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

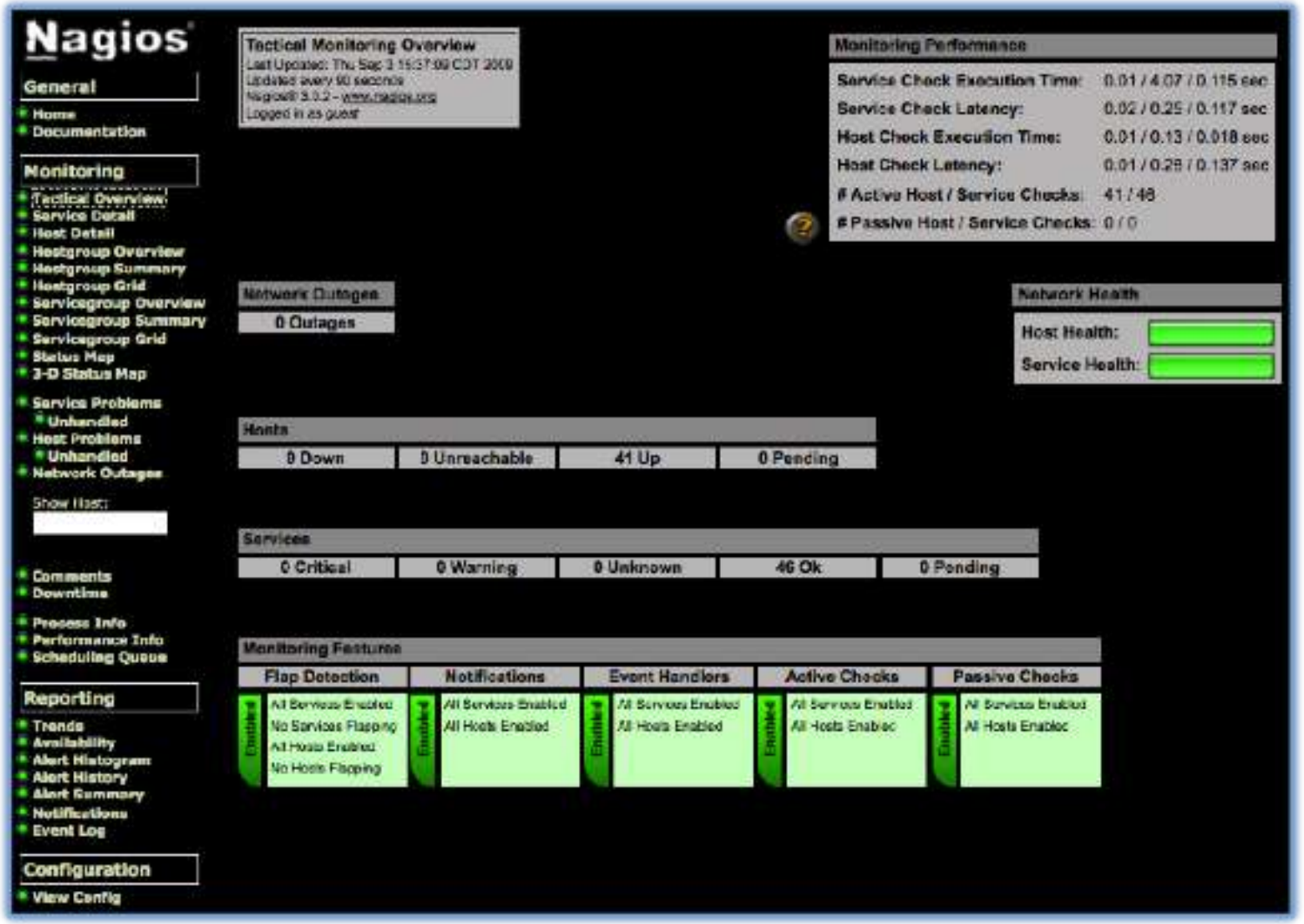
- Possibly the most used open source network monitoring software
- Web interface for viewing status, browsing history, scheduling downtime etc
- Sends out alerts via E-mail. Can be configured to use other mechanisms, e.g. SMS

