

webserver

Paul Cobbaut

webserver

Paul Cobbaut

lt-1.9

Publication date Wed 07 May 2014 06:44:07 PM CEST

Abstract

This book is meant to be used in an instructor-led training. For self-study, the intent is to read this book next to a working Linux computer so you can immediately do every subject, practicing each command.

This book is aimed at novice Linux system administrators (and might be interesting and useful for home users that want to know a bit more about their Linux system). However, this book is not meant as an introduction to Linux desktop applications like text editors, browsers, mail clients, multimedia or office applications.

More information and free .pdf available at <http://linux-training.be> .

Feel free to contact the author:

- Paul Cobbaut: paul.cobbaut@gmail.com, <http://www.linkedin.com/in/cobbaut>

Contributors to the Linux Training project are:

- Serge van Ginderachter: serge@ginsys.be, build scripts; infrastructure setup; minor stuff
- Hendrik De Vloed: hendrik.devloed@ugent.be, buildheader.pl script

We'd also like to thank our reviewers:

- Wouter Verhelst: wouter@grep.be, <http://grep.be>
- Geert Goossens: mail.goossens.geert@gmail.com, <http://www.linkedin.com/in/geert-goossens>
- Elie De Brauwer: elie@de-brauwer.be, <http://www.de-brauwer.be>
- Christophe Vandeplass: christophe@vandeplass.com, <http://christophe.vandeplass.com>
- Bert Desmet: bert@devnox.be, <http://bdesmet.be>
- Rich Yonts: richyonts@gmail.com,

Copyright 2007-2014 Paul Cobbaut

Permission is granted to copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License**, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled 'GNU Free Documentation License'.

Table of Contents

1. apache web server	1
1.1. introduction to apache	2
1.2. port virtual hosts on Debian	9
1.3. named virtual hosts on Debian	12
1.4. password protected website on Debian	14
1.5. troubleshooting apache on Debian	15
1.6. virtual hosts example	16
1.7. aliases and redirects	16
1.8. more on .htaccess	16
1.9. traffic	16
1.10. practice: apache	17
Index	18

Chapter 1. apache web server

In this chapter we learn how to setup a web server with the **apache** software.

According to NetCraft (http://news.netcraft.com/archives/web_server_survey.html) about seventy percent of all web servers are running on Apache. The name is derived from a **patchy** web server, because of all the patches people wrote for the NCSA httpd server.

Later chapters will expand this web server into a LAMP stack (Linux, Apache, Mysql, Perl/PHP/Python).

1.1. introduction to apache

1.1.1. installing on Debian

This screenshot shows that there is no **apache** server installed, nor does the **/var/www** directory exist.

```
root@debian7:~# ls -l /var/www
ls: cannot access /var/www: No such file or directory
root@debian7:~# dpkg -l | grep apache
```

To install **apache** on Debian:

```
root@debian7:~# aptitude install apache2
The following NEW packages will be installed:
  apache2 apache2-mpm-worker{a} apache2-utils{a} apache2.2-bin{a} apache2.2-com\
mon{a} libapr1{a} libaprutil1{a} libaprutil1-dbd-sqlite3{a} libaprutil1-ldap{a}\
ssl-cert{a}
0 packages upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,487 kB of archives. After unpacking 5,673 kB will be used.
Do you want to continue? [Y/n/?]
```

After installation, the same two commands as above will yield a different result:

```
root@debian7:~# ls -l /var/www
total 4
-rw-r--r-- 1 root root 177 Apr 29 11:55 index.html
root@debian7:~# dpkg -l | grep apache | tr -s ' '
ii apache2 2.2.22-13+deb7u1 amd64 Apache HTTP Server metapackage
ii apache2-mpm-worker 2.2.22-13+deb7u1 amd64 Apache HTTP Server - high speed th\
readed model
ii apache2-utils 2.2.22-13+deb7u1 amd64 utility programs for web servers
ii apache2.2-bin 2.2.22-13+deb7u1 amd64 Apache HTTP Server common binary files
ii apache2.2-common 2.2.22-13+deb7u1 amd64 Apache HTTP Server common files
```

1.1.2. installing on RHEL/CentOS

Note that Red Hat derived distributions use **httpd** as package and process name instead of **apache**.

