# Nagios Core - Installing Nagios Core From Source

Installing Nagios Core From Source

This document describes how to install Nagios Core from source.

This guide is broken up into several sections and covers different operating system (OS) distributions. If your OS Distribution is not included in this guide then please contact us to see if we can get it added. Some distributions may be missing as we don't have access to a test environment that allows us to develop the documentation.

Nagios Core 4.4.3 and Nagios Plugins 2.2.1 is what this guide instructs you to install, however future versions should also work fine with these steps.

This documentation is broken up into two distinct sections:

- Install Nagios Core
- Install Nagios Plugins

This separation is to make a clear distinction as to what prerequisite packages are required by the OS it is being installed on. For example the SNMP packages are installed as part of the Nagios Plugins section, as SNMP is not required by Nagios Core.

Please select your OS:

- Red Hat Enterprise Linux (RHEL)
- CentOS
- Oracle Linux
- Ubuntu
- SUSE SLES | openSUSE Leap
- Debian
- Raspbian
- Fedora
- Arch Linux
- Gentoo
- FreeBSD
- Solaris
- Apple OS X

CentOS | RHEL | Oracle Linux

### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. Steps to do this are as follows.

```
sed -i 's/SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
setenforce 0
```

### Prerequisites

Perform these steps to install the pre-requisite packages.

```
yum install -y gcc glibc glibc-common wget unzip httpd php gd gd-devel perl postfix
```

### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure
make all
```

### Create User And Group

This creates the `nagios` user and group. The `apache` user is also added to the `nagios` group.

```
make install-groups-users
usermod -a -G nagios apache
```

### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
make install
```

### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot. The Apache `httpd` service is also configured at this point.

===== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
make install-daemoninit
chkconfig --level 2345 httpd on
```

===== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
make install-daemoninit
systemctl enable httpd.service
```

Information on starting and stopping services will be explained further on.

### Install Command Mode

This installs and configures the external command file.

```
make install-commandmode
```

### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
make install-config
```

### Install Apache Config Files

This installs the Apache web server configuration files. Also configure Apache settings if required.

```
make install-webconf
```

### Configure Firewall

You need to allow port 80 inbound traffic on the local firewall so you can reach the Nagios Core web interface.

====== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x ======

```
iptables -I INPUT -p tcp --destination-port 80 -j ACCEPT
service iptables save
ip6tables -I INPUT -p tcp --destination-port 80 -j ACCEPT
service ip6tables save
```

====== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x ======

```
firewall-cmd --zone=public --add-port=80/tcp
firewall-cmd --zone=public --add-port=80/tcp --permanent
```

### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

### Start Apache Web Server

====== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x ======

```
service httpd start
```

====== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x ======

```
systemctl start httpd.service
```

### Start Service / Daemon

This command starts Nagios Core.

====== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x ======

```
service nagios start
```

====== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x ======

```
systemctl start nagios.service
```

### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

### Prerequisites

Make sure that you have the following packages installed.

====== CentOS 5.x ======

```
yum install -y gcc glibc glibc-common make gettext automake wget openssl-devel net-snmp net-snmp-utils epel-release
yum install -y perl-Net-SNMP
cd /tmp
wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.60.tar.gz
tar xzf autoconf-2.60.tar.gz
cd /tmp/autoconf-2.60
./configure
make
make install
```

====== CentOS 6.x / 7.x ======

```
yum install -y gcc glibc glibc-common make gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils epel-release
yum install -y perl-Net-SNMP
```

====== RHEL 5.x | Oracle Linux 5.x ======

```
cd /tmp
wget http://archives.fedoraproject.org/pub/archive/epel/epel-release-latest-5.noarch.rpm
```

```
rpm -ihv epel-release-latest-5.noarch.rpm
yum install -y gcc glibc glibc-common make gettext automake wget openssl-devel net-snmp net-snmp-utils
yum install -y perl-Net-SNMP
wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.60.tar.gz
tar xzf autoconf-2.60.tar.gz
cd /tmp/autoconf-2.60
./configure
make
make install
```

===== RHEL 6.x | Oracle Linux 6.x =====

```
cd /tmp
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
rpm -ihv epel-release-latest-6.noarch.rpm
yum install -y gcc glibc glibc-common make gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils
yum install -y perl-Net-SNMP
```

===== RHEL 7.x =====

```
cd /tmp
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
rpm -ihv epel-release-latest-7.noarch.rpm
subscription-manager repos --enable=rhel-7-server-optional-rpms
yum install -y gcc glibc glibc-common make gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils
yum install -y perl-Net-SNMP
```

===== Oracle Linux 7.x =====

```
yum install -y yum-utils
yum-config-manager --enable ol7_optional_latest
cd /tmp
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
rpm -ihv epel-release-latest-7.noarch.rpm
yum install -y gcc glibc glibc-common make gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils
yum install -y perl-Net-SNMP
```

#### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

#### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
make
make install
```

#### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

#### Service / Daemon Commands

Different Linux distributions have different methods of starting / stopping / restarting / status Nagios.

===== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
service nagios start
service nagios stop
service nagios restart
service nagios status
```

===== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
systemctl start nagios.service
systemctl stop nagios.service
systemctl restart nagios.service
systemctl status nagios.service
```

Ubuntu

#### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. SELinux is not enabled by default on Ubuntu. If you would like to see if it is installed run the following command:

```
sudo dpkg -l selinux*
```

#### Prerequisites

Perform these steps to install the pre-requisite packages.

