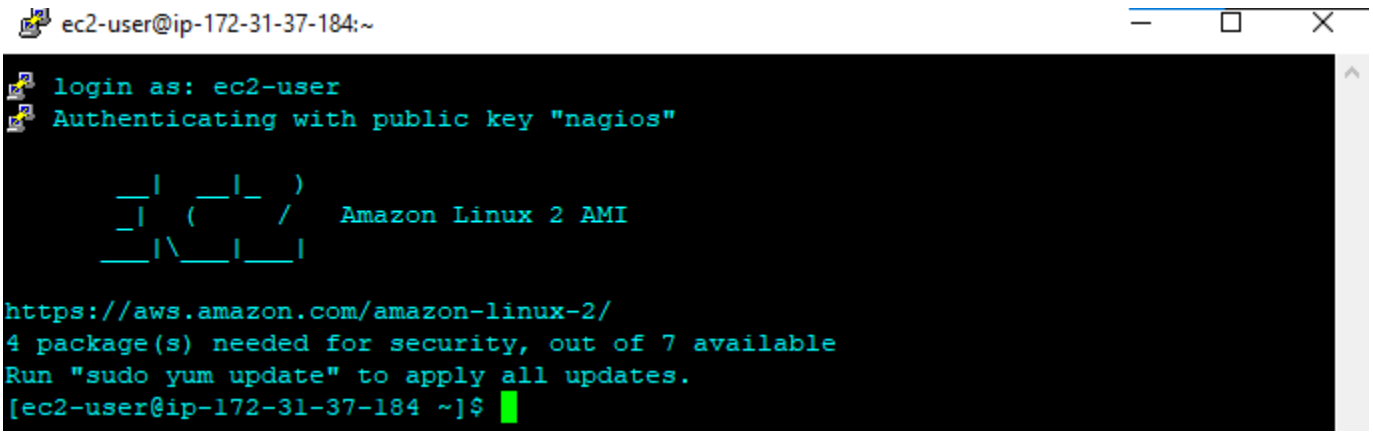
Nagios Configuration with AWS

Create AWS EC2

- Amazon Linux 2 Kernel 5.10 AMI EC2
- Provide network group for SSH, HTTP, HTTPS, TCP
- Create 2 instances as below
    - Master
    - Slave

Putty Connection

- Download Putty
- Copy Public IP of Master
- Connect with authentication
- Username: ec2-user



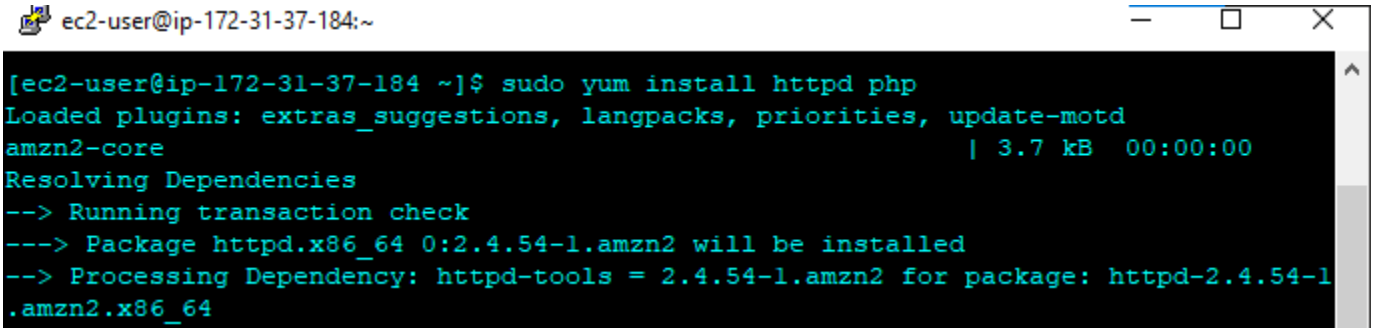
Install Pre-requisite Software

Nagios requires the following packages are installed on your server prior to installing nagios:

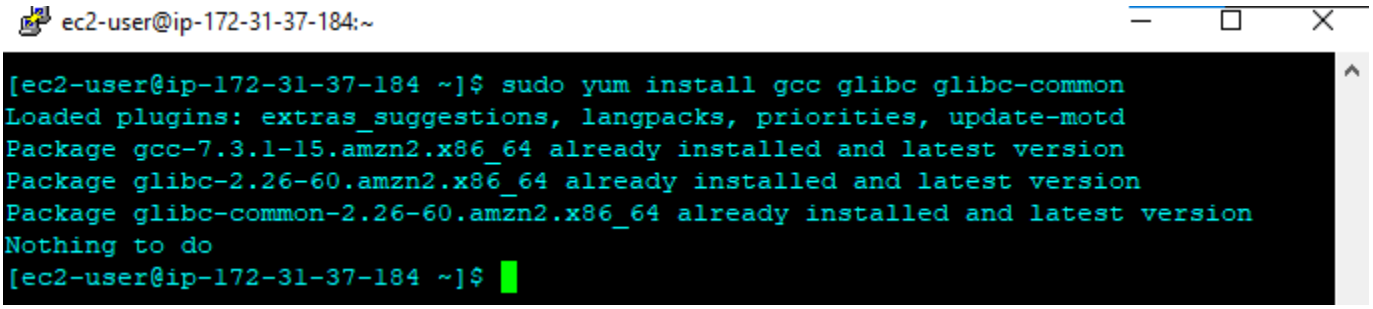
- Apache
- PHP
- GCC compiler collection (glibc GNU C library, glibc-common: common libraries)
- GD development libraries (Graphic libraries)

You can use yum to install these packages by running the following commands (as ec2-user):

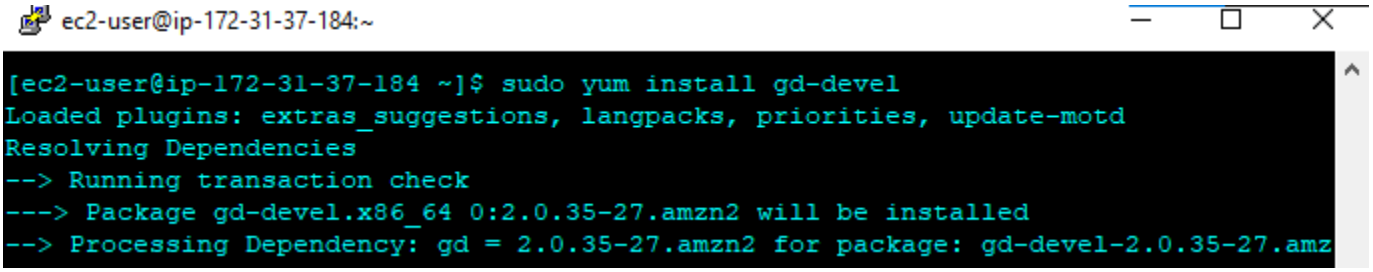
sudo yum install httpd php



sudo yum install gcc glibc glibc-common



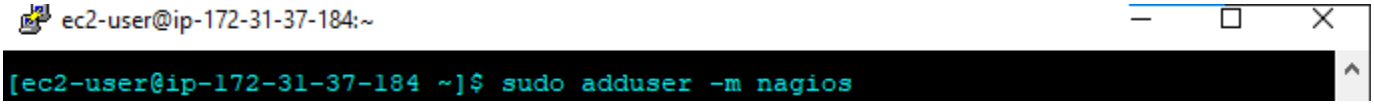
sudo yum install gd gd-devel



Create Account Information

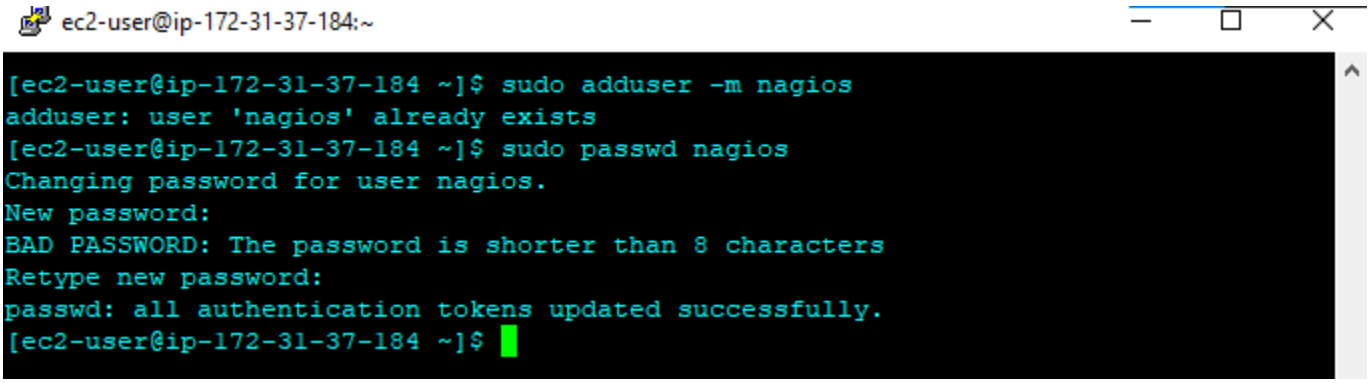
You need to set up a Nagios user. Run the following commands:

sudo adduser -m nagios //creates the user's home directory -m



sudo passwd nagios

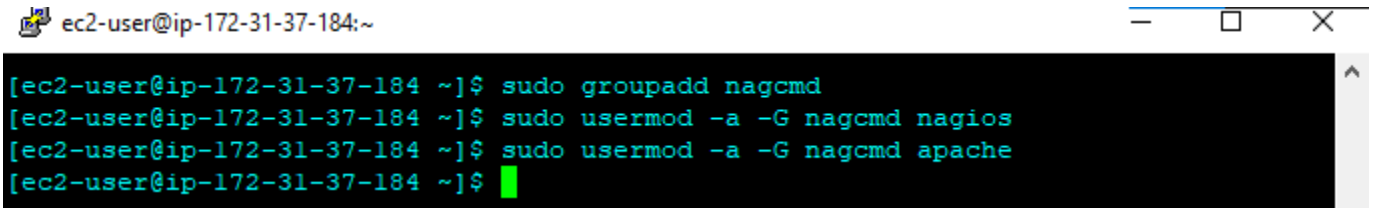
//Type the new password twice.



sudo groupadd nagcmd

sudo usermod -a -G nagcmd nagios //-a append and -G group

sudo usermod -a -G nagcmd apache

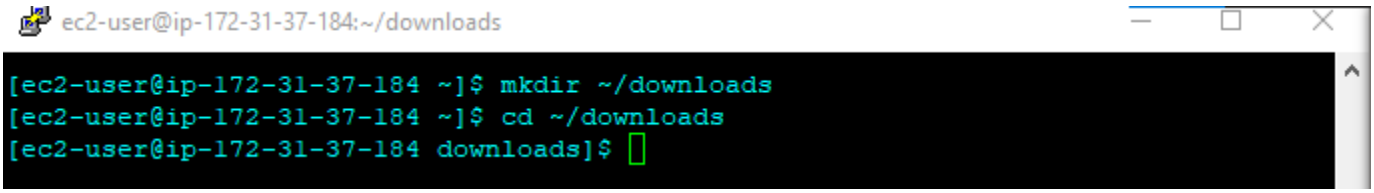


Download Nagios Core and the Plugins

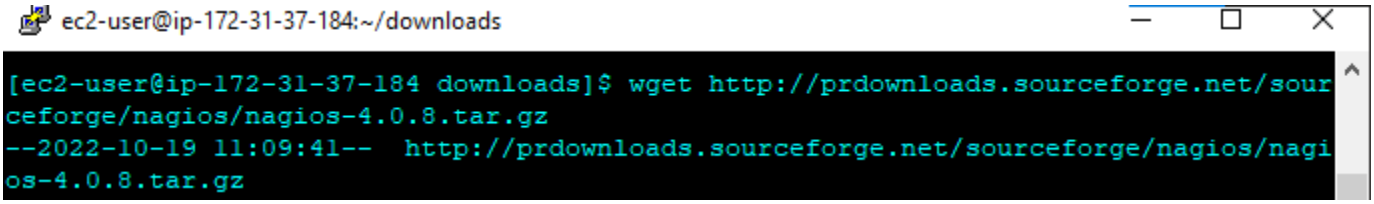
Create a directory for storing the downloads.

mkdir ~/downloads

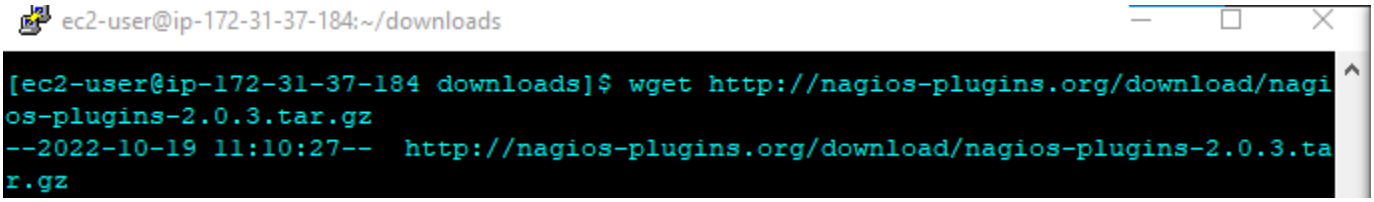
cd ~/downloads



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



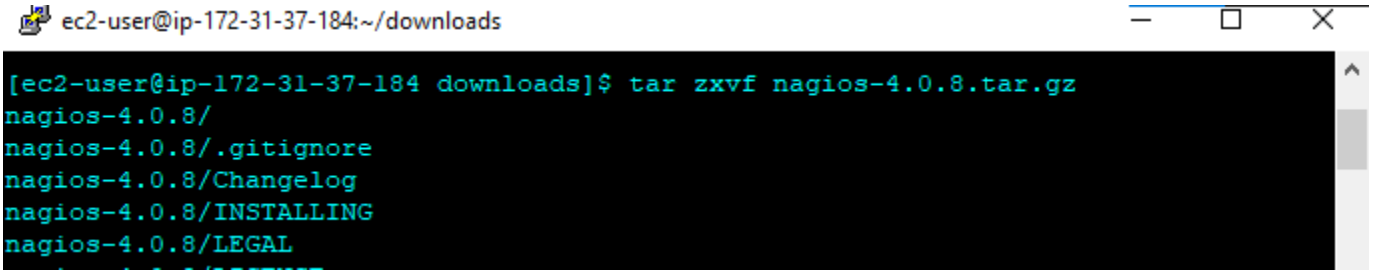
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz



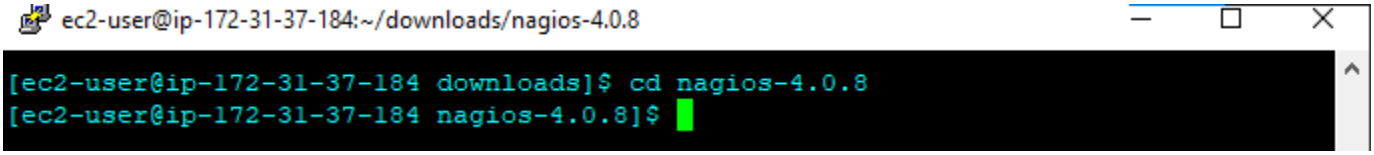
Compile and Install Nagios

Extract the Nagios source code tarball.

tar zxvf nagios-4.0.8.tar.gz

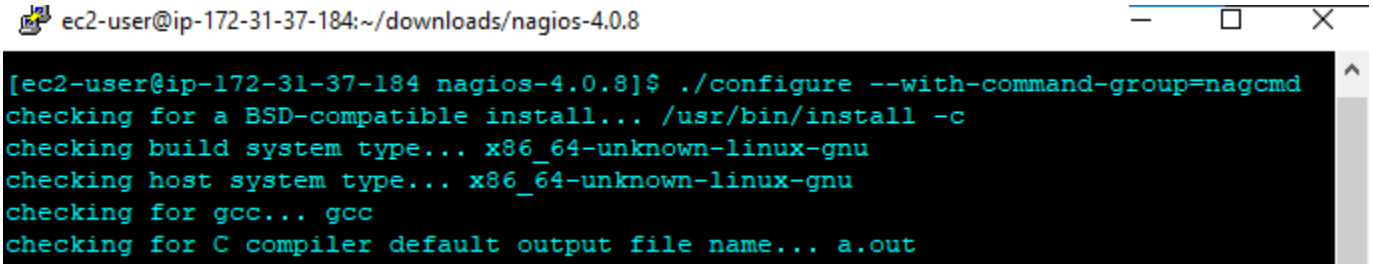


cd nagios-4.0.8



Run the configuration script with the name of the group which you have created in the above step.

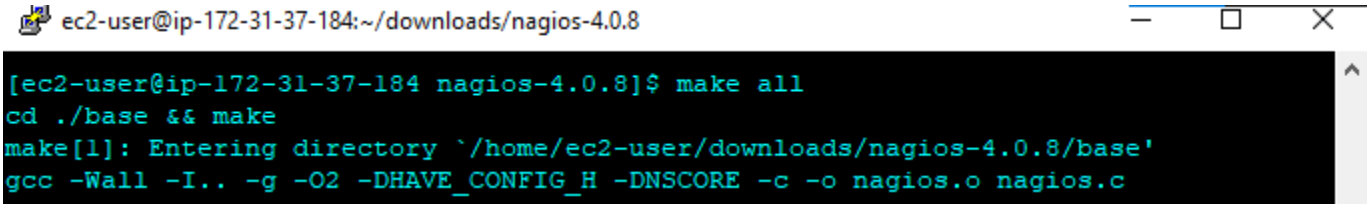
./configure --with-command-group=nagcmd



Compile and Install Nagios

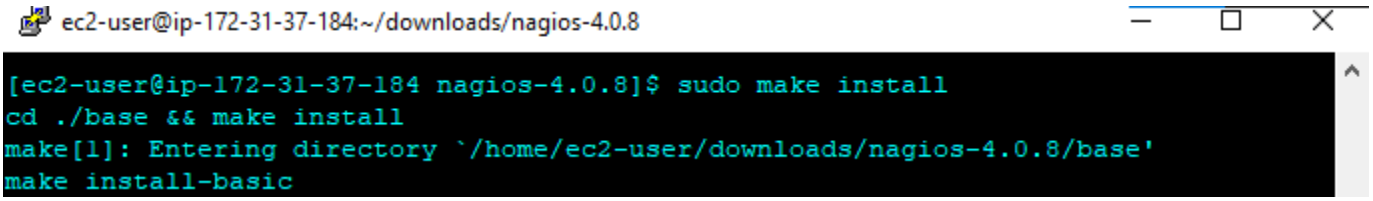
Compile the Nagios source code.

make all

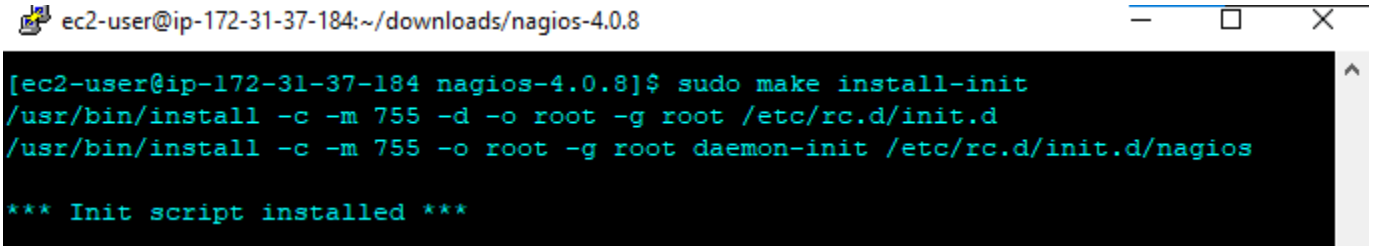


Install binaries, init script, sample config files and set permissions on the external command directory.

