

NAGIOS – Aki mindenről elsőként értesül

Ami el tud romlani, az egyszer el is fog. A szolgáltatási szint megállapodások (SLA) világában azonban a problémákról minél hamarabb értesülnünk kell. Erre a feladatra kiváló eszköz a nagios, amely egy rendkívül sokoldalú monitorozó program. Bármit képes nyomon követni, a web kiszolgálónk elérhetőségétől kezdve, a kvóta túllépésen át, egészen egy adott kapcsolat csomagvesztésének arányáig.

Telepítés

A *nagios* telepítéséhez töltsük le a legfrissebb verziót a <http://www.nagios.org/> oldalról, majd csomagoljuk ki, fordítsuk le, és telepítsük az 1. listában látható módon.

A *nagios* valójában semmit nem monitoroz, hanem egy keretrendszert biztosít, amely különféle segédprogramokat (*plugin*) használ a szolgáltatások vizsgálatára. Ezért a *nagios* addig semmi hasznosat nem tehet, amíg nem adunk neki hozzá „szerszámokat”. Töltsük hát le ezeket az eszközöket a honlapról, az „Official Nagios Plugins” címszó alatt találjuk meg, majd telepítsük ezt is a szokásos módon (2. lista).

Számtalan programot gyűjtöttek össze, amelyek többek között lehetőséget adnak Oracle adatbázis monitorozására, SNMP lekérdezések elvégzésére, RADIUS szerver ellenőrzésére, LDAP keresésekre, stb. Érdeemes elolvasni a REQUIREMENTS fájlt, amelyben le van írva, hogy melyik *plugin* telepítéséhez mi szükséges. Ha nem találjuk meg itt a szükséges programot, magunk is elkészíthetjük azt, de erről majd később.

Konfiguráljunk!

A */opt/nagios/etc* könyvtárban találhatóak a *nagios* példa konfigurációs fájlok egy „sample” végződéssel. Nevezzük át mindet úgy, hogy eltávo-

1. Lista A nagios telepítése

```
groupadd ng
useradd -g ng -d /opt/nagios
  -s /bin/sh ng
tar zxvf nagios-2.5.tar.gz
cd nagios-2.5
./configure --prefix=/opt/
  -nagios --with-nagios-user=
  -ng --with-nagios-group=ng
  -disable-statuswrl --with-
  -cgiurl=/cgi-bin/nagios
make
su -c 'make install; make
  install-config'
```

lítjuk ezt, azaz például a *nagios.cfg-sample* állományból *nagios.cfg* legyen. Nézzük meg ezeket a fájlokat! A *nagios.cfg* tartalmazza a *nagios* futásával kapcsolatos paramétereket, például hova naplózzon, milyen felhasználó nevében fusson, különböző naplózási-, riasztási beállításokat, stb. A legtöbb változó használható alapértelmezett értékkel rendelkezik, első körben elég a *cfg_file* paramétert beállítani, amely azt határozza meg, hol vannak az objektum definíciók. Ha csak néhány objektumot akarunk monitorozni, akkor jó választás a *minimal.cfg*, amely minden információt egyetlen helyen definiál. Nagyobb hálózatok esetén azonban lehetőség van az azonos típusú leíró információ-

2. Lista A nagios segédprogramok telepítése

```
tar zxvf nagios-plugins-
  1.4.3.tar.gz
cd nagios-plugins-1.4.3
./configure --prefix=/opt/
  -nagios --with-nagios-user=
  -ng --with-nagios-group=ng
  -with-cgiurl=/cgi-bin/
  -nagios
make
su -c 'make install'
```

ók külön állományokba szervezésére. A *nagios* a riasztásokat a mail parancs segítségével küldi ki, beállításától függően e-mail címre vagy mobil telefonra SMS üzenetben. A konkrét módja és formátuma a *misccommands.cfg* állományban található, az alapértelmezett értékek általában megfelelőek. A *nagios* a használt segédprogramokat, amelyek a tényleges munkát végzik, a *checkcommands.cfg* fájlban definiálja, amelyben a *nagios-plugin* csomag programjai vannak felsorolva. Ha saját programot is akarunk használni, azokat is itt kell megadni. A cikkben egy egyszerű, néhány elemből álló konfigurációt mutatok be. Első lépésként szerkesszük a *minimal.cfg* állományt!

