Don Bosco Institute of Technology, Mumbai 400070
Department of Information Technology

Experiment 9

Name: Simar Kaur
Roll NO: 16
Adv Dev OPs lab
Date: 16 September 2022

Aim: To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Theory:
Definition:
Nagios is a popular monitoring tool many DevOps teams use to ensure thorough and efficient tracking of systems, devices, apps, and services.
Nagios runs periodic checks on critical thresholds and metrics to monitor for system changes and potential problems. If the software runs into an issue, the tool notifies admins and can also run automated scripts to contain and remedy the situation.
The tool allows users to track the state and performance of:

- **Hardware (routers, switches, firewalls, dedicated server, workstations, printers, etc.).**
- **Networks.**
- **Apps.**
- **Services.**
- **Business processes.**
- **Operating systems (Windows, Linux, Unix, and OSX).**

What does it monitor?

- **Memory and disk usage.**
- **CPU loads.**
- **The number of running processes.**
- **Log files.**

- System availability.
- Response Time
- URL and content monitoring metrics.
- Services and network protocols (SMTP, POP3, HTTP, etc.).

Benefits:

- Nagios plugins, which are free to download and develop, provide high levels of flexibility.
- The solution's alerting and response systems ensure high levels of security.
- Fast detection of outages, website defacement, and hijacking attempts.
- Capacity planning helps plan future IT upgrades.
- As the tool is open source, skilled teams can adjust the software to fit business needs.

Features:

- A robust log management system.
- Parallel processing that ensures fast detection of outages and hardware issues.
- Allows you to set up monitoring on machines across multiple locations. All devices then send outputs to the central Nagios server.
- Uses topology to determine dependencies.
- Allows users to define the hierarchy of networks using parent hosts.
- Support for both agent-based and agentless configurations. Nagios Remote Plugin Executor (NRPE) agent allows you to run remote scripts and plugins.
- A highly customizable UI with multi-tenant capabilities that allow you to assign custom visibility to each user.
- Multiple APIs enable simple integration with both in-house and third-party apps.
- An archive of configuration snapshots enables you to revert to a previous working setup if the system runs into a serious problem.

Nagios Plugins:

- **Official plugins: Nagios develops and maintains around 50 official add-ons.**
- **Community plugins: There are over 3,000 available third-party plugins developed by community members.**
- **Custom add-ons: If your team has the right skills, the open-source nature of the tool allows you to create custom plugins.**

All plugins follow a status code. The table below explains the default exit code statuses and their descriptions:

EXIT CODE	STATUS	DESCRIPTION
0	OK	The system is working fine
1	WARNING	The system continues to operate but requires attention
2	CRITICAL	The system is not working correctly
3	UNKNOWN	The plugin cannot assess the status of the host or service

Nagios GUI:

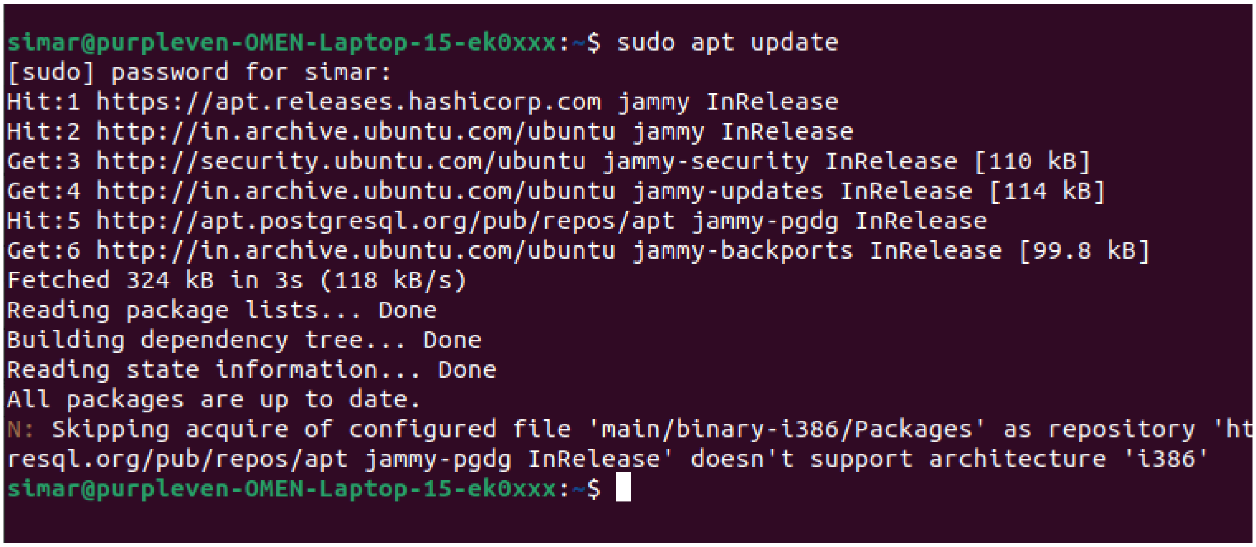
The Nagios XI GUI provides viewing of the most critical monitoring data and has six submenus:

- **Quick view: Quick view is where you find the Tactical Overview, an overall summary of all hosts and services you are currently monitoring.**
- **Details: This menu displays detailed summaries of all hosts and services. The Details tab shows all statistics, commands, and services running for each host.**
- **Graphs: This menu shows a series of scalable performance graphs for all monitored services for each host.**
- **Maps: A graphical status map of all hosts and parent-child relationships. You can view the map as a balloon or treemap.**
- **Incident management: This menu summarizes all current notifications and alerts for services and hosts.**
- **Monitoring process: This menu displays data about the tool's internal operations.**

Output:

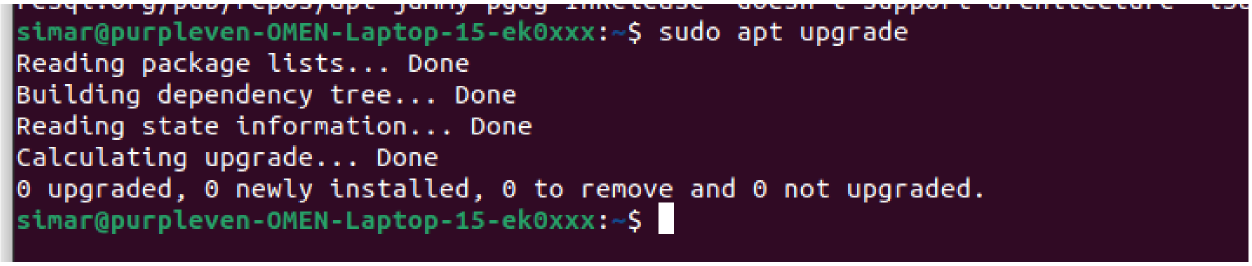
Installation:

1. You should first check the web for the latest available packages:



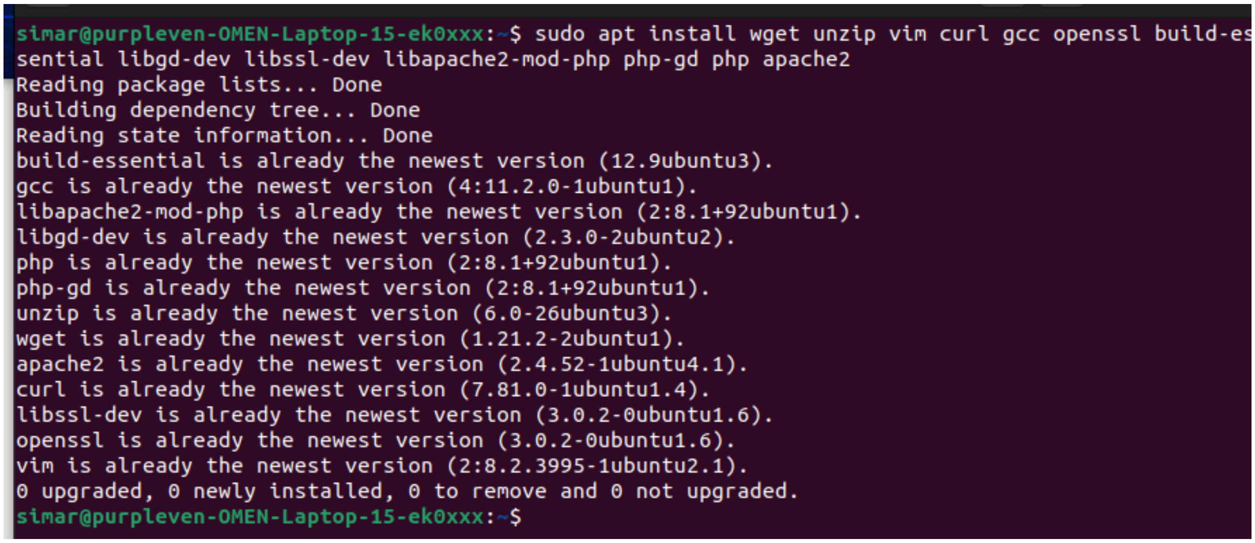
```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo apt update
[sudo] password for simar:
Hit:1 https://apt.releases.hashicorp.com jammy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Hit:5 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Fetched 324 kB in 3s (118 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository 'ht
resql.org/pub/repos/apt jammy-pgdg InRelease' doesn't support architecture 'i386'
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$
```

2. Next, upgrade the system packages to the latest versions:



```
resql.org/pub/repos/apt jammy-pgdg InRelease' doesn't support architecture 'i3
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$
```

After updating the system, you need to install the packages required to run Core.

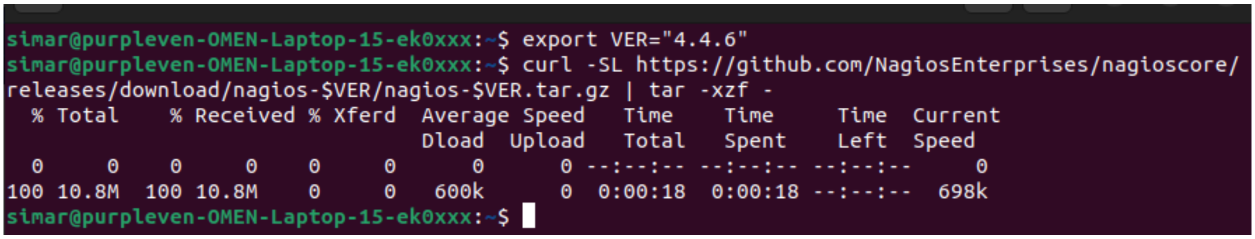


```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo apt install wget unzip vim curl gcc openssl build-es
sential libgd-dev libssl-dev libapache2-mod-php php-gd php apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
gcc is already the newest version (4:11.2.0-1ubuntu1).
libapache2-mod-php is already the newest version (2:8.1+92ubuntu1).
libgd-dev is already the newest version (2.3.0-2ubuntu2).
php is already the newest version (2:8.1+92ubuntu1).
php-gd is already the newest version (2:8.1+92ubuntu1).
unzip is already the newest version (6.0-26ubuntu3).
wget is already the newest version (1.21.2-2ubuntu1).
apache2 is already the newest version (2.4.52-1ubuntu4.1).
curl is already the newest version (7.81.0-1ubuntu1.4).
libssl-dev is already the newest version (3.0.2-0ubuntu1.6).
openssl is already the newest version (3.0.2-0ubuntu1.6).
vim is already the newest version (2:8.2.3995-1ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$
```

Download Nagios Core:

To do so, browse to the official git repository and select the latest release. Alternatively, you can download the tool from the official website.

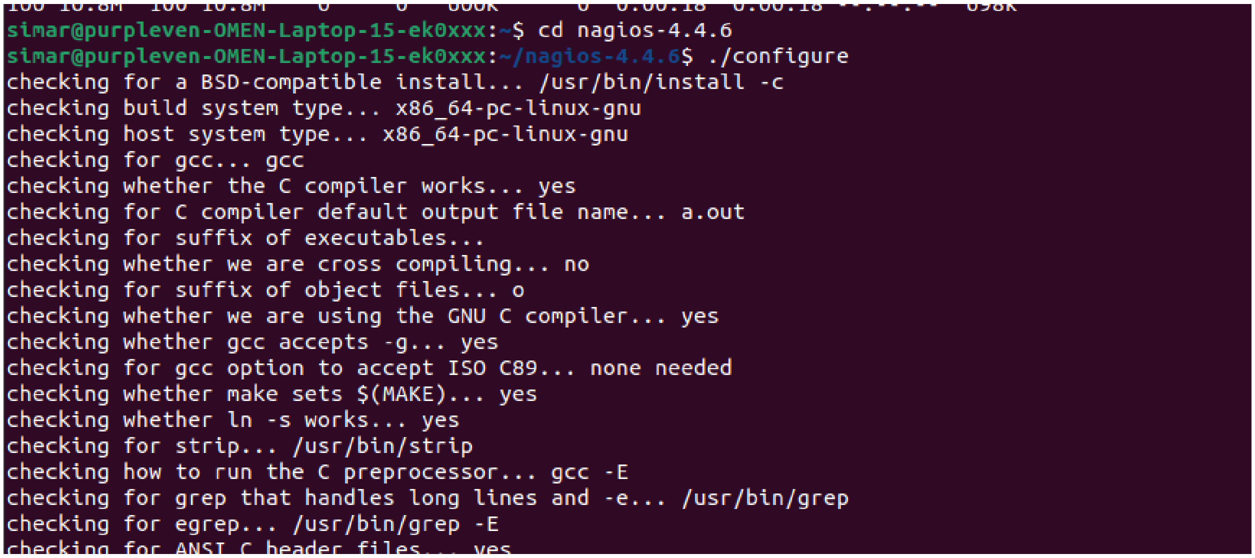
And use the curl command: This command downloads a directory called nagios-4.4.6 and adds it to your current working directory.



