Сетевое управление и мониторинг

NAGIOS

These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license (http://creativecommons.org/licenses/by-nc/3.0/)

Введение

Инструменты сетевого мониторинга

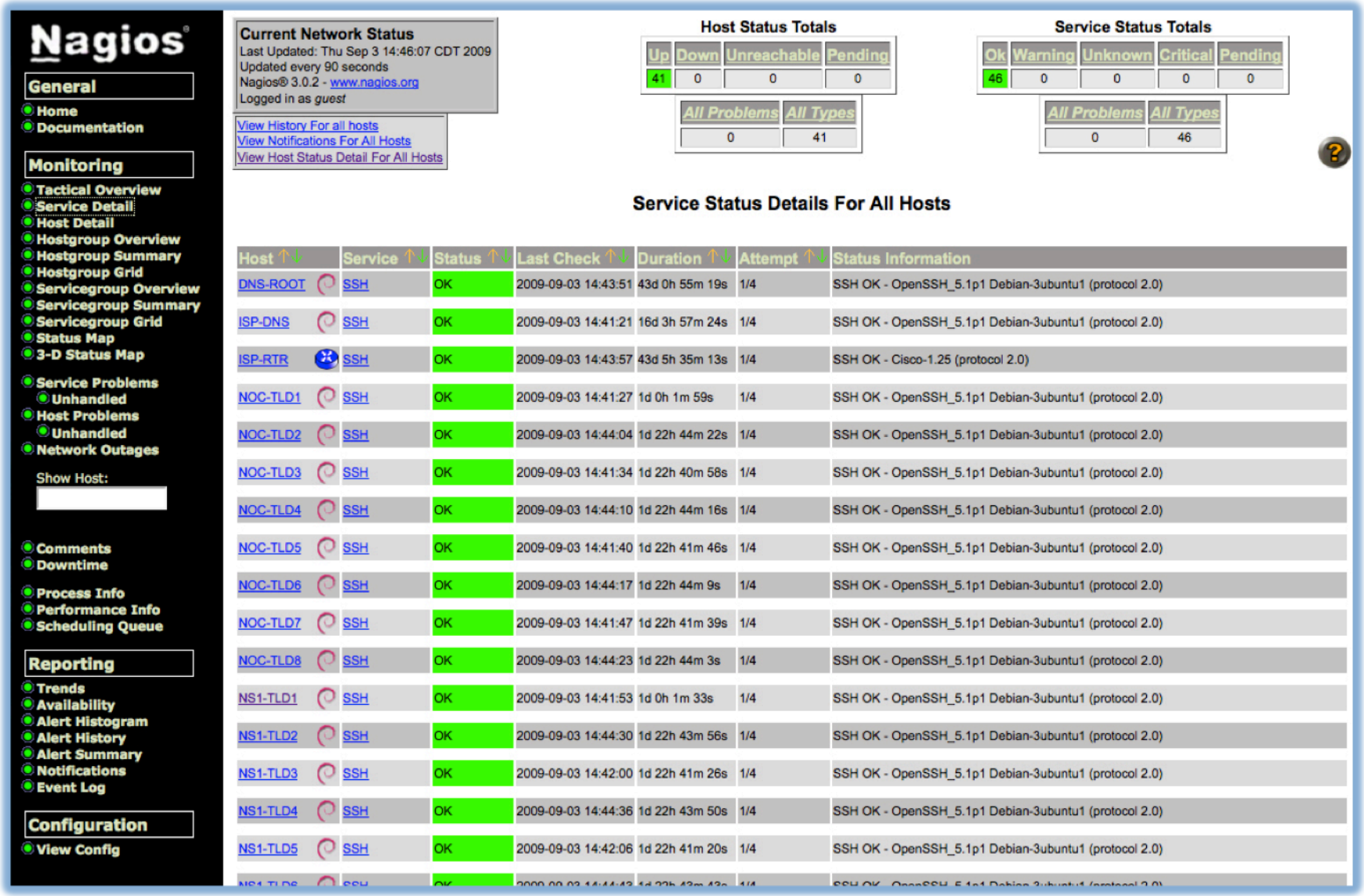
- Доступность
- Надежность
- Производительность

Nagios активно мониторит доступность устройств и сервисов

Введение

- Вероятно, наиболее часто используемая программа сетевого мониторинга с открытыми исходными текстами
- Web-интерфейс для просмотра состояний, истории, планирования техобслуживания и т.д.
- Отправляет предупреждения по электронной почте. Может быть настроен использовать другие механизмы, например SMS

Пример: страница деталей сервиса



Возможности

Использует топологию для определения зависимостей

- Понимает разницу между тем, что не *работает*, и тем, что *недоступно*. Не выполняет ненужные проверки и не шлет излишние предупреждения

Позволяет задавать, как отправлять сообщения, основываясь на комбинации:

- Контактов и списков контактов
- Устройств и групп устройств
- Сервисов и групп сервисов
- Определенных часов для людей и групп
- Состоянии сервиса

Плагины

Плагины используются для проверки сервисов и устройств:

- Архитектура Nagios достаточно проста, поэтому создание новых плагинов на любом языке не является сложной задачей.
- Существуют ***тысячи*** плагинов.
  - ✓ http://exchange.nagios.org/
  - ✓ http://nagiosplugins.org/

Предустановленные плагины в Ubuntu

/usr/lib/nagios/plugins

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

/etc/nagios-plugins/config

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

Как работают проверки

- Время от времени Nagios выполняет плагин и проверяет состояние каждого сервиса. Возможные ответы:
  - OK
  - WARNING
  - CRITICAL
  - UNKNOWN
- Если сервис не OK, ему назначается состояние “мягкой” ошибки. После нескольких повторений (3-х по умолчанию) он переходит в состояние “жесткой” ошибки. В этот момент отсылается предупреждение.
- Вы также можете выполнять внешние обработчики событий при изменениях состояния

Как работают проверки, прод.

Параметры

- Обычный интервал для проверок
- Интервал повторов (когда не OK)
- Максимальное количество повторов
- Когда производить проверки
- Когда отсылать сообщения

Планирование проверок

- Nagios “размазывает” проверки по времени, чтобы избежать пиковых нагрузок
- Web-интерфейс показывает, когда произойдет следующая проверка

Понятие ”родителей”

