DEPLOYMENT GUIDE
Version 1.0

Deploying the BIG-IP LTM with the Nagios Open Source Network Monitoring System

Deploying F5 with Nagios Open Source Network Monitoring System

Welcome to the F5 and Nagios deployment guide. This guide provides detailed procedures for configuring F5 devices with the Nagios Open Source Network Monitoring System.

Network Management Systems (NMS) provide insight to device and application performance, system stability, network trouble spots and a host of other items that affect a company's ability to provide computer network services. Recent advances in automated event correlation, data storage and retrieval, as well as open source systems that allow greater integration, have greatly improved today's Network Management Systems.

The Nagios system and network monitoring application is an open source project. Nagios maintains network status information by polling objects for specific service responses. Examples of monitored network services include SMTP, DNS, HTTP, and so on.

For more information on the Nagios project, see ***http://www.nagios.org/*** or ***http://sourceforge.net/projects/nagios/***.

For further Nagios resources and information, see ***http://community.nagios.org/***.

For more information on F5 devices described in this guide, see ***http://www.f5.com/products/big-ip/.***

To provide feedback on this deployment guide or other F5 solution documents, contact us at ***solutionsfeedback@f5.com***.

Prerequisites and configuration notes

The following are prerequisites and configuration notes about this deployment:

- The BIG-IP LTM system must be running v9.0 or later. We highly recommend using v9.4 or later. Examples shown in this document are from a v10.0 system.
- As with most open source projects newer Nagios version may exist, examples shown in this document are from a Nagios 3.0.6 deployment running on a Linux Ubuntu 8.10 server.
- This document is written with the assumption that the reader is familiar with BIG-IP LTM systems, open source tools, Linux server systems administration and the Nagios monitoring platform.
- Additionally, the system administrator should be familiar with the basic concepts of the Simple Network Management Protocol (SNMP). For more information on configuring these products, consult the appropriate documentation.

Product versions and revision history

Product and versions tested for this deployment guide:

Product Tested	Version Tested
BIG-IP Local Traffic Manager (LTM)	v10.0
Nagios Network Graphing Solution	v3.0.6
SNMP	v2c

Revision history:

Document Version	Description
1.0	New deployment guide

Configuration example

The BIG-IP LTM system is used to add high availability and traffic management to a pool of servers providing network services and enterprise applications such as web (HTTP) traffic. Many network administrators deploy management solutions on a dedicated network to monitor and managed their hardware. A typically deployment of a monitored BIG-IP LTM is shown in the following illustration.

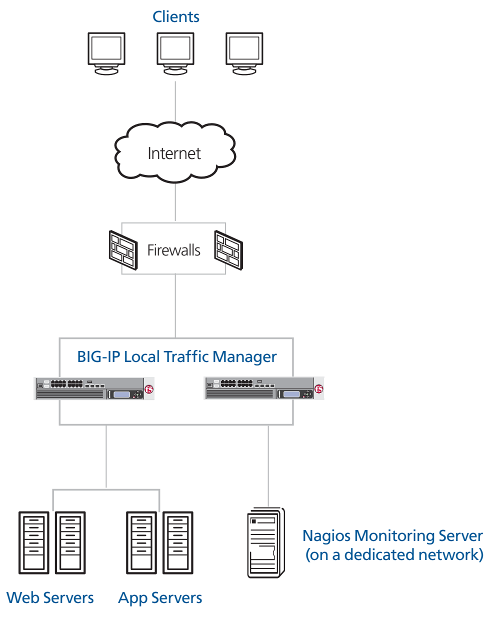


Figure 1 *Logical configuration example*

Configuring the BIG-IP LTM for monitoring with the Nagios Network Graphing Solution

This deployment guide is divided into the following sections:

- *Performing the initial configuration tasks*, on page 3
- *Adding a BIG-IP LTM host to the Nagios monitored environment*, on page 6
- *Advanced configuration*, on page 10

We recommend you save your existing BIG-IP configuration before you begin the procedures in this Deployment Guide. For information on backing up or restoring a BIG-IP LTM configuration, refer to the appropriate BIG-IP LTM manual, available on *Ask F5*.

Performing the initial configuration tasks

In this section, we configure the BIG-IP LTM with the required SNMP configuration to allow communication from the Nagios monitoring server.

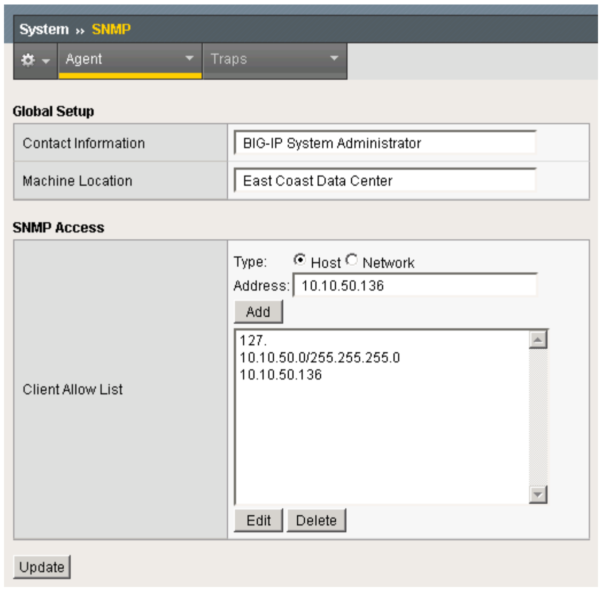
Configuring SNMP

The first procedure in this deployment is allow the BIG-IP LTM system to accept requests and reply with data to specific SNMP requests from the monitoring system.

