AfNOG 2010 Network Monitoring and Management Tutorial

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
```

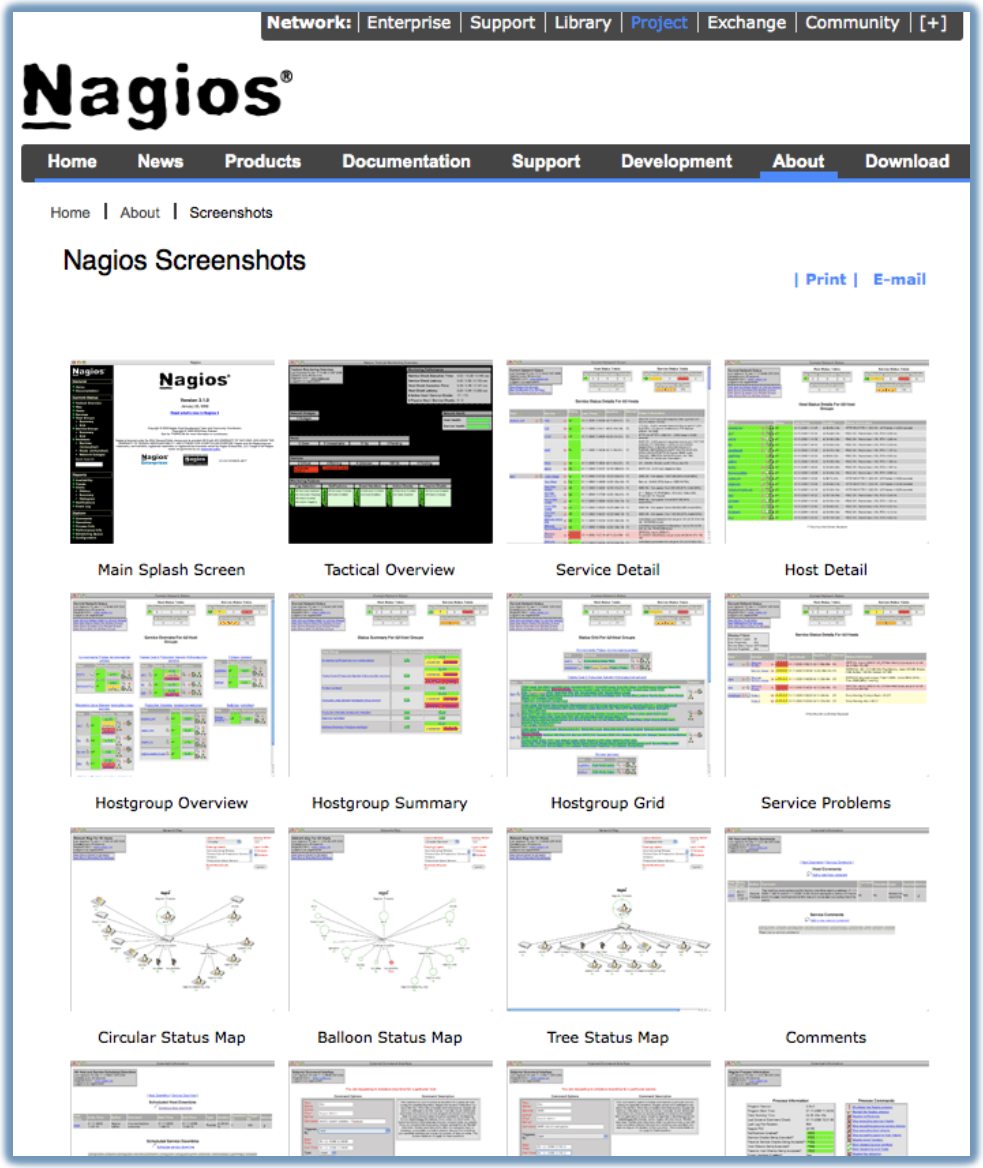
Nagios web interface is here:

http://localhost/nagios3/

Nagios Web Interface

We'll demonstrate this now…

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.
- Nagios uses parallel checking and forking.
  - Version 3 of Nagios 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.

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.
- 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.

