Nagios 3.x + NConf - настройка системы мониторинга Nagios 3.x и утилиты конфигурирования NConf

Опубликовано muff в Пнд, 2010-10-04 03:48



Понадобилось настроить систему мониторинга. Раньше пользовался с этой целью системой мониторинга Nagios 2.x, однако уже есть возможность воспользоваться 3 версией. В последней не только исправлены найденные ранее ошибки, добавлены новые макросы и многое другое, но и пересмотрен алгоритм сканирования, с целью устранить один из главных недостатков этой системы – медлительность при проверке больших сетей. В 2.x все тесты проходят практически последовательно, а в новой редакции задачи выполняются параллельно. Хотя вторая версия еще развивается, очевидно, что в будущем все силы будут брошены на третью ветку.

Проект возник в 2002 году, хотя первое время он был известен как **NetSaint**. Его лидером является программист Этан Галстад. Само слово **Nagios**, по информации на сайте www.nagios.org [1], – это рекурсивный акроним, который расшифровывается, как **Nagios Ain't Gonna Insist On Sainthood** («***Nagios не собирается настаивать на святости***») – намек на предыдущее название проекта. Функциональность расширяется за счет плагинов и аддонов, большая часть из которых доступна на странице закачки.

Общеобразовательная часть завершена, приступаем к установке. Установка будет выполнена из системы портов:

```
# cd /usr/ports/net-mgmt/nagios && make install clean && rehash
```

Автоматически устанавливаются плагины, расширяющие основной функционал системы.

Также, во время установки система предлагает создать пользователя и групу **nagios**. Советую воспользоваться данным предложением.

По завершении установки советую ознакомиться с инструкциями, котрые вывел Nagios после установки:

```
**********************************************************************

 Enable Nagios in /etc/rc.conf with the following line:

   nagios_enable="YES"

 Configuration templates are available in /usr/local/etc/nagios as
 *.cfg-sample files.  Copy them to *.cfg files where required and
 edit to suit your needs.  Documentation is available in HTML form
 in /usr/local/www/nagios/docs.
```

```
If you don't already have a web server running, you will need to
install and configure one to finish off your Nagios installation.
When used with Apache, the following should be sufficient to publish
the web component of Nagios (modify the allow list to suit):

  <Directory /usr/local/www/nagios>
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
    php_flag engine on
    php_admin_value open_basedir /usr/local/www/nagios/:/var/spool/nagios/
  </Directory>

  <Directory /usr/local/www/nagios/cgi-bin>
    Options ExecCGI
  </Directory>

  ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/
  Alias /nagios/ /usr/local/www/nagios/

**********************************************************************
```

Приступаем к выполнению предписаний... Добавим в **rc.conf** загрузку **Nagios**:

```
# echo '# Nagios' >> /etc/rc.conf
# echo 'nagios_enable="YES"' >> /etc/rc.conf
```

Добавляем в конфигурационный файл Apache (/usr/local/etc/apache22/httpd.conf) следующий блок:

```
# ****************************************** #

Alias /nagios/ "/usr/local/www/nagios/"

<Directory /usr/local/www/nagios/>
        Order deny,allow
        Allow from all
        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /usr/local/etc/nagios/.htpasswd.nagios
        Require valid-user
</Directory>

<Directory /usr/local/www/nagios/cgi-bin/>
        Options ExecCGI
        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /usr/local/etc/nagios/.htpasswd.nagios
        Require valid-user
</Directory>

# ****************************************** #
```

Поскольку указана необходимость авторизации, добавим пользователя в указанный файл авторизации:

```
# htpasswd -c /usr/local/etc/nagios/.htpasswd.nagios username
New password:
Re-type new password:
Adding password for user username
```

Обратите внимание на ключ "-c" - он указывает на необходимость создания файла авторизации. При последующем добавлении пользователей этот ключ не указываем.

Примечание: необходима поддержка обработки cgi-скриптов. В секции **<IfModule mime_module>** должна быть расскоментирована строка **"AddHandler cgi-script .cgi"**.

