



## Projectwerk - Documentatie



## Administration

Jorn Jambers, Wout Decré & Jan Collijs  
3 TI  
juni 2009

Katholieke Hogeschool Leuven  
Departement Gezondheidszorg en Technologie  
Herestraat 49 - 3000 LEUVEN  
Tel. +32 16 37 52 00  
Fax +32 16 37 52 99  
gt@khleuven.be

## INHOUDSOPGAVE

<b>Inhoudsopgave</b>	<b>1</b>
<b>Inleiding</b>	<b>3</b>
<b>Virtualisatie</b>	<b>4</b>
Virtualbox . . . . .	4
Qemu . . . . .	4
<b>Kickstart</b>	<b>6</b>
Download . . . . .	6
Configuratie . . . . .	6
<b>Puppet</b>	<b>15</b>
Kickstart installatie en configuratie . . . . .	15
Puppet master op Ubuntu . . . . .	16
Puppet slave op Ubuntu . . . . .	16
Handmatige installatie en configuratie . . . . .	17
Puppet master op FreeBSD . . . . .	17
Puppet master op Ubuntu . . . . .	19
Puppet configuratie . . . . .	21
MySQL databank aanmaken op master . . . . .	22
Beheer . . . . .	25
Het aanmaken van een nieuwe node . . . . .	25
Activeren/deactiveren/resetten van modules . . . . .	25
Modules . . . . .	26

Apache . . . . .	26
Bind . . . . .	27
Samenvatting . . . . .	27
Verwijderen van een node . . . . .	27

<b>Bronnen</b>	<b>28</b>
----------------	-----------

# Inleiding

Deze howto werd opgesteld in het kader van het projectwerk aan de Katholieke Hogeschool Leuven in de richting Toegepaste Informatica.

Aan de hand van dit document zou je in staat moeten zijn om op zelfstandige basis een netwerk van enkele servers op te zetten. Deze opstelling kan dan op een geautomatiseerde manier worden beheerd met behulp van het open source pakket puppet<sup>1</sup>.

Door het schrijven van enkele scripts hebben we getracht om de kracht van dit pakket op een gebruiksvriendelijke manier te benaderen.

Wij wensen u alvast veel leergenoegen toe met het volgen van onze opstelling.

Succes.

---

<sup>1</sup><http://reductivelabs.com/products/puppet/>

# Virtualisatie

Om het testen van de verschillende onderdelen makkelijker te maken hebben we gebruik gemaakt van virtuele machines. Na heel wat geklungel met qemu hebben we besloten om het over een andere boeg te gooien en met virtualbox in zee te gaan.

Meer informatie vind je terug op de website.

## Virtualbox

VirtualBox onder Ubuntu installeren

```
# sudo apt-get install virtualbox-ose
```

Om een virtuele Ubuntu machine te kunnen booten in VirtualBox, moet je PAE/NX enablen onder General - Advanced in de settings van de specifieke ubuntu image onder VirtualBox.

Je kan de netwerkkaart van je virtuele machine binden op je host interface, zodat je in hetzelfde netwerk zit als je host computer.

## Qemu

Wij raden het gebruik van qemu af wegens de vele problemen met betrekking tot virtualisatie van het netwerkgebeuren en de snelheid, gebruiksgemak.

Echter voor diegene die het toch wil proberen hier de opstelling die wij gebruikten om met qemu te kunnen werken.

Om Qemu onder Ubuntu te installeren:

```
# sudo apt-get install qemu
```

Puppet master:

```
# qemu-img create -f qcow2 master.qcow2 10G
# qemu -hda master.qcow2 -boot d -cdrom /dev/scd0 &
# qemu -hda master.qcow2 -net nic , macaddr=52:54:00:12:36:01 -net
  socket , listen=:1234 &
```

Puppet slave:

```
# qemu-img create -f qcow2 slave.qcow2 10G
# qemu -hda slave.qcow2 -boot d -cdrom /dev/scd0 &
# qemu -hda slave.qcow2 -net nic , macaddr=52:54:00:12:36:02 -net
  socket , connect=:1234 &
```

Op die manier worden er twee virtuele systemen aangemaakt en opgestart welke met elkaar in verbinding komen te staan. Later in deze howto wordt duidelijk hoe deze dienen te worden geconfigureerd.

# Kickstart

In dit hoofdstuk leggen we uit hoe je zelf een kickstart cdrom kan aanmaken aan de hand van onze configuratie om het installeren van een server binnen de opstelling nog sneller te laten verlopen.

## Download

Je hebt geen aparte software nodig om Kickstart te gebruiken, kickstart werd reeds standaard meegeleverd met Ubuntu. Je moet dus enkel Ubuntu's server editie downloaden.

Om het ISO bestand te wijzigen, gebruiken we ISO Master.

```
# sudo apt-get install isomaster
```

Een handige tool om een eerste keer een Kickstart configuratie te maken is Kickstart Configurator.

```
# sudo apt-get install system-config-kickstart
```

In deze howto gebruiken we de bestanden ksmaster.cfg en ksslave.cfg, deze zijn gemaakt m.b.v. Kickstart Configurator.

