Linux Advanced Systems Administration

Web Server - Virtual Host facility

- Our current Apache configuration supports a single web site, using the hostname as the URL
- In practise, most web servers would have to support many more than this, so how could this be achieved?:-
    - * By having additional copies of *httpd,* each one looking after its own site.
    - * By having a single copy of *httpd* that manages more than one site.
- The single copy option can be achieved using Apache's Virtual Hosts facility, and this is how the system should be configured.
- Virtual hosts can be IP-based or Name based (The preferred method)
- Imagine that we now have an additional new web site directory under */var/www* called *lintrain*.
- So we now have two sites:-
    - * Our original */var/www/website/docs,* accessible simply as *http://hostname*
    - * A new web site directory */var/www/lintrain/docs* with no server at present.
- Whar we will do is use the Virtual Hosts facility to serve each site, and use more conventional web server names.

Web Server - Virtual Host facility

- We need to place entries in the *httpd.conf* files for each site, as follows:- (Note that there are preferred places where these should be placed - have a look for the examples already in the file)

```
NameVirtualHost     192.168.200.100
<VirtualHost        www.fatrain.com>
ServerAdmin         root@carlton
DocumentRoot        /var/www/website/docs
ServerName          www.fatrain.com
ErrorLog            /var/www/logs/website_error_log
TransferLog         /var/www/logs/website_xfer_log
</VirtualHost>
<VirtualHost        www.lintrain.com>
ServerAdmin         root@carlton
DocumentRoot        /var/www/lintrain/docs
ServerName          www.lintrain.com
ErrorLog            /var/www/logs/lintrain_error_log
TransferLog         /var/www/logs/lintrain_xfer_log
</VirtualHost>
```

* The */var/www/logs* directory must be created.

* The *NameVirtualHost* directive specifies the IP address of the web server (In the example, this is *carlton* on 192.168.200.100)

* The hostnames *www.fatrain.com* and *www.lintrain.com* must be in */etc/hosts*

Web Server - Virtual Host facility

- Example hosts entry:-

```
192.168.200.100  carlton  www.lintrain.com   www.fatrain.com
```

- We must also add entries to *httpd.conf* to ensure that the sites are accessible:-

```
<Directory "/var/www/website">
   Options Indexes Includes FollowSymLinks
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
<Directory "/var/www/lintrain">
   Options Indexes Includes FollowSymLinks
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
```

  * Place these just after the VirtualHost entries just inserted.

- Note: Remove the original *DocumentRoot* directive.
- Sample sites are available under */net/duck/software/www*, in directories *website* (the existing site) and *lintrain*.
- Please use these to set up your own Apache virtual hosts, but use different hostnames to *www.fatrain.com* and *www.lintrain.com*. (PTO)

Web Server - Virtual Host facility - Notes

- As well as using the VirtualHost directive, we have also used a few other entries which have not been explained.
- We will be taking a look at these, including the use of CGI scripts, very soon, so please just accept their use for now!

Linux Advanced Systems Administration

Web Server - Controlling Virtual Hosts

- When running the web server with a number of web sites, httpd will start a number of child versions of itself to cater for the additional requests.
- There are a number of directives which can be used to set limits:-

MaxClients n

* Effectively limits the number of *httpd* servers which can be running. Default 150.

MaxRequestsPerChild n

* Each invocation of *httpd* can handle only this number of requests, then it is killed. This guards against software errors in *httpd* having too much effect. Default 100.

MaxSpareServers n
MinSpareServers n

* Keep at least MinSpareServers always active (default 5); do not allow more than MaxSpareServers to be waiting (DEFAULT 20)

StartServers n

* Start *StartServers* processes when *hpptd* is initially invoked, i.e. do not wait for demand to build up before starting them.