To verify whether **apache** is installed in CentOS/RHEL:

```
[root@centos65 ~]# rpm -q httpd
package httpd is not installed
[root@centos65 ~]# ls -l /var/www
ls: cannot access /var/www: No such file or directory
```

To install apache on CentOS:

```
[root@centos65 ~]# yum install httpd
```

After running the **yum install httpd** command, the Centos 6.5 server has apache installed and the **/var/www** directory exists.

```
[root@centos65 ~]# rpm -q httpd
httpd-2.2.15-30.el6.centos.x86_64
[root@centos65 ~]# ls -l /var/www
total 16
drwxr-xr-x. 2 root root 4096 Apr  3 23:57 cgi-bin
drwxr-xr-x. 3 root root 4096 May  6 13:08 error
drwxr-xr-x. 2 root root 4096 Apr  3 23:57 html
drwxr-xr-x. 3 root root 4096 May  6 13:08 icons
[root@centos65 ~]#
```

1.1.3. running apache on Debian

This is how you start **apache2** on Debian.

```
root@debian7:~# service apache2 status
Apache2 is NOT running.
root@debian7:~# service apache2 start
Starting web server: apache2apache2: Could not reliably determine the server's \
fully qualified domain name, using 127.0.1.1 for ServerName
.
```

To verify, run the **service apache2 status** command again or use **ps**.

```
root@debian7:~# service apache2 status
Apache2 is running (pid 3680).
root@debian7:~# ps -C apache2
  PID TTY          TIME CMD
 3680 ?            00:00:00 apache2
 3683 ?            00:00:00 apache2
 3684 ?            00:00:00 apache2
 3685 ?            00:00:00 apache2
root@debian7:~#
```

Or use **wget** and **file** to verify that your web server serves an html document.

```
root@debian7:~# wget 127.0.0.1
--2014-05-06 13:27:02-- http://127.0.0.1/
Connecting to 127.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 177 [text/html]
Saving to: `index.html'

100%[=====>] 177  --.-K/s  in 0s

2014-05-06 13:27:02 (15.8 MB/s) - `index.html' saved [177/177]

root@debian7:~# file index.html
index.html: HTML document, ASCII text
root@debian7:~#
```

Or verify that apache is running by opening a web browser, and browse to the ip-address of your server. An Apache test page should be shown.

You can do the following to quickly avoid the 'could not reliably determine the fqdn' message when restarting apache.

```
root@debian7:~# echo ServerName Debian7 >> /etc/apache2/apache2.conf
root@debian7:~# service apache2 restart
Restarting web server: apache2 ... waiting .
root@debian7:~#
```


1.1.4. running apache on CentOS

Starting the **httpd** on RHEL/CentOS is done with the **service** command.

```
[root@centos65 ~]# service httpd status
httpd is stopped
[root@centos65 ~]# service httpd start
Starting httpd: httpd: Could not reliably determine the server's fully qualifie\
d domain name, using 127.0.0.1 for ServerName
[ OK ]
[root@centos65 ~]#
```

To verify that **apache** is running, use **ps** or issue the **service httpd status** command again.

```
[root@centos65 ~]# service httpd status
httpd (pid 2410) is running...
[root@centos65 ~]# ps -C httpd
  PID TTY          TIME CMD
 2410 ?            00:00:00 httpd
 2412 ?            00:00:00 httpd
 2413 ?            00:00:00 httpd
 2414 ?            00:00:00 httpd
 2415 ?            00:00:00 httpd
 2416 ?            00:00:00 httpd
 2417 ?            00:00:00 httpd
 2418 ?            00:00:00 httpd
 2419 ?            00:00:00 httpd
[root@centos65 ~]#
```

