Name : Nidhi Rijhwani
Subject : Adv Devops Exp No 10

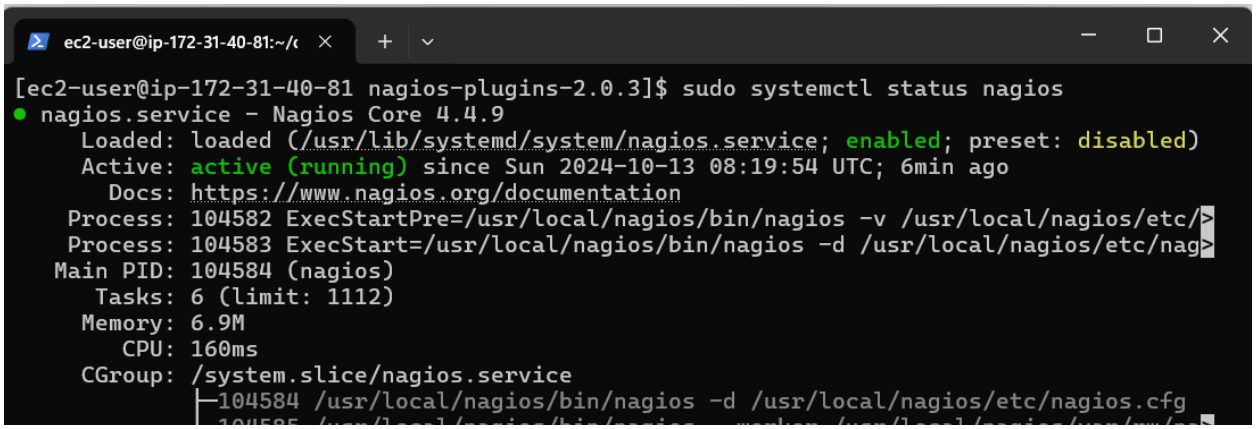
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Theory :

1. Port Monitoring
- Purpose: Ensures that specific ports are open and services are responding.
- Implementation: Using the check_tcp plugin, Nagios can check if a TCP port is accepting connections. This is crucial for monitoring services like HTTP, FTP, or custom applications.
2. Service Monitoring
- Purpose: Monitors the health and performance of services running on servers.
- Implementation: Service checks can include HTTP responses, database connection checks, and application health checks. Plugins like check_http and check_mysql are commonly used.
3. Server Monitoring
- Windows Monitoring: Uses plugins such as check_wmi to gather performance metrics from Windows servers, including CPU usage, memory, disk space, and services status.
- Linux Monitoring: Utilizes native Linux commands (like check_load, check_disk, etc.) to monitor system metrics, ensuring the server is operating efficiently.
4. Configuration
- Monitoring configurations are defined in .cfg files, specifying hosts, services, and checks. The Nagios server periodically polls these configurations to perform checks.

Steps :

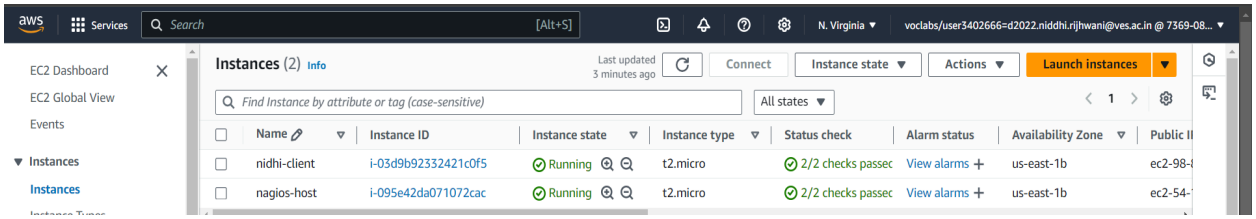
1. **To Confirm that Nagios is running on the server side, run this sudo systemctl status nagios on the "NAGIOS HOST".**



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.



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-172-31-40-81 nagios-plugins-2.0.3]$ ps -ef | grep nagios
nagios    104584       1  0 08:19 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    104585  104584  0 08:19 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    104586  104584  0 08:19 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    104587  104584  0 08:19 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    104588  104584  0 08:19 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    104589  104584  0 08:19 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user  105254    2401  0 08:30 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-40-81 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**

```
[ec2-user@ip-172-31-40-81 nagios-plugins-2.0.3]$ 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 linux host folder

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

```
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserve
r.cfg
```

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

- Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
- Change address to the public IP address of your LINUX CLIENT.

