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

Theory:

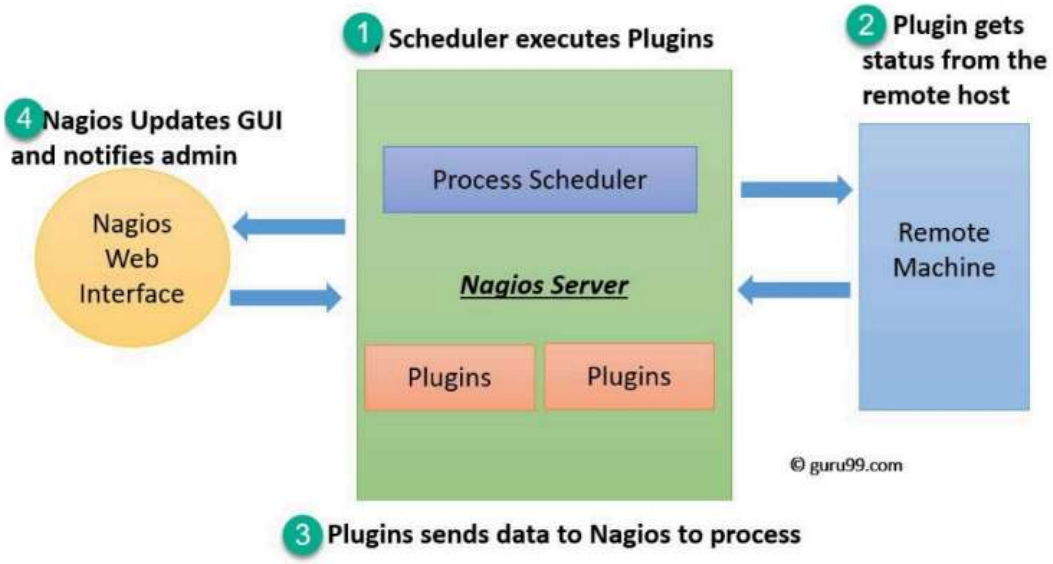
What is Nagios?
Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the 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- Helps you to plan for infrastructure upgrades before outdated systems create failures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
- Utilises 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

Name : Sandeshkumar.M.Yadav **Div : D15C** **Roll : 61**

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.