To prevent the 'Could not reliably determine the fqdn' message, issue the following command.

```
[root@centos65 ~]# echo ServerName Centos65 >> /etc/httpd/conf/httpd.conf
[root@centos65 ~]# service httpd restart
Stopping httpd: [ OK ]
Starting httpd: [ OK ]
[root@centos65 ~]#
```

1.1.5. index file on CentOS

CentOS does not provide a standard `index.html` or `index.php` file. A simple `wget` gives an error.

```
[root@centos65 ~]# wget 127.0.0.1
--2014-05-06 15:10:22-- http://127.0.0.1/
Connecting to 127.0.0.1:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2014-05-06 15:10:22 ERROR 403: Forbidden.
```

Instead when visiting the ip-address of your server in a web browser you get a **`noindex.html`** page. You can verify this using `wget`.

```
[root@centos65 ~]# wget http://127.0.0.1/error/noindex.html
--2014-05-06 15:16:05-- http://127.0.0.1/error/noindex.html
Connecting to 127.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5039 (4.9K) [text/html]
Saving to: "noindex.html"

100%[=====] 5,039      --.-K/s   in 0s

2014-05-06 15:16:05 (289 MB/s) - "noindex.html" saved [5039/5039]

[root@centos65 ~]# file noindex.html
noindex.html: HTML document text
[root@centos65 ~]#
```

Any custom **`index.html`** file in `/var/www/html` will immediately serve as an index for this web server.

```
[root@centos65 ~]# echo 'Welcome to my website' > /var/www/html/index.html
[root@centos65 ~]# wget http://127.0.0.1
--2014-05-06 15:19:16-- http://127.0.0.1/
Connecting to 127.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22 [text/html]
Saving to: "index.html"

100%[=====] 22          --.-K/s   in 0s

2014-05-06 15:19:16 (1.95 MB/s) - "index.html" saved [22/22]

[root@centos65 ~]# cat index.html
Welcome to my website
```

1.1.6. default website

Changing the default website of a freshly installed apache web server is easy. All you need to do is create (or change) an `index.html` file in the `DocumentRoot` directory.

To locate the `DocumentRoot` directory on Debian:

```
root@debian7:~# grep DocumentRoot /etc/apache2/sites-available/default
DocumentRoot /var/www
```

This means that `/var/www/index.html` is the default web site.

```
root@debian7:~# cat /var/www/index.html
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
root@debian7:~#
```

This screenshot shows how to locate the `DocumentRoot` directory on RHEL/CentOS.

```
[root@centos65 ~]# grep ^DocumentRoot /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
```

RHEL/CentOS have no default web page (only the `noindex.html` error page mentioned before). But an `index.html` file created in `/var/www/html/` will automatically be used as default page.

```
[root@centos65 ~]# echo '<html><head><title>Default website</title></head><body>\
><p>A new web page</p></body></html>' > /var/www/html/index.html
[root@centos65 ~]# cat /var/www/html/index.html
<html><head><title>Default website</title></head><body><p>A new web page</p></b\
ody></html>
[root@centos65 ~]#
```

1.1.7. apache configuration

There are many similarities, but also a couple of differences when configuring **apache** on Debian or on CentOS. Both Linux families will get their own chapters with examples.

All configuration on RHEL/CentOS is done in **/etc/httpd**.

```
[root@centos65 ~]# ls -l /etc/httpd/
total 8
drwxr-xr-x. 2 root root 4096 May  6 13:08 conf
drwxr-xr-x. 2 root root 4096 May  6 13:08 conf.d
lrwxrwxrwx. 1 root root   19 May  6 13:08 logs -> ../../var/log/httpd
lrwxrwxrwx. 1 root root   29 May  6 13:08 modules -> ../../usr/lib64/httpd/modu\
les
lrwxrwxrwx. 1 root root   19 May  6 13:08 run -> ../../var/run/httpd
[root@centos65 ~]#
```

