Important: This guide has been archived. While the content in this guide is still valid for the products and versions listed in the document, it is no longer being updated and may refer to F5 or third party products or versions that have reached end-of-life or end-of-support. For a list of current guides, see https://f5.com/solutions/deployment-guides.

DEPLOYMENT GUIDE

DEPLOYING THE BIG-IP LTM SYSTEM WITH CITRIX PRESENTATION SERVER 3.0 AND 4.5

Deploying F5 BIG-IP Local Traffic Manager with Citrix Presentation Server

Welcome to the F5 BIG-IP Deployment Guide for Citrix® Presentation Server. This guide contains step-by-step procedures for configuring the BIG-IP Local Traffic Manager (LTM) for directing traffic, ensuring application availability, improving performance and providing a flexible layer of security for Citrix Presentation Server version 3.0 and 4.5.

Citrix Presentation Server provides a run-time environment for applications to be hosted on the server and accessed over the network or by using web protocols, with just keyboard strokes, mouse movements and screen updates being exchanged between the client and the server. The BIG-IP LTM provides mission critical availability, enhanced security, simple scalability and high operational resiliency to the Citrix Presentation Server deployment so that users can access resources from any device in any location as easily and securely as from within the corporate LAN.

In a Citrix Presentation Server environment, the BIG-IP LTM provides intelligent traffic management and high-availability by monitoring and managing connections to the Citrix Web Interface and the Citrix XML Broker components. In addition, the built-in performance optimization capabilities of the LTM provide faster operations to facilitate a better end-user experience. The LTM also keeps persistence records for certain connections to always be directed to the same server for a specified period of time, to ensure that the workflow in the Presentation Server environment is fully preserved.

For an overview on Citrix Presentation Server environments, see *Appendix C: Overview of Citrix Presentation Server environment*, on page 42.

For more information on Citrix Presentation Server, see **www.citrix.com/English/ps2/products/product.asp?contentID=186**

For more information on the F5 BIG-IP LTM, see **www.f5.com/products/big-ip/product-modules/local-traffic-manager.html**

Prerequisites and configuration notes

The following are prerequisites for this solution:

- The BIG-IP LTM system must be running version 9.1 or later. We strongly recommend version 9.4 or later.
- The Citrix Presentation Server must be running version 3.0 or 4.5. The BIG-IP LTM health monitors in this guide are dependent on the version of Citrix Presentation Server you are running. See the appropriate Appendix when configuring the health monitors.
- Briefly review the basic configuration tasks and the few pieces of information, such as IP addresses, that you should gather in preparation for completing this configuration.

Configuration example

This configuration example describes the typical configuration of the BIG-IP LTM system to monitor and manage the critical components of a Citrix Presentation Server environment, namely the Web Interface servers and the XML Broker servers.

In this implementation, traffic to the Citrix Web Interface servers and the Citrix XML Broker servers is managed by the F5 BIG-IP LTM system, and when necessary, ensures that each client connects to the same member of the farm across multiple sessions using persistence on the BIG-IP LTM. The F5 BIG-IP LTM system is also setup to monitor the Citrix Web Interface servers and Citrix XML Broker servers to ensure availability and automatically mark down servers that are not operating correctly. The ability to terminate SSL sessions in order to offload this processing from the Presentation Servers is also available with a simple addition of the Client SSL profile to the web interface virtual server referred to in this guide.

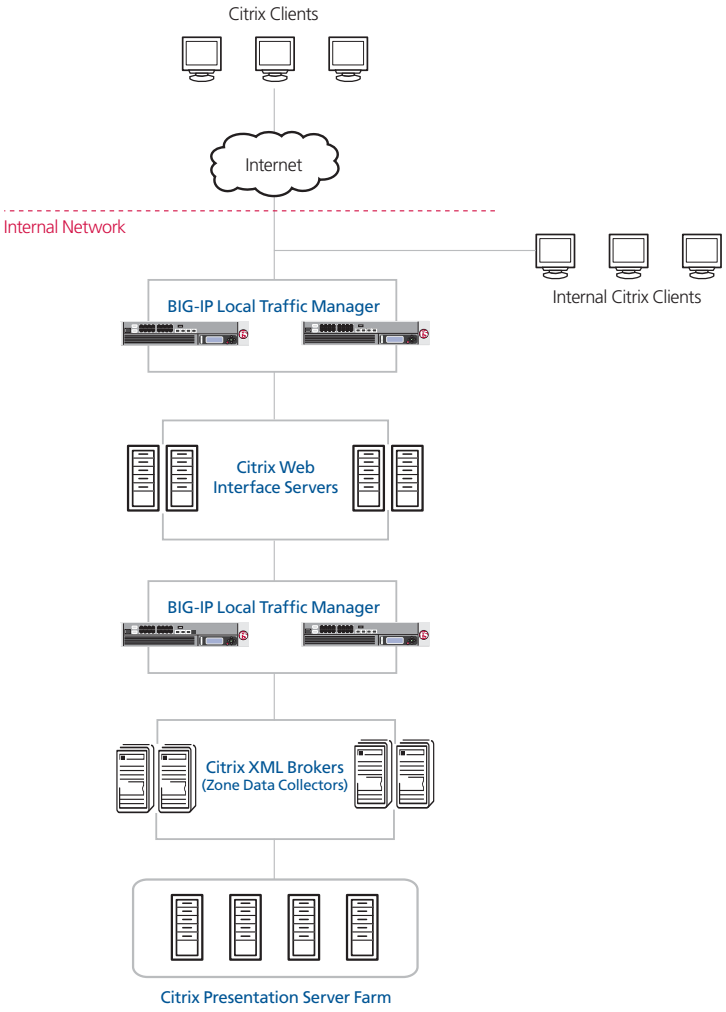

Figure 1 *Logical configuration example*

The Citrix Web Interface is the gateway to the applications hosted by the Citrix Presentation Server farm. All initial user requests for application access are first sent to the Web Interface. This can result in a large number of connections to a single Web Interface server if the connections are not properly load balanced. While the resource utilization of the Web Interface servers is low given the tasks they perform, it is important that users are being directed to the server that can generate and deliver the application list to users as fast as possible. The BIG-IP LTM system's load balancing algorithms easily prevent over-burdening Web Interface servers, intelligently distributing traffic based on a number of different factors.

If a Citrix Web Interface server goes down or is otherwise unavailable, the BIG-IP LTM system's advanced health monitors immediately detect the failed server and transparently direct the requests to available servers, so user requests are not affected. The BIG-IP LTM also ensures optimized delivery of the HTML pages containing the application list or ICA files by utilizing the F5's TCP/IP optimization technology (TCP Express).

The Citrix Presentation Servers designated as XML Brokers perform the bulk of the processing necessary for a user to launch an application. It is important that the XML Service requests from the Web Interface be directed to the XML Broker server that can authenticate and generate the application list quickly. It is also important to quickly detect the failure of a XML Broker server so that user requests for application access are not affected. The BIG-IP LTM prevents over-burdening of the XML Broker servers by utilizing the built-in load balancing algorithms to direct traffic appropriately.

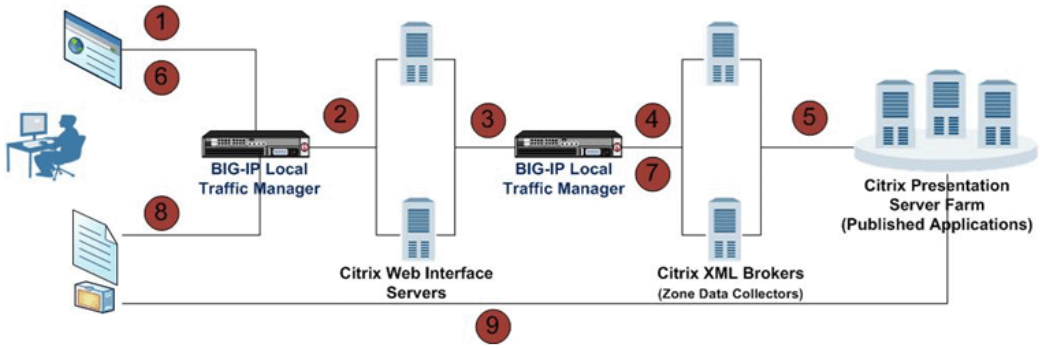

Figure 2 *Data flow in a BIG-IP LTM - Citrix Presentation Server environment*

1. The user initiates request for applications by sending credentials to the Web Interface Virtual Server on the LTM, using a web browser.
2. The BIG-IP LTM determines an appropriate Web Interface server and sends the HTTP request to it.
3. Upon receiving the user's request, the Web Interface server sends an XML request to the XML Broker Virtual Server on the LTM for a list of applications the user has access to with the user's credentials.

4. The BIG-IP LTM determines an appropriate XML Broker server and sends the XML request to it.
5. The XML Broker server authenticates the user and retrieves a list of applications that the user can access from the Presentation Server Farm. The XML Broker sends an XML response with information about the list of applications that the user can access, to the Web Interface server that requested it. Upon receiving the XML response, the Web Interface server publishes a HTML page with the application list to the user.
6. When the user clicks on an application icon on the HTML page, the BIG-IP LTM receives this request and sends it to the original Web Interface server that published the list, based on persistence set on the Virtual Server. The Web Interface server receives this request and sends an XML query, for the address of a target Presentation Server for the application, to the XML Broker Virtual Server on the BIG-IP LTM.
7. The BIG-IP LTM sends the XML request to the XML Broker server. The XML Broker identifies the address of the least busy Presentation Server hosting the application and relays it back to the Web Interface server that requested it using the XML Service.
8. Upon receiving the response, the Web Interface generates a customized ICA file for the application and delivers it to the user.
9. The ICA file is executed by the client and the user initiates an ICA connection with the target Presentation Server.

◆ Tip

Although two pairs of BIG-IP devices are depicted in this configuration, only one pair of BIG-IP devices are necessary to achieve the functionality in this deployment. The second pair is depicted here to provide clarity on the logical placement of the BIG-IP LTMs in a Citrix Presentation Server environment.

Configuring the BIG-IP LTM system for deployment with Citrix Presentation Server

To configure the BIG-IP LTM system for integration with Citrix Presentation Server, you must complete the following procedures:

◆ Tip

We recommend you save your existing BIG-IP configuration before you begin the procedures in this Deployment Guide. To save your BIG-IP LTM configuration, see ***Appendix D: Backing up and restoring the BIG-IP LTM system****, on page 45*

The BIG-IP LTM system offers both web-based and command line configuration tools, so that users can work in the environment that they are most comfortable with. This Deployment Guide contains procedures to configure the BIG-IP LTM system using the BIG-IP web-based Configuration utility only. If you are familiar with using the bigpipe command line interface you can use the command line to configure the BIG-IP device; however, we recommend using the Configuration utility.