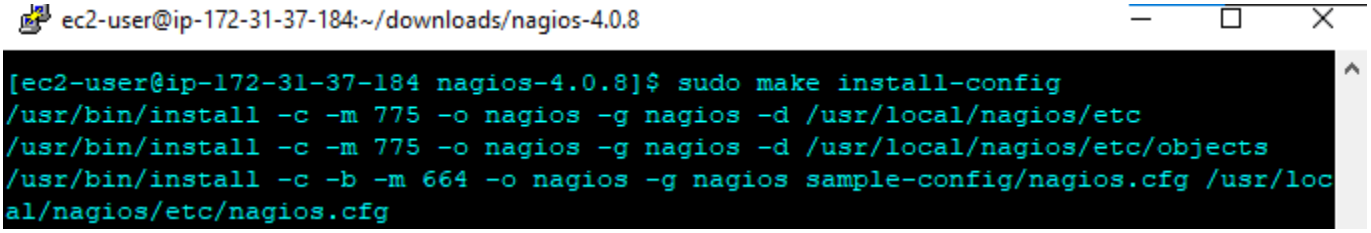
sudo make install



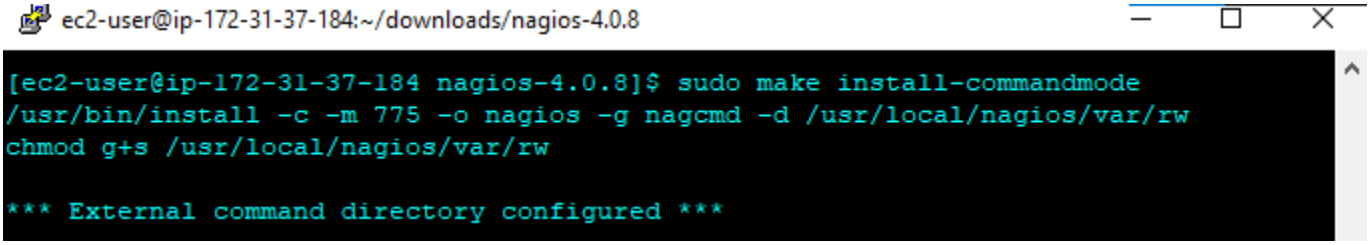
sudo make install-init



sudo make install-config



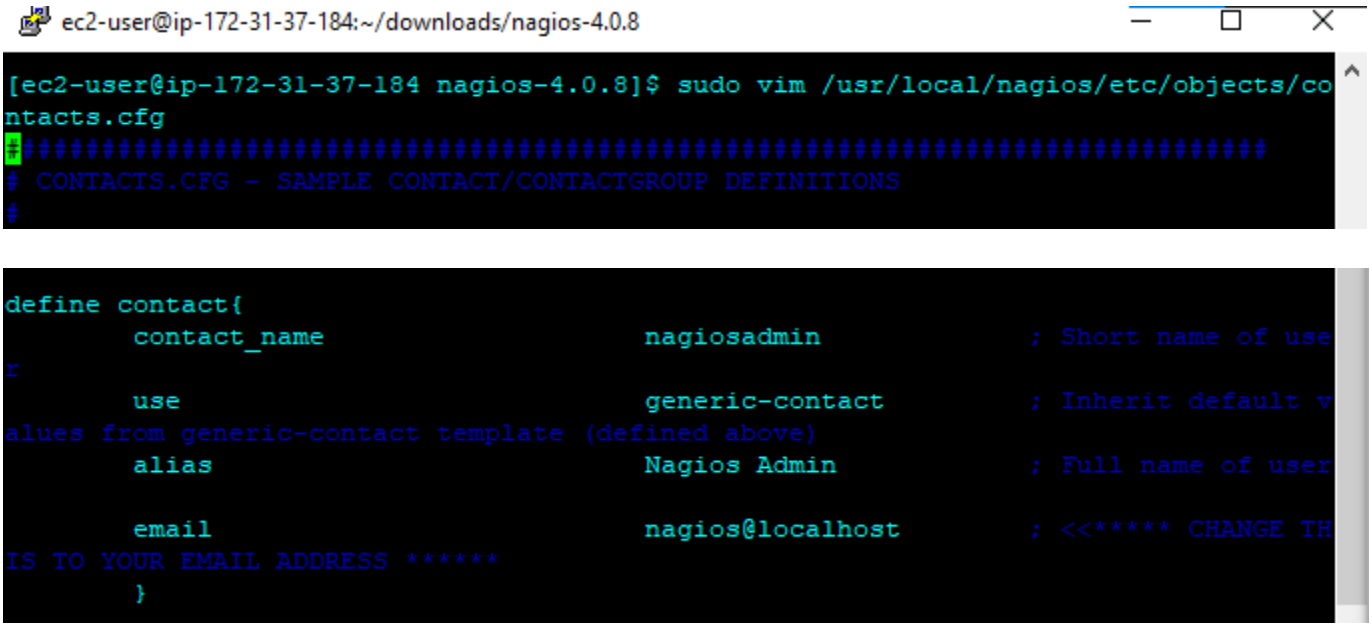
sudo make install-commandmode



Customize Configuration

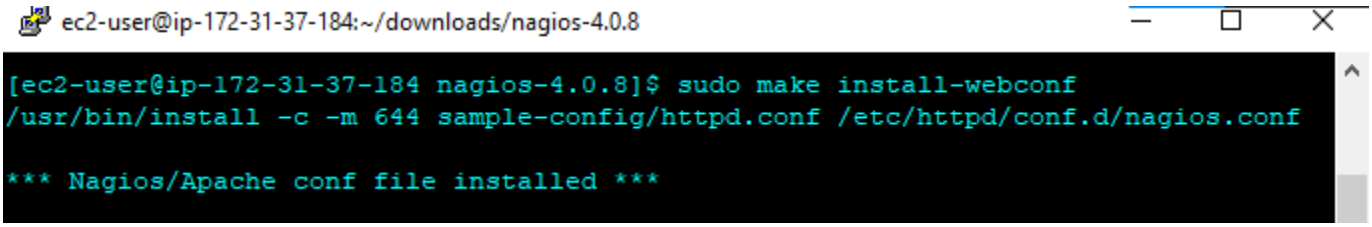
Change E-mail address with the nagiosadmin contact definition you'd like to use for receiving Nagios alerts.

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



Configure the Web Interface

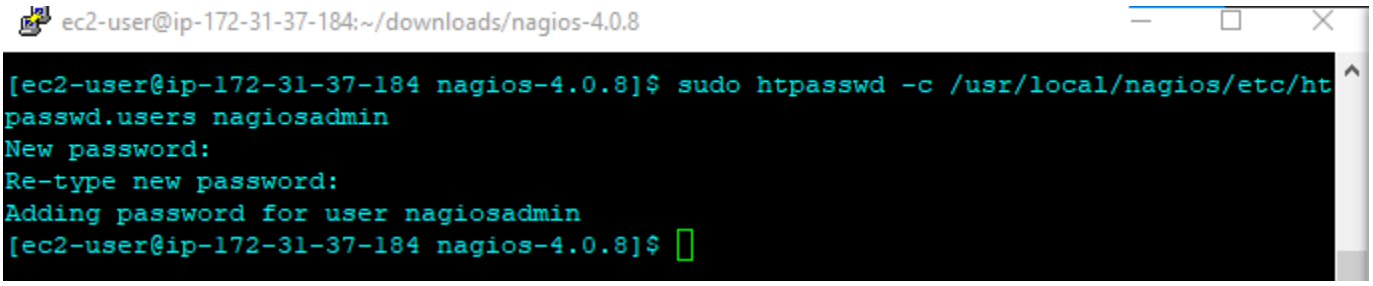
sudo make install-webconf



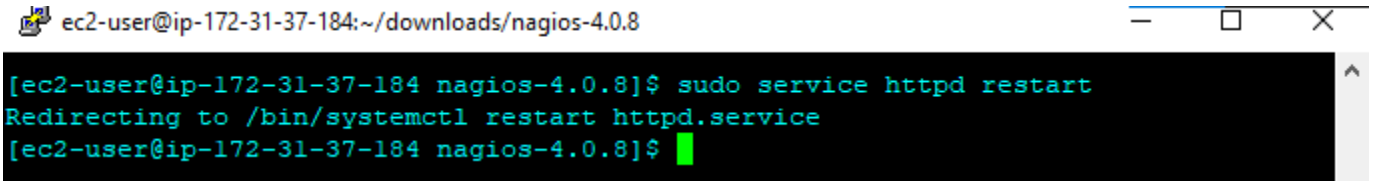
Create a nagiosadmin account for logging into the Nagios web interface. Note the password you need it while login to nagios web console.

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

//Type the new password twice.



sudo service httpd restart //Restart service

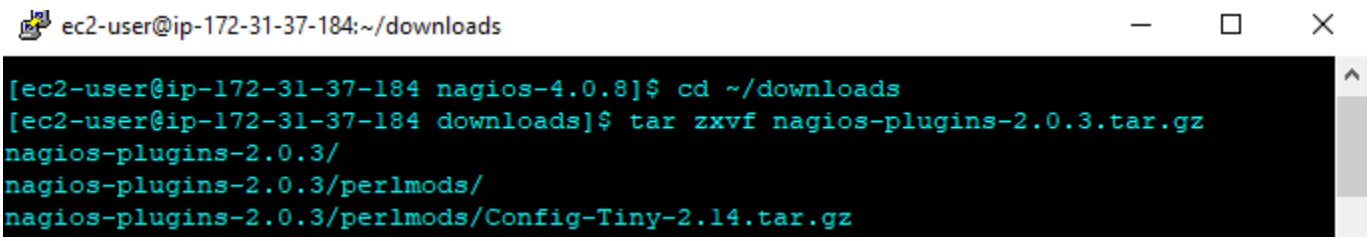


Compile and Install the Nagios Plugins

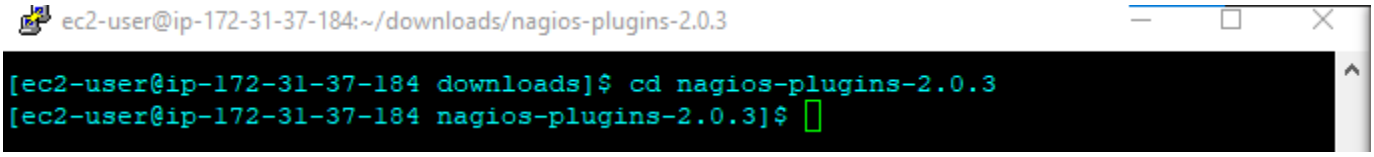
Extract the Nagios plugins source code tarball.

cd ~/downloads

tar zxvf nagios-plugins-2.0.3.tar.gz



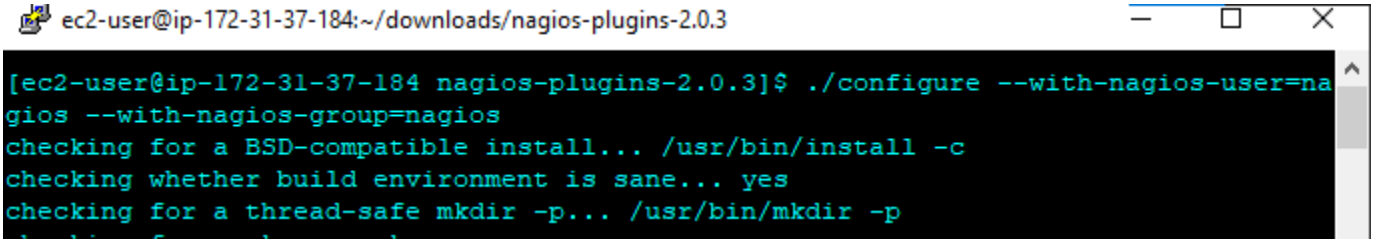
cd nagios-plugins-2.0.3



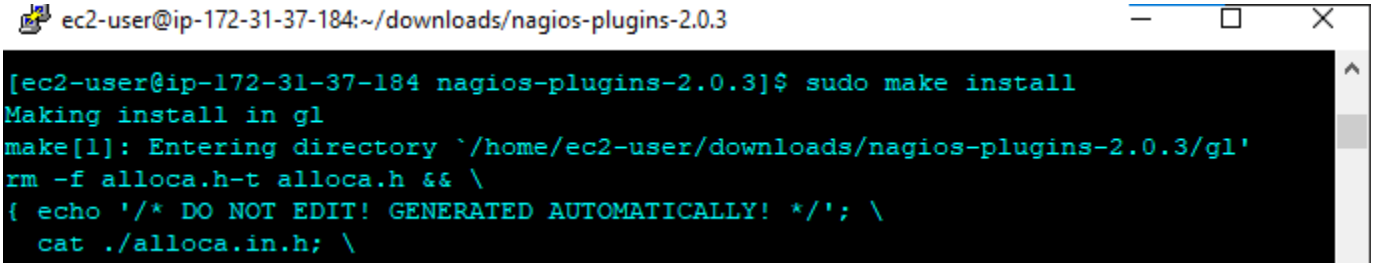
Compile and Install the Nagios Plugins

Compile and install the plugins.

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



sudo make install

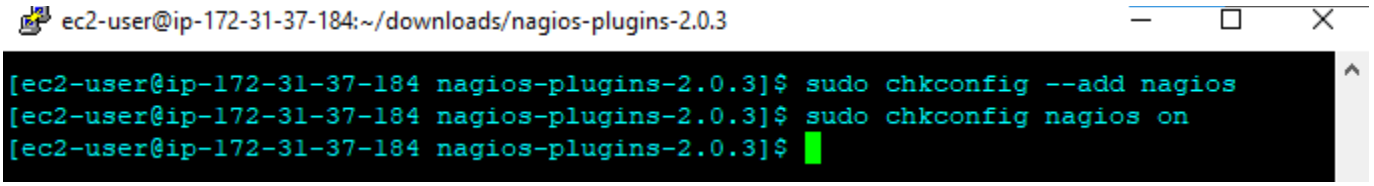


Start Nagios

Add Nagios to the list of system services and have it automatically started when the system boots.

sudo chkconfig --add nagios

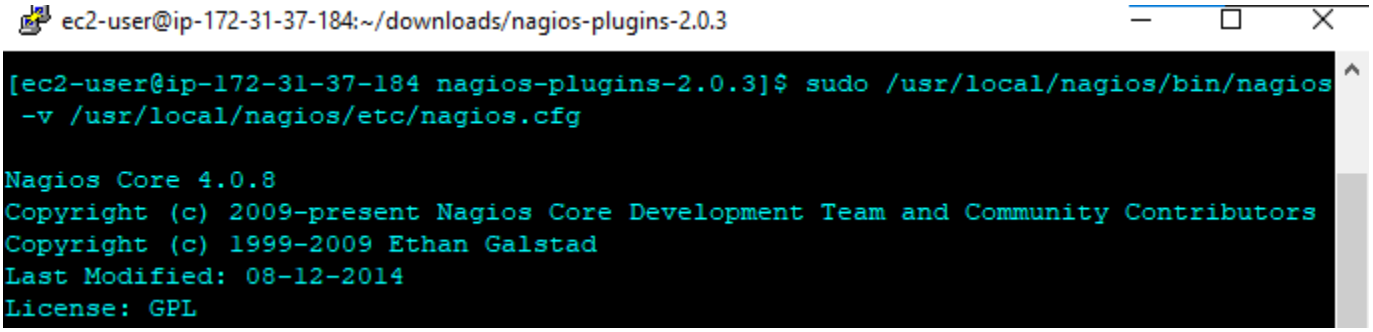
sudo chkconfig nagios on



Start Nagios

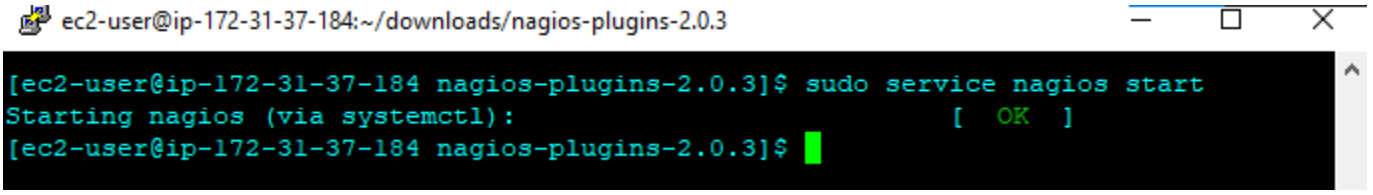
Verify the sample Nagios configuration files.

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



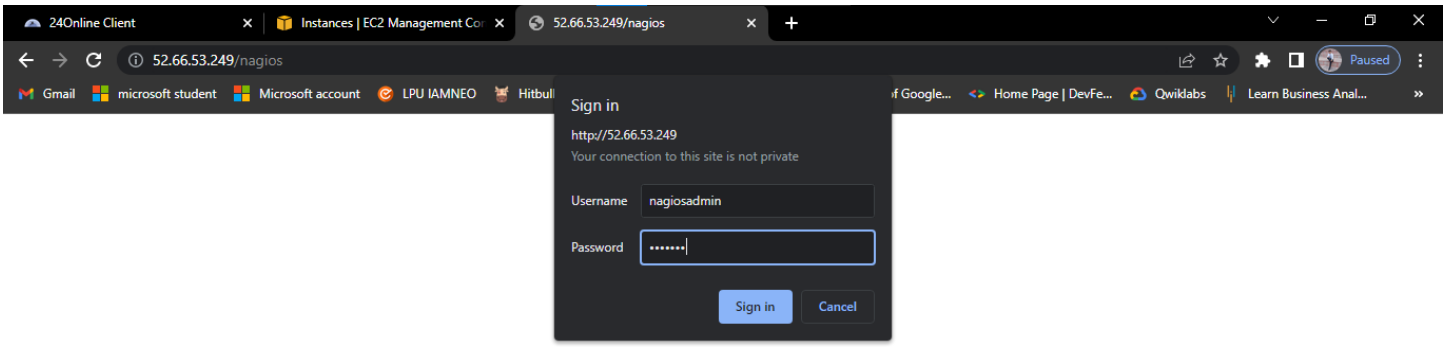
If there are no errors, start Nagios.