Debian (and ubuntu/mint/...) use **/etc/apache2**.

```
root@debian7:~# ls -l /etc/apache2/
total 72
-rw-r--r-- 1 root root  9659 May  6 14:23 apache2.conf
drwxr-xr-x 2 root root  4096 May  6 13:19 conf.d
-rw-r--r-- 1 root root  1465 Jan 31 18:35 envvars
-rw-r--r-- 1 root root 31063 Jul 20  2013 magic
drwxr-xr-x 2 root root  4096 May  6 13:19 mods-available
drwxr-xr-x 2 root root  4096 May  6 13:19 mods-enabled
-rw-r--r-- 1 root root   750 Jan 26 12:13 ports.conf
drwxr-xr-x 2 root root  4096 May  6 13:19 sites-available
drwxr-xr-x 2 root root  4096 May  6 13:19 sites-enabled
root@debian7:~#
```

1.2. port virtual hosts on Debian

1.2.1. default virtual host

Debian has a virtualhost configuration file for its default website in `/etc/apache2/sites-available/default`.

```
root@debian7:~# head -2 /etc/apache2/sites-available/default
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
```

1.2.2. three extra virtual hosts

In this scenario we create three additional websites for three customers that share a clubhouse and want to jointly hire you. They are a model train club named **Choo Choo**, a chess club named **Chess Club 42** and a hackerspace named **hunter2**.

One way to put three websites on one web server, is to put each website on a different port. This screenshot shows three newly created **virtual hosts**, one for each customer.

```
root@debian7:~# vi /etc/apache2/sites-available/choochoo
root@debian7:~# cat /etc/apache2/sites-available/choochoo
<VirtualHost *:7000>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/choochoo
</VirtualHost>
root@debian7:~# vi /etc/apache2/sites-available/chessclub42
root@debian7:~# cat /etc/apache2/sites-available/chessclub42
<VirtualHost *:8000>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/chessclub42
</VirtualHost>
root@debian7:~# vi /etc/apache2/sites-available/hunter2
root@debian7:~# cat /etc/apache2/sites-available/hunter2
<VirtualHost *:9000>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/hunter2
</VirtualHost>
```

Notice the different port numbers 7000, 8000 and 9000. Notice also that we specified a unique **DocumentRoot** for each website.

1.2.3. three extra ports

We need to enable these three ports on apache in the `ports.conf` file.

```
root@debian7:~# vi /etc/apache2/ports.conf
root@debian7:~# grep ^Listen /etc/apache2/ports.conf
Listen 80
Listen 7000
Listen 8000
Listen 9000
```

1.2.4. three extra websites

Next we need to create three **DocumentRoot** directories.

```
root@debian7:~# mkdir /var/www/choochoo
root@debian7:~# mkdir /var/www/chessclub42
root@debian7:~# mkdir /var/www/hunter2
```

And we have to put some really simple website in those directories.

```
root@debian7:~# echo 'Choo Choo model train Choo Choo' > /var/www/choochoo/index.html
root@debian7:~# echo 'Welcome to chess club 42' > /var/www/chessclub42/index.html
root@debian7:~# echo 'HaCkInG iS fUn At HuNtEr2' > /var/www/hunter2/index.html
```

1.2.5. enabling extra websites

The last step is to enable the websites with the **a2ensite** command. This command will create links in **sites-enabled**.

