

# Install Nagios Core on CentOS 7

## Step 1: Connect to your server

```
ssh root@IP_ADDRESS -p PORT_NUMBER
```

and replace **IP\_ADDRESS** and **PORT\_NUMBER** with your actual server IP address and SSH port number.

```
[root@localhost ~]# yum update
```

```
[root@localhost ~]#
[root@localhost ~]# yum update
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
epel/x86_64/metalink | 8.9 kB 00:00
* base: bd.mirror.vanehost.com
* epel: ftp.yz.yamagata-u.ac.jp
* extras: mirrors.bupt.edu.cn
* remi-php73: remi.mirror.liteserver.nl
* remi-safe: remi.mirror.liteserver.nl
* updates: mirrors.njupt.edu.cn
base | 3.6 kB 00:00
epel | 4.7 kB 00:00
extras | 2.9 kB 00:00
mariadb | 3.4 kB 00:00:00
remi-php73 | 3.0 kB 00:00:00
remi-safe | 3.0 kB 00:00:00
updates | 2.9 kB 00:00:00
webtatic | 3.6 kB 00:00:00
(1/5): epel/x86_64/updateinfo | 1.0 MB 00:00:01
(2/5): remi-php73/primary_db | 258 kB 00:00:01
(3/5): remi-safe/primary_db | 2.3 MB 00:00:03
(4/5): epel/x86_64/primary_db | 7.0 MB 00:00:06
(5/5): updates/7/x86_64/primary_db | 18 MB 00:00:06
Resolving Dependencies
--> Running transaction check
```

## Step 2: Install LAMP

For Nagios Core to work on your server you will also need to set up a LAMP or any other web hosting stack. If you already have working web hosting stack installed on your server you can skip this and go to the next step of this tutorial.

To install a LAMP stack with Apache, MariaDB and PHP 7 on your CentOS server, run the following command:

```
[root@localhost ~]# yum install httpd mariadb-server php php-mysql -y
```

```
[root@localhost ~]#
[root@localhost ~]# yum install httpd mariadb-server php php-mysql -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: bd.mirror.vanehost.com
* epel: hkg.mirror.rackspace.com
* extras: mirrors.bupt.edu.cn
* remi-php73: remi.mirror.liteserver.nl
* remi-safe: remi.mirror.liteserver.nl
* updates: mirrors.njupt.edu.cn
```

When the MariaDB installation is complete, you can also run the following command to secure your MariaDB installation:

```
[root@localhost ~]# mysql_secure_installation
```

You will also need to enable MariaDB and Apache to start on boot with:

```
[root@localhost ~]# systemctl enable httpd.service
```

```
[root@localhost ~]# systemctl enable mariadb.service
```

```
[root@localhost ~]# systemctl enable httpd.service →
[root@localhost ~]# systemctl enable mariadb.service →
[root@localhost ~]# yum install gcc glibc glibc-common wget gd gd-devel perl postfix -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: bd.mirror.vanehost.com
 * epel: ftp.yz.yamagata-u.ac.jp
 * extras: mirrors.bupt.edu.cn
 * remi-php73: remi.mirror.liteserver.nl
 * remi-safe: remi.mirror.liteserver.nl
 * updates: mirrors.njupt.edu.cn
Package glibc-2.17-326.el7_9.x86_64 already installed and latest version
Package glibc-common-2.17-326.el7_9.x86_64 already installed and latest version
Package wget-1.14-18.el7_6.1.x86_64 already installed and latest version
Package gd-2.0.35-27.el7_9.x86_64 already installed and latest version
Package 4:perl-5.16.3-299.el7_9.x86_64 already installed and latest version
Package 2:postfix-2.10.1-9.el7.x86_64 already installed and latest version
Resolving Dependencies
```

## Step 3: Install the required packages

The following packages are also be required by the Nagios Core software. You can install them with the following command:

```
[root@localhost ~]# yum install gcc glibc glibc-common wget gd gd-devel perl postfix -y
```

```
[root@localhost ~]# systemctl enable httpd.service →
[root@localhost ~]# systemctl enable mariadb.service →
[root@localhost ~]# yum install gcc glibc glibc-common wget gd gd-devel perl postfix -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: bd.mirror.vanehost.com
 * epel: ftp.yz.yamagata-u.ac.jp
 * extras: mirrors.bupt.edu.cn
 * remi-php73: remi.mirror.liteserver.nl
 * remi-safe: remi.mirror.liteserver.nl
 * updates: mirrors.njupt.edu.cn
Package glibc-2.17-326.el7_9.x86_64 already installed and latest version
Package glibc-common-2.17-326.el7_9.x86_64 already installed and latest version
Package wget-1.14-18.el7_6.1.x86_64 already installed and latest version
Package gd-2.0.35-27.el7_9.x86_64 already installed and latest version
Package 4:perl-5.16.3-299.el7_9.x86_64 already installed and latest version
Package 2:postfix-2.10.1-9.el7.x86_64 already installed and latest version
Resolving Dependencies
```

## Step 4: Download and Install Nagios Core

Let's download the latest stable version of the Nagios Core.

First, navigate to the `tmp` directory on your server with:

```
[root@localhost tmp]# cd /tmp
```

and run the following command to download the tar archive file:

```
[root@localhost tmp]# wget -O nagioscore.tar.gz  
https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
```

```
[root@localhost ~]#  
[root@localhost ~]# cd /tmp  
[root@localhost tmp]# wget -O nagioscore.tar.gz https://github.com/NagiosEnterpr  
ises/nagioscore/archive/nagios-4.4.2.tar.gz  
--2022-12-06 12:15:55-- https://github.com/NagiosEnterprises/nagioscore/archive  
/nagios-4.4.2.tar.gz  
Resolving github.com (github.com)... 20.205.243.166  
Connecting to github.com (github.com)|20.205.243.166|:443... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/refs/t  
ags/nagios-4.4.2 [following]  
--2022-12-06 12:15:56-- https://codeload.github.com/NagiosEnterprises/nagioscor  
e/tar.gz/refs/tags/nagios-4.4.2  
Resolving codeload.github.com (codeload.github.com)... 20.205.243.165  
Connecting to codeload.github.com (codeload.github.com)|20.205.243.165|:443... c  
onected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [application/x-gzip]  
Saving to: 'nagioscore.tar.gz'  
  
[          <=>          ] 11,301,457  3.38MB/s  in 3.2s  
  
2022-12-06 12:16:00 (3.38 MB/s) - 'nagioscore.tar.gz' saved [11301457]
```

Once the download is complete, execute the following command to extract the archive file:

```
[root@localhost tmp]# tar xzf nagioscore.tar.gz
```

```
[root@localhost tmp]# tar xzf nagioscore.tar.gz
```

Once the archive is extracted, move to the `nagioscore-nagios-4.4.2` directory:

```
[root@localhost tmp]# cd /tmp/nagioscore-nagios-4.4.2
```

```
[root@localhost tmp]# cd /tmp/nagioscore-nagios-4.4.2
```

and run the following command in order to configure the installer and prepare the Nagios Core source code for the compilation process:

```
[root@localhost nagioscore-nagios-4.4.2]# ./configure
```

```
[root@localhost nagioscore-nagios-4.4.2]# ./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
```

After the configuration is completed, you can now compile Nagios Core by execution the following command:

```
[root@localhost nagioscore-nagios-4.4.2]# make all
```

```
[root@localhost nagioscore-nagios-4.4.2]# make all
cd ./base && make
make[1]: Entering directory `/tmp/nagioscore-nagios-4.4.2/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o logging.o logging.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
```