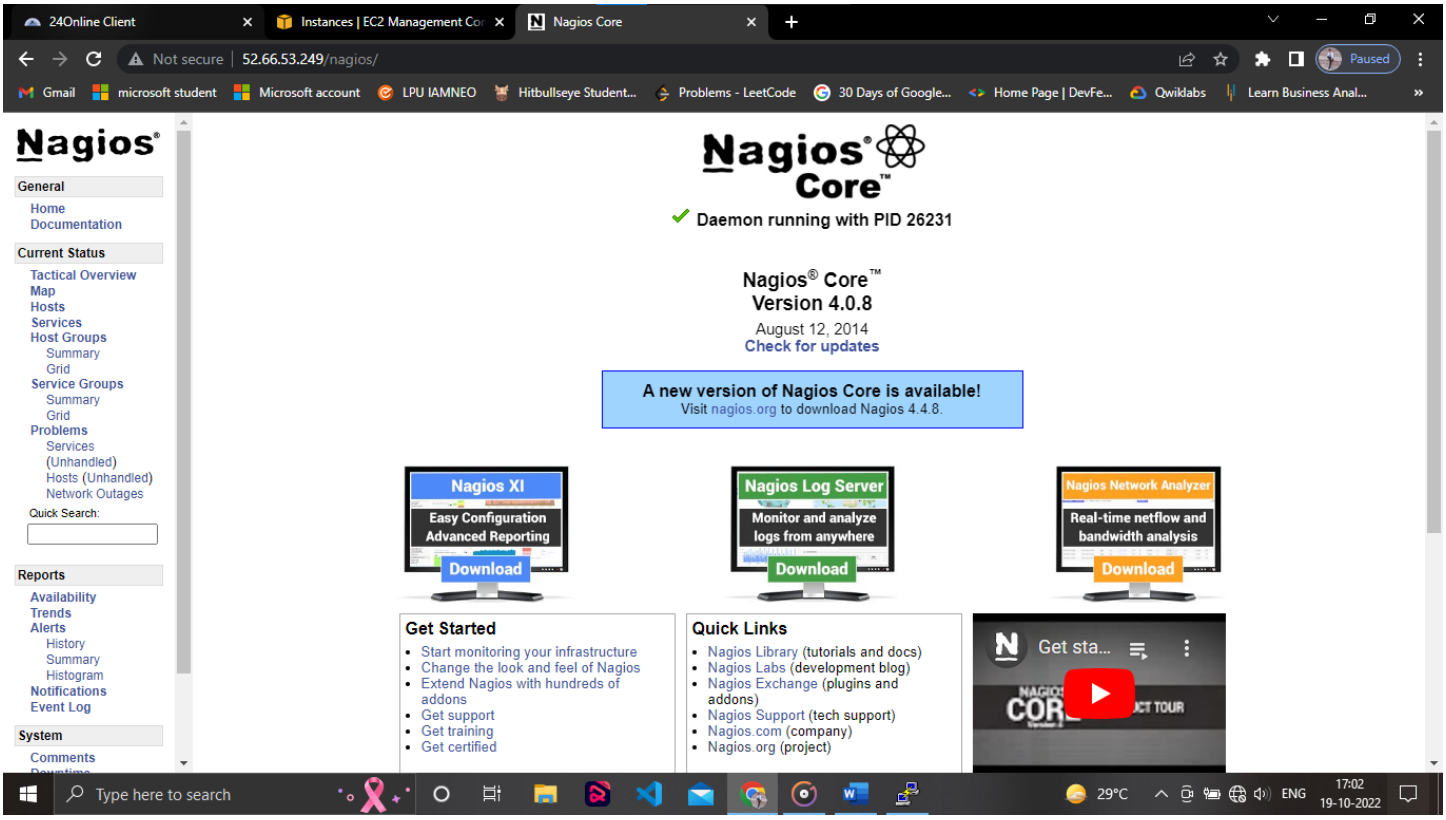
sudo service nagios start



Access the Nagios web interface to do this you will need to know the Public DNS or IP for your instance, you can get this from the Instance section of the EC2 Console if you do not already know it. You'll be prompted for the username (nagiosadmin) and password you specified earlier.

(Public ip of master)/nagios



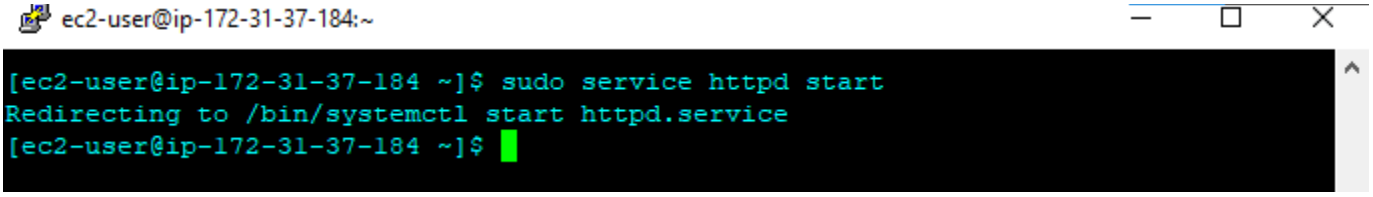


Nagios Adding Hosts

Starting the Nagios service in Master

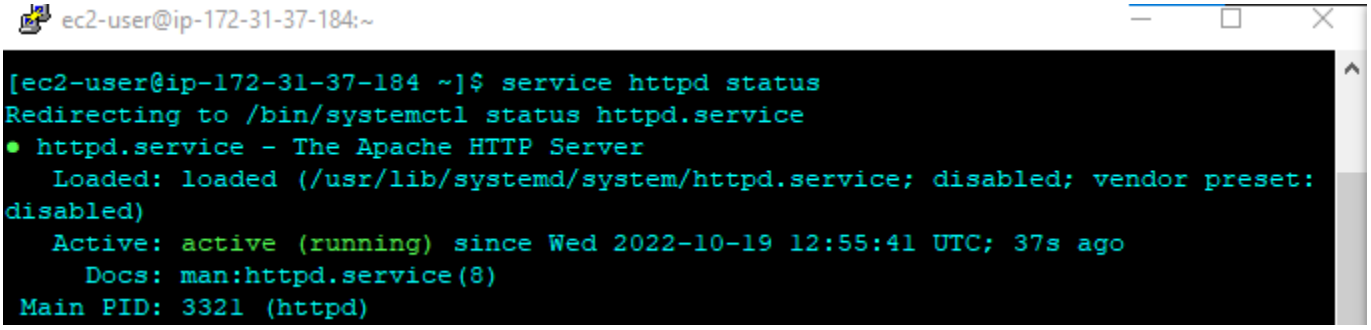
Start httpd

sudo service httpd start

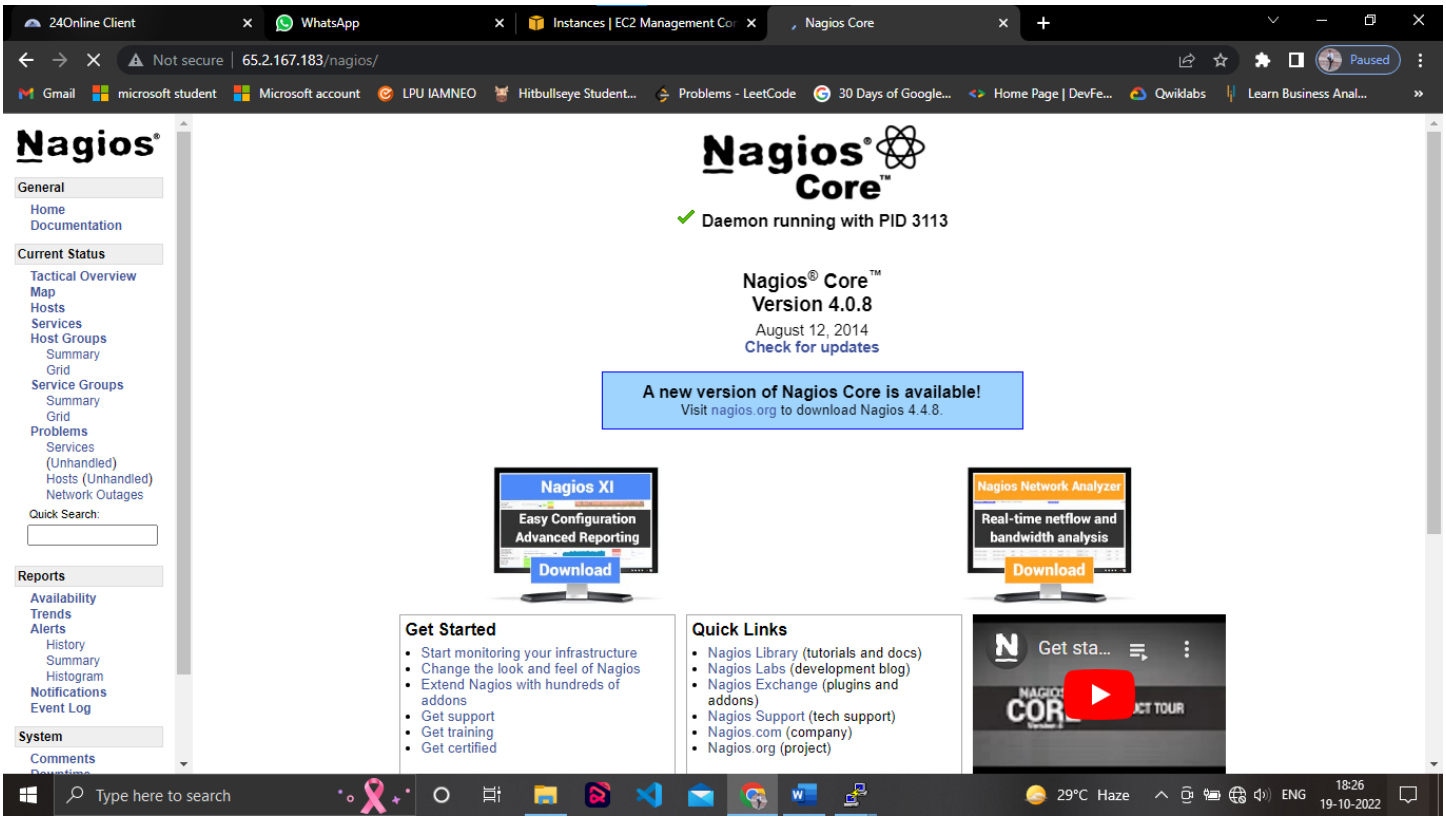


Test if Apache running

service httpd status



Check from browser



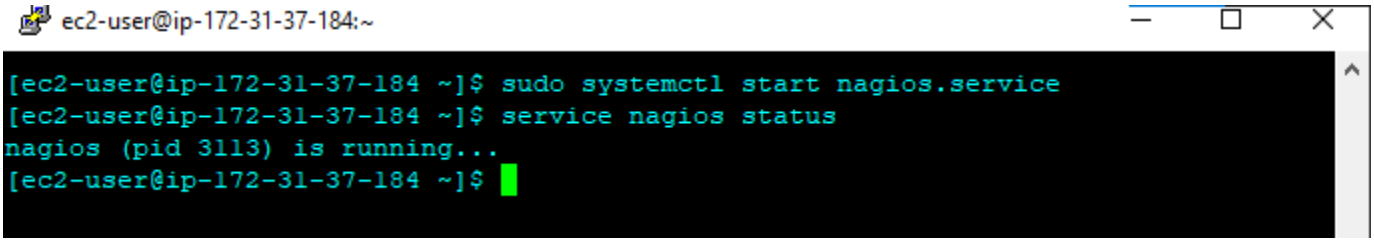
Starting the Nagios service in Master

Start Nagios

sudo systemctl start nagios.service

Check if it running

service nagios status



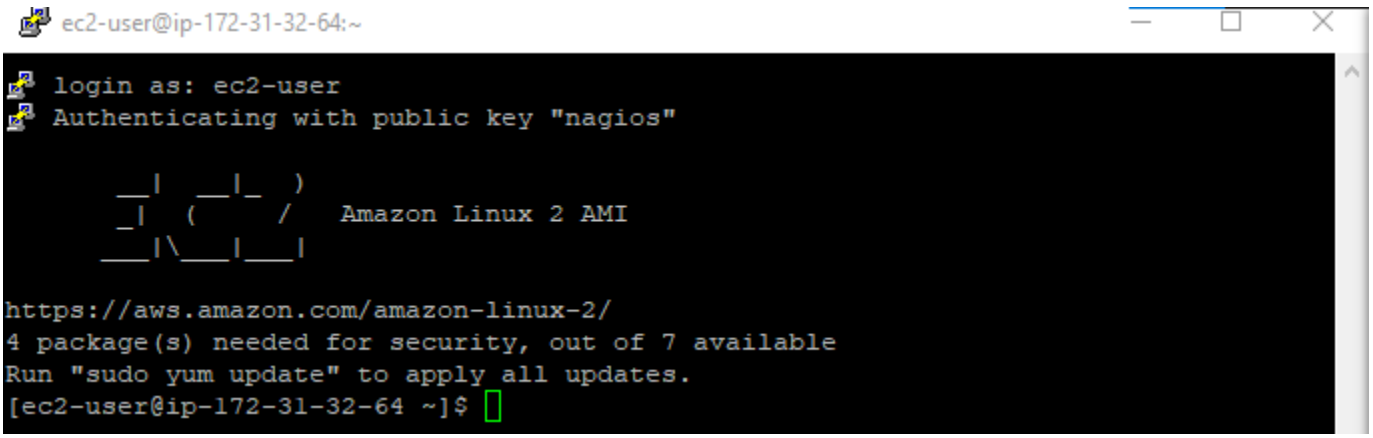
Check from browser

Create AWS EC2

- Amazon Linux 2 Kernel 5.10 AMI EC2
- Provide all network group
- Create the number of slave instances as below
  - Slave1
  - Slave2
  - ….
  - ….
  - Slaven

Putty Connection

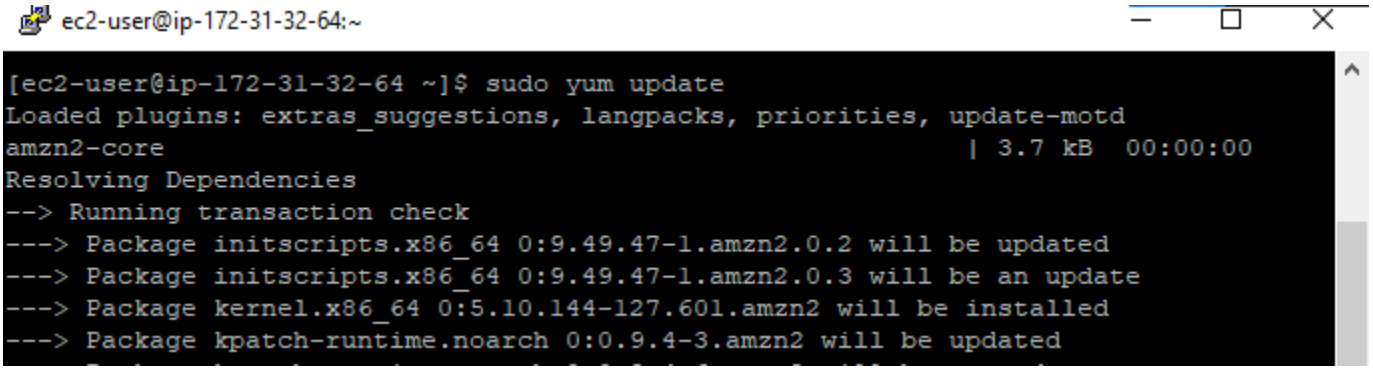
- Download Putty
- Copy Public IP of master
- Connect with authentication
- Username: ec2-user



Install Pre-requisites for Slave

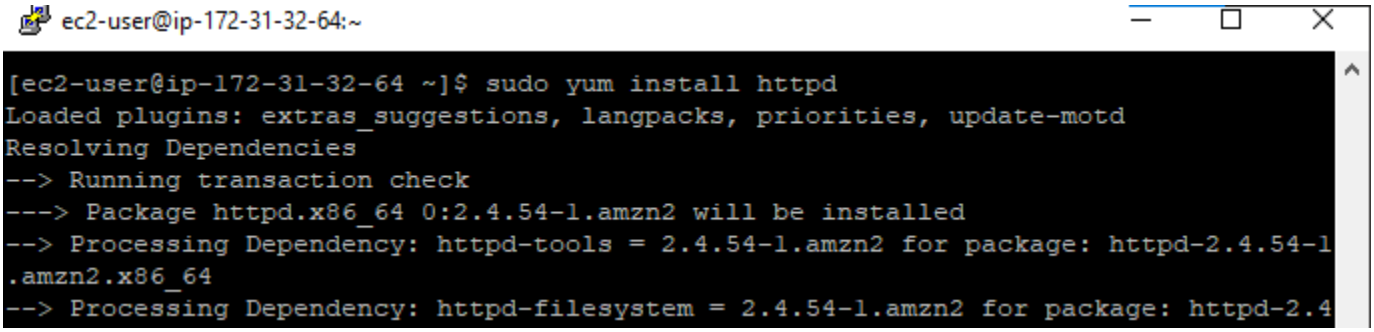
Update server

sudo yum update



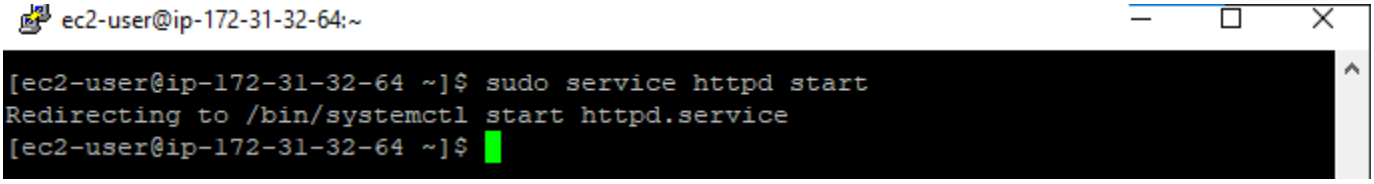
Install httpd

sudo yum install httpd



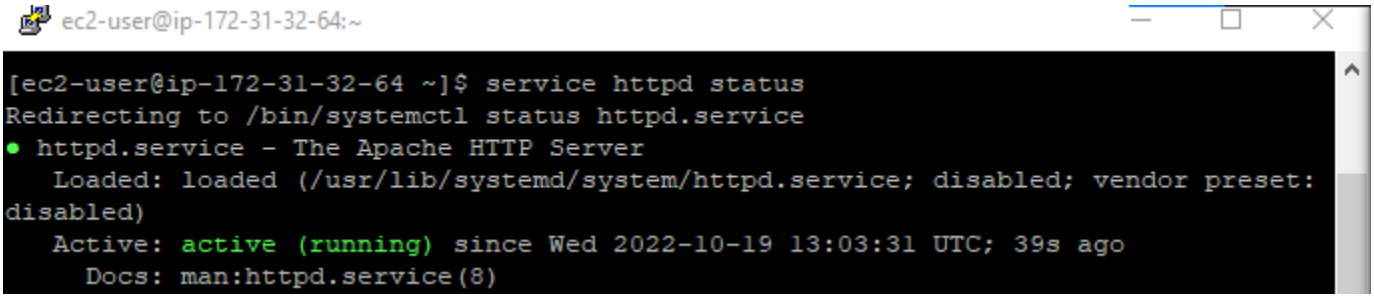
Start httpd

sudo service httpd start



Test if Apache running

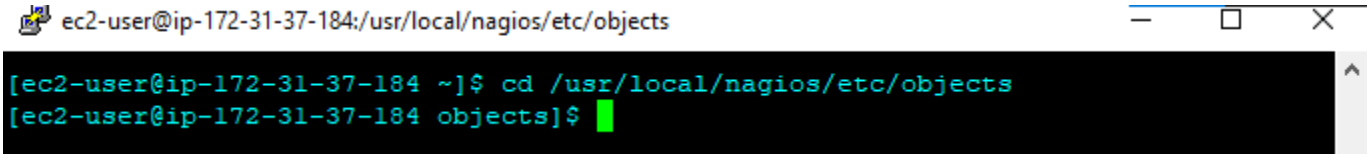
service httpd status



Configuration on Master

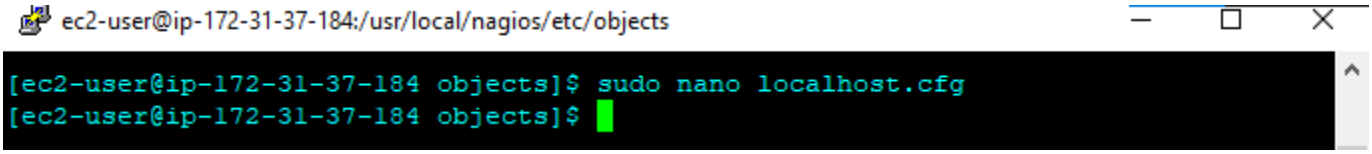
Move to objects folder

cd /usr/local/nagios/etc/objects



Check localhost.cfg

sudo nano localhost.cfg



Copy the content

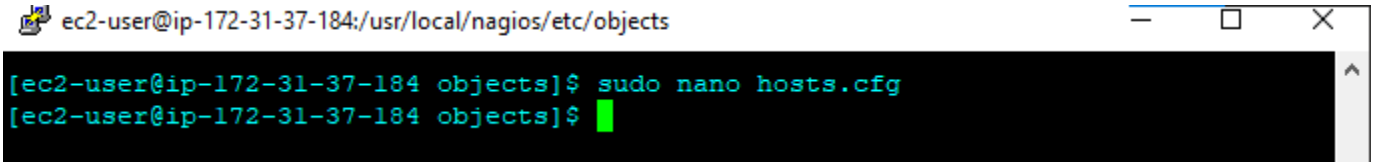
```
# Define a host for the local machine

define host{
        use                     linux-server            ; Name of host template to$
                                                        ; This host definition wil$
                                                        ; in (or inherited by) the$
        host_name               localhost
        alias                   localhost
        address                 127.0.0.1
        }
```

