

Name-Ishan Kiran Joshi Div -D15C Roll No-21 A.Y.-2024-25

Experiment 10

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

Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

Monitoring Using Nagios:

Step 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

```
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: qh: echo service query handler registered
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: qh: help for the query handler registered
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: wproc: Successfully registered manager as @wproc with query handler
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: wproc: Registry request: name=Core Worker 2009;pid=2009
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: wproc: Registry request: name=Core Worker 2008;pid=2008
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: wproc: Registry request: name=Core Worker 2010;pid=2010
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: wproc: Registry request: name=Core Worker 2007;pid=2007
Oct 02 04:31:39 ip-172-31-91-133.ec2.internal nagios[2006]: Successfully launched command file worker with pid 2011
lines 1-27
[ec2-user@ip-172-31-91-133 ~]$
```

You can now proceed if you get the above message/output.

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

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

Recents **Quick Start**

Amazon Linux macOS **Ubuntu** Windows Red Hat SUSE Linux
aws Mac ubuntu Microsoft Red Hat SUSI

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

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

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

For Key pair : Click on create key and make key of type RSA with nagios .pem . Key will be downloaded to your local machine.

Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.

Key pair name - required
nagios [Create new key pair](#)

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

Network [Info](#)
vpc-01c3016e39e818298

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-24 sg-04fb81acb7470e11a X
VPC: vpc-01c3016e39e818298

[Compare security group rules](#)

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)
launch-wizard-24

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

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

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).

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3395154=Ishan_Joshi @ 6870-5842-2407

Key pair name - required

nagios

Create new key pair

▼ Network settings

Info

Edit

Network

Info

vpc-01c3016e39e818298

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-24 sg-04fb81acb7470e11a

VPC: vpc-01c3016e39e818298

Compare security group rules

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-0866a3c8686eaebea

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-24

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

Cancel

Launch instance

Review commands

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .

Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ishan>ssh -i "nagios.pem" ubuntu@ec2-54-86-242-196.compute-1.amazonaws.com
Warning: Identity file nagios.pem not accessible: No such file or directory.
The authenticity of host 'ec2-54-86-242-196.compute-1.amazonaws.com (54.86.242.196)' can't be established.
ED25519 key fingerprint is SHA256:HK1kLDI+I/LPWNFxGW6qxlJp22PfXkiRlnDbrupc820.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-86-242-196.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
ubuntu@ec2-54-86-242-196.compute-1.amazonaws.com: Permission denied (publickey).
```

Successfully connected to the instance

```
aws Services Search [Alt+S] N. Virginia voclabs/user3395154-Ishan_Joshi @ 6870-5842-2407

System information as of Wed Oct 2 04:44:44 UTC 2024

System load: 0.0          Processes:           106
Usage of /:  22.9% of 6.71GB Users logged in:       0
Memory usage: 21%         IPv4 address for enX0: 172.31.45.35
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 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-172-31-45-35:~$
```

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

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

ps -ef | grep nagios

Step 5: Now Become root user and create root directories.

```
[ec2-user@ip-172-31-91-133 ~]$ ps -ef | grep nagios
nagios      2006      1  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      2007    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2008    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2009    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2010    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2011    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    3145    2703  0 04:45 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-133 ~]$
```

sudo su

mkdir /usr/local/nagios/etc/objects/monitorhosts

mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

```
[ec2-user@ip-172-31-91-133 ~]$ sudo su
[root@ip-172-31-91-133 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-91-133 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-91-133 ec2-user]#
```

Step 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**

```
[root@ip-172-31-91-133 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-91-133 ec2-user]#
```

Step 7: Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

Change **hostname** to **linuxserver**.

Change **address** to the public IP of your Linux client. Set

hostgroup_name to **linux-servers1**.

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

```
# 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            54.86.242.196
}
```

GNU nano 5.8 /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg Modified

