Document Category: Setup

Version: 0.2

Status: Draft

Last Updated: Oct 07, 2015

Nagios Cookbook

Contents

[Contents 2](#_Toc431982754)

[1 Introduction 4](#_Toc431982755)

[1.1 Scope 4](#_Toc431982756)

[1.2 Related documents 4](#_Toc431982757)

[1.3 Terminology 4](#_Toc431982758)

[2 Pre-requisites 5](#_Toc431982759)

[2.1 Base OS 5](#_Toc431982760)

[2.2 Install Apache Web server 5](#_Toc431982761)

[2.3 Install PHP 6](#_Toc431982762)

[2.4 Install development tools 7](#_Toc431982763)

[2.5 Download Nagios Source code packages 7](#_Toc431982764)

[3 Installing Nagios 8](#_Toc431982765)

[3.1 Create Nagios users and groups 8](#_Toc431982766)

[3.2 Extract Nagios core and plugins 8](#_Toc431982767)

[3.3 Configure Nagios Core 8](#_Toc431982768)

[3.4 Install and Configure the Nagios Web Interface 11](#_Toc431982769)

[3.5 Compile and Install the Nagios plugins 11](#_Toc431982770)

[3.6 Post-configuration 11](#_Toc431982771)

[4 Installing nagiosgraph 14](#_Toc431982772)

[4.1 Pre-requisites 14](#_Toc431982773)

[4.2 Install nagiosGraph 15](#_Toc431982774)

[4.3 Configure Nagios for NagiosGraph 17](#_Toc431982775)

[4.4 Verification 18](#_Toc431982776)

[5 Solace Plugin for Nagios 20](#_Toc431982777)

[5.1 Hosts and Services 20](#_Toc431982778)

[5.2 Host groups and Service groups 20](#_Toc431982779)

[5.3 Host and Service checks 21](#_Toc431982780)

[5.4 Active and Passive checks 21](#_Toc431982781)

[5.5 Commands available for the Solace plugin 22](#_Toc431982782)

[5.5.1 Active Checks 22](#_Toc431982783)

[5.5.2 Passive Checks 24](#_Toc431982784)

[6 Installing the Nagios Plugin for Solace 25](#_Toc431982785)

[6.1 Install required Perl modules 25](#_Toc431982786)

[6.2 Stop Nagios 26](#_Toc431982787)

[6.3 Copy executables 26](#_Toc431982788)

[6.4 Copy config files 26](#_Toc431982789)

[6.5 Set up Credentials Directory 28](#_Toc431982790)

[6.6 Create object configurations 28](#_Toc431982791)

[6.6.2 Thresholds for monitoring services 32](#_Toc431982792)

[6.7 Verification 33](#_Toc431982793)

[6.7.1 Object definitions 33](#_Toc431982794)

[6.7.2 File permissions 33](#_Toc431982795)

[6.7.3 Nagios pre-flight check 34](#_Toc431982796)

[6.7.4 Network level verification 34](#_Toc431982797)

[6.8 Start the Nagios service 34](#_Toc431982798)

[7 Additional Documentation 36](#_Toc431982799)

[8 Outstanding Issues 37](#_Toc431982800)

[Approvals 38](#_Toc431982801)

[History 39](#_Toc431982802)

# Introduction

## Scope

This document outlines the procedures for building and installing the Nagios Core and basic Nagios plugins from source, as well as NagiosGraph. After going through the procedures in this document, the user will be able to set up a vanilla Nagios dashboard for monitoring purposes.

The procedure has been tested on a minimal version of CentOS 6.7 and the versions of Nagios installed were:

* nagios-4.1.1
* nagios-plugins-2.1.1
* nagiosgraph-1.5.2

## Related documents

These documents contain information related to the information in this document.

| Document Number | Document Title | Version (Optional) |
| --- | --- | --- |
|  |  |  |

Table 1‑1 Related Documents

## Terminology

| Term / Acronym / Abbreviation | Definition |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Table 1‑2 Terminology

# Pre-requisites

## Base OS

This document assumes that the base operating system is CentOS - however the procedure should work for any UNIX-like system with similar commands.

Once CentOS has booted up, apply the latest updates:

|  |
| --- |
| [root@centosServer ~]# yum -y update |

## Install Apache Web server

* To install Apache, enter the following command:

|  |
| --- |
| [root@centosServer ~]# yum install httpd -y |

* Modify "/etc/httpd/conf/httpd.conf" to add an entry for the serverName:

|  |
| --- |
| # ServerName gives the name and port that the server uses to identify itself.  # This can often be determined automatically, but we recommend you specify  # it explicitly to prevent problems during startup.  #  # If this is not set to valid DNS name for your host, server-generated  # redirections will not work. See also the UseCanonicalName directive.  #  # If your host doesn't have a registered DNS name, enter its IP address here.  # You will have to access it by its address anyway, and this will make  # redirections work in a sensible way.  ServerName centosServer:80 |

* Start the Apache service and configure it to start automatically when the server boots:

|  |
| --- |
| [root@centosServer ~]# service httpd start  [root@centosServer ~]# chkconfig httpd on |

* Modify the firewall to allow incoming connections to the Apache server port 80 from remote systems. Edit "/etc/sysconfig/iptables" and add the following lines at the end:

|  |
| --- |
| -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEP |

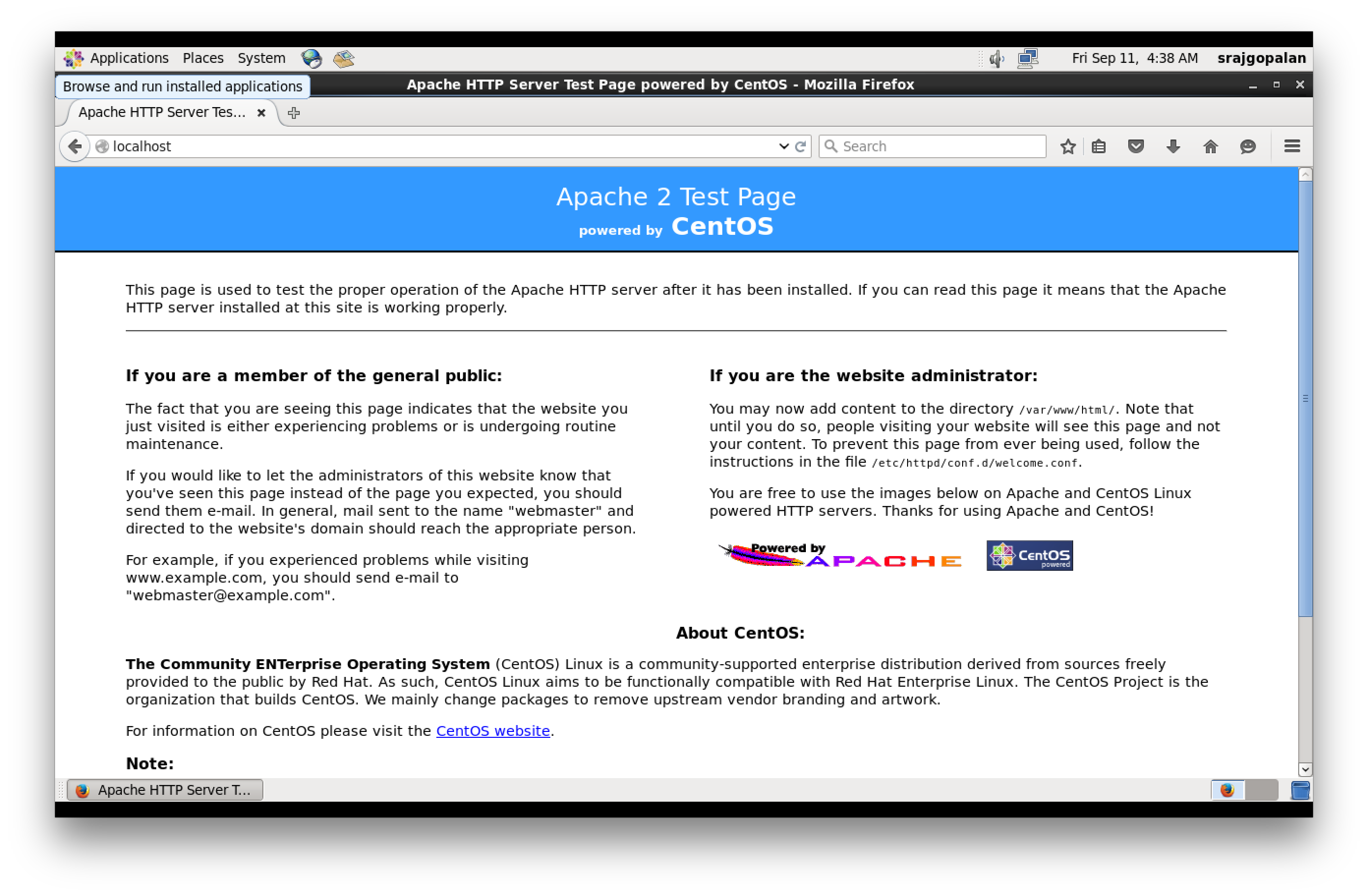
* Restart iptables for the rules to take effect:

|  |
| --- |
| [root@centosServer ~]# service iptables restart |

* Configure Apache to start every time the system boots

|  |
| --- |
| [root@centosServer ~]# chkconfig --add httpd  [root@centosServer ~]# chkconfig --level 35 httpd on |

* To verify the installation of Apache, navigate to http://localhost/ or http://<server-ip-address>



## Install PHP

* Install PHP with the following command:

|  |
| --- |
| [root@centosServer ~]# yum install php -y |

* To verify the installation of PHP, run the below command:

|  |
| --- |
| [root@centosServer ~]# php -i  phpinfo()  PHP Version => 5.3.3  <output snipped> |

## Install development tools

Run each of the below commands to install the respective development tools:

|  |
| --- |
| [root@centosServer ~]# yum install gcc gcc-c++ make openssl-devel  [root@centosServer ~]# yum install -y libxml2 libxml2-devel libxslt libxslt-devel  [root@centosServer ~]# yum install gd gd-devel gcc glibc glibc-common |

## Download Nagios Source code packages

Download the latest version of the source code tarballs for NagiosCore and Nagios Plugins to a temporary directory such as "~/Downloads". Grab the tarballs from https://www.nagios.org/projects/nagios-core/.

# Installing Nagios

## Create Nagios users and groups

* Create a new nagios user and set a password. Also create a nagcmd group in order to allow external commands to be submitted via the web interface

|  |
| --- |
| [root@centosServer ~]# useradd -m -s /bin/bash nagios  [root@centosServer ~]# passwd  [root@centosServer ~]# /usr/sbin/groupadd nagcmd |

* On older UNIX systems, a nagios group is not automatically created and this needs to be done manually:

|  |
| --- |
| [root@centosServer ~]# /usr/sbin/groupadd nagios  [root@centosServer ~]# /usr/sbin/usermod -G nagios nagios |

* Add both the nagios and apache users to the nagcmd group

|  |
| --- |
| [root@centosServer ~]# usermod -a -G nagcmd nagios  [root@centosServer ~]# usermod -a -G nagcmd apache |

## Extract Nagios core and plugins

* Extract the Nagios core and plugins packages downloaded in Section 2.5

|  |
| --- |
| [root@centosServer ~]# tar -xvf <package-name>.tar.gz |

Once extracted, two directories should appear:

|  |
| --- |
| [root@centosServer Downloads]# ll  total 14684  drwxrwxr-x. 18 root root 4096 Sep 10 01:11 nagios-4.1.1  -rw-r--r--. 1 root root 11206656 Sep 10 01:07 nagios-4.1.1.tar.gz  drwxr-xr-x. 15 root root 4096 Sep 10 01:22 nagios-plugins-2.1.1  -rw-r--r--. 1 root root 2677352 Sep 10 01:07 nagios-plugins-2.1.1.tar.gz |

## Configure Nagios Core

This procedure will install Nagios Core and plugins under "/usr/local/nagios"

* Enter the directory for the nagios core source and run the configure file

|  |
| --- |
| [root@centosServer Downloads]# **cd nagios-4.1.1**  [root@centosServer nagios-4.1.1]# **./configure --with-command-group=nagcmd**  <output-snipped>  \*\*\* Configuration summary for nagios 4.1.1 08-19-2015 \*\*\*:  General Options:  -------------------------  Nagios executable: nagios  Nagios user/group: nagios,nagios  Command user/group: nagios,nagcmd  Event Broker: yes  Install ${prefix}: /usr/local/nagios  Install ${includedir}: /usr/local/nagios/include/nagios  Lock file: ${prefix}/var/nagios.lock  Check result directory: ${prefix}/var/spool/checkresults  Init directory: /etc/rc.d/init.d  Apache conf.d directory: /etc/httpd/conf.d  Mail program: /bin/mail  Host OS: linux-gnu  IOBroker Method: epoll  Web Interface Options:  ------------------------  HTML URL: http://localhost/nagios/  CGI URL: http://localhost/nagios/cgi-bin/  Traceroute (used by WAP): /bin/traceroute  Review the options above for accuracy. If they look okay,  type 'make all' to compile the main program and CGIs. |

* Compile the Nagios Source code

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **make all**  <output snipped>  \*\*\* Compile finished \*\*\*  If the main program and CGIs compiled without any errors, you  can continue with installing Nagios as follows (type 'make'  without any arguments for a list of all possible options):  make install  - This installs the main program, CGIs, and HTML files  ....  ....  For more information on obtaining support for Nagios, visit:  https://support.nagios.com  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enjoy. |

* Install the binaries that were compiled

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **make install** |

* Install the init script

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **make install-init**  /usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d  /usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios  \*\*\* Init script installed \*\*\* |

* Install command-mode to make Nagios work with the command line

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **make install-commandmode**  /usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw  chmod g+s /usr/local/nagios/var/rw  \*\*\* External command directory configured \*\*\* |

* Install sample nagios config files

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **make install-config**  /usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc  /usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects  /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg  <output snipped>  \*\*\* Config files installed \*\*\*  Remember, these are \*SAMPLE\* config files. You'll need to read  the documentation for more information on how to actually define  services, hosts, etc. to fit your particular needs. |

* In order to get email notifications for alerts generated by Nagios, edit "/usr/local/nagios/etc/objects/contacts.cfg" to the contact that will receive the alerts.

|  |
| --- |
| 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 nagios@localhost ; <<\*\*\*\*\* CHANGE THIS TO YOUR EMAIL ADDRESS \*\*\*\*\*\*  } |

In order for Nagios to send out email alerts, mail needs to be configured on the server - this is out of the scope of this document.

## Install and Configure the Nagios Web Interface

* Navigate to the directory containing the compiled Nagios source, and install the web interface for Nagios:

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **make install-webconf**  /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf  \*\*\* Nagios/Apache conf file installed \*\*\* |

* Create a user "nagiosadmin" for the web interface and assign a password to it:

|  |
| --- |
| [root@centosServer nagios-4.1.1]# **htpasswd -c /usr/local/nagios/etc/htpasswd.users** nagiosadmin  New password:  Re-type new password:  Adding password for user nagiosadmin |

* Restart Apache for the new settings to take effect:

|  |
| --- |
| [root@centosServer nagios-4.1.1]# service httpd restart  Stopping httpd: [ OK ]  Starting httpd: [ OK ]  [root@centosServer nagios-4.1.1]# |

## Compile and Install the Nagios plugins

Compile and install the Nagios plugins downloaded in Section 2.5.

|  |
| --- |
| [root@centosServer Downloads]# cd nagios-plugins-2.1.1  [root@centosServer nagios-plugins-2.1.1]# ./configure --with-nagios-user=nagios --with-nagios-group=nagios  [root@centosServer nagios-plugins-2.1.1]# make  [root@centosServer nagios-plugins-2.1.1]# make install |

## Post-configuration

* Verify that Nagios installed correctly by running a pre-flight check. If the procedure was followed, no errors should be seen.

