# ICINGA in einer Virtuellen Umgebung mit Ubuntu 9.10 Desktop installieren.

### Inhalt

ICINGA Einführung 2
Ubuntu 9.10 Desktop Installation
Vorbereitung für die ICINGA Installation
Benutzer und Gruppen Informationen anlegen9
ICINGA und die IDOUtils kompilieren und installieren9
MySql Datenbank anlegen 11
Nagios Plugins installieren13
ICINGA Grundkonfiguration15
Ordnerstruktur anlegen 15
houstgroup.cfg
localhost.cfg
templates.cfg
icinga.cfg
timeperiods.cfg
icinga.fritz.box.cfg
tnetd.fritz.box.cfg
Windows Firewall
commands.cfg 31
fritz.box.cfg



# **ICINGA Einführung**

### Einführung:

Diese Anleitung ist dazu gedacht, dir eine einfache Anweisung zu liefern, wie die Software ICINGA innerhalb von 30 Minuten aus dem Quellcode installiert und deinen lokalen Rechner überwacht.

Diese Anleitung enthält Anweisungen für eine einzige Linux-Distributionen: Ubuntu 9.10 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:

Wenn du dieser Anweisungen folgst, wird ein lauffähiges ICINGA System sein:

- Icinga und die Plugins werden unterhalb von /usr/local/icinga installiert sein
- Icinga wird so konfiguriert sein, dass es einige Dinge auf deinem lokalen System bzw. entfernten Windows System überwacht (CPU-Auslastung, Plattenbelegung, usw.)
- Das Icinga-Web-Interface ist erreichbar unter http://localhost/icinga/
- Eine Datenbank, die von Icinga mit Hilfe von IDOUtils gefüllt wird

### Voraussetzungen:

Bei einigen Teilen der Installation benötigst du root-Zugang zu deiner Maschine. 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
- GCC-Compiler
- C/C++ development libraries
- GD-Development-Libraries
- libdbi/libdbi-Treiber, eine Datenbank wie z.B. MySQL

### Auf los geht's los.

Meine Testumgebung wurde mit einer Virtuellen Ubuntu 9.10 Desktop System, einen Windows 7 Enterprise Client und einer Fritzbox als Router durchgeführt. Auch wurde das ganze schon in einer Windows Server 2008 R2 Hyper-V Umgebung getestet und ausgeführt. Werde aber in dieser Dokumentation nur auf das vorgeführte System eingehen

Also viel Spaß beim Lesen und nachmachen. 😊



# **Ubuntu 9.10 Desktop Installation**

Auf folgender Webseite bekommt man das ISO Image für Ubuntu 9.10 Desktop x86/x64  $\rightarrow$  <u>http://releases.ubuntu.com/9.10/</u>

Für meine Testumgebung setze ich von Sun Microsystems - VirtualBox ein → <a href="http://www.virtualbox.org/">http://www.virtualbox.org/</a>

Virtuelle Maschine anlegen, ISO Image einbinden, Virtuelle Maschine starten.



Deutsche Sprache auswählen und mit der Enter Taste weiter



Den Eintrag "Ubuntu installieren" auswählen und mit der Enter Taste weiter



lo localization	Willkommen
sturianu	Bereit zur Installation? Nachdem Sie einige Fragen
lahasa Indonesia	beantwortet haben, kann der Inhalt der Live-CD auf diesem
losanski	Rechner installiert werden, so dass Sie das System mit
atalà	volici deschwindigkeit und onne die co bendtzen kommen.
eština	Das Beantworten der Fragen sollte nur wenige Minuten in
ymraeg	Anspruch nehmen.
ansk	Bitte wählen Sie die Sprache, die für die Installation
eutsch	benutzt werden soll. Die Sprache wird auch die
iesti	Standardsprache für das installierte System.
nglish	
spañol	
speranto	
uskara	
rançais	
iaeilge	
ialego	
Irvatski	
taliano	
lurdî	Wenn Sie einen Internetzugang haben, lesen Sie die
atviski	Informationen über Probleme, die Sie betreffen
ietuviškai	könnten, zu erhalten.
lagyar	Informationen zur aktuellen Version
a dealer de	×

