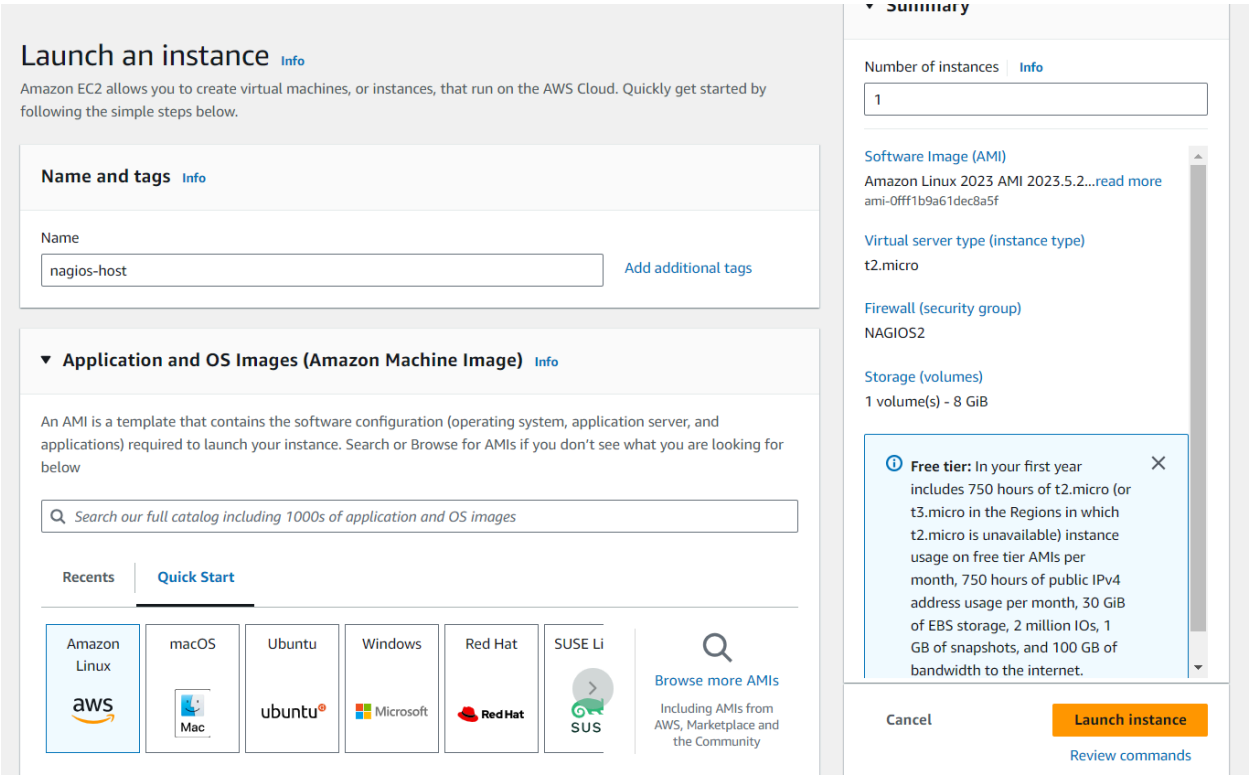
Veydant Sharma D15C 50 AdvDevop

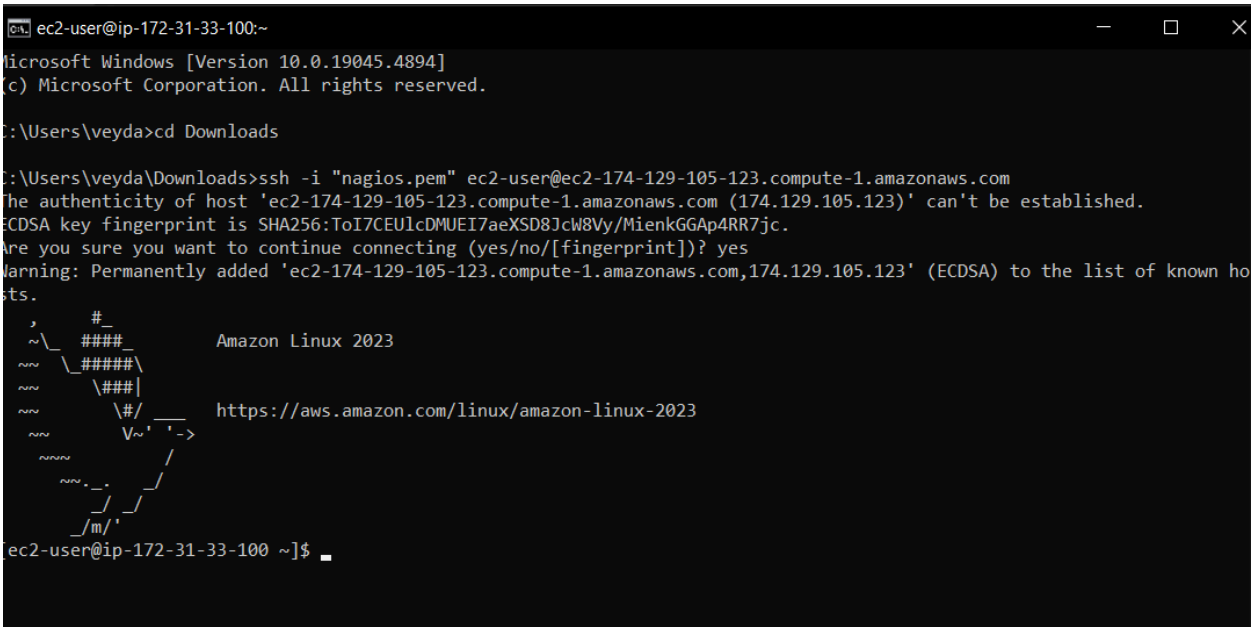
Experiment 9

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

Step 1: Create an ec2 instance, select amazon linux, t2.micro , select a key name and launch instance.



Step 2 : Connect to the instance using the SSH client, and copy and paste the command on the terminal



Step 3: First, run the following command:- sudo yum update This command will check for any updates for the YUM library.

```
[ec2-user@ip-172-31-35-21 ~]$ sudo yum update
Last metadata expiration check: 0:02:03 ago on Tue Oct  8 05:47:23 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

Step 4: Run the command: sudo yum install httpd php This installs an Apache server and a PHP on your instance.

```
[ec2-user@ip-172-31-35-21 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:02:58 ago on Tue Oct  8 05:47:23 2024.
Dependencies resolved.
================================================================================
 Package                 Arch      Version                         Repository      Size
================================================================================
Installing:
 httpd                   x86_64    2.4.62-1.amzn2023               amazonlinux     48 k
 php8.3                  x86_64    8.3.10-1.amzn2023.0.1           amazonlinux     10 k
Installing dependencies:
 apr                     x86_64    1.7.2-2.amzn2023.0.2            amazonlinux    129 k
 apr-util                x86_64    1.6.3-1.amzn2023.0.1            amazonlinux     98 k
 generic-logos-httpd     noarch    18.0.0-12.amzn2023.0.3          amazonlinux     19 k
 httpd-core              x86_64    2.4.62-1.amzn2023               amazonlinux    1.4 M
 httpd-filesystem        noarch    2.4.62-1.amzn2023               amazonlinux     14 k
```

Step 5: Run the command: sudo yum install gcc glibc glibc-common This installs the C/C++ compiler (GCC) along with the necessary C libraries required for compiling and running C programs.

