How to install nagios for Ubuntu 16.04 LTS

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

User and group configuration

For Nagios to run, you have to create a new user for Nagios. We will name the user "nagios" and additionally create a group named "nagcmd".

We add the new user to the group as shown below:

```
sudo useradd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagios,nagcmd www-data
```

Installing Nagios

Step 1 - Download and extract the Nagios core

```
cd ~
#wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.0.tar.gz
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.3.2.tar.gz
tar -xzf nagios*.tar.gz
cd nagios-4.3.2
```

Step 2 - Compile Nagios

Before you build Nagios, you will have to configure it with the user and the group you have created earlier.

```
./configure --with-nagios-group=nagios --with-command-group=nagcmd
```

How to install Nagios

```
make all
sudo make install
sudo make install-commandmode
sudo make install-init
sudo make install-config
sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
```

Copy eventhandler directory to the nagios directory:

```
sudo cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/
sudo chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
```

Step 3 - Install the Nagios Plugins

Download and extract the Nagios plugins:

```
cd ~
wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
#wget https://nagios-plugins.org/download/nagios-plugins-2.1.2.tar.gz
tar -xzf nagios-plugins*.tar.gz
cd nagios-plugin-2.2.1/
```

Install the Nagios plugin's with the commands below:

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

Step 4 - Configure Nagios

After the installation phase is complete, you can find the default configuration of Nagios in /usr/local/nagios/.

We will configure Nagios and Nagios contact.

Edit default nagios configuration with vim:

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

uncomment line 51 for the host monitor configuration.
cfg_dir=/usr/local/nagios/etc/servers
Save and exit.
```

Add a new folder named servers:

```
sudo mkdir -p /usr/local/nagios/etc/servers
```

The Nagios contact can be configured in the contact.cfg file.

Replace the default email with your own email.

```
sudo vi /usr/local/nagios/etc/objects/contacts.cfg
```

Configuring Apache

Step 1 - enable Apache modules

```
sudo a2enmod rewrite
sudo a2enmod cgi
```

You can use the htpasswd command to configure a user nagiosadmin for the nagios web interface

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

and type your password.

Step 2 - enable the Nagios virtualhost

```
sudo ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/
```

Step 3 - Start Apache and Nagios

```
service apache2 restart
service nagios start
```

When Nagios starts, you may see the following error :

```
Starting nagios (via systemctl): nagios.serviceFailed
```

And this is how to fix it:

```
cd /etc/init.d/
cp /etc/init.d/skeleton /etc/init.d/nagios
```

Now edit the Nagios file:

```
sudo vi /etc/init.d/nagios
```

... and add the following code:

```
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
```

Make it executable and start Nagios:

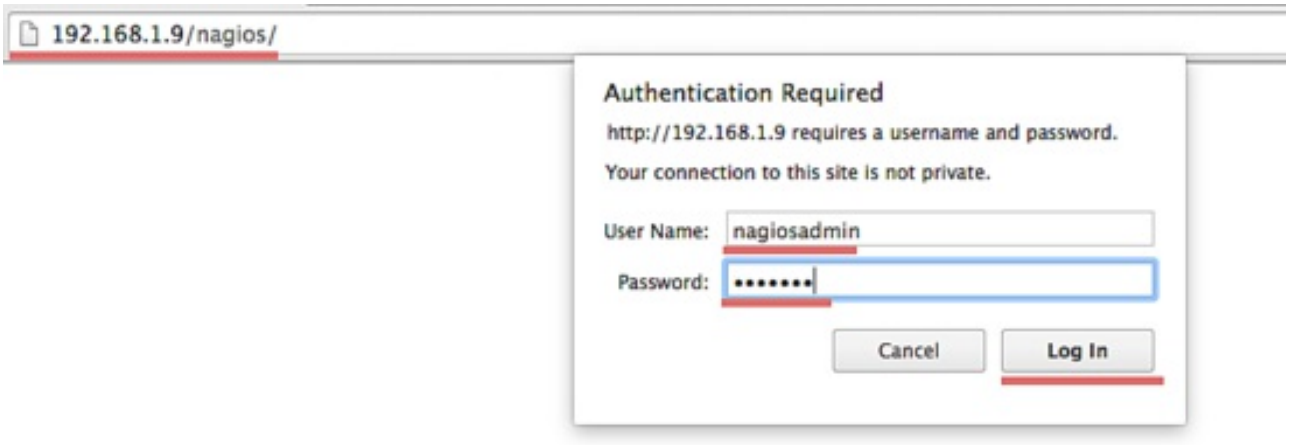
```
sudo chmod +x /etc/init.d/nagios
sudo chown root:root /etc/init.d/nagios
sudo service apache2 restart
service nagios start
sudo service nagios start

sudo update-rc.d nagios defaults
sudo update-rc.d nagios enable
```

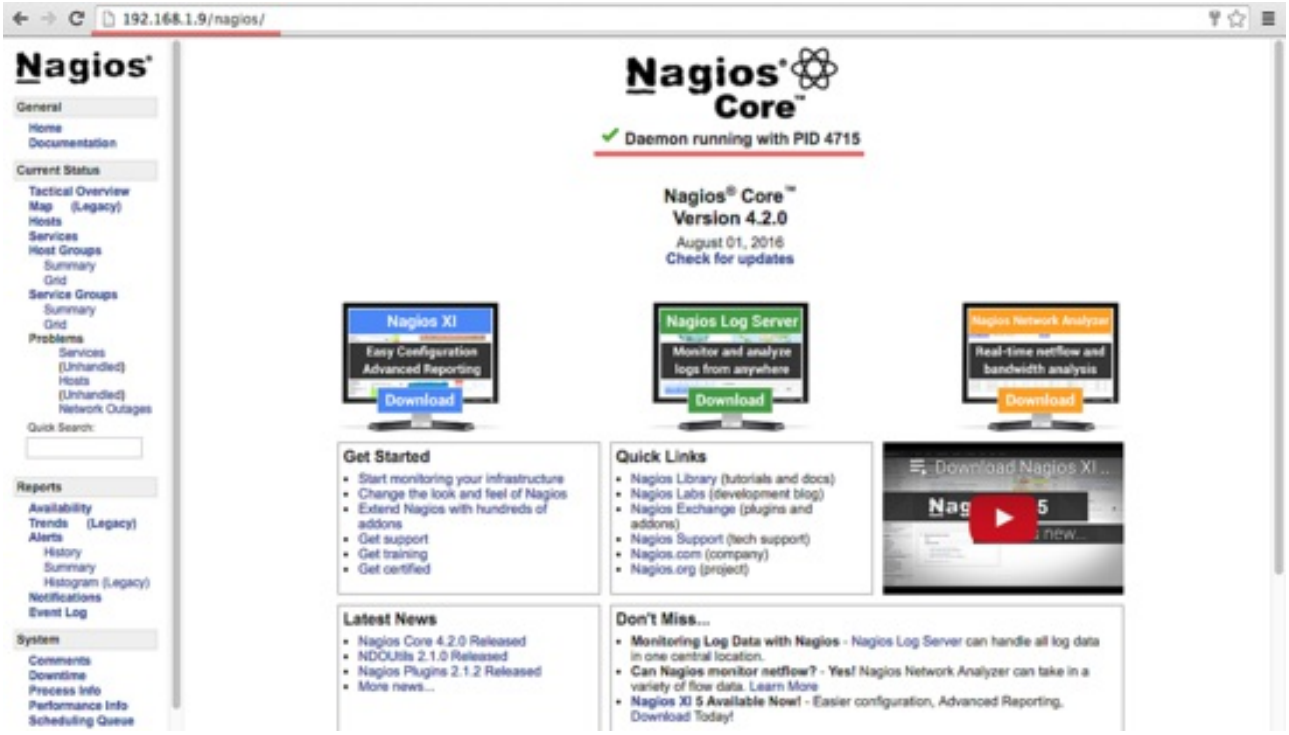
Testing the Nagios Server

Please open your browser and access the Nagios server ip, in my case: http://127.0.0.1/nagios

Nagios Login with apache htpasswd.



Nagios Admin Dashboard



Adding a Host to Monitor

In this tutorial, I will add an Ubuntu host to monitor to the Nagios server we have made above.

Nagios Server IP : 10.100.x.x
Ubuntu Host IP : 192.168.1.10

Step 1 - Connect to ubuntu host

```
ssh root@192.168.1.10
```

Step 2 - Install NRPE Service

```
sudo apt-get install nagios-nrpe-server nagios-plugins
```

Step 3 - Configure NRPE

After the installation is complete, edit the nrpe file /etc/nagios/nrpe.cfg:

```
sudo vi /etc/nagios/nrpe.cfg
```

... and add Nagios Server IP 192.168.1.9 to the server_address.

```
server_address=10.100.x.x
```

Step 4 - Restart NRPE

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

Step 5 - Add Ubuntu Host to Nagios Server

Please connect to the Nagios server:
ssh root@10.100.x.x

Then create a new file for the host configuration in /usr/local/nagios/etc/servers/.
sudo vi /usr/local/nagios/etc/servers/ubuntu_host.cfg

Add the following lines:

```
# Ubuntu Host configuration file

define host {
        use                             linux-server
        host_name                       ubuntu_host
        alias                           Ubuntu Host
        address                         192.168.1.10
        register                        1
}

define service {
      host_name                         ubuntu_host
      service_description               PING
      check_command                     check_ping!100.0,20%!500.0,60%
      max_check_attempts                2
      check_interval                    2
      retry_interval                    2
      check_period                      24x7
      check_freshness                   1
      contact_groups                    admins
      notification_interval             2
      notification_period               24x7
      notifications_enabled             1
      register                          1
}

define service {
      host_name                         ubuntu_host
      service_description               Check Users
      check_command           check_local_users!20!50
      max_check_attempts                2
```

```
      check_interval                2
      retry_interval                2
      check_period                  24x7
      check_freshness               1
      contact_groups                admins
      notification_interval         2
      notification_period           24x7
      notifications_enabled         1
      register                      1
}

define service {
      host_name                     ubuntu_host
      service_description           Local Disk
      check_command                 check_local_disk!20%!10%!/
      max_check_attempts            2
      check_interval                2
      retry_interval                2
      check_period                  24x7
      check_freshness               1
      contact_groups                admins
      notification_interval         2
      notification_period           24x7
      notifications_enabled         1
      register                      1
}

define service {
      host_name                     ubuntu_host
      service_description           Check SSH
      check_command                 check_ssh
      max_check_attempts            2
      check_interval                2
      retry_interval                2
      check_period                  24x7
      check_freshness               1
      contact_groups                admins
      notification_interval         2
      notification_period           24x7
      notifications_enabled         1
      register                      1
}

define service {
      host_name                     ubuntu_host
      service_description           Total Process
      check_command                 check_local_procs!250!400!RSZDT
      max_check_attempts            2
      check_interval                2
      retry_interval                2
      check_period                  24x7
      check_freshness               1
```

