Name: Alok Yadav Div: D15C Roll No. 59

EXPERIMENT NO. 10

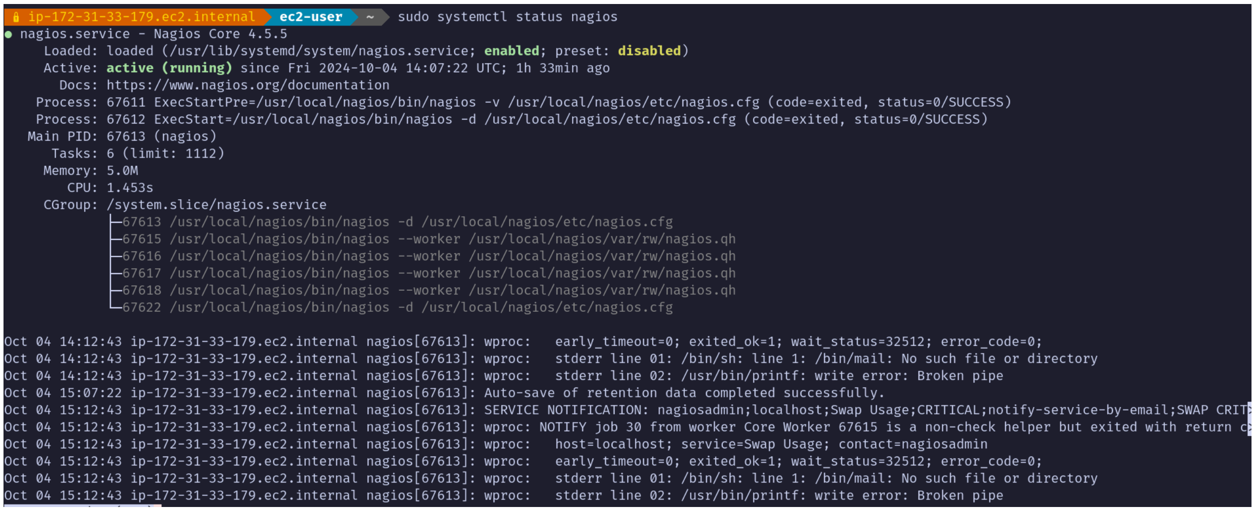
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Prerequisites:
Nagios Server running on Amazon Linux Machine.

STEPS:

1. To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host).

   sudo systemctl status nagios

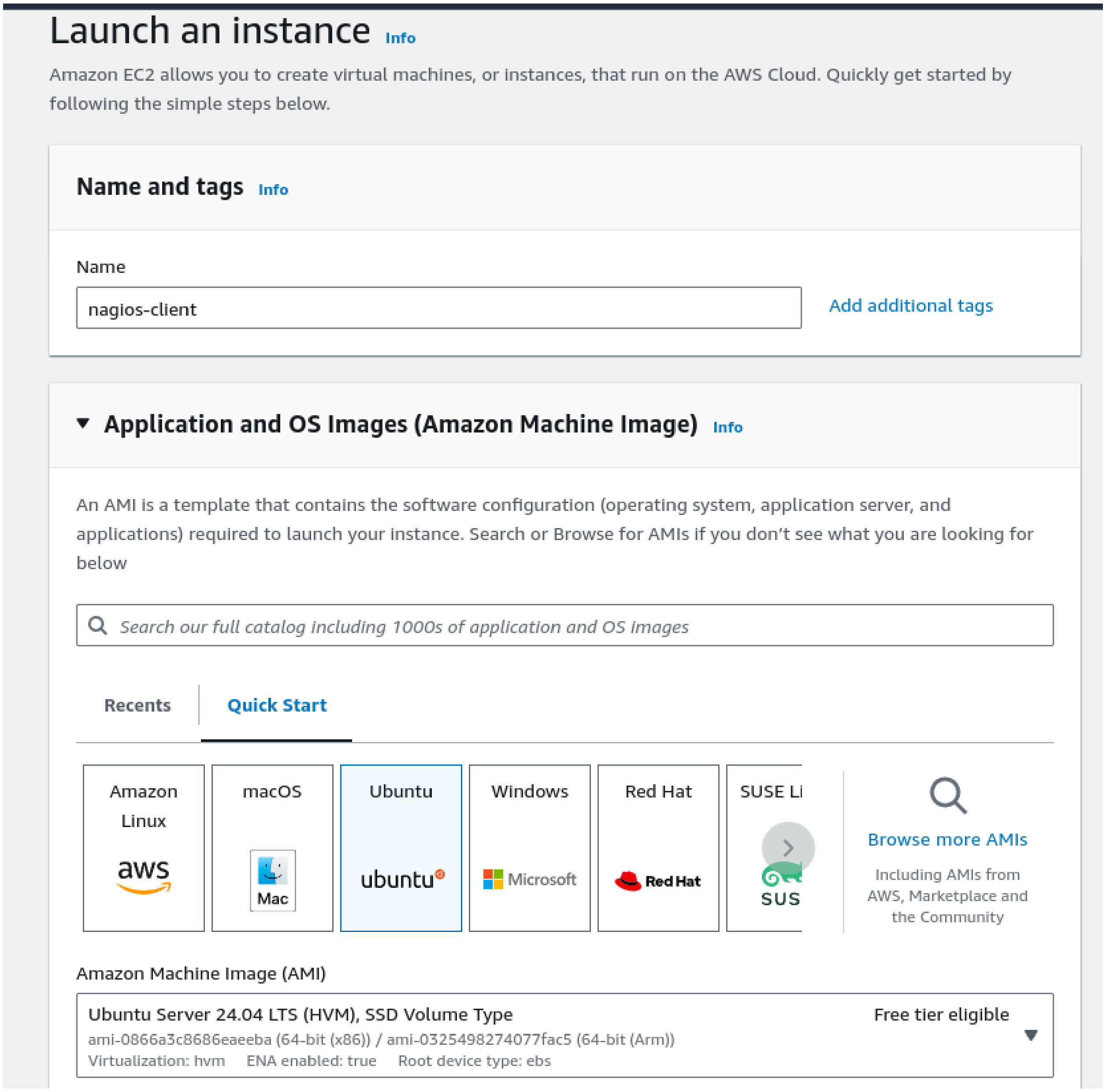


```
🔒 ip-172-31-33-179.ec2.internal  ec2-user  ~  sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-10-04 14:07:22 UTC; 1h 33min ago
       Docs: https://www.nagios.org/documentation
    Process: 67611 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 67612 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 67613 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.0M
        CPU: 1.453s
     CGroup: /system.slice/nagios.service
             ├─67613 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─67615 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67616 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67617 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67618 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─67622 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 14:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 04 14:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 04 14:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   stderr line 02: /usr/bin/printf: write error: Broken pipe
Oct 04 15:07:22 ip-172-31-33-179.ec2.internal nagios[67613]: Auto-save of retention data completed successfully.
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRIT>
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc: NOTIFY job 30 from worker Core Worker 67615 is a non-check helper but exited with return c>
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   host=localhost; service=Swap Usage; contact=nagiosadmin
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 04 15:12:43 ip-172-31-33-179.ec2.internal nagios[67613]: wproc:   stderr line 02: /usr/bin/printf: write error: Broken pipe
```