```
[ec2-user@ip-172-31-35-21 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:03:21 ago on Tue Oct  8 05:47:23 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
================================================================================
 Package               Arch      Version                       Repository      Size
================================================================================
Installing:
 gcc                   x86_64    11.4.1-2.amzn2023.0.2         amazonlinux     32 M
Installing dependencies:
 annobin-docs          noarch    10.93-1.amzn2023.0.1          amazonlinux     92 k
 annobin-plugin-gcc    x86_64    10.93-1.amzn2023.0.1          amazonlinux    887 k
 cpp                   x86_64    11.4.1-2.amzn2023.0.2         amazonlinux     10 M
 gc                    x86_64    8.0.4-5.amzn2023.0.2          amazonlinux    105 k
```

Step 6: Run the command: sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-35-21 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:44 ago on Tue Oct  8 05:47:23 2024.
Dependencies resolved.
================================================================================
 Package                   Arch     Version                     Repository     Size
================================================================================
Installing:
 gd                        x86_64   2.3.3-5.amzn2023.0.3        amazonlinux   139 k
 gd-devel                  x86_64   2.3.3-5.amzn2023.0.3        amazonlinux    38 k
Installing dependencies:
 brotli                    x86_64   1.0.9-4.amzn2023.0.2        amazonlinux   314 k
 brotli-devel              x86_64   1.0.9-4.amzn2023.0.2        amazonlinux    31 k
 bzip2-devel               x86_64   1.0.8-6.amzn2023.0.2        amazonlinux   214 k
 cairo                     x86_64   1.17.6-2.amzn2023.0.1       amazonlinux   684 k
 cmake-filesystem          x86_64   3.22.2-1.amzn2023.0.4       amazonlinux    16 k
```

