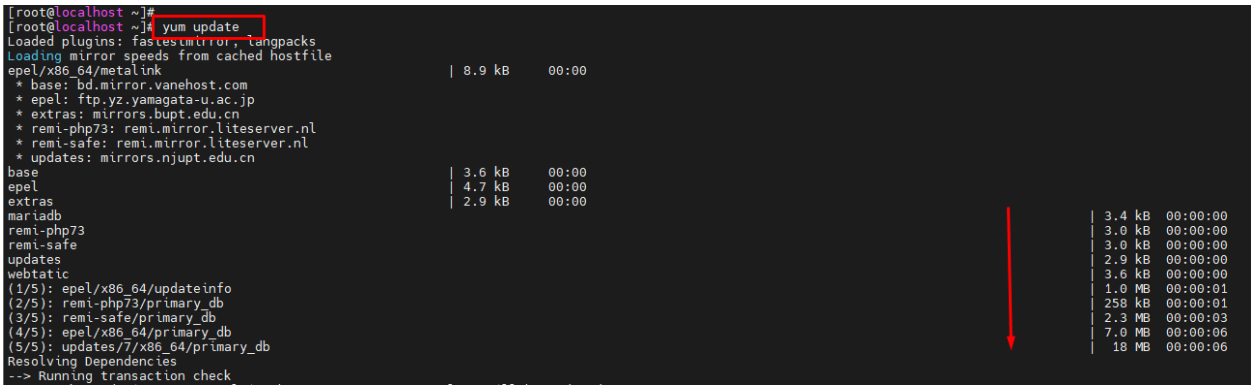
Install Nagios Core on CentOS 7

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



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:

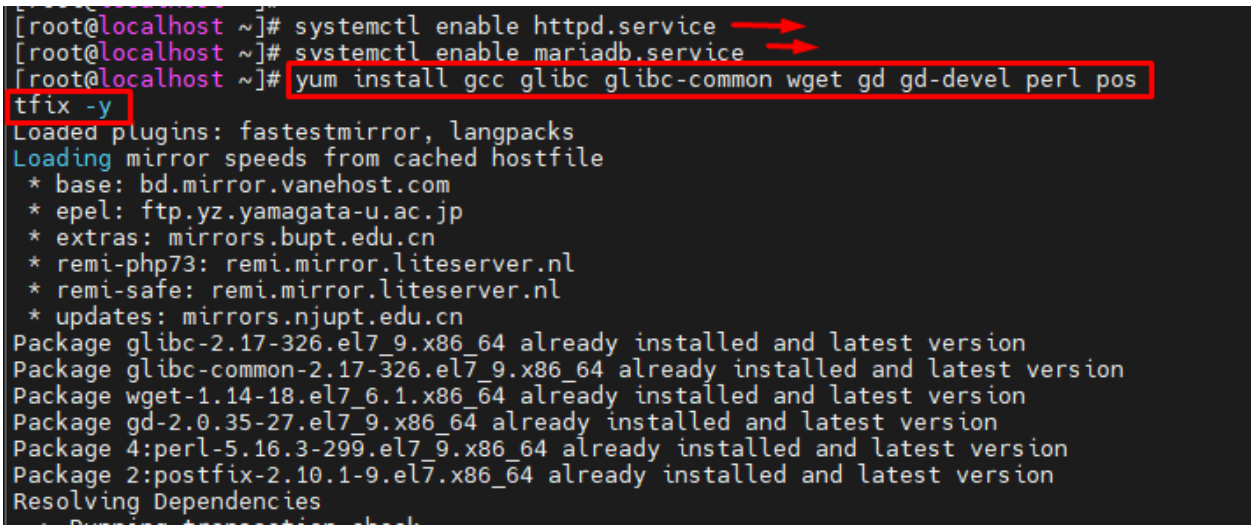
[root@localhost ~]# yum install httpd mariadb-server php php-mysql -y

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:

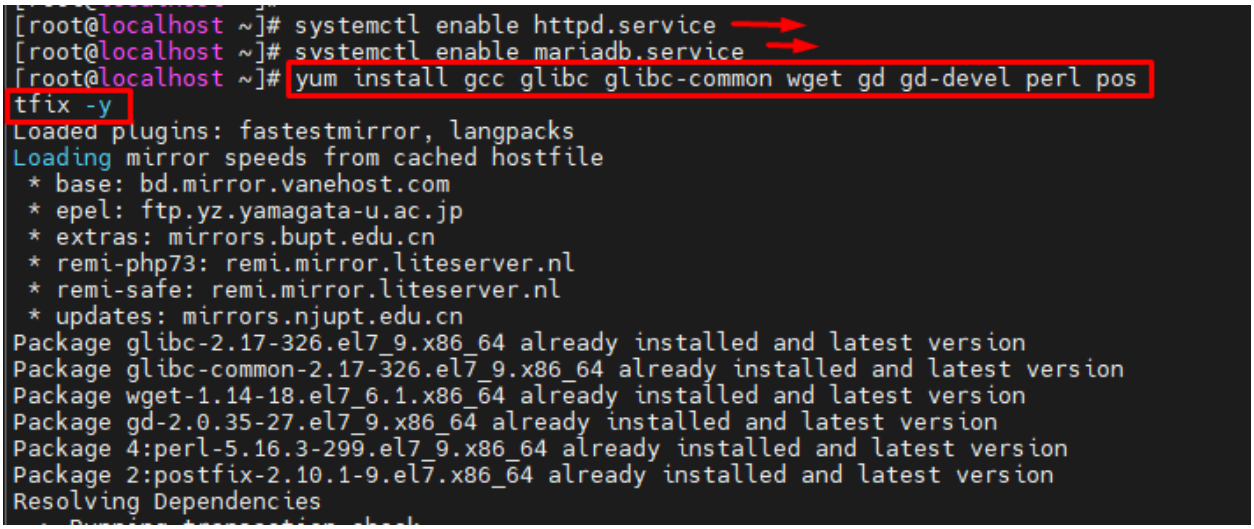
```
[root@localhost ~]# systemctl enable httpd.service

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



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
```



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
onnected.
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
```

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 ../commo
n/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o query-handler.o query-han
dler.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/mac
ros.c
```

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

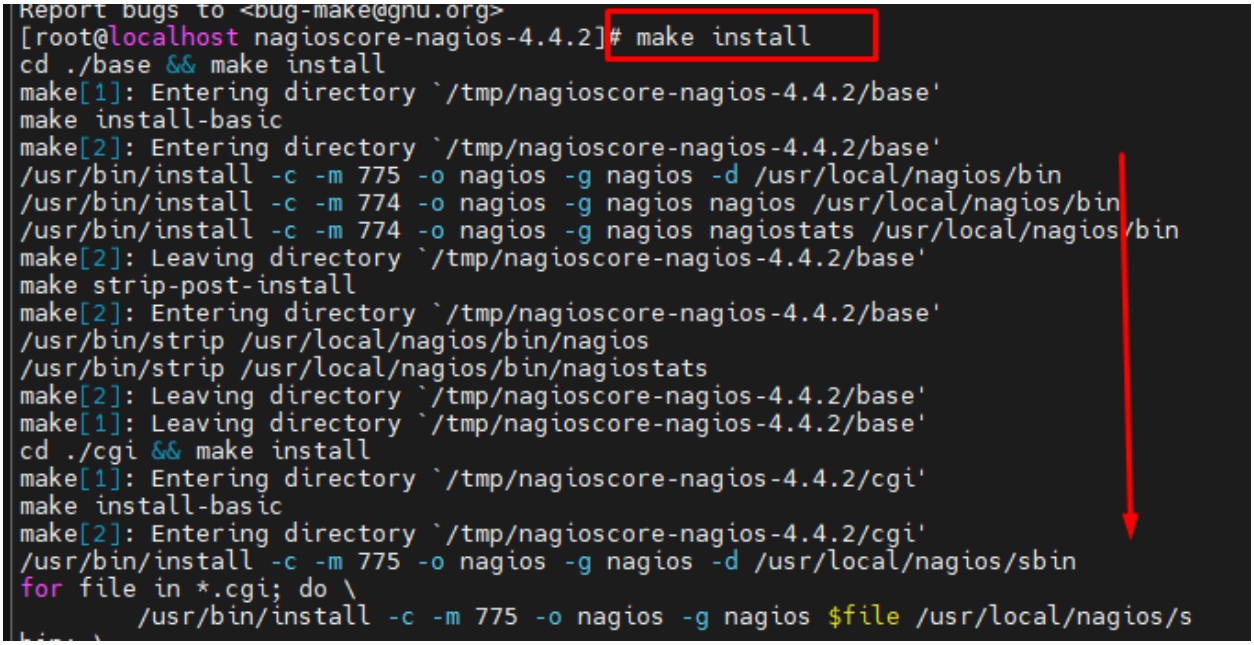
[root@localhost nagioscore-nagios-4.4.2]# make install-groups-users

