MONITORING WITH NAGIOS

Nagios: Nagios is an open source continuous monitoring tool which monitors network, applications and servers. It can find and repair problems detected in the infrastructure, and stop future issues before they affect the end users. It gives the complete status of your IT infrastructure and its performance.

Why Nagios?

It can monitor Database servers such as SQL Server, Oracle, Mysql, Postgres
It gives application level information (Apache, Postfix, etc.).
Provides active development.
Has excellent support form huge active community.
Nagios runs on any operating system.

Benefits of Nagios

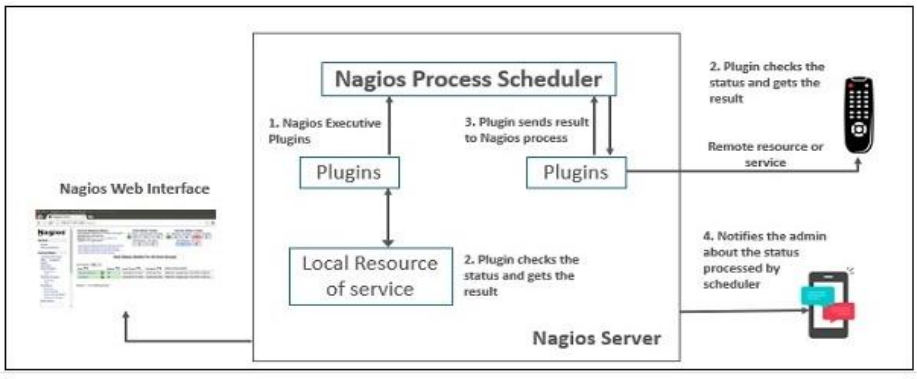
Nagios offers the following benefits for the users −

- It helps in getting rid of periodic testing.
- It reduces maintenance cost without sacrificing performance.
- It provides timely notification to the management of control and breakdown.

Nagios Architecture

The following points are worth notable about Nagios architecture −

- Nagios has server-agent architecture.
- Nagios server is installed on the host and plugins are installed on the remote hosts/servers which are to be monitored.
- Nagios sends a signal through a process scheduler to run the plugins on the local/remote hosts/servers.
- Plugins collect the data (CPU usage, memory usage etc.) and sends it back to the scheduler.
- Then the process schedules send the notifications to the admin/s and updates Nagios GUI.



Nagios Core (one of the nagios product which is absolutely free)

It is the core on monitoring IT infrastructure. Nagios XI product is also fundamentally based on Nagios core. Whenever there is any issue of failure in the infrastructure, it sends an alert/notification to the admin who can take the action quickly to resolve the issue. This tool is absolutely free.

Nagios XI, Nagios Log Server, Nagios Fusion, Nagios Network Analyser are also nagios products that are paid.

Nagios Plugins

Plugins helps to monitor databases, operating systems, applications, network equipment, protocols with Nagios. Plugins are compiled executables or script (Perl or non-Perl) that extends Nagios functionality to monitor servers and hosts. Nagios will execute a Plugin to check the status of a service or host. Nagios can be compiled with support for an embedded Perl interpreter to execute Perl plugins. Without it, Nagios executes Perl and non-Perl plugins by forking and executing the plugins as an external command.

Types of Nagios Plugins

Nagios has the following plugins available in it −

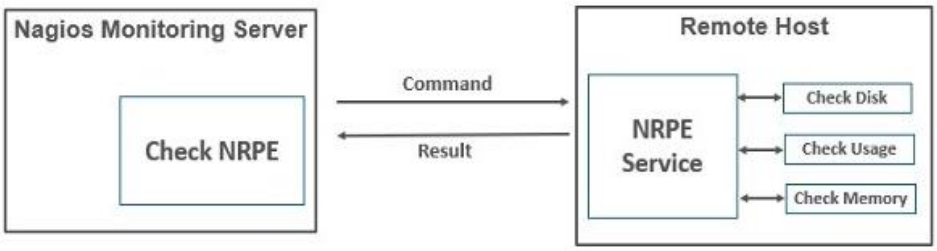
Official Nagios Plugins − There are 50 official Nagios Plugins. Official Nagios plugins are developed and maintained by the official Nagios Plugins Team.

Community Plugins − There are over 3,000 third party Nagios plugins that have been developed by hundreds of Nagios community members.

Custom Plugins − You can also write your own Custom Plugins. There are certain guidelines that must be followed to write Custom Plugins.

Nagios-NRPE

The Nagios daemon which run checks on remote machines in NRPE (Nagios Remote Plugin Executor). It allows you to run Nagios plugins on other machines remotely. You can monitor remote machine metrics such as disk usage, CPU load etc. It can also check metrics of remote windows machines through some windows agent addons.



Hosts and services

Nagios is the most popular tool which is used to monitor hosts and services running in our IT infrastructure. Hosts and service configurations are the building blocks of Nagios Core.

- Host is just like a computer; it can be a physical device or virtual.
- Services are those which are used by Nagios to check something about a host.

Features

- Nagios Core is open source, hence free to use.
- Powerful monitoring engine which can scale and manage 1000s of hosts and servers.
- Fast alerting system, sends alerts to admins immediately after any issue is identified.
- Multiple plugins available to support Nagios, custom coded plugins can also be used with Nagios.
- It has good log and database system storing everything happening on the network with ease.

Applications

- Monitor host resources such as disk space, system logs etc.
- Monitor network resources – http, ftp, smtp, ssh etc.
- Monitor windows/linux/unix/web applications and its state.
- Send alerts/notifications.
- via email, sms, pager of any issue on infrastructure.
- Recommending when to upgrade the IT infrastructure.

Installation of Nagios

Step By Step method for installing Nagios in Amazon Linux

Step 1: Install Prerequisite Software

Step 2: Create Account Information

Step 3: Download Nagios Core and the Plugins

Step 4: Compile and Install Nagios

Step 5: Customize Configuration

Step 6: Configure the Web Interface

Step 7: Compile and Install the Nagios Plugins

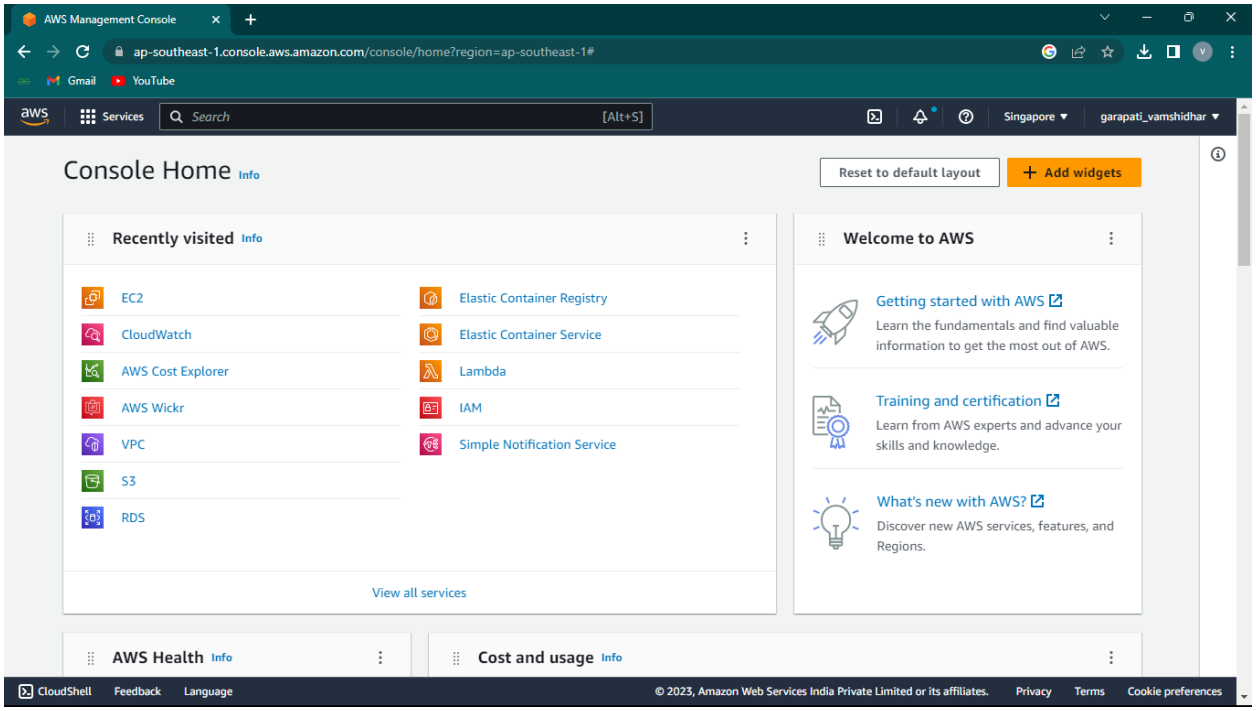
Step 8: Start Nagios

Step 9: Update AWS Security Group

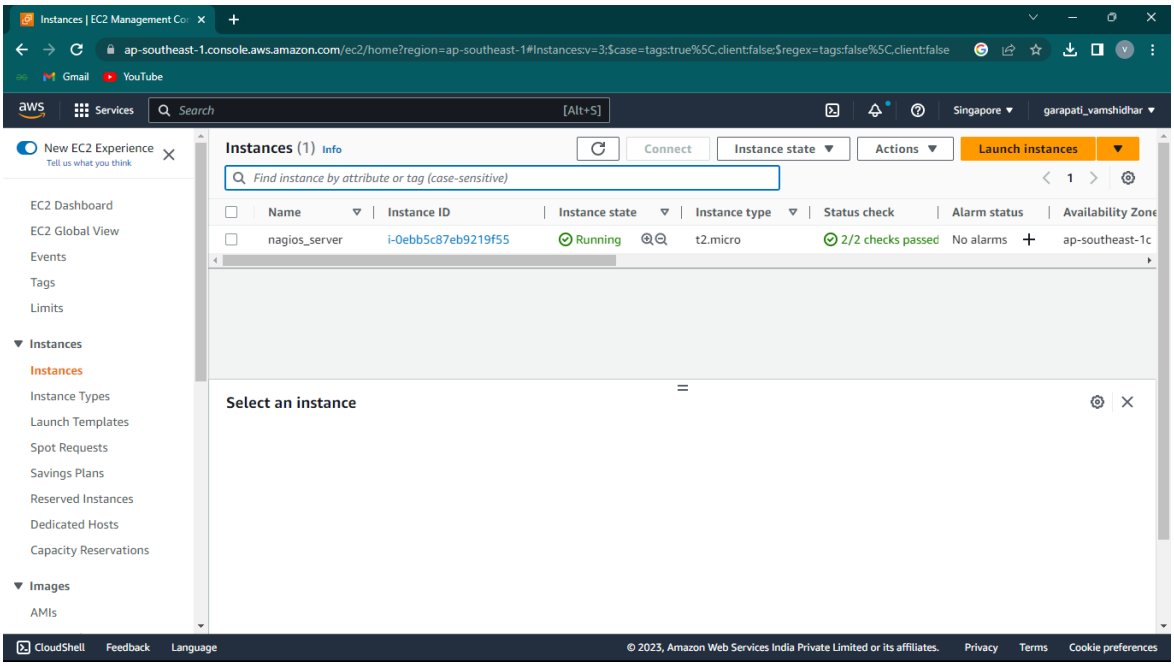
Step 10: Log in to the Web Interface.

To Start Nagios Core Installation you must have your EC2 instance up and running and have already configured SSH access to the instance.
For this we have to launch EC2 instance.

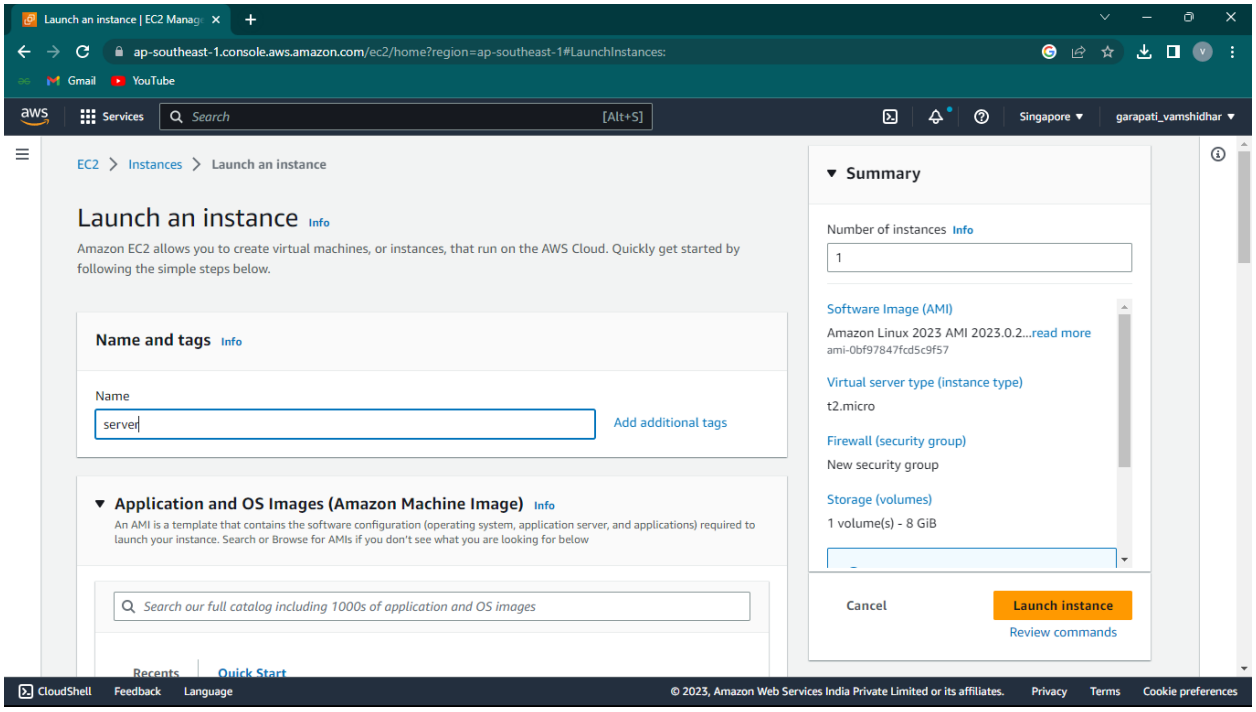
1.First Login to AWS console page.



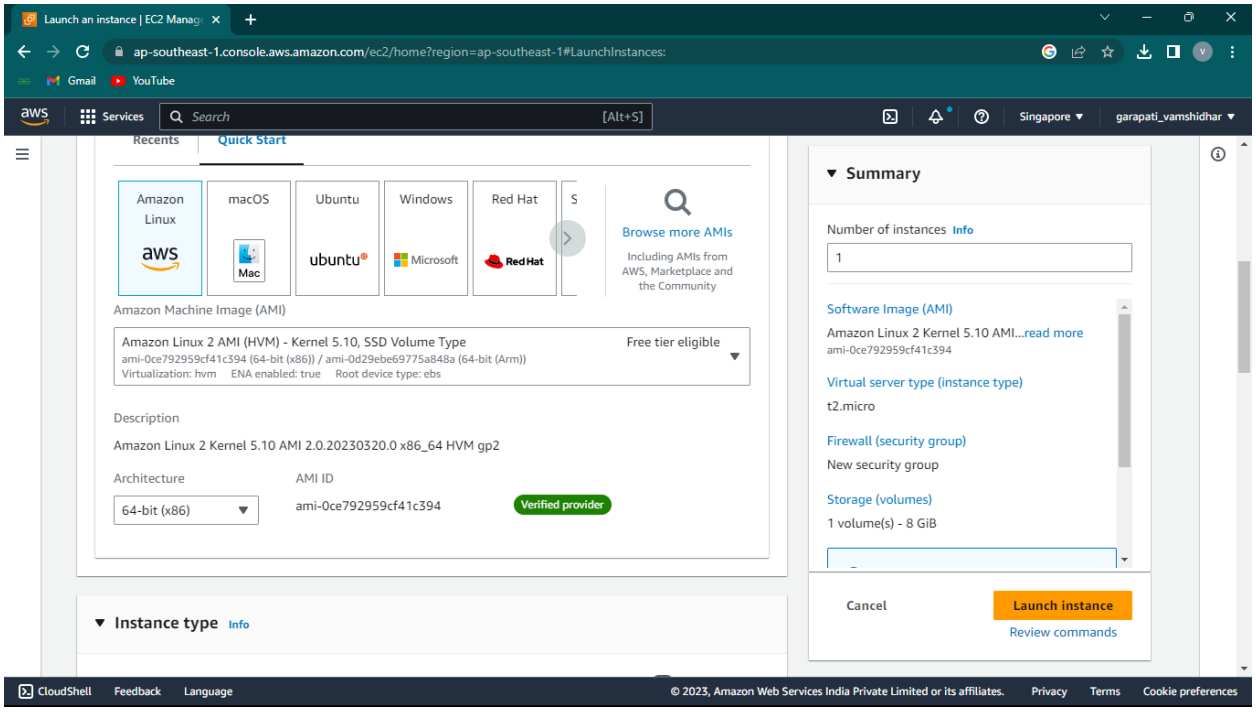
2.Click on Launch instances to launch instance.



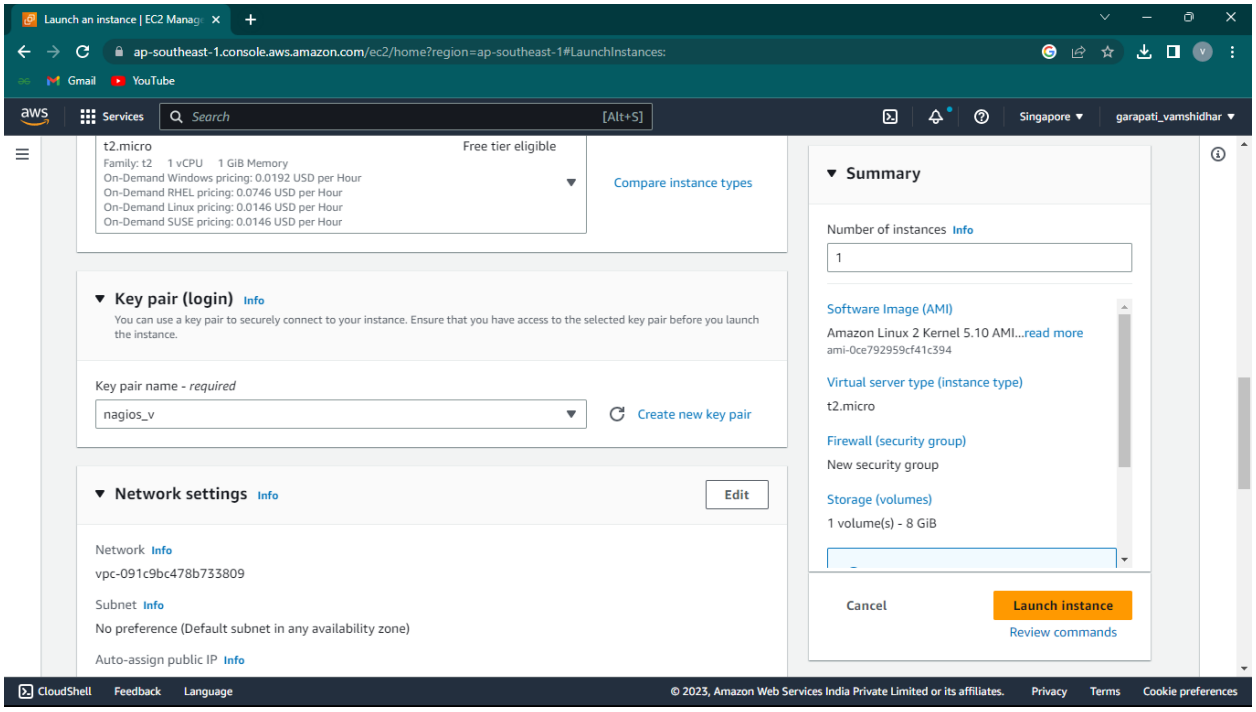
3.Give name to your instance.



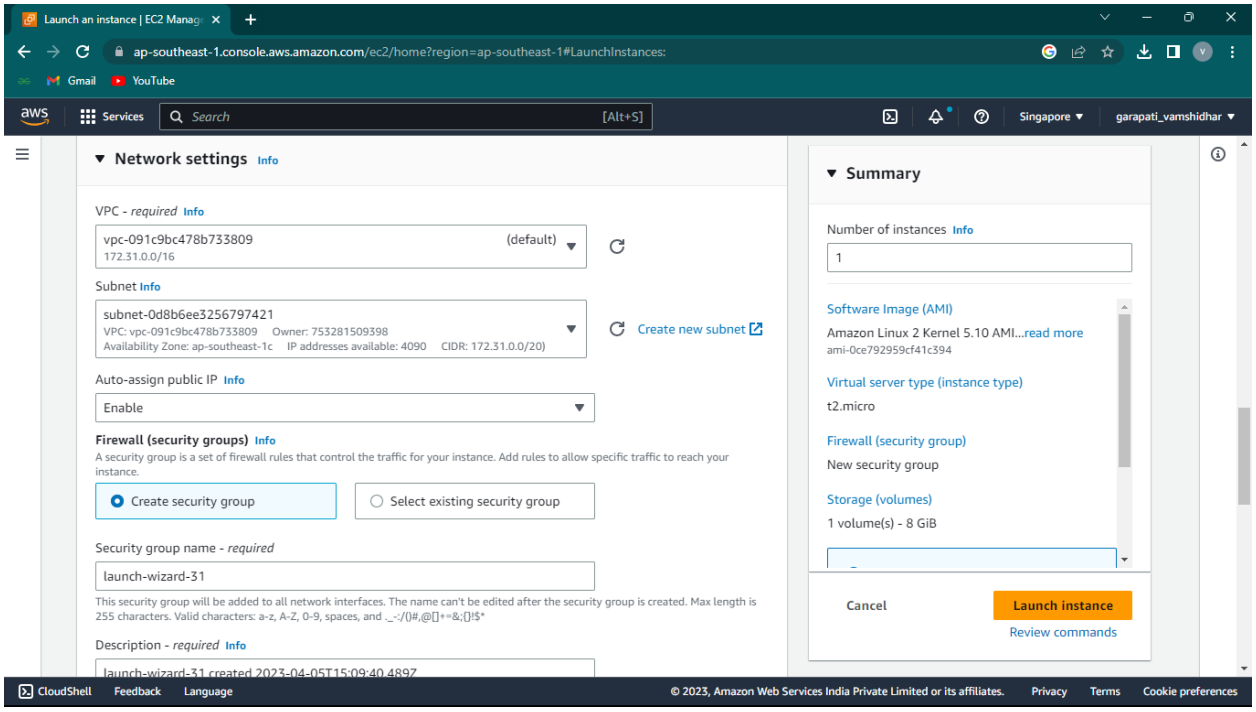
4.Select Amazon linux as Amazon Machine Image.



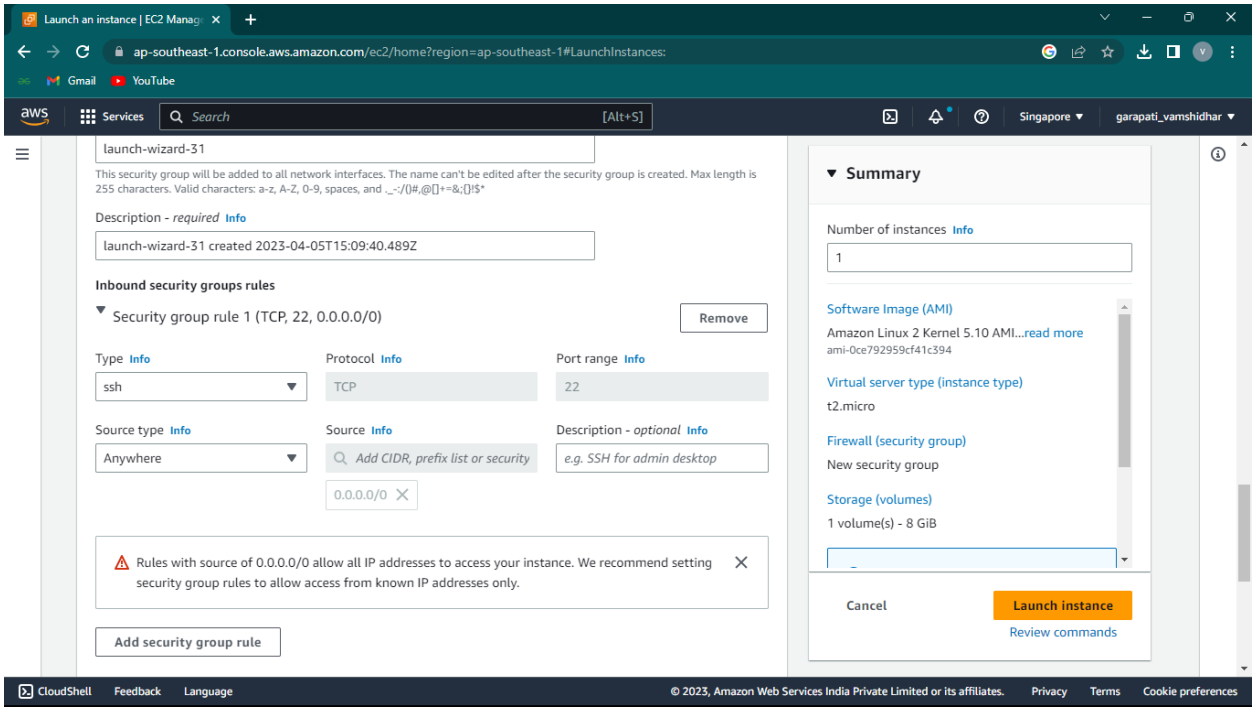
5.Create new key pair in pem format.



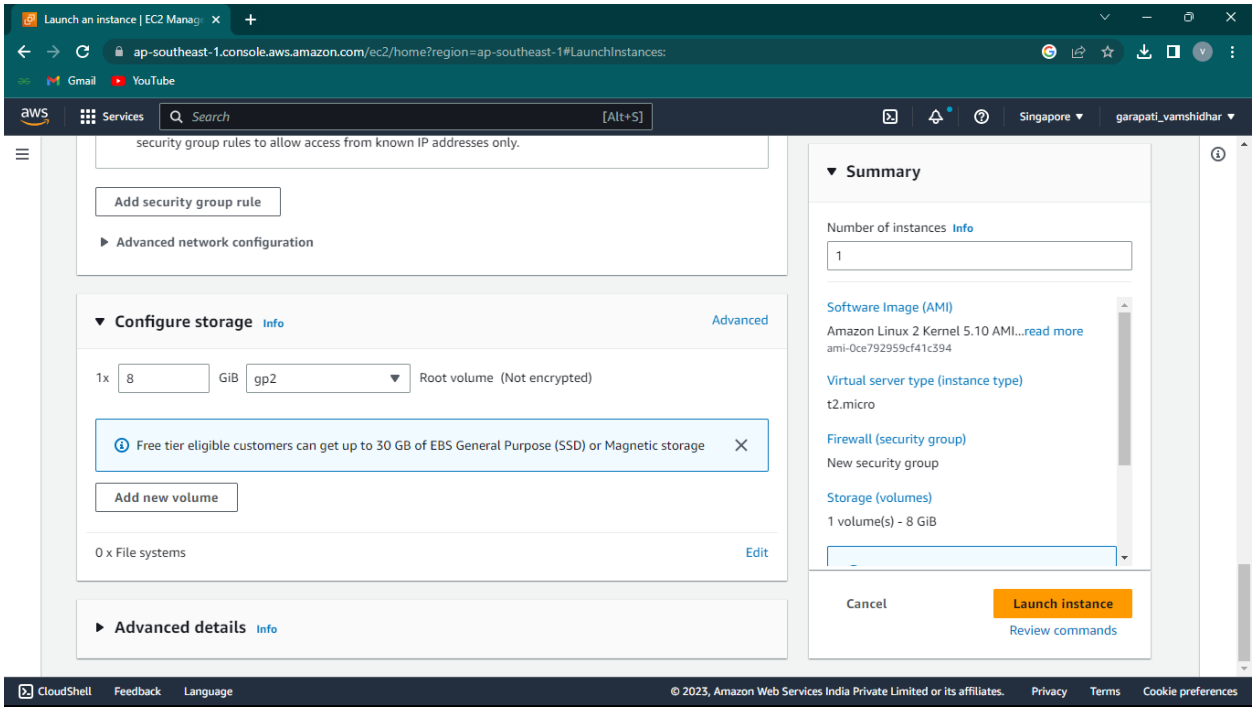
6.In network settings use default VPC and default subnet.



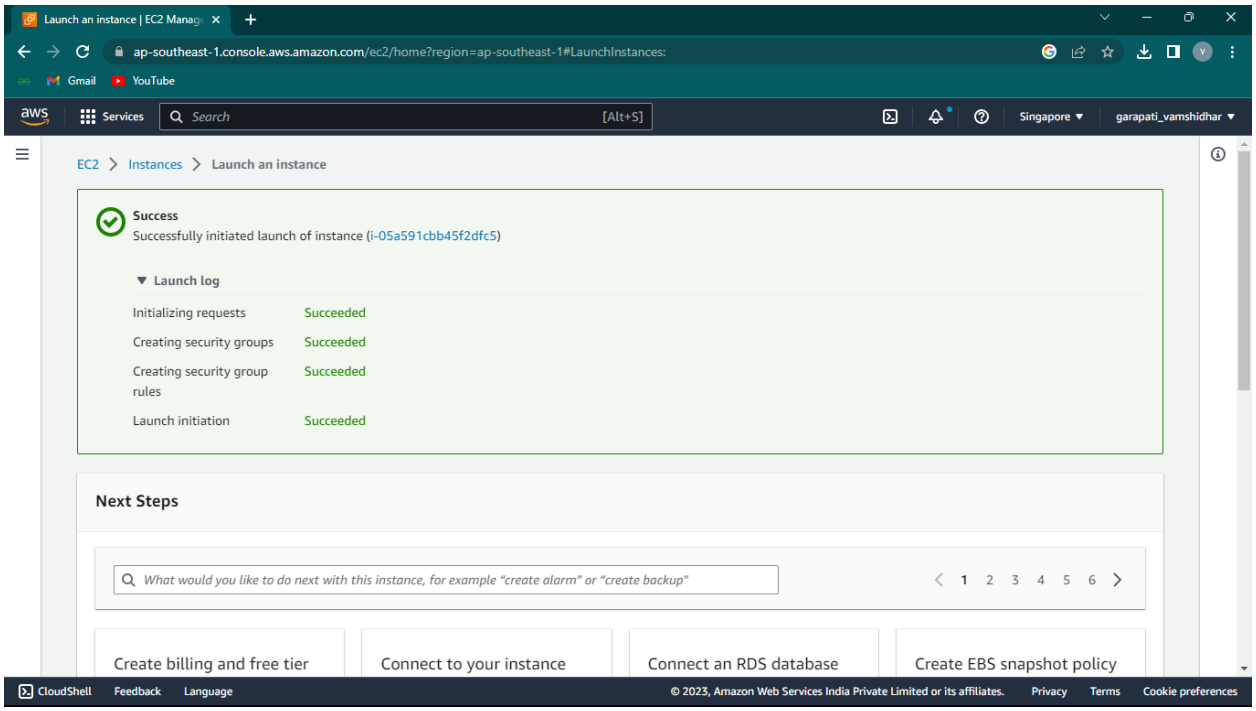
7.Create new security group and edit inbound rules as ssh type with source type as anywhere.



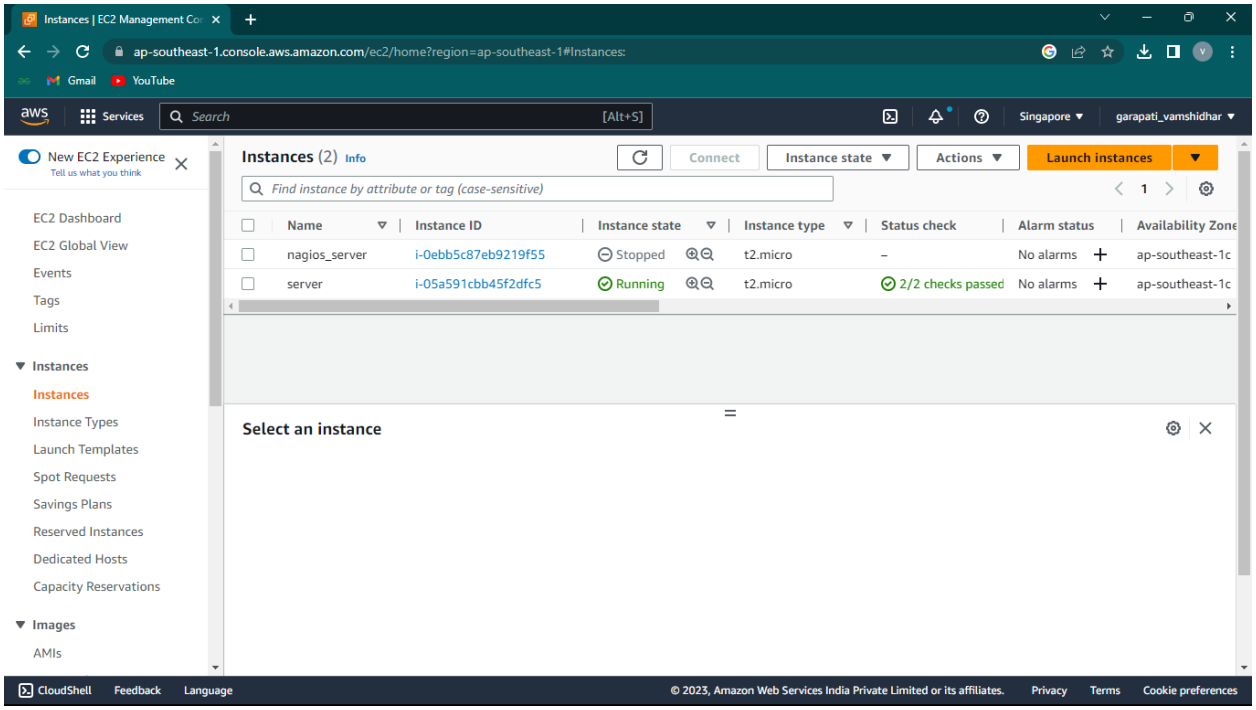
8.Click on launch instance to launch the instance.



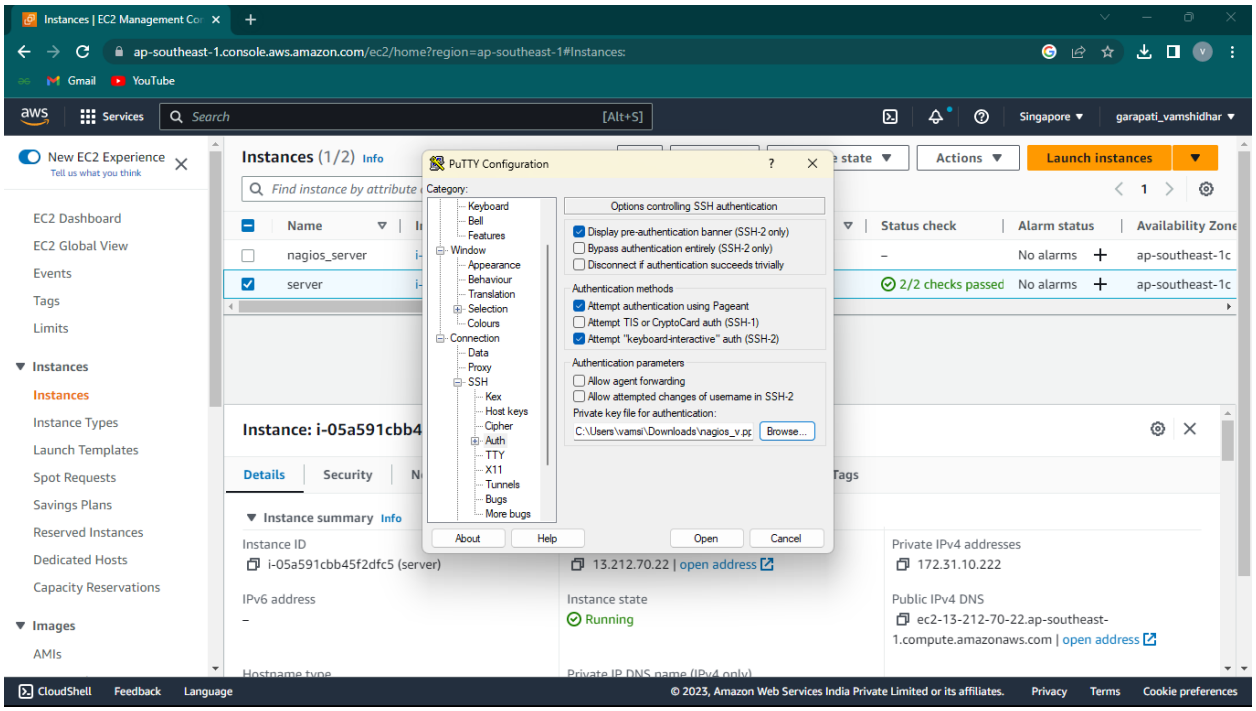
9.Instance is launched successfully.



10.The instance is in running state



11.Connect to EC2 instance using puttygen.



Installation steps(putty)

Step 1: Install Prerequisite Software

Nagios requires the following packages are installed on your server prior to installing Nagios:

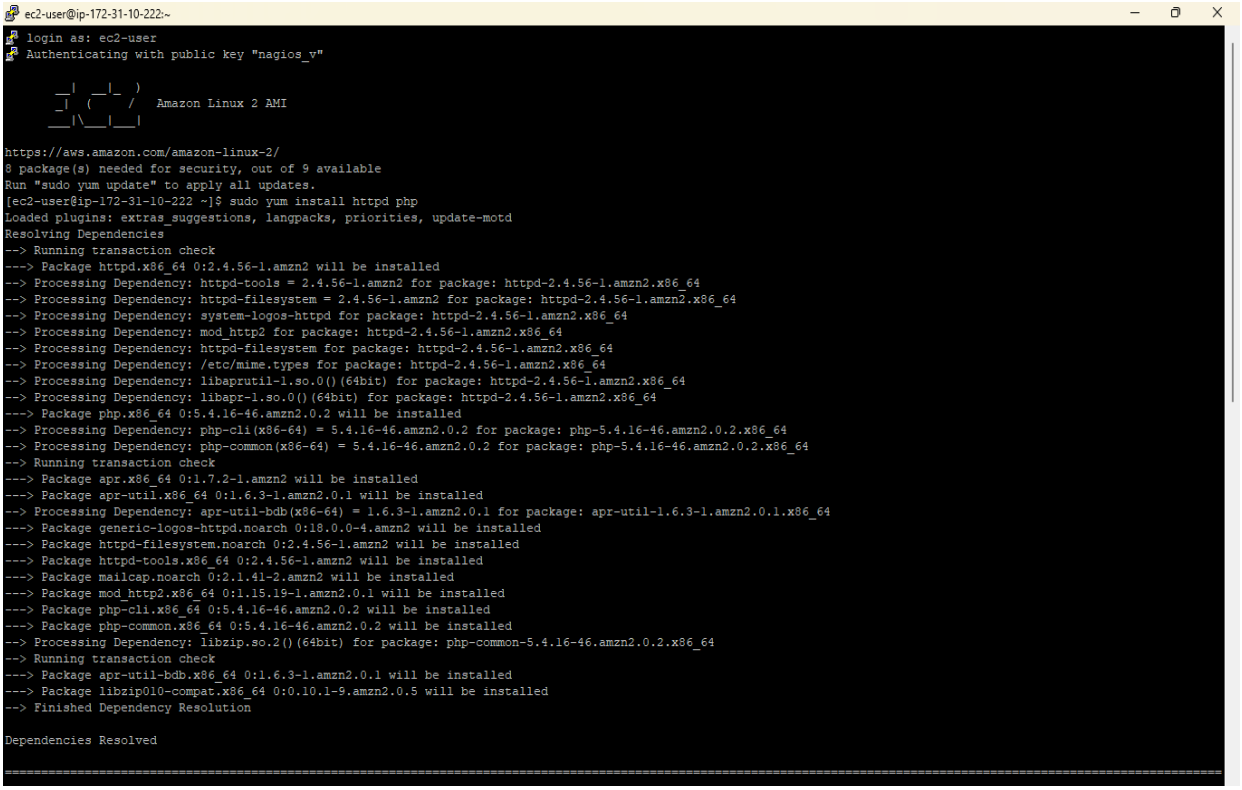
* Apache
* PHP
* GCC compiler
* GD development libraries

You can use yum to install these packages by running the following commands (as ec2-user):

$sudo yum install httpd php

$sudo yum install gcc glibc glibc-common

$sudo yum install gd gd-devel




```
Complete!
[ec2-user@ip-172-31-10-222 ~]$ sudo yum install gcc glibc glibc-common
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package glibc-2.26-62.amzn2.x86_64 already installed and latest version
Package glibc-common-2.26-62.amzn2.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package gcc.x86_64 0:7.3.1-15.amzn2 will be installed
--> Processing Dependency: cpp = 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libsanitizer >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libquadmath >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libmpx >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libitm >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libcilkrts >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libatomic >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libmpfr.so.4()(64bit) for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libmpc.so.3()(64bit) for package: gcc-7.3.1-15.amzn2.x86_64
--> Running transaction check
---> Package cpp.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package glibc-devel.x86_64 0:2.26-62.amzn2 will be installed
--> Processing Dependency: glibc-headers = 2.26-62.amzn2 for package: glibc-devel-2.26-62.amzn2.x86_64
--> Processing Dependency: glibc-headers for package: glibc-devel-2.26-62.amzn2.x86_64
---> Package libatomic.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package libcilkrts.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package libitm.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package libmpc.x86_64 0:1.0.1-3.amzn2.0.2 will be installed
---> Package libmpx.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package libquadmath.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package libsanitizer.x86_64 0:7.3.1-15.amzn2 will be installed
---> Package mpfr.x86_64 0:3.1.1-4.amzn2.0.2 will be installed
--> Running transaction check
```

```
Complete!
[ec2-user@ip-172-31-10-222 ~]$ sudo yum install gd gd-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package gd.x86_64 0:2.0.35-27.amzn2 will be installed
--> Processing Dependency: libfontconfig.so.1()(64bit) for package: gd-2.0.35-27.amzn2.x86_64
--> Processing Dependency: libXpm.so.4()(64bit) for package: gd-2.0.35-27.amzn2.x86_64
--> Processing Dependency: libX11.so.6()(64bit) for package: gd-2.0.35-27.amzn2.x86_64
---> Package gd-devel.x86_64 0:2.0.35-27.amzn2 will be installed
--> Processing Dependency: zlib-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Processing Dependency: libpng-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Processing Dependency: libjpeg-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Processing Dependency: libXpm-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Processing Dependency: libX11-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Processing Dependency: freetype-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Processing Dependency: fontconfig-devel for package: gd-devel-2.0.35-27.amzn2.x86_64
--> Running transaction check
---> Package fontconfig.x86_64 0:2.13.0-4.3.amzn2 will be installed
--> Processing Dependency: fontpackages-filesystem for package: fontconfig-2.13.0-4.3.amzn2.x86_64
--> Processing Dependency: dejavu-sans-fonts for package: fontconfig-2.13.0-4.3.amzn2.x86_64
---> Package fontconfig-devel.x86_64 0:2.13.0-4.3.amzn2 will be installed
--> Processing Dependency: pkgconfig(uuid) for package: fontconfig-devel-2.13.0-4.3.amzn2.x86_64
--> Processing Dependency: pkgconfig(expat) for package: fontconfig-devel-2.13.0-4.3.amzn2.x86_64
---> Package freetype-devel.x86_64 0:2.8-14.amzn2.1.1 will be installed
---> Package libX11.x86_64 0:1.6.7-3.amzn2.0.2 will be installed
--> Processing Dependency: libX11-common >= 1.6.7-3.amzn2.0.2 for package: libX11-1.6.7-3.amzn2.0.2.x86_64
--> Processing Dependency: libxcb.so.1()(64bit) for package: libX11-1.6.7-3.amzn2.0.2.x86_64
---> Package libX11-devel.x86_64 0:1.6.7-3.amzn2.0.2 will be installed
--> Processing Dependency: pkgconfig(xcb) >= 1.11.1 for package: libX11-devel-1.6.7-3.amzn2.0.2.x86_64
--> Processing Dependency: pkgconfig(xproto) for package: libX11-devel-1.6.7-3.amzn2.0.2.x86_64
--> Processing Dependency: pkgconfig(xcb) for package: libX11-devel-1.6.7-3.amzn2.0.2.x86_64
--> Processing Dependency: pkgconfig(kbproto) for package: libX11-devel-1.6.7-3.amzn2.0.2.x86_64
---> Package libXpm.x86_64 0:3.5.12-9.amzn2.0.1 will be installed
---> Package libXpm-devel.x86_64 0:3.5.12-9.amzn2.0.1 will be installed
--> Processing Dependency: libXt.so.6()(64bit) for package: libXpm-devel-3.5.12-9.amzn2.0.1.x86_64
--> Processing Dependency: libXext.so.6()(64bit) for package: libXpm-devel-3.5.12-9.amzn2.0.1.x86_64
```

Step 2: Create Account Information

You need to set up a Nagios user. Run the following commands:

$sudo adduser -m nagios

$sudo passwd nagios

Type the new password twice.

```
Complete!
[ec2-user@ip-172-31-10-222 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-10-222 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 6 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-10-222 ~]$
```

$sudo groupadd nagcmd
$sudo usermod -a -G nagcmd nagios
$sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-10-222 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-10-222 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-10-222 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-10-222 ~]$
```

Step 3: Download Nagios Core and the Plugins
Create a directory for storing the downloads.

$mkdir ~/downloads

$cd ~/downloads

$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

```
[ec2-user@ip-172-31-10-222 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2023-04-05 15:22:34--  http://nagios-plugins.org/download/nagios-plugins-2.0.3.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|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz'

100%[=============================================================================================================>] 2,659,772   1.58MB/s   in 1.6s

2023-04-05 15:22:36 (1.58 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]

[ec2-user@ip-172-31-10-222 downloads]$
```

Step 4: Compile and Install Nagios
Extract the Nagios source code tarball.

$tar zxvf nagios-4.0.8.tar.gz

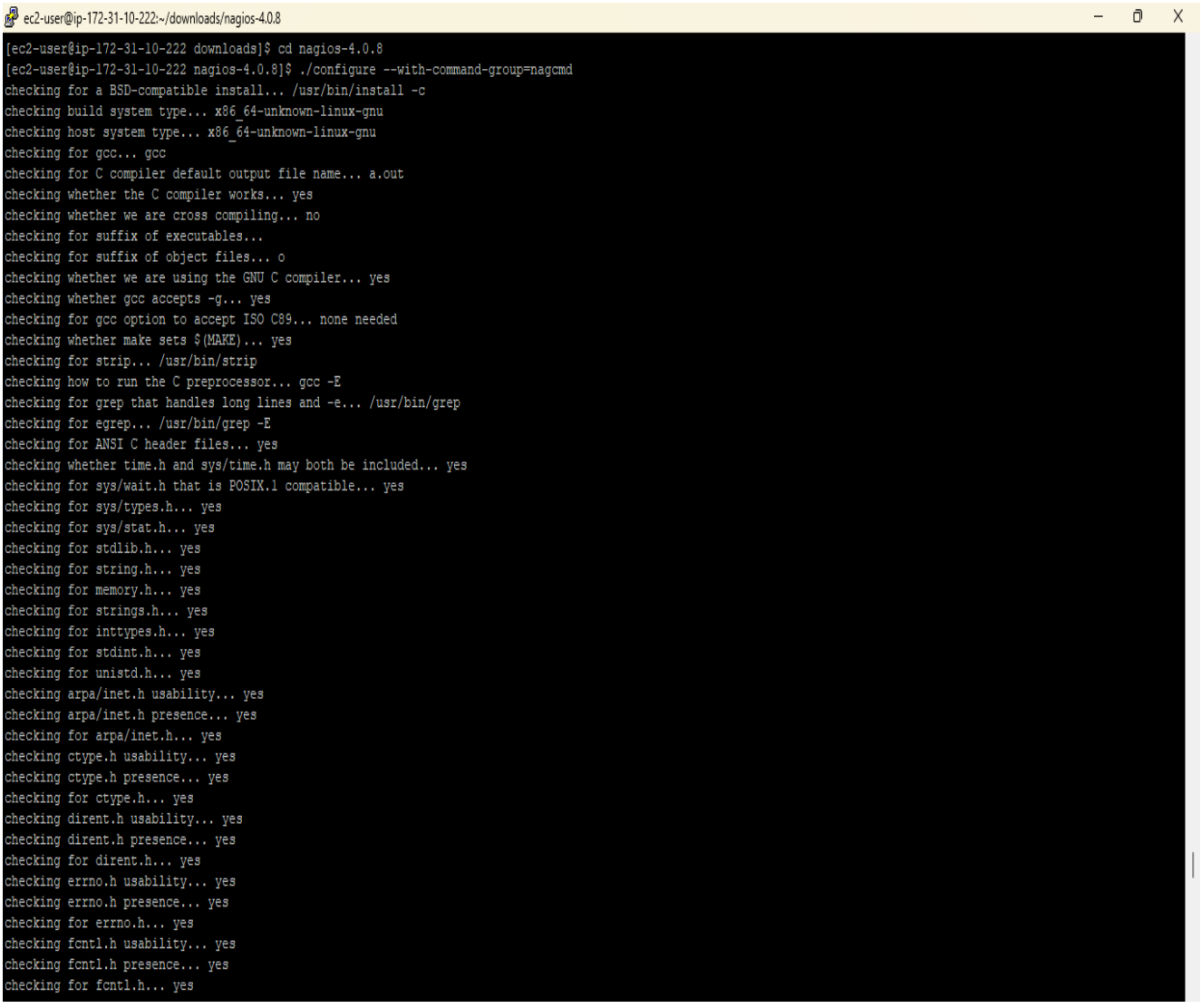
```
ec2-user@ip-172-31-10-222:~/downloads
[ec2-user@ip-172-31-10-222 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
nagios-4.0.8/THANKS
nagios-4.0.8/UPGRADING
nagios-4.0.8/base/
nagios-4.0.8/base/.gitignore
nagios-4.0.8/base/Makefile.in
nagios-4.0.8/base/broker.c
nagios-4.0.8/base/checks.c
nagios-4.0.8/base/commands.c
nagios-4.0.8/base/config.c
nagios-4.0.8/base/events.c
nagios-4.0.8/base/flapping.c
nagios-4.0.8/base/logging.c
nagios-4.0.8/base/nagios.c
nagios-4.0.8/base/nagiostats.c
nagios-4.0.8/base/nebmods.c
nagios-4.0.8/base/nerd.c
nagios-4.0.8/base/netutils.c
nagios-4.0.8/base/notifications.c
nagios-4.0.8/base/perfdata.c
nagios-4.0.8/base/query-handler.c
nagios-4.0.8/base/sehandlers.c
nagios-4.0.8/base/sretention.c
nagios-4.0.8/base/utils.c
nagios-4.0.8/base/workers.c
nagios-4.0.8/base/wp-phash.c
nagios-4.0.8/base/wpres-phash.h
nagios-4.0.8/base/wpres.gperf
nagios-4.0.8/cgi/
nagios-4.0.8/cgi/.gitignore
nagios-4.0.8/cgi/Makefile.in
nagios-4.0.8/cgi/archivejson.c
nagios-4.0.8/cgi/archiveutils.c
nagios-4.0.8/cgi/avail.c
nagios-4.0.8/cgi/cgiauth.c
nagios-4.0.8/cgi/cgiutils.c
nagios-4.0.8/cgi/cmd.c
```

Change the directory to nagios-4.0.8 by using cd command

$cd nagios-4.0.8

Run the configuration script with the name of the group which you have created in the above step.

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



```
config.status: creating contrib/Makefile
config.status: creating cgi/Makefile
config.status: creating html/Makefile
config.status: creating module/Makefile
config.status: creating worker/Makefile
config.status: creating worker/ping/Makefile
config.status: creating xdata/Makefile
config.status: creating daemon-init
config.status: creating t/Makefile
config.status: creating t-tap/Makefile
config.status: creating include/config.h
config.status: creating lib/snprintf.h
config.status: creating lib/iobroker.h

Creating sample config files in sample-config/ ...


*** Configuration summary for nagios 4.0.8 08-12-2014 ***:

 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:  ${prefix}/var/nagios.lock
   Check result directory:  ${prefix}/var/spool/checkresults
          Init directory:  /etc/rc.d/init.d
  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-172-31-10-222 nagios-4.0.8]$
```

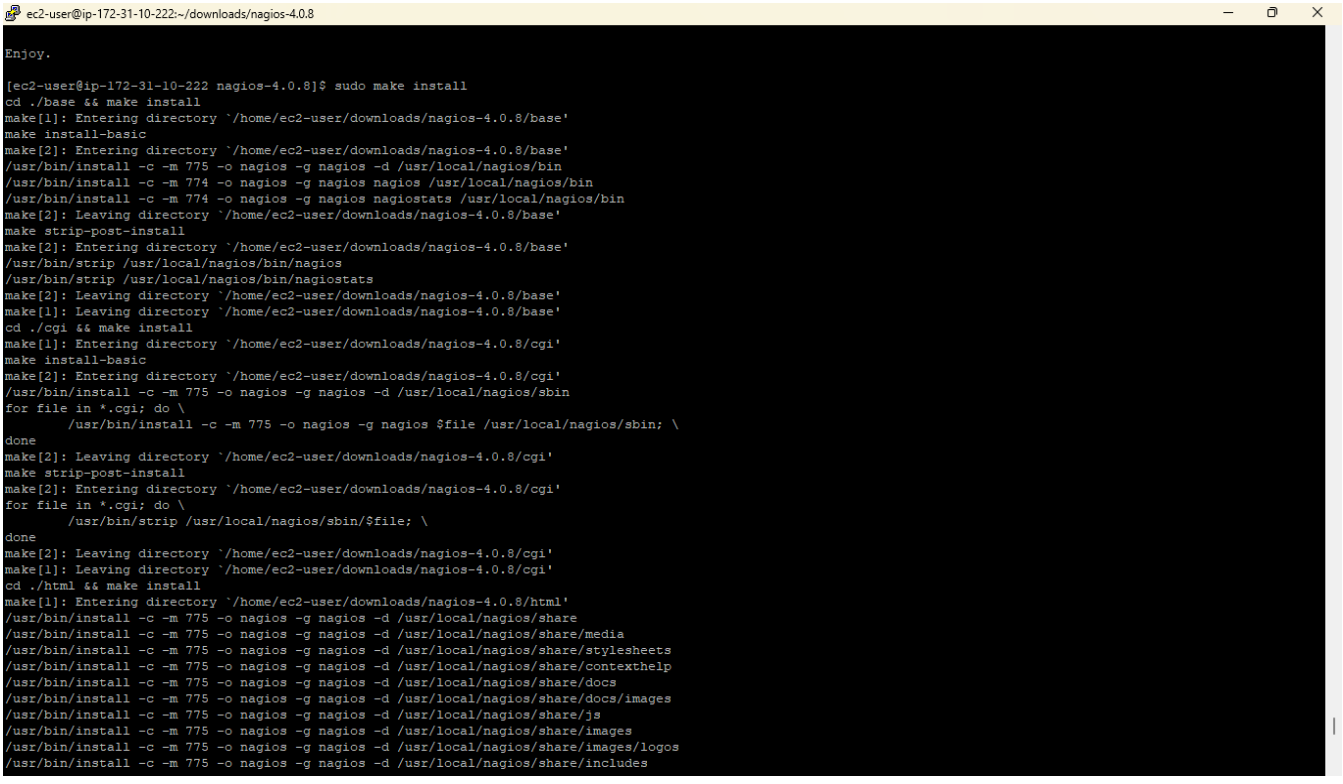
Compile the Nagios source code.

$make all

```
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ make all
cd ./base && make
make[1]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o nerd.o nerd.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o commands.o commands.c
commands.c: In function 'process_passive_service_check':
commands.c:2247:12: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
  cr.source = command_worker.source_name;
            ^
commands.c: In function 'process_passive_host_check':
commands.c:2339:12: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
  cr.source = command_worker.source_name;
            ^
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o flapping.o flapping.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o logging.o logging.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
```

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

$sudo make install



$sudo make install-init

```
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios

*** Init script installed ***
```

$sudo make install-config

```
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios

*** Init script installed ***

[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ 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.

[ec2-user@ip-172-31-10-222 nagios-4.0.8]$
```

$sudo make install-commandmode

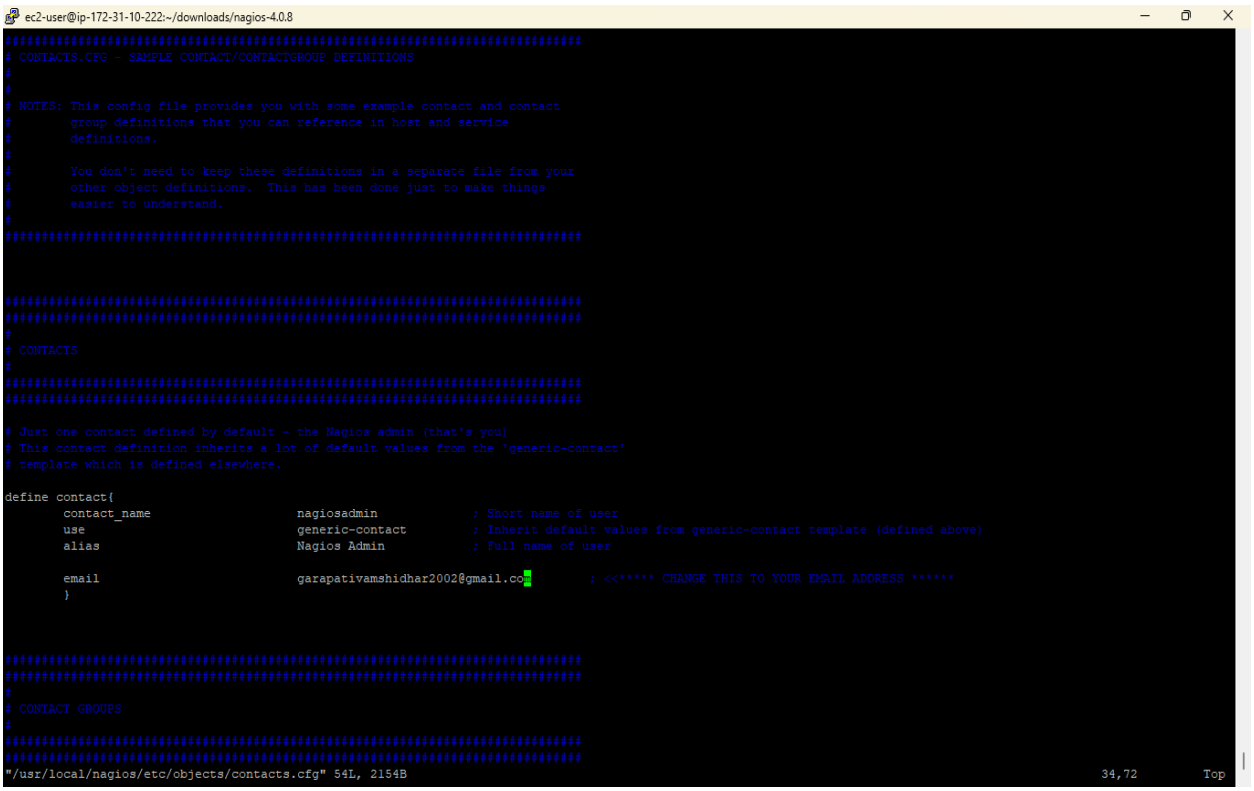
```
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

[ec2-user@ip-172-31-10-222 nagios-4.0.8]$
```

Step 5: Customize Configuration

$sudo vim /usr/local/nagios/etc/objects/contacts.cfg

```
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ sudo vim /usr/local/nagios/etc/objects/contacts.cfg
```

Change E-Mail address with nagiosadmin contact definition you'd like to use for receiving Nagios alerts.



Step 6: Configure the Web Interface

$sudo make install-webconf

```
[ec2-user@ip-172-31-10-222 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-172-31-10-222 nagios-4.0.8]$
```

Create a nagiosadmin account for logging into the Nagios web interface. Note the password you need it while login to Nagios web console.

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

```
[ec2-user@ip-172-31-10-222 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
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$
```

$sudo service httpd restart

Step 7: Compile and Install the Nagios Plugins

Extract the Nagios plugins source code tarball.

$cd ~/downloads

$tar zxvf nagios-plugins-2.0.3.tar.gz

```
[ec2-user@ip-172-31-10-222 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
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ cd ~/downloads
[ec2-user@ip-172-31-10-222 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
nagios-plugins-2.0.3/ABOUT-NLS
nagios-plugins-2.0.3/configure.ac
nagios-plugins-2.0.3/Makefile.in
nagios-plugins-2.0.3/config.h.in
```

$cd nagios-plugins-2.0.3

Compile and install the plugins.

$./configure --with-nagios-user=nagios --with-nagios-group=nagios
$ make

```
[ec2-user@ip-172-31-10-222 downloads]$ cd nagios-plugins-2.0.3
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ ./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 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
```

```
                  --enable-libtap... no
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ make
make  all-recursive
make[1]: Entering directory `/home/ec2-user/downloads/nagios-plugins-2.0.3'
Making all in gl
make[2]: Entering directory `/home/ec2-user/downloads/nagios-plugins-2.0.3/gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
  cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
```

$sudo make install

```
make[2]: Leaving directory `/home/ec2-user/downloads/nagios-plugins-2.0.3'
make[1]: Leaving directory `/home/ec2-user/downloads/nagios-plugins-2.0.3'
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ sudo make install
```

Step 8: Start Nagios

Add Nagios to the list of system services and have it automatically start when the system boots.

$sudo chkconfig --add nagios
$sudo chkconfig nagios on

```
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ sudo chkconfig --add nagios
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ sudo chkconfig nagios on
[ec2-user@ip-172-31-10-222 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.
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-172-31-10-222 nagios-plugins-2.0.3]$
```

If there are no errors, start Nagios.

$sudo service nagios start

```
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ sudo service nagios start
Starting nagios (via systemctl):                           [  OK  ]
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$
```

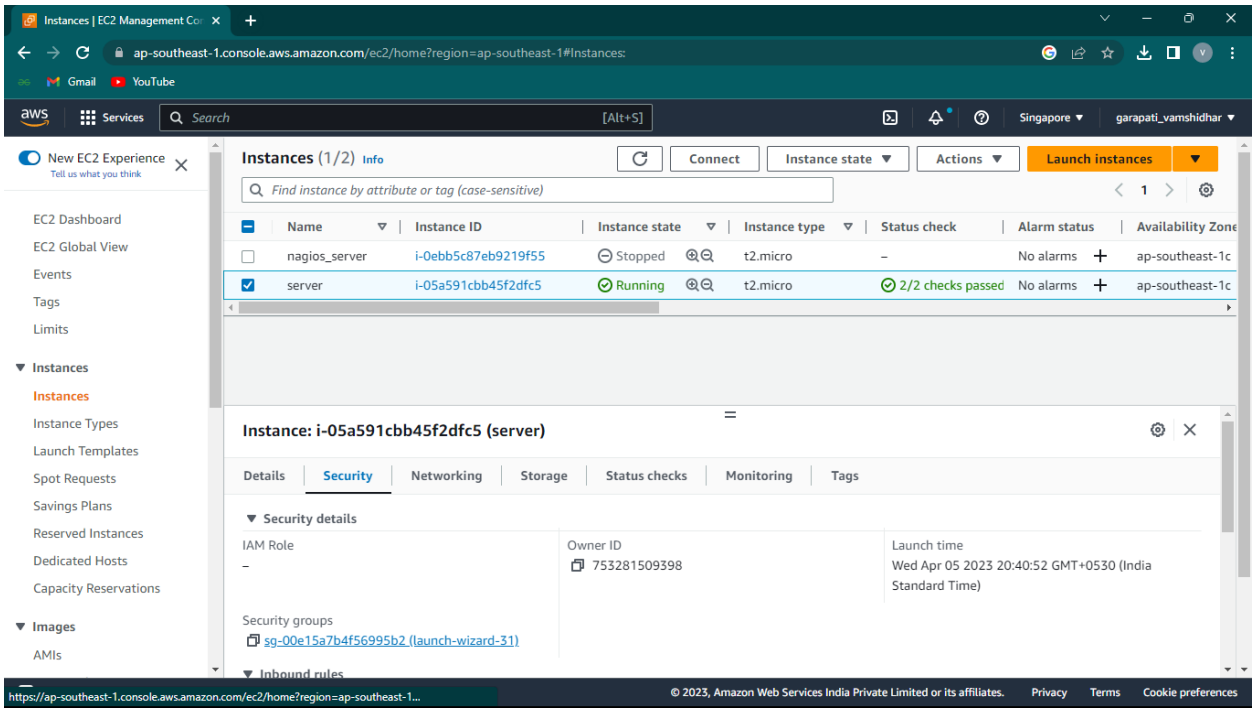
Step 9: Update AWS Security Group

you need to open port 80 on the new AWS EC2 server to incoming traffic so you can connect to the new Nagios webpage.

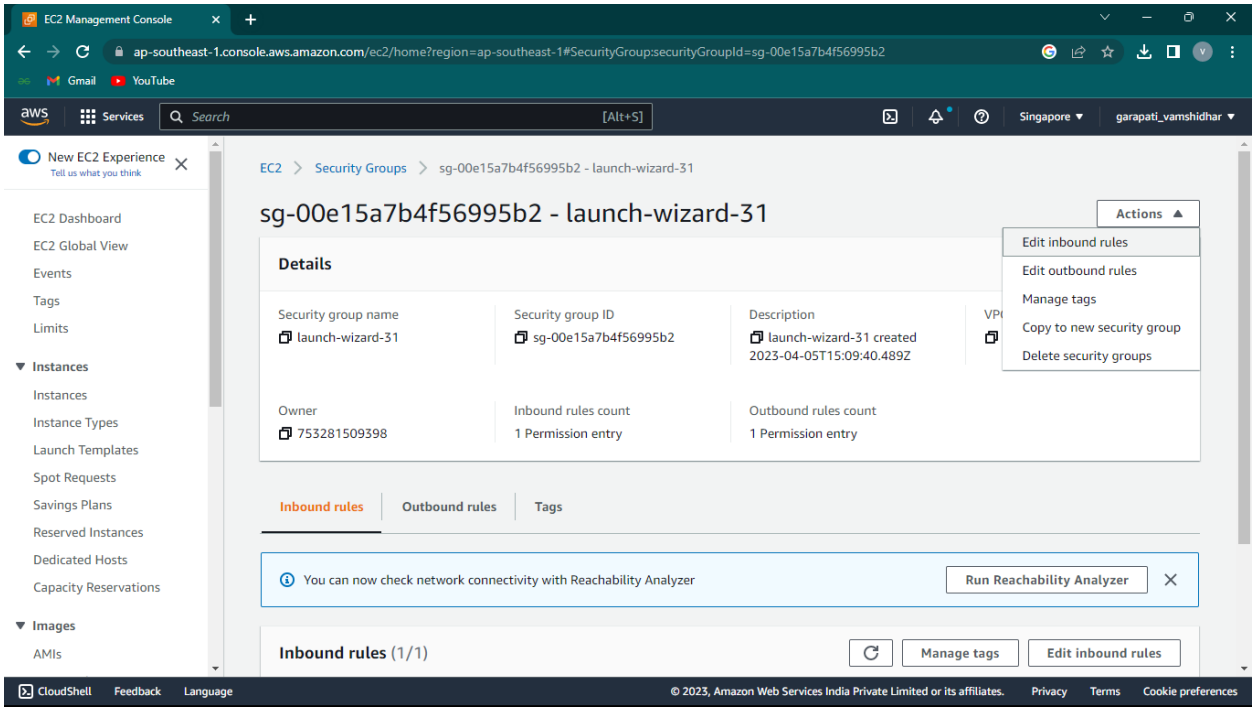
* From the EC2 console select Security Groups from the left navigation pane.
* Select the Security Group applicable for the instance that Nagios was installed on and open the Inbound tab
* If there is no rule to allow HTTP traffic on port 80 then click edit in the Inbound tab to add a new rule
* Click on New Rule button
* Scroll down to select HTTP from the list of Type

* If you want to be able to access Nagios from anywhere then select Save, otherwise enter the IP address or range of IP address you want to be able to access it from then select Save.

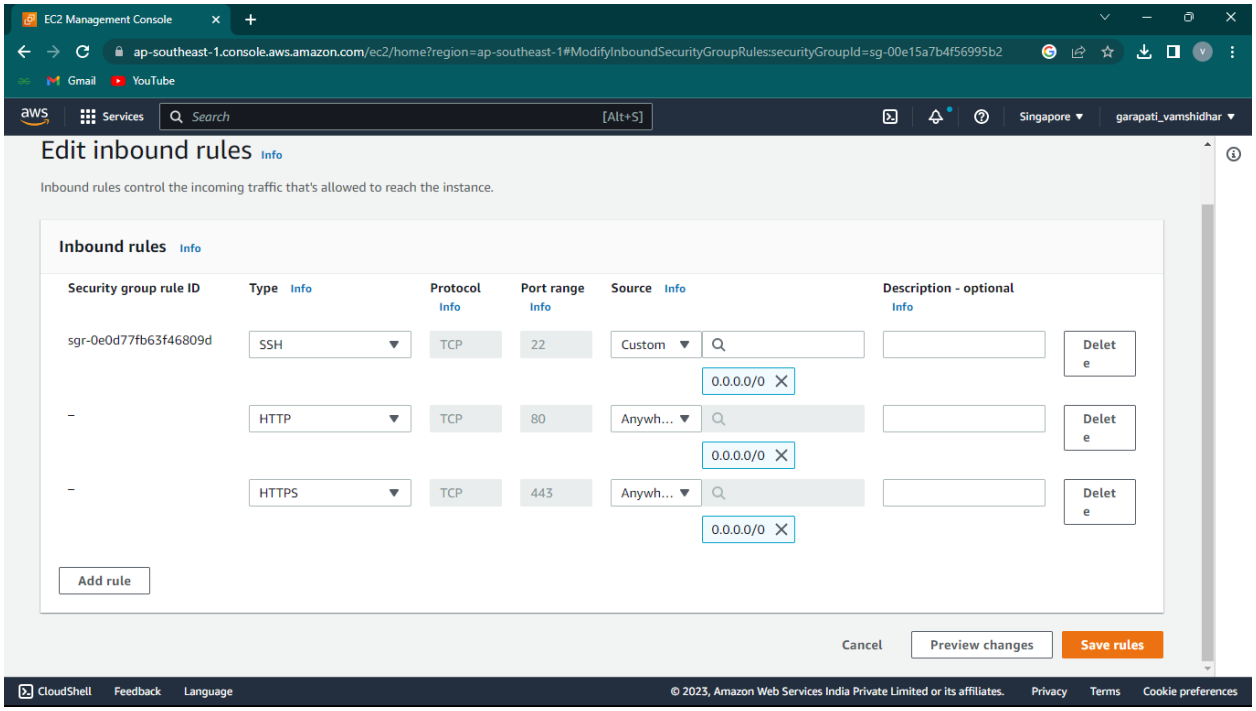
- Now go to EC2 instance



- Go to security



- Now edit the inbound rules

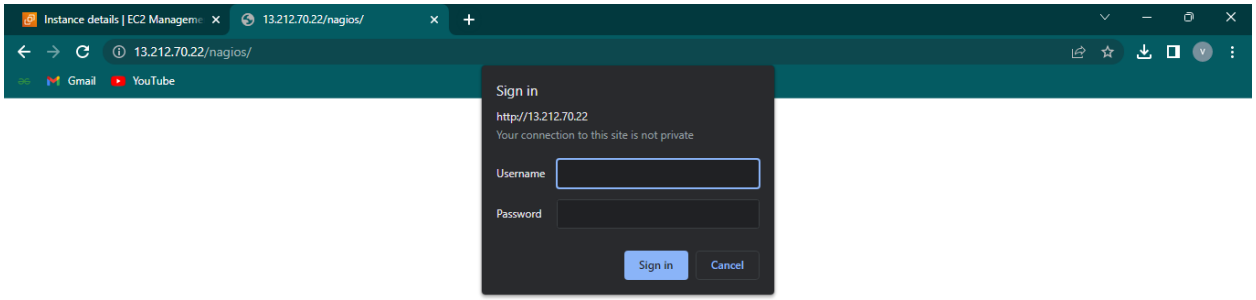


Step 10: Log in to the Web Interface

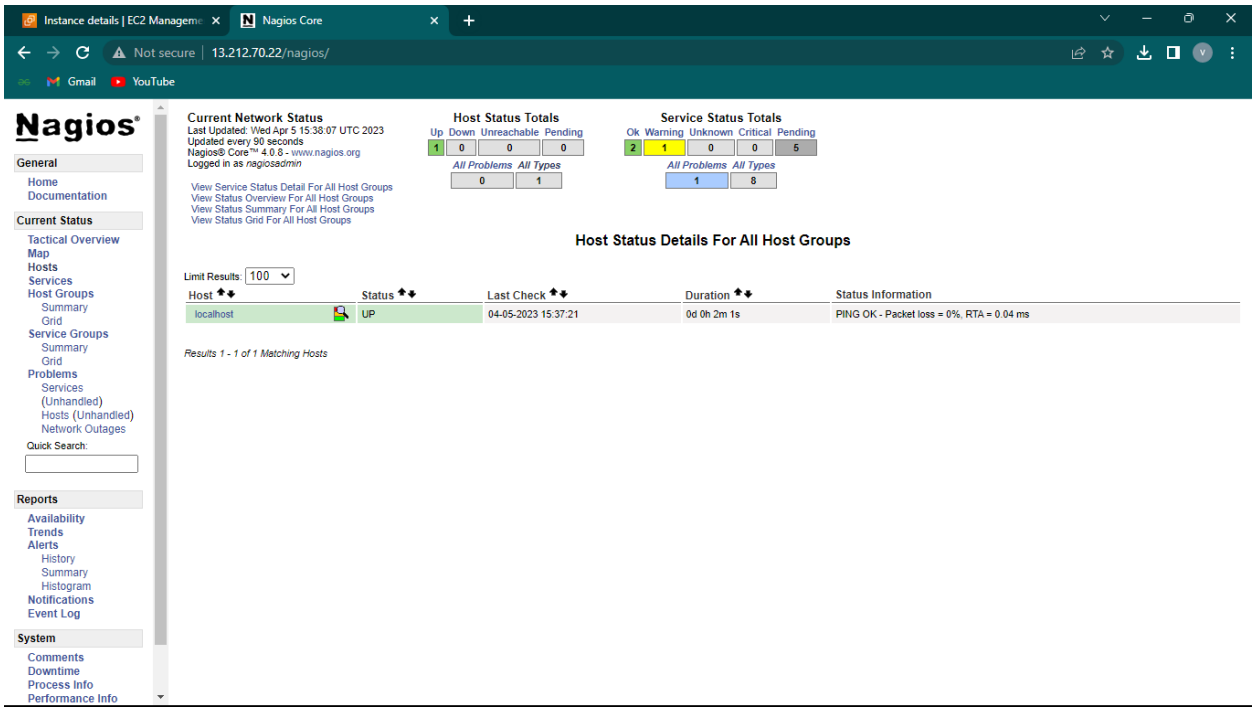
access the Nagios web interface to do this you will need to know the Public DNS or IP for your instance, you can get this from the Instance section of the EC2 Console if you do not already know it. You'll be prompted for the username (nagiosadmin) and password you specified earlier.

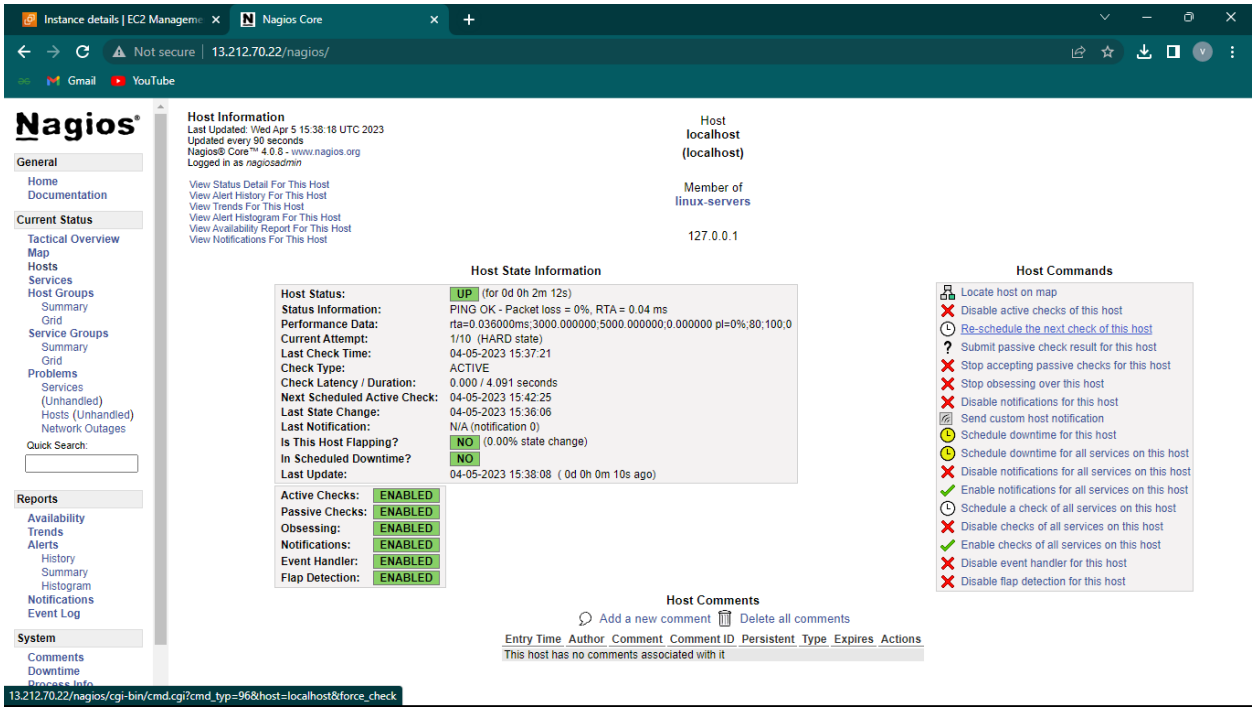
Eg:Ipaddress/nagios

Sign in to the Nagios by using username and password.



- This is the Interface of the Nagios Core





Monitoring with different ip's

1.In the first step connect to the root user and switch to the /etc directory and also check the list of files available in that directory.

$sudo su

$cd /etc

```
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ sudo su
[root@ip-172-31-10-222 nagios-plugins-2.0.3]# cs /etc
bash: cs: command not found
[root@ip-172-31-10-222 nagios-plugins-2.0.3]# cd /etc
[root@ip-172-31-10-222 etc]# ls
acpi                cron.weekly           groff                issue.net       mtab                ppp              rwtab.d          sysctl.conf
adjtime             csh.cshrc             group                krb5.conf       my.cnf              prelink.conf.d   sasl2            sysctl.d
aliases             csh.login             group-               krb5.conf.d     my.cnf.d            printcap         scl              systemd
aliases.db          dbus-1                grub2.cfg            ld.so.cache     nanorc              profile          screenrc         system-release
alternatives        default               grub2-efi.cfg        ld.so.conf      netconfig           profile.d        securetty        system-release-cpe
amazon              depmod.d              grub.d               ld.so.conf.d    NetworkManager      protocols        security         terminfo
anacrontab          dhcp                  gshadow              libaudit.conf   networks            python           selinux          tmpfiles.d
at.deny             DIR_COLORS            gshadow-             libnl           nfs.conf            rc0.d            services         trusted-key.key
audisp              DIR_COLORS.256color   gss                  libuser.conf    nfsmount.conf       rc1.d            sestatus.conf    udev
audit               DIR_COLORS.lightbgcolor gssproxy           locale.conf     nsswitch.conf       rc2.d            setuptool.d      updatedb.conf
bash_completion.d   dracut.conf           hibagent-config.cfg  localtime       nsswitch.conf.bak   rc3.d            shadow           update-motd.d
bashrc              dracut.conf.d         hibinit-config.cfg   login.defs      openldap            rc4.d            shadow-          vconsole.conf
binfmt.d            e2fsck.conf           host.conf            logrotate.conf  opt                 rc5.d            shells           vimrc
chkconfig.d         environment           hostname             logrotate.d     os-release          rc6.d            skel             virc
chrony.conf         ethertypes            hosts                lsm             pam.d               rc.d             ssh              wgetrc
chrony.d            exports               hosts.allow          lvm             passwd              rc.local         ssl              X11
chrony.keys         exports.d             hosts.deny           machine-id      passwd-             request-key.conf statetab        xdg
cifs-utils          filesystems           httpd                magic           php.d               request-key.d    statetab.d       xinetd.d
cloud               fonts                 idmapd.conf          mailcap         php.ini             resolv.conf      subgid           yum
cron.d              fstab                 image-id             man_db.conf     pkcs11              rpc              subuid           yum.conf
cron.daily          gcrypt                init.d               mime.types      pki                 rpm              sudo.conf        yum.repos.d
cron.deny           GeoIP.conf            inittab              mke2fs.conf     plymouth            rsyncd.conf      sudoers
cron.hourly         GeoIP.conf.default    inputrc              modprobe.d      pm                  rsyslog.conf     sudoers.d
cron.monthly        gnupg                 iproute2             modules-load.d  popt.d              rsyslog.d        sudo-ldap.conf
crontab             GREP_COLORS           issue                motd            postfix             rwtab            sysconfig
[root@ip-172-31-10-222 etc]#
```

2. Now switch to ssh directory and change the authentication and password rules for the access of root user.

$cd ssh

$ls

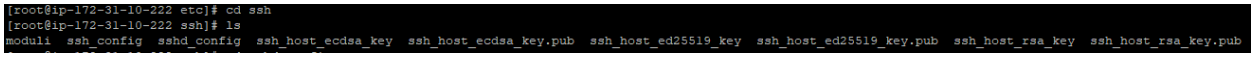
$nano sshd_config

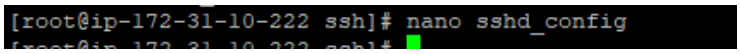
Change the rules:

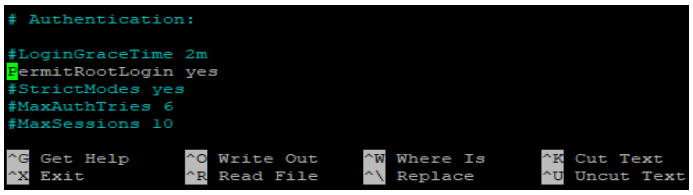
#PermitRootLogin Yes ---> PermitRootLogin Yes

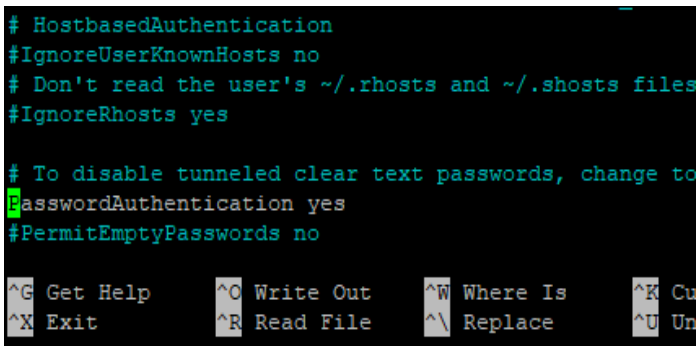
#PasswordAuthentication Yes---> PasswordAuthentication Yes

Now save the changes.





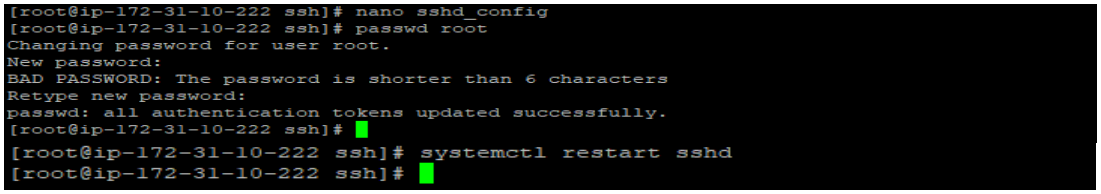




3.Generate the password for the login and restart sshd.

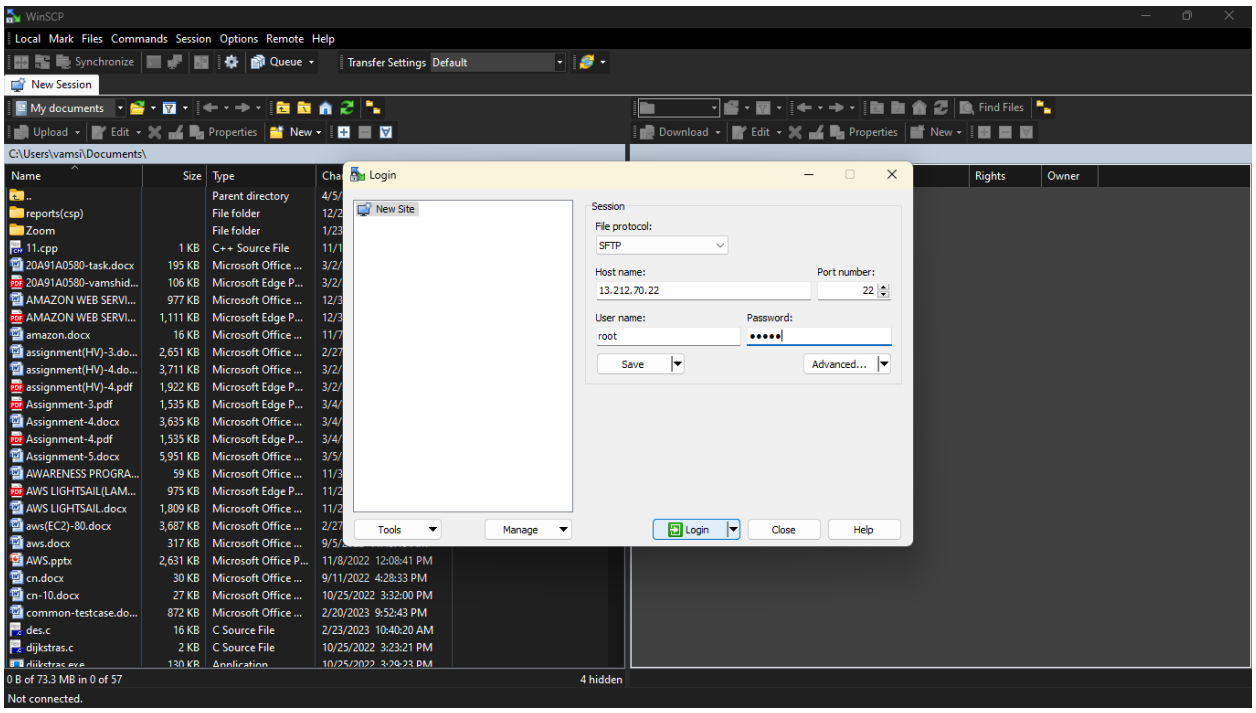
$passwd root

$systemctl restart sshd



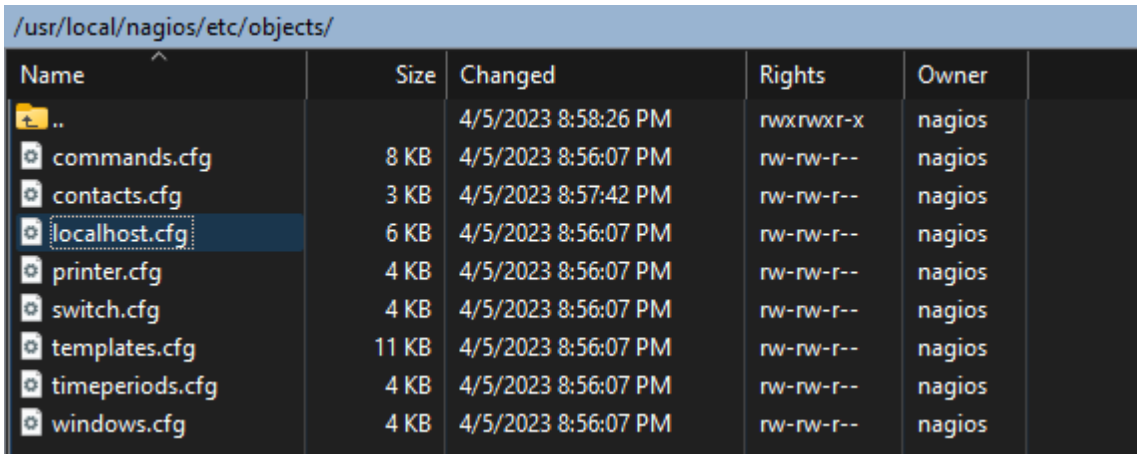
4.Install WinSCP tool for generating and modifying the configuration files instead of using command line interface (CLI).

5.After the installation give the host name (public Ip), username(root) and password.



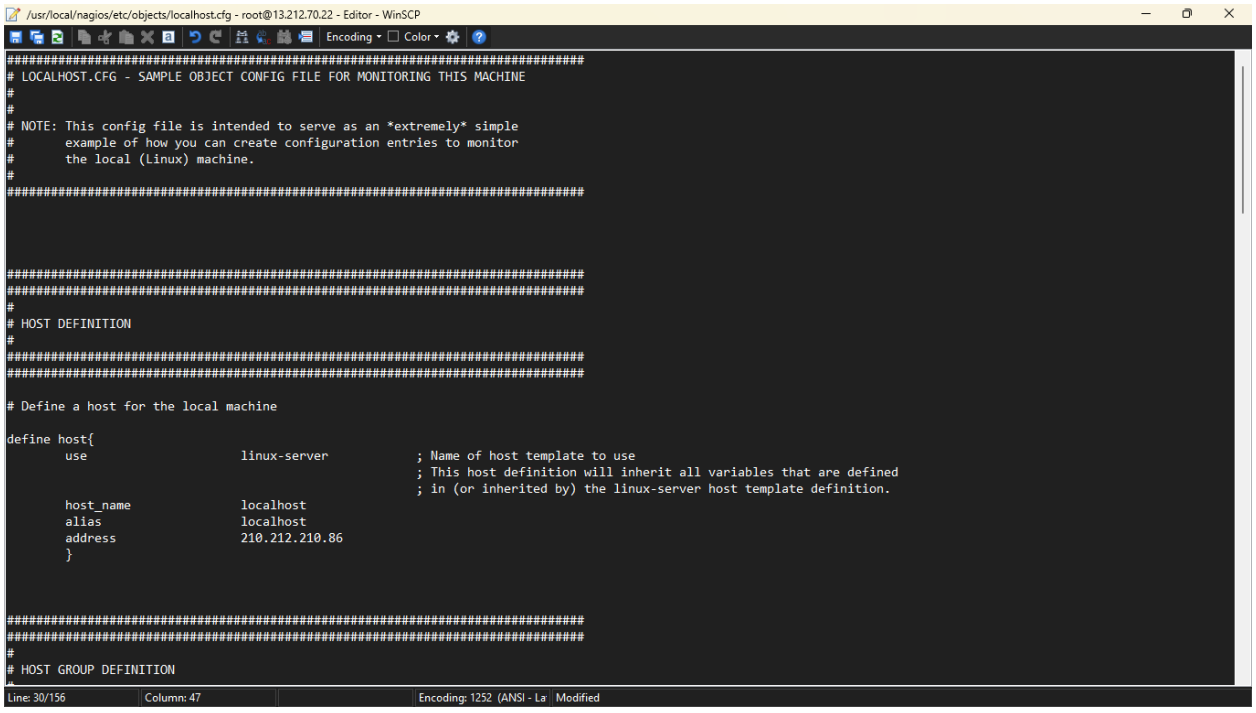
6.Now switch to the localhost.cfg using the below path

Path: **/usr/local/nagios/etc/objects/**



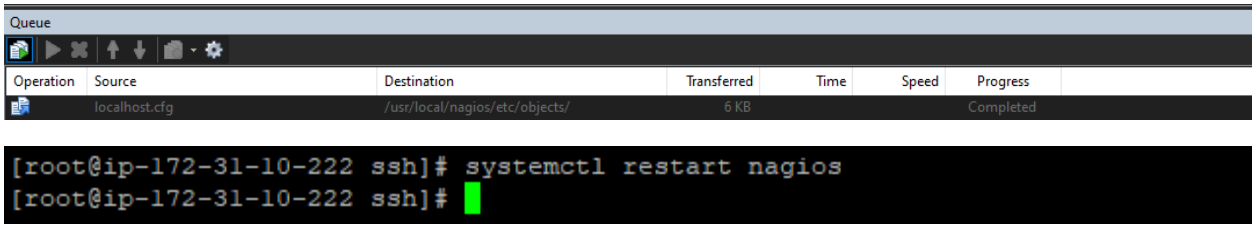
7.In the host definition modify the ip address which is to be monitored. Here we are using codemind

Monitoring code mind IP (**210.212.210.86**)

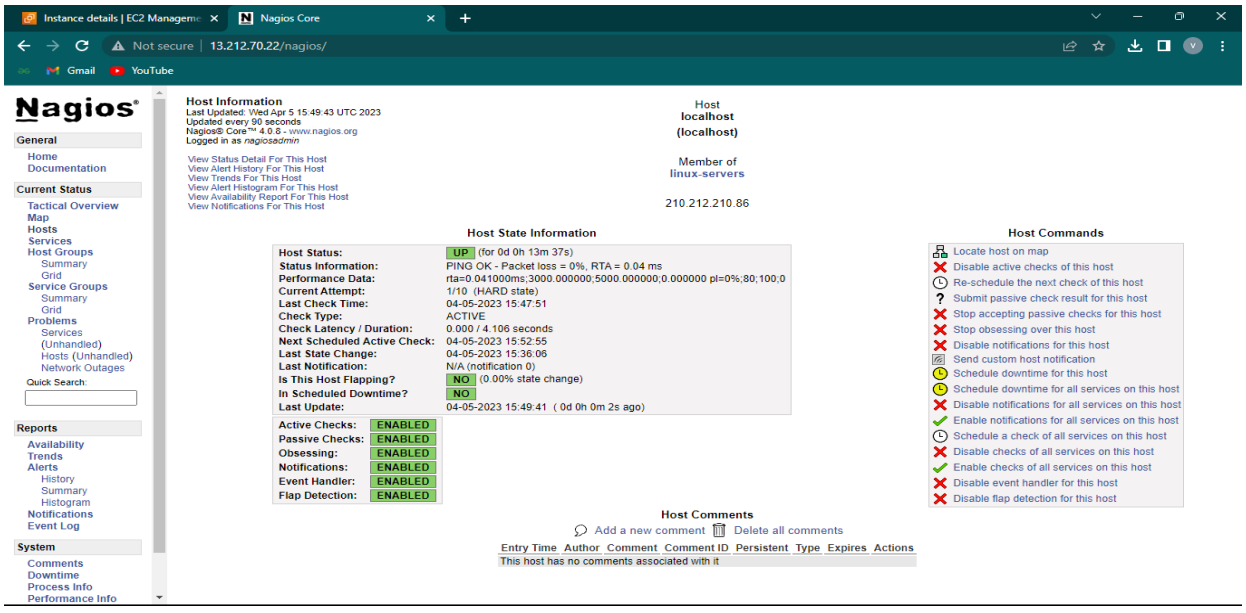


8.After modifying the localhost.cfg file save the file and restart the nagios.

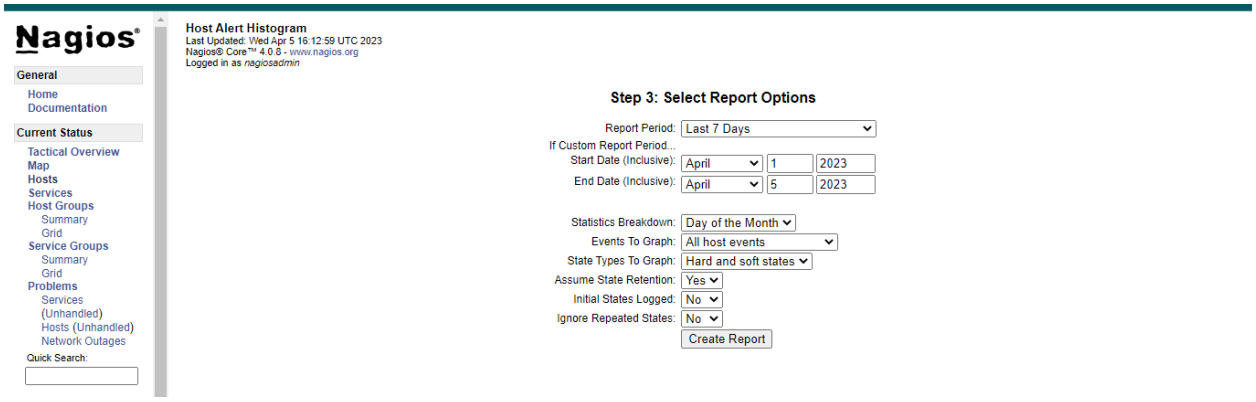
$systemctl restart nagios

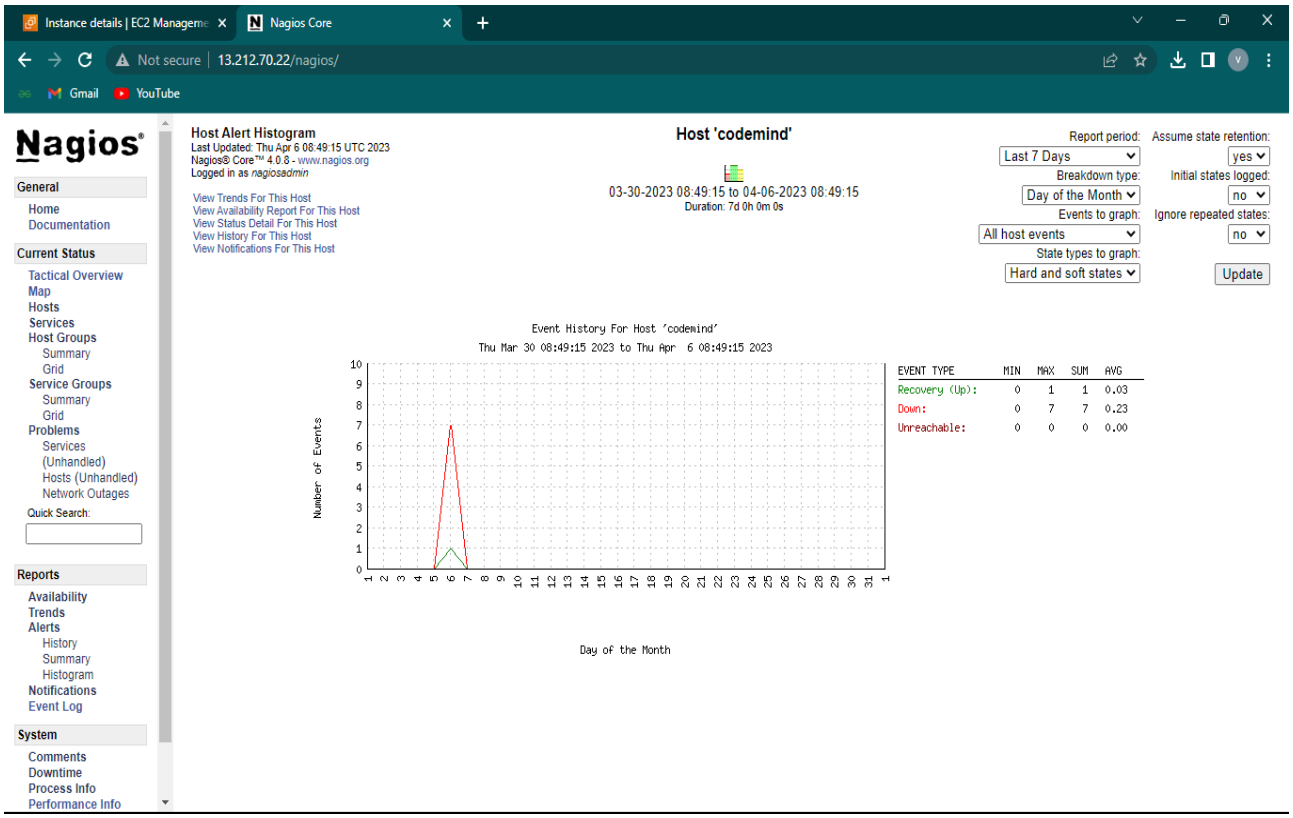



```
[root@ip-172-31-10-222 ssh]# systemctl restart nagios
[root@ip-172-31-10-222 ssh]#
```



9.In the histogram we can find the graph which is easy to understand.





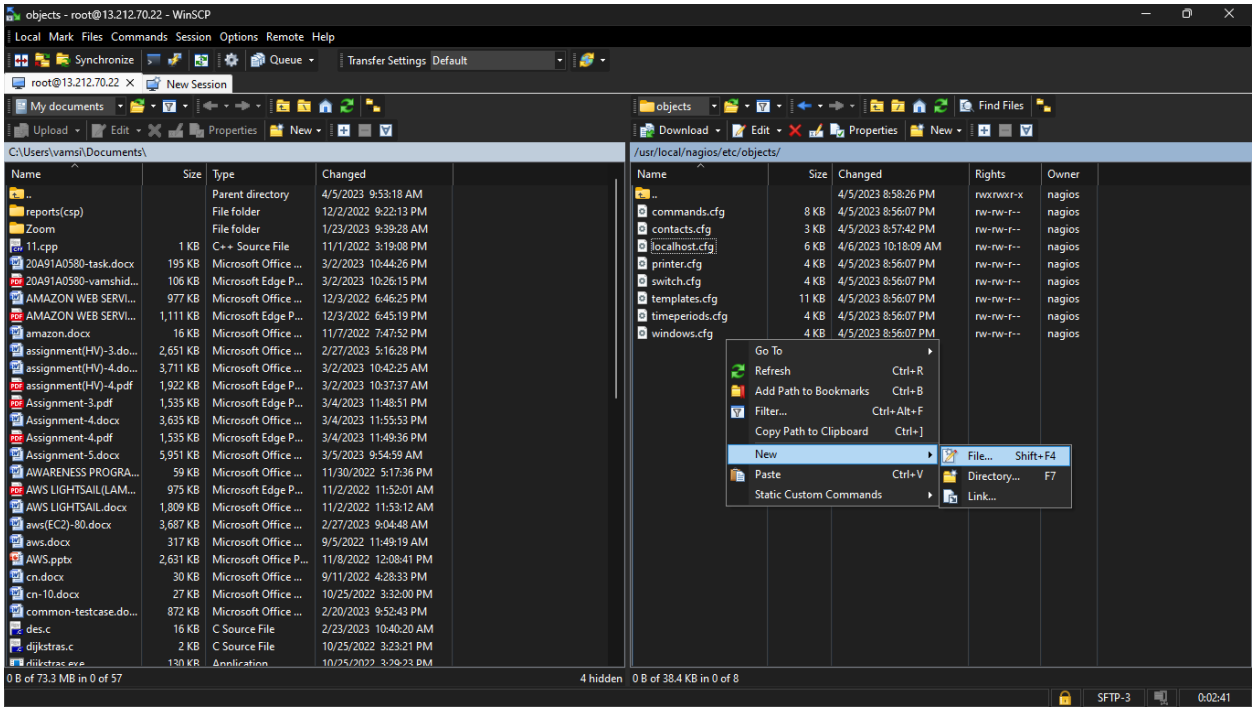
Adding hosts and monitoring:

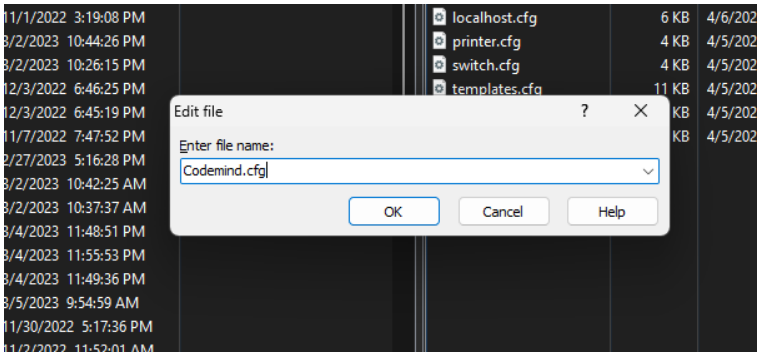
1.Switch to the path below:

Path: **/usr/local/nagios/etc/objects/**

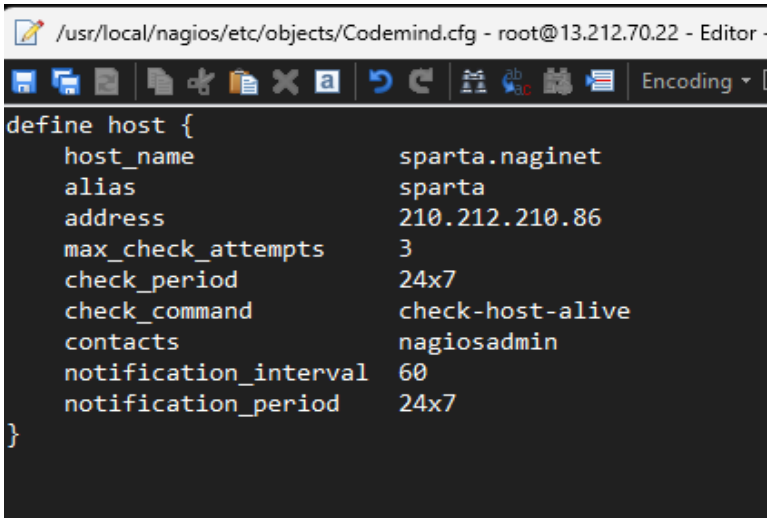
2.create a new file with .cgf extention.

Eg:Codemind.cfg





3.Enter the below code with the required ip and save it.



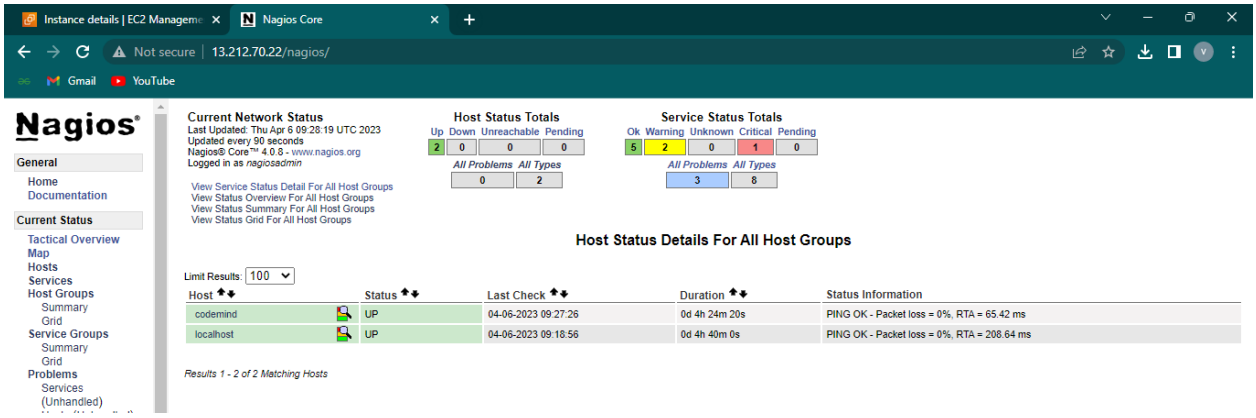
4.Switch to the path below and add the mentioned line in nagios.cfg file. Restart the nagios.

Path: **/usr/local/nagios/etc/objects/codemind.cfg**

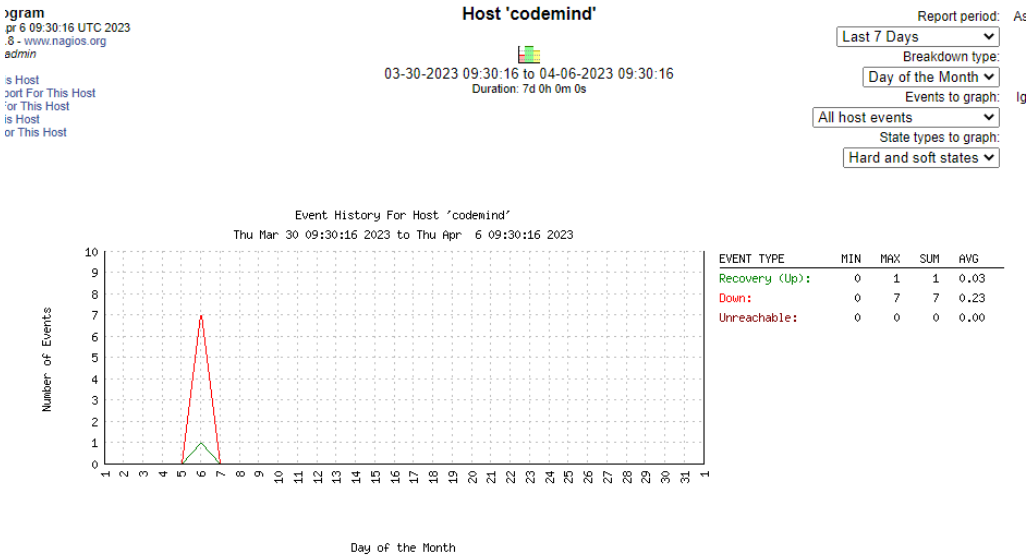
```
#   rampup_change - # of jobs to add to jobs_limit when ramping up
# NOTE: The backoff_limit and rampup_limit are NOT used by anything currently,
#       so if your system is under load nothing will actively modify the jobs
#       even if you have these options enabled, they are for external
#       connector information only.  However, if you change the jobs_max or
#       jobs_min manually here or through the query handler interface that
#       WILL affect your system
#loadctl_options=jobs_max=100;backoff_limit=10;rampup_change=5
cfg_file=/usr/local/nagios/etc/objects/codemind.cfg
```