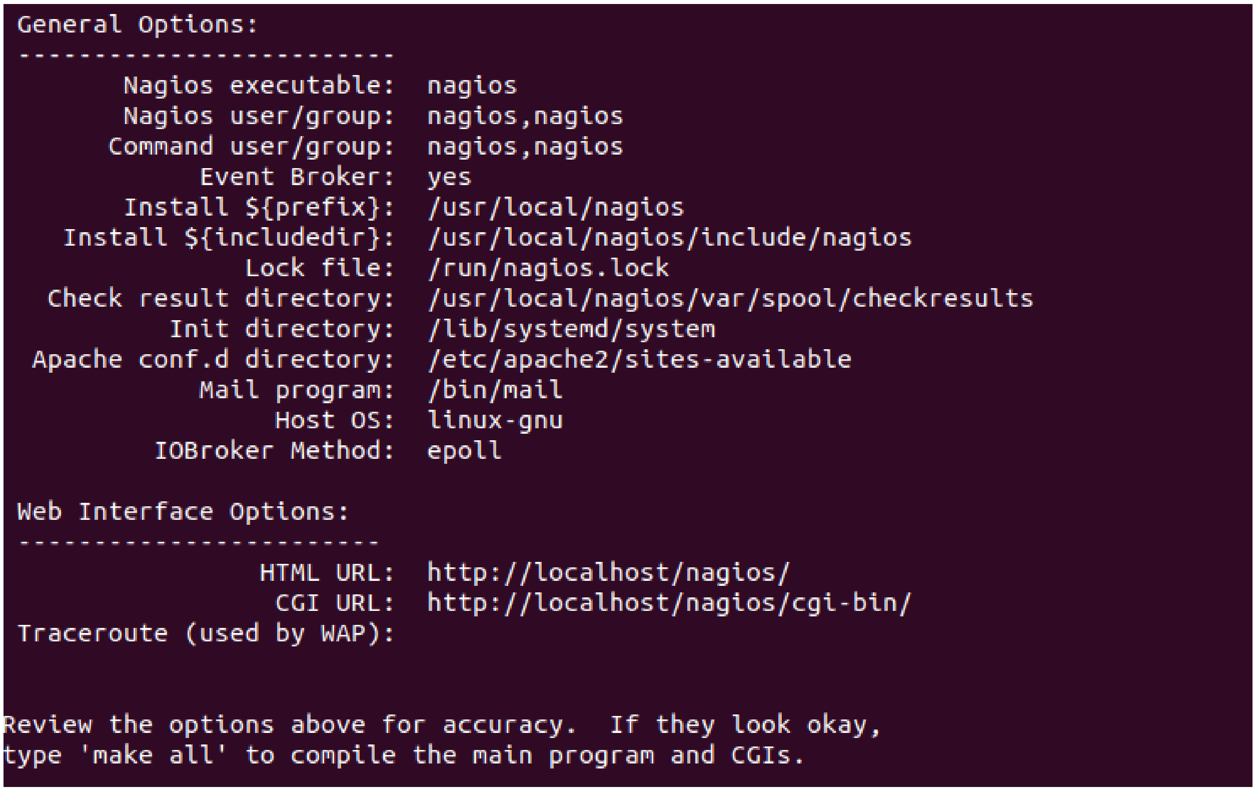
```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$ export VER="4.4.6"
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$ curl -SL https://github.com/NagiosEnterprises/nagioscore/
releases/download/nagios-$VER/nagios-$VER.tar.gz | tar -xzf -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 10.8M  100 10.8M    0     0   600k      0  0:00:18  0:00:18 --:--:--  698k
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$
```

Install Nagios:

Navigate into the Nagios directory and run the configure script:

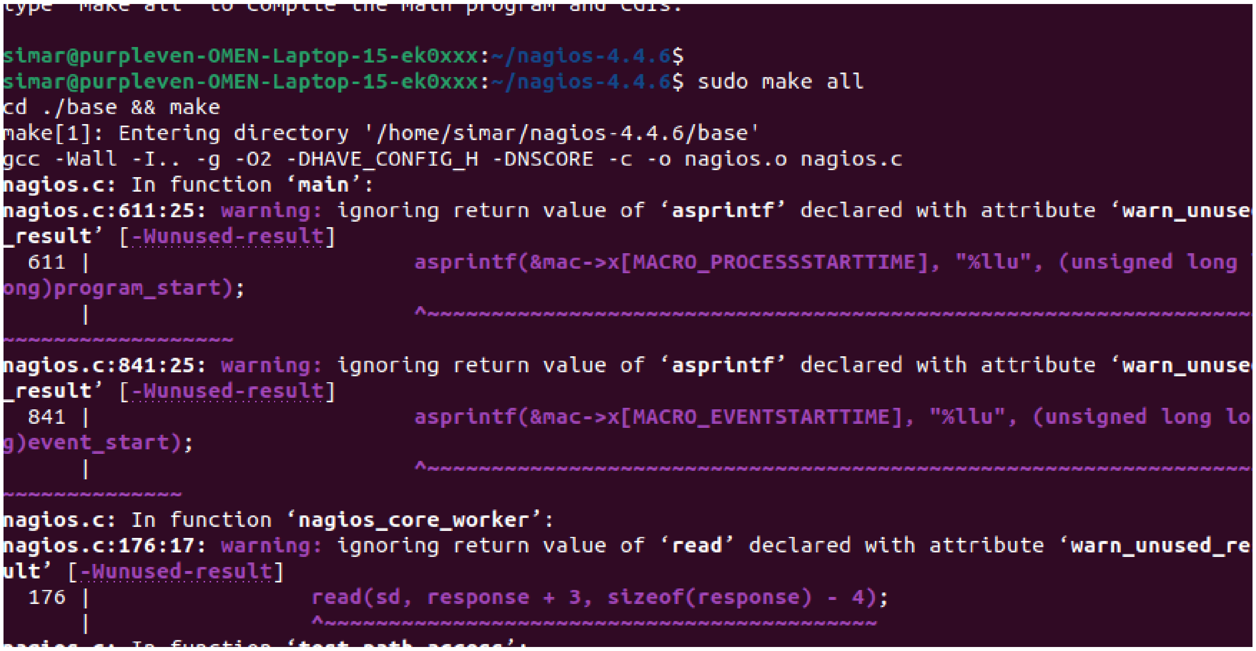


```
100 10.8M  100 10.8M    0     0   606k      0  0:00:18  0:00:18 --:--:--  698k
simar@purpleven-OMEN-Laptop-15-ek0xxx:~$ cd nagios-4.4.6
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-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 whether ln -s works... 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... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
```



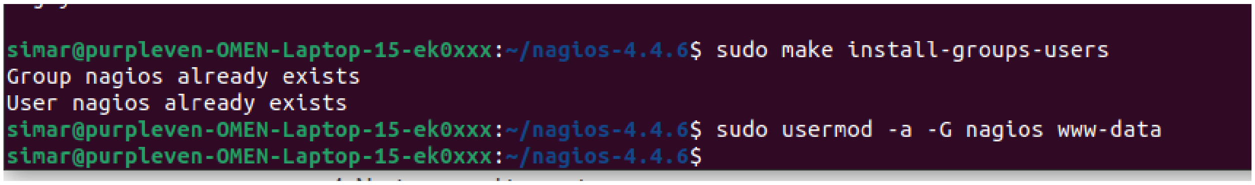
```
 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagios
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  /run/nagios.lock
   Check result directory:  /usr/local/nagios/var/spool/checkresults
           Init directory:  /lib/systemd/system
  Apache conf.d directory:  /etc/apache2/sites-available
             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.
```

Run the make all command to compile the program alongside the CGIs:



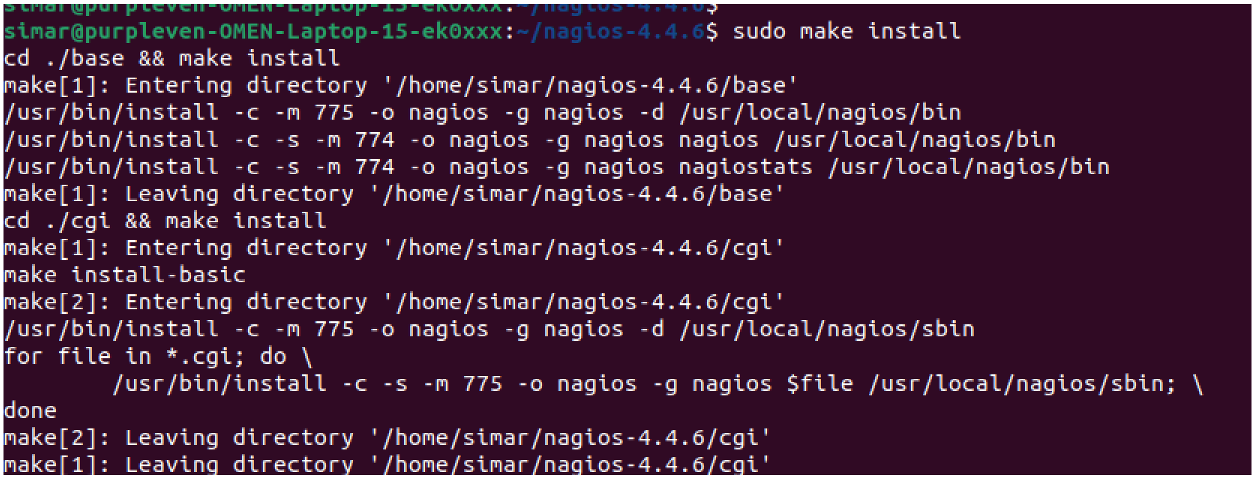
```
type 'make all' to compile the main program and CGIs.

simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo make all
cd ./base && make
make[1]: Entering directory '/home/simar/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
nagios.c: In function ‘main’:
nagios.c:611:25: warning: ignoring return value of ‘asprintf’ declared with attribute ‘warn_unuse
_result’ [-Wunused-result]
  611 |                         asprintf(&mac->x[MACRO_PROCESSSTARTTIME], "%llu", (unsigned long 
ong)program_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~
nagios.c:841:25: warning: ignoring return value of ‘asprintf’ declared with attribute ‘warn_unuse
_result’ [-Wunused-result]
  841 |                         asprintf(&mac->x[MACRO_EVENTSTARTTIME], "%llu", (unsigned long lo
g)event_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~
nagios.c: In function ‘nagios_core_worker’:
nagios.c:176:17: warning: ignoring return value of ‘read’ declared with attribute ‘warn_unused_re
ult’ [-Wunused-result]
  176 |                 read(sd, response + 3, sizeof(response) - 4);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c: In function ‘test_path_access’:
```

Next, we need to create group users:

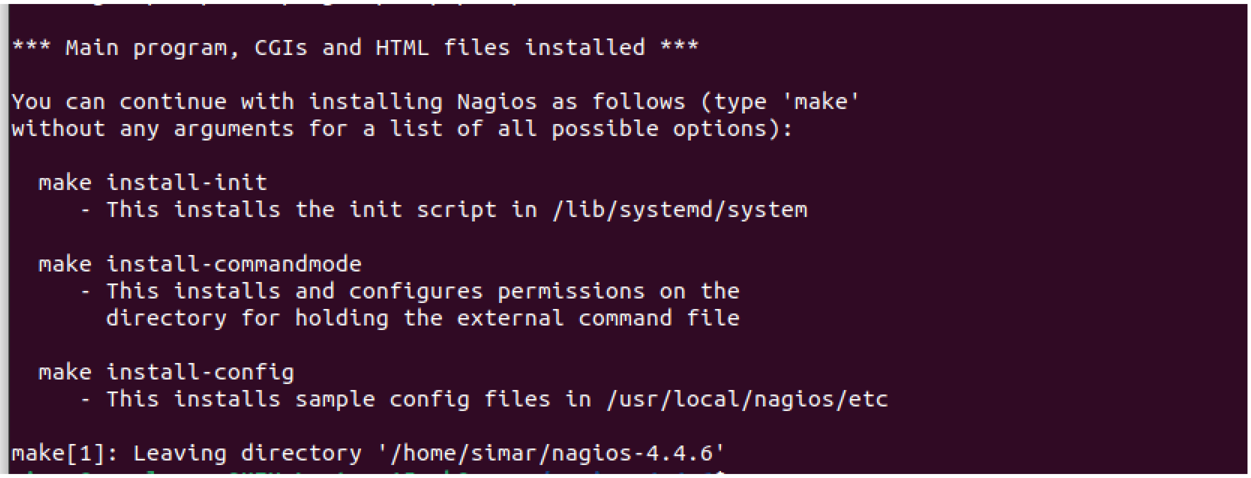


```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo make install-groups-users
Group nagios already exists
User nagios already exists
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo usermod -a -G nagios www-data
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
```

Now install Nagios Core on your Ubuntu system:

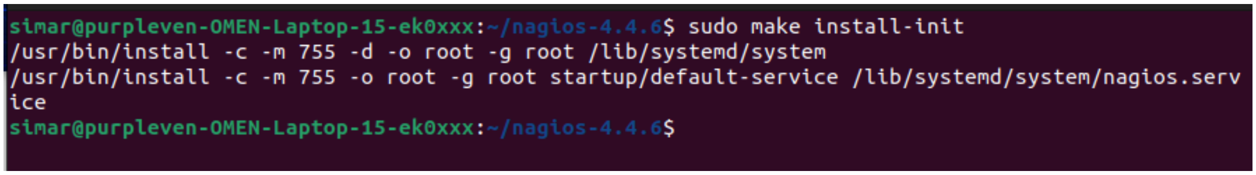


```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/simar/nagios-4.4.6/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/simar/nagios-4.4.6/base'
cd ./cgi && make install
make[1]: Entering directory '/home/simar/nagios-4.4.6/cgi'
make install-basic
make[2]: Entering directory '/home/simar/nagios-4.4.6/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 -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/simar/nagios-4.4.6/cgi'
make[1]: Leaving directory '/home/simar/nagios-4.4.6/cgi'
```



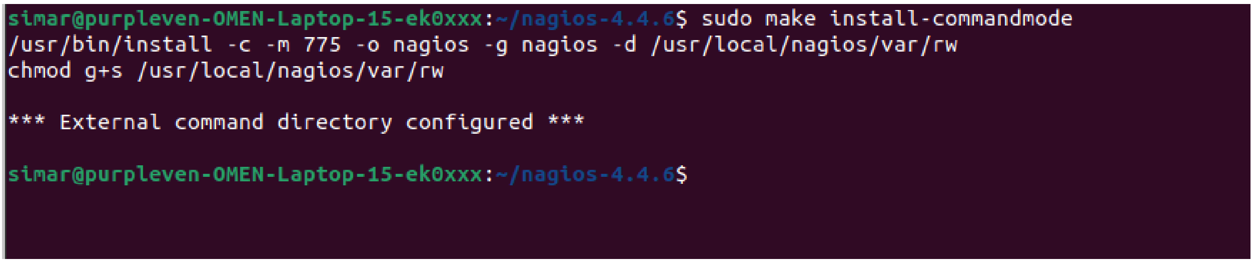
```
*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc

make[1]: Leaving directory '/home/simar/nagios-4.4.6'
```

As you can see, some additional instructions appear on the screen. Run the following command to install the init script in the */lib/systemd/system* path:



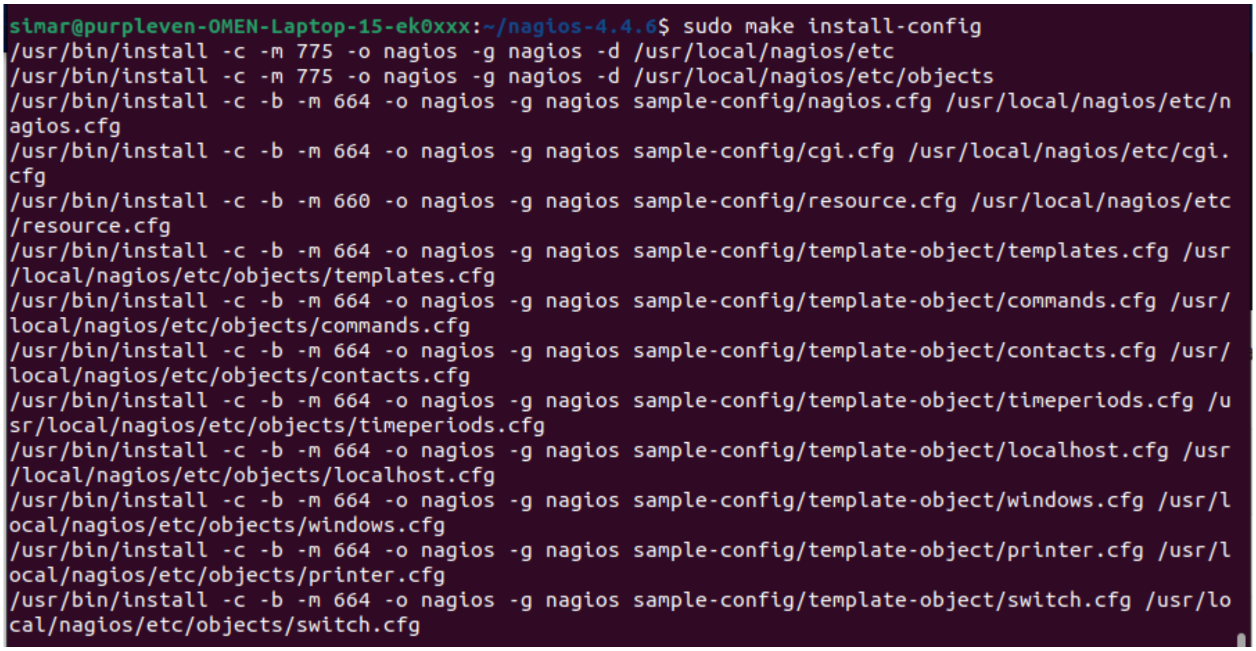
```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.serv
ice
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
```

Next, install and configure permissions on the directory:



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
```

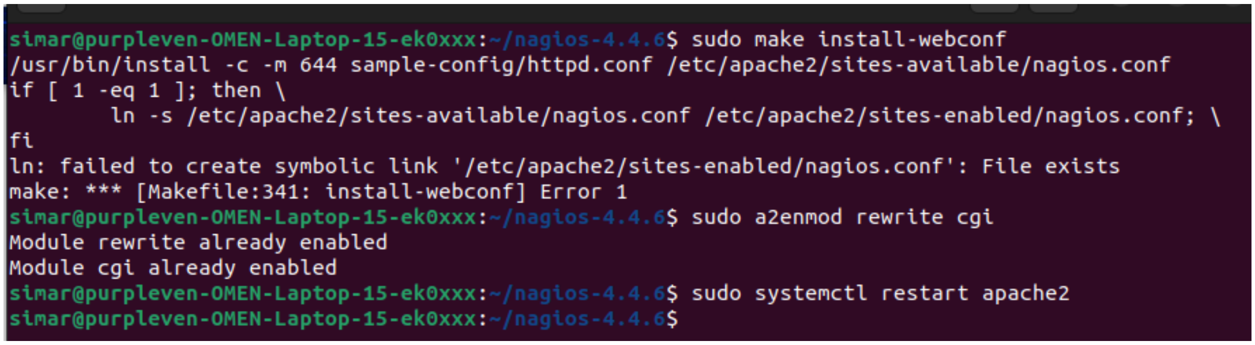
Finally, install sample config files in */usr/local/nagios/etc/*:



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ 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/n
agios.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 /u
sr/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/l
ocal/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/l
ocal/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/lo
cal/nagios/etc/objects/switch.cfg
```

Set up Apache and Nagios UI

You need to enable the Apache module required for the Nagios web interface, so run the following command:



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
if [ 1 -eq 1 ]; then \
        ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
ln: failed to create symbolic link '/etc/apache2/sites-enabled/nagios.conf': File exists
make: *** [Makefile:341: install-webconf] Error 1
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo a2enmod rewrite cgi
Module rewrite already enabled
Module cgi already enabled
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo systemctl restart apache2
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
```

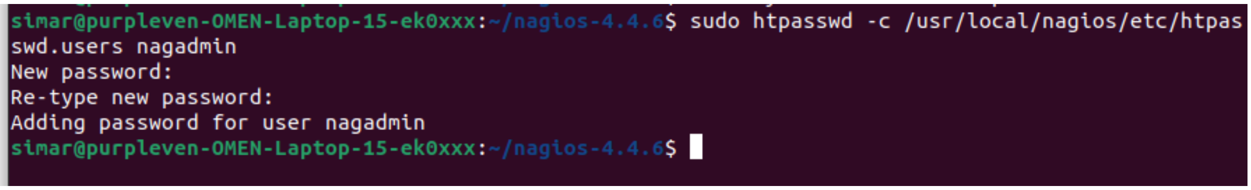
Type in the following command for the classic Nagios monitoring theme:

```
$ sudo make install-classicui
```

Create the First Nagios User

We now need to create a user that can log in to Nagios. The following command creates a user called *nagadmin*:

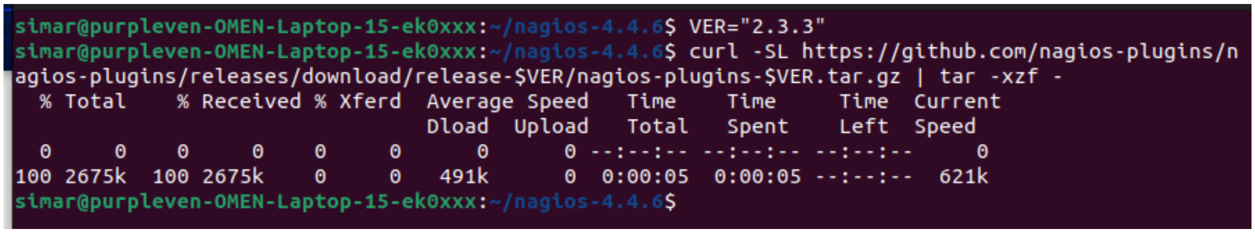
You need to provide a password for the user and confirm it (by default, passwords are stored in */usr/local/nagios/etc/htpasswd.users*).



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ sudo htpasswd -c /usr/local/nagios/etc/htpas
swd.users nagadmin
New password:
Re-type new password:
Adding password for user nagadmin
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
```

Install Nagios Plugins

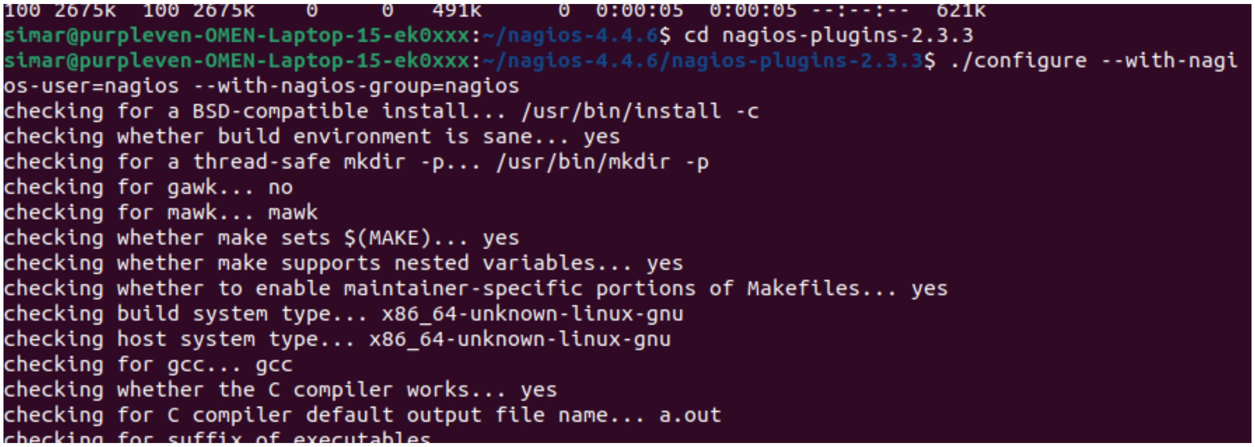
To download plugins, type the following command:



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ VER="2.3.3"
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ curl -SL https://github.com/nagios-plugins/n
agios-plugins/releases/download/release-$VER/nagios-plugins-$VER.tar.gz | tar -xzf -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 2675k  100 2675k    0     0   491k      0  0:00:05  0:00:05 --:--:--  621k
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$
```

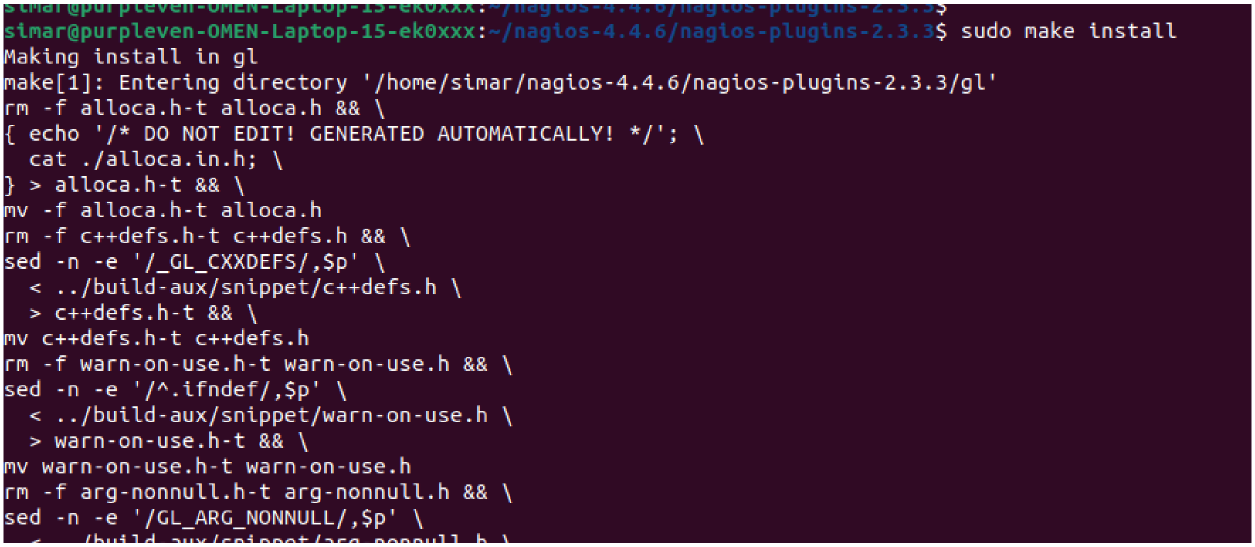
This command creates a new directory (*nagios-plugins-2.3.3*) in your current working directory. To install plugins, you first need to navigate to the new directory:

Now compile the plugins from source:

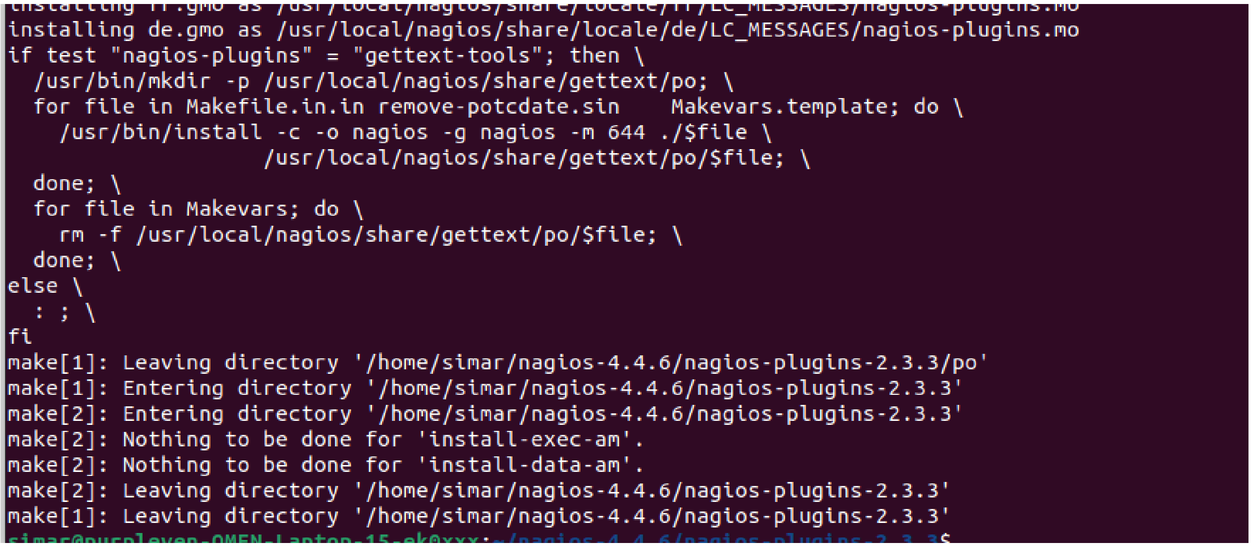


```
100 2675k  100 2675k    0     0   491k      0  0:00:05  0:00:05 --:--:--  621k
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6$ cd nagios-plugins-2.3.3
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$ ./configure --with-nagi
os-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable 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 whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
```

Now compile the plugins from source:

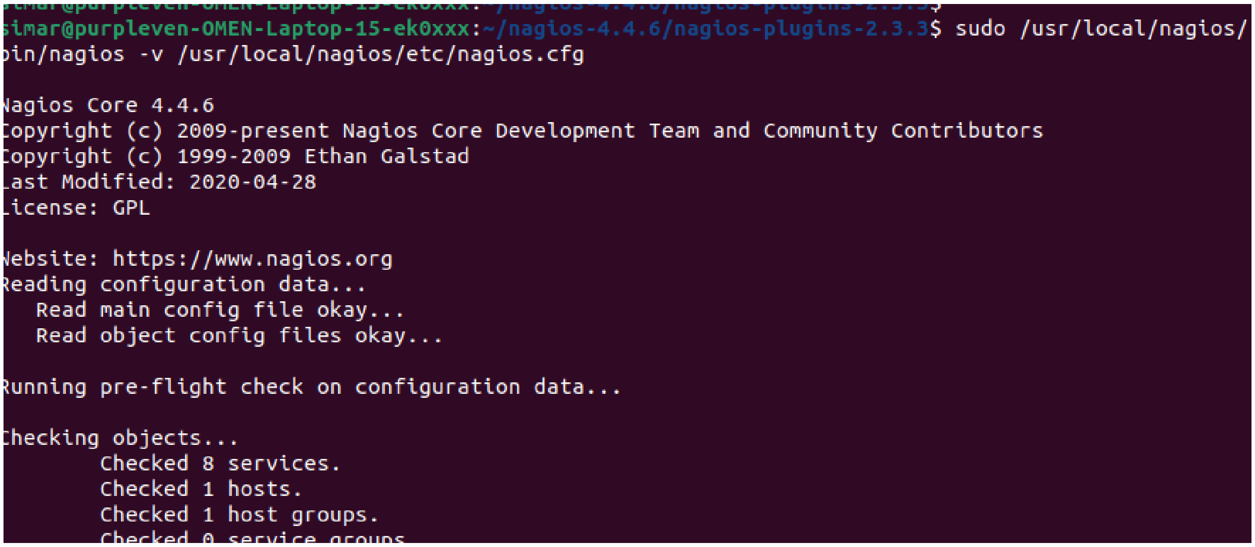


```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$ sudo make install
Making install in gl
make[1]: Entering directory '/home/simar/nagios-4.4.6/nagios-plugins-2.3.3/gl'
rm -f alloca.h-t alloca.h && \
{ 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/,$p' \
  < ../build-aux/snippet/c++defs.h \
  > c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^.ifndef/,$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 \
```

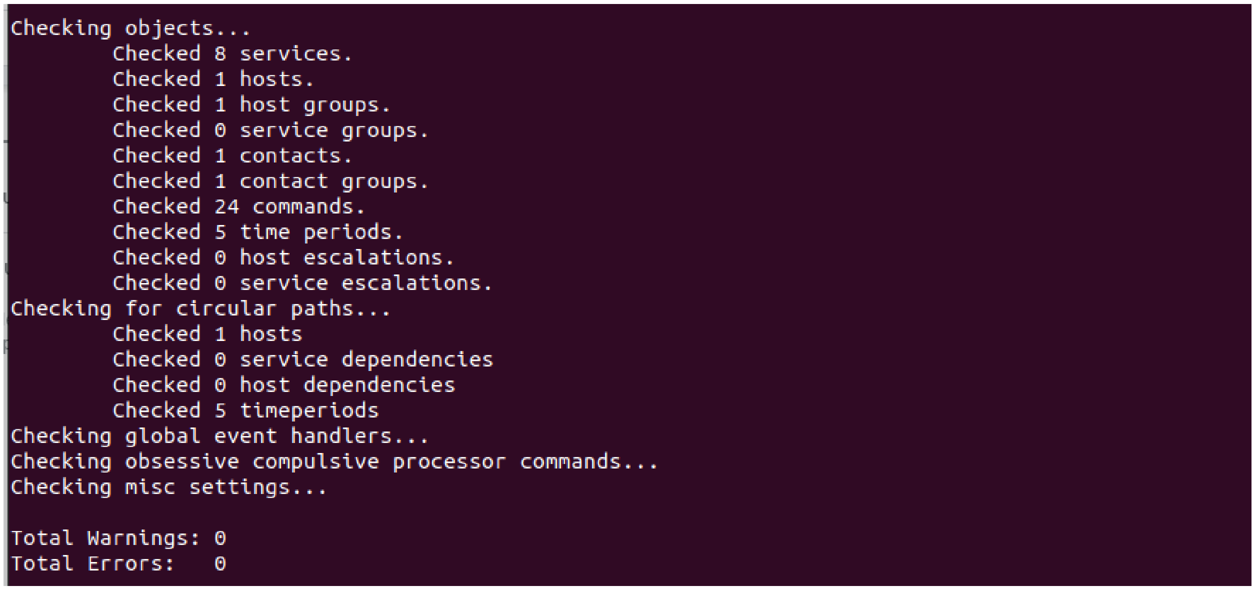


```
installing fr.gmo as /usr/local/nagios/share/locale/fr/LC_MESSAGES/nagios-plugins.mo
installing de.gmo as /usr/local/nagios/share/locale/de/LC_MESSAGES/nagios-plugins.mo
if test "nagios-plugins" = "gettext-tools"; then \
  /usr/bin/mkdir -p /usr/local/nagios/share/gettext/po; \
  for file in Makefile.in.in remove-potcdate.sin    Makevars.template; do \
    /usr/bin/install -c -o nagios -g nagios -m 644 ./$file \
                    /usr/local/nagios/share/gettext/po/$file; \
  done; \
  for file in Makevars; do \
    rm -f /usr/local/nagios/share/gettext/po/$file; \
  done; \
else \
  : ; \
fi
make[1]: Leaving directory '/home/simar/nagios-4.4.6/nagios-plugins-2.3.3/po'
make[1]: Entering directory '/home/simar/nagios-4.4.6/nagios-plugins-2.3.3'
make[2]: Entering directory '/home/simar/nagios-4.4.6/nagios-plugins-2.3.3'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/simar/nagios-4.4.6/nagios-plugins-2.3.3'
make[1]: Leaving directory '/home/simar/nagios-4.4.6/nagios-plugins-2.3.3'
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$
```

To make sure all configurations are in order, run the following command:



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$ sudo /usr/local/nagios/
bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 1 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
```

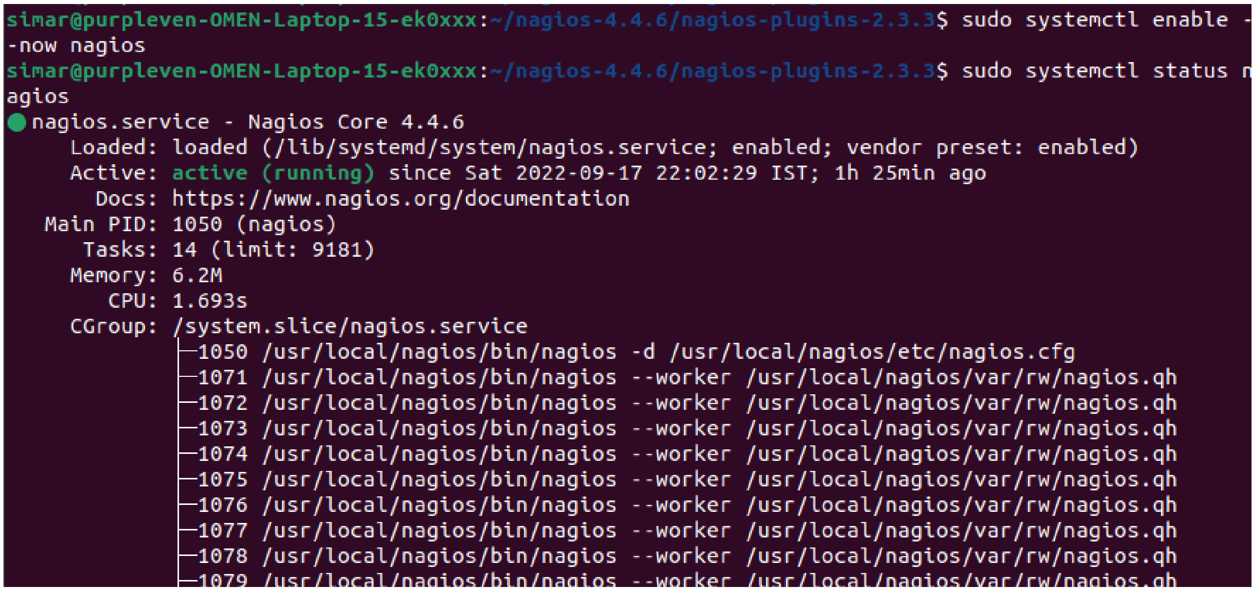


```
Checking objects...
        Checked 8 services.
        Checked 1 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 1 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0
```

Start the Nagios Daemon

The last step is to start the Nagios service, which we achieve with the following command:

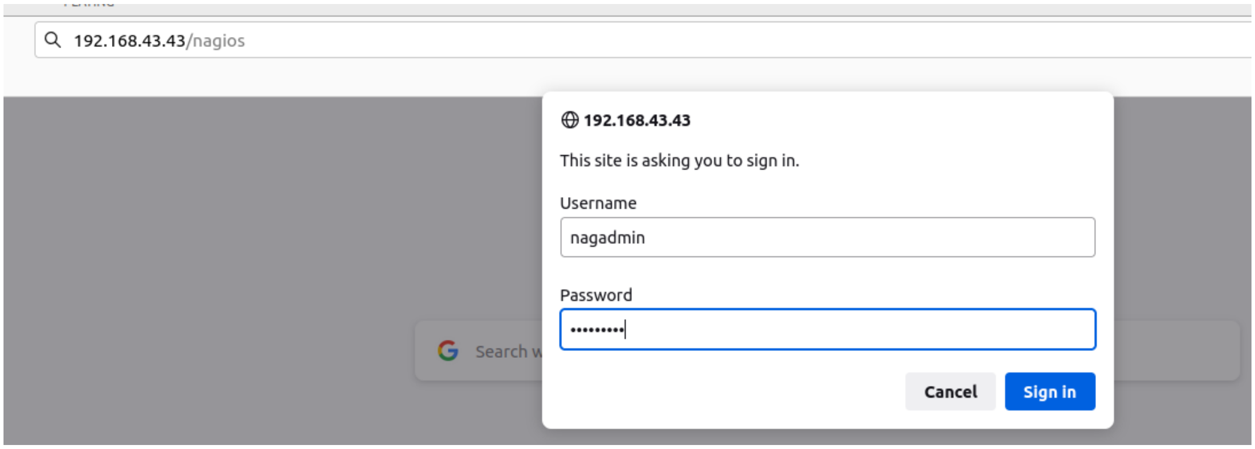
To make sure the tool is running, use the following command:



```
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$ sudo systemctl enable -
-now nagios
simar@purpleven-OMEN-Laptop-15-ek0xxx:~/nagios-4.4.6/nagios-plugins-2.3.3$ sudo systemctl status n
agios
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-09-17 22:02:29 IST; 1h 25min ago
       Docs: https://www.nagios.org/documentation
   Main PID: 1050 (nagios)
      Tasks: 14 (limit: 9181)
     Memory: 6.2M
        CPU: 1.693s
     CGroup: /system.slice/nagios.service
             ├─1050 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─1071 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1072 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1073 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1074 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1075 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1076 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1077 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1078 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─1079 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
```

You can now access the tool by opening your browser and navigating to the *http://server-IP/nagios* URL.

Once prompted, type in the credentials to sign in and you are ready to start monitoring.



192.168.43.43/nagios

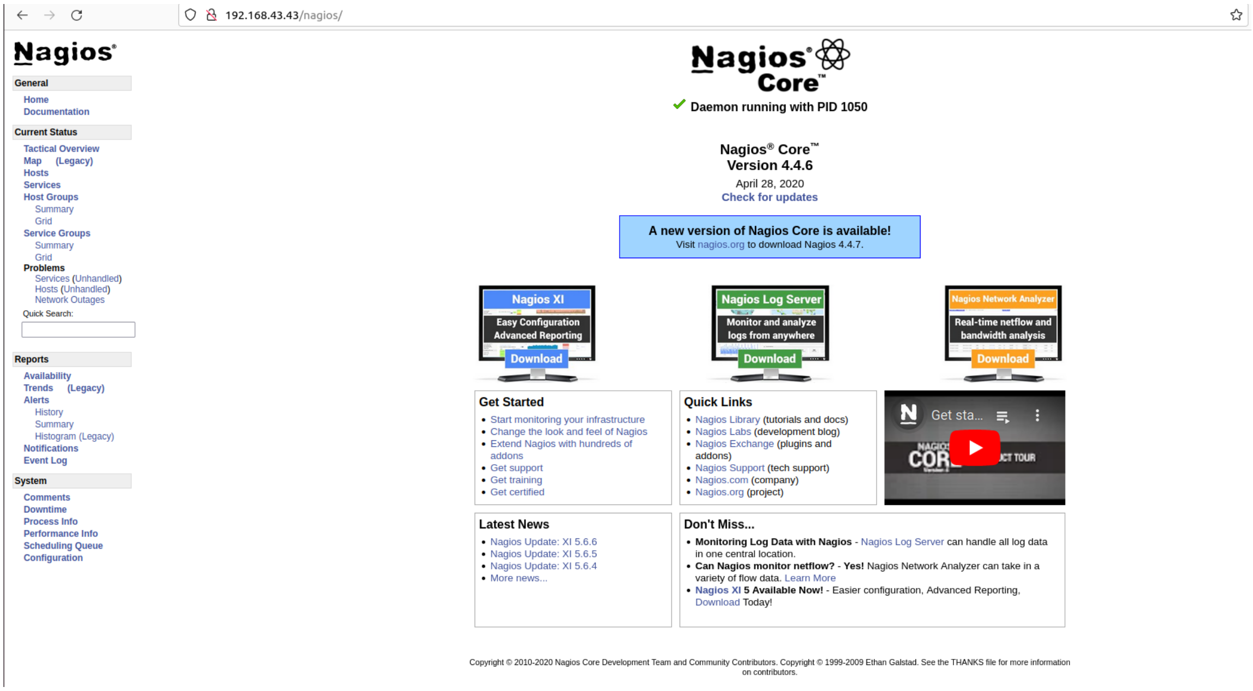
192.168.43.43

This site is asking you to sign in.

Username: nagadmin

Password: •••••••••

Installed and Logged in:



192.168.43.43/nagios/

Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends (Legacy)
- Alerts
  - History
  - Summary
  - Histogram (Legacy)
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Nagios® Core™

✓ Daemon running with PID 1050

Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.4.7.

Get Started
- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links
- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News
- Nagios Update: XI 5.6.6
- Nagios Update: XI 5.6.5
- Nagios Update: XI 5.6.4
- More news...

Don't Miss...
- **Monitoring Log Data with Nagios** - Nagios Log Server can handle all log data in one central location.
- **Can Nagios monitor netflow? - Yes!** Nagios Network Analyzer can take in a variety of flow data. Learn More
- Nagios XI **5 Available Now!** - Easier configuration, Advanced Reporting, Download Today!

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Hence,Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine is understood.