Introduction

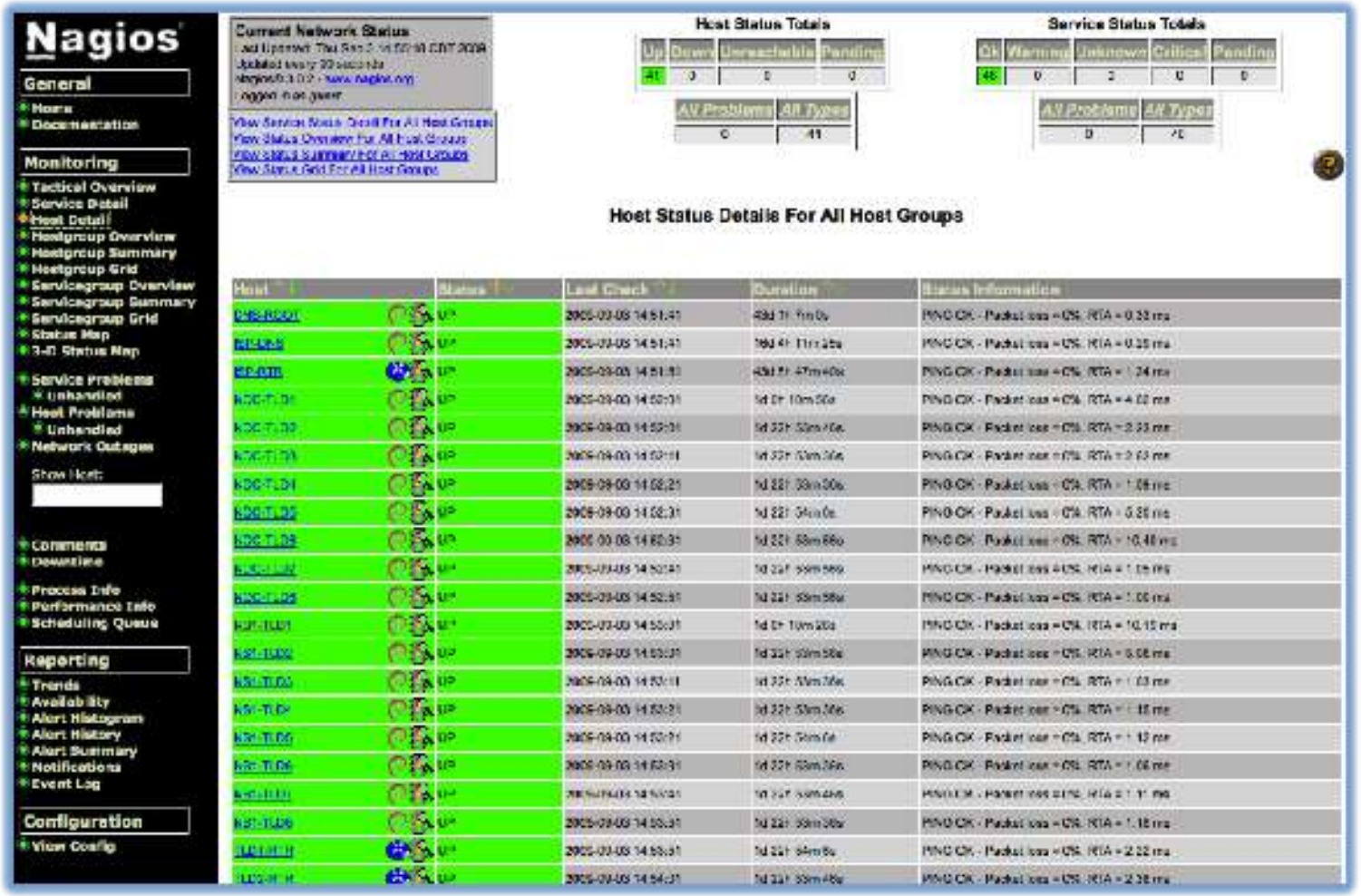
Nagios actively monitors the availability

- of Hosts (devices)
- and Services

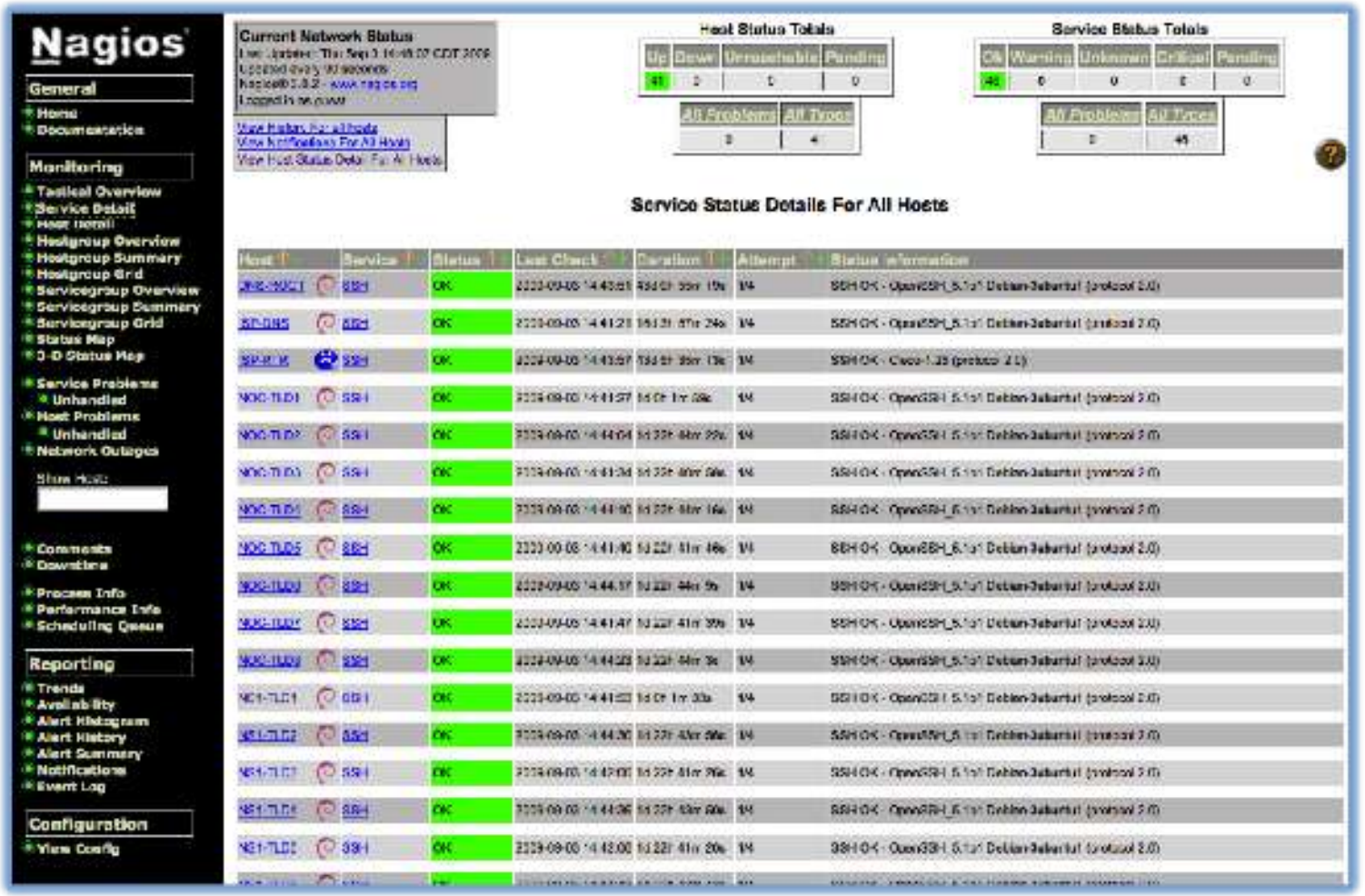
Nagios: General View



Host Detail View



Service Detail View



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

```
nsrc@s1:~$ ls /usr/lib/nagios/plugins
check_apt      check_disk      check_hpjd      check_jabber    check_mysql      check_ntp_time  check_real      check_ssh     check_wave
check_breeze   check_disk_smb  check_http      check_ldap      check_mysql_query check_nwstat    check_rpc       check_ssmtp   negate
check_by_ssh   check_dns       check_icmp      check_ldaps     check_nagios     check_oracle    check_rta_multi check_swap    urlize
check_cland    check_dummy     check_ide_smart check_load      check_nntp       check_overcr    check_sensors   check_tcp     utils.pm
check_cluster  check_file_age  check_ifoperstatus check_log    check_nntps      check_pgsql     check_simap     check_time    utils.sh
check_dbi      check_flexlm    check_ifstatus  check_mailq     check_nt         check_ping      check_smtp      check_udp
check_dhcp     check_ftp       check_imap      check_mrtg      check_ntp        check_pop       check_snmp      check_ups
check_dig      check_host      check_ired      check_mrtgtraf  check_ntp_peer   check_procs     check_spop      check_users
nsrc@s1:~$
```

/etc/nagios-plugins/config

```
nsrc@s1:~$ ls /etc/nagios-plugins/config/
apt.cfg     disk_smb.cfg  fping.cfg  http.cfg     mail.cfg   netware.cfg  pgsql.cfg   real.cfg     tcp_udp.cfg
breeze.cfg  dns.cfg       ftp.cfg    ifstatus.cfg mailq.cfg  news.cfg     ping.cfg    rpc-nfs.cfg  telnet.cfg
dhcp.cfg    dummy.cfg     games.cfg  ldap.cfg     mrtg.cfg   nt.cfg       procs.cfg   snmp.cfg     users.cfg
disk.cfg    flexlm.cfg    hppjd.cfg  load.cfg     mysql.cfg  ntp.cfg      radius.cfg  ssh.cfg
nsrc@s1:~$
```

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

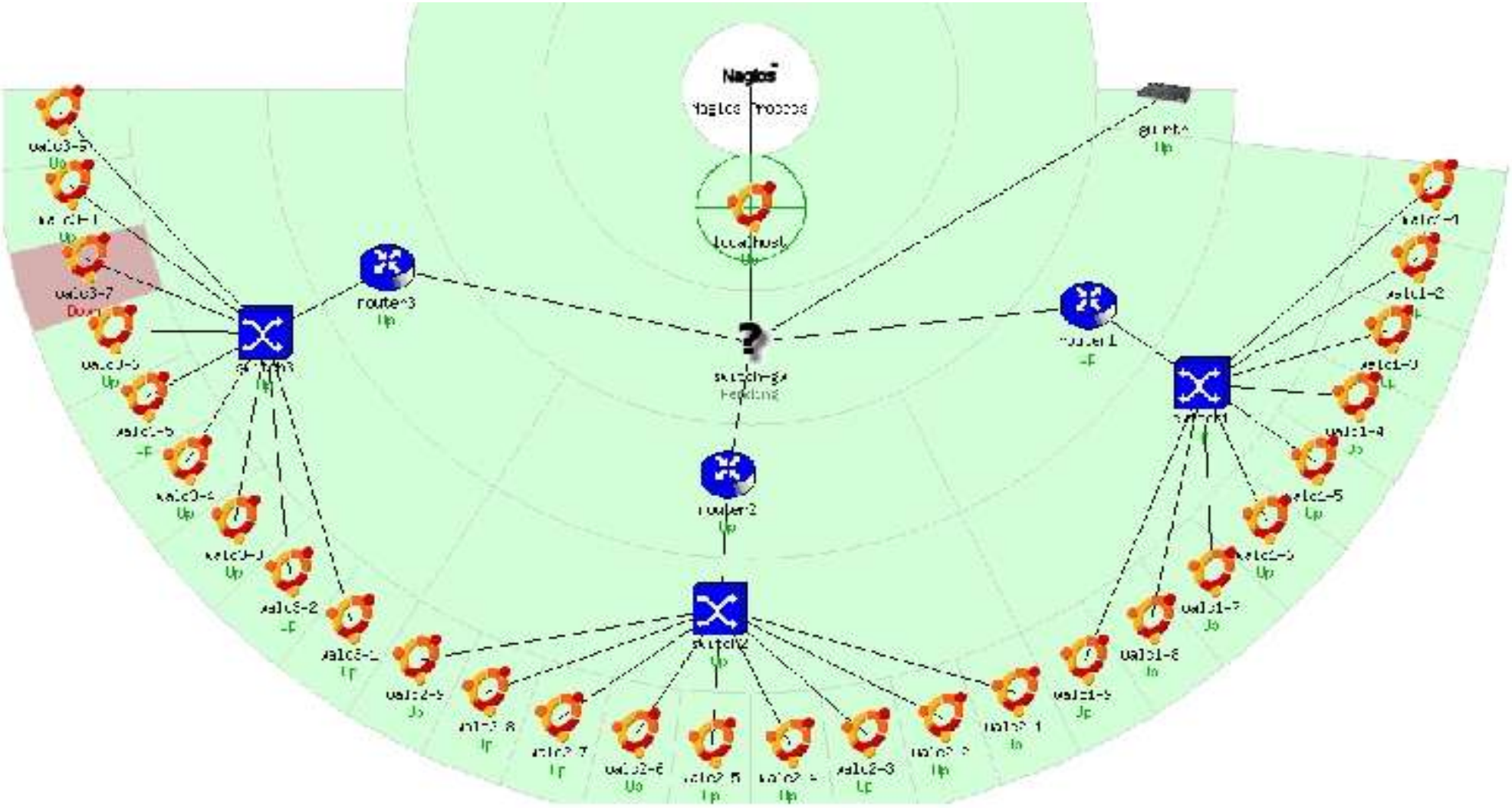
Hosts can have parents:

- 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).

