NAGIOS

Introduction:-

As a sys admin we can manually monitor the server by executing commands and writing scripts, but it will be difficult for sys admin to manage thousands of servers in our environment

So there is a third party tools for monitoring infrastructure automatically, which reduces manual effort

Linux monitoring system is one of the most important tasks of sys admin to check CPU load network traffic, memory consumption, logged in users, disk space and services.

What is monitoring tool?

It is a software which monitors remote servers at regular interval of time to detect before something goes wrong

1. Monitoring tool detects the problem of remote host and their services before the user does

Ex: mail server failure, hard drive overloaded, network issues

1. Monitoring tool informs to the system administrator when something goes wrong in the remote host

Why monitoring tool:-

1. High availability
2. Reduce downtime
3. Supports both virtual and physical machines
4. Monitors entire infrastructure like servers switches services & applications

Types of monitoring:-

1. System monitoring is which monitors servers, network devices like Nagios, solar winds, zabbix
2. Application monitoring is which monitors applications like Newrelic, grafana, kibana
3. Log monitoring is which monitors log management like Elk, spluk

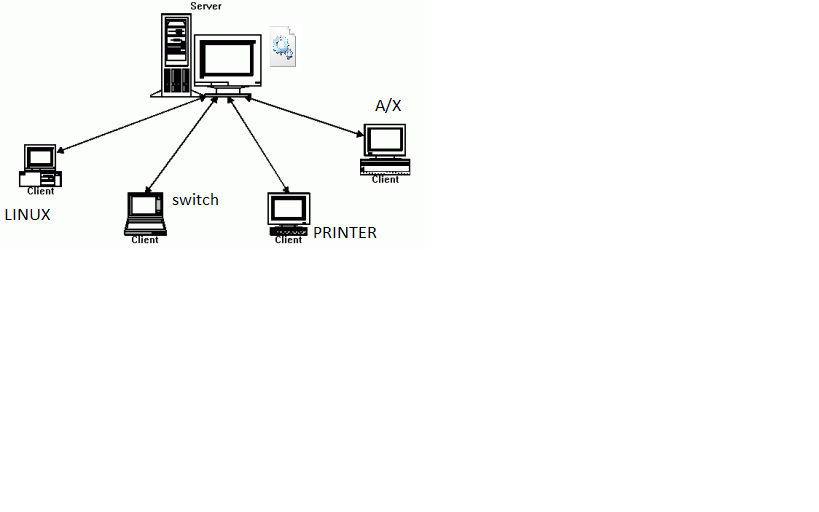
Vendors:-

1. Nagios
2. Solar winds
3. Zabbix
4. Gangula
5. Elk
6. Splunk
7. newrelic

why nagios:-

1. It checks if a server/service/application is up and running
2. Notify if server is down
3. It gives reports of downtime on your servers
4. It triggers alert to particular group/individual
5. Gathers statistics of server (performance graph of how many servers went down)
6. Nagios can be used for linux and windows based systems
7. Open source

NAGIOS ARCHITECTURE:-

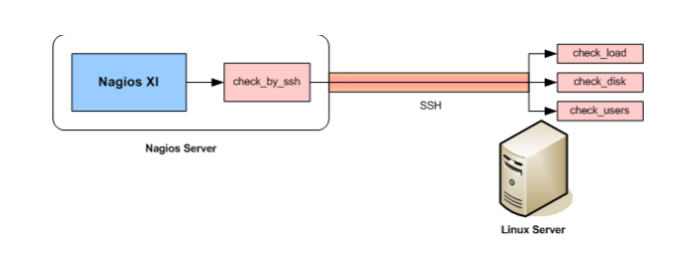


Nagios is a server client architecture. Nagios software is installed on central server which will monitor remote hosts

NAGIOS WORKFLOW:-

Nagios server has some set of config files, in which we are going to define commands, services & hosts information. There will be a plugin installed on nagios server. This plugin allows nagios to execute commands/checks/scripts on remote hosts and sends back the result to nagios server which will be stored in nagios database and we can see the result in nagios dashboard

If the Nagios server didn’t get result back immediately it triggers an alert via email, sms, and phone if we are not available escalated to other teams