Linux Advanced Systems Administration

Web Server - CGI Scripts

- CGI (Common Gateway Interface) scripts are an important part of the webserver setup.
- CGI scripts, often written in *perl*, are run as a result of user interaction with the web site.
- This is an example HTML web page which activates a CGI script:-

```
<HTML>
<HEAD>
<TITLE>Training Example </TITLE>
</HEAD>
<BODY BGCOLOR=white text=blue link=green vlink=red>
<CENTER><H1>Training example </H1>
<FORM method=get action="http:/cgi-bin/demo.pl">
Please enter first data field <INPUT name=cat><BR>
Please enter second data field <INPUT name=dog><BR>
Then select the go button to see what your perl program does
<INPUT TYPE=submit Value="go!">
</FORM>
</BODY>
</HTML>
```

- The CGI script being called is *demo.pl*, located (apparently) under */cgi-bin.*

Web Server - CGI Scripts

- This is the CGI script:-

```
#!/usr/bin/perl
$_ = $ENV{"QUERY_STRING"};
($cat,$dog) = /^cat=(.*)&dog=(.*)$/;

print <<"PAGE" ;
Content-type: text/html

<HTML><BODY BGCOLOR=white text=black>
The strings the perl program received were:<BR>
First field: $cat<BR>Second field: $dog<P>
</BODY></HTML>
PAGE
```

- The location of the HTML page is, of course, within a directory under the web site.
- CGI scripts can also be held within the web site directory, but this could cause security problems if the script sources are accessible from outside.
- Ideally, they should be held elsewhere on the web server, in a directory that cannot be accessed as part of the web site.
- In our previous (initial) example, we defined the location of this directory using the *ScriptAlias* directive.

Web Server - CGI Scripts - Location

- Here is how we defined it:-

```
ScriptAlias   /cgi-bin/   "/var/www/website/cgi-bin/"
```

* Then, just below:-

```
<Directory "/var/www/website/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

- When a web link to a CGI script specifies */cgi-bin* then the web server can locate it under the specified directory.
- To test this facility, we need to add our CGI script and web page to our existing setup - we will do this using the *lintrain* web site.

Linux Advanced Systems Administration

Web Server - CGI Scripts - Example

- To begin with, copy the script and the web page from *duck*:-

```
# cd   /net/duck/software/www
# ls
CGI-example   lintrain   website
# cp   -r   CGI-example   /var/www
# cd   /var/www/CGI-example
# cp   profile.html   ../lintrain/docs
# cp   demo.pl   ../cgi-bin
```

  *   Remember to make the script execuatble!

```
# chmod  +x   ../cgi-bin/demo.pl
```

- To keep things consistent, and avoid having multiple *cgi-bin* directories, we will move all the *website/cgi-bin* stuff into the *cgi-bin* in */var/www*:-

```
# cd  /var/www/website/cgi-bin
# mv   *   /var/www/cgi-bin
```

- Now edit the *index.html* under the *lintrain/docs* directory to add a link to the new web page:-

```
# cd  /var/www/lintrain/docs
# vi  index.html
```

  *   Add the following in some appropriate position:-

```
</P>
<P><B>
Click <A href="profile.html"><STRONG> HERE </STRONG></A> to see the cgi
example form.  </B>
</P>
```

Linux Advanced Systems Administration

Web Server - CGI Scripts

- Now we must edit *httpd.conf*:-

```
ScriptAlias   /cgi-bin/   "/var/www/website/cgi-bin/"
```

* Change to:-

```
ScriptAlias   /cgi-bin/   "/var/www/cgi-bin”
```

* Then, just below:-

```
<Directory "/var/www/website/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