```
define hostgroup {

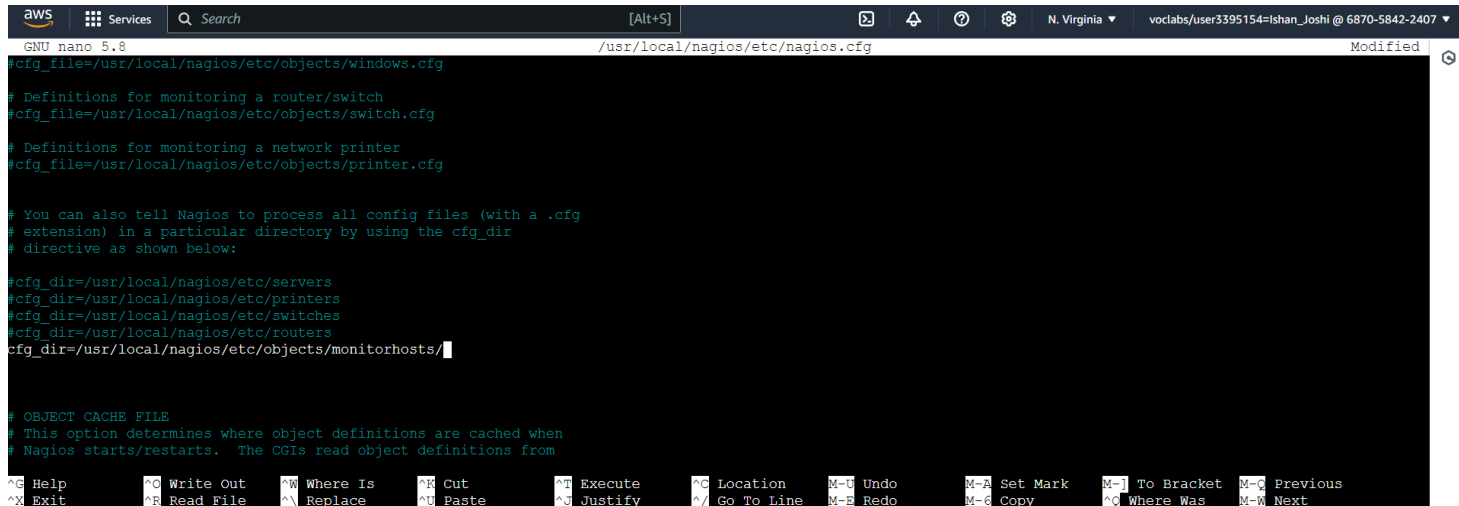
    hostgroup_name     linux-servers1       ; The name of the hostgroup
    alias              Linux Servers        ; Long name of the group
    members            localhost            ; Comma separated list of hosts that belong to this group
}
```