## Configuratie

In dit deel leggen we uit hoe je de eigenlijke iso image aanmaakt voor het installeren van een ubuntu server waarbij enkel de benodigde gegevens voor de puppet configuratie dienen ingegeven te worden

Met ISO Master open je de ubuntu.iso image.

**NOTA** *In ISO Master, ga naar Tools - Options, en vul bij Editor gedit of een tekstverwerker naar keuze in.*

Vervolgens ga je naar de map isolinux (van de ubuntu iso), en kiest om het bestand isolinux.cfg te wijzigen (F4)

Vervang:

```
LABEL install
  menu label ^Install Ubuntu Server (+ Puppet)
  kernel /install/vmlinuz
  append ks=cdrom:/ks.cfg fb=false file=/cdrom/preseed/ubuntu-
server.seed initrd=/install/initrd.gz quiet --
```

Door onderstaande labels (best te kopiëren vanop de wiki:

```
LABEL installslave
  menu label ^Install Ubuntu Server as Puppet slave
  kernel /install/vmlinuz
  append ks=cdrom:/ksslave.cfg fb=false file=/cdrom/preseed/
ubuntu-server.seed initrd=/install/initrd.gz quiet --
LABEL installmaster
  menu label ^Install Ubuntu Server as Puppet master
  kernel /install/vmlinuz
  append ks=cdrom:/ksmaster.cfg fb=false file=/cdrom/preseed/
ubuntu-server.seed initrd=/install/initrd.gz quiet --
```

Met behulp van dit bestand kunnen we het root menu van de boot cdrom aanpassen. Zo geven we een label mee voor de puppetmaster en een label voor de puppetslave. Op die manier kan er met behulp van 1 cdrom gekozen worden voor de installatie van een puppetmaster of van een puppetslave.

Kies Go back, zodat je terug in de root (/) map bent.

**NOTA** *Volgende bestanden kopieër je best vanop de wiki* Met je favoriete editor maak je een nieuw bestand en noemt het ksmaster.cfg:

```
#Generated by Kickstart Configurator
#platform=x86
#System language
lang en_US
#Language modules to install
langsupport en_US
#System keyboard
keyboard be
#System mouse
mouse
#System timezone
timezone Europe/Brussels
#Root password (= $$32Puppet)
rootpw --iscrypted $1$HUUqY2cv$EDqnPg0QxtPBIM9d22ynC/
#Initial user
```



```

user --disabled
#Reboot after installation
reboot
#Use text mode install
text
#Install OS instead of upgrade
install
#Use CDROM installation media
cdrom
#System bootloader configuration
bootloader --location=mbr
#Clear the Master Boot Record
zerombr yes
#Partition clearing information
clearpart --all --initlabel
#Disk partitioning information
part / --fstype ext3 --size 1 --grow
part swap --size 1024
#System authorization information
auth --useshadow --enablemd5
#Network information
network --bootproto=dhcp --device=eth0
#Firewall configuration
firewall --disabled
#Do not configure the X Window System
skipx
%packages
openssh-server
dialog
mysql-server
python-mysqldb
puppet
puppetmaster
%post --interpreter=/bin/sh
#Capture input from tty6, and send output to tty6
chvt 6
exec < /dev/tty6 > /dev/tty6

clear

/media/cdrom/puppet/master.sh

#Capture input from tty1, and send output to tty1
chvt 1
exec < /dev/tty1 > /dev/tty1

```

Maak ook het bestand ksslave.cfg aan:

```

#Generated by Kickstart Configurator
#platform=x86
#System language
lang en_US
#Language modules to install
langsupport en_US

```

```

#System keyboard
keyboard be
#System mouse
mouse
#System timezone
timezone Europe/Brussels
#Root password (= $$32Puppet)
rootpw --iscrypted $1$HUUqY2cv$EDqnPg0QxtPBIM9d22ynC/
#Initial user
user --disabled
#Reboot after installation
reboot
#Use text mode install
text
#Install OS instead of upgrade
install
#Use CDROM installation media
cdrom
#System bootloader configuration
bootloader --location=mbr
#Clear the Master Boot Record
zerombr yes
#Partition clearing information
clearpart --all --initlabel
#Disk partitioning information
part / --fstype ext3 --size 1 --grow
part swap --size 1024
#System authorization information
auth --useshadow --enablemd5
#Network information
network --bootproto=dhcp --device=eth0
#Firewall configuration
firewall --disabled
#Do not configure the X Window System
skipx
%packages
dialog
puppet
%post --interpreter=/bin/sh
#Capture input from tty6, and send output to tty6
chvt 6
exec < /dev/tty6 > /dev/tty6

clear

/media/cdrom/puppet/slave.sh

#Capture input from tty1, and send output to tty1
chvt 1
exec < /dev/tty1 > /dev/tty1

```

Plaats deze bestanden (ksmaster.cfg en ksslave.cfg) onder de root map (/) m.b.v. ISO Master.

