

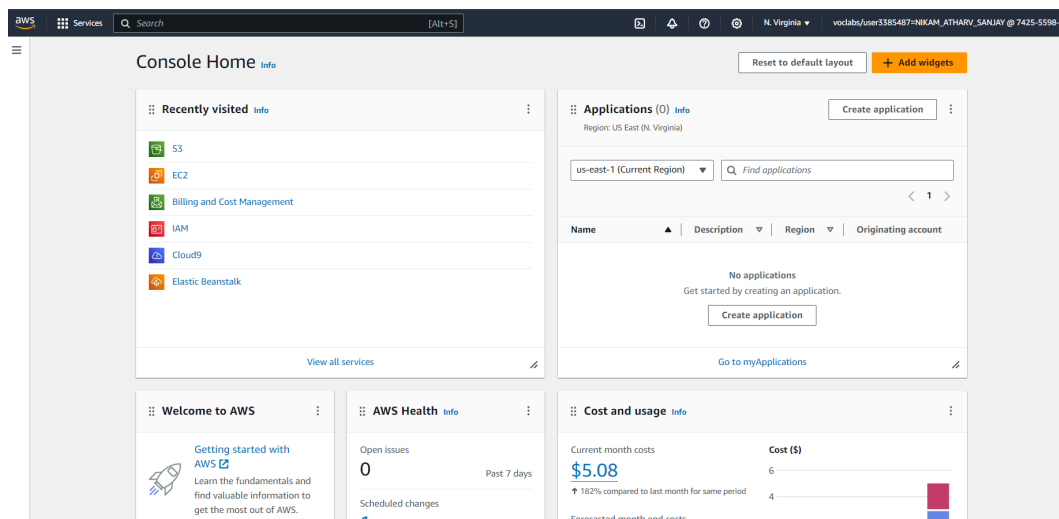
Experiment 9

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

STEPS

Step 1: Accessing AWS EC2

Log in to your AWS account, search for EC2 in the services section, and open the interface. Click on "Create Instance" and select **Amazon Linux** as the OS image for the instance.



Step 2: Setting Up Key Pair

If you don't already have a key pair or **.pem** file, generate a new key pair. If one exists, select it. Ensure that the **.pem** file is saved securely on your system for future use.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

NagiosAtharv

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair

Step 3: Configuring Security Group

Navigate to the **Security Groups** section from the left panel. Locate the security group linked to your instance, click on the instance ID, and edit the **Inbound Rules**. Add the following rules: HTTP, HTTPS, Custom TCP (port 5666), All ICMP (IPv4 and IPv6), and All traffic. Save the rules.

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Security Groups (1) Info					
<input type="text" value="Find resources by attribute or tag"/>					
<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-03a60d807bfc11237	default	vpc-0f7970ea32a533bcc	default VPC security group

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Nagios

Name cannot be edited after creation.

Description [Info](#)

Allow ssh acces to developers

VPC [Info](#)

vpc-0f7970ea32a533bcc

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
-	SSH	TCP	22	Anywhere...	0.0.0.0/0	Delete
-	HTTP	TCP	80	Anywhere...	0.0.0.0/0	Delete
-	HTTPS	TCP	443	Anywhere...	0.0.0.0/0	Delete
-	All ICMP - IPv4	ICMP	All	Anywhere...	0.0.0.0/0	Delete

Add rule

Cancel

Preview changes

Save rules

✔ Security group (sg-0b59e140edaa1f431 | Nagios) was created successfully

Details

[EC2](#) > [Security Groups](#) > sg-0b59e140edaa1f431 - Nagios

sg-0b59e140edaa1f431 - Nagios

Actions ▼

Details

Security group name Nagios	Security group ID sg-0b59e140edaa1f431	Description Allow ssh acces to developers	VPC ID vpc-0f7970ea32a533bcc
Owner 742555988891	Inbound rules count 0 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#)[Outbound rules](#)[Tags](#)

Step 4: Connecting to EC2 Instance

Back in the EC2 dashboard, click on the instance ID and choose **Connect**. Under **SSH Client**, copy the provided command. Open your terminal where the **.pem** file is located, paste the SSH command, and execute it to connect.

