Name-Ishan Kiran Joshi Div -D15C Roll No-21 A.Y.-2024-25

Experiment 10

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

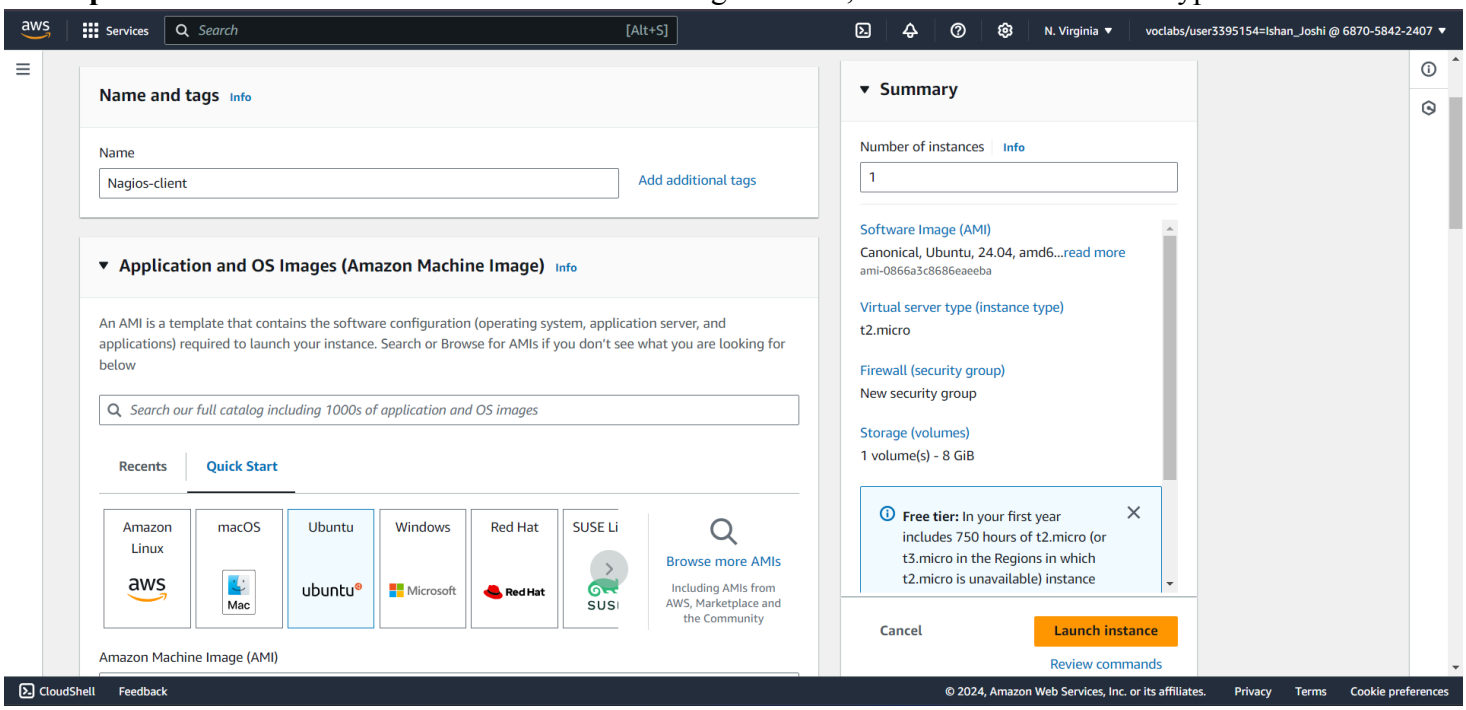
Prerequisites:
AWS Academy or Personal account.
Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

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

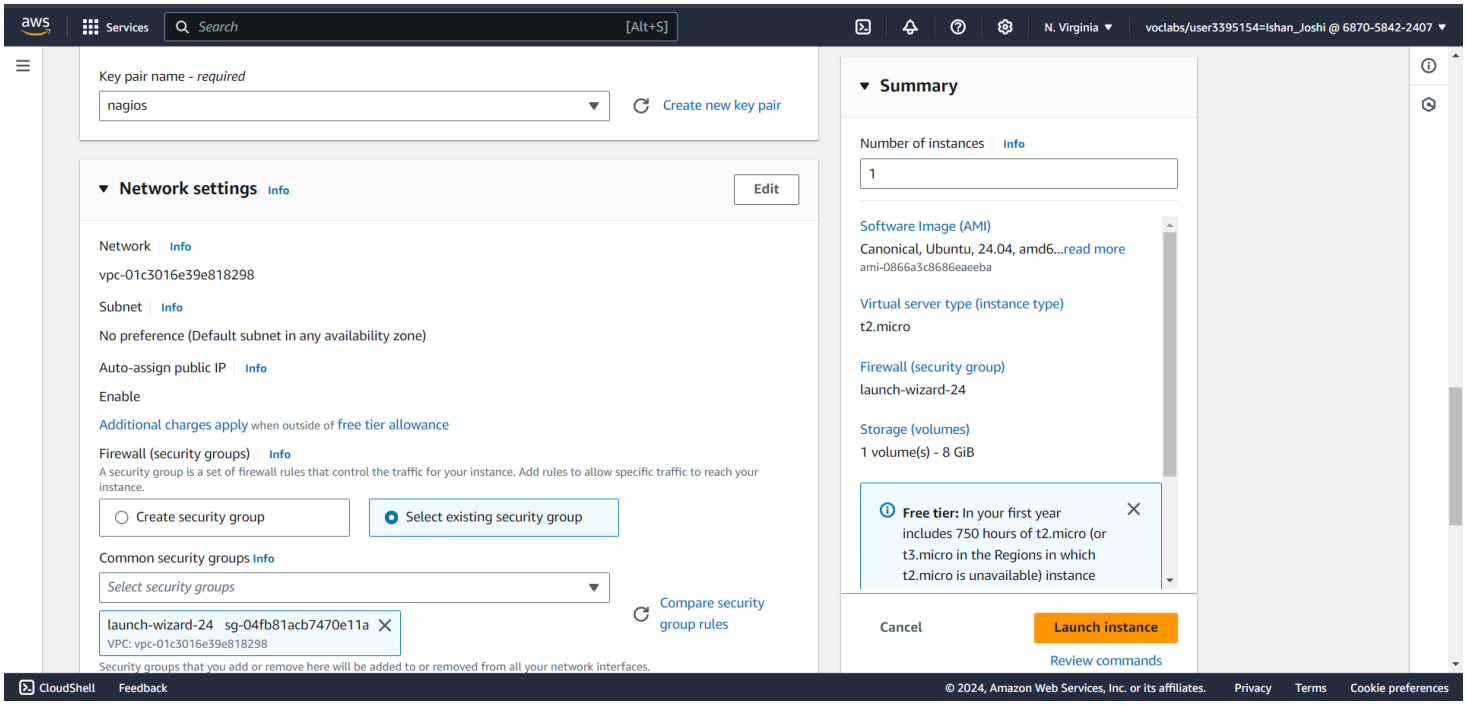
You can now proceed if you get the above message/output.

