Network Management & Monitoring

NAGIOS
Introduction

Network Monitoring Tools

- Availability
- Reliability
- Performance

Nagios actively monitors the availability of devices and services
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
Example: Service Detail view

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempt</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS-ROOT</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:43:51</td>
<td>43d 0h 55m 19s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>IP-PNS</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:41:21</td>
<td>16d 3h 57m 24s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>IP-RTR</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:43:57</td>
<td>43d 5h 35m 13s</td>
<td>1/4</td>
<td>SSH OK - Ccass-1.25 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD1</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:41:27</td>
<td>1d 0h 1m 59s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD2</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:44:04</td>
<td>1d 2h 44m 22s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD3</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:41:34</td>
<td>1d 2h 40m 58s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD4</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:44:10</td>
<td>1d 2h 44m 16s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD5</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:41:40</td>
<td>1d 2h 41m 46s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD6</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:44:17</td>
<td>1d 2h 44m 9s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD7</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:41:47</td>
<td>1d 2h 41m 39s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NOC-TLD8</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:44:23</td>
<td>1d 2h 44m 3s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NS1-TLD1</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:41:53</td>
<td>1d 0h 1m 33s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NS1-TLD2</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:44:30</td>
<td>1d 2h 43m 56s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NS1-TLD3</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:42:00</td>
<td>1d 2h 41m 26s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NS1-TLD4</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:44:36</td>
<td>1d 2h 43m 50s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NS1-TLD5</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:42:06</td>
<td>1d 2h 41m 20s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
<tr>
<td>NS1-TLD6</td>
<td>SSH</td>
<td>OK</td>
<td>2009-09-03 14:42:00</td>
<td>1d 2h 41m 26s</td>
<td>1/4</td>
<td>SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)</td>
</tr>
</tbody>
</table>
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 in Ubuntu

/usr/lib/nagios/plugins

check_apt  check_bgpstate  check_breeze  check_by_ssh  check_clamd  check_cluster  check_dhcp  check_dig  check_disk  check_disk_smb  check_dns  check_dummy
check_file_age  check_flexlm  check_jabber  checkldap  check_lldp  check_linux_raid  check_load  check_log  check_mailq  check_mrtg  check_mysql
check_mrtgstatus  check_imap  check_msql_query  check_nagios  check_ntp  check_ntp_peer  check_ntp_time  check_nwstat  check_oracle  check_overcr
check_sensors  check_pgsq1  check_ping  check_pop  check_procs  check_rctf
check_radius  check_real  check_rpc  check_rta_multi  check_rtcp_time  check_simap  check_sensors
check_tcp  check_udp  check_uss  check_users  check_wave
checkNSURLize  check_urlize  checks.sh  utility.pm  utility.sh

/etc/nagios-plugins/config

apt.cfg  disk-smb.cfg  ftp.cfg  ldap.cfg  mysql.cfg  ntp.cfg  radius.cfg  ssh.cfg
breeze.cfg  dns.cfg  hppjd.cfg  load.cfg  netware.cfg  psql.cfg  real.cfg  tcp_udp.cfg
dhcp.cfg  dummy.cfg  http.cfg  mail.cfg  news.cfg  ping.cfg  rpc-nfs.cfg  telnet.cfg
disk.cfg  flexlm.cfg  ifstatus.cfg_mrtg.cfg  nt.cfg  procs.cfg  smtp.cfg  users.cfg
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
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
Demo Nagios
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.nsre.org/nagios3/
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
Hosts 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
Monitoring a single host

pcs.cfg

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

- 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

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**

```plaintext
define host {
    host_name        pcl
    alias            pcl in group 1
    address          pcl.ws.nsnc.org
    use              generic-host
    notification_interval  120
    contact_groups    admins,managers
}
```
Defining services (direct way)

```plaintext
define host {
    host_name    pcl
    alias        pcl in group 1
    address      pcl.ws.nsre.org
    use          generic-host
}

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

define service {
    host_name    pcl
    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 template

generic-service_nagios2.cfg

define service {
    name generic-service
    active_checks_enabled 1
    passive_checks_enabled 1
    parallelize_check 1
    obsess_over_service 0
    check_freshness 1
    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 w,u,c,r
    contact_groups admins
    register 0
}
Overriding defaults

Again, settings can be overridden per service

services_nagios2.cfg

define service {
    host_name     pcl
    service_description HTTP
    check_command  check_http
    use           generic-service
    contact_groups admins,managers
    max_check_attempts 3
}
Repeated 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**

```text
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

```
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
}
```

E.g. if hostgroup “http-servers” contains pc1 and pc2 then Nagios creates HTTP service checks for both hosts. The service checks are called “pc1,HTTP” and “pc2,HTTP”
Instead of saying “this hostgroup contains these PCs” you can say “this PC belongs to these hostgroups”

No need for the “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**

```plaintext
define host {
    host_name     pcl
    alias         pcl in group 1
    address       pcl.ws.nsfc.org
    use           generic-host
    hostgroups    ssh-servers, http-servers, debian-servers
}
```