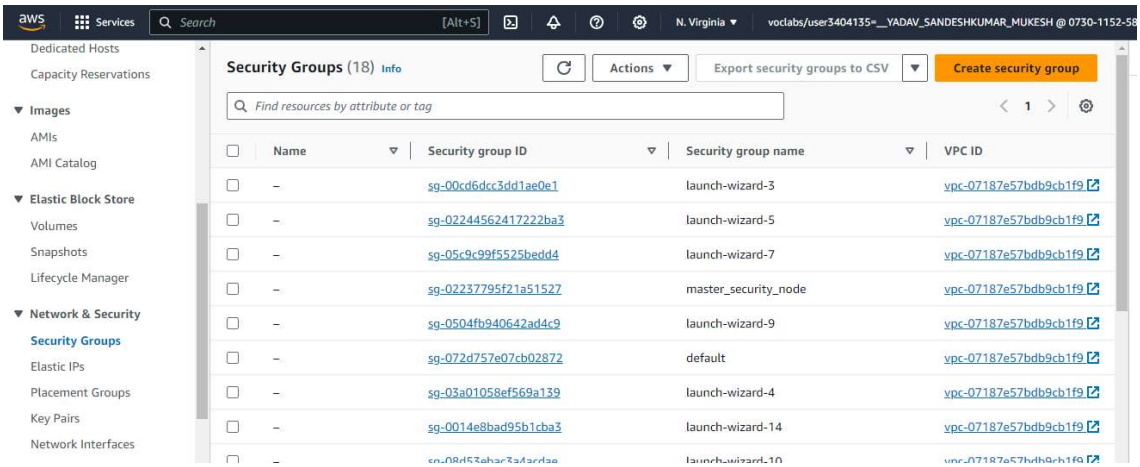


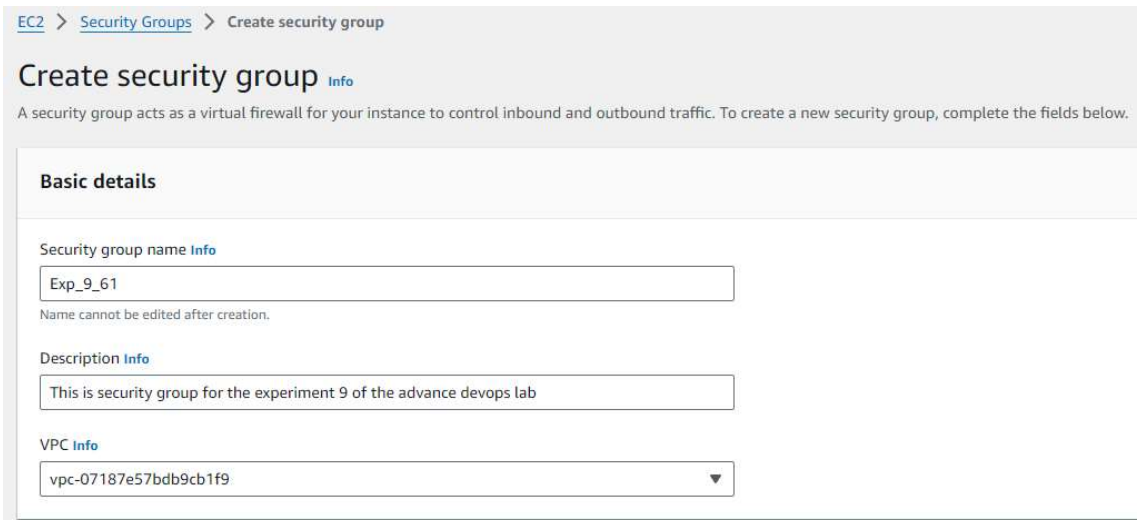
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

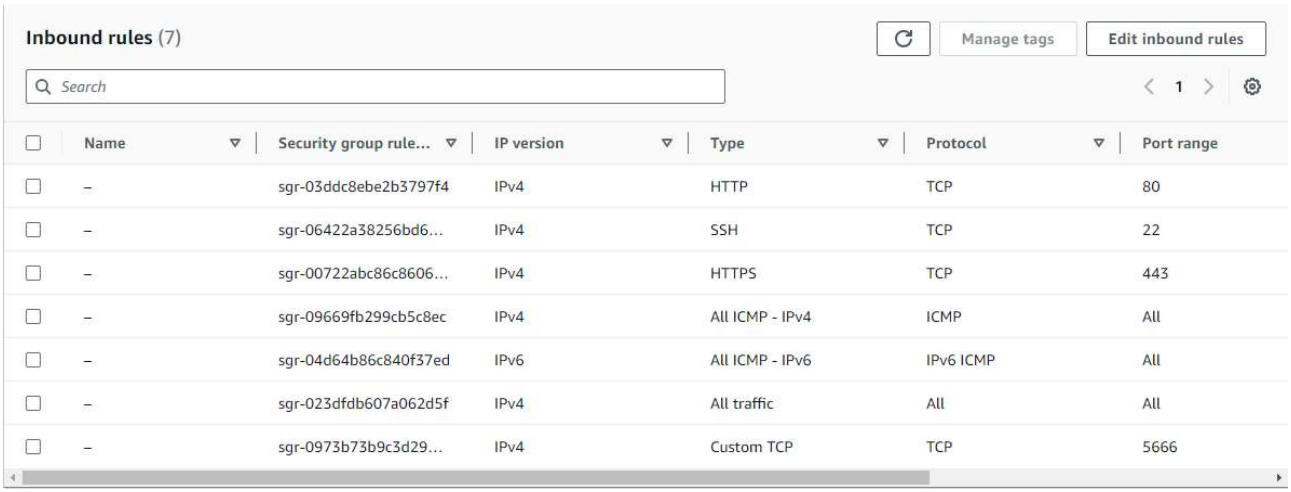
Name : Sandeshkumar.M.Yadav **Div : D15C** **Roll : 61**

Steps :