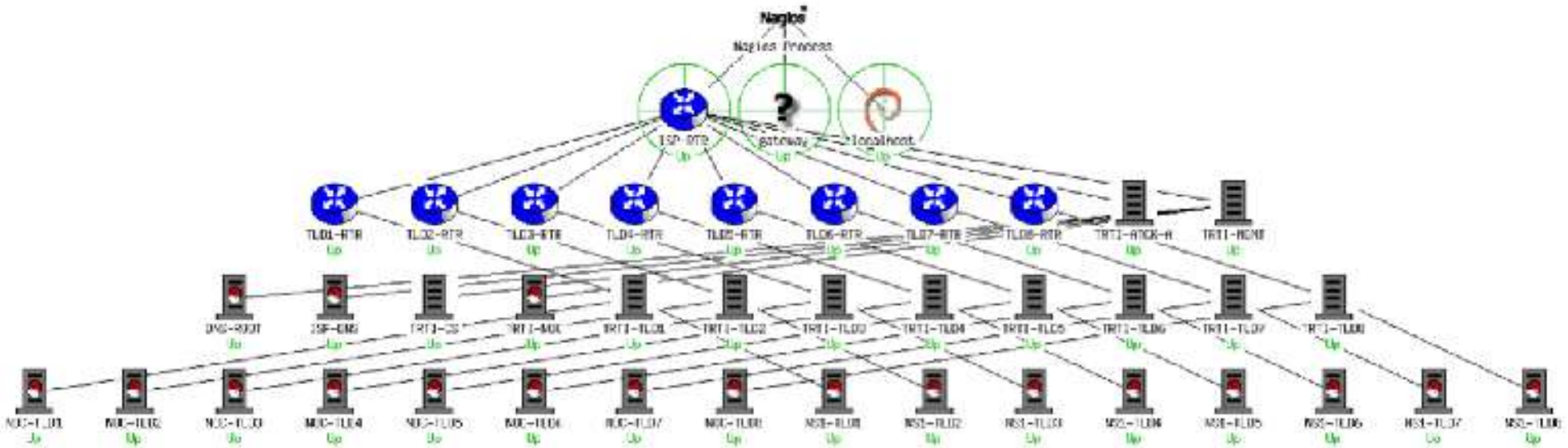
Network Viewpoint

- 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

Network Viewpoint



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.nsrc.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 pc1
   alias     pc1 in group 1
   address   pc1.ws.nsrc.org
   use       generic-host
}
```

copy settings from this template

- 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

Generic Host Template

generic-host_nagios2.cfg

```
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                    pc1
   alias                        pc1 in group 1
   address                      pc1.ws.nsrc.org
   use                          generic-host
   notification_interval        120
   contact_groups               admins,managers
}
```

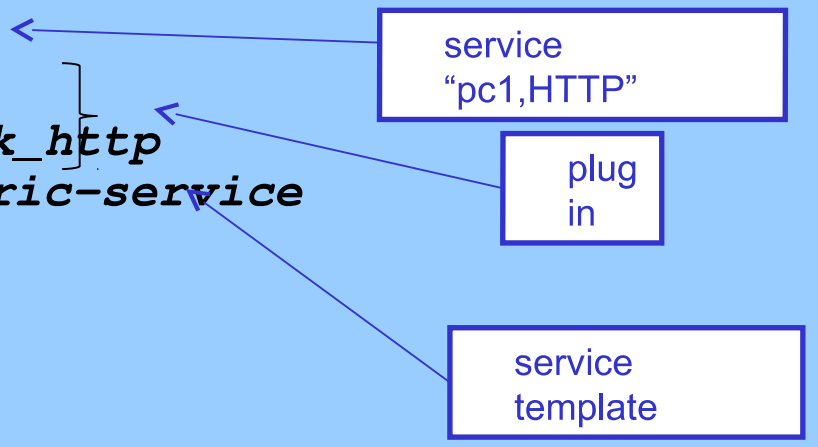
Defining Services: Direct Way

pcs.cfg

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

define service {
    host_name               pc1
    service_description     HTTP
    check_command           check_http
    use                     generic-service
}

define service {
    host_name               pc1
    service_description     SSH
    check_command           check_ssh
    use                     generic-service
}
```



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

Generic Service Templates

```
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
        }
```

generic-service_nagios2.cfg

(comments have been removed)

Overriding Defaults

Again, settings can be overridden per service

services_nagios2.cfg

```
define service {
   host_name               pc1
   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

```
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

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,debian-servers
}
```

extinfo_nagios2.cfg