Step 7: Run the commands: sudo adduser -m nagios sudo passwd nagios

```
[ec2-user@ip-172-31-35-21 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-35-21 ~]$ sudo passwd -m nagios
passwd: bad argument -m: unknown option
[ec2-user@ip-172-31-35-21 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
```

```
[ec2-user@ip-172-31-35-21 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-35-21 ~]$ sudo usermod -a -G nagcmd apache
```

Step 10: mkdir ~/downloads cd ~/downloads This creates a directory named ‘downloads’, to store the files of the nagios server that are downloaded

```
[ec2-user@ip-172-31-35-21 ~]$ : mkdir ~/downloads
[ec2-user@ip-172-31-35-21 ~]$ cd ~/downloads
```

Step 11: wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz The above command installs the latest version of nagios-core
Step 12: wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz The above command installs the latest version of nagios-plugins.

```
[ec2-user@ip-172-31-35-21 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.g
--2024-10-09 05:42:49--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: ‘nagios-4.5.5.tar.gz.1’

nagios-4.5.5.tar.gz.1        100%[=============================================>]   1.97M  5.56MB/s    in 0.4s

2024-10-09 05:42:50 (5.56 MB/s) - ‘nagios-4.5.5.tar.gz.1’ saved [2065473/2065473]

[ec2-user@ip-172-31-35-21 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-09 05:43:04--  https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.4.11.tar.gz.1’

nagios-plugins-2.4.11.tar.gz. 100%[=============================================>]   2.62M  5.49MB/s    in 0.5s

2024-10-09 05:43:05 (5.49 MB/s) - ‘nagios-plugins-2.4.11.tar.gz.1’ saved [2753049/2753049]
```

Step 13:
tar zxvf nagios-4.5.5.tar.gz This extracts the nagios-core files into the same directory using the tar command.
Use ‘ls’ command to find the correct directory
run the ‘sudo yum install openssl-devel’ command.

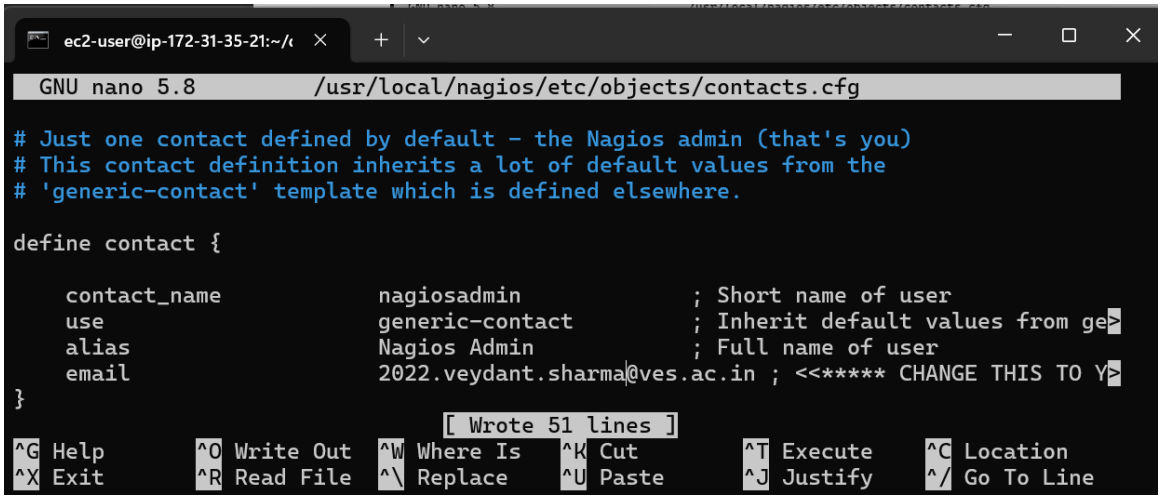
```
[ec2-user@ip-172-31-35-21 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-4.5.5.tar.gz.1  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-35-21 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 23:58:21 ago on Tue Oct  8 05:47:23 2024.
Package openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
```

Step 14: Next, we must compile all components of this software according to the instructions in the Makefile. To do so, use the following command: make all Then, sudo make install sudo make install-init sudo make install-config sudo make install-commandmode

```
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
make -C ../lib
```

```
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
```

Step 15: : We need to update the email linked with this server to our email for it to send notifications (if any needed). sudo nano /usr/local/nagios/etc/objects/contacts.cfg



Step 17: sudo make install-webconf This installs the necessary configuration files for the Nagios web interface.