===== Ubuntu 14.x / 15.x =====

```
sudo apt-get update
sudo apt-get install -y autoconf gcc libc6 make wget unzip apache2 apache2-utils php5 libgd2-xpm-dev
```

===== Ubuntu 16.x / 17.x =====

```
sudo apt-get update
sudo apt-get install -y autoconf gcc libc6 make wget unzip apache2 php libapache2-mod-php7.0 libgd2-xpm-dev
```

===== Ubuntu 18.x =====

```
sudo apt-get update
sudo apt-get install -y autoconf gcc libc6 make wget unzip apache2 php libapache2-mod-php7.2 libgd-dev
```

#### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

#### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
sudo make all
```

#### Create User And Group

This creates the `nagios` user and group. The `www-data` user is also added to the `nagios` group.

```
sudo make install-groups-users
sudo usermod -a -G nagios www-data
```

#### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
sudo make install
```

#### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot.

```
sudo make install-daemoninit
```

Information on starting and stopping services will be explained further on.

#### Install Command Mode

This installs and configures the external command file.

```
sudo make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
sudo make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files and configures Apache settings.

```
sudo make install-webconf
sudo a2enmod rewrite
sudo a2enmod cgi
```

#### Configure Firewall

You need to allow port 80 inbound traffic on the local firewall so you can reach the Nagios Core web interface.

```
sudo ufw allow Apache
sudo ufw reload
```

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

===== Ubuntu 14.x =====

Need to restart it because it is already running.

```
sudo service apache2 restart
```

===== Ubuntu 15.x / 16.x / 17.x /18.x =====

Need to restart it because it is already running.

```
sudo systemctl restart apache2.service
```

#### Start Service / Daemon

This command starts Nagios Core.

===== Ubuntu 14.x =====

```
sudo start nagios
```

===== Ubuntu 15.x / 16.x / 17.x / 18.x =====

```
sudo systemctl start nagios.service
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

#### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

#### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

### Prerequisites

Make sure that you have the following packages installed.

```
sudo apt-get install -y autoconf gcc libc6 libmcrypt-dev make libssl-dev wget bc gawk dc build-essential snmp libnet-snmp-perl gettext
```

### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
sudo ./tools/setup
sudo ./configure
sudo make
sudo make install
```

### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

### Service / Daemon Commands

Different Linux distributions have different methods of starting / stopping / restarting / status Nagios.

===== Ubuntu 14.x =====

```
sudo start nagios
sudo stop nagios
sudo restart nagios
sudo status nagios
```

===== Ubuntu 15.x / 16.x / 17.x / 18.x =====

```
sudo systemctl start nagios.service
sudo systemctl stop nagios.service
sudo systemctl restart nagios.service
sudo systemctl status nagios.service
```

SUSE SLES | openSUSE Leap

### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. SELinux is not enabled by default on SUSE. If you would like to see if it is enabled run the following command:

```
ls -la /etc/selinux/config
```

If the file does not exist, SELinux is not enabled.

### Prerequisites

Perform these steps to install the pre-requisite packages.

===== SUSE SLES 11.3 =====

```
cd /tmp
wget 'https://nu.novell.com/repo/$RCE/SLE11-SDK-SP3-Pool/sle-11-x86_64/rpm/x86_64/sle-sdk-release-11.3-1.69.x86_64.rpm'
wget 'https://nu.novell.com/repo/$RCE/SLE11-SDK-SP3-Pool/sle-11-x86_64/rpm/x86_64/sle-sdk-release-SDK-11.3-1.69.x86_64.rpm'
sudo rpm -ivh sle-sdk-release-*
sudo suse_register
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php53 apache2-mod_php53 gd gd-devel
```

===== SUSE SLES 11.4 =====

```
cd /tmp
wget 'https://nu.novell.com/repo/$RCE/SLE11-SDK-SP4-Pool/sle-11-x86_64/rpm/x86_64/sle-sdk-release-11.4-1.55.x86_64.rpm'
wget 'https://nu.novell.com/repo/$RCE/SLE11-SDK-SP4-Pool/sle-11-x86_64/rpm/x86_64/sle-sdk-release-SDK-11.4-1.55.x86_64.rpm'
sudo rpm -ivh sle-sdk-release-*
sudo suse_register
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php53 apache2-mod_php53 gd gd-devel
```

===== SUSE SLES 12 =====

```
sudo SUSEConnect -p sle-sdk/12/x86_64
sudo SUSEConnect -p sle-module-web-scripting/12/x86_64
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php5 apache2-mod_php5 gd gd-devel
```

===== SUSE SLES 12.1 =====

```
sudo SUSEConnect -p sle-sdk/12.1/x86_64
sudo SUSEConnect -p sle-module-web-scripting/12/x86_64
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php5 apache2-mod_php5 gd gd-devel
```

===== SUSE SLES 12.2 =====

```
sudo SUSEConnect -p sle-sdk/12.2/x86_64
sudo SUSEConnect -p sle-module-web-scripting/12/x86_64
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php5 apache2-mod_php5 gd gd-devel
```

===== SUSE SLES 12.3 =====

```
sudo SUSEConnect -p sle-sdk/12.3/x86_64
sudo SUSEConnect -p sle-module-web-scripting/12/x86_64
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php5 apache2-mod_php5 gd gd-devel
```

===== openSUSE Leap 42.x =====

```
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php5 apache2-mod_php5 gd gd-devel
```

===== openSUSE Leap 15.x =====

```
sudo zypper --non-interactive install autoconf gcc glibc make wget unzip apache2 apache2-utils php7 apache2-mod_php7 gd gd-devel
```

#### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

#### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
sudo ./configure --with-httpd-conf=/etc/apache2/vhosts.d
sudo make all
```

#### Create User And Group

This creates the `nagios` user and group. The `apache` user is also added to the `nagios` group.

