

nagios

A study material for the students of GLS University

What is Continuous Monitoring

Continuous Monitoring (CM) is the practice of constantly tracking applications, servers, networks, and infrastructure in real time to ensure they are running smoothly.

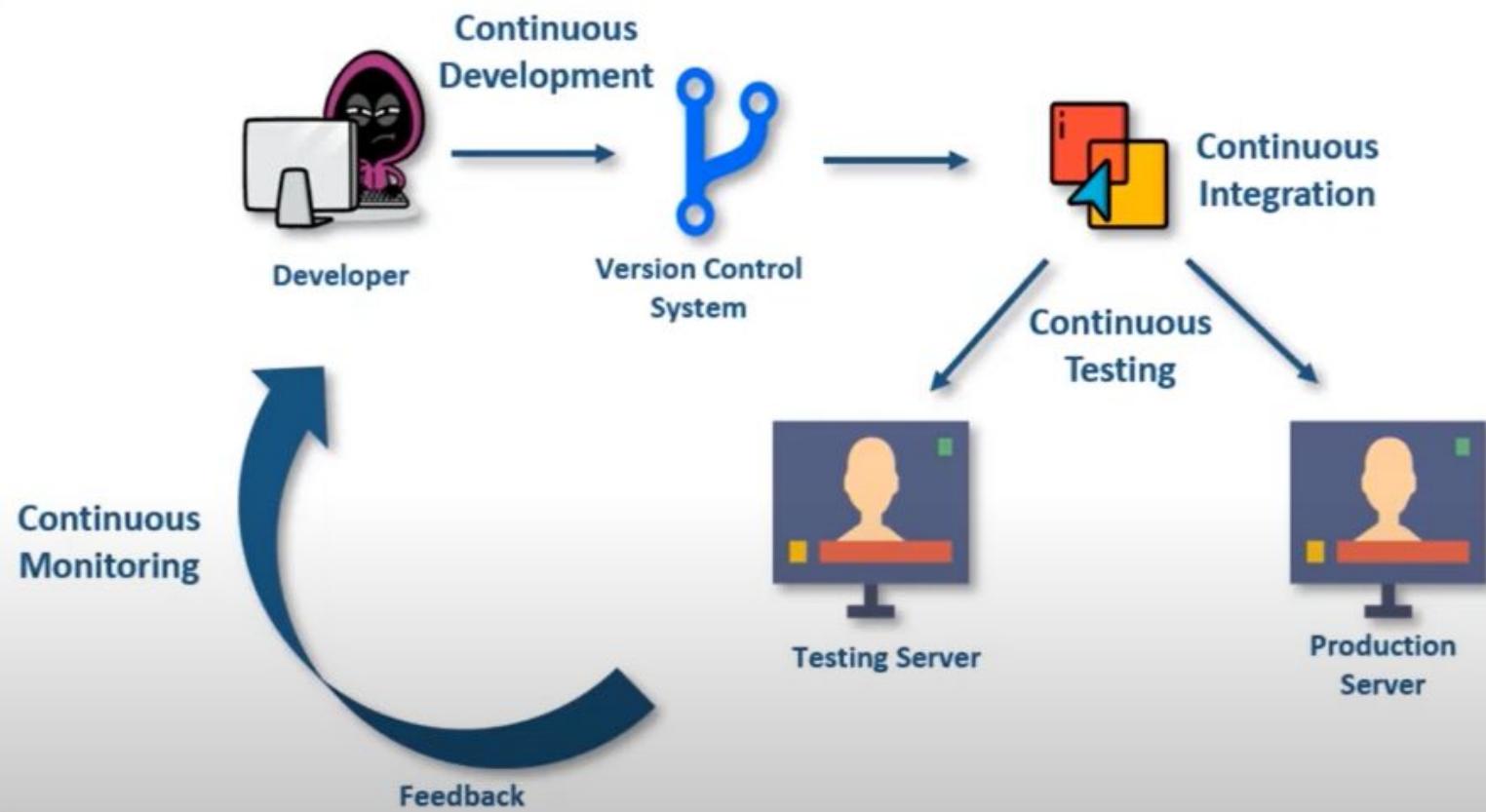
It helps DevOps teams quickly detect errors, downtime, security threats, or performance issues.

In simple words: It keeps “eyes” on the system 24x7 to make sure everything is healthy.

Why Continuous Monitoring

- It detects any network or server problems
- It determine the root cause of any issues
- It maintains the security and availability of the service
- It monitors and troubleshoot server performance issues
- It can respond to issues at the first sign of a problem
- Monitors your entire infrastructure and business processes.

Software Delivery Pipeline



Popular Tools for Continuous Monitoring

- Nagios – Server & network monitoring, alerting system.
- Prometheus + Grafana – Metrics monitoring with visual dashboards.
- ELK Stack (Elasticsearch, Logstash, Kibana) – Log management & analysis.
- Datadog – Cloud-based monitoring & observability.
- Splunk – Big data log analysis and security monitoring.
- New Relic – Application Performance Monitoring (APM).
- Zabbix – Network, server, and cloud monitoring.

Introduction to Nagios

Nagios is one of the oldest and most popular open-source monitoring tools.

Developed to monitor hosts and services and designed to inform the network incidents before end-users, clients do.

It watches hosts and services which we specify and alerts when things go bad and when things get recovered.

Initially developed for server and application monitoring, it is now widely used to monitor networks availability.



History of Nagios

1999 – NetSaint Created: Developed by Ethan Galstad as an open-source tool to monitor servers and networks.

2002 – Renamed Nagios: Trademark issues → NetSaint became Nagios (“Nagios Ain’t Gonna Insist On Sainthood”).

2005–2007 – Rapid Adoption & Commercial Support: Gained global popularity. Nagios Enterprises LLC founded → introduced commercial support & Nagios XI.

2010s – Ecosystem Expansion: Add-ons like PNP4Nagios (graphs), Nagios Fusion (centralized view), Nagios Log Server (logs). Became a comprehensive monitoring solution for enterprises.

2020s – Present Day: Nagios Core (open-source) and Nagios XI (enterprise) widely used. Part of a large ecosystem: Core, XI, Log Server, Network Analyzer, Fusion. Trusted by thousands of organizations worldwide.

Why Nagios?

Real-time Monitoring: Constantly checks health of systems.

Alerting: Sends notifications (email/SMS) when something fails.

High Availability: Helps maintain uptime of critical services.

Flexibility: Supports monitoring for both on-premises and cloud infrastructure.

Community Support: Thousands of ready-to-use plugins available.

Key Features of Nagios

- Oldest
- Good Log and Database System
- Informative and attractive web interface
- Automatically send alerts if condition changes
- Helps you to detect network error or server crashes
- You can monitor the entire business process and IT infrastructure with a single pass.
- Monitor network services like http, smtp, snmp, ftp, ssh, pop, DNS, LDAP etc.

Phases of Continuous Monitoring

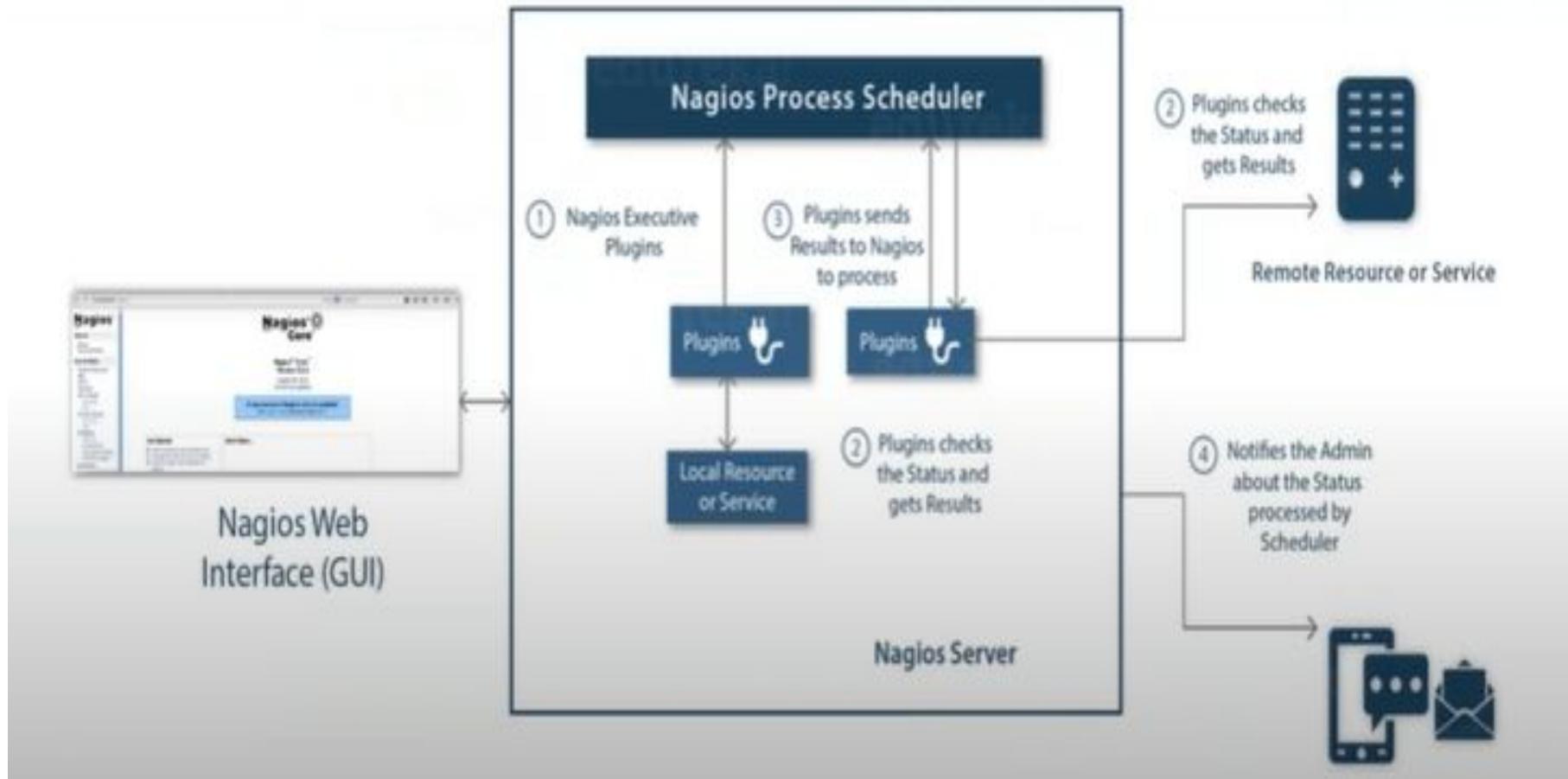
Phase	Key Activities
Define	Identify systems, KPIs, thresholds
Establish	Set policies, roles, tools, architecture
Implement	Configure tools, agents, dashboards
Operate	Monitor continuously, respond to alerts
Review	Analyze trends, optimize monitoring, update policies

Nagios Architecture

Nagios is a client server architecture usually on a network, a nagios server is running on host and plugins are running on the remote host which should you monitor.

How does it work -

- Maintain all details in configuration file
- Daemon read those details what data to be collected
- Daemon uses NRPE plugin to collect data from nodes and stores it in it's own database
- Finally show everything in dashboard



Nagios Terminology

Plugins: Which is external program that can consist of either a script or complied executable.

Host: Is a server or any network device which you want monitor.

Service: Is any metric (Ex. CPU, Memory Usage)

Users: Who has access to web interface

Contacts: Individual administrator or end-user(Ex: Email-Id or Phone-number)

Contact Groups: Grouping contact together(Ex: All Linux Administrators into single group)

Acknowledgement: Temporarily suppress alert notifications

Downtime: Planned activity (Ex. Upgrading software, host hardware replacement etc.)

Nagios Terminology

Latency: Difference between scheduled to run and when it does actually run.

State: SOFT and HARD to avoid false positive alerts.