```
[root@ip-172-31-10-222 ssh]# systemctl restart nagios
[root@ip-172-31-10-222 ssh]#
```

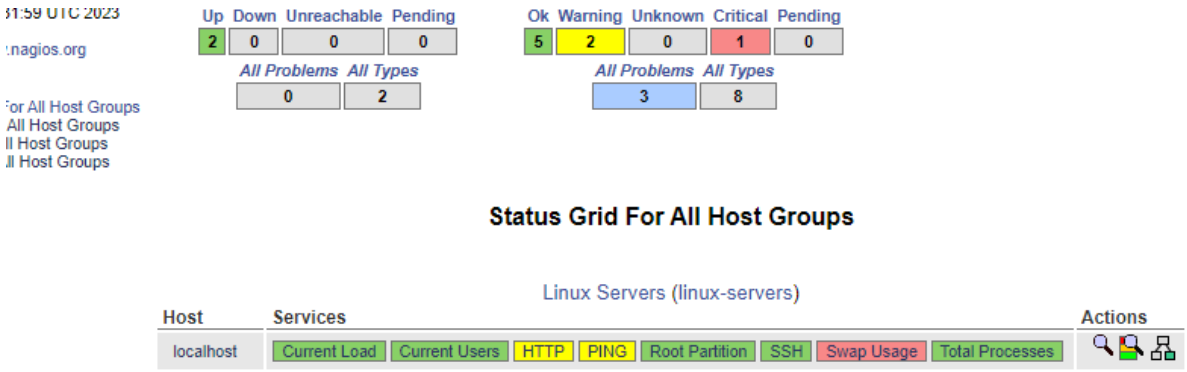
5.In the url ip/nagios we can file the host that was added.



6.In the histograms we can observe the graph.

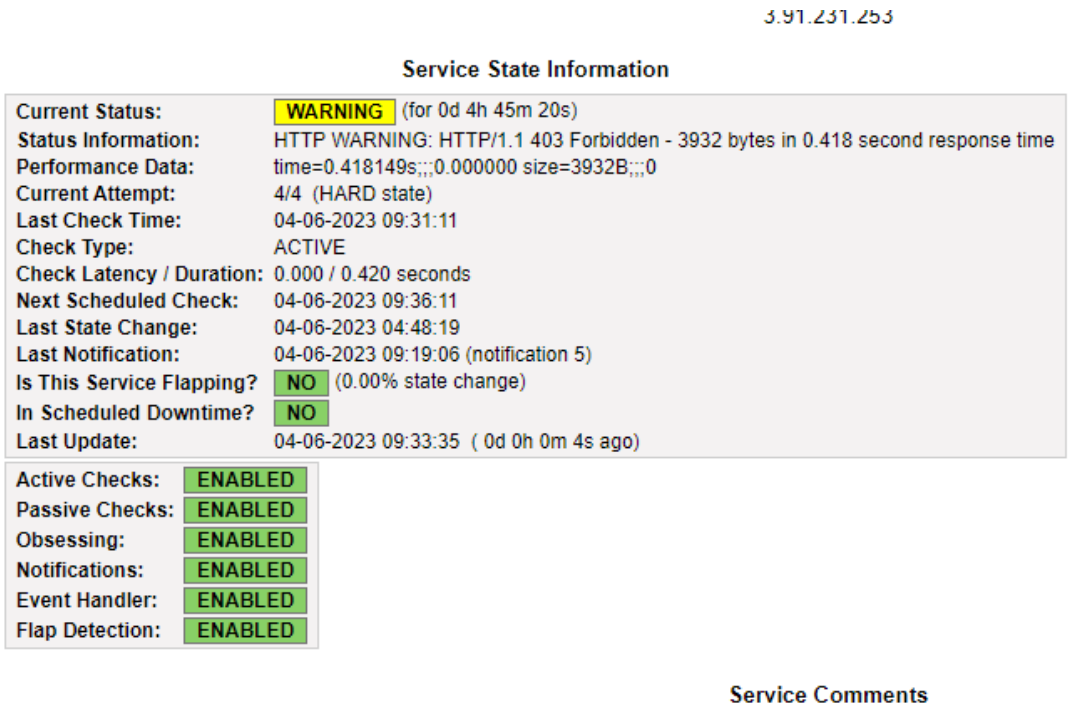


7.In the grid we can find many options like SSH,HTTP,PING etc.



8.Selecting the options we can observe HTTP, SSH, storage used, ping status,etc.

HTTP status:



DISK status:

3.91.231.253

Service State Information

Current Status:	OK (for 0d 17h 55m 24s)
Status Information:	DISK OK - free space: / 6276 MB (76% inode=98%):
Performance Data:	/=1903MB;6543;7361;0;8179
Current Attempt:	1/4 (HARD state)
Last Check Time:	04-06-2023 09:29:18
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 0.001 seconds
Next Scheduled Check:	04-06-2023 09:34:18
Last State Change:	04-05-2023 15:38:36
Last Notification:	N/A (notification 0)
Is This Service Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	04-06-2023 09:33:55 (0d 0h 0m 5s ago)

Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

USERS status:

Service State Information

Current Status:	OK (for 0d 17h 57m 43s)
Status Information:	USERS OK - 1 users currently logged in
Performance Data:	users=1;20;50;0
Current Attempt:	1/4 (HARD state)
Last Check Time:	04-06-2023 09:29:56
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 0.002 seconds
Next Scheduled Check:	04-06-2023 09:34:56
Last State Change:	04-05-2023 15:36:44
Last Notification:	N/A (notification 0)
Is This Service Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	04-06-2023 09:34:25 (0d 0h 0m 2s ago)

Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED