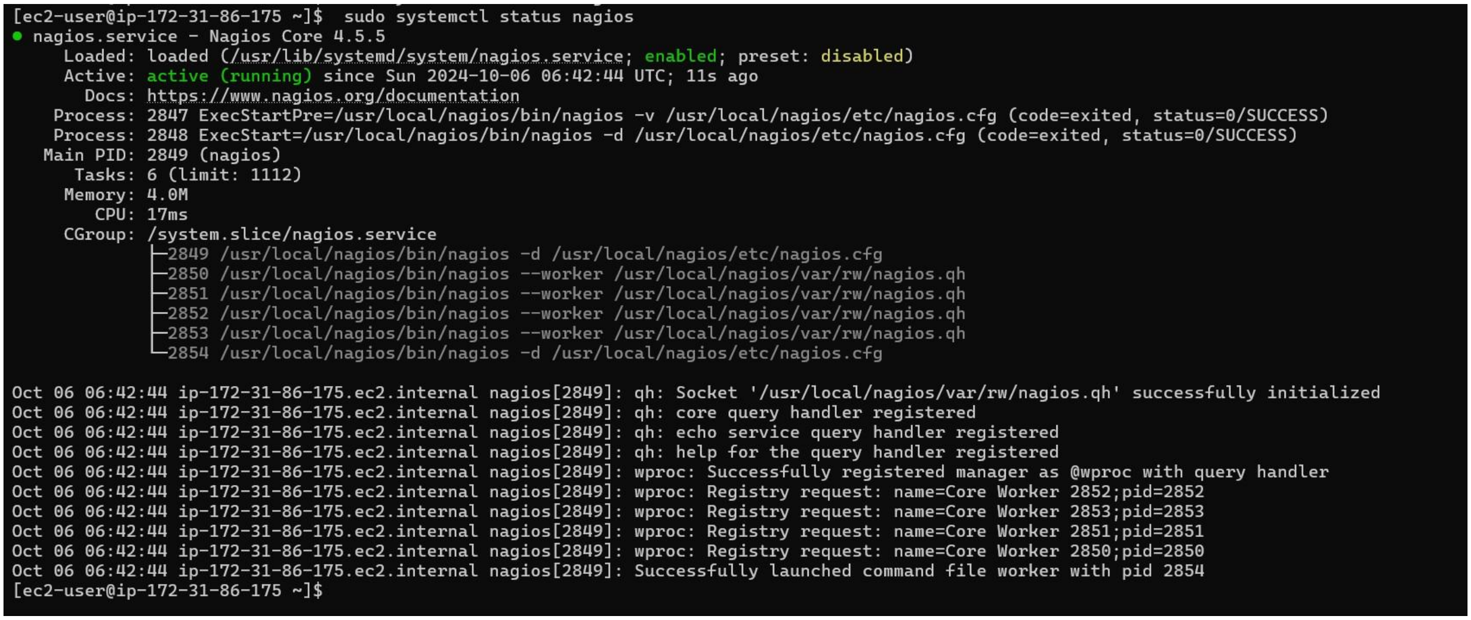
Aditya A Kirtane D15C 26

Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Step 1: Confirming Nagios on the Server

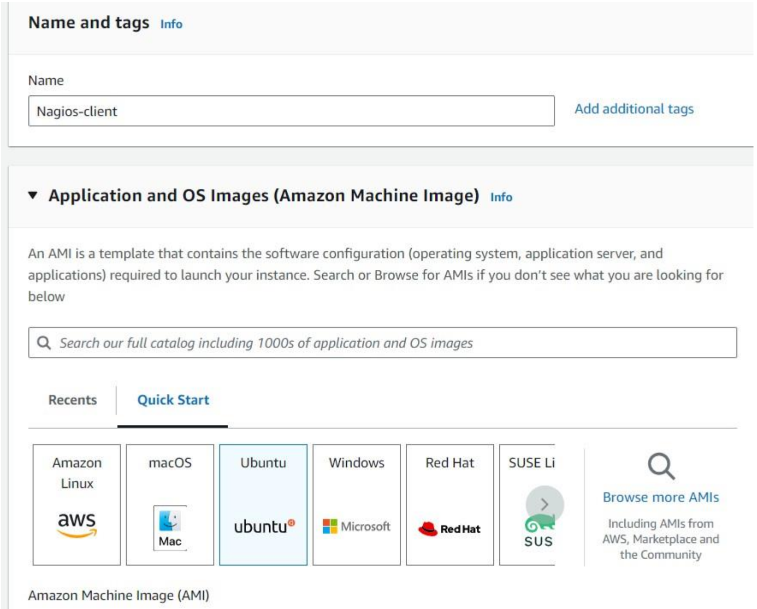
First, ensure that Nagios is running on your server by executing the following command on your Amazon Linux machine (Nagios-host):
`sudo systemctl status nagios`.



```
[ec2-user@ip-172-31-86-175 ~]$  sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-10-06 06:42:44 UTC; 11s ago
       Docs: https://www.nagios.org/documentation
    Process: 2847 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 2848 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 2849 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.0M
        CPU: 17ms
     CGroup: /system.slice/nagios.service
             ├─2849 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─2850 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2851 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2852 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2853 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─2854 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: core query handler registered
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: echo service query handler registered
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: qh: help for the query handler registered
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Successfully registered manager as @wproc with query handler
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2852;pid=2852
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2853;pid=2853
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2851;pid=2851
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: wproc: Registry request: name=Core Worker 2850;pid=2850
Oct 06 06:42:44 ip-172-31-86-175.ec2.internal nagios[2849]: Successfully launched command file worker with pid 2854
[ec2-user@ip-172-31-86-175 ~]$
```

Step 2: Creating EC2 Instance

Next, create a new EC2 instance named **Nagios-client** with the Ubuntu AMI and **t2.micro** instance type. Generate an RSA key pair (.pem file), or use an existing one if available. Make sure to select the security group used in your previous Nagios-host setup.



