

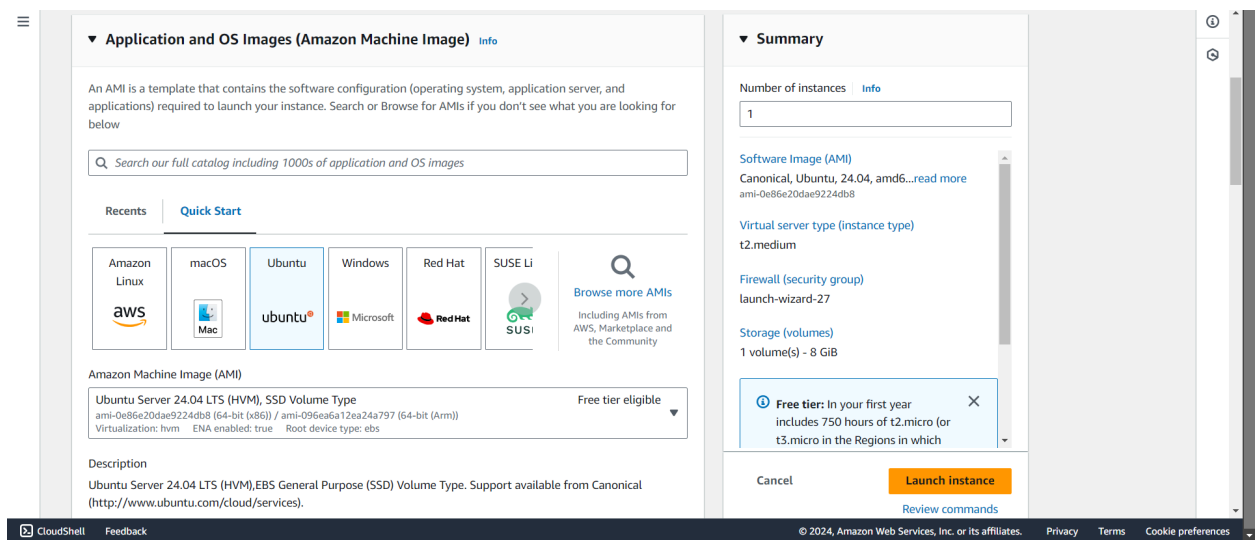
## Experiment No:10

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

**PREREQUISITES :** We should have an Amazon Linux instance with nagios already set up.

### Step 1: Set up ubuntu instance

- 1) Log in to your AWS account. Look for EC2 in the services menu. Open the interface and select Create Instance.



*Select The OS Image as Ubuntu.*

- 2) Ensure that you choose the same private key you created for the Amazon Linux instance. Additionally, select the same security group that you configured for the Linux instance.

The image displays two screenshots of the AWS Management Console during the instance launch process.

**Top Screenshot: Instance type and Key pair selection**

- Instance type:** A dropdown menu is set to `t2.medium`. To the right, there are links for [Info](#), [Get advice](#), [All generations](#), and [Compare instance types](#). A note states: "Additional costs apply for AMIs with pre-installed software".
- Key pair (login):** A section explaining that a key pair is used for secure connection. A dropdown for "Key pair name - required" is set to `Lab9_41`. A link [Create new key pair](#) is available.
- Network settings:** A section with an [Info](#) link and an [Edit](#) button.
- Summary:** Shows "Number of instances" as 1. The "Software Image (AMI)" is `Canonical, Ubuntu, 24.04, amd64...`. The "Virtual server type (instance type)" is `t2.medium`. The "Firewall (security group)" is `launch-wizard-27`. The "Storage (volumes)" is `1 volume(s) - 8 GiB`. A "Free tier" notification indicates 750 hours of `t2.micro` or `t3.micro` are included in the first year. Buttons for [Cancel](#), [Launch instance](#), and [Review commands](#) are at the bottom.

