Nagios®

Introduction

- A key measurement tool for actively monitoring availability of devices and services.
- Possible the most used open source network monitoring software.
- Has a web interface.
  - Uses CGIs written in C for faster response and scalability.
- Can support up to thousands of devices and services.

Installation

In Debian/Ubuntu:

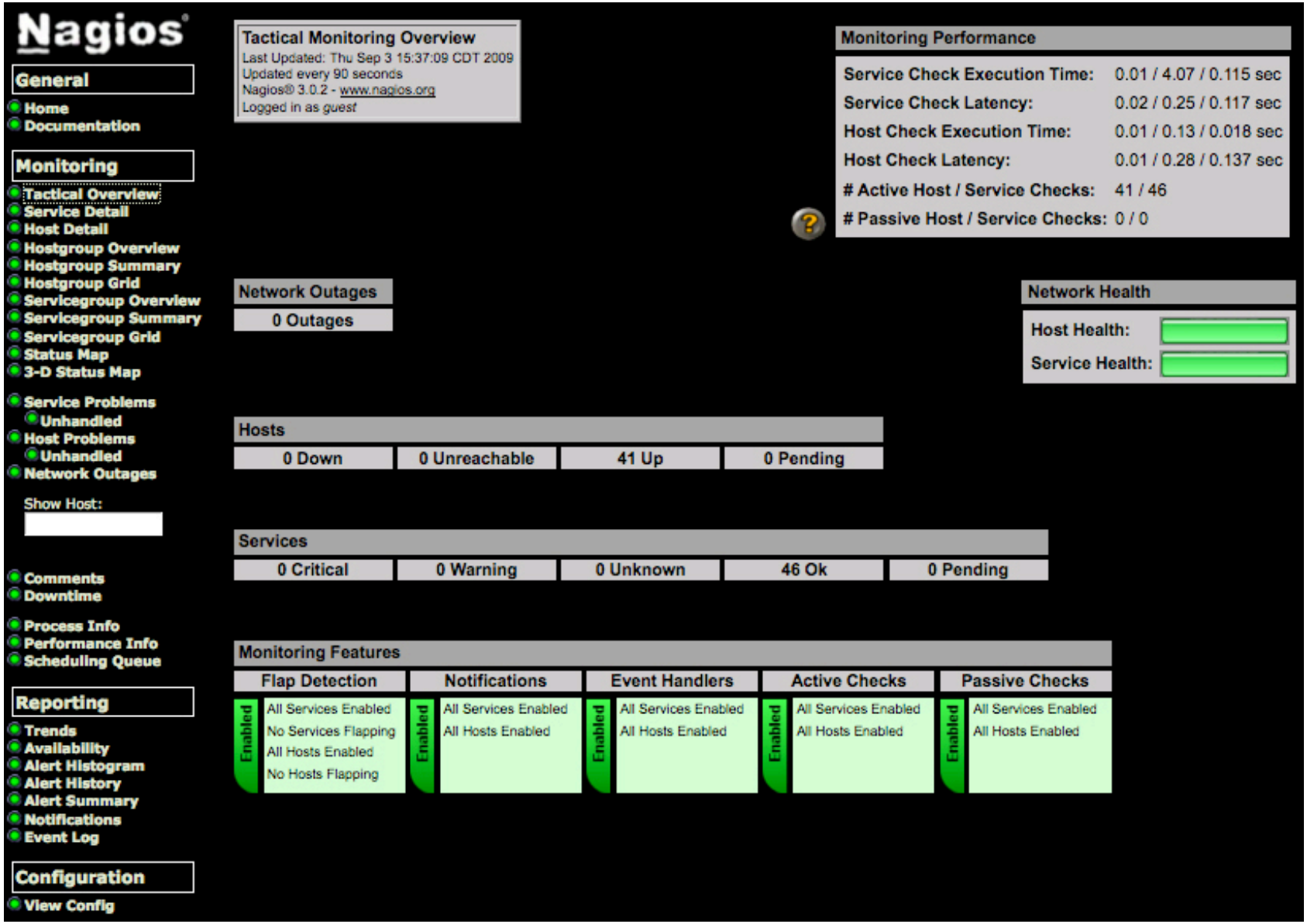
- `# apt-get install nagios3`
- Files are installed here:

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

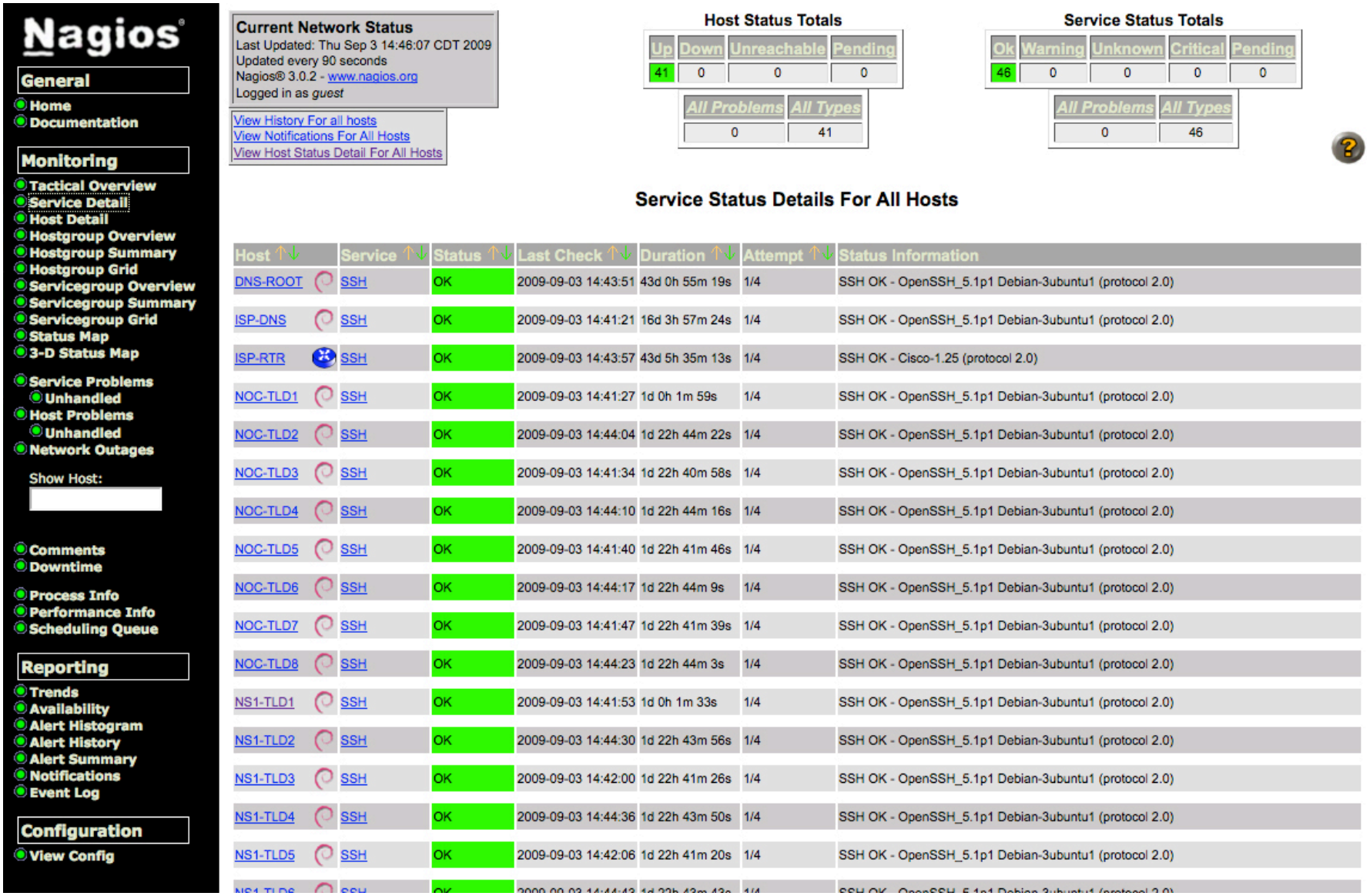
- Access Nagios via the web here:

http://192.168.10X.30/nagios3/

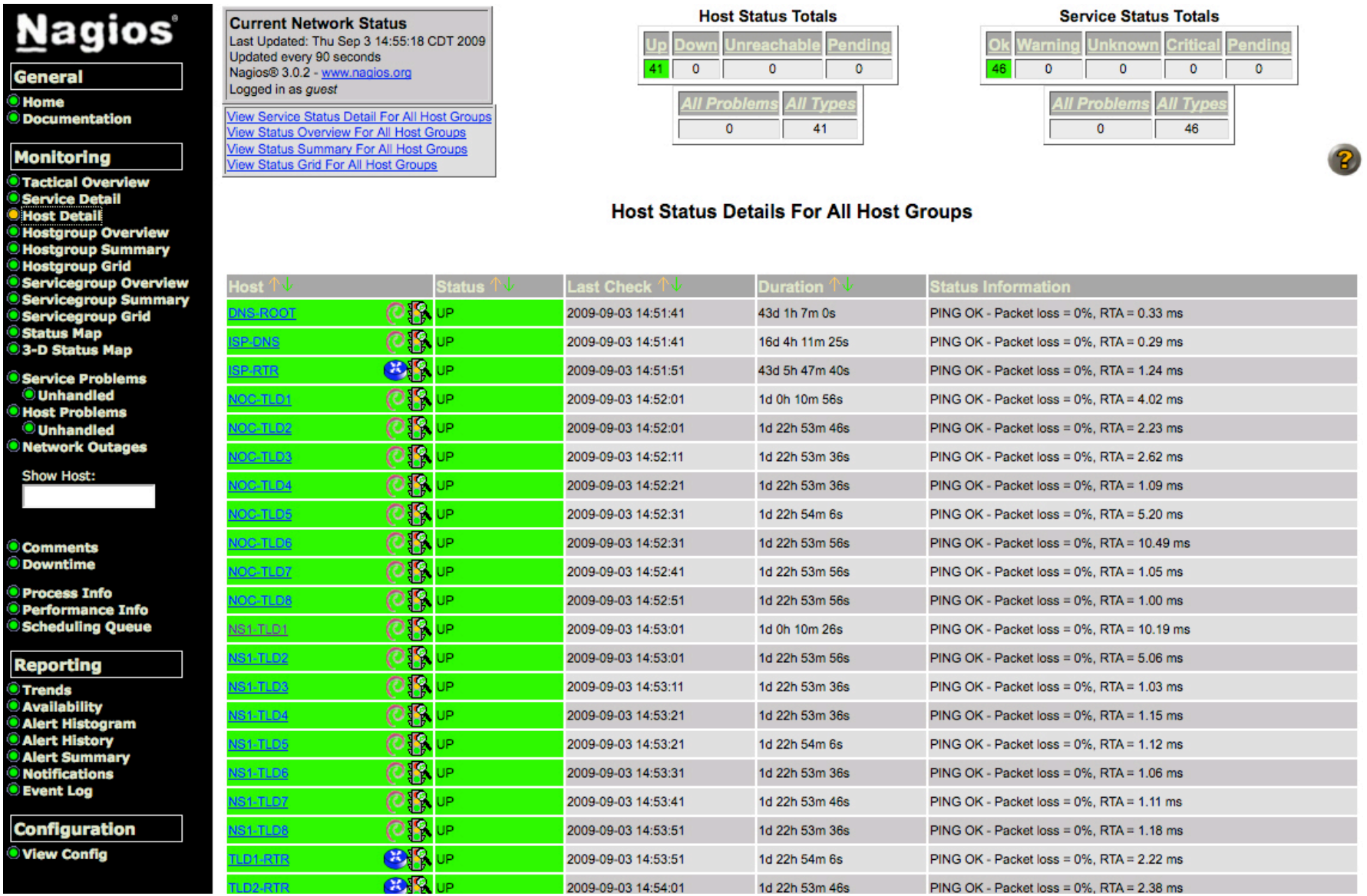
Nagios: Tactical Overview



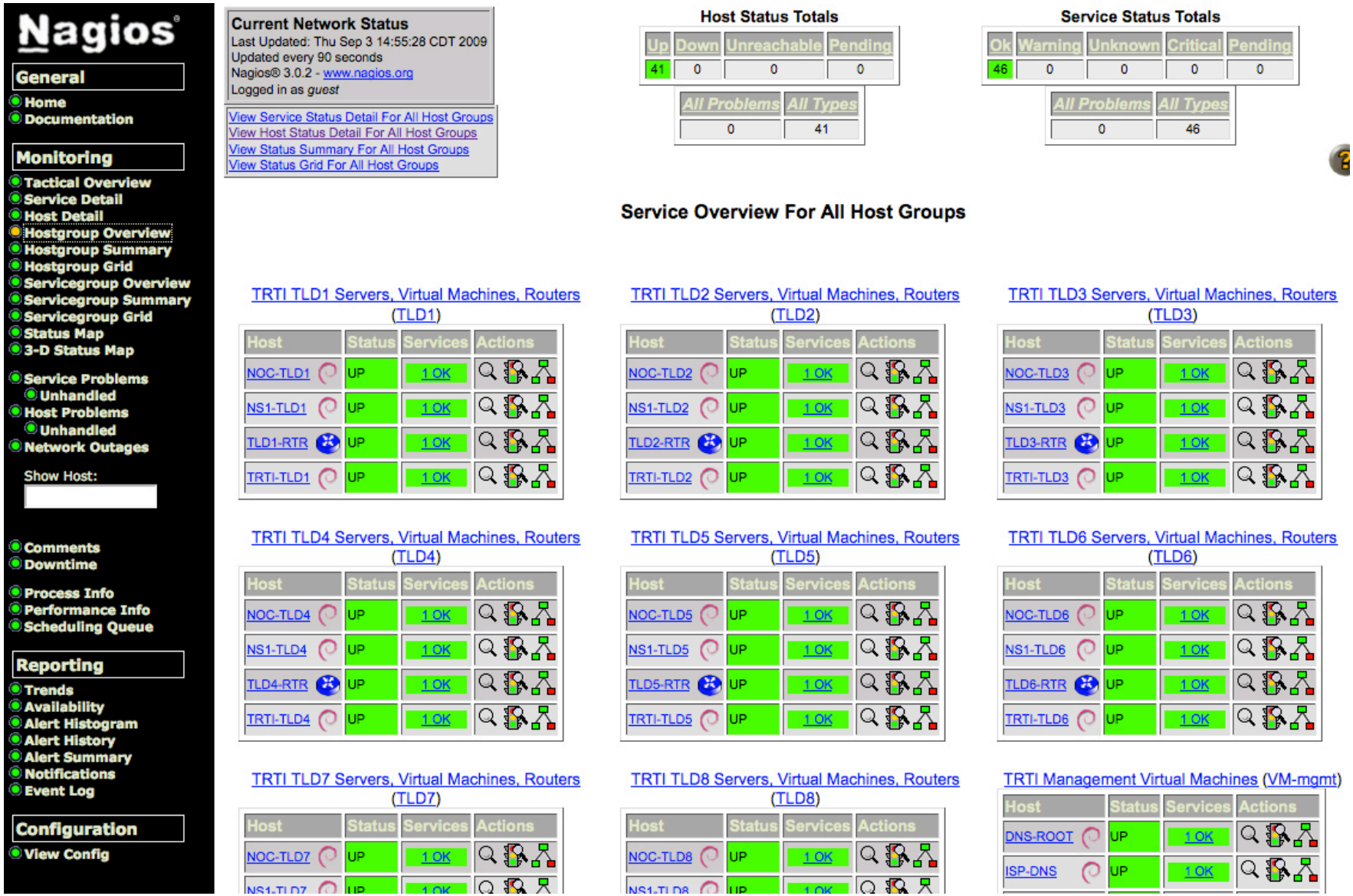
Nagios: Service Detail



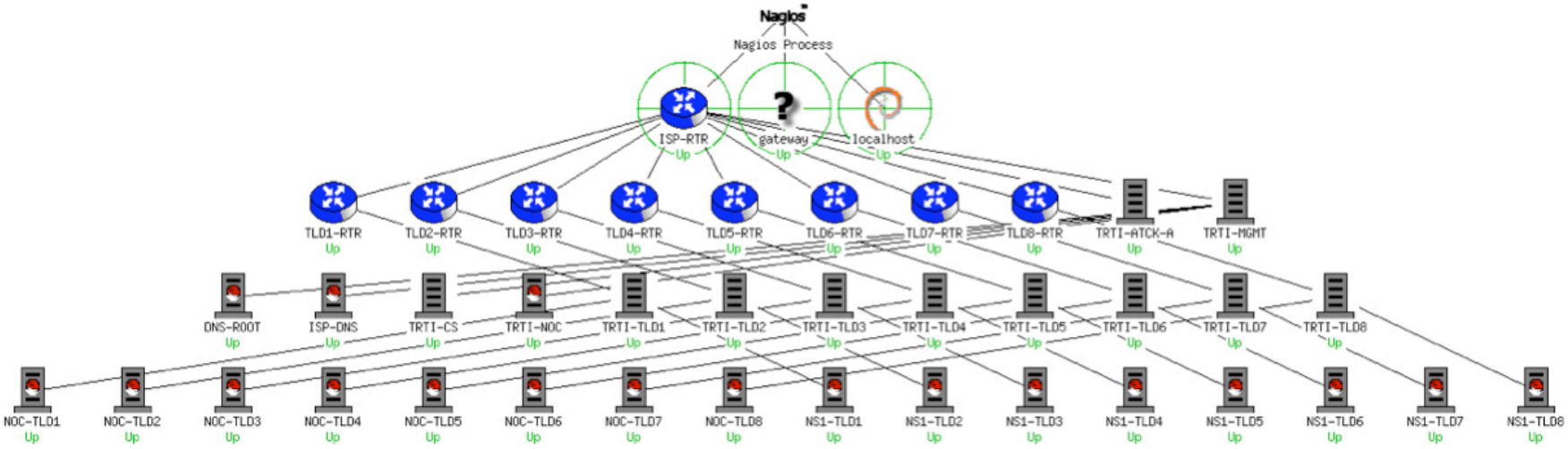
