

## **Nagios Core Server Installation and Configuration**

Nagios® Core™ is an Open Source system and network monitoring application. It watches hosts and services that you specify, alerting you when things go bad and when they get better.

Some of the many features of Nagios Core include:

- Monitoring of network services (SMTP, POP3, HTTP, NNTP, PING, etc.)
- Monitoring of host resources (processor load, disk usage, etc.)
- Simple plugin design that allows users to easily develop their own service checks
- Parallelized service checks
- Ability to define network host hierarchy using "parent" hosts, allowing detection of and distinction between hosts that are down and those that are unreachable
- Contact notifications when service or host problems occur and get resolved (via email, pager, or user-defined method)
- Ability to define event handlers to be run during service or host events for proactive problem resolution
- Automatic log file rotation
- Support for implementing redundant monitoring hosts
- Optional web interface for viewing current network status, notification and problem history, log file, etc.

### **System Requirements**

The only requirement of running Nagios Core is a machine running Linux (or UNIX variant) that has network access and a C compiler installed (if installing from source code).

A web server (preferably Apache)

Thomas Boutell's gd library version 1.6.3 or higher (required by the statusmap and trends CGIs)

### **Licensing**

Nagios Core is licensed under the terms of the GNU General Public License Version 2 as published by the Free Software Foundation. This gives you legal permission to copy, distribute and/or modify Nagios under certain conditions. Read the 'LICENSE' file in the Nagios distribution or read the online version of the license for more details.

### **Nagios Server Installation**

This document is designed to install Nagios core server on a Virtual Machine with Centos 9 installed. However the same steps may be followed on a physical machine also. These steps may also work on RedHat 9/ Rocky Linux 9 also.

Create a virtual machine with 1 or 2 CPU's , 4/8 GB RAM, Minimum 50 GB hard disk. Install Centos 9 on it. The Nagios core server provides a web based interface thus to access it a browser is required. Thus please install Centos 9 with GUI so the local browser can be used to access this web interface. After installation please set the timezone and date time properly. Also set proper hostname and IP Address. After installation logon to the OS. To install Nagios either root login or user with sudo permission is required. Here user with sudo permission is assumed. After logon, open a terminal and run following command.

```
sudo yum update -y
```

After the update finishes successfully, if required run following command to upgrade the system.

```
sudo yum upgrade -y
```

Once the system is updated and upgraded, Nagios installation can be started.

Install pre-requisite packages using following command.

1. `yum install -y gcc glibc glibc-common wget unzip httpd php php-cli gd gd-devel openssl-devel net-snmp perl perl-Net-SNMP -y`
2. `yum install -y make gettext autoconf net-snmp-utils epel-release automake`

Disable SELINUX

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

Download Nagios core source code and extract it.

1. `cd /tmp`
2. `wget -O nagioscore.tar.gz https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.2.tar.gz`
3. `tar xzf nagioscore.tar.gz`
4. `cd nagios-4.5.2/`

Compile and install Nagios Core Server

1. `./configure` # there is a .(dot) before slash.
2. `make all`
3. `make install-groups-users` #Create nagios user and group
4. `usermod -a -G nagios apache` # add nagios user to apache group
5. `make install` # installs nagios to /usr/local/nagios directory
6. `make install-commandmode` # Adds nagios commands
7. `make install-config` # Adds nagios configuration files
8. `make install-webconf` # Adds the apache configuration for nagios
9. `make install-daemoninit` # Adds nagios as a service to the system
10. `htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin`

# the above command creates a user in apache by the name nagiosadmin. You can provide any name. This command will prompt for a password. This username is used to login to the Nagios core server web interface.

Configure firewall to open http (port 80) port.

1. `firewall-cmd --zone=public --add-port=80/tcp` # opens port immediately
2. `firewall-cmd --zone=public --add-port=80/tcp --permanent` # Adds port in configuration file

Enable and start apache web server.

```
systemctl enable httpd
```

```
systemctl start httpd
```

The Nagios core server basic installation is complete with these steps. To verify that Apache web server and basic Nagios server are correctly installed, open web browser on the server and type the url <http://localhost/nagios>.

The site will prompt for a username and password. If you followed the steps mentioned in this document then the username is nagiosadmin. Provide the password given. On successful login the following web page should be displayed.



However still the Nagios service is not running as displayed in the above image. We need to configure and start the Nagios service.

### Install Nagios Plugins

1. `cd /tmp`
2. `sudo wget --no-check-certificate -O nagios-plugins.tar.gz` <https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz>
3. `tar xzf nagios-plugins.tar.gz`
4. `cd /tmp/nagios-plugins-release-2.2.1/`

5. `./tools/setup` *# there is a .(dot) in the beginning before /*
6. `./configure`
7. `make`
8. `make install`
9. `yum -y install nagios-plugins-nrpe` *# install NRPE plugin*

#### Configure the nagios server.

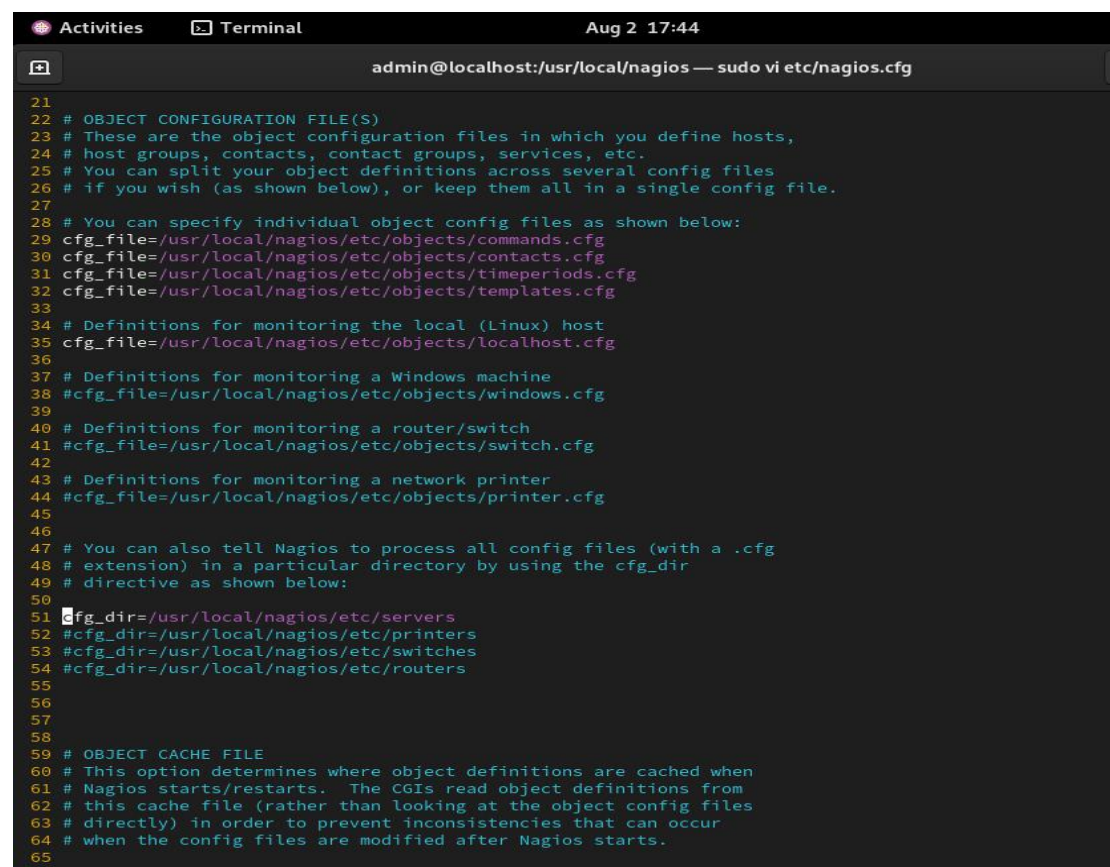
The nagios core server is installed in the `/usr/local/nagios` directory. Inside this directory the `etc` directory contains the nagios configuration files. The main configuration file is `nagios.cfg`.

Edit this file and enable servers directory. This will enable nagios to consider the configuration files stored in this server directory also.

```
cd /usr/local/nagios/etc
```

```
sudo vi nagios.cfg
```

In this file go to line number 51. Remove the `#` from this line as shown below.



```
21
22 # OBJECT CONFIGURATION FILE(S)
23 # These are the object configuration files in which you define hosts,
24 # host groups, contacts, contact groups, services, etc.
25 # You can split your object definitions across several config files
26 # if you wish (as shown below), or keep them all in a single config file.
27
28 # You can specify individual object config files as shown below:
29 cfg_file=/usr/local/nagios/etc/objects/commands.cfg
30 cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
31 cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
32 cfg_file=/usr/local/nagios/etc/objects/templates.cfg
33
34 # Definitions for monitoring the local (Linux) host
35 cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
36
37 # Definitions for monitoring a Windows machine
38 #cfg_file=/usr/local/nagios/etc/objects/windows.cfg
39
40 # Definitions for monitoring a router/switch
41 #cfg_file=/usr/local/nagios/etc/objects/switch.cfg
42
43 # Definitions for monitoring a network printer
44 #cfg_file=/usr/local/nagios/etc/objects/printer.cfg
45
46
47 # You can also tell Nagios to process all config files (with a .cfg
48 # extension) in a particular directory by using the cfg_dir
49 # directive as shown below:
50
51 cfg_dir=/usr/local/nagios/etc/servers
52 #cfg_dir=/usr/local/nagios/etc/printers
53 #cfg_dir=/usr/local/nagios/etc/switches
54 #cfg_dir=/usr/local/nagios/etc/routers
55
56
57
58
59 # OBJECT CACHE FILE
60 # This option determines where object definitions are cached when
61 # Nagios starts/restarts. The CGIs read object definitions from
62 # this cache file (rather than looking at the object config files
63 # directly) in order to prevent inconsistencies that can occur
64 # when the config files are modified after Nagios starts.
65
```

Save the file.

