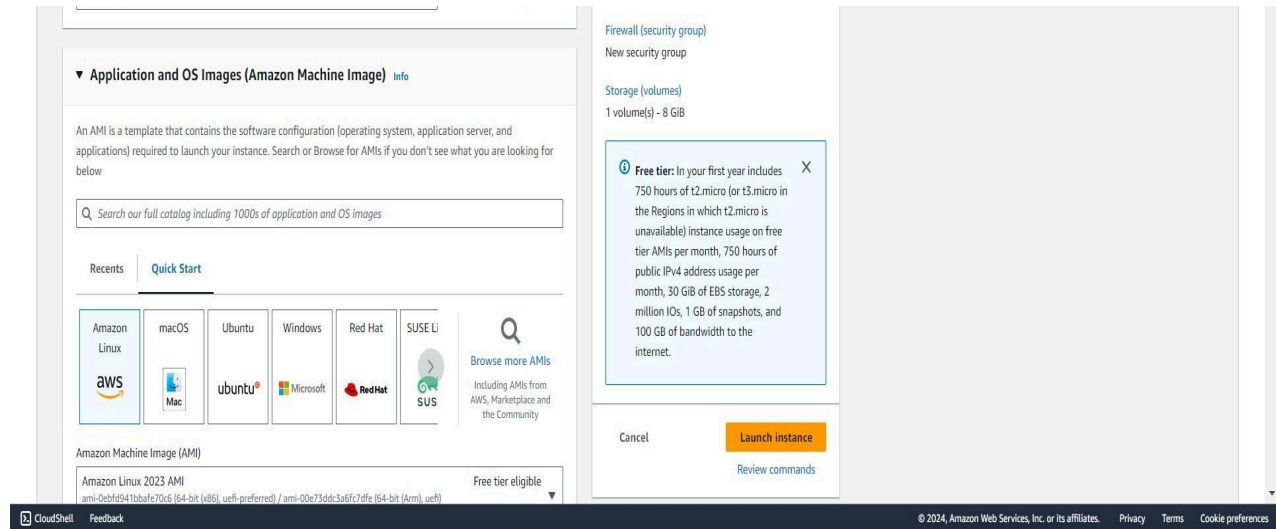


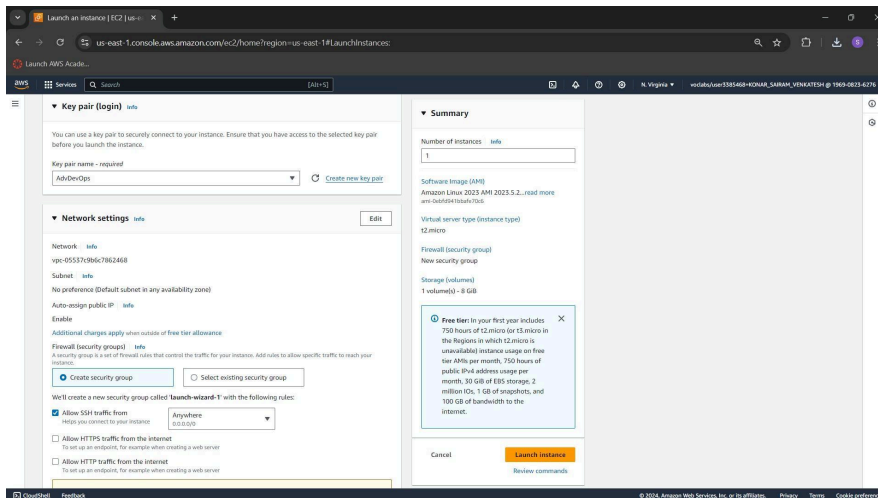
**Aim:** To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

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.

