Adv DevOps Practical 10

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

Theory:

Port and Service Monitoring

Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

Windows/Linux Server Monitoring

Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

Monitoring Using Nagios:

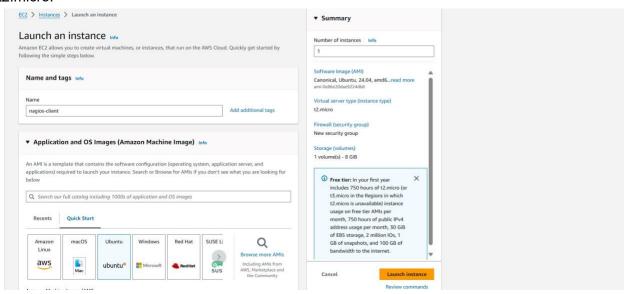
Step 1: To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host). **sudo systemctl status nagios**

```
[ec2-user@ip-Ti72-31-91-91 -]$ sudo systemctl status nagios

* napios service - Nagios Core 4.5.5
Loaded: Loaded (/usr/lib/system/system/nagios.service; enabled; preset: disabled)
Active: active (running) since Sun 2024-09-29 16:18:08 UT; 21min ago
Docs: https://www.nagios.org/documentation
Process: 1942 ExecStatt=fusr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Process: 1944 ExecStatt=fusr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Main PID: 1946 (nagios)
Tasks: 8 (limit: 1112)
Memory: 7.7M
CPU: 387ms
CGroup: /system.slice/nagios.service
-1946 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
-1947 /usr/local/nagios/bin/nagios -d /usr/local/nagios/var/rw/nagios.qh
-1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/scgf
-1946 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
-1956 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/scfg
-2888 /usr/local/nagios/bin/nagios --worker /usr/local/nagios.cfg
-2899 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios.cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios/cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios.cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios.cfg
-2899 /usr/local/nagios/bin/nagios--worker /usr/local/nagios.cfg
-2899 /usr/local/nagios/bin/nagios--worke
```

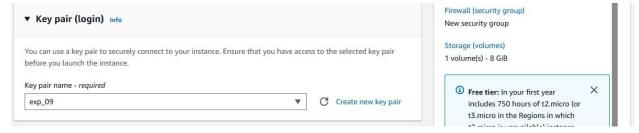
You can now proceed if you get the above message/output.

Step 2: Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

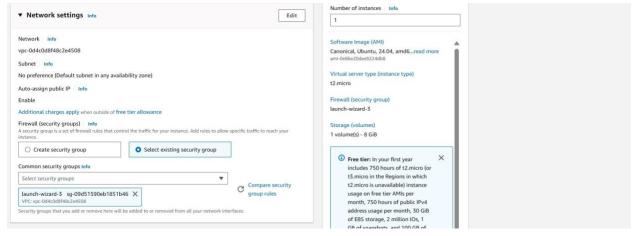


For Key pair: Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .

Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.

```
PS C:\Users\ MUSKAANNN \( \to \) ssh \( -i \) "Downloads/exp_09 \( \text{pem} \) ubuntu@ec2-44-286-245-149 \compute-1 \text{.amazonaws.com} \)
The authenticity of host 'ec2-44-286-245-149 \compute-1 \text{.amazonaws.com} \( (44 \) 286 \cdot 245 \cdot 149 \) ' can't be established.

ED25519 key fingerprint is ShA256:DT+AA+mkcydh3kOJ2vEpm4ZsA6FL+LM4m1QSImddAHg.
This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-286-245-149 \compute-1 \text{.amazonaws.com'} \( (ED25519) \) to the list of known hosts.

Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

* Documentation: https://help.ubuntu.com

* Management: https://landscape.canonical.com

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo_root" for details.

ubuntu@ip-172-31-92-146:~$
```

Now perform all the commands on the Nagios-host till step 10

Step 4: Now on the server Nagios-host run the following command.

ps -ef | grep nagios

```
[ec2-user@ip-172-3i-91-91 ~]$ ps -ef | grep nagios
nagios 1946 1 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 1947 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1948 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1949 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1950 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1956 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --d /usr/local/nagios/car/rw/nagios.qh
nagios 1950 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --d /usr/local/nagios/etc/nagios.cfg
root 3090 3055 0 16:40 pts/0 00:00:00 sudo systemctl status nagios
root 3093 3092 0 16:40 pts/1 00:00:00 sudo systemctl status nagios
ce2-user 3014 3890 0 16:59 pts/2 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-01 -]$
```

Step 5: Now Become root user and create root directories.

sudo su

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

```
[ec2-user@ip-172-31-91-91 ~]$ sudo su [root@ip-172-31-91-91 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts [root@ip-172-31-91-91 ec2-user]# |
```

Step 6: Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

cp /usr/local/nagios/etc/objects/localhost.cfg

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

[root@ip-172-31-91-91 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg [root@ip-172-31-91-91 ec2-user]# |

Step 7:Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

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

Change <u>hostname</u> to **linuxserver**.