Step 18: sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin This creates a user named ‘nagiosadmin’ to access the nagios web interface. Create a password and keep it in mind as it will be required in the future steps.

Step 19: Restart the apache server to apply all the recent configurations. sudo service httpd restart
Step 20: cd ~/downloads tar zxvf nagios-plugins-2.4.11.tar.gz This changes the directory to the ‘downloads’ directory and extracts the files for nagios-plugins

```
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$  sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-35-21 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-35-21 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
```

Step 21: cd nagios-plugins-2.4.11 ./configure --with-nagios-user=nagios --with-nagios-group=nagios This installs the configurations for the nagios-plugins files.

```
[ec2-user@ip-172-31-35-21 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-35-21 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
```

Step22:: sudo chkconfig --add nagios
sudo chkconfig nagios on

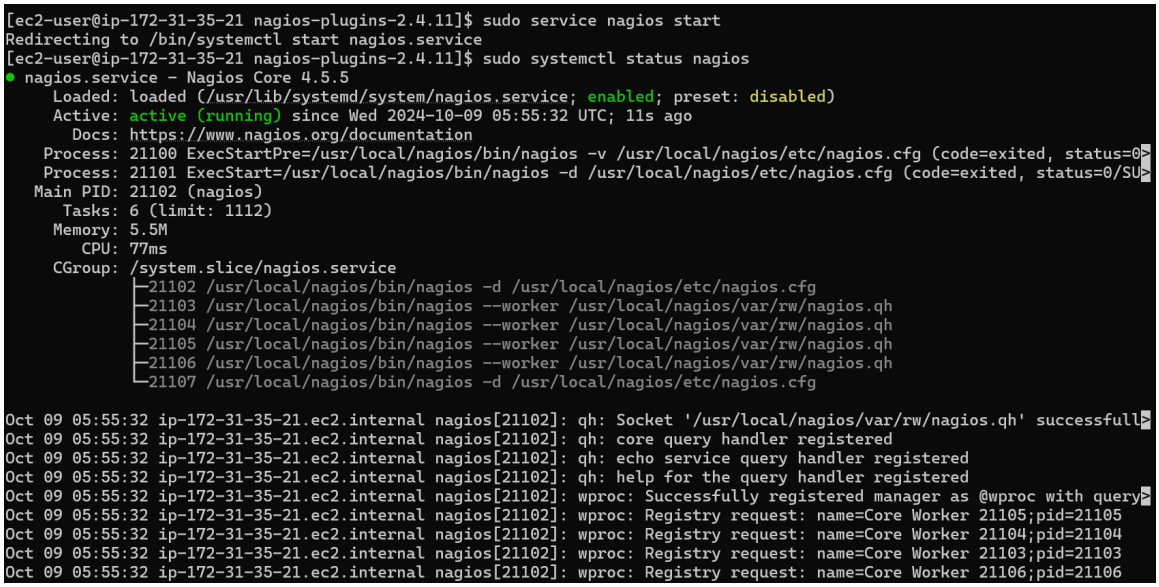
```
[ec2-user@ip-172-31-35-21 nagios-plugins-2.4.11]$  sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-35-21 nagios-plugins-2.4.11]$ sudo /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...
```

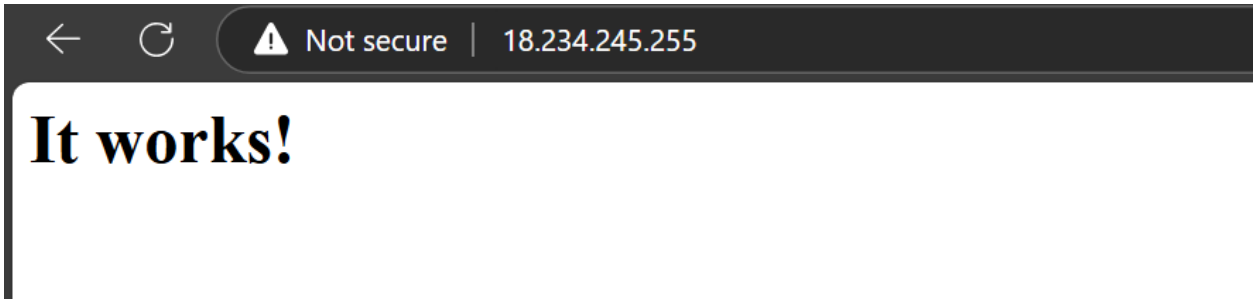
```
[ec2-user@ip-172-31-35-21 downloads]$ tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
```

