Network Monitoring & Management
Nagios

Network Startup Resource Center
www.nsrc.org

Introduction

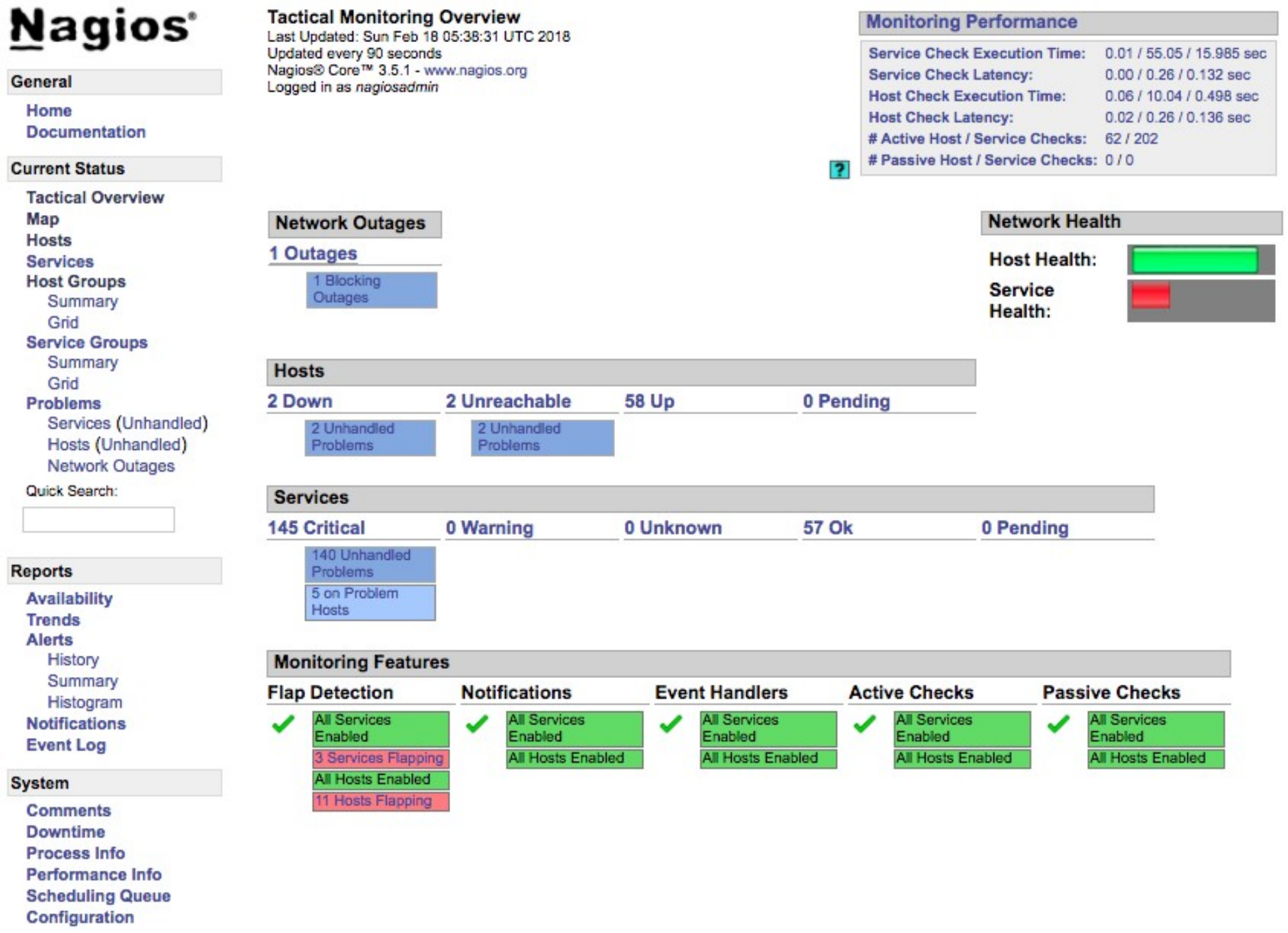
- Possibly the most used open source network monitoring software
- Web interface for viewing status, browsing history, scheduling downtime etc|
- Sends out alerts via E-mail. Can be configured to use other mechanisms, e.g. SMS

Introduction

Nagios actively monitors availability of

- Hosts (devices)
- Services

Nagios: Tactical Overview



Host Detail View

Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends
- Alerts
  - History
  - Summary
  - Histogram
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Current Network Status
Last Updated: Sun Feb 18 05:38:00 UTC 2018
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *nagiosadmin*

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
58	2	2	0

All Problems	All Types
4	62

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
57	0	0	145	0

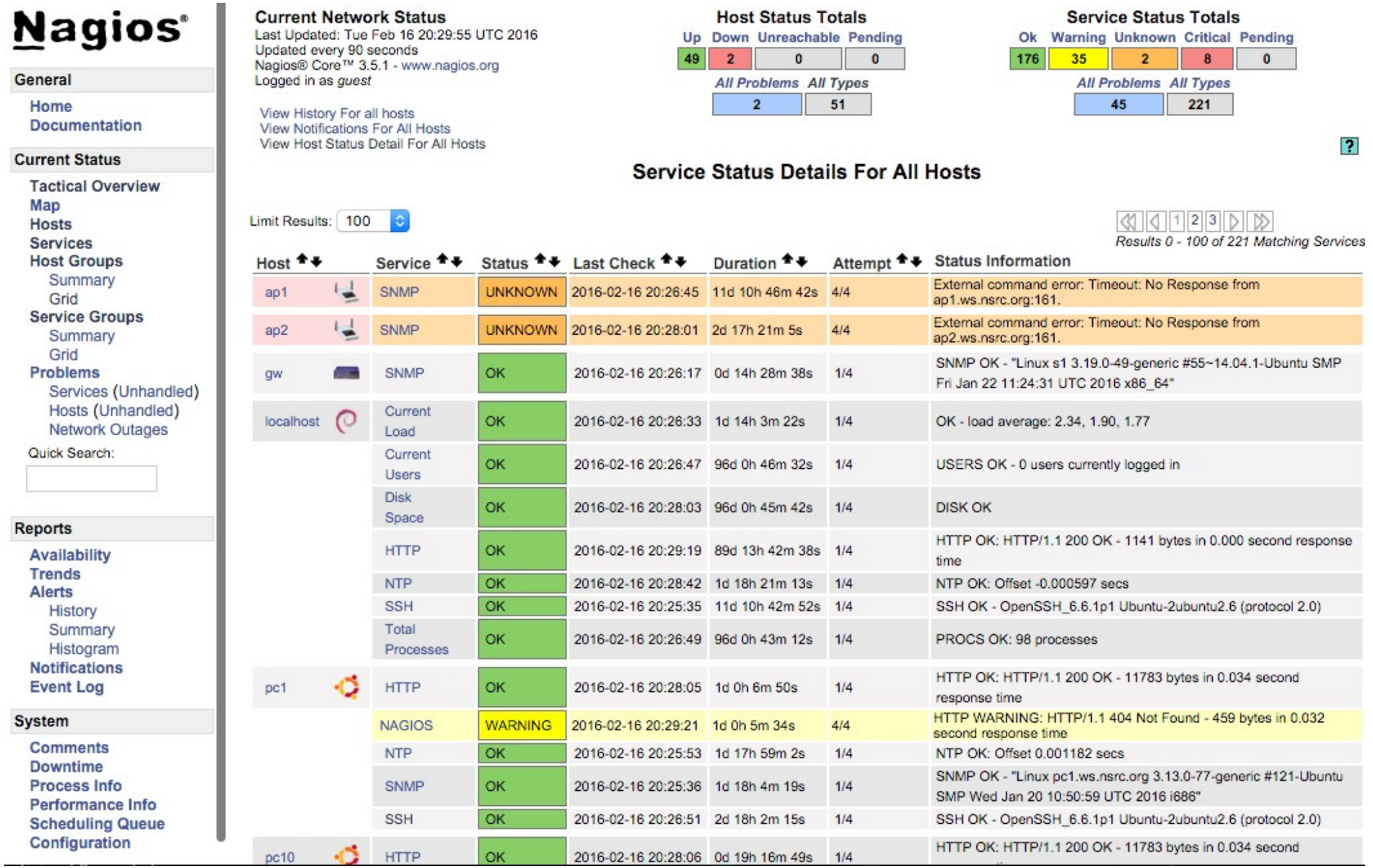
All Problems	All Types
145	202

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
ap1	UNREACHABLE	2018-02-18 05:36:41	0d 0h 16m 39s	CRITICAL: IPv4/ap1.ws.nsrc.org CRITICAL
ap2	UNREACHABLE	2018-02-18 05:36:31	133d 16h 57m 15s	CRITICAL: IPv4/ap2.ws.nsrc.org CRITICAL
bdr1.campus1	UP	2018-02-18 05:33:51	0d 0h 14m 29s	OK: IPv6/bdr1.campus1.ws.nsrc.org OK, IPv4/bdr1.campus1.ws.nsrc.org OK
bdr1.campus2	UP	2018-02-18 05:33:51	0d 0h 14m 39s	OK: IPv6/bdr1.campus2.ws.nsrc.org OK, IPv4/bdr1.campus2.ws.nsrc.org OK
bdr1.campus3	UP	2018-02-18 05:33:31	0d 0h 14m 29s	OK: IPv6/bdr1.campus3.ws.nsrc.org OK, IPv4/bdr1.campus3.ws.nsrc.org OK
bdr1.campus4	UP	2018-02-18 05:33:51	0d 0h 14m 39s	OK: IPv6/bdr1.campus4.ws.nsrc.org OK, IPv4/bdr1.campus4.ws.nsrc.org OK
bdr1.campus5	UP	2018-02-18 05:36:31	0d 0h 14m 49s	OK: IPv6/bdr1.campus5.ws.nsrc.org OK, IPv4/bdr1.campus5.ws.nsrc.org OK
bdr1.campus6	UP	2018-02-18 05:36:41	0d 0h 14m 49s	OK: IPv6/bdr1.campus6.ws.nsrc.org OK, IPv4/bdr1.campus6.ws.nsrc.org OK
core1.campus1	UP	2018-02-18 05:33:41	0d 0h 14m 39s	OK: IPv6/core1.campus1.ws.nsrc.org OK, IPv4/core1.campus1.ws.nsrc.org OK
core1.campus2	UP	2018-02-18 05:33:41	0d 0h 14m 49s	OK: IPv6/core1.campus2.ws.nsrc.org OK, IPv4/core1.campus2.ws.nsrc.org OK
core1.campus3	UP	2018-02-18 05:33:21	0d 0h 14m 39s	OK: IPv6/core1.campus3.ws.nsrc.org OK, IPv4/core1.campus3.ws.nsrc.org OK
core1.campus4	UP	2018-02-18 05:33:21	0d 0h 14m 49s	OK: IPv6/core1.campus4.ws.nsrc.org OK, IPv4/core1.campus4.ws.nsrc.org OK
core1.campus5	UP	2018-02-18 05:37:01	0d 0h 14m 59s	OK: IPv6/core1.campus5.ws.nsrc.org OK, IPv4/core1.campus5.ws.nsrc.org OK
core1.campus6	UP	2018-02-18 05:37:11	0d 0h 14m 59s	OK: IPv6/core1.campus6.ws.nsrc.org OK, IPv4/core1.campus6.ws.nsrc.org OK
gw	UP	2018-02-18 05:37:11	3d 9h 42m 19s	OK: IPv6/gw.ws.nsrc.org OK, IPv4/gw.ws.nsrc.org OK
host1.campus1	UP	2018-02-18 05:33:51	0d 0h 14m 29s	OK: IPv6/host1.campus1.ws.nsrc.org OK, IPv4/host1.campus1.ws.nsrc.org OK
host1.campus2	UP	2018-02-18 05:33:51	0d 0h 14m 39s	OK: IPv6/host1.campus2.ws.nsrc.org OK, IPv4/host1.campus2.ws.nsrc.org OK
host1.campus3	UP	2018-02-18 05:33:31	0d 0h 14m 29s	OK: IPv6/host1.campus3.ws.nsrc.org OK, IPv4/host1.campus3.ws.nsrc.org OK
host1.campus4	UP	2018-02-18 05:33:51	0d 0h 14m 39s	OK: IPv6/host1.campus4.ws.nsrc.org OK, IPv4/host1.campus4.ws.nsrc.org OK
host1.campus5	UP	2018-02-18 05:37:41	0d 0h 14m 49s	OK: IPv6/host1.campus5.ws.nsrc.org OK, IPv4/host1.campus5.ws.nsrc.org OK
host1.campus6	UP	2018-02-18 05:37:41	0d 0h 14m 49s	OK: IPv6/host1.campus6.ws.nsrc.org OK, IPv4/host1.campus6.ws.nsrc.org OK
host2.campus1	UP	2018-02-18 05:33:51	0d 0h 14m 29s	OK: IPv6/host2.campus1.ws.nsrc.org OK, IPv4/host2.campus1.ws.nsrc.org OK
host2.campus2	UP	2018-02-18 05:33:51	0d 0h 14m 39s	OK: IPv6/host2.campus2.ws.nsrc.org OK, IPv4/host2.campus2.ws.nsrc.org OK
host2.campus3	UP	2018-02-18 05:33:31	0d 0h 14m 29s	OK: IPv6/host2.campus3.ws.nsrc.org OK, IPv4/host2.campus3.ws.nsrc.org OK
host2.campus4	UP	2018-02-18 05:33:51	0d 0h 14m 39s	OK: IPv6/host2.campus4.ws.nsrc.org OK, IPv4/host2.campus4.ws.nsrc.org OK

Service Detail View



Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends
- Alerts
  - History
  - Summary
  - Histogram
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Current Network Status
Last Updated: Tue Feb 16 20:29:55 UTC 2016
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *guest*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
49	2	0	0

All Problems	All Types
2	51

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
176	35	2	8	0

All Problems	All Types
45	221

Service Status Details For All Hosts

Limit Results: 100

Results 0 - 100 of 221 Matching Services

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ap1	SNMP	UNKNOWN	2016-02-16 20:26:45	11d 10h 46m 42s	4/4	External command error: Timeout: No Response from ap1.ws.nsrc.org:161.
ap2	SNMP	UNKNOWN	2016-02-16 20:28:01	2d 17h 21m 5s	4/4	External command error: Timeout: No Response from ap2.ws.nsrc.org:161.
gw	SNMP	OK	2016-02-16 20:26:17	0d 14h 28m 38s	1/4	SNMP OK - "Linux s1 3.19.0-49-generic #55~14.04.1-Ubuntu SMP Fri Jan 22 11:24:31 UTC 2016 x86_64"
localhost	Current Load	OK	2016-02-16 20:26:33	1d 14h 3m 22s	1/4	OK - load average: 2.34, 1.90, 1.77
	Current Users	OK	2016-02-16 20:26:47	96d 0h 46m 32s	1/4	USERS OK - 0 users currently logged in
	Disk Space	OK	2016-02-16 20:28:03	96d 0h 45m 42s	1/4	DISK OK
	HTTP	OK	2016-02-16 20:29:19	89d 13h 42m 38s	1/4	HTTP OK: HTTP/1.1 200 OK - 1141 bytes in 0.000 second response time
	NTP	OK	2016-02-16 20:28:42	1d 18h 21m 13s	1/4	NTP OK: Offset -0.000597 secs
	SSH	OK	2016-02-16 20:25:35	11d 10h 42m 52s	1/4	SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6 (protocol 2.0)
	Total Processes	OK	2016-02-16 20:26:49	96d 0h 43m 12s	1/4	PROCS OK: 98 processes
pc1	HTTP	OK	2016-02-16 20:28:05	1d 0h 6m 50s	1/4	HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.034 second response time
	NAGIOS	WARNING	2016-02-16 20:29:21	1d 0h 5m 34s	4/4	HTTP WARNING: HTTP/1.1 404 Not Found - 459 bytes in 0.032 second response time
	NTP	OK	2016-02-16 20:25:53	1d 17h 59m 2s	1/4	NTP OK: Offset 0.001182 secs
	SNMP	OK	2016-02-16 20:25:36	1d 18h 4m 19s	1/4	SNMP OK - "Linux pc1.ws.nsrc.org 3.13.0-77-generic #121-Ubuntu SMP Wed Jan 20 10:50:59 UTC 2016 i686"
	SSH	OK	2016-02-16 20:26:51	2d 18h 2m 15s	1/4	SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6 (protocol 2.0)
pc10	HTTP	OK	2016-02-16 20:28:06	0d 19h 16m 49s	1/4	HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.034 second

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

```
2. nsrc@s1: ~ (ssh)
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_clamd     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_ircd          check_mrtgtraf  check_ntp_peer     check_procs     check_spop       check_users
nsrc@s1:~$
```

```
2. nsrc@s1: ~ (ssh)
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

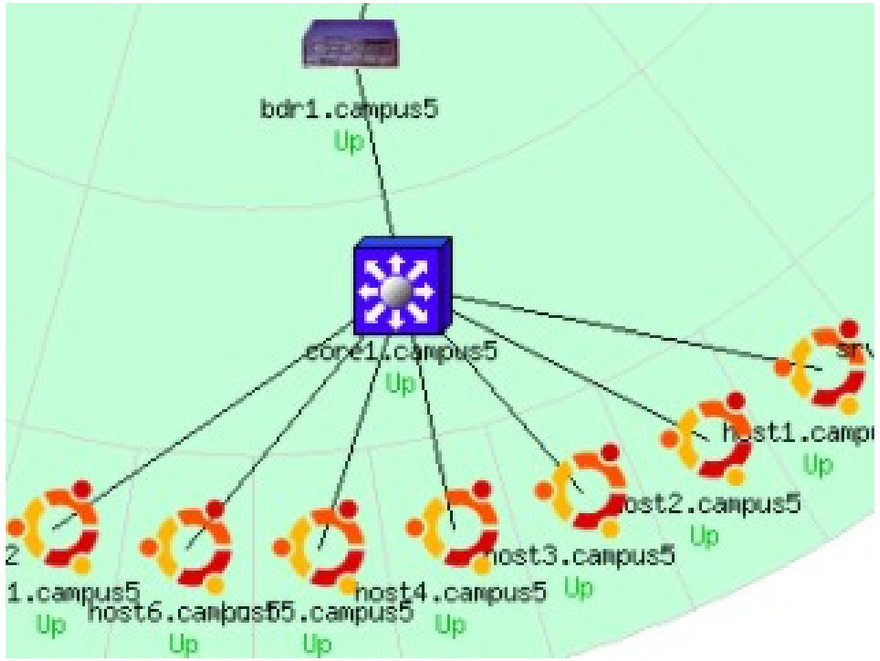
Scheduling

- 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:

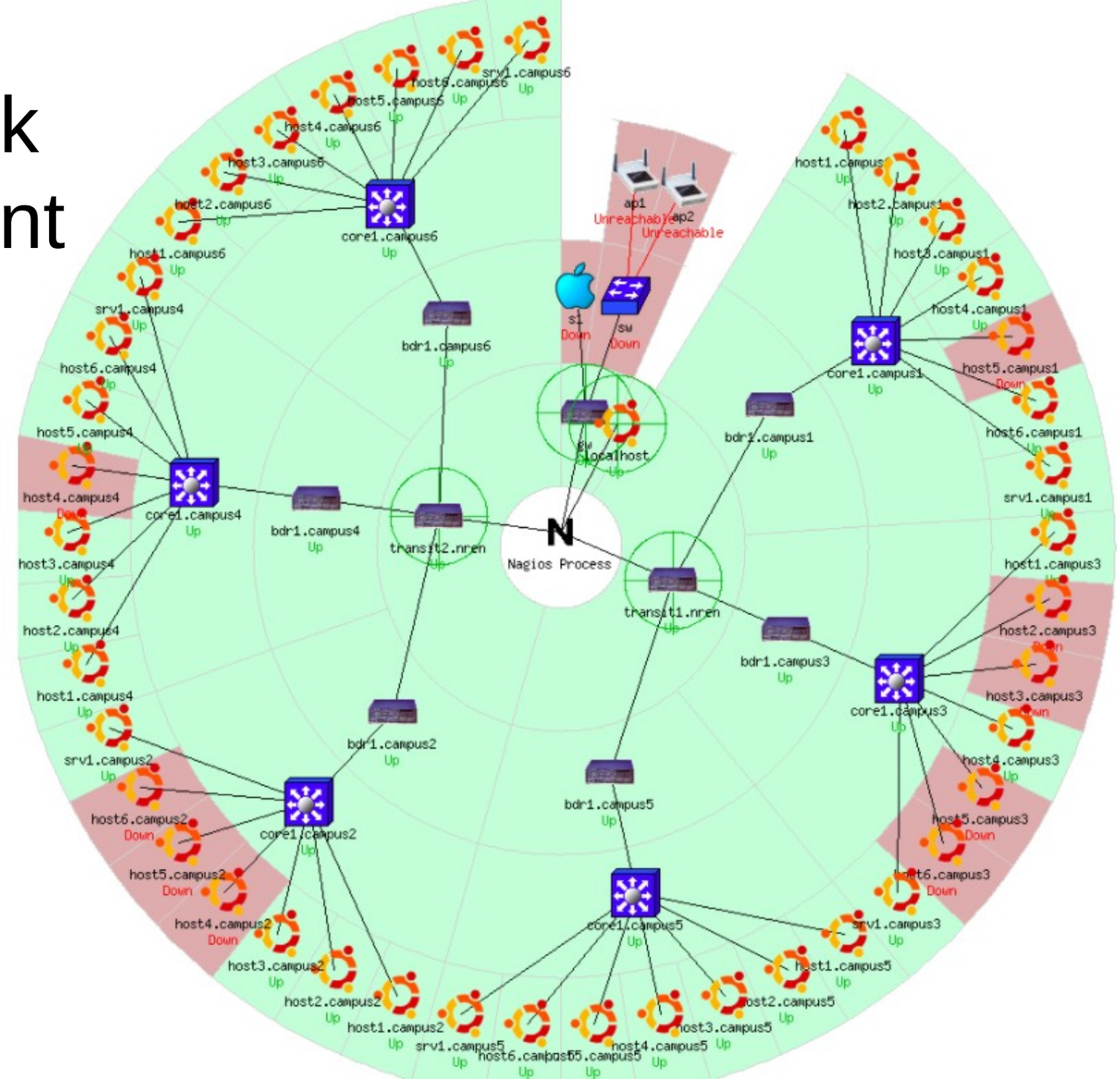
- The parent of a server connected to a switch would be the switch.
- Allows us to specify the dependencies between devices.
- Avoids sending alarms when parent does not respond.
- A node can have multiple parents (dual homed).



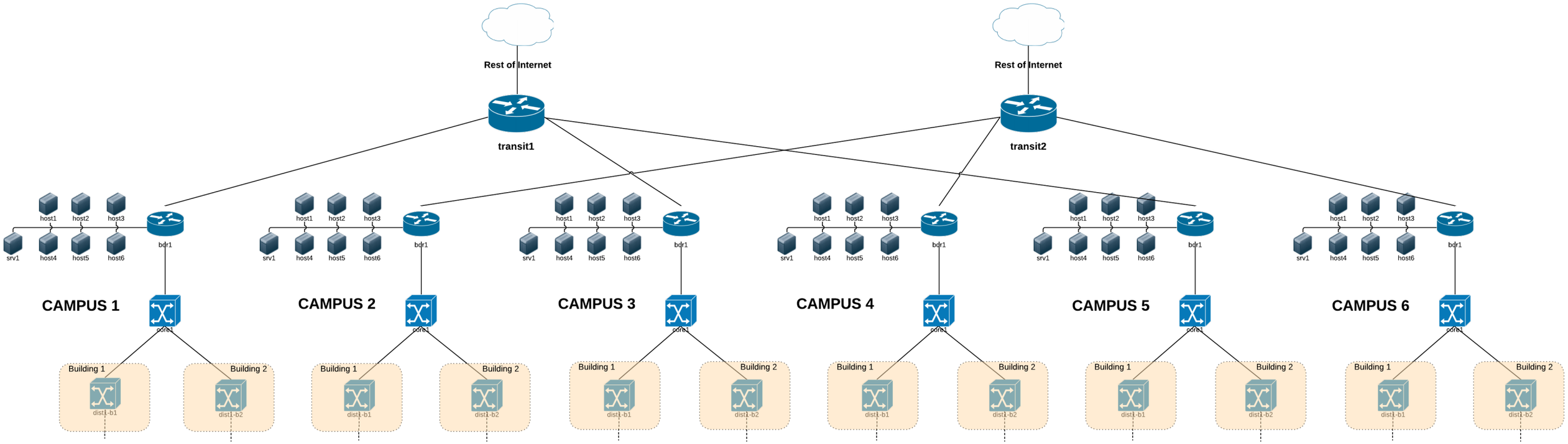
Network Viewpoint

- 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

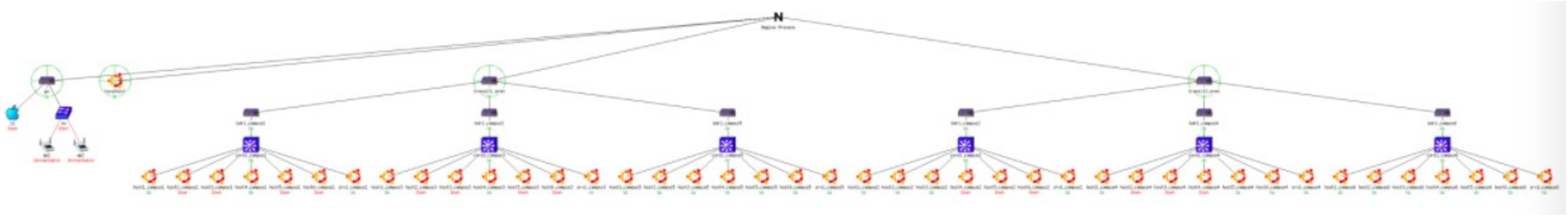


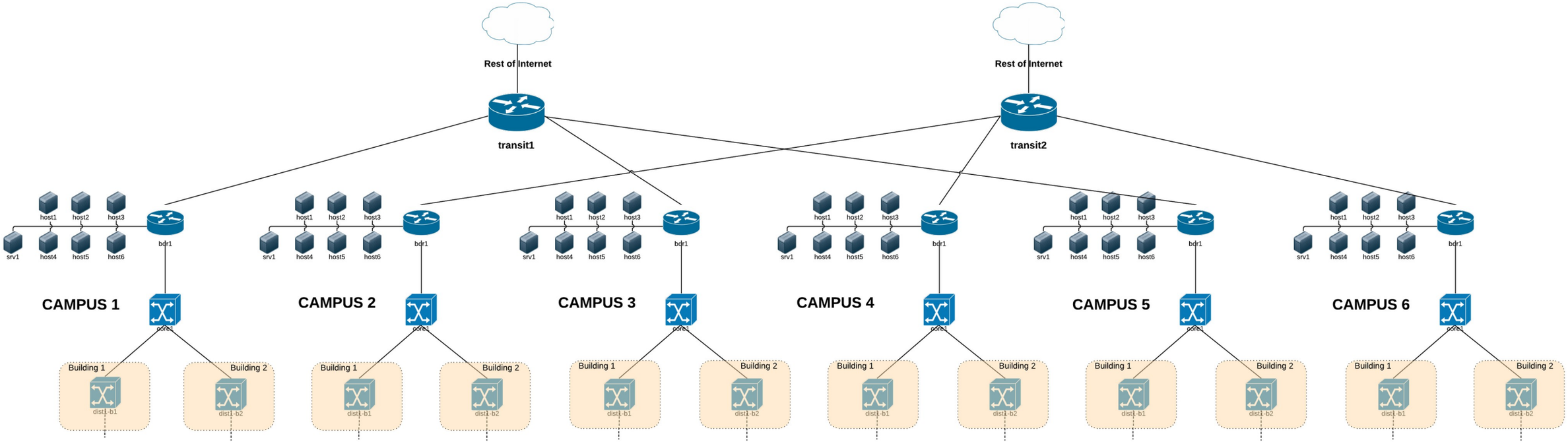
Collapsed Tree Network View



Do you recognize this…?

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://hostX.campusY.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 is defined in text files in directory:
  - `/etc/nagios3/conf.d/*.cfg`
  - Details on these files is available at:
  - http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html

- Default configuration is in several files with different objects in different files, but 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

```
define host {
        alias           Host 1 Campus 1
        host_name       host1.campus1
        address         host1.campus1.ws.nsrc.org
        use             generic-host
}
```

copy settings from this template

This is a minimal working configuration

- 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!
}
```

- There is a lot defined here
- We'll explain what these items mean

Overriding Defaults

All settings in `generic-host_nagios2.cfg` can be overridden per host in the local definition

For example:

```
define host {
        alias           Host 1 Campus 1
        host_name       host1.campus1
        address         host1.campus1.ws.nsrc.org
        use             generic-host
        notification_interval   120
        contact_groups          admins,managers
}
```

The template we override below

Defining Services: Direct Way

```
define host {
       alias           Server 1 Campus 1
       host_name       host1.campus1
       address         host1.campus1.ws.nsrc.org
       use             generic-host
}

define service {
   host_name                host1.campus1
   service_description      HTTP
   check_command            check_http
   use                      generic-service
}

define service {
   host_name                host1.campus1
   service_description      SSH
   check_command            check_ssh
   use                      generic-service
}
```

service "host1.campus1,HTTP"

plugin

service template

service "host1.campus1,SSH"

Service Checks

The combination of host + service is a unique identifier for the service check, e.g.

- "host1.campus1,HTTP"
- "host1.campus1,SSH"
- "host2.campus1,HTTP"
- "host2.campus1,SSH"

- *check_command* points to the plugin
  - Each plugin has options you can specify, if you wish otherwise defaults are often just fine.
- *service template* pulls in settings for how often the check is done, and who and when to alert

Generic Service Templates

generic-service_nagios2.cfg

```
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
        }
```

(comments have been removed)

Overriding Defaults

Again, settings can be overridden per service

services_nagios2.cfg

```
define service {
   host_name               host1.campus1
   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          host1.campus1,host2.campus1
}

define hostgroup {
   hostgroup_name   ssh-servers
   alias            SSH servers
   members          host1.campus1,host2.campus1
}
```

Monitoring Services in Hostgroups

services_nagios2.cfg

```
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
}
```

if hostgroup "http-servers" contains srv1.campus1 & srv2.campus1 then Nagios creates HTTP service checks for both hosts. The service checks are called "srv1.campus1,HTTP" and "srv2.campus1,HTTP"

Alternative View

- “This hostgroup contains these Servers”

or:

- “This server belongs to these hostgroups”
- No need for “members” line in hostgroups file

Alternative Group Membership

```
define host {
       alias          Host 1 Campus 1
       host_name      host1.campus1
       address        host1.campus1.ws.nsrc.org
       use            generic-host
       hostgroups     ssh-servers,http-servers
}

define host {
       alias          Host 2 Campus 1
       host_name      host2.campus1
       address        host2.campus1.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

```
define host {
       alias          Host 1 Campus 1
       host_name      host1.campus1
       address        host1.campus1.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

