Aim: Installing and Configuring Nagios for Docker Monitoring

Objective:

To set up a functional Nagios Core monitoring system.

To monitor Docker containers and key performance metrics (CPU, memory, container status).

To configure Nagios plugins and scripts for Docker integration.

To generate alerts for container or service failures.

Description:

Nagios is an open-source monitoring tool used to track the health of IT infrastructure. In this project, Nagios will be set up to monitor Docker containers running on a host machine. Since Nagios does not natively support Docker, this requires integrating it with custom scripts or plugins (e.g., check_docker or check_docker_container). The goal is to ensure visibility into the containerized environment and automate alerting in case of issues.

Implementation:

1. Environment Setup:

- *OS: Ubuntu 20.04 (or similar Linux distro)*
- *Docker Engine installed and running*
- *Nagios Core 4.x*

2. Installing Nagios Core:

sudo apt update

sudo apt install -y autoconf gcc make libgd-dev libmcrypt-dev libssl-dev dc build-essential snmp

sudo useradd nagios

sudo groupadd nagcmd

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd www-data

 # Download and compile Nagios Core

cd /tmp

wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz

tar xzf nagios-4.4.6.tar.gz

cd nagios-4.4.6

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

make all

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

sudo make install-webconf

3. Installing Nagios Plugins:

cd /tmp

wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz

tar xzf nagios-plugins-2.3.3.tar.gz

cd nagios-plugins-2.3.3

./configure --with-nagios-user=nagios --with-nagios-group=nagios

make

sudo make install

4. Configure Web Access:

- *Create Nagios admin user:*

5. Install Docker Monitoring Plugin:

- *Download a plugin like check_docker or create a custom script.*
- *Place it in /usr/local/nagios/libexec/*
- *Make it executable:*

chmod +x /usr/local/nagios/libexec/check_docker

6. Add Docker Checks to Nagios Configuration:

- *Edit commands.cfg:*

```
define command {
  command_name    check_docker_container
  command_line    $USER1$/check_docker -C $ARG1$
}
```

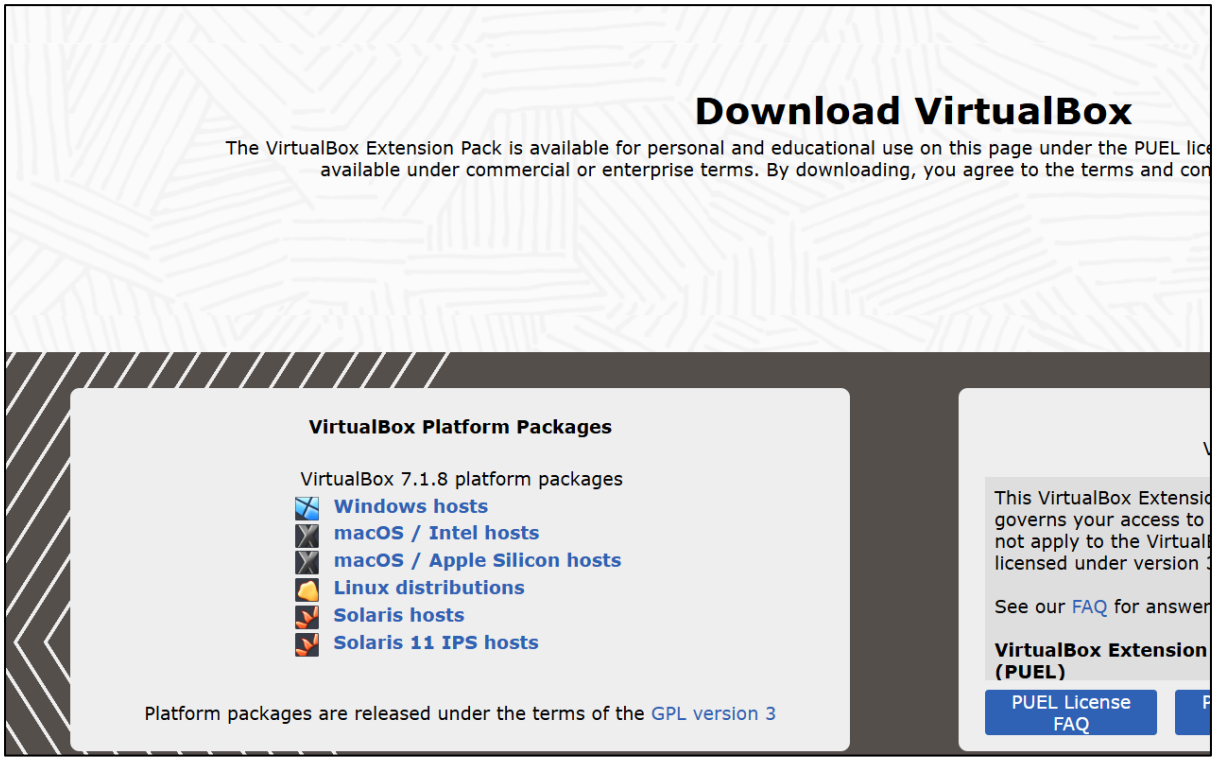
Add a host and service for Docker in nagios.cfg:

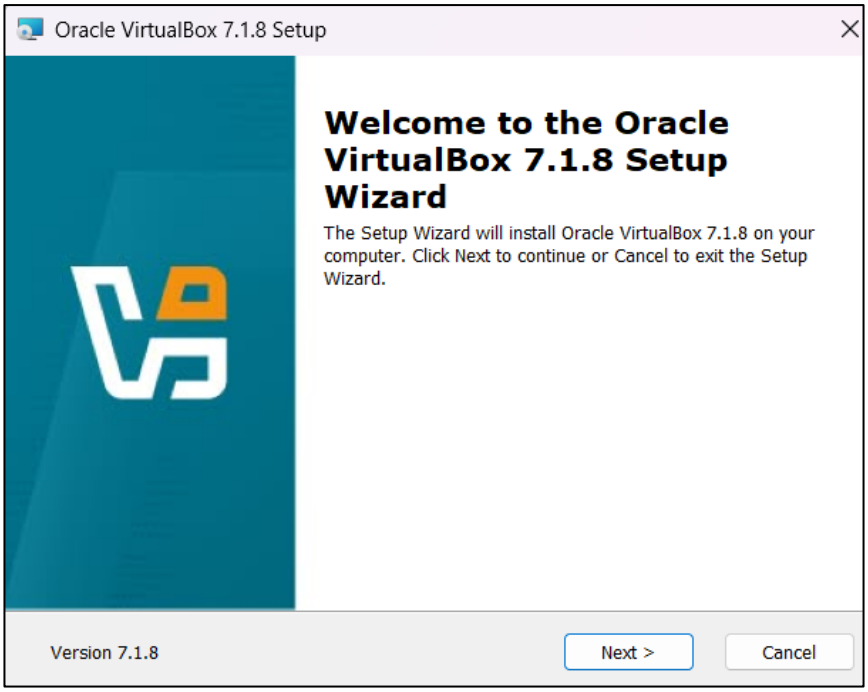
```
define host {
  use             linux-server
  host_name       DockerHost
  address         127.0.0.1
}

define service {
  use                 generic-service
  host_name           DockerHost
  service_description Docker Container Status
  check_command       check_docker_container!nginx
}
```