The screenshot shows the 'Launch an instance' page in the AWS Management Console. The page title is 'Launch an instance' with an 'Info' link. Below the title, a brief description states: 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.' The main section is 'Name and tags' with a text input field containing 'Nagios-host' and an 'Add additional tags' button. Below this is the 'Application and OS Images (Amazon Machine Image)' section, which includes a search bar and a 'Quick Start' tab. Under the 'Quick Start' tab, there are several AMI options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Li. Each option has a corresponding logo and a 'Browse more AMIs' link.

The screenshot shows the 'Network settings' page in the AWS Management Console. The page title is 'Network settings' with an 'Info' link and an 'Edit' button. The page is divided into several sections: 'Network' with a text input field containing 'vpc-0f7970ea32a533bcc'; 'Subnet' with a text input field containing 'No preference (Default subnet in any availability zone)'; 'Auto-assign public IP' with a text input field containing 'Enable'; 'Firewall (security groups)' with a text input field containing 'Nagios sg-0b59e140edaa1f431' and a 'VPC: vpc-0f7970ea32a533bcc' dropdown menu. There are also buttons for 'Create security group', 'Select existing security group', and 'Compare security group rules'. A note at the bottom states: 'Security groups that you add or remove here will be added to or removed from all your network interfaces.'

[EC2](#) > [Instances](#) > [i-084ba8cc8adfce2a](#) > [Connect to instance](#)

Connect to instance Info

Connect to your instance i-084ba8cc8adfce2a (Nagios-host) using any of these options


EC2 Instance Connect



Session Manager


SSH client


EC2 serial console


Instance ID

 **i-084ba8cc8adfce2a** (Nagios-host)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is NagiosAtharv.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "NagiosAtharv.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-44-202-137-247.compute-1.amazonaws.com`

 **Command copied**

 `ssh -i "NagiosAtharv.pem" ec2-user@ec2-44-202-137-247.compute-1.amazonaws.com`

 **Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

```
PS D:\Advancedevops key> ssh -i "NagiosAtharv.pem" ec2-user@ec2-44-201-253-108.compute-1.amazonaws.com
```

A newer release of "Amazon Linux" is available.
Version 2023.5.20241001:
Run "/usr/bin/dnf check-release-update" for full release and version update info

```
, #_
~\####_      Amazon Linux 2023
~~ \#####\
~~  \###|
~~   \|/     https://aws.amazon.com/linux/amazon-linux-2023
~~    V~' '->
~~~~
~~~~_._./
~~~~_/_/_/
~~~~_/m/' '->
```

```
[ec2-user@ip-172-31-86-175 ~]$
```

Step 5: Updating YUM Packages

Run `sudo yum update` to update the system's YUM package manager. This ensures the latest updates and security patches are installed

```
=====
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-86-175 ~]$
```

Step 6: Installing Apache and PHP

Install the necessary web server components, Apache and PHP, by running:
`sudo yum install httpd php.`

```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install httpd php
Last metadata expiration check: 1 day, 13:28:04 ago on Fri Oct 4 03:32:13 2024.
Dependencies resolved.
===== Package
Architecture Version Repository Size
-----
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:
apr-util-openssl x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 17 k
mod_http2 x86_64 2.0.27-1.amzn2023.0.3 amazonlinux 166 k
mod_lua x86_64 2.4.62-1.amzn2023.0.1 amazonlinux 61 k
php8.3-fpm x86_64 8.3.10-1.amzn2023.0.1 amazonlinux 1.9 M
php8.3-mbstring x86_64 8.3.10-1.amzn2023.0.1 amazonlinux 528 k
php8.3-opcache x86_64 8.3.10-1.amzn2023.0.1 amazonlinux 379 k
php8.3-pdo x86_64 8.3.10-1.amzn2023.0.1 amazonlinux 89 k
```

```

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

Complete!

