PacNOG 5

Papeete, French Polynesia
17 June 2009

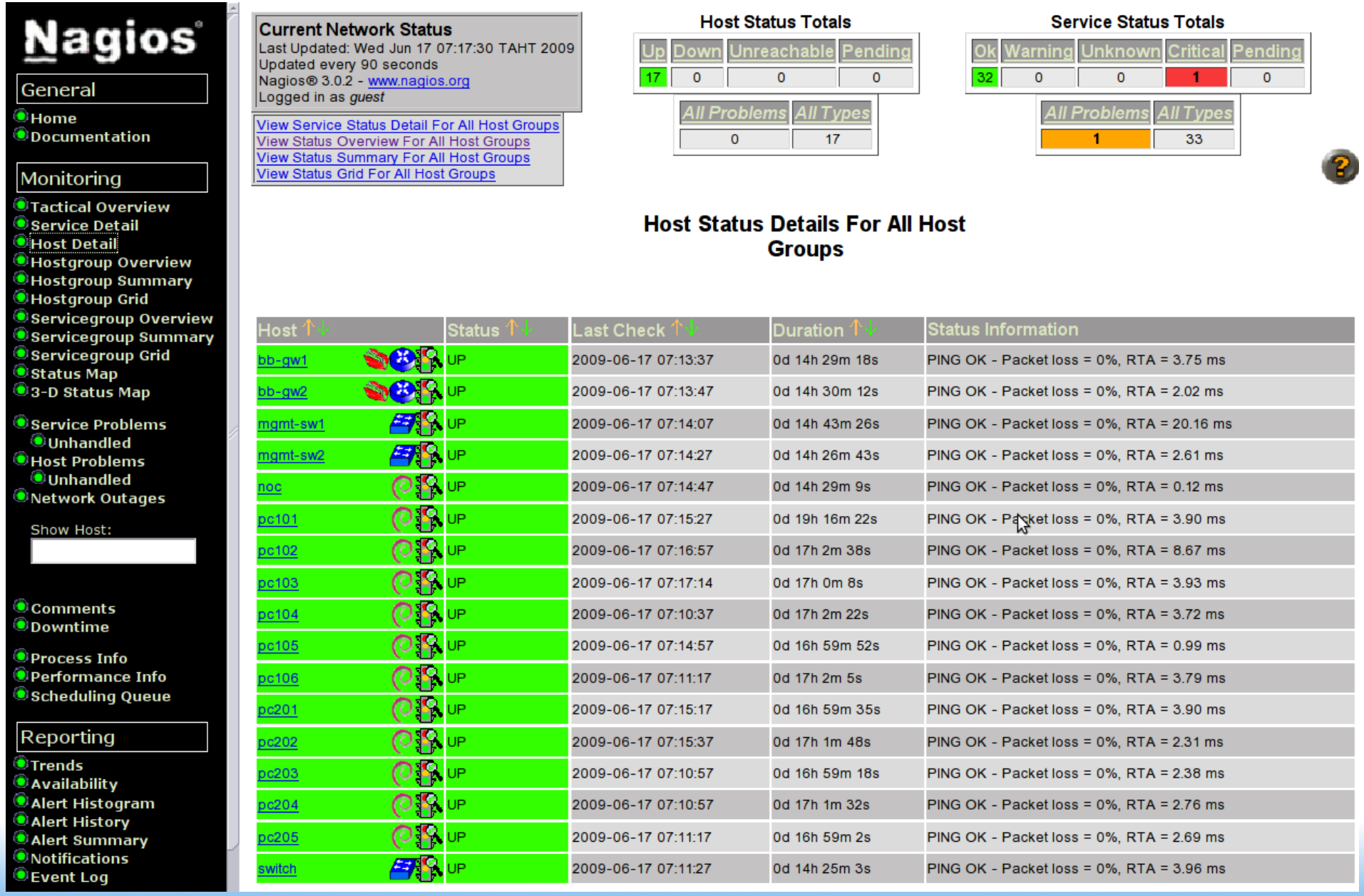
Hervey Allen

Introduction

Nagios: a measurement tool that actively monitors availability of devices and services:

- **Popular:** One of the most used open source network monitoring software packages.
- **Fast:** Uses CGI functionality written in C for faster response and scalability.
- **Scalable:** Can support up to thousands of devices and services.
- **Modular**
- **Cool-Looking Web Interface®**

"Cool-Looking Web Interface®"



Features: 1

Modular

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

Features: Plug-Ins or Modular

The Nagios package in Ubuntu comes with a number 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

There are many more available (e.g.)...

http://sourceforge.net/projects/nagiosplugins

Features: 2

Fast and Scalable

- Compiled, binary CGIs and common plug-ins for faster performance.
- Parallel checking and forking of checks to support large numbers of devices.
  - This has been considerably improved in version 3 of Nagios.
- Improvement of efficiency is a controversial topic in the Nagios community. There is now a fork, *icinga*, trying to re-write Nagios in a different manner.