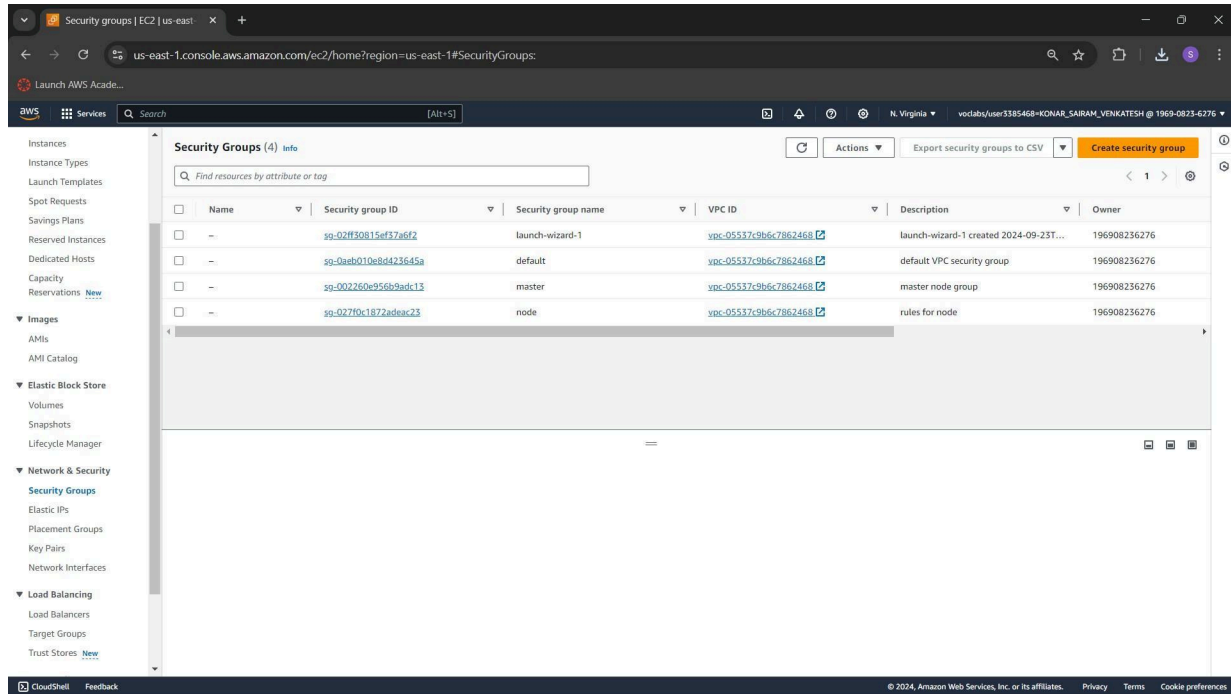
Step 2: If you do not have a private key created or a .pem file created, click on create a key pair. Else select the key pair that you had created before. (Make sure you know where the .pem file for that key is present on your system)



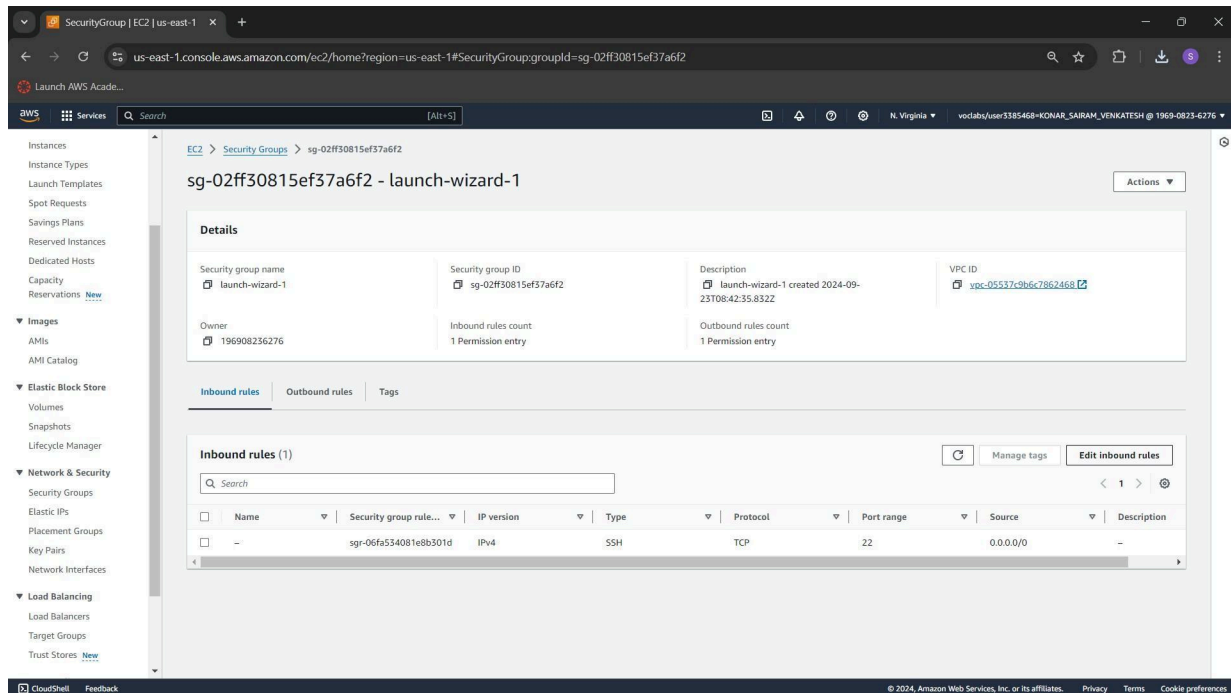
AWS will create a security group for this instance. Keep the name of that instance saved.

