

Adv DevOps Practical 9

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

Theory:

What is Nagios?

Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately.

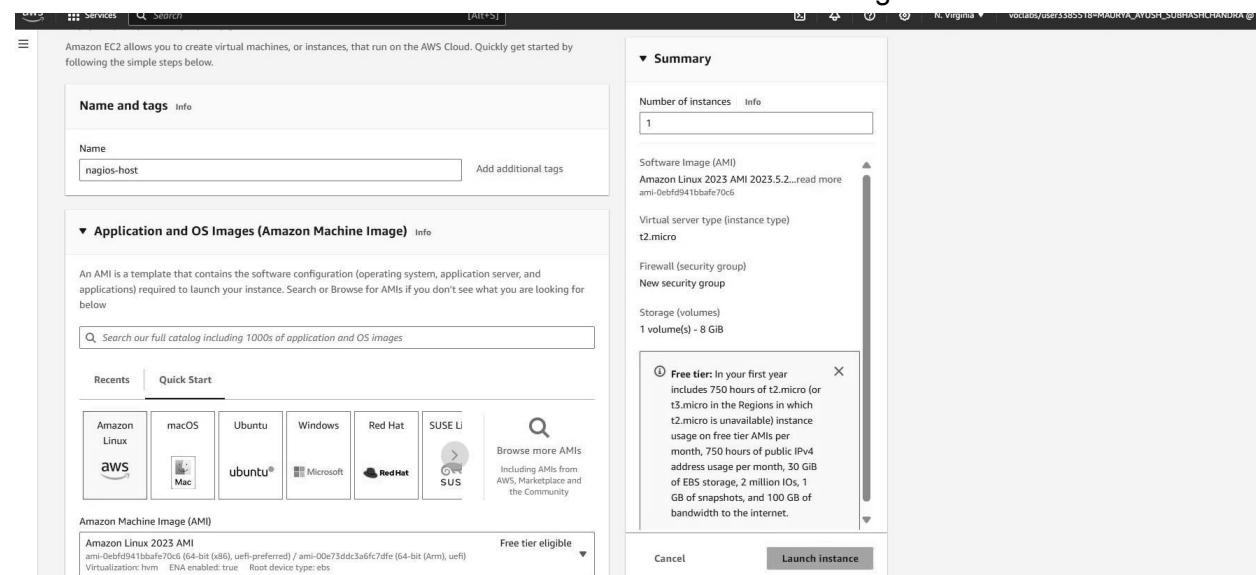
Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture

Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

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



The screenshot shows the AWS EC2 instance creation wizard. The first step, "Instance type", shows a selection for a t2.micro instance. It includes details like Family: t2, 1 vCPU, 1 GiB Memory, and Current generation: true. Pricing information indicates it's Free tier eligible. The second step, "Key pair (login)", shows a dropdown for "exp_09" and a link to "Create new key pair". A note says "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." The third step, "Security group", shows a note about creating a new security group named "launch-wizard-3" with rules for SSH and HTTPS. It also includes a warning about allowing all IP addresses. The fourth step, "Firewall (security group)", shows a new security group named "New security group" with one rule: "Allow SSH traffic from Anywhere (0.0.0.0/0)". A note states that the free tier includes 750 hours of t2.micro usage. The fifth step, "Launch instances", shows a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv6. One instance, "nagios-host", is running. The sixth step, "Images", lists security groups assigned to the instance, including "sg-09d51590eb1851b46 (launch-wizard-3)". The seventh step, "Elastic Block Store", is partially visible.

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

The screenshot shows the AWS Security Groups page. It displays a list of six security groups: "launch-wizard-2", "NodeGroup", "launch-wizard-1", "default", "MasterGroup", and "launch-wizard-3". Each entry includes fields for Name, Security group ID, Security group name, VPC ID, Description, and Owner. The "Description" column shows notes like "launch-wizard-2 created 2024-09-27T..." and "launch-wizard-3 created 2024-09-29T...". The "Owner" column shows the ID 217253764927 for all groups except "NodeGroup". The "Actions" button at the top right allows for creating a new security group.

[EC2](#) > [Security Groups](#) > sg-09d51590eb1851b46

sg-09d51590eb1851b46 - launch-wizard-3

[Actions ▾](#)

Details	
Security group name launch-wizard-3	Security group ID sg-09d51590eb1851b46
Owner 217253764927	Description launch-wizard-3 created 2024-09-29T06:49:51.498Z
	VPC ID vpc-0d4c0d8f48c2e4508
Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

[Inbound rules](#) | [Outbound rules](#) | [Tags](#)

Inbound rules (1)

Edit inbound rules									
C Manage tags Edit inbound rules < 1 > ⌂									
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
<input type="checkbox"/>	-	sgr-Dec19557ab93305...	IPv4	SSH	TCP	22	0.0.0.0/0	-	Delete

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <small>Info</small>									
Security group rule ID	Type	Info	Protocol	Info	Port range	Info	Source	Info	Description - optional
sgr-Dec19557ab9330565	SSH		TCP		22		Custom		<input type="text"/> Delete
-	HTTP		TCP		80		Anywhere-Int...		<input type="text"/> Delete
-	All ICMP - IPv6		IPv6 ICMP		All		Anywhere-Int...		<input type="text"/> Delete
-	HTTPS		TCP		443		Anywhere-Int...		<input type="text"/> Delete
-	All traffic		All		All		Anywhere-Int...		<input type="text"/> Delete
-	Custom TCP		TCP		5666		Anywhere-Int...		<input type="text"/> Delete
-	All ICMP - IPv4		ICMP		All		Anywhere-Int...		<input type="text"/> Delete

[Add rule](#)

Security group name launch-wizard-3		Security group ID sg-09d51590eb1851b46	Description launch-wizard-3 created 2024-09-29T06:49:51.498Z	VPC ID vpc-0d4c0d8f48c2e4508
Owner 217253764927	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry		

[Inbound rules](#) | [Outbound rules](#) | [Tags](#)

Inbound rules (7)

Edit inbound rules									
C Manage tags Edit inbound rules < 1 > ⌂									
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
<input type="checkbox"/>	-	sgr-034c50eff5e5fa00	IPv4	All ICMP - IPv6	IPv6 ICMP	All	0.0.0.0/0	-	Delete
<input type="checkbox"/>	-	sgr-038d0d3791dfcc60e	IPv4	HTTPS	TCP	443	0.0.0.0/0	-	Delete
<input type="checkbox"/>	-	sgr-0e8ad1dd008b14...	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-	Delete
<input type="checkbox"/>	-	sgr-Dec19557ab93305...	IPv4	SSH	TCP	22	0.0.0.0/0	-	Delete
<input type="checkbox"/>	-	sgr-00a0e56d560959f45	IPv4	HTTP	TCP	80	0.0.0.0/0	-	Delete
<input type="checkbox"/>	-	sgr-064c062d69916f84	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-	Delete
<input type="checkbox"/>	-	sgr-0613b7b6aa9d30def	IPv4	All traffic	All	All	0.0.0.0/0	-	Delete

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

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

Connect to instance Info

Connect to your instance i-0011127bbfdb2f467 (nagios-host) using any of these options

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID
 i-0011127bbfdb2f467 (nagios-host)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp_09.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "exp_09.pem"
4. Connect to your instance using its Public DNS:
 ec2-44-204-11-28.compute-1.amazonaws.com

Example:
 ssh -i "exp_09.pem" ec2-user@ec2-44-204-11-28.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.

Or open command prompt and paste ssh command.

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ayush Maurya>ssh -i "Downloads/exp_09.pem" ec2-user@ec2-44-204-11-28.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-11-28.compute-1.amazonaws.com (44.204.11.28)' can't be established.
ED25519 key fingerprint is SHA256:v20KH/ezl9iu7/RT6m8LWkgWzEJnnQIqrG9gKwzwC14.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-11-28.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      _ _###_
      \_\####\   Amazon Linux 2023
      ~~ \###|
      ~~ \#/   https://aws.amazon.com/linux/amazon-linux-2023
      ~~ V~' '-->
      ~~ .-. / 
      ~~ /_/
      _/m/' 

Last login: Sun Sep 29 07:11:40 2024 from 18.206.107.27
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo yum update

```
[ec2-user@ip-172-31-91-91 ~]$ 
sudo yum update
Last metadata expiration check: 0:19:03 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:29 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
 Package           Architecture   Version        Repository      Size
=====
Installing:
httpd             x86_64        2.4.62-1.amzn2023
php8_3            x86_64        8.3.10-1.amzn2023.0.1
=====
Installing dependencies:
apr               x86_64        1.7.2-2.amzn2023.0.2
apr-util          x86_64        1.6.3-1.amzn2023.0.1
generic-logos-httd noarch       18.0.0-12.amzn2023.0.3
httpd-core        x86_64        2.4.62-1.amzn2023
httpd-filesystem noarch       2.4.62-1.amzn2023
httpd-tools       x86_64        2.4.62-1.amzn2023
libbrotli         x86_64        1.0.9-4.amzn2023.0.2
libsodium          x86_64        1.0.19-4.amzn2023
libxslt           x86_64        1.1.34-5.amzn2023.0.2
mailcap            noarch       2.1.49-3.amzn2023.0.3
nginx-filesystem noarch       1:1.24.0-1.amzn2023.0.4
php8_3-clii       x86_64        8.3.10-1.amzn2023.0.1
php8_3-common     x86_64        8.3.10-1.amzn2023.0.1
php8_3-process   x86_64        8.3.10-1.amzn2023.0.1
php8_3-xml        x86_64        8.3.10-1.amzn2023.0.1
=====
Installing weak dependencies:
apr-util-openssl x86_64        1.6.3-1.amzn2023.0.1
mod_http2         x86_64        2.0.27-1.amzn2023.0.3
mod_lua           x86_64        2.4.62-1.amzn2023
php8_3-fpm        x86_64        8.3.10-1.amzn2023.0.1
php8_3-mbstring   x86_64        8.3.10-1.amzn2023.0.1
php8_3-opcache    x86_64        8.3.10-1.amzn2023.0.1
php8_3-pdo        x86_64        8.3.10-1.amzn2023.0.1
php8_3-sodium    x86_64        8.3.10-1.amzn2023.0.1
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
  Installing : php8_3-common-8.3.10-1.amzn2023.0.1.x86_64
  Installing : apr-1.7.2-1.amzn2023.0.2.x86_64
  Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64
  Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch
Running scriptlet: httpd-filesystem-2.4.62-1.amzn2023.noarch
=====
22 MB/s | 10 MB  00:00
1/1
1/25
2/25
3/25
4/25
5/25
6/25
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:41 ago on Sun Sep 29 06:56:15 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
 Package           Architecture   Version        Repository      Size
=====
Installing:
gcc               x86_64        11.4.1-2.amzn2023.0.2
=====
Installing dependencies:
annobin-docs      noarch       18.93-1.amzn2023.0.1
annobin-plugin-gcc x86_64        18.93-1.amzn2023.0.1
cpp               x86_64        11.4.1-2.amzn2023.0.2
gc                x86_64        8.0.4-5.amzn2023.0.2
glibc-devel       x86_64        2.34-52.amzn2023.0.11
glibc-headers-x86 noarch       2.34-52.amzn2023.0.11
guile2            x86_64        2.2.7-2.amzn2023.0.3
kernel-headers    x86_64        6.1.109-118.189.amzn2023
libgcc            x86_64        1.2.1-1.amzn2023.0.2
libtool-ltdl     x86_64        2.4.7-1.amzn2023.0.3
libcrypt-devel    x86_64        4.4.33-7.amzn2023
make              x86_64        1:4.3-5.amzn2023.0.2
=====
Transaction Summary
=====
Install 13 Packages
=====
Total download size: 52 M
=====
Installed:
annobin-docs-10.93-1.amzn2023.0.1.noarch
gc-8.0.4-5.amzn2023.0.2.x86_64
glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
libmpc-1.2.1-2.amzn2023.0.2.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
=====
CPP-11.4.1-2.amzn2023.0.2.x86_64
glibc-devel-2.34-52.amzn2023.0.11.x86_64
kernel-headers-6.1.109-118.189.amzn2023.x86_64
libcrypt-devel-4.4.33-7.amzn2023.x86_64
=====
Complete!
```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:21:30 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
 Package           Architecture   Version        Repository      Size
=====
Installing:
gd               x86_64        2.3.3-5.amzn2023.0.3
gd-devel          x86_64        2.3.3-5.amzn2023.0.3
=====
Installing dependencies:
brotli            x86_64        1.0.9-4.amzn2023.0.2
brotli-devel      x86_64        1.0.9-4.amzn2023.0.2
bz2p-devel        x86_64        1.0.8-1.amzn2023.0.2
cairo             x86_64        1.17.6-2.amzn2023.0.1
cmake-filesystem x86_64        3.22.2-1.amzn2023.0.4
fontconfig        x86_64        2.13.94-2.amzn2023.0.2
=====
amazonlinux      139 k
amazonlinux      38 k
amazonlinux      314 k
amazonlinux      31 k
amazonlinux      211 k
amazonlinux      604 k
amazonlinux      16 k
amazonlinux      273 k
```

```
Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libDX11-1.7.2-2.amzn2023.0.4.x86_64
libEGL-1.4.10-3.amzn2023.0.4.x86_64
libExt-1.3.4-6.amzn2023.0.2.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.2.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-18.amzn2023.0.6.x86_64
libsep0-devel-3.4-3.amzn2023.0.3.x86_64
libwebrtc-1.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-10.40.1-2.amzn2023.0.2.x86_64
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
cmake-fs-3.22.2-1.amzn2023.0.4.x86_64
fonts-fs-1.2.0.5-12.amzn2023.0.2.noarch
gd-2.3.3-5.amzn2023.0.3.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-0.9-6.amzn2023.0.4.x86_64
libXext-1.3.4-6.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-devel-2.1.6.37-18.amzn2023.0.6.x86_64
libtiff-4.0-4.amzn2023.0.18.x86_64
libwebrtc-1.2.4-1.amzn2023.0.6.x86_64
libxml2-devel-2.18.4-1.amzn2023.0.6.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-devel-2.3.3-5.amzn2023.0.3.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-0.9-6.amzn2023.0.4.x86_64
libXpm-devel-3.0.10-1.amzn2023.0.4.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-devel-4.0-4.amzn2023.0.18.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch

Complete!
```

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

```
sudo adduser -m nagios
sudo passwd nagios
(password : ayushmau)
```

```
Complete!
[ec2-user@ip-172-31-91-91 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-91-91 ~]$ |
```

6. Create a new user group
sudo groupadd nagcmd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-91-91 ~]$ |
```

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

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

```
[ec2-user@ip-172-31-91-91 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-91-91 ~]$ |
```

8. Create a new directory for Nagios downloads

```
mkdir
~/downloads cd
```

```
[ec2-user@ip-172-31-91-91 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-91-91 ~]$ |
~/downloads
```

9. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ..
[ec2-user@ip-172-31-91-91 ~]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-29 09:11:59-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz 100%[=====] 1.97M 5.07MB/s in 0.4s
2024-09-29 09:11:59 (5.07 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-91-91 downloads]$ |
```

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

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ cd ..
[ec2-user@ip-172-31-91-91 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-29 09:14:28-- 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-2.4 100%[=====] 2.62M 6.92MB/s in 0.4s
```

10. Use tar to unzip and change to that directory. tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
nagios-4.0.8/THANKS
nagios-4.0.8/UPGRADING
nagios-4.0.8/base/
nagios-4.0.8/base/.gitignore
```

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

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

Here we go an error

```
[ec2-user@ip-172-31-91-91 downloads]$ ./configure --with-command-group=nagcmd
-bash: ./configure: No such file or directory
[ec2-user@ip-172-31-91-91 downloads]$ |
```

Solution

Navigate to nagios folder in downloads

```
[ec2-user@ip-172-31-91-91 downloads]$ ls
nagios-4.0.8 nagios-4.0.8.tar.gz nagios-plugins-2.0.3.tar.gz
[ec2-user@ip-172-31-91-91 downloads]$ cd nagios-4.0.8
[ec2-user@ip-172-31-91-91 nagios-4.0.8]$ |
```

Error 2: Cannot find SSL headers.

Solution: Install openssl dev library

Steps:

sudo yum install openssl-devel

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 2:24:05 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
 Package           Arch      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:
```

