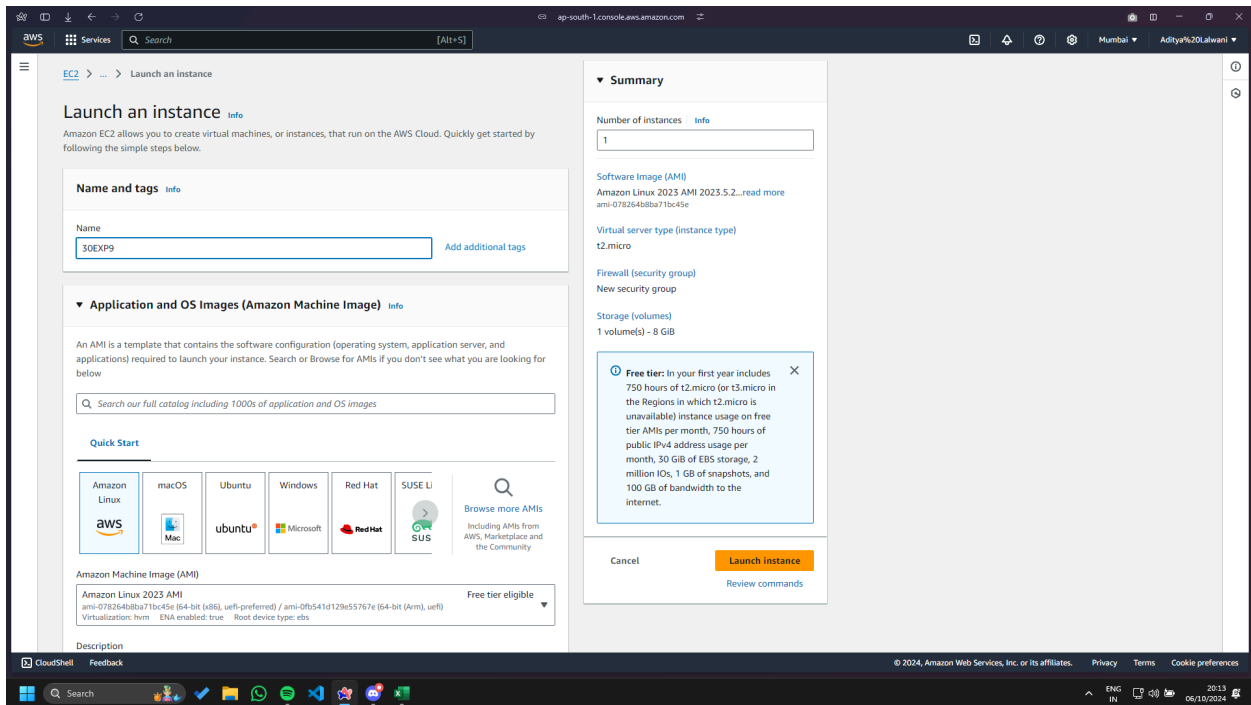
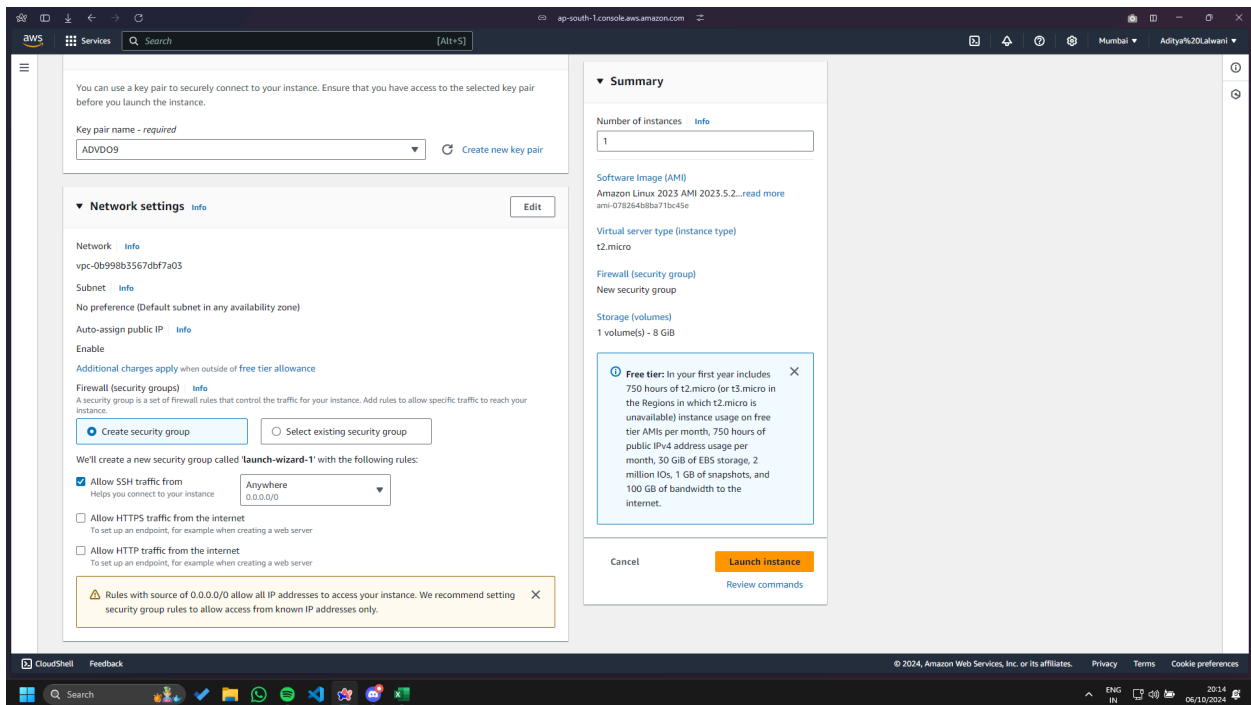


Login to your AWS account. Search for EC2 on services. Open the interface and click on Create Instance.

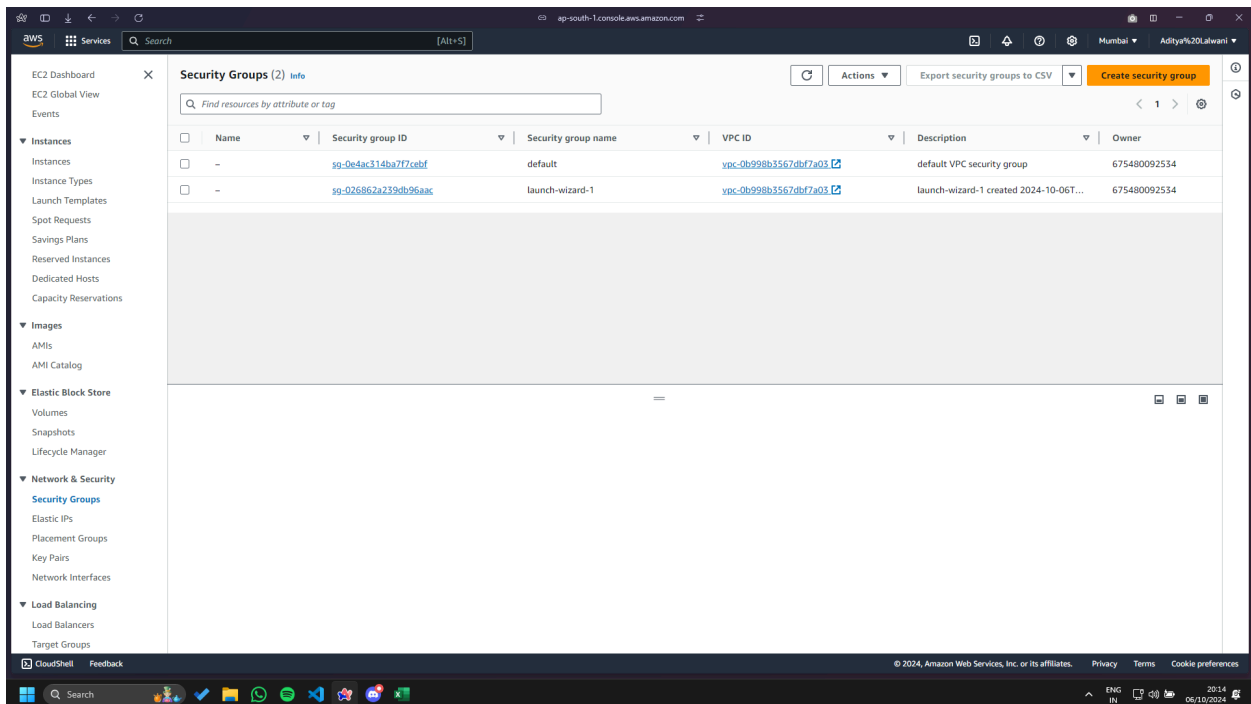


Select The OS Image as Amazon Linux.

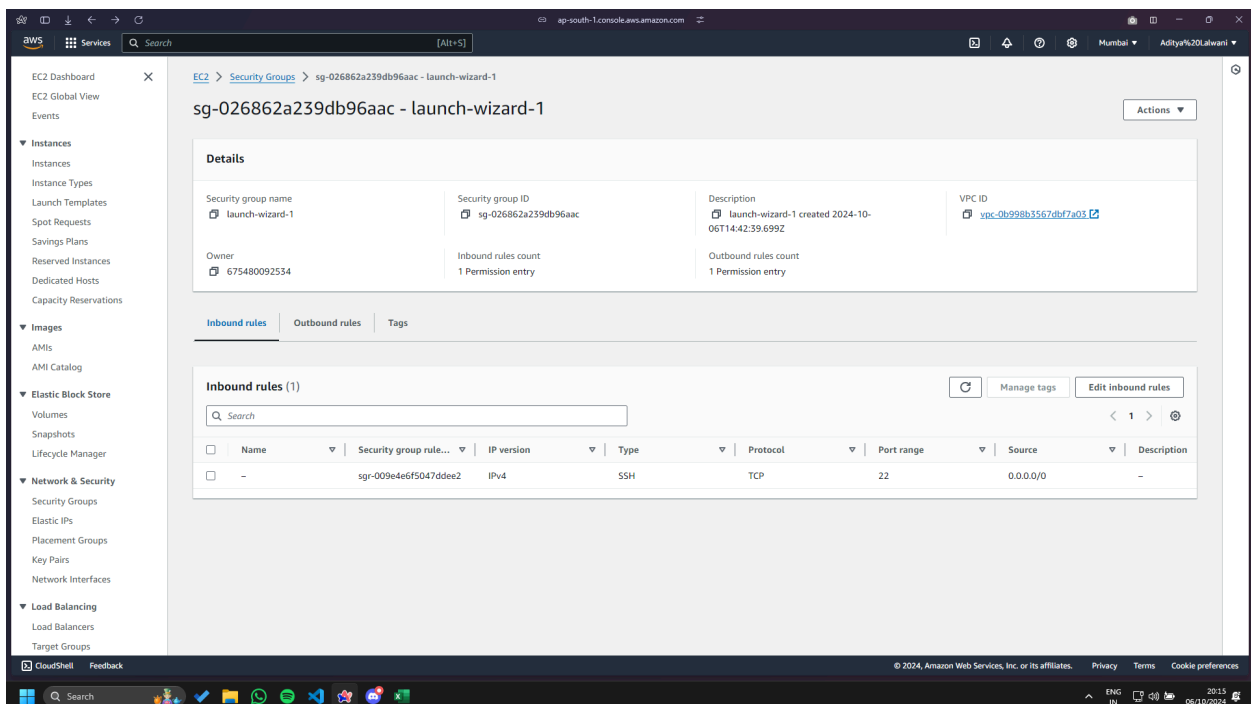
If you do not have a private key created or a .pem file created, click on create a key pair. Else select the key pair that you had created before. (Make sure you know where the .pem file for that key is present on your system)



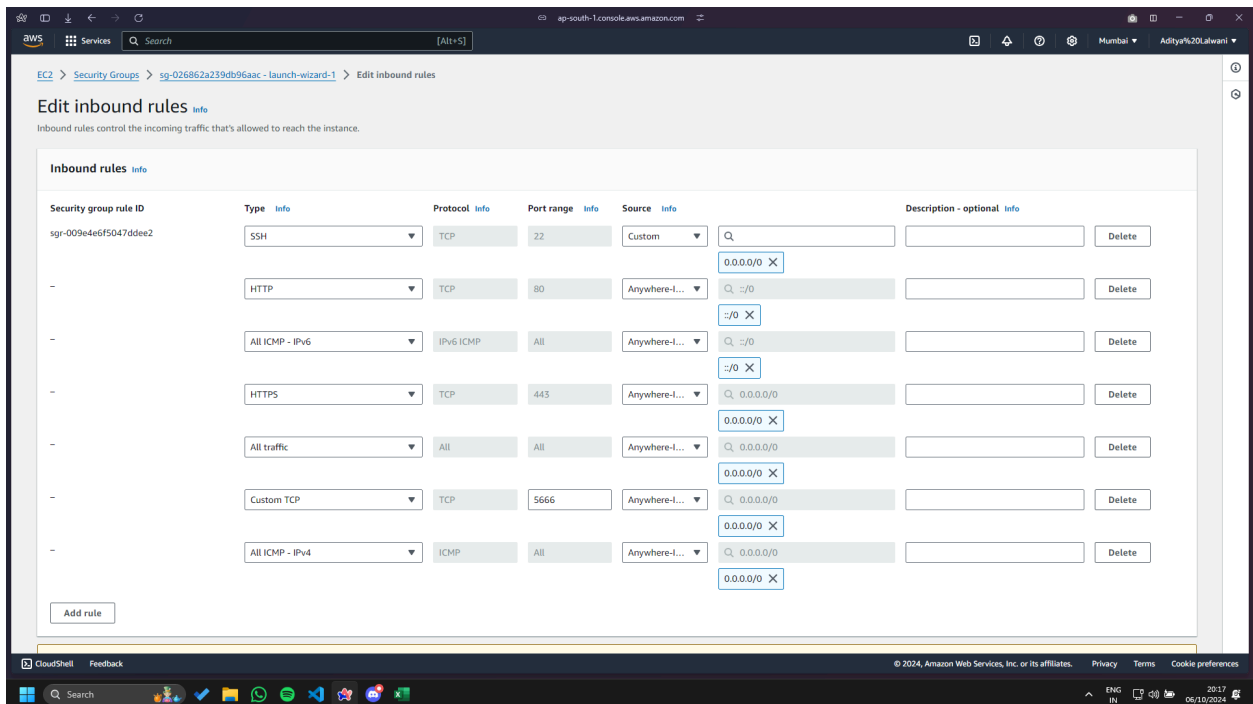
AWS will create a security group for this instance. Keep the name of that instance saved. After creating the instance, click on Security Groups from the left side pane. Find the security group that was created for your instance. Click on the instance ID for that group.



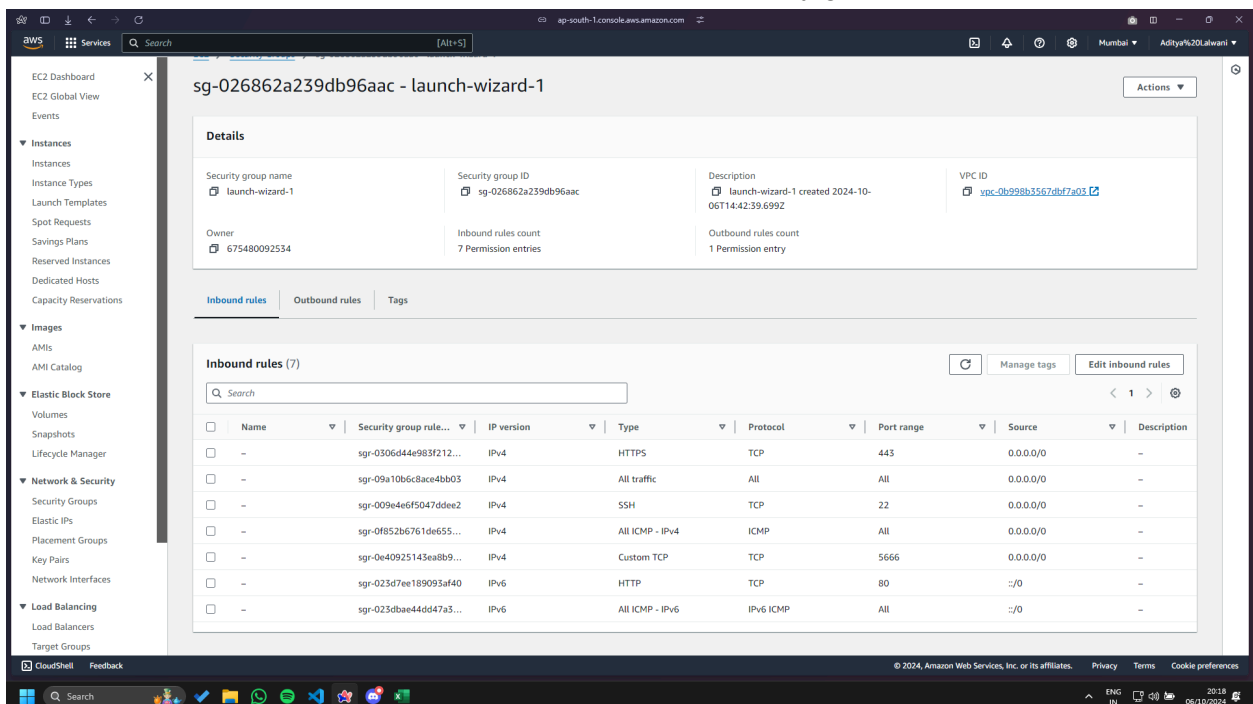
Here, click on Edit Inbound Rules.



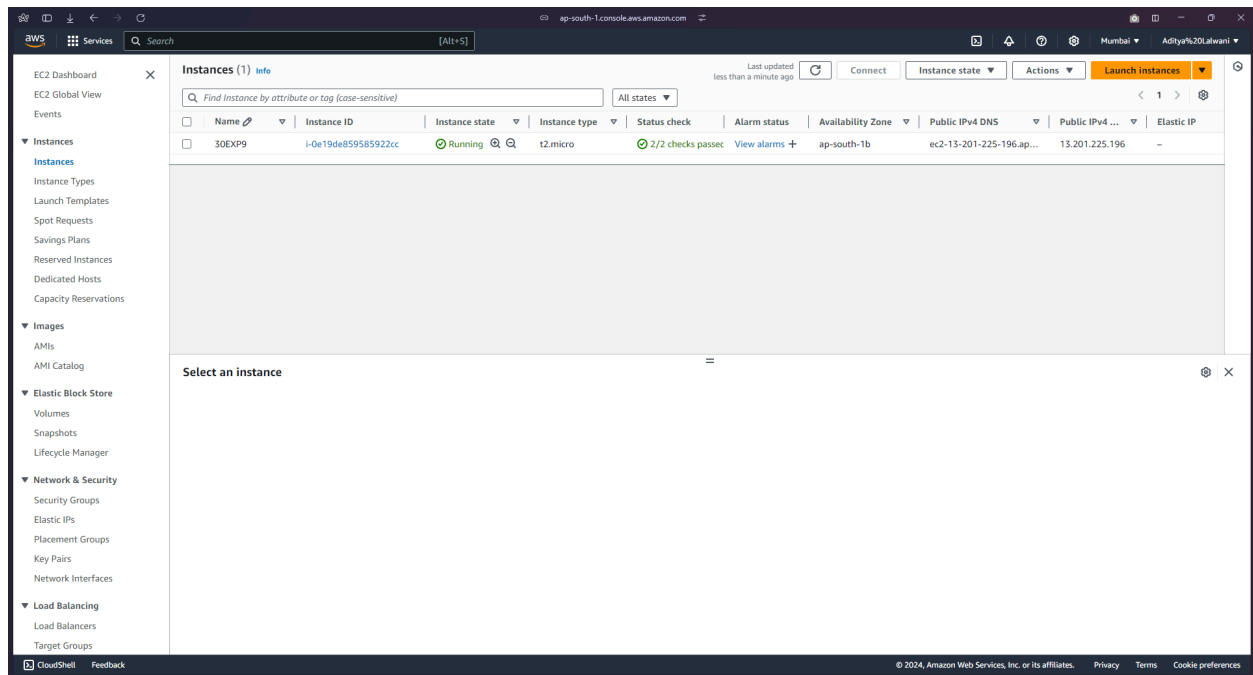
Now, click on add rules, and add the rules for the following protocols:
HTTP, All ICMP - IPv6, HTTPS, All traffic, Custom TCP (Port 5666), All ICMP - IPv4



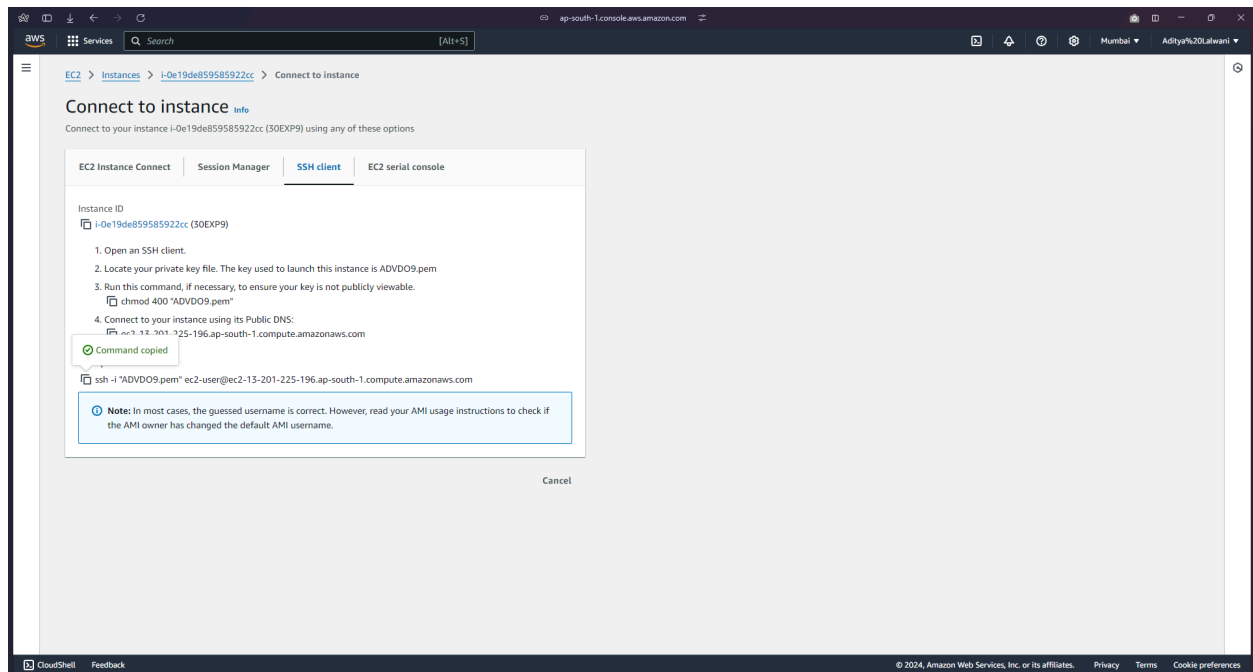
Click on save. This will add all the inbound rules to the security group.



Now come back to the instances screen. Click on the instance ID of your instance. Then click on Connect.



Click on SSH client. Copy the example command.



Step 5: Now, we have to connect our local OS terminal to the instance using SSH. For this,

Open terminal where the private key file is located (.pem)
Paste the copied SSH command and run it.

```
Warning: Permanently added 'ec2-35-154-231-184.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
#_
~\_ ##### Amazon Linux 2023
~\_ #####
~\_ \###|
~\_ \#/ --- https://aws.amazon.com/linux/amazon-linux-2023
~\_ V# ' ->
~\_ /# '
~\_ /m '
[ec2-user@ip-172-31-11-56 ~]$ |
```

Now we start working on this terminal. First run the command
`sudo yum update`
This command will check for any updates for the YUM library.

```
[ec2-user@ip-172-31-11-56 ~]$ sudo yum update
Last metadata expiration check: 0:00:55 ago on Sun Oct 6 15:30:26 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

We are going to install an Apache server and a PHP on this instance. For that, run this
command.

`sudo yum install httpd php`

```
[ec2-user@ip-172-31-11-56 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:09 ago on Sun Oct 6 15:30:26 2024.
Dependencies resolved.
```

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

```
Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
libxslt-1.1.34-5.amzn2023.0.2.x86_64
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64
php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64
php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64
php8.3-process-8.3.10-1.amzn2023.0.1.x86_64
php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64
apr-util-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-core-2.4.62-1.amzn2023.x86_64
httpd-tools-2.4.62-1.amzn2023.x86_64
libsodium-1.0.19-4.amzn2023.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_lua-2.4.62-1.amzn2023.x86_64
php8.3-8.3.10-1.amzn2023.0.1.x86_64
php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64
php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-11-56 ~]$ |
```

Next we install C/C++ compiler (GCC) along with the necessary C libraries required for compiling and running C programs. Use the following command.

`sudo yum install gcc glibc glibc-common`

```
[ec2-user@ip-172-31-11-56 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:01:48 ago on Sun Oct 6 15:30:26 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package                                Architecture      Version                                Repository        Size
=====
Installing:
gcc                                     x86_64            11.4.1-2.amzn2023.0.2                amazonlinux        32 M
Installing dependencies:
annobin-docs                          noarch            10.93-1.amzn2023.0.1                amazonlinux        92 k
annobin-plugin-gcc                    x86_64            10.93-1.amzn2023.0.1                amazonlinux        887 k
cpp                                    x86_64            11.4.1-2.amzn2023.0.2                amazonlinux        10 M
gc                                      x86_64            8.0.4-5.amzn2023.0.2                amazonlinux        105 k
glibc-devel                            x86_64            2.34-52.amzn2023.0.11                amazonlinux        27 k

Installed:
annobin-docs-10.93-1.amzn2023.0.1.noarch      annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
cpp-11.4.1-2.amzn2023.0.2.x86_64              gc-8.0.4-5.amzn2023.0.2.x86_64
gcc-11.4.1-2.amzn2023.0.2.x86_64              glibc-devel-2.34-52.amzn2023.0.11.x86_64
glibc-headers-x86-2.34-52.amzn2023.0.11.noarch guile22-2.2.7-2.amzn2023.0.3.x86_64
kernel-headers-6.1.109-118.189.amzn2023.x86_64 libmpc-1.2.1-2.amzn2023.0.2.x86_64
libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64      libxcrypt-devel-4.4.33-7.amzn2023.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-11-56 ~]$ |
```

We would also need GD library and its development tools. For that, run this command

`sudo yum install gd gd-devel`

```
[ec2-user@ip-172-31-11-56 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:02:29 ago on Sun Oct 6 15:30:26 2024.
Dependencies resolved.
=====
Package                                Architecture      Version                                Repository        Size
=====
Installing:
gd                                     x86_64            2.3.3-5.amzn2023.0.3                amazonlinux        139 k
gd-devel                              x86_64            2.3.3-5.amzn2023.0.3                amazonlinux        38 k
Installing dependencies:
brotli                                x86_64            1.0.9-4.amzn2023.0.2                amazonlinux        314 k
brotli-devel                          x86_64            1.0.9-4.amzn2023.0.2                amazonlinux        31 k
bzip2-devel                            x86_64            1.0.8-6.amzn2023.0.2                amazonlinux        214 k

libpng-2:1.6.37-10.amzn2023.0.6.x86_64      libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64  libsepol-devel-3.4-3.amzn2023.0.3.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64        libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64         libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64         libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64  pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64     pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64         sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch      xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64        zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
[ec2-user@ip-172-31-11-56 ~]$ |
```

Now, we create a user called as 'nagios' and make sure that it has a home directory, and set up a password for it.

sudo adduser -m nagios

sudo passwd nagios

```
[ec2-user@ip-172-31-11-56 ~]$ 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:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-11-56 ~]$ |
```

Create a user group called as 'nagcmd' to execute nagios commands.

sudo groupadd nagcmd

```
[ec2-user@ip-172-31-11-56 ~]$ sudo groupadd nagcmd
```

Add users apache and nagios to this user group.

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-11-56 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

We create a directory downloads, to store the files of nagios server that are downloaded.

mkdir ~/downloads

cd ~/downloads

```
[ec2-user@ip-172-31-11-56 ~]$ mkdir ~/downloads
cd ~/downloads
```

Now we need to install the latest versions of nagios-core and nagios-plugins. Go to the respective websites and check whether a better version is available. If not run these commands. If newer versions are available, then right click on the download button → Copy link address. Paste this link address in place of the current link in command.

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

```
[ec2-user@ip-172-31-11-56 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-06 15:34:42-- 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  913KB/s  in 2.2s
2024-10-06 15:34:46 (913 KB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
```

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

```
[ec2-user@ip-172-31-11-56 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-06 15:35:05-- 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 893KB/s in 3.0s

2024-10-06 15:35:09 (893 KB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-11-56 downloads]$ |
```

Now, we need to extract nagios-core file into the same directory. For that, we will use tar command.

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-11-56 downloads]$ tar zxvf 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/worker/ping/.gitignore
nagios-4.5.5/worker/ping/Makefile.in
nagios-4.5.5/worker/ping/worker-ping.c
nagios-4.5.5/xdata/
nagios-4.5.5/xdata/.gitignore
nagios-4.5.5/xdata/Makefile.in
nagios-4.5.5/xdata/xcddefault.c
nagios-4.5.5/xdata/xcddefault.h
nagios-4.5.5/xdata/xodtemplate.c
nagios-4.5.5/xdata/xodtemplate.h
nagios-4.5.5/xdata/xpddefault.c
nagios-4.5.5/xdata/xpddefault.h
nagios-4.5.5/xdata/xrddefault.c
nagios-4.5.5/xdata/xrddefault.h
nagios-4.5.5/xdata/xsddefault.c
nagios-4.5.5/xdata/xsddefault.h
[ec2-user@ip-172-31-7-70 downloads]$ |
```

We need to ensure that Nagios uses a specific group (in this case, `nagcmd`) for executing external commands.

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

```
[ec2-user@ip-172-31-11-56 downloads]$ ./configure --with-command-group=nagcmd
-bash: ./configure: No such file or directory
```

Solution: Navigate to the nagios-4.5.5 folder in downloads.

Steps: ls

```
[ec2-user@ip-172-31-11-56 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-11-56 downloads]$ |
```

- `cd nagios-4.5.5`
- `./configure --with-command-group=nagcmd`

```
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-11-56 nagios-4.5.5]$ |
```

Steps:

```
sudo yum install openssl-devel
```

```
[ec2-user@ip-172-31-11-56 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:08:11 ago on Sun Oct 6 15:30:26 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
=====
```

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

*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

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:         /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-172-31-11-56 nagios-4.5.5]$ |
```

We need to compile all components of this software according to the instruction in the Makefile.

For that, use this command:

make all

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
/usr/bin/install -c -b -m 666 -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.

/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-172-31-11-56 nagios-4.5.5]$ |
```

We need to update the email linked with this server to our email for it to send notifications (if any needed).

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

```
# 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.aditya.lalwani@ves.ac.in ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}


```

We need to install the necessary configuration files for the Nagios web interface. Run the following command.

sudo make install-webconf

```
[ec2-user@ip-172-31-11-56 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-172-31-11-56 nagios-4.5.5]$ |
```

Now we need to setup a user to access this nagios web interface. So we run this command to create a user called 'nagiosadmin'.

Keep this username and password saved as it is needed to login to the web interface.

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

```
[ec2-user@ip-172-31-11-56 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-172-31-11-56 nagios-4.5.5]$ |
```

Restart the apache server to apply all the recent configurations.

`sudo service httpd restart`

```
[ec2-user@ip-172-31-11-56 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

Now we go back to the downloads folder and extract the files of nagios plugin.

`cd ~/downloads`

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

```
nagios-plugins-2.4.11/po/fr.gmo
nagios-plugins-2.4.11/po/de.gmo
nagios-plugins-2.4.11/po/nagios-plugins.pot
nagios-plugins-2.4.11/po/stamp-po
nagios-plugins-2.4.11/po/ChangeLog
nagios-plugins-2.4.11/po/LINGUAS
nagios-plugins-2.4.11/release
[ec2-user@ip-172-31-11-56 downloads]$ |
```

we need to install the configurations for these files.

`cd nagios-plugins-2.4.11`

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

```
[ec2-user@ip-172-31-11-56 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
```

```
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
[ec2-user@ip-172-31-11-56 nagios-plugins-2.4.11]$ |
```

compile all components of this software according to the instruction in the Makefile. For that, use the commands:

make all

sudo make install

```
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering 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-172-31-11-56 nagios-plugins-2.4.11]$ |
```

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

```
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-172-31-11-56 nagios-plugins-2.4.11]$ |
```

sudo service nagios start

```
[ec2-user@ip-172-31-11-56 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-11-56 nagios-plugins-2.4.11]$
```

sudo systemctl status nagios

```
[ec2-user@ip-172-31-11-56 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 2024-10-06 15:48:17 UTC; 2min 7s ago
     Docs: https://www.nagios.org/documentation
  Process: 64622 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0>
  Process: 64623 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU>
 Main PID: 64624 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 6.1M
      CPU: 296ms
  CGroup: /system.slice/nagios.service
          └─64624 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
            └─64625 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─64626 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                └─64627 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  └─64628 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    └─64629 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: qh: core query handler registered
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: qh: echo service query handler registered
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: qh: help for the query handler registered
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: wproc: Successfully registered manager as @w>
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: wproc: Registry request: name=Core Worker 64>
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: wproc: Registry request: name=Core Worker 64>
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: wproc: Registry request: name=Core Worker 64>
Oct 06 15:48:17 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: wproc: Registry request: name=Core Worker 64>
Oct 06 15:48:19 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: Successfully launched command file worker wi>
Oct 06 15:50:09 ip-172-31-11-56.ap-south-1.compute.internal nagios[64624]: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;1>
lines 1-28/28 (END)
```

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.

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, and Target Groups. The main content area displays the 'Instance summary for i-0f78ca365d69fc75d (30ADO9)'. The instance is in the 'Running' state. Key details include: Instance ID: i-0f78ca365d69fc75d (30ADO9), Public IPv4 address: 35.154.231.184, Private IPv4 address: 172.31.11.56, Hostname type: IP name: ip-172-31-11-56.ap-south-1.compute.internal, Instance type: t2.micro, VPC ID: vpc-0b998b3567db7a03, Subnet ID: subnet-05ae14f0ca0ce43cd, IAM Role: -, IMDSv2: Required. Below the summary, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is active, showing instance details like Platform (Amazon Linux (Inferred)), Platform details (Linux/UNIX), Stop protection (Disabled), Instance auto-recovery (Default), AMI ID (ami-078264b8ba71bc45e), AMI name (al2023-ami-2023.5.20241001.1-kernel-6.1-x86_64), Launch time (Sun Oct 06 2024 20:59:54 GMT+0530 (India Standard Time) (23 minutes)), Lifecycle (normal), Key pair assigned at launch, Monitoring (disabled), Termination protection (Disabled), AMI location (amazon/al2023-ami-2023.5.20241001.1-kernel-6.1-x86_64), Stop-hibernate behavior (Disabled), and State transition reason.

Open a new tab. In the address bar type `http://<publicipaddress>/nagios`. This would be in the output

The screenshot shows the Nagios Core web interface. The top navigation bar includes links for General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid, Service Groups, Summary, Grid, Problems (Unhandeled), Hosts (Unhandeled), Network Outages, Quick Search, Reports, Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, System, Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration. The main content area displays the Nagios Core logo, version 4.5.5, and the daemon running with PID 64624. Below this, there are sections for Get Started, Latest News, Don't Miss..., and Quick Links. The 'Get Started' section includes links for Start monitoring your infrastructure, Change the look and feel of Nagios, Extend Nagios with hundreds of add-ons, Get support, Get training, and Get certified. The 'Quick Links' section includes links for Nagios Library (tutorials and docs), Nagios Labs (development blog), Nagios Exchange (plugins and add-ons), Nagios Support (tech support), Nagios.com (company), and Nagios.org (project). The 'Latest News' and 'Don't Miss...' sections are currently empty. The footer contains copyright information for Nagios Core Development Team and Community Contributors, and a link to the Nagios Core license.