```
#####
#
# SERVICE DEFINITIONS
#
#####
```

Step 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/`

Run the command : `nano /usr/local/nagios/etc/nagios.cfg`



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

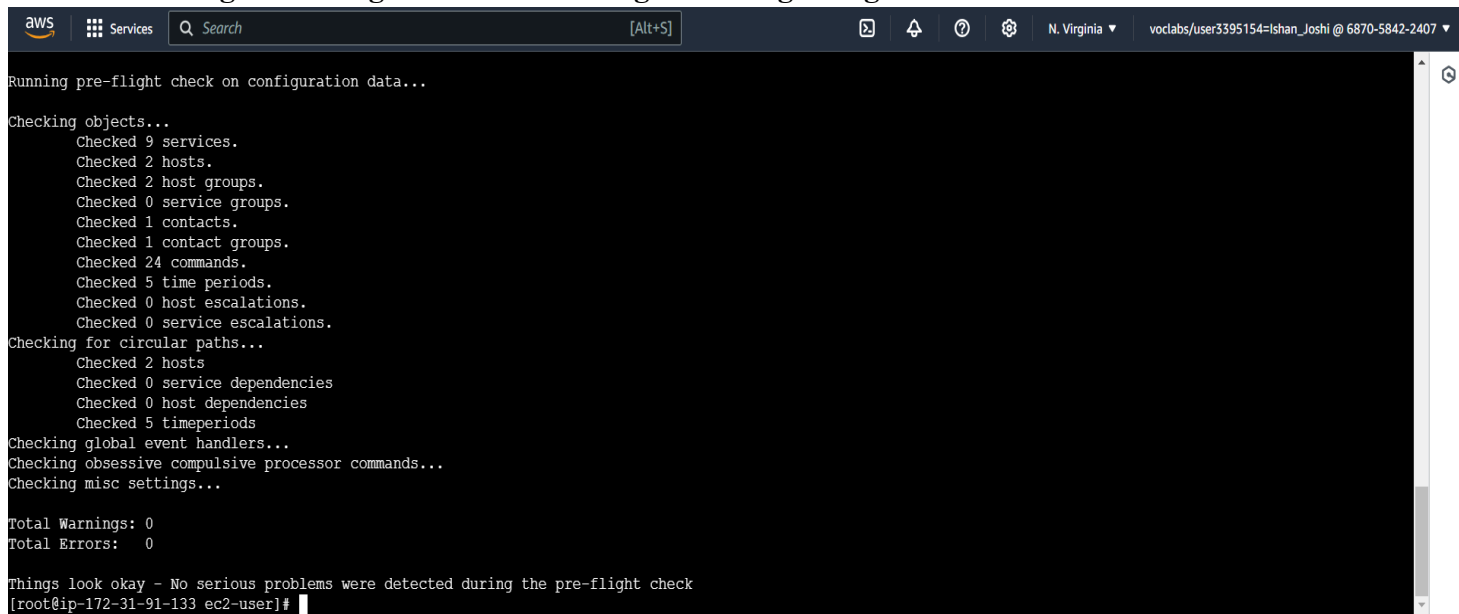
# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#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/

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CGIs read object definitions from
```

Step 9: Now Verify the configuration files by running the following commands.

Run the command : `/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`



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

Checking objects...
  Checked 9 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
[root@ip-172-31-91-133 ec2-user]#
```

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

service nagios restart

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-133 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-91-133 ec2-user]#
```

Step 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

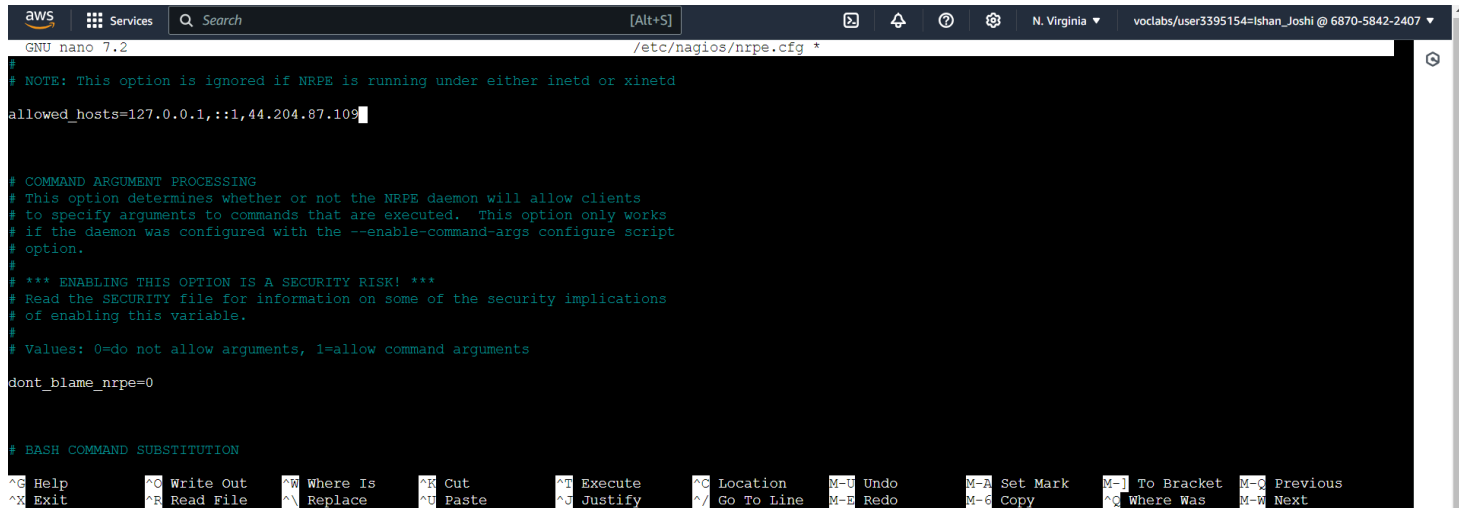
```
aws Services Search [Alt+S] N. Virginia voclabs/user3395154=Ishan_Joshi @ 6870-5842-2407
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-45-35:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbt64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
  libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard mysql-common python3-gpg python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common
  samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
Suggested packages:
  cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib fping postfix | sendmail-bin
  | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients python3-dnspython cifs-utils
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbt64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
  libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind
  samba-common samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
```

```
aws Services Search [Alt+S] N. Virginia voclabs/user3395154=Ishan_Joshi @ 6870-5842-2407
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-172-31-45-35:~$
```