The links are not there yet...

```
root@debian7:~# cd /etc/apache2/
root@debian7:/etc/apache2# ls sites-available/
chessclub42 choochoo default default-ssl hunter2
root@debian7:/etc/apache2# ls sites-enabled/
000-default
```

So we run the **a2ensite** command for all websites.

```
root@debian7:/etc/apache2# a2ensite choochoo
Enabling site choochoo.
To activate the new configuration, you need to run:
  service apache2 reload
root@debian7:/etc/apache2# a2ensite chessclub42
Enabling site chessclub42.
To activate the new configuration, you need to run:
  service apache2 reload
root@debian7:/etc/apache2# a2ensite hunter2
Enabling site hunter2.
To activate the new configuration, you need to run:
  service apache2 reload
```

The links are created, so we can tell **apache**.

```
root@debian7:/etc/apache2# ls sites-enabled/
000-default chessclub42 choochoo hunter2
root@debian7:/etc/apache2# service apache2 reload
Reloading web server config: apache2.
root@debian7:/etc/apache2#
```

1.2.6. testing the three websites

Testing the model train club named **Choo Choo** on port 7000.

```
root@debian7:/etc/apache2# wget 127.0.0.1:7000
--2014-05-06 21:16:03-- http://127.0.0.1:7000/
Connecting to 127.0.0.1:7000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32 [text/html]
Saving to: `index.html'

100%[=====>] 32          --.-K/s   in 0s

2014-05-06 21:16:03 (2.92 MB/s) - `index.html' saved [32/32]

root@debian7:/etc/apache2# cat index.html
Choo Choo model train Choo Choo
```

Testing the chess club named **Chess Club 42** on port 8000.

```
root@debian7:/etc/apache2# wget 127.0.0.1:8000
--2014-05-06 21:16:20-- http://127.0.0.1:8000/
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25 [text/html]
Saving to: `index.html.1'

100%[=====>] 25          --.-K/s   in 0s

2014-05-06 21:16:20 (2.16 MB/s) - `index.html.1' saved [25/25]

root@debian7:/etc/apache2# cat index.html.1
Welcome to chess club 42
```

Testing the hacker club named **hunter2** on port 9000.

```
root@debian7:/etc/apache2# wget 127.0.0.1:9000
--2014-05-06 21:16:30-- http://127.0.0.1:9000/
Connecting to 127.0.0.1:9000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26 [text/html]
Saving to: `index.html.2'

100%[=====>] 26          --.-K/s   in 0s

2014-05-06 21:16:30 (2.01 MB/s) - `index.html.2' saved [26/26]

root@debian7:/etc/apache2# cat index.html.2
HaCkInG iS fUn At HuNtEr2
```

Cleaning up the temporary files.

```
root@debian7:/etc/apache2# rm index.html index.html.1 index.html.2
```

Try testing from another computer using the ip-address of your server.

1.3. named virtual hosts on Debian

1.3.1. named virtual hosts

The chess club and the model train club find the port numbers too hard to remember. They would prefer to have their website accessible by name.

We continue work on the same server that has three websites on three ports. We need to make sure those websites are accessible using the names **choochoo.local**, **chessclub42.local** and **hunter2.local**.

We start by creating three new virtualhosts.

```
root@debian7:/etc/apache2/sites-available# vi choochoo.local
root@debian7:/etc/apache2/sites-available# vi chessclub42.local
root@debian7:/etc/apache2/sites-available# vi hunter2.local
root@debian7:/etc/apache2/sites-available# cat choochoo.local
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName choochoo.local
    DocumentRoot /var/www/choochoo
</VirtualHost>
root@debian7:/etc/apache2/sites-available# cat chessclub42.local
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName chessclub42.local
    DocumentRoot /var/www/chessclub42
</VirtualHost>
root@debian7:/etc/apache2/sites-available# cat hunter2.local
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName hunter2.local
    DocumentRoot /var/www/hunter2
</VirtualHost>
root@debian7:/etc/apache2/sites-available#
```

Notice that they all listen on **port 80** and have an extra **ServerName** directive.

1.3.2. name resolution

We need some way to resolve names. This can be done with DNS, which is discussed in another chapter. For this demo it is also possible to quickly add the three names to the **/etc/hosts** file.

```
root@debian7:/etc/apache2/sites-available# grep ^192 /etc/hosts
192.168.42.50 choochoo.local
192.168.42.50 chessclub42.local
192.168.42.50 hunter2.local
```

Note that you may have another ip address...

