SANOG 16

July 15th - 19th 2010 - Paro, Bhutan

GZ Kabir <gzkabir@bdcom.com>

Network Monitoring and Management

Slides Adopted From : Hervey Allen and Phil Regnauld, APRICOT - 2010

INTRODUCTION

- Nagios is a powerful monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes
- A key measurement tool for actively monitoring availability of devices and services.
- Possibly the most used open source network monitoring software.
- Has a web interface.
- Can support up to thousands of devices and services, if properly configured.

How Nagios Works

- Checks services
- If the service is down, checks the host
- If the host is down, checks its parent
- Find the highest-level thing that's down
- Retest it a few times (e.g. 5 or 10 times)
- If it stays down:
  - Figure out who to notify
  - Send them a message
  - Keep notifying them until it comes back up
- **It nags us!**

What Nagios Provides

By using Nagios, one can:

- Plan for infrastructure upgrades before outdated systems cause failures
- Respond to issues at the first sign of a problem
- Automatically fix problems when they are detected
- Coordinate technical team responses
- Ensure your organization's SLAs are being met
- Ensure IT infrastructure outages have a minimal effect on your organization's bottom line
- Monitor your entire infrastructure and business processes

Network Monitoring Software – Commercial

- Big Brother
  - Old, doesn't scale very well, shell scripts
- HP OpenView
  - Huge, complex, difficult to extend, proprietary
- Userful DiscoverStation 4.0
  - Does it still exist?
- IBM Tivoli, BMC patrol
  - Very large management & monitoring suites

Network Monitoring Software – Open Source

- OpenNMS
  - Java, increasingly popular, but complex
- ZenOSS
  - Automatic discovery, inventory, VMware monitoring
- Zabbix
  - Looks interesting, does capacity planning
- Hyperic
  - Targeted at application monitoring

Features of Nagios

- Small, relatively easy to understand
- Lightweight and fast
- Easy to extend (write new plugins)
- Large library of agent-less monitoring plugins
- Agents for Windows and most Unixes
- Free and open source

… but it doesn't do graphing

… configuration can be cumbersome

Installing Nagios in FreeBSD

- Install the packages for Nagios itself and the common plugins:
  - ***pkg_add -r nagios nagios-plugins***
- Enable Nagios by adding the following line to */etc/rc.conf*:
  - ***nagios_enable="YES"***
- Make sure Apache is installed and running – in */etc/rc.conf* if not present:
  - ***apache22_enable="YES"***
  - And then start Apache with this command:
  - ***sudo /usr/local/etc/rc.d/apache22 start***

Installing Nagios in FreeBSD

- Install the packages for Nagios itself and the common plugins:
  - ***pkg_add -r nagios nagios-plugins***
- Enable Nagios by adding the following line to */etc/rc.conf*:
  - ***nagios_enable="YES"***
- Make sure Apache is installed and running – in */etc/rc.conf* if not present:
  - ***apache22_enable="YES"***
  - And then start Apache with this command:
  - ***sudo /usr/local/etc/rc.d/apache22 start***

Installing Nagios in Ubuntu/Debian

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

- Files are installed here:

```
/etc/nagios3
/etc/nagios3/conf.d
/etc/nagios-plugins/conf
/usr/share/nagios3/htdocs/images/logos
/usr/sbin/nagios3
/usr/sbin/nagios3stats
```

Nagios web interface is here:

http://localhost/nagios3/

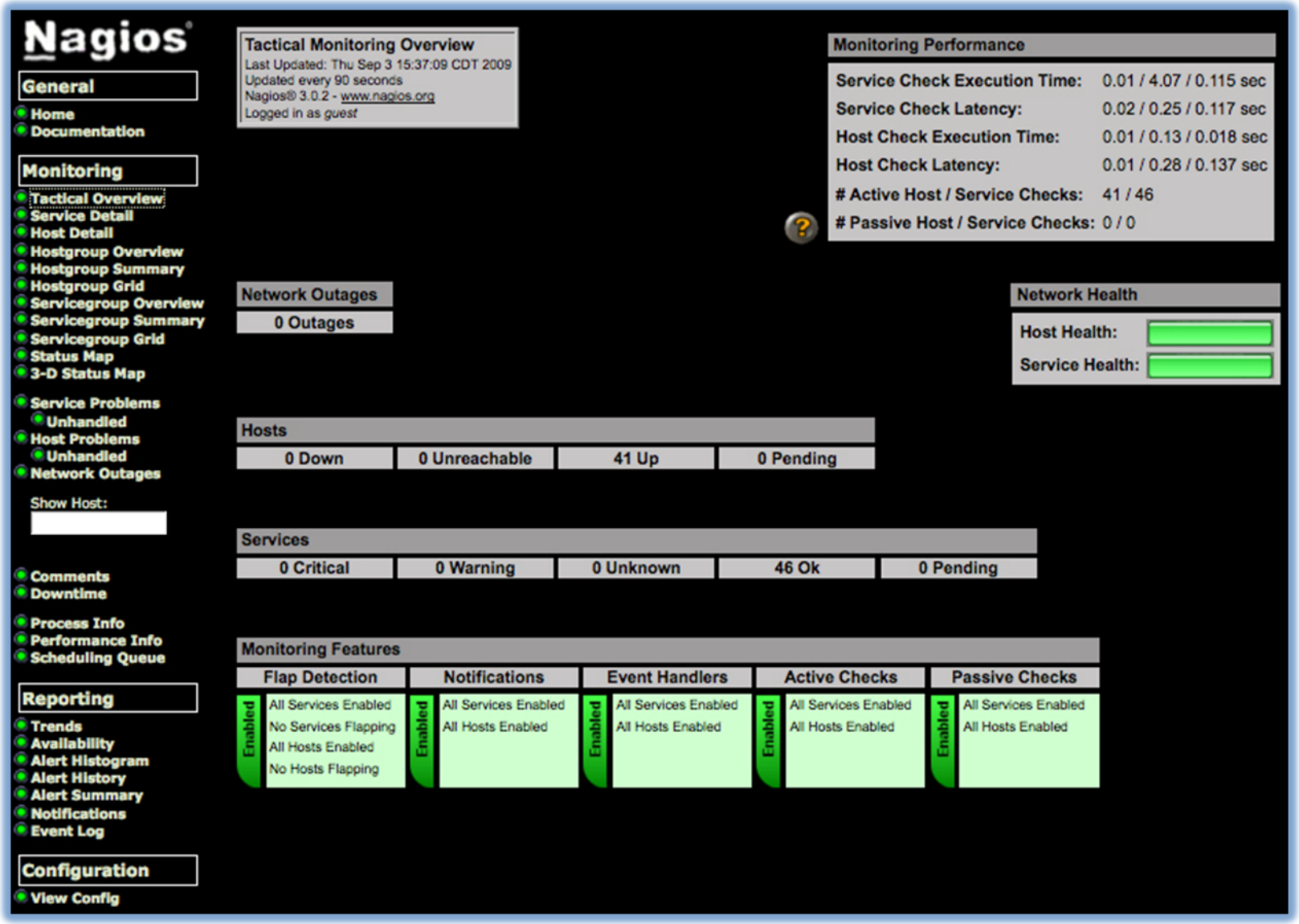
Configuring Apache (1)

