Configuration and deployment of Nagios monitoring tool

Prerequisites

- Nagios Core installed.
- Slack account with access

Nagios installation

Prerequisites

- Apache
  - $ sudo apt install apache2
  - $ sudo ufw allow 'Apache' / sudo ufw allow 80
  - $ You may check: http://your_server_ip

- PHP installed.
  - $ sudo apt install php libapache2-mod-php php-mysql

Step 1 — Installing Nagios 4.4.6

- Install required packages:
  - $ sudo apt install autoconf gcc make unzip libgd-dev libmcrypt-dev libssl-dev dc snmp libnet-snmp-perl gettext
- Download nagios 4.4.6 to home directory :
  - $ curl -L -O https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.4.tar.gz
  - $ tar zxf nagios-4.4.6.tar.gz
- Run configure script:
  - $ cd nagioscore-nagios-4.4.6
  - $ ./configure --with-httpd-conf=/etc/apache2/sites-enabled --with-mail=/usr/sbin/sendmail
- Compile Nagios
  - $ make all
- Create nagios user and group:
  - $ sudo make install-groups-users

- Install Nagios binary files, service files, and its sample configuration files:
    - $ sudo make install
    - $ sudo make install-daemoninit
    - $ sudo make install-commandmode
    - $ sudo make install-config
- install the Apache configuration files and configure its settings:
    - $ sudo make install-webconf
- Enable the Apache rewrite and cgi modules:
    - $ sudo a2enmod rewrite
    - $ sudo a2enmod cgi
- add the web server user, **www-data**, to the **nagios** group(enables issuing of external commands via the web interface to Nagios):
    - $ sudo usermod -a -G nagios www-data
- create an admin user called **nagiosadmin** that can access the Nagios web interface:
    - $ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
- Restart apache
- Install Nagios plugins from home directory:
    - curl -L -O https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
    - tar zxf nagios-plugins-2.2.1.tar.gz
    - cd nagios-plugins-2.2.1
    - ./configure
- Build plugins:
    - make
    - sudo make install
- NB: If you are to monitor remote hosts
    - Download check_nrpe Plugin to your home directory,configure and build it:
        - curl -L -O https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz
        - tar zxf nrpe-3.2.1.tar.gz
        - cd nrpe-3.2.1
        - ./configure
        - make check_nrpe
        - sudo make install-plugin
        - make check_nrpe

        - sudo make install-plugin
- Perform initial configurations:
    - Open /usr/local/nagios/etc/nagios.cfg and uncomment:
        - cfg_dir=/usr/local/nagios/etc/servers
    - Create /usr/local/nagios/etc/objects/contacts.cfg and add:

```
define contact{
        contact_name        nagiosadmin      ; Short name of user
        use                 generic-contact    ; Inherit default values from
generic-contact template (defined above)
        alias               Nagios Admin   ; Full name of user
        email               your_email@your_domain.com       ; <<*****
CHANGE THIS TO YOUR EMAIL ADDRESS ******
```

    - Open /usr/local/nagios/etc/objects/commands.cfg and add:

```
define command{
      command_name check_nrpe
      command_line $USER1$/check_nrpe -H $HOSTADDRESS$
-c $ARG1$
}
```

- Restart nagios

Step 2 — Installing the Nagios Plugins

Install the plugins bundle, download it to your home directory with curl:

- $ cd ~
- $ curl -L -O https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz

Extract the NRPE archive and navigate into the extracted directory:

- $ tar zxf nagios-plugins-2.2.1.tar.gz
- $ cd nagios-plugins-2.2.1

Now run:

- $ ./configure

Now build and install the plugins:

- $ make
- $ sudo make install

Step 3 — Installing the check_nrpe Plugin

Nagios monitors remote hosts using the Nagios Remote Plugin Executor, or NRPE. It consists of two pieces:

- The check_nrpe plugin that the Nagios server uses.
- The NRPE daemon, which runs on the remote hosts and sends data to the Nagios server.

