Name: Anish Kulkarni Roll No.: 29 Class: D15C AY: 2024-25

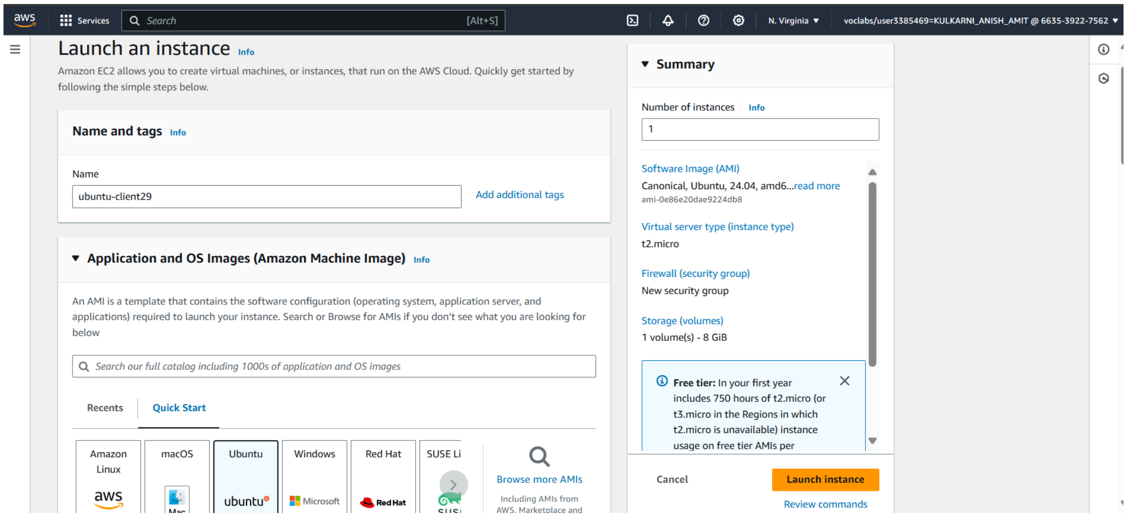
Experiment 10

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

Prerequisites: An Amazon Linux instance with nagios (nagios-server) is already set up.

Steps:

Step 1: Navigate to EC2 on the AWS console using the 'Services' section and click on 'Create instance'. Give your instance a name and choose 'Ubuntu' as the instance type.



Launch an instance

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

Name: ubuntu-client29

Application and OS Images (Amazon Machine Image)

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

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more ami-0e86e20dae9224db8

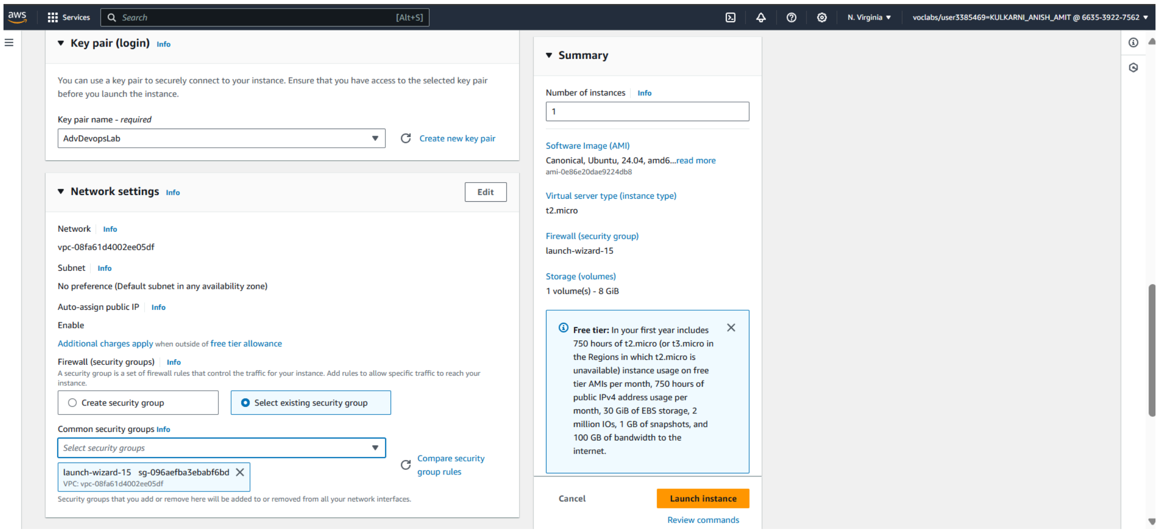
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per

Ensure that you choose the same key pair and security group for the Ubuntu client instance as you did for the Nagios host instance. Then, click on 'Create instance'.



Key pair (login)

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*: AdvDevopsLab

Network settings

Network: vpc-08fa61d4002ee05df

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

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

launch-wizard-15 sg-096aefba3ebabf6bd
VPC: vpc-08fa61d4002ee05df

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

Summary

Number of instances: 1

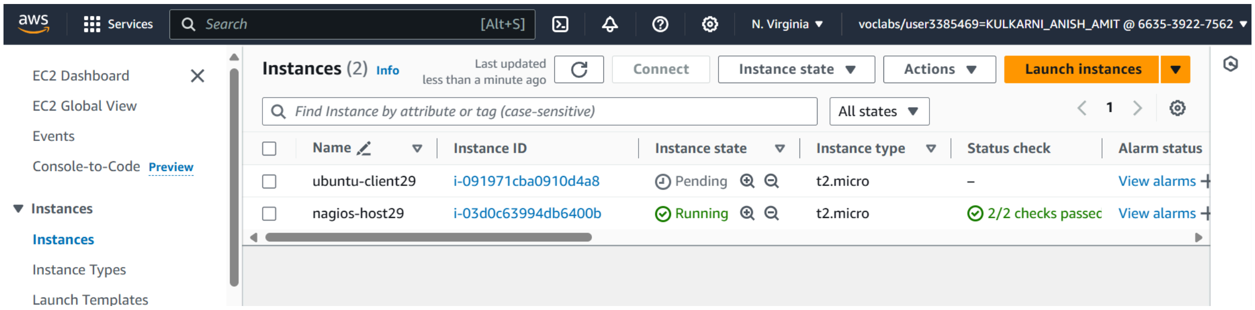
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more ami-0e86e20dae9224db8