- Create */usr/local/etc/apache22/Includes/nagios.conf* with the following lines:
  - ```
    <Directory /usr/local/www/nagios>
        Order deny,allow
        Deny from all
        Allow from 127.0.0.1
        Allow from 196.200.219.0/24
    </Directory>
    ```
  - ```
    <Directory /usr/local/www/nagios/cgi-bin>
        Options ExecCGI
    </Directory>
    ```
  - ```
    ScriptAlias /nagios/cgi-bin/
    /usr/local/www/nagios/cgi-bin/
    ```
  - ```
    Alias /nagios/ /usr/local/www/nagios/
    ```

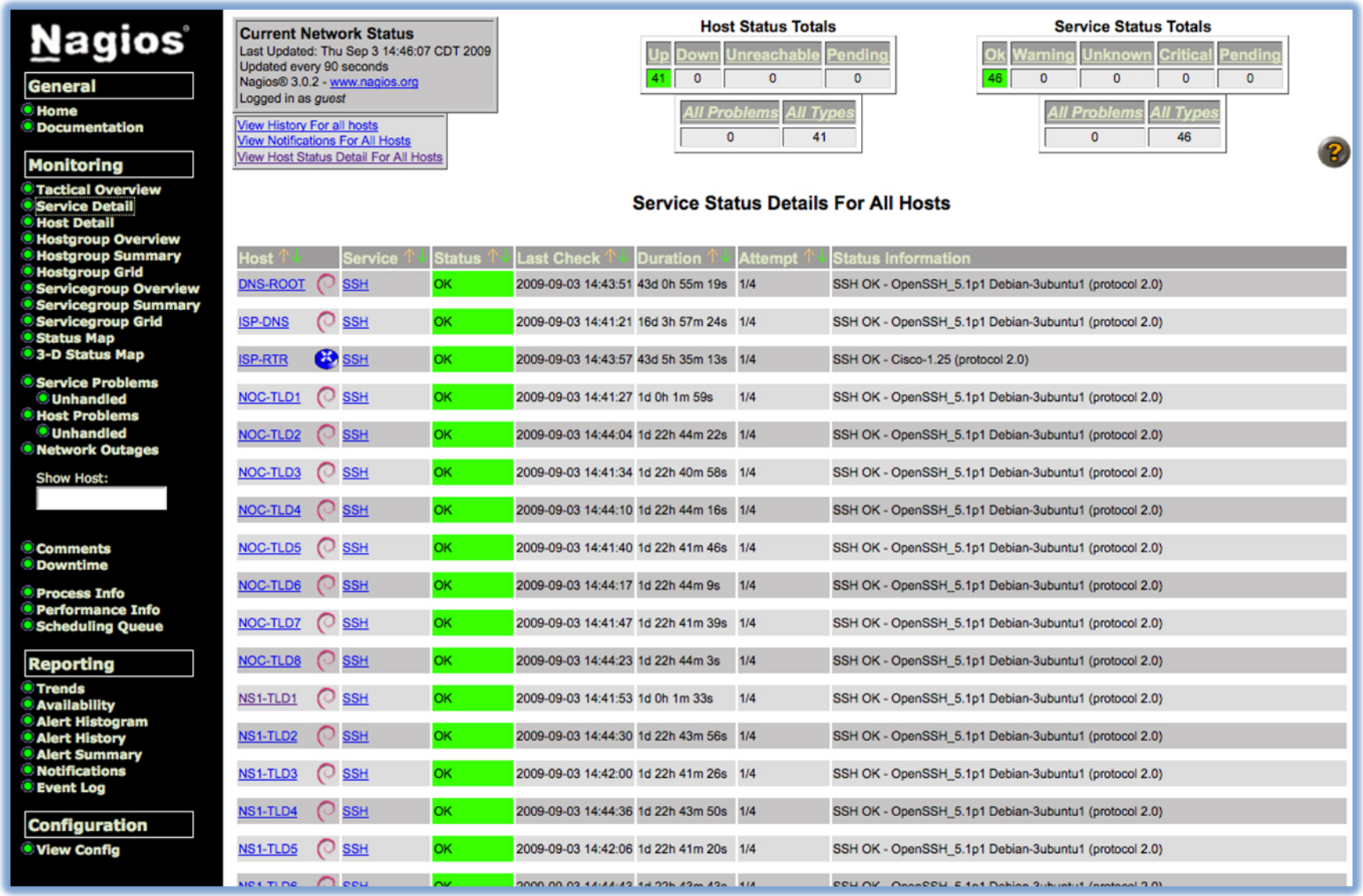
Configuring Apache (2)

- Tell Apache to reload its configuration:
  - ***sudo /usr/local/etc/rc.d/apache22 reload***
  - `Performing sanity check on apache22 configuration:`
  - `Syntax OK`
  - `Performing a graceful restart`
- Browse to http://localhost/nagios/:
  - Should show “Version 3.x.x”