7. Restart Nagios:

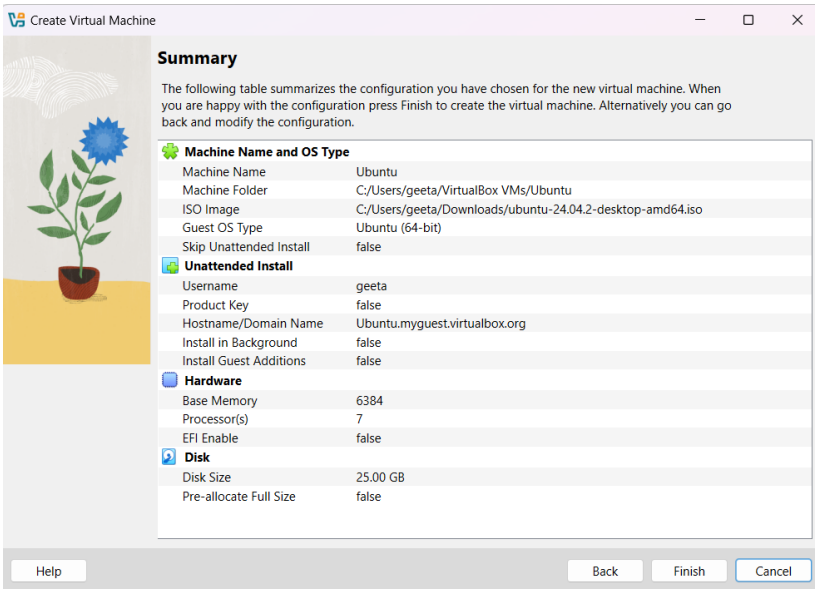
sudo systemctl restart nagios

https://www.virtualbox.org/wiki/Downloads





https://ubuntu.com/download/desktop




```
geeta@Ubuntu:/tmp$ sudo apt install apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapr1t64 libaprutil1t64
The following NEW packages will be installed:
  apache2-utils libapr1t64 libaprutil1t64
0 upgraded, 3 newly installed, 0 to remove and 1 not upgraded.
Need to get 297 kB of archives.
After this operation, 907 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

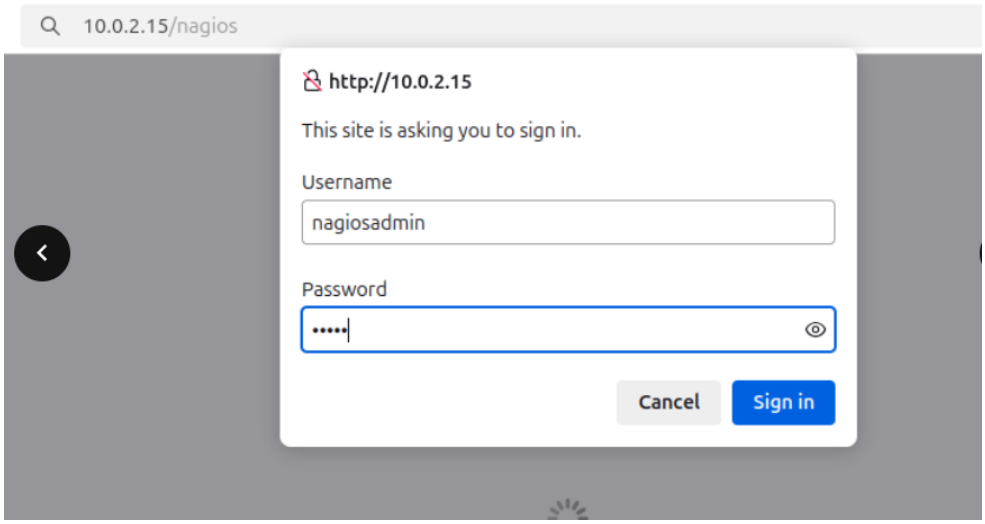
```
geeta@Ubuntu: ~/Downloads/nagios-4.5.9
geeta@Ubuntu:~/Downloads$ ls
nagios-4.5.9  nagios-4.5.9.tar.gz
geeta@Ubuntu:~/Downloads$ tar -xzvf nagios-4.5.9.tar.gz
nagios-4.5.9/
nagios-4.5.9/.github/
nagios-4.5.9/.github/workflows/
nagios-4.5.9/.github/workflows/test.yml
nagios-4.5.9/.gitignore
nagios-4.5.9/CONTRIBUTING.md
nagios-4.5.9/Changelog
nagios-4.5.9/INSTALLING
nagios-4.5.9/LEGAL
nagios-4.5.9/LICENSE
nagios-4.5.9/Makefile.in
nagios-4.5.9/README.md
nagios-4.5.9/THANKS
nagios-4.5.9/UPGRADING
nagios-4.5.9/aclocal.m4
nagios-4.5.9/autoconf-macros/
nagios-4.5.9/autoconf-macros/.gitignore
nagios-4.5.9/autoconf-macros/CHANGELOG.md
nagios-4.5.9/autoconf-macros/LICENSE
nagios-4.5.9/autoconf-macros/LICENSE.md
nagios-4.5.9/autoconf-macros/README.md
