EXPERIMENT NO. 09 DATE –

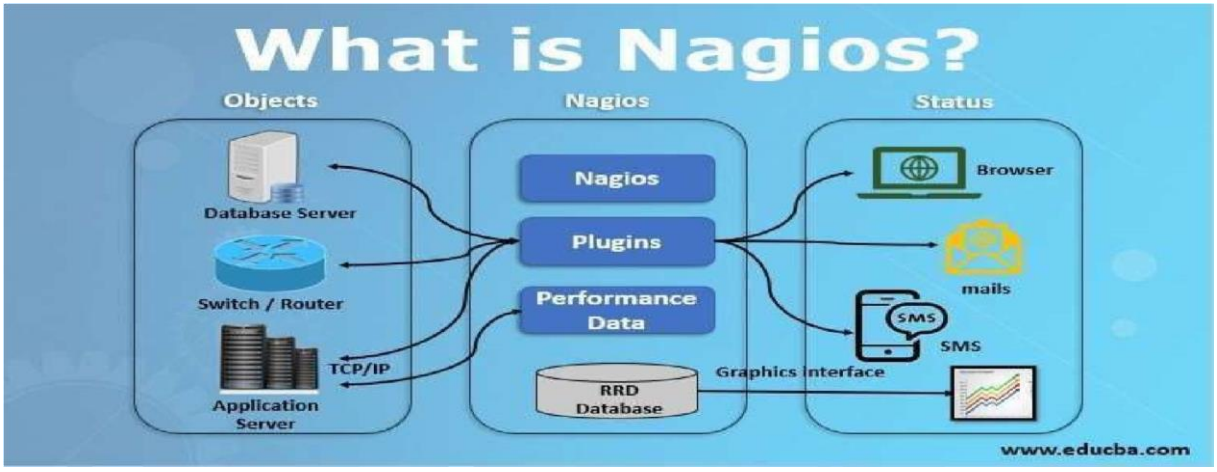
Aim -

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

Theory -

Nagios XI provides a monitoring, alerting, graphing, and reporting platform for your entire infrastructure, including servers, operating systems, applications, network devices, websites, hypervisors, cloud servers, and much more. Thousands of community plugins and the ability to write your own custom plugins enable you to monitor virtually anything.

- The Nagios Core 4 engine provides the powerful, flexible foundation, which Nagios XI makes faster and easier to leverage than ever before.
- Dashboards empower you to create custom visualizations for an at-a-glance view of the data most important to you.
- Alerts can be sent vial email or text, in addition to the ability to send an SNMP trap or execute a custom script when problems are found.
- Reports and Graphs enable you to leverage the performance and state history data Nagios XI collects, and can be exported and shared. Reports such as Capacity Planning empower you to project future usage based on historical performance.



Advantages

Implementing effective server monitoring with Nagios offers the following benefits:

- Increased server, services, process, and application availability
- Fast detection of network and server outages and protocol failures
- Fast detection of failed servers, services, processes and batch jobs

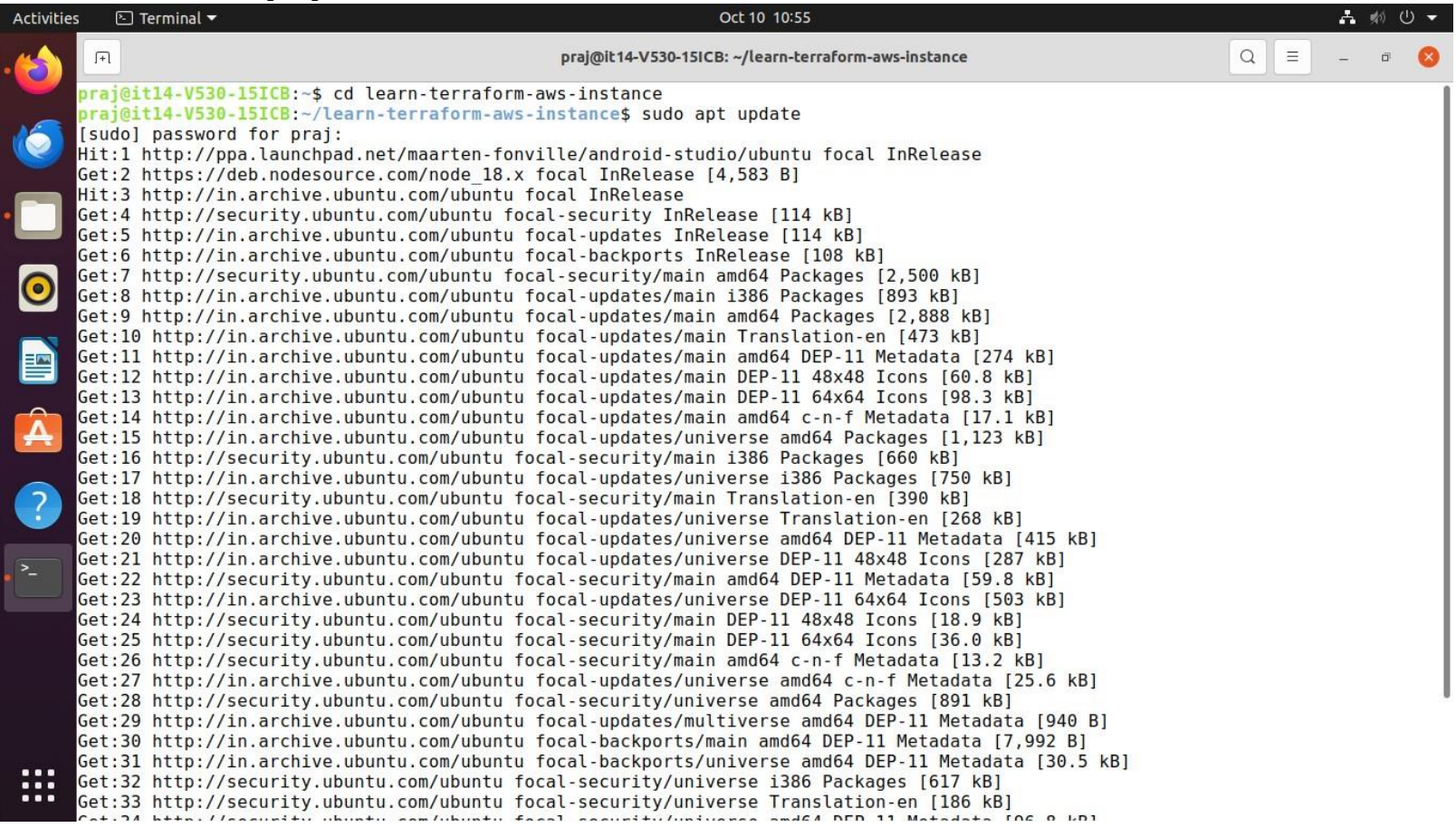
Steps:

Step 1 - Install Packages Dependencies

First, we will update the Ubuntu repository and install some packages dependencies for the Nagios installation.

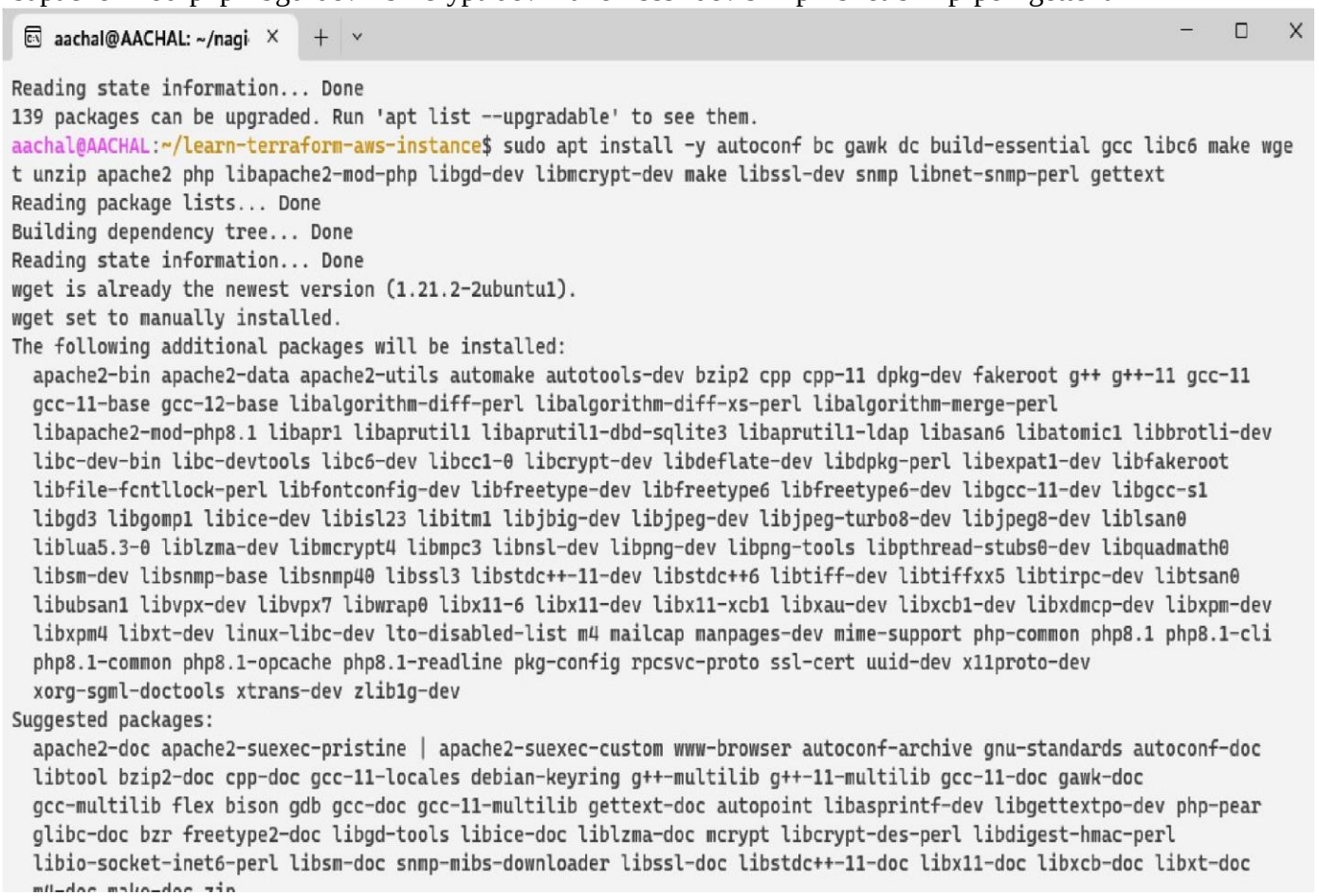
Update the Ubuntu repository using the apt command below.

