Snehal Patil D15A 39

Advanced DevOps Lab

Experiment No: 9

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● Helps you to plan for infrastructure upgrades before outdated systems create failures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

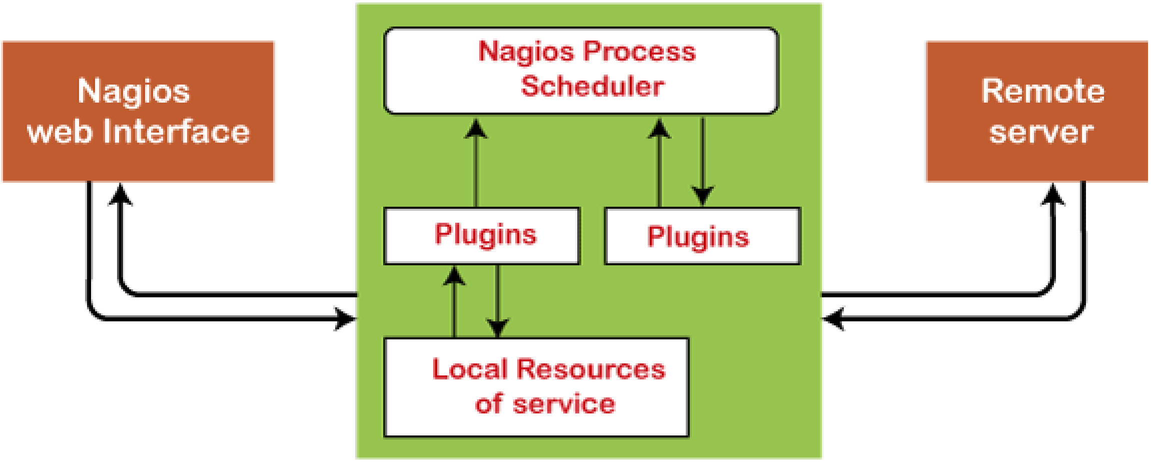
● You can troubleshoot the performance issues of the server.

- 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
- Support for implementing redundant monitoring hosts

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

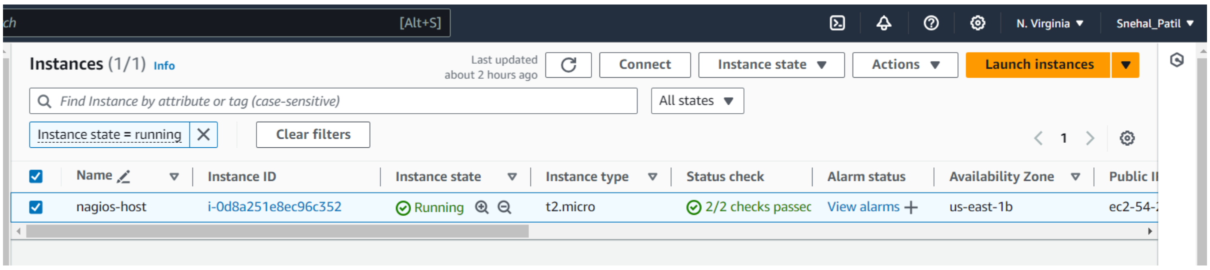
4. The process scheduler updates

Installation of Nagios

Prerequisites: AWS Free Tier

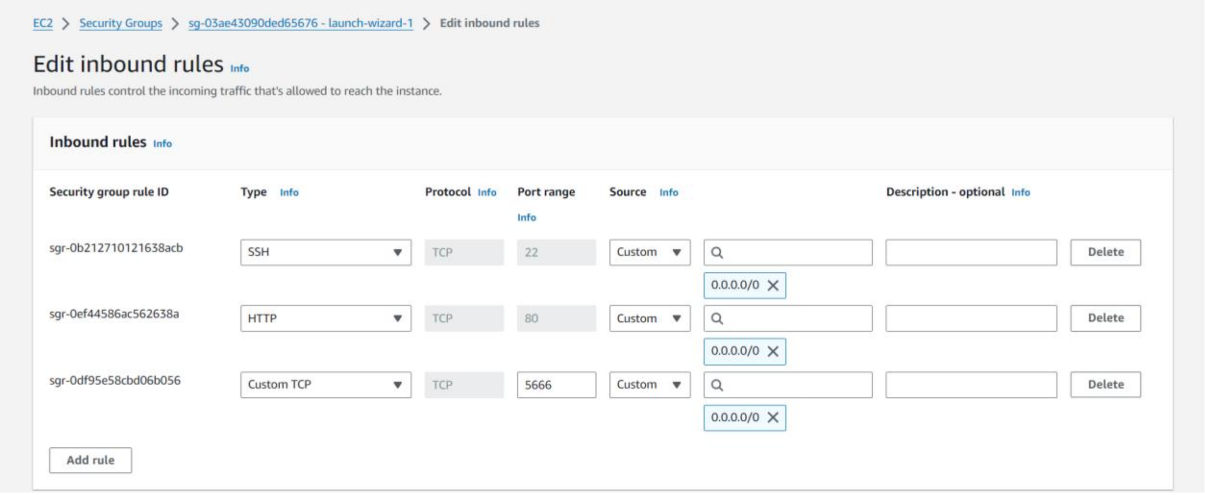
Steps:

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



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public II
☑	nagios-host	i-0d8a251e8ec96c352	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-54-:

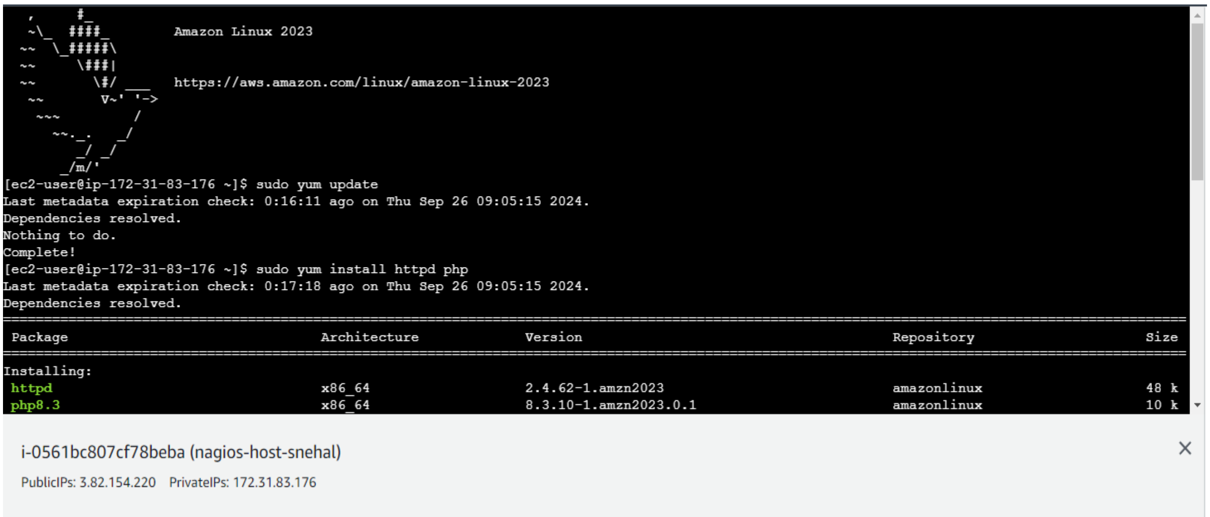
2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.


EC2 > Security Groups > sg-03ae43090ded65676 - launch-wizard-1 > Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Delete
Add rule

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0b212710121638acb	SSH	TCP	22	Custom	0.0.0.0/0 ✕	
sgr-0ef44586ac562638a	HTTP	TCP	80	Custom	0.0.0.0/0 ✕	
sgr-0df95e58cbd06b056	Custom TCP	TCP	5666	Custom	0.0.0.0/0 ✕	

You have to edit the inbound rules of the specified Security Group for this.

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



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-83-176 ~]$ sudo yum update
Last metadata expiration check: 0:16:11 ago on Thu Sep 26 09:05:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-83-176 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:17:18 ago on Thu Sep 26 09:05:15 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
```

i-0561bc807cf78beba (nagios-host-snehal)

PublicIPs: 3.82.154.220 PrivateIPs: 172.31.83.176

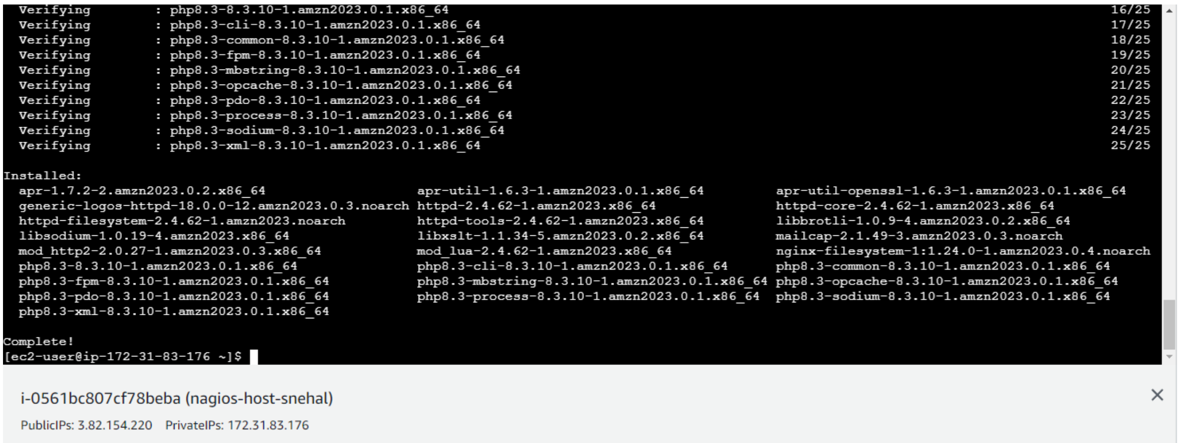
4. Update the package indices and install the following packages using yum

sudo yum update

sudo yum install httpd php

sudo yum install gcc glibc glibc-common

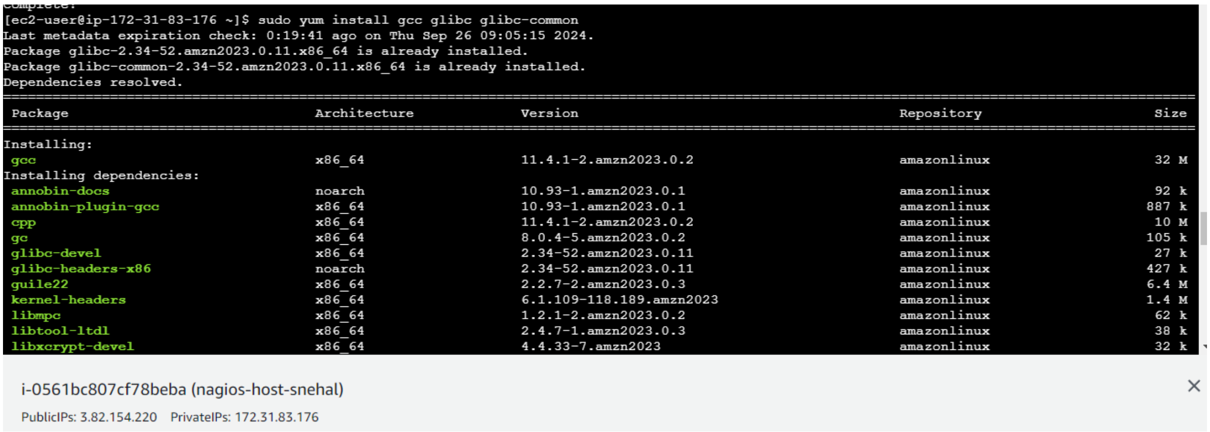
sudo yum install gd gd-devel