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

sudo nano /etc/nagios/nrpe.cfg



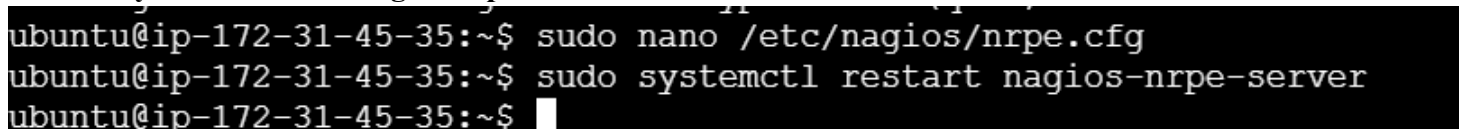
```
aws Services Search [Alt+S] N. Virginia voclabs/user3395154=Ishan_Joshi @ 6870-5842-2407
GNU nano 7.2 /etc/nagios/nrpe.cfg
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,:::1,44.204.87.105

# 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.
#
# Values: 0=do not allow arguments, 1=allow command arguments
dont_blame_nrpe=0

# BASH COMMAND SUBSTITUTION
^G Help      ^O Write Out ^W Where Is  ^R Cut       ^T Execute   ^C Location  ^M Undo      ^M Set Mark  ^M To Bracket ^M Previous
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^_ Go To Line ^B Redo      ^M Copy      ^C Where Was  ^M Next
```

Step 13: Now restart the NRPE server by this command.

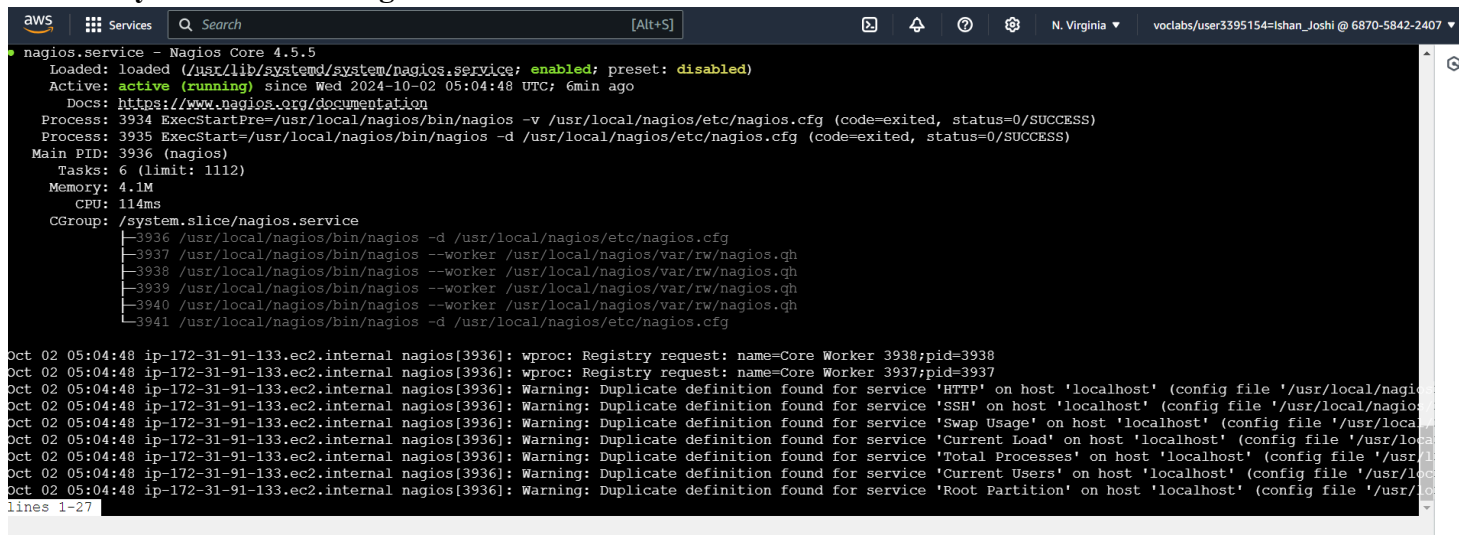
sudo systemctl restart nagios-nrpe-server



```
ubuntu@ip-172-31-45-35:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-45-35:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-45-35:~$
```

Step 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



```
aws Services Search [Alt+S] N. Virginia voclabs/user3395154=Ishan_Joshi @ 6870-5842-2407
* nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Wed 2024-10-02 05:04:48 UTC; 6min ago
     Docs: https://www.nagios.org/documentation
   Process: 3934 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 3935 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Main PID: 3936 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 4.1M
      CPU: 114ms
   CGroup: /system.slice/nagios.service
           └─3936 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           └─3937 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─3938 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─3939 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─3940 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─3941 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: wproc: Registry request: name=Core Worker 3938;pid=3938
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: wproc: Registry request: name=Core Worker 3937;pid=3937
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
Oct 02 05:04:48 ip-172-31-91-133.ec2.internal nagios[3936]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg')
lines 1-27
```

```
sudo systemctl status httpd
sudo systemctl start httpd sudo
systemctl enable httpd
```

```
[root@ip-172-31-91-133 ec2-user]# sudo systemctl status httpd
o 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-172-31-91-133 ec2-user]# sudo systemctl start httpd sudo systemctl enable httpd
Failed to start sudo.service: Unit sudo.service not found.
Failed to start systemctl.service: Unit systemctl.service not found.
Failed to start enable.service: Unit enable.service not found.
[root@ip-172-31-91-133 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-91-133 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-172-31-91-133 ec2-user]#
```

Step 15: Now to check Nagios dashboard go to <http://<Nagios-host ip>/nagios> .

The screenshot shows the Nagios Core 4.5.5 dashboard. The sidebar on the left contains navigation links for General, Current Status, Reports, and System. The main content area features the Nagios Core logo, version 4.5.5, and a status message 'Daemon running with PID 66172'. There are sections for 'Get Started', 'Quick Links', 'Latest News', and 'Don't Miss...'. The bottom of the dashboard includes copyright information and a license notice.

Now Click on Hosts from left side panel

The screenshot shows the Nagios Core 4.5.5 dashboard with the Hosts section selected. The sidebar on the left is visible. The main content area displays 'Current Network Status' and 'Host Status Totals'. Below these, there is a table titled 'Host Status Details For All Host Groups' showing details for 'linuxserver' and 'localhost'.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-02-2024 05:14:48	0d 0h 10m 37s	PING OK - Packet loss = 0%, RTA = 1.66 ms
localhost	UP	10-02-2024 05:13:31	0d 19h 8m 53s	PING OK - Packet loss = 0%, RTA = 0.03 ms

We can see our linuxserver now click on it we can see the host information.