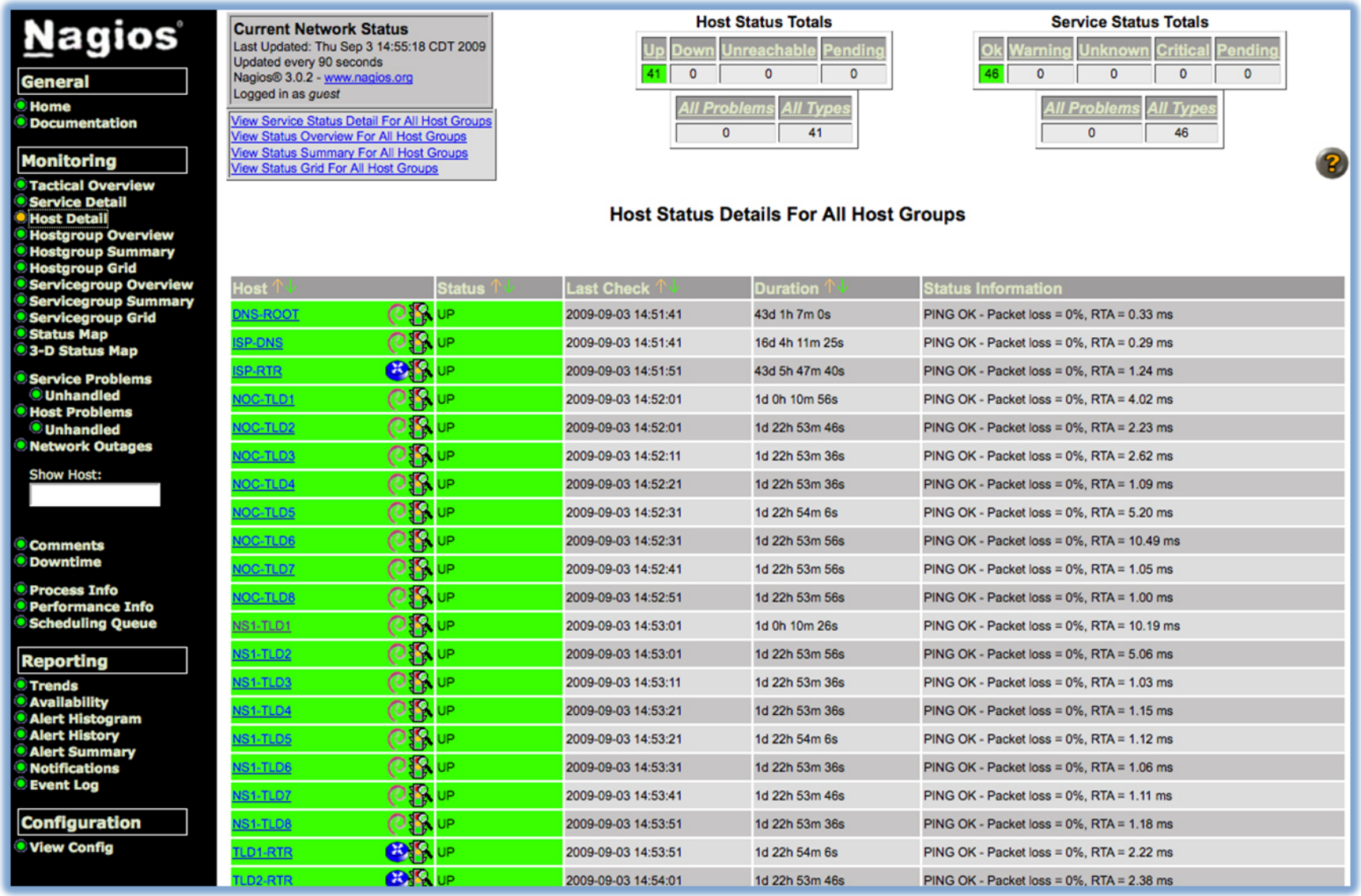
General View



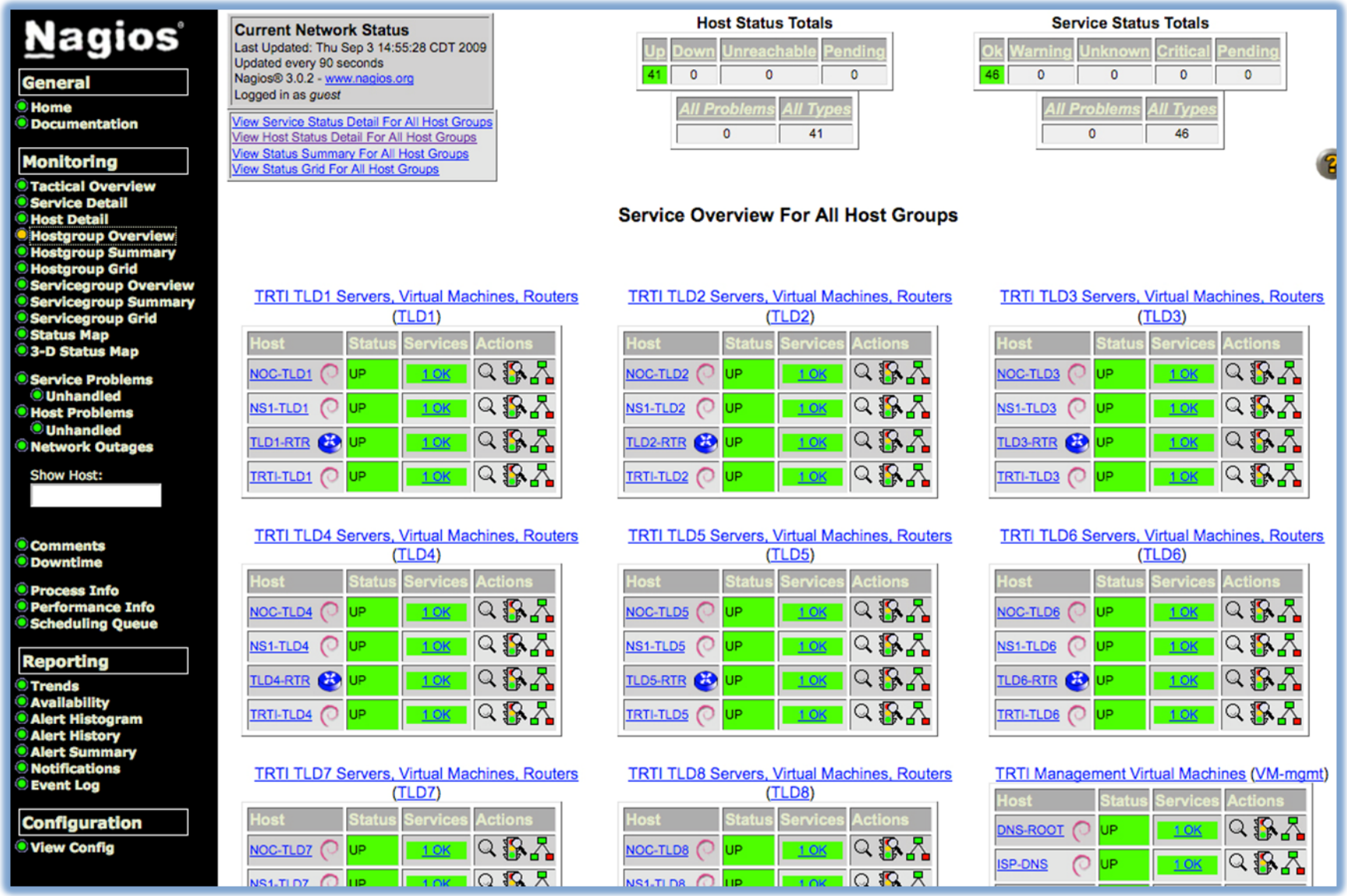
Service Detail



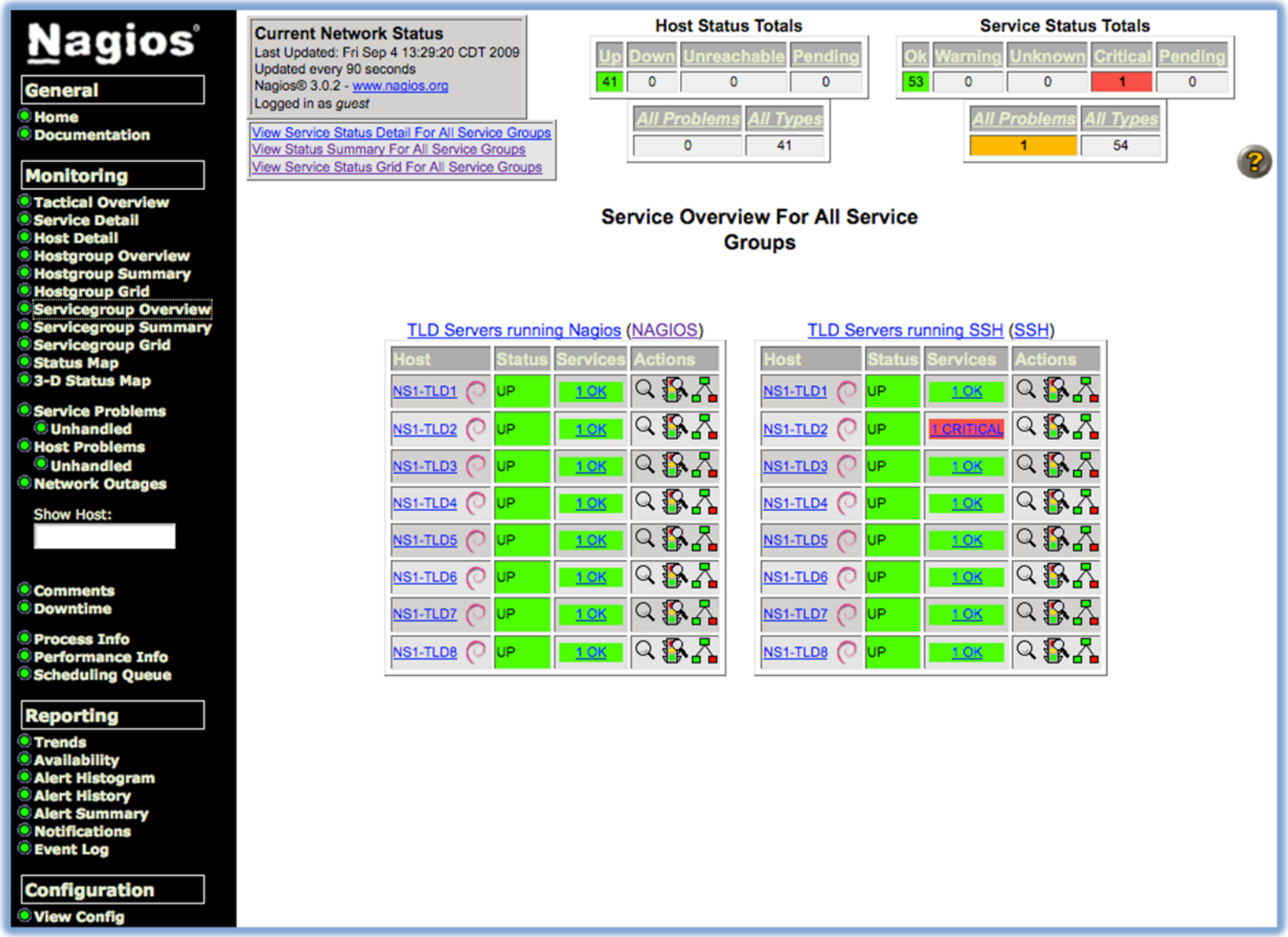
Host Detail


Nagios
General
Home
Documentation
Monitoring
Tactical Overview
Service Detail
Host Detail
Hostgroup Overview
Hostgroup Summary
Hostgroup Grid
Servicegroup Overview
Servicegroup Summary
Servicegroup Grid
Status Map
3-D Status Map
Service Problems
Unhandled
Host Problems
Unhandled
Network Outages
Show Host:
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Alert Summary
Notifications
Event Log
Configuration
View Config
Current Network Status
Last Updated: Thu Sep 3 14:55:18 CDT 2009
Updated every 90 seconds
Nagios® 3.0.2 - www.nagios.org
Logged in as guest
View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
41 0 0 0
All Problems All Types
0 41
Service Status Totals
Ok Warning Unknown Critical Pending
46 0 0 0 0
All Problems All Types
0 46
Host Status Details For All Host Groups
Host Status Last Check Duration Status Information
DNS-ROOT UP 2009-09-03 14:51:41 43d 1h 7m 0s PING OK - Packet loss = 0%, RTA = 0.33 ms
ISP-DNS UP 2009-09-03 14:51:41 16d 4h 11m 25s PING OK - Packet loss = 0%, RTA = 0.29 ms
ISP-RTR UP 2009-09-03 14:51:51 43d 5h 47m 40s PING OK - Packet loss = 0%, RTA = 1.24 ms
NOC-TLD1 UP 2009-09-03 14:52:01 1d 0h 10m 56s PING OK - Packet loss = 0%, RTA = 4.02 ms
NOC-TLD2 UP 2009-09-03 14:52:01 1d 22h 53m 46s PING OK - Packet loss = 0%, RTA = 2.23 ms