Name and tags Info

Name

Nagios-client

Add additional tags

▼ **Application and OS Images (Amazon Machine Image)** Info

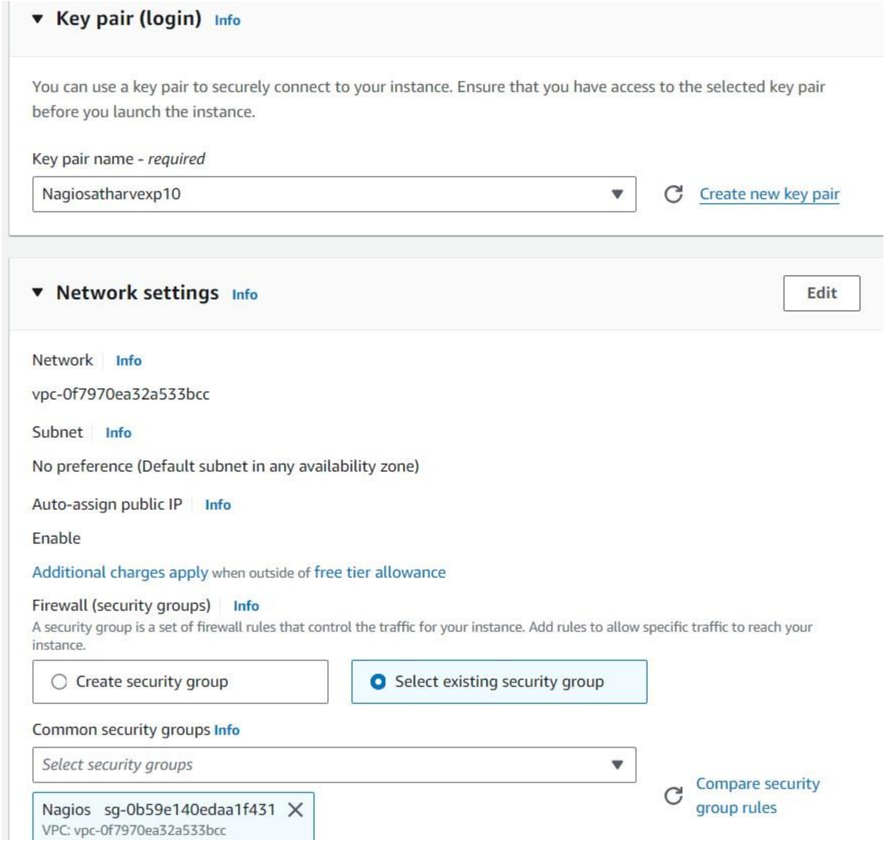
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Step 3: Connecting to the Instance

After creating the EC2 instance, connect to it. Navigate to the folder where the key (.pem) is stored on your local machine. Copy the provided SSH command from the instance's **SSH Client** section and paste it into your terminal.



▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Nagiosatharvexp10 ▼

Create new key pair

▼ **Network settings** Info

Edit

Network Info

vpc-0f7970ea32a533bcc

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

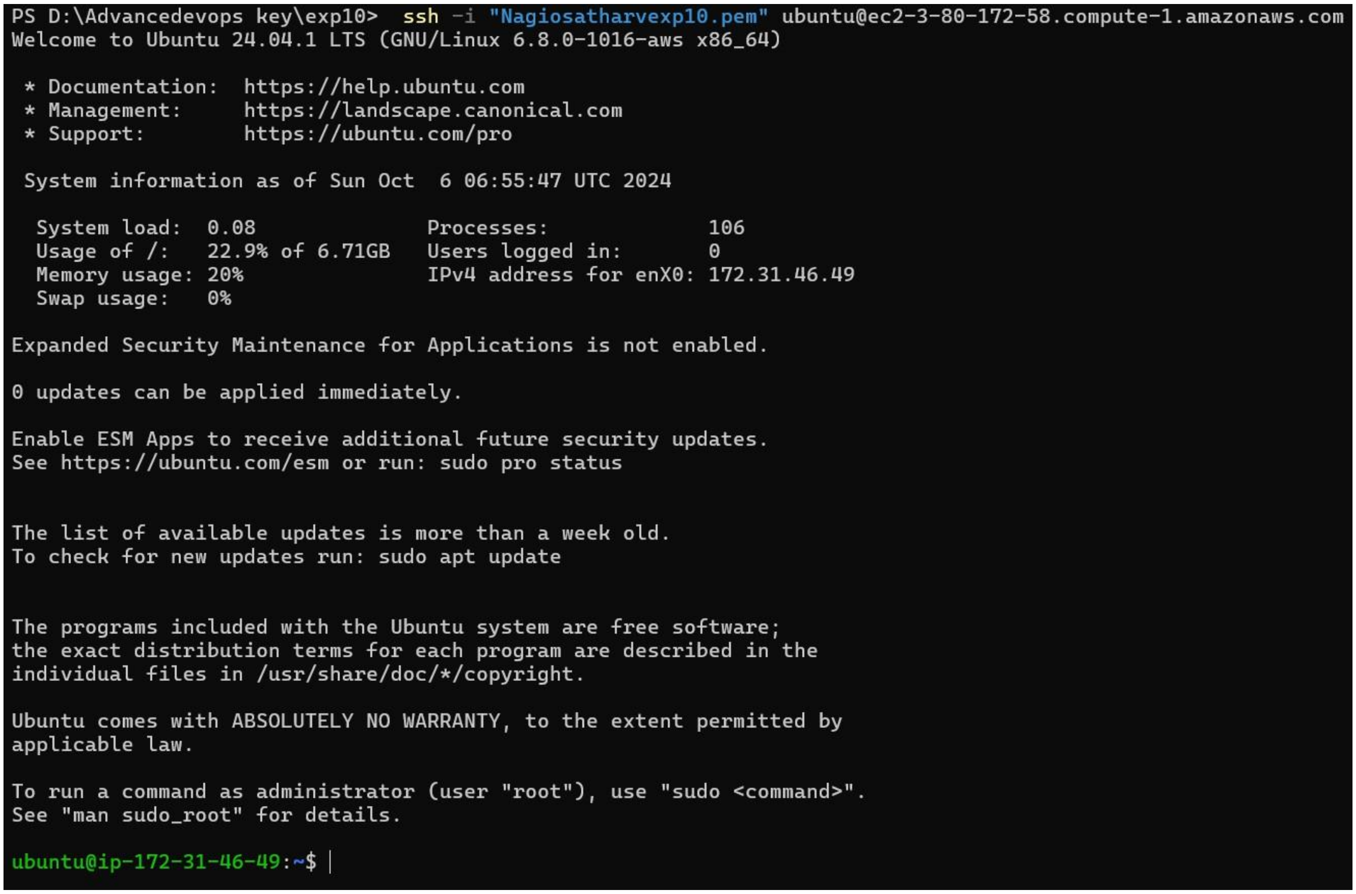
○ Create security group ● Select existing security group

Common security groups Info

Select security groups ▼

Compare security group rules

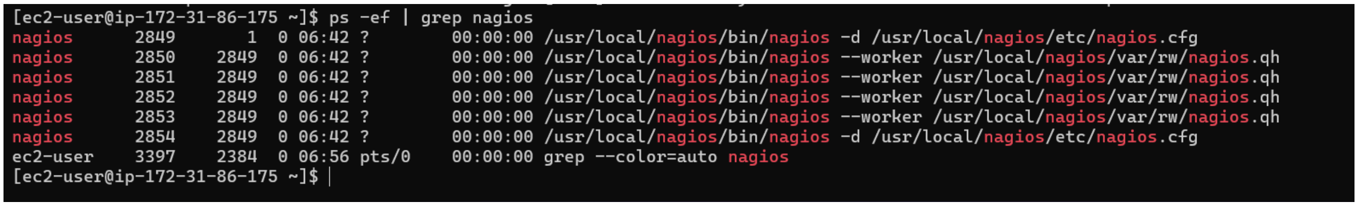
Nagios sg-0b59e140edaa1f431 ✕
VPC: vpc-0f7970ea32a533bcc



```
PS D:\Advancedevops key\exp10>  ssh -i "Nagiosatharvexp10.pem" ubuntu@ec2-3-80-172-58.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Oct  6 06:55:47 UTC 2024

  System load:  0.08               Processes:             106
  Usage of /:   22.9% of 6.71GB    Users logged in:       0
  Memory usage: 20%                IPv4 address for enX0: 172.31.46.49
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-49:~$
```

Step 4: Checking Nagios Status

To confirm Nagios is running correctly on the Nagios-host, execute the following:
`ps -ef | grep nagios`.

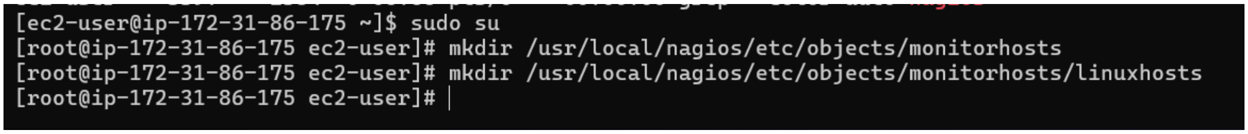


```
[ec2-user@ip-172-31-86-175 ~]$ ps -ef | grep nagios
nagios      2849       1  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      2850    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2851    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2852    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2853    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2854    2849  0 06:42 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    3397    2384  0 06:56 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-86-175 ~]$
```

Step 5: Creating Root Directories

Switch to the root user and create necessary directories for monitoring hosts:

```
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts.
```

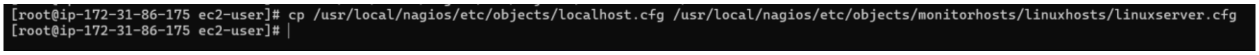


```
[ec2-user@ip-172-31-86-175 ~]$ sudo su
[root@ip-172-31-86-175 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-86-175 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-86-175 ec2-user]# |
```

Step 6: Configuring Monitoring for Linux Server

Copy the sample Nagios configuration file for localhost and create a new configuration file for the Linux server:

cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

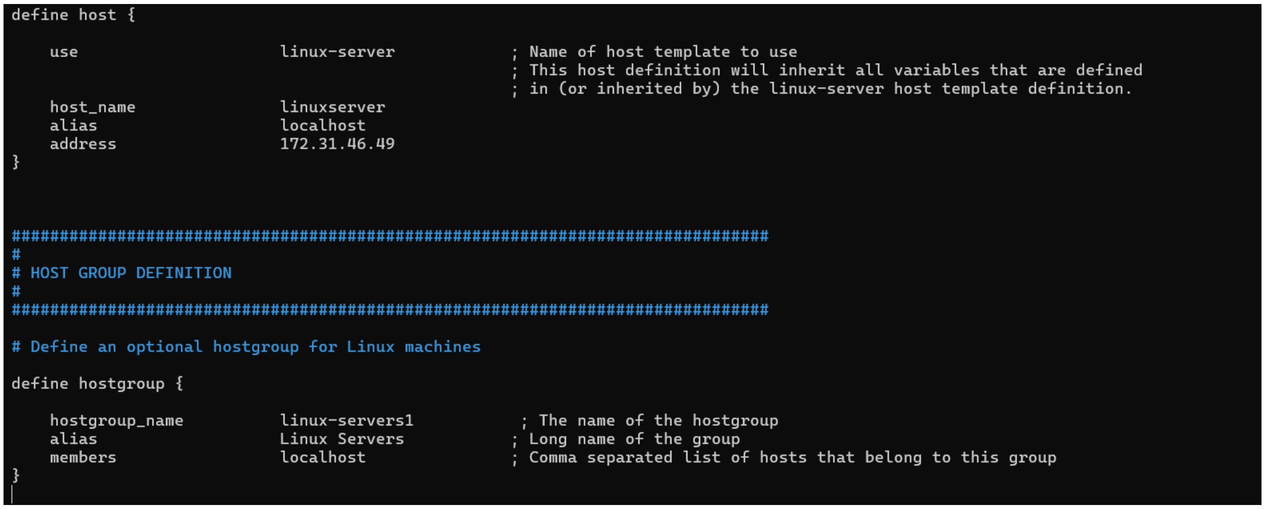


```
[root@ip-172-31-86-175 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-86-175 ec2-user]# |
```

Step 7: Editing Linux Server Configuration

Open the `linuxserver.cfg` file and modify the hostname, IP address, and hostgroup as follows:

- **hostname:** linuxserver
- **address:** Public IP of the Linux client
- **hostgroup_name:** linux-servers1
  Use the command:
  ```
  nano
  /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg.
  ```



```
define host {

    use                     linux-server            ; Name of host template to use
                                                    ; This host definition will inherit all variables that are defined
                                                    ; in (or inherited by) the linux-server host template definition.

    host_name               linuxserver
    alias                   localhost
    address                 172.31.46.49
}



###############################################################################
#
# HOST GROUP DEFINITION
#
###############################################################################

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name          linux-servers1           ; The name of the hostgroup
    alias                   Linux Servers           ; Long name of the group
    members                 localhost               ; Comma separated list of hosts that belong to this group
}
|
```

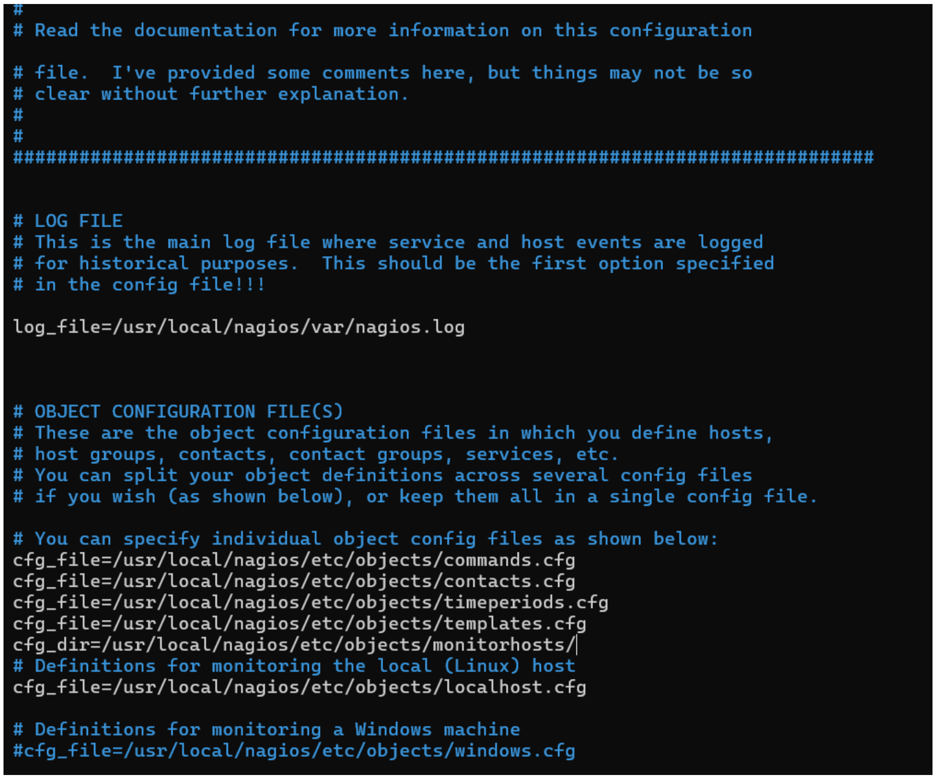
Step 8: Updating Nagios Configuration

Add the following line to Nagios' main configuration file to include the monitoring hosts directory:

`cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/`.

Edit the file using:

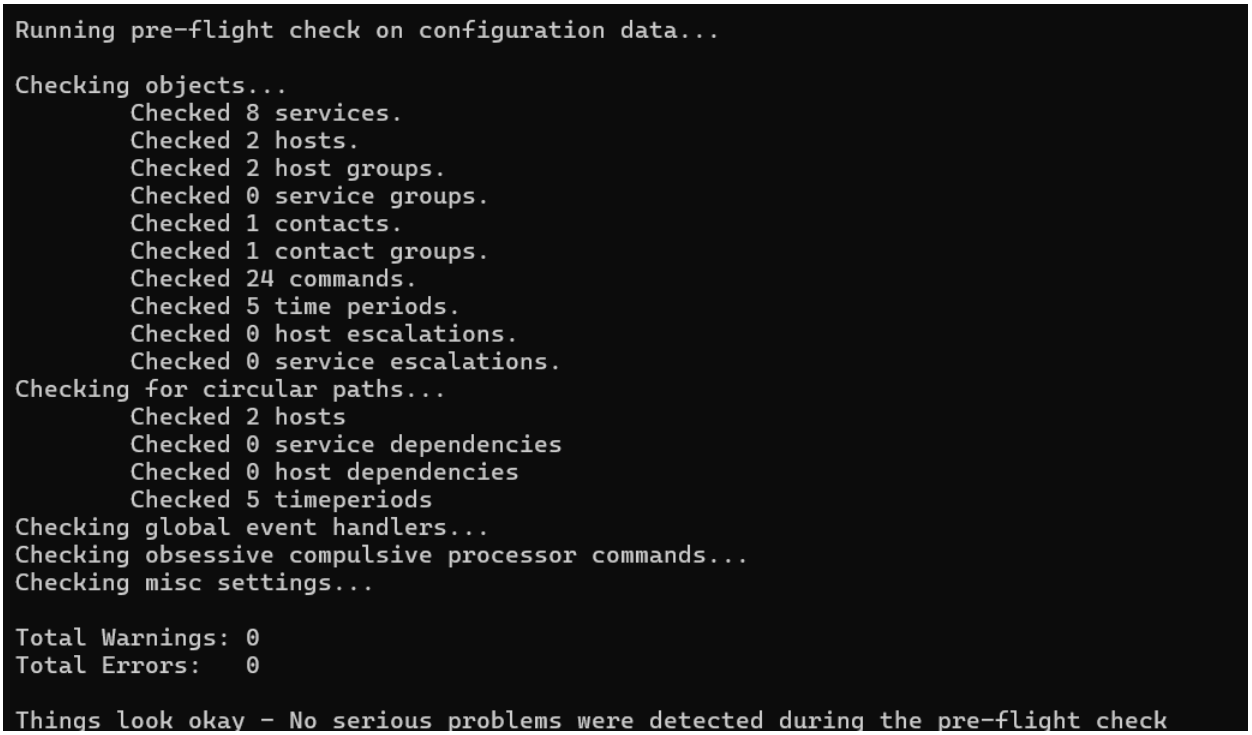
`nano /usr/local/nagios/etc/nagios.cfg`.



```
#
# Read the documentation for more information on this configuration

# file.  I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
###############################################################################


# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes.  This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log



# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/|
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Step 9: Verifying Nagios Configuration

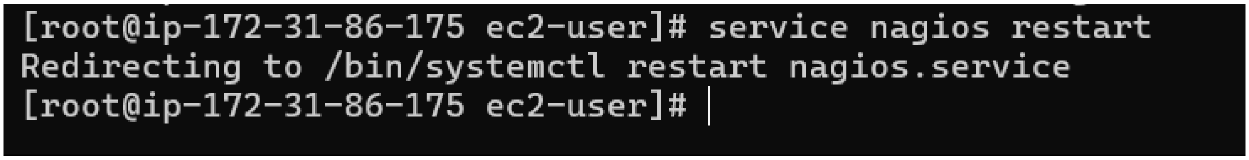
To check for any syntax errors in your configuration, run: `/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`.



