Name : Sandeshkumar.M.Yadav **Div : D15C** **Roll : 61**

Experiment No:10

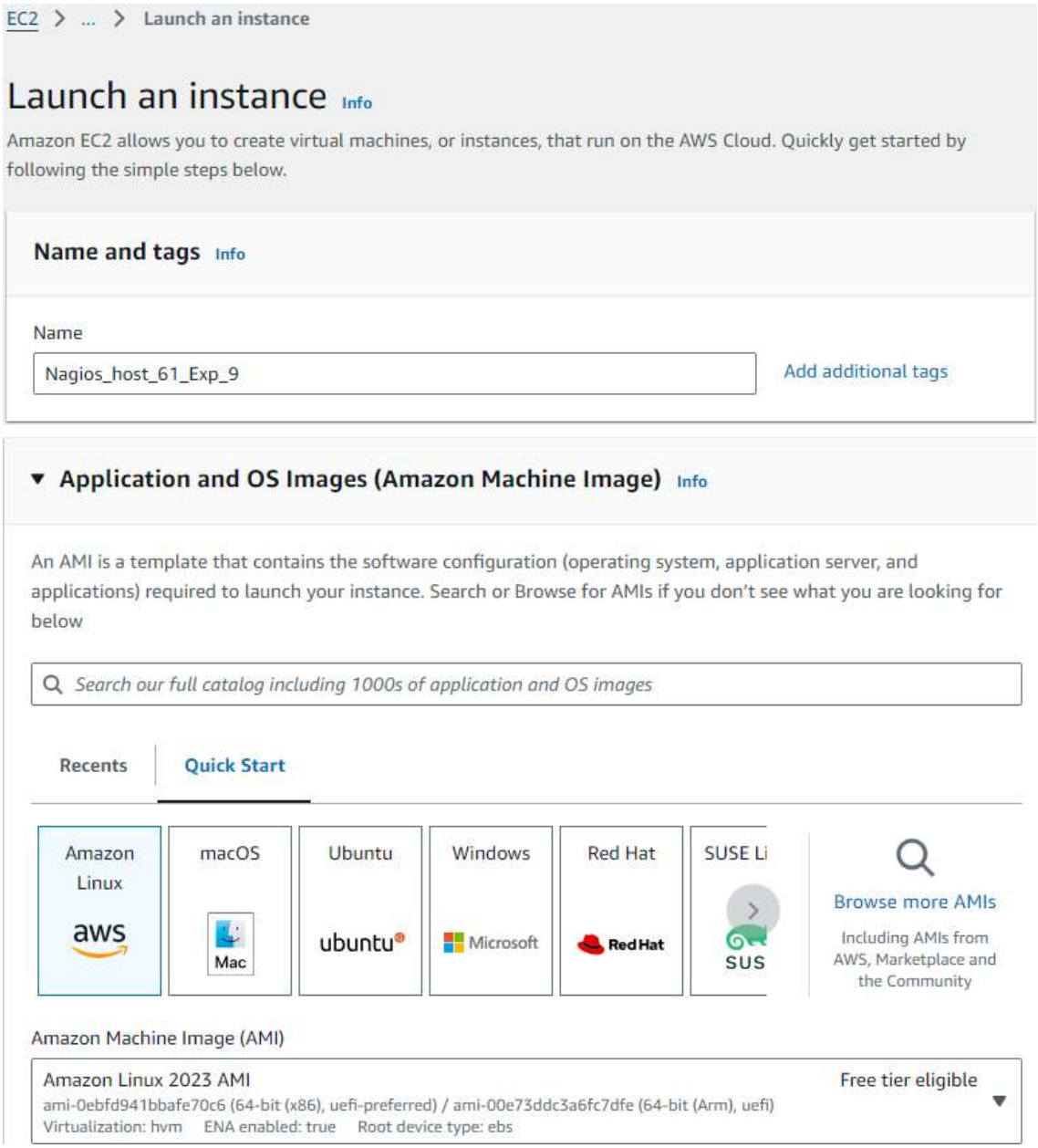
AIM: To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

