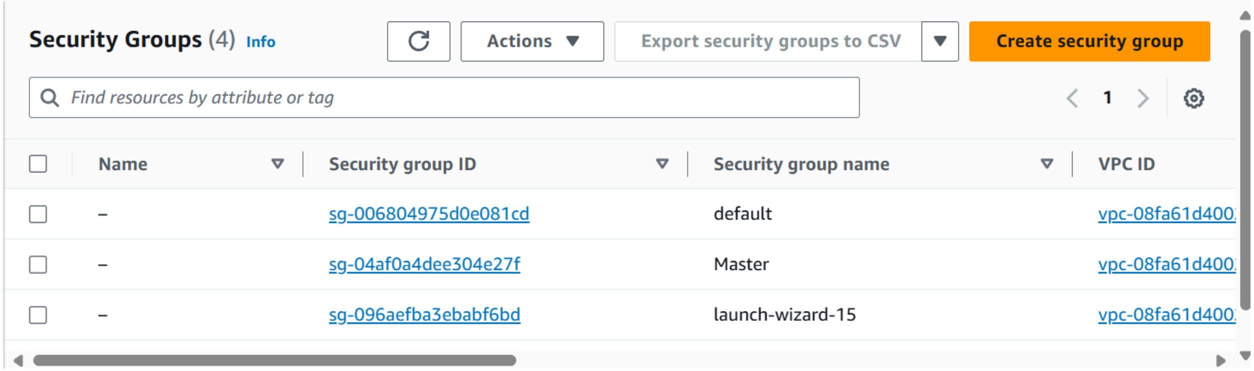
Name: Anish Kulkarni Roll No.: 29 Class: D15C AY: 2024-25

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:

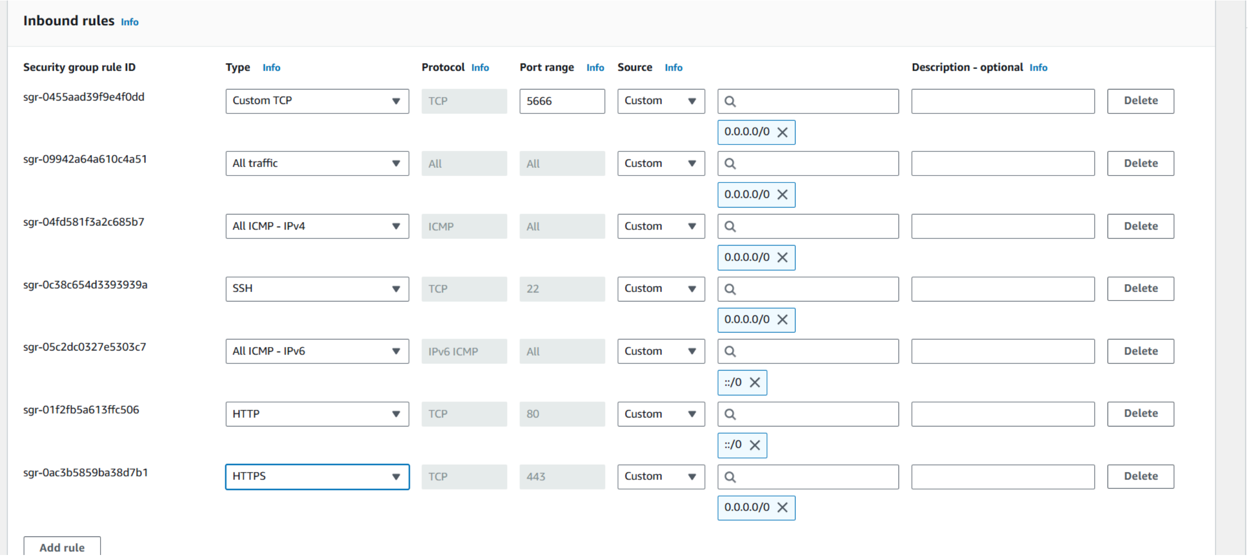
Step 1: Navigate to the EC2 section on your AWS console using the 'Services' section. Then, from the options in the left-side panel, click on 'Security groups'. Next, click on 'Create security group'.



Security Groups (4) Info
Actions ▼
Export security groups to CSV ▼
Create security group
Find resources by attribute or tag
1

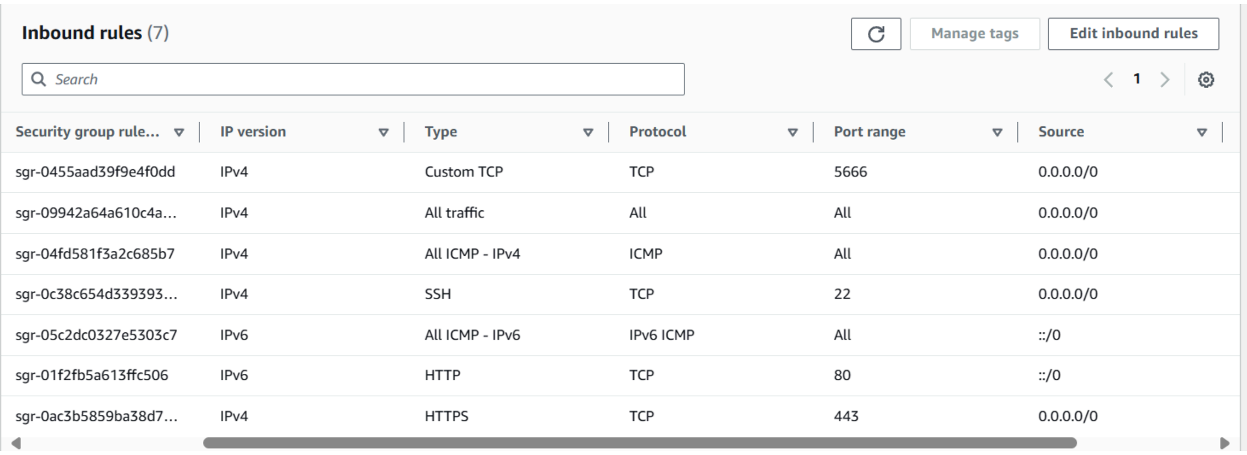
□	Name	Security group ID	Security group name	VPC ID
□	–	sg-006804975d0e081cd	default	vpc-08fa61d400
□	–	sg-04af0a4dee304e27f	Master	vpc-08fa61d400
□	–	sg-096aefba3ebabf6bd	launch-wizard-15	vpc-08fa61d400

Give your security group a name (here, the name is launch-wizard-15) and then in the 'Inbound rules' section, click on 'Edit'. Then, click on add rules, and add the rules for the following protocols:
HTTP, All ICMP - IPv6, HTTPS, All traffic, Custom TCP (Port 5666), All ICMP - IPv4



Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0455aad39f9e4f0dd	Custom TCP	TCP	5666	Custom	0.0.0.0/0		Delete
sgr-09942a64a610c4a51	All traffic	All	All	Custom	0.0.0.0/0		Delete
sgr-04fd581f3a2c685b7	All ICMP - IPv4	ICMP	All	Custom	0.0.0.0/0		Delete
sgr-0c38c654d3393939a	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-05c2dc0327e5303c7	All ICMP - IPv6	IPv6 ICMP	All	Custom	::/0		Delete
sgr-01f2fb5a613ffc506	HTTP	TCP	80	Custom	::/0		Delete
sgr-0ac3b5859ba38d7b1	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete

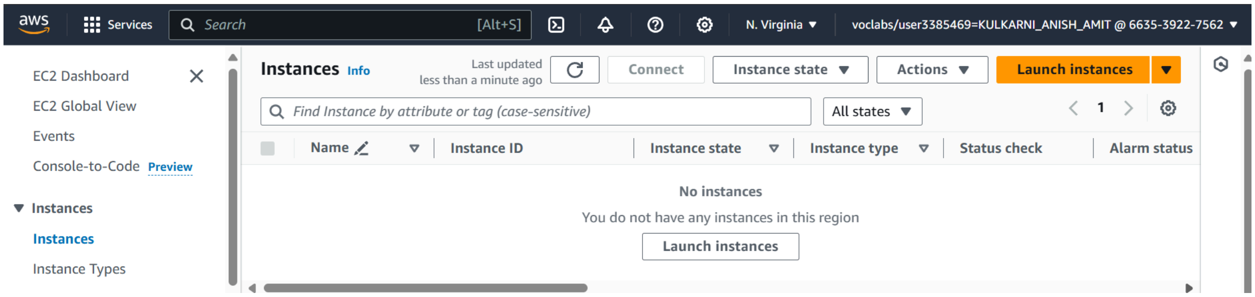
Your security group with the required inbound rules gets created as such:-



Inbound rules (7)

Security group rule...	IP version	Type	Protocol	Port range	Source
sgr-0455aad39f9e4f0dd	IPv4	Custom TCP	TCP	5666	0.0.0.0/0
sgr-09942a64a610c4a...	IPv4	All traffic	All	All	0.0.0.0/0
sgr-04fd581f3a2c685b7	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0
sgr-0c38c654d339393...	IPv4	SSH	TCP	22	0.0.0.0/0
sgr-05c2dc0327e5303c7	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0
sgr-01f2fb5a613ffc506	IPv6	HTTP	TCP	80	::/0
sgr-0ac3b5859ba38d7...	IPv4	HTTPS	TCP	443	0.0.0.0/0

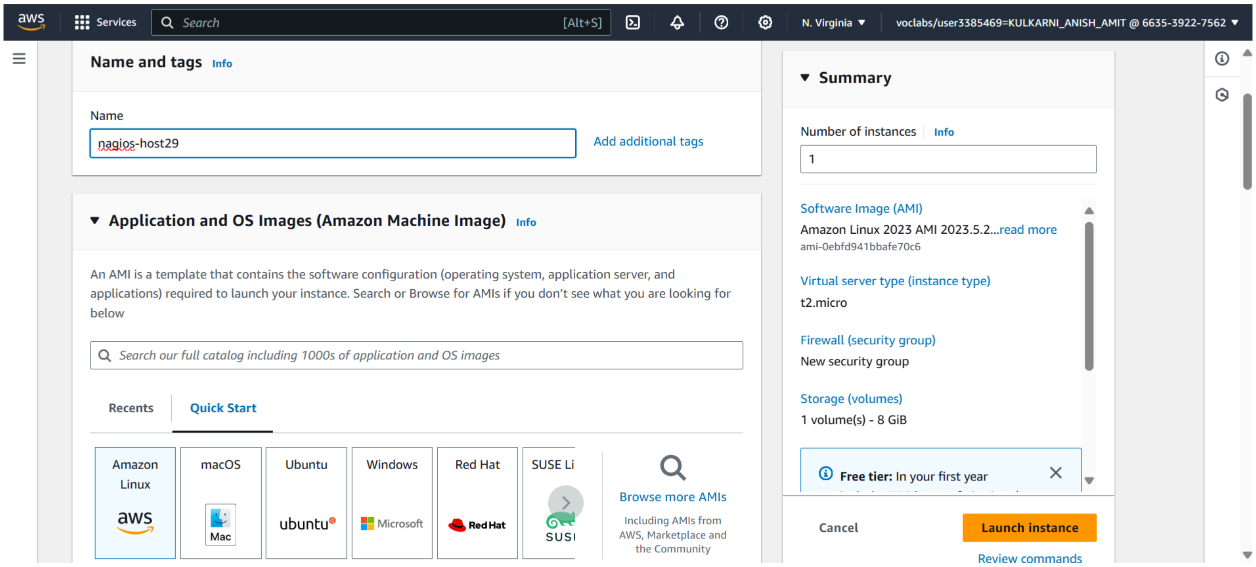
Step 2: Navigate to the EC2 section and click on 'Launch instances'.