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-15

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

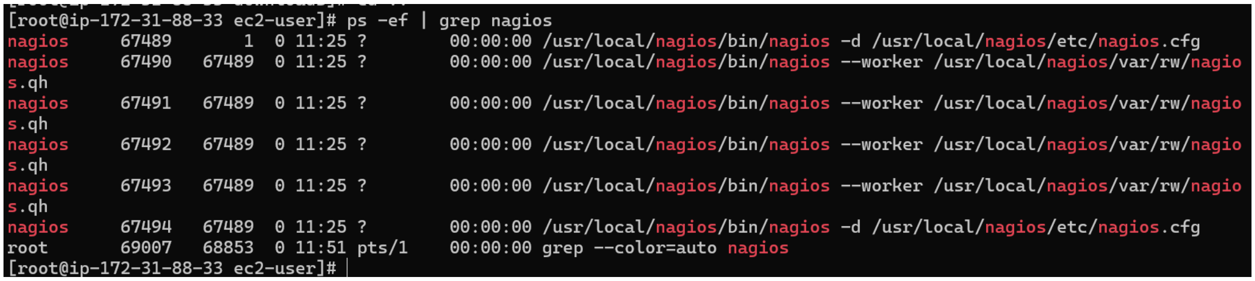


Name	Instance ID	Instance state	Instance type	Status check	Alarm status
ubuntu-client29	i-091971cba0910d4a8	Pending	t2.micro	–	View alarms +
nagios-host29	i-03d0c63994db6400b	Running	t2.micro	2/2 checks passec	View alarms +

Your Ubuntu client instance gets created along with the Nagios host instance.

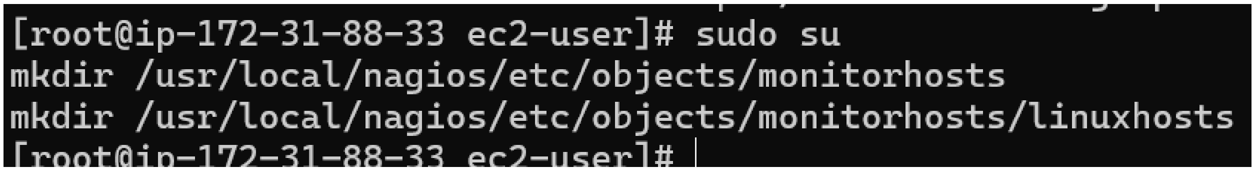
Step 2: Click on the instance ID of your nagios-server instance and click on ‘Connect’. Then, click on ‘SSH client’ and copy the command under ‘Example’. Then, open the terminal in the folder where the .pem file for your instance’s key pair is located and paste the SSH command that you just copied. This connects your instance to your local terminal using SSH.

Step 3: ps -ef | grep nagios
Run the above command on the nagios-host instance. This verifies whether the nagios service is running or not.



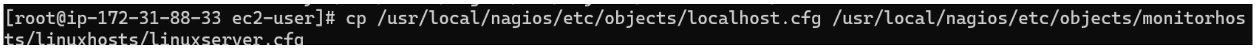
```
[root@ip-172-31-88-33 ec2-user]# ps -ef | grep nagios
nagios     67489       1  0 11:25 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     67490   67489  0 11:25 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios     67491   67489  0 11:25 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios     67492   67489  0 11:25 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios     67493   67489  0 11:25 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios     67494   67489  0 11:25 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root       69007   68853  0 11:51 pts/1    00:00:00 grep --color=auto nagios
[root@ip-172-31-88-33 ec2-user]#
```

Step 4: sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
This makes you the root user and creates two folders with the above paths.



```
[root@ip-172-31-88-33 ec2-user]# sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-88-33 ec2-user]#
```

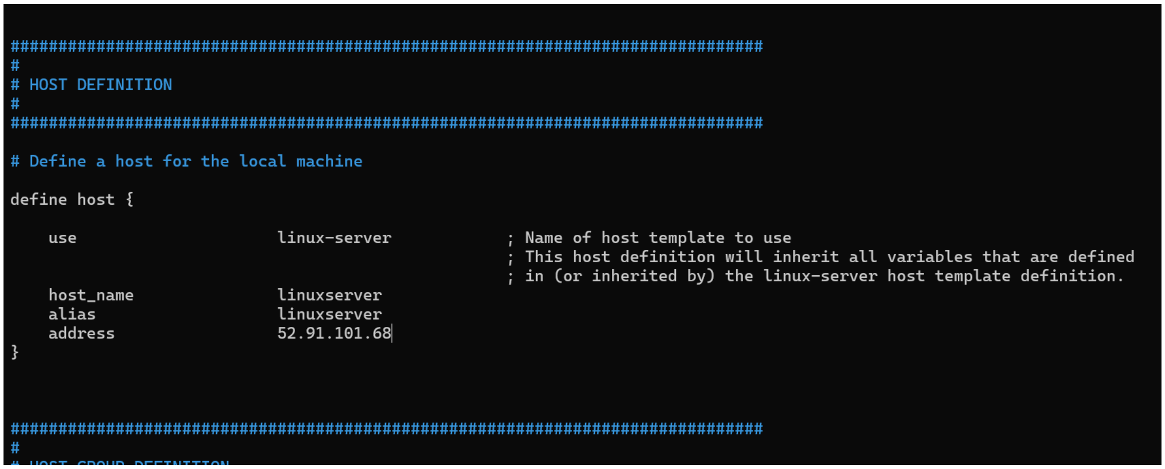
Step 5: We need to create a config file in this folder. So, copy the contents of the existing localhost config to the new file ‘linuxserver.cfg’.
cp /usr/local/nagios/etc/objects/localhost.cfg
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg



```
[root@ip-172-31-88-33 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhos
ts/linuxhosts/linuxserver.cfg
```

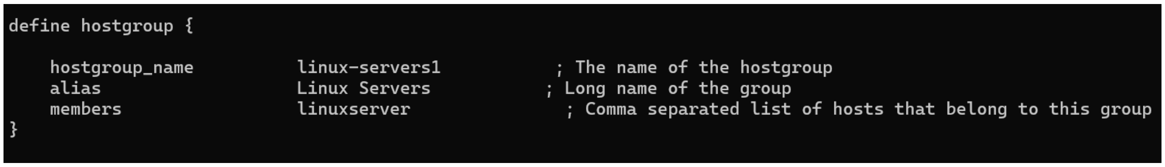
Step 6: We need to make some changes in this config file. Open it using nano editor:-
nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

1. Change hostname and alias from ‘hostname’ to ‘linuxserver’.
2. Change address to the public ip address of the ubuntu-client instance.



```
###############################################################################
#
# HOST DEFINITION
#
###############################################################################

# 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                   linuxserver
    address                 52.91.101.68
}



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

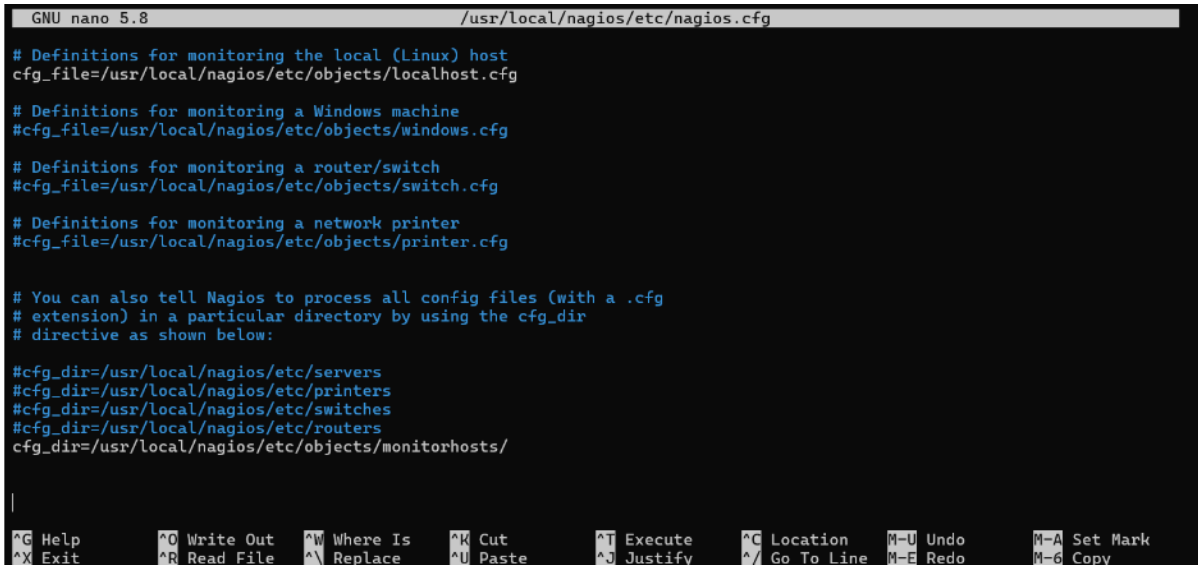
Change hostgroup_name to ‘linux-servers1’.



```
define hostgroup {

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

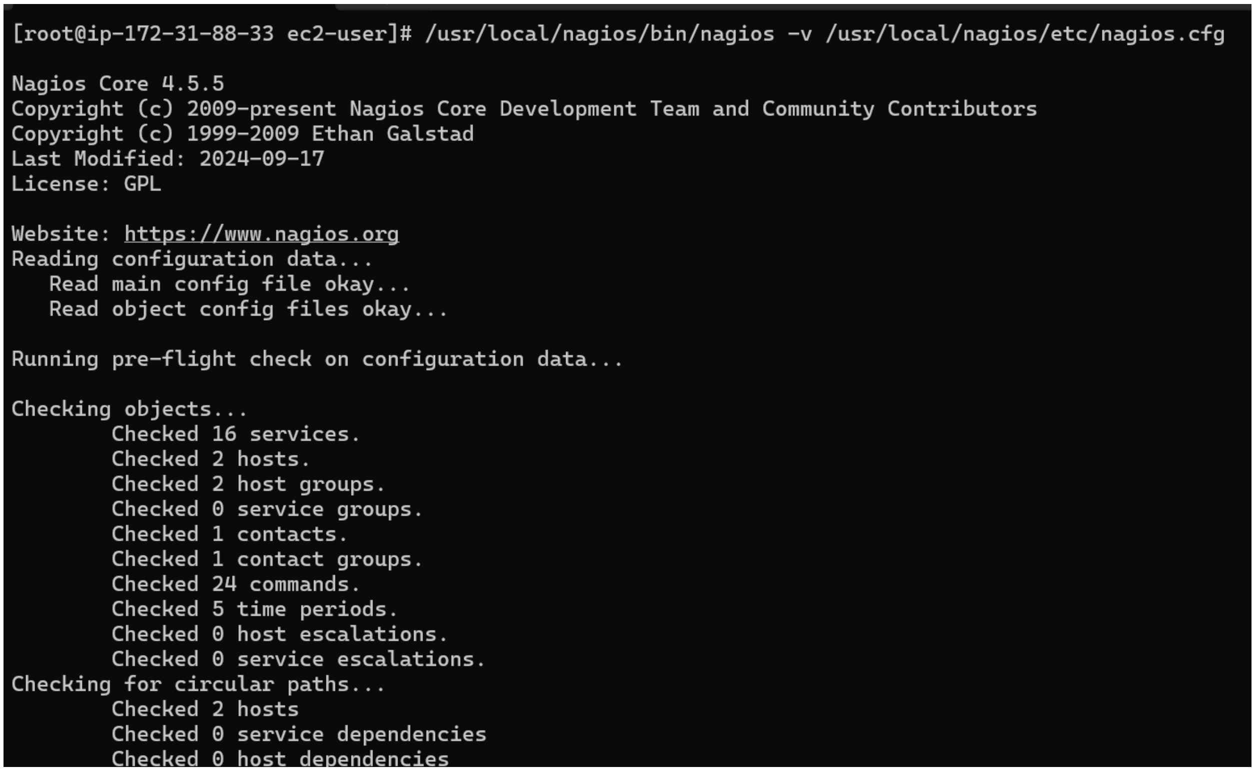
Change all the subsequent occurrences of hostname in the file from ‘localhost’ to linuxserver’.

