

# ICINGA 1.0.2 in einer Microsoft Hyper-V Maschine mit Ubuntu 10.04 Desktop installieren.

## Inhalt

ICINGA 1.0.2 Einführung.....	3
Ubuntu 10.04 Desktop Installation .....	4
Hyper-V Addons aktivieren .....	7
Ubuntu 10.04 Updates installieren .....	9
Ubuntu 10.04 vorbereiten für die Icinga 1.0.2 Installation.....	9
Benutzer und Gruppen für Icinga 1.0.2 anlegen .....	10
ICINGA 1.0.2 und die IDOUtils kompilieren und installieren.....	11
Contact.cfg bearbeiten.....	12
MySql Datenbank anlegen .....	13
Icinga 1.0.2 Webkomponenten installieren .....	14
Nagios Plugins 1.4.15 installieren.....	15
ICINGA 1.0.2 Konfiguration überprüfen .....	15
Installation des neuen Icinga 1.0.1 Web Frontend .....	17
Datenbank-Installation.....	19
Anlegen der Datenbank.....	19
Icinga-Web Konfiguration.....	20
Apache-Konfiguration .....	20
ICINGA Grundkonfiguration .....	23
Ordnerstruktur anlegen.....	23
hostgroup.cfg .....	24
localhost.cfg .....	25
templates.cfg.....	25
icinga.cfg.....	28
timeperiods.cfg.....	29
icinga.fritz.box.cfg .....	30
tnetd.fritz.box.cfg.....	32

Windows Firewall .....	36
commands.cfg .....	38
fritz.box.cfg .....	39
E-Mail Benachrichtigung auf dem Server aktivieren .....	39

## ICINGA 1.0.2 Einführung

Diese Anleitung ist dazu gedacht, dir ein einfaches Howto zu liefern, wie die Monitoring Software ICINGA 1.0.2 innerhalb von 60 Minuten aus dem Quellcode installiert und deinen lokalen Rechner überwacht. Dieses Howto enthält Anweisungen für eine einzige Linux-Distributionen: Ubuntu 10.04 Desktop. Ähnliche Distributionen werden wahrscheinlich auch funktionieren, darunter Fedora, OpenSuse, RedHat, CentOS, Debian und SLES. Leider habe ich persönlich nicht die Zeit alle Distributionen zu testen bzw. zu Dokumentieren.

Was dabei herauskommt:

Du wirst am Ende ein lauffähiges ICINGA 1.0.2 System haben:

- Icinga 1.0.2 und die Nagios Plugins 1.4.14 werden unterhalb von `/usr/local/icinga` installiert sein
- Icinga 1.0.2 wird so konfiguriert sein, dass es einige Dinge auf deinem lokalen System bzw. entfernten Windows System überwacht (CPU-Auslastung, Plattenbelegung, usw.)
- Das Standard Icinga-Web-Interface ist später unter <http://servername/icinga> erreichbar.
- Das neue Icinga Web-Interface ist später unter <http://servername/icinga-web> erreichbar.
- Lauffähige E-Mail Benachrichtigung

Eine Datenbank, die von Icinga 1.0.2 mit Hilfe von IDOUtils gefüllt wird

Voraussetzungen:

Bei einigen Teilen der Installation benötigst du root-Zugang. IDOUtils benutzt die libdbi und die libdbi-Treiber für verschiedene Datenbanken. Die Development-Libraries werden ebenfalls benötigt. Die folgenden Beispiele zeigen, wie die IDOUtils mit libdbi für MySQL installiert werden.

Stell bitte sicher, dass die folgenden Pakete installiert sind, bevor du fortfährst.

- Apache Web Server
- GCC-Compiler
- C/C++ development libraries
- GD-Development-Libraries
- libdbi/libdbi-Treiber, eine Datenbank wie z.B. MySQL

Meine Testumgebung ist ein Ubuntu 10.04 Desktop System, das in einer Microsoft Hyper-V Server R2 Virtuellen Maschine installiert wurde.

Also viel Spaß beim Lesen und nachmachen. ☺

Referenzen für dieses Howto waren:

NAGIOS Portal → <http://www.nagios-portal.org> und die Icinga Webseite → <http://www.icinga.org/>

Weitere Howto's findest du auf meiner Webseite <http://www.thurnhofer.net>

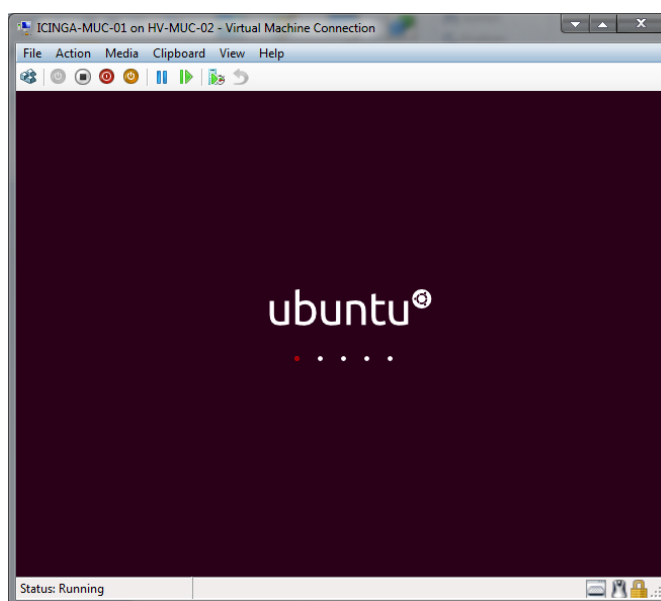
## Ubuntu 10.04 Desktop Installation

Auf folgender Webseite bekommt man das ISO Image für Ubuntu 10.04 Desktop x86/x64

X86 → <http://releases.ubuntu.com/lucid/ubuntu-10.04-desktop-i386.iso>

X64 → <http://releases.ubuntu.com/lucid/ubuntu-10.04-desktop-amd64.iso>

In Microsoft Hyper-V lege ich eine Virtuelle Maschine an und binde das ISO Image ein, danach starte ich die Virtuelle Maschine und beginne die Installation von Ubuntu 10.04 Desktop.



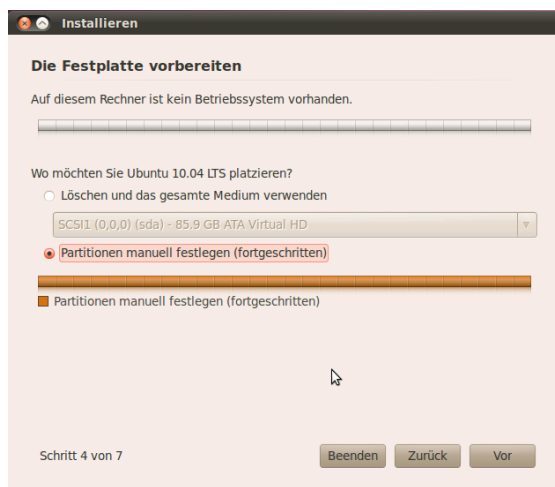
Wähle Deutsch aus und klicke auf **Ubuntu 10.04 LTS installieren**



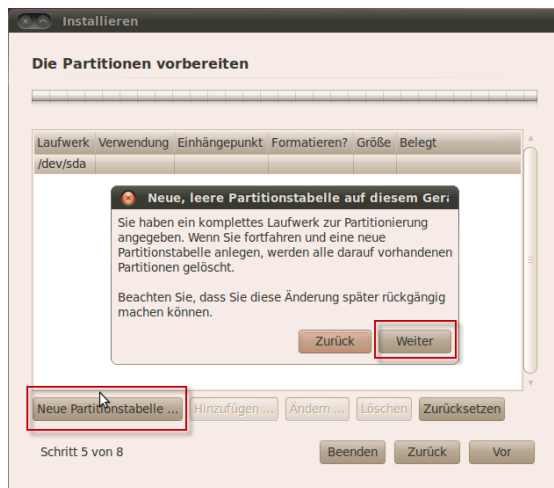
Übernehme die Standardzeitzone und klicke auf **Vor**



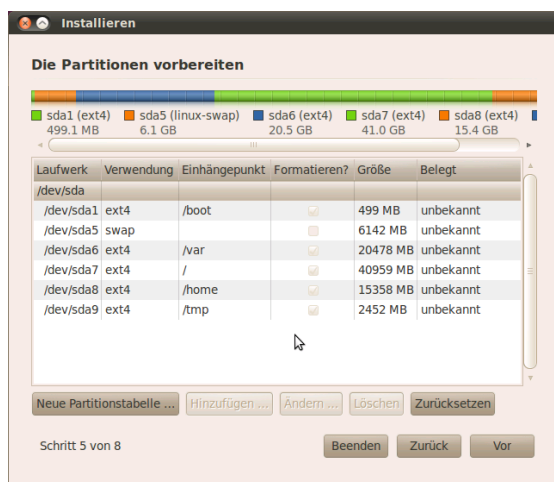
Auch bei der Tastaturbelegung belasse ich den Standard und klicke auf **Vor**



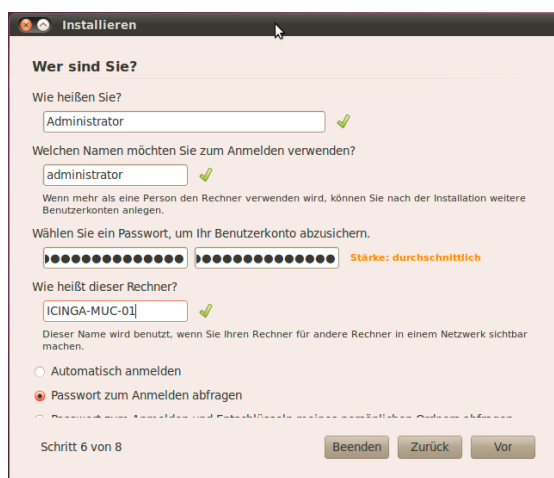
Bei der Festplattenpartitionierung wähle ich den **Manuellen Modus** aus und klicke auf **Vor**



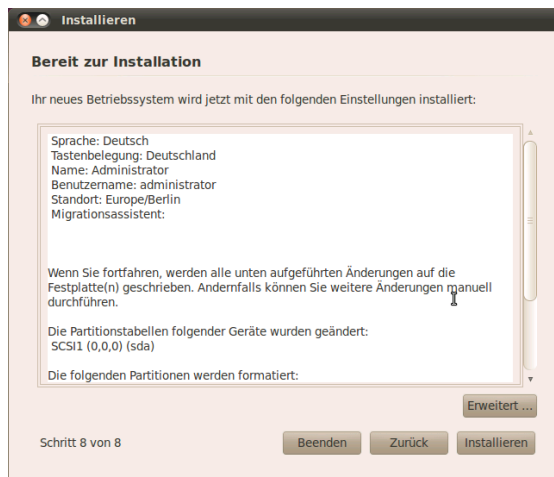
Nun wird eine **neue Partitionstabelle** angelegt (Warnmeldung mit Weiter bestätigen)



Partitionen werden wie im Bild angelegt (bei einer 80GB Partition) 500MB /boot EXT4, 1,5-2\*RAM SWAP, 40GB / EXT4, 2,5GB /tmp EXT4, 20GB /var EXT4, REST /home EXT4



Benutzerinformationen eingeben und auf **Vor** klicken



Installation starten/abschließen/Server durchstarten

## Hyper-V Addons aktivieren

Nach dem ersten Neustart → loggen wir uns ins System ein → und starten ein Terminal über Anwendungen → Zubehör → Terminal

Mit den Befehl **sudo -s** und dem (Root Passwort) wird man zum Admin in der Konsole. Mit dem Text Editor Programm **nano** bearbeite ich im Verlauf des Howto's meine Konfigurationsdateien.

