Monday, April 21. 2008

Nagios - die Idee und Implementierung (auch KonfigurationsÃ¼bersicht genannt)

Die Konfiguration von nagios ist durchaus komplex, aber wenn man es richtiig anstellt ist es einfach, nach und nach alles einzurichten.

Es gibt verschiedene Sektionen die wichtig sind:

Hosts
Dienste(Services)
Check-Kommandos
Zeitperioden
Kontakte, an den Benachrichtigungen gehen sollen.

Es gibt noch mehr, aber davon erzÃ¤hle ich spÃ¤ter.

Hosts: das sind die GerÃ¤te die Ã¼berwacht werden sollen - das kÃ¶nnen Rechner sein (PCs, Unix, Windows...) aber auch Router oder Switche; oder auch Thermometer, sofern sie via tcp/ip abfragbar sind; eventuell snmp sprechen oder etwas Ã¤hnliches. Bei der Host-Konfiguration muss man mindestens eine IP-Adresse und/oder Hostnamen angeben, damit eindeutig geklÃ¤rt ist welcher Rechner da gemonitored wird.

Dienste: sind die Programme oder Prozesse die von nagios Ã¼berprÃ¼ft werden. Jeder Dienst braucht als Mindestangabe den Host auf dem er laufen soll - sonst hÃ¤tte die Definition auch keinen Sinn. Innerhalb eines Dienstes wird der Name (und Alias) definiert und auch welches Check-Kommando ausgefÃ¼hrt werden soll um den Dienst zu Ã¼berprÃ¼fen.

Check-Kommandos: Das sind die Kommandos die ausgefÃ¼hrt werden um den Dienst zu Ã¼berprÃ¼fen. WÃ¤hrend beim Dienst noch nach Nagiosart Variablen Ã¼bergeben werden (wie der Hostname des zu prÃ¼fenden Rechners) wird hier das Kommando definiert, inklusive dem Aufruf. Quasi eine Ãœbersetzung von Nagios-Konfiguration in ÃœberprÃ¼fungs-Konfiguration.

Zeitperioden: Manche Dienste sollen nur zu einer bestimmten Zeit oder an bestimmten Tagen Ã¼berhaupt geprÃ¼ft werden. Diese kÃ¶nnen definiert werden - zum Beispiel kann gesagt werden dass es unkritisch ist, wenn die Temperatur am Wochenende auf 40Â°C steigt; das kÃ¶nnte ja in Ordnung sein.

Kontakte: Falls es mal Probleme gibt, muss natÃ¼rlich auch jemand benachrichtigt werden. Diese Personen werden in den Kontakten definiert - inklusive der MÃ¶glichkeit, wie die Nachrichten weitergegeben werden.

FÃ¼r Recher, Dienste und Kontakte gibt es auch noch GruppierungsmÃ¶glichkeiten; so dass man zum Beispiel 10 verschiedene Rechner zu einer Gruppe zusammenfassen kann (Kunde A, Kunde B...). Genauso kann man Dienste in einer Gruppe zusammenfassen - Dienste, die nicht gleich sind bzw. nicht dieselben Checks haben, aber dasselbe Ziel. Und Kontaktgruppen ermÃ¶glichen es, Nutzer zusammenzufassen, wenn jemand benachrichtigt werden soll.

FÃ¼r alle diese KonfigurationsmÃ¶glichkeiten gibt es sogenannte Templates - Vorlagen. In den von Nagios mitgelieferten Vorlagen sind alle mÃ¶glichen Parameter bereits gesetzt und meistens auch sinnvoll. Ich habe allerdings festgestellt dass es fÃ¼r mich sinnvoller ist zusÃ¤tzliche Gruppen (zum Beispiel bei den Diensten) zu bauen. Ich habe dann fÃ¼r den Dienst ein Template gebaut und musste dann pro Schnitstelle die ich Ã¼berwachen wollte nur noch sagen wie sie heisst und welcher Port Ã¼berwacht werden soll. Alles andere wurde durch den Gebrauch einer Vorlage dann standardsiert.

Ganz wichtig: in der Konfiguration kommt immer wieder der Parameter "Alias" vor. Das ist nicht nur der Alias fÃ¼r den Dienst/Rechner/wasauchimmer, sondern die KurzerklÃ¤rung des Dienstes wie er auf der Webseite angezeigt wird.