No instances

You do not have any instances in this region

Give your instance a name, choose 'Amazon Linux' as the instance type, insert the key pair for which you have the .pem file available in the 'Key pair' section, choose the security group that you created in Step 1 in the 'Network settings' section, keep all other options as default and click on 'Launch instance'.



aws
Services
Search [Alt+S]
N. Virginia
voclabs/user3385469=KULKARNI_ANISH_AMIT @ 6635-3922-7562

Name and tags Info

Name: nagios-host29

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
SUSE Li
SUSE
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances Info: 1

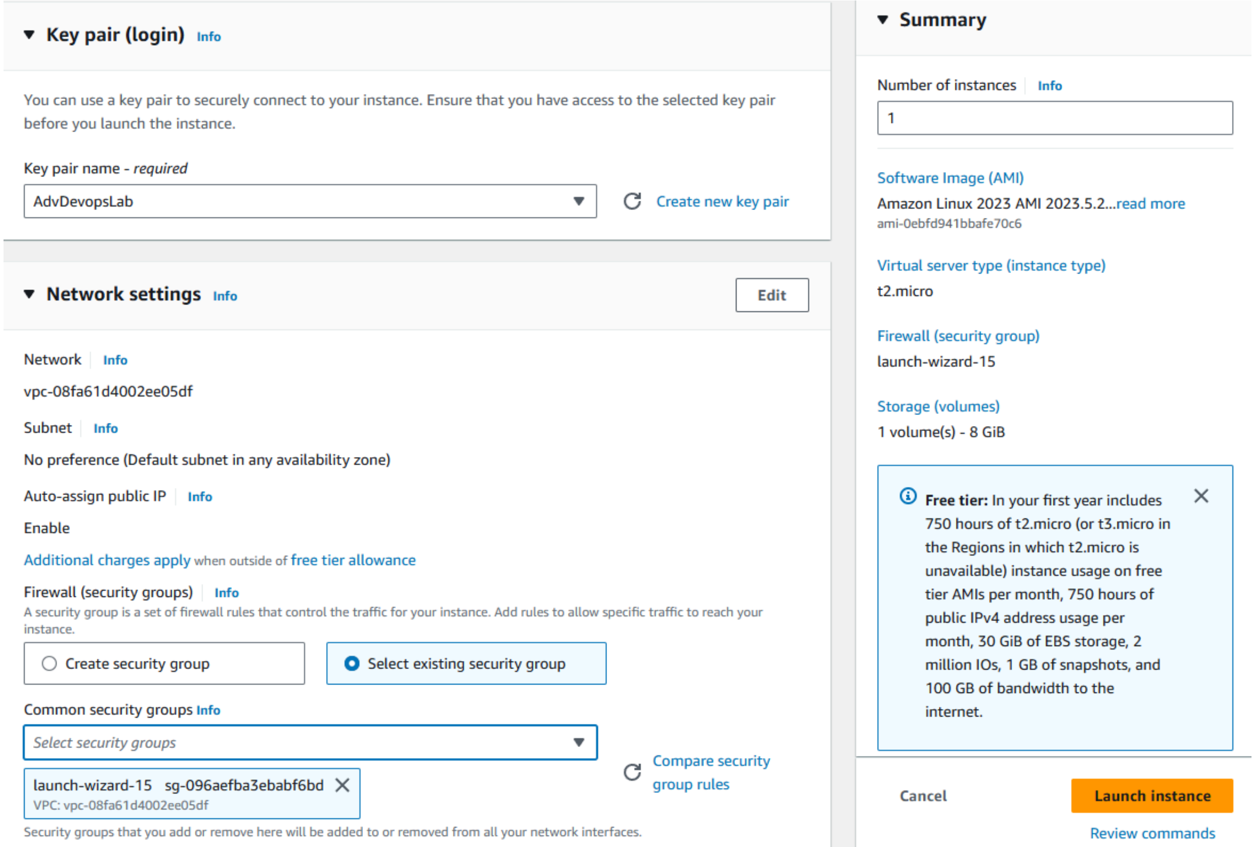
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-0ebfd941bbafe70c6

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year



▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

AdvDevopsLab ▼ Create new key pair

▼ **Network settings** Info [Edit]

Network Info

vpc-08fa61d4002ee05df

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups Info

Select security groups ▼

launch-wizard-15 sg-096aefba3ebabf6bd ✕
VPC: vpc-08fa61d4002ee05df

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ **Summary**

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.5.2...read more

ami-0ebfd941bbafe70c6

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-15

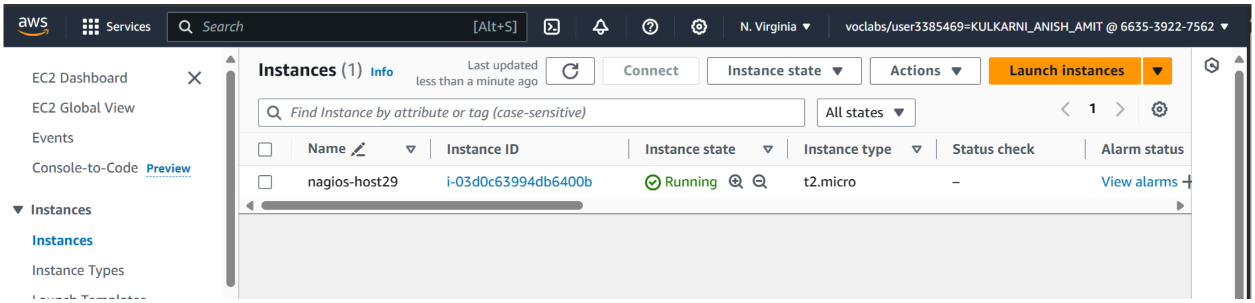
Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance]

Review commands



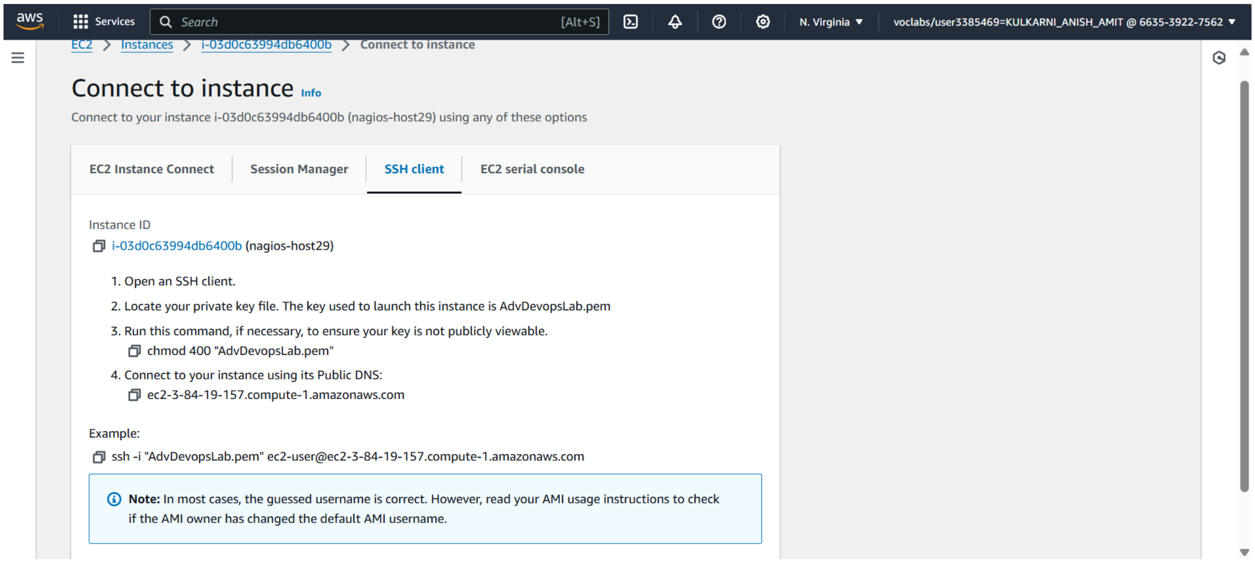
Instances (1) Info — Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	nagios-host29	i-03d0c63994db6400b	✓ Running	t2.micro	–	View alarms +

Your instance gets created.

Step 3: Click on the instance ID of your instance and click on ‘Connect’. Then, click on ‘SSH client’ and copy the command under ‘Example’.



aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3385469=KULKARNI_ANISH_AMIT @ 6635-3922-7562 ▼
EC2 > Instances > i-03d0c63994db6400b > Connect to instance

Connect to instance Info

Connect to your instance i-03d0c63994db6400b (nagios-host29) using any of these options

EC2 Instance Connect
Session Manager
SSH client
EC2 serial console

Instance ID

i-03d0c63994db6400b (nagios-host29)

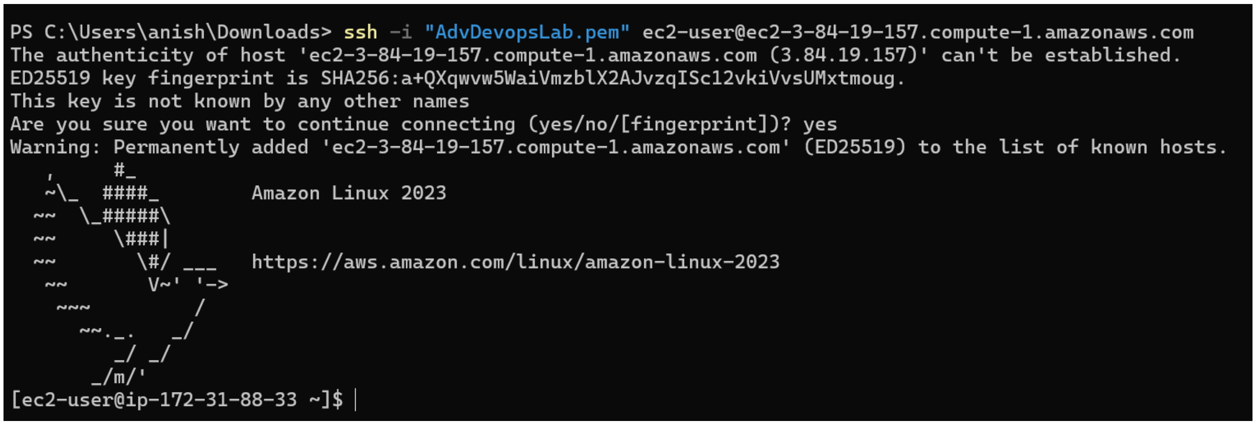
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is AdvDevopsLab.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "AdvDevopsLab.pem"
4. Connect to your instance using its Public DNS:
   ec2-3-84-19-157.compute-1.amazonaws.com

Example:

ssh -i "AdvDevopsLab.pem" ec2-user@ec2-3-84-19-157.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.

Step 4: Now, we need to connect our local terminal to the instance using SSH. To do so, open the terminal in the folder where the .pem file for your instance’s key pair is located and paste the SSH command that you copied in Step 3.



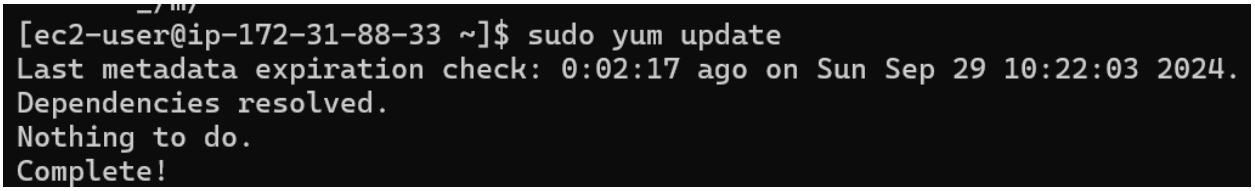
```
PS C:\Users\anish\Downloads> ssh -i "AdvDevopsLab.pem" ec2-user@ec2-3-84-19-157.compute-1.amazonaws.com
The authenticity of host 'ec2-3-84-19-157.compute-1.amazonaws.com (3.84.19.157)' can't be established.
ED25519 key fingerprint is SHA256:a+QXqwvw5WaiVmzblX2AJvzqISc12vkiVvsUMxtmoug.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-84-19-157.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-88-33 ~]$ 
```

This connects your instance to your local terminal using SSH.

