



**GERENCIAMENTO DE REDES**

**WAGNER BRAGA, LUZIA NOGUEIRA, ANIELSON RICARDO**

**REDES DE COMPUTADORES  
SEGURANÇA DA INFORMAÇÃO**

**2018**

O Nagios é uma ferramenta de gerenciamento de redes gratuita, que tem por objetivo facilitar o trabalho do gestor da rede diminuindo para o mesmo a necessidade de um constante monitoramento manual. Através do Nagios, é possível verificar nos agentes monitorados comportamentos que ocorreram, estão ocorrendo e, em alguns casos, prever comportamentos futuros.

Algumas das principais características do Nagios são:

- Detecção rápida de falha de infraestrutura;
- Os alertas podem ser entregues ao pessoal técnico via mensagens de diversos tipos como e-mail, SMS, alertas sonoros;
- Escalas para monitorar milhares de nós(hosts);
- Monitora serviços de rede (SMTP, POP3, HTTP, NNTP, ICMP, SNMP);
- Monitora recursos de computadores ou equipamentos de rede (carga do processador, uso de disco, logs do sistema) na maioria dos sistemas operacionais com suporte a rede, mesmo o Microsoft Windows com o plug-in NRPE\_NT;
- Monitoração remota suportada através de túneis criptografados SSH ou SSL;
- Suporte para implementação de monitoração redundante;
- Excelente interface web para visualização do atual status da rede, notificações, histórico de problemas, arquivos de log, etc.;
- Código Personalizado;
- Software Open Source;

Por essas características fica fácil visualizar a utilidade do Nagios e são muitas as vantagens de usá-lo, no entanto, também possui suas falhas. A seguir algumas vantagens e desvantagens da aplicação.

#### VANTAGENS:

- Age prontamente e executa ações de resolução sobre os problemas, notificando quando é detectado um problema e quando o mesmo é considerado resolvido, reduzindo o tempo de resposta em caso de problemas;
- Manutenção preventiva, já que o programa também pode emitir alerta sobre condições anteriores à falha dos serviços;
- Monitora toda sua infraestrutura de rede. Capacidade de monitorar vários servidores, agregando posteriormente numa única interface todos os parâmetros monitorados;
- Aumento da produtividade dos administradores e analistas, visto que monitora os ativos autonomamente, permitindo também adicionar novos plug-ins;
- Integração com plug-ins, que o próprio usuário pode escrever para facilitar e automatizar ainda mais a monitoração de elementos dentro da rede. Estes plug-ins podem ser desenvolvidos nas mais diversas linguagens de programação.

## DESVANTAGENS:

- Falta de recurso gráfico de configuração, exigindo um vasto conhecimento por parte do usuário dos fundamentos teóricos de redes de computadores;
- O processo de configuração costuma ser um tanto árduo, até o administrador adquirir uma maior intimidade com o processo;
- A interface Web não disponibiliza ferramentas de configuração do próprio sistema, sendo que todo o processo tem que ser realizado através da alteração manual dos ficheiros;
- Dificuldade na apresentação das informações sobre o estado de serviços de instalação, independente da conexão de rede, sendo impossível verificar se um serviço está indisponível por um problema no link ou por uma falha no software;
- Devido à bugs não corrigidos, apresenta limitações para se adaptar à rápida evolução das redes atuais.

Para instalarmos o Nagios e realizarmos o monitoramento, precisaremos, antes de tudo, realizar alguns passos de instalação para algumas aplicações e pacotes dos quais ele necessita. A seguir será descrito um tutorial de como realizar esta instalação e também de como fazer as configurações básicas.

## TUTORIAL

Para este trabalho utilizaremos como sistema operacional o Ubuntu Server 16.04 em uma máquina virtual. O Ubuntu Server não possui interface gráfica por padrão, isto o deixa mais leve e consideravelmente mais rápido, principalmente para o uso em máquinas virtuais. Utilizamos o Oracle Virtua Box, como programa de emulação para as MV's. Serão emulados dois dispositivos, um para ser o host servidor do Nagios e outro para ser o cliente a ser monitorado.

O Nagios necessita de alguns pré-requisitos, tais como, compilador GCC e alguns outros pacotes essenciais. Neste tutorial mostramos como instalar os pré-requisitos e como configurar os programas já instalados para o pleno funcionamento do Nagios.

## PRÉ-REQUISITOS:

Instalar os pré-requisitos é uma tarefa bastante simples onde rodaremos apenas um comando:

```
ubuntu@ubuntu-server:~$ sudo apt-get install wget build-essential apache2 php apache2-mod-php7.0 php-gd libgd-dev sendmail unzip
```

Para o Nagios funcionar é necessário criar um usuário para ele. Vamos criar um usuário “**nagios**” e um grupo chamado “**nagcmd**”, em seguida adicionaremos o usuário ao grupo.

```
ubuntu@ubuntu-server:~$ sudo useradd nagios
ubuntu@ubuntu-server:~$ sudo groupadd nagcmd
ubuntu@ubuntu-server:~$ sudo usermod -a -G nagcmd nagios
usermod: group 'nagcmd' does not exist
ubuntu@ubuntu-server:~$ sudo usermod -a -G nagcmd nagios
ubuntu@ubuntu-server:~$ sudo usermod -a -G nagios,nagcmd www-data
ubuntu@ubuntu-server:~$
```

Feito isto, vamos ao processo de instalação do Nagios propriamente dito. O primeiro de tudo é baixar o arquivo do “**Nagios core**”, utilizaremos a versão 4.2.0. Depois de baixado, o arquivo é extraído e entramos na pasta extraída.

```
ubuntu@ubuntu-server:~$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.0.tar.gz
--2018-05-26 16:25:23-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.0.tar.gz
Resolvendo assets.nagios.com (assets.nagios.com)... 72.14.181.71, 2600:3c00::f03c:91ff:fedf:b021
Conectando-se a assets.nagios.com (assets.nagios.com)[72.14.181.71]:443... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 11155104 (11M) [application/x-gzip]
Salvando em: "nagios-4.2.0.tar.gz"

nagios-4.2.0.tar.gz 100%[=====] 10,64M 398KB/s 1n 18s
2018-05-26 16:25:42 (600 KB/s) - "nagios-4.2.0.tar.gz" salvo [11155104/11155104]

ubuntu@ubuntu-server:~$
```

```
ubuntu@ubuntuserver:~$ tar -xzf nagios*.tar.gz
ubuntu@ubuntuserver:~$ cd nagios-4.2.0
ubuntu@ubuntuserver:~/nagios-4.2.0$
```

Agora vamos compilar os pacotes e realizar as primeiras configurações. É preciso observar que a compilação deve ser realizada com o usuário e grupo, criados para o Nagios.

```

ubuntu@ubuntu-server:~/nagios-4.2.0$ ./configure --with-nagios-group=nagios --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking forunistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
checking ctype.h usability... yes
checking ctype.h presence... yes
checking for ctype.h... yes
checking dirent.h usability... yes
checking dirent.h presence... yes
checking for dirent.h... yes
checking errno.h usability... yes
checking errno.h presence... yes
checking for errno.h... yes
checking fcntl.h usability... yes
checking fcntl.h presence... yes
checking for fcntl.h... yes
checking getopt.h usability... yes
checking getopt.h presence... yes
checking for getopt.h... yes
checking grp.h usability... yes
checking grp.h presence... yes
checking for grp.h... yes
checking for getpass.h... yes

```

Ao final será apresentado o seguinte resultado:

```

*** Configuration summary for nagios 4.2.0 08-01-2016 ***:

General Options:
-----
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: ${prefix}/var/nagios.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory: /etc/init.d
Apache conf.d directory: /etc/httpd/conf.d
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP):

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

