

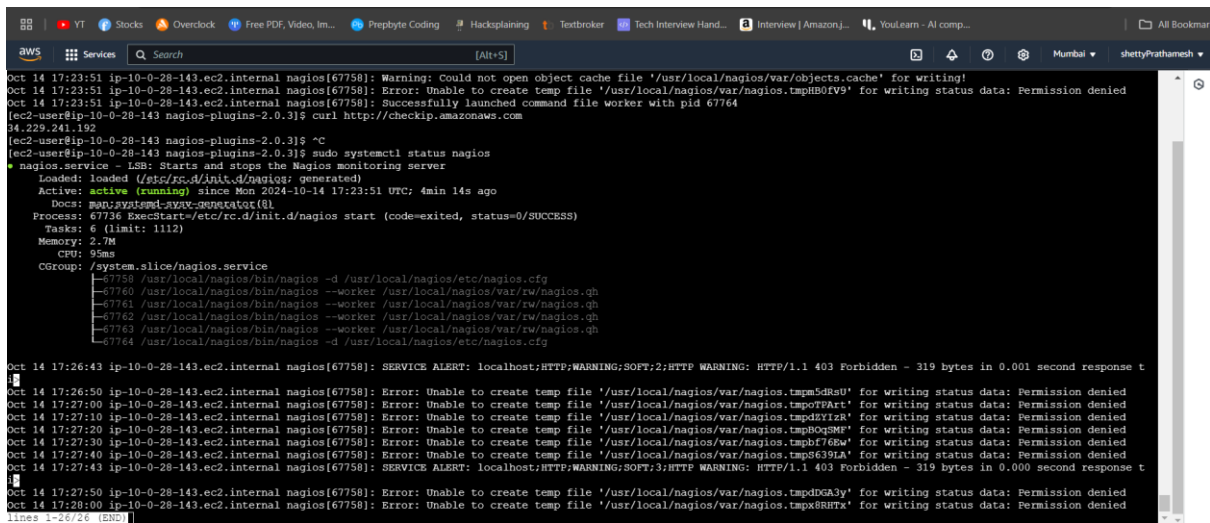
Experiment 10

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

Steps:

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

1. To Confirm that Nagios is running on the server side, run this `sudo systemctl status nagios` on the "NAGIOS HOST".



```
Oct 14 17:23:51 ip-10-0-28-143.ec2.internal nagios[67758]: Warning: Could not open object cache file '/usr/local/nagios/var/objects.cache' for writing!
Oct 14 17:23:51 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpB0tV9' for writing status data: Permission denied
Oct 14 17:23:51 ip-10-0-28-143.ec2.internal nagios[67758]: Successfully launched command file worker with pid 67764
[ec2-user@ip-10-0-28-143 nagios-plugins-2.0.3]$ curl http://checkip.amazonaws.com
94.229.241.102
[ec2-user@ip-10-0-28-143 nagios-plugins-2.0.3]$ ^C
[ec2-user@ip-10-0-28-143 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; generated)
   Active: active (running) since Mon 2024-10-14 17:23:51 UTC; 4min 14s ago
     Docs: man:systemd-sv-generator(8)
   Process: 67736 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
    Tasks: 6 (limit: 1112)
   Memory: 2.7M
     CPU: 95ms
   CGroup: /system.slice/nagios.service
           └─67758 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             └─67760 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─67761 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─67762 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                   └─67763 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─67764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

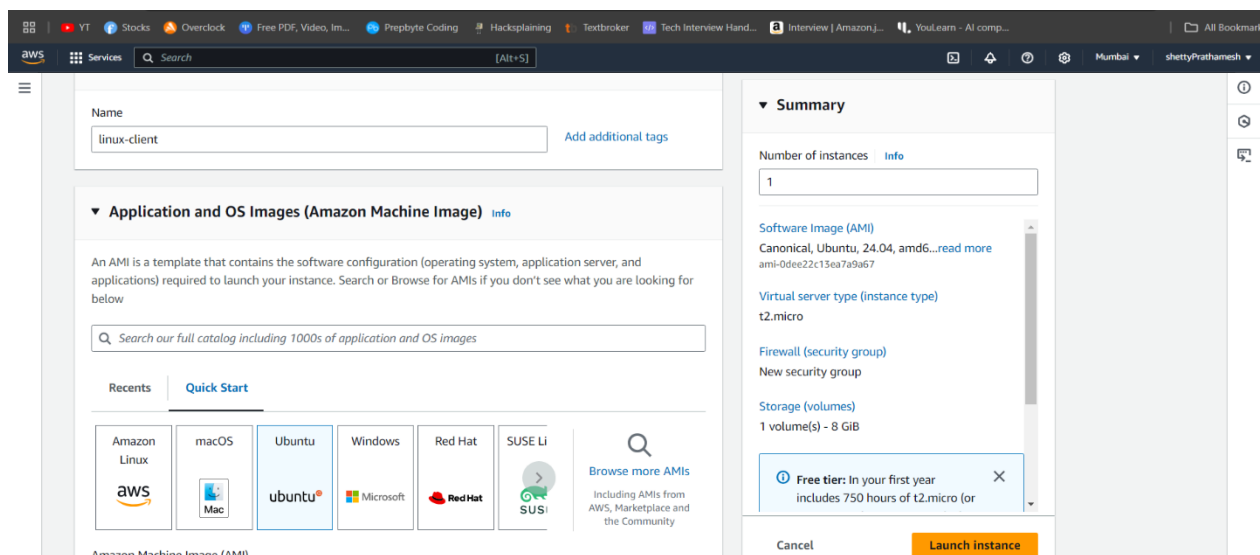
Oct 14 17:26:43 ip-10-0-28-143.ec2.internal nagios[67758]: SERVICE ALERT: localhost:HTTP:WARNING:SOFT:2:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response t
Oct 14 17:26:50 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpm5dRsU' for writing status data: Permission denied
Oct 14 17:27:00 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpoTfVt' for writing status data: Permission denied
Oct 14 17:27:10 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpdYtZM' for writing status data: Permission denied
Oct 14 17:27:20 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpB0q3MP' for writing status data: Permission denied
Oct 14 17:27:30 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpbF7Gw' for writing status data: Permission denied
Oct 14 17:27:40 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpS639LA' for writing status data: Permission denied
Oct 14 17:27:43 ip-10-0-28-143.ec2.internal nagios[67758]: SERVICE ALERT: localhost:HTTP:WARNING:SOFT:3:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response t
Oct 14 17:27:50 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpdDGA3y' for writing status data: Permission denied
Oct 14 17:28:00 ip-10-0-28-143.ec2.internal nagios[67758]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpx8RHTx' for writing status data: Permission denied
lines 1-26/26 (END)
```

You can proceed if you get this message.

2. Before we begin,

To monitor a Linux machine, create an Ubuntu 20.04 server EC2 Instance in AWS.

Provide it with the same security group as the Nagios Host and name it 'linux-client' alongside the host.



The screenshot shows the AWS Management Console 'Launch Instance' wizard. The 'Name' field is set to 'linux-client'. Under 'Application and OS Images (Amazon Machine Image)', the 'Ubuntu' tab is selected, showing 'Canonical, Ubuntu, 24.04, amd64...'. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Firewall (security group)' is set to 'New security group'. The 'Storage (volumes)' section shows '1 volume(s) - 8 GiB'. A 'Free tier' banner indicates 'In your first year includes 750 hours of t2.micro (or)'. The 'Launch instance' button is highlighted in orange.

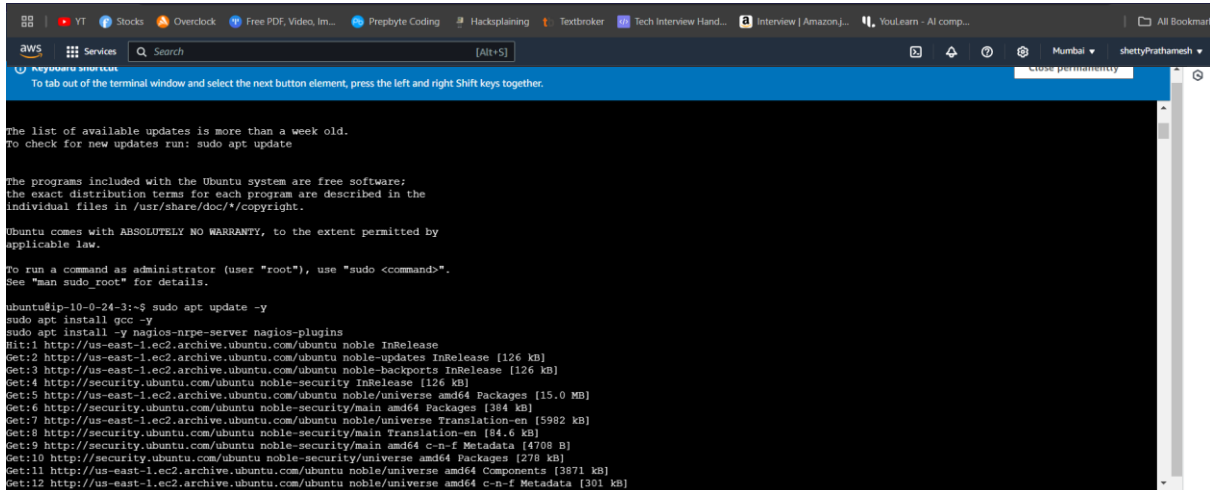
For now, leave this machine as is, and go back to your nagios HOST machine.

Step 3: On client side make a package index update and install gcc, nagios-nrpe-server and the plugins.

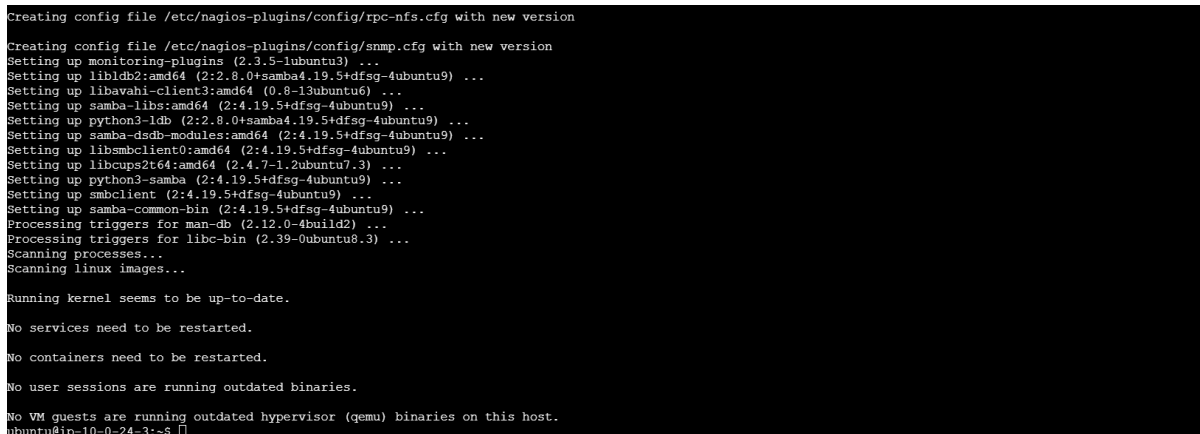
```
sudo apt update -y
```

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

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



```
ubuntu@ip-10-0-24-3:~$ 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 [384 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 [84.6 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4708 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [278 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [361 kB]
```



```
Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libbd2-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.

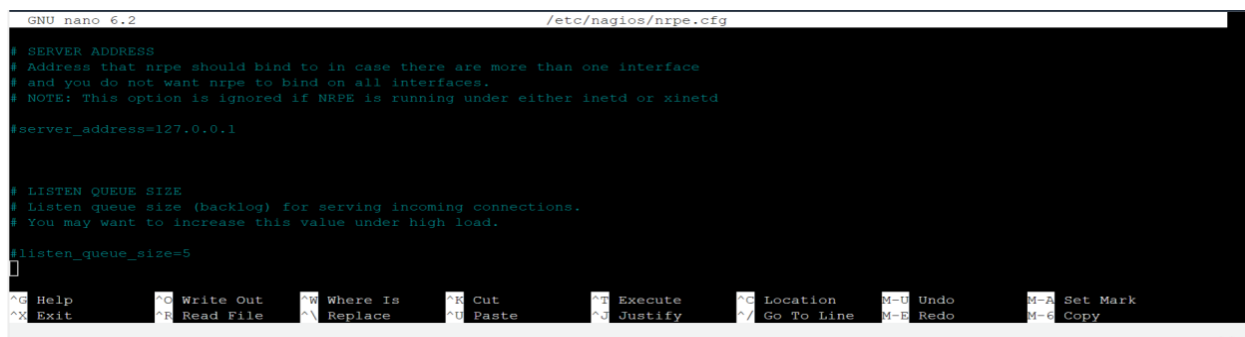
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-24-3:~$
```

Step 4: Open nrpe.cfg file to make changes.

```
sudo nano /etc/nagios/nrpe.cfg
```



```
GNU nano 6.2 /etc/nagios/nrpe.cfg
# SERVER ADDRESS
# Address that nrpe should bind to in case there are more than one interface
# and you do not want nrpe to bind on all interfaces.
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

#server_address=127.0.0.1

# LISTEN QUEUE SIZE
# Listen queue size (backlog) for serving incoming connections.
# You may want to increase this value under high load.

#listen_queue_size=5
]
```

```
GNU nano 2.2 /etc/nagios/nrpe.cfg

rpe_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

llowed hosts=127.0.0.1,34.229.241.192
erver_address=10.0.24.3

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

Help      Write Out  Where Is   Cut        Execute    Location   Undo      Set Mark   To Bracket  Previous  Back
Exit      Read File  Replace    Paste      Justify    Go To Line Redo      Copy       Where Was  Next     Forward
```

Step 5: Restart the NRPE server

`sudo systemctl restart nagios-nrpe-server`

```
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

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.
root@ip-172-31-41-41:/home/ubuntu# sudo nano /etc/nagios/nrpe.cfg
root@ip-172-31-41-41:/home/ubuntu# sudo nano /etc/nagios/nrpe.cfg
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl restart nagios-nrpe-server
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl status nagios-nrpe-server
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
```

Step 6: On the server run this command

`ps -ef | grep nagios`

```
ubuntu@ip-10-0-24-3:~$ ps -ef | grep nagios
nagios   3301      1  0 17:38 ?        00:00:00 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f
ubuntu   6685    1441  0 17:48 pts/0    00:00:00 grep --color=auto nagios
ubuntu@ip-10-0-24-3:~$
```

Step 7: Become a root user and create 2 folders 1.sudo su 2.mkdir /usr/local/nagios/etc/objects/monitorhosts 3.mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts Copy the sample localhost.cfg file to linuxhost folder 4.cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
root@ip-172-31-44-151:/home/ubuntu# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhos
ts/linuxserver.cfg
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

Step 8: Open linuxserver.cfg using nano and make the following changes

nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg Change the hostname to linux server (EVERYWHERE ON THE FILE) Change address to the public IP address of your LINUX CLIENT.

```
GNU nano 6.2 /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          localhost
    alias              localhost
    address            127.0.0.1
}

^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

i-03a3e79fc5ab0a056 (cutenagios_server) X
```

```
GNU nano 6.2 /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg *
#####
# HOST GROUP DEFINITION
#####
# Define an optional hostgroup for Linux machines
define hostgroup {
    hostgroup_name     linux-servers        ; 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
}

^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
```

Change hostgroup_name under hostgroup to linux-servers1

Step 9: Open the Nagios Config file and add the following line nano /usr/local/nagios/etc/nagios.cfg Add this line
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
GNU nano 6.2 /usr/local/nagios/etc/nagios.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
Save modified buffer?
Y Yes
N No      ^C Cancel
```

Step 10: Verify the configuration files.

```
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
root@ip-172-31-44-151:/home/ubuntu# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.14
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2023-08-01
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 8 services.
  Checked 1 hosts.
  Checked 1 host 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 1 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-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/nagios.cfg
```

Step 11: Restart the nagios service service nagios restart

Sudo systemctl status nagios