Ex: - there is a nagios server and linux host. I have defined in config file to check communication for every 1 minute on particular linux host. The daemon (sshd/nrpe) reads the config file and connects to the server, executes ping command and the result will be sent back to nagios server. Nrpe installed on server is plugin whereas nrpe running on client is daemon. Nrpe connects to remote server via SSL connection

CONNECTION:-

1. tcp

PORT NO:-

5666=nrpe

80,443=snmp

161,162=inbound

5432=postgress

25,465,587=email

135,445=windows management interface

22=ssh

12489=

DAEMON:-

1. nrpe = this daemon run on client side
2. xinetd = listens for incoming request over the network like which port it is which service it is

CONFIGURATION FILES:-

1. /usr/local/nagios/bin----------by default nagios will be installed here.
2. /usr/local/nagios/sbin--------------by this file we will get webpage
3. /usr/local/nagios/libexec---------all plugins present in this directory
4. /usr/local/nagios/share--------------php files for reports, dashboard etc..
5. /usr/local/nagios/etc--------------main config file where daemon will read
6. /usr/local/nagios/var------------log files, pid etc..
7. Status.dat-----------------this config file is used to store the output of the commands executed on remote hosts
8. PLUGINS:-

Plugins are nothing but the scripts/files that are executable. These scripts may be written in shell script/python/c/ruby/perl

Plugins execute on remote server and return some code, based on that code we know the status of plugins

1. If plugin executed and encounters okay, then status code is 0
2. If plugin executed and encounters warning, then status code is 1
3. If plugin executed and encounters critical, then status code is 2
4. If plugin doesn’t know how to handle, then status code is 3

* All plugins are present in libexec directory
* If you don’t know how to use the plugin, then choose a plugin and execute as below

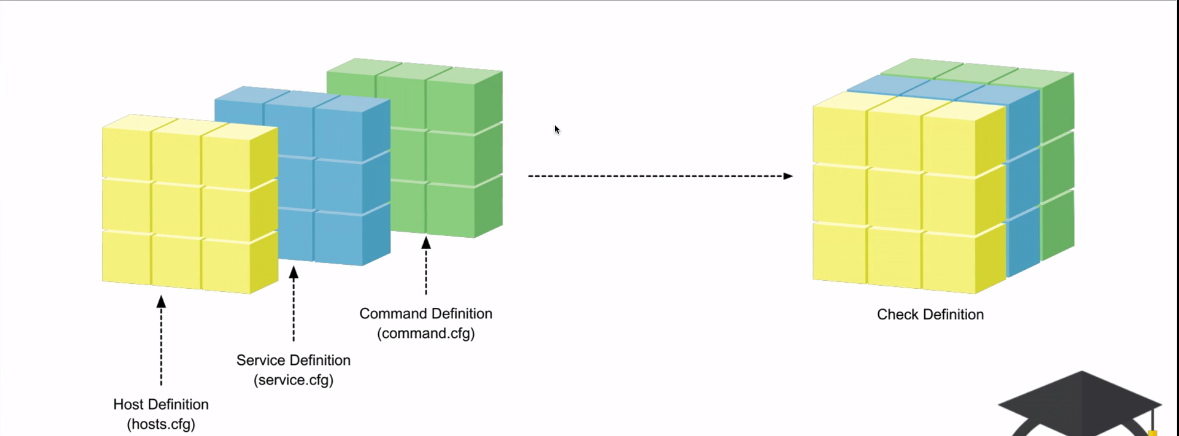
#./plugin

Ex: - ./check\_ping -H host1.com

Where . / represents current directory, h=help

1. CHECKS:-

A check needs host, service & command information to perform on nagios clients. Hosts, services & commands are tightly couple