```

Agora sim, utilizamos uma séria de comandos “**make**” para instalar o Nagios.

```

ubuntu@ubuntu:~/nagios-4.2.0$ sudo make all
cd ./base && make
make[1]: Entering directory '/home/ubuntu/nagios-4.2.0/base'
gcc -Wall -I. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o notifications.o notifications.c
notifications.c: In function 'service_notification':
notifications.c:214:3: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&mac.x[MACRO_SERVICENOTIFICATIONNUMBER], "%d", svc->current_notification_number);
   ^
notifications.c:220:3: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&mac.x[MACRO_SERVICENOTIFICATIONID], "%lu", svc->current_notification_id);
   ^
notifications.c: In function 'notify_contact_of_service':
notifications.c:781:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;CUSTOM ($SERVICESTATES);%s;%s;SERVICEOUTPUTS;$NOTIFICATIONAUTHORS;$NOTIFICATIONCOMMENTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:784:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;ACKNOWLEDGEMENT ($SERVICESTATES);%s;%s;SERVICEOUTPUTS;$NOTIFICATIONAUTHORS;$NOTIFICATIONCOMMENTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:787:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;FLAPPINGSTART ($SERVICESTATES);%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:790:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;FLAPPINGSTOP ($SERVICESTATES);%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:793:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;FLAPPINGDISABLED ($SERVICESTATES);%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:796:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;DOWNTIMESTART ($SERVICESTATES);%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:799:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;DOWNTIMEEND ($SERVICESTATES);%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:802:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;DOWNTIMECANCELLED ($SERVICESTATES);%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c:805:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "SERVICE NOTIFICATION: %s;%s;%s;SERVICESTATES;%s;%s;SERVICEOUTPUTS\n", cntct->name, svc->host_name, svc->description, command_name_ptr);
   ^
notifications.c: In function 'host_notification':
notifications.c:1171:3: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&mac.x[MACRO_HOSTNOTIFICATIONNUMBER], "%d", hst->current_notification_number);
   ^
notifications.c:1177:3: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&mac.x[MACRO_HOSTNOTIFICATIONID], "%lu", hst->current_notification_id);
   ^
notifications.c: In function 'notify_contact_of_host':
notifications.c:1678:6: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
   asprintf(&temp_buffer, "HOST NOTIFICATION: %s;%s;CUSTOM ($HOSTSTATES);%s;%s;HOSTOUTPUTS;$NOTIFICATIONAUTHORS;$NOTIFICATIONCOMMENTS\n", cntct->name, hst->name, command_name_ptr);
   ^

```

```

ubuntu@ubuntu:~/nagios-4.2.0$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ubuntu/nagios-4.2.0/base'
make install-basic
make[2]: Entering directory '/home/ubuntu/nagios-4.2.0/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios naglostats /usr/local/nagios/bin
make[2]: Leaving directory '/home/ubuntu/nagios-4.2.0/base'
make strip-post-install
make[2]: Entering directory '/home/ubuntu/nagios-4.2.0/base'
/usr/bin/strip /usr/local/nagios/bin/nagios
/usr/bin/strip /usr/local/nagios/bin/naglostats
make[2]: Leaving directory '/home/ubuntu/nagios-4.2.0/base'
make[1]: Leaving directory '/home/ubuntu/nagios-4.2.0/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ubuntu/nagios-4.2.0/cgi'
make install-basic
make[2]: Entering directory '/home/ubuntu/nagios-4.2.0/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ubuntu/nagios-4.2.0/cgi'
make strip-post-install
make[2]: Entering directory '/home/ubuntu/nagios-4.2.0/cgi'
for file in *.cgi; do \
    /usr/bin/strip /usr/local/nagios/sbin/$file; \
done
make[2]: Leaving directory '/home/ubuntu/nagios-4.2.0/cgi'
make[1]: Leaving directory '/home/ubuntu/nagios-4.2.0/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ubuntu/nagios-4.2.0/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes/rss
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes/rss/extlib

```



```
ubuntu@ubuntu-server:~/nagios-4.2.0$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

ubuntu@ubuntu-server:~/nagios-4.2.0$
```

```
ubuntu@ubuntu-server:~/nagios-4.2.0$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/init.d/nagios

*** Init script installed ***

ubuntu@ubuntu-server:~/nagios-4.2.0$
```

```

ubuntu@ubuntu-server:~/nagios-4.2.0$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files. You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

ubuntu@ubuntu-server:~/nagios-4.2.0$

```

E finalizamos fazendo as devidas cópias de diretórios para o diretório do Nagios dando as devidas permissões.

```

ubuntu@ubuntu-server:~/nagios-4.2.0$ sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
ubuntu@ubuntu-server:~/nagios-4.2.0$ sudo cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/
ubuntu@ubuntu-server:~/nagios-4.2.0$ sudo chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
ubuntu@ubuntu-server:~/nagios-4.2.0$