Nagios: Host Detail



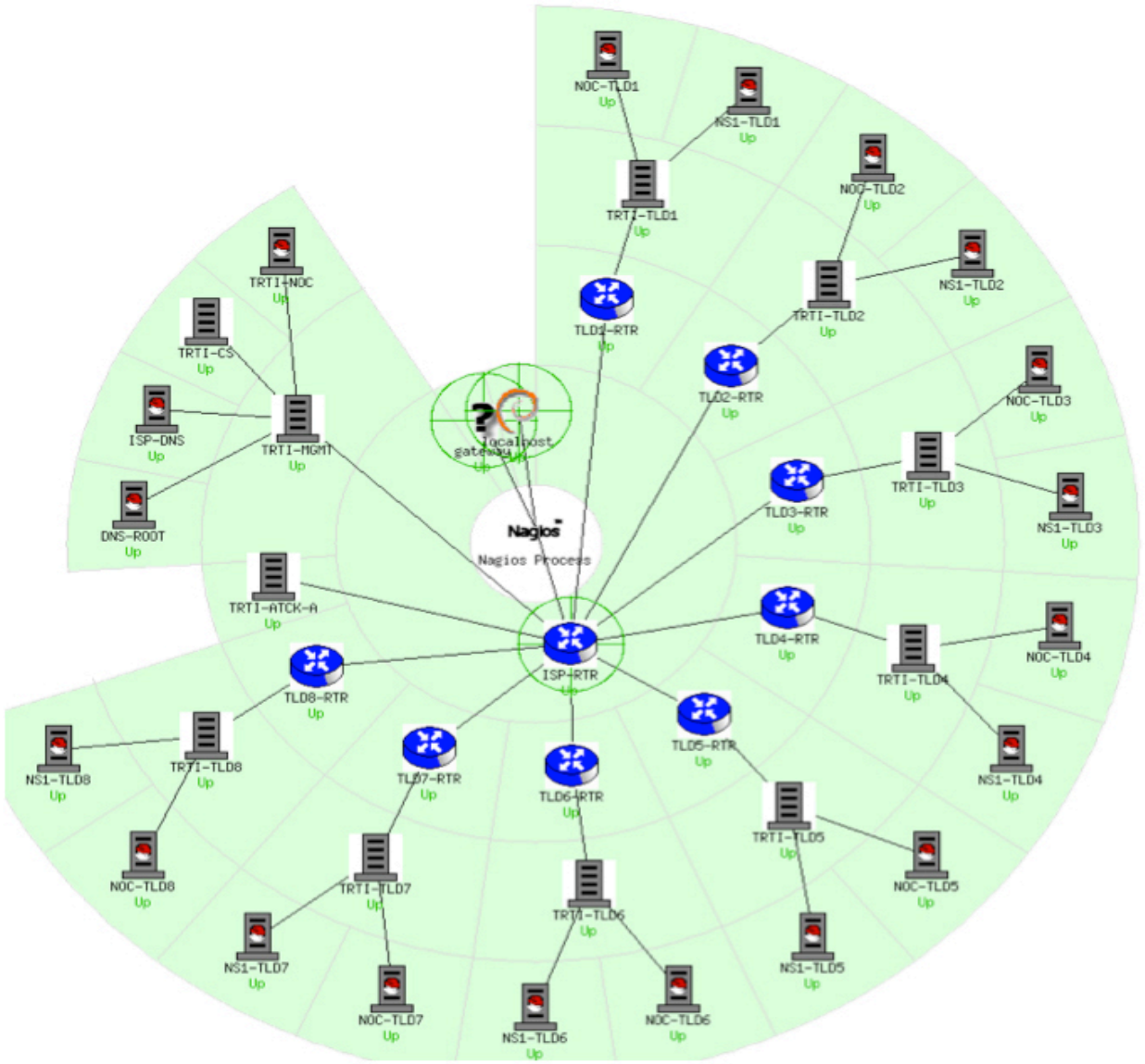
Nagios: Hostgroup Summary



Nagios: Status Map: Collapsed Tree



Nagios: Status Map: Circular (Marked Up)



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.
- Nagios uses parallel checking and forking.
  - Version 3 of Nagions does this better.

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.

Yet More Features...