Change address to the public IP of your Linux client.

Set hostgroup name to linux-servers1.

Step 8: Now update the Nagios config file .Add the following line in the file. Line to add : > nano /usr/local/nagios/etc/nagios.cfg

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

Step 9: Now Verify the configuration files by running the following commands.

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

```
Inning took okay - No serious problems were detected during the pre-tight check

Irook@ip-172-51-91-91 ec2-uscr)# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Running pre-flight check on configuration data...

Checking objects...

Checked 8 services.

Checked 8 services.

Checked 9 service groups.

Checked 1 contacts.

Checked 1 contacts.

Checked 1 contacts.

Checked 1 contact groups.

Checked 2 time periods.

Checked 3 service escalations.

Checked 8 service escalations.

Checked 9 service dependencies

Checked 9 host escalations.

Checked 9 host dependencies

Checked 9 host dependencies

Checked 5 host dependencies

Checked 5 hosessave communists...

Checking global event handlers...

Checking siosessive compulsive processor commands...

Checking misc settings...

Jotal Warnings: 0

Total Warnings: 0

Things look okay - No serious problems were detected during the pre-flight check
```

Step 10: Now restart the services of nagios by running the following command.

service nagios restart

```
[root@ip-172-31-91-91 ec2-user]# service nagios restart

Restarting nagios (via systemctl):

[root@ip-172-31-91-91 ec2-user]# |
```

Step 11: Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

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

```
ubuntu@ip-17-31-02-106:-$ sudo apt update -y sudo apt install pc- y sudo apt install pc- y sudo apt install yc- y sudo apt install -y nagios-nrpe-server nagios-plugins

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [126 kB] 
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB] 
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB] 
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB] 
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe and60 Packages [15 8 MB] 
Get:6 http://us-cast-1.ec2.archive.ubuntu.com/ubuntu noble/universe and60 Packages [180 kB] 
Get:6 http://security.ubuntu.com/ubuntu noble-security/main and60 Packages [180 kB] 
Get:6 http://security.ubuntu.com/ubuntu noble-security/main and60 c-n-f Metadata [180 kB] 
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe and60 Packages [172 kB] 
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe and60 Packages [172 kB] 
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe and60 Packages [172 kB] 
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe and60 Packages [183 kB] 
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe and60 Packages [183 kB] 
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe and60 Packages [183 kB] 
Get:15 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:16 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:17 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:18 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:19 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:19 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:20 http://security.ubuntu.com/ubuntu noble-security/inverse and60 Packages [183 kB] 
Get:2
```

Step 12: Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host IP address. **sudo nano /etc/nagios/nrpe.cfg**

```
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

# NOU can real closed 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,::1,34.207.68.187

# 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.

# **** ENABLING THIS OPTION IS A SECURITY RISK! ****
```

Step 13: Now restart the NRPE server by this command.

sudo systemctl restart nagios-nrpe-server

```
0 upgraded, 0 newly installed, 0 to remove and 139 not upgraded.
ubuntu@ip-172-31-92-146:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-92-146:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-92-146:~$
```

Step 14: Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

sudo systemctl status nagios

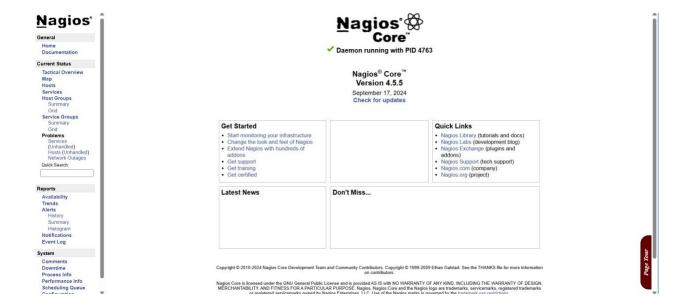
sudo systemctl status httpd

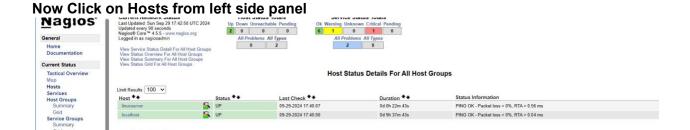
sudo systemctl start httpd sudo systemctl enable httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-91-91 ~]$ |
```

Step 15: Now to check Nagios dashboard go to http://<nagios host ip>/nagios Eg. http://34.207.68.187/nagios

Enter username as nagiosadmin and password which you set in Exp 9.





Conclusion:

Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search: Results 1 - 2 of 2 Matching Hosts

Grid Problems

In this practical, we set up a Nagios host and client to monitor services and server performance on both Linux and Windows servers. We configured Nagios on an Amazon Linux machine to monitor critical services like HTTP, SSH, and system resources, ensuring their availability and health. By creating and configuring a new EC2 instance as the Nagios client, we enabled seamless communication between the client and host for efficient service monitoring. This setup helps ensure uptime and quick detection of issues across the infrastructure.