NOC-TLD3 UP 2009-09-03 14:52:11 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 2.62 ms
NOC-TLD4 UP 2009-09-03 14:52:21 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.09 ms
NOC-TLD5 UP 2009-09-03 14:52:31 1d 22h 54m 6s PING OK - Packet loss = 0%, RTA = 5.20 ms
NOC-TLD6 UP 2009-09-03 14:52:31 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 10.49 ms
NOC-TLD7 UP 2009-09-03 14:52:41 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 1.05 ms
NOC-TLD8 UP 2009-09-03 14:52:51 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 1.00 ms
NS1-TLD1 UP 2009-09-03 14:53:01 1d 0h 10m 26s PING OK - Packet loss = 0%, RTA = 10.19 ms
NS1-TLD2 UP 2009-09-03 14:53:01 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 5.06 ms
NS1-TLD3 UP 2009-09-03 14:53:11 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.03 ms
NS1-TLD4 UP 2009-09-03 14:53:21 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.15 ms
NS1-TLD5 UP 2009-09-03 14:53:21 1d 22h 54m 6s PING OK - Packet loss = 0%, RTA = 1.12 ms
NS1-TLD6 UP 2009-09-03 14:53:31 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.06 ms
NS1-TLD7 UP 2009-09-03 14:53:41 1d 22h 53m 46s PING OK - Packet loss = 0%, RTA = 1.11 ms
NS1-TLD8 UP 2009-09-03 14:53:51 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.18 ms
TLD1-RTR UP 2009-09-03 14:53:51 1d 22h 54m 6s PING OK - Packet loss = 0%, RTA = 2.22 ms
TLD2-RTR UP 2009-09-03 14:54:01 1d 22h 53m 46s PING OK - Packet loss = 0%, RTA = 2.38 ms

