Name: Manav Jawrani

Roll No.: 19

Subject: Advanced DevOps

Experiment No.: 10

Experiment 10

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

Theory:

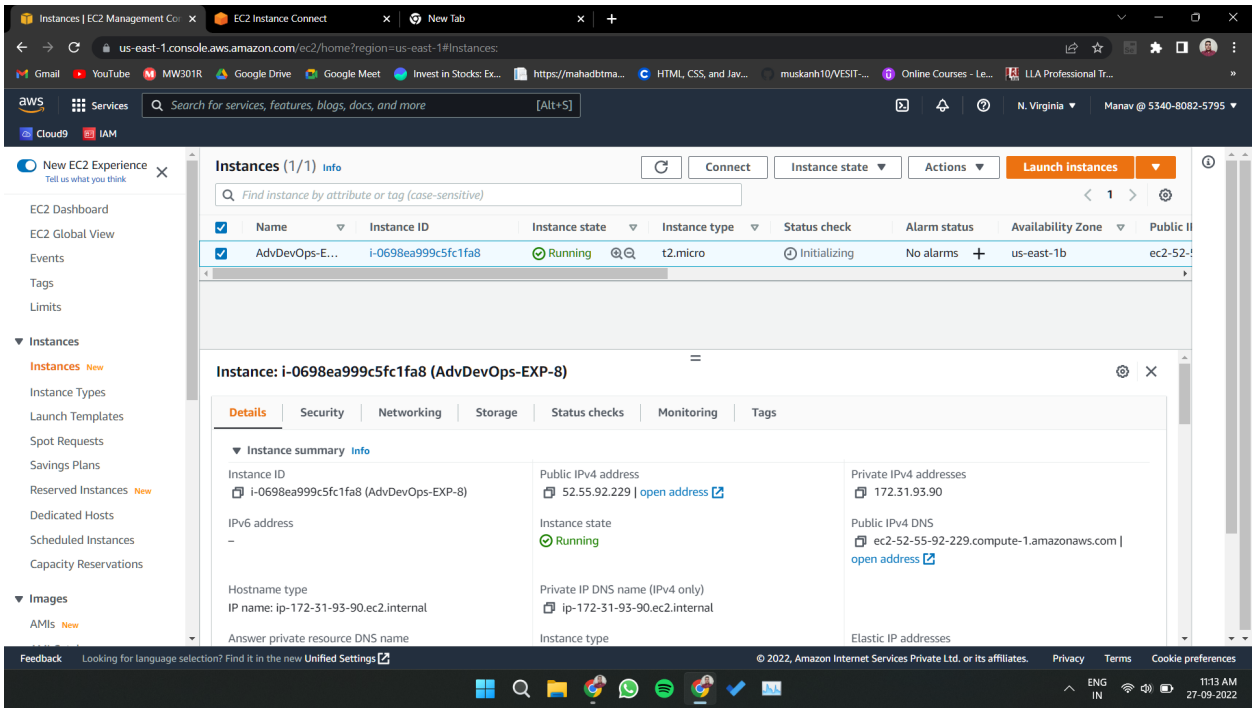
Since we had already gone through the introduction, features of ~~nagi~~ nagios, here we will discuss the importance of ~~nagio~~ nagios and how it helps in service monitoring. The importance is as follows:

1. It can describe the event handler that executes at the time of host events to take proper action.
2. Along with this it supports in redundancy in monitoring hosts
3. It can also be monitored in hardware tools like a probe for alarm, that can send collected information through the network by configured written plugins.
4. The remote monitoring can be established through nagios remote plugin executor via SSH, encrypted channels.
5. Nagios has other services like Nagios remote plugin executor (NRPE), Nagios remote data processor (NRDP) and many more.

Implementation:

Prerequisites:

AWS Free Tier, Nagios Server running on Amazon Linux Machine which we had previously created in Experiment 9.



Step 1: Start the apache server using the command.

sudo systemctl start httpd

Step 2: Start the nagios server using:

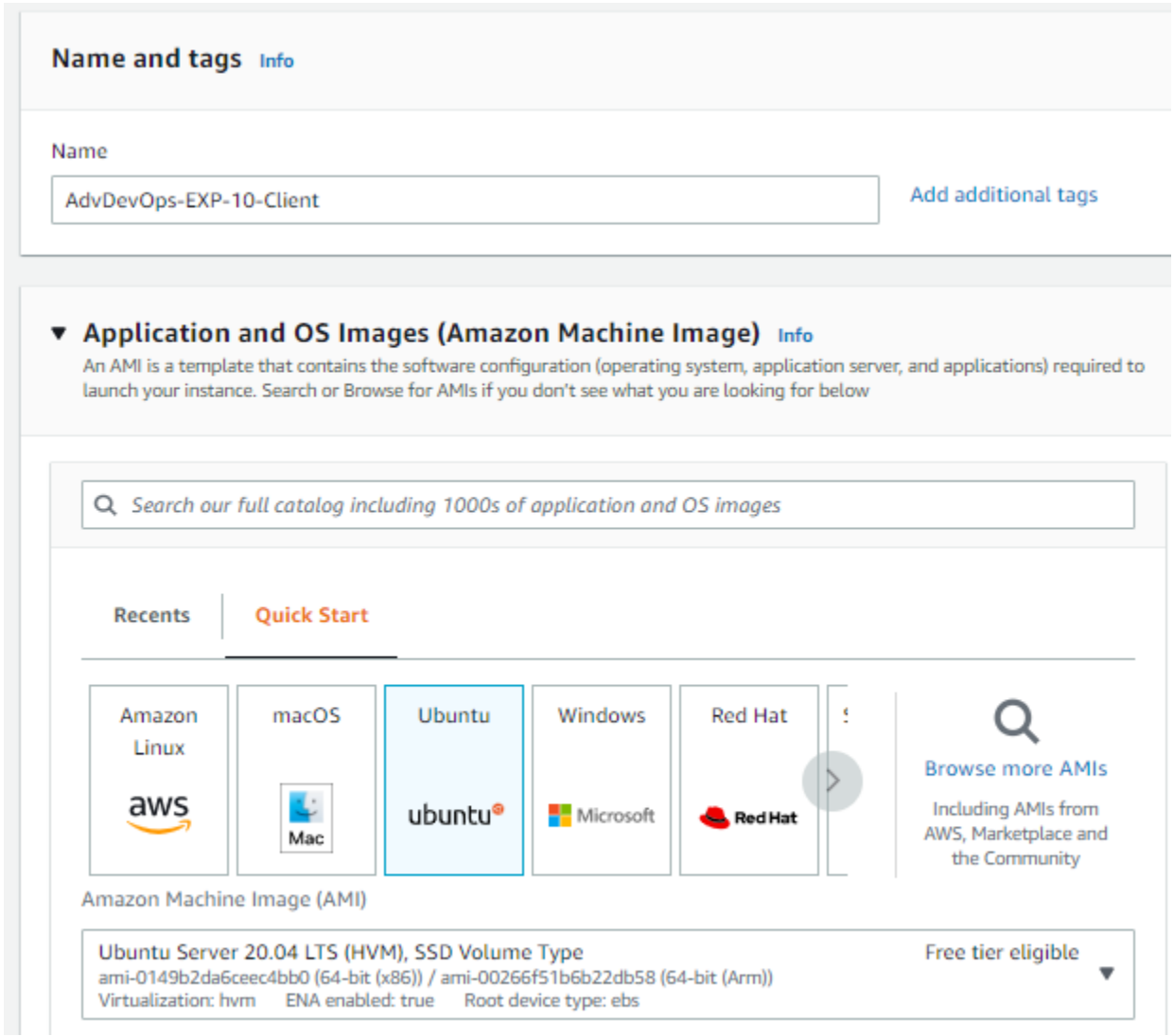
sudo systemctl start nagios

Also confirm that Nagios is running on the server side, run this

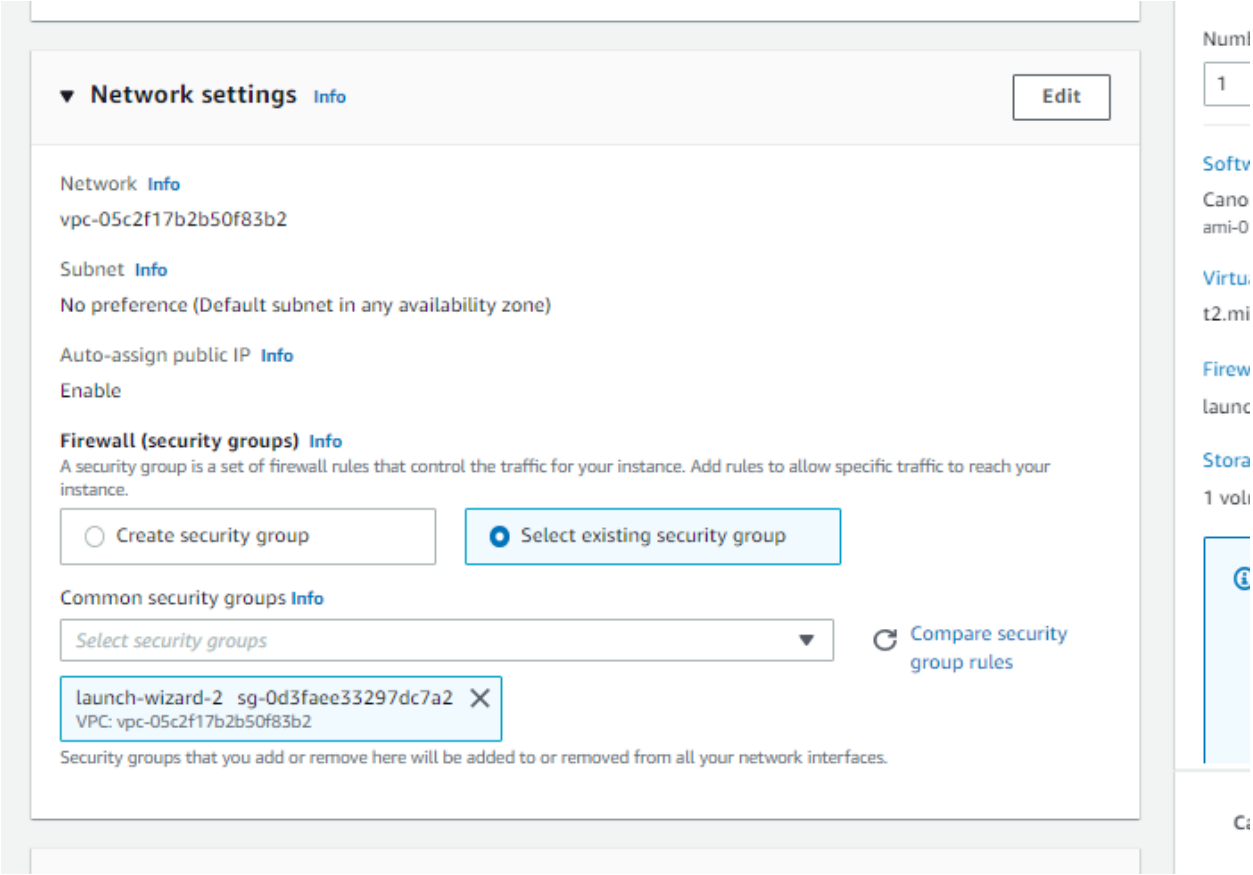
sudo systemctl status nagios

```
[ec2-user@ip-172-31-93-90 ~]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios; bad; vendor preset: disabled)
   Active: active (running) since Tue 2022-09-27 05:40:21 UTC; 4min 27s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2974 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─3028 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─3030 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─3031 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─3032 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─3033 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─3034 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 27 05:40:21 ip-172-31-93-90.ec2.internal nagios[3028]: wproc: Registry request: name=Core Worker 3031;pid=3031
Sep 27 05:40:21 ip-172-31-93-90.ec2.internal nagios[3028]: wproc: Registry request: name=Core Worker 3030;pid=3030
Sep 27 05:40:21 ip-172-31-93-90.ec2.internal nagios[3028]: Successfully launched command file worker with pid 3034