```
Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 2 hosts.
        Checked 2 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 2 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
```

Step 10: Restarting Nagios

Restart the Nagios service to apply the configuration changes: `sudo service nagios restart`.

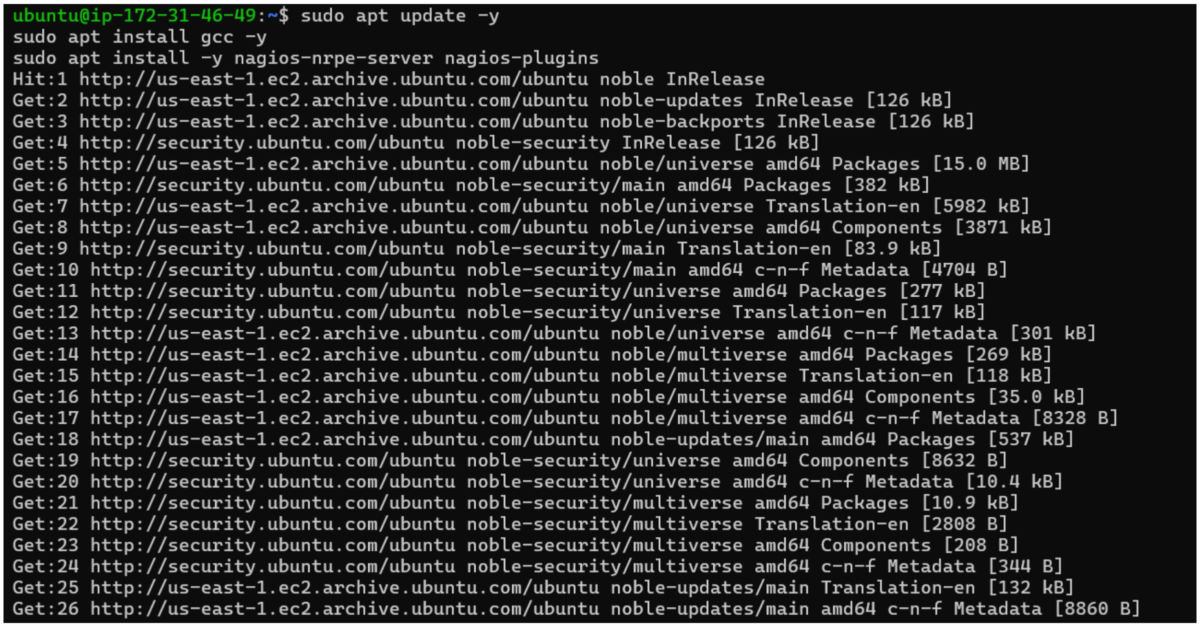


```
[root@ip-172-31-86-175 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-86-175 ec2-user]#
```

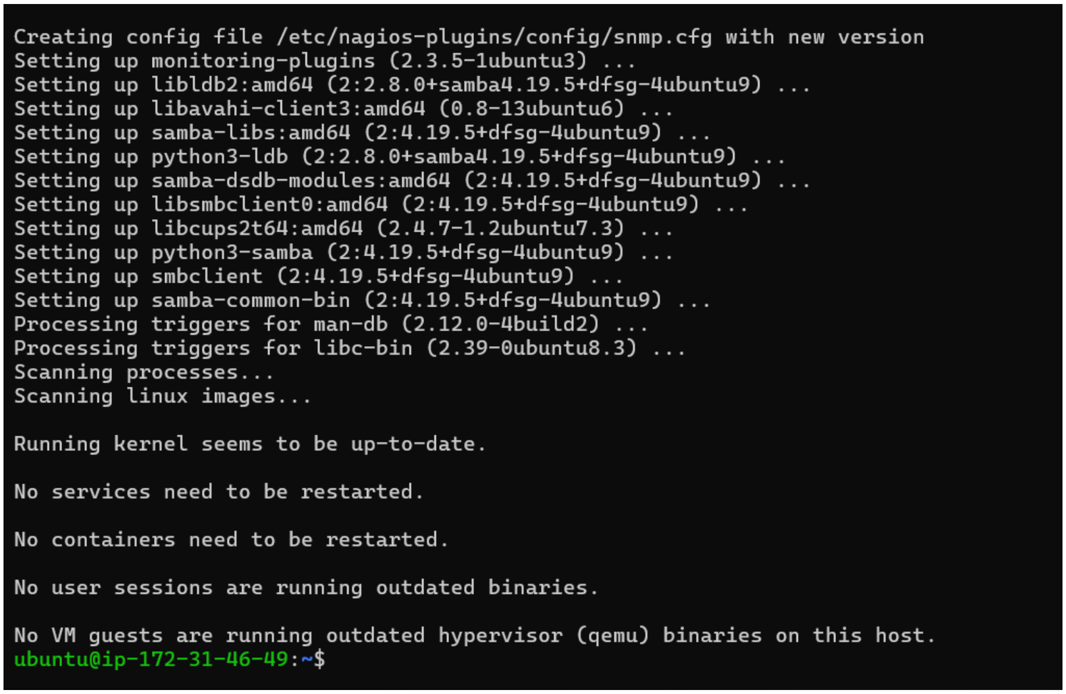
Step 11: Installing NRPE on Nagios Client

Connect to the Nagios-client instance and update the system. Then install NRPE and necessary Nagios plugins using:

sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins



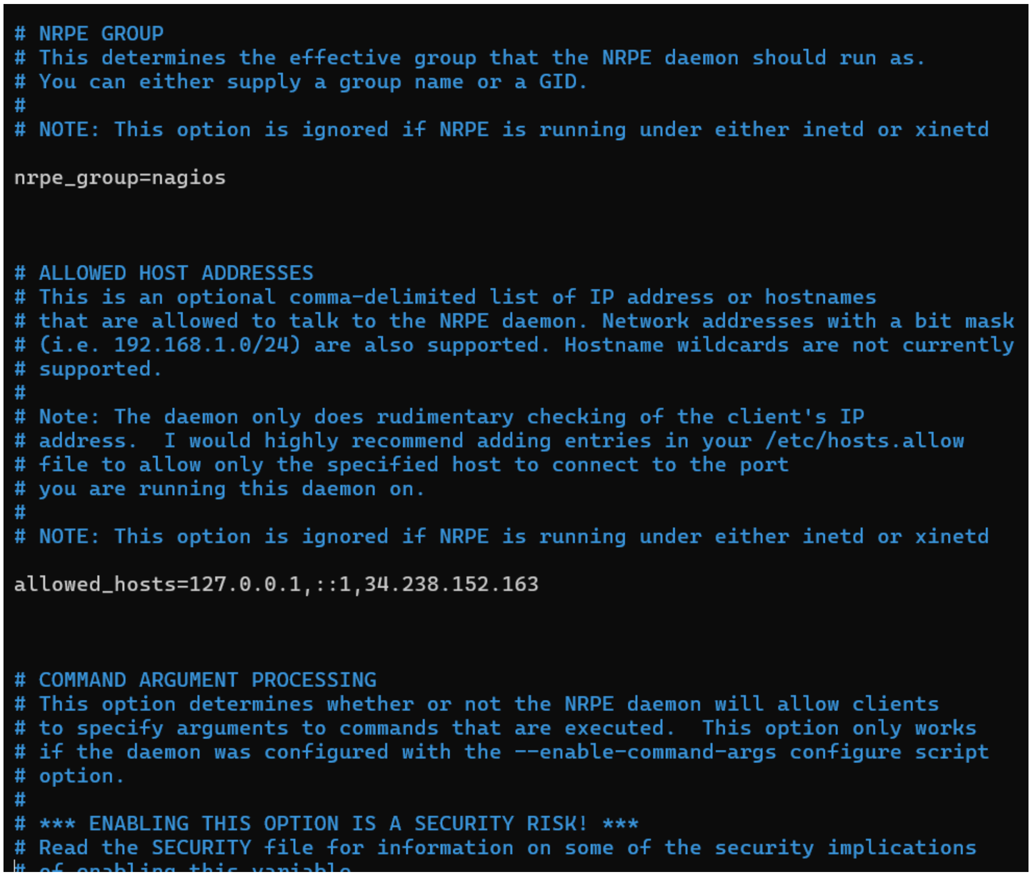
```
ubuntu@ip-172-31-46-49:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:24 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
```