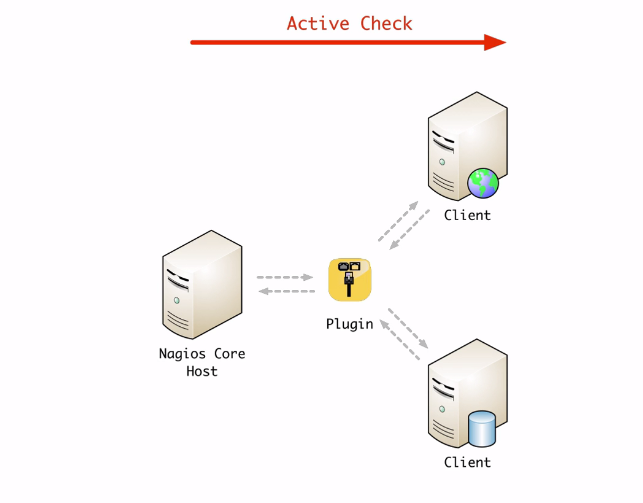


Host definition (hosts.cfg) + Service definition (service.cfg) +Commands definition (commands.cfg) = Check definition

There are two types of checks

1. Active checks:-

Active checks are initiated/originated by nagios server. Nagios server monitors each and every step. Nagios executes checks on remote hosts and the results are sent back to nagios server



1. Passive checks:-

In passive checks, client manages check logic. Client initiates the check and sends the result back to nagios server

Passive checks are used with 2 plugins ie.., NCSA, NRDP

ex:- emailing script results

1. HOSTS, HOSTS DEFNITION, OBJECTS:-

Hosts are any devices which are connected to network that your nagios monitor ex: - desktop, servers, printers, network devices, applications, etc….

Host object refers to single device

Host definition are used to define host objects. We define host definitions in a file ex: - localhost.cfg is a host definition file

We can create our own host definition files

#vi localhost.cfg

----------------------------------------------------------

define host{

hostname linux1.com [the name of remote server where plugins execute]

alias http server [other name of the server]

check\_period 24\*7 []

check\_interval 50 [at how many regular interval of time should check]

retry\_interval 10 [how many minutes should nagios wait before it retries]

max\_check\_attempts 3 [how many attempts should nagios check]

check\_command check\_ping [what check it has to execute on remote host]

contact\_groups linux-team [to which group it should be notified when something goes wrong]

notification\_enabled 1 [if set to 1 then notification will be recived]

flap\_detection\_enabled 1 [flap is a condition where network interface/servers goes and up and down repeatedly]

process\_performance\_data 1 []

retain\_status\_information 1 [if nagios is restarted then it has to maintain all the previous collected data]

notification\_interval 10 [this will send notification till someone acknowledge]

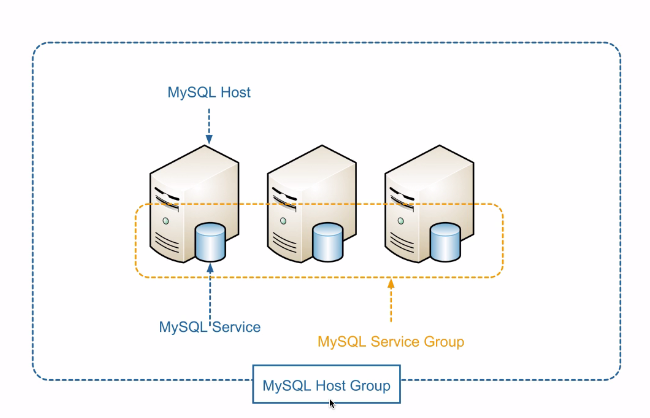
notification\_option d,u,r [this option lets nagios when notification to be sent]

}

\*\*these hostname, alias, check\_period ……notification\_option are called directives

------------------------------------------------------------------------------------------

Hostgroups are combination of all hosts defined together

Ex:- instead of defining services for each and every mysql servers, we group all myql servers and define a service for it

1. MACROS:-

Instead of specifying the long path every time, everwhere or anything which is used and reusable, which will be difficult. So macros are used as substitute

Ex:- /usr/local/nagios/libexec this path is reused in many files. So we substitute this with macro

$USER1$=/usr/local/nagios/libexec

This $USER1$ is called macro and will be used in every file

Similarly you have lot of macros like $HOSTNAME$

1. SERVICES, SERVICE DEFNITION, SERVICE OBJECTS:-

A process/function that is monitored on remote host

Ex:- cpu, disk, memory usages, dns

Just like host definition we also have service definition

#vi service.cfg

---------------------------------------------------------------------------

define service{

hostname localhost

service\_description ping

check\_command check\_ping

}

---------------------------------------------------------------------------

1. COMMANDS, COMMAND DEFNITION, COMMAND OBJECTS:-

A command object is a single command whereas command definition is a file where we have multiple commands we can run our own commands, modify values