### Willkommensbildschirm mit Vor bestätigen



### Regionalsettings auswählen und mit Vor bestätigen



Tastaturlayout auswählen und mit *Vor* bestätigen (Bei Hyper-V gibt es hier schon die ersten Probleme, da der Tastatur Treiber nicht sauber geladen wird – mit dem Englischen Layout kommt man aber weiter)



ICINGA (wird ausgeführt) - Sun VirtualBox	
Maschine Geräte Hilfe	
Die Festplatte vorbereiten	
Auf diesem Rechner ist kein Betriebssystem vorhanden.	
No möchten Sie Ubuntu 9.10 platzieren?	
<ul> <li>Löschen und das gesamte Medium verwenden</li> </ul>	
SCSI1 (0,0,0) (sda) - 21.5 GB ATA VBOX HARDDISK	
O Partitionen manuell festlegen (fortgeschritten)	
Pa	rtitionen manuell festlegen (fortgeschritten)
	l>
Schritt 4 von 6	Beenden Zurück Vor

Partitionen manuell festlegen auswählen und mit Vor bestätigen.



Neue Partitionstabelle anlegen – Warnmeldung lesen und mit Weiter bestätigen

)ie Part	itionen vo	rbereiten		₽			
	sda1 (ext4) 70.6 MB	sda5 (linux-sv 1.4 GB	vap) 🔳 sda6 972.7	(ext4) 🔲 7 MB	sda7 (ext4) 3.8 GB	sda8 (ext4) 11.2 GB	sda9 (ext4) 2.6 GB
aufwerk	Verwendung	Einhängepunkt	Formatieren?	Größe	Belegt		
dev/sda							
/dev/sda1	ext4	/boot	2	73 MB	unbekannt		
/dev/sda5	swap			1538 MB	unbekannt		
/dev/sda6	ext4	/tmp		1019 MB	unbekannt		
/dev/sda7	ext4	/var		4096 MB	unbekannt		
/dev/sda8	ext4	1	2	12000 MB	unbekannt		
/001/3000							
/dev/sda9	ext4	/home		2738 MB	unbekannt		
/dev/sda9	ext4	home	E.	2738 MB	unbekannt		

Festplatte wurde jetzt von mir so eingerichtet wie im Bild gezeigt, (Unix Administratoren werde ich hier nichts erzählen, denke das Sie wissen was sie machen.☺)

75MB /boot EXT4, 1,5-2\*RAM SWAP, 12GB / EXT4, 1GB /tmp EXT4, 4GB /var EXT4, REST /home EXT4

Bei einer größeren Festplatte natürlich /root, /home, /var dementsprechend anpassen.



FICINGA [wird ausgeführt] - Sur	VirtualBox		
Maschine Geräte Hilfe			
Wer sind Sie?			
Wie heißen Sie?			
Administrator			
Welchen Namen möchten S	Sie zum Anmelden verwenden?		
administrator			
Wenn mehr als eine Person o Benutzerkonten anlegen.	len Rechner verwenden wird, können Si	e nach der Installation mehrere	
Wählen Sie ein Passwort, u	m Ihr Benutzerkonto abzusichern.		
Wie heißt dieser Rechner?			
ICINGA			
Dieser Name wird benutzt, w	enn Sie Ihren Rechner für andere Rechn	er in einem Netzwerk sichtbar machen.	
O Automa			
Mein Passwort ist zum /	Anmelden erforderlich		
<ul> <li>Mein Passwort ist zum /</li> </ul>	Anmelden und dem Entschlüsseln m	neines persönlichen Ordners erforder	lich
Schritt 6 von 7		Beenden	Zurück Vor

Benutzerdaten anlegen und mit Vor bestätigen



Zusammenfassung überprüfen und wenn alles passt auf Installieren klicken.



Ubuntu System wird Installiert





Nach der Installation den Client/Server neu starten, Image aus dem CDROM entfernen.

Nach dem nächsten Neustart alle Updates von Ubuntu installieren über System --> Systemverwaltung --> Aktualisierungsverwaltung



Ubuntu ist fertig für die Vorbereitung von ICINGA.



