Experiment 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.

Why We Need Nagios tool?
Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitor and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Features of Nagios

Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.

- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.

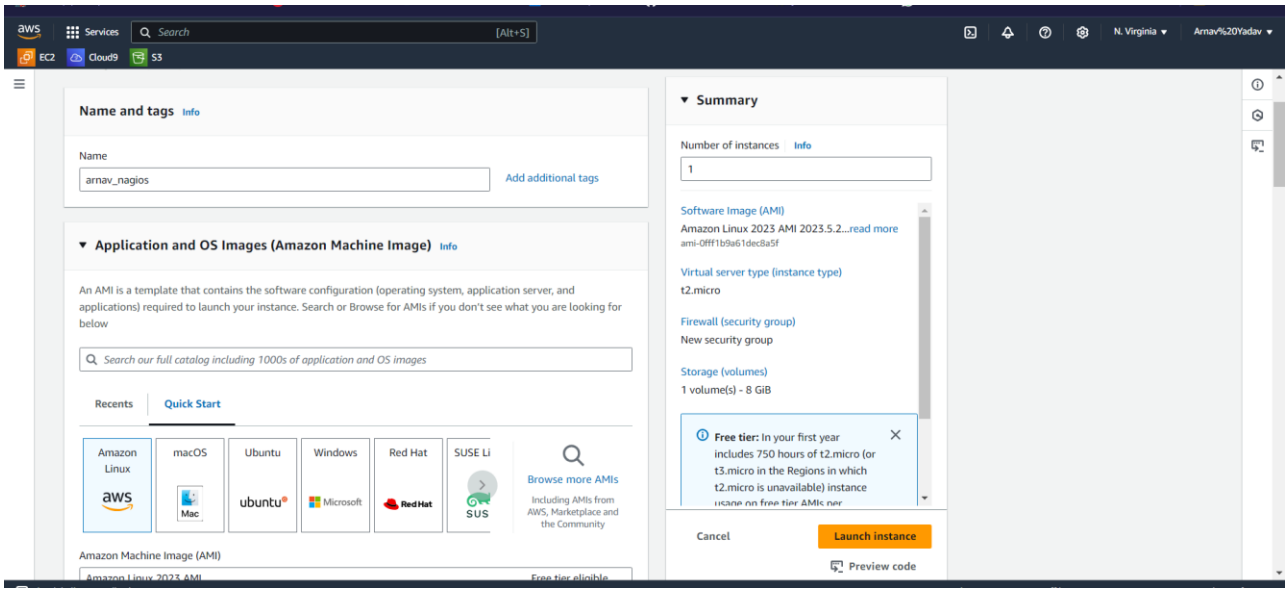
1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Installation of Nagios

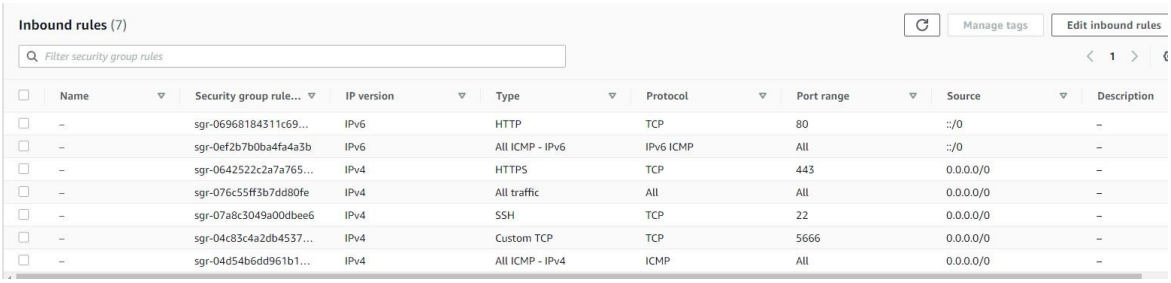
Prerequisites: AWS Free

Tier Steps:

