

Projektdokumentation

Thema des Projektes:	Überwachung der von der Firma eingesetzten Server
Name:	Sascha Schwarz
Datum der Abgabe:	25.04.2008
Ausbildungsberuf:	Fachinformatiker/Systemintegration
Betrieb:	Firma
Ausbildender Betrieb:	Firma

Inhalt der Projektdokumentation

1	Projektbeschreibung	3
	1.1 Einleitung.....	3
	1.2 Projektbeschreibung.....	3
	1.3 Projektumfang.....	3
2	Ist-Analyse	3
	2.1 Ist-Analyse.....	3
	2.2 Administration.....	4
3	Soll-Konzept	5
	3.1 Anforderungen.....	5
4	Projektplanung	6
	4.1 Projektphasen mit Zeitplanung.....	6
	4.2 Meilensteinplanung.....	7
5	Evaluierung der Software	8
	5.1 Auswahlmatrix.....	8
	5.2 Auswahl der Software.....	8
	5.3 Kosten-Nutzen-Analyse.....	9
6	Überwachungsübersicht	10
7	Realisierung	10
	7.1 Anlegen des Users und der Gruppe <i>nagios</i>	10
	7.2 Kompilieren und Installieren des Quellcodes.....	10
	7.3 Installation der Nagios-Plugins.....	11
	7.4 Installation von <i>nrpe</i> auf den zu überwachenden Servern.....	12
	7.5 Konfiguration von Nagios.....	13
8	Qualitätssicherung	14
	8.1 Grundsätzliche Überprüfungen der Konfiguration.....	14
	8.2 Abschlusstests	15
	Anhang A – Glossar	16
	Anhang B – Beispiele der Konfigurationsdateien	17
	Anhang C – Übersicht der Server	21
	Anhand D – Quellen	22

1 Projektbeschreibung

1.1 Einleitung

Die Firma ist ein junges Unternehmen mit einem besonderen Leistungsangebot für den Online-Handel. Diese muss den industriellen Anforderungen unserer Kunden an die Verfügbarkeit und Performance bei einem sehr hohen quantitativen Wachstum gerecht werden. Es werden Produktclips für die größten Onlineshops in Deutschland bereitgestellt.

Dazu wird eine komplexe und hochverfügbare IT-Umgebung in verteilten Rechenzentren betrieben.

1.2 Projektbeschreibung

Ziel des Projektes ist die Überwachung der wichtigsten Server der Firma. Es handelt sich hierbei um 3 Webserver, einen Officeserver und einen Streaming-Server. Probleme sollten zeitnah erfasst werden und der zuständige Mitarbeiter soll auf dem schnellsten Weg benachrichtigt werden. Die Benachrichtigung erfolgt dabei per E-Mail und per SMS. Diese Überwachung und Benachrichtigung soll helfen, die Ausfallzeiten zu minimieren und Probleme mit den Ressourcen frühzeitig planen zu können. Der aktuelle Status der verschiedenen Server soll jederzeit über ein Benutzerinterface einzusehen sein.

1.3 Projektumfang

Der Umfang des Projektes ist die Evaluierung, technische Umsetzung und abschließende Tests der Realisierung.

Zu dem Projekt gehören die Auswahl einer geeigneten Überwachungssoftware, Auswahl der zu überwachenden Dienste, Ressourcen und die Konfiguration. Zum Abschluss des Projektes gehört die Durchführung von Funktionstests. Die Firma hat 5 unterschiedliche Softwareprodukte zur Auswahl gestellt.

2 Ist-Analyse

2.1 Ist-Analyse

Die Server, die überwacht werden sollen stehen teils im Rechenzentrum und teils in den Büroräumen der Firma. Sie verfügen über eine über das Internet erreichbare IP Adresse. Der Officeserver (Debian Linux) steht im Firmengebäude und ist über Twisted Pair (1000Mbit) mit dem Switch verbunden. Die 3 Webserver werden bei Host Europa gehostet und laufen ebenfalls unter Debian Linux. Der Streaming-Server (SUN Solaris) verfügt über eine 100Mbit Anbindung und wird bei Host Europa gehostet.

Der Streaming-Server verarbeitet die Requests der Kunden und stellt ihnen das angeforderte Videomaterial zur Verfügung. Auf dem ersten der Webserver ist die Internetpräsenz der Firma zu finden. Es handelt sich um ein Webhostingpaket und es ist kein Root-Zugang möglich. Der zweite Webserver steht den Herstellern zur Verfügung, um Produktvideos hochladen zu können. Der dritte Webserver fungiert als Datenbankserver mit MySQL und stellt dem Workflow verschiedene Webservices zur Verfügung.

