Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra

Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab
Experiment 09

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

Roll No.	22
Name	Sarthak Harade
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO5: To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.
Grade:	

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

What is Nagios?

Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately.

Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture.

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

● Detects all types of network or server issues

● Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem

● Active monitoring of your entire infrastructure and business processes

● Allows you to monitor and troubleshoot server performance issues

● Helps you to plan for infrastructure upgrades before outdated systems create failures

● You can maintain the security and availability of the service

● Automatically fix problems in a panic situation

Features of Nagios

Following are the important features of Nagios monitoring tool:

● Relatively scalable, Manageable, and Secure

● Good log and database system

● Informative and attractive web interfaces

● Automatically send alerts if condition changes

● If the services are running fine, then there is no need to do check that host is an alive

● Helps you to detect network errors or server crashes

● You can troubleshoot the performance issues of the server.

● The issues, if any, can be fixed automatically as they are identified during the monitoring process

● You can monitor the entire business process and IT infrastructure with a single pass

● The product's architecture is easy to write new plugins in the language of your choice

● Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files

● Utilizes topology to determine dependencies

● Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.

● Helps you to define network host hierarchy using parent hosts.

● Ability to define event handlers that runs during service or host events for proactive problem resolution.

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.

1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.

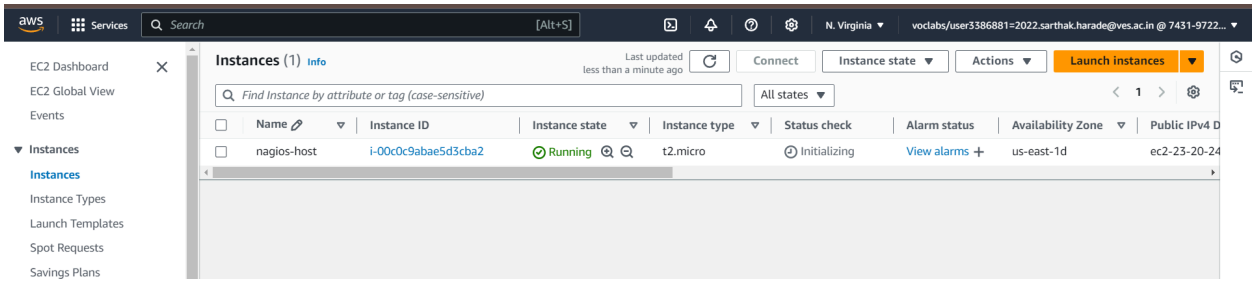
2. The plugin gets the status from the remote host

3. The plugin sends the data to the process scheduler

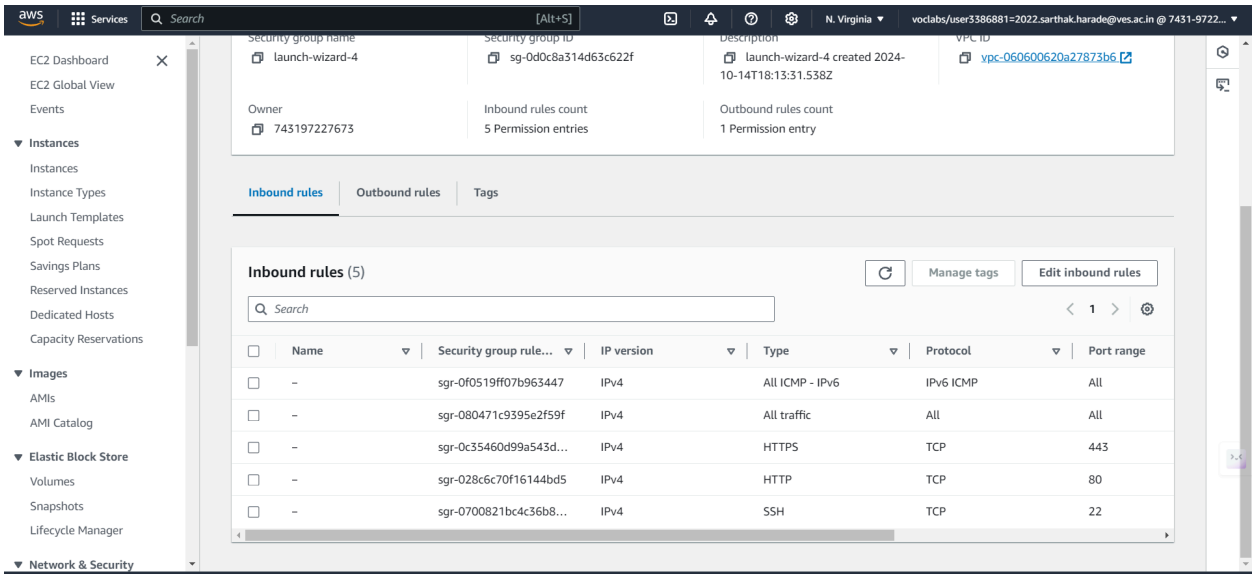
4. The process scheduler updates the GUI and notifications are sent to admins.