[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
```

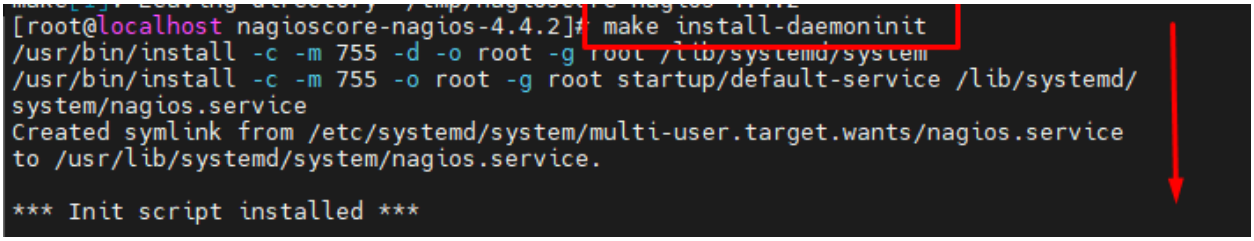
Now, you can finally run the following command in order to install Nagios Core:

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



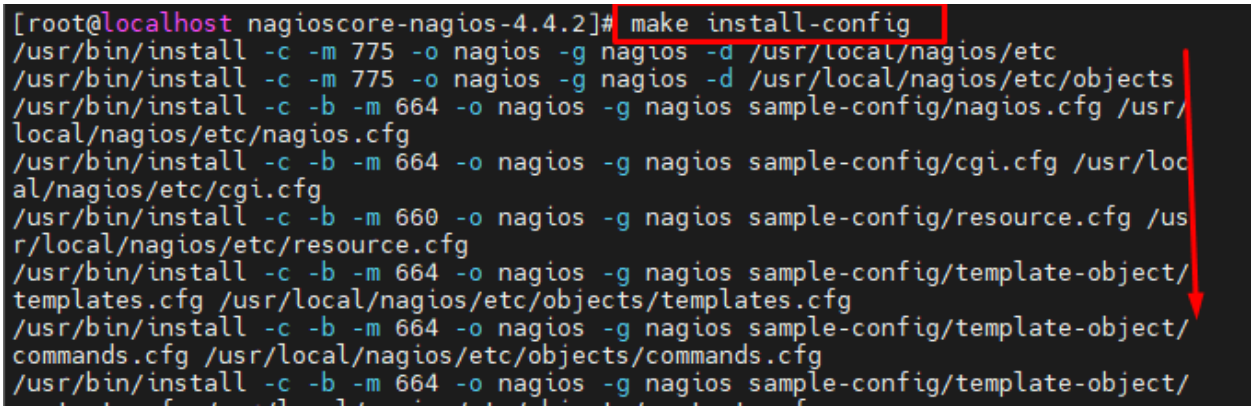
To, install the initialization script which can be used to manage your Nagios service, run the following command:

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



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

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



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.con
f
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.co
nf; \
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
```