To allow SNMP Communication

1. On the Main tab, expand **System**, and then click **SNMP**.
   The SNMP configuration screen opens.
2. In the **Contact Information** box, type the name of the person who administers the SNMP service for this system. In our example, we type **BIG-IP System Administrator**.
3. In the Machine Location box, type the location of this device. In our example, we type **East Coast Data Center.**
4. From the Client Allow List section, in the **Address** box, type the address of your Nagios server, and then click the **Add** button.

   Note: *You may want to enter the entire monitoring network IP address range in this step. This is useful if you have additional monitoring systems on the management network that are also monitoring the BIG-IP LTM. To provide a more secure environment the administrator may wish to add only individual IP addresses. Examples of both conditions are shown in Figure 2.*
5. Click the **Update** button.



Global Setup

Contact Information	BIG-IP System Administrator
Machine Location	East Coast Data Center

SNMP Access

Type: [✓ Host] Network
Address: 10.10.50.136

Client Allow List:
127.
10.10.50.0/255.255.255.0
10.10.50.136

Figure 2 *Configuring SNMP information on the BIG-IP LTM*

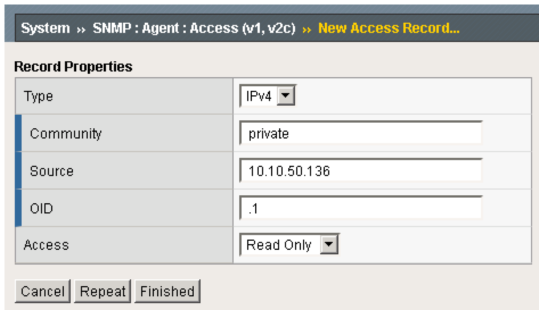
Configuring the community name strings

Next, we need to set the community name strings to allow the SNMP agent on the BIG-IP LTM to communicate with the Nagios server. The community name strings are essentially passwords used to authenticate an SNMP **GET** request.

To configure the community name strings

1. On the Main tab, expand **System**, and then click **SNMP**.
   The SNMP configuration screen opens.
2. From the Menu bar, point your mouse at **Agent**, and then click **Access (v1, v2c)**.
3. Click the **Create** button.
   The New Access Record page opens.
4. From the **Type** list, select the appropriate IP version for your implementation. While IPv6 plug-ins exist for Nagios, they are not part of the standard Nagios implementation. In our example, we leave the IP version at the default setting, **IPv4**.
5. In the **Community** box, type a name for the community. In our example, we type **private**.
6. In the **Source** box, type the IP address of the Nagios monitoring server. In our example, we type **10.10.50.136**.

7. In the **OID** box, type the start of the object identifier (OID) you want to monitor. You must at least type **.1** in this box (which is the first portion of any OID string).
8. From the Access list, select the appropriate access level. In our example, we select **Read Only**.
9. Click the **Finished** button.


System » SNMP : Agent : Access (v1, v2c) » New Access Record...
Cancel
Repeat
Finished

Record Properties

Type	IPv4
Community	private
Source	10.10.50.136
OID	.1
Access	Read Only

Figure 3

This concludes the initial configuration tasks.

Adding a BIG-IP LTM host to the Nagios monitored environment

The Nagios platform provides methods to monitor particular devices or services. Service monitors (HTTP, FTP, and so on) can quickly be established using the plug-ins available with the distribution. Plug-ins for Nagios are Perl or other similar scripts that, when executed, send a specific data query to a device. The value returned from the plug-in is what Nagios tracks. Nagios, by itself, does not what it is monitoring, or the actual state of a monitored device or service; the plug-ins do all the work. Fortunately, many plug-ins have been written and are available for downloads. The following locations maintain the latest plug-ins:

- Nagios Plugins Project: ***http://nagiosplug.sourceforge.net/***
- Nagios Downloads Page: ***http://www.nagios.org/download/***
- NagiosExchange.org: ***http://www.nagiosexchange.org/***

Establishing a Service Monitor for BIG-IP LTM

The Nagios community makes available many pre-configured monitors to assist you in configuring your system. In the following example, we modify the network device monitoring scripts for routers and switches to quickly establish a set of monitors for the BIG-IP LTM.

To establish a service monitor for the BIG-IP LTM

1. The main Nagios configuration file for the Ubuntu server configuration is found in the **/usr/local/nagios/etc/** directory. Edit this file to remove the comment (**#**) from the statement

```
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
```

   This defines the path Nagios uses to look for object definitions that define the types of monitors to be enabled.