$ sudo apt update



After this, install packages dependencies for Nagios installation.

$ sudo apt install -y autoconf bc gawk dc build-essential gcc libc6 make wget unzip apache2 php libapache2mod-php libgd-dev libmcrypt-dev make libssl-dev snmp libnet-snmp-perl gettext



Step 2 - Install Nagios Core 4.4.6

In this step, we will install the latest stable version Nagios Core 4.4.6. And we will install it manually from the source.

Go to your home directory and download the Nagios Core source code.

$ cd ~/

$ wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz

```
praj@it14-V530-15ICB:~/learn-terraform-aws-instance$ cd ~/
praj@it14-V530-15ICB:~$ wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
--2023-10-10 10:59:32--  https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/refs/tags/nagios-4.4.6 [following]
--2023-10-10 10:59:34--  https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/refs/tags/nagios-4.4.6
Resolving codeload.github.com (codeload.github.com)... 20.207.73.88
Connecting to codeload.github.com (codeload.github.com)|20.207.73.88|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz                [                                        <=>         ]  10.81M   581KB/s    in 54s

2023-10-10 11:00:29 (204 KB/s) - 'nagios-4.4.6.tar.gz' saved [11333431]

praj@it14-V530-15ICB:~$
```

Extract the Nagios package and go to the extracted Nagios directory.

$ tar -xf nagios-4.4.6.tar.gz
$ cd nagioscore-*/

```
praj@it14-V530-15ICB:~$ tar -xf nagios-4.4.6.tar.gz
praj@it14-V530-15ICB:~$ cd nagioscore-*/
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo ./configure --with-httpd-conf=/etc/apache2/sitesenabled $ sudo make
configure: WARNING: you should use --build, --host, --target
```

- Compile and Install Nagios

First, compile Nagios source code and define the Apache virtual host configuration for Nagios.

$ sudo ./configure --with-httpd-conf=/etc/apache2/sitesenabled $ sudo make all

```
See `config.log' for more details
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo ./configure --with-httpd-conf=/etc/apache2/sitesenabled $ sudo
configure: WARNING: you should use --build, --host, --target
configure: WARNING: invalid host type: $
configure: WARNING: you should use --build, --host, --target
configure: WARNING: you should use --build, --host, --target
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... Invalid configuration `$': machine `$' not recognized
configure: error: /bin/bash ./config.sub $ failed
```

```
cd ./base && make
make[1]: Entering directory '/home/ubuntu/nagioscore-nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
nagios.c: In function 'main':
nagios.c:611:4: warning: ignoring return value of 'asprintf', declared with attr
ibute warn_unused_result [-Wunused-result]
  611 |    asprintf(&mac->x[MACRO_PROCESSSTARTTIME], "%llu", (unsigned long long
)program_start);
      |    ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~
nagios.c:841:4: warning: ignoring return value of 'asprintf', declared with attr
ibute warn_unused_result [-Wunused-result]
  841 |    asprintf(&mac->x[MACRO_EVENTSTARTTIME], "%llu", (unsigned long long)e
vent_start);
      |    ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~
```

Create the Nagios user and group, and add the 'www-data' Apache user to the 'nagios' group.

$ sudo make install-groups-users

```
sudo apt install mono-devel

praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install-groups-users
```

```
Group nagios already exists
User nagios already exists
```

$ sudo usermod -a -G nagios www-data

```
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo usermod -a -G nagios www-data
usermod: group 'nagios' does not exist
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install
```

Install Nagios binaries, service daemon script, and the command mode $ sudo

make install

```
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo usermod -a -G nagios www-data
usermod: group 'nagios' does not exist
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install
```

```
cd ./base && make install
make[1]: Entering directory '/home/ubuntu/nagioscore-nagios-4.4.6/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 nagiostats /usr/local/nagios/b
in
make[1]: Leaving directory '/home/ubuntu/nagioscore-nagios-4.4.6/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ubuntu/nagioscore-nagios-4.4.6/cgi'
make install-basic
make[2]: Entering directory '/home/ubuntu/nagioscore-nagios-4.4.6/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/nagio
s/sbin; \
```

$ sudo make install-daemoninit

```
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install
sudo: make: command not found
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install-daemoninit
```

```
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/
system/nagios.service

*** Init script installed ***
```

$ sudo make install-commandmode

```
sudo: make: command not found
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install-commandmode
sudo: make: command not found
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$  sudo make install-config
```

```
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

After that, install the sample script configuration.

$ sudo make install-config

```
sudo: make: command not found
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install-commandmode
sudo: make: command not found
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$  sudo make install-config
```

```
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/
local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/loc
al/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /us
r/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
```

Then install the Apache configuration for Nagios and activate the mod_rewrite and mode_cgi modules.

