**Anexa 6**

Instalarea sistemelor de monitorizare Nagios

**Установка системы мониторинга NAGIOS**

**Содержание**

[1. Операционная система. 3](#_Toc405910920)

[2. Установка системы мониторинга и соответствующих компонентов. 4](#_Toc405910921)

[3. Настройка сервера системы мониторинга. 5](#_Toc405910922)

[3.1 Настройка apache24. 5](#_Toc405910923)

[3.2 Настройка Nagios. 6](#_Toc405910924)

[3.2.1 cfg.conf 6](#_Toc405910925)

[3.2.2 nagios.cfg 7](#_Toc405910926)

[3.2.3 resource.cfg 9](#_Toc405910927)

[3.2.4 commands.cfg 9](#_Toc405910928)

[3.2.5 contacts.cfg 12](#_Toc405910929)

[3.2.6 linuxhost.cfg 13](#_Toc405910930)

[3.2.7 winhost.cfg 16](#_Toc405910931)

[3.2.8 servicegroup.cfg 18](#_Toc405910932)

[3.2.9 nagiosgraph.conf 22](#_Toc405910933)

[3.3 Проверка после настройки Nagios 23](#_Toc405910934)

[4. Настройка клиентов. 23](#_Toc405910935)

[5. Изображение действующей системы мониторинга Nagios 24](#_Toc405910936)

#

# Операционная система.

Так как в организации используются сервера на операционных системах Windows и Unix (FreeBSD). Система мониторинга будет построена на системе Unix (FreeBSD 9.3).

 Операционная система FreeBSD установлена на виртуальной среде. Конфигурация 2хXeon процессора, 1024 MB оперативной памяти 40GB HDD пространства, 1 LAN карта (локальный адрес).

Установка происходила в стандартном режиме, через ISO образ. Во время установки был задан пароль root пользователя, создан новый пользователь и настроена сетевая карта (IP, Netmask, Broadcast, DNS)

После установки были выполнены все необходимые обновления. Во FreeBSD 9.1 и более поздние включена программа "самонастройки" ("bootstrap") pkgng. Она скачивает и устанавливает основную утилиту pkgng.

# /usr/sbin/pkg

Действующие инсталляции FreeBSD требуют преобразования базы данных установленных пакетов утилиты pkg\_install к новому формату. Для выполнения конвертирования, нужно запустить.

# pkg2ng

Обновление установленных пакетов при помощи pkgng

# pkg upgrade

Обновления безопасности системы.

# freebsd-update fetch

# freebsd-update install

Также был настроен ntp клиент, для синхронизации времени.

В системе были произведены мелкие настройки по визуализации, сменен основной редактор.

# Установка системы мониторинга и соответствующих компонентов.

Обновим бинарные исходники.

# pkg update

Так как Nagios использует веб интерфейс и часть интерфейса на php, установим apache c поддержкой php:

# pkg install apache24

# pkg install php5 mod\_php5

Для обработки cgi (другая часть интерфейса) установим fcgiwrap:

# pkg install fcgiwrap

Ставим Nagios:

# pkg install nagios

Ставим стандартные плагины для Nagios:

# pkg install nagios-plugins nagios-snmp-plugins nagios-snmp-plugins-extras

Для рисования графиков используем модуль nagiosgraph, так-же нам понадобится rrdtool - набор утилит для работы с RRD:

# pkg install rrdtool nagiosgraph

# Настройка сервера системы мониторинга.

## Настройка apache24.

Главный конфигурационный файл apache находиться по пути /usr/local/etc/apache24/http.conf.

Приводим его к соответствующему виду:

Для поддержки php, раскоментируем строчку

LoadModule php5\_module libexec/apache24/libphp5.so

Добавим в директиву ifModule

AddType application/x-httpd-php .php

AddType application/x-httpd-php-source .phps

В apache будем использовать виртуальные хосты (<http://domainame/virtulhost>), для этого раскоментируем строчку

Include etc/apache24/extra/httpd-vhosts.conf

Все остальные параметры остовляем по умолчанию.

Добавим виртуальные хосты для Nagios, /usr/local/etc/apache24/extra/httpd-vhost.conf

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

<Directory /usr/local/www/nagios/>

 Options None

 AllowOverride None

 AuthName "Nagios Access"

 AuthType Basic

 AuthUserFile /usr/local/etc/nagios/htpasswd.users

 Require valid-user

</Directory>

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

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

 Options ExecCGI

 AllowOverride None

 AuthName "Nagios Access"

 AuthType Basic

 AuthUserFile /usr/local/etc/nagios/htpasswd.users

 Require valid-user

</Directory>

Создадим файл паролей и заводим пользователя:

# /usr/local/sbin/htpasswd -c /usr/local/etc/nagios/htpasswd.users nagiosadmin

Указываем пароль, подтверждаем пароль.