Steps:

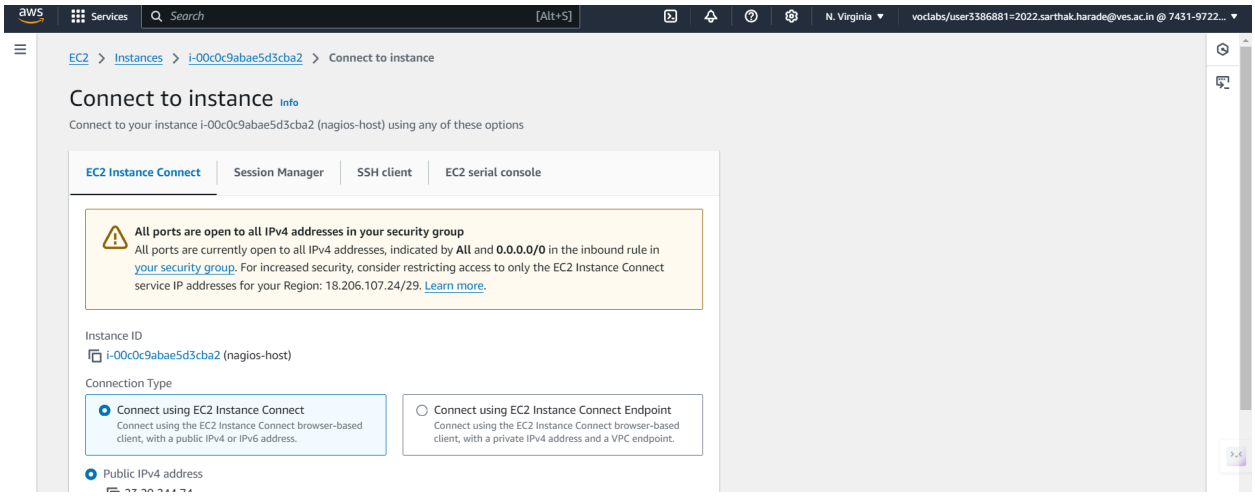
1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host

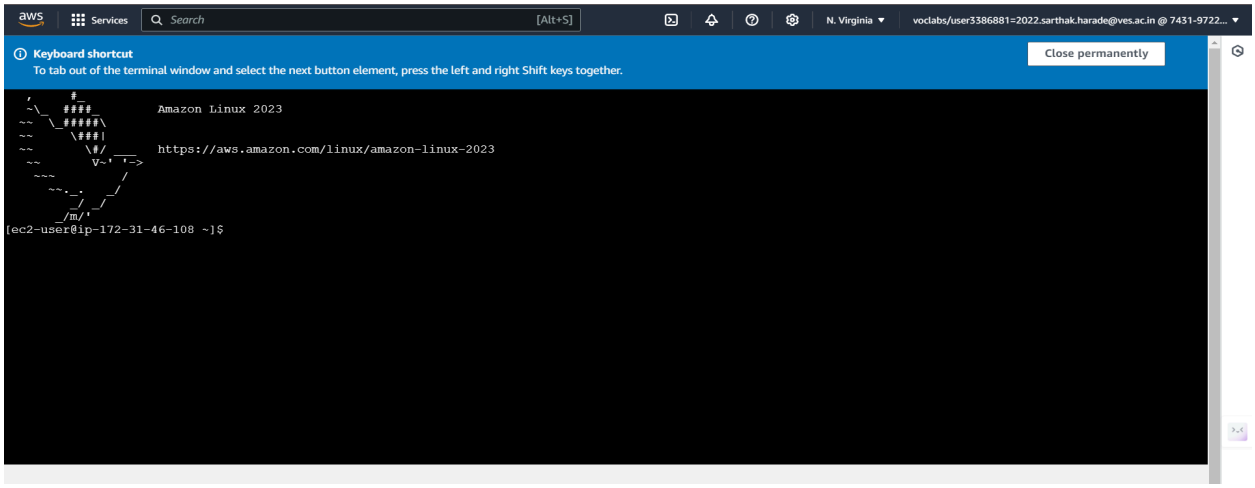


2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.



3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.