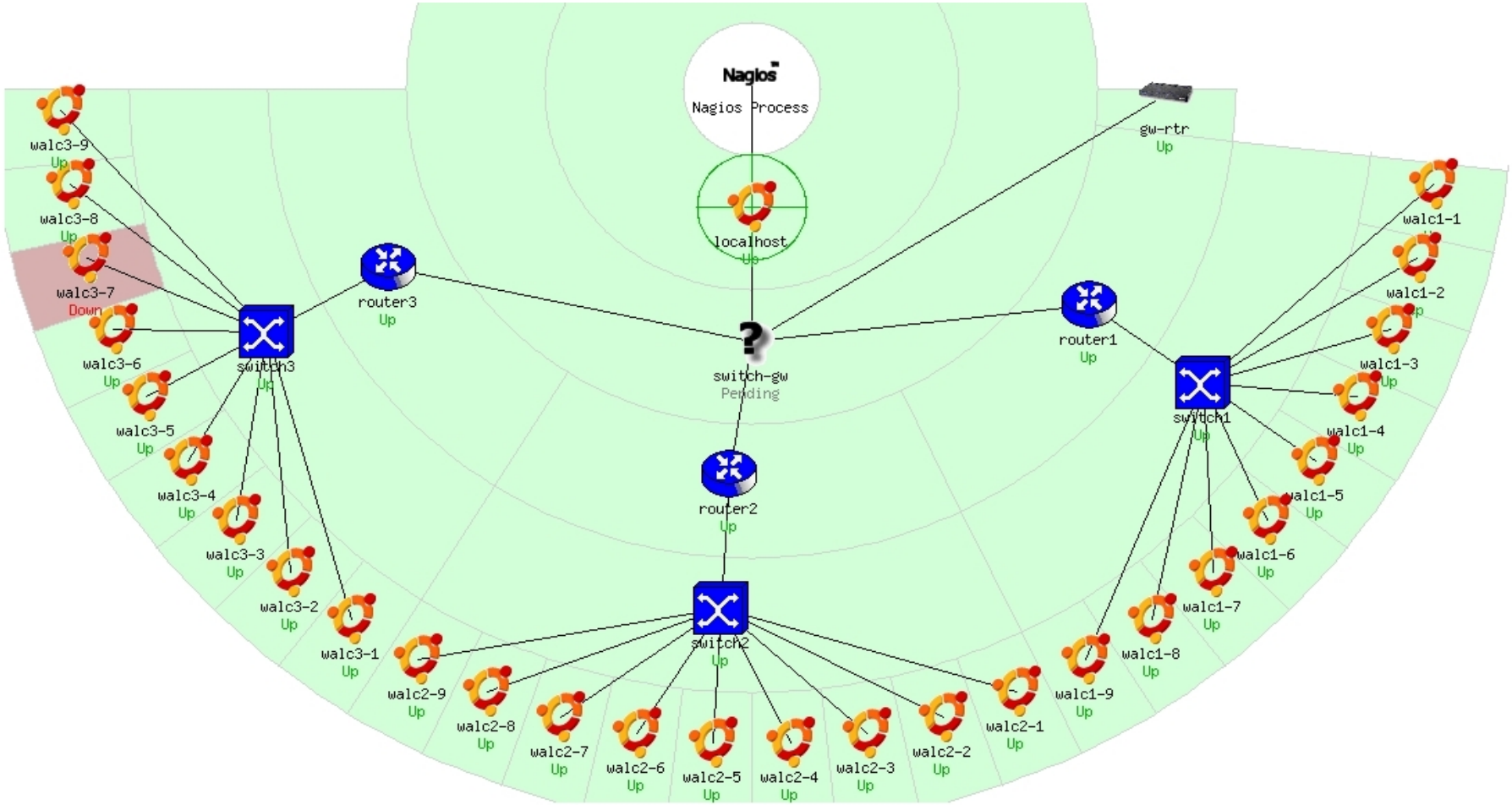
У хостов могут быть “родители”:

- Родитель PC, подключенной к маршрутизатору будет этот маршрутизатор.
- Дает нам задавать зависимости между устройствами.
- Не посылает предупреждения если “родитель” не отвечает.
- Узел может иметь несколько “родителей”.

Сетевая точка зрения

- Местоположение сервера Nagios определяет точку зрения на сеть.
- Сервер Nagios становится “корнем” дерева зависимостей

Сетевая точка зрения



Демонстрация

Установка

В Debian/Ubuntu

```
# apt-get install nagios3
```

Важные каталоги

```
/etc/nagios3
/etc/nagios3/conf.d
/etc/nagios-plugins/config
/usr/lib/nagios/plugins
/usr/share/nagios3/htdocs/images/logos
```

Web-интерфейс Nagios находится тут:

http://pcN.ws.nsrc.org/nagios3/

Настройка

- Конфигурация в текстовых файлах
  - /etc/nagios3/conf.d/*.cfg
  - Детали описаны http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html
- Конфигурация по умолчанию разбита на несколько файлов с разными объектами в разных файлах, но вы можете организовать конфигурацию любым удобным для вас способом
- Всегда проверяйте конфигурации перед перезапуском – в противном случае система мониторинга может перестать работать!
  - `nagios3 -v /etc/nagios3/nagios.cfg`

Настройка хостов и сервисов

Основана на шаблонах

- Сохраняет время, избегая повторений

Существуют шаблоны по умолчанию с параметрами для

- *Обычного хоста* (generic-host_nagios2.cfg)
- *Обычного сервися* (generic-service_nagios2.cfg)

- Индивидуальные настройки могут быть переопределены
- Значения по умолчанию имеют смысл

Мониторинг одного хоста

pcs.cfg

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

- Это минимальный рабочий конфиг
  - Вы только пингуете хост; Nagios предупредит вас о том, что вы не мониторите никаких сервисов
- Имя файла – любое, заканчивающееся на **`.cfg`**
- Организуйте устройства как нравится, например, похожие хосты в одном файле

Общий шаблон хоста

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!
}
```

Переопределение умолчаний

Все настройки могут быть переопределены для каждого хоста

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
}
```

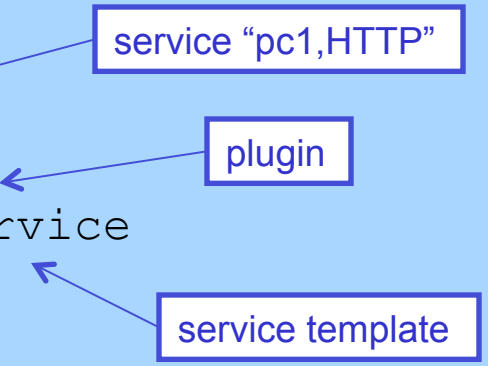
Определение сервисов (прямой способ)

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
}
```



Проверки сервисов

- Комбинация хост + сервис уникально идентифицирует проверку сервиса, например
  - “pc1,HTTP”
  - “pc1,SSH”
  - “pc2,HTTP”
  - “pc2,SSH”
- *check_command* указывает на плагин
- *Шаблон сервиса* добавляет настройки того, как часто выполняется проверка, а также кому и когда отсылать предупреждения

Общий шаблон сервиса

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
        }
```

*комментарии были удалены

Переопределение умолчаний

Опять-таки, настройки могут быть переопределены для каждого сервиса

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
}
```

Проверки одинаковых сервисов

- Часто нужно мониторить одинаковые сервисы на многих машинах
- Чтобы не повторять одну и ту же конфигурацию много раз, лучше определить проверку сервиса для всех машин в *группе хостов*

Создание групп хостов

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
}
```

Мониторинг сервисов в группе

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
}
```

Например если группа “http-servers” содержит pc1 и pc2, Nagios создаст проверки сервиса HTTP для обоих машин. Проверки сервиса будут названы “pc1,HTTP” и“pc2,HTTP”

Альтернативная точка зрения

- Вместо того, чтобы говорить “эта группа содержит следующие машины”, можно сказать что “эта машина принадлежит следующим группам машин”
- Тогда строка “members” в определении группы не нужна

Альтернативное членство в группе

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
}
```