Connecting to the BIG-IP device

Use the following procedure to access the BIG-IP web-based Configuration utility using a web browser.

To connect to the BIG-IP system using the Configuration utility

1. In a browser, type the following URL:
   https://<administrative IP address of the BIG-IP device>
   A Security Alert dialog box appears, click **Yes**.
   The authorization dialog box appears.
2. Type your user name and password, and click **OK**.
   The Welcome screen opens.

Once you are logged onto the BIG-IP system, the Welcome screen of the new Configuration utility opens. From the Configuration utility, you can configure and monitor the BIG-IP system, as well as access online help, download SNMP MIBs and Plug-ins, and even search for specific objects.

Creating the health monitors

To ensure that traffic is directed to those servers that are responding to requests, it is important to configure health monitors on the BIG-IP LTM to verify the availability of the servers being load balanced. In this configuration, health monitors are setup for the Citrix Web Interface servers and the Citrix XML Brokers. For this deployment, we use one of F5's advanced health monitors that attempts to retrieve explicit content from the nodes. The health monitors check the nodes (IP address and port they are listening on), and based on whether correct behavior is noticed from the nodes being monitored, mark them up for the LTM to forward traffic, or mark them down so that no new requests are sent to them.

The way the BIG-IP LTM health monitors are configured depends on which version of Citrix Presentation Server you are running.

Creating the Web Interface health monitor

The first monitor we create is for the Citrix Web Interface devices. Use the following procedure. Additional information about this health monitor can be found in the Appendices depending on what Presentation Server version you are using. See *Appendix A: Configuring the BIG-IP LTM health monitors for Presentation Server 3.0*, on page 25 or *Appendix B: Configuring the BIG-IP LTM health monitors for Presentation Server 4.5*, on page 31 for more information.

To configure the Citrix Web Interface health monitor

1. On the Main tab, expand **Local Traffic**, and then click **Monitors**. The Monitors screen opens.
2. Click the **Create** button. The New Monitor screen opens.
3. In the **Name** box, type a name for the Monitor. In our example, we type **CitrixWeb**.
4. From the **Type** list, select **HTTP**.
5. In the Configuration section, in the **Interval** and **Timeout** boxes, type an appropriate Interval and Timeout. We recommend using an Interval of **4** and a Timeout of **9**.
   See the Appendix appropriate for your Presentation Server version for more information on setting these values.

6. In the **Send String** box, type a Send String specific to the application being checked. This is dependent on your version of Presentation Server:

   For Presentation Server 4.5, we type:

   ```
   GET /Citrix/AccessPlatform
   ```

   For Presentation Server 3.0, we leave the Send String at the default:

   ```
   GET /
   ```

7. In the **Receive String** box, type a Receive String specific to the application being checked. This is dependent on your version of Presentation Server:

   For Presentation Server 4.5, we type:

   ```
   Citrix
   ```

   For Presentation Server 3.0, we type:

   ```
   WebInterface.htm
   ```

   Important: See *Appendix A: Configuring the BIG-IP LTM health monitors for Presentation Server 3.0*, on page 25 or *Appendix B: Configuring the BIG-IP LTM health monitors for Presentation Server 4.5*, on page 31 for more information on these strings.

8. Click the **Finished** button.

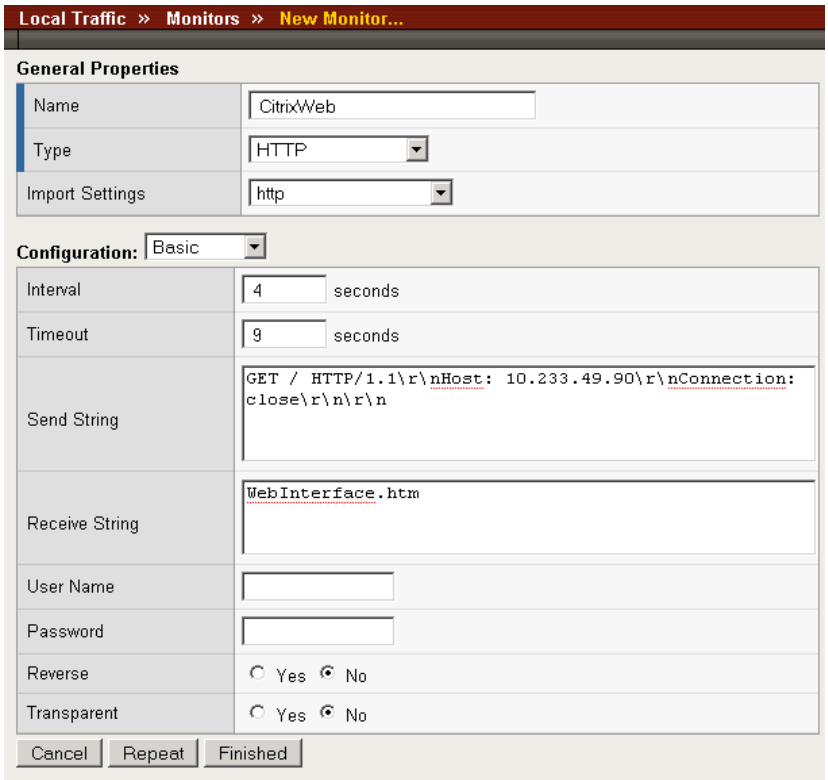

Figure 3 *Creating the Citrix Web Interface health monitor*

Creating the Citrix WebLogon health monitor

The next step is to create a second health monitor for the Web Interface devices. Additional information on this monitor can be found in *Defining the Citrix Web Logon health monitor*, on page 26.

To configure the Citrix Web Logon health monitor

1. On the Main tab, expand **Local Traffic**, and then click **Monitors**. The Monitors screen opens.
2. Click the **Create** button. The New Monitor screen opens.
3. In the **Name** box, type a name for the Monitor. In our example, we type **CitrixWebLogon**.
4. From the **Type** list, select **HTTP**. The HTTP Monitor configuration options appear.
5. In the Configuration section, in the **Interval** and **Timeout** boxes, type an appropriate Interval and Timeout.
   Because of the extended time it takes to complete the login, compared to just receiving basic web page text as the previous monitor does, we recommend a longer time period for the interval and the timeout, still maintaining the same ratio (1:2 +1).

   For this application we recommend using an interval of **15** and a timeout of **31**.
6. In the **Send String** box, type a Send String specific to the application being checked.
7. In the **Receive String** box, type a Receive String specific to the application being checked.

   Important: *The Send and Receive strings depend on the version of Presentation Server you are using. See* ***Defining the Citrix Web Logon health monitor***, on page 26 *for* ***Presentation Server 3.0*** *or* ***Defining the Citrix Web Logon health monitor***, on page 32 *for* ***Presentation Server 4.5*** *for more information on these strings.*
8. Click the **Finished** button (see Figure 4).

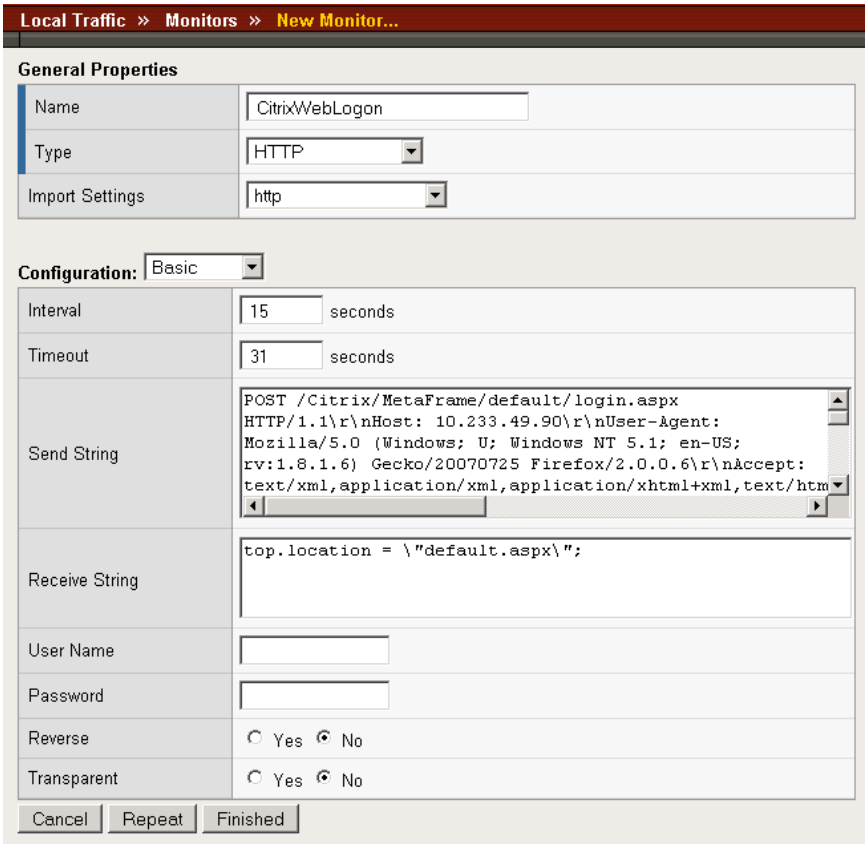

Figure 4 *Creating the Citrix WebLogon monitor for Presentation Server 3.0*

Note

For a complete explanation of the health monitor used in this section, see ***Defining the Citrix Web Logon health monitor****, on page 26 for* ***Presentation Server 3.0*** *or* ***Defining the Citrix Web Logon health monitor****, on page 32.*

We recommend initially testing new monitors with at least a 1:3 +1 ratio between the Interval and the Timeout values. After the monitor functionality is verified, the values can be adjusted to obtain the best balance between rapidly determining service failure, and sufficient time to allow the monitor to complete to avoid falsely marking nodes down and adversely affecting the application being monitored. In this configuration, we use a 1:2 +1 ratio to more quickly recognize a server failure.

Creating the Citrix XML Broker health monitor

The final monitor we create in this configuration is for the Citrix XML Broker devices. This monitor is also dependent on the version of Presentation Server you are using. For more information on this monitor, see:

- Presentation Server 3.0:
  Defining the Citrix XML Broker health monitor, on page 28
- Presentation Server 4.5:
  Defining the Citrix XML Broker health monitor, on page 33.