Sep 27 05:42:13 ip-172-31-93-90.ec2.internal nagios[3028]: SERVICE ALERT: localhost;HTTP;CRITICAL;HARD;4;connect to address 127.0.0.1 and port 80: Connection refused
Sep 27 05:44:43 ip-172-31-93-90.ec2.internal nagios[3028]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL...ero size.
Sep 27 05:44:43 ip-172-31-93-90.ec2.internal nagios[3028]: wproc: NOTIFY job 2 from worker Core Worker 3033 is a non-check helper but exited with return code 127
Sep 27 05:44:43 ip-172-31-93-90.ec2.internal nagios[3028]: wproc:   host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 27 05:44:43 ip-172-31-93-90.ec2.internal nagios[3028]: wproc:   early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 27 05:44:43 ip-172-31-93-90.ec2.internal nagios[3028]: wproc:   stderr line 01: /bin/sh: /bin/mail: No such file or directory
Sep 27 05:44:43 ip-172-31-93-90.ec2.internal nagios[3028]: wproc:   stderr line 02: /usr/bin/printf: write error: Broken pipe
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-93-90 ~]$
```

Step 3: To monitor a Linux machine, create a Ubuntu 20.04 EC2 Instance on AWS and name it as **"AdvDevOps-EXP-10-Client"**



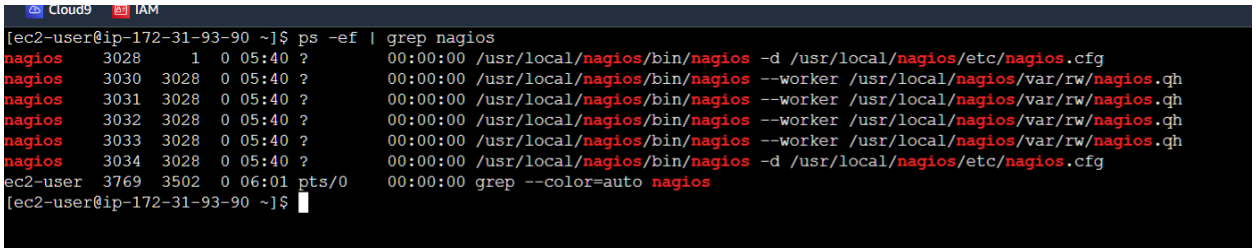
Provide it with the same security group as the Nagios Host.



For now, leave this machine as it is, and go back to your nagios HOST machine.

