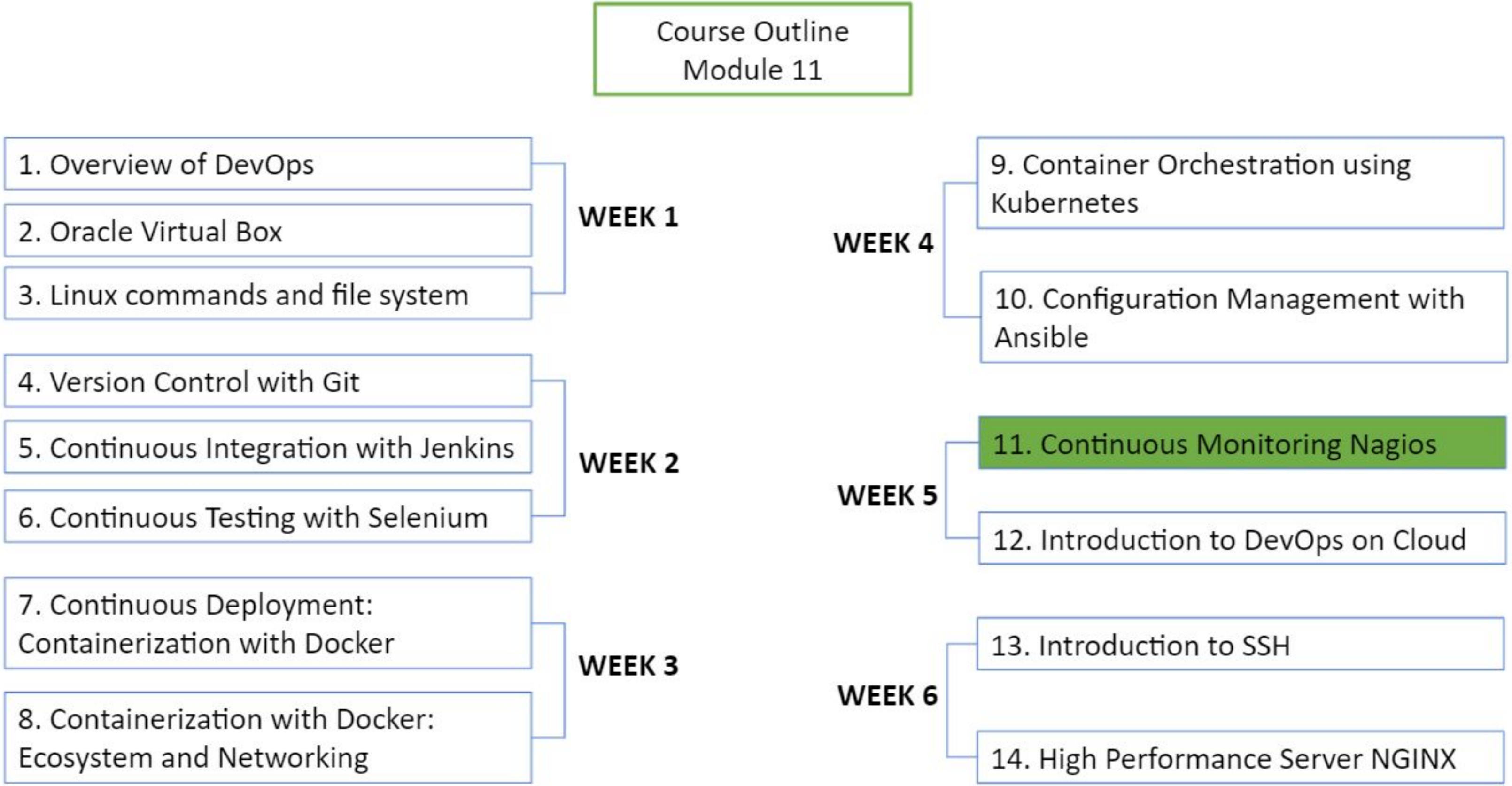
Nagios

Continuous Monitoring With Nagios

Week 5



Course Outline
Module 11

WEEK 1

1. Overview of DevOps
2. Oracle Virtual Box
3. Linux commands and file system

WEEK 2

4. Version Control with Git
5. Continuous Integration with Jenkins
6. Continuous Testing with Selenium

WEEK 3

7. Continuous Deployment: Containerization with Docker
8. Containerization with Docker: Ecosystem and Networking

WEEK 4

9. Container Orchestration using Kubernetes
10. Configuration Management with Ansible

WEEK 5

11. Continuous Monitoring Nagios
12. Introduction to DevOps on Cloud

WEEK 6

13. Introduction to SSH
14. High Performance Server NGINX

Topics

- Continuous Monitoring
- Introduction to Nagios
- Nagios Architecture
- Objects in Nagios
- States in Nagios
- Nagios Dashboard

Objective

At the end of this module you will be able to

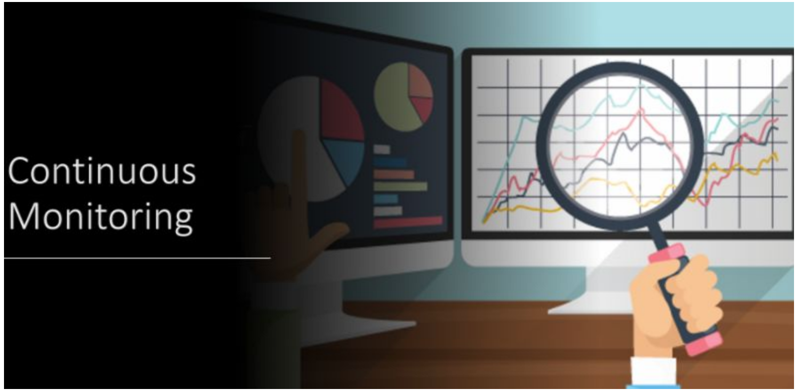
- Understand Continuous Monitoring
- Introduction to Nagios
- Install Nagios
- Learn about the Nagios Plugins (NRPE) and Objects
- Understand different types of states in Nagios
- Execute different Nagios Commands and Notifications
- Understand about Nagios Dashboard and how to monitor a remote host

Why we use Monitoring?

- System monitoring is crucial in an organization’s mission critical system
- Monitoring alerts system admins about a problem in the system beforehand
- It enables preventive measures before any issue affects the system
- Monitoring is essential in ensuring system availability
  - Keeping track of the status of a system
  - For example: Server disk space, file system status, status changes in the services etc.