To configure the Citrix XML Broker health monitor from the Configuration utility

1. On the Main tab, expand **Local Traffic**, and then click **Monitors**. The Monitors screen opens.
2. Click the **Create** button. The New Monitor screen opens.
3. In the **Name** box, type a name for the Monitor. In our example, we type **InternalTicketTag**.
4. From the **Type** list, select **HTTP**. The HTTP Monitor configuration options appear.
5. In the Configuration section, in the **Interval** and **Timeout** boxes, type an appropriate Interval and Timeout. For this monitor, we recommend using the default interval of **5** and the default timeout of **16**.
6. In the **Send String** box, type a Send String specific to the application being checked.
7. In the **Receive String** box, type a Receive String specific to the application being checked.

   Important: *The Send and Receive strings depend on the version of Presentation Server you are using. See* ***Defining the Citrix XML Broker health monitor***, on page 28 *for* ***Presentation Server 3.0*** *or* ***Defining the Citrix XML Broker health monitor***, on page 33 *for* ***Presentation Server 4.5*** *for more information on these strings.*
8. Click the **Finished** button (see Figure 5).

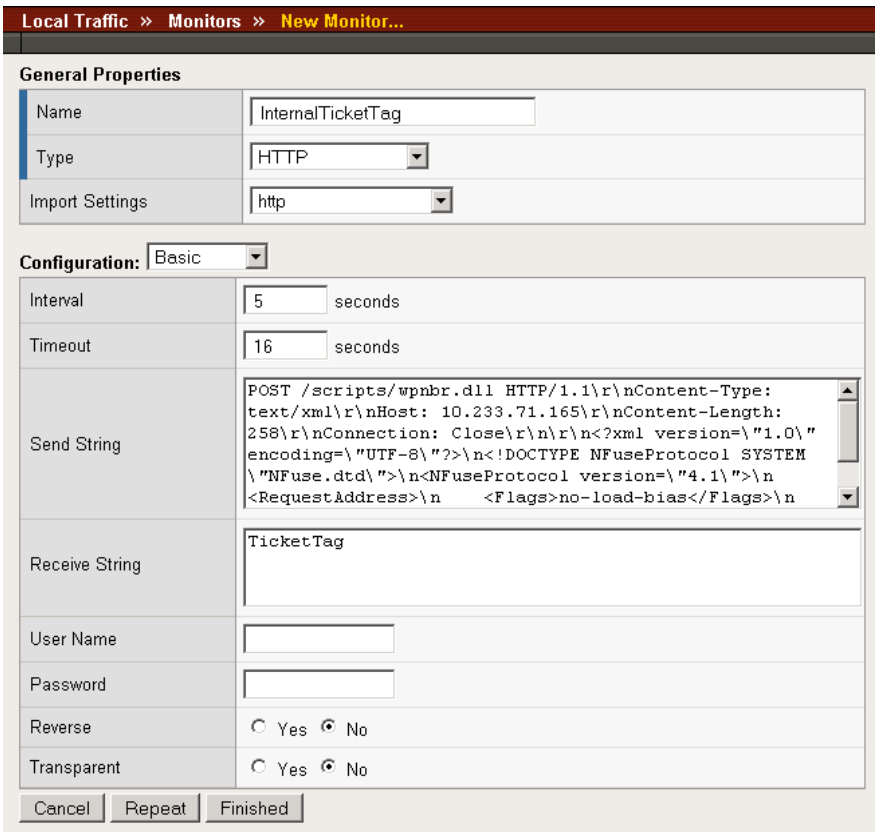

Figure 5 *Configuring the Citrix XML Monitor for Presentation Server 3.0*

Note

The last line of the Send String includes a variable ***<AppName>*** *which requires the name of an actual documentation title that appears on the second screen of the application list. This example uses* ***notepad*** *for testing, but a production application should be substituted and byte size specified earlier in the send string might need to be altered as well. See the Appendix appropriate for your Presentation Server version for additional details.*

Creating the pools

The next step is to create a pool on the BIG-IP LTM system for the Citrix Web Interface servers and Citrix XML Broker servers. A BIG-IP LTM pool is a set of devices grouped together to receive traffic according to a load balancing method. In this configuration, we create a pool for each of the major two services in the Presentation Server environment being load balanced: the Citrix Web Interface and the Citrix XML Broker.

Creating the Citrix Web Interface pool

The first pool we create is the Citrix Web Interface pool.

To create the Citrix Web Interface server pool

1. On the Main tab, expand **Local Traffic**, and then click **Pools**. The Pool screen opens.
2. In the upper right portion of the screen, click the **Create** button. The New Pool screen opens.
3. In the **Name** box, enter a name for your pool. In our example, we use **web_pool**.
4. In the Health Monitors section, select the name of the monitor you created in *Creating the Web Interface health monitor*, on page 6, and click the Add (**<<**) button. In our example, we select **CitrixWeb** and click the Add (<<) button, and then select the name of the monitor you created in *Creating the Citrix WebLogon health monitor*, on page 8. In our example, we select **CitrixWebLogon** and click the Add (<<) button again. Both monitors should now be in the **Active** list.
5. From the Load Balancing Method list, choose your preferred load balancing method (different load balancing methods may yield optimal results for a particular network). In our example, we select **Least Connections (member)**.
6. For this pool, we leave the Priority Group Activation **Disabled**.
7. In the New Members section, make sure the **New Address** option button is selected.
8. In the **Address** box, type the address of the first server. In our example, we type **192.168.10.1**.
9. In the **Service Port** box, type the service number you want to use for this device, or specify a service by choosing a service name from the list. In our example, we type **80**.
10. Click the **Add** button to add the member to the list.
11. Repeat steps 8-10 for each server you want to add to the pool. In our example, we repeat these steps for the remaining server in this pool, **192.168.10.2**.
12. Click the **Finished** button (see Figure 6).

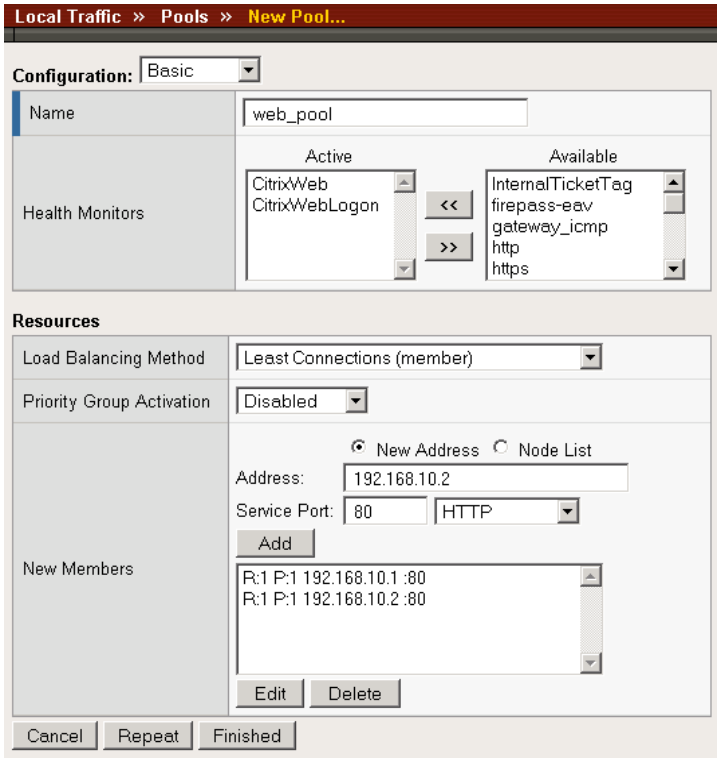

Figure 6 *Creating the Citrix Web Interface pool*

Creating the Citrix XML Broker pool

Next we create a pool for the XML Broker devices.

To create the Citrix XML Broker server pool

1. On the Main tab, expand **Local Traffic**, and then click **Pools**. The Pool screen opens.
2. In the upper right portion of the screen, click the **Create** button. The New Pool screen opens.
3. In the **Name** box, enter a name for your pool. In our example, we use **xml_pool**.
4. In the Health Monitors section, select the name of the monitor you created in *Creating the Citrix XML Broker health monitor*, on page 10, and click the Add (**<<**) button. In our example, we select **InternalTicketTag**.
5. From the Load Balancing Method list, choose your preferred load balancing method (different load balancing methods may yield optimal results for a particular network). In our example, we select **Least Connections (member)**.
6. For this pool, we leave the Priority Group Activation **Disabled**.

7. In the New Members section, make sure the **New Address** option button is selected.
8. In the **Address** box, type the address of the first XML Broker device. In our example, we type **192.168.20.1**.
9. In the **Service Port** box, type the service number you want to use for this device, or specify a service by choosing a service name from the list. In our example, we type **80**.
10. Click the **Add** button to add the member to the list.
11. Repeat steps 8-10 for each server you want to add to the pool. In our example, we repeat these steps three times for the remaining servers in this pool, **192.168.20.2-.4**.
12. Click the **Finished** button.

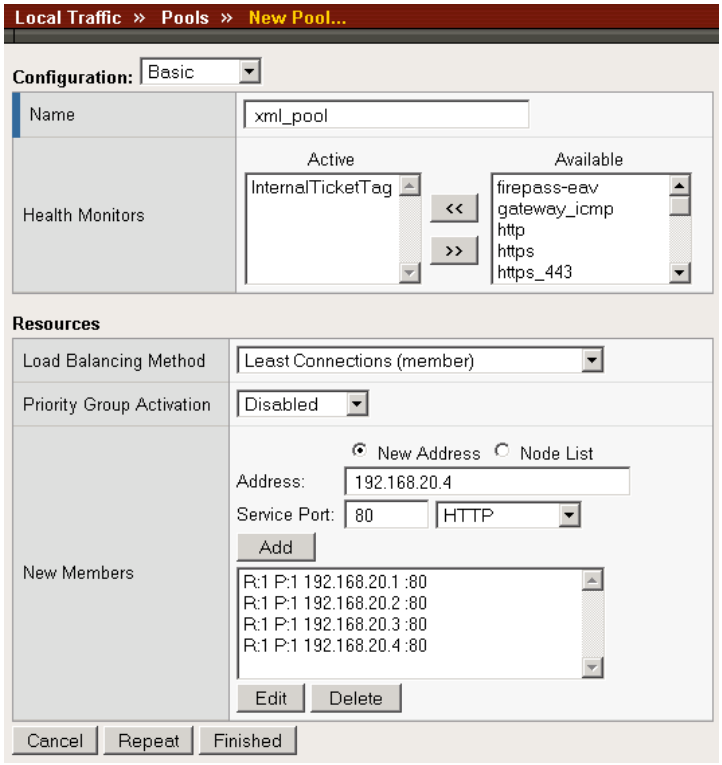

Figure 7 *Creating the Citrix XML Interface Pool*

Creating Profiles

BIG-IP version 9.0 and later uses profiles for greater control over managing network traffic while making network traffic management easy and efficient. A profile is an object that contains user-configurable settings, with default values, for controlling the behavior of a particular type of network traffic, such as HTTP connections.

