Network Monitoring & Management: Nagios

Network Startup Resource Center
www.nsrc.org

These materials are licensed under the Creative Commons Attribution-NonCommercial 4.0 International license (http://creativecommons.org/licenses/by-nc/4.0/)
Introduction

- Possibly the most used open source network monitoring software
- Web interface for viewing status, browsing history, scheduling downtime etc
- Sends out alerts via E-mail. Can be configured to use other mechanisms, e.g. SMS
Introduction

Nagios actively monitors the availability

• of Hosts (devices)

• and Services
# Nagios: General View

## Tactical Monitoring Overview
Last Updated: Tue Feb 16 20:13:28 UTC 2016
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as guest

### Network Outages
- **0 Outages**

### Network Health
- **Host Health:**
- **Service Health:**

### Monitoring Features

<table>
<thead>
<tr>
<th>Flap Detection</th>
<th>Notifications</th>
<th>Event Handlers</th>
<th>Active Checks</th>
<th>Passive Checks</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="checkmark" alt="All Services Enabled" /></td>
<td><img src="checkmark" alt="All Services Enabled" /></td>
<td><img src="checkmark" alt="All Services Enabled" /></td>
<td><img src="checkmark" alt="All Services Enabled" /></td>
<td><img src="checkmark" alt="All Services Enabled" /></td>
</tr>
<tr>
<td><img src="no_checkmark" alt="No Services Flapping" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
</tr>
<tr>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
</tr>
<tr>
<td><img src="no_checkmark" alt="No Hosts Flapping" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
<td><img src="checkmark" alt="All Hosts Enabled" /></td>
</tr>
</tbody>
</table>

### Monitoring Performance

| Service Check Execution Time: | 0.00 / 10.03 / 0.465 sec |
| Host Check Execution Time:   | 0.01 / 3.01 / 0.139 sec |
| Host Check Latency:          | 0.03 / 0.29 / 0.130 sec |
| # Active Host / Service Checks: | 51 / 221 |
| # Passive Host / Service Checks: | 0 / 0 |