2024-  
25

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.

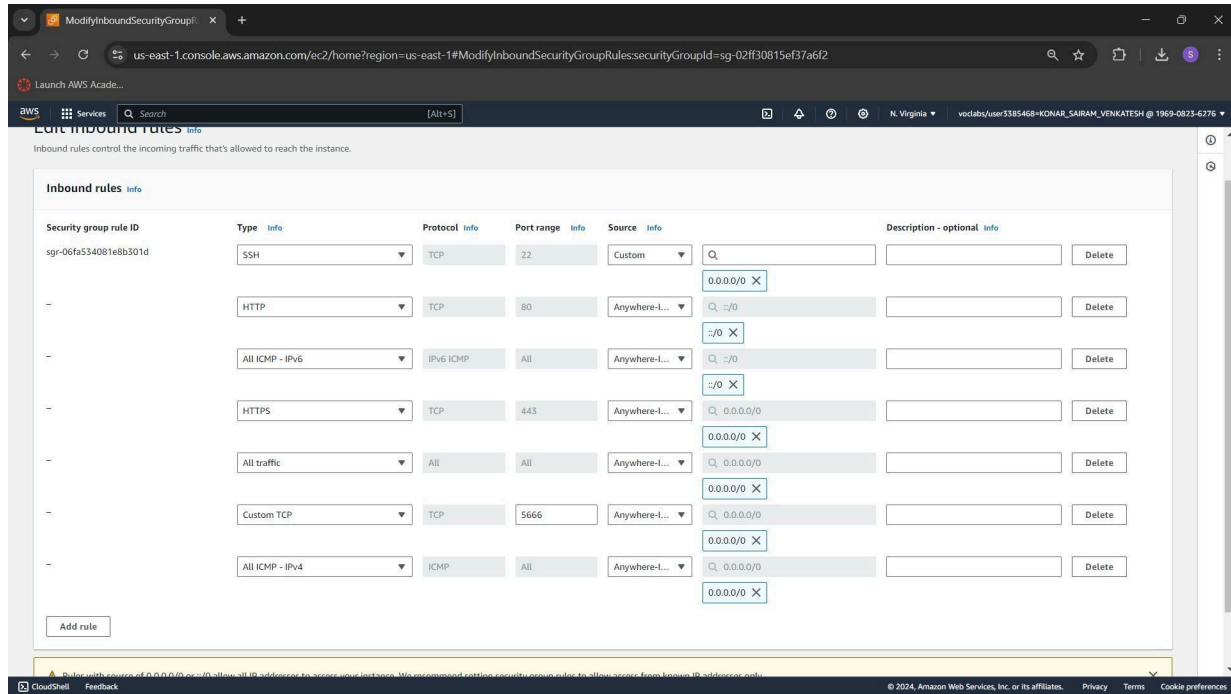


Here, click on Edit Inbound Rules.