Step 7: Open the Nagios config file using the following command:
nano /usr/local/nagios/etc/nagios.cfg
Then, add the following line to the config file:
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

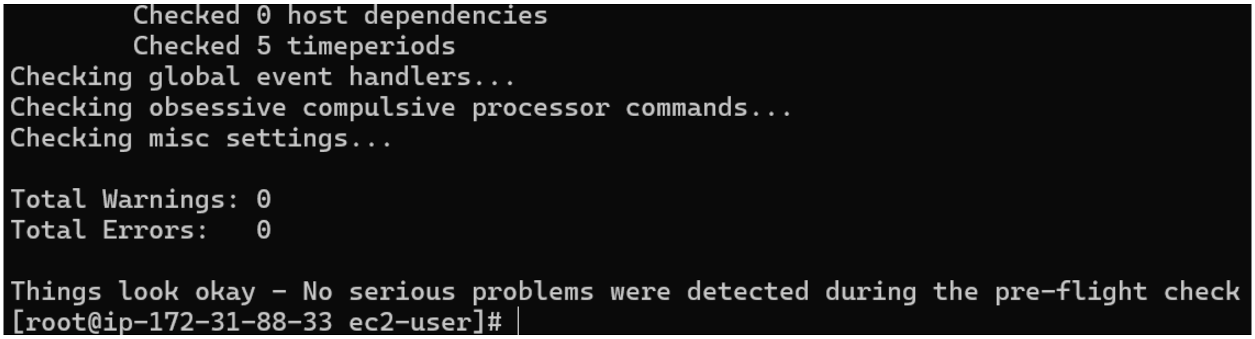


```
  GNU nano 5.8                          /usr/local/nagios/etc/nagios.cfg

# 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

# Definitions for monitoring a router/switch
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg


# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/



^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Step 8: Now we verify the configuration files and check that they contain no errors using the following command:
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

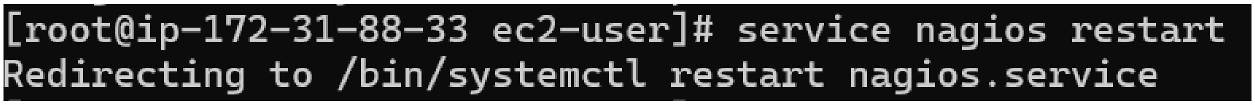


```
[root@ip-172-31-88-33 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
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 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
```



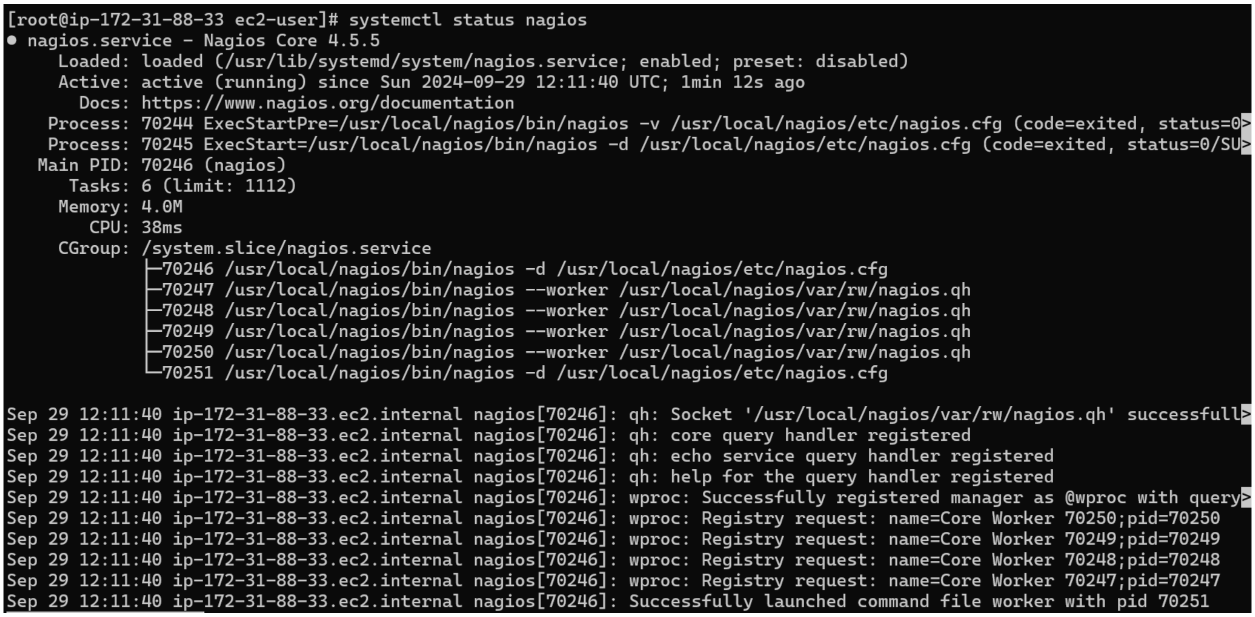
```
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-88-33 ec2-user]# 
```

Step 9: Once the files are verified and it is confirmed that there are no errors, we must restart the server.
service nagios restart



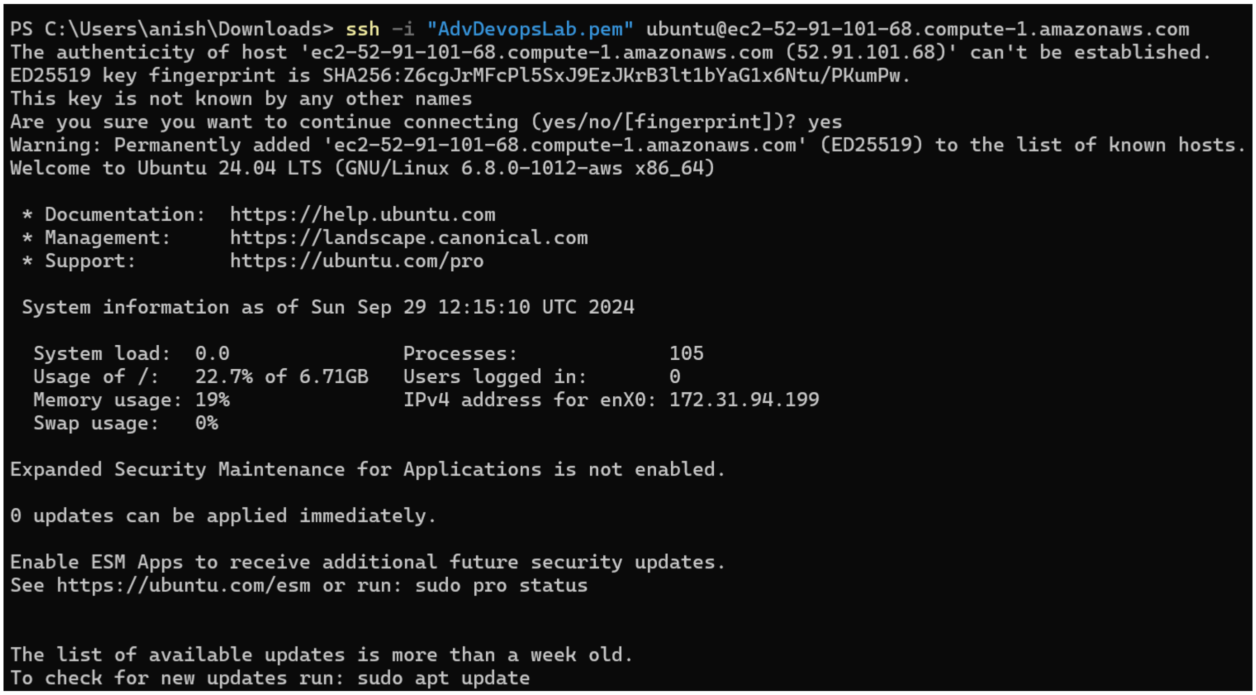
```
[root@ip-172-31-88-33 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