Although it is possible to use the default profiles, we strongly recommend that you create new profiles based on the default parent profiles. Creating new profiles allows you to easily modify the profile settings specific to your deployment, and ensures that you do not accidentally overwrite the default profile. We also recommend however, that you use the default settings in the cookie persistence profile for this configuration (Insert method, Session based).

These profiles use new optimized profiles available in BIG-IP LTM version 9.4 and later. If you are using a BIG-IP LTM version prior to 9.4, the ***Configuration Guide for BIG-IP Local Traffic Management*** for version 9.4 (available on **AskF5**) shows the differences between the base profiles and the optimized profile types. Use this guide to manually configure the optimization settings.

Creating an HTTP profile

The first new profile we create is an HTTP profile. The HTTP profile contains numerous configuration options for how the BIG-IP LTM system handles HTTP traffic. Our testing with Citrix Presentation Server shows that while we see a great deal of benefit from compression, caching only produces a minimal improvement. Therefore, we recommend using the **http-wan-optimized-compression** parent profile. This profile uses specific compression (among other) settings to optimize traffic over the WAN.

If you are not using version 9.4, or do not have compression or caching licensed, you can choose the default HTTP parent profile, or one of the other optimized HTTP parent profiles.

Citrix Presentation Server must have access to the IP address of the connecting clients in order to be fully functional. Some of the BIG-IP LTM features used in this Deployment Guide obscure this information. To overcome this, we use the following HTTP profile to insert an **X-Forwarded-For** header into the HTTP header. This supplies the IP address of the client so it is available to Citrix Presentation Server.

To create a new HTTP profile

1. On the Main tab, expand **Local Traffic**, and then click **Profiles**. The HTTP Profiles screen opens.
2. In the upper right portion of the screen, click the **Create** button.
3. In the **Name** box, type a name for this profile. In our example, we type **citrix-http-opt**.
4. From the **Parent Profile** list, select **http-wan-optimized-compression**.
5. From the Insert **XForward For** row, click the Custom box, and then select **Enabled** from the list.
6. Modify any of the other settings as applicable for your network. In our example, we leave the settings at their default levels.
7. Click the **Finished** button.

Creating the TCP profiles

The next profiles we create are the TCP profiles. If most of the Citrix Presentation Server users are accessing the devices via a Local Area Network, we recommend using the **tcp-lan-optimized** (for server-side TCP connections) parent profile. If the majority of the users are accessing the system from remote or home offices, we recommend using an additional TCP profile, called **tcp-wan-optimized** (for client side TCP connections). The use of these optimized profiles is optional, you can alternatively use the base TCP parent profile if appropriate for your configuration.
In our example, we leave these profiles at their default levels; you can configure any of the options as applicable for your network.

Creating the LAN optimized TCP profile

First we configure the LAN optimized profile. If you are not using version 9.4 or do not want to use this optimized profile, you can choose the default TCP parent profile.

To create a new TCP profile

1. On the Main tab, expand **Local Traffic**, and then click **Profiles**. The HTTP Profiles screen opens.
2. On the Menu bar, from the **Protocol** menu, click **tcp**.
3. In the upper right portion of the screen, click the **Create** button. The New TCP Profile screen opens.
4. In the **Name** box, type a name for this profile. In our example, we type **citrix-tcp-lan**.
5. From the **Parent Profile** list, select **tcp-lan-optimized** if you are using BIG-IP LTM version 9.4 or later; otherwise select **tcp**.
6. Modify any of the settings as applicable for your network. In our example, we leave the settings at their default levels.
7. Click the **Finished** button.

Creating the WAN optimized TCP profile

Now we configure the WAN optimized profile. Remember, if most of the users are accessing the system over the LAN or other low latency links, you do not need to create this profile.

To create a new TCP WAN optimized profile

1. On the Main tab, expand **Local Traffic**, and then click **Profiles**. The HTTP Profiles screen opens.
2. On the Menu bar, from the **Protocol** menu, click **tcp**.
3. In the upper right portion of the screen, click the **Create** button. The New TCP Profile screen opens.

4. In the **Name** box, type a name for this profile. In our example, we type **citrix-tcp-wan**.
5. From the **Parent Profile** list, select **tcp-wan-optimized**.
6. Modify any of the settings as applicable for your network. In our example, we leave the settings at their default levels.
7. Click the **Finished** button.

Creating the persistence profile

The next profile we create is a Persistence profile. We recommend using persistence for Citrix devices, although the type of persistence depends on your configuration. In our example, use cookie persistence (HTTP cookie insert).

To create a new cookie persistence profile based on the default profile

1. On the Main tab, expand **Local Traffic**, and then click **Profiles**.
   The HTTP Profiles screen opens.
2. On the Menu bar, click **Persistence**.
   The Persistence Profiles screen opens.
3. In the upper right portion of the screen, click the **Create** button.
   The New Persistence Profile screen opens.
4. In the **Name** box, type a name for this profile. In our example, we type **citrix-cookie**.
5. From the **Persistence Type** list, select **Cookie**.
   The configuration options for cookie persistence appear.
6. Modify any of the settings as applicable for your network. In our example, we leave the settings at their default levels.
7. Click the **Finished** button.

Creating the OneConnect profile

The final profile we create is a OneConnect™ profile. OneConnect allows the BIG-IP LTM to keep HTTP connections alive and reuse them, thereby reducing the overhead of creating and destroying the connections. This obscures the clients IP address, and is the primary reason why we configured the HTTP profile to insert the X-Forwarded-For header.

To create a new OneConnect profile

1. On the Main tab, expand **Local Traffic**, and then click **Profiles**.
   The HTTP Profiles screen opens.
2. On the Menu bar, from the **Other** menu, click **OneConnect**.

3. In the upper right portion of the screen, click the **Create** button. The New HTTP Profile screen opens.
4. In the **Name** box, type a name for this profile. In our example, we type **citrix-oneconnect**.
5. From the **Parent Profile** list, ensure that **oneconnect** is selected.
6. Modify any of the other settings as applicable for your network. In our example, we leave the settings at their default levels.
7. Click the **Finished** button.

Creating the virtual servers

A virtual server with its virtual IP address is the visible, routable entity through which the servers in a load balancing pool are made available to the client (the IP address to give clients or add to DNS).

The next step in the configuration is to configure a virtual server that references the pools and profiles created in the preceding sections.

Creating the Citrix Web Interface virtual server

The first virtual server we create is for the Citrix Web Interface servers.

To create the Citrix Web Interface virtual server

1. On the Main tab, expand **Local Traffic**, and then click **Virtual Servers**. The Virtual Servers screen opens.
2. In the upper right portion of the screen, click the **Create** button. The New Virtual Server screen opens.
3. In the **Name** box, type a name for this virtual server. In our example, we type **citrix_ps.company.com**.
4. In the **Destination** section, select the **Host** option button.
5. In the **Address** box, type the IP address of this virtual server. In our example, we use **172.27.92.118**.

6. In the **Service Port** box, type **80**, or select **HTTP** from the list.

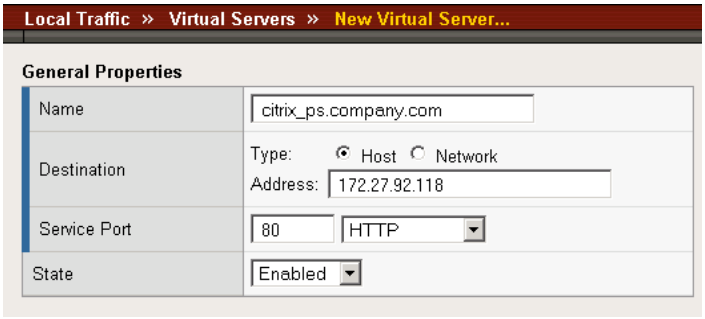

Figure 8 *Creating the Citrix virtual server*

7. From the Configuration list, select **Advanced**.
   The Advanced configuration options appear.
8. Leave the **Type** list at the default setting: **Standard**.
9. From the **Protocol Profile (Client)** list select the name of the profile you created in the *Creating the WAN optimized TCP profile* section. If you did not create a WAN optimized profile, select the LAN optimized profile as in the following Step. In our example, we select **citrix-tcp-wan**.
10. From the **Protocol Profile (Server)** list, select the name of the profile you created in the *Creating the LAN optimized TCP profile* section. In our example, we select **citrix-tcp-lan**.
11. From the **OneConnect Profile** list, select the name of the profile you created in the *Creating the OneConnect profile* section. In our example, we select **citrix-oneconnect**.
12. From the HTTP Profile list, select the name of the profile you created in the *Creating an HTTP profile* section. In our example, we select **citrix-http-opt** (see Figure 9.

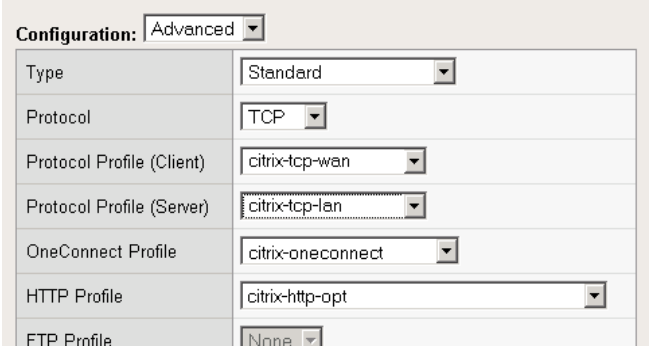

Figure 9 *Selecting the Citrix profiles for the virtual server*

13. In the Resources section, from the **Default Pool** list, select the pool you created in *Creating the Citrix Web Interface pool*, on page 12. In our example, we select **web_pool**.
14. From the **Default Persistence Profile** list, select the persistence profile you created in the *Creating the persistence profile* section. In our example, we select **citrix-cookie**.
15. Click the **Finished** button (see Figure 10).