Create the servers directory in the etc directory.

```
sudo mkdir /usr/local/nagios/etc/servers
```

There is a commands.cfg file. This file stores the commands that nagios server will execute. We have installed the NRPE plugin. We need to define a command for this plugin. Thus nagios server can execute this plugin to monitor the clients.

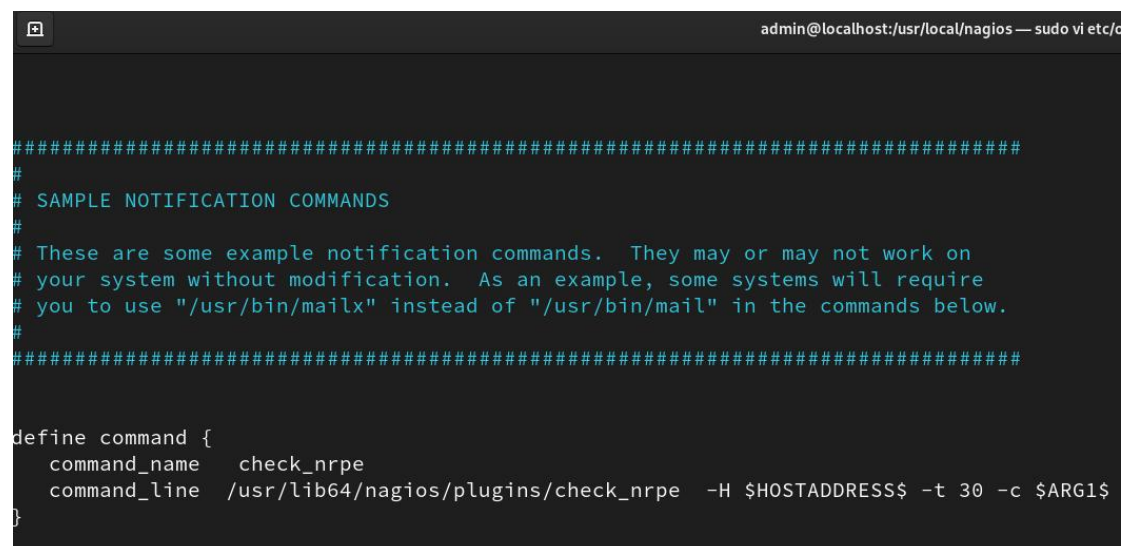
Edit the commands file and add the definition for the NRPE command.

```
sudo vi /usr/local/nagios/etc/objects/commands.cfg
```

In the file add following lines anywhere.

```
define command {  
    command_name    check_nrpe  
    command_line     /usr/lib64/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -t 30 -c $ARG1$  
}
```

This is shown in the below image.



```
admin@localhost:/usr/local/nagios — sudo vi etc/c  
#####  
#  
# 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  
# you to use "/usr/bin/mailx" instead of "/usr/bin/mail" in the commands below.  
#  
#####  
  
define command {  
    command_name    check_nrpe  
    command_line     /usr/lib64/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -t 30 -c $ARG1$  
}
```

Save the file.

Whenever any configuration file related to nagios server is modified or newly created, it is recommended that you verify it for any syntax errors. This will prevent nagios service from failing to restart.

To verify nagios for errors use following command

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

The output of the command should display 0 warnings and 0 errors as shown in the below image.

```
[admin@localhost nagios]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.2
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-04-30
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 25 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

This command will just check for syntax errors. Any configuration error will not be detected. The nagios service will be started however the function related to that configuration will not work and nagios will fail to display monitoring result.

Open the NRPE port in the firewall.

```
firewall-cmd --permanent --add-port=5666/tcp
```

```
firewall-cmd --reload
```

Now start the nagios service and enable it for auto start using following commands.

```
systemctl start nagios
```

```
systemctl enable nagios
```

Now refresh the Nagios website URL in the browser.

Following page should be displayed. Now the Nagios Service is running and its information will be displayed. The PID will be different on different machines.

# Nagios®

- General
  - Home
  - Documentation
- Current Status
  - Tactical Overview
  - Map (Legacy)
  - Hosts
  - Services
  - Host Groups
    - Summary
    - Grid
  - Service Groups
    - Summary
    - Grid
  - Problems
    - Services (Unhandled)
    - Hosts (Unhandled)
    - Network Outages
  - Quick Search:
- Reports
  - Availability
  - Trends (Legacy)
  - Alerts
    - History
    - Summary
    - Histogram (Legacy)
  - Notifications
  - Event Log
- System
  - Comments
  - Downtime
  - Process Info
  - Performance Info
  - Scheduling Queue
  - Configuration

Daemon running with PID 53644

**Nagios® Core™**  
Version 4.5.2  
April 30, 2024  
[Check for updates](#)

### Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

### Latest News

### Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

### Don't Miss...

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

Click on the Hosts option on the left side of the page. It will display following page. The default configuration of Nagios includes configuration files for the local server.

# 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

### Current Network Status

Last Updated: Fri Aug 2 18:56:56 IST 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.2 - www.nagios.org  
Logged in as nagiosadmin

View Service Status Detail For All Host Groups  
View Status Overview For All Host Groups  
View Status Summary For All Host Groups  
View Status Grid For All Host Groups

### Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems All Types

0	1
---	---

### Service Status Totals

Ok	Warning	Unknown	Critical	Pending
2	0	0	0	6

All Problems All Types

0	8
---	---

### Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	
localhost	UP	08-02-2024 18:55:14	0d 0h 1m 45s+	F

Results 1 - 1 of 1 Matching Hosts

Similarly Click on the Services option below the Hosts option. The following page is displayed.

# Nagios®

- General
  - Home
  - Documentation
- Current Status
  - Tactical Overview
  - Map (Legacy)
  - Hosts
  - Services
  - Host Groups
    - Summary
    - Grid
  - Service Groups
    - Summary
    - Grid
  - Problems
    - Services (Unhandled)
    - Hosts (Unhandled)
    - Network Outages
  - Quick Search:
- Reports
  - Availability
  - Trends (Legacy)
  - Alerts

### Current Network Status

Last Updated: Fri Aug 2 20:32:28 IST 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.2 - www.nagios.org  
Logged in as nagiosadmin

View History For all hosts  
View Notifications For All Hosts  
View Host 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
7	1	0	0	0

All Problems All Types

1	8
---	---

### Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	08-02-2024 19:05:48	0d 1h 36m 40s	1/4	OK - load average: 0.05, 0.14, 0.25
	Current Users	OK	08-02-2024 19:06:26	0d 1h 36m 2s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	08-02-2024 19:05:05	0d 1h 32m 24s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 2714165 bytes in 0.570 second response time
	PING	OK	08-02-2024 19:02:41	0d 1h 34m 47s	1/4	PING OK - Packet loss = 0%, RTA = 0.20 ms
	Root Partition	OK	08-02-2024 19:03:18	0d 1h 34m 10s	1/4	DISK OK - free space: / 234219 MB (96.96% inode=100%)
	SSH	OK	08-02-2024 19:03:56	0d 1h 33m 32s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	OK	08-02-2024 19:04:33	0d 1h 32m 55s	1/4	SWAP OK - 100% free (9215 MB out of 9215 MB)
	Total Processes	OK	08-02-2024 19:05:11	0d 1h 32m 17s	1/4	PROCS OK: 77 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services

The Nagios server by default monitors the local server for these services. Click on any service name like Current Load and it will display the details about that service.

### Add a Linux server for monitoring to Nagios server.

This will be the Linux server that will be monitored by the Nagios. For this create another virtual machine. This virtual machine configuration can be 1/2 GB RAM, 1 CPU and 20GB hard disk. The configuration depends on the actual hardware configuration of your laptop.

Install any Linux OS this virtual machine. Here Centos 9 is installed.

After Installation of OS make sure to configure proper timezone. Also make sure the IP address is in the same network as the Nagios server. Check using ping. Update and upgrade the OS.

Install NRPE agent on this client.

The NRPE agent is installed from the source code.

1. `yum install epel-release -y` # Fedora EPEL repository installation (optional)
2. `cd /tmp`
3. `wget https://assets.nagios.com/downloads/nagiosxi/agents/linux-nrpe-agent.tar.gz`
4. `tar xzf linux-nrpe-agent.tar.gz`
5. `sudo -i` # The fullinstall script requires root privileges
6. `cd /tmp/linux-nrpe-agent`
7. `./fullinstall` # NRPE agent install script. (dot before / in the command)

The above script executes and then it will prompt the following. Here enter the Nagios server IP address. This configures NRPE agent to accept communication from the Nagios server.

allowed\_from: type-nagios\_server\_ip

Once the script execution is complete. Check if NRPE agent is running using following command.

`systemctl status nrpe`

```
admin@localhost:~$ sudo systemctl status nrpe
● nrpe.service - Nagios Remote Plugin Executor
   Loaded: loaded (/usr/lib/systemd/system/nrpe.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-08-03 02:48:32 EDT; 21min ago
     Docs: http://www.nagios.org/documentation
    Main PID: 915 (nrpe)
      Tasks: 1 (limit: 22829)
     Memory: 1.4M
        CPU: 10ms
    CGroup: /system.slice/nrpe.service
            └─915 /usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -f

Aug 03 02:48:32 localhost.localdomain systemd[1]: Started Nagios Remote Plugin Executor.
Aug 03 02:48:32 localhost.localdomain nrpe[915]: Starting up daemon
Aug 03 02:48:32 localhost.localdomain nrpe[915]: Server listening on 0.0.0.0 port 5666.
Aug 03 02:48:32 localhost.localdomain nrpe[915]: Server listening on :: port 5666.
Aug 03 02:48:32 localhost.localdomain nrpe[915]: Warning: Daemon is configured to accept command arguments
Aug 03 02:48:32 localhost.localdomain nrpe[915]: Listening for connections on port 5666
Aug 03 02:48:32 localhost.localdomain nrpe[915]: Allowing connections from: 127.0.0.1,192.168.198.167
admin@localhost:~$
```

The nrpe service should be running. At the end of the output read the line which displays the message "Allowing connections from:". Make sure the Nagios server IP is displayed and is correct.



The NRPE configuration file is located in the `/usr/local/nagios/etc` directory on the client. The name of the file is `nrpe.cfg`.

In this file on line 79 you will find the `allowed_hosts` parameter. This is where you can provide the Nagios server IP address. Thus in the above step if you have given incorrect Nagios server IP address then you can correct it here. This is shown below.