Ich öffne die Konfigurationsdatei **/etc/initramfs-tools/modules** und aktiviere folgende Hyper-V Module/Treiber.

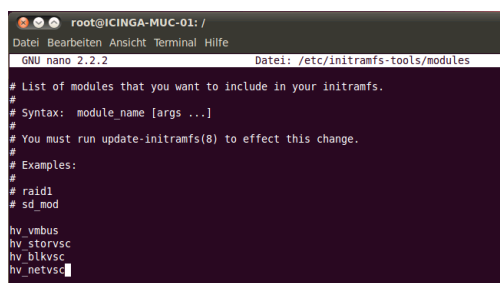
**nano /etc/initramfs-tools/modules**

hv\_vmbus

hv\_storvsc

hv\_blkvsc

hv\_netvsc



Wenn ich die vier Module eingetragen habe, kann ich mit folgendem Tastaturkürzel die Datei Speichern und das Programm **nano** verlassen. STRG + O (Speichern) + Enter, STRG + X (Verlassen von nano)

Nach dem Speichern muss ein neues Boot Image generiert werden, um beim Booten auch die neu aktivierten Module mit zu starten.

### ***update-initramfs -u***

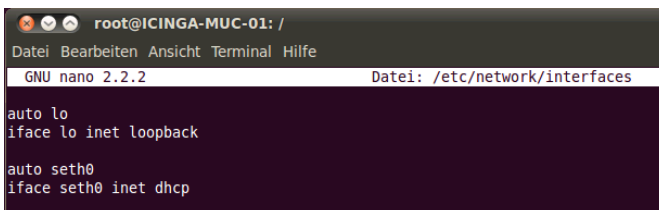
```
root@ICINGA-MUC-01:/# sudo update-initramfs -u
update-initramfs: Generating /boot/initrd.img-2.6.32-21-generic
```

Netzwerkkarte auf DHCP konfigurieren/Server neu starten

### ***nano /etc/network/interfaces***

```
auto seth0
```

```
iface seth0 inet dhcp
```



```
root@ICINGA-MUC-01: /
Datei Bearbeiten Ansicht Terminal Hilfe
GNU nano 2.2.2 Datei: /etc/network/interfaces
auto lo
iface lo inet loopback
auto seth0
iface seth0 inet dhcp
```

Nach dem Neustart schauen wir nach, ob auch alle Module sauber geladen wurden.

### ***lsmod | grep hv\_\****

Wenn ja, dann vergeben wir der neuen Netzwerkkarte eine statische IP Adresse.

### ***sudo -s***

### ***nano /etc/network/interfaces***

```
auto seth0
```

```
iface seth0 inet static
```

```
address 192.168.178.30
```

```
network 192.168.178.0
```

```
netmask 255.255.255.0
```

```
broadcast 192.168.178.255
```

```
gateway 192.168.178.1
```

```
dns-nameservers 192.168.178.1
```

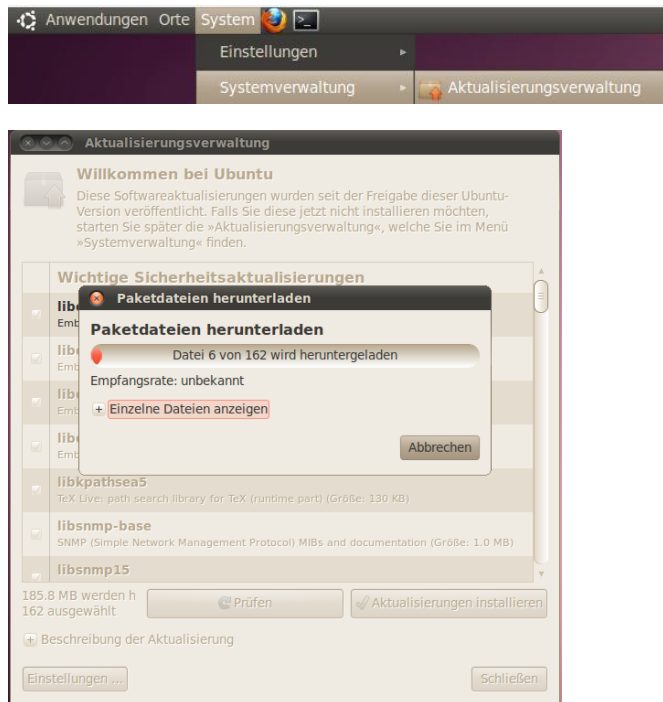
Nach dem nächsten Neustart sollte die Netzwerkkarte sauber konfiguriert sein

### ***ifconfig seth0***



## Ubuntu 10.04 Updates installieren

Updates installiere ich über System --> Systemverwaltung --> Aktualisierungsverwaltung



## Ubuntu 10.04 vorbereiten für die Icinga 1.0.2 Installation

C-Compiler/Linux Header Files Installieren

***apt-get install build-essential linux-headers-generic***

Überflüssige und alte Dateien vom System löschen

***apt-get autoremove*** (Die folgenden Pakete werden ENTFERNT: linux-headers-2.6.32-21 linux-headers-2.6.32-21-generic)

Web-Server und GD-Libraries installieren

***apt-get install apache2 libgd2-xpm-dev***

MySQL Server – lib Treiber Installieren

***apt-get install libjpeg62 libjpeg62-dev libpng12-0 libpng12-dev***

***apt-get install mysql-server mysql-client libdbi0 libdbi0-dev libdbd-mysql***

***apt-get install libmysqlclient15-dev***

Perl Installieren

***apt-get install libconfig-inifiles-perl***

PHP 5 Installieren

***apt-get install php5 php5-mysql***

SNMP installieren

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

SSL Installieren

***apt-get install openssl libcurl4-openssl-dev***

phpmyadmin Installieren

***sudo apt-get install phpmyadmin***

## Benutzer und Gruppen für Icinga 1.0.2 anlegen

***sudo -s***

***/usr/sbin/useradd -m icinga***

***passwd icinga*** (2x Passwort eingeben)

Wenn noch nicht vorhanden dann die icinga Gruppe anlegen

***/usr/sbin/groupadd icinga***

Damit du über das Webinterface Befehle an Icinga senden kannst, legst du noch eine neue Gruppe ***icinga-cmd*** an und fügst den Webbenutzer und den Icinga Benutzer dieser Gruppe hinzu.

***/usr/sbin/groupadd icinga-cmd***

***/usr/sbin/usermod -a -G icinga-cmd icinga***

***/usr/sbin/usermod -a -G icinga-cmd www-data***

## ICINGA 1.0.2 und die IDOUtills kompilieren und installieren

Wechsel in das lokale Source-Verzeichnis von Ubuntu

```
cd /usr/local/src
```

Download von den ICINGA 1.0.2 Quelldateien

```
wget http://sourceforge.net/projects/icinga/files/icinga/1.0.2/icinga-1.0.2.tar.gz/download
```

```
tar xvzf icinga-1.0.2.tar.gz
```

```
cd icinga-1.0.2
```

```
./configure --with-command-group=icinga-cmd --enable-idoutils
```

```
*** Configuration summary for icinga-core 1.0.2 06-30-2010 ***:
General Options:
-----
Icinga executable:  icinga
Icinga user/group:  icinga,icinga
Command user/group: icinga,icinga-cmd
Embedded Perl:     no
Event Broker:      yes
Build IDOUtills:   yes
Install ${prefix}:  /usr/local/icinga
Lock file:         ${prefix}/var/icinga.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory:    /etc/init.d
Apache conf.d directory: /etc/apache2/conf.d
Mail program:      /bin/mail
Host OS:           linux-gnu

Web Interface Options:
-----
HTML URL:  http://localhost/icinga/
CGI URL:   http://localhost/icinga/cgi-bin/
Traceroute (used by WAP):

Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

Jetzt wird der ICINGA-Source-Code kompiliert mit

```
make all
```

Installieren bitte die Binaries, das Init-Script, Beispiel-Konfigurationsdateien, APIs und setze die Berechtigungen für das External-Command-Verzeichnis.

```
make install
```

```
make install-init
```

```
make install-config
```

```
make install-commandmode
```

```
make install-idoutils
```

```
make install-api
```

## Contact.cfg bearbeiten

***nano /usr/local/icinga/etc/objects/contacts.cfg*** (Admin E-Mail Adresse anpassen)

STRG + W für suchen in nano

STRG + O + Enter fürs Speichern

STRG + X fürs beenden von nano

```
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.

define contact{
    contact_name    icingaadmin        ; Short name of user
    use             generic-contact    ; Inherit default values fr
    alias           Icinga Admin       ; Full name of user
    email           email@adresse.de   ; <<***** CHANGE THIS TO YO
}
```

Jetzt ändern wir noch die Beispiel Dateien im Verzeichnis ***/usr/local/icinga/etc***

Mit dem Befehl ***mv*** benennen wir die Dateien um und mit ***cp*** kopieren wir sie nur, und behalten die Originale.

***cd /usr/local/icinga/etc***

***cp idomod.cfg-sample idomod.cfg***

***cp ido2db.cfg-sample ido2db.cfg***

Aktiviere das idomod-Eventbroker-Modul. Editiere die ***/usr/local/icinga/etc/icinga.cfg*** und suche nach den Eintrag "***broker\_module***" und aktiviere die Zeile (entfernen von # vor der Zeile) bzw. fügen die folgende Zeile hinzu wenn nicht vorhanden (***broker\_module=/usr/local/icinga/bin/idomod.o config\_file=/usr/local/icinga/etc/idomod.cfg***).

***nano /usr/local/icinga/etc/icinga.cfg***

STRG + W für suchen

STRG + O (Enter) Speichern

STRG + X fürs beenden von nano