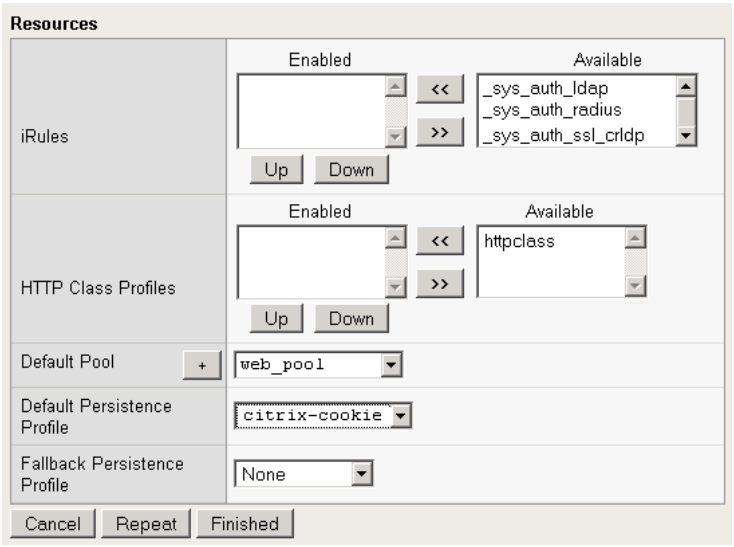

***Figure 10** Adding the Pool and Persistence profile to the virtual server*

Creating the Citrix XML Broker virtual server

Next we create a virtual server for the Citrix XML Broker servers.

To create the Citrix XML Broker virtual server

1. On the Main tab, expand **Local Traffic**, and then click **Virtual Servers**. The Virtual Servers screen opens.
2. In the upper right portion of the screen, click the **Create** button. The New Virtual Server screen opens.
3. In the **Name** box, type a name for this virtual server. In our example, we type **citrix.xml.site.com**.
4. In the **Destination** section, select the **Host** option button.
5. In the **Address** box, type the IP address of this virtual server. In our example, we use **192.168.20.100**.
6. In the **Service Port** box, type **80**, or select **HTTP** from the list.

7. From the Configuration list, select **Advanced**.
   The Advanced configuration options appear.
8. Leave the **Type** list at the default setting: **Standard**.
9. From the **Protocol Profile (Client)** list select the name of the profile you created in the *Creating the WAN optimized TCP profile* section. If you did not create a WAN optimized profile, select the LAN optimized profile as in the following Step. In our example, we select **citrix-tcp-wan**.
10. From the **Protocol Profile (Server)** list, select the name of the profile you created in the *Creating the LAN optimized TCP profile* section. In our example, we select **citrix-tcp-lan**.
11. From the **OneConnect Profile** list, select the name of the profile you created in the *Creating the OneConnect profile* section. In our example, we select **citrix-oneconnect**.
12. From the HTTP Profile list, select the name of the profile you created in the *Creating an HTTP profile* section. In our example, we select **citrix-http-opt**.
13. In the Resources section, from the **Default Pool** list, select the pool you created in *Creating the Citrix Web Interface pool*, on page 12. In our example, we select **xml_pool**.
14. From the **Default Persistence Profile** list, select the persistence profile you created in the *Creating the persistence profile* section. In our example, we select **citrix-cookie**.
15. Click the **Finished** button.

Creating a default SNAT

A secure network address translation (SNAT) ensures the proper routing of connections between the Web Interface servers to the XML Broker servers. For the configuration described in this deployment guide, we configure a default SNAT. While not every network topology requires a default SNAT, if the Web Interface and XML Broker are on the same subnet, a default SNAT is mandatory.

For more information on SNATs, see the BIG-IP LTM documentation.

◆ Note

If you do not want source address translation on client connections from the external VLAN, you can disable the default SNAT for the external VLAN.

To create a default SNAT

1. On the Main tab, expand **Local Traffic**, and then click **SNATs**.
   The SNATs screen opens.
2. In the upper right portion of the screen, click the **Create** button.
   The New SNAT screen opens.

3. In the **Name** box, type a name for this SNAT. In our example, we type **DefaultSNAT**.
4. In the **Translation** list, select **Automap**.
5. ***Optional:*** If you to disable (or enable) the default SNAT for specific VLANs, from the VLAN Traffic list, select either **Enabled on** or **Disabled on** from the list. From the Available list, select the appropriate VLAN and click the Add (**<<**) button to move it to the Selected list.
6. Click the **Finished** button.

Synchronizing the BIG-IP configuration if using a redundant system

If you are using a redundant BIG-IP configuration, the final step is to synchronize the configuration to the peer BIG-IP device.

To synchronize the configuration using the Configuration utility

1. On the Main tab, expand **System**.
2. Click **High Availability**.
   The Redundancy screen opens.
3. On the Menu bar, click **ConfigSync**.
4. Click the **Self --> Peer** button.
   The configuration synchronizes with its peer.

Configuring the Citrix Presentation Server Environment for the F5 BIG-IP LTM System

The Citrix Presentation Server environment needs to be reconfigured for integration with the F5 BIG-IP LTM system. The Citrix Web Interface is the only component within the Citrix Presentation Server environment that needs to be reconfigured for this deployment.

Configuring the Citrix Web Interface

The Web Interface servers must be reconfigured to point to the XML Broker Virtual Server address on the BIG-IP LTM. To complete the reconfiguration, you use the following procedure.

To reconfigure the Citrix Web Interface

1. Open the Citrix Access Management Console from a Web Interface server.
2. Select the Web Interface site: **Citrix Resources - Configuration Tools - Web Interface - http://**
3. From the middle column, select **Manage Server Farms**.
4. Click the appropriate server farm, and select **Edit**.
5. Select the existing entries, which should be pointing directly to the original XML Broker server addresses, and click **Remove**.
6. Click **Add**.
7. Type the XML Broker's Virtual Server address that you configured in *Creating the Citrix XML Broker virtual server*, on page 20. In our example, we type **192.168.20.100**.
8. Click **OK**.

Configuring Citrix to Retrieve the Correct Client IP address

Citrix Presentation Server needs to be configured to look for the client IP address in the X-Forwarded-For HTTP header. Otherwise, every connection will appear to be coming from the BIG-IP LTM and not from its actual location. This can only be done by editing two Java files on each of the Web Interface servers. To complete the reconfiguration, you use the following procedure.

To reconfigure the Citrix to Read X-Forwarded-For headers for the Client IP address

1. Open the files **Inetpub\wwwroot\Citrix\AccessPlatform\app_data\auth\serverscripts\include.aspxf** and **Inetpub\wwwroot\Citrix\AccessPlatform\app_data\site\serverscripts\include.aspxf** on the Web Interface server, and find the function named **getClientAddress**.

   In version 4.5, it looks like this:

```
public string getClientAddress() {
    if ( Session[SV_AGE_CLIENT_IP] != null ) {
        return (string) Session[SV_AGE_CLIENT_IP];
    } else {
        return Request.UserHostAddress;
    }
}
```

2. Edit these functions so that they look like the following:

```
public string getClientAddress() {
    if ( Session[SV_AGE_CLIENT_IP] != null ) {
        return (string) Session[SV_AGE_CLIENT_IP];
    } else if
(Request.ServerVariables["HTTP_X_FORWARDED_FOR"]  != null
) {
        return (string)
Request.ServerVariables["HTTP_X_FORWARDED_FOR"];
    } else {
        return Request.UserHostAddress;
    }
}
```

3. Repeat this for each Web Interface server.

Important

Remember to restart each Web Interface server for the changes to take effect.

Appendix A: Configuring the BIG-IP LTM health monitors for Presentation Server 3.0

This Appendix contains a detailed explanation of the BIG-IP LTM health monitors used for Citrix Presentation Server 3.0.

Defining the Citrix Web Interface health monitors

In the configuration example used in this guide, two custom HTTP Extended Content Verification (ECV) monitors were used to monitor the health of the Citrix Web Interface - CitrixWeb and Citrix WebLogon.

A HTTP ECV monitor is used to check the status of Hypertext Transfer Protocol (HTTP) traffic. The HTTP ECV monitor attempts to open a connection to the server on port 80, and uses the HTTP protocol to send and receive content. The check is successful when the content matches the Receive String value.

Defining the Web Interface health monitor

When a user tries to access the Web Interface server from a web browser, the first page received is usually the Web Interface welcome page, which includes the login form. The Web Interface prevents access to any of the site's other main scripts directly prior to authentication and includes a built-in URL filter that directs the user to the login page on such access.

If the web server hosting the Web Interface server is operational and if the Web Interface server is running, a simple request to the IP address of the server should result in a response with a valid welcome page consisting of the login fields. By default, the first page at which all users start **/default/default.aspx** for Web Interface version 3.x. This is also the page that the users are redirected to if authentication fails.

For this monitor, we send a simple GET request to the Web Interface server (**GET /**) for the default page of the site and expect to see the string **WebInterface.htm** embedded in the URL or content. The check passes if the response from the Web Interface server has the valid string. This helps us determine that both the web server and the Web Interface server are responding and operational.

Figure 11contains the Send String in our example:

```
GET /
```

Figure 11 *Example Send String for the Web Interface monitor*

Defining the Citrix Web Logon health monitor

The Web Interface server first authenticates a user before displaying the list of published applications that the user can access. The Citrix Web Interface can be configured for explicit authentication, single sign-on, smart card authentication or anonymous authentication. For the purposes of this deployment guide and the configuration example, we are assuming that the Citrix Web Interface is configured for Explicit Authentication.

When a user accesses the Web Interface server from a web browser, the Web Interface generates a web login form requiring valid User Name, Password and Domain values. When the user clicks on the Log In button, the credentials entered in the login form are sent to the Web Interface server using HTTP or HTTPS (depending on the Web Interface server configuration). The Web Interface server translates the user credentials received from the HTTP POST request into XML, and forwards them on to the XML Broker as part of the application retrieval request. The XML Broker validates the credentials and responds to the Web Interface with the list of applications that the user has access to. A sample user login request to a Web Interface server is shown below.

```
POST /Citrix/MetaFrame/default/login.aspx HTTP/1.1

Content-Type: application/x-www-form-urlencoded

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6

state=LOGIN&LoginType=Explicit&user=user1&password=password1&do
main=domain1&login=Log+In
```

If the Web Interface server is operational and accepting requests, a successful logon event to the Web Interface will result in a response with a list of applications available to the user and content such as the application icons and application launch files. By default, the Web Interface will direct the authenticated user to **/site/default.aspx** immediately after a successful logon attempt. Moreover, the web page received by an authenticated user response includes the contents of the file **/site/footer.txt** within the page.

For this monitor, we send a POST request to the Web Interface server with the credentials of a valid user in the Citrix Presentation Server environment. The POST request sent to the Web Interface server closely mimics a real web browser request to ensure that it is accepted by the server and to ensure the receipt of a valid response that indicates a functional server. A valid response if the user authentication is successful would include the string **top.location = "default.aspx"** embedded in the content. The check passes if the response from the Web Interface server has the valid string. This helps us determine that the Web Interface server is responding, fully functional and accepting requests.

Figure 12 contains the complete Send String in our example.

```
POST /Citrix/MetaFrame/default/login.aspx HTTP/1.1\r\nHost:
10.233.49.90\r\nUser-Agent: Mozilla/5.0 (Windows; U; Windows
NT 5.1; en-US; rv:1.8.1.6) Gecko/20070725
Firefox/2.0.0.6\r\nAccept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.
9,text/plain;q=0.8,image/png,*/*;q=0.5\r\nAccept-Language:
en-us,en;q=0.5\r\nAccept-Charset:
ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\nConnection:
close\r\nReferer:
http://10.233.49.90/Citrix/MetaFrame/default/login.aspx\r\nCo
okie: icaClientAvailable=false; icaIsPassThrough=0;
icaScreenResolution=1440x900;
NFuseLogin=NFuse_ClientName=WIx005fSIEx005fXqCO4rYChx002dvT&N
Fuse_LogonMode=Ex0078plicit;
NFuseMode=NFuse_WindowType=seamless&NFuse_CurrentFolder=;
ASP.NET_SessionId=zhg241450ijgney2oifhnw45;
NFuseLogin=NFuse_ClientName=WIx005fSIEx005fXqCO4rYChx002dvT&N
Fuse_LogonMode=Ex0078plicit;
NFuseMode=NFuse_WindowType=seamless&NFuse_CurrentFolder=\r\nC
ontent-Type:
application/x-www-form-urlencoded\r\nContent-Length:
84\r\n\r\nstate=LOGIN&LoginType=Explicit&user=lbtest1&passwor
d=password&domain=PD&login=Log+In\n
```

Figure 12 *Send String for the Web Logon health monitor*

Important

Variables in red text must be replaced with a valid IP address, user name, password or domain. The IP Address should be set to the DNS name or the IP address of the Virtual Server.

In addition, the Content Length value must be replaced with the new total length of the string beginning with ***state=LOGIN*** *and ending with* ***login=Log+In****. This value changes depending on the length of the actual user name, password and domain values used. A count of the characters, minus a count of any escape (\) characters, is all that is required to determine the Content Length.*

Note

Text of the Send String entry used in this section can be cut and paste into the ***Send String*** *field in the BIG-IP LTM GUI health monitor page to speed the development of monitors for the Citrix Presentation Server application (v3.0 and v4.5). Text of the Receive String entry in the configuration section can be cut & paste into the* ***Receive String*** *field in the BIG-IP LTM GUI on the health monitor page, when using the corresponding Send String. The script text here can be used as a guideline for how a Send String or Receive String should be built.*

All script text is subject to change depending on many factors including application version, application developer, changes in content, application alterations, system or network design changes, and other factors.

Defining the Citrix XML Broker health monitor

This section contains a detailed explanation of the Citrix XML Broker health monitor used in this guide. In the configuration example used in this guide, a custom HTTP Extended Content Verification (ECV) monitor was used to monitor the health of the XML Brokers and the Citrix Presentation Server farm - InternalTicketTag.

When an authenticated user clicks on an application icon from the web page displaying the list of applications the user can access, a request to launch the application is sent to the Web Interface server. The Web Interface server queries the XML Broker for the address of the least busy Presentation Server that hosts the application that the user is requesting. Once the target server is identified and checked to ensure that it is responding, the XML Broker responds to the Web Interface server with the IP Address of the server. A proper response from the XML Broker results in the Web Interface server responding to the user with a **launch.ica** file for the requested application. The **launch.ica** file is executed and the user initiates an ICA connection to access the application on the target Presentation Server.

When the user clicks an application icon, the Web Interface server sends a request, similar to the following example, to the XML Broker server. In this example, a request for the application Notepad is generated.

```
POST /scripts/wpnbr.dll HTTP/1.1
Content-Type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.1">
    <RequestAddress>
        <Flags>no-load-bias</Flags>
        <Name>
            <AppName>notepad</AppName>
        </Name>
    </RequestAddress>
</NFuseProtocol>
```

Once the target server has been identified, the XML Broker responds to the Web Interface request with the IP address, with a response similar to the following example:

```
HTTP/1.1 200 OK
Content-type: text/xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.1">
    <ResponseAddress>
        <ServerAddress addresstype="dot">5.214.10.251</ServerAddress>
        <ServerType>win32</ServerType>
        <ConnectionType>tcp</ConnectionType>
```

```
        <ClientType>ica30</ClientType>
        <TicketTag>IMAHostId:13093</TicketTag>
        <FarmLoadHint>0</FarmLoadHint>
    </ResponseAddress>
</NFuseProtocol>
```

When the IP Address of the least busy target server for a published application cannot be determined, the Web Interface server requests results in a response with errors and does not include the **<TicketTag>** entries. This error condition can occur for any number of reasons, such as if the XML Service on the XML Broker is not functioning correctly, if there are problems with the IMA service on the target server or the XML Broker, if logons are disable on the target server, or if the Local Host Cache on the XML Broker is invalid, and so on. A sample response with the error condition where the XML Broker could not find a result to the Web Interface query is shown in the following example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.1">
    <ResponseAddress>
        <ErrorId>unspecified</ErrorId>
        <MPSError type="IMA">0x800000024</MPSError>
        <BrowserError>0x00000024</BrowserError>
    </ResponseAddress>
</NFuseProtocol>
```

If the XML Broker server is functioning correctly and the Presentation Server farm is operational and accepting requests for applications hosted on the farm, and if there is at least one Presentation Server available with the requested application, a successful query from the Web Interface server will result in a response with **<TicketTag>** entry embedded in the content.

For this monitor, the BIG-IP LTM sends a POST request to the XML Broker server for the IP Address of a target server hosting the Notepad application. The POST request sent to the XML Broker server closely mimics a real XML request that a Web Interface server would send to ensure that it is accepted by the XML Broker server and for the receipt of a valid response. If the request is successful, the XML response sent by the XML Broker server would include the string **TicketTag** embedded in content. The check passes if the response from the XML Broker server has the valid string. This helps determine that the XML Broker server is operational and that the Presentation Server farm is fully functional and accepting application requests.

The following is the code for the Send String in our monitor:

```
POST /scripts/wpnbr.dll HTTP/1.1\r\nContent-Type:
text/xml\r\nHost: 10.233.71.165\r\nContent-Length:
251\r\nConnection: Close\r\n\r\n<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\n<!DOCTYPE NFuseProtocol SYSTEM
\"NFuse.dtd\">\n<NFuseProtocol version=\"4.1\">\n
<RequestAddress>\n    <Flags>no-load-bias</Flags>\n    <Name>
<AppName>notepad</AppName> </Name>
</RequestAddress>\n</NFuseProtocol>\n
```

Figure 13 *Send string for the XML Broker monitor*

Important

Variables in red text must be replaced with a valid IP address and an application name that is actually hosted on the Presentation Server farm. The IP Address should be set to the DNS name or the IP address of the Virtual Server.

In addition, the Content Length value must be replaced with the new total length of the string beginning with ***<?xml version=\"1.0\"*** *and ending with* ***</NFuseProtocol>****. This value changes depending on the length of the actual user name, password and domain values used. A count of the characters, minus a count of any escape (\) characters, is all that is required to determine the Content Length.*

Note

Text of the Send String used in this section can be cut & paste into the ***Send String*** *field in the BIG-IP LTM GUI health monitor page to speed the development of monitors for the Citrix Presentation Server application (v3.0 and v4.5). Text of the Receive String from the configuration section, can be cut and paste into the* ***Receive String*** *field in the BIG-IP LTM GUI on the health monitor page, when using the corresponding Send String. The script text here can be used as a guideline for how a Send String or Receive String should be built.*

All script text is subject to change depending on many factors including application version, application developer, changes in content, application alterations, system or network design changes, and other factors.

Appendix B: Configuring the BIG-IP LTM health monitors for Presentation Server 4.5

This Appendix contains a detailed explanation of the BIG-IP LTM health monitors used for Citrix Presentation Server 4.5.

Defining the Citrix Web Interface health monitors

In the configuration example used in this guide, two custom HTTP Extended Content Verification (ECV) monitors were used to monitor the health of the Citrix Web Interface - CitrixWeb and Citrix WebLogon.

A HTTP ECV monitor is used to check the status of Hypertext Transfer Protocol (HTTP) traffic. The HTTP ECV monitor attempts to open a connection to the server on port 80, and uses the HTTP protocol to send and receive content. The check is successful when the content matches the Receive String value.

Defining the Web Interface health monitor

When a user tries to access the Web Interface server from a web browser, the first page received is usually the Web Interface welcome page, which includes the login form. The Web Interface prevents access to any of the site's other main scripts directly prior to authentication and includes a built-in URL filter that directs the user to the login page on such access.

If the web server hosting the Web Interface server is operational and if the Web Interface server is running, a simple request to the IP address of the server should result in a response with a valid welcome page consisting of the login fields. By default, the first page at which all users start is **/auth/login.aspx** for Web Interface version 4.x. This is also the page that the users are redirected to if authentication fails.

For this monitor, we send a simple GET request to the Web Interface server (**GET / Citrix/AccessPlatform**) for the default page of the site and expect to see the string **Citrix** embedded in the content. The check passes if the response from the Web Interface server returns the valid string. This helps us determine that both the web server and the Web Interface server are responding and operational.

The following is the Send String in our example:

```
GET /Citrix/AccessPlatform
```

Figure 14 *Example Send String for the Web Interface monitor*

Defining the Citrix Web Logon health monitor

The Web Interface server first authenticates a user before displaying the list of published applications that the user can access. The Citrix Web Interface can be configured for explicit authentication, single sign-on, smart card authentication or anonymous authentication. For the purposes of this deployment guide and the configuration example, we are assuming that the Citrix Web Interface is configured for Explicit Authentication.

When a user accesses the Web Interface server from a web browser, the Web Interface generates a web login form requiring valid User Name, Password and Domain values. When the user clicks on the Log In button, the credentials entered in the login form are sent to the Web Interface server using HTTP or HTTPS (depending on the Web Interface server configuration). The Web Interface server translates the user credentials received from the HTTP POST request into XML, and forwards them on to the XML Broker as part of the application retrieval request. The XML Broker validates the credentials and responds to the Web Interface with the list of applications that the user has access to. A sample user login request to a Web Interface server is shown below.

