
NAME

nagios-check.pl - Eine oder mehrere Nagios-Instanzen überprüfen, und den Status zusammenfassen.

BESCHREIBUNG

Dieses Script meldet sich auf der Weboberfläche der Nagios Instanzen an und untersucht diese auf das Vorhandensein von Services mit dem Status *CRITICAL*, *UNKNOWN*, *WARNING* und Hosts mit dem Status *DOWN* oder *UNREACHABLE*. Ist auf einer aller definierten Instanzen ein Host oder ein Service mit einem erkannten Status, so wird das gemeldet.

HINWEIS: Dieses Script wurde mit einem Nagios 3.0.6 entwickelt und getestet. Die Funktion kann für andere Versionen nicht garantiert werden!

KONFIGURATION

Die Konfiguration erfolgt über die Konfigurationsdatei 'nagios-check.yml', welche in dem selben Ordner wie das Script selbst erwartet wird. Die Konfigurationsdatei ist im YAML Format.

Bei dem YAML Format können mehrere Werte "zusammengefasst" werden. Hierbei ist die Anzahl der Leerzeichen oder Tabulatoren für einen Abschnitt immer gleich zu halten.

Kommentare beginnen in YAML mit einer Raute (#).

Beispiel-Konfiguration

```
config:      # Die Basis-Konfiguration ist unter 'config:'
  interval: 5 # Wie lange nach dem Check aller Instanzen gewartet werden
soll, bis
  # sie erneut gecheckt werden. (Sekunden)
  interval_betw_inst: 0 # Wie viel Zeit zwischen den Instanzen vergehen
soll (Sekunden)
  maxretrys: 2 # Maximale Anzahl der erneuten versuche, bis ein 'nagdown'
gesendet werden soll.
  verbose: 0 # Debug Informationen ausgeben. (1)

alarmd:      # Konfiguration für den AlarmClient genaueres siehe
Dokumentation zu AlarmClient.pm.
  host: '127.0.0.1'
  port: 5061
  client: nagios
  pass: test
  maxretrys: 5
  after: 5
  separator: '::'

instances:   # Unter diesem Punkt werden die einzelnen Nagios Instanzen
definiert.
  transnet:   # Name der Instanz
    statuscgi: 'https://nagios.example.com/cgi-bin/nagios3/status.cgi' #
URL der Service Details Seite.
    host: 'nagios.example.com' # Hostname der Instanz (wird für die
Authentifikation gebraucht)
    port: 443 # Port des Webinterfaces (80 = http; 443 = https)
    realm: 'Nagios Access' # Wo sich authentifiziert werden soll.
    user: 'nagioscheck' # Benutzername
    pass: 'Pa$$w0rD.' # und Passwort für die Weboberfläche

nagios2:     # Beispiel für 2. Instanz
  statuscgi: 'http://nagios2.example.net/cgi-bin/nagios3/status.cgi'
```

```
host: 'nagios.example.net'  
port: 80  
realm: 'Nagios Access'  
user: 'checker'  
pass: 'pwTest'
```

GESENDETE ALARME

Jeder Alarm außer 'nagdown' wird erneut übertragen, sobald ein Service mehr den entsprechenden Status hat als davor. Ein Alarm wird erst aufgehoben, wenn auf keinem der Instanzen mehr ein entsprechender Status vorliegt. **Folgende Alarime werden an den alarmd.pl Server übertragen:**

* warning

Falls eine Instanz eine *WARNING* enthält.

* unknown

Falls eine Instanz ein *UNKNOWN* enthält.

* critical

Falls eine Instanz ein *CRITICAL* enthält.

* down

Falls eine Instanz ein *DOWN* enthält.

* nagdown

Falls eine Instanz nach vordefinierter Anzahl erneuter versuche nicht erreichbar ist.

FUNKTIONSWEISE

In dem Script wird eine Endlosschleife gestartet, in welcher für die jeweilige Nagios Instanz die Service Details Seite aufgerufen wird. Diese wird auf das vorkommen von eines erkannten Status überprüft und gegebenenfalls ein Alarm an den **alarmd.pl** Server gesendet. Hierfür wird das Perl-Modul **AlarmClient.pm** benutzt. Zwischen dem Status *DOWN* und *UNREACHABLE* wird nicht unterschieden.

Kann eine Nagios Instanz nicht erreicht werden, so wird deren Retry-Counter (`$retrys`) für eins erhöht. Überschreitet ein Retry-Counter das vordefinierte Limit ('`maxretrys`'), so wird ein 'nagdown' Alarm an den Server übermittelt.

AUTOR

nagios-check.pl - Geschrieben von Florian Schießl (florian@trans.net) im Rahmen des Projektes zur IHK Abschlussprüfung zum Fachinformatiker für Systemintegration Sommer 2013.
Entwicklungsbeginn ist der 11.3.2013.

SIEHE AUCH

AlarmClient.pm - Das verwendete Perl Modul zur Kommunikation mit dem **alarmd.pl** Server.

alarmd.pl - Der Alarm-Server.