```
● nagios.service - Nagios Core 4.4.14
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-09-30 08:54:01 UTC; 20s ago
     Docs: https://www.nagios.org/documentation
   Process: 55285 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 55286 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 55287 (nagios)
    Tasks: 6 (limit: 1141)
   Memory: 5.3M
      CPU: 252ms
   CGroup: /system.slice/nagios.service
           └─55287 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             └─55288 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─55289 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─55290 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                   └─55291 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─55292 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 08:54:01 ip-172-31-44-151 nagios[55287]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
lines 1-19
```

Step 12: Now, check your nagios dashboard and you'll see a new host being added.

The screenshot shows the Nagios web interface. On the left is a sidebar with navigation links: General, Home, Documentation, Current Status, Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, Problems, Reports, Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, Event Log, System, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

The main content area is titled "Current Network Status" and shows "Last Updated: Sat Sep 30 18:22:09 UTC 2023". Below this, there are two summary tables:

Up	Down	Unreachable	Pending
0	0	0	0

OK	Warning	Unknown	Critical	Pending
13	0	0	3	0

Below these are two more summary tables:

All Types
0

All Types
2

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

Host **	Status **	Last Check **	Duration **	Status Information
linuxserver	UP	09-30-2023 18:17:06	0d 0h 5m 3s	PING OK - Packet loss = 0%, RTA = 0.62 ms
linuxserver	UP	09-30-2023 18:20:14	0d 0h 28m 7s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Results 1 - 2 of 2 Matching Hosts

The screenshot shows the Nagios web interface. On the left is a sidebar with navigation links: General, Home, Documentation, Current Status, Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, Problems, Reports, Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, Event Log, System, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

The main content area is titled "Service Status Details For All Hosts" and shows "Last Updated: Tue Oct 3 23:38:11 UTC 2023". Below this, there are two summary tables:

Up	Down	Unreachable	Pending
0	0	0	0

OK	Warning	Unknown	Critical	Pending
13	0	0	3	0

Below these are two more summary tables:

All Types
0

All Types
2

The "Service Status Details For All Hosts" table shows the following data:

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
linuxserver	Current Load	OK	10-03-2023 23:34:51	3d 13h 47m 10s	1/4	OK - load average: 0.00, 0.02, 0.00
linuxserver	Current Users	OK	10-03-2023 23:35:29	3d 13h 46m 32s	1/4	USERS OK - 2 users currently logged in
linuxserver	HTTP	CRITICAL	10-03-2023 23:36:06	0d 0h 12m 5s	4/4	CRITICAL - Socket timeout
linuxserver	PING	OK	10-03-2023 23:36:44	0d 0h 1m 27s	1/4	PING OK - Packet loss = 0%, RTA = 0.60 ms
linuxserver	Root Partition	OK	10-03-2023 23:37:21	3d 13h 44m 40s	1/4	DISK OK - free space: / 4859 MB (62.78% inode=88%):
linuxserver	SSH	OK	10-03-2023 23:37:59	0d 0h 0m 12s	1/4	SSH OK - OpenSSH_8.9p1 Ubuntu-3ubuntu0.1 (protocol 2.0)
linuxserver	Swap Usage	CRITICAL	10-03-2023 23:38:36	3d 13h 43m 25s	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	10-03-2023 23:34:14	3d 13h 42m 47s	1/4	PROCS OK: 39 processes with STATE = RSZDT
linuxserver	Current Load	OK	10-03-2023 23:35:10	3d 14h 43m 33s	1/4	OK - load average: 0.00, 0.02, 0.00
linuxserver	Current Users	OK	10-03-2023 23:35:47	3d 14h 42m 55s	1/4	USERS OK - 2 users currently logged in
linuxserver	HTTP	CRITICAL	10-03-2023 23:36:25	3d 14h 42m 18s	1/4	HTTP OK: HTTP/1.1 200 OK - 10945 bytes in 0.000 second response time
linuxserver	PING	OK	10-03-2023 23:37:02	3d 14h 41m 40s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
linuxserver	Root Partition	OK	10-03-2023 23:37:40	3d 14h 41m 3s	1/4	DISK OK - free space: / 4859 MB (62.78% inode=88%):
linuxserver	SSH	OK	10-03-2023 23:38:17	3d 14h 40m 25s	1/4	SSH OK - OpenSSH_8.9p1 Ubuntu-3ubuntu0.4 (protocol 2.0)
linuxserver	Swap Usage	CRITICAL	10-03-2023 23:38:55	3d 14h 36m 48s	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	10-03-2023 23:33:24	3d 14h 39m 10s	1/4	PROCS OK: 40 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

As you can see, we have our linuxserver up and running. It is showing critical status on HTTP due to permission errors and swap because there is no partition created.

In this case, we have monitored - Servers: 1 linux server

Services: swap

Ports: 22, 80 (ssh, http)

Processes: User status, Current load, total processes, root partition, etc.

Recommended Cleanup

- Terminate both of your EC-2 instances to avoid charges.
- Delete the security group if you created a new one (it won't affect your bill, you may avoid it)

Conclusion:

Thus, we learned about service monitoring using Nagios and successfully monitored a Linux Server and monitored its different ports and services using Nagios and NRPE.