Auf dem Officeserver ist die zentrale und folgende Dienste: LDAP, Samba und das Encoding der Videoclips. Ein Netzwerkplan ist im Anhang C zu finde. Derzeit findet keine Überwachung der Server statt.

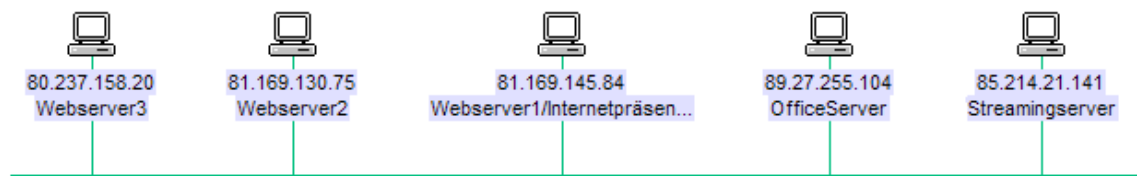


Abbildung 1: Serverübersicht

2.2 Administration

Die Administration erfolgt durch festangestellte Mitarbeiter und externe Firmen. Die Mitarbeiter der Firma haben vollen Zugriff auf alle Maschinen, die externen Mitarbeiter haben vollen Zugriff auf die jeweiligen Server, die sie verwalten.

3 Soll-Konzept

3.1 Anforderungen

Durch ein Gespräch mit dem Abteilungsleiter wurde folgende Anforderungen definiert, die alle durch die Software erfüllt werden müssen:

- auf allen Servern sollen die lokalen Ressourcen wie CPU Auslastung und Festplattenkapazität überwacht werden;
- Überprüfung, ob die einzelnen Server via Ping erreichbar sind;
- der aktuelle Status jedes einzelnen Servers soll jederzeit über ein Userinterface abrufbar sein. Dieses Interface soll mittels Username und Passwort geschützt werden;
- Definition von Eskalationsstufen – immer nur den zuständigen Mitarbeiter benachrichtigen; unnötige Meldungen reduzieren;
- freie und quelloffene Software;
- im Problemfall soll der zuständige Mitarbeiter Montags-Freitags von 8 – 17 Uhr per Mail kontaktiert werden; zu allen anderen Zeiten per SMS;
- spätere Anpassungen sollen problemlos möglich sein;
- Gruppieren von Hosts;
- die Webserver sollen auf deren Standarddienste kontrolliert werden wie FTP, POP3, IMAP, SMTP und SSH;
- beim Streaming-Server soll in regelmäßigen Abständen ein Teststreaming-URL aufgerufen werden;
- OTRS muss überwacht werden können;
- Programmierung eigener Plugins.

4 Projektplanung

4.1 Projektphasen mit Zeitplanung

Diese Auflistung entspricht der Planung aus dem Projektantrag.

Tätigkeit	Solldauer	Istdauer	Differenz	Begründung
Analysephase	2h	2h	0h	
IST-Zustand ermitteln	1h	1h	0h	
SOLL-Zustand ermitteln	1h	1h	0h	
Planungsphase	8,5h	10h	+1,5h	
Projektplanung anfertigen	1h	1h	0h	
Meilensteinplanung	0,5h	0,5h	0h	
Evaluierung der Software	5,5h	7h	1,5h	Die Evaluierung war deutlich umfangreicher als erwartet, da nicht jede Lösung eine Online-Demo hat.
Kosten/Nutzen Analyse	1,5 h	1,5 h	0 h	
Realisierungsphase	9,5h	7h	-2,5h	
Beschaffung der Software	1h	1h	0h	
Installation des Überwachungsservers	2,5h	0h	-2,5h	Es wurde ein bereits verfügbarer Webserver nach Absprache mit der Geschäftsführung gewählt
Installation und Konfiguration der Überwachungssoftware	6h	6h	0h	
Abschlussphase	15h	16h	+1h	
Testlauf und Fehlersuche	4h	4h	0h	
Dokumentation	9h	9h	0h	
Übergabe	2h	3h	+1h	Die Übergabe hat aufgrund mehrerer individueller Fragen länger gedauert.
Gesamt	35h	35h	0h	