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

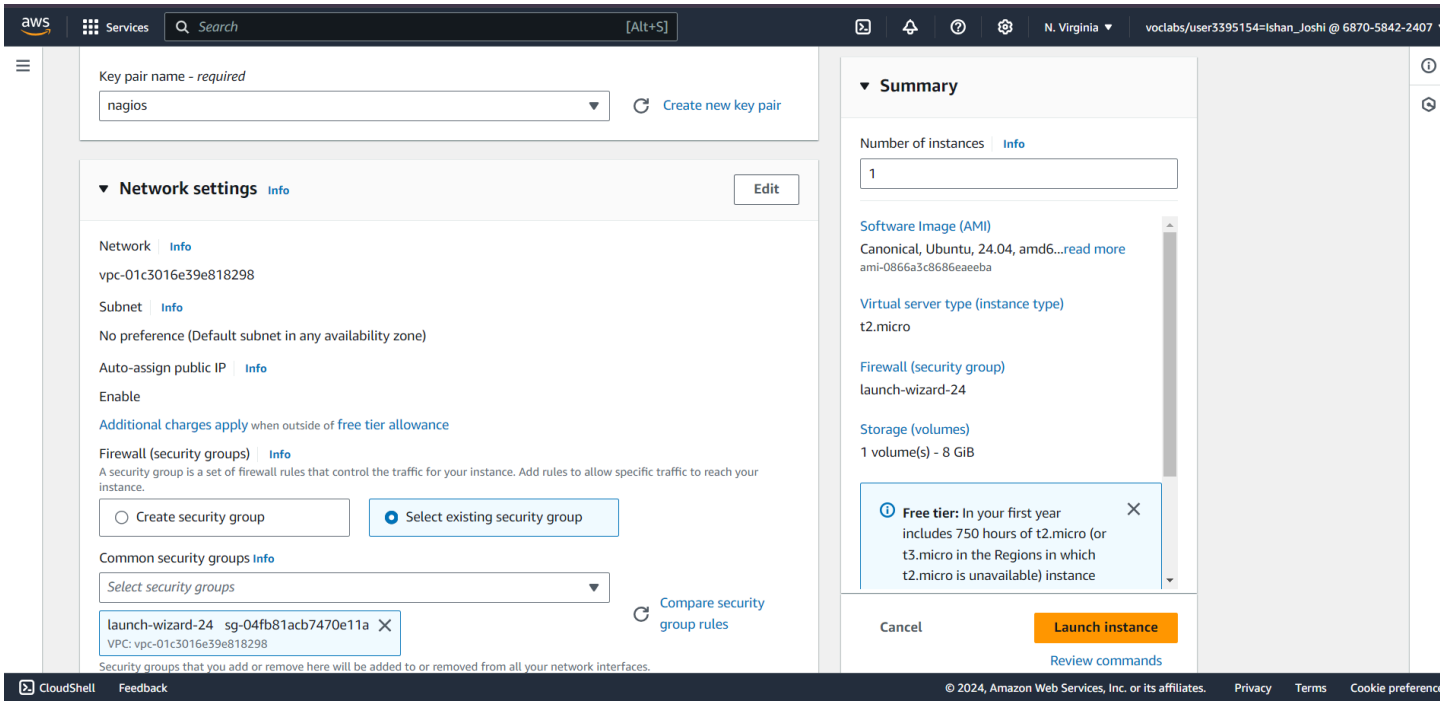


For Key pair : Click on create key and make key of type RSA with nagios .pem . Key will be downloaded to your local machine.