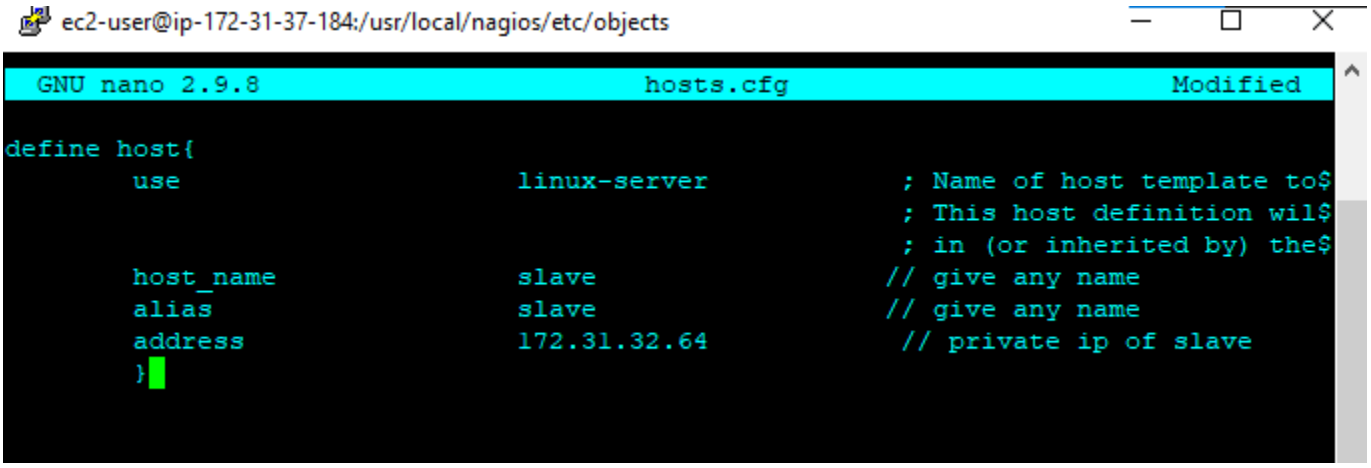
Configuration on Master

Create a new file hosts.cfg

sudo nano hosts.cfg



Content of the hosts file

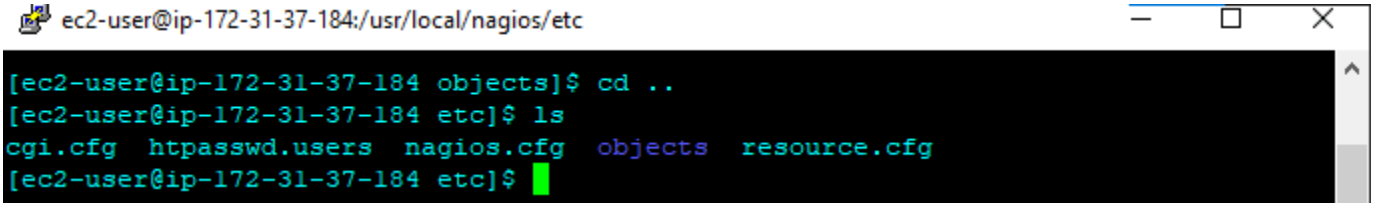


Configuration on Master

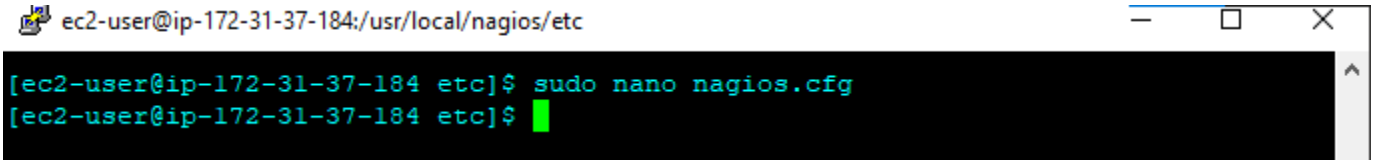
We need to provide the path of this file to the nagios.cfg file.

cd ..

ls



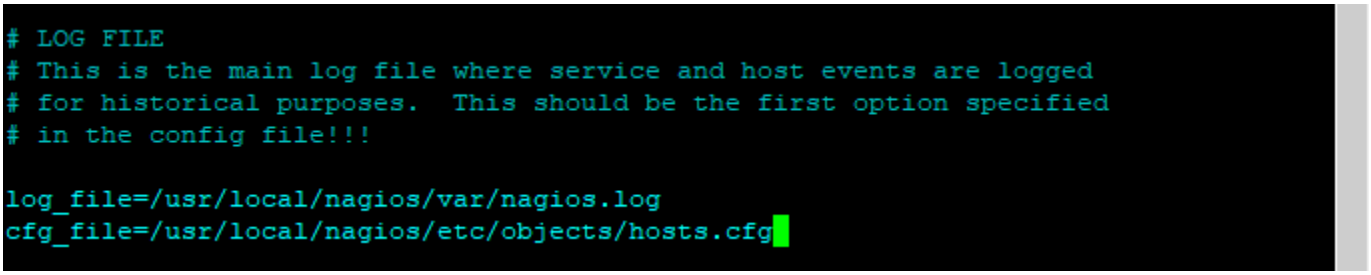
sudo nano nagios.cfg



Add the line

cfg_file=/usr/local/nagios/etc/objects/hosts.cfg

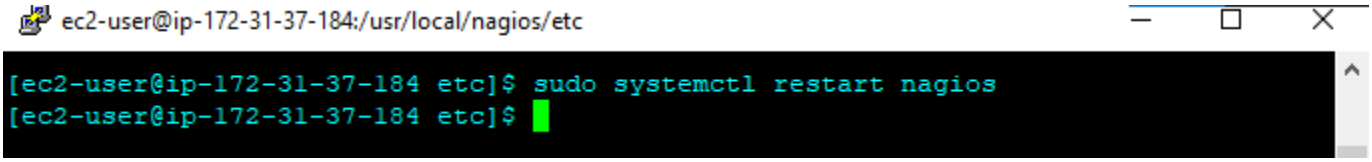
//Path of hosts.cfg file



Configuration on Master

Restart nagios to start with new configuration

sudo systemctl restart nagios



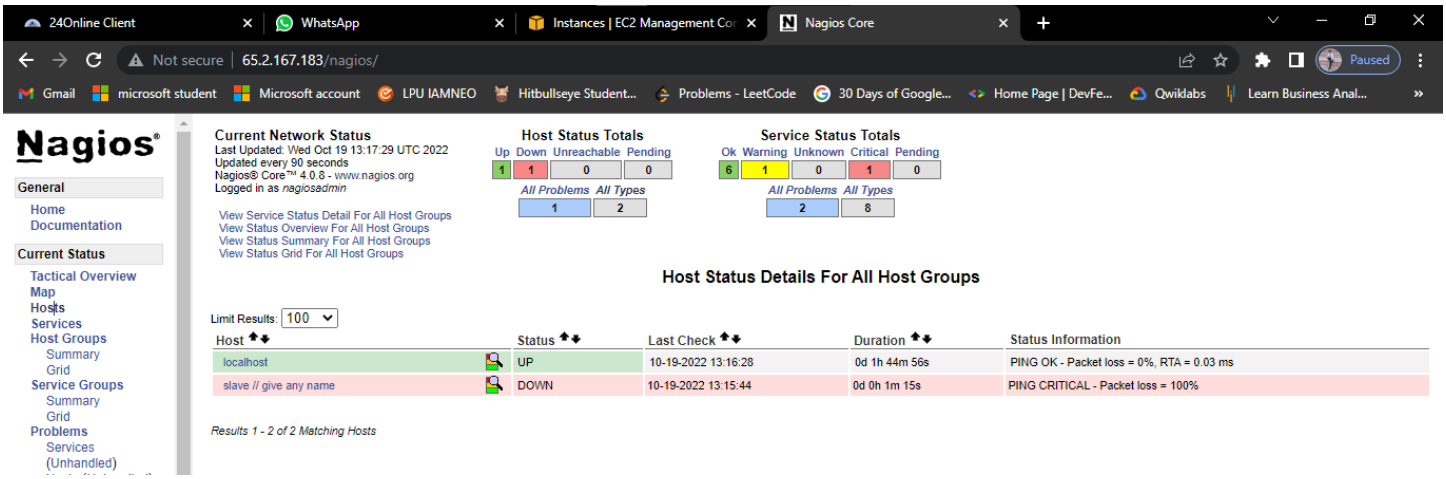
Check the hosts in Nagios interface

- Copy ip address of Master
- Login from browser with following

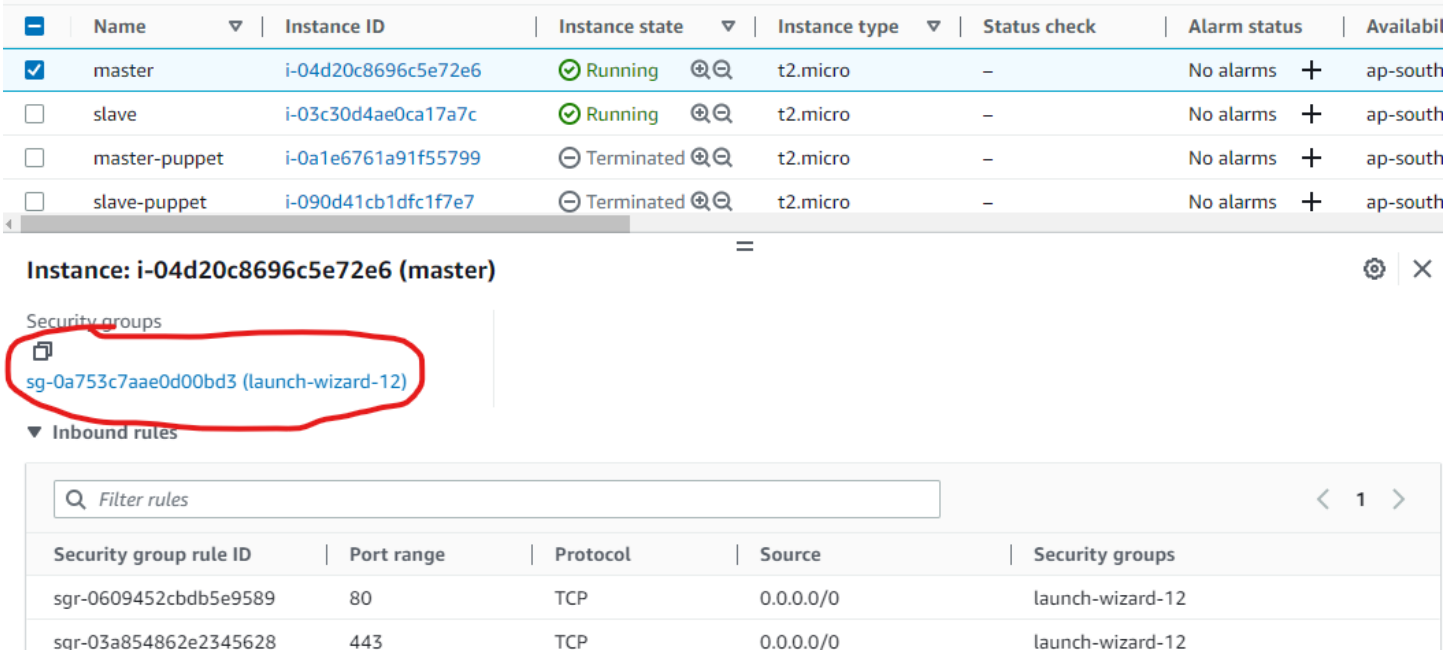
  <public ip of master>/nagios

- Provide username as **nagiosadmin** and password (provided earlier)
- Go to hosts and check if the new slave server is added.

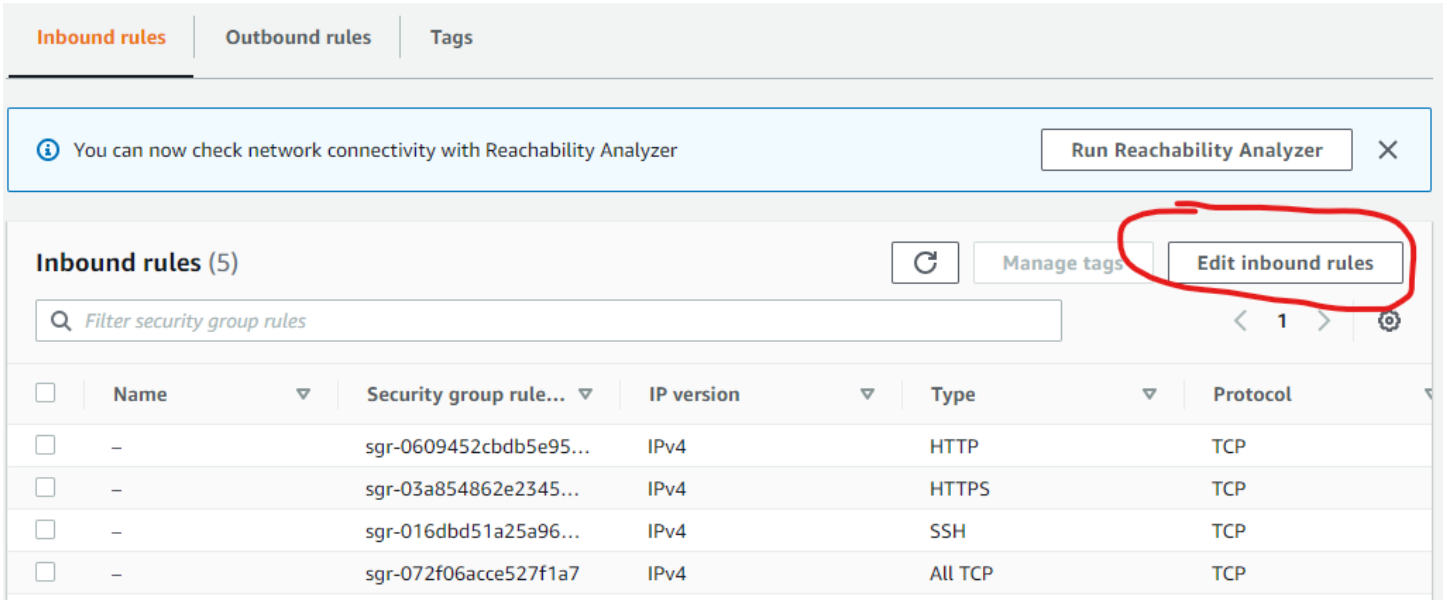
Check the host in the Nagios interface



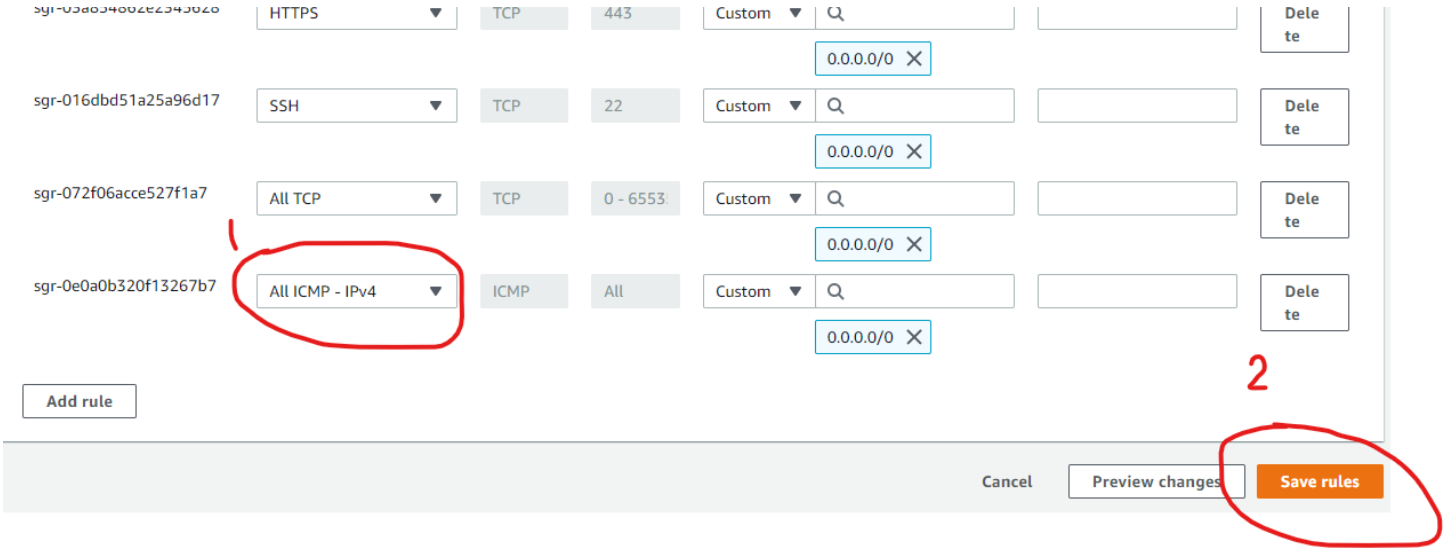
Add ICMP inbound rules to Master and Slave



