Name: Aditya Dubey **Div: D15C** **Roll No: 10**

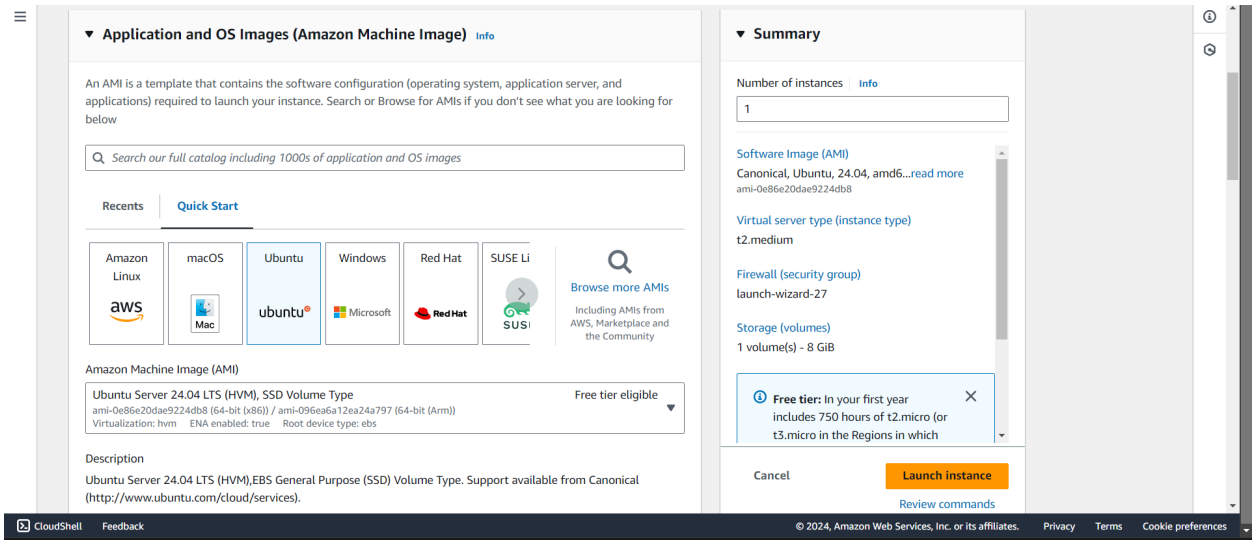
Experiment No:10

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

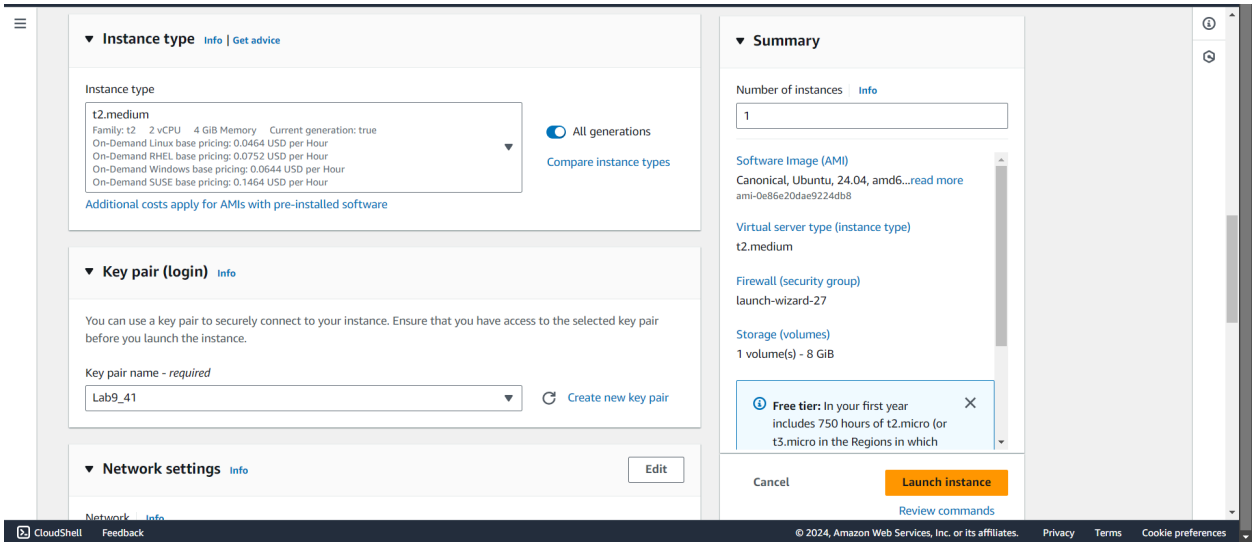
1) Log in to your AWS account. Look for EC2 in the services menu. Open the interface and select Create Instance.