   As nagios service on main machine is running, we can proceed further

2. Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

Name: Alok Yadav Div: D15C Roll No. 59



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: nagios-client

Application and OS Images (Amazon Machine Image) Info

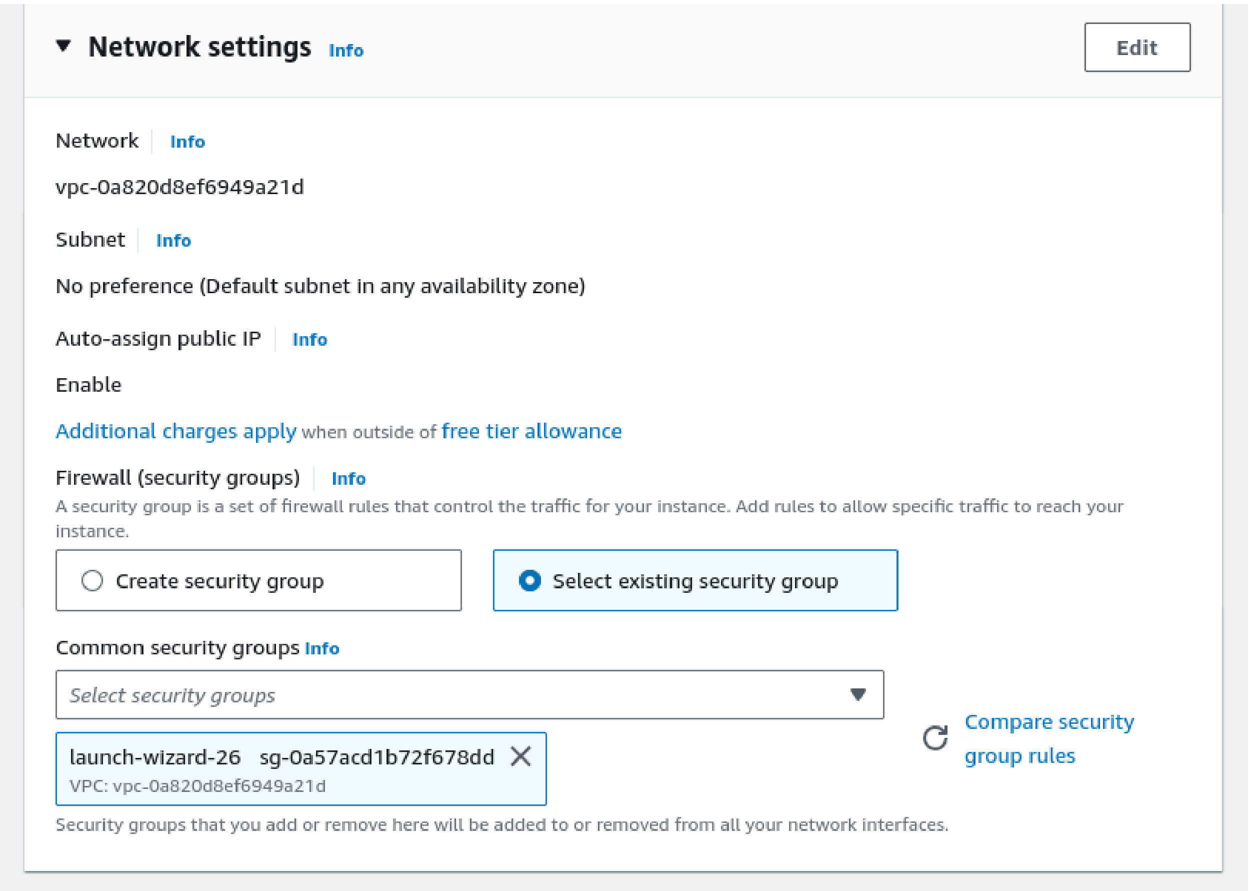
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Select the Existing Security Group and select the Security Group that we have created in
Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