Для добавления других пользователей параметр –c не используем.

Для поддержки php, переменуем файл php-fpm.conf

cp /usr/local/etc/php-fpm.conf.default php-fpm.conf

Оставим все по умолчанию.

Для запуска apache24 добавим в rc.conf

apache24\_enable=”YES”

## Настройка Nagios.

В папке /usr/local/etc/nagios/ находятся основные файлы:

**cgi.cfg** - этот файл в основном содержит разграничение прав доступа к системе через веб.

**nagios.cfg** - основной конфигурационный файл. Его переименовывать нельзя.

**resource.cfg** - В стандартном варианте он содержит пути к компонентам.

В папке /usr/local/etc/nagios/objects находятся файлы с описанием всех устройств которых будет обслуживать Nagios. Так как у нас будет больше чем одно устройство, а со временем их может увеличиться до десятков, файлы мы будем создавать, название файла – тип устройства.

**commands.cfg** - файл, в котором описаны команды вызова модулей и оповещений. При добавлении нового модуля его надо прописывать именно туда.

**contacts.cfg** – для того, чтобы получать оповещения о состоянии хостов и сервисов.

**linuxhost.cfg** – файл, в котором будут описаны все устройства на базе Linux-Unix.

**localhost.cfg** – файл, в котором описан сам сервер с Nagios.

**winhost.cfg** – файл, в котором будут описаны все устройства на базе Windows.

**servicegroup.cfg** – файл, для группировки сервисов. Для удобства просмотра определенного сервиса в который входят определенные хосты.

**templates.cfg** – файл, в котором описываются шаблоны. Шаблоны служат для оформления общей логики для таких сущностей как контакты, сервисы, хосты и т.д.

**timeperiods.cfg** – файл, в котором описывается временной период для оповещении о проблемах.

**upshost.cfg** – файл, в котором будут описаны все устройства UPS.

**switch.cfg** - файл, в котором будут описаны все устройства SWITCH.

**hostextinfo.cfg** – файл, в котором привязываются иконки для определенных хостов.

В папке /usr/local/etc/nagios /nagiosgraph находятся файлы, в которых описывается все что связано с графическими рисунками. Нас интересует два файла:

**nagiosgraph.conf** – основной файл.

**map** – файл, для отображения определенных параметров на рисунках.

**Приведем содержимое файлов к следующему виду:**

### cfg.conf

main\_config\_file=/usr/local/etc/nagios/nagios.cfg

physical\_html\_path=/usr/local/www/nagios

url\_html\_path=/nagios

show\_context\_help=0

use\_pending\_states=1

use\_authentication=1

use\_ssl\_authentication=0

authorized\_for\_system\_information=nagiosadmin

authorized\_for\_configuration\_information=nagiosadmin

authorized\_for\_system\_commands=nagiosadmin

authorized\_for\_all\_services=nagiosadmin

authorized\_for\_all\_hosts=nagiosadmin

authorized\_for\_all\_service\_commands=nagiosadmin

authorized\_for\_all\_host\_commands=nagiosadmin

default\_statusmap\_layout=5

default\_statuswrl\_layout=4

ping\_syntax=/sbin/ping -n -c 5 $HOSTADDRESS$

refresh\_rate=90

result\_limit=250

escape\_html\_tags=1

action\_url\_target=\_blank

notes\_url\_target=\_blank

lock\_author\_names=1

navbar\_search\_for\_addresses=1

navbar\_search\_for\_aliases=1

### nagios.cfg

log\_file=/var/spool/nagios/nagios.log

cfg\_file=/usr/local/etc/nagios/objects/commands.cfg

cfg\_file=/usr/local/etc/nagios/objects/contacts.cfg

cfg\_file=/usr/local/etc/nagios/objects/timeperiods.cfg

cfg\_file=/usr/local/etc/nagios/objects/templates.cfg

cfg\_file=/usr/local/etc/nagios/objects/localhost.cfg

cfg\_file=/usr/local/etc/nagios/objects/linuxhost.cfg

cfg\_file=/usr/local/etc/nagios/objects/winhost.cfg

cfg\_file=/usr/local/etc/nagios/objects/windows\_nsclient.cfg

cfg\_file=/usr/local/etc/nagios/objects/servicegroup.cfg

object\_cache\_file=/var/spool/nagios/objects.cache

precached\_object\_file=/var/spool/nagios/objects.precache

resource\_file=/usr/local/etc/nagios/resource.cfg

status\_file=/var/spool/nagios/status.dat

status\_update\_interval=10

nagios\_user=nagios

nagios\_group=nagios

check\_external\_commands=1

command\_file=/var/spool/nagios/rw/nagios.cmd

lock\_file=/var/spool/nagios/nagios.lock

temp\_file=/var/spool/nagios/nagios.tmp

temp\_path=/tmp

event\_broker\_options=-1

log\_rotation\_method=d

log\_archive\_path=/var/spool/nagios/archives

use\_syslog=1

log\_notifications=1

log\_service\_retries=1

log\_host\_retries=1

log\_event\_handlers=1

log\_initial\_states=0

log\_external\_commands=1

log\_passive\_checks=1

service\_inter\_check\_delay\_method=s

max\_service\_check\_spread=30

service\_interleave\_factor=s

host\_inter\_check\_delay\_method=s

max\_host\_check\_spread=30

max\_concurrent\_checks=0

check\_result\_reaper\_frequency=10

max\_check\_result\_reaper\_time=30

check\_result\_path=/var/spool/nagios/checkresults

max\_check\_result\_file\_age=3600

cached\_host\_check\_horizon=15

cached\_service\_check\_horizon=15

enable\_predictive\_host\_dependency\_checks=1

enable\_predictive\_service\_dependency\_checks=1

soft\_state\_dependencies=0

auto\_reschedule\_checks=0

auto\_rescheduling\_interval=30

auto\_rescheduling\_window=180

service\_check\_timeout=60

host\_check\_timeout=30

event\_handler\_timeout=30

notification\_timeout=30

ocsp\_timeout=5

perfdata\_timeout=5

retain\_state\_information=1

state\_retention\_file=/var/spool/nagios/retention.dat

retention\_update\_interval=60

use\_retained\_program\_state=1

use\_retained\_scheduling\_info=1

retained\_host\_attribute\_mask=0

retained\_service\_attribute\_mask=0

retained\_process\_host\_attribute\_mask=0

retained\_process\_service\_attribute\_mask=0

retained\_contact\_host\_attribute\_mask=0

retained\_contact\_service\_attribute\_mask=0

interval\_length=60

check\_for\_updates=1

bare\_update\_check=0

use\_aggressive\_host\_checking=0

execute\_service\_checks=1

accept\_passive\_service\_checks=1

execute\_host\_checks=1

accept\_passive\_host\_checks=1

enable\_notifications=1

enable\_event\_handlers=1

process\_performance\_data=1

service\_perfdata\_file=/var/spool/nagios/perfdata.log

service\_perfdata\_file\_template=$LASTSERVICECHECK$||$HOSTNAME$||$SERVICEDESC$||$SERVICEOUTPUT$||$SERVICEPERFDATA$

service\_perfdata\_file\_mode=a

service\_perfdata\_file\_processing\_interval=30

service\_perfdata\_file\_processing\_command=process-service-perfdata

obsess\_over\_services=0

obsess\_over\_hosts=0

translate\_passive\_host\_checks=0

passive\_host\_checks\_are\_soft=0

check\_for\_orphaned\_services=1

check\_for\_orphaned\_hosts=1

check\_service\_freshness=1

service\_freshness\_check\_interval=60

service\_check\_timeout\_state=c

check\_host\_freshness=0

host\_freshness\_check\_interval=60

additional\_freshness\_latency=15

enable\_flap\_detection=1

low\_service\_flap\_threshold=5.0

high\_service\_flap\_threshold=20.0

low\_host\_flap\_threshold=5.0

high\_host\_flap\_threshold=20.0

date\_format=us

illegal\_object\_name\_chars=`~!$%^&\*|'"<>?,()=

illegal\_macro\_output\_chars=`~$&|'"<>

use\_regexp\_matching=0

use\_true\_regexp\_matching=0

admin\_email=admin@urgenta.md

admin\_pager=admin@urgenta.md

daemon\_dumps\_core=0

use\_large\_installation\_tweaks=0

enable\_environment\_macros=0

debug\_level=0

debug\_verbosity=1

debug\_file=/var/spool/nagios/nagios.debug

max\_debug\_file\_size=1000000

allow\_empty\_hostgroup\_assignment=0

### resource.cfg

$USER1$=/usr/local/libexec/nagios

$USER7$=-C community -2

$USER8$=email

$USER9$=password

$USER10$=mail server

### commands.cfg

define command{

 command\_name notify-host-by-email

 command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/local/bin/sendEmail -s $USER10$ -xu $USER8$ -xp $USER9$ -t $CONTACTEMAIL$ -f $USER8$ -l /var/log/sendEmail -u "\*\* $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ \*\*" -m "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n"