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

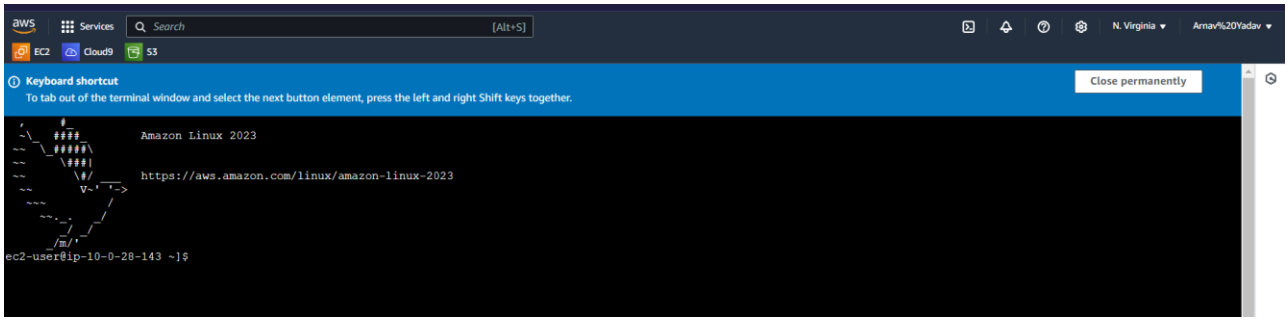


2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.



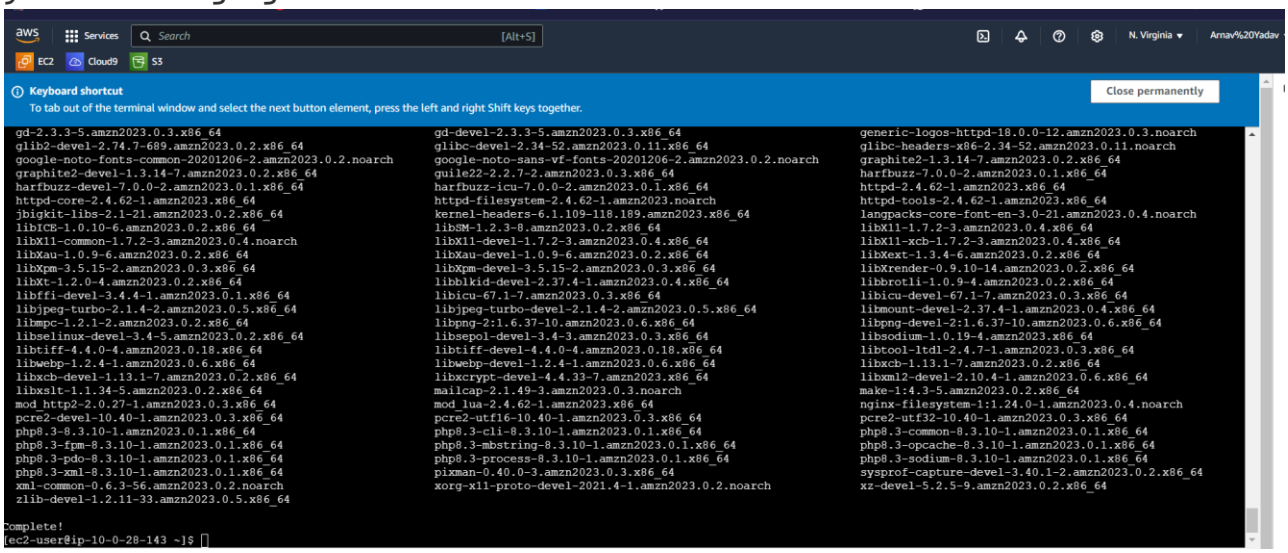
You have to edit the inbound rules of the specified Security Group for this.

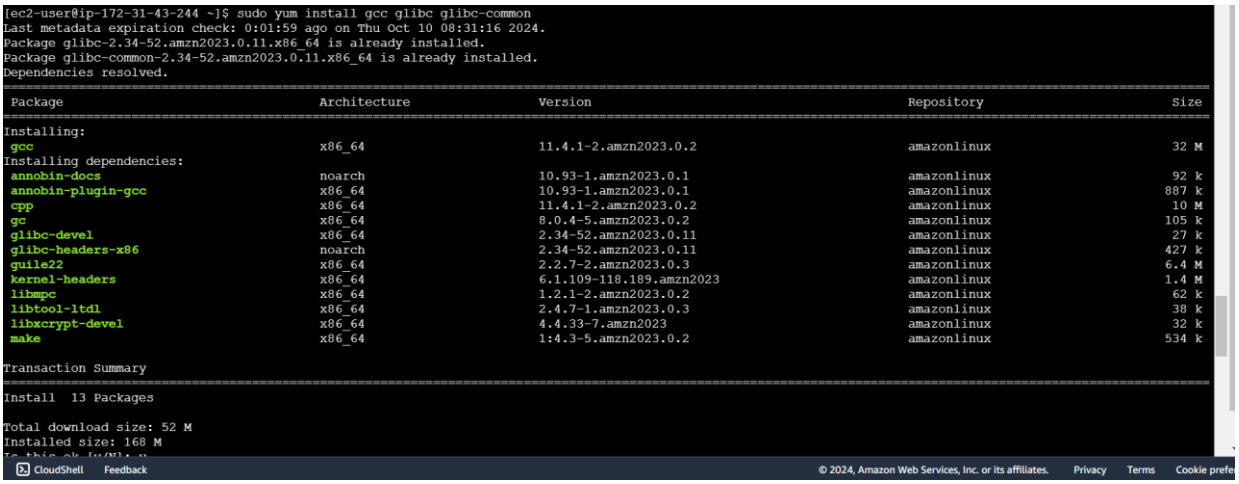
3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.