```
        contact_groups                  admins
        notification_interval           2
        notification_period             24x7
        notifications_enabled           1
        register                        1
}
```

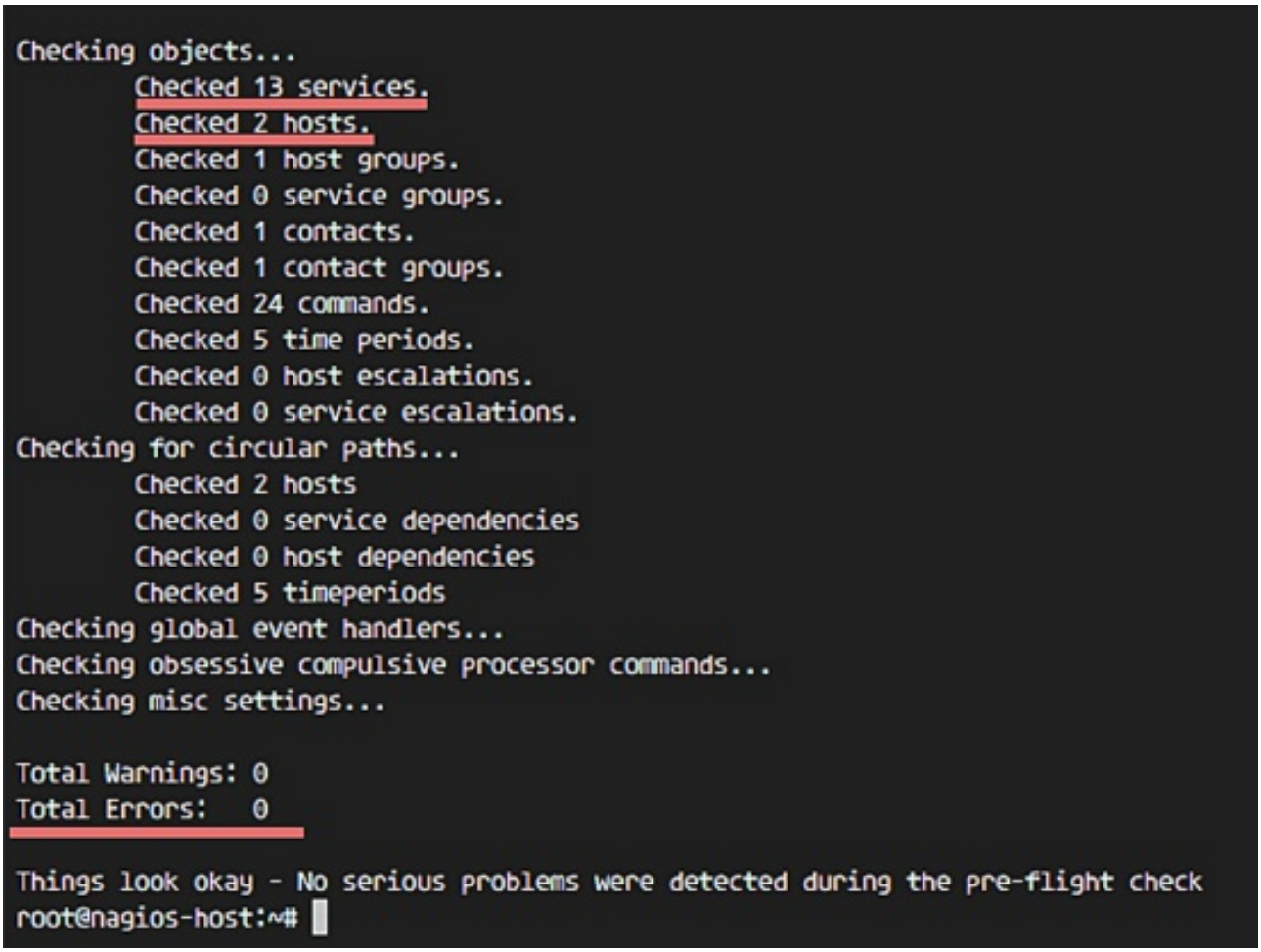
You can find many check_command in /usr/local/nagios/etc/objects/commands.cfg file.

See there if you want to add more services like DHCP, POP etc.

And now check the configuration:

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

... to see if the configuration is correct.



Step 6 - Restart all services

On the Ubuntu Host start NRPE Service:

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

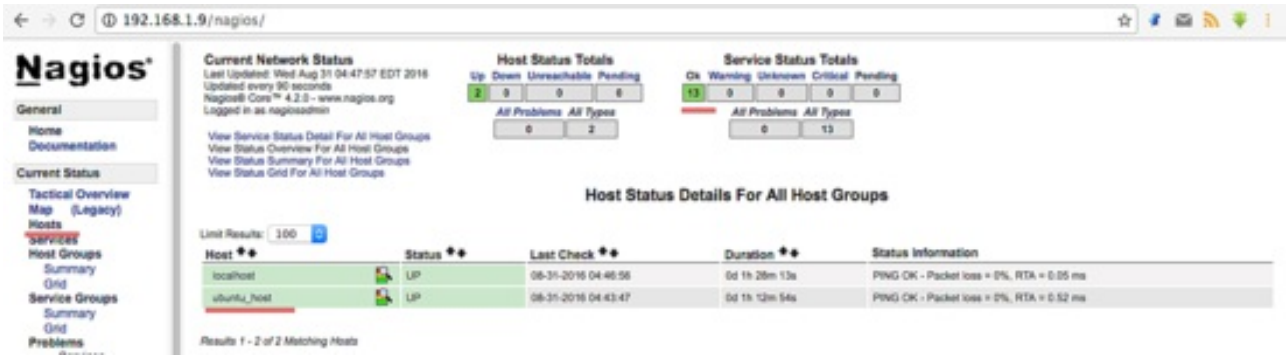
... and on the Nagios server, start Apache and Nagios:

```
sudo service apache2 restart
sudo service nagios restart
```

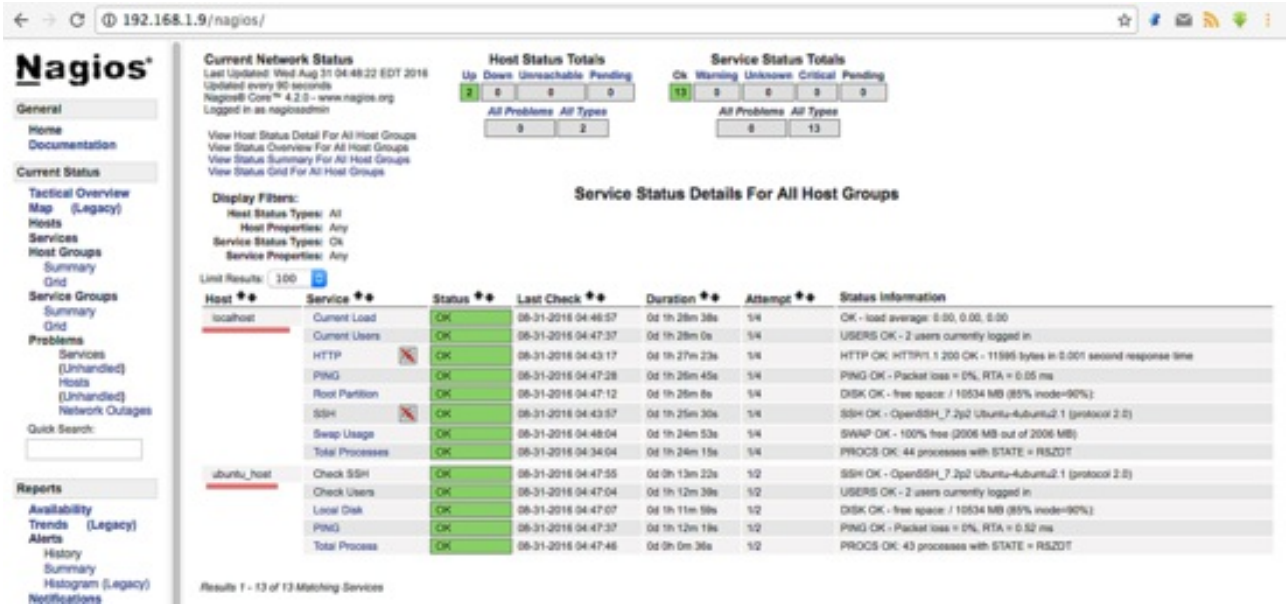
Step 7 - Testing the Ubuntu Host

Open the Nagios server from the browser and see the ubuntu_host being monitored.

The Ubuntu host is available on monitored host.



All services monitored without error.



Conclusion
Nagios is an open source application for monitoring a system. Nagios has been widely used because of the ease of configuration. Nagios in support by various plugins, and you can even create your own plugins.

Look here for more information:
https://nagios-plugins.org/doc/guidelines.html

Setup a custom notification

e.g., Whatsapp

https://www.unixmen.com/send-nagios-alert-notification-using-whatsapp/

cd /usr/local/nagios/libexec/
cd /usr/local/nagios/etc/objects/
cp commands.cfg commands.cfg.old

```
# Add commands to
vi /usr/local/nagios/etc/objects/commands.cfg
```

define command{
command_name notify-service-by-mutt
command_line echo "***** Nagios *****

Notification Type: $NOTIFICATIONTYPE$

Service: $SERVICEDESC$
Host: $HOSTALIAS$
Address: $HOSTADDRESS$
State: $SERVICESTATE$

Date/Time: $LONGDATETIME$

Additional Info:

$SERVICEOUTPUT$
" | mutt -e "set content_type=text/html" user@gmail.com -F /home/cisadmin/.muttrc-csinfo -s "Nagios alert: $HOSTALIAS{{content}}quot;
}

```
sudo vi /usr/local/nagios/etc/objects/contacts.cfg

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                           XXXXXX@gmail.com    ; <<***** CHANGE
THIS TO YOUR EMAIL ADDRESS ******
        service_notification_commands   notify-service-by-mutt ; notify-
service-by-email
        host_notification_commands      notify-service-by-mutt
        }
```

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
If there are no errors, restart the nagios service:

service nagios restart
```

Note: /usr/local/nagios/etc/objects/localhost.cfg has a good guide for host config files in /usr/local/nagios/etc/objects/servers

e.g.,

```
vi /usr/local/nagios/etc/servers/131.230.133.20.cfg
```

```
# Ubuntu Host configuration file

define host {
        use                             linux-server
        host_name                       pc00
        alias                           pc00
        address                         131.230.133.20
        register                        1
}

# Define a service to "ping" the local machine
define service {
      host_name                         pc00
      service_description               PING
      check_command                     check_ping!100.0,20%!500.0,60%
      max_check_attempts                5
      check_interval                    2
      retry_interval                    2
      check_period                      24x7
      check_freshness                   1
      contact_groups                    admins
      notification_interval             0
      notification_period               24x7
      notifications_enabled             1
      register                          1
}


# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

#define service {
      #host_name                         pc00
      #service_description               Check Users
      #check_command           check_local_users!20!50
      #max_check_attempts                2
      #check_interval                    2
      #retry_interval                    2
      #check_period                      24x7
      #check_freshness                   1
      #contact_groups                    admins
      #notification_interval             2
      #notification_period               24x7
      #notifications_enabled             1
      #register                          1
#}

# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if
# < 10% free space on partition.
```

```
# define service {
#       host_name                       pc00
#       service_description             Local Disk
#       check_command                   check_local_disk!20%!10%!/
#       max_check_attempts              2
#       check_interval                  2
#       retry_interval                  2
#       check_period                    24x7
#       check_freshness                 1
#       contact_groups                  admins
#       notification_interval           2
#       notification_period             24x7
#       notifications_enabled           1
#       register                        1
# }


# Define a service to check SSH on the local machine.
# Disable notifications for this service by default, as not all users may have
SSH enabled.
# notifications_enabled           0

define service {
      host_name                       pc00
      service_description             Check SSH
      check_command                   check_ssh
      max_check_attempts              5
      check_interval                  2
      retry_interval                  2
      check_period                    24x7
      check_freshness                 1
      contact_groups                  admins
      notification_interval           0
      notification_period             24x7
      notifications_enabled           1
      register                        1
}

# Define a service to check the number of currently running procs
# on the local machine.  Warning if > 250 processes, critical if
# > 400 processes.

# define service {
#       host_name                       pc00
#       service_description             Total Process
#       check_command                   check_local_procs!450!800!RSZDT
#       max_check_attempts              2
#       check_interval                  2
#       retry_interval                  2
#       check_period                    24x7
```

```
#       check_freshness                 1
#       contact_groups                  admins
#       notification_interval           2
#       notification_period             24x7
#       notifications_enabled           1
#       register                        1
# }
```