▼ **Network settings** Info

Edit

Network | Info

vpc-0a820d8ef6949a21d

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 ▼

launch-wizard-26 sg-0a57acd1b72f678dd ✕
VPC: vpc-0a820d8ef6949a21d

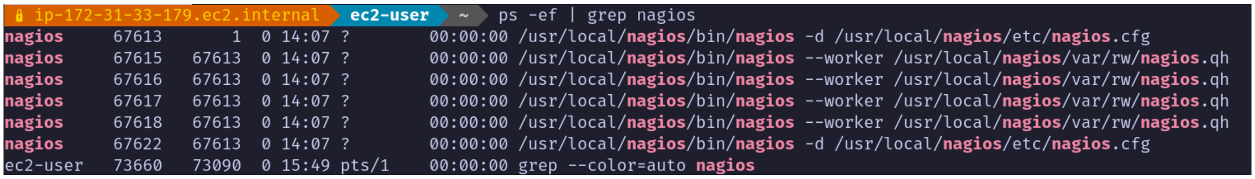
Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Now perform all the commands on the Nagios-host till step 10

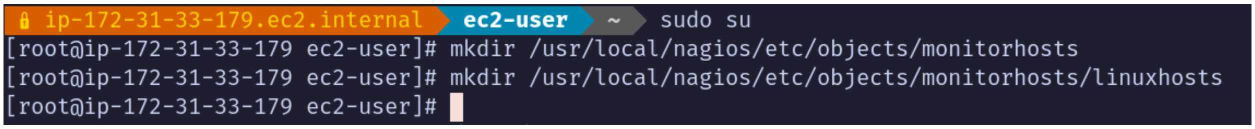
4. Now on the server Nagios-host run the following command.

ps -ef | grep nagios



```
ip-172-31-33-179.ec2.internal  ec2-user  ~  ps -ef | grep nagios
nagios     67613       1  0 14:07 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     67615   67613  0 14:07 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67616   67613  0 14:07 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67617   67613  0 14:07 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67618   67613  0 14:07 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67622   67613  0 14:07 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user   73660   73090  0 15:49 pts/1    00:00:00 grep --color=auto nagios
```

5. Now Become root user and create root directories.
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts



```
 ip-172-31-33-179.ec2.internal  ec2-user  ~  sudo su
[root@ip-172-31-33-179 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-33-179 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-33-179 ec2-user]# 
```

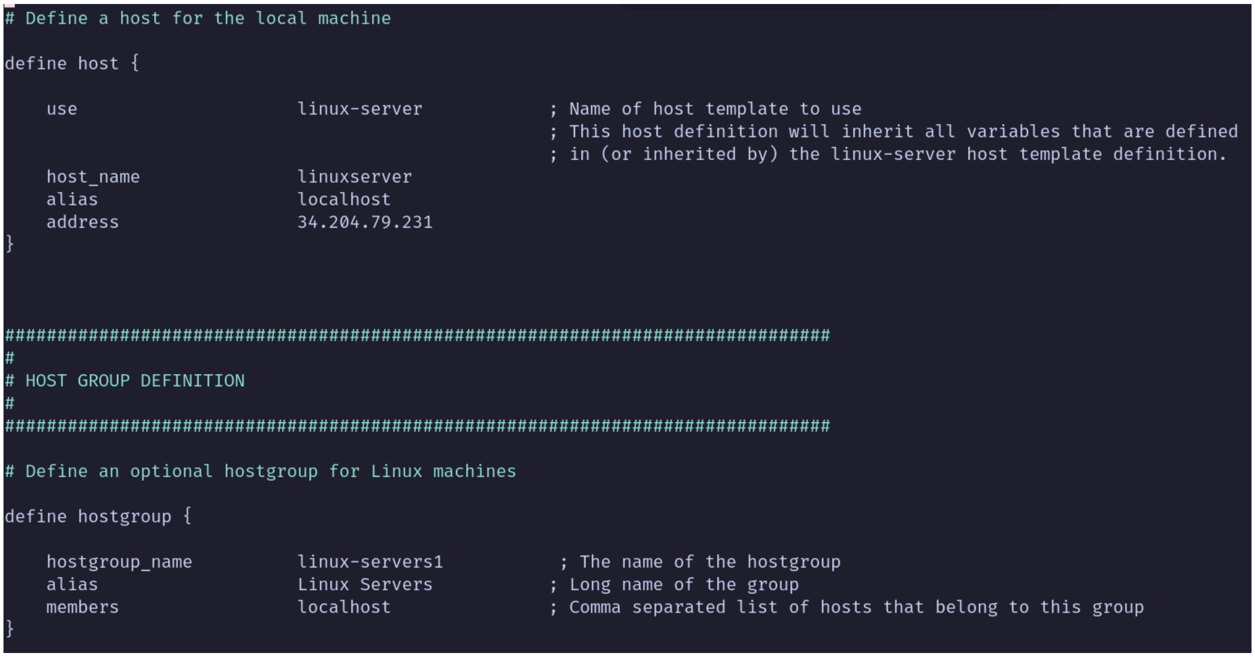
6. Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