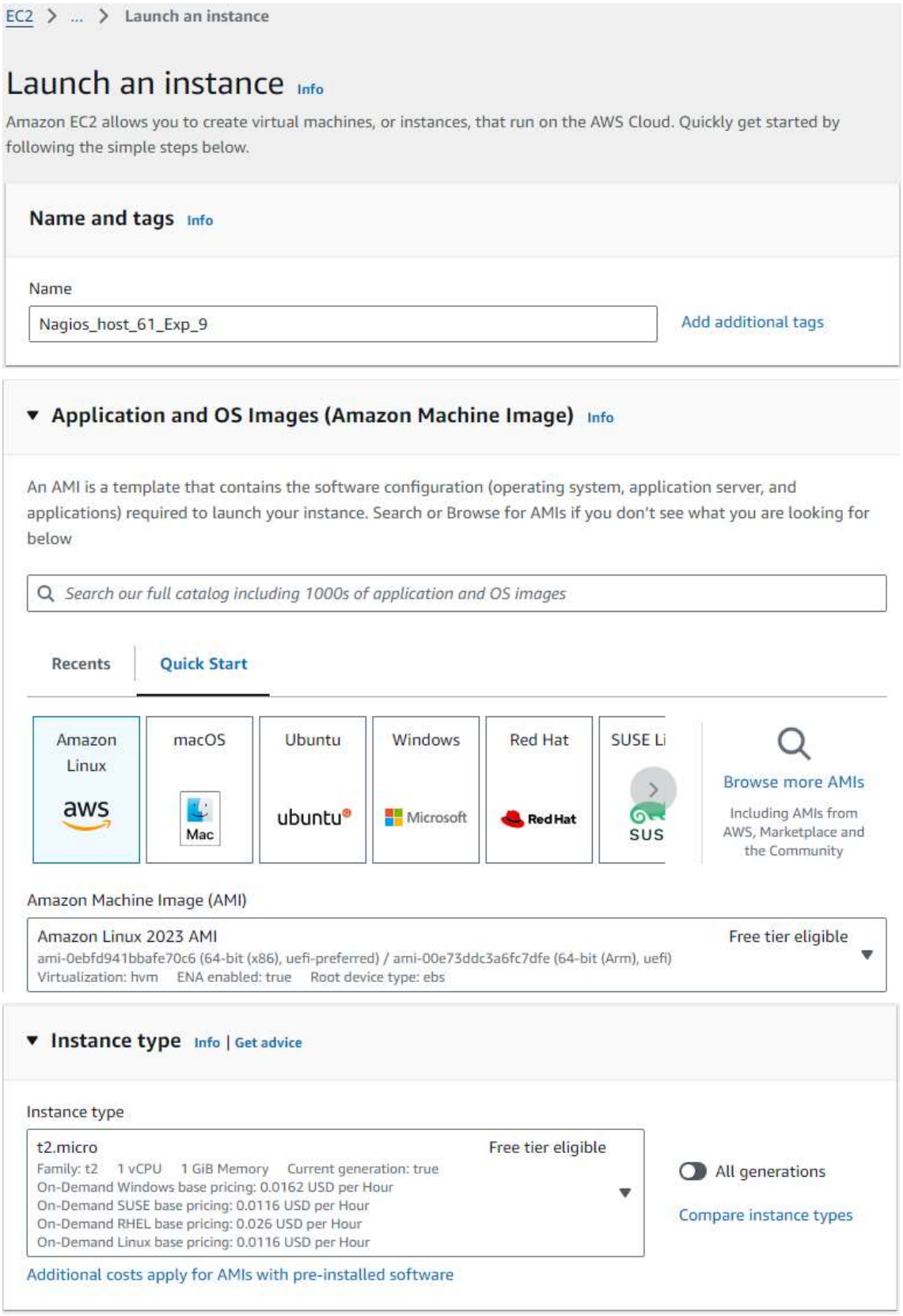
Step 1: : Login to your AWS account Personal / Academy. Click on EC2 instance then click on Create Security Group. Give the name as Nagios and any description and add the following inbounds rules.

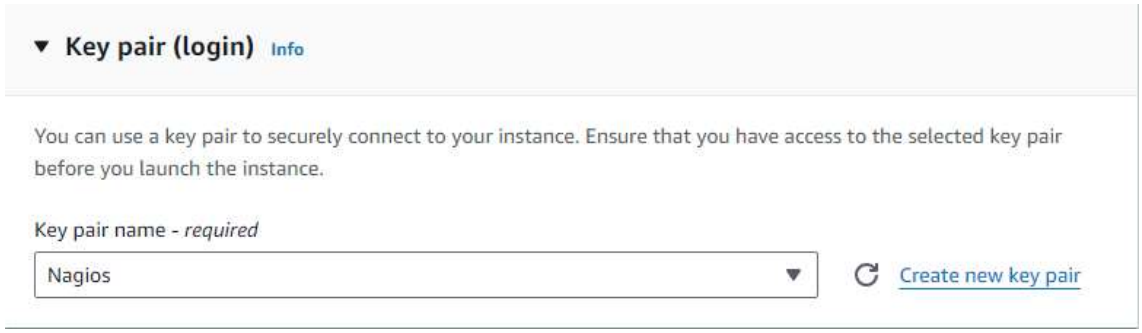




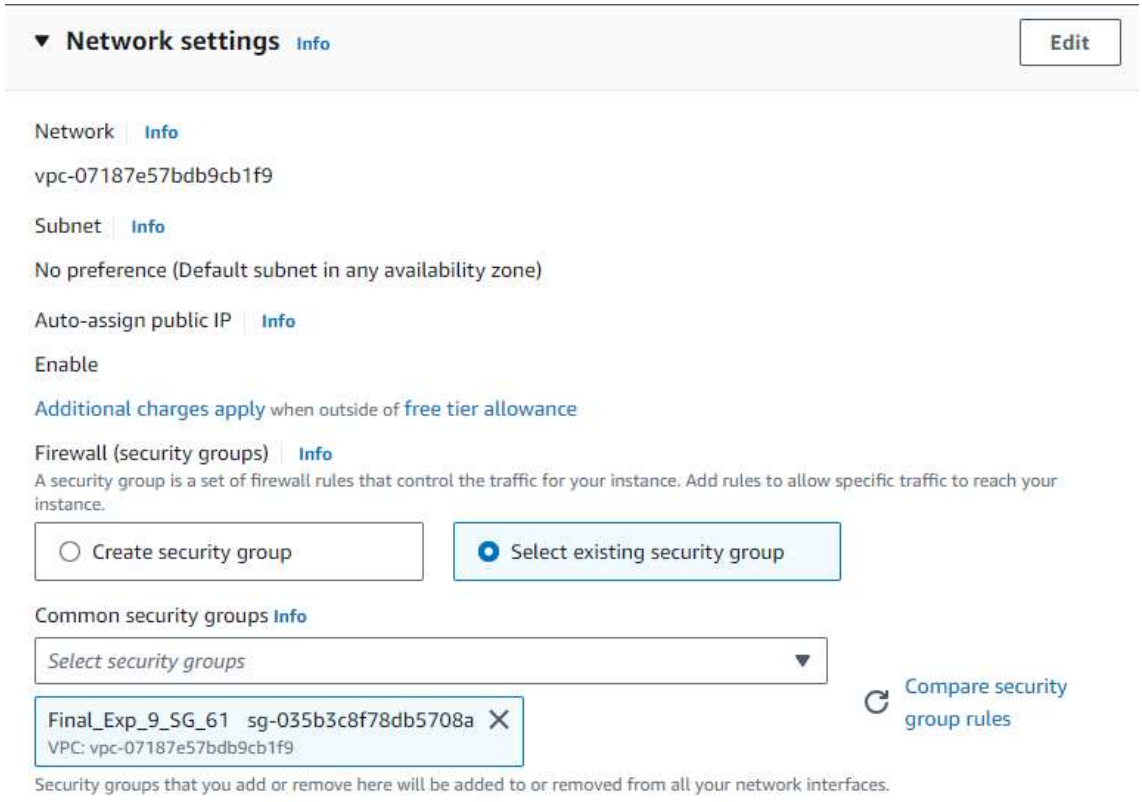


Step 2: Now Create a new EC2 instance. Name: Nagios-host ,AMI: Amazon Linux, Instance Type: t2.micro