You also need to run the following commands to create the `nagios` user and group and add `apache` to the `nagios` group:

```
[root@localhost nagioscore-nagios-4.4.2]# make install-groups-users
```

```
[root@localhost nagioscore-nagios-4.4.2]# usermod -a -G nagios apache
```

```
[root@localhost nagioscore-nagios-4.4.2]# make install-groups-users
groupadd -r nagios
useradd -g nagios nagios
[root@localhost nagioscore-nagios-4.4.2]# usermod -a -G nagios apache
```

Now, you can finally run the following command in order to install Nagios Core:



```
[root@localhost nagioscore-nagios-4.4.2]# make install
```

```
Report bugs to <bug-make@gnu.org>
[root@localhost nagioscore-nagios-4.4.2]# make install
cd ./base && make install
make[1]: Entering directory `/tmp/nagioscore-nagios-4.4.2/base'
make install-basic
make[2]: Entering directory `/tmp/nagioscore-nagios-4.4.2/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 nagiosstats /usr/local/nagios/bin
make[2]: Leaving directory `/tmp/nagioscore-nagios-4.4.2/base'
make strip-post-install
make[2]: Entering directory `/tmp/nagioscore-nagios-4.4.2/base'
/usr/bin/strip /usr/local/nagios/bin/nagios
/usr/bin/strip /usr/local/nagios/bin/nagiosstats
make[2]: Leaving directory `/tmp/nagioscore-nagios-4.4.2/base'
make[1]: Leaving directory `/tmp/nagioscore-nagios-4.4.2/base'
cd ./cgi && make install
make[1]: Entering directory `/tmp/nagioscore-nagios-4.4.2/cgi'
make install-basic
make[2]: Entering directory `/tmp/nagioscore-nagios-4.4.2/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/s
```

To, install the initialization script which can be used to manage your Nagios service, run the following command:

```
[root@localhost nagioscore-nagios-4.4.2]# make install-daemoninit
```

```
make[1]: Leaving directory `/tmp/nagioscore-nagios-4.4.2'
[root@localhost nagioscore-nagios-4.4.2]# make install-daemoninit
/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.service
Created symlink from /etc/systemd/system/multi-user.target.wants/nagios.service
to /usr/lib/systemd/system/nagios.service.

*** Init script installed ***
```

Next, run the following command to install the Nagios sample configuration files:

```
[root@localhost nagioscore-nagios-4.4.2]# make install-config
```

```
[root@localhost nagioscore-nagios-4.4.2]# 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/lo
cal/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /us
r/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/
```

Run the following command to install and configure the external command file to make Nagios Core to work from the command line:

```
[root@localhost nagioscore-nagios-4.4.2]# make install-commandmode
```

```
[root@localhost nagioscore-nagios-4.4.2]# 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 ***
```

The following command will install the Apache web server configuration files:

```
[root@localhost nagioscore-nagios-4.4.2]# make install-webconf
```

```
[root@localhost nagioscore-nagios-4.4.2]# make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***
```

After all the installations are complete, restart your apache service with:

```
[root@localhost nagioscore-nagios-4.4.2]# systemctl restart httpd
```

## Step 5: Create nagiosadmin User Account

To be able to log into Nagios, you will need to create an Apache user account.

You need to run the following command to create a new user account called `nagiosadmin` and assign a password to it:

```
[root@localhost nagioscore-nagios-4.4.2]# htpasswd -c
/usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
[root@localhost nagioscore-nagios-4.4.2]# systemctl restart httpd
[root@localhost nagioscore-nagios-4.4.2]# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password: 123
Re-type new password:
Adding password for user nagiosadmin
[root@localhost nagioscore-nagios-4.4.2]#
```

With this step, the main Nagios Core installation is now complete. However, for Nagios Core to operate properly you will also need to install the Nagios Plugins as explained in the next step.