4.2 Meilensteinplanung

Nr.	Meilenstein	Planziel	Ziel erreicht am	Bemerkung
1	Projektstart	24.03.2008	24.03.2008	
2	Ist- und Soll Zustand ermitteln	24.03.2008	24.03.2008	
3	Projektplanung	25.03.2008	25.03.2008	
4	Erstellung der Kriterien für die Auswahlmatrix			Anforderungen an die Software zusammengestellt
6	Evaluierung einer Softwarelösung	25.03.2008	25.03.2008	Die Evaluierung hat ergeben, dass Nagios das beste Produkt ist. Die Geschäftsführung entschied sich ebenfalls für Nagios
7	Beschaffung der Software und Basis-Installation	26.03.2008	26.03.2008	Download auf www.nagios.org
8	Konfiguration der gewählten Softwarelösung	26.03.2008	26.03.2008	
9	Durchführung von Tests	27.03.2008	27.03.2008	Testszzenarien mit gezielten Unterbrechungen von Leitungen und Abschalten von Diensten
10	Erstellung der Dokumentation	28.03.2008	28.03.2008	
11	Projektabschluss	28.03.2008	28.03.2008	

5 Evaluieren der Software

5.1 Auswahlmatrix

Die Firma hat folgende Softwarelösungen zur Auswahl gestellt.

	Nagios®	Opensmart	openNMS®	BigBrother
Preis	0€	0€	0€	100\$ pro monitored Node
Frontend	Browser CGIs	Browser CGIs	eigenes	Browser CGIs
Festplattenüberwachung	✓	✓	✓	✓
CPU Idle Time	✓	✗	✓ (snmp)	✓
Load Average	✓	✓	✓ (snmp)	✓
ping	✓	✓	✓	✓
Webserver Standarts (FTP,http, POP3, IMAP)	✓	✓	✓	✓
Gruppieren von Hosts	✓	✗	✓	✓
Service- und Hostabhängigkeiten	✓	✗	✓	✓
Empfehlung des Abteilungsleiters	✓	✗	✗	✗
Erfahrungswerte	✓	✗	✗	✗
Netzwerk-Visualisierung	✓	✓	✗	✓
OTRS-Unterstützung	✓	✗	✓	✗
SSH	✓	✓	✓	✓

5.2 Auswahl der Software

Nagios erfüllt alle Anforderungen die erfüllt werden sollen. Deshalb wird dieses Produkt gewählt.

Nagios ist frei im Quelltext und unentgeltlich auf der Homepage runterzuladen. Alle Systeme können zentral verwaltet werden und Änderungen an der IT können einfach und zeitnah erfasst werden. Es ist ohne weiteres möglich eigene Plugins zu programmieren, was im Zusammenhang mit den OTRS und dem Streamingserver einen enormen Vorteil bedeutet. Durch die Anpassung der Abhängigkeit der Hosts und Services kann gewährleistet werden, dass wirklich nur sinnvolle Warn- und Fehlermeldungen per Mail und SMS an den entsprechenden Mitarbeiter verschickt werden. Unnötige Nachrichten werden minimiert. Nagios hat eine belebte und große Community, die sehr hilfreich bei Fragen oder Probleme ist.

OpenNMS ist ein ebenfalls sehr gutes Produkt, das viele der Anforderungen erfüllt. Dadurch, dass jedoch die lokalen Ressourcen der einzelnen Server über SNMP kontrolliert werden und SNMP immer wieder durch Sicherheitslücken aufgefallen ist, hat die Geschäftsführung in Absprache mit mir diese Software abgelehnt.

OpenSmart und BigBrother erfüllen nicht den Umfang, der gewünscht ist, es ist nicht möglich die Überwachung mit diesen Tool vorzunehmen wie sie von der Geschäftsführung erwünscht ist.

5.3 Kosten-Nutzen-Analyse

Damit Produktvideos erfolgreich an die Onlineshop und Portal ausgeliefert werden können, muss gewährleistet werden das die Serverausfallzeiten so gering wie möglich gehalten zu werden. Jeder nicht beantwortet Request mindert den Umsatz.

Herstellern muss es möglich sein, jederzeit neues Videomaterial hochzuladen, damit die Shops mit aktuellen Videos versorgt werden können. Eine hochverfügbare IT-Infrastruktur sichert die Auslieferung von Videoclips, da dies das Kerngeschäft der ist, ist der Nutzen einer Überwachungssoftware sehr hoch.

