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

## **Step 1: Confirming Nagios on the Server**

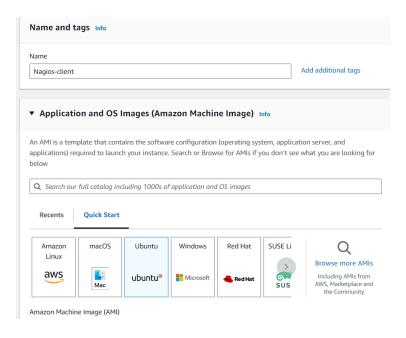
First, ensure that Nagios is running on your server by executing the following command on your Amazon Linux machine (Nagios-host):

sudo systemctl status nagios.

```
[ec2-user@ip-172-31-86-175 -]$ 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 Sun 2024-10-86 86:42:44 UTC; 11s ago
Docs: https://www.nagios.org/documentation
Process: 2847 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=8/SUCCESS)
Process: 2848 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=8/SUCCESS)
Main PID: 2849 (nagios)
Tasks: 6 (limit: 1112)
Memory: 4, 0M
CPU: 17ms
CGroup: /system.slice/nagios.service
-2849 /usr/local/nagios/bin/nagios - worker /usr/local/nagios/var/rw/nagios.qh
-2850 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2851 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2853 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2854 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2854 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2855 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2853 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2854 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2855 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-2856 /usr/local/nagios/bin/nagios --worker /usr/local/n
```

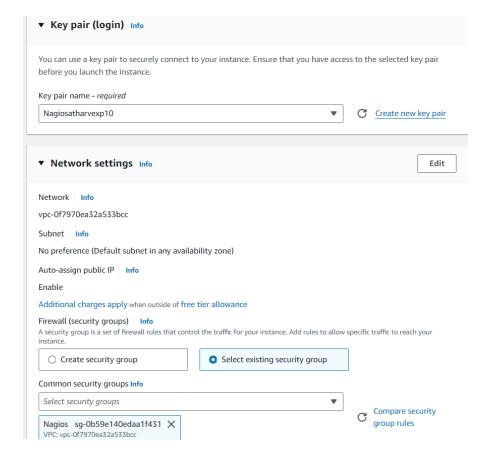
# **Step 2: Creating EC2 Instance**

Next, create a new EC2 instance named **Nagios-client** with the Ubuntu AMI and **t2.micro** instance type. Generate an RSA key pair (.pem file), or use an existing one if available. Make sure to select the security group used in your previous Nagios-host setup.



# **Step 3: Connecting to the Instance**

After creating the EC2 instance, connect to it. Navigate to the folder where the key (.pem) is stored on your local machine. Copy the provided SSH command from the instance's **SSH Client** section and paste it into your terminal.



```
PS D:\Advancedevops key\exp10> ssh -i "Nagiosatharvexp10.pem" ubuntu@ec2-3-80-172-58.compute-1.amazonaws.com Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)
 * Documentation: https://help.ubuntu.com
                    https://landscape.canonical.com
https://ubuntu.com/pro
 * Management:
 System information as of Sun Oct 6 06:55:47 UTC 2024
  System load: 0.08
                                    Processes:
                                                             106
  Usage of /: 22.9% of 6.71GB Users logged in:
  Memory usage: 20%
                                    IPv4 address for enX0: 172.31.46.49
  Swap usage:
Expanded Security Maintenance for Applications is not enabled.
O updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-46-49:~$ |
```

# **Step 4: Checking Nagios Status**

To confirm Nagios is running correctly on the Nagios-host, execute the following: ps -ef | grep nagios.

```
[ec2-user@ip-172-31-86-175 ~]$ ps -6
                                                                                          -ef | grep nagios
? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
pts/0 00:00:00 grep --color=auto nagios
                               2850
                                                     2849
                                                                  0 06:42 ?
                                                     2849 0 06:42 ?
                               2851
                               2852
                                                     2849 0 06:42 ?
                                                    2849 0 06:42 ?
2849 0 06:42 ?
                               2853
                               2854
                                                     2384 0 06:56 pts/0
ec2-user
                               3397
[ec2-user@ip-172-31-86-175 ~]$ |
```