**Bottom Screenshot: Network settings and Summary**

- Network settings:** Shows "Network" as `vpc-04fadb026daa5d28` and "Subnet" as "No preference (Default subnet in any availability zone)". "Auto-assign public IP" is set to "Enable". A note says "Additional charges apply when outside of free tier allowance". Under "Firewall (security groups)", "Create security group" is unselected and "Select existing security group" is selected. A dropdown for "Common security groups" shows `launch-wizard-27 sg-0608ac064fada5668`. A link [Compare security group rules](#) is present.
- Summary:** Similar to the top screenshot, showing 1 instance, Ubuntu AMI, `t2.medium` type, `launch-wizard-27` firewall, and 8 GiB storage. The [Launch instance](#) button is highlighted in orange.

Instance is:

The screenshot shows the instance details in the AWS console. The instance name is `nagios_client_41...`, its ID is `i-0d74a72c6a0429781`, and its state is **Running**. The instance type is `t2.medium`. There are icons for Connect, SSH client, and other actions.

- 3) Now return to the instances screen. Click on the instance ID of your instance, then select Connect. Click on SSH client and copy the example command. Next, we need to connect our local OS terminal to the instance using SSH. To do this, open the terminal where the private key file (.pem) is stored. Paste the copied SSH command and execute it.

EC2 > Instances > i-0d74a72c6a0429781 > Connect to instance

## Connect to instance [Info](#)

Connect to your instance i-0d74a72c6a0429781 (nagios\_client\_41Lab10) using any of these options

EC2 Instance Connect

Session Manager

**SSH client**

EC2 serial console

Instance ID  
i-0d74a72c6a0429781 (nagios\_client\_41Lab10)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Lab9\_41.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 "Lab9\_41.pem"
4. Connect to your instance using its Public DNS:  
ec2-3-86-39-170.compute-1.amazonaws.com

Example:  
ssh -i "Lab9\_41.pem" ubuntu@ec2-3-86-39-170.compute-1.amazonaws.com

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

*Copy the example command*

```
ubuntu@ip-172-31-86-92: ~  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\praja\OneDrive\Desktop\Newfolder> ssh -i "Lab9_41.pem" ubuntu@ec2-3-86-39-170.compute-1.amazonaws.com  
The authenticity of host 'ec2-3-86-39-170.compute-1.amazonaws.com (3.86.39.170)' can't be established.  
ED25519 key fingerprint is SHA256:JPN0h3iHhSxolnxMooo98IB3xhjC8bD9+I1NuUgbyF4.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-3-86-39-170.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 Sat Sep 28 11:36:50 UTC 2024  
  
System load:  0.0          Processes:            112  
Usage of /:   22.7% of 6.71GB Users logged in:      0  
Memory usage: 5%          IPv4 address for enX0: 172.31.86.92  
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
```

*Successfully connected the instance via SSH*

**Step 2:** On Nagios Host machine (Linux) execute the following which we have already created as a prerequisites:

- 1) We need to verify whether the nagios service is running or not. For that, run this command : **ps -ef | grep nagios**

```
[ec2-user@ip-172-31-84-149 ~]$ ps -ef | grep nagios
nagios 67665      1  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 67666    67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 67667    67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 67668    67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 67669    67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 67670    67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 70887    3276  0 10:48 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-84-149 ~]$
```

- 2) Next, switch to the root user and create a directory at the path '`/usr/local/nagios/etc/objects/monitorhosts/linuxhosts`'.

**sudo su**

**mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts**

```
[ec2-user@ip-172-31-84-149 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-84-149 ec2-user]#
```

- 3) We need to create a configuration file in this directory. To do this, copy the contents of the existing localhost configuration into the new file named '`linuxserver.cfg`'.

**cp /usr/local/nagios/etc/objects/localhost.cfg**

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

```
[root@ip-172-31-84-149 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg': No such file or directory
```

So make the second directory again and run the cp command

```
[root@ip-172-31-84-149 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-84-149 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-84-149 ec2-user]#
```

We need to make some changes in this config file. Open it using a nano editor.