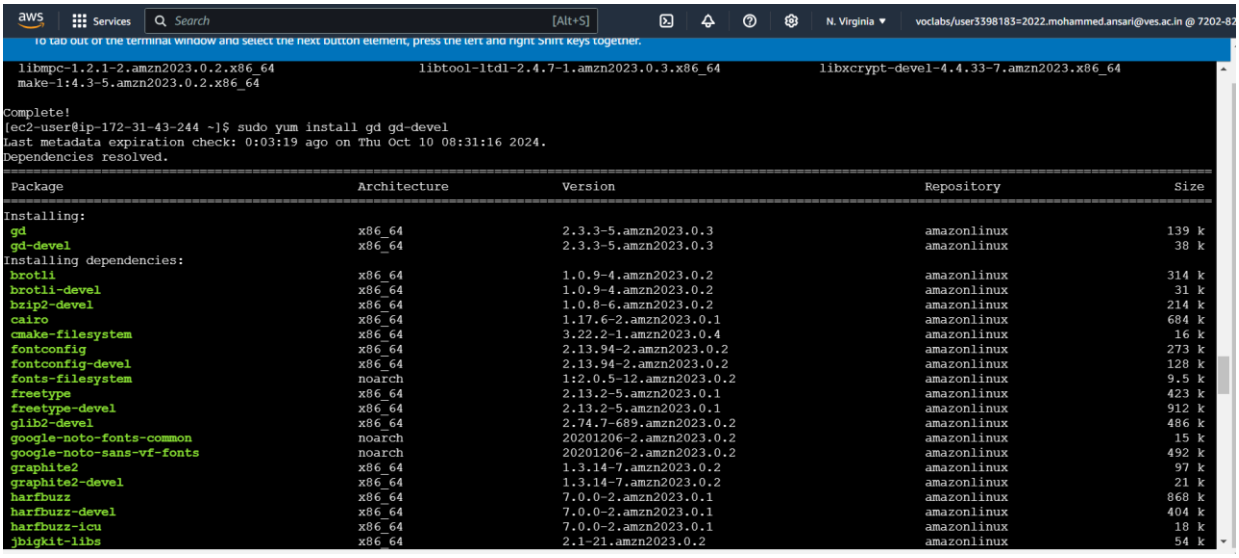


4. Update the package indices and install the following

packages using yum sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-commonsudo
yum install gd gd-devel