## Step 6: Install Nagios Plugins

Before you download and install the Nagios plugins, you need to make sure that the following packages are installed on your CentOS 7 server:

```
[root@localhost nagioscore-nagios-4.4.2]# yum install gcc glibc glibc-common  
make gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils epel-  
release perl-Net-SNMP -y
```

```
[root@localhost nagioscore-nagios-4.4.2]#  
[root@localhost nagioscore-nagios-4.4.2]# yum install gcc glibc glibc-common mak  
e gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils epel-rele  
ase perl-Net-SNMP -y  
Loaded plugins: fastestmirror, langpacks  
Loading mirror speeds from cached hostfile  
* base: bd.mirror.vanehost.com  
* epel: ftp.yz.yamagata-u.ac.jp  
* extras: mirrors.bupt.edu.cn  
* remi-php73: remi.mirror.liteserver.nl  
* remi-safe: remi.mirror.liteserver.nl  
* updates: mirrors.cqu.edu.cn  
Package gcc-4.8.5-44.el7.x86_64 already installed and latest version  
Package glibc-2.17-326.el7_9.x86_64 already installed and latest version
```

To download and extract the latest version of the Nagios Plugins to the `tmp` directory on your server, run the following commands:

```
[root@localhost nagioscore-nagios-4.4.2]# cd /tmp
```

```
Complete!  
[root@localhost nagioscore-nagios-4.4.2]# cd /tmp  
[root@localhost tmp]# wget --no-check-certificate -O nagios-plugins.tar.gz
```

```
[root@localhost tmp]# wget --no-check-certificate -O nagios-plugins.tar.gz  
https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
```

```
[root@localhost tmp]# wget --no-check-certificate -O nagios-plugins.tar.gz https  
://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz  
--2022-12-06 12:20:39-- https://github.com/nagios-plugins/nagios-plugins/archiv  
e/release-2.2.1.tar.gz  
Resolving github.com (github.com)... 20.205.243.166  
Connecting to github.com (github.com)|20.205.243.166|:443... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: https://codeload.github.com/nagios-plugins/nagios-plugins/tar.gz/refs/  
tags/release-2.2.1 [following]  
--2022-12-06 12:20:40-- https://codeload.github.com/nagios-plugins/nagios-plugi  
ns/tar.gz/refs/tags/release-2.2.1  
Resolving codeload.github.com (codeload.github.com)... 20.205.243.165  
Connecting to codeload.github.com (codeload.github.com)|20.205.243.165|:443... c  
onected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [application/x-gzip]  
Saving to: 'nagios-plugins.tar.gz'  
  
[ <=> ] 2,049,050 1023KB/s in 2.0s  
2022-12-06 12:20:42 (1023 KB/s) - 'nagios-plugins.tar.gz' saved [2049050]
```

```
[root@localhost tmp]#tar zxf nagios-plugins.tar.gz
```

```
[root@localhost tmp]# tar zxf nagios-plugins.tar.gz
[root@localhost tmp]# cd /tmp/nagios-plugins-release-2.2.1/
[root@localhost nagios-plugins-release-2.2.1]# ./tools/setup
which: no gnumake in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin)
Found GNU Make at /usr/bin/gmake ... good.
configure.ac:47: installing 'build-aux/compile'
configure.ac:12: installing 'build-aux/config.guess'
configure.ac:12: installing 'build-aux/config.sub'
configure.ac:9: installing 'build-aux/install-sh'
configure.ac:9: installing 'build-aux/missing'
Makefile.am: installing './INSTALL'
gl/Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
```