Setup mutt email account

```
sudo -i
mkdir -p /home/nagios
su nagios
cd ~
mkdir ~/Mail
cp <.muttrc> ~/.mutt-csinfo.rc
```

Setup a custom daily nagios report email

```
vi /etc/cron.daily/nagios-reporter.pl
```

Paste the following into nagios-reporter.pl

```
#!/usr/bin/perl -w
#
# Nagios overnight/daily/weekly/monthly reporter
#
# Fetches Nagios report from web, processes HTML/CSS and emails to someone
# Written by Rob Moss, 2005-07-26, coding@mossko.com
#
# Use at your own risk, knoweledge of perl required.
#
# Version 1.3.1
# - Overnight, Daily, Weekly, Monthly reports
#

use strict;
use Getopt::Long;
use Net::SMTP;
use LWP::UserAgent;
#use Date::Manip;
use v5.16;


my $mailhost    =   'mail.domain.edu';              #   Fill these in!
my $maildomain  =   'domain.edu';                       #   Fill these in!
my $mailfrom    =   'user@domain.edu';                  #   Fill these in!
my $mailto      =   'user@gmail.com';                   #   Fill these in!
my $timeout     =   30;
my $mailsubject =   '';
my $mailbody    =   '';
```

```
my $logfile     =   '/usr/local/nagios/var/mail.log';   #   Where would you
like your logfile to live?
my $debug       =   1;                                  #   Set the debug level
to 1 or higher for information

my $type        =   '';
my $repdateprev;
my $reporturl;

my $nagssbody;
my $nagsssummary;

my $webuser     =   'nagiosadmin';                          #   Set this to a
read-only nagios user (not nagiosadmin!)
my $webpass     =   '<PASSWORD_HERE>';                          #   Set this to
a read-only nagios user (not nagiosadmin!)
my $webbase     =   'http://127.0.0.1/nagios';                  #   Set this to
the base of Nagios web page
my $webcssembed =   0;


GetOptions (
    "debug=s"   =>  \$debug,
    "help"      =>  \&help,
    "type=s"    =>  \$type,
    "email=s"   =>  \$mailto,
    "embedcss"  =>  \$webcssembed,
);


if (not defined $type or $type eq "") {
    help();
    exit;
}
elsif ($type eq "overnight") {
    report_overnight();
}
elsif ($type eq "daily") {
    report_daily();
}
elsif ($type eq "weekly") {
    report_weekly();
}
elsif ($type eq "monthly") {
    report_monthly();
}
else {
    die("Unknown report type $type\n");
}
```

```
debug(1,"reporturl: [$reporturl]");

$mailbody = http_request($reporturl);
if ($webcssembed) {
    # Stupid hacks for dodgy notes
    $nagssbody      =   http_request("$webbase/stylesheets/summary.css");
    $nagsssummary = "<style type=\"text\/css\">\n";
    foreach ( split(/\n/,$nagssbody) ) {
        chomp;
        if (not defined $_ or $_ eq "" ) {
            next;
        }
        $nagsssummary .= "<!-- $_ -->\n";
    }
    $nagsssummary .= "</style>\n";
    $nagsssummary .= "<base href=\"$webbase/cgi-bin/\">\n";

    $mailbody =~ s@<LINK REL=\'stylesheet\' TYPE=\'text/css\'
HREF=\'/stylesheets/common.css\'>@@;
    $mailbody =~ s@<LINK REL=\'stylesheet\' TYPE=\'text/css\'
HREF=\'/stylesheets/summary.css\'>@$nagsssummary@;
}



open(FILE, "> /tmp/nagios-report-htmlout.html") or warn "can't open file
/tmp/nagios-report-htmlout.html: $!\n";
print FILE $mailbody;
close FILE;



sendmail();


##############################################################################
sub help {
print <<_END_;

Nagios web->email reporter program.

$0 <args>

--help
    This screen

--email=<email>
    Send to this address instead of the default address
    "$mailto"

--type=overnight
    Overnight report, from 17h last working day to Today (9am)
```

```
--type=daily
    Daily report, 09:00 last working day to Today (9am)
--type=weekly
    Weekly report, 9am 7 days ago, until 9am today (run at 9am friday!)
--type=monthly
    Monthly report, 1st of prev month at 9am to last day of month, 9am

--embedcss
    Downloads the CSS file and embeds it into the main HTML to enable
    Lotus Notes to work (yet another reason to hate Notes)

_END_

exit 1;

}

##############################################################################
sub report_monthly {
    # This should be run on the 1st of every month
    $repdateprev = DateCalc("yesterday",1);
    debug(1,"repdateprev = $repdateprev");
#               #2006072116:48:37
    my ($repsday, $repsmonth, $repsyear, $repshour ) = 0;
    $repdateprev =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repsday = 01;
    $repsmonth = $2;
    $repsyear = $1;
    $repshour = 0;

    my ($repeday, $repemonth, $repeyear, $repehour ) = 0;
    my $repdatenow = ParseDate("today");
    debug(1,"repdatenow = $repdatenow");
    $repdatenow =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repeday = $3;
    $repemonth = $2;
    $repeyear = $1;
    $repehour = 0;

    $reporturl  =   "$webbase/cgi-bin/summary.cgi?
report=1&displaytype=1&timeperiod=custom" .

"&smon=$repsmonth&sday=$repsday&syear=$repsyear&shour=$repshour&smin=0&ssec=0"
.

"&emon=$repemonth&eday=$repeday&eyear=$repeyear&ehour=$repehour&emin=0&esec=0"
.

'&hostgroup=all&servicegroup=all&host=all&alerttypes=3&statetypes=2&hoststates=
3&servicestates=56&limit=500';
    $mailsubject = "Nagios alerts for month $repsmonth/$repsyear";
```

```
}

################################################################################
sub report_weekly {
    # This should be run on Friday, 9am
    $repdateprev = Date_PrevWorkDay("today",5);
    debug(1,"repdateprev = $repdateprev");
                #2006072116:48:37
    my ($repsday, $repsmonth, $repsyear, $repshour ) = 0;
    $repdateprev =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repsday = $3;
    $repsmonth = $2;
    $repsyear = $1;
    $repshour = 9;

    my ($repeday, $repemonth, $repeyear, $repehour ) = 0;
    my $repdatenow = ParseDate("today");
    debug(1,"repdatenow = $repdatenow");
    $repdatenow =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repeday = $3;
    $repemonth = $2;
    $repeyear = $1;
    $repehour = 9;

    $reporturl  =   "$webbase/cgi-bin/summary.cgi?
report=1&displaytype=1&timeperiod=custom" .

"&smon=$repsmonth&sday=$repsday&syear=$repsyear&shour=$repshour&smin=0&ssec=0"
.

"&emon=$repemonth&eday=$repeday&eyear=$repeyear&ehour=$repehour&emin=0&esec=0"
.

'&hostgroup=all&servicegroup=all&host=all&alerttypes=3&statetypes=2&hoststates=
3&servicestates=56&limit=500';
    $mailsubject = "Nagios alerts for week ending
$repsday/$repsmonth/$repsyear";

}


################################################################################
sub report_daily {
    #$repdateprev = Date_PrevWorkDay("today",1);
    #debug(1,"repdateprev = $repdateprev");
                #2006072116:48:37
    my ($repsday, $repsmonth, $repsyear, $repshour ) = 0;
    #$repdateprev =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repsday = $3;
    $repsmonth = $2;
    $repsyear = $1;
    $repshour = 7;
```

```
    my ($repeday, $repemonth, $repeyear, $repehour ) = 0;
    #my $repdatenow = ParseDate("today");
    #debug(1,"repdatenow = $repdatenow");
    #$repdatenow =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repeday = $3;
    $repemonth = $2;
    $repeyear = $1;
    $repehour = 7;

    #$reporturl =   "$webbase/cgi-bin/summary.cgi?
report=1&displaytype=1&timeperiod=custom" .
    #
"&smon=$repsmonth&sday=$repsday&syear=$repsyear&shour=$repshour&smin=0&ssec=0"
.
    #
"&emon=$repemonth&eday=$repeday&eyear=$repeyear&ehour=$repehour&emin=0&esec=0"
.
    #
'&hostgroup=all&servicegroup=all&host=all&alerttypes=3&statetypes=2&hoststates=
3&servicestates=56&limit=500';

    #$reporturl = "http://127.0.0.1/nagios3/cgi-bin/summary.cgi?
report=1&displaytype=1&timeperiod=last24hours&hostgroup=all&servicegroup=all&ho