2. To enable the object definitions, edit the **switch.cfg** file to add the BIG-IP device we want to monitor. This file is found in the **/usr/local/nagios/etc/objects/** directory.

   In this step, we add the host_name, alias, address, as well as the group information to the **hostgroups** line. Substitute the values from the following example values with the values appropriate for your configuration.

   Note: *Nagios does not use DNS for identifying or polling particular devices, only the IP address. The host name entered in this file is the one that will be displayed in the main Nagios status page and is not necessarily associated with a DNS entry.*

```
define host{
use                    generic-switch   ; Inherit default values from a template
host_name              BIG-IP LTM       ; The name we are giving to this switch
alias                  BIG-IP LTM V10   ; A longer name associated with the switch
address                10.10.50.135     ; IP address of the switch
hostgroups             switches         ; Host groups this switch is associated with
}
```

This completes the host definition.

3. Next, we add **service definitions** to the **switch.cfg** configuration file. Service definitions allow us to define what will be monitored for a particular host. While still in the **switch.cfg** file, use the following example to create an ICMP echo request to the defined host. This monitor collects round trip time information and reports packet loss for the host.

```
# Create a service to PING to switch

define service{
use                    generic-service; Inherit values from a template
host_name              BIG-IP LTM; The name of the host the service is associated with
service_description    PING    ; The service description
check_command          check_ping!200.0,20%!600.0,60%; The command to monitor the service
normal_check_interval  5 ; Check the service every 5 minutes under normal conditions
retry_check_interval   1 ; Re-check the service every minute until its final/hard state is
                           determined
}
```

Pay attention to the **check_command check_ping** entry. The values **200.0,20%!600.0,60%** dictate the status display for this service on the main Nagios page. The service will be marked CRITICAL if the round trip average (RTA) is greater than 600 milliseconds or the packet loss is 60% or more; marked WARNING if the RTA is greater than 200 ms or the packet loss is 20% or more; or marked OK if the RTA is less than 200 ms and the packet loss is less than 20%. Any of these values can be edited to better tailor this monitor to your specific environment. In our example, we use the defaults as shown.

4. Add a statement which uses the **check_snmp** plugin to query the BIG-IP host we defined for the SNMP variable for System Uptime. System Uptime is the recorded time since the device was last restarted.
Match the community name string defined in *Configuring the community name strings*, on page 4, for this device in the statement in the **switch.cfg** file.

```
# Monitor uptime via SNMP

define service{
use                    generic-service; Inherit values from a template
host_name              BIG-IP LTM
service_description    Uptime
check_command          check_snmp!-C private -o sysUpTime.0
}
```

5. Save the **switch.cfg** file and exit the text editor.
6. From the command line of the Nagios server, verify your configuration with the **–v** start option for Nagios. If there are any errors in the files you just edited, a warning message is displayed. Use the following syntax:

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

7. If there are no errors, restart Nagios using the following command:

   ```
   /etc/rc5.d/S99nagios –reload
   ```

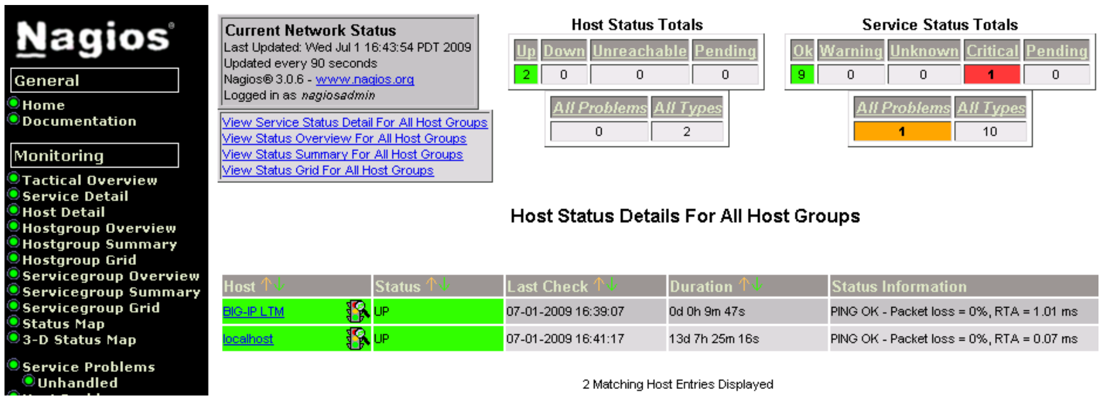
Viewing the Nagios BIG-IP monitor

Now you can log on to the main Nagios web interface where the information on the monitored hosts resides.

To view the Nagios monitor

1. From a web browser that can access the Nagios server, type the default URL for the Nagios device. In our example, we type

   `http://10.10.50.136/Nagios`
2. After logging in, from the left navigation pane, click **Host Detail**. The host you defined appears, showing the average round trip time to the host, as well as packet loss count, if any.