```
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo make install-webcon
sudo: make: command not found
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ for svc in Apache ssh
> do
```

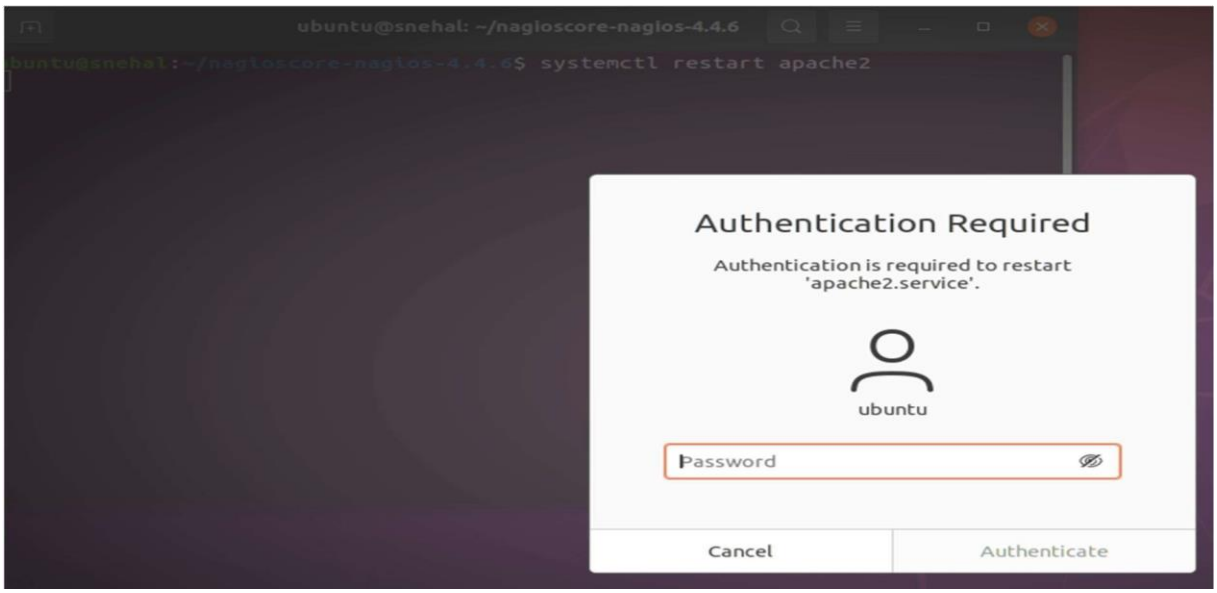
$ sudo make install-webconf

```
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/n
agios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/apache2/sites-enabled/nagios.conf /etc/apache2/sites-enabled/
nagios.conf; \
fi

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

$ sudo a2enmod rewrite cgi

Now restart the Apache service.

$ systemctl restart apache2



And we have installed the Nagios Core 4.4.6.

- Create nagiosadmin user

After installing the Nagios Core, we will add the basic authentication for accessing the Nagios dashboard. And we will be using the basic Apache authentication.

Create a new apache basic authentication for the user the "nagiosadmin". $

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

strong password.

And we have created a new user 'nagiosadmin' for the Nagios dashboard authentication.

- Setup UFW Firewall

For the firewall configuration, you will need to add the Apache service and the Nagios server port to the UFW firewall.

Add the SSH and Apache HTTP port using the ufw command below.

$ for svc in Apache ssh do
sudo ufw allow $svc done

```
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ for svc in Apache ssh
> do
> sudo ufw allow $svc
> done
ERROR: Could not find a profile matching 'Apache'
Rules updated
Rules updated (v6)
```

Next, start the UFW firewall service and add it to the system boot.
$ sudo ufw enable

```
Rules updated (v6)
praj@it14-V530-15ICB:~/nagioscore-nagios-4.4.6$ sudo ufw enable
Firewall is active and enabled on system startup
```

Type 'y' and the UFW firewall service will be activate.

Now check all available rules using the command below.

$ sudo ufw status numbered

Now we will get both the SSH and Apache services added to the UFW firewall.

```
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] Apache                     ALLOW IN    Anywhere
[ 2] 22/tcp                     ALLOW IN    Anywhere
[ 3] Apache (v6)                ALLOW IN    Anywhere (v6)
[ 4] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

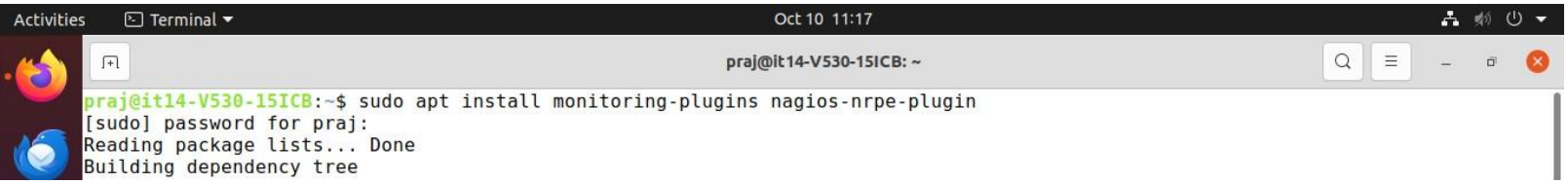
And finally, you've completed the Nagios Core installation on the Ubuntu 20.04 server.

Step 3 - Install Nagios Plugins and NRPE Plugin

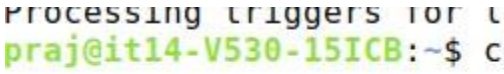
After installing the Nagios Core, we will install the Nagios Plugins and NRPE Plugins.

Both Nagios and NRPE plugins are available by default on the Ubuntu repository. You can install those packages using the apt command below.

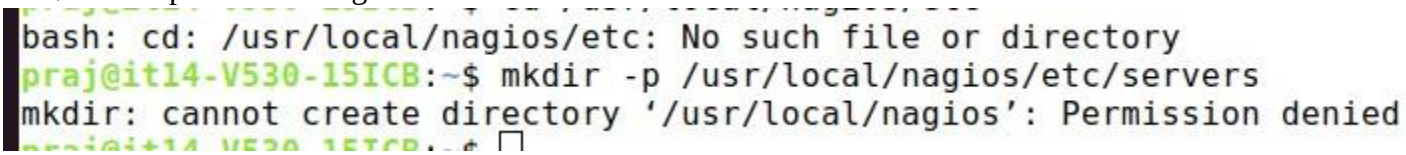
$ sudo apt install monitoring-plugins nagios-nrpe-plugin



Once the installation is complete, go to the nagios installation directory "/usr/local/nagios/etc" and create a new directory for for storing all server hosts configuration. $ cd /usr/local/nagios/etc

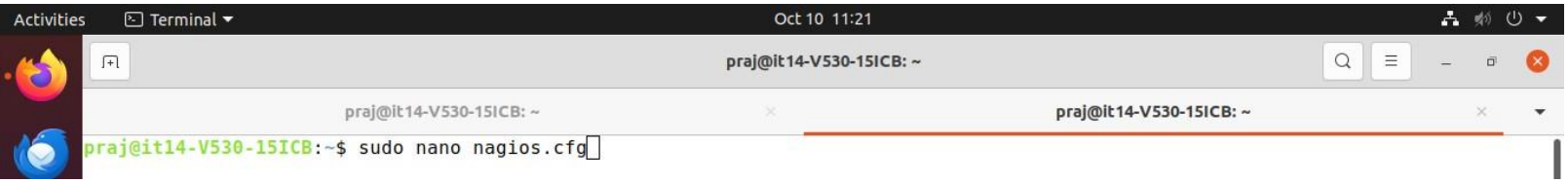


$ mkdir -p /usr/local/nagios/etc/servers



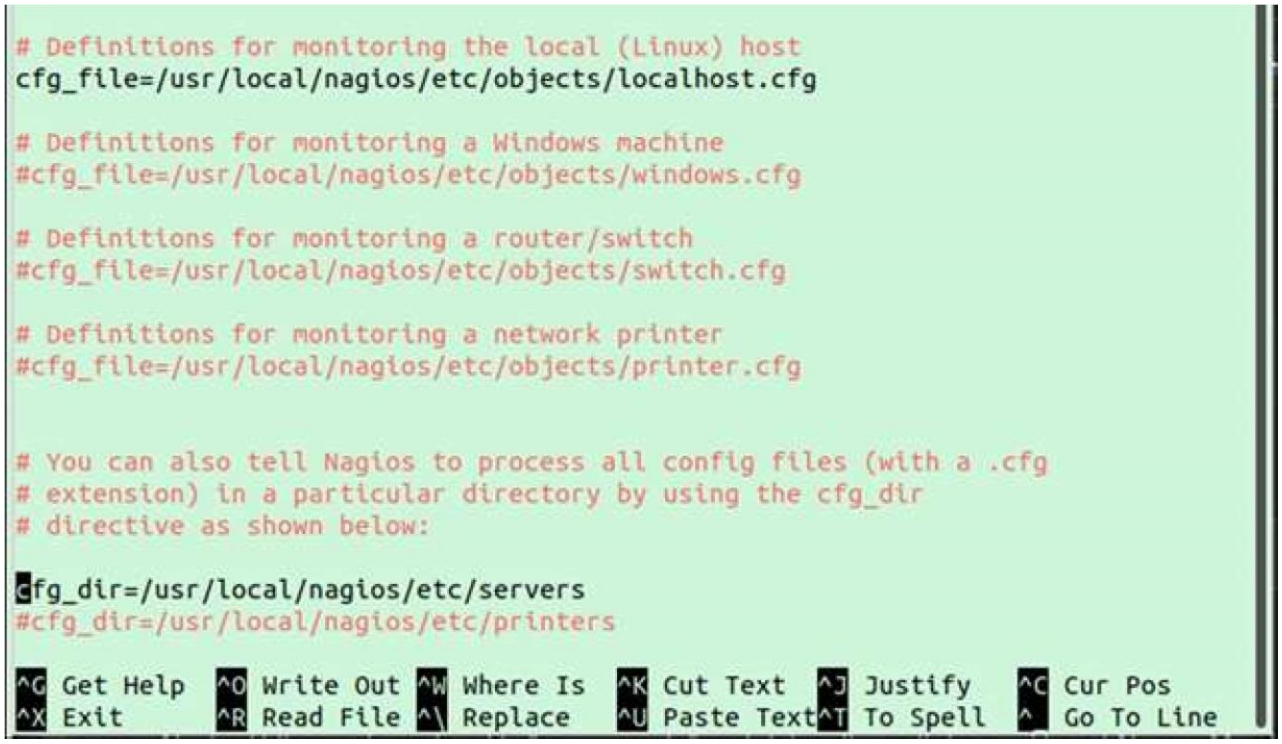
Next, edit the Nagios configuration 'nagios.cfg' using vim editor. $

sudo nano nagios.cfg



Uncomment the 'cfg_dir' option that will be used for sotring all server hots configurations.

cfg_dir=/usr/local/nagios/etc/servers \



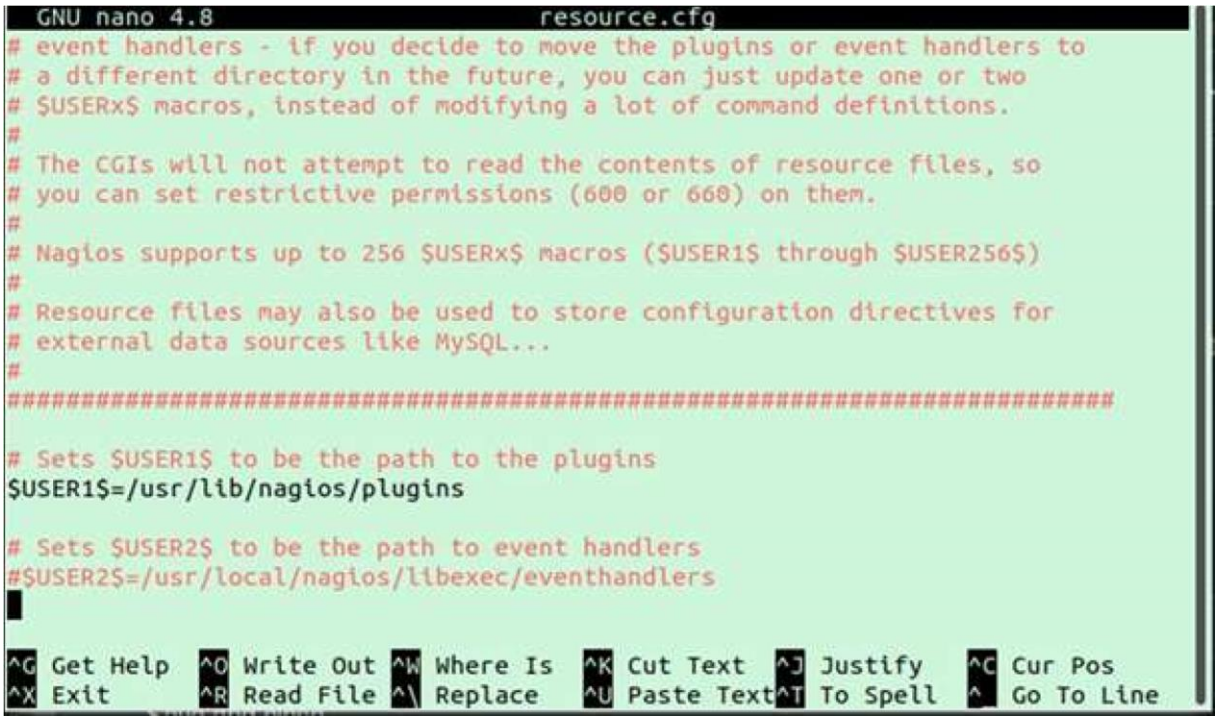
Save and close.

Next edit the configuration file "resource.cfg" and define the path binary files of Nagios Monitoring Plugins.

$ sudo nano resource.cfg

Define the Nagios Monitoring Plugins path by changing the default configuration as below.

$USER1$=/usr/lib/nagios/plugins



Save and close.

After that, add the nagios admin email contacts by editing the configuration file "objects/contacts.cfg".
$ sudo nano objects/contacts.cfg

Change the email address with your own.