1.3.3. enabling virtual hosts

Next we enable them with **a2ensite**.

```
root@debian7:/etc/apache2/sites-available# a2ensite choochoo.local
Enabling site choochoo.local.
To activate the new configuration, you need to run:
    service apache2 reload
root@debian7:/etc/apache2/sites-available# a2ensite chessclub42.local
Enabling site chessclub42.local.
To activate the new configuration, you need to run:
    service apache2 reload
root@debian7:/etc/apache2/sites-available# a2ensite hunter2.local
Enabling site hunter2.local.
To activate the new configuration, you need to run:
    service apache2 reload
```

1.3.4. reload and verify

After a **service apache2 reload** the websites should be available by name.

```
root@debian7:/etc/apache2/sites-available# !se
service apache2 reload
Reloading web server config: apache2.
root@debian7:/etc/apache2/sites-available# wget chessclub42.local
--2014-05-06 21:37:13-- http://chessclub42.local/
Resolving chessclub42.local (chessclub42.local)... 192.168.42.50
Connecting to chessclub42.local (chessclub42.local)|192.168.42.50|:80... connec\
ted.
HTTP request sent, awaiting response... 200 OK
Length: 25 [text/html]
Saving to: `index.html'

100%[=====>] 25          --.-K/s   in 0s

2014-05-06 21:37:13 (2.06 MB/s) - `index.html' saved [25/25]

root@debian7:/etc/apache2/sites-available# cat index.html
Welcome to chess club 42
```

1.4. password protected website on Debian

You can secure files and directories in your website with a **.htaccess** file that refers to a **.htpasswd** file. The **htpasswd** command can create a **.htpasswd** file that contains a userid and an (encrypted) password.

This screenshot creates a user and password for the hacker named **cliff** and uses the **-c** flag to create the **.htpasswd** file.

```
root@debian7:~# htpasswd -c /var/www/.htpasswd cliff
New password:
Re-type new password:
Adding password for user cliff
root@debian7:~# cat /var/www/.htpasswd
cliff:$apr1$vuj1l0KL$./SZ4w9q0swhX93pQ0PVp.
```

Hacker **rob** also wants access, this screenshot shows how to add a second user and password to **.htpasswd**.

```
root@debian7:~# htpasswd /var/www/.htpasswd rob
New password:
Re-type new password:
Adding password for user rob
root@debian7:~# cat /var/www/.htpasswd
cliff:$apr1$vuj1l0KL$./SZ4w9q0swhX93pQ0PVp.
rob:$apr1$HN1n1FFt$nr1pF0H.IW11/1DRq4lQo0
```

Both Cliff and Rob chose the same password (hunter2), but that is not visible in the **.htpasswd** file because of the different salts.

Next we need to create a **.htaccess** file in the **DocumentRoot** of the website we want to protect. This screenshot shows an example.

```
root@debian7:~# cd /var/www/hunter2/
root@debian7:/var/www/hunter2# cat .htaccess
AuthUserFile /var/www/.htpasswd
AuthName "Members only!"
AuthType Basic
require valid-user
```

Note that we are protecting the website on **port 9000** that we created earlier.

And because we put the website for the Hackerspace named hunter2 in a subdirectory of the default website, we will need to adjust the **AllowOverride** parameter in **/etc/apache2/sites-available/default** as this screenshot shows (with line numbers on Debian7, your may vary).

```
9      <Directory /var/www/>
10          Options Indexes FollowSymLinks MultiViews
11          AllowOverride Authconfig
12          Order allow,deny
13          allow from all
14      </Directory
```

Now restart the apache2 server and test that it works!