Step 10: systemctl status nagios
Using the above command, we check the status of the nagios server and ensure that it is active (running).



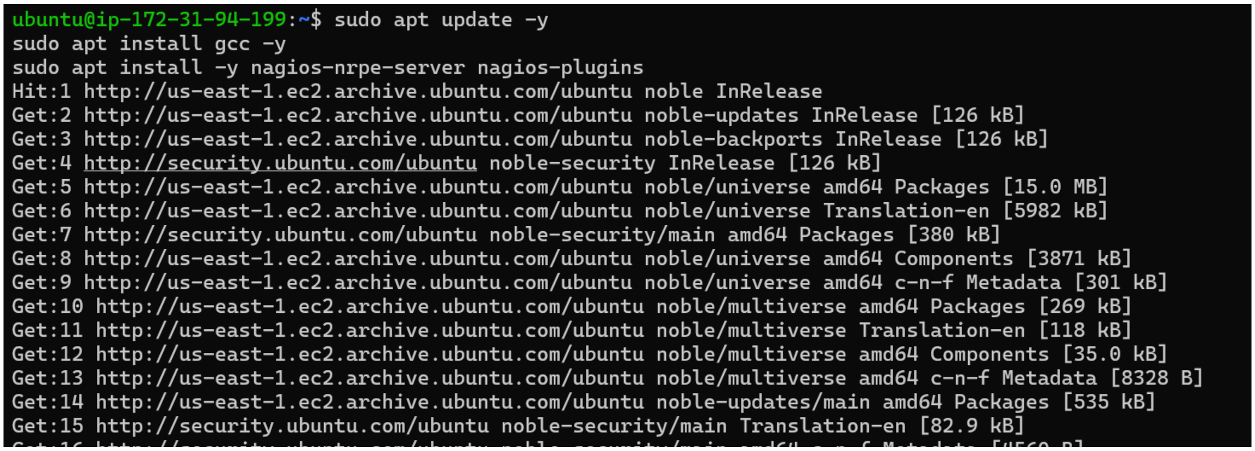
```
[root@ip-172-31-88-33 ec2-user]# systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-09-29 12:11:40 UTC; 1min 12s ago
       Docs: https://www.nagios.org/documentation
    Process: 70244 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0>
    Process: 70245 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU>
   Main PID: 70246 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 4.0M
        CPU: 38ms
     CGroup: /system.slice/nagios.service
             ├─70246 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─70247 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─70248 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─70249 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─70250 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─70251 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfull>
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: qh: core query handler registered
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: qh: echo service query handler registered
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: qh: help for the query handler registered
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: wproc: Successfully registered manager as @wproc with query>
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: wproc: Registry request: name=Core Worker 70250;pid=70250
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: wproc: Registry request: name=Core Worker 70249;pid=70249
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: wproc: Registry request: name=Core Worker 70248;pid=70248
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: wproc: Registry request: name=Core Worker 70247;pid=70247
Sep 29 12:11:40 ip-172-31-88-33.ec2.internal nagios[70246]: Successfully launched command file worker with pid 70251
```

Step 11: Connect your ubuntu-client instance to your local terminal using SSH in the same way as you connected the nagios-host instance to your local terminal using SSH (follow Step 2)