```
  Verifying        : php8.3-8.3.10-1.amzn2023.0.1.x86_64                                                                16/25
  Verifying        : php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64                                                            17/25
  Verifying        : php8.3-common-8.3.10-1.amzn2023.0.1.x86_64                                                         18/25
  Verifying        : php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64                                                            19/25
  Verifying        : php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64                                                       20/25
  Verifying        : php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64                                                        21/25
  Verifying        : php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64                                                            22/25
  Verifying        : php8.3-process-8.3.10-1.amzn2023.0.1.x86_64                                                        23/25
  Verifying        : php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64                                                         24/25
  Verifying        : php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64                                                            25/25

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64                   apr-util-1.6.3-1.amzn2023.0.1.x86_64          apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch httpd-2.4.62-1.amzn2023.x86_64                httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch         httpd-tools-2.4.62-1.amzn2023.x86_64          libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  libsodium-1.0.19-4.amzn2023.x86_64                libxslt-1.1.34-5.amzn2023.0.2.x86_64          mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64            mod_lua-2.4.62-1.amzn2023.x86_64              nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
  php8.3-8.3.10-1.amzn2023.0.1.x86_64               php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64       php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64           php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64 php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64           php8.3-process-8.3.10-1.amzn2023.0.1.x86_64  php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-83-176 ~]$
```

i-0561bc807cf78beba (nagios-host-snehal)

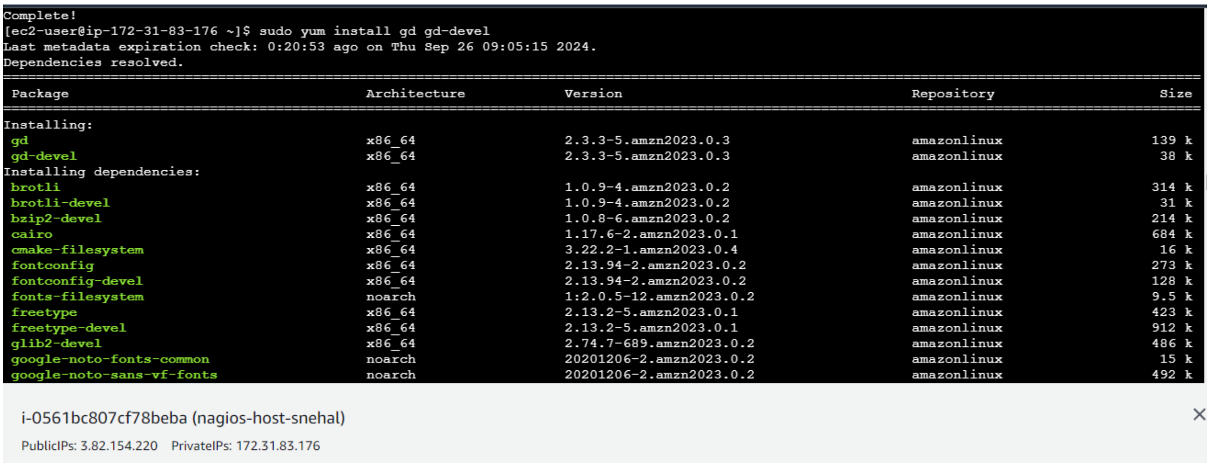
PublicIPs: 3.82.154.220 PrivateIPs: 172.31.83.176



```
Complete!
[ec2-user@ip-172-31-83-176 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:19:41 ago on Thu Sep 26 09:05:15 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
```

i-0561bc807cf78beba (nagios-host-snehal)

PublicIPs: 3.82.154.220 PrivateIPs: 172.31.83.176

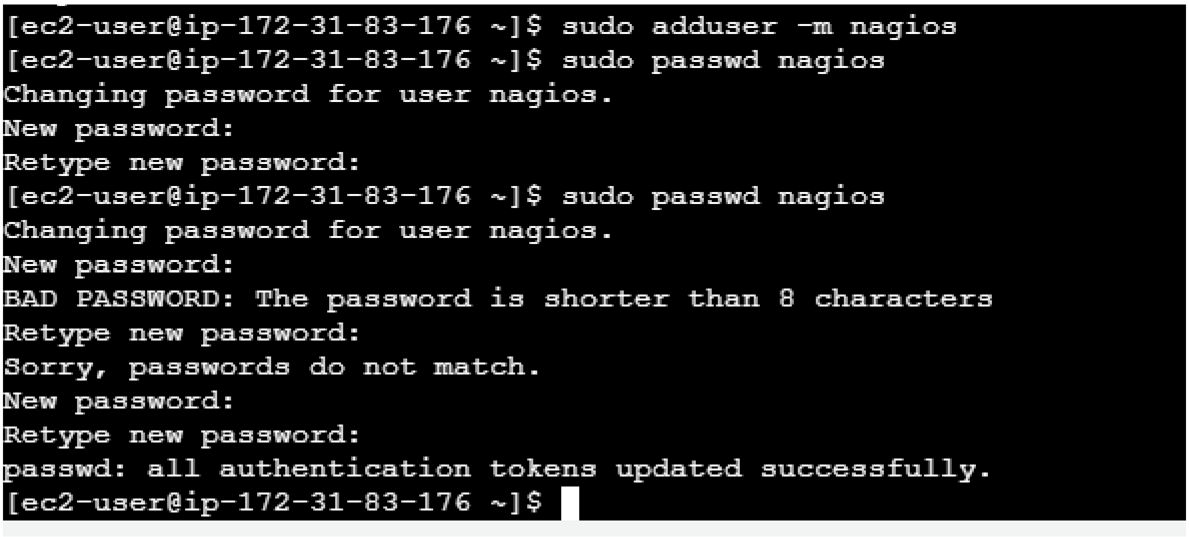


```
Complete!
[ec2-user@ip-172-31-83-176 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:20:53 ago on Thu Sep 26 09:05:15 2024.
Dependencies resolved.
================================================================================================================================
 Package                            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
```

i-0561bc807cf78beba (nagios-host-snehal)

PublicIPs: 3.82.154.220 PrivateIPs: 172.31.83.176

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



```
[ec2-user@ip-172-31-83-176 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-83-176 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
[ec2-user@ip-172-31-83-176 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-83-176 ~]$
```

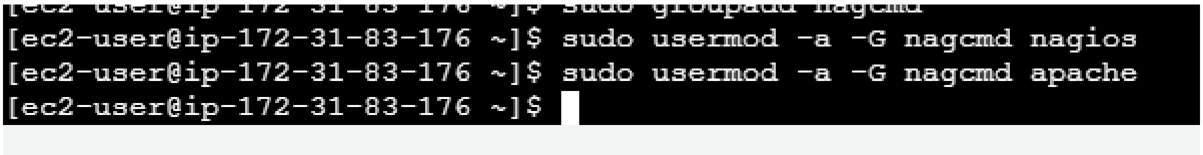
6. Create a new user group

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

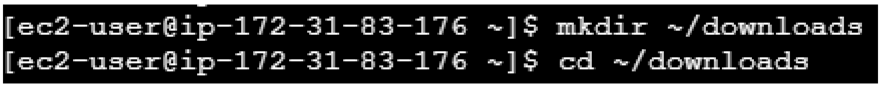


```
[ec2-user@ip-172-31-83-176 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-83-176 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-83-176 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-83-176 ~]$
```

8. Create a new directory for Nagios downloads

mkdir ~/downloads

cd ~/downloads



```
[ec2-user@ip-172-31-83-176 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-83-176 ~]$ cd ~/downloads
```

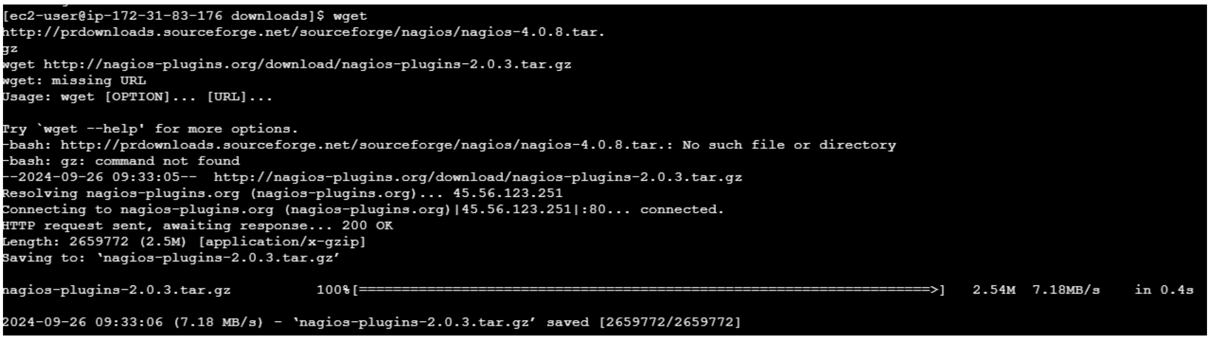
9. Use wget to download the source zip files.

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



```
[ec2-user@ip-172-31-83-176 downloads]$ 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
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.
-bash: http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.: No such file or directory
-bash: gz: command not found
--2024-09-26 09:33:05--  http://nagios-plugins.org/download/nagios-plugins-2.0.3.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|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.0.3.tar.gz’

nagios-plugins-2.0.3.tar.gz         100%[===================================================================>]   2.54M  7.18MB/s    in 0.4s