```
 GNU nano 4.8                    objects/contacts.cfg                 Modified
#
###############################################################################

# 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 fr>
    alias                   Nagios Admin            ; Full name of user
    email                   raisnehalonline@gmail.com ; <<***** CHANGE THIS TO >
}



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

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

Save and close.

Now define the nrpe check command by editing the configuration file

"objects/commands.cfg". $ sudo nano objects/commands.cfg

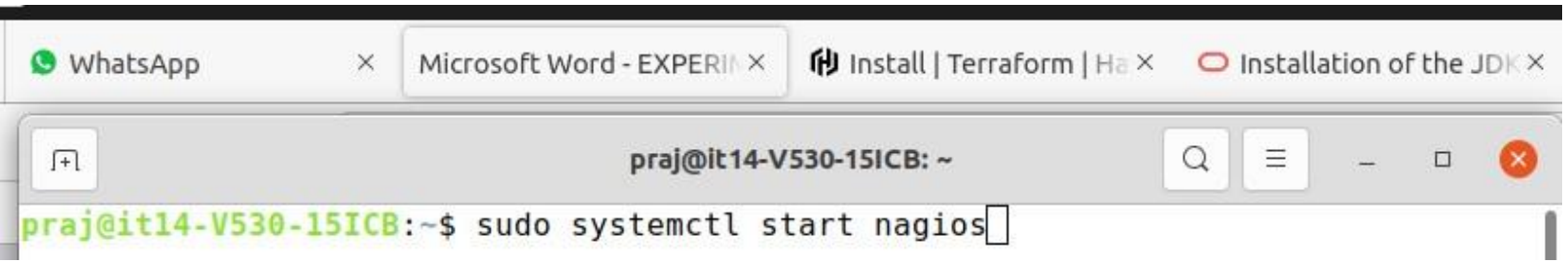
```
praj@it14-V530-15ICB: ~
praj@it14-V530-15ICB:~$ sudo nano objects/commands.cfg
```