Maak het bestand master.sh aan (dit is het bestand waarmee de configuratie wordt weggeschreven om een puppet master op te zetten):

```
#!/bin/sh
#####
# Puppet master installation #
#####

backtitle=" Puppet_master_configuration"

# Temporarily file to read answers from dialog
answer='mktemp'

dialog --sleep 1 --backtitle "$backtitle" --infobox "Starting Puppet_configuration" 3 33

# Puppet master network settings
dialog --backtitle "$backtitle" --nocancel --inputbox "IP address_of_this_Puppet_master:" 8 37 2> $answer
ipmaster='cat ${answer}'
dialog --backtitle "$backtitle" --nocancel --inputbox "Netmask_of_this_Puppet_master:" 8 34 "255.255.255.0" 2> $answer
nmmaster='cat ${answer}'
dialog --backtitle "$backtitle" --nocancel --inputbox "Gateway_of_this_Puppet_master:" 8 34 2> $answer
gwmaster='cat ${answer}'
echo -e "auto_lo\ niface_lo\ inet_loopback\n\ nauto_eth0\ niface_eth0\ inet_static\n\ taddress_${ipmaster}\n\ tnetmask_${nmmaster}\n\ tgateway_${gwmaster}" > /etc/network/interfaces
dialog --backtitle "$backtitle" --nocancel --inputbox "Nameserver(s)_of_this_Puppet_master_(seperated_by_):" 8 60 2> $answer
nameservers='cat $answer | tr --delete ' ' | tr ', ' "\n"'
echo -n "" > /etc/resolv.conf
for ns in $nameservers; do
    echo "nameserver_$ns" >> /etc/resolv.conf
done
dialog --backtitle "$backtitle" --nocancel --inputbox "FQDN_of_this_Puppet_master:" 8 40 2> $answer
fqdnmaster='cat ${answer}'
hostname $fqdnmaster
echo $fqdnmaster > /etc/hostname
hostmaster='echo ${fqdnmaster} | cut -d '.' -f 1'
echo -e "\n127.0.0.1\t${fqdnmaster}\_${hostmaster}.puppet" >> /etc/hosts

# Copy Puppet configuration
dialog --sleep 1 --backtitle "$backtitle" --infobox "Copying Puppet_configuration" 3 32
cp -R /media/cdrom/puppet/manifests /etc/puppet/
cp -R /media/cdrom/puppet/modules /etc/puppet/
cp -R /media/cdrom/puppet/templates /etc/puppet/
chmod 700 /etc/puppet
chown puppet:puppet /etc/puppet
```

```

# Set up Puppet configuration
dialog --sleep 1 --backtitle "$backtitle" --infobox "Setting up
Puppet configuration" 3 35
nsline=""
for ns in $nameservers; do
    nsline="${nsline}"${ns}" "
done
nsline='echo $nsline | sed 's/,,$/'
cat /etc/puppet/manifests/templates.pp | sed "s/^.*\$nameservers
==\[\]\$/\t\$nameservers==[${nsline}]/" > /etc/puppet/
manifests/templates.pp
cat /etc/puppet/modules/bind/manifests/init.pp | sed "s/^.*\$
$forwarders==\[\]\$/\t\$forwarders==[${nsline}]/" > /etc/
puppet/modules/bind/manifests/init.pp

# Set up MySQL
dialog --sleep 1 --backtitle "$backtitle" --infobox "Setting up
MySQL" 3 20
/etc/init.d/mysql start
dialog --backtitle "$backtitle" --nocancel --inputbox "Database
name:" 8 37 "puppet" 2> $answer
dbname='cat ${answer}'
/usr/bin/mysqladmin create $dbname
/usr/bin/mysql $dbname < /media/cdrom/puppet/puppet.sql
dialog --backtitle "$backtitle" --nocancel --inputbox "Database
username:" 8 37 "puppet" 2> $answer
dbuser='cat ${answer}'
dialog --backtitle "$backtitle" --nocancel --insecure --
passwordbox "Database password:" 8 37 "puppet" 2> $answer
dbpass='cat ${answer}'
echo "CREATE USER '${dbuser}'@'localhost' IDENTIFIED BY '${
dbpass}';" | /usr/bin/mysql
echo "GRANT SELECT, INSERT, UPDATE, DELETE ON ${dbname}.* TO '${
dbuser}'@'localhost';" | /usr/bin/mysql
/usr/bin/mysqladmin flush-privileges
dialog --backtitle "$backtitle" --nocancel --insecure --
passwordbox "MySQL root password:" 8 30 "\$32Puppet" 2>
$answer
mysqlpass='cat ${answer}'
/usr/bin/mysqladmin -u root password "$mysqlpass"

# Copy scripts
dialog --sleep 1 --backtitle "$backtitle" --infobox "Copying
scripts" 3 19
cp /media/cdrom/puppet/mysql-to-puppet.py /media/cdrom/puppet/
mysql-to-nagios.py /media/cdrom/puppet/puppetman.py /usr/
local/bin/
chmod 700 /usr/local/bin/mysql-to-puppet.py /usr/local/bin/mysql
-to-nagios.py /usr/local/bin/puppetman.py
cat /usr/local/bin/mysql-to-puppet.py | sed "s/^mysql_database_
=.*$/mysql_database='${dbname}']/" > /usr/local/bin/mysql-to
-puppet.py
cat /usr/local/bin/mysql-to-puppet.py | sed "s/^mysql_username_

