# Nagios

# check_mk

- download new version: https://checkmk.com/download.php?HTML=yes
- # yum/zypper install <##>
- # omd stop
- # omd update
- # omd start

# Introduction

This guide is intended to provide you with simple instructions on how to install Nagios from source (code) on openSUSE and have it monitoring your local machine inside of 20 minutes. No advanced installation options are discussed here - just the basics that will work for 95% of users who want to get started.

These instructions were written based on an openSUSE 10.2 installation.

# Required Packages

Make sure you've installed the following packages on your openSUSE installation before continuing. You can use yast to install packages under openSUSE.

- apache2
- C/C++ development libraries

## Create Account Information

Become the root user.

```
su -l
```

Create a new nagios user account and give it a password.

```
/usr/sbin/useradd -m nagios
passwd nagios
```

Create a new nagios group. Add the nagios user to the group.

```
/usr/sbin/groupadd nagios
/usr/sbin/usermod -G nagios nagios
```

Create a new nagcmd group for allowing external commands to be submitted through the web interface. Add both the nagios user and the apache user to the group.

```
/usr/sbin/groupadd nagcmd
/usr/sbin/usermod -a -G nagcmd nagios
/usr/sbin/usermod -a -G nagcmd wwwrun
```

# Download Nagios and the Plugins

Create a directory for storing the downloads.

```
mkdir ~/downloads
cd ~/downloads
```

Download the source code tarballs of both Nagios and the Nagios plugins (visit http://www.nagios.org/download/ for links to the latest versions). These directions were tested with Nagios 3.1.1 and Nagios Plugins 1.4.11.

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz
wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.11.tar.gz
```

# Compile and Install Nagios

Extract the Nagios source code tarball.

```
cd ~/downloads
tar xzf nagios-3.2.3.tar.gz
cd nagios-3.2.3
```

Run the Nagios configure script, passing the name of the group you created earlier like so:

```
./configure --with-command-group=nagcmd
```

Compile the Nagios source code.

```
make all
```

Install binaries, init script, sample config files and set permissions on the external command directory.

```
make install
```
```
make install-init
```
```
make install-config
```
```
make install-commandmode
```

Don't start Nagios yet - there's still more that needs to be done...

# Customize Configuration

Sample configuration files have now been installed in the /usr/local/nagios/etc directory. These sample files should work fine for getting started with Nagios. You'll need to make just one change before you proceed...

Edit the /usr/local/nagios/etc/objects/contacts.cfg config file with your favorite editor and change the email address associated with the nagiosadmin contact definition to the address you'd like to use for receiving alerts.

```
vi /usr/local/nagios/etc/objects/contacts.cfg
```

# Configure the Web Interface

Install the Nagios web config file in the Apache conf.d directory.

```
make install-webconf
```

Create a nagiosadmin account for logging into the Nagios web interface. Remember the password you assign to this account - you'll need it later.

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

Restart Apache to make the new settings take effect.

```
service apache2 restart
```
```
or
```
```
systemctl restart apache2
```

Note Note: Consider implementing the ehanced CGI security measures described below to ensure that your web authentication credentials are not compromised.

# Compile and Install the Nagios Plugins

Extract the Nagios plugins source code tarball.

```
cd ~/downloads
```
```
tar xzf nagios-plugins-1.4.11.tar.gz
```
```
cd nagios-plugins-1.4.11
```

Compile and install the plugins.

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

# Start Nagios

Add Nagios to the list of system services and have it automatically start when the system boots.

```
chkconfig --add nagios
chkconfig nagios on
```

Verify the sample Nagios configuration files.

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

If there are no errors, start Nagios.

```
service nagios start
or
systemctl start nagios
```

# Login to the Web Interface

You should now be able to access the Nagios web interface at the URL below. You'll be prompted for the username (nagiosadmin) and password you specified earlier.

http://localhost/nagios/

Click on the "Service Detail" navbar link to see details of what's being monitored on your local machine. It will take a few minutes for Nagios to check all the services associated with your machine, as the checks are spread out over time.

# Other Modifications

Make sure your machine's firewall rules are configured to allow access to the web server if you want to access the Nagios interface remotely.

You can do this by:

- Opening the control center
- Select 'Open Administrator Settings' to open the YaST administrator control center
- Select 'Firewall' from the 'Security and Users' category
- Click the 'Allowed Services' option in the Firewall Configuration window

- Add 'HTTP Server' to the allowed services list for the 'External Zone'
- Click 'Next' and 'Accept' to activate the new firewall settings

Configuring email notifications is outside the scope of this documentation. Refer to your system documentation, search the web, or look to the Nagios Support Portal or Nagios Community Wiki for specific instructions on configuring your openSUSE system to send email messages to external addresses.

# Security

This is intended to be an introduction for implementation of stronger authentication and server security focused around the CGI web interface.

**Stronger Authentication using Digest Authentication.**

If you have followed the quickstart guides, chances are that you are using Apache's Basic Authentication. Basic Authentication will send your username and password in "clear text" with every http request. Consider using a more secure method of authentication such as Digest Authentication which creates a MD5 Hash of your username and password to send with each request.

**Forcing TLS/SSL for all Web Communication.**

Apache provides TLS/SSL through the mod_ssl module. TLS/SSL provides a secure tunnel between the client and server that prevents eavesdropping and tampering using strong publickey/privatekey cryptography.

**Locking Down Apache Using Access Controls.**

Consider locking down access to the Nagios box to your IP address, IP address range, or IP subnet. If you require access outside your network you could use VPN or SSH Tunnels. This is a easy and strong to limit access to HTTP/HTTPS on your system.

# Implementing Digest Authentication

The implementation of Digest Authentication is simple. You will have to create the new type of password file using the 'htdigest' tool, then modify the Apache configuration for nagios (typically /etc/httpd/conf.d/nagios.conf).

Create a new passwords file using the 'htdigest' tool. The difference that you will notice if you are familiar with 'htpasswd' tools is the requirement to supply a 'realm' argument. Where 'realm' in this case refers to the value of the 'AuthName' directive in the Apache configuration.

`htdigest -c /usr/local/nagios/etc/.digest_pw "Nagios Access" nagiosadmin`

Next, edit the Apache configuration file for Nagios (typically /etc/httpd/conf.d/nagios.conf) using the following example.