===== SUSE SLES 11.x =====

```
sudo make install-groups-users
sudo /usr/sbin/usermod -A nagios wwwrun
```

===== SUSE SLES 12.x | openSUSE Leap 42.x / 15.x =====

```
sudo make install-groups-users
sudo /usr/sbin/usermod -a -G nagios wwwrun
```

#### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
sudo make install
```

#### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot.

```
sudo make install-daemoninit
```

Information on starting and stopping services will be explained further on.

#### Install Command Mode

This installs and configures the external command file.

```
sudo make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
sudo make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files. Also configure Apache settings if required.

===== SUSE SLES 11.x =====

```
sudo make install-webconf
sudo /usr/sbin/a2enmod rewrite
sudo /usr/sbin/a2enmod cgi
sudo /usr/sbin/a2enmod version
sudo /usr/sbin/a2enmod php5
sudo /sbin/chkconfig --set apache2 on
```

===== SUSE SLES 12.x | openSUSE Leap 42.x =====

```
sudo make install-webconf
sudo /usr/sbin/a2enmod rewrite
sudo /usr/sbin/a2enmod cgi
sudo /usr/sbin/a2enmod version
sudo /usr/sbin/a2enmod php5
sudo systemctl enable apache2.service
```

===== openSUSE Leap 15.x =====

```
sudo make install-webconf
sudo /usr/sbin/a2enmod rewrite
sudo /usr/sbin/a2enmod cgi
sudo /usr/sbin/a2enmod version
sudo /usr/sbin/a2enmod php7
sudo systemctl enable apache2.service
```

#### Configure Firewall

You need to allow port 80 inbound traffic on the local firewall so you can reach the Nagios Core web interface.

===== SUSE SLES 11.x =====

```
sudo sed -i '/FW_SERVICES_EXT_TCP=/s/\"$/\ 80\"/' /etc/sysconfig/SuSEfirewall2
sudo /sbin/service SuSEfirewall2_init restart
sudo /sbin/service SuSEfirewall2_setup restart
```

===== SUSE SLES 12.x =====

```
sudo /usr/sbin/SuSEfirewall2 open EXT TCP 80
sudo systemctl restart SuSEfirewall2
```

===== openSUSE Leap 42.x =====

Port 80 is enabled when Apache is installed, nothing needs to be done.

===== openSUSE Leap 15.x =====

```
sudo firewall-cmd --zone=public --add-port=80/tcp
sudo firewall-cmd --zone=public --add-port=80/tcp --permanent
```

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
sudo htpasswd2 -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

===== SUSE SLES 11.x =====

```
sudo /sbin/service apache2 start
```

===== SUSE SLES 12.x | openSUSE Leap 42.x / 15.x =====

```
sudo systemctl start apache2.service
```

#### Start Service / Daemon

This command starts Nagios Core.

===== SUSE SLES 11.x =====

```
sudo /sbin/service nagios start
```

===== SUSE SLES 12.x | openSUSE Leap 42.x / 15.x =====

```
sudo systemctl start nagios.service
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

#### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

#### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

#### Prerequisites

Make sure that you have the following packages installed.

===== SUSE SLES 11.x / 12.x | openSUSE Leap 42.x =====

```
sudo zypper --non-interactive install autoconf gcc glibc libmcrypt-devel make libopenssl-devel wget gettext gettext-runtime automake net-snmp perl-Net-SNMP
```

===== openSUSE Leap 15.x =====

```
sudo zypper --non-interactive install autoconf gcc glibc libgcrypt-devel make libopenssl-devel wget gettext gettext-runtime automake net-snmp perl-Net-SNMP
```

#### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

#### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
sudo ./tools/setup
sudo ./configure
sudo make
sudo make install
```

#### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

#### Service / Daemon Commands

Different Linux distributions have different methods of starting / stopping / restarting / status Nagios.

===== SUSE SLES 11.x =====

```
sudo /sbin/service nagios start
sudo /sbin/service nagios stop
```

```
sudo /sbin/service nagios restart
sudo /sbin/service nagios status
```

===== SUSE SLES 12.x | openSUSE Leap 42.x / 15.x =====

```
sudo systemctl start nagios.service
sudo systemctl stop nagios.service
sudo systemctl restart nagios.service
sudo systemctl status nagios.service
```

Debian | Raspbian

All steps on Debian require to run as root. To become root simply run:

Debian:

```
su
```

Raspbian:

```
sudo -i
```

All commands from this point onwards will be as root.

#### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. SELinux is not enabled by default on Debian. If you would like to see if it is installed run the following command:

```
dpkg -l selinux*
```

#### Prerequisites

Perform these steps to install the pre-requisite packages.

===== 7.x / 8.x =====

```
apt-get update
apt-get install -y autoconf gcc libc6 make wget unzip apache2 apache2-utils php5 libgd2-xpm-dev
```

===== 9.x =====

```
apt-get update
apt-get install -y autoconf gcc libc6 make wget unzip apache2 apache2-utils php libgd-dev
```

#### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

#### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure --with-httpd-conf=/etc/apache2/sites-enabled
make all
```

#### Create User And Group

This creates the `nagios` user and group. The `www-data` user is also added to the `nagios` group.

