Experiment 9

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- Helps you to plan for infrastructure upgrades before outdated systems create failures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

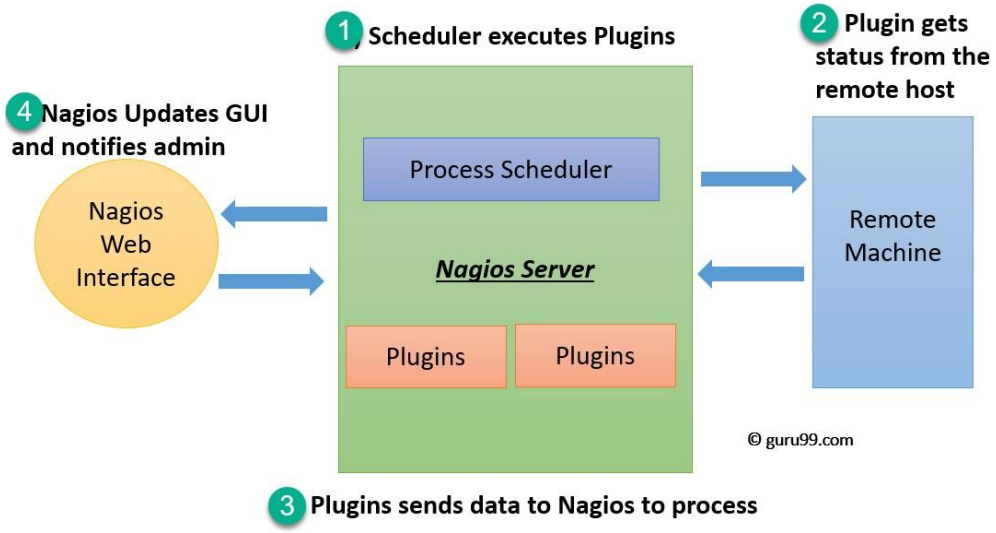
Following are the important features of Nagios monitoring tool:

- 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.



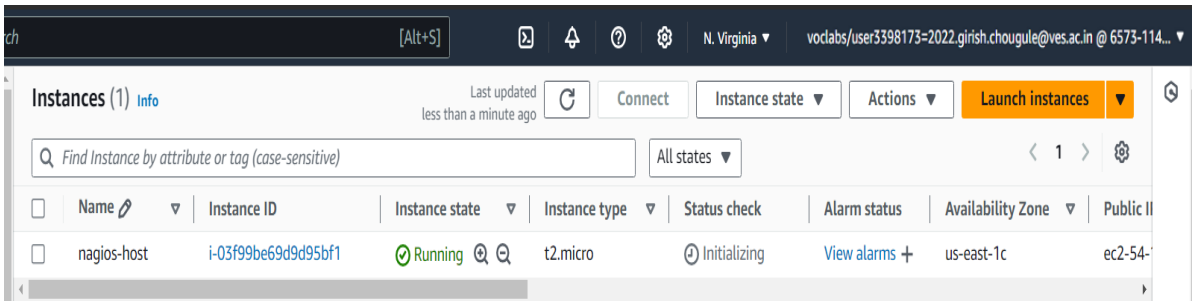
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute theplugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Installation of Nagios

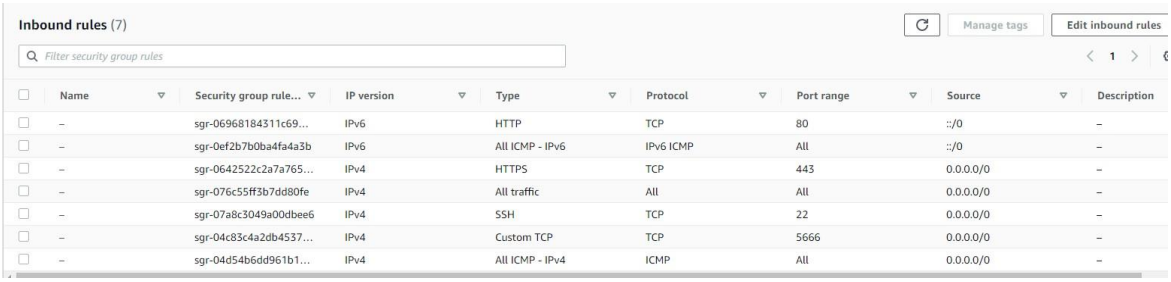
Prerequisites: AWS Free Tier

Steps:

1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host

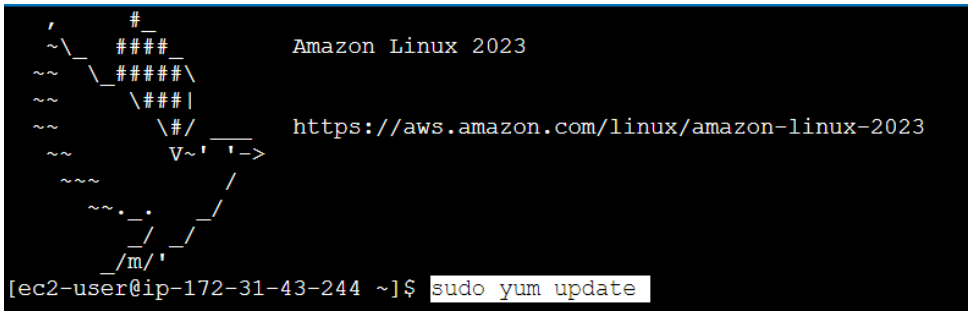


2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.



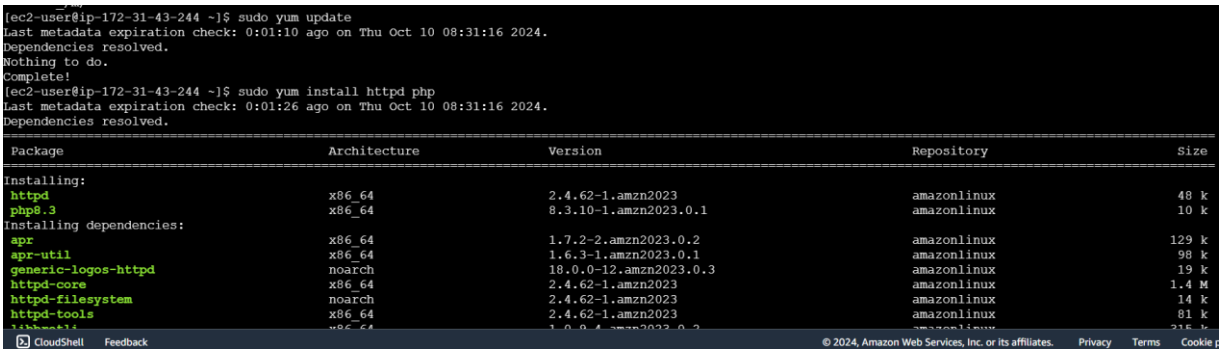
You have to edit the inbound rules of the specified Security Group for this.

3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.



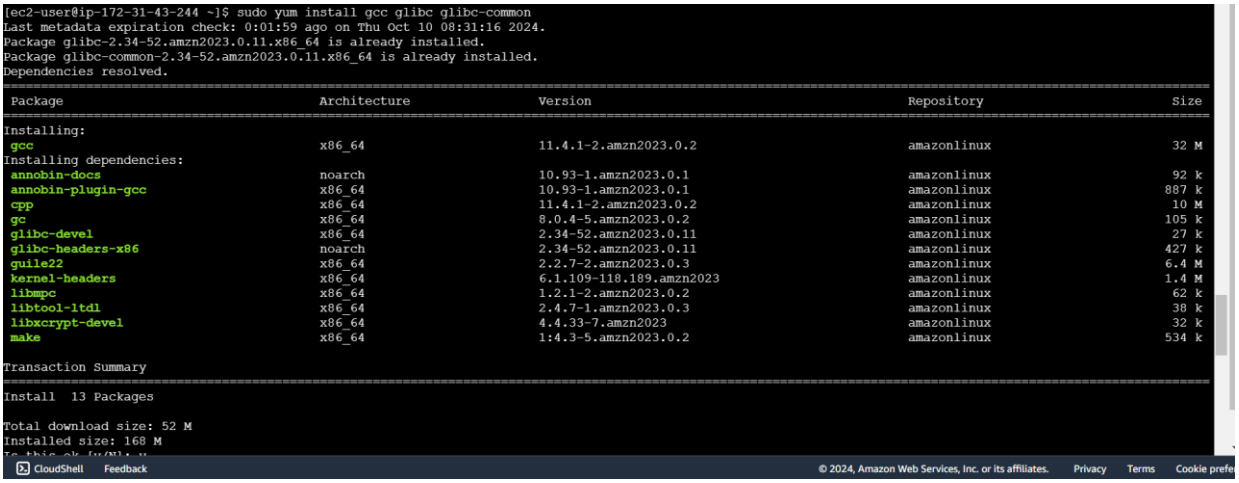
4. Update the package indices and install the following packages using yum

sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-commonsudo yum install gd
gd-devel



```
[ec2-user@ip-172-31-43-244 ~]$ sudo yum update
Last metadata expiration check: 0:01:10 ago on Thu Oct 10 08:31:16 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-43-244 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:26 ago on Thu Oct 10 08:31:16 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
php8.3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	10 k
Installing dependencies:				
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux	81 k

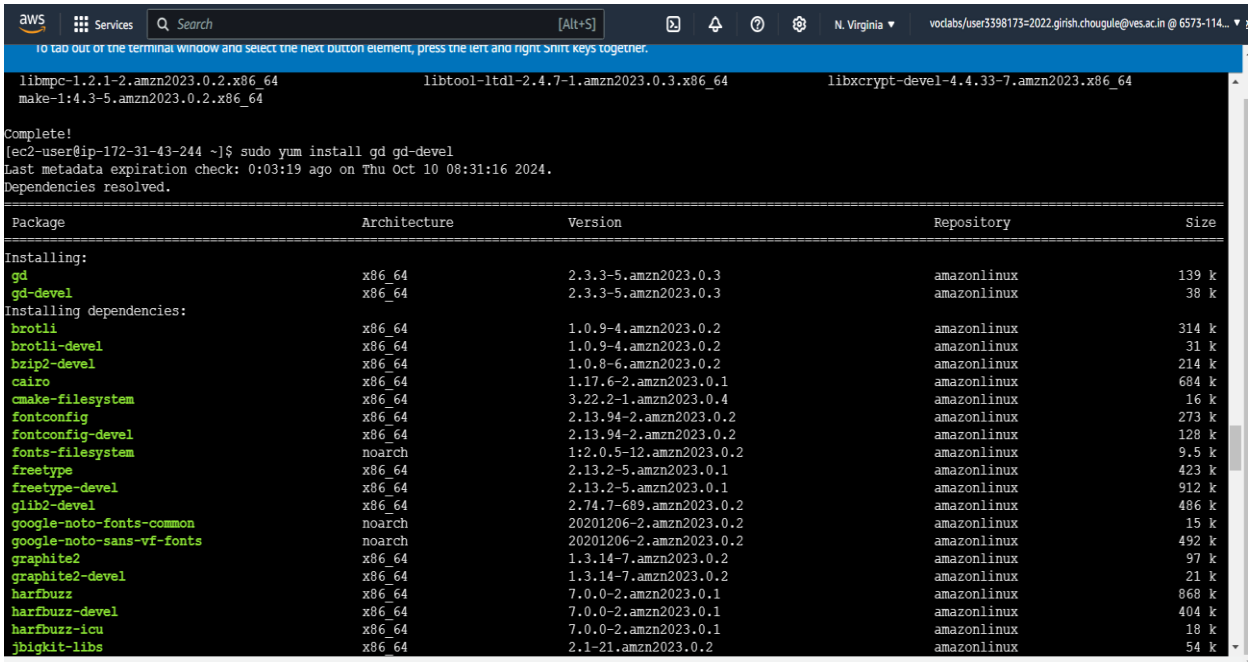


```
[ec2-user@ip-172-31-43-244 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:01:59 ago on Thu Oct 10 08:31:16 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

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

```
Transaction Summary
Install  13 Packages

Total download size: 52 M
Installed size: 168 M
```


```
  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-172-31-43-244 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:19 ago on Thu Oct 10 08:31:16 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
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486 k
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15 k
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492 k
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97 k
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21 k
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868 k
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404 k
harfbuzz-icu	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18 k
jbigkit-libs	x86_64	2.1-21.amzn2023.0.2	amazonlinux	54 k

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

```
sudo adduser -m nagios
sudo passwd nagios
```

```
[ec2-user@ip-172-31-43-244 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-43-244 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

6. Create a new user group

   sudo groupadd nagcmd

7. Use these commands so that you don't have to use sudo for Apache and Nagios

   sudo usermod -a -G nagcmd nagiossudo
   usermod -a -G nagcmd apache

8. Create a new directory for Nagios downloads

   mkdir ~/downloadscd
   ~/downloads

```
[ec2-user@ip-172-31-43-244 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-43-244 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-43-244 ~]$ mkdir ~/downloads
cd ~/downloads
```

9. Use wget to download the source zip files.

   wget
   http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz

   wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz

10. Use tar to unzip and change to that directory.

    tar zxvf nagios-4.0.8.tar.gz

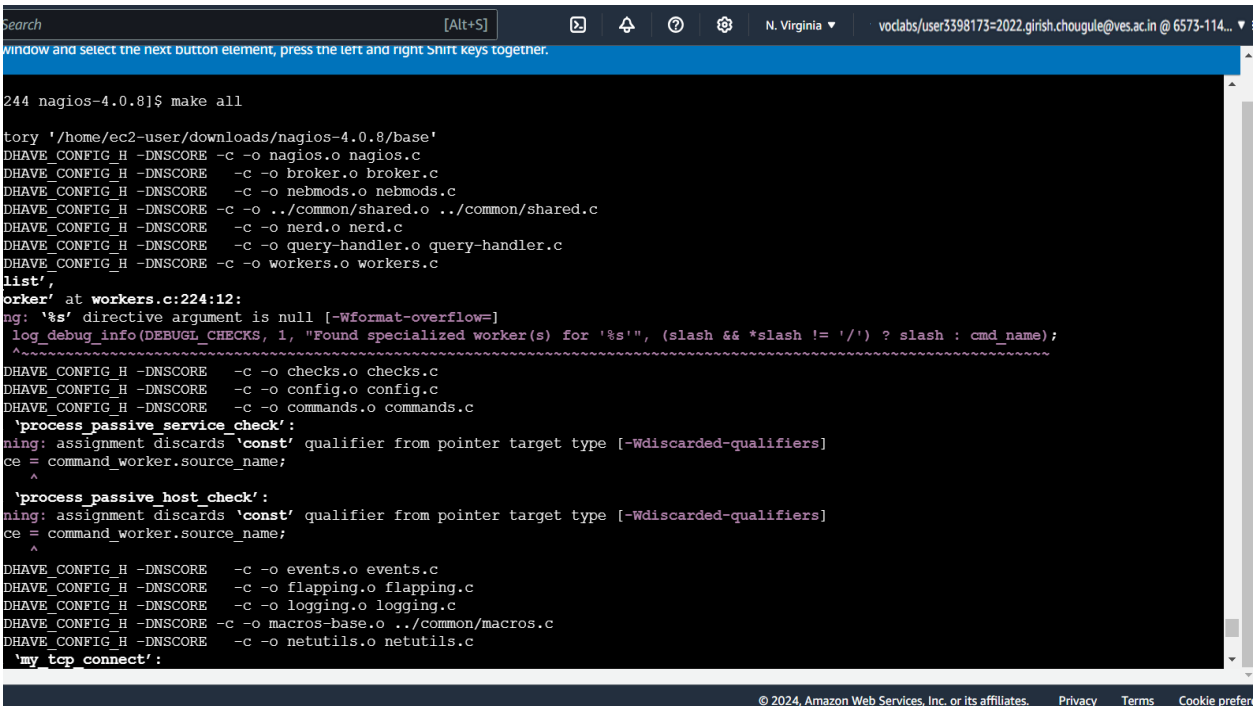
11. Run the configuration script with the same group name you previously created.

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

```
[ec2-user@ip-172-31-43-244 downloads]$ cd nagios-4.0.8
[ec2-user@ip-172-31-43-244 nagios-4.0.8]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
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
checking whether make sets $(MAKE)... yes
```

12. Compile the source code.

    make all



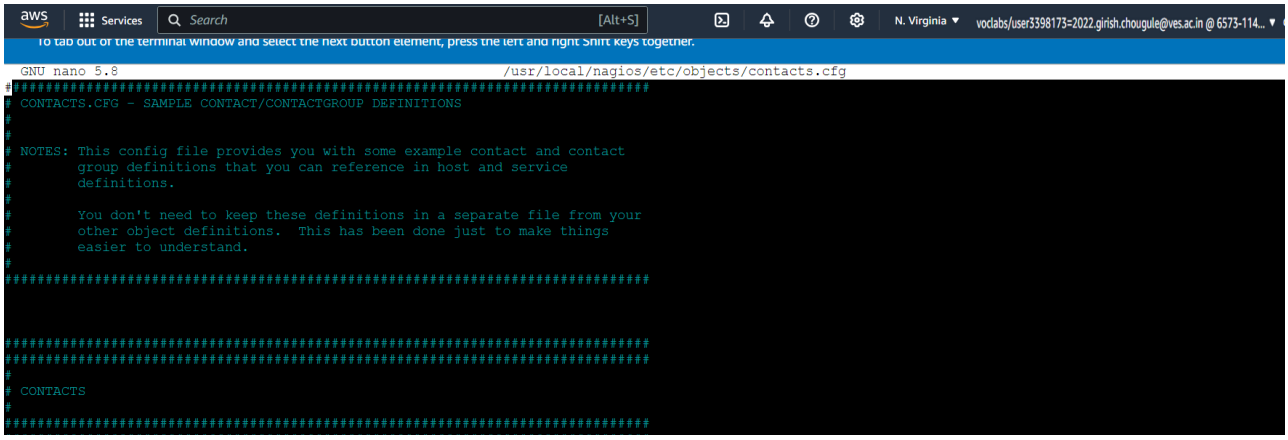
13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

    sudo make install sudo

    make install-init

    sudo make install-config sudo

    make install-commandmode



14. Edit the config file and change the email address.

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

```
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.mohammed.ansari@ves.ac.in    ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
        }
```

15. Configure the web interface.

    sudo make install-webconf

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

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

```
[ec2-user@ip-172-31-43-244 nagios-4.0.8]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf

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

[ec2-user@ip-172-31-43-244 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

17. Restart Apache

    sudo service httpd restart

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

    cd ~/downloads
    tar zxvf nagios-plugins-2.0.3.tar.gz

