



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.	44
Name	Ganesh Sanjay Pandhre
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.

THEORY :

Port and Service Monitoring:

Port and service monitoring is essential in maintaining the performance and security of a network. Ports are communication endpoints for various services running on a machine, and monitoring them ensures that critical services like SSH, HTTP, and others are active and functioning properly. Service monitoring tracks the status and availability of different services to ensure uninterrupted operations.

Nagios and NRPE:

Nagios is an open-source tool used for monitoring servers, networks, and infrastructure. It can detect system failures and performance issues, making it vital for real-time monitoring. NRPE (Nagios Remote Plugin Executor) extends Nagios' capabilities by enabling monitoring of remote Linux/Windows servers. It allows the Nagios server to execute monitoring scripts (plugins) on remote machines to gather data about system health, services, and ports.

Windows and Linux Server Monitoring:

Monitoring Windows and Linux servers is crucial in both large and small IT environments. Each server's health, including CPU usage, memory, disk space, and running services, must be constantly tracked to prevent downtimes. Nagios can be set up to monitor servers across platforms, offering insights into specific system parameters such as swap usage, active processes, and running ports, helping to avoid system overload or failures.

Ports and Services Monitored:

- SSH (Port 22): Monitored for secure remote access to the server.
- HTTP (Port 80): Monitored to check the availability of web servers and their services.
- Services Monitoring: Apart from ports, Nagios helps monitor key server services like user status, system load, total processes, and the state of critical system partitions (e.g., root partition).

Alerts and Notifications:

Nagios, along with NRPE, continuously monitors these parameters and sends alerts to administrators when thresholds are breached or if a service is down. This proactive approach enables quick resolution before an issue escalates, minimizing system downtime and performance degradation.

```
[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/)
  Process: 93967 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/)
 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
```

```
GNU nano 5.8 /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg Modified
#####
#
# 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              localhost
    address            127.0.0.1
}

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

```
[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]$
```

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

#####
#
# SERVICE DEFINITIONS
#
#####
# Define a service to "ping" the local machine
define service {
```

```
# 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.
```

```
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]#
```

```
ubuntu@ip-172-31-4-175:~$ sudo systemctl restart nagios-nrpe-server  
ubuntu@ip-172-31-4-175:~$
```

```
# 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
```

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

Current Network Status
Last Updated: Sat Oct 12 13:54:07 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.9 - www.nagios.org
Logged in as *nagiosadmin*

View Service 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: 0 All Types: 2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	2	0

All Problems: 3 All Types: 15

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-12-2024 13:53:31	0d 0h 9m 36s	PING OK - Packet loss = 0%, RTA = 0.58 ms
localhost	UP	10-12-2024 13:53:46	0d 0h 44m 41s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Results 1 - 2 of 2 Matching Hosts

CONCLUSION :

Thus, we learned about port and service monitoring using Nagios and successfully monitored a Linux server. Using Nagios and NRPE, we were able to track server performance and monitor key services and ports effectively.