Step 5: First, run the following command:-

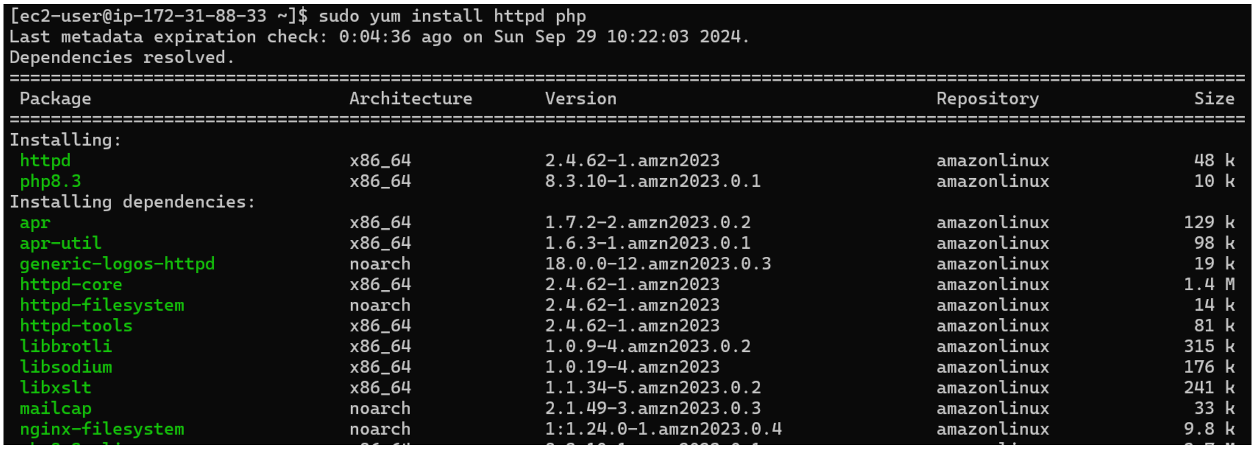
sudo yum update

This command will check for any updates for the YUM library.

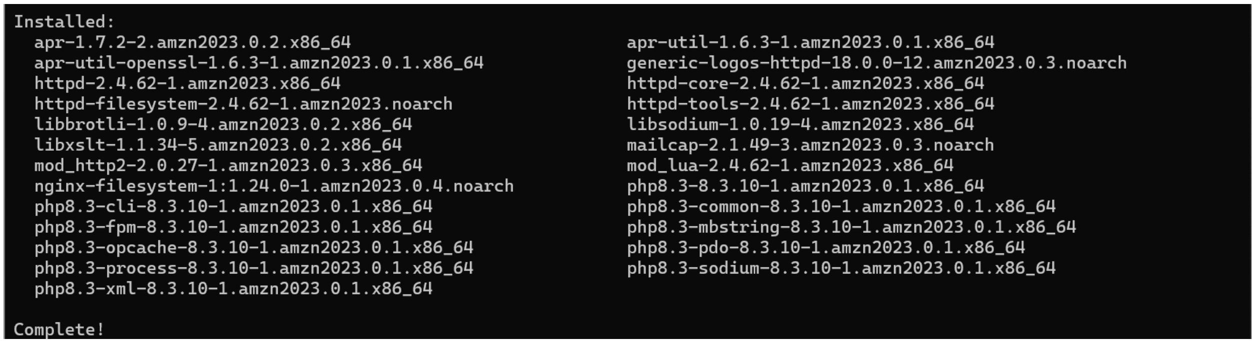


```
[ec2-user@ip-172-31-88-33 ~]$ sudo yum update
Last metadata expiration check: 0:02:17 ago on Sun Sep 29 10:22:03 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

Step 6: Run the command:
sudo yum install httpd php
This installs an Apache server and a PHP on your instance.

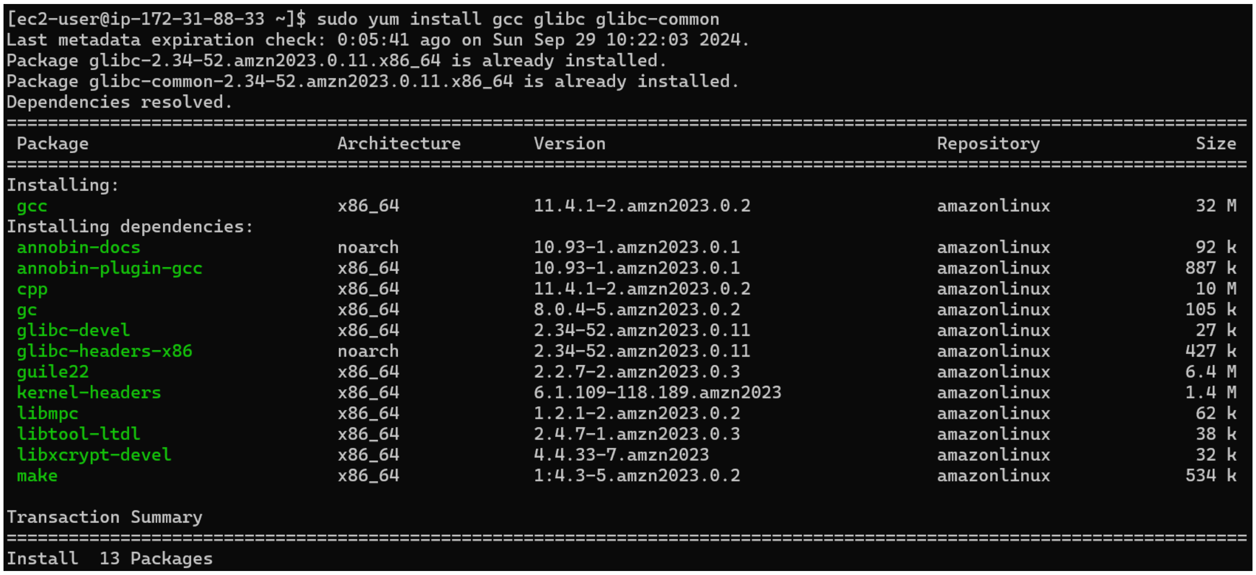


```
[ec2-user@ip-172-31-88-33 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:04:36 ago on Sun Sep 29 10:22:03 2024.
Dependencies resolved.
========================================================================================================================
 Package                          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
```

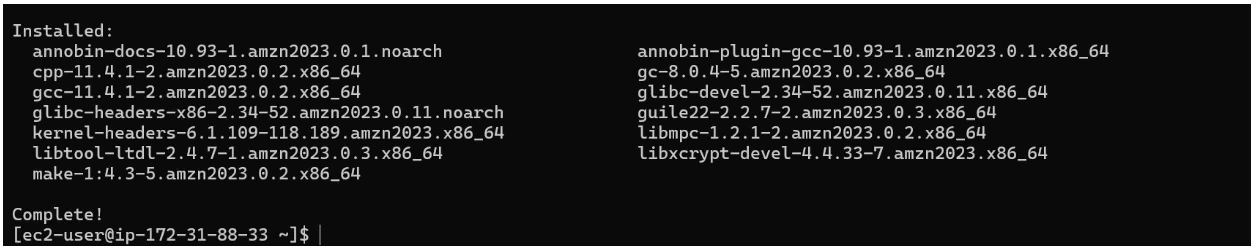


```
Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64                          apr-util-1.6.3-1.amzn2023.0.1.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64             generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.62-1.amzn2023.x86_64                           httpd-core-2.4.62-1.amzn2023.x86_64
  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!
```

Step 7: 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.

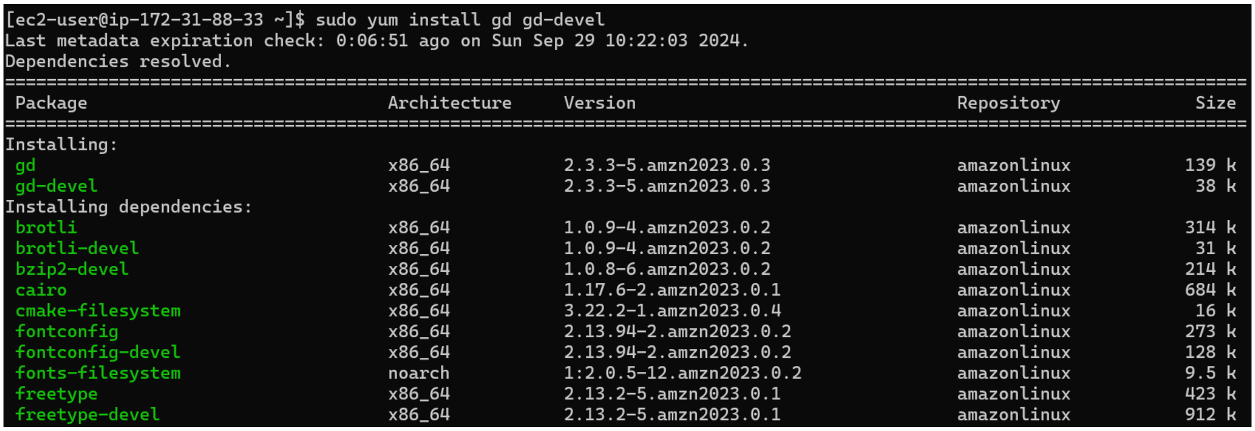


```
[ec2-user@ip-172-31-88-33 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:05:41 ago on Sun Sep 29 10:22:03 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
========================================================================================================================
 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
```

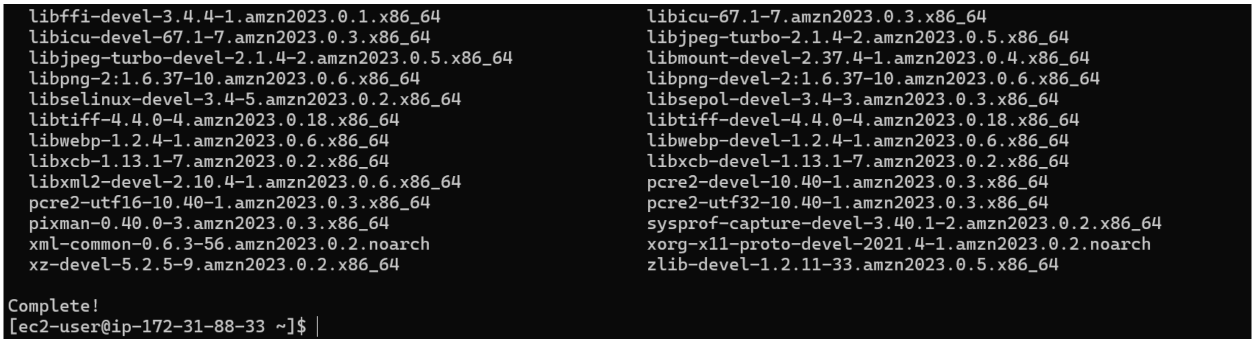


```
Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch                annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
  cpp-11.4.1-2.amzn2023.0.2.x86_64                        gc-8.0.4-5.amzn2023.0.2.x86_64
  gcc-11.4.1-2.amzn2023.0.2.x86_64                        glibc-devel-2.34-52.amzn2023.0.11.x86_64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

Complete!
[ec2-user@ip-172-31-88-33 ~]$
```

Step 8: Run the command:
sudo yum install gd gd-devel

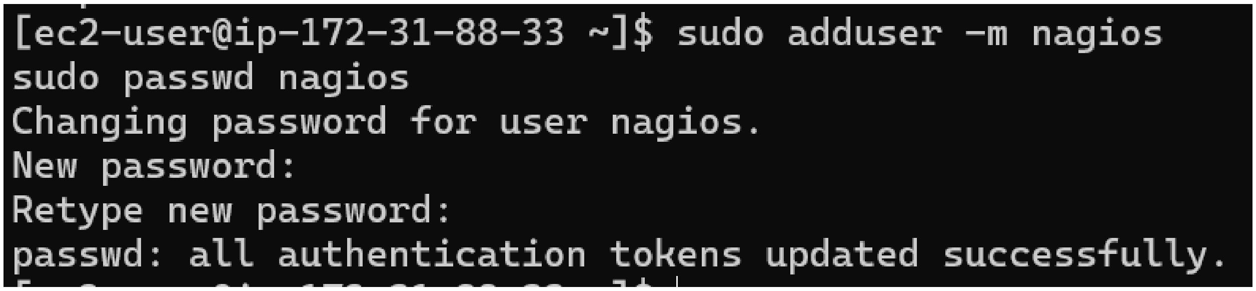