### Hosts
- **2 Down**
- **0 Unreachable**
- **49 Up**
- **0 Pending**

### Services
- **8 Critical**
- **35 Warning**
- **2 Unknown**
- **176 Ok**
- **0 Pending**

### Problems
- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages
## Host Status Details For All Host Groups

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ap1</td>
<td>DOWN</td>
<td>2016-02-16 20:22:32</td>
<td>1d 10h 33m 41s</td>
<td>CRITICAL - Host Unreachable (ap1.ws.nsrc.org)</td>
</tr>
<tr>
<td>ap2</td>
<td>DOWN</td>
<td>2016-02-16 20:22:32</td>
<td>2d 7h 7m 6s</td>
<td>CRITICAL - Host Unreachable (ap2.ws.nsrc.org)</td>
</tr>
<tr>
<td>gw</td>
<td>UP</td>
<td>2016-02-16 20:22:32</td>
<td>95d 20h 46m 29s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.69 ms</td>
</tr>
<tr>
<td>localhost</td>
<td>UP</td>
<td>2016-02-16 20:22:32</td>
<td>95d 0h 43m 46s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.01 ms</td>
</tr>
<tr>
<td>pc1</td>
<td>UP</td>
<td>2016-02-16 20:22:42</td>
<td>2d 17h 59m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 19.04 ms</td>
</tr>
<tr>
<td>pc10</td>
<td>UP</td>
<td>2016-02-16 20:22:42</td>
<td>2d 17h 59m 10s</td>
<td>PING OK - Packet loss = 0%, RTA = 18.77 ms</td>
</tr>
<tr>
<td>pc11</td>
<td>UP</td>
<td>2016-02-16 20:22:52</td>
<td>2d 18h 3m 10s</td>
<td>PING OK - Packet loss = 0%, RTA = 13.16 ms</td>
</tr>
<tr>
<td>pc12</td>
<td>UP</td>
<td>2016-02-16 20:24:02</td>
<td>1d 18h 54m 59s</td>
<td>PING OK - Packet loss = 0%, RTA = 18.66 ms</td>
</tr>
<tr>
<td>pc13</td>
<td>UP</td>
<td>2016-02-16 20:23:02</td>
<td>2d 18h 3m 10s</td>
<td>PING OK - Packet loss = 0%, RTA = 15.75 ms</td>
</tr>
<tr>
<td>pc14</td>
<td>UP</td>
<td>2016-02-16 20:23:12</td>
<td>2d 18h 1m 50s</td>
<td>PING OK - Packet loss = 0%, RTA = 15.88 ms</td>
</tr>
<tr>
<td>pc15</td>
<td>UP</td>
<td>2016-02-16 20:23:12</td>
<td>2d 18h 3m 10s</td>
<td>PING OK - Packet loss = 0%, RTA = 16.08 ms</td>
</tr>
<tr>
<td>pc16</td>
<td>UP</td>
<td>2016-02-16 20:23:22</td>
<td>2d 17h 33m 10s</td>
<td>PING OK - Packet loss = 0%, RTA = 15.74 ms</td>
</tr>
<tr>
<td>pc17</td>
<td>UP</td>
<td>2016-02-16 20:23:32</td>
<td>2d 17h 39m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 10.98 ms</td>
</tr>
<tr>
<td>pc18</td>
<td>UP</td>
<td>2016-02-16 20:23:32</td>
<td>2d 17h 39m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 20.86 ms</td>
</tr>
<tr>
<td>pc19</td>
<td>UP</td>
<td>2016-02-16 20:23:42</td>
<td>2d 17h 39m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 16.92 ms</td>
</tr>
<tr>
<td>pc2</td>
<td>UP</td>
<td>2016-02-16 20:23:42</td>
<td>2d 18h 3m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 18.88 ms</td>
</tr>
<tr>
<td>pc20</td>
<td>UP</td>
<td>2016-02-16 20:23:52</td>
<td>2d 17h 39m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 12.05 ms</td>
</tr>
<tr>
<td>pc21</td>
<td>UP</td>
<td>2016-02-16 20:24:02</td>
<td>2d 17h 30m 20s</td>
<td>PING OK - Packet loss = 0%, RTA = 16.25 ms</td>
</tr>
<tr>
<td>pc22</td>
<td>UP</td>
<td>2016-02-16 20:24:02</td>
<td>2d 17h 39m 20s</td>
<td>PING OK - Packet loss = 0%, RTA = 15.97 ms</td>
</tr>
<tr>
<td>pc23</td>
<td>UP</td>
<td>2016-02-16 20:24:12</td>
<td>2d 17h 39m 20s</td>
<td>PING OK - Packet loss = 0%, RTA = 15.41 ms</td>
</tr>
<tr>
<td>pc24</td>
<td>UP</td>
<td>2016-02-16 20:24:12</td>
<td>2d 17h 39m 20s</td>
<td>PING OK - Packet loss = 0%, RTA = 15.80 ms</td>
</tr>
<tr>
<td>pc25</td>
<td>UP</td>
<td>2016-02-16 20:24:22</td>
<td>2d 17h 39m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 14.71 ms</td>
</tr>
</tbody>
</table>
### Current Network Status

- **Last Updated:** Tue Feb 16 20:29:55 UTC 2016
- **Updated every 90 seconds**
- **Nagios® Core™ 3.5.1 - www.nagios.org**
- **Logged in as guest**

#### Current Status

- **Tactical Overview**
- **Map**
- **Hosts**
- **Services**
- **Host Groups**
- **Summary**
- **Grid**
- **Problems**
- **Services (Unhandled)**
- **Hosts (Unhandled)**
- **Network Outages**

#### Reports

- **Availablity**
- **Trends**
- **Alerts**
- **History**
- **Summary**
- **Histogram**
- **Notifications**
- **Event Log**

#### System

- **Comments**
- **Downtime**
- **Process Info**
- **Performance Info**
- **Scheduling Queue**
- **Configuration**

### Host Status Totals

<table>
<thead>
<tr>
<th></th>
<th>Up</th>
<th>Down</th>
<th>Unreachable</th>
<th>Pending</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All Problems</strong></td>
<td>49</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>All Types</strong></td>
<td>51</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Service Status Totals

<table>
<thead>
<tr>
<th></th>
<th>Ok</th>
<th>Warning</th>
<th>Unknown</th>
<th>Critical</th>
<th>Pending</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All Problems</strong></td>
<td>170</td>
<td>35</td>
<td>2</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td><strong>All Types</strong></td>
<td>45</td>
<td></td>
<td></td>
<td>221</td>
<td></td>
</tr>
</tbody>
</table>

### Service Status Details For All Hosts

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempt</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>gw</td>
<td>SNMP</td>
<td>OK</td>
<td>2016-02-16 20:26:17</td>
<td>0d 14h 28m 38s</td>
<td>1/4</td>
<td>SNMP OK - &quot;Linux s1 3.19.0-49-generic #55~14.04.1-Ubuntu SMP Fri Jan 22 11:24:31 UTC 2016 x86_64&quot;</td>
</tr>
<tr>
<td>localhost</td>
<td>Current Load</td>
<td>OK</td>
<td>2016-02-16 20:26:33</td>
<td>1d 14h 3m 22s</td>
<td>1/4</td>
<td>OK - load average: 2.34, 1.90, 1.77</td>
</tr>
<tr>
<td>localhost</td>
<td>Current Users</td>
<td>OK</td>
<td>2016-02-16 20:26:47</td>
<td>96d 0h 46m 32s</td>
<td>1/4</td>
<td>USERS OK - 0 users currently logged in</td>
</tr>
<tr>
<td>localhost</td>
<td>Disk Space</td>
<td>OK</td>
<td>2016-02-16 20:28:03</td>
<td>96d 0h 45m 42s</td>
<td>1/4</td>
<td>DISK OK</td>
</tr>
<tr>
<td>localhost</td>
<td>HTTP</td>
<td>OK</td>
<td>2016-02-16 20:29:19</td>
<td>89d 13h 42m 38s</td>
<td>1/4</td>
<td>HTTP OK: HTTP/1.1 200 OK - 1141 bytes in 0.000 second response time</td>
</tr>
<tr>
<td>localhost</td>
<td>NTP</td>
<td>OK</td>
<td>2016-02-16 20:28:42</td>
<td>1d 18h 21m 13s</td>
<td>1/4</td>
<td>NTP OK: Offset -0.000597 secs</td>
</tr>
<tr>
<td>localhost</td>
<td>SSH</td>
<td>OK</td>
<td>2016-02-16 20:25:35</td>
<td>11d 10h 42m 52s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6 (protocol 2.0)</td>
</tr>
<tr>
<td>localhost</td>
<td>Total Processes</td>
<td>OK</td>
<td>2016-02-16 20:26:49</td>
<td>96d 0h 43m 12s</td>
<td>1/4</td>
<td>PROCES OK: 98 processes</td>
</tr>
<tr>
<td>pc1</td>
<td>HTTP</td>
<td>OK</td>
<td>2016-02-16 20:28:05</td>
<td>1d 0h 6m 50s</td>
<td>1/4</td>
<td>HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.034 second response time</td>
</tr>
<tr>
<td>pc1</td>
<td>NAGIOS</td>
<td>WARNING</td>
<td>2016-02-16 20:29:21</td>
<td>1d 0h 5m 34s</td>
<td>4/4</td>
<td>HTTP WARNING: HTTP/1.1 404 Not Found - 459 bytes in 0.032 second response time</td>
</tr>
<tr>
<td>pc1</td>
<td>NTP</td>
<td>OK</td>
<td>2016-02-16 20:25:53</td>
<td>1d 17h 59m 26s</td>
<td>1/4</td>
<td>NTP OK: Offset -0.001192 secs</td>
</tr>
<tr>
<td>pc1</td>
<td>SNMP</td>
<td>OK</td>
<td>2016-02-16 20:25:36</td>
<td>1d 18h 4m 19s</td>
<td>1/4</td>
<td>SNMP OK - &quot;Linux pc1.ws.nsrc.org 3.13.0-77-generic #121-Ubuntu SMP Wed Jan 20 10:56:59 UTC 2016 x86_64&quot;</td>
</tr>
<tr>
<td>pc1</td>
<td>SSH</td>
<td>OK</td>
<td>2016-02-16 20:26:51</td>
<td>2d 18h 2m 15s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6 (protocol 2.0)</td>
</tr>
<tr>
<td>pc10</td>
<td>HTTP</td>
<td>OK</td>
<td>2016-02-16 20:28:08</td>
<td>0d 19h 16m 49s</td>
<td>1/4</td>
<td>HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.034 second response time</td>
</tr>
</tbody>
</table>
Features