```
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

servicegroups.cfg

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

Configuring Topology

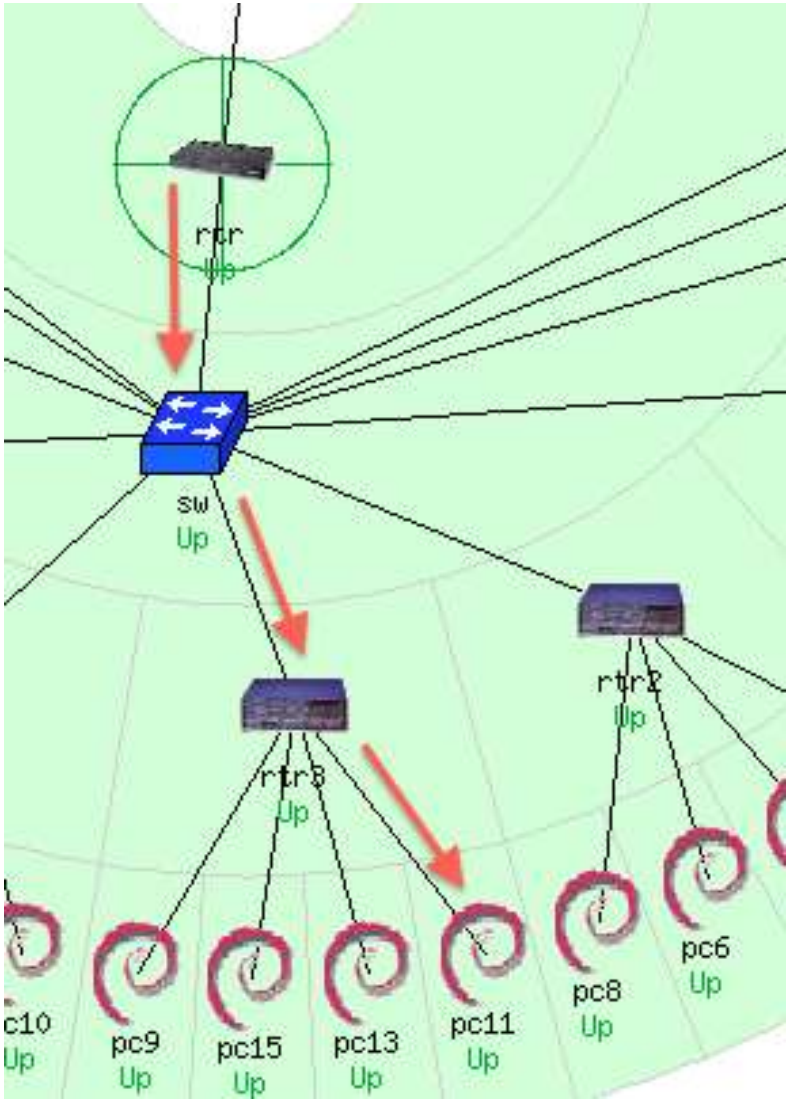
pcs.cfg

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

parent host

- 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

```
define host {
    use             generic-host
    host_name       rtr
    alias           Gateway Router
    address         10.10.0.254     }
```

SW

```
define host {
    use             generic-host
    host_name       sw
    alias           Backbone Switch
    address         10.10.0.253
    parents         rtr     }
```

RTR3

```
define host {
    use             generic-host
    host_name       rtr3
    alias           router 3
    address         10.10.3.254
    parents         sw      }
```

PC11…

Out of Band (OOB) Notifications

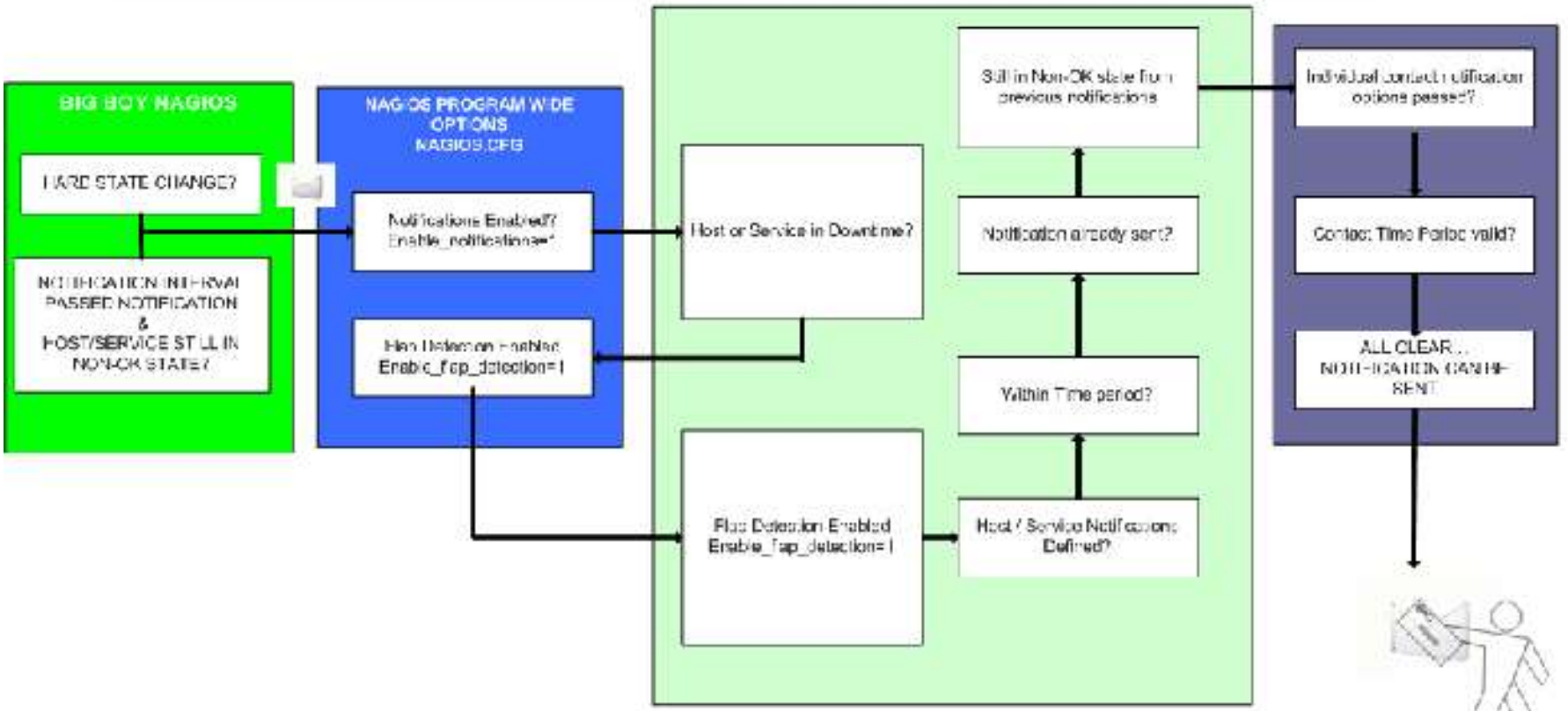
A critical item to remember: an SMS or message system that is independent from your network.

- 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/

NAGIOS - NOTIFICATION FLOW DIAGRAM



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

CONTACTED IS THE NOTIFICATION MESSAGE

References

- **Nagios web site**
  http://www.nagios.org/
- **Nagios plugins site**
  http://www.nagiosplugins.org/
- *Nagios System and Network Monitoring*, by Wolfgang Barth. Good book about Nagios.
- **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

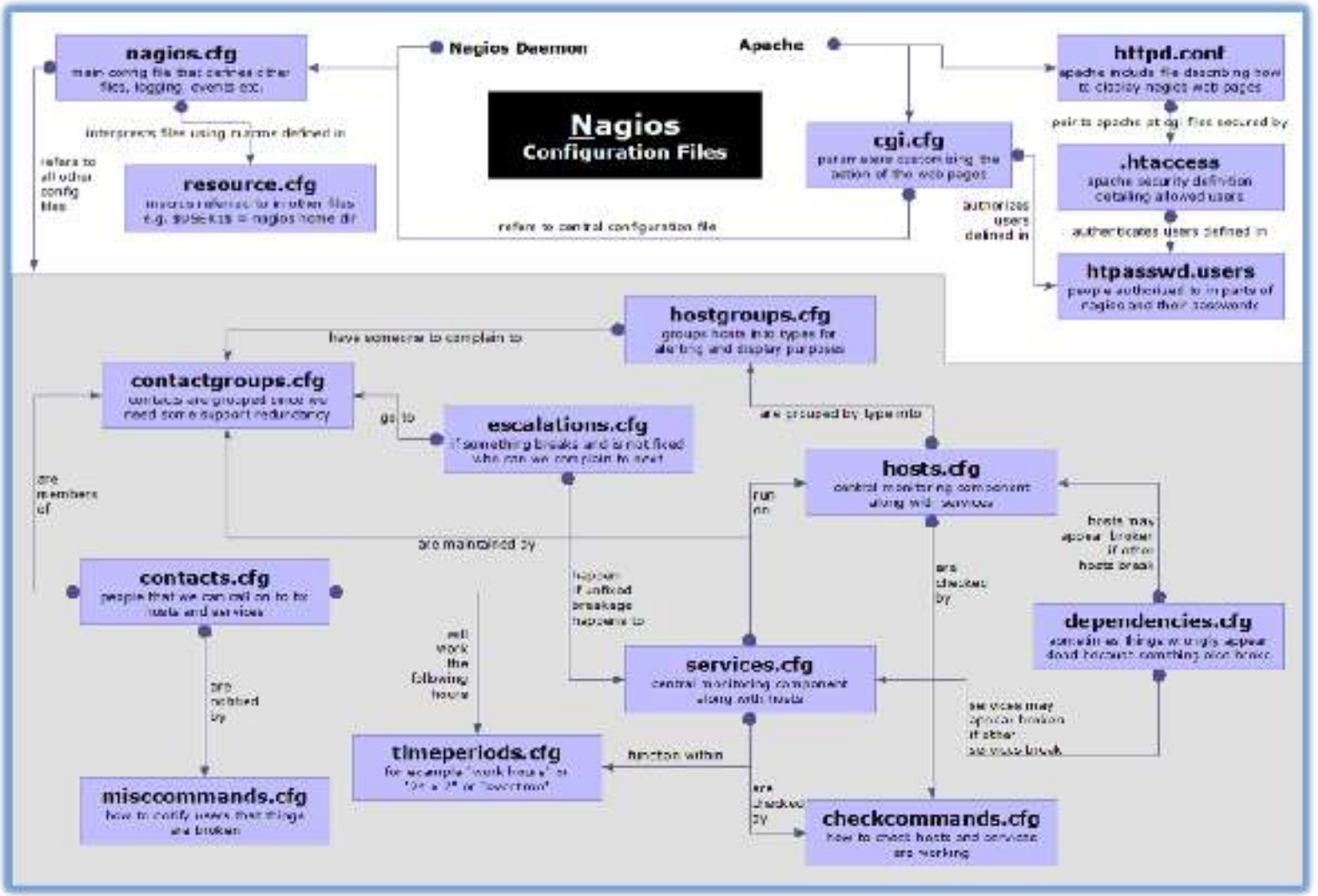
Service Notification Options

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



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: *Tunning 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
        }
```

Configuring Service/Host Checks

```
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 '$ARG1$' '$HOSTADDRESS$'
}
```

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

- 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
}
```

this is $ARG1$

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: $SERVICEDESC$\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: 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

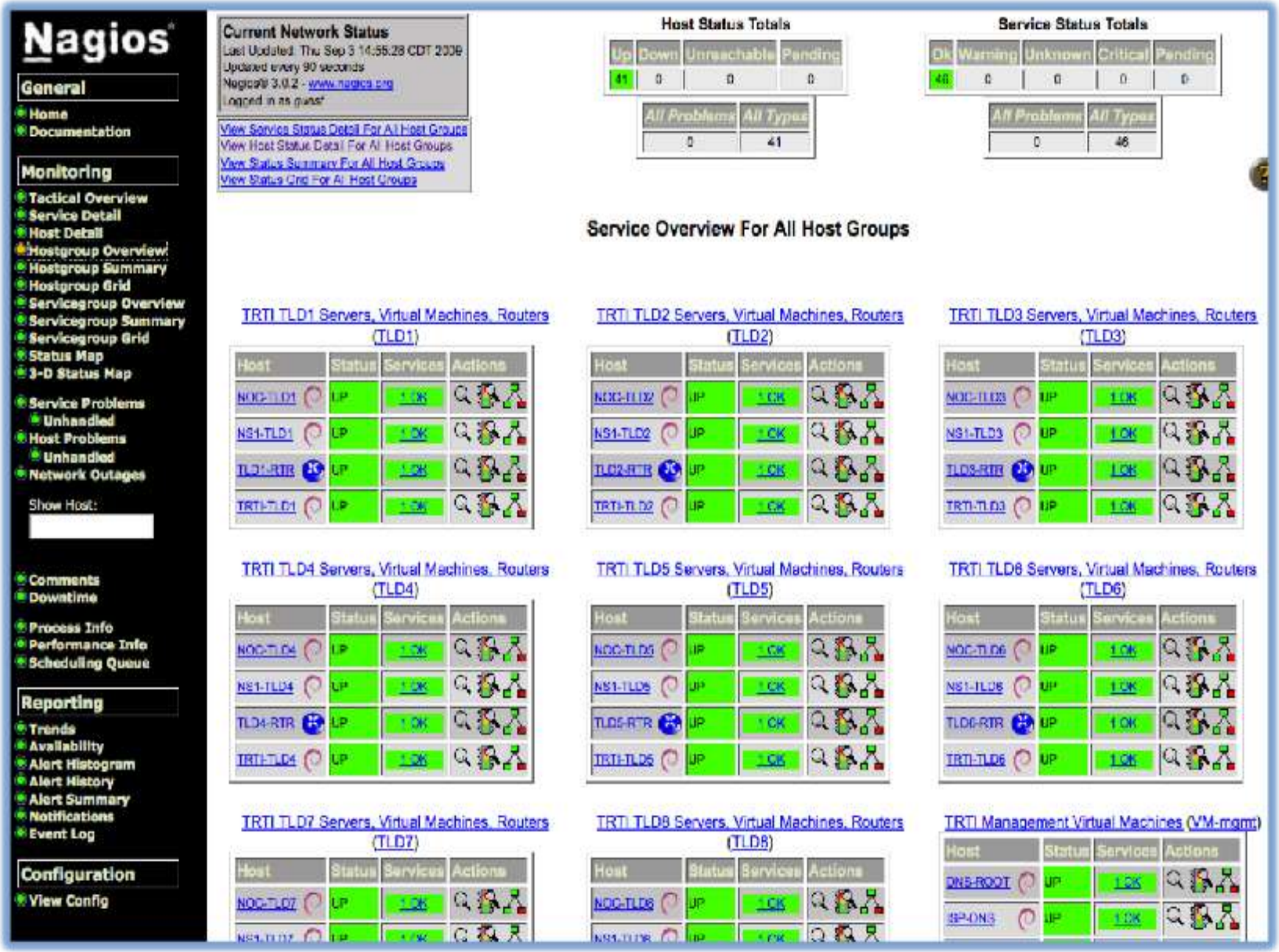
```
# check that ssh services are running
define service {
        hostgroup_name              ssh-servers
        service_description         SSH
        check_command               check_ssh
        use                         generic-service
        notification_interval       0
}
```

