

Huiswerk Linux: Installatie WordPress

WordPress is, net als Apache een typisch Linux project: geschreven onder copyleft en geschreven door een groep van PHP-ontwikkelaars. WordPress is oorspronkelijk gemaakt voor het bijhouden van een blog, maar tegenwoordig kun je er ook websites mee maken. De opdracht voor deze week is: installeer het WordPress CMS.

Over WordPress

WordPress is een *Content Management System* (CMS). De meest elegante (lees: professionele) manier om WordPress te installeren is in één van de *home* directories. WordPress kan nogal groot worden, met name als bezoekers van een WordPress web-site plaatjes en films gaan opladen. Dan heb je veel schijfruimte nodig. De *home* directories van de Linux gebruikers krijgen altijd de meeste schijfruimte toegewezen.

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**.

Zorg ervoor dat de *LAMP* servers *Apache* en *MySQL* gestart zijn. Dit kun je doen we met ons zelfgeschreven shell-script (zie: *Daemon shell-script*), maar je kunt het ook 'handmatig' doen:

```
service apache start
service mysql start
```

Andere Linux gebruikers

Sudo voor de overige Linux-smaken: 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

Zorg ervoor dat de *LAMP* servers *Apache* en *MySQL* gestart zijn. Dit kun je doen we met ons zelfgeschreven shell-script (zie: *Daemon shell-script*), maar je kunt het ook 'handmatig' doen:

```
service apache2 start
service mysql start
```

Let op: als je de melding *mysqld already running* krijgt, maar de *mysqld* reageert helemaal niet, dan heb je een *unclean shut-down*¹ gehad. Verwijder dan het socket-bestand (*/tmp/mysql.sock*) en probeer het nog een keer.

Website neerladen en uitpakken

Zorg er allereerst voor dat je in je *home*-directory staat. Dit doe je met het commando *cd*:

```
cd /home/daniel
```

Let op: vul in plaats van *daniel* je eigen gebruikersnaam in. Als je niet weet wat deze is, gebruik het commando *whoami*. Het kan zijn dat er nog helemaal geen gebruikers op je systeem zijn. Je moet dan minstens één gebruiker aanmaken (zie: *Problemen oplossen*).

Nu kunnen we de broncode voor WordPress neerladen. Dit doen we met het *wget* (www get document) programma:

```
wget -N http://www.commandlinerevolution.nl/Source/wordpress-3.9.1.tar.gz
```

¹ Een *unclean shutdown* wordt meestal veroorzaakt tijdens een server-crash. In Cygwin wil dit nog wel eens gebeuren als je je computer afsluit. De *mysqld* merkt dit niet en het socket-bestand wordt niet verwijderd.

Het programma *wget* maakt een kopie van het neergeladen bestand met een volgnummer als het bestand al een keer is neergeladen. Dit is niet handig. Via de optie *N* (newer) geven we aan dat *wget* het eventueel bestaande bestand vervangt als het nieuwer is.

De broncode is ingepakt in het *tar* (tape archive, in Linux slang: tarball) formaat. Pak het bestand uit middels het *tar* commando:

```
tar -xvf wordpress-3.9.1.tar.gz
```

Met de optie *x* (extract) geven we aan dat we het bestand willen uitpakken. Middels optie *v* (verbose) geven we aan dat we het uitpak-proces willen zien. Met optie *f* (file) geven we aan dat het om een bestand gaat.

Je ziet nu dat het *tar* programma de bestanden uitpakt naar de map *wordpress*. Wacht tot dit klaar is.

Database configureren

WordPress werkt met gegevens uit een database in de MySQL server. Deze moeten we eerst aanmaken. Dit doen we met het programma *mysqladmin*:

```
mysqladmin create -u root wordpress
```

Via het commando *create* geven we aan dat we een nieuwe database willen aanmaken. Via optie *u* (user) geven we aan dat de database moet worden aangemaakt namens de *root* gebruiker.

