



Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra

Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab

Experiment 10

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

Roll No.	22
Name	Sarthak Harade
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO5: To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.
Grade:	

- **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”.

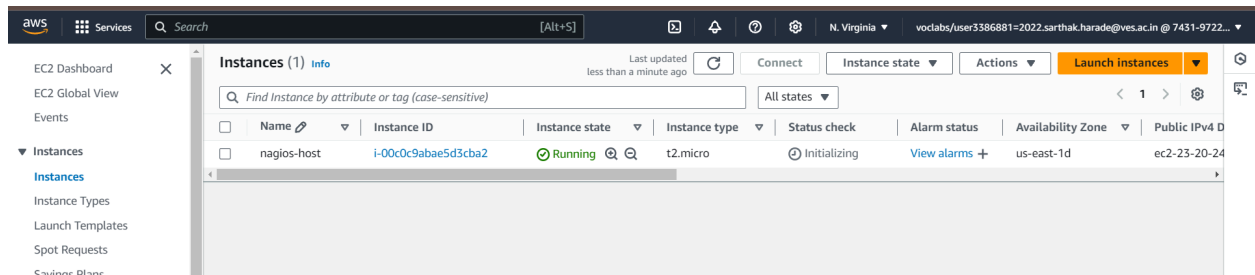
```
[ec2-user@ip-172-31-5-147 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.9
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-10-12 13:09:24 UTC; 15min ago
     Docs: https://www.nagios.org/documentation
  Process: 93966 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 93967 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Main PID: 93968 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 6.9M
      CPU: 522ms
   CGroup: /system.slice/nagios.service
           └─93968 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             └─93969 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─93970 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─93971 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                   └─93972 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─93973 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

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.



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

3. On the server, run this command

ps -ef | grep nagios

```
[ec2-user@ip-172-31-5-147 nagios-plugins-2.0.3]$ ps -ef | grep nagios
nagios      93968      1  0 13:09 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      93969  93968  0 13:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios      93970  93968  0 13:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios      93971  93968  0 13:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios      93972  93968  0 13:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios      93973  93968  0 13:09 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    95038    66019  0 13:26 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-5-147 nagios-plugins-2.0.3]$
```

4. Become a root user and create 2 folders

sudo su

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

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

5. Copy the sample localhost.cfg file to linuxhost folder

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

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

6. Open linuxserver.cfg using nano and make the following changes

nano

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

Change the hostname to linuxserver (EVERYWHERE ON THE FILE)

Change address to the public IP address of your LINUX CLIENT.

```
# HOST DEFINITION
#
#####
# Define a host for the local machine
define host{
    use                linuxserver          ; 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              linuxserver
    address             54.198.255.177
}

#####
# HOST GROUP DEFINITION
#
#####
^G Help      ^O Write Out ^R Where Is  ^K Cut       ^T Execute  ^G Location M-U Undo      M-A Set Mark M-I To Bracket M-C
^X Exit      ^F Read File ^V Replace  ^U Paste     ^J Justify  ^_ Go To Line M-E Redo    M-6 Copy     ^C Where Was M-W
```

Change hostgroup_name under hostgroup 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        linuxserver[]; Comma separated list of hosts that belong to this group
}

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

Everywhere else on the file, change the hostname to linuxserver instead of localhost.

7. 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/
```

```
# EXPERIMENTAL load controlling options
# To get current defaults based on your system issue a command to
# the query handler. Please note that this is an experimental feature
# and not meant for production use. Used incorrectly it can induce
# enormous latency.
# #core loadctl
# jobs_max - The maximum amount of jobs to run at one time
# jobs_min - The minimum amount of jobs to run at one time
# jobs_limit - The maximum amount of jobs the current load lets us run
# backoff_limit - The minimum backoff_change
# backoff_change - # of jobs to remove from jobs_limit when backing off
# rampup_limit - Minimum rampup_change
# rampup_change - # of jobs to add to jobs_limit when ramping up
# NOTE: The backoff_limit and rampup_limit are NOT used by anything currently,
# so if your system is under load nothing will actively modify the jobs
# even if you have these options enabled, they are for external
# connector information only. However, if you change the jobs_max or
# jobs_min manually here or through the query handler interface that
# WILL affect your system
#loadctl_options=jobs_max=100;backoff_limit=10;rampup_change=5
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

```
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Un
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Re
```

8. Verify the configuration files

```

Running pre-flight check on configuration data...

Checking objects...
  Checked 15 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-172-31-5-147 ec2-user]#

```

You are good to go if there are no errors.

9. Restart the nagios service

service nagios restart

```

Starting nagios (via systemctl): [ OK ]
[ec2-user@ip-172-31-44-218 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
   Active: active (running) since Sat 2024-10-12 09:59:46 UTC; 51s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 66468 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
    Tasks: 6 (limit: 1112)
   Memory: 2.1M
      CPU: 51ms
    CGroup: /system.slice/nagios.service
            └─66490 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
            └─66492 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
            └─66493 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
            └─66494 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
            └─66495 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh

```

Now it is time to switch to the client machine.

10. SSH into the machine or simply use the EC2 Instance Connect

11. Make a package index update and install gcc, nagios-nrpe-server and the plugins.

sudo apt update -y

sudo apt install gcc -y

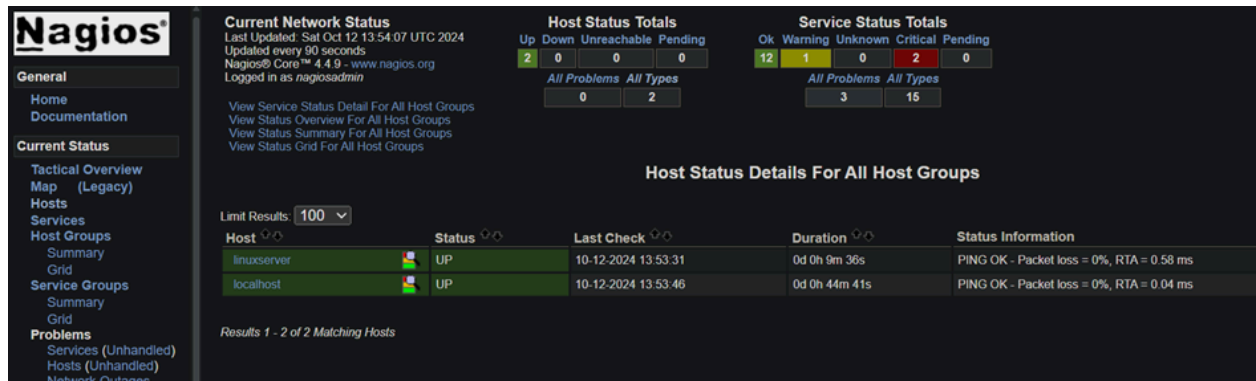
sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-42-197:~$ sudo apt update -y
apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
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://security.ubuntu.com/ubuntu noble-security/main Translation-en [84.6 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 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://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
```

```
# 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, 13.232.100.28
```

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



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.