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

```
[root@ip-172-31-84-149 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

Change **hostname** and **alias** to **linuxserver** .Change address to **public ip address of client instance** (Ubuntu instance)

```
# Define a host for the local machine

define host {

    use                linux-server

    host_name          linuxserver
    alias              linuxserver
    address             3.86.39.170
}
```

Change **hostgroup\_name** to **linux-servers1**

```
# 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
}

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

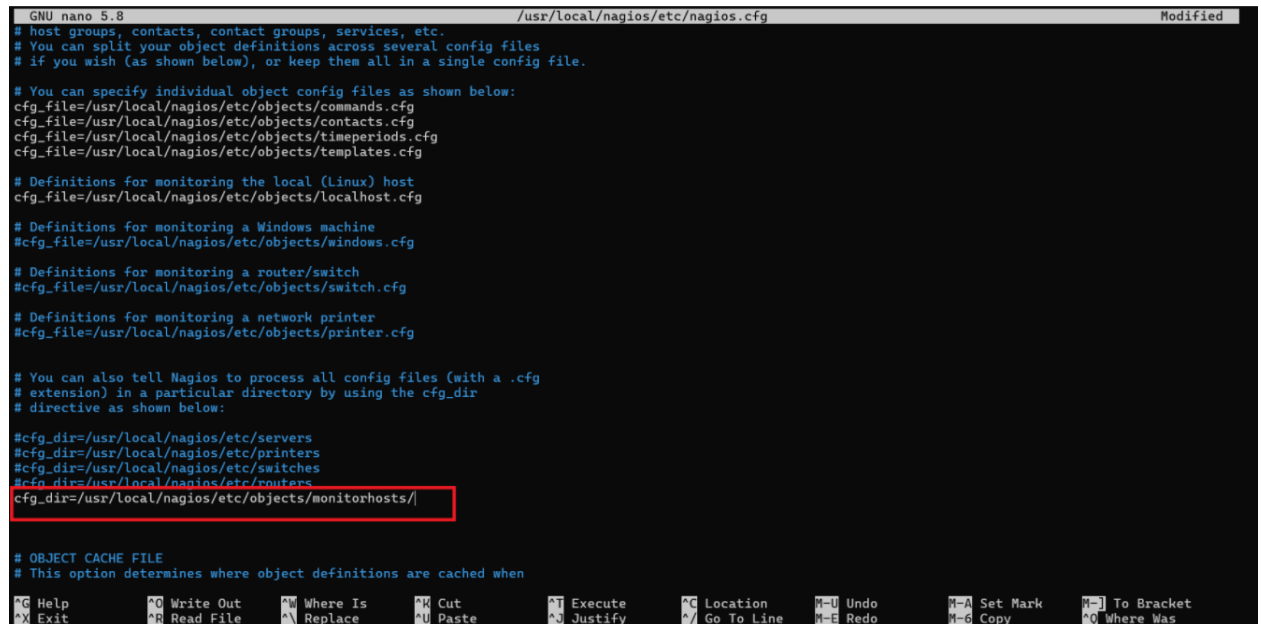
Change the **occurrences of hostname** further in the document from **localhost** to **linuxserver**

Now, we need to edit the nagios configuration file to add this directory. Run this command

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

```
[root@ip-172-31-84-149 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

and add the following line `cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/`



```
GNU nano 5.8 /usr/local/nagios/etc/nagios.cfg Modified
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg

# 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/

# OBJECT CACHE FILE
# This option determines where object definitions are cached when

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

Now we verify the configuration files. `/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`

```
[root@ip-172-31-84-149 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.
```

```

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
[root@ip-172-31-84-149 ec2-user]# |

```

Once the files are verified, we need to restart the server: **service nagios restart**

```

Things look okay - No serious problems were detected during
[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-84-149 ec2-user]# |

```

```

[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-84-149 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 Sat 2024-09-28 11:30:31 UTC; 3min 57s ago
     Docs: https://www.nagios.org/documentation
   Process: 73417 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 73418 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 73419 (nagios)
      Tasks: 6 (limit: 4658)
     Memory: 4.2M
        CPU: 113ms
    CGroup: /system.slice/nagios.service
            └─73419 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
               └─73420 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  └─73421 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─73422 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                        └─73423 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                           └─73425 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73423;pid=73423
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73421;pid=73421
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73420;pid=73420
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73422;pid=73422
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: Successfully launched command file worker with pid 73425
Sep 28 11:32:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: Linuxserver;HTTP;CRITICAL;SOFT;1;connect to address 3.86.39.170 and port 80: Co
Sep 28 11:33:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: Linuxserver;HTTP;CRITICAL;SOFT;2;connect to address 3.86.39.170 and port 80: Co
Sep 28 11:34:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: Linuxserver;HTTP;CRITICAL;SOFT;3;connect to address 3.86.39.170 and port 80: Co
lines 1-28/28 (END)