```
73 # address. I would highly recommend adding a
74 # file to allow only the specified host to co
75 # you are running this daemon on.
76 #
77 # NOTE: This option is ignored if NRPE is run
78
79 allowed_hosts=127.0.0.1,192.168.198.167
80
81
82
```

Scroll down to line number 197 and you will find example commands specified. The `check_users` command will check the number of users active on this client. The `check_load` command will check the processor load. Similarly other command check hard disk space, total processes etc. For most of the commands the `-w` option specifies the warning conditions. The `-c` option specifies the critical conditions.

Here we enable `check_users`, `check_load`, `check_hda1` and `check_total_procs` as shown below. To enable just remove the hash at the beginning of the line. If these commands are already enabled then you don't have to do anything.

```
197 # The following examples use hardcoded command arguments...
198
199 command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
200 command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
201 command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/hda1
202 #command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
203 command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
204
```

Save the file.

Restart the `nrpe` service using `sudo systemctl restart nrpe` command.

The NRPE agent uses TCP port 5666. The installation script automatically opens the port in the firewall.

You can confirm it using command `sudo firewall-cmd --list-all`. The output will be as shown below.

```
[admin@localhost etc]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 5666/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Now the Nagios server needs to be configured so that it starts communicating with the client and starts monitoring it.

##### Perform following tasks on the Nagios Server #####

Logon to Nagios Server.

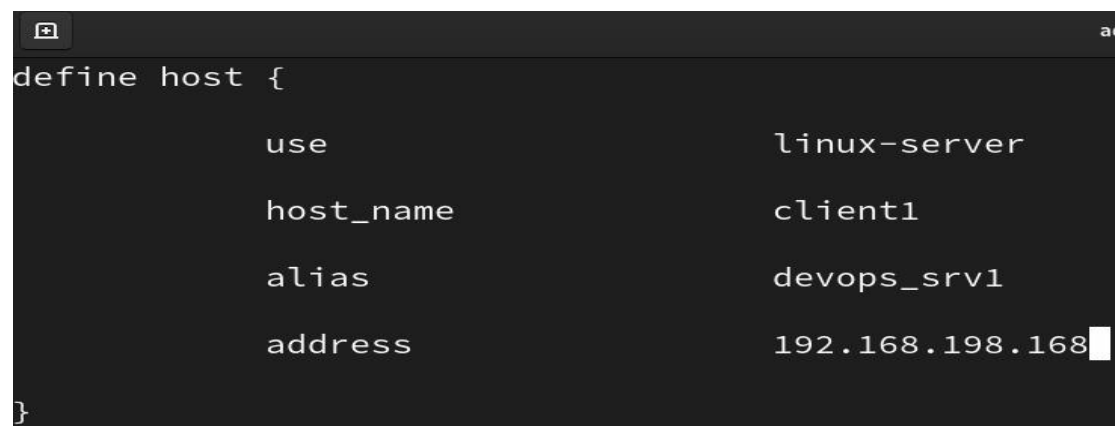
Open terminal. You need to create a file for this client in the servers directory. The name of the file is generally the name of the host. Here the name is given as client1.

```
sudo vi /usr/local/nagios/etc/servers/client1.cfg
```

Type the following in the file.

```
define host {  
  
    use          linux-server  
  
    host_name    centos0.client1  
  
    alias        centos7.client1  
  
    address      client_ip  
  
}
```

Here specify the hostname in the host\_name. In the alias you can specify the other name of the server. In the address specify the IP address of the client. This is shown as below.



```
define host {  
  
    use          linux-server  
  
    host_name    client1  
  
    alias        devops_srv1  
  
    address      192.168.198.168  
  
}
```

Save the file.

Verify the Nagios configuration files for any syntax errors. Use following command.

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

There should be no errors or warnings.

Restart the Nagios service using following command.

```
sudo systemctl restart nagios
```

Now open the browser on the nagios server. Go to the URL <http://localhost/nagios>. Login as nagiosadmin user.

Click Hosts option on left side. The page that is displayed should show the new client added for monitoring as shown below.

The screenshot shows the Nagios web interface. On the left is a sidebar with a 'Hosts' link selected. The main area shows a summary of network status and a table of host details. The table lists 'client1' and 'localhost' as being 'UP'.

Host	Status	Last Check	Duration
client1	UP	08-03-2024 14:45:44	0d 0h 0m 39s+
localhost	UP	08-03-2024 14:42:55	0d 17h 53m 30s

However click the services option below Hosts. On that page you will not be able to see the client that we added above. This is because we have not informed the Nagios server about what we want to monitor on this client.

Thus edit the client1.cfg file and define the services to monitor. We will use the services that we enabled on the client in the nrpe.cfg file.

```
sudo vi /usr/local/nagios/etc/servers/client1.cfg
```

Add following below the define host section in the file.

```
define service {
    use                generic-service
    host_name          client1
    service_description Current Users
    check_command       check_nrpe!check_users
}

define service {
    use                generic-service
    host_name          client1
    service_description Current Load
    check_command       check_nrpe!check_load
}

define service {
    use                generic-service
    host_name          client1
    service_description Check Root partition
    check_command       check_nrpe!check_hda1
}

define service {
    use                generic-service
    host_name          client1
    service_description Total Processes
    check_command       check_nrpe!check_total_procs
}
```

This will look as shown below.



```
define host {
    use                linux-server
    host_name          client1
    alias              devops_srv1
    address            192.168.198.168
}

define service {
    use                generic-service
    host_name          client1
    service_description Current Users
    check_command       check_nrpe!check_users
}

define service {
    use                generic-service
    host_name          client1
    service_description Current Load
    check_command       check_nrpe!check_load
}

define service {
    use                generic-service
    host_name          client1
    service_description Check Root partition
    check_command       check_nrpe!check_hda1
}

define service {
    use                generic-service
    host_name          client1
    service_description Total Processes
    check_command       check_nrpe!check_total_procs
}
```

Save the file.

Check the configuration files for any syntax error using following command.

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

There should be no errors or warnings.

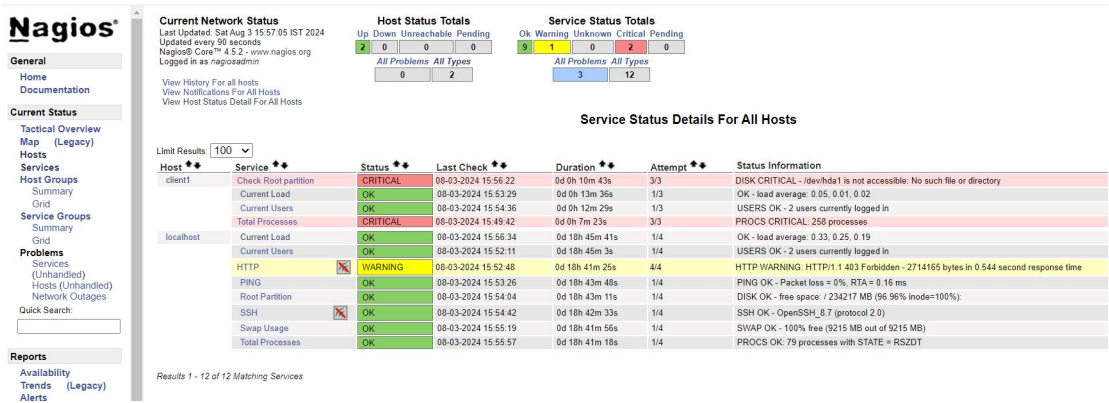
Restart the Nagios service using following command.

```
sudo systemctl restart nagios
```

Now go to the web interface of the Nagios server. Click Services option.

Now the client1 will be displayed. However initially the status of the services will be shown as pending.

Wait for some time. Then click the Services option. It will show the services as shown below.



For any service that is displayed as critical, please read the status information column carefully.

For example

Total Processes	CRITICAL	08-03-2024 15:49:42	0d 0h 8m 54s	3/3	PROCS CRITICAL: 258 processes
-----------------	----------	---------------------	--------------	-----	-------------------------------

In the above image the status information is Process Critical: 258 processes. This means Nagios is able to get the information about the service for monitoring.

Now look at the below image.

Service	Status	Last Check	Duration	Attempt	Status Information
Check Root partition	CRITICAL	08-03-2024 15:56:22	0d 0h 13m 43s	3/3	DISK CRITICAL - /dev/hda1 is not accessible: No such file or directory

In this image the status information states that /dev/hda1 is not accessible. It means the hard disk name is different on the client. Thus you need to go to the client edit the nrpe.cfg file. Modify the check\_hda1 command.

Also if no route host is shown in the status information column then check - A. The nrpe service is running on the client. B. The port TCP 5666 is open in the firewall. C. The IP address of the client is correctly mentioned in the client cfg file on the nagios server.

Now on the client used for this lab the **sudo fdisk -l** command displays the name of the hard disk file as shown below.

```
[admin@localhost etc]$ sudo fdisk -l
Disk /dev/nvme0n1: 500 GiB, 536870912000 bytes, 1048576000 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4d0011b0

Device      Boot      Start          End      Sectors  Size Id Type
/dev/nvme0n1p1 *        2048        2099199      2097152    1G 83 Linux
/dev/nvme0n1p2            2099200      497027071  494927872   236G 83 Linux
/dev/nvme0n1p3      497027072      515901439   18874368    9G 82 Linux swap / Solaris
/dev/nvme0n1p4      515901440      1048575999   532674560   254G 5 Extended
/dev/nvme0n1p5      515903488      1048575999   532672512   254G 83 Linux
```

Thus the name of the hard disk file is /dev/nvme0n1 and not /dev/hda. In your client this name may be different. Thus find out the correct name.