```
make install-groups-users
usermod -a -G nagios www-data
```

#### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
make install
```

#### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot.

```
make install-daemoninit
```

Information on starting and stopping services will be explained further on.

#### Install Command Mode

This installs and configures the external command file.

```
make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files and configures the Apache settings.

```
make install-webconf
a2enmod rewrite
a2enmod cgi
```

#### Configure Firewall

You need to allow port 80 inbound traffic on the local firewall so you can reach the Nagios Core web interface.

```
iptables -I INPUT -p tcp --destination-port 80 -j ACCEPT
apt-get install -y iptables-persistent
```

Answer yes to saving existing rules

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

===== 7.x =====

Need to restart it because it is already running.

```
service apache2 restart
```

===== 8.x / 9.x =====

Need to restart it because it is already running.

```
systemctl restart apache2.service
```

#### Start Service / Daemon

This command starts Nagios Core.

===== 7.x =====

```
service nagios start
```

===== 8.x / 9.x =====

```
systemctl start nagios.service
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

#### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

#### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

#### Prerequisites

Make sure that you have the following packages installed.

```
apt-get install -y autoconf gcc libc6 libmcrypt-dev make libssl-dev wget bc gawk dc build-essential snmp libnet-snmp-perl gettext
```

#### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

#### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
make
make install
```

#### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

#### Service / Daemon Commands

Different Linux distributions have different methods of starting / stopping / restarting / status Nagios.

===== 7.x =====

```
service nagios start
service nagios stop
service nagios restart
service nagios status
```

===== 8.x / 9.x =====

```
systemctl start nagios.service
systemctl stop nagios.service
```

```
systemctl restart nagios.service
systemctl status nagios.service
```

Fedora

#### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. Steps to do this are as follows.

```
sed -i 's/SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
setenforce 0
```

#### Prerequisites

Perform these steps to install the pre-requisite packages.

```
dnf install -y gcc glibc glibc-common perl httpd php wget gd gd-devel
dnf update -y
```

#### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

#### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure
make all
```

#### Create User And Group

This creates the `nagios` user and group. The `apache` user is also added to the `nagios` group.

```
make install-groups-users
usermod -a -G nagios apache
```

#### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
make install
```

#### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot. The Apache `httpd` service is also configured at this point.

```
make install-daemoninit
systemctl enable httpd.service
```

Information on starting and stopping services will be explained further on.

#### Install Command Mode

This installs and configures the external command file.

```
make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files. Also configure Apache settings if required.

```
make install-webconf
```

#### Configure Firewall

You need to allow port 80 inbound traffic on the local firewall so you can reach the Nagios Core web interface.

```
firewall-cmd --zone=FedoraServer --add-port=80/tcp
firewall-cmd --zone=FedoraServer --add-port=80/tcp --permanent
```

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

```
systemctl start httpd.service
```

#### Start Service / Daemon

This command starts Nagios Core.

```
systemctl start nagios.service
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

**BUT WAIT ...**

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

### Prerequisites

Make sure that you have the following packages installed.

```
dnf install -y gcc glibc glibc-common openssl-devel perl wget gettext make net-snmp net-snmp-utils perl-Net-SNMP automake autoconf
```

### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
make
make install
```

### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

### Service / Daemon Commands

These commands are for starting / stopping / restarting / status Nagios.

```
systemctl start nagios.service
systemctl stop nagios.service
systemctl restart nagios.service
systemctl status nagios.service
```

Arch Linux

### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. SELinux is not installed on a base build of Arch Linux. If you would like to see if it is enabled run the following command:

```
ls -la /etc/selinux/config
```

If the file does not exist, SELinux is not enabled.

### Prerequisites

Perform these steps to install the pre-requisite packages.

```
pacman --noconfirm -Syyu
pacman --noconfirm -S gcc glibc make wget unzip apache php gd traceroute php-apache
```

### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure --with-httpd-conf=/etc/httpd/conf/extra
make all
```

### Create User And Group

This creates the `nagios` user and group. The `http` user is also added to the `nagios` group.

```
make install-groups-users
usermod -a -G nagios http
```

### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
make install
```

### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot. The Apache `httpd` service is also configured at this point.

```
make install-daemoninit
systemctl enable httpd.service
```

#### Install Command Mode

This installs and configures the external command file.

```
make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files.

```
make install-webconf
```

As well as installing the Apache web server configuration files, there are additional changes required to the Apache config file **`httpd.conf`** to enable modules.

Commands that do all of these changes are below, but first is a list of the changes being made.

Change this:

```
LoadModule mpm_event_module modules/mod_mpm_event.so
```

To this:

```
#LoadModule mpm_event_module modules/mod_mpm_event.so
```

Change this:

```
#LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

To this:

```
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

Change this:

```
<IfModule !mpm_prefork_module>
        #LoadModule cgid_module libexec/apache24/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
        #LoadModule cgi_module libexec/apache24/mod_cgi.so
</IfModule>
```

To this:

```
<IfModule !mpm_prefork_module>
        LoadModule cgid_module libexec/apache24/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
        LoadModule cgi_module libexec/apache24/mod_cgi.so
</IfModule>
```

Change this:

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

To this:

```
<IfModule dir_module>
    DirectoryIndex index.php index.html index.htm AddType application/x-httpd-php .phpAddType application/x-httpd-php-source .phps
</IfModule>
```

Add these lines to the end of the file:

```
LoadModule php7_module modules/libphp7.so
Include "conf/extra/nagios.conf"
Include "conf/extra/php7_module.conf"
<FilesMatch ".php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch ".phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
```

Execute the following commands to make the changes described above:

```
sed -i 's/^LoadModule mpm_event_module modules\/mod_mpm_event\.so/#LoadModule mpm_event_module modules\/mod_mpm_event\.so/g' /etc/httpd/conf/httpd.conf
sed -i 's/^#LoadModule mpm_prefork_module modules\/mod_mpm_prefork\.so/LoadModule mpm_prefork_module modules\/mod_mpm_prefork\.so/g' /etc/httpd/conf/httpd.conf
sed -i 's/DirectoryIndex index.html/DirectoryIndex index.php index.html index.htm AddType application\/x-httpd-php .phpAddType application\/x-httpd-php-source .phps/g' /etc/httpd/conf/httpd.conf
sed -i 's/#LoadModule cgid_module/LoadModule cgid_module/g' /etc/httpd/conf/httpd.conf
sed -i 's/#LoadModule cgi_module/LoadModule cgi_module/g' /etc/httpd/conf/httpd.conf
echo 'LoadModule php7_module modules/libphp7.so' >> /etc/httpd/conf/httpd.conf
echo 'Include "conf/extra/nagios.conf"' >> /etc/httpd/conf/httpd.conf
echo 'Include "conf/extra/php7_module.conf"' >> /etc/httpd/conf/httpd.conf
printf '\n<FilesMatch ".php$">\n' >> /etc/httpd/conf/httpd.conf
printf '\tSetHandler application/x-httpd-php\n' >> /etc/httpd/conf/httpd.conf
printf '</FilesMatch>\n' >> /etc/httpd/conf/httpd.conf
printf '<FilesMatch ".phps$">\n' >> /etc/httpd/conf/httpd.conf
printf '\tSetHandler application/x-httpd-php-source\n' >> /etc/httpd/conf/httpd.conf
printf '</FilesMatch>\n' >> /etc/httpd/conf/httpd.conf
```

#### Configure Firewall

Arch Linux does not have a firewall enabled in a fresh installation. Please refer to the Arch Linux documentation on allowing TCP port 80 inbound.

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **`-c`** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

```
systemctl start httpd.service
```

#### Start Service / Daemon

This command starts Nagios Core.

```
systemctl start nagios.service
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

#### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

#### Prerequisites

Make sure that you have the following packages installed.

```
pacman --noconfirm -S autoconf gcc glibc make openssl wget perl gettext net-snmp perl-net-snmp automake autoconf
```

#### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

#### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
make
make install
```

#### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

### Service / Daemon Commands

Different Linux distributions have different methods of starting / stopping / restarting / status Nagios.

```
systemctl start nagios.service
systemctl stop nagios.service
systemctl status nagios.service
```

Gentoo

#### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. SELinux is not installed on a base build of Gentoo. If you would like to see if it is enabled run the following command:

```
ls -la /etc/selinux/config
```

If the file does not exist, SELinux is not enabled.

#### Prerequisites

Perform these steps to install the pre-requisite packages.

```
emerge --sync
mkdir -p /etc/portage/package.use
echo "www-servers/apache dir cgi cgid event prefork apache2_modules_version" >> /etc/portage/package.use/apache
echo "dev-lang/php apache2" >> /etc/portage/package.use/php
echo "app-eselect/eselect-php apache2" >> /etc/portage/package.use/eselect-php
echo "media-libs/gd gd png jpeg" >> /etc/portage/package.use/gd
emerge --noreplace sys-devel/gcc sys-libs/glibc net-misc/wget app-arch/unzip www-servers/apache dev-lang/php media-libs/gd
```

#### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

#### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure --with-httpd-conf=/etc/apache2/vhosts.d --sysconfdir=/usr/local/nagios/etc
make all
```

#### Create User And Group

This creates the `nagios` user and group. The `http` user is also added to the `nagios` group.

```
make install-groups-users
usermod -a -G nagios apache
```

#### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
make install
```

#### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot. The `apache2` service also needs to be configured to start at boot.

===== openrc =====

```
make install-daemoninit
rc-update add apache2 default
```

===== systemd =====

```
make install-daemoninit
systemctl enable apache2.service
```

#### Install Command Mode

This installs and configures the external command file.

```
make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files.

```
make install-webconf
```

As well as installing the Apache web server configuration files, there are additional changes required to the Apache config file **`apache2`** to enable modules.

Commands that do all of these changes are below, but first is a list of the changes being made.

In the file `/etc/conf.d/apache2` change this:

```
APACHE2_OPTS="-D DEFAULT_VHOST -D INFO -D SSL -D SSL_DEFAULT_VHOST -D LANGUAGE"
```

To this:

```
APACHE2_OPTS="-D DEFAULT_VHOST -D INFO -D SSL -D SSL_DEFAULT_VHOST -D LANGUAGE -D PHP"
```

The `/run/apache_ssl_mutex` directory also needs to be created.

Execute the following commands to make the change described above:

```
sed -i '/^APACHE2_OPTS=/s/\([^"]*\)"$/\1 '"-D PHP"'"/' /etc/conf.d/apache2
mkdir -p /run/apache_ssl_mutex
```

#### Configure Firewall

Gentoo does not have a firewall enabled in a fresh installation. Please refer to the Gentoo documentation on allowing TCP port 80 inbound.

