'únic en el que es poden incloure fitxers per la seva execució. Es solen reconèixer per la seva extensió (*.pl, *.php, *.php3, *.py, *.sh) o pel seu encapçalament dins del fitxer, on s'especifica el metallenguatge utilitzant en la seva programació.

Per indicar si un directori té permès o no l'execució de programes al servidor (i l'assignació de les extensions reconegudes) s'utilitza la paraula clau d'opcions *Options* +/- *Valor* (com abans s'ha comentat). Els possibles valors de la paraula clau *Options* són els següents:

<i>Valor opció definida</i>	<i>Què vol dir</i>
All	Totes les opcions incloses excepte MultiViews
ExecCGI	Es permet l'execució de programes CGI
FollowSymLinks	El servidor permet links simbòlics al directori
Includes	Els SSI (Server Side Includes) són permesos al directori
IncludesNOEXEC	Els SSI (Server Side Includes) són permesos al directori, però les entrades <code>#exec cgi</code> i <code>#exec cmd</code> no estan permeses
Indexes	Si no existeix un <code>index.html</code> en aquest directori el servidor retorna un llistat formatat del seu contingut, en cas contrari un error
SymLinksIfOwnerMatch	El servidor permet links simbòlics al directori sempre que el propietari del link i del destí siguin el mateix
MultiViews	Està activat la negociació dels continguts per aquest directori o no

Per tant ens cal afegir `+ExecCGI` a la nostra línia d'opcions per el directori on volem introduir programes que s'han d'executar al servidor per part dels nostres visitants.

Abans de continuar cal fer una reflexió sobre el que pot o no significar activar l'execució de programes al servidor. En primera instància, el servei WEB “normal” representa una despesa molt petita de recursos del nostre servidor: un ordinador amb poca potència de càlcul i poca memòria ens pot oferir un servei raonable. Ara bé, en activar l'execució dels programes al servidor, que pot implicar, per exemple, negociacions de bases de dades (juntament amb un servidor MySQL i el PHP formant el que s'anomena LAMP o WAMP), pot arribar a multiplicar per mil les necessitats del nostre servidor (en potència de càlcul i en memòria). És necessari tenir ben present que una mala utilització dels recursos CGI pot implicar forats de seguretat importants en el servidor WEB que poden donar com a resultat la caiguda generalitzada del servei.

5.5 Servidors virtuals

Apache té la possibilitat de crear servidors virtuals. Els servidors virtuals són aquells que comparteixen l'adreça i la màquina amb altres servidors WEB. És a dir, per exemple, que si tenim un domini d'Internet amb el nom: <http://www.iesebre.com> que apunta a l'adreça IP 213.97.30.37 i volem que aquest servidor comparteixi l'adreça amb un altre servidor web corresponent al domini