Features: 3

- Uses “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, 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: 4

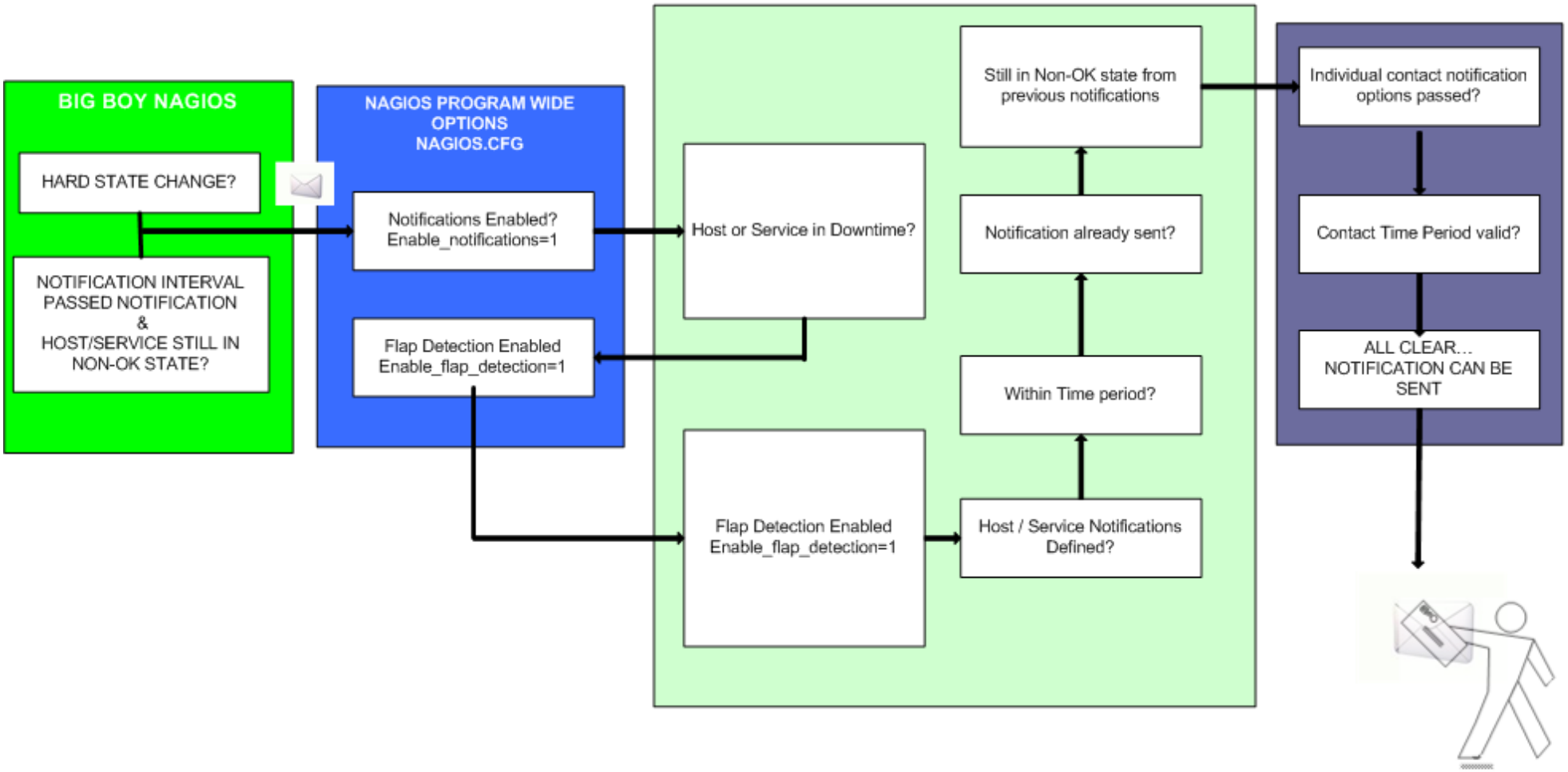
- **Topology Aware:** To determine dependencies.
  - *Differentiates between what is down vs. what is not available. This way it avoids running unnecessary checks. This is done using parent-child relationships between devices.*
- **Notifications:** How they are sent is based on combinations of:
  - *Contacts 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.*

Features: 5

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

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 (scripts, actions to take)

How Checks Work

- **Parameters:** Set in /etc/nagios3/nagios.cfg:
  - Normal checking interval
  - Re-check interval
  - Maximum number of checks.
  - Period for each check
- Services check(s) only happen when a node responds (ping check or “is alive = yes”):
  - Remember a node can be:
    - DOWN
    - UNREACHABLE

(What's the difference?)