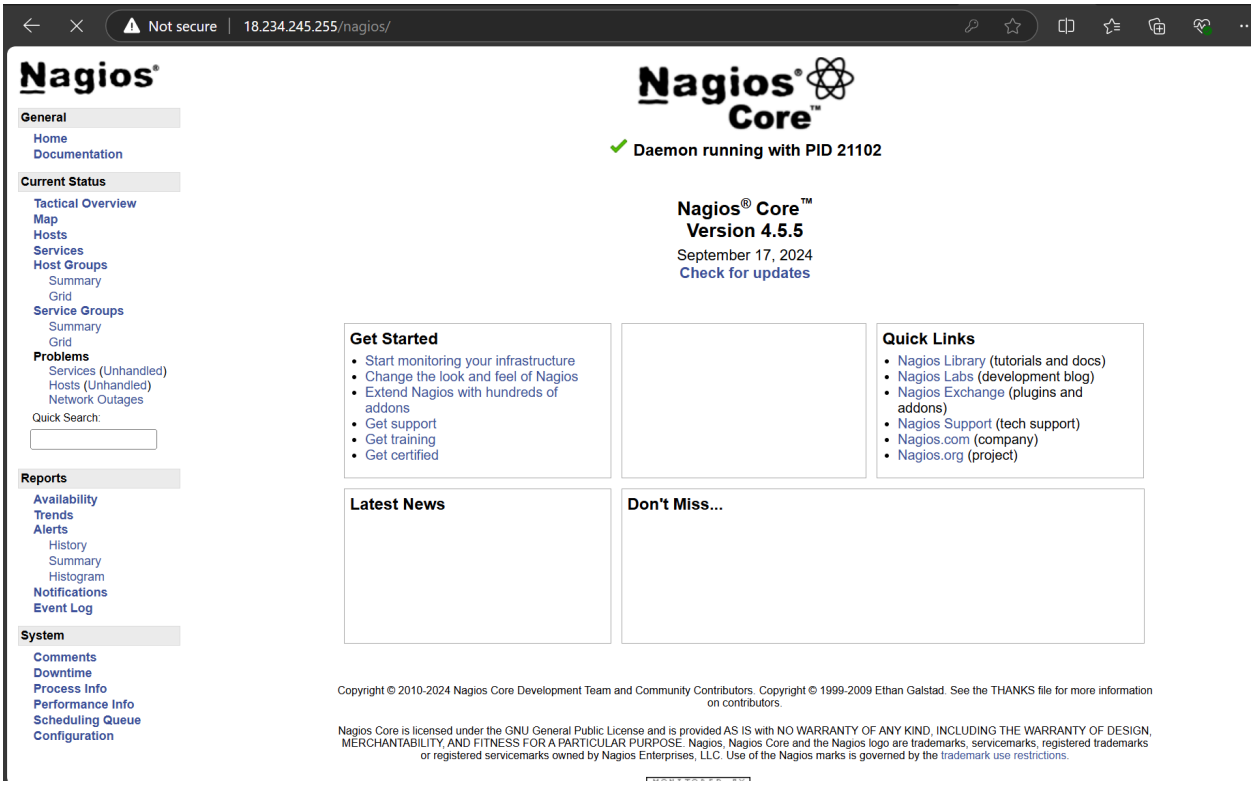
Step 23: sudo service nagios start This starts the Nagios service.
sudo systemctl status nagios This checks the status of Nagios. Ensure that it is ‘active(running)’.



Veydant Sharma D15C 50 AdvDevop

Step 24: In the address bar, enter ‘http://<ipv4 address>/nagios’.





Veydant Sharma D15C 50 AdvDevop

Conclusion:

- In the above experiment, we learned how to install and configure Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.
- We faced some issues with the configuration command, which were fixed with the right directory and the sudo yum install openssl-devel' command.
- Once the setup was complete, we hosted the Nagios server and accessed the Nagios dashboard by pasting the public IPv4 address of our instance in the browser