```

Step 7: Installing C/C++ Compiler

Install the GCC compiler along with C libraries using:

```
sudo yum install gcc glibc glibc-common.
```

```

[ec2-user@ip-172-31-86-175 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 1 day, 13:29:48 ago on Fri Oct 4 03:32:13 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
Installed size: 168 M
Is this ok [y/N]:

```

```

Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch
  cpp-11.4.1-2.amzn2023.0.2.x86_64
  gcc-11.4.1-2.amzn2023.0.2.x86_64
  glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
  kernel-headers-6.1.109-118.189.amzn2023.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64
  annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
  gc-8.0.4-5.amzn2023.0.2.x86_64
  glibc-devel-2.34-52.amzn2023.0.11.x86_64
  guile22-2.2.7-2.amzn2023.0.3.x86_64
  libmpc-1.2.1-2.amzn2023.0.2.x86_64
  libxcrypt-devel-4.4.33-7.amzn2023.x86_64

```

Step 8: Installing GD Library

For rendering images in Nagios, install the GD library with:

```
sudo yum install gd gd-devel.
```

```
[ec2-user@ip-172-31-86-175 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 1 day, 13:31:19 ago on Fri Oct 4 03:32:13 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-filessystem                   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
graphite2                           x86_64                 1.3.14-7.amzn2023.0.2   amazonlinux             97 k
graphite2-devel                    x86_64                 1.3.14-7.amzn2023.0.2   amazonlinux             21 k
harfbuzz                            x86_64                 7.0.0-2.amzn2023.0.1   amazonlinux             868 k
=====
```

```
Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-2.3.3-5.amzn2023.0.3.x86_64
glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libICE-1.0.10-6.amzn2023.0.2.x86_64
libX11-1.7.2-3.amzn2023.0.4.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.9-6.amzn2023.0.2.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-2:1.6.37-10.amzn2023.0.6.x86_64
libsdl2-devel-2.0.18-1.amzn2023.0.2.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
fonts-filessystem-1:2.0.5-12.amzn2023.0.2.noarch
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
gd-devel-2.3.3-5.amzn2023.0.3.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-common-1.7.2-3.amzn2023.0.4.noarch
libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
libXpm-3.5.15-2.amzn2023.0.3.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
libsepol-devel-3.4-3.amzn2023.0.3.x86_64
libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
```

Complete!

```
[ec2-user@ip-172-31-86-175 ~]$
```


Step 9: Creating Nagios User

Create a user called **nagios**, provide it with a home directory, and set a password:

```
sudo adduser -m nagios
sudo passwd nagios.
```

```
Complete!
[ec2-user@ip-172-31-86-175 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-86-175 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-86-175 ~]$
```

Step 10: Creating Nagcmd Group

Create a group called **nagcmd** to manage Nagios command executions:

```
sudo groupadd nagcmd.
```

Step 11: Adding Users to Nagcmd Group

Add both **nagios** and **apache** users to the **nagcmd** group:

```
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache.
```

```
[ec2-user@ip-172-31-86-175 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-86-175 ~]$ sudo usermod -a -G nagcmd apache
```

Step 12: Creating Download Directory

Create a directory to store Nagios files:

```
mkdir ~/downloads
```

Navigate to the directory using:

```
cd ~/downloads.
```

```
[ec2-user@ip-172-31-86-175 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-86-175 ~]$ cd ~/downloads
[ec2-user@ip-172-31-86-175 downloads]$ _
```

Step 13: Downloading Nagios Core and Plugins

Download the latest versions of Nagios Core and Plugins. If no newer versions are available, use:

```
wget
```

```
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
```

```
wget
```

```
https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz.
```

```
[ec2-user@ip-172-31-86-175 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-05 17:11:46-- 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'

nagios-4.5.5.tar.gz      100%[=====>] 1.97M  5.10MB/s  in 0.4s
2024-10-05 17:11:46 (5.10 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
[ec2-user@ip-172-31-86-175 downloads]$ _
```

```
[ec2-user@ip-172-31-86-175 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-05 17:12:19-- 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'

nagios-plugins-2.4.11.tar.gz      100%[=====>] 2.62M  6.37MB/s  in 0.4s
2024-10-05 17:12:19 (6.37 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
[ec2-user@ip-172-31-86-175 downloads]$
```

