Name: Harsh Dalvi Subject: Advance DevOps , Sem: SEM V
Roll No: 13 Class / Batch: TE-IT / Batch B

Experiment No. 9

Aim: To explain Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine. **(LO1, LO5)**

Theory:

What is Nagios?

- Nagios is a free and open-source software application for computer systems. It is used for monitoring the systems, networks and infrastructure.
- Its original name was NetSaint developed by Ethan Galstad with the group of some developers in the year 1999.
- This software application mainly provides the services of monitoring and alerting for switches, applications, and servers inside the DevOps culture.
- It is also used to notify the users when the things go bad, and also alerts them when the things become better.
- This software application is written in C language, which was mainly designed to run under the Linux operating systems. But it can also run under Unix and Windows operating system.

This software application is available in its following two variants:

1. **Nagios Core:** The Nagios is also known as a Nagios Core. It is an open-source product. It is released on 1st March in the year 2002.
2. **Nagios XI:** It is an extended version of Nagios. It provides advanced components and tools for monitoring. This variant needs a software license.

Why to use Nagios?

Following are some reasons for using the Nagios software application:

1. It helps the users of this software application to easily find the root cause of any problem.
2. It is also used to detect all the possible networks.
3. We can easily maintain the issues of security and also detect the availability of the services.
4. This application automatically fix the problems or issues when occurs.
5. Before the failure of a system, it helps you to update the infrastructure.
6. Using this application, user can quickly detect any type of infrastructure issues.
7. It also monitors the various servers of database such as SQL Server, MySQL.

Features of Nagios

Following are the various features or characteristics of Nagios application:

1. This software application perfectly stores the data, because there is a log management system.
2. Plugins of data graphing are available in this application.
3. It defines the event handlers which are running during the services.
4. It has an optional web interface for viewing the notifications, log files, etc.

5. Users can also set up this application in 'distribute' system, if they require to have a monitoring system in multiple locations for monitoring.
6. It helps the users for defining the hierarchy of networks hosts using parent hosts.
7. It monitors the various services of a network such as SMTP, HTTP, FTP, SSH.
8. It also monitors the various resources of a server such as memory, processor, and disk drives, system logs.
9. Users can easily monitor or detect the whole infrastructure of IT and business process with a single pass.
10. It provides the remote monitoring through SSL and SSH encrypted tunnels.

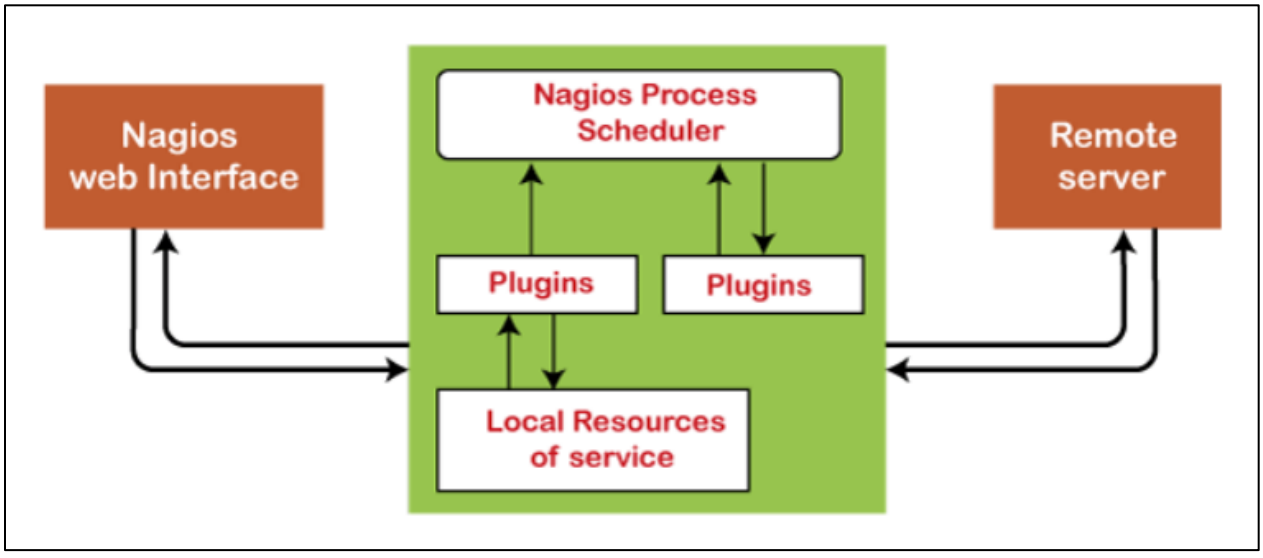
Architecture of Nagios:

The architecture of Nagios is built on the basis of server-client architecture.
The server of Nagios usually run on a host and the plugins run on the remote server/ or remote host which are to be monitored.

The plugins of Nagios collect the useful data and send them to the process scheduler, which displays the information over the graphical user interface (GUI).

Following are the three main components in the architecture of Nagios application:

1. Scheduler
2. GUI
3. Plugin.



1. **Scheduler:** The scheduler is the server part of the Nagios system. This component checks the plugins at regular interval and according to their results perform some action. The scheduler is a component of server part of Nagios. It sends a signal to execute the plugins at the remote host.

2. **GUI:** It is a user interface of the Nagios system, which is displayed on the web pages generated by the CGI. GUI can be a button to red or green, graph, sound, etc. The button of green color becomes red color on GUI, when the plugins returns an error or warning. When the soft alert is raised many times, a hard alert is raised, then the Nagios server sends a notification to the administrator.

3. **Plugins:** Plugins is a component of the Nagios system, which is configurable by the users. This component detects the services and return the results to the server of Nagios. Nagios plugins provide low-level intelligence on how to monitor anything and everything with Nagios Core. Plugins operate acts as a standalone application, but they are designed to be executed by Nagios Core. It connects to Apache that is controlled by CGI to display the result. Moreover, a database connected to Nagios to keep a log file.

Advantages of Nagios Application

Following are the benefits or advantages of Nagios:

1. This application is an open-source software, so we can use and edit it freely.
2. It provides various plugins, which are free to download and develop.
3. You can easily understand the plugin architecture.
4. The main advantage of this application software is that, it quickly detects the failed services, servers, and the batch jobs.
5. It also monitors or detects the failures of network and protocols quickly.
6. This software application also handles the warnings and critical situations.
7. We can also set up a monitoring system on various machines across multiple locations, so that they communicate all their outputs to the central Nagios server.

Disadvantages of Nagios:

Following are the limitations or disadvantages of Nagios:

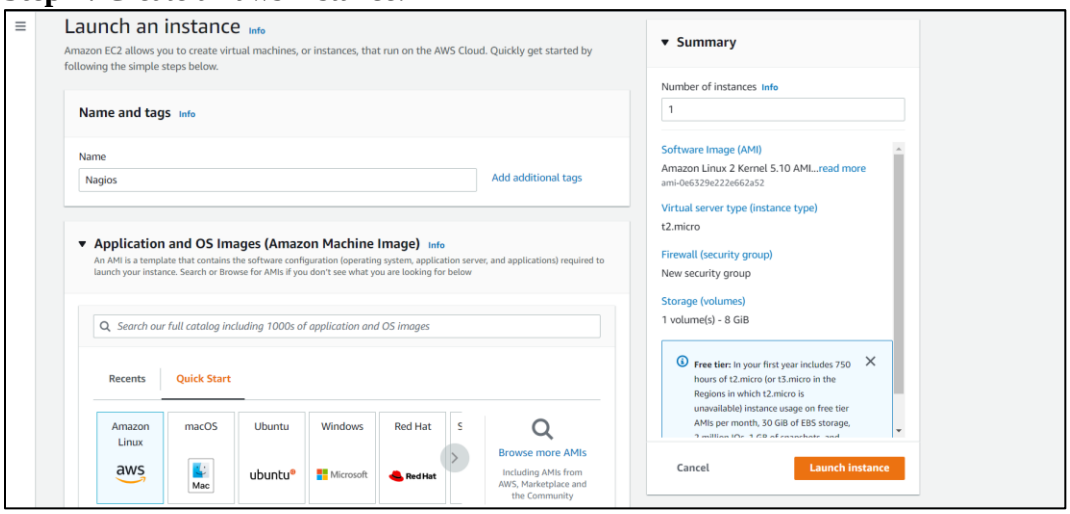
1. If we want to add the advance features of Nagios, then these features are not available in its free version, but only available on the application of Nagios XI, which is very expensive to use.
2. The interface of the Nagios core is confusing.
3. This application software cannot manage the network but only monitor the network.
4. It cannot monitor the throughput of a network.
5. In this application, there are various files of configuration, which are very hard for the user to configure them.
6. This Nagios application treats each device as a host.