Now run

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

```
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: /run/nagios.lock
Check result directory: /usr/local/nagios/var/spool/checkresults
Init directory: /lib/systemd/system
Apache conf.d directory: /etc/httpd/conf.d
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/bin/traceroute

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ |
```

12. Compile the source code.

make all

```
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
```

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

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ make all

sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -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../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -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-overflo
w=]
  253 |           log_debug_info(DEBUGL_CHECKS, 1, "Found specialized
worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
          |
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -T.. -T../lib -T../include -T../include -T.. -g -O2 -DHAVE
```

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

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

```

#
# CONTACTS
#
#####
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

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

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

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {
    contactgroup_name   admins
    alias              Nagios Administrators
    members            nagiosadmin
}

```

And change email with your email

15. Configure the web interface.

sudo make install-webconf

```

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$

```

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

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

```

[ec2-user@ip-172-31-91-91 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-172-31-91-91 nagios-4.5.5]$

```

Password: Ayushmau

17. Restart Apache

sudo service httpd restart

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ |
```

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

cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/argnonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config-test/
```

19. Compile and install plugins

cd nagios-plugins-2.4.11

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

```
[ec2-user@ip-172-31-91-91 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...
checking whether we are cross compiling... no
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 gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
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 Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
```

make

sudo make install

```
[ec2-user@ip-172-31-91-91 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 && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^.\.ifndef/, $p' \
< ../build-aux/snippet/warn-on-use.h \
> warn-on-use.h-t && \
mv warn-on-use.h-t warn-on-use.h
rm -f arg-nonnull.h-t arg-nonnull.h && \
sed -n -e '/GL_ARG_NONNULL/, $p' \
< ../build-aux/snippet/arg-nonnull.h \
> arg-nonnull.h-t && \
mv arg-nonnull.h-t arg-nonnull.h
/usr/bin/mkdir -p arpa
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$
```

20. Start Nagios

Add Nagios to the list of system services

sudo chkconfig --add nagios

sudo chkconfig nagios on

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo chkconfig --add nagio
s
sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Synchronizing state of nagios.service with SysV service script with /usr/lib
/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service →
/usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ |
```

Verify the sample configuration files

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

Error

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

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

Solution:

Create the missing directory: If the directory is missing, create it with the necessary permissions:

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

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
sudo chmod 775 /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$
```

Now run again

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

```
[ec2-user@ip-172-31-91-91 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
```

sudo service nagios start

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo service nagios start
Starting nagios (via systemctl): [ OK ]
```

21. Check the status of Nagios

sudo systemctl status nagios

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
  Active: active (running) since Sun 2024-09-29 08:04:30 UTC; 37s ago
    Docs: man:systemd-sysv-generator(8)
 Process: 68037 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
   Tasks: 6 (limit: 1112)
  Memory: 2.0M
     CPU: 47ms
    CGroup: /system.slice/nagios.service
            └─68059 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─68061 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─68062 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─68063 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─68064 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─68065 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68063;pid=68063
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68062;pid=68062
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68064;pid=68064
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68061;pid=68061
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Warning: Could not open object cache file '/usr/local/nagios/var/objec>
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmxp2N>
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Successfully launched command file worker with pid 68065
Sep 29 08:04:39 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpTng>
Sep 29 08:04:49 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpAfy>
Sep 29 08:04:59 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpCtq>
lines 1-26/26 (END)
```

Error:

The log messages suggest that Nagios is unable to create temporary files, particularly in the directory `/usr/local/nagios/var/`. This is typically caused by permission issues, or the directory might not exist.

Solution:

Firstly check whether `/usr/local/nagios/var/` is there or not. If yes.....

ls -ld /usr/local/nagios/var/

Change ownership: Set the correct ownership for the Nagios user and group:

sudo chown -R nagios:nagcmd /usr/local/nagios/var

Set permissions: Ensure the directory has the right permissions:

sudo chmod -R 775 /usr/local/nagios/var

Restart Nagios: After adjusting the ownership and permissions, restart the Nagios service:

sudo systemctl restart nagios

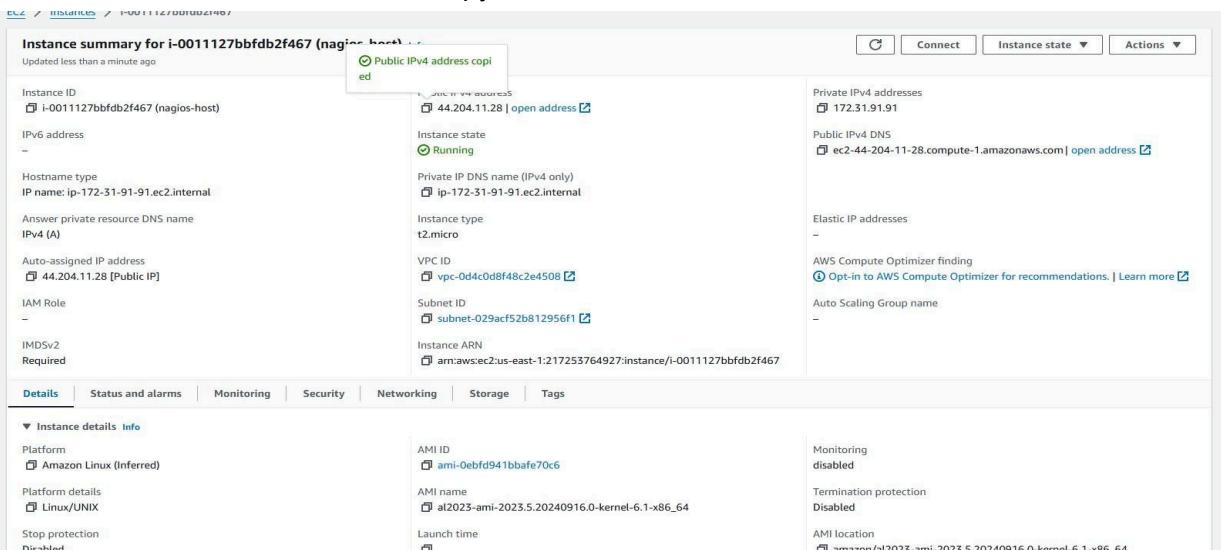
```
drwxr-xr-x. 4 root root 112 Sep 29 08:04 /usr/local/nagios/var/
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagcmd /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chmod -R 775 /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ |
```

Now run again

```
[ec2-user@ip-172-31-91-91 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 Sun 2024-09-29 08:51:47 UTC; 42min ago
     Docs: https://www.nagios.org/documentation
   Tasks: 6 (limit: 1112)
   Memory: 2.9M
      CPU: 562ms
     CGroup: /system.slice/nagios.service
             ├─71188 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─71190 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71191 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71192 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71193 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─71194 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: Registry request: name=Core Worker 71191;pid=71191
Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: Registry request: name=Core Worker 71190;pid=71190
Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: Successfully launched command file worker with pid 71194
Sep 29 08:59:22 ip-172-31-91-91.ec2.internal nagios[71188]: SERVICE ALERT: localhost;HTTP;WARNING;HARD;4;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes i
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CR>
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: NOTIFY job 10 from worker Core Worker 71192 is a non-check helper but exited with return>
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Lines 1-25/25 (END)
```

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



23. Open up your browser and look for

http://<your_public_ip_address>/nagios Enter username as nagiosadmin and

password which you set in Step 16.

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

The screenshot shows the Nagios Core web interface. On the left, there's a vertical sidebar with navigation links: General, Home, Documentation, Current Status (with sub-links like Tactical Overview, Map, Hosts, Services, Host Groups, Summary), Service Groups (Summary, Grid), Problems (Services, Unhandled, Events Unhandled, Network Outages), Quick Search, Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Overview). The main content area has a header "Nagios® Core™ Version 4.5.5" dated September 17, 2024, with a link to "Check for updates". It features several sections: "Get Started" with a bulleted list of steps; "Latest News" and "Don't Miss..." which are currently empty; and a "Quick Links" section with links to Nagios Library, Labs, Exchange, Support, and the official website. A footer at the bottom contains copyright information and a license notice.

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

Conclusion:

In this practical, we successfully installed and configured Nagios Core along with Nagios plugins and NRPE on an Amazon EC2 instance. We created a Nagios user, set up necessary permissions, and resolved common installation errors. Finally, we verified the setup by accessing the Nagios web interface, confirming that our monitoring system was fully operational.