

MONITORING WITH NAGIOS

P.Sai Pradeep	20A91A05H6
G.Vamshidhar	20A91A0580
K.D.P.N.Jyothi	20A91A05F5
Ch.Sai Poojitha	20A91A05E2

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, Postgresql
- It gives application level information (Apache, Postfix, etc.).
- Provides active development.
- Has excellent support from huge active community.
- Nagios runs on any operating system.

Benefits of Nagios

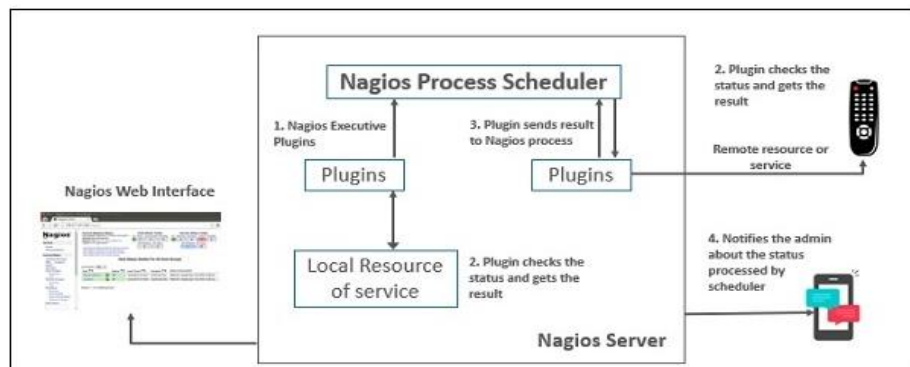
Nagios offers the following benefits for the users –

- 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 –

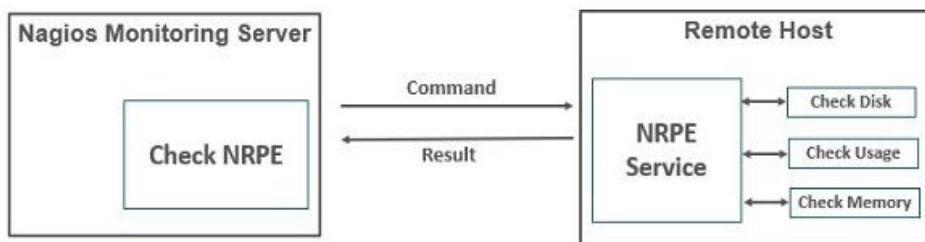
Official Nagios Plugins – There are 50 official Nagios Plugins. Official Nagios plugins are developed and maintained by the official Nagios Plugins Team.

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

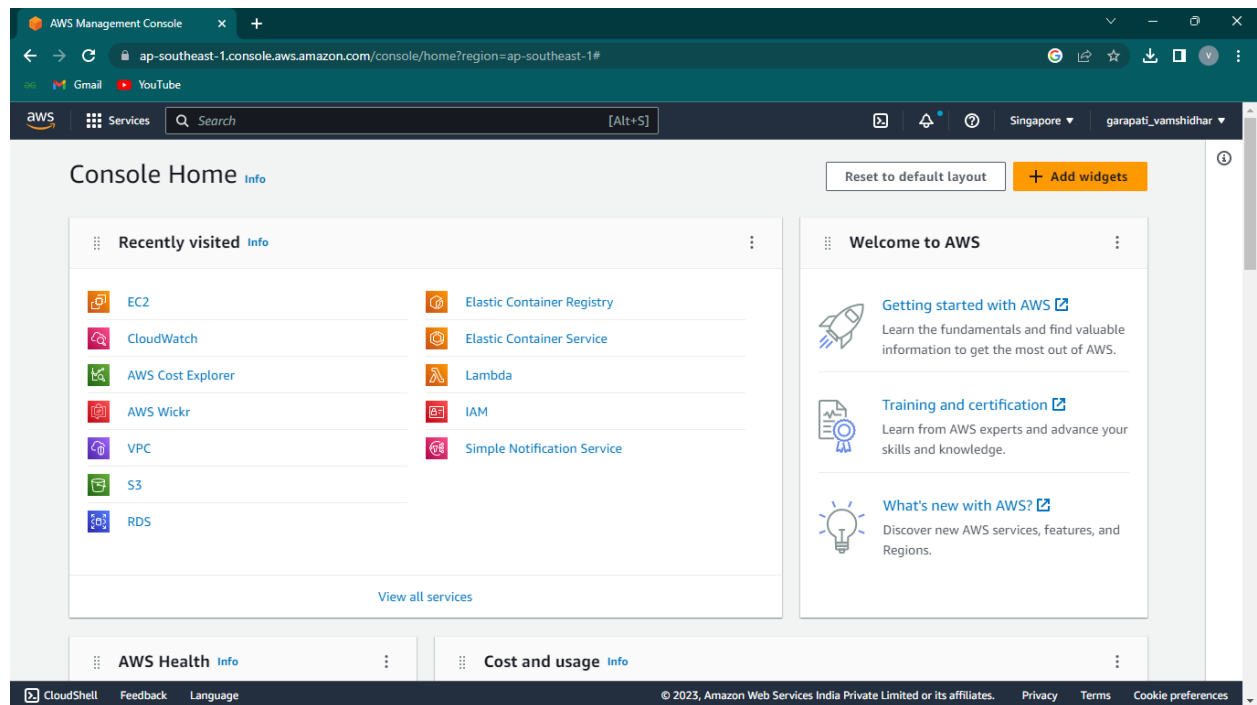
Step 8: Start Nagios

Step 9: Update AWS Security Group

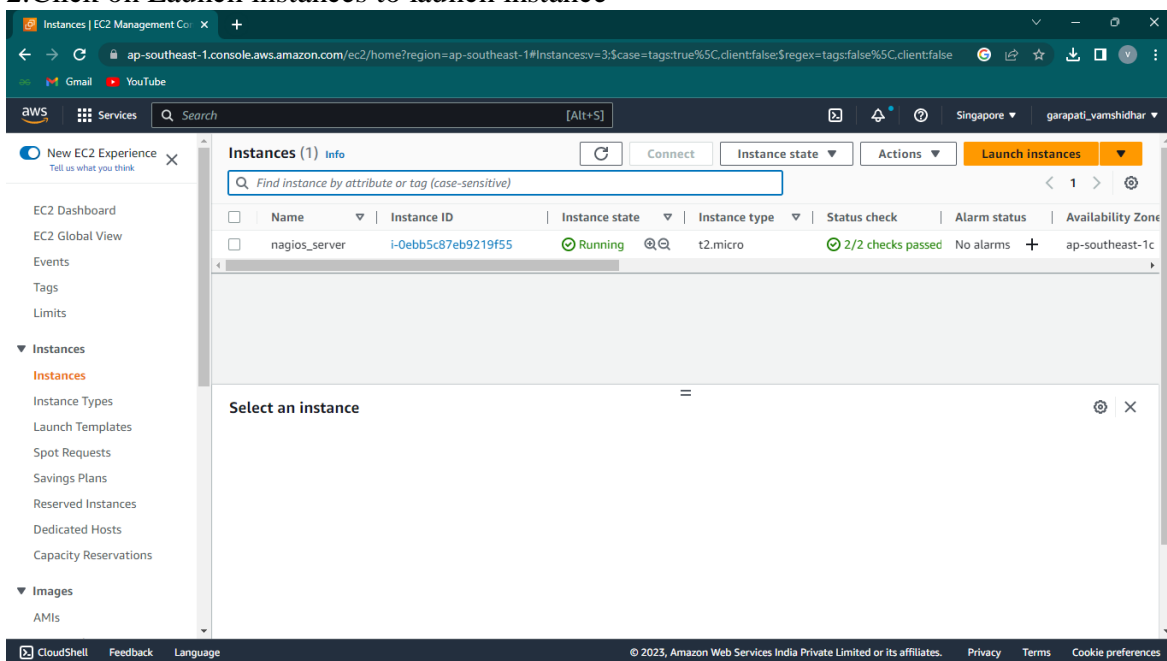
Step 10: Log in to the Web Interfac

*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.

