Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

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

Theory:

What is Nagios?

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

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitor and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Features of Nagios

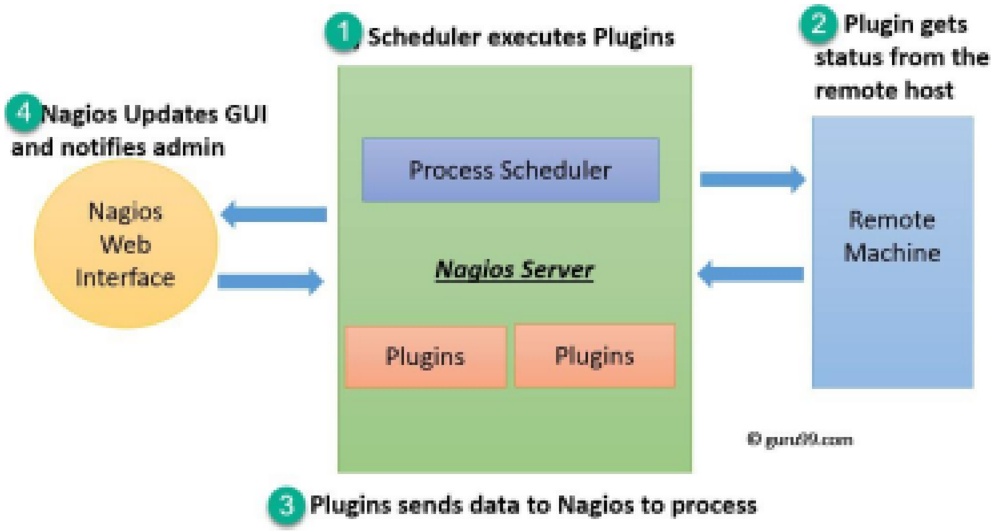
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts

- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

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


1 Scheduler executes Plugins
2 Plugin gets status from the remote host
4 Nagios Updates GUI and notifies admin
Nagios Web Interface
Process Scheduler
Nagios Server
Plugins
Plugins
Remote Machine
© guru99.com
3 Plugins sends data to Nagios to process

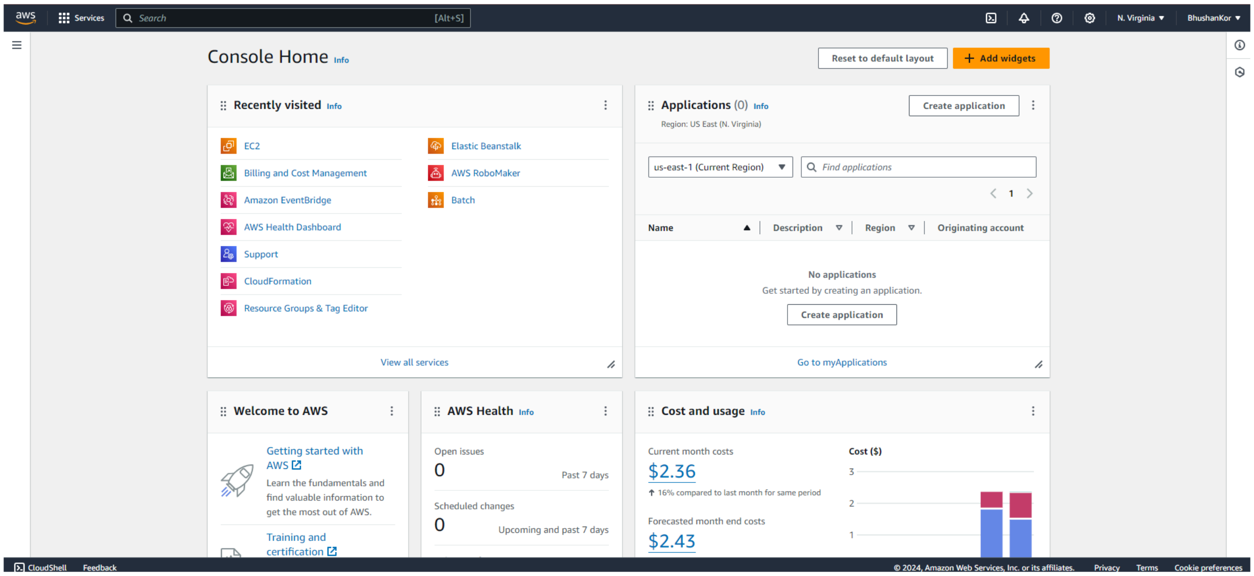
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

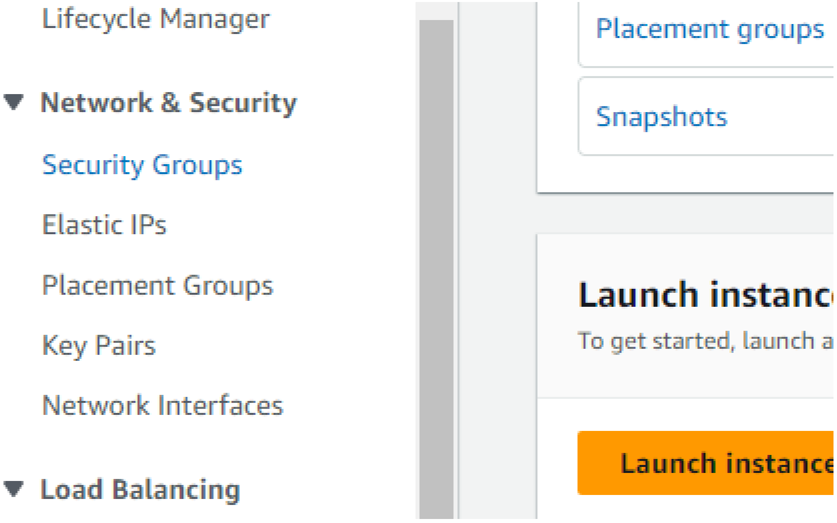
Prerequisites: AWS Personal or Academy Account.

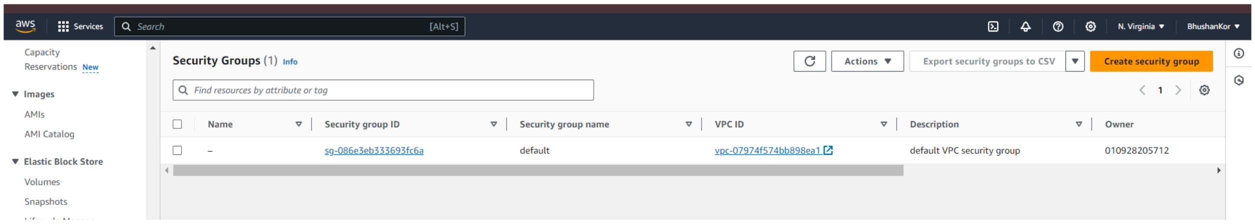
Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

Step 1: Login to your AWS account Personal / Academy. Click on EC2 instance then click on Create Security Group. Give the name as Nagios and any description and add the following inbounds rules.

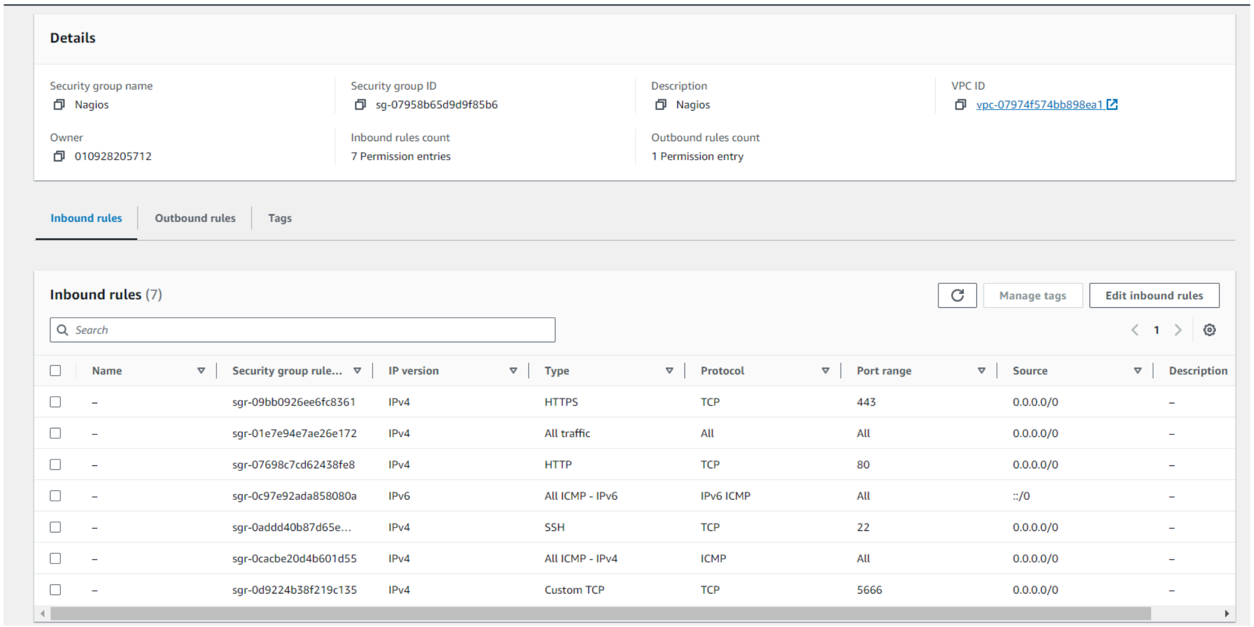






Security Groups (1) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner
–	sg-086e3eb333693fc6a	default	vpc-07974f574bb898ea1	default VPC security group	010928205712