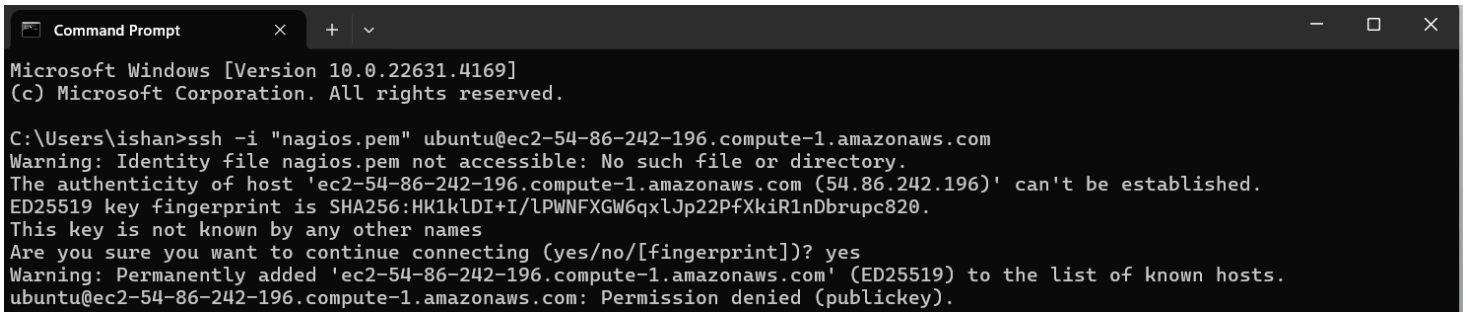
Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



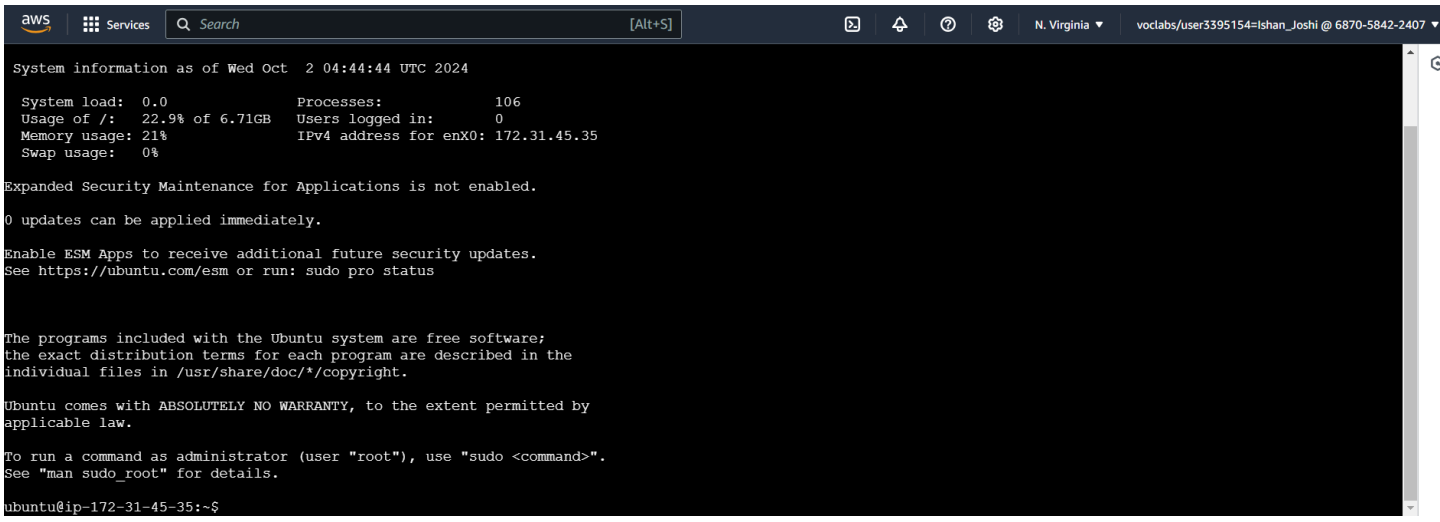
Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



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.



Successfully connected to the instance



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

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

```
[ec2-user@ip-172-31-91-133 ~]$ ps -ef | grep nagios
nagios      2006       1  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      2007    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2008    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2009    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2010    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2011    2006  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    3145    2703  0 04:45 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-133 ~]$
```

sudo su

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

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

```
[ec2-user@ip-172-31-91-133 ~]$ sudo su
[root@ip-172-31-91-133 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-91-133 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-91-133 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-172-31-91-133 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-91-133 ec2-user]#
```

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

Change hostname to **linuxserver**.
Change address to the public IP of your Linux client. Set
hostgroup_name to **linux-servers1**.

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

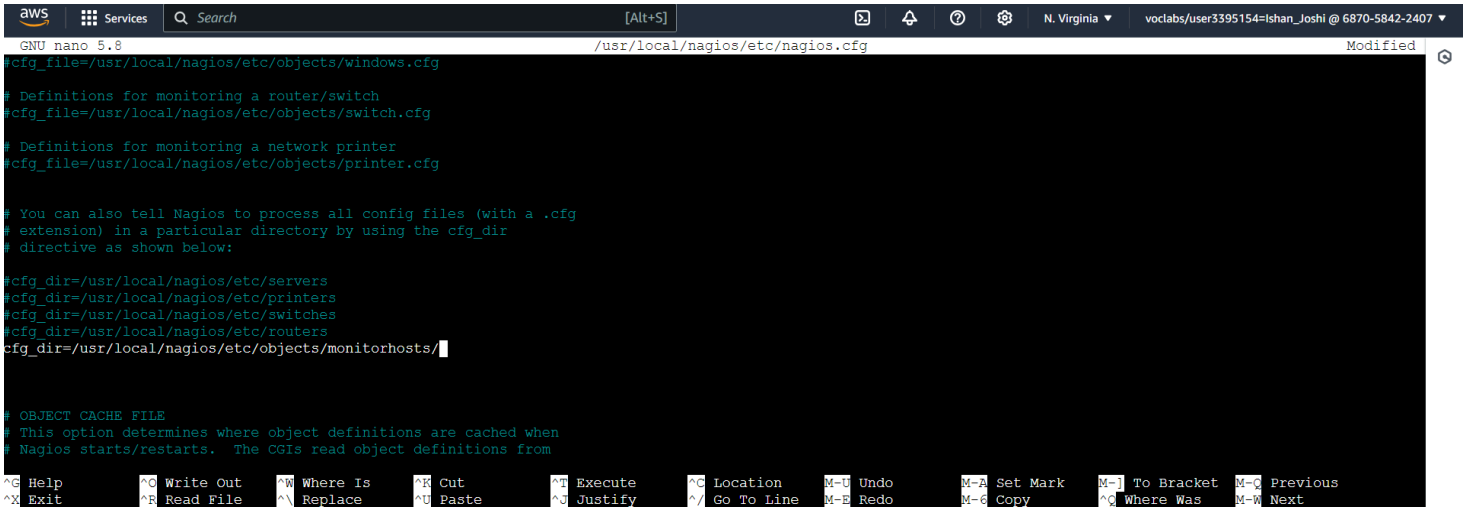
```
# Define a host for the local machine

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                 54.86.242.196
}
```

```
  GNU nano 5.8                          /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg                Modified
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
#
##############################################################################
```