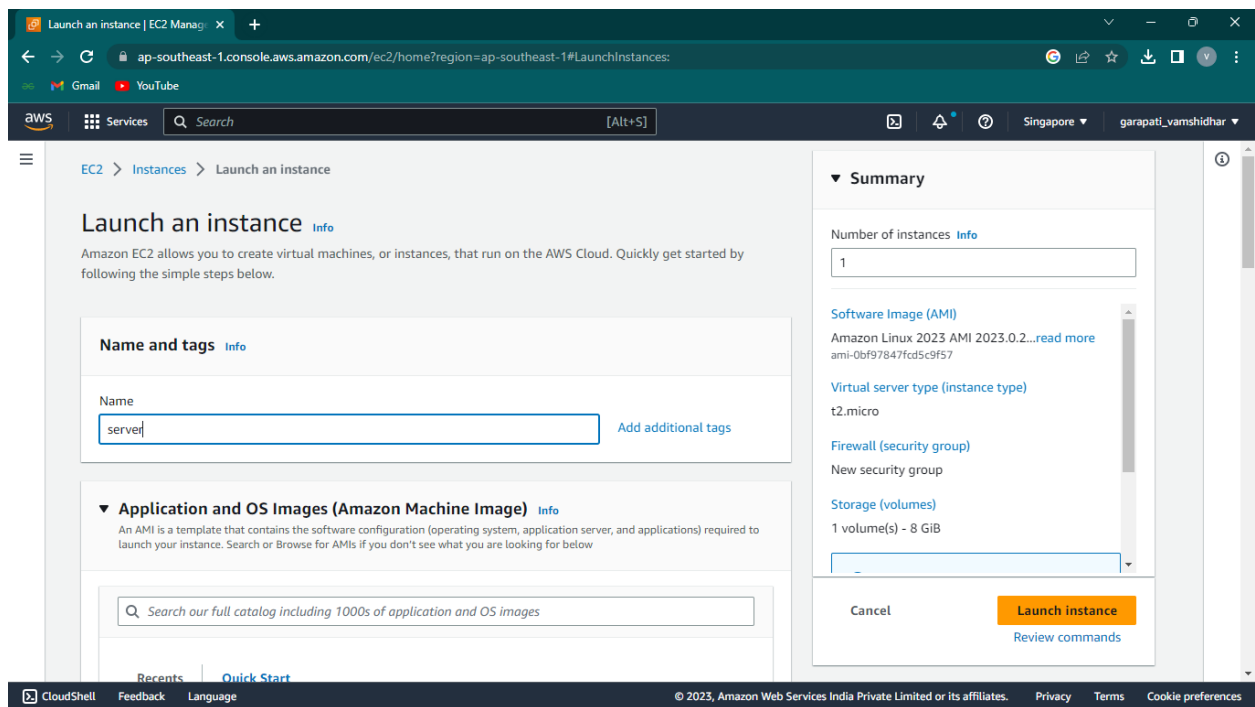
1.First Login to AWS console page



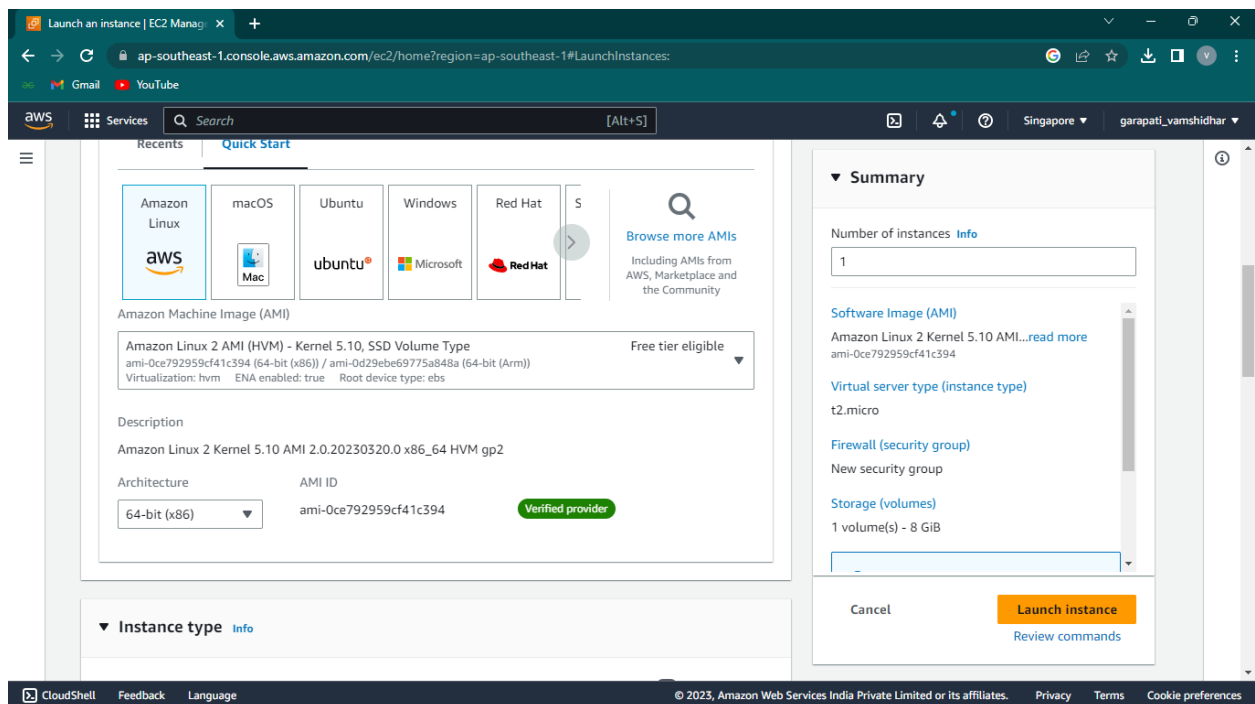
2.Click on Launch instances to launch instance



3. Give name to your instance



4. Select Amazon linux as Amazon Machine Image



5. Create new key pair in ppk format

Launch an instance | EC2 Manag... x +

ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#LaunchInstances:

Services Search [Alt+S]

t2.micro Free tier eligible Compare instance types

Family: t2 1 vCPU 1 GiB Memory
On-Demand Windows pricing: 0.0192 USD per Hour
On-Demand RHEL pricing: 0.0746 USD per Hour
On-Demand Linux pricing: 0.0146 USD per Hour
On-Demand SUSE pricing: 0.0146 USD per Hour

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

nagios_v Create new key pair

▼ Network settings Info Edit

Network Info
vpc-091c9bc478b733809

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0ce792959cf41c394

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Review commands

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

6. In network settings use default VPC and default subnet

Launch an instance | EC2 Manag... x +

ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#LaunchInstances:

Services Search [Alt+S]

▼ Network settings Info

VPC - required Info
vpc-091c9bc478b733809 (default)
172.31.0.0/16

Subnet Info
subnet-0d8b6ee3256797421
VPC: vpc-091c9bc478b733809 Owner: 753281509398
Availability Zone: ap-southeast-1c IP addresses available: 4090 CIDR: 172.31.0.0/20

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required
launch-wizard-31
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./!@#%^&*~

Description - required Info
launch-wizard-31 created 2023-04-05T15:09:40.489Z

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0ce792959cf41c394

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Review commands

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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

Launch an instance | EC2 Manag... x +

ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#LaunchInstances:

Services Search [Alt+S]

launched-wizard-31

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()@[]+=&:;!\$*

Description - required [Info](#)

launched-wizard-31 created 2023-04-05T15:09:40.489Z

Inbound security groups rules

Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

Type [Info](#) ssh Protocol [Info](#) TCP Port range [Info](#) 22

Source type [Info](#) Anywhere Source [Info](#) 0.0.0.0/0 Description - optional [Info](#) e.g. SSH for admin desktop

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

Summary

Number of instances [Info](#) 1

Software Image (AMI) Amazon Linux 2 Kernel 5.10 AMI...[read more](#) ami-0ce792959cf41c394

Virtual server type (instance type) t2.micro

Firewall (security group) New security group

Storage (volumes) 1 volume(s) - 8 GiB

Cancel Launch instance [Review commands](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

8. Click on launch instance to launch the instance

Launch an instance | EC2 Manag... x +

ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#LaunchInstances:

Services Search [Alt+S]

security group rules to allow access from known IP addresses only.