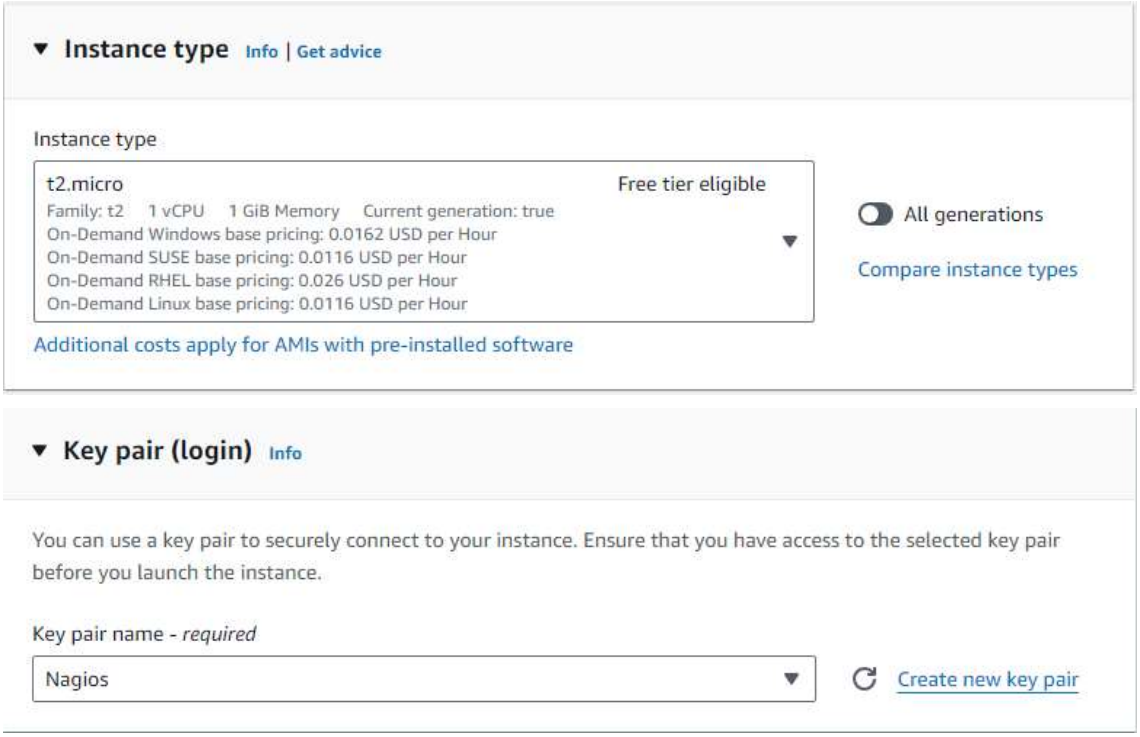
PREREQUISITES : We should have an Amazon Linux instance with nagios already set up.

Step 1: Set up ubuntu instance

1) Log in to your AWS account. Look for EC2 in the services menu. Open the interface and select Create Instance.

2) Ensure that you choose the same private key you created for the Amazon Linux instance. Additionally, select the same security group that you configured for the Linux instance.





3) Now return to the instances screen. Click on the instance ID of your instance, then select Connect. Click on SSH client and copy the example command. Next, we need to connect our local OS terminal to the instance using SSH. To do this, open the terminal where the private key file (.pem) is stored. Paste the copied SSH command and execute it.

All ports are open to all IPv4 addresses in your security group
All ports are currently open to all IPv4 addresses, indicated by **All** and **0.0.0.0/0** in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. Learn more.

Instance ID
i-02fd1850ce480fe74 (Nagios_host_61_Exp_9)

Connection Type

- (selected) Connect using EC2 Instance Connect
  Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
- Connect using EC2 Instance Connect Endpoint
  Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

- (selected) Public IPv4 address
  98.83.138.8
- IPv6 address
  –

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect

Step 2: On Nagios Host machine (Linux) execute the following which we have already created as a prerequisites:

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

```
[root@ip-172-31-35-58 ~]# ps -ef | grep nagios
nagios     55508       1  0 03:55 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     55509   55508  0 03:55 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.q
nagios     55510   55508  0 03:55 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.q
nagios     55511   55508  0 03:55 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.q
nagios     55512   55508  0 03:55 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.q
nagios     55513   55508  0 03:55 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root       57705    2695  0 04:29 pts/1    00:00:00 grep --color=auto nagios
[root@ip-172-31-35-58 ~]#
```

2) Next, switch to the root user and create a directory at the path '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts'.

sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

```
[root@ip-172-31-35-58 ~]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-35-58 ~]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-35-58 ~]#
```

3) We need to create a configuration file in this directory. To do this, copy the contents of the existing localhost configuration into the new file named 'linuxserver.cfg'. cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
[root@ip-172-31-35-58 ~]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-35-58 ~]#
```

So make the second directory again and run the cp command

```
[root@ip-172-31-84-149 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-84-149 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-84-149 ec2-user]#
```

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

```
[root@ip-172-31-84-149 ec2-user]# 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


    host_name               linuxserver
    alias                   linuxserver
    address                 3.86.39.170
}
```

Change hostgroup_name to linux-servers1

```
# 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 that are defined
                                                    ; in (or inherited by) the linux-server host template definition.
    host_name               linuxserver
    alias                   localhost
    address                 52.23.153.85
}
```

Change the occurrences of hostname further in the document from localhost to linuxserver
Now, we need to edit the nagios configuration file to add this directory. Run this command
nano /usr/local/nagios/etc/nagios.cfg

```
[root@ip-172-31-84-149 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

and add the following line cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
  GNU nano 5.8                    /usr/local/nagios/etc/nagios.cfg                    Modified
# 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

# 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


# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

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

```
[root@ip-172-31-84-149 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...
```

Once the files are verified, we need to restart the server: service nagios restart

```
[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

```
[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-84-149 ec2-user]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sat 2024-09-28 11:30:31 UTC; 3min 57s ago
       Docs: https://www.nagios.org/documentation
    Process: 73417 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 73418 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 73419 (nagios)
      Tasks: 6 (limit: 4658)
     Memory: 4.2M
        CPU: 113ms
     CGroup: /system.slice/nagios.service
             ├─73419 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─73420 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─73421 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─73422 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─73423 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─73425 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

Step 3: Execute the following on Nagios Client machine (Ubuntu)
1) First, check for any available updates, and then proceed to install gcc, the Nagios NRPE server, and Nagios plugins.
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

```
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 #6: sshd[1071,1184]
 ubuntu @ user manager service: systemd[1077]

No VM guests are running outdated hypervisor (qemu) binaries on this host
```

2) We need to include the public IP address of our Nagios host machine (Linux) in the NRPE configuration file. sudo nano /etc/nagios/nrpe.cfg

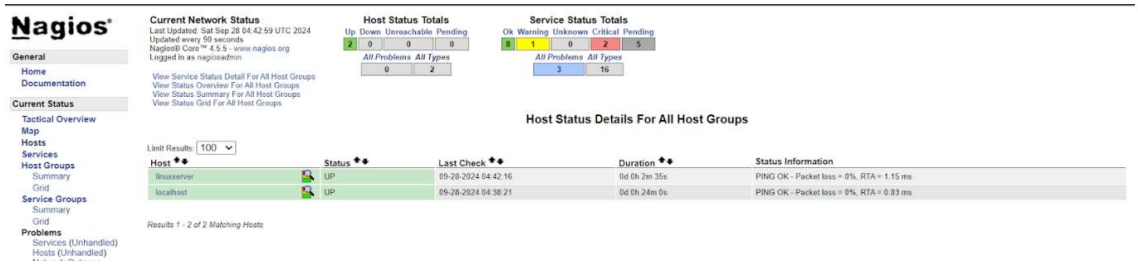
```
#############################################################################
#
#  Sample NRPE Config File
#
#  Notes:
#
#  This is a sample configuration file for the NRPE daemon.  It needs to be
#  located on the remote host that is running the NRPE daemon, not the host
#  from which the check_nrpe client is being executed.
#
#############################################################################


# LOG FACILITY
# The syslog facility that should be used for logging purposes.

log_facility=daemon



# LOG FILE
# If a log file is specified in this option, nrpe will write to
# that file instead of using syslog.

#log_file=/var/log/nrpe.log
```

Under allowed_hosts, add the nagios host ip address (public)