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **`-c`** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Restart Apache Web Server

===== openrc =====

```
rc-service apache2 restart
```

===== systemd =====

```
systemctl restart apache2.service
```

#### Start Service / Daemon

This command starts Nagios Core.

===== openrc =====

```
rc-service nagios start
```

===== systemd =====

```
systemctl start nagios.service
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

**BUT WAIT ...**

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

### Prerequisites

Make sure that you have the following packages installed.

```
emerge --noreplace sys-devel/gcc sys-libs/glibc net-misc/wget sys-devel/make sys-devel/gettext sys-devel/automake sys-devel/autoconf dev-libs/openssl net-analyzer/net-snmp dev-perl/Net-SNMP
```

### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
make
make install
chmod u+s /bin/ping
chmod u+s /bin/ping6
```

### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

### Service / Daemon Commands

The different init systems have different methods of starting / stopping / restarting / status Nagios.

===== openrc =====

```
rc-service nagios start
rc-service nagios stop
rc-service nagios status
rc-service nagios restart
```

===== systemd =====

```
systemctl start nagios.service
systemctl stop nagios.service
systemctl status nagios.service
systemctl restart nagios.service
```

FreeBSD

### Security-Enhanced Linux

This guide is based on SELinux being disabled or in permissive mode. SELinux is not enabled by default on FreeBSD. If you would like to see if it is enabled run the following command:

```
ls -la /etc/selinux/config
```

If the file does not exist, SELinux is not enabled.

### Prerequisites

Perform these steps to install the pre-requisite packages.

```
pkg install -y wget autoconf automake gmake gettext gcc apache24 php70 php70-extensions mod_php70 libgd
```

### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure --with-httpd-conf=/usr/local/etc/apache24/Includes
gmake all
```

### Create User And Group

This creates the `nagios` user and group. The `www` user is also added to the `nagios` group.

```
gmake install-groups-users
pw group mod nagios -m www
```

### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
gmake install
chown nagios:nagios /usr/local/nagios/bin/*
```

### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot.

```
gmake install-daemoninit
echo '/usr/local/etc/rc.d/nagios start' >> /etc/rc.local
```

Information on starting and stopping services will be explained further on.

### Install Command Mode

This installs and configures the external command file.

```
gmake install-commandmode
```

### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
gmake install-config
```

### Install Apache Config Files

This installs the Apache web server configuration files.

```
gmake install-webconf
```

As well as installing the Apache web server configuration files, there are additional changes required to the Apache config file **`httpd.conf`** to enable modules.

Commands that do all of these changes are below, but first is a list of the changes being made.

Change this:

```
<IfModule !mpm_prefork_module>
        #LoadModule cgid_module libexec/apache24/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
        #LoadModule cgi_module libexec/apache24/mod_cgi.so
</IfModule>
```

To this:

```
<IfModule !mpm_prefork_module>
        LoadModule cgid_module libexec/apache24/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
        LoadModule cgi_module libexec/apache24/mod_cgi.so
</IfModule>
```

Change this:

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

To this:

```
<IfModule dir_module>
    DirectoryIndex index.php index.html index.htm AddType application/x-httpd-php .phpAddType application/x-httpd-php-source .phps
</IfModule>
```

Add these lines to the end of the file:

```
<FilesMatch ".php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch ".phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
```

Execute the following commands to make the changes described above:

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
echo 'apache24_enable="YES"' >> /etc/rc.conf
sed -i '' 's/DirectoryIndex index.html/DirectoryIndex index.php index.html index.htm AddType application\/x-httpd-php .phpAddType application\/x-httpd-php-source .phps/g' /usr/local/etc/apache24/httpd.conf
sed -i '' 's/#LoadModule cgid_module/LoadModule cgid_module/g' /usr/local/etc/apache24/httpd.conf
sed -i '' 's/#LoadModule cgi_module/LoadModule cgi_module/g' /usr/local/etc/apache24/httpd.conf
printf '\n<FilesMatch ".php$">\n' >> /usr/local/etc/apache24/httpd.conf
printf '\tSetHandler application/x-httpd-php\n' >> /usr/local/etc/apache24/httpd.conf
printf '</FilesMatch>\n' >> /usr/local/etc/apache24/httpd.conf
printf '<FilesMatch ".phps$">\n' >> /usr/local/etc/apache24/httpd.conf
printf '\tSetHandler application/x-httpd-php-source\n' >> /usr/local/etc/apache24/httpd.conf
printf '</FilesMatch>\n' >> /usr/local/etc/apache24/httpd.conf
```

### Configure Firewall

Please refer to the FreeBSD documentation for information on how to enable or configure IP Filter to allow TCP port 80 inbound.

Documentation - Firewalls

### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **`-c`** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

### Start Apache Web Server

```
service apache24 start
```

### Start Service / Daemon

This command starts Nagios Core.

```
service nagios start
```

### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

**BUT WAIT ...**

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

### Prerequisites

Make sure that you have the following packages installed. In the steps below, when installing FreeBSD packages you will be prompted with screens asking what you would like installed. You can just press **`Enter`** to accept the default selections.

```
pkg install -y wget autoconf automake gettext gcc openssl-devel net-snmp p5-Net-SNMP-Util
```

### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
gmake
gmake install
```

### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

### Service / Daemon Commands

Different Linux distributions have different methods of starting / stopping / restarting / status Nagios.