Step 14: Extracting Nagios Core

Extract the Nagios Core files in the same directory using:

```
tar zxvf nagios-4.5.5.tar.gz.
```

```
[ec2-user@ip-172-31-86-175 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
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LLEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
```

```
nagios-4.5.5/xdata/xcddefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-86-175 downloads]$
```

Step 15: Configuring Nagios Core

```
[ec2-user@ip-172-31-86-175 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-86-175 downloads]$
```

If there's an error regarding SSL headers, install OpenSSL using:

`sudo yum install openssl-devel.`

```
[ec2-user@ip-172-31-86-175 downloads]$ sudo yum install openssl-devel
Last metadata expiration check: 1 day, 13:46:43 ago on Fri Oct  4 03:32:13 2024.
Dependencies resolved.
=====
Package                                Architecture          Version                Repos
=====
Installing:
openssl-devel                          x86_64                 1:3.0.8-1.amzn2023.0.14  amazo

Transaction Summary
=====
Install 1 Package

Total download size: 3.0 M
Installed size: 4.7 M
```

To ensure Nagios uses the `nagcmd` group for external commands, run:

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

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
```

Step 16: Compiling and Installing Nagios

Compile Nagios by running:

```
make all
```

Install with:

```
General Options:
-----
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: /run/nagios.lock
Check result directory: /usr/local/nagios/var/spool/checkresults
Init directory: /lib/systemd/system
Apache conf.d directory: /etc/httpd/conf.d
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll
```

```
sudo make install
```

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install
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 nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
```

sudo make install-init

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.c
fg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resour
ce.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/
nagios/etc/objects/templates.cfg
```

sudo make install-commandmode.

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

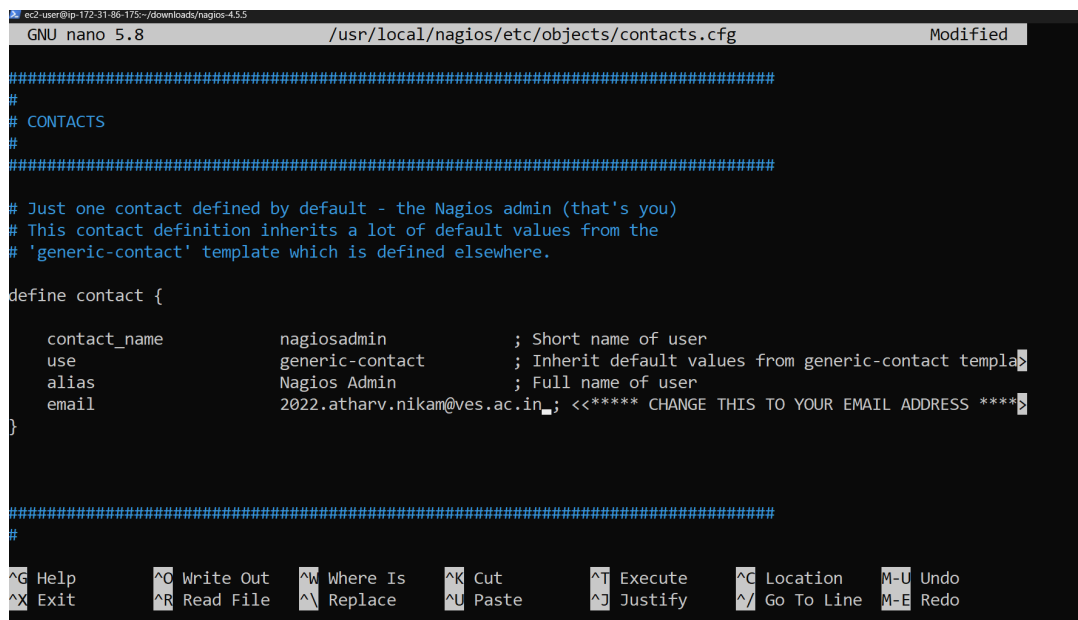
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 17: Configuring Email for Notifications

To configure notifications, open the `contacts.cfg` file and set your email address under `define contact{}` using:

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

Save your changes with CTRL+O and exit with CTRL+X.