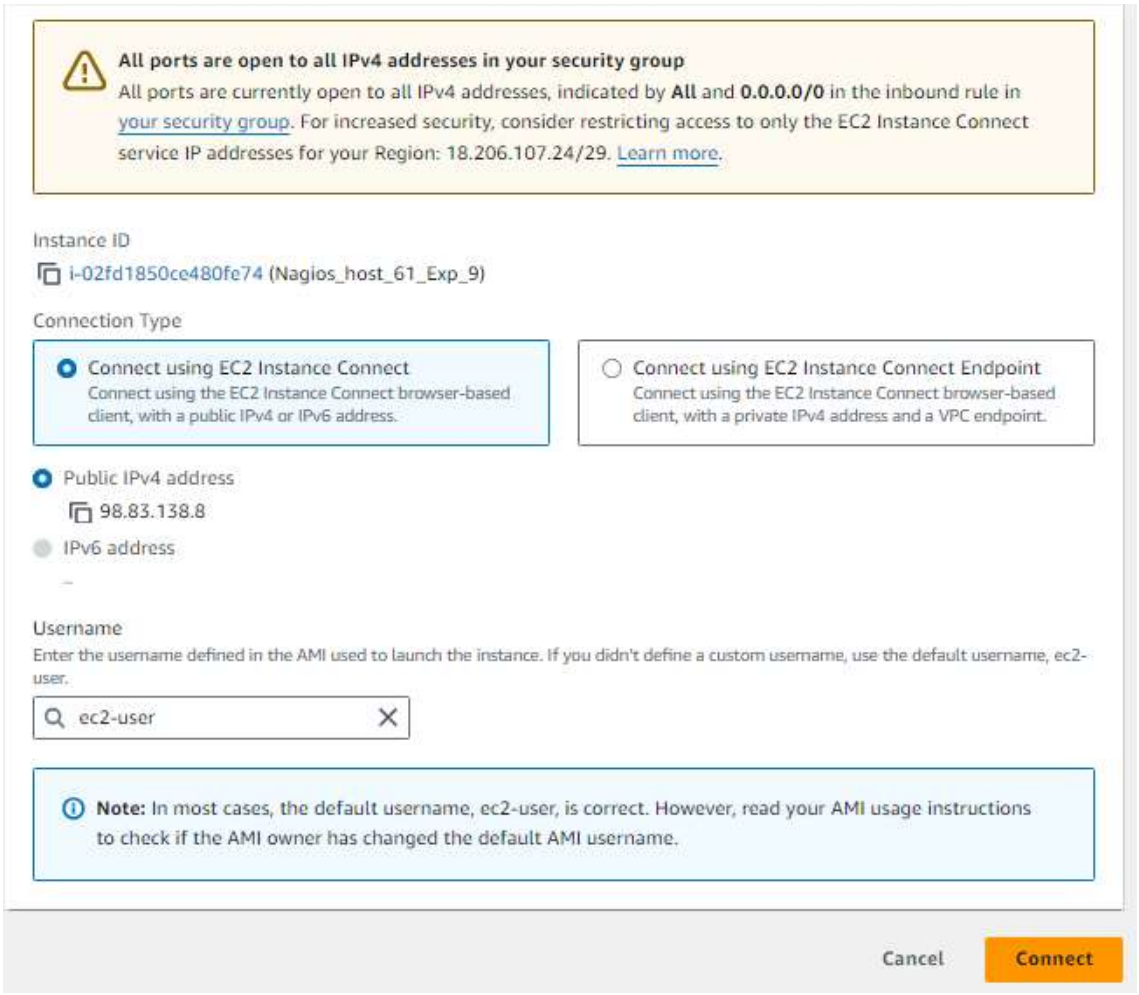




Select the Existing Security Group and select the Security Group we have created in Step 1.



Step 3: Now After creating the EC2 Instance click on connect and scroll down and connect on web server



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

```
[root@ip-172-31-35-58 ec2-user]# sudo adduser -m nagios
[root@ip-172-31-35-58 ec2-user]# sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-35-58 ec2-user]#
```

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

```
[root@ip-172-31-41-153 ec2-user]# sudo groupadd nagcmd
[root@ip-172-31-41-153 ec2-user]# sudo usermod -a -G nagcmd nagios
[root@ip-172-31-41-153 ec2-user]# sudo usermod -a -G nagcmd apache
usermod: user 'apache' does not exist
[root@ip-172-31-41-153 ec2-user]#
```

It looks like the user apache doesn't exist on your Amazon Linux machine. On Amazon Linux, the web server user is typically called apache (for the Apache HTTP Server)

Install Apache (if not installed): If you confirm that Apache is not installed, you can install it with:
sudo yum install httpd

Then, start the Apache service:
sudo systemctl start httpd
sudo systemctl enable httpd

Add the nagios and apache users to the nagcmd group: If the Apache user is confirmed, you can add it to the group as you intended:
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

```
[root@ip-172-31-41-153 ec2-user]# mkdir ~/downloads
[root@ip-172-31-41-153 ec2-user]# cd ~/downloads
[root@ip-172-31-41-153 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

```
[root@ip-172-31-41-153 downloads]# wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-02 09:49:39--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 3.215.172.219, 18.208.125.13, 34.237.219.119, ...
Connecting to go.nagios.org (go.nagios.org)|3.215.172.219|: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_camp
=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-02 09:49:39--  http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Downlo
rm&utm_campaign=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_ca
n=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-02 09:49:39--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Downlo
orm&utm_campaign=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  7.53MB/s    in 0.3s
```

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

```
[root@ip-172-31-41-153 downloads]# wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-02 09:50:25--  https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz        100%[=================================================================>]   2.62M  9.09MB/s    in 0.3s

2024-10-02 09:50:26 (9.09 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[root@ip-172-31-41-153 downloads]#
```

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))

```
[root@ip-172-31-41-153 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/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
[root@ip-172-31-41-153 downloads]#
```

Step 9: Now change the directory to nagios-4.5.5

```
[root@ip-172-31-41-153 downloads]# cd nagios-4.5.5
[root@ip-172-31-41-153 nagios-4.5.5]#
```

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

```
[root@ip-172-31-41-153 downloads]# cd nagios-4.5.5
[root@ip-172-31-41-153 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... no
checking for cc... no
checking for cl.exe... no
checking for clang... no
configure: error: in '/root/downloads/nagios-4.5.5':
configure: error: no acceptable C compiler found in $PATH
See 'config.log' for more details
```

The error you're encountering indicates that there is no C compiler installed on your system. The ./configure script is trying to find a C compiler (such as gcc or clang) but is failing because none is available.

To resolve this, you need to install a C compiler. You can install gcc, the GNU Compiler Collection, by running the following command on your Amazon Linux machine:
sudo yum groupinstall "Development Tools"

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo yum groupinstall "Development Tools"
Last metadata expiration check: 0:11:58 ago on Wed Oct  2 09:43:20 2024.
No match for group package "system-rpm-config"
No match for group package "rcs"
No match for group package "pkgconfig"
Dependencies resolved.
================================================================================
 Package                    Architecture        Version                    Repository          Size
================================================================================
Installing group/module packages:
 autoconf                   noarch              2.69-36.amzn2023.0.3       amazonlinux        666 k
 automake                   noarch              1.16.5-9.amzn2023.0.3      amazonlinux        677 k
 bison                      x86_64              3.7.4-2.amzn2023.0.2       amazonlinux        925 k
 byacc                      x86_64              2.0.20210109-2.amzn2023.0.3 amazonlinux        90 k
 cscope                     x86_64              15.9-15.amzn2023.0.3       amazonlinux        288 k
 ctags                      x86_64              5.9~1.20210725.0.amzn2023.0.2 amazonlinux     719 k
 diffstat                   x86_64              1.64-4.amzn2023.0.2        amazonlinux         43 k
 doxygen                    x86_64              2:1.9.4-1.amzn2023.0.3     amazonlinux        4.7 M
 elfutils                   x86_64              0.188-3.amzn2023.0.2       amazonlinux        525 k
```

```
  urw-base35-gothic-fonts-20200910-6.amzn2023.0.2.noarch
  urw-base35-nimbus-mono-ps-fonts-20200910-6.amzn2023.0.2.noarch
  urw-base35-nimbus-roman-fonts-20200910-6.amzn2023.0.2.noarch
  urw-base35-nimbus-sans-fonts-20200910-6.amzn2023.0.2.noarch
  urw-base35-p052-fonts-20200910-6.amzn2023.0.2.noarch
  urw-base35-standard-symbols-ps-fonts-20200910-6.amzn2023.0.2.noarch
  urw-base35-z003-fonts-20200910-6.amzn2023.0.2.noarch
  utf8proc-2.6.1-2.amzn2023.0.2.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch
  xorg-x11-fonts-ISO8859-1-100dpi-7.5-31.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64
  zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
[root@ip-172-31-41-153 nagios-4.5.5]#
```