Here we want to monitor the root (/) partition. In this client the /dev/nvme0n1p2 is mounted as root partition. Thus we need to specify this partition in the nrpe.cfg file.  
Now go to the client machine and edit the nrpe.cfg file.

```
sudo vi /usr/local/nagios/etc/nrpe.cfg
```

Scroll down to line number 201 as shown below.

```
199 command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
200 command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
201 command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/hda1
202 #command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
203 command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
```

In that check\_hda1 command line at the end it is /dev/hda1. Change it to your hard disk file name as shown below.

```
199 command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
200 command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
201 command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/nvme0n1p2
202 #command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
203 command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
```

You can change the name of the command from check\_hda1 to check\_root. But then you need to update this in the define service section of the client cfg file on the nagios server.

Save the file. Restart the nrpe service.

Now after some time check in the nagios web interface in the services option. Now the heck root partition service will be working.

Earlier

Service	Status	Last Check	Duration	Attempt	Status Information
Check Root partition	CRITICAL	08-03-2024 15:56:22	0d 0h 13m 43s	3/3	DISK CRITICAL - /dev/hda1 is not accessible: No such file or directory

Now

Host	Service	Status	Last Check	Duration	Attempt	Status Information
client1	Check Root partition	OK	08-03-2024 16:36:46	0d 0h 1m 44s	1/3	DISK OK - free space: /var/tmp 234350 MiB (97.02% inode=100%)

This is how you have successfully added a Linux server for monitoring in the Nagios.

### Running bash script as nrpe command on linux host

There are NRPE plugins to monitor some of the parameters like CPU Load, Users etc. However you may need to monitor some other parameters or services running on clients. In such cases you may write your own scripts or programs and execute them as NRPE commands.

Here we will write a script to monitor httpd service installed on the client. The NRPE agent will check the service. If the service is not installed it should show the status as critical. If the service is installed but it is not running then it will display the status as warning. If the service is running then the status will be displayed as OK.

For this we need to write a shell script on the client which will perform the service checking task. Then we need to define a command in the nrpe.cfg file providing the path of this script.

Finally we need to define a service for this script on the Nagios server in the client cfg file.

### Write shell script

Login to the client. Open terminal. We will store our script in the `/usr/local/nagios/libexec/` directory along with other nagios plugins.

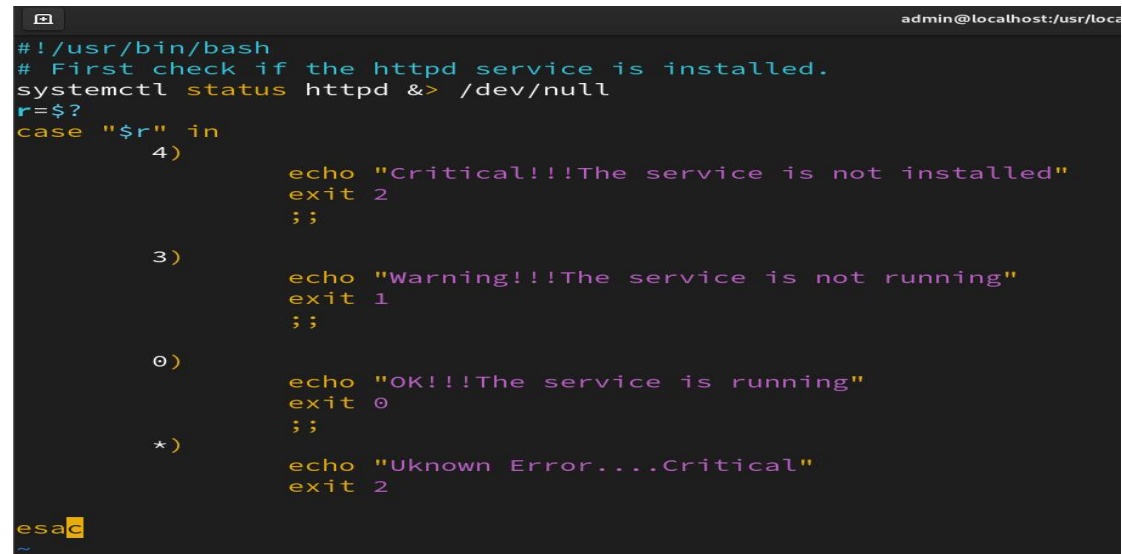
There is already a `check_http` plugin. Thus we will create our script by name `check_web.sh`

```
sudo vi /usr/local/nagios/libexec/check_web.sh
```

Add following script in the following.

```
#!/usr/bin/bash
# First check if the httpd service is installed.
systemctl status httpd &> /dev/null
r=$?
case "$r" in
    4)
        echo "Critical!!!The service is not installed"
        exit 2
        ;;
    3)
        echo "Warning!!!The service is not running"
        exit 1
        ;;
    0)
        echo "OK!!!The service is running"
        exit 0
        ;;
    *)
        echo "Uknown Error....Critical"
        exit 2
esac
```

It will be as shown below.

A screenshot of a terminal window with a dark background. The terminal title bar shows 'admin@localhost: /usr/local/nagios/libexec/'. The script content is displayed in a color-coded font: blue for comments, green for commands, and yellow for exit codes. The script checks the status of the httpd service using systemctl. It uses a case statement to handle different exit codes from systemctl: 4 for 'not installed', 3 for 'not running', 0 for 'running', and any other code for 'unknown error'. Each case has a corresponding echo message and an exit command. The script ends with 'esac'.

```
admin@localhost: /usr/local/nagios/libexec/
#!/usr/bin/bash
# First check if the httpd service is installed.
systemctl status httpd &> /dev/null
r=$?
case "$r" in
    4)
        echo "Critical!!!The service is not installed"
        exit 2
        ;;
    3)
        echo "Warning!!!The service is not running"
        exit 1
        ;;
    0)
        echo "OK!!!The service is running"
        exit 0
        ;;
    *)
        echo "Uknown Error....Critical"
        exit 2
esac
```

Save the file.

Now we need to provide executable permission to the script.

```
sudo chmod +x /usr/local/nagios/libexec/check_web.sh
```

Next we need to define this script as a NRPE command in the nrpe.cfg file. This will allow NRPE to execute this script as a command.

```
sudo vi /usr/local/nagios/etc/nrpe.cfg
```

Add following line in the file.

```
command[check_web]=/usr/local/nagios/libexec/check_web.sh
```

The file will look as shown below. (Line 204 shows the above command)

```
198
199 command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
200 command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
201 command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/nvme0n1p2
202 #command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
203 command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
204 command[check_web]=/usr/local/nagios/libexec/check_web.sh
205
```

Save the file.

Restart the NRPE service using command

```
sudo systemctl restart nrpe
```

##### Perform following steps on Nagios Server #####

Now we need to configure Nagios Server to execute this command for monitoring.

First we will check using command line whether NRPE is able to execute command remotely.

Login to the Nagios server. Open terminal. Use following command.

```
sudo /usr/lib64/nagios/plugins/check_nrpe -H 192.168.198.168 -c check_web
```

The output will be as shown below.

```
admin@localhost:~
[admin@localhost ~]$ sudo /usr/lib64/nagios/plugins/check_nrpe -H 192.168.198.168 -c check_web
Critical!!!The service is not installed
```

This confirms that the Nagios server is able to communicate to the NRPE agent on the client and execute our script. As the httpd service is not installed on the client the output of the script shows the status as critical.

Now we will define it as a service to monitor in the client configuration.

Edit the client configuration file.

```
sudo vi /usr/local/nagios/etc/servers/client1.cfg
```

Define a service for monitoring httpd service using our script.



```

define service {
    use          generic-service
    host_name    client1
    service_description Check httpd status
    check_command check_nrpe!check_web
}

```

This is shown below.

```

define service {
    use          generic-service
    host_name    client1
    service_description Check httpd status
    check_command check_nrpe!check_web
}

```

Save the file.

Verify the Nagios configuration files for any kind of errors.

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

There should not be any warning or errors.

Restart the nagios service using following command.

```
sudo systemctl restart nagios
```

Now go the web interface of the Nagios server.

Click on Services option.

The following output should be shown. It may take some time to update.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
client1	Check Root partition	OK	08-06-2024 15:07:55	2d 20h 53m 34s	1/3	DISK OK - free space: /var/tmp 234264 MIB (96.98% inode=100%):
	Check httpd status	CRITICAL	08-06-2024 15:08:59	2d 18h 5m 11s	3/3	Critical!!!The service is not installed

Now install the httpd service on the client. And then heck here the output after some time. It should be as shown below.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
client1	Check Root partition	OK	08-06-2024 15:07:55	2d 20h 56m 25s	1/3	DISK OK - free space: /var/tmp 234256 MIB (96.98% inode=100%):
	Check httpd status	WARNING	08-06-2024 15:17:08	0d 0h 0m 20s	3/3	Warning!!!The service is not running

Now start the httpd service on the client and then check the status here. It should be in Green colour as shown below.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
client1	Check Root partition	OK	08-06-2024 15:37:55	2d 21h 18m 39s	1/3	DISK OK - free space: /var/tmp 234256 MIB (96.98% inode=100%):
	Check httpd status	OK	08-06-2024 15:39:24	0d 0h 0m 18s	1/3	OK!!!The service is running

This is how you can run any script or program through Nagios for monitoring a parameter of a client.

## Adding Windows server for monitoring to Nagios Server

This will be the Windows server that will be monitored by the Nagios. For this create another virtual machine. This virtual machine configuration can be 2 GB RAM, 1 CPU and 60GB hard disk. The configuration depends on the actual hardware configuration of your laptop. Make sure this machine is in the same network as the Nagios server.