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

7. Open linuxserver.cfg using nano and make the following changes in all Positions everywhere in file

nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

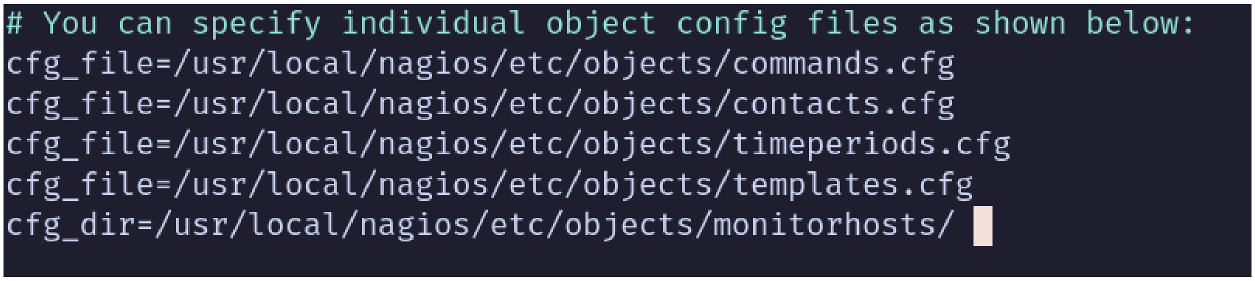
Change hostname to **linuxserver.**
Change address to the **public IP of your Linux client**.
Set hostgroup_name to **linux-servers1**.



```
# Define a host for the local machine

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                 34.204.79.231
}



###############################################################################
#
# 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
}
```

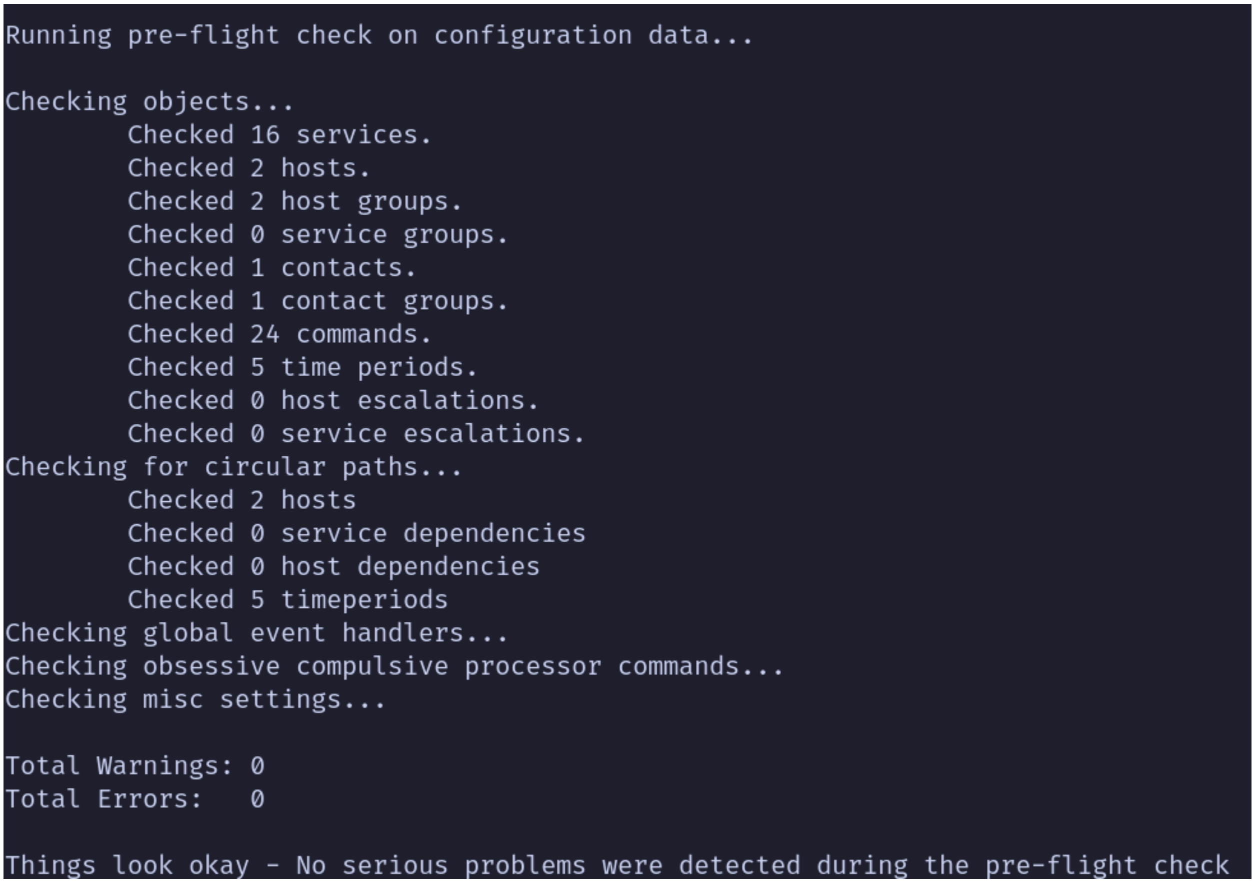
8. Now update the Nagios config file .Add the following line in the file. Line to add
:
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

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



```
# 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/
```

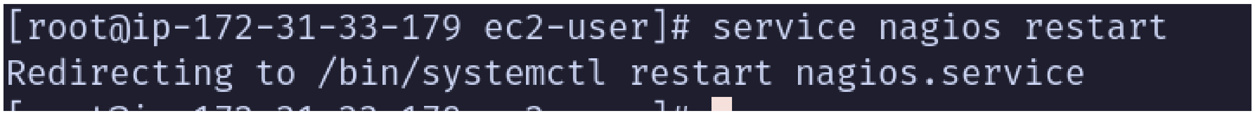
9. Now Verify the configuration files by running the following commands.
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