Benefits of Continuous Monitoring

- It helps in getting rid of periodic testing
- It detects split-second failures when the wrist strap is still in the “intermittent” stage
- It reduces maintenance cost without sacrificing performance
- It provides timely notification to the management of control and breakdown



Available Monitoring Tools in the Market

Log Monitoring

DATADOG

splunk>

Real-time Monitoring

Nagios

ZABBIX

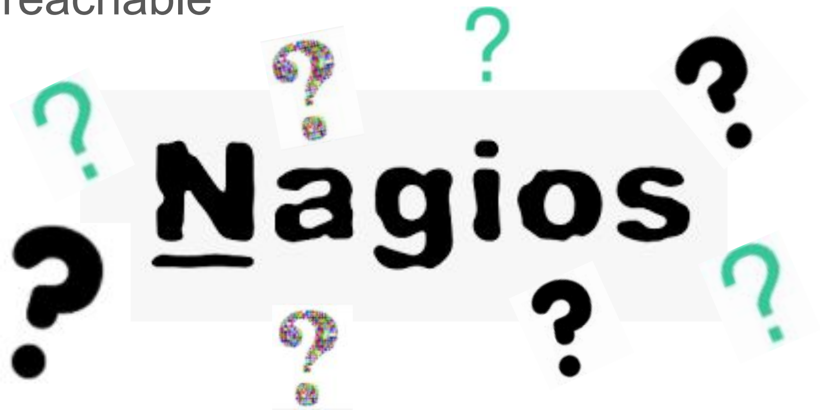
Amazon CloudWatch

Container Monitoring

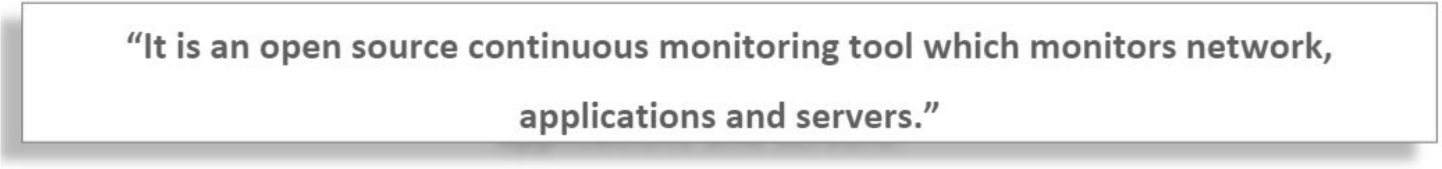
Prometheus

Why Nagios?

- It can monitor database server such as SQL Server, Oracle, MySql, Postgres
- It gives allocation level information (Apache, Postfix, LDAP, Citrix etc)
- Active Development
- Active Community
- Nagios can work on multiple Operating Systems
- It can ping to see if host is reachable



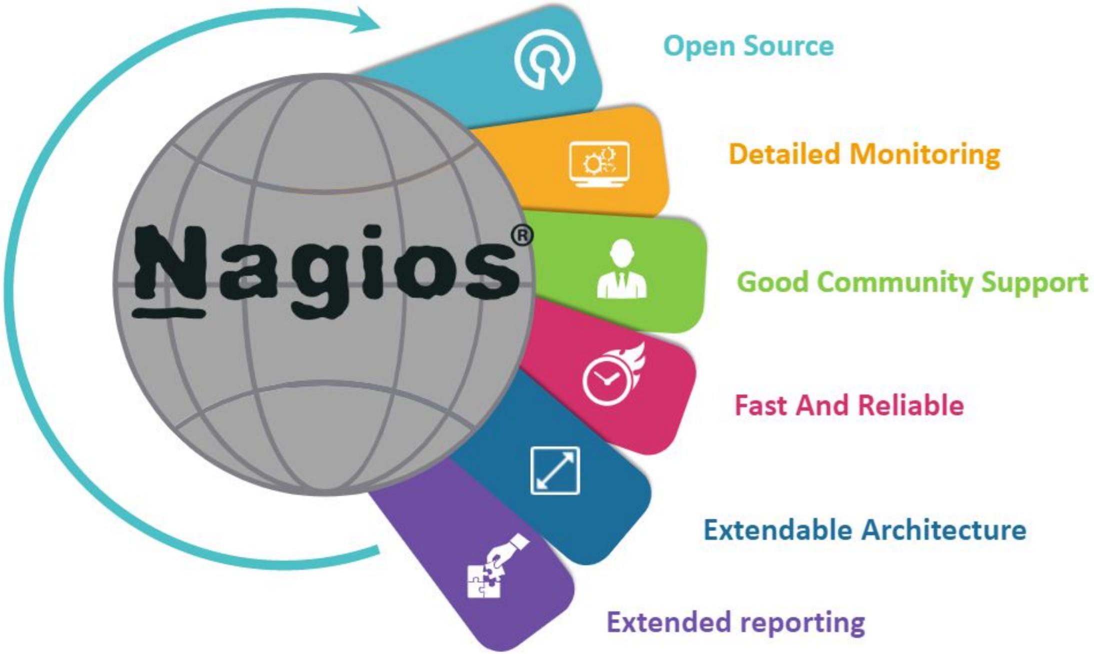
What is Nagios?



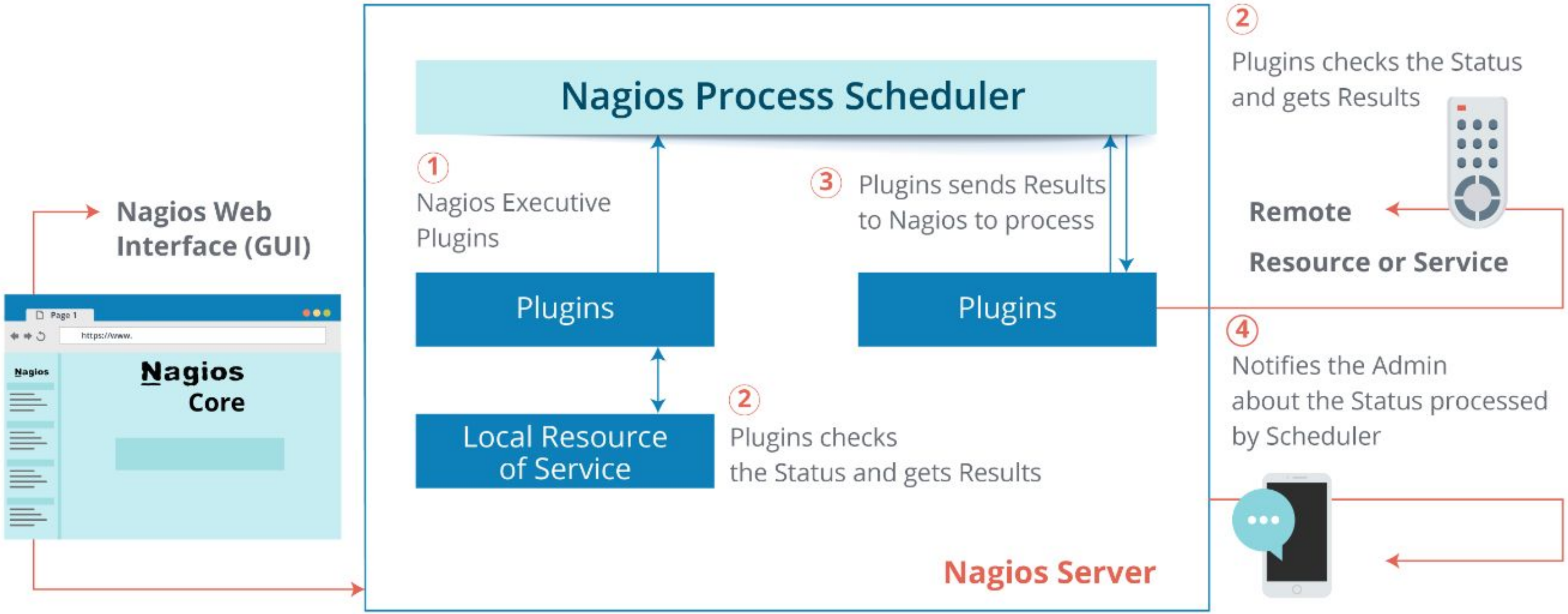
> “It is an open source continuous monitoring tool which monitors network, applications and servers.”