```
## BEGIN APACHE CONFIG SNIPPET - NAGIOS.CONF

ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"

<Directory "/usr/local/nagios/sbin">

   Options ExecCGI

   AllowOverride None

   Order allow,deny

   Allow from all

   AuthType Digest

   AuthName "Nagios Access"

   AuthDigestFile /usr/local/nagios/etc/.digest_pw

   Require valid-user

</Directory>



Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">

   Options None

   AllowOverride None

   Order allow,deny
```

```
    Allow from all

    AuthType Digest

    AuthName "Nagios Access"

    AuthDigestFile /usr/local/nagios/etc/.digest_pw

    Require valid-user

</Directory>

## END APACHE CONFIG SNIPPETS
```

Then, restart the Apache service so the new settings can take effect.

`/etc/init.d/httpd restart`

# Implementing Forced TLS/SSL

Make sure you've installed Apache and OpenSSL. By default you should have mod_ssl support if you are still having trouble you may find help reading Apache's TLS/SSL Encryption Documentation.

Next, verify that TLS/SSL support is working by visiting your Nagios Web Interface using HTTPS (https://your.domain/nagios). If it is working you can continue on to the next steps that will force using HTTPS and block all HTTP requests for the Nagios Web Interface. If you are having trouble visit Apache's TLS/SSL Encryption Documentation and Google for troubleshooting your specific Apache installation. Next, edit the Apache configuration file for Nagios (typically /etc/httpd/conf.d/nagios.conf) by adding the 'SSLRequireSSL' directive to both the 'sbin' and 'share' directories.

```
## BEGIN APACHE CONFIG SNIPPET - NAGIOS.CONF

ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"

<Directory "/usr/local/nagios/sbin">

    ...

    SSLRequireSSL
```

```
   ...

</Directory>



Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">

   ...

   SSLRequireSSL

   ...

</Directory>

## END APACHE CONFIG SNIPPETS
```

Restart the Apache service so the new settings can take effect.

`/etc/init.d/httpd restart`

# Implementing IP subnet lockdown

The following example will show how to lock down Nagios CGIs to a specific IP address, IP address range, or IP subnet using Apache's access controls.

Edit the Apache configuration file for Nagios (typically /etc/httpd/conf.d/nagios.conf) by using the 'Allow', 'Deny', and 'Order' directives using the following as an example.

```
## BEGIN APACHE CONFIG SNIPPET - NAGIOS.CONF

ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"

<Directory "/usr/local/nagios/sbin">

   ...
```

```
   AllowOverride None
   Order deny,allow
   Deny from all
   Allow from 127.0.0.1 10.0.0.25       # Allow single IP addresses
   Allow from 10.0.0.0/255.255.255.0        # Allow network/netmask pair
   Allow from 10.0.0.0/24           # Allow network/nnn CIDR spec
   ...
</Directory>


Alias /nagios "/usr/local/nagios/share"
<Directory "/usr/local/nagios/share">
   ...
   AllowOverride None
   Order deny,allow
   Deny from all
   Allow from 127.0.0.1 10.0.0.25       # Allow single IP addresses
   Allow from 10.0.0.0/255.255.255.0        # Allow network/netmask pair
   Allow from 10.0.0.0/24           # Allow network/nnn CIDR spec
   ...
</Directory>
```

```
## END APACHE CONFIG SNIPPET
```

Important Notes

- Digest Authentication sends data in the clear but not your username and password.
- Digest Authentication is not as universally supported as Basic Authentication.
- TLS/SSL has potential for "man-in-the-middle attacks". MITM attacks are vulnerable if an attacker is able to insert itself between the server and client such as in a Phishing attack, ISP monitoring, or corporate LAN firewall certificate resigning. So read up on certificate verification!
- Apache access controls only protect the HTTP/HTTPS protocols. Look into IPtables for strong system wide firewall control.
- Most importantly, Security is a moving target so stay informed and do research! Perhaps by listening to a Podcast such as "Security Now!".

---

Revision #3
Created 2026-04-01 17:13:34 CEST by Philip
Updated 2026-04-13 19:27:21 CEST by Philip