<http://www.rekursoste.org> caldrà que definim un servidor virtual al nostre Apache. En ficar una o l'altra adreça, els DNS resoldran la mateixa IP però en arribar la petició del client al servidor WEB, com que a l'encapçalament porta el nom del domini, aquest haurà de ser capaç de diferenciar-los i de servir les pàgines que en corresponen a cadascun que, naturalment, són diferents. Això significa que podem disposar d'un sol servidor (màquina) que contingui en ell mateix a diversos dominis d'Internet amb pàgines web diferenciades amb opcions diferents, continguts diferents i restriccions particulars. Mirem què és el que volem dir. Fem un *ping* als dos dominis que s'han comentat per veure la resposta:

```

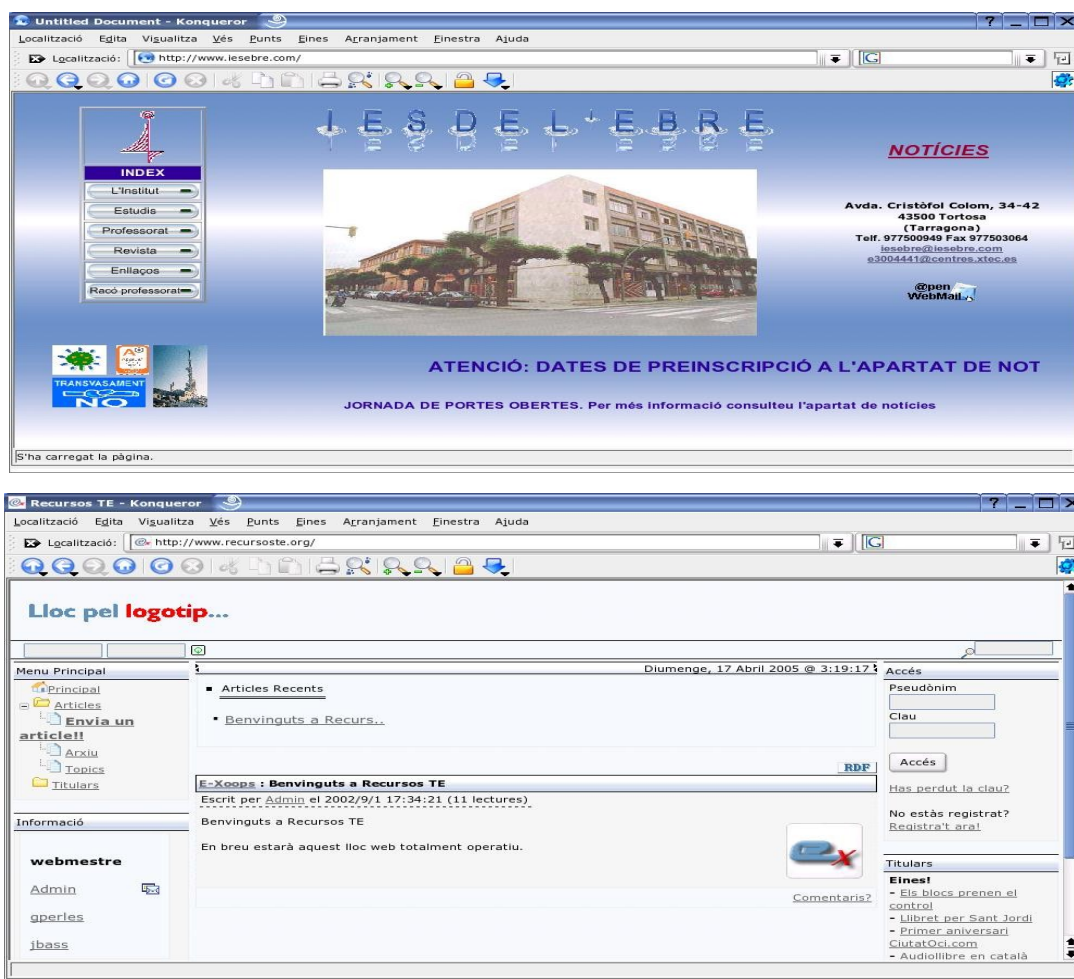
web@estacio-1:/etc/apache2 - Intèrpret de comandaments - Konsole
Sessió Edita Visualitza Punts Arranjament Ajuda

estacio-1:~ # ping www.iesebre.com
PING centre.iesebre.com (213.97.30.37) 56(84) bytes of data.
64 bytes from 37.Red-213-97-30.pooles.rina-tde.net (213.97.30.37): icmp_seq=1 ttl=253 time=104 ms
64 bytes from 37.Red-213-97-30.pooles.rina-tde.net (213.97.30.37): icmp_seq=2 ttl=253 time=104 ms

--- centre.iesebre.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5189ms
rtt min/avg/max/mdev = 104.266/104.612/104.959/0.473 ms
estacio-1:~ #
estacio-1:~ #
estacio-1:~ # ping www.rekursoste.org
PING www.rekursoste.org (213.97.30.37) 56(84) bytes of data.
64 bytes from 37.Red-213-97-30.pooles.rina-tde.net (213.97.30.37): icmp_seq=1 ttl=253 time=105 ms
64 bytes from 37.Red-213-97-30.pooles.rina-tde.net (213.97.30.37): icmp_seq=2 ttl=253 time=102 ms

--- www.rekursoste.org ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5185ms
rtt min/avg/max/mdev = 102.624/104.208/105.792/1.584 ms
estacio-1:~ #
  
```

En tots dos casos la IP de resposta és exactament la mateixa. El servidor al que apunten tots dos dominis és el mateix. Mirem què passa si fem una visita a la seva web:



Com es pot veure, són dues webs diferents, encara que estiguin “servides” pel mateix servidor Apache corrent en la mateixa màquina servidora. Per aquesta configuració cal definir els servidors virtuals. Aquest es poden definir per port o per nom:

- x Si la definició del servidor virtual és per port la connexió a aquest servidor es farà per un port diferent al 80 (què és l'estàndard). És a dir que definim un servidor virtual amb el mateix nom que el servidor principal però que escolta en un port diferent (8080, 9000), i que aquest respondrà a les peticions dels clients web que rebi el servidor per aquest port.
- x En la definició per nom el servidor és capaç de distingir d'encapçalament de la petició on va incorporat el nom del domini i s'encarrega de dirigir la informació i les opcions segons els fitxers de configuració corresponents al servidor virtual en qüestió. La petició es realitza sempre pel mateix port (80).