Die Kosten einer solchen Lösung sind im Vergleich zum Nutzen sehr gering.

Installation und Konfiguration	*€
Lizenzkosten	*€
Laufende Kosten pro Jahr	*€
Total	*€

Die Kosten eines Ausfalls, der z.B. erst eine Stunde später bemerkt wird, kostet schon mehr als das Projekt. Ein Ausfall von einer Stunde von durchschnittlich XXX* Request in der Stunde kostet: XXX* Euro

* genaue Zahlen dürfen an dieser Stelle nicht genannt werden

6 Überwachungsübersicht

	ftp	pop3	smtp	HDD	CPU	Ping	http	SSH	Zombies
Werserver1 IP:81.169.130.84	✓	✓	✓	✗	✗	✓	✓	✗	✗
Webserver2 IP:81.169.130.75	✓	✗	✗	✓	✗	✓	✓	✓	✓
Webserver3 IP:80.237.158.20	✗	✗	✗	✓	✓	✓	✗	✓	✓
Officeserver IP:89.27.255.104	✗	✗	✗	✓	✓	✓	✗	✓	✓
Streamingserver IP:85.214.21.141	✗	✗	✗	✓	✓	✓	✓	✗	✓

7 Realisierung

Nagios wird auf dem zweiten Webserver installiert, auf dem die Hersteller ihre Videos hochladen. Dort sind genug Ressourcen frei, da dort ausschließlich Datentransfer stattfindet. Dieses war keine Entscheidung, sondern eine Vorgabe.

7.1 Anlegen des Users und der Gruppe „nagios“

Nagios benötigt einen User unter der später der Prozess ausgeführt werden kann. Der Benutzer soll nagios heißen, die Gruppe erhält den gleich Namen. Der User nagios hat die UID 103 und die Gruppe nagios hat die GID 104. Das Home-Verzeichnis für den neuen User wird auf `/var/run/nagios2` festgesetzt. Das Homeverzeichnis wird mit `mkdir /home/nagios` angelegt.

7.2 Kompilieren und Installieren des Quellcodes

Verwendet wird die Version 2.11, die man auf der Homepage von Nagios(<http://www.nagios.org>) kostenfrei herunterladen kann. Nach dem Herunterladen muss das Archiv noch mit dem Befehl: `tar xfvz nagios-2.11.tar.gz` entpackt werden. Durch Ausführen des Shell-Skriptes `configure` werden Systeminformationen gesammelt und dem Compiler übergeben. Zusätzlich werden noch folgende Parameter übergeben:

Parameter	Wert	Beschreibung
-with-htmlurl	/nagios2	Die URL unter der später die HTML Seiten liegen
-with-cgiurl	/usr/lib/cgi-bin/nagios2	Pfad zu den CGI-Programmen für das Webinterface
-prefix	/var/run/nagios2	Installationspfad
-with-nagios-user	nagios	User unter dem die Prozesse ausgeführt werden sollen
-with-nagios-group	nagios	Gruppenkennung unter der die Nagiosprozesse ausgeführt werden sollen
-with-openssl	/usr/bin/openssl	Pfad zur Installation von OpenSSL

Nach dem Aufruf von `configure` startet man das Übersetzen des Quelltextes (kompilieren) mit dem Befehl `make`. Zum Installieren in das Homeverzeichnis führt man `make install` aus.

Nach der Installation wird der Username und das Passwort mittels `htpasswd -c /etc/nagios2/htpasswd.users nagiosadmin` gesetzt. Nach der Bestätigung dieses Kommandos wird man aufgefordert ein Passwort zu wählen. Jetzt muss in der Datei `/etc/nagios2/cgi.cfg` das Flag `use_authentication=0` auf 1 gesetzt werden, damit Nagios nur authentifizierte Benutzer zulässt.

Zum Abschluss der Installation wird Nagios neu gestartet (`/etc/init.d/nagios2 restart`) und im Browser unter der Adresse [WEBADRESSE](#) aufgerufen. Mit dem erfolgreichen Aufruf der Nagiosoberfläche ist die Installation abgeschlossen.