- It allows you to detect and repair problems and mitigate future issues before they affect end-user and customers
- It is developed for monitoring servers, applications and network
- It provides a centralized view of your entire IT infrastructure and detailed up-to-date status information

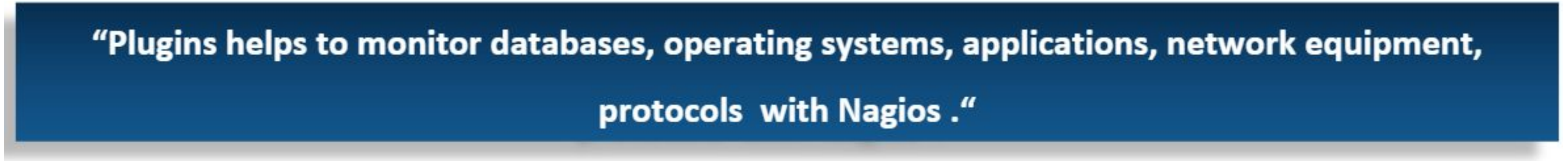
Features of Nagios



Nagios Architecture


Nagios Web Interface (GUI)
Page 1
https://www.
Nagios
Nagios Core
Nagios Process Scheduler
1
Nagios Executive Plugins
3
Plugins sends Results to Nagios to process
Plugins
Plugins
Local Resource of Service
2
Plugins checks the Status and gets Results
Nagios Server
2
Plugins checks the Status and gets Results
Remote Resource or Service
4
Notifies the Admin about the Status processed by Scheduler

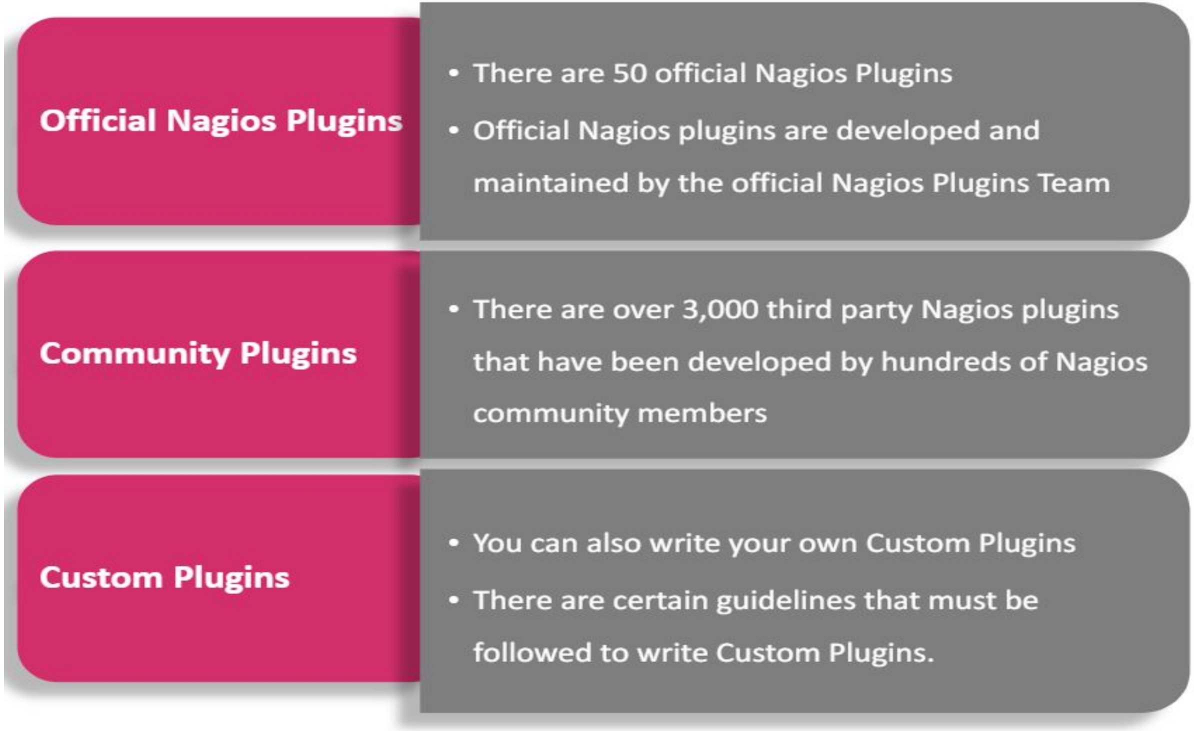
Nagios Plugins



“Plugins helps to monitor databases, operating systems, applications, network equipment, protocols with Nagios .“

- Plugins are compiled executables or scrips (Perl or non-Perl) that extends Nagios functionality to monitor servers and hosts
- Nagios will execute Plugin to check the status of a service or host
- Nagios can be compiled with support for an embedded Perl interpreter to execute Perl plugins
- Without it, Nagios executes Perl and non-Perl plugins by forking and executing the plugins as an external command
- Nagios comes with 50 plugins as default installation, these are binary files
- Check plugins in directory: /usr/local/nagios/libexec