Add security group rule

Advanced network configuration

Configure storage [Info](#) [Advanced](#)

1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

0 x File systems [Edit](#)

Advanced details [Info](#)

Summary

Number of instances [Info](#) 1

Software Image (AMI) Amazon Linux 2 Kernel 5.10 AMI...[read more](#) ami-0ce792959cf41c394

Virtual server type (instance type) t2.micro

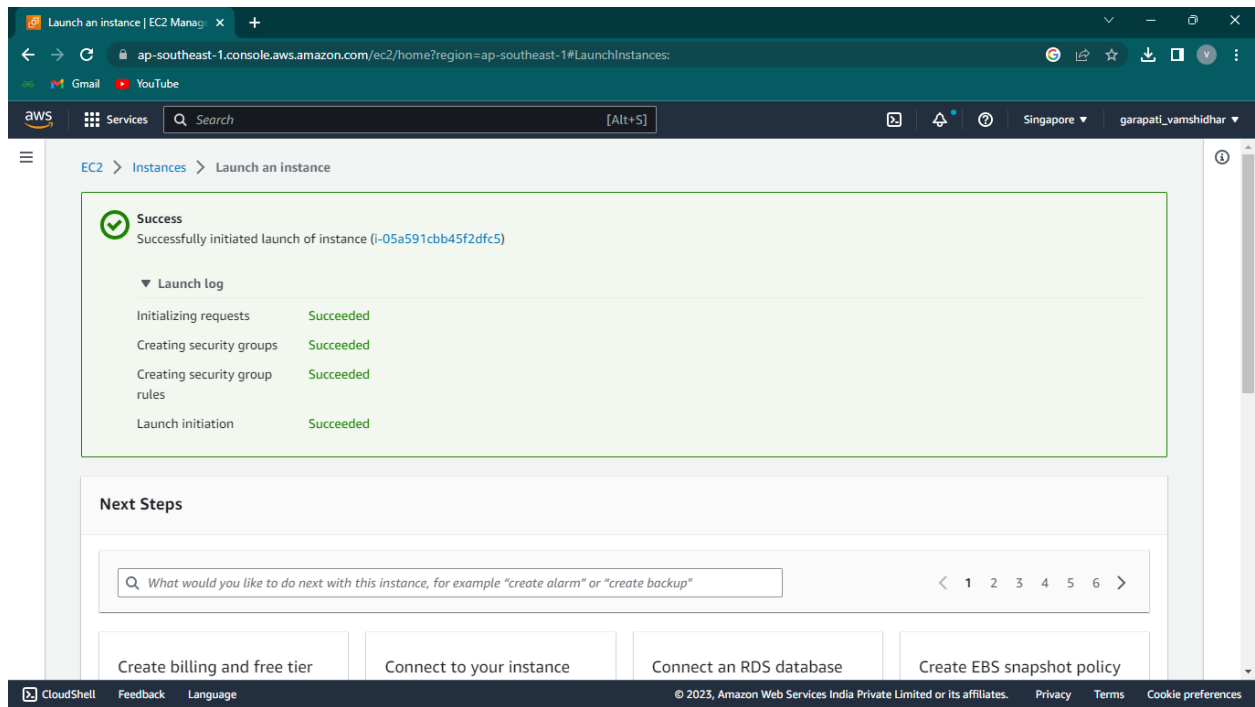
Firewall (security group) New security group

Storage (volumes) 1 volume(s) - 8 GiB

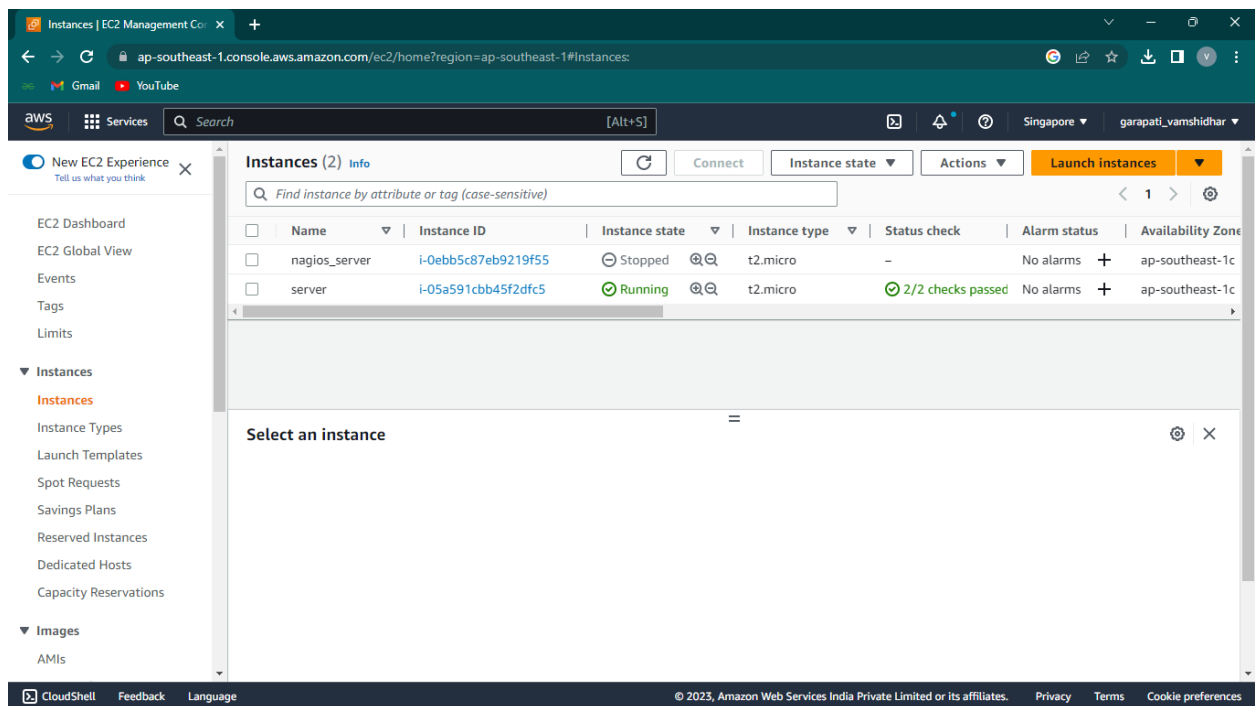
Cancel Launch instance [Review commands](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

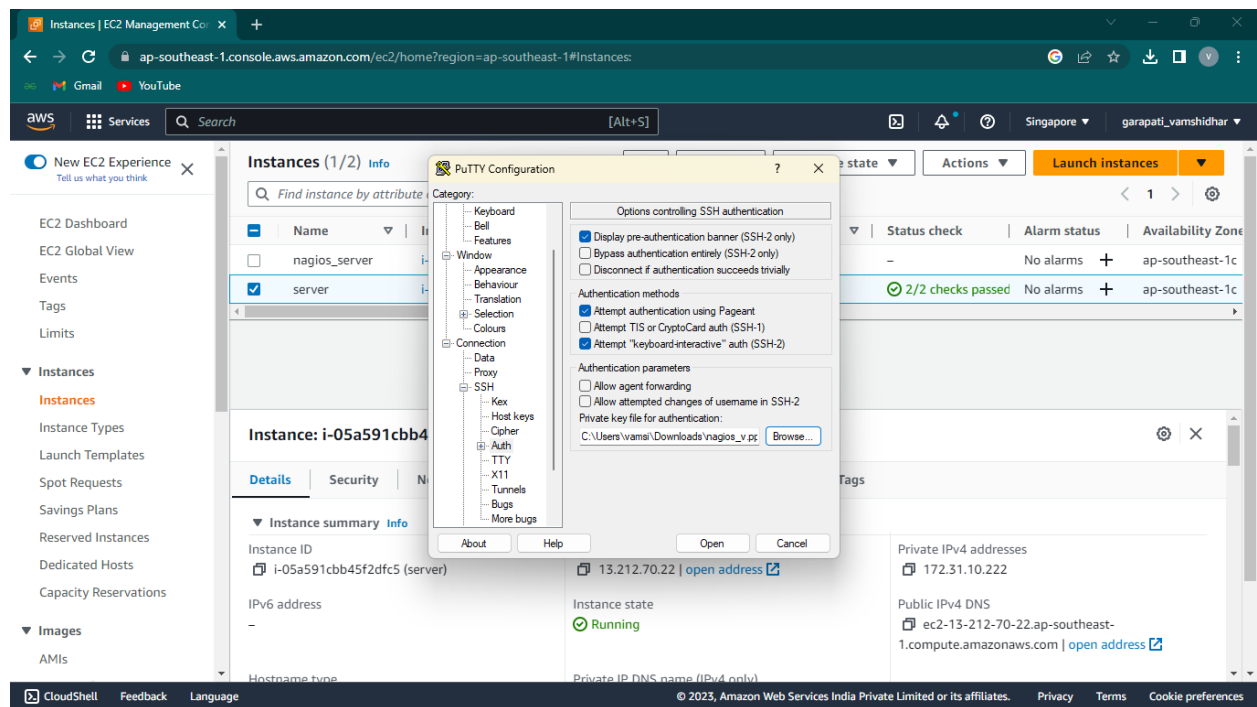
9.Instance is launched successfully



10.The instance is in running state



11. Connect to EC2 instance using putty



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
```

ec2-user@ip-172-31-10-222:~

login as: ec2-user

Authenticating with public key "nagios_v"