Step 8: Now update the Nagios config file .Add the following line in the file.

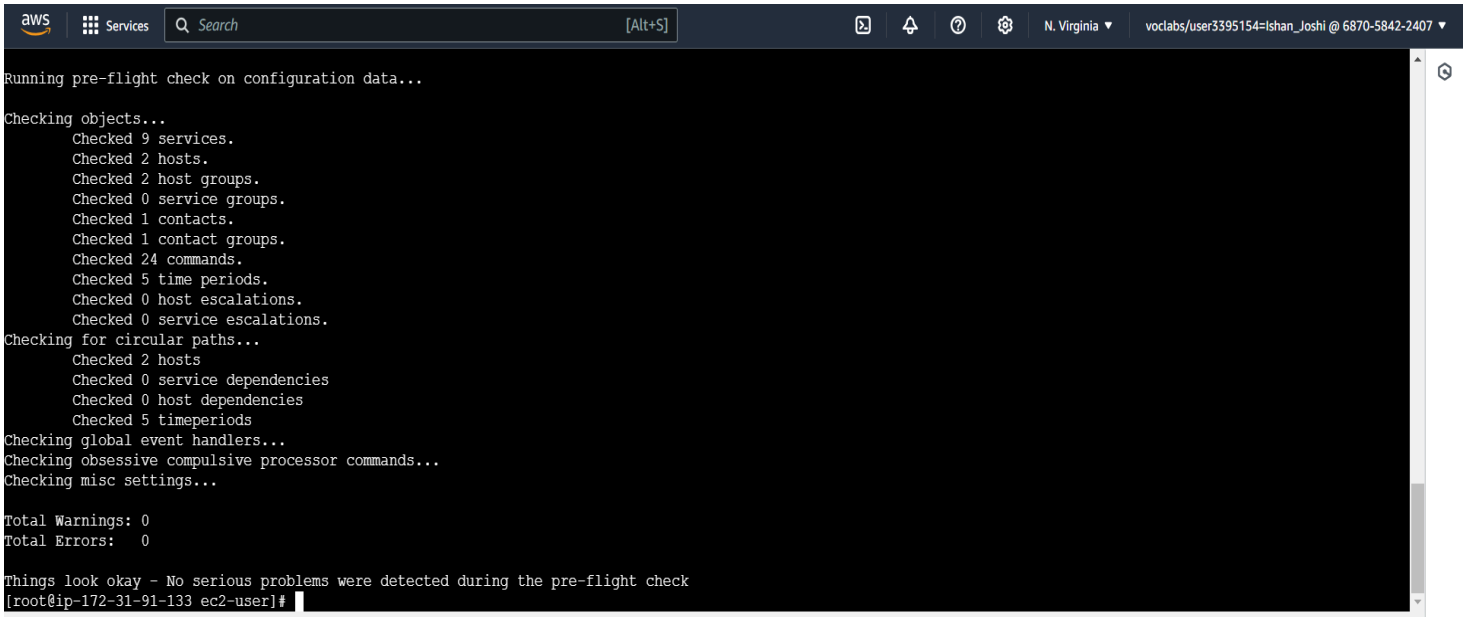
Line to add : cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