Utilizes topology to determine dependencies.

- Differentiates between what is *down* vs. what is *unreachable*. Avoids running unnecessary checks and sending redundant alarms.

Allows you to define how to send notifications based on combinations of:

- Contacts and lists of contacts
- Devices and groups of devices
- Services and groups of services
- Defined hours by persons or groups.
- The state of a service.
Plugins

Plugins are used to verify services and devices:

- Nagios architecture is simple enough that writing new plugins is fairly easy in the language of your choice.
- There are many, many plugins available (thousands).
  - http://exchange.nagios.org/
  - http://nagiosplugins.org/
Pre-installed Plugins for Ubuntu

/usr/lib/nagios/plugins

/etc/nagios-plugins/config
How Checks Work

- Periodically Nagios calls a plugin to test the state of each service. Possible responses are:
  - OK
  - WARNING
  - CRITICAL
  - UNKNOWN

- If a service is not OK it goes into a “soft” error state. After a number of retries (default 3) it goes into a “hard” error state. At that point an alert is sent.

- You can also trigger external event handlers based on these state transitions.
How Checks Work (Continued)

Parameters

- Normal checking interval
- Retry interval (i.e. when not OK)
- Maximum number of retries
- Time period for performing checks
- Time period for sending notifications

Scheduling

- Nagios spreads its checks throughout the time period to even out the workload
- Web UI shows when next check is scheduled
Hierarchy: The Concept of Parents

Hosts can have parents:

- The parent of a **PC** connected to a **switch** would be the **switch**.
- Allows us to specify the dependencies between devices.
- Avoids sending alarms when parent does not respond.
- A node can have multiple parents (dual homed).
Network Viewpoint

• Where you locate your Nagios server will determine your point of view of the network.

• The Nagios server becomes the “root” of your dependency tree
Collapsed Tree Network View
Demo of Nagios

http://noc.ws.nsrc.org/nagios3/
nagiosadmin: lab_password
Installation

In Debian/Ubuntu

```
# apt-get install nagios3
```

Key directories

```
/etc/nagios3
/etc/nagios3/conf.d
/etc/nagios-plugins/config
/usr/lib/nagios/plugins
/usr/share/nagios3/htdocs/images/logos
```

Nagios web interface is here:

```
http://pcN.ws.nsnc.org/nagios3/
```
Host and Services Configuration

Based on templates

- This saves lots of time avoiding repetition

There are default templates with default parameters for a:

- generic host (generic-host_nagios2.cfg)
- generic service (generic-service_nagios2.cfg)

• Individual settings can be overridden
• Defaults are all sensible
Configuration

• Configuration defined in text files
  • /etc/nagios3/conf.d/*.cfg
  • Details at http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html

• The default config is broken into several files with different objects in different files, but actually you can organise it how you like

• Always verify before restarting Nagios – otherwise your monitoring system may die!
  • nagios3 –v /etc/nagios3/nagios.cfg
Monitoring a Single Host

`pcs.cfg`

define host {
    host_name  pcl
    alias     pcl in group 1
    address   pcl.ws.nsre.org
    use       generic-host
}