2024-09-26 09:33:06 (7.18 MB/s) - ‘nagios-plugins-2.0.3.tar.gz’ saved [2659772/2659772]
```

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

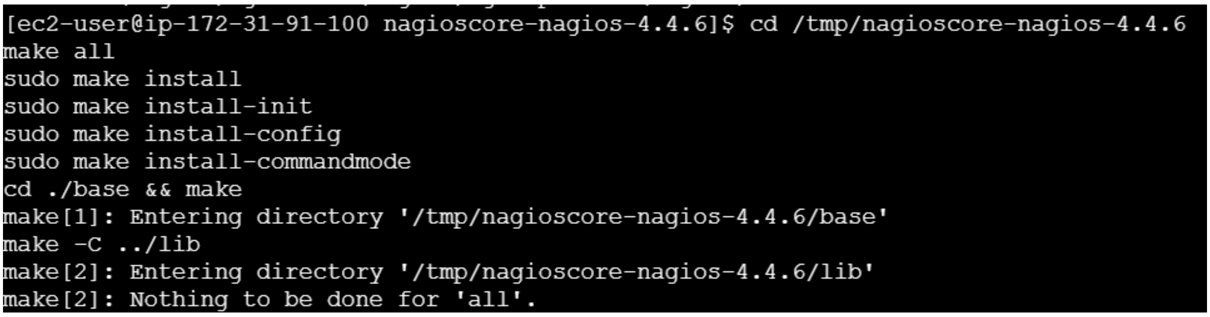
command directory.

sudo make install

sudo make install-init

sudo make install-config

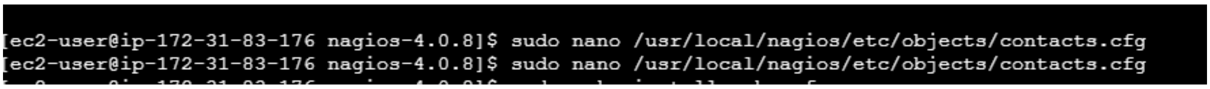
sudo make install-commandmode



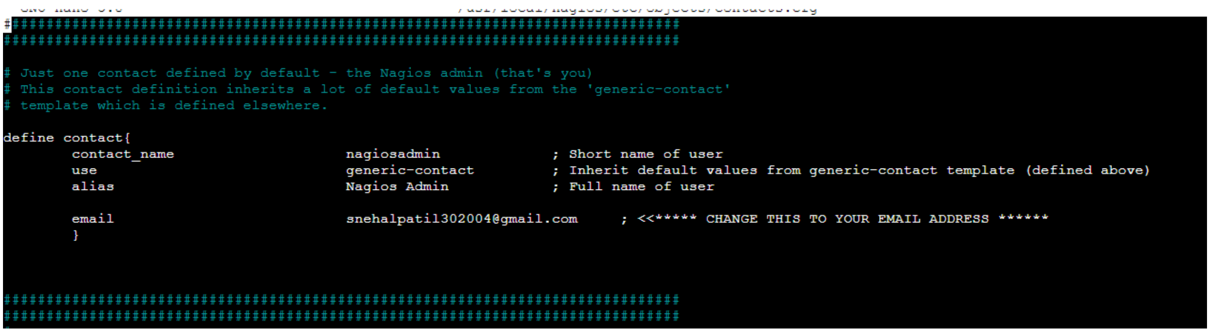
```
[ec2-user@ip-172-31-91-100 nagioscore-nagios-4.4.6]$ cd /tmp/nagioscore-nagios-4.4.6
make all
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make
make[1]: Entering directory '/tmp/nagioscore-nagios-4.4.6/base'
make -C ../lib
make[2]: Entering directory '/tmp/nagioscore-nagios-4.4.6/lib'
make[2]: Nothing to be done for 'all'.
```

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

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



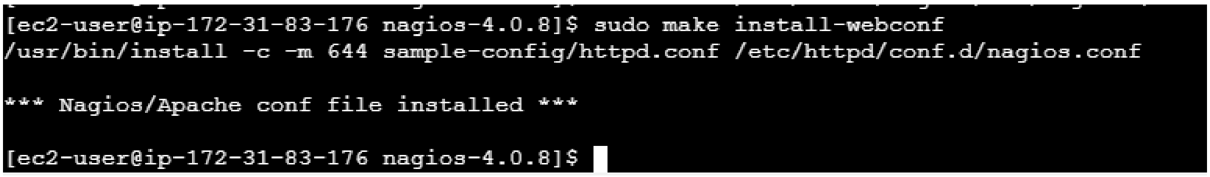
```
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
```



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

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.

define contact{
        contact_name                    nagiosadmin             ; Short name of user
        use                             generic-contact         ; Inherit default values from generic-contact template (defined above)
        alias                           Nagios Admin            ; Full name of user

        email                           snehalpatil302004@gmail.com     ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
        }


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

15. Configure the web interface.

sudo make install-webconf