Types of Plugins



Official Nagios Plugins

- There are 50 official Nagios Plugins
- Official Nagios plugins are developed and maintained by the official Nagios Plugins Team

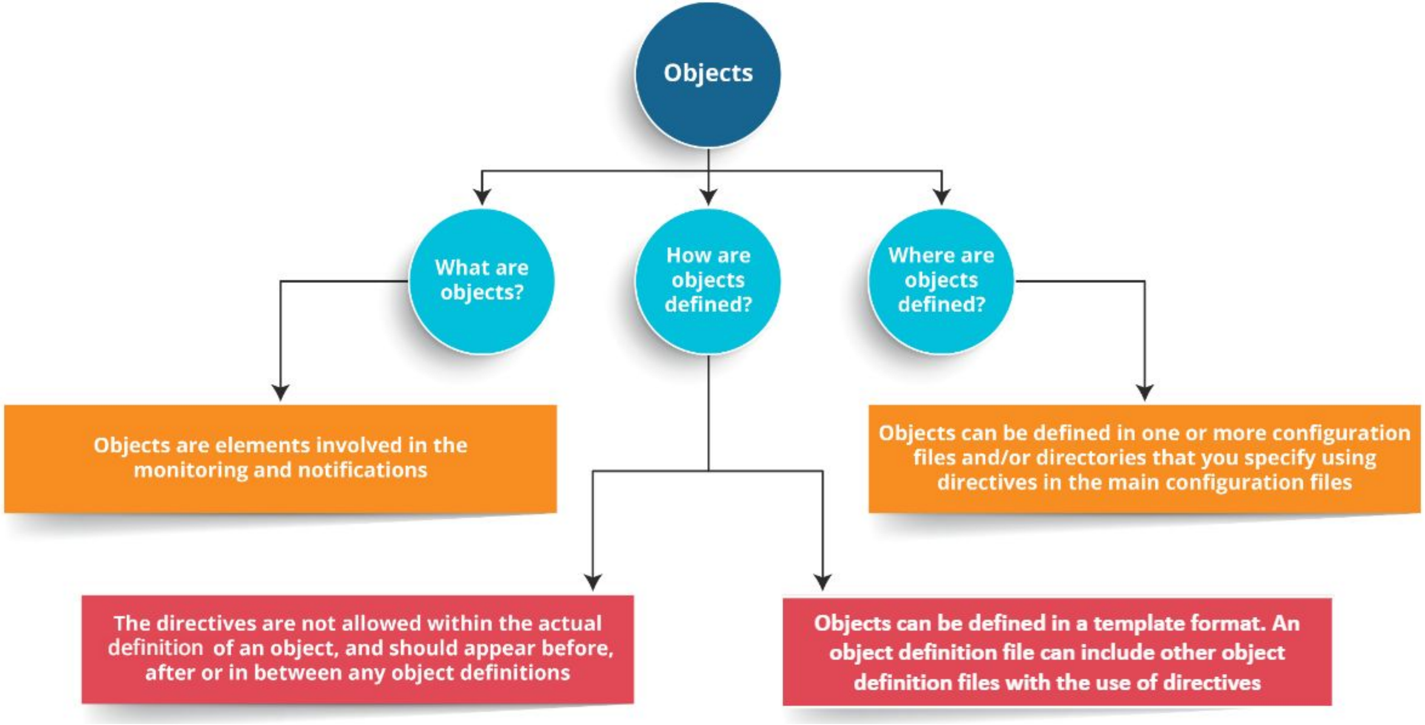
Community Plugins

- There are over 3,000 third party Nagios plugins that have been developed by hundreds of Nagios community members

Custom Plugins

- You can also write your own Custom Plugins
- There are certain guidelines that must be followed to write Custom Plugins.

Nagios Objects



Objects

What are objects?

How are objects defined?

Where are objects defined?

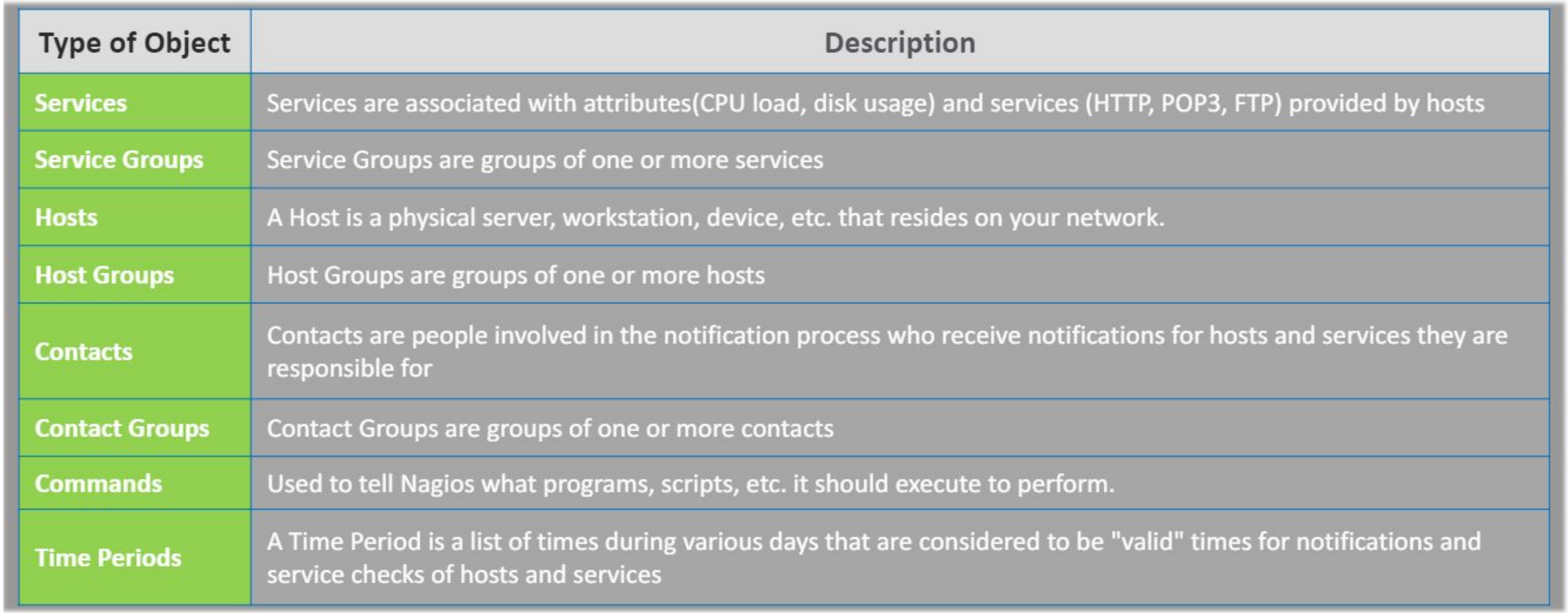
Objects are elements involved in the monitoring and notifications

