



Qualitätssicherung mit Hilfe von Nagios

LOS 11.05.2009

Eckart Fellner

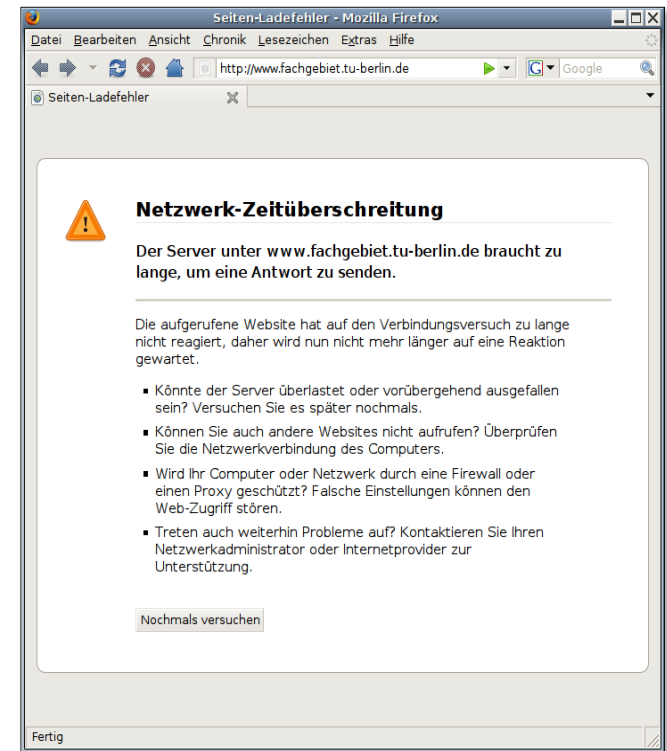
IT Dienstleistungszentrum der TU Berlin

- Motivation
- Nagios
 - Vor- und Nachteile
 - Objekte
 - Funktionsweise
- Ausblick

Vormittag, 10 Uhr

Der Chef tobt ...

... seine Präsentation liegt auf dem
WWW-Server, der nicht antwortet ...



Also wird die Hotline angerufen,
die Hotline fragt den Webmaster.



Der WWW-Server funktioniert tadellos.

Also liegt es wohl am Netz?

Oder am AFS, LDAP, Kerberos?

Die Suche dauert länger als die Lösung.

Wie können wir die Suche und die Ausfallzeit
verkürzen?

Nagios® ist ein einfaches, aber sehr flexibles
Werkzeug zum Testen von Services und
Rechnern ('hosts').

- Nagios ist einfach
 - kennt nur wenige Zustände

Host Status Totals				Service Status Totals				
Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
521	4	0	7	944	1	7	5	0

- flexibel
 - sehr einfache Integration beliebiger Tests (alles, was programmiert werden kann, kann auch getestet werden).
- mächtige Test-Logik
 - Abbildung von Abhängigkeiten und Eskalationsstrategien
 - Problemerkennung im Vorfeld möglich

Bei einer komplexen IT-Landschaft

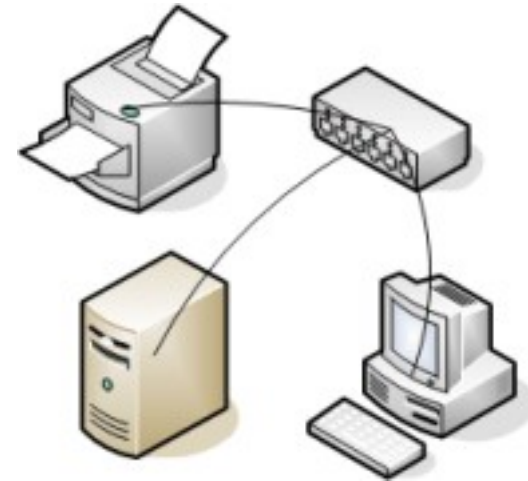
- sind Konfiguration und Pflege aufwändig
- besteht die Gefahr der Überflutung mit Status-Informationen
- Nagios „weiß“ nicht, was getestet wird und hat dementsprechend auch keine Logik, die Ergebnisse zu interpretieren

Nagios kennt im Wesentlichen folgende Objekte:

- Hosts und Services
- Contacts und Timeperiods
- Plugins und Commands

Hosts sind ein zentrales Objekt von Nagios mit folgenden wesentlichen Attributen:

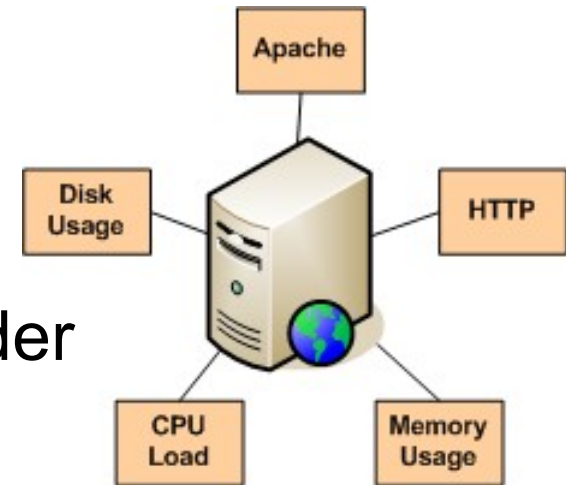
- Hosts sind typischerweise Geräte wie Server, PCs, Drucker, Router
- Hosts bedienen Services
- Sie können in einer Beziehung zueinander stehen



Hosts können in *Hostgroups* zusammengefasst werden.

Services sind an Hosts gebunden und können folgende Eigenschaften haben:

- Eigenschaften von Hosts, wie z.B. CPU-Auslastung
- Services des Hosts, wie z.B. Web- oder Maildienste



Services können in *Servicegroups* zusammengefasst werden.

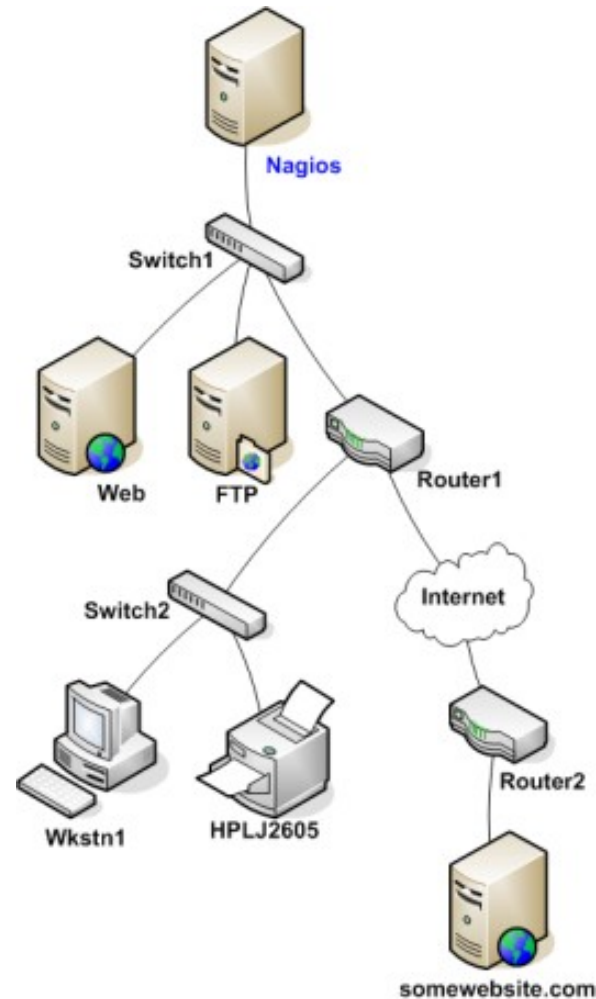
In *Contacts* wird definiert, wer über bestimmte Ereignisse, z.B. via E-Mail oder SMS, informiert werden soll.



