

# Professional Monitoring

## Nagios nur starten reicht nicht

**Hendrik Bäcker**

(hbaecker@baecker-its.de)

www.baecker-its.de

28. Mai

LinuxTag 2008

# Themenübersicht

- 1 Früher war alles noch in Ordnung
- 2 Nagios Konfiguration
- 3 Eskalationen
- 4 Human Ressources
- 5 Baselining
- 6 Meldewege



Aller Anfang ist leicht!

## In der Regel...

- ... startet Nagios im kleinen Kämmerlein.
- ... deckt es Anfangs nur einen Teilbereich ab (Netzwerk oder Webserver).
- ... ist es die "Spielwiese" des Admins/Azubis.
- ... will man zumindest mal angucken bevor man Geld ausgibt.



Aller Anfang ist leicht!

## Aber was, wenn...

- ... es der Organisation gefällt?
- ... es mächtiger wird, als man Anfangs dachte?
- ... es quasi über Nacht zu **dem** Monitoringsystem wird?
- ... es nun nicht nur Webserver, sondern einfach alles Überwachen soll?

## Nagios kann:

- Zwischen Webserver, Switch und Router unterscheiden
- Netzwerkausfälle erkennen
- Auf **Ihre** Anforderungen reagieren
- Beliebter Community Spruch: “Alles was **SIE** können”



- 1 Früher war alles noch in Ordnung
- 2 Nagios Konfiguration**
- 3 Eskalationen
- 4 Human Ressources
- 5 Baselining
- 6 Meldewege

## Am Anfang war das Chaos...

- kleiner Test = kleine Config
- große Umgebung = Chaos???

## Übersicht schaffen mit Templates

```
define host{
    host_name          bogus-router
    alias              Bogus Router #1
    address            192.168.1.254
    parents            server-backbone

    check_command      check-host-alive
    check_interval     5
    retry_interval     1
    max_check_attempts 5
    check_period       24x7

    process_perf_data  1
    retain_nonstatus_information 0

    contact_groups     router-admins
    notification_interval 30
    notification_period 24x7
    notification_options d,u,r
}
```

## Templates 1

```
define host{
    name                    host_check_template_24x7
    check_command           check-host-alive
    check_interval         5
    retry_interval         1
    max_check_attempts     5
    check_period           24x7
    register                0
}
```

```
define host{
    name                    host_contact_routeradmins
    contact_groups         router-admins
    notification_interval  30
    notification_period    24x7
    notification_options   d,u,r
    register                0
}
```



## Templates 2

```
define host{
    name                               enable_perfddata
    process_perf_data                  1
    register                            0
}
```

## Voila!

```
define host{
    host_name          bogus-router
    alias              Bogus Router #1
    address            192.168.1.254
    parents            server-backbone

    use                enable_perfdata, \
host_check_template_24x7, \
host_contact_routeradmins
}
```



## Fazit: Templates

- Konfigurationsaufwand wird vermindert
- Übersicht schaffen durch Vereinheitlichung
- Sonderfälle nach wie vor möglich!  
(Bsp. Host Definition überschreibt Template)



- 1 Früher war alles noch in Ordnung
- 2 Nagios Konfiguration
- 3 Eskalationen**
- 4 Human Resources
- 5 Baselining
- 6 Meldewege



Eskalationen: Was tun, wenn es brennt?

## Übliche Unternehmensstruktur:

- 1 First Level Support
- 2 Second Level Support
- 3 (Third Level Support)
- 4 Leitungsebene
- 5 Geschäftsführung

## Eskalationen - Teil 1

### Default:

Nagios informiert das Helpdesk

Aber: **Helpdesk wird von Kunden überrannt!**

### Level 2:

Störung wurde **nicht** quittiert.

Start der Eskalation => Information an Second Level

## Eskalationen - Teil 1

### Default:

Nagios informiert das Helpdesk

Aber: **Helpdesk wird von Kunden überrannt!**

### Level 2:

Störung wurde **nicht** quittiert.

Start der Eskalation => Information an Second Level

## Eskalationen - Teil 2

### Level 3:

Störung wurde **noch immer nicht** quittiert  
Second Level aber bereits mit Entstörung beschäftigt!?!

### Level 4:

Wenn's denn so wichtig ist:  
Information an die Leitungsebene



## Eskalationen - Teil 2

### Level 3:

Störung wurde **noch immer nicht** quittiert  
Second Level aber bereits mit Entstörung beschäftigt!?!