 }

define command{

 command\_name notify-service-by-email

 command\_line /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$SERVICEOUTPUT$\n" | /usr/local/bin/sendEmail -s $USER10$ -xu $USER8$ -xp $USER9$ -t $CONTACTEMAIL$ -f $USER8$ -l /var/log/sendEmail -u "\*\* $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ \*\*" -m "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$"

 }

define command{

 command\_name check-host-alive

 command\_line $USER1$/check\_ping -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -p 5

 }

define command{

 command\_name check\_local\_disk

 command\_line $USER1$/check\_disk -w $ARG1$ -c $ARG2$ -p $ARG3$

 }

define command{

 command\_name check\_local\_load

 command\_line $USER1$/check\_load -w $ARG1$ -c $ARG2$

 }

define command{

 command\_name check\_local\_procs

 command\_line $USER1$/check\_procs -w $ARG1$ -c $ARG2$ -s $ARG3$

 }

define command{

 command\_name check\_local\_users

 command\_line $USER1$/check\_users -w $ARG1$ -c $ARG2$

 }

define command{

 command\_name check\_local\_swap

 command\_line $USER1$/check\_swap -w $ARG1$ -c $ARG2$

 }

define command{

 command\_name check\_local\_mrtgtraf

 command\_line $USER1$/check\_mrtgtraf -F $ARG1$ -a $ARG2$ -w $ARG3$ -c $ARG4$ -e $ARG5$

 }

define command{

 command\_name check\_ftp

 command\_line $USER1$/check\_ftp -H $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_hpjd

 command\_line $USER1$/check\_hpjd -H $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_snmp

 command\_line $USER1$/check\_snmp -H $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_http

 command\_line $USER1$/check\_http -I $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_ssh

 command\_line $USER1$/check\_ssh $ARG1$ $HOSTADDRESS$

 }

define command{

 command\_name check\_dhcp

 command\_line $USER1$/check\_dhcp $ARG1$

 }

define command{

 command\_name check\_ping

 command\_line $USER1$/check\_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5

 }

define command{

 command\_name check\_pop

 command\_line $USER1$/check\_pop -H $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_imap

 command\_line $USER1$/check\_imap -H $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_smtp

 command\_line $USER1$/check\_smtp -H $HOSTADDRESS$ $ARG1$

 }

define command{

 command\_name check\_tcp

 command\_line $USER1$/check\_tcp -H $HOSTADDRESS$ -p $ARG1$ $ARG2$

 }

define command{

 command\_name check\_udp

 command\_line $USER1$/check\_udp -H $HOSTADDRESS$ -p $ARG1$ $ARG2$

 }

define command{

 command\_name check\_nt

 command\_line $USER1$/check\_nt -H $HOSTADDRESS$ -p 12489 -s SED465i -v $ARG1$ $ARG2$

 }

define command {

 command\_name check\_snmp\_load\_v1

 command\_line $USER1$/check\_snmp\_load.pl -H $HOSTADDRESS$ $USER7$ -T $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$

}

define command {

 command\_name check\_snmp\_int\_v1

 command\_line $USER1$/check\_snmp\_int.pl -H $HOSTADDRESS$ $USER7$ -n $ARG1$ $ARG2$

}

define command{

 command\_name check\_snmp\_process\_v1

 command\_line $USER1$/check\_snmp\_process.pl -H $HOSTADDRESS$ $USER7$ -n $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$

}

define command{

 command\_name check\_snmp\_mem\_v1

 command\_line $USER1$/check\_snmp\_mem.pl -H $HOSTADDRESS$ $USER7$ $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$

}

define command{

 command\_name check\_disk\_snmp

 command\_line $USER1$/check\_disk\_snmp -H $HOSTADDRESS$ -s rourgenta -w 85% -c 90% -d /

}

define command{

 command\_name check\_snmp\_storage\_v1

 command\_line $USER1$/check\_snmp\_storage.pl -H $HOSTADDRESS$ $USER7$ -m $ARG1$ -w $ARG2$ -c $ARG3$ $ARG4$

}

define command{

 command\_name check\_disk\_snmp\_cache

 command\_line $USER1$/check\_disk\_snmp -H $HOSTADDRESS$ -s rourgenta -w 85% -c 90% -d /cache

}

define command{

 command\_name check\_disk\_snmp\_svn

 command\_line $USER1$/check\_disk\_snmp -H $HOSTADDRESS$ -s rourgenta -w 85% -c 90% -d /svn

}

define command{

 command\_name process-host-perfdata

 command\_line /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTATE$\t$HOSTATTEMPT$\t$HOSTSTATETYPE$\t$HOSTEXECUTIONTIME$\t$HOSTOUTPUT$\t$HOSTPERFDATA$\n" >> /var/spool/nagios/host-perfdata.out