```
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# vim /usr/local/nagios/etc/objects/monitorho
sts/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-ser
ver host template definition.
    host_name               linuxserver
    alias                   linuxserver
    address                 98.83.233.82
}
```

```
###############################################################################
#
# HOST GROUP DEFINITION
#
###############################################################################

# 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 th
at belong to this group
}
```

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/

```
[root@ip-172-31-40-81 nagios-plugins-2.0.3]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-40-81 nagios-plugins-2.0.3]#
```

```
root@ip-172-31-40-81:/home/
  GNU nano 5.8                    /usr/local/nagios/etc/nagios.cfg                    Modi
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/
```

8. Verify the configuration files

```
        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-172-31-40-81 nagios-plugins-2.0.3]#
```

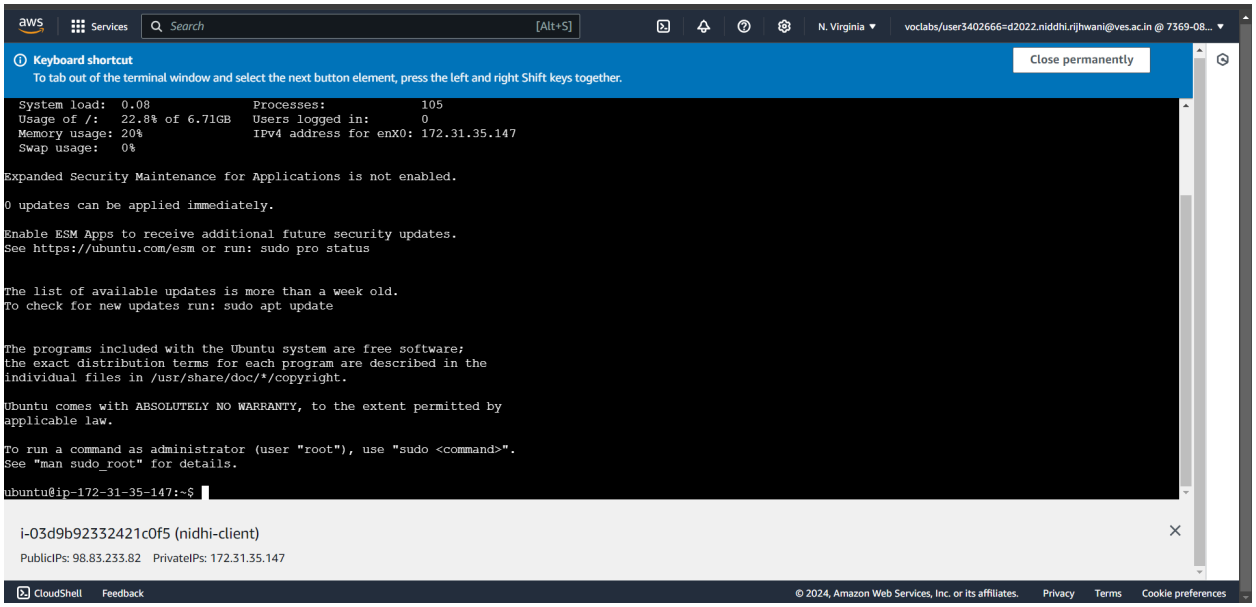
You are good to go if there are no errors.

9. Restart the nagios service

- service nagios restart

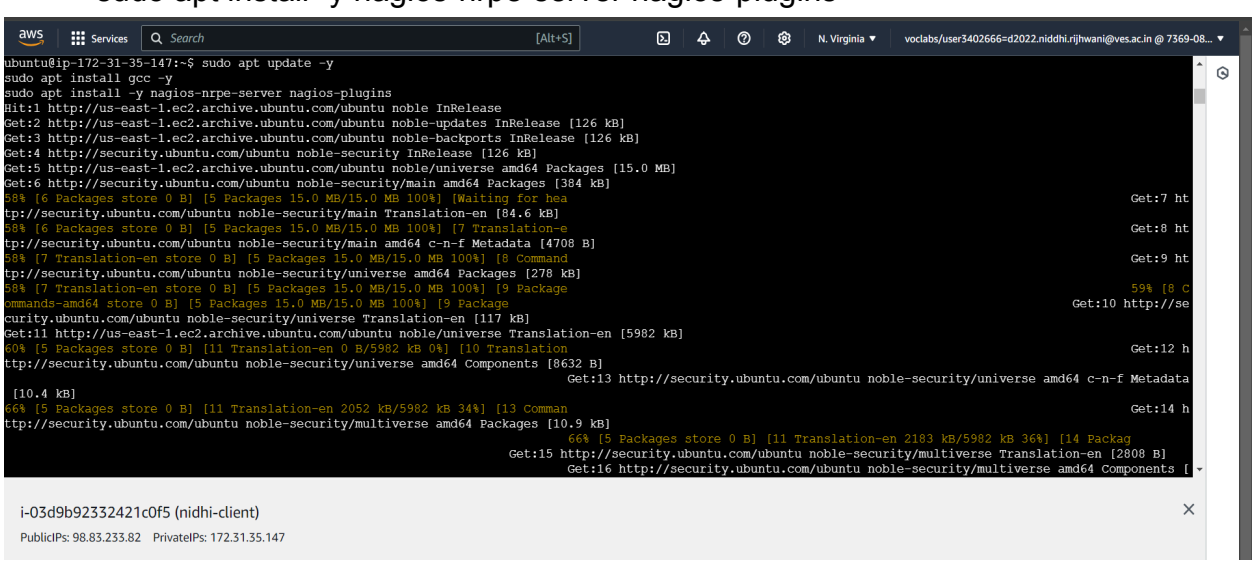
Now it is time to switch to the client machine.

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



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



12. Open nrpe.cfg file to make changes.

- sudo nano /etc/nagios/nrpe.cfg

Under allowed_hosts, add your nagios host IP address like so