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

Checking objects...
        Checked 16 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
```

We got no errors and warnings for current configuration

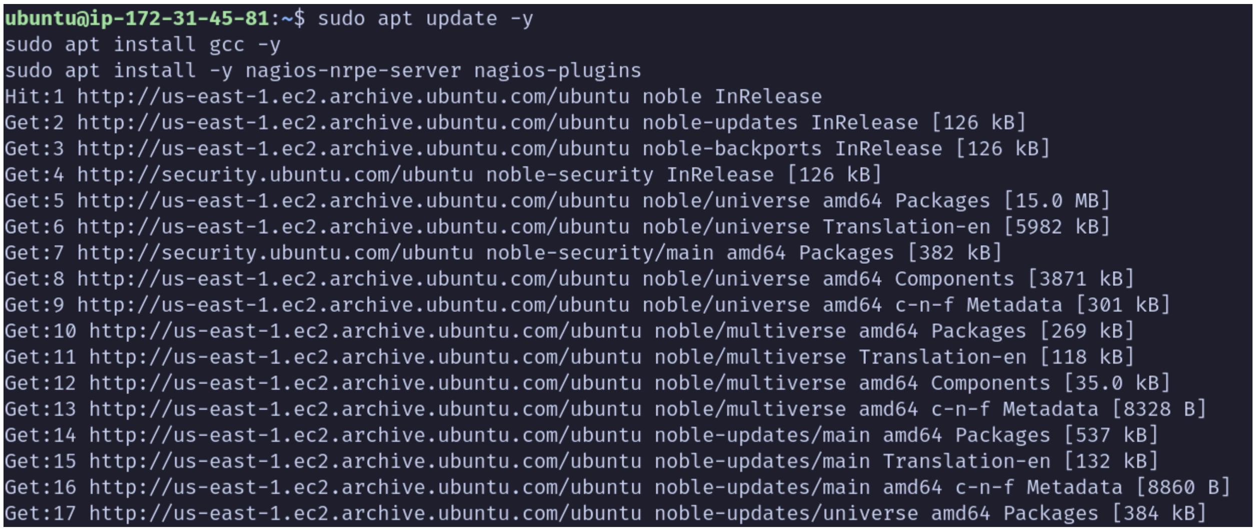
10. Now restart the services of nagios by running the following command.
service nagios restart



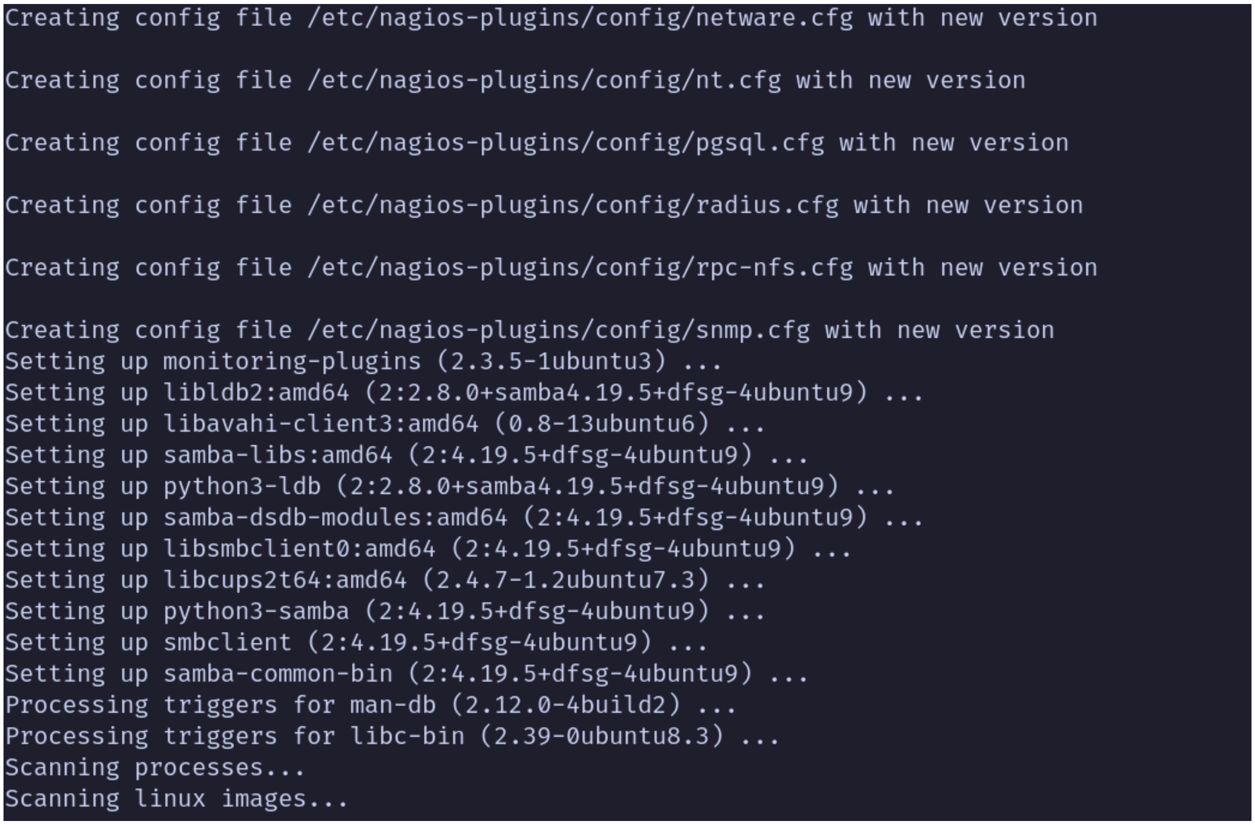
```
[root@ip-172-31-33-179 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

11. Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

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

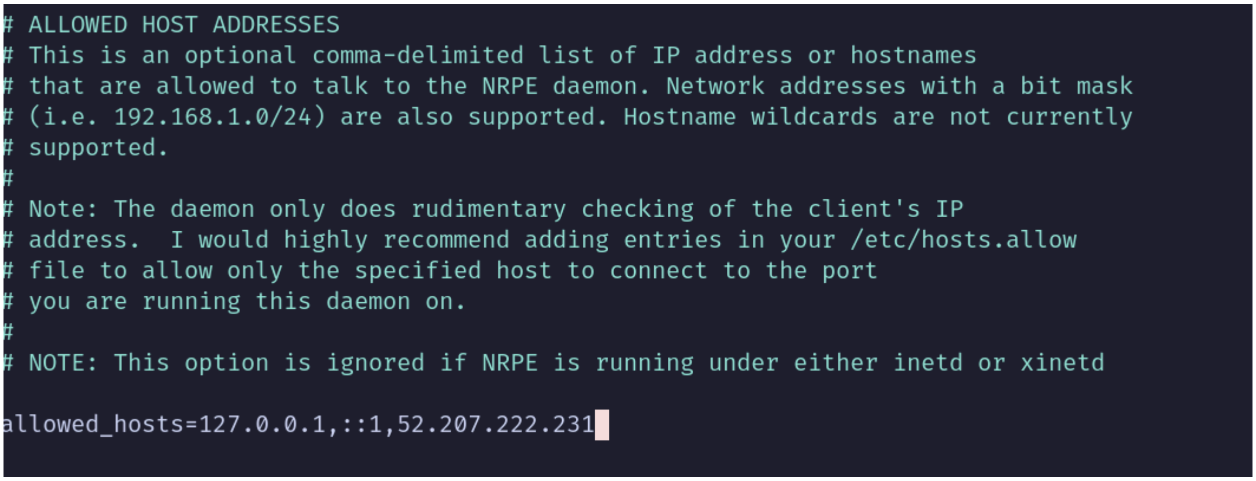


```
ubuntu@ip-172-31-45-81:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
```



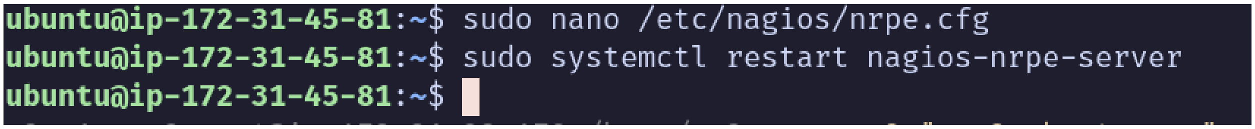
```
Creating config file /etc/nagios-plugins/config/netware.cfg with new version

Creating config file /etc/nagios-plugins/config/nt.cfg with new version

Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version

Creating config file /etc/nagios-plugins/config/radius.cfg with new version

Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version

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...
```

12. Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host public IP address.



```
# 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,::1,52.207.222.231
```

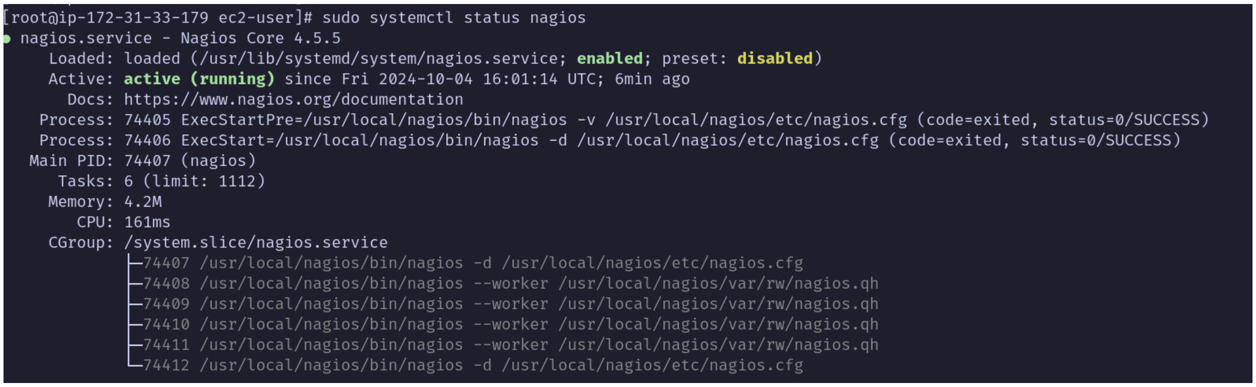
13. Now restart the NRPE server by this command.
sudo systemctl restart nagios-nrpe-server



```
ubuntu@ip-172-31-45-81:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-45-81:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-45-81:~$ 
```

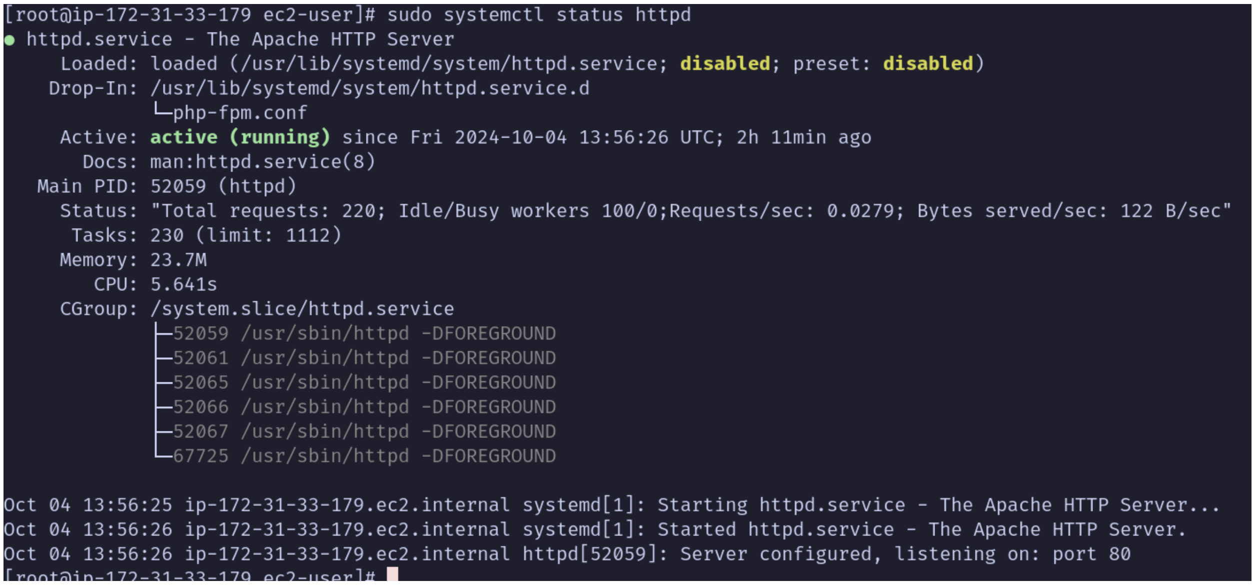
14. Now again check the status of Nagios by running this command on Nagios-host and
also check httpd is active and run the command to active it.

