

Sunday, April 20. 2008

Nagios - eine Einführung

Ich habe dieses Blog etwas schleifen lassen. Aber das soll sich wieder ändern; ich werde es wieder mit Leben füllen. Anfangen will ich dabei mit einem neuen (und gleichzeitig alten) Projekt - Systemmonitoring.

In der Firma haben wir ein Monitoringsystem, welches inzwischen recht alt ist. Es funktioniert wunderbar, wir haben es auf unsere Bedürfnisse angepasst. Aber in den letzten Monaten haben wir ein paar Probleme festgestellt. Wir könnten natürlich den Hersteller bitten sich die Probleme anzuschauen - aber deren Reaktion wird sein uns zu sagen, wir sollen auf eine aktuelle Version upgraden. Dafür müssten wir auch alle unsere Scripts anpassen. Dann können wir aber auch gleich was neues nehmen...

Da ich im vorigen Job schonmal nagios ausprobiert hatte bin ich der Meinung es hat nach ca. 6 Jahren eine neue Chance verdient. Dann gleich nagios3 und schauen was die Neuigkeiten sind.

Warum Nagios? Nun, es gibt mehrere Gründe für mich dafür:

Ich kenne einige Leute die nagios nutzen, und zwar auch große Installationen. Das heißt es scheint performant genug zu sein

Es ist - wenn man es einmal verstanden hat - recht einfach, auch komplexe Aufgaben damit abzubilden

Es ist Open Source. Das heißt ich kann notfalls jemanden fragen der C/C++ kann ob der Sourcecode das tut was ich will. Alternativ kann ich meine eigenen Plugins schreiben um Checks zu machen

Ich kann es nicht nur unter Unix nutzen sondern auch unter Windows. Zumindest die Checks können dort laufen, was für uns durchaus wichtig ist.

Die Weboberfläche ist - wenn man es richtig macht - übersichtlich und auch Management-geeignet. Gerade in den letzten Tagen habe ich schätzen gelernt mit einem Blick sehen zu können ob und welche Services kaputt sein könnten

Ich kann Abhängigkeiten bauen - wenn der Switch kaputt ist sind logischerweise alle Services dahinter auch nicht erreichbar; also brauche ich dafür keine Meldungen extra; maximal auf der Webseite aber bitte nicht per Mail.

Die Config ist am anfang zwar verwirrend, aber je mehr man mit Templates arbeitet/arbeiten kann umso einfacher werden Spezialanforderungen (bei uns heißt das Gefälligkeit)

Ein paar Sachen fehlen mir selbst oder ich habe sie noch nicht richtig gefunden - Anbindung an Munin in einer Weise wo man nicht von munin die Daten bekommt sondern Nagios sie munin gibt, aber auch sowas werde ich in den nächsten Wochen evaluieren weil man dann wunderbare Grafiken bekommt über Langzeitverhalten. Durchaus ein spannendes Thema.

Bei uns gab/gibt es mit diesem System zwei Ziele:

Ein Monitoringsystem für alle Welten haben: Bisher haben wir das alte Monitoringsystem einmal in der Windows- und einmal in der Unix-Welt. Da sich bei uns die Struktur ändert (ein Servicedesk aka First Line of Defence vor allen anderen) soll alles auf einem System zu sehen sein.

Wir brauchen etwas wo man mit einem Blick erkennt ob es eine Störung gibt und wo sie liegt. Und erst wenn man den "kaputten" Host oder Service anklickt soll man sehen welcher Teil genau kaputt ist.

In den nächsten Tagen oder Stunden werde ich dazu mehr schreiben.

Posted by rince in Tutorials at 15:47

Danke, das kommt genau zur rechten Zeit.

Denn direkt nach Abgabe meiner Diplom-Arbeit darf ich mich mit dem "Redeployment" der Nagios-Installation bei meinem Arbeitgeber befassen, inkl. dem Monitoring von Windows und Unix-Maschinen.

HOWTOs gibt es ja viele, aber ein Beispiel aus der realen Welt fehlte mir bisher.

Also immer fleißig weiter, ich harre der Dinge, die da kommen.

Anonymous on Apr 20 2008, 19:30

Schau mal hier, was die Anbindung von Munin an Nagios angeht:
<http://munin.projects.linpro.no/wiki/HowToContactNagios>

In dem Buch, ist das ziemlich ausführlich beschrieben:
https://www.opensourcepress.de/index.php?26&backPID=178&tt_products=152
Anonymous on Jun 25 2008, 23:28

Hallo Gabriele,

okay, bisher habe ich nur das Buch zu Nagios von Wolfram Barth. Aber in meiner Situation hilft diese Konstellation nicht:

Meine zu überprüfenden Dienste nutzen bereits passive Checks um ihre Stati bekanntzugeben, weil ich nsca in unsere System/Check-Scripte mit einbaue. Aktive Checks sind da eher schwierig weil wir keine Standardsoftware nutzen oder bereits für ein anderes Monitoringsystem Checks haben. Da brauche ich dann nicht noch einen zusätzlichen aktiven Check; da reicht es wenn das normale Checking denselben Status via passive Check an nagios meldet. Vorteil für mich: ich kriege mit ob das andere Monitoringsystem überhaupt checkt; Paranoia (oder Erfahrung) lässt grüßen.

Da hilft es mir aber nicht, wenn munin dazwischen sein soll. Munin selbst verbindet sich ja mit einem Agenten auf der fernen Maschine (was schon in produktiven Umgebungen schwierig werden kann), schreibt dann die Werte in seine RR-Datenbank und generiert die Grafiken. Gut und schön. Die Idee ist gut - solange man nicht passive Checks nutzt. Passive Checks senden selbst ja zu einer von ihnen gewählten Zeit den Status / Wert zu. Das geht aber mit munin nicht, das selbst fragen will.

Daher muss ich mir ein anderes Konzept ausdenken. Ob es möglich ist, nagios (zB über nd2db) regelmäßig um Stati zu fragen wäre eine Möglichkeit. quasi munin einen Agenten für nagios zu geben wäre halt die andere Richtung.
Anonymous on Jun 25 2008, 23:45