```
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libldb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcups2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-46-49:~$
```

Step 12: Configuring NRPE

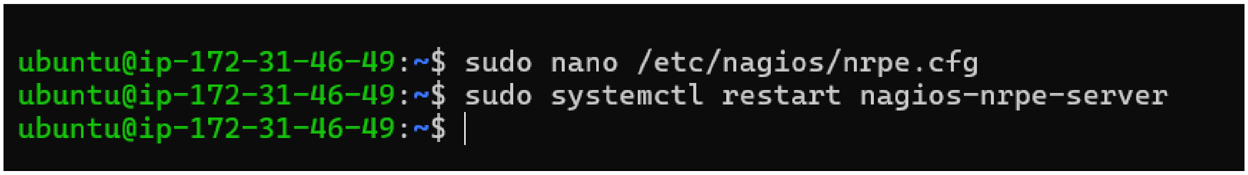
Edit the NRPE configuration file to allow the Nagios-host to communicate with the client. Add the Nagios-host’s IP address under `allowed_hosts`:
`sudo nano /etc/nagios/nrpe.cfg`.



```
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_group=nagios



# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,::1,34.238.152.163



# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed.  This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
```

Step 13: Restarting NRPE

After editing the NRPE configuration, restart the NRPE server:
`sudo systemctl restart nagios-nrpe-server`.

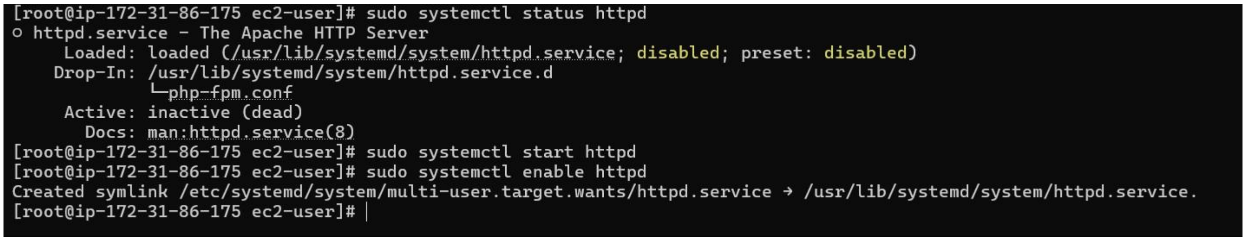


```
ubuntu@ip-172-31-46-49:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-46-49:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-46-49:~$ |
```

Step 14: Checking Nagios and HTTPD Services

On the Nagios-host, check the status of Nagios and ensure that the HTTPD service is active:

```
sudo systemctl status nagios
sudo systemctl status httpd.
```



```
[root@ip-172-31-86-175 ec2-user]# sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-86-175 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-86-175 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-86-175 ec2-user]# |
```

If HTTPD is not active, start and enable it:

```
sudo systemctl start httpd
sudo systemctl enable httpd.
```

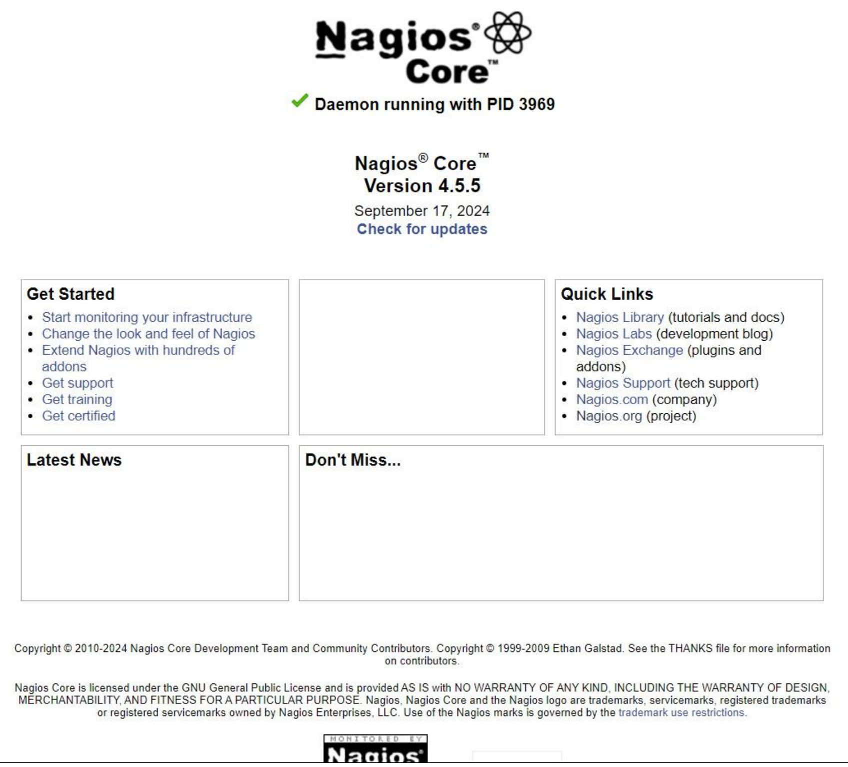
Step 15: Accessing Nagios Dashboard

To view the Nagios dashboard, open your browser and go to:

`http://<Nagios-host-ip>/nagios`.

Click on **Hosts** from the left panel to view the status of your Linux server.

Name:Atharv Nikam Div:D15C Roll:36



Daemon running with PID 3969

Nagios® Core™
Version 4.5.5

September 17, 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

Quick Links

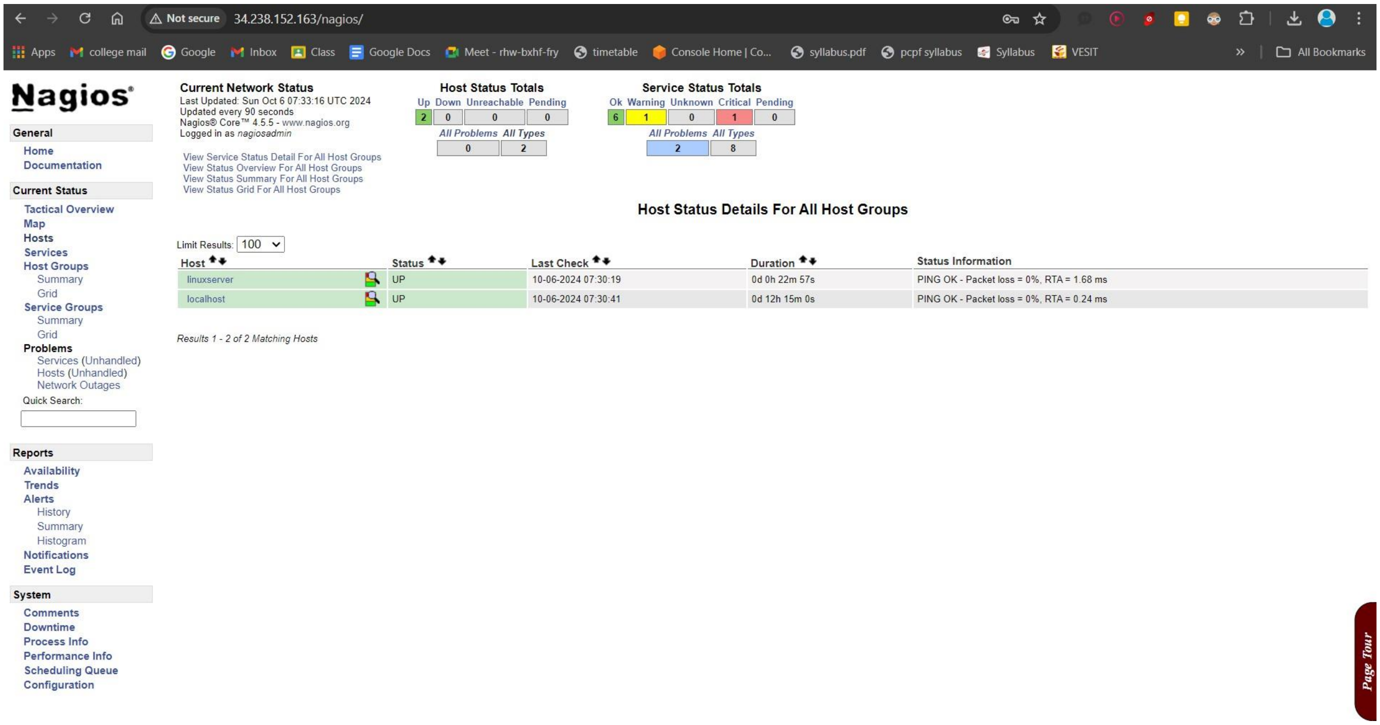
- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

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.



Current Network Status
Last Updated: Sun Oct 6 07:33:16 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - 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
2	0	0	0

All Problems	*All Types*
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	1	0

All Problems	*All Types*
2	8

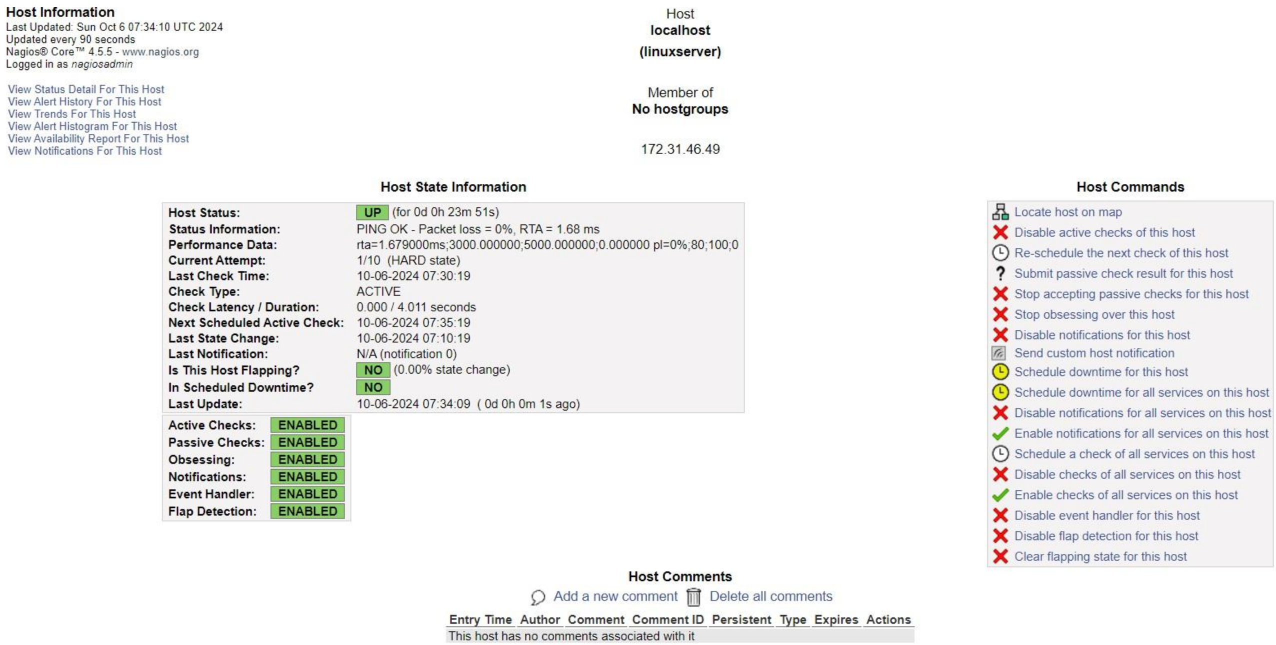
Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-06-2024 07:30:19	0d 0h 22m 57s	PING OK - Packet loss = 0%, RTA = 1.68 ms
localhost	UP	10-06-2024 07:30:41	0d 12h 15m 0s	PING OK - Packet loss = 0%, RTA = 0.24 ms

Results 1 - 2 of 2 Matching Hosts

Name:Atharv Nikam Div:D15C Roll:36



Host Information
Last Updated: Sun Oct 6 07:34:10 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as *nagiosadmin*

View Status Detail For This Host
View Alert History For This Host
View Trends For This Host
View Alert Histogram For This Host
View Availability Report For This Host
View Notifications For This Host

Host
localhost
(linuxserver)

Member of
No hostgroups

172.31.46.49

Host State Information

Host Status:	UP (for 0d 0h 23m 51s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.68 ms
Performance Data:	rta=1.679000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-06-2024 07:30:19
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.011 seconds
Next Scheduled Active Check:	10-06-2024 07:35:19
Last State Change:	10-06-2024 07:10:19
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-06-2024 07:34:09 (0d 0h 0m 1s ago)

Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

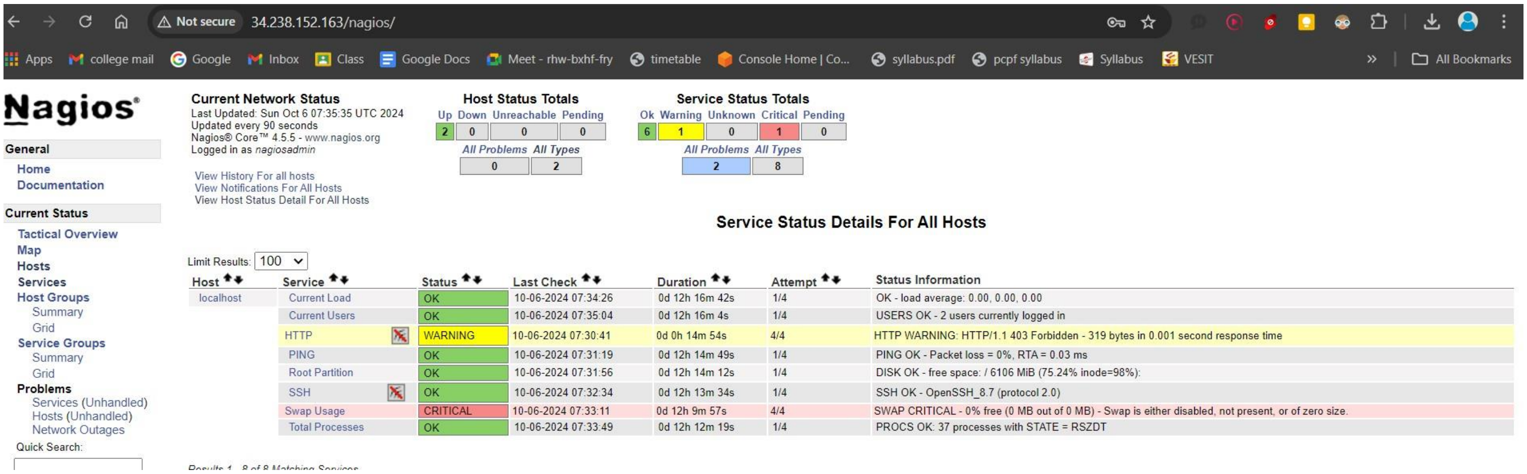
Host Commands

- Locate host on map
- ✗ Disable active checks of this host
- Re-schedule the next check of this host
- ? Submit passive check result for this host
- ✗ Stop accepting passive checks for this host
- ✗ Stop obsessing over this host
- ✗ Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- ✗ Disable notifications for all services on this host
- ✓ Enable notifications for all services on this host
- Schedule a check of all services on this host