Install any Windows OS this virtual machine. Here Windows Server 2016 is installed.

After the Windows server Installation, set the correct timezone, date and time, Computer name and IP address. Restart the server after changing the name of the server.

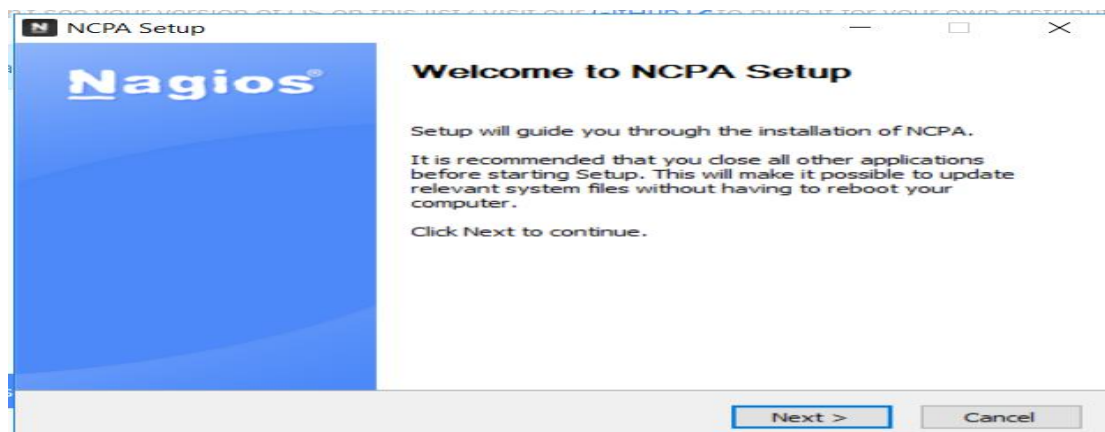
Install NCPA on Windows Server.

1. Go to the following URL

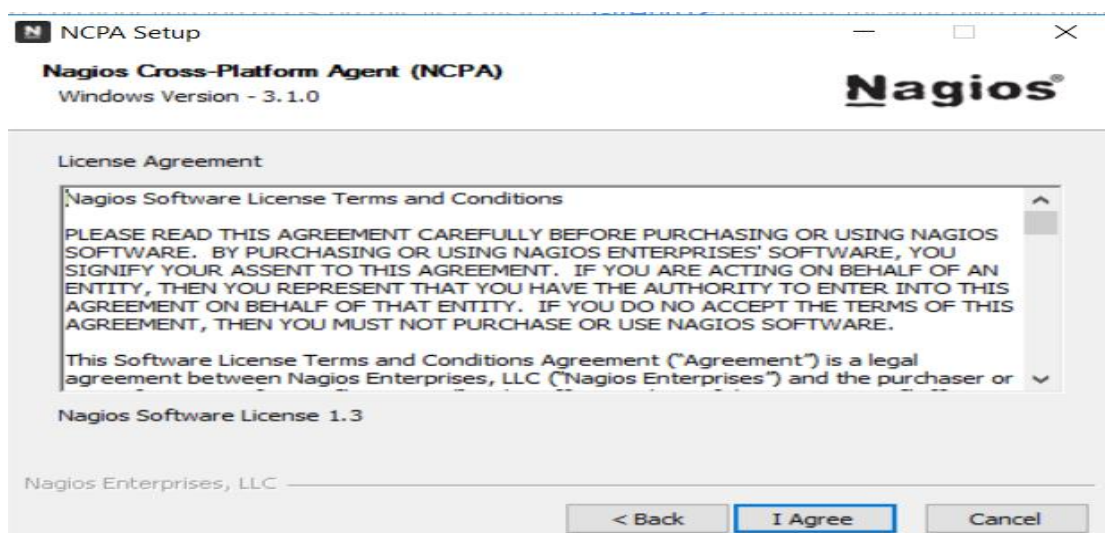
<https://www.nagios.org/ncpa/#downloads>

Click EXE installer for Windows. Save the installer file. Once the download completes, double click the file to start the installation.

Following window opens.

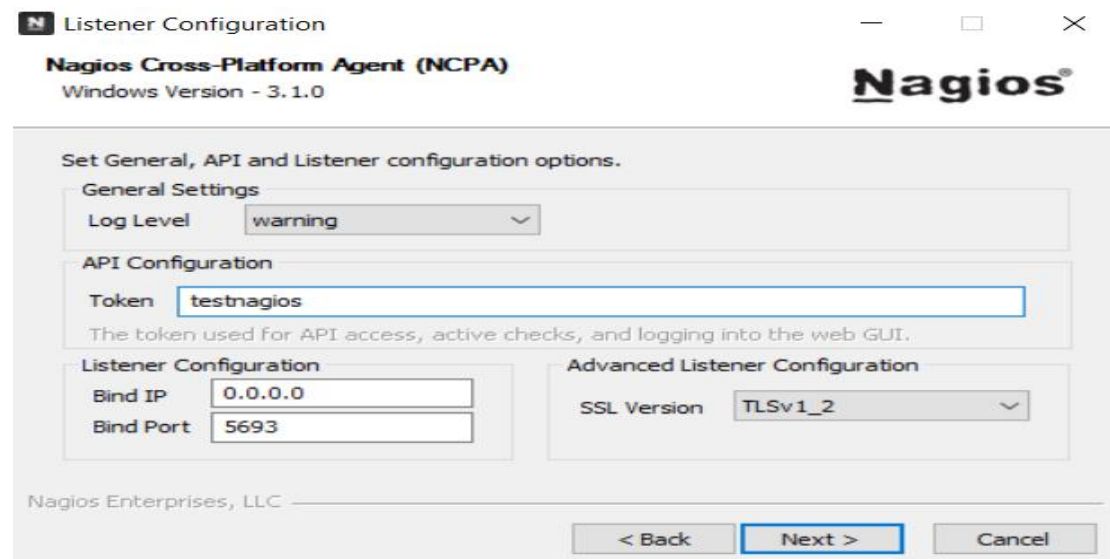


Click Next.



Click I Agree to accept the license agreement.

On the following screen specify Token in API Configuration. This token is any string that will work as a password between Nagios Server and this Windows server. Here testnagios string is given as token. You need to provide this token while you configure Nagios server.



**Listener Configuration**

**Nagios Cross-Platform Agent (NCPA)**  
Windows Version - 3.1.0

Set General, API and Listener configuration options.

**General Settings**

Log Level:

**API Configuration**

Token:   
The token used for API access, active checks, and logging into the web GUI.

**Listener Configuration**

Bind IP:   
Bind Port:

**Advanced Listener Configuration**

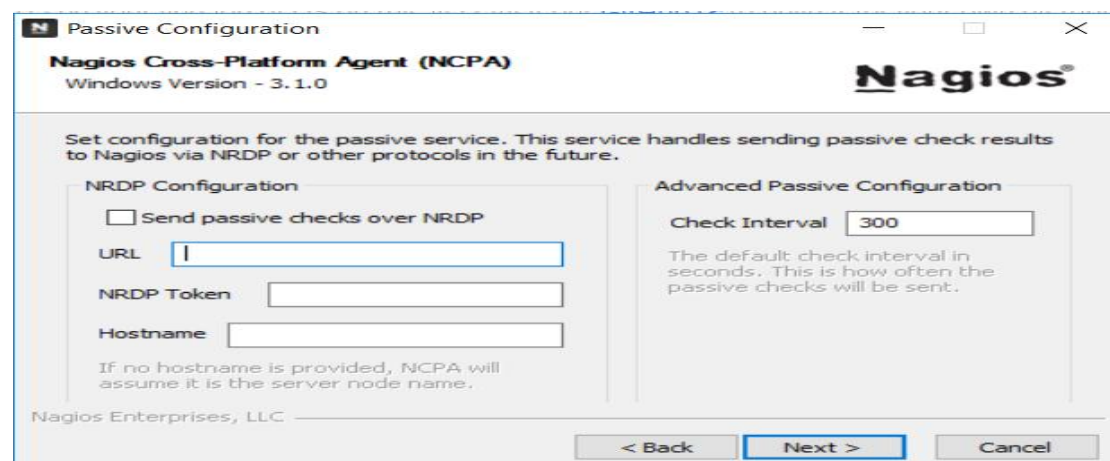
SSL Version:

Nagios Enterprises, LLC

< Back   **Next >**   Cancel

Keep all other options as default. Click Next.

The following screen is displayed.



**Passive Configuration**

**Nagios Cross-Platform Agent (NCPA)**  
Windows Version - 3.1.0

Set configuration for the passive service. This service handles sending passive check results to Nagios via NRDP or other protocols in the future.

**NRDP Configuration**

☐ Send passive checks over NRDP

URL:

NRDP Token:

Hostname:   
If no hostname is provided, NCPA will assume it is the server node name.

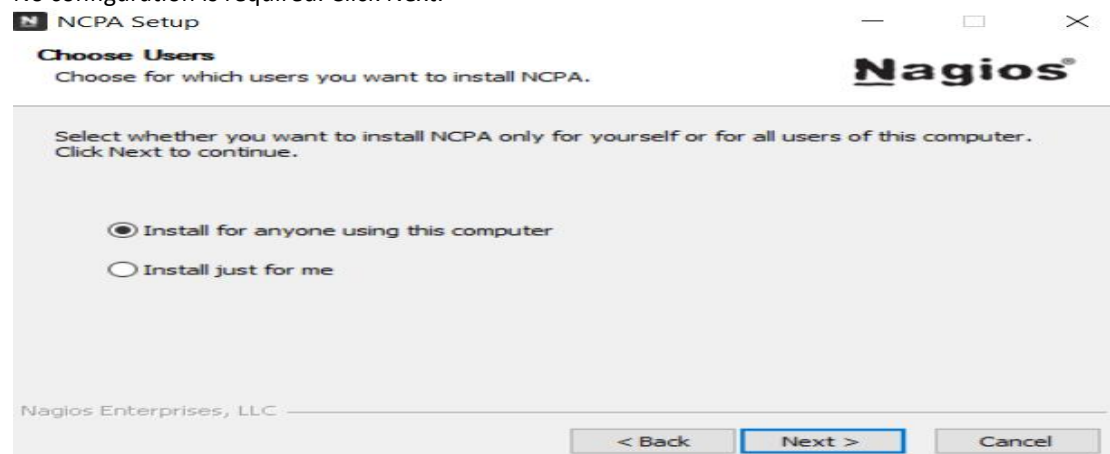
**Advanced Passive Configuration**

Check Interval:   
The default check interval in seconds. This is how often the passive checks will be sent.