Step 4: On the server, run this command

ps -ef | grep nagios



Step 5: Login into root user and create 2 folders

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

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

```
[root@ip-172-31-93-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-93-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

Step 6: Copy the sample localhost.cfg file to linux host folder
cp /usr/local/nagios/etc/objects/localhost.cfg
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

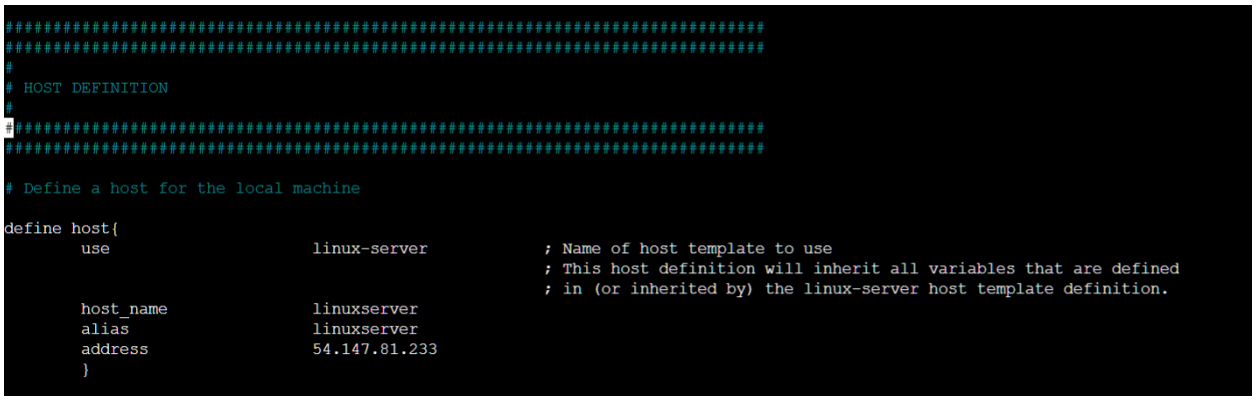
```
Cloud9  IAM
[root@ip-172-31-93-90 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg  /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-93-90 ec2-user]#
```

Step 7: Open linuxserver.cfg using nano and make the following changes

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

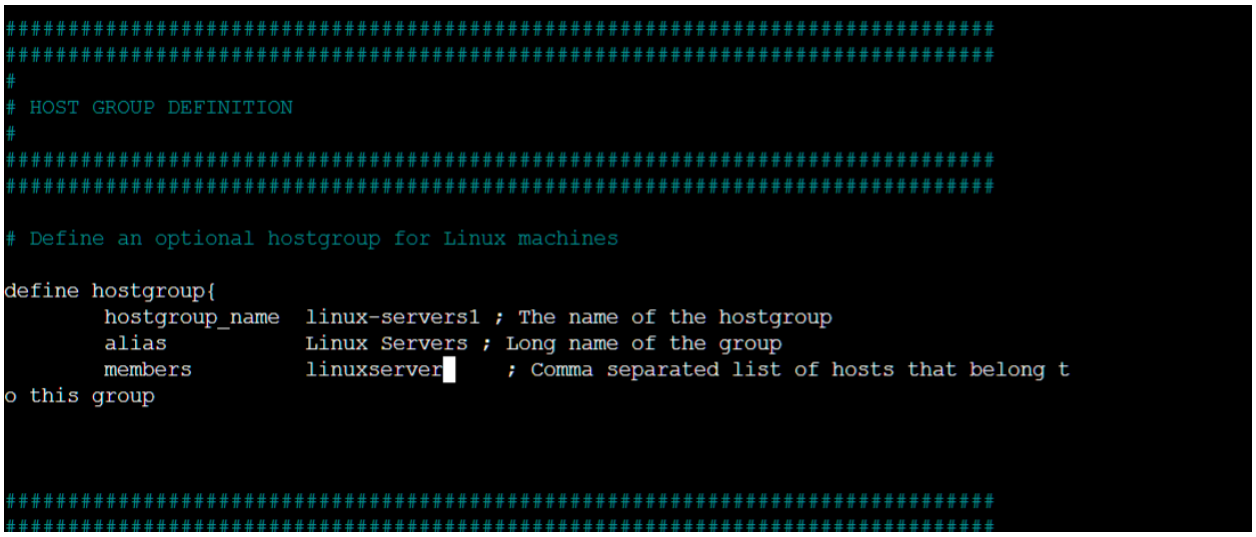
```
  GNU nano 2.9.8                    /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

##############################################################################
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
#
# NOTE: This config file is intended to serve as an *extremely* simple
#       example of how you can create configuration entries to monitor
#       the local (Linux) machine.
#
##############################################################################



##############################################################################
##############################################################################
#
# HOST DEFINITION
#
##############################################################################
##############################################################################

                                [ Read 156 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket  M-▲ Previous
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line M-E Redo  M-6 Copy Text  M-W WhereIs Next M-▼ Next
```

Change the hostname to linuxserver (EVERYWHERE ON THE FILE) Change address to the public IP address of your **LINUX CLIENT**.



```
###############################################################################
###############################################################################
#
# HOST DEFINITION
#
###############################################################################
###############################################################################

# 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                   linuxserver
       address                 54.147.81.233
       }
```

Change **hostgroup_name** under hostgroup to linux-servers1



```
###############################################################################
###############################################################################
#
# HOST GROUP DEFINITION
#
###############################################################################
###############################################################################

# 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         linuxserver    ; Comma separated list of hosts that belong t
o this group



###############################################################################
###############################################################################
```

Everywhere else on the file, change the hostname to linuxserver instead of localhost (there will be around 5/6 changes).

Step 8: Open the Nagios Config file and add the following line

nano /usr/local/nagios/etc/nagios.cfg
##Add this line
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
# 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
```

Step 9: Verify the configuration files using the command:

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

```
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
```

You are good to go if there are no errors.

