Introduction to system monitoring with Nagios, Check_MK and Open Monitoring Distribution (OMD)

Iñigo Aldazabal Mensa – Centro de Física de Materiales (CSIC-UPV/EHU)

HPCK’14 Barcelona, 13-14th January 2014
Intro
- Why monitoring?
- What to monitor?
- How to monitor?

Nagios
- Introduction
- Active and passive monitoring checks, plugins and extensions
- Considerations

Check_MK
- Introduction
- Check_MK Agent
- Architecture
- Multisite front-end

OMD
- Introduction
- Included Software/packages
- Installation
- OMD General Overview - Components
Iñigo Aldazabal

Introduction to system monitoring with Nagios, Check_MK and OMD
Why monitoring?

- hardware fails
- software fails
- disks get full
- backups not working
- water flows into Data Centers
- ...

We all do some kind of monitoring, but monitoring systems do not get bored, and do it 24x7.

Is Good to now about these things as they happen, even better beforehand, in order to take correcting actions.

The bad part: correcting the problems just as (or before) they happen may give the false illusion that no job is being done on our part making your hard labour being underestimated.
What to monitor?

In general:
- computers
- printers
- network equipment
- servers (both physical and virtual) / appliances
- ...

In our more specific case (HPC systems):
- cluster head nodes
- compute nodes (disk space, NFS mounts, SMART status, ...)
- backups
- storage systems
- Data Center environment (temperature, water, ...)
- ...

Iñigo Aldazabal
Introduction to system monitoring with Nagios, Check_MK and OMD
We want a network monitoring solution providing monitoring + alerting + historical data for analysis.

Lots of options, both free and proprietary software: Nagios, Zabbix, Groundwork, Cacti, Munin, ...

Extensibility is a must, as we are dealing with very specific (HPC) systems, and we do script things!

We did choose Nagios (OMD/Check_MK came later):

- Well established, de facto industry standard.
- Long trajectory and big user base (i.e. support, tutorials, etc.).
- Very flexible notification system.
- Extensive set of plugins.
- Open Source
How to monitor?

- We want a network monitoring solution providing monitoring + alerting + historical data for analysis.
- Lots of options, both free and proprietary software: Nagios, Zabbix, Groundwork, Cacti, Munin, ...
- Extensibility is a must, as we are dealing with very specific (HPC) systems, and we do script things!
- We did choose Nagios (OMD/Check_MK came later):
  - Well established, de facto industry standard.
  - Long trajectory and big user base (i.e. support, tutorials, etc.).
  - Very flexible notification system.
  - Extensive set of plugins.
  - Open Source

Your mileage may vary. Any solution is better than no solution!
Introduction to system monitoring with Nagios, Check_MK and OMD

- **Intro**
  - Why monitoring?
  - What to monitor?
  - How to monitor?

- **Nagios**
  - Introduction
  - Active and passive monitoring checks, plugins and extensions
  - Considerations

- **Check_MK**
  - Introduction
  - Check_MK Agent
  - Architecture
  - Multisite front-end

- **OMD**
  - Introduction
  - Included Software/packages
  - Installation
  - OMD General Overview - Components
What is Nagios?

Open Source computer and network monitoring system which monitors hosts and services, and alerts us when they go under undesired behaviour.

What is monitored:

- Network services (SMTP, POP3, HTTP...)
- Network connected equipment (ping, SNMP...)
- Systems (CPU load, free disk space, hard disk health, backup status, ...)

How does it alerts:

- email
- SMS
- ...
Basic Concepts

- **Host**: the physical equipments (*ping*).
- **Services**: Resources to be monitored within an specific hosts (http response, printer toner levels, hard disk SMART status, backups status, ...).
- **Plugins**: Programs (scripts or executable code) which can be run from the command line in order to verify the state of a host or service, typically named as `check_xxx` (check_http, check_printer, check_smart, check_backup ...).
- **Contacts and Contact Groups**: People to be notified and how they are notified.
- **Time Periods**: Week days and time intervals in which a host/service has to be monitored.
Active monitoring: ping, check_http...

Passive monitoring, asynchronous by nature: SNMP traps, security alerts, ...

Nagios Server

Passive Check Initiated by Client

Nagios Clients Actively Checked Using NRPE or Active Service Checks
Active Monitoring

Run on the Nagios server itself ("remote" checks): SNMP, ping, check_http, check_printer...

Run on the systems being monitored ("local" checks): Nagios Remote Plugin Executor (NRPE)
Passive Monitoring

Run on the remote hosts: Nagios Service Check Acceptor (NSCA)

SNMP Traps: Net-SNMP + SNMP Trap Translator (SNMPTT)
### Host checks

- Hosts are checked by the Nagios daemon at defined regular intervals (1 min. in OMD).
- Hosts that are checked can be in one of three different states:
  - **UP**
  - **UNREACHABLE**
  - **DOWN**

<table>
<thead>
<tr>
<th>state</th>
<th>Host</th>
<th>Icons</th>
<th>Alias</th>
<th>OK</th>
<th>Wa</th>
<th>Un</th>
<th>Cr</th>
<th>Pd</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td>localhost</td>
<td>![icon]</td>
<td>localhost</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>Host</th>
<th>Icons</th>
<th>Alias</th>
<th>OK</th>
<th>Wa</th>
<th>Un</th>
<th>Cr</th>
<th>Pd</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td>compute-0-0</td>
<td>![icon]</td>
<td>compute-0-0</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DOWN</td>
<td>compute-0-1</td>
<td>![icon]</td>
<td>compute-0-1</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

*refresh: 30 secs*
Service checks

- Service are checked by the Nagios daemon at defined regular intervals (1 min. in OMD).
- Services that are checked can be in one of four different states:
  - OK
  - WARNING
  - UNKNOWN
  - CRITICAL
- Service checks are performed by plugins, which can return a state of OK, WARNING, UNKNOWN, or CRITICAL.
- When a service changes its state, Nagios takes appropriate action.

Detecting and dealing with state changes is what Nagios is all about.
host – services example

<table>
<thead>
<tr>
<th>State</th>
<th>Service</th>
<th>Icons</th>
<th>Status detail</th>
<th>Age</th>
<th>Checked</th>
<th>Perf-O-Meter</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Check_MK</td>
<td></td>
<td>OK - Agent version 1.2.2p3, execution time 0.9 sec</td>
<td>39 hrs</td>
<td>58 sec</td>
<td>0.9s</td>
</tr>
<tr>
<td>OK</td>
<td>CPU load</td>
<td></td>
<td>OK - 15min load 0.07 at 2 CPUs</td>
<td>37 hrs</td>
<td>57 sec</td>
<td>0.1</td>
</tr>
<tr>
<td>OK</td>
<td>CPU utilization</td>
<td></td>
<td>OK - user: 6.4%, system: 1.3%, wait: 0.7%</td>
<td>37 hrs</td>
<td>57 sec</td>
<td>8%</td>
</tr>
<tr>
<td>OK</td>
<td>Disk IO SUMMARY</td>
<td></td>
<td>OK - 0.00B/sec read, 58.33kB/sec write, IOs: 4.90/sec, Latency: 3.35ms</td>
<td>39 hrs</td>
<td>57 sec</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td>fs_/</td>
<td></td>
<td>OK - 34.2% used (5.26 of 15.4 GB), (levels at 80.0/90.0%), trend: +16.49MB / 24 hours</td>
<td>39 hrs</td>
<td>57 sec</td>
<td>34%</td>
</tr>
<tr>
<td>UNKN</td>
<td>fs_/state/partition1</td>
<td></td>
<td>Manually set to Unknown by omdadmin</td>
<td>53 sec</td>
<td>53 sec</td>
<td></td>
</tr>
<tr>
<td>WARN</td>
<td>fs_/var</td>
<td></td>
<td>Manually set to Warning by omdadmin</td>
<td>15 sec</td>
<td>15 sec</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td>HTTP check_http</td>
<td></td>
<td>HTTP OK; HTTP/1.1 200 OK - 467 bytes in 0.001 second response time</td>
<td>37 hrs</td>
<td>6 sec</td>
<td>0.5 ms</td>
</tr>
<tr>
<td>CRIT</td>
<td>Interface 2</td>
<td></td>
<td>Manually set to Critical by omdadmin</td>
<td>3 sec</td>
<td>3 sec</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td>Interface 3</td>
<td></td>
<td>OK - [peth1] (up) 100MB/s, in: 2.91kB/s(0.0%), out: 5.19kB/s(0.0%)</td>
<td>39 hrs</td>
<td>57 sec</td>
<td>0.0%</td>
</tr>
<tr>
<td>OK</td>
<td>Interface 4</td>
<td></td>
<td>OK - [eth0] (up) speed unknown, in: 660.33B/s, out: 223.93B/s</td>
<td>39 hrs</td>
<td>57 sec</td>
<td>660.3B/s 223.9B/s</td>
</tr>
</tbody>
</table>
Nagios Plugins (I)

Nagios itself does not provide any check. Everything is done through...

Plugins

Plugins are compiled executables or scripts (Perl scripts, shell scripts, etc.) that can be run from a command line to check the status or a host or service. Nagios uses the results from plugins to determine the current status of hosts and services on your network (typically named check_xxx).

How are services monitored:

- Nagios runs the plugin (eg. check_http).
- The plugin does “something” and gives the result back to Nagios.
- Nagios processes the result and takes the corresponding actions.
Nagios Plugins (II)

Plugin Structure

- `check_stuff [HostIP] [-w<warning level>] [-c<critical level>]`
- **return values:** 0 (OK), 1 (Warning), 2 (Critical), 3 (Unknown)
- **stdout:** message | optional performance data
- **Performance data:**
  - `label=value[UOM];[warn];[crit];[min];[max]`

Example:
```
# ./check_enviromux_mini.py 192.168.1.123 -w 35 -c 45 -s temperature1
OK - Temperature CRAC-1 sensor reading is 31.6 Celsius | Temperature_CRAC-1=31.6;35.0;45.0;0.;50.
```
Nagios Plugins (II)

Plugin Structure

- `check_stuff [HostIP] [-w<warning level>] [-c<critical level>]`
- **return values:** 0 (OK), 1 (Warning), 2 (Critical), 3 (Unknown)
- **stdout:** message | optional performance data
- **Performance data:**
  - `label=value[UOM];[warn];[crit];[min];[max]`

Example:

```sh
# ./check_enviromux_mini.py 192.168.1.123 -w 35 \   -c 45 -s temperature1
OK - Temperature CRAC-1 sensor reading is 31.6 Celsius|Temperature\_CRAC-1=31.6;35.0;45.0;0.;50.
```
Nagios Plugins (III)

There are plugins for:

- HTTP, POP3, IMAP, FTP, SSH, DHCP...
- CPU load, disk usage, memory usage, connected users, ...
- routers, switches...

Official Nagios plugins at http://nagiosplugins.org

Public repository for Nagios plugins at Nagios Exchange with ~ 3000 plugins, addons, utils, ... http://exchange.nagios.org/
Nagios Plugins (III)

There are plugins for:
- HTTP, POP3, IMAP, FTP, SSH, DHCP...
- CPU load, disk usage, memory usage, connected users, ...
- routers, switches...

Official Nagios plugins at [http://nagiosplugins.org](http://nagiosplugins.org)

Public repository for Nagios plugins at Nagios Exchange with ~ 3000 plugins, addons, utils, ... [http://exchange.nagios.org/](http://exchange.nagios.org/)

Do not reinvent the wheel! Search around for something similar first.
Nagios Plugins - Local check example (I)

#!/bin/bash

# Counts number of files in /tmp. Harcoded levels w=50, c=100.

count=$(ls -l /tmp | wc --lines)

if [ $count -lt 50 ]; then
  echo "OK - $count files in /tmp | /tmp=$count;50;100;;"
  exit 0
elif [ $count -lt 100 ]; then
  echo "WARNING - $count files in /tmp | /tmp=$count;50;100;;"
  exit 1
elif [ $count -ge 100 ]; then
  echo "CRITICAL - $count files in /tmp | /tmp=$count;50;100;;"
  exit 2
else
  echo "UNKNOWN - $count files in /tmp | /tmp=$count;50;100;;"
  exit 3
fi

# /usr/lib/check_mk_agent/local/filecount_tmp
CRITICAL - 126 files in /tmp | /tmp=126;50;100;;
# echo $?
2
Considerations

Pros

- Plugins very easy to write/adapt.
- Can monitor almost everything network connected (SNMP).
- Very flexible alerting system.
- A lot of existing plugins and addons.