Now rerun the command ./configure --with-command-group=nagcmd
At the end we have found the error of cannot find ssl header

```
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
[root@ip-172-31-41-153 nagios-4.5.5]#
```

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

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo yum install openssl-devel
Last metadata expiration check: 0:15:35 ago on Wed Oct  2 09:43:20 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                  28 MB/s | 3.0 MB     00:00
--------------------------------------------------------------------------------
Total                                                           18 MB/s | 3.0 MB     00:00
```

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

```
[root@ip-172-31-41-153 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 for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... 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.

[root@ip-172-31-41-153 nagios-4.5.5]#
```

Step 11: Now run the following commands to steup the Nagios.
sudo make install
Before that run make all to compile main program and CGIs

```
     - 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.

[root@ip-172-31-41-153 nagios-4.5.5]#
```

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo make install
cd ./base && make install
make[1]: Entering directory '/root/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 '/root/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/root/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/root/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 '/root/downloads/nagios-4.5.5/cgi'
```

sudo make install-init

```
[root@ip-172-31-41-153 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
[root@ip-172-31-41-153 nagios-4.5.5]#
```

sudo make install-config

```
[root@ip-172-31-41-153 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.

[root@ip-172-31-41-153 nagios-4.5.5]#
```

sudo make install-webconf

The error you're seeing indicates that the directory /etc/httpd/conf.d/ does not exist, which means that Apache HTTP Server (httpd) might not be installed on your system.

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
/usr/bin/install: cannot create regular file '/etc/httpd/conf.d/nagios.conf': No such file or directory
make: *** [Makefile:351: install-webconf] Error 1
[root@ip-172-31-41-153 nagios-4.5.5]#
```

To resolve this, follow these steps:
sudo yum install httpd
sudo mkdir -p /etc/httpd/conf.d
sudo make install-webconf
sudo systemctl start httpd
sudo systemctl enable httpd

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo yum install httpd
Last metadata expiration check: 0:25:04 ago on Wed Oct  2 09:43:20 2024.
Dependencies resolved.
================================================================================
 Package                         Architecture            Version
================================================================================
Installing:
 httpd                           x86_64                  2.4.62-1.amzn2023
Installing dependencies:
 generic-logos-httpd             noarch                  18.0.0-12.amzn2023.0.3
 httpd-core                      x86_64                  2.4.62-1.amzn2023
 httpd-filesystem                noarch                  2.4.62-1.amzn2023
 httpd-tools                     x86_64                  2.4.62-1.amzn2023
 mailcap                         noarch                  2.1.49-3.amzn2023.0.3
Installing weak dependencies:
 mod_http2                       x86_64                  2.0.27-1.amzn2023.0.3
 mod_lua                         x86_64                  2.4.62-1.amzn2023

Transaction Summary
================================================================================
Install  8 Packages
```

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo mkdir -p /etc/httpd/conf.d
[root@ip-172-31-41-153 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 ***

[root@ip-172-31-41-153 nagios-4.5.5]# sudo systemctl start httpd
[root@ip-172-31-41-153 nagios-4.5.5]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-41-153 nagios-4.5.5]#
```

Now sudo make install-webconf Has been successfully run

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

```
[root@ip-172-31-41-153 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
[root@ip-172-31-41-153 nagios-4.5.5]#
```

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

```
[root@ip-172-31-41-153 nagios-4.5.5]# sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ip-172-31-41-153 nagios-4.5.5]#
```

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

```
[root@ip-172-31-41-153 nagios-4.5.5]# cd ~/downloads
[root@ip-172-31-41-153 downloads]# tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
nagios-plugins-2.4.11/gl/m4/
nagios-plugins-2.4.11/gl/m4/00gnulib.m4
nagios-plugins-2.4.11/gl/m4/absolute-header.m4
nagios-plugins-2.4.11/gl/m4/alloca.m4
nagios-plugins-2.4.11/gl/m4/arpa_inet_h.m4
nagios-plugins-2.4.11/gl/m4/base64.m4
nagios-plugins-2.4.11/gl/m4/btowc.m4
```

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