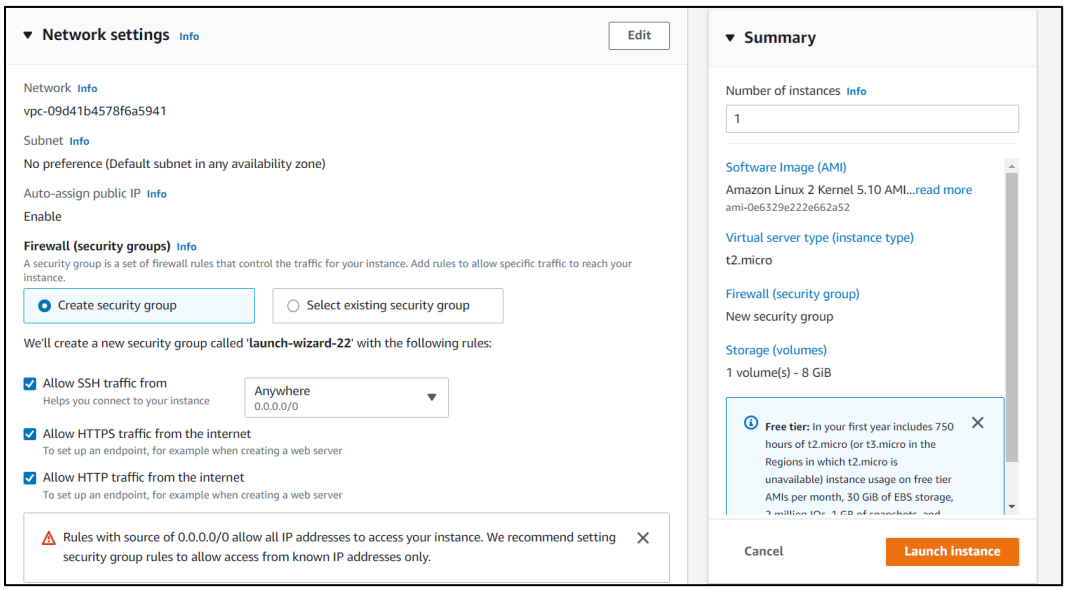
Name: Harsh Dalvi
Roll No: 13

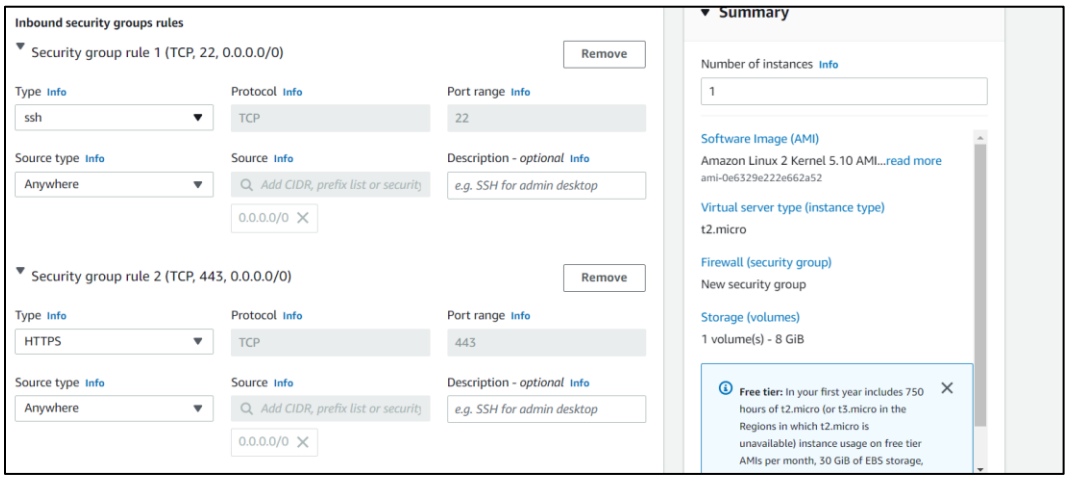
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

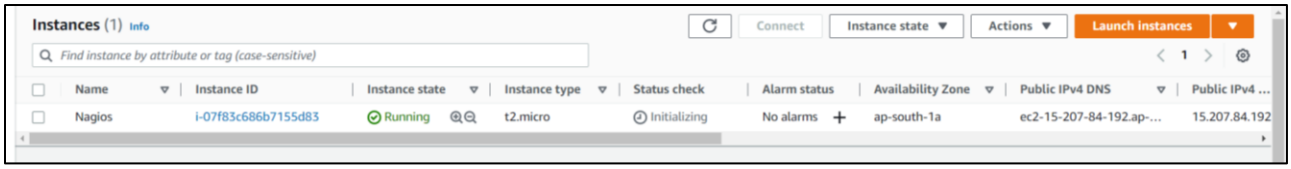
Steps to perform the Experiment:

Step 1: Create an aws instance.

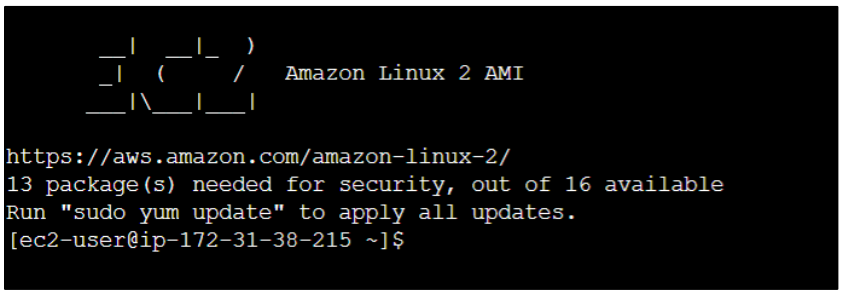


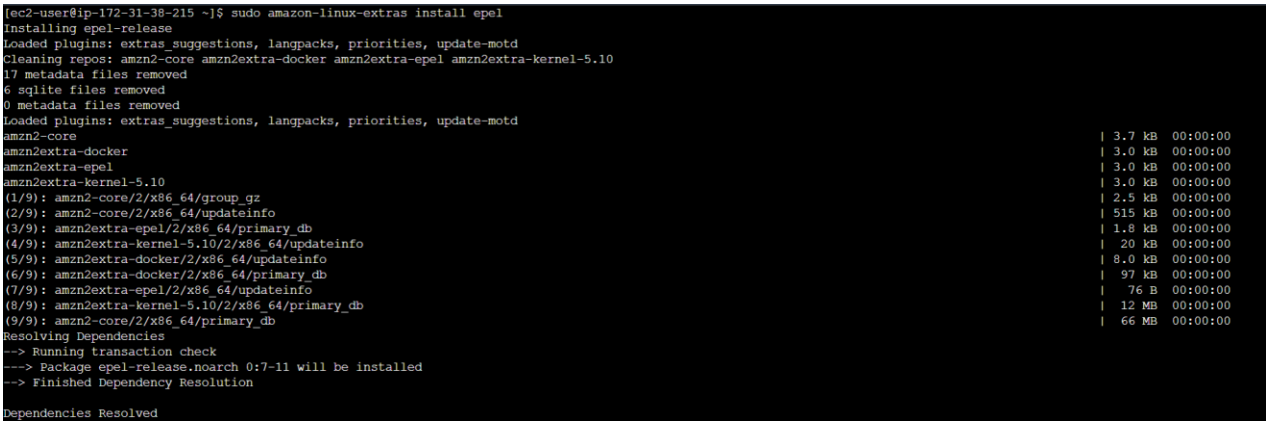