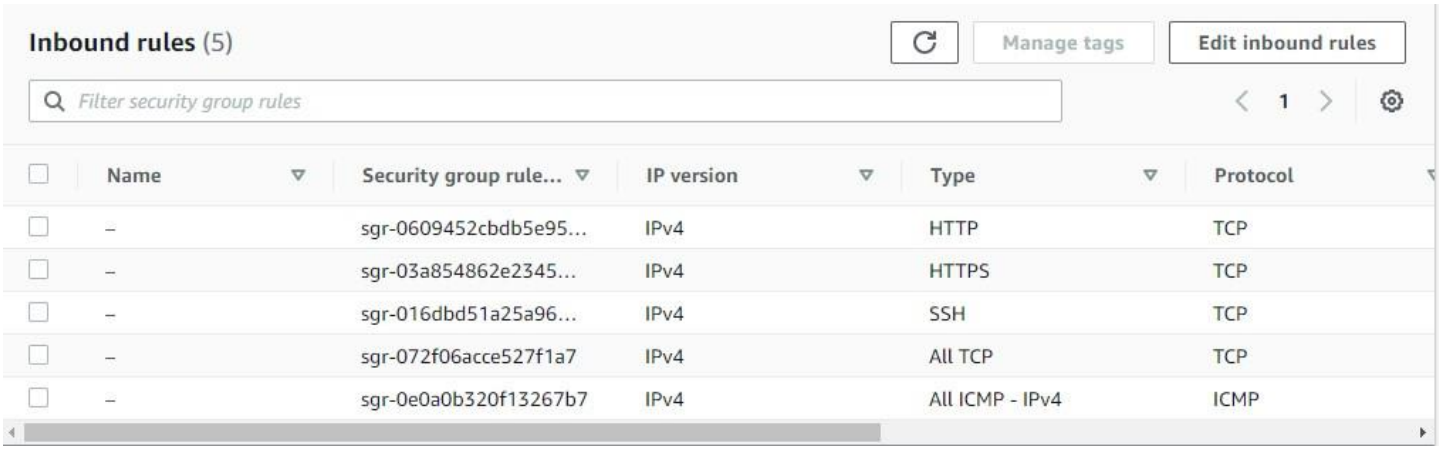
Edit inbound rules



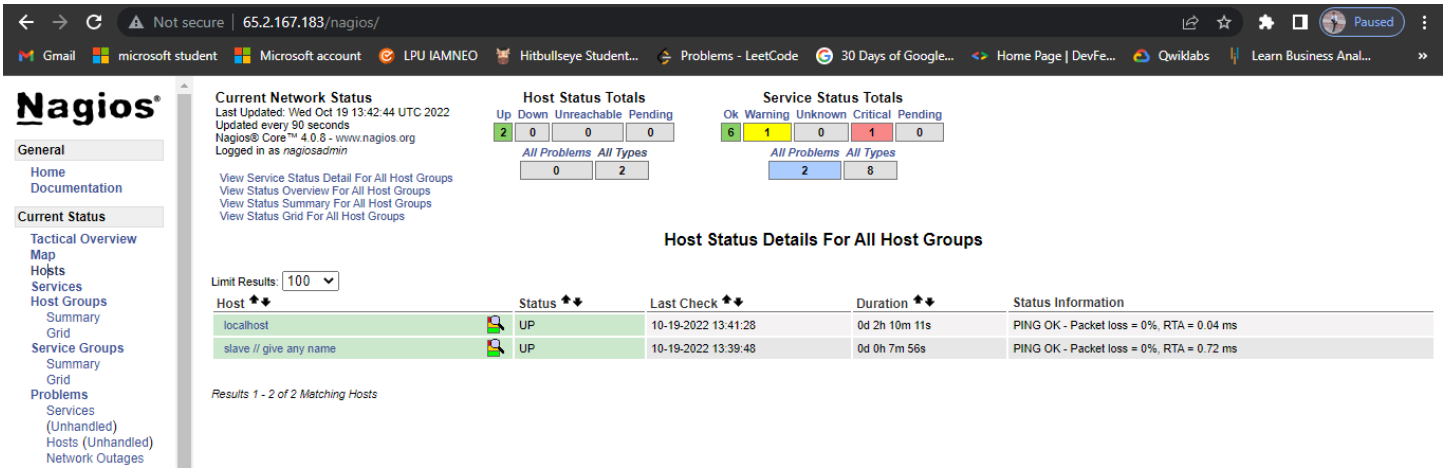
Add inbound rule of ICMP


HTTPS
TCP
443
Custom
0.0.0.0/0
Delete
sgr-016dbd51a25a96d17
SSH
TCP
22
Custom
0.0.0.0/0
Delete
sgr-072f06acce527f1a7
All TCP
TCP
0 - 6553
Custom
0.0.0.0/0
Delete
sgr-0e0a0b320f13267b7
All ICMP - IPv4
ICMP
All
Custom
0.0.0.0/0
Delete
2
Add rule
Cancel
Preview changes
Save rules

Added rule


Inbound rules (5)
Manage tags
Edit inbound rules
Filter security group rules
1
Name
Security group rule...
IP version
Type
Protocol
– sgr-0609452cbdb5e95... IPv4 HTTP TCP
– sgr-03a854862e2345... IPv4 HTTPS TCP
– sgr-016dbd51a25a96... IPv4 SSH TCP
– sgr-072f06acce527f1a7 IPv4 All TCP TCP
– sgr-0e0a0b320f13267b7 IPv4 All ICMP - IPv4 ICMP

Check the hosts from Nagios Master interface

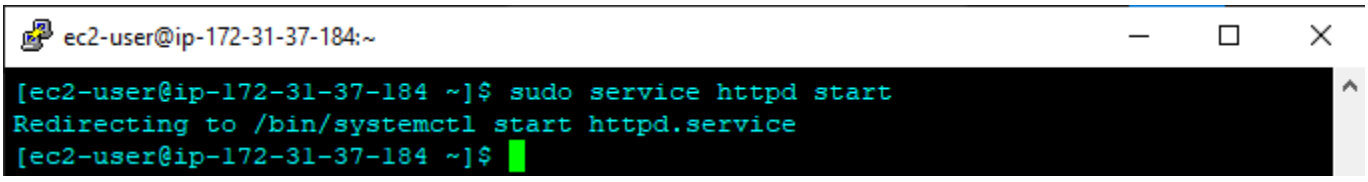


Nagios Adding Service

Start Apache and test it from Master

Start httpd

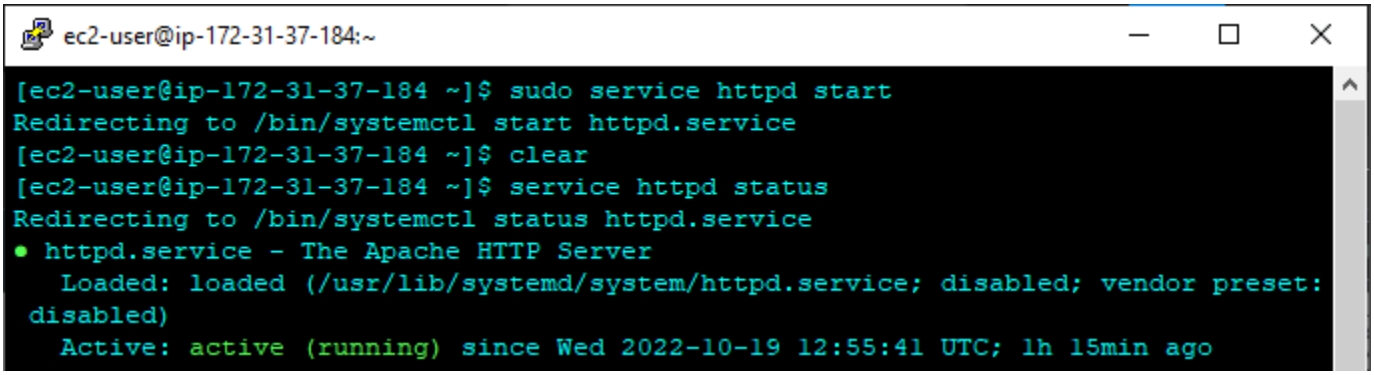
sudo service httpd start



Test if Apache is running

Service httpd status

Check from browser



Start the Nagios service in Master

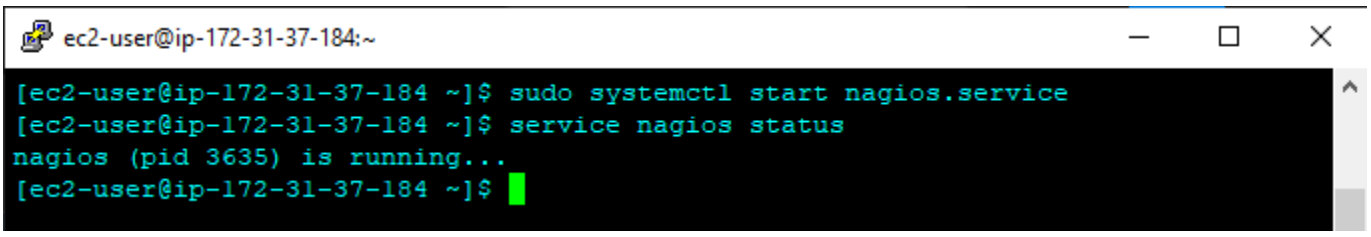
Start Nagios

sudo systemctl start nagios.service

Check if it is running

service nagios status

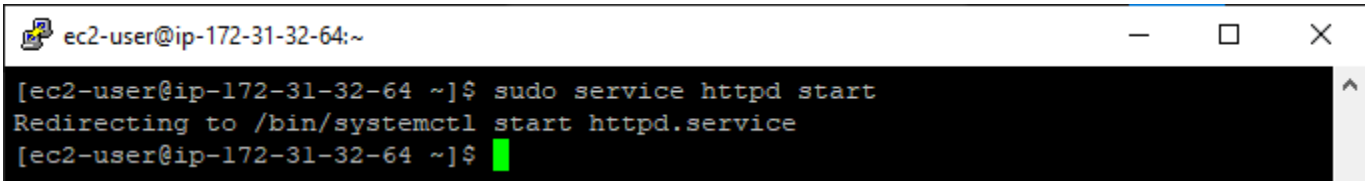
Check from browser



Start Apache and test it from Slave

Start httpd

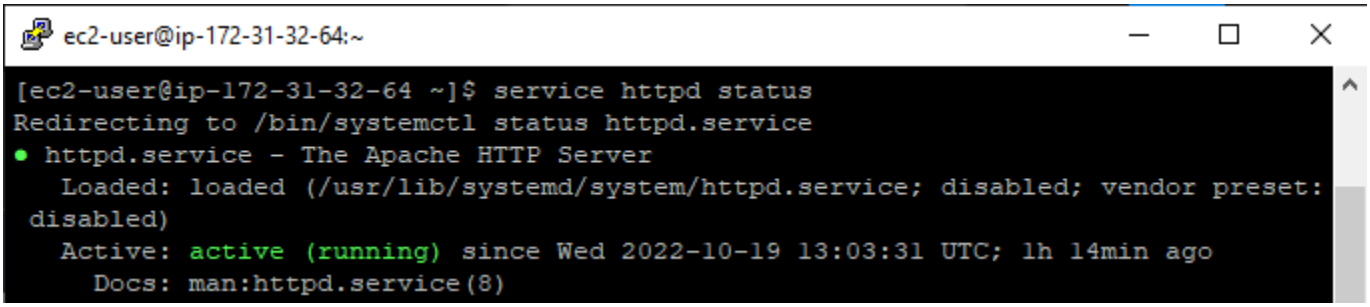
sudo service httpd start



Test if Apache running

service httpd status

Check from browser



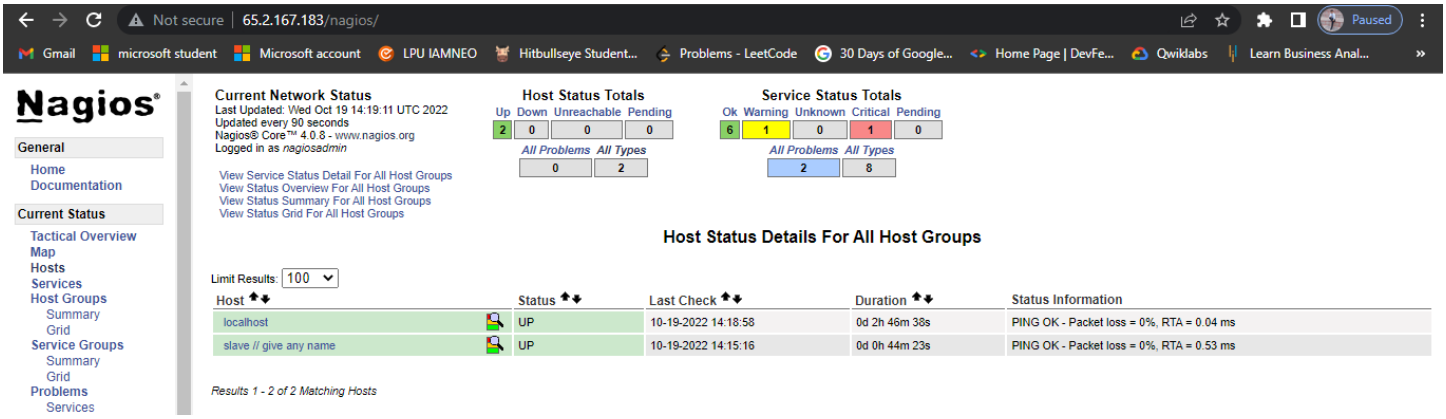
Check the hosts in the Nagios interface

- Copy ip address of Master
- Login from browser with following

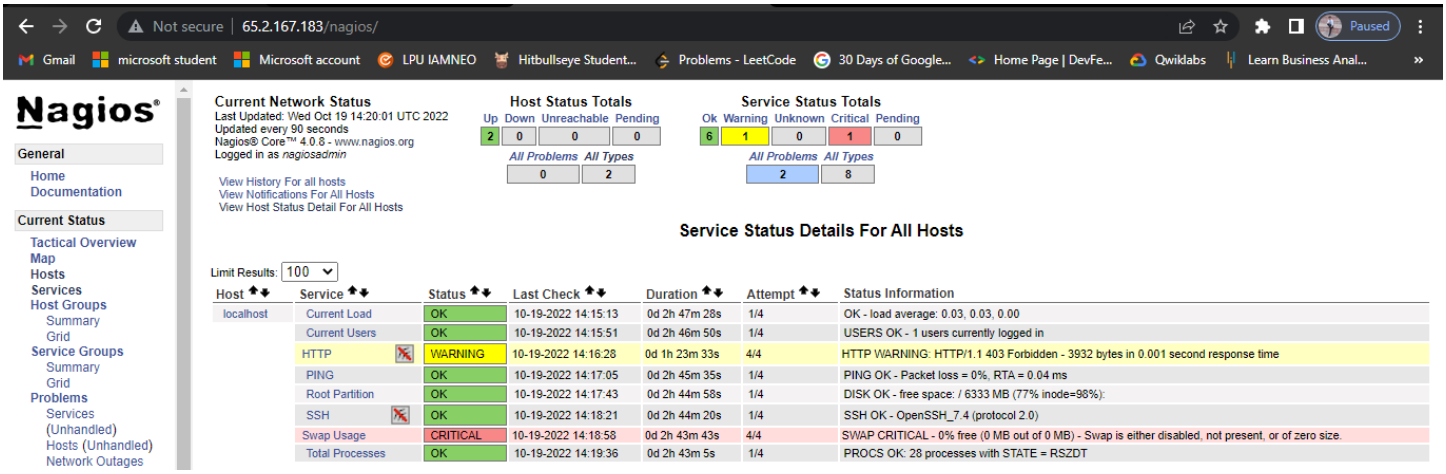
  <public ip of master>/nagios

- Provide username as **nagiosadmin** and password (provided earlier)
- Go to hosts and check if the new slave server is added.

Check the hosts in the Nagios interface



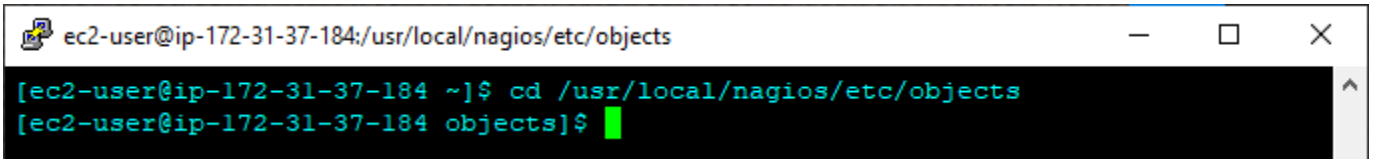
Check the service in the Nagios interface



Configuration on Master

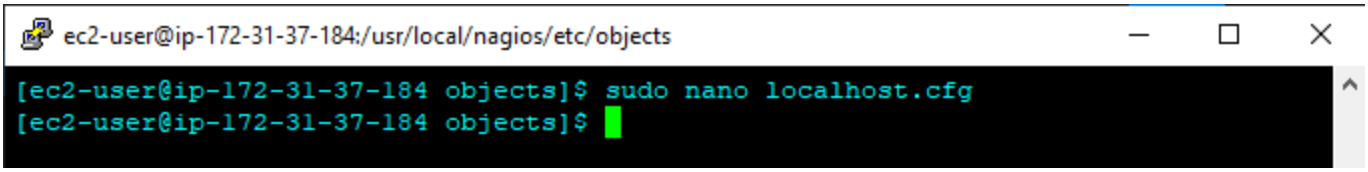
Move to objects folder

cd /usr/local/nagios/etc/objects



Check localhost.cfg

sudo nano localhost.cfg



Check services definition

Copy the services definitions

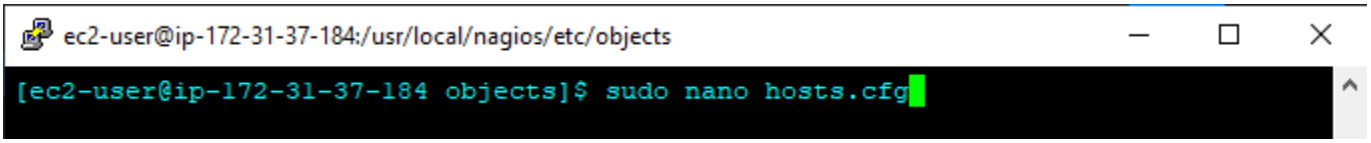
```
# Define a service to "ping" the local machine

define service{
        use                             local-service         ; Name of service t$
        host_name                       localhost
        service_description             PING
        check_command                   check_ping!100.0,20%!500.0,60%
        }
```

Configuration on Master

Open hosts.cfg

sudo nano hosts.cfg



Contents of the hosts file

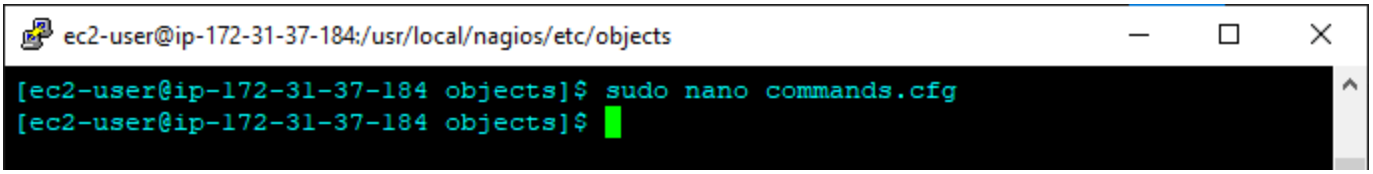
```
# Define a service to "ping" the local machine

define service{
        use                             local-service         ; Name of service t$
        host_name                       localhost
        service_description             PING
        check_command                   check_ping!100.0,20%!500.0,60%
        }
```