Run the command : **nano /usr/local/nagios/etc/nagios.cfg**



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

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



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

service nagios restart

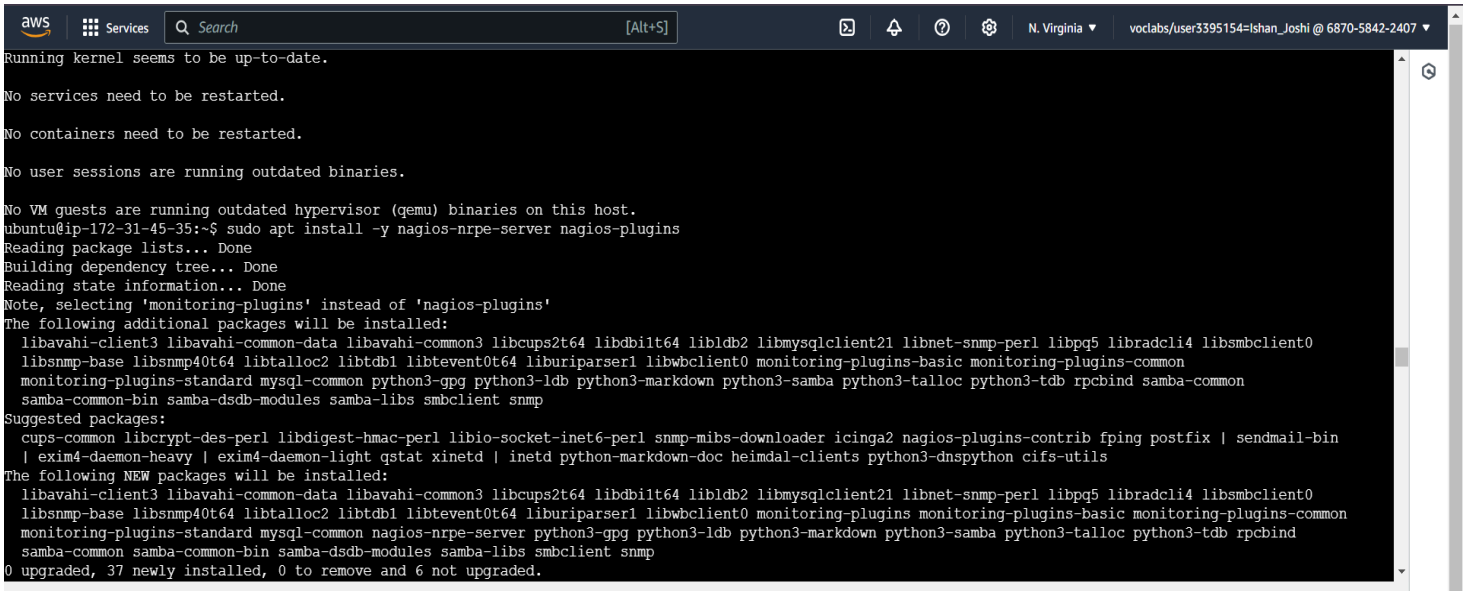
```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-133 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-91-133 ec2-user]#
```

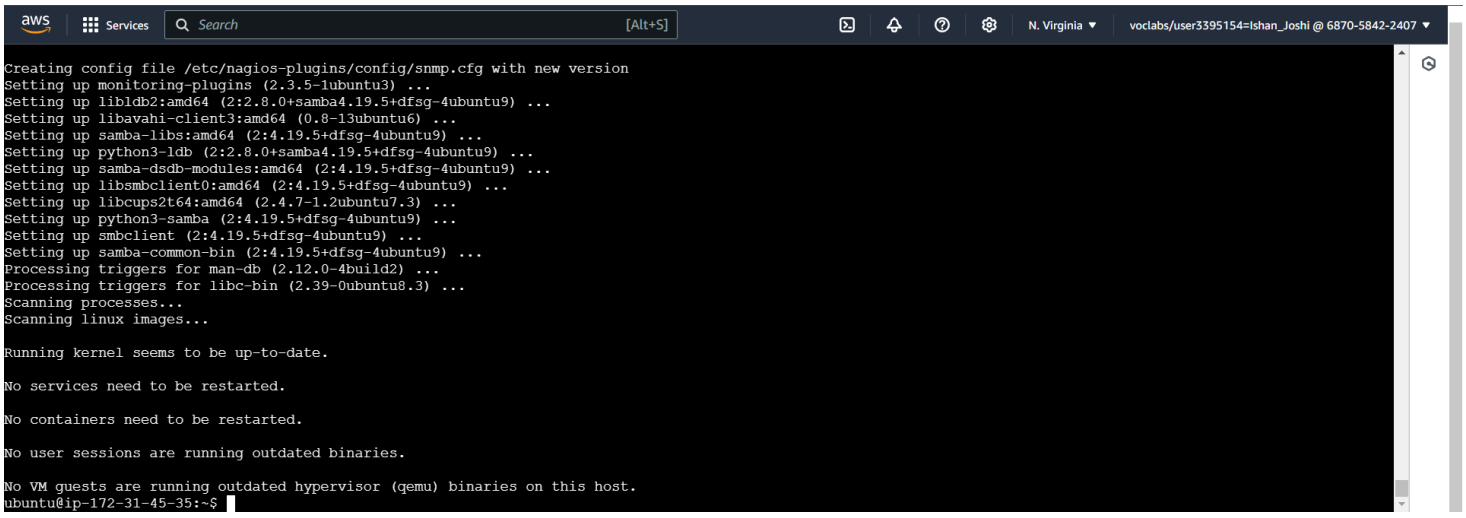
Step 11: Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

sudo apt update -y

sudo apt install gcc -y

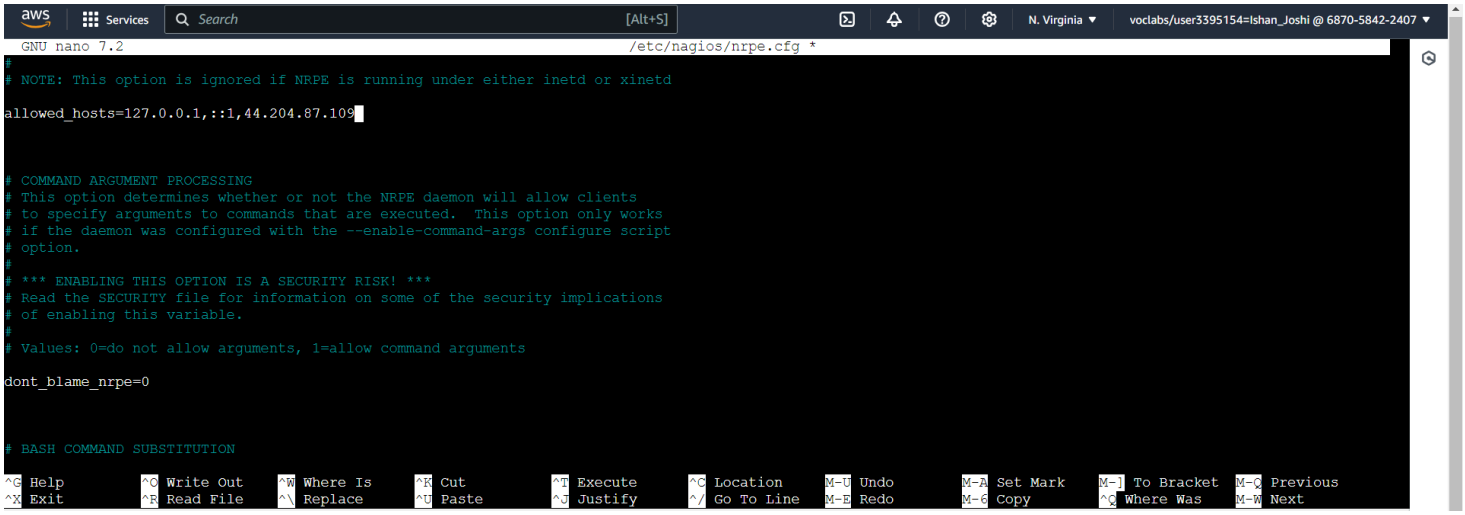
sudo apt install -y nagios-nrpe-server nagios-plugins