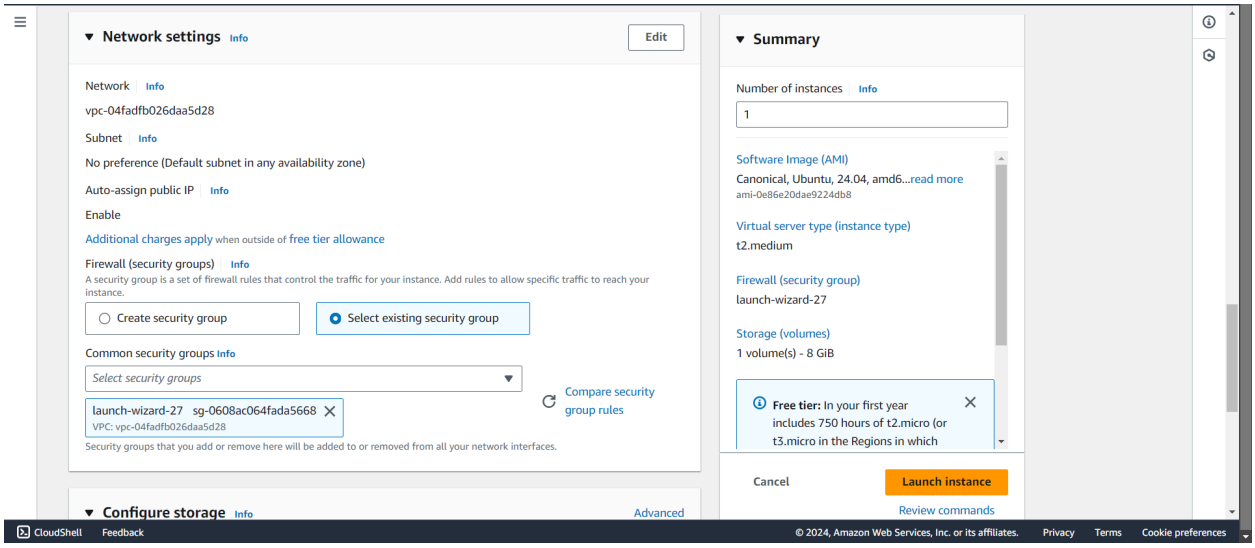


Select The OS Image as Ubuntu.

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.

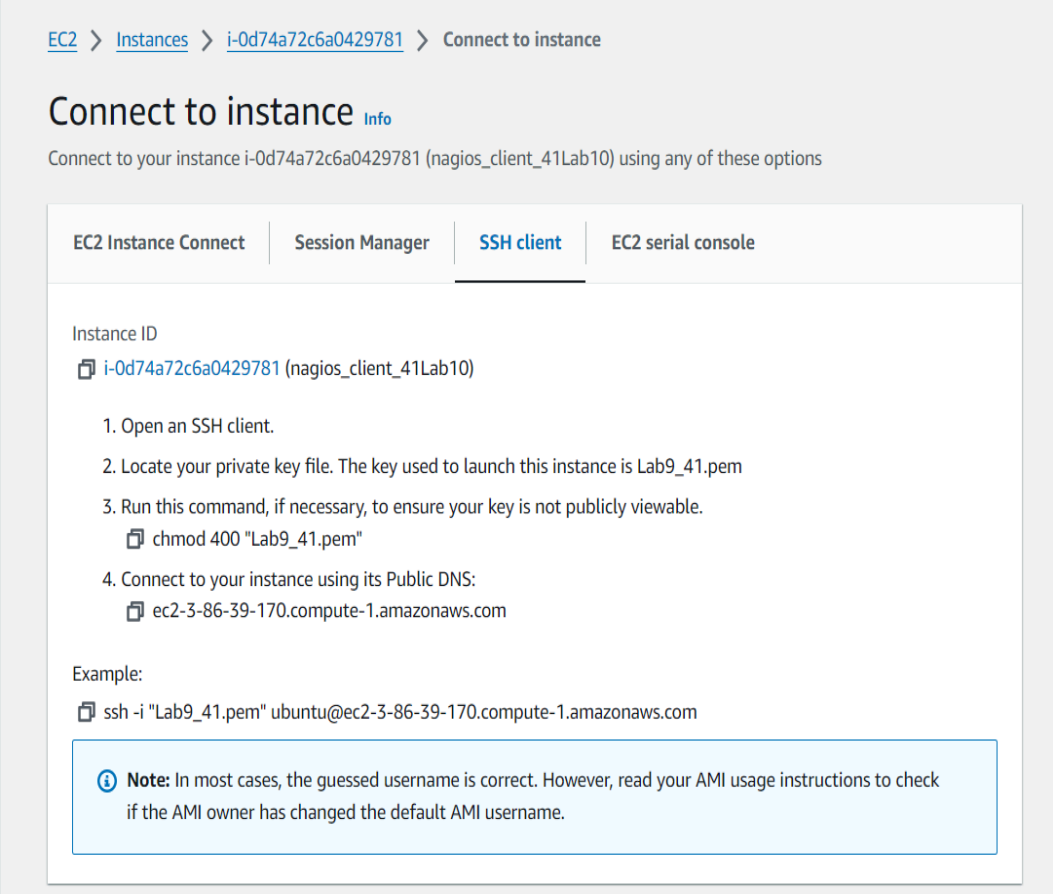
Name: Aditya Dubey **Div: D15C** **Roll No: 10**



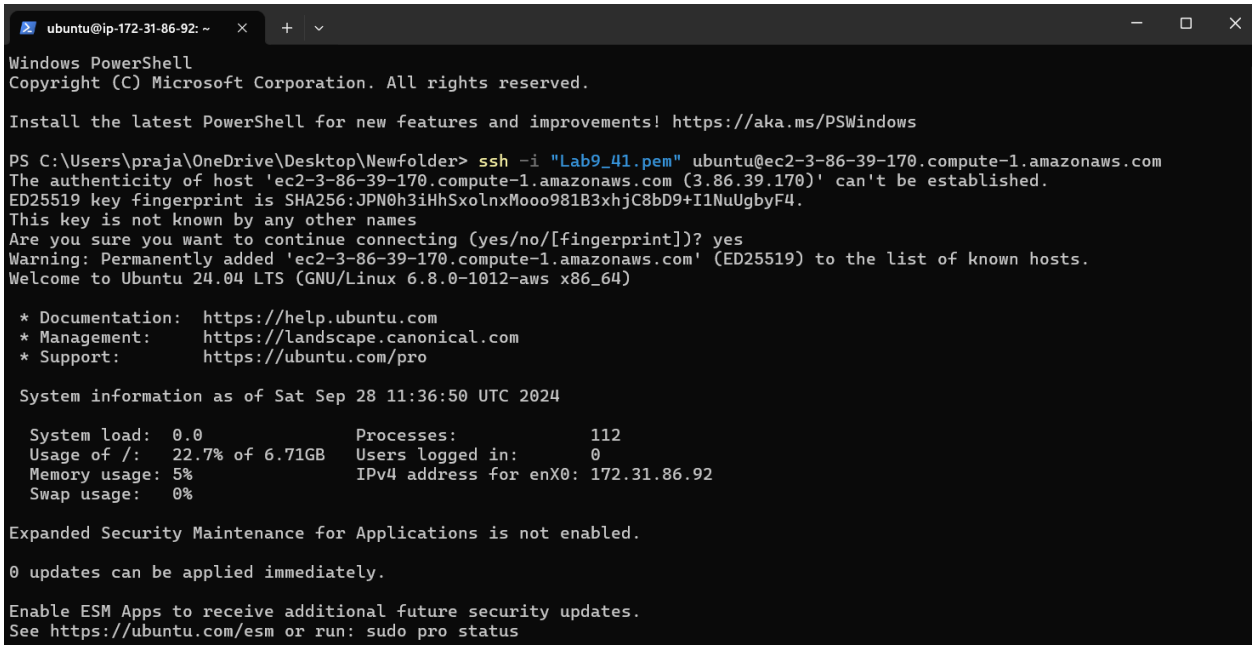


Instance is:

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.



Copy the example command



*Successfully connected the instance via **SSH***

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**

```
[ec2-user@ip-172-31-84-149 ~]$ ps -ef | grep nagios
nagios     67665       1  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     67666   67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67667   67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67668   67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67669   67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67670   67665  0 09:51 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user   70887    3276  0 10:48 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-84-149 ~]$
```

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

```
[ec2-user@ip-172-31-84-149 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-84-149 ec2-user]#
```

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-172-31-84-149 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg': No such file or directory
```

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 an optional hostgroup for Linux machines

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 belong to this group
}



##############################################################################
#
# SERVICE DEFINITIONS
#
##############################################################################
```

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


# OBJECT CACHE FILE
# This option determines where object definitions are cached when

^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
```

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...

Checking objects...
        Checked 16 services.
        Checked 2 hosts.
        Checked 2 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
```

```
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-172-31-84-149 ec2-user]#
```

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]#
```

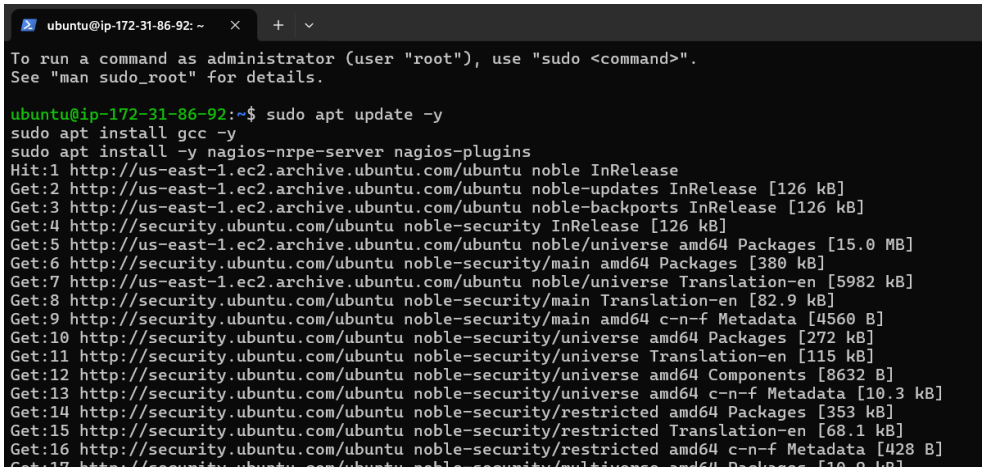
```
[root@ip-172-31-84-149 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-84-149 ec2-user]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     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

Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73423;pid=73423
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73421;pid=73421
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73420;pid=73420
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: wproc: Registry request: name=Core Worker 73422;pid=73422
Sep 28 11:30:31 ip-172-31-84-149.ec2.internal nagios[73419]: Successfully launched command file worker with pid 73425
Sep 28 11:32:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;1;connect to address 3.86.39.170 and port 80: Co>
Sep 28 11:33:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;2;connect to address 3.86.39.170 and port 80: Co>
Sep 28 11:34:23 ip-172-31-84-149.ec2.internal nagios[73419]: SERVICE ALERT: linuxserver;HTTP;CRITICAL;SOFT;3;connect to address 3.86.39.170 and port 80: Co>
lines 1-28/28 (END)
```

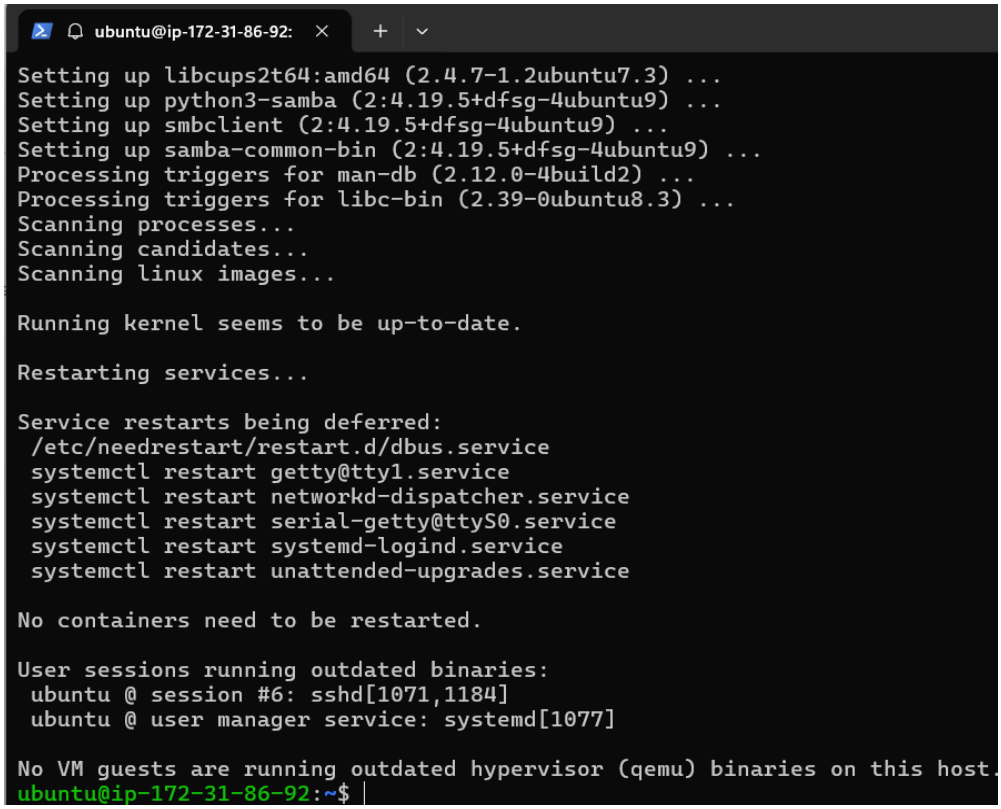
Name: Aditya Dubey **Div: D15C** **Roll No: 10**

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.

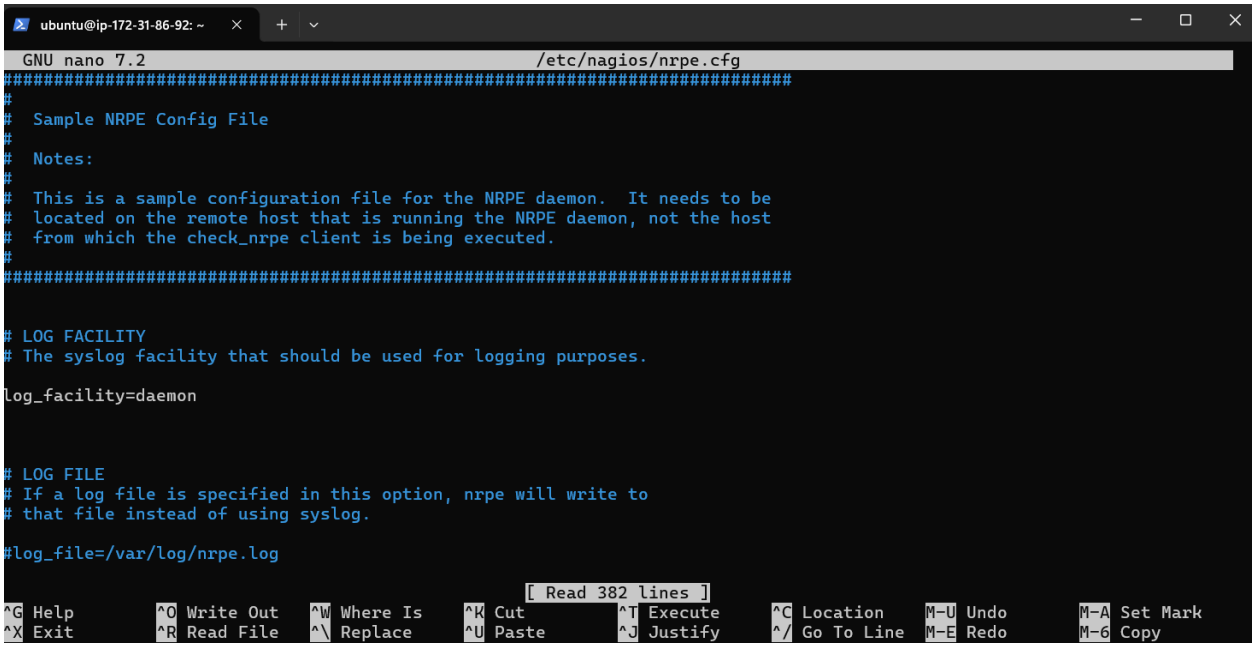
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins



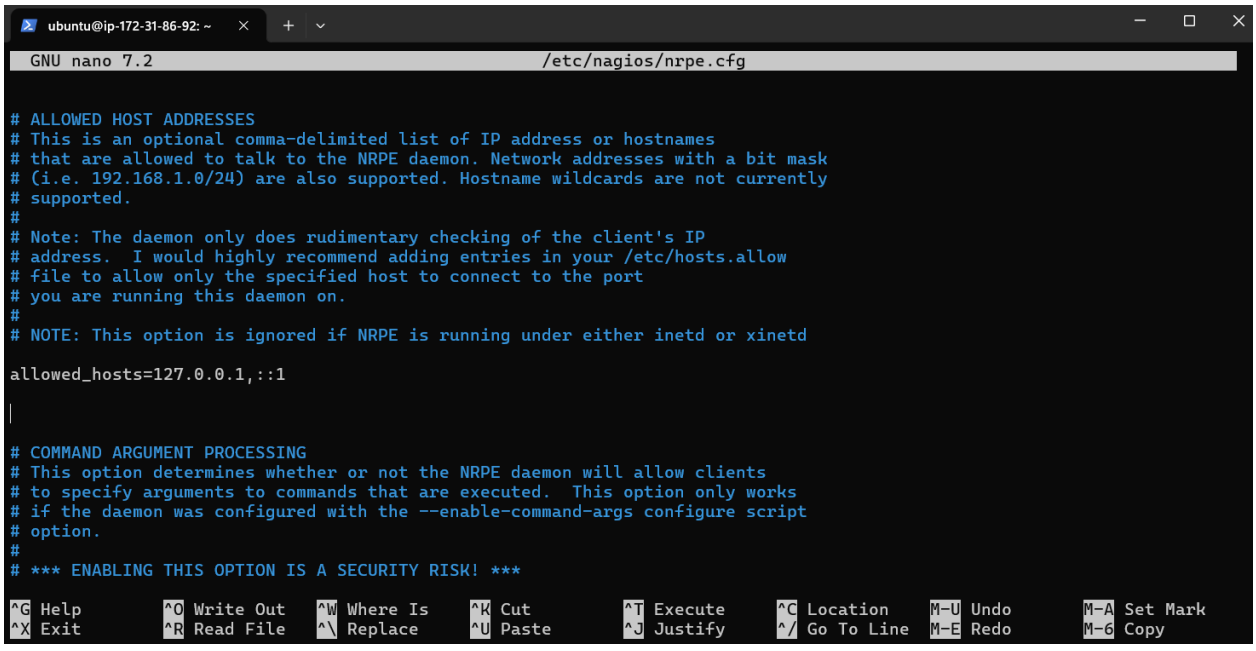


Name: Aditya Dubey **Div: D15C** **Roll No: 10**