Download to home directory:

- $ cd ~
- $ curl -L -O https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz

Extract the NRPE archive and navigate to its directory:

- $ tar zxf nrpe-3.2.1.tar.gz
- $ cd nrpe-3.2.1

Configure the check_nrpe plugin:

- $ ./configure

Now build and install check_nrpe plugin:

- $ make check_nrpe
- $ sudo make install-plugin

Step 4 — Configuring Nagios

Open nagios.cfg and uncomment the line below:

- $ sudo nano /usr/local/nagios/etc/nagios.cfg

...
cfg_dir=/usr/local/nagios/etc/servers
…

Create the directory that will store the configuration file for the server that will be monitored:

- $ sudo mkdir /usr/local/nagios/etc/servers

Open the Nagios contacts configuration:

- $ sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```
  GNU nano 4.8                    /usr/local/nagios/etc/objects/contacts.cfg                    Modified

define contact {

    contact_name          nagiosadmin           ; Short name of user
    use                   generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias                 Nagios Admin          ; Full name of user
    email                 musomoletsane@gmail.com ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}
```

- $ sudo nano /usr/local/nagios/etc/objects/commands.cfg

```
define command{
        command_name check_nrpe
        command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  ^Q Where Was
```

Now start Nagios and enable it to start when the server boots:

- $ sudo systemctl start nagios

<u>Finally:</u>

Open web browser, and go to Nagios server:
http://192.168.19.140/nagios/

Creating Monitoring Services in Nagios

Create a folder under /usr/local/nagios/etc/objects to store related configurations by running the following command:

- $ sudo mkdir /usr/local/nagios/etc/objects/example-dir

Then store Nagios commands for check_nagios in a file named commands.cfg. Create it for editing:

- $ sudo nano /usr/local/nagios/etc/objects/example-dir/commands.cfg

For example:

```
  GNU nano 4.8                    /usr/local/nagios/etc/objects/postgresql/commands.cfg
define command {
    command_name          check_postgres_connection
    command_line          /usr/local/nagios/libexec/check_postgres_connection --dbservice=$ARG1$
}

define command {
    command_name          check_postgres_database_size
    command_line          /usr/local/nagios/libexec/check_postgres_database_size --dbservice=$ARG1$ --critical='$ARG2$'
}

define command {
    command_name          check_postgres_locks
    command_line          /usr/local/nagios/libexec/check_postgres_locks --dbservice=$ARG1$
}

define command {
    command_name          check_postgres_backends
    command_line          /usr/local/nagios/libexec/check_postgres_backends --dbservice=$ARG1$
}
```

Save the file

Define the host and its monitoring services in a file named *services.cfg*

- $ sudo nano /usr/local/nagios/etc/objects/example-dir/services.cfg

For example:

```
define host {
        use                     linux-server
        host_name               postgres
        check_command           check_postgres_connection!managed-db
}

define service {
        use                     generic-service
        host_name               postgres
        service_description     PostgreSQL Connection
        check_command           check_postgres_connection!managed-db
        notification_options    w,u,c,r,f,s
}

define service {
        use                     generic-service
        host_name               postgres
        service_description     PostgreSQL Database Size
        check_command           check_postgres_database_size!managed-db!8176495
        notification_options    w,u,c,r,f,s
}

define service {
        use                     generic-service
        host_name               postgres
        service_description     PostgreSQL Locks
        check_command           check_postgres_locks!managed-db
        notification_options    w,u,c,r,f,s
}