```

```

    =.*$/mysql_username_\=${dbuser}'/" > /usr/local/bin/mysql-to-
    -puppet.py
cat /usr/local/bin/mysql-to-puppet.py | sed "s/^mysql_password_\
    =.*$/mysql_password_\=${dbpass}'/" > /usr/local/bin/mysql-to-
    -puppet.py
cat /usr/local/bin/mysql-to-nagios.py | sed "s/^mysql_database_\
    =.*$/mysql_database_\=${dbname}'/" > /usr/local/bin/mysql-to-
    -nagios.py
cat /usr/local/bin/mysql-to-nagios.py | sed "s/^mysql_username_\
    =.*$/mysql_username_\=${dbuser}'/" > /usr/local/bin/mysql-to-
    -nagios.py
cat /usr/local/bin/mysql-to-nagios.py | sed "s/^mysql_password_\
    =.*$/mysql_password_\=${dbpass}'/" > /usr/local/bin/mysql-to-
    -nagios.py
cat /usr/local/bin/puppetman.py | sed "s/^mysql_database_\=${dbname}'/" > /usr/local/bin/puppetman.py
cat /usr/local/bin/puppetman.py | sed "s/^mysql_username_\=${dbuser}'/" > /usr/local/bin/puppetman.py
cat /usr/local/bin/puppetman.py | sed "s/^mysql_password_\=${dbpass}'/" > /usr/local/bin/puppetman.py

dialog --backtitle "$backtitle" --nocancel --pause "Puppet_
    master_configuration_completed" 8 41 5

```

En dan nog het bestand slave.sh (dit is het bestand waarmee de configuratie wordt weggeschreven om een puppet master op te zetten):

```

#!/bin/sh
#####
# Puppet slave installation #
#####

backtitle="Puppet_slave_configuration"

# Temporarily file to read answers from dialog
answer='mktemp'

dialog --sleep 1 --backtitle "$backtitle" --infobox "Starting_
    Puppet_configuration" 3 33

# Puppet slave network settings
dialog --backtitle "$backtitle" --nocancel --inputbox "IP_
    address_of_this_Puppet_slave:" 8 36 2> $answer
ipslave='cat ${answer}'
dialog --backtitle "$backtitle" --nocancel --inputbox "Netmask_
    of_this_Puppet_slave:" 8 33 "255.255.255.0" 2> $answer
nmslave='cat ${answer}'
dialog --backtitle "$backtitle" --nocancel --inputbox "Gateway_
    of_this_Puppet_slave:" 8 33 2> $answer
gwslave='cat ${answer}'
echo -e "auto_lo\niface_lo inet loopback\n\nauto_eth0\niface_
    eth0 inet static\n\taddress_${ipslave}\n\tnetmask_${nmslave}\n
    \tgateway_${gwslave}" > /etc/network/interfaces

```

```

dialog --backtitle "$backtitle" --nocancel --inputbox "
    Nameserver(s) of this Puppet master (seperated by ,):" 8 60
2> $answer
nameservers='cat $answer | tr --delete ' ' | tr ',,' "\n"'
echo -n "" > /etc/resolv.conf
for ns in $nameservers; do
    echo "nameserver $ns" >> /etc/resolv.conf
done
dialog --backtitle "$backtitle" --nocancel --inputbox "FQDN of
    this Puppet slave:\nThis may NOT BE 'puppet.something'" 9 40
2> $answer
fqdnslave='cat ${answer}'
hostname $fqdnslave
echo $fqdnslave > /etc/hostname
hostslave='echo ${fqdnslave} | cut -d '.' -f 1'
echo -e "\n127.0.0.1\t${fqdnslave}\t${hostslave}" >> /etc/hosts

# Puppet slave master information
dialog --backtitle "$backtitle" --nocancel --inputbox "IP
    address of Puppet master:" 8 32 2> $answer
ipmaster='cat ${answer}'
dialog --backtitle "$backtitle" --nocancel --inputbox "FQDN of
    Puppet master:" 8 30 2> $answer
fqdnmaster='cat ${answer}'
hostmaster='echo ${fqdnmaster} | cut -d '.' -f 1'
echo -e "${ipmaster}\t${fqdnmaster}\t${hostmaster}\tpuppet" >> /
    etc/hosts
echo -e "\n[puppetd]" >> /etc/puppet/puppet.conf
echo "server=${fqdnmaster}" >> /etc/puppet/puppet.conf
chmod 700 /etc/puppet

# Wait for signed certificate
dialog --sleep 1 --backtitle "$backtitle" --infobox "Starting
    Puppet daemon" 3 26
puppetd &
dialog --backtitle "$backtitle" --msgbox "~~~~~Press<OK>~
    when the Puppet administrator signed the certificate." 6 41

dialog --backtitle "$backtitle" --nocancel --pause "Puppet slave
    configuration complete" 8 40 5

```

Maak deze bestanden (master.sh en slave.sh) uitvoerbaar:

```
# chmod 755 master.sh slave.sh
```

Maak een map puppet aan onder de root dir (/) m.b.v. ISO Master.

Plaats de 2 scripts onder de juist aangemaakte puppet map m.b.v. ISO Master.