How Checks Work: 2

In this manner it can take some time before a host changes 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.

- /etc/nagios3/nagios.cfg
- Lots of configuration settings and combinations
- Default settings have been tested for large install

The Concept of “Parents”

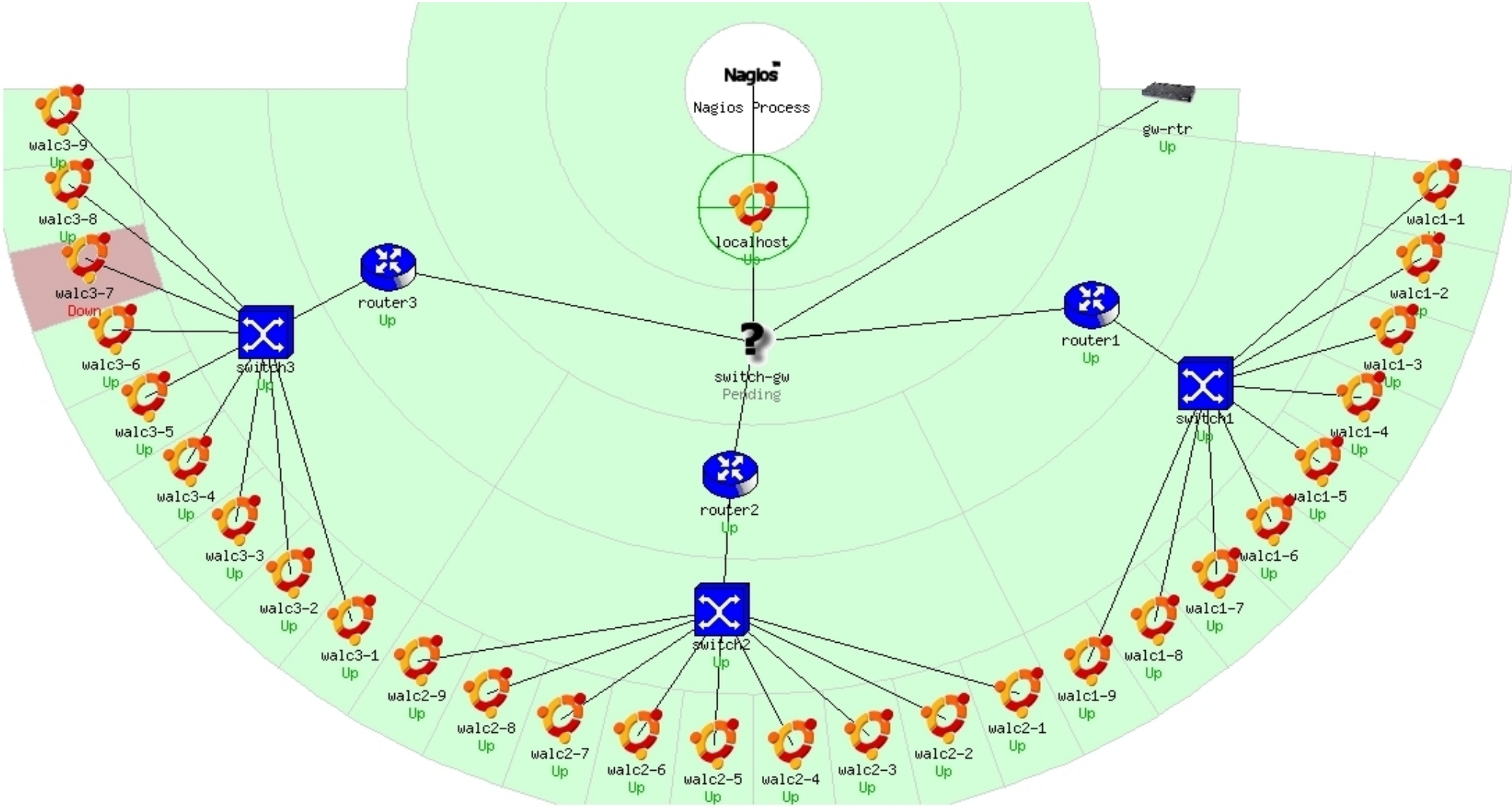
- Nodes can have parents.
  - For example, the parent of a PC connected to the switch *mgmt-sw1* would be *mgmt-sw1*.
  - This allows us to specify the network dependencies that exist between machines, switches, routers, etc.
  - This avoids having Nagios send alarms when a parent does not respond.
  - **Note:** A node can have multiple parents.