Хосты и сервисы удобно определены в одном месте

Другие использования групп хостов

Выбор иконок для диаграммы статусов

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
}
```

Опционально: группы сервисов

- Также можно организовывать сервисы в “группы сервисов”
- Таким образом родственные или зависимые сервиы можно просматривать вместе в web-интерфейсе
- Сами сервисы должны уже быть определены

servicegroups.cfg

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

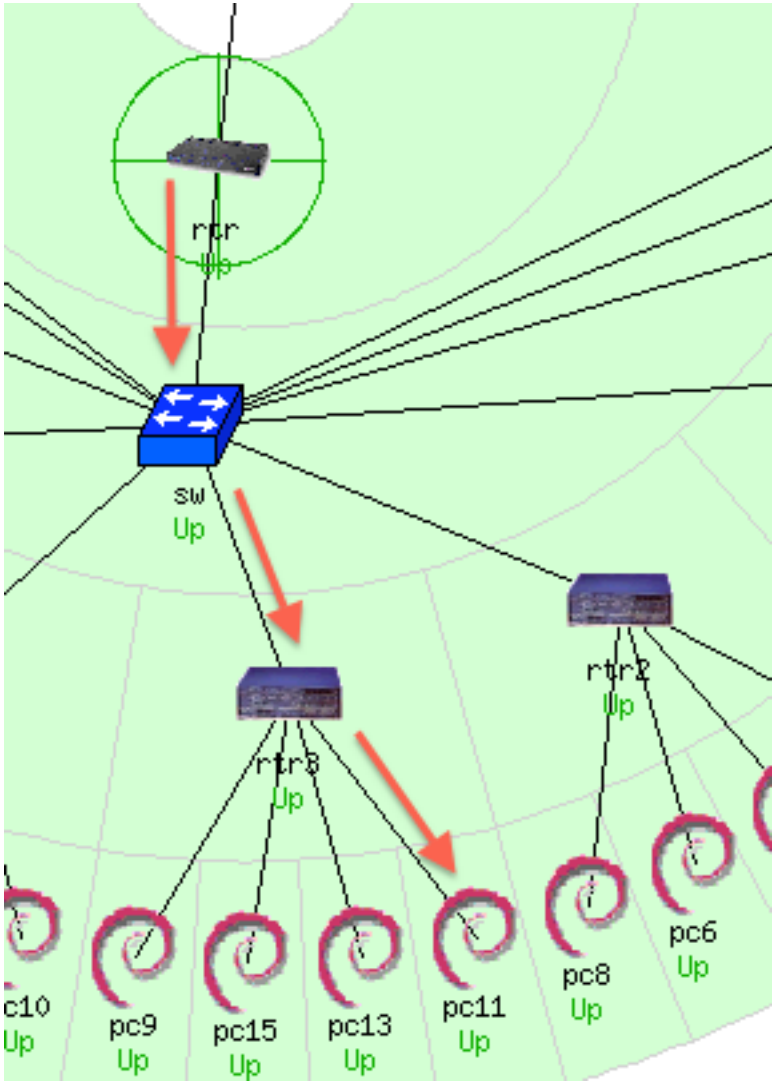
Конфигурация топологии

pcs.cfg

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

- Означает “pc1 на другой стороне rtr1 (относительно Nagios)”
- Если rtr1 ломается, pc1 станет “недоступным”, а не “неработающим”
- Предотвращает каскадные предупреждения если rtr1 перестает работать
- Позволяет Nagios создавать диаграммы состояния сети

Другая точка зрения на конфигурацию



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…

Уведомления “вне диапазона” (OOB)

Важно иметь независимую от вашей сети систему доставки сообщений (SMS или что-то другое)

- Мобильный телефон или другое устройство с SIM-картой, подсоединенное к серверу Nagios
- Доступные инструменты:

gammu: http://wammu.eu/
gnokii: http://www.gnokii.org/
sms-tools: http://smstools3.kekekasvi.com/

Ссылки

- **Nagios вебсайт**
  http://www.nagios.org/
- **Сайт для плагинов Nagios**
  http://www.nagiosplugins.org/
- *Хорошая книга Nagios System and Network Monitoring*, автор Wolfgang Barth.
- **Неофициальный сайт для плагинов**
  http://nagios.exchange.org/
- **Учебник по Nagios от Debian**
  http://www.debianhelp.co.uk/nagios.htm
- **Коммерческая поддержка Nagios**
  http://www.nagios.com/

Вопросы?

?

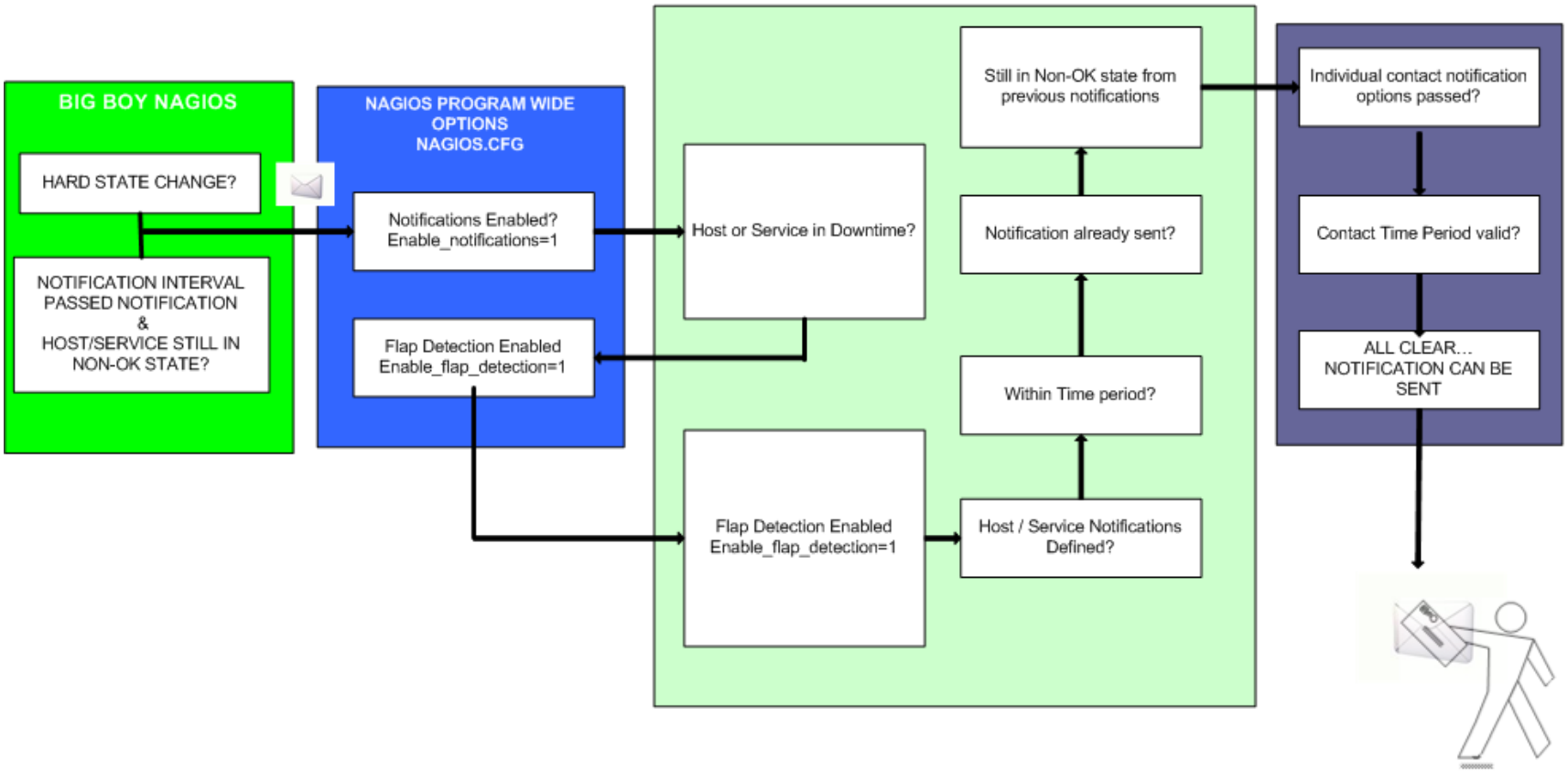
Дополнительные детали

Несколько дополнительных слайдов, могущих оказаться полезными…

Больше возможностей…

- Можно подтвердить получение предупреждения.
  - Пользователь может добавлять комментарии к событию
- Можно определять периоды техобслуживания
  - Для устройства или группы устройств
- Сохраняет историю доступности и создает отчеты
- Может детектировать ситуации “не работает” - “работает” - “опять не работает” и подавлять дополнительные сообщения.
- Поддерживает разные методы отправления сообщений:
  - e-mail, pager, SMS, всплывающее окно, звуковые, и т.д…
- Позволяет определять уровни событий для эскалации

NAGIOS - NOTIFICATION FLOW DIAGRAM



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

CONTACT GETS THE NOTIFICATION MESSAGE