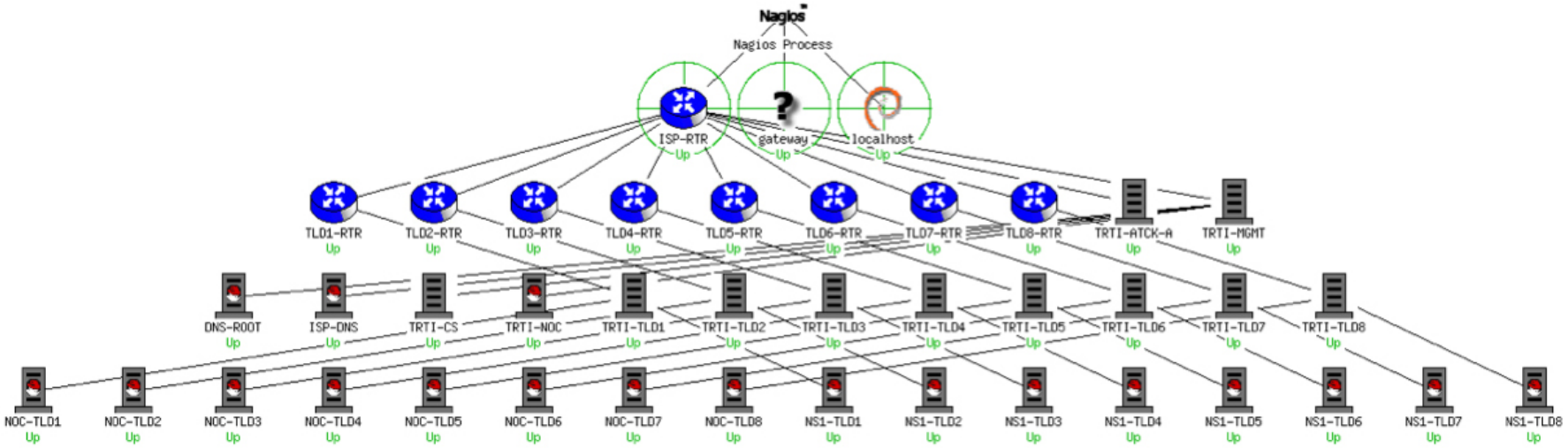
Host Groups Overview



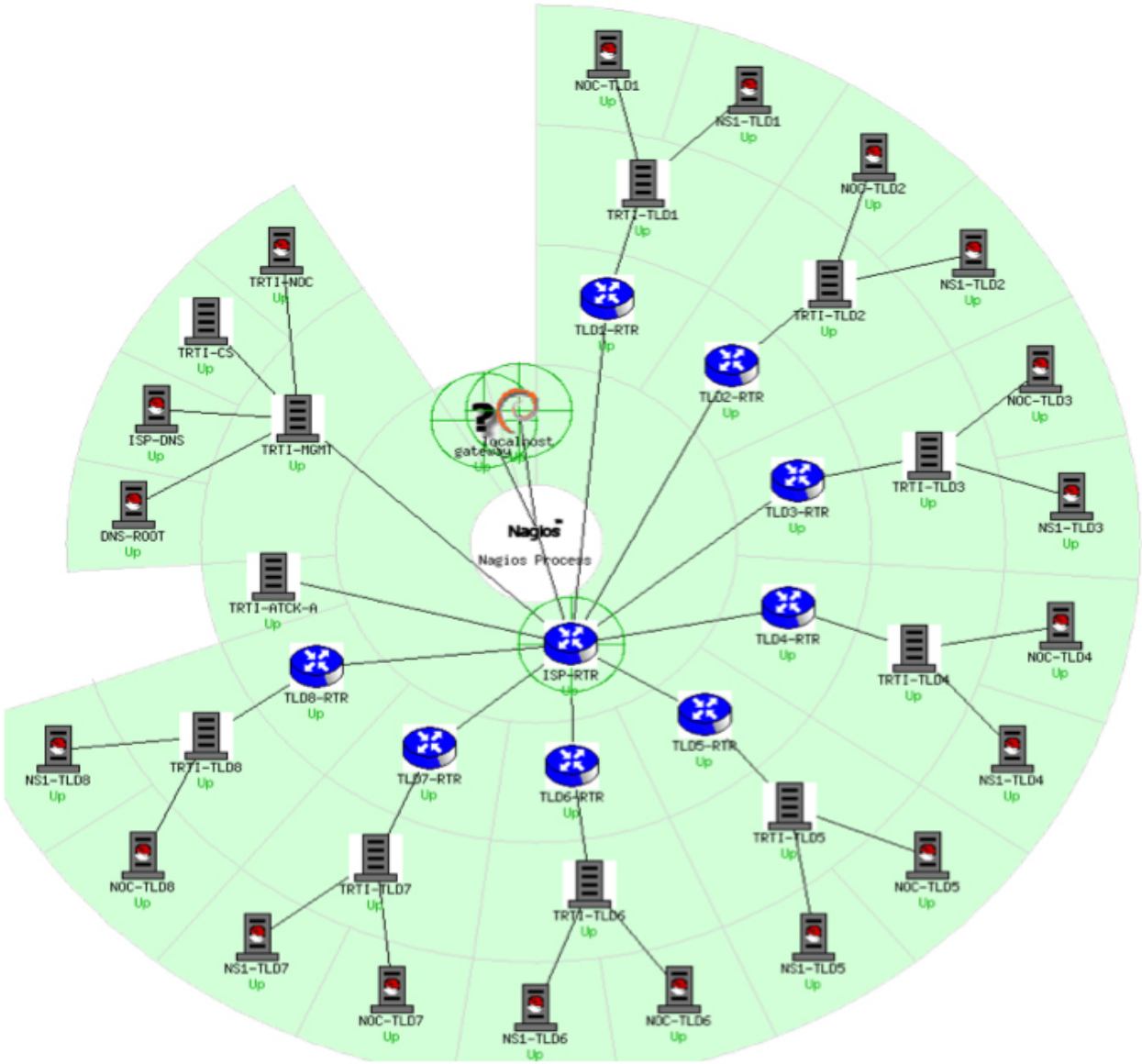
Service Groups Overview



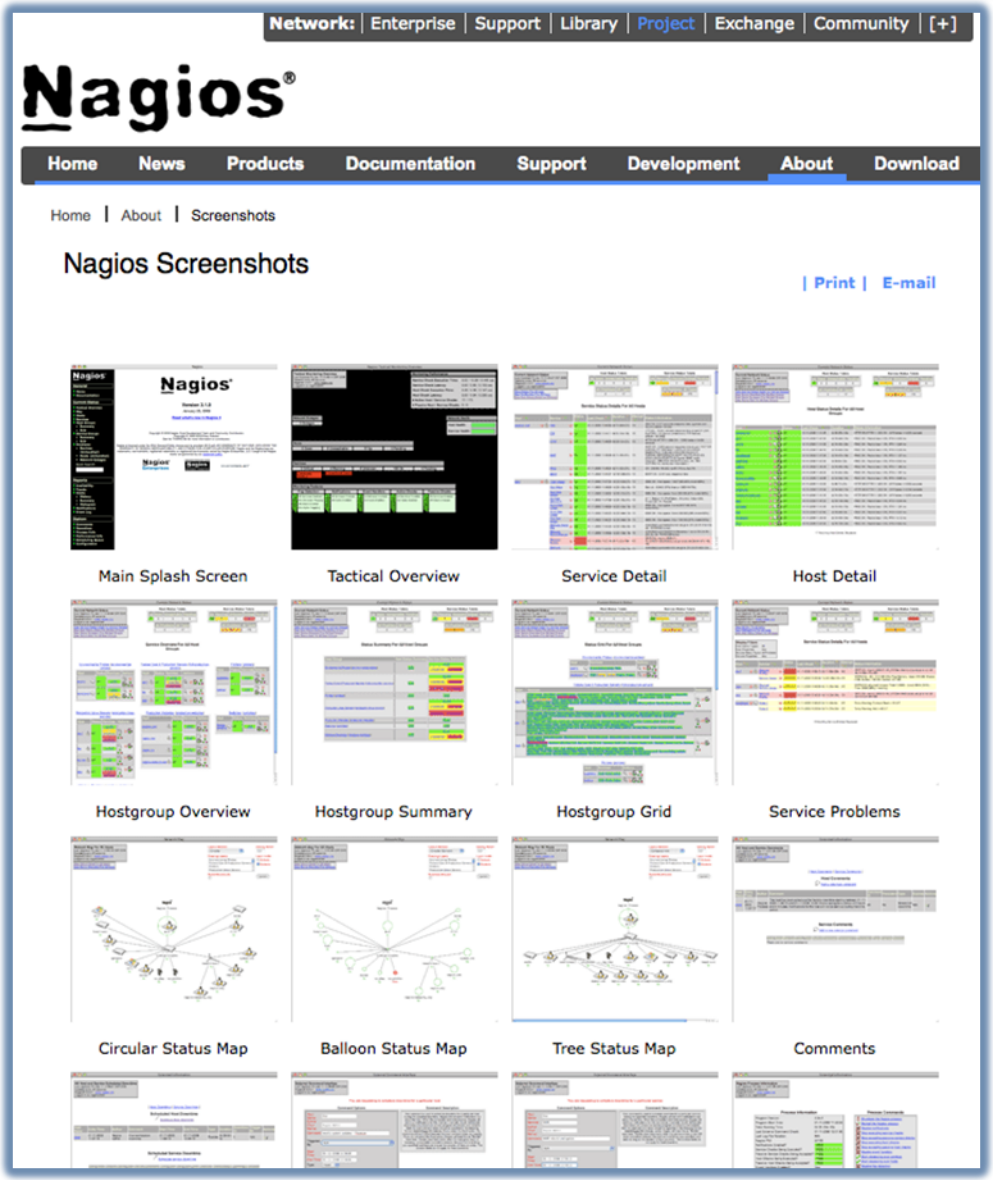
Collapsed tree status map



Marked-up circular status map



More sample screenshots



Many more sample Nagios screenshots available here:

http://www.nagios.org/about/screenshots

Features

- Verification of availability is delegated to plugins:
  - The product's architecture is simple enough that writing new plugins is fairly easy in the language of your choice.
  - There are many, many plugins available

    (Meetthecommunity, Nagios Exchange).
- Nagios uses parallel checking and forking.
  - Version 3 of Nagios does this better.

Features cont.

Nagios Exchange

- **Addons** (344)
- **Certified Compatible** (2)
- **Cool Stuff** (6)
- **Demos** (2)
- **Distributions** (11)
- **Documentation** (41)
- **Graphics and Logos** (30)
- **Multimedia** (54)
- **Patches** (13)
- **Plugins** (1668)
- **Seedcamp** (5)
- **Translations** (4)
- **Tutorials** (11)
- **Uncategorized** (97)
- **Utilities** (3)

Features cont.

- Has intelligent checking capabilities. Attempts to distribute the server load of running Nagios (for larger sites) and the load placed on devices being checked.