- Utilizes topology to determine dependencies.
  - Nagios differentiates between what is down vs. what is not available. This way it avoids running unnecessary checks.
- 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 hours by persons or groups.
  - The state of a service.

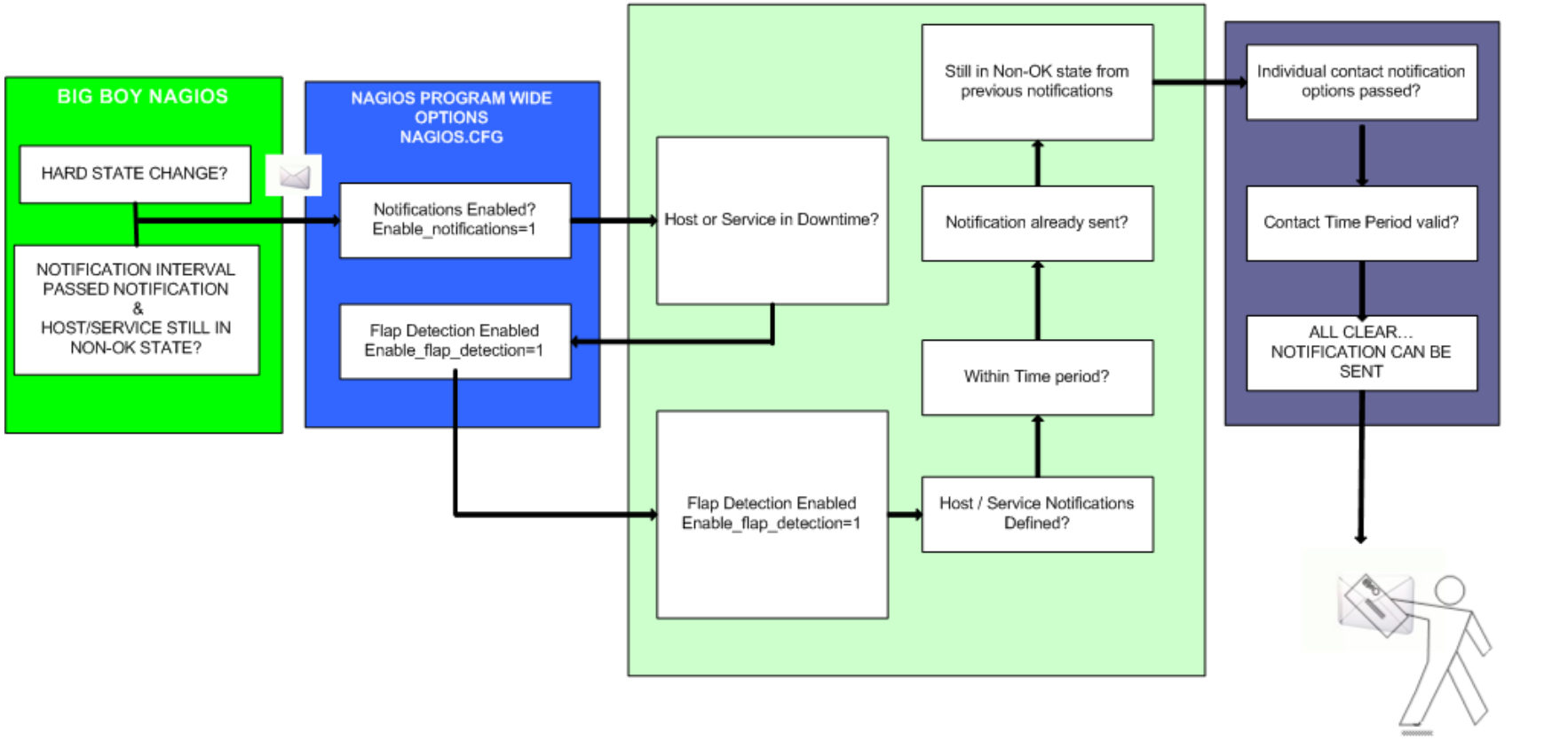
And, even more...

Service state:

When configuration 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 such as:
  - e-mail, pager, SMS, winpopup, audio, etc...
- Allows you to define notification levels. Critical feature.

How Checks Work

- A node, host or 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

- 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

- 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.