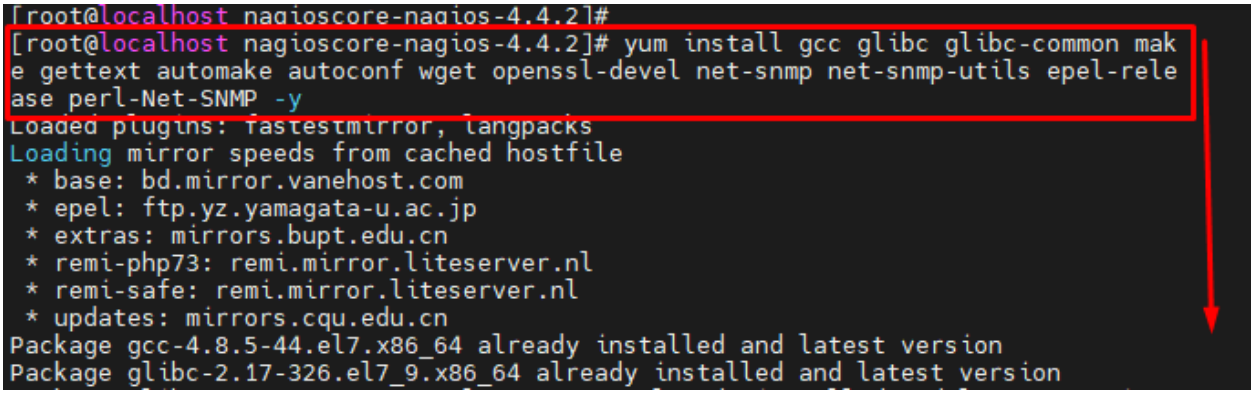
```
[root@localhost nagioscore-nagios-4.4.2]# systemctl restart httpd
[root@localhost nagioscore-nagios-4.4.2]# htpasswd -c /usr/local/nagios/etc/htpa
sswd.users nagiosadmin
New password:
123
Re-type new password:
Adding password for user nagiosadmin
[root@localhost nagioscore-nagios-4.4.2]#
```

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

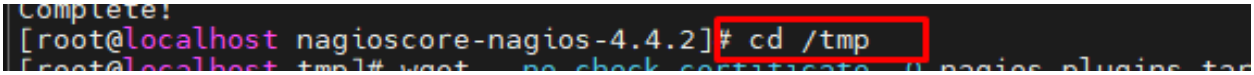
Before you download and install the Nagios plugins, you need to make sure that the following packages are installed on your CentOS 7 server:

[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

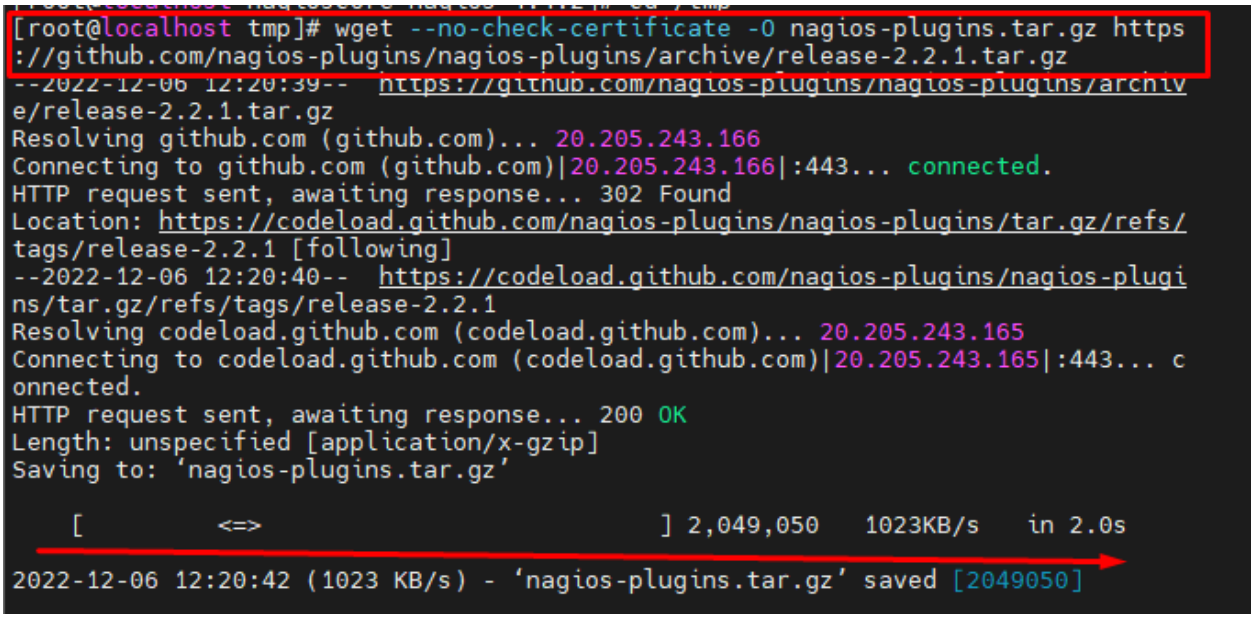


To download and extract the latest version of the Nagios Plugins to the tmp directory on your server, run the following commands:

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



[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]#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/bi
n)
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]# 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
which: no gnumake in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bi
n)
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

```
[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 ; \
```

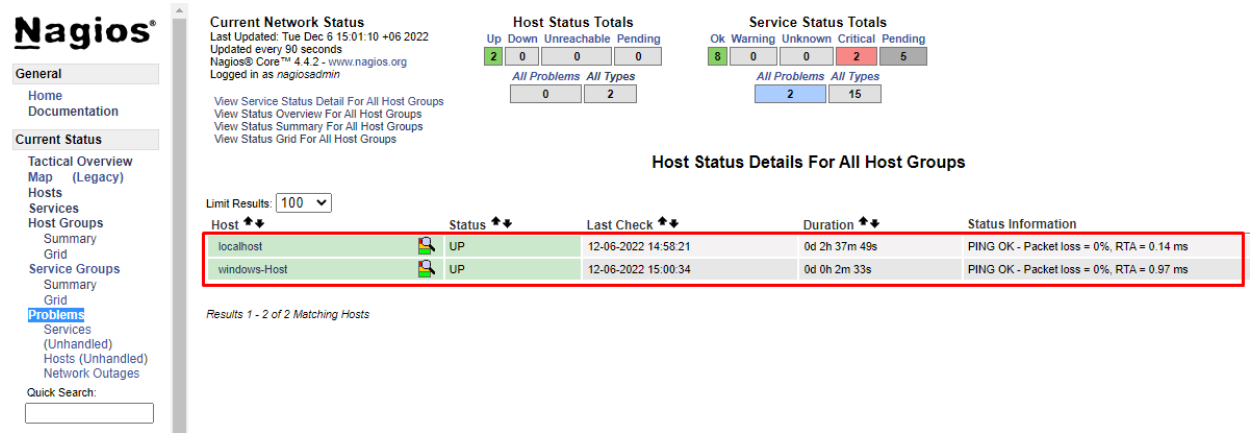
Step 7: Accessing Nagios Core

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:

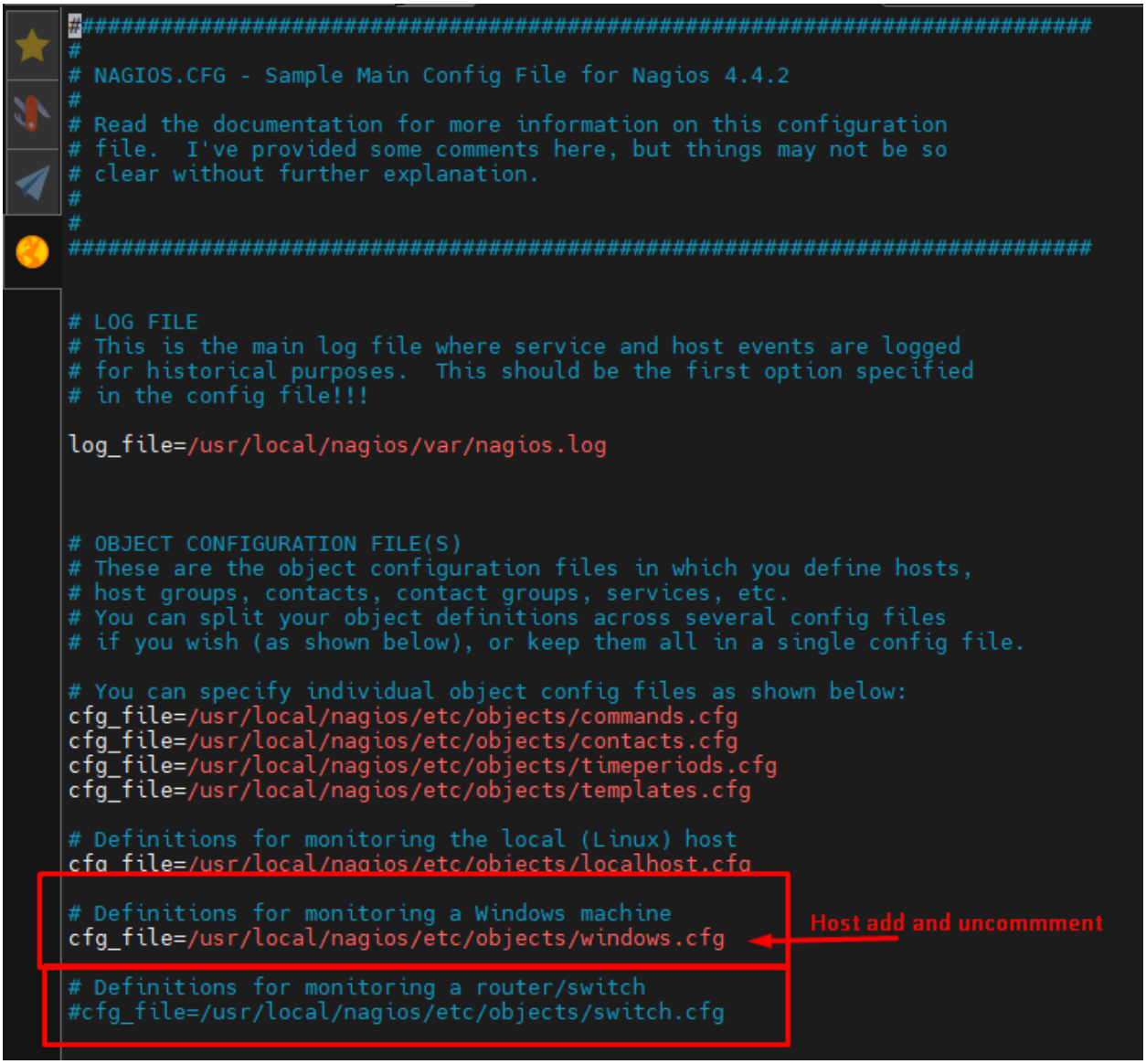
[root@localhost nagios-plugins-release-2.2.1]#systemctl start nagios

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:




```
[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
```



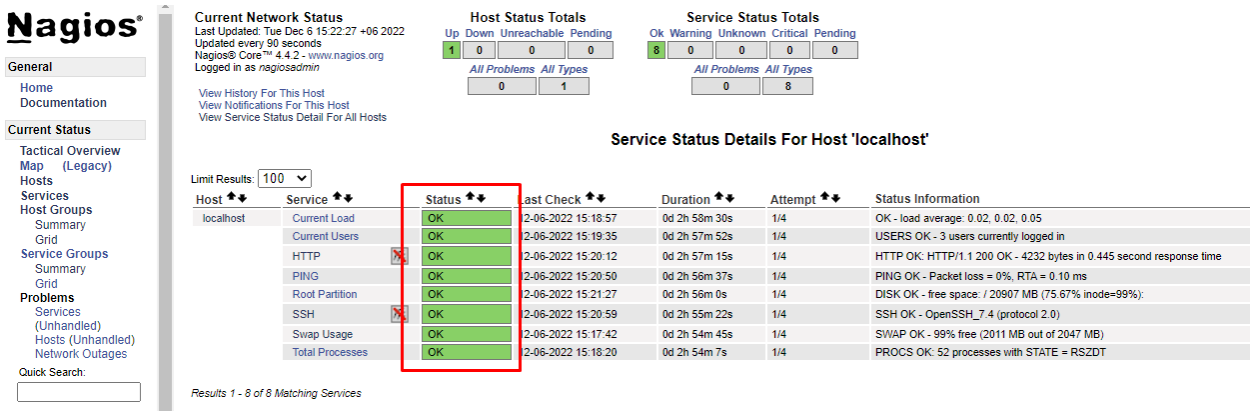
[root@localhost etc]# cd objects/

[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
```


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 15:22:27 +06 2022
Updated every 90 seconds
Nagios® Core™ 4.4.2 - 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: 100
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