Sairam Konar D15C 27 2024-25

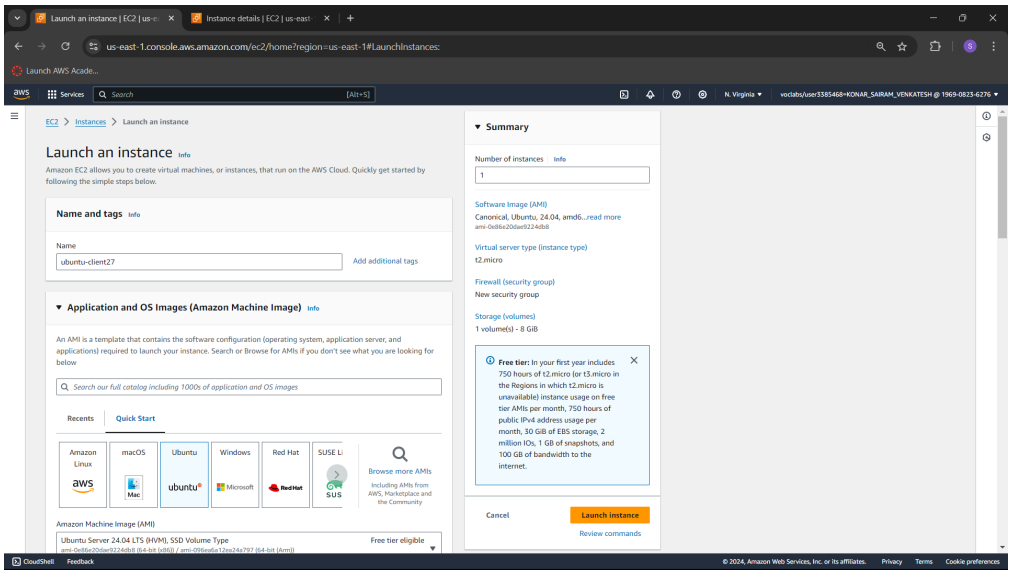
Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Prerequisites:

1) An Amazon Linux instance with nagios already set up.

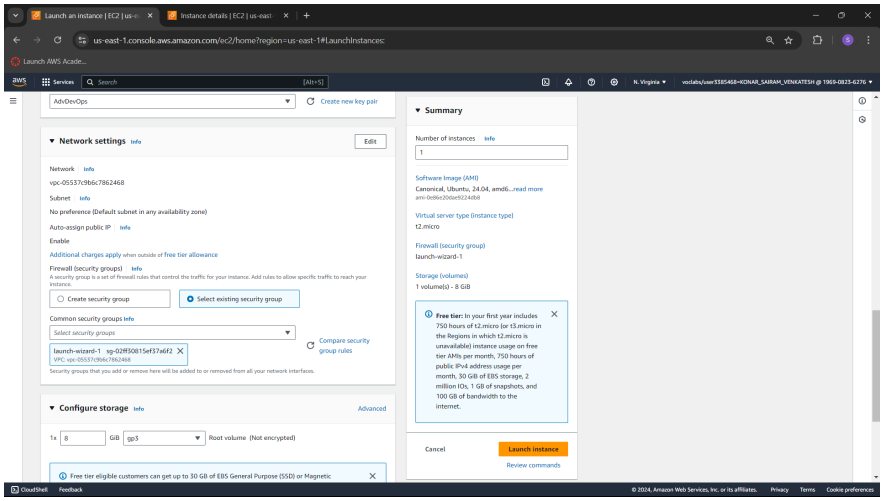
Step 1: Set up ubuntu instance

1) Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.



Select The OS Image as Ubuntu.

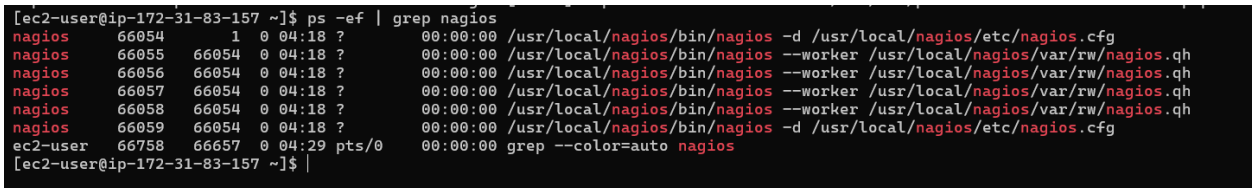
2) Make sure to select the same private key that you created for the Amazon Linux instance. Also select the same security group as you created for the Linux instance.



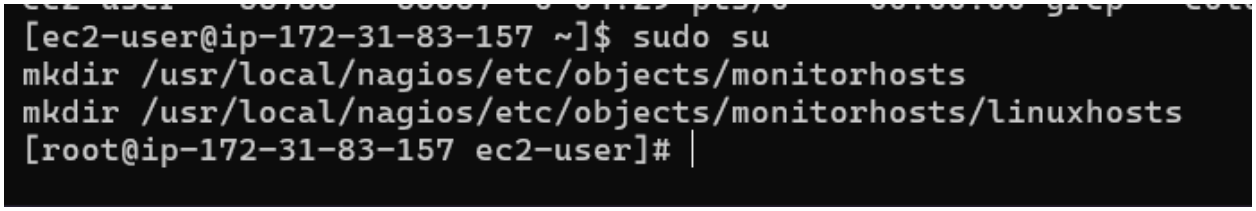
3) Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect. Click on SSH client. Copy the example command. Now, we have to connect our local OS terminal to the instance using SSH. For this, open terminal wher the private key file is located (.pem). Paste the copied SSH command and run it.

Step 2: Execute the following on Nagios Host machine (Linux)

1) We need to verify whether the nagios service is running or not. Fo that, run this command.
   ps -ef | grep nagios



2) Now, make yourself as the root user, and create a folder with the path '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts'
   sudo su
   mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts



3) We need to create a config file in this folder. So, copy the contents of the existing localhost config to the new file 'linuxserver.cfg'.
   cp /usr/local/nagios/etc/objects/localhost.cfg
   /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

4) We need to make some changes in this config file. Open it using nano editor.
   nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

   Change **hostname** and **alias** to **linuxserver**
   Change address to **public ip address of client instance** (Ubuntu instance)

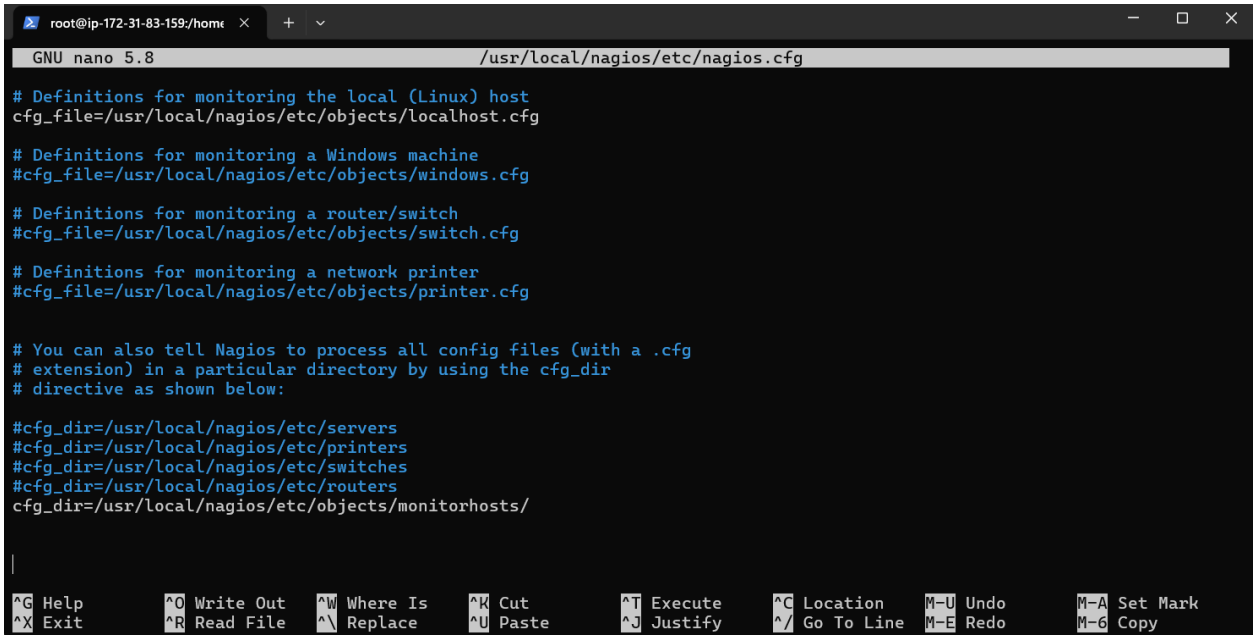
```
# Define a host for the local machine

define host{
        use                     linux-server            ; Name of host temp>
                                                        ; This host definit>
                                                        ; in (or inherited >
        host_name               linuxserver
        alias                   linuxserver
        address                 32.226.136.73
        }
```