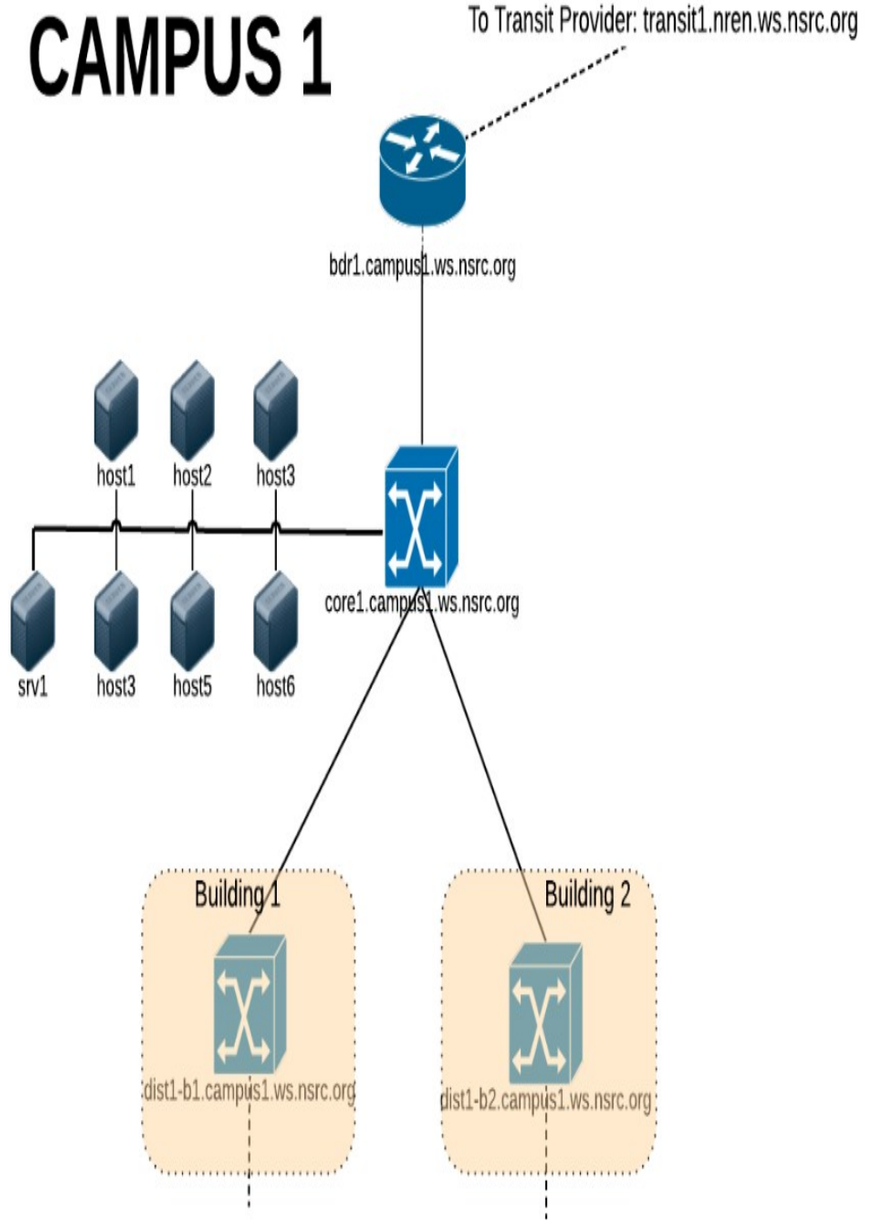
```
define host {
       alias           Host1 Campus1
       host_name       host1.campus1
       address         host1.campus1.ws.nsrc.org
       use             generic-host
       parents         core1.campus1
}
```

parent host

- Indicates “host1.campus1 is on the far side of core1.campus1”
- If core1.campus1 goes down, host1.campus1 is “unreachable”, not “down”

- Prevents a cascade of alerts if core1.campus1 goes down
- Also allows Nagios to draw cool status map

Another View of Configuration



transit1.nren

```
define host {
    use             generic-host
    host_name       transit1.nren
    alias           Transit Router 1
    address         transit1.nren.ws.nsrc.org  }
```

bdr1.campus1

```
define host {
    use             generic-host
    host_name       bdr1.campus1
    alias           Border Router 1, Campus 1
    address         bdr1.campus1.ws.nsrc.org
    parents         transit1.nren       }
```

core1.campus1

```
define host {
    use             generic-host
    host_name       core1.campus1
    alias           Core Router 1, Campus1
    address         core1.campus1.ws.nsrc.org
    parents         bdr1.campus1      }
```

host1.campus1…
srv1.campus1… → **core1.campus1**

dist1-b1.campus1… → **core1.campus1**

Out of Band (OOB) Notifications

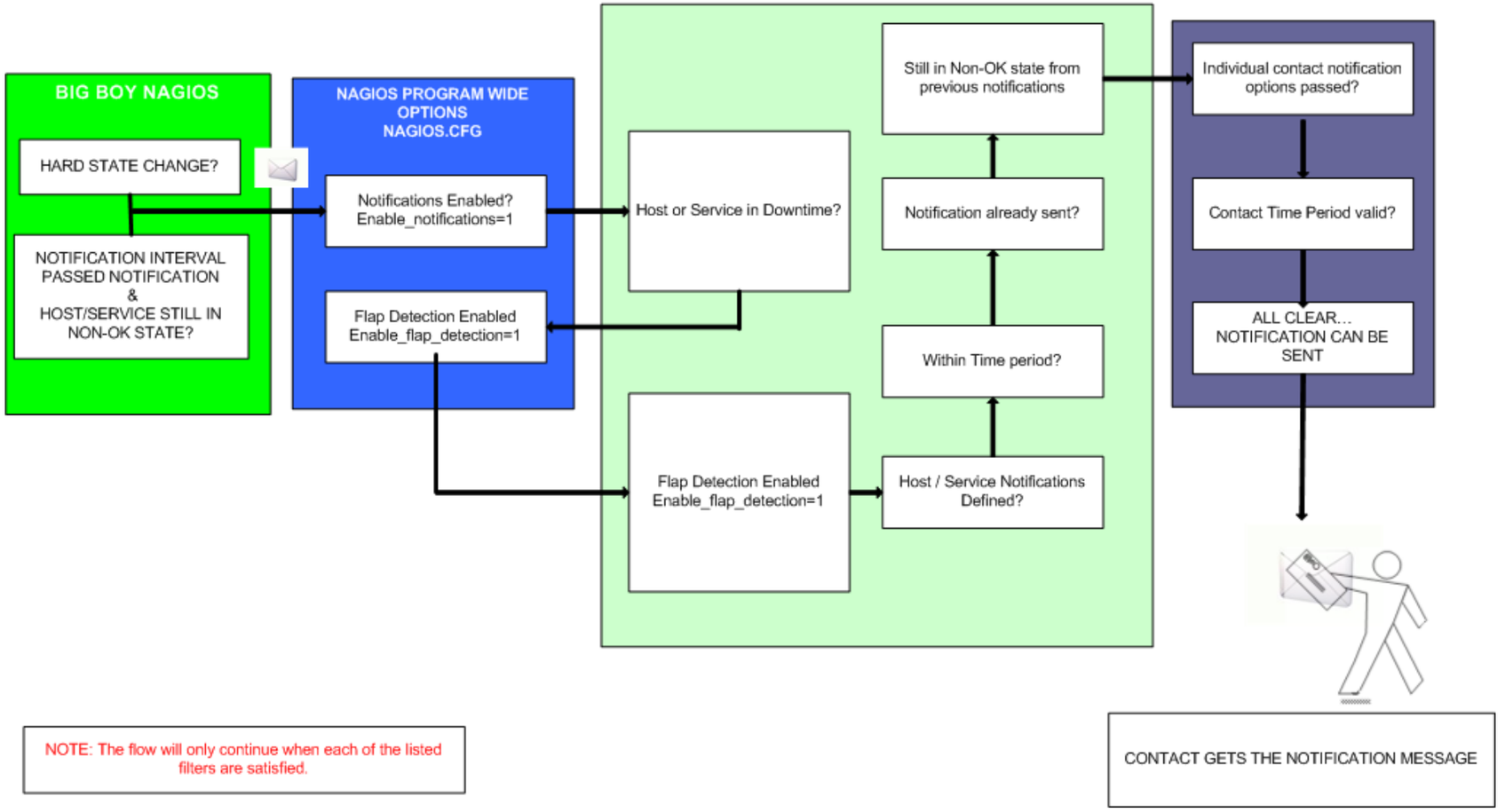
A critical item to remember: an SMS or message system that is independent from your network.

- 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/
Kannel*: http://www.kannel.org/

*Can be used on a Raspberry Pi

NAGIOS - NOTIFICATION FLOW DIAGRAM



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

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…

Your instructor may go over some of these, including the various states for hosts and services…

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

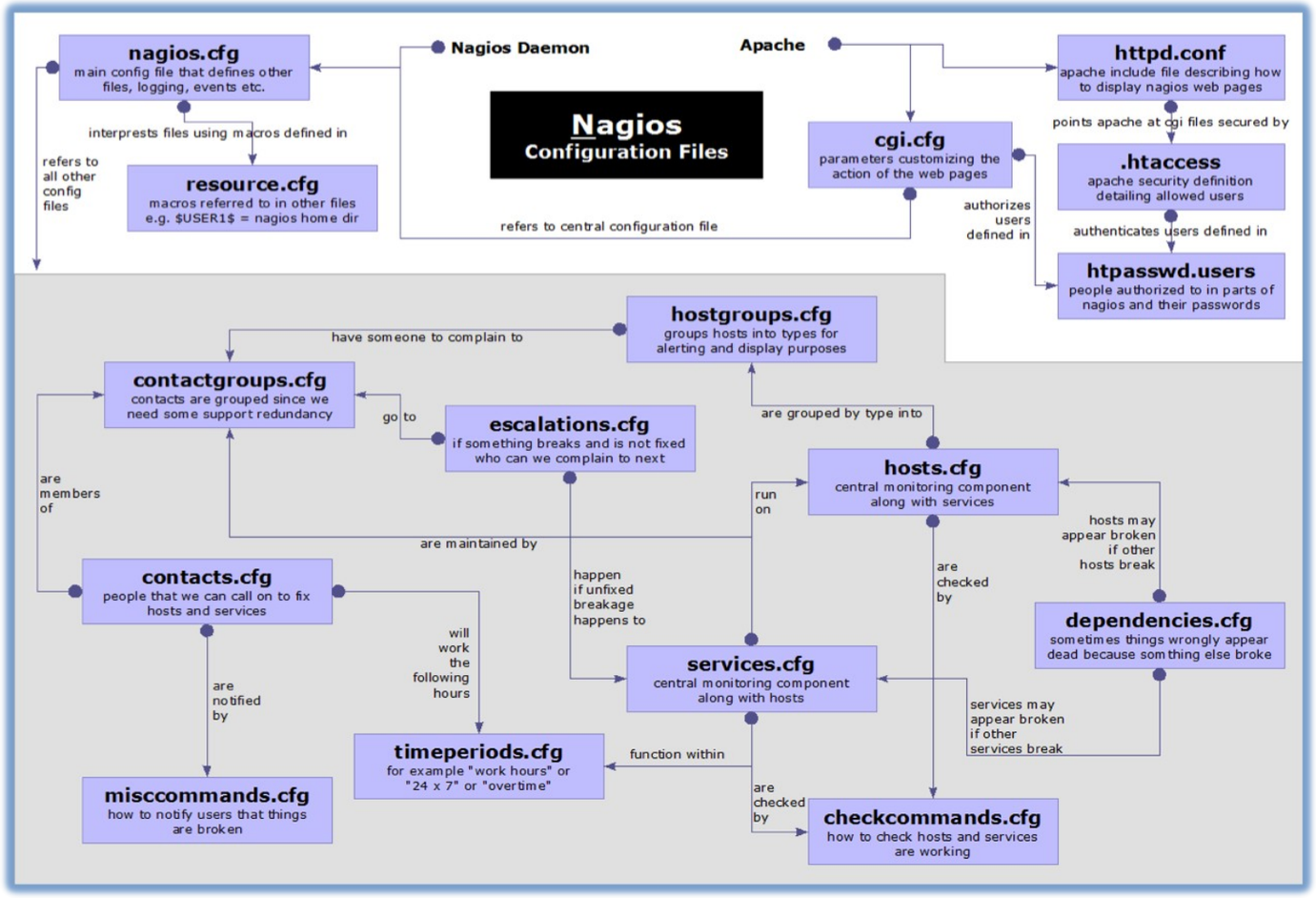
Service Notification Options

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

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

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

- 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: coreX
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             host1.campus1,HTTP,host1.campus1,SMTP,host1.campus1,POP, \
                       host1.campus1,IMAP,host1.campus1,RAID,host1.campus1,LDAP, \
                       host2.campus1,HTTP,host2.campus1,SMTP,host2.campus1,POP, \
                       host2.campus1,IMAP,host2.campus1,RAID,host2.campus1,LDAP

}
```

Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends
- Alerts
  - History
  - Summary
  - Histogram
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Current Network Status

Last Updated: Mon Feb 20 15:02:21 UTC 2017
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *nagiosadmin*

View Service Status Detail For All Host Groups
View Host Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
61	1	0	0

All Problems	All Types
1	62

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
142	0	0	60	0

All Problems	All Types
60	202

Status Summary For All Host Groups

Host Group	Host Status Summary	Service Status Summary
All Servers (all)	61 UP 1 DOWN : 1 Unhandled	142 OK 60 CRITICAL : 59 Unhandled, 1 on Problem Hosts
Access Points (aps)	2 UP	2 CRITICAL : 2 Unhandled
Cisco 3745 Routers (cisco3745)	6 UP	5 OK 7 CRITICAL : 7 Unhandled
Cisco 7200 Routers (cisco7200)	9 UP	10 OK 7 CRITICAL : 7 Unhandled
HTTP servers (http-servers)	43 UP	126 OK 42 CRITICAL : 42 Unhandled
Mac Mini (mac-servers)	1 UP	1 OK 1 CRITICAL : 1 Unhandled
NAGIOS Servers (nagios-servers)	36 UP	108 OK 36 CRITICAL : 36 Unhandled
SNMP Servers (snmp-servers)	60 UP 1 DOWN : 1 Unhandled	136 OK 60 CRITICAL : 59 Unhandled, 1 on Problem Hosts
SSH servers (ssh-servers)	58 UP	141 OK 57 CRITICAL : 57 Unhandled
Switches (switches)	1 DOWN : 1 Unhandled	1 CRITICAL : 1 on Problem Hosts
Ubuntu Linux Servers (ubuntu-servers)	43 UP	126 OK 42 CRITICAL : 42 Unhandled

Service Groups Overview

Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends
- Alerts
  - History
  - Summary
  - Histogram
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Current Network Status
Last Updated: Sun Feb 18 05:51:38 UTC 2018
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *nagiosadmin*

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
44	16	2	0

All Problems	All Types
18	62

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
43	0	0	159	0

All Problems	All Types
159	202

Service Overview For All Service Groups

Hosts accepting SNMP (host_snmp)

Host	Status	Services	Actions
host1.campus1	UP	1 CRITICAL	
host1.campus2	UP	1 CRITICAL	
host1.campus3	UP	1 CRITICAL	
host1.campus4	UP	1 CRITICAL	
host1.campus5	UP	1 CRITICAL	
host1.campus6	UP	1 CRITICAL	
host2.campus1	DOWN	1 CRITICAL	
host2.campus2	UP	1 CRITICAL	
host2.campus3	DOWN	1 CRITICAL	
host2.campus4	DOWN	1 CRITICAL	
host2.campus5	UP	1 CRITICAL	

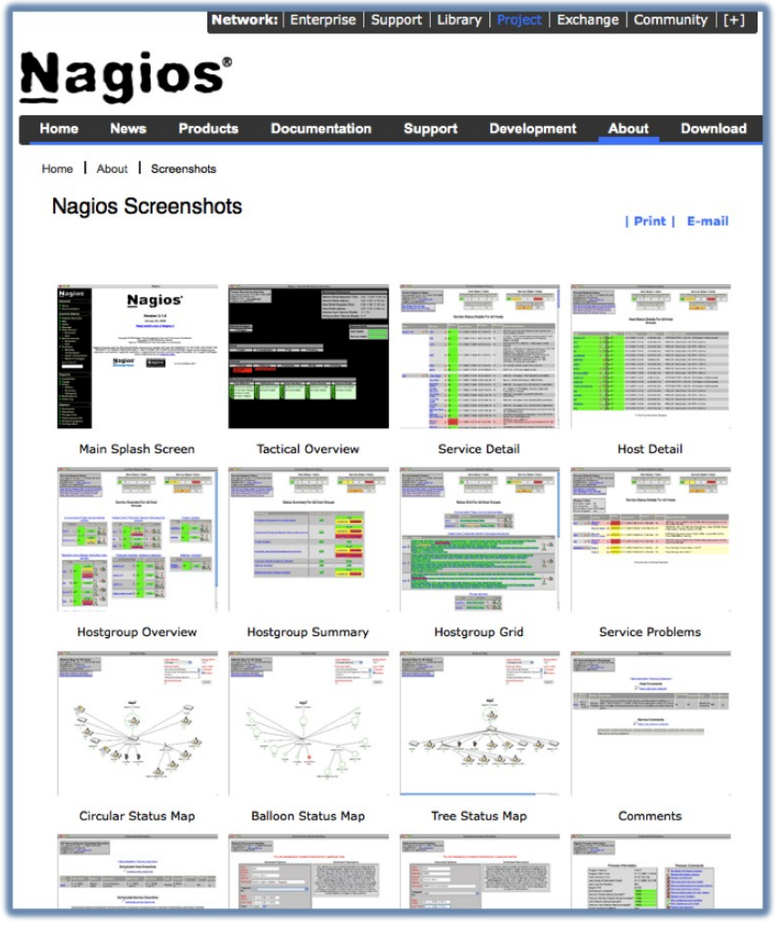
Hosts accepting SSH (host_ssh)

Host	Status	Services	Actions
host1.campus1	UP	1 OK	
host1.campus2	UP	1 OK	
host1.campus3	UP	1 OK	
host1.campus4	UP	1 OK	
host1.campus5	UP	1 OK	
host1.campus6	UP	1 OK	
host2.campus1	DOWN	1 CRITICAL	
host2.campus2	UP	1 OK	
host2.campus3	DOWN	1 CRITICAL	
host2.campus4	DOWN	1 CRITICAL	
host2.campus5	UP	1 OK	

Routers accepting SNMP (router_snmp)

Host	Status	Services	Actions
bdr1.campus1	UP	1 CRITICAL	
bdr1.campus2	UP	1 CRITICAL	
bdr1.campus3	UP	1 CRITICAL	
bdr1.campus4	UP	1 CRITICAL	
bdr1.campus5	UP	1 CRITICAL	
bdr1.campus6	UP	1 CRITICAL	
core1.campus1	UP	1 CRITICAL	
core1.campus2	UP	1 CRITICAL	
core1.campus3	UP	1 CRITICAL	
core1.campus4	UP	1 CRITICAL	
core1.campus5	UP	1 CRITICAL	

More Sample Screenshots



Many more sample Nagios screenshots available here:

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