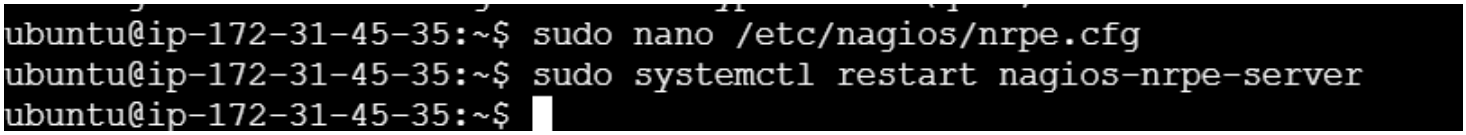
Step 12: Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host IP address.

sudo nano /etc/nagios/nrpe.cfg



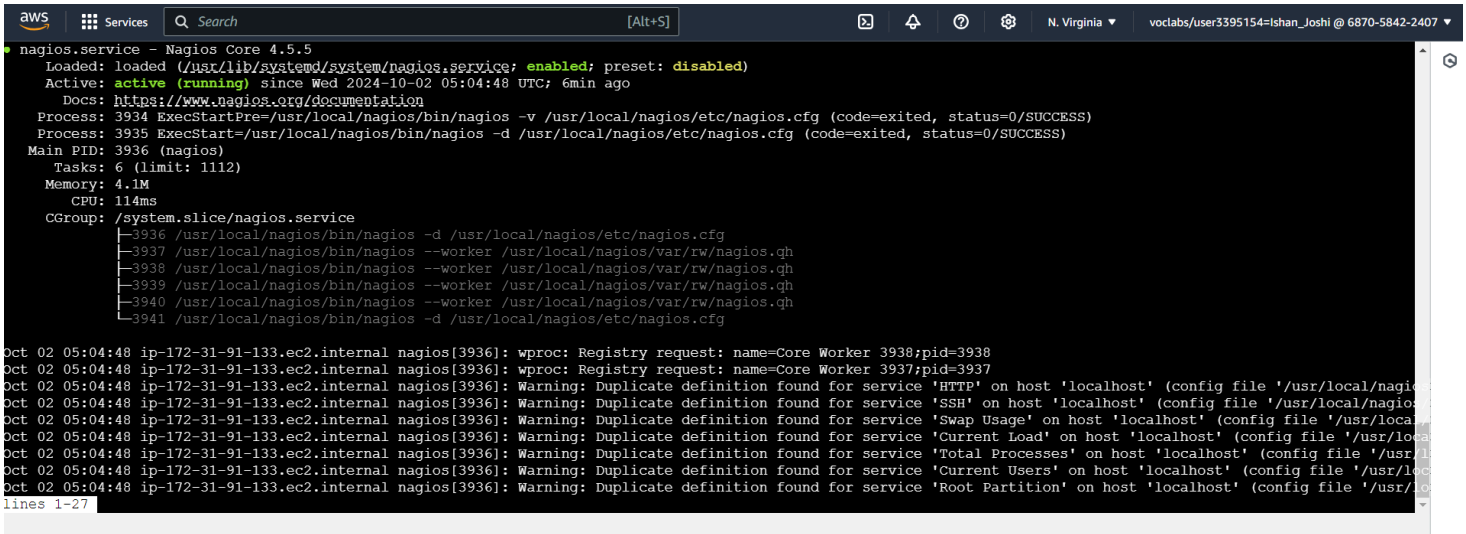
Step 13: Now restart the NRPE server by this command.

sudo systemctl restart nagios-nrpe-server



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.