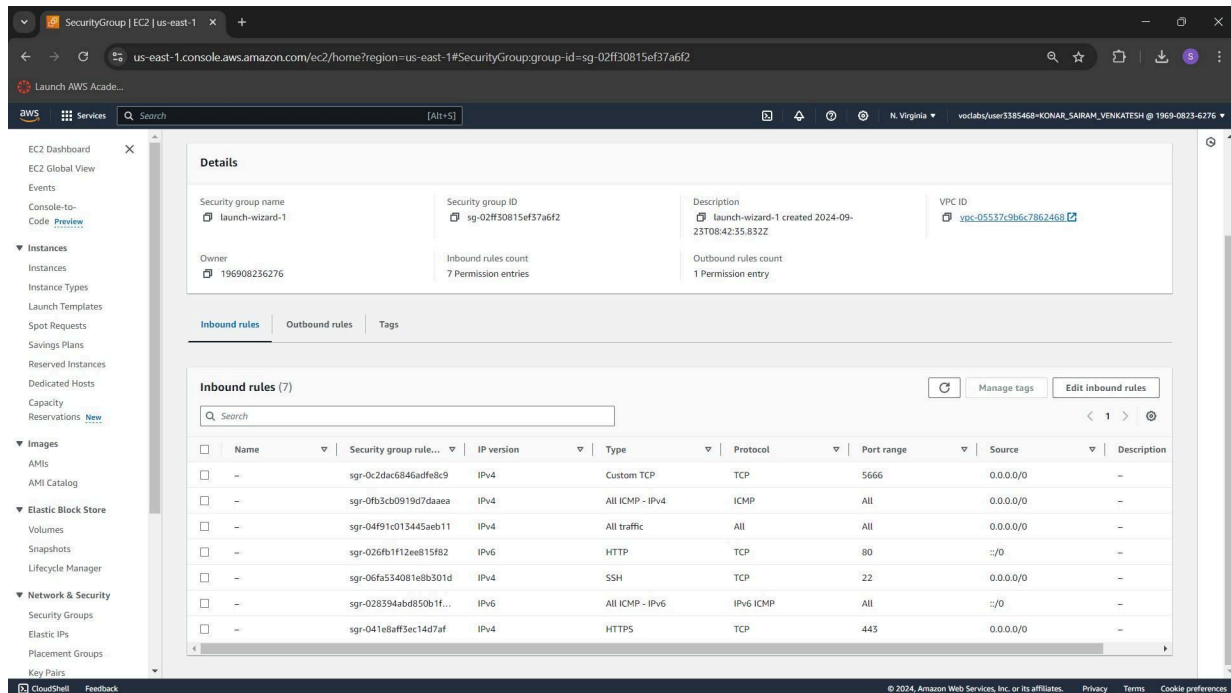


# Kaushal Galav D15C 11 EXP 9

Now, click on add rules, and add the rules for the following protocols:  
HTTP, All ICMP - IPv6, HTTPS, All traffic, Custom TCP (Port 5666), All ICMP - IPv4



Click on save. This will add all the inbound rules to the security group.



2024-  
25

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.

The screenshot shows the AWS Management Console interface for connecting to an EC2 instance. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-0e1c706c2051d5121`. The console header includes the AWS logo, a search bar, and the user's profile information: `veclab4/user3385468-KONAR_SAIRAM_VENKATESH @ 1969-0823-6276`.

The main content area is titled "Connect to instance" with a sub-header "Connect to your instance i-0e1c706c2051d5121 (nagios-host-27) using any of these options". Below this, there are four tabs: "EC2 Instance Connect", "Session Manager", "SSH client" (which is selected), and "EC2 serial console".

Under the "SSH client" tab, the "Instance ID" is listed as `i-0e1c706c2051d5121 (nagios-host-27)`. A list of steps is provided:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `AdvDevOps.pem`.
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 "AdvDevOps.pem"`
4. Connect to your instance using its Public DNS:  
`ec2-3-89-111-169.compute-1.amazonaws.com`

An "Example:" section shows the command: `ssh -i "AdvDevOps.pem" ec2-user@ec2-3-89-111-169.compute-1.amazonaws.com`. A note below states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username."

A "Cancel" button is located at the bottom right of the content area.

The footer of the console shows "cloudShell" and "Feedback" on the left, and copyright information "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences" on the right.

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

[illegible]

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.

```
[ec2-user@ip-172-31-83-157 ~]$ sudo yum update
Last metadata expiration check: 0:04:50 ago on Sat Sep 28 03:46:46 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

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
```

```
[ec2-user@ip-172-31-83-157 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:04:58 ago on Sat Sep 28 03:46:46 2024.
Dependencies resolved.
=====
Package                               Architecture      Version                               Repository      Size
=====
Installing:
  httpd                               x86_64            2.4.62-1.amzn2023.0.1               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.amzn2023.0.1               amazonlinux     1.4 M
  httpd-filesystem                   noarch            2.4.62-1.amzn2023.0.1               amazonlinux     14 k
  httpd-tools                        x86_64            2.4.62-1.amzn2023.0.1               amazonlinux     81 k
  libbrotli                           x86_64            1.0.9-4.amzn2023.0.2               amazonlinux     315 k
  libsodium                          x86_64            1.0.19-4.amzn2023.0.1               amazonlinux     176 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                   noarch            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
=====

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64
  generic-logos-httpd-18.0-0-12.amzn2023.0.3.noarch
  httpd-filesystem-2.4.62-1.amzn2023.0.3.noarch
  libsodium-1.0.19-4.amzn2023.0.3.x86_64
  mod_lua-2.0.27-1.amzn2023.0.3.x86_64
  php8.3-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-xsl-8.3.10-1.amzn2023.0.1.x86_64

  apr-util-1.6.3-1.amzn2023.0.1.x86_64
  httpd-2.4.62-1.amzn2023.0.1.x86_64
  httpd-tools-2.4.62-1.amzn2023.0.1.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  libxml2-1.1.34-5.amzn2023.0.2.x86_64
  mod_lua-2.4.62-1.amzn2023.0.1.x86_64
  php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-process-8.3.10-1.amzn2023.0.1.x86_64

  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  httpd-core-2.4.62-1.amzn2023.0.1.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
  php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-83-157 ~]$ sudo yum install gcc glibc glibc-common
```

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-172-31-83-157 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:05:58 ago on Sat Sep 28 03:46:46 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
=====

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
gcc-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-172-31-83-157 ~]$
```

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-172-31-83-157 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:11:36 ago on Sat Sep 28 03:46:46 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-filessystem                  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
glib2-devel                         x86_64            2.74.7-689.amzn2023.0.2            amazonlinux     486 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-filessystem-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-sans-vf-fonts-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-3.4-3.amzn2023.0.3.x86_64
libtiff-4.0.0-4.amzn2023.0.18.x86_64       libtiff-devel-4.0.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64        libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64        libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64  pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64    pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64        sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch     xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64       zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
[ec2-user@ip-172-31-83-157 ~]$
```



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
```

```
[ec2-user@ip-172-31-83-157 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

nagios

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

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-83-157 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-83-157 ~]$ |
```

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-172-31-83-157 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-83-157 ~]$
```

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

```
mkdir ~/downloads
```

```
cd ~/downloads
```

```
[ec2-user@ip-172-31-83-157 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-83-157 downloads]$ |
```

Step 14: Now we need to install the latest versions of nagios-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:fe7:45ce
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-172-31-83-157 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-28 04:06:15-- 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  5.22MB/s  in 0.5s
2024-09-28 04:06:16 (5.22 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-83-157 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-172-31-83-157 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/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/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-172-31-83-157 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`

```
[ec2-user@ip-172-31-83-157 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
```

Another **error** could be **Cannot find SSL headers**.

To solve this, we need to install OpenSSL Dev Library

Steps:

`sudo yum install openssl-devel`

```
checking for a BSD-compatible install... /usr/bin/install -c
[ec2-user@ip-172-31-83-157 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:21:59 ago on Sat Sep 28 03:46:46 2024.
Dependencies resolved.
=====
Package                                Repository                               Architecture      Size                Version
=====
Installing:
openssl-devel                          amazonlinux                               x86_64             3.0 M               1:3.0.8-1.amzn2023.0.14
=====
Transaction Summary
=====
Install 1 Package
Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y

```

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

```
[ec2-user@ip-172-31-83-157 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

```

```
*** 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-172-31-83-157 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-172-31-83-157 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios 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
```

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

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

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

Step 18: We need to update the email linked with this server to our email for it to send notifications (if any needed).

`sudo nano /usr/local/nagios/etc/objects/contacts.cfg`

```
GNU nano 5.8 /usr/local/nagios/etc/objects/contacts.cfg Modified
# CONTACTS.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 (defined above)
    alias              Nagios Admin         ; Full name of user
    email              2022.salram.konar@ves.ac.in ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****>
}

#####

# CONTACT GROUPS
#
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.
define contactgroup {
    contactgroup_name admins
    alias              Nagios Administrators
    members             nagiosadmin
}

#####

#####
```

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-172-31-83-157 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-172-31-83-157 nagios-4.5.5]$ |
```

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-172-31-83-157 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

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

`sudo service httpd restart`

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-83-157 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-83-157 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-172-31-83-157 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/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-172-31-83-157 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-172-31-83-157 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...
```

```
config.status: creating plugins-scripts/utils.sh
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-83-157 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

```
TI
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-172-31-83-157 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



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-172-31-90-224 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 452 (Check result path '/usr/local/nagios/var/spool/checkresults' is not a valid directory)
Error processing main config file!

[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo chmod 775 /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL
```

Error: Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 452 (Check result path '/usr/local/nagios/var/spool/checkresults' is not a valid directory)

It is an error in processing main config file!

Solution: Create the missing directory, set the permissions, verify it.

- `sudo mkdir -p /usr/local/nagios/var/spool/checkresults` (Create)
- `sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults`
- `sudo chmod 775 /usr/local/nagios/var/spool/checkresults` (permissions)

Now rerun the command

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

```
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-83-157 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

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

Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods

Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-83-157 nagios-plugins-2.4.11]$
```

Step 27:

sudo service nagios start

```
[ec2-user@ip-172-31-83-157 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-83-157 nagios-plugins-2.4.11]$ |
```

sudo systemctl status nagios

```
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo systemctl restart nagios
Failed to restart nagios.service: Unit nagios.service not found.
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo service nagios start
Reloading system: [ OK ]
Starting nagios (via systemctl): [ OK ]
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ sudo systemctl status nagios
nagios.service - LSB: Starts and stops the Nagios monitoring server
Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
Active: active (running) since Mon 2024-09-23 09:31:32 UTC; 3min 4s ago
Docs: man:systemd-sysv-generator(8)
Process: 66885 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
Tasks: 6 (limit: 1112)
Memory: 2.4M
CPU: 85ms
CGroup: /system.slice/nagios.service
├─66907 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
├─66909 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
├─66910 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
├─66911 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
├─66912 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
└─66913 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
Sep 23 09:33:21 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmppegZ2SN' for writing status data: Permissi>
Sep 23 09:33:24 ip-172-31-90-224.ec2.internal nagios[66907]: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;1;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 secon>
Sep 23 09:33:31 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpSrMcT9' for writing status data: Permissi>
Sep 23 09:33:41 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpuSNLk4' for writing status data: Permissi>
Sep 23 09:33:51 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpv8BENB' for writing status data: Permissi>
Sep 23 09:34:01 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpCvoHut' for writing status data: Permissi>
Sep 23 09:34:11 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpoVTuLF' for writing status data: Permissi>
Sep 23 09:34:21 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpHUEUI9' for writing status data: Permissi>
Sep 23 09:34:24 ip-172-31-90-224.ec2.internal nagios[66907]: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;2;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 secon>
Sep 23 09:34:31 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpppy1CO9' for writing status data: Permissi>
lines 1-26/26 (END)
[ec2-user@ip-172-31-90-224 nagios-plugins-2.0.3]$ ls -ld /usr/local/nagios/var
```

Error:Sep 23 09:34:31 ip-172-31-90-224.ec2.internal nagios[66907]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpppy1CO9' for writing status data: Permissi>

Solution:

- ls -ld /usr/local/nagios/var
- sudo chown -R nagios:nagios /usr/local/nagios/var
- sudo chmod -R 755 /usr/local/nagios/var
- sudo systemctl restart nagios

Rerun the command

sudo systemctl status nagios