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

```
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.


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 14:55:28 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 Host Status Detail 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
Service Overview For All Host Groups
TRTI TLD1 Servers, Virtual Machines, Routers (TLD1)
TRTI TLD2 Servers, Virtual Machines, Routers (TLD2)
TRTI TLD3 Servers, Virtual Machines, Routers (TLD3)
TRTI TLD4 Servers, Virtual Machines, Routers (TLD4)
TRTI TLD5 Servers, Virtual Machines, Routers (TLD5)
TRTI TLD6 Servers, Virtual Machines, Routers (TLD6)
TRTI TLD7 Servers, Virtual Machines, Routers (TLD7)
TRTI TLD8 Servers, Virtual Machines, Routers (TLD8)
TRTI Management Virtual Machines (VM-mgmt)
Host Status Services Actions
NOC-TLD1 UP 1 OK
NS1-TLD1 UP 1 OK
TLD1-RTR UP 1 OK
TRTI-TLD1 UP 1 OK
NOC-TLD2 UP 1 OK
NS1-TLD2 UP 1 OK
TLD2-RTR UP 1 OK
TRTI-TLD2 UP 1 OK
NOC-TLD3 UP 1 OK
NS1-TLD3 UP 1 OK
TLD3-RTR UP 1 OK
TRTI-TLD3 UP 1 OK
NOC-TLD4 UP 1 OK
NS1-TLD4 UP 1 OK
TLD4-RTR UP 1 OK
TRTI-TLD4 UP 1 OK
NOC-TLD5 UP 1 OK
NS1-TLD5 UP 1 OK
TLD5-RTR UP 1 OK
TRTI-TLD5 UP 1 OK
NOC-TLD6 UP 1 OK
NS1-TLD6 UP 1 OK
TLD6-RTR UP 1 OK
TRTI-TLD6 UP 1 OK
NOC-TLD7 UP 1 OK
NOC-TLD8 UP 1 OK
DNS-ROOT UP 1 OK
SP-DNS UP 1 OK

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: Fri Sep 4 13:29:20 CDT 2009
Updated every 90 seconds
Nagios® 3.0.2 - www.nagios.org
Logged in as guest

View Service Status Detail For All Service Groups
View Status Summary For All Service Groups
View Service Status Grid For All Service 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
53	0	0	1	0

All Problems	All Types
1	54

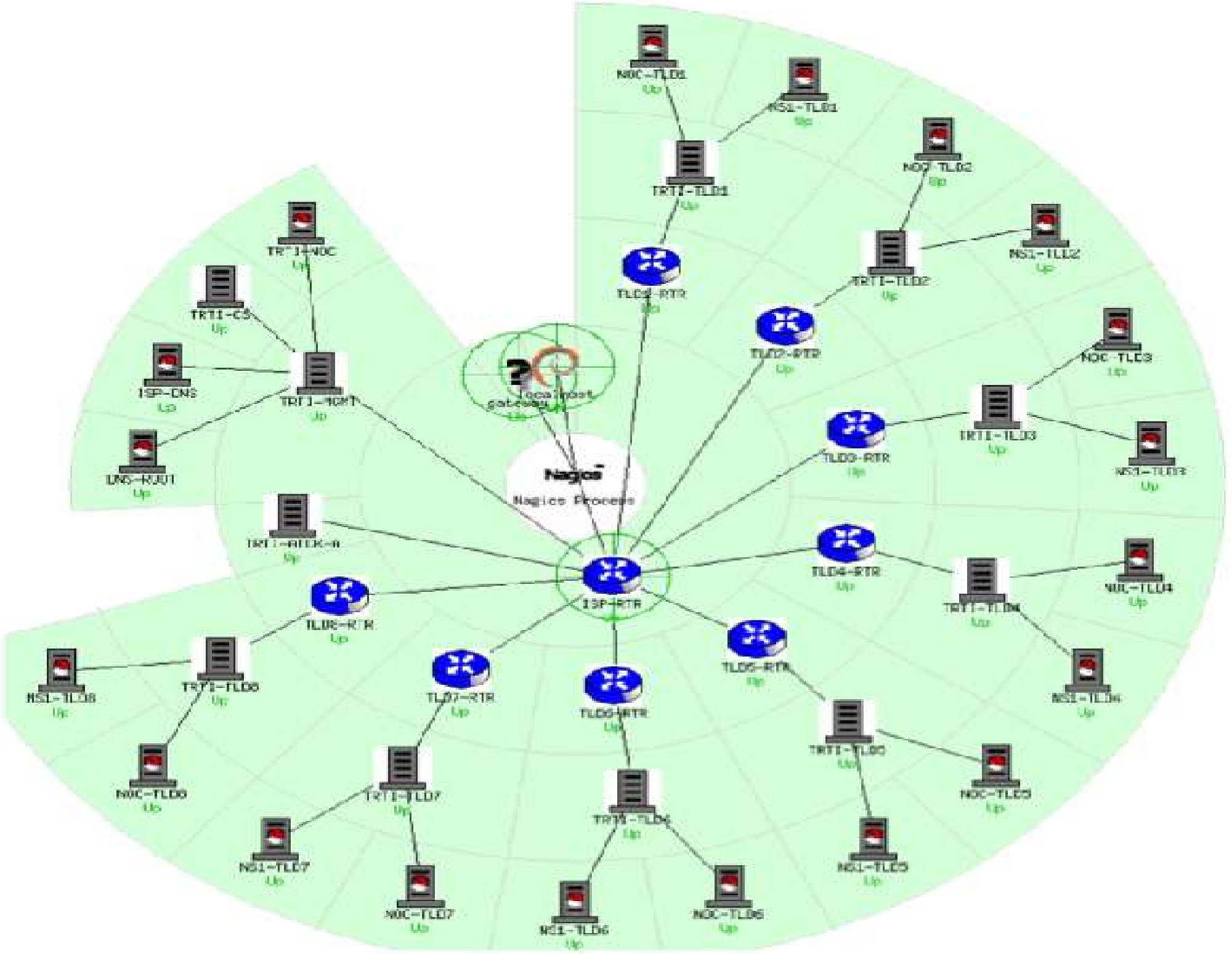
Service Overview For All Service Groups

TLD Servers running Nagios (NAGIOS)

Host	Status	Services	Actions
NS1-TLD1	UP	1 OK	
NS1-TLD2	UP	1 OK	
NS1-TLD3	UP	1 OK	
NS1-TLD4	UP	1 OK	
NS1-TLD5	UP	1 OK	
NS1-TLD6	UP	1 OK	
NS1-TLD7	UP	1 OK	
NS1-TLD8	UP	1 OK	

TLD Servers running SSH (SSH)

Host	Status	Services	Actions
NS1-TLD1	UP	1 OK	
NS1-TLD2	UP	1 CRITICAL	
NS1-TLD3	UP	1 OK	
NS1-TLD4	UP	1 OK	
NS1-TLD5	UP	1 OK	
NS1-TLD6	UP	1 OK	
NS1-TLD7	UP	1 OK	
NS1-TLD8	UP	1 OK	


NOC-TLD1
NS1-TLD1
TRI1-TLD1
TLD1-RTR
TRI1-NOC
TRI1-CS
ISP-DNS
TRI1-MGMT
UNS-RUU1
localhost
gateway
Nagios
Nagios Process
NOC-TLD2
NS1-TLD2
TRI1-TLD2
TLD2-RTR
NOC-TLD3
TRI1-TLD3
TLD3-RTR
NS1-TLD3
TRI1-ATTK-R
ISP-RTR
TLD4-RTR
TRI1-TLD4
NOC-TLD4
TLDR-RTR
NS1-TLD8
TRI1-TLD8
TLD7-RTR
TLD8-RTR
TLD5-RTR
NS1-TLD4
NOC-TLD8
TRI1-TLD7
TRI1-TLD6
TRI1-TLD5
NOC-TLD5
NS1-TLD7
NOC-TLD7
NS1-TLD6
NOC-TLD6
NS1-TLD5

More Sample Screenshots

Many more sample Nagios screenshots available here:

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