



Linnéuniversitetet

Institutionen för datavetenskap, fysik och matematik

Fördjupningsuppgift

Nagios Core



Författare Robin Jonsson
Handledare Marcus Wilhelmsson
Termin HT12
Kurskod 1DV427



Innehåll

1. Inledning	3
2. Syfte	3
3. Nagios - Uppbyggnad	3
Main configuration file	3
Resource file(s)	4
Object Definition Files	4
4. Nagios - Övervakning	5
Plugins	5
5. Diskussion	6
6. Referenser	7



1. Inledning

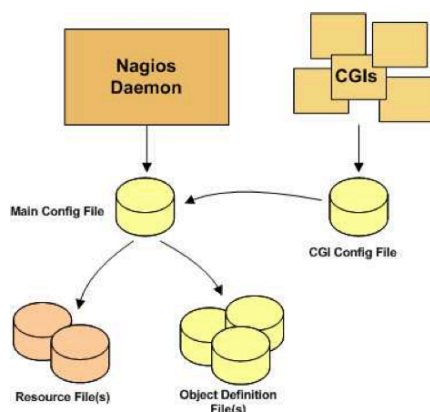
Nagios är ett kraftfullt övervakningssystem som låter organisationer identifiera och lösa problem i IT-infrastrukturen innan de drabbar produktionen. Nagios grundades ursprungligen under namnet NetSaint, men släpptes 2009 under namnet Nagios. Nagios är släppt som Open Source och är publicerat under GNU General Public License version 2 [1]. Med Nagios får användaren möjlighet att övervaka maskiner och tjänster som kan anpassas relativt enkelt med hjälp utav skraddarsydda plugins. Nagios är anpassat för att användas under Linux, men ska även fungera under de flesta UNIX-varianter. Några exempel på tjänster som går att övervaka är SMTP, POP3, HTTP, NNTP, PING och liknande. Det går även att övervaka maskinresurser såsom Processorlast, diskutnyttjande och liknande. Då en tjänst fallerar eller om det sker ett fel med en check så meddelas administratören. Nagios ger även möjlighet att gruppera maskiner och skapa en hierarkisk karta över dem. På så vis kan man skapa beroenden och automatiskt meddela att även underliggande tjänster drabbas om ett ovanliggande beroende fallerar [2].

2. Syfte

Syftet med denna rapport är att ge en grundläggande förståelse för hur Nagios är uppbyggt, hur det fungerar och hur man gör grundläggande konfiguration samt övervakar några vanliga tjänster.

3. Nagios - Uppbyggnad

Nagios är en komplex programvara och administratören är i de allra flesta fall förpassad till kommandoraden för all typ av konfiguration. Är du främmande med denna är antagligen inte Nagios för dig. Har man någon erfarenhet av någon Nagiosbaserad övervakningsprogramvara, såsom exempelvis OP5 känner du antagligen igen dig [2].



Figur 3.1 - Roadmap konfigurationsfiler

Main configuration file

Huvudkonfigurationsfilen innehåller ett antal direktiv som styr hur Nagios-demonen fungerar. Som vi kan se i *Figur 3.1* används huvudkonfigurationsfilen både av CGI:s

och Nagios-demonen. Huvudkonfigurationsfilen innehåller även information om var man hittar ytterligare objektkonfigurationsfiler. I *Figur 3.2* visas exempelvis var kommandon, kontaktuppgifter, mallar och liknande definieras.

```
cat nagios.cfg | sed -e '/^#/d' $1 | sed '/^$/d'
(...)
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
resource_file=/usr/local/nagios/etc/resource.cfg
(...)
```

Figur 3.2 - nagios.cfg

Resource file(s)

Resursfilerna används främst för att lagra användarnamn och lösenord, skyddat från CGIs. Som kan ses i *Figur 3.3* är det även här macros i form exempelvis sökvägar definieras.

```
# Sets $USER1$ to be the path to the plugins
$USER1$=/usr/local/nagios/libexec

# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/local/nagios/libexec/eventhandlers

# Store some usernames and passwords (hidden from the
CGIs)
#$USER3$=someuser
#$USER4$=somepassword
```

Figur 3.3 - resource.cfg

Object Definition Files

Filer som används för att definiera hosts, tjänster, hostgrupper, kontakter, kontaktgrupper, kommandon och liknande kallas för objektdefinierade filer (eng. Object definition files). Dessa filer är i regel placerade under */usr/local/nagios/etc/objects* och här under kan det se ut enligt nedanstående:

```
commands.cfg localhost.cfg printer.cfg templates.cfg windows.cfg
contacts.cfg marla.cfg switch.cfg timeperiods.cfg
```

Figur 3.4 - /usr/local/nagios/etc/objects

Konfigurationsfilerna som finns under ovan angiven objects-katalogen kan se lite olika ut, några är fördefinierade, andra inte. Konfigurationsfilen *marla.cfg* är en av författaren definierade host. Denna fil innehåller information om vilken mall som ska användas, vilka även dessa är definierade under ovan angiven destination under

filen *templates.cfg*. Mallen innehåller bland annat information om hur ofta kontroller ska utföras. Utöver detta så innehåller även *marla.cfg* (se figur 3.5) information om vilka checkar som ska köras på hosten. Checkarna är i sin tur definierade under *commands.cfg* enligt figur 3.6.

```
define host {  
    use                linux-server  
    host_name          marla.somenboss.se  
    alias              marla.somenboss.se  
    address            172.16.255.254  
}  
  
    host_name          marla.somenboss.se  
    service_description Disk_temp  
    check_command      check_nrpe!disk_temp  
}
```

Figur 3.5 - *marla.cfg* (användardefinierat objekt)

```
define command {  
    command_name check_nrpe  
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$  
}
```

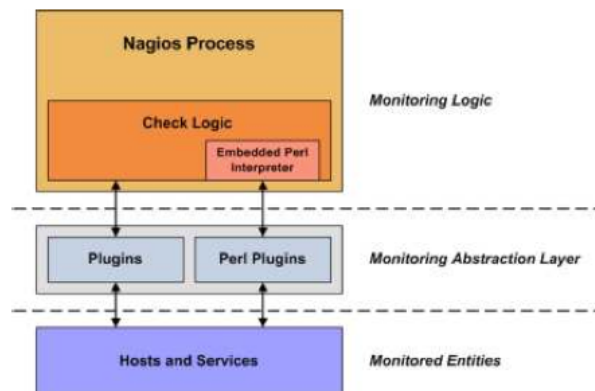
Figur 3.6 - *commands.cfg*

4. Nagios - Övervakning

Plugins

Ett plugin är en kompillerad exekverbarfil eller ett script (Perl-script, shell-script etc.) som kan köras från kommandoraden för att kontrollera statusen på en host eller en tjänst. Nagios använder resultatet från pluginet för att avgöra statusen på hosten eller tjänsten [2].

Pluginet exekveras av Nagios eller av en agent och kontrollerar i sin tur något av scriptet definierat. Nagios hanterar resultatet och kan utföra nödvändiga åtgärder såsom notifieringar[2].



Figur 4.1 - Nagios plugins

I Figur 4.1 påvisar [2] att plugins fungerar som ett abstraktionslager mellan monitorn och de faktiska tjänsterna och hostarna som övervakas. Fördelen med denna metod är det enkelt går att byta ut abstraktionslagret, pluginen, och således kan övervakningen skräddarsys till den specifika IT-miljön.

Genom att använda så kallade Macros ges administratören bland annat möjlighet att skicka med argument till scriptet och kan således på ett dynamiskt och enkelt sätt ställa in exempelvis vilka gränser som ska ses som kritiska. I Figur 4.2 presenteras ett exempel där -H representerar hostnamnet och -c representerar den check som agenten NRPE ska exekvera på hosten [2].

```
/usr/local/nagios/libexec/check_nrpe -H marla.somenboss.se -c disk_temp
```

Figur 4.2 - Macron

5. Diskussion

Syftet med denna fördjupningsuppgift var att skapa en grundläggande förståelse kring hur Nagios är uppbyggt, hur det fungerar samt hur man övervakar några vanliga tjänster. Eftersom tiden är begränsad för denna uppgift var det tyvärr tvunget att begränsa arbetet relativt mycket. Författaren anser sig fått en något mer djupgående förståelse över hur Nagios är uppbyggt. Till skillnad från exempelvis OP5 är Nagios släppt som öppen källkod och ger därför administratören helt andra möjligheter att på egen hand anpassa hela programvaran efter behov. Priset man får betala är användarvänligheten.

6. Referenser

[1] Nagios, "About", *About Nagios*; <http://www.nagios.org/about>

[2] E. G. Nagios Core Development Team, "Nagios Core Version 3.x Documentation", 2010 [Online]. Available:
<http://nagios.sourceforge.net/docs/nagioscore-3-en.pdf>. [Accessed]