

# Advanced DevOps

## Lab Experiment 10

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

Nagios. **Steps:**

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

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

```
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-10-06 11:51:46 UTC; 12min ago
     Docs: https://www.nagios.org/documentation
  Main PID: 89956 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 3.2M
      CPU: 221ms
   CGroup: /system.slice/nagios.service
           └─89956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           └─89957 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─89958 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─89959 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─89960 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─89961 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

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.

Provide it with the same security group as the Nagios Host and name it ‘linux-client’ alongside the host.

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

<

1

>

<input type="checkbox"/>	Name <div></div>	Instance ID	Instance state <div></div>	Instan
<input type="checkbox"/>	nagios	i-0a4d4aa2e0db0e809	<div>Running</div> <div></div> <div></div>	t2.micr
<input type="checkbox"/>	nagios-host	i-0fe5cf179a1f684e5	<div>Running</div> <div></div> <div></div>	t2.micr

<

1

>

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-172-31-38-150 nagios-plugins-2.4.11]$ ps -ef | grep nagios
nagios      89956      1  0 11:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      89957    89956  0 11:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      89958    89956  0 11:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      89959    89956  0 11:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      89960    89956  0 11:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      89961    89956  0 11:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    91366    2723  0 12:12 pts/0    00:00:00 grep --color=auto nagios
ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$
```

4. Become a root user and create 2 folders

```
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir
```

```
/nagios.cfg
ec2-user    91366    2723  0 12:12 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo su
[root@ip-172-31-38-150 nagios-plugins-2.4.11]# mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-38-150 nagios-plugins-2.4.11]#
```

/usr/local/nagios/etc/objects/monitorhosts/linuxhosts 5. Copy

```
[root@ip-172-31-38-150 nagios-plugins-2.4.11]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-38-150 nagios-plugins-2.4.11]# nano
[root@ip-172-31-38-150 nagios-plugins-2.4.11]#
```

the sample localhost.cfg file to linuxhost folder

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

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

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

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

## CLIENT.

```
#####
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
# NOTE: This config file is intended to serve as an *extremely* simple
#       example of how you can create configuration entries to monitor
#       the local (Linux) machine.
#
#####

#####

# 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 tha
                                                ; in (or inherited by) the linux-server host template

    defhost_name       localhost
    alias              localhost
    address            34.229.230.178
}

#####

# HOST GROUP DEFINITION
#
#####

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name     linux-servers                ; The name of the hostgroup
    ^G Alias           ^C Write Out Linux Servers    ^K Cut; Long name of the group ^L Location ^M-U Undo
    ^X Emembersk       ^R Read File localhostace     ^U Past; Comma separated list of hosts that belong to this g
    GroupCopy
}
```

Change hostgroup\_name under hostgroup to linux-servers1

```

# Define a service to "ping" the local machine

define service {
    use                local-service        ; Name of service template to use
    host_name          localhost
    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
# < 10% free space on partition. if < 20% free, critical if

define service {
    use                local-service        ; Name of service template to use
    host_name          localhost
    service_description Root Partition
    check_command       check_local_disk!20%!10%!/
}

# Define a service to check the number of currently logged in
# users on the local machine. Warning if > 20 users, critical
# if > 50 users.

define service {
    use                local-service        ; Name of service template to use
    host_name          localhost
    service_description Current Users
    check_command       check_local_users!20!50
}

# Define a service to check the number of currently running procs
# on the local machine. Warning if > 250 processes, critical if

```

Everywhere else on the file, change the hostname to linuxserver instead of localhost.

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/
```

```

# NAGIOS.CFG - Sample Main Config File for Nagios 4.4.0
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#####

# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log


# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts
# Definitions for monitoring the local (Linux) host
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

```

Read 13^T Execute ^C Location M-U Und  
 ^X Exit Mark ^R Read File ^N Replace ^U Paste ^J Justify ^/ Go To Line M-E Rec

8. Verify the configuration files

```

Running pre-flight check on configuration data...