- Configuration is done in simple, plain text files, but that can contain much detail and are based on templates.
- Nagios reads it's configuration from an entire directory. You decide how to define individual files.

Features cont.

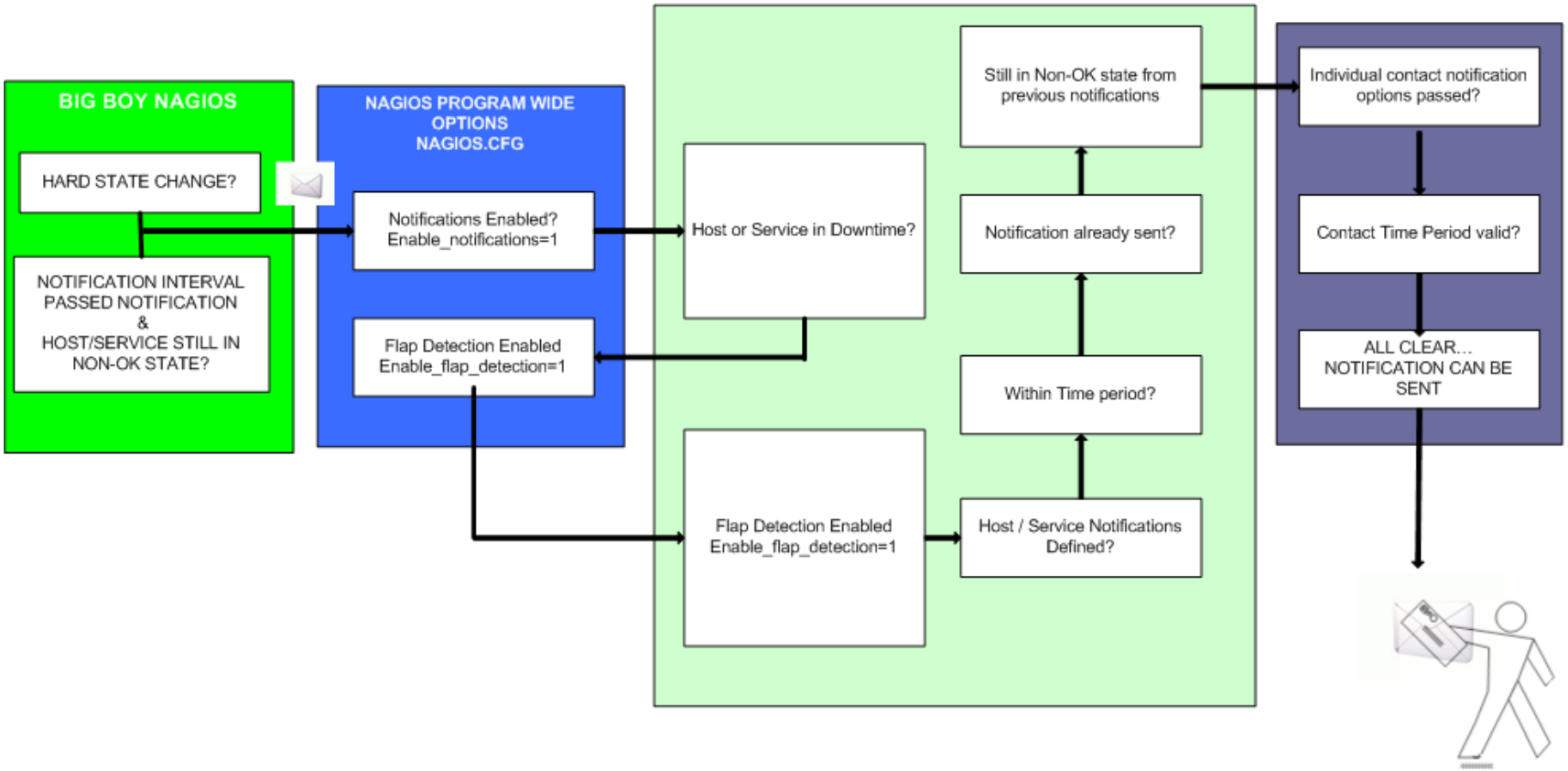
- Utilizes topology to determine dependencies.
  - Nagios differentiates between what is down vs. what is not available. This way it avoids running unnecessary checks or alarming you about things it doesn't know the status of.
- Nagios allows you to define how you send notifications based on combinations of:
  - Contacts and lists of contacts
  - Devices and groups of devices
  - Services and groups of services
  - Defined time periods, by persons or groups.
  - The state of a service.

And, even more...

Service state:

- When configuring a service you have the following notification options:
  - **d:** DOWN: The service is down (not available)
  - **u:** UNREACHABLE: When the host is not visible
  - **r:** RECOVERY: (OK) Host is coming back up
  - **f:** FLAPPING: When a host first starts or stops or it's state is undetermined.
  - **n:** NONE: Don't send any notifications

NAGIOS - NOTIFICATION FLOW DIAGRAM



NOTE: The flow will only continue when each of the listed filters are satisfied.