3. Lista Az értesítési adataim

```
define contact{
    contact_name      sj
    alias             SJ
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,r
    service_notification_commands notify-by-email,notify-
        ↪ by-epager
    host_notification_commands host-notify-by-email,
        ↪ host-notify-by-epager
    email            jsuto@freemail.hu
    pager            1234567@sms.pgsm.hu
}
```

4. Lista csoport egyetlen taggal

```
define contactgroup{
    contactgroup_name admins
    alias             Nagios Administrators
    members          sj
}
```

5. Lista Egy gép paraméterei

```
define host{
    use                generic-host
    host_name          sj.acts.hu
    alias              sj.acts.hu
    address            192.168.1.22
    check_command      check-host-alive
    max_check_attempts 10
    check_period       24x7
    notification_interval 120
    notification_period 24x7
    notification_options d,r
    contact_groups     admins
}
```

Távolítsuk el a „COMMANDS” rész define command bejegyzéseit, hiszen ezek a parancsok más állományokban már szerepelnek. Ezután készítsük el a kontakt bejegyzésünket, az enyém a 3. listában látható módon néz ki. Megadtam a nevemet, email címemet, illetve SMS-elérhetőséget, továbbá hogy állandóan (7x24) jöjjenek az értesítések, illetve hogy mind az e-mail

címre, mind a mobilomra küldjön értesítést az eseményekről. Természetesen több kapcsolattartói bejegyzést is készíthetünk.

Az értesítendő személyeket csoportokba lehet összefogni. Ennek az a haszna, hogy ha például 5 személy felügyeli a hálózatot, akkor elég a csoportra hivatkozni. Igény szerint több csoportot is létre lehet hozni, és mind-

egyikben több személy is lehet, ez azonban egy kis hálózat egyetlen emberrel, amint az a 4. listából is látszik. Több tag esetén vesszővel elválasztva lehet őket felsorolni.

A HOSTS szekcióban kell felsorolni az egyes objektumokat (azaz jellemzően számítógépeket, útválasztókat, nyomtatókat és más hálózati eszközöket), amelyeket monitorozni akarunk. Az én egyik gépem az 5. lista szerint néz ki.

Megadtam az objektum nevét, IP-címét, a 2 értesítés közötti időtartamot (2 óra), hogy a riasztásokat az admin csoport kapja meg, ill. az objektum állapotát ellenőrző program nevét (*check-host-alive*), amely nem tesz mást, mint a ping programmal ellenőrzi a gép elérhetőségét.

Végül az egyes gépekhez szolgáltatásokat kell rendelnünk, amelyeket a *nagios* ellenőrizni fog. Az előbbi gépen futó web kiszolgáló elérhetőségét a 6. listában látható bejegyzés hatására fogja megvizsgálni a *nagios* (6. Lista). Megadtam a gép nevét, a szolgáltatás megnevezését (*HTTP*), állandóan (24x7) figyeljük, 5 percnként fut le az állapotot figyelő program, 4 óránként (240 perc) küldünk újabb riasztást, ha addig a hiba nem hárul el, itt is az admin csoport kapja meg az értesítéseket, továbbá a *check_http* néven szereplő program végzi az ellenőrzést. Egy géphez természetesen több szolgáltatás is rendelhető, ebben az esetben minden szolgáltatáshoz egy *'define service'* blokkot kell készíteni.

Már csak egy lépés van hátra: a konfiguráció ellenőrzése, amelyet a

```
/opt/nagios/bin/nagios -v
↪ /opt/nagios/etc/nagios.cfg
```

parancs segítségével tehetünk meg. Ha minden rendben van, akkor a 7. listához hasonló üzenetet kapunk. Ezek után indítsuk el a

```
/opt/nagios/bin/nagios
↪ /opt/nagios/etc/nagios.cfg &
```

paranccsal a háttérben.

Levelezzünk!