Cons

- Hard to configure.
- Outdated and somehow confusing interface.
- Does not provide historical time series data, is “just“ an alerting system.
Nagios extensions leverage the “cons“:

- Hard to configure → NagiosQL, LConf, NConf, Centreon...
- Outdated and somehow confusing interface → Thruk, Centreon, GroundWork...
- Does not provide historical time series data → PNP4Nagios, nagiosgraph
- Visualization → NagVis

Even harder to configure!
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Why monitoring?</td>
</tr>
<tr>
<td></td>
<td>What to monitor?</td>
</tr>
<tr>
<td></td>
<td>How to monitor?</td>
</tr>
<tr>
<td>Nagios</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Active and passive monitoring checks, plugins and extensions</td>
</tr>
<tr>
<td></td>
<td>Considerations</td>
</tr>
<tr>
<td>Check_MK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Check_MK Agent</td>
</tr>
<tr>
<td></td>
<td>Architecture</td>
</tr>
<tr>
<td></td>
<td>Multisite front-end</td>
</tr>
<tr>
<td>OMD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Included Software/packages</td>
</tr>
<tr>
<td></td>
<td>Installation</td>
</tr>
<tr>
<td></td>
<td>OMD General Overview - Components</td>
</tr>
</tbody>
</table>

Iñigo Aldazabal
Introduction to system monitoring with Nagios, Check_MK and OMD
Check_MK is a collection of extensions for Nagios which together with PNP4Nagios and NagVis constitutes a complete, 100% Open Source, IT-Monitoring-System

Main components:

- **check_mk agent**: automatic service recognition and configuration generator.
- **Multisite**: web frontend.
- **Web Administration Tool - WATO**: complete administration of a Check_MK-based system over a browser.
- **Check_MK Event Console**: integrates the processing of log messages and SNMP-Traps into the monitoring.
NRPE multiple checks → just one check per host + passive checks in the monitoring server!

- Automatic service recognition.
- More than 300 included checks.
Architecture of a Check_MK based monitoring solution
Multisite web front-end

And all tied up by the “Multisite” web front end, giving access to all the components.
Check_MK Plugins - Local check example

#!/bin/sh
# /usr/lib/check_mk_agent/local/check_mk_dmraid
# Checks status of a dmraid disk array.

raid_status='dmraid -s | grep status | awk '{print $3}''
if [ "$raid_status" = "ok" ] ; then
    echo "0 dmraid - OK - RAID Status: ${raid_status}"
    exit 0
else
    raid_full_info='dmraid -s | paste -sd ","'
    disks_info='dmraid -r |paste -sd ";"'
    full_errror=${raid_full_info} ; ${disks_info}
    echo "2 dmraid - CRITICAL - RAID Status: "$raid_status" - "${raid_full_info} " *** Disks info --> "${disks_info}"
    exit 2
fi

# /usr/lib/check_mk_agent/local/check_mk_dmraid
0 dmraid - OK - RAID Status: ok
• Intro
  • Why monitoring?
  • What to monitor?
  • How to monitor?

• Nagios
  • Introduction
  • Active and passive monitoring
  • checks, plugins and extensions
  • Considerations

• Check_MK
  • Introduction
  • Check_MK Agent
  • Architecture
  • Multisite front-end

• OMD
  • Introduction
  • Included Software/packages
  • Installation
  • OMD General Overview - Components
OMD - The Open Monitoring Distribution

“Bundle” of Nagios based monitoring software, integrated and configured in such a way that greatly simplifies the installation, maintenance and update of the whole system. Prebuilt packages are provided for enterprise distributions.

Features:

- Multiple instances – *sites* – of the system in the same server (e.g. test and production sites).
- Separate operators/users per instance.
- Trivial creation of new sites.
- Support for concurrent different OMD versions in one server.
- ...
OMD Software

- Nagios
  - nagios-plugins
  - nsca
  - check_nrpe
- Icinga
- Shinken
- NagVis
- pnp4nagios
- rrdtool/rrdcached
- Check_MK
- MK Livestatus
- Multisite
- Dokuwiki
- Thruk
- ...

Iñigo Aldazabal
Introduction to system monitoring with Nagios, Check_MK and OMD
Installation example (SLES)

First install the package matching your operating system:

```bash
# zypper install omd-1.10-sles11sp3-31.x86_64.rpm
```

Now create a monitoring instance (OMD calls this a “site”):

```bash
# omd create foo
```

And let’s start the “site”, i.e. Nagios and all other processes (Nagios, apache, rrdcached...)

```bash
# omd start foo
```

And login to the “Multisite” we interface at http://localhost/foo and start adding hosts / services.
Components

**Check_MK**
- Automatic configuration
- Perform checks
- Status view
- Web-based administration
- Analyses
- Event processing

**NagVis**
- Visualisation
- Geo Maps

**PNP4Nagios**
- Historiography of performance data
- Resource planning

**OMD**
- Installation
- Updates
- Site-management (multi-location)
- Cluster-Management

*Components of the Check_MK Monitoring System*
References

Nagios


Check_MK


OMD