Current Network Status
Last Updated: Wed Jul 1 16:43:54 PDT 2009
Updated every 90 seconds
Nagios® 3.0.6 - www.nagios.org
Logged in as *nagiosadmin*

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems	All Types
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
9	0	0	1	0

All Problems	All Types
1	10

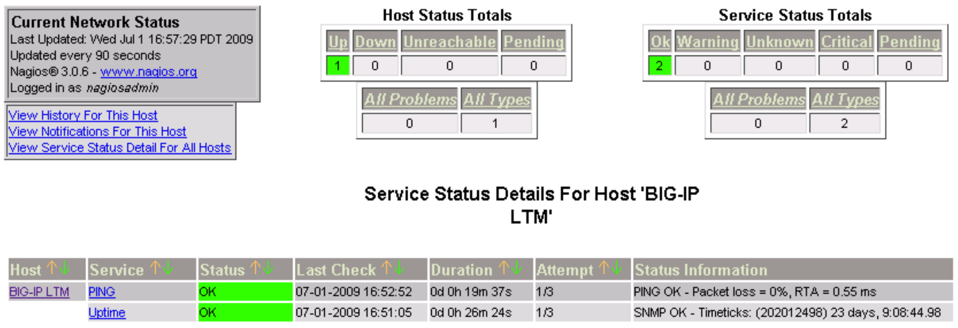
Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
BIG-IP LTM	UP	07-01-2009 16:39:07	0d 0h 9m 47s	PING OK - Packet loss = 0%, RTA = 1.01 ms
localhost	UP	07-01-2009 16:41:17	13d 7h 25m 16s	PING OK - Packet loss = 0%, RTA = 0.07 ms

2 Matching Host Entries Displayed

Figure 4 *Viewing the Host Detail from the Nagios UI*

3. From the Current Network Status box, click **View Host Status Detail for All Host Groups**.
   The SNMP collected value for system up time is shown in the Status Information column (see Figure 5).
   Tip: *If you have multiple hosts, in the Host column, click* ***BIG-IP LTM****, and then click the* ***View Status Detail For This Host*** *link to view details for this host only.*



Current Network Status
Last Updated: Wed Jul 1 16:57:29 PDT 2009
Updated every 90 seconds
Nagios® 3.0.6 - www.nagios.org
Logged in as *nagiosadmin*

View History For This Host
View Notifications For This Host
View Service Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	All Types
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
2	0	0	0	0

All Problems	All Types
0	2

Service Status Details For Host 'BIG-IP LTM'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
BIG-IP LTM	PING	OK	07-01-2009 16:52:52	0d 0h 19m 37s	1/3	PING OK - Packet loss = 0%, RTA = 0.55 ms
	Uptime	OK	07-01-2009 16:51:05	0d 0h 26m 24s	1/3	SNMP OK - Timeticks: (202012498) 23 days, 9:08:44.98

Figure 5 *Viewing the BIG-IP LTM Service Status Details*

The basic Nagios configuration for monitoring BIG-IP LTM is now complete.

Advanced configuration

The following sections describe options for enhanced monitoring of a BIG-IP LTM in a networked environment.

Parent child relationships and services dependencies

Creating Parent Child relationships and establishing Service Dependencies helps eliminate what are commonly called *event storms*. Event storms can occur when a network device, such as a switch, is between a monitoring server and the devices it is monitoring, like a server farm. Another example is if a particular server service, such as a web service, is based on the availability of other services. Typically, if the switch were to experience a failure and the monitoring server could no longer poll the downstream devices, both the switch and servers in the farm would be marked **critical**. The system administrator would receive a "storm" of events notifying him or her that many devices were unreachable with a status of **critical**.

With the proper parent child relationships or service dependencies configuration, devices that would have previously been marked **critical**, like the downstream servers, can now be marked as **unreachable**, **unknown**, or maintain their operational status. Only the networking device itself is now marked **critical**. This difference in status can help shorten troubleshooting cycles during a network event, allowing the administrator to focus on the devices marked **critical** first.

Nagios provides a number of ways to establish relationships between devices or services. These methodologies are discussed in detail in the advanced configurations section of the Nagios administration guide and should be considered for monitoring of Application Delivery Networks to shorten troubleshooting in complex networks.

In the following example, we build a service dependency relationship between the web services available from a virtual server on the BIG-IP LTM and the web services available on the virtual server pool members. We want to configure the Nagios system to alert us if the LTM experiences a failure, but not alert us on the loss of the virtual server web service, because it is dependant of the availability of the pool member web services.

Monitoring the servers in the pool

The first task is to configure Nagios monitors for our web servers. In our example scenario, we have two Microsoft IIS servers providing web services. These servers are in a load balancing pool on the BIG-IP LTM system (for specific instructions on configuring a BIG-IP LTM pool or other BIG-IP objects, see the ***Configuration Guide for BIG-IP Local Traffic Management*** for your LTM version, available on Ask F5.

To configure Nagios monitors

1. On the IIS servers, install the NSClient++ agent software. The NSClient can be found at ***http://sourceforge.net/projects/nscplus/***.

   This agent acts as a communication proxy with the Nagios plug-in for monitoring of windows services.

