Nagios 4 on Ubuntu 16

Install some base packages. I recommend following this guide as root on a new VPS or using sudo su, it will make running setup just a touch easier.

```
apt-get install php-gd build-essential apache2 wget php apache2-mod-php7.0 libgd-dev unzip sendmail
```

Now add some users for Nagios to use

```
useradd nagios
groupadd naggroup
usermod -a -G naggroup nagios
usermod -a -G nagios,naggroup www-data
```

Download Nagios Core (latest stable version at the time of this article is 4.3.1 however please check the Nagios website and replace the version number below with the latest)

```
cd ~ && wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.3.1.tar.gz
tar -xzf nagios*.tar.gz
cd nagios-*
```


Now we need to configure Nagios and compile it using the user and group we set out above

```
./configure --with-nagios-group=nagios --with-command-group=naggroup
make all
make install
make install-commandmode
make install-init
make install-config
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-
```

available/nagios.conf

The next step is grabbing plugins from Nagios Plugins. Check their website for the most up to date version. At the time of writing it was 2.2.0. Plugins are what enable us to run the different checks on users, processes, ssh, http, dns, MySQL etc.

```
cd ~ && wget https://nagios-plugins.org/download/nagios-plugins-2.2.0.tar.gz
tar -xzf nagios-plugins*.tar.gz
cd nagios-plugins-*
./configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl
make
make install
```

Let's go and configure Nagios itself now.

```
nano /usr/local/nagios/etc/nagios.cfg
```


You'll want to uncomment the line that adds the servers configuration directory as /usr/local/nagios/etc/servers

Now we create the directory and open the default contact.

```
mkdir -p /usr/local/nagios/etc/servers
nano /usr/local/nagios/etc/objects/contacts.cfg
```

Where it says "CHANGE THIS TO YOUR EMAIL ADDRESS" go ahead and change the email listed there. We will need to configure SMTP for outbound mail now as well.

Let's edit the commands configuration file so that we can use 'sendEmail' to actually fire off emails. See the SendEmail page for command line options. These are configured to use TLS on 587 with SMTP auth. We use Mailgun for our transactional email. We first need to install it from the repositories and then configure it.

```
sudo apt-get install sendemail
nano /usr/local/nagios/etc/objects/commands.cfg

# 'notify-host-by-email' command definition
define command{
        command_name    notify-host-by-email
  command_line  /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/sendEmail -s $USER7$ -t $CONTACTEMAIL$ -f $USER5$ -l /var/log/sendEmail -o tls=yes -xu $USER9$ -xp $USER10$ -u "** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" -m "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n"
}

# 'notify-service-by-email' command definition
define command{
        command_name    notify-service-by-email
        command_line  /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$" | /usr/bin/sendEmail -o tls=yes -s $USER7$ -t $CONTACTEMAIL$ -f $USER5$ -l /var/log/sendEmail -xu $USER9$ -xp $USER10$ -u "** $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" -m "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$"
}
```

And where we have added those variables we also need to add the corresponding host, user, pass etc to our resourcefile:

```
# SMTP config

# FROM Email
$USER5$=nagios@yourcompany.com

# SMTP server
$USER7$=smtp.company.org:587

# SMTP user
$USER9$=postmaster@mg.com.au

# SMTP password
$USER10$=password
```


Let's configure Apache with some modules and add a new nagios admin user to a htpasswd file to protect the install. Once you've typed the password in twice we make a symbolic link to enable the Nagios vhost inapache.

```
a2enmod rewrite

a2enmod cgi

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/
```

Now we add an init.d file to start nagios as a service:

```
/etc/init.d/skeleton /etc/init.d/nagios
nano /etc/init.d/nagios
```

Ensure the following code is in the file

```
DESC="Nagios"
NAME=nagios
DAEMON=/usr/local/nagios/bin/$NAME
DAEMON_ARGS="-d /usr/local/nagios/etc/nagios.cfg"
PIDFILE=/usr/local/nagios/var/$NAME.lock
```

Start the service:

```
chmod +x /etc/init.d/nagios
systemctl daemon-reload
systemctl enable nagios
service apache2 restart
service nagios start
```

Next step is we want to enable a free SSL certificate for our host. Make sure you have a DNS name pointing to your new VPS such as nagios.yourcompany.com and then go ahead and install the certbot to grab a free SSL cert from LetsEncrypt. After the last line you'll be prompted by certbot that it can't find a default servername so go and enter in your DNS name there. Make sure you choose Option 2 at the end to force SSL on all requests so that HTTP is redirected to HTTPS.

```
add-apt-repository ppa:certbot/certbot
apt-get update
apt-get install python-certbot-apache
certbot --apache
```

Visit https://nagios.yourcompany.com/nagios/ or whichever domain you used and check that Nagios is up and running.