# Vorbereitung für die ICINGA Installation

Folgende Pakete müssen noch nachinstalliert werden, geht wie folgt: Ich öffne ein Terminal über Anwendungen → Zubehör → Terminal Bei geöffnetem Terminal übernehme ich mit folgendem Befehl die Root Rechte

sudo –s (Passwort)

 root@JCINGA: ~

 Datei
 Bearbeiten
 Ansicht
 Terminal
 Hilfe

 administrator@ICINGA:-\$
 sudo -s
 [sudo]
 [sudo]
 s

 [sudo]
 ressword for administrator:
 root@ICINGA:-\*
 sudo -s
 [sudo]
 ressword for administrator:

 root@ICINGA:-\*
 apt-get install build-essential apache2
 libgd2-xpm-dev

C-Compiler - Web-Server und GD-Libraries installieren apt-get install build-essential apache2 libgd2-xpm-dev

MySql Server – lib Treiber installieren 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

SNMP installieren apt-get install snmp snmpd libnet-snmp-perl

PHP Installieren apt-get install php5 php5-mysql

PEAR Installieren apt-get install php-pear php5-Idap php5-snmp php5-gd

SSL Installieren sudo apt-get install openssl libcurl4-openssl-dev



# Benutzer und Gruppen Informationen anlegen

Die ICINGA-Installation besteht aus dem Anlegen des Benutzers "icinga", der Vergabe eines Passworts, kompilieren der Source Dateien, Kopieren der Dateien und abschließenden Anpassungen.

Anlegen des Benutzers icinga

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 und die IDOUtils kompilieren und installieren

Wechsel in das lokale Source-Verzeichnis von Ubuntu *cd /usr/local/src* 

Download von den ICINGA Quelldateien wget <u>http://sourceforge.net/projects/icinga/files/icinga/1.0.1/icinga-1.0.1.tar.gz/download</u>

Download Dateien Extrahieren tar xvzf icinga-1.0.1.tar.gz

Wechsel in das Verzeichnis icinga-1.0.1 *cd icinga-1.0.1* 

Führe bitte das ICINGA-Konfigurations-Script aus ./configure --with-command-group=icinga-cmd --enable-idoutils



### Ausgabe vom Konfigurations-Script

*** Configuration summary for icinga-core 1.0.1 03-03-2010 ***:	
General Options: Icinga executable: icinga Icinga user/group: icinga,icinga Command user/group: icinga,icinga-cmd Embedded Perl: no Event Broker: yes Build IDOUTIS: yes Install \${prefix}: yusr/local/icinga Lock file: \${prefix}/var/icinga.lock Check result directory: /etc/jache2/conf.d Mail program: /bin/mail	
Host US: Linux-gnu	
GI URL: http://localhost/icinga/cgi-bin/ GI URL: http://localhost/icinga/cgi-bin/ Traceroute (used by WAP):	
Review the options above for accuracy. If they look okay,	

Jetzt wird der ICINGA-Source-Code kompiliert mit *make all* 

Mit folgendem Befehl werden die Binaries installiert *make fullinstall* 

Beispiel-Konfigurationsdateien werden durch folgenden Befehl

### make install-config

in das Verzeichnis /usr/local/icinga/etc/ kopiert.

Passe bitte in der Konfigurationsdatei **"/usr/local/icinga/etc/objects/contacts.cfg"** mit einem Editor deine e-Mail-Adresse in der *icingaadmin*-Kontaktdefinition an, so dass es die Adresse enthält, die im Falle von Alarmen benachrichtigt werden soll.