```
[ec2-user@ip-172-31-88-33 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:06:51 ago on Sun Sep 29 10:22:03 2024.
Dependencies resolved.
========================================================================================================================
 Package                             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
```



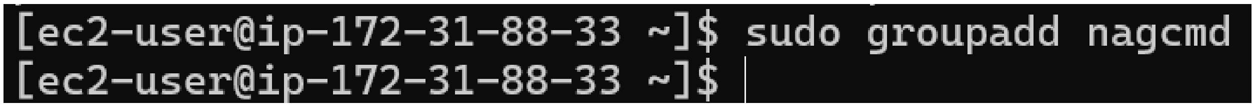
```
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64                 libicu-67.1-7.amzn2023.0.3.x86_64
  libicu-devel-67.1-7.amzn2023.0.3.x86_64                  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
  libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64          libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
  libpng-2:1.6.37-10.amzn2023.0.6.x86_64                   libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
  libselinux-devel-3.4-5.amzn2023.0.2.x86_64               libsepol-devel-3.4-3.amzn2023.0.3.x86_64
  libtiff-4.4.0-4.amzn2023.0.18.x86_64                     libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
  libwebp-1.2.4-1.amzn2023.0.6.x86_64                      libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-1.13.1-7.amzn2023.0.2.x86_64                      libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
  libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64               pcre2-devel-10.40-1.amzn2023.0.3.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                      sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch                  xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64                     zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
[ec2-user@ip-172-31-88-33 ~]$
```

Step 9: Run the commands:
sudo adduser -m nagios
sudo passwd nagios
This creates a user named ‘nagios’, ensures it has a home directory and sets up a password for it.



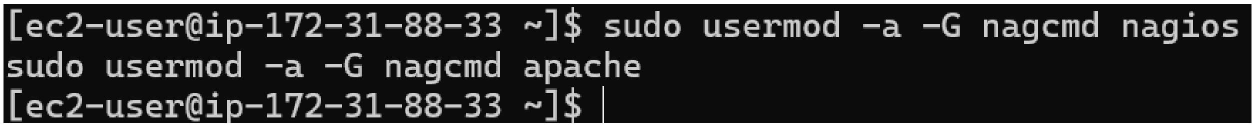
```
[ec2-user@ip-172-31-88-33 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Step 10: Create a user group named ‘nagcmd’ to execute nagios commands.
sudo groupadd nagcmd



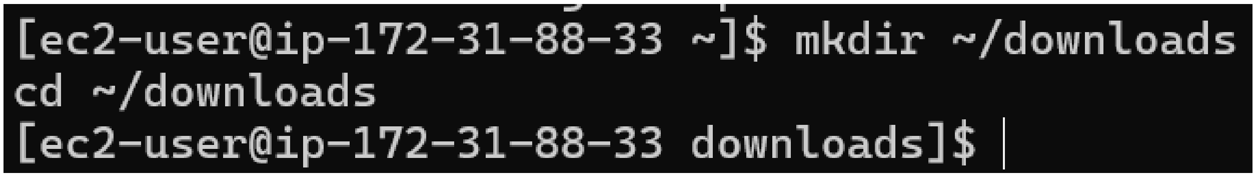
```
[ec2-user@ip-172-31-88-33 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-88-33 ~]$ 
```

Step 11: Add users apache and nagios to this user group.
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache



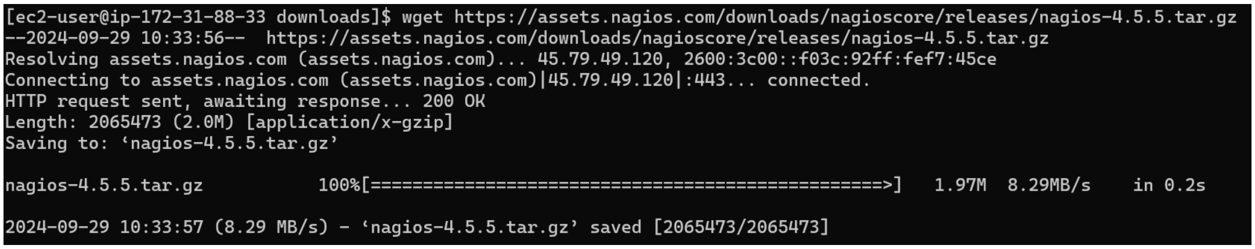
```
[ec2-user@ip-172-31-88-33 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-88-33 ~]$ 
```

Step 12: mkdir ~/downloads
cd ~/downloads
This creates a directory named ‘downloads’, to store the files of the nagios server that are downloaded.



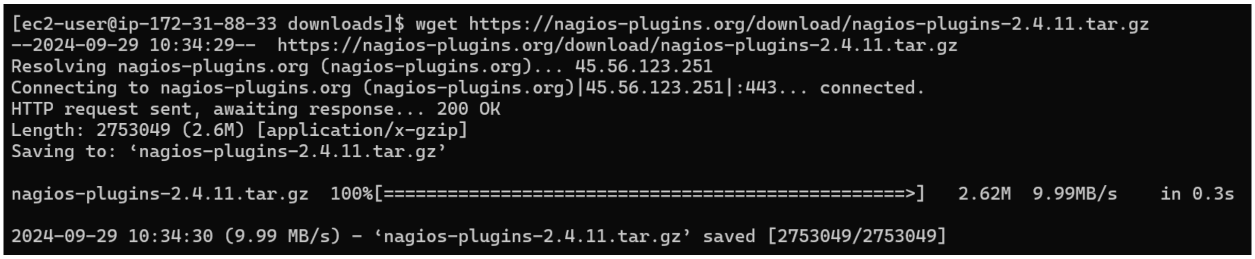
```
[ec2-user@ip-172-31-88-33 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-88-33 downloads]$ 
```

Step 13: wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
The above command installs the latest version of nagios-core.



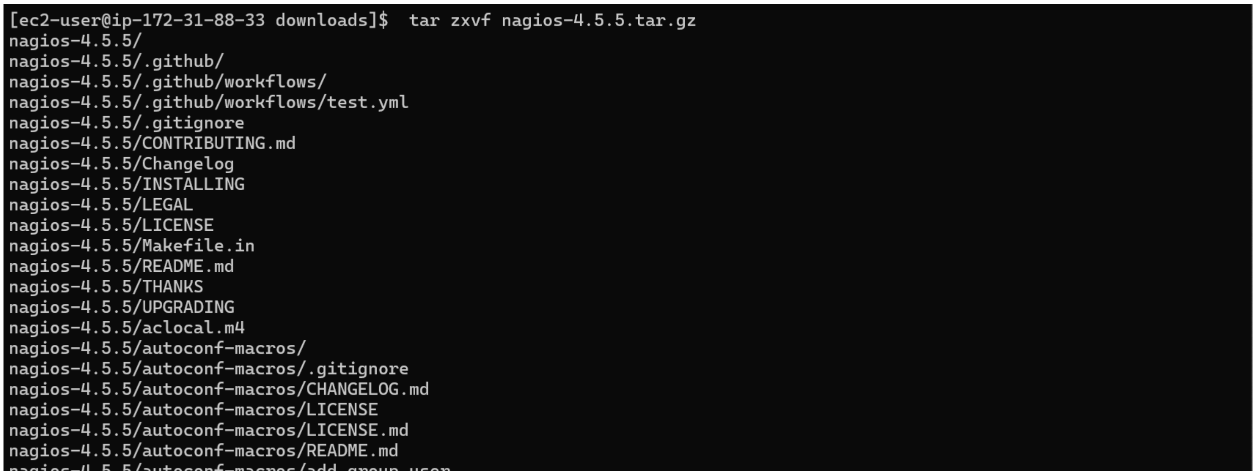
```
[ec2-user@ip-172-31-88-33 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-29 10:33:56--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
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  8.29MB/s    in 0.2s

2024-09-29 10:33:57 (8.29 MB/s) - ‘nagios-4.5.5.tar.gz’ saved [2065473/2065473]
```

Step 14: wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
The above command installs the latest version of nagios-plugins.

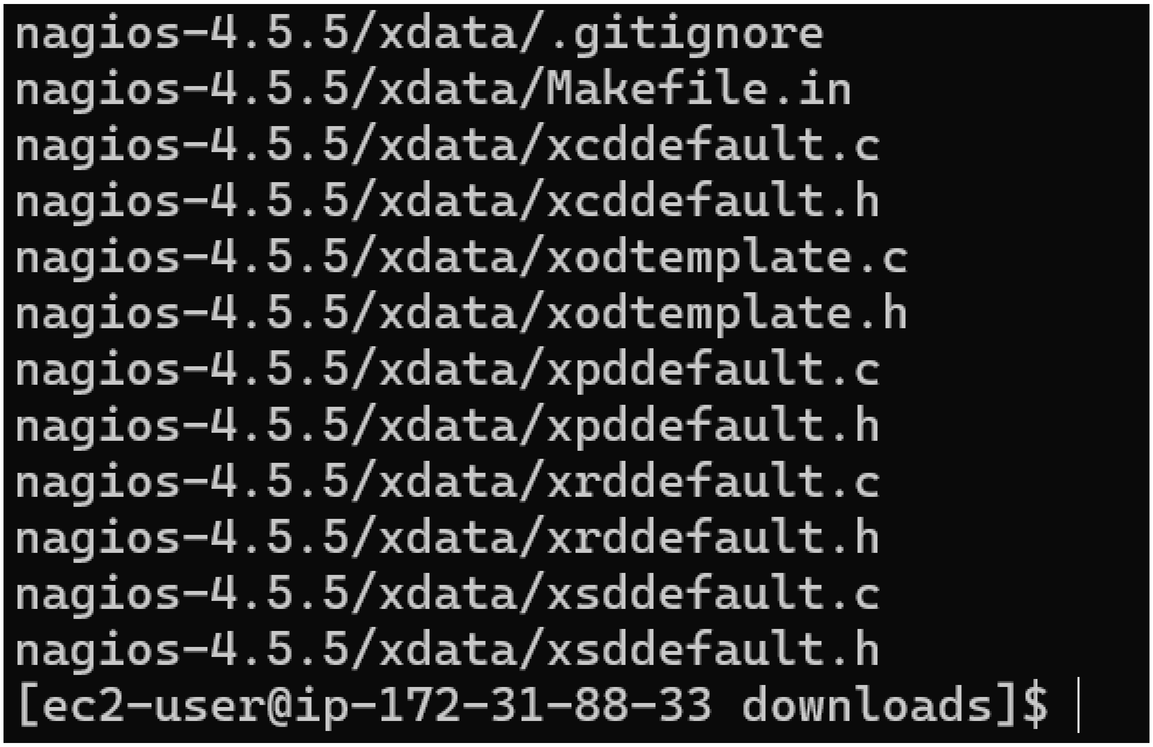


```
[ec2-user@ip-172-31-88-33 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-29 10:34:29--  https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (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  9.99MB/s    in 0.3s

2024-09-29 10:34:30 (9.99 MB/s) - ‘nagios-plugins-2.4.11.tar.gz’ saved [2753049/2753049]
```

Step 15: tar zxvf nagios-4.5.5.tar.gz
This extracts the nagios-core files into the same directory using the tar command.

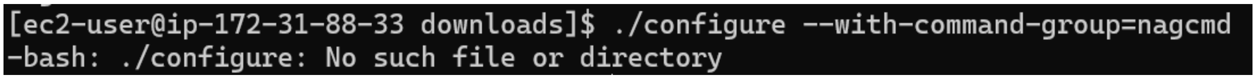


```
[ec2-user@ip-172-31-88-33 downloads]$  tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
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/xdata/.gitignore
nagios-4.5.5/xdata/Makefile.in
nagios-4.5.5/xdata/xcddefault.c
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-172-31-88-33 downloads]$
```

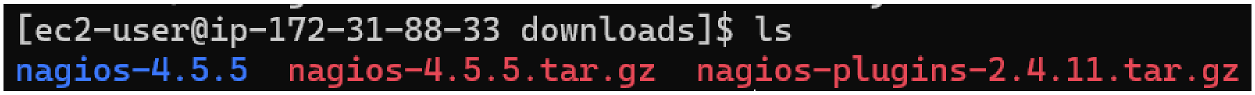
Step 16: ./configure --with-command-group=nagcmd
This command ensures that Nagios uses a specific group (in this case, nagcmd) for executing external commands.



```
[ec2-user@ip-172-31-88-33 downloads]$ ./configure --with-command-group=nagcmd
-bash: ./configure: No such file or directory
```

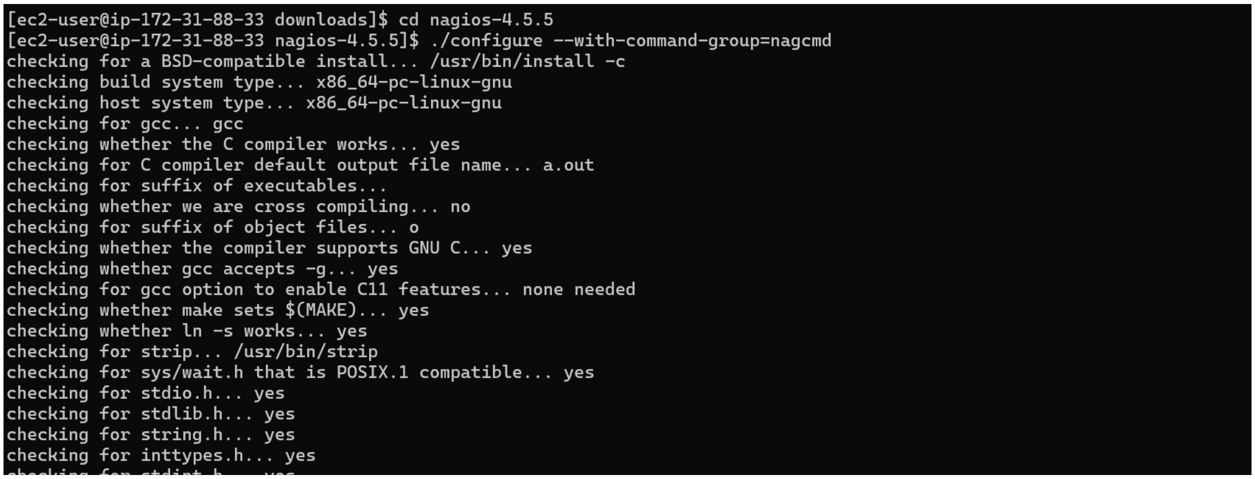
But, we encounter an error as we weren’t in the correct directory.

Use ‘ls’ command to find the correct directory.

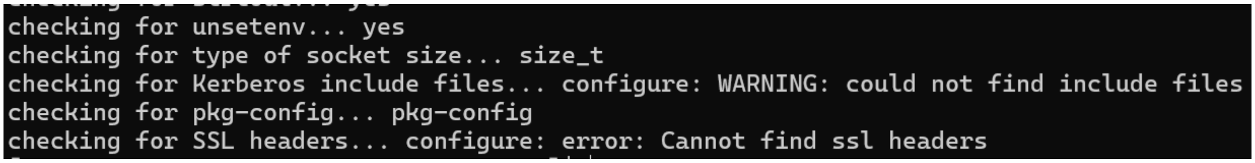


```
[ec2-user@ip-172-31-88-33 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
```

Use cd to change directory to the correct directory. Then, run the ‘./configure --with-command-group=nagcmd’ command again.



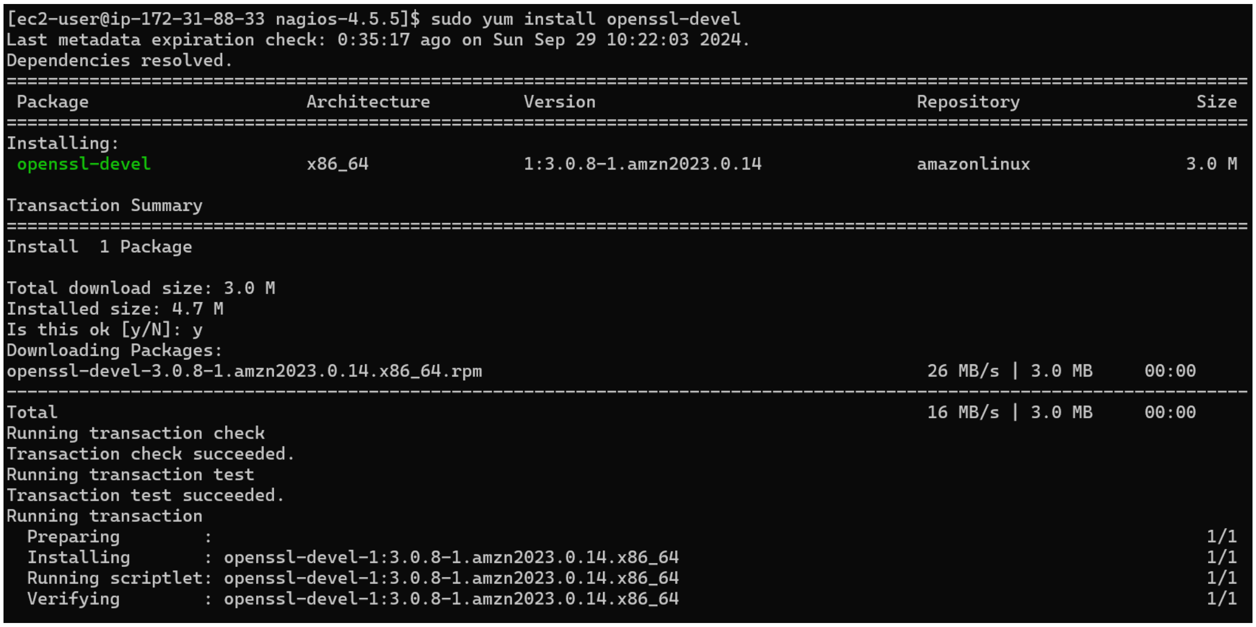
```
[ec2-user@ip-172-31-88-33 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-88-33 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking 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
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
```



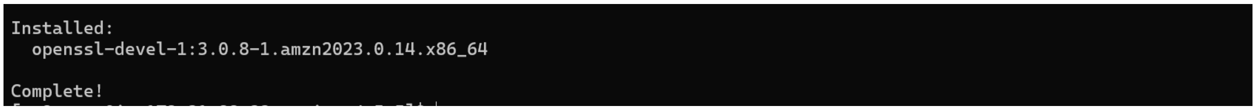
```
checking for unsetenv... yes
checking for type of socket size... size_t
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

Another error occurs which says that ssl headers cannot be found.

To fix the above error, run the ‘sudo yum install openssl-devel’ command.

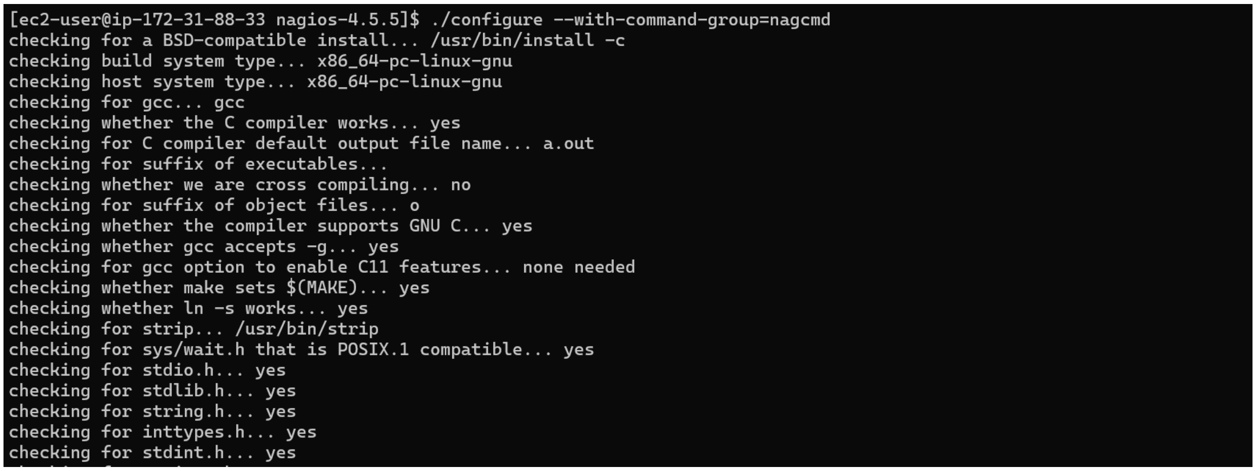


```
[ec2-user@ip-172-31-88-33 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:35:17 ago on Sun Sep 29 10:22:03 2024.
Dependencies resolved.
========================================================================================================================
 Package                    Architecture         Version                                Repository                 Size
========================================================================================================================
Installing:
 openssl-devel              x86_64               1:3.0.8-1.amzn2023.0.14                amazonlinux               3.0 M

Transaction Summary
========================================================================================================================
Install  1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm                                          26 MB/s | 3.0 MB     00:00
------------------------------------------------------------------------------------------------------------------------
Total                                                                                   16 MB/s | 3.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                1/1
  Installing       : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                   1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                   1/1
  Verifying        : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                   1/1
```

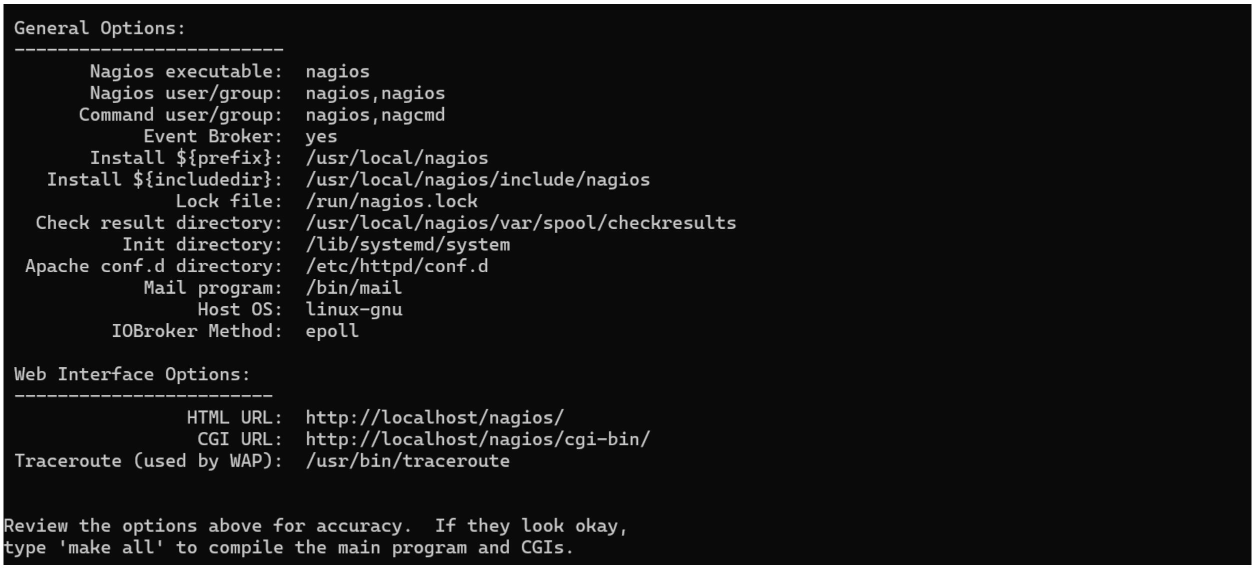


```
Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64

Complete!
```

Then, run the ‘./configure --with-command-group=nagcmd’ command again.



```
[ec2-user@ip-172-31-88-33 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
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
```



```
 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

 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/bin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

Step 17: 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

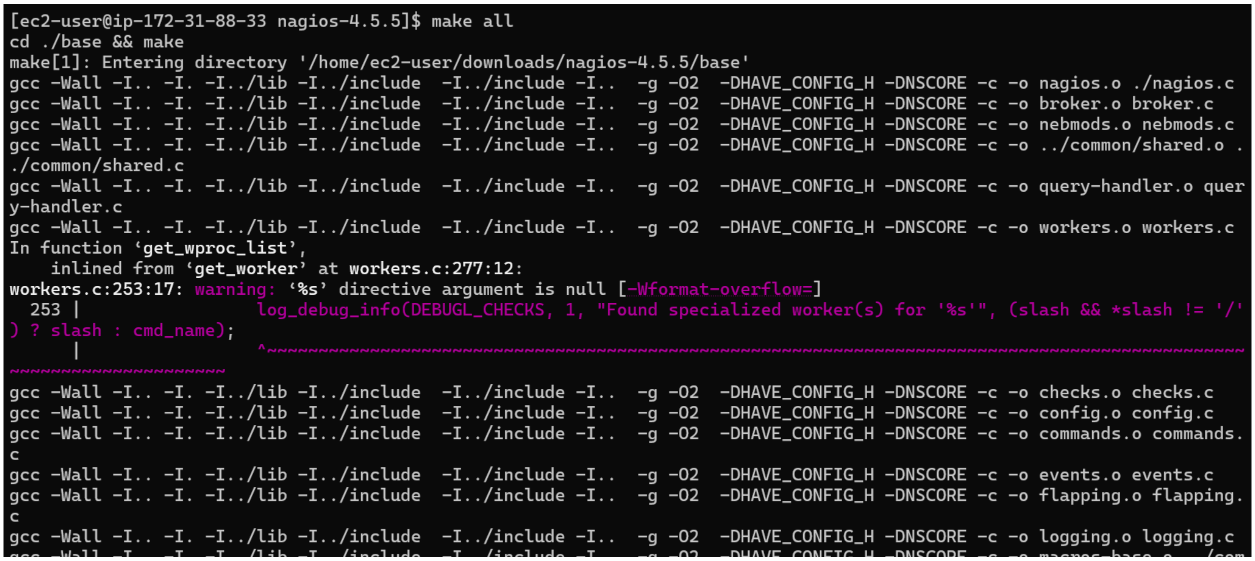
Then,

sudo make install

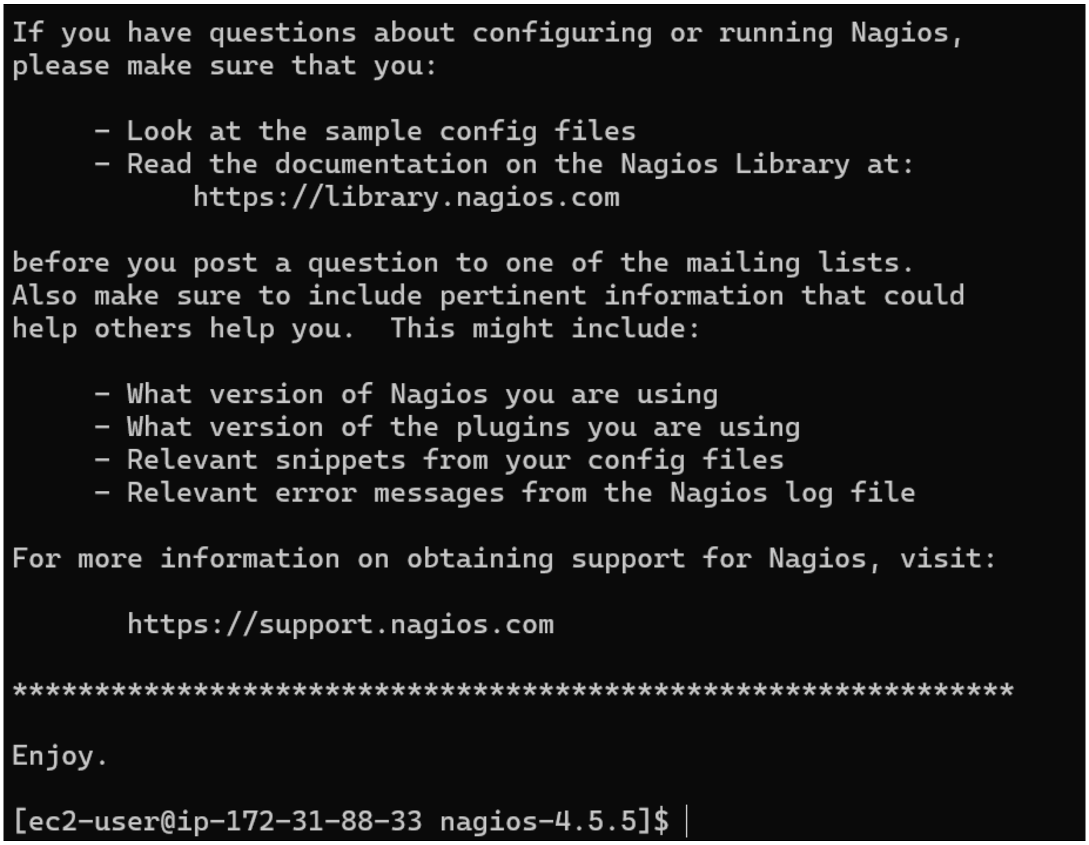
sudo make install-init

sudo make install-config

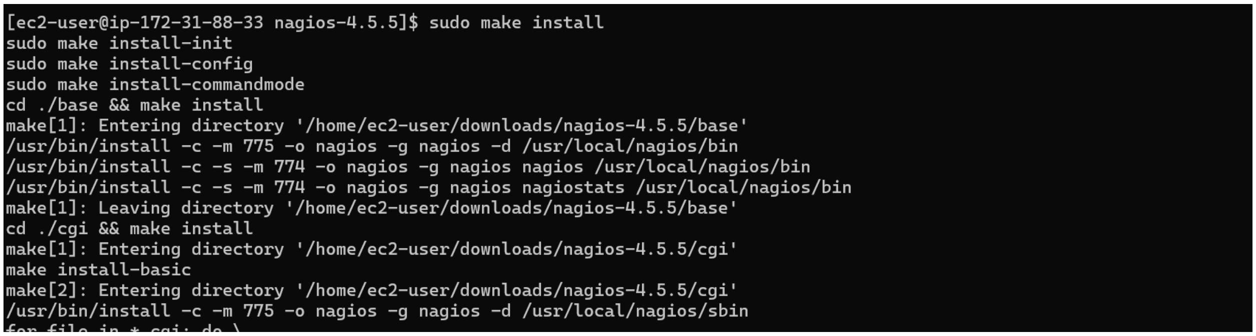
sudo make install-commandmode



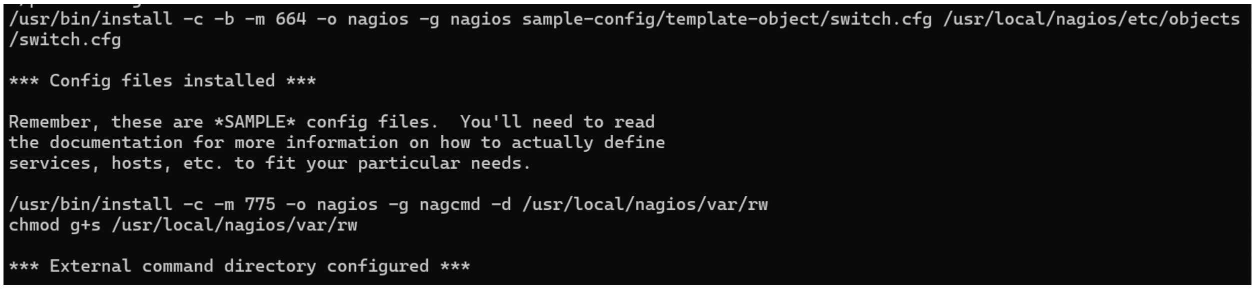
```
[ec2-user@ip-172-31-88-33 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o .
./common/shared.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o quer
y-handler.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function ‘get_wproc_list’,
    inlined from ‘get_worker’ at workers.c:277:12:
workers.c:253:17: warning: ‘%s’ directive argument is null [-Wformat-overflow=]
  253 |                 log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/'
) ? slash : cmd_name);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.
c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.
c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o logging.o logging.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../com
```



```
If you have questions about configuring or running Nagios,
please make sure that you:

     - Look at the sample config files
     - Read the documentation on the Nagios Library at:
           https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

     - What version of Nagios you are using
     - What version of the plugins you are using
     - Relevant snippets from your config files
     - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

*************************************************************

Enjoy.

[ec2-user@ip-172-31-88-33 nagios-4.5.5]$ 
```



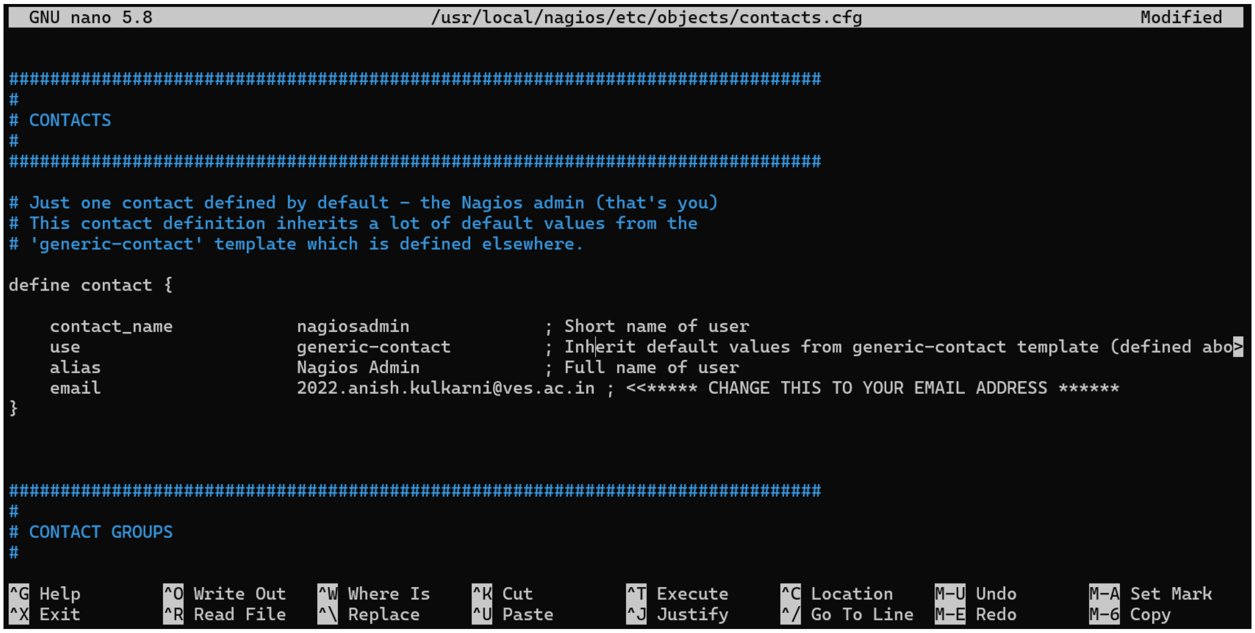
```
[ec2-user@ip-172-31-88-33 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/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'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
```



```
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects
/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

/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 ***
```

Step 18: 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

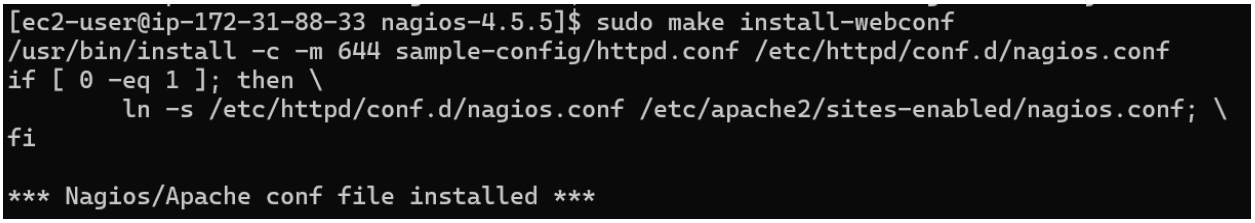


```
  GNU nano 5.8                       /usr/local/nagios/etc/objects/contacts.cfg                                Modified


###############################################################################
#
# 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 (defined abo>
    alias                   Nagios Admin            ; Full name of user
    email                   2022.anish.kulkarni@ves.ac.in ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}


###############################################################################
#
# CONTACT GROUPS
#
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy
```

In the email section, enter your email address. Then, ‘Write out’ your file and ‘Exit’.