Add the following configuration to the end of the line.

```
###############################################################################

define command {

    command_name    process-host-perfdata
    command_line    /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTA>
}


define command {

    command_name    process-service-perfdata
    command_line    /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERV>
}
define command{
        command_name check_nrpe
        command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

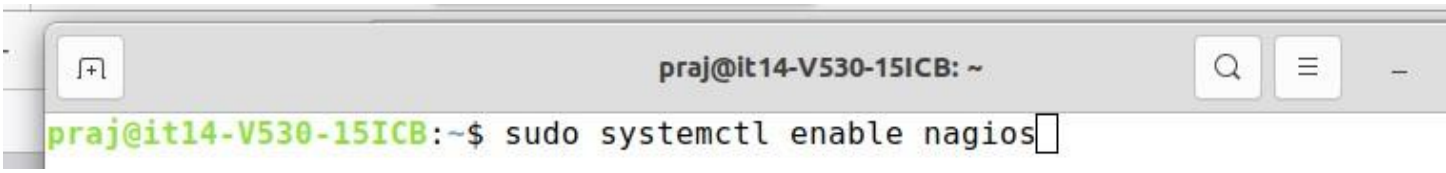
Save and close, and the Nagioscore configuration has been completed.

Next, start the Nagios service and add it to the system boot. $ sudo systemctl start nagios

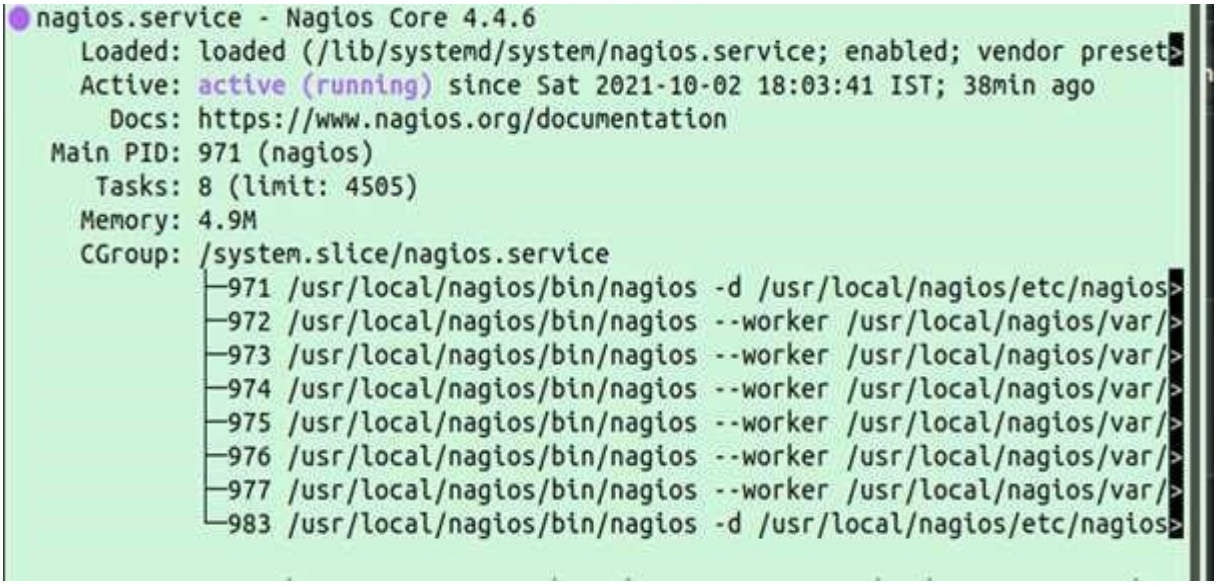


$ sudo systemctl enable nagios

$ sudo systemctl status nagios

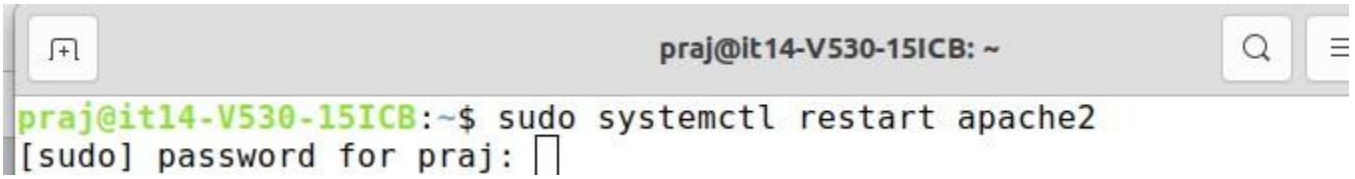