```

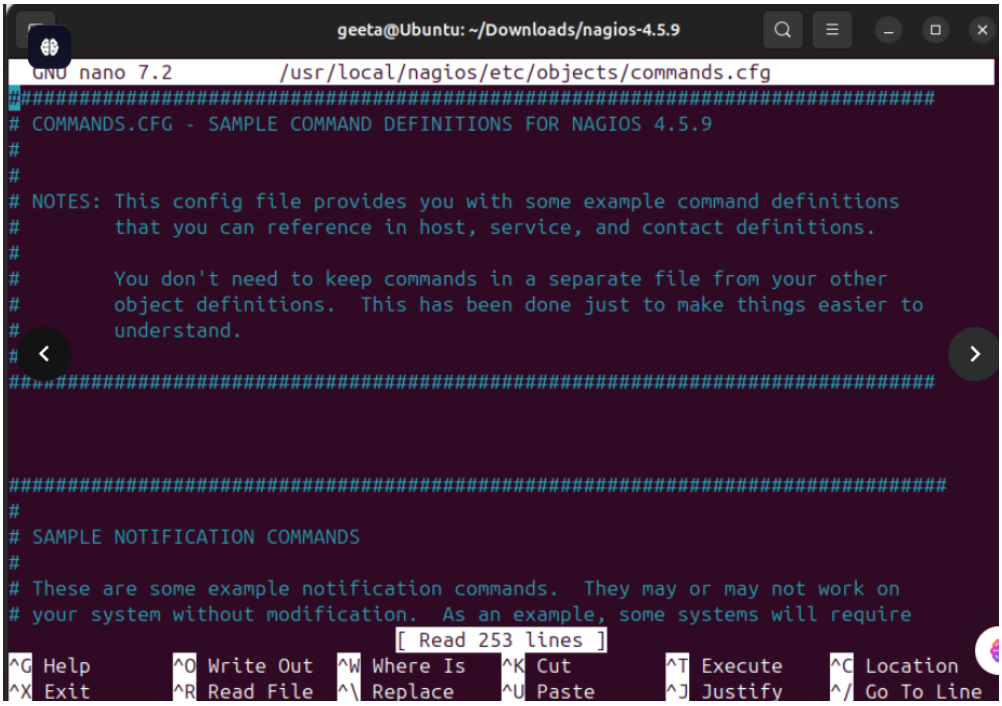
```
geeta@Ubuntu:~/Downloads/nagios-4.5.9$ sudo apt update
sudo apt install -y build-essential libgd-dev libapache2-mod-php php7.4 php7.4-cli
php7.4-mbstring unzip wget curl libssl-dev apache2
Hit:1 http://in.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package php7.4 is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
```

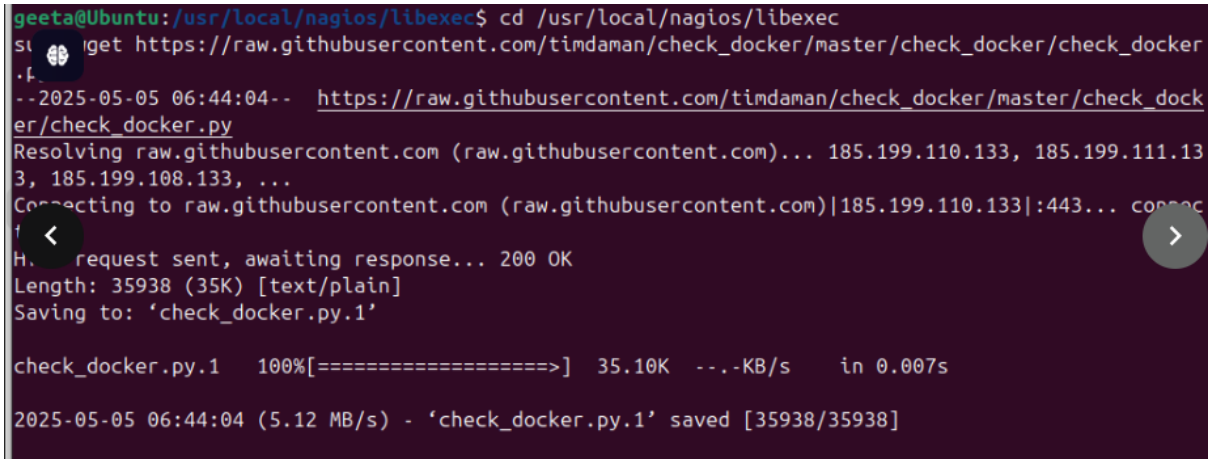
```
geeta@Ubuntu:~/Downloads/nagios-4.5.9$ sudo make all
cd ./base && make
make[1]: Entering directory '/home/geeta/Downloads/nagios-4.5.9/base'
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2 -I/usr/include/
openssl -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
./nagios.c: In function ‘main’:
./nagios.c:611:25: warning: ignoring return value of ‘asprintf’ declared with attri
bute ‘warn_unused_result’ [-Wunused-result]
  611 |                         asprintf(&mac->x[MACRO_PROCESSSTARTTIME], "%llu", (
unsigned long long)program_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
./nagios.c:841:25: warning: ignoring return value of ‘asprintf’ declared with attri
bute ‘warn_unused_result’ [-Wunused-result]
  841 |                         asprintf(&mac->x[MACRO_EVENTSTARTTIME], "%llu", (un
signed long long)event_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