Nagios Enterprises, LLC

< Back   **Next >**   Cancel

No configuration is required. Click Next.



**NCPA Setup**

**Choose Users**  
Choose for which users you want to install NCPA.

Select whether you want to install NCPA only for yourself or for all users of this computer. Click Next to continue.

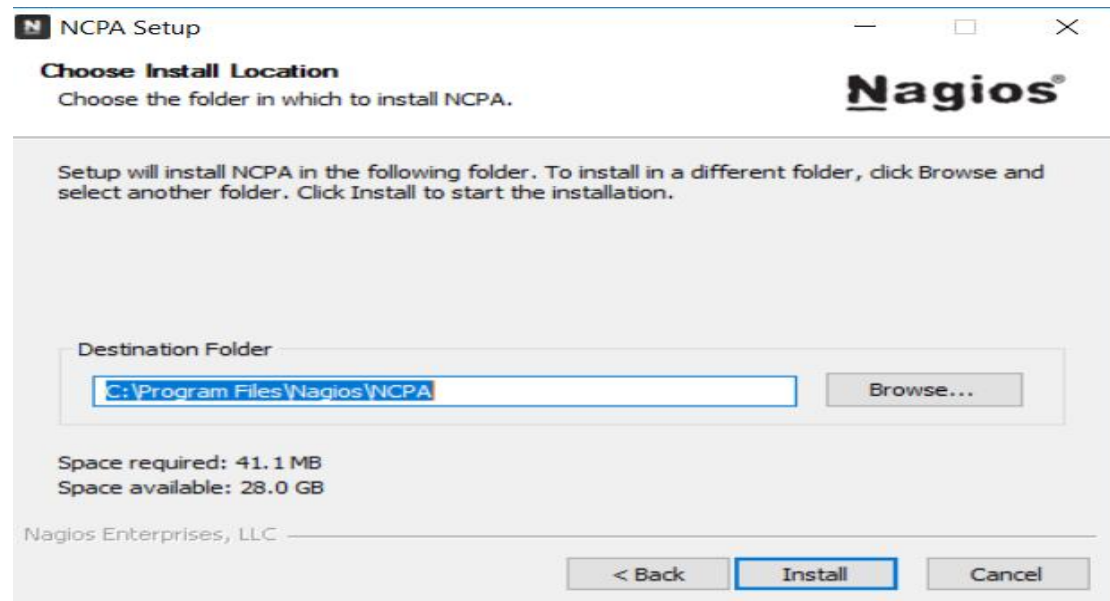
☒ Install for anyone using this computer

☐ Install just for me

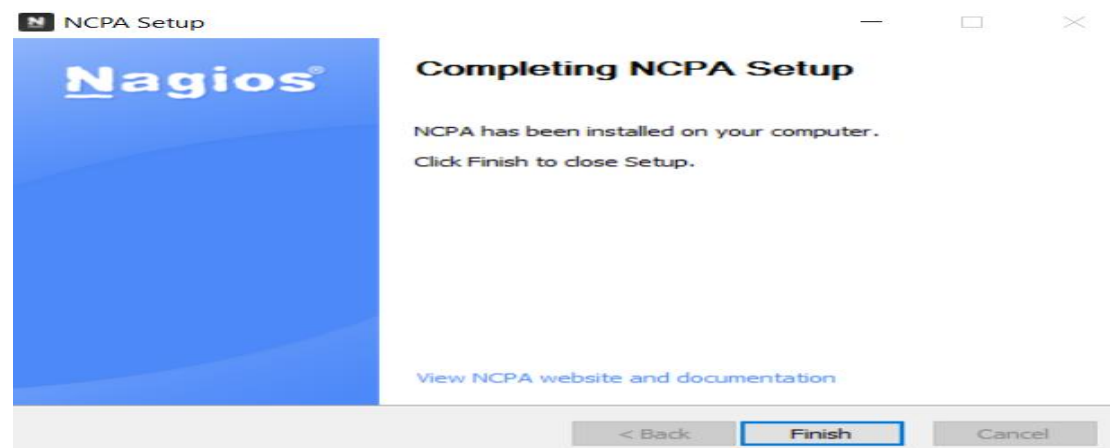
Nagios Enterprises, LLC

< Back   **Next >**   Cancel

Select the appropriate option as per your requirement. Click Next.

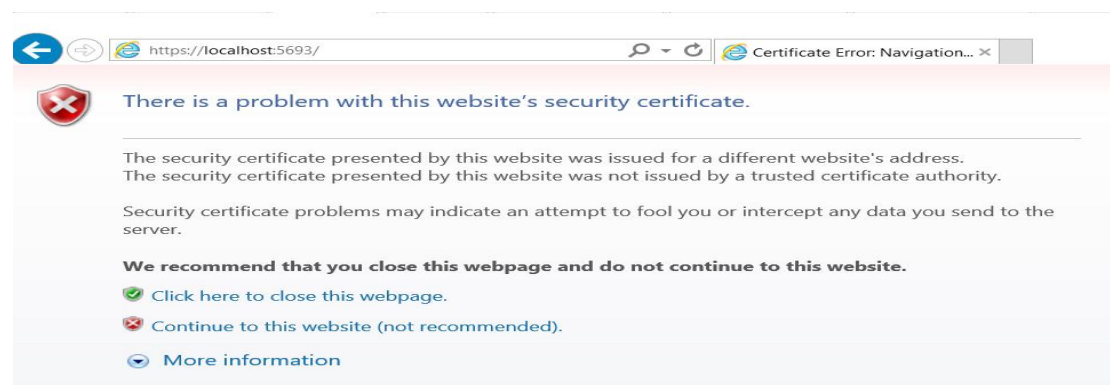


Keep the default installation directory as shown above. Click Install.

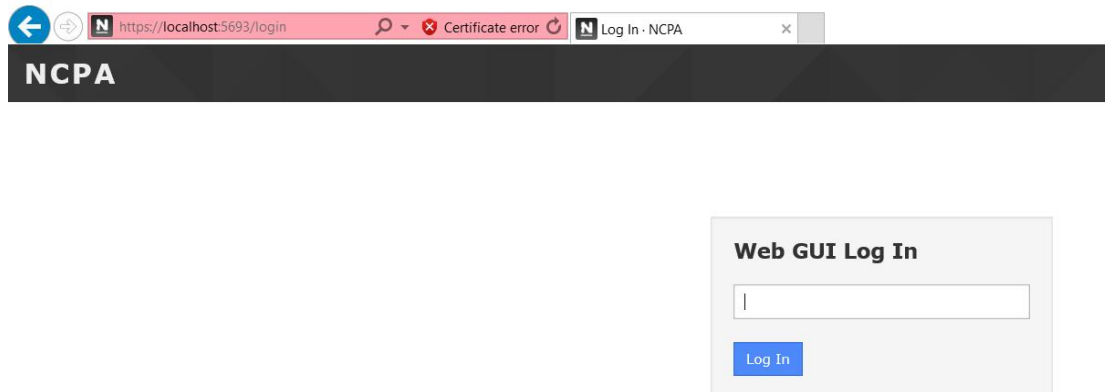


Once the installation is complete, above screen is displayed. Click Finish.

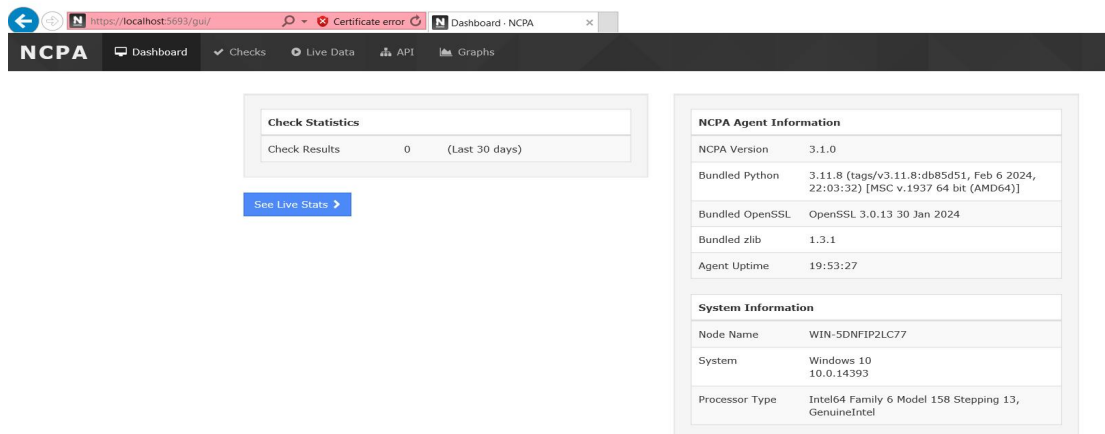
This how the NCPA agent is installed on the Windows server. Now verify the installation. Open browser on the Windows server. Go to the URL <https://localhost:5693/> . The NCPA uses the port number 5693. The following web page may open.