7.3 Installation der Nagios-Plugins

Damit Nagios überhaupt in der Lage ist, Dienste und Ressourcen zu überprüfen, müssen die verschiedenen Plugins installiert werden. Das Kernprogramm liefert die Funktionalität, um diese Plugins aufzurufen, auszuwerten und zu visualisieren. Die Plugins können ebenfalls unter www.nagios.org heruntergeladen werden. Zum Vorbereiten der Installation wird das Script `configure` ausgeführt. Zum Kompilieren und Installieren müssen die Befehle `make` und `make install` in der Shell ausgeführt werden. Beim Ausführen von `configure` kamen die Warnungen, dass der Radius und SNMP Plugin nicht installiert werden können, da benötigte Pakete nicht vorhanden sind. Diese Meldung wurde aber nicht weiter beachtet, da SNMP und Radius im Betrieb nicht eingesetzt werden.

Da die Plugins alles eigenständige Programme darstellen, kann man sie ohne weiteres durch das Aufrufen in der Konsole testen. Zum Test der Installation wird das Plugin `check_icmp` ausgeführt.

Durch den Befehl `cd /usr/lib/nagios/plugins` wechselt man in das Verzeichnis in dem die einzelnen Plugins liegen. Zum Testen wird der Befehl `./check_icmp -H IPADRESSE` ausgeführt. Dieses Plugin führt ein Ping aus. Man erhält folgendes Ausgabe:

```
OK - 85.214.21.141: rta 0.341ms, lost
0%|rta=0.341ms;200.00;500.00;0; pl=0%;40;80;;
```

Die Ausgabe der Plugins erfolgt auf einer Zeile. Nagios wertet nur die ersten 300Byte der jeweiligen Ausgabe aus, der Rest der Ausgabe wird ignoriert. Alle weiteren benötigten Plugins werden auf die gleiche Art und Weise auf Ihre Funktionalität überprüft.

7.4 Installation und Konfiguration von *nrpe* auf den zu überwachenden Servern

Der Nagios Remote Plugin Executor(nrpe) wird benötigt, um lokale Ressourcen wie zum Beispiel die freie Festplattenkapazität und die CPU Auslastung zu überprüfen. Der *nrpe* führt aber nicht diese Überprüfungen selbst aus, sondern führt die lokalen Plugins auf dem System aus und übermittelt deren Ausgabe. Daher müssen diese Plugins auf dem Remote Rechner auch installiert sein.

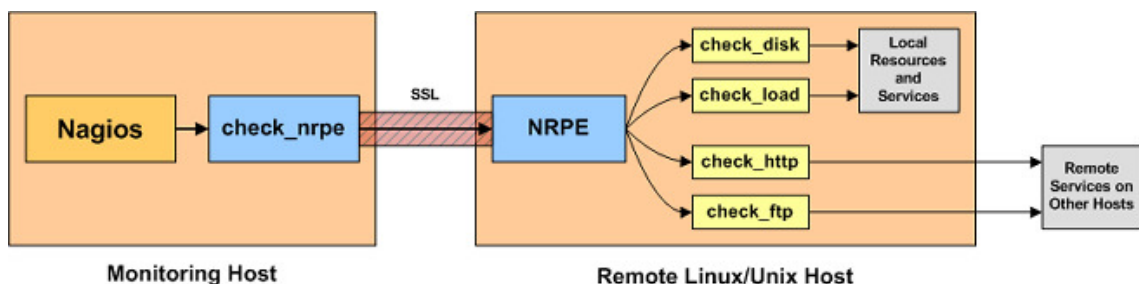


Abbildung 2: Funktionsweise vom Nagios Remote Plugin Executor

Auf den Servern wird jeweils der Befehl `apt-get install nagios-nrpe-server` ausgeführt. Durch diesen Befehl wird der NRPE-Dienst und die Plugins automatisch heruntergeladen und installiert. Der Vorteil von der Installation mittels `apt-get` ist, dass abhängige Packet ebenfalls installiert werden. In diesem Fall werden die Plugins mit installiert.

Es werden die gleichen Tests durchgeführt wie bei der Installation der Plugins auf dem Nagiosserver um sicherzustellen, dass die Installation erfolgreich abgeschlossen wurde. In der Konfigurationsdatei vom *nrpe* (`etc/nagios/nrpe.cfg`) muss eingestellt werden, welche lokalen Plugins vom *nrpe* ausgeführt werden dürfen. Die Syntax eines solchen Eintrages ist:

```
command[check_load]=/usr/lib/nagios/plugins/check_load
-w 15,10,5 -c 30,25,20
```