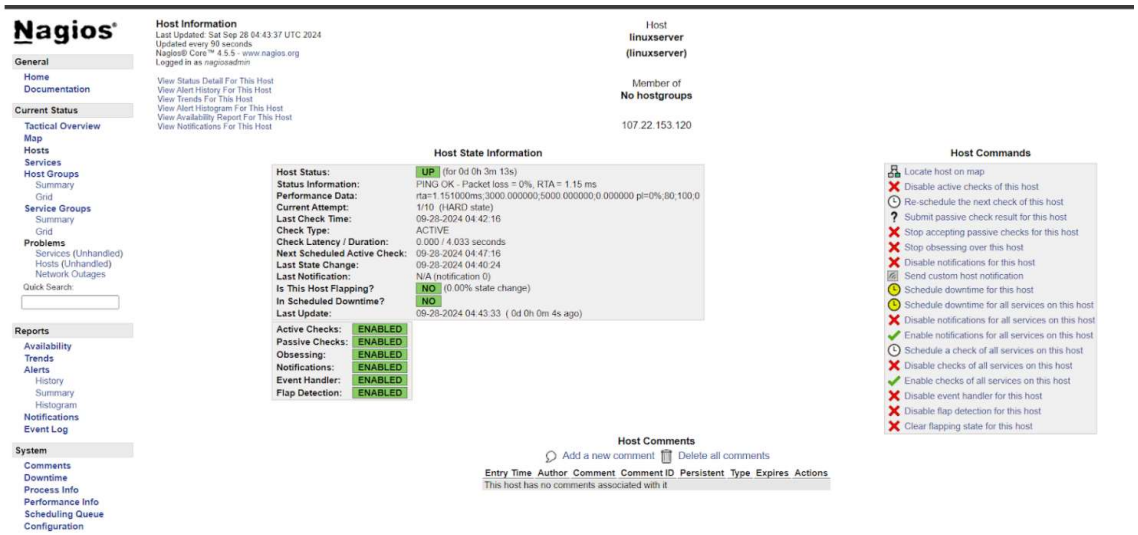
```
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does 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


# 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! ***
```

Name : Sandeshkumar.M.Yadav **Div : D15C** **Roll : 61**

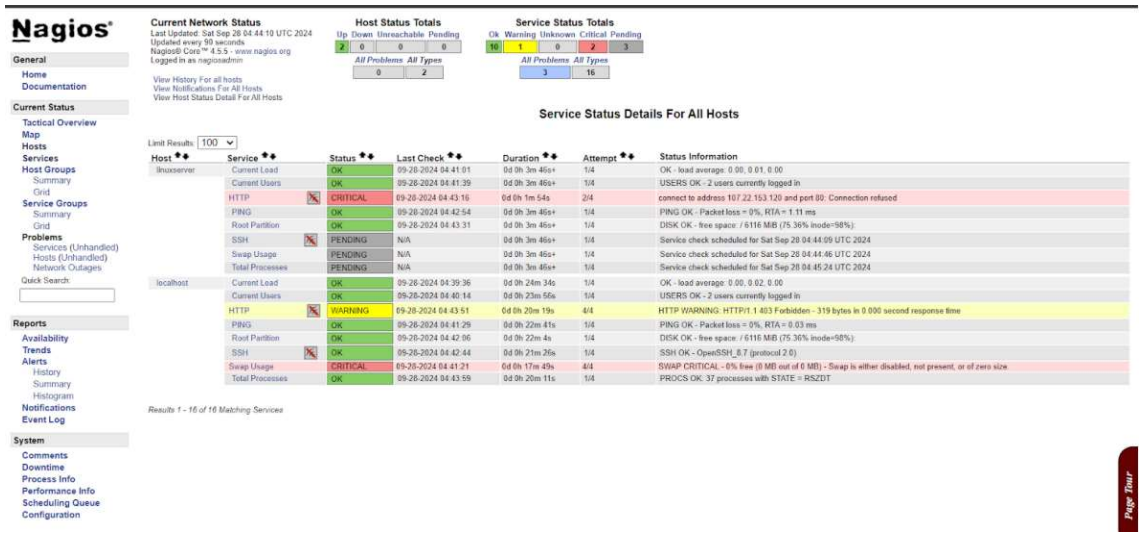
Step 4: Check the Nagios Dashboard.Go to Nagios dashboard, click on hosts.Here, we can see that the linuxserver is also added as a host.



Click on linuxserver. we can check all the information about linuxserver host.



Click on services. Here we can see all the services that are being monitored by linuxserver.



CONCLUSION :

In this experiment, we successfully conducted port and service monitoring along with server monitoring using Nagios. We set up an Amazon Linux instance as the Nagios host and linked it with an Ubuntu instance as the monitored host. The configuration involved updating the Nagios host to recognize the Ubuntu server, editing the necessary configuration files, and installing Nagios NRPE on the client machine. Once the setup was complete, the `linuxserver` was successfully added as a monitored host in the Nagios dashboard, allowing us to view and track its services in real time.