

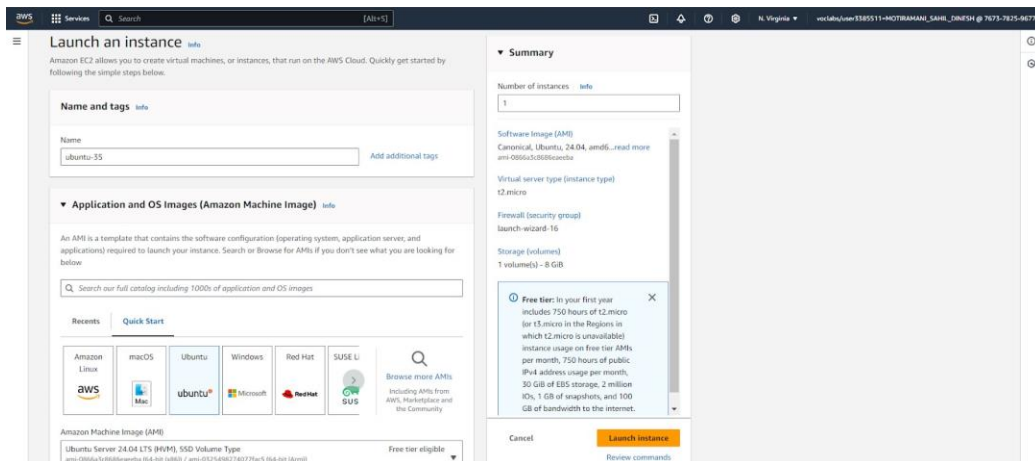
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Prerequisites:

- 1) An Amazon Linux instance with nagios already set up.

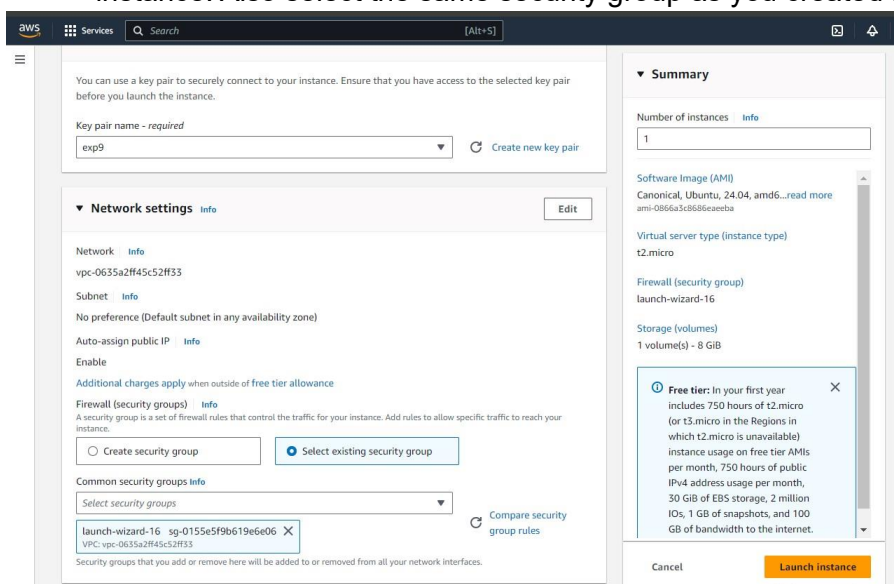
Step 1: Set up ubuntu instance

- 1) Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.



Select The OS Image as Ubuntu.

- 2) Make sure to select the same private key that you created for the Amazon Linux instance. Also select the same security group as you created for the Linux instance.



- 3) Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect. Click on SSH client. Copy the example command. Now, we have to connect our local OS terminal to the instance using SSH. For this, open terminal where the private key file is located (.pem). Paste the copied SSH command and run it.

Step 2: Execute the following on Nagios Host machine (Linux)

- 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-172-31-39-94 nagios-4.4.6]$ ps -ef | grep nagios
nagios 68289 1 0 11:02 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 68290 68289 0 11:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 68291 68289 0 11:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 68292 68289 0 11:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 68293 68289 0 11:02 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 68294 68289 0 11:02 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 71786 2942 0 11:51 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$
```

- 2) Now, make yourself as the root user, and create a folder with the path
'/usr/local/nagios/etc/objects/monitorhosts/linuxhosts'

sudo su

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

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-39-94 nagios-4.4.6]#
```

- 3) 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

- 4) 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

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 ; 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 3.80.168.49
}
```

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        localhost      ; Comma separated list of hosts that
```

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

- 5) Now, we need to edit the nagios configuration file to add this directory.
nano /usr/local/nagios/etc/nagios.cfg Run this command and add the following line
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
# 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/
```

- 6) Now we verify the configuration files.

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```
[root@ip-172-31-39-94 nagios-4.4.6]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
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 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 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
```

7) Once the files are verified, we need to restart the server.

service nagios restart

```
[root@ip-172-31-39-94 nagios-4.4.6]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-39-94 nagios-4.4.6]# |
```

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

- 1) First, we check for any new updates, then we install gcc, 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-172-31-44-65:~$ 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 [382 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
```

```
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-ls: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-172-31-44-65:~$
```

- 2) We need to add the public IP address of our host Nagios machine (Linux) to the nrpe configuration file.

sudo nano /etc/nagios/nrpe.cfg

Under `allowed_hosts`, add the nagios host ip address (public)

```
# NRPE USER
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_user=nagios

# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# 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=127.0.0.1,34.207.239.4
```

Step 4: Check the Nagios Dashboard

1) Go to Nagios dashboard, click on hosts.

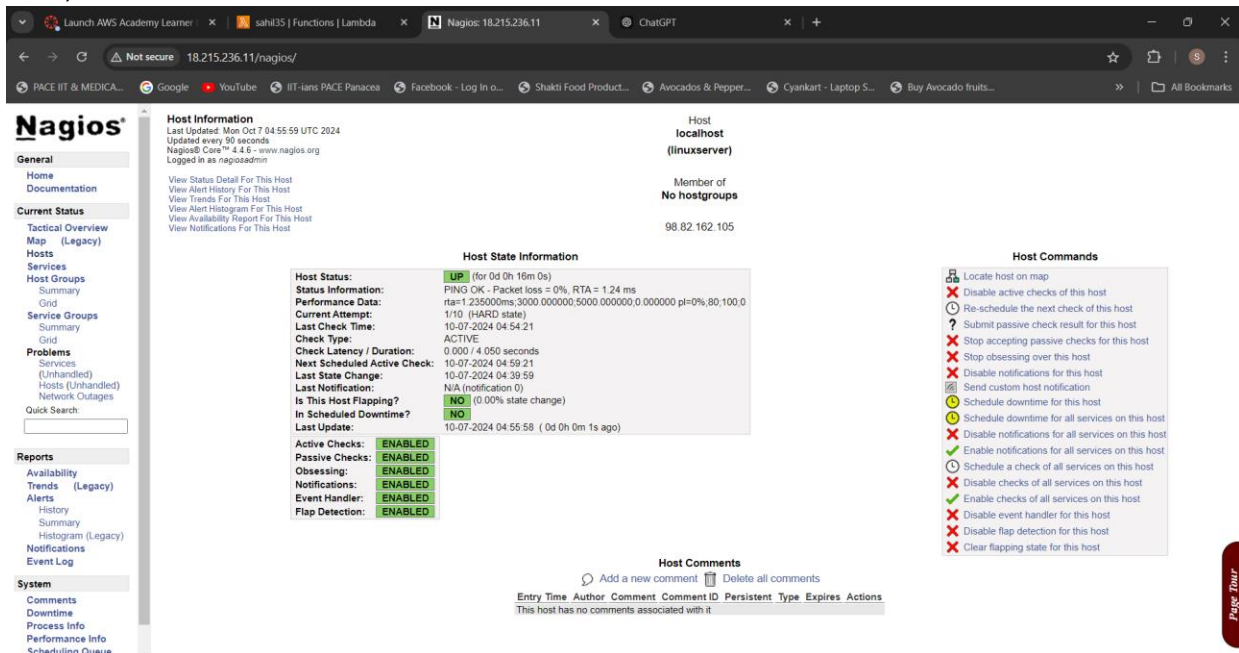
Here, we can see that the linuxserver is also added as a host.

The screenshot shows the Nagios web interface. The left sidebar contains navigation links: General (Home, Documentation), Current Status (Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Summary, Grid, Service Groups, Summary, Grid), and Problems (Services, (Unhandled), Hosts (Unhandled), Network Outages). The main content area displays the 'Host Status Details For All Host Groups' table.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-07-2024 04:54:21	0d 0h 14m 37s	PING OK - Packet loss = 0%, RTA = 1.24 ms
localhost	UP	10-07-2024 04:50:52	0d 17h 52m 26s	PING OK - Packet loss = 0%, RTA = 0.03 ms

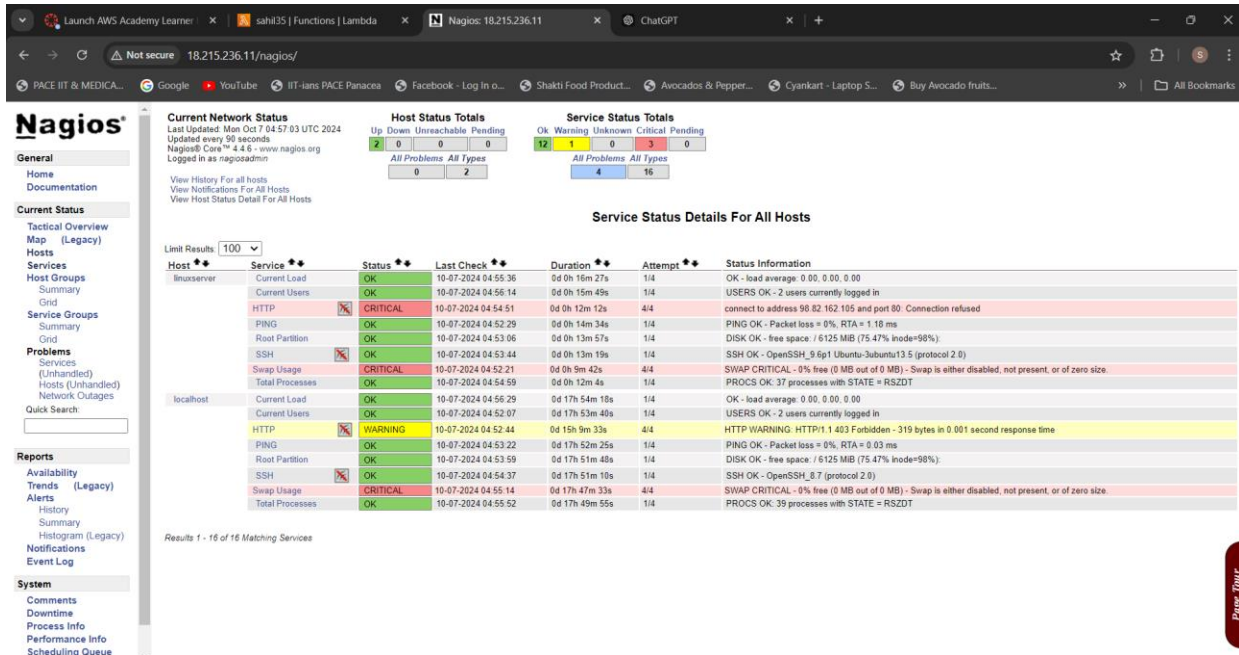
Results 1 - 2 of 2 Matching Hosts

2) Click on linuxserver. Here, we can check all the information about linuxserver host.