```

**Step 3:** Execute the following on Nagios Client machine (Ubuntu)

- 1) First, check for any available updates, and then proceed to install gcc, the Nagios NRPE server, and Nagios plugins.

```
sudo apt update -y
```

```
sudo apt install gcc -y
```

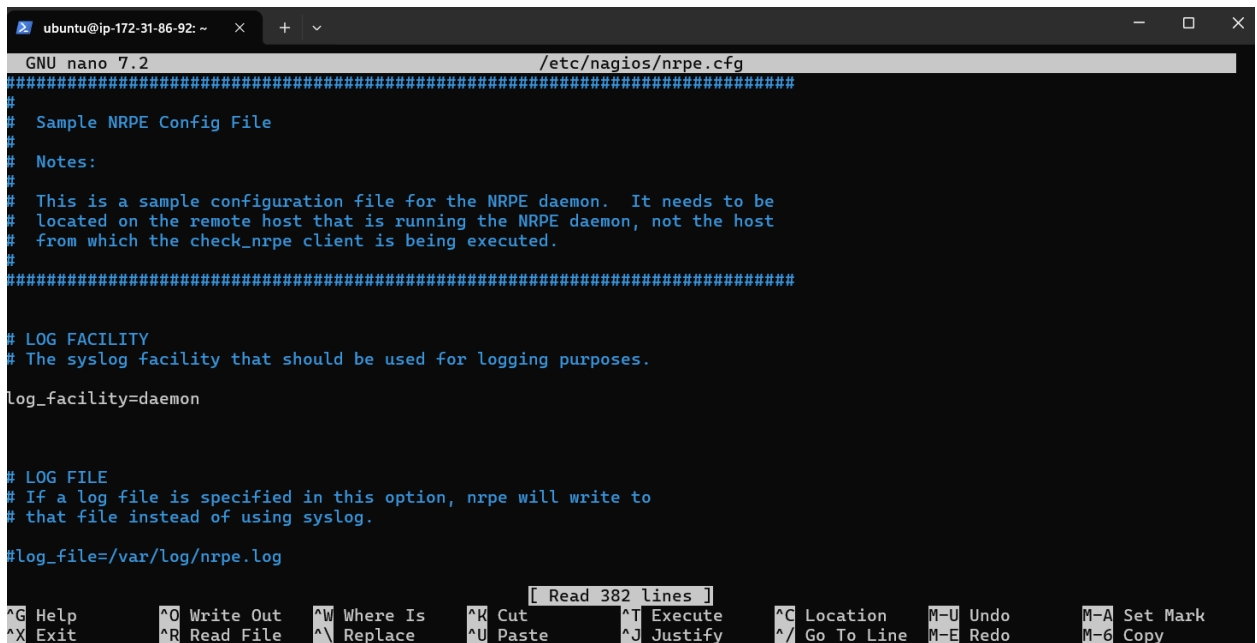
```
sudo apt install -y nagios-nrpe-server nagios-plugins
```

```
ubuntu@ip-172-31-86-92: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-86-92:~$ sudo apt update -y  
sudo apt install gcc -y  
sudo apt install -y nagios-nrpe-server nagios-plugins  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]  
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]  
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]  
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]  
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]  
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]  
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]  
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]  
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]  
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]  
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]  
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.0 kB]
```

```
ubuntu@ip-172-31-86-92: ~  
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 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 #6: sshd[1071,1184]  
ubuntu @ user manager service: systemd[1077]  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-86-92:~$
```

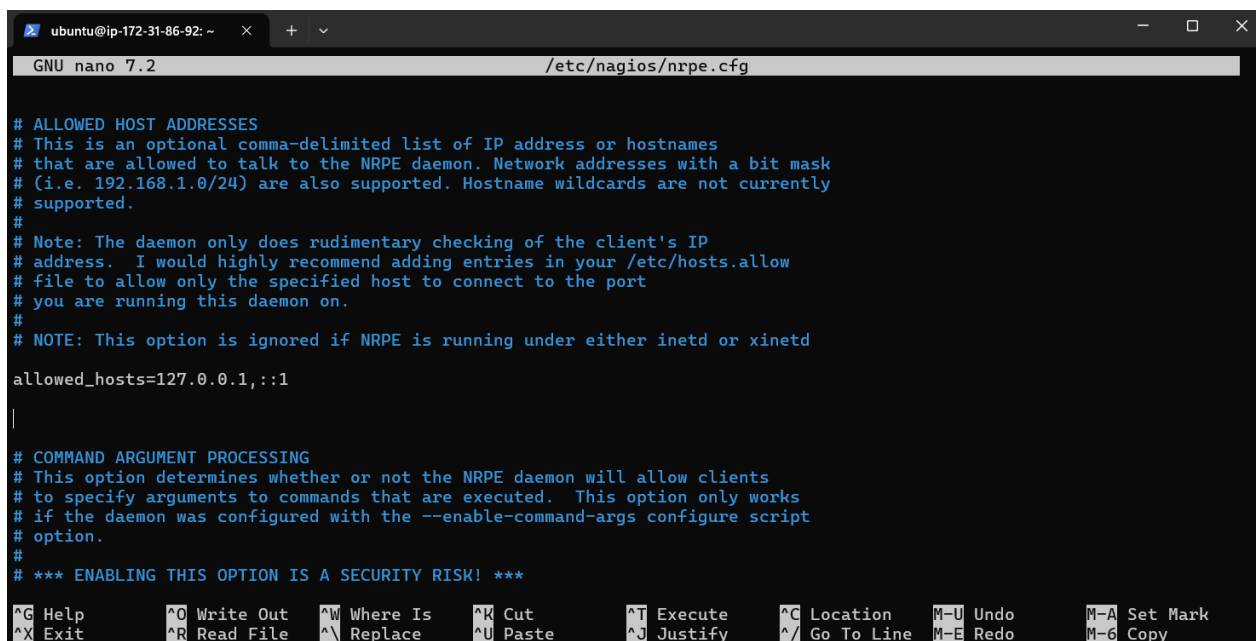


- 2) We need to include the public IP address of our Nagios host machine (Linux) in the NRPE configuration file. **sudo nano /etc/nagios/nrpe.cfg**



```
ubuntu@ip-172-31-86-92: ~  
GNU nano 7.2 /etc/nagios/nrpe.cfg  
#####  
# Sample NRPE Config File  
#  
# Notes:  
#  
# This is a sample configuration file for the NRPE daemon. It needs to be  
# located on the remote host that is running the NRPE daemon, not the host  
# from which the check_nrpe client is being executed.  
#  
#####  
# LOG FACILITY  
# The syslog facility that should be used for logging purposes.  
log_facility=daemon  
#  
# LOG FILE  
# If a log file is specified in this option, nrpe will write to  
# that file instead of using syslog.  
log_file=/var/log/nrpe.log  
[ Read 382 lines ]  
^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-G Copy
```

*Under allowed\_hosts, add the nagios host ip address (public)*



```
ubuntu@ip-172-31-86-92: ~  
GNU nano 7.2 /etc/nagios/nrpe.cfg  
# 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  
|  
# 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! ***  
^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-G Copy
```

**Step 4:** Check the Nagios Dashboard. Go to Nagios dashboard, click on hosts. Here, we can see that the linuxserver is also added as a host.

The screenshot shows the Nagios web interface. The top navigation bar includes links for General, Current Status, Reports, and System. The main content area displays the 'Current Network Status' and 'Host Status Totals'. The 'Host Status Totals' table shows the following data:

Up	Down	Unreachable	Pending
2	0	0	0

The 'Service Status Totals' table shows the following data:

Ok	Warning	Unknown	Critical	Pending
8	1	0	2	5

The 'Host Status Details For All Host Groups' table shows the following data:

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 04:42:16	04 0h 2m 35s	PING OK - Packet loss = 0%, RTA = 1.15 ms
localhost	UP	09-28-2024 04:38:21	04 0h 24m 6s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Click on linuxserver. we can check all the information about linuxserver host.