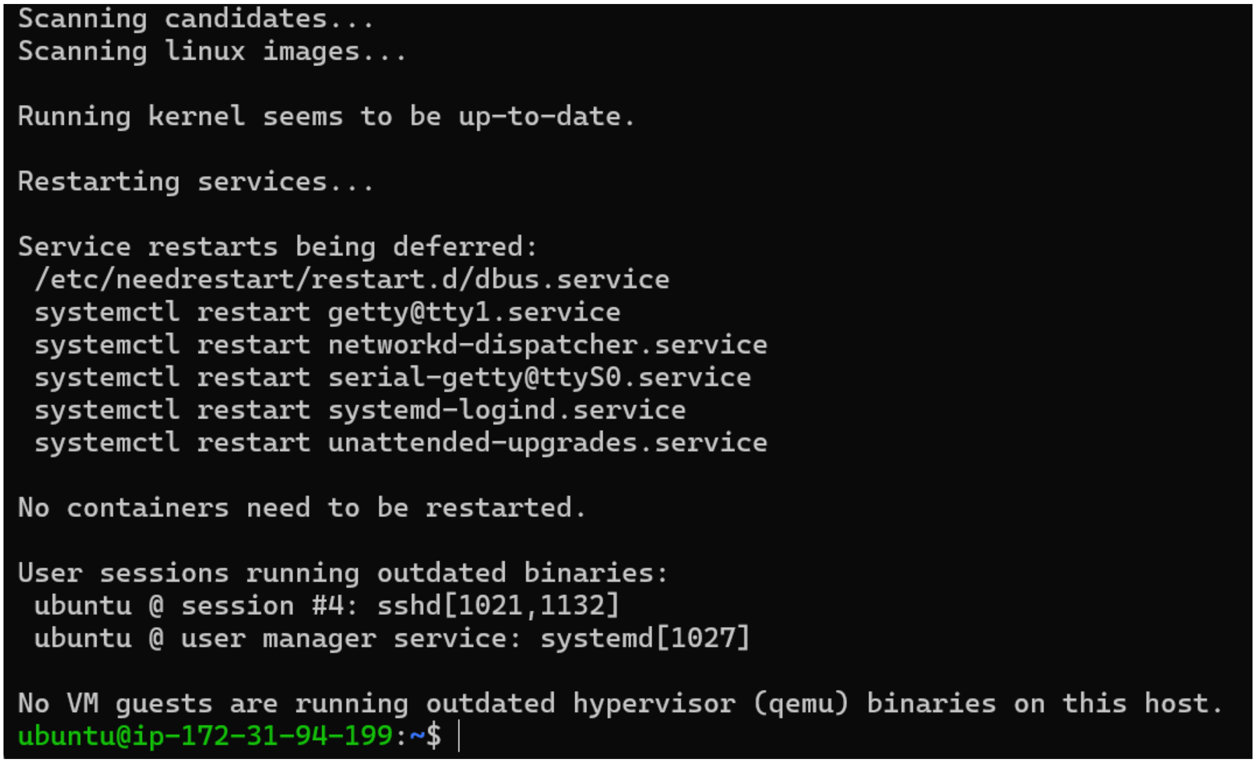


```
PS C:\Users\anish\Downloads> ssh -i "AdvDevopsLab.pem" ubuntu@ec2-52-91-101-68.compute-1.amazonaws.com
The authenticity of host 'ec2-52-91-101-68.compute-1.amazonaws.com (52.91.101.68)' can't be established.
ED25519 key fingerprint is SHA256:Z6cgJrMFcPl5SxJ9EzJKrB3lt1bYaG1x6Ntu/PKumPw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-101-68.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Sun Sep 29 12:15:10 UTC 2024

  System load:  0.0               Processes:             105
  Usage of /:   22.7% of 6.71GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for enX0: 172.31.94.199
  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
```

Step 12: On your ubuntu-client instance, run the following commands:-
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
The above commands check for any new updates and then install gcc, Nagios NRPE server and Nagios plugins.



```
ubuntu@ip-172-31-94-199:~$ 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 [380 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 [535 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
```



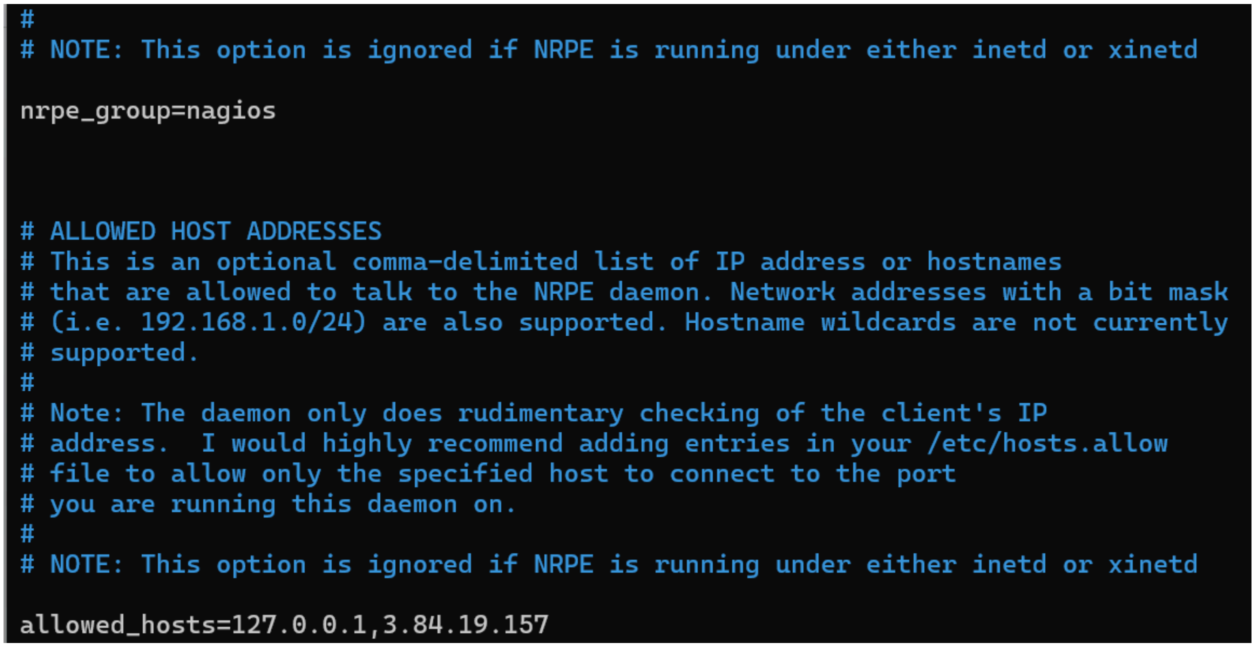
```
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #4: sshd[1021,1132]
 ubuntu @ user manager service: systemd[1027]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-94-199:~$
```