CONTACT GETS THE NOTIFICATION MESSAGE

Features, features, 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.
- ✓ *Can detect flapping and suppress additional notifications.*
- Allows for multiple notification methods:
  - e-mail, pager, SMS, winpopup, audio, etc...

How checks work

- A node/host/device consists of one or more service checks (PING, HTTP, MYSQL, SSH, etc)
- Periodically Nagios checks each service for each node and determines if state has changed. State changes are:
  - CRITICAL
  - WARNING
  - UNKNOWN
- For each state change you can assign:
  - Notification options (as mentioned before)
  - Event handlers

How checks work continued

Parameters

- Normal checking interval
- Re-check interval
- Maximum number of checks.
- Period for each check

- Node checks only happen when on services respond (assuming you've configured this).
  - A node can be:
    - DOWN
    - UNREACHABLE

How checks work continued

In this manner it can take some time before a host change's its state to “down” as Nagios first does a service check and then a node check.

By default Nagios does a node check 3 times before it will change the nodes state to down.

You can, of course, change all this.

The concept of “parents”

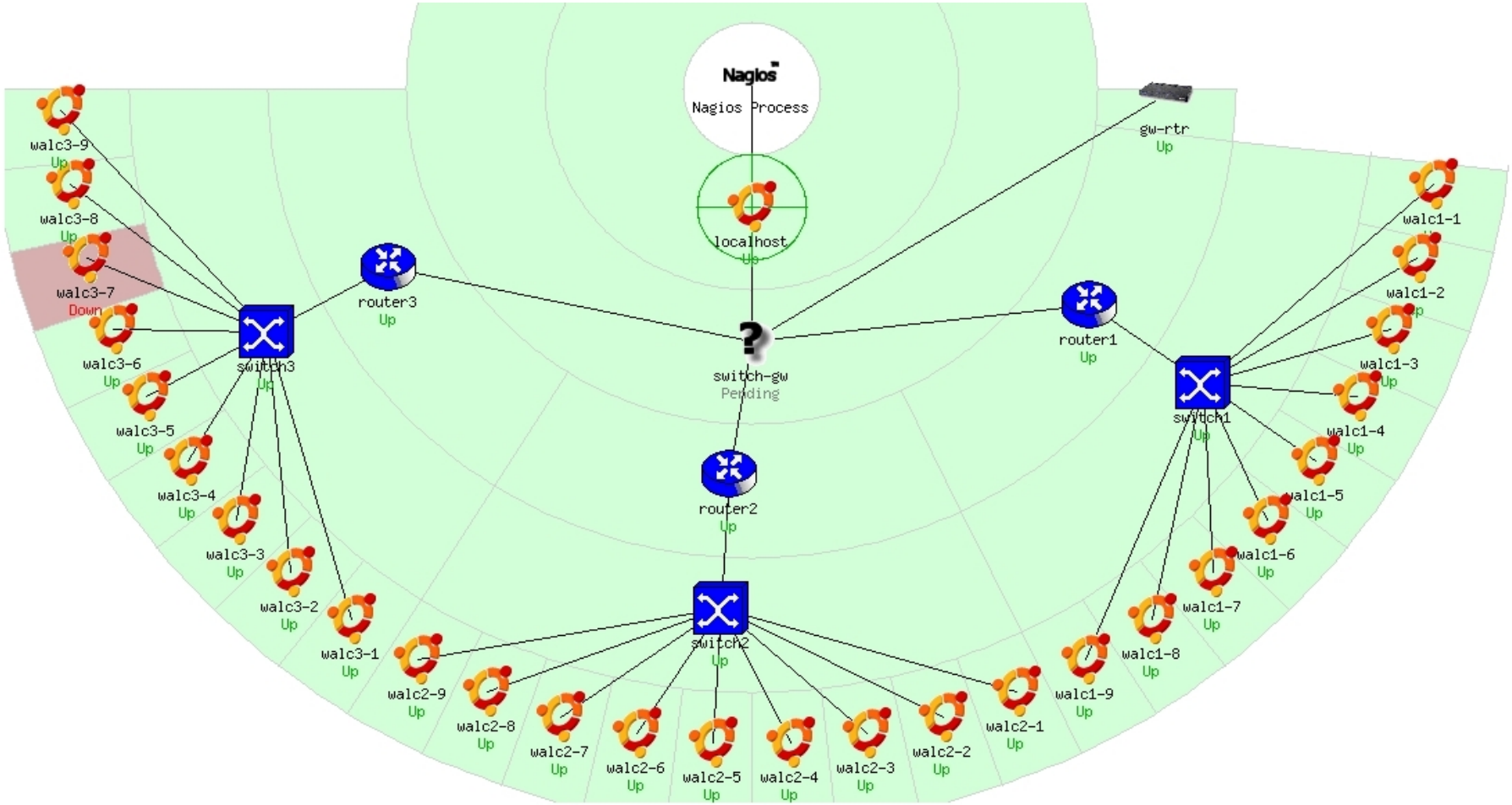
Nodes can have parents:

- For example, the parent of a PC connected to a switch would be the switch.
- This allows us to specify the network dependencies that exist between machines, switches, routers, etc.
- This avoids having Nagios send alarms when a parent does not respond.
- A node can have multiple parents.