### Level 4:

Wenn's denn so wichtig ist:  
Information an die Leitungsebene

## Beispiel: Eskalationen

**Standard** Zwischen 7.00 und 17.00 versendet Nagios eine E-Mail an 1st Level.

Zwischen 17.00 und 07.00 geht der Alert ans Operating (24h Erreichbar)

Alarmintervall: 60 Minuten

**Eskalation 1** Sollte die Störung weiter bestehen:  
E-Mail Alert zusätzlich an die Fachabteilung + SMS an Fachverantwortlichen + Neues Alarmintervall: 30 Minuten

**Eskalation 2** E-Mail an 1st Level/Operating + E-Mail an Fachabteilung + **Anruf** bei Fachverantwortlichen + **Anruf** beim Projektleiter + Neues Alarmintervall: 15 Minuten

## Fazit: Eskalationen

- Es gibt Tag für Tag Fehlersituationen.  
Die einen haben mehr Auswirkungen auf den Betrieb, die anderen weniger.
- Nagios **kann** dafür sorgen, dass alle Parteien informiert werden.
- Während einer Eskalation sind "Breakouts" aus den normalen Intervallen und Alarmierungsverfahren möglich.  
(Bsp: Statt 60 Minuten, nun 15 Minuten Interval oder SMS statt E-Mail Alarm)



- 1 Früher war alles noch in Ordnung
- 2 Nagios Konfiguration
- 3 Eskalationen
- 4 Human Ressources**
- 5 Baselining
- 6 Meldewege



## Merke:

Das Überwachungssystem ist nur so gut,  
wie die Mitarbeiter die daran arbeiten.

## Regeln für den Nagios Admin

- 1 Fehlalarme sind zu vermindern => Vertrauen
- 2 Abhängigkeiten von der Infrastruktur:  
(Warum 30 Webserver als defekt melden, wenn der Switch davor ein Problem hat?)
- 3 Überblick bewahren:  
Bei 100 Services hat man noch den Überblick.  
Was ist bei 500? 1.000? 10.000?
- 4 **Vertrauensbasis zum Monitoringsystem schaffen**



## Motivation

- Ein Überwachungssystem das hält was es verspricht
- Schneller Zugriff auf aktuelle Probleme
- Zeitnahe Reaktionsmöglichkeit
- Grafische Langzeitauswertungen
- schnelle und einfache Reports



## Störungen am Fließband

Man kennt es von Produktionsstraßen und Fließbandarbeit:  
Eine Störung tritt auf, das Fließband steht still und eine Hupe verkündet laut schallend einen Fehler.

Nagios 'hupt' geduldig bis:

- Eine Störung sich von selbst erledigt
- Die Hupe einfach 'abgeschaltet' wird
- Ein Wartungsfenster definiert wird
- Die Störung **quitiert** wird



## Störungen am Fließband

Man kennt es von Produktionsstraßen und Fließbandarbeit:  
Eine Störung tritt auf, das Fließband steht still und eine Hupe verkündet laut schallend einen Fehler.

## Nagios 'hupt' geduldig bis:

- Eine Störung sich von selbst erledigt
- Die Hupe einfach 'abgeschaltet' wird
- Ein Wartungsfenster definiert wird
- **Die Störung quittiert wird**



## Acknowledgements

- Verhindern erneute Alarmierung
- Sind im Webinterface gekennzeichnet
- Zeigen an, dass die Entstörung eingeleitet wurde



## Wartungsfenster

Der Feind jeder SLA...



## Downtimes

- verhindern bereits die erste Alarmierung
- werden im Webinterface entsprechend dargestellt
- können mit Kommentaren zum Wartungsfenster versehen werden
- werden in den Reports gesondert behandelt
- sollten aber im Vorfeld schon bekannt sein!

## Fazit: Human Resources

- Man wird Nagios nicht ernst nehmen, wenn die Überwachung der Infrastruktur nicht konsistent ist.
- Nagios verliert die Glaubwürdigkeit, wenn es ständig Fehler anzeigt die bereits bearbeitet werden.
- Die Damen und Herren des First Levels werden sich freudig über volle Postfächer bedanken, wenn die Kollegen aus dem Netzwerkbereich 'mal eben' die Core Switches aufgrund eines Upgrades booten mussten...

## Kann auch teuer werden...

Eine fehlende Downtime für eine dringende Netzwerkwartung löst eine Alarmierungsflut aus.

Nagios verschickt, konsequent und fleißig, für alle 500 betroffenen Systeme eine SMS - **Whoops!**



- 1 Früher war alles noch in Ordnung
- 2 Nagios Konfiguration
- 3 Eskalationen
- 4 Human Resources
- 5 Baselining**
- 6 Meldewege

## Von Snapshots zur Einsicht:

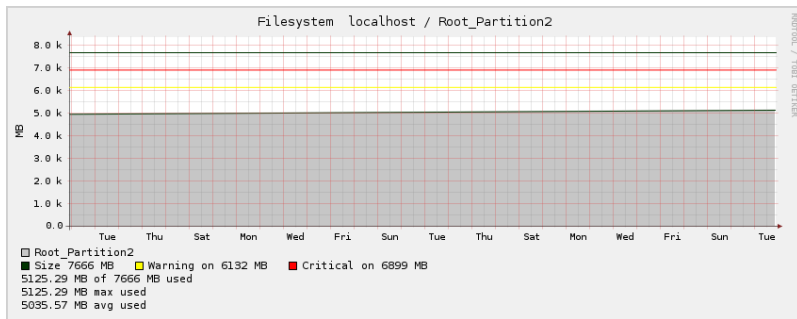
- Prima!
- Nagios kann etwas Überwachen
- Nagios kann auch E-Mails senden
- Nagios kann sogar prima eskalieren
- Aber irgendwie ist es doch nur eine Momentaufnahme...