```
[ec2-user@ip-172-31-83-176 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-172-31-83-176 nagios-4.0.8]$
```

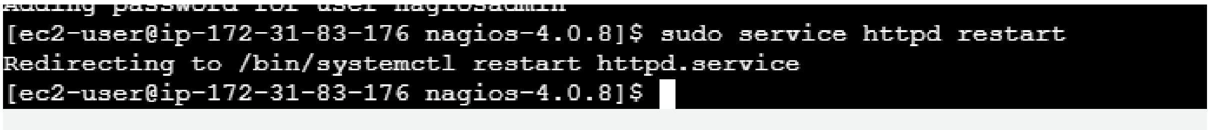
16. Create a nagiosadmin account for nagios login along with password. You'll have to

specify the password twice.

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

17. Restart Apache

sudo service httpd restart

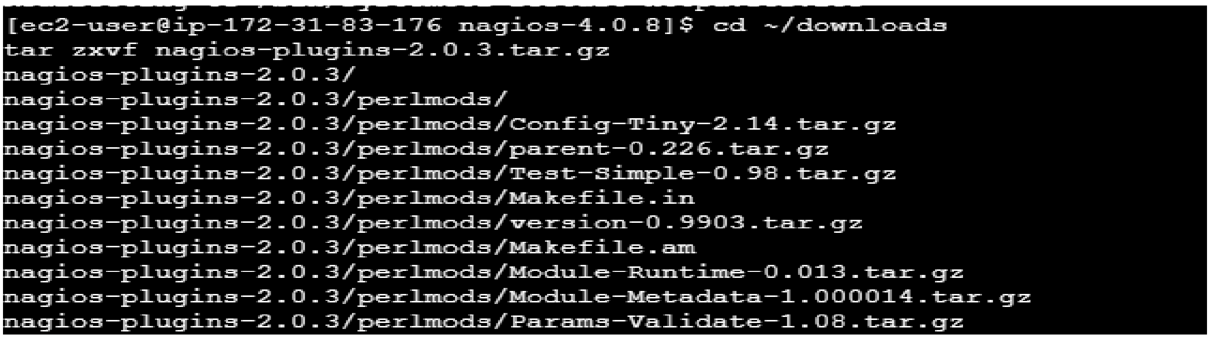


```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$
```

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

cd ~/downloads

tar zxvf nagios-plugins-2.0.3.tar.gz



```
[ec2-user@ip-172-31-83-176 nagios-4.0.8]$ cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
nagios-plugins-2.0.3/
nagios-plugins-2.0.3/perlmods/
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.in
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.am
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.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

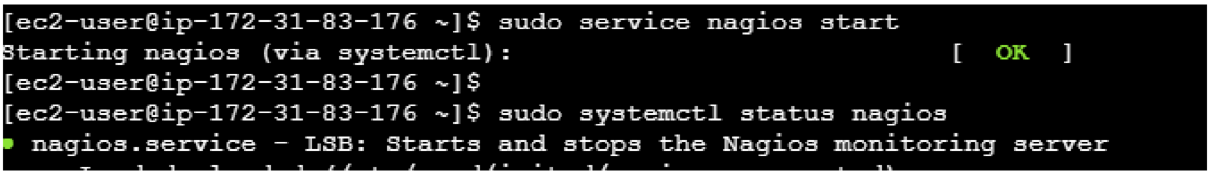
sudo chkconfig nagios on

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.

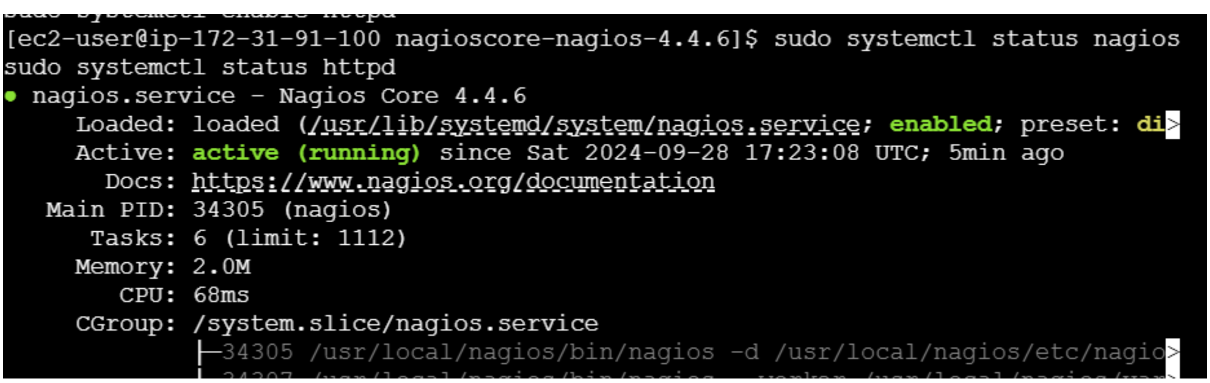
sudo service nagios start



```
[ec2-user@ip-172-31-83-176 ~]$ sudo service nagios start
Starting nagios (via systemctl):                           [  OK  ]
[ec2-user@ip-172-31-83-176 ~]$
[ec2-user@ip-172-31-83-176 ~]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
```

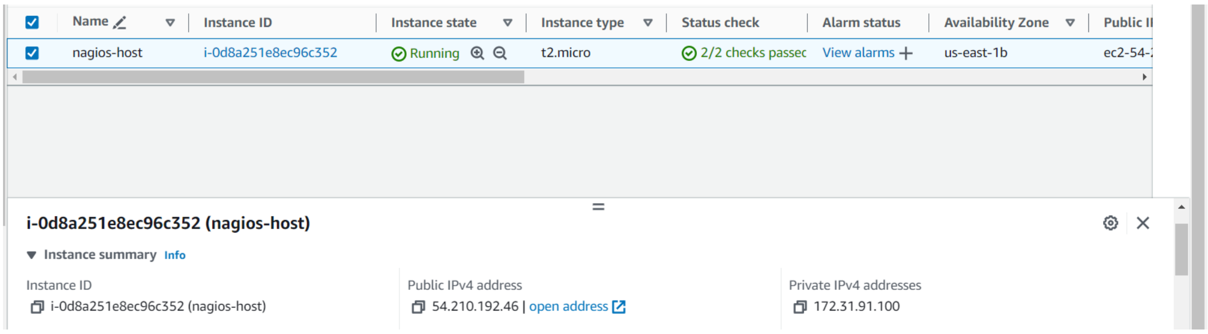
21. Check the status of Nagios

sudo systemctl status nagios



```
sudo systemctl enable httpd
[ec2-user@ip-172-31-91-100 nagioscore-nagios-4.4.6]$ sudo systemctl status nagios
sudo systemctl status httpd
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: di>
     Active: active (running) since Sat 2024-09-28 17:23:08 UTC; 5min ago
       Docs: https://www.nagios.org/documentation
   Main PID: 34305 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 2.0M
        CPU: 68ms
     CGroup: /system.slice/nagios.service
             ├─34305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagio>
```

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



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public II
☑	nagios-host	i-0d8a251e8ec96c352	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-54-:

i-0d8a251e8ec96c352 (nagios-host)

Instance summary Info

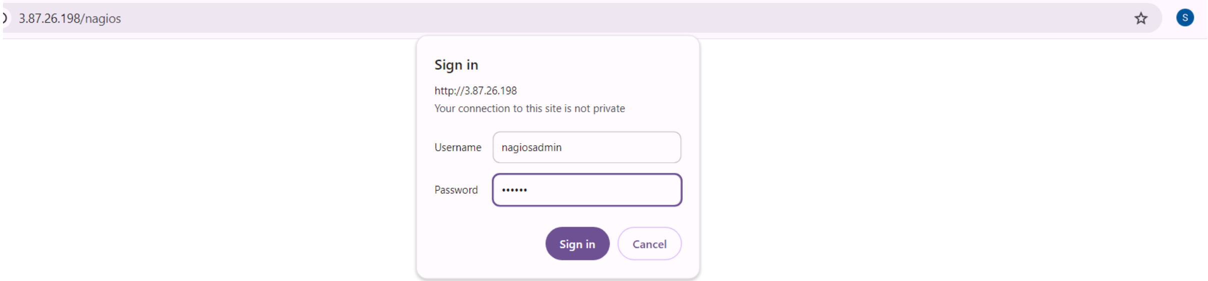
Instance ID
i-0d8a251e8ec96c352 (nagios-host)

Public IPv4 address
54.210.192.46 | open address

Private IPv4 addresses
172.31.91.100

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

Enter username as nagiosadmin and password which you set in Step 16.



3.87.26.198/nagios

Sign in

http://3.87.26.198

Your connection to this site is not private

Username: nagiosadmin

Password: ••••••

Extra:

Faced Problem

Permission Forbidden Error

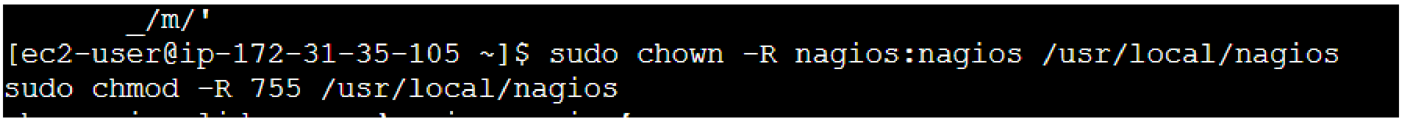
Problem: While accessing the Nagios web interface, I encountered "403 Forbidden" errors.

Solution: This issue typically arises due to improper file permissions or SELinux settings. To resolve it:

Adjust File Permissions: Ensure that the Nagios directory and its files have the correct permissions. Execute the following commands:

sudo chown -R nagios:nagios /usr/local/nagios

sudo chmod -R 755 /usr/local/nagios

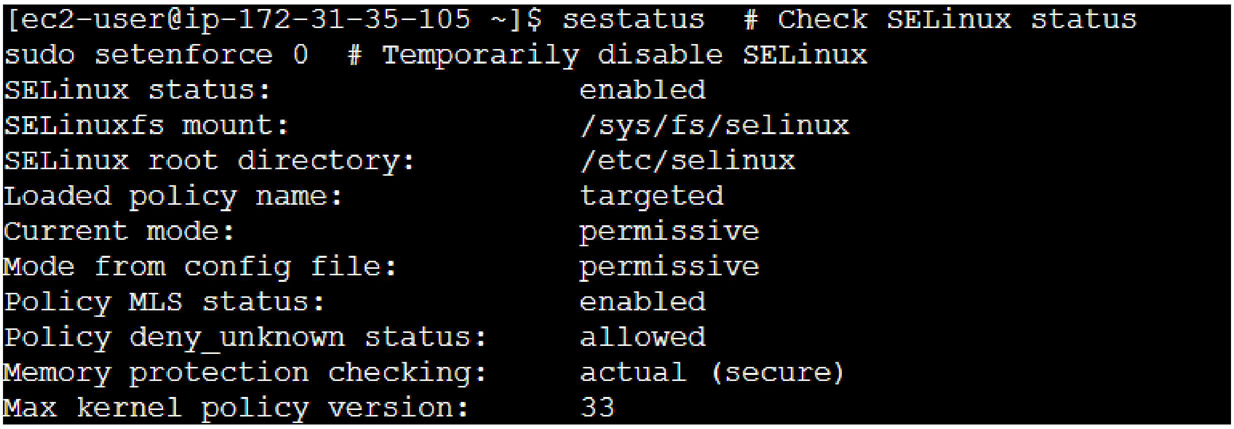


```
      _/m/'
[ec2-user@ip-172-31-35-105 ~]$ sudo chown -R nagios:nagios /usr/local/nagios
sudo chmod -R 755 /usr/local/nagios
```

Configure SELinux (if applicable): If SELinux is enabled on your instance, it might block access. To check the status and modify it:

sestatus # Check SELinux status

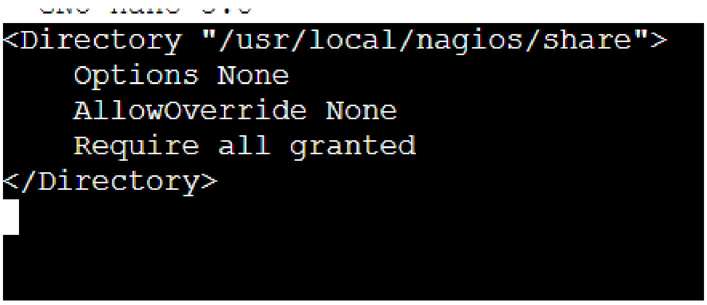
sudo setenforce 0 # Temporarily disable SELinux



```
[ec2-user@ip-172-31-35-105 ~]$ sestatus  # Check SELinux status
sudo setenforce 0  # Temporarily disable SELinux
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

Apache Configuration: Ensure that the Apache configuration allows access to the Nagios directory. You may need to add the following lines to your /etc/httpd/conf.d/nagios.conf:

```
<Directory "/usr/local/nagios/share">
    Options None
    AllowOverride None
    Require all granted
</Directory>
```



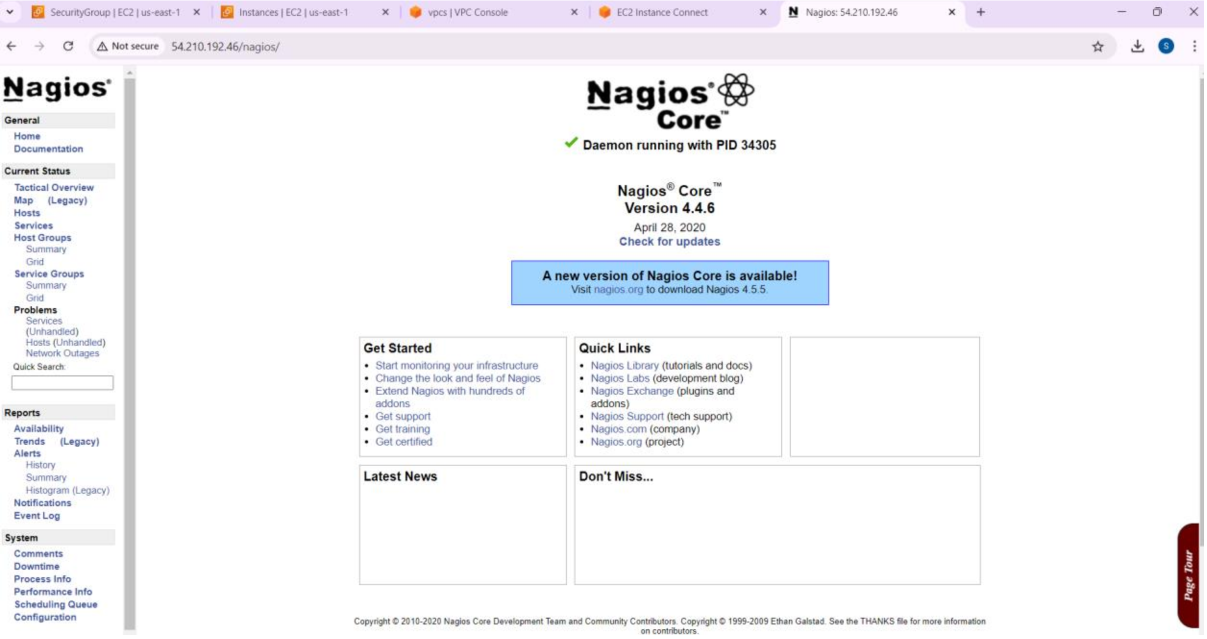
```
<Directory "/usr/local/nagios/share">
    Options None
    AllowOverride None
    Require all granted
</Directory>
```

After making changes, restart Apache:

```
sudo service httpd restart
```

24. After entering the correct credentials, I am able to

see this page.



✓ Daemon running with PID 34305

Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

Don't Miss...

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion:

We successfully installed Nagios on an Amazon Linux EC2 instance in the AWS Free Tier. By configuring the security group, setting up users and groups, and installing the necessary packages, we prepared the environment for Nagios. After downloading and compiling the software, we configured the web interface and authenticated the admin account. Accessing the Nagios dashboard confirmed a successful setup, enabling effective monitoring of systems and services. This installation lays the groundwork for a robust monitoring solution tailored to your needs.