Per configurar un servidor virtual ens cal retocar alguns fitxers de configuració. El fitxer *httpd.conf* (que en la versió anterior d'Apache era el principal fitxer de configuració) conté una entrada que fa referència a la definició de servidors virtuals. En el nostre cas és:

```
Include /etc/apache2/vhosts.d/iesebre.conf
Include /etc/apache2/vhosts.d/recursoste.conf
NameVirtualHost *:80
```

Això ens permet indicar-li al servidor web que tenim dos servidors virtuals compartint el servei web en la mateixa màquina i escoltant pel mateix port (80). Aquest servidors virtuals tenen uns fitxers de configuració particulars que s'anomenen *iesebre.conf* i *recursoste.conf* que es troben sota el directori */etc/apache2/vhosts.d/*. Mirem el contingut d'aquest fitxers per saber com estan definits:

iesebre.conf

```
<VirtualHost *:80>
ServerAdmin ocastell@iesebre.com
ServerName iesebre.com
ServerAlias www.iesebre.com
ServerRoot /home/web
DocumentRoot /home/web/htdocs
ErrorLog /var/log/apache2/iesebre_error.log
CustomLog /var/log/apache2/iesebre_access.log combined
HostNameLookups off
UseCanonicalName off
ServerSignature On
ScriptAlias /cgi-bin/ "/home/web/cgi-bin/"
  <Directory "/home/web/cgi-bin">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
  </Directory>
  <IfModule mod_userdir.c>
    UserDir public_html
    Include /etc/apache2/mod_userdir.conf
  </IfModule>
  <Directory "/home/web/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
```



```
    Order allow,deny
    Allow from all
</Directory>
</VirtualHost>
```

recursoste.conf

```
<VirtualHost *:80>
ServerAdmin ocastell@recursoste.org
ServerName recursoste.org
ServerAlias www.recursoste.org
ServerRoot /home/recursoste
DocumentRoot /home/recursoste/htdocs
ErrorLog /var/log/apache2/recursoste_error.log
CustomLog /var/log/apache2/recursoste_access.log combined
HostNameLookups off
UseCanonicalName off
ServerSignature On
ScriptAlias /cgi-bin/ "/home/recursoste/cgi-bin/"
    <Directory "/home/recursoste/cgi-bin">
        AllowOverride None
        Options +ExecCGI -Includes
        Order allow,deny
        Allow from all
    </Directory>
    <IfModule mod_userdir.c>
        UserDir public_html
        Include /etc/apache2/mod_userdir.conf
    </IfModule>
    <Directory "/home/recursoste/htdocs">
        Options Indexes FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Com es pot observar, les diferències entre tots dos fitxers de configuració estan bàsicament en el nom del domini que es serveix i en la localització de l'arrel del servidor web. Per aquest motiu, tots dos dominis serveixen pàgines diferents. I les diferències no es queden a aquest nivell. Com es pot veure en les opcions dels fitxers anteriors, podríem definir directoris diferenciats per a les zones web dels usuaris que fossin d'un o d'altre domini, permisos d'accés diferents, execució de CGIs diferenciades i altres moltes opcions, que fan del servidor Apache2 una de les millors eleccions com a servidor WEB.

Com a pràctica d'aquest capítol al taller, els assistents al curs han de ser capaços de reproduir els casos explicats durant el mateix:

1. Cal que publiquen una pàgina web general i una per un usuari del sistema.
2. Han de definir un servidor virtual per nom i altre per port.
3. Han de crear un usuari web per modificar la pàgina principal.

Referències:

- [1] Projecte Apache. Pàgina WEB Oficial: <http://www.apache.org>
- [2] Estadístiques d'ús de servidors WEB: http://news.netcraft.com/archives/web_server_survey.html
- [3] Ús de versions Apache: http://news.netcraft.com/archives/web_server_survey.html