

Name : SHIVEN BANSAL

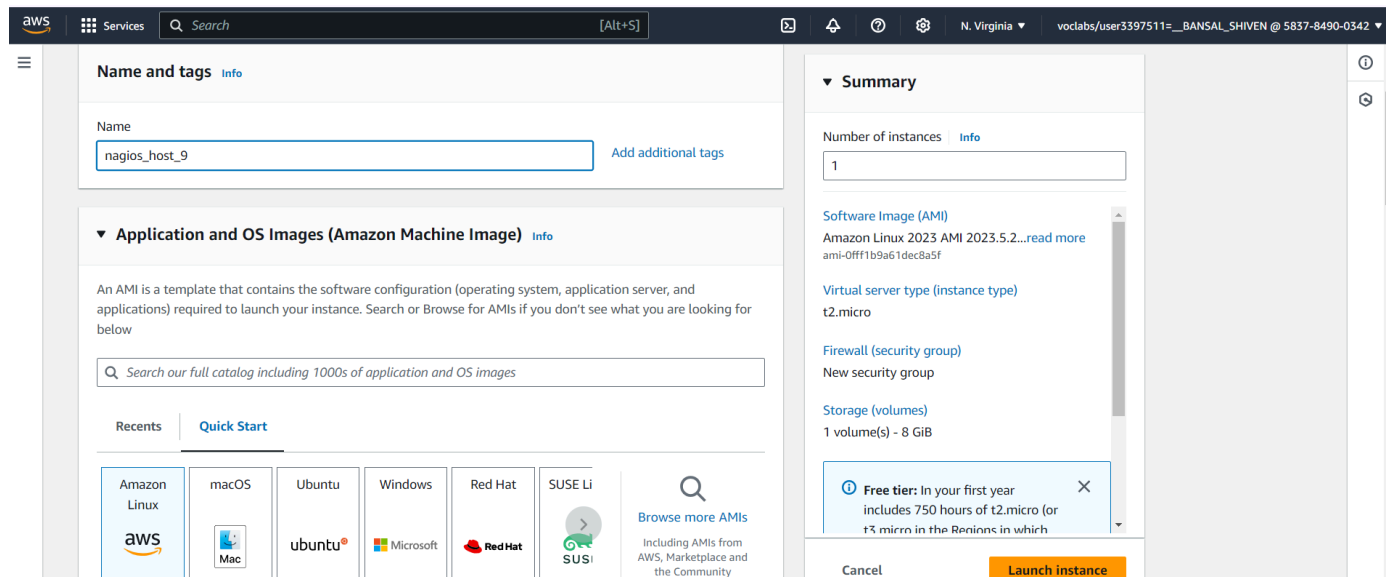
Roll no : 03

Class : D15C

## Experiment-9

Steps to perform the experiment:

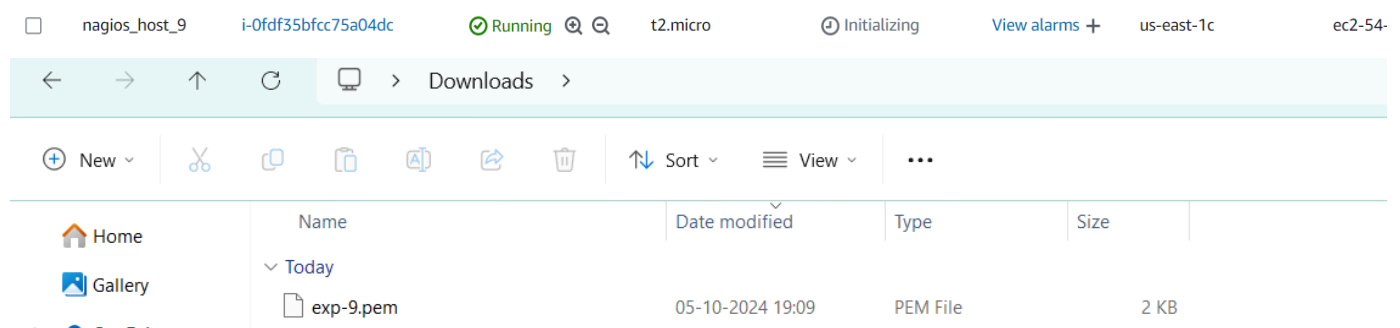
Step1) Create an EC2 instance. keep the settings as default.



Create a new key pair login and save the downloaded file in a folder of your local desktop.