Step 13: Run the following command:

sudo nano /etc/nagios/nrpe.cfg

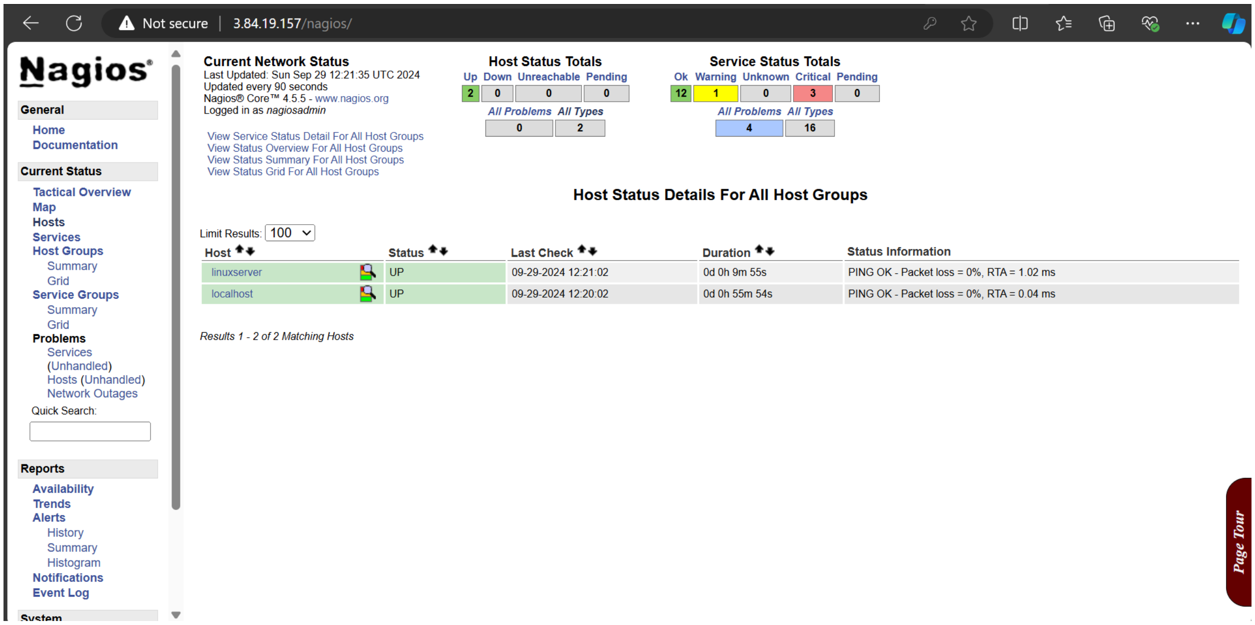
The above command opens the NRPE config file. Here, we need to add the public IP address of our host nagios-host instance to the NRPE configuration file.

Under allowed_hosts, add the nagios-host public IPv4 address.



```
#
# 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=127.0.0.1,3.84.19.157
```

Step 14: Navigate to the Nagios dashboard. Click on 'hosts'. We see that linuxserver has been added as a host.



Current Network Status
Last Updated: Sun Sep 29 12:21:35 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

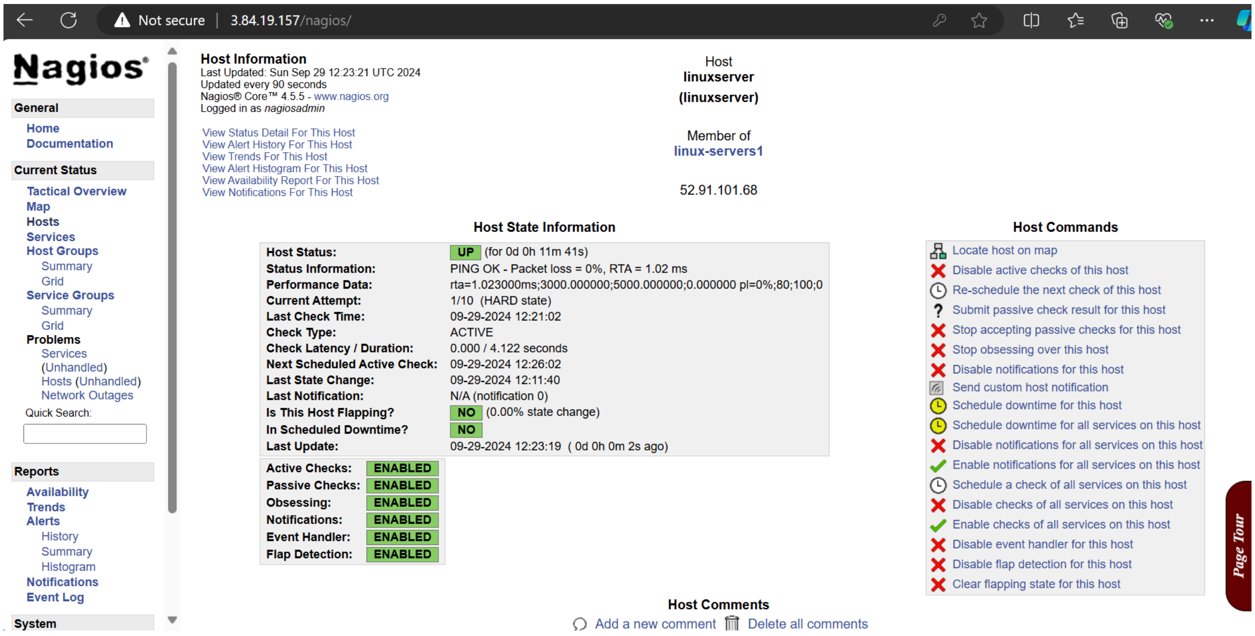
Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-29-2024 12:21:02	0d 0h 9m 55s	PING OK - Packet loss = 0%, RTA = 1.02 ms
localhost	UP	09-29-2024 12:20:02	0d 0h 55m 54s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Results 1 - 2 of 2 Matching Hosts

Click on ‘linuxserver’. Here, we can access all information about the ‘linuxserver’ host.



