

Huiswerk Linux: Apache back-end configuratie

We hebben in de afgelopen weken een complete Apache installatie, inclusief twee websites geïnstalleerd. We gaan nu nog een stapje verder. De opdracht voor deze week is: configureer Apache als *back-end* server en installeer een tweede webserver welke gaat fungeren als *front-end* server.

Over Apache als *back-end*

PHP is een zeer populaire scripting-taal. Dit komt voornamelijk omdat bijna alle *Content Management Systems (CMS)*, zoals *WordPress*, *Joomla*, *Drupal* en het *Zend framework* deze scripting-taal gebruiken.

Het nadeel van PHP is dat *it sucks*¹. Big-time. Dit kun je letterlijk opvatten. Als je namelijk PHP geïnstalleerd hebt, aangevuld met bijvoorbeeld *Drupal*, dan gaat je opvallen dat Apache je systeem begint *leeg te zuigen*. Er vindt ineens heel erg veel *disk-activity* plaats en je *CPU* wordt bijna 100% in beslag genomen door Apache. Dit heeft tot gevolg dat alle Daemons op je systeem heel erg langzaam gaan werken. En als je PHP-website ook nog goed bezocht wordt, moet je rekening houden met *Denial Of Service (DOS)*, *freeze-ups* en *crashes*.

Hier zijn een aantal oplossingen voor. De meest voor de hand liggende is natuurlijk: gebruik geen PHP. Maar dat is vaak geen optie. Een oplossing die de laatste tijd steeds meer wordt gebruikt, is het plaatsen van een *front-end* webserver². De front-end is een tweede, supersnelle webserver, genaamd NGinX (spreek uit als: *Engine X*). Deze handelt alle webverkeer van buiten af en stuurt de bezoekers naar Apache (de *back-end*).

Het *sudo* mechanisme

Het configureren van de meeste Linux applicaties doen we namens de *root* gebruiker. We moeten daarom tijdelijk inloggen als Administrator (*root*).

Cygwin gebruikers

Sudo voor Cygwin gebruikers: rechts-klik op het icoon van de Cygwin terminal, en kies voor **Als administrator uitvoeren**. Als Apache nog niet op je systeem staat, update dan eerst de package-database:

```
apt-get update
```

En installeer vervolgens het *apache* pakket:

```
apt-get install apache
```

Start vervolgens de Apache webserver. Dit doen we met het *service* programma:

```
service apache start
```

Andere Linux gebruikers

Sudo voor **Debian** distributies (*Ubuntu*, *Knoppix*, *Kali*): start een terminal met de toetsencombinatie: **<Ctrl><Alt>-T**. We gebruiken het commando *sudo* alleen om in te loggen met het *su* (super user) commando. Daardoor blijven we ingelogd:

```
sudo su
```

Als Apache nog niet op je systeem staat, update dan eerst de package-database en installeer vervolgens het *apache2* en het *apache2-doc* pakket:

```
apt-get update  
apt-get install apache2 apache2-doc
```

Op **Slackware** distributies, zoals *Suse Linux*, heet de terminal *Konsole* en gebruiken we de *zypper* installer. Update eerst de package database en installeer vervolgens het *apache2* pakket:

```
zypper refresh  
zypper install apache2
```

¹ PHP is als programmeer-taal zeer omstreven. Geef in Google de zoekwoorden 'PHP sucks' in en je begrijpt wat hiermee bedoeld wordt.

² Een andere oplossing is een zogenaamde *dedicated server*. De website wordt dan op een aparte server geplaatst, waardoor de andere Daemons geen last hebben van suckage. Maar als de website erg goed bezocht wordt is ook dit niet meer voldoende.

De luisterende interfaces controleren

Allereerst gaan we kijken op welke interfaces Apache op dit moment luistert. Dit doen we met het *netstat* programma:

netstat -na

De optie *n* (numeric) laat alleen IP-adressen zien, geen domeinnamen. De optie *a* (all) geeft ook de luisterende processen weer. De output is van een *Cygwin* Linux systeem en ziet er ongeveer als volgt uit:

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	10.0.20.22:139	0.0.0.0:0	LISTENING
TCP	10.0.20.22:1025	10.0.20.26:445	ESTABLISHED
TCP	127.0.0.1:1033	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5206	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5354	0.0.0.0:0	LISTENING

We kunnen zien dat Apache (poort 80) op het IP-adres 0.0.0.0 “luistert”. Dit IP-adres is geen echt IP-adres, maar geeft aan dat Apache op *alle* interfaces luistert, dus zowel op de *localhost* (127.0.0.1) als je *wifi interface* en *ethernet interface*.