Checking objects...
    Checked 8 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

```

You are good to go if there are no errors.

## 9. Restart the nagios service

service nagios restart

```

Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-38-150 ec2-user]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-10-06 12:55:30 UTC; 32s ago
     Docs: https://www.nagios.org/documentation
   Process: 94331 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
code=exited, status=0/SUCCESS
   Process: 94332 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Main PID: 94333 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 1.8M
      CPU: 18ms
   CGroup: /system.slice/nagios.service
           └─94333 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             └─94334 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─94335 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─94336 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                   └─94337 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                     └─94338 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 12:55:30 ip-172-31-38-150.ec2.internal nagios[94333]: wproc: Registry request: name=Core Worker 94334;pid=94334
Oct 06 12:55:30 ip-172-31-38-150.ec2.internal nagios[94333]: Warning: Duplicate definition found for service 'HTTP' on host
Oct 06 12:55:30 ip-172-31-38-150.ec2.internal nagios[94333]: Warning: Duplicate definition found for service 'SSH' on host
Oct 06 12:55:30 ip-172-31-38-150.ec2.internal nagios[94333]: Warning: Duplicate definition found for service 'Swap Usage' on host
Oct 06 12:55:30 ip-172-31-38-150.ec2.internal nagios[94333]: Warning: Duplicate definition found

```

Now it is time to switch to the client machine.

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

```
PS C:\Users\91799> cd C:\Users\91799\Desktop\awsKey
PS C:\Users\91799\Desktop\awsKey> ssh -i "pratik.pem" ec2-user@ec2-34-229-230-178.compute-1.amazonaws.com
The authenticity of host 'ec2-34-229-230-178.compute-1.amazonaws.com (34.229.230.178)' can't be established.
ED25519 key fingerprint is SHA256:xL/Zr/DslzhcBtLGSAAtcu1Q4LLZ/zEo+L0Yg3fHs4Rc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-229-230-178.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```

```
      #_
    ~\_ #####_          Amazon Linux 2023
   ~~ \_#####\
   ~~  \####|
   ~~   \#/___ _ https://aws.amazon.com/linux/amazon-linux-2023
   ~~     V~' '->
       ~~~
        ~_. _ _/_/_/
         _/_/_/_/_/_/
          _/m/'
```

```
[ec2-user@ip-172-31-45-49 ~]$
```

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

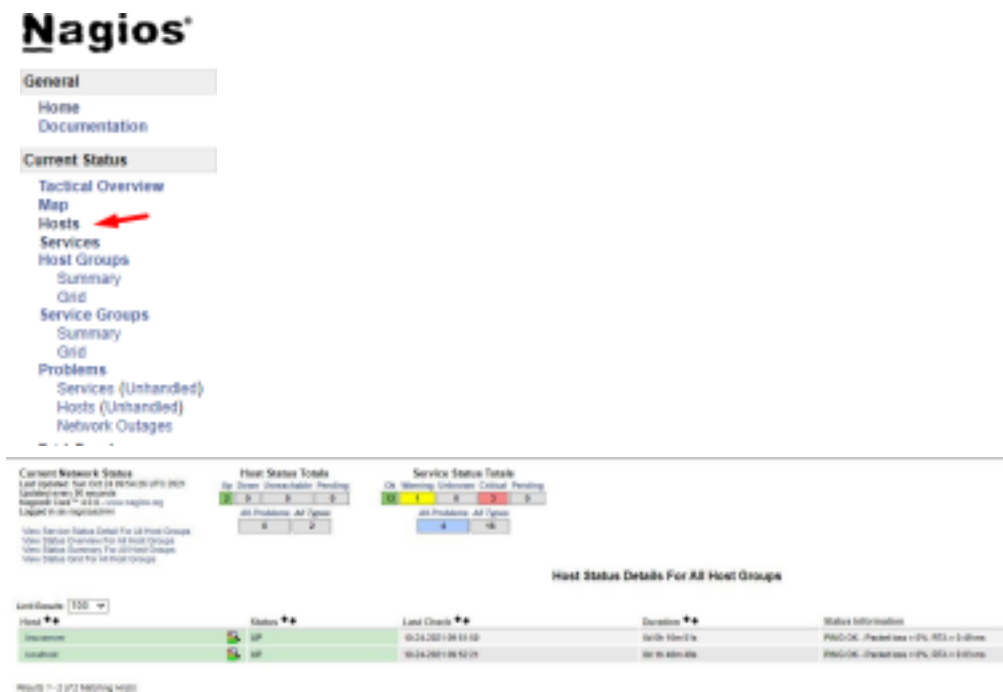
```
ubuntu@ip-172-31-32-54: ~  
GNU nano 4.8  
# file to allow only the specified host to connect  
# you are running this daemon on.  
#  
# NOTE: This option is ignored if NRPE is running  
allowed_hosts=127.0.0.1,13.233.227.254
```