4. Update the package indices and install the following packages using yum

sudo yum update

sudo yum install httpd php

sudo yum install gcc glibc glibc-common

sudo yum install gd gd-devel

```
      _/ _/
    _/m/'
[ec2-user@ip-172-31-46-108 ~]$ sudo yum update
Last metadata expiration check: 0:07:23 ago on Mon Oct 14 18:15:00 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-46-108 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:08:09 ago on Mon Oct 14 18:15:00 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture        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
 httpd-tools             x86_64              2.4.62-1.amzn2023        amazonlinux         81 k
 libbrotli               x86_64              1.0.9-4.amzn2023.0.2     amazonlinux        315 k
 libsodium               x86_64              1.0.19-4.amzn2023        amazonlinux        176 k
 libxslt                 x86_64              1.1.34-5.amzn2023.0.2    amazonlinux        241 k
 mailcap                 noarch              2.1.49-3.amzn2023.0.3    amazonlinux         33 k
 nginx-filesystem        noarch              1:1.24.0-1.amzn2023.0.4  amazonlinux        9.8 k
 php8.3-cli              x86_64              8.3.10-1.amzn2023.0.1    amazonlinux        3.7 M
 php8.3-common           x86_64              8.3.10-1.amzn2023.0.1    amazonlinux        737 k
 php8.3-process          x86_64              8.3.10-1.amzn2023.0.1    amazonlinux         45 k
 php8.3-xml              x86_64              8.3.10-1.amzn2023.0.1    amazonlinux        154 k
Installing weak dependencies:
```

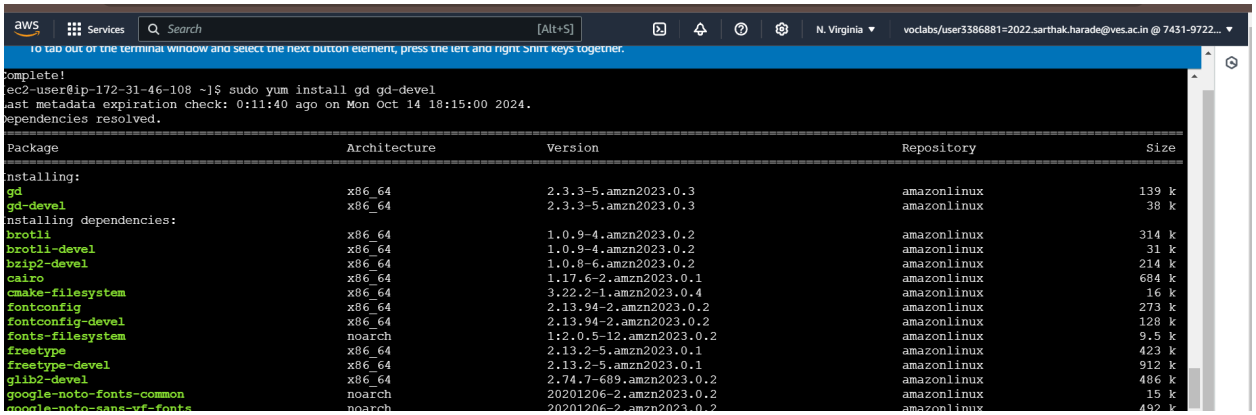
```
[ec2-user@ip-172-31-46-108 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:09:03 ago on Mon Oct 14 18:15:00 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

Package	Architecture	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
glibc-devel	x86_64	2.34-52.amzn2023.0.11	amazonlinux	27 k
glibc-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux	427 k
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.109-118.189.amzn2023	amazonlinux	1.4 M
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 k
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 k
libxcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 k
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux	534 k

Transaction Summary

Install 13 Packages

Total download size: 52 M



Package	Architecture	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
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fonts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912 k
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486 k
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15 k
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492 k

5. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.

sudo adduser -m nagios

sudo passwd nagios

```
Complete!
[ec2-user@ip-172-31-46-108 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-46-108 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-46-108 ~]$
```

6. Create a new user group

sudo groupadd nagcmd

7. Use these commands so that you don’t have to use sudo for Apache and Nagios

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-46-108 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-46-108 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-46-108 ~]$
```

8. Create a new directory for Nagios downloads

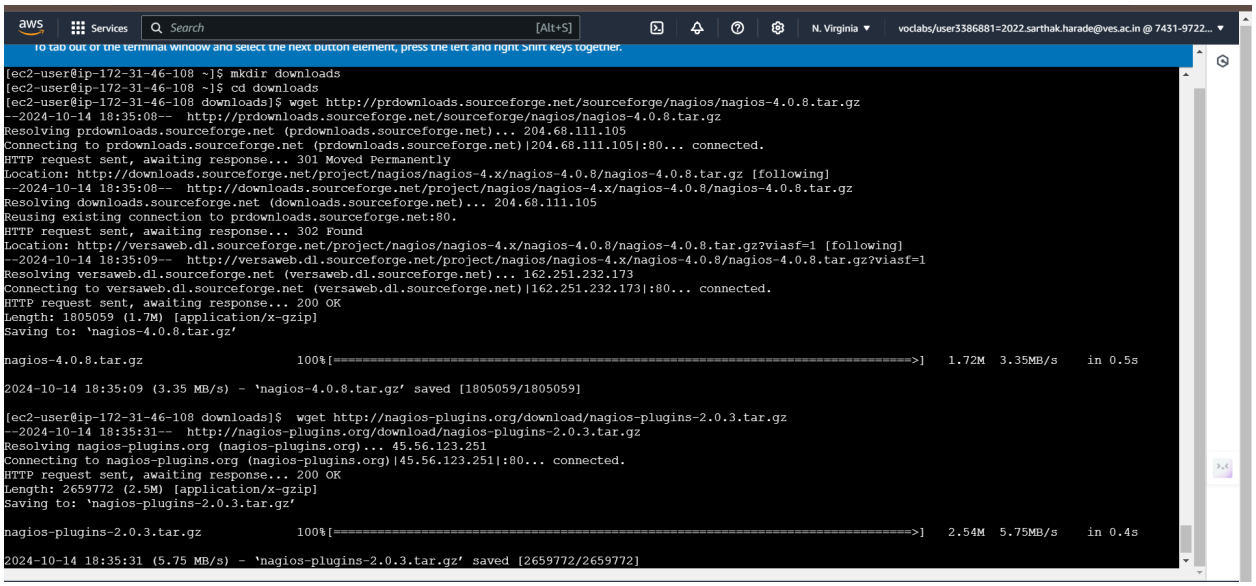
mkdir ~/downloads

cd ~/downloads

9. Use wget to download the source zip files.

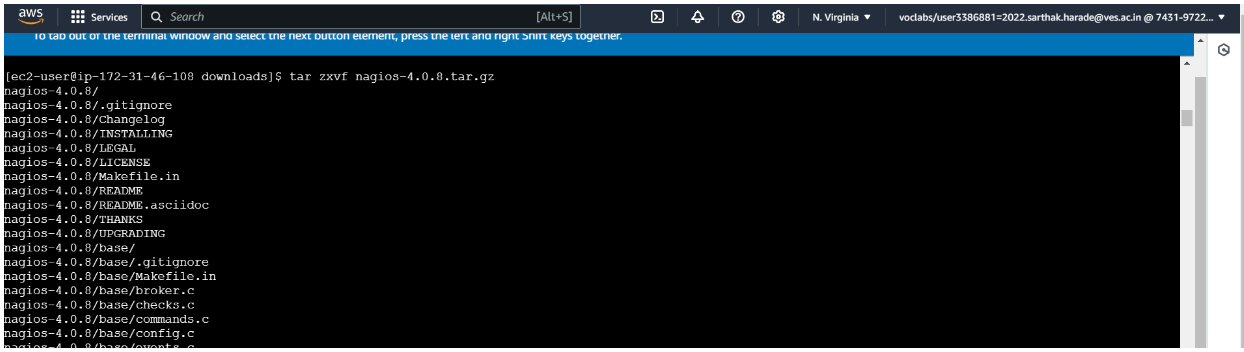
Wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz

wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz



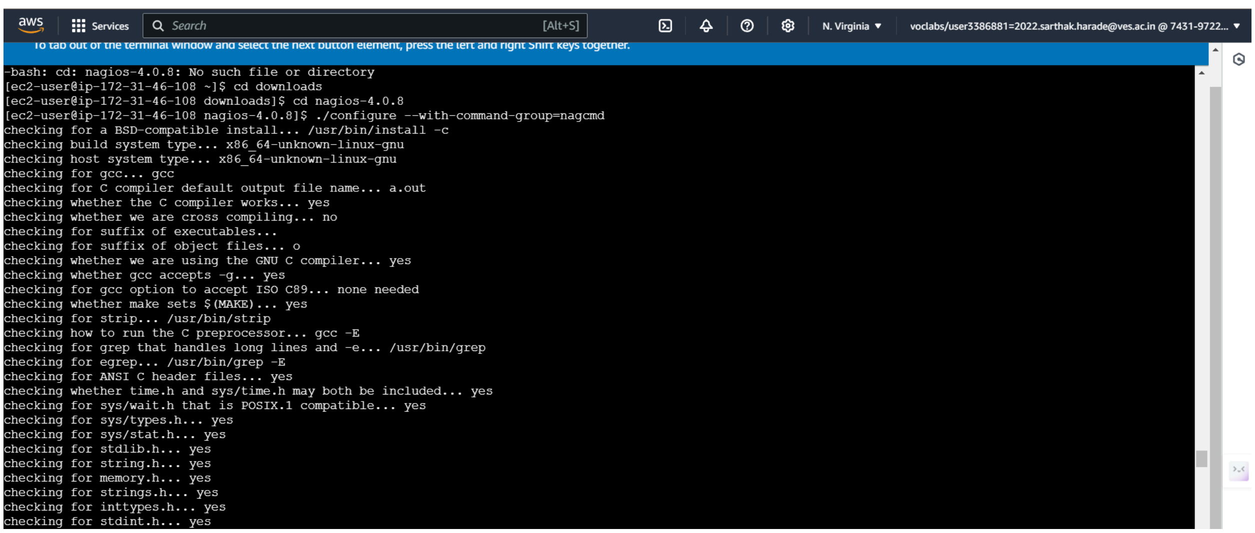
10. Use tar to unzip and change to that directory.