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



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 (**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

# Example:
#
#
broker\_module=<modulepath> [moduleargs]
#broker\_module=/somewhere/module1.0
#broker\_module=/somewhere/module2.0 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 Datenbank Server auf *mysql -u root -p (Passwort)* 

Datenbank icinga anlegen mysql> CREATE DATABASE icinga;

Benutzer icinga anlegen 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;

Berechtigungen für den Benutzer icinga setzen mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON icinga.\* TO 'icinga'@'localhost';

mysql> FLUSH PRIVILEGES; mysql> quit;



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



### Kontrollieren ob auch alles Sauber ausgeführt wurde (für Misstrauische 🙂)



Datenbankschema der Datenbank icinga hinzufügen

cd /usr/local/src/icinga-1.0.1/module/idoutils/db/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

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

cd /usr/local/src/icinga-1.0.1/ make install-webconf 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, bekommen wir 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 beide 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.fritz.box
192.168.178.30	icinga.fritz.box

# Nagios Plugins installieren

Wir wechseln wieder in das Verzeichnis *cd /usr/local/src* 

Laden die Quelldateien vom Server wget <u>http://surfnet.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.14.tar.gz</u>

Extrahieren die Quelldateien tar xvzf nagios-plugins-1.4.14.tar.gz

Wechseln in das Verzeichnis der Quelldateien *cd nagios-plugins-1.4.14* 

Führen das Konfiguratios- 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



### ICINGA Konfiguration überprüfen /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg

ICINGA 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 - Glückwunsch! Du hast ICINGA erfolgreich installiert.

### http://localhost/icinga



# **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 euerem Netzwerk überwachen könnt. Umso komplexer die Umgebung umso mehr Gehirnschmalz muss investiert werden damit ICINGA/NAGIOS sauber konfiguriert wird.

Sehr gute Anregung  $\rightarrow$  <u>http://www.nagios-wiki.de/nagios/tips/nagios\_optimal\_konfigurieren</u>

Oder das Buch von Wolfgang Barth  $\rightarrow$  <u>http://www.amazon.de/Nagios-System-Netzwerkmonitoring-Wolfgang-Barth/dp/3937514910/ref=sr 1 2?ie=UTF8&s=books&qid=1273768468&sr=8-2</u>

# 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 old objects.



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





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

# houstgroup.cfg

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

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, 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.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			
}				
*****	***************************************			

### Sieht dann in der ICINGA Oberfläche wie folgt aus

ry For All Host Groups r All Host Groups											
				Service Over	view For	All Host (	Groups				
Server	Standor	t 1 (location1	1)	Serve	er Stando	ort 2 (locatio	m2 <b>)</b>		Router	(location	3)
Host	Status	Services A	Actions	Host	Status	Services	Actions	Host S	Status	Services	Actions
icinga.fritz.box	UP	11 OK		tnetd.fritz.box	UP	4 OK		fritz.box	UP	1 OK	
Firma Must	ermann	Inc. <b>(</b> muster	mann)								
Host	Status	Services A	Actions								
fritz.box	UP	1 OK									
icinga.fritz.box	UP	11 OK									
tnetd.fritz.box	UP	4 OK									

# localhost.cfg

Im nächste Schritt kopiere ich die Original Datei localhost.cfg in meinen Ordner standort1, 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)



Dasselbe mache ich auch bei Standort 2 und 3

# templates.cfg

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



#######################################	*************	*#############	*########	*######	ŧ
# CONTACT TEMPLATES					
*****			*****	*######	ŧ
define contact{					
			c		

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	s notify-service-by-ema	ail ; 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
1		

#######################################	**************	*****
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!
}		

****		*****
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!

}

19

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.



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.

# OBJECT CONFIGURATION FILE(S) # These are the object configuration files in which you define hosts, # host groups, contacts, contact groups, services, etc. # You can split your object definitions across several config files # if you wish (as shown below), or keep them all in a single config file. # You can specify individual object config files as shown below: #cfg\_file=/usr/local/icinga/etc/objects/conmands.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/localhost.cfg #cfg\_file=/usr/local/icinga/etc/objects/localhost.cfg #cfg\_file=/usr/local/icinga/etc/objects/windows.cfg #cfg\_file=/usr/local/icinga/etc/objects/witch.cfg #cfg\_file=/usr/local/icinga/etc/objects/printer.cfg # Definitions for monitoring a Folder Share 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. Valid options # include: # us (MM-DD-YYYY HH:MM:SS) # euro (DD-MM-YYYY HH:MM:SS) <mark>iso</mark>8601 # (YYYY-MM-DD HH:MM:SS) strict-<mark>iso</mark>8601 # (YYYY-MM-DDTHH:MM:SS) # date\_format<mark>=</mark>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
1		

# This defines a modified "24x7" timeperiod that covers every day of the

# year, except for U.S. holidays (defined in the timeperiod above).

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



# define host{

use	linux-server	; Host Template (templates.cfg)
host_name	icinga.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 (Phyisikalischer Switch/Router)
}		

***************************************	####
# Ping Local Machine	

#### define service{

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

# users on the local machine.

define service{	
use	linux-service
host_name	icinga.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.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.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.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.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.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
hast name	isingo frita hoy
nost_name	
service_description	_home Partition
check_command	check_local_disk!20%!10%!/home
}	

Helle's Portfolio

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

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

Sieht nach erfolgreicher Abfrage von ICINGA wie folgt aus

#### Status 🗤 Last Check 🗤 Duration 🗤 Attempt 🗤 Status Information Host AT Service AT 13-05-2010 20:07:25 0d 5h 27m 9s PROCS OK: 60 processes with STATE = RSZDT nga.fritz.box Alle Proz 1/5 CPU State 13-05-2010 20:08:25 0d 5h 26m 9s 1/5 OK - load average: 0.00, 0.03, 0.08 HTTP Server 13-05-2010 20:09:25 0d 5h 25m 9s HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,000 second response tin 1/5 PING 13-05-2010 20:05:25 0d 5h 24m 9s 1/5 OK - 192.168.178.30: rta 0,226ms, lost 0% Boot Partition DISK OK - free space: /boot 30 MB (46% inode=99%): 13-05-2010 20:05:42 0d 4h 28m 52s 1/513-05-2010 20:06:38 0d 4h 27m 56s DISK OK - free space: / 8157 MB (76% inode=78%): Root Partition 1/5 Swap Partitior 13-05-2010 20:07:34 0d 4h 27m 0s SWAP OK - 100% free (1456 MB out of 1466 MB) 1/5 home Partition 13-05-2010 20:08:31 0d 4h 26m 3s 1/5 DISK OK - free space: /home 2259 MB (92% inode=98%): tmp Partition 13-05-2010 20:04:27 0d 4h 25m 7s 1/5DISK OK - free space: /tmp 891 MB (98% inode=99%): 13-05-2010 20:05:23 0d 4h 24m 11s DISK OK - free space: /var 2941 MB (80% inode=96%): var Partition 1/5 13-05-2010 20:08:55 0d 5h 25m 39s USERS OK - 1 users currently logged in angemeldete B 1/5

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

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  $\rightarrow$  <u>http://nsclient.org/nscp/</u>  $\rightarrow$  und es handelt sich um den NSClient++ Agent.

Download  $\rightarrow$  x86 oder x64 Clients  $\rightarrow$  <u>http://sourceforge.net/projects/nscplus/files/nscplus/</u>



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:\

	🐌 modules	15.03.2010 18:57
	🐌 scripts	15.03.2010 18:57
	📋 changelog.txt	13.10.2009 00:02
Name	counters.defs	13.10.2009 00:02
Name	📄 license.txt	13.10.2009 00:02
Benutzer	📕 Nagios Usage Guide.pdf	13.10.2009 00:02
Intel	👜 nsc.ini	13.05.2010 14:41
Jintei	📄 nsclient.log	13.05.2010 14:41
🐴 MSOCache	📕 NSClient++ Reference Manual.pdf	13.10.2009 00:02
NSClient++	NSClient++.exe	13.10.2009 00:02
Perflogs	NSClient++.pdb	13.10.2009 00:02
	💠 nstray.exe	13.10.2009 00:02
📗 ProgramData	nstray.pdb	13.10.2009 00:02
🐌 Programme	📄 readme.txt	13.10.2009 00:02

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.