Diesen definierten Befehl kann man vom Nagiosserver nun über den Plugin `check_nrpe` aufrufen. Ein Testaufruf vom Nagiosserver sieht wie folgt aus:

```
check_nrpe -H 85.214.21.141 -c check_load
```

Man erhält folgende Ausgabe auf dem Nagiosserver:

```
OK - load average:0.00, 0.00,  
0.02|load1=0.00;15.000;30.000;0;  
load5=0.000;10.000;25.000;0;  
load15=0.020;5.000;20.000;0;
```

Mit dieser Ausgabe wird der Test erfolgreich beendet. Auf den Servern noch in der *nrpe.cfg* der Wert vom Flag `allowed_hosts` auf die IP Adresse des Nagiosserver gesetzt werden. So werden nur Anfragen von der IP des Nagiosserver verarbeitet.

7.5 Konfiguration von Nagios

Nach der Installation hat Nagios alle Konfigurationsdateien ohne Inhalt angelegt. Es sind aber Sample-Dateien dabei, die kopiert werden und anschließend abgeändert.

```
z.b. cp nagios.cfg-sample nagios.cfg
```

Hauptkonfigurationsdatei *nagios.cfg*

Die *nagios.cfg* ist die zentrale Konfigurationsdatei von Nagios. Diese Datei verweist auf alle weiteren Konfigurationsdateien, mit Ausnahme der CGI-Konfiguration. In der *nagios.cfg* werden die einzelnen Dateien durch `cfg_file=` festgelegt. Diese Zeilen werden ersetzt durch die Zeile `cfg_dir=/etc/nagios2/conf.d`. Dieses Verzeichnis wird von Nagios rekursiv durchsucht nach allen Dateien mit der Endung *.cfg*. Diese Änderungen macht die spätere Verwaltung einfacher.

Command-Konfigurationsdatei *commands.cfg*

In dieser Datei sind alle Kommandos. Alles, was Nagios macht, wird in dieser Datei definiert. Auch hier wird die Sampledatei kopiert, da die Sampledatei recht umfangreich ist.

Es wird ein Kommando hinzugefügt. Es ist das Kommando um SMS zu verschicken. Da die Firma bereits ein Programm installiert hat um SMS zu verschicken, wird auf dieses lokale Programm auch zurück gegriffen.

CGI-Konfigurationsdatei *cgi.cfg*

Diese Datei ist für das Web Frontend zuständig. Hier wird auch wieder die Sampledatei kopiert, da in dieser Datei bereits die richtigen Pfade gesetzt wurden. Zugriffsrechte können hier bestimmt werden, momentan ist aber nur ein User vorhanden, also bekommt dieser User Zugriff auf alle CGI-Programme des Web Frontends.

Ressourcen-Konfigurationsdatei *resource.cfg*

In dieser Datei können Makros definiert werden. Nach dem Kopieren der Sampledatei ist ein Makro vorhanden:

```
$USER1$=/usr/lib/nagios/plugins  
es ist der Pfad zu den Plugins.
```

Objekt-Konfigurationsdateien *conf.d/*.cfg*

Nagios hat viele Verschiedene Objekte. Es können unter anderem Hosts, Services, Kontakte, Gruppen oder auch Kommandos sein. Ein wichtiges Feature hierbei ist die Möglichkeit der Vererbung. Die Prinzipelle Syntax ist:

```
define object-type {  
    parameter wert  
    parameter wert  
    ...}
```

Die verschiedenen Objekte werden in die *.cfg Dateien eingetragen. Es können Vorlagen (Templates) erstellt werden, deren Eigenschaft vererbt werden können. Beispiele für die Definition sind im Anhang B zu finden.

Nach dem Anpassen der verschiedenen Konfigurationsdateien und Eintragen aller Daten wird die Nagioskonfiguration mit dem Befehl `nagios2 -v nagios.cfg` überprüft. Dieser Befehl überprüft die komplette Konfiguration. Wenn die Konfiguration korrekt ausgeführt wurde, erhält man in der letzten Zeile der Ausgabe folgende Meldung:

```
Things look okay - No serious problems were detected  
during the pre-flight check
```

Nagios wird anschließend neugestartet durch Aufrufen von `/etc/init.d/nagios2 restart`.

8 Qualitätssicherung

Im Folgenden werden die Schritte beschrieben die zur Qualitätssicherung nötig sind.