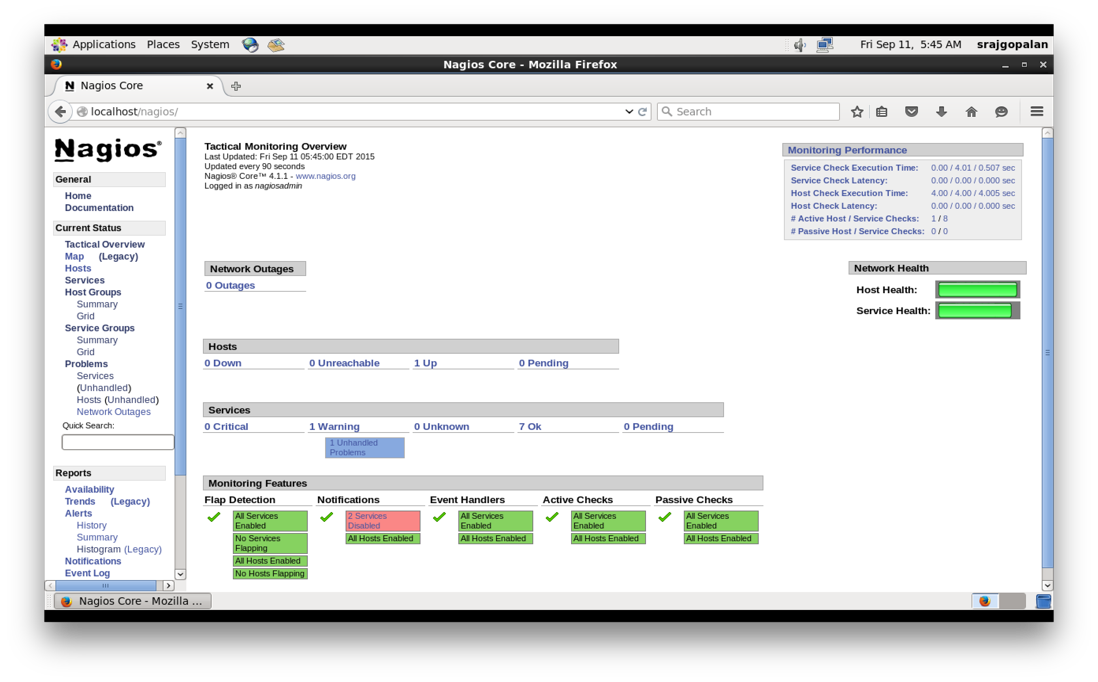
|  |
| --- |
| [root@centosServer ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg  Nagios Core 4.1.1  Copyright (c) 2009-present Nagios Core Development Team and Community Contributors  Copyright (c) 1999-2009 Ethan Galstad  Last Modified: 08-19-2015  License: GPL  Website: https://www.nagios.org  Reading configuration data...  Read main config file okay...  Read object config files okay...  Running pre-flight check on configuration data...  ...  ...  ...  Total Warnings: 0  Total Errors: 0  Things look okay - No serious problems were detected during the pre-flight check |

* Configure Nagios to start every time the system starts

|  |
| --- |
| [root@centosServer ~]# chkconfig --add nagios  [root@centosServer ~]# chkconfig --level 35 nagios on |

* Start Nagios

|  |
| --- |
| [root@centosServer ~]# service nagios start |

* Access the web interface at: http://<server-ip>/nagios

# Installing nagiosgraph

nagiosgraph is an add-on package to Nagios that enables the display of historical data monitored by Nagios in a graphical format

## Pre-requisites

* Nagios Core and plugins must be installed as per Section 3
* Install the following developer tools:

|  |
| --- |
| [root@centosServer ~]# yum -y install perl-GD php-gd rrdtool-perl rrdtool-php rrdtool perl-CGI perl-Time-HiRes |

* Download the latest version of nagiosgraph from https://exchange.nagios.org and extract it to a temporary location like "~/Downloads"

|  |
| --- |
| [root@centosServer Downloads]# tar -xzf <nagiosgraph-x.x.x>.tar.gz |

* Install CPAN if not present, and download the Nagios::Config module. Answer all prompts with default values.

|  |
| --- |
| [root@centosServer ~]# yum -y install perl-CPAN  [root@centosServer ~]# cpan Nagios::Config |

* Check the pre-requisites for the installation of NagiosGraph. If any dependency check fails, ensure that it is installed before proceeding

|  |
| --- |
| [root@centosServer nagiosgraph-1.5.2]# **./install.pl --check-prereq**  checking required PERL modules  Carp...1.11  CGI...3.51  Data::Dumper...2.124  Digest::MD5...2.39  File::Basename...2.77  File::Find...1.14  MIME::Base64...3.08  POSIX...1.17  RRDs...1.3008  Time::HiRes...1.9721  checking optional PERL modules  GD...2.44  Nagios::Config...36  checking nagios installation  found nagios exectuable at /usr/local/nagios/bin/nagios  found nagios init script at /etc/init.d/nagios  checking web server installation  found apache executable at /usr/sbin/httpd  found apache init script at /etc/init.d/httpd |

## Install nagiosGraph

This procedure assumes that Nagios Core and plugins have been installed under "/usr/local/nagios". If this location is different, modify the --prefix flag of the install script accordingly. Answer the prompts in the install script - default answers were used when testing this procedure.

|  |
| --- |
| [root@centosServer nagiosgraph-1.5.2]# **./install.pl --layout overlay --prefix /usr/local/nagios/**  checking required PERL modules  Carp...1.11  CGI...3.51  Data::Dumper...2.124  Digest::MD5...2.39  File::Basename...2.77  File::Find...1.14  MIME::Base64...3.08  POSIX...1.17  RRDs...1.3008  Time::HiRes...1.9721  checking optional PERL modules  GD...2.44  Nagios::Config... 36  checking nagios installation  found nagios exectuable at /usr/local/nagios/bin/nagios  found nagios init script at /etc/init.d/nagios  checking web server installation  found apache executable at /usr/sbin/httpd  found apache init script at /etc/init.d/httpd  Destination directory (prefix)? [/usr/local/nagios]  Location of configuration files (etc-dir)? [/usr/local/nagios/etc/nagiosgraph]  Location of executables? [/usr/local/nagios/libexec]  Location of CGI scripts? [/usr/local/nagios/sbin]  Location of documentation (doc-dir)? [/usr/local/nagios/docs/nagiosgraph]  Location of examples? [/usr/local/nagios/docs/nagiosgraph/examples]  Location of CSS and JavaScript files? [/usr/local/nagios/share]  Location of utilities? [/usr/local/nagios/docs/nagiosgraph/util]  Location of state files (var-dir)? [/var/nagios]  Location of RRD files? [/var/nagios/rrd]  Location of log files (log-dir)? [/var/nagios]  Path of log file? [/var/nagios/nagiosgraph.log]  Path of CGI log file? [/var/nagios/nagiosgraph-cgi.log]  Base URL? [/nagios]  URL of CGI scripts? [/nagios/cgi-bin]  URL of CSS file? [/nagios/nagiosgraph.css]  URL of JavaScript file? [/nagios/nagiosgraph.js]  URL of Nagios CGI scripts? [/nagios/cgi-bin]  Path of Nagios performance data file? [/tmp/perfdata.log]  username or userid of Nagios user? [nagios]  username or userid of web server user? [apache]  Modify the Nagios configuration? [n]  Modify the Apache configuration? [n]  configuration:  ng\_prefix /usr/local/nagios  ng\_etc\_dir /usr/local/nagios/etc/nagiosgraph  ng\_bin\_dir /usr/local/nagios/libexec  ng\_cgi\_dir /usr/local/nagios/sbin  ng\_doc\_dir /usr/local/nagios/docs/nagiosgraph  ng\_examples\_dir /usr/local/nagios/docs/nagiosgraph/examples  ng\_www\_dir /usr/local/nagios/share  ng\_util\_dir /usr/local/nagios/docs/nagiosgraph/util  ng\_var\_dir /var/nagios  ng\_rrd\_dir /var/nagios/rrd  ng\_log\_dir /var/nagios  ng\_log\_file /var/nagios/nagiosgraph.log  ng\_cgilog\_file /var/nagios/nagiosgraph-cgi.log  ng\_url /nagios  ng\_cgi\_url /nagios/cgi-bin  ng\_css\_url /nagios/nagiosgraph.css  ng\_js\_url /nagios/nagiosgraph.js  nagios\_cgi\_url /nagios/cgi-bin  nagios\_perfdata\_file /tmp/perfdata.log  nagios\_user nagios  www\_user apache  modify\_nagios\_config n  nagios\_config\_file  nagios\_commands\_file  modify\_apache\_config n  apache\_config\_dir  apache\_config\_file  Continue with this configuration? [y] y |