The Nagios service is up and running, check using the following command.



As a result, the Nagios service is up and running. Now we need to restart the Apache service to apply a new Nagios configuration.

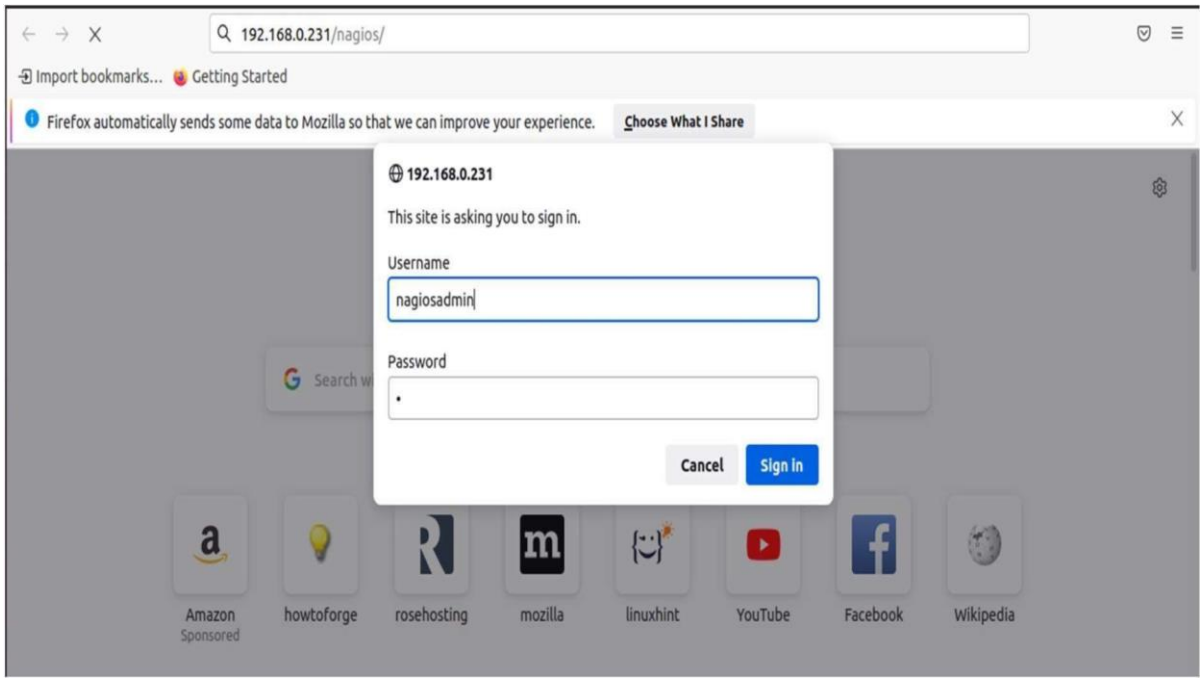
$ sudo systemctl restart apache2

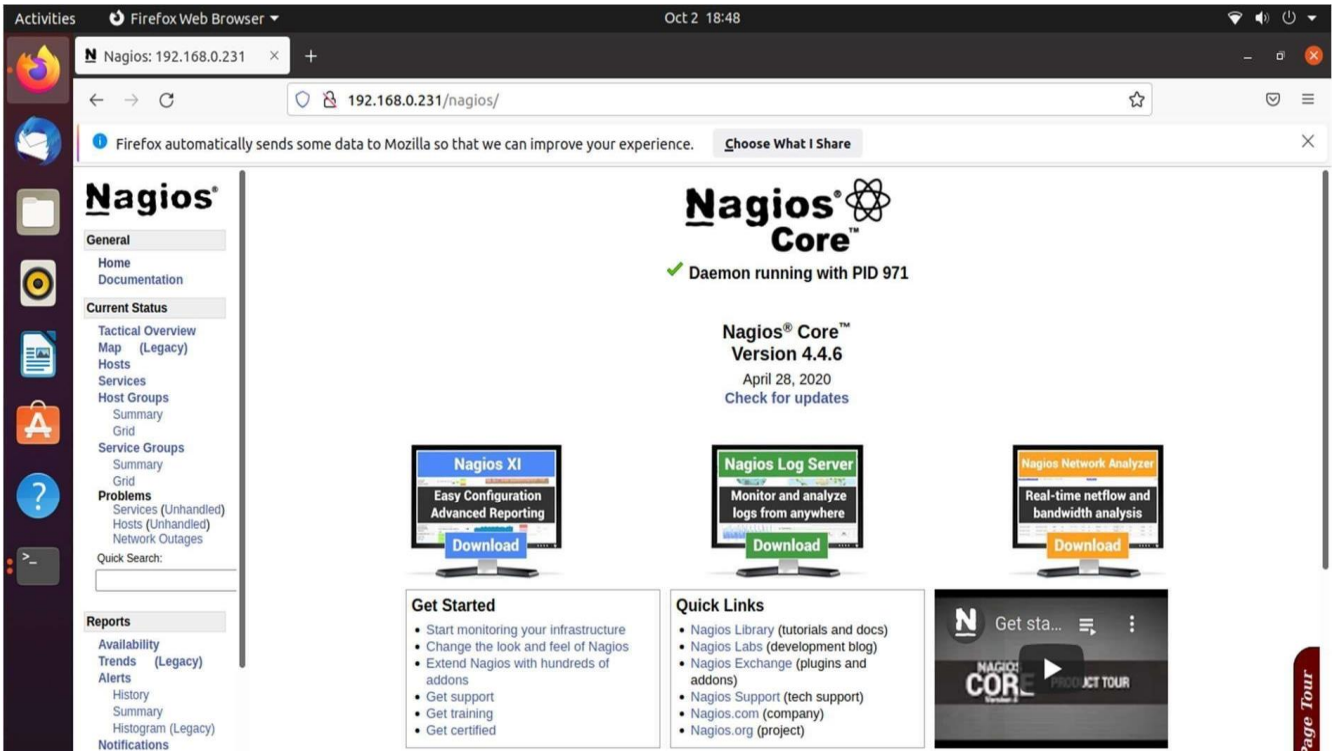


And the Nagios configuration has been completed.
Open your web browser and type the server IP address following the "nagios" URL path.

http://your_ip_add/nagios/

Log in with the user "nagiosadmin" and type your password





Conclusion -

Nagios Log Server greatly simplifies the process of searching your log data. Set up alerts to notify you when potential threats arise, or simply filter your data to quickly audit your system. With Log Server, you get all of your data in one location, with high availability and fail-over built right in. Thus, with all commands we have installed Nagios on Ubuntu servers.