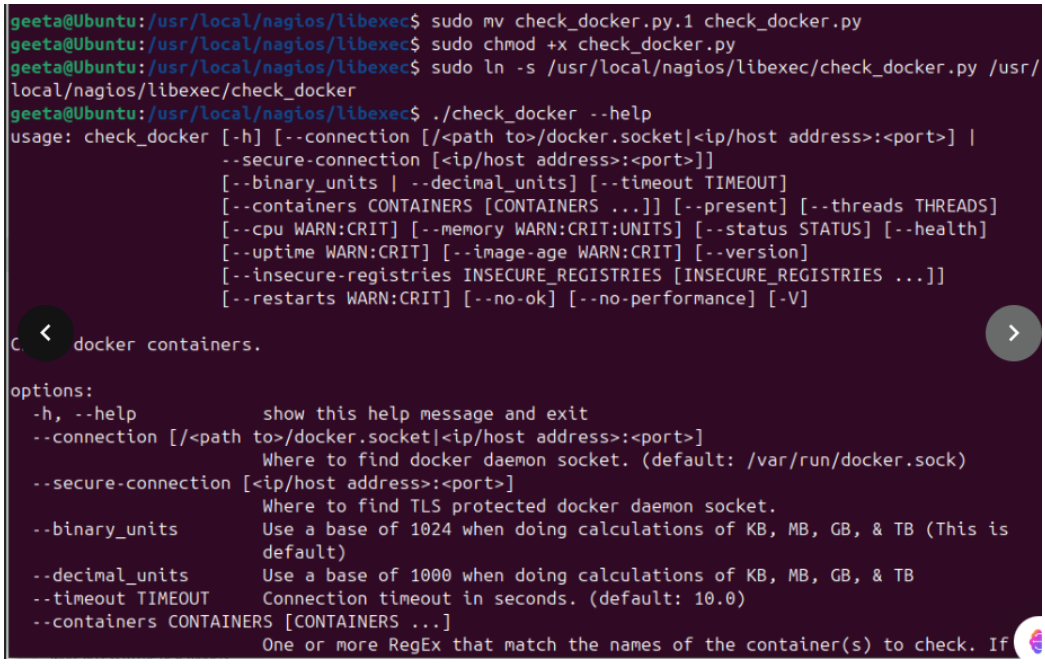
```
I    ou have questions about configuring or running Nagios,
Please make sure that you:

     - Look at the sample config files
     - Read the documentation on the Nagios Library at:
          https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

     - What version of Nagios you are using
     - What version of the plugins you are using
     - Relevant snippets from your config files
     - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

--------------------------------------------------------

Enjoy.
```