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**

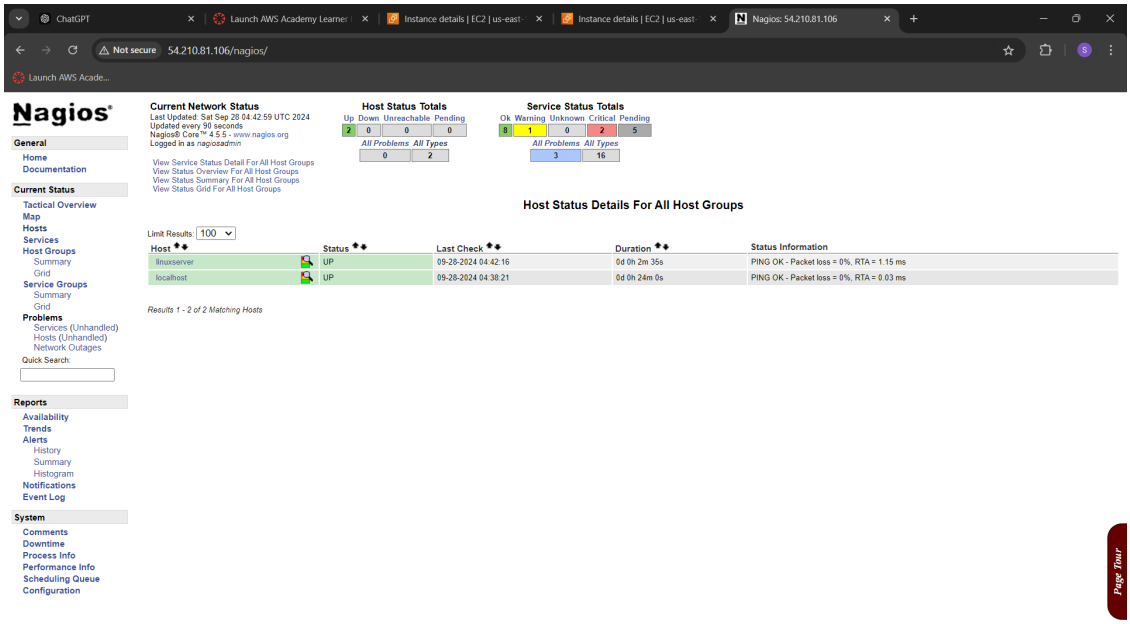


Under allowed_hosts, add the nagios host ip address (public)

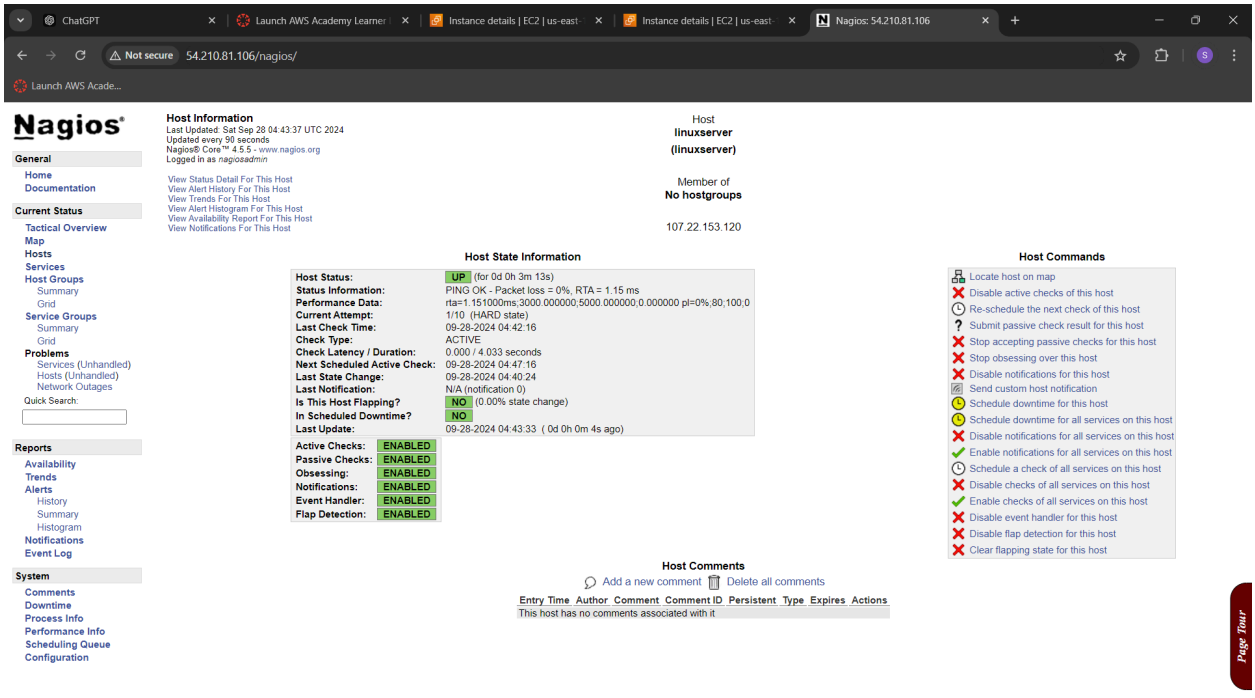


Name: Aditya Dubey **Div: D15C** **Roll No: 10**

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.

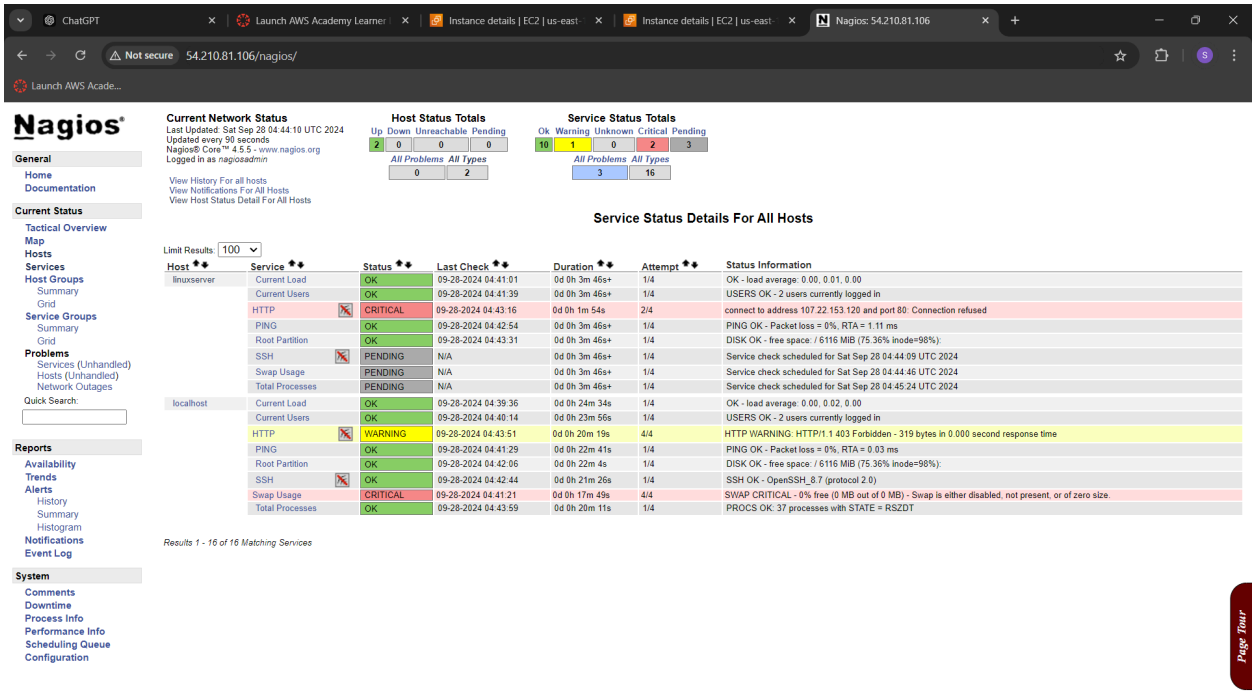


Click on linuxserver. we can check all the information about linuxserver host.



Name: Aditya Dubey **Div: D15C** **Roll No: 10**

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



CONCLUSION:

In this experiment, we learned to conduct port service monitoring and server monitoring using Nagios. To do this, we require a Linux instance to host the Nagios dashboard and a separate Ubuntu instance linked as a second host. We need to configure the Linux instance and include the IP address of the Ubuntu instance. Subsequently, we must replicate the initial setup from the Linux instance on the Ubuntu instance by adding the IP address of the Linux instance to the list of allowed hosts. After restarting the NRPE server, we should see the ‘linuxserver’ host listed.

Name: Aditya Dubey **Div: D15C** **Roll No: 10**