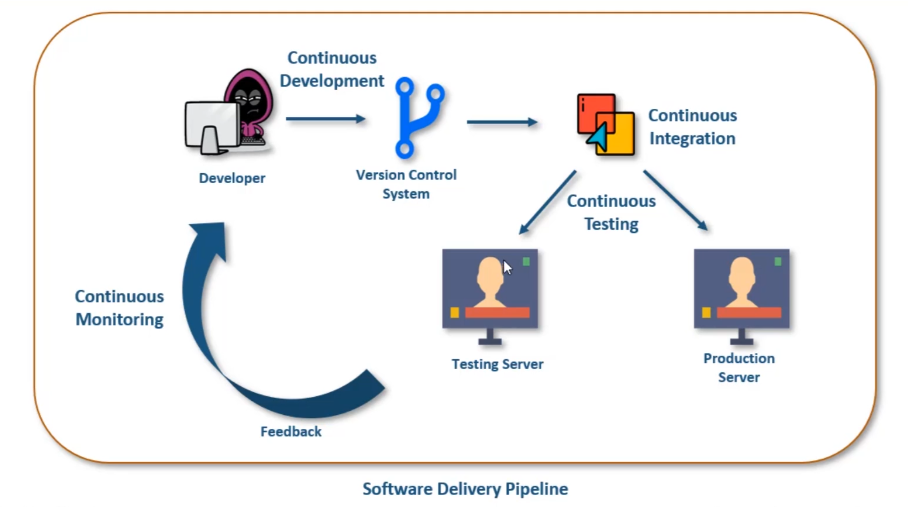
Continuous Monitoring:



CM tools:

1. Nagios
2. ELK Stack
3. Splunk
4. etc..

Nagios:

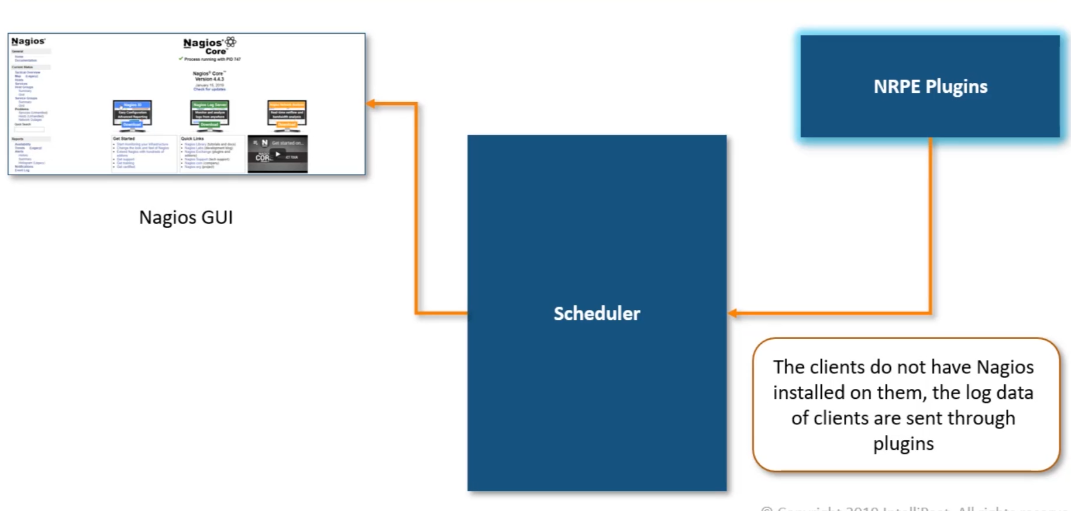
Nagios is open source software which is used to monitor systems, network and infrastructure. Also it will follow master slave architecture. Nagios offers monitoring and alerting services for servers, switches and applications.