```
#broker_module=/somewhere/module1.o
#broker_module=/somewhere/module2.o arg1 arg2=3 debug=0
# Uncomment this to enable idomod.o
broker_module=/usr/local/icinga/bin/idomod.o config_file=/usr/local/icinga/etc/idomod.cfg
```

## MySQL Datenbank anlegen

Mit folgendem Befehl baut man eine Verbindung zum MySQL Datenbank Server auf

**mysql -u root -p** (Passwort)

**CREATE DATABASE icinga;**

**GRANT USAGE ON \*.\* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga' WITH**

**MAX\_QUERIES\_PER\_HOUR 0 MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0;**

Berechtigungen für den Benutzer icinga setzen

**GRANT SELECT , INSERT , UPDATE , DELETE ON icinga.\* TO 'icinga'@'localhost';**

**FLUSH PRIVILEGES;**

**quit;**

So sieht es aus wenn man die Befehle in der Konsole eingibt

```
root@icinga-muc-01:/usr/local/icinga/etc# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 5.1.41-3ubuntu12.3 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE icinga;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT USAGE ON *.* TO 'icinga'@'localhost' IDENTIFIED BY 'icinga' WITH
-> MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT , INSERT , UPDATE , DELETE ON icinga.* TO 'icinga'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

Datenbankschema der Datenbank icinga hinzufügen

**cd /usr/local/src/icinga-1.0.2/module/ido2db/mysql**

**mysql -u root -p icinga < mysql.sql** (Passwort eingeben)

Informationen in der ido2db.cfg überprüfen und ggf. anpassen

**nano /usr/local/icinga/etc/ido2db.cfg**

db\_servertype=mysql

db\_port=3306

db\_user=icinga

db\_pass=icinga

## Icinga 1.0.2 Webkomponenten installieren

Installiere bitte die ICINGA-Web-Konfigurationsdatei ins Apache conf.d-Verzeichnis.

```
cd /usr/local/src/icinga-1.0.2/
```

```
make install-webconf
```

Installiere und konfiguriere das klassische Web-Interface

Icinga stellt das klassische Web-Interface zur Verfügung ("Classic Web", "die CGIs").

```
cd /usr/local/src/icinga-1.0.2/
```

```
make cgis
```

```
make install-cgis
```

Web Admin User anlegen und Passwort vergeben

```
htpasswd -c /usr/local/icinga/etc/htpasswd.users icingaadmin (2x neues Passwort eingeben)
```

Zusätzlichen neuen Benutzer hinzufügen

```
htpasswd /usr/local/icinga/etc/htpasswd.users <USERNAME> (2x neues Passwort)
```

Apache Server neu starten

```
/etc/init.d/apache2 restart
```

Nach dem neu starten vom Apache Server, bekommt man immer eine Hinweismeldung angezeigt die

So lautet:

```
* Reloading web server config apache2
```

```
apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for
```

```
ServerName
```

Diesen Schönheitsfehler lösen wir in dem wir die **/etc/hosts** öffnen und zwei Einträge abändern.

```
nano /etc/hosts
```

```
127.0.1.1 _TAB_ Server.domainname.de
```

Statische IP Adresse vom Server \_TAB\_ Server.domainname.de

```
127.0.0.1    localhost
127.0.1.1    icinga-muc-01.fritz.box
192.168.178.30 icinga-muc-01.fritz.box
```

## Nagios Plugins 1.4.15 installieren

Wir wechseln wieder in das Source Verzeichnis

```
cd /usr/local/src
```

Laden die Quelldateien vom Server

```
wget http://surfnet.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz
```

Extrahieren die Quelldateien

```
tar xvf nagios-plugins-1.4.15.tar.gz
```

Wechseln in das Verzeichnis der Quelldateien

```
cd nagios-plugins-1.4.15
```

Führen das Konfigurations- Script aus

```
./configure --prefix=/usr/local/icinga --with-nagios-user=icinga
```

Kompilieren das Ganze und installieren im Anschluss die Plugins

```
make
```

```
make install
```

Nach der Installation der Plugins starten wir IDOUtils (IDOUtils muss immer vor Icinga gestartet werden)

```
/etc/init.d/ido2db start → update-rc.d ido2db defaults
```

## ICINGA 1.0.2 Konfiguration überprüfen

```
/usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Anstatt die Pfade für das Binary und die Konfigurationsdatei anzugeben kannst du auch folgenden Befehl eingeben:

```
/etc/init.d/icinga checkconfig
```

Die Ausführung ergibt einen Return-Code ungleich Null, wenn die Konfiguration Fehler enthält. Das kann sinnvoll sein, wenn man Icinga automatisch starten will.

Wenn man folgenden Befehl eingibt

```
/etc/init.d/icinga show-errors
```



dann wird eine Datei mit den gefundenen Fehlern erstellt. Falls es Fehler gibt, wird der Inhalt der Datei angezeigt ("show-errors" ab Icinga 1.0.2).

ICINGA Dienst/Deamon starten

**/etc/init.d/icinga start**

Fügen ICINGA zu der Liste der System-Services hinzu und sorgen somit für einen automatischen Start, wenn das System hochfährt.

**update-rc.d icinga defaults**

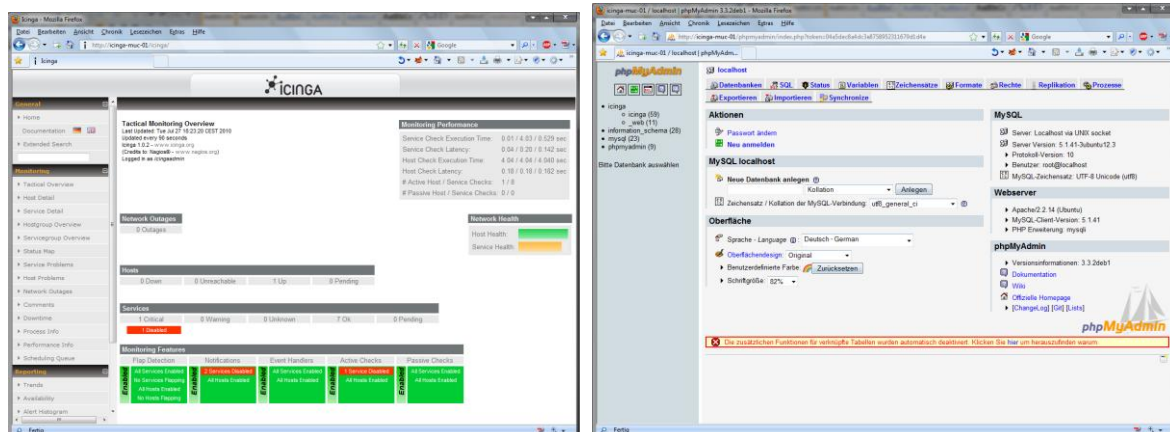
Stelle sicher, dass die Firewall-Einstellungen deiner Maschine einen Zugriff auf das Web-Interface ermöglichen, wenn du von anderen Rechnern darauf zugreifen willst.

**iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT**

Fertig - Herzlichen Glückwunsch! ICINGA 1.0.2 + phpmyadmin wurde erfolgreich installiert.

<http://servername/icinga>

<http://servername/phpmyadmin>





## Installation des neuen Icinga 1.0.1 Web Frontend

```
apt-get install php5-xmllrpc php5-xsl php5-ldap php5-gd
```

```
apt-get install git-core
```

```
cd /usr/local/src/
```

```
git clone git://git.icinga.org/icinga-web.git
```

Wenn der Service nicht verfügbar ist, funktioniert es immer noch mit folgendem Befehl

```
wget http://sourceforge.net/projects/icinga/files/icinga-web/1.0.1/icinga-web-1.0.1.tar.gz/download
```

Extrahiere das Archiv wie folgt:

```
tar xzvf icinga-web-1.0.1.tar.gz
```

Wechsel in das Verzeichnis:

```
cd icinga-web-1.0.1
```

Icinga Web stellt verschiedene configure Optionen zur Verfügung z.B.

```
./configure --prefix=/usr/local/icinga-web --with-web-user=www-data --with-web-group=www-data --with-db-type=mysql --with-db-host=localhost --with-db-port=3306 --with-db-name=icinga_web --with-db-user=icinga_web --with-db-pass=icinga_web --with-icinga-api=/usr/local/icinga/share/icinga-api
```

das Icinga Webinterface mit: --> **make install** --> unter /usr/local/icinga-web installiert.

PHP-Abhängigkeiten

Bitte prüfe die PHP-Abhängigkeiten mit:

```
make testdeps
```

```
PHP extensions
3/20 Test php5-xsl (REQUIRED) ... (xsl found v0.1) OK
4/20 Test php5-ldap ... (ldap found) OK
5/20 Test php5-pdo (REQUIRED) ... (PDO found v1.0.4dev) OK
6/20 Test php5-dom (REQUIRED) ... (dom found v20031129) OK
7/20 Test php5-session (REQUIRED) ... (session found) OK
8/20 Test php5-spl (REQUIRED) ... (SPL found v0.2) OK
9/20 Test php5-pcre (REQUIRED) ... (pcre found) OK
10/20 Test php5-tokenizer (REQUIRED) ... (tokenizer found v0.1) OK
11/20 Test php5-libxml (REQUIRED) ... (libxml found) OK
12/20 Test php5-reflection (REQUIRED) ... (Reflection found v$Revision: 293036 $) OK
13/20 Test php5-gettext (REQUIRED) ... (gettext found) OK

Optional pdo drivers
14/20 Test php5-pdo-mysql ... (pdo mysql found v1.0.2) OK
15/20 Test php5-pdo-pgsql ... (Extension pdo_pgsql does not exist) FAIL

Optional php extension
16/20 Test php5-soap ... (soap found) OK
17/20 Test php5-xmllrpc ... (xmllrpc found v0.51) OK
18/20 Test php5-iconv ... (iconv found) OK
19/20 Test php5-gd ... (gd found) OK
20/20 Test php.ini memory_limit ... (memory_limit='1') FAIL

All over result: PASS (required 12/12, optional 6/8, all 18/20, time 0.00s)
Exit (status=0)
```

Die "magic\_quote\_gpc" in der Apache- und der CLI-php.ini auf "Off" setzen!

**nano /etc/php5/apache2/php.ini**

STRG + W magic\_quotes\_gpc = suchen

```
; Production Value: Off
; http://php.net/magic-quotes-gpc
magic_quotes_gpc = Off
```

**nano /etc/php5/cli/php.ini**

STRG + W magic\_quotes\_gpc = suchen

STRG + W memory\_limit suchen und Wert auf 128M setzen

STRG + O fürs Speichern

STRG + X fürs Schließen der Datei

```
; Development Value: Off
; Production Value: Off
; http://php.net/magic-quotes-gpc
magic_quotes_gpc = Off
```

```
;max_input_nesting_level = 64
; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 128M
;
; Error handling and logging ;
;
```

**/etc/init.d/apache2 restart**

**cd /usr/local/src/icinga-web**

nochmal **make testdeps** ausführen

```
PHP (core) tests
1/20 Test php version >= 5.2.3 (REQUIRED) ... (version 5.3.2-1ubuntu4.2 >= 5.2.3) OK
2/20 Test for PEAR (REQUIRED) ... OK

PHP extensions
3/20 Test php5-xsl (REQUIRED) ... (xsl found v0.1) OK
4/20 Test php5-ldap ... (ldap found) OK
5/20 Test php5-pdo (REQUIRED) ... (pdo found v1.0.4dev) OK
6/20 Test php5-dom (REQUIRED) ... (dom found v20031129) OK
7/20 Test php5-session (REQUIRED) ... (session found) OK
8/20 Test php5-spl (REQUIRED) ... (spl found v0.2) OK
9/20 Test php5-pcre (REQUIRED) ... (pcre found) OK
10/20 Test php5-tokenizer (REQUIRED) ... (tokenizer found v0.1) OK
11/20 Test php5-libxml (REQUIRED) ... (libxml found) OK
12/20 Test php5-reflection (REQUIRED) ... (Reflection found vsRevision: 293036 $) OK
13/20 Test php5-gettext (REQUIRED) ... (gettext found) OK

Optional pdo drivers
14/20 Test php5-pdo-mysql ... (pdo mysql found v1.0.2) OK
15/20 Test php5-pdo-pgsql ... (Extension pdo_pgsql does not exist) FAIL

Optional php extension
16/20 Test php5-soap ... (soap found) OK
17/20 Test php5-xmlrpc ... (xmlrpc found v0.51) OK
18/20 Test php5-iconv ... (iconv found) OK
19/20 Test php5-gd ... (gd found) OK
20/20 Test php.ini memory limit ... (memory limit='134217728') OK

All over result: PASS (required 12/12, optional 7/8, all 19/20, time 0.00s)
Exit (status=0)
```

Jetzt sollte es passen.

## Datenbank-Installation

Icinga Web benötigt eine eigene Datenbank z.B. icinga\_web. Du kannst theoretisch die Datenbank von IDOUtils mitverwenden, dies wird allerdings nicht empfohlen, um bei Upgrades keine Probleme zu haben.

Wir logen uns auf den MySQL Server ein, vergeben die Berechtigungen für den icinga\_web Benutzer

***icinga\_user (Bsp: icinga\_web)***

***icinga\_password (Bsp: icinga\_web)***

***mysql -u root -p*** (Passwort)

***GRANT USAGE ON icinga\_web.\* TO 'icinga\_user'@'localhost' IDENTIFIED BY 'icinga\_password' WITH MAX\_QUERIES\_PER\_HOUR 0 MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0;***

***GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, INDEX ON icinga\_web.\* TO 'icinga\_user'@'localhost';***

***FLUSH PRIVILEGES;***

***quit;***

## Anlegen der Datenbank

Icinga Web bringt Doctrine mit, womit in Zukunft die Datenbank verwaltet wird. Mittels 'make' kann diese initialisiert, upgegradet oder gelöscht werden.

***make db-initialize*** - legt eine Icinga web Datenbank an und füllt diese mit initialen Werten

***make db-drop*** - löscht die komplette Datenbank, inklusive Securityabfrage um ungewolltes Löschen zu verhindern

***make db-upgrade*** - führt ein Upgrade von einer alten Version durch (ähnlich wie db-initialize)

***make db-doc2sql*** - falls Sie SQL Scripts beispielsweise für Packaging oder manuelles Setup benötigen, können Sie diese Option verwenden um die SQL scripts aus Doctrine zu extrahieren

Um die Datenbankanlage-Befehle nutzen zu können, müssen Sie dem Benutzer der diese DB Befehle ausführen soll, auch die entsprechenden Privilegien zuordnen. Falls derjenige Benutzer, der für das Icinga Web in configure vorgesehen wurde, zu wenig Berechtigungen hat, gibt es in 'make' eine Abfrage, ob Sie einen anderen Benutzer mit mehr Privilegien verwenden wollen. Falls dies nicht funktioniert, sollten sie etc/build.properties editieren und einen root Benutzer einfügen.

Das Installieren der Datenbank funktioniert mittels:

***make db-initialize***

## Icinga-Web Konfiguration

Im Normalfall können die Datenbankeinstellungen während des „**configure**“ vornehmen werden. Sollten man diese allerdings anpassen wollen oder im Fehlerfall überprüfen wollen, öffnen wir folgende Datei.

(./configure ..... --with-db-type=mysql --with-db-host=localhost --with-db-port=3306 --with-db-name=icinga\_DB --with-db-user=icinga\_user --with-db-pass=icinga\_password .....)

**nano /usr/local/src/icinga-web-1.0.1/app/config/databases.xml**

```
<databases default="icinga_web">
  <database name="icinga_web" class="AgaviDoctrineDatabase">
    <!-- Doctrine dsn string: http://www.doctrine-project.org/documentation/manual/1\_1/en/introduction-to-connection -->
    <ae:parameter name="dsn">mysql://icinga_user:icinga_password@127.0.0.1:3306/icinga_DB</ae:parameter>
    <!-- Generic credentials -->
    <!-- <ae:parameter name="username">icinga_web</ae:parameter> -->
    <!-- <ae:parameter name="password">icinga_web</ae:parameter> -->
    <!-- DB encoding type -->
    <ae:parameter name="charset">utf8</ae:parameter>
    <!-- Doctrine_Manager configuration -->
    <ae:parameter name="manager_attributes">
    <!-- This allows lazy loading of the models -->
    <ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
    </ae:parameter>
    <!-- The path to our models -->
    <ae:parameter
name="load_models">%core.module_dir%/AppKit/lib/database/models/generated</ae:parameter>
    <ae:parameter name="models_directory">%core.module_dir%/AppKit/lib/database/models</ae:parameter>
  </database>
</databases>
```

## Apache-Konfiguration

Voraussetzungen:

**mod\_rewrite**, vielleicht muss ein Verweis erstellt werden:

**In -s /etc/apache2/mods-available/rewrite.load /etc/apache2/mods-enabled/rewrite.load**

**sudo a2enmod headers** → [http://httpd.apache.org/docs/2.0/mod/mod\\_headers.html](http://httpd.apache.org/docs/2.0/mod/mod_headers.html)

**sudo a2enmod rewrite** → [http://httpd.apache.org/docs/2.0/de/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/2.0/de/mod/mod_rewrite.html)

.htaccess editieren

***nano /usr/local/icinga-web/pub/.htaccess***

DirectoryIndex index.php

Options -MultiViews -Indexes +FollowSymLinks  
Order allow,deny  
Allow from all

```
<IfModule mod_rewrite.c>
RewriteEngine On
```

```
# This depends on your path
# on independent hosts the base is '/'
```

```
RewriteBase /icinga-web/
```

```
# If the requested URL does not exist (it's likely an agavi route),
# pass it as path info to index.php, the Agavi dispatch script.
```

```
RewriteRule ^$ index.php?/ [QSA,L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule (.*) index.php?/$1 [QSA,L]
```

```
</IfModule>
```

```
<IfModule mod_deflate.c>
SetOutputFilter DEFLATE
```

```
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4.[0-678] no-gzip
```

```
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
BrowserMatch \bMSI[E] !no-gzip !gzip-only-text/html
```

```
Header append Vary User-Agent env=!dont-vary
```

```
</IfModule>
```

```
<IfDefine APACHE2>
```

```
AcceptPathInfo On
```

```
</IfDefine>
```

```
#<IfModule mod_auth_basic.c>
```

```
# AuthType Basic
# AuthName "My http basic auth realm"
# AuthUserFile /path/to/my/htusers
# require valid-user
#</IfModule>
```

Bitte erstelle noch einen neuen Alias im Konfigurationsverzeichnis des Webserver (hier in der icinga-web.conf, wenn nicht vorhanden dann bitte anlegen):

***touch icinga-web.conf***

***nano /etc/apache2/conf.d/icinga-web.conf***

Alias /icinga-web /usr/local/icinga-web/pub

```
<directory /usr/local/icinga-web/pub>
AllowOverride All
</directory>
```

Leere den Cache mit folgendem Befehl:

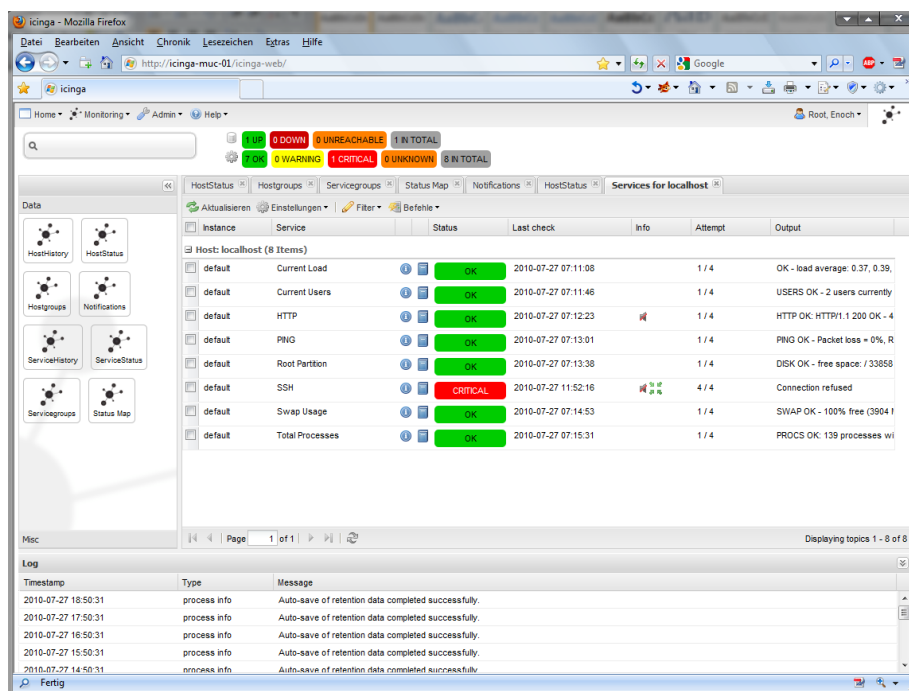
```
rm /usr/local/icinga-web/app/cache/config/*.php
```

und starte den Webserver neu:

```
/etc/init.d/apache2 restart
```

Öffne den Webbrowser <http://servername/icinga-web>.

Überprüfe ob das Webinterface ohne Fehler startet, und logge dich mit folgendem User 'root' und Passwort 'password' ein.



## ICINGA Grundkonfiguration

Es gibt viele Möglichkeiten ICINGA/NAGIOS richtig zu konfigurieren, hier möchte ich euch meine Lösung vorschlagen, wie ihr schnell und einfach eine kleine Umgebung in eurem Netzwerk überwachen könnt. Umso komplexer die Umgebung umso mehr Überlegung muss investiert werden damit ICINGA/NAGIOS sauber konfiguriert wird.

Sehr gute Anregung → [http://www.nagios-wiki.de/nagios/tips/nagios\\_optimal\\_konfigurieren](http://www.nagios-wiki.de/nagios/tips/nagios_optimal_konfigurieren)

Oder das Buch von Wolfgang Barth → [http://www.amazon.de/Nagios-System-Netzwerkmonitoring-Wolfgang-Barth/dp/3937514910/ref=sr\\_1\\_2?ie=UTF8&s=books&qid=1273768468&sr=8-2](http://www.amazon.de/Nagios-System-Netzwerkmonitoring-Wolfgang-Barth/dp/3937514910/ref=sr_1_2?ie=UTF8&s=books&qid=1273768468&sr=8-2)

## Ordnerstruktur anlegen

Aber nun zu meiner Lösung die ich für mich jetzt am besten gefunden habe und auch heute noch einsetze. Ich lege im Verzeichnis **/usr/local/icinga/etc** einen neuen Ordner mit meinen Firmennamen (**company**) an und benenne den alten Ordner **objects** um in **objects\_old**.