8.1 Grundsätzliche Überprüfung der Konfiguration

Zur grundsätzlichen Überprüfung der Konfiguration wird das Web Frontend über den Browser aufgerufen. Nach der Eingabe von Benutzernamen und Passwort, erhält man die Nagiosstartseite. Über den Menüpunkt „Host Detail“ erhält man eine Übersicht aller Hosts. Es wurden alle 5 Host eingetragen. Durch Klicken auf „Service Detail“ kann man die ausgegebene Liste mit der Überwachungsliste vergleichen. Es werden alle gewünschten Services überwacht. Nach dem Aufrufen und der Kontrolle der Ausgabe der einzelnen CGI-Scripte werden diese Tests positiv abgeschlossen.

8.2 Abschließende Tests

TestszENARIO	Durchführung	erwartetes Ergebnis	tatsächliches Ergebnis
Officeserver ist nicht via Ping erreichbar	Die Firewall am Server wurde so konfiguriert, dass keine Pings beantwortet werden	E-Mail an den zuständigen Mitarbeiter	Administrator wurde via E-Mail benachrichtigt.
Kein Speicherplatz verfügbar für Hersteller, um Produktvideos hochzuladen	Verfügbarer Speicherplatz wurde kurzzeitig gleich gesetzt mit der aktuellen Datenmenge	Benachrichtigung des zuständigen freien Mitarbeiters, besteht das Problem länger als 24 Stunden muss der Administrator benachrichtigt werden.	Freier Mitarbeiter wurde benachrichtigt; Test konnte keine 24 Stunden ausgeführt werden, um die zweite Benachrichtigungsstufe zu testen.
FTP Zugang auf dem Webserver nicht möglich	Port 21 wird per Firewall geblockt	Ab einer Ausfallszeit von länger als 2 Stunden wird der Auszubildende benachrichtigt; besteht das Problem länger als 8 Stunden wird der Administrator benachrichtigt.	Nach 2 Stunden hat der Auszubildende eine E-Mail erhalten; nach 8 Stunden hat der Administrator eine SMS bekommen, da es ein Samstag war.
Fehler beim Aufrufen des Streamings	Die Datei zum Teststeaming wurde auf dem Server umbenannt	E-Mail an den freien Mitarbeiter, SMS an den Administrator und an die Geschäftsführung, wenn der Fehler länger als 30min besteht.	Alle bis auf die Geschäftsführung wurden korrekt benachrichtigt. Fehler: falsche Handynummer eingetragen.

Anhang A

Glossar

CGI-Programm/Script

Programm, das auf das Common User Interface zurückgreift, um Webseiten dynamisch zu gestalten. In Nagios werden die Statusberichte im Web Frontend mittels CGI-Programme erstellt.

GID

Group identification (Gruppen Identifikation) - ist eine einmalige im Unixsystem für eine Gruppe vergebene Identifikationsnummer.

NRPE

Nagios Remote Plugin Executor – ermöglicht Nagios Plugins auf entfernten Rechner auszuführen und sendet deren Ausgabe an den Nagiosserver.

Makro

Liefert nach dem Aufruf einen bestimmten Wert zurück.

OTRS

Open Ticket Request System – ist ein Trouble Ticket System

Request

Aus Sicht der Firma- die Anfrage eines Kunden(Shops) nach einem Produktvideo

Sampledatei

Dateien, die bei der Installation mit kopiert werden und beispielhafte Konfigurationen enthalten.

SNMP

Simple Network Management Protocol – ist ein Protokoll, das es ermöglicht, Netzwerkgeräte zu überwachen

Streaming

Aus Sicht der Firma – das Ausliefern eines Produktvideos an einen Shop.

UID

User identification(User Identifikation) - ist eine einmalige im Unixsystem für einen User vergebene Identifikationsnummer.

Anhang B

Konfigurationsdateien

contacts_nagios2.cfg

In dieser Datei werden die verschiedenen Kontakte definiert. Alle freien Mitarbeiter, Administratoren und die Geschäftsführung werden in diese Datei in folgender Form eingetragen:

```
define contact{
    contact_name          ssschwarz
    alias                 Sascha Schwarz
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    service_notification_commands notify_by_email
    host_notification_commands host-notify-by-sms
    email                E-MAIL@E-MAIL.E\_MAIL
    pager                +49 (NUMBER)
}
```

In diesem Beispiel wird der Kontakt ssschwarz angelegt. Er wird 24 Stunden und 7 Tage die Woche benachrichtigt, bei allen Meldung.

