

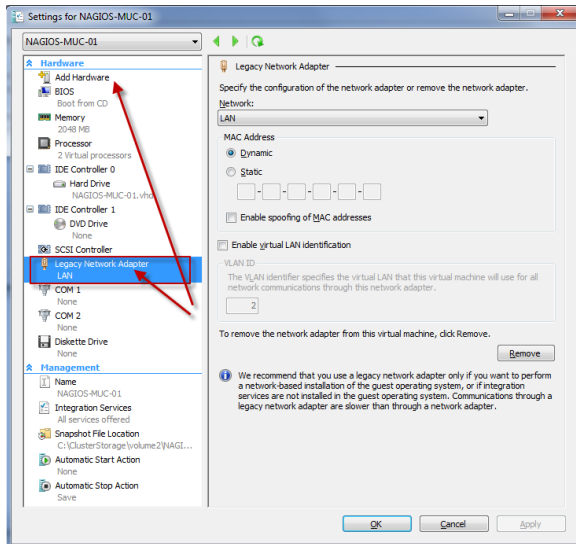
Nagios Server in einer Hyper-V Virtuellen Maschine installieren.

Inhalt

Ubuntu Linux in einer virtuellen Hyper-V Maschine installieren	2
Ubuntu Kernel Update installieren	8
Hyper-V Driver hinzufügen	10
Ubuntu Kernel Kompilieren.....	12
Vorbereitung um NAGIOS installieren zu können.....	14
Nagios installieren	14
Nagios Plugins installieren.....	15
Nagios NDOutils installieren.....	16
Nagios Webseite einrichten	16
Nagios das erste Mal starten.....	17
Remoteverwaltung am Server einschalten	18
PNP4NAGIOS installieren für die Grafische Darstellung	19
PNP4NAGIOS Grundkonfigurieren	20
Das Nagios Web Frontend konfigurieren	24

Ubuntu Linux in einer virtuellen Hyper-V Maschine installieren

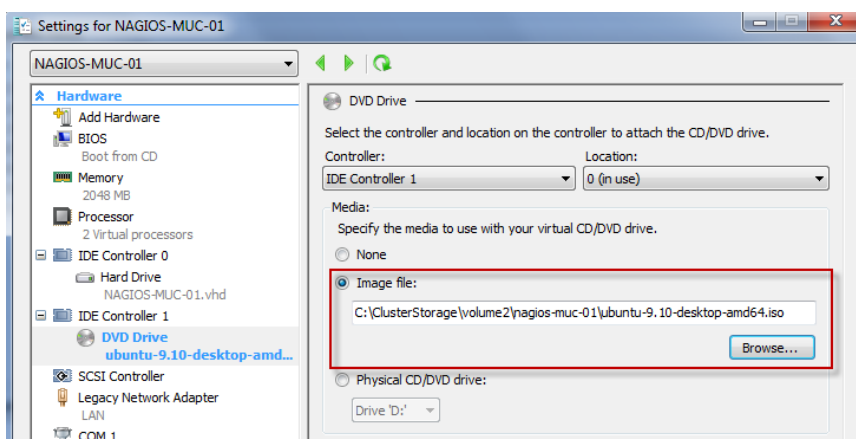
Wir legen über den Failover Cluster Manager oder über die Hyper-V Management Konsole einen neuen Server an.



Um Linux sauber unter Hyper-V installieren zu können müssen wir der Virtuellen Maschine einen Legacy Network Adapter hinzufügen und dann alte Netzwerk Adapter löschen.
Im nächsten Schritt kopieren wir das *"ubuntu-9.10-desktop-amd64.iso"* Image ins lokale Verzeichnis von unserem angelegten Server.

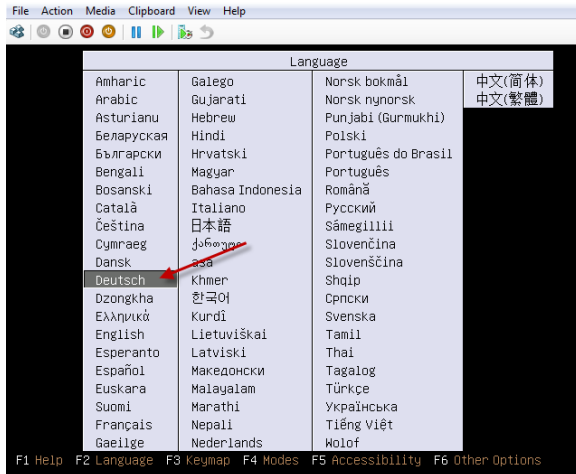
Name	Änderungsdatum	Typ	Größe
Virtual Machines	31.03.2010 10:02	Dateiordner	
NAGIOS-MUC-01.vhd	31.03.2010 10:02	VHD-Datei	202 KB
ubuntu-9.10-desktop-amd64.iso	17.03.2010 08:50	Datenträgerabbild...	707.376 KB

und verknüpfe das Image in meiner Virtuellen Maschine.

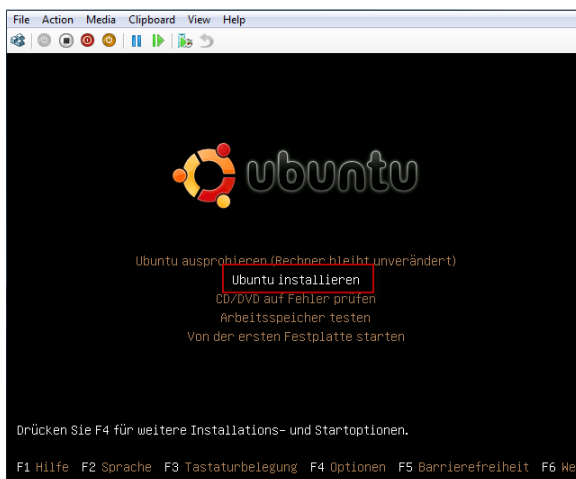


Ubuntu Desktop Client installieren

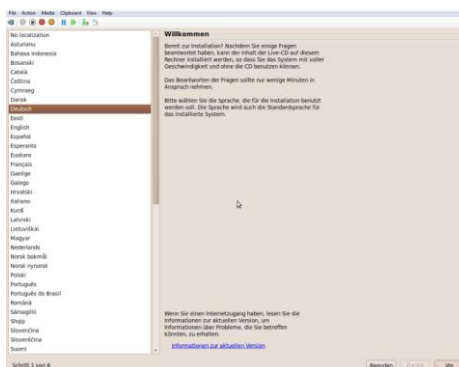
Im nächsten Schritte starte ich den Server und beginne die Installation



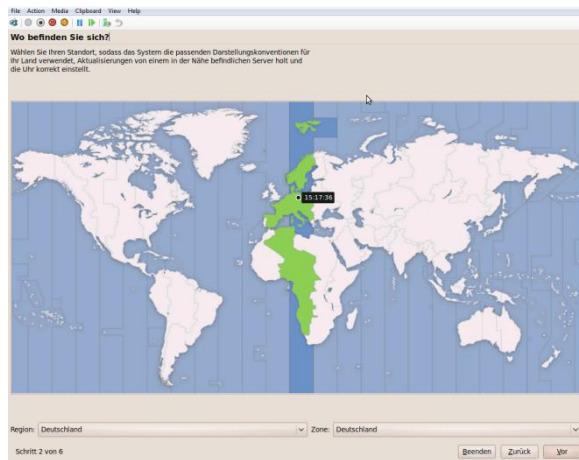
Wähle die Deutsche Sprache mit der Tastatur aus und Klick auf Enter



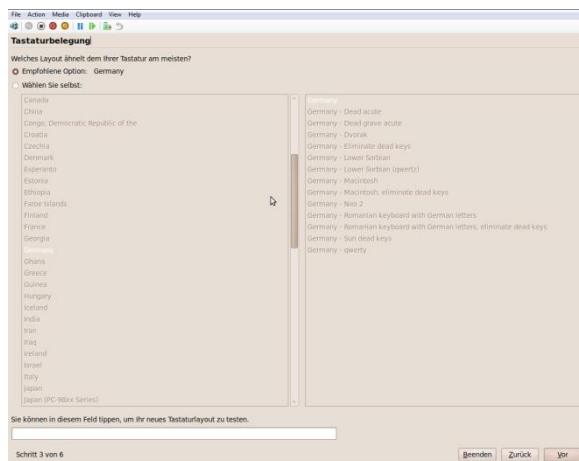
Wähle "**Ubuntu installieren**" aus und Klick auf Enter



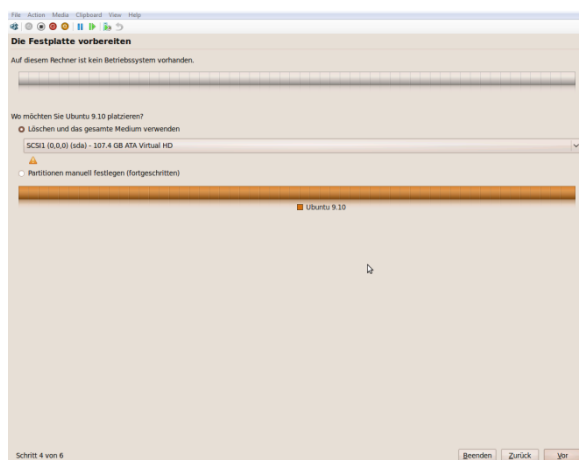
Klicke auf Vor



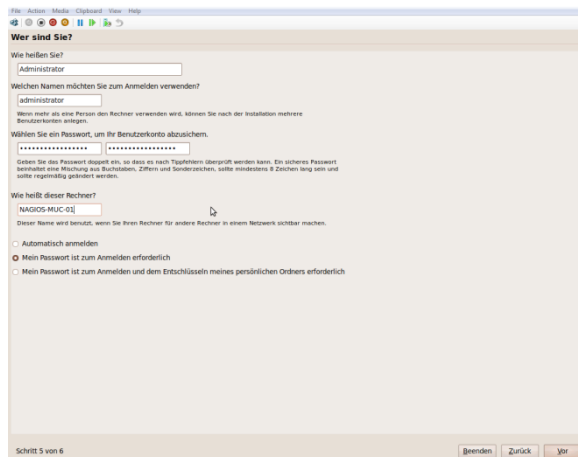
Klicke auf Vor



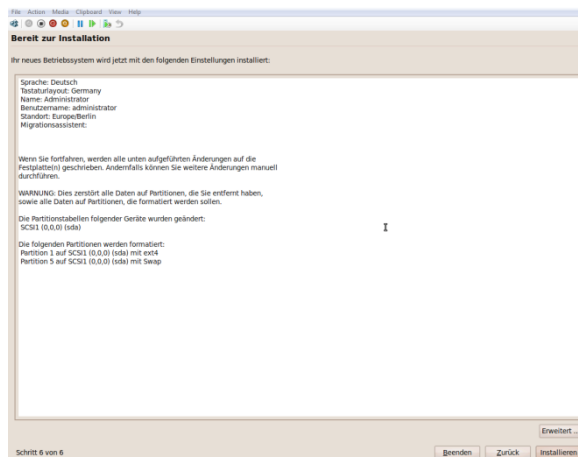
Klicke auf Vor



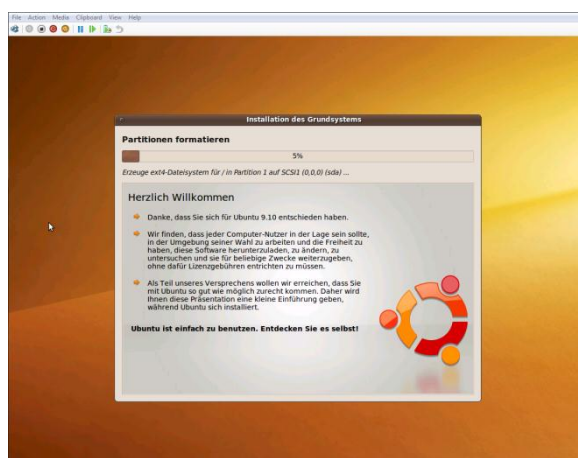
Lasse mir vom System die Festplatte einrichten und klicke auf Vor



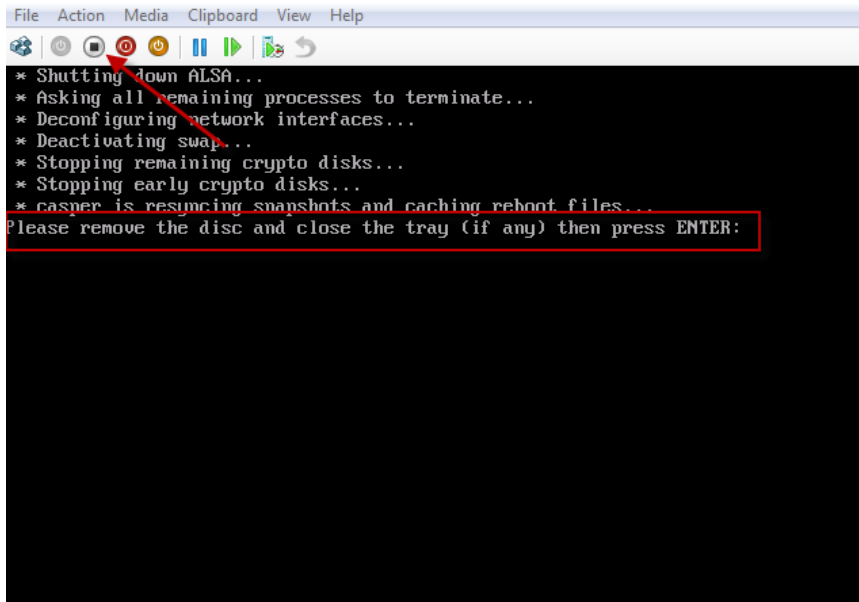
Gebe die Benutzerinformationen ein und klicke auf Vor



klicke auf Installieren

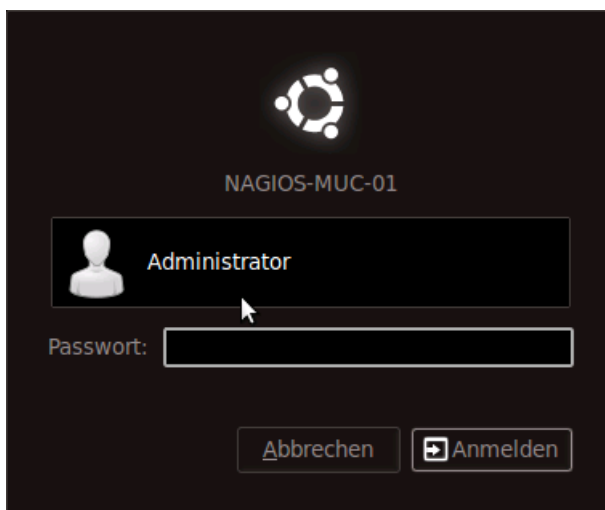


Die Installation beginnt und dauert ca. 20 Minuten. Nach der Installation den Server einmal durchstarten.

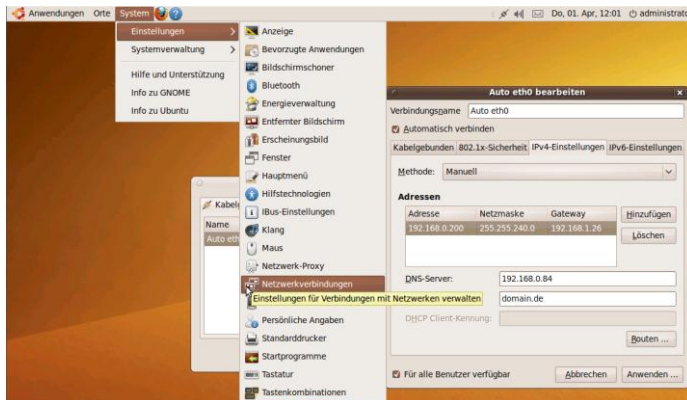


Bei der Meldung **"Please remove the disk and closes the tray....."** schalte ich kurz die Virtuelle Maschine aus und starte sie gleich wieder neu (Hyper-V Server nimmt automatisch das Image aus der CD-Rom Konfiguration raus)

Nach dem Neustart loggen wir uns das erste Mal ein.



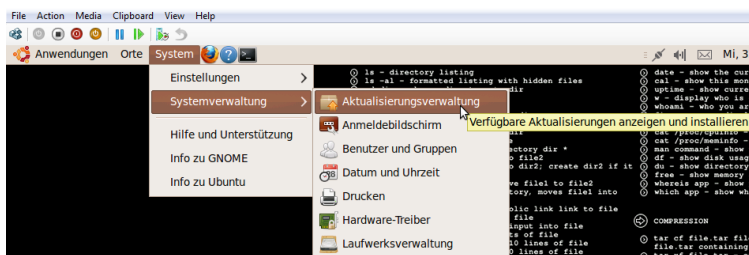
Nach dem Login müssen noch ein paar Patche für Ubuntu installiert werden und eine feste IP-Adresse vergeben werden.