```
sudo -s (Password)
su - icinga
bash
cd /usr/local/icinga/etc
mkdir company
cd company
```

In diesen Ordner **company** lege ich weitere Unterordner mit meinen Standorten (**standort1**, **2** und **3**) an.

```
mkdir standort1 standort2 standort3
```

Kopiere mir folgende Dateien (**commands.cfg**, **contacts.cfg**, **templates.cfg**, **timeperiods.cfg**) vom Original Ordner **/usr/local/icinga/etc/objects\_old** in das Verzeichnis **/usr/local/icinga/etc/company**



## hostgroup.cfg

Lege in meinen **company** Ordner eine neue Datei, mit dem Namen **hostgroup.cfg** an und fülle diese Datei mit folgendem Inhalt.

```
cd /usr/local/icinga/etc/company
touch hostgroup.cfg
nano hostgroup.cfg
```

Der erste Eintrag beinhaltet meine erste Hostgruppe mit dem Firmennamen (mustermann), der Alias ist frei wählbar (z.B. Firma Mustermann Inc.), bei den Members möchte ich alle Server der Firma zusammenfassen, das wird mit dem Stern\* makiert, bei den Eintrag **hostgroup\_members** verschachtel ich alle anderen Hostgruppen der einzelnen Standorte (location1, 2 und 3), Der Rest der Datei denke ich ist Selbsterklärend.







```
#####
# HOST GROUP Firma Mustermann Inc.
#####
define hostgroup{
    hostgroup_name      mustermann
    alias               Firma Mustermann Inc.
    members             *
    hostgroup_members   location1,location2,location3
}
#####
# HOST GROUP Standort 1
#####
define hostgroup{
    hostgroup_name      location1
    alias               Server Standort 1
    members             icinga-muc-01.fritz.box,server2.fritz.box,server3.fritz.box,usw. (Bitte ersetzen)
}
#####
# HOST GROUP Standort 2
#####
define hostgroup{
    hostgroup_name      location2
    alias               Server Standort 2
    members             tnetd.fritz.box (Bitte ersetzen)
}
#####
# HOST GROUP Router
#####
define hostgroup{
    hostgroup_name      location3
    alias               Router
    members             fritz.box
}
#####
```









Würde dann in der ICINGA Oberfläche wie folgt aussehen

ty for All Host Groups  
r All Host Groups

Service Overview For All Host Groups

Server Standort 1 (location1)				Server Standort 2 (location2)				Router (location3)			
Host	Status	Services	Actions	Host	Status	Services	Actions	Host	Status	Services	Actions
icinga.fritz.box	UP	11 OK	 	tetd.fritz.box	UP	4 OK	 	fritz.box	UP	1 OK	 

Firma Mustermann Inc. (mustermann)			
Host	Status	Services	Actions
fritz.box	UP	1 OK	 
icinga.fritz.box	UP	11 OK	 
tetd.fritz.box	UP	4 OK	 

## localhost.cfg

Im nächste Schritt kopiere ich die Original Datei **localhost.cfg** in meinen Ordner standort1, 2 und 3. Nach dem kopieren benenne ich die Datei so um, wie mein Server heißt. (Bitte vergesst die Endung der Datei .cfg nicht, alle Dateien die nicht so enden ignoriert ICINGA)

```
cp /usr/local/icinga/etc/objects/localhost.cfg /usr/local/icinga/etc/company/standort1
cp /usr/local/icinga/etc/objects/localhost.cfg /usr/local/icinga/etc/company/standort2
cp /usr/local/icinga/etc/objects/localhost.cfg /usr/local/icinga/etc/company/standort3
```

```
mv /usr/local/icinga/etc/company/standort1/localhost.cfg
/usr/local/icinga/etc/company/standort1/icinga-muc-01.fritz.box.cfg
```

```
mv /usr/local/icinga/etc/company/standort2/localhost.cfg
/usr/local/icinga/etc/company/standort2/tetd.fritz.box.cfg
```

```
mv /usr/local/icinga/etc/company/standort3/localhost.cfg
/usr/local/icinga/etc/company/standort3/fritz.box.cfg
```

## templates.cfg

Wenn das erledigt ist dann Konfiguriere ich meine templates.cfg Datei

Nähere Informationen findet Ihr hier: <http://docs.icinga.org/latest/de/objectdefinitions.html>

```
#####
# CONTACT TEMPLATES
#####
define contact{
    name                generic-contact                ; The name of this contact template
    service_notification_period 24x7                ; service notifications can be sent anytime
    host_notification_period 24x7                ; host notifications can be sent anytime
    service_notification_options w,u,c,r,f,s        ; send notifications for all service states, flapping events, and
    host_notification_options d,u,r,f,s            ; send notifications for all host states, flapping events, and scheduled
    service_notification_commands notify-service-by-email ; send service notifications via email
}
```

```

host_notification_commands    notify-host-by-email    ; send host notifications via email
register                      0    ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL CONTACT
    }
#####
# LINUX HOST TEMPLATES
#####
define host{
name                linux-server    ; The name of this host template
notifications_enabled    1    ; Host notifications are enabled
event_handler_enabled    1    ; Host event handler is enabled
flap_detection_enabled    1    ; Flap detection is enabled
failure_prediction_enabled    1    ; Failure prediction is enabled
process_perf_data    1    ; Process performance data
retain_status_information    1    ; Retain status information across program restarts
retain_nonstatus_information    1    ; Retain non-status information across program restarts
check_period    24x7    ; By default, Linux hosts are checked round the clock
check_interval    5    ; Actively check the host every 5 minutes
retry_interval    1    ; Schedule host check retries at 1 minute intervals
max_check_attempts    5    ; Check each Linux host 10 times (max)
check_command    check-host-alive    ; Default command to check Linux hosts
notification_period    24x7    ; Linux admins hate to be woken up, so we only notify during the day
notification_interval    30    ; Resend notifications every 2 hours
notification_options    d,u,r    ; Only send notifications for specific host states
contact_groups    admins    ; Notifications get sent to the admins by default
register    0    ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
    }
#####
# WINDOWS HOST TEMPLATES
#####
define host{
name                windows-server    ; The name of this host template
notifications_enabled    1    ; Host notifications are enabled
event_handler_enabled    1    ; Host event handler is enabled
flap_detection_enabled    1    ; Flap detection is enabled
failure_prediction_enabled    1    ; Failure prediction is enabled
process_perf_data    1    ; Process performance data
retain_status_information    1    ; Retain status information across program restarts
retain_nonstatus_information    1    ; Retain non-status information across program restarts
check_period    24x7    ; By default, Linux hosts are checked round the clock
check_interval    5    ; Actively check the host every 5 minutes
retry_interval    1    ; Schedule host check retries at 1 minute intervals
max_check_attempts    5    ; Check each Linux host 10 times (max)
check_command    check-host-alive    ; Default command to check Linux hosts
notification_period    24x7    ; Linux admins hate to be woken up, so we only notify during the day
notification_interval    30    ; Resend notifications every 2 hours
notification_options    d,u,r    ; Only send notifications for specific host states
contact_groups    admins    ; Notifications get sent to the admins by default
register    0    ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
    }
#####
# ROUTER HOST TEMPLATES
#####
define host{
name                router-host    ; The name of this host template
notifications_enabled    1    ; Host notifications are enabled
event_handler_enabled    1    ; Host event handler is enabled
flap_detection_enabled    1    ; Flap detection is enabled
failure_prediction_enabled    1    ; Failure prediction is enabled
process_perf_data    1    ; Process performance data
retain_status_information    1    ; Retain status information across program restarts
retain_nonstatus_information    1    ; Retain non-status information across program restarts
check_period    24x7    ; By default, Linux hosts are checked round the clock

```

```

check_interval      5                ; Actively check the host every 5 minutes
retry_interval      1                ; Schedule host check retries at 1 minute intervals
max_check_attempts  5                ; Check each Linux host 10 times (max)
check_command       check-host-alive ; Default command to check Linux hosts
notification_period  24x7            ; Linux admins hate to be woken up, so we only notify during the day
notification_interval 30            ; Resend notifications every 2 hours
notification_options d,u,r          ; Only send notifications for specific host states
contact_groups      admins          ; Notifications get sent to the admins by default
register            0                ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
}

#####
# WINDOWS SERVICE TEMPLATES
#####
define service{
name                windows-service ; The 'name' of this service template
active_checks_enabled 1            ; Active service checks are enabled
passive_checks_enabled 1          ; Passive service checks are enabled/accepted
parallelize_check     1            ; Active service checks should be parallelized (disabling this can lead to
obsess_over_service   1            ; We should obsess over this service (if necessary)
check_freshness       0            ; Default is to NOT check service 'freshness'
notifications_enabled 1            ; Service notifications are enabled
event_handler_enabled 1            ; Service event handler is enabled
flap_detection_enabled 1           ; Flap detection is enabled
failure_prediction_enabled 1       ; Failure prediction is enabled
process_perf_data     1            ; Process performance data
retain_status_information 1        ; Retain status information across program restarts
retain_nonstatus_information 1     ; Retain non-status information across program restarts
is_volatile           0            ; The service is not volatile
check_period          24x7        ; The service can be checked at any time of the day
check_interval        5            ; Actively check the host every 5 minutes
retry_interval        1            ; Schedule host check retries at 1 minute intervals
max_check_attempts    5            ; Check each Linux host 10 times (max)
contact_groups        admins       ; Notifications get sent out to everyone in the 'admins' group
notification_options   c,r         ; Send notifications about warning, unknown, critical, and recovery events
notification_interval  20          ; Re-notify about service problems every hour
notification_period    24x7        ; Notifications can be sent out at any time
register              0            ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
}