A *nagios* az eredményeket egy naplóban vezeti, amely a */opt/nagios/var* könyvtárban található. A minden

6. Lista Egy web kiszolgáló definiálása

```
define service{
    use                generic-service
    host_name          sj.acts.hu
    service_description HTTP
    is_volatile        0
    check_period       24x7
    max_check_attempts 4
    normal_check_interval 5
    retry_check_interval 1
    contact_groups     admins
    notification_options w,u,c,r
    notification_interval 240
    notification_period 24x7
    check_command      check_http
}
```

9. Lista Riasztás e-mailben

```
Date: Tue, 24 Oct 2006
↳ 11:08:30 +0200
From: ng@acts.hu
To: jsuto@freemail.hu
Subject: Host DOWN alert for
↳ localhost!

***** Nagios *****

Notification Type: PROBLEM
Host: localhost
State: DOWN
Address: 127.0.0.1
Info: /bin/ping -n -U -w 10
↳ -c 1 127.0.0.1

Date/Time: Tue Oct 24
↳ 11:08:30 CEST 2006
```

7. Lista A nagios konfigurációs állományai rendben vannak

```
Nagios 2.5
Copyright (c) 1999-2006 Ethan
↳ Galstad
↳ (http://www.nagios.org)
Last Modified: 07-13-2006
License: GPL

Reading configuration data...

Running pre-flight check on
↳ configuration data...

Checking services...
    Checked 3 services.
Checking hosts...
    Checked 2 hosts.

.... sok szöveget kiír még

Total warnings: 0
Total Errors: 0

Things look okay - No serious
↳ problems were detected
↳ during the pre-flight check
```

8. Lista A status.dat fájl részlete

```
service {
    host_name=sj.acts.hu

    service_description=HTTP
    ....
    plugin_output=HTTP OK
↳ HTTP/1.1 200 OK - 7342
↳ bytes in 0.002 seconds
    performance_data=
↳ time=0.002078s;;;0.000000
↳ size=7342B;;;0
    ....
    last_check=1161681000
    next_check=1161681300
}
```

A gépemen a *ping* programról levettem korábban a *setuid* bitet, ezért a *nagios* hibát jelzett, amit – lásd 9. lista – elküldött e-mailben. A

```
setfacl -m u:ng:x /bin/ping
```

paranccsal korrigáltam a hibát, aminek hatására pár percen belül meg is jött a 10. listában látható nyugta, hogy minden rendben. Mindkét eseményről értesítést kaptam a mobilomra is. Előfordulhat, hogy egy SMS túl hosszú, hogy 1 üzenetbe beleférjen, ebben az esetben a mobil operátor feldarabolja. Ebben az esetben érdemes szerkeszteni

10. Lista Levél a szolgáltatás helyreállításáról

```
Date: Tue, 24 Oct 2006
↳ 11:13:30 +0200
From: ng@acts.hu
To: jsuto@freemail.hu
Subject: Host UP alert for
↳ localhost!

***** Nagios *****

Notification Type: RECOVERY
Host: localhost
State: UP
Address: 127.0.0.1
Info: PING OK - Packet loss
↳ = 0%, RTA = 0.05 ms

Date/Time: Tue Oct 24
↳ 11:13:30 CEST 2006
```

a *misccommands.cfg* fájlt, és fazonra igazítani, hogy rövidebb legyen, néhány elem elhagyható belőle. A *nagios* azonban többet is tud ennél, egy web oldalon megnézhetjük az egyes gépek ill. szolgáltatások állapotát, a múltbeli eseményeket, grafikonokat, trendeket, stb. Ehhez telepítsünk egy web kiszolgálót (ha még nincs). A példában egy dedikált *Apache virtualhost*-ot fogunk a *nagios*

ellenőrzéskor frissülő *status.dat* állományban található az összes gép szolgáltatása, ahol a beállítások mellett látható a használt *plugin* tesztjének eredménye is, amely például a 8. listán látható módon nézhet ki, ha minden rendben van.

11. Lista Az Apache kiszolgáló beállítása

```
<VirtualHost 192.168.1.22:80>
  ServerName nagios.ceg.hu
  DocumentRoot
    ↪ /opt/nagios/share

  <Directory
    ↪ /opt/nagios/share>
    Options Indexes
    AllowOverride Limit
    ↪ AuthConfig
  </Directory>

  <Directory
    ↪ /opt/nagios/sbin>
    AuthUserFile
    ↪ /opt/nagios/sbin/
    ↪ .htpasswd
    AuthName "nagios"
    AuthType basic

    require valid-user

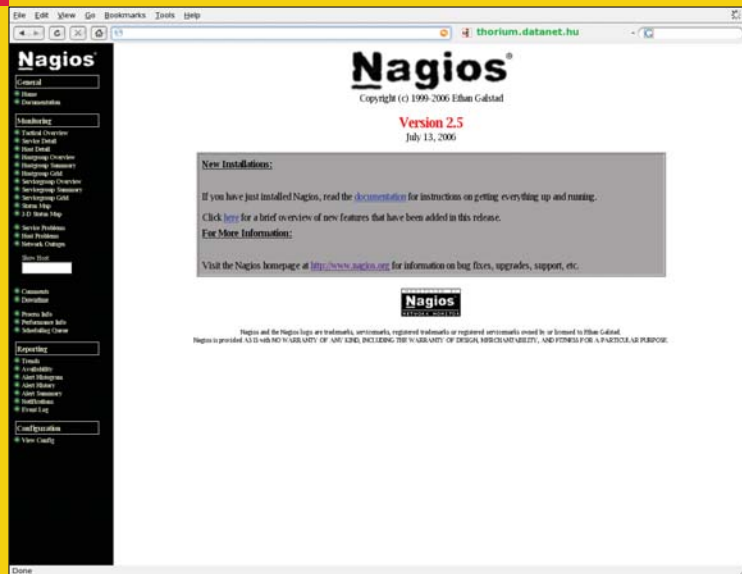
    order deny,allow
    deny from all
    allow from 192.168.1.22

    satisfy all
  </Directory>

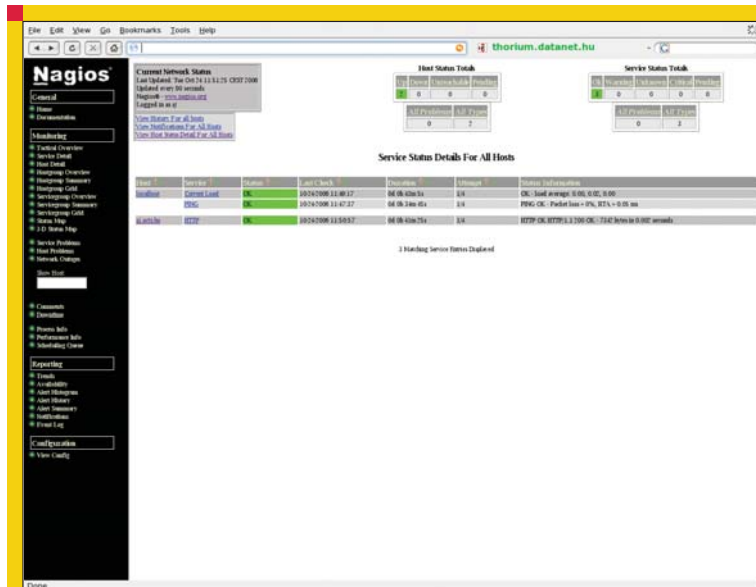
  ScriptAlias /cgi-bin/
    ↪ nagios/ /opt/nagios/sbin/
</VirtualHost>
```

rendelkezésére bocsátani.

A 11. listában látható konfigurációs fájl részlet biztosítja, hogy a <http://nagios.ceg.hu/> címen meg tudjuk nézni a szolgáltatások állapotát. A `/opt/nagios/sbin` könyvtárba pedig tegyünk egy `.htpasswd` fájlt is tenni a megfelelő hozzáférés szabályozás érdekében. Ahhoz, hogy egy böngészővel elérhessük a `nagios` eredményeit, szerkesszük a `cgi.cfg` fájlt, és az `authorized_for_*` paraméterekhez adjuk hozzá saját azonosító nevünket, amely a `.htpasswd` állományban is szerepel, különben hibát fogunk kapni a web felületen. Nyissuk meg böngészővel a <http://nagios.ceg.hu/> címet. Ekkor az 1. ábrához hasonló képet fogunk kapni.



1. ábra A nagios bejelentkező felülete



2. ábra Az egyes gépek összes szolgáltatásának állapota

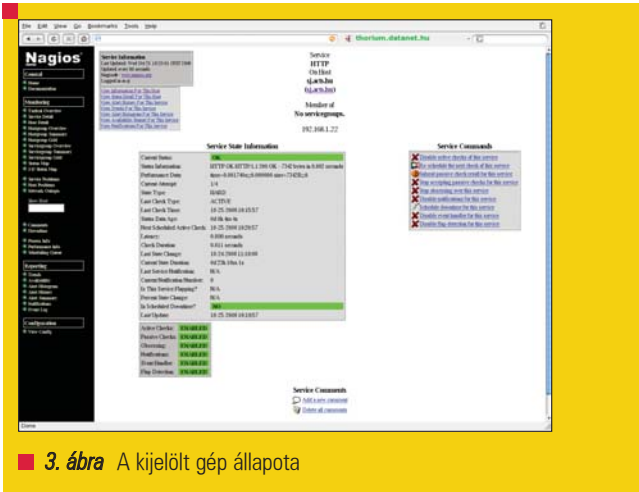
Leggyakrabban a „Service Detail” menüt szoktam használni, ahol – mint az a 2. ábrán látszik – könnyen át lehet tekinteni az egyes gépek szolgáltatásainak állapotát.

Az adott gép nevére vagy valamelyik szolgáltatására kattintva annak státusza a 3. ábrán látható módon jelenik meg, ahol több statisztikai adat is szerepel, például hányadik próbálkozás eredményét látjuk, mikor futott le az ellenőrzés, mi az eredménye, mikor lesz a következő teszt, stb.

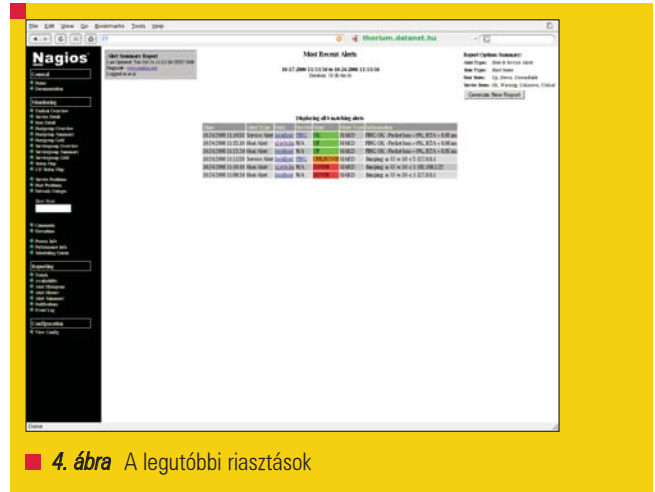
Az „Alert Summary” menüre kattintva egy űrlap segítségével be tudjuk állítani (például gép, szolgáltatás, dátum),

hogy a riportban pontosan mi szerepeljen. Egy lehetséges kimenet a 4. ábrán látható.

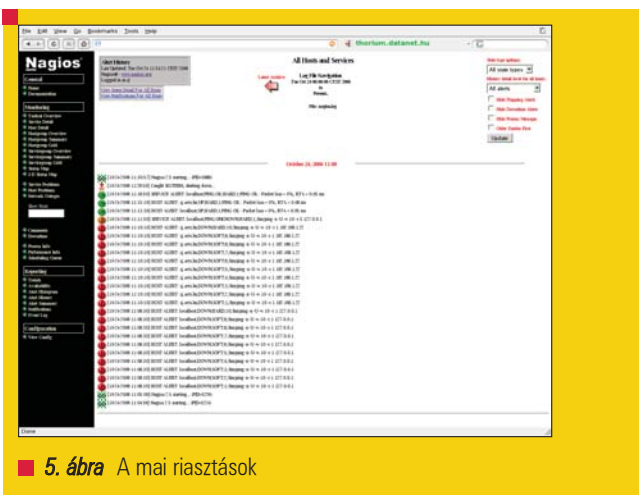
Az „Alert History” menü az adott napi riasztásokat mutatja meg. Az 5. ábrán látható az összes riasztás részletei, például dátum, gép neve, állapot, típus és a használt segédprogram a paramétereivel együtt. Az állapot lehet úgynevezett **HARD** vagy **SOFT**. Ha egy ellenőrzés nem sikerül, akkor az adott szolgáltatás **SOFT** állapotba kerül. Ha a következő alkalommal rendben válaszol, akkor nincs riasztás. Ha 4 egymást követő alkalommal (max_check_attempts változóval