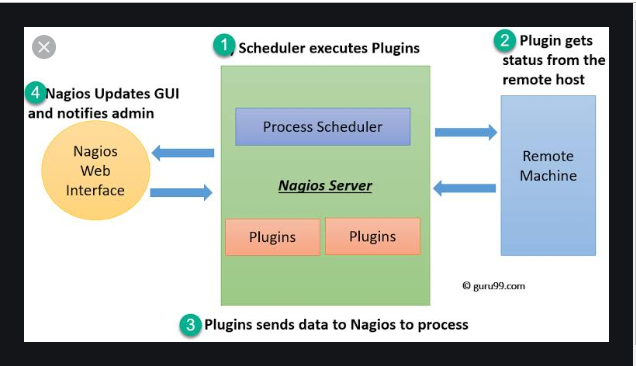
3 main components:

Nagios GUI: It is like dashboard which is used to monitor the services of servers and also check which all hosts are monitored. Also status and logs.

Scheduler: All the work of scheduling. Like what to check and when to check.

NRPE Plugins: The clients do not have nagios installed on them, the log data of clients are sent through plugins.



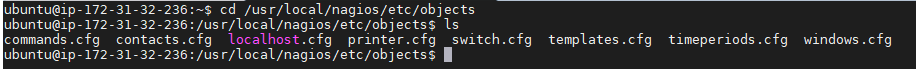


Refer below URL for Nagios setup which works perfectly.

<https://blog.serverdensity.com/howto-install-nagios-in-30-minutes-and-jumpstart-your-monitoring/>

Nagios Components:

Objects: These are configuration files which acts as building blocks of nagios which contributes for CM. These are elements that are involved in the monitoring and notification logic. When nagios starts or reloads it reads all .cfg files within the object directories.

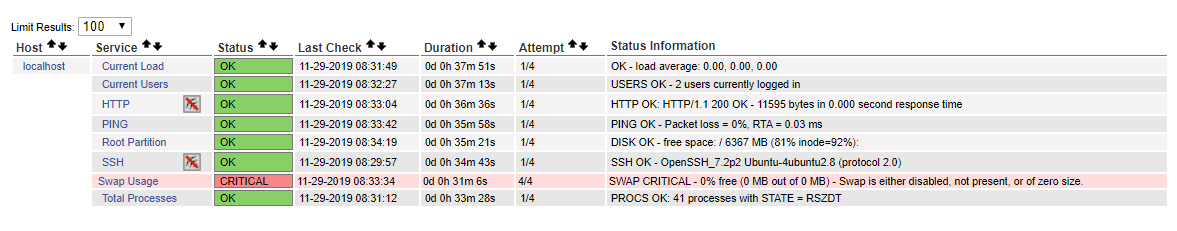


commands.cfg is used to define own command and also have internal commands which nagios defined and can be used. These commands basically runs on client. So it creates a log and this log is sent back to nagios server to analyse.

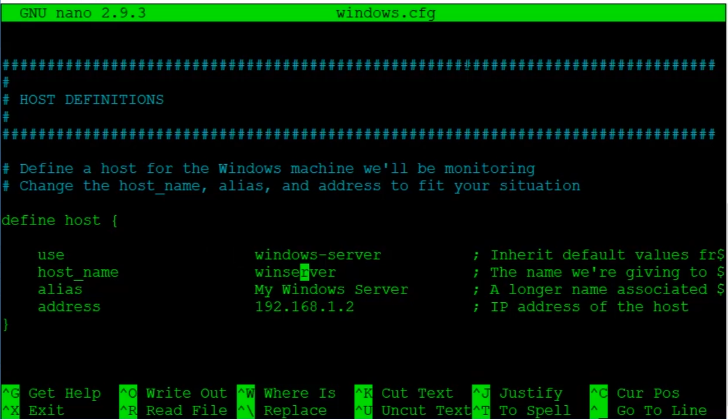
localhost.cfg: All the services like host, hostgroup, service definitions like (ping, root partition, current users) that can be monitored by itself are defined in localhost.cfg. As you know Nagios can monitor itself. All these service are defines in services section in gui.

contacts.cfg : all the recipients to whom you want to send notifications can be configured in this file. Also contact group can be specified.

templates.cfg : all templates are defined in this file.

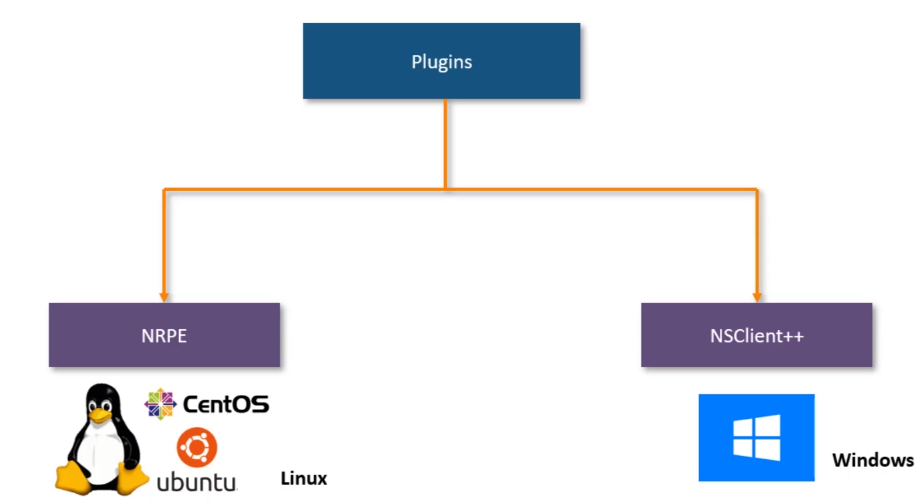


Use keyword is used for calling predefined template. Below we have template for windows server so we have called it.



Plug-in : This is piece of software which is installed in client or host which is used to communicate with server. Using this host can interact with nagios server and send application logs.

Types Of Plugins:

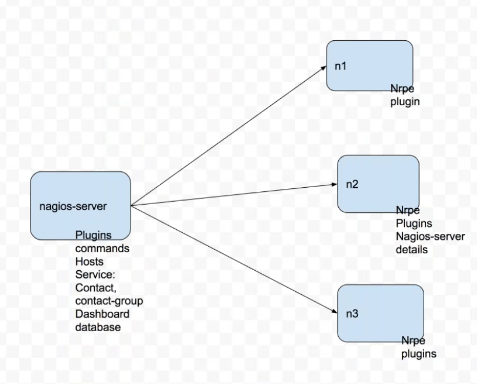


Plugins are piece of software that one installs on nagios slave and host,using which host can interact with server and send application logs.

Types of plugins:

NRPE for unix/linux

NSClient++ for windows



In all the slave nodes we install NRPE agent or process.

Nagios server will request nrpe agents in slave nodes to execute and the output is sent back to nagios server.

In Nagios sever:

Host information: Target node information

These are mapped using services

services are mapped with plugins or commands

Plugins will be there in both master and slave. It is nothing but program

Plugins are converted to commands

These commands are mapped to services.

In slave nodes we have only NRPE and slave nodes.

# The following examples use hardcoded command arguments...

command[check\_users]=/usr/lib64/nagios/plugins/check\_users -w 15 -c 30

command[check\_load]=/usr/lib64/nagios/plugins/check\_load -w 15,10,5 -c 30,25,20

command[check\_root]=/usr/lib64/nagios/plugins/check\_disk -w 20% -c 10% -p /dev/xvda1

command[check\_zombie\_procs]=/usr/lib64/nagios/plugins/check\_procs -w 5 -c 10 -s Z

command[check\_total\_procs]=/usr/lib64/nagios/plugins/check\_procs -w 900 -c 1000

command[check\_uptime]=/usr/lib64/nagios/plugins/check\_uptime

command[check\_tcp\_es]=/usr/lib64/nagios/plugins/check\_tcp -H localhost -p 9200

command[check\_jenkins]=/usr/lib64/nagios/plugins/check\_tcp -H localhost -p 8080

# The following examples allow user-supplied arguments and can

# only be used if the NRPE daemon was compiled with support for

# command arguments \*AND\* the dont\_blame\_nrpe directive in this

# config file is set to '1'. This poses a potential security risk, so

# make sure you read the SECURITY file before doing this.

#command[check\_users]=/usr/lib64/nagios/plugins/check\_users -w $ARG1$ -c $ARG2$

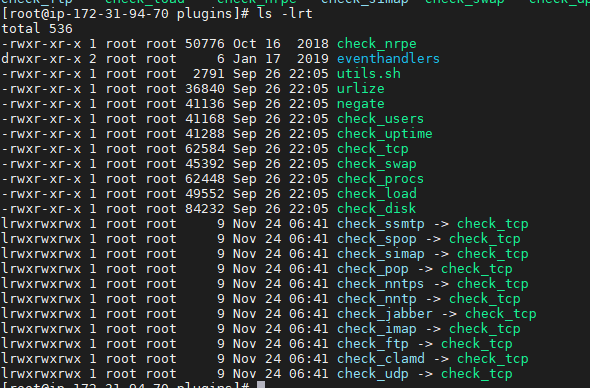
#command[check\_load]=/usr/lib64/nagios/plugins/check\_load -w $ARG1$ -c $ARG2$

#command[check\_disk]=/usr/lib64/nagios/plugins/check\_disk -w $ARG1$ -c $ARG2$ -p $ARG3$

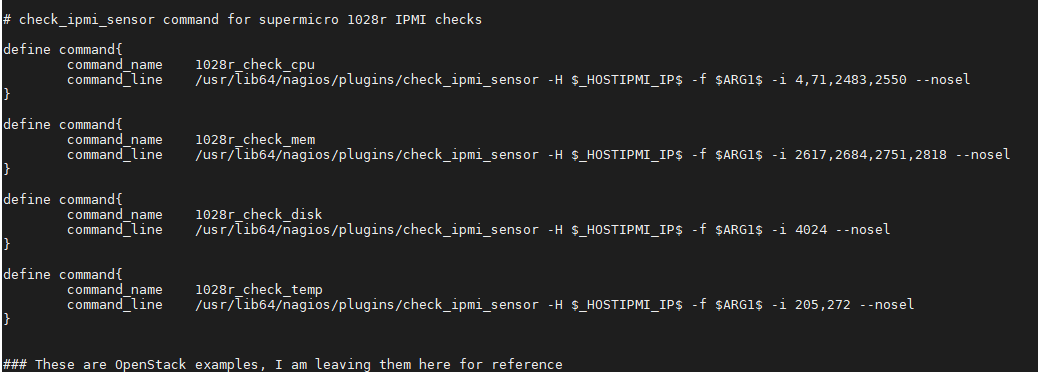
#command[check\_procs]=/usr/lib64/nagios/plugins/check\_procs -w $ARG1$ -c $ARG2$ -s $ARG3$

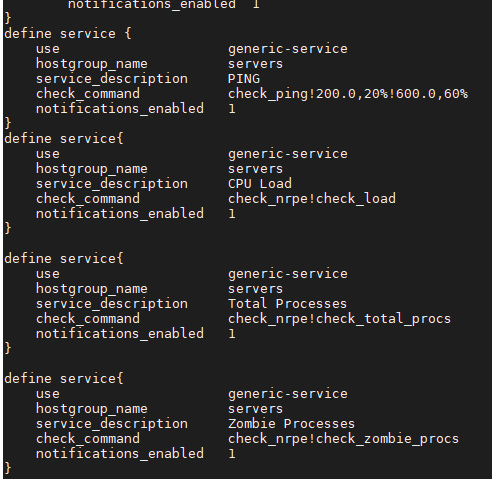
Plugins are converted to commands and these commands are mapped to services

Plugins in nagios target node:



These plugins are converted to commands in nagios host

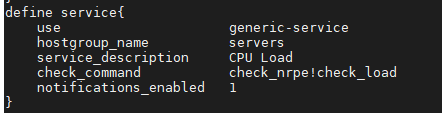




For example, you can take service CPU LOAD :

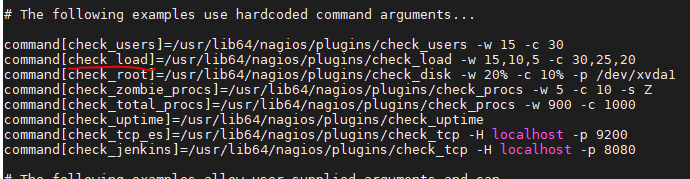


So if you got to nagios server and got to **/etc/nagios/conf.d/server.cfg** you will find below service related to **CPU Load**:



So in nrpe side check\_load command is mapped and will be executed.