Onder dezelfde map plaats je ook de 3 Python scripts (mysql-to-puppet.py, mysql-to-nagios.py en puppetman.py). Je vind ze op Puppet#Scripts.

Nog steeds onder dezelfde map plaats je ook de (blanco) Puppet configuratiebestanden. Je vindt ze op wiki.

Verder dien je ook nog een puppet.sql bestand te creëren voor het aanmaken van de databank in deze map puppet plaatsen mbv ISO master. De inhoud voor dit bestand is tevens terug te vinden op de wiki

Optioneel kan je de splash image wijzigen. Download PuppetSplash.pcx, en plaats het onder isolinux nadat je de naam wijzigde naar splash.pcx.

**NOTA** *Om een bestand te overschrijven met ISO Master, dien je het eerst te Removen, en dan te Adden.*

Save As ks.iso

# Puppet

Bij aanvang van dit hoofdstuk zou je in het bezit moeten zijn van een werkende puppet iso image. Deze kan je bekomen door het vorige hoofdstuk te volgen.

Ofwel ben je in het bezit van een werkende kickstart puppet cdrom welke via de erkende kanalen te verkrijgen is.

## Kickstart installatie en configuratie

Bij het gebruik van fysieke machines dien je te booten vanop cdrom. Bij gebruik van virtuele machines booten vanop de iso image als vanzelfsprekend.

Vooraleer je een machine installeert dien je wel enkele parameters voor jezelf op te maken als zijnde

- IP adres van de machine (zowel voor slave als master)
- Netmasker
- Gateway
- Nameserver
- FQDN (master.domein.be)
- database naam, gebruikersnaam & wachtwoord, root wachtwoord mysql

**NOTA** *Het is niet onverstandig deze gegevens op voorhand neer te pennen aangezien deze tijdens beide installaties van pas kunnen komen.*



## Puppet master op Ubuntu

Boot met de Kickstart CD (ks.iso) en volg de instructies voor de master installatie.

## Puppet slave op Ubuntu

**NOTA** *Het is belangrijk dat u de puppet master reeds geïnstalleerd hebt en op hebt staan alvorens de slave te installeren.*

Boot met de Kickstart CD (ks.iso) en volg de instructies voor de slave installatie.

De eerste keer dat een Puppet slave node connecteert naar de Puppet master stopt de installatie even, de nieuwe node is nog niet gekend.

Er dient nog een certificaat te worden getekend. Om een lijst van nog te tekenen nodes te krijgen gebruik je puppetca op de puppetmaster:

```
# puppetca --list
```

Voorbeeld output:

```
web.puppets.be
```

Om deze dan te ondertekenen gebruik je ook puppetca:

```
# puppetca --sign web.puppets.be
```

Voorbeeld output:

```
Signed web.puppets.be
```

Wanneer dit niet onmiddellijk is gelukt kan u gerust verder gaan met de installatie en deze authenticatie achteraf uitvoeren op volgende manier.

Puppet slave:

```
# puppetd --server puppetmaster.domein.be --verbose --  
waitforcert 60
```

Puppet master:

```
# puppetca --list  
# puppetca --sign web.puppets.be
```

Wanneer je bevestiging krijgt dat het certificaat werd ondertekent is de authenticatie geslaagd en kan je verder gaan met de howto.

## Handmatige installatie en configuratie

**NOTA** *Wanneer je gebruik maakt van de kickstart cdrom mag je dit deel overslaan aangezien dit mbv de installatie cdrom reeds uitgevoerd werd.*

Met behulp van de kickstart cdrom kan je kiezen om van een server een puppet master machine of puppet slave machine te maken binnen de opstelling. Vervolgens worden de nodige stappen genomen om van deze machine het juiste puppet systeem te maken.

Wij opteerden voor het gebruik van kickstart om op een zo efficiënt mogelijke manier een server zo snel en performant mogelijk operationeel te krijgen.

In volgend deel wordt uitgelegd hoe u handmatig vertrekkende vanuit een standaard linux distributie van een server een puppet master of slave maakt.

### Puppet master op FreeBSD

**NOTA** *Draait op FreeBSD 7.2-RELEASE, de volledige installatie en configuratie van FreeBSD is out-of-the-scope van deze howto.*

We hebben er voor gekozen om de configuratie van de master zelf niet via Puppet te doen. Zou er iets fouts met Puppet gebeuren, dan hebben we een grotere kans dat we het via de master kunnen herstellen.

Puppet master:

```
# cd /usr/ports/sysutils/puppet
# make install clean
```

MySQL installatie:

```
# cd /usr/ports/databases/mysql51-server
# make install clean
```

In /etc/rc.conf:

```
puppetmasterd_enable="YES"
mysql_enable="YES"
```

### Nagios op FreeBSD

Binnen onze opstelling hebben wij er ook voor gezorgd dat het mogelijk is op een zeer gebruiksvriendelijke manier een nagios monitoring server op te nemen.

Hoe je dergelijke installatie manueel kan uitvoeren op een FreeBSD systeem vind je hier terug.