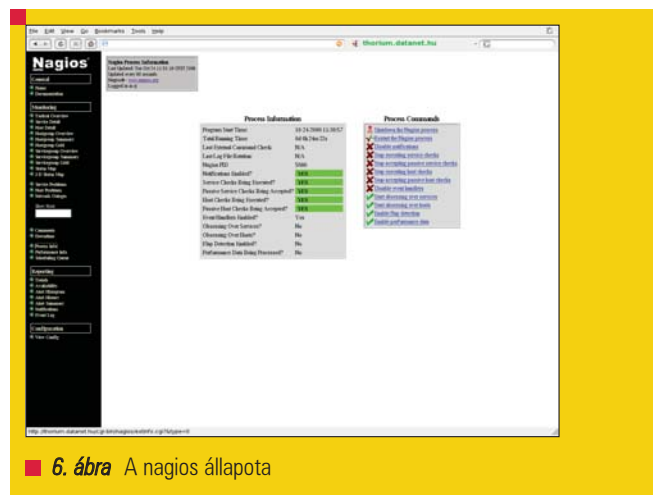
3. ábra A kijelölt gép állapota



4. ábra A legutóbbi riasztások



5. ábra A mai riasztások



6. ábra A nagios állapota

```

12. Lista Az első nagios segéd-
programom

#!/bin/sh

LIMIT=100
START=`date +%s`
HOST=localhost

N=`mysqladmin -defaults
  -file=~/.my.cnf processlist
  | wc -l`

STOP=`date +%s`

RUNTIME=`expr $STOP - $START`

echo "$START;$HOST;$RUNTIME;
  `kapcsolatok szama: $N`

if [ $N -gt $LIMIT ]; then
  exit 2; fi
exit 0
    
```

konfigurálható érték a *nagios.cfg* fájlban) is elérhetetlen vagy nem megfelelő a szolgáltatás, akkor az állapot **HARD** lesz, és riasztást küld a *nagios*. Ezzel a mechanizmussal elkerülhető, hogy egy rövid átmeneti zavar miatti SMS felébressze az üzemeltetőket éjjel. A *nagios* folyamat állapotáról a 6. ábrán látható módon is kaphatunk néhány információt, például *PID*, indítás ideje és néhány konfigurációs érték.

Szadjuk testre!

A *nagios* lehetőséget ad arra, hogy saját segédprogramot is használjunk. Ehhez írunk kell egy tetszőleges nyelvű (például Perl, C, TCL, bash, stb.) programot, ami elvégzi az adott szolgáltatás ellenőrzését, kiír néhány adatot, végül beállítja a visszatérési értéket. Nézzünk meg a 12. listában szereplő példát, amely leellenőrzi a *MySQL* adatbázist, és riaszt, ha túl sok nyitott kapcsolat van. Látható, hogy a program visszatérési értéke 0, ha minden rendben van, illet-

ve hiba esetén 2. Az összes lehetséges visszatérési érték a */opt/nagios/libexec/utlils.sh* fájlban található. A *nagios* nem csak rendszeres lekérdezések elvégzésére használható, de megfelelő program segítségével akár reagálni is lehet az esetleges hibákra. Az előbbi példában amellet, hogy riasztást küld a túl sok szimultán kapcsolatról, akár újra is indíthatja (megfelelő jogosultság birtokában) az adatbázis kiszolgálót. A *nagios* legfőbb erénye azonban az, hogy a lehetőségeknek csak az Olvasó képzelete szab határt. Ha egy igazán rugalmas és megbízható monitorozó alkalmazást keres, a *nagios* kitűnő választás.



Sütő János
(jsuto@freemail.hu)
1997 óta használ Slackware Linux-ot. Szabadidejében a postfix clapf nevű vírus- és spam-szűrőjét polírozza.