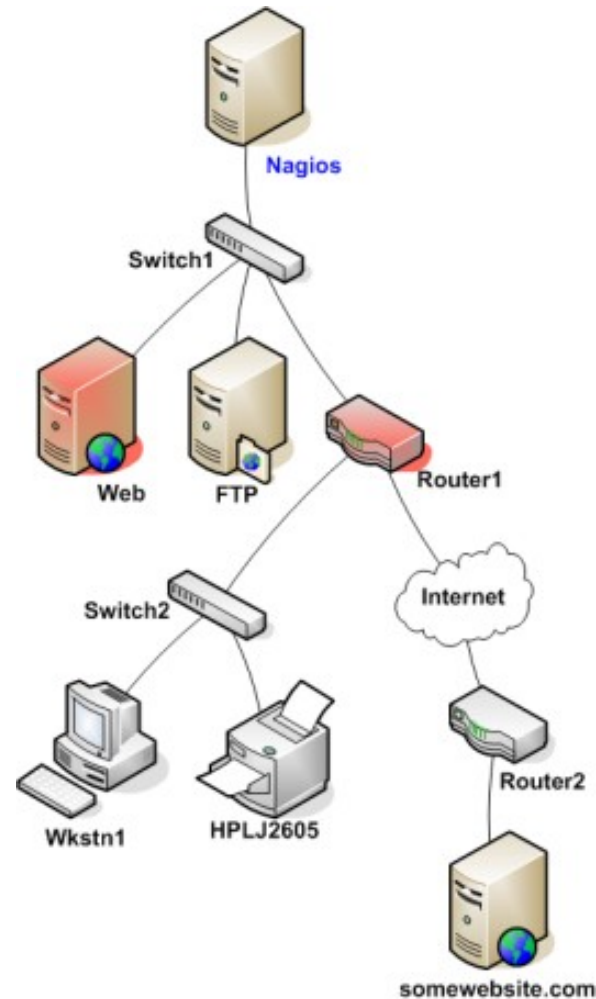
In *Timeperiods* wird definiert, wann die Tests durchgeführt und zu welcher Zeit Nachrichten versandt werden sollen.

- *Plugins* sind ganz normale Programme in beliebiger Programmiersprache.
- In den *Commands* wird die Nutzung von *Plugins* definiert für
 - Host- und Service-Tests
 - den Versand von Benachrichtigungen
 - die Behandlung von Ereignissen

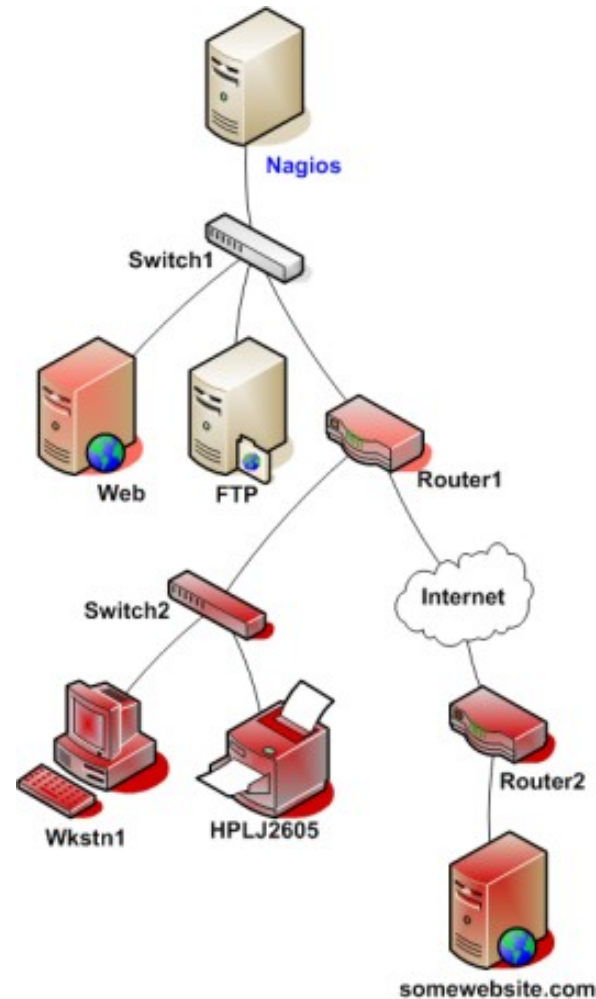
In Nagios können Abhängigkeiten von Hosts und Services definiert werden.



Fallen einzelne
Hosts oder
Services aus, z.B.
Server *Web* und
Router1 ...



... dann erkennt die Logik von Nagios selbständig die Nichterreichbarkeit der „dahinter“ liegenden Hosts und Services.



Derzeit werden

- 540 Hosts (davon 470 Netzwerkgeräte) und
- 976 Services

mit Nagios überwacht.

Der Zugang erfolgt für registrierte Nutzer mit dem tubIT-Account.

- Abbildung komplexer Abhängigkeiten
- Abbildung einer Eskalationsstrategie
- Abbildung von Geschäftsprozessen
- Verbesserung der Web-Oberfläche

Habe ich Ihr Interesse geweckt?
Nagios können Sie live erleben:

<https://nag-2.tubit.tu-berlin.de/nagios/>

Accounts:

gast

fio