```
service nagios start
service nagios stop
service nagios restart
service nagios status
```

Solaris

Tested with Solaris 11.

### Security-Enhanced Linux

SELinux is not implemented in Solaris and hence is not an issue.

### Prerequisites

Perform these steps to install the pre-requisite packages.

```
echo 'export PATH=$PATH:/opt/csw/bin:/usr/xpg4/bin:/usr/sfw/bin' >> ~/.profile
source ~/.profile
pkgadd -d http://get.opencsw.org/now
answer all
answer y
perl -ni.bak -le 'print; print "mirror=http://mirrors.ibiblio.org/opencsw/stable" if /mirror=/' /etc/opt/csw/pkgutil.conf
pkgutil -y -i autoconf automake unzip gd
pkg install gcc-45 web/php-53 apache-php53 gd
```

### Downloading the Source

```
cd /tmp
wget -O nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
./configure --with-httpd-conf=/etc/apache2/2.2/conf.d --with-gd-inc=/usr/include/gd2
gmake all
```

### Create User And Group

This creates the `nagios` user and group. The `webservd` user is also added to the `nagios` group.

```
gmake install-groups-users
usermod -G nagios webservd
```

### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
gmake install
```

### Install Service / Daemon

This installs the service or daemon files and also configures them to start on boot. The Apache `httpd` service is also configured at this point.

```
gmake install-daemoninit
svcbundle -o /lib/svc/manifest/site/nagios.xml -s service-name=application/nagios -s rc-script=/etc/init.d/nagios:2 -s model=contract
```

Information on starting and stopping services will be explained further on.

#### Install Command Mode

This installs and configures the external command file.

```
gmake install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
gmake install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files. Also configure Apache settings if required.

```
gmake install-webconf
```

#### Configure Firewall

On a manually networked system, IP Filter is not enabled by default. Please refer to the Solaris documentation for information on how to enable or configure IP Filter to allow TCP port 80 inbound.

Documentation - Configuring IP Filter

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

```
svcadm enable apache22
```

#### Start Service / Daemon

This command starts Nagios Core.

```
svcadm restart manifest-import
svcadm enable nagios
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

#### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

#### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. The following steps will walk you through installing Nagios Plugins.

These steps install nagios-plugins 2.2.1. Newer versions will become available in the future and you can use those in the following installation steps. Please see the releases page on GitHub for all available versions.

Please note that the following steps install most of the plugins that come in the Nagios Plugins package. However there are some plugins that require other libraries which are not included in those instructions. Please refer to the following KB article for detailed installation instructions:

Documentation - Installing Nagios Plugins From Source

#### Prerequisites

Already completed as part of the Nagios installation. Some of the plugins require the NET::SNMP perl module. Please refer to the following documentation:

http://www.net-snmp.org/docs/README.solaris.html

#### Downloading The Source

```
cd /tmp
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
tar zxf nagios-plugins.tar.gz
```

#### Compile + Install

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
gmake
gmake install
```

#### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

### Service / Daemon Commands

These commands are for starting / stopping / restarting / status Nagios.

```
svcadm enable nagios
svcadm disable nagios
svcadm restart nagios
svcs nagios
svcs -xv nagios
svcadm clear nagios
```

The following KB article has important information on Solaris and services:

How To Clear Solaris Service Maintenance Status

Apple OS X

### Security-Enhanced Linux

SELinux is not implemented in Apple OS X and hence is not an issue.

### Prerequisites

First, make sure Xcode is installed. If it is not installed visit the App Store and install Xcode (*3.8GB download*).

Then you need to download and install MacPorts, instructions for doing this can be found here:

https://www.macports.org/install.php

Once you've done this, follow these steps in a terminal session:

```
sudo xcodebuild -license
View the agreement and then type agree
sudo xcode-select --install
```

You will be prompted on your Mac display to install the component, click install

Click Agree

Wait while it is downloaded and installed

Continue in your terminal session (you may need to close and open your terminal session):

```
sudo /opt/local/bin/port install zlib libpng jpeg gd2 apache2 php70 php70-apache2handler
```

### Downloading the Source

```
cd /tmp
curl -L -o nagioscore.tar.gz https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.3.tar.gz
tar xzf nagioscore.tar.gz
```

### Compile

```
cd /tmp/nagioscore-nagios-4.4.3/
sudo ./configure --with-httpd-conf=/opt/local/apache2/conf/extra --with-gd-lib=/opt/local/lib --with-gd-inc=/opt/local/include
sudo make all
```

### Create User And Group

Currently Nagios Core does not have a built in script for creating the nagios user and group. However NRPE v3 does have this script, so we're going to download it and run it which makes things easy.

```
sudo make install-groups-users
sudo dseditgroup -o edit -a _www -t user nagios
```

### Install Binaries

This step installs the binary files, CGIs, and HTML files.

```
sudo make install
```

### Install Service / Daemon

This installs the script in **`/etc/rc.d/init.d/nagios`** which is not the correct way it should be run in OS X however a native OS X launch script does not exist in this version of Nagios Core.