The screenshot shows the Nagios web interface with the 'Host Information' for 'linuxserver' selected. The 'Host Information' section displays the following data:

Host Information
Last Updated: Sat Sep 28 04:43:37 UTC 2024
Updated every 50 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

The 'Host State Information' section displays the following data:

Host State Information
Host Status: UP (for 0d 0h 3m 13s)
Status Information: PING OK - Packet loss = 0%, RTA = 1.15 ms
Performance Data: rta=1.151000ms,3000.000000,5000.000000,0.000000 pl=0%,80,100,0
Current Attempt: 1/10 (HARD state)
Last Check Time: 09-28-2024 04:42:16
Check Type: ACTIVE
Check Latency / Duration: 0.000 / 4.033 seconds
Next Scheduled Active Check: 09-28-2024 04:47:16
Last State Change: 09-28-2024 04:40:24
Last Notification: N/A (notification 0)
Is This Host Flapping? NO (0.00% state change)
In Scheduled Downtime? NO
Last Update: 09-28-2024 04:43:33 ( 0d 0h 0m 4s ago)

The 'Host Commands' section displays the following data:

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

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

The screenshot displays the Nagios web interface at the URL 54.210.81.106/nagios/. The interface includes a sidebar with navigation links such as General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, System, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

The main content area shows the 'Service Status Details For All Hosts' table. The table has columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The status of services is color-coded: OK (green), WARNING (yellow), CRITICAL (red), and PENDING (grey).

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	09-28-2024 04:41:01	0d 0h 3m 46s+	1/4	OK - load average: 0.00, 0.01, 0.00
	Current Users	OK	09-28-2024 04:41:39	0d 0h 3m 46s+	1/4	USERS OK - 2 users currently logged in
	HTTP	CRITICAL	09-28-2024 04:43:16	0d 0h 1m 54s	2/4	connect to address 107.22.163.120 and port 80: Connection refused
	PING	OK	09-28-2024 04:42:54	0d 0h 3m 46s+	1/4	PING OK - Packet loss = 0%, RTA = 1.11 ms
	Root Partition	OK	09-28-2024 04:43:31	0d 0h 3m 46s+	1/4	DISK OK - free space / 6116 MB (75.36% inode=98%)
	SSH	PENDING	N/A	0d 0h 3m 46s+	1/4	Service check scheduled for Sat Sep 28 04:44:09 UTC 2024
	Swap Usage	PENDING	N/A	0d 0h 3m 46s+	1/4	Service check scheduled for Sat Sep 28 04:44:46 UTC 2024
	Total Processes	PENDING	N/A	0d 0h 3m 46s+	1/4	Service check scheduled for Sat Sep 28 04:45:24 UTC 2024
	localhost	OK	09-28-2024 04:39:36	0d 0h 24m 34s	1/4	OK - load average: 0.00, 0.02, 0.00
	Current Users	OK	09-28-2024 04:40:14	0d 0h 23m 56s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	WARNING	09-28-2024 04:43:51	0d 0h 20m 19s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
	PING	OK	09-28-2024 04:41:29	0d 0h 22m 41s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	09-28-2024 04:42:06	0d 0h 22m 4s	1/4	DISK OK - free space / 6116 MB (75.36% inode=98%)
	SSH	OK	09-28-2024 04:42:44	0d 0h 21m 26s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	09-28-2024 04:41:21	0d 0h 17m 49s	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	09-28-2024 04:43:59	0d 0h 20m 11s	1/4	PROCS OK: 37 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

## CONCLUSION:

In this experiment, we learned to conduct port service monitoring and server monitoring using Nagios. To do this, we require a Linux instance to host the Nagios dashboard and a separate Ubuntu instance linked as a second host. We need to configure the Linux instance and include the IP address of the Ubuntu instance. Subsequently, we must replicate the initial setup from the Linux instance on the Ubuntu instance by adding the IP address of the Linux instance to the list of allowed hosts. After restarting the NRPE server, we should see the 'linuxserver' host listed.

**Name: Aditya Dubey**

**Div: D15C**

**Roll No: 10**