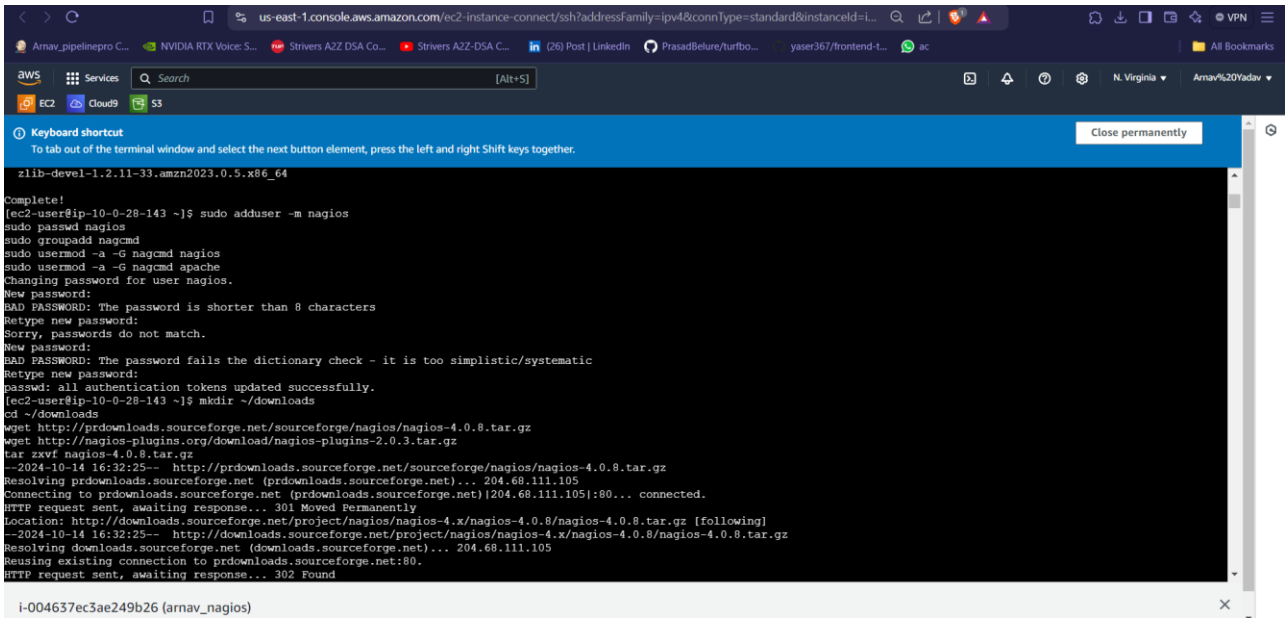






5. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.

```
sudo adduser -m
nagiossudo passwd
nagios
```



6. Create a new user group

```
sudo groupadd nagcmd
```

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
```

8. Create a new directory for Nagios downloads

    mkdir
    ~/downloadscd
    ~/downloads

```
[ec2-user@ip-172-31-43-244 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-43-244 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-43-244 ~]$ mkdir ~/downloads
cd ~/downloads
```

9. Use wget to download the source zip files.

    wget
    http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz

    wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz

10. Use tar to unzip and change to that directory.

    tar zxvf nagios-4.0.8.tar.gz

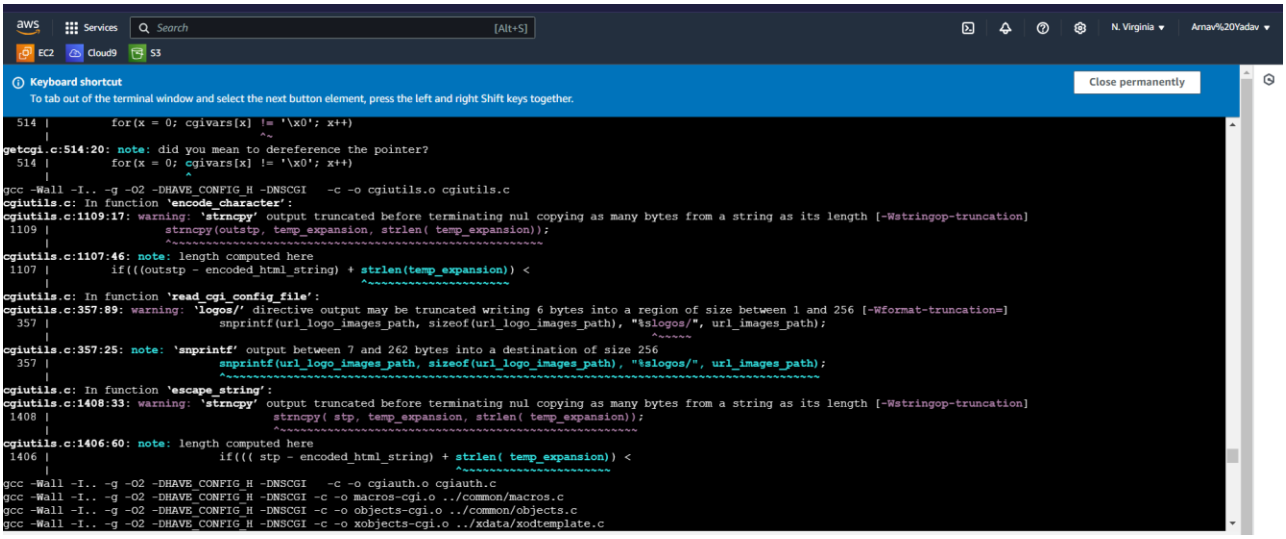
11. Run the configuration script with the same group name you previously created.

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