The Idea of Network Viewpoint

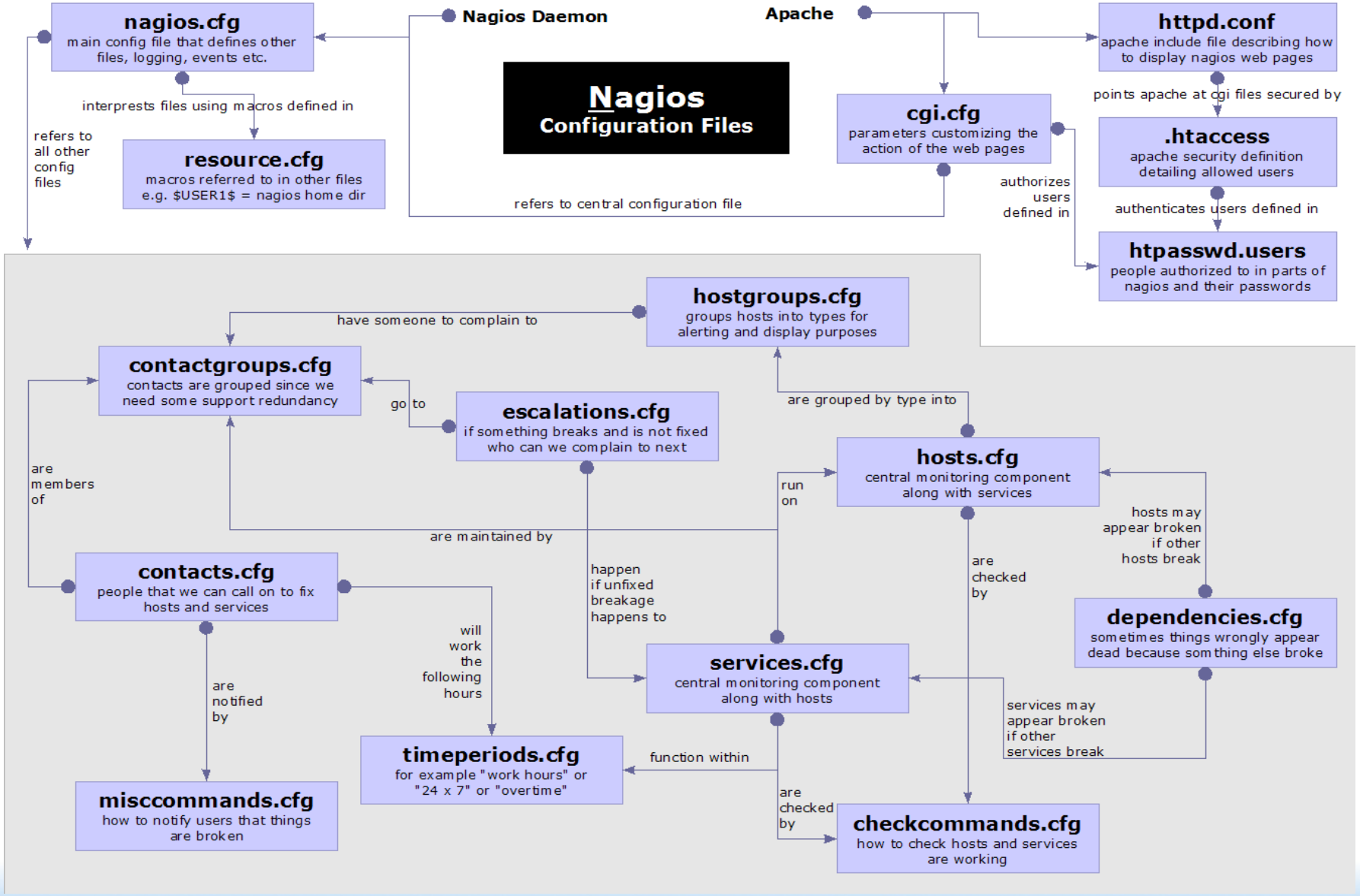
- Where you locate your Nagios server will determine your point of view of the network.
- Nagios allows for parallel Nagios boxes that run at other locations on a network.
- Often it makes sense to place your Nagios server nearer the border of your network vs. in the core, or...

  Have someone else run checks for you from an external location as well.

Network Viewpoint



Nagios Configuration Files



Configuration Files

- Located in /etc/nagios3/ (in Ubuntu)
- Important files include:
  - cgi.cfg — Controls the web interface and security options.
  - commands.cfg — The commands that Nagios uses for notifications (i.e. sending email)
  - 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 configfiles:

- 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/servers.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/nagios2/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.sourceforce.net/docs/2_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

`conf.d/timeperiods_nagios2.cfg:` 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 do 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, gnoki can help.

Some References

- http://www.nagios.org/
- http://sourceforge.net/projects/nagiosplugins
- http://www.nagiosexchange.org/
- http://www.debianhelp.co.uk/nagios.htm
- http://www.nagios.com/: Commercial Nagios support
- *Nagios,* by O'Reilly Media, Inc.
- *Nagios. System and Network Monitoring,* by Wolfgang Barth.