- ✗ Disable checks of all services on this host
- ✓ Enable checks of all services on this host
- ✗ Disable event handler for this host
- ✗ Disable flap detection for this host
- ✗ Clear flapping state for this host

Host Comments

Add a new comment Delete all comments

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it							



Current Network Status
Last Updated: Sun Oct 6 07:35:35 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - 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
2	0	0	0

All Problems	*All Types*
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	1	0

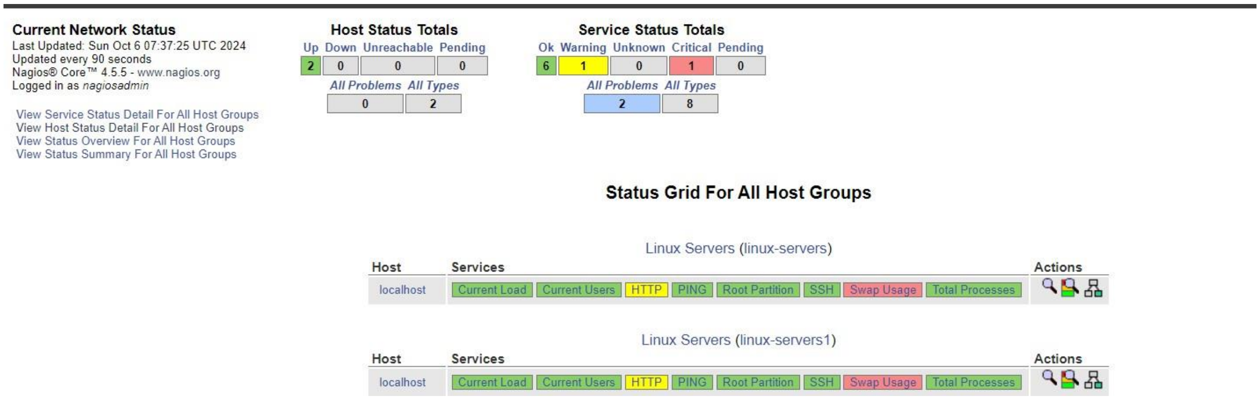
All Problems	*All Types*
2	8

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-06-2024 07:34:26	0d 12h 16m 42s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-06-2024 07:35:04	0d 12h 16m 4s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	10-06-2024 07:30:41	0d 0h 14m 54s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
	PING	OK	10-06-2024 07:31:19	0d 12h 14m 49s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-06-2024 07:31:56	0d 12h 14m 12s	1/4	DISK OK - free space: / 6106 MiB (75.24% inode=98%):
	SSH	OK	10-06-2024 07:32:34	0d 12h 13m 34s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-06-2024 07:33:11	0d 12h 9m 57s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-06-2024 07:33:49	0d 12h 12m 19s	1/4	PROCS OK: 37 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services



Current Network Status
Last Updated: Sun Oct 6 07:37:25 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as *nagiosadmin*

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

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems	*All Types*
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	1	0

All Problems	*All Types*
2	8

Status Grid For All Host Groups

Linux Servers (linux-servers)

Host	Services	Actions
localhost	Current Load Current Users HTTP PING Root Partition SSH Swap Usage Total Processes	

Linux Servers (linux-servers1)

Host	Services	Actions
localhost	Current Load Current Users HTTP PING Root Partition SSH Swap Usage Total Processes	

Conclusion:

This experiment was designed to set up monitoring for ports, services, and a Linux server using Nagios. By carefully configuring both the Nagios host and client, we were able to monitor essential network services and assess server performance. The experiment demonstrated how Nagios can be used effectively to track system metrics such as CPU usage and memory consumption. This hands-on experience highlights the importance of proactive monitoring in maintaining server health and ensuring the availability of critical services across Linux and Windows platforms.