```
[ec2-user@ip-172-31-43-244 nagios-4.0.8]$ cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
nagios-plugins-2.0.3/
nagios-plugins-2.0.3/perlmods/
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.in
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile.am
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.0.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.0.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.0.3/perlmods/Makefile
nagios-plugins-2.0.3/perlmods/Perl-OSType-1.003.tar.gz
nagios-plugins-2.0.3/perlmods/install_order
nagios-plugins-2.0.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.0.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.0.3/perlmods/Module-Build-0.4007.tar.gz
```

19. Compile and install plugins

    cd nagios-plugins-2.0.3
    ./configure --with-nagios-user=nagios --with-nagios-group=nagiosmake
    sudo make install

```
[ec2-user@ip-172-31-43-244 downloads]$ cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
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 to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
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
checking for style of include used by make... GNU
```

20. Start Nagios

Add Nagios to the list of system services

sudo chkconfig --add nagiossudo
chkconfig nagios on

Verify the sample configuration files

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

If there are no errors, you can go ahead and start Nagios.

sudo service nagios start

```
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagios /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-43-244 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...
   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.
```

21. Check the status of Nagios

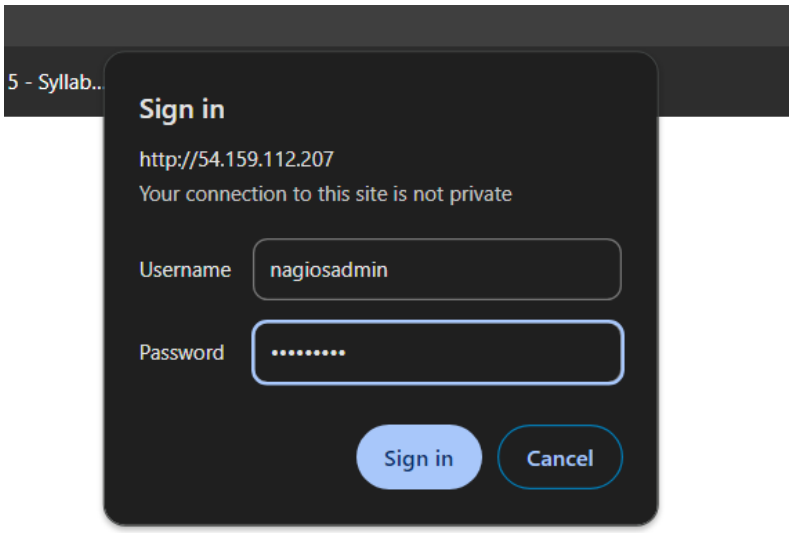
sudo systemctl status nagios

```
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$ sudo systemctl start nagios
[ec2-user@ip-172-31-43-244 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 Thu 2024-10-10 08:54:46 UTC; 5s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 66392 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
      Tasks: 5 (limit: 1112)
     Memory: 1.4M
        CPU: 37ms
     CGroup: /system.slice/nagios.service
             ├─66414 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─66416 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66417 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66418 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─66419 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh

Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: nerd: Channel hostchecks registered successfully
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: nerd: Channel servicechecks registered successfully
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: nerd: Channel opathchecks registered successfully
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: nerd: Fully initialized and ready to rock!
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: wproc: Successfully registered manager as @wproc with query handler
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: wproc: Registry request: name=Core Worker 66419;pid=66419
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: wproc: Registry request: name=Core Worker 66418;pid=66418
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: wproc: Registry request: name=Core Worker 66417;pid=66417
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: wproc: Registry request: name=Core Worker 66416;pid=66416
Oct 10 08:54:46 ip-172-31-43-244.ec2.internal nagios[66414]: Warning: Could not open object cache file '/usr/local/nagios/var/objects.cache' for writing!
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$
```

22. Go back to EC2 Console and copy the Public IP address of this instance

23. Open up your browser and look for **http://<your_public_ip_address>/nagios**



Enter username as nagiosadmin and password which you set in Step 16.

24. After entering the correct credentials, you will see this page.

This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion:

Thus, we learned about Nagios and successfully set it up as a host on our Amazon Linux machine.