* In order for nagiosgraph to work correctly, the selinux policy should be in 'permissive' or 'disabled' mode.

|  |
| --- |
| [root@centosServer objects]# vim /etc/sysconfig/selinux  # This file controls the state of SELinux on the system.  # SELINUX= can take one of these three values:  # enforcing - SELinux security policy is enforced.  # permissive - SELinux prints warnings instead of enforcing.  # disabled - No SELinux policy is loaded.  SELINUX=disabled  # SELINUXTYPE= can take one of these two values:  # targeted - Targeted processes are protected,  # mls - Multi Level Security protection.  SELINUXTYPE=targeted |

To make this change permanent, restart the server

## Configure Nagios for NagiosGraph

* Edit the following six parameters in 'usr/local/nagios/etc/nagios.cfg" to allow it to process performance data and pass it to nagiosgraph

|  |
| --- |
| process\_performance\_data=1  service\_perfdata\_file=/tmp/perfdata.log  service\_perfdata\_file\_template=$LASTSERVICECHECK$||$HOSTNAME$||$SERVICEDESC$||$SERVICEOUTPUT$||$SERVICEPERFDATA$  service\_perfdata\_file\_mode=a  service\_perfdata\_file\_processing\_interval=30    service\_perfdata\_file\_processing\_command=process-service-perfdata-for-nagiosgraph |

* Add a command in the nagios commands object file "/usr/local/nagios/etc/objects/commands.cfg" to define the "process-service-perfdata-for-nagiosgraph" command:

|  |
| --- |
| # command to process nagios performance data for nagiosgraph  define command {  command\_name process-service-perfdata-for-nagiosgraph  command\_line /usr/local/nagios/libexec/insert.pl  } |

* In order to redirect the outputs of services checked by Nagios to nagiosgraph, create a service definition for a graphed service in "/usr/local/nagios/etc/objects/templates.cfg"

|  |
| --- |
| # graphed service  define service {  name graphed-service  action\_url /nagios/cgi-bin/show.cgi?host=$HOSTNAME$&service=$SERVICEDESC$&geom=1000x200' onMouseOver='showGraphPopup(this)' onMouseOut='hideGraphPopup()' rel='/nagios/cgi-bin/showgraph.cgi?host=$HOSTNAME$&service=$SERVICEDESC$&period=week&rrdopts=-w+450+-j  register 0  } |

* Finally, add or modify a service definition to use the graphed service template, for example modify the "PING" service for the localhost checks in "/usr/local/nagios/etc/objects/localhost.cfg"

|  |
| --- |
| # Define a service to "ping" the local machine  define service{  use local-service, graphed-service ; Name of service template to use  host\_name localhost  service\_description PING  check\_command check\_ping!100.0,20%!500.0,60%  } |

## Verification

* Perform a pre-flight check to ensure that all commands have been added in correctly

|  |
| --- |
| [root@centosServer ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg  Nagios Core 4.1.1  Copyright (c) 2009-present Nagios Core Development Team and Community Contributors  Copyright (c) 1999-2009 Ethan Galstad  Last Modified: 08-19-2015  License: GPL  Website: https://www.nagios.org  Reading configuration data...  Read main config file okay...  Read object config files okay...  Running pre-flight check on configuration data...  ...  ...  ...  Total Warnings: 0  Total Errors: 0  Things look okay - No serious problems were detected during the pre-flight check |

* Restart the Nagios service for it to pick up the changes

|  |
| --- |
| [root@centosServer objects]# **service nagios restart**  Running configuration check...  Stopping nagios:. done.  Starting nagios: done. |

* To verify that nagiosgraph installed correctly, go to "http://localhost/nagios/cgi-bin/showconfig.cgi". Check that all the fields have a status "OK". Note that the nagiosgraph log file may not be present if no services have been configured to use nagiosgraph.
* When Nagios starts up, navigate to the Service view. The service that was added as a "graphed-service" should have an icon next to it, which when clicked on, should display the historical output for that service in a graphical format.
* For any errors:
  + Check the nagiosgraph log file, whose location can be found at "/usr/local/nagios/etc/nagiosgraph/nagiosgraph.conf" under the property "logfile"
  + Check if RRD files are being generated, these can be found at "/usr/local/nagios/etc/nagiosgraph/nagiosgraph.conf" under the property "rrddir"

# Solace Plugin for Nagios

The Solace plugin for Nagios is capable of monitoring the various parameters on the Solace Appliance, both at the system level and the message-vpn level.

The following points describe the main features of the Solace plugin for Nagios.

## Hosts and Services

In Nagios’ context, all the devices being monitored are classified as hosts or services.

* A host is used in order to define a physical server, workstation, etc that resides on the network. Hosts have an address (IP or MAC) that uniquely identifies them, and have one or more services running on them.
* A service can be running on one or more hosts. These can be
  + attributes of the host such as CPU load, disk space, etc
  + services provided by the host, such as HTTP, POP3,SSH, etc
  + other items associated with the host, such as DNS records

In order to comply with the guidelines for the Nagios framework, Solace routers, Message VPNs, and physical servers are displayed as host objects on the Nagios web interface. Attributes of these hosts, as well as services running on them are displayed as service objects on the web interface. Examples of services include:

* + Alarms, disk, redundancy, replication, etc. are services running on a Solace router;
  + Message VPN status, Message-spool, queues and topics are services running on the Message VPN;
  + CPU load, Disk usage, Swap usage, etc. are services running on the Nagios Server
  + net.Solace services hosted on servers.

Custom adapters are also configured to be displayed as individual hosts on the Web interface, while their respective functionality is displayed as a service.

## Host groups and Service groups

Hosts and services monitored by Nagios can be grouped in order for simplifying configuration or for display purposes on the web interface.

* Host groups: These are used to groups one or more hosts together.
* Service groups: These are used to group one or more hosts running the same service

These concepts are implemented by the Solace plugin for Nagios as follows:

* Host groups are used to group all the VPNs running on a Solace router.
* Service groups are used to group all the hosts running a net.Solace service. At a minimum, a service groups will contain three members – these are the Solace router and message VPN that the net.Solace service is connecting to, and one or more servers that are hosting the net.Solace service.

This enables a user to quickly identify the relationships between Solace routers and message VPNs, as well as the net.Solace services and the servers hosting them.

## Host and Service checks

Plugins developed for Nagios expose functionality for monitoring the availability of hosts and services – these are generally executables or scripts. Nagios periodically runs these scripts/executables at a configured interval of time. The syntax for running these checks, as well as the parameters to be passed to them is known as a command. These are defined in a configuration file and read by Nagios on start-up.

NOTE: There is necessary not a one-on-one correspondence between a check command and a script in the plugin. Different commands can be defined, which call the same script/ executable with different parameters.

The output of commands of a Nagios plugin can be one of the following:

|  |  |
| --- | --- |
| **Plugin Result** | **Meaning** |
| OK | The object being monitored is up |
| WARNING | The object being monitored is up, but has reached a warning threshold |
| CRITICAL | The object being monitored is either down or has reached a critical threshold |
| UNKNOWN | Invalid command line arguments were supplied to the plugin or other low-level failures have occurred that have prevented it from performing the operation |

Table 3: Outputs of a Nagios plugin

The output of a plugin is used by Nagios to determine the state of the host, as follows:

|  |  |
| --- | --- |
| **Plugin Result** | **Preliminary Host state** |
| OK | UP |
| WARNING | UP or DOWN |
| CRITICAL | DOWN |
| UNKNOWN | DOWN |

Table 4: Preliminary Host check states

The WARNING result is interpreted as the host is down, if the option use\_aggressive\_host\_checking is available.

If the preliminary host state is down, Nagios will attempt to check the host’s parent to see if the host is really down or if it is unreachable.

|  |  |  |
| --- | --- | --- |
| **Preliminary Host state** | **Parent Host state** | **Final Host state** |
| DOWN | At least one parent is UP | DOWN |
| DOWN | All parents are DOWN or UNREACHABLE | UNREACHABLE |

For more information on how Nagios distinguishes between DOWN and UNREACHABLE states, refer to <http://nagios.sourceforge.net/docs/3_0/networkreachability.html>

The state of a service directly translates to the output of a plugin – refer to Table 3 for details.

## Active and Passive checks

Nagios can monitor hosts and service in two ways: active and passive checks.

* An active check is one that is initiated by the Nagios process and run on a scheduled basis, or on-demand as needed. Active checks poll a host or service for status information.
* A passive check is one that is initiated by external applications or submitted to Nagios for processing.

For more information on active and passive checks, refer to <http://nagios.sourceforge.net/docs/3_0/activechecks.html> and <http://nagios.sourceforge.net/docs/3_0/passivechecks.html> respectively. Details of the active and passive checks performed by the Solace plugin for Nagios are described in the next section.

## Object Definition files

Object configuration files are used to define hosts, services, hostgroups, contacts, contactgroups, commands, etc. This is where the objects to be monitored are defined, as well as additional details of how to monitor them. These are specified in the Nagios main configuration file using the cfg\_file or cfg\_dir options:

|  |  |
| --- | --- |
| **Object Definition** | **Description** |
| **Commands** | This file is used to define commands used by Nagios to check the status of the various hosts and services |
| **Hosts and services** | This file is used to store definitions of hosts and services to be monitored by Nagios and their corresponding check commands |
| **Contacts** | This file is used to define who should be notified in the event of a problem that Nagios detects with a host or service. Contact groups can be used to specify a groups of contacts to be notified. |
| **Time Periods** | This file is used to control various aspects of the monitoring and alerting logic, such as specify working hours when service notifications and escalations are to be sent out, or suppressed such as public holidays |
| **Templates** | Template files use Nagios’s feature of Object inheritance. Templates are used when there are a large number of similar objects to be monitored. These can be hosts, such as a windows hosts or Linux host; services such as a graphed or non-graphed service; or a contact template. Once these templates are defined, they can be used in object definitions and their attributes need not be redefined every time. |

For more information on the above object definition files and how to configure them, refer to <http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html>

## Commands available for the Solace plugin

The Nagios plugin exposes a variety of functions in order to monitor the various components in the GBM infrastructure. These have been used to define commands to check status of hosts and services in the GBM infrastructure, and are listed below:

### Active Checks

Listed below are the active check commands defined for the Solace plugin, the corresponding script and its description and output. The commands are defined in the configuration file **/etc/objects/solace.cfg** under the Nagios installation directory.

Note that only the conditions where OK, WARNING and CRITICAL values are returned are listed and not the UNKNOWN value. This is because the UNKNOWN state is returned when incorrect parameters are passed to the plugin or on other error conditions which prevent the plugin from performing the check. In each of these conditions, a message is returned which specifies the error condition.

Table 5 lists the commands used for monitoring attributes, resources and services on a Solace Router:

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Plugin Script** | **Description** | **Return Value** |
| **alarm** | alarm.pl | Monitors the status of system alarms for a Solace router. | Returns OK if there are no alarms, or CRITICAL if any are found, with their details |
| **bridge\_uni** | bridge.pl | Monitors the status of a unidirectional VPN bridge on a Solace router. | Returns OK if the VPN bridge is up, and CRITICAL if the bridge is shutdown, or operationally down. |
| **bridge\_bi** | Monitors the status of a bidirectional VPN bridge on a Solace router. | Returns OK if the VPN bridge is up, and CRITICAL of the bridge is shutdown, or operationally down for either inbound, outbound or both directions. |
| **disk** | disk.pl | Monitors the status of the disks on a Solace router. | Returns OK if both disks are up and the RAID is in a redundant state; WARNING if the RAID is a non-redundant state but the disks are up, and CRITICAL if either of the disks are down |
| **env** | environment.pl | Monitors the status of the system environment for a Solace router, such as Chassis fan speed and CPU temperature | Returns OK if the all the system environment variables are within their configured thresholds; WARNING or CRITICAL if the warning or critical thresholds have been exceeded, respectively |
| **hardware** | hardware.pl | Monitors the status of the system hardware information for a Solace router, such as power redundancy, fibre channel, external disk states, etc | Returns OK if all the system hardware components are operational; WARNING if they are in a non-redundant state or critical if they are all down. |
| **iface\_eth** | iface.pl | Monitors the status of the network interface for the management port. This command can also be used to | Returns OK if the interface is up and CRITICAL if it is down |
| **Iface\_lag<#>** | Monitors the status of a network interface for the message backbone. | Returns OK if all the members in the LAG are up; WARNING or CRITICAL if the number of available/operational members in the lag have crossed their warning or critical thresholds respectively. Returns critical if all the members in the LAG are down. |
| **memory** | memory.pl | Monitors the memory usage on a Solace router | Returns OK if the memory usage is within configured thresholds; WARNING or CRITICAL if it has crossed the warning or critical thresholds respectively. |
| **message-spool** | message-spool.pl | Monitors the message-spool state and usage on a Solace router. | Returns OK if the message spool is up; CRITICAL if the message-spool is down on a router ;WARNING or CRITICAL if the message-spool parameters have crossed their warning or critical thresholds respectively. |
| **partition** | partition.pl | Monitors the state of a given disk partition on a Solace router. | Returns OK if the partition’s usage is within the configured threshold; WARNING or CRITICAL if it has crossed the warning or critical thresholds respectively. |
| **redundancy** | redundancy.pl | Monitors the state of redundancy on a Solace router | Returns OK if the redundancy service is up; CRITICAL if any of the parameters in the redundancy configuration are down |
| **check\_solace\_router\_pair** | Check\_tcp | Monitors the state of a Solace HA pair | Returns OK if the cluster is reachable – if either the primary or backup router are up; CRITICAL if the cluster is down. |

Table 5: Router Check commands

Table 6 lists the commands used for monitoring the status of attributes, resources and services on a message VPN:

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Plugin Script** | **Description** | **Actions** |
| **vpn** | vpn.pl | Monitors the status of a message VPN on a Solace router | Returns OK if the VPN is up and its parameters are within configured thresholds; WARNING or CRITICAL if the message-vpn parameters have exceeded their warning or critical thresholds respectively. |
| **queue** | queue.pl | Monitors the status of a queue configured on a Solace router | Returns OK if the queue is up and its spool usage is within the configured thresholds; WARNING or CRITICAL if the usage has exceeded the warning or critical thresholds, respectively. |
| **spool\_vpn** | message-spool\_vpn.pl | Monitors the status of the message-spool allocated to a message VPN | Returns OK if the message spool is configured for a VPN and its usage is within configured thresholds; WARNING or CRITICAL if it the usage has crossed the warning or critical thresholds respectively. |
| **replication** | replication.pl | Monitors the status of the replication service for a given message VPN | Returns OK if the replication service is up on a message VPN; CRITICAL if replication is not configured for the VPN, in a shutdown state, or if the replication queue is not bound. |

Table 6: Message VPN check commands

### Passive Checks

* **Admin listener:** The admin listener is a functionality provided by the net.Solace adapter. Whenever a client requests for information on a service topic, if it detects that no server is servicing the topic, an event is sent out on the topic: mon/AL/svc/<service-name>/\_RR/<GUID>

The admin listener command listens on this topic for each VPN being monitored. When an event is published on this topic, it collects it and submits a passive check to Nagios. Nagios will then send out a notification that the corresponding service is down, and reflect the state of the service on the Web Interface GUI. This alert indicates that a service is down, and is similar to the down event returned by the active check for a service.

The admin listener can be found at the following directory in the Nagios plugin "libexec/scripts"

# Installing the Nagios Plugin for Solace

This section contains the details of manually upgrading an existing installation of the Solace plugin for Nagios to a newer version. It is assumed that Nagios core, along with the Nagios plugin for Solace are already installed. The default location for the nagios install is /usr/local/nagios and this is referred to as [NAGIOS\_HOME].

Nagios will be installed by default under /usr/local/nagios – this will be referred to as [NAGIOS\_HOME] in the rest of the document. The nagios home directory will have a folder structure as follows:

|  |
| --- |
| [root@nagiosServer nagios]# ls -lrt  total 504  drwxrwxr-x. 2 nagios nagcmd 4096 Mar 27 2014 bin  drwxr-xr-x. 2 root nagcmd 4096 Mar 27 2014 include  drwxrwxr-x. 2 nagios nagcmd 4096 Mar 27 2014 sbin  drwxr-xr-x. 3 root root 4096 Mar 27 2014 docs  drwxr-xr-x. 3 root root 4096 Mar 30 2014 plugins  drwxrwxr-x. 11 nagios nagcmd 4096 May 15 04:34 share  drwxrwxr-x. 4 nagios nagcmd 4096 Jun 7 15:11 libexec  drwxrwxr-x. 5 nagios nagcmd 4096 Jun 9 05:50 etc  drwxrwxr-x. 6 nagios nagcmd 4096 Jun 9 06:09 var |

The Nagios plugin for Solace contains only the libexec and etc directories.

## Install required Perl modules

Nagios requires Perl to be installed - the current version of the plugin has been tested on Perl v5.10.

To check the installed version of Perl and if all the modules are installed, run the check script which can be found in the Solace plugin at "libexec/scripts/checkPerl.sh".

|  |
| --- |
| [nagios@nagiosServer nagios]# /usr/local/nagios/libexec/scripts/checkPerl.sh  Perl Version 5.10 detected  Config::Std ... OK  Data::Dumper ... OK  File::Basename ... OK  File::Slurp ... OK  File::Copy ... OK  Cwd ... OK  Getopt::Long ... OK  LWP::UserAgent ... OK  Nagios::Plugin ... OK  Nagios::Plugin::DieNicely ... OK  Pod::Usage ... OK  Scalar::Util ... OK  Switch ... OK  Try::Tiny ... OK  XML::LibXML ... OK  All required Perl modules are present. |

If a module is not installed, install it from CPAN – refer to <http://www.cpan.org/modules/INSTALL.html>

The list of required Perl modules is below:

* Config::Std
* Cwd
* Data::Dumper
* File::Basename
* File::Slurp
* File::Copy
* Getopt::Long
* LWP:UserAgent
* Nagios::Plugin
* Nagios::Plugin::DieNicely
* Pod::Usage
* Scalar::Util
* Try::Tiny
* XML::LibXML

## Stop Nagios

If Nagios is already running, stop it

In the event that the Nagios process needs to be shutdown, issue the following command:

|  |
| --- |
| [nagios@nagiosServer var]# service nagios stop  Stopping nagios: done. |

## Copy executables

Replace the contents of the [NAGIOS\_HOME]/libexec directory, which contains the scripts and executables for the Nagios plugin, with that of the libexec in the Solace plugin.

Ensure that all the binaries under this directory are executable. If not, run the below command to make them executable:

|  |
| --- |
| chmod +x <file name> |

## Copy config files

* Copy the below highlighted config files in the "etc" directory of the Solace Plugin to the [NAGIOS\_HOME]/etc directory.

|  |
| --- |
| [root@nagiosServer nagios]# ls -lrt etc  total 136  -rw-r--r--. 1 root nagcmd 46 Mar 27 2014 htpasswd.users  drwxr-xr-x. 2 nagios nagcmd 4096 Feb 2 17:43 nagiosgraph  -rw-rw----. 1 nagios nagcmd 2480 Jun 4 05:43 resource.cfg  -rw-rw-r--. 1 nagios nagcmd 12015 Jun 4 09:27 cgi.cfg  -rw-r--r--. 1 nagios nagcmd 45471 Jun 9 05:19 nagios.cfg  drwxrwxr-x. 6 nagios nagcmd 4096 Jun 9 05:33 objects |

* Copy the below config files from /etc/objects in the Solace Plugin to [NAGIOS\_HOME]/etc/objects. I

|  |
| --- |
| [root@nagiosServer nagios]# ls -lrt etc  total 72  drwxrwxrwx. 2 root root 4096 Sep 1 02:49 netSolaceServers  drwxrwxrwx. 2 root root 4096 Sep 1 03:18 solaceVpns  drwxrwxrwx. 2 root root 4096 Sep 1 03:29 solaceRouters  drwxrwxrwx. 2 root root 4096 Sep 1 03:35 solaceAdapters  drwxr-xr-x. 2 root root 4096 Sep 1 04:02 sample  -rwxrwxrwx. 1 root root 7185 Aug 19 00:34 localhost.cfg  -rwxrwxrwx. 1 root root 8200 Aug 28 04:22 commands.cfg  -rwxrwxrwx. 1 root root 3388 Aug 28 04:23 contacts.cfg  -rwxrwxrwx. 1 root root 5902 Aug 28 04:23 solace.cfg  -rwxrwxrwx. 1 root root 3479 Aug 28 04:24 timeperiods.cfg  -rwxrwxrwx. 1 root root 12152 Aug 31 23:09 templates.cfg |

* Edit the configuration files:
  + Edit the variable $USER100$ in [NAGIOS\_HOME]/etc/resource.cfg to reflect the sender email address for notifications sent out by Nagios.
  + Edit [NAGIOS\_HOME]/etc/nagios.cfg to add the below lines in the section for configuration directories:

**Sample:**

|  |
| --- |
| # OBJECT CONFIGURATION FILE(S)  # These are the object configuration files in which you define hosts,  # host groups, contacts, contact groups, services, etc.  # You can split your object definitions across several config files  # if you wish (as shown below), or keep them all in a single config file.  # 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  # Path to .cfg file containing command definitions for the Solace Plugin:  cfg\_file=/usr/local/nagios/etc/objects/solace.cfg  # Path to .cfg file containing object definitions for  # monitoring the local host  cfg\_file=/usr/local/nagios/etc/objects/localhost.cfg  # You can also tell Nagios to process all config files (with a .cfg  # extension) in a particular directory by using the cfg\_dir  # directive as shown below:  # Path to directory containing object definitions for  # Solace Routers  cfg\_dir=/usr/local/nagios/etc/objects/solaceRouters  # Path to directory containing object definitions for  # Solace VPNs  cfg\_dir=/usr/local/nagios/etc/objects/solaceVpns  # This directory contains definitions for monitoring  # a Windows machine, router/switch and network printer  # cfg\_dir=/usr/local/nagios/etc/objects/sample  #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 |

* Edit [NAGIOS\_HOME]/etc/objects/contacts.cfg to configure the email address that receives notifications sent out by Nagios.

## Set up Credentials Directory

The Credentials directory is used to store sensitive information such as passwords for logging into the appliance and VPNs to be monitored. These files should be given read-only permissions by users other than Nagios in order to prevent unauthorized access. The path to this directory should be defined in the resources.cfg configuration file, so that the Solace plugin for Nagios can pick it up on start-up. The Solace plugin will read all the files in this directory and pick up the appropriate information.

|  |
| --- |
| [nagios@nagiosServer config]# ls -lrt  total 12  -rw-r--r--. 1 nagios nagcmd 152 Apr 23 04:41 localhost.cfg  -rw-r--r--. 1 nagios nagcmd 293 Jun 10 23:45 GBMSOLACEDEV01.cfg  -rw-r--r--. 1 nagios nagcmd 301 Jun 10 23:46 GBMSOLACEDEV02.cfg |

Create a separate credentials file with the information on each solace router to be monitored, and one for the Nagios Server (localhost). These files can be given read-only permissions as they contain sensitive information.

For sample credential files, refer to [NAGIOS\_HOME]/etc/samples/sampleCredentials/ and copy them to the directory which will contain credential files for your install.

Once added, updated $USER5$ variable in [NAGIOS\_HOME]/etc/resource.cfg with the path to the credentials directory.

## Create object configurations

This section walks through a simple scenario for setting up one router, one message-VPN and an instance of a Solace adapter. The setup can be done using setup scripts that can be found in the directory: [NAGIOS\_HOME]/libexec/scripts/createNagiosObjectConfigs/. These scripts prompt the user for input values and generate Object definition files for Nagios. Some of these questions will have default values for the answers within box brackets ([ ]).

All scripts may be run independently, except the when creating clusters – the members of the clusters (routers, VPNs and adapters) must be set up prior to running the script.

All the create scripts generate object definition files for default values of the thresholds. If these are to be changed, please refer to Section 6.6.2.

The directory has the following structure:

|  |
| --- |
| drwxr-xr-x. 2 root root 4096 Aug 27 03:54 common  -rwxr-xr-x. 1 root root 3088 Aug 19 01:00 createSolaceAdapter.pl  -rwxr-xr-x. 1 root root 3205 Aug 27 05:40 createSolaceAdapterCluster.pl  -rwxr-xr-x. 1 root root 2726 Aug 19 01:00 createSolaceRouter.pl  -rwxr-xr-x. 1 root root 2139 Aug 27 05:34 createSolaceRouterPair.pl  -rwxr-xr-x. 1 root root 3361 Aug 21 05:00 createSolaceVpn.pl  drwxr-xr-x. 2 root root 4096 Aug 27 05:35 outputs  drwxr-xr-x. 2 root root 4096 Aug 27 04:08 templates |

* The ‘templates’ directory contains templates for the sample configuration files. These should not be modified.
* The ‘common’ directory contains utility and helper functions used by the create scripts
* The ‘output’ directory is where the outputs of the scripts will be placed.

There are five types of “create” scripts which can be used to generated object definition files.

* createSolaceRouterConfig.pl – Creates a Nagios object definition for a Solace router
* createSolaceVpnConfig.pl – Creates a Nagios object definition for a Solace Message-vpn

**Note:** The Solace plugins allow you to configure a display name for the message-vpn that can be different from the actual name of the VPN on the router. This is optional and allows you to easily identify a VPN at the Nagios Web UI however it is recommended that these are kept the same. If different values are used for the message-vpn name on the appliance and the display name, care should be taken to ensure that the appropriate values are supplied when creating the configuration objects.

The operation of these scripts is described in the following sections:

#### Configuring a router object

* Prior to running this script, the following values must be known:
* Name of the Solace router
* Management IP address and port
* Management and Message backbone interface to be monitored (can be obtained by running “show interface” on the Solace router CLI)
* File system partition to be monitored (can be obtained by running “show disk details on the Solace router CLI)
* Names of VPNs in the router to be monitored by Nagios
* Run the createSolaceRouter.pl which can be found at: [NAGIOS\_HOME]/libexec/scripts/createNagiosObjectConfigs/. The script will prompt the user for values required for the generation of the file.

|  |
| --- |
| [root@nagiosServer createNagiosObjectConfigs]# ./createSolaceRouterConfig.pl  Enter the router name: **GBMSOLACEDEV01**  Enter the management IP address in dotted decimal format (xx.xx.xx.xx): **118.189.120.250**  Enter the port number for the management IP: [80] **1380**  Enter the name of the management interface to be monitored: [eth1]  Enter the name of the message-backbone interface to be monitored: [1/1/lag1]  Enter the name of the file system partition to be monitored: [/dev/md2]  Enter the name of the hostgroup for this router: [GBMSOLACEDEV01]  Enter the Nagios hostnames(display names) of VPNs to be monitored in this router(comma separated): **VPN\_GBM\_DEMO1,VPN\_GBM\_DEMO2**  The Object configuration file for the Solace router has been created at: /usr/local/nagios/libexec/scripts/createNagiosObjectConfigs/outputs/GBMSOLACEDEV01.cfg |

**Note:** The output script will be named using the hostname of the router – if the file already exists, it will be overwritten.

* Copy the generated output file to [NAGIOS\_HOME]/etc/objects/solaceRouters/. If any mistakes were made during the generation of the file, these can be manually corrected if required.
* The script only caters for monitoring a single management and message backbone interface for the router. For monitoring additional interfaces:
  + Create a new user defined macro for the interface name by copying the below in Section 1:
    - \_iface\_mgmt<N> for monitoring an additional management interface;
    - \_iface\_msgb<N> for monitoring an addition message backbone interface;
  + Change the value of N to a unique one
  + Copy the service definition for an interface in Section 4 and modify the 3rd argument to reflect the newly created macros
  + Restart the Nagios service.
* The script only caters for monitoring a single file system partition for the router. For monitoring additional partitions:
  + Create a new user defined macro for the partition name by copying the below in Section 1:
    - \_partition<N>\_fs to reflect the partition name
    - \_partition<N>\_wu to reflect the warning threshold on file system usage
    - \_partition<N>\_cu to reflect the critical threshold on file system usage
  + Change the value of N to a unique one
  + Copy the service definition in Section 4 for a partition and modify the 3rd, 4th and 5th arguments to reflect the newly created macro
  + Restart the Nagios service.

#### Configuring a message VPN object

* Prior to running this script, the following values must be known:
* Name of the Message-VPN on the Solace router (can be obtained by running “show message-vpn \* ” on the Solace router CLI)
* Optional – Display name of the VPN in Nagios
* Name of the Solace router hosting this VPN
* Management and message-backbone IP address and port
* Management and Message backbone interface to be monitored (can be obtained by running “show interface” on the Solace router CLI)
* Names of queues to be monitored on this VPN (can be obtained by running “show queue \* message-vpn <name>” on the Solace router CLI)
* Names of VPN bridges to be monitored on this VPN (can be obtained by running “show bridge \* message-vpn <name>”
* Run the createSolaceVpn.pl which can be found at: [NAGIOS\_HOME]/libexec/scripts/createNagiosObjectConfigs/. The script will prompt the user for values required for the generation of the file.

|  |
| --- |
| [root@nagiosServer createNagiosObjectConfigs]# ./createSolaceVpnConfig.pl  Enter the name of the message-vpn on the router: **VPN\_GBM\_DEMO1**  Enter the Nagios host name (display name) of the VPN: [**VPN\_GBM\_DEMO1**]  Enter the management IP address in dotted decimal format (xx.xx.xx.xx): **118.189.120.250**  Enter the port number for the management IP: [80] **1380**  Enter the name of the router hosting this VPN: **GBMSOLACEDEV01**  Enter the message-backbone IP in dotted decimal format (xx.xx.xx.xx): **118.189.120.253**  Enter the port number for the message-backbone IP: [55555]  Are there any queues to be monitored on this VPN?(Y/N) [Y] **y**  Enter the name of the queue: **GBM\_DEMO\_QUEUE01**  Are there any VPN bridges to be monitored on this VPN?(Y/N) [Y] **y**  Enter the name of the VPN Bridge: **GBM\_BRIDGE01**  The Object configuration file for the Solace router has been created at: /usr/local/nagios/libexec/scripts/createNagiosObjectConfigs/outputs/VPN\_GBM\_DEMO1.cfg  If the there are no queues or VPN bridges to be monitored, please delete the appropriate service checks from the file. For monitoring additional queues and VPN bridges, refer to the Nagios Operational Guide |

**Note:** The output script will be named using the hostname of the VPN – if the file already exists, it will be overwritten.

* Copy the generated output file to [NAGIOS\_HOME]/etc/objects/solaceVpns/. If any mistakes were made during the generation of the file, these can be manually corrected if required.
* This script only caters for the monitoring of a single queue on the VPN. For monitoring additional queues:
  + Create macros for the new queue definition by copying the below in Section 1:
    - \_queue<N>
    - \_queue<N>\_wd
    - \_queue<N>\_cd
  + Change the value of N to a unique one
  + Copy the service definition for a queue in Section 2 and modify the 4th, 5th and 6th arguments to reflect the newly created macros
  + Restart the Nagios service.
* This script only caters for the monitoring of a single VPN bridge. For monitoring additional VPN bridges:
  + Create a macro for the VPN bridge name by copying the below in Section 1:
    - \_bridge<N>
  + Change the value of N to a unique one
  + Copy the service definition for a VPN bridge in Section 2 and modify the 3rd argument to reflect the newly created macro
  + Restart the Nagios Service.
* Section 4 onwards contains definitions of services discovered by service discovery. These lines should not be modified during the normal operation of Nagios, unless if resetting service discovery. For resetting service discovery, refer to Section **Error! Reference source not found.**

### Thresholds for monitoring services

In Section 1 of each sample host definition file, there are user defined macros which are used to set the warning and critical thresholds for services being monitored on that host. In general, the output of the service is determined as follows:

* OK, if the output of the check is lesser than the warning threshold;
* WARNING, if the output of the check is greater than the warning but lesser than the critical threshold
* CRITICAL, if the output of the check is greater than the critical threshold

The threshold that nagios checks the output of a service against can be either a floor or a ceiling.

The below example specifies the threshold for memory usage as a ceiling – when the memory usage exceeds the warning threshold of 60%, a WARNING alert will be generated. When it exceeds the critical threshold of 80%, a CRITICAL alert will be generated.

|  |
| --- |
| #User defined macros for memory checks- Update these parameters as required  \_mem\_w 60 ; Warning threshold for memory (%)  \_mem\_c 80 ; Critical threshold for memory (%) |

The below example specifies the threshold for the number of power supplies as a floor – when the number of power supplies goes below the warning threshold of 2, a WARNING alert will be generated. When it goes below the critical threshold of 1, a CRITICAL alert will be generated. In order to specify a threshold as a floor, add a colon “:” after the threshold value in the object configuration file.

|  |
| --- |
| #User defined macros for hardware checks- Update these thresholds as required  \_hw\_wps 2: ; Power Supply Warning    \_hw\_cps 1: ; Power Supply Critical |

## Verification

Once Nagios has been configured, the configuration can be verified as described below:

### Object definitions

Verify that the below object definition files are present, and configured as required:

|  |  |  |
| --- | --- | --- |
| **Configuration file** | **Description** | **Path** |
| **nagios.cfg** | Nagios main configuration file | /usr/local/nagios/etc/nagios.cfg |
| **resource.cfg** | Resource file | /usr/local/nagios/etc/resource.cfg |
| **commands.cfg** | Nagios command definition | /usr/local/nagios/etc/objects/commands.cfg |
| **contacts.cfg** | Nagios contacts definition | /usr/local/nagios/etc/objects/contacts.cfg |
| **timeperiods.cfg** | Nagios timeperiods definition | /usr/local/nagios/etc/objects/timeperiods.cfg |
| **solace.cfg** | Command definition for the nagios plugin for Solace | /usr/local/nagios/etc/objects/solace.cfg |
| **Solace router definitions** | These contain the host definition of Solace Routers to be monitored | /usr/local/nagios/etc/objects/solaceRouters/\*.cfg |
| **Solace VPN definitions** | These contain the host definition of Solace VPNs to be monitored | /usr/local/nagios/etc/objects/solaceVpns/\*.cfg |
| **Credentials files** | These contain the credentials for Nagios to log onto the hosts being monitored | Defined in resource.cfg |

### File permissions

All configuration files should be readable and writable by the 'nagios' user, as these files are read and written to with status information during the monitoring process. All files should be owner by the nagios user, and should have read-write permissions (666).

To change the ownership of a file to the nagios user and set correct permissions, run the below commands:

|  |
| --- |
| [root@nagiosServer ~]# chown nagios <file-name>  [root@nagiosServer ~]# chgrp nagcmd <filename>  [root@nagiosServer ~]# chmod 666 <filename> |

The resource config file can be given read-only permissions by the nagios user as it contains sensitive information.

### Nagios pre-flight check

The nagios pre-flight check can be used to verify basic Nagios configuration -

|  |
| --- |
| [root@nagiosServer ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg |

Any errors in configuration files (such as syntax errors or circular references) will be displayed – these need to be fixed before starting the Nagios service. If the pre-flight check completes without any errors, nagios is properly configured to be started.

### Network level verification

Ensure that Nagios is able to reach the message-backbone and management IP address of all Solace routers which are to be monitored. In addition, the nagios server should be able to reach all the servers which will be hosting services – they should have ICMP enabled.

In order to check network reachability, use the PING command:

|  |
| --- |
| [nagios@nagiosServer ~]$ ping 10.0.1.121  PING 10.0.1.121 (10.0.1.121) 56(84) bytes of data.  64 bytes from 10.0.1.121: icmp\_seq=1 ttl=128 time=1.03 ms  64 bytes from 10.0.1.121: icmp\_seq=2 ttl=128 time=0.756 ms  64 bytes from 10.0.1.121: icmp\_seq=3 ttl=128 time=0.792 ms  64 bytes from 10.0.1.121: icmp\_seq=4 ttl=128 time=0.771 ms  ^C  --- 10.0.1.121 ping statistics ---  4 packets transmitted, 4 received, 0% packet loss, time 3480ms  rtt min/avg/max/mdev = 0.756/0.837/1.032/0.118 ms |

## Start the Nagios service

The Nagios core package, when installed, adds the nagios service script to /etc/init.d. In order to start the nagios service, simply start this service as follows. Nagios should be run as the nagios user, and not root.

|  |
| --- |
| [nagios@nagiosServer var]# sudo service nagios start  Starting nagios: done. |

If there are any errors in the Nagios configuration, these will be displayed and these will have to be resolved before starting Nagios. If not, the Nagios process has been successfully started.

Check the logs to validate that Nagios has started successfully. The default location of the log file is [NAGIOS\_HOME]/var/nagios.log.

|  |
| --- |
| [1433749600] Nagios 4.0.4 starting... (PID=9969)  [1433749600] Local time is Mon Jun 08 15:46:40 SGT 2015  [1433749600] LOG VERSION: 2.0  [1433749600] qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized  [1433749600] qh: core query handler registered  [1433749600] nerd: Channel hostchecks registered successfully  [1433749600] nerd: Channel servicechecks registered successfully  [1433749600] nerd: Channel opathchecks registered successfully  **[1433749600] nerd: Fully initialized and ready to rock!**  [1433749600] wproc: Successfully registered manager as @wproc with query handler  [1433749600] wproc: Registry request: name=Core Worker 9974;pid=9974  [1433749600] wproc: Registry request: name=Core Worker 9973;pid=9973  [1433749600] wproc: Registry request: name=Core Worker 9972;pid=9972  [1433749600] wproc: Registry request: name=Core Worker 9971;pid=9971 |

Additionally, check that the nagios process is running:

|  |
| --- |
| [root@nagiosServer var]# ps -ef | grep nagios.cfg  nagios 9969 1 0 15:46 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg  nagios 9975 9969 0 15:46 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg  root 10213 5895 0 15:48 pts/4 00:00:00 grep nagios.cfg |

# Additional Documentation

For more information on additional tasks such as troubleshooting common errors and operational maintenance tasks, refer to the comprehensive Nagios Runbook created for GBM, which can be found at: svn://192.168.1.202/svn/pro-serv/trunk/GBM/Monitoring/Nagios/Nagios-GBM-RunBook.docx

# Outstanding Issues

* Perl module Nagios::Plugin is tombstoned in CPAN - this needs to be replaced with Monitoring::Plugin

Approvals

This document has been read and approved by the following people, responsible for its implementation. Approval is indicated by an email showing approval. Those approving below indicate that the contents of this document are correct and complete and agree to their implementation:

| Title | Name | Approval |
| --- | --- | --- |
|  |  |  |

History

| Version | Status | Date | Author | Reason for changes |
| --- | --- | --- | --- | --- |
| 0.1 | Draft | Sep 11th, 2015 | Shrikanth Rajgopalan | Initial Draft |
| 0.2 | Draft | Oct 7th,2015 | Shrikanth Rajgopalan | Added details on installing the Nagios plugin |