```
sudo make install-daemoninit
```

You will configure it to start at boot using a Global Daemon property list.

Open the vi text editor with the following command:

```
sudo vi /Library/LaunchDaemons/org.nagios.nagios.plist
```

This creates a new file.

Press **`i`** on the keyboard to enter *insert* mode.

Paste the following into the vi editor:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple/DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>org.nagios.nagios</string>
        <key>UserName</key>
        <string>nagios</string>
        <key>GroupName</key>
        <string>nagios</string>
        <key>Program</key>
        <string>/etc/rc.d/init.d/nagios</string>
        <key>ProgramArguments</key>
        <array>
                <string>nagios</string>
                <string>start</string>
        </array>
        <key>KeepAlive</key>
        <dict>
                <key>SuccessfulExit</key>
                <false/>
                <key>NetworkState</key>
                <true/>
        </dict>
        <key>RunAtLoad</key>
        <true/>
        <key>ProcessType</key>
        <string>Background</string>
</dict>
</plist>
```

Press **`escape`** on the keyboard to exit *insert* mode.

Type **:wq** on the and press **Enter**.

The file has been saved and Nagios will automatically start on the next boot.

Information on starting and stopping services will be explained further on.

#### Install Command Mode

This installs and configures the external command file.

```
sudo make install-commandmode
```

#### Install Configuration Files

This installs the *SAMPLE* configuration files. These are required as Nagios needs some configuration files to allow it to start.

```
sudo make install-config
```

#### Install Apache Config Files

This installs the Apache web server configuration files.

```
sudo make install-webconf
```

As well as installing the Apache web server configuration files, there are additional changes required to the Apache config file **httpd.conf** to enable modules.

Commands that do all of these changes are below, but first is a list of the changes being made.

Change this:

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

To this:

```
<IfModule dir_module>
    DirectoryIndex index.php index.html index.htm AddType application/x-httpd-php .phpAddType application/x-httpd-php-source .phps
</IfModule>
```

Add these lines to the end of the file:

```
Include conf/extra/nagios.conf
Include conf/extra/mod_php70.conf
```

Execute the following commands to make the changes described above:

```
sudo cp /opt/local/etc/php70/php.ini-production /opt/local/etc/php70/php.ini
cd /opt/local/apache2/modules/
sudo /opt/local/apache2/bin/apxs -a -e -n php7 mod_php70.so
printf '\nInclude conf/extra/nagios.conf\n' | sudo tee -a /opt/local/apache2/conf/httpd.conf
printf '\nInclude conf/extra/mod_php70.conf\n' | sudo tee -a /opt/local/apache2/conf/httpd.conf
sudo sed -i '' 's/DirectoryIndex index.html/DirectoryIndex index.php index.html index.htm AddType application\/x-httpd-php .phpAddType application\/x-httpd-php-source .phps/g' /opt/local/apache2/conf/httpd.conf
```

#### Configure Firewall

The firewall in OS X is turned off by default. Please refer to the Apple documentation for information on how to enable or configure TCP port 80 inbound.

#### Create nagiosadmin User Account

You'll need to create an Apache user account to be able to log into Nagios.

The following command will create a user account called nagiosadmin and you will be prompted to provide a password for the account.

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

When adding additional users in the future, you need to remove **-c** from the above command otherwise it will replace the existing nagiosadmin user *(and any other users you may have added)*.

#### Start Apache Web Server

```
sudo /opt/local/bin/port load apache2
```

#### Start Service / Daemon

This command starts Nagios Core.

```
sudo /etc/rc.d/init.d/nagios start
```

#### Test Nagios

Nagios is now running, to confirm this you need to log into the Nagios Web Interface.

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

`http://10.25.5.143/nagios`

`http://core-013.domain.local/nagios`

You will be prompted for a username and password. The username is `nagiosadmin` (you created it in a previous step) and the password is what you provided earlier.

Once you have logged in you are presented with the Nagios interface. Congratulations you have installed Nagios Core.

#### BUT WAIT ...

Currently you have only installed the Nagios Core engine. You'll notice some errors under the hosts and services along the lines of:

```
(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_load, ...) failed. errno is 2: No such file or directory
```

These errors will be resolved once you install the Nagios Plugins, which is covered in the next step.

#### Installing The Nagios Plugins

Nagios Core needs plugins to operate properly. These steps are going to install the Nagios Plugins that are available via MacPorts. Future versions of the Nagios Plugins package will be updated to support OS X, until then the version available via MacPorts will be fine.

These get installed to `/opt/local/libexec/nagios/` so the `/usr/local/nagios/etc/`**resource.cfg** file also needs to be updated and Nagios restarted (covered in the steps below).

```
sudo /opt/local/bin/port install nagios-plugins
sudo sed -i '' 's/\/usr\/local\/nagios\/libexec/\/opt\/local\/libexec\/nagios/g' /usr/local/nagios/etc/resource.cfg
sudo /etc/rc.d/init.d/nagios stop
sudo /etc/rc.d/init.d/nagios start
```

#### Test Plugins

Point your web browser to the ip address or FQDN of your Nagios Core server, for example:

```
http://10.25.5.143/nagios
```

```
http://core-013.domain.local/nagios
```

Go to a host or service object and "Re-schedule the next check" under the Commands menu. The error you previously saw should now disappear and the correct output will be shown on the screen.

#### Service / Daemon Commands

These commands are for starting / stopping / restarting / status Nagios.

```
sudo /etc/rc.d/init.d/nagios start
sudo /etc/rc.d/init.d/nagios stop
sudo /etc/rc.d/init.d/nagios restart
sudo /etc/rc.d/init.d/nagios status
```

## Final Thoughts

For any support related questions please visit the Nagios Support Forums at:

http://support.nagios.com/forum/

---

Article ID: 96
Created On: Wed, May 20, 2015 at 2:07 PM
Last Updated On: Tue, Jan 15, 2019 at 9:22 AM
Authored by: **jolson**

Online URL: https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html