Step 19: sudo make install-webconf

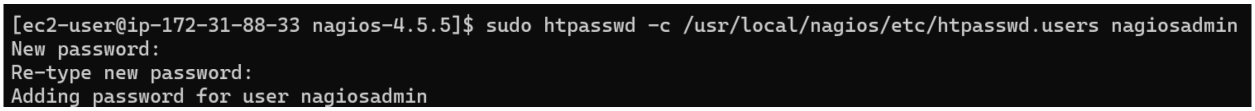
This installs the necessary configuration files for the Nagios web interface.



```
[ec2-user@ip-172-31-88-33 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 20: sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

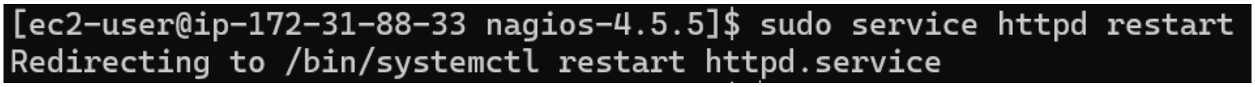
This creates a user named 'nagiosadmin' to access the nagios web interface. Create a password and keep it in mind as it will be required in the future steps.



```
[ec2-user@ip-172-31-88-33 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
```

Step 21: Restart the apache server to apply all the recent configurations.

sudo service httpd restart

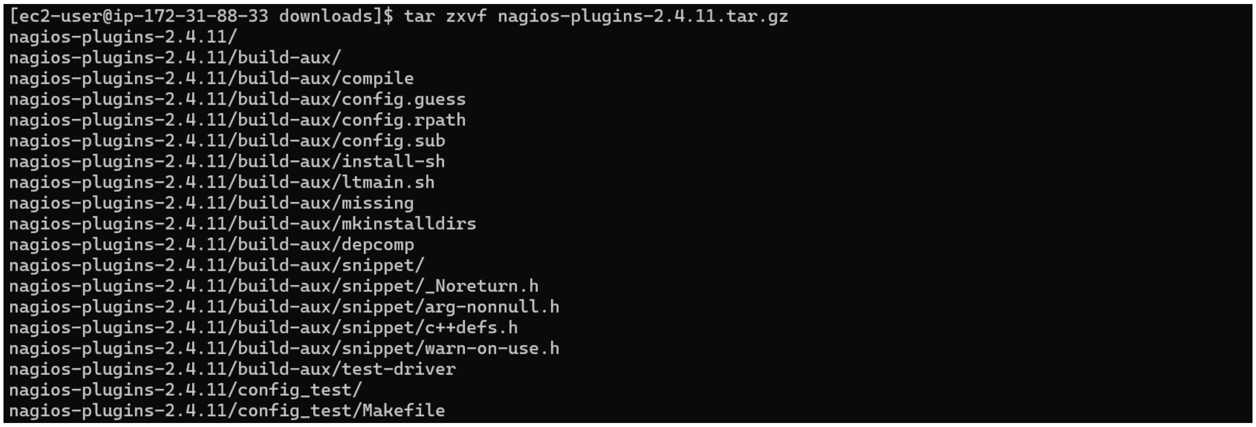


```
[ec2-user@ip-172-31-88-33 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

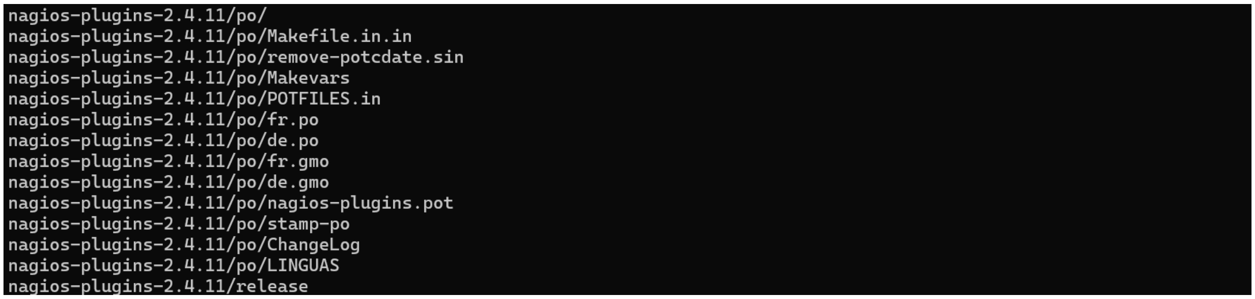
Step 22: cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz

This changes the directory to the 'downloads' directory and extracts the files for nagios-plugins.

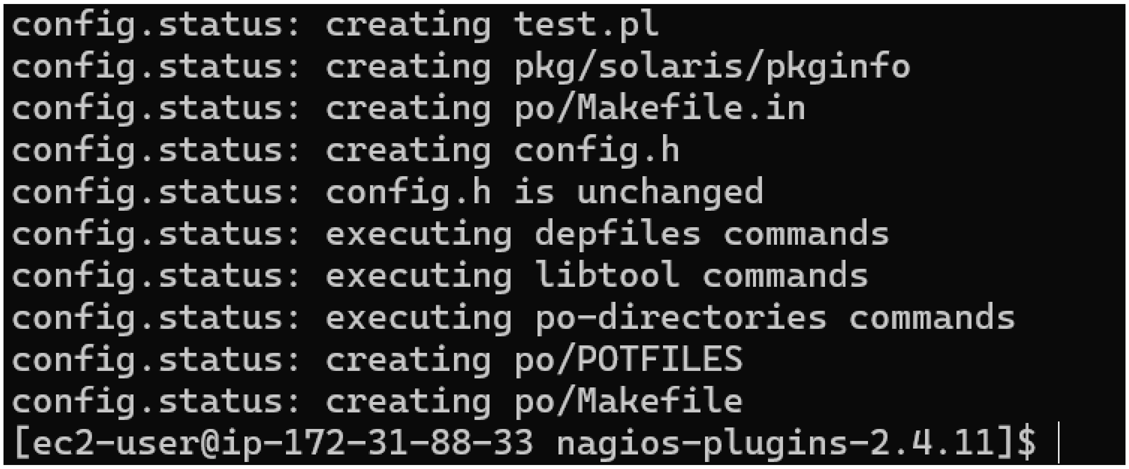


```
[ec2-user@ip-172-31-88-33 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
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/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
```



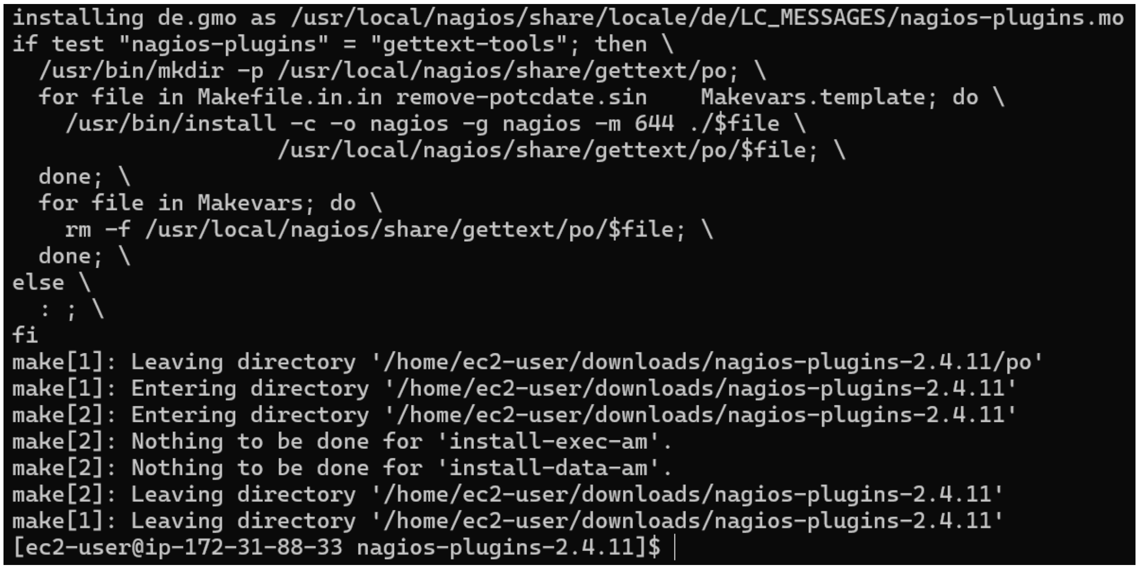
```
nagios-plugins-2.4.11/po/
nagios-plugins-2.4.11/po/Makefile.in.in
nagios-plugins-2.4.11/po/remove-potcdate.sin
nagios-plugins-2.4.11/po/Makevars
nagios-plugins-2.4.11/po/POTFILES.in
nagios-plugins-2.4.11/po/fr.po
nagios-plugins-2.4.11/po/de.po
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
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
```

Step 23: cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
This installs the configurations for the nagios-plugins files.



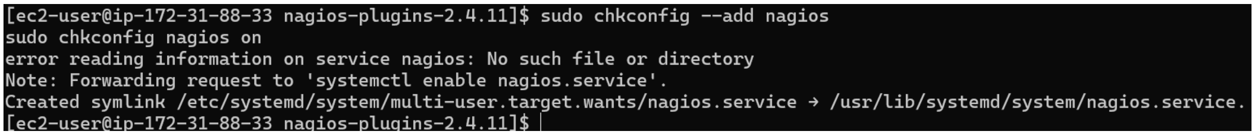
```
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$
```

Step 24: Next, we must compile all components of this software according to the instructions in the Makefile. To do so, use the following commands:
make
sudo make install



```
installing de.gmo as /usr/local/nagios/share/locale/de/LC_MESSAGES/nagios-plugins.mo
if test "nagios-plugins" = "gettext-tools"; then \
  /usr/bin/mkdir -p /usr/local/nagios/share/gettext/po; \
  for file in Makefile.in.in remove-potcdate.sin    Makevars.template; do \
    /usr/bin/install -c -o nagios -g nagios -m 644 ./$file \
                    /usr/local/nagios/share/gettext/po/$file; \
  done; \
  for file in Makevars; do \
    rm -f /usr/local/nagios/share/gettext/po/$file; \
  done; \
else \
  : ; \
fi
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$
```

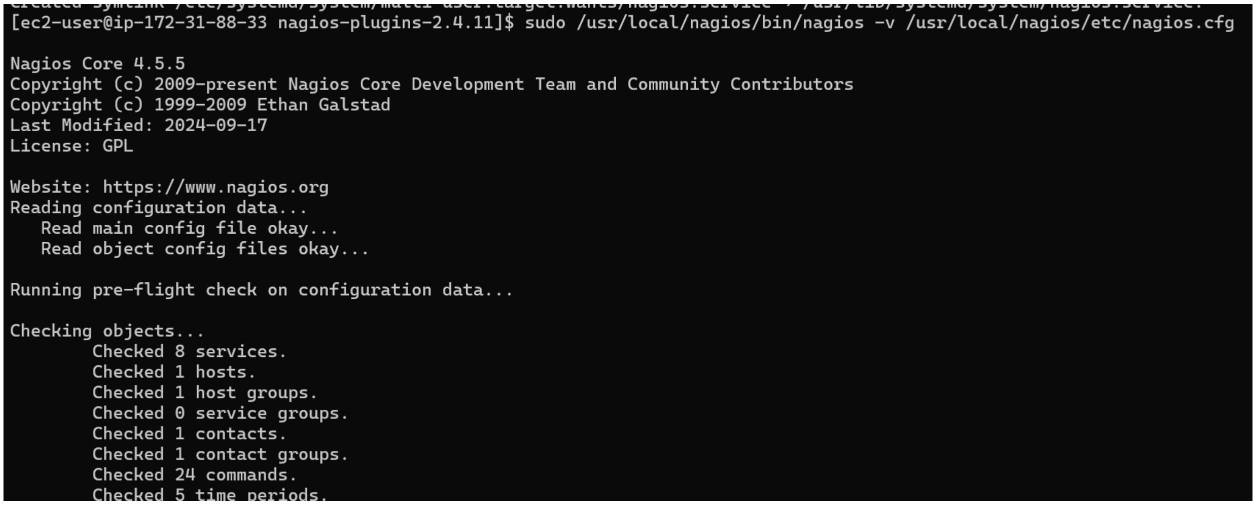
Step 25: sudo chkconfig --add nagios
sudo chkconfig nagios on
This registers the Nagios service with the system ensuring that it can manage the server status.