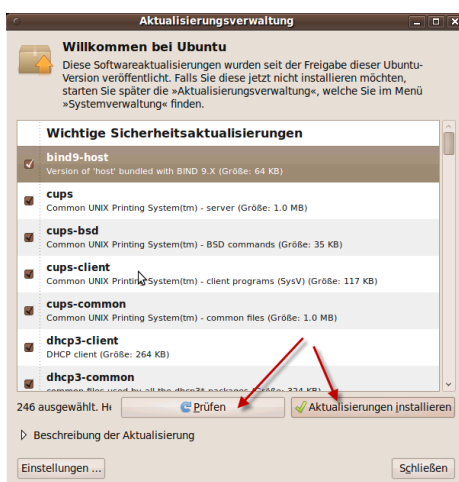
Über System → Einstellungen → Netzwerkverbindungen vergibt man dem Server eine Statische IP-Adresse. (Notfalls den Server nochmals durchstarten, wenn sich die Netzwerkkarte nicht automatisch aktualisiert)

Oder folgenden Befehl eingeben

sudo /etc/init.d/networking



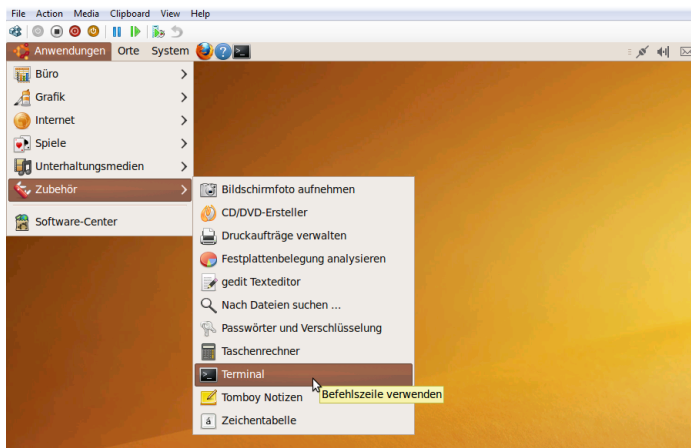
Ich starte die Aktualisierungsverwaltung über **System --> Systemverwaltung --> Aktualisierungsverwaltung**, dort drücke ich den Button **Prüfen** --> gebe das Administratoren Passwort ein und klicke anschließend auf **Aktualisierungen installieren**.



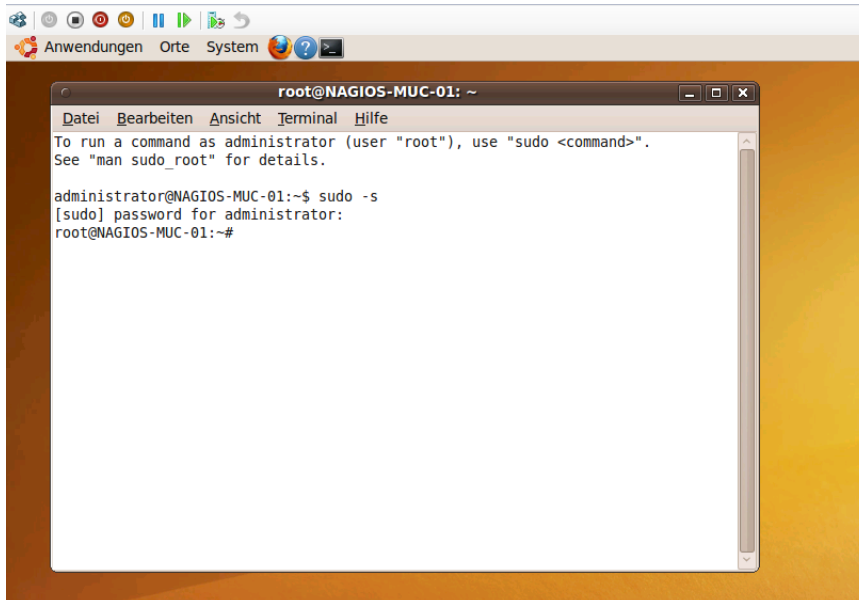
Die Pakete werden von Ubuntu heruntergeladen, und anschließen gleich installiert.

Ubuntu Kernel Update installieren

Danach starte ich den Server neu, und installieren nach dem nächsten Login das Kernelupdate, um die Hyper-V Integration Tools zu integrieren.



Als erstes starte ich eine Terminal Konsole unter **Anwendungen --> Zubehör --> Terminal**



Und gebe meinen ersten Befehl ein, um die Root Rechte in der Konsole zu erhalten.

sudo -s
Password

Jetzt geht es auch schon los.


```

root@NAGIOS-MUC-01: ~
Datei Bearbeiten Ansicht Terminal Hilfe
administrator@NAGIOS-MUC-01:~$ sudo -s
[sudo] password for administrator:
root@NAGIOS-MUC-01:~# apt-get install build-essential
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut
Lese Status-Informationen ein... Fertig
Die folgenden Pakete wurden automatisch installiert und werden nicht länger benö-
tigt:
  linux-headers-2.6.31-14 linux-headers-2.6.31-14-generic
Verwenden Sie »apt-get autoremove«, um sie zu entfernen.
Die folgenden zusätzlichen Pakete werden installiert:
  dpkg-dev fakeroot g++ g++-4.4 libstdc++6-4.4-dev patch
Vorgeschlagene Pakete:
  debian-keyring debian-maintainers g++-multilib g++-4.4-multilib gcc-4.4-doc
  libstdc++6-4.4-dbg libstdc++6-4.4-doc diff-doc
Die folgenden NEUEN Pakete werden installiert:
  build-essential dpkg-dev fakeroot g++ g++-4.4 libstdc++6-4.4-dev patch
0 aktualisiert, 7 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 7.836kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 25,6MB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren [J/n]? j

```

Der erste Befehl für den C Compiler (Entwicklungsumgebung installieren) lautet ***apt-get install build-essential*** danach wird auf den Remote Server von Debian geprüft ob das Paket vorhanden ist, wenn ja wird der Download angestoßen und installiert (daher auch noch die Abfrage ob man fortfahren möchte) Diese Abfrage bestätigt man mit den Buchstaben "J"

apt-get install ncurses-dev

apt-get install kernel-package

Kernel Source Dateien downloaden

Wir wechseln das Verzeichnis ***cd /usr/src*** und starten den Download.

wget -c <http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.10.tar.bz2>

Extrahieren die Kernel Dateien mit folgenden Befehl und wechseln in das Verzeichnis vom Kernel.

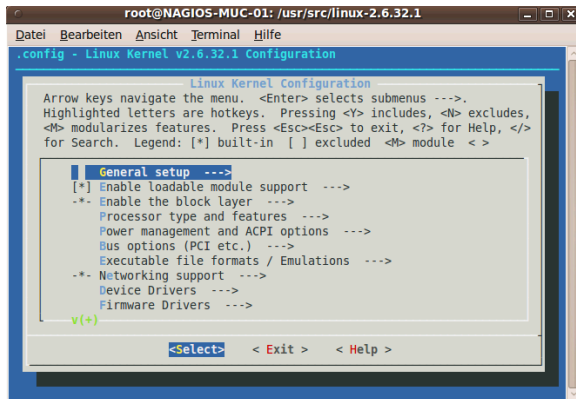
bzip2 -d linux-2.6.32.10.tar.bz2

tar xf linux-2.6.32.10.tar

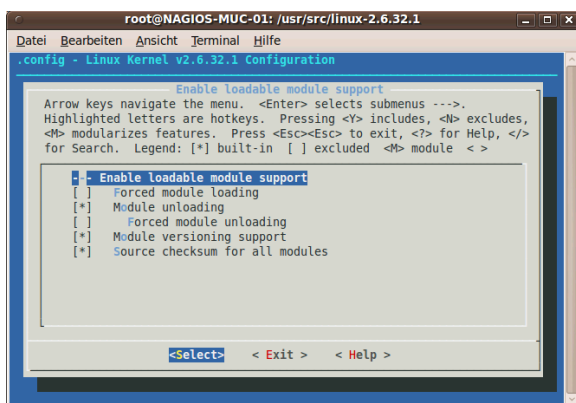
cd linux-2.6.32.10

Hyper-V Driver hinzufügen

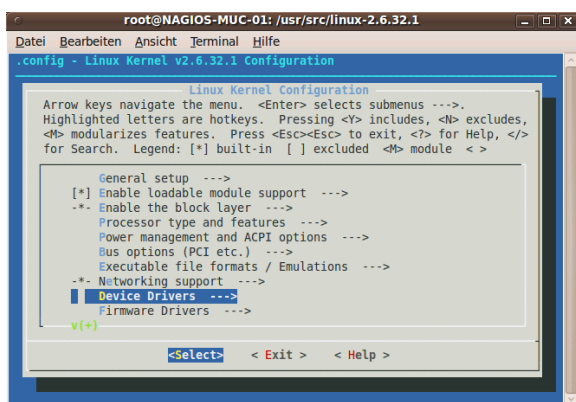
Jetzt bereiten wir die Kernel Dateien für das Kompilieren vor
make menuconfig



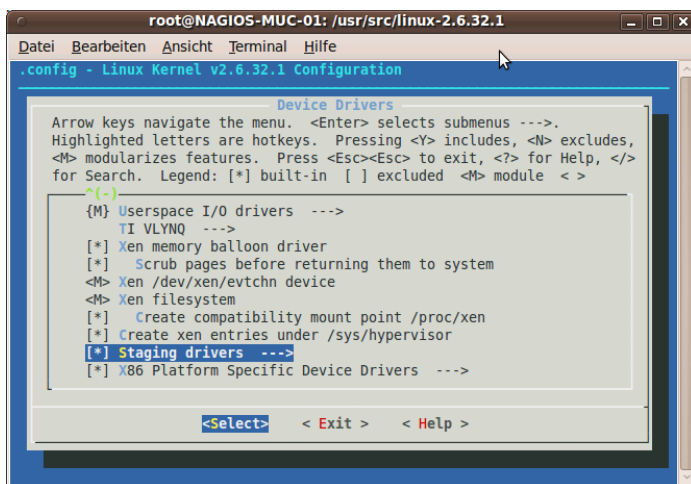
Nach kurzer Zeit startet die Kernel Konfiguration Oberfläche



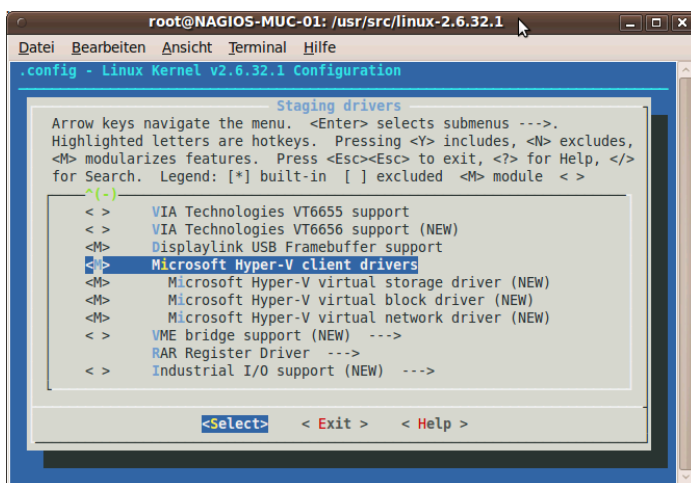
Unter „**Enable loadable module support**“ sieht man alle Module die vorab geladen sind, das sollte aber soweit passen wie oben im Bild gezeigt. (Mit Exit verlassen wir wieder die Oberfläche)



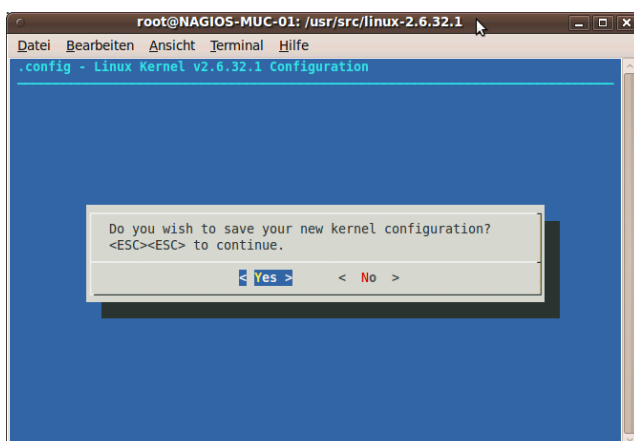
Jetzt klicken wir auf „**Device Driver**“ und bestätigen das mit Enter



Hier scrollen wir fast ganz nach unten zu den „**Staging drivers**“ und bestätigen das wieder mit Enter



Fast ganz unten findet man die „**Microsoft Hyper-V client drivers**“ die wir mit der Leertaste aktivieren und mit Exit verlassen. Alle anderen Fenster verlassen wir ebenfalls mit Exit und Speichern die Kernel Konfiguration.



Wenn wir die Kernelvorbereitung erfolgreich verlassen haben, Kompilieren wir den Kernel mit folgenden Befehl.

Ubuntu Kernel Kompilieren

make-kpkg clean

fakeroot make-kpkg --initrd --revision=hv.custom.1.0 kernel_image

Wenn der letzte Befehl abgesetzt wurde, dauert der Vorgang des Kompilierens ca. 3 Stunden.

```

root@NAGIOS-MUC-01: /usr/src/linux-2.6.32.1
Datei Bearbeiten Ansicht Terminal Hilfe
CC arch/x86/kernel/bootflag.o
CC arch/x86/kernel/e820.o
CC arch/x86/kernel/pci-dma.o
CC arch/x86/kernel/quirks.o
CC arch/x86/kernel/i8237.o
CC arch/x86/kernel/topology.o
CC arch/x86/kernel/kdebugfs.o
CC arch/x86/kernel/alternative.o
CC arch/x86/kernel/i8253.o
CC arch/x86/kernel/pci-nommu.o
CC arch/x86/kernel/tsc.o
CC arch/x86/kernel/io_delay.o
CC arch/x86/kernel/rtc.o
CC arch/x86/kernel/trampoline.o
CC arch/x86/kernel/process.o
CC arch/x86/kernel/i387.o
CC arch/x86/kernel/xsave.o
CC arch/x86/kernel/ptrace.o
CC arch/x86/kernel/tls.o
CC arch/x86/kernel/step.o
CC arch/x86/kernel/stacktrace.o
CC arch/x86/kernel/acpi/boot.o
CC arch/x86/kernel/acpi/sleep.o

```

Nach dem der Kernel kompiliert wurde, wird vom System eine Kerneldatei gebaut die wir jetzt installieren.

cd ../

ls -l

dpkg -i linux-image-2.6.32.10_hv.custom.1.0_amd64.deb

```

root@NAGIOS-MUC-01: /usr/src
Datei Bearbeiten Ansicht Terminal Hilfe
b
Wähle vormals abgewähltes Paket linux-image-2.6.32.1.
(Lese Datenbank ... 137781 Dateien und Verzeichnisse sind derzeit installiert.)
Entpacke linux-image-2.6.32.1 (aus linux-image-2.6.32.1_hv.custom.1.0_amd64.deb)
...
Done.
Richte linux-image-2.6.32.1 ein (hv.custom.1.0) ...
Running depmod.
Finding valid ramdisk creators.
Using mkinitramfs-kpkg to build the ramdisk.
Running postinst hook script update-grub.
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-2.6.32.1
Found initrd image: /boot/initrd.img-2.6.32.1
Found linux image: /boot/vmlinuz-2.6.31-20-generic
Found initrd image: /boot/initrd.img-2.6.31-20-generic
Found linux image: /boot/vmlinuz-2.6.31-14-generic
Found initrd image: /boot/initrd.img-2.6.31-14-generic
Found memtest86+ image: /boot/memtest86+.bin
done
Examining /etc/kernel/postinst.d.
run-parts: executing /etc/kernel/postinst.d/nvidia-common
^[[2~
root@NAGIOS-MUC-01: /usr/src#

```

Nach der Installation vom Kernel, ändern wir die Netzwerkkonfiguration von der Virtuellen Netzwerkkarte. (von Legacy Network Adapter in Network Adapter)

```
sed -i 's/eth/seth/g' /etc/network/interfaces
```

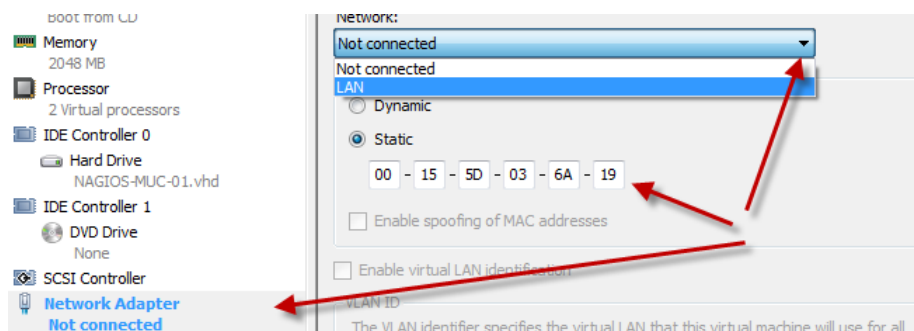
Konfiguriere noch die Netzwerk Module für den Boot Vorgang

```
echo -e "hv_vmbus\hv_storvsc\hv_blkvsc\hv_netvsc" >> /etc/initramfs-tools/modules
update-initramfs -u -k 2.6.32.10
```

Und zu guter Letzt, um die neue Netzwerkkarte ansprechen zu können, fahr ich das System herunter

mit **init 0**

und kopiere mir die MAC Adresse von dem Legacy Adapter, lösche den Legacy Adapter und generiere in Hyper-V einen neuen Netzwerk Adapter mit der alten MAC Adresse.



Nach dem das System neu gestartet wurde, überprüfe ich noch ein paar Punkte, bevor wir mit der Vorbereitung für NAGIOS weitermachen.

ifconfig seth0 → Zeigt mir an ob die Netzwerkkarte sauber angesprochen wird

uname -a → Zeigt mir die Kernel Version an

lsmod | grep hv → Zeigt mir die Treibermodule an

dd if=/dev/zero of=/root/testfile bs=4M count=1000 → Festplatten Performance Test

```

administrator@NAGIOS-MUC-01: ~
Datei Bearbeiten Ansicht Terminal Hilfe
administrator@NAGIOS-MUC-01:~$ ifconfig seth0
seth0    Link encap:Ethernet  Hardware Adresse 00:15:5d:03:6a:1a
         inet Adresse:192.168.0.200  Bcast:192.168.15.255  Maske:255.255.240.0
         inet6-Adresse: fe80::215:5dff:fe03:6a1a/64  Gültigkeitsbereich:Verbindu
ng
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
         RX packets:1785 errors:0 dropped:0 overruns:0 frame:0
         TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
         Kollisionen:0  Sendewarteschlangenlänge:1000
         RX bytes:212592 (212.5 KB)  TX bytes:10470 (10.4 KB)

administrator@NAGIOS-MUC-01:~$ uname -a
Linux NAGIOS-MUC-01 2.6.32.10 #1 SMP Thu Apr 1 11:25:16 CEST 2010 x86_64 GNU/Lin
ux
administrator@NAGIOS-MUC-01:~$ lsmod | grep hv
hv_netvsc          32678  0
hv_blkvsc          27698  3
hv_storvsc         24585  0
hv_vmbus           49273  3 hv_netvsc,hv_blkvsc,hv_storvsc
administrator@NAGIOS-MUC-01:~$

root@NAGIOS-MUC-01:~# dd if=/dev/zero of=/root/testfile bs=4M count=1000
1000+0 Datensätze ein
1000+0 Datensätze aus
4194304000 Bytes (4,2 GB) kopiert, 31,9756 s, 131 MB/s

```

Die ersten drei Tests zeigen mir, dass das Kernelupdate sauber installiert wurde, Netzwerkkarte von Hyper-V wird sauber erkannt und funktioniert, Kernel Version passt, Storage Treiber sind ebenfalls sauber integriert worden.

Daher machen wir weiter mit der Vorbereitung um NAGIOS installieren zu können.

Vorbereitung um NAGIOS installieren zu können

Wir müssen noch einige Pakete nachinstallieren, bevor NAGIOS sauber installiert werden kann.

Ich füge in die Dokumentation nur die Installationsbefehle ein. (Bitte für Internetverbindung sorgen und Root Rechte in der Terminal Konsole)

Web-Server und PHP5

```
apt-get install apache2
```

```
apt-get install php5 php5-mysql
```

```
apt-get install php-pear php5-ldap php5-snmp php5-gd
```

MySQL-Server

```
apt-get install mysql-server-5.0 libmysqlclient15-dev
```

RRDTool

```
apt-get install rrdtool librrds-perl
```

Perl

```
apt-get install libconfig-inifiles-perl
```

SNMP

```
apt-get install snmp snmpd libnet-snmp-perl
```

GD-libraries

```
apt-get install libgd2-xpm libgd2-xpm-dev libpng12-dev
```

SSL (optional)

```
apt-get install openssl libcurl4-openssl-dev
```

Nagios installieren

Die Nagios-Installation besteht aus dem Anlegen des Benutzers „nagios“, der Vergabe eines Passworts, kompilieren der Source Dateien, Kopieren der Dateien und abschließenden Anpassungen.

Anlegen des Benutzers nagios

```
/usr/sbin/useradd -m nagios
```

```
passwd nagios (2 x Passwort eingeben)
```

Anlegen einer Gruppe, um externe Befehle nutzen zu können

```
/usr/sbin/groupadd nagcmd  
/usr/sbin/usermod -G nagcmd,nagios nagios  
/usr/sbin/usermod -G nagcmd www-data
```

Download und Kompilierung

Es gibt immer wieder neue Versionen von Nagios, aber die Installationsroutine ändert sich nicht.

```
cd /usr/local/src/  
wget http://switch.dl.sourceforge.net/sourceforge/nagios/nagios-3.2.1.tar.gz
```

Archiv entpacken und wechseln des Verzeichnisses:

```
tar -xzf nagios-3.2.1.tar.gz  
cd nagios-3.2.1
```

Konfiguration und Installation:

```
./configure --prefix=/usr/local/nagios --with-command-group=nagcmd --enable-nanosleep
```

Bitte untersuchen Sie die Ausgaben auf Fehler. Falls die Meldung kommt, dass die GD-Libraries nicht gefunden werden, dann sollten Sie noch einmal „**sudo apt-get install libgd2-xpm libgd2-xpm-dev libpng12-dev**“ ausführen, möglicherweise wurden die Pakete nicht korrekt installiert. Bei der Fehlermeldung „No package 'glib-2.0' found“ sollte „**sudo apt-get install libglib2-dev**“ helfen.

```
make all  
make install  
make install-init  
make install-commandmode  
make install-config
```

Nagios Plugins installieren

Nagios Plugins downloaden

```
cd /usr/local/src  
wget http://surfnet.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.14.tar.gz
```

Archiv entpacken und wechseln des Verzeichnisses:

```
tar -xzf nagios-plugins-1.4.14.tar.gz  
cd nagios-plugins-1.4.14
```

Konfiguration und Installation:

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios  
make  
make install
```

Nagios NDOutils installieren

Download des Archivs

```
cd /usr/local/src  
wget http://surfnet.dl.sourceforge.net/sourceforge/nagios/ndoutils-1.4b9.tar.gz
```

Archiv entpacken und wechseln des Verzeichnisses:

```
tar -xzf ndoutils-1.4b9.tar.gz  
cd ndoutils-1.4b9
```

Konfiguration und Installation:

```
./configure --prefix=/usr/local/nagios/ --enable-mysql --disable-pgsql  
make  
make install
```

Nach der Erstellung der Binaries und Libraries müssen diese noch kopiert werden

```
cp /usr/local/src/ndoutils-1.4b9/src/ndomod-*.o /usr/local/nagios/bin/  
cp /usr/local/src/ndoutils-1.4b9/src/ndo2db* /usr/local/nagios/bin/
```

Nagios Webseite einrichten

Nagios-Web-Konfigurationsdateien ins Apache conf.d-Verzeichnis installieren

```
make install-webconf
```

nagiosadmin-Konto anlegen, um sich am Web-Interface anmelden zu können.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users administrator (2 x Passwort eingeben)
```


Apache Server neu starten, damit die Änderungen wirksam werden.