Objects can be defined in one or more configuration files and/or directories that you specify using directives in the main configuration files

The directives are not allowed within the actual definition of an object, and should appear before, after or in between any object definitions

Objects can be defined in a template format. An object definition file can include other object definition files with the use of directives

Object Types and Definitions



Type of Object	Description
Services	Services are associated with attributes(CPU load, disk usage) and services (HTTP, POP3, FTP) provided by hosts
Service Groups	Service Groups are groups of one or more services
Hosts	A Host is a physical server, workstation, device, etc. that resides on your network.
Host Groups	Host Groups are groups of one or more hosts
Contacts	Contacts are people involved in the notification process who receive notifications for hosts and services they are responsible for
Contact Groups	Contact Groups are groups of one or more contacts
Commands	Used to tell Nagios what programs, scripts, etc. it should execute to perform.
Time Periods	A Time Period is a list of times during various days that are considered to be "valid" times for notifications and service checks of hosts and services

Nagios Installation

- Need to spin up 2 EC2 VMs if using AWS or open 2 virtual box VMs
- Nagios is not available as a binary package hence it require manual installation. Nagios needs to be installed from source. Here are the steps:
    - Change to root user: *sudo -i*
    - Download and Install Apache (All Nagios dependencies):
    - *apt-get update && apt-get install build-essential apache2 php openssl perl make php-gd libgd-dev libapache2-mod-php libperl-dev libssl-dev daemon wget apache2-utils unzip*
    - For VMbox use command: *apt-get update && apt install -y autoconf bc gawk dc build-essential gcc libc6 make wget unzip apache2 php libapache2-mod-php libgd-dev libmcrcd ypt-dev make libssl-dev snmp libnet-snmp-perl gettext*
    - Check if Apache is installed in Nagios Server by pasting public IP or static IP into the web browser
    - Create nagios user and nagcmd group and add the nagios and apache user to the part of the nagcmd group
    - *useradd nagios*
    - *groupadd nagcmd*
    - *usermod -a -G nagcmd nagios*
    - *usermod -a -G nagcmd www-data*

Nagios Installation (cont)

- Install Nagios Core
  - *cd /tmp*
  - *wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz*
  - *tar -zxvf /tmp/nagios-4.4.6.tar.gz*
  - *cd /tmp/nagios-4.4.6/*
- Perform the below steps to compile the Nagios from the source code
  - *./configure --with-nagios-group=nagios --with-command-group=nagcmd --with-httpd_conf=/etc/apache2/sites-enabled/*
  - *make all*
  - *make install*
  - *make install-init*
  - *make install-config*
  - *make install-commandmode*
- Execute the below command in the terminal to install Nagios web interface
  - *make install-webconf*
- Create Nagios Login and Password (use “nagios” as password)
  - *htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin*
- Run the following command: *a2enmod cgi*
- Restart Apache Service to make the new settings take effect
  - *systemctl restart apache2.service*

Nagios Installation (cont)

- In web browser type: <server public-ip/nagios>. Click Hosts and you will see an error. This is because Nagios plugins has not been installed yet. Install Nagios plugins:

  ```
  cd /tmp
  wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
  tar -zxvf /tmp/nagios-plugins-2.3.3.tar.gz
  cd /tmp/nagios-plugins-2.3.3/
  ```

- Compile and install the plugins

  ```
  ./configure --with-nagios-user=nagios --with-nagios-group=nagios
  make
  make install
  ```

- Verify the sample Nagios configuration files. You should have “0” warnings and errors

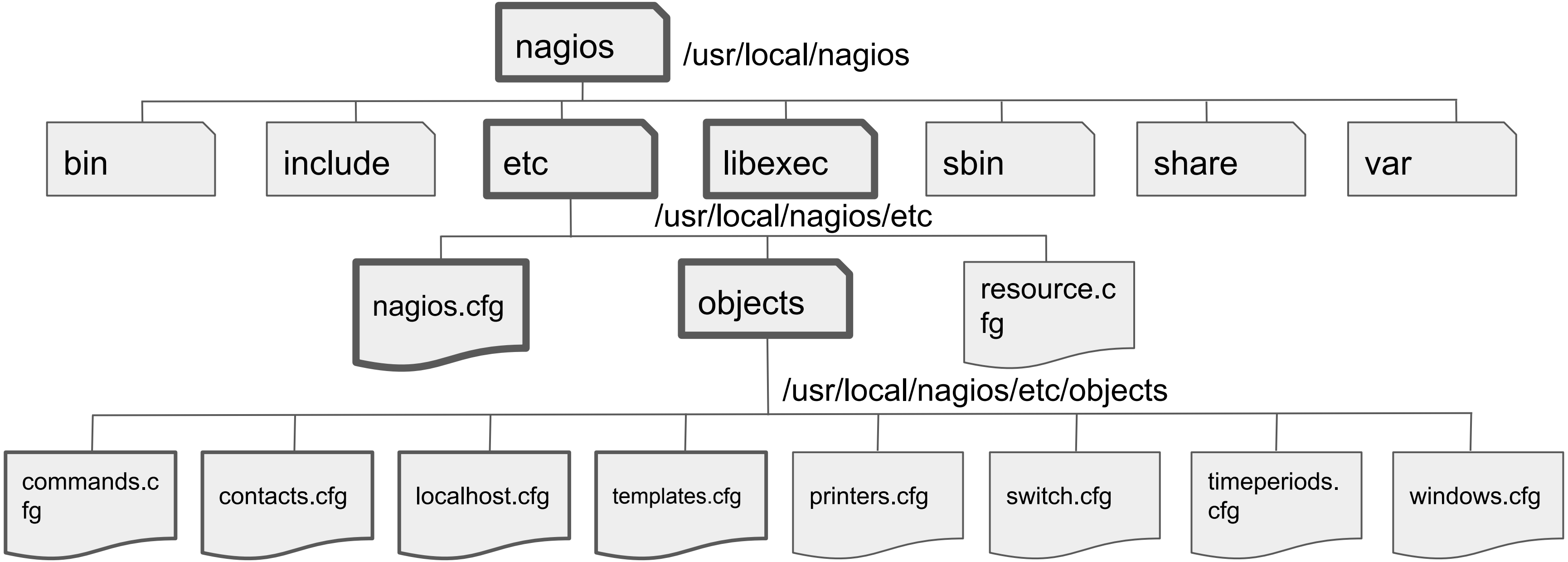
  ```
  /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
  ```