Once the Nagios Plugins archive is extracted, run the following commands to compile and install the Nagios Plugins on your server:

```
[root@localhost tmp]#cd /tmp/nagios-plugins-release-2.2.1/
```


```
[root@localhost tmp]# tar zxf nagios-plugins.tar.gz
[root@localhost tmp]# cd /tmp/nagios-plugins-release-2.2.1/
[root@localhost nagios-plugins-release-2.2.1]# ./tools/setup
```

```
[root@localhost nagios-plugins-release-2.2.1]#./tools/setup
```

```
[root@localhost tmp]# cd /tmp/nagios-plugins-release-2.2.1/
[root@localhost nagios-plugins-release-2.2.1]# ./tools/setup
which: no gnumake in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin)
Found GNU Make at /usr/bin/gmake ... good.
configure.ac:47: installing 'build-aux/compile'
configure.ac:12: installing 'build-aux/config.guess'
configure.ac:12: installing 'build-aux/config.sub'
configure.ac:9: installing 'build-aux/install-sh'
configure.ac:9: installing 'build-aux/missing'
Makefile.am: installing './INSTALL'
gl/Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
```

```
[root@localhost nagios-plugins-release-2.2.1]#./configure
```

```
parallel-tests: installing 'build-aux/test-driver'
[root@localhost nagios-plugins-release-2.2.1]# ./configure
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... gawk
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
```





```
[root@localhost nagios-plugins-release-2.2.1]#Make
```

```
config.status: creating po/Makefile
[root@localhost nagios-plugins-release-2.2.1]# make
make all-recursive
make[1]: Entering directory `/tmp/nagios-plugins-release-2.2.1'
Making all in gl
make[2]: Entering directory `/tmp/nagios-plugins-release-2.2.1/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 && \
```

```
[root@localhost nagios-plugins-release-2.2.1]#make install
```

```
make[1]: Leaving directory `/tmp/nagios-plugins-release-2.2.1'
[root@localhost nagios-plugins-release-2.2.1]# make install
Making install in gl
make[1]: Entering directory `/tmp/nagios-plugins-release-2.2.1/gl'
make install-recursive
make[2]: Entering directory `/tmp/nagios-plugins-release-2.2.1/gl'
make[3]: Entering directory `/tmp/nagios-plugins-release-2.2.1/gl'
make[4]: Entering directory `/tmp/nagios-plugins-release-2.2.1/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 ; \
```

## Step 7: Accessing Nagios Core

After you have successfully installed Nagios Core and the Nagios Plugins your CentOS 7 system, you can use the following command to start the Nagios service:

```
[root@localhost nagios-plugins-release-2.2.1]#systemctl start nagios
```

To access Nagios Core, open your browser and navigate to <http://YOUR-IP-ADDRESS/nagios> and log in using the **user: nagiosadmin password : 123** user account which you have created in one of the previous steps in this tutorial.

After you have successfully logged in, you will be presented with the Nagios Core home screen as shown in the image below:

# 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 19523

Nagios® Core™  
Version 4.4.2

August 16, 2018  
Check for updates



**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 Core Selected as SourceForge "Project of the Week"
- NCPA 2 Released
- Nagios Core 4.2.4 and NSCA 2.9.2 Released
- 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!](#)

# 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:

## Current Network Status

Last Updated: Tue Dec 6 15:01:10 +06 2022  
Updated every 90 seconds  
Nagios® Core™ 4.4.2 - [www.nagios.org](http://www.nagios.org)  
Logged in as nagiosadmin

[View Service Status Detail For All Host Groups](#)  
[View Status Overview For All Host Groups](#)  
[View Status Summary For All Host Groups](#)  
[View Status Grid For All Host Groups](#)

## Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0
All Problems		All Types	
0		2	

## Service Status Totals

Ok	Warning	Unknown	Critical	Pending
8	0	0	2	5
All Problems		All Types		
2		15		

## Host Status Details For All Host Groups

Limit Results:

Host	Status	Last Check	Duration	Status Information
localhost	UP	12-06-2022 14:58:21	0d 2h 37m 49s	PING OK - Packet loss = 0%, RTA = 0.14 ms
windows-Host	UP	12-06-2022 15:00:34	0d 0h 2m 33s	PING OK - Packet loss = 0%, RTA = 0.97 ms