Step 10: Restart the nagios service

service nagios restart

And then check the status

```
[root@ip-172-31-93-90 ec2-user]# service nagios restart
Restarting nagios (via systemctl):                         [  OK  ]
[root@ip-172-31-93-90 ec2-user]# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios; bad; vendor preset: disabled)
   Active: active (running) since Tue 2022-09-27 06:31:19 UTC; 12s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 4103 ExecStop=/etc/rc.d/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 4110 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─4132 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─4134 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─4135 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─4136 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─4137 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─4138 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─4139 /usr/local/nagios/libexec/check_ping -H 25.55.92.229 -w 3000.0,80% -c 5000.0,100% -p 5
           └─4140 /usr/bin/ping -n -U -w 30 -c 5 25.55.92.229

Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: nerd: Channel hostchecks registered successfully
Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: nerd: Channel servicechecks registered successfully
Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: nerd: Channel opathchecks registered successfully
Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: nerd: Fully initialized and ready to rock!
Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: wproc: Successfully registered manager as @wproc with query handler
Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: wproc: Registry request: name=Core Worker 4137;pid=4137
Sep 27 06:31:19 ip-172-31-93-90.ec2.internal nagios[4132]: wproc: Registry request: name=Core Worker 4136;pid=4136
```

Now it is time to switch to the client machine.

Step 11: Connect to your client machine using the EC2 Instance Connect feature.

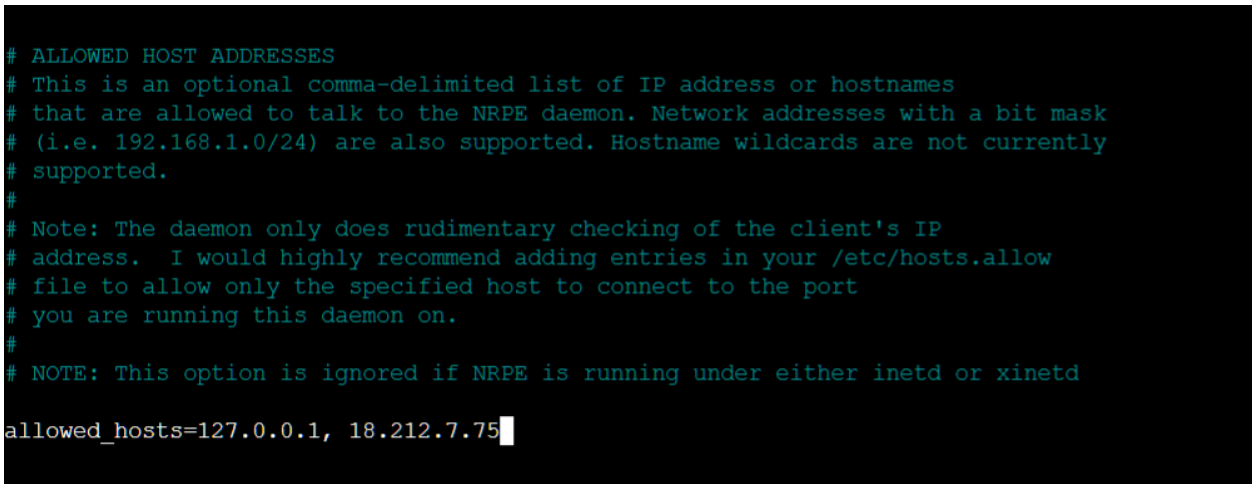
```
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Sep 27 06:44:45 UTC 2022

  System load:  0.88              Processes:             105
  Usage of /:   19.6% of 7.57GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for eth0: 172.31.95.126
  Swap usage:   0%

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


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@ip-172-31-95-126:~$
```

Step 12: Make a package index update and install gcc, nagios-nrpe-server and the plugins.
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

Step 13: Open nrpe.cfg file to make changes.

sudo nano /etc/nagios/nrpe.cfg

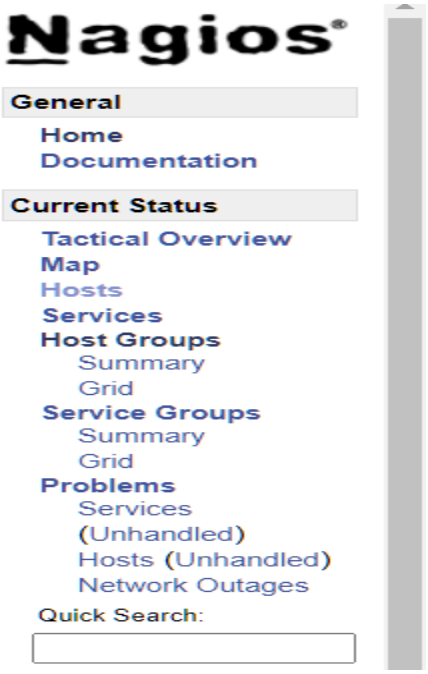
Under allowed_hosts, add your nagios host IP address like so