- Enable Nagios to start automatically at system startup and start Nagios service

  ```
  systemctl enable nagios
  systemctl start nagios
  ```

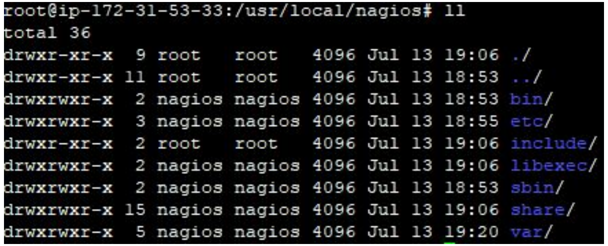
- Your localhost on Nagios dashboard should be up and services should come up

Nagios File System



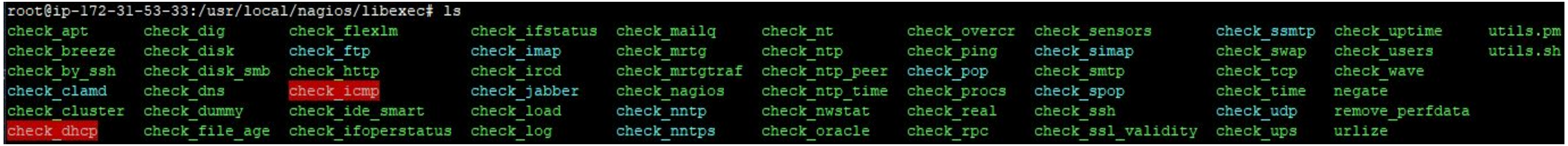
Nagios File System (cont)

- Nagios is installed under directory /usr/local/nagios/



```
root@ip-172-31-53-33:/usr/local/nagios# ll
total 36
drwxr-xr-x  9 root   root   4096 Jul 13 19:06 ./
drwxr-xr-x 11 root   root   4096 Jul 13 18:53 ../
drwxrwxr-x  2 nagios nagios 4096 Jul 13 18:53 bin/
drwxrwxr-x  3 nagios nagios 4096 Jul 13 18:55 etc/
drwxr-xr-x  2 root   root   4096 Jul 13 19:06 include/
drwxrwxr-x  2 nagios nagios 4096 Jul 13 19:06 libexec/
drwxrwxr-x  2 nagios nagios 4096 Jul 13 18:53 sbin/
drwxrwxr-x 15 nagios nagios 4096 Jul 13 19:06 share/
drwxrwxr-x  5 nagios nagios 4096 Jul 13 19:20 var/
```

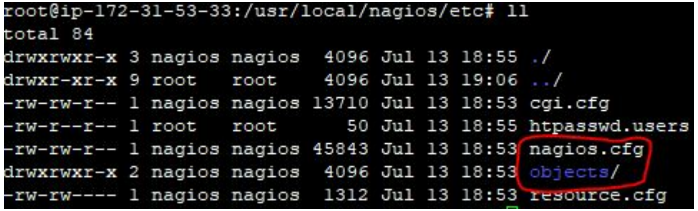
- Under this directory libexec contain all the plugins as binary files. By default there are total 50 plus plugins installed



```
root@ip-172-31-53-33:/usr/local/nagios/libexec# ls
check_apt      check_dig       check_flexlm        check_ifstatus  check_mailq     check_nt        check_overcr  check_sensors       check_ssmtp  check_uptime     utils.pm
check_breeze   check_disk      check_ftp           check_imap      check_mrtg      check_ntp       check_ping    check_simap         check_swap   check_users      utils.sh
check_by_ssh   check_disk_smb  check_http          check_ircd      check_mrtgtraf  check_ntp_peer  check_pop     check_smtp          check_tcp    check_wave
check_clamd    check_dns       check_icmp          check_jabber    check_nagios    check_ntp_time  check_procs   check_spop          check_time   negate
check_cluster  check_dummy     check_ide_smart     check_load      check_nntp      check_nwstat    check_real    check_ssh           check_udp    remove_perfdata
check_dhcp     check_file_age  check_ifoperstatus  check_log       check_nntps     check_oracle    check_rpc     check_ssl_validity  check_ups    urlize
```

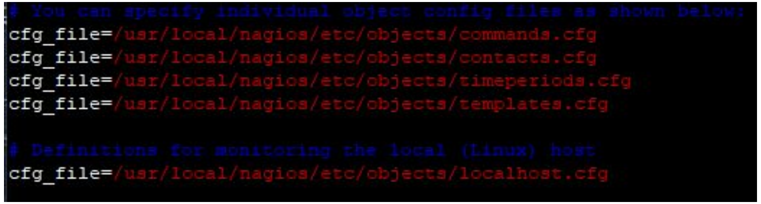
Nagios File System (cont)

- The etc directory is the main operational directory and contain nagios.cfg file and objects directory /usr/local/nagios/etc/objects/



```
root@ip-172-31-53-33:/usr/local/nagios/etc# ll
total 84
drwxrwxr-x 3 nagios nagios  4096 Jul 13 18:55 ./
drwxr-xr-x 9 root   root    4096 Jul 13 19:06 ../
-rw-rw-r-- 1 nagios nagios 13710 Jul 13 18:53 cgi.cfg
-rw-r--r-- 1 root   root      50 Jul 13 18:55 htpasswd.users
-rw-rw-r-- 1 nagios nagios 45843 Jul 13 18:53 nagios.cfg
drwxrwxr-x 2 nagios nagios  4096 Jul 13 18:53 objects/
-rw-rw---- 1 nagios nagios  1312 Jul 13 18:53 resource.cfg
```

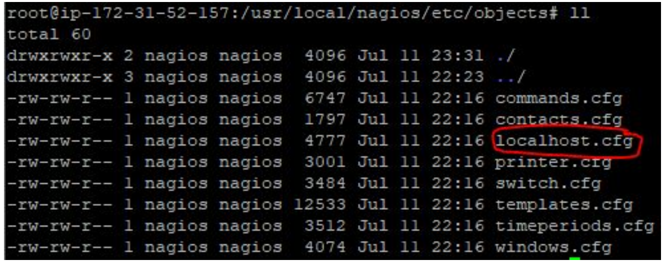
- All the object resources in Nagios are defined inside the nagios.cfg file. This file determines what kind of resource/objects can be declared on the dashboard which can be external servers (linux/windows), log files, printers, routers, switches etc



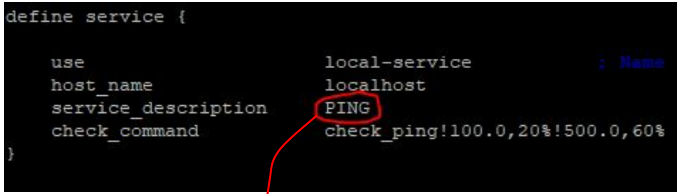
```
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
```