Change hostgroup_name to **linux-servers1**

```
define hostgroup{
        hostgroup_name  linux-servers1 ; The name of the hostgroup
        alias           Linux Servers ; Long name of the group
        members         localhost     ; Comma separated list of hosts that >
        }
```

Change the **occurrences of hostname** further in the document from **localhost** to **linuxserver**

5) Now, we need to edit the nagios configuration file to add this directory.
   nano /usr/local/nagios/etc/nagios.cfg
   Run this command and add the following line
   cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/



6) Now we verify the configuration files.
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```
[root@ip-172-31-83-157 ec2-user]# /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

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
        Checked 16 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-172-31-83-157 ec2-user]#
```

7) Once the files are verified, we need to restart the server.
service nagios restart

```
[root@ip-172-31-83-159 nagios-plugins-2.0.3]# service nagios restart
Restarting nagios (via systemctl):                         [  OK  ]
[root@ip-172-31-83-159 nagios-plugins-2.0.3]#
```

Step 3: Execute the following on Nagios Client machine (Ubuntu)

1) First, we check for any new updates, then we install gcc, nagios nrpe server and nagios plugins.
   sudo apt update -y
   sudo apt install gcc -y
   sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-81-89:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates 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/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

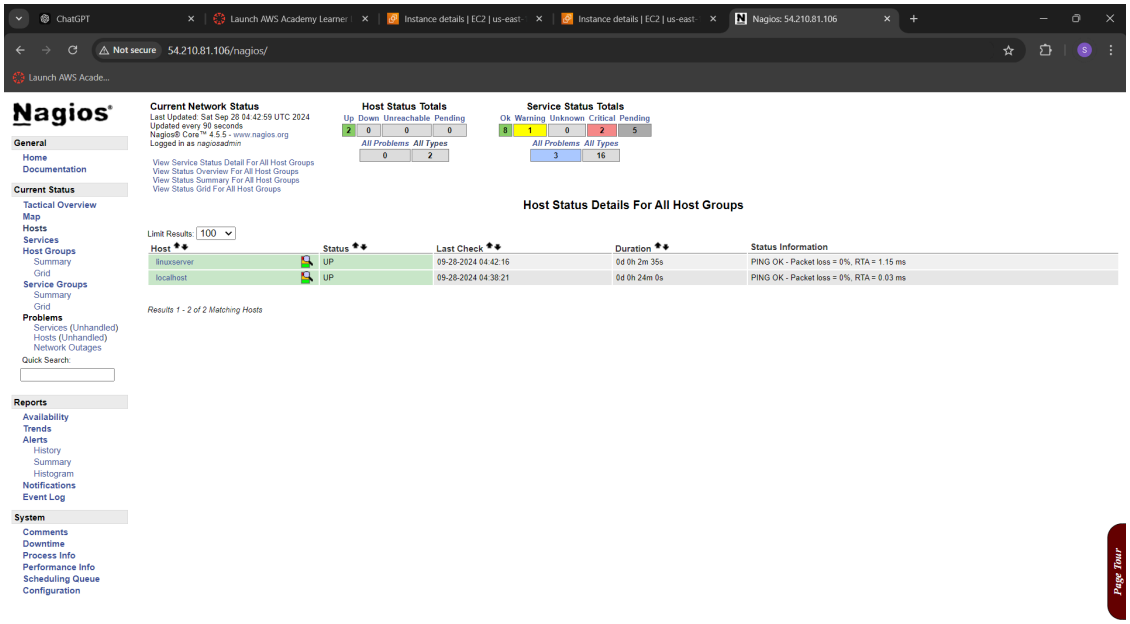
```
Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #4: sshd[1495,1569]
 ubuntu @ user manager service: systemd[1500]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-81-89:~$
```

2) We need to add the public IP address of our host Nagios machine (Linux) to the nrpe configuration file.
sudo nano /etc/nagios/nrpe.cfg
Under allowed_hosts, add the nagios host ip address (public)

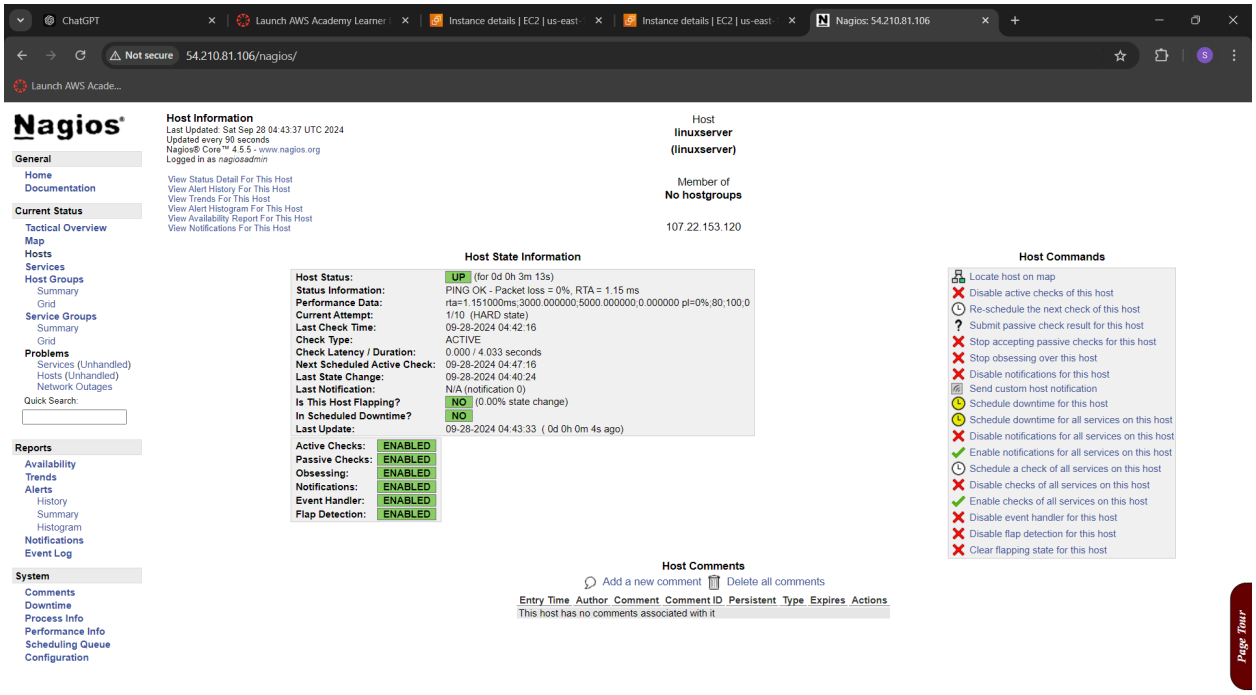
```
  GNU nano 7.2                                  /etc/nagios/nrpe.cfg *
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_user=nagios