13. Restart the NRPE server

```
sudo systemctl restart nagios-nrpe-server
```

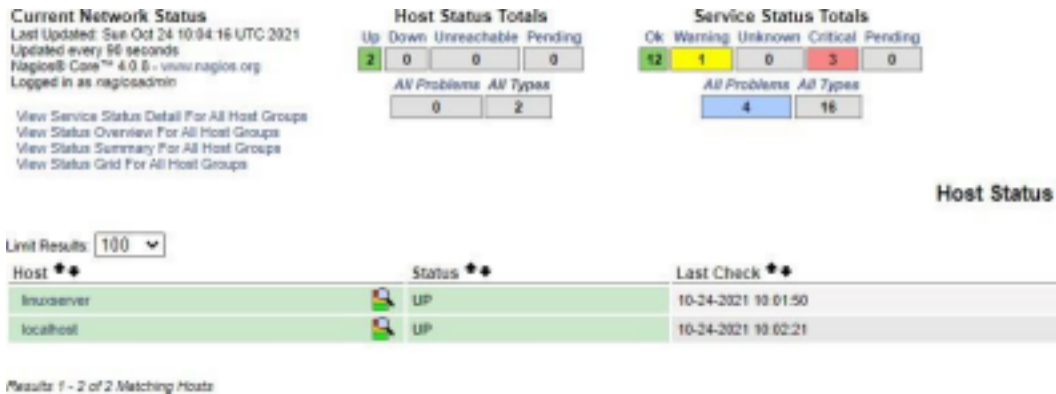
14. Now, check your nagios dashboard and you'll see a new host being added.

Click on Hosts.



Click on linuxserver to see the host details





You can click Services to see all services and ports being monitored.

Limit Results: 100						
Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	10-24-2021 10:02:35	0d 0h 25m 27s	1/4	OK - load average: 0.08, 6.04, 8.86
	Current Users	OK	10-24-2021 09:58:11	0d 0h 26m 48s	1/4	LOGGED OK - 1 users currently logged in
	HTTP	CRITICAL	10-24-2021 10:01:50	0d 0h 22m 12s	4/4	connect to address 13.254.58.2 and port 80: Conn
	PING	OK	10-24-2021 09:58:36	0d 0h 23m 34s	1/4	PING OK - Packet loss = 3%, RTT = 8.51 ms
	Root Partition	OK	10-24-2021 10:00:06	0d 0h 23m 57s	1/4	DISK OK - free space: 14202 MB (76% used=48%)
	SSH	OK	10-24-2021 10:00:45	0d 0h 22m 18s	1/4	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu1.2 (2
	Swap Usage	CRITICAL	10-24-2021 09:59:29	0d 0h 21m 42s	4/4	SWAP CRITICAL - 0% free (0 MB out of 8 MB) - 5
	Total Processes	OK	10-24-2021 10:01:56	0d 0h 21m 4s	1/4	PROCS OK: 27 processes with STATE = R520T

As you can see, we have our linuxserver up and running. It is showing critical status on HTTP due to permission errors and swap because there is no partition created.

**In this case, we have monitored -**

**Servers: 1 linux server**

**Services: swap**

**Ports: 22, 80 (ssh, http)**

**Processes: User status, Current load, total processes, root partition, etc.**

## Recommended Cleanup

- Terminate both of your EC-2 instances to avoid charges.
- Delete the security group if you created a new one (it won't affect your bill, you may avoid it)

## Conclusion:

Nagios is a widely used open-source monitoring tool designed to monitor systems, networks, and infrastructure. It can alert administrators when things go wrong and notify them of recovery. The monitoring includes various aspects, such as services, system performance, port status, and network health.