Nagios File System (cont)

- Each element inside the Nagios dashboard is an object and stored inside the objects directory under the extension “.cfg”. These are the configuration files. All Nagios objects reside inside these cfg files
- Go to directory: cd /usr/local/nagios/etc/objects and list all files

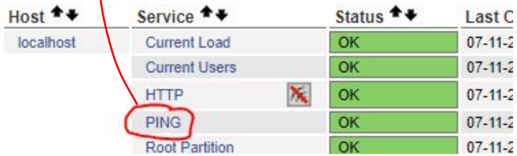


```
root@ip-172-31-52-157:/usr/local/nagios/etc/objects# ll
total 60
drwxrwxr-x 2 nagios nagios  4096 Jul 11 23:31 ./
drwxrwxr-x 3 nagios nagios  4096 Jul 11 22:23 ../
-rw-rw-r-- 1 nagios nagios  6747 Jul 11 22:16 commands.cfg
-rw-rw-r-- 1 nagios nagios  1797 Jul 11 22:16 contacts.cfg
-rw-rw-r-- 1 nagios nagios  4777 Jul 11 22:16 localhost.cfg
-rw-rw-r-- 1 nagios nagios  3001 Jul 11 22:16 printer.cfg
-rw-rw-r-- 1 nagios nagios  3484 Jul 11 22:16 switch.cfg
-rw-rw-r-- 1 nagios nagios 12533 Jul 11 22:16 templates.cfg
-rw-rw-r-- 1 nagios nagios  3512 Jul 11 22:16 timeperiods.cfg
-rw-rw-r-- 1 nagios nagios  4074 Jul 11 22:16 windows.cfg
```



```
define service {

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

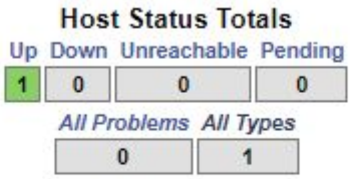
- Open the localhost.cfg and you will see the services



Host	Service	Status	Last C
localhost	Current Load	OK	07-11-2
	Current Users	OK	07-11-2
	HTTP	OK	07-11-2
	PING	OK	07-11-2
	Root Partition	OK	07-11-2

Service States

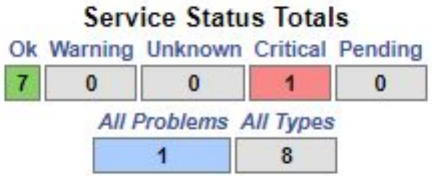
- A host status can have 4 states
  - Up State
  - Down State
  - Unreachable State
  - Pending State
- A service status can have 5 states
  - OK State
  - Warning State
  - Unknown State
  - Critical State
  - Pending State
- We call plugins and designate values for the status check



Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

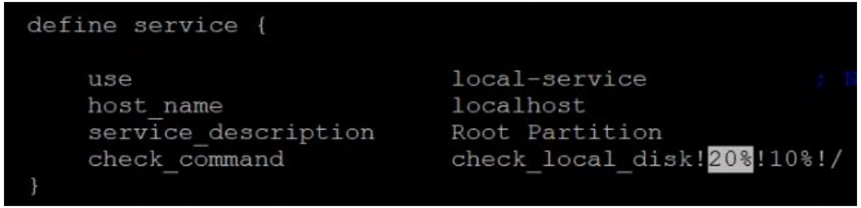
All Problems	All Types
0	1



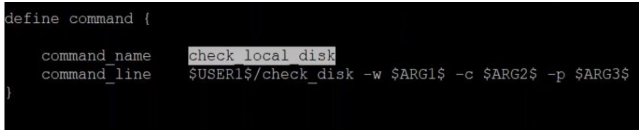
Service Status Totals

Ok	Warning	Unknown	Critical	Pending
7	0	0	1	0

All Problems	All Types
1	8



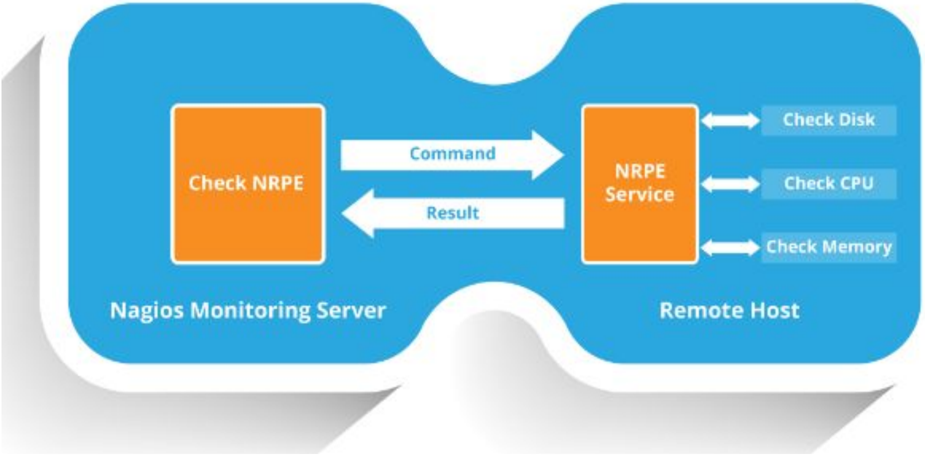
```
define service {

    use                         local-service
    host_name                   localhost
    service_description         Root Partition
    check_command               check_local_disk!20%!10%!/
}
```



```
define command {

    command_name     check_local_disk
    command_line     $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
}
```

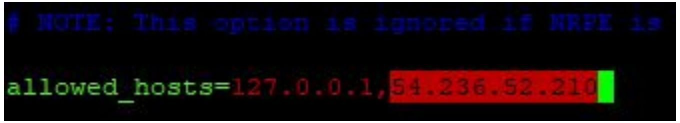
Nagios Remote Plugin Executor (NRPE)

- NRPE allows you to remotely execute Nagios plugins on other Linux machines. This allows you to monitor remote machine metrics such as disk usage, CPU load etc.
- It can communicate with some of the Windows agent addons, so you can execute scripts and check metrics on remote windows machine as well.



Monitoring a Remote Server

- Connect to your remote machine (host) as root and install NRPE and Nagios plugins
  - *sudo -i*
  - *apt-get update*
  - *apt-get install -y nagios-nrpe-server nagios-plugins*
- Modify the NRPE configuration file to accept the connection from the Nagios server, Edit the /etc/nagios/nrpe.cfg file
  - *cd /etc/nagios/*
  - *vim nrpe.cfg*