Проверяем синтаксис конфигурационного файла, и если все в порядке, перезапускаем Apache:

```
# apachectl configtest
Syntax OK
# apachectl graceful
```

Теперь можно проверить, обрабатывает ли веб-сервер запросы. Для этого необходимо подключиться по адресу http://ip_servera/nagios/ [2]

Должно отобразиться окно авторизации, а после успешной авторизации - главное окно мониторинговой системы:



[3]

Теперь можно пойти двумя путями. Первый - отредактировать конфигурационные файлы Nagios вручную, второй - установить интерфейс конфигурирования. Ввиду админской лени, остановимся на втором варианте.

Для Nagios второй ветки я использовал Fruity. Но, поскольку решил перейти на 3 ветку Nagios, то и интерфейс конфигурирования протестируем какой-то новый.

Выбор пал на **Nconf (Enterprise Nagios configurator)** - дополнение для конфигурирования Nagios, написанное на PHP. В портах Nconf не был обнаружен, поэтому идем на страницу проекта [4] и ищем последнюю версию для загрузки. На момент написания статьи - это была версия NConf 1.2.6 от 26.11.2009.

```
# cd /usr/local/www
# fetch http://sourceforge.net/projects/nconf/files/nconf/1.2.6-0/nconf-1.2.6-0.tgz
```

Распакуем архив и сделаем владельцем файлов и каталогов пользователя www:

```
# tar -xzf nconf-1.2.6-0.tgz
# chown -R www:www nconf
```

Добавим в конфигурационный файл Apache еще один блок:

```
Alias /nconf/ "/usr/local/www/nconf/"
<Directory /usr/local/www/nconf/>
        Order deny,allow
        Allow from all
        AuthName "Nagios configurator Access"
        AuthType Basic
        AuthUserFile /usr/local/etc/nagios/.htpasswd.nagios
        Require valid-user
</Directory>
```

Соответственно, "передергиваем" Apache:

```
# apachectl configtest
Syntax OK
# apachectl graceful
```

Пора ознакомиться с требованиями и особенностями инсталляции. Все описано в файле **/usr/local/www/nconf/README**. Требования следующие:

```
NConf system requirements:

    * Apache webserver    * PHP 5 or higher, php-mysql, php-ldap (only if using LDAP
 auth)    * MySQL 5.0.2 or higher (with InnoDB)    * Perl 5.6 or higher, perl-DBI, p
erl-DBD-MySQL    * Nagios 3.x or Icinga 0.8x (binary necessary for testing generated
 config)

php.ini settings:

    * short_open_tag = On    * register_globals = Off    * magic_quotes_gpc = Off
```

Теперь по очереди...

*** Apache webserver**.

```
# pkg_info | grep apache
apache-2.2.16      Version 2.2.x of Apache web server with prefork MPM.
```

Apache - установлен. Пропускаем.

*** PHP 5 or higher, php-mysql**.

```
# pkg_info | grep php


php52-5.2.14        PHP Scripting Languagephp52-bz2-5.2.14    The bz2 shared extensi
on for phpphp52-ctype-5.2.14  The ctype shared extension for phpphp52-curl-5.2.14
The curl shared extension for phpphp52-dom-5.2.14    The dom shared extension for ph
```

```
pphp52-filter-5.2.14 The filter shared extension for phpphp52-gd-5.2.14     The gd s
hared extension for phpphp52-gettext-5.2.14 The gettext shared extension for phpphp5
2-hash-5.2.14   The hash shared extension for phpphp52-iconv-5.2.14  The iconv share
d extension for phpphp52-json-5.2.14   The json shared extension for phpphp52-mbstri
ng-5.2.14 The mbstring shared extension for phpphp52-mcrypt-5.2.14 The mcrypt shared
 extension for phpphp52-mhash-5.2.14  The mhash shared extension for phpphp52-mysql-
5.2.14  The mysql shared extension for phpphp52-openssl-5.2.14 The openssl shared ex
tension for phpphp52-pcre-5.2.14   The pcre shared extension for phpphp52-session-5.
2.14 The session shared extension for phpphp52-simplexml-5.2.14 The simplexml shared
 extension for phpphp52-snmp-5.2.14   The snmp shared extension for phpphp52-sockets
-5.2.14 The sockets shared extension for phpphp52-spl-5.2.14    The spl shared exten
sion for phpphp52-xml-5.2.14    The xml shared extension for phpphp52-zip-5.2.14
The zip shared extension for phpphp52-zlib-5.2.14   The zlib shared extension for ph
pphpMyAdmin-3.3.6    A set of PHP-scripts to manage MySQL over the web
```

С PHP тоже все в порядке.

*** MySQL 5.0.2 or higher (with InnoDB)**.

```
# pkg_info | grep mysql
mysql-client-5.1.50_1 Multithreaded SQL database (client)
mysql-server-5.1.50 Multithreaded SQL database (server)
php52-mysql-5.2.14  The mysql shared extension for php
```

MySQL 5.1... Все в порядке.

*** Perl 5.6 or higher, perl-DBI, perl-DBD-MySQL.**

```
# pkg_info | grep perl
p5-GD-2.45        A perl5 interface to Gd Graphics Library version2
p5-GD-Graph-1.44.01_4 Graph plotting module for perl5
perl-5.10.1_2       Practical Extraction and Report Language
```

Perl установлен, а вот perl-DBI и perl-DBD-MySQL необходимо доустановить.

*** Nagios 3.x or Icinga 0.8x.**

```
# pkg_info | grep nagios
nagios-3.2.2_1      Extremely powerful network monitoring system
nagios-plugins-1.4.15_1,1 Plugins for Nagios
```

Ну, и Nagios... Конечно же установлен.

Выполним установку **perl-DBI**:

```
# cd /usr/ports/databases/p5-DBI && make install clean && rehash
```

Аналогично выполним установку **perl-DBD-MySQL**. Поскольку у меня версия MySQL v.5.1.50, то устанавливать буду p5-DBD-mysql51:

```
# cd /usr/ports/databases/p5-DBD-mysql51 && make install clean && rehash
```

Теперь проверяем настройки PHP. У меня все опции соответствовали требуемым. Кстати, проверить настройки PHP можно следующим образом. Создадим файл info.php следующего содержания:

```
# cat /usr/local/www/nconf/info.php

<?php
```

```
        phpinfo();
?>
```

Чтобы просмотреть настройки PHP, необходимо в браузере войти на следующий адрес: **http://ip_servera/nconf/info.php**.

Условия выполнены... Приступим к инсталляции. Можно выполнить все действия, описанные в README. А можно запустить скрипт установки: **http://ip_servera/nconf/INSTALL.php**.

Скриншоты установки:

Проверка системных требований.



[5]

Шаг 1. Конфигурация базы данных.

Предварительно создадим саму базу данных и пользователя с необходимыми правами.

```
# mysql -u root -p
Enter password:
mysql> CREATE DATABASE nconf;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL ON `nconf`.* TO 'nconf'@'localhost' IDENTIFIED BY
'nconf_user_pass_here';
Query OK, 0 rows affected (0.04 sec)
```

Теперь заполним необходимые поля и жмем "Next"



[6]

Таблицы созданы успешно...



[7]

Шаг 2. Конфигурирование.

Необходимо указать правильный путь к бинарнику Nagios - **/usr/local/bin/nagios**



[8]

Шаг 3. Настройка аутентификации.

Без аутентификации. Кто имеет доступ к Nagios, тот имеет доступ и к NConf (настройки Apache).



[9]

Шаг 4. Завершение установки.

Необходимо удалить инсталяционные файлы и директории:

```
# rm -R /usr/local/www/nconf/INSTALL
# rm /usr/local/www/nconf/INSTALL.php
# rm -R /usr/local/www/nconf/UPDATE
# rm /usr/local/www/nconf/UPDATE.php
```



[10]

Результат.

Если все прошло успешно, в результате имеем установленный NConf.



[11]

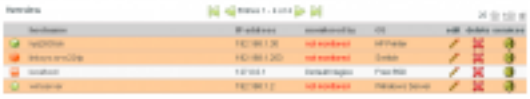
Конфигурирование NConf

Для начала советую побродить по меню для знакомства с возможностями... Можно обнаружить, что некоторые базовые настройки уже добавлены. Уже мониторится локальный хост.

Добавим в мониторинг коммутатор с IP-адресом 10.10.0.2 и изменим некоторые параметры... Теперь все по очереди.

Hosts

Имеем по умолчанию в наличии 4 хоста для мониторинга:



[12]

Мониторится из них только один - **localhost**, остальные отключены. Поскольку необходимо добавить коммутатор, то я не буду добавлять новый хост, а отредактирую присутствующий в примере **linksys-srw224p**.