#vi commands.cfg

--------------------------------------------------------------------------------------------

define command{

command name check\_host\_by\_ping

command line $USER1$/../libexec/check\_ping –H$HOSTADDRESS$ -w 2000.00,80% -c 5000.00,100% -p 5

}

\*where w=warning, c=critical

-----------------------------------------------------------------------------------------------

1. TIMEPERIOD:-

Set services according to time period

1. CONTACTS:-

If something goes wrong then we will notified using contacts. We have contact objects, contact defnitions

#vi contacts.cfg

----------------------------------------------------------------------------------------

define contact{

contact\_name nagios-admin [whom to contact]

alias NagiosAdmin [othername]

email [nagiosadmin@gmail.com](mailto:nagiosadmin@gmail.com) [to which email notification has to be sent]

service\_notification\_period 24\*7 [time period]

host\_notification\_period normal []

service\_notification\_option w,u,f,c,r []

host\_notification\_option d,u,r, f []

service\_notification\_commands notify-service-by-email [we already have this command notify-service-by-email in commands .cfg we just need to specify here]

host\_notification\_command notify-host-by-email []

}

define contact group{

contact group-name

}

1. TEMPLATES:-

Templates are readily available all you need to do is just edit and use accordingly

There are templates for hosts, contacts, services

\*templates.cfg is config file for templates

1. NOTIFICATIONS:-

When a status is changed then notification are sent



1. Host status:-
2. Down (d):-

If server is completely down or not running

1. Unreachable (u):-

If server is not up and running and if no network communication

1. None (n):-

If server is running normal

1. Recovery (r):-

If server is down and did not come back to normal position

1. service status:-
2. warning (w):-

If service matches threshold values

1. critical (c):-

If service exceed threshold value

1. recovery (r):-

Past critical but now normal is regarded as recovery

1. unknown (u):- if service is not able to run and unable to show output

ex:- $USER1$/check\_ping -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -p 5

1. VERIFYING CONFIG FILE:-

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

After modifying config files, the above command let us know if there are any errors creeping

And restart nagios service if everything is fine

#service nagios restart

1. ADDING HOSTS:-

You can either add host entries in usr/local/nagios/etc/objects/ localhost.cfg (or) you can create your own host file and add hosts

Ex:- create your own file and update its entry in nagios config file

#cd /usr/local/nagios/etc/

#touch linuxhosts.cfg

Now add this host entry in nagios.cfg file

# vi /usr/local/nagios/etc/nagios.cfg

cfg\_file=/usr/local/nagios/etc/linuxhosts.cfg

now modify your linuxhosts.cfg file

#vi /usr/local/nagios/etc/linuxhosts.cfg -------------------------------------------------------------------------------------

## Default Linux Host Template ##

define host{

name linux-box ; Name of this template

use generic-host ; Inherit default values

check\_period 24x7

check\_interval 5

retry\_interval 1

max\_check\_attempts 10

check\_command check-host-alive

notification\_period 24x7

notification\_interval 30

notification\_options d,r

contact\_groups admins

register 0 ; DONT REGISTER THIS - ITS A TEMPLATE

}

## Default

define host{

use linux-box ; Inherit default values from a template

host\_name host2.com ; The name we're giving to this server

alias jenkins ; A longer name for the server

address 192.168.197.201 ; IP address of Remote Linux host

}

----------------------------------------------------------------------------------------------------

1. CONFIGURING HOSTGROUPS:-

Hostgroup is grouping of one or multiple hosts.