- This is a minimal working config
- You are just pinging the host; Nagios will warn that you are not monitoring any services
- The filename can be anything ending `.cfg`
- Organise your devices however you like – e.g. related hosts in the same file
define host {
    name  generic-host ; The name of this host template
    notifications_enabled  1 ; Host notifications are enabled
    event_handler_enabled  1 ; Host event handler is enabled
    flap_detection_enabled  1 ; Flap detection is enabled
    failure_prediction_enabled  1 ; Failure prediction is enabled
    process_perf_data  1 ; Process performance data
    retain_status_information  1 ; Retain status information across program restarts
    retain_nonstatus_information  1 ; Retain non-status information across restarts
    check_command  check-host-alive
    max_check_attempts  10
    notification_interval  0
    notification_period  24x7
    notification_options  d,u,r
    contact_groups  admins
    register  0 ; DON’T REGISTER THIS DEFINITION —
                   ; IT’S NOT A REAL HOST, JUST A TEMPLATE!
}

Overriding Defaults

All settings can be overridden per host

```
pcs.cfg

define host {
    host_name               pcl
    alias                   pcl in group 1
    address                 pcl.ws.nsric.org
    use                     generic-host
    notification_interval   120
    contact_groups          admins,managers
}
```
Defining Services: Direct Way

define host {
    host_name pcl
    alias pcl_ingroup
    address pcl.ws.nsric.org
    use generic-host
}

define service {
    host_name pcl
    service_description HTTP
    check_command check_http
    use generic-service
}

define service {
    host_name pcl
    service_description SSH
    check_command check_ssh
    use generic-service

    service 'pcl, HTTP'
    plugin
    service template

    pcs.cfg

Service Checks

• The combination of host + service is a unique identifier for the service check, e.g.
  • “pc1,HTTP”
  • “pc1,SSH”
  • “pc2,HTTP”
  • “pc2,SSH”

• `check_command` points to the plugin

• `service template` pulls in settings for how often the check is done, and who and when to alert
define service{
    name                            generic-service
    active_checks_enabled           1
    passive_checks_enabled          1
    parallelize_check               1
    obsess_over_service             1
    check_freshness                 0
    notifications_enabled           1
    event_handler_enabled           1
    flap_detection_enabled          1
    failure_prediction_enabled      1
    process_perf_data               1
    retain_status_information       1
    retain_nonstatus_information    1
    notification_interval           0
    is_volatile                     0
    check_period                    24x7
    normal_check_interval           5
    retry_check_interval            1
    max_check_attempts              4
    notification_period             24x7
    notification_options            w,u,c,r
    contact_groups                  admins
    register                        0    ; DONT REGISTER THIS DEFINITION
}
Overriding Defaults

Again, settings can be overridden per service

services_nagios2.cfg

```define service {
    host_name            pcl
    service_description  HTTP
    check_command        check_http
    use                  generic-service
    contact_groups       admins,managers
    max_check_attempts   3
}
```
Repeating Service Checks

- Often we are monitoring an identical service on many hosts.
- To avoid duplication, a better way is to define a service check for all hosts in a *hostgroup*.
Creating Hostgroups

hostgroups_nagios2.cfg

define hostgroup {
    hostgroup_name   http-servers
    alias            HTTP servers
    members          pc1,pc2
}

define hostgroup {
    hostgroup_name   ssh-servers
    alias            SSH servers
    members          pc1,pc2
}
Monitoring Services in Hostgroups

```plaintext
define service {
    hostgroup_name       http-servers
    service_description  HTTP
    check_command        check_http
    use                  generic-service
}

define service {
    hostgroup_name       ssh-servers
    service_description  SSH
    check_command        check_ssh
    use                  generic-service
}
```

`services_nagios2.cfg`

*if hostgroup “http-servers” contains pc1 & pc2 then Nagios creates HTTP service checks for both hosts. The service checks are called “pc1,HTTP” and “pc2,HTTP”*
Alternative View

• “this hostgroup contains these PCs”
  
or:

• “this PC belongs to these hostgroups”

• No need for “members” line in hostgroups file
Alternative Group Membership

```pcs.cfg

define host {
    host_name    pc1
    alias        pc1 in group 1
    address      pc1.ws.nsrc.org
    use          generic-host
    hostgroups   ssh-servers,http-servers
}

define host {
    host_name    pc2
    alias        pc2 in group 1
    address      pc2.ws.nsrc.org
    use          generic-host
    hostgroups   ssh-servers,http-servers
}
```

Hosts and services conveniently defined in the same place
Other Uses for Hostgroups

Choosing icons for the status map

```conf
define host {
    host_name    pc1
    alias        pc1 in group 1
    address      pc1.ws.nsnc.org
    use          generic-host
    hostgroups   ssh-servers,http-servers,debian-servers
}
```

```conf
define hostextinfo {
    hostgroup_name     debian-servers
    notes               Debian GNU/Linux servers
    icon_image          base/debian.png
    statusmap_image     base/debian.gd2
}
```
Optional: Servicegroups

- Services can be grouped into a “servicegroup”
- This is so related or dependent services can be viewed together in the web interface
- The services themselves must already exist

```plaintext
define servicegroup {
    servicegroup_name   mail-services
    alias       Services comprising the mail platform
    members     web1,HTTP,web2,HTTP,mail1,IMAP,db1,MYSQL
}
```
Configuring Topology

```
define host {
    host_name   pc1
    alias       pc1 in group 1
    address     pc1.ws.nsrc.org
    use         generic-host
    parents     rtr1
}
```

- This means “pc1 is on the far side of rtr1”
- If rtr1 goes down, pc1 is “unreachable”, not “down”
- Prevents a cascade of alerts if rtr1 goes down
- Also allows Nagios to draw cool status map
Another View of Configuration

**RTR**
declare host {
    use
generic-host
    host_name rtr
    alias Gateway Router
    address 10.10.0.254
}

**SW**
declare host {
    use
generic-host
    host_name sw
    alias Backbone Switch
    address 10.10.0.253
    parents rtr
}

**RTR3**
declare host {
    use
generic-host
    host_name rtr3
    alias router 3
    address 10.10.3.254
    parents sw
}

**PC11...**
Out of Band (OOB) Notifications

A critical item to remember: an SMS or message system that is independent from your network.

- You can utilize a cell phone connected to the Nagios server, or a USB dongle with SIM card
- You can use packages like:
  
gammu: http://wammu.eu/
gnokii: http://www.gnokii.org/
sms-tools: http://smstools3.kekekasvi.com/

I use a Raspberry Pi with Kannel: http://www.kannel.org/
NOTE: The flow will only continue when each of the listed filters are satisfied.
References

• Nagios web site
  http://www.nagios.org/

• Nagios plugins site
  http://www.nagiosplugins.org/


• Unofficial Nagios plugin site
  http://nagios.exchange.org/

• A Debian tutorial on Nagios
  http://www.debianhelp.co.uk/nagios.htm

• Commercial Nagios support
  http://www.nagios.com/
Additional Details

A few additional slides you may find useful or informative...
More Features

• Allows you to acknowledge an event.
  – A user can add comments via the GUI

• You can define maintenance periods
  – By device or a group of devices

• Maintains availability statistics and generates reports

• Can detect flapping and suppress additional notifications.

• Allows for multiple notification methods:
  – e-mail, pager, SMS, winpopup, audio, etc...

• Allows you to define notification levels for escalation
Host Notification Options

Host state:
When configuring a host you can be notified on the following conditions:

- **d:** DOWN
- **u:** UNREACHABLE
- **r:** RECOVERY
- **f:** FLAPPING (start/end)
- **s:** SCHEDULED DOWNTIME (start/end)
- **n:** NONE
Service Notification Options

Service state:

When configuring a service you can be notified on the following conditions:

- **w**: WARNING
- **c**: CRITICAL
- **u**: UNKNOWN
- **r**: RECOVERY
- **f**: FLAPPING (start/end)
- **s**: SCHEDULED DOWNTIME (start/end)
- **n**: NONE
Configuration Files

- **nagios.cfg**
  - Main config file that defines other files, logging, events etc.
  - Interprets files using macros defined in other config files.

- **resource.cfg**
  - Macros referred to in other files.
  - E.g. $USER1$ = nagios home dir.

- **httpd.conf**
  - Apache include file describing how to display nagios web pages.
  - Points apache at the files secured by other config files.

- **cgi.cfg**
  - Parameters customizing the action of the web pages.
  - Points apache at the files secured by other config files.

- **.htaccess**
  - Apache security definition detailing allowed users.

- **htpasswd.users**
  - People authorized to in parts of nagios and their passwords.

- **Nagios Daemons**
  - Interprets files using macros defined in other config files.
  - Calls to other config files.

- **.log**
  - System log files.

- **checkcommands.cfg**
  - How to check hosts and services are working.

- **check.command**
  - How to check hosts and services are working.

- **dependencies.cfg**
  - Sometimes things wrong appear dead because something else broke.

- **checkcommand.cfg**
  - How to check hosts and services are working.

- **interfaces.cfg**
  - Configures network interfaces.

- **interfaces2.cfg**
  - Configures network interfaces.

- **hosts.cfg**
  - Central monitoring component along with services.
  - Groups hosts into types for alerting and display purposes.

- **hostgroups.cfg**
  - Groups hosts into types for alerting and display purposes.

- **contactgroups.cfg**
  - Contacts are grouped since we need some support redundancy.

- **escalations.cfg**
  - If something breaks and is not fixed who can we complain to next.

- **contacts.cfg**
  - People that we can call on to fix hosts and services.
  - Are maintained by.
  - Will work the following hours.

- **timeperiods.cfg**
  - For example "work hours" or "24 x 7" or "awards".

- **services.cfg**
  - Central monitoring component along with services.

- **checkcommands.cfg**
  - How to check hosts and services are working.
Debian/Ubuntu Configuration Files

Located in /etc/nagios3/

Important files include:

- **nagios.cfg** Main configuration file.
- **cgi.cfg** Controls the web interface and security options.
- **commands.cfg** The commands that Nagios uses for notifications.
- **conf.d/*** All other configuration goes here!
More Configuration Files

Under conf.d/*

- contacts_nagios2.cfg — users and groups
- extinfo_nagios2.cfg — make your UI pretty
- generic-host_nagios2.cfg — default host template
- generic-service_nagios2.cfg — default service template
- host-gateway_nagios3.cfg — upstream router definition
- hostgroups_nagios2.cfg — groups of nodes
- localhost_nagios2.cfg — definition of nagios host
- services_nagios2.cfg — what services to check
- timeperiods_nagios2.cfg — when to check who to notify
More Configuration Files

Under conf.d some other possible config files:

- **servicegroups.cfg**  Groups of nodes and services
- **pcs.cfg**          Sample definition of PCs (hosts)
- **switches.cfg**     Definitions of switches (hosts)
- **routers.cfg**      Definitions of routers (hosts)
Main Configuration Details

Global settings

**File:** /etc/nagios3/nagios.cfg

- Says where other configuration files are.
- General Nagios behavior:
  - For large installations you should tune the installation via this file.
  - See: *Tuning Nagios for Maximum Performance*
    http://nagios.sourceforge.net/docs/3_0/tuning.html
CGI Configuration

/etc/nagios3/cgi.cfg

- You can change the CGI directory if you wish
- Authentication and authorization for Nagios use:
  - Activate authentication via Apache's .htpasswd mechanism, or using RADIUS or LDAP.
  - Users can be assigned rights via the following variables:
    - authorized_for_system_information
    - authorized_for_configuration_information
    - authorized_for_system_commands
    - authorized_for_all_services
    - authorized_for_all_hosts
    - authorized_for_all_service_commands
    - authorized_for_all_host_commands
Time Periods

This defines the base periods that control checks, notifications, etc.

- Defaults: 24 x 7
- Adjust as needed, such as work-week only.
- Set up new time period for “outside regular hours”, etc.

```
# '24x7'
define timeperiod{
  timeperiod_name 24x7
  alias 24 Hours A Day, 7 Days A Week
  sunday 00:00-24:00
  monday 00:00-24:00
  tuesday 00:00-24:00
  wednesday 00:00-24:00
  thursday 00:00-24:00
  friday 00:00-24:00
  saturday 00:00-24:00
}
```
Define command {
  command_name check_ssh
  command_line /usr/lib/nagios/plugins/check_ssh '$HOSTADDRESS$
}
define command {
  command_name check_ssh_port
  command_line /usr/lib/nagios/plugins/check_ssh -p '$ARG1$' '$HOSTADDRESS$
}

• Notice the same plugin can be invoked in different ways (“commands”)
• Command and arguments are separated by exclamation marks (!)
• e.g. to check SSH on a non-standard port, you can do it like this:

define service {
  hostgroup_name ssh-servers-2222
  service_description SSH-2222
  check_command check_ssh_port!2222
  use generic-service
}

/etc/nagios-plugins/config/ssh.cfg
Notification Commands

Use any command you want!

