

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:

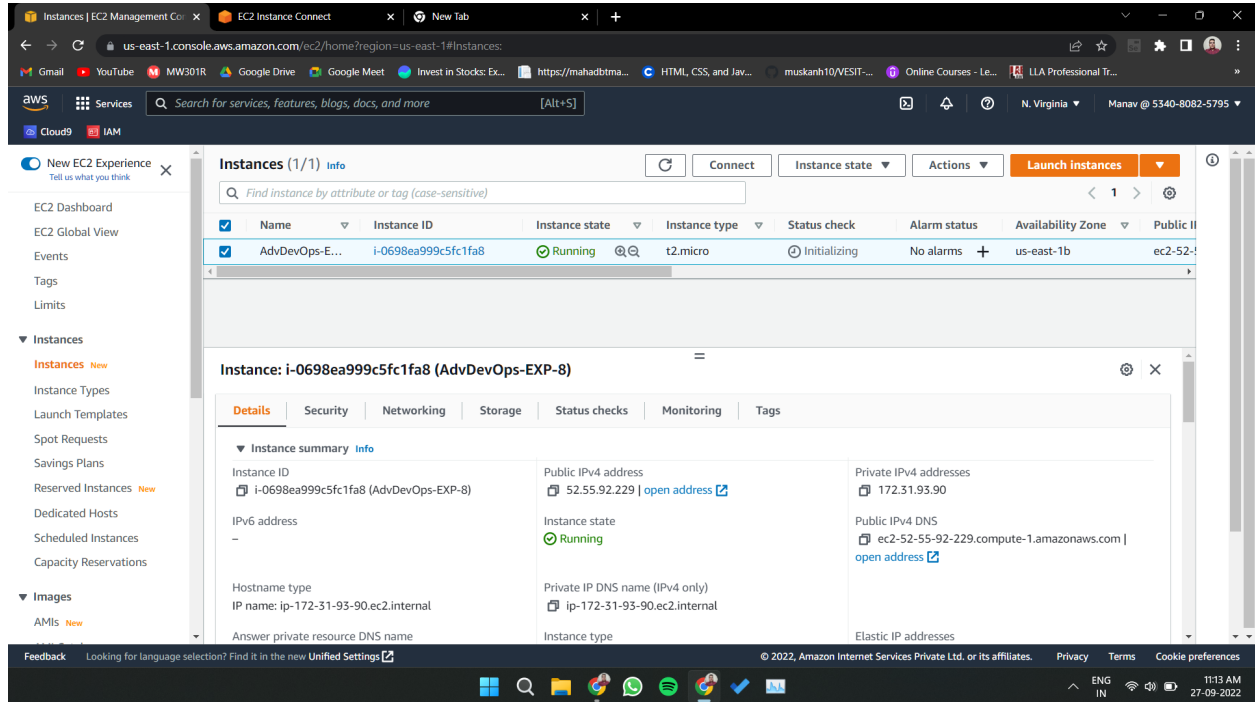
Since we had already gone through the introduction, features of ~~ag~~ nagios, here we will discuss the importance of ~~ag~~ nagios and how it helps in service monitoring. The importance is as follows:

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)
  Main PID: 3028
   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: services: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=325121; 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/printenv: 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**”

Name and tags

Name

AdvDevOps-EXP-10-Client

Add additional tags

▼ Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

aws

macOS

Mach

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type

ami-0149b2da6ceec4bb0 (64-bit (x86)) / ami-00266f51b6b22db58 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

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

Network settings Info Edit

Network Info
vpc-05c2f17b2b50f83b2

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups Info
Select security groups ▼ Compare security group rules

launch-wizard-2 sg-0d3faee33297dc7a2 X
VPC: vpc-05c2f17b2b50f83b2

Security groups that you add or remove here will be added to or removed from all your network interfaces.

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

```
[ec2-user@ip-172-31-93-90 ~]$ ps -ef | grep nagios
nagios    3028      1  0 05:40 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    3030    3028  0 05:40 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3031    3028  0 05:40 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3032    3028  0 05:40 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3033    3028  0 05:40 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    3034    3028  0 05:40 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user  3769    3502  0 06:01 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-93-90 ~]$
```


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 ~]# sudo su
[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

```
[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   ^R Cut Text   ^J Justify    ^C Cur Pos   M-U Undo     M-R Mark Text  M-I To Bracket M-^ Previous
^X Exit      ^F Read File  ^E Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line  M-S Redo     M-C 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 to  
    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
# This option determines where object definitions are cached when
```

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

```
# 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, 18.212.7.75
```

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.

Nagios®

General

[Home](#)
[Documentation](#)

Current Status

[Tactical Overview](#)
[Map](#)
[Hosts](#)
[Services](#)
[Host Groups](#)
 [Summary](#)
 [Grid](#)
[Service Groups](#)
 [Summary](#)
 [Grid](#)
[Problems](#)
 [Services](#)
 [\(Unhandled\)](#)
 [Hosts \(Unhandled\)](#)
 [Network Outages](#)

Quick Search:

You will such a page.

The screenshot displays the Nagios Core web interface in a browser window. The address bar shows the URL `18.212.7.75/nagios/`. The interface includes a left-hand navigation menu with sections: General, Current Status, Reports, and System. The 'Current Status' section is expanded, showing links to Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, and Quick Search. The main content area displays the 'Current Network Status' (last updated Sat Oct 1 16:05:33 UTC 2022), 'Host Status Totals' (Up: 2, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 11, Warning: 1, Unknown: 0, Critical: 2, Pending: 0). Below these, a table titled 'Host Status Details For All Host Groups' shows details for two hosts: 'linuxserver' and 'localhost', both with a status of 'UP'. The table columns include Host, Status, Last Check, Duration, and Status Information. The bottom of the interface shows a Windows taskbar with the time 09:35 PM on 01-10-2022.

Current Network Status
Last Updated: Sat Oct 1 16:05:33 UTC 2022
Updated every 50 seconds
Nagios® Core™ 4.0.8 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
11	1	0	2	0

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-01-2022 16:05:23	0d 0h 1m 58s	PING OK - Packet loss = 0%, RTA = 0.58 ms
localhost	UP	10-01-2022 16:03:22	5d 7h 3m 4s	PING OK - Packet loss = 0%, RTA = 0.05 ms

Results 1 - 2 of 2 Matching Hosts

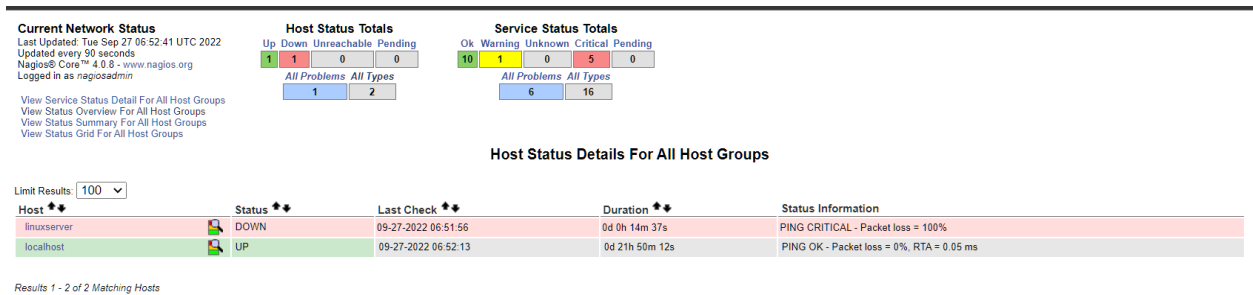
You can click Services to see all services and ports being monitored.

The screenshot displays the Nagios Core web interface in a browser window. The browser's address bar shows the URL `18.212.7.75/nagios/`. The interface is divided into several sections:

- Host Information:** Located at the top left, it shows the host name `linuxserver`, its IP address `54.147.81.233`, and the member group `linux-servers1`. It also indicates the last update time and the user logged in as `nagiosadmin`.
- Host State Information:** This section provides a detailed overview of the host's current state. It shows the host status as **UP** (for 0d 0h 2m 6s). Key performance data includes PING OK, packet loss of 0%, and RTA of 0.58 ms. The current attempt is 1/10 (HARD state), and the last check time is 10-01-2022 16:05:23. The check type is ACTIVE, and the check latency/duration is 0.000 / 4.096 seconds. The next scheduled active check is at 10-01-2022 16:10:27. The last state change was at 10-01-2022 16:03:35, and the last notification was N/A (notification 0). The host is not flapping, and there is no scheduled downtime. The last update was at 10-01-2022 16:05:40 (0d 0h 0m 1s ago).
- Host Commands:** This section lists various commands that can be executed on the host, such as "Locate host on map", "Disable active checks of this host", "Re-schedule the next check of this host", "Submit passive check result for this host", "Stop accepting passive checks for this host", "Stop obsessing over this host", "Disable notifications for this host", "Send custom host notification", "Schedule downtime for this host", "Schedule downtime for all services on this host", "Disable notifications for all services on this host", "Enable notifications for all services on this host", "Schedule a check of all services on this host", "Disable checks of all services on this host", "Enable checks of all services on this host", "Disable event handler for this host", and "Disable flap detection for this host".
- Host Comments:** This section allows users to add or delete comments for the host. It shows a table with columns for Entry Time, Author, Comment, Comment ID, Persistent, Type, Expires, and Actions. Currently, there are no comments associated with this host.
- Left Sidebar:** This sidebar contains navigation links for various sections of the Nagios interface, including General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid, Service Groups, Summary, Grid, Problems, Services (Unhandled), Hosts (Unhandled), Network Outages, Quick Search, Reports, Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, System, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

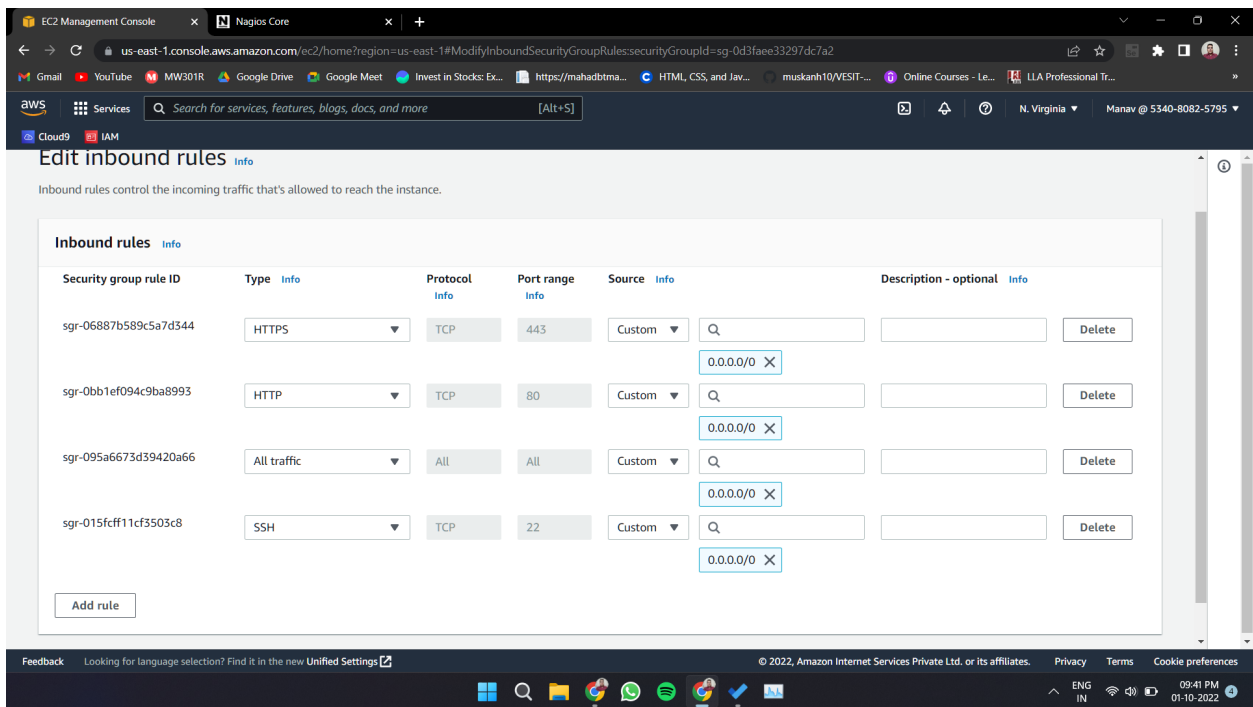
Finally we monitored the client using the nagios tool.

****Note- If you get any error like this****



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

Inbound rules:



Outbound rules:

Edit outbound rules [Info](#)

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
sgr-00a4a2c68bc899ef1	HTTP	TCP	80	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sgr-0d8f3dbb0bdaf517d	SSH	TCP	22	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sgr-0333dc0db63cec94e	All traffic	All	All	Custom <input type="text" value="0.0.0.0/0"/>		Delete
sgr-083cd4e6581e90cae	HTTPS	TCP	443	Custom <input type="text" value="0.0.0.0/0"/>		Delete

[Add rule](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

09:43 PM 01-10-2022

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

Conclusion :

We ~~don't~~ 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 leaves.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 ~~offic~~ effectively included the host and monitored the client.