* Change to:-

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

- Now stop and restart *httpd*:-

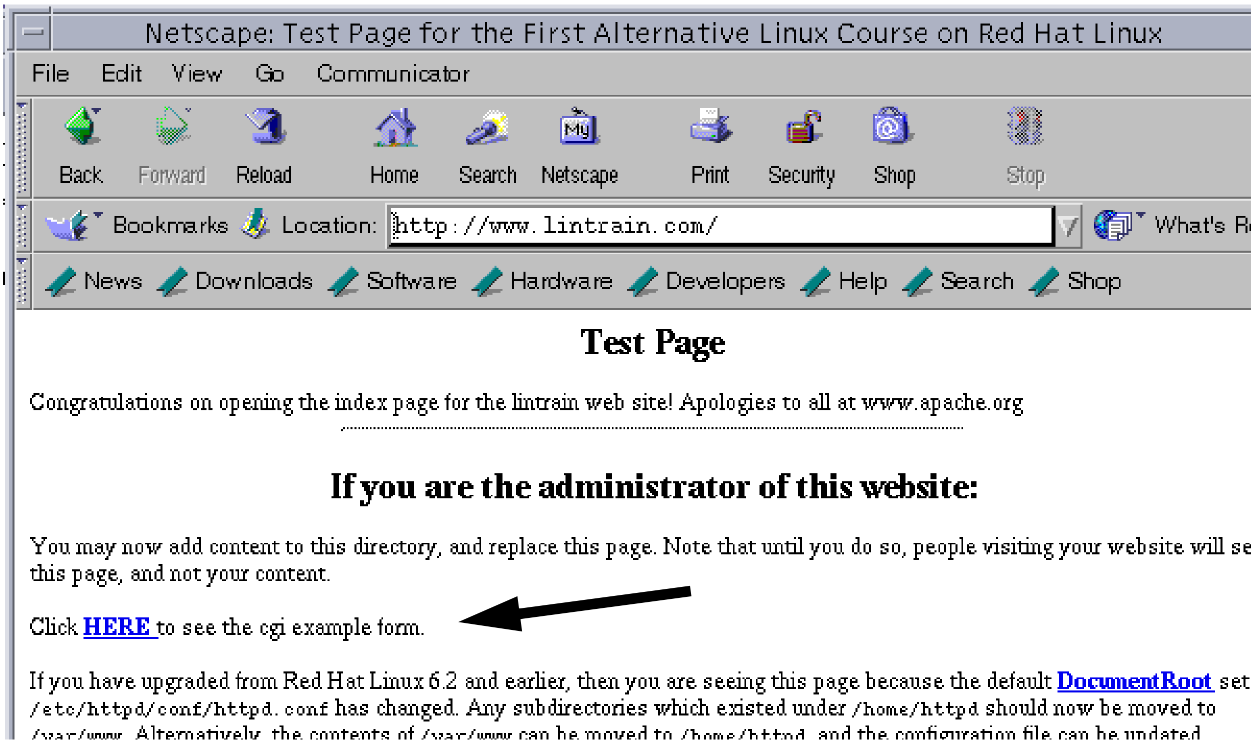
```
#  /etc/rc.d/init.d/httpd   stop

#  /etc/rc.d/init.d/httpd   start
```

- Now point your browser at *www.lintrain.com* once again:-

Linux Advanced Systems Administration

Web Server - CGI Scripts


Netscape: Test Page for the First Alternative Linux Course on Red Hat Linux
File Edit View Go Communicator
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.lintrain.com/
What's R
News Downloads Software Hardware Developers Help Search Shop

Test Page

Congratulations on opening the index page for the lintrain web site! Apologies to all at www.apache.org

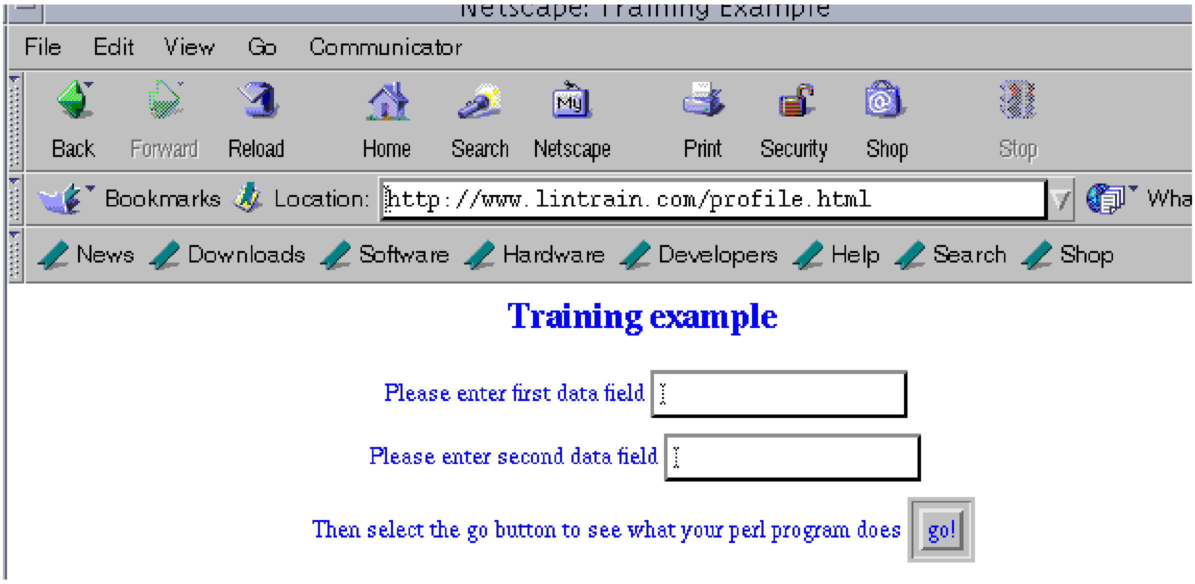
If you are the administrator of this website:

You may now add content to this directory, and replace this page. Note that until you do so, people visiting your website will se this page, and not your content.

Click HERE to see the cgi example form.

If you have upgraded from Red Hat Linux 6.2 and earlier, then you are seeing this page because the default DocumentRoot set /etc/httpd/conf/httpd.conf has changed. Any subdirectories which existed under /home/httpd should now be moved to /var/www. Alternatively, the contents of /var/www can be moved to /home/httpd, and the configuration file can be updated

- The newly-inserted link is indicated.
- Click on the link to see:-


Netscape: Training Example
File Edit View Go Communicator
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.lintrain.com/profile.html
Wha
News Downloads Software Hardware Developers Help Search Shop

Training example

Please enter first data field

Please enter second data field

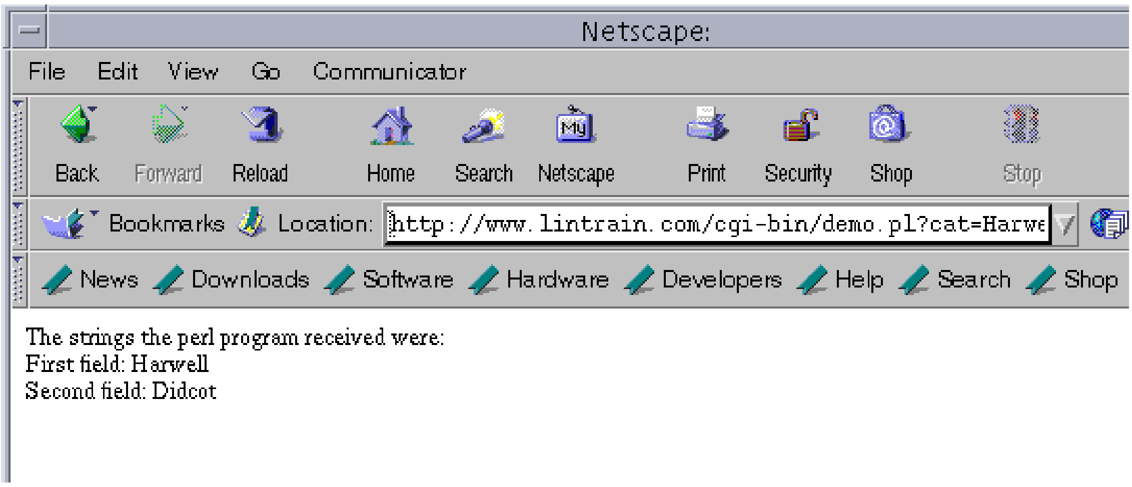
Then select the go button to see what your perl program does go!

- Complete the fields and click on Go.

Linux Advanced Systems Administration

Web Server - CGI Scripts

- I entered *Harwell* and *Didcot*, so I will see:-


Netscape:
File Edit View Go Communicator
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.lintrain.com/cgi-bin/demo.pl?cat=Harwe
News Downloads Software Hardware Developers Help Search Shop

The strings the perl program received were:
First field: Harwell
Second field: Didcot

- So we have now integrated the CGI scripting facility into our web server, and can now confidently add more scripts to make our site much more powerful.
- There is a lot to CGI script writing, which (sorry!) is beyond the scope of this course.