**NOTA** *Dit is niet verplicht voor een Puppet master, je kan van een Puppet slave (node) een Nagios server maken.*

Apache installatie:

```
# cd /usr/ports/www/apache22
# make install clean
```

Nagios installatie:

```
# cd /usr/ports/net-mgmt/nagios
# make install clean
```

In /etc/rc.conf:

```
apache22_enable="YES"
nagios_enable="YES"
```

Aanmaken van /usr/local/etc/apache22/Includes/nagios.conf:

```
<Directory /usr/local/www/nagios>
    Order deny,allow
    Allow from all

    AuthName "Nagios Access"
    AuthType Basic
    AuthUserFile /usr/local/etc/nagios/htpasswd.users
    require valid-user
</Directory>

<Directory /usr/local/www/nagios/cgi-bin>
    Options ExecCGI
</Directory>

ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/
Alias /nagios/ /usr/local/www/nagios/
```

Aanmaken Apache login bestand voor de Nagios administrator:

```
# htpasswd -c /usr/local/etc/nagios/htpasswd.users nagiosadmin
```

Kopie maken van default configs:

```
# cd /usr/local/etc/nagios
# cp cgi.cfg-sample cgi.cfg
# cp nagios.cfg-sample nagios.cfg
# cp resource.cfg-sample resource.cfg
# cd objects
# cp commands.cfg-sample commands.cfg
```

```
# cp contacts.cfg-sample contacts.cfg
# cp localhost.cfg-sample localhost.cfg
# cp templates.cfg-sample templates.cfg
# cp timeperiods.cfg-sample timeperiods.cfg
```

Start services:

```
# /usr/local/etc/rc.d/apache22 start
# /usr/local/etc/rc.d/nagios start
```

U kan nu surfen naar <http://hostname/nagios/> (vergeet de '/' op het einde niet!!)

## Puppet master op Ubuntu

**NOTA** *Draait op Ubuntu Server 8.04.2 LTS, de volledige installatie en configuratie van Ubuntu is out-of-the-scope van deze howto.*

We hebben er voor gekozen om de configuratie van de master zelf niet via Puppet te doen. Zou er iets fouts met Puppet gebeuren, dan hebben we een grotere kans dat we het via de master kunnen herstellen.

OpenSSH:

```
# sudo apt-get install openssh-server
```

Puppet master:

```
# sudo apt-get install puppet
```

MySQL installatie:

```
# sudo apt-get install mysql-server
```

## Nagios op Ubuntu

Binnen onze opstelling hebben wij er ook voor gezorgd dat het mogelijk is op een zeer gebruiksvriendelijke manier een nagios monitoring server op te nemen. Hoe je dergelijke installatie manueel kan uitvoeren op een Ubuntu systeem vind je hier terug.

Apache installatie:

```
# sudo apt-get install apache2
```

Nagios installatie:

```
# sudo apt-get install nagios2 nagios-plugins nagios-images
```

Aanmaken Apache login bestand voor de Nagios administrator:

```
# cd /etc/nagios2  
# sudo htpasswd -c htpasswd.users nagiosadmin
```

U kan nu surfen naar `http://hostname/nagios2`

## Puppet configuratie

Puppet maakt gebruik van manifests voor de configuratie van de verschillende slave systemen. Wij genereerden enkele van deze manifests volgens onze opstelling. Om hiervan gebruik te maken gaat u als volgt te werk:

Onder FreeBSD:

```
# mkdir -p /usr/local/etc/puppet/manifests
# mkdir /usr/local/etc/puppet/templates
# mkdir /usr/local/etc/puppet/modules
```

Onder Ubuntu:

```
# mkdir -p /etc/puppet/manifests
# mkdir /etc/puppet/templates
# mkdir /etc/puppet/modules
```

Download de bestanden hier: [PuppetConfig.tar.gz](#)

## Manifests

hebben altijd de .pp extensie. Het eerste bestand dat wordt ingelezen is site.pp. Dit bestand laadt alle andere in. In het mysql\_nodes.pp bestand komen de gegenereerde nodes uit de (MySQL) databank, je moet dus dat bestand niet manueel aanpassen! Custom nodes kun je in het nodes.pp bestand zetten.

**NOTA** *Hier komen eerder globale bestanden van het besturingssysteem.*

Aan te passen:

- manifests/templates.pp: nameserver(s) aanpassen - vb: \$nameservers = ["192.168.1.1", "192.168.1.2"]
- modules/bind/manifests/init.pp: nameserver(s) aanpassen - vb: \$forwarders = ["192.168.1.1"]
- users/virt\_users.pp + groups/itteam.pp: gebruiker(s) aanpassen

De standaard wachtwoorden zijn \$\$32Puppet.

## Templates

Templates zijn dynamische configuratiebestanden (= gegenereerd door Puppet configuraties). Ze hebben altijd de .erb extensie.

**NOTA** *Hier komen eerder globale bestanden van het besturingssysteem.*

## Modules

Modules zijn manifests die samenhangen (vb. Apache module, Bind module, ...). Elke module beschikt over een EIGEN files, manifests, en templates map.

De files map is handig om statische configuraties door te sturen, de manifests voor Puppet configuraties, en de templates om dynamische configuraties (= gegenereerd door Puppet configuraties) door te sturen.