sudo systemctl status nagios



```
[root@ip-172-31-33-179 ec2-user]# 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 Fri 2024-10-04 16:01:14 UTC; 6min ago
       Docs: https://www.nagios.org/documentation
    Process: 74405 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 74406 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 74407 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.2M
        CPU: 161ms
     CGroup: /system.slice/nagios.service
             ├─74407 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─74408 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─74409 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─74410 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─74411 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─74412 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

sudo systemctl status httpd



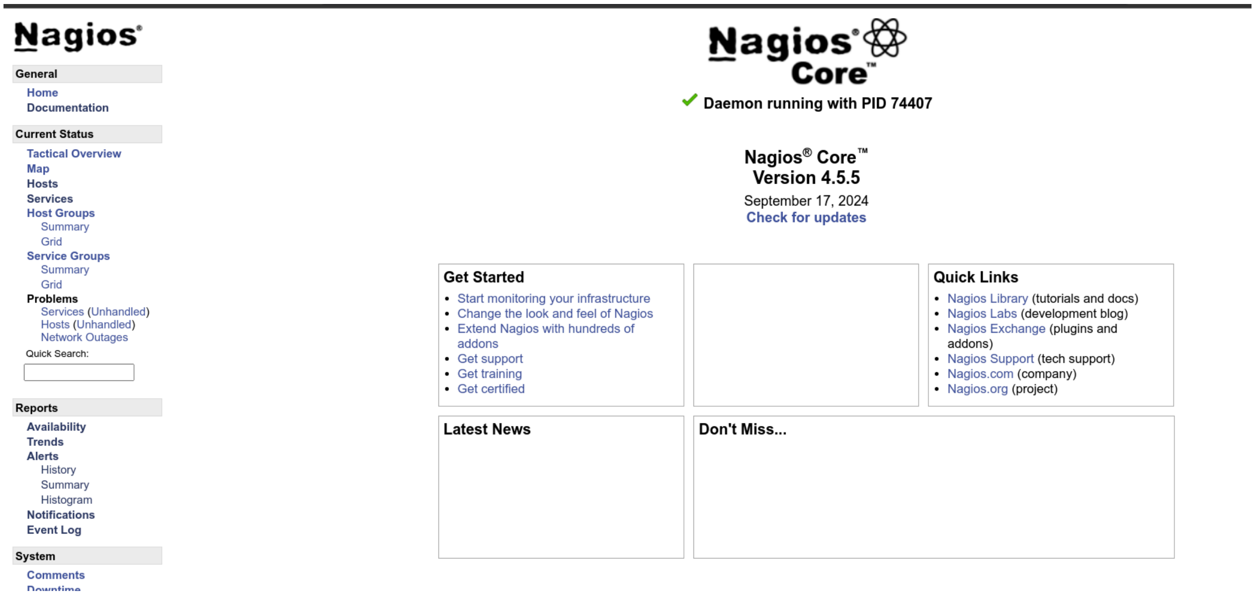
```
[root@ip-172-31-33-179 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2024-10-04 13:56:26 UTC; 2h 11min ago
       Docs: man:httpd.service(8)
   Main PID: 52059 (httpd)
     Status: "Total requests: 220; Idle/Busy workers 100/0;Requests/sec: 0.0279; Bytes served/sec: 122 B/sec"
      Tasks: 230 (limit: 1112)
     Memory: 23.7M
        CPU: 5.641s
     CGroup: /system.slice/httpd.service
             ├─52059 /usr/sbin/httpd -DFOREGROUND
             ├─52061 /usr/sbin/httpd -DFOREGROUND
             ├─52065 /usr/sbin/httpd -DFOREGROUND
             ├─52066 /usr/sbin/httpd -DFOREGROUND
             ├─52067 /usr/sbin/httpd -DFOREGROUND
             └─67725 /usr/sbin/httpd -DFOREGROUND

