EXPERIMENT NO. 9

NAME : PRANAV POL CLASS : D15A ROLL NO. : 42

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

Theory:

Nagios is a comprehensive monitoring and alerting platform designed to keep track of IT infrastructure, networks, and applications. It provides real-time monitoring, alerting, and reporting capabilities to ensure the health and performance of critical systems.

Key Components of Nagios

1. **Nagios Core**: The open-source foundation of the Nagios monitoring system. It provides the basic framework for monitoring and alerting.
2. **Nagios XI**: A commercial version of Nagios that offers advanced features, a more user-friendly interface, and additional support options.
3. **Nagios Log Server**: A tool for centralized log management, allowing you to view, analyze, and archive logs from various sources.
4. **Nagios Network Analyzer**: Provides detailed insights into network traffic and bandwidth usage.
5. **Nagios Fusion**: Centralizes monitoring data from multiple Nagios instances, providing a unified view of the entire network.

Monitoring Capabilities

1. **Port Monitoring**: Nagios can monitor specific network ports to ensure they are open and responsive. This is crucial for services that rely on these ports.
2. **Service Monitoring**: Nagios checks the status of various services (e.g., web servers, databases) to ensure they are running smoothly.
3. **Server Monitoring**: Nagios can monitor both Windows and Linux servers using agents like NSClient++ for Windows and NRPE for Linux. This includes metrics like CPU usage, memory usage, disk space, and more.

How Nagios Works

1. **Configuration**: Administrators define what to monitor and how to monitor it using configuration files.
2. **Plugins**: Nagios uses plugins to gather information about the status of various services and hosts. These plugins can be custom scripts or pre-built ones.

3. **Scheduling**: Nagios schedules regular checks of the defined services and hosts using the configured plugins.
4. **Alerting**: If a check indicates a problem, Nagios triggers an alert. Alerts can be configured to escalate if not acknowledged within a certain timeframe.
5. **Log Management**: Centralizing and analyzing logs from various sources to detect issues and ensure compliance.

Implementation :

Prerequisites

- AWS Free Tier
- Nagios Server running on an Amazon Linux Machine

1. Confirm Nagios is Running on the Server

Commands -

sudo systemctl status nagios

- Proceed if you see that Nagios is active and running.

```
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Thu 2024-09-26 09:09:51 UTC; 1min 34s ago
       Docs: https://www.nagios.org/documentation
    Process: 68229 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 68230 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 68231 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 2.3M
        CPU: 33ms
     CGroup: /system.slice/nagios.service
             ├─68231 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─68232 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68233 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68234 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68235 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─68236 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: qh: core query handler registered
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: qh: echo service query handler registered
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: qh: help for the query handler registered
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: wproc: Successfully registered manager as @wproc with query handler
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: wproc: Registry request: name=Core Worker 68234;pid=68234
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: wproc: Registry request: name=Core Worker 68235;pid=68235
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: wproc: Registry request: name=Core Worker 68233;pid=68233
Sep 26 09:09:51 ip-172-31-80-22.ec2.internal nagios[68231]: wproc: Registry request: name=Core Worker 68232;pid=68232
lines 1-27
```

2. Create an Ubuntu 20.04 Server EC2 Instance

- Name it linux-client.
- Use the same security group as the Nagios Host.

3. Verify Nagios Process on the Server

Commands -

ps -ef | grep nagios

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ ps -ef | grep nagios
nagios     68231       1  0 09:09 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     68232   68231  0 09:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68233   68231  0 09:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68234   68231  0 09:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68235   68231  0 09:09 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     68236   68231  0 09:09 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user   69851    2909  0 09:38 pts/0    00:00:00 grep --color=auto nagios
```

4. Become Root User and Create Directories

Commands -

sudo su

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

5. Copy Sample Configuration File

Commands -

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

```
[root@ip-172-31-80-22 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-80-22 ec2-user]# sudo nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

6. Edit the Configuration File

Commands -

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

- Change hostname to linuxserver everywhere in 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                     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               linuxserver
    alias                   linuxserver
    address                 3.85.25.81
}
```

- Change hostgroup_name under hostgroup to linux-servers1.

```
###############################################################################
#
# 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 that belong to this group
}
```

7. Update Nagios Configuration

Commands -

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

- Add the following line:
  cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
aws   Services   Search                                                    [Al
  GNU nano 5.8                                                            /usr/loc

# 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 Configuration Files

Commands -

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

- Ensure there are no errors.