## MySQL databank aanmaken op master

**NOTA** *Wanneer je gebruik maakt van de kickstart cdrom mag je dit deel overslaan aangezien dit mbv de installatie cdrom reeds uitgevoerd werd.*

Om alle gegevens van de verschillende nodes binnen het netwerk op een centrale plaats te bewaren, maken we gebruik van een MySQL databank. Deze kan m.b.v. scripts worden aangesproken om de data uit te lezen, aan te passen, of te creëren.

De volgende SQL code kan je inladen met behulp van het "'mysql'" commando, of met een webtool zoals "'phpMyAdmin'".

```
# mysqladmin -p create puppet
# mysql -p puppet < puppet.sql
```

Vergeet ook geen MySQL gebruiker aan te maken:

```
# mysql -p
mysql> CREATE USER 'puppet'@'localhost' IDENTIFIED BY '
puppet';
mysql> GRANT SELECT,INSERT,UPDATE ON puppet.* TO 'puppet
'@'localhost' WITH GRANT OPTION;
mysql> quit
# mysqladmin -p flush-privileges
```

**NOTA** *Indien je een andere databank, gebruikersnaam of wachtwoord zou willen gebruiken, vergeet deze dan ook niet aan te passen in de scripts!*

```
CREATE TABLE IF NOT EXISTS 'apache_virtualhosts' (
  'apache_vhid' int(11) NOT NULL AUTO_INCREMENT,
  'nodeid' int(11) NOT NULL,
  'status' tinyint(1) NOT NULL,
  'hostname' varchar(80) NOT NULL,
  PRIMARY KEY ('apache_vhid'),
  UNIQUE KEY 'nodeid' ('nodeid','hostname')
);
```

```
CREATE TABLE IF NOT EXISTS 'bind_domains' (
  'bind_domid' int(11) NOT NULL AUTO_INCREMENT,
  'nodeid' int(11) NOT NULL,
  'status' tinyint(1) NOT NULL,
```

```

    'domain' varchar(80) NOT NULL,
    PRIMARY KEY ('bind_domid'),
    UNIQUE KEY 'nodeid' ('nodeid', 'domain')
);

CREATE TABLE IF NOT EXISTS 'bind_subdomains' (
    'bind_subdomid' int(11) NOT NULL AUTO_INCREMENT,
    'bind_domid' int(11) NOT NULL,
    'nodeid' int(11) NOT NULL,
    'status' tinyint(1) NOT NULL,
    'hostname' varchar(80) NOT NULL,
    'ip' varchar(15) NOT NULL,
    PRIMARY KEY ('bind_subdomid'),
    UNIQUE KEY 'bind_domid' ('bind_domid', 'nodeid', 'hostname')
);

CREATE TABLE IF NOT EXISTS 'nodes' (
    'nodeid' int(11) NOT NULL AUTO_INCREMENT,
    'hostname' varchar(80) NOT NULL,
    'ip' varchar(15) NOT NULL,
    'netmask' varchar(15) NOT NULL,
    'gateway' varchar(15) NOT NULL,
    'module_apache' tinyint(1) DEFAULT '0',
    'module_nagios' tinyint(1) DEFAULT '0',
    'module_php' tinyint(1) DEFAULT '0',
    'module_mysql' tinyint(1) DEFAULT '0',
    'module_bind' tinyint(1) DEFAULT '0',
    PRIMARY KEY ('nodeid'),
    UNIQUE KEY 'hostname' ('hostname')
);

```

## Scripts

**NOTA** Wanneer je gebruik maakt van de kickstart cdrom mag je dit deel overslaan aangezien dit mbv de installatie cdrom reeds uitgevoerd werd.

Om de scripts uit te voeren zonder het volledige path te moeten opgeven kun je ze onder de map `"/usr/local/bin"` zetten. Je zou ook de `$PATH` (shell) variable kunnen aanpassen.

### MySQL to Puppet

Dit script kan mogelijk in cron worden gezet. Aangezien het wordt uitgevoerd na het management script, is het "niet noodzakelijk".

```

# crontab -e

# Om de 15 minuten
*/15 * * * * /usr/local/bin/mysql\_to\_puppet.py

```



Download: [PuppetMySQLToPuppet](#)

### **MySQL to Nagios**

Dit script kan mogelijk in cron worden gezet. Aangezien het wordt uitgevoerd na het management script, is het "'niet noodzakelijk" '.

```
# crontab -e  
  
# Om de 15 minuten  
/15 */* */usr/local/bin/mysql_to_nagios.py
```

Download: [PuppetMySQLToNagios](#)

### **Management script**

Met behulp van dit script kunnen er verschillende nodes worden gecreëerd, aangepast en verwijderd. Je kan nodes functies (vb. webserver) toekennen en afnemen.

Download: [PuppetManagement](#)

## Beheer

Voor je begint met het volgen van dit deel dient je opstelling reeds volledig te zijn opgezet, zijnde de puppet master, de puppet slave, de databank en de 2 scriptjes mysql to nagios - puppet. Het beheerscript maakt gebruik van deze opstelling.

Ook het aanpassen van de credentials van de sql connectie in het script is aangewezen om niet voor onaangename verrassingen komen te staan.