Oct 04 13:56:25 ip-172-31-33-179.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 04 13:56:26 ip-172-31-33-179.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 04 13:56:26 ip-172-31-33-179.ec2.internal httpd[52059]: Server configured, listening on: port 80
[root@ip-172-31-33-179 ec2-user]#
```

Both nagios and httpd service is running fine on host system

15. Now to check Nagios dashboard go to http://<nagios host ip>/nagios
Eg. **http://34.207.68.187/nagios**

Enter username as nagiosadmin and password which you set in Exp 9 if prompted.

Name: Alok Yadav Div: D15C Roll No. 59



✓ Daemon running with PID 74407

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

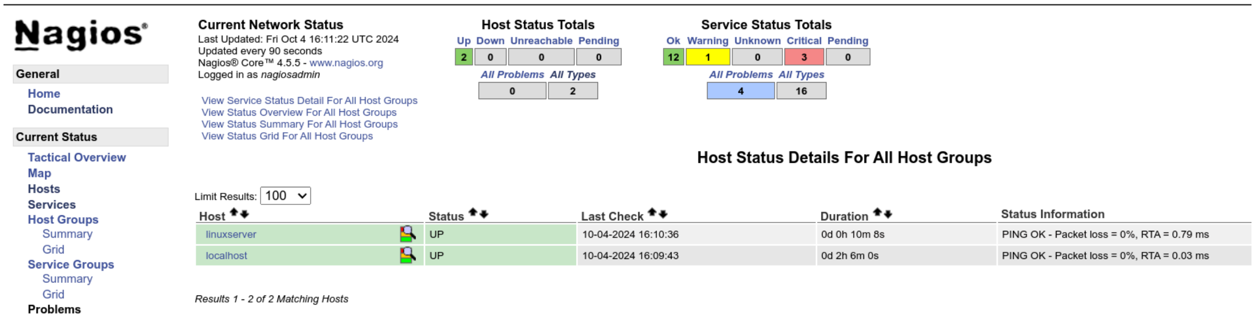
Quick Links

- 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...

Now Click on Hosts from left side panel



Current Network Status
Last Updated: Fri Oct 4 16:11:22 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
12	1	0	3	0

All Problems	*All Types*
4	16

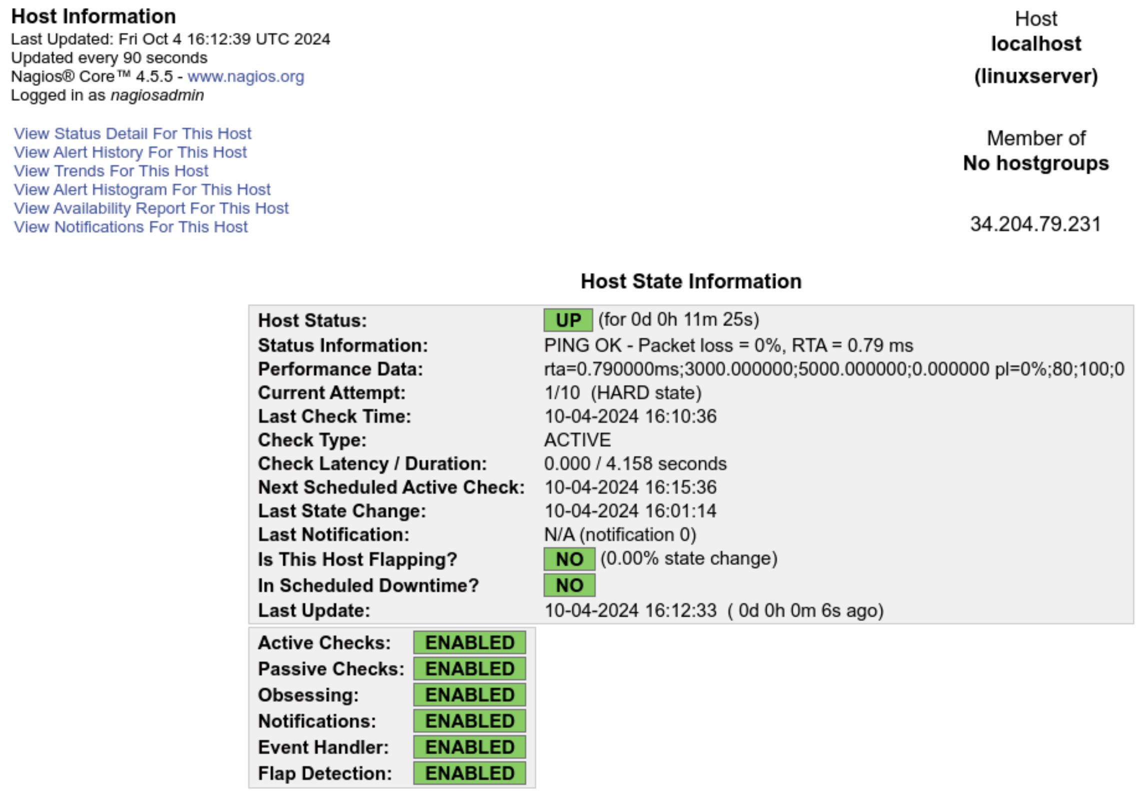
Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-04-2024 16:10:36	0d 0h 10m 8s	PING OK - Packet loss = 0%, RTA = 0.79 ms
localhost	UP	10-04-2024 16:09:43	0d 2h 6m 0s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

Our nagios client is showing up on nagios host dashboard



Host Information
Last Updated: Fri Oct 4 16:12:39 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

34.204.79.231

Host State Information

Host Status:	UP (for 0d 0h 11m 25s)
Status Information:	PING OK - Packet loss = 0%, RTA = 0.79 ms
Performance Data:	rta=0.790000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-04-2024 16:10:36
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.158 seconds
Next Scheduled Active Check:	10-04-2024 16:15:36
Last State Change:	10-04-2024 16:01:14
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-04-2024 16:12:33 (0d 0h 0m 6s ago)

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

Here we can see current status of nagios client mahine which is up and running fine

Conclusion:

In this experiment, we created a new EC2 Linux instance and set up the Nagios client on it. We then connected the client to the Nagios host machine, allowing us to monitor alerts and status for both the host and clients on a single dashboard. It is important to configure the Nagios client with the correct host IP, as failure to do so will prevent the connection. After completing the setup, we were able to successfully view the system health details of both the client and host on the unified Nagios dashboard.