Host Information
Last Updated: Sun Sep 29 12:23:21 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
linuxserver
(linuxserver)

Member of
linux-servers1

52.91.101.68

Host State Information

Host Status:	UP (for 0d 0h 11m 41s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.02 ms
Performance Data:	rta=1.023000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	09-29-2024 12:21:02
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.122 seconds
Next Scheduled Active Check:	09-29-2024 12:26:02
Last State Change:	09-29-2024 12:11:40
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	09-29-2024 12:23:19 (0d 0h 0m 2s ago)

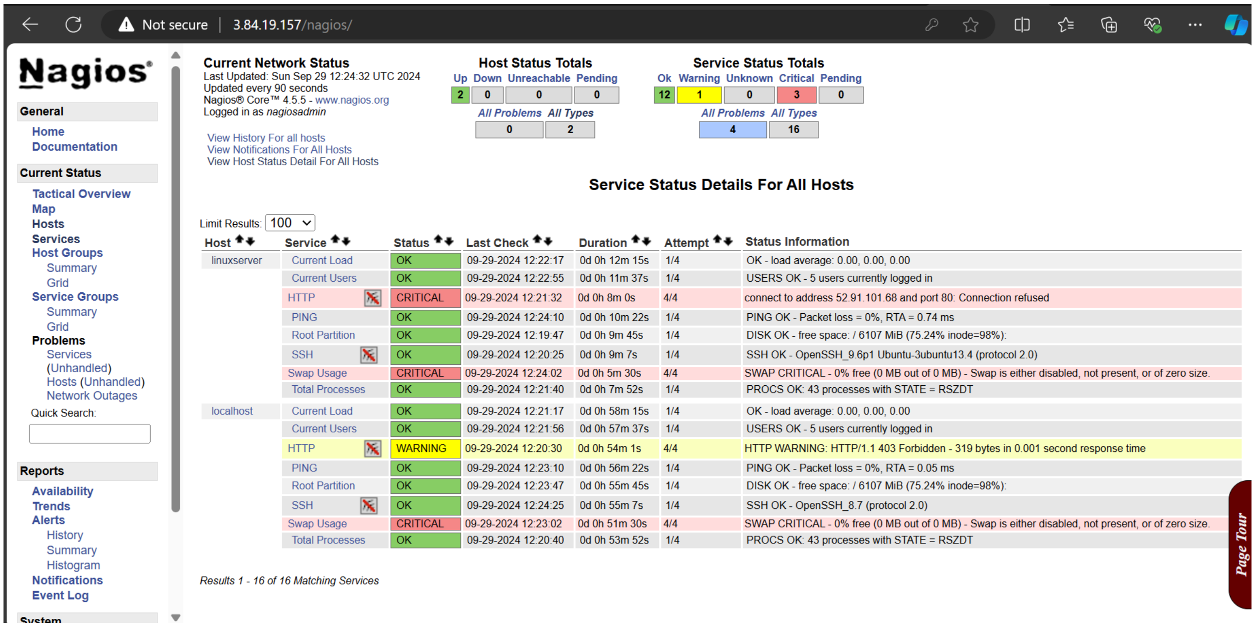
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

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

Click on ‘Services’. Here, we can see all the services that are being monitored by ‘linuxserver’.



Current Network Status
Last Updated: Sun Sep 29 12:24:32 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
12	1	0	3	0

All Problems	*All Types*
4	16

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	09-29-2024 12:22:17	0d 0h 12m 15s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	09-29-2024 12:22:55	0d 0h 11m 37s	1/4	USERS OK - 5 users currently logged in
	HTTP	CRITICAL	09-29-2024 12:21:32	0d 0h 8m 0s	4/4	connect to address 52.91.101.68 and port 80: Connection refused
	PING	OK	09-29-2024 12:24:10	0d 0h 10m 22s	1/4	PING OK - Packet loss = 0%, RTA = 0.74 ms
	Root Partition	OK	09-29-2024 12:19:47	0d 0h 9m 45s	1/4	DISK OK - free space: / 6107 MiB (75.24% inode=98%):
	SSH	OK	09-29-2024 12:20:25	0d 0h 9m 7s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-3ubuntu13.4 (protocol 2.0)
	Swap Usage	CRITICAL	09-29-2024 12:24:02	0d 0h 5m 30s	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	09-29-2024 12:21:40	0d 0h 7m 52s	1/4	PROCS OK: 43 processes with STATE = RSZDT
localhost	Current Load	OK	09-29-2024 12:21:17	0d 0h 58m 15s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	09-29-2024 12:21:56	0d 0h 57m 37s	1/4	USERS OK - 5 users currently logged in
	HTTP	WARNING	09-29-2024 12:20:30	0d 0h 54m 1s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
	PING	OK	09-29-2024 12:23:10	0d 0h 56m 22s	1/4	PING OK - Packet loss = 0%, RTA = 0.05 ms
	Root Partition	OK	09-29-2024 12:23:47	0d 0h 55m 45s	1/4	DISK OK - free space: / 6107 MiB (75.24% inode=98%):
	SSH	OK	09-29-2024 12:24:25	0d 0h 55m 7s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	09-29-2024 12:23:02	0d 0h 51m 30s	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	09-29-2024 12:20:40	0d 0h 53m 52s	1/4	PROCS OK: 43 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

Conclusion: In this experiment, we learned how to perform port, service monitoring, Windows/Linux server monitoring using Nagios. To do so, we needed a nagios-host EC2 Linux instance which was used to host the Nagios server and dashboard. We created an Ubuntu client instance to connect to the host. We set up some configurations on the Linux instance and added the public IP address of the Ubuntu instance in it. We also set up some configurations on the Ubuntu client instance and added the IP address of the Linux server instance in it. Then, we made sure to add the Linux server instance as a ‘allowed host’ for the Ubuntu client instance. After restarting the NRPE server, we can see the ‘linuxserver’ host added.