Configuration on Master

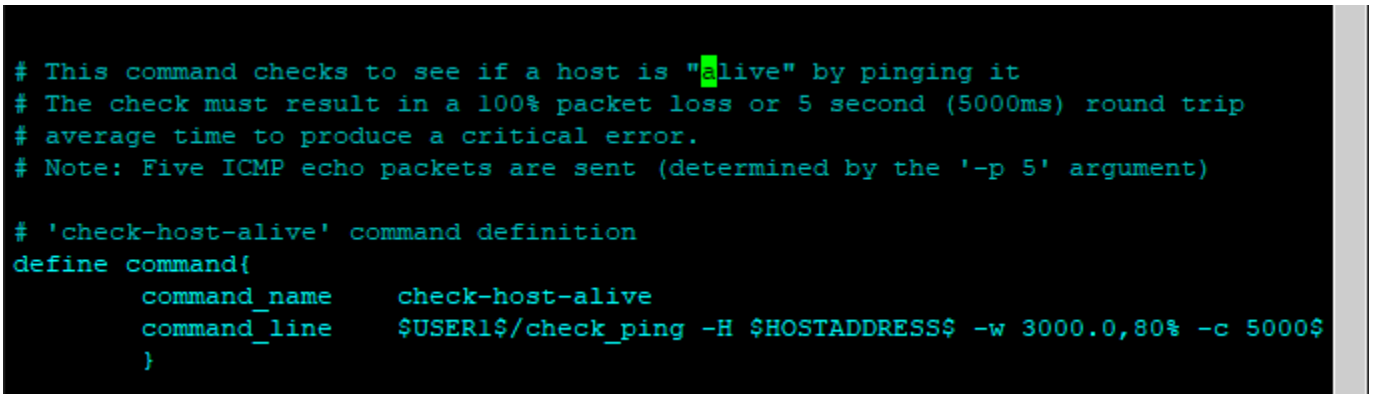
Goto command.cfg

sudo nano commands.cfg



Search for host alive service

CTRL+w and write alive and enter



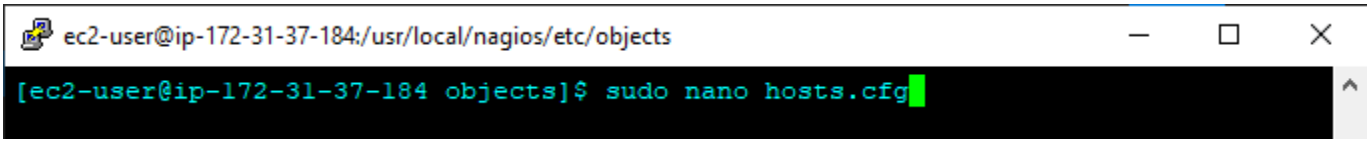
Check the content and copy line

command_name check-host-alive

Configuration on Master

Open hosts.cfg

sudo nano hosts.cfg



Edit the file

```
ec2-user@ip-172-31-37-184:/usr/local/nagios/etc/objects
  GNU nano 2.9.8                          hosts.cfg

define host{
        use                     linux-server            ; Name of host template t$
                                                        ; This host definition wi$
                                                        ; in (or inherited by) th$
        host_name               slave
        alias                   slave
        address                 172.31.32.64
        }
define service{
        use                             generic-service         ; Name of service$
        host_name                       slave
        service_description             check host alive
        check_command                   check-host-alive
        check_interval                  1
        retry_interval                  1
}
```

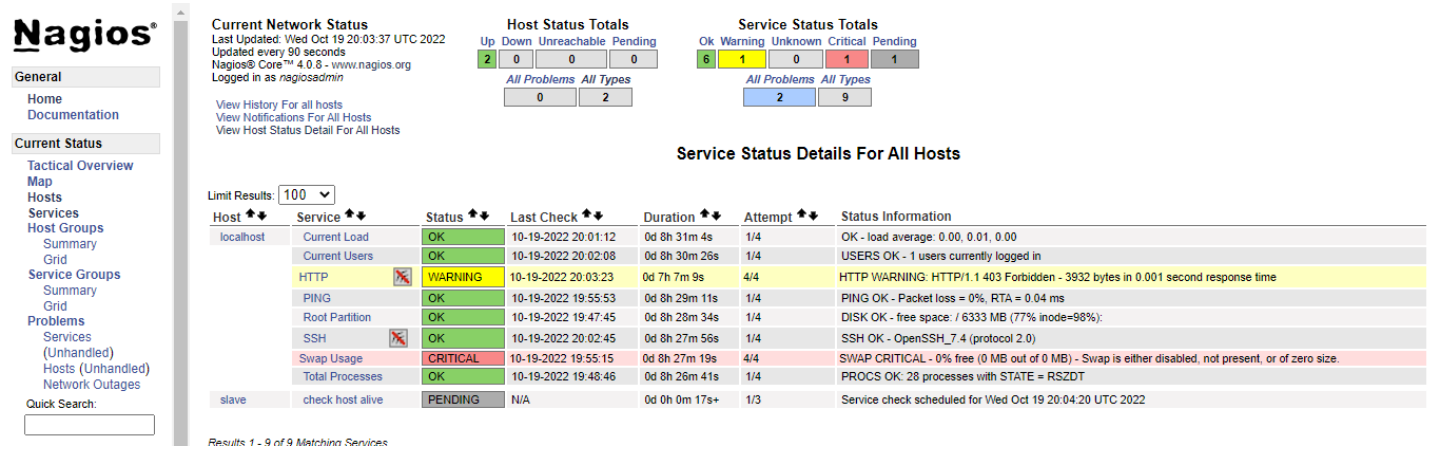
Configuration on Master

Restart nagios to start with new configuration

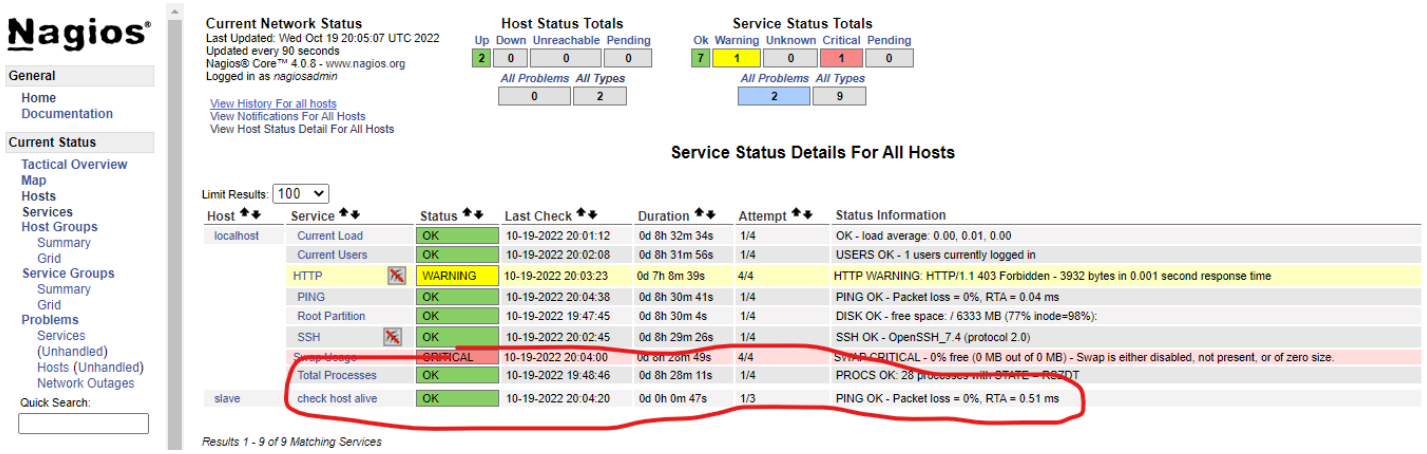
sudo systemctl restart nagios

```
ec2-user@ip-172-31-37-184:/usr/local/nagios/etc/objects
[ec2-user@ip-172-31-37-184 objects]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-37-184 objects]$
```

Check services in nagios interface

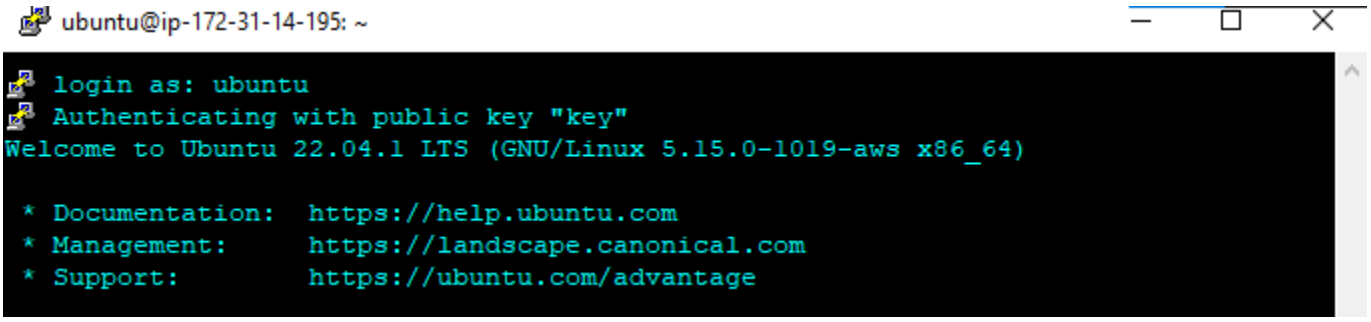


Check services in nagios interface



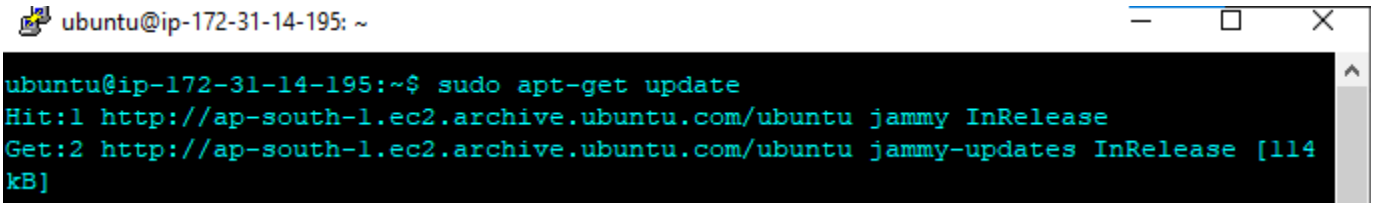
Puppet configuration

Master Configuration

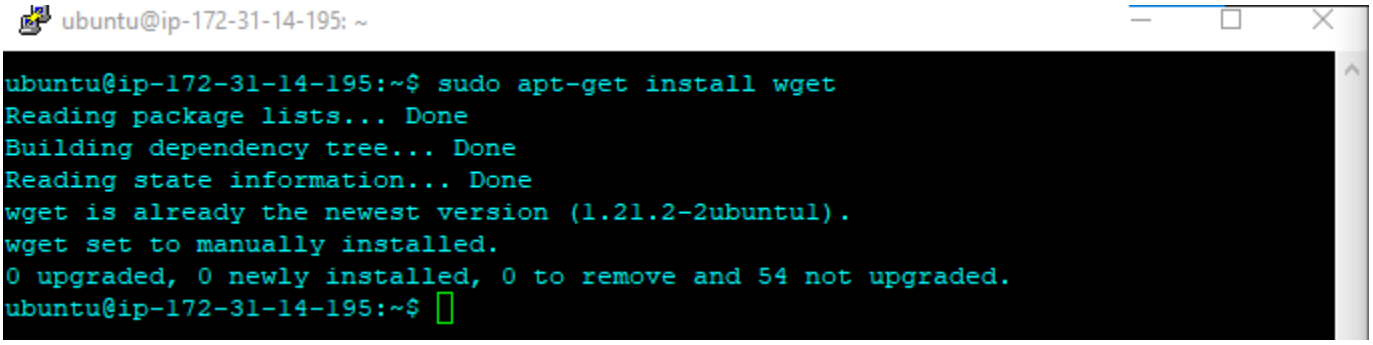


Update servers

sudo apt-get update

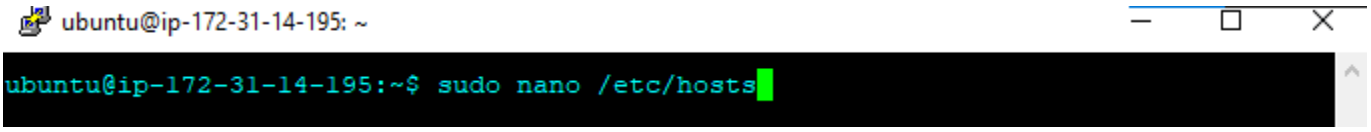


sudo apt-get install wget

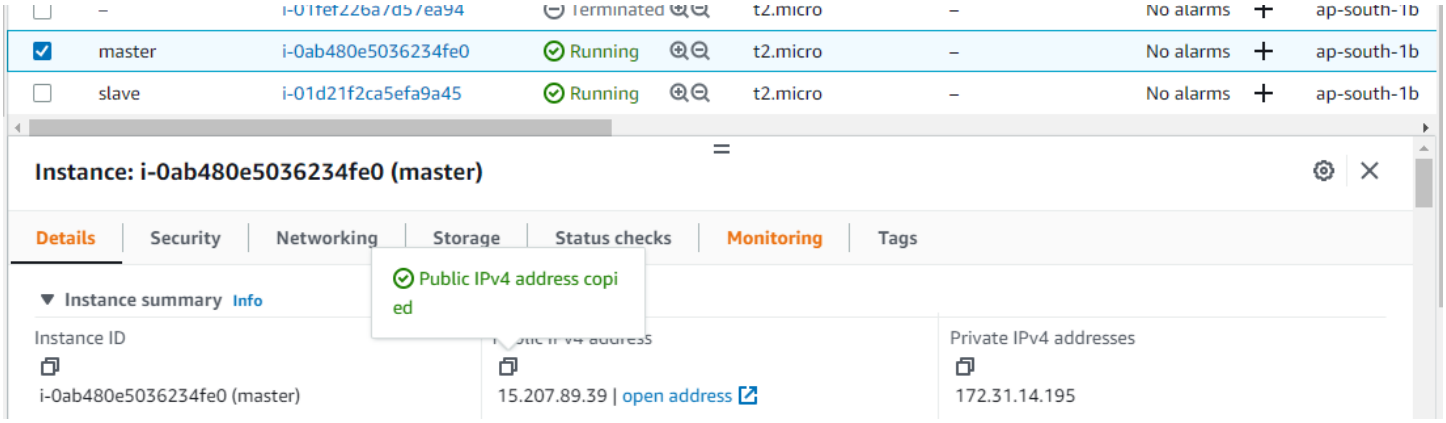


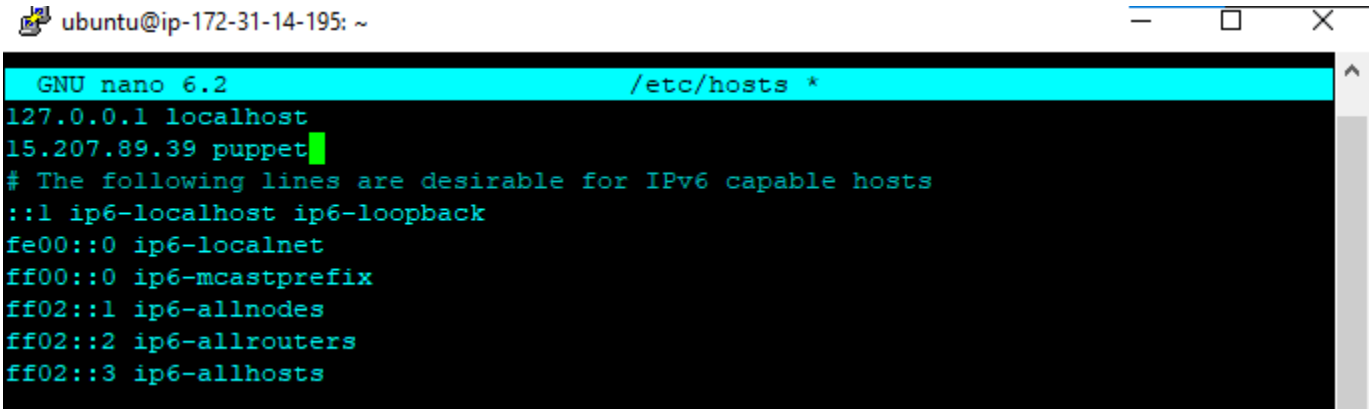
change the host

sudo nano /etc/hosts



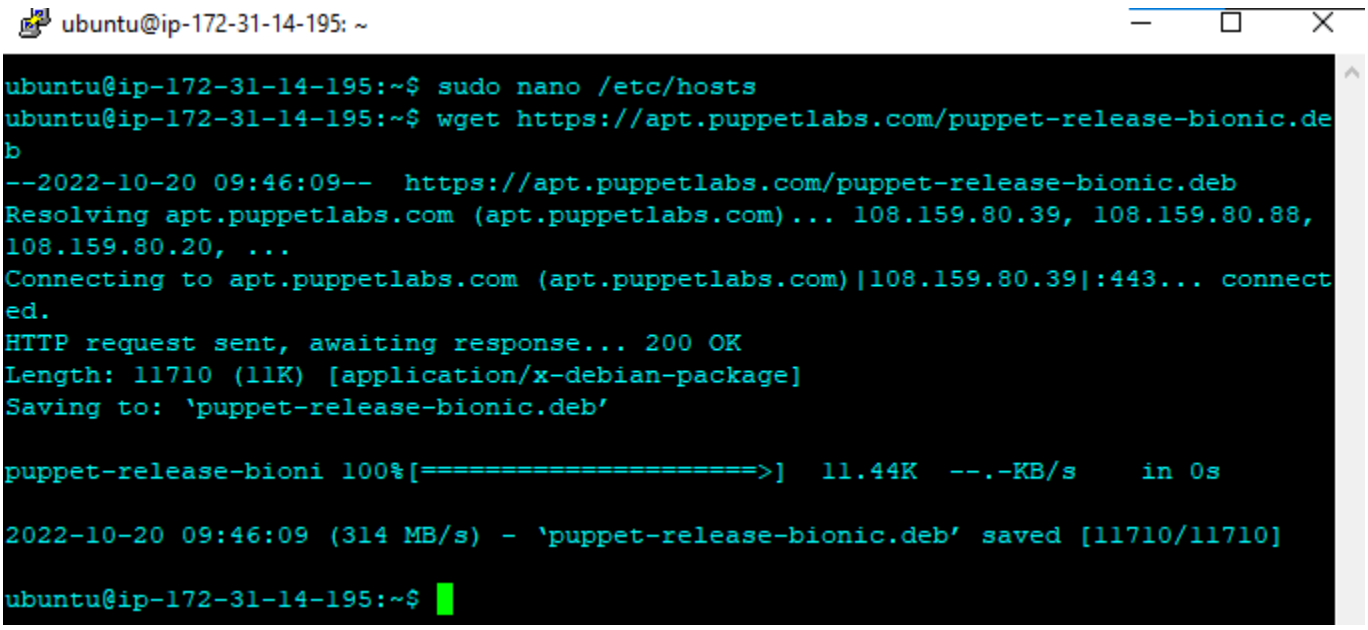
<public ip of master> puppet





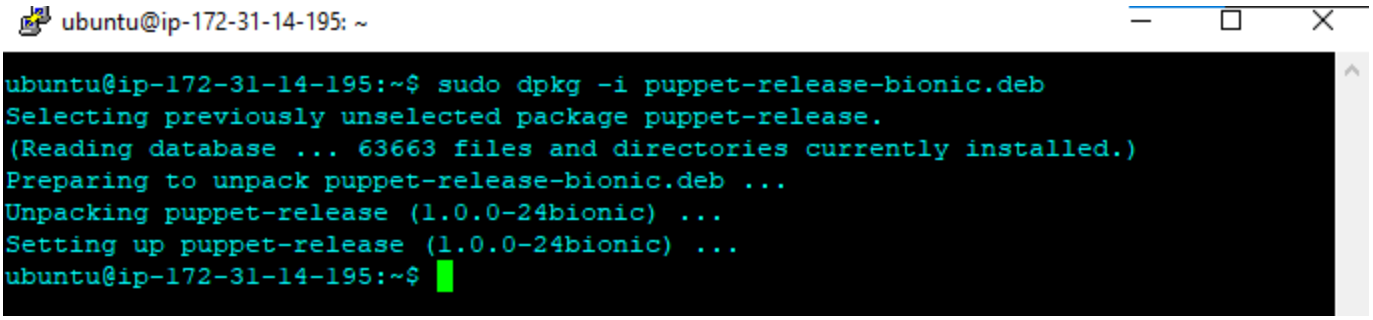
Download files (Debian package)