#####
# LINUX SERVICE TEMPLATES
#####
define service{
name                linux-service ; The name of this service template
active_checks_enabled 1            ; Active service checks are enabled
passive_checks_enabled 1          ; Passive service checks are enabled/accepted
parallelize_check     1            ; Active service checks should be parallelized (disabling this can lead to
obsess_over_service   1            ; We should obsess over this service (if necessary)
check_freshness       0            ; Default is to NOT check service 'freshness'
notifications_enabled 1            ; Service notifications are enabled
event_handler_enabled 1            ; Service event handler is enabled
flap_detection_enabled 1           ; Flap detection is enabled
failure_prediction_enabled 1       ; Failure prediction is enabled
process_perf_data     1            ; Process performance data
retain_status_information 1        ; Retain status information across program restarts
retain_nonstatus_information 1     ; Retain non-status information across program restarts
is_volatile           0            ; The service is not volatile
check_period          24x7        ; The service can be checked at any time of the day
check_interval        5            ; Actively check the host every 5 minutes
retry_interval        1            ; Schedule host check retries at 1 minute intervals
max_check_attempts    5            ; Check each Linux host 10 times (max)
contact_groups        admins       ; Notifications get sent out to everyone in the 'admins' group

```

```
notification_options      c,r          ; Send notifications about warning, unknown, critical, and recovery events
notification_interval      20           ; Re-notify about service problems every hour
notification_period        24x7         ; Notifications can be sent out at any time
register                   0           ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
    }

#####
# ROUTER SERVICE TEMPLATES
#####
define service{
    name                router-service    ; The name of this service template
    active_checks_enabled 1              ; Active service checks are enabled
    passive_checks_enabled 1             ; Passive service checks are enabled/accepted
    parallelize_check      1             ; Active service checks should be parallelized (disabling this can lead to
    obsess_over_service    1             ; We should obsess over this service (if necessary)
    check_freshness        0             ; Default is to NOT check service 'freshness'
    notifications_enabled  1             ; Service notifications are enabled
    event_handler_enabled  1             ; Service event handler is enabled
    flap_detection_enabled  1             ; Flap detection is enabled
    failure_prediction_enabled 1         ; Failure prediction is enabled
    process_perf_data      1             ; Process performance data
    retain_status_information 1           ; Retain status information across program restarts
    retain_nonstatus_information 1        ; Retain non-status information across program restarts
    is_volatile            0             ; The service is not volatile
    check_period            24x7         ; The service can be checked at any time of the day
    check_interval          5            ; Actively check the host every 5 minutes
    retry_interval          1            ; Schedule host check retries at 1 minute intervals
    max_check_attempts      5            ; Check each Linux host 10 times (max)
    contact_groups          admins       ; Notifications get sent out to everyone in the 'admins' group
    notification_options      c,r        ; Send notifications about warning, unknown, critical, and recovery events
    notification_interval      20        ; Re-notify about service problems every hour
    notification_period        24x7      ; Notifications can be sent out at any time
    register                 0           ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!
    }

#####
```

## icinga.cfg

Als nächstes gebe ich noch in der Hauptkonfigurations-Datei **icinga.cfg** an, in welchen Ordner ICINGA meine neuen Konfigurationsdateien findet.

<http://docs.icinga.org/latest/de/configmain.html>

Ich öffne als root die **icinga.cfg** und füge folgenden Eintrag hinzu.

```
sudo nano /usr/local/icinga/etc/icinga.cfg
cfg_dir=/usr/local/icinga/etc/company
```

alle anderen CFG Dateien die in den Ordner **/usr/local/icinga/etc/objects** verweisen. Kommentiere ich mit einem # aus.

```
# You can specify individual object config files as shown below:
#cfg_file=/usr/local/icinga/etc/objects/commands.cfg
#cfg_file=/usr/local/icinga/etc/objects/contacts.cfg
#cfg_file=/usr/local/icinga/etc/objects/timeperiods.cfg
#cfg_file=/usr/local/icinga/etc/objects/templates.cfg

# Definitions for monitoring the local (Linux) host
#cfg_file=/usr/local/icinga/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/icinga/etc/objects/windows.cfg

# Definitions for monitoring a router/switch
#cfg_file=/usr/local/icinga/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/icinga/etc/objects/printer.cfg

cfg_dir=/usr/local/icinga/etc/company
```

Und da ich gerne das Europäische Datumsformat lesen möchte, ändere ich noch den Eintrag **date\_format** von **us** (Standard) in **euro** um.

```
# DATE FORMAT OPTION
# This option determines how short dates are displayed. Va
# include:
#   us          (MM-DD-YYYY HH:MM:SS)
#   euro        (DD-MM-YYYY HH:MM:SS)
#   iso8601     (YYYY-MM-DD HH:MM:SS)
#   strict-iso8601 (YYYY-MM-DDTHH:MM:SS)
#
date_format=euro_
```

## timeperiods.cfg

Bei der Datei `timeperiods.cfg` sollte man unbedingt die Feiertage von seiner Region anpassen, diese Einstellung sind wichtig damit ICINGA weiß wann es den Administratoren genehm ist, dass man Sie benachrichtigt. 😊

```
#####
# Some U.S. holidays
# Note: The timeranges for each holiday are meant to *exclude* the holidays from being
# treated as a valid time for notifications, etc. You probably don't want your pager
# going off on New Year's. Although you're employer might... :-)

define timeperiod{
    name                us-holidays
    timeperiod_name     us-holidays
    alias               U.S. Holidays

    january 1          00:00-00:00 ; New Years
    monday -1 may      00:00-00:00 ; Memorial Day (last Monday in May)
    july 4              00:00-00:00 ; Independence Day
    monday 1 september 00:00-00:00 ; Labor Day (first Monday in September)
    thursday -1 november 00:00-00:00 ; Thanksgiving (last Thursday in November)
    december 25         00:00-00:00 ; Christmas
}

#####
# This defines a modified "24x7" timeperiod that covers every day of the
# year, except for U.S. holidays (defined in the timeperiod above).
```

```
define timeperiod{
timeperiod_name 24x7_sans_holidays
    alias          24x7 Sans Holidays
    use            us-holidays          ; Get holiday exceptions from other timeperiod
    sunday         00:00-24:00
    monday         00:00-24:00
    tuesday        00:00-24:00
    wednesday      00:00-24:00
    thursday       00:00-24:00
    friday         00:00-24:00
    saturday       00:00-24:00
}
```

```
#####
```

## icinga.fritz.box.cfg

Um die Grundkonfiguration langsam aber sicher abschließen zu können, zeige ich noch wie ich meine beiden Clients definiert habe, um ICINGA anzuweisen bestimmte Services zu überprüfen.

Als erstes kommt der ICINGA Server selbst an die Reihe

```
#####
```

```
# HOST DEFINITION
```

```
#####
```

```
define host{
use            linux-server          ; Host Template (templates.cfg)
host_name     icinga-muc-01.fritz.box ; FQDN - Servername
alias         ICINGA IT Monitoring Server ; Frei definierbarer Anzeigename
address       192.168.178.30          ; Statische IP Adresse vom Server
hostgroups    location1              ; Hostgruppenname (hostgroup.cfg)
parents       fritz.box              ; Unmittelbarer Nachbar (Physisikalischer Switch/Router)
}
```

```
#####
```

```
# SERVICE DEFINITION
```

```
#####
```

```
# Ping Local Machine
```

```
define service{
use            linux-service          ; Host Template (templates.cfg)
host_name     icinga-muc-01.fritz.box ; FQDN - Servername
service_description PING              ; Frei definierbarer Anzeigename
check_command  check_ping!100.0,20%!500.0,60% ; Check_Command (commands.cfg)
}
```

```
#####
```

```
# Define a service to check the number of currently logged in
# users on the local machine.
```

```
define service{
use            linux-service
host_name     icinga-muc-01.fritz.box
service_description angemeldete Benutzer
check_command  check_local_users!10!20
}
```

```
#####
```

```
# Check the disk space of the boot partition

define service{
    use                linux-service
    host_name          icinga-muc-01.fritz.box
    service_description _Boot Partition
    check_command       check_local_disk!20%!10%!/boot
}
#####
# Define a service to check the swap usage the local machine.

define service{
    use                linux-service
    host_name          icinga-muc-01.fritz.box
    service_description _Swap Partition
    check_command       check_local_swap!20!10
}
#####
# Check the disk space of the root partition

define service{
    use                linux-service
    host_name          icinga-muc-01.fritz.box
    service_description _Root Partition
    check_command       check_local_disk!20%!10%!/
}
#####
# Check the disk space of the tmp partition

define service{
    use                linux-service
    host_name          icinga-muc-01.fritz.box
    service_description _tmp Partition
    check_command       check_local_disk!20%!10%!/tmp
}
#####
# Check the disk space of the var partition

define service{
    use                linux-service
    host_name          icinga-muc-01.fritz.box
    service_description _var Partition
    check_command       check_local_disk!20%!10%!/var
}
#####
# Check the disk space of the home partition

define service{
    use                linux-service
    host_name          icinga-muc-01.fritz.box
    service_description _home Partition
    check_command       check_local_disk!20%!10%!/home
}
#####
```

```
# Define a service to check the number of currently running procs

define service{
  use                linux-service
  host_name          icinga-muc-01.fritz.box
  service_description Alle Prozesse
  check_command       check_local_procs!250!400!RSZDT
}

#####
# Define a service to check the load on the local machine.

define service{
  use                linux-service
  host_name          icinga-muc-01.fritz.box
  service_description CPU State
  check_command       check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

#####
# Define a service to check HTTP on the local machine.

define service{
  use                linux-service
  host_name          icinga-muc-01.fritz.box
  service_description HTTP Server
  check_command       check_http
  notifications_enabled 0
}

#####
```

Sieht nach erfolgreicher Abfrage von ICINGA wie folgt aus

**Service Status Details For Host  
'icinga.fritz.box'**

Host	Service	Status	Last Check	Duration	Attempt	Status Information
icinga.fritz.box	Alle Prozesse	OK	13-05-2010 20:07:25	0d 5h 27m 9s	1/5	PROCS OK: 60 processes with STATE = RSZDT
	CPU State	OK	13-05-2010 20:08:25	0d 5h 26m 9s	1/5	OK - load average: 0.00, 0.03, 0.08
	HTTP Server	OK	13-05-2010 20:09:25	0d 5h 25m 9s	1/5	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,000 second response time
	PING	OK	13-05-2010 20:05:25	0d 5h 24m 9s	1/5	OK - 192.168.178.30: rta 0,226ms, lost 0%
	Boot Partition	OK	13-05-2010 20:05:42	0d 4h 28m 52s	1/5	DISK OK - free space: /boot 30 MB (46% inode=99%):
	Root Partition	OK	13-05-2010 20:06:38	0d 4h 27m 56s	1/5	DISK OK - free space: / 8157 MB (76% inode=78%):
	Swap Partition	OK	13-05-2010 20:07:34	0d 4h 27m 0s	1/5	SWAP OK - 100% free (1456 MB out of 1466 MB)
	home Partition	OK	13-05-2010 20:08:31	0d 4h 26m 3s	1/5	DISK OK - free space: /home 2259 MB (92% inode=98%):
	tmp Partition	OK	13-05-2010 20:04:27	0d 4h 25m 7s	1/5	DISK OK - free space: /tmp 891 MB (98% inode=99%):
	var Partition	OK	13-05-2010 20:05:23	0d 4h 24m 11s	1/5	DISK OK - free space: /var 2941 MB (80% inode=96%):
	angemeldete Benutzer	OK	13-05-2010 20:08:55	0d 5h 25m 39s	1/5	USERS OK - 1 users currently logged in

11 Matching Service Entries Displayed

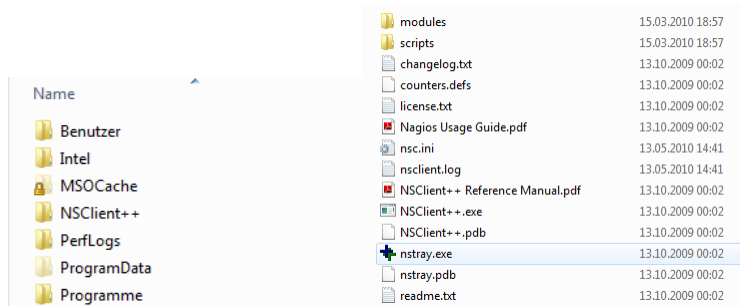
## tnetd.fritz.box.cfg

Ähnlich läuft es auch bei den Windows Server/Client ab, nur benötige ich für den Windows Server/Client noch zusätzlich einen Agent der im Hintergrund läuft. Diesen Agent bekommt man im Internet → <http://nscient.org/nscpl/> → und es handelt sich um den NSClient++ Agent.

Download → x86 oder x64 Clients → <http://sourceforge.net/projects/nscplplus/files/nscplplus/>

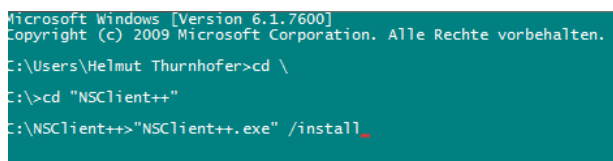


Diesen Client extrahiere ich und benenne den Ordner um in NSClient++ und kopiere diesen Ordner auf jeden Server/Client der überwacht werden soll lokal in das Verzeichnis C:\

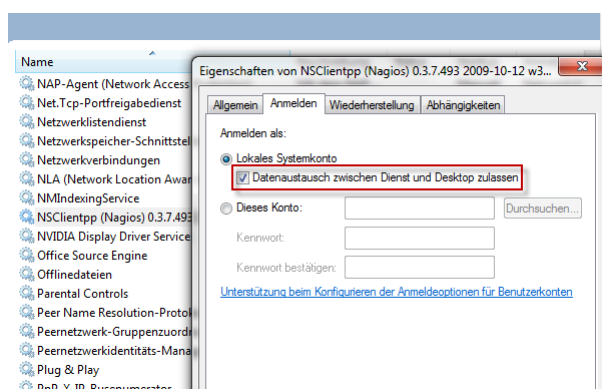


Danach rufe ich eine CMD Console auf und wechsel in das C:\ Verzeichnis

```
cd \
cd NSClient++
NSClient++.exe /install
```



Nach erfolgreicher Installation des Dienstes, öffnen wir die services.msc über Start → Ausführen und setzen das Häkchen „**Datenaustausch zwischen Dienst und Desktop zulassen**“ im Reiter Anmelden, danach können wir den Dienst getrost starten. Überprüft im Reiter Allgemein das der Dienst im Starttyp Automatisch drinstehen hat.



Im nächsten Schritt muss noch lokal am Client die **nsc.ini** bearbeitet werden, ich führe hier nur die Dinge auf, bei denen ich am Anfang vom Eintrag das Semikolon entfernt habe.

Unter [modules] wurden folgende DLL's freigeschaltet

FileLogger.dll  
 CheckSystem.dll  
 CheckDisk.dll  
 NSClientListener.dll  
 NRPEListener.dll  
 SysTray.dll  
 CheckEventLog.dll  
 CheckHelpers.dll

Unter [Settings] wurden folgende Punkte konfiguriert

**password=client\_password** (Wenn Ihr hier ein Passwort einträgt, dann muss das in der commands.cfg ebenfalls mit angegeben werden → command\_line \$USER1\$/check\_nt -H \$HOSTADDRESS\$ -p 12489 -sclient\_password -v \$ARG1\$ \$ARG2\$)  
**allowed\_hosts=192.168.178.30**  
**use\_file=1**  
**shared\_session=1** (Wegen Windows Vista/7 bzw. Server 2008/2008 R2)

Unter [NSClient]

**allowed\_hosts=192.168.178.30**  
**port=12489**

Unter [Check System]

**CPUBufferSize=1h**

Unter [NSCA Agent]

**hostname=tetrd.fritz.box**  
**nsca\_host=192.168.178.30**

jetzt starten wir nochmal den NSClient Dienst durch mit **net stop/start NSClientpp** (Hier aufpassen das das Shortcut Icon unten rechts bei der Uhr ebenfalls beendet wurde)



```
C:\NSClient++>net stop NSClientpp
NSClientpp (Nagios) 0.3.7.493 2009-10-12 w32 wird beendet.
NSClientpp (Nagios) 0.3.7.493 2009-10-12 w32 wurde erfolgreich beendet.

C:\NSClient++>net start NSClientpp
NSClientpp (Nagios) 0.3.7.493 2009-10-12 w32 wird gestartet.
NSClientpp (Nagios) 0.3.7.493 2009-10-12 w32 wurde erfolgreich gestartet.
```

Und die Konfigurations-Datei auf den ICINGA Server sieht wie folgt aus

```
#####
# HOST DEFINITION
#####

define host{
use                windows-server                ; Host Template (templates.cfg)
host_name          tnetd.fritz.box                ; FQDN - Servername
alias              TestClient für Doku            ; Frei definierbarer Anzeigenname
address            192.168.178.22                ; Statische IP Adresse vom Server
hostgroups         location2                      ; Hostgruppenname (hostgroup.cfg)
parents            fritz.box                      ; Unmittelbarer Nachbar (Physikalischer Switch/Router)
}
#####
# SERVICE DEFINITION
#####
# Server Uptime

define service{
use                windows-service
host_name          tnetd.fritz.box
service_description Server Uptime
check_command      check_nt!UPTIME
}
#####
# CPU Auslastung

define service{
use                windows-service
host_name          tnetd.fritz.box
service_description CPU Auslastung
check_command      check_nt!CPULOAD!-l 10,90,95,30,90,95
}
#####
# Memory Auslastung

define service{
use                windows-service
host_name          tnetd.fritz.box
service_description Memory Auslastung
check_command      check_nt!MEMUSE!-w 80 -c 90
}
#####
# Monitoring C:\ Festplatte

define service{
use                windows-service
host_name          tnetd.fritz.box
service_description C:\ Festplatte
check_command      check_nt!USEDISKSPACE!-l c -w 80% -c 90%
}
#####
```

```
# Service Check

define service{
    use                windows-service
    host_name          tnetd.fritz.box
    service_description Service Check
    check_command       check_nt!SERVICESTATE!-d SHOWALL -l 'NSClientpp'
}

#####
```

Sieht dann nach Erfolgreichen Check von ICINGA wie folgt aus

**Service Status Details For Host  
'tnetd.fritz.box'**

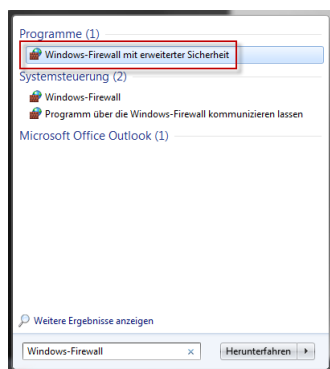
Host	Service	Status	Last Check	Duration	Attempt	Status Information
tnetd.fritz.box	C:\ Festplatte	OK	13-05-2010 21:04:55	0d 6h 20m 54s	1/5	c: - total: 49,90 Gb - used: 25,49 Gb (51%) - free 24,41 Gb (49%)
	CPU Auslastung	OK	13-05-2010 21:00:55	0d 6h 19m 54s	1/5	CPU Load 15% (10 min average) 8% (30 min average)
	Memory Auslastung	OK	13-05-2010 21:04:01	0d 0h 21m 48s	1/5	Memory usage: total:4092,39 Mb - used: 2005,13 Mb (49%) - free: 2087,27 Mb (51%)
	Server Uptime	OK	13-05-2010 21:03:55	0d 0h 21m 54s	1/5	System Uptime - 0 day(s) 8 hour(s) 9 minute(s)
	Service Check	OK	13-05-2010 21:05:39	0d 0h 0m 10s	1/5	NSClientpp: Started

5 Matching Service Entries Displayed

Sollte aber auf den neuen Windows Server 2008/2008 R2 oder neuen Windows 7 Clients die Checks nicht sauber laufen, liegt es mit großer Wahrscheinlichkeit an der Firewall Einstellung oder das der Client Agent nicht gestartet ist.

## Windows Firewall

Firewall konfiguriert Ihr wie folgt → Windows Logo → Windows-Firewall eingeben, und das Ganze mit der erweiterten Sicherheit aufrufen.



Danach werden zwei neue Regeln erstellt, eine Eingehende und eine Ausgehende, beide Regeln sehen gleich aus.

Welchen Regeltyp möchten Sie erstellen?

☒ **Programm**  
Regel, die die Verbindungen für ein Programm steuert.

☐ **Port**  
Regel, die die Verbindungen für einen TCP- oder UDP-Port steuert.

☐ **Vordefiniert:**  
Anmeldedienst  
Regel, die die Verbindungen für einen Windows-Vorgang steuert.

☐ **Benutzerdefiniert**  
Benutzerdefinierte Regel

Wir wählen das Programm aus

Betrifft diese Regel alle oder nur ein bestimmtes Programm?

☐ **Alle Programme**  
Die Regel wird auf alle Computerverbindungen angewendet, die mit anderen Regeleigenschaften übereinstimmen.

☒ **Dieser Programmpfad:**  
   
Beispiel: c:\Pfad\Programm.exe  
          %ProgramFiles%\Browser\Browser.exe

Geben den Pfad zur NSClient++.exe an

Welche Aktion soll durchgeführt werden, wenn eine Verbindung die angegebenen Bedingungen erfüllt?

☒ **Verbindung zulassen**  
Dies umfasst sowohl mit IPsec geschützte als auch nicht mit IPsec geschützte Verbindungen.

☐ **Verbindung zulassen, wenn sie sicher ist**  
Dies umfasst nur mithilfe von IPsec authentifizierte Verbindungen. Die Verbindungen werden mit den Einstellungen in den IPsec-Eigenschaften und -regeln im Knoten "Verbindungssicherheitsregel" gesichert.

☐ **Verbindung blockieren**

Setzen den Punkt auf Verbindung zulassen

Wann wird diese Regel angewendet?

☒ **Domäne**  
Wird angewendet, wenn ein Computer eine Verbindung mit der Firmendomäne hat.

☒ **Privat**  
Wird angewendet, wenn ein Computer eine Verbindung mit einem privaten Netzwerk hat.

☒ **Öffentlich**  
Wird angewendet, wenn ein Computer eine Verbindung mit einem öffentlichen Netzwerk hat.

Wo soll die Regel angewandt werden? Denke ist selbsterklärend. 😊 Obwohl ich bei den Windows Server 2008 Core Server tatsächlich Domäne und Öffentlich anhäkeln musste, um eine Verbindung zum ICINGA/NAGIOS Server zu bekommen.

Name:  
ICINGA - NSClient++ InBound

Beschreibung (optional):

Dann gebt der Regel noch einen Namen und das sollte es gewesen sein.

## commands.cfg

Die letzte Änderung für die Grundkonfiguration ist das ändern der Service abfragen in der commands.cfg

Bei dem Service **check\_host\_alive** ändere ich das Command von check\_ping in check\_icmp

```
# 'check-host-alive' command definition
define command{
    command_name    check-host-alive
    command_line    $USER1$/check_icmp -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -p 5
}
```

Und dasselbe mache ich auch noch bei dem Command **check\_ping**

```
# 'check_ping' command definition
define command{
    command_name    check_ping
    command_line    $USER1$/check_icmp -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
}
```

Grund dafür ist das mir der check\_ping keine Performance Daten ausgibt check\_icmp aber schon, die Daten benötige ich zu einen späteren Zeitpunkt.

## fritz.box.cfg

Und nun als letztes definiere ich meinen ersten parent (Router oder Switch an dem der Server oder Client physikalisch angeschlossen ist)

```
#####
# HOST DEFINITION
#####

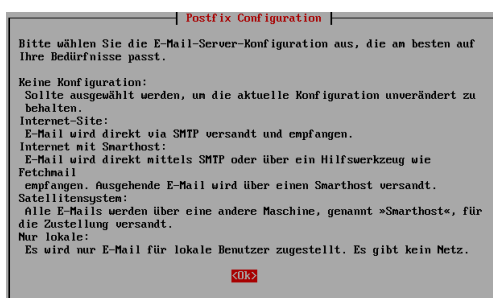
define host{
use          router-host      ; Host Template (templates.cfg)
host_name    fritz.box        ; FQDN - Servername
alias        Fritz Box        ; Frei definierbarer Anzeigename
address      192.168.178.1     ; Statische IP Adresse vom Server
hostgroups   location3        ; Hostgruppenname (hostgroup.cfg)
}
#####
# SERVICE DEFINITION
#####

define service{
use          router-service
host_name    fritz.box
service_description PING
check_command check_ping!100.0,20%!500.0,60%
}
#####
```

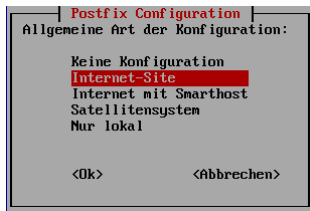
## E-Mail Benachrichtigung auf dem Server aktivieren

Um sich über ICINGA Nachrichten zuschicken zu lassen, wie z.B. den Status der einzelnen Services, muss Standardmäßig bei Ubuntu noch ein paar Dinge auf den Server Installieren bzw. Konfigurieren. Als erstes benötigen wir ein Programm das über SMTP Mails verschicken kann und ein weiteres Programm um über die Konsole Testnachrichten schicken zu können.

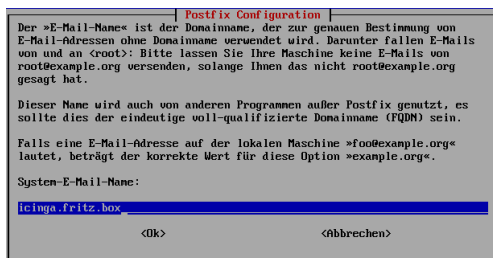
***sudo apt-get install postfix libsasl2-modules mailx***



Beim Installieren von Postfix startet der Konfigurations- Wizard, diesen bestätigen wir mit OK



Wählen irgendeine Einstellung aus und klicken auf OK



Bestätigen den Hostnamen und drücken nicht auf OK sondern die Enter Taste, danach wird der Konfigurations- Wizard geschlossen und die Installation schließt ab.

***sudo nano /etc/postfix/main.cf***

Im nächsten Schritt rufen wir die Konfigurationsdatei von Postfix auf und stellen händisch die passenden Parameter ein. Geht viel einfacher. ☺

```
myhostname = icinga.fritz.box
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = icinga.fritz.box
relayhost = smtp.provider.de
mailbox_size_limit = 51200000
recipient_delimiter =
inet_interfaces = all
default_transport = smtp
relay_transport = smtp
inet_protocols = ipv4

disable_dns_lookups = yes
```

Mehr über die Parameter könnt ihr hier lesen → <http://wiki.ubuntuusers.de/postfix>

Mit STRG + O + Enter speichert Ihr die Datei und mit STRG + X verlasst ihr nano.



```
# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

myhostname = icinga.fritz.box
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = icinga.fritz.box
relayhost = smtp.
mailbox_size_limit = 51200000
recipient_delimiter =
inet_interfaces = all
default_transport = smtp
relay_transport = smtp
inet_protocols = ipv4

disable_dns_lookups = yes

smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noplaintext noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl_password
sender_canonical_maps = hash:/etc/postfix/sender_canonical
```

Jetzt ruft Ihr eure hosts Datei auf und tragt dort euren SMTP Server von eurer Firma oder Provider ein.

***nano /etc/hosts --> 192.168.0.100 smtp.provider.de***

Solltet Ihr tatsächlich Probleme haben mit irgendwelchen Authentifizierungen beim SMTP Server helfen euch noch folgende Parameter weiter. (Bitte hier ein wenig spielen, bis Dato habe ich Sie noch nicht benötigt)

***sudo nano /etc/postfix/main.cf***

***smtp\_sasl\_auth\_enable = yes***

***smtp\_sasl\_security\_options = noplaintext noanonymous***

***smtp\_sasl\_password\_maps = hash:/etc/postfix/sasl\_password***

***sender\_canonical\_maps = hash:/etc/postfix/sender\_canonical***

***sudo touch /etc/postfix/sasl\_password***

***sudo nano /etc/postfix/sasl\_password***

***smtp.provider.de      username:ganzgeheimesspassword***

***sudo chmod 600 /etc/postfix/sasl\_password***

***sudo touch /etc/postfix/sender\_canonical***

***sudo nano /etc/postfix/sender\_canonical***

***benutzer      [email@adresse.de](mailto:email@adresse.de)***

***sudo postmap /etc/postfix/sasl\_password***

***sudo postmap /etc/postfix/sender\_canonical***

***sudo /etc/init.d/postfix restart***

So und zu guter Letzt must ihr noch die commands.cfg von icinga anpassen.

**`sudo nano /usr/local/icinga/etc/company/commands.cfg`**

dort findet Ihr am Anfang von der Datei die beiden Service Definitionen für die Mail Einstellungen.  
In der **command\_line** Zeile scrollt Ihr ganz ans Ende bis Ihr die Parameter **/etc/mail** findet, diese Parameter ersetzt Ihr bitte durch **/usr/bin/mailx**

```
#####
#
# SAMPLE NOTIFICATION COMMANDS
#
# These are some example notification commands. They may or may not work on
# your system without modification. As an example, some systems will require
# you to use "/usr/bin/mailx" instead of "/usr/bin/mail" in the commands below.
#
#####

# 'notify-host-by-email' command definition
define command{
    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Icinga *****\n\nNotificatio$
}

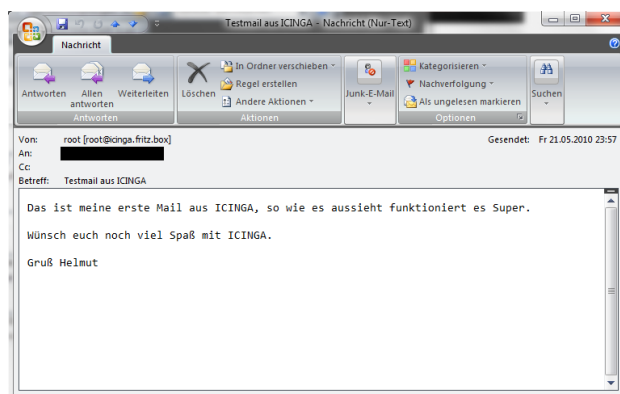
# 'notify-service-by-email' command definition
define command{
    command_name    notify-service-by-email
    $DATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$"; /usr/bin/mailx -s "## $NOT$
}
```

Danach Speichert Ihr alles und könnt den Server noch einmal durchstarten.

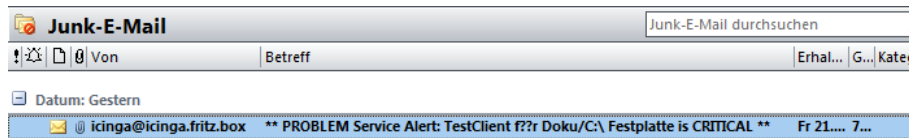
Nach dem Neustart bitte überprüfen ob auch alle Dienste gestartet sind (ido2db, icinga, apache2, postfix)

Wenn dem so ist könnt Ihr mal eine Testmail über die Konsole schicken.

```
root@ICINGA:~# mail email@adresse.de
Subject: Testmail aus ICINGA
Das ist meine erste Mail aus ICINGA, so wie es aussieht funktioniert es Super.
Wünsch euch noch viel Spaß mit ICINGA.
Gruß Helmut
.
EOT
```



Und sobald man seinen ersten Fehler produziert in dem man zum Beispiel auf den Windows Client einmal ein Dienst Stoppt, mach icinga genau das was es soll.



E-Mails verschicken wenn es darauf ankommt. ☺

Ich für meinen Teil wäre jetzt fertig, wenn alles Richtig gemacht wurde habt Ihr ein lauffähiges ICINGA 1.0.2 System.

Wünsche euch viel Spaß damit. ☺

Gruß Helmut Thurnhofer