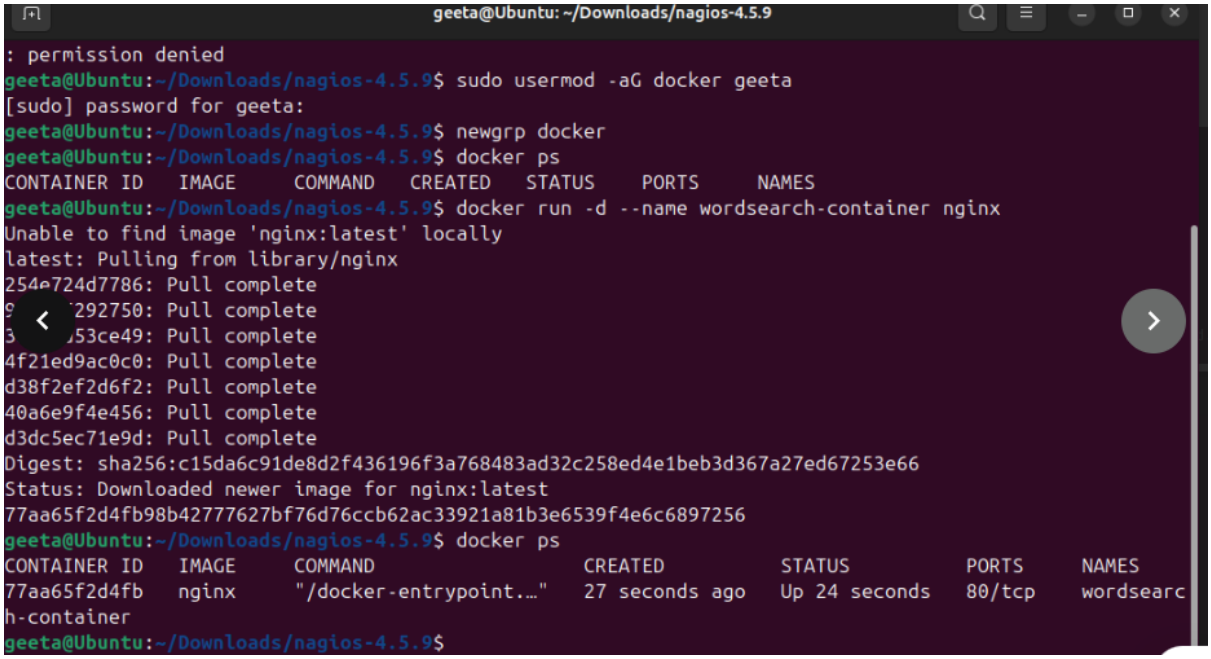
```
services, hosts, etc. to fit your particular needs.

/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 ***

/    bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available
ios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/na
gios.conf; \
fi

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

geeta@Ubuntu:~/Downloads/nagios-4.5.9$
```