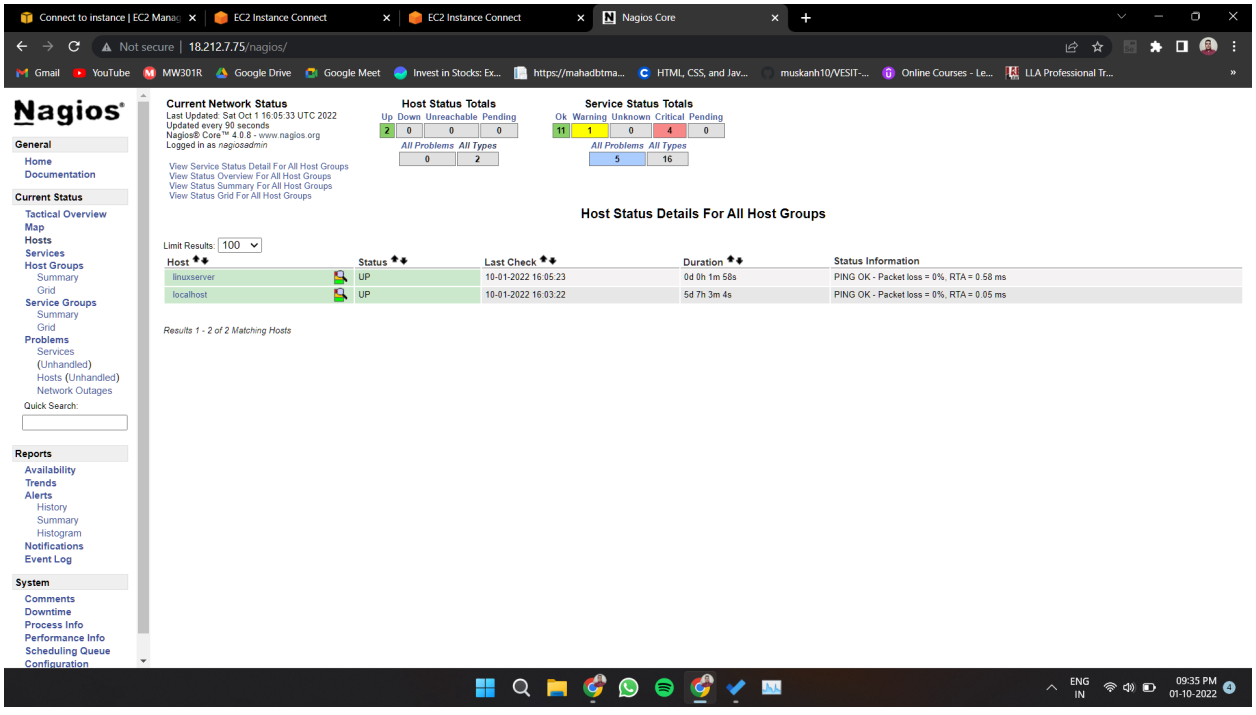
Step 14: Restart the NRPE server

sudo systemctl restart nagios-nrpe-server

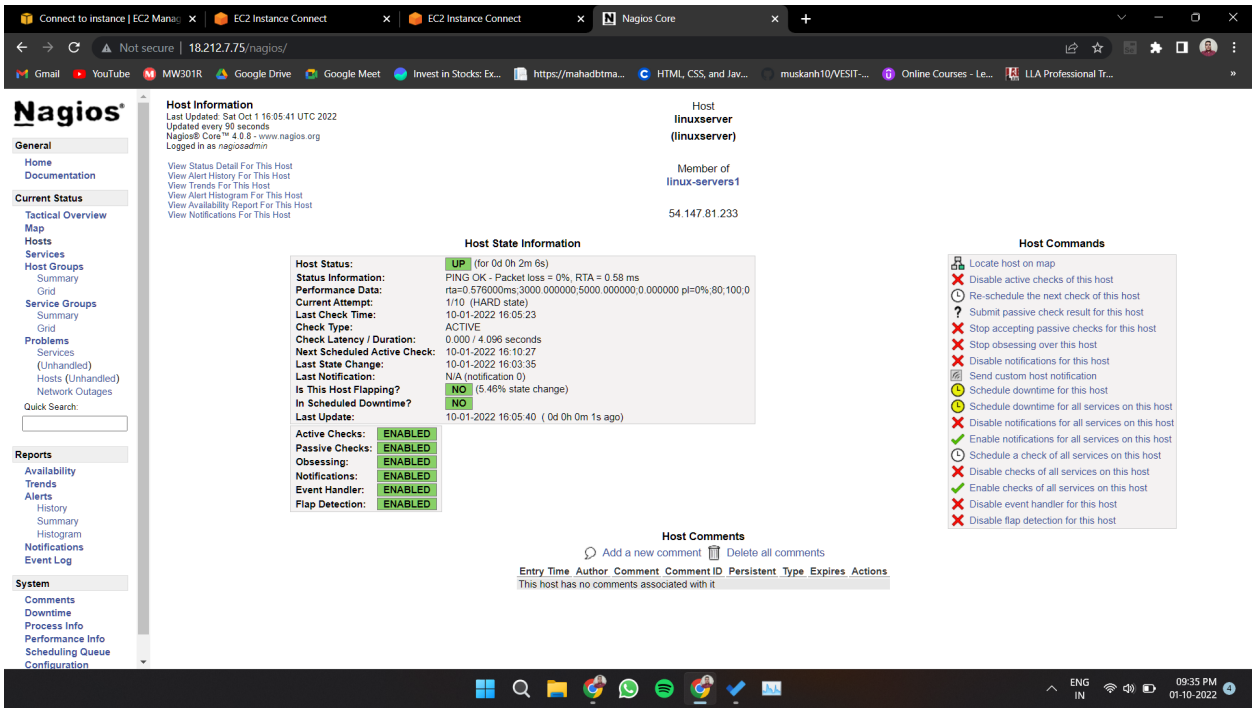
Step 15: Now, check your nagios dashboard and you'll see a new host being added.
Click on Hosts.