### Het aanmaken van een nieuwe node

**NOTA** *Wanneer je gebruik maakt van de kickstart cdrom mag je dit deel overslaan aangezien dit mbv de installatie cdrom reeds uitgevoerd werd.*

Wanneer er een nieuwe node in het systeem dient te worden opgenomen, maak je gebruik van de parameter -C (-create). Deze parameter vereist dat je de node parameter ook meegeeft met -n (-node), evenals het IP adres van de nieuwe node, de netmask, en het IP adres van de Puppet master.

```
# puppetman.py -C -n <FQDN> -I <IP> -N <netmask> -G <gateway>
```

Als dit succesvol werd uitgevoerd werd deze nieuwe node aan de centrale databank toegevoegd.

### Activeren/deactiveren/resetten van modules

Standaard wordt een node aangemaakt zonder de activatie van een module. De modules zijn: apache, nagios, php, mysql, en bind. Er kunnen andere modules worden toegevoegd, maar dan moeten de scripts en de databank worden aangepast.

Elke module kan in 3 toestanden worden geplaatst: enabled (1), disabled (-1) of cleared (0).

**NOTA** *Je kan ook "\*" gebruiken om een status te wijzigen van alle modules. Vergeet zeker niet de aanhalingstekens mee te geven!*

#### Enabled (0)

Wanneer een module op enabled wordt ingesteld, dan wordt deze service actief (geinstalleerd + gecontroleerd op actief) op de betreffende node.

```
# puppetman.py -n <FQDN> -e <[*|apache|nagios|php|mysql|bind]>
```

## Disabled (-1)

Wanneer een module op disabled wordt ingesteld, dan wordt deze service inactief (niet genstalleerd + gecontroleerd op inactief) op de betreffende node. De configuratie wordt ook verwijderd.

Bij de Apache module blijven websites (onder /var/www) behouden na het disablen.

```
# puppetman.py -n <FQDN> -d <*|apache|nagios|php|mysql|bind>
```

## Cleared (0)

Wanneer een module op cleared wordt ingesteld, dan wordt deze service genegeerd. Dit wil zeggen:

- Indien hij enabled was, blijft hij actief maar wordt er niet meer op gecontroleerd.
- Indien hij disabled was, blijft hij inactief maar wordt er niet meer op gecontroleerd.

De reden voor deze status is: indien je een module eerst hebt enabled, maar daarna wil je ze toch disablen, dan kun je ze even disablen. Tot de Puppet slave node connecteert en de module (service) verwijderd. Hierna moet hij niet elke keer controleren of de service nog steeds disabled is, maar kun je ze op de cleared status zetten. Elke module staat standaard op de cleared status.

```
# . puppetman.py -n <FQDN> -c <*|apache|nagios|php|mysql|bind>
```

## Modules

### Apache

Wanneer voor een node de module apache werd geactiveerd, kan je voor deze node makkelijk Virtuele Hosts beheren. Je kan er aanmaken (addvh), verwijderen (deletevh), enablen (enablevh) en disablen (disablevh).

Enkele voorbeelden:

```
# puppetman.py -n web.puppets.be -M apache addvh www.puppets.be
# puppetman.py -n web.puppets.be -M apache disablevh www.puppets
  .be
# puppetman.py -n web.puppets.be -M apache enablevh www.puppets.
  be
# puppetman.py -n web.puppets.be -M apache deletevh www.puppets.
  be
```

## Bind

Wanneer voor een node de module bind werd geactiveerd, kan je voor deze node makkelijk (sub)domeinen beheren. Je kan er aanmaken (adddomain of addsubdomain), verwijderen (deletedomain of deletesubdomain), enablen (enabledomain of enablesubdomain) en disablen (disabledomain of disablesubdomain).

Enkele voorbeelden:

```
# puppetman.py -n web.puppets.be -M bind adddomain puppets.be
# puppetman.py -n web.puppets.be -M bind addsubdomain puppets.be
  www 192.168.1.35
# puppetman.py -n web.puppets.be -M bind disablesubdomain
  puppets.be www
# puppetman.py -n web.puppets.be -M bind addsubdomain puppets.be
  foo 192.168.1.36
# puppetman.py -n web.puppets.be -M bind deletedomain puppets.be
```

## Samenvatting

Een samenvatting van de verschillende nodes in de databank krijg je als je geen opties meegeeft:

```
# puppetman.py
```

Om meer gedetailleerde informatie van een node te zien geef je de FQDN van de node mee:

```
# puppetman.py -n web.puppets.be
```

## Verwijderen van een node

Met deze optie wordt een node volledig met al zijn Virtuele Hosts, domeinen en sub-domeinen uit de databank verwijderd.

```
# puppetman.py -D <FQDN>
```

# Bronnen

- <http://reductivelabs.com/trac/puppet>
- <http://www.apress.com/book/view/1590599780>
- <http://docs.python.org/genindex-all.html>
- [http://nagios.sourceforge.net/docs/2\\_0/](http://nagios.sourceforge.net/docs/2_0/)
- [http://en.wikipedia.org/wiki/Kickstart\\_\(Linux\)](http://en.wikipedia.org/wiki/Kickstart_(Linux))