```

## PLUGINS:

Como foi dito o Nagios disponibiliza uma série de serviços de monitoramento e para que funcionem é necessário que seja feita a instalação de plugins. O primeiro passo é fazer o download dos pacotes e extraí-los para conseguirmos entrar no diretório desses plugins. O link para download é o seguinte: “<https://nagios-plugins.org/download/nagios-plugins-2.1.2.tar.gz>”; o comando para é extrair é o “**taz -xzf nome\_do\_arquivo** “. Depois de extraído entramos no diretório, então instalamos.

```
ubuntu@ubuntu-server:~/nagios-plugins-2.1.2$ sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl
[sudo] senha para ubuntu:
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
checking whether gcc and cc understand -c and -o together... yes
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking forunistd.h... yes
```

```
ubuntu@ubuntu-server:~/nagios-plugins-2.1.2$ sudo make
make all-recursive
make[1]: Entering directory '/home/ubuntu/nagios-plugins-2.1.2'
Making all in gl
make[2]: Entering directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
  cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/, $p' \
  < ../build-aux/snippet/c++defs.h \
  > c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^\.ifnndef/, $p' \
  < ../build-aux/snippet/warn-on-use.h \
  > warn-on-use.h-t && \
mv warn-on-use.h-t warn-on-use.h
rm -f arg-nonnull.h-t arg-nonnull.h && \
sed -n -e '/_GL_ARG_NONNULL/, $p' \
  < ../build-aux/snippet/arg-nonnull.h \
  > arg-nonnull.h-t && \
mv arg-nonnull.h-t arg-nonnull.h
/bin/mkdir -p arpa
rm -f arpa/inet.h-t arpa/inet.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
  sed -e 's|@'GUARD_PREFIX'@|GL|g' \
    -e 's|@'INCLUDE_NEXT'@|include_next|g' \
    -e 's|@'PRAGMA_SYSTEM_HEADER'@|#pragma GCC system_header|g' \
    -e 's|@'PRAGMA_COLUMNS'@|1|g' \
    -e 's|@'HAVE_FEATURES_H'@|1|g' \
    -e 's|@'NEXT_ARPA_INET_H'@|<arpa/inet.h>|g' \
    -e 's|@'HAVE_ARPA_INET_H'@|1|g' \
    -e 's|@'GNULIB_INET_NTOP'@|1|g' \
    -e 's|@'GNULIB_INET_PTON'@|0|g' \
    -e 's|@'HAVE_DECL_INET_NTOP'@|1|g' \
```

```
ubuntu@ubuntu-server:~/nagios-plugins-2.1.2$ sudo make install
Making install in gl
make[1]: Entering directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
make install-recursive
make[2]: Entering directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
make[3]: Entering directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
make[4]: Entering directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
if test yes = no; then \
  case 'linux-gnu' in \
    darwin[56]*) \
      need_charset_alias=true ;; \
    darwin* | cygwin* | mingw* | pw32* | cegcc*) \
      need_charset_alias=false ;; \
    *) \
      need_charset_alias=true ;; \
  esac ; \
else \
  need_charset_alias=false ; \
fi ; \
if $need_charset_alias; then \
  /bin/bash ../build-aux/mkinstalldirs /usr/local/nagios/lib ; \
fi ; \
if test -f /usr/local/nagios/lib/charset.alias; then \
  sed -f ref-add.sed /usr/local/nagios/lib/charset.alias > /usr/local/nagios/lib/charset.tmp ; \
  /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
  rm -f /usr/local/nagios/lib/charset.tmp ; \
else \
  if $need_charset_alias; then \
    sed -f ref-add.sed charset.alias > /usr/local/nagios/lib/charset.tmp ; \
    /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
    rm -f /usr/local/nagios/lib/charset.tmp ; \
  fi ; \
fi
make[4]: Nothing to be done for 'install-data-am'.
make[4]: Leaving directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
make[3]: Leaving directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
make[2]: Leaving directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
make[1]: Leaving directory '/home/ubuntu/nagios-plugins-2.1.2/gl'
Making install in tar
```

## CONFIGURAÇÃO:

Depois da fase de instalação a configuração padrão do Nagios estará no arquivo “**nagios.cfg**”.

```
ubuntu@ubuntu-server:~$ sudo nano -c /usr/local/nagios/etc/nagios.cfg
ubuntu@ubuntu-server:~$
```

Descomente uma linha em primeiro momento como na imagem abaixo. Mais precisamente a linha 51 “**cfg\_dir=/usr/local/nagios/etc/servers**”.

```
GNU nano 2.5.3                               Arquivo: /usr/local/nagios/etc/nagios.cfg
# if you wish (as shown below), or keep them all in a single config file.
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
# Definitions for monitoring a router/switch
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg
# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:
cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CGIs read object definitions from
# this cache file (rather than looking at the object config files
# directly) in order to prevent inconsistencies that can occur
# when the config files are modified after Nagios starts.
object_cache_file=/usr/local/nagios/var/objects.cache
# PRE-CACHED OBJECT FILE
# This options determines the location of the precached object file.
# If you run Nagios with the -p command line option, it will preprocess
# your object configuration file(s) and write the cached config to this
lin 51/1349 (3%), col 1/38 (2%), carac 1757/44830 (3%)
```

No arquivo “**contacts.cfg**” podemos configurar o contato do Nagios substituindo o e-mail padrão pelo nosso próprio e-mail.

```
GNU nano 2.5.3                               Arquivo: /usr/local/nagios/etc/objects/contacts.cfg
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.
define contact{
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact      ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin         ; Full name of user
    email             wagnerbragacrf@alu.ufc.br ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****>>
}
#####
#####
#
# CONTACT GROUPS
#
#####
#####
# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.
define contactgroup{
    contactgroup_name admins
    alias             Nagios Administrators
    members           nagiosadmin
}
```

## CONFIGURAÇÃO DO APACHE:

Como o programa do Nagios é acessado via Web é necessário a configuração do mesmo no Apache. Assim vamos ativar os módulos no apache e o nosso virtualhost Nagios.

```
Arquivo Editar Ver Pesquisar Terminar Ajuda
ubuntu@ubuntu-server:~$ sudo a2enmod rewrite
```

```
Arquivo Editar Ver Pesquisar Terminar Ajuda
ubuntu@ubuntu-server:~$ sudo a2enmod cgi
```

Adicionamos um usuário administrador para a interface Web do Nagios.

```
Arquivo Editar Ver Pesquisar Terminar Ajuda
ubuntu@ubuntu-server:~$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
ubuntu@ubuntu-server:~$
```

Agora vamos ativar o virtualhost Nagios, como dito antes.

```
ubuntu@ubuntu-server:~$ sudo ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/  
ubuntu@ubuntu-server:~$
```

Basta agora que reiniciemos o serviço do apache e iniciemos o do Nagios.

```
ubuntu@ubuntu-server:~$ service apache2 restart  
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====  
Authentication is required to restart 'apache2.service'.  
Authenticating as: ubuntu server 1604,,, (ubuntu)  
Password:  
==== AUTHENTICATION COMPLETE ====  
ubuntu@ubuntu-server:~$
```



```
ubuntu@ubuntuserver:/etc/init.d$ service nagios start
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'nagios.service'.
Authenticating as: ubuntu server 1604,,, (ubuntu)
Password:
==== AUTHENTICATION COMPLETE ====
ubuntu@ubuntuserver:/etc/init.d$
```

Caso apresente erro para iniciar o serviço do Nagios basta seguir o seguinte procedimento:

```
ubuntu@ubuntuserver:~$ sudo cp /etc/init.d/skeleton /etc/init.d/nagios
```

Abrimos o arquivo “**/etc/init.d/nagios**” e deixamos com o exemplo abaixo:

```
GNU nano 2.5.3                               Arquivo: nagios

#!/bin/sh
# kFreeBSD do not accept scripts as interpreters, using #!/bin/sh and sourcing.
if [ true != "$INIT_D_SCRIPT_SOURCED" ] ; then
    set "$0" "$@"; INIT_D_SCRIPT_SOURCED=true . /lib/init/init-d-script
fi
### BEGIN INIT INFO
# Provides:          skeleton
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Example initscript
# Description:       This file should be used to construct scripts to be
#                   placed in /etc/init.d. This example start a
#                   single forking daemon capable of writing a pid
#                   file. To get other behaviours, implement
#                   do_start(), do_stop() or other functions to
#                   override the defaults in /lib/init/init-d-script.
### END INIT INFO

# Author: Foo Bar <foobar@baz.org>
#
# Please remove the "Author" lines above and replace them
# with your own name if you copy and modify this script.

DESC="Description of the service"
DAEMON=/usr/sbin/daemonexecutablename
DESC = "Nagios"
NOME = nagios
DAEMON = /usr / local / nagios / bin / $ NOME
DAEMON_ARGS = "- d /usr/local/nagios/etc/nagios.cfg"
PIDFILE = /usr / local / nagios / var / $ NAME.lock
```

Por último basta deixar o arquivo como executável através do comando “**chmod +x**”.  
Reiniciamos o apache novamente e iniciamos o Nagios.

Pronto. O Nagios Web já funcionará.