```
ec2-user@ip-172-31-86-175:~/downloads/nagios-4.5.5
GNU nano 5.8 /usr/local/nagios/etc/objects/contacts.cfg Modified

#####
# CONTACTS
#
#####

# 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
    alias           Nagios Admin       ; Full name of user
    email           2022.atharv.nikam@ves.ac.in; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****>>

}

#####
#
#####

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_/ Go To Line M-E Redo
```

Step 18: Installing Nagios Web Interface

Install the web interface configuration by running:

```
sudo make install-webconf.
```

```
[ec2-user@ip-172-31-86-175 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 ***
```

Step 19: Creating Web Interface User

Create a user called `nagiosadmin` to access the Nagios web interface:

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

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadm
in
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 20: Restarting Apache Server

Restart the Apache server to apply the new configurations:

```
sudo service httpd restart.
```

```
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-86-175 nagios-4.5.5]$
```

Step 21: Extracting Nagios Plugins

Extract the Nagios plugins in the `downloads` directory using:

```
tar zxvf nagios-plugins-2.4.11.tar.gz.
```

```
[ec2-user@ip-172-31-86-175 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
```

```
nagios-plugins-2.4.11/po/POTFILES.in
nagios-plugins-2.4.11/po/fr.po
nagios-plugins-2.4.11/po/de.po
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-86-175 downloads]$
```



```
[ec2-user@ip-172-31-86-175 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath

nagios-plugins-2.4.11/build-aux/ltmain.sh
```

Step 22: Configuring Nagios Plugins

Navigate to the extracted folder and configure Nagios plugins with:

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

```
config.status: creating plugins-scripts/utils.sh
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged

config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

```
gins/utils.o -L. ../lib/libnagiosplug.a ../gl/libgnu.a -lresolv -lssl -lcrypto -lpthread
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/plugins-root'
Making all in po
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 23: Compiling and Installing Plugins

Compile and install the plugins using:

`make`

`sudo make install.`

```
fi
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 24: Registering Nagios as a Service

To make Nagios manageable as a system service, run:

`sudo chkconfig --add nagios`

`sudo chkconfig nagios on`

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo chkconfig nagios on
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.
```

Step 25: Verifying Configuration

Verify the Nagios configuration for any syntax errors:

`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...
  Read main config file okay...
  Read object config files okay...
```

```
Checking misc settings...
```

```
Total Errors: 0
```

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Step 25: Verifying Configuration

Verify the Nagios configuration for any syntax errors:

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

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo service nagios start  
Redirecting to /bin/systemctl start nagios.service  
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$
```

Check its status with:

```
sudo systemctl status nagios.
```

```
[ec2-user@ip-172-31-86-175 nagios-plugins-2.4.11]$ sudo systemctl status nagios  
● nagios.service - Nagios Core 4.5.5  
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)  
   Active: active (running) since Sat 2024-10-05 19:18:16 UTC; 14s ago  
     Docs: https://www.nagios.org/documentation  
   Process: 68151 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=ex  
   Process: 68152 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exit  
   Main PID: 68153 (nagios)  
     Tasks: 6 (limit: 1112)  
    Memory: 5.5M  
       CPU: 79ms  
   CGroup: /system.slice/nagios.service  
           └─68153 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg  
             └─68154 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh  
               └─68155 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh  
                 └─68156 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh  
                   └─68157 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh  
  
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: Socket '/usr/local/nagios/var/rw/nagios  
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: core query handler registered  
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: echo service query handler registered  
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: help for the query handler registered  
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Successfully registered manager as @  
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 68
```