#### **Step 5: Creating Root Directories**

Switch to the root user and create necessary directories for monitoring hosts:

```
sudo su
```

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

```
[ec2-user@ip-172-31-86-175 ~]$ sudo su
[root@ip-172-31-86-175 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-86-175 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-86-175 ec2-user]# |
```

## **Step 6: Configuring Monitoring for Linux Server**

Copy the sample Nagios configuration file for localhost and create a new configuration file for the Linux server:

cp /usr/local/nagios/etc/objects/localhost.cfg
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
[root@ip-172-31-86-175 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg [root@ip-172-31-86-175 ec2-user]# |
```

# **Step 7: Editing Linux Server Configuration**

Open the linuxserver.cfg file and modify the hostname, IP address, and hostgroup as follows:

hostname: linuxserver

• address: Public IP of the Linux client

hostgroup name: linux-servers1

Use the command:

nano

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

```
define host {
                                                                ; 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.
    use
                                  linux-server
    host_name
                                  linuxserver
                                  localhost
172.31.46.49
    alias
# HOST GROUP DEFINITION
define hostgroup {
                                                                 ; The name of the hostgroup
; Long name of the group
; Comma separated list of hosts that belong to this group
    hostgroup_name
                                  linux-servers1
                                  Linux Servers
localhost
     alias
    members
```

## **Step 8: Updating Nagios Configuration**

Add the following line to Nagios' main configuration file to include the monitoring hosts directory:

cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/.
Edit the file using:

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

#### **Step 9: Verifying Nagios Configuration**

To check for any syntax errors in your configuration, run:

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

```
Running pre-flight check on configuration data...
Checking objects...
        Checked 8 services.
        Checked 2 hosts.
        Checked 2 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 2 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...
Total Warnings: 0
Total Errors:
Things look okay - No serious problems were detected during the pre-flight check
```

# Step 10: Restarting Nagios

Restart the Nagios service to apply the configuration changes: sudo service nagios restart.

```
[root@ip-172-31-86-175 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-86-175 ec2-user]# |
```

## **Step 11: Installing NRPE on Nagios Client**

Connect to the Nagios-client instance and update the system. Then install NRPE and necessary Nagios plugins using:

sudo apt update -y sudo apt install gcc -y sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-46-49:~$ sudo apt update -y
sudo apt install gcc -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-padates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [301 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-multiverse amd64 Components [35.0 kB]
Get:18 http://secast-1.ec2.archive.ubuntu.com/ubuntu noble-multiverse amd64 Components [35.0 kB]
Get:19 http://secast-1.ec2.archive.ubuntu.com/ubuntu noble-multiverse amd64 Components [35.0 kB]
Get:19 http://secast-1.ec2.archive.ubuntu.com/ubuntu noble-multiverse amd64 Components [3632 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [3632 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Com
```

```
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version Setting up monitoring-plugins (2.3.5-lubuntu3) ...
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 python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
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 binaries on this host. ubuntu@ip-172-31-46-49:~$
```

## **Step 12: Configuring NRPE**

Edit the NRPE configuration file to allow the Nagios-host to communicate with the client. Add the Nagios-host's IP address under allowed\_hosts: sudo nano /etc/nagios/nrpe.cfg.

```
# 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
# 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,::1,34.238.152.163
# 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
```

# **Step 13: Restarting NRPE**

After editing the NRPE configuration, restart the NRPE server: sudo systemctl restart nagios-nrpe-server.

```
ubuntu@ip-172-31-46-49:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-46-49:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-46-49:~$ |
```

# **Step 14: Checking Nagios and HTTPD Services**

On the Nagios-host, check the status of Nagios and ensure that the HTTPD service is active:

```
sudo systemctl status nagios sudo systemctl status httpd.
```

If HTTPD is not active, start and enable it:

```
sudo systemctl start httpd
sudo systemctl enable httpd.
```

# **Step 15: Accessing Nagios Dashboard**

To view the Nagios dashboard, open your browser and go to:

```
http://<Nagios-host-ip>/nagios.
```

Click on **Hosts** from the left panel to view the status of your Linux server.



✓ Daemon running with PID 3969

#### Nagios<sup>®</sup> Core<sup>™</sup> Version 4.5.5

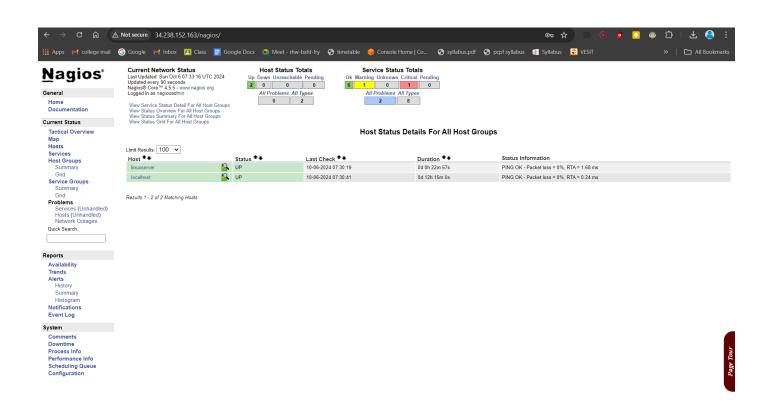
September 17, 2024 Check for updates



Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

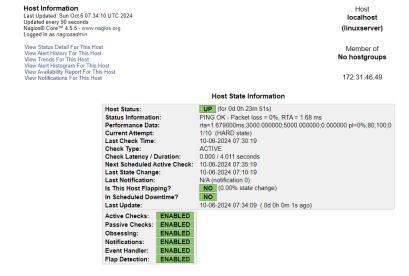
Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks servicemarks owned by Nagios Enterprises, LIC. Use of the Nagios marks is governed by the trademark use restrictions.

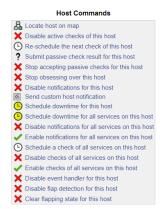
## Nagios\*

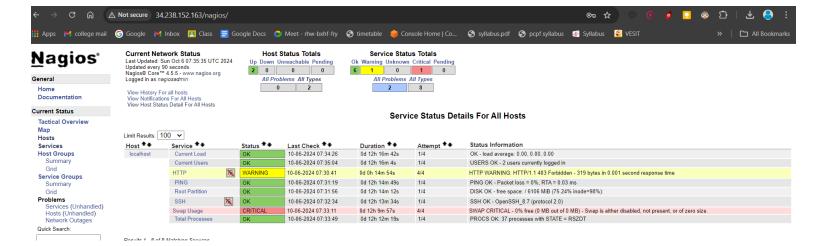


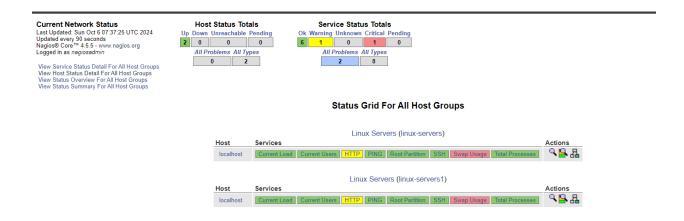
Host

**Host Comments** Entry Time Author Comment Comment ID Persistent Type Expires Actions
This host has no comments associated with it









#### **Conclusion:**

This experiment was designed to set up monitoring for ports, services, and a Linux server using Nagios. By carefully configuring both the Nagios host and client, we were able to monitor essential network services and assess server performance. The experiment demonstrated how Nagios can be used effectively to track system metrics such as CPU usage and memory consumption. This hands-on experience highlights the importance of proactive monitoring in maintaining server health and ensuring the availability of critical services across Linux and Windows platforms.