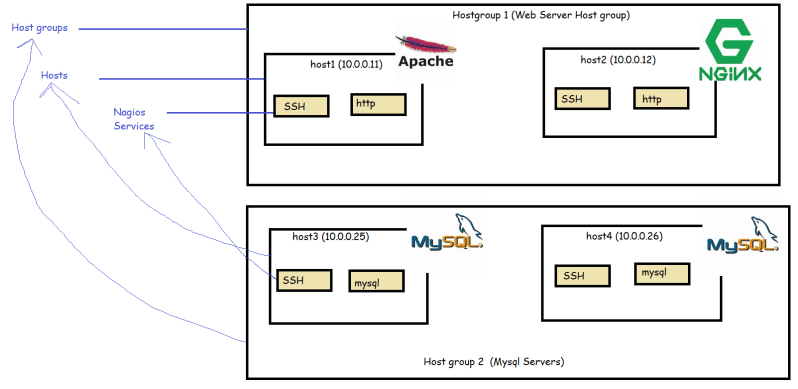
22/Jun/2021

DAY-1 NAG-1

System Monitoring with Nagios

**Nagios** is the most popular, open source, powerful monitoring system. It enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes. Nagios has the capability of monitoring application, services, entire IT infrastructure.

* Nagios is a popular tool for System Monitoring
* System monitoring in Nagios is split into two categories
  + hosts: Represent physical or virtual device (server, router, printer)
  + services: Represents particular functionality of hosts (ssh, http) 
* Host groups are logical collection of hosts
* In Nagios when we perform checks, it uses four states
  + Ok
  + Warning
  + Critical
  + UnKnown
* To collect the States for the Checks Nagios uses plugins
* Nagios performs all of its checks using plugins. These are external components for which Nagios passes information on what should be checked and what are warning, critical and ok limits are.
* Plugins are responsible for performing the checks and analyzing results. The output from check is status.
* Nagios provides set of standard plugins that allow performance checks for almost all the services. Moreover if you need to perform a specific check, Nagios provides an approach to write your own plugins in any language.

Main features

* Nagios Main strength is flexibility, it has a mechanism to react automatically to problems and a powerful notification system.
* For this flexibility is based on object definition system
* The object definitions are based on few types of objects
  + Commands
  + Time periods
  + Hosts and Host Groups
  + Services
  + Contacts and Contact Groups
  + Notifications

Soft and hard states

* A service is down for a very short time or the test is temporarily failed etc are the normal failures
* When a previous states of check is different from the current one (Critical), Nagios will re-test the host or service for a couple of times to make sure change is permanent. Nagios assumes the new result is soft-state, after additional test have verified that the new state is permanent, it is considered to be hard state.

Nagios Offerings

* Nagios Core: Free to use open source version of Nagios [Refer Here](https://www.nagios.org/projects/nagios-core/)

(<https://www.nagios.org/projects/nagios-core/>)

* Nagios XI: Enterprise Network Monitoring Solution (Paid) [Refer Here](https://www.nagios.com/products/nagios-xi/?__hstc=118811158.21a2ab880911a6b2ca5e817cb93d443a.1624330147597.1624330147597.1624330147597.1&__hssc=118811158.1.1624330147598&__hsfp=645633051)

<https://www.nagios.com/products/nagios-xi/?__hstc=118811158.21a2ab880911a6b2ca5e817cb93d443a.1624330147597.1624330147597.1624330147597.1&__hssc=118811158.1.1624330147598&__hsfp=645633051>

Installing Nagios 4.x

* [Refer Here](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart.html) for the installation guide (<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart.html>)
* For ubuntu specific installation steps [Refer Here](https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html#Ubuntu) (<https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html#Ubuntu>)

DAY-2 NAG-2 **23/Jun/2021**

## Configuring Nagios

* Nagios stores its configuration in a separate directory. Usually in /etc/nagios or /usr/local/nagios/etc
* The main configuration file is called **nagios.cfg**.
* The syntax in the main nagios configuration file

<parameter>=<value>

Configuration options:

<ReferHere> (<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/configma>in.html) for the official docs

* The Nagios option resource file defines a file to store the user variables. This file can be used to store additional information that can be accessed in all object definitions. These usually contain sensitive data as they can only be used in object definitions

## Understanding macro definitions

| **Macro** | **Description** |
| --- | --- |
| HOSTNAME | Short, unique name of the host; maps to host\_name directive in the host object |
| HOSTADDRESS | The IP or hostname of the hosts; maps to address directive in host object |
| HOSTDISPLAYNAME | Description of host; Maps to the alias directive in host object |
| HOSTSTATE | The current state of host |
| HOSTGROUPNAMES | Short names of all host groups a host belongs to (Comma separated) |
| LASTHOSTCHECK | The date and time of last check of the host (in UNIX timestamp) |
| LASTHOSTSTATE | The last known state of host |
| SERVICEDESC | Description of service |
| SERVICESTATE | The current state of Service |
| CONTACTNAME | unique name of contact |
| CONTACTALIAS | Description of Contact |
| CONTACTEMAIL | The email address of contact |
| CONTACTGROUPNAME | Short names of all contact groups (comma separated) |

* For the complete list of macros [Refer Here](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/macrolist.html) (<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/macrolist.html>)

## Configuring hosts

* <ReferHere> (<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/objectdefinitions.html#host>) for the official documentation of host configuration
* Sample

define host {

host\_name node1

alias Node 1 AWS

address 172.31.29.140

check\_command check-host-alive

max\_check\_attempts 5

check\_interval 5

retry\_interval 1

check\_period 24x7

notification\_period 24x7

notification\_interval 120

notification\_options d,u,r

contact\_groups admins

}

* Things to know for Nagios configuration
  + Object Definitions
    - Host
    - Host Group
    - Services
    - Commands
    - Contact
    - Templates
* DAY-3 NAG-3 24/June/2021

## Configuring Nagios object definitions

* [Refer Here](https://github.com/asquarezone/ElasticStackZone/commit/ec6031253c62262af9ec839510b6d3a72bd5efff) for the default Nagios configuration (<https://github.com/asquarezone/ElasticStackZone/commit/ec6031253c62262af9ec839510b6d3a72bd5efff>)

## Understanding how checks work

* Nagios requires all plugins to follow a specific, easy to follow behaviour
* Nagios relies on exit codes of the Nagios plugins

| **Exit code** | **Status** | **Description** |
| --- | --- | --- |
| 0 | OK | Working correctly |
| 1 | WARNING | Working but needs attention |
| 2 | CRITICAL | Not working or requires attention |
| 3 | UNKNOWN | Plugin was unable to determine the status of host or service |

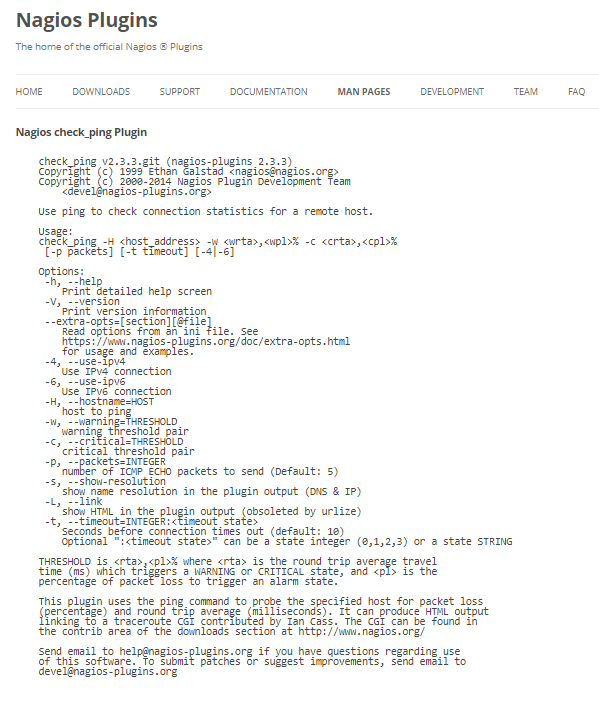
* Standard output from command is not parsed by Nagios and is usually formatted in the following way

