

Taller de Administración de Servidores Linux CC5308

Clase 18: 19 de abril de 2011

A PA_tCHy sE_rver

@chidalgo
chidalgo [at] nic . cl

Copyright © 2011
Creative Commons 3.0-cl by-nc-sa

Apache: instalación

- Forma fácil:
 - *yum install httpd*
 - *apt-get install apache2*
- Forma menos fácil:
 - Fuente: *httpd-2.X.XX.tar.gz*
 - *tar zxvf httpd-2.X.XX.tar.gz*
 - *./configure --prefix=/usr/local/apache2*
--enable-module=shared
--enable-modules="modulo1,modulo2"
 - *make && make install*

Apache: configuración

- Debian:
 - `/etc/apache2/apache2.conf`
 - `/etc/apache2/conf.d/*`
 - `/etc/apache2/mods-{available|enabled}`
 - `/etc/apache2/sites-{available|enabled}`
- RedHat
 - `/etc/httpd/conf/httpd.conf`
 - `/etc/httpd/conf.d/*`
 - `/etc/httpd/modules`

Apache: configuración

- Archivo configuración
 - ServerRoot “/etc/httpd”
 - Timeout 300
 - KeepAlive On
 - KeepAliveTimeout 15
 - <IfModule modulo.c>.....</IfModule>
 - LoadModule foo_module modules/mod_foo.so
 - Listen 192.168.0.1:80
 - User: apache
 - Group: apache
 - Include /etc/httpd/conf.d/
 - Include /etc/apache2/mods-enabled/{ *.load | *.conf }

Apache: configuración

- Logs:
 - ErrorLog /var/log/apache2/error.log
 - LogLevel warn
 - LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" **vhost_combined**
 - LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" **combined**
 - LogFormat "%h %l %u %t \"%r\" %>s %O" **common**
 - LogFormat "%{Referer}i -> %U" **referer**
 - LogFormat "%{User-agent}i" **agent**
 - CustomLog /var/log/apache2/other_vhosts_access.log **vhost_combined**

Apache: MPMs

- Multi-Processing Modules
 - Diferentes módulos, diferentes necesidades
 - Prefork (default)
 - Worker
 - Event
 - Itk
 - Otros... (distintos OS)

Apache: MPMs

- Prefork:

```
<IfModule prefork.c>  
StartServers      8  
MinSpareServers   5  
MaxSpareServers   20  
ServerLimit       256  
MaxClients        256  
MaxRequestsPerChild 4000  
</IfModule>
```

Apache: MPMs

- Worker:

```
<IfModule worker.c>  
StartServers      2  
MaxClients       150  
MinSpareThreads  25  
MaxSpareThreads  75  
ThreadsPerChild  25  
MaxRequestsPerChild 0  
</IfModule>
```


Apache: .htaccess

- Directivas para cambiar el comportamiento de directorios particulares
 - Se escriben en archivo **.htaccess** del directorio en cuestión
 - conf: AllowOverride None
 - Deshabilita los .htaccess

```
<Directory />  
Allowoverride All  
</Directory>  
  
<Location />  
Options +IncludesNoExec -ExecCGI  
  
Options +ExecCGI  
AddHandler cgi-script cgi pl  
</Location>
```

Apache: mod_authz_host

- Autorización basada en hosts

```
SetEnvIf User-Agent ^KnockKnock/2\..0 let_me_in
<Directory /directoriosecreto>
Order Deny,Allow
Deny from all
Allow from 10.1
Allow from 10 172.20 192.168.2
Allow from 10.1.0.0/16
Allow from 2001:db8::a00:20ff:fea7:ccea
Allow from 2001:db8::a00:20ff:fea7:ccea/10
Allow from env=let_me_in
</Directory>
```

Apache: mod_auth_basic

- Autenticación básica
- conf: AllowOverride AuthConfig
- htpasswd
 - *htpasswd -c /usr/local/apache/passwd/passwords usuario*

```
<Directory /usr/local/apache/htdocs/intranet>  
AuthType Basic  
AuthName "Acceso a la Intranet"  
AuthUserFile /www/passwd/users  
AuthGroupFile /www/passwd/groups  
Require group customers  
Order allow,deny  
Allow from internal.com  
Satisfy any  
</Directory>
```

Apache: mod_auth_digest

- HTTP Digest (MD5) (RFC 2617)
- conf: AllowOverride AuthConfig
- htdigest
 - *htdigest -c /usr/local/apache/passwd/digest "Private" usuario*

```
<Location />  
  AuthName 'Private'  
  
  AuthType Digest  
  AuthDigestProvider file  
  AuthDigestDomain /  
  AuthUserFile /data/www/sitename.tld/.htdigest  
  
  Require valid-user  
</Location>
```

Apache: configuración

- VirtualHosts:
 - Antes que todo:
 - Listen 80
 - Listen 443
 - NameVirtualHost *:80
 - Para muchos host virtuales
 - Base de datos
 - MySQL
 - mod_rewrite
 - <http://httpd.apache.org/docs/2.0/vhosts/mass.html>

Apache: control

- Demonio: `man { httpd | apache2 }(8)`
 - Por lo general no se invoca directamente
 - `httpd -d /path/to/serverroot`
 - `httpd -D parámetro`
 - `httpd -l`
 - `httpd -L`
 - `httpd -M`
- Control: `man { apachectl | apache2ctl }(8)`
 - `apache2ctl start | stop | configtest | status | fullstatus`
 - Módulos compilados: `apache2ctl -l`