```
[root@ip-172-31-41-153 downloads]# cd nagios-plugins-2.4.11
[root@ip-172-31-41-153 nagios-plugins-2.4.11]# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
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
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 we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
```

```
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[root@ip-172-31-41-153 nagios-plugins-2.4.11]#
```

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

```
[root@ip-172-31-41-153 nagios-plugins-2.4.11]# sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
[root@ip-172-31-41-153 nagios-plugins-2.4.11]# sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[root@ip-172-31-41-153 nagios-plugins-2.4.11]# sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
[root@ip-172-31-41-153 nagios-plugins-2.4.11]#
```

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

```
[root@ip-172-31-41-153 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...
```

```
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
[root@ip-172-31-41-153 nagios-plugins-2.4.11]#
```

cd

sudo service nagios start

```
[root@ip-172-31-41-153 nagios-plugins-2.4.11]# cd
[root@ip-172-31-41-153 ~]# sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[root@ip-172-31-41-153 ~]#
```

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

```
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
[root@ip-172-31-41-153 ~]#
```

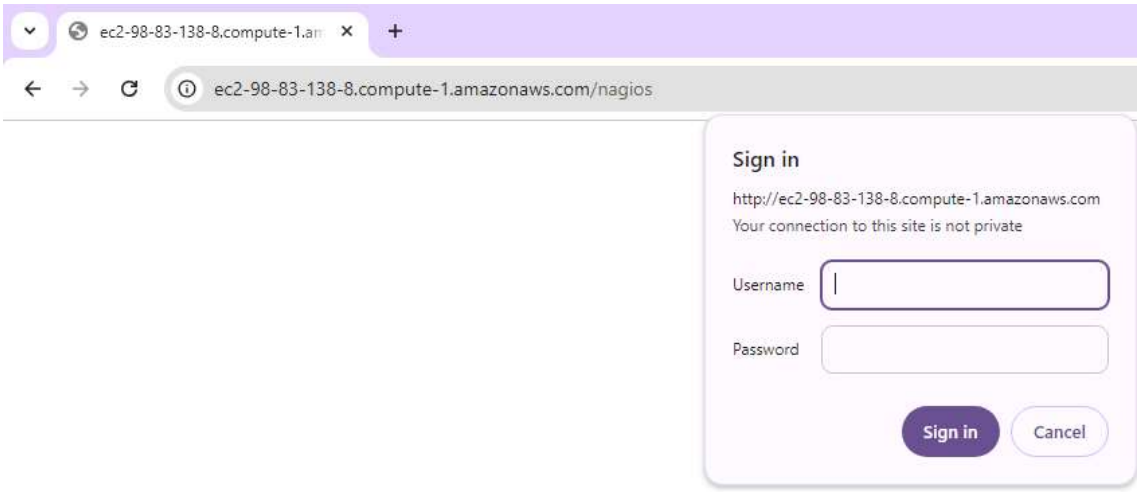
sudo systemctl status nagios

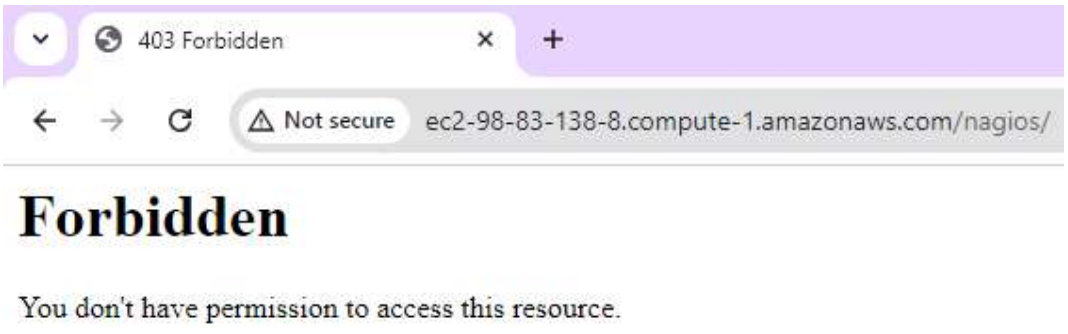
```
[root@ip-172-31-41-153 ~]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-02 10:23:28 UTC; 2min 14s ago
       Docs: https://www.nagios.org/documentation
   Main PID: 55480 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.5M
        CPU: 109ms
     CGroup: /system.slice/nagios.service
             ├─55480 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─55481 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─55482 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─55483 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─55484 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─55485 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 10:23:28 ip-172-31-41-153.ec2.internal nagios[55480]: wproc: Registry request: name=Core Worker 55483;pid=55483
Oct 02 10:23:28 ip-172-31-41-153.ec2.internal nagios[55480]: wproc: Registry request: name=Core Worker 55482;pid=55482
Oct 02 10:23:28 ip-172-31-41-153.ec2.internal nagios[55480]: wproc: Registry request: name=Core Worker 55481;pid=55481
Oct 02 10:23:28 ip-172-31-41-153.ec2.internal nagios[55480]: Successfully launched command file worker with pid 55485
Oct 02 10:23:28 ip-172-31-41-153.ec2.internal nagios[55480]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdout) stderr: execvp(/usr/local/n>
Oct 02 10:24:05 ip-172-31-41-153.ec2.internal nagios[55480]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD;1;(No output on stdout) stderr:>
Oct 02 10:24:28 ip-172-31-41-153.ec2.internal nagios[55480]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(/usr/local/n>
lines 1-23
[root@ip-172-31-41-153 ~]#
```

Step 15: We can see we have successfully launched the Nagios now . Open http://<instance public ip >/nagios/ here it is http://52.23.153.85/nagios we can see the running web page of nagios

Or else
http://ec2-52-23-153-85.compute-1.amazonaws.com/nagios





Step 16 :

1. Check Apache Configuration
Make sure that Apache is configured to allow access to the Nagios web interface.

Open the Nagios configuration file for Apache, typically found at /etc/httpd/conf.d/nagios.conf or a similar location.
sudo nano /etc/httpd/conf.d/nagios.conf

Look for the section that defines access controls. It might look something like this:

```
<Directory "/usr/local/nagios/share">
   Options None
   AllowOverride None
   Require all granted
</Directory>
```

Ensure that the Require all granted directive is present. If it's set to Require all denied, change it to Require all granted.

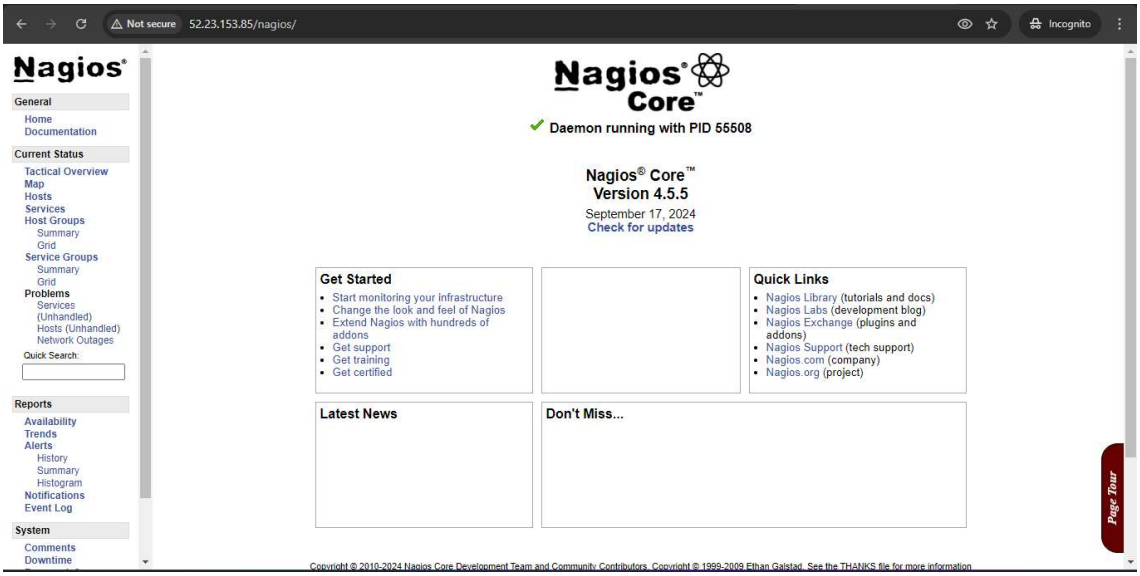
2.Check Directory Permissions
Ensure that the Apache user (apache or www-data depending on your distribution) has the correct permissions to access the Nagios web directory.

sudo chown -R apache:apache /usr/local/nagios/share
3.SELinux (if applicable)
If SELinux is enabled, it may block access even if Apache permissions are correct. You can check the status with:
sestatus

If it is enabled, you might need to adjust the security context:
sudo chcon -R -t httpd_sys_content_t /usr/local/nagios/share

4.Restart Apache
After making changes, restart the Apache service:

sudo systemctl restart httpd



Open
http://<instance public ip >/nagios/ here it is http://18.234.24.186/nagios we can see the running web page of nagios

Or else
http://ec2-52-23-153-85.compute-1.amazonaws.com/nagios

Conclusion :

In this experiment, we have set up Nagios Core with plugins on Amazon Linux, which will help us understand continuous monitoring and installation. It is important to note that the set of rules added in Step 1 are crucial for the success of this experiment. By configuring Nagios, we enable effective monitoring of systems, networks, and applications, allowing for timely alerts and proactive management of infrastructure performance and security in a DevOps environment.