```
# NOTE: This option is ignored if NRPE is

allowed_hosts=127.0.0.1,54.236.52.210
```

- Add the Nagios servers IP address, separated by comma, then save and exit
  - Scroll down and add server public IP address
  - Add check_swap and check_root commands under the command lines, save and exit:

    ```
    command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 30% -c 10%
    command[check_root]=/usr/lib/nagios/plugins/check_root -w 70% -c 80%
    ```

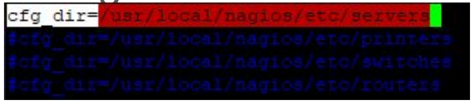
- Test Nagios Check. The output will show PROCS OK
  - */usr/lib/nagios/plugins/check_procs -w 150 -c 200*
  - Restart NRPE service: *systemctl restart nagios-nrpe-server*

Monitoring a Remote Server (cont)

- Enable Firewall and IP table inside the server machine by running these commands. If you are using EC2 instance then first you need to define a custom TCP port 5666 in both EC2 machines, server and host :
  - *apt install firewalld*
  - *firewall-cmd --permanent --add-port=5666/tcp*
  - *firewall-cmd --reload*
  - *iptables -I INPUT -p tcp --dport 5666 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT*
  - *iptables -I OUTPUT -p tcp --sport 5666 -m conntrack --ctstate ESTABLISHED -j ACCEPT*
- Switch to Nagios server and install NRPE plugin
  - *apt install -y nagios-nrpe-plugin*

Monitoring a Remote Server (cont)

- Edit configuration files in the Nagios server for it to enable monitoring
  - First edit nagios.cfg file: *vim /usr/local/nagios/etc/nagios.cfg*
  - Uncomment line cfg_dir=/usr/local/nagios/etc/servers
  - Create a new directory called server in etc: *mkdir /usr/local/nagios/etc/servers*



```
cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

- Now it's time to configure the Nagios server to monitor the remote client machine, and You'll need to create a command definition in Nagios object configuration file to use the check_nrpe plugin:
  - *vim /usr/local/nagios/etc/objects/commands.cfg*
  - Scroll to the bottom of the file and paste the following:

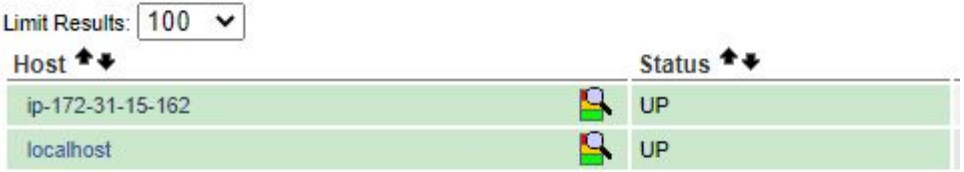
```
# .check_nrpe. command definition
define command{
command_name check_nrpe
command_line /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -t 30 -c
$ARG1$
}
```

Monitoring a Remote Server (cont)

- Now need to add the Host server to the Nagios server. Create a client configuration file to define the host and service definitions of remote Linux host.
  - *vim /usr/local/nagios/etc/servers/hostconfig.cfg*
  - Step 1: copy and paste the content inside the file (copy from Google Doc)
  - Step 2: type 'hostname' to print hostname in host server
  - Step 3: copy the hostname and replace in the file for host, hostgroup and service
  - Step 4: in host definition replace IP address with public IP of host server
  - Step 5: after making changes save and exit
  - Verify Nagios for any error: The output will show PROCS OK
  - */usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg*
  - To initiate change restart Nagios service: *systemctl restart nagios*

Check Result in Dashboard

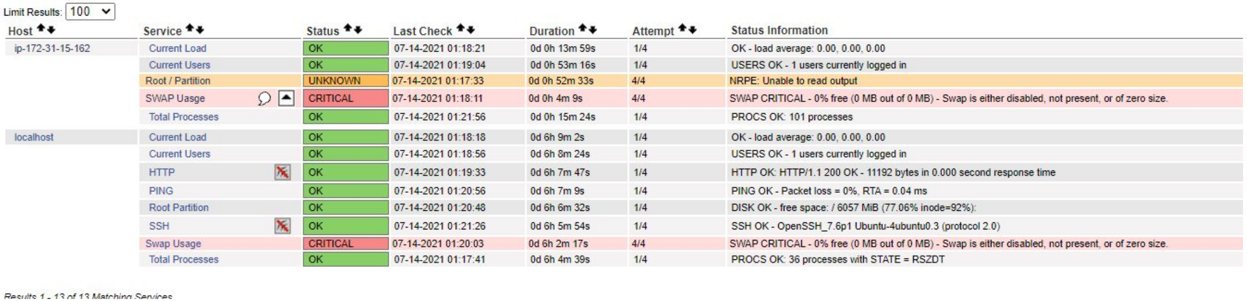
- Wait and check your Nagios dashboard and you should get a new host in your Host list. Wait for another minute for new host “ip-172-31-15-162” state to become OK



Limit Results: 100

Host	Status
ip-172-31-15-162	UP
localhost	UP

- Check Services and you should see most of the services up and running



Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ip-172-31-15-162	Current Load	OK	07-14-2021 01:18:21	0d 0h 13m 59s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	07-14-2021 01:19:04	0d 0h 53m 16s	1/4	USERS OK - 1 users currently logged in
	Root / Partition	UNKNOWN	07-14-2021 01:17:33	0d 0h 52m 33s	4/4	NRPE: Unable to read output
	SWAP Uasge	CRITICAL	07-14-2021 01:18:11	0d 0h 4m 9s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	07-14-2021 01:21:56	0d 0h 15m 24s	1/4	PROCS OK: 101 processes
localhost	Current Load	OK	07-14-2021 01:18:18	0d 6h 9m 2s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	07-14-2021 01:18:56	0d 6h 8m 24s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	07-14-2021 01:19:33	0d 6h 7m 47s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.000 second response time
	PING	OK	07-14-2021 01:20:56	0d 6h 7m 9s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	07-14-2021 01:20:48	0d 6h 6m 32s	1/4	DISK OK - free space: / 6057 MiB (77.06% inode=92%):
	SSH	OK	07-14-2021 01:21:26	0d 6h 5m 54s	1/4	SSH OK - OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 (protocol 2.0)
	Swap Usage	CRITICAL	07-14-2021 01:20:03	0d 6h 2m 17s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	07-14-2021 01:17:41	0d 6h 4m 39s	1/4	PROCS OK: 36 processes with STATE = RSZDT

Results 1 - 13 of 13 Matching Services

Thank You!