

CSN09101

Networked Services

Week 8: Essential Apache

Module Leader: Dr Gordon Russell

Lecturers: G. Russell

This lecture

- Configuring Apache
- Mod_rewrite
- Discussions

Configuring Apache

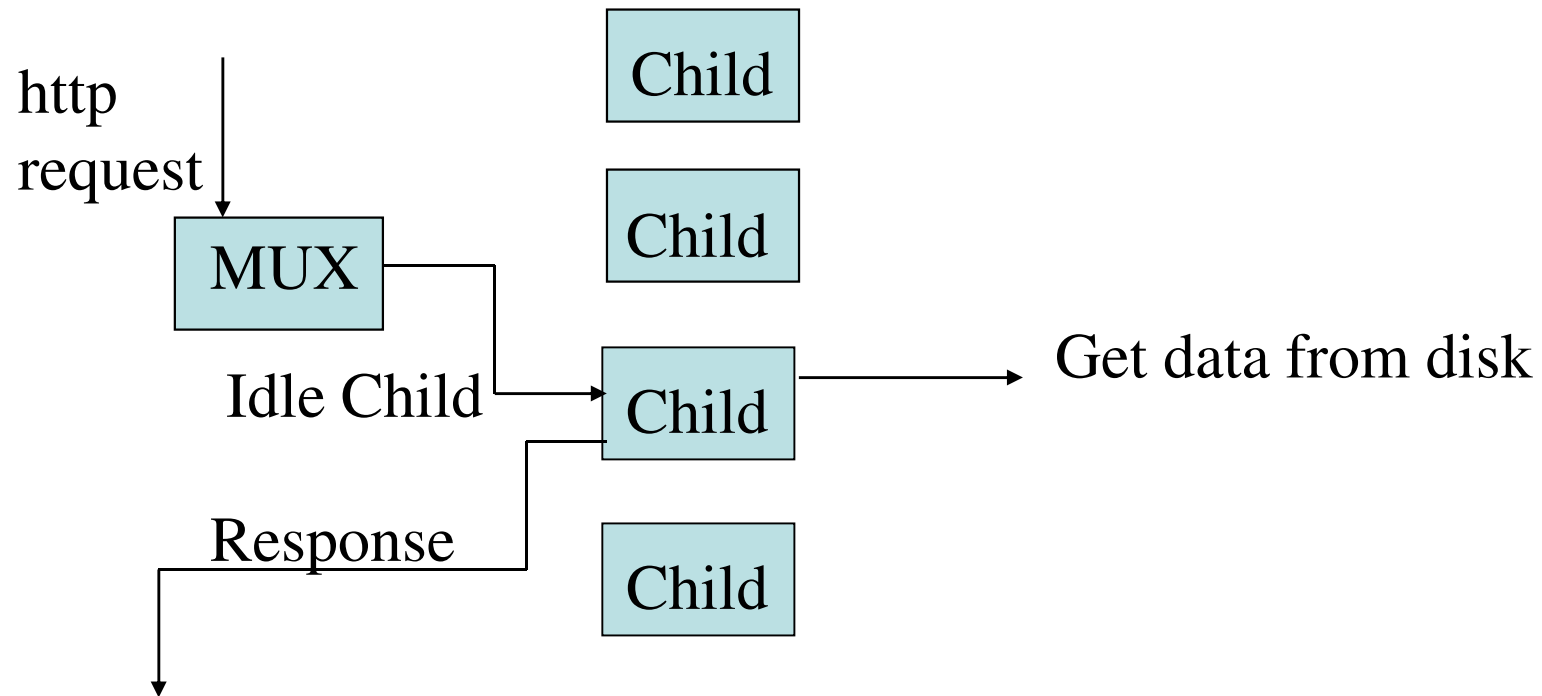
Apache

- Very well known and respected http server.
- Used commercially.
- Freely available from <http://www.apache.org>
- Plenty of plugins.
- Relatively easy and flexible to configure.
- Fast and Reliable.

Server Architectures

- In most designs of server, you either use
 - Threaded model
 - Forking model
 - Asynchronous Architecture
- A threaded model needs special OS support to provide lightweight threads. Not used in Apache for security and reliability reasons.
- Forking means that each new request which arrives is handled by a whole process. This is the Apache way.
- Asynchronous. Some web servers exist with this model, where one process handles everything with complex IO code. Good for fast processing of simple web pages.