Параметры уведомлений (хост)

Состояние хоста:

При конфигурации хоста вы можете получать сообщения о следующих ситуациях:

- **d:** DOWN
- **u:** UNREACHABLE
- **r:** RECOVERY
- **f:** FLAPPING (start/end)
- **s:** SCHEDULED DOWNTIME (start/end)
- **n:** NONE

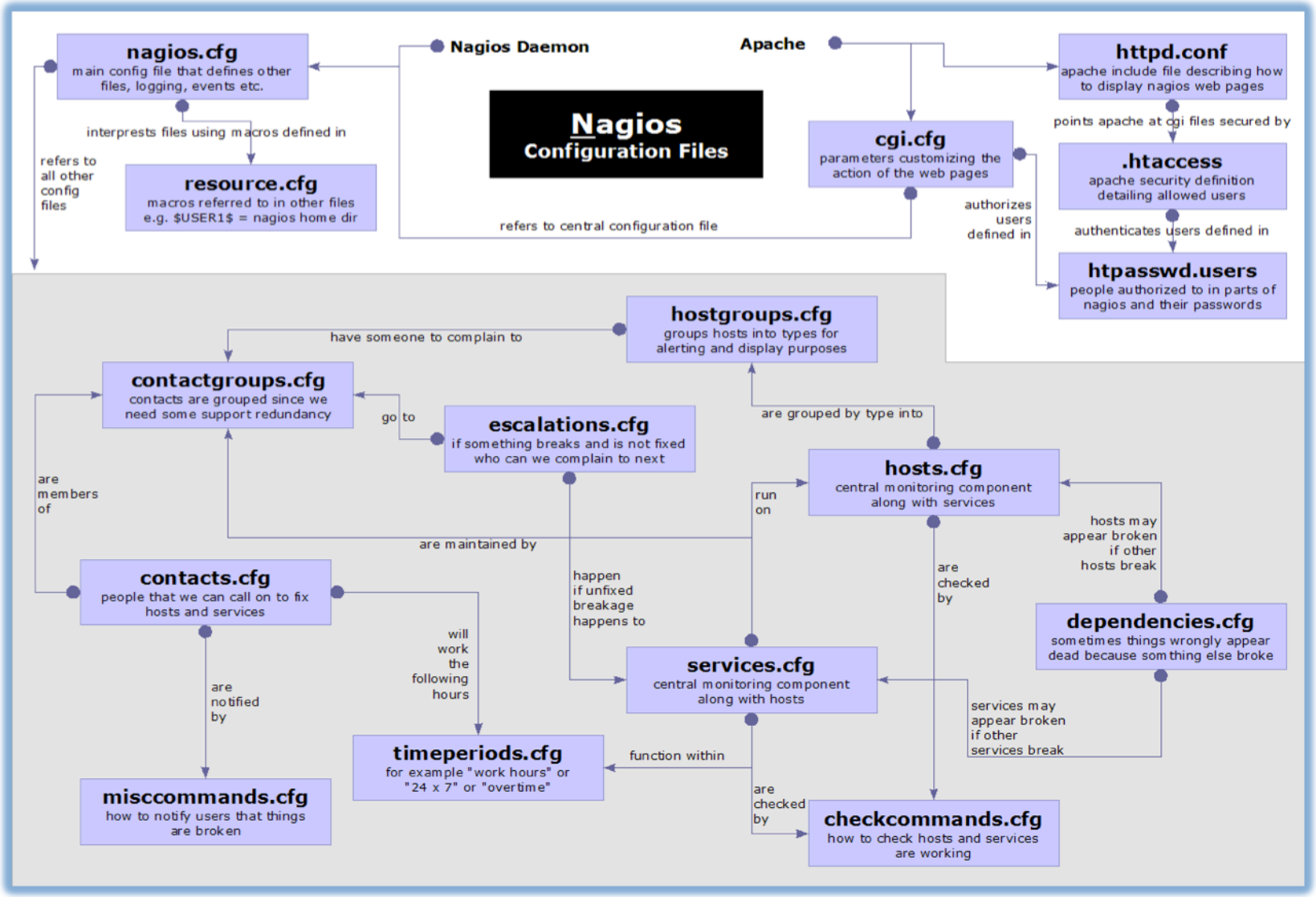
Параметры уведомлений (сервис)

Состояние сервиса:

При конфигурации сервиса вы можете получать сообщения о следующих ситуациях:

- **w:** WARNING
- **c:** CRITICAL
- **u:** UNKNOWN
- **r:** RECOVERY
- **f:** FLAPPING (start/end)
- **s:** SCHEDULED DOWNTIME (start/end)
- **n:** NONE

Файлы конфигурации (официальные)



Debian/Ubuntu структура файлов конфигурации

Находятся в /etc/nagios3/

Некоторые важные файлы:

- nagios.cfg — Главный файл конфигурации.
- cgi.cfg — Управляет web-интерфейсом и параметрами защиты.
- commands.cfg — Команды, используемые для передачи сообщений.
- conf.d/* — Все прочие параметры конфигурации!

Файлы конфигурации, прод.

В каталоге conf.d/

- contacts_nagios2.cfg — пользователи и группы
- extinfo_nagios2.cfg — красивый интерфейс
- generic-host_nagios2.cfg — шаблон хоста
- generic-service_nagios2.cfg — шаблон сервиса
- host-gateway_nagios3.cfg — задание роутера
- hostgroups_nagios2.cfg — группы узлов
- localhost_nagios2.cfg — определение хоста Nagios
- services_nagios2.cfg — список сервисов
- timeperiods_nagios2.cfg — когда проверять и кому сообщать

Файлы конфигурации, прод.

Некоторые другие файлы в conf.d:

- servicegroups.cfg — Группы сервисов
- pcs.cfg — Определения PC
- switches.cfg — Определения маршрутизаторов
- routers.cfg — Определения роутеров

Детали основной конфигурации

Глобальные настройки

Файл: `/etc/nagios3/nagios.cfg`

- Указывает, где находятся прочие файлы.
- Общее поведение Nagios:
  - Для крупных инсталляций желательно настраивать конфигурацию в этом файле.
  - См.: Настройка *Nagios для максимальной производительности* **http://nagios.sourceforge.net/docs/3_0/tuning.html**

Конфигурация CGI

/etc/nagios3/cgi.cfg

- Можно поменять каталог CGI
- Аутентификация и авторизация пользователей:
  - Активируйте аутентификацию через механизм .htpasswd в Apache, или через RADIUS либо LDAP.
  - Пользователи могут иметь разные права, настраиваемые следующими параметрами:
    - 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

Отрезки времени

Определение периодов времени, управляющих проверками, посылкой предупреждений и т.д.

- По умолчанию: 24 x 7
- Можно менять на например “только в рабочие дни”