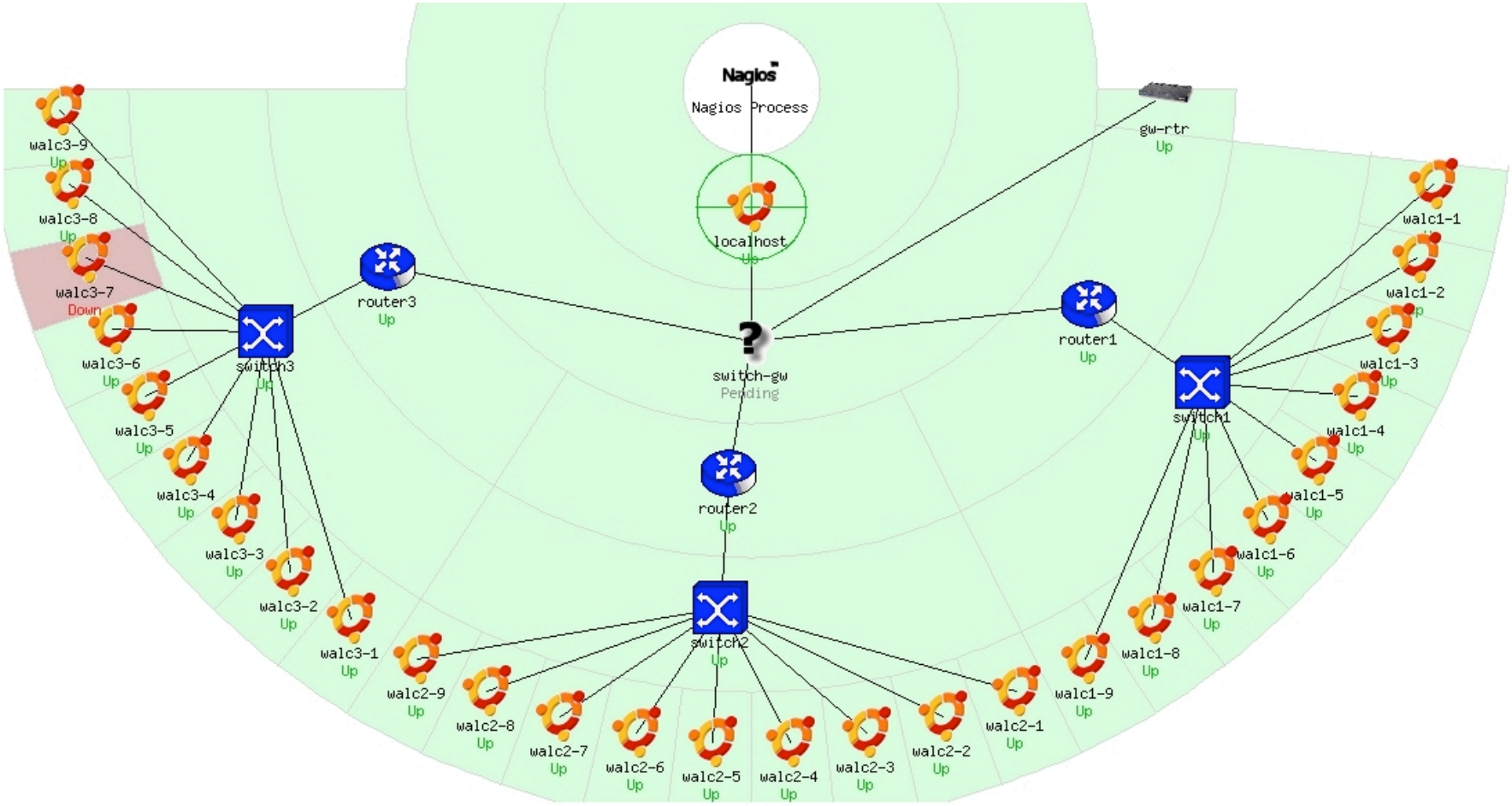
The Concept of “Parents”

- 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.