Apache Forking Model



Initial Settings

StartServers	8
MinSpareServers	5
MaxSpareServers	20
MaxClients	150
MaxRequestsPerChild	1000

- These options are important, but often the least likely to be changed from the defaults!

Important Files

- `/etc/init.d/httpd` – the server control script
- `/etc/httpd/conf/http.conf` – the main conf file.

- Remember when changing the configurations it is only reread on a server reload or restart.
- Errors and other details are logged by default in `/var/log/httpd/` as `access_log`, `error_log`, as `suexec.log`.

Reload or Restart

- Reload is the best option to use.
- With a reload, apache checks your configuration file, and switches to it only if it contains no errors.
- If it has errors, it keeps using the old configuration.
- This allows you to reconfigure a server with no downtime.
- Restart shuts down then starts the server...
- Look in the error log for help (e.g. `/var/log/httpd/error_log`), or syslog (e.g. `/var/log/messages`).
- Remember to use the service command for this:
 - `Service httpd start|stop|reload|restart|status`
- You can easily make errors in the config file. You can check for errors using
 - `Service httpd configtest`

Mimic a Browser

- To understand how a sever is running is it sometimes useful to make requests at the keyboard of a server and see the results as text.
- Telnet can do this, so long as you have learned some basic HTTP commands.
- The two important ones are:
 - HEAD – Give information on a page.
 - GET – Give me the whole page.

- In HTTP 1.1 we can use virtual hosts.
- This allows multiple hosts to share a single server.
- Each host has a different name.
- The name of the host you want to answer a query is given as part of a page request.
- This is only supported in HTTP 1.1 and beyond.

```
$ telnet linuxzoo.net 80  
HEAD / HTTP/1.1  
Host: linuxzoo.net
```

```
HTTP/1.1 200 OK
```

```
Date: Mon, 01 Nov 2008 15:06:44 GMT
```

```
Server: Apache/2.0.46 (Red Hat)
```

```
Last-Modified: Fri, 29 Oct 2008 14:47:22 GMT
```

```
ETag: "4981dd-920-22ea7280"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 2336
```

```
Content-Type: text/html; charset=UTF-8
```

```
$ telnet linuxzoo.net 80  
HEAD / HTTP/1.1  
Host: db.grussell.org
```

HTTP/1.1 200 OK

Date: Mon, 01 Nov 2008 15:08:52 GMT

Server: Apache/2.0.46 (Red Hat)

Last-Modified: Thu, 21 Oct 2008 09:12:33 GMT

ETag: "3c8066-a37-86c9a240"

Accept-Ranges: bytes

Content-Length: 2615

Content-Type: text/html; charset=UTF-8

VirtualHosts

- The sharing of a single IP to provide multiple hostnames is well supported in Apache.
- The part of the conf file which handles this is called <VirtualHost>
- Each part holds a list of hostnames it can handle
- The first host found in the file is always considered the default, so if no VirtualHost section matches the first block is done instead.

```
<VirtualHost>
```

```
ServerAdmin me@grussell.org
```

```
DocumentRoot /home/gordon/public_html
```

```
ServerName grussell.org
```

```
ServerAlias www.grussell.org grussell.org.uk
```

```
ErrorLog logs/gr-error_log
```

```
CustomLog logs/gr-access_log combined
```

```
</VirtualHost>
```

public_html

- Where apache runs on a server used by many different servers, it would be useful for each user to be able to build their own web pages which the server could serve.
- But the virtualhost configuration takes only a single document root, and each user has their own directories in /home.
- You could make the root /home
 - All of the files in /home would be accessible, not just web pages.
 - It's a bit disgusting...
- Instead, apache supports web pages appearing in a users home directory, under the subdirectory public_html.

public_html access

- Urls of the form
 - `http://linuxzoo.net/~gordon/file.html`
- Refer to
 - `/home/gordon/public_html/file.html`
- This feature must first be switched on in `httpd.conf`.
- To activate it, find the line
 - `UserDir disable`
- Then either delete the line, or put “#” (the comment character) in front of it.
- Then find the following line and delete the ‘#’ character.
 - `#UserDir public_html`
- Remember to reload the server.

Linuxzoo tutorials

- Each time you book a linuxzoo machine, you will likely get a different IP and hostname.
- Each time you come in, check your hostname with “hostname”.

```
$ hostname
```

```
host-5-5.linuxzoo.net
```

- In this example, virtual hosts `vm-5-5.linuxzoo.net`, as well as `host-5-5` and `web-5-5` will be proxied to your machine.
- Warning: If the server on which your virtual machine fails, you will be moved to a different machine and a different IP. You need to check your hostname when you boot!