PLUGIN STATUS - status description

* Nagios plugins use options for their configuration
  + -h, –help: This provides help
  + -V, –version: This prints the version
  + -v , –verbose: This prints the detailed information on what plugin is doing
  + -t , –timeout
  + -w, –warning
  + -c, –critical
  + -H, –hostname
  + -4, –use-ipv4
  + -6, –use-ipv6
* All the network related plugins have the following standard option
  + -p, –port
  + -w, –warning
  + -c, –critical
  + -s, –send: This provides the string that will be sent to server
  + -e, –expect: This provides the string that should be sent back from server
  + -q, –quit: This provides the string to be sent to close the connection
  + -S, –ssl
  + -D, –certificate

## Testing the connection to remote host

* For this we have check\_ping plugin [Refer Here](http://nagios-plugins.org/doc/man/check_ping.html) for official documentation (<http://nagios-plugins.org/doc/man/check_ping.html>)



* Creating the command for the host-alive

define command {

command\_name check-host-alive

command\_line $USER1$/check\_ping -H $HOSTADDRESS$ -w 2000,50% -c 5000,60% -p 5

}

## Testing the connectivity using TCP and UDP

* [Refer Here](https://www.monitoring-plugins.org/doc/man/check_tcp.html) for the official documentation(<https://www.monitoring-plugins.org/doc/man/check_tcp.html>)

|  |
| --- |
| The check\_tcp Plugin check\_tcp v2.3git (monitoring-plugins 2.3git)  Copyright (c) 1999 Ethan Galstad <nagios@nagios.org>  Copyright (c) 1999-2008 Monitoring Plugins Development Team  <devel@monitoring-plugins.org>  This plugin tests TCP connections with the specified host (or unix socket).  Usage:  check\_tcp -H host -p port [-w <warning time>] [-c <critical time>] [-s <send string>]  [-e <expect string>] [-q <quit string>][-m <maximum bytes>] [-d <delay>]  [-t <timeout seconds>] [-r <refuse state>] [-M <mismatch state>] [-v] [-4|-6] [-j]  [-D <warn days cert expire>[,<crit days cert expire>]] [-S <use SSL>] [-E]  Options:  -h, --help  Print detailed help screen  -V, --version  Print version information  --extra-opts=[section][@file]  Read options from an ini file. See  https://www.monitoring-plugins.org/doc/extra-opts.html  for usage and examples.  -H, --hostname=ADDRESS  Host name, IP Address, or unix socket (must be an absolute path)  -p, --port=INTEGER  Port number (default: none)  -4, --use-ipv4  Use IPv4 connection  -6, --use-ipv6  Use IPv6 connection  -E, --escape  Can use \n, \r, \t or \\ in send or quit string. Must come before send or quit option  Default: nothing added to send, \r\n added to end of quit  -s, --send=STRING  String to send to the server  -e, --expect=STRING  String to expect in server response (may be repeated)  -A, --all  All expect strings need to occur in server response. Default is any  -q, --quit=STRING  String to send server to initiate a clean close of the connection  -r, --refuse=ok|warn|crit  Accept TCP refusals with states ok, warn, crit (default: crit)  -M, --mismatch=ok|warn|crit  Accept expected string mismatches with states ok, warn, crit (default: warn)  -j, --jail  Hide output from TCP socket  -m, --maxbytes=INTEGER  Close connection once more than this number of bytes are received  -d, --delay=INTEGER  Seconds to wait between sending string and polling for response  -D, --certificate=INTEGER[,INTEGER]  Minimum number of days a certificate has to be valid.  1st is #days for warning, 2nd is critical (if not specified - 0).  -S, --ssl  Use SSL for the connection.  --sni=STRING  SSL server\_name  -w, --warning=DOUBLE  Response time to result in warning status (seconds)  -c, --critical=DOUBLE  Response time to result in critical status (seconds)  -t, --timeout=INTEGER  Seconds before connection times out (default: 10)  -v, --verbose  Show details for command-line debugging (output may be truncated by  the monitoring system)  Send email to help@monitoring-plugins.org if you have questions regarding  use of this software. To submit patches or suggest improvements, send email  to devel@monitoring-plugins.org |

## Testing the websites

* [Refer Here](https://www.monitoring-plugins.org/doc/man/check_http.html) for the official documentation of check\_http plugin(https://www.monitoring-plugins.org/doc/man/check\_http.html)

|  |
| --- |
| The check\_http Plugin check\_http v2.3git (monitoring-plugins 2.3git)  Copyright (c) 1999 Ethan Galstad <nagios@nagios.org>  Copyright (c) 1999-2013 Monitoring Plugins Development Team  <devel@monitoring-plugins.org>  This plugin tests the HTTP service on the specified host. It can test  normal (http) and secure (https) servers, follow redirects, search for  strings and regular expressions, check connection times, and report on  certificate expiration times.  Usage:  check\_http -H <vhost> | -I <IP-address> [-u <uri>] [-p <port>]  [-J <client certificate file>] [-K <private key>]  [-w <warn time>] [-c <critical time>] [-t <timeout>] [-L] [-E] [-a auth]  [-b proxy\_auth] [-f <ok|warning|critcal|follow|sticky|stickyport>]  [-e <expect>] [-d string] [-s string] [-l] [-r <regex> | -R <case-insensitive regex>]  [-P string] [-m <min\_pg\_size>:<max\_pg\_size>] [-4|-6] [-N] [-M <age>]  [-A string] [-k string] [-S <version>] [--sni]  [-T <content-type>] [-j method]  check\_http -H <vhost> | -I <IP-address> -C <warn\_age>[,<crit\_age>]  [-p <port>] [-t <timeout>] [-4|-6] [--sni]  In the first form, make an HTTP request.In the second form, connect to the server and check the TLS certificate.NOTE: One or both of -H and -I must be specified  Options:  -h, --help  Print detailed help screen  -V, --version  Print version information  --extra-opts=[section][@file]  Read options from an ini file. See  https://www.monitoring-plugins.org/doc/extra-opts.html  for usage and examples.  -H, --hostname=ADDRESS  Host name argument for servers using host headers (virtual host)  Append a port to include it in the header (eg: example.com:5000)  -I, --IP-address=ADDRESS  IP address or name (use numeric address if possible to bypass DNS lookup).  -p, --port=INTEGER  Port number (default: 80)  -4, --use-ipv4  Use IPv4 connection  -6, --use-ipv6  Use IPv6 connection  -S, --ssl=VERSION[+]  Connect via SSL. Port defaults to 443. VERSION is optional, and prevents  auto-negotiation (2 = SSLv2, 3 = SSLv3, 1 = TLSv1, 1.1 = TLSv1.1,  1.2 = TLSv1.2). With a '+' suffix, newer versions are also accepted.  --sni  Enable SSL/TLS hostname extension support (SNI)  -C, --certificate=INTEGER[,INTEGER]  Minimum number of days a certificate has to be valid. Port defaults to 443  (when this option is used the URL is not checked.)  -J, --client-cert=FILE  Name of file that contains the client certificate (PEM format)  to be used in establishing the SSL session  -K, --private-key=FILE  Name of file containing the private key (PEM format)  matching the client certificate  -e, --expect=STRING  Comma-delimited list of strings, at least one of them is expected in  the first (status) line of the server response (default: HTTP/1.)  If specified skips all other status line logic (ex: 3xx, 4xx, 5xx processing)  -d, --header-string=STRING  String to expect in the response headers  -s, --string=STRING  String to expect in the content  -u, --url=PATH  URL to GET or POST (default: /)  -P, --post=STRING  URL encoded http POST data  -j, --method=STRING (for example: HEAD, OPTIONS, TRACE, PUT, DELETE, CONNECT, CONNECT:POST)  Set HTTP method.  -N, --no-body  Don't wait for document body: stop reading after headers.  (Note that this still does an HTTP GET or POST, not a HEAD.)  -M, --max-age=SECONDS  Warn if document is more than SECONDS old. the number can also be of  the form "10m" for minutes, "10h" for hours, or "10d" for days.  -T, --content-type=STRING  specify Content-Type header media type when POSTing  -l, --linespan  Allow regex to span newlines (must precede -r or -R)  -r, --regex, --ereg=STRING  Search page for regex STRING  -R, --eregi=STRING  Search page for case-insensitive regex STRING  --invert-regex  Return CRITICAL if found, OK if not  -a, --authorization=AUTH\_PAIR  Username:password on sites with basic authentication  -b, --proxy-authorization=AUTH\_PAIR  Username:password on proxy-servers with basic authentication  -A, --useragent=STRING  String to be sent in http header as "User Agent"  -k, --header=STRING  Any other tags to be sent in http header. Use multiple times for additional headers  -E, --extended-perfdata  Print additional performance data  -B, --show-body  Print body content below status line  -L, --link  Wrap output in HTML link (obsoleted by urlize)  -f, --onredirect=<ok|warning|critical|follow|sticky|stickyport>  How to handle redirected pages. sticky is like follow but stick to the  specified IP address. stickyport also ensures port stays the same.  -m, --pagesize=INTEGER<:INTEGER>  Minimum page size required (bytes) : Maximum page size required (bytes)  -w, --warning=DOUBLE  Response time to result in warning status (seconds)  -c, --critical=DOUBLE  Response time to result in critical status (seconds)  -t, --timeout=INTEGER  Seconds before connection times out (default: 10)  -v, --verbose  Show details for command-line debugging (output may be truncated by  the monitoring system)  Notes:  This plugin will attempt to open an HTTP connection with the host.  Successful connects return STATE\_OK, refusals and timeouts return STATE\_CRITICAL  other errors return STATE\_UNKNOWN. Successful connects, but incorrect response  messages from the host result in STATE\_WARNING return values. If you are  checking a virtual server that uses 'host headers' you must supply the FQDN  (fully qualified domain name) as the [host\_name] argument.  This plugin can also check whether an SSL enabled web server is able to  serve content (optionally within a specified time) or whether the X509  certificate is still valid for the specified number of days.  Please note that this plugin does not check if the presented server  certificate matches the hostname of the server, or if the certificate  has a valid chain of trust to one of the locally installed CAs.  Examples:  CHECK CONTENT: check\_http -w 5 -c 10 --ssl -H www.verisign.com  When the 'www.verisign.com' server returns its content within 5 seconds,  a STATE\_OK will be returned. When the server returns its content but exceeds  the 5-second threshold, a STATE\_WARNING will be returned. When an error occurs,  a STATE\_CRITICAL will be returned.  CHECK CERTIFICATE: check\_http -H www.verisign.com -C 14  When the certificate of 'www.verisign.com' is valid for more than 14 days,  a STATE\_OK is returned. When the certificate is still valid, but for less than  14 days, a STATE\_WARNING is returned. A STATE\_CRITICAL will be returned when  the certificate is expired.  CHECK CERTIFICATE: check\_http -H www.verisign.com -C 30,14  When the certificate of 'www.verisign.com' is valid for more than 30 days,  a STATE\_OK is returned. When the certificate is still valid, but for less than  30 days, but more than 14 days, a STATE\_WARNING is returned.  A STATE\_CRITICAL will be returned when certificate expires in less than 14 days  CHECK SSL WEBSERVER CONTENT VIA PROXY USING HTTP 1.1 CONNECT:  check\_http -I 192.168.100.35 -p 80 -u https://www.verisign.com/ -S -j CONNECT -H www.verisign.com  all these options are needed: -I <proxy> -p <proxy-port> -u <check-url> -S(sl) -j CONNECT -H <webserver>  a STATE\_OK will be returned. When the server returns its content but exceeds  the 5-second threshold, a STATE\_WARNING will be returned. When an error occurs,  a STATE\_CRITICAL will be returned. By adding a colon to the method you can set the method used  inside the proxied connection: -j CONNECT:POST  Send email to help@monitoring-plugins.org if you have questions regarding  use of this software. To submit patches or suggest improvements, send email  to devel@monitoring-plugins.org |

## Testing the databases

* MySQL
  + check\_mysql [Refer Here](https://www.monitoring-plugins.org/doc/man/check_mysql.html) (<https://www.monitoring-plugins.org/doc/man/check_mysql.html>)
  + check\_mysql\_query [Refer Here](https://www.monitoring-plugins.org/doc/man/check_mysql_query.html) (<https://www.monitoring-plugins.org/doc/man/check_mysql_query.html>)
* PostgreSQL
  + check\_pgsql [Refer Here](https://www.monitoring-plugins.org/doc/man/check_pgsql.html) (<https://www.monitoring-plugins.org/doc/man/check_pgsql.html>)
* Oracle: [Refer Here](https://www.monitoring-plugins.org/doc/man/check_oracle.html)(<https://www.monitoring-plugins.org/doc/man/check_oracle.html>)
* For installing plugins [Refer Here](https://exchange.nagios.org/directory/Plugins) (<https://exchange.nagios.org/directory/Plugins>)
* Now we have a basic idea on how to use plugins to fetch the basic connectivity
* [Refer Here](https://github.com/asquarezone/ElasticStackZone/commit/f776b1a3adba6037060b103d71ea31818698b5cf) for sample commands written (<https://github.com/asquarezone/ElasticStackZone/commit/f776b1a3adba6037060b103d71ea31818698b5cf>)



## Area of concern

* If we need to alert/notify the user on some details such as
  + free disk space
  + process
  + CPU stress

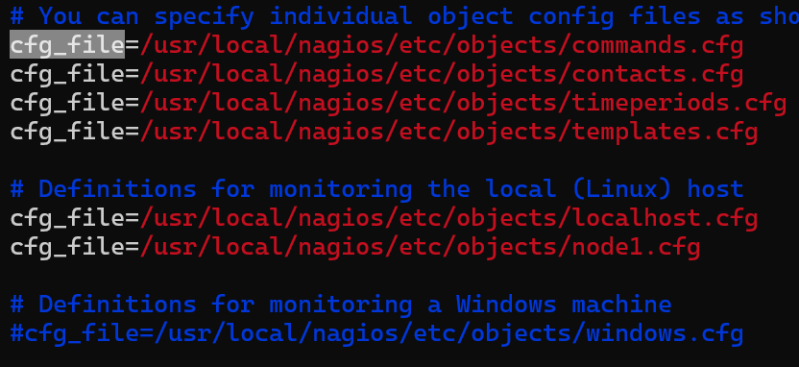
DAY-3 NAG-3

25/Jun/2021

Configuring Nagios

* Create Templates for the below so that we can add all the generic information inside the template
  + host definitions
  + service definitions
* Create Commands and define all the necessary commands for performing checks
* Create hostgroups and add different hosts to host groups (allservers, webservers, dbservers )
* Now create service defintions where we define checks using commands defined at the host group level

Exercise

* We have default nagios configuration in /usr/local/nagios/etc/objects
* Now try to make that default nagios configuration broken down into
  + nagioscfg
    - commands
    - timeperiods
    - templates
    - contacts
    - hosts
    - services
* Change the nagios.cfg to use cfg\_dir instead of cfg\_file

DAY-4 NAG-4

26/Jun/2021

25/Jun/2021

Configuring Nagios

* Create Templates for the below so that we can add all the generic information inside the template
  + host definitions
  + service definitions
* Create Commands and define all the necessary commands for performing checks
* Create hostgroups and add different hosts to host groups (allservers, webservers, dbservers )
* Now create service defintions where we define checks using commands defined at the host group level

Exercise

* We have default nagios configuration in /usr/local/nagios/etc/objects
* Now try to make that default nagios configuration broken down into
  + nagioscfg
    - commands
    - timeperiods
    - templates
    - contacts
    - hosts
    - services
* Change the nagios.cfg to use cfg\_dir instead of cfg\_file 