Alexander van Deursen, Fotolia

Tuning in with Hobbit, Nagios, and monit

FREE MONITOR

Open source tools such as Hobbit, Nagios, and monit provide system monitoring for large or small networks. **BY JULIET KEMP**

If you look after more than one computer, eventually you'll realize you could benefit from some form of automatic monitoring – certainly after someone surprises you with the announcement that a system you are responsible for just crashed. In addition to reducing the response time for a downed system, system monitoring can also help you identify problems in advance – before the situation becomes an emergency. Even if you only have one computer, advanced notification that your disks are getting full or that *sshd* is down can save considerable time and stress.

If you have a specific service or situation you want to monitor, you could, of course, brew your own custom monitoring script and trigger it with cron. However, you really do not need to go reinventing wheels when several open source applications will handle the job for you. In this article, I look at three of the main contenders – Hobbit, Nagios, and monit. All are open source and freely available. All have good points and limitations. The ideal solution depends on your network, your experience, and your needs.

Hobbit Monitor

Hobbit [1] is an open source monitoring system inspired by Big Brother [2]. With Hobbit, you can monitor anything from tiny to enormous networks. It is available in package form for Ubuntu, Fedora, and several Linux distros. Debian users can use Hobbit packages for Lenny/Sid, as well as an Etch backport. The most recent release was three years ago, so it's not clear whether the project is still under active development; however, I have been using Hobbit at work for some months, and it performs well.

The Hobbit monitoring system is centralized, so you'll need a central Hobbit server, plus client software on each machine you want to monitor. The information is served up through a web interface on the central server, so you also need Apache2. The installation is straightforward; you should have a basic system running pretty quickly.

By editing just a couple of well-documented text files, you can manage your Hobbit configuration. The hosts you intend to monitor are all specified through a single file (one line per host, with service information on the same line as the host name and address), and service checks are already defined for you. The warn/alarm settings for various services and situations are defined through another file.

The typical alerts are possible – email is the most obvious option – but you also can plug in any script you want and configure more exotic responses. The online documentation describes a technique for forwarding alerts to a mobile phone, for example. Because you can attach more than one alert to a particular condition, you could have one email message sent immediately, then another to an escalation address after an hour. Also, you can configure an alert to autorepeat and to acknowledge a fix. The documentation online is a little sparse, but the Tips/Tricks page is useful.

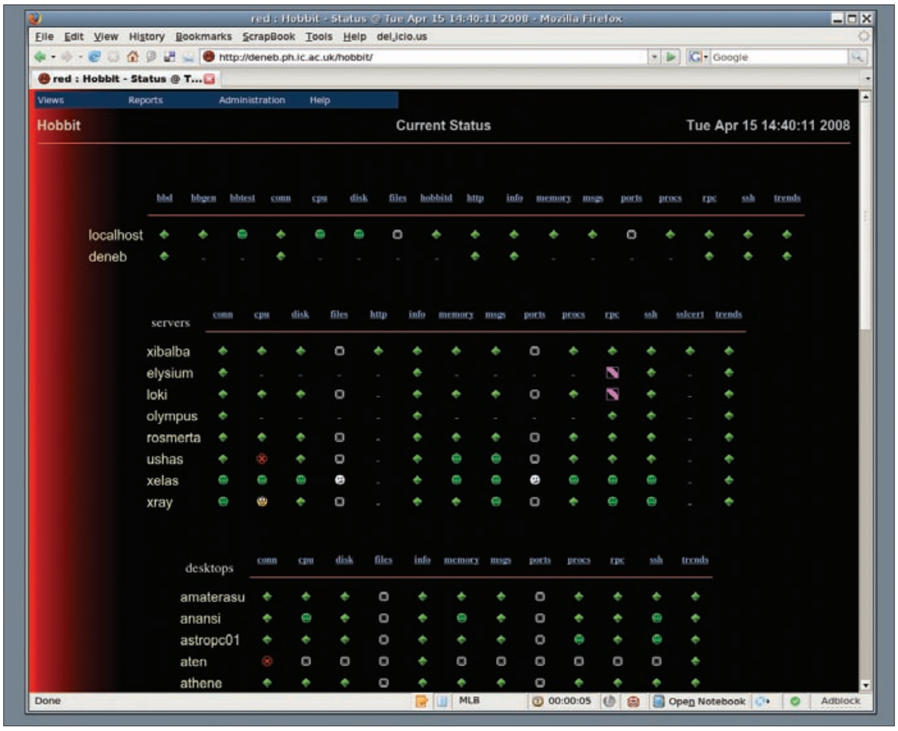

red : Hobbit - Status @ Tue Apr 15 14:40:11 2008 - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help del.icio.us
http://deneb.ph.ic.ac.uk/hobbit/
red : Hobbit - Status @ T...
Views Reports Administration Help
Hobbit
Current Status
Tue Apr 15 14:40:11 2008
bbd bbgen bbtest conn cpu disk files hobbitd http info memory msgs ports procs rpc ssh trends
localhost
deneb
servers
conn cpu disk files http info memory msgs ports procs rpc ssh sslcert trends
xibalba
elysium
loki
olympus
rosmerta
ushas
xelas
xray
desktops
conn cpu disk files info memory msgs ports procs rpc ssh trends
amaterasu
anansi
astropc01
aten
athene
Done

Figure 1: The Hobbit web interface, showing groups of machines.

The Hobbit web interface (Figure 1) is colorful and easy to read. Clicking through on any machine supplies more details, and you can divide machines into groups to make it easier to navigate.

Once Hobbit is running, it needs very little intervention, although you might want to make some tweaks initially as you work out what tends to crop up in your system. After that, however, you can leave it to its own devices and it will just keep going. Hobbit is straightforward and fairly basic in its setup, and it does most of the things you might want. The web-based display is clear and easy to understand at a glance.

Nagios

Nagios [3] is a bit more difficult and time consuming than Hobbit is to get set up and configured correctly. The flip side is that Nagios is powerful. Nagios installs from a tarball or in package form from your distro package manager. Debian, Ubuntu, and Fedora/CentOS all have packages available – make sure you're getting at least Nagios 2. If you want the most up-to-date version (3.0.1 at the time of writing), you'll probably need to download the tarball.

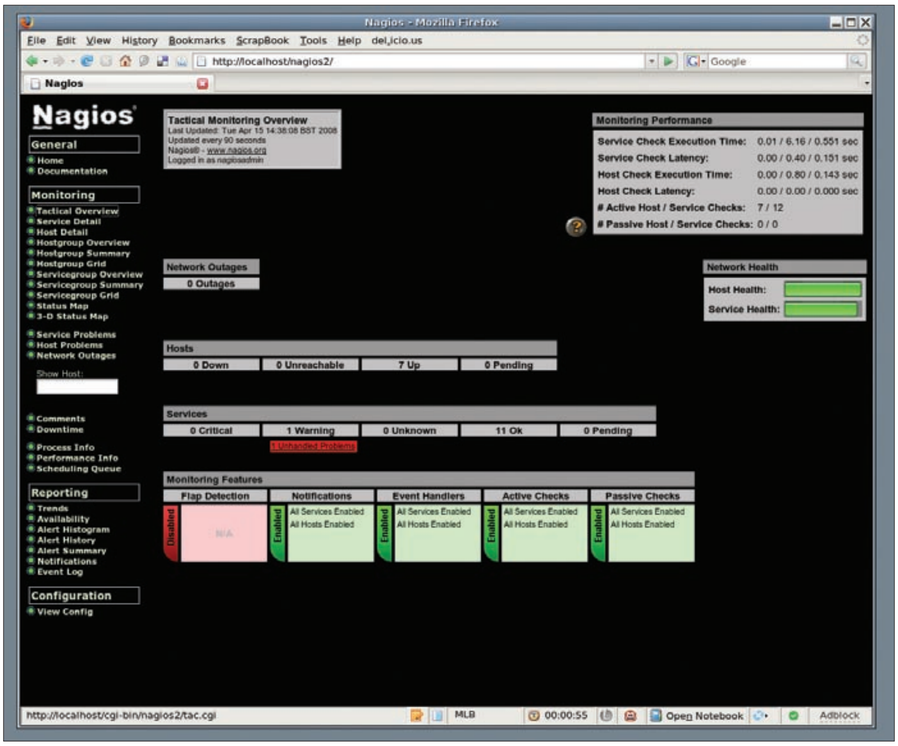
The Nagios system relies on plugins, which are basic Unix commands that return an exit code and a message to Nagios, providing information on the state of the service you are monitoring. A huge number of plugins are available. If you install from your distro package manager, you should get a handful of the most useful plugins as an automatic dependency, or you can download a plugin tarball from the Nagios website.

Despite the complexity, it's possible to get a very basic Nagios system up and running pretty fast. Although Nagios reports via a web interface, the configuration is all done in text files. The configuration files are a bit confusing initially – Debian separates things out into separate files by default, and I'd recommend this as a helpful practice. The documentation is clear and very comprehensive.

As with Hobbit, you have to specify each host you want to monitor. Nagios is slightly more difficult to configure than Hobbit, in that multiple options need to be set (as opposed to the Hobbit system of having a single line per host in a single file). However, you can set up a default template that will significantly reduce the amount of typing.

Nagios can monitor services on either a per-host basis or through a hostgroup. For example, you could have a hostgroup of all web servers, all SSH servers, and so on. Nagios allows wildcards, so it's easy to define an "all hosts" group or set up services that cover all hosts. The commands used to check services are defined within the plugin packages.

The web report interface (Figure 2) again requires a basic Apache2 install. Unlike Hobbit, Nagios provides extensive authentication options for the web interface. In the configuration, you can control which users can see which information on which services, and you can also specify which users can issue com-



Tactical Monitoring Overview
Last Updated: Tue Apr 15 14:38:08 BST 2008
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as nagiosadmin

Monitoring Performance
Service Check Execution Time: 0.01 / 6.16 / 0.551 sec
Service Check Latency: 0.00 / 0.40 / 0.151 sec
Host Check Execution Time: 0.00 / 0.80 / 0.143 sec
Host Check Latency: 0.00 / 0.00 / 0.000 sec
Active Host / Service Checks: 7 / 12
Passive Host / Service Checks: 0 / 0

Network Outages
0 Outages

Network Health
Host Health:
Service Health:

Hosts
0 Down | 0 Unreachable | 7 Up | 0 Pending

Services
0 Critical | 1 Warning | 0 Unknown | 11 Ok | 0 Pending
1 Unhandled Problems

Monitoring Features

Flap Detection	Notifications	Event Handlers	Active Checks	Passive Checks
N/A	All Services Enabled All Hosts Enabled	All Services Enabled All Hosts Enabled	All Services Enabled All Hosts Enabled	All Services Enabled All Hosts Enabled

Figure 2: The Nagios web interface Tactical Monitoring Overview page.

mands for specific hosts. The display is very readable, with a variety of ways to look at the data.

Nagios will issue configurable alerts in particular situations. The system includes an escalation feature that triggers a further alert action after a specified length of time.

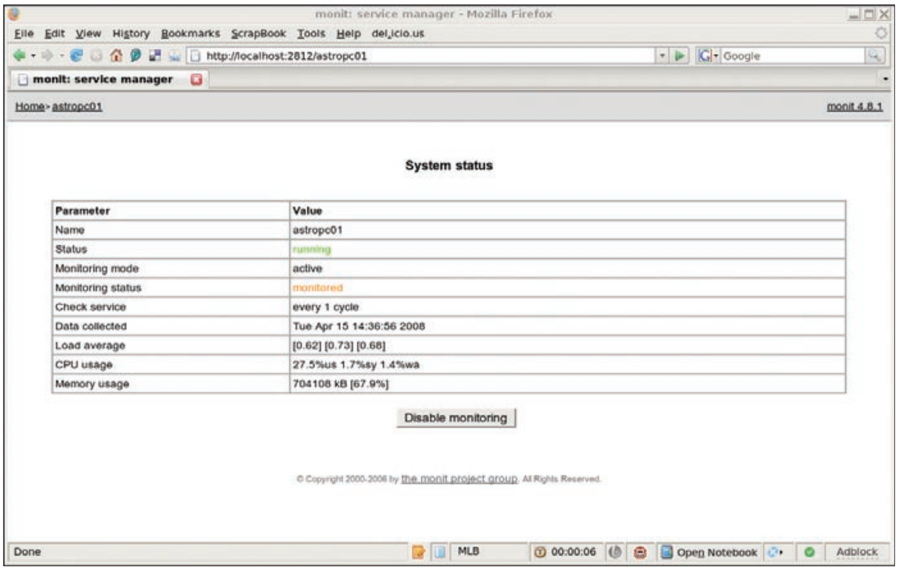
Via the plugin interface, you can define almost anything you want. For example, you can define your own commands to use to check particular services and use macros to make these more extensible and easily readable. However, the enormous number of plugins already available means there's a fair chance that you won't need to do any such thing.

Nagios also can restart services if they fail with the use of a script that invokes Cfengine (or another similar system).

Although Nagios is more difficult to configure than Hobbit, it is much more powerful and configurable once the system is working. The Nagios monitoring system is nice for large, professional networks, but it might be overkill for a small network.

monit

Monit [4] manages and monitors processes, services, files, directories, and other system variables – either locally or remotely. Either install from source code or find a package for your favorite distro. Like the other tools described in this article, monit will send alert email messages, and it provides a web interface (Figure 3). The web interface is, unsurprisingly, a bit more basic than those provided by Hobbit or Nagios. One significant advantage of monit is that you easily can set it up to restart services automatically if they fail. Monit is a standalone system that doesn't rely on plugins, although it does happily integrate with *init* and *rc-* scripts, which is how services are restarted.


monit: service manager - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help del.icio.us
http://localhost:2812/astropc01
Google
monit: service manager
Home› astropc01
monit 4.8.1
Disable monitoring
© Copyright 2000-2006 by the monit project group. All Rights Reserved.
Done
MLB
00:00:06
Open Notebook
Adblock

System status

Parameter	Value
Name	astropc01
Status	running
Monitoring mode	active
Monitoring status	monitored
Check service	every 1 cycle
Data collected	Tue Apr 15 14:36:56 2008
Load average	[0.62] [0.73] [0.68]
CPU usage	27.5%us 1.7%sy 1.4%wa
Memory usage	704108 kB [67.9%]

Figure 3: The rather more basic monit web interface.

Other Options

Other open source monitoring tools such as Munin [5] and mon [6] do the job reasonably well but lack the power of Nagios. If you really want to experiment, you can eschew all this software and construct your own monitoring software. Ping, cron-apt (for Debian), logwatch, and similar utilities are all useful for this. However, for the vast majority of users, this is a serious waste of time. Your life will be much easier if you let someone else do the programming. All of the solutions discussed in this article are sufficiently configurable and tweakable that even the most controlling admins should be able to get what they want.

The default configuration file has everything commented out, so you need to go through it, uncommenting and editing what you need. Unlike some other tools, monit does not automatically use default values. If a statement is commented out in the *monitrc* file, then it is out of use. It is quick to set up – primarily because it only really monitors the local host, so the long lists of other hosts required for Hobbit and Nagios aren't needed. Monit allows basic (e.g., ping) checks to other hosts, so if you have host dependencies (e.g., a machine that needs to access another machine to get at a MySQL database), monit could warn you that the MySQL machine was down.

This does mean that it doesn't provide the sort of services that Nagios and Hobbit do, in which you can have a central server monitor all your hosts. What it does, though, it does well, and it does have the major advantage that it can restart services when they fail. The online manual is helpful and fairly comprehensive, and mailing lists and other support are available.

Monit is great for monitoring a single host, especially because it will restart systems. The monit system is not as effective for a larger network, although it might work quite well alongside Hobbit or Nagios, enabling centralized monitoring, as well as restarting local services.

Conclusion

Hobbit, Nagios, and monit all do a decent job. Nagios is the most powerful, but setting it up to get the full benefit of that power can be difficult. A basic system is reasonably quick to set up once you understand how the files work, and the power will be there so you can extend your monitoring effort in the future.

If you only have one or two systems, monit is probably a better bet than setting up Nagios. Monit also really excels at system recovery, which Hobbit and Nagios don't handle natively (although Nagios can be set up for system recovery with other software).

Hobbit is a reasonable balance between the two, especially if your needs are not too complex, but it's not as extensible or configurable as Nagios, and it doesn't heal services as monit does.

For myself, I'm sufficiently impressed with Nagios that the next project on my todo list is switching to it! ■

INFO

[1] Hobbit: *http://hobbitmon.sourceforge.net/*

[2] Big Brother: *http://www.bb4.org/*

[3] Nagios: *http://www.nagios.org/*

[4] monit: *http://www.tildeslash.com/monit/*

[5] Munin: *http://www.linpro.no/projects/munin/*

[6] mon: *http://mon.wiki.kernel.org/index.php/Main_Page*