Also create a new security group. In my case its name will 'launch-wizard-10'.

Later we will edit rules of this security group.



Now to edit security groups, select your security group and click on edit inbound rules. Add these security rules.

The screenshot shows the AWS Management Console interface for editing a security group. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. The main content area is titled 'Security Groups (1/1)' and shows a table with one security group: 'launch-wizard-10' with ID 'sg-0d09ee3439417acdb'. Below the table, the 'Inbound rules' tab is selected, showing a single rule for SSH on port 22. The 'Edit inbound rules' button is visible in the top right of the rule list.

The screenshot shows the 'Inbound rules' page for the security group 'sgr-0ddfc18ee62c79c7c'. The page displays a list of rules with columns for Type, Protocol, Port range, Source, and Description. The rules are as follows:

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere-I...	
All ICMP - IPv6	IPv6 ICMP	All	Anywhere-I...	
HTTPS	TCP	443	Anywhere-I...	
All traffic	All	All	Anywhere-I...	
Custom TCP	TCP	5666	Anywhere-I...	
All ICMP - IPv4	ICMP	All	Anywhere-I...	
SSH	TCP	22	Anywhere-I...	

Each rule has a 'Delete' button next to it. The 'SSH' rule is highlighted with a blue border. At the bottom left, there is an 'Add rule' button.

✓ Inbound security group rules successfully modified on security group (sg-0d09ee3439417acdb | launch-wizard-10)

► Details

## Security Groups (13) [Info](#)



Actions ▼

Export security groups to CSV ▼

Create security group

Q Find resources by attribute or tag

< 1 >

<input type="checkbox"/>	Name ▼	Security group ID ▼	Security group name ▼	VPC ID ▼	Description
<input type="checkbox"/>	aws-cloud9-Shiven-...	<a href="#">sg-074bfe8c35f05a239</a>	aws-cloud9-Shiven-Bansal-547c58c60...	<a href="#">vpc-0381e49e607677b63</a> <a href="#">↗</a>	Secu
<input type="checkbox"/>	-	<a href="#">sg-092cec49b153d08cc</a>	default	<a href="#">vpc-0381e49e607677b63</a> <a href="#">↗</a>	defa
<input type="checkbox"/>	-	<a href="#">sg-0e61a434621cedc90</a>	master-new	<a href="#">vpc-0381e49e607677b63</a> <a href="#">↗</a>	exp2
<input type="checkbox"/>	-	<a href="#">sg-019a8d4e4864ce132</a>	launch-wizard-4	<a href="#">vpc-0381e49e607677b63</a> <a href="#">↗</a>	laun
<input type="checkbox"/>	-	<a href="#">sg-0d09ee3439417acdb</a>	launch-wizard-10	<a href="#">vpc-0381e49e607677b63</a> <a href="#">↗</a>	laun
<input type="checkbox"/>	-	<a href="#">sg-08e9d1a1d0863f1c0</a>	launch-wizard-5	<a href="#">vpc-0381e49e607677b63</a> <a href="#">↗</a>	laun

now navigate to instances, click on the instance which was created earlier and click on connect.  
now copy the ssh command and just replace the .pem file with its actual location in your computer.

[EC2](#) > [Instances](#) > [i-0fdf35bfcc75a04dc](#) > Connect to instance

## Connect to instance [Info](#)

Connect to your instance i-0fdf35bfcc75a04dc (nagios\_host\_9) using any of these options

EC2 Instance Connect

Session Manager

**SSH client**

EC2 serial console

Instance ID

[i-0fdf35bfcc75a04dc](#) (nagios\_host\_9)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp-9.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "exp-9.pem"`
4. Connect to your instance using its Public DNS:  
 `ec2-54-158-150-185.compute-1.amazonaws.com`

Example:

`ssh -i "exp-9.pem" ec2-user@ec2-54-158-150-185.compute-1.amazonaws.com`

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

```
C:\Users\ADMIN\Downloads>ssh -i "exp-9.pem" ec2-user@ec2-54-158-150-185.compute-1.amazonaws.com
The authenticity of host 'ec2-54-158-150-185.compute-1.amazonaws.com (54.158.150.185)' can't be established.
ED25519 key fingerprint is SHA256:MLHx7pxctPRettGDRiKHoksaoK0tmmsQsZs1BuQKnQeo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-158-150-185.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```

```
sudo yum update
```

```
sudo yum install httpd php
```

Select y when asked i prompt.

Package	Architecture	Version	Repository	Size
<b>Installing:</b>				
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
php8.3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	10 M
<b>Installing dependencies:</b>				
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux	176 k
libxslt	x86_64	1.1.34-5.amzn2023.0.2	amazonlinux	241 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
nginx-filesystem	noarch	1:1.24.0-1.amzn2023.0.4	amazonlinux	9.8 k
php8.3-cli	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	3.7 M
php8.3-common	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	737 k
php8.3-process	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	45 k
php8.3-xml	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	154 k
<b>Installing weak dependencies:</b>				
apr-util-openssl	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	17 k
mod_http2	x86_64	2.0.27-1.amzn2023.0.3	amazonlinux	166 k
mod_lua	x86_64	2.4.62-1.amzn2023	amazonlinux	61 k
php8.3-fpm	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	1.9 M
php8.3-mbstring	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	528 k
php8.3-opcache	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	379 k
php8.3-pdo	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	89 k
php8.3-sodium	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	41 k

```
Transaction Summary
=====
Install 25 Packages
```

## sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-47-54 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:15:43 ago on Sat Oct 5 13:40:09 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
```

Package	Architecture	Version	Repository
Installing:			
gcc	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux
Installing dependencies:			
annobin-docs	noarch	10.93-1.amzn2023.0.1	amazonlinux
annobin-plugin-gcc	x86_64	10.93-1.amzn2023.0.1	amazonlinux
cpp	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux
gc	x86_64	8.0.4-5.amzn2023.0.2	amazonlinux
glibc-devel	x86_64	2.34-52.amzn2023.0.11	amazonlinux
glibc-headers-x86	noarch	2.34-52.amzn2023.0.11	amazonlinux
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux
kernel-headers	x86_64	6.1.109-118.189.amzn2023	amazonlinux
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux
libxcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux
Transaction Summary			
Install 13 Packages			

## sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-47-54 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:16:31 ago on Sat Oct 5 13:40:09 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	S
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	13
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	3
Installing dependencies:				
brotnli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31
brotnli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	3
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	21
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	68
cmake-filessystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	1
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	27
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	12
fonts-filessystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	42
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	91
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	48
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	1

## sudo adduser -m

## nagios sudo passwd

```
[ec2-user@ip-172-31-47-54 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-47-54 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-47-54 ~]$ |
```

## nagios

## sudo groupadd nagcmd

```
[ec2-user@ip-172-31-47-54 ~]$ sudo groupadd nagcmd
```

```
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

```
[ec2-user@ip-172-31-47-54 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-47-54 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-47-54 ~]$ |
```

```
mkdir ~/downloads
cd ~/downloads
```

```
[ec2-user@ip-172-31-47-54 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-47-54 ~]$ cd ~/downloads
[ec2-user@ip-172-31-47-54 downloads]$ |
```

wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz>

```
[ec2-user@ip-172-31-47-54 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-05 14:08:52-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz          100%[=====] 1.97M  5.22MB/s  in 0.4s

2024-10-05 14:08:53 (5.22 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-47-54 downloads]$ |
```

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

```
[ec2-user@ip-172-31-47-54 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-05 14:09:32-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M  7.35MB/s  in 0.4s

2024-10-05 14:09:33 (7.35 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-47-54 downloads]$ |
```

tar xzvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-47-54 downloads]$ tar xzvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LLEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
```

Now we have to first navigate to the nagios-4.5.5 folder in downloads.

- commands to enter:

ls (verify whether nagios-4.5.5 exists). Then go inside nagios 4.5.5 using cd.

```
[ec2-user@ip-172-31-47-54 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-47-54 downloads]$ cd nagios-4.5.5/
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ |
```

### we now have to install openssl dev library

The OpenSSL development library, or openssl-devel contains include files that help develop applications that use cryptographic algorithms and protocols

- commands to enter:

sudo yum install openssl-devel

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:31:58 ago on Sat Oct  5 13:40:09 2024.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
openssl-devel                          x86_64            1:3.0.8-1.amzn2023.0.14  amazonlinux       3.0 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm                                31 MB/s | 3.0 MB  00:00
-----
Total                                                                    21 MB/s | 3.0 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                1/1
  Installing    : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                1/1
  Verifying     : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                1/1

Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64

Complete!
```

Then finally we can run the commands like usual.

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

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
```

