## **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)

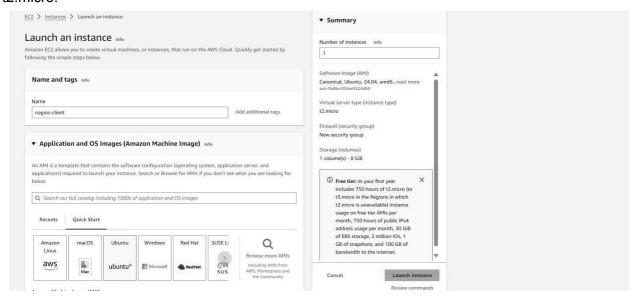
#### **Monitoring Using Nagios:**

**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-172-31-91-91 -]$ sudo systemtl status nagios
e nagios.service - Nagios Core 4.5.5
Loaded: loaded (Loayrllab/system/nagios.service; enabled; preset; disabled)
Active: active (running) since Sun 2024-09-29 16:18:08 UTC; 21min ago
Docs: https://www.nagios.org/documentation
Process: 1942 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Process: 1942 ExecStartPre=/usr/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 --w orker /usr/local/nagios/var/rm/nagios.qh
-1949 /usr/local/nagios/bin/nagios --w orker /usr/local/nagios/var/rm/nagios.qh
-1949 /usr/local/nagios/bin/nagios --w orker /usr/local/nagios/var/rm/nagios.qh
-1949 /usr/local/nagios/bin/nagios --w orker /usr/local/nagios/var/rm/nagios.qh
-1956 /usr/local/nagios/bin/nagios --w orker /usr/local/nagios/var/rm/nagios.gh
-1956 /usr/local/nagios/bin/nagios --w orker /usr/local/nagios/var/rm/nagios.gh
-1956 /usr/local/nagios/bin/nagios --w orker /
```

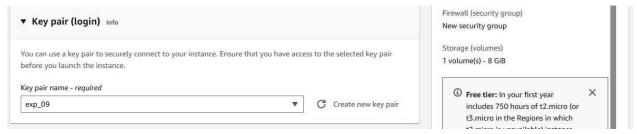
You can now proceed if you get the above message/output.

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

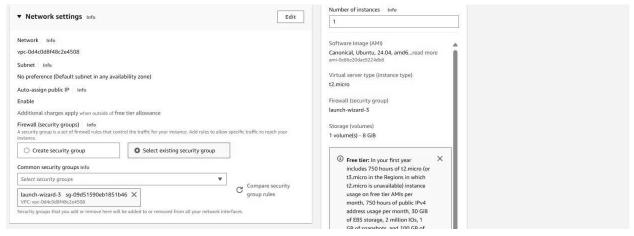


**For Key pair :** Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

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 i> ssh -i "Downloads/exp_09.pem" ubuntu@ec2-44-206-245-149.compute-1.amazonaws.com
The authenticity of host 'ec2-44-206-245-149.compute-1.amazonaws.com (44.206.245.149)' can't be established.
ED25519 key fingerprint is SHA256:DT+AA+mkcydh3kOJ2vEpm4ZsA6F+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 'ec2-44-206-245-149.compute-1.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

* Manadement: https://landscabe.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@pp-172-31-92-146:-$ |
```

## Now perform all the commands on the Nagios-host till step 10 Stop 4: Now on the server Nagios host run the following command

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

### ps -ef | grep nagios

#### **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
[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 <u>hostgroup name</u> 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

```
Inamps took okay - mo serious problems were detected during the pre-tight check
[root@ip-17-31-91-91 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 1999-2089 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Running pre-flight check on configuration data...

Checking objects...
Checked 8 services.
Checked 2 hosts.
Checked 2 hosts.
Checked 1 contacts.
Checked 1 contacts.
Checked 1 contacts.
Checked 1 contacts.
Checked 2 time periods.
Checked 2 time periods.
Checked 3 time periods.
Checked 6 service escalations.
Checked 8 service descalations.
Checked 9 host secalations.
Checked 9 host secalations.
Checked 9 host dependencies
Checked 6 host dependencies
Checked 6 host dependencies
Checked 6 host dependencies
Checked 7 host dependencies
Checked 7 host dependencies
Checked 8 host dependencies
Checked 9 host dependencies
Checking global event handlers...

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

```
Inings took oray - No serious problems were detected during the pre-right thetk
[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

```
subontWisp-172-31-92-146:-$ sudo apt update -y sudo apt install yec -y sudo apt install -y nagios-nrpe-server nagios-plugins

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-mydates 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-security inRelease [126 kB] 
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-security/main and64 Packages [13.6 MB] 
Get:6:7 http://security.ubuntu.com/ubuntu noble-security/main and64 Packages [388 kB] 
Get:8 http://security.ubuntu.com/ubuntu noble-security/main and64 c-n-f Metadata [1560 B] 
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [272 kB] 
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [272 kB] 
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [272 kB] 
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [278 kB] 
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [278 kB] 
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [278 kB] 
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [278 kB] 
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe and64 packages [28] 
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe and64 Genoments [852 B] 
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe and64 Genoments [868 B] 
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe and64 Genoments [868 B] 
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe and64 Genoments [868 B] 
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe and64 Genoments [868 B] 
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe and64
```

# **Step 12:** Open <u>nrpe.cfg file to make changes.Under allowed\_hosts</u>, add your <u>nagios host IP address</u>. **sudo nano /etc/nagios/nrpe.cfg**

```
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_group=nagios

# 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,::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

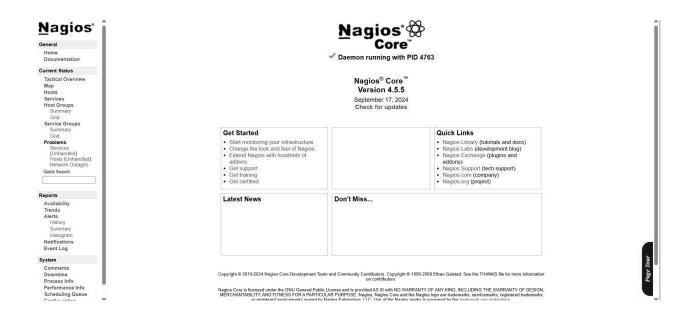
#### sudo systemctl status httpd

### sudo systemctl start httpd sudo systemctl enable httpd

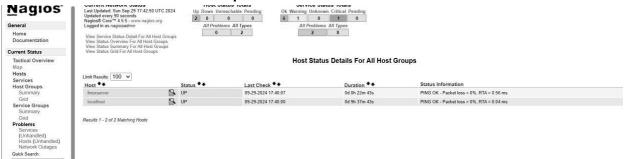
```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-91-91 ~]$ |
```

# 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.



#### Now Click on Hosts from left side panel



#### **Conclusion:**

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.