```
# '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
        }
```

Конфигурация проверок сервисов/хостов

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

- Один и тот же плагин может вызываться по разному (“команды”)
- Команды и аргументы разделяются восклицательными знаками (!)
- Например, проверка SSH на нестандартном порту:

```
define service {
   hostgroup_name          ssh-servers-2222
   service_description     SSH-2222
   check_command           check_ssh_port!2222
   use                     generic-service
}
```

this is $ARG1$

Команды уведомления

Позволяет использовать любые желаемые команды. Мы можем настроить создание тикетов в 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

Конфигурация групп сервисов

```
# 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
}
```

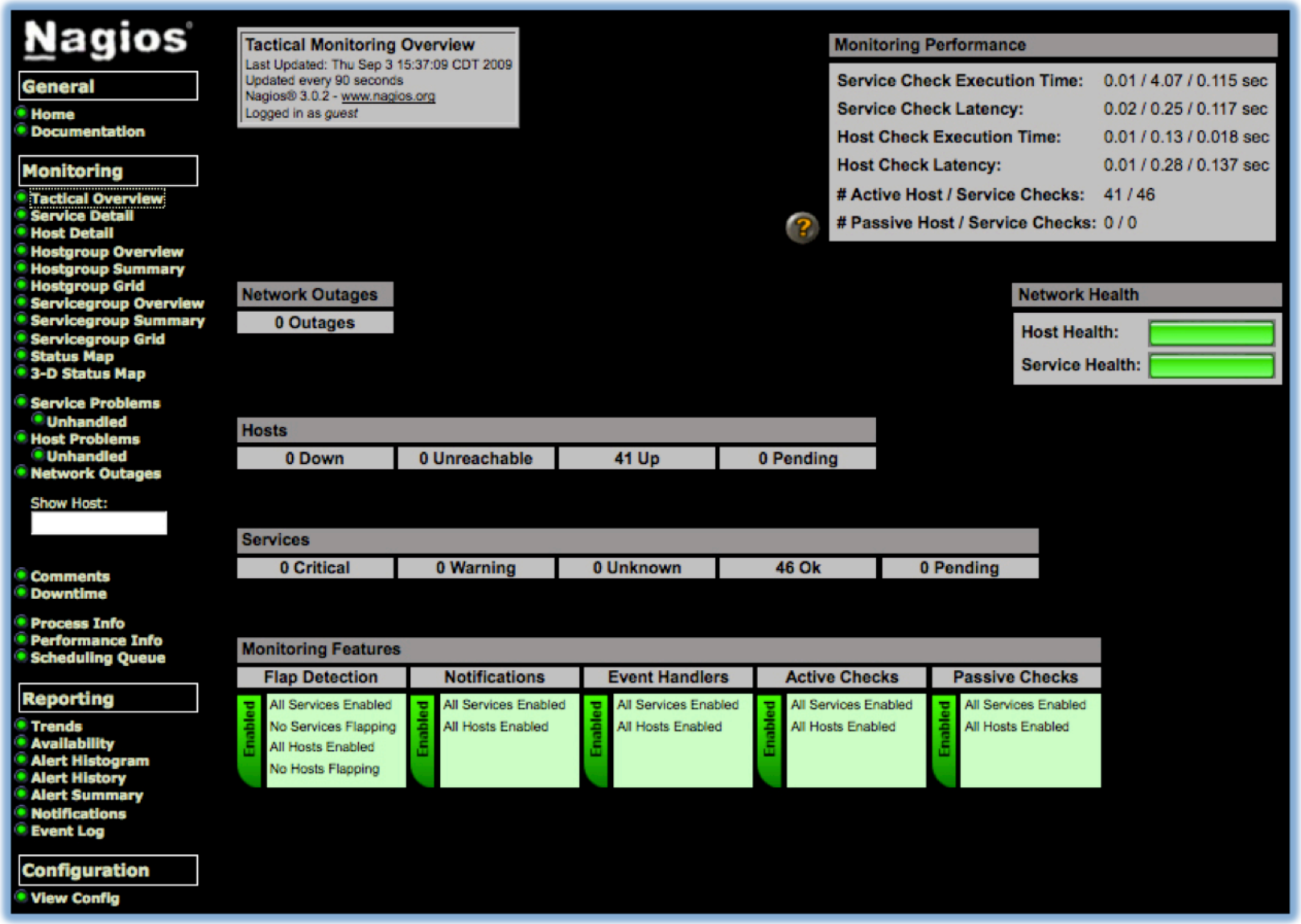
“service_description” важен, если мы хотим создавать группы сервисов. Вот пример:

```
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
}
```

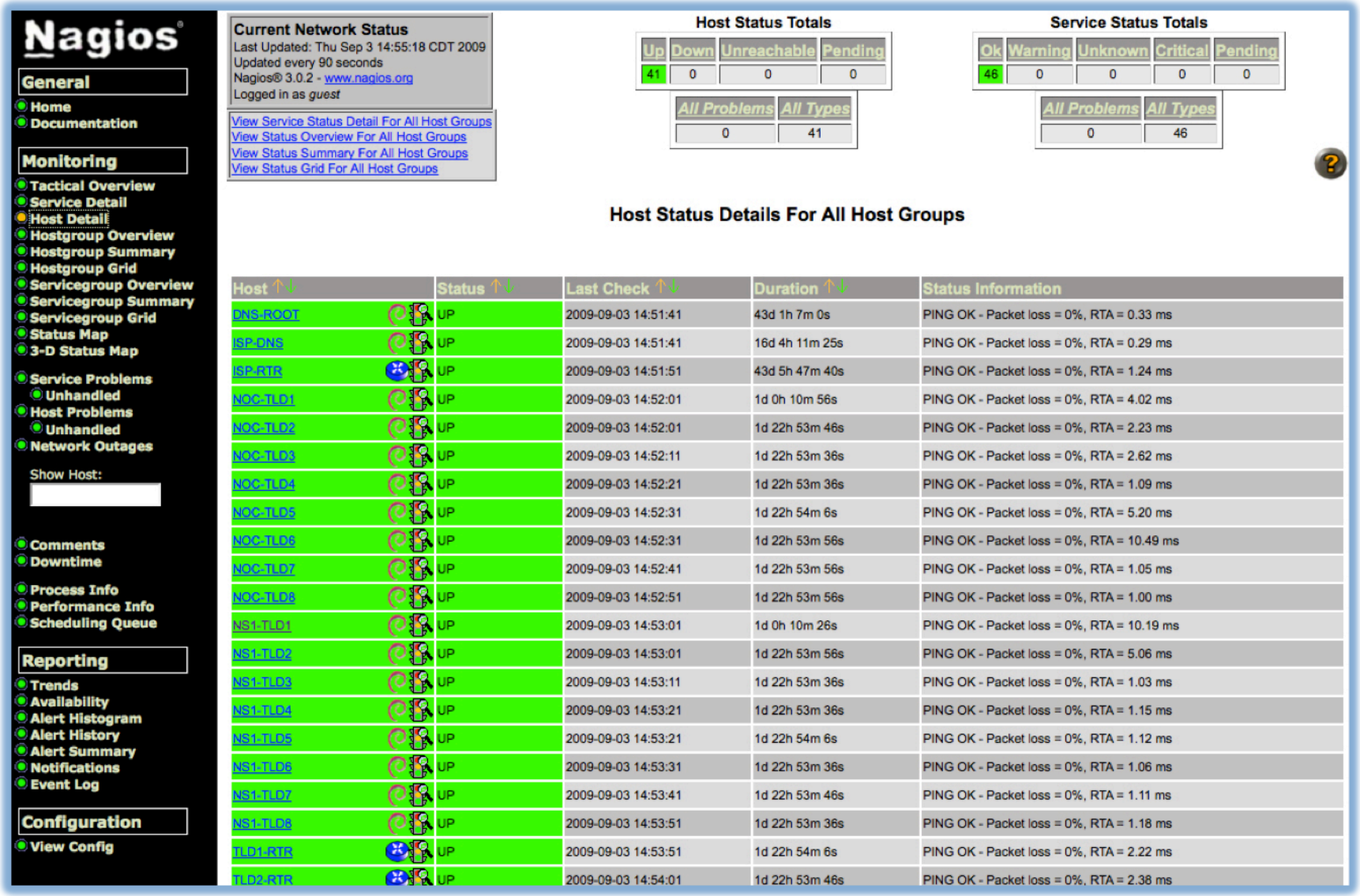
Снимки экранов

Примеры снимков экрана.

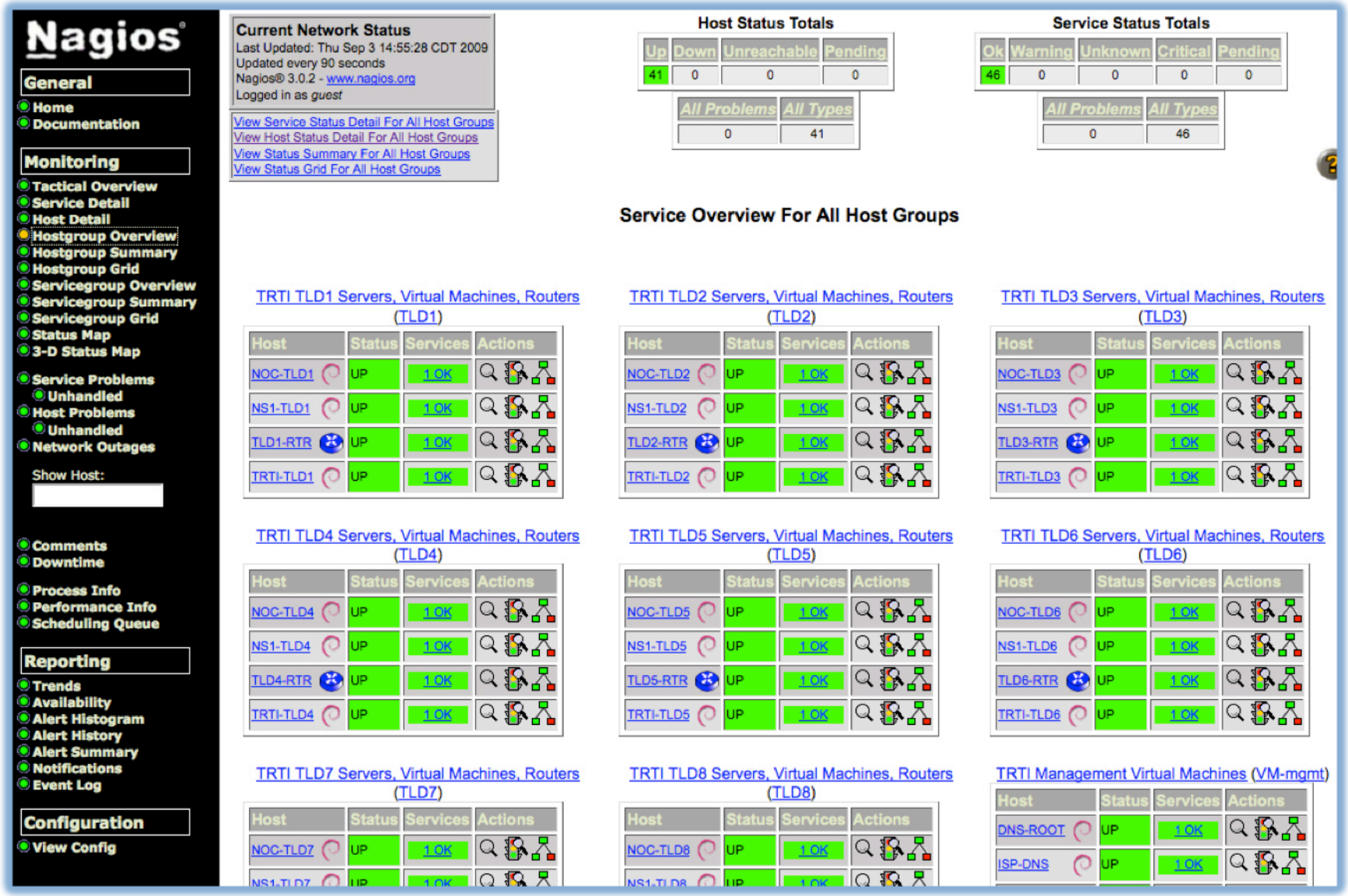
Общий обзор


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
Tactical Monitoring Overview
Last Updated: Thu Sep 3 15:37:09 CDT 2009
Updated every 90 seconds
Nagios® 3.0.2 - www.nagios.org
Logged in as guest
Monitoring Performance
Service Check Execution Time: 0.01 / 4.07 / 0.115 sec
Service Check Latency: 0.02 / 0.25 / 0.117 sec
Host Check Execution Time: 0.01 / 0.13 / 0.018 sec
Host Check Latency: 0.01 / 0.28 / 0.137 sec
Active Host / Service Checks: 41 / 46
Passive Host / Service Checks: 0 / 0
Network Outages
0 Outages
Network Health
Host Health:
Service Health:
Hosts
0 Down
0 Unreachable
41 Up
0 Pending
Services
0 Critical
0 Warning
0 Unknown
46 Ok
0 Pending
Monitoring Features
Flap Detection
Enabled
All Services Enabled
No Services Flapping
All Hosts Enabled
No Hosts Flapping
Notifications
Enabled
All Services Enabled
All Hosts Enabled
Event Handlers
Enabled
All Services Enabled
All Hosts Enabled
Active Checks
Enabled
All Services Enabled
All Hosts Enabled
Passive Checks
Enabled
All Services Enabled
All Hosts Enabled

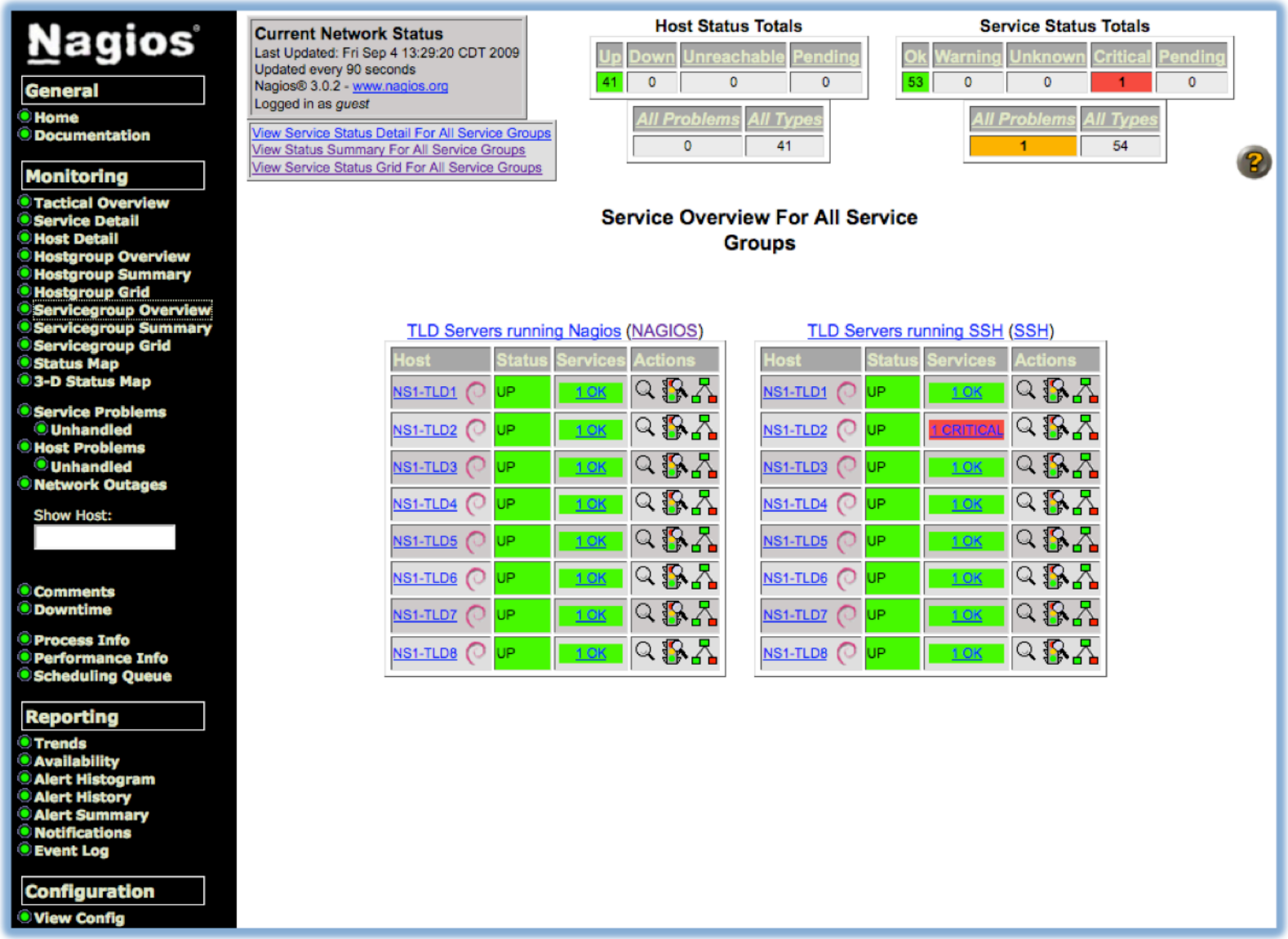
Детали хоста



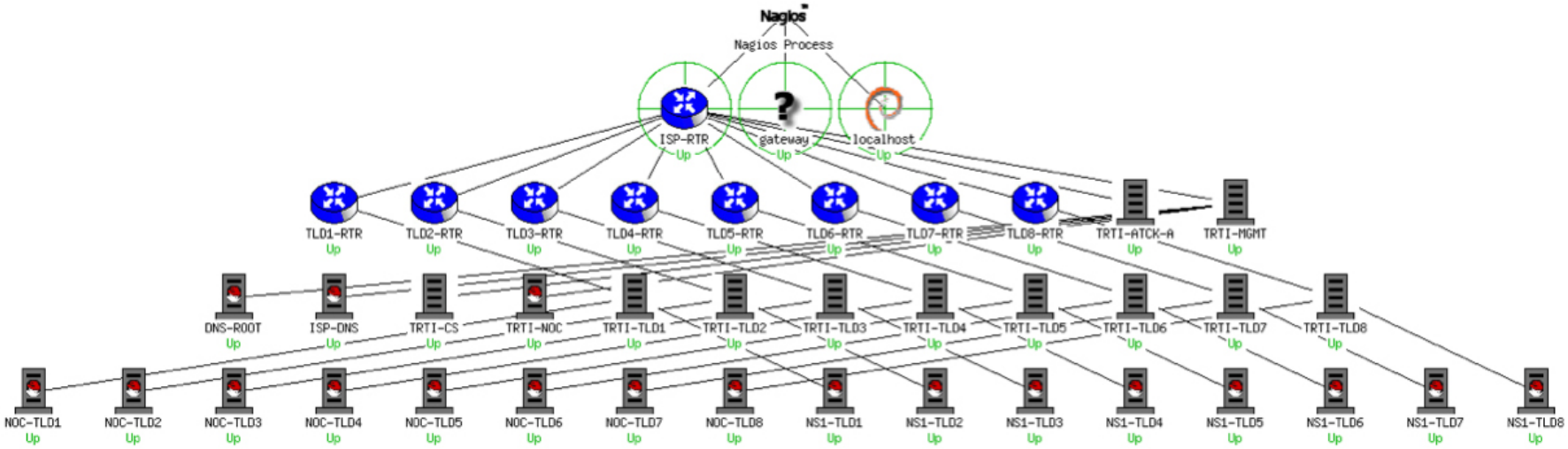
Обзор групп хостов



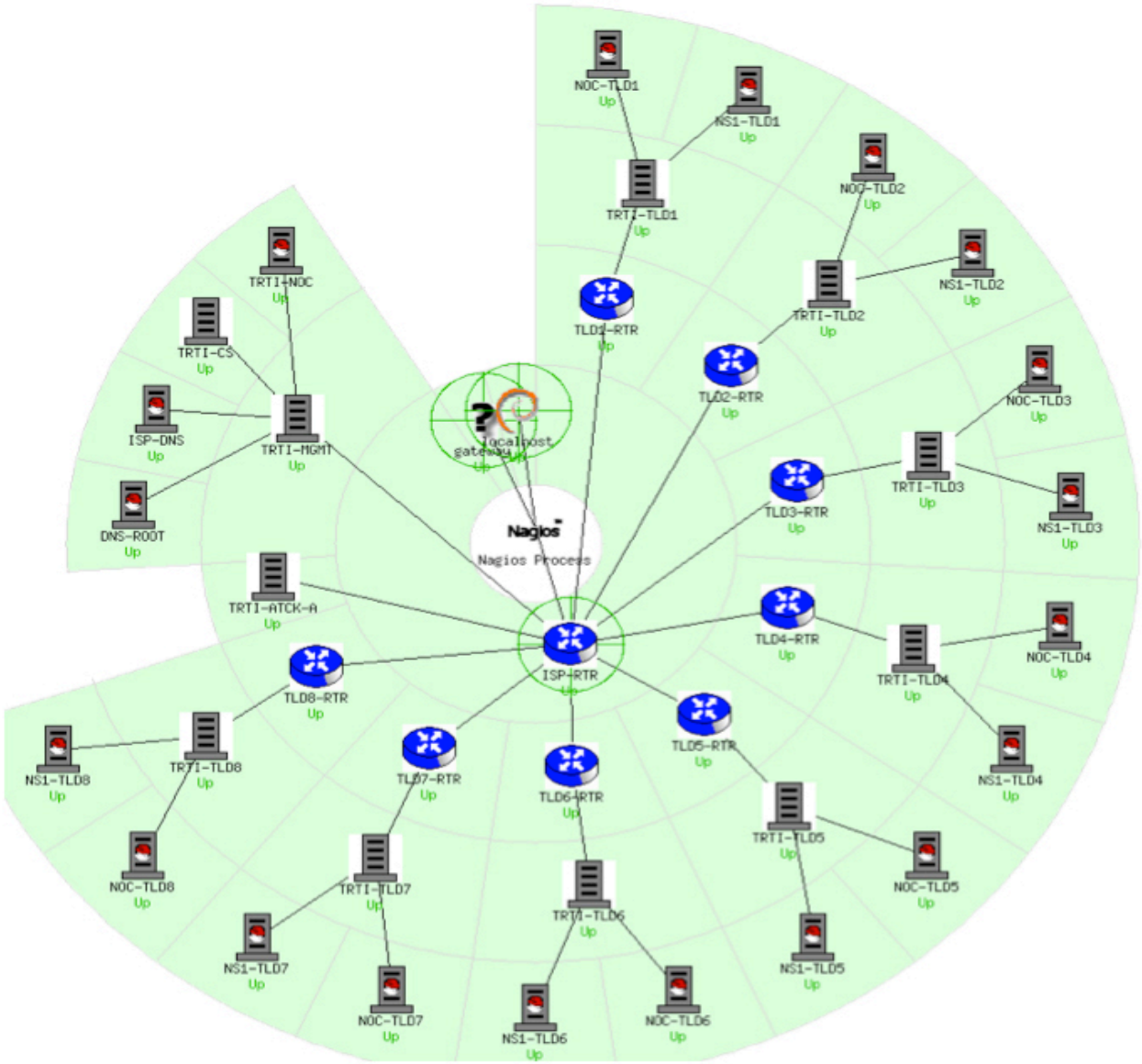
Обзор групп сервисов



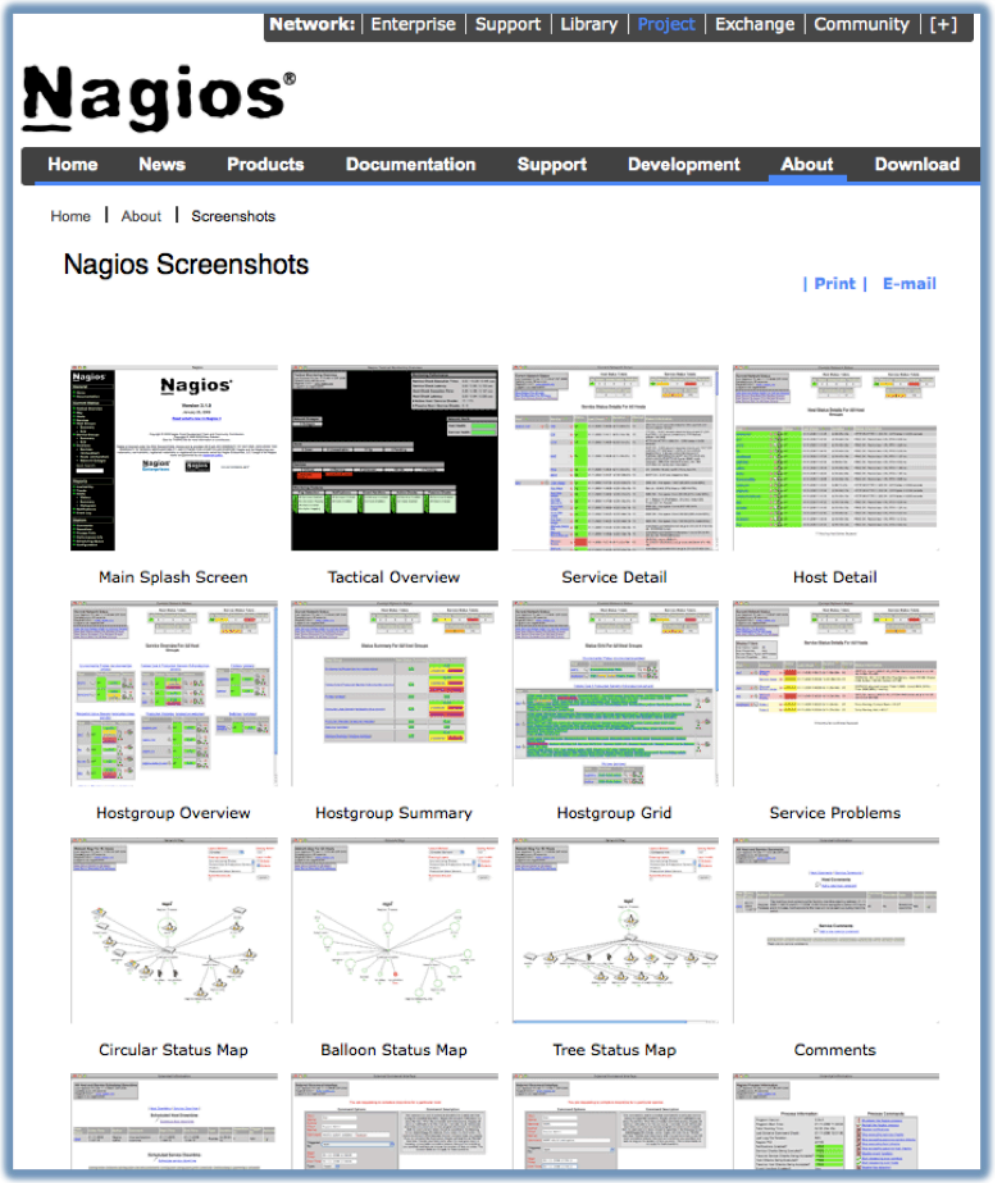
”Схлопнутая” диаграмма состояний



Размеченная круговая диаграмма состояний



Еще примеры снимков экранов



Намного больше примеров здесь:

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