**extinfo_nagios2.cfg**

```plaintext
define hostextinfo {
    hostgroup_name   debian-servers
    notes            Debian GNU/Linux servers
    icon_image       base/debian.png
    statusmap_image  base/debian.gd2
}
```
You can also group together services 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**

```plain
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**

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

- This means “pc1 is on the far side of rtr1”
- If rtr1 goes down, pc1 is marked “unreachable” rather than “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/
References

• Nagios web site
  http://www.nagios.org/

• Nagios plugins site
  http://www.nagiosplugins.org/


• 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/
Questions?
A few additional slides you may find useful or informative...
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 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
NOTE: The flow will only continue when each of the listed filters are satisfied.
Notification Options (Host)

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
Notification Options (Service)

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 (Official)

- **nagios.cfg**: Main config file that defines other files, logging, events etc.
- **resource.cfg**: Macros referred to in other files. E.g., $USER$ = nagios home dir
- **contactgroups.cfg**: Contacts are grouped since we need some support redundancy
- **contacts.cfg**: People that we can call on to fix hosts and services
- **misccommands.cfg**: How to notify users that things are broken
- **timeperiods.cfg**: For example "work hours" or "24 x 7" or "overtimes"
- **dependencies.cfg**: Sometimes things wrongly appear dead because something else broke
- **services.cfg**: Central monitoring component along with hosts
- **hostgroups.cfg**: Groups hosts into types for alerting and display purposes
- **escalations.cfg**: If something breaks and is not fixed who can we complain to next
- **hosts.cfg**: Central monitoring component along with services
- **checkcommands.cfg**: How to check hosts and services are working
- **checkcommands.cfg**: Central monitoring component along with hosts
- **httpd.conf**: Apache includes file describing how to display nagios web pages
- **.htaccess**: Points Apache at CGI files secured by.
- **htpasswd.users**: People authorized to in parts of nagios and their passwords
- **cgi.cfg**: Parameters customizing the action of the web pages
- **auth.conf**: Authorizes users defined in.
- **authusers.cfg**: Authenticates users defined in.
Debian/Ubuntu config file layout

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!
Configuration files continued

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
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: Tuning 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
- Could adjust as needed, such as work-week only.
- Could adjust a new time period for “outside of regular hours”, etc.

```plaintext
# '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

```plaintext
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:
```
Notification commands

Allows you to utilize any command you wish. We could use this to generate tickets in RT.

```plaintext
# 'notify-by-email' command definition
define command{
    command_name    notify-by-email
    command_line    /usr/bin/printf "%b" "Service: $SERVICEDESC$
                    Host: $HOSTNAME$
                    In: $HOSTALIAS$
                    Address: $HOSTADDRESS$
                    State: $SERVICESTATE$
                    Info: $SERVICEOUTPUT$
                    Date: $SHORTDATETIME$"
                    | /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-rtr
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
}

A few sample screen shots from a Nagios install.
### Host Status Details For All Host Groups

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR1-RTC</td>
<td>UP</td>
<td>2009-09-03 14:51:41</td>
<td>43d 1h 7m 0s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.33 ms</td>
</tr>
<tr>
<td>RP-DIM</td>
<td>UP</td>
<td>2009-09-03 14:51:41</td>
<td>16d 4h 11m 25s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.29 ms</td>
</tr>
<tr>
<td>RP-RTB</td>
<td>UP</td>
<td>2009-09-03 14:51:51</td>
<td>43d 5h 47m 40s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.24 ms</td>
</tr>
<tr>
<td>NOC-LEC</td>
<td>UP</td>
<td>2009-09-03 14:52:01</td>
<td>1d 6h 10m 56s</td>
<td>PING OK - Packet loss = 0%, RTA = 4.02 ms</td>
</tr>
<tr>
<td>NOC-T6C</td>
<td>UP</td>
<td>2009-09-03 14:52:01</td>
<td>1d 22h 53m 48s</td>
<td>PING OK - Packet loss = 0%, RTA = 2.23 ms</td>
</tr>
<tr>
<td>NOC-T6D</td>
<td>UP</td>
<td>2009-09-03 14:52:11</td>
<td>1d 22h 53m 38s</td>
<td>PING OK - Packet loss = 0%, RTA = 2.62 ms</td>
</tr>
<tr>
<td>NOC-T6E</td>
<td>UP</td>
<td>2009-09-03 14:52:21</td>
<td>1d 22h 53m 38s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.09 ms</td>
</tr>
<tr>
<td>NOC-T6F</td>
<td>UP</td>
<td>2009-09-03 14:52:21</td>
<td>1d 22h 54m 6s</td>
<td>PING OK - Packet loss = 0%, RTA = 5.20 ms</td>
</tr>
<tr>
<td>NOC-T6G</td>
<td>UP</td>
<td>2009-09-03 14:52:31</td>
<td>1d 22h 53m 56s</td>
<td>PING OK - Packet loss = 0%, RTA = 10.49 ms</td>
</tr>
<tr>
<td>NOC-T6H</td>
<td>UP</td>
<td>2009-09-03 14:52:41</td>
<td>1d 22h 53m 56s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.05 ms</td>
</tr>
<tr>
<td>NOC-T6I</td>
<td>UP</td>
<td>2009-09-03 14:52:51</td>
<td>1d 22h 53m 56s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.00 ms</td>
</tr>
<tr>
<td>NOC-T6J</td>
<td>UP</td>
<td>2009-09-03 14:53:01</td>
<td>1d 0h 10m 26s</td>
<td>PING OK - Packet loss = 0%, RTA = 10.19 ms</td>
</tr>
<tr>
<td>NOC-T6K</td>
<td>UP</td>
<td>2009-09-03 14:53:01</td>
<td>1d 22h 53m 56s</td>
<td>PING OK - Packet loss = 0%, RTA = 9.06 ms</td>
</tr>
<tr>
<td>NOC-T6L</td>
<td>UP</td>
<td>2009-09-03 14:53:11</td>
<td>1d 22h 53m 36s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.03 ms</td>
</tr>
<tr>
<td>NS1-T6B</td>
<td>UP</td>
<td>2009-09-03 14:53:21</td>
<td>1d 22h 53m 36s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.15 ms</td>
</tr>
<tr>
<td>NS1-T6C</td>
<td>UP</td>
<td>2009-09-03 14:53:21</td>
<td>1d 22h 54m 6s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.12 ms</td>
</tr>
<tr>
<td>NS1-T6D</td>
<td>UP</td>
<td>2009-09-03 14:53:31</td>
<td>1d 22h 53m 56s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.06 ms</td>
</tr>
<tr>
<td>NS1-T6E</td>
<td>UP</td>
<td>2009-09-03 14:53:41</td>
<td>1d 22h 53m 46s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.11 ms</td>
</tr>
<tr>
<td>NS1-T6F</td>
<td>UP</td>
<td>2009-09-03 14:53:51</td>
<td>1d 22h 53m 36s</td>
<td>PING OK - Packet loss = 0%, RTA = 1.18 ms</td>
</tr>
<tr>
<td>NS1-T6G</td>
<td>UP</td>
<td>2009-09-03 14:53:51</td>
<td>1d 22h 54m 6s</td>
<td>PING OK - Packet loss = 0%, RTA = 2.22 ms</td>
</tr>
<tr>
<td>NS1-T6H</td>
<td>UP</td>
<td>2009-09-03 14:54:01</td>
<td>1d 22h 53m 48s</td>
<td>PING OK - Packet loss = 0%, RTA = 2.38 ms</td>
</tr>
</tbody>
</table>
### Host Groups Overview

#### Service Overview For All Host Groups

**TRTI TLD1 Servers, Virtual Machines, Routers (TLD1)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD1</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD1</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD1-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD1</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD2 Servers, Virtual Machines, Routers (TLD2)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD2</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD2</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD2-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD2</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD3 Servers, Virtual Machines, Routers (TLD3)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD3</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD3</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD3-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD3</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD4 Servers, Virtual Machines, Routers (TLD4)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD4</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD4</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD4-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD4</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD5 Servers, Virtual Machines, Routers (TLD5)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD5</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD5</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD5-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD5</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD6 Servers, Virtual Machines, Routers (TLD6)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD6</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD6</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD6-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD6</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD7 Servers, Virtual Machines, Routers (TLD7)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD7</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD7</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD7-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD7</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI TLD8 Servers, Virtual Machines, Routers (TLD8)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC-TLD8</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD8</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD8-RTS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TRTI-TLD8</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>

**TRTI Management Virtual Machines (VM-mgmt)**

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS-ROOT</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>IP-CNS</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>
Service Groups Overview

The image shows a screenshot of a Nagios monitoring interface focusing on service groups. It includes a table with columns for Host, Status, Services, and Actions, detailing the status of different hosts within the network. The interface provides a visual overview of network status, listing hosts that are up and those with critical statuses.

### Current Network Status
- Last Updated: Fri Sep 4 13:32:20 CDT 2009
- Updated every 90 seconds
- Nagios® 3.0.2 - www.nagios.org
- Logged in as guest

### Host Status Totals
- Up: 41, Down: 0, Unreachable: 0, Pending: 0
- Critical: 1, Warning: 0, Unknown: 0

### Service Status Totals
- All Problems: 1, All Types: 54

### Service Overview For All Service Groups

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Services</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLD Servers running Nagios (NAGIOS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS1-TLD1</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD2</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD3</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD4</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD5</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD6</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD7</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD8</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>TLD Servers running SSH (SSH)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS1-TLD1</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD2</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD3</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD4</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD5</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD6</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD7</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
<tr>
<td>NS1-TLD8</td>
<td>UP</td>
<td>1.0K</td>
<td></td>
</tr>
</tbody>
</table>
Collapsed tree status map
Marked-up circular status map
More sample screenshots

Many more sample Nagios screenshots available here:

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