```
 _ | _ | _  
 _ | ( _ / _  
 _ | \ _ | _  
      Amazon Linux 2 AMI
```

<https://aws.amazon.com/amazon-linux-2/>

8 package(s) needed for security, out of 9 available

Run "sudo yum update" to apply all updates.

[ec2-user@ip-172-31-10-222 ~]\$ sudo yum install httpd php

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd

Resolving Dependencies

--> Running transaction check

---> Package httpd.x86_64 0:2.4.56-1.amzn2 will be installed

--> Processing Dependency: httpd-tools = 2.4.56-1.amzn2 for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: httpd filesystem = 2.4.56-1.amzn2 for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: system-logos-httpd for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: mod http2 for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: httpd filesystem for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: /etc/mime.types for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.56-1.amzn2.x86_64

--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.56-1.amzn2.x86_64

---> Package php.x86_64 0:5.4.16-46.amzn2.0.2 will be installed

--> Processing Dependency: php-cli(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-46.amzn2.0.2.x86_64

--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-46.amzn2.0.2.x86_64

--> Running transaction check

---> Package apr.x86_64 0:1.7.2-1.amzn2 will be installed

---> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed

--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64

---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed

---> Package httpd filesystem.noarch 0:2.4.56-1.amzn2 will be installed

---> Package httpd-tools.x86_64 0:2.4.56-1.amzn2 will be installed

---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed

---> Package mod http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed

---> Package php-cli.x86_64 0:5.4.16-46.amzn2.0.2 will be installed

---> Package php-common.x86_64 0:5.4.16-46.amzn2.0.2 will be installed

--> Processing Dependency: libzip.so.2()(64bit) for package: php-common-5.4.16-46.amzn2.0.2.x86_64

--> Running transaction check

---> Package apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 will be installed

---> Package libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

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  1.98MB/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
```

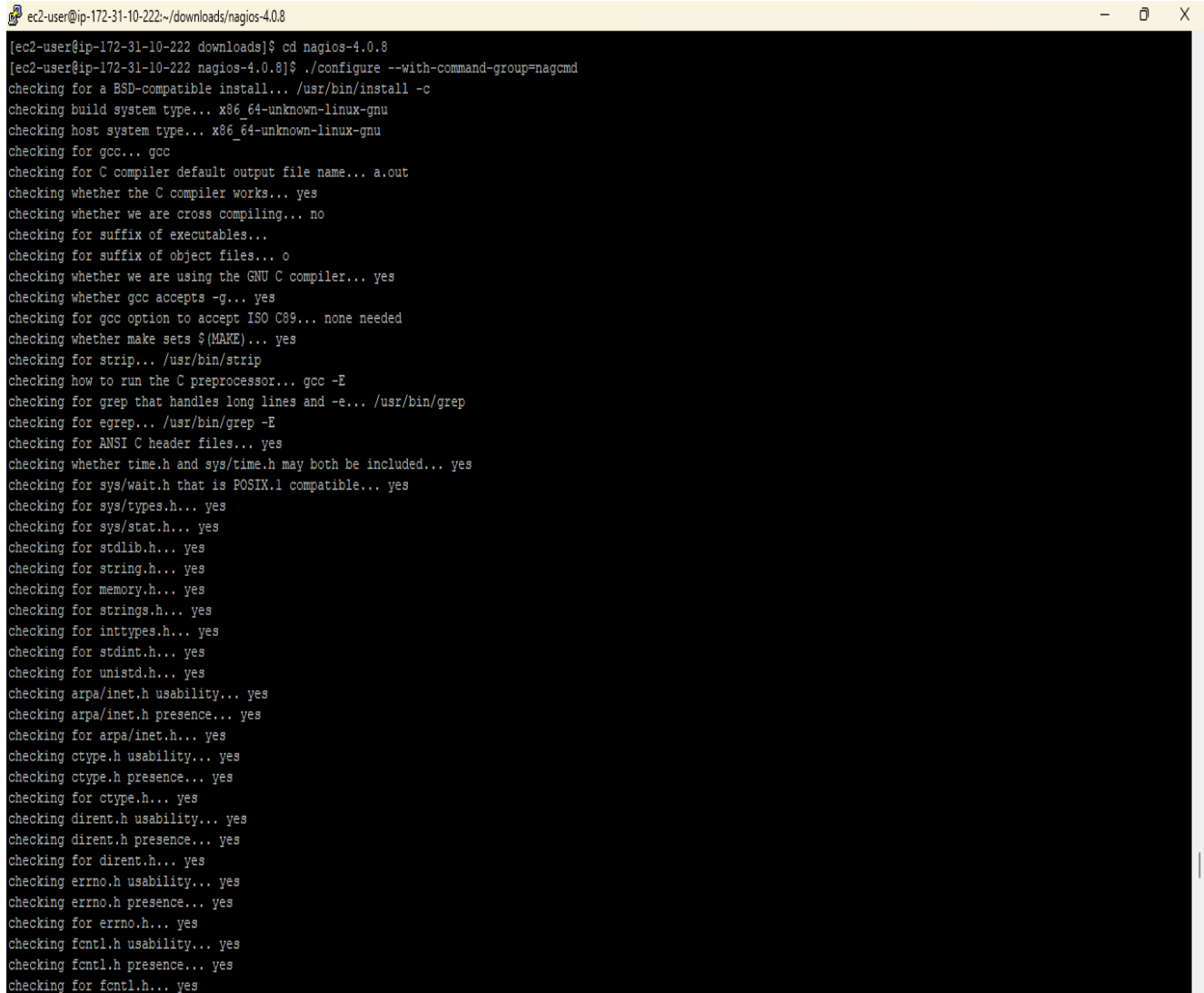
```
ec2-user@ip-172-31-10-222:~/downloads
[ec2-user@ip-172-31-10-222 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LLEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
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/nagiosstats.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



```
ec2-user@ip-172-31-10-222:~/downloads/nagios-4.0.8
[ec2-user@ip-172-31-10-222 downloads]$ cd nagios-4.0.8
[ec2-user@ip-172-31-10-222 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
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
checking ctype.h usability... yes
checking ctype.h presence... yes
checking for ctype.h... yes
checking dirent.h usability... yes
checking dirent.h presence... yes
checking for dirent.h... yes
checking errno.h usability... yes
checking errno.h presence... yes
checking for errno.h... yes
checking fcntl.h usability... yes
checking fcntl.h presence... yes
checking for fcntl.h... yes
```

```
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 nebmmods.o nebmmods.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

```
ec2-user@ip-172-31-10-222:~/downloads/nagios-4.0.8
Enjoy.

[ec2-user@ip-172-31-10-222 nagios-4.0.8]$ sudo make install
cd ./base && make install
make[1]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/base'
make install-basic
make[2]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[2]: Leaving directory `/home/ec2-user/downloads/nagios-4.0.8/base'
make strip-post-install
make[2]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/base'
/usr/bin/strip /usr/local/nagios/bin/nagios
/usr/bin/strip /usr/local/nagios/bin/nagiosstats
make[2]: Leaving directory `/home/ec2-user/downloads/nagios-4.0.8/base'
make[1]: Leaving directory `/home/ec2-user/downloads/nagios-4.0.8/base'
cd ./cgi && make install
make[1]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/cgi'
make install-basic
make[2]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory `/home/ec2-user/downloads/nagios-4.0.8/cgi'
make strip-post-install
make[2]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/cgi'
for file in *.cgi; do \
    /usr/bin/strip /usr/local/nagios/sbin/$file; \
done
make[2]: Leaving directory `/home/ec2-user/downloads/nagios-4.0.8/cgi'
make[1]: Leaving directory `/home/ec2-user/downloads/nagios-4.0.8/cgi'
cd ./html && make install
make[1]: Entering directory `/home/ec2-user/downloads/nagios-4.0.8/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
```


\$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

```
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.0.8'
[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.

```
ec2-user@ip-172-31-10-222:~/downloads/nagios-4.0.8
#####
# 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              garapativamshidhar2002@gmail.co ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****>>
}

#####
#
# CONTACT GROUPS
#
#####

"/usr/local/nagios/etc/objects/contacts.cfg" 54L, 2154B                                     34,72 Top
```

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-libcap: 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

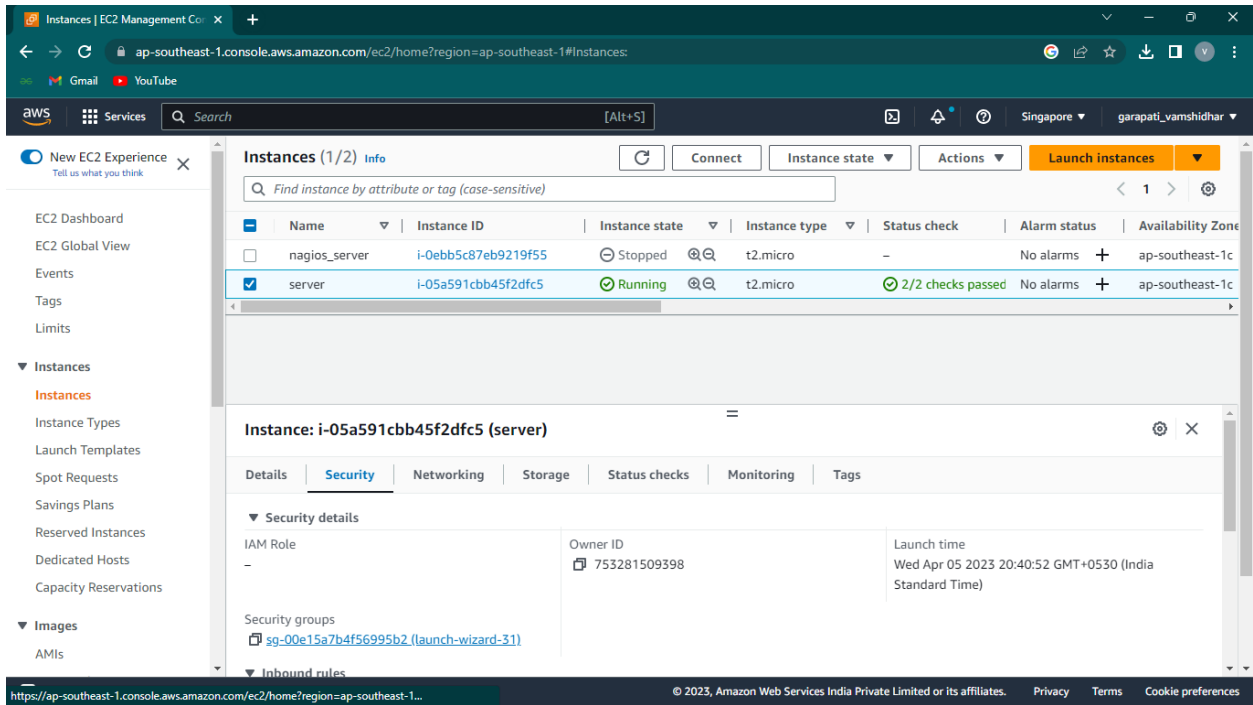
```
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$ sudo service nagios start
Starting nagios (via systemctl): [ OK ]
[ec2-user@ip-172-31-10-222 nagios-plugins-2.0.3]$
```

Step 9: Update AWS Security Group

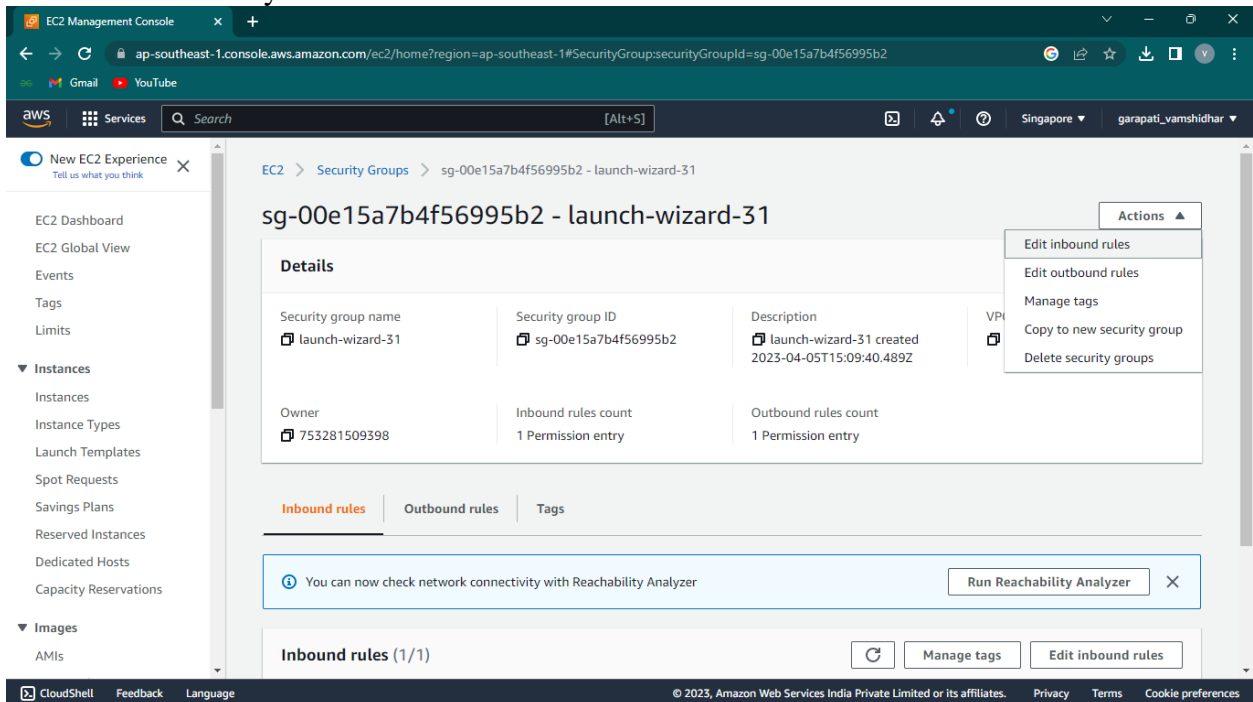
you need to open port 80 on the new AWS EC2 server to incoming traffic so you can connect to the new Nagios webpage.

- * 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.

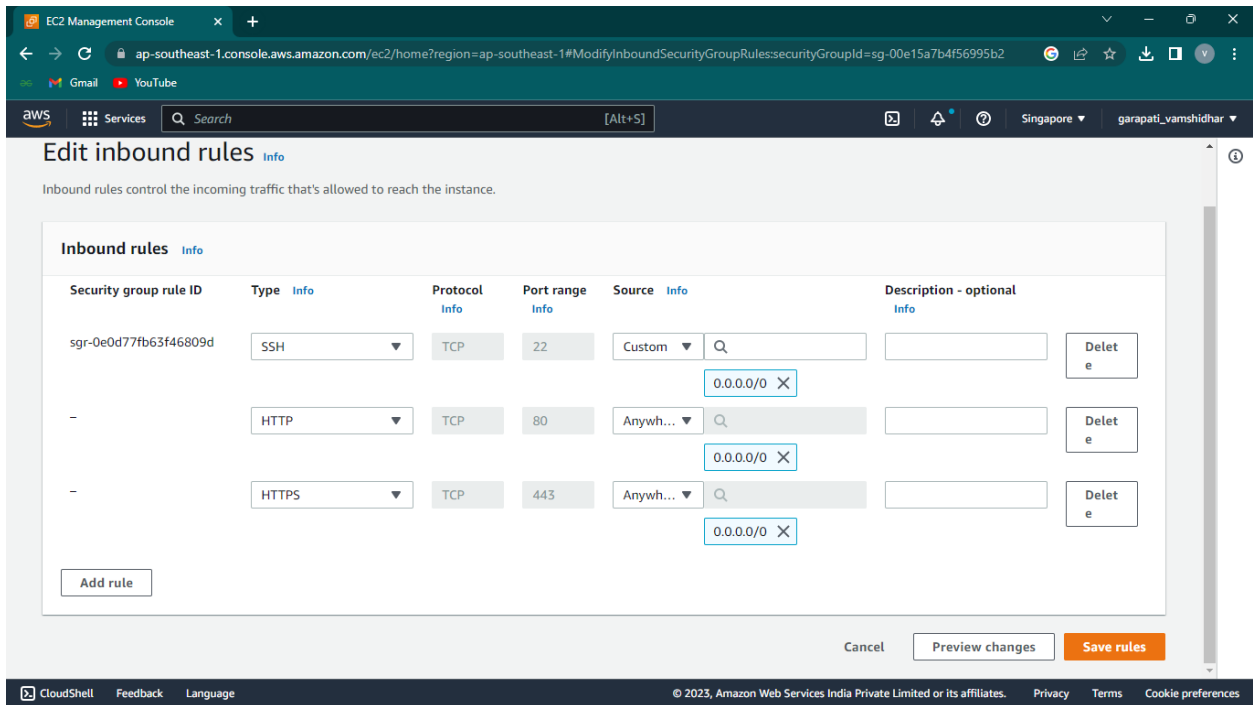
- Now go to EC2 instance



- Go to security



- Now edit the inbound rules



Step 10: Log in to the Web Interface

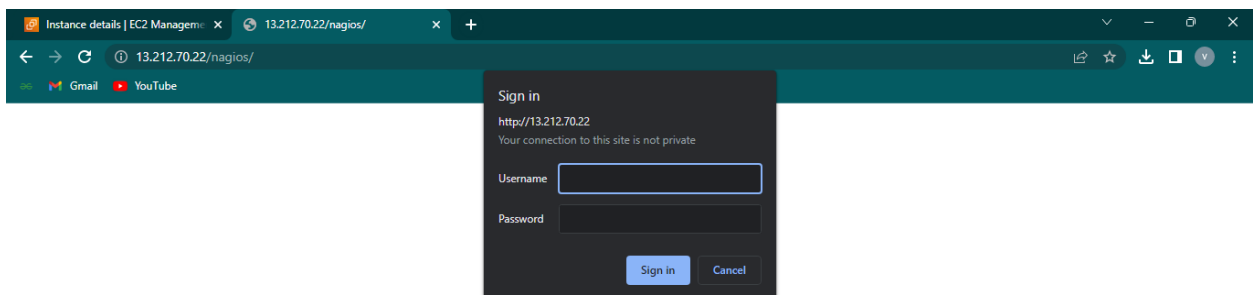
access the Nagios web interface to do this you will need to know the Public DNS or IP for your instance, you can get this from the Instance section of the EC2 Console if you do not already know it. You'll be prompted for the username (nagiosadmin) and password you specified earlier.

Eg:Ipaddress/nagios

Sign in to the Nagios by using username and password.

Username: nagiosadmin

Password: sai@1



- This is the Interface of the Nagios Core

Nagios® Core™
 ✓ Daemon running with PID 26219
Nagios® Core™ Version 4.0.8
 August 12, 2014
[Check for updates](#)

A new version of Nagios Core is available!
 Visit nagios.org to download Nagios 4.4.10.

Nagios XI
 Easy Configuration
 Advanced Reporting
[Download](#)

Nagios Log Server
 Monitor and analyze logs from anywhere
[Download](#)

Nagios Network Analyzer
 Real-time netflow and bandwidth analysis
[Download](#)

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

Don't Miss...

Nagios®

Current Network Status
 Last Updated: Wed Apr 5 15:36:07 UTC 2023
 Updated every 30 seconds
 Nagios® Core™ 4.0.8 - www.nagios.org
 Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0
All Problems		All Types	
0	1		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
2	1	0	0	5
All Problems		All Types		
1	8			

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
localhost	UP	04-05-2023 15:37:21	0d 0h 2m 1s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Results 1 - 1 of 1 Matching Hosts

Nagios Core Host Information

Last Updated: Wed Apr 5 15:38:18 UTC 2023
 Updated every 90 seconds
 Nagios® Core™ 4.0.8 - www.nagios.org
 Logged in as nagiosadmin

Host: localhost (localhost)
 Member of: linux-servers
 127.0.0.1

Host State Information

Host Status: **UP** (for 0d 0h 2m 12s)
 Status Information: PING OK - Packet loss = 0%, RTA = 0.04 ms
 Performance Data: rta=0.036000ms;3000.000000;5000.000000;0.000000 pi=0%;80;100.0
 Current Attempt: 1/10 (HARD state)
 Last Check Time: 04-05-2023 15:37:21
 Check Type: ACTIVE
 Check Latency / Duration: 0.000 / 4.091 seconds
 Next Scheduled Active Check: 04-05-2023 15:42:25
 Last State Change: 04-05-2023 15:36:06
 Last Notification: N/A (notification 0)
 Is This Host Flapping? **NO** (0.00% state change)
 In Scheduled Downtime? **NO**
 Last Update: 04-05-2023 15:38:08 (0d 0h 0m 10s ago)

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

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable checks of all services on this host
- Disable event handler for this host
- Disable flap detection for this host

Host Comments

Add a new comment Delete all comments

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it							

13.212.70.22/nagios/cgi-bin/cmd.cgi?cmd_type=96&host=localhost&force_check

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          netconf             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-            libnsl              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               setupool.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              inittab             mke2fs.conf          plymouth              rpm                 sudo.conf           yum.repos.d
cron.deny            GeoIP.conf           inputrc             modprobe.d           pm                    rsyslog.conf        sudoers             sudoers.d
cron.hourly          GeoIP.conf.default  iproute2            modules-load.d        popt.d               rsyslog.d           sudo-ldap.conf      sysconfig
cron.monthly         gnupg               iproute2            modules-load.d        postfix               rsyslog.d           sudo-ldap.conf      sysconfig
crontab              GREP_COLORS          issue               motd                 postfix               rwtab
```


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.

```
[root@ip-172-31-10-222 etc]# cd ssh
[root@ip-172-31-10-222 ssh]# ls
moduli  ssh_config  sshd_config  ssh_host_ecdsa_key  ssh_host_ecdsa_key.pub  ssh_host_ed25519_key  ssh_host_ed25519_key.pub  ssh_host_rsa_key  ssh_host_rsa_key.pub
```

```
[root@ip-172-31-10-222 ssh]# nano sshd_config
[root@ip-172-31-10-222 ssh]#
```

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text
```

```
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to
PasswordAuthentication yes
#PermitEmptyPasswords no

^G Get Help      ^O Write Out    ^W Where Is     ^K Cu
^X Exit          ^R Read File    ^\ Replace      ^U Un
```

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

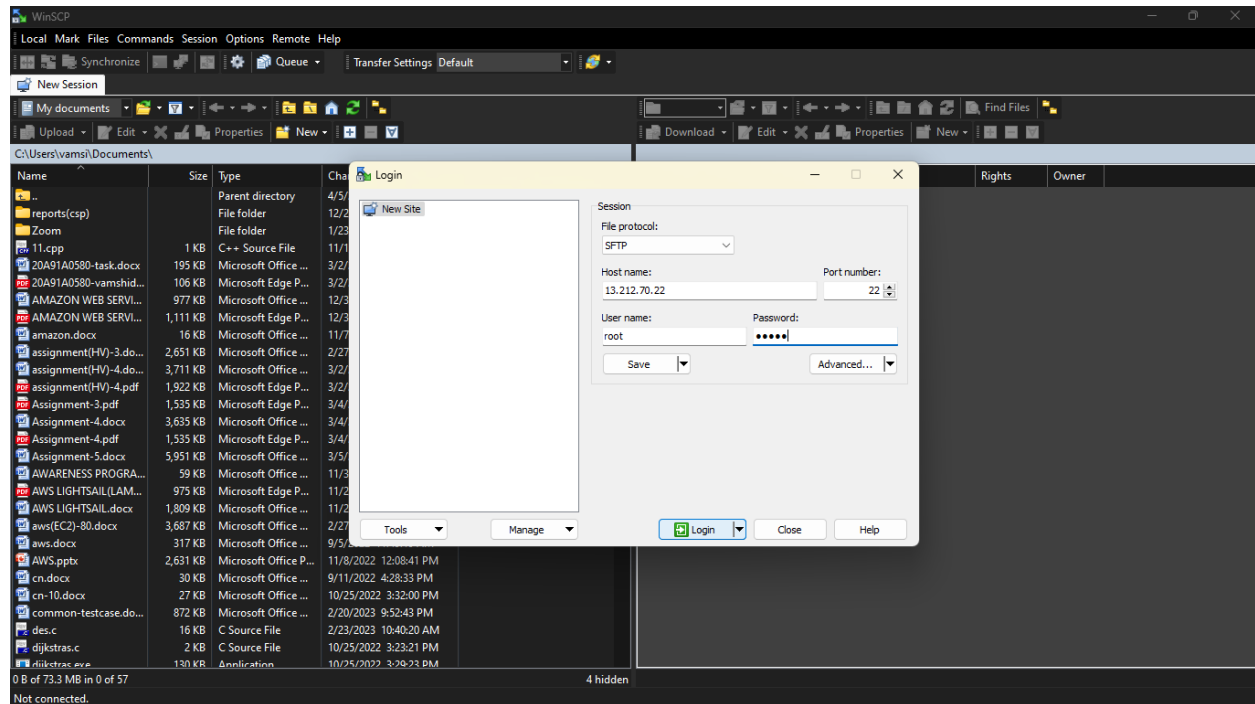
```
$passwd root
```

```
$systemctl restart sshd
```

```
[root@ip-172-31-10-222 ssh]# nano sshd_config
[root@ip-172-31-10-222 ssh]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 6 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-10-222 ssh]#
[root@ip-172-31-10-222 ssh]# systemctl restart sshd
[root@ip-172-31-10-222 ssh]#
```

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/

/usr/local/nagios/etc/objects/					
Name	Size	Changed	Rights	Owner	
..		4/5/2023 8:58:26 PM	rw-rwxr-x	nagios	
commands.cfg	8 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	
contacts.cfg	3 KB	4/5/2023 8:57:42 PM	rw-rw-r--	nagios	
localhost.cfg	6 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	
printer.cfg	4 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	
switch.cfg	4 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	
templates.cfg	11 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	
timeperiods.cfg	4 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	
windows.cfg	4 KB	4/5/2023 8:56:07 PM	rw-rw-r--	nagios	

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)

```
#####
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
#
# NOTE: This config file is intended to serve as an *extremely* simple
#       example of how you can create configuration entries to monitor
#       the local (Linux) machine.
#
#####

#####
#
# HOST DEFINITION
#
#####

# Define a host for the local machine

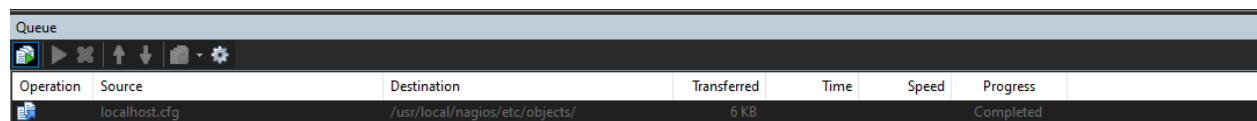
define host{
    use                linux-server          ; Name of host template to use
                                           ; This host definition will inherit all variables that are defined
                                           ; in (or inherited by) the linux-server host template definition.

    host_name          localhost
    alias              localhost
    address             210.212.210.86
}

#####
#
# HOST GROUP DEFINITION
#
#####
```

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]#
```

Instance details | EC2 Management

Nagios Core

← → ↻ ⚠ Not secure | 13.212.70.22/nagios/

Gmail

YouTube

Nagios®

General

Home

Documentation

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary

Grid

Service Groups

Summary

Grid

Problems

Services

(Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Availability

Trends

Alerts

History

Summary

Histogram

Notifications

Event Log

System

Comments

Downtime

Process Info

Performance Info

Host Information

Last Updated: Wed Apr 5 15:49:43 UTC 2023

Updated every 30 seconds

Nagios® Core™ 4.0.8 - www.nagios.org

Logged in as nagiosadmin

Host

localhost

(localhost)

Member of

linux-servers

210.212.210.86

Host State Information

Host Status: UP (for 0d 0h 13m 37s)

Status Information: PING OK - Packet loss = 0%, RTA = 0.04 ms

Performance Data: rta=0.041000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0

Current Attempt: 1/10 (HARD state)

Last Check Time: 04-05-2023 15:47:51

Check Type: ACTIVE

Check Latency / Duration: 0.000 / 4.106 seconds

Next Scheduled Active Check: 04-05-2023 15:52:55

Last State Change: 04-05-2023 15:36:06

Last Notification: N/A (notification 0)

Is This Host Flapping? NO (0.00% state change)

In Scheduled Downtime? NO

Last Update: 04-05-2023 15:49:41 (0d 0h 0m 2s ago)

Active Checks: ENABLED

Passive Checks: ENABLED

Obsessing: ENABLED

Notifications: ENABLED

Event Handler: ENABLED

Flap Detection: ENABLED

Host Commands

📍

Locate host on map

✖

Disable active checks of this host

🕒

Re-schedule the next check of this host

?

