Name:Sahil Motiramani   Div:D15C   Rollno:35

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

Theory:

What is Nagios?

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

Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture.

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

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- Helps you to plan for infrastructure upgrades before outdated systems create failures ● You can maintain the security and availability of the service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

Features of Nagios

Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive ● Helps you to detect network errors or server crashes

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

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.
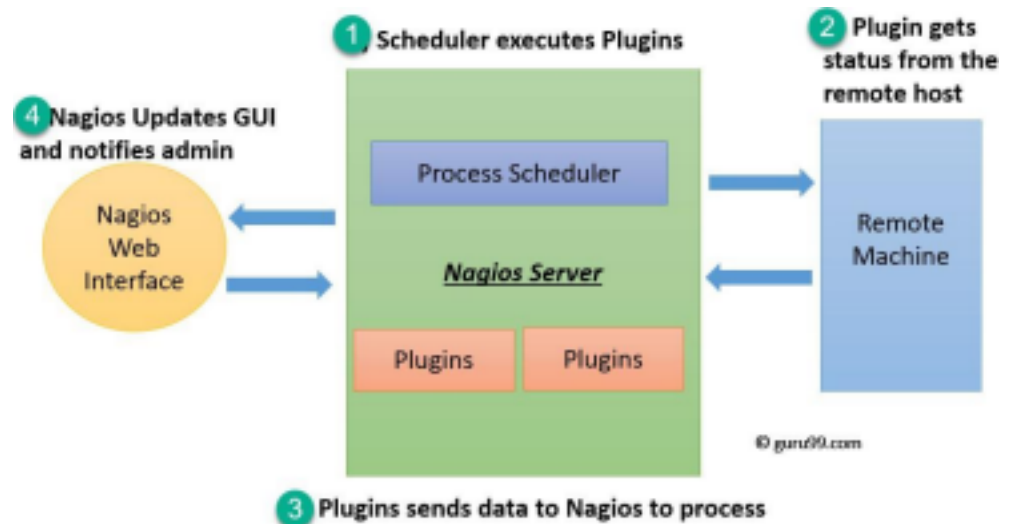
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Installation of Nagios

Prerequisites: AWS Free Tier

Name:Sahil Motiramani   Div:D15C   Rollno:35

Step 1: Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.

Select The OS Image as Amazon Linux.

Step 2: If you do not have a private key created or a .pem file created, click on create a key pair. Else select the key pair that you had created before. (Make sure you know where the .pem file for that key is present on your system)

AWS will create a security group for this instance. Keep the name of that instance saved.

Step 3: After creating the instance, click on Security Groups from the left side pane. Find the security group that was created for your instance. Click on the instance ID for that group.
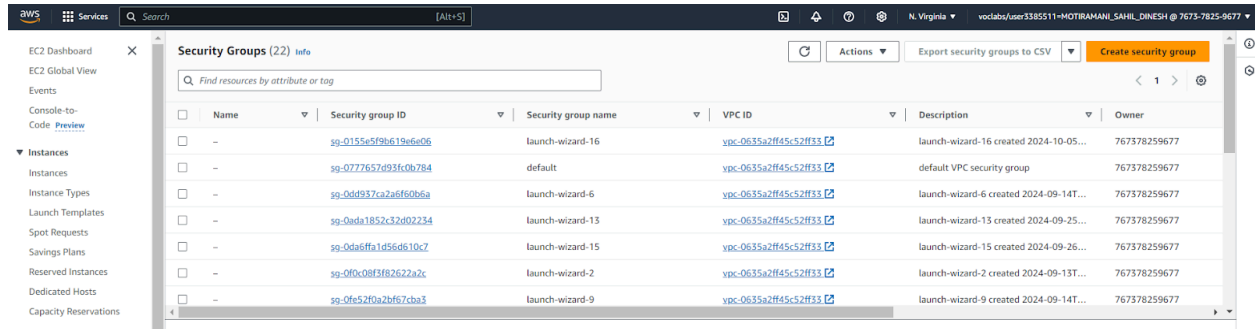
Here, click on Edit Inbound Rules.
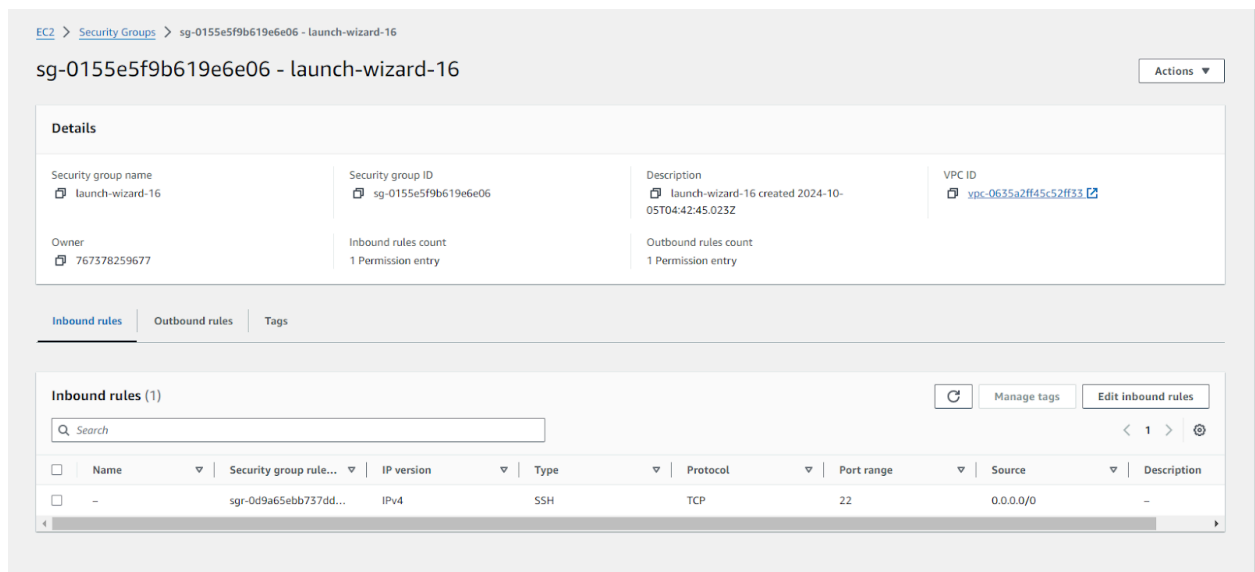
Now, click on add rules, and add teh rules for the following protocols:
HTTP, All ICMP - IPv6, HTTPS, All traffic, Custom TCP (Port 5666), All ICMP - IPv4
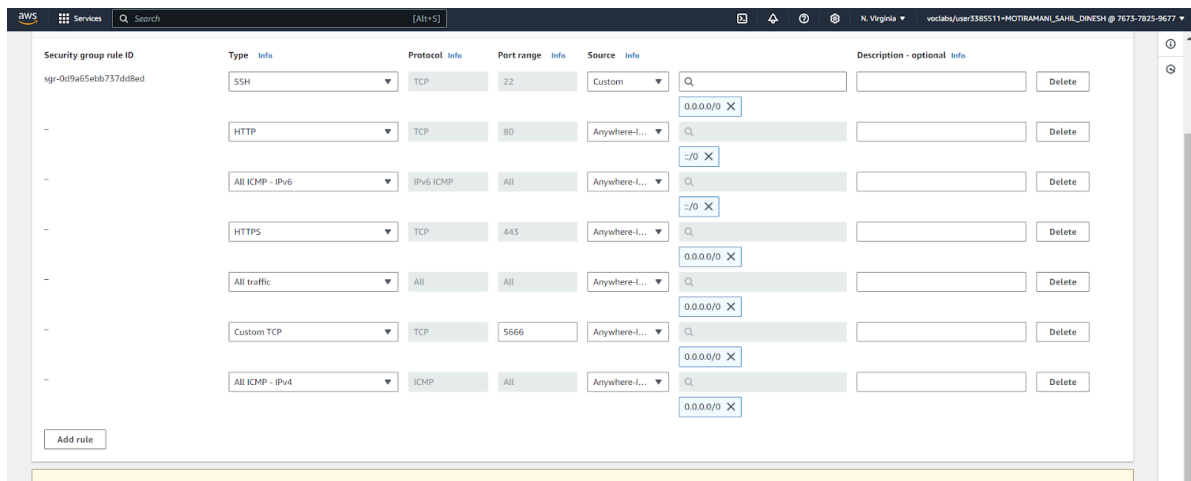
Click on save. This will add all the inbound rules to the security group.

Step 4: Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect.

Click on SSH client. Copy the example command.

Step 5: Now, we have to connect our local OS terminal to the instance using SSH. For this,
Open terminal wher the private key file is located (.pem)
Paste the copied SSH command and run it.

Step 6: Now we start working on this terminal. First run the command
sudo yum update
This command will check for any updates for the YUM library.

Step 7: We are going to install an Apache server and a PHP on this instance. For that, run this
command.
sudo yum install httpd php

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

Complete!
[ec2-user@ip-172-31-39-94 ~]$
```

Step 8: Next we install C/C++ compiler (GCC) along with the necessary C libraries required for compiling and running C programs. Use the following command.
sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-39-94 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:21:56 ago on Sun Oct  6 10:11:16 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
================================================================================================================================
 Package                  Architecture         Version                             Repository           Size
================================================================================================================================
Installing:
 gcc                      x86_64               11.4.1-2.amzn2023.0.2               amazonlinux           32 M
Installing dependencies:
 annobin-docs             noarch               10.93-1.amzn2023.0.1               amazonlinux           92 k
 annobin-plugin-gcc       x86_64               10.93-1.amzn2023.0.1               amazonlinux          887 k
 cpp                      x86_64               11.4.1-2.amzn2023.0.2              amazonlinux           10 M
 gc                       x86_64               8.0.4-5.amzn2023.0.2              amazonlinux          105 k
 glibc-devel              x86_64               2.34-52.amzn2023.0.11              amazonlinux           27 k
 glibc-headers-x86        noarch               2.34-52.amzn2023.0.11              amazonlinux          427 k
 guile22                  x86_64               2.2.7-2.amzn2023.0.3              amazonlinux          6.4 M
 kernel-headers           x86_64               6.1.109-118.189.amzn2023          amazonlinux          1.4 M
 libmpc                   x86_64               1.2.1-2.amzn2023.0.2              amazonlinux           62 k
 libtool-ltdl             x86_64               2.4.7-1.amzn2023.0.3              amazonlinux           38 k
 libxcrypt-devel          x86_64               4.4.33-7.amzn2023                 amazonlinux           32 k
 make                     x86_64               1:4.3-5.amzn2023.0.2              amazonlinux          534 k
```

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

Complete!
[ec2-user@ip-172-31-39-94 ~]$
```

Step 9: We would also need GD library and its development tools. For that, run this command
sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-39-94 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:23:03 ago on Sun Oct  6 10:11:16 2024.
Dependencies resolved.
================================================================================================================================
 Package                  Architecture         Version                             Repository           Size
================================================================================================================================
Installing:
 gd                       x86_64               2.3.3-5.amzn2023.0.3               amazonlinux          139 k
 gd-devel                 x86_64               2.3.3-5.amzn2023.0.3               amazonlinux           38 k
Installing dependencies:
 brotli                   x86_64               1.0.9-4.amzn2023.0.2              amazonlinux          314 k
 brotli-devel             x86_64               1.0.9-4.amzn2023.0.2              amazonlinux           31 k
 bzip2-devel              x86_64               1.0.8-6.amzn2023.0.2              amazonlinux          214 k
 cairo                    x86_64               1.17.6-2.amzn2023.0.1              amazonlinux          684 k
 cmake-filesystem         x86_64               3.22.2-1.amzn2023.0.4              amazonlinux           16 k
 fontconfig               x86_64               2.13.94-2.amzn2023.0.2            amazonlinux          273 k
 fontconfig-devel         x86_64               2.13.94-2.amzn2023.0.2            amazonlinux          128 k
 fonts-filesystem         noarch               1:2.0.5-12.amzn2023.0.2          amazonlinux          9.5 k
 freetype                 x86_64               2.13.2-5.amzn2023.0.1            amazonlinux          423 k
 freetype-devel           x86_64               2.13.2-5.amzn2023.0.1            amazonlinux          912 k
```

```
Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64                brotli-devel-1.0.9-4.amzn2023.0.2.x86_64          bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
  cairo-1.17.6-2.amzn2023.0.1.x86_64                cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64     fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64    fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch   freetype-2.13.2-5.amzn2023.0.1.x86_64
  freetype-devel-2.13.2-5.amzn2023.0.1.x86_64       gd-2.3.3-5.amzn2023.0.3.x86_64                    gd-devel-2.3.3-5.amzn2023.0.3.x86_64
  glib2-devel-2.74.7-689.amzn2023.0.2.x86_64        google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch  google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  graphite2-1.3.14-7.amzn2023.0.2.x86_64            graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64      harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64        harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64          jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
  langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch libICE-1.0.10-6.amzn2023.0.2.x86_64               libSM-1.2.3-8.amzn2023.0.2.x86_64
  libX11-1.7.2-3.amzn2023.0.4.x86_64                libX11-common-1.7.2-3.amzn2023.0.4.noarch         libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
  libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64            libXau-1.0.9-6.amzn2023.0.2.x86_64                libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64               libXpm-3.5.15-2.amzn2023.0.3.x86_64               libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
  libXrender-0.9.10-14.amzn2023.0.2.x86_64          libXt-1.2.0-4.amzn2023.0.2.x86_64                 libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64          libicu-67.1-7.amzn2023.0.3.x86_64                 libicu-devel-67.1-7.amzn2023.0.3.x86_64
  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64         libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64   libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
  libpng-2:1.6.37-10.amzn2023.0.6.x86_64            libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64      libselinux-devel-3.4-5.amzn2023.0.2.x86_64
  libsepol-devel-3.4-5.amzn2023.0.2.x86_64          libtiff-4.4.0-4.amzn2023.0.18.x86_64              libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
  libwebp-1.2.4-1.amzn2023.0.6.x86_64               libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64         libxcb-1.13.1-7.amzn2023.0.2.x86_64
  libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64         libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64        pcre2-devel-10.40-1.amzn2023.0.3.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64           pcre2-utf32-10.40-1.amzn2023.0.3.x86_64           pixman-0.40.0-3.amzn2023.0.3.x86_64
  sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64 xml-common-0.6.3-56.amzn2023.0.2.noarch          xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64              zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
```

Step 10: Now, we create a user called as 'nagios' and make sure that it has a home directory, and set up a password for it.
sudo adduser -m nagios
sudo passwd nagios

```
[ec2-user@ip-172-31-39-94 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-39-94 ~]$
```

Step 11: Create a user group called as 'nagcmd' to execute nagios commands.
sudo groupadd nagcmd

```
[ec2-user@ip-172-31-39-94 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-39-94 ~]$
```

Step 12: Add users apache and nagios to this user group.
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-39-94 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-39-94 ~]$
```

Step 13: We create a directory downloads, to store the files of nagios server that are downloaded.
mkdir ~/downloads
cd ~/downloads

```
[ec2-user@ip-172-31-39-94 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-39-94 downloads]$
```

Step 14:Now we need to install the latest versions of nogios-core and nagios-plugins. Go to the respective websites and check whether a better version is available.If newer versions are available, then right click on the download button → Copy link address.
Paste this link address in place of the current link in command.
If not run these commands.

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

```
[ec2-user@ip-172-31-83-157 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-28 04:04:23--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz          100%[===============================================================>]   1.97M  5.36MB/s    in 0.4s

2024-09-28 04:04:24 (5.36 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-83-157 downloads]$
```

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

```
[ec2-user@ip-172-31-39-94 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-06 10:37:57--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz          100%[===============================================================>]   1.97M  6.17MB/s    in 0.3s

2024-10-06 10:37:58 (6.17 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-39-94 downloads]$
```

Step 15:Now, we need to extract nagios-core file into the same directory. For that, we will use tar command.
tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-39-94 downloads]$ tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
```

```
nagios-4.5.5/xdata/.gitignore
nagios-4.5.5/xdata/Makefile.in
nagios-4.5.5/xdata/xcddefault.c
nagios-4.5.5/xdata/xcddefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-39-94 downloads]$
```

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

An error could be encountered here: ./configure: no such path or directory

Solution: Navigate to the nagios-4.5.5 folder in downloads. (version could vary)

Steps: ls

- cd nagios-4.5.5 (use the version shown by your ls command)
- ./configure --with-command-group=nagcmd

Another error could be Cannot find SSL headers.

To solve this, we need to install OpenSSL Dev Library

Steps:
sudo yum install openssl-devel

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

```
[ec2-user@ip-172-31-39-94 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
```

```
*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

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


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


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

[ec2-user@ip-172-31-39-94 nagios-4.5.5]$
```

Step 17: We need to compile all components of this software according to the instruction in the Makefile. For that, use this command:
make all

Then,
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install: cannot stat 'nagios': No such file or directory
make[1]: *** [Makefile:188: install] Error 1
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
```

```
*** Config files installed ***

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

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

*** External command directory configured ***
```

Step 18: We need to update the email linked with this server to our email for it to send notifications (if any needed).
sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```
  GNU nano 5.8                                            /usr/local/nagios/etc/objects/contacts.cfg
###############################################################################
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
#        group definitions that you can reference in host and service
#        definitions.
#
#        You don't need to keep these definitions in a separate file from your
#        other object definitions.  This has been done just to make things
#        easier to understand.
#
###############################################################################


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

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

define contact {

    contact_name         nagiosadmin             ; Short name of user
    use                  generic-contact         ; Inherit default values from generic-contact template (defi
    alias                Nagios Admin            ; Full name of user
    email                2022.sahil.motiramani@ves.ac.in ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}



###############################################################################
#
# CONTACT GROUPS
#
###############################################################################
```

Here, change the email under 'define contact{}' to your email address.
To save this use the following shortcut sequence CTRL+O→Enter→CTRL+X.

CTRL+O: Overwrite the existing file with edited file
CTRL+X: Exit nano editor.

Step 19: We need to install the necessary configuration files for the Nagios web interface. Run the following command.
sudo make install-webconf

```
[ec2-user@ip-172-31-39-94 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-172-31-39-94 nagios-4.5.5]$ S
```

Step 20: Now we need to setup a user to access this nagios web interface. So we run this command to create a user called 'nagiosadmin'.

Keep this username and password saved as it is needed to login to the web interface.
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-39-94 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-172-31-39-94 nagios-4.5.5]$
```

Step 21: Restart the apache server to apply all the recent configurations.
sudo service httpd restart

```
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-39-94 nagios-4.5.5]$
```

Step 22: Now we go back to the downloads folder and extract the files of nagios plugin.
cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz (Version may vary)

```
[ec2-user@ip-172-31-39-94 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/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-39-94 downloads]$
```

Step 23: Again, we need to install the configurations for these files.
cd nagios-plugins-2.4.11 (version may vary)
./configure --with-nagios-user=nagios --with-nagios-group=nagios

```
[ec2-user@ip-172-31-39-94 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-94 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagio:
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
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
```

```
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-39-94 nagios-plugins-2.4.11]$
```

Step 24: We need to compile all components of this software according to the instruction in the Makefile. For that, use the commands:
make
sudo make install

```
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-39-94 nagios-plugins-2.4.11]$
```

Step 25: We need to register the Nagios service with the system, which would make it able to manage the server status. So run the following commands
sudo chkconfig --add nagios
sudo chkconfig nagios on

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl enable nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl start nagios
```

Step 26: We need to verify the Nagios configuration for any syntax errors or issues before starting or restarting the Nagios service.
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ 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-172-31-39-94 nagios-4.4.6]$ sudo systemctl enable nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo systemctl start nagios
[ec2-user@ip-172-31-39-94 nagios-4.4.6]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
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-172-31-39-94 nagios-4.4.6]$
```

Step 27:

sudo service nagios start

sudo systemctl status nagios

Step 28: Now, go to EC2 instance and click on instance id. Then, click on the copy icon just before the public ip address on public IP.

Name:Sahil Motiramani   Div:D15C   Rollno:35

Step 29: Open a new tab. In the address bar type http://<publicipaddress>/nagios.
This would be in the output

Conclusion:

In this experiment, we have learned how to install and set up Nagios Core, Nagios Plugins, and NRPE on a Linux machine. We are working with an Amazon Linux OS instance that has been configured with the necessary security rules. It's crucial to ensure that the links for Nagios Core and Nagios Plugins used in the wget commands are current and up-to-date. After downloading, we need to extract and configure these files to prevent any issues when starting the server. Once the setup is complete, we can launch the Nagios server. By using the public IP address of the EC2 instance, we can access the Nagios dashboard by entering that IP into a web browser.