The screenshot displays the Nagios web interface for the host 'localhost (linuxserver)'. The interface includes a sidebar with navigation options like General, Current Status, Tactical Overview, and Reports. The main content area shows 'Host Information' with details such as 'Last Updated: Mon Oct 7 04:55:59 UTC 2024' and 'Nagios® Core™ 4.4.6'. Below this, 'Host State Information' provides a comprehensive overview of the host's health, including 'Status Information' (UP), 'Performance Data' (PING OK), 'Current Attempt' (1/10), 'Last Check Time' (10-07-2024 04:54:21), 'Check Type' (ACTIVE), 'Check Latency / Duration' (0.000 / 4.050 seconds), 'Next Scheduled Active Check' (10-07-2024 04:59:21), 'Last State Change' (10-07-2024 04:39:59), 'Last Notification' (N/A), 'Is This Host Flapping?' (NO), 'In Scheduled Downtime?' (NO), and 'Last Update' (10-07-2024 04:55:58). A 'Host Commands' section on the right lists various actions like 'Locate host on map', 'Disable active checks of this host', and 'Re-schedule the next check of this host'. A 'Host Comments' section at the bottom allows for adding or deleting comments.

3) 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 'Service Status Details For All Hosts'. The interface includes a sidebar with navigation options like General, Current Status, Tactical Overview, and Reports. The main content area shows 'Current Network Status' with details such as 'Last Updated: Mon Oct 7 04:57:03 UTC 2024' and 'Nagios® Core™ 4.4.6'. Below this, 'Host Status Totals' and 'Service Status Totals' provide a summary of the overall system health. The 'Service Status Details For All Hosts' table lists the status of various services across different hosts, including 'linuxserver' and 'localhost'. The table columns include 'Host', 'Service', 'Status', 'Last Check', 'Duration', 'Attempt', and 'Status Information'. The table shows that most services are in a 'OK' state, but there are some 'CRITICAL' and 'WARNING' states for certain services like 'HTTP' and 'SSH' on the 'localhost' host.

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.

Conclusion:

In this experiment, we set up port and server monitoring using Nagios.

1. Linux Instance: Hosts the Nagios dashboard and server.
2. Ubuntu Instance : Acts as the second monitored host.
3. Configuration:
 - Add the Ubuntu instance's IP to the Nagios server's configuration.
 - On the Ubuntu instance, configure the NRPE server and allow the Nagios server's IP.
4. Restart NRPE: After configuration, restart the NRPE service on Ubuntu.
5. Monitor: The Ubuntu instance will appear as "linuxserver" on the Nagios dashboard for monitoring.