/etc/init.d/apache2 reload

Nagios zu der Liste der System-Services hinzufügen um Nagios automatisch zu starten wenn der Server hochfährt.

```
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc2.d/S99nagios
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc3.d/S99nagios
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc4.d/S99nagios
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc5.d/S99nagios
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc0.d/K99nagios
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc1.d/K99nagios
/usr/local/src/nagios-3.2.1# ln -s /etc/init.d/nagios /etc/rc6.d/K99nagios
```

Nagios das erste Mal starten

Die Nagios Webseite das erste Mal im Browser aufrufen

<http://servername/nagios>

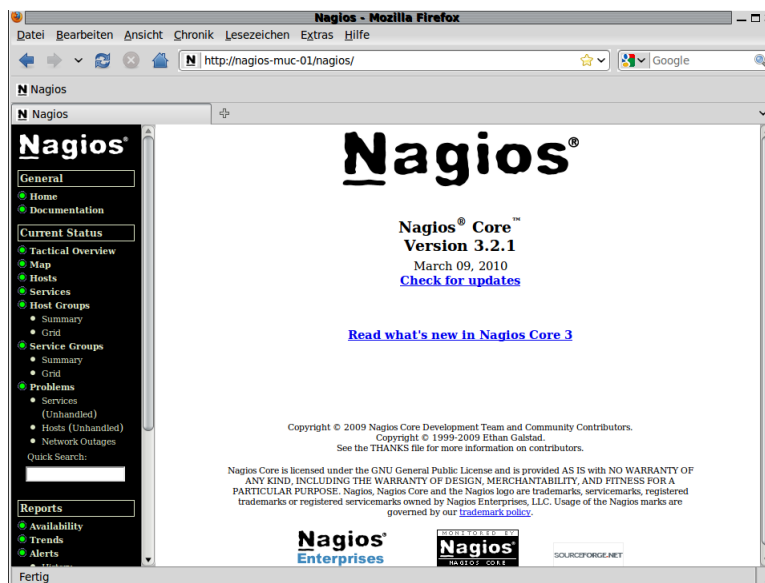
Username: administrator

Passwort: *****

oder

Username: nagiosadmin

Passwort: nagios



OK die Nagios Webseite ist erreichbar.

Möchte man noch ein oder zwei User der `htpasswd.users` Datei hinzufügen um auf das Webinterface von Nagios zugreifen zu können.

Benutzt man folgenden Befehl:

`htpasswd /etc/local/nagios/etc/htpasswd.users neuerBenutzer (2x Passwort)`

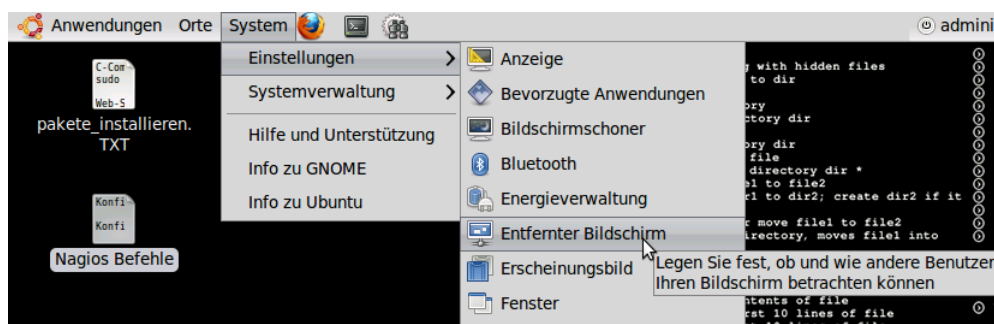
Der User ist aber nicht frei wählbar, sondern muss mit den Namen eines Kontakt übereinstimmen, lediglich die Gruppe `www-data` muss die Datei lesen können, daher ändern wir noch die Rechte für die Datei.

`chown www-data /etc/local/nagios/etc/htpasswd.users`

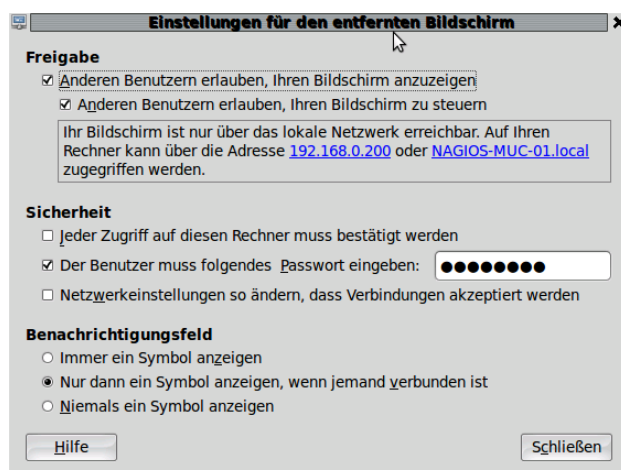
`chmod 600 /etc/local/nagios/etc/htpasswd.users`

Remoteverwaltung am Server einschalten

Um den Nagios Server besser verwalten zu können schalte ich noch die Remoteverwaltung von VNC ein. Über **System → Einstellungen → Entfernter Bildschirm**



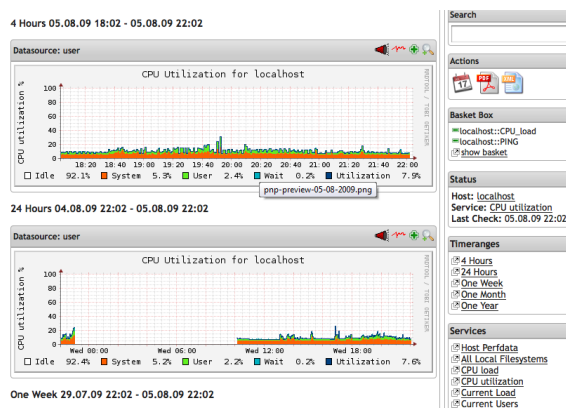
wird die Einstellung vorgenommen.



Hierzu setze ich die Häkchen wie oben gezeigt und vergebe noch ein Passwort. Danach kann mit den VNC Viewer auf den Server zugegriffen werden.

PNP4NAGIOS installieren für die Grafische Darstellung

PNP4NAGIOS dient dazu, die Werte die Nagios von den einzelnen Servern sammelt bzw. erhält grafisch aufzubereiten.



Wir öffnen wieder ein Terminal und melden uns als Administrator an der Konsole an.

sudo -s (Passwort)

danach stellen wir eine Verbindung zum Internet her und starten den Download.

cd /usr/local/src

wget <http://sourceforge.net/projects/pnp4nagios/files/PNP-0.6/pnp4nagios-0.6.3.tar.gz>

Entpacken die ZIP Datei und wechseln in das Verzeichnis.

tar -xvzf pnp4nagios-0.6.3.tar.gz

cd pnp4nagios-0.6.3

Jetzt bereiten wir die Dateien für die Installation vor.

./configure

Nach kurzer Zeit sollte man folgende Aussage zu sehen bekommen, diese Ausgabe überprüfen ob alle Dateipfade passen und die Installation von PNP4Nagios anstoßen.

*** Configuration summary for pnp4nagios-0.6.3 03-16-2010 ***

General Options:

```

-----
Nagios user/group:      nagios nagios
Install directory:     /usr/local/pnp4nagios
HTML Dir:              /usr/local/pnp4nagios/share
Config Dir:            /usr/local/pnp4nagios/etc
Location of rrdtool binary: /usr/bin/rrdtool Version 1.3.1
RRDs Perl Modules:     FOUND (Version 1.3001)
RRD Files stored in:   /usr/local/pnp4nagios/var/perfdata
process_perfdata.pl Logfile: /usr/local/pnp4nagios/var/perfdata.log
Perfdata files (NPCD) stored in: /usr/local/pnp4nagios/var/spool
Web Interface Options: -----
HTML URL:              http://localhost/pnp4nagios/
Apache Config File:    /etc/apache2/conf.d/pnp4nagios.conf
Review the options above for accuracy. If they look okay,
type 'make all' to compile.

```

ein **make /all** kompiliert nun die in C geschriebenen Komponenten wie NPCD
make /install kopiert alles an die richtige Stelle im Dateisystem. Die Pfade wurden ja beim

./configure bereits gezeigt. Nach der Installation der Programm- und HTML-Dateien wird mit
make install-webconf eine Konfigurationsdatei in das Konfigurationsverzeichnis des Apache-Web-
Servers
kopiert. Danach kann noch ein **make install-config** aufgerufen werden. Damit werden Config-Files für
process_perfdata.pl und npcd nach etc/pnp kopiert.

Wird das INIT Script für den NPCD benötigt, so sorgt **make install-init**

Wenn das alles sauber durchgelaufen ist wir der Apache Webserver neu gestartet mit

/etc/init.d/apache2 restart

PNP4NAGIOS Grundkonfigurieren

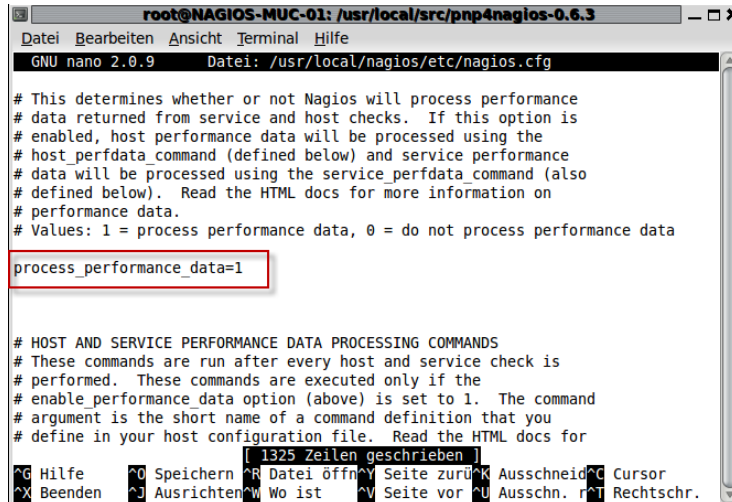
Wir konfigurieren das Ganze im BULK-Modus. Der Bulk-Mode ist etwas komplizierter als der Synchronous-Mode, reduziert die Last auf dem Nagios Server jedoch merklich, da nun nicht mehr für jeden Service bzw. Host zusätzlich der Datensammler process_perfdata.pl gestartet werden muss.

Im Bulk-Mode schreibt Nagios die Daten in einem definierten Format in eine temporäre Datei. Diese Datei wiederum wird periodisch von process_perfdata.pl verarbeitet. Um den Start und den Intervall

kümmert sich dabei Nagios selbst. Auch hier muss die Verarbeitung der Performance-Daten in der `nagios.cfg` eingeschaltet werden.

Wir öffnen die `nagios.cfg` mit folgenden Befehl

`nano /usr/local/nagios/etc/nagios.cfg`



```

root@NAGIOS-MUC-01: /usr/local/src/pnp4nagios-0.6.3
Datei Bearbeiten Ansicht Terminal Hilfe
GNU nano 2.0.9 Datei: /usr/local/nagios/etc/nagios.cfg

# This determines whether or not Nagios will process performance
# data returned from service and host checks. If this option is
# enabled, host performance data will be processed using the
# host_perfdata_command (defined below) and service performance
# data will be processed using the service_perfdata_command (also
# defined below). Read the HTML docs for more information on
# performance data.
# Values: 1 = process performance data, 0 = do not process performance data

process_performance_data=1

# HOST AND SERVICE PERFORMANCE DATA PROCESSING COMMANDS
# These commands are run after every host and service check is
# performed. These commands are executed only if the
# enable_performance_data option (above) is set to 1. The command
# argument is the short name of a command definition that you
# define in your host configuration file. Read the HTML docs for

```

Suchen den Eintrag **`process_performance_data=0`** und ändern diesen Eintrag auf **`process_performance_data=1`** ab. Denn Eintrag speichern wir mit **`STRG + O + Enter`**

Zusätzlich müssen noch folgende Parameter der `nagios.cfg` Datei hinzugefügt werden.

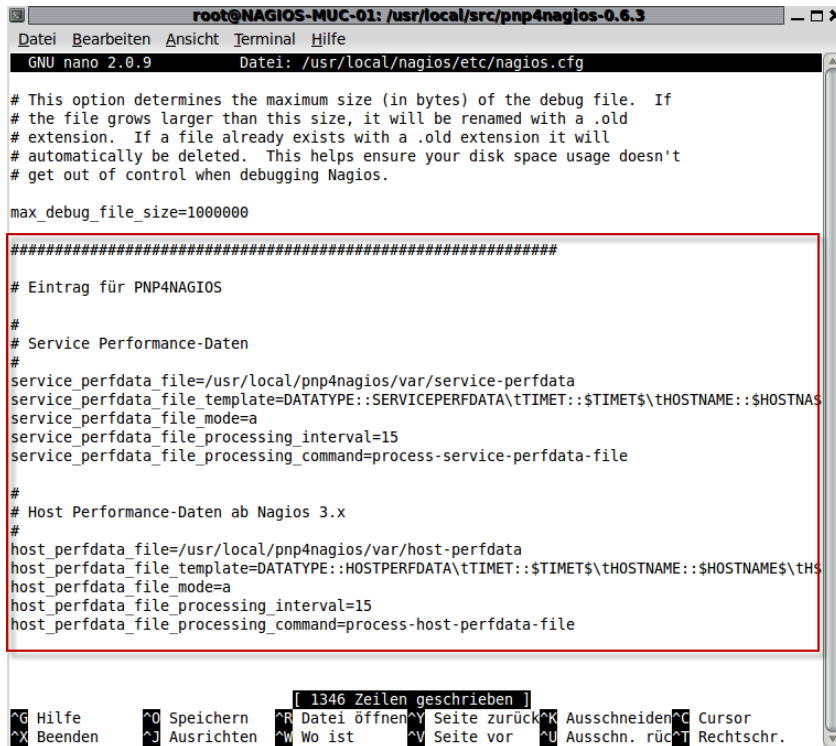
```

#
# Service Performance-Daten
#
service_perfddata_file=/usr/local/pnp4nagios/var/service-perfddata
service_perfddata_file_template=DATATYPE::SERVICEPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$HOST
OSTNAME$\tSERVICEDESC::$SERVICEDESC$\tSERVICEPERFDATA::$SERVICEPERFDATA$\tSERVICECH
ECKCOMMAND::$SERVICECHECKCOMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::$HO
STSTATETYPE$\tSERVICESTATE::$SERVICESTATE$\tSERVICESTATETYPE::$SERVICESTATETYPE$
service_perfddata_file_mode=a
service_perfddata_file_processing_interval=15
service_perfddata_file_processing_command=process-service-perfddata-file

#
# Host Performance-Daten ab Nagios 3.x
#
host_perfddata_file=/usr/local/pnp4nagios/var/host-perfddata
host_perfddata_file_template=DATATYPE::HOSTPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$HOST
NAME$\tHOSTPERFDATA::$HOSTPERFDATA$\tHOSTCHECKCOMMAND::$HOSTCHECKCOMMAND$\t
HOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::$HOSTSTATETYPE$
host_perfddata_file_mode=a
host_perfddata_file_processing_interval=15
host_perfddata_file_processing_command=process-host-perfddata-file

```

Daher gehen wir an das Ende der Datei und kopieren die oberen Parameter in die nagios.cfg, mit **STRG + O + Enter** das Ganze speichern und mit **STRG + X** das Programm nano verlassen.



```

root@NAGIOS-MUC-01: /usr/local/src/pnp4nagios-0.6.3
Datei Bearbeiten Ansicht Terminal Hilfe
GNU nano 2.0.9 Datei: /usr/local/nagios/etc/nagios.cfg

# This option determines the maximum size (in bytes) of the debug file. If
# the file grows larger than this size, it will be renamed with a .old
# extension. If a file already exists with a .old extension it will
# automatically be deleted. This helps ensure your disk space usage doesn't
# get out of control when debugging Nagios.

max_debug_file_size=1000000

#####

# Eintrag für PNP4NAGIOS
#
# Service Performance-Daten
#
service_perfdata_file=/usr/local/pnp4nagios/var/service-perfdata
service_perfdata_file_template=DATATYPE::SERVICEPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$HOSTNAME
service_perfdata_file_mode=a
service_perfdata_file_processing_interval=15
service_perfdata_file_processing_command=process-service-perfdata-file

#
# Host Performance-Daten ab Nagios 3.x
#
host_perfdata_file=/usr/local/pnp4nagios/var/host-perfdata
host_perfdata_file_template=DATATYPE::HOSTPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$HOSTNAME\tHOST
host_perfdata_file_mode=a
host_perfdata_file_processing_interval=15
host_perfdata_file_processing_command=process-host-perfdata-file

[ 1346 Zeilen geschrieben ]
^G Hilfe      ^O Speichern  ^R Datei öffnen ^Y Seite zurück ^K Ausschneiden ^C Cursor
^X Beenden    ^J Ausrichten ^W Wo ist      ^V Seite vor  ^U Ausschn. rück ^T Rechtschr.