```
[ec2-user@ip-172-31-43-244 downloads]$ cd nagios-4.0.8
[ec2-user@ip-172-31-43-244 nagios-4.0.8]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
```

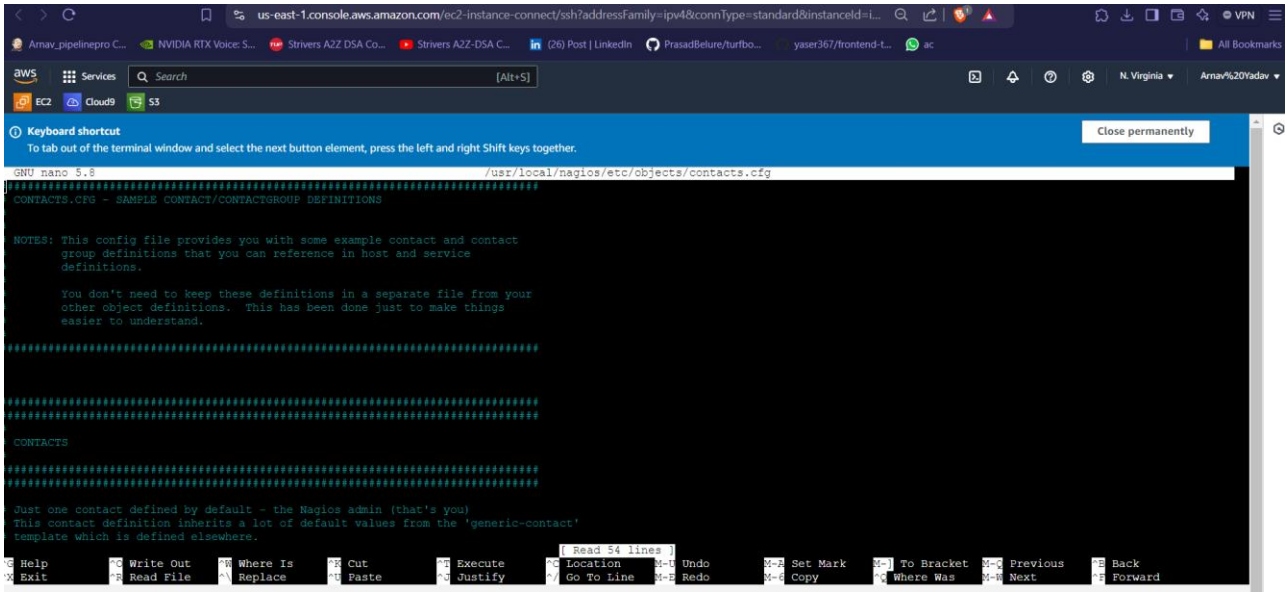
12. Compile the source code.

    make all



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
```



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

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

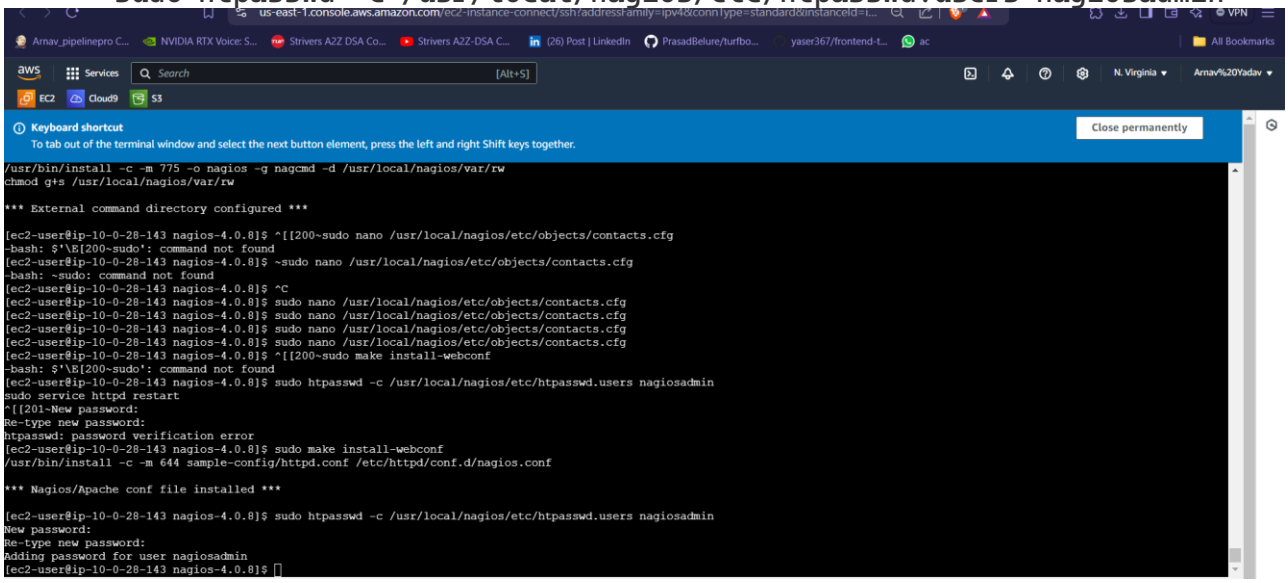
```
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.

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                           arnav20604@gmail.com    ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
       }


^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket   M-Q Previous     ^B Back
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was     M-W Next         ^F Forward
```

15. Configure the web interface.

```
sudo make install-webconf
```

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
```

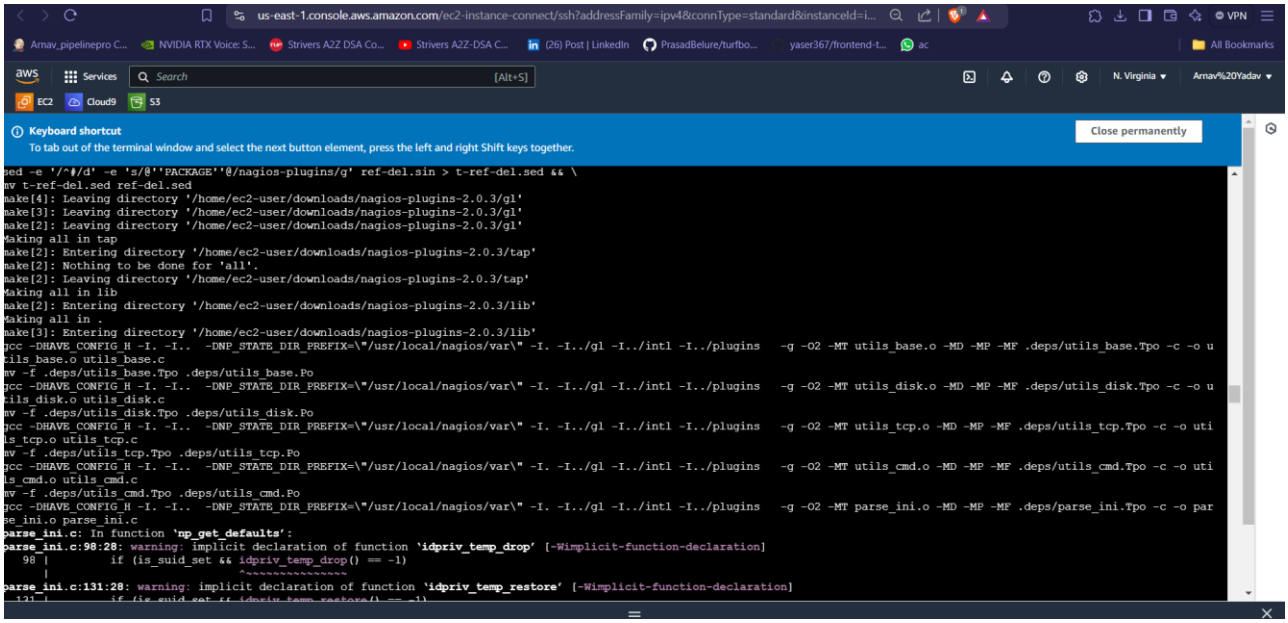


17. Restart Apache

```
sudo service httpd restart
```

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

```
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
```



19. Compile and install plugins

```
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-
group=nagiosmake
sudo make install
```

```
[ec2-user@ip-172-31-43-244 downloads]$ cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
```

20. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
sudo chkconfig nagios on
```

Verify the sample configuration files

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

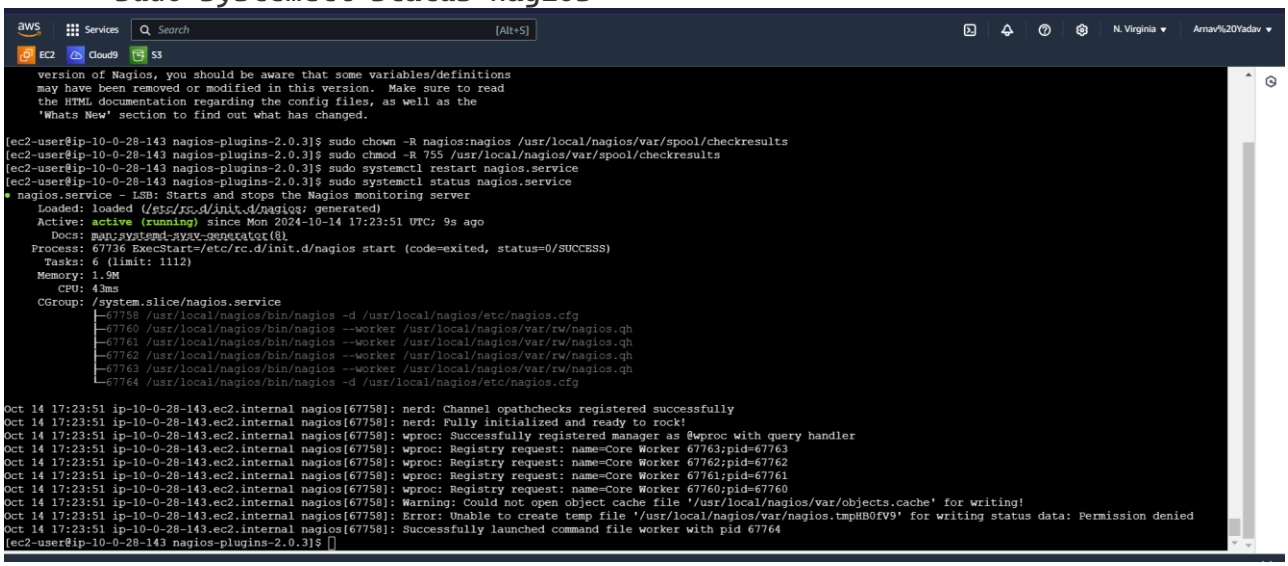
If there are no errors, you can go ahead and start Nagios.

sudo service nagios start

```
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagios /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-43-244 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 1 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
```

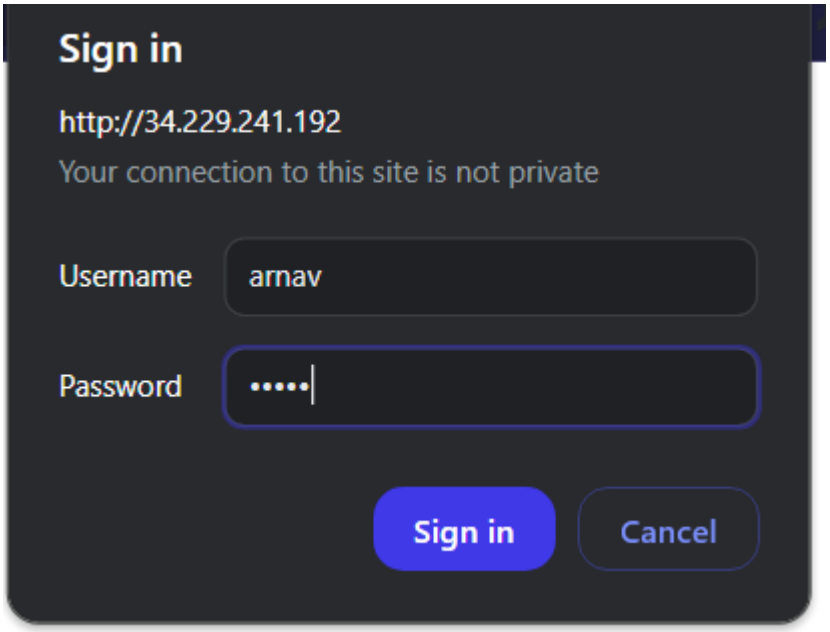
21. Check the status of Nagios

sudo systemctl status nagios



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.

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

Conclusion:

Thus, we learned about Nagios and successfully set it up as a host on our Amazon Linux machine.