WordPress configureren

WordPress draait *out-of-the-box*. Dat wil zeggen: nadat je de *PHP* broncode geïnstalleerd hebt, start deze meteen op. Configuratie kun je achteraf doen. De uitzondering hierop is de connectie met de database. WordPress moet weten hoe de database heet en hoe het moet inloggen op de database. Dit kan worden aangegeven in het WordPress configuratie bestand. Ga allereerst naar de zoëven aangemaakte directory. Dit doe je met het *cd* (change directory) commando:

```
cd wordpress/
```

Vervolgens maken we een kopie van het voorbeeld-bestand voor de configuratie van WordPress. Dit doen we met het commando *cp* (copy):

```
cp wp-config-sample.php wp-config.php
```

Nu kunnen we de database-gegevens invullen. Dit doen we met de *vi* editor:

```
vi wp-config.php
```

Je ziet nu de instellingen van WordPress. Toets *i* (insert) om in de *INSERT* modus te komen, wijzig de vetgedrukte instellingen:

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define('DB_NAME', 'wordpress');  
  
/** MySQL database username */  
define('DB_USER', 'root');  
  
/** MySQL database password */  
define('DB_PASSWORD', '0000');  
  
/** MySQL hostname */  
define('DB_HOST', '127.0.0.1');
```

Let op: in sommige distributies heeft de *root* gebruiker van MySQL geen wachtwoord. Haal dan de vier nullen weg, maar laat de aanhalingstekens staan.

Ga nu uit de *INSERT* modus met de **<Esc>** toets en geef de commando's *w* (write) en *q* (quit):

```
:wq
```

Het bestand wordt nu opgeslagen en we zijn terug op de commando prompt.

DNS configureren

We hebben een domeinnaam nodig voor de nieuwe web-site. Normaal koop je een domeinnaam bij de Stichting Internet Domeinregistratie (SIDN) via een *Internet Service Provider* (ISP). Ook kun je de domeinnaam aanmaken in de DNS-server van het bedrijf of instelling waarvan je de Admin bent. Wij gebruiken de *localhost*². Open het *hosts* bestand in het *vi* programma:

```
vi /etc/hosts
```

Ga naar het einde van het bestand met de (hoofdletter) **G** toets. We zijn nu nog in de *VISUAL* modus. Voeg een regel toe met het commando **o** (open). We zijn nu in *INSERT* modus. Geef de volgende regel in:

```
127.0.0.1    wordpress.localhost
```

Let op: gebruik een *tab*, geen spaties om het ip-adres en de domeinnaam van elkaar te scheiden. Ga nu uit de *INSERT* modus met de **<Esc>** toets en geef de commando's *w* (write) en *q* (quit):

```
:wq
```

Het bestand wordt nu opgeslagen en we zijn terug op de commando prompt.

Apache configureren

Om ervoor te zorgen dat Apache naar de nieuwe domeinnaam “luistert” gaan we een apart bestandje voor de nieuwe website aanmaken.

| Cygwin gebruikers | Andere Linux gebruikers |
|---|---|
| In een Cygwin distributie staan de Apache configuratiebestanden in <i>/etc/httpd</i> . Zorg ervoor dat je in die directory staat: <pre>cd /etc/httpd</pre> | De Apache configuratiebestanden van Debian distributies (<i>Ubuntu</i> , <i>Knoppix</i> , <i>Kali</i> , etc.) staan in <i>/etc/apache2</i> . Ga in de map <i>sites-enabled</i> staan: <pre>cd /etc/apache2/sites-enabled</pre> <p>Op Slackware distributies, zoals <i>Suse Linux</i> worden de configuratiebestanden bijgehouden in de <i>vhosts.d</i> (virtual hosts) directory: <pre>cd /etc/apache2/vhosts.d</pre></p> |

Maak een configuratiebestand voor de nieuwe website aan. Dit doen we met de *vi* editor:

```
vi wordpress.localhost.conf
```

Je ziet een leeg scherm. Toets **i** (insert) om in de *INSERT* modus gaan. Geef nu de volgende regels in:

```
<VirtualHost 127.0.0.1>
    ServerName wordpress.localhost
    DocumentRoot /home/daniel/wordpress
    AddType application/x-httpd-php .php
    DirectoryIndex index.php
</VirtualHost>
```

Let op: vul in plaats van *daniel* je eigen gebruikersnaam in.

² De localhost is van buiten niet bereikbaar. Als je wilt dat de website van buiten bereikbaar is, zul je een *Fully Qualified Domain Name* (FQDN) moeten gebruiken.

Zoals je aan bovenstaande regels kunt zien bestaat een VirtualHost uit een verzameling instellingen die afwijken van de standaard-instellingen van Apache. Normaal geef je hier aan op welke netwerk-interface (IP-adres) Apache moet “luisteren”. Maar wij willen dat Apache op alle netwerkkaarten van onze machine gaat luisteren. Dit geven we aan met een sterretje. Onze website is niet versleuteld, dus we geven bovendien aan dat Apache alleen op poort 80 moet luisteren (*:80).

Met de optie ServerName geven we aan voor welke website de instellingen gelden. Met de optie DocumentRoot geven we aan waar de documenten voor de nieuwe website zich bevinden. Via de optie AddType geven we aan dat documenten met de extensie *.php* moeten worden uitgevoerd als PHP script. Met de optie DirectoryIndex zorgen we ervoor dat de website opstart met het *index.php* script.

Ga nu uit de *INSERT* modus met de <Esc> toets en geef de commando's *w* (write) en *q* (quit):

```
:wq
```

Het bestand wordt nu opgeslagen en we zijn terug op de commando prompt.

Configuratie testen

Om te kijken of Apache goed is geconfigureerd, kunnen we de configuratie testen. Dit doen we met het *apachectl* (Apache Control) programma:

```
apachectl -S
```

Met de optie *S* (settings) kunnen we een lijst met alle geconfigureerde websites tonen. De output moet er ongeveer als volgt uitzien:

```
Processing config directory: /etc/httpd/*.conf
Processing config file: /etc/httpd/wordpress.localhost.conf
[Sat Apr 23 19:04:00 2016] [alert] httpd: Could not determine the server's fully qualified
domain name, using 10.0.20.22 for ServerName
[Sat Apr 23 19:04:00 2016] [warn] NameVirtualHost optiplex:80 has no VirtualHosts
VirtualHost configuration:
127.0.0.1:80 is a NameVirtualHost
    default server localhost (/etc/httpd/conf/access.conf:16)
    port 80 namevhost localhost (/etc/httpd/conf/access.conf:16)
    port 80 namevhost wordpress.localhost (/etc/httpd/wordpress.localhost.conf:1)
```

Het hoera moment

Eerst moeten we Apache herstarten, zodat de gewijzigde instellingen worden ingelezen. Dit doen we met het *service* programma:

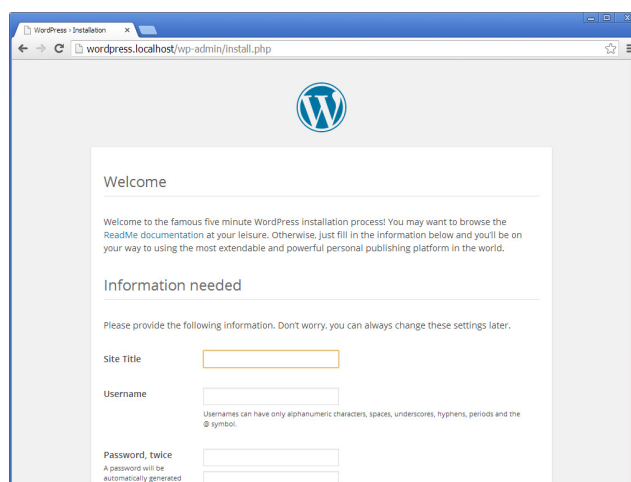
```
service apache restart
```

Let op: in sommige Linux distributies heet de Apache service **apache2**.

Tijd voor het hoera-moment. Open een webbrowser, zoals *Chrome* of *Firefox* en type de volgende URL in de adres-balk:

```
http://wordpress.localhost
```

Je zou nu het volgende scherm (afbeelding rechts) moeten kunnen zien.



Huiswerk opsturen

Je kunt de opdracht aftekenen door een e-mail met daarin de de inhoud van het configuratiebestand te versturen. Dit doen we met een pijpleiding tussen de commando's *cat* en *email*. Zorg dat het onderwerp van de e-mail tussen aanhalingstekens staat:

```
apachectl -S | email -s "Installatie WordPress" docent@localhost
```

Als je geen foutmelding ziet is de e-mail succesvol verstuurd.

Administrator uitloggen

We hebben de PHP broncode voor de *WordPress* website op de juiste locatie geïnstalleerd met de juiste toegangsrechten, maar we zijn nog steeds ingelogd als Administrator (*root*).

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

Problemen oplossen

Dit hoofdstuk kun je overslaan als alles goed werkt. Als het niet werkt, kijk dan altijd eerst in het Apache foutmeldingen-log.

Cygwin gebruikers

De logbestanden op een Cygwin systeem staan in de */var/log/httpd* directory.

```
tail /var/log/httpd/error_log
```

Andere Linux gebruikers

De logbestanden op andere Linux systemen staan in de */var/log/apache2* directory.

```
tail /var/log/apache2/error.log
```

In *Slackware* distributies, zoals *Suse Linux*, is de bestandsnaam *error_log*:

```
tail /var/log/apache2/error_log
```

Gebruiker bestaat niet

Het kan zijn dat je tijdens de installatie geen gebruiker voor jezelf hebt aangemaakt. Je kunt deze in dat geval aanmaken met het *useradd* commando:

```
useradd -m daniel
```

Let op: gebruik hier je eigen naam. Middels de optie *m* (mkdir) zorgen we ervoor dat ook de home-directory voor de nieuwe gebruiker wordt aangemaakt.

Geen rechten in Apache

Het kan zijn dat je web-browser de volgende foutmelding toont:

```
Forbidden
You don't have permission to access /wp-admin/install.php on this server.
```

Apache is vanaf versie 2.4 strenger geworden. Apache weigert PHP scripts uit te voeren, tenzij je uitdrukkelijk aangeeft dat het mag. We kunnen dit instellen in het configuratiebestand van de website. Open het bestand opnieuw met de *vi* editor:

```
vi debian.mshome.conf
```

Voeg de vetgedrukte regels toe:

```
<VirtualHost 127.0.0.1>
  ServerName wordpress.localhost
  DocumentRoot /home/daniel/wordpress
  AddType application/x-httpd-php .php
```

```
DirectoryIndex index.php
</VirtualHost>
<Directory /home/daniel/wordpress>
    Require all granted
</Directory>
```

Herstart Apache, zodat de nieuwe configuratie wordt ingelezen en probeer het opnieuw.

Browser wil pagina neerladen

Als je de naam van je nieuwe website invult, krijg je een download-venstertje in je web-browser. Dit is natuurlijk niet de bedoeling. Je zou de WordPress website moeten krijgen. Dit komt omdat in sommige distributies de PHP module niet geactiveerd is. Het activeren van de PHP-module doen we met het *a2enmod* (Apache2 enable module) commando:

```
a2enmod php7
```

Herstart Apache, zodat de nieuwe configuratie wordt ingelezen en probeer het opnieuw.

Geen rechten in MySQL

Het kan ook zijn dat je web-browser de volgende foutmelding toont:

```
Error establishing a database connection
```

Deze melding is opzettelijk vaag, zodat hackers geen informatie krijgen over de exacte foutmelding. Administrators kunnen in het log van MySQL kijken wat de exacte foutmelding is. Dit doe je met het *tail* (staart) commando.

| Cygwin gebruikers | Andere Linux gebruikers |
|---|---|
| Het MySQL logbestanden op een Cygwin systeem staat in de <i>/var/log</i> directory: | Het logbestand op Debian systemen (<i>Ubuntu, Knoppix, Kali</i> , etc.) heet <i>error.log</i> : |
| tail /var/log/mysqld.log | tail /var/log/mysql/error.log |
| | Het logbestand op andere Linux systemen heet <i>mysqld.log</i> : |
| | tail /var/log/mysql/mysqld.log |

In mijn distributie (Debian) krijg ik de volgende output :

```
2018-01-18 18:30:58 140582545670720 [Note] Plugin 'FEEDBACK' is disabled.
2018-01-18 18:30:58 140581925775104 [Note] InnoDB: Dumping buffer pool(s) not yet started
2018-01-18 18:30:58 140582545670720 [Note] Server socket created on IP: '127.0.0.1'.
2018-01-18 18:30:58 140582545670720 [Note] Reading of all Master_info entries succeeded
2018-01-18 18:30:58 140582545670720 [Note] Added new Master_info '' to hash table
2018-01-18 18:30:58 140582545670720 [Note] /usr/sbin/mysqld: ready for connections.
Version: '10.1.26-MariaDB-0+deb9u1' socket: '/var/run/mysqld/mysqld.sock' port: 3306
Debian 9.1
```

Vreemd. We krijgen niets te zien over een mislukte connectie. Wat we moeten doen is ervoor zorgen dat MySQL meer meldingen geeft³. Dit doen we door het configuratiebestand van MySQL te wijzigen.

```
vi /etc/mysql/my.cnf
```

Ik krijg het volgende te zien. Het bestand ziet er op iedere distributie anders uit, maar de instellingen zijn hetzelfde. Voeg de vetgedrukte regels toe:

³ De reden waarom MySQL in een standaarddistributie geen foutmeldingen geeft, is dat dit nogal veel extra rekenkracht kost. Stel je voor dat er dagelijks 1000 bezoekers op je website komen. Daarvoor moet MySQL iedere keer extra regels in het log schrijven. Hackers maken hier gebruik van om je server te laten crashen. Ze sturen 1000 *page-requests* per seconde. MySQL probeert voor iedere *request* regels in het log te schrijven. MySQL kan dit niet aan en de hele server loopt vast. Dit noemen we een *amplification attack*.

```
# If the same option is defined multiple times, the last one will apply.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# This group is read both both by the client and the server
# use it for options that affect everything
#
[server]
log_warnings=2

[client-server]
```

Eerst moeten we de MySQL daemon herstarten zodat de instellingen opnieuw worden ingelezen. Dit doen we met het volgende commando:

```
service mysql restart
```

Vraag opnieuw de website op door in je browser op de *Refresh* knop te drukken en kijk opnieuw in het log. Nu krijgen we de volgende regel te zien:

```
2018-01-18 18:31:30 140582545024768 [Warning] Access denied for user 'root'@'localhost'
```

Dit betekent dat je het verkeerde wachtwoord gebruikt hebt in het WordPress configuratiebestand (zie: *WordPress configureren*).

Website ziet er vreemd uit

Het kan zijn dat je WordPress website er vreemd uitziet (plaatje rechts). Dit is niet de bedoeling. WordPress is nogal slordig als het gaat om domeinnamen. Je hebt de website eerder onder een andere domeinnaam geactiveerd. Die staat nog in de website, maar is niet meer beschikbaar. Dit kun je oplossen door de WordPress database te verwijderen. Dit doe je met het *drop* commando van het *mysqladmin* programma:

```
mysqladmin drop -u root wordpress
```

Waarschuwing: dit kun je alleen doen als jouw WordPress website nog niet gebruikt is!! Dit commando verwijdert alle gebruikersgegevens uit je website.

Maak nu de WordPress database weer aan. Dit doe je met het *create* commando van het *mysqladmin* programma:

```
mysqladmin create -u root wordpress
```

Toets **<Ctrl+R>** (*Reload*) in je webbrowser. De website is nu weer klaar voor gebruik.