tar zxvf nagios-4.0.8.tar.gz



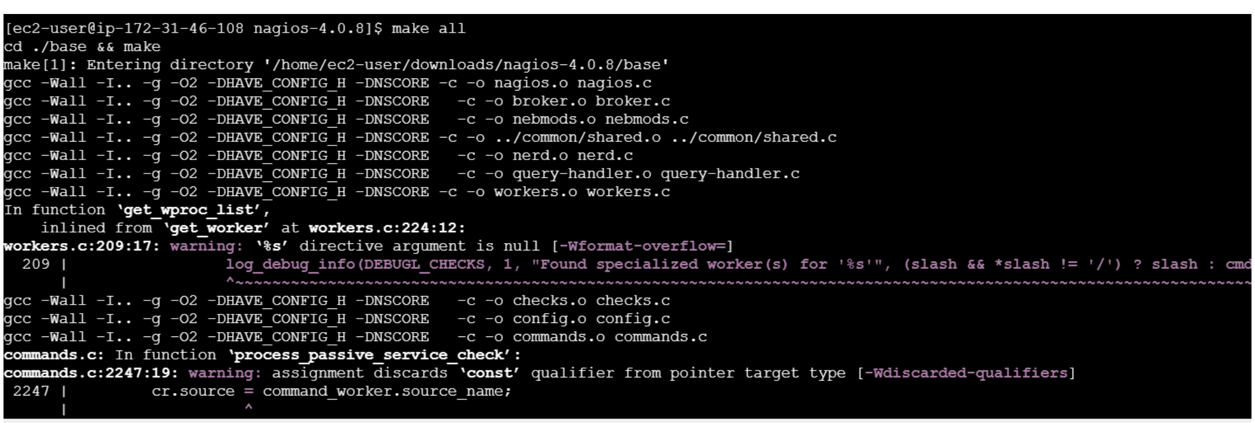
11. Run the configuration script with the same group name you previously created.

./configure --with-command-group=nagcmd



12. Compile the source code.

make all



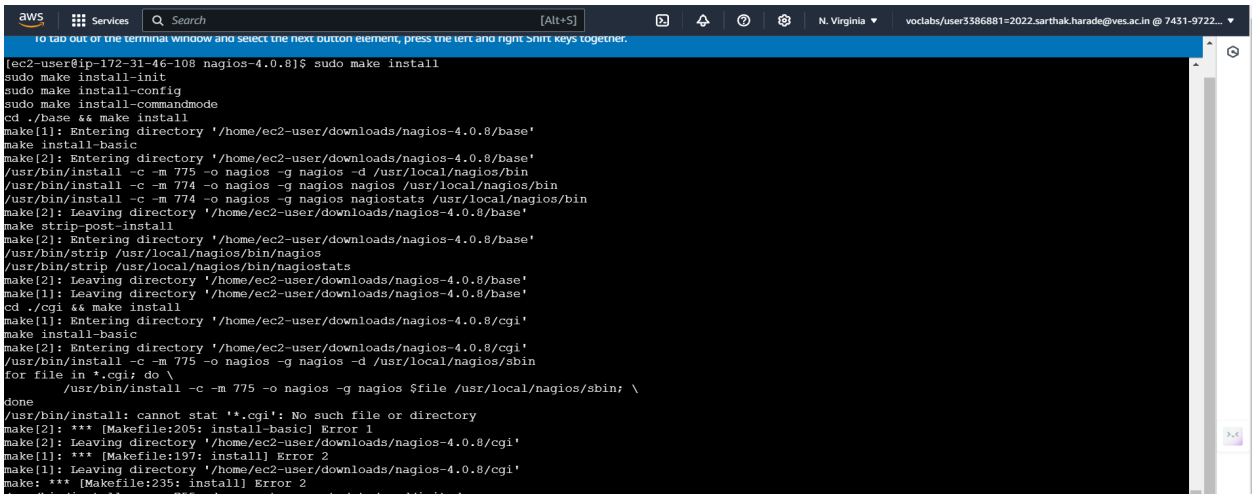
13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