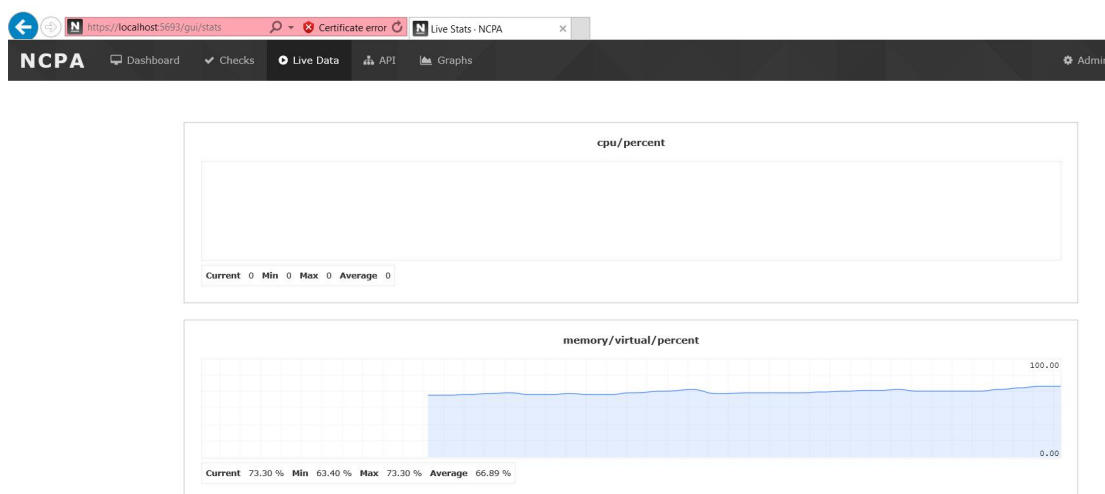
Click Continue to the website option. The following web page will be displayed.



Here type the token specified during NCPA installation. In this guide the token given is **testnagios**. Click Log In button. Following web page should be displayed.



Click Live Stats button. The following page should open.



This verifies that NCPA is installed and working properly. Now add this server to Nagios for monitoring.

## Configure Nagios Server to Monitor the Windows Server

Perform the following tasks on the Nagios server.

Log in to the Nagios server. Open terminal and execute following commands.

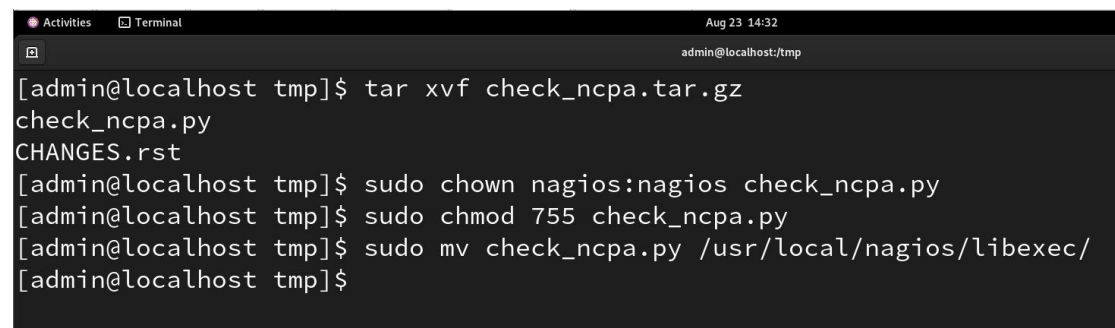
```
cd /tmp
Wget https://assets.nagios.com/downloads/ncpa/check\_ncpa.tar.gz
```

A terminal window titled 'Terminal' with the date 'Aug 23 14:29' in the top right corner. The prompt is 'admin@localhost:~'. The user enters 'cd /tmp'. The prompt changes to 'admin@localhost tmp'. The user enters 'wget https://assets.nagios.com/downloads/ncpa/check\_ncpa.tar.gz'. The terminal shows the progress of the download, including the URL, the file size (5047 bytes), and the save location. The download completes successfully.

```
[admin@localhost ~]$ cd /tmp
[admin@localhost tmp]$ wget https://assets.nagios.com/downloads/ncpa/check_ncpa.tar.gz
--2024-08-23 14:29:23-- https://assets.nagios.com/downloads/ncpa/check_ncpa.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5047 (4.9K) [application/x-gzip]
Saving to: 'check_ncpa.tar.gz'

check_ncpa.tar.gz      100%[=====] 4.93K --.-KB/s   in 0s
2024-08-23 14:29:25 (75.1 MB/s) - 'check_ncpa.tar.gz' saved [5047/5047]
```

```
tar xvf check_ncpa.tar.gz
chown nagios:nagios check_ncpa.py
chmod 775 check_ncpa.py
mv check_ncpa.py /usr/local/nagios/libexec
```

A terminal window titled 'Terminal' with the date 'Aug 23 14:32' in the top right corner. The prompt is 'admin@localhost:tmp'. The user enters 'tar xvf check\_ncpa.tar.gz'. The terminal shows the extraction of 'check\_ncpa.py' and the creation of a 'CHANGES.rst' file. The user then enters 'sudo chown nagios:nagios check\_ncpa.py', 'sudo chmod 755 check\_ncpa.py', and 'sudo mv check\_ncpa.py /usr/local/nagios/libexec/'. The terminal shows the successful execution of these commands.

```
[admin@localhost tmp]$ tar xvf check_ncpa.tar.gz
check_ncpa.py
CHANGES.rst
[admin@localhost tmp]$ sudo chown nagios:nagios check_ncpa.py
[admin@localhost tmp]$ sudo chmod 755 check_ncpa.py
[admin@localhost tmp]$ sudo mv check_ncpa.py /usr/local/nagios/libexec/
[admin@localhost tmp]$
```

There is a commands.cfg file. This file stores the commands that Nagios server will execute. We have installed the NCPA plugin. We need to define a command for this plugin. Thus Nagios server can execute this plugin to monitor the clients.

Edit the commands file and add the definition for the NRPE command.

```
sudo vi /usr/local/nagios/etc/objects/commands.cfg
```

In the file add following lines anywhere.

```
define command {
    command_name check_ncpa
    command_line $USER1$/check_ncpa.py -H $HOSTADDRESS$ $ARG1$
}
```

*This is as shown below.*

```
define command {
    command_name    check_ncpa
    command_line    $USER1$/check_ncpa.py -H $HOSTADDRESS$ $ARG1$
}
```

Save the file.

Now create a file for this Windows server in the servers directory. The name of the file is generally the name of the host. Here the name is given as win-srv1.

```
sudo vi /usr/local/nagios/etc/servers/win-srv1.cfg
```

We will define the host and the services to monitor on this host in this file. Type the following in the file.

```
define host {
    host_name        Win-srv1
    address          192.168.198.171
    check_command     check_ncpa!-t 'testnagios' -P 5693 -M system/agent_version
    max_check_attempts 5
    check_interval    5
    retry_interval     1
    check_period      24x7
    contacts          nagiosadmin
    notification_interval 60
    notification_period 24x7
    notifications_enabled 1
    icon_image        ncpa.png
    statusmap_image    ncpa.png
    register          1
}

define service {
    host_name        Win-srv1
    service_description CPU Usage
    check_command     check_ncpa!-t 'testnagios' -P 5693 -M cpu/percent -w 20 -c 40 -q
    'aggregate=avg'
    max_check_attempts 5
    check_interval    5
    retry_interval     1
    check_period      24x7
    notification_interval 60
    notification_period 24x7
    contacts          nagiosadmin
    register          1
}

define service {
    host_name        Win-srv1
    service_description Memory Usage
    check_command     check_ncpa!-t 'testnagios' -P 5693 -M memory/virtual -w 50 -c 80 -u G
    max_check_attempts 5
    check_interval    5
    retry_interval     1
}
```



```

check_period      24x7
notification_interval 60
notification_period 24x7
contacts          nagiosadmin
register          1
}

define service {
    host_name      Win-srv1
    service_description Process Count
    check_command   check_ncpa!-t 'testnagios' -P 5693 -M processes -w 150 -c 200
    max_check_attempts 5
    check_interval  5
    retry_interval   1
    check_period     24x7
    notification_interval 60
    notification_period 24x7
    contacts         nagiosadmin
    register         1
}

```

Please make sure you type proper token in the check\_ncpa command as highlighted above. This is the string given as token during installation of NCPA on Windows server.

Save the file.

Verify the Nagios configuration files for any kind of errors.

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

There should not be any warning or errors.

Restart the nagios service using following command.

```
sudo systemctl restart nagios
```

Now go to the web interface of the Nagios server.

The screenshot shows the Nagios web interface at 192.168.198.167. The interface includes a sidebar with navigation links and a main content area with several status panels.

**Current Network Status**  
 Last Updated: Fri Aug 23 15:15:44 IST 2024  
 Updated every 90 seconds  
 Nagios® Core™ 4.5.2 - www.nagios.org  
 Logged in as nagiosadmin

**Host Status Totals**

Up	Down	Unreachable	Pending
3	0	0	0

**Service Status Totals**

Ok	Warning	Unknown	Critical	Pending
11	1	0	1	3

**Host Status Details For All Host Groups**

Host	Status	Last Check	Duration	Status Information
Win-srv1	UP	08-23-2024 15:15:08	0d 0h 0m 38s+	OK - Agent_version was [3.1.0]
client1	UP	08-23-2024 15:15:09	0d 3h 0m 35s	PING OK - Packet loss = 0%, RTA = 1.18 ms
localhost	UP	08-23-2024 15:11:41	19d 18h 21m 19s	PING OK - Packet loss = 0%, RTA = 0.17 ms

Results: 1 - 3 of 3 Matching Hosts

You will see the Windows server in the Hosts.



Click Services button. The Winsrv1 host entry should be present.

