

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.

Name and tags [Info](#)

Name
ubuntu-client10 [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

Recents Quick Start

Amazon Linux macOS **Ubuntu** Windows Red Hat SUSE Linux [Browse more AMIs](#)

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0866a3c8b686eaeaba

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 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.

[Cancel](#) [Launch instance](#) [Review commands](#)

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) [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
nagios-9key [Create new key pair](#)

Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0381e49e607677b63

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
group9 sg-013cdc8a1aac0de8c [Compare security group rules](#)
VPC: vpc-0381e49e607677b63

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

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0866a3c8b686eaeaba

Virtual server type (instance type)
t2.micro

Firewall (security group)
group9

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.

[Cancel](#) [Launch instance](#) [Review commands](#)

Instances (3) Info								
Find instance by attribute or tag (case-sensitive)				All states ▾				
Instance state = running ✕				Clear filters				
<input type="checkbox"/>	Name ↗ ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS
<input type="checkbox"/>		i-040355adcc131cadd	Running 🔍 🔍	t2.micro	2/2 checks passed 🔍	View alarms +	us-east-1c	ec2-54-226-117-238.compute-1.amazonaws.com
<input type="checkbox"/>	nagios-9	i-0fd2965bf41ae9cd0	Running 🔍 🔍	t2.micro	2/2 checks passed 🔍	View alarms +	us-east-1c	ec2-18-234-72-188.compute-1.amazonaws.com
<input type="checkbox"/>	ubuntu-client10	i-016e919a97365096a	Running 🔍 🔍	t2.micro	Initializing 🔍	View alarms +	us-east-1c	ec2-54-80-53-159.compute-1.amazonaws.com

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.

```
[ec2-user@ip-172-31-35-113 ~]$ ps -ef | grep nagios
nagios    64399      1    0 13:48 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    64401    64399    0 13:48 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    64402    64399    0 13:48 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    64403    64399    0 13:48 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    64404    64399    0 13:48 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    64407    64399    0 13:48 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user  65271    65245    0 14:01 pts/0    00:00:00 grep --color=auto nagio
```

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.

```
[ec2-user@ip-172-31-35-113 ~]$ sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-35-113 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.

```
GNU nano 5.8 /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
#####
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
#
# NOTE: This config file is intended to serve as an *extremely* simple
#       example of how you can create configuration entries to monitor
#       the local (Linux) machine.
#
#####

#####
#
# 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            54.80.53.159
}
}
```

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  ^U Undo      ^M Set Mark
^X Exit      ^R Read File ^N Replace   ^V Paste     ^J Justify   ^_ Go To Line ^E Redo      ^G 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-172-31-35-113 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...
```

```
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 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/SU
   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\ADMIN> cd .\Downloads\
PS C:\Users\ADMIN\Downloads> ssh -i "nagios-9key.pem" ubuntu@ec2-54-80-53-159.compute-1.amazonaws.com
The authenticity of host 'ec2-54-80-53-159.compute-1.amazonaws.com (54.80.53.159)' can't be established.
ED25519 key fingerprint is SHA256:ictbLzPlp2YFXqDXkAH4CzoAWHCOQfhqnoXYFJGZ5q8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-80-53-159.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
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 Mon Oct  7 14:15:43 UTC 2024

System load:  0.0          Processes:      104
Usage of /:   22.8% of 6.71GB Users logged in:  0
Memory usage: 20%         IPv4 address for enX0: 172.31.38.70
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-172-31-38-70:~$ |
```

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-172-31-38-70:~$ 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]
```

```
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-172-31-38-70:~$ |
```

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.

```

GNU nano 7.2 /etc/nagios/nrpe.cfg *
# 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=127.0.0.1,18.234.72.188|

# 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.

```

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

The screenshot shows the Nagios web interface at 18.234.72.188/nagios/. The dashboard includes a sidebar with navigation links, a top section for network status and totals, and a main section for host status details.

Current Network Status
 Last Updated: Mon Oct 7 14:22:35 UTC 2024
 Updated every 50 seconds
 Nagios® Core™ 4.5.5 - www.nagios.org
 Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems: 0, All Types: 2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	1	0

All Problems: 2, All Types: 8

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-07-2024 14:19:04	0d 0h 8m 31s	PING OK - Packet loss = 0%, RTA = 0.94 ms
localhost	UP	10-07-2024 14:17:40	0d 0h 34m 17s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Results 1 - 2 of 2 Matching Hosts

Page Tour

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

The screenshot displays the Nagios web interface for the 'linuxserver' host. The interface is divided into several sections:

- Host Information:** Shows the host name 'linuxserver', member of 'linux-servers1', and IP address '54.80.53.159'.
- Host State Information:** Displays the current status as 'UP', last check time, and performance data.
- Host Commands:** A list of commands that can be executed on the host, such as 'Locate host on map', 'Disable active checks of this host', etc.
- Host Comments:** A section for adding or deleting comments about the host.

The left sidebar contains navigation links for General, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, and System.

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

The screenshot displays the Nagios web interface for the 'Services' section of the 'linuxserver' host. The interface shows a table of services being monitored, including their status, last check time, duration, attempt, and status information.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Users	OK	10-07-2024 14:18:55	0d 0h 35m 2s	1/4	OK - load average: 0.00, 0.00, 0.00
linuxserver	HTTP	WARNING	10-07-2024 14:23:10	0d 0h 30m 47s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
linuxserver	PING	OK	10-07-2024 14:20:48	0d 0h 33m 9s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
linuxserver	Root Partition	OK	10-07-2024 14:21:25	0d 0h 32m 32s	1/4	DISK OK - free space: / 6116 MB (75.36% inode=98%)
linuxserver	SSH	OK	10-07-2024 14:22:03	0d 0h 31m 54s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
linuxserver	Swap Usage	CRITICAL	10-07-2024 14:20:40	0d 0h 28m 17s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
linuxserver	Total Processes	OK	10-07-2024 14:23:18	0d 0h 30m 39s	1/4	PROCS OK: 34 processes with STATE = RSZDT

The left sidebar contains navigation links for General, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, and System.

Conclusion : In this experiment, we explored how to monitor ports, services, and both Windows and Linux servers using Nagios. To achieve this, we launched a Nagios-hosted EC2 Linux instance, which served as the platform for running the Nagios server and dashboard. Additionally, we deployed an Ubuntu client instance that connected to the Nagios host.

We configured the necessary settings on the Linux instance, including adding the Ubuntu client's public IP address. Similarly, we made configuration changes on the Ubuntu client, where we added the IP address of the Nagios-hosted Linux instance. We also ensured that the Linux server instance was permitted as an authorized host on the Ubuntu client. After restarting the NRPE service, we verified that the 'linuxserver' host was successfully added for monitoring.