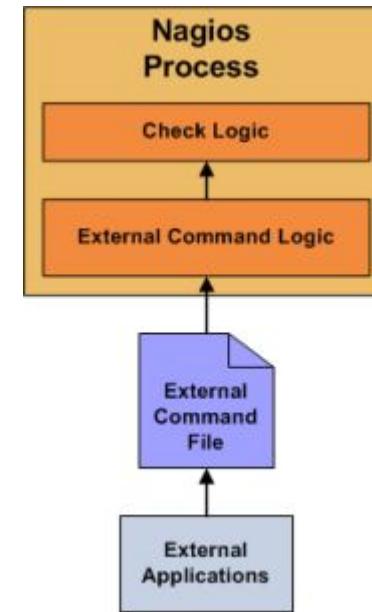
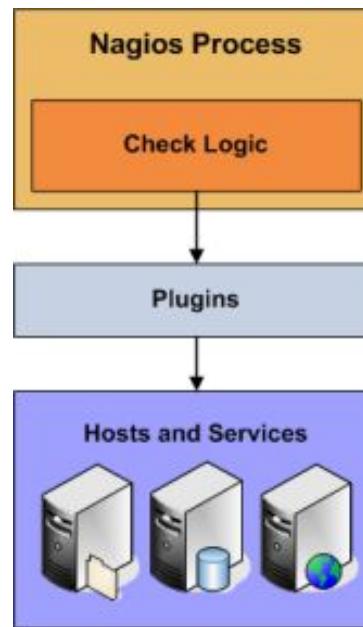
Agent: Usually daemons or services that must be places on the client to listen for connection coming from the NAgios Server.

Host Group: Group the hosts. (Ex: OS_Hosts, Oracle_Servers, Window_Servers)

Service Group: Grouping multiple services (similar). (Ex: Web_services, CPU_Load, Memory _Usage)

Active vs. Passive Checks

Active Checks	Passive Checks
Initiated by Nagios Server	Initiated by External Application
Result submitted to scheduler directly for processing	Result stored in external command file



Installing Nagios Core From Source

1. Visit the below link -

<https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html>

2. Select your Operating System and follow the instructions.

Ex- for kali linux choose Debian and follow the given instructions.

Directory Overview

/usr/local/nagios/etc - Config files (main + objects)

/usr/local/nagios/libexec - Plugins

/usr/local/nagios/var - Logs, status data

/usr/local/nagios/share - Web interface files

/etc/init.d/nagios - Service startup script

Host and Service Configuration

Nagios defines everything using text-based configuration files.

Key Config Files :

File	Role
nagios.cfg	Master config file
objects/commands.cfg	Command definitions
objects/contacts.cfg	Who gets alerts
objects/localhost.cfg	Default monitoring for local machine
objects/templates.cfg	Predefined “use” templates

Verify Plugin Directory

Nagios plugins are stored in:

```
cd /usr/local/nagios/libexec
```

```
ls
```

Access Service Detail Page

1. Open Nagios Web Interface → <http://localhost/nagios>
Or <http://<<ip>>/nagios>
2. Login as nagiosadmin
3. Navigate to: Current Status → Service Detail

It displays list of monitored services like PING, HTTP, SSH, Current Load, etc.

Verify HTTP Service Check

Look for the service labeled “HTTP” under the host “localhost”.

It should appear in a table with columns like:

Service, Status, Last Check, Duration, Attempt and Status Information.

Run the HTTP check manually:

```
/usr/local/nagios/libexec/check_http -H localhost
```

check_http

The check_http plugin is used in Nagios to verify the status of HTTP or HTTPS web services.

It checks things like:

- HTTP response code (200 OK, 404 Not Found, etc.)
- Response time
- SSL certificate expiry
- Response content

Syntax - /usr/local/nagios/libexec/check_http [options]

check_http

Basic HTTP check - /usr/local/nagios/libexec/check_http -H localhost

HTTPS check with SSL verification - /usr/local/nagios/libexec/check_http -H example.com -S

Check response time thresholds - /usr/local/nagios/libexec/check_http -H example.com -w 2 -c 5

Warns if response time >2s, critical if >5s.

Check for specific text on webpage - /usr/local/nagios/libexec/check_http -H example.com -u /index.html -s "Welcome"

Warns if “Welcome” not found in response.

SSL certificate expiry check -

/usr/local/nagios/libexec/check_http -H example.com -S -C 10

Warns if SSL certificate expires in less than 10 days.

Example Nagios Service Definition

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

```
define service{
    use generic-service
    host_name WebServer
    service_description HTTP Service
    check_command check_http!-H example.com -w 2 -c 5
}
```

Verify Nagios Configuration -

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

Expected Output-

Total Warning :0

Total Errors:0

Run Ping Check (verify host response)

check_ping is used by Nagios to verify network connectivity and latency to a host.

It sends ICMP Echo Requests (ping packets) and reports:

- Packet loss (%)
- Round-trip average time (ms)
- Status (OK / WARNING / CRITICAL) based on thresholds

Syntax -

```
/usr/local/nagios/libexec/check_ping -H <host> -w <warning> -c <critical> [-p packets] [-t timeout]
```

```
/usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w 100.0,20% -c 200.0,40%
```

```
/usr/local/nagios/libexec/check_ping -H ipv6.google.com -6 -w 200.0,20% -c 400.0,50%
```

Common Options and Arguments

Option	Description	Example
<code>-H <host></code>	Hostname or IP address to ping.	<code>-H 192.168.1.1</code>
<code>-w <warn></code>	Warning threshold in format <code>rta,pl%</code> where <code>rta</code> = round-trip average (ms), <code>pl</code> = packet loss (%).	<code>-w 100.0,20%</code>
<code>-c <crit></code>	Critical threshold in same format as <code>-w</code> .	<code>-c 200.0,50%</code>
<code>-p <packets></code>	Number of ICMP packets to send (default: 5).	<code>-p 3</code>
<code>-t <timeout></code>	Plugin timeout in seconds (default: 10).	<code>-t 5</code>
<code>-4</code>	Use IPv4 only.	<code>-4</code>
<code>-6</code>	Use IPv6 only.	<code>-6</code>
<code>-n</code>	Numeric output only (no DNS lookup).	<code>-n</code>

Example Nagios Service Definition

Services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  Ping Check  
    check_command        check_ping!100.0,20%!200.0,50%  
}
```

commands.cfg:

```
define command{  
    command_name    check_ping  
    command_line    /usr/local/nagios/libexec/check_ping -H  
'$HOSTADDRESS$' -w '$ARG1$' -c '$ARG2$'  
}
```

check_ssh

The check_ssh plugin verifies whether an SSH service (port 22 by default) is reachable and responding on a host.

It's useful for monitoring servers that require remote secure access.

Syntax - /usr/local/nagios/libexec/check_ssh <hostname or IP> [options]

Example -

Check SSH Service on Localhost

/usr/local/nagios/libexec/check_ssh localhost

Specify Timeout

/usr/local/nagios/libexec/check_ssh localhost -t 5

Meaning: The check will fail if no SSH response is received within 5 seconds.

Example Nagios Service Definition

Commands.cfg -

```
define command{  
    command_name    check_ssh  
    command_line     /usr/local/nagios/libexec/check_ssh '$HOSTADDRESS$'  
}
```

Services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  SSH Service  
    check_command        check_ssh  
}
```

Schedule Downtime for SSH Service

Step 1: Login to Nagios Web UI

Step 2: Go to “Service Detail”

Step 3: Select the SSH Service

Step 4: Click on “Schedule Downtime”

Step 5: Fill Downtime Details

Start Time, End Time, Duration, Fixed downtime

Click Commit.

Step 6: Verify Downtime

Reports → Scheduled Downtime

check_ftp

The check_ftp plugin checks the status of an FTP (File Transfer Protocol) service on a given host.

It verifies if:

- The FTP port (default 21) is open
- The server responds with a valid FTP greeting message
- Optionally tests login authentication if username/password are provided

Syntax - /usr/local/nagios/libexec/check_ftp -H <hostname> [options]

Example -

Check FTP Service on Localhost : /usr/local/nagios/libexec/check_ftp -H localhost

Anonymous Login Check : /usr/local/nagios/libexec/check_ftp -H localhost -A

FTP Authentication Test: /usr/local/nagios/libexec/check_ftp -H localhost -u nagios -p nagios123

Example Nagios Service Definition

Commands.cfg -

```
define command{  
    command_name    check_ftp  
    command_line     /usr/local/nagios/libexec/check_ftp -H '$HOSTADDRESS$'  
}
```