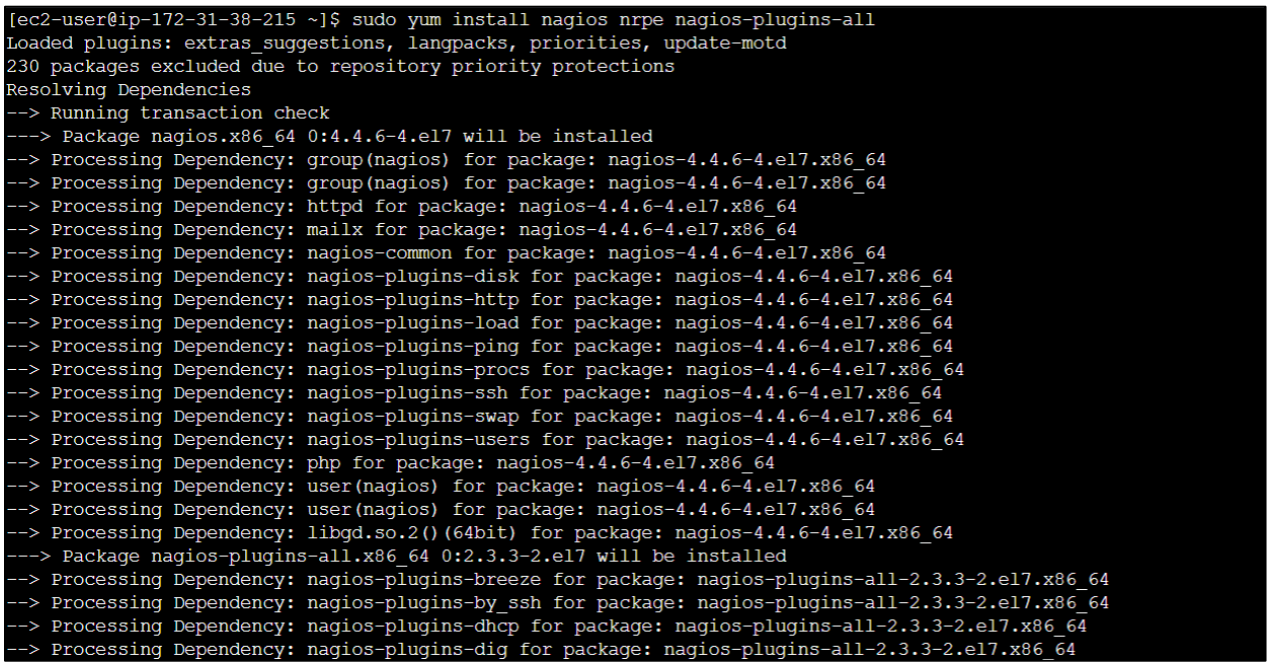


Step 2: Connect terminal.





Step 3: Install Nagios plugin.



Step 4: Auto Start Nagios.

```
[ec2-user@ip-172-31-38-215 ~]$ sudo su
[root@ip-172-31-38-215 ec2-user]# chkconfig --level 3 nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/nagios.service to /usr/lib/systemd/system/nagios.service.
```

Step 5: Install Httpd.

```
[root@ip-172-31-38-215 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
230 packages excluded due to repository priority protections
Package httpd-2.4.54-1.amzn2.x86_64 already installed and latest version
Nothing to do
```

Step 6: Start Httpd.

```
[root@ip-172-31-38-215 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
```

```
[root@ip-172-31-38-215 ec2-user]# chkconfig httpd on
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
```

Step 7: Install php.

```
[root@ip-172-31-38-215 ec2-user]# yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
230 packages excluded due to repository priority protections
Package php-5.4.16-46.amzn2.0.2.x86_64 already installed and latest version
Nothing to do
```

Step 8: To Create contact. Open contact cfg.

```
[root@ip-172-31-38-215 ec2-user]# vi /etc/nagios/objects/contact.cfg
```

Step 9: Edit contact details.

```
define contact{
contact_name     nagiosadmin
use              generic-contact
alias            Nagios Admin
email            harsh@gmail.com
}
~
```

Step 10: Check Nagios configuration. If any error found, resolve errors by restarting.

```
[root@ip-172-31-38-215 ec2-user]# /usr/sbin/nagios -v /etc/nagios/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
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 8 services.
        Checked 1 hosts.
        Checked 1 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 1 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
```

Step 11: Rechecking if the errors found in Nagios configuration are solved.

Step 12: Check Nagios Status.

```
[root@ip-172-31-38-215 ec2-user]# service nagios status
Redirecting to /bin/systemctl status nagios.service
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://www.nagios.org/documentation
```

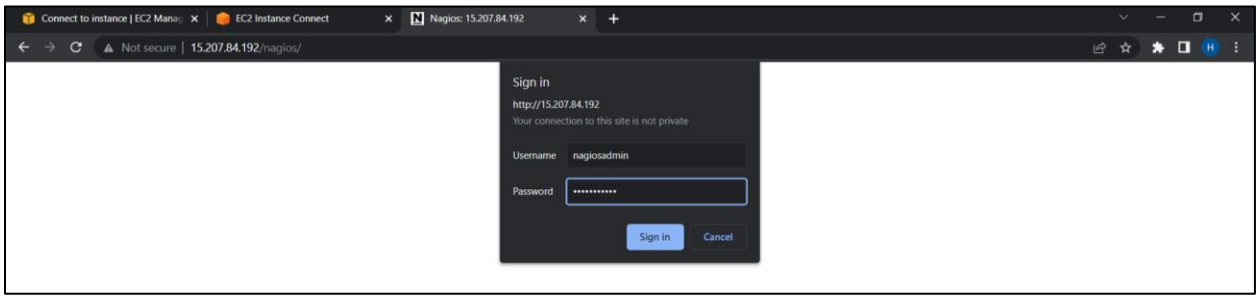
```
[root@ip-172-31-38-215 ec2-user]# service nagios start
Redirecting to /bin/systemctl start nagios.service
```

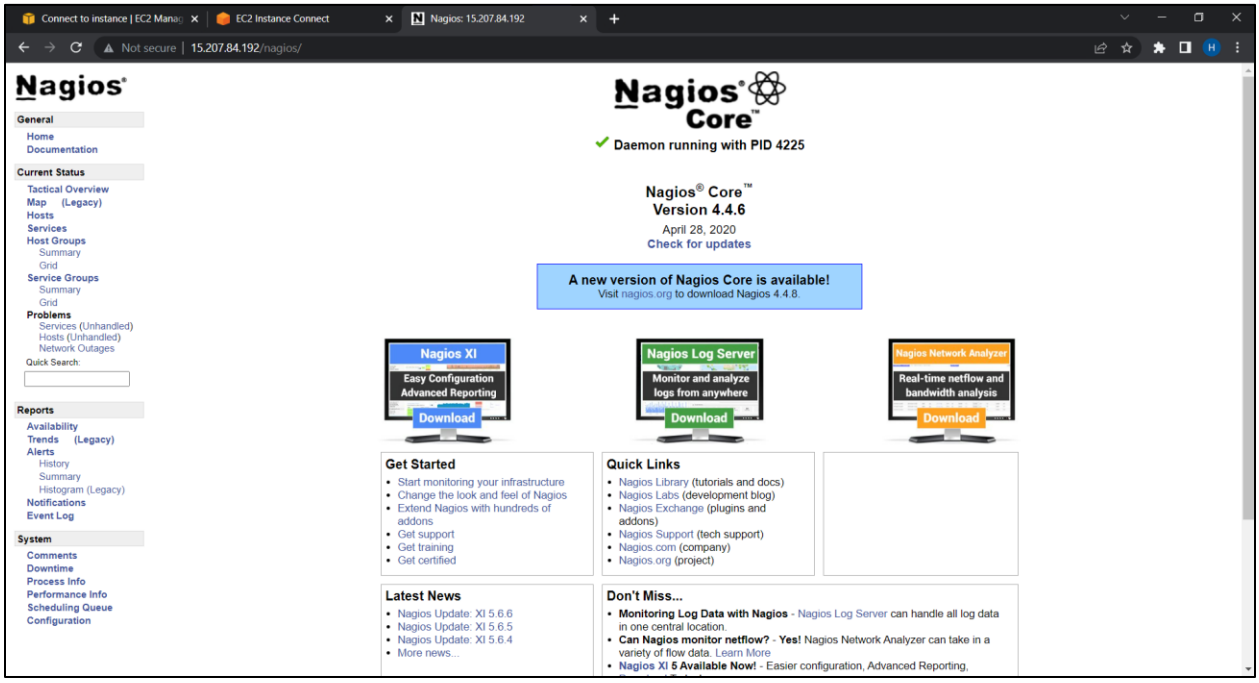
```
[root@ip-172-31-38-215 ec2-user]# service nagios status
Redirecting to /bin/systemctl status nagios.service
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-10-27 13:42:38 UTC; 40s ago
     Docs: https://www.nagios.org/documentation
  Process: 4222 ExecStart=/usr/sbin/nagios -d /etc/nagios/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 4221 ExecStartPre=/usr/sbin/nagios -v /etc/nagios/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 4225 (nagios)
   CGroup: /system.slice/nagios.service
           ├─4225 /usr/sbin/nagios -d /etc/nagios/nagios.cfg
           ├─4226 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           ├─4227 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           ├─4228 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           ├─4229 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           └─4230 /usr/sbin/nagios -d /etc/nagios/nagios.cfg

Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: qh: Socket '/var/spool/nagios/cmd/nagios.qh' successfully initialized
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: qh: core query handler registered
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: qh: echo service query handler registered
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: qh: help for the query handler registered
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: wproc: Successfully registered manager as @wproc with query handler
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: wproc: Registry request: name=Core Worker 4229;pid=4229
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: wproc: Registry request: name=Core Worker 4228;pid=4228
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: wproc: Registry request: name=Core Worker 4227;pid=4227
Oct 27 13:42:38 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: wproc: Registry request: name=Core Worker 4226;pid=4226
Oct 27 13:42:39 ip-172-31-38-215.ap-south-1.compute.internal nagios[4225]: Successfully launched command file worker with pid 4230
```

Step 13: Enable Service

```
[root@ip-172-31-38-215 ec2-user]# chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
```

Step s14: Sign in to Nagios monitoring tool, nagios window will open.





Conclusion: From this experiment, it is concluded that we have understood the concepts about Continuous Monitoring and Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor). In this experiment we have installed and configured Nagios on Linux machine. The continuous monitoring tool which is NAGIOS and also about its feature, architecture, advantages & disadvantages is used for monitoring if any problem occur then it will alert the nodes connected to it because it is based on the active server. Hence, we have successfully achieved the Lab Outcome One and Five (LO1 and LO5). Also, we have achieved PO1, PO2, PO3, PO4, PO5, PO9, PO10 and PO12 from this experiment.