define service {
        use                     generic-service
        host_name               postgres
        service_description     PostgreSQL Backends
                                                        [ Read 37 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go
```

Save the file

Explicitly tell Nagios to read config files from this new directory, by editing the general Nagios config file:

- $ sudo nano /usr/local/nagios/etc/nagios.cfg

Add the following highlighted line:

...

cfg_dir=/usr/local/nagios/etc/objects/example-dir

cfg_dir=/usr/local/nagios/etc/servers

...

Save and close the file.

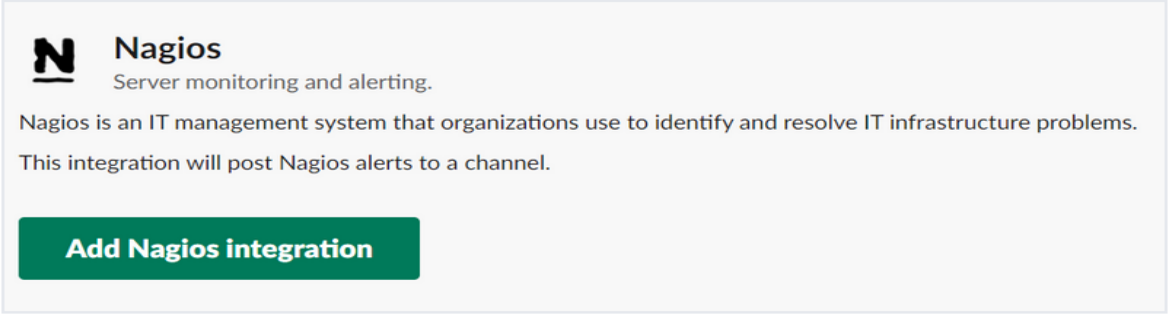
Before restarting Nagios, check the validity of the configuration by running the following command:

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

- $ sudo systemctl restart nagios

Configuring Slack Alerting

Head over to the [Nagios app] in the Slack App Directory and press on **Add Configuration**. Then, find a page for adding the Nagios Integration:



Press on **Add Nagios Integration**. When the page loads, scroll down and take note of the token, because you'll need it further on.

Install the required Perl prerequisites by running the following command:

- $ sudo apt install libwww-perl libcrypt-ssleay-perl -y

Then, download the plugin to Nagios plugin directory:

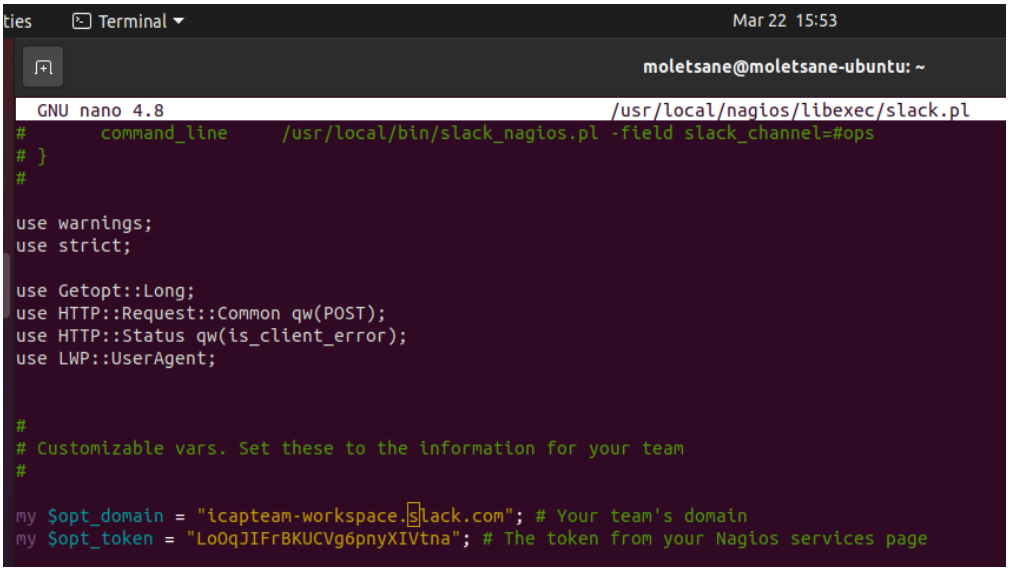
- $ sudo curl https://raw.githubusercontent.com/tinyspeck/services-examples/master/nagios.pl -o slack.pl

Make it executable:

- $ sudo chmod +x slack.pl

Edit it to connect to your workspace using the token from Slack:

- $ sudo nano slack.pl



The script will now be able to send proper requests to Slack, which is tested by running the following command:

- $ *./slack.pl -field slack_channel=#icapteam-workspace -field HOSTALIAS="Test Host" -field HOSTSTATE="UP" -field HOSTOUTPUT="Host is UP" -field NOTIFICATIONTYPE="RECOVERY"*

Create a contact for Slack and two commands that will send messages to it. Store this config in a file named slack.cfg, in the same folder as the previous config files:

- $ sudo nano /usr/local/nagios/etc/objects/postgresql/slack.cfg

```
  GNU nano 4.8                                  /usr/local/nagios/etc/objects/postgresql/slack.cf
define contact {
        contact_name                            slack
        alias                                   Slack
        service_notification_period             24x7
        host_notification_period                24x7
        service_notification_options            w,u,c,f,s,r
        host_notification_options               d,u,r,f,s
        service_notification_commands           notify-service-by-slack
        host_notification_commands              notify-host-by-slack
}

define command {
        command_name    notify-service-by-slack
        command_line    /usr/local/nagios/libexec/slack.pl -field slack_channel=#general
}

define command {
        command_name    notify-host-by-slack
        command_line    /usr/local/nagios/libexec/slack.pl -field slack_channel=#general
}
```

To enable alerting via the slack contact defined in the contacts.cfg config file, located under /usr/local/nagios/etc/objects/. Open it for editing:

- $ sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```
###############################################################################
#
# CONTACT GROUPS
#
###############################################################################

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {

     contactgroup_name       admins
     alias                   Nagios Administrators
     members                 nagiosadmin,slack
}

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go T
```

Open /usr/local/nagios/etc/nagios.cfg file and change the value of:

- $ enable_environment_macros=0 to 1.

Test the validity of the Nagios configuration:

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

Restart Nagios

Test the Slack integration:

- Send out a custom notification via the web interface

- Press **PostgreSQL Backends >> Send custom service notification**