## Von Snapshots zur Einsicht:

- Prima!
- Nagios kann etwas Überwachen
- Nagios kann auch E-Mails senden
- Nagios kann sogar prima eskalieren
- **Aber irgendwie ist es doch nur eine Momentaufnahme...**

## Frage:

Nagios meldet die verbleibende freie Kapazität einer Festplatte als kritisch:  
Normal oder nicht?



**Figure:** PNP4Nagios zeigt einen leichten Anstieg über 3 Wochen (check\_disk)

Wen kümmerts?

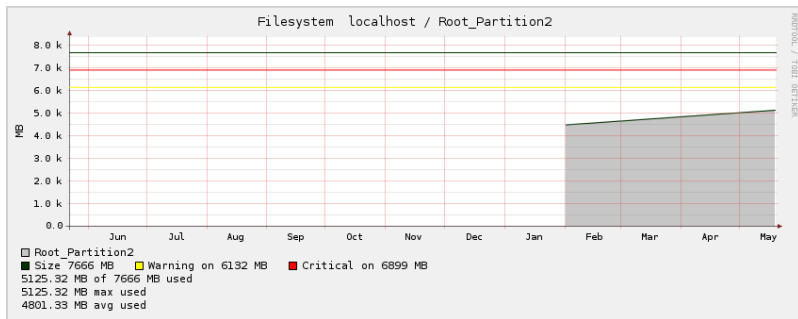


Figure: Über 4 Monate sieht es schon anders aus...

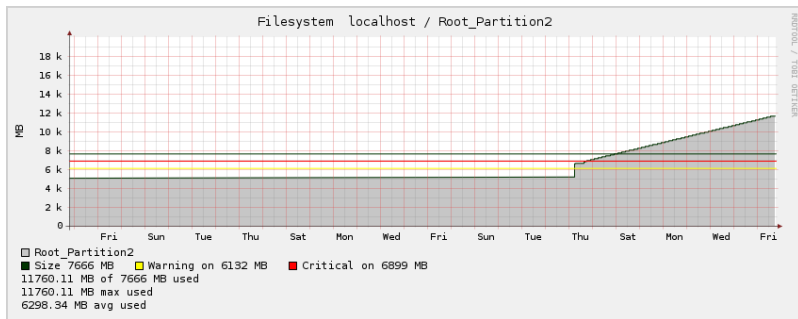


Figure: Ohne Worte...

## Anwendungsbeispiele:

**Festplatten** Wie groß ist der Datenzuwachs und wann wird der Platz eng?

**CPU** Zu welcher Tageszeit hat das System IO Waits?

**Traffic** MRTG/Cacti inkl. Nagios Alarmierung

**HTTP** Größe und Antwortzeitverhalten meiner Webseite

**Logfiles** Warum tritt mein Oracle Fehler immer nur um Mitternacht auf? Und warum zum Henker trifft das auf die Sicherheitszeit?

## Fazit: Baselining

- Viele Nagios Plugins liefern Performancedaten, die graphisch dargestellt werden können
- Langzeitgraphen lassen das 'Grundrauschen' und somit 'Ausreißer' erkennen
- Graphen lassen Rückschlüsse auf Änderungen schließen
- Sind sehr beliebt auf der Managementebene



- 1 Früher war alles noch in Ordnung
- 2 Nagios Konfiguration
- 3 Eskalationen
- 4 Human Resources
- 5 Baselining
- 6 Meldewege**



Melden macht frei...

... und belastet den Vorgesetzten.

## Alarmierungsverfahren:

**E-Mail** Standard! Asynchrones Medium

Was aber bei Ausfall des Mailservers?

**Ticket System** Warum nicht? Shell, Perl & Co bringen uns fast überall hin.

**SMS** Möglich - GSM Modem, Mail2SMS Dienste, Webgateways, ISDN Karten

**Chats** Beliebt - Alerts über Instant Messaging, warum nicht auch Intelligent?  
(Interaktion via Chatbots)

**Telefon** Immer erreichbar: Bsp. Asterisk + ISDN und die Welt steht uns offen

**Allgemein** Was eine Schnittstelle hat, lässt sich nutzen!

# Summary

- Nagios skaliert gut auch in großen Umgebungen - Tuningwissen erforderlich
- Nur auf Einzelchecks zu achten verschleiert den Gesamteindruck
- Nagios bietet viele Möglichkeiten die sich vernünftig nutzen lassen - man muss es nur tun!
- Auch wenn OpenSource - Supportangebote in Deutschland nehmen zu
- "Germany has a wonderful strong community"  
(Zitat: Ethan Galstad)



# Fragen?

Danke  
und viel Spaß auf dem  
LinuxTag 2008 in Berlin  
:o)