sudo make install

sudo make install-init

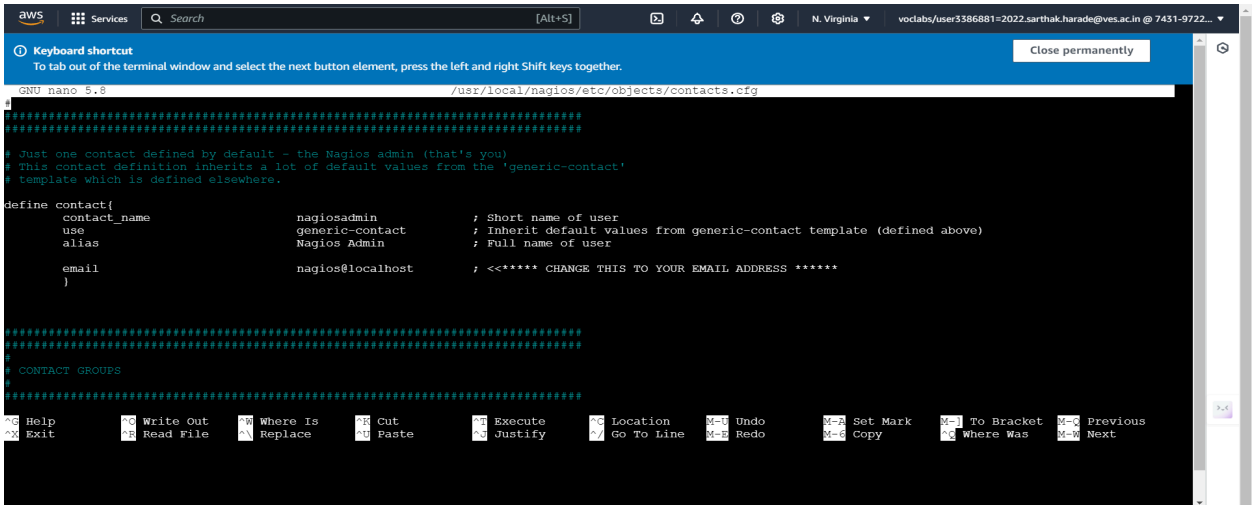
sudo make install-config

sudo make install-commandmode



14. Edit the config file and change the email address.

sudo nano /usr/local/nagios/etc/objects/contacts.cfg



15. Configure the web interface.

sudo make install-webconf

```
[ec2-user@ip-172-31-46-108 nagios-4.0.8]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf

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

[ec2-user@ip-172-31-46-108 nagios-4.0.8]$
```

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-46-108 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-46-108 nagios-4.0.8]$
```

17. Restart Apache

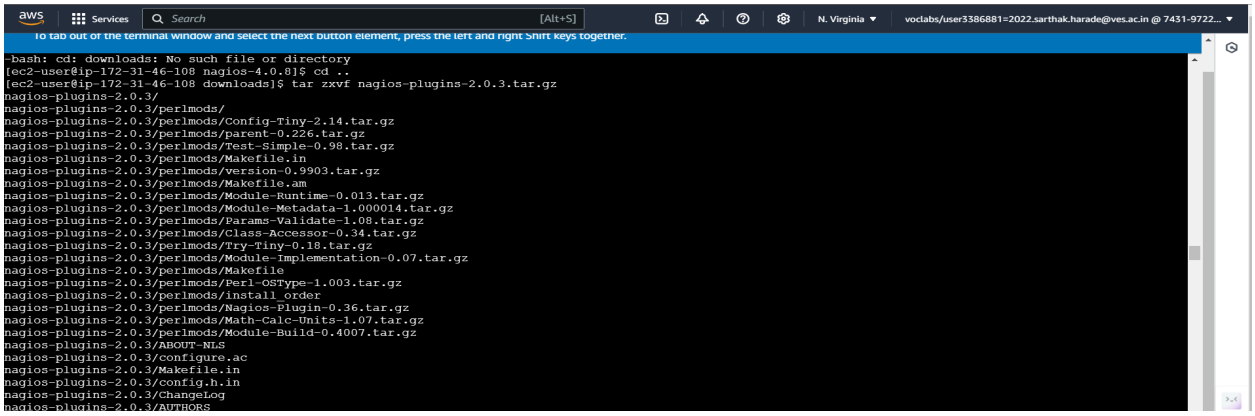
sudo service httpd restart

```
[ec2-user@ip-172-31-46-108 nagios-4.0.8]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-46-108 nagios-4.0.8]$
```

18. Go back to the downloads folder and unzip the plugins zip file.

cd ~/downloads

tar zxvf nagios-plugins-2.0.3.tar.gz



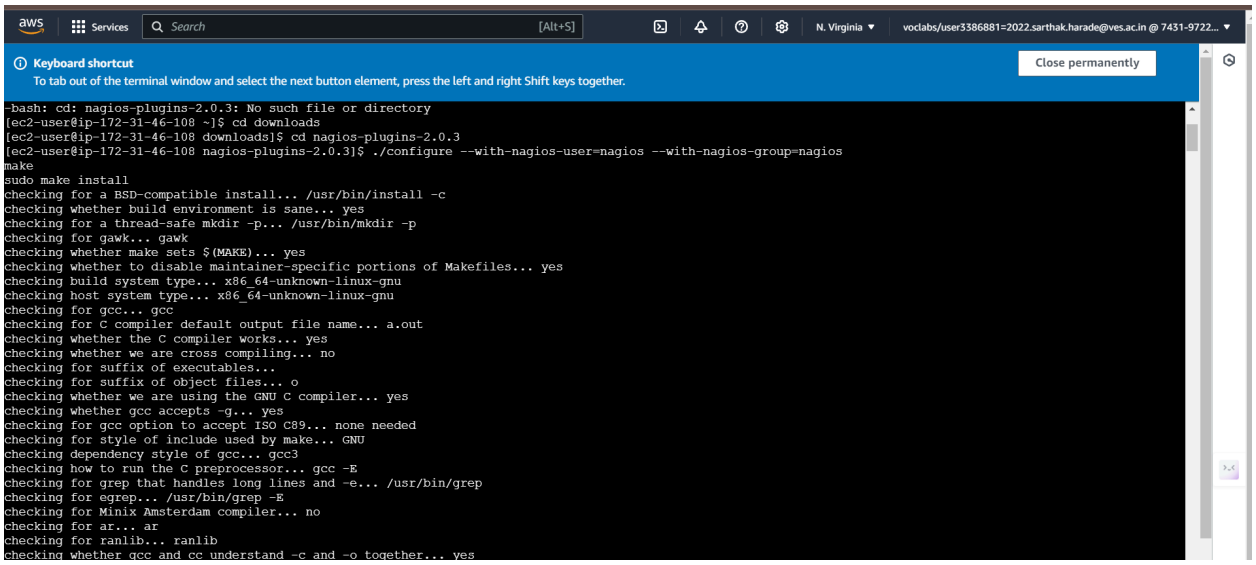
19. Compile and install plugins

cd nagios-plugins-2.0.3

./configure --with-nagios-user=nagios --with-nagios-group=nagios

make

sudo make install



20. Start Nagios

Add Nagios to the list of system services

sudo chkconfig --add nagios

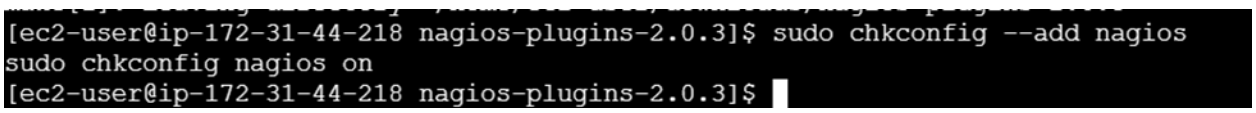
sudo chkconfig nagios on

Verify the sample configuration files

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

If there are no errors, you can go ahead and start Nagios.

sudo service nagios start



21. Check the status of Nagios

sudo systemctl status nagios

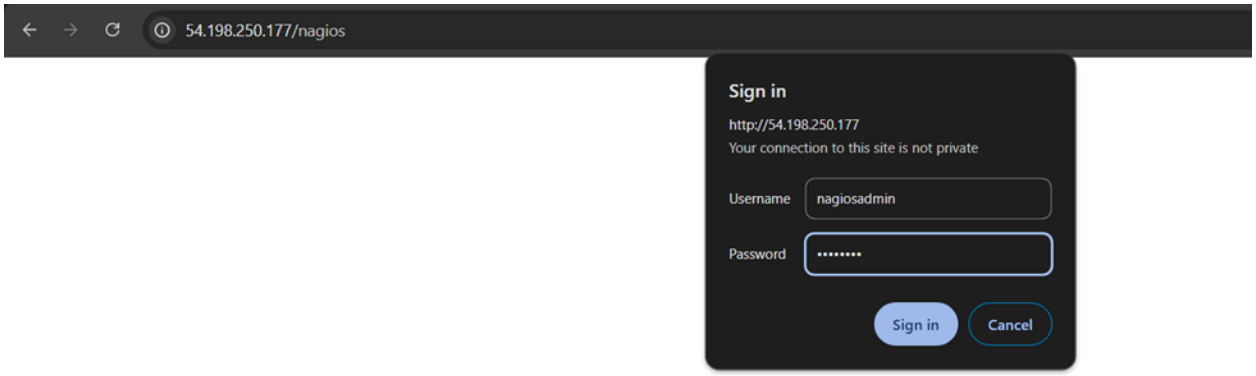
22. Go back to EC2 Console and copy the Public IP address of this instance

```
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-44-218 nagios-plugins-2.0.3]$ sudo service nagios start
Reloading systemd:                                         [  OK  ]
Starting nagios (via systemctl):                           [  OK  ]
[ec2-user@ip-172-31-44-218 nagios-plugins-2.0.3]$
```

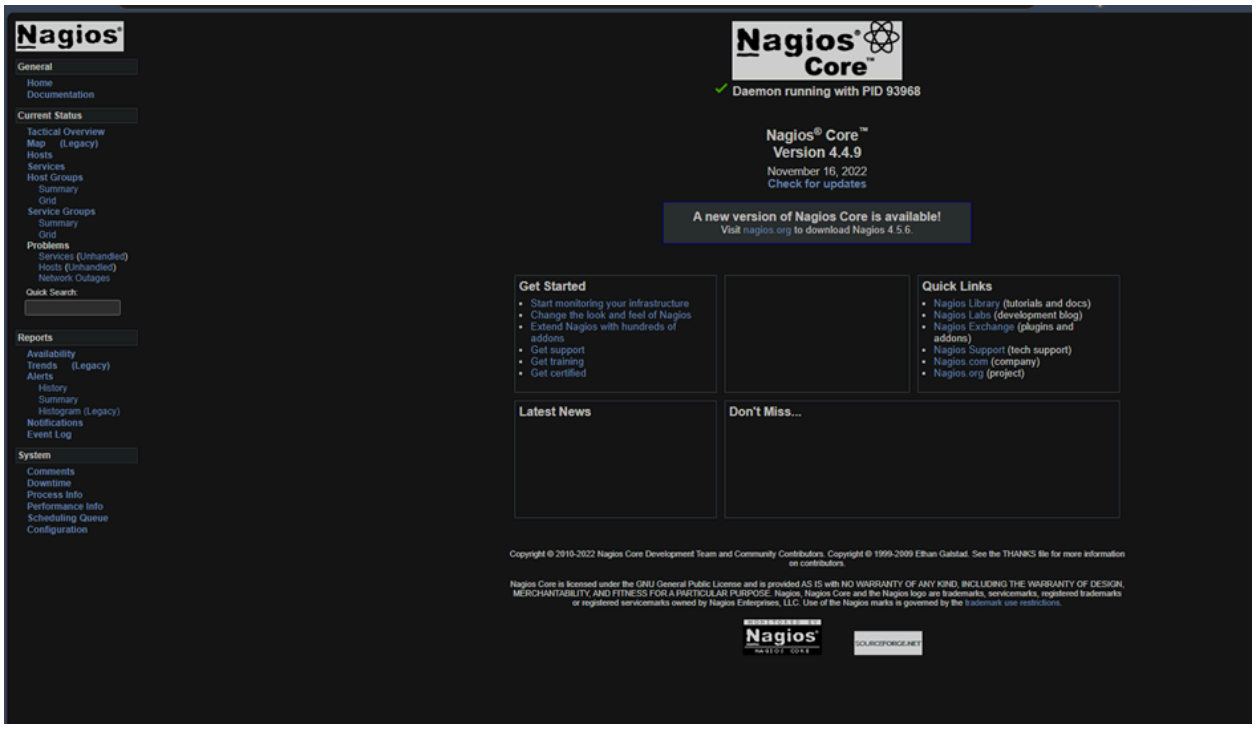
```
Starting nagios (via systemctl):                           [  OK  ]
[ec2-user@ip-172-31-44-218 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
     Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
     Active: active (running) since Sat 2024-10-12 09:59:46 UTC; 51s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 66468 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
      Tasks: 6 (limit: 1112)
     Memory: 2.1M
        CPU: 51ms
     CGroup: /system.slice/nagios.service
             ├─66490 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─66492 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66493 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66494 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66495 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─66496 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 12 09:59:46 ip-172-31-44-218.ec2.internal nagios[66490]: wproc: Registry request: name=Core Worker 66493;pid=66493
Oct 12 09:59:46 ip-172-31-44-218.ec2.internal nagios[66490]: wproc: Registry request: name=Core Worker 66492;pid=66492
Oct 12 09:59:46 ip-172-31-44-218.ec2.internal nagios[66490]: Warning: Could not open object cache file '/usr/local/nagios/var/objects.cache' for writing!
Oct 12 09:59:46 ip-172-31-44-218.ec2.internal nagios[66490]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmp7129Os' for writing status data
Oct 12 09:59:46 ip-172-31-44-218.ec2.internal nagios[66490]: Successfully launched command file worker with pid 66496
Oct 12 09:59:55 ip-172-31-44-218.ec2.internal nagios[66490]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmp8hbM0K' for writing status data
```

23. Open up your browser and look for http://<your_public_ip_address>/nagios



24. After entering the correct credentials, you will see this page.



- **Conclusion:**

Thus, we learned about Nagios and successfully set it up as a host on our Amazon Linux machine.