Submit passive check result for this host

✖

Stop accepting passive checks for this host

✖

Stop obsessing over this host

✖

Disable notifications for this host

✖

Send custom host notification

🕒

Schedule downtime for this host

🕒

Schedule downtime for all services on this host

✖

Disable notifications for all services on this host

✓

Enable notifications for all services on this host

🕒

Schedule a check of all services on this host

✖

Disable checks of all services on this host

✓

Enable checks of all services on this host

✖

Disable event handler for this host

✖

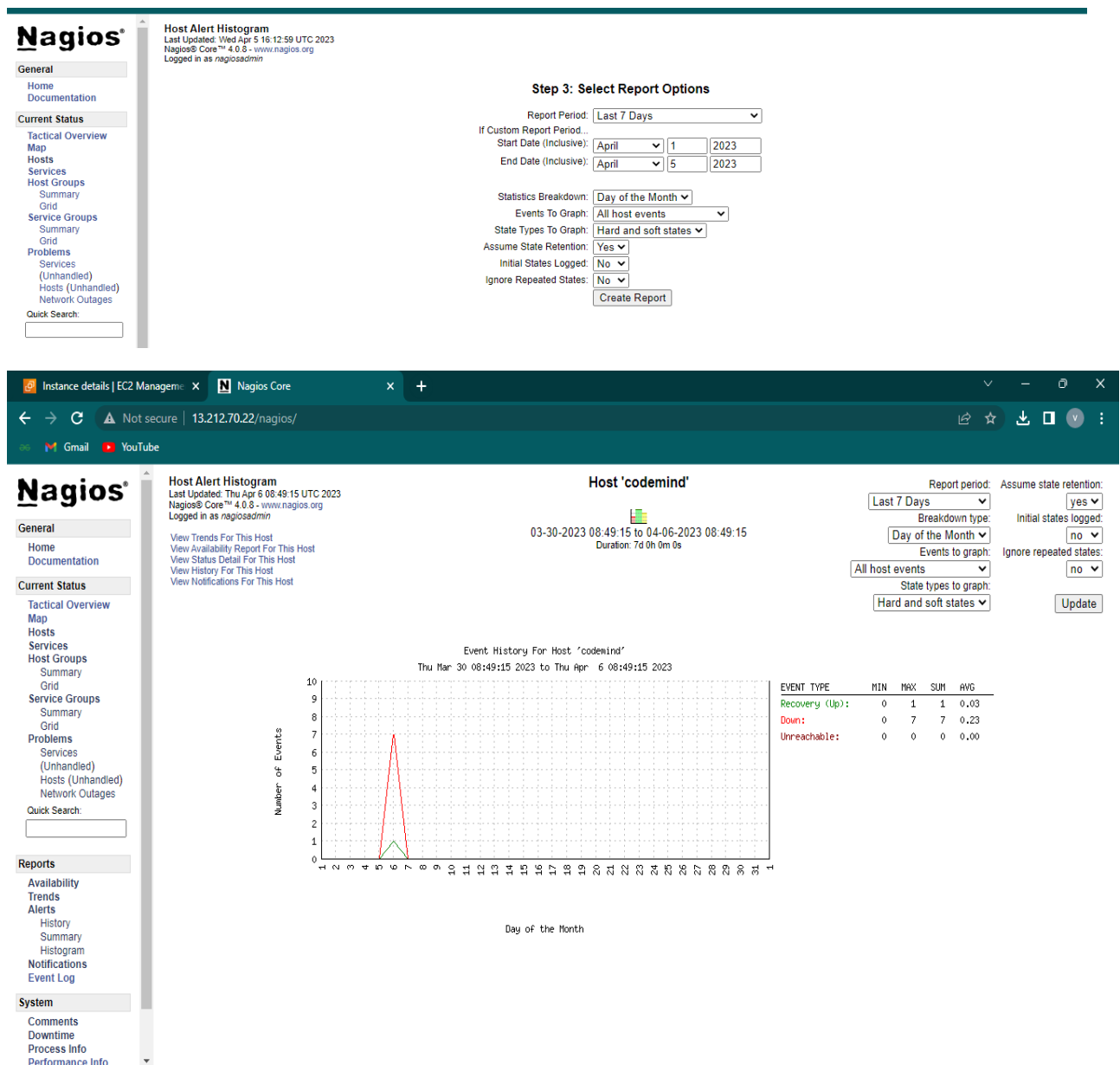
Disable flap detection for this host

Host Comments

🗨 Add a new comment 🗑 Delete all comments

Entry Time	Author	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it						

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



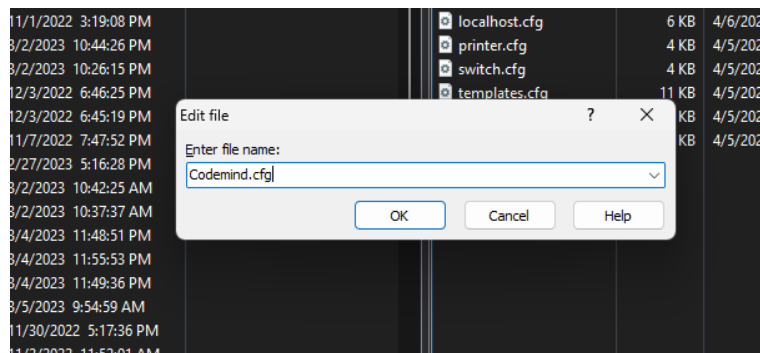
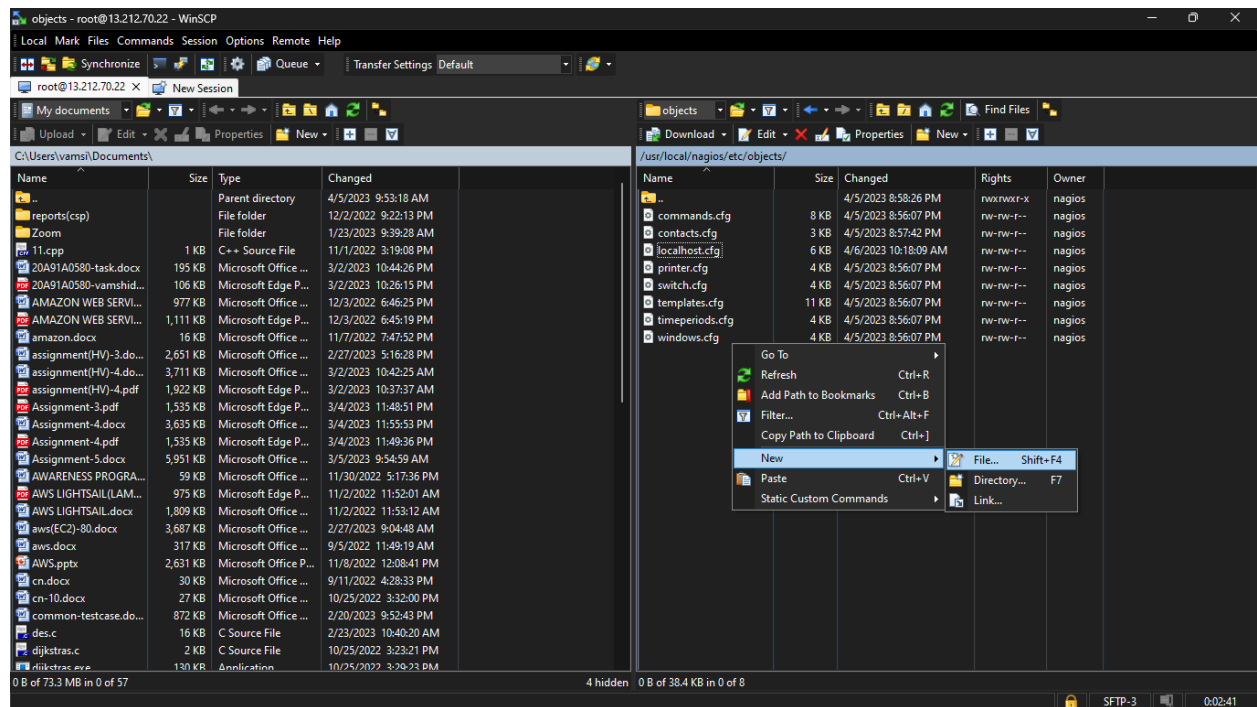
Adding hosts and monitoring:

1. Switch to the path below:

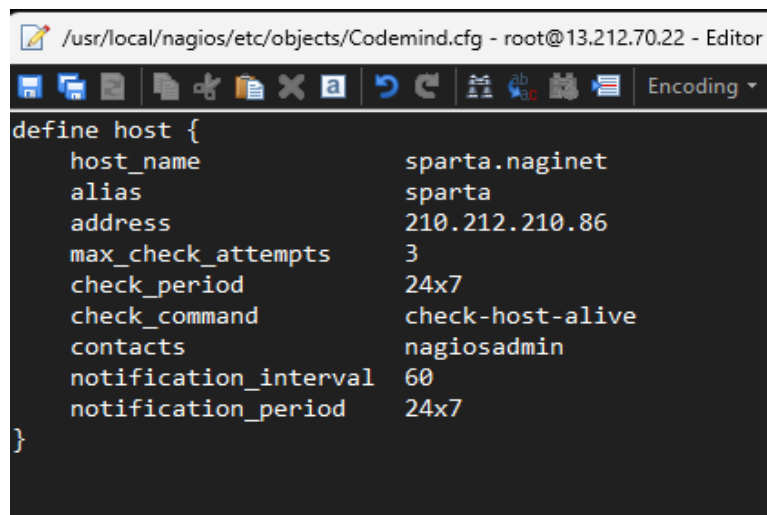
Path: `/usr/local/nagios/etc/objects/`

2. create a new file with `.cfg` extension.

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
```

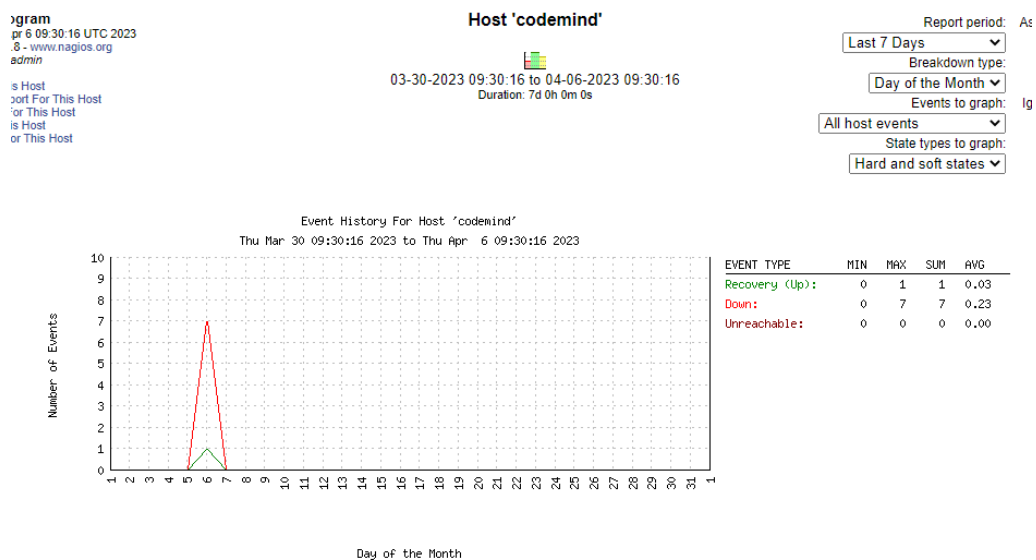
```
[root@ip-172-31-10-222 ssh]# systemctl restart nagios
[root@ip-172-31-10-222 ssh]#
```

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

The screenshot shows the Nagios Core web interface. The left sidebar contains navigation links like General, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Grid, Problems, and Services (Unhandled). The main content area displays 'Host Status Details For All Host Groups'. It includes a table with columns: Host, Status, Last Check, Duration, and Status Information. The table lists two hosts: 'codemind' and 'localhost', both with a status of 'UP'. The 'codemind' host has a last check of '04-06-2023 09:27:26' and a duration of '0d 4h 24m 20s'. The 'localhost' host has a last check of '04-06-2023 09:18:56' and a duration of '0d 4h 40m 0s'. Above the table, there are summary statistics for Host Status Totals and Service Status Totals.

Host	Status	Last Check	Duration	Status Information
codemind	UP	04-06-2023 09:27:26	0d 4h 24m 20s	PING OK - Packet loss = 0%, RTA = 65.42 ms
localhost	UP	04-06-2023 09:18:56	0d 4h 40m 0s	PING OK - Packet loss = 0%, RTA = 208.64 ms



6. In the histograms we can observe the graph.



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

01:59 UTC 2023	Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
nagios.org	2	0	0	0	5	2	0	1	0
	All Problems		All Types		All Problems		All Types		
For All Host Groups	0		2		3		8		
All Host Groups									
All Host Groups									
All Host Groups									

Status Grid For All Host Groups

Linux Servers (linux-servers)									
Host	Services								Actions
localhost	Current Load	Current Users	HTTP	PING	Root Partition	SSH	Swap Usage	Total Processes	  

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

HTTP status:

3.91.231.253

Service State Information

Current Status:	WARNING (for 0d 4h 45m 20s)
Status Information:	HTTP WARNING: HTTP/1.1 403 Forbidden - 3932 bytes in 0.418 second response time
Performance Data:	time=0.418149s;;;0.000000 size=3932B;;;0
Current Attempt:	4/4 (HARD state)
Last Check Time:	04-06-2023 09:31:11
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 0.420 seconds
Next Scheduled Check:	04-06-2023 09:36:11
Last State Change:	04-06-2023 04:48:19
Last Notification:	04-06-2023 09:19:06 (notification 5)
Is This Service Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	04-06-2023 09:33:35 (0d 0h 0m 4s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Service Comments

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:

3.91.231.253

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**