Services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  FTP Service  
    check_command        check_ftp  
}
```

check_dhcp

The check_dhcp plugin checks whether a DHCP server on the network is available and responding properly.

It sends a DHCPDISCOVER packet and waits for a DHCPOFFER response.

This helps verify:

- DHCP service availability
- Response time
- IP lease offers from DHCP servers

Syntax - /usr/local/nagios/libexec/check_dhcp [options]

Example -

Basic DHCP Check : sudo /usr/local/nagios/libexec/check_dhcp

Specify Interface : sudo /usr/local/nagios/libexec/check_dhcp -i eth0

Check Specific DHCP Server: sudo /usr/local/nagios/libexec/check_dhcp -s 192.168.1.1 -i eth0

Request a Specific IP Address: sudo /usr/local/nagios/libexec/check_dhcp -u 192.168.1.50 -i eth0

Example Nagios Service Definition

Commands.cfg -

```
define command{  
    command_name    check_dhcp  
    command_line     sudo /usr/local/nagios/libexec/check_dhcp -i '$ARG1$'  
}
```

Services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description   DHCP Service  
    check_command         check_dhcp!eth0  
}
```

Log History

To view Nagios log history, analyze previous monitoring results, and identify any warning or critical events generated by the monitoring plugins.

- Open the Nagios Web Interface
- In the left hand side bar navigate to Log History
- Understand the Log Sections

Event Log -

- Shows real-time and past system events, such as:
 - ❖ Service checks
 - ❖ Notifications sent
 - ❖ State changes (OK, WARNING, CRITICAL, UNKNOWN)
 - ❖ Nagios process start/stop events
- Identify Warnings and Critical Events
Look for entries where STATE = WARNING, CRITICAL
- Use Alert History Report

Color-coded table:

- OK
- WARNING
- CRITICAL
- UNKNOWN

Command: sudo tail -n 30 /usr/local/nagios/var/nagios.log

Check Disk Space

The `check_disk` plugin is used to monitor disk usage (space available, used percentage, and partitions) on local or remote systems. It alerts when the disk space usage exceeds the defined warning (-w) or critical (-c) thresholds.

Syntax - `/usr/local/nagios/libexec/check_disk -w <warn%> -c <crit%> [options]`

Example -

Check All Mounted Filesystems: `/usr/local/nagios/libexec/check_disk -w 80% -c 90%`

Check a Specific Partition: `/usr/local/nagios/libexec/check_disk -w 80% -c 90% -p /`

Check Multiple Partitions: `/usr/local/nagios/libexec/check_disk -w 85% -c 95% -p / -p /home`

Nagios Configuration

Define Command in commands.cfg -

```
define command{  
    command_name    check_disk  
    command_line    /usr/local/nagios/libexec/check_disk -w $ARG1$ -c  
$ARG2$ -p $ARG3$  
}
```

Define Service in services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  Disk Usage  
    check_command        check_disk!80%!90%!/
```

Check CPU Load

The check_load plugin checks the average system load (CPU utilization and process queue) over the past 1, 5, and 15 minutes. It helps identify if the server is under heavy CPU load or overloaded.

Syntax -

```
/usr/local/nagios/libexec/check_load -w  
<wload1>,<wload5>,<wload15> -c <cload1>,<cload5>,<cload15>
```

What is System Load?

- Load = number of processes waiting for CPU time.
- On a single-core system, a load of 1.00 means 100% utilization.
- On a 4-core CPU, a load of 4.00 means full utilization.

Example

Basic Check - /usr/local/nagios/libexec/check_load -w 5.0,4.0,3.0 -c 10.0,6.0,4.0

When System is Busy (Warning): /usr/local/nagios/libexec/check_load -w 0.5,0.4,0.3 -c 1.0,0.8,0.6

State	Description	Example Output
OK	Load below warning	OK - load average: 0.20, 0.15, 0.10
WARNING	Load between warning and critical	WARNING - load average: 1.2, 1.0, 0.9
CRITICAL	Load above critical	CRITICAL - load average: 3.5, 3.0, 2.8
UNKNOWN	Invalid input or plugin error	UNKNOWN - invalid parameters

Nagios Configuration

Commands.cfg -

```
define command{  
    command_name    check_load  
    command_line    /usr/local/nagios/libexec/check_load -w $ARG1$ -c  
$ARG2$  
}
```

Services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  CPU Load  
    check_command        check_load!5.0,4.0,3.0!10.0,6.0,4.0  
}
```

Check Memory Usage

The check_mem plugin checks the system memory (RAM) usage on the local or remote machine.

It alerts when memory utilization exceeds a specified warning (-w) or critical (-c) threshold.

Note: check_mem is not included by default in Nagios Core — it must be downloaded or installed manually.

Installation -

Step1 : Download

```
cd /usr/local/nagios/libexec
```

```
sudo wget
```

https://raw.githubusercontent.com/justintime/nagios-plugins/master/check_mem/check_mem.pl

Step 2: Make Executable

```
sudo chmod +x check_mem.pl
```

```
sudo mv check_mem.pl check_mem
```

Step 3: Verify

```
/usr/local/nagios/libexec/check_mem -w 80 -c 90
```

check_mem

Syntax - /usr/local/nagios/libexec/check_mem -w <warn%>
-c <crit%>

Example:

Basic Memory Check: /usr/local/nagios/libexec/check_mem
-w 80 -c 90

Show in MB: /usr/local/nagios/libexec/check_mem -w 80 -c
90 -f

Nagios Configuration

Commands.cfg:

```
define command{  
    command_name    check_mem  
    command_line     /usr/local/nagios/libexec/check_mem -w $ARG1$ -c  
$ARG2$  
}
```

services.cfg:

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  Memory Usage  
    check_command        check_mem!80!90  
}
```

SNMP Monitoring in Nagios

What is SNMP?

SNMP (Simple Network Management Protocol) is a standard protocol used to monitor and manage network devices such as:

Routers

Switches

Firewalls

Servers

Printers

It allows Nagios to query device performance data (CPU, memory, interface traffic, etc.) using SNMP agents.

How SNMP Works

Component	Role	Example
SNMP Manager	Requests data from agents (Nagios server)	check_snmp plugin
SNMP Agent	Runs on the monitored device	snmpd service
MIB (Management Information Base)	Database of monitored parameters	IF-MIB , HOST-RESOURCES-MIB
OID (Object Identifier)	Unique number identifying each monitored item	.1.3.6.1.2.1.1.3.0 (System uptime)

Enable SNMP on Linux (Agent Side)

Step 1: Install SNMP Agent

```
sudo apt install snmpd -y
```

Step 2: Edit Configuration

```
sudo nano /etc/snmp/snmpd.conf
```

Step 3: Modify for Read-Only Access

```
agentAddress udp:161,udp6:[::1]:161
rocommunity public default -V systemonly
```

Step 4: Restart Service

```
sudo systemctl restart snmpd
```

Step 5: Verify SNMP Agent

```
snmpwalk -v2c -c public localhost
```

Nagios SNMP Plugin

Nagios provides the check_snmp plugin for SNMP monitoring.

Location: /usr/local/nagios/libexec/check_snmp

Syntax - /usr/local/nagios/libexec/check_snmp -H <host> -o <OID> -C <community>
[options]

Example -

Check System Uptime : /usr/local/nagios/libexec/check_snmp -H localhost -o
.1.3.6.1.2.1.1.3.0 -C public

Check CPU Load: /usr/local/nagios/libexec/check_snmp -H localhost -o
.1.3.6.1.4.1.2021.10.1.3.1 -C public

Check Available RAM: /usr/local/nagios/libexec/check_snmp -H localhost -o
.1.3.6.1.4.1.2021.4.6.0 -C public

With Warning/Critical Thresholds: /usr/local/nagios/libexec/check_snmp -H localhost
-o .1.3.6.1.4.1.2021.10.1.3.1 -C public -w 2 -c 4

Plugin Return States

State	Description	Example Output
OK	Value within limits	SNMP OK - CPU 0.3
WARNING	Exceeds warning limit	SNMP WARNING - Load 2.5
CRITICAL	Exceeds critical limit	SNMP CRITICAL - Memory 95%
UNKNOWN	Communication issue	SNMP UNKNOWN - Timeout

Nagios Configuration

Commands.cfg -

```
define command{  
    command_name    check_snmp  
    command_line    /usr/local/nagios/libexec/check_snmp -H  
'$HOSTADDRESS$' -o '$ARG1$' -C public -w '$ARG2$' -c '$ARG3$'  
}
```

Services.cfg -

```
define service{  
    use                  generic-service  
    host_name            localhost  
    service_description  SNMP Uptime  
    check_command        check_snmp!.1.3.6.1.2.1.1.3.0!100!200  
}
```

Thank You