Chapter 4. eenvoudige webserver opzetten

Table of Contents

We ondernemen enkele stappen voor het opzetten van een eenvoudige Apache webserver.

We bestuderen kort hoe we software beheren op onze Ubuntu Linux en ook we logfiles kunnen bekijken. We hebben ook een plattetekst (ascii tekst bestanden worden in het Engels flat files genoemd) editor nodig om configuratiebestanden te beheren, en om een mini website te bouwen.

4.1. overzicht

Welke stappen hebben we nodig (met onze beperkte kennis van Linux) om een webserver op te zetten:

```
1. Hoe software installeren
2. Hoe een tekstbestand editeren
3. Hoe een kleine testwebpagina in html maken
4. Hoe een webserver configureren
5. Hoe een daemon (service) starten en stoppen
```

Voor stap 1 beperken we ons even tot software op onze virtuele machines in de klas. Stap 2 is een inleiding tot **vi**. Stap 3 gaat over enkel **tags** die nodig zijn om een browser iets te laten tonen. Stap 4 is ons doel en kan meteen een herhaling zijn vanwat we over http besproken hebben in de vorige module. En met een beetje geluk halen we vandaag al stap 5, maar dan hebben we wel hard gewerkt!

4.2. software beheren op Linux

We bekijken enkele commando's om software te beheren. In de Linux wereld heb je (grotendeels) twee formaten genaamd Debian packages en RPM Packages. Debian, Ubuntu, *buntu en Mint gebruiken allemaal packages die geleverd worden als een .deb bestand. De populairste tools om deze te beheren zijn **dpkg**, **apt-get**, **aptitude** en **synaptic**(grafisch). Elke recente distributie heeft wel een 'software center' in het grafische menu dat gebaseerd is op deze tools.

De basis tool is **dpkg**. Deze werkt met individuele packages en een niet-intuitieve syntax. Dankzij **apt-get** en **aptitude** hoeven we deze nauwelijks nog te kennen. Hieronder een screenshot van **dpkg -l** in combinatie met **grep** om na te kijken of **apache2** geinstalleerd is.

```
root@mac~# dpkg -l | grep apache2
ii apache2             2.2.20-1ubuntu1.2  Apache HTTP Server metapack...
ii apache2-mpm-worker 2.2.20-1ubuntu1.2  Apache HTTP Server - high s...
ii apache2-utils       2.2.20-1ubuntu1.2  utility programs for webser...
ii apache2.2-bin       2.2.20-1ubuntu1.2  Apache HTTP Server common b...
ii apache2.2-common    2.2.20-1ubuntu1.2  Apache HTTP Server common f...
```

In dit (geknipte) screenshot hierboven is apache reeds aanwezig op de server. Als dit niet het geval is, dan heb je volgende opties:

```
aptitude install apache2
apt-get install apache2
```

Beide commando's zullen het package genaamd **apache2** installeren incluis alle packages die nodig zijn (dependencies).

Je kan packages weer verwijderen met:

```
aptitude remove apache2
apt-get remove apache2
```

Meer info op linux-training.be in het boek **Linux System Administration**.

4.3. vi

De leraar legt de basis uit van **vi**. Deze editor is standaard aanwezig op zowat elke Unix en Linux sinds de jaren 70. Alhoewel moeilijk om te leren, is **vi** niet moeilijk om te gebruiken.

Op Ubuntu kan het handig zijn om eerst **aptitude install vim** uit te voeren. **vim** staat voor **vi improved** en heeft enkele voordelen voor beginners.

4.4. html

We weten nog dat een webserver html paginas stuurt naar een web browser. Om de goede werking van onze server te testen, hebben we dus een kleine test webpagina nodig.

Je kan het eenvoudig houden en dit werkende voorbeeld downloaden via:

```
wget linux-training.be/files/studentfiles/index.html
```

Of je kan zelf met **vi** een klein html bestand maken. Wat je nodig hebt zijn de tags **html**, **head**, **title** en **body**.

```
paul@mac~$ cat index.html
<html>
<head><title>litte test website</title></head>
<body>
<h1>Welcome to linux-training.be</h1>
<p>42</p>
</body>
</html>
```

4.5. apache2

We hebben van onze Ubuntu Linux een webserver gemaakt aan de hand van de volgende stappen.

Installatie van de webserver software genaamnd Apache.

```
apt-get update ; apt-get install apache2
```

of

```
aptitude update ; aptitude install apache2
```

Je kan testen of Apache geinstalleerd is met **dpkg** zoals vermeld hierboven.

Om de webserver te starten of te stoppen gebruiken we tegenwoordig het **service** commando:

```
root@ubu1110:~# service apache2 stop
 * Stopping web server apache2
... waiting                                                    [ OK ]
```

```
root@ubu1110:~# service apache2 start
 * Starting web server apache2                                    [ OK ]
root@ubu1110:~# service apache2 restart
 * Restarting web server apache2
... waiting                                                       [ OK ]
```

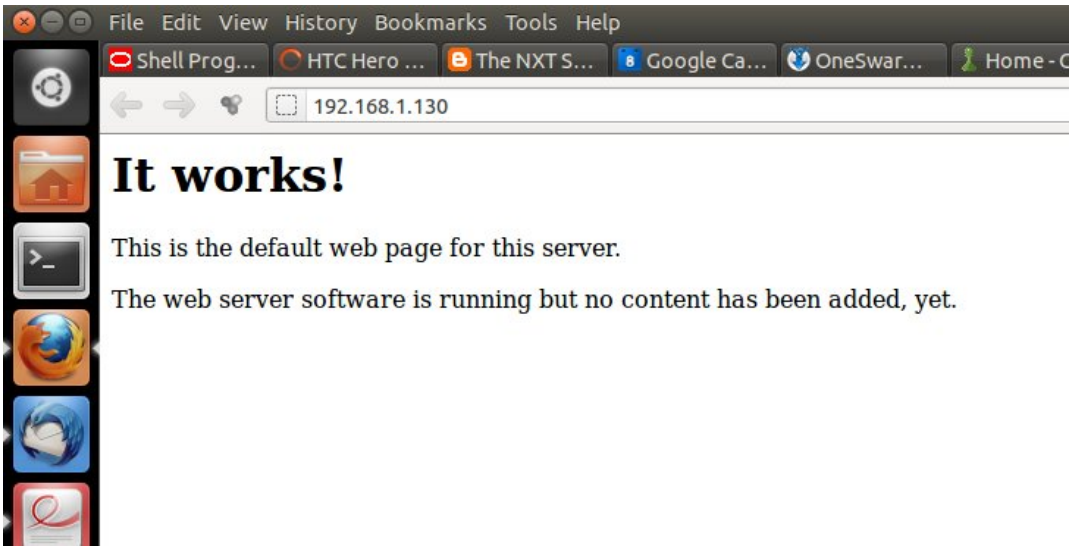
De oude manier van werken is door via **/etc/init.d/** rechtstreeks het script aan te spreken.

```
root@ubu1110:~# /etc/init.d/apache2 stop
 * Stopping web server apache2
... waiting                                                       [ OK ]
root@ubu1110:~# /etc/init.d/apache2 start
 * Starting web server apache2                                   [ OK ]
root@ubu1110:~# /etc/init.d/apache2 restart
 * Restarting web server apache2
... waiting .                                                     [ OK ]
root@ubu1110:~#
```

Het zou kunnen dat **Apache** bij de start komt klagen dat hij zijn ServerName niet vindt. Dit kan je oplossen door een ServerName te zetten in het **fqdn** bestand.

```
root@ubu1110:~# echo 'ServerName ubu1110.local' > /etc/apache2/conf.d/fqdn
root@ubu1110:~# cat /etc/apache2/conf.d/fqdn
ServerName ubu1110.local
```

Je kan nu surfen naar je eigen ip adres, en daar de default pagina van Apache2 zien.



Als dit niet lukt, herbekijk dan terug de vorige stappen alvorens verder te gaan!!

4.6. twee extra websites op de apache zetten

We hebben in de vorige module gezien dat na de tcp-handshake tussen web browser en webserver er een http request gaat van de client naar de server, met daarin behalve ip-adres en poort ook de naam van de gevraagde website. We gaan apache nu zo configureren dat hij aan de hand van de naam van de website gaat kiezen welke website hij toont.

We beginnen met twee namen te kiezen voor onze websites. Ik koos voor:

```
polsite1.local
polsite2.local
```

Opdat dit zou werken vanop een andere computer in het netwerk is het belangrijk dat die computer de namen kan vertalen in het correcte ip adres van onze webserver. (Dit kan je vinden door **ifconfig** te typen op de webserver). Ik zet de volgende informatie in de **/etc/hosts** file van de webserver en van de client van waarop ik de websites ga testen:

```
192.168.1.130   polsite1.local
192.168.1.130   polsite2.local
```

Let op, deze lijnen zijn **toegevoegd** aan /etc/hosts. Als je thuis een DNS server hetb staan, kan je daar twee A records toevoegen.

De volgende stap is het maken van twee websites. Hiervoor heb ik twee directories aangemaakt, en daarin telkens een unieke **index.html** gezet.

De eerste website:

```
root@ubu1110:~# cat /var/www/polsite1/index.html
<html>
<head><title>litte test website</title></head>
<body>
<h1>Welcome to polsite1</h1>
<p>1 1 1 1 1 1 1 1</p>
</body>
</html>
```

De tweede website:

```
root@ubu1110:~# cat /var/www/polsite2/index.html
<html>
<head><title>litte test website</title></head>
<body>
<h1>Welcome to polsite2</h1>
<p>2 2 2 2 2 2 2 2 2 2</p>
</body>
</html>
```

Als dit klaar is, komt het moeilijke stuk, de configuratie van Apache zelf aanpassen. Als je de man page leest, en voorbeelden op internet zoekt, dan zou dit moeten lukken. Als je ongeduldig bent, of niet zelf wil zoeken, kan je ook de volgende voorbeelden bekijken.

Apache voorziet een directory genaamd **sites-enabled** waar je per extra website een bestand kan zetten met de configuratie voor de website. Ik heb er de volgende twee bestanden gezet:

```
root@ubu1110:/etc/apache2/sites-enabled# cat polsite1
<VirtualHost *:80>
DocumentRoot /var/www/polsite1
ServerName polsite1.local
</VirtualHost>
root@ubu1110:/etc/apache2/sites-enabled# cat polsite2
<VirtualHost *:80>
DocumentRoot /var/www/polsite2
ServerName polsite2.local
</VirtualHost>
```

Nu Apache herstarten, en het zou moeten werken.

Probeer als oefening eens om een website te maken die wordt getoond op een andere poort.

4.7. log bestanden bekijken

Met het eerder geziene **tail** commando kunnen we de laatste berichten in een log bestand bekijken.

Hier bijvoorbeeld de laatste tien berichten van de dhcp server.

```
root@debian6:~# tail /var/log/dhcpd.log
Mar  8 12:47:29 gwen dhcpd: Dynamic and static leases present for 192.168.1.30.
Mar  8 12:47:29 gwen dhcpd: Remove host declaration mac or remove 192.168.1.30
Mar  8 12:47:29 gwen dhcpd: from the dynamic address pool for 192.168.1.0/24
Mar  8 12:47:29 gwen dhcpd: DHCPREQUEST for 192.168.1.30 from 00:26:bb:12:7a:5e via et
Mar  8 12:47:29 gwen dhcpd: DHCPACK on 192.168.1.30 to 00:26:bb:12:7a:5e via eth0
```

```
Mar  8 13:45:26 gwen dhcpd: Dynamic and static leases present for 192.168.1.30.
Mar  8 13:45:26 gwen dhcpd: Remove host declaration mac or remove 192.168.1.30
Mar  8 13:45:26 gwen dhcpd: from the dynamic address pool for 192.168.1.0/24
Mar  8 13:45:26 gwen dhcpd: DHCPREQUEST for 192.168.1.30 from 00:26:bb:12:7a:5e via et
Mar  8 13:45:26 gwen dhcpd: DHCPACK on 192.168.1.30 to 00:26:bb:12:7a:5e via eth0
```

De dns server draait onder de naam **named** (name daemon) en zet standaard zijn berichten in **/var/log/syslog**. Deze laatste is een algemen log voor zowat alles wat er op je Debian of Ubuntu server gebeurt.

Hieronder enkele berichten na een herstart van de dns server.

```
root@debian6:~# tail -8 /var/log/syslog.log
Mar  8 14:26:57 gwen named[4505]: zone 0.in-addr.arpa/IN: loaded serial 1
Mar  8 14:26:57 gwen named[4505]: zone 127.in-addr.arpa/IN: loaded serial 1
Mar  8 14:26:57 gwen named[4505]: zone 255.in-addr.arpa/IN: loaded serial 1
Mar  8 14:26:57 gwen named[4505]: zone clint.be/IN: loaded serial 2012082800
Mar  8 14:26:57 gwen named[4505]: zone netsec.local/IN: loaded serial 2012091100
Mar  8 14:26:57 gwen named[4505]: zone localhost/IN: loaded serial 2
Mar  8 14:26:57 gwen named[4505]: running
Mar  8 14:26:57 gwen named[4505]: zone netsec.local/IN: sending notifies (serial 20120
```