# 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=127.0.0.1,54.210.81.106



# 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.

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was
```

Step 4: Check the Nagios Dashboard

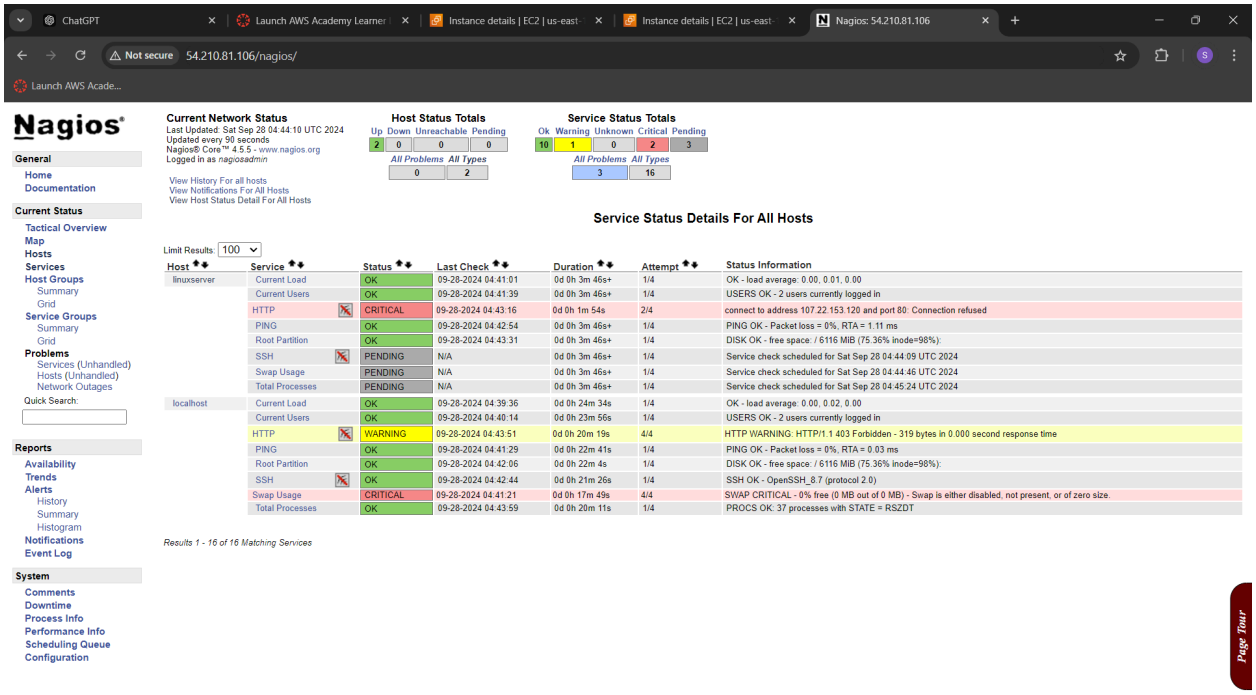
1) Go to Nagios dashboard, click on hosts.
Here, we can see that the linuxserver is also added as a host.



2) Click on linuxserver. Here, we can check all the information about linuxserver host.



3) Click on services. Here we an see all the services that are being monitored by linuxserver.



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.

Conclusion:

In this experiment, we learned to perform port service monitoring and server monitoring using Nagios. For this, we need the Linux instance used to host the Nagios dashboard and server. Also, we would need an Ubuntu instance which would be linked to a second host. We need to set up some configurations on the Linux instance and add the IP address of the Ubuntu instance. After that, we need to make the same initial setup on the ubuntu instance as the linu instance. Add the Ip address of linux instance in allowed hosts. After restarting the NRPE server, we can see the ‘linuxserver’ host added.