The concept of “parents”

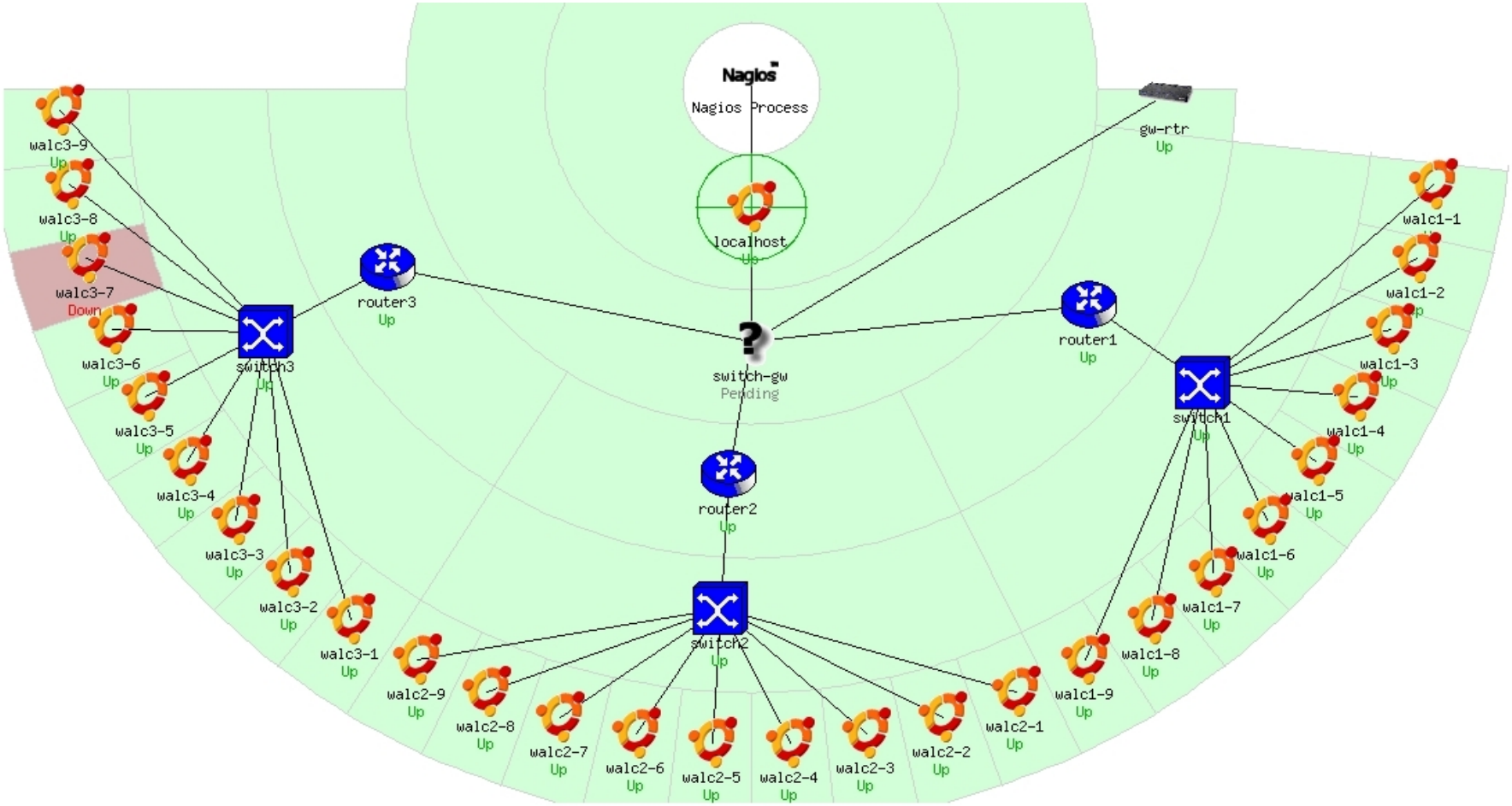
Nodes can have parents:

- 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



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 continued

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 continued

Under conf.d some other possible configfiles:

- host-gateway.cfg — Default route definition
- extinfo.cfg — Additional node information
- servicegroups.cfg — 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 in Ubuntu

check_bgpstate check_hpjd check_mailq check_overcr
check_ssmtp check_breeze check_httpcheck_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
check_wave check_disk_smb check_jabber check_nt
check_sensors check_dns check_ldap check_ntp check_spop
check_simap check_dummy check_ldaps check_ntp_peer
check_smtp check_file_age check_linux_raid check_ntp_time
check_snmp check_flexlm check_load check_nwstat

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
}
```

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/