Ex:- grouping all apache webservers

#cd /usr/local/nagios/etc

#touch hostsgroup.cfg

Now add this entry in nagios.cfg

#vi /usr/local/nagios/etc/nagios.cf

cfg\_file=/usr/local/nagios/etc/hostsgroup.cfg

now edit the hostsgroup as you wanted

#vi /usr/local/nagios/etc/hostsgroup.cfg

---------------------------------------------------------------------------------------------------

define hostgroup{

hostgroup\_name webserver

alias apache

members localhost, host2.com

}

----------------------------------------------------------------------------------------------------

1. ADDING SERVICES:-

You can either add services entries in usr/local/nagios/etc/objects/ localhost.cfg (or) you can create your own service file and add services

Ex:- create your own file and update its entry in nagios config file

#cd /usr/local/nagios/etc

#touch services.cfg

Add this service.cfg file in nagios.cfg file

#vi /usr/local/nagios/etc/nagios.cfg

cfg\_file=/usr/local/nagios/etc/services.cfg

now add service defnitions and modify however you want

#vi /usr/local/nagios/etc/services.cfg ---------------------------------------------------------------------------------------------

define service{

use generic-service

host\_name host2.com

service\_description Total Processes

check\_command check\_nrpe!check\_total\_procs

}

---------------------------------------------------------------------------------------------

1. CONFIGURING SERVICE GROUP:-

Grouping of one or more services together so that we can easily check services at one place (same services but hosts are different)

Ex:- mysql service running on different hosts/servers

#cd /usr/local/nagios/etc

#touch servicesgroup.cfg

Now update this in nagios config file

#vi /usr/local/nagios/etc/nagios.cfg

cfg\_file=/usr/local/nagios/etc/servicesgroup.cfg

#vi /usr/local/nagios/etc/servicegroup.cfg

----------------------------------------------------------------------------------------------------

define servicegroup{

servicegroup\_name nfs-servers

alias network-f.s

members host2.com,PING,localhost,PING ;//host1,service,host2,service…

}

---------------------------------------------------------------------------------------------------

1. CONFIGURING NRPE COMMAND:-

Now add nrpe command definition in commands.cfg file

#vi /usr/local/nagios/etc/services.cfg ---------------------------------------------------------------------------------------------

define command{

command\_name check\_nrpe

command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

}

---------------------------------------------------------------------------------------------

1. ADDING OUR PLUGINS:-

We can add our own custom plugins and let them run on remote hosts

1. Add your plugin in libexec directory

#cd /usr/local/nagios/libexec

#vi check\_load\_average.sh (\*give any name with any extension)

Your script goes here

1. change the permission of plugin

#chown nagios:nagios /usr/local/nagios/libexec/check\_load\_average.sh

#chmod 777 check\_load\_average.

1. Now add our plugin entry in commands.cfg

#vi /usr/local/nagios/etc/objects/commands.cfg

define command{

command\_name check\_load\_average

command\_line $USER1$/check\_load\_average.sh -H $HOSTADDRESS$

}

1. Add the command in your service file may be localhost.cfg

#vi localhost.cfg/#vi services.cfg/#vi hosts.cfg

define service{

use generic-service

host\_name Host2.com

service\_description check\_load\_average

check\_command check\_load\_average

notifications\_enabled 1

}

1. Check nagios config file for errors

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

1. restart nagios services

#service nagios restart

1. WEB INTERFACE:-

NAGIOS OPERATION:-

1. Open browser and enter nagios link ex:- http://hostname/nagios
2. Give your credentials
3. You will see homepage with customized dashboards for a host
4. Dashboard contains dash lets which can be added removed modified

DASHBOARD:-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HOST | SERVICE | STATUS | LASTCHECK | DURATION | ATTEMPT | STATUSINFO |
| Cisco | Ping | Ok | 04.12.2017 05.20pm | 0d0h42m18s | 1/3 | ping ok packet loss=0% |
| 192.168.43.5 | http | Warning | 04.12.2017 05.20pm | 0d0h42m18s | 3/3 | http warning 403 forbidden |
| Google | /dev/hda1 free space | Critical | 04.12.2017 05.20pm | 0d0h42m18s | 3/3 | Disk critical free space |

DASHBOARD OPTIONS:-

1. HOST=local or remote
2. SERVICE=Ssh, disk space, http, load, etc…
3. STATUS=(green is ok),(yellow is warning),(red is critical)
4. LAST CHECK=date
5. DURATION=time taken
6. ATTEMPT=out of n attempts how many attempts done and succeeded
7. STATUS INFO=server status