wget https://apt.puppetlabs.com/puppet-release-bionic.deb



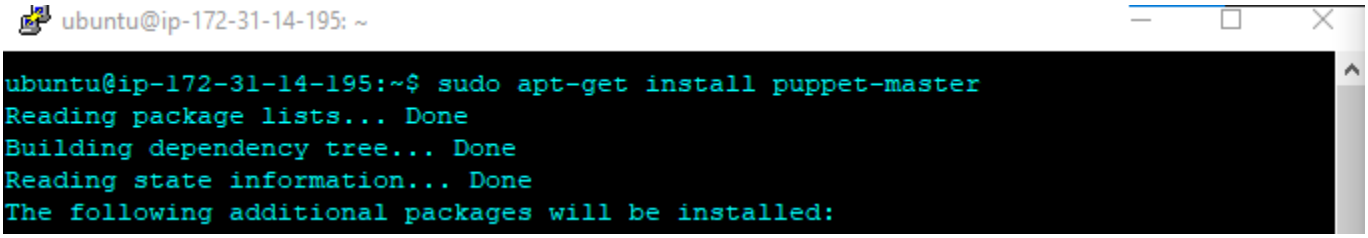
Unzip the file

sudo dpkg -i puppet-release-bionic.deb



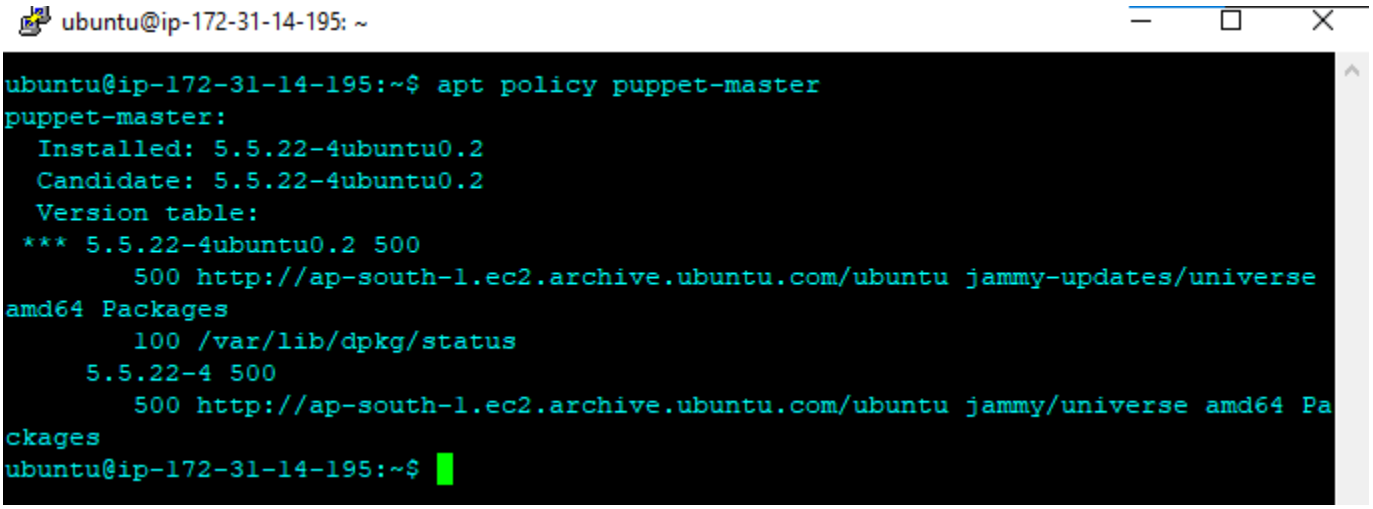
Install puppet master

sudo apt-get install puppet-master



Check puppet master policy

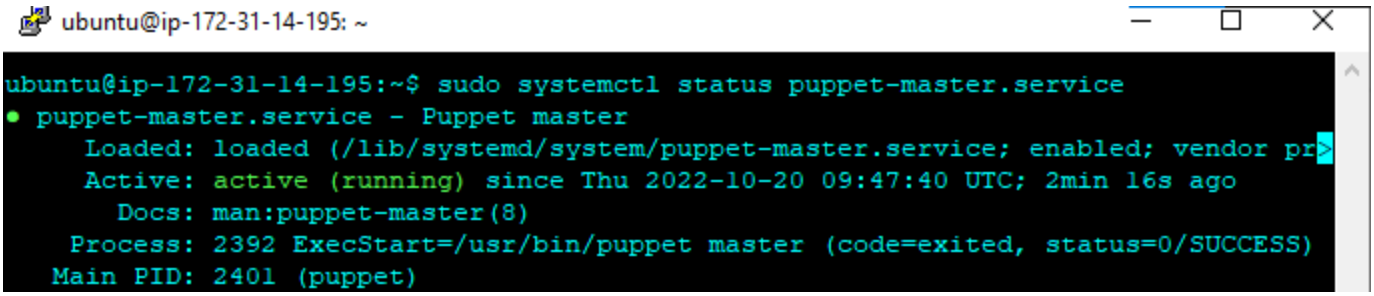
apt policy puppet-master



Check the puppet master active status

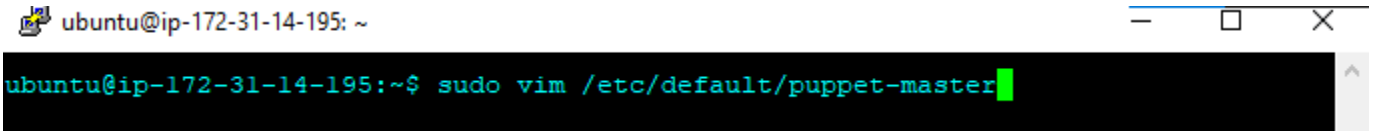
sudo systemctl status puppet-master.service

ctrl+c



Goto file to change args

sudo vim /etc/default/puppet-master



Change java args for priamry memory

Xmx specifies the maximum memory allocation pool for a Java virtual machine

(JVM), while Xms specifies the initial memory allocation pool

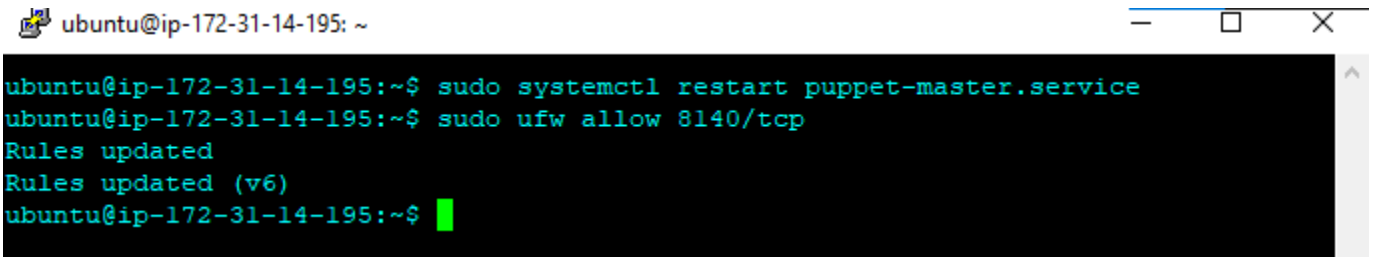
JAVA_ARGS="-Xms512m -Xmx512m"

Restart Puppet master

sudo systemctl restart puppet-master.service

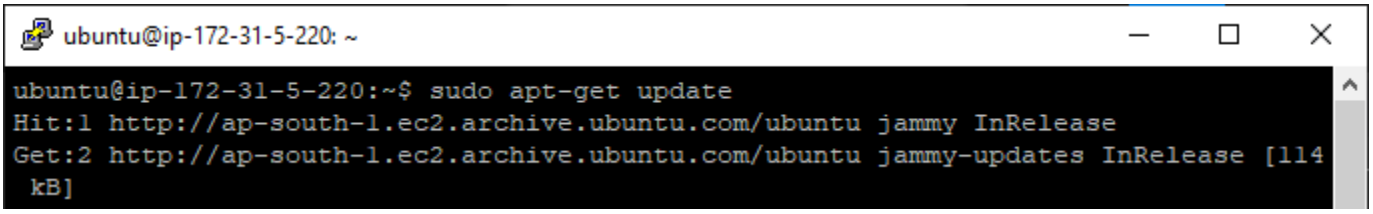
Open port to communicate with slave

sudo ufw allow 8140/tcp

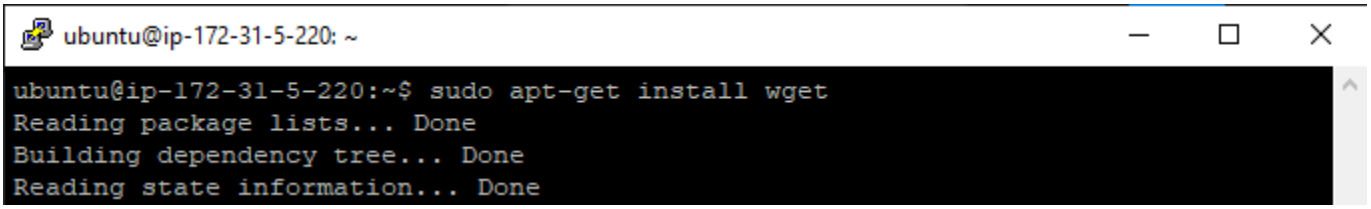


Slave Configuration

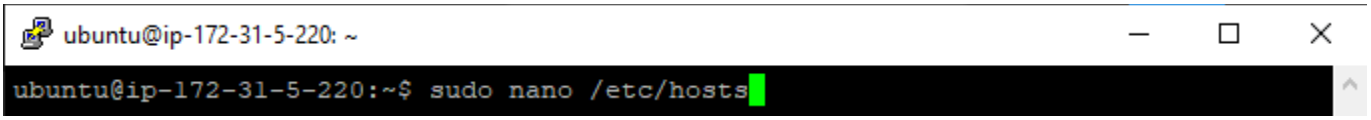
sudo apt-get update



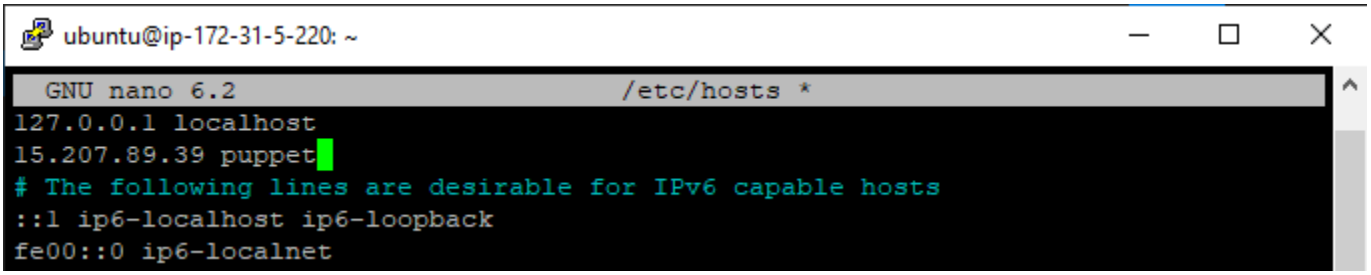
sudo apt-get install wget



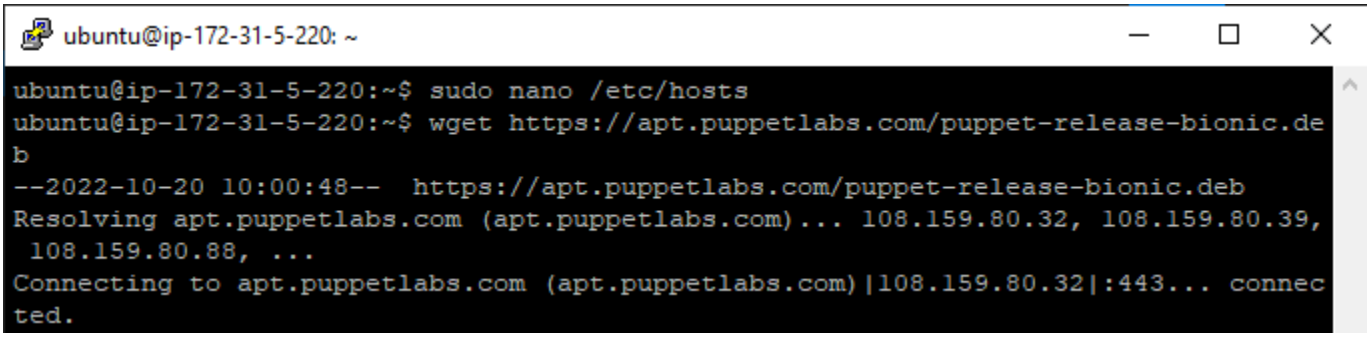
sudo nano /etc/hosts



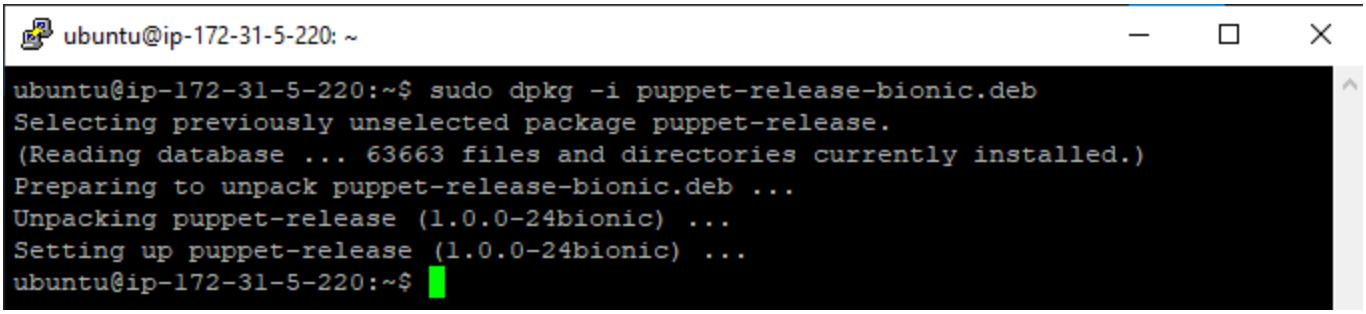
<public ip of master> puppet



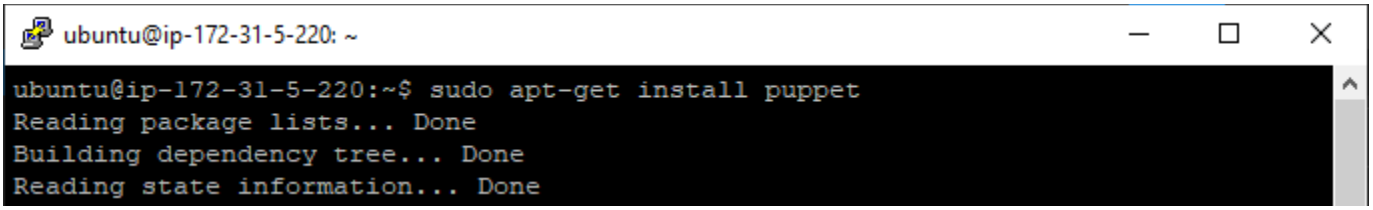
wget https://apt.puppetlabs.com/puppet-release-bionic.deb



sudo dpkg -i puppet-release-bionic.deb

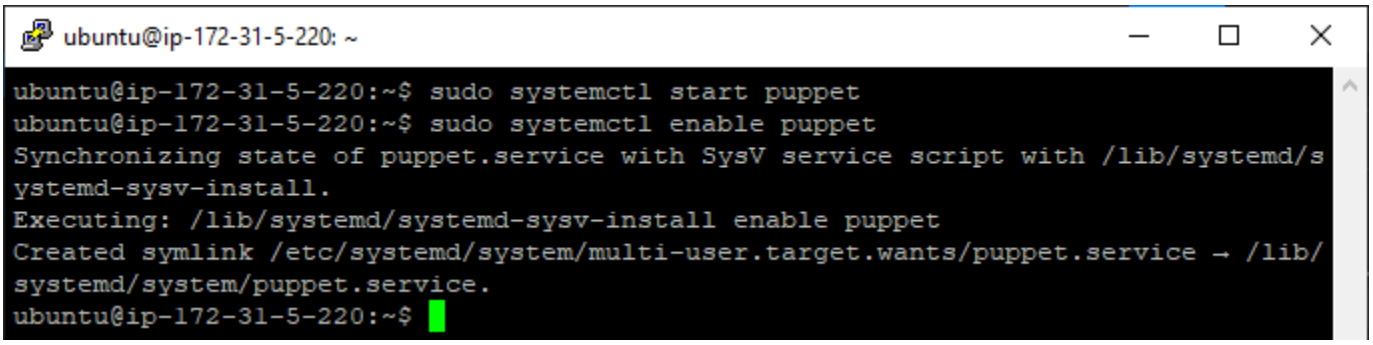


sudo apt-get install puppet



sudo systemctl start puppet

sudo systemctl enable puppet

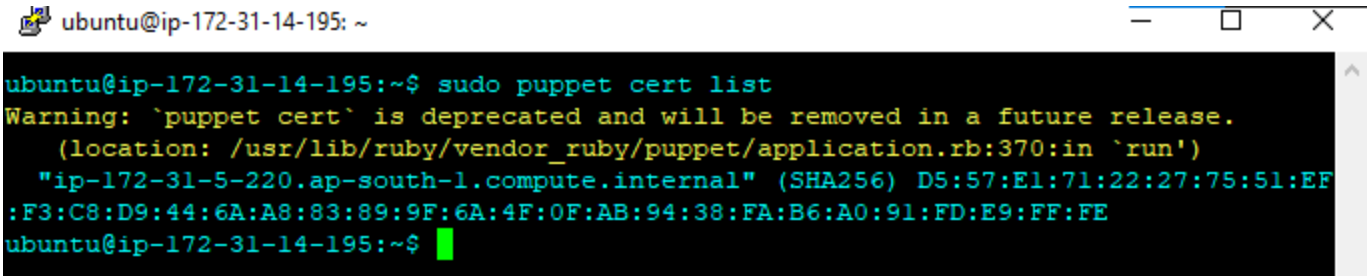


Configuration of the slave is done.