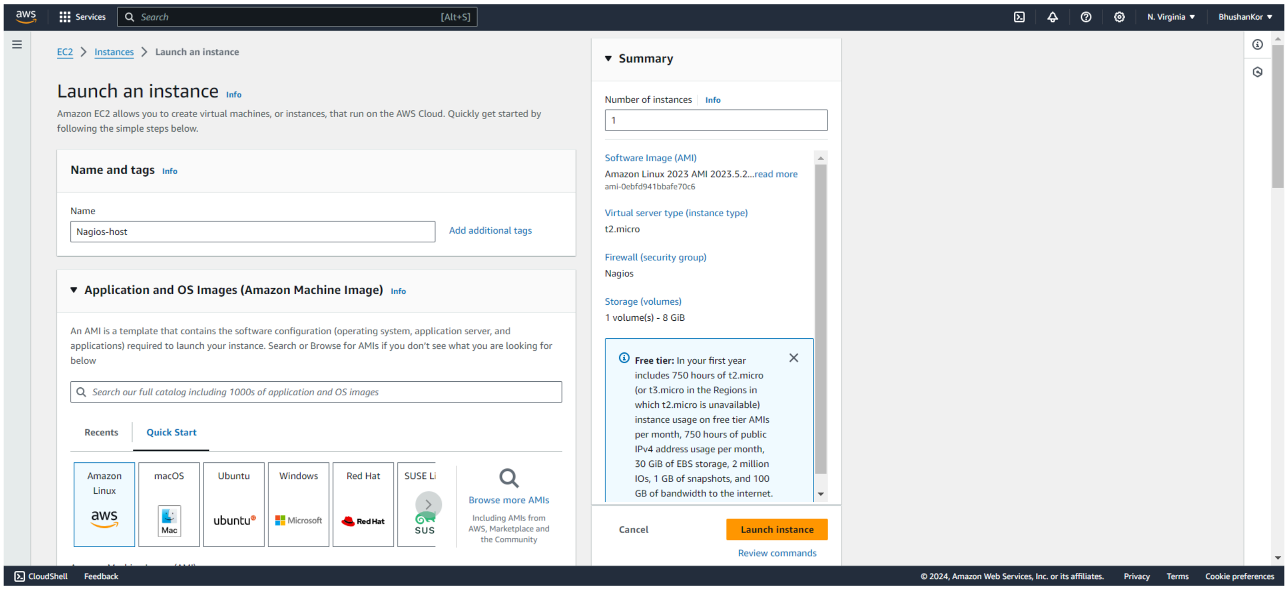
Details

Security group name Nagios	Security group ID sg-07958b65d9d9f85b6	Description Nagios	VPC ID vpc-07974f574bb898ea1
Owner 010928205712	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (7)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-09bb0926ee6fc8361	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
–	sgr-01e7e94e7ae26e172	IPv4	All traffic	All	All	0.0.0.0/0	–
–	sgr-07698c7cd62438fe8	IPv4	HTTP	TCP	80	0.0.0.0/0	–
–	sgr-0c97e92ada858080a	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	–
–	sgr-0addd40b87d65e...	IPv4	SSH	TCP	22	0.0.0.0/0	–
–	sgr-0cacbe20d4b601d55	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	–
–	sgr-0d9224b38f219c135	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	–

Step 2: Now Create a new EC2 instance. Name: Nagios-host ,AMI: Amazon Linux, Instance Type: t2.micro.



Launch an instance Info

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.

Name and tags Info

Name: Nagios-host

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
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances: 1

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

Virtual server type (instance type)
t2.micro

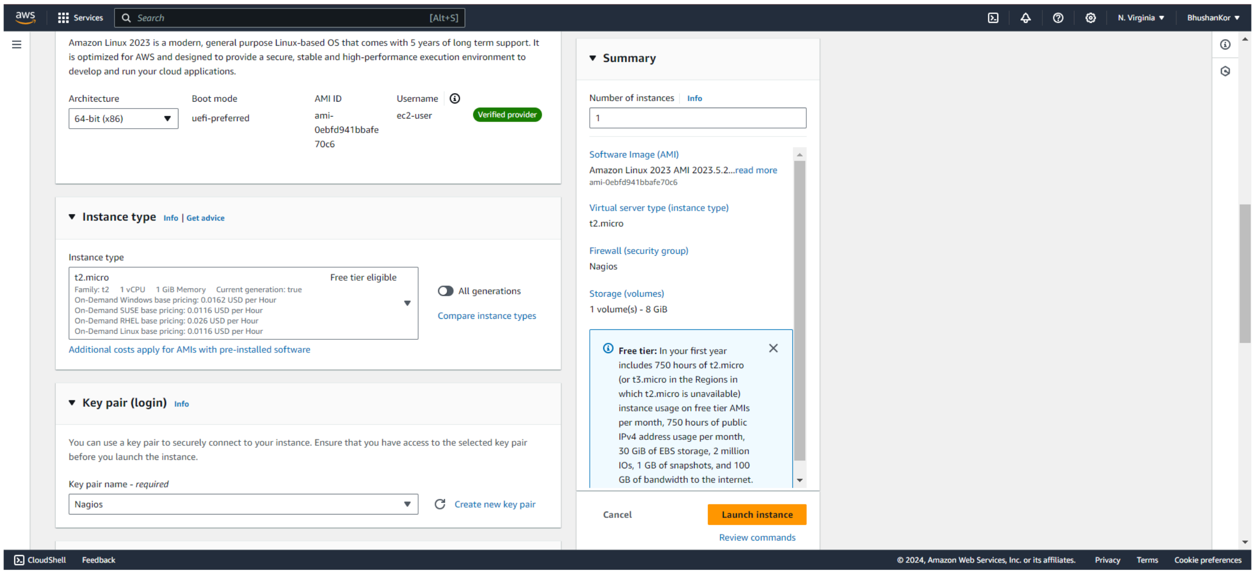
Firewall (security group)
Nagios

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.

For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID	Username	
64-bit (x86)	uefi-preferred	ami-0ebfd941bbafe70c6	ec2-user	Verified provider

▼ Instance type Info | Get advice

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software

▼ 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*
Nagios
Create new key pair

▼ 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

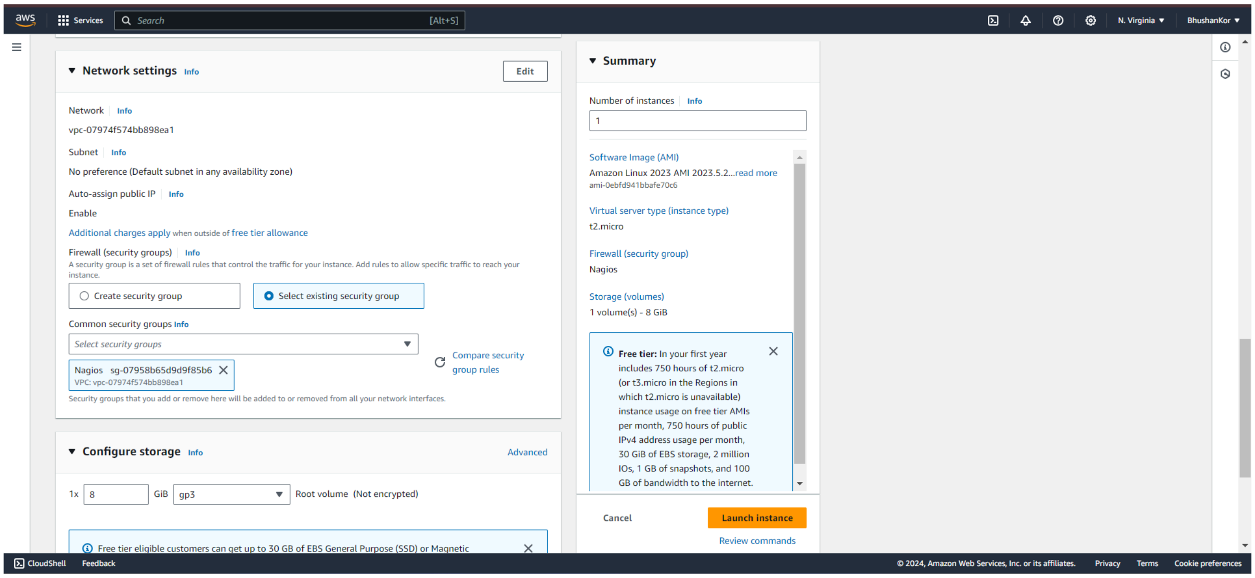
Firewall (security group)
Nagios

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

Select the Existing Security Group and select the Security Group we have created in Step 1.



▼ Network settings Info Edit

Network Info
vpc-07974f574bb898ea1

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

Nagios sg-07958b65d9d9f85b6
VPC: vpc-07974f574bb898ea1

Compare security group rules

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

▼ Configure storage Info Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic

▼ 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

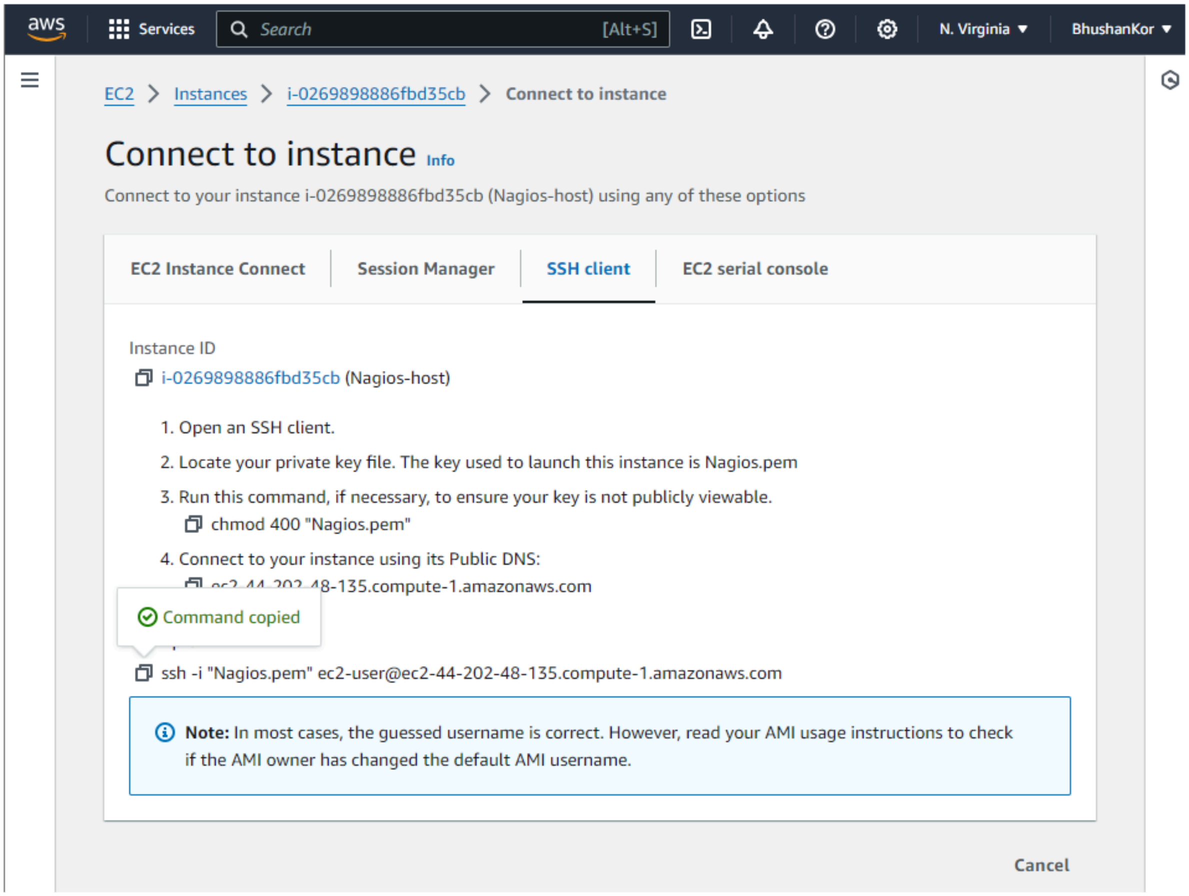
Firewall (security group)
Nagios

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

Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .



EC2 > Instances > i-0269898886fbd35cb > Connect to instance

Connect to instance Info

Connect to your instance i-0269898886fbd35cb (Nagios-host) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID

i-0269898886fbd35cb (Nagios-host)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Nagios.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "Nagios.pem"
4. Connect to your instance using its Public DNS:
   ec2-44-202-48-135.compute-1.amazonaws.com

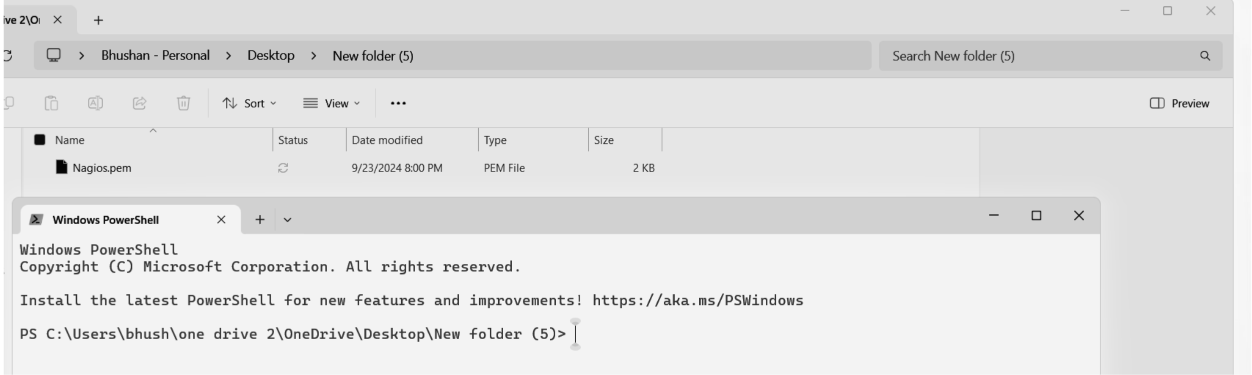
Command copied

```
ssh -i "Nagios.pem" ec2-user@ec2-44-202-48-135.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

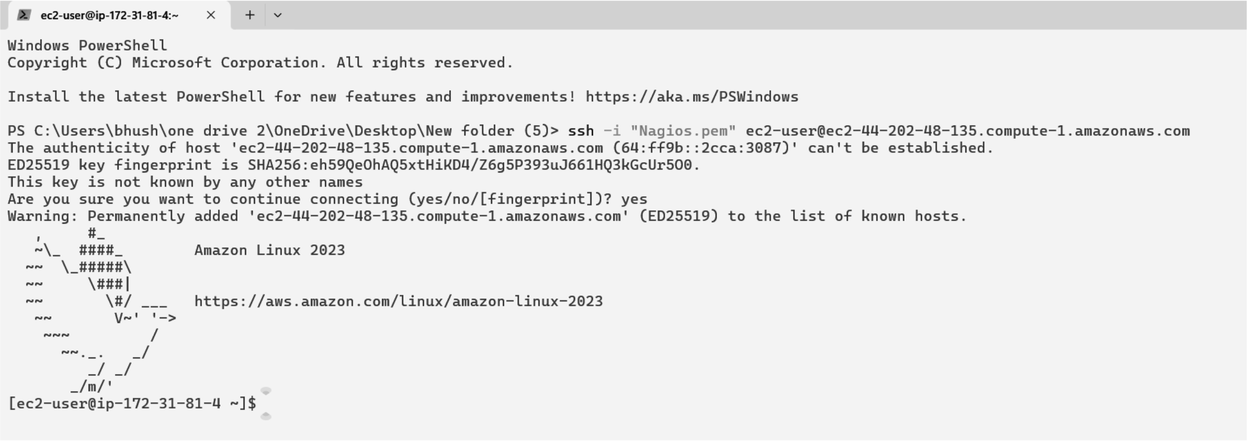
Now open the terminal in the folder where your key(RSA key with .pem) is located.and paste that copied command.



Name	Status	Date modified	Type	Size
Nagios.pem		9/23/2024 8:00 PM	PEM File	2 KB

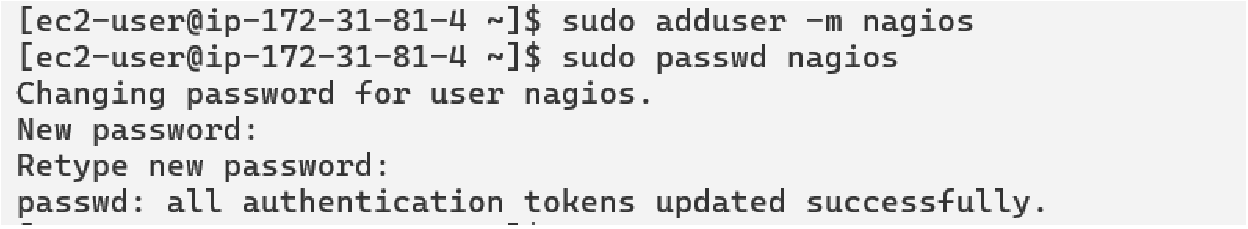
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\bhush\one drive 2\OneDrive\Desktop\New folder (5)>
```

Successfully connected to the instance.



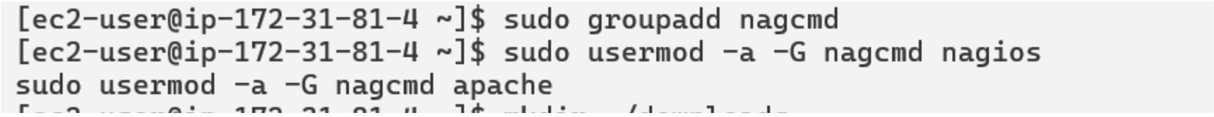
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\bhush\one drive 2\OneDrive\Desktop\New folder (5)> ssh -i "Nagios.pem" ec2-user@ec2-44-202-48-135.compute-1.amazonaws.com
The authenticity of host 'ec2-44-202-48-135.compute-1.amazonaws.com (64:ff9b::2cca:3087)' can't be established.
ED25519 key fingerprint is SHA256:eh59QeOhAQ5xtHiKD4/Z6g5P393uJ661HQ3kGcUr5O0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-202-48-135.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-172-31-81-4 ~]$
```

Step 4: Now Run the following command to make a new user.
sudo adduser -m nagios
sudo passwd nagios



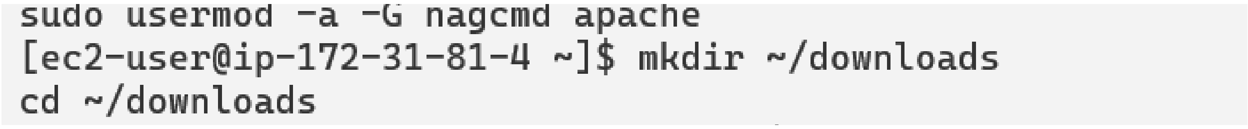
```
[ec2-user@ip-172-31-81-4 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-81-4 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Step 5: Now Run the following command to make a new user group.
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache



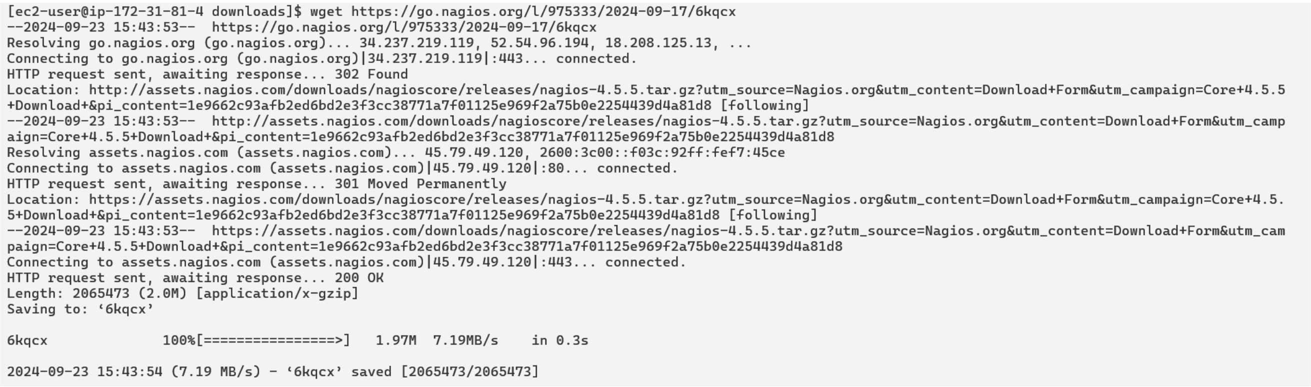
```
[ec2-user@ip-172-31-81-4 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-81-4 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

Step 6: Now make a new directory and go to that directory.
mkdir ~/downloads
cd ~/downloads



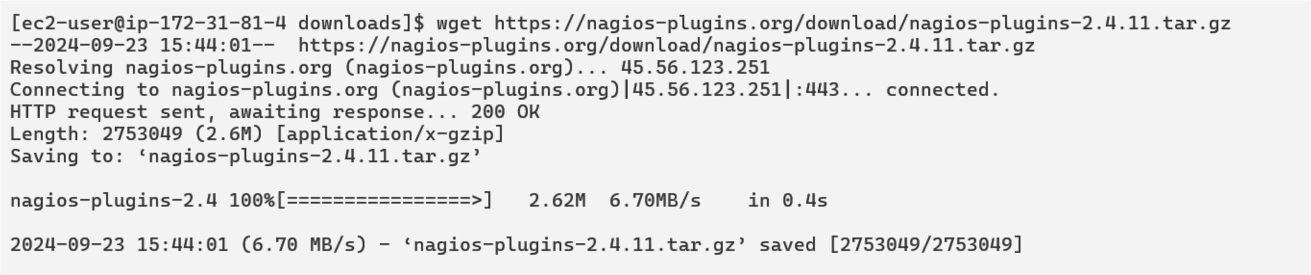
```
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-81-4 ~]$ mkdir ~/downloads
cd ~/downloads
```

Step 7: Now to download the Nagios 4.5.5 and Nagios-plugins 2.4.11 run the following commands respectively.
wget https://go.nagios.org/l/975333/2024-09-17/6kqcx



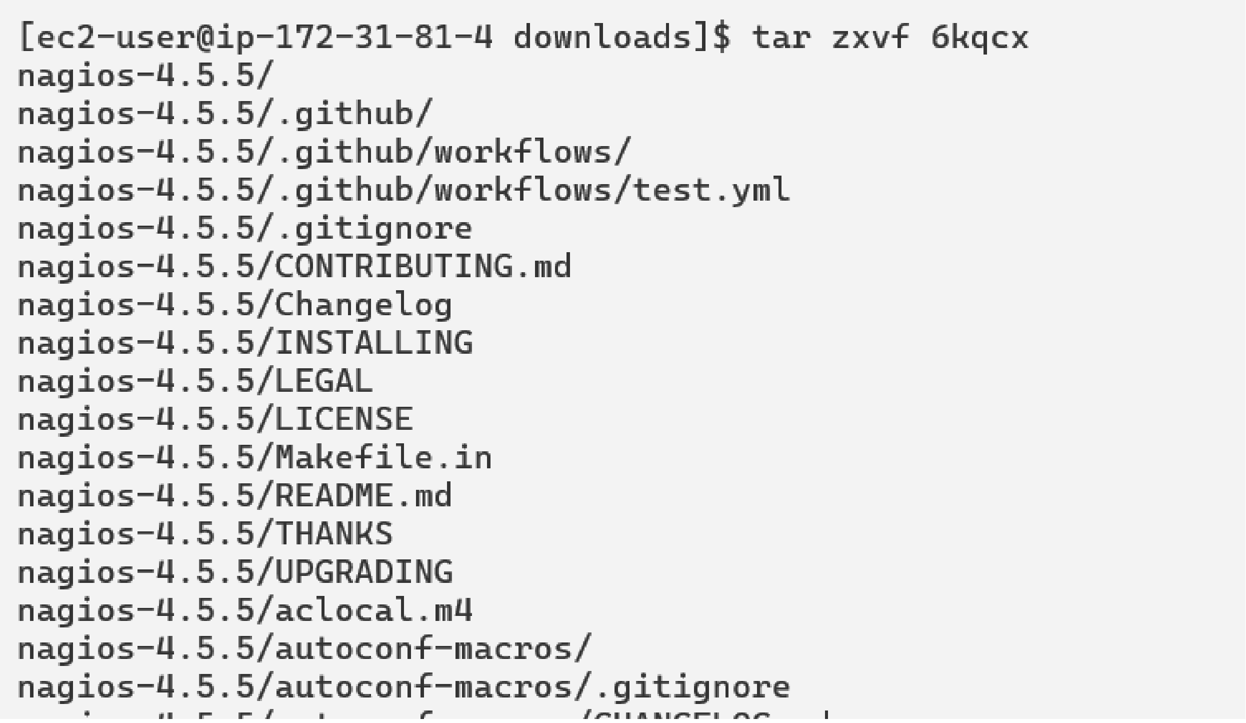
```
[ec2-user@ip-172-31-81-4 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-09-23 15:43:53--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 34.237.219.119, 52.54.96.194, 18.208.125.13, ...
Connecting to go.nagios.org (go.nagios.org)|34.237.219.119|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5
+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-09-23 15:43:53--  http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_camp
aign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
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|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.
5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-09-23 15:43:53--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_cam
paign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
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: ‘6kqcx’

6kqcx              100%[================>]   1.97M  7.19MB/s    in 0.3s

2024-09-23 15:43:54 (7.19 MB/s) - ‘6kqcx’ saved [2065473/2065473]
```

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



```
[ec2-user@ip-172-31-81-4 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-23 15:44:01--  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-2.4 100%[================>]   2.62M  6.70MB/s    in 0.4s

2024-09-23 15:44:01 (6.70 MB/s) - ‘nagios-plugins-2.4.11.tar.gz’ saved [2753049/2753049]
```

Step 8: Now to extract the files from the downloaded Nagios 4.5.5 run the following command.
tar zxvf 6kqcx (Replace 6kqcx with your saved file name of Nagios 4.5.5 refer above screenshot(1st))

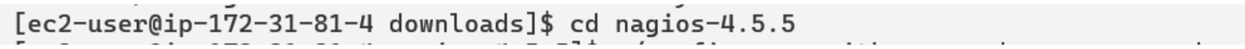


```
[ec2-user@ip-172-31-81-4 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
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
```

Name:Bhushan Mukund Kor **Academic Year:2024-2025**

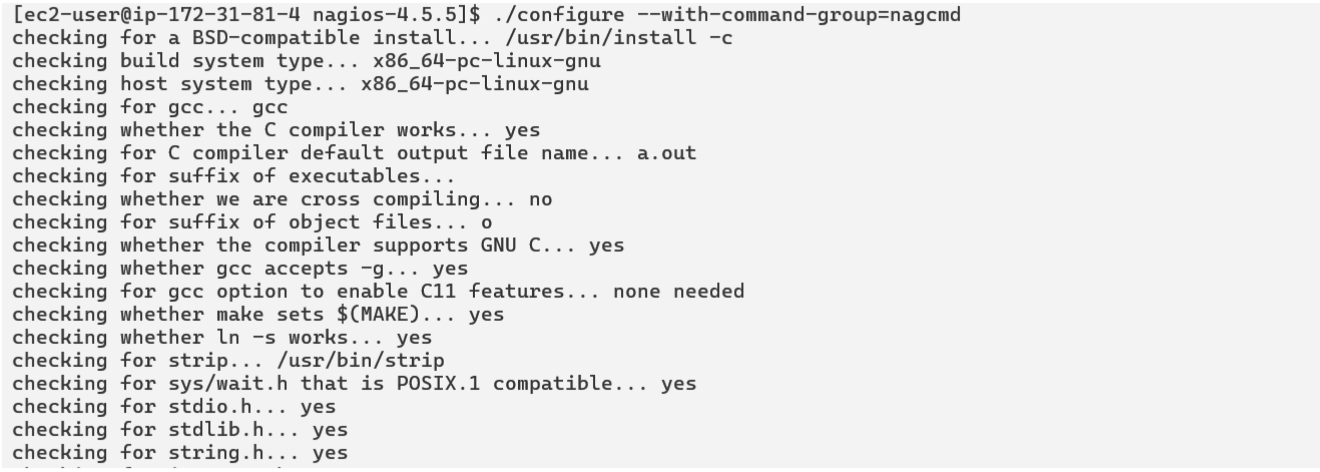
Division: D15C **Roll No: 28**

Step 9: Now change the directory to nagios-4.5.5 (Or which version you have downloaded)



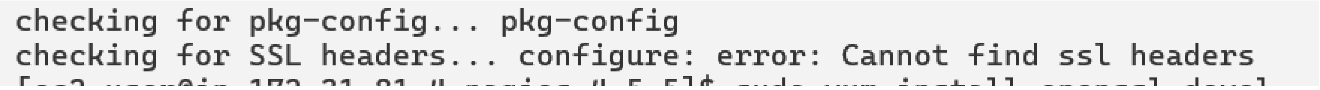
```
[ec2-user@ip-172-31-81-4 downloads]$ cd nagios-4.5.5
```

Step 10: Now run the following command to configure.
./configure --with-command-group=nagcmd



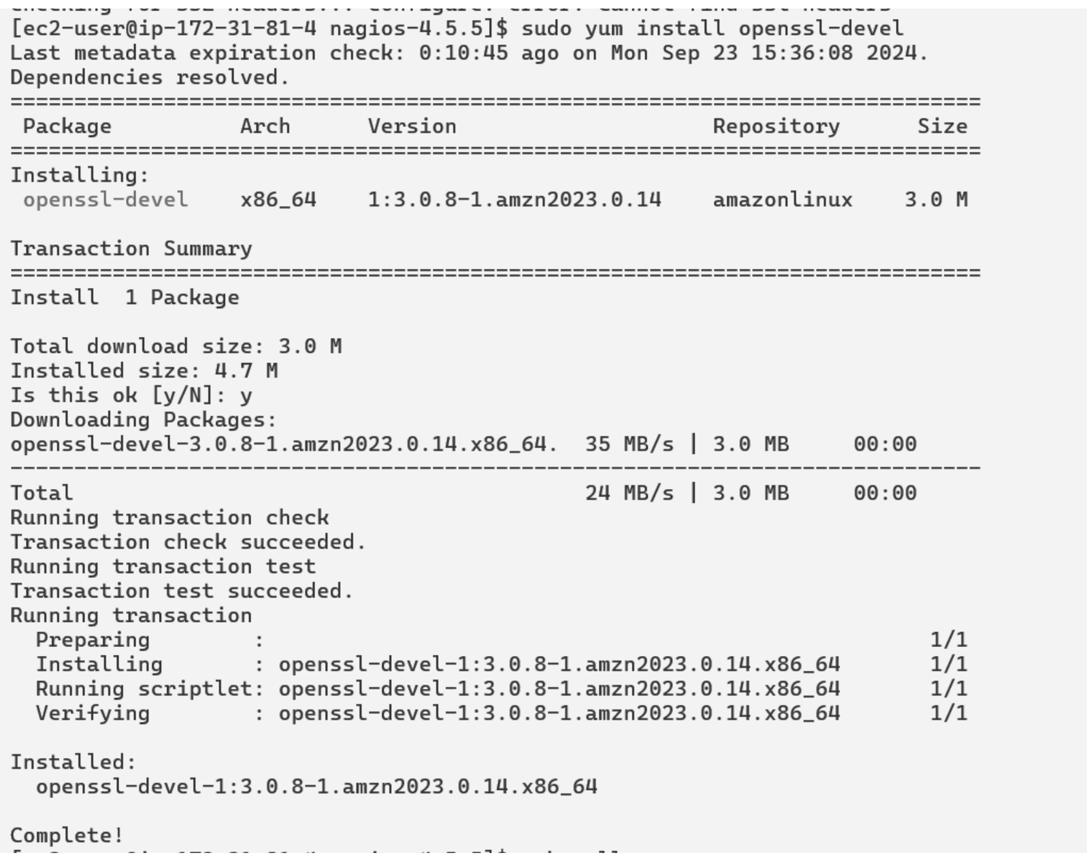
```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... 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
```

At the end we have found the error of cannot find ssl header .



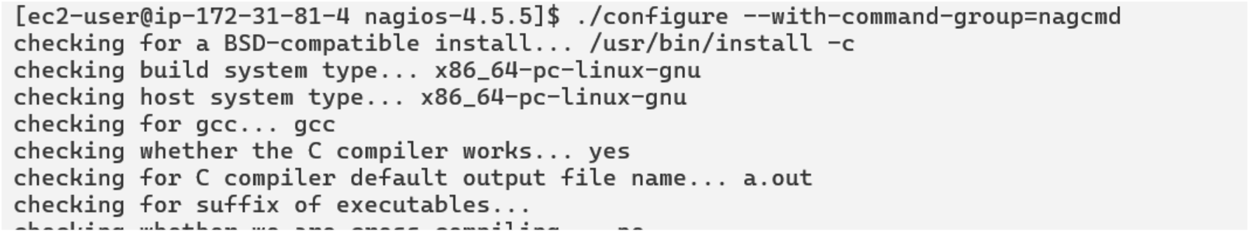
```
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

So run following command to install ssl.
sudo yum install openssl-devel

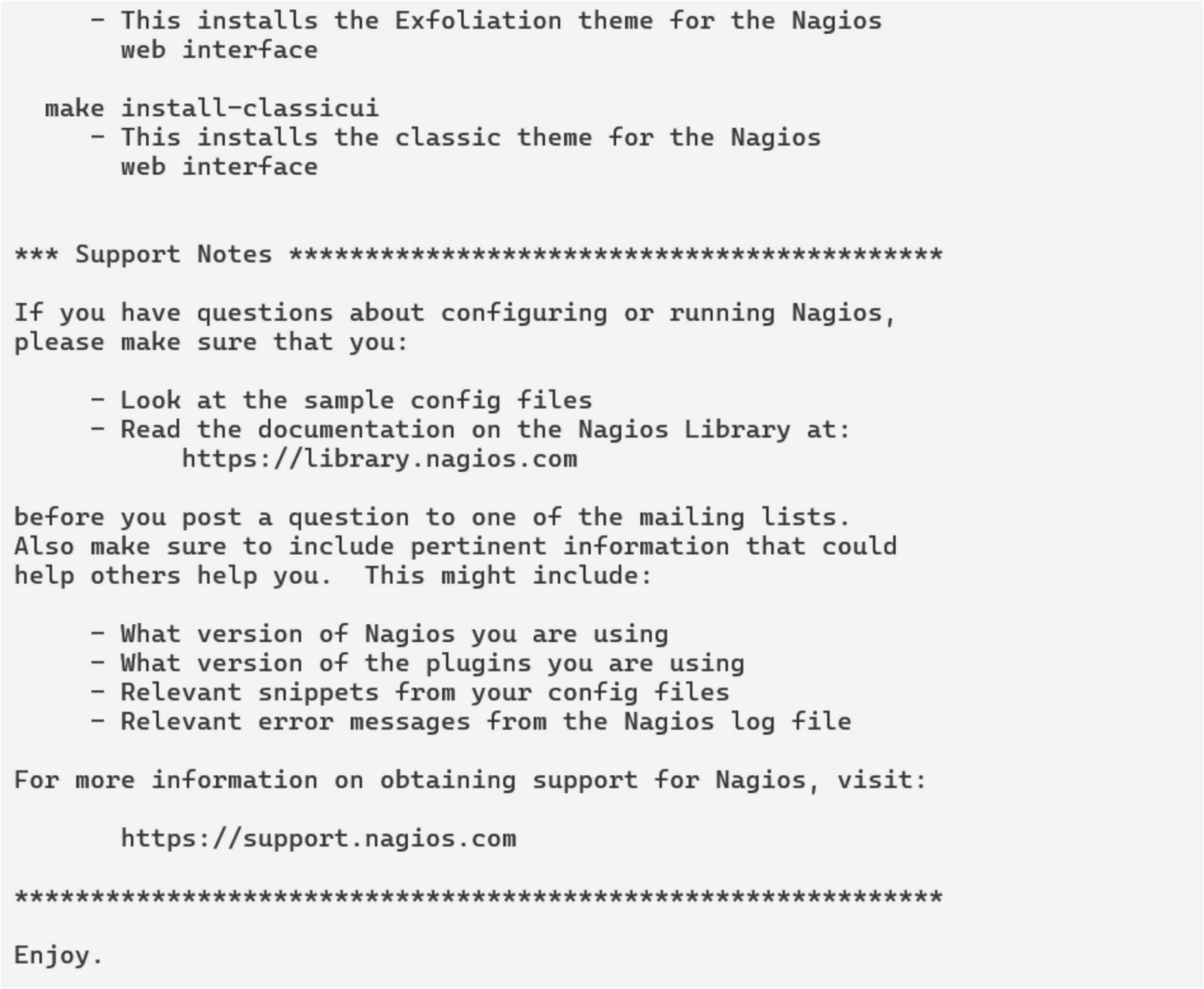


```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:10:45 ago on Mon Sep 23 15:36:08 2024.
Dependencies resolved.
================================================================================
 Package           Arch      Version                     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.  35 MB/s | 3.0 MB     00:00
--------------------------------------------------------------------------------
Total                                        24 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!
```

Now rerun the command **./configure --with-command-group=nagcmd**

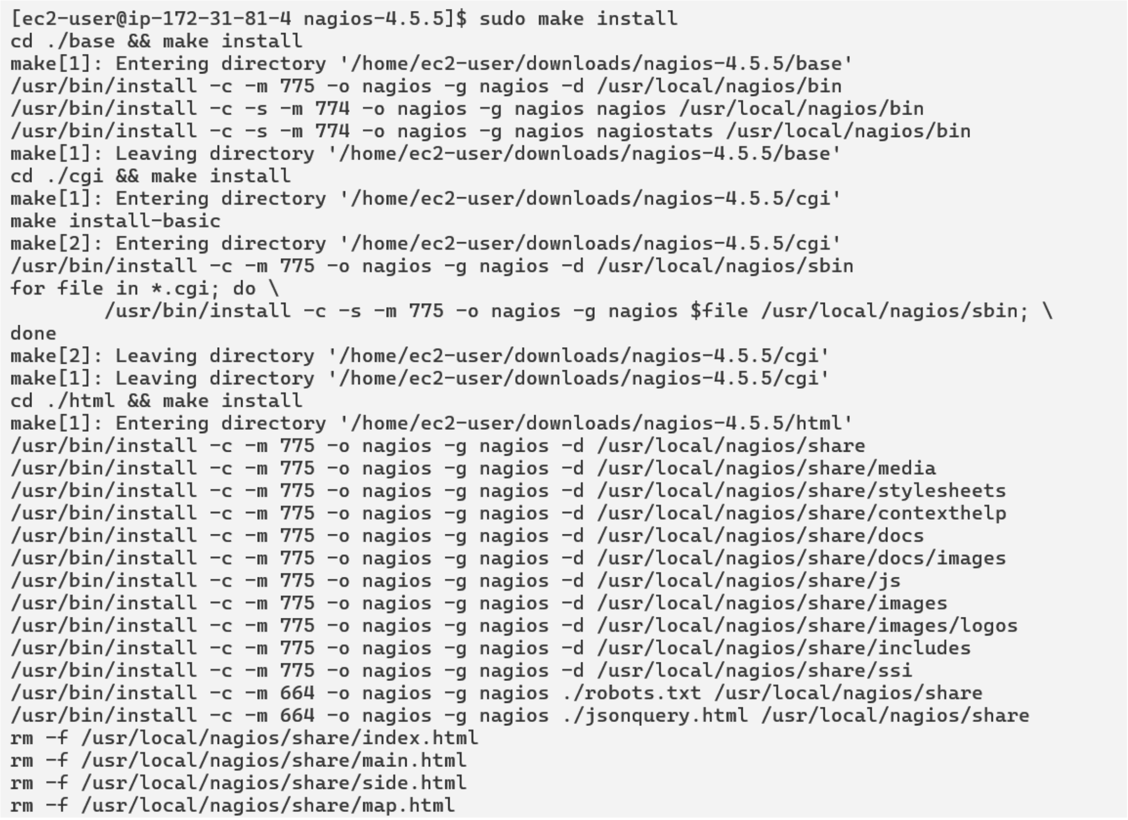


```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
```

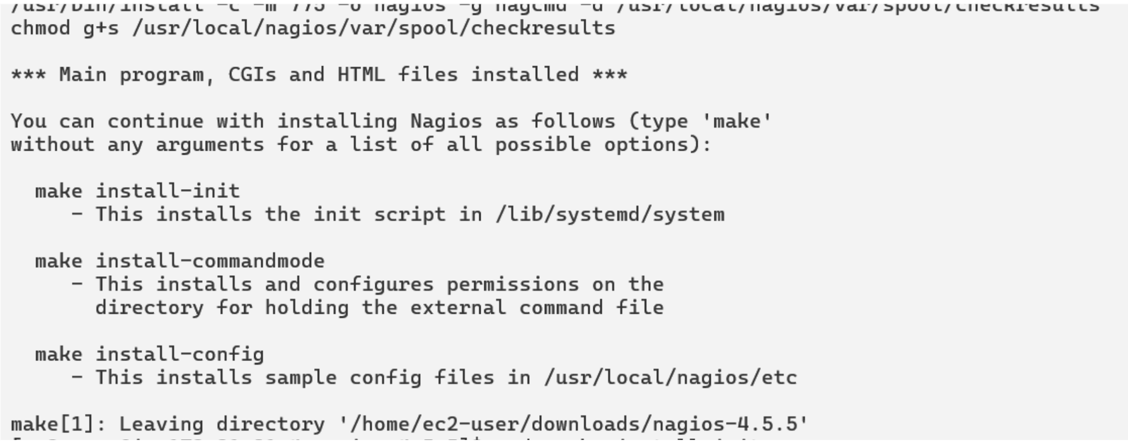


```
     - This installs the Exfoliation theme for the Nagios
       web interface

  make install-classicui
     - This installs the classic theme for the Nagios
       web interface


*** Support Notes *******************************************

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.
```

Step 11: Now run the following commands to steup the Nagios.
sudo make install

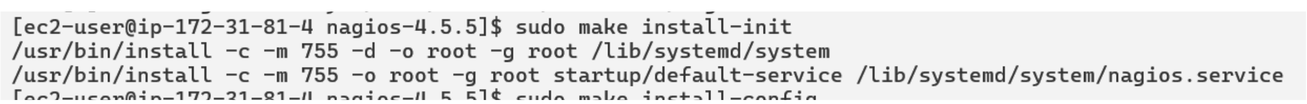


```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios 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 -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 664 -o nagios -g nagios ./robots.txt /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios ./jsonquery.html /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
rm -f /usr/local/nagios/share/map.html
```



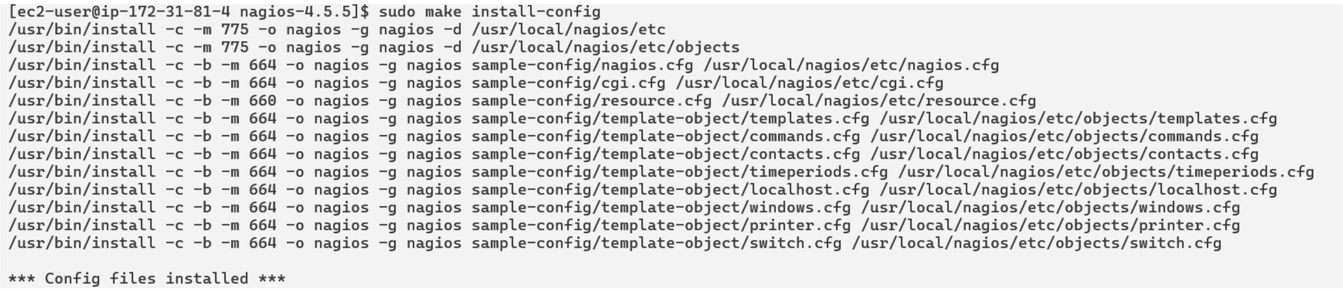
```
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/spool/checkresults
chmod g+s /usr/local/nagios/var/spool/checkresults

*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /lib/systemd/system

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc

make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5'
```

sudo make install-init



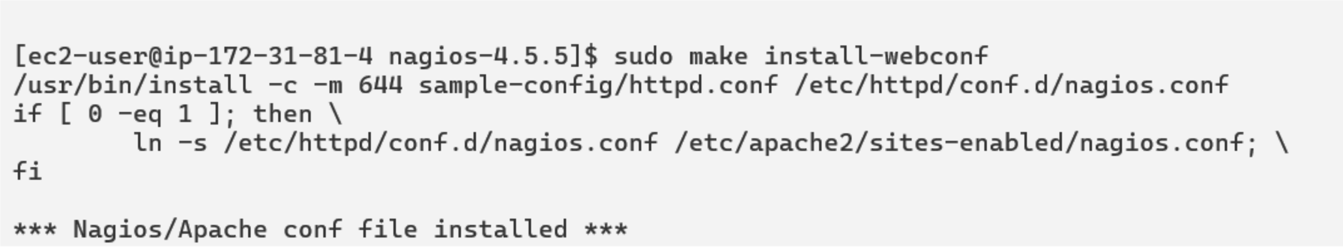
```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo make install-config
```

sudo make install-config



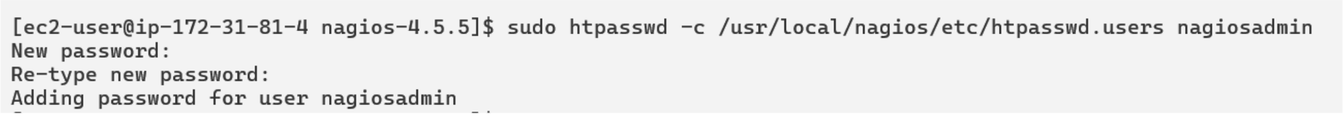
```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/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/resource.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
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/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 ***
```

sudo make install-webconf



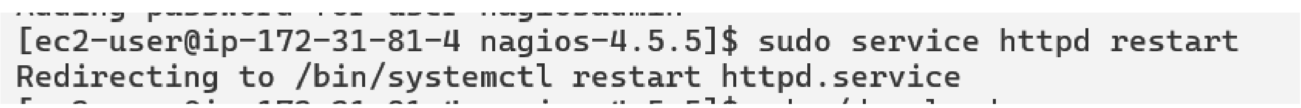
```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin **(To set the password)**



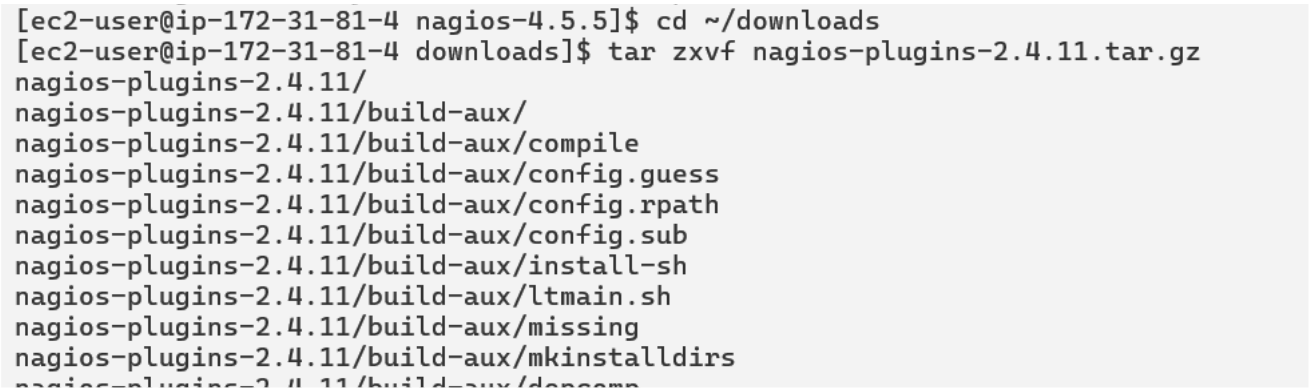
```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Now to restart the httpd service run the following command.
sudo service httpd restart



```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

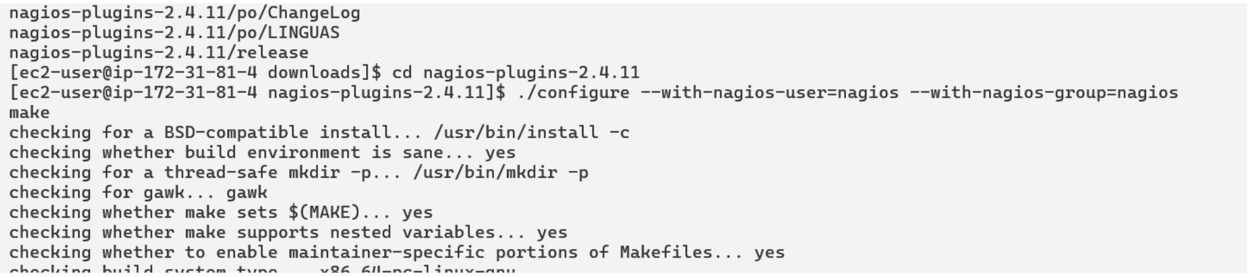
Step 12: Now to extract the files from the downloaded Nagios plugin 2.4.11 run the following command first change the directory.
cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz (According to your version)



```
[ec2-user@ip-172-31-81-4 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-81-4 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
```

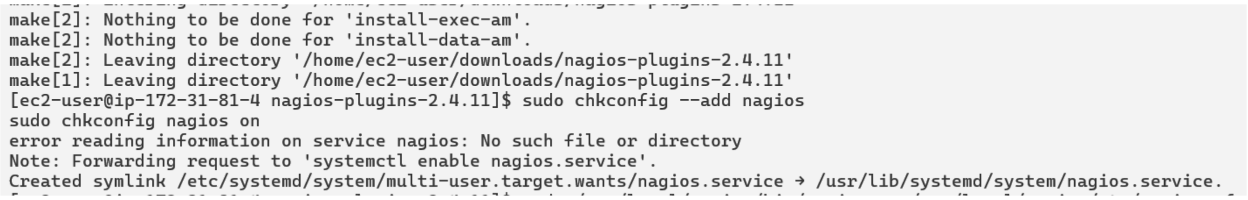
Step 13: Now change the directory to nagios-plugins-2.4.11 and run the config command to configure.
cd nagios-plugins-2.4.11

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



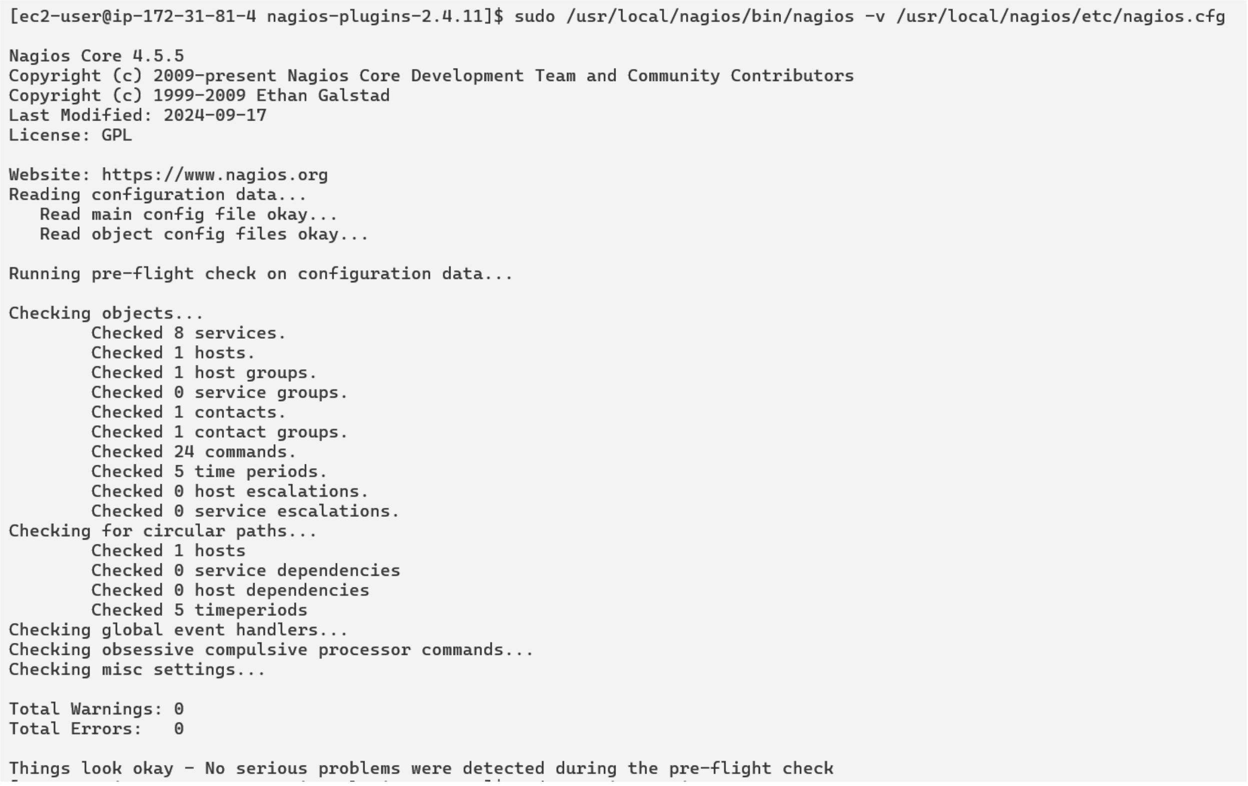
```
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-81-4 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-81-4 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
```

Step 14: Run the following commands to check nagios and start it.
sudo chkconfig --add nagios



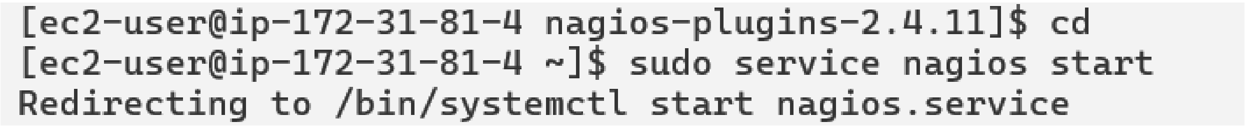
```
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-81-4 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.
```

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



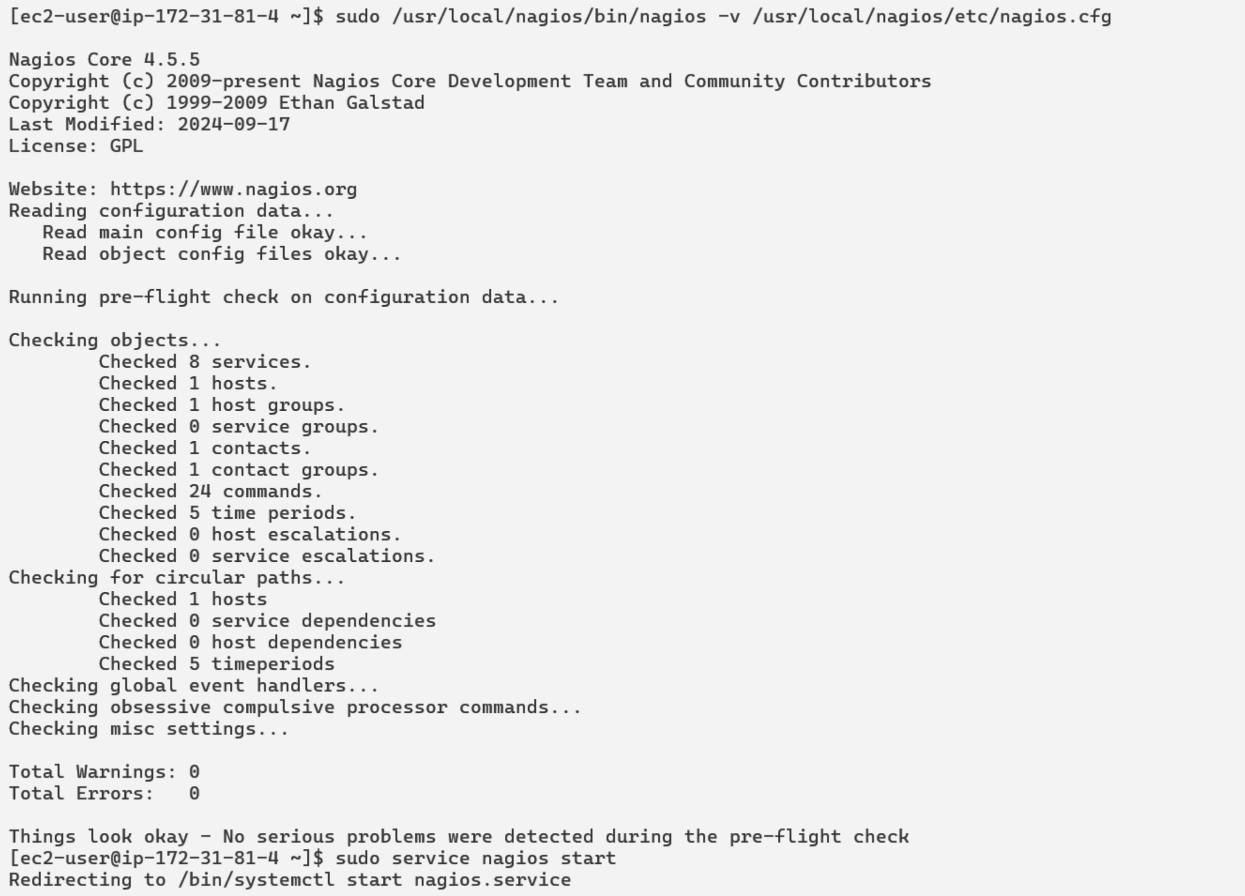
```
[ec2-user@ip-172-31-81-4 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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 host escalations.
        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
```

cd
sudo service nagios start



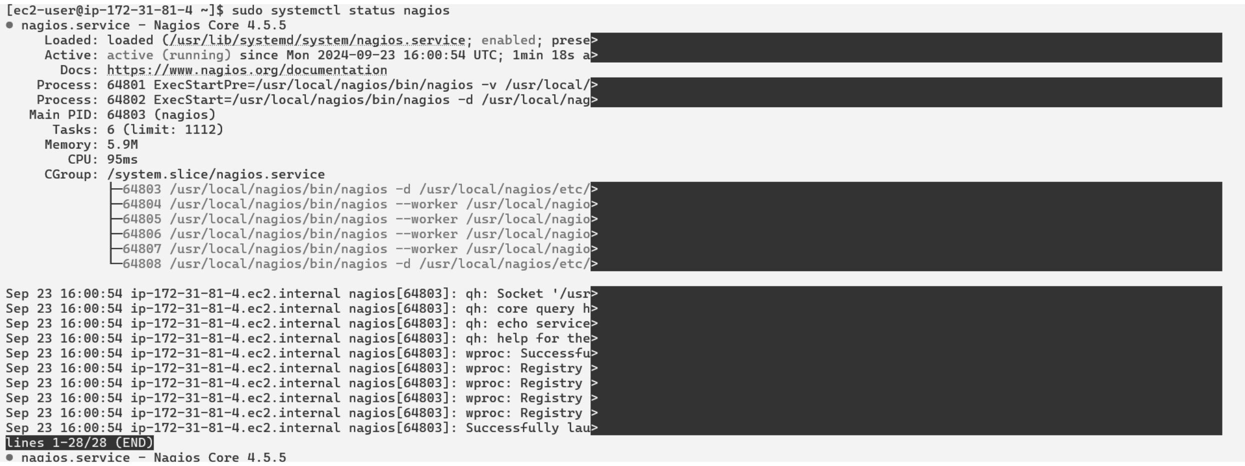
```
[ec2-user@ip-172-31-81-4 nagios-plugins-2.4.11]$ cd
[ec2-user@ip-172-31-81-4 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