st=all&alerttypes=3&statetypes=3&hoststates=7&servicestates=120&limit=100";
  $reporturl = "http://127.0.0.1/nagios/cgi-bin/summary.cgi?
report=1&displaytype=1&timeperiod=last7days&hostgroup=all&servicegroup=all&host
=all&alerttypes=3&statetypes=3&hoststates=7&servicestates=120&limit=100";

    #$mailsubject = "Nagios alerts for 24 hours $repsday/$repsmonth/$repsyear
${repshour}h to present";
    $mailsubject = "Nagios alerts for last week";

}

################################################################################
sub report_overnight {
    $repdateprev = Date_PrevWorkDay("today",1);
    debug(1,"repdateprev = $repdateprev");
                #2006072116:48:37
    my ($repsday, $repsmonth, $repsyear, $repshour ) = 0;
    $repdateprev =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repsday = $3;
    $repsmonth = $2;
    $repsyear = $1;
    $repshour = 17;

    my ($repeday, $repemonth, $repeyear, $repehour ) = 0;
    my $repdatenow = ParseDate("today");
    debug(1,"repdatenow = $repdatenow");
    $repdatenow =~ /(\d\d\d\d)(\d\d)(\d\d)(.*)/;
    $repeday = $3;
```

```
    $repemonth = $2;
    $repeyear = $1;
    $repehour = 9;

    $reporturl  =   "$webbase/cgi-bin/summary.cgi?
report=1&displaytype=1&timeperiod=custom" .

"&smon=$repsmonth&sday=$repsday&syear=$repsyear&shour=$repshour&smin=0&ssec=0"
.

"&emon=$repemonth&eday=$repeday&eyear=$repeyear&ehour=$repehour&emin=0&esec=0"
.

'&hostgroup=all&servicegroup=all&host=all&alerttypes=3&statetypes=2&hoststates=
3&servicestates=56&limit=500';
    $mailsubject = "Nagios overnight alerts from $repsday/$repsmonth/$repsyear
${repshour}h to present";

}

################################################################################
sub http_request {
    my $ua;
    my $req;
    my $res;

    my $geturl = shift;
    if (not defined $geturl or $geturl eq "") {
        warn "No URL defined for http_request\n";
        return 0;
    }
    $ua = LWP::UserAgent->new;
    $ua->agent("Nagios Report Generator " . $ua->agent);
    $req = HTTP::Request->new(GET => $geturl);
    $req->authorization_basic($webuser, $webpass);
    $req->header(   'Accept'            =>  'text/html',
                    'Content_Base'      =>  $webbase,
                );

    # send request
    $res = $ua->request($req);

    # check the outcome
    if ($res->is_success) {
        debug(1,"Retreived URL successfully");
        print "Output: " . $res->decoded_content . "\n";
        return $res->decoded_content;
    }
    else {
        print "Error: " . $res->status_line . "\n";
        return 0;
    }
```

```
}

##############################################################################
sub debug {
    my ($lvl,$msg) = @_;
    if ( defined $debug and $lvl <= $debug ) {
        chomp($msg);
        print localtime(time) .": $msg\n";
    }
    return 1;
}

############################################################
sub sendmail {


    #system("echo test");
    #say `echo test`;
    say `echo "$mailbody" | mutt -e "set content_type=text/html" $mailto  -F
~/.muttrc-csinfo -s "Nagios alerts for 24 hours"`;
    #echo $mailbody




    #system("echo $mailbody");
    #system("echo test");
    #echo "***** Nagios *****<BR><BR>Notification Type:
$NOTIFICATIONTYPE{{content}}lt;BR>Host: $HOSTNAME{{content}}lt;BR>State:
$HOSTSTATE{{content}}lt;BR>Address: $HOSTADDRESS{{content}}lt;BR>Info:
$HOSTOUTPUT{{content}}lt;BR><BR>Date/Time: $LONGDATETIME{{content}}lt;BR>" |
mutt -e "set content_type=text/html" $CONTACTEMAIL$ -F ~/.muttrc-csinfo -s "**
$NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **
}
```

```
sudo chmod +x /etc/cron.daily/nagios-reporter.pl
crontab -e
```

```
# m h  dom mon dow   command
0 6 * * * /etc/cron.daily/nagios-reporter.pl --type=daily
```

Confirm the report works

```
/etc/cron.daily/nagios-reporter.pl --type=daily
```