Nagios - "рихтуем" систему под себя. Создание команд проверки.

Опубликовано muff в Втр, 2010-10-05 01:45



Nagios настроен [1] и работает... Теперь попробуем описать ситуации, когда пришлось поработать "напильником".

"Длинный пинг".

Мониторится корпоративная сеть, к тому же офисы регионально и техплощадки регионально разбросаны. В результате имеем пинг к некоторым хостам в удаленном офисе - 400 милисекунд (да, я не опечатался, как никак 8 тыс. км и 22 хопа по traceroute). **Nagios** "возмущается" на такое дело, и сообщает, что "PING CRITICAL - Packet loss = 0%, RTA = 403.51 ms".

Решение оказалось несложным. Просматриваем сервисы, назначенные этому хосту и жмем кнопку редактирования команды "check_ping". Изменяем параметры "ARG1" и "ARG2" на подходящие значения, где "ARG1" - Warning level, а "ARG2" - Critical Level.

Генерируем конфигурацию и идем пить пиво...

Результат: PING OK - Packet loss = 0%, RTA = 407.56 ms

"check dns"

Необходимо проверять, обрабатывает **DNS**-сервер запросы, или нет. Поможет в этом плагин "**check_dns**". Необходимое условие - **DNS**-сервер должен обрабатывать запросы с сервера, на котором установлен **Nagios**.

Создаем новую "Checkcommand" - "check_dns".

Для этого в меню "Additional Items", пункт "Checkcommands" жмем кнопку "Add" и заполняем форму следующим образом:

check command name	check_dns
default service name	
check command line	\$USER1\$/check_dns -H <u>www.yahoo.com</u> [2] -s \$HOSTADDRESS\$
command description	ARG1=Multiple options
default command params	!
amount of params	1

Результат: DNS OK: 0.233 seconds response time. www.yahoo.com [2] returns 87.248.122.122

"check telnet"

Необходимо проверять возможность подключения к хосту по **Telnet**. Стандартного плагина нету, но мы знаем, что телнет запущен на 23 порту, протокол TCP. Поэтому

воспользуемся возможностями "check_tcp".

Создаем новую "Checkcommand" - "check telnet":

check command name	check_telnet
default service name	
check command line	\$USER1\$/check_tcp -H \$HOSTADDRESS\$ -p 23
command description	ARG1=Multiple options
default command params	!
amount of params	1

Результат: TCP OK - 0.001 second response time on port

23|time=0.001366s;;;0.000000;10.000000

"check_vpn"

Необходимо проверять возможность подключения по VPN-у. Стандартного плагина нету, а VPN работает на порту 1723, протокол TCP. Поэтому воспользуемся возможностями "**check_tcp**".

Создаем новую "Checkcommand" - "check_vpn":

	_ !
check command name	check_vpn
default service name	
check command line	\$USER1\$/check_tcp -H \$HOSTADDRESS\$ -p 1723
command description	ARG1=Multiple options
default command params	!
amount of params	1

Результат: TCP OK - 0.001 second response time on port 1723|time=0.000582s;;;0.000000;10.000000

"check mysql"

Heoбходимо проверять возможность подключения к БД MySQL. Воспользуемся возможностями плагина "check_mysql".

Создаем новую "Checkcommand" - "check mysql":

отория от	<u>-</u>
check command name	check_mysql
default service name	
check command line	\$USER1\$/check_mysql -H \$HOSTADDRESS\$ -u \$ARG1\$ -p \$ARG2\$
command description	ARG1=Multiple options,ARG2=Multiple options
default command params	!nagios!nagios_pass
amount of params	2

Не забываем на проверяемом хосте добавить пользователя MySQL:

mysql -u muff -p

Enter password:

mysql> CREATE USER 'nagios'@'ip_nagios_server_here' IDENTIFIED BY 'nagios-pass'; Query OK, 0 rows affected (0.00 sec)

Результат: Uptime: 613210 Threads: 4 Questions: 2573878 Slow gueries: 0 Opens: 67 Flush

tables: 1 Open tables: 59 Queries per second avg: 4.197

"check_ntp"

Будем проверять, работает ли сервер времени на хосте. Воспользуемся возможностями плагина "**check_ntp**".

Создаем новую "Checkcommand" - "check ntp":

check command name	check_ntp
default service name	
check command line	<pre>\$USER1\$/check_ntp -H \$HOSTADDRESS\$ -w \$ARG1\$ -c \$ARG2\$</pre>
command description	ARG1=Warn level,ARG2=Critical Level
default command params	!0.4!0.9
amount of params	2

Результат: NTP OK: Offset 0.005161 secs

Пока все... Нестандартные проверки будут добавляться по мере необходимости...

Кстати. Доступные команды проверки можно найти в каталоге /usr/local/libexec/nagios.

Источник (получено 2025-12-13 07:13):

http://muff.kiev.ua/content/nagios-rikhtuem-sistemu-pod-sebya-sozdanie-komand-proverki

Ссылки:

[1]

http://muff.kiev.ua/content/nastroika-sistemy-monitoringa-nagios-3x-i-utility-konfigurirovaniya-nconf [2] http://www.yahoo.com