You will such a page.

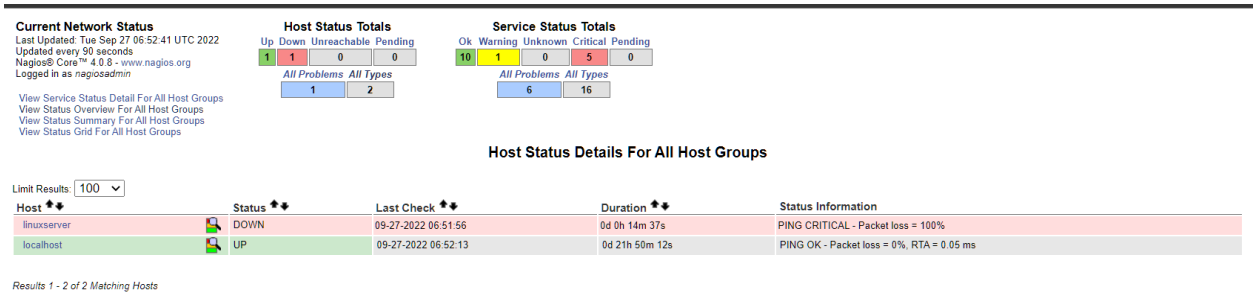


You can click Services to see all services and ports being monitored.



Finally we monitored the client using the nagios tool.

Note- If you get any error like this



Current Network Status
Last Updated: Tue Sep 27 06:52:41 UTC 2022
Updated every 90 seconds
Nagios® Core™ 4.0.8 - www.nagios.org
Logged in as *nagiosadmin*

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
1	1	0	0

All Problems	All Types
1	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
10	1	0	5	0

All Problems	All Types
6	16

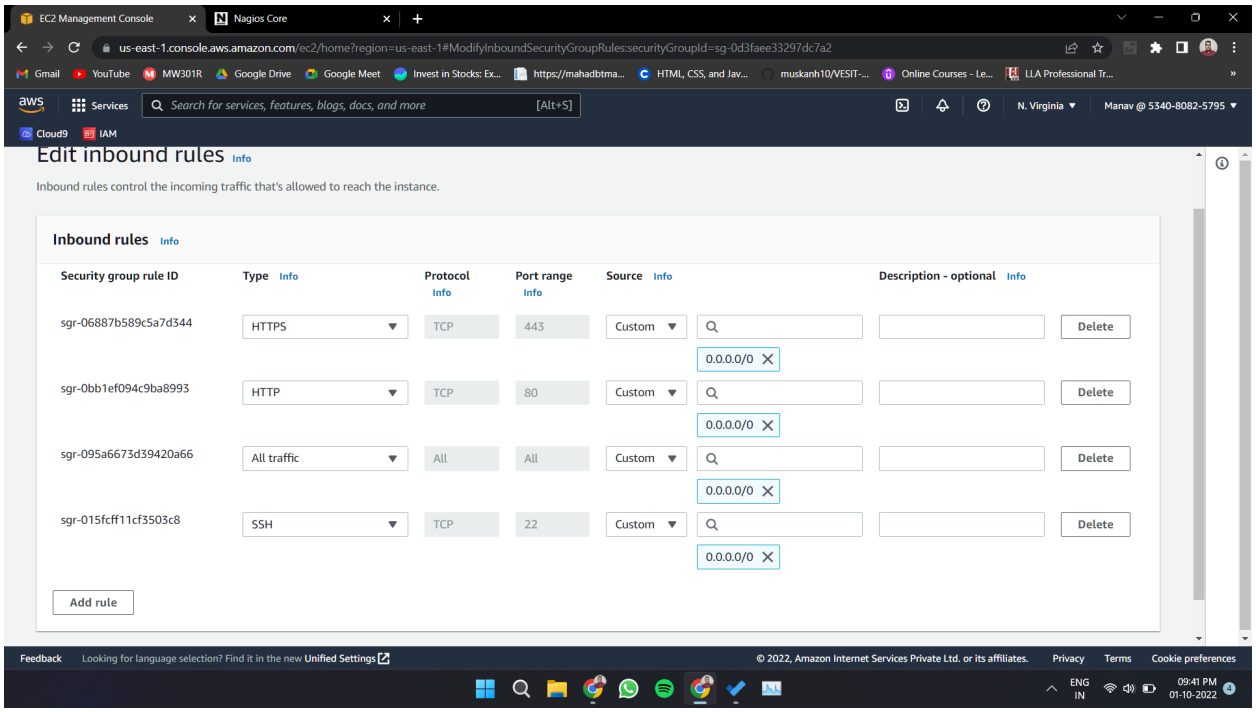
Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
linuxserver	DOWN	09-27-2022 06:51:56	0d 0h 14m 37s	PING CRITICAL - Packet loss = 100%
localhost	UP	09-27-2022 06:52:13	0d 21h 50m 12s	PING OK - Packet loss = 0%, RTA = 0.05 ms