Жмем "Edit" и правим до следующего состояния:



[13]

Сохраняем изменения. Удаляем неактивные хосты и добавляем мониторинг необходимых сервисов для каждого оставшегося хоста. Для коммутатора я оставил только **check_ping**, а для сервера набор сервисов был следующий:



[14]

Переходим на закладку **Hostgroups**.

Hostgroups

На этой закладке я добавил еще одну групу устройств - **voip**, а **linux-servers** переименовал в **unix-servers**. Также, зайдя в **hostgroup name - switches**, добавил свой комутатор.

Contacts

На этой вкладке отредактировал существующего пользователя **nagiosadmin** на свои контактные данные:



[15]

Самая примитивная конфигурация готова...

Теперь пора внести некоторые изменения в **Nagios**. Скопируем предложенные ***-sample** файлы в файлы конфигурации:

```
# cd /usr/local/etc/nagios
# cp cgi.cfg-sample cgi.cfg
# cp nagios.cfg-sample nagios.cfg
# cp resource.cfg-sample resource.cfg
```

Теперь необходимо отредактировать файл **nagios.cfg**. Необходимо обнаружить все входжения в файл строк, которые начинаются на **cfg_dir** и **cfg_file** и удалить либо закомментировать их. Вместо удаленных строк необходимо добавить такие:

```
cfg_dir=/usr/local/etc/nagios/global
cfg_dir=/usr/local/etc/nagios/Default_collector
```

Создадим каталог, куда будуть импортироваться конфигурационные файлы:

```
# mkdir /usr/local/etc/nagios/import
```

Теперь отредактируем скрипт переноса конфигурации, который идет в комплекте с **NConf**. В результате получим:

```
# cat /usr/local/www/nconf/ADD-ONS/deploy_local.sh

#!/bin/sh

OUTPUT_DIR="/usr/local/www/nconf/output/"
NAGIOS_DIR="/usr/local/etc/nagios/"
TEMP_DIR=${NAGIOS_DIR}"import/"
CONF_ARCHIVE="NagiosConfig.tgz"

if [ ! -e ${TEMP_DIR} ] ; then
mkdir -p ${TEMP_DIR}
fi

if [ ${OUTPUT_DIR}${CONF_ARCHIVE} -nt ${TEMP_DIR}${CONF_ARCHIVE} ] ; then
```

```
cp -p ${OUTPUT_DIR}${CONF_ARCHIVE} ${TEMP_DIR}${CONF_ARCHIVE}
tar -xf ${TEMP_DIR}${CONF_ARCHIVE} -C ${NAGIOS_DIR}
/usr/local/etc/rc.d/nagios reload
fi

exit
```

Добавим в **/etc/crontab** такую строку:

```
* * * * * root /bin/sh /usr/local/www/nconf/ADD-ONS/deploy_local.sh
```

Перезапустим cron, чтобы изменения вступили в силу:

```
# killall -HUP cron
```

Кстати, необходимо также скопировать файлы изображений для их корректного отображения:

```
# cp -r /usr/local/www/nconf/img/logos /usr/local/www/nagios/images/
```

Теперь сгенерируем конфигурационные файлы Nagios-а, нажав в NConf на ссылку **Generate Nagios config**. В результате имеем в каталоге **/usr/local/www/nconf/output/** архив с конфигурационными файлами. Осталось подождать минуту-две (чтобы запустился скрипт /usr/local/www/nconf/ADD-ONS/deploy_local.sh) и запустить Nagios:

```
# sh /usr/local/etc/rc.d/nagios start
Performing sanity check of nagios configuration: OK
Starting nagios.
```

Переходим на web-интерфейс Nagios наблюдаем примерно такую картину:



[16]

Теперь дело за малым - добавить необходимые хосты и указать сервисы для мониторинга. Базовая настройка закончена.

Примечания:

1. Внес изменения в файл **/usr/local/etc/nagios/nagios.cfg**. Изменил параметр **date_format**. Новое значение - **euro**.

2. Чтобы Nagios "обрел голос", необходимо расскоментировать следующие строки в файле **/usr/local/etc/nagios/cgi.cfg**:
   - host_unreachable_sound=hostdown.wav
   - host_down_sound=hostdown.wav
   - service_critical_sound=critical.wav
   - service_warning_sound=warning.wav
   - service_unknown_sound=warning.wav
   Также необходимо разместить сами файлы в каталоге **/usr/local/www/nagios/media/**.
   А скачать файлы можно по следующим ссылкам:
   - critical.wav [17]
   - hostdown.wav [18]
   - warning.wav [19]

3. В дефолтной конфигурации можна наблюдать вот такие сообщения о нехватке прав доступа:
   
   [20]
   "Лечится" это назначением прав доступа к определенным разделам в файле **/usr/local/etc/nagios/cgi.cfg**. Я же, не заморачиваясь, разрешил полный доступ всем авторизованым пользователям:

```
# cat /usr/local/etc/nagios/cgi.cfg | grep authorized

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=*
#authorized_for_read_only=user1,user2

# sh /usr/local/etc/rc.d/nagios reload
```

4. Рекомендую также включить "**Passive Checks**".
   В **NConf** в разделе "**Nagios Servers**" отображаем "**Nagios-collectors**", нажав кнопку "**Show**" напротив этого параметра. Далее необходимо отредактировать доступный сервер "**Default Nagios**" и параметр "**passive checking**" устанавливаем в значение "**1**".
   Сохраняем изменения и жмем "**Generate Nagios config**".

5. Есть возможность изменить периодичность ротации логов. За периодичночсть ротации логов отвечает параметр **log_rotation_method** в файле

/usr/local/etc/nagios/nagios.cfg. Доступные значения:
- **n** - ***none*** - не выполнять ротацию логов
- **h** - ***hourly*** - ротация выполняется каждый час
- **d** - ***daily*** - ежедневная ротация (в полночь, каждый день)
- **w** - ***weekly*** - еженедельно (в полночь, в ночь с субботы на воскресенье)
- **m** - ***monthly*** - ежемесячно (в полночь последнего дня месяца).

Установил у себя еженедельную ротацию логов: **log_rotation_method=w**

6. В настройках по умолчанию, уведомления на почту не приходят. Причина - ошибочные пути к мыльнику.
"Лечится" это очень просто. Необходимо вывести список "**Misccommands**". Нас интересуют такие "**misc command name**":
- **notify-host-by-email**
- **notify-service-by-email**

Их значения по умолчанию такие:
notify-host-by-email:
/usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s "** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" $CONTACTEMAIL$

notify-service-by-email:
/usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$" | /bin/mail -s "** $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" $CONTACTEMAIL$

Все что нам нужно - это изменить путь с **/bin/mail** на **/usr/bin/mail**. Ну и дать команду "**Generate Nagios config**".

Источник (получено *2026-11-20 00:14*):
http://muff.kiev.ua/content/nagios-3x-nconf-nastroika-sistemy-monitoringa-nagios-3x-i-utility-konfigurirovaniya-nconf

Ссылки:
[1] http://www.nagios.org
[2] http://ip_servera/nagios/
[3] http://muff.kiev.ua/files/imagepicker/1/nagios.png
[4] http://www.nconf.org/
[5] http://muff.kiev.ua/files/imagepicker/1/nconf0.png
[6] http://muff.kiev.ua/files/imagepicker/1/nconf1.png
[7] http://muff.kiev.ua/files/imagepicker/1/nconf2.png
[8] http://muff.kiev.ua/files/imagepicker/1/nconf3.png
[9] http://muff.kiev.ua/files/imagepicker/1/nconf4.png
[10] http://muff.kiev.ua/files/imagepicker/1/nconf5.png
[11] http://muff.kiev.ua/files/imagepicker/1/nconf6.png
[12] http://muff.kiev.ua/files/imagepicker/1/nconf7.png
[13] http://muff.kiev.ua/files/imagepicker/1/nconf8.png
[14] http://muff.kiev.ua/files/imagepicker/1/nconf10.png
[15] http://muff.kiev.ua/files/imagepicker/1/nconf9.png
[16] http://muff.kiev.ua/files/imagepicker/1/nagios1.png
[17] http://muff.kiev.ua/files/critical.wav
[18] http://muff.kiev.ua/files/hostdown.wav
[19] http://muff.kiev.ua/files/warning.wav
[20] http://muff.kiev.ua/files/imagepicker/1/nagios2.png