Results 1 - 2 of 2 Matching Hosts

```
[root@localhost ~]# cd /usr/local/nagios/etc/
```

```
[root@localhost etc]# ls
```

```
[root@localhost etc]# vim nagios.cfg
```

```
# Definitions for monitoring a Windows machine
```

```
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

```
#####
#
# NAGIOS.CFG - Sample Main Config File for Nagios 4.4.2
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#####

# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log

# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# 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
```

Host add and uncomment

```
[root@localhost etc]# cd objects/
[root@localhost objects]# ls
[root@localhost objects]# vim windows.cfg
```

```

define host {
    use                windows-server          ; Inherit default values from a template
    host_name          windows-Host            ; The name we're giving to this host
    alias              My Windows Server       ; A longer name associated with the host
    address             10.200.10.27           ; IP address of the host
}

#####
#
# HOST GROUP DEFINITIONS
#
#####

# Define a hostgroup for Windows machines
# All hosts that use the windows-server template will automatically be a member of this group

define hostgroup {
    hostgroup_name     windows-servers         ; The name of the hostgroup
    alias              Windows Servers         ; Long name of the group
}

#####
#
# SERVICE DEFINITIONS
#
#####

# Create a service for monitoring the version of NSClient++ that is installed
# Change the host_name to match the name of the host you defined above

define service {

```

```

# Change the host_name to match the name of the host you defined above

```

```

define service {
    use                generic-service
    host_name          windows-Host
    service_description NSClient++ Version
    check_command      check_nt!CLIENTVERSION
}

```

```

# Create a service for monitoring the uptime of the server
# Change the host_name to match the name of the host you defined above

```

```

define service {
    use                generic-service
    host_name          windows-Host
    service_description Uptime
    check_command      check_nt!UPTIME
}

```

```

# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined above

```

```

define service {
    use                generic-service
    host_name          windows-Host
    service_description CPU Load
    check_command      check_nt!CPULOAD!-l 5,80,90
}

```

```

# Create a service for monitoring memory usage
# Change the host_name to match the name of the host you defined above

```



## 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:

**Current Network Status**  
 Last Updated: Tue Dec 6 15:22:27 +06 2022  
 Updated every 90 seconds  
 Nagios® Core™ 4.4.2 - [www.nagios.org](http://www.nagios.org)  
 Logged in as nagiosadmin

[View History For This Host](#)  
[View Notifications For This Host](#)  
[View Service Status Detail For All Hosts](#)

Host Status Totals			
Up	Down	Unreachable	Pending
1	0	0	0
All Problems		All Types	
0		1	

Service Status Totals				
Ok	Warning	Unknown	Critical	Pending
8	0	0	0	0
All Problems		All Types		
0		8		

## Service Status Details For Host 'localhost'

Limit Results:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	2-06-2022 15:18:57	0d 2h 58m 30s	1/4	OK - load average: 0.02, 0.02, 0.05
	Current Users	OK	2-06-2022 15:19:35	0d 2h 57m 52s	1/4	USERS OK - 3 users currently logged in
	HTTP	OK	2-06-2022 15:20:12	0d 2h 57m 15s	1/4	HTTP OK: HTTP/1.1 200 OK - 4232 bytes in 0.445 second response time
	PING	OK	2-06-2022 15:20:50	0d 2h 56m 37s	1/4	PING OK - Packet loss = 0%, RTA = 0.10 ms
	Root Partition	OK	2-06-2022 15:21:27	0d 2h 56m 0s	1/4	DISK OK - free space: / 20907 MB (75.67% inode=99%):
	SSH	OK	2-06-2022 15:20:59	0d 2h 55m 22s	1/4	SSH OK - OpenSSH_7.4 (protocol 2.0)
	Swap Usage	OK	2-06-2022 15:17:42	0d 2h 54m 45s	1/4	SWAP OK - 99% free (2011 MB out of 2047 MB)
	Total Processes	OK	2-06-2022 15:18:20	0d 2h 54m 7s	1/4	PROCS OK: 52 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services