Not secure 44.204.87.109/nagios/

Nagios®

General

Home

Documentation

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary

Grid

Service Groups

Summary

Grid

Problems

Services

(Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Availability

Trends

Alerts

History

Summary

Histogram

Notifications

Event Log

System

Comments

Downtime

Process Info

Performance Info

Scheduling Queue

Configuration

Host Information

Last Updated: Wed Oct 2 05:15:58 UTC 2024

Updated every 90 seconds

Nagios® Core™ 4.5.5 - www.nagios.org

Logged in as nagiosadmin

Host

localhost

(linuxserver)

Member of

No hostgroups

54.86.242.196

Host State Information

Host Status: **UP** (for 0d 0h 11m 10s)

Status Information: PING OK - Packet loss = 0%, RTA = 1.66 ms

Performance Data: rta=1.663000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0

Current Attempt: 1/10 (HARD state)

Last Check Time: 10-02-2024 05:14:48

Check Type: ACTIVE

Check Latency / Duration: 0.001 / 4.012 seconds

Next Scheduled Active Check: 10-02-2024 05:19:48

Last State Change: 10-02-2024 05:04:48

Last Notification: N/A (notification 0)

Is This Host Flapping? **NO** (0.00% state change)

In Scheduled Downtime? **NO**

Last Update: 10-02-2024 05:15:57 (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

Not secure 44.204.87.109/nagios/

Nagios®

General

Home

Documentation

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary

Grid

Service Groups

Summary

Grid

Problems

Services

(Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Availability

Trends

Alerts

History

Summary

Histogram

Notifications

Event Log

System

Comments

Downtime

Process Info

Performance Info

Scheduling Queue

Configuration

Current Network Status

Last Updated: Wed Oct 2 05:16:31 UTC 2024

Updated every 90 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 All Types

0 2

Service Status Totals

Ok Warning Unknown Critical Pending

7 1 0 1 0

All Problems All Types

2 9

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	PING	OK	10-02-2024 05:15:54	0d 0h 10m 37s	1/4	PING OK - Packet loss = 0%, RTA = 1.75 ms
localhost	Current Load	OK	10-02-2024 05:12:16	0d 19h 9m 23s	1/4	OK - load average: 0.00, 0.00, 0.00
localhost	Current Users	OK	10-02-2024 05:12:54	0d 19h 8m 45s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	WARNING	10-02-2024 05:13:31	0d 0h 3m 0s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
localhost	PING	OK	10-02-2024 05:14:09	0d 19h 7m 30s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
localhost	Root Partition	OK	10-02-2024 05:14:46	0d 19h 6m 53s	1/4	DISK OK - free space: / 6114 MiB (75.33% inode=98%)
localhost	SSH	OK	10-02-2024 05:15:24	0d 19h 6m 15s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
localhost	Swap Usage	CRITICAL	10-02-2024 05:16:01	0d 19h 2m 38s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
localhost	Total Processes	OK	10-02-2024 05:11:39	0d 19h 5m 0s	1/4	PROCS OK: 31 processes with STATE = RSZDT

Results 1 - 9 of 9 Matching Services

Conclusion: In conclusion, this experiment concentrated on monitoring network ports, services, and a Linux server with Nagios. By following a systematic approach, we effectively configured Nagios to oversee vital network services on the Linux server. With both the Nagios host and client set up, we gained the ability to track system performance, ensure the availability of services, and monitor crucial metrics such as CPU and memory usage. This comprehensive monitoring not only enhances our operational awareness but also empowers us to respond swiftly to any emerging issues. Ultimately, this setup lays the groundwork for maintaining a robust and reliable IT environment.