We could use this to generate tickets in RT.

```
# 'notify-by-email' command definition
define command{
    command_name   notify-by-email
    command_line   /usr/bin/printf "%b" "Service: $SERVICEDESC$
                    Host: $HOSTNAME$
                    In: $HOSTALIAS$
                    Address: $HOSTADDRESS$
                    State: $SERVICESTATE$
                    Info: $SERVICEOUTPUT$
                    Date: $SHORTDATETIME$" | /bin/mail -s '$NOTIFICATIONTYPE$:
                          $HOSTNAME$/$SERVICEDESC$ is $SERVICESTATE$' $CONTACTEMAIL$
}
```

From: nagios@nms.localdomain
To: router_group@localdomain
Subject: Host DOWN alert for TLD1-RTR!
Date: Thu, 29 Jun 2006 15:13:30 -0700

Host: gw
In: Core_Routers
State: DOWN
Address: 192.0.2.100
Date/Time: 06-29-2006 15:13:30
Info: CRITICAL - Plugin timed out after 6 seconds
Group Service Configuration

```plaintext
# check that ssh services are running
define service {
    hostgroup_name           ssh-servers
    service_description      SSH
    check_command            check_ssh
    use                      generic-service
    notification_interval    0
}
```

The "service_description" is important if you plan to create Service Groups. Here is a sample Service Group definition:

```plaintext
define servicegroup{
    servicegroup_name Webmail
    alias              web-mta-storage-auth
    members            srvr1,HTTP,srvr1,SMTP,srvr1,POP, \ 
                        srvr1,IMAP,srvr1,RAID,srvr1,LDAP, \ 
                        srvr2,HTTP,srvr2,SMTP,srvr2,POP, \ 
                        srvr2,IMAP,srvr2,RAID,srvr2,LDAP
}
```
Screen Shots

A few sample screen shots from a Nagios install.
### Status Summary For All Host Groups

<table>
<thead>
<tr>
<th>Host Group</th>
<th>Host Status Summary</th>
<th>Service Status Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Servers (all)</td>
<td>49 Up</td>
<td>176 OK</td>
</tr>
<tr>
<td>Access Points (aps)</td>
<td>2 DOWN : 2 Unhandled</td>
<td>8 CRITICAL : 8 Unhandled</td>
</tr>
<tr>
<td>Cisco 7200 Routers (cisco7200)</td>
<td>10 UP</td>
<td>2 UNKNOWN : 2 on Problem Hosts</td>
</tr>
<tr>
<td>Debian GNU/Linux Servers (debian-servers)</td>
<td>1 UP</td>
<td>7 OK</td>
</tr>
<tr>
<td>HTTP servers (http-servers)</td>
<td>37 UP</td>
<td>35 WARNING : 35 Unhandled</td>
</tr>
<tr>
<td>Mac Mini (mac-servers)</td>
<td>1 UP</td>
<td>1 CRITICAL : 1 Unhandled</td>
</tr>
<tr>
<td>NAGIOS Servers (nagios-servers)</td>
<td>36 UP</td>
<td>35 WARNING : 35 Unhandled</td>
</tr>
<tr>
<td>NTP Clients (ntp-servers)</td>
<td>47 UP</td>
<td>35 WARNING : 35 Unhandled</td>
</tr>
<tr>
<td>SNMP Servers (snmp-servers)</td>
<td>48 UP</td>
<td>35 WARNING : 35 Unhandled</td>
</tr>
</tbody>
</table>
More Sample Screenshots

Many more sample Nagios screenshots available here:

http://www.nagios.org/about/screenshots
Nagios Lab Classroom Topology