Ebenfalls werden 3 Gruppen angelegt: Geschäftsführung, freie Mitarbeiter, Administratoren.

commands.cfg

In dieser Datei werden Kommandos definiert. Eine Definition sieht wie folgt aus:

```
define command{
    command_name          host-notify-by-sms
    command_line          /usr/bin/printf "%b"
                        "Host '$HOSTALIAS$' is
                        $HOSTSTATE$\nInfo:
                        $HOSTOUTPUT$\nTime:
                        $LONGDATETIME$" |
                        /usr/bin/mmsms -n
                        $CONTACTPAGER$
}
```

Es wird das Kommando host-notify-by-sms erzeugt, das Kommando greift auf den Befehl /usr/bin/mmsms zurück.

host-gateway_nagios2.cfg

In dieser Datei müssen alle Host eingetragen werden. Es ist aber möglich, einen Generic-Host anzulegen, der dann als Vorlage für alle weiteren Host fungieren kann.

Generic-Host:

```

define host {
    name                generic-host
    notifications_enabled 1
    event_handler_enabled 1
    flap_detection_enabled 1
    failure_prediction_enabled 1
    process_perf_data 1
    retain_status_information 1
    retain_nonstatus_information 1
    check_command        check-host-alive
    max_check_attempts   10
    notification_interval 0
    notification_period   24x7
    notification_options  d,u,r
    contact_groups        admins
    register              0
}

```

Ein Host wird mit dem Eintrag:

```

define host {
    host_name            webserver1
    alias                server.de
    address              IPADRESSE
    use                  generic-host
}

```

definiert. Es wird hier der erste Webserver angelegt, der die IP Adresse IPADRESSE hat. Dieser Server benutzt generic-host als Vorlage.

hostgroups_nagios2.cfg

In dieser Datei werden die die Hosts gruppiert. Beispieleintrag:

```

define hostgroup {
    hostgroup_name      streaming
    alias               Streaming-Server
    members              streamingserver
}

```

Hier wird die Gruppe für die Streaming-Server angelegt, in der je nach Bedarf weitere Server hinzugefügt werden können

services_nagios2.cfg

In dieser Datei werden alle Services definiert. Ein Service ist ein Check eines Dienstes oder einer Ressource.

Als ersten kann man auch hier wieder den generic-service definieren, der später als Vorlage für weite Definitionen dienen soll.

```
define service {
    name                       generic-service
    active_checks_enabled     1
    passive_checks_enabled    1
    parallelize_check         1
    obsess_over_service       1
    check_freshness           0
    notifications_enabled     1
    event_handler_enabled     1
    flap_detection_enabled    1
    failure_prediction_enabled 1
    process_perf_data         1
    retain_status_information  1
    retain_nonstatus_information 1
    notification_interval     0
    is_volatile               0
    check_period              24x7
    normal_check_interval     5
    retry_check_interval      1
    max_check_attempts        4
    notification_options      w,u,c,r
    contact_group             admins
    register                   0
}
```

Die unterschiedlichen Services, die benötigt werden, können anschließend wie folgt definiert werden.

```
define service {
    hostgroup_name            streaming
    service_description       Streaming-Load
    check_command             check_nrpe!check_disk
                             !10% 5% /var
    use                       generic-service
}
```

In diesem Beispiel wird ein Check auf allen Streamingservern über den nrpe lokal ausgeführt.

timeperiods_nagios2.cfg

In dieser Datei können Zeitspannen definiert werden. In diesem Beispiel definiert man die Arbeitszeit der Administratoren:

```
define timeperiod {
    timeperiod_name          workhours
    alias                    Arbeitszeit der Admins
    monday                  08:00-16:00
    tuesday                 08:00-16:00
    wednesday              08:00-16:00
    thursday               08:00-16:00
    Friday                 08:00-16:00
}
```

Die tägliche Arbeitszeit der Administratoren wird hier von Montags bis Freitags von 08:00 – 16:00 festgelegt.

Anhang C

Übersicht der Server

ÜBERSICHTSBILD*

Server	IP	Aufgabe

Anhang D

Quellen

Literatur:

Linux Magazin – Technical Review 02
Wolfgang Barth – Nagios System- und Netzwerk-Monitoring

Webseiten:

<http://www.nagios-portal.de>
<http://entwickler.de>
<http://www.nagios-portal.de>
<http://www.nagios-wiki.de>
<http://wikipedia.org>
<http://www.nagios.org/>