```

Die Parameter und deren Bedeutung im Einzelnen:

- **service_perfdata_file** Der Pfad zur temporären Datei, in der die Daten gesammelt werden sollen.
- **service_perfdata_file_template** Das Format der temporären Datei. Hier werden die Daten über Nagios-Macros definiert.
- **service_perfdata_file_mode** Die Option "a" definiert, dass an die Datei angehängen werden soll.
- **service_perfdata_file_processing_interval** Das Intervall beträgt 15 Sekunden
- **service_perfdata_file_processing_command** das Command, das im definierten Intervall aufgerufen werden soll.

Als nächstes müssen wir Nagios noch ein paar Commandos mitteilen, dazu öffnen wir die bereits vorher angelegte Datei *commands.cfg* mit folgendem Befehl

nano /usr/local/nagios/etc/fctgroup

gehen ans Ende der Datei und kopieren folgendes in die Datei

```
define command{
    command_name process-service-perfdata-file
    command_line /usr/local/pnp4nagios/libexec/process_perfdata.pl --
bulk=/usr/local/pnp4nagios/var/service-perfdata
}

define command{
    command_name process-host-perfdata-file
    command_line /usr/local/pnp4nagios/libexec/process_perfdata.pl --
bulk=/usr/local/pnp4nagios/var/host-perfdata
}
```

mit **STRG + O + Enter** das Ganze speichern und mit **STRG + X** das Programm nano verlassen.

Wenn bis jetzt alles sauber funktioniert hat, kann PNP zum ersten Mal im Browser aufgerufen werden. Bei der Installation mit den Standardeinstellungen erfolgt der Aufruf über **<http://nagios-muc-01/pnp4nagios>**

Dann sollte man folgendes sehen

PNP4Nagios Environment Tests

The following options are determined by "configure". If any of the tests have failed, consult the [documentation](#) for more information on how to correct the problem.

PNP4Nagios Version	pnp4nagios-0.6.3
Prefix	/usr/local/pnp4nagios
RRD Storage	/usr/local/pnp4nagios/var/perfdata is readable.
RRDtool Binary	/usr/bin/rrdtool is executable by PHP
PHP GD extension	Pass
PHP function proc_open()	Pass
PHP zlib extension	Pass
PHP session extension	Pass
PHP JSON extension	Pass
PHP magic_quotes_gpc	PHP magic_quotes_gpc is deprecated
Apache Rewrite Module	Pass

Wenn das Apache Rewrite Module deaktiviert ist kann man es mit folgenden Befehl aktivieren.

a2enmod rewrite

Falls die Meldung "**PHP magic_quotes_gpc is deprecated**" zu sehen ist, dann setzen wir in der php.ini den Wert auf Off.

nano /etc/php5/apache2/php.ini

Suchen den Eintrag mit **STRG + W**, Speichern den Eintrag mit **STRG + O + Enter** und verlassen das Programm mit **STRG + X**


```

root@NAGIOS-MUC-01: /usr/local/src/pnp4nagios-0.6.3
Datei Bearbeiten Ansicht Terminal Hilfe
GNU nano 2.8.9 Datei: /etc/php5/apache2/php.ini

; would contain the GET information).  If you don't use these variables, you
; should turn it off for increased performance.
register_argc_argv = On

; When enabled, the SERVER and ENV variables are created when they're first
; used (Just In Time) instead of when the script starts. If these variables
; are not used within a script, having this directive on will result in a
; performance gain. The PHP directives register_globals, register_long_arrays,
; and register_argc_argv must be disabled for this directive to have any affect.
auto_globals_jit = On

; Maximum size of POST data that PHP will accept.
post_max_size = 8M

; Magic quotes
;
; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = Off

; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with '' instead of \').
magic_quotes_sybase = Off

; Automatically add files before or after any PHP document.
auto_prepend_file =

[ 1254 Zeilen geschrieben ]

```

danach starten wir den Apache Server neu um die php.ini neu einzulesen.

/etc/init.d/apache2 restart

Nach erneutem Aufruf von PNP4NAGIOS im Webbrowser sollte alles passen.

PNP4Nagios Environment Tests

The following options are determined by "configure". If any of the tests have failed, consult the [documentation](#) for more information on how to correct the problem.

PNP4Nagios Version	pnp4nagios-0.6.3
Prefix	/usr/local/pnp4nagios
RRD Storage	/usr/local/pnp4nagios/var/perfdata is readable.
RRDtool Binary	/usr/bin/rrdtool is executable by PHP
PHP GD extension	Pass
PHP function proc_open()	Pass
PHP zlib extension	Pass
PHP session extension	Pass
PHP JSON extension	Pass
PHP magic_quotes_gpc	Off
Apache Rewrite Module	Pass

Das Nagios Web Frontend konfigurieren

Als erstes definieren wir zwei neue Nagios-Templates. Diese Templates kopieren wir in die bereits bestehende Datei **template.cfg**

```

define host {
    name    host-pnp
    register 0
    action_url /pnp4nagios/graph?host=$HOSTNAME$&srv=_HOST_' class='tips'
    rel='/pnp4nagios/popup?host=$HOSTNAME$&srv=_HOST_
}

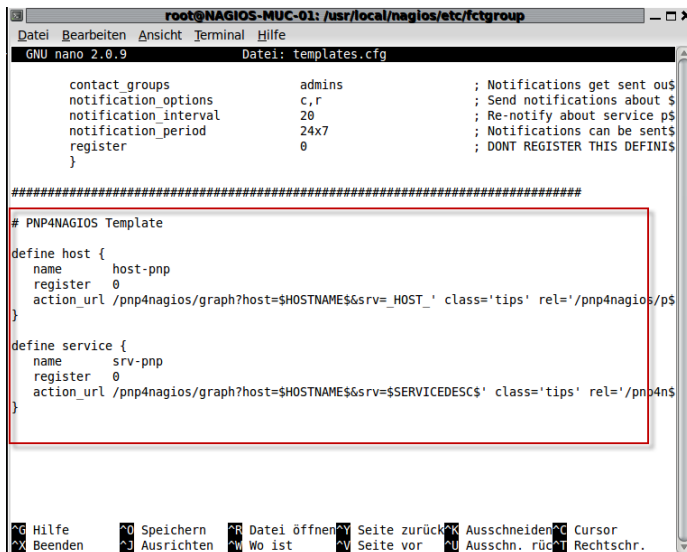
```



```
define service {
    name    srv-pnp
    register 0
    action_url /pnp4nagios/graph?host=$HOSTNAME&srv=$SERVICEDESC' class='tips'
    rel='/pnp4nagios/popup?host=$HOSTNAME&srv=$SERVICEDESC$
}
```

Wir öffnen wieder mit dem Programm nano folgende Datei

nano /usr/local/nagios/etc/fctgroup/template.cfg



```
root@NAGIOS-MUC-01: /usr/local/nagios/etc/fctgroup
Datei Bearbeiten Ansicht Terminal Hilfe
GNU nano 2.0.9 Datei: templates.cfg

contact_groups      admins          ; Notifications get sent ou$
notification_options c,r              ; Send notifications about $
notification_interval 20              ; Re-notify about service p$
notification_period   24x7            ; Notifications can be sent$
register             0                ; DONT REGISTER THIS DEFINIS
}

#####

# PNP4NAGIOS Template

define host {
    name    host-pnp
    register 0
    action_url /pnp4nagios/graph?host=$HOSTNAME&srv=_HOST_' class='tips' rel='/pnp4nagios/p$
}

define service {
    name    srv-pnp
    register 0
    action_url /pnp4nagios/graph?host=$HOSTNAME&srv=$SERVICEDESC$' class='tips' rel='/pnp4n$
}

Hilfe Speichern Datei öffnen Seite zurück Ausschneiden Cursor
Beenden Ausrichten Wo ist Seite vor Ausschn. rück Rechtschr.
```

gehen ans Ende der Datei und kopieren die beiden Templates dort hinein. Mit **STRG + O + Enter** speichern wir das Ganze und mit **STRG + X** verlassen wir das Programm.