 }

define command{

 command\_name process-service-perfdata

 command\_line /usr/bin/perl /usr/local/libexec/nagiosgraph/insert.pl

### contacts.cfg

define contact{

 contact\_name Tvetcov ; Short name of user

 use generic-contact ; Inherit default values from generic-contact template (defined above)

 alias Igori Tvetcov ; Full name of user

 email admin@urgenta.md ; <<\*\*\*\*\* CHANGE THIS TO YOUR EMAIL ADDRESS \*\*\*\*\*\*

 }

define contactgroup{

 contactgroup\_name admins

 alias Nagios Administrators

 members Tvetcov

 }

### linuxhost.cfg

define host{

 use remote-server

 host\_name proxy

 address xxx.xxx.xxx.xxx

}

define host{

 use remote-server

 host\_name dns

 address xxx.xxx.xxx.xxx

}

define hostgroup{

 hostgroup\_name linux-servers

 alias Remote Servers

 members proxy,dns

}

define service{

 use remote-service ; Name of service template to use

 hostgroups linux-servers

 service\_description PING

 action\_url /nagios/cgi-bin/nagiosgraph/show.cgi?host=$HOSTNAME$&service=$SERVICEDESC$&geom=450x150

 check\_command check\_ping!100.0,20%!500.0,60%

}

define service{

 use remote-service ; Name of service template to use

 hostgroups linux-servers

 service\_description SSH

 check\_command check\_ssh

 notifications\_enabled 0

}

define service{

 use remote-service ; Name of service template to use

 hostgroups linux-servers

 service\_description CPU Load

 check\_command check\_snmp\_load\_v1!netsl!4,3,3!8,5,5

}

define service{

 use remote-service ; Name of service template to use

 hostgroups linux-servers

 service\_description Check Disk /

 check\_command check\_disk\_snmp

}

define service{

 use remote-service ; Name of service template to use

 host\_name proxy

 service\_description Check Disk /cache

 check\_command check\_disk\_snmp\_cache

}

define service{

 use remote-service ; Name of service template to use

 host\_name dns

 service\_description Check Disk /svn

 check\_command check\_disk\_snmp\_svn

}

define service {

 use remote-service ; Name of service template to use

 host\_name proxy

 service\_description Lan Ethernet

 check\_command check\_snmp\_int\_v1!"em0"

}

define service {

 use remote-service ; Name of service template to use

 host\_name proxy

 service\_description Wan Ethernet

 check\_command check\_snmp\_int\_v1!"fxp0"

}

define service {

 use remote-service ; Name of service template to use

 host\_name dns

 service\_description Lan Ethernet

 check\_command check\_snmp\_int\_v1!"fxp0"

}

define service {

 use remote-service ; Name of service template to use

 host\_name dns

 service\_description Wan Ethernet

 check\_command check\_snmp\_int\_v1!"fxp1"

}

define service {

 use remote-service ; Name of service template to use

 host\_name proxy

 service\_description Check Squid

 check\_command check\_snmp\_process\_v1!squid!1,5!0!-2 -m 750,800 -u 90,99

}

define service {

 use remote-service ; Name of service template to use

 hostgroups linux-servers

 service\_description Check Memory

 check\_command check\_snmp\_mem\_v1!-N!95,60!99,90

Приведем содержимое файла localhost.cfg к следующему виду:

define host{

 use freebsd-server ; Name of host template to use

 ; This host definition will inherit all variables that are defined

 ; in (or inherited by) the freebsd-server host template definition.