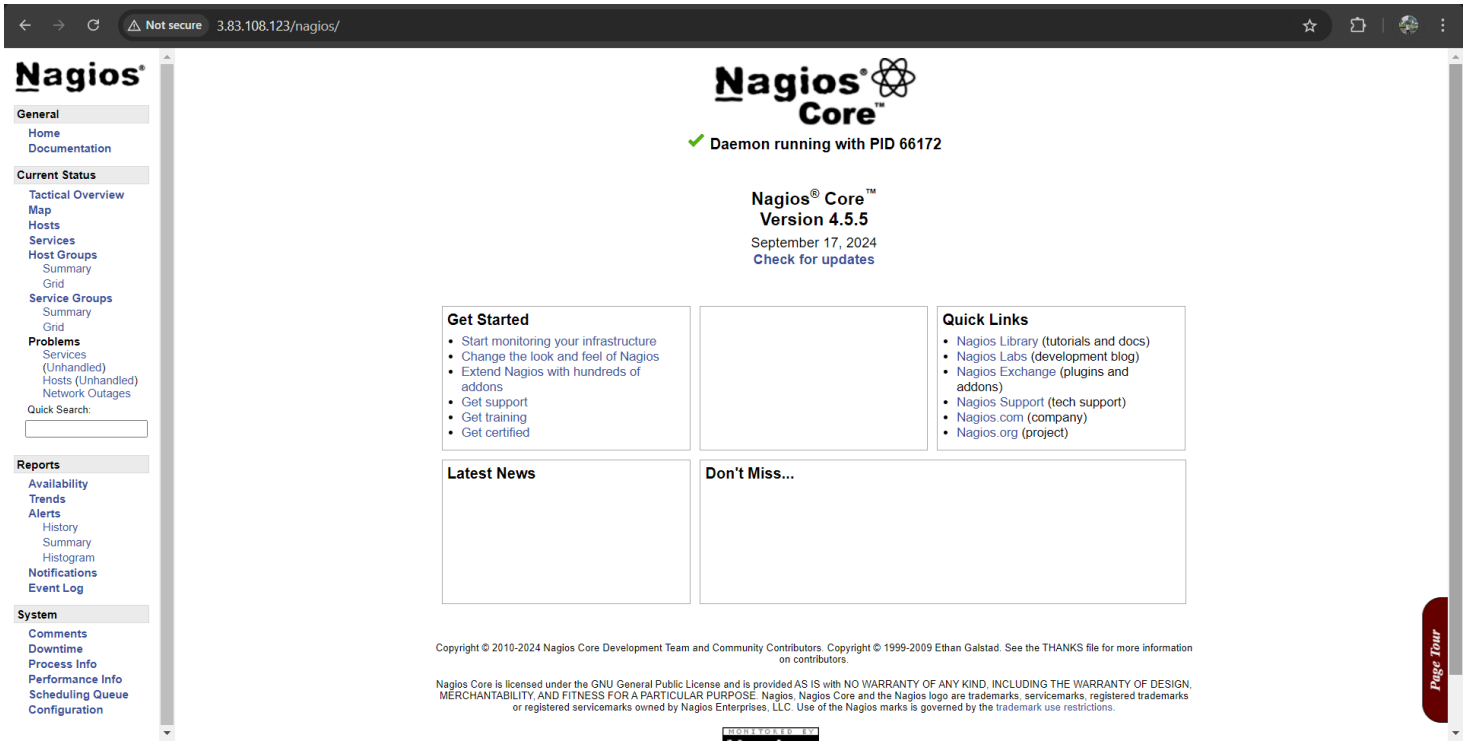
sudo systemctl status nagios



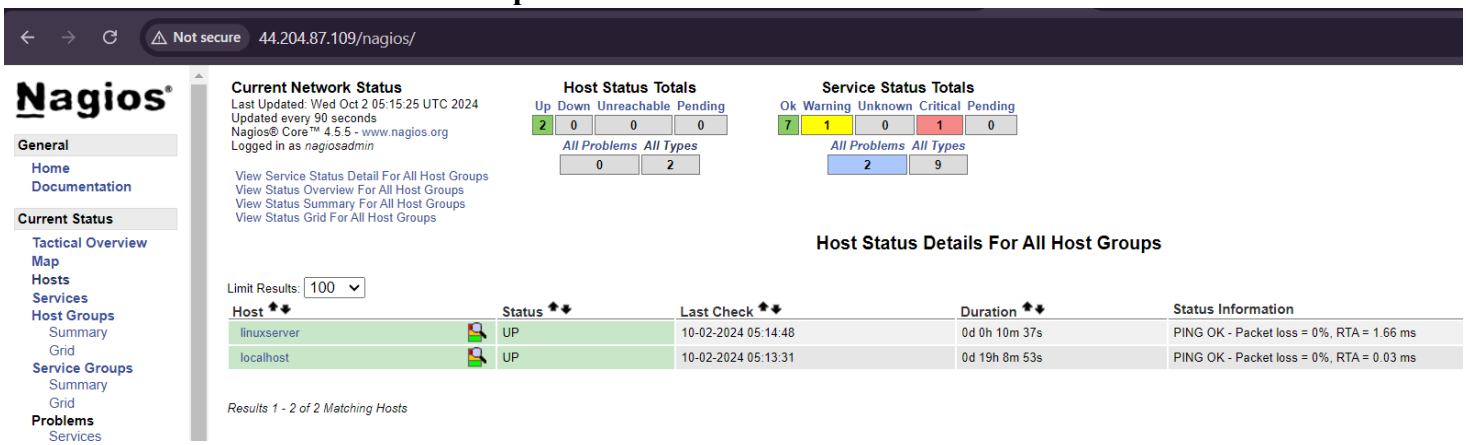
sudo systemctl status httpd
sudo systemctl start httpd sudo
systemctl enable httpd

```
[root@ip-172-31-91-133 ec2-user]# sudo systemctl status httpd
o httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-91-133 ec2-user]# sudo systemctl start httpd sudo systemctl enable httpd
Failed to start sudo.service: Unit sudo.service not found.
Failed to start systemctl.service: Unit systemctl.service not found.
Failed to start enable.service: Unit enable.service not found.
[root@ip-172-31-91-133 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-91-133 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-91-133 ec2-user]#
```