Apache configureren

We gaan ervoor zorgen dat Apache alleen op de *localhost* gaat luisteren. Hiervoor gebruiken we de Apache *Listen* optie. Deze kunnen we wijzigen via het Apache configuratie-bestand.

Cygwin gebruikers	Andere Linux gebruikers
<p>We openen het configuratiebestand met de <i>vi</i> tekstverwerker:</p> <p>vi /etc/httpd/conf/httpd.conf</p> <p>Om de juiste instelling te vinden kun je een zoekopdracht geven. Dit doe je met het zoek-symbool (“/”), gevolgd door de instelling die je zoekt:</p> <p>/Listen</p>	<p>Bij de overige Linux-smaken (<i>Ubuntu, Debian, Mint, etc.</i>) staat het bestand in de <i>/etc/apache2</i> directory. Het originele <i>httpd.conf</i> bestand is in tweeën opgesplitst: <i>apache2.conf</i> en <i>ports.conf</i>. We openen het configuratiebestand voor de poort-instellingen met de <i>vi</i> tekstverwerker:</p> <p>vi /etc/apache2/ports.conf</p>

Het bestand ziet er in iedere Linux-distributie anders uit, maar de instellingen zijn hetzelfde. Je ziet een paar voorbeeld-instellingen. Deze zijn te herkennen aan het hekje (“#”) dat ervoor staat. Daardoor zijn ze niet actief. Wij gaan onze eigen instelling toevoegen.

Toets **i** (insert) om naar de *INSERT* modus te gaan en voeg de vetgedrukte regel toe:

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
#Listen 3000
#Listen 12.34.56.78:80
#Listen dimension:80
Listen localhost:80
```

Let op: het kan zijn dat er al iets is ingesteld. Plaats dan een hekje voor de instelling, zodat deze wordt gede-activeerd (zie bovenstaand). Met de nieuwe instelling geven we aan dat Apache alleen op de *localhost* (127.0.0.1) en op poort 80 moet luisteren.

Toets tot slot <Esc> om uit de *INSERT* modus te komen en geef de commando's *w* (write) en *q* (quit):

```
:wq
```

Het configuratie-bestand wordt opgeslagen en we zijn weer terug op de Linux command-line.

Het hoera-moment

Om ervoor te zorgen dat de nieuwe instelling wordt ingelezen, moeten we Apache herstarten. Dit doen we met het *service* programma:

```
service apache --full-restart
```

Als het goed is, luistert Apache nu alleen nog op de localhost (127.0.0.1). Dit kunnen we controleren met het *netstat* commando:

```
netstat -na
```

De optie *n* (numeric) laat alleen IP-adressen zien, geen domeinnamen. De optie *a* (all) geeft ook de luisterende processen weer. De output is van een Cygwin Linux systeem en moet er ongeveer als volgt uitzien:

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	10.0.20.22:139	0.0.0.0:0	LISTENING
TCP	127.0.0.1:80	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1028	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1038	127.0.0.1:1039	ESTABLISHED
TCP	127.0.0.1:1039	127.0.0.1:1038	ESTABLISHED
TCP	127.0.0.1:1063	127.0.0.1:1064	ESTABLISHED
TCP	127.0.0.1:1064	127.0.0.1:1063	ESTABLISHED
TCP	127.0.0.1:1065	127.0.0.1:1066	ESTABLISHED
TCP	127.0.0.1:1066	127.0.0.1:1065	ESTABLISHED

Huiswerk opsturen

Je kunt de opdracht nu aftekenen. Dit doen je met een pijpleiding tussen de commando's *netstat* en *email*. Zorg dat het onderwerp van de e-mail tussen aanhalingstekens staat:

```
netstat -na | email -s "Apache back-end configuratie" docent@localhost
```

Als er geen foutmelding verschijnt, is de e-mail correct verstuurd.

Administrator uitloggen

We hebben de Apache web-server laten luisteren op de *localhost*, maar we zijn nog steeds ingelogd als Administrator (*root*).

Geef tot slot <Ctrl>-D om de Administrator uit te loggen.