```

[ec2-user@ip-172-31-83-157 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-09-28 04:18:59 UTC; 12s ago
     Docs: https://www.nagios.org/documentation
   Process: 66052 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 66053 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 66054 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 5.6M
      CPU: 77ms
   CGroup: /system.slice/nagios.service
           └─66054 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             66055 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             66056 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             66057 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             66058 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             66059 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: qh: core query handler registered
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: qh: echo service query handler registered
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: qh: help for the query handler registered
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: wproc: Registry request: name=Core Worker 66057;pid=66057
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: wproc: Registry request: name=Core Worker 66058;pid=66058
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: wproc: Registry request: name=Core Worker 66056;pid=66056
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: wproc: Registry request: name=Core Worker 66055;pid=66055
Sep 28 04:18:59 ip-172-31-83-157.ec2.internal nagios[66054]: Successfully launched command file worker with pid 66059
[ec2-user@ip-172-31-83-157 nagios-plugins-2.4.11]$

```

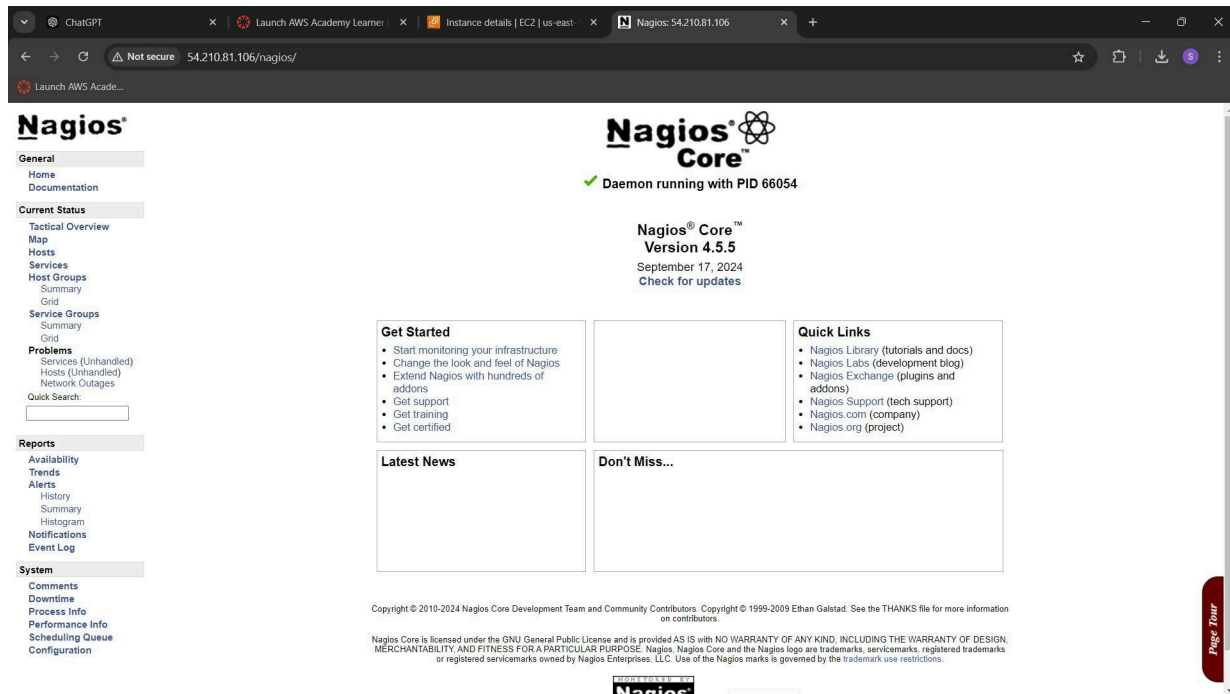
Step 28: Now, go to EC2 instance and click on instance id. Then, click on the copy icon just before the public ip address on public IP.

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, and various instance-related services. The main content area displays the 'Instance details' for the instance 'i-0e1c706c2051d5121' (nagios-host-27). The instance is in the 'Running' state. Key details include:

- Instance ID:** i-0e1c706c2051d5121 (nagios-host-27)
- Public IPv4 address:** 3.89.111.169 (with a copy icon)
- Private IPv4 address:** 172.31.90.224
- Public IPv4 DNS:** ec2-3-89-111-169.compute-1.amazonaws.com (with a copy icon)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-90-224.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-05537c9b6c7862468
- Subnet ID:** subnet-0188b4375dc68c2fb
- Instance ARN:** arn:aws:ec2:us-east-1:196908236276:instance/i-0e1c706c2051d5121
- Platform:** Amazon Linux (Inferred)
- Platform details:** Linux/UNIX
- Stop protection:** Disabled
- Launch time:** Mon Sep 23 2024 14:13:55 GMT+0530 (India Standard Time) (about 1 hour)
- AMI ID:** ami-0ebfd941bbafe70c6
- AMI name:** al2023-ami-2023.5.20240916.0-kernel-6.1-x86\_64
- Monitoring:** disabled
- Termination protection:** Disabled
- AMI location:** amazon/al2023-ami-2023.5.20240916.0-kernel-6.1-x86\_64

The bottom of the console shows the footer with copyright information for Amazon Web Services, Inc. or its affiliates, and links to Privacy, Terms, and Cookie preferences.

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 configure Nagios Core, Nagios Plugins and NRPE on a Linux machine. We are using an Amazon Linux OS instance configured with the need security rules. We need to make sure that the Nagios-core and Nagios-plugins links that are used are the ones which are up-to dat (wget commands).It is needed to extract and configure these files so that no issues are detected while starting the server. Once all the setup is complete, we can start the nagios server. Using the public IP address of the EC2 instance, we can access the nagios dashboard by navigating that IP to nagios.