Experiment 10

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

Theory:

Introduction

Monitoring network ports and services is essential for maintaining the health and security of IT infrastructure. Nagios is a powerful open-source monitoring tool that enables organizations to track the availability and performance of their servers, services, and network devices. This practical focuses on the theoretical aspects of performing port and service monitoring, as well as monitoring Windows and Linux servers using Nagios.

Importance of Port and Service Monitoring

Ports are communication endpoints for applications on a server, identified by port numbers. Monitoring these ports is crucial for several reasons:

- Security: Open ports can be potential entry points for unauthorized access. Monitoring helps identify any unauthorized or unexpected open ports, reducing the risk of security breaches.
- Availability: Ensuring that critical services are running on their designated ports is vital for maintaining application availability. If a service goes down, it can lead to significant disruptions in business operations.
- Performance: Monitoring ports allows administrators to track the performance of applications. High latency or failure to respond on specific ports can indicate underlying issues that need to be addressed.

Nagios Architecture

Nagios operates on a client-server model:

- Nagios Core: The central monitoring engine that performs checks on hosts and services.
- Plugins: Scripts that extend Nagios's functionality by performing specific checks (e.g., checking if a port is open or if a service is running).
- NRPE (Nagios Remote Plugin Executor): A daemon that allows Nagios to execute plugins on remote hosts, enabling monitoring of systems not directly accessible by the Nagios server.

Monitoring Ports with Nagios

Nagios can monitor both TCP and UDP ports using various plugins. The process involves:

1. Plugin Selection: Administrators can choose from numerous community-provided plugins designed for port monitoring. For example, check_open_port is a plugin that checks specified ports on a host and alerts if any unauthorized ports are found open[1].

2. Configuration: Each plugin must be configured with the necessary parameters, such as the IP address of the host and the specific ports to monitor. This configuration ensures that Nagios can accurately check the status of each port.
3. Alerting Mechanism: When a monitored port becomes unavailable or an unexpected port opens, Nagios triggers alerts via email or SMS, allowing administrators to take immediate action.

Monitoring Services

In addition to port monitoring, Nagios also allows for service monitoring:

- Service Checks: Nagios can check whether specific services (e.g., HTTP, FTP) are running on designated ports. This is typically done using plugins like check_http or check_ftp, which attempt to connect to the service and verify its operational status.
- Custom Checks: Administrators can create custom checks tailored to their environment's needs, ensuring comprehensive monitoring across all critical services.

Windows and Linux Server Monitoring

Nagios supports monitoring across different operating systems, including Windows and Linux:

- Linux Server Monitoring: Using NRPE or SSH, administrators can perform checks on Linux servers remotely. Common checks include CPU load, disk usage, memory consumption, and service status.
- Windows Server Monitoring: For Windows environments, Nagios uses agents like NSClient++ to facilitate communication between the Nagios server and Windows hosts. This allows for checks similar to those performed on Linux systems but tailored for Windows-specific metrics.

Implementation:

Prerequisites

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

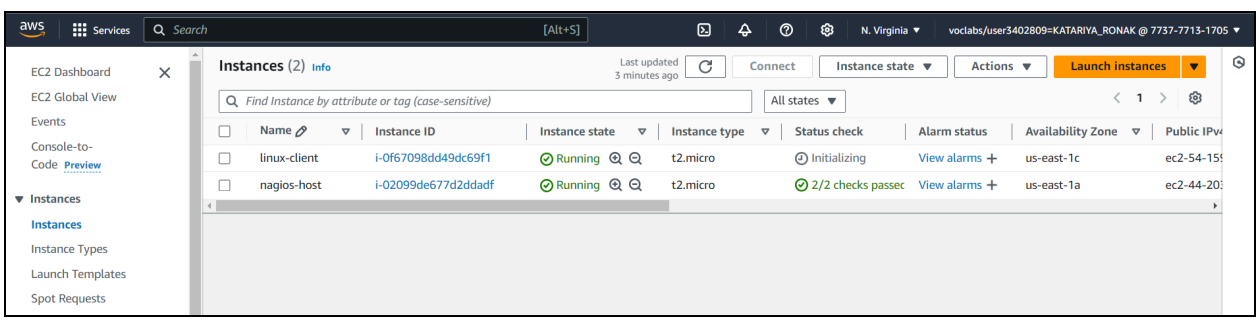
Steps:

1. Confirm Nagios is Running on the Server
Commands -
sudo systemctl status nagios

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

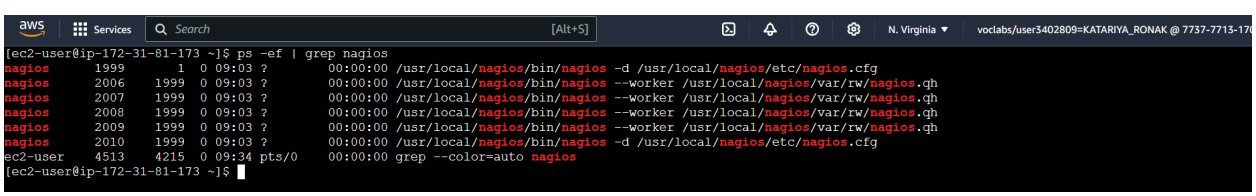
```
● ip-172-31-81-173.ec2.internal
    State: running
    Units: 296 loaded (incl. loaded aliases)
     Jobs: 0 queued
   Failed: 0 units
    Since: Wed 2024-10-02 09:03:18 UTC; 26min ago
  systemd: 252.23-2.amzn2023
   CGroup: /
           ├─init.scope
           │ └─1 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
           ├─system.slice
           │ ├─acpid.service
           │ │ ├─1955 /usr/bin/systemd-inhibit --what=handle-suspend-key:handle-hibernate-key --who=noah "-
           │ │ └─1996 /usr/sbin/acpid -f
           │ ├─amazon-ssm-agent.service
           │ │ └─2341 /usr/bin/amazon-ssm-agent
           │ ├─atd.service
           │ │ └─2352 /usr/sbin/atd -f
           │ ├─auditd.service
           │ │ └─1778 /sbin/auditd
           │ ├─chronyd.service
           │ │ └─2375 /usr/sbin/chronyd -F 2
           │ ├─dbus-broker.service
           │ │ ├─1963 /usr/bin/dbus-broker-launch --scope system --audit
           │ │ └─1971 dbus-broker --log 4 --controller 9 --machine-id ec2c59ef5fdf3c9248d24ff5801dc348 --ma
           │ ├─gssproxy.service
           │ │ └─1998 /usr/sbin/gssproxy -D
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



4. Become Root User and Create Directories

Commands -
sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

```
[ec2-user@ip-172-31-81-173 ~]$ sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-81-173 ec2-user]# ls
```

5. Copy Sample Configuration File
Commands -
cp /usr/local/nagios/etc/objects/localhost.cfg
/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.

```
  GNU nano 5.8                    /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linu
}



###############################################################################
#
# SERVICE DEFINITIONS
#
###############################################################################

# Define a service to "ping" the local machine

define service {

    use                     local-service           ; Name of service template to use
    host_name               linuxserver
    service_description     PING
    check_command           check_ping!100.0,20%!500.0,60%
}



# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
```

```
  GNU nano 5.8                                  /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
# 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                 54.159.91.82
}



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

- Change hostgroup_name under hostgroup to linux-servers1.

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/

```
Services   Search                                   [Alt+S]
  GNU nano 5.8                                  /usr/local/nagios/etc/nagios.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/



# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts.  The CGIs read object definitions from
# this cache file (rather than looking at the object config files
# directly) in order to prevent inconsistencies that can occur
# when the config files are modified after Nagios starts.

object_cache_file=/usr/local/nagios/var/objects.cache

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

8. Verify Configuration Files

Commands -
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
● Ensure there are no errors.

```
aws  Services  Search  [Alt+S]

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 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-172-31-81-173 ec2-user]#
```

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

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-45-86:~$ 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 [380 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 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 [83.1 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8676 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [380 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [157 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 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>

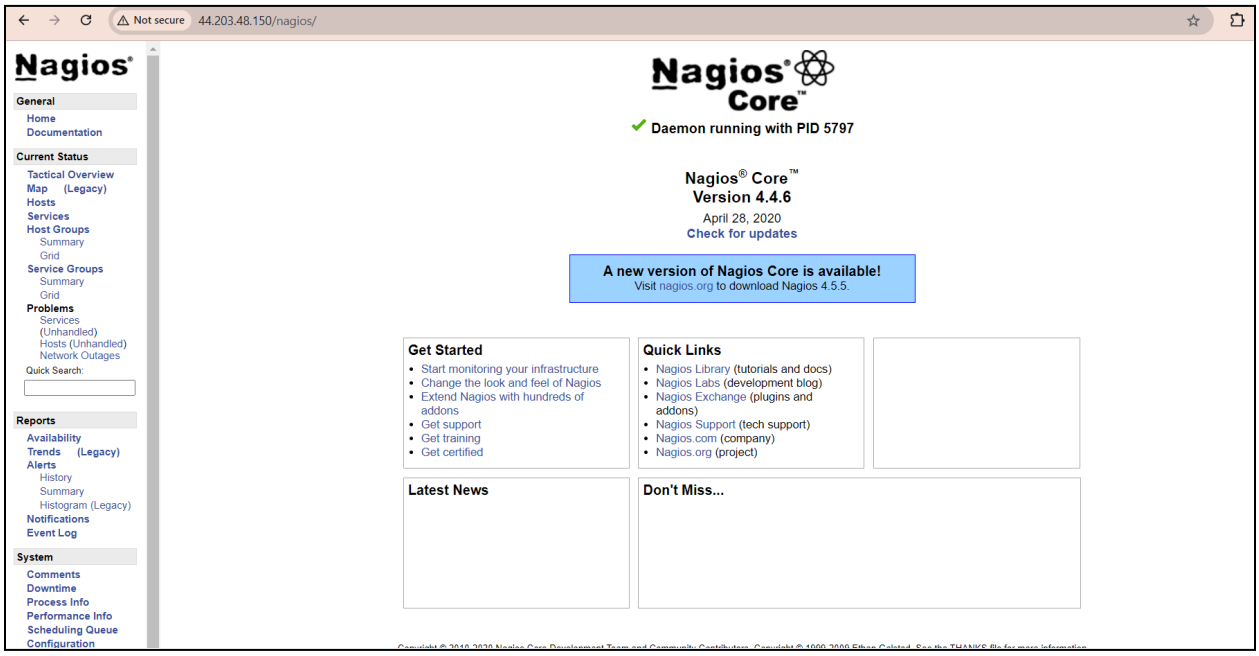
```
GNU nano 7.2                                        /etc/nagios/nrpe.cfg *
# user and is running in standalone mode.

pid_file=/run/nagios/nrpe.pid



# PORT NUMBER
# Port number we should wait for connections on.
# NOTE: This must be a non-privileged port (i.e. > 1024).
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

server_port=5666



# SERVER ADDRESS
# Address that nrpe should bind to in case there are more than one interface
# and you do not want nrpe to bind on all interfaces.
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

#server_address=127.0.0.1
allowed_hosts=127.0.0.1,44.203.48.150

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

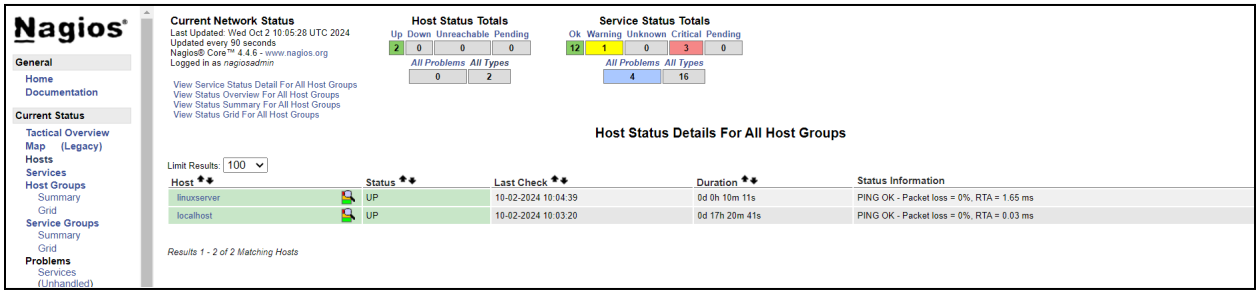
13. Restart NRPE Server

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





Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search:

Reports
- Availability
- Trends (Legacy)
- Alerts
  - History
  - Summary
  - Histogram (Legacy)
- Notifications
- Event Log

Current Network Status
Last Updated: Wed Oct 2 10:06:02 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

View Host Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems	*All Types*
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0

All Problems	*All Types*
4	16

Service Status Details For All Host Groups

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	10-02-2024 10:05:54	0d 0h 10m 8s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 10:01:32	0d 0h 9m 30s	1/4	USERS OK - 3 users currently logged in
	HTTP	CRITICAL	10-02-2024 10:05:09	0d 0h 5m 53s	4/4	connect to address 54.159.91.82 and port 80: Connection refused
	PING	OK	10-02-2024 10:02:47	0d 0h 8m 15s	1/4	PING OK - Packet loss = 0%, RTA = 1.68 ms
	Root Partition	OK	10-02-2024 10:03:24	0d 0h 7m 38s	1/4	DISK OK - free space: / 5973 MiB (73.60% inode=98%):
	SSH	OK	10-02-2024 10:04:02	0d 0h 7m 0s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-3ubuntu13.5 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 10:02:39	0d 0h 3m 23s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-02-2024 10:05:17	0d 0h 5m 45s	1/4	PROCS OK: 39 processes with STATE = RSZDT
localhost	Current Load	OK	10-02-2024 10:03:57	0d 17h 20m 38s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 10:04:35	0d 17h 20m 0s	1/4	USERS OK - 3 users currently logged in
	HTTP	WARNING	10-02-2024 10:05:12	0d 17h 16m 23s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
	PING	OK	10-02-2024 10:05:50	0d 17h 18m 45s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-02-2024 10:01:27	0d 17h 18m 8s	1/4	DISK OK - free space: / 5974 MiB (73.60% inode=98%):
	SSH	OK	10-02-2024 10:02:05	0d 17h 17m 30s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 10:05:20	0d 1h 0m 42s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-02-2024 10:03:20	0d 17h 16m 15s	1/4	PROCS OK: 39 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services