```
[root@ip-172-31-80-22 ec2-user]# sudo /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 16 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 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
```

9. Restart Nagios Service

Commands -

sudo systemctl restart nagios

10. SSH into the Client Machine

- Use SSH or EC2 Instance Connect to access the linux-client.

11. Update Package Index and Install Required Packages

Commands -

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

```
ubuntu@ip-172-31-88-112:~$ sudo apt update -y
sudo apt install gcc -y
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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [378 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.0 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4548 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [271 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
```

12. Edit NRPE Configuration File

Commands -

sudo nano /etc/nagios/nrpe.cfg

- Add your Nagios host IP address under allowed_hosts:
  allowed_hosts=<Nagios_Host_IP>

```
  GNU nano 7.2                                                    /etc/nagios/nr
# 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=127.0.0.1,18.208.138.41



# 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.
#
```

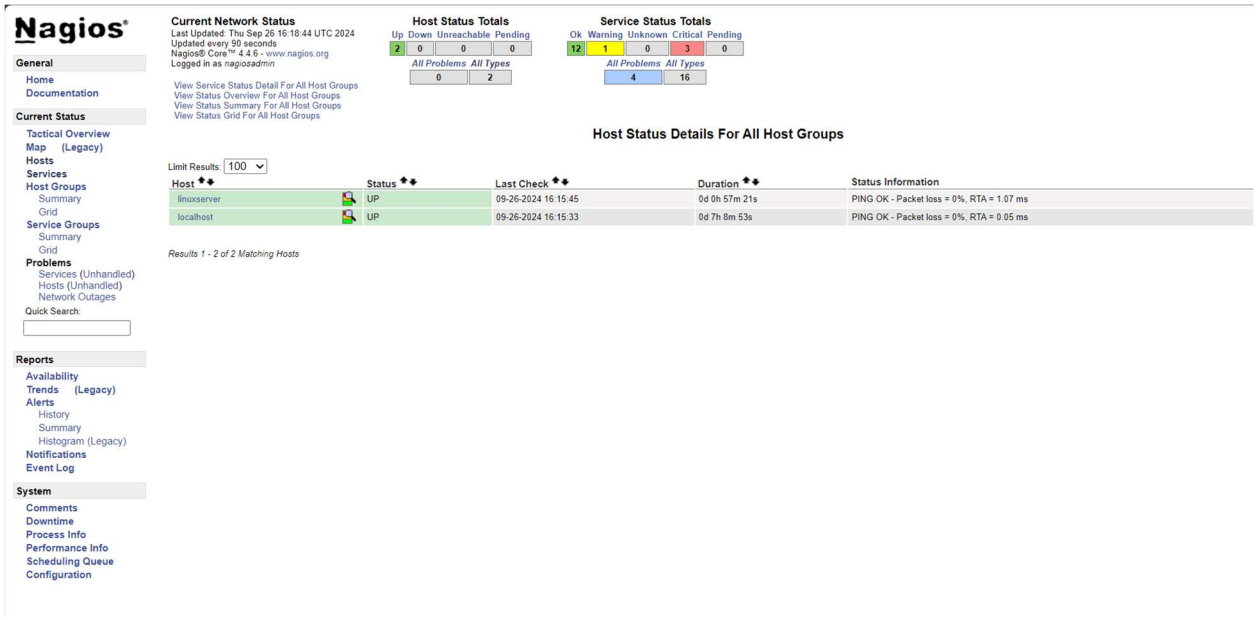
13. Restart NRPE Server

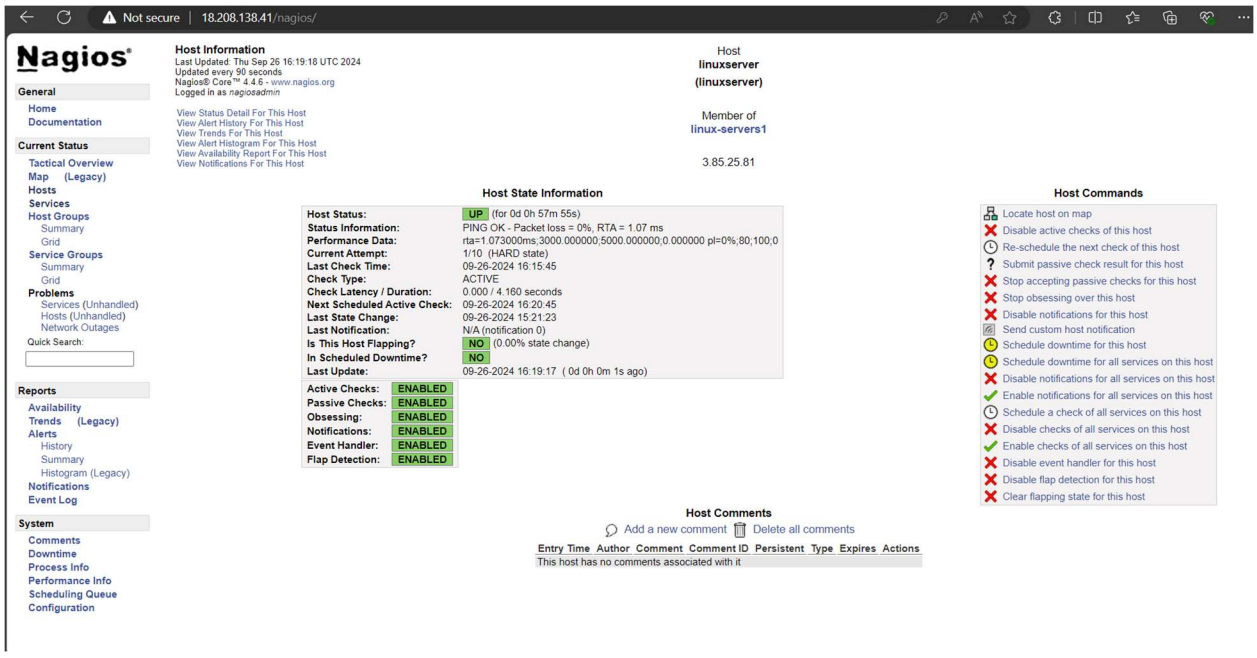
Commands -

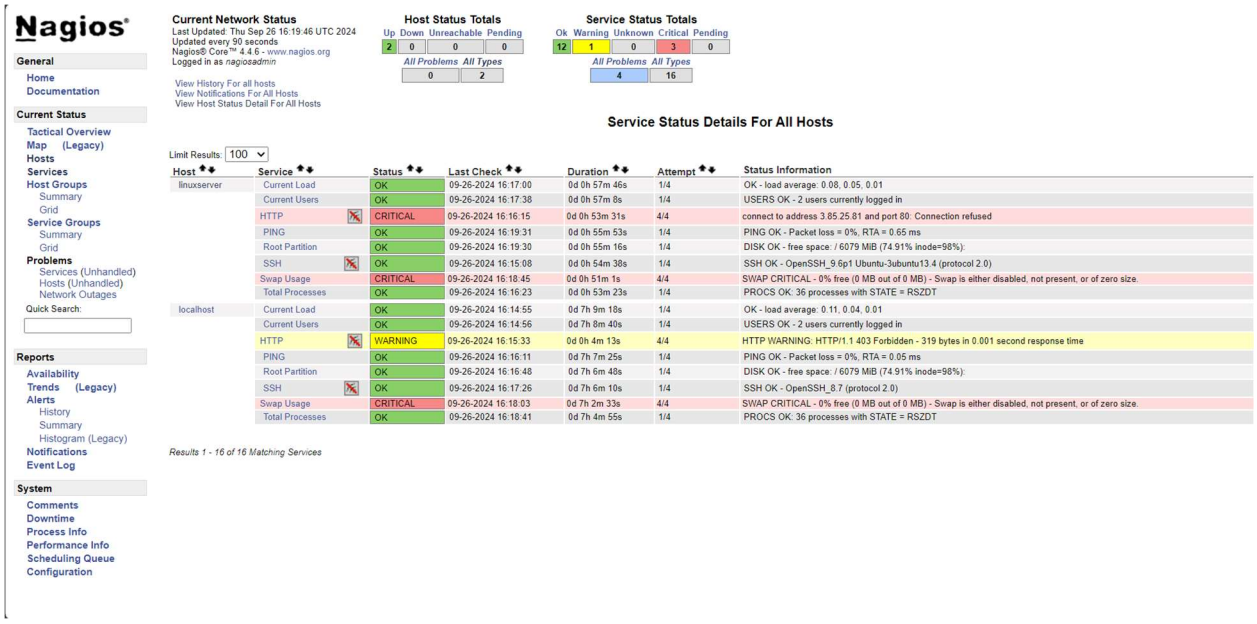
sudo systemctl restart nagios-nrpe-server

14. Check Nagios Dashboard

- Open your browser and navigate to http://<Nagios_Host_IP>/nagios.
- Log in with nagiosadmin and the password you set earlier.
- You should see the new host linuxserver added.
- Click on Hosts to see the host details.
- Click on Services to see all services and ports being monitored







Conclusion:

To perform port, service, and Windows/Linux server monitoring using Nagios, configure the necessary plugins and agents, define the monitoring parameters in the configuration files, and set up alerting mechanisms to ensure timely notifications of any issues. This comprehensive approach ensures robust monitoring and quick response to potential problems, maintaining the health and performance of your IT infrastructure.