So you can check whether these commands are there in nagios slave:



In client or slavenode commands or plugins will be executed. So these plugins will be mapped to nagios services in nagios server. So whenever command or plugin is executed the values are populated in nagios server.

Creating custom plugin:

For ex refer below pythin script:

#!/usr/bin/python

import os, sys

used\_space=os.popen("df -h / | grep -v Filesystem | awk '{print $5}'").readline().strip()

if used\_space < "85%":

print "OK - %s of disk space used." % used\_space

sys.exit(0)

elif used\_space == "85%":

print "WARNING - %s of disk space used." % used\_space

sys.exit(1)

elif used\_space > "85%":

print "CRITICAL - %s of disk space used." % used\_space

sys.exit(2)

else:

print "UKNOWN - %s of disk space used." % used\_space

sys.exit(3)

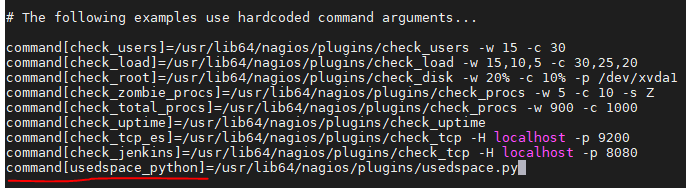
In client or target node:

Go to /**usr/lib64/nagios/plugins** and create **usedspace.py** file.

Paste the above script in **usedspace.py** file.

Now go to **/etc/nagios/nrpe.cfg** file. Add below command

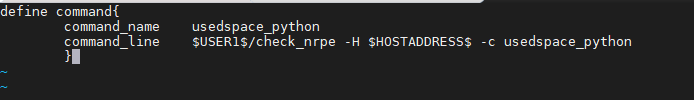
**command[usedspace\_python]=/usr/lib64/nagios/plugins/usedspace.py**

****

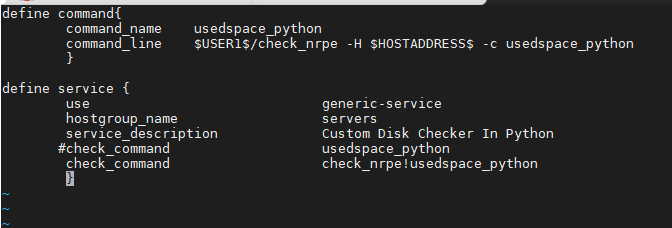
Once done we have to map this command or plugin to command in nagios host or master.

In Nagios host or master:

Go to /etc/nagios/conf.d directory and either you can add it in server.cfg or create separate custom.cfg

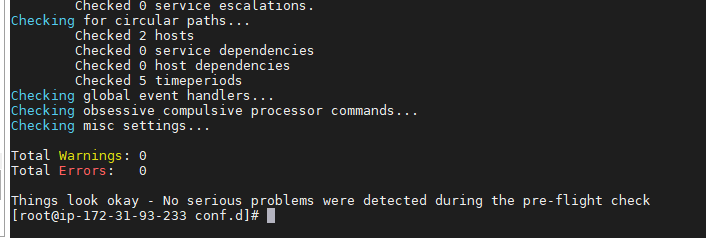


Also define service in custom.cfg. 3 things are mapped custom service and hostgroup.



To check any syntax errors in nagios master issue below command

**nagios -v ../nagios.cfg**

****

Now restart nrpe service in client or slave:

**service nrpe restart**

Also restart nagios service and httpd service in master:

**service nagios restart**

**service httpd restart**

Once done will get below plugin enabled:

****

Only infra monitoring can be done using Nagios.