

**Adv DevOps Exp 09**

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- Helps you to plan for infrastructure upgrades before outdated systems create failures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

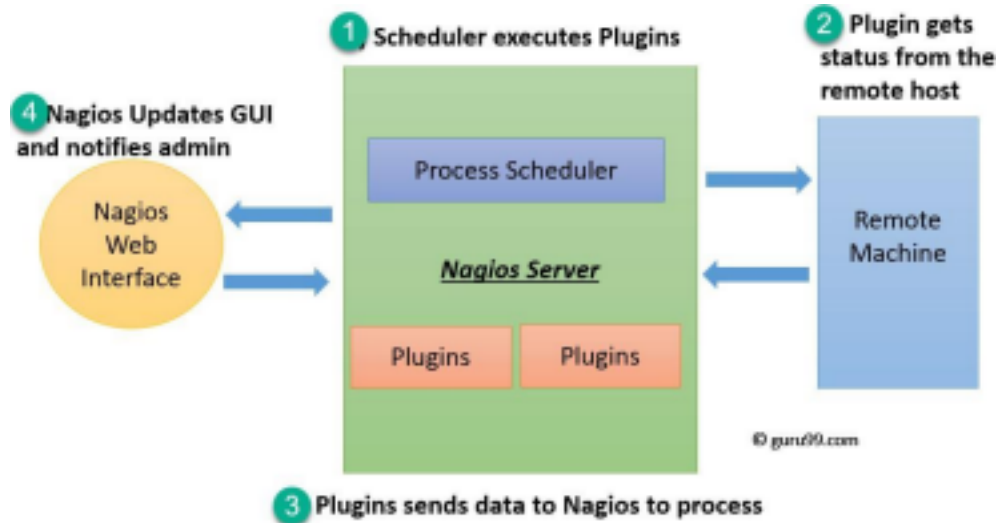
Following are the important features of Nagios monitoring tool:

- 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 alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive

- problem resolution
- Support for implementing redundant monitoring hosts

### Nagios Architecture

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



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

**Step 1: Create a security group with the required configurations**

I have created a new security group with a name 'newsecurity'

[EC2](#) > [Security Groups](#) > Create security group

## Create security group [Info](#)

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

### Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

I have modified the INBOUND RULES as follows

Type	Protocol	Port range	Source	Description - optional	
HTTP	TCP	80	Anywh... ::/0 X		Delete
HTTPS	TCP	443	Anywh... 0.0.0.0/0 X		Delete
SSH	TCP	22	Anywh... 0.0.0.0/0 X		Delete
All ICMP - IPv6	IPv6 ICMP	All	Anywh... ::/0 X		Delete
All ICMP - IPv4	ICMP	All	Anywh... 0.0.0.0/0 X		Delete
All traffic	All	All	Anywh... 0.0.0.0/0 X		Delete
Custom TCP	TCP	5666	Anywh... 0.0.0.0/0 X		Delete

**Step 2: Create ec2 instance**

Name it as nagios-host. Select instance type as amazon-linux and choose the already created key pair and security group

**Name and tags** [Info](#)

Name

nagios-host

Add additional tags

**▼ 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

Number

1

Software

Amazon ami-0ebfc

Virtual s

t2.micro

Firewall

newsecu

Storage

1 volum

Fi

ir

t2

t2

u:

Cancel

**▼ 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

mohit

Create new key pair

**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

Select security groups

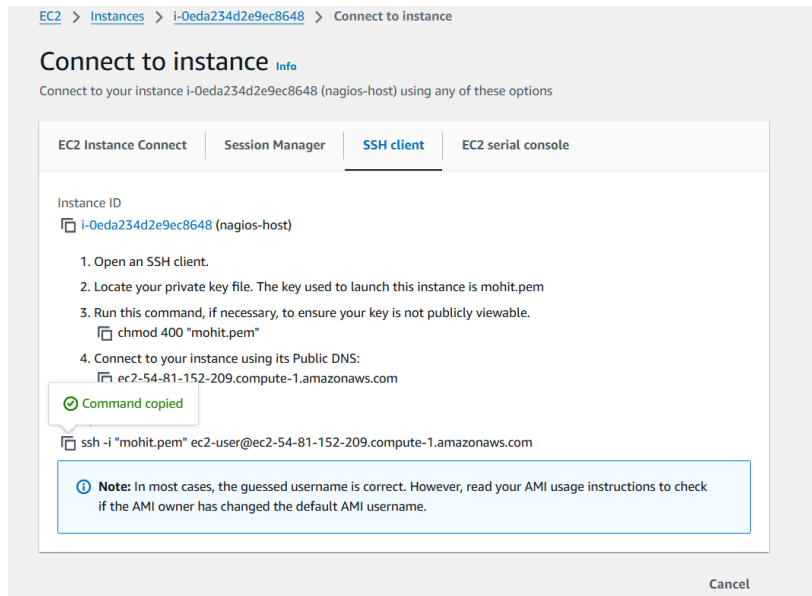
newsecurity sg-05d7468fe3a2f7a8e X  
VPC: vpc-0aa3db8937df8678b

Compare security group rules

1 volum

Fi

Copy the given ssh command, as we will require it for logging into our nagios-host instance from our windows powershell



**Step 3:** Open an administrative powershell and remotely login using the above mentioned ssh command

```

ec2-user@ip-172-31-92-249:~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\WINDOWS\system32> cd C:\Users\Dell\Downloads
PS C:\Users\Dell\Downloads> ssh -i "mohit.pem" ec2-user@ec2-54-81-152-209.compute-1.amazonaws.com

#
#####
~\  #####\
~~~  \#####|
~~~~  \###|
~~~~  \#/  https://aws.amazon.com/linux/amazon-linux-2023
~~~~  V~'  '->
~~~~
~~~~  _*_*_
~~~~  _/  _/
~~~~  _/m/'  _/
~~~~

Last login: Mon Sep 30 09:25:13 2024 from 125.99.93.18

#
#####
~\  #####\
~~~  \#####|
~~~~  \###|
~~~~  \#/  https://aws.amazon.com/linux/amazon-linux-2023
~~~~  V~'  '->
~~~~
~~~~  _*_*_
~~~~  _/  _/
~~~~  _/m/'  _/
~~~~

Last login: Mon Sep 30 09:25:13 2024 from 125.99.93.18
[ec2-user@ip-172-31-92-249 ~]$ sudo yum update
Last metadata expiration check: 0:13:13 ago on Mon Sep 30 09:23:03 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-92-249 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:13:23 ago on Mon Sep 30 09:23:03 2024.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!

```

And then run these commands

**sudo yum update**

**sudo yum install httpd php**

```
ec2-user@ip-172-31-41-160:~$ sudo yum update
Last metadata expiration check: 0:01:37 ago on Wed Oct  2 12:28:33 2024.
Dependencies resolved.
Nothing to do.
complete!
ec2-user@ip-172-31-41-160:~$ sudo yum install httpd php
Last metadata expiration check: 0:01:45 ago on Wed Oct  2 12:28:33 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              14.7 M
httpd-filesystem                       noarch                2.4.62-1.amzn2023      amazonlinux              14 k
httpd-tools                            x86_64                2.4.62-1.amzn2023      amazonlinux              81 k
libmodssl                              x86_64                1.0.9-4.amzn2023.0.2   amazonlinux              315 k
libtool-ltdl                           x86_64                1.0-10-4.amzn2023      amazonlinux              170 k
libxml2                                 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                       x86_64                1.1.24-0-1.amzn2023.0.4 amazonlinux              9.8 k
php8.3-cli                             x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              3.7 M
php8.3-common                          x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              737 k
php8.3-process                         x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              45 k
php8.3-xml                             x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              154 k
Installing weak dependencies:
apr-util-openssl                       x86_64                1.6.3-1.amzn2023.0.1   amazonlinux              17 k
mod_http2                              x86_64                2.0.27-1.amzn2023.0.3  amazonlinux              166 k
mod_lua                                x86_64                2.4.62-1.amzn2023      amazonlinux              61 k
php8.3-fpm                             x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              1.9 M
php8.3-mbstring                        x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              526 k
php8.3-opcache                         x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              370 k
php8.3-pdo                             x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              89 k
php8.3-sodium                           x86_64                8.3.10-1.amzn2023.0.1  amazonlinux              41 k

Transaction Summary
-----
Install 25 Packages
```

**sudo yum install gcc glibc glibc-common**

```
ec2-user@ip-172-31-41-160:~$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:02 ago on Wed Oct  2 12:28:33 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              32 k
libxcrypt-devel                         x86_64                4.4.33-7.amzn2023        amazonlinux              32 k
make                                    x86_64                1:4.3-5.amzn2023.0.2     amazonlinux              534 k

Transaction Summary
-----
Install 13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]: y
Downloading Packages:
(1/13): annobin-docs-10.93-1.amzn2023.0.1.noarch.rpm                                852 kB/s | 92 kB  00:00
(2/13): annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64.rpm                        6.5 MB/s | 887 kB  00:00
(3/13): gc-8.0.4-5.amzn2023.0.2.x86_64.rpm                                         2.3 MB/s | 105 kB  00:00
(4/13): glibc-devel-2.34-52.amzn2023.0.11.x86_64.rpm                             1.1 MB/s | 27 kB  00:00
(5/13): cpp-11.4.1-2.amzn2023.0.2.x86_64.rpm                                       32 MB/s | 10 MB  00:00
(6/13): glibc-headers-x86-2.34-52.amzn2023.0.11.noarch.rpm                       2.9 MB/s | 427 kB  00:00
(7/13): kernel-headers-6.1.109-118.189.amzn2023.x86_64.rpm                       16 MB/s | 1.4 MB  00:00
(8/13): libmpc-1.2.1-2.amzn2023.0.2.x86_64.rpm                                    2.1 MB/s | 62 kB  00:00
(9/13): guile22-2.2.7-2.amzn2023.0.3.x86_64.rpm                                   27 MB/s | 6.4 MB  00:00
(10/13): libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64.rpm                             322 kB/s | 38 kB  00:00
(11/13): libxcrypt-devel-4.4.33-7.amzn2023.x86_64.rpm                             1.4 MB/s | 32 kB  00:00
```

**sudo yum install gd gd-devel**

```

ec2-user@ip-172-31-41-160:~$ sudo yum install gd gd-devel
Last metadata expiration check: 0:02:25 ago on Wed Oct  2 12:28:33 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            1.17.8-2.amzn2023.0.1  amazonlinux           604 k
cmake-filesystem                    x86_64            3.22.2-1.amzn2023.0.4  amazonlinux           16 k
fontconfig                          x86_64            2.13.04-2.amzn2023.0.2  amazonlinux           273 k
fontconfig-devel                   x86_64            2.13.04-2.amzn2023.0.2  amazonlinux           128 k
fonttts                             x86_64            1:2.0.5-12.amzn2023.0.2  amazonlinux           9.5 k
freetype                            x86_64            2.13.2-5.amzn2023.0.1  amazonlinux           423 k
freetype-devel                     x86_64            2.13.2-5.amzn2023.0.1  amazonlinux           912 k
glib2-devel                         x86_64            2.74.7-689.amzn2023.0.2  amazonlinux           486 k
google-noto-fonts-common            noarch            20201206-2.amzn2023.0.2  amazonlinux           15 k
google-noto-sans-vf-fonts           noarch            20201206-2.amzn2023.0.2  amazonlinux           492 k
graphite2                           x86_64            1.3.14-7.amzn2023.0.2  amazonlinux           97 k
graphite2-devel                    x86_64            1.3.14-7.amzn2023.0.2  amazonlinux           21 k
harfbuzz                           x86_64            7.0.0-2.amzn2023.0.1  amazonlinux           868 k
harfbuzz-devel                     x86_64            7.0.0-2.amzn2023.0.1  amazonlinux           484 k
harfbuzz-icu                       x86_64            7.0.0-2.amzn2023.0.1  amazonlinux           18 k
libgk11-libs                       x86_64            2.1-21.amzn2023.0.2  amazonlinux           54 k
langpacks-core-font-en             noarch            3.0-21.amzn2023.0.4  amazonlinux           10 k
libICE                              x86_64            1.0.10-6.amzn2023.0.2  amazonlinux           71 k
libSM                               x86_64            1.2.3-8.amzn2023.0.2  amazonlinux           42 k
libX11                              x86_64            1.7.2-3.amzn2023.0.4  amazonlinux           657 k
libX11-common                      noarch            1.7.2-3.amzn2023.0.4  amazonlinux           152 k
libX11-devel                       x86_64            1.7.2-3.amzn2023.0.4  amazonlinux           939 k
libX11-xcb                         x86_64            1.7.2-3.amzn2023.0.4  amazonlinux           12 k
libXau                             x86_64            1.0.9-6.amzn2023.0.2  amazonlinux           31 k
libXau-devel                       x86_64            1.0.9-6.amzn2023.0.2  amazonlinux           14 k
libXext                             x86_64            1.3.4-6.amzn2023.0.2  amazonlinux           41 k
libXpm                              x86_64            3.5.15-2.amzn2023.0.3  amazonlinux           65 k
libXpm-devel                      x86_64            3.5.15-2.amzn2023.0.3  amazonlinux           59 k
libXrender                         x86_64            0.9.10-14.amzn2023.0.2  amazonlinux           28 k
libxt                               x86_64            1.2.0-4.amzn2023.0.2  amazonlinux           181 k
libXt-devel                        x86_64            1.2.0-4.amzn2023.0.2  amazonlinux           15 k

```

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

**sudo adduser -m nagios**

**sudo passwd nagios**

```

[ec2-user@ip-172-31-41-160 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-41-160 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-41-160 ~]$

```

Create a new user group & create a new directory for Nagios downloads using the following commands

**sudo groupadd nagcmd**

**sudo usermod -a -G nagcmd nagios**

**sudo usermod -a -G nagcmd apache**

**mkdir ~/downloads**

**cd ~/downloads**

Use **wget** to download the source zip files.

In this step, we are downloading, the latest version of nagios and the necessary plugins required to carry out the tasks of setting up a nagios server

wget <https://sourceforge.net/projects/nagios/files/latest/download>

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 downloads]$ wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-10-02 12:34:21-- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)[172.64.150.145]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-T3MNSZPzP-6la2TltvoGCG7vV7QGVH08n3tC24QehfMw7vHcokBGHg2iIRxbmfugII0LccNfXa0ixg3jZkG3w3D3D&use_mirror=phoenixnap&= [following]
--2024-10-02 12:34:21-- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-T3MNSZPzP-6la2TltvoGCG7vV7QGVH08n3tC24QehfMw7vHcokBGHg2iIRxbmfugII0LccNfXa0ixg3jZkG3w3D3D&use_mirror=phoenixnap&= [following]
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)[204.68.111.105]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://phoenixnap.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viasf=1 [following]
--2024-10-02 12:34:21-- https://phoenixnap.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viasf=1
Resolving phoenixnap.dl.sourceforge.net (phoenixnap.dl.sourceforge.net)... 184.164.141.26
Connecting to phoenixnap.dl.sourceforge.net (phoenixnap.dl.sourceforge.net)[184.164.141.26]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

download                               100%[=====>] 1.97M 4.23MB/s in 0.5s

2024-10-02 12:34:22 (4.23 MB/s) - 'download' saved [2065473/2065473]
```

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

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-02 12:34:46-- 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 7.48MB/s in 0.4s

2024-10-02 12:34:46 (7.48 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

```
ec2-user@ip-172-31-92-249:~/downloads
[ec2-user@ip-172-31-92-249 ~]$ cd ~/downloads
[ec2-user@ip-172-31-92-249 downloads]$ wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-09-30 09:54:56-- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)[172.64.150.145]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RDvAvnMaShLF3gu4RXTSVxrTZ6F6xJVhVA0zp81bPgbyzLMcDDAALgtEC1P0Kr0cgJN23bWktariCjQTVrkG3D3D&use_mirror=netactuate&= [following]
--2024-09-30 09:54:56-- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RDvAvnMaShLF3gu4RXTSVxrTZ6F6xJVhVA0zp81bPgbyzLMcDDAALgtEC1P0Kr0cgJN23bWktariCjQTVrkG3D3D&use_mirror=netactuate&= [following]
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)[204.68.111.105]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viasf=1 [following]
--2024-09-30 09:54:57-- https://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?viasf=1
Resolving netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)... 104.225.3.66
Connecting to netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)[104.225.3.66]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

download                               100%[=====>] 1.97M ---KB/s in 0.07s

2024-09-30 09:54:57 (29.8 MB/s) - 'download' saved [2065473/2065473]

[ec2-user@ip-172-31-92-249 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
--2024-09-30 09:56:53-- https://nagios-plugins.org/download/nagios-plugins-2.4.9.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: 2754403 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.9.tar.gz'

nagios-plugins-2.4.9.tar.gz           100%[=====>] 2.63M 7.54MB/s in 0.3s

2024-09-30 09:56:54 (7.54 MB/s) - 'nagios-plugins-2.4.9.tar.gz' saved [2754403/2754403]
```

Now, we run the next command in the following manner

tar xzvf <nagios-4.5.5 version> (for me it has gotten saved as 'download')

So i wrote **tar xzvf download**



```
ec2-user@ip-172-31-41-160:~/downloads
[ec2-user@ip-172-31-41-160 downloads]$ tar zxvf download
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/ChangeLog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LLEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
nagios-4.5.5/autoconf-macros/ax_nagios_get_inetd
nagios-4.5.5/autoconf-macros/ax_nagios_get_init
nagios-4.5.5/autoconf-macros/ax_nagios_get_os
nagios-4.5.5/autoconf-macros/ax_nagios_get_paths
nagios-4.5.5/autoconf-macros/ax_nagios_get_ssl
nagios-4.5.5/base/
nagios-4.5.5/base/.gitignore
nagios-4.5.5/base/Makefile.in
nagios-4.5.5/base/broker.c
nagios-4.5.5/base/checks.c
nagios-4.5.5/base/commands.c
nagios-4.5.5/base/config.c
nagios-4.5.5/base/events.c
nagios-4.5.5/base/flapping.c
nagios-4.5.5/base/logging.c
nagios-4.5.5/base/nagios.c
nagios-4.5.5/base/nagiosstats.c
nagios-4.5.5/base/nebmods.c
nagios-4.5.5/base/nerd.c
```

After which we are supposed to **change our directory** over there

For eg. **cd nagios-4.5.5...** depending on the version that we have downloaded

Next, Run this command (make sure that you are working inside nagios-4.x.x directory)

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

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-41-160 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
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking forunistd.h... yes
checking forarpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking forfcntl.h... yes
checking for getopt.h... yes
checking forgrp.h... yes
checking forlibgen.h... yes
checking forlimits.h... yes
checking formath.h... yes
checking fornetdb.h... yes
checking fornetinet/in.h... yes
checking forpwd.h... yes
checking forregex.h... yes
checking forsignal.h... yes
checking forsocket.h... no
checking forstdarg.h... yes
```

```

checking for strcasecmp... yes
checking for strtoul... 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
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$

```

After running this command, we get an **error related to ssl header being absent**

For that purpose, we are to run the following command.

**sudo yum install openssl-devel** (for ssl header)

```

[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:12:11 ago on Wed Oct  2 12:28:33 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                                                                    17 MB/s | 3.0 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  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!
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$

```

Now, Re-run **./configure --with-command-group=nagcmd**

After this, run **make all** command

```

ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 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 -DMSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o broker.o broker.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o netmods.o netmods.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -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(DEBBUG_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 -DMSCORE -c -o checks.o checks.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o config.o config.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o commands.o commands.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o events.o events.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o flapping.o flapping.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o logging.o logging.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o macros-base.o ../common/macros.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o netutils.o netutils.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o notifications.o notifications.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o sehndlers.o sehndlers.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o utils.o utils.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o retention-base.o ../xdata/xrdddefault.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o comments-base.o ../common/comments.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o xcomments-base.o ../xdata/xcddefault.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o objects-base.o ../common/objects.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o xobjects-base.o ../xdata/xodtemplate.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o statusdata-base.o ../common/statusdata.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o xstatusdata-base.o ../xdata/xsdddefault.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o perfdata-base.o ../perfdata.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o xperfdata-base.o ../xdata/xpdddefault.c
gcc -Wall -I. -I. -I./lib -I./include -I./include -I. -g -O2 -DHAVE_CONFIG_H -DMSCORE -c -o downtime-base.o ../common/downtime.c
make -C ./lib
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/lib'
gcc -Wall -g -O2 -I. -I./include -DHAVE_CONFIG_H -c squeeze.c -o squeeze.o
gcc -Wall -g -O2 -I. -I./include -DHAVE_CONFIG_H -c kvvec.c -o kvvec.o
gcc -Wall -g -O2 -I. -I./include -DHAVE_CONFIG_H -c iocache.c -o iocache.o
gcc -Wall -g -O2 -I. -I./include -DHAVE_CONFIG_H -c iobroker.c -o iobroker.o
gcc -Wall -g -O2 -I. -I./include -DHAVE_CONFIG_H -c bitmap.c -o bitmap.o
gcc -Wall -g -O2 -I. -I./include -DHAVE_CONFIG_H -c dhash.c -o dhash.o

```

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
on doing this. Pay particular attention to the docs on
object configuration files, as they determine what/how
things get monitored!

make install-webconf
- This installs the Apache config file for the Nagios
  web interface

make install-exfoliation
- 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.
```

Run the following set of commands to ensure that  
**sudo make install**

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
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 ./jquery.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
rm -f /usr/local/nagios/share/rss-*
rm -f /usr/local/nagios/share/graph-header.html
rm -f /usr/local/nagios/share/histogram.html
rm -f /usr/local/nagios/share/histogram-form.html
rm -f /usr/local/nagios/share/histogram-graph.html
rm -f /usr/local/nagios/share/histogram-links.html
rm -f /usr/local/nagios/share/infobox.html
rm -f /usr/local/nagios/share/map.php
```

**sudo make install-init**

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 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
```

**sudo make install-config**

```
[ec2-user@ip-172-31-41-160 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 ***

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.
```

**sudo make install-webconf**

```
[ec2-user@ip-172-31-41-160 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 ***

[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

Next, we are supposed to create a nagiosadmin account for nagios login along with password. Specify the password twice.

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

```
[ec2-user@ip-172-31-41-160 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
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

Restart Apache

**sudo service httpd restart**

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

**cd ~/downloads**

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

```
ec2-user@ip-172-31-41-160:~/downloads
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-41-160 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
```

Compile and install plugins

**cd nagios-plugins-2.4.11**

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

Run the following command:

**sudo chkconfig --add nagios**

On running the above command

```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
```

If this is the output that one is getting, then it means that the init script is missing...

We can check this by running `ls /etc/init.d/`

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.9]$ ls /etc/init.d/
README  functions
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.9]$
```

With `ls` command, we must see a file named `nagios`, which i was not able to see

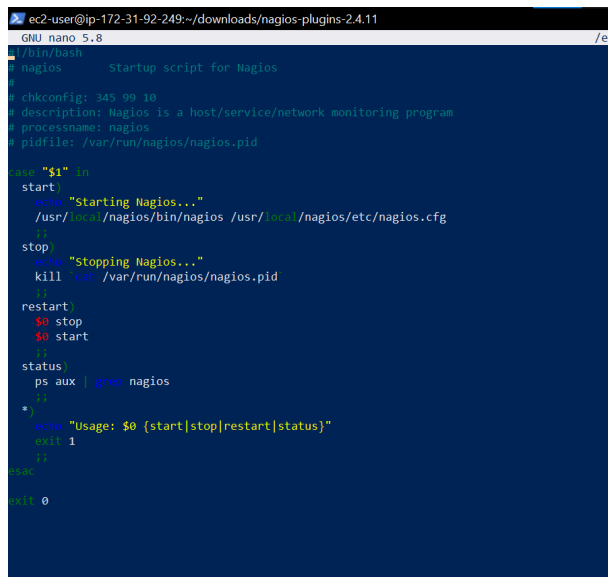
If the Init Script is Missing i.e If you don't see the nagios script in `/etc/init.d/`, you can create it manually. Here's how:

Run the following command:

**sudo nano /etc/init.d/nagios**

Within this file, paste the following script

```
#!/bin/bash
# nagios      Startup script for Nagios
#
# chkconfig: 345 99 10
# description: Nagios is a host/service/network monitoring program
# processname: nagios
# pidfile: /var/run/nagios/nagios.pid
case "$1" in
    start)
        echo "Starting Nagios..."
        /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
        ;;
    stop)
        echo "Stopping Nagios..."
        kill `cat /var/run/nagios/nagios.pid`
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    status)
        ps aux | grep nagios
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac
exit 0
```

A screenshot of a terminal window with a dark blue background. The title bar shows 'ec2-user@ip-172-31-92-249:~/downloads/nagios-plugins-2.4.11'. The terminal content shows the same script as the previous block, with some lines highlighted in red (e.g., 'start', 'stop', 'restart', '\$0 stop', '\$0 start'). The cursor is at the end of the last line 'exit 0'.

```
ec2-user@ip-172-31-92-249:~/downloads/nagios-plugins-2.4.11
GNU nano 5.8 /etc
#!/bin/bash
# nagios      Startup script for Nagios
#
# chkconfig: 345 99 10
# description: Nagios is a host/service/network monitoring program
# processname: nagios
# pidfile: /var/run/nagios/nagios.pid
case "$1" in
    start)
        echo "Starting Nagios..."
        /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
        ;;
    stop)
        echo "Stopping Nagios..."
        kill `cat /var/run/nagios/nagios.pid`
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    status)
        ps aux | grep nagios
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac
exit 0
```

Make the Script Executable: After saving the file, run the following command to make it executable:

**sudo chmod +x /etc/init.d/nagios**

Run **sudo chkconfig --add nagios** again

And then run **sudo chkconfig nagios on**

```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo nano /etc/init.d/nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chmod +x /etc/init.d/nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Synchronizing state of nagios.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$
```

**sudo service nagios start**

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.11]$ sudo service nagios start
Starting Nagios...

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
Nagios 4.5.5 starting... (PID=72261)
Local time is Tue Oct 01 20:59:58 UTC 2024
wproc: Successfully registered manager as @wproc with query handler
wproc: Registry request: name=Core Worker 72265;pid=72265
wproc: Registry request: name=Core Worker 72264;pid=72264
wproc: Registry request: name=Core Worker 72263;pid=72263
wproc: Registry request: name=Core Worker 72262;pid=72262
Successfully launched command file worker with pid 72266
wproc: NOTIFY job 4 from worker Core Worker 72262 is a non-check helper but exited with return code 127
wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
```

Get your public IPv4 address from your instance. We will require it for connecting to our nginx server

The screenshot displays the AWS Management Console interface for an EC2 instance named 'nagios-host' with ID 'i-0eda234d2e9ec8648'. The instance is in a 'Running' state. A tooltip indicates that the public IPv4 address '44.203.161.215' has been copied. The instance details include:

- Instance ID:** i-0eda234d2e9ec8648 (nagios-host)
- IPv6 address:** -
- Hostname type:** IP name: ip-172-31-92-249.ec2.internal
- Answer private resource DNS name:** IPv4 (A)
- Auto-assigned IP address:** -
- Instance state:** Running
- Private DNS name (IPv4 only):** ip-172-31-92-249.ec2.internal
- Instance type:** t2.micro
- VPC ID:** -
- Private IPv4 addresses:** 172.31.92.249
- Public IPv4 DNS:** ec2-44-203-161-215.compute-1.amazonaws.com
- Elastic IP addresses:** -
- AWS Compuete Optimizer findings:** -



Browse for this url: `http://<your_public_ip_address>/nagios`

The browser may ask you for your nagios credentials which set in the earlier steps

The username is nagiosadmin and enter the password that you set earlier



### Conclusion:

In this experiment, we successfully installed and configured Nagios Core on an Amazon Linux EC2 instance, showcasing its role in continuous monitoring within a DevOps environment. We learned about user management and service configuration, emphasizing Nagios's ability to monitor systems and networks effectively. This experience laid the groundwork for enhancing infrastructure reliability and integrating advanced monitoring strategies in future projects.