```
ubuntu@ip-172-31-35-147:~$ sudo nano /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 mas
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currentl
# 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, 54.164.84.20
```

13. Restart the NRPE server

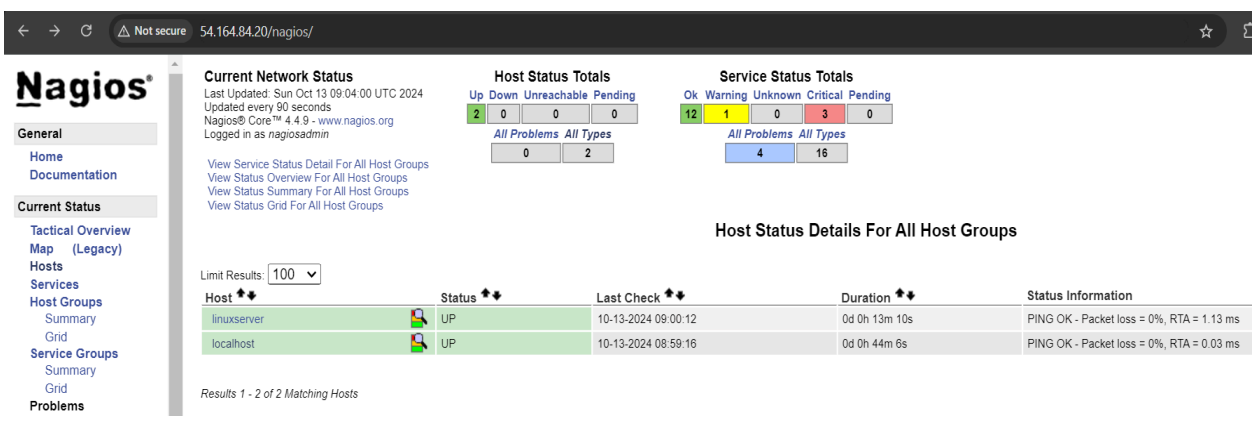
- sudo systemctl restart nagios-nrpe-server

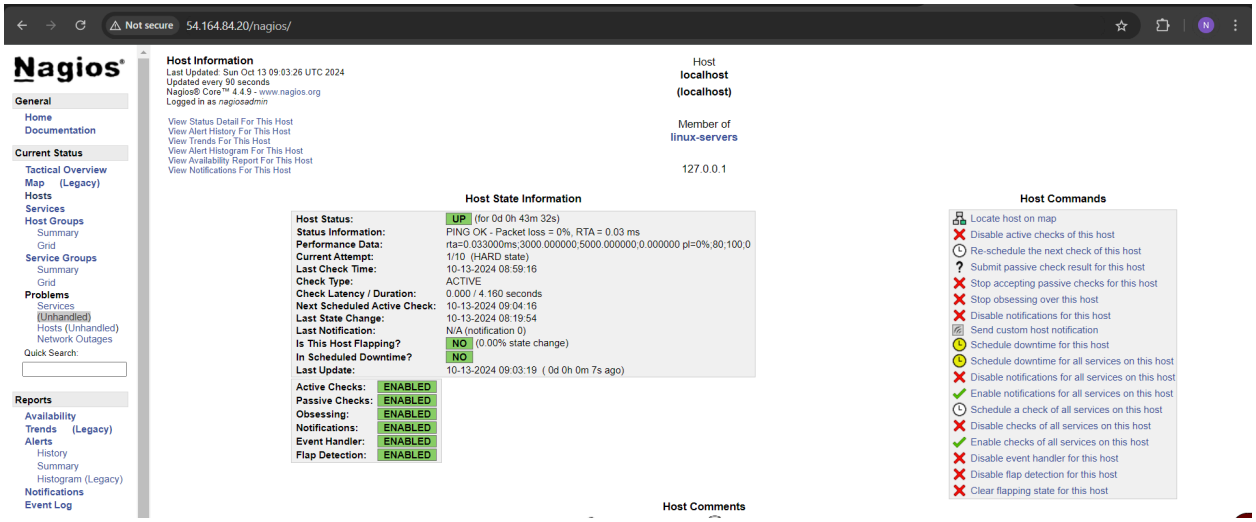
```
ubuntu@ip-172-31-35-147:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-35-147:~$
```

i-03d9b92332421c0f5 (nidhi-client)

PublicIPs: 98.83.233.82 PrivateIPs: 172.31.35.147

14. Now, check your nagios dashboard and you'll see a new host being added. Click on Hosts.





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.