1.5. troubleshooting apache on Debian

When apache restarts, it will verify the syntax of files in **/etc/apache2** and it will tell you the name of the faulty file, the line number and an explanation of the error.

```
root@debian7:~# service apache2 restart
apache2: Syntax error on line 268 of /etc/apache2/apache2.conf: Syntax error o\
n line 1 of /etc/apache2/sites-enabled/chessclub42: /etc/apache2/sites-enabled\
/chessclub42:4: <VirtualHost> was not closed.\n/etc/apache2/sites-enabled/ches\
club42:1: <VirtualHost> was not closed.
Action 'configtest' failed.
The Apache error log may have more information.
failed!
```

Below you see the problem... a missing / before on line 4.

```
root@debian7:~# cat /etc/apache2/sites-available/chessclub42
<VirtualHost *:8000>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/chessclub42
<VirtualHost>
```

Let us force another error by renaming the directory of one of our websites:

```
root@debian7:~# mv /var/www/choochoo/ /var/www/chooshoo
root@debian7:~# !ser
service apache2 restart
Restarting web server: apache2Warning: DocumentRoot [/var/www/choochoo] does no\
t exist
Warning: DocumentRoot [/var/www/choochoo] does not exist
... waiting Warning: DocumentRoot [/var/www/choochoo] does not exist
Warning: DocumentRoot [/var/www/choochoo] does not exist
.
```

As you can see, apache will tell you exactly what is wrong.

You can also troubleshoot by connecting to the website via a browser and then checking the apache log files in **/var/log/apache**.

1.6. virtual hosts example

Below is a sample virtual host configuration. This virtual hosts overrules the default Apache **ErrorDocument** directive.

```
<VirtualHost 83.217.76.245:80>
ServerName cobbaut.be
ServerAlias www.cobbaut.be
DocumentRoot /home/paul/public_html
ErrorLog /home/paul/logs/error_log
CustomLog /home/paul/logs/access_log common
ScriptAlias /cgi-bin/ /home/paul/cgi-bin/
<Directory /home/paul/public_html>
  Options Indexes IncludesNOEXEC FollowSymLinks
  allow from all
</Directory>
ErrorDocument 404 http://www.cobbaut.be/cobbaut.php
</VirtualHost>
```

1.7. aliases and redirects

Apache supports aliases for directories, like this example shows.

```
Alias /paul/ "/home/paul/public_html/"
```

Similarly, content can be redirected to another website or web server.

```
Redirect permanent /foo http://www.foo.com/bar
```

1.8. more on .htaccess

You can do much more with **.htaccess**. One example is to use **.htaccess** to prevent people from certain domains to access your website. Like in this case, where a number of referer spammers are blocked from the website.

```
paul@lounge:~/cobbaut.be$ cat .htaccess
# Options +FollowSymlinks
RewriteEngine On
RewriteCond %{HTTP_REFERER} ^http://(www\.)?buy-adipex.fw.nu.*$ [OR]
RewriteCond %{HTTP_REFERER} ^http://(www\.)?buy-levitra.asso.ws.*$ [NC,OR]
RewriteCond %{HTTP_REFERER} ^http://(www\.)?buy-tramadol.fw.nu.*$ [NC,OR]
RewriteCond %{HTTP_REFERER} ^http://(www\.)?buy-viagra.lookin.at.*$ [NC,OR]
...
RewriteCond %{HTTP_REFERER} ^http://(www\.)?www.healthinsurancehelp.net.*$ [NC]
RewriteRule .* - [F,L]
paul@lounge:~/cobbaut.be$
```

1.9. traffic

Apache keeps a log of all visitors. The **webalizer** is often used to parse this log into nice html statistics.

1.10. practice: apache

1. Verify that Apache is installed and running.
2. Browse to the Apache HTML manual.
3. Create three virtual hosts that listen on ports 8472, 31337 and 1201. Test that it all works.
4. Create three named virtual hosts startrek.local, starwars.local and stargate.local. Test that it all works.
5. Create a virtual hosts that listens on another ip-address.
6. Protect one of your websites with a user/password combo.

Index

Symbols

/etc/apache2, 8

/etc/httpd, 8

.htaccess, 16

.htpasswd, 14

A

apache2, 4

H

htpasswd(1), 14

httpd, 5

W

webalizer, 16