2. From the Microsoft Management Console for Windows Services, double-click (or right-click and select Properties) NSClient++.

3. Click the Log On tab, and then check the **Allow service to interact with desktop** box. No additional configuration of the agent is needed.

4. Download and install the current, published collection of monitoring plug-ins for Nagios according to the Ubuntu Quick Start Guide instructions (included in the Nagios Version 3.x documentation).

5. Using a text editor, open the nagios.cfg file. In our example, we type: **vi /usr/local/nagios/etc/nagios.cfg**

6. In the nagios.cfg file, find the following line:

   ```
   #cfg_file=/usr/local/nagios/etc/objects/windows.cfg
   ```

   Remove the leading pound (**#**) sign from the start of the line.

7. Save the file.
   The Definitions section of the configuration file should look like the following example:

```
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg

# Definitions for monitoring a router/switch
cfg_file=/usr/local/nagios/etc/objects/switch.cfg
```

Figure 1.1 *Definitions section of the nagios.cfg file*

8. Open the **windows.cfg** file with a text editor. In our example, we type
   vi /usr/local/nagios/etc/objects/windows.cfg

9. Add the Windows servers you want to monitor to the file. In our example we add **web_serv1** and **web_serv2**. Use the following syntax:

```
define host{
        use         windows-server; Inherit default values from a Windows server template
        host_name   web_serv1
        alias       web_serv1
        address     10.10.50.180
        }
```

```
define host{
       use         windows-server; Inherit default values from a Windows server template
       host_name   web_serv2
       alias       web_serv2
       address     10.10.50.179
       }
```

Now that the host definition has been established, the next step is to define the service monitor checks. The service check definitions have all been added, you only need to include the **host_name** you established in step 9.

10. Edit the host name for each check you want to execute. The following example is for the Uptime monitor; the remainder of the service check definitions are included in *Appendix A: Service Monitor Check Definitions*, on page 19 for your reference.

```
define service{
   use generic-service
   host_name web_serv1
   service_description Uptime
   check_command check_nt!UPTIME
   }


define service{
   use generic-service
   host_name web_serv2
   service_description Uptime
   check_command check_nt!UPTIME
   }
```

11. Next, we add the service monitor for the virtual server web service and the host definition for this server using the following syntax:

```
define host{
        use             windows-server  ; Inherit default values from a template
        host_name       web_serv3       ; The name we are giving to this host
        alias           web_serv3       ; A longer name associated with the host
        address         10.10.1.78      ; IP address of the host
        }

define service{
        use                     generic-service
        host_name               web_serv3
        service_description     HTTP
        check_command           check_http
        }
```

12. Add the service dependency, using the following syntax:

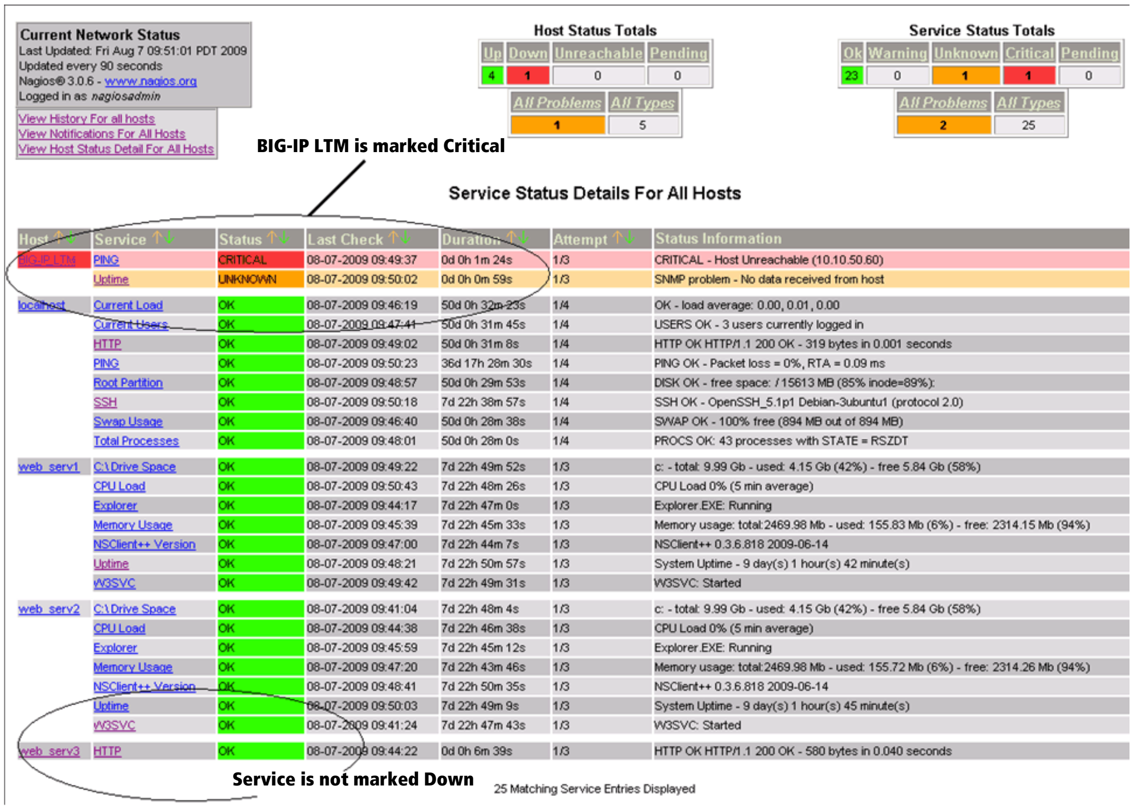
```
define servicedependency{
    dependent_host                  web_serv3
    dependent_service_description   HTTP
    host_name                       web_serv1, web_serv2
    service_description             W3SVC
    execution_failure_criteria      o
    notification_failure_criteria   o
    }
```

13. Here **o** indicates the OK status for the web service. Other options include **w** (warning), **u** (unknown), or **c** (critical).
14. Save the file.
15. Verify that the configuration file is correct using the Nagios Start command with the -v option the following syntax:

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

If there are no errors, restart the nagios service.

Now, if the BIG-IP LTM stops responding, the web service of web_serv3 is not marked **down** if the web services of web_serv1 and web_serv2 are still responding. This allows the administrator to focus on the root problem without having to address additional alerts from the system and maintains the service availability metrics for the application web service. The screenshot on the following page represents this result:



Current Network Status
Last Updated: Fri Aug 7 09:51:01 PDT 2009
Updated every 90 seconds
Nagios® 3.0.6 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
4	1	0	0

All Problems	All Types
1	5

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
23	0	1	1	0

All Problems	All Types
2	25

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
BIG-IP LTM	PING	CRITICAL	08-07-2009 09:49:37	0d 0h 1m 24s	1/3	CRITICAL - Host Unreachable (10.10.50.60)
	Uptime	UNKNOWN	08-07-2009 09:50:02	0d 0h 0m 59s	1/3	SNMP problem - No data received from host
localhost	Current Load	OK	08-07-2009 09:46:19	50d 0h 32m 23s	1/4	OK - load average: 0.00, 0.01, 0.00
	Current Users	OK	08-07-2009 09:47:41	50d 0h 31m 45s	1/4	USERS OK - 3 users currently logged in
	HTTP	OK	08-07-2009 09:49:02	50d 0h 31m 8s	1/4	HTTP OK HTTP/1.1 200 OK - 319 bytes in 0.001 seconds
	PING	OK	08-07-2009 09:50:23	36d 17h 28m 30s	1/4	PING OK - Packet loss = 0%, RTA = 0.09 ms
	Root Partition	OK	08-07-2009 09:48:57	50d 0h 29m 53s	1/4	DISK OK - free space: / 15613 MB (85% inode=89%):
	SSH	OK	08-07-2009 09:50:18	7d 22h 38m 57s	1/4	SSH OK - OpenSSH_5.1p1 Debian-3ubuntu1 (protocol 2.0)
	Swap Usage	OK	08-07-2009 09:46:40	50d 0h 28m 38s	1/4	SWAP OK - 100% free (894 MB out of 894 MB)
	Total Processes	OK	08-07-2009 09:48:01	50d 0h 28m 0s	1/4	PROCS OK: 43 processes with STATE = RSZDT
web_serv1	C:\ Drive Space	OK	08-07-2009 09:49:22	7d 22h 49m 52s	1/3	c: - total: 9.99 Gb - used: 4.15 Gb (42%) - free 5.84 Gb (58%)
	CPU Load	OK	08-07-2009 09:50:43	7d 22h 48m 26s	1/3	CPU Load 0% (5 min average)
	Explorer	OK	08-07-2009 09:44:17	7d 22h 47m 0s	1/3	Explorer.EXE: Running
	Memory Usage	OK	08-07-2009 09:45:39	7d 22h 45m 33s	1/3	Memory usage: total:2469.98 Mb - used: 155.83 Mb (6%) - free: 2314.15 Mb (94%)
	NSClient++ Version	OK	08-07-2009 09:47:00	7d 22h 44m 7s	1/3	NSClient++ 0.3.6.818 2009-06-14
	Uptime	OK	08-07-2009 09:48:21	7d 22h 50m 57s	1/3	System Uptime - 9 day(s) 1 hour(s) 42 minute(s)
	W3SVC	OK	08-07-2009 09:49:42	7d 22h 49m 31s	1/3	W3SVC: Started
web_serv2	C:\ Drive Space	OK	08-07-2009 09:41:04	7d 22h 48m 4s	1/3	c: - total: 9.99 Gb - used: 4.15 Gb (42%) - free 5.84 Gb (58%)
	CPU Load	OK	08-07-2009 09:44:38	7d 22h 46m 38s	1/3	CPU Load 0% (5 min average)
	Explorer	OK	08-07-2009 09:45:59	7d 22h 45m 12s	1/3	Explorer.EXE: Running
	Memory Usage	OK	08-07-2009 09:47:20	7d 22h 43m 46s	1/3	Memory usage: total:2469.98 Mb - used: 155.72 Mb (6%) - free: 2314.26 Mb (94%)
	NSClient++ Version	OK	08-07-2009 09:48:41	7d 22h 50m 35s	1/3	NSClient++ 0.3.6.818 2009-06-14
	Uptime	OK	08-07-2009 09:50:03	7d 22h 49m 9s	1/3	System Uptime - 9 day(s) 1 hour(s) 45 minute(s)
	W3SVC	OK	08-07-2009 09:41:24	7d 22h 47m 43s	1/3	W3SVC: Started
web_serv3	HTTP	OK	08-07-2009 09:44:22	0d 0h 6m 39s	1/3	HTTP OK HTTP/1.1 200 OK - 580 bytes in 0.040 seconds

25 Matching Service Entries Displayed

Figure 2 *Screenshot of Nagios Service Status Details screen*

Notifications

The monitoring scripts available with Nagios provide basic information on monitored state change. For example, the display state of a device changes from **OK** to **Critical** because a ping monitor returns a negative response. Timely notifications of these events are useful for rapid service recovery. By default, Nagios includes methods to send an email message with specific event information when a device exhibits a monitored state change. In this section, we show how to configure this email notification. Our example uses the **mailx** service on the Ubuntu server.

To configure email notifications

1. On the Ubuntu server, execute the following command as root:

   ```
   apt-get install mailx
   ```

2. Open the **commands.cfg** file (found in /usr/local/nagios/etc/objects) to change **/usr/bin/mail** to **/usr/bin/mailx** using the following commands:

```
# 'notify-host-by-email' command definition
define command{
```

```
        command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification
Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTA
DDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/mailx -s
"** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" $CONTACTEMAIL$
        }

# '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/mailx -s "** $NOTIFICATIONTYPE$ Service Al
ert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" $CONTACTEMAIL$
        }
```

3. Save the file.

In all cases, when a notification is generated it must past through a series of filters before it is actually sent. The first such filter is whether notifications are enabled on a system wide basis. The next task is to verify notifications are enabled.

1. Open the file (found in /usr/local/nagios/etc/) in a text editor.
2. In the notifications section, verify that notifications are enabled. The section should look like the following:

```
# NOTIFICATIONS OPTION
# This determines whether or not Nagios will sent out any host or
# service notifications when it is initially (re)started.
# Values: 1 = enable notifications, 0 = disable notifications

enable_notifications=1
```

3. Save the file.

Next, the system verifies the device being monitored is not in a period of scheduled downtime. Scheduled downtime is the status that allows for devices to be offline with no notifications sent should a device demonstrate a state change. Similarly, if the device is in a flapping state, i.e. it is transitioning between states within a specified period, no notifications will be sent.

There are two time filters that also effect whether a state change of a device generates an alert. The time of the state change must be within the time period defined in the notification_period option and be outside the time (in minutes) of the notification_interval setting. If a new notification occurs within the notification_interval no new notifications will be sent. This helps to reduce the number of alert notifications sent related to one service outage.

Once all of the above checks are completed, and the notification passes all the filters the notification passes through a contact filter. Contact filters determine who should receive a notice for a given state change of a particular host and when that notice should be sent.

In our example, we monitor the state change of the W3SVC service on one of the web servers, and add notification related items to the service definition. The service definitions will send an alert email any time during the day to an email group called admins@mycompany.com whenever the W3SVC service enters a warning or critical state.

To monitor the state change of the W3SVC service

1. Open the file (in /usr/local/nagios/etc/objects) in a text editor.
2. Find the following entry:

   and change it to look like the following:

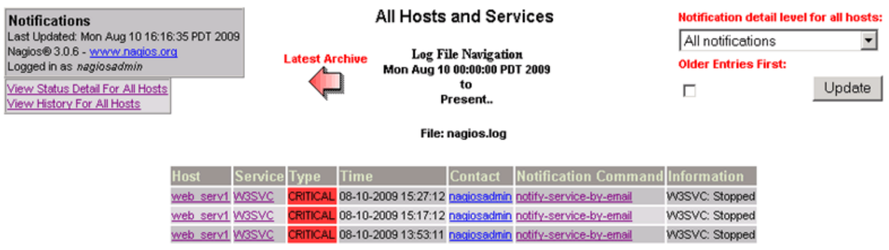
```
check_period             24x7
notification_interval    10
notification_period      24x7
notification_options     w,c
contact_groups           admins
}
```

3. The contact group is configured in the file. In our example, it looks like the following:

```
define contact{
        contact_name          nagiosadmin
        use                   generic-contact
```

```
                alias                   Nagios Admin
                email                   admins@mycompany.com
                }
define contactgroup{
                contactgroup_name       admins
                alias                   Nagios Administrators
                members                 nagiosadmin
                }
```

The result can be traced in the **View Notifications For All Hosts** section of the Nagios web GUI, or the **nagios.log** file in **/usr/local/nagios/var/** directory.



Notifications
Last Updated: Mon Aug 10 16:16:35 PDT 2009
Nagios® 3.0.6 - www.nagios.org
Logged in as *nagiosadmin*

View Status Detail For All Hosts
View History For All Hosts

All Hosts and Services

Latest Archive

Log File Navigation
Mon Aug 10 00:00:00 PDT 2009
to
Present..

File: nagios.log

Notification detail level for all hosts: All notifications
Older Entries First: ☐

Host	Service	Type	Time	Contact	Notification Command	Information
web_serv1	W3SVC	CRITICAL	08-10-2009 15:27:12	nagiosadmin	notify-service-by-email	W3SVC: Stopped
web_serv1	W3SVC	CRITICAL	08-10-2009 15:17:12	nagiosadmin	notify-service-by-email	W3SVC: Stopped
web_serv1	W3SVC	CRITICAL	08-10-2009 13:53:11	nagiosadmin	notify-service-by-email	W3SVC: Stopped

Figure 3 *All Hosts and Services screen showing the W3SVC service checks*

Additional SNMP Monitors

Nagios is primarily a status monitoring system. While SNMP queries can be made for specific hosts (as shown in this guide), storing and displaying this data for historical trending analysis is generally conducted with other tools such as Cacti, or commercial products like Hewlett Packard's OpenView Network Node Manager. Nagios SNMP monitors can collect a value returned from an SNMP GET request and initiate a notification based on a comparison of that value with elements configured in your **.cfg** files. An example of that type of monitor can be found at the following site: ***http://www.monitoringexchange.org/cgi-bin/page.cgi?g=Detailed%2F2447.html;d=1***.

Other Resources

Like many popular open source projects, contributors have provided monitoring scripts that may already match your needs. Some of the resources available for monitoring BIG-IP devices can be found from Nagios monitoring links or through Internet searches. For example, ***http://www.monitoringexchange.org*** has several scripts that monitor the number of servers in a particular pool and can alert if that number changes by a certain percentage.

Conclusion

Open source monitoring tools provide the platform to collect and display availably data and system information based on the administrators' preference. F5 provides the means to identify and collect specific data elements from their devices through their SNMP MIB and compliance to industry standards. Together, long term performance data and information useful for trouble shooting network problems can be made available.

This completes the BIG-IP LTM and Nagios Monitoring Server deployment guide. To provide feedback on this deployment guide or other F5 solution documents, contact us at solutionsfeedback@f5.com.

Appendix A: Service Monitor Check Definitions

The following are service monitor check definitions. These checks are from the Nagios documentation, version 3.x on pages 20 and 21. Please check their website for possible changes, they are listed below for your reference. All of the following continue with the web_serv1 and web_serv2 host names from our examples earlier in this document.

NSClient++ Version
Add the following service definition to monitor the version of the NSClient++ addon that is running on the Windows server. This is useful when it comes time to upgrade your Windows servers to a newer version of the add-on, as you'll be able to tell which Windows machines still need to be upgraded to the latest version of NSClient++.

```
   service_description NSClient++ Version
   check_command check_nt!CLIENTVERSION
   }

define service{
   use generic-service
   host_name web_serv2
   service_description NSClient++ Version
   check_command check_nt!CLIENTVERSION
   }
```

Windows Server uptime
Add the following service definition to monitor the uptime of the Windows server.

CPU utilization
Add the following service definition to monitor the CPU utilization on the Windows server and generate a CRITICAL alert if the 5-minute CPU load is 90% or more or a WARNING alert if the 5-minute load is 80% or greater.

Memory Usage
Add the following service definition to monitor memory usage on the Windows server and generate a CRITICAL alert if memory usage is 90% or more or a WARNING alert if memory usage is 80% or greater.

C:\ drive
Add the following service definition to monitor usage of the C:\ drive on the Windows server and generate a CRITICAL alert if disk usage is 90% or more or a WARNING alert if disk usage is 80% or greater.

```
  check_command check_nt!USEDDISKSPACE!-l c -w 80 -c 90
}
```

```
define service{
  use generic-service
  host_name web_serv2
  service_description C:\ Drive Space
  check_command check_nt!USEDDISKSPACE!-l c -w 80 -c 90
  }
```

W3SVC service state

Add the following service definition to monitor the W3SVC service state on the Windows machine and generate a CRITICAL alert if the service is stopped.

```
define service{
   use generic-service
   host_name web_serv1
   service_description W3SVC
   check_command check_nt!SERVICESTATE!-d SHOWALL -l W3SVC
   }

define service{
   use generic-service
   host_name web_serv2
   service_description W3SVC
   check_command check_nt!SERVICESTATE!-d SHOWALL -l W3SVC
   }
```

Monitor the Explorer.exe process

Add the following service definition to monitor the Explorer.exe process on the Windows machine and generate a CRITICAL alert if the process is not running.

```
define service{
   use generic-service
   host_name web_serv1
   service_description Explorer
   check_command check_nt!PROCSTATE!-d SHOWALL -l Explorer.exe
   }

define service{
   use generic-service
   host_name web_serv2
   service_description Explorer
   check_command check_nt!PROCSTATE!-d SHOWALL -l Explorer.exe
   }
```