Apache: virtualhosts

- Puerto 80, con logs propios

```
<VirtualHost *:80>
  ServerName www.foo.com

  # if you want this vhost to listen to extra names, uncomment the next line
  # ServerAlias foo.com www.bar.com bar.com

  DocumentRoot /var/www/www.foo.com/htdocs

  CustomLog /var/log/apache/www.foo.com-access.log combined
  ErrorLog /var/log/apache/www.foo.com-error.log
</VirtualHost>
```

Apache: virtualhosts

- Puerto 80, varios Virtual Hosts

```
<VirtualHost *:80>
  ServerName www.foo.com

  # if you want this vhost to listen to extra names, uncomment the next line
  # ServerAlias foo.com www.bar.com bar.com

  DocumentRoot /var/www/www.foo.com/htdocs

  CustomLog /var/log/apache/www.foo.com-access.log combined
  ErrorLog /var/log/apache/www.foo.com-error.log
</VirtualHost>

<VirtualHost *:80>
  ServerName mail.foo.com

  DocumentRoot /var/www/mail.foo.com/htdocs

  CustomLog /var/log/apache/mail.foo.com-access.log combined
  ErrorLog /var/log/apache/mail.foo.com-error.log
</VirtualHost>
```


Apache: virtualhosts

- VirtualHost básico, con logging + auth

```
<VirtualHost *:80>
  ServerName www.foo.com
  DocumentRoot /var/www/www.foo.com/htdocs

  CustomLog /var/log/apache/www.foo.com-access.log combined
  ErrorLog /var/log/apache/www.foo.com-error.log

  AuthUserFile /var/www/www.foo.com/.htpasswd
  AuthType Basic
  AuthName "Authorised Users Only"
  Require valid-user
</VirtualHost>
```

Apache: virtualhosts

- VirtualHost con SSL

```
<VirtualHost _default_:443>  
# Only one virtual host allowed on this port, because name-based  
# virtual hosting doesn't work with SSL  
ServerName www.foo.com  
DocumentRoot /var/www/www.foo.com/htdocs  
  
CustomLog /var/log/apache/www.foo.com-access.log combined  
ErrorLog /var/log/apache/www.foo.com-error.log  
  
SSLEngine On  
  
</VirtualHost>
```

Apache: mod_ssl

- Encriptación SSL/TLS
 - LoadModule ssl_module modules/mod_ssl.so

```
NameVirtualHost 192.168.1.1:443
```

```
<VirtualHost 192.168.1.1:443>
```

```
    SSLEngine on
```

```
    SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
```

```
    SSLCertificateFile    /etc/ssl/star.example.com.crt
```

```
    SSLCertificateKeyFile /etc/ssl/star.example.com.key
```

```
    ServerName    "one.example.com"
```

```
    DocumentRoot  "/var/www/html/one"
```

```
    CustomLog     "/var/log/httpd/one-access.log" combined
```

```
    ErrorLog      "/var/log/httpd/one-error.log"
```

```
    <Directory /var/www/html>
```

```
        AllowOverride none
```

```
        Order Allow,Deny
```

```
        Allow from all
```

```
    </Directory>
```

```
</VirtualHost>
```

Apache: mod_rewrite

- Expresiones regulares en Apache!
 - Poderoso
 - Muy poderoso!
 - Gran poder, gran responsabilidad
 - OjO con la seguridad
- Posibles usos:
 - Atrapar paths inexistentes
 - VirtualHosting masivo
 - Robo de imágenes
 - Manipular el “query string”
 - <http://www.dominio.cl/script?name=juanito>

Apache: mod_rewrite

- Ejemplo: robo de imágenes

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !=""
RewriteCond %{HTTP_REFERER} !example\.com [NC]
RewriteRule \.(jpe?g|gif|png)$ - [F,NC]

RewriteCond %{REQUEST_URI} !=/images/go_away.png
RewriteRule \.(jpe?g|gif|png)$ /images/go_away.png [NC,L]
```

Apache: CGI

- Ejecución de scripts en el servidor
 - mod_cgi

```
ScriptAlias /cgi-bin/ /usr/local/apache2/cgi-bin/
```

```
<Directory /home/*/public_html/cgi-bin>  
Options ExecCGI  
SetHandler cgi-script  
</Directory>
```

- Alternativas:
 - mod_fastcgi
 - Interpretes persistentes (mod_perl, mod_python et al)

Apache: docs

- Dónde mirar?

- Doc oficial:

- <http://httpd.apache.org/docs/2.2/>

- Wiki:

- <http://wiki.apache.org/httpd/FrontPage>

- mod_rewrite

- <http://httpd.apache.org/docs/2.0/misc/rewriteguide.html>

- Docs de la distro

- Localización de archivos, paquetes, comandos

Actividad Personal

- Instalar apache
 - Ojalá de la forma no fácil ;)
- Ver qué módulos vienen
 - `apachectl -l`
- Activar módulos `auth_digest`, `auth_basic`, `mod_cgi`
- Lanzar demonio apache
- Crear un directorio con permisos para 1 IP y ejecución de CGIs
- Crear script “Hello World”, colocar dentro del directorio y llamarlo via browser

Actividad Personal

- “Hello World” en shell UNIX

```
#!/bin/sh
echo "Content-type: text/html\n"

# el HTML
echo "<html>"
echo "<head><title>Hola Mundo</title></head>"
echo "<body>"
echo "<h1>Hola Mundo!</h1>"
Echo ":)"
echo "</body>"
echo "</html>"
```