Monitoring mit Icinga/Nagios

Sven Übelacker

26.01.2011, Vortrag Computer Stammtisch Harburg

Grundlegendes

Warum Netzwerk- und Dienste-Monitoring?

- Kontrolle über IT-Ressourcen
  ↪ Verfügbarkeit ist Teil der IT-Sicherheit
  ↪ evtl. gesetzlich vorgeschrieben durch Basel II/III
- Entscheidungshilfe in der IT-Planung
  ↪ Engpässe und freie Ressourcen früher erkennen
  ↪ Kennzahlen wichtig für Change-, Capacity-, Availability- und Problemmanagement
- Management braucht Überblick und Entscheidungshilfe (Risikomanagement)
  ↪ Teil vom IT Service Management (ITIL)
  ↪ Service Level Agreements (SLAs) einhalten

Nagios

- Nagios ist ein Quasistandard im Open Source Monitoring
- Nagios Projekt seit 1999 aktiv und von Ethan Galstadt entwickelt
- Icinga ist ein Nagiosfork mit identischer Konfigurationssyntax, entstanden, da Ethan etlichen Benutzerwünschen nicht nachkam
- Icinga kommt mit schickerem Webfrontend daher, erweitertem Funktionsumfang und wartet durch feste Releasezyklen auf (nächster Release in 20 Tagen)
- weitere freie Monitoring Projekte: Zabbix, Cacti, OpenNMS, Munin und Zenoss

Zustandsüberprüfung

- Host alive Check via ICMP Ping
- Service Checks über Nagios Plugins
  ↪ wie `check_http` oder `check_ssh`
  ↪ Nagios Server hat direkten Check-Zugriff auf Ressourcen
- Remote Checks (auch für kaskadierte Checks)
  ↪ SNMP (aktiv) + traps!
  ↪ NRPE = Nagios Remote Plugin Executor (aktiv)
  ↪ `check_by_ssh` (aktiv)
  ↪ NSCA = Nagios Service Check Acceptor (passiv)

Kontaktmöglichkeiten

- E-Mail
- Jabber
- SMS
- Handy Apps
- Firefox Plugin

  ↪ `https://addons.mozilla.org/en-US/firefox/addon/nagios-checker/`
- oder etwas Selbstgebautes

Konfiguration

- Nagios bringt keine Web-basierte Konfiguration mit
- Icinga bietet das Einsehen der Konfigurationdaten
- WebConfig mit
  ↪ NConf (`http://www.nconf.org/`)
  ↪ oder NagioSQL (`http://www.nagiosql.org/`)

Allgemeine Konfiguration

/etc/icinga/icinga.cfg

```
cfg_file=
cfg_dir=
resource_file=
log_file=
nagios_user= (least privileges!)
command_check_interval=
use_syslog=
```

Konfiguration des Webinterfaces

/etc/icinga/cgi.cfg

```
use_authentication=1
use_ssl_authentication=0
authorized_for_system_commands=/C=DE/O=El
Vikingo/OU=IT/CN=Sven
Uebelacker/emailAddress=sven@uebelhacker.de
authorized_for_all_service_commands=
authorized_for_all_host_commands=
authorized_for_read_only=
```

↪ hier am Beispiel für X.509 Authentisierung

Zugriffsschutz via X.509

Statt eines einfachen Benutzerpasswerttupels ändert sich bei X.509 Authentisierung die Syntax wie folgt:

/etc/icinga/htpasswd.users

```
/C=DE/O=El Vikingo/OU=IT/CN=Sven
Uebelacker/emailAddress=svenuebelhacker.de:xxj31ZMTZzkVA
```

Host Checks

```
define host{
        name genericHost
        max_check_attempts 5
        check_period 24x7
        notification_interval 30
        notification_period 24x7
        notification_options d, u, r, f
        notifications_enabled 1
        contact_groups admins
        register 0
}
```

Host Checks

```
define host{
        name pingableHost
        use genericHost
        check_command check-host-alive
        register 0
}
define host{
        use pingableHost
        host_name Mail
        alias mail.uebelhacker.de
        address 86.110.64.51
        parents Switchname
        notes Uebelhacker Mailserver
}
```

Service Checks

```
define service{
        name genericService
        active_checks_enabled 1
        max_check_attempts 5
        normal_check_interval 5
        retry_check_interval 1
        check_period 24x7
        notification_interval 30
        notification_period 24x7
        notification_options w,c,r
        contact_groups admins
        register 0
}
```

Service Checks

```
define service{
        use genericService
        host_name Mail
        service_description ssh
        check_command check_ssh
}
```

Check-Definition

```
define command{
        command_name check_ssh
        command_line $USER1$/check_ssh $ARG1$ $HOSTADDRESS$
}
```

Check Plugins

Viele Nagios/Icinga Plugins werden mitgeliefert, weitere sind hier zu finden:

- `http://exchange.nagios.org/`
- `http://nagiosplugins.org/`
- `http://www.monitoringexchange.org/`
- `http://www.nagios-wiki.de/nagios/plugins/start`
- oder selber schreiben!

Die Pfade zu Plugins können durch Shortcuts (USER*) definiert werden:

/etc/icinga/resource.cfg

```
$USER1$=/usr/lib/nagios/plugins
$USER2$=/usr/local/lib/nagios/plugins
```

Check Plugins selber schreiben

Plugins liefern einen Rückgabewert zwischen 0 und 3 sowie einen Text auf stdout. Die Rückgabewerte bedeuten:

- 0 = OK
- 1 = WARNING
- 2 = CRITICAL
- 3 = UNKNOWN

Text auf stdout wird in Nagios unter „Status Information“ dargestellt und ist Freitext

Ausnahme: Will man zusätzlich Performancedaten zurückgeben, so dass man auch mithilfe von pnp4nagios RRD Grafiken bekommt, müssen nach dem Text gefolgt von einer Pipe die Performancedaten kommen.

Beispiel: Temperatur-Sensorabfrage

```
Temperature OK: 18.21 - between 10 and 25 |
'temperature'=18.21;25;30;0
```

Als erstes gibt man den gemessenen Wert an, dann optional den warning Wert und critical Wert mit Semikolon für horizontale Linien getrennt. Anschließend ist eine Bereichsangabe für die Abbildung möglich, also Minimum und Maximum des gemessenen Wertes.

Visualisierungen werden mit dem RRDTool aus Performancedaten generiert, bekannte Werkzeuge sind:

- PNP4Nagios
- NagVis

Demonstratio

- Nagios Demo
  `http://demo.nagiosadmin.de/nagios`
  User: nagiosadmin, Passwort: nagiosadmin
- Icinga Classic
  `http://classic.demo.icinga.org/icinga/`
- Icinga Web
  `http://web.demo.icinga.org/icinga-web/`
- NagVis
  `http://guest:guest@nagios.demo.netways.de/`
- NagioSQL
  `http://demo.nagiosql.org/nagiosql_en/`
  User: demo, Passwort: demo

Fragen?