## make all

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmodes.o nebmodes.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
    inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
   253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
       |         ^~~~~~
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
```

### \*\*\* Support Notes \*\*\*\*\*

If you have questions about configuring or running Nagios, please make sure that you:

- Look at the sample config files
- Read the documentation on the Nagios Library at:  
<https://library.nagios.com>

before you post a question to one of the mailing lists. Also make sure to include pertinent information that could help others help you. This might include:

- What version of Nagios you are using
- What version of the plugins you are using
- Relevant snippets from your config files
- Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

\*\*\*\*\*

Enjoy.

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
```



Now the next command will take us to nano editor:

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

```
GNU nano 5.8 /usr/local/nagios/etc/objects/contacts.cfg
#####
# 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            nagios@localhost ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}

#####
[ Read 51 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-J To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_/ Go To Line  M-E Redo     M-G Copy     ^Q Where Was
```

Change your email

```
define contact {
    contact_name    nagiosadmin        ; Short name of user
    use              generic-contact    ; Inherit default values from generic-contact template (defined above)
    alias            Nagios Admin       ; Full name of user
    email            2022.shiven.bansal@ves.ac.in ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}
```

Press ctrl + O and  
enter Press ctrl + X

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ |
```

sudo make install-webconf

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf;
fi

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

## Adding password for nagios admin

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

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ |
```

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ |
```

```
cd ~/downloads
```

```
tar zxvf nagios-plugins-2.4.11.tar.gz
```

```
[ec2-user@ip-172-31-47-54 nagios-4.5.5]$ cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
```

```
cd nagios-plugins-2.4.11
```

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

```
[ec2-user@ip-172-31-47-54 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
```

make

sudo make install

```
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ make
sudo make install
make all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/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
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/, $p' \
  < ../build-aux/snippet/c++defs.h \
  > c++defs.h-t && \
```

sudo chkconfig --add nagios

sudo chkconfig nagios on

```
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ |
```

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

```
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ |
```

sudo service nagios start

```
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$
```

sudo systemctl status nagios

```
Redirecting to /bin/systemctl status nagios.service
[ec2-user@ip-172-31-47-54 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-10-05 14:36:11 UTC; 22s ago
     Docs: https://www.nagios.org/documentation
   Process: 65322 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 65323 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Main PID: 65324 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 5.7M
      CPU: 80ms
   CGroup: /system.slice/nagios.service
           └─65324 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           └─65325 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─65326 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─65327 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─65328 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─65329 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

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

EC2 > Instances > i-0fdf35bfcc75a04dc

### Instance summary for i-0fdf35bfcc75a04dc (nagios\_host\_9)

Updated less than a minute ago

Instance ID  
i-0fdf35bfcc75a04dc (nagios\_host\_9)

IPv6 address  
-

Public IPv4 address  
54.158.150.185 | open address

Instance state  
Running

Lastly, go to your web browser and type "http://nagios" Replace public-IPv4-address with the public ip address of your instance which you copied. You will get a prompt to enter the username and password that have been set for nagios admin

← → ↻ 54.158.150.185/nagios

Sign in

http://54.158.150.185

Your connection to this site is not private

Username

Password

Sign in Cancel

**Nagios**

General  
Home  
Documentation

Current Status  
Tactical Overview  
Map  
Hosts  
Services  
Host Groups  
Grid  
Service Groups

Problems  
Services  
Hosts  
Network Outages

Reports  
Availability  
Trends  
Alerts  
History  
Summary  
Histogram  
Notifications  
Event Log

System  
Comments  
Downtime  
Process Info  
Performance Info  
Scheduling Queue

**Nagios Core**  
✓ Daemon running with PID 65324

**Nagios Core™**  
Version 4.5.5  
September 17, 2024  
[Check for updates](#)

**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...**

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered trademarks owned by Nagios Enterprises, LLC. Use of the Nagios logo is governed by the trademark use regulations.

## Conclusion:

In this experiment, we installed and configured Nagios Core, Nagios Plugins, and NRPE on a Linux system to enable continuous monitoring. Nagios is a vital tool in the DevOps ecosystem, providing real-time detection of network and server issues, which helps ensure the health of the infrastructure. Its scalability, robust security, and automated alerting system enhance the effectiveness of monitoring. By integrating NRPE, we extended Nagios' capabilities to monitor remote hosts, enabling proactive issue resolution. With its highly customizable architecture and extensive plugin support, Nagios proves to be indispensable for maintaining service uptime and operational stability.