# Network Monitoring, Management and Automation

# Nagios®

## npNOG 5

Dec 8 - 12, 2019

# Introduction

Nagios® Core™

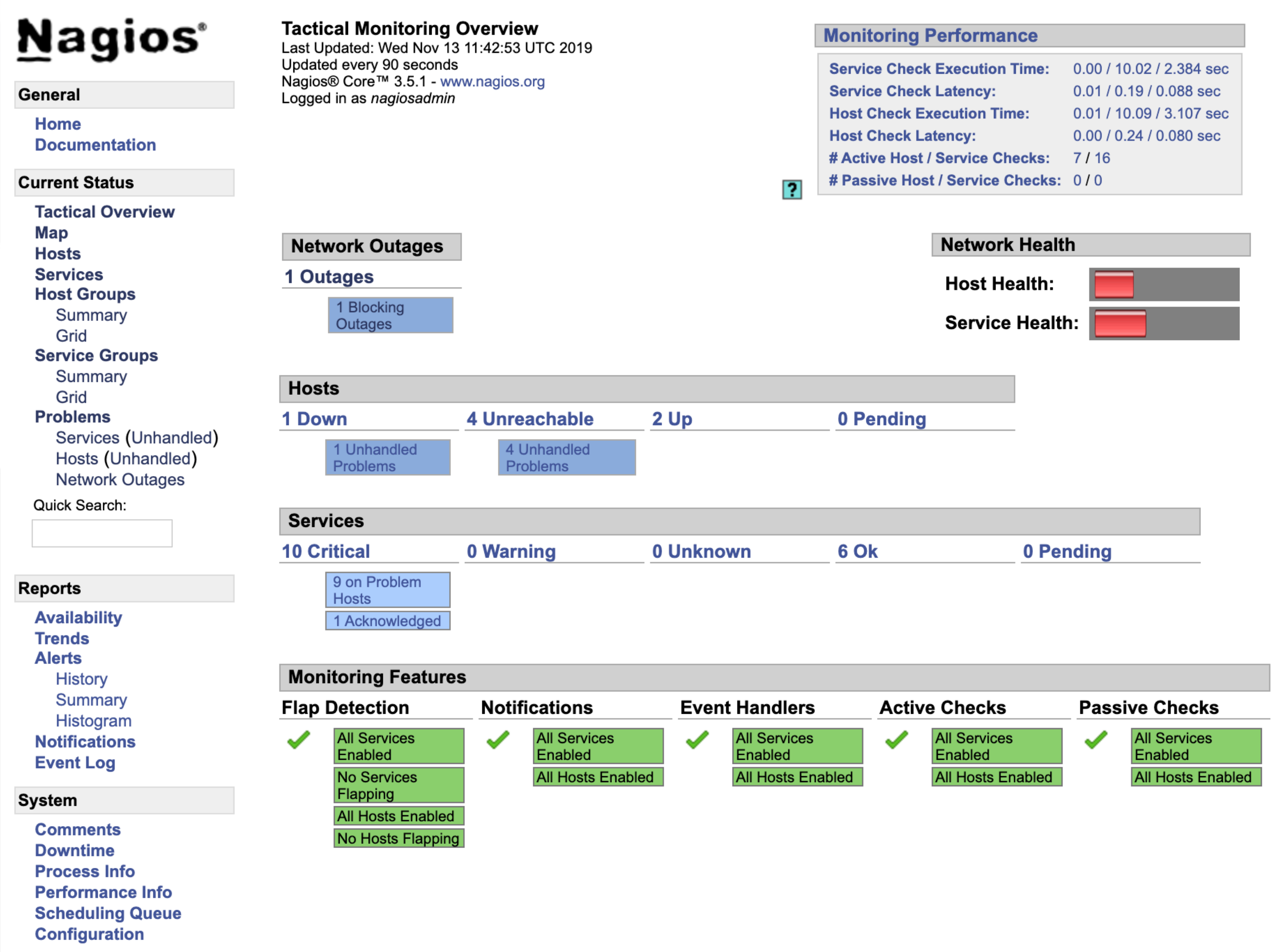
- Possibly the most used open source network monitoring software
- Web interface for viewing status, browsing history, scheduling downtime etc
- Sends out alerts via E-mail. Can be configured to use other mechanisms, e.g. SMS

# Introduction (Contd...)

Nagios actively monitors the ***availability*** of

- Hosts (devices)
- Services

# Nagios: Tactical 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

**Tactical Monitoring Overview**

Last Updated: Wed Nov 13 11:42:53 UTC 2019
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *nagiosadmin*

**Monitoring Performance**

| | |
|---|---|
| Service Check Execution Time: | 0.00 / 10.02 / 2.384 sec |
| Service Check Latency: | 0.01 / 0.19 / 0.088 sec |
| Host Check Execution Time: | 0.01 / 10.09 / 3.107 sec |
| Host Check Latency: | 0.00 / 0.24 / 0.080 sec |
| # Active Host / Service Checks: | 7 / 16 |
| # Passive Host / Service Checks: | 0 / 0 |

**Network Outages**

1 Outages

1 Blocking Outages

**Network Health**

Host Health:

Service Health:

**Hosts**

| 1 Down | 4 Unreachable | 2 Up | 0 Pending |
|---|---|---|---|
| 1 Unhandled Problems | 4 Unhandled Problems | | |

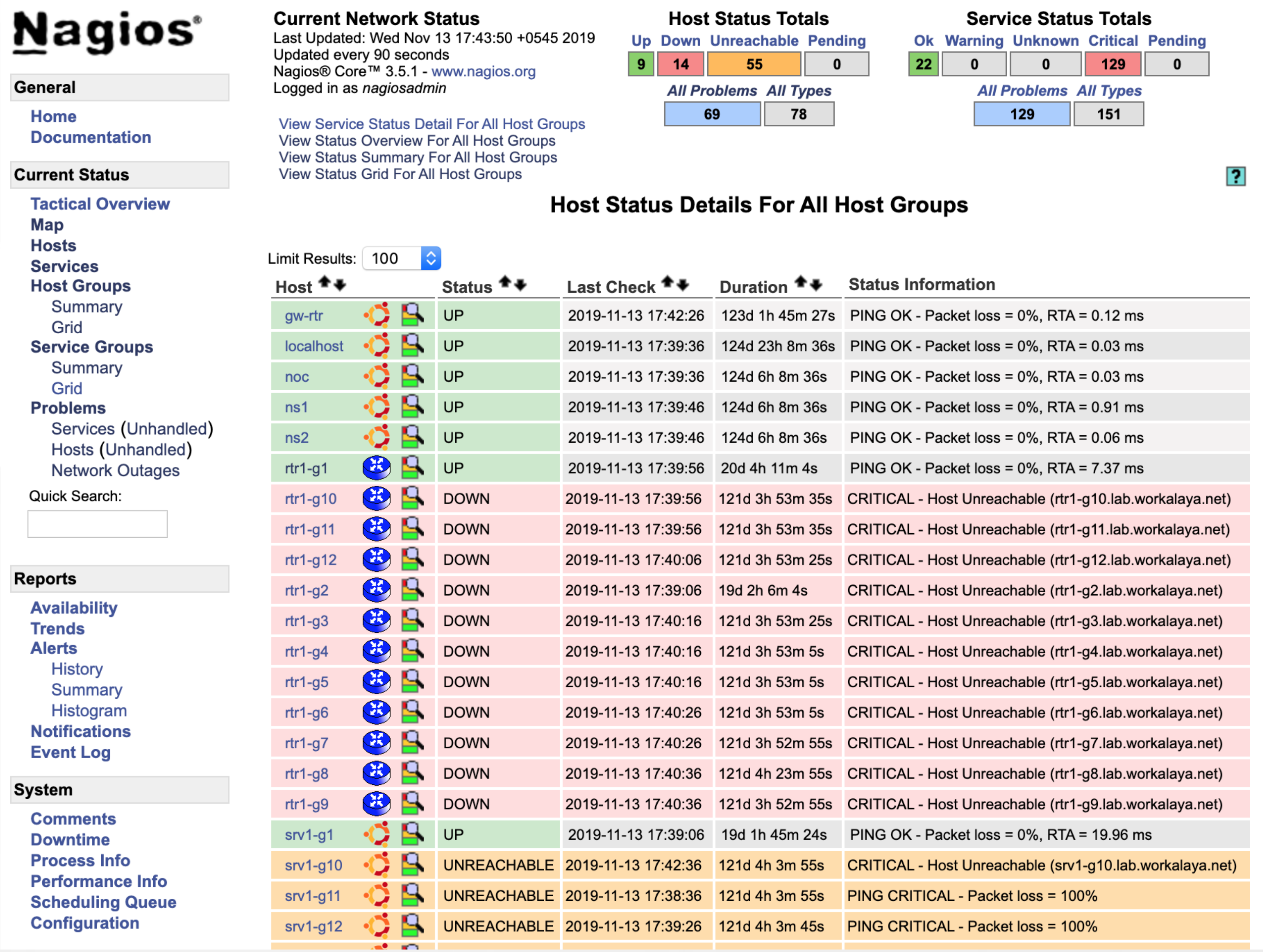
**Services**

| 10 Critical | 0 Warning | 0 Unknown | 6 Ok | 0 Pending |
|---|---|---|---|---|
| 9 on Problem Hosts<br>1 Acknowledged | | | | |

**Monitoring Features**

| Flap Detection | Notifications | Event Handlers | Active Checks | Passive Checks |
|---|---|---|---|---|
| ✓ All Services Enabled<br>No Services Flapping<br>All Hosts Enabled<br>No Hosts Flapping | ✓ All Services Enabled<br>All Hosts Enabled | ✓ All Services Enabled<br>All Hosts Enabled | ✓ All Services Enabled<br>All Hosts Enabled | ✓ All Services Enabled<br>All Hosts Enabled |

# Nagios: 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: Wed Nov 13 17:43:50 +0545 2019
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 |
|---|---|---|---|
| 9 | 14 | 55 | 0 |

| *All Problems* | *All Types* |
|---|---|
| 69 | 78 |

**Service Status Totals**

| Ok | Warning | Unknown | Critical | Pending |
|---|---|---|---|---|
| 22 | 0 | 0 | 129 | 0 |

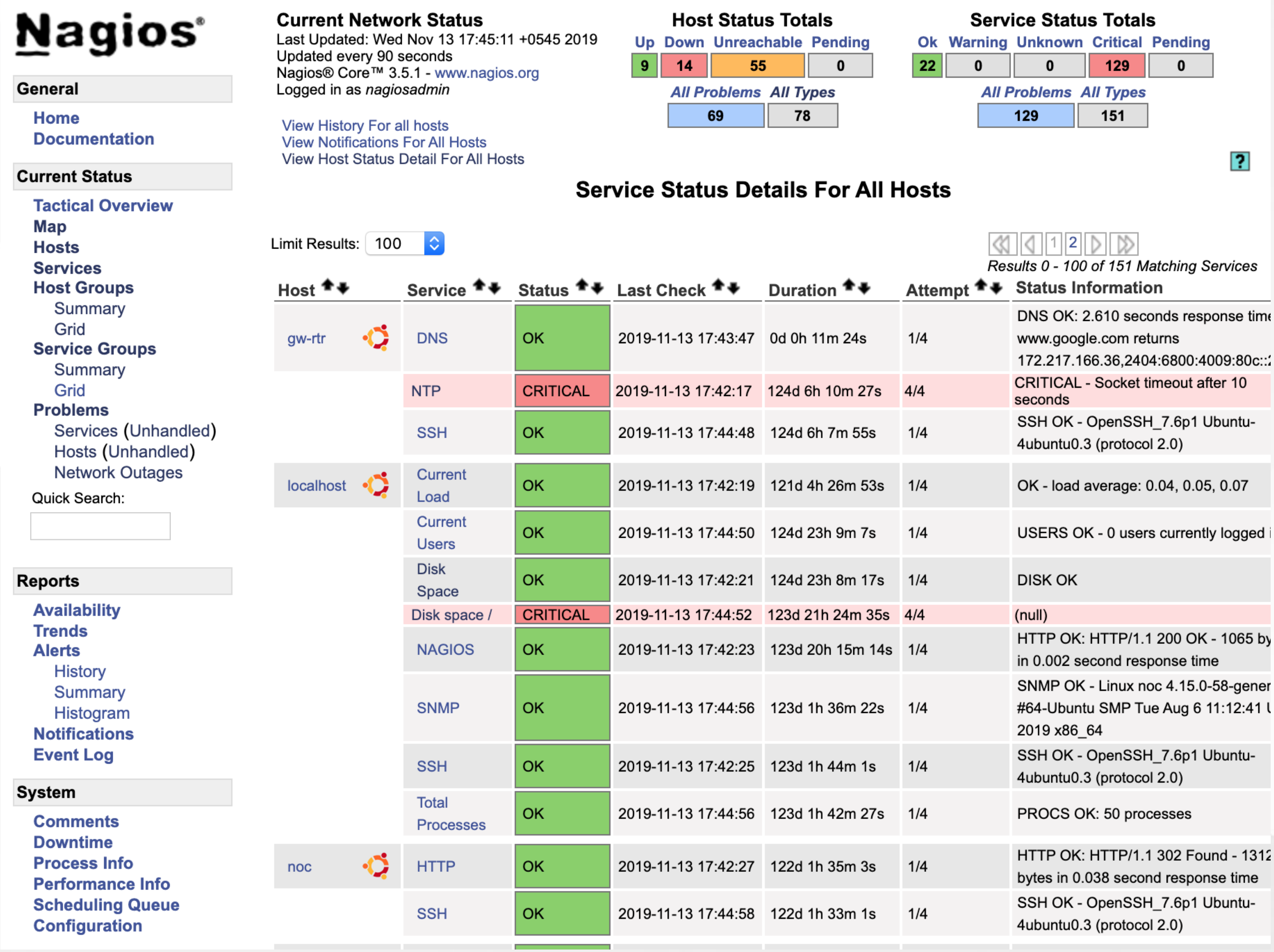
| *All Problems* | *All Types* |
|---|---|
| 129 | 151 |

## Host Status Details For All Host Groups

Limit Results: 100

| Host | Status | Last Check | Duration | Status Information |
|---|---|---|---|---|
| gw-rtr | UP | 2019-11-13 17:42:26 | 123d 1h 45m 27s | PING OK - Packet loss = 0%, RTA = 0.12 ms |
| localhost | UP | 2019-11-13 17:39:36 | 124d 23h 8m 36s | PING OK - Packet loss = 0%, RTA = 0.03 ms |
| noc | UP | 2019-11-13 17:39:36 | 124d 6h 8m 36s | PING OK - Packet loss = 0%, RTA = 0.03 ms |
| ns1 | UP | 2019-11-13 17:39:46 | 124d 6h 8m 36s | PING OK - Packet loss = 0%, RTA = 0.91 ms |
| ns2 | UP | 2019-11-13 17:39:46 | 124d 6h 8m 36s | PING OK - Packet loss = 0%, RTA = 0.06 ms |
| rtr1-g1 | UP | 2019-11-13 17:39:56 | 20d 4h 11m 4s | PING OK - Packet loss = 0%, RTA = 7.37 ms |
| rtr1-g10 | DOWN | 2019-11-13 17:39:56 | 121d 3h 53m 35s | CRITICAL - Host Unreachable (rtr1-g10.lab.workalaya.net) |
| rtr1-g11 | DOWN | 2019-11-13 17:39:56 | 121d 3h 53m 35s | CRITICAL - Host Unreachable (rtr1-g11.lab.workalaya.net) |
| rtr1-g12 | DOWN | 2019-11-13 17:40:06 | 121d 3h 53m 25s | CRITICAL - Host Unreachable (rtr1-g12.lab.workalaya.net) |
| rtr1-g2 | DOWN | 2019-11-13 17:39:06 | 19d 2h 6m 4s | CRITICAL - Host Unreachable (rtr1-g2.lab.workalaya.net) |
| rtr1-g3 | DOWN | 2019-11-13 17:40:16 | 121d 3h 53m 25s | CRITICAL - Host Unreachable (rtr1-g3.lab.workalaya.net) |
| rtr1-g4 | DOWN | 2019-11-13 17:40:16 | 121d 3h 53m 5s | CRITICAL - Host Unreachable (rtr1-g4.lab.workalaya.net) |
| rtr1-g5 | DOWN | 2019-11-13 17:40:16 | 121d 3h 53m 5s | CRITICAL - Host Unreachable (rtr1-g5.lab.workalaya.net) |
| rtr1-g6 | DOWN | 2019-11-13 17:40:26 | 121d 3h 53m 5s | CRITICAL - Host Unreachable (rtr1-g6.lab.workalaya.net) |
| rtr1-g7 | DOWN | 2019-11-13 17:40:26 | 121d 3h 52m 55s | CRITICAL - Host Unreachable (rtr1-g7.lab.workalaya.net) |
| rtr1-g8 | DOWN | 2019-11-13 17:40:36 | 121d 4h 23m 55s | CRITICAL - Host Unreachable (rtr1-g8.lab.workalaya.net) |
| rtr1-g9 | DOWN | 2019-11-13 17:40:36 | 121d 3h 52m 55s | CRITICAL - Host Unreachable (rtr1-g9.lab.workalaya.net) |
| srv1-g1 | UP | 2019-11-13 17:39:06 | 19d 1h 45m 24s | PING OK - Packet loss = 0%, RTA = 19.96 ms |
| srv1-g10 | UNREACHABLE | 2019-11-13 17:42:36 | 121d 4h 3m 55s | CRITICAL - Host Unreachable (srv1-g10.lab.workalaya.net) |
| srv1-g11 | UNREACHABLE | 2019-11-13 17:38:36 | 121d 4h 3m 55s | PING CRITICAL - Packet loss = 100% |
| srv1-g12 | UNREACHABLE | 2019-11-13 17:39:26 | 121d 4h 3m 45s | PING CRITICAL - Packet loss = 100% |

# Nagios: Service Detail View



**Current Network Status**

Last Updated: Wed Nov 13 17:45:11 +0545 2019
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *nagiosadmin*

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

**Host Status Totals**

| Up | Down | Unreachable | Pending |
|---|---|---|---|
| 9 | 14 | 55 | 0 |

| *All Problems* | *All Types* |
|---|---|
| 69 | 78 |

**Service Status Totals**

| Ok | Warning | Unknown | Critical | Pending |
|---|---|---|---|---|
| 22 | 0 | 0 | 129 | 0 |

| *All Problems* | *All Types* |
|---|---|
| 129 | 151 |

**Service Status Details For All Hosts**

Limit Results: 100

1 2

*Results 0 - 100 of 151 Matching Services*

| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|---|---|---|---|---|---|---|
| gw-rtr | DNS | OK | 2019-11-13 17:43:47 | 0d 0h 11m 24s | 1/4 | DNS OK: 2.610 seconds response time www.google.com returns 172.217.166.36,2404:6800:4009:80c::2 |
| | NTP | CRITICAL | 2019-11-13 17:42:17 | 124d 6h 10m 27s | 4/4 | CRITICAL - Socket timeout after 10 seconds |
| | SSH | OK | 2019-11-13 17:44:48 | 124d 6h 7m 55s | 1/4 | SSH OK - OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 (protocol 2.0) |
| localhost | Current Load | OK | 2019-11-13 17:42:19 | 121d 4h 26m 53s | 1/4 | OK - load average: 0.04, 0.05, 0.07 |
| | Current Users | OK | 2019-11-13 17:44:50 | 124d 23h 9m 7s | 1/4 | USERS OK - 0 users currently logged i |
| | Disk Space | OK | 2019-11-13 17:42:21 | 124d 23h 8m 17s | 1/4 | DISK OK |
| | Disk space / | CRITICAL | 2019-11-13 17:44:52 | 123d 21h 24m 35s | 4/4 | (null) |
| | NAGIOS | OK | 2019-11-13 17:42:23 | 123d 20h 15m 14s | 1/4 | HTTP OK: HTTP/1.1 200 OK - 1065 by in 0.002 second response time |
| | SNMP | OK | 2019-11-13 17:44:56 | 123d 1h 36m 22s | 1/4 | SNMP OK - Linux noc 4.15.0-58-gener #64-Ubuntu SMP Tue Aug 6 11:12:41 U 2019 x86_64 |
| | SSH | OK | 2019-11-13 17:42:25 | 123d 1h 44m 1s | 1/4 | SSH OK - OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 (protocol 2.0) |
| | Total Processes | OK | 2019-11-13 17:44:56 | 123d 1h 42m 27s | 1/4 | PROCS OK: 50 processes |
| noc | HTTP | OK | 2019-11-13 17:42:27 | 122d 1h 35m 3s | 1/4 | HTTP OK: HTTP/1.1 302 Found - 1312 bytes in 0.038 second response time |
| | SSH | OK | 2019-11-13 17:44:58 | 122d 1h 33m 1s | 1/4 | SSH OK - OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 (protocol 2.0) |

# 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

```
check_apt       check_file_age      check_imap         check_nagios    check_pop        check_swap
check_breeze    check_flexlm        check_ircd         check_nntp      check_procs      check_tcp
check_by_ssh    check_fping         check_jabber       check_nntps     check_real       check_time
check_clamd     check_ftp           check_ldap         check_nt        check_rpc        check_udp
check_cluster   check_game          check_ldaps        check_ntp       check_rta_multi  check_ups
check_dbi       check_host          check_load         check_ntp_peer  check_sensors    check_users
check_dhcp      check_hpjd          check_log          check_ntp_time  check_simap      check_wave
check_dig       check_http          check_mailq        check_nwstat    check_smtp       negate
check_disk      check_icmp          check_mrtg         check_oracle    check_snmp       urlize
check_disk_smb  check_ide_smart     check_mrtgtraf     check_overcr    check_spop       utils.pm
check_dns       check_ifoperstatus  check_mysql        check_pgsql     check_ssh        utils.sh
check_dummy     check_ifstatus      check_mysql_query  check_ping      check_ssmtp
```

## /usr/lib/nagios/plugins

```
apt.cfg        dns.cfg      games.cfg      load.cfg    netware.cfg  ping.cfg     ssh.cfg
breeze.cfg     dummy.cfg    hppjd.cfg      mail.cfg    news.cfg     procs.cfg    tcp_udp.cfg
dhcp.cfg       flexlm.cfg   http.cfg       mailq.cfg   nt.cfg       real.cfg     telnet.cfg
disk-smb.cfg   fping.cfg    ifstatus.cfg   mrtg.cfg    ntp.cfg      rpc-nfs.cfg  users.cfg
disk.cfg       ftp.cfg      ldap.cfg       mysql.cfg   pgsql.cfg    snmp.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

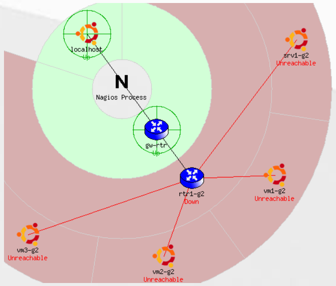
# How Checks Work (Continued)

- **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

# Hierarchy: The Concept of Parents

Hosts can have parents:

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



# More complex YAML example

```
A list with 3 items
|
|  each item is a hash (key-value pairs)
|  |
V  V
- do: laundary  <-- simple value
  item:
    - shirts    <-- list value (note indentation)
    - trousers
- do: shopping
  item:
    - bread
    - jam
- do: relax
  eat:
    - chips
    - fruits
```

# Ansible Playbook

```
Top level: a list of "plays"
|  Each play has "hosts" plus "tasks" and/or "roles"
|  |
V  V
- hosts:
    - vm1-g1.lab.workalaya.net
    - vm2-g2.lab.workalaya.net
  tasks:
    - name: install Apache
      action: package name=apache2 state=present
    - name: ensure Apache is started
      action: service name=apache2 state=started
- hosts: dns_servers
  roles:
    - dns_server
    - ntp
```

# Ansible Roles

- A bundle of related tasks/handlers/templates

**roles/*<rolename>*/tasks/main.yml**
**roles/*<rolename>*/handlers/main.yml**
**roles/*<rolename>*/defaults/main.yml**
**roles/*<rolename>*/files/...**
**roles/*<rolename>*/templates/...**

- Recommended way to make re-usable configs
- Not all these files need to be present

# Ansible Tags

- Each role or individual task can be labelled with one or more "tags"
- When you run a playbook, you can tell it only to run tasks with a particular tag: -t <tag>
- Lets you selectively run parts of playbooks

# Ansible Inventory

- Lists all hosts which Ansible may manage
- Simple "INI" format, not YAML
- Can define groups of hosts
- Default is /etc/ansible/hosts
  - Can override using -i <filename>

# Inventory (hosts) example

```
[dns_servers]          <-- Name of group
ns1.lab.workalaya.net  <-- Hosts in this group
ns2.lab.workalaya.net

[vms]
vm1-g1.lab.workalaya.net
vm1-g1.lab.workalaya.net

[nagios_server]
noc.lab.workalaya.net
vm1-g1.lab.workalaya.net
vm1-g1.lab.workalaya.net
```

```
Note:
- the same host can be listed under multiple groups.
- Group "all" is created automatically
```

# Inventory variables

- You can set variables on hosts or groups of hosts
- Variables can make tasks behave differently when applied to different hosts
- Variables can be inserted into templates
- Some variables control how Ansible connects

# Setting host vars

- Directly in the inventory (hosts) file

```
[core_servers]
ns1.lab.workalaya.net ansible_connection=local
ns2.lab.workalaya.net
```

- In file host_vars/pc2.example.com

```
ansible_ssh_host: 10.10.0.241
ansible_ssh_user: root
flurble:
  - foo
  - bar
```

  - This is in YAML and is preferred

# Setting group vars

- **group_vars/dns_servers**

```
# More YAML
flurble:
  - foo-foo
  - bar-foo
```

- **group_vars/all**

```
# More YAML, applies to every host
ansible_ssh_user: lab
ansible_beccome_pass: yourpass
```

```
Note: host vars take priority over group vars
```

# Ansible Facts

- Facts are variables containing information collected automatically about the target host
- Things like what OS is installed, what interfaces it has, what disk drives it has
- Can be used to adapt roles automatically to the target system
- Gathered every time Ansible connects to a host (unless playbook has "gather_facts: no")

# Showing facts

```
~$ ansible vmX-gY.lab.workalaya.net -m setup | less

vmX-gY.lab.workalaya.net | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "100.68.X.21"
        ],
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "12/12/2018",
        "ansible_bios_version": "6.00",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-4.15.0-58-generic",
            "ro": true,
            "root": "/dev/mapper/lab--main--vg-root"
        },
        "ansible_date_time": {
            "date": "2019-11-13",
            "day": "13",
            "epoch": "1573634010",
```

# jinja2 template examples

- Insert a variable into text

```
INTERFACES="{{ dhcp_interfaces }}"
```

- Looping over lists

```
search lab.workalaya.net
{% for host in dns_servers %}
nameserver {{ host }}
{% endfor %}
```

# Many other cool features

- conditionals

```
- action: package name=apache2 state=present
  when: ansible_os_family=='Debian'
```

- Loops

```
- action: package name={{item}} state=present
  with_items:
    - openssh-server
    - rsync
    - telnet
```

# Getting up-to-date Ansible

- Your package manager's version may be old
- For Ubuntu LTS: use the PPA

```
apt-get install python-software-properties
add-apt-repository ppa:rquillo/ansible
apt-get update
apt-get install ansible
```

- or, if using python venv

```
(venv) vmX-gY@ansible-gY:~/ansible-playbook$ pip install --upgrade ansible
```

# More info and documentation

- https://docs.ansible.com/
- https://jinja.palletsprojects.com/
- https://yaml.org/

?