Web access from the prompt

- The prompt is fast and convenient for admin purposes, but when you are debugging http sometimes “telnet” is not sufficient.
- There are a few other tools you can use at the prompt.
 - elinks
 - lwp-request
 - wget
- However, there is no simple replacement for actually using a real browser to check your pages.

\$ elinks http://linuxzoo.net



```
root@lzmain:/var/named/chroot/var/named
Welcome to linuxzoo

If you can see this check that

* Javascript is enabled

Linuxzoo Penguin Icon

Welcome to linuxzoo

Learn Linux from the safety of your chair using a remote private linux
machine with root access.

* Welcome to linuxzoo
* Our environment
* Essential Linux
* System Administration

Status: Web system is operating normally. FREE server restored and in
testing. Everything should be back to normal.

Look at the Our Environment link, and then Running Your Machine for
getting started.

Quick start hints: register/login, Join Queue, Switch On (in Control tab),
Wait for successful boot, click the Connect tab, and then click "telnet:
linuxzoo.net" (or type telnet linuxzoo.net at your command prompt).
Username root, password secure.

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Tutlinks:  intro1 intro2 wildcard permission pipe vi essential admin net
           fwall DNS diag Apache1 Apache2 MySQL1 MySQL2

Useful:    Quiz Forums

Site Links: XMLZoo ActiveSQL ProgZoo SQLZoo

http://linuxzoo.net/page/start.html [-----]
```

Copy http to your directory

- `lwp-request http://linuxzoo.net > file.html`
 - The data is obtained and then printed to the screen.
 - In this case that is redirected to `file.html`
- `wget http://linuxzoo.net`

```
$ wget http://linuxzoo.net
--19:20:11-- http://linuxzoo.net/
Resolving linuxzoo.net... 146.176.166.1
Connecting to linuxzoo.net|146.176.166.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4785 (4.7K) [text/html]
Saving to: `index.html'
100%[=====>] 4,785    --.-K/s  in 0s
19:20:11 (304 MB/s) - `index.html' saved [4785/4785]
```

SELinux and Apache

- SELinux secures apache, and SELinux security of files in public_html is by default quite strong.
- Check if SELinux allows files to be published from public_html by
 - `getsebool httpd_read_user_content`
 - If this is 0 then publishing files is forbidden.
- Set SELinux to allow public_html publishing using:
 - `setsebool -P httpd_read_user_content 1`
 - This may take 20 or more seconds. Be patient.
 - The setting will be forgotten if you get a new image in the linuxzoo interface.
- SELinux requires the file security (shown by `ls -Z`) to be:
 - `unconfined_u:object_r:httpd_user_content_t:s0`
 - However this should happen automatically provided you create files in public_html
 - You can set the type of say filename.html (but remember you should not have to) using:
 - `chcon -t httpd_user_content_t filename.html`

mod_rewrite

URL Rewriting

- A useful module in apache is `mod_rewrite`.
- This allows us to change URLs dynamically.
- This can be useful to, for example,
 - Change the URL of aliases in a domain so that they always give the name you want.
 - Support directories and files being moved without breaking bookmarked URLs.
 - Provide a variety of proxying methods.

Methods

- `mod_rewrite` has many functions...
- The key functions are:
 - `RewriteCondition` – an IF statement
 - `RewriteRule` – an action (`doit`) statement.
- These can be placed almost anywhere in the apache configuration files.
- We will concentrate on their use in `VirtualHost` areas of `httpd.conf`.
- To work, the area must also have:
 - `RewriteEngine on`

rewriteRule

- Basic for of this rule is:

```
RewriteRule URL-reg-exp New-URL
```

- For instance, you have moved /old.txt to /new.txt

```
RewriteRule /old.txt /new.txt
```

Regular Expressions

- The match comparison is a regular expression.
- Useful aspects of regular expressions include:
- Text matching:
 - . Any single Character
 - [chars] One of the characters in chars
 - [^chars] None of the characters in chars
 - Text1|Text2 Either “Text1” or “Text2”

Quantifiers and Grouping

- Quantifiers:
 - ? 0 or 1 of the preceding text
 - * 0 or N of the preceding text
 - + 1 or N of the preceding text
- Grouping
(text) A text group – Can mark the border of an alternative or for RHS reference as \$N

Anchors and Escaping

- Anchors:

^ Start of the URL

\$ End of the URL

- Escaping

\char Allows you to use a character as the “char”. For instance, \^ is the ^ character and not the start of the URL.

Back References

- $\$N$ corresponds to a group from the URL match.
- For example, rewrite any URL ending in .txt to .html one could write:

```
RewriteRule      (.*)\.txt      $1.html
```

More complex example.

- Rewrite a URL ending with directory /demo/ to use /hia/ instead...

```
RewriteRule ^(.*)/demo/(.*)$ $1/hia/$2
```

Additional Flags

- At the end of the RewriteRule can be a number of flags.
- The Flags are listed in [brackets], eg [F,G] for flags F and G.
- These change or enhance the behaviour of the match.

Options:

- R or R=code - This sends the browser the new URL as an external REDIRECTION. The code can be the type or redirection, such as 302 for MOVED TEMPORARILY (the default).
- F - Send back FORBIDDEN.
- G - Send back GONE
- P - Proxy – Forward the request
- L - Last – do not look at any more rules.

Options Cont...

- C - chain – If the pattern matches do the next rule, otherwise ignore the remaining rules.
- NC - case insensitive.
- There are many more options, but these are the important ones.

Complex example

- If the URL has /work/ in it, rewrite /work/ to /home/.
- In addition, if the URL did have /work/ in it, replace “hello.txt” with “bye.txt”.

RewriteRule ^(.*)/work/(.*)\$ \$1/home/\$2 [C]

RewriteRule ^(.*)hello.txt\$ \$1/bye.txt [L]

RewriteCond

- This command performs tests or RULES.
- If the test matches, then the next test is checked.
- If all tests match, then the RewriteRule which follows the tests is performed.
- If any Cond does not match, processing skips on till after the Rule(s) in this block.

- Basic Form of RewriteCond

RewriteCond TestString ConditionString

- The value of the TestString is compared to the conditionstring.
- Condition String can be any type of regular expression.
- TestString can be one of a huge variety of things, including variables and file tests.

Variables:

- Here are some of the important variables:
 - REMOTE_ADDR
 - REMOTE_HOST
 - HTTP_HOST
 - REQUEST_URI (e.g. /index.html) (Its URI not URL)
 - REQUEST_FILENAME (e.g. /home/gordon/...)
- You use these as `%{REMOTE_ADDR}` etc.
- There are over 20 variables available.

Flags

- RewriteCond can take flags in the same way as RewriteRule.
- There are only 2 flags:
 - NC – case insensitive
 - OR – or the Conds together.
- Normally all rules have to be true before the Rule is done, with OR the rule is done if ANY Cond is true.

Example 1:

- If 10.20.0.5 tries to view /gordon/index.html, redirect the page reference to /gordon/bye.html.

```
RewriteCond %{REMOTE_ADDR} ^10\.20\.0\.5$
```

```
RewriteRule ^/gordon/index.html$ /gordon/bye.html [L]
```


Example 2:

- My VirtualHost has grussell.org, www.grussell.org, and www.grussell.org.uk.
- Rewrite all requests to grussell.org.

RewriteEngine on

RewriteCond %{HTTP_HOST} !^grussell\.org\$

RewriteRule ^(.*)\$ http://grussell.org\$1 [L,R]

Example 3:

- Rewrite *.grussell.org to grussell.org, and *.grussell.org.uk to grussell.org.uk.

RewriteEngine on

RewriteCond %{HTTP_HOST} ^.+grussell\.org\$

RewriteRule ^(.*)\$ http://grussell.org\$1 [L,R]

RewriteCond %{HTTP_HOST} ^.+grussell\.org\.uk\$

RewriteRule ^(.*)\$ http://grussell.org.uk\$1 [L,R]

Discussions

Discussion

- Apache runs as a user, usually “apache” or “httpd”. For apache to serve a file from a user’s public_html directory, what permissions would be required?

Discussion

- Here are some mock exam questions you should now be able to answer:

Question 1

- To test a web server which is hosting the virtual host “grussell.org”, using only telnet, what would you type at the telnet prompt?

Question 2

What fields would you expect to have to define in a VirtualHost definition in apache?

Question 3

Supply `mod_rewrite` instructions such that a request for `http://grussell.org/~uta` gets redirected externally and permanently to `http://upriss.org.uk`.