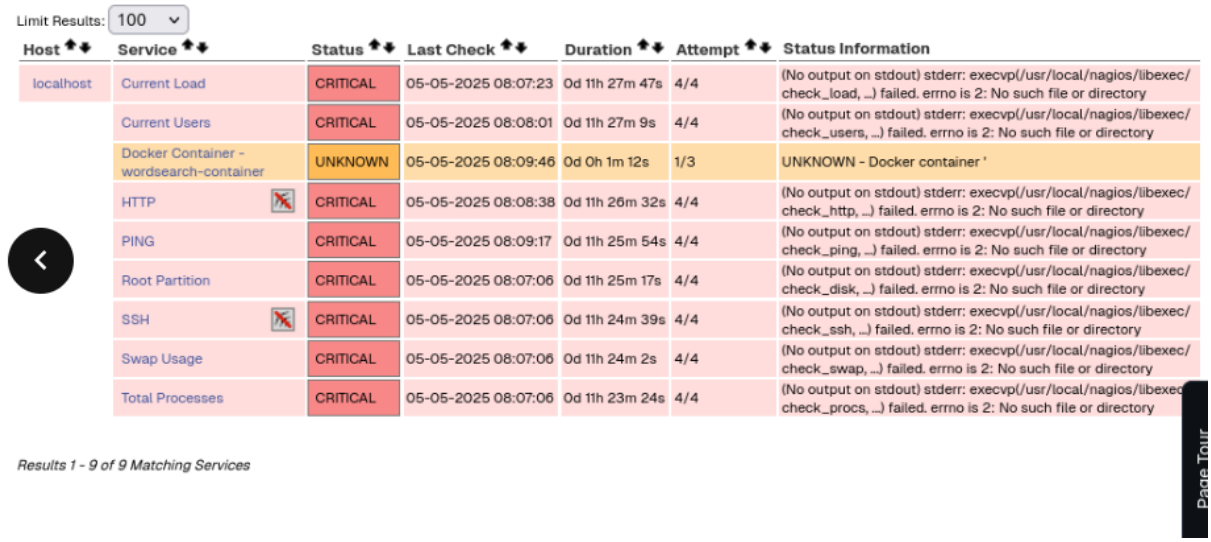



geeta@Ubuntu: ~/Downloads/nagios-4.5.9
GNU nano 7.2 /usr/local/nagios/etc/objects/commands.cfg
COMMANDS.CFG - SAMPLE COMMAND DEFINITIONS FOR NAGIOS 4.5.9
NOTES: This config file provides you with some example command definitions
that you can reference in host, service, and contact definitions.
You don't need to keep commands in a separate file from your other
object definitions. This has been done just to make things easier to
understand.
SAMPLE NOTIFICATION COMMANDS
These are some example notification commands. They may or may not work on
your system without modification. As an example, some systems will require
[Read 253 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line


geeta@Ubuntu:/usr/local/nagios/libexec$ cd /usr/local/nagios/libexec
get https://raw.githubusercontent.com/timdaman/check_docker/master/check_docker/check_docker
--2025-05-05 06:44:04-- https://raw.githubusercontent.com/timdaman/check_docker/master/check_docker/check_docker.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
request sent, awaiting response... 200 OK
Length: 35938 (35K) [text/plain]
Saving to: 'check_docker.py.1'
check_docker.py.1 100%[==========>] 35.10K --.-KB/s in 0.007s
2025-05-05 06:44:04 (5.12 MB/s) - 'check_docker.py.1' saved [35938/35938]


geeta@Ubuntu:/usr/local/nagios/libexec$ sudo mv check_docker.py.1 check_docker.py
geeta@Ubuntu:/usr/local/nagios/libexec$ sudo chmod +x check_docker.py
geeta@Ubuntu:/usr/local/nagios/libexec$ sudo ln -s /usr/local/nagios/libexec/check_docker.py /usr/local/nagios/libexec/check_docker
geeta@Ubuntu:/usr/local/nagios/libexec$ ./check_docker --help
usage: check_docker [-h] [--connection [/<path to>/docker.socket|<ip/host address>:<port>] |
--secure-connection [<ip/host address>:<port>]]
[--binary_units | --decimal_units] [--timeout TIMEOUT]
[--containers CONTAINERS [CONTAINERS ...]] [--present] [--threads THREADS]
[--cpu WARN:CRIT] [--memory WARN:CRIT:UNITS] [--status STATUS] [--health]
[--uptime WARN:CRIT] [--image-age WARN:CRIT] [--version]
[--insecure-registries INSECURE_REGISTRIES [INSECURE_REGISTRIES ...]]
[--restarts WARN:CRIT] [--no-ok] [--no-performance] [-V]
docker containers.
options:
-h, --help show this help message and exit
--connection [/<path to>/docker.socket|<ip/host address>:<port>]
Where to find docker daemon socket. (default: /var/run/docker.sock)
--secure-connection [<ip/host address>:<port>]
Where to find TLS protected docker daemon socket.
--binary_units Use a base of 1024 when doing calculations of KB, MB, GB, & TB (This is default)
--decimal_units Use a base of 1000 when doing calculations of KB, MB, GB, & TB
--timeout TIMEOUT Connection timeout in seconds. (default: 10.0)
--containers CONTAINERS [CONTAINERS ...]
One or more RegEx that match the names of the container(s) to check. If





Conclusion:

The integration of Nagios with Docker enhances visibility into containerized environments by providing real-time monitoring and proactive alerting. Through the installation and configuration steps, we successfully set up Nagios Core to track the status and performance of Docker containers using custom plugins. This setup helps ensure high availability and quick issue resolution by alerting administrators to container failures or resource bottlenecks. Overall, Nagios proves to be a powerful and flexible tool for monitoring modern, container-based infrastructure when paired with the right plugins and configurations.