Master

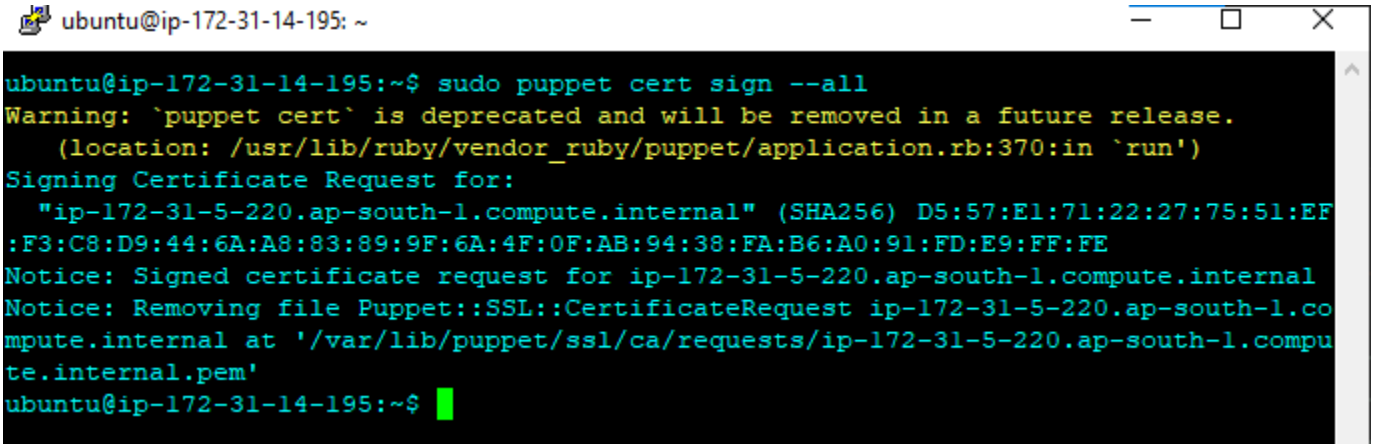
Need to check the certificate request

sudo puppet cert list



If there is any certificate then sign them

sudo puppet cert sign –all



Create Manifests

sudo mkdir -p /etc/puppet/code/environments/production/manifests/

sudo nano /etc/puppet/code/environments/production/manifests/site.pp

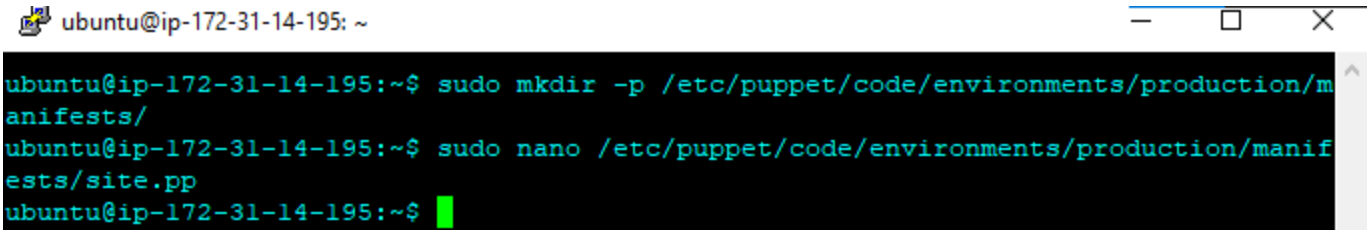
file{'/tmp/puppet_test.txt': #resource type file and filename

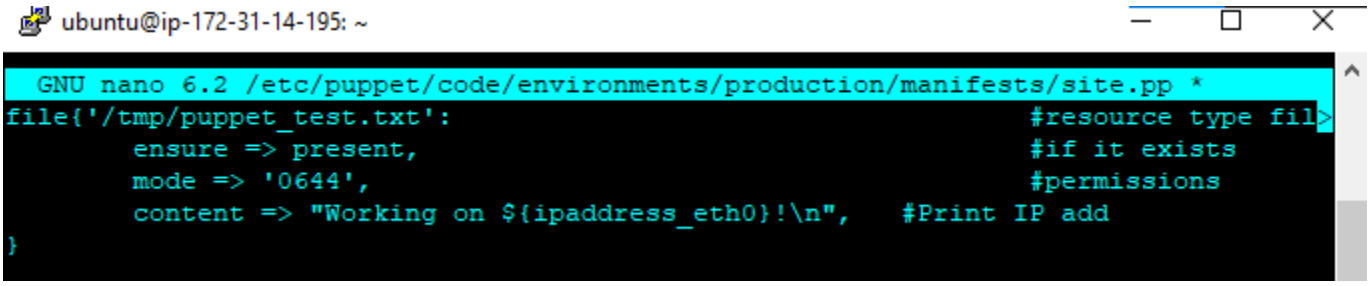
ensure => present, #if it exists

mode => '0644', #permissions

content => "Working on ${ipaddress_eth0}!\n", #Print IP add

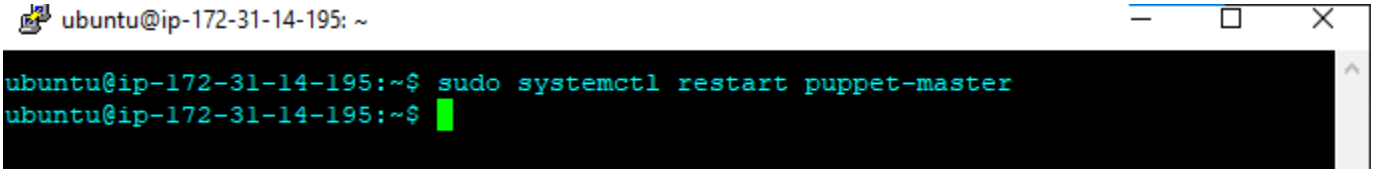
}





Restart puppet master

sudo systemctl restart puppet-master

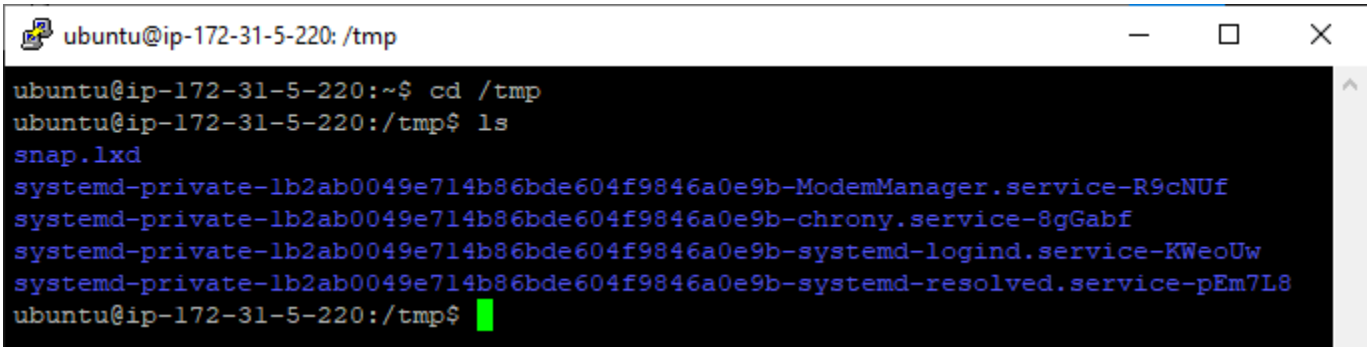


Slave

Check if any file exist on in temp starts with i

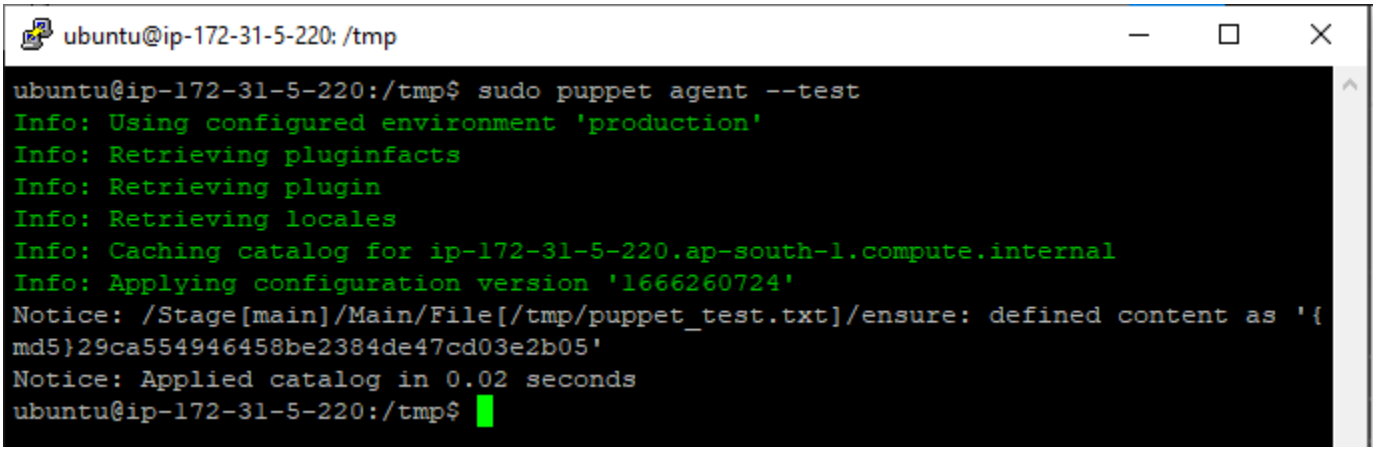
cd /tmp

ls



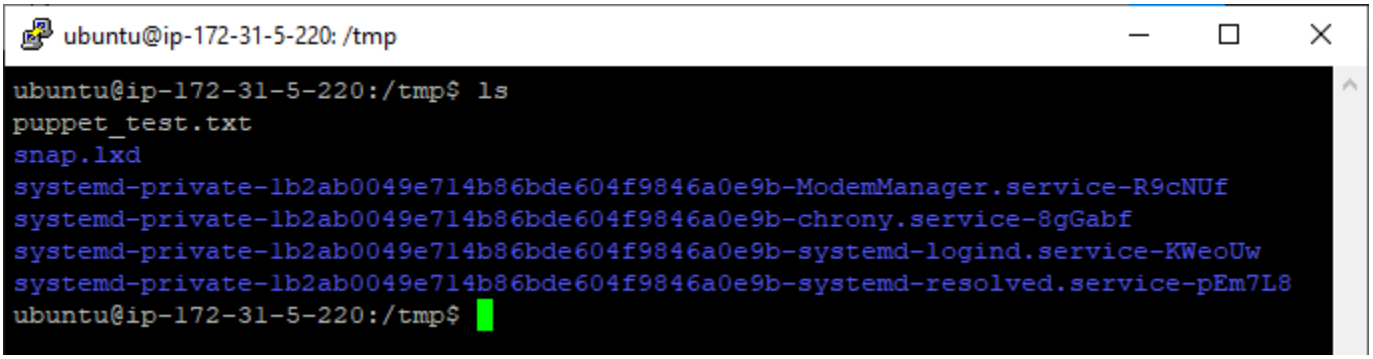
Slave asked the master if there is any changes required for this server

sudo puppet agent –test



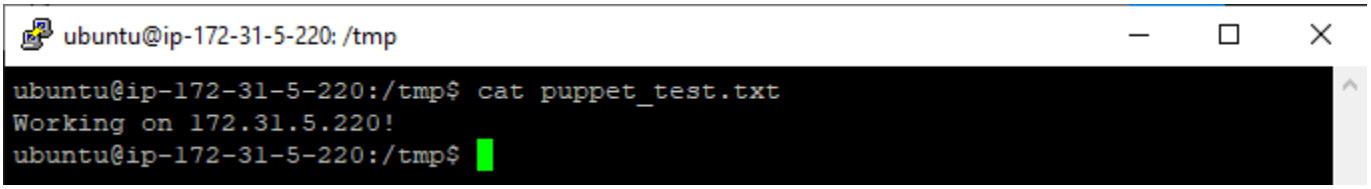
Check if any file is created

ls



Check the contents

cat puppet_test.txt



Configuration of master-slave communication is over.

Master

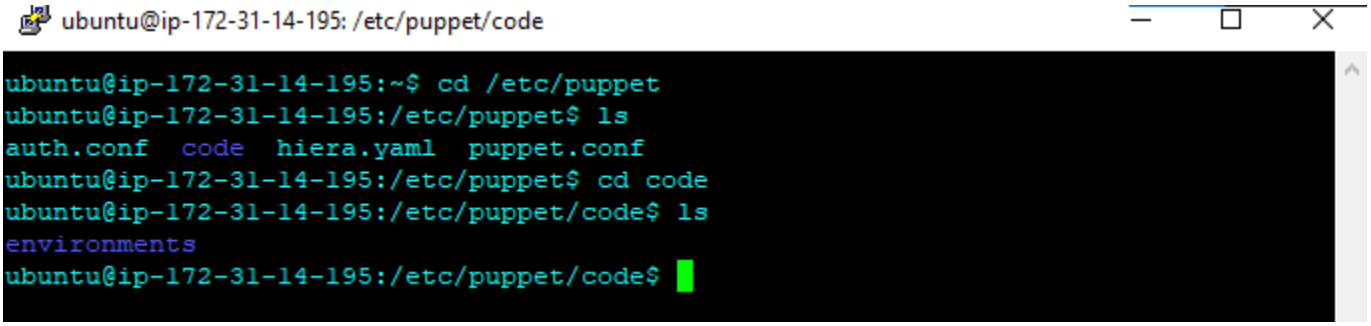
Create Manifests

cd /etc/puppet

ls

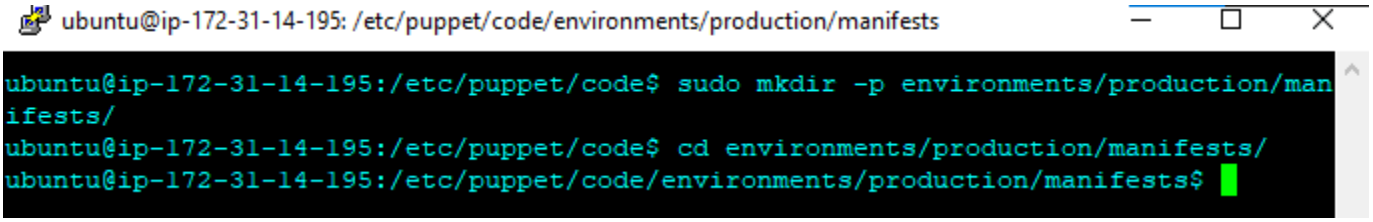
cd code

ls



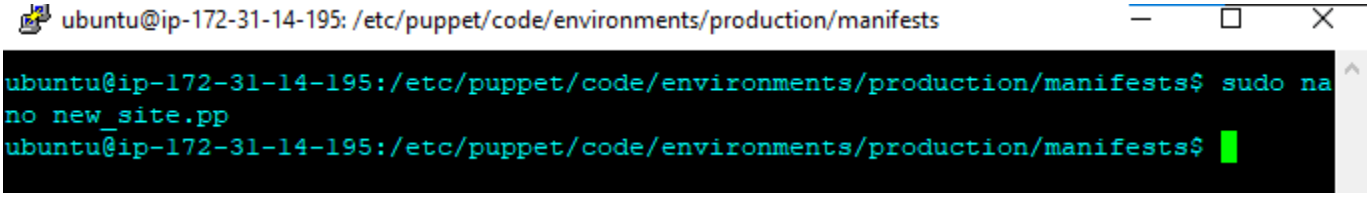
sudo mkdir -p environments/production/manifests/

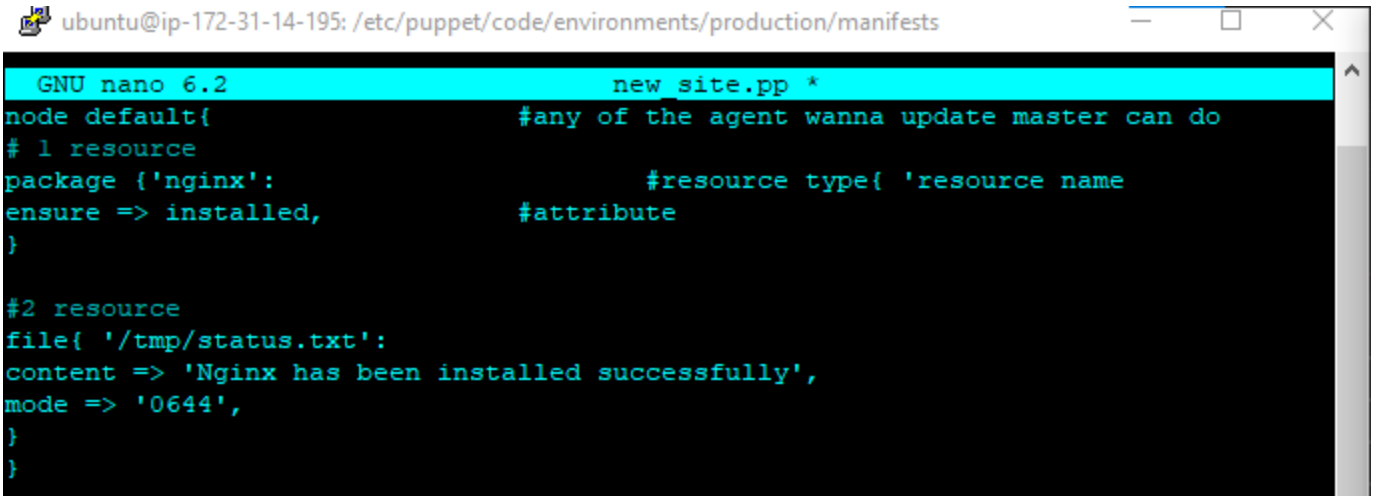
cd environments/production/manifests/



Create a file new_site1.pp

sudo nano new_site.pp





Save the file

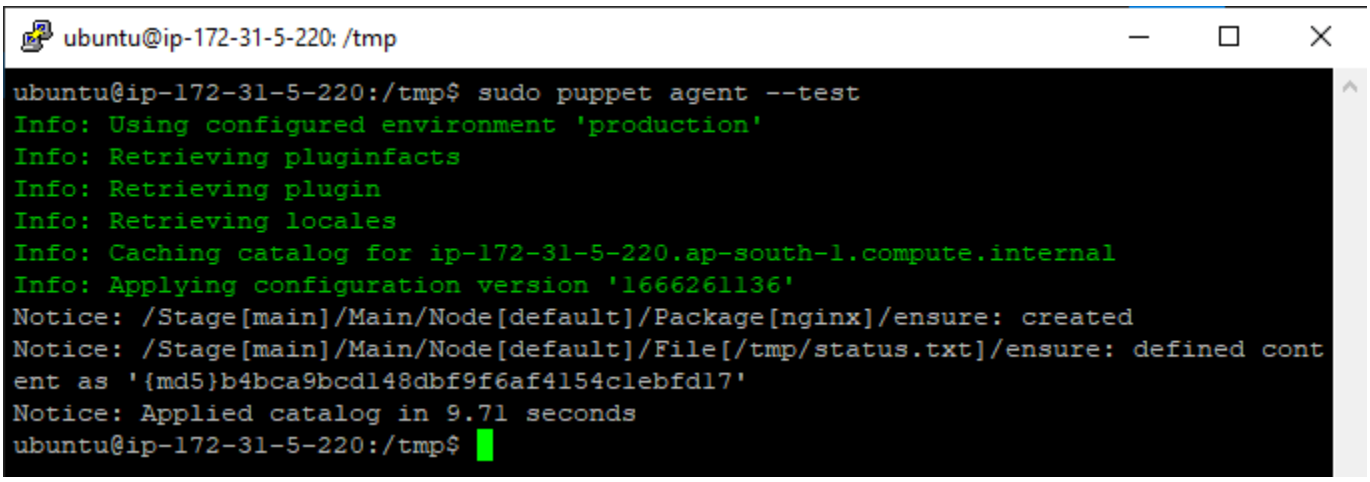
ctrl+x

y

Enter

Slave

sudo puppet agent –test

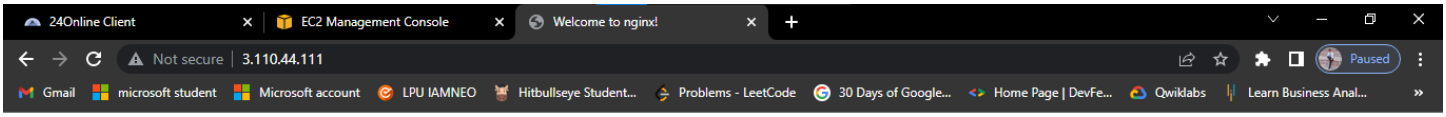


Copy public ip of slave

Open browser

Paste ip hit enter

Nginx website is visible



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.