Wie hÃ¤ngt das nun miteinander alles zusammen? Ich fange mal oben an.
Ich habe eine Anzahl von Rechnern, von denen will ich einerseits wissen ob sie leben und ob sie Webserver spielen. Diese ÃœberprÃ¼fung ist fÃ¼r mich aber nur inden Standard-Arbeitsstunden (9-17 Uhr) wichtig. Und auch nur ich soll

dabei erstmal informiert werden.

Die ganzen Konfigurationen habe ich mal ins Wiki gelegt; das ist Ã¼bersichtlicher.

Generell gilt: Wo die einzelnen Definitionen stehen ist fast unabhÃ¤ngig; in der nagios.cfg kann man ganze Verzeichnisse angeben in dem die Konfigurationsdateien zu finden sind. Diese mÃ¼ssen nur die Endung .cfg haben. Ob man jetzt sich das ganze nach Diensten, Rechnern oder anderer Logik folgend strukturiert bleibt jedem Selbst Ã¼berlassen.

Ich habe fÃ¼r einige Applikationen einzelne Config-Dateien weil ich genau weiss dass ich nur dort dann Ã¤ndern muss. Andere dienste (Web) sind bei den Hosts selbst definiert.

Was muss ich also definieren? Wenn ich ganz von vorne anfangen will: Eine Vorlage fÃ¼r die Rechner. in dem schreibe ich (weil ich faul bin nur hinein, wie er heisst (host_name und alias) und die IP-Adresse (address).
Ich kann noch viel mehr definieren, aber anfangs will ich das gar nicht - ich will ja erstmal sehen was daraus entsteht.

Als nÃ¤chstes frage ich mich: Was will ich auf den Rechner eigentlich monitoren? Also, erst einmal will ich generell ein Ping absetzen kÃ¶nnen, dann will ich einmal Web und einmal Mail testen kÃ¶nnen. Also muss ich dafÃ¼r jeweils ein Kommando definieren - pro Rechner kann ich dann einen Dienst generieren der dieses Kommando nutzt. Wenn man sich die commands.cfg im Wiki genauer anschaut, sieht man das es dort Variablen gibt - $HOSTADDRESS$ zum Beispiel. Genau deswegen gibt es die Trennung zwischen Dienste und Check-Kommandos: beim Aufruf des Dienstes wird die Variable $HOSTADDRESS$ mit Inhalt gefÃ¼llt. Damit kann dieselbe Check-Kommandodefinition fÃ¼r verschiedene Dienste genommen werden; bei Pings zum Beispiel kann es interessanter sein; LAN-Strecken anders zu monitoren als WAN-Strecken.

Als nÃ¤chstes definiere ich die Dienste (Services) pro Rechner, die ich abfragen will. Wie man in der Beschreibung sieht, soll serverA nur Web machen, serverB Web und Mail, serverC nur Mail.

Da mehr als ein Rechner die Dienste hat baue ich Service-Gruppen. In diesen Gruppen definiere ich den Namen der Gruppe und die Mitglieder - wenn ich weitere Rechner dazunehme zu dem Dienst, muss ich sie nur der Gruppe hinzufÃ¼gen, mehr nicht.

Dementsprechend mÃ¼ssen auch die Dienste geschrieben werden. Wobei ich hier nur Beispiele mache und deswegen die eigene service.cfg dafÃ¼r nehme.

Jetzt kÃ¶nnte auch klarwerden, wie bei den Checks das $HOSTADDRESS$ gefÃ¼llt wird und wie dann die Checks aufgerufen werden. $USER1$ ist Ã¼brigens ein Makro das vorher definiert wurde - da muss man den langen Pfad bis zum Check-Kommando nicht immer ausschreiben.

Zu guter Letzt habe ich ja gesagt die Dienste sollen nur zu bestimmten Zeiten getestet werden. das mache ich in der timeperiod.cfg

Auch wenn es unÃ¼bersichtlich aussieht anfangs - wenn man sich einmal klargemacht hat wie man monitoren muss und wie, ist diese Konfiguration recht Ã¼bersichtlich.

Posted by rince in Tutorials at 18:50