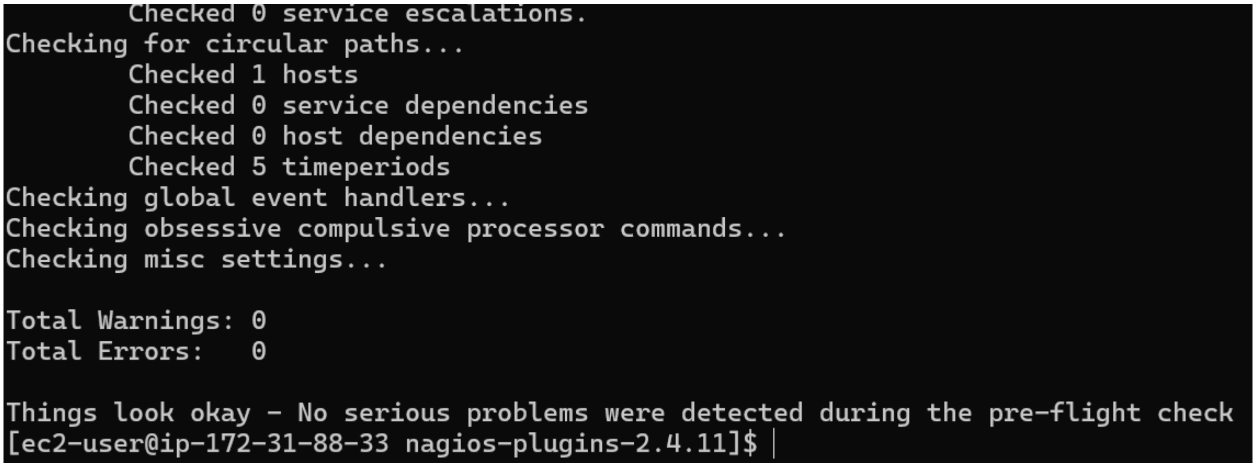
```
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$
```

Step 26: sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

This command checks and verifies that the sample configuration files has no errors.



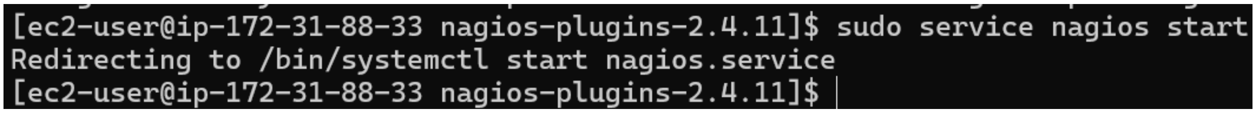
```
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 1 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
```



```
        Checked 0 service escalations.
Checking for circular paths...
        Checked 1 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$
```

Step 27: sudo service nagios start

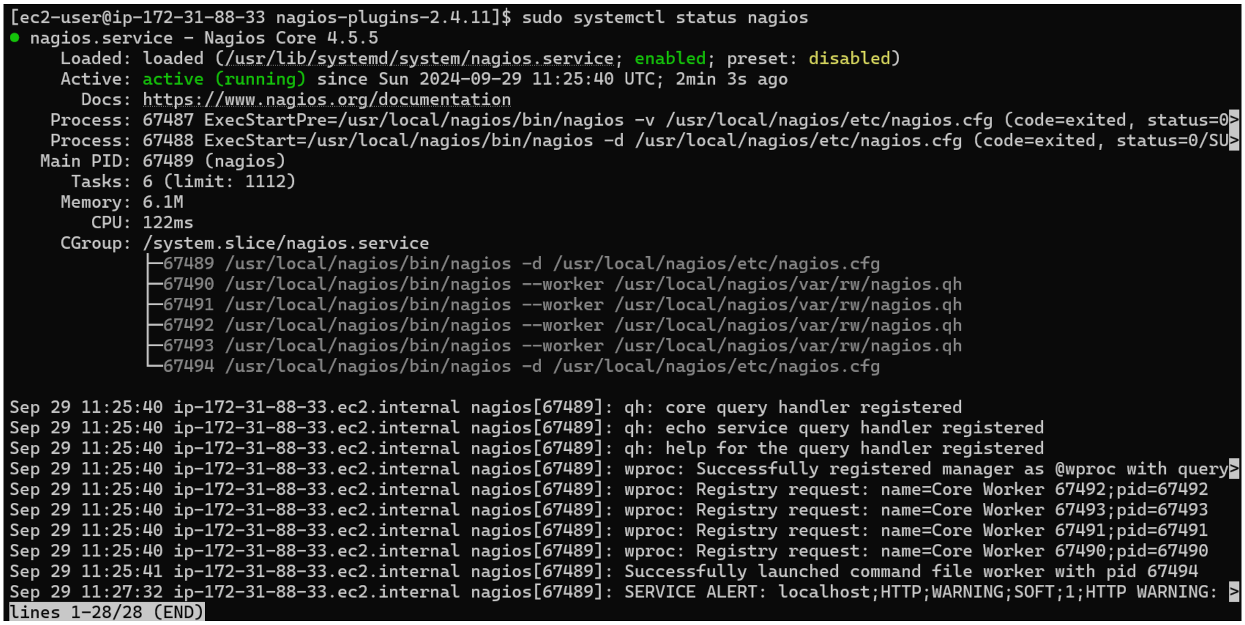
This starts the Nagios service.



```
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-88-33 nagios-plugins-2.4.11]$
```

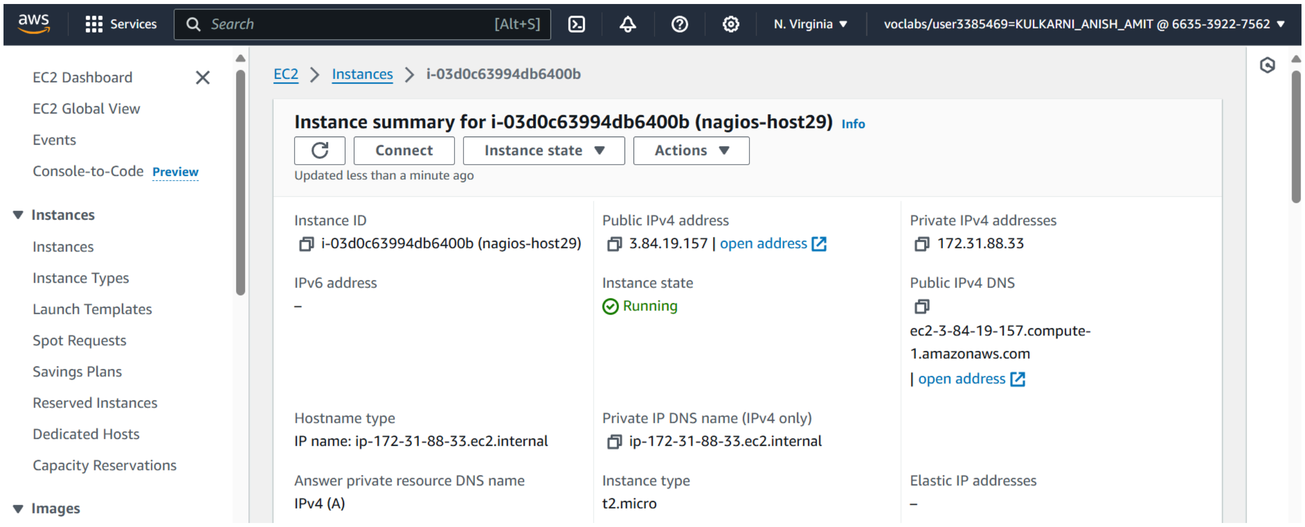
Step 28: sudo systemctl status nagios

This checks the status of Nagios. Ensure that it is ‘active(running)’.



```
[ec2-user@ip-172-31-88-33 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 Sun 2024-09-29 11:25:40 UTC; 2min 3s ago
       Docs: https://www.nagios.org/documentation
    Process: 67487 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0>
    Process: 67488 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU>
   Main PID: 67489 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 6.1M
        CPU: 122ms
     CGroup: /system.slice/nagios.service
             ├─67489 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─67490 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67491 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67492 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67493 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─67494 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: qh: core query handler registered
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: qh: echo service query handler registered
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: qh: help for the query handler registered
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: wproc: Successfully registered manager as @wproc with query>
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: wproc: Registry request: name=Core Worker 67492;pid=67492
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: wproc: Registry request: name=Core Worker 67493;pid=67493
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: wproc: Registry request: name=Core Worker 67491;pid=67491
Sep 29 11:25:40 ip-172-31-88-33.ec2.internal nagios[67489]: wproc: Registry request: name=Core Worker 67490;pid=67490
Sep 29 11:25:41 ip-172-31-88-33.ec2.internal nagios[67489]: Successfully launched command file worker with pid 67494
Sep 29 11:27:32 ip-172-31-88-33.ec2.internal nagios[67489]: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;1;HTTP WARNING: >
lines 1-28/28 (END)
```

Step 29: Navigate to your EC2 instance and copy the public IPv4 address.



Instance summary for i-03d0c63994db6400b (nagios-host29)

Updated less than a minute ago

Instance ID: i-03d0c63994db6400b (nagios-host29)

Public IPv4 address: 3.84.19.157 | open address

Private IPv4 addresses: 172.31.88.33

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-84-19-157.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-88-33.ec2.internal

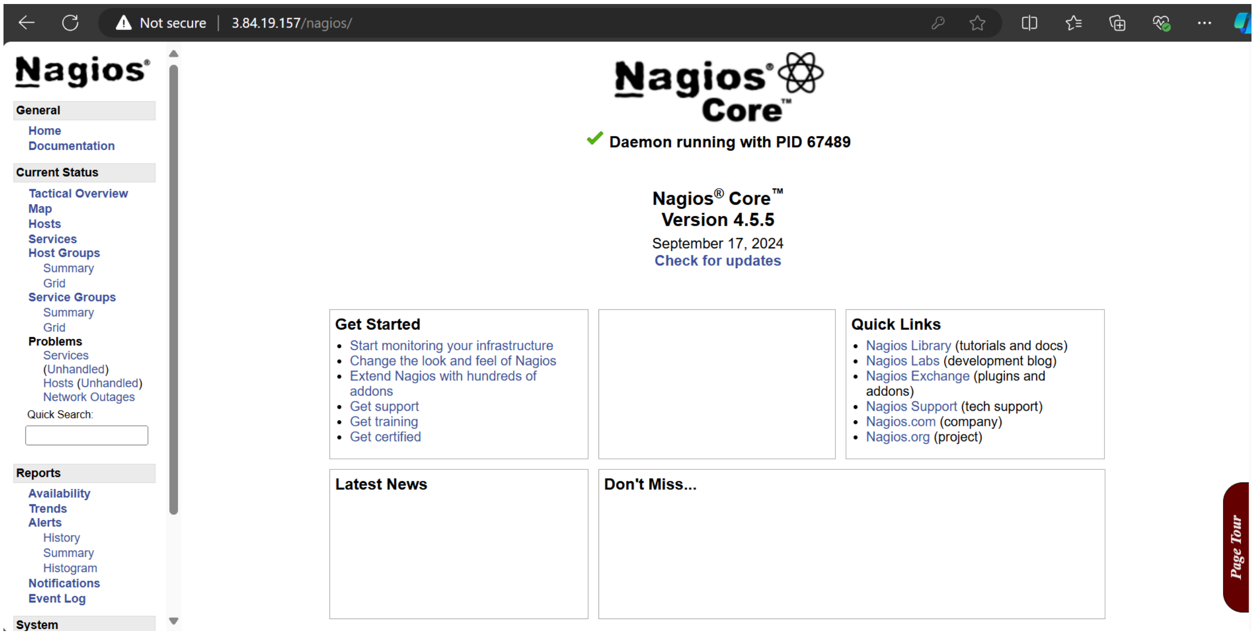
Private IP DNS name (IPv4 only): ip-172-31-88-33.ec2.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.micro

Elastic IP addresses: –

Step 30: In the address bar, enter ‘http://<publicipaddress>/nagios’.



✓ Daemon running with PID 67489

Nagios® Core™
Version 4.5.5
September 17, 2024
Check for updates

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...

The above page is visible.

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 created an EC2 Linux instance with the required security rules. Then, we installed the latest versions of nagios-core and nagios-plugins and configured them to ensure that they contained no errors. 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.