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



```
[ec2-user@ip-172-31-81-4 ~]$ 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 host escalations.
        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-172-31-81-4 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

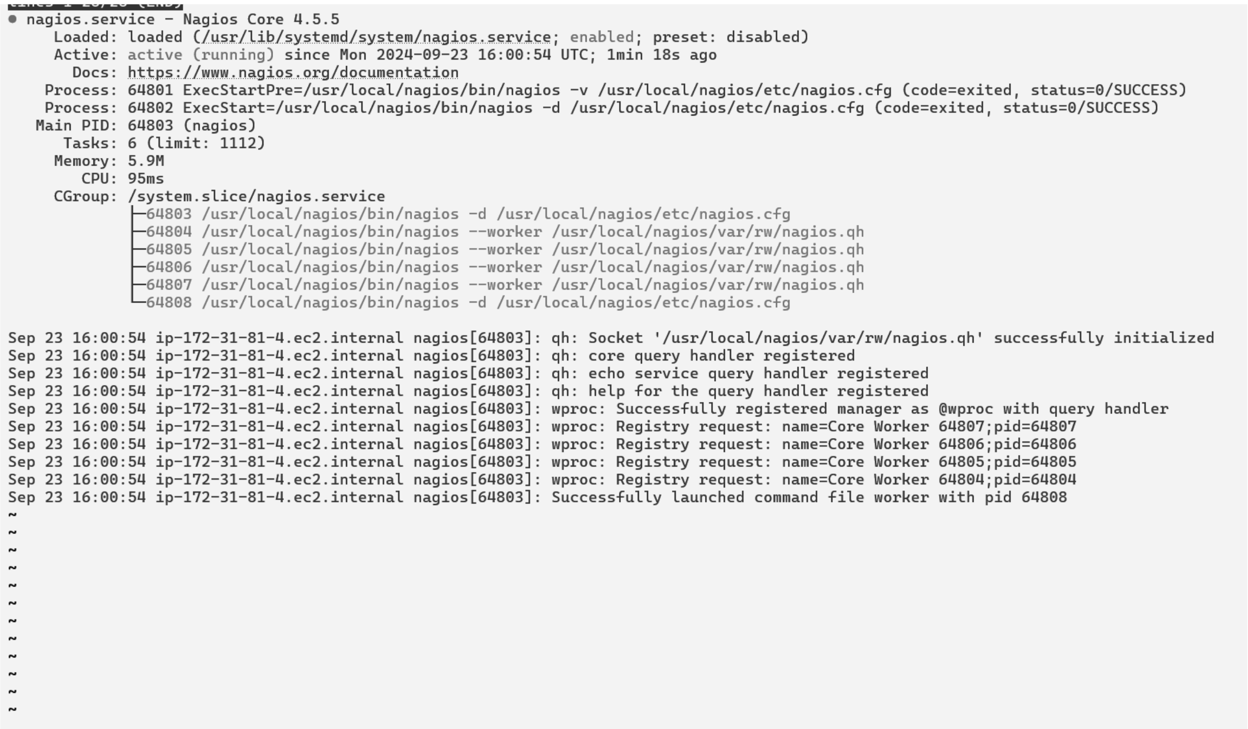
sudo systemctl status nagios



```
[ec2-user@ip-172-31-81-4 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; prese>
     Active: active (running) since Mon 2024-09-23 16:00:54 UTC; 1min 18s a>
       Docs: https://www.nagios.org/documentation
    Process: 64801 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/>
    Process: 64802 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nag>
   Main PID: 64803 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.9M
        CPU: 95ms
     CGroup: /system.slice/nagios.service
             ├─64803 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/>
             ├─64804 /usr/local/nagios/bin/nagios --worker /usr/local/nagio>
             ├─64805 /usr/local/nagios/bin/nagios --worker /usr/local/nagio>
             ├─64806 /usr/local/nagios/bin/nagios --worker /usr/local/nagio>
             ├─64807 /usr/local/nagios/bin/nagios --worker /usr/local/nagio>
             └─64808 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/>

Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: Socket '/usr>
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: core query h>
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: echo service>
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: help for the>
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Successfu>
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry >
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry >
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry >
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry >
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: Successfully lau>
lines 1-28/28 (END)
● nagios.service - Nagios Core 4.5.5
```

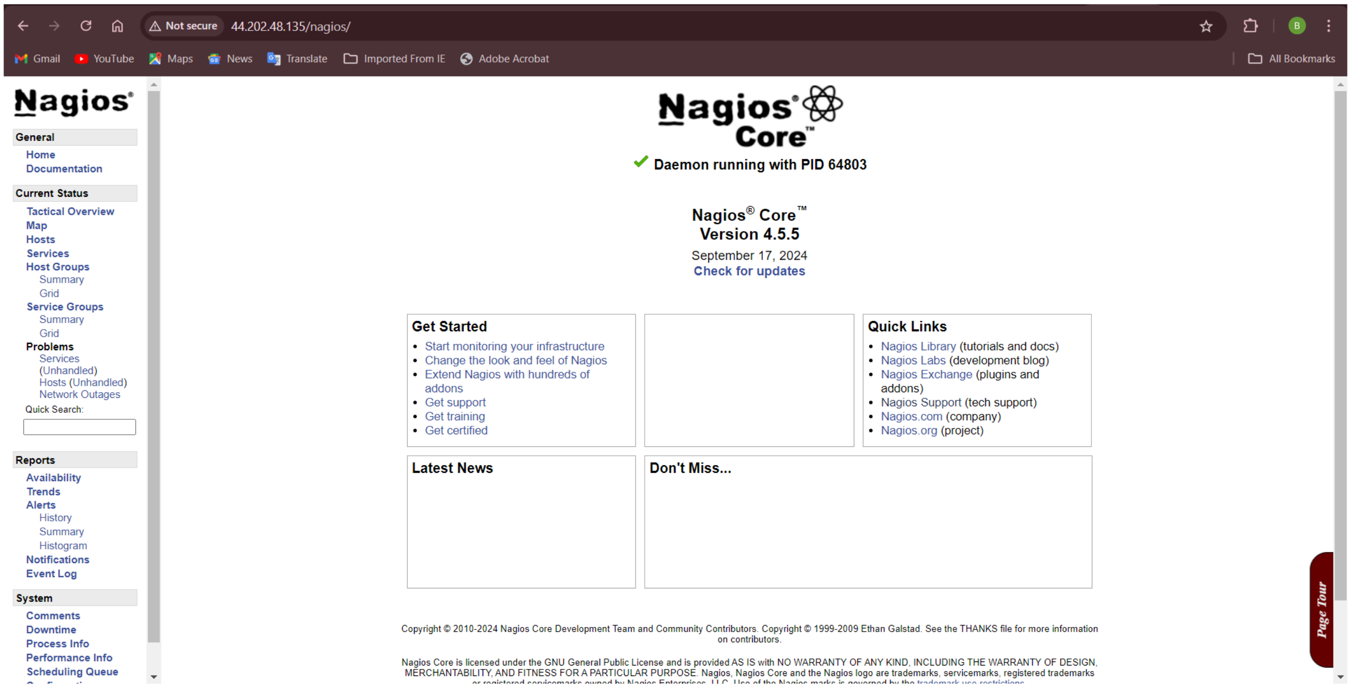
Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**



```
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-09-23 16:00:54 UTC; 1min 18s ago
       Docs: https://www.nagios.org/documentation
    Process: 64801 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 64802 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 64803 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.9M
        CPU: 95ms
     CGroup: /system.slice/nagios.service
             ├─64803 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─64804 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64805 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64806 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─64807 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─64808 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: core query handler registered
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: echo service query handler registered
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: qh: help for the query handler registered
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Successfully registered manager as @wproc with query handler
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64807;pid=64807
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64806;pid=64806
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64805;pid=64805
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: wproc: Registry request: name=Core Worker 64804;pid=64804
Sep 23 16:00:54 ip-172-31-81-4.ec2.internal nagios[64803]: Successfully launched command file worker with pid 64808
~
~
~
~
~
~
~
~
~
~
~
~
```

Step 15: We can see we have successfully launched the Nagios now . Open **http://<instance public ip >/nagios/** here it is http://44.202.48.135/nagios we can see the running web page of nagios.



✓ Daemon running with PID 64803

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...

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

Conclusion: In this experiment, we have setup the Nagios core with plugins on Amazon Linux. Which will help us to understand Continuous monitoring and Installation. It is important to note that whatever set of rules we have added in step 1 are very important for this experiment.