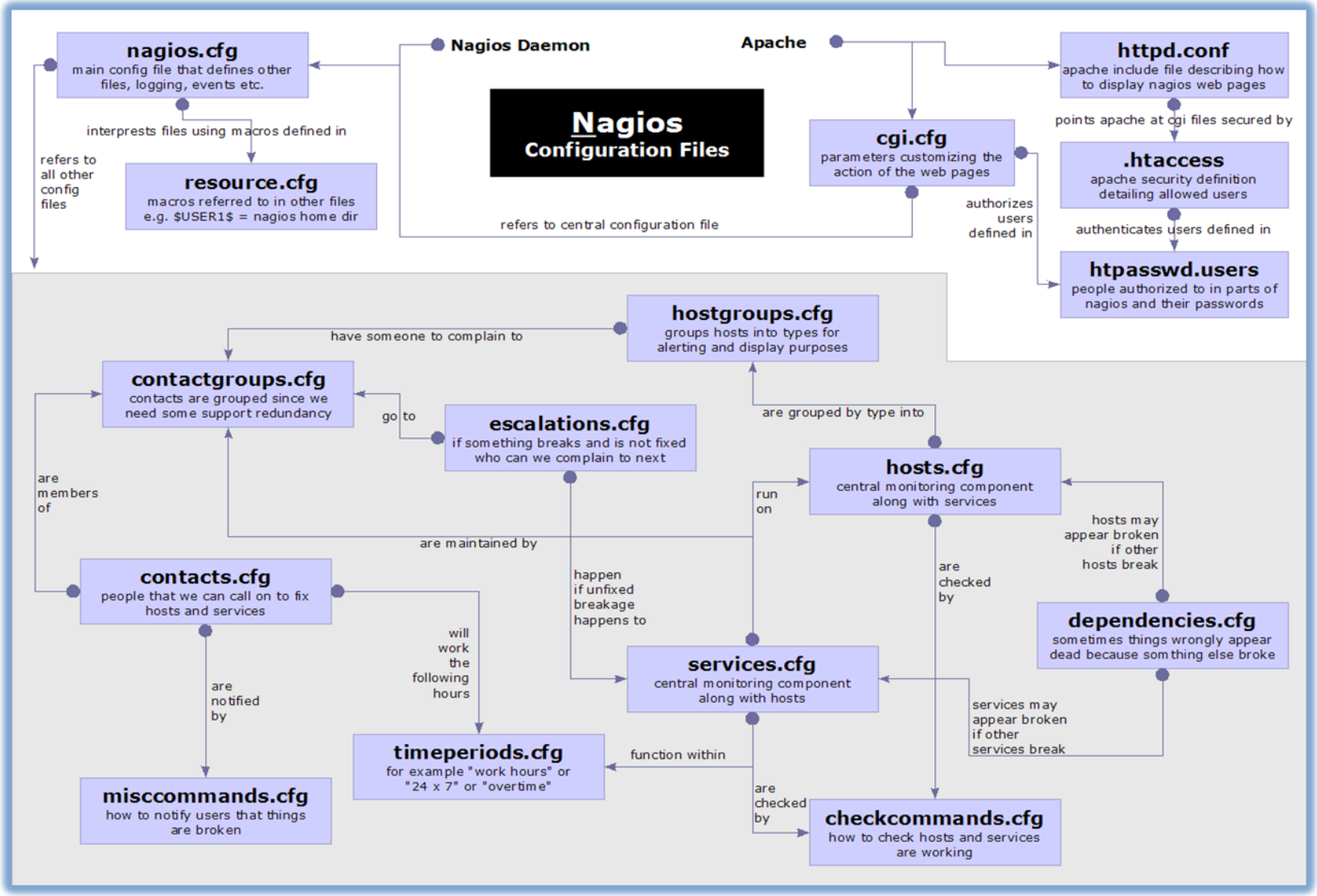
Network viewpoint concept

- Where you locate your Nagios server will determine your point of view of the network.
- Nagios allows for parallel Nagios boxes that run at other locations on a network.
- Often it makes sense to place your Nagios server nearer the border of your network vs. in the core.

Network viewpoint



Nagios configuration files



Configuration Files

Located in /usr/local/etc/nagios/

Important files include:

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

Configuration files continued

Under objects/*

- contacts.cfg — users and groups
- commands.cfg — commands used by checks
- templates.cfg — reusable definitions
- timeperiods.cfg — timeperiods (24x7, weekend, …)

Your own configurations go under objects/ normally.

- On Ubuntu, all this is in /etc/nagios3/, and /etc/nagios3/conf.d/

Configuration files continued

Under objects/ some other possible config files:

- host-gateway.cfg — Default route definition
- extinfo.cfg — Additional node information
- servicegroups.cfig — Groups of nodes and services
- localhost.cfg — Define the Nagios server itself
- pcs.cfg — Sample definition of PCs (hosts)
- switches.cfg — Definitions of switches (hosts)
- routers.cfg — Definitions of routers (hosts)

Pre-installed plugins

check_bgpstate	check_hpjd	check_mailq	check_overcr
check_ssmtp	check_breeze	check_http	check_mrtg
check_pgsql	check_swap	check_by_ssh	check_icmp
check_mrtgtraf	check_ping	check_tcp	check_clamd
check_ide_smart	check_mysql	check_pop	check_time
check_cluster	check_ifoperstatus	check_mysql_query	
check_procs	check_udp	check_dhcp	check_ifstatus
check_nagios	check_radius	check_ups	check_dig
check_imap	check_nntp	check_real	check_users
check_disk	check_ircd	check_nntps	check_rpc

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: *Tunning Nagios for Maximum Performance* **http://nagios.sourceforge.net/docs/2_0/tuning.html**

CGI configuration

/usr/local/etc/nagios/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
- Could adjust as needed, such as work week only.
- Could adjust a new time period for “outside of 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
        }
```

Configuring service/host checks:

Definition of “host alive”

```
# 'check-host-alive' command definition
define command{
        command_name    check-host-alive
        command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 2000.0,60% -c
5000.0,100% -p 1 -t 5
 }
```

- Located in /usr/local/etc/nagios/objects/commands.cfg
- While these are “service” or “host” checks Nagios refers to them as “commands”

Notification commands

Allows you to utilize any command you wish.
We'll 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: $SERVICEDESC$\nHost:
$HOSTNAME$\nIn: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\nInfo: $SERVICEOUTPUT$\nDate: $SHORTDATETIME$" | /bin/mail
-s '$NOTIFICATIONTYPE$: $HOSTNAME$/$SERVICEDESC$ is $SERVICESTATE$'
$CONTACTEMAIL$
        }
```

From: nagios@nms.localdomain
To: grupo-redes@localdomain
Subject: Host DOWN alert for switch1!
Date: Thu, 29 Jun 2006 15:13:30 -0700

Host: switch1
In: Core_Switches
State: DOWN
Address: 111.222.333.444
Date/Time: 06-29-2006 15:13:30
Info: CRITICAL - Plugin timed out after 6 seconds

Nodes and services configuration

Based on templates

- This saves lots of time avoiding repetition
- Similar to Object Oriented programming

Create default templates with default parameters for a:

- generic node
- generic service
- generic contact

Generic node template

```
define host{
        name                            generic-host
        notifications_enabled           1
        event_handler_enabled           1
        flap_detection_enabled          1
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
        check_command                   check-host-alive
        max_check_attempts              5
        notification_interval           60
        notification_period             24x7
        notification_options            d,r
        contact_groups                  nobody
        register                        0
        }
```

Individual node configuration

```
define host{
        use                     generic-host
        host_name               switch1
        alias                   Core_switches
        address                 192.168.1.2
        parents                 router1
        contact_groups          switch_group
}
```

Generic service configuration

```
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
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
        is_volatile                     0
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            1
        notification_interval           60
        notification_period             24x7
        notification_options            c,r
        register                        0
        }
```

Individual service configuration

```
define service{
        host_name                       switch1
        use                             generic-service
        service_description             PING
        check_command                   check-host-alive
        max_check_attempts              5
        normal_check_interval           5
        notification_options            c,r,f
        contact_groups                  switch-group
}
```

Group service configuration

```
# 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 ; set > 0 if you want to be renotified
}
```

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

```
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
}
```

Beeper and sms messages

- It's important to integrate Nagios with something available outside of work
  - Problems occur after hours... (unfair, but true)
- A critical item to remember: an SMS or message system should be independent from your network.
  - You can utilize a modem and a telephone line
  - Packages like sendpage, qpage or gnokii can help.

References

- **Nagios web site**
  http://www.nagios.org/
- **Nagios plugins site**
  http://sourceforge.net/projects/nagiosplug/
- *Nagios System and Network Monitoring*, by Wolfgang Barth. Good book about Nagios.
- **Unofficial Nagios plugin site**
  http://www.nagiosexchange.org/
- **A Debian tutorial on Nagios**
  http://www.debianhelp.co.uk/nagios.htm
- **Commercial Nagios support**
  http://www.nagios.com/