```
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: Socket '/usr/local/nagios/var/rw/nagios'
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: core query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: echo service query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: qh: help for the query handler registered
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Successfully registered manager as @
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: wproc: Registry request: name=Core Worker 6
Oct 05 19:18:16 ip-172-31-86-175.ec2.internal nagios[68153]: Successfully launched command file worker w
lines 3-28/28 (END)
```

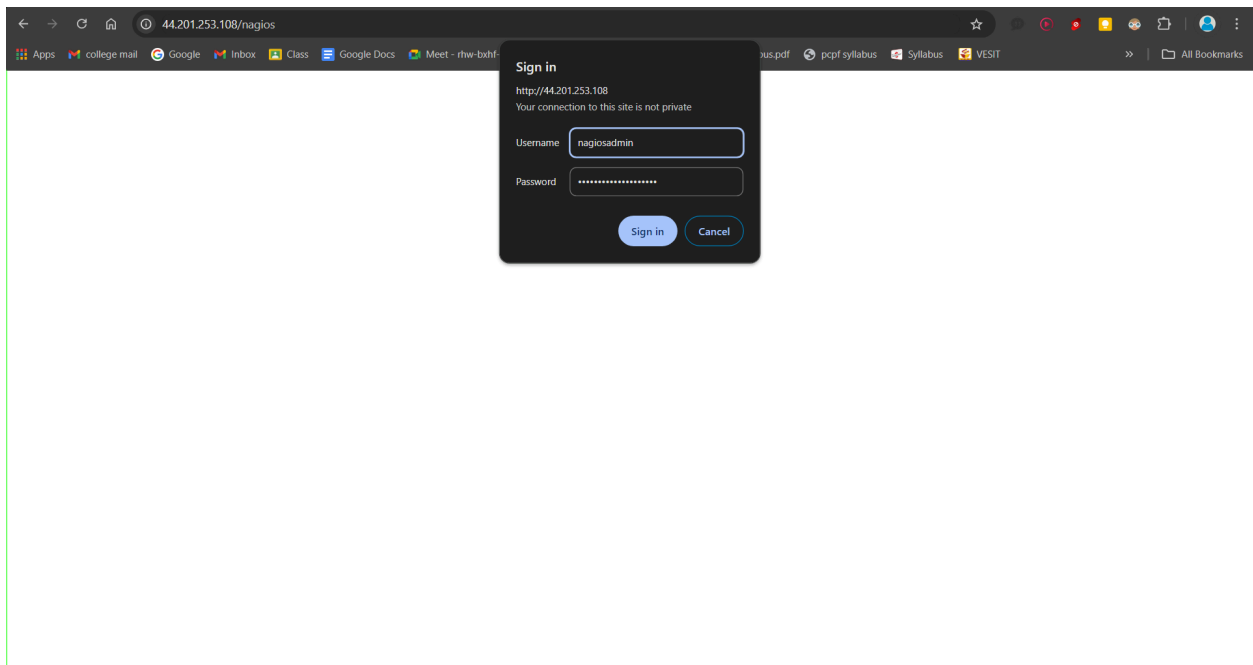
Step 27: Accessing Nagios Dashboard

Copy the public IP address of your EC2 instance and type it in your browser, appending **/nagios** to access the dashboard:

<http://<public-ip-address>/nagios>.

Username:nagiosadmin

password:(the password you have set)



The screenshot shows a web browser window with the Nagios Core 4.5.5 interface. The browser's address bar shows the URL `44.201.253.108/nagios/`. The interface includes a left sidebar with navigation links for General, Current Status, Reports, and System. The main content area displays the Nagios Core logo, version information (4.5.5), and a status message indicating the daemon is running with PID 68153. Below this, there are sections for 'Get Started', 'Quick Links', 'Latest News', and 'Don't Miss...'. The 'Get Started' section lists steps like monitoring infrastructure, changing the look, extending with addons, getting support, training, and certification. The 'Quick Links' section provides links to Nagios Library, Labs, Exchange, Support, company, and project pages. The 'Latest News' and 'Don't Miss...' sections are currently empty. At the bottom, there is a copyright notice for 2010-2024 Nagios Core Development Team and Community Contributors, and a disclaimer about the GNU General Public License and trademarks.

Conclusion:

In this experiment, I successfully explored the process of installing and configuring Nagios Core, Plugins, and NRPE on a Linux-based system. By carefully configuring the necessary security settings, installing dependencies, and resolving common issues like permission errors and missing directories, I ensured a smooth setup. This setup allows for efficient monitoring using Nagios, and I accessed the monitoring interface through the EC2 instance's public IP, demonstrating the effectiveness of continuous monitoring in a cloud environment.