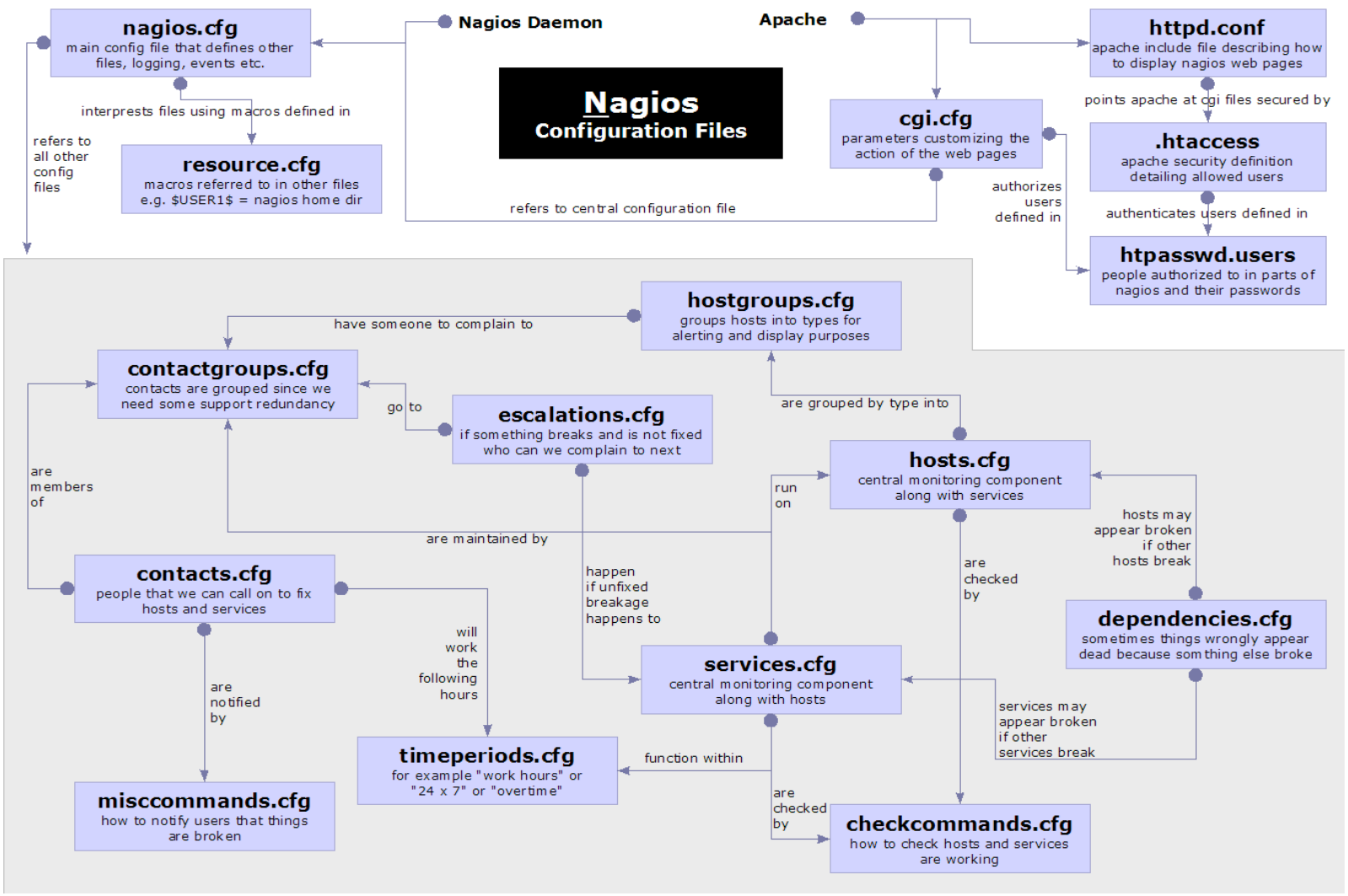
The Idea of Network Viewpoint

- 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 /etc/nagios3/
- 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.
  - conf.d/* — All other configuration goes here!

Configuration Files

Under conf.d/* (*sample only*)

▪ contacts_nagios3.cfg	users and groups
▪ generic-host_nagios2.cfg	default host template
▪ generic-service_nagios2.cfg	default service template
▪ hostgroups_nagios2.cfg	groups of nodes
▪ services_nagios2.cfg	what services to check
▪ timeperiods_nagios2.cfg	when to check and who to notifiy

Configuration Files

Under conf.d 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)

Plugin Configuration

- The Nagios package in Ubuntu comes with a bunch of pre-installed plugins:
- apt.cfg breeze.cfg dhcp.cfg disk-smb.cfg disk.cfg dns.cfg dummy.cfg flexlm.cfg fping.cfg ftp.cfg games.cfg hppjd.cfg http.cfg ifstatus.cfg ldap.cfg load.cfg mail.cfg mrtg.cfg mysql.cfg netware.cfg news.cfg nt.cfg ntp.cfg pgsql.cfg ping.cfg procs.cfg radius.cfg real.cfg rpc-nfs.cfg snmp.cfg ssh.cfg tcp_udp.cfg telnet.cfg users.cfg vsz.cfg

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

- File: /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
  - 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

- Define how you are going to test a service.

```
# '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 /etc/nagios-plugins/config, then adjust in /etc/nagios3/conf.d/services_nagios2.cfg

Notification Commands

Allows you to utilize any command you wish. You can this for generating 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 Configuration

```
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
}
```

Beeper/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.

Some References

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