 host\_name localhost

 alias localhost

 address xxx.xxx.xxx.xxx

 }

define hostgroup{

 hostgroup\_name freebsd-servers ; The name of the hostgroup

 alias FreeBSD Servers ; Long name of the group

 members localhost ; Comma separated list of hosts that belong to this group

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description PING

 check\_command check\_ping!100.0,20%!500.0,60%

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description Root Partition

 check\_command check\_local\_disk!20%!10%!/

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description Current Users

 check\_command check\_local\_users!20!50

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description Total Processes

 check\_command check\_local\_procs!250!400!RSZDT

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description Current Load

 check\_command check\_local\_load!5.0,4.0,3.0!10.0,6.0,4.0

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description Swap Usage

 check\_command check\_local\_swap!20!10

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description SSH

 check\_command check\_ssh

 notifications\_enabled 0

 }

define service{

 use local-service ; Name of service template to use

 hostgroups freebsd-servers

 service\_description HTTP

 check\_command check\_http

 notifications\_enabled 0

 }

### winhost.cfg

define host{

 use windows-server

 host\_name dc2.urgenta.md

 address xxx.xxx.xxx.xxx

}

define host{

 use windows-server

 host\_name dc.urgenta.md

 address xxx.xxx.xxx.xxx

}

define host{

 use windows-server

 host\_name imagistica

 address xxx.xxx.xxx.xxx

}

define host{

 use windows-server

 host\_name iis.urgenta.md

 address xxx.xxx.xxx.xxx

}

define host{

 use windows-server

 host\_name bd.urgenta.md

 address xxx.xxx.xxx.xxx

}

define host{

 use windows-server

 host\_name personal.cnsp.local

 address xxx.xxx.xxx.xxx

}

define host{

 use windows-server

 host\_name 903.samu.local

 address xxx.xxx.xxx.xxx

}

define hostgroup{

 hostgroup\_name windows-servers

 alias Windows Servers

 members dc2.urgenta.md, dc.urgenta.md, iis.urgenta.md, bd.urgenta.md, 903.samu.local, personal.cnsp.local, imagistica

}

define service{

 use generic-service ; Name of service template to use

 hostgroups windows-servers

 service\_description PING

 action\_url /nagios/cgi-bin/nagiosgraph/show.cgi?host=$HOSTNAME$&service=$SERVICEDESC$&geom=450x150

 check\_command check\_ping!100.0,20%!500.0,60%

}

define service{

 use generic-service

 hostgroups windows-servers

 service\_description Uptime

 check\_command check\_nt!UPTIME

 servicegroups UPTIME

}

define service{

 use generic-service

 hostgroups windows-servers

 service\_description NTload

 check\_command check\_nt!CPULOAD!-l 5,80,90

 servicegroups CPU

}

define service{

 use generic-service

 hostgroups windows-servers

 service\_description NTmem

 check\_command check\_nt!MEMUSE!-w 80 -c 90

 servicegroups MEMORY

}

define service{

 use generic-service

 hostgroups windows-servers

 service\_description NTdiskC

 check\_command check\_nt!USEDDISKSPACE!-l c -w 80 -c 90

 servicegroups DISK

}

define service{

 use generic-service

 host\_name imagistica

 service\_description D:\ Drive Space

 check\_command check\_nt!USEDDISKSPACE!-l d -w 80 -c 90

 servicegroups DISK

}

### servicegroup.cfg

define servicegroup{

 servicegroup\_name PING

 alias Ping check

}

Приведем содержимое файла templates.cfg к следующему виду:

define contact{

 name generic-contact ; The name of this contact template

 service\_notification\_period 24x7 ; service notifications can be sent anytime

 host\_notification\_period 24x7 ; host notifications can be sent anytime

 service\_notification\_options w,u,c,r,f,s ; send notifications for all service states, flapping events, and scheduled downtime events

 host\_notification\_options d,u,r,f,s ; send notifications for all host states, flapping events, and scheduled downtime events

 service\_notification\_commands notify-service-by-email ; send service notifications via email

 host\_notification\_commands notify-host-by-email ; send host notifications via email

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL CONTACT, JUST A 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

 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 program restarts

 notification\_period 24x7 ; Send host notifications at any time

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!

 }

define host{

 name freebsd-server ; The name of this host template

 use generic-host ; This template inherits other values from the generic-host template

 check\_period 24x7 ; By default, FreeBSD hosts are checked round the clock

 check\_interval 5 ; Actively check the host every 5 minutes

 retry\_interval 1 ; Schedule host check retries at 1 minute intervals

 max\_check\_attempts 10 ; Check each FreeBSD host 10 times (max)

 check\_command check-host-alive ; Default command to check FreeBSD hosts

 notification\_period workhours ; FreeBSD admins hate to be woken up, so we only notify during the day

 ; Note that the notification\_period variable is being overridden from

 ; the value that is inherited from the generic-host template!

 notification\_interval 120 ; Resend notifications every 2 hours

 notification\_options d,u,r ; Only send notifications for specific host states

 contact\_groups admins ; Notifications get sent to the admins by default

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!

 }

define host{

 name remote-server ; The name of this host template

 use generic-host ; This template inherits other values from the generic-host template

 check\_period 24x7 ; By default, FreeBSD hosts are checked round the clock

 check\_interval 5 ; Actively check the host every 5 minutes

 retry\_interval 1 ; Schedule host check retries at 1 minute intervals

 max\_check\_attempts 10 ; Check each FreeBSD host 10 times (max)

 check\_command check-host-alive ; Default command to check FreeBSD hosts

 notification\_period workhours ; FreeBSD admins hate to be woken up, so we only notify during the day

 ; Note that the notification\_period variable is being overridden from

 ; the value that is inherited from the generic-host template!

 notification\_interval 120 ; Resend notifications every 2 hours

 notification\_options d,u,r ; Only send notifications for specific host states

 contact\_groups admins ; Notifications get sent to the admins by default

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!

 }

define host{

 name windows-server ; The name of this host template

 use generic-host ; Inherit default values from the generic-host template

 check\_period 24x7 ; By default, Windows servers are monitored round the clock

 check\_interval 5 ; Actively check the server every 5 minutes

 retry\_interval 1 ; Schedule host check retries at 1 minute intervals

 max\_check\_attempts 10 ; Check each server 10 times (max)

 check\_command check-host-alive ; Default command to check if servers are "alive"

 notification\_period 24x7 ; Send notification out at any time - day or night

 notification\_interval 30 ; Resend notifications every 30 minutes

 notification\_options d,r ; Only send notifications for specific host states

 contact\_groups admins ; Notifications get sent to the admins by default

 hostgroups windows-servers ; Host groups that Windows servers should be a member of

 register 0 ; DONT REGISTER THIS - ITS JUST A TEMPLATE

 }

define host{

 name generic-printer ; The name of this host template

 use generic-host ; Inherit default values from the generic-host template

 check\_period 24x7 ; By default, printers are monitored round the clock

 check\_interval 5 ; Actively check the printer every 5 minutes

 retry\_interval 1 ; Schedule host check retries at 1 minute intervals

 max\_check\_attempts 10 ; Check each printer 10 times (max)

 check\_command check-host-alive ; Default command to check if printers are "alive"

 notification\_period workhours ; Printers are only used during the workday

 notification\_interval 30 ; Resend notifications every 30 minutes

 notification\_options d,r ; Only send notifications for specific host states

 contact\_groups admins ; Notifications get sent to the admins by default

 register 0 ; DONT REGISTER THIS - ITS JUST A TEMPLATE

 }

define host{

 name generic-switch ; The name of this host template

 use generic-host ; Inherit default values from the generic-host template

 check\_period 24x7 ; By default, switches are monitored round the clock

 check\_interval 5 ; Switches are checked every 5 minutes

 retry\_interval 1 ; Schedule host check retries at 1 minute intervals

 max\_check\_attempts 10 ; Check each switch 10 times (max)

 check\_command check-host-alive ; Default command to check if routers are "alive"

 notification\_period 24x7 ; Send notifications at any time

 notification\_interval 30 ; Resend notifications every 30 minutes

 notification\_options d,r ; Only send notifications for specific host states

 contact\_groups admins ; Notifications get sent to the admins by default

 register 0 ; DONT REGISTER THIS - ITS JUST A TEMPLATE

 }

define service{

 name generic-service ; The 'name' of this service template

 active\_checks\_enabled 1 ; Active service checks are enabled

 passive\_checks\_enabled 1 ; Passive service checks are enabled/accepted

 parallelize\_check 1 ; Active service checks should be parallelized (disabling this can lead to major performance problems)

 obsess\_over\_service 1 ; We should obsess over this service (if necessary)

 check\_freshness 0 ; Default is to NOT check service 'freshness'

 notifications\_enabled 1 ; Service notifications are enabled

 event\_handler\_enabled 1 ; Service event handler is enabled

 flap\_detection\_enabled 1 ; Flap detection 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 program restarts

 is\_volatile 0 ; The service is not volatile

 check\_period 24x7 ; The service can be checked at any time of the day

 max\_check\_attempts 3 ; Re-check the service up to 3 times in order to determine its final (hard) state

 normal\_check\_interval 10 ; Check the service every 10 minutes under normal conditions

 retry\_check\_interval 2 ; Re-check the service every two minutes until a hard state can be determined

 contact\_groups admins ; Notifications get sent out to everyone in the 'admins' group

 notification\_options w,u,c,r ; Send notifications about warning, unknown, critical, and recovery events

 notification\_interval 60 ; Re-notify about service problems every hour

 notification\_period 24x7 ; Notifications can be sent out at any time

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!

 }

define service{

 name ups-service ; The 'name' of this service template

 active\_checks\_enabled 1 ; Active service checks are enabled

 passive\_checks\_enabled 1 ; Passive service checks are enabled/accepted

 parallelize\_check 1 ; Active service checks should be parallelized (disabling this can lead to major performance problems)

 obsess\_over\_service 1 ; We should obsess over this service (if necessary)

 check\_freshness 0 ; Default is to NOT check service 'freshness'

 notifications\_enabled 1 ; Service notifications are enabled

 event\_handler\_enabled 1 ; Service event handler is enabled

 flap\_detection\_enabled 1 ; Flap detection 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 program restarts

 is\_volatile 0 ; The service is not volatile

 check\_period 24x7 ; The service can be checked at any time of the day

 max\_check\_attempts 3 ; Re-check the service up to 3 times in order to determine its final (hard) state

 normal\_check\_interval 15 ; Check the service every 10 minutes under normal conditions

 retry\_check\_interval 1 ; Re-check the service every two minutes until a hard state can be determined

 contact\_groups admins ; Notifications get sent out to everyone in the 'admins' group

 notification\_options w,c,r ; Send notifications about warning, unknown, critical, and recovery events

 notification\_interval 120 ; Re-notify about service problems every hour

 notification\_period 24x7 ; Notifications can be sent out at any time

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!

 }

define service{

 name local-service ; The name of this service template

 use generic-service ; Inherit default values from the generic-service definition

 max\_check\_attempts 4 ; Re-check the service up to 4 times in order to determine its final (hard) state

 normal\_check\_interval 5 ; Check the service every 5 minutes under normal conditions

 retry\_check\_interval 1 ; Re-check the service every minute until a hard state can be determined

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!

 }

define service{

 name remote-service ; The name of this service template

 use generic-service ; Inherit default values from the generic-service definition

 max\_check\_attempts 4 ; Re-check the service up to 4 times in order to determine its final (hard) state

 normal\_check\_interval 5 ; Check the service every 5 minutes under normal conditions

 retry\_check\_interval 1 ; Re-check the service every minute until a hard state can be determined

 register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!

 }

### nagiosgraph.conf

logfile = /var/spool/nagios/nagiosgraph.log

cgilogfile = /var/log/nagiosgraph-cgi.log

perflog = /var/spool/nagios/perfdata.log

rrddir = /var/spool/nagios/rrd

mapfile = /usr/local/etc/nagios/nagiosgraph/map

nagiosgraphcgiurl = /nagios/cgi-bin/nagiosgraph

nagioscgiurl = /nagios/cgi-bin

javascript = /nagios/stylesheets/nagiosgraph.js

stylesheet = /nagios/stylesheets/nagiosgraph.css

groupdb = /usr/local/etc/nagios/nagiosgraph/groupdb.conf

debug = 5

geometries = 650x50,800x100,1000x200,2000x100

colorscheme = 3

colors = 90d080,30a030,90c0e0,304090,ffc0ff,a050a0,ffc060,c07020

plotas = LINE2

plotasLINE1 = load5,data;load15,data

plotasLINE2 =

plotasLINE3 =

plotasAREA = idle,data;system,data;user,data;nice,data

plotasTICK =

stack = system,data;user,data;nice,data

lineformat = warn=LINE1,D0D050;crit=LINE1,D05050

timeall = day,week,month,year

timehost = day,week,month

timeservice = day,week,month

timegroup = day,week,month

expand\_timeall = day,week,month,year

expand\_timehost = week

expand\_timeservice = week

expand\_timegroup = day

timeformat\_now = %H:%M:%S %d %b %Y %Z

timeformat\_day = %H:%M %e %b

timeformat\_week = %e %b

timeformat\_month = Week %U

timeformat\_quarter = Week %U

timeformat\_year = %b %Y

showtitle = true

withmaximums = PING,HTTP

withminimums = PING,HTTP

heartbeat = 600

maximums = Current Load,.\*;Current Users,.\*;Total Processes,.\*;PLW,.\*

minimums =

lasts =

dbseparator = subdir

## Проверка после настройки Nagios

После всех манипуляций с настройкой Nagios, можно проверить правильность настроек командой /usr/local/bin/nagios -v /usr/local/etc/nagios/nagios.cfg

Для запуска Nagios добавим в файл rc.conf

nagios\_enable=”YES”

Запустим все сервисы

service apache24 start

service nagios start

Теперь можно открыть в любом браузере нашу страничку <http://dnsnameserver/nagios>

# Настройка клиентов.

Для Windows клиентов нам понадобится скачать и установить программу NSClient++. При установке указываем IP адрес сервера с установленным Nagios, указываем пароль для windows клиента и ставим галочки на против нужных модулей.

 Для Linux и других устройств будем использовать протокол SNMP. Для установки SNMP в FreeBSD воспользуемся коммандой:

# pkg install net-snmp

 Для настройки нам нужно указать community – имя, права, адрес.

rocommunity name ip address сервера с Nagios.

 SNMP работает по порту 161, так что если используется firewall необходимо открыть соответсвующий порт

# Изображение действующей системы мониторинга Nagios