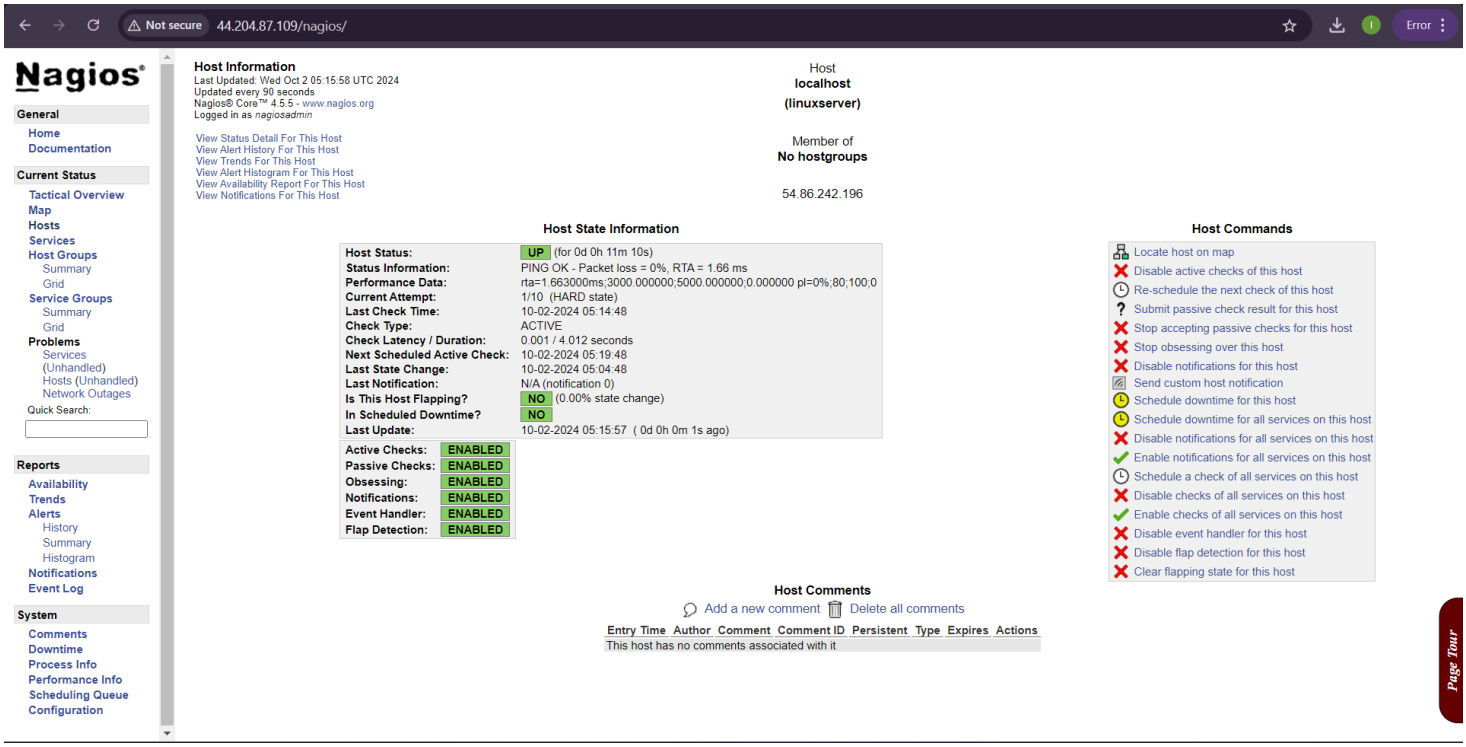
Step 15: Now to check Nagios dashboard go to http://<Nagios-host ip>/nagios .



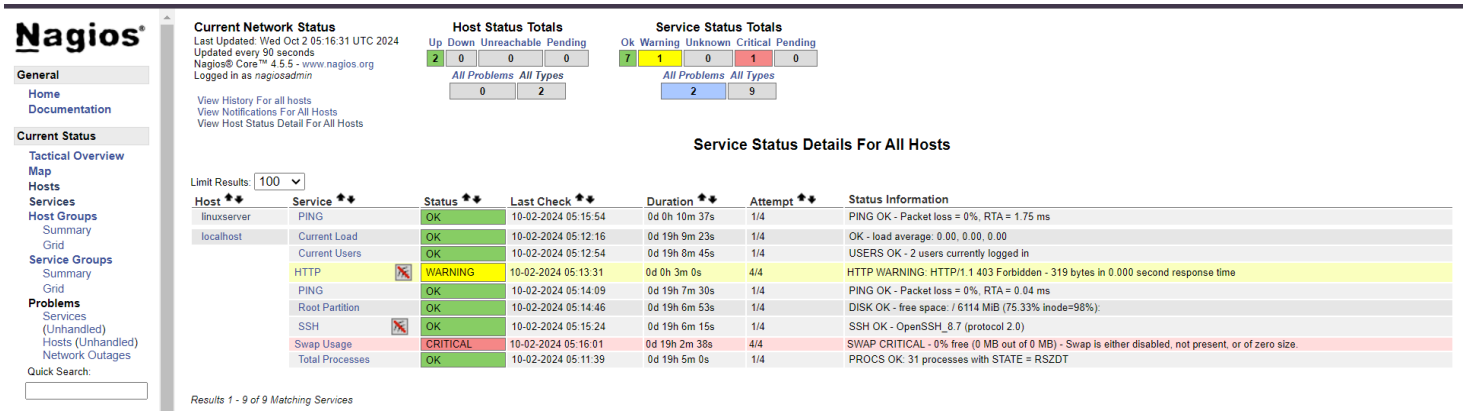
Now Click on Hosts from left side panel



We can see our linuxserver now click on it we can see the host information.



Current Network Status



Conclusion: In conclusion, this experiment concentrated on monitoring network ports, services, and a Linux server with Nagios. By following a systematic approach, we effectively configured Nagios to oversee vital network services on the Linux server. With both the Nagios host and client set up, we gained the ability to track system performance, ensure the availability of services, and monitor crucial metrics such as CPU and memory usage. This comprehensive monitoring not only enhances our operational awareness but also empowers us to respond swiftly to any emerging issues. Ultimately, this setup lays the groundwork for maintaining a robust and reliable IT environment.