The screenshot shows the Nagios web interface for the host 192.168.198.167. The 'Services' button is highlighted in the left sidebar. The main content area displays 'Service Status Details For All Hosts' with a table listing various services and their current status.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Win-srv1	CPU Usage	CRITICAL	08-23-2024 15:16:19	0d 0h 0m 47s	1/5	CRITICAL: Percent was 73.40 %
Win-srv1	Memory Usage	PENDING	N/A	0d 0h 2m 0s+	1/5	Service check scheduled for Fri Aug 23 15:17:33 IST 2024
Win-srv1	Process Count	PENDING	N/A	0d 0h 2m 0s+	1/5	Service check scheduled for Fri Aug 23 15:18:47 IST 2024
client1	Check Root partition	OK	08-23-2024 15:15:12	18d 21h 38m 24s	1/3	DISK OK - free space: /var/tmp 234061 MB (96.90% inode=100%)
client1	Check httpd status	OK	08-23-2024 15:12:16	0d 2h 56m 50s	1/3	OK!The service is running
client1	Current Load	OK	08-23-2024 15:14:09	0d 1h 4m 57s	1/3	OK - load average: 0.01, 0.20, 0.30
client1	Current Users	OK	08-23-2024 15:14:09	0d 1h 4m 57s	1/3	USERS OK - 2 users currently logged in
client1	Total Processes	CRITICAL	08-23-2024 15:19:07	18d 23h 43m 49s	3/3	PROCS CRITICAL: 264 processes
localhost	Current Load	OK	08-23-2024 15:16:45	19d 18h 22m 7s	1/4	OK - load average: 0.06, 0.18, 0.27
localhost	Current Users	OK	08-23-2024 15:10:06	19d 18h 21m 29s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	WARNING	08-23-2024 15:15:44	19d 18h 17m 51s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 2714165 bytes in 0.628 second response time
localhost	PING	OK	08-23-2024 15:16:49	19d 18h 20m 14s	1/4	PING OK - Packet loss = 0%, RTT = 0.09 ms
localhost	Root Partition	OK	08-23-2024 15:10:06	19d 18h 19m 37s	1/4	DISK OK - free space: / 234095 MB (96.91% inode=100%)
localhost	SSH	OK	08-23-2024 15:15:41	19d 18h 18m 59s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
localhost	Swap Usage	OK	08-23-2024 15:15:28	19d 18h 18m 22s	1/4	SWAP OK - 100% free (9215 MB out of 9215 MB)
localhost	Total Processes	OK	08-23-2024 15:16:44	19d 18h 17m 44s	1/4	PROCS OK: 79 processes with STATE = RSZDT

Some of the services may display status as pending. Wait for some time and you should see the updates about these monitored services.

The screenshot shows the Nagios web interface after a refresh. The 'Services' button is still highlighted. The 'Service Status Details For All Hosts' table now shows updated statuses for several services.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Win-srv1	CPU Usage	CRITICAL	08-23-2024 15:20:22	0d 0h 4m 20s	5/5	CRITICAL: Percent was 100.00 %
Win-srv1	Memory Usage	WARNING	08-23-2024 15:24:34	0d 0h 2m 9s	3/5	WARNING: Memory usage was 71.20 % (Available: 1.24 GB, Total: 4.29 GB, Free: 1.24 GB, Used: 3.06 GB)
Win-srv1	Process Count	OK	08-23-2024 15:23:47	0d 0h 5m 55s	1/5	OK: Process count was 57
client1	Check Root partition	OK	08-23-2024 15:15:12	18d 21h 46m 0s	1/3	DISK OK - free space: /var/tmp 234061 MB (96.90% inode=100%)
client1	Check httpd status	OK	08-23-2024 15:22:16	0d 3h 4m 26s	1/3	OK!The service is running
client1	Current Load	OK	08-23-2024 15:24:09	0d 1h 12m 33s	1/3	OK - load average: 1.28, 0.52, 0.33
client1	Current Users	OK	08-23-2024 15:24:09	0d 1h 12m 33s	1/3	USERS OK - 2 users currently logged in
client1	Total Processes	CRITICAL	08-23-2024 15:20:07	18d 23h 51m 25s	3/3	PROCS CRITICAL: 262 processes
localhost	Current Load	OK	08-23-2024 15:21:45	19d 18h 29m 43s	1/4	OK - load average: 0.05, 0.09, 0.20
localhost	Current Users	OK	08-23-2024 15:20:01	19d 18h 29m 5s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	WARNING	08-23-2024 15:20:44	19d 18h 25m 27s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 2714165 bytes in 0.751 second response time
localhost	PING	OK	08-23-2024 15:21:49	19d 18h 27m 50s	1/4	PING OK - Packet loss = 0%, RTT = 0.11 ms
localhost	Root Partition	OK	08-23-2024 15:24:11	19d 18h 27m 13s	1/4	DISK OK - free space: / 234097 MB (96.91% inode=100%)
localhost	SSH	OK	08-23-2024 15:20:41	19d 18h 26m 35s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
localhost	Swap Usage	OK	08-23-2024 15:20:28	19d 18h 25m 58s	1/4	SWAP OK - 100% free (9215 MB out of 9215 MB)
localhost	Total Processes	OK	08-23-2024 15:21:44	19d 18h 25m 28s	1/4	PROCS OK: 86 processes with STATE = RSZDT

Configuring Nagios server to send Email alerts using GMail.

Create a separate GMail account for this lab. Login to this new GMail account. Select the option



On the page that opens, click security option and then click 2 Step verification option to enable it.

The screenshot shows the 'How you sign in to Google' page. The '2-Step Verification' option is selected, and the status is 'On since Aug 29'.

Then log out and login again. Again go to Manage your Google Account option. Search for App password option. Go to the App password page. Create a password. Copy the password displayed. You need to provide it in the Nagios configuration file.

Cyrus SASL (Simple Authentication and Security Layer) is a framework that provides authentication and encryption services in various applications. In the context of email, Cyrus SASL can be used to enable secure authentication for SMTP (Simple Mail Transfer Protocol) servers.

```
sudo yum install postfix -y
```

```
sudo yum install cyrus-sasl cyrus-sasl-lib cyrus-sasl-plain
```

```
sudo mkdir -p /etc/postfix/sasl/
```

```
sudo vi /etc/postfix/sasl/sasl_passwd
```

Type following line in the file.

```
[smtp.gmail.com]:587 username@gmail.com:applicationpassword
```

Save the file.

```
sudo postmap /etc/postfix/sasl/sasl_passwd
```

The above command creates a database file.

```
sudo chown root:root /etc/postfix/sasl/sasl_passwd /etc/postfix/sasl/sasl_passwd.db
```

```
sudo chown 0600 /etc/postfix/sasl/sasl_passwd /etc/postfix/sasl/sasl_passwd.db
```

following commands will modify the postfix configuration file /etc/postfix/main.cf to add the required configuration.

```
sudo postconf -e relayhost=[smtp.gmail.com]:587
```

```
sudo postconf -e smtp_sasl_auth_enable=yes
```

```
sudo postconf -e smtp_sasl_security_options=noanonymous
```

```
sudo Postconf -e smtp_sasl_password_maps=hash:/etc/postfix/sasl/sasl_passwd
```

```
sudo Postconf -e smtp_tls_security_level=encrypt
```

```
sudo postconf -e smtp_tls_security_level=verify
```

```
sudo Postconf -e smtp_tls_CAfile=/etc/ssl/certs/ca-bundle.crt
```

Then edit the following file.

```
sudo vi /etc/aliases
```

Add the following line.

```
root: username@gmail.com
```

Save the file.

```
sudo systemctl enable postfix
```

```
sudo systemctl restart postfix
```

```
sudo dnf install s-nail -y
```

```
echo "Test" | /usr/bin/s-nail -vvv -s "Test Subject" username@gmail.com
```

After this if everything is properly configured, you should receive an email on your gmail account.

```
sudo vim /usr/local/nagios/etc/objects/commands.cfg
```

The following section may be already present. Comment the existing section and put the following section in this file.

```
define command {  
  
    command_name    notify-host-by-email  
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:  
$NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo:  
$HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | mailx -vvv -s "*** $NOTIFICATIONTYPE$ Host  
Alert: $HOSTNAME$ is $HOSTSTATE$ ***" $CONTACTEMAIL$  
}
```

```
define command {  
  
    command_name    notify-service-by-email  
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:  
$NOTIFICATIONTYPE$\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress:  
$HOSTADDRESS$\nState: $SERVICESTATE$\nDate/Time: $LONGDATETIME$\n\nAdditional  
Info:\n\n$SERVICEOUTPUT$\n" | mailx -vvv -s "*** $NOTIFICATIONTYPE$ Service Alert:  
$HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ ***" $CONTACTEMAIL$  
}
```

Save file.

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

Existing section will look as shown below.

```
define contact {  
  
    contact_name    nagiosadmin        ; Short name of user  
    use              generic-contact    ; Inherit default values from generic-contact template (defined  
above)  
    alias            Nagios Admin       ; Full name of user  
    email            username1@gmail.com ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****  
}
```

Add following ontact as shown below.

```
define contact{  
    contact_name    webadmin  
    use              generic-contact  
    alias            Web Admin  
    email            username1@orelit.com  
    service_notification_commands    notify-service-by-email  
    host_notification_commands    notify-host-by-email  
    service_notification_period    24x7  
    host_notification_period    24x7  
    service_notification_options    w,u,c,r,f  
    host_notification_options    d,u,r,f  
}
```

The contactgroup section is also present. Add the names of the contacts in the members.

```
define contactgroup {
    contactgroup_name    admins
    alias                 Nagios Administrators
    members               webadmin,nagiosadmin
}
```

Save the file

Now configure the host definition file to send emails.

```
sudo vi /usr/local/nagios/etc/servers/client1.cfg
```

Add following

```
define host {

    use                linux-server

    host_name          client1

    alias              devops_srv1

    address            192.168.198.168

    contact_groups      admins

}

define service {
    use                generic-service
    host_name          client1
    service_description    Check httpd status
    check_command       check_nrpe!check_web
    contacts             webadmin
}
```

Save file.

Check the nagios configuration for any errors using following command.

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

If there are no warnings or errors, then restart the nagios service.

```
sudo systemctl restart nagios
```

Now stop the httpd service on the Linux client. Wait for some time and check if you receive an email. Start the httpd service and again wait for some time and then check if an email is received.