Now we need to add a host to monitor. What we are going to do is add all of our servers we wish to monitor in /usr/local/nagios/etc/servers/

Call the host whatever you like, I've just use web01 below for simplicity pretending it's a web server. Make sure you modify the variables to suit your needs.

```
nano /usr/local/nagios/etc/servers/web01.cfg
```

```
define host {
   use                  linux-server
   host_name            web01
   alias                web01.mycompany.com.au
   address              123.12.34.12
   register             1
}

define service{
   host_name            web01
   use                  generic-service
   service_description  PING
   check_command        check_ping!150.0,10%!200.0,60%
   register             1
}

define service{
   host_name            web01
   use                  generic-service
   service_description  Current Users
   check_command        check_nrpe!check_users
   register             1
}

define service{
```

```
    host_name            web01
    use                  generic-service
    service_description  Root Partition
    check_command        check_nrpe!check_vda1
    register             1
}

define service{
    host_name            web01
    use                  generic-service
    service_description  Current Load
    check_command        check_nrpe!check_load
    register             1
}

define service{
    host_name            web01
    use                  generic-service
    service_description  SSH
    check_command        check_ssh
    register             1
}

define service{
    host_name            web01
    use                  generic-service
    service_description  HTTP
    check_command        check_http!web01.mycompany.com.au!-S --onredirect=follow
    register             1
}

define service{
```

```
  host_name              web01
  use                    generic-service
  service_description    Total Processes
  check_command          check_total_procs!150!200!RSZDT
  register               1
}
```

Now that's done, let's go and edit the service template. Make sure you bookmark https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/objectdefinitions.htmlwhich is the Nagios Core object definitions page. You'll be using Ctrl+F on this page a lot to understand what everything means.

In the sample code I've pasted above, you'll see 'use generic-service' a lot is mentioned. This is because it's using a Nagios 'template' which is stored in another file. A template means we re-use code and don't have to type it out all over again. Common things that we normally set once are:

– Contact groups, who is notified when something goes wrong

– Notification intervals

– Time periods

– How many times do you want to check a host before saying "This is actually down" ?

Your template config file is /usr/local/nagios/etc/objects/templates.cfg

```
nano /usr/local/nagios/etc/objects/templates.cfg
```

Now that you've zipped in and out of there let's add an NRPE check command and plugin to our install so that we can query remote servers.

```
apt-get install nagios-nrpe-plugin
cp /usr/lib/nagios/plugins/check_nrpe /usr/local/nagios/libexec/
```

At the bottom of your commands file add a check_nrpe command.

```
nano /usr/local/nagios/etc/objects/commands.cfg

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

Check the config to make sure we haven't made any mistakes and apply it with a reload:

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

And now let's go to our web01 host and install the nrpe server. Make sure you do this on every host you wish to manage. If you are not using IP tables, substitute a similar command like ufw in place. Also substitute the IP address with that of your Nagios server so that you're only accepting connections from one host for security reasons.


```
iptables -A INPUT -s 12.34.11.11 -p tcp --dport 5666 -j ACCEPT
iptables -A INPUT -p tcp --dport 5666 -j DROP
netfilter-persistent save

apt-get install nagios-nrpe-server nagios-plugins -y
nano /etc/nagios/nrpe.cfg
```

In the config file you've just opened up, edit the allowed_hosts directive and add in the IP address of your Nagios server. So it should read something like 127.0.0.1,12.34.11.11

Go down near the bottom of the file where the check commands are and substitute in the following, making sure you change to suit your environment. Remember the command definitions are done on the client side so let's get the users, processes, load averages etc. right because these aren't changed on the server (but can be if you allow arguments to be passed via NRPE, not recommended for security reasons).

Remember that the values specified here are 'warning' and 'critical' levels. i.e. with check users, anything 5 or over is a warning, anything 10 or over is critical. These are 'logged in users'. For load, the values for 1, 5 and 15 minute averages are specified. So in the example below if the load exceeds 0.8 for 5 minutes it's going to go to warning, and if it exceeds 1.0 for 15 minutes it's going to go critical. Disk checks are specified in amount of free space i.e. 20% free space is warning and 10% free space is critical.

```
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 1,0.8,0.5 -c 2,1.5,1
command[check_vda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/vda1
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

Restart NRPE server

```
service nagios-nrpe-server restart
```

Now you should start to see information populating in your web interface:

https://nagios.yourcompany.com/nagios/