Results 1 - 2 of 2 Matching Hosts

Then , edit the inbound and outbound rules of the security groups
Inbound rules:



EC2 Management Console | Nagios Core
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-0d3faee33297dc7a2

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

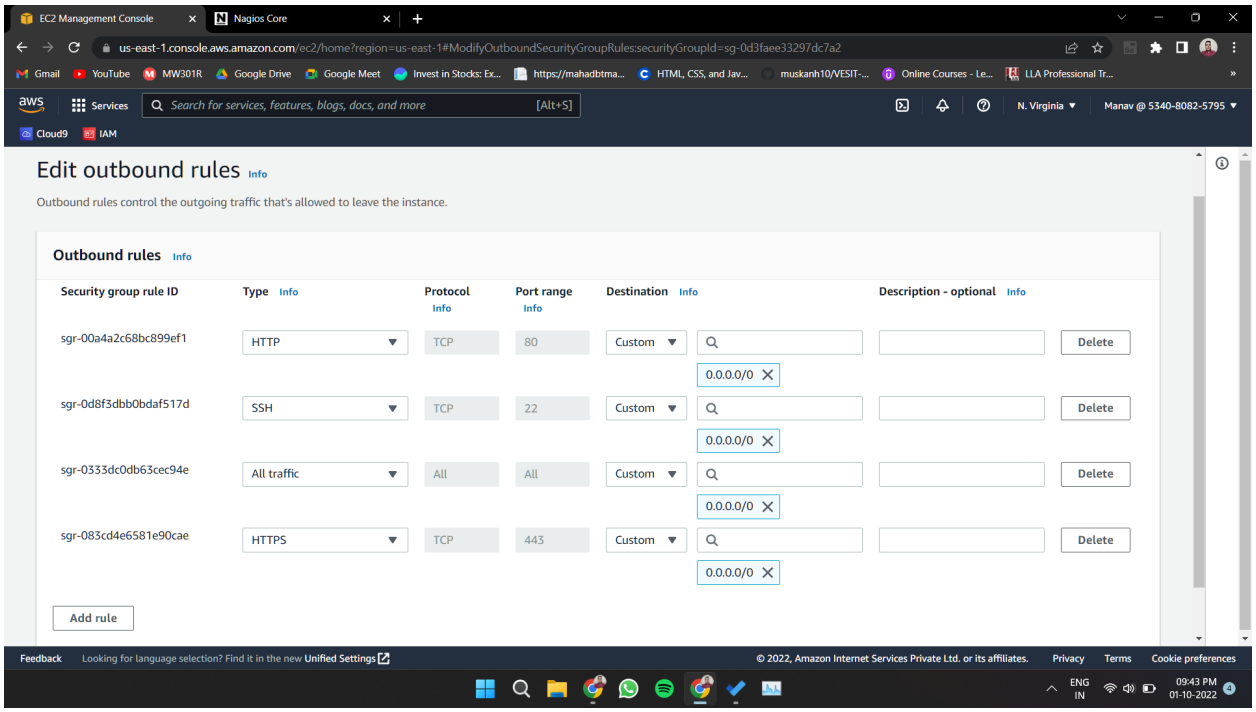
Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-06887b589c5a7d344	HTTPS	TCP	443	Custom	0.0.0.0/0	Delete
sgr-0bb1ef094c9ba8993	HTTP	TCP	80	Custom	0.0.0.0/0	Delete
sgr-095a6673d39420a66	All traffic	All	All	Custom	0.0.0.0/0	Delete
sgr-015fcff11cf3503c8	SSH	TCP	22	Custom	0.0.0.0/0	Delete

Add rule

Feedback | Looking for language selection? Find it in the new Unified Settings | © 2022, Amazon Internet Services Private Ltd. or its affiliates. | Privacy | Terms | Cookie preferences

Outbound rules:



Also , after the experiment terminate all the instances which were used.

Conclusion:

We ~~dont~~ launched the nagios server in the same manner as in the prior experiment, creating two files in the root user. We modified the nagios config file as well as the linux server.cfg as necessary. Then we restarted the nagios service, checked the configuration files, connected to EC2 instance, made sure the system was up to date and installed the necessary packages. After making modifications to .cfg, we restarted the NRPE server. As a result, we ~~effic~~ effectively included the best and monitored the client.