Name	Eigenschaften von NSClientop (Nagios) 0.3.7.493 2009-10-12 w3
🔍 NAP-Agent (Network Access	
🔍 Net.Tcp-Portfreigabedienst	Allgemein Anmelden Wiederherstellung Abhängigkeiten
🔍 Netzwerklistendienst	
🔍 Netzwerkspeicher-Schnittstel	Anmelden als:
🔍 Netzwerkverbindungen	Lokales Systemkonto
🔍 NLA (Network Location Awar	Datenaustausch zwischen Dienst und Desktop zulassen
🔍 NMIndexingService	
🐫 NSClientpp (Nagios) 0.3.7.493	Dieses Konto:
🔍 NVIDIA Display Driver Service	Kennwort:
🔍 Office Source Engine	
🔍 Offlinedateien	Kennwort bestätigen:
🔍 Parental Controls	Unterstützung beim Konfigurieren der Anmeldeoptionen für Benutzerkonten
🔅 Peer Name Resolution-Proto	
🔅 Peernetzwerk-Gruppenzuord	
🔍 Peernetzwerkidentitäts-Mana	
🔍 Plug & Play	
Q PnP-X-IP-Busenumerator	

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\_passwort (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\_passwort -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=tnetd.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

### 

# HUST DEFINITION		
#####################	*************************************	*******
define host{		
use	windows-server	; Host Template (templates.cfg)
host_name	tnetd.fritz.box	; FQDN - Servername
alias	TestClient für Doku	; Frei definierbarer Anzeigename
address	192.168.178.22	; Statische IP Adresse vom Server
hostgroups	location2	; Hostgruppenname (hostgroup.cfg)
parents	fritz.box	; Unmittelbarer Nachbar (Phyisikalischer Switch/Router)
}		
#######################################		*****
# SERVICE DEFINITIO	N	
##################	*****	*****
# 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!-I 10,90,95,30,90,	95
}		
######################	***************************************	******
# Memory Auslastur	g	
define consider (		
define service{	windows sonviso	
use bost name	that fritz hav	
service description	Memory Auslastung	
check command	check ntIMEMUSEL-w 80 -c 90	
, ######################		****
# Monitoring (.) Fes	tnlatte	
	(platte	
define service{		
use	windows-service	
host_name	tnetd.fritz.box	
service_description	C:\ Festplatte	
check_command	check_nt!USEDDISKSPACE!-l c -w 80%	-c 90%
}		



# Service Check	
define service{ use host_name service_description check_command } ###################################	windows-service tnetd.fritz.box Service Check check_nt!SERVICESTATE!-d SHOWALL -I 'NSClientpp'
Sieht dann nac	h 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  $\rightarrow$  Windows Logo  $\rightarrow$  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.



W	elchen Regeltyp möchten Sie erstellen?			
۲	Programm Regel, die die Verbindungen für ein Programm steuert.			
0	) Port Regel, die die Verbindungen für einen TCP-oder UDP-Port steuert.			
	Anmeldedienst			
	Regel, die die Verbindungen für einen Windows-Vorgang steuert.			
0	Benutzerdefiniert Benutzerdefinierte Regel			

### Wir wählen das Programm aus



### Geben den Pfad zur NSClient++.exe an



### Setzen den Punkt auf Verbindung zulassen

Wann wird diese Regel angewendet?	
Domāne Wird angewendet, wenn ein Computer eine Verbindung mit der	Firmendomäne hat.
Privat Wird angewendet, wenn ein Computer eine Verbindung mit eine	em privaten Netzwerk hat.
Offentlich Wird angewendet, wenn ein Computer eine Verbindung mit ein	em öffentlichen Netzwerk hat.

Wo soll die Regel angewandt werden? Denke ist selbsterklärend. <sup>(2)</sup> Obwohl ich bei den Windows Server 2008 Core Server tatsächlich Domäne und Öffentlich anhäkeln musste, um eine Verbindung zum ICINGA/NAGIOS Server zu bekommen.



ICINGA - NSClient++ I	nBound		
Beschreibung (optiona	):		

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

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

So ab jetzt wäre die Grundkonfiguration abgeschlossen und Ihr habt ein lauffähiges ICINGA System.

Wünsche euch viel Spaß damit. 🙂

Gruß Helmut Thurnhofer

32