```
POST /Citrix/MetaFrame/default/login.aspx HTTP/1.1

Content-Type: application/x-www-form-urlencoded

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6

state=LOGIN&LoginType=Explicit&user=user1&password=password1&do
main=domain1&login=Log+In
```

If the Web Interface server is operational and accepting requests, a successful logon event to the Web Interface will result in a response with a list of applications available to the user and content such as the application icons and application launch files. By default, the Web Interface will direct the authenticated user to **/site/default.aspx** immediately after a successful logon attempt. Moreover, the web page received by an authenticated user response includes the contents of the file **/site/footer.txt** within the page.

For this monitor, we send a POST request to the Web Interface server with the credentials of a valid user in the Citrix Presentation Server environment. The POST request sent to the Web Interface server closely mimics a real web browser request to ensure that it is accepted by the server and to ensure the receipt of a valid response that indicates a functional server. A valid response if the user authentication is successful would include the string **/Citrix/AccessPlatform/site/default.aspx** embedded in the content. The check passes if the response from the Web Interface server has the valid string. This helps us determine that the Web Interface server is responding, fully functional and accepting requests.

The following is the complete Send String in our example:

```
POST /Citrix/AccessPlatform/auth/login.aspx
HTTP/1.1\r\nReferer:
http://10.133.40.50/Citrix/AccessPlatform/auth/login.aspx\r\n
Content-Type:
application/x-www-form-urlencoded\r\nContent-Length:
136\r\nConnection:
Close\r\n\r\nLoginType=Explicit&user=u1&password=Password&dom
ain=CITRIX&submitMode=submit&slLanguage=en&ReconnectAtLoginOp
tion=DisconnectedAndActive\n
```

Figure 15 *Example Send String for the Web Logon monitor*

Important

Variables in red text must be replaced with a valid IP address, user name, password or domain. The IP Address should be set to the DNS name or the IP address of the Virtual Server.

In addition, the Content Length value must be replaced with the new total length of the string beginning with ***LoginType=Explicit*** *and ending with* ***DisconnectedAndActive****. This value changes depending on the length of the actual user name, password and domain values used. A count of the characters, minus a count of any escape (\) characters, is all that is required to determine the Content Length.*

Note

Text of the Send String entry used in this section can be cut and paste into the ***Send String*** *field in the BIG-IP LTM GUI health monitor page to speed the development of monitors for the Citrix Presentation Server application (v3.0 and v4.5). Text of the Receive String entry in the configuration section can be cut & paste into the* ***Receive String*** *field in the BIG-IP LTM GUI on the health monitor page, when using the corresponding Send String. The script text here can be used as a guideline for how a Send String or Receive String should be built.*

All script text is subject to change depending on many factors including application version, application developer, changes in content, application alterations, system or network design changes, and other factors.

Defining the Citrix XML Broker health monitor

This Appendix contains a detailed explanation of the Citrix XML Broker health monitor used in this guide. In the configuration example used in this guide, a custom HTTP Extended Content Verification (ECV) monitor was used to monitor the health of the XML Brokers and the Citrix Presentation Server farm - InternalTicketTag.

When an authenticated user clicks on an application icon from the web page displaying the list of applications the user can access, a request to launch the application is sent to the Web Interface server. The Web Interface server queries the XML Broker for the address of the least busy Presentation Server that hosts the application that the user is requesting. Once the target server is identified and checked to ensure that it is responding, the XML Broker responds to the Web Interface server with the IP Address of the server. A proper response from the XML Broker results in the Web Interface server responding to the user with a **launch.ica** file for the requested application. The **launch.ica** file is executed and the user initiates an ICA connection to access the application on the target Presentation Server.

When the user clicks an application icon, the Web Interface server sends a request, similar to the following example, to the XML Broker server. In this example, a request for the application Notepad is generated.

```
POST /scripts/wpnbr.dll HTTP/1.1
Content-Type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.1">
    <RequestAddress>
        <Flags>no-load-bias</Flags>
        <Name>
            <AppName>notepad</AppName>
        </Name>
    </RequestAddress>
</NFuseProtocol>
```

Once the target server has been identified, the XML Broker responds to the Web Interface request with the IP address, with a response similar to the following example:

```
HTTP/1.1 200 OK
Content-type: text/xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.1">
    <ResponseAddress>
        <ServerAddress addresstype="dot">5.214.10.251</ServerAddress>
        <ServerType>win32</ServerType>
        <ConnectionType>tcp</ConnectionType>
        <ClientType>ica30</ClientType>
        <TicketTag>IMAHostId:13093</TicketTag>
        <FarmLoadHint>0</FarmLoadHint>
    </ResponseAddress>
</NFuseProtocol>
```

When the IP Address of the least busy target server for a published application cannot be determined, the Web Interface server requests results in a response with errors and does not include the **<TicketTag>** entries. This error condition can occur for any number of reasons, such as if the XML Service on the XML Broker is not functioning correctly, if there are problems with the IMA service on the target server or the XML Broker, if logons are disable on the target server, or if the Local Host Cache on the XML Broker is invalid, and so on. A sample response with the error condition where the XML Broker could not find a result to the Web Interface query is shown in the following example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.1">
    <ResponseAddress>
        <ErrorId>unspecified</ErrorId>
        <MPSError type="IMA">0x800000024</MPSError>
        <BrowserError>0x00000024</BrowserError>
    </ResponseAddress>
</NFuseProtocol>
```

If the XML Broker server is functioning correctly and the Presentation Server farm is operational and accepting requests for applications hosted on the farm, and if there is at least one Presentation Server available with the requested application, a successful query from the Web Interface server will result in a response with **<TicketTag>** entry embedded in the content.

For this monitor, the BIG-IP LTM sends a POST request to the XML Broker server for the IP Address of a target server hosting the Notepad application. The POST request sent to the XML Broker server closely mimics a real XML request that a Web Interface server would send to ensure that it is accepted by the XML Broker server and for the receipt of a valid response. If the request is successful, the XML response sent by the XML Broker server would include the string **TicketTag** embedded in content. The check passes if the response from the XML Broker server has the valid string. This helps determine that the XML Broker server is operational and that the Presentation Server farm is fully functional and accepting application requests.

The following is the code for the Send String in our monitor:

```
POST /scripts/wpnbr.dll HTTP/1.1\r\nContent-Type:
text/xml\r\nHost: 10.133.40.100\r\nContent-Length:
251\r\nConnection: Close\r\n\r\n<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\n<!DOCTYPE NFuseProtocol SYSTEM
\"NFuse.dtd\">\n<NFuseProtocol version=\"4.1\">\n
<RequestAddress>\n    <Flags>no-load-bias</Flags>\n    <Name>
<AppName>Notepad</AppName> </Name>
</RequestAddress>\n</NFuseProtocol>\n
```

Figure 16 *Example Send String for the XML Broker monitor*

Important

Variables in red text must be replaced with a valid IP address and an application name that is actually hosted on the Presentation Server farm. The IP Address should be set to the DNS name or the IP address of the Virtual Server.

In addition, the Content Length value must be replaced with the new total length of the string beginning with ***<?xml version=\"1.0\"*** *and ending with* ***</NFuseProtocol>****. This value changes depending on the length of the actual user name, password and domain values used. A count of the characters, minus a count of any escape (\) characters, is all that is required to determine the Content Length.*

Note

Text of the Send String used in this section can be cut & paste into the ***Send String*** *field in the BIG-IP LTM GUI health monitor page to speed the development of monitors for the Citrix Presentation Server application (v3.0 and v4.5). Text of the Receive String from the configuration section, can be cut & paste into the* ***Receive String*** *field in the BIG-IP LTM GUI on the health monitor page, when using the corresponding Send String. The script text here can be used as a guideline for how a Send String or Receive String should be built.*

All script text is subject to change depending on many factors including application version, application developer, changes in content, application alterations, system or network design changes, and other factors.

Configuring alternate Send and Receive strings

Another good candidate for a health monitor Send string is one that performs application enumeration. This is the method by which the Web Interface asks the XML Broker to enumerate which applications are available. The resultant information is used to populate the list of applications that the user may select.

```
POST /scripts/wpnbr.dll
HTTP/1.1
Content-Type: text/xml
Host: 10.133.40.100
Content-Length: 310
Connection: Close
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<!DOCTYPE NFuseProtocol SYSTEM \"NFuse.dtd\">
<NFuseProtocol version=\"4.6\">
    <RequestAppData>
        <DesiredDetails>defaults</DesiredDetails>
        <ServerType>all</ServerType>
        <ClientType>ica30</ClientType>
        <ClientType>content</ClientType>
    </RequestAppData>
</NFuseProtocol>
```

The XML Broker responds back with the following:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NFuseProtocol SYSTEM "NFuse.dtd">
<NFuseProtocol version="4.6">
    <ResponseAppData>
      <AppDataSet>
        <Scope traverse="onelevel"/>
        <AppData>
          <InName>Notepad</InName>
          <FName>Notepad</FName>
          <Details>
            <Settings appisdisabled="false"
appisdesktop="false">
              <Folder/>
              <Description>notepad</Description>
              <WinWidth>1024</WinWidth>
              <WinHeight>768</WinHeight>
              <WinColor>8</WinColor>
              <WinType>pixels</WinType>
              <WinScale>1</WinScale>
              <SoundType minimum="false">basic</SoundType>
              <VideoType minimum="false">none</VideoType>
              <Encryption minimum="false">basic</Encryption>
              <AppOnDesktop value="false"/>
              <AppInStartmenu value="false"/>
              <PublisherName>Farm2</PublisherName>
```

```
                <SSLEnabled>false</SSLEnabled>
                <RemoteAccessEnabled>false</RemoteAccessEnabled>
              </Settings>
            </Details>
            <SeqNo>1206119789</SeqNo>
            <ServerType>win32</ServerType>
            <ClientType>ica30</ClientType>
          </AppData>
          <AppData>
            <InName>Wireshark</InName>
            <FName>Wireshark</FName>
            <Details>
              <Settings appisdisabled="false"
appisdesktop="false">
                <Folder/>
                <Description/>
                <WinWidth>1024</WinWidth>
                <WinHeight>768</WinHeight>
                <WinColor>8</WinColor>
                <WinType>pixels</WinType>
                <WinScale>1</WinScale>
                <SoundType minimum="false">basic</SoundType>
                <VideoType minimum="false">none</VideoType>
                <Encryption minimum="false">basic</Encryption>
                <AppOnDesktop value="false"/>
                <AppInStartmenu value="false"/>
                <PublisherName>Farm2</PublisherName>
                <SSLEnabled>false</SSLEnabled>
                <RemoteAccessEnabled>false</RemoteAccessEnabled>
              </Settings>
            </Details>
            <SeqNo>1206048557</SeqNo>
            <ServerType>win32</ServerType>
            <ClientType>ica30</ClientType>
          </AppData>
        </AppDataSet>
      </ResponseAppData>
</NFuseProtocol>
```

If the XML Broker is healthy, there should be an **<AppData>** section for each application available. As a result, it is easy to ensure that it is working by setting the Receive string to look for the name of one of your available applications. In our case, we chose **Notepad**.

The following is the code for the Send String in our monitor:

```
POST /scripts/wpnbr.dll HTTP/1.1\r\nContent-Type:
text/xml\r\nHost: 10.133.40.100\r\nContent-Length:
310\r\nConnection: Close\r\n\r\n<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\n<!DOCTYPE NFuseProtocol SYSTEM
\"NFuse.dtd\">\n<NFuseProtocol version=\"4.6\">\n
<RequestAppData>\n
<DesiredDetails>defaults</DesiredDetails>\n
<ServerType>all</ServerType>\n
<ClientType>ica30</ClientType>\n
<ClientType>content</ClientType>\n </RequestAppData>\n
</NFuseProtocol>\n
```

Send string for the alternate XML Broker monitor

In our example, the Receive string is simply **Notepad**.

Important

Variables in red text must be replaced with a valid IP address and an application name that is actually hosted on the Presentation Server farm. The IP Address can be set to any Host name or IP Address that the XML Broker server will accept. The DNS name for the Web Interface Virtual Server is commonly used here. This may require unique monitors for each node if there is not a suitable universal value for the pool. In addition, the Content Length value must be replaced with the new total length of the string beginning with ***<?xml version=\"1.0\"*** *and ending with* ***</NFuseProtocol>****. This value changes depending on the length of the actual user name, password and domain values used. A count of the characters, minus a count of any escape (\) characters, is all that is required to determine the Content Length.*

Note

Text of the Send String used in this section can be cut & paste into the ***Send String*** *field in the BIG-IP LTM GUI health monitor page to speed the development of monitors for the Citrix Presentation Server application (v3.0 and v4.5). Text of the* ***Receive String*** *from the configuration section, can be cut and paste into the Receive String field in the BIG-IP LTM GUI on the health monitor page, when using the corresponding Send String. The script text here can be used as a guideline for how a Send String or Receive String should be built.*

All script text is subject to change depending on many factors including application version, application developer, changes in content, application alterations, system or network design changes, and other factors.

Optional advanced XML Broker health monitors

The following two optional health monitors check the XML Broker's ability to validate credentials. This is already partially checked by the web interface login monitor. However, if a login problem is with a particular XML Broker and not with the web interface server, the BIG-IP LTM could mark the web interface server down but leave the broken XML Broker up. These two monitors mark down the XML Broker instead of the Web Interface.

Both of the following monitors are reverse monitors. A reverse monitor is one in which the receive string is something we expect not to see in a normal case. The servers will be marked down if this string is seen.

Review the following optional monitors and, if applicable, choose the one best suited for your configuration.

XML Broker Credential Validation Health Monitor 1

The first health monitor sends a **RequestValidateCredentials** request with dummy user credentials to the XML Broker. We expect to see a credentials-failed error message. However, should the XML Broker lose its connection to its domain controller and thus lose the ability to validate users credentials, a different error message is reported.

This monitor is configured as a reverse monitor. We check that the receive string, **<MPSError type=\"IMA\">0x80130007</MPSError>**, is not returned.

The following is the code for the Send String in this monitor:

```
POST /scripts/wpnbr.dll HTTP/1.1\r\nContent-Type:
text/xml\r\nHost: 10.133.40.100\r\nContent-Length:
329\r\n\r\n<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\n<!DOCTYPE NFuseProtocol SYSTEM
\"NFuse.dtd\">\n<NFuseProtocol version=\"4.6\">\n
<RequestValidateCredentials>\n <Credentials>\n
<UserName>user</UserName>\n <Password
encoding=\"cleartext\">Password</Password>\n
<Domain>citrix</Domain>\n </Credentials>\n
</RequestValidateCredentials>\n</NFuseProtocol>\n
```

Send string for the optional XML Broker monitor

Important

The variables in Red text should be replaced as applicable for your deployment.

The following is the Receive String for this monitor:

```
<MPSError type=\"IMA\">0x80130007</MPSError>
```

Send string for the optional XML Broker monitor

When configuring this monitor on the BIG-IP LTM, the **User Name** should be **user**, and the **Password** should be **password**.
In the **Reverse** row, click the **Yes** button.

XML Broker Credential Validation health monitor 2

This health monitor tests that the XML Broker is capable of validating real user credentials. This monitor sends a **RequestValidateCredentials** request with a real user name, password, and domain name. These credentials will be passed in clear text, so if that is a security concern for your organization, the previous monitor may be preferable. In this case, we configure the monitor as a reverse monitor and mark the server down when the receive string, **<ErrorId>**, is returned. This string will only exist in the response if a failure occurs. For this health monitor, make certain to replace the user name, password, and domain name with valid entries and edit the Content-Length value to match.

The following is the code for the Send String in this monitor:

```
POST /scripts/wpnbr.dll HTTP/1.1\r\nContent-Type:
text/xml\r\nHost: 10.133.40.100\r\nContent-Length:
329\r\n\r\n<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\n<!DOCTYPE NFuseProtocol SYSTEM
\"NFuse.dtd\">\n<NFuseProtocol version=\"4.6\">\n
<RequestValidateCredentials>\n <Credentials>\n
<UserName>user</UserName>\n <Password
encoding=\"cleartext\">Password</Password>\n
<Domain>citrix</Domain>\n </Credentials>\n
</RequestValidateCredentials>\n</NFuseProtocol>\n
```

Send string for the optional XML Broker monitor

Important

The variables in Red text should be replaced as applicable for your deployment.

The following is the Receive String for this monitor:

```
<ErrorId>
```

Send string for the optional XML Broker monitor

When configuring this monitor on the BIG-IP LTM, from the **Reverse** row, click the **Yes** button.

Appendix C: Overview of Citrix Presentation Server environment

Citrix Presentation Server allows users to access and run applications hosted on a server or a farm of servers running Citrix Presentation Server software. In a Citrix Presentation Server environment, almost all of the application processing occurs on the server with only keystrokes, mouse-clicks and screen updates being exchanged by the client and the server. A Citrix Presentation Server farm is a grouping of multiple servers running the Citrix Presentation Server software, administered as a single entity.

A typical Citrix Presentation Server environment that allows users to access the applications using a web browser consists of the following components:

- Citrix Web Interface
- Citrix Presentation Server XML Brokers
- Citrix Presentation Server Zone Data Collectors
- Citrix Presentation Server farm

The Citrix Web Interface allows users to access applications hosted on the Citrix Presentation Server farm using a web browser. The Citrix Web Interface runs on a separate Web Server, such as a Microsoft Windows 2003 server running IIS. The Citrix Web Interface retrieves a list of applications hosted on the Citrix Presentation Server farm that the user can access, and publishes them as HTML pages that the users can view in a standard web browser.

The Citrix Web Interface servers and the Citrix Presentation Server farm communicate using the Citrix XML Service. The Citrix XML Service is a component of the Citrix Presentation Server and is present on all the Citrix Presentation Servers as well as the Citrix Web Interface servers. The Citrix XML Service wraps responses from IMA operations such as retrieving a list of applications that a user has access to, server address resolution among others, and encodes them into XML to be transmitted over HTTP.

In large deployments, a small number of Citrix Presentation Servers are designated as XML Brokers, dedicated to collecting data from other servers in the farm, such as the list of applications hosted on the farm that are available to the client. In such large deployments, the Citrix XML Brokers typically also act as the Zone Data Collectors to determine the least busy servers in the farm that can serve the applications, and providing the requested information to the Web Interface servers. One of the primary functions of the Citrix Zone Data Collector is to identify the least busy server within the farm or zone for a published application, send an IMA ping to ensure that the server is alive, and in case the target server has multiple IP Addresses determine the appropriate IP Address for the given client IP, before returning the target address to the XML Broker.

The following diagram illustrates the logical data flow between the client, the Citrix Web Interface and the Citrix Presentation Server farm.

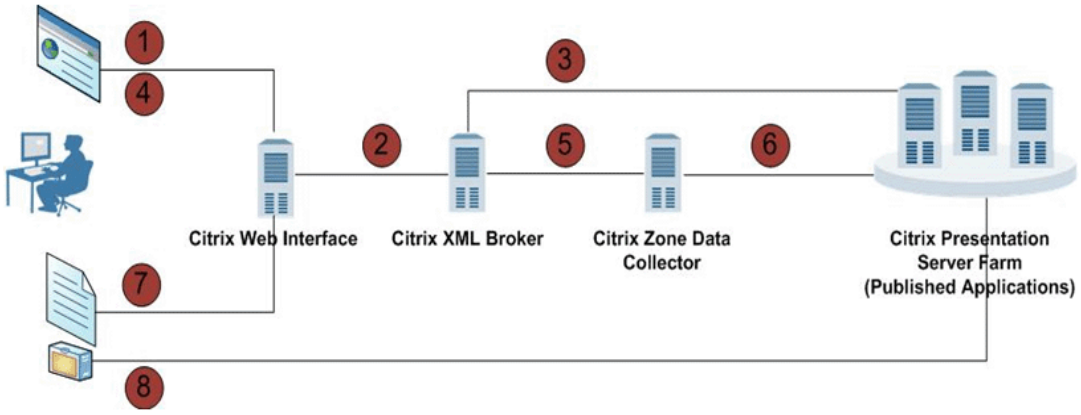

Figure 17 *Data flow in a Citrix Presentation Server environment*

1. The user initiates request for applications by sending credentials to the Web Interface, using a web browser.
2. The Web Interface sends an application retrieval request with the user's credentials to the XML Broker using the XML Service.
3. The XML Broker authenticates the user and retrieves the list of applications that the user can access from the Presentation Server Farm. The XML Broker returns the information about each application the user has access to, to the Web Interface using the XML Service. The Web Interface then publishes a HTML page with the application list to the user.
4. When the user clicks on an application icon on the HTML page, the Web Interface receives this request and uses the XML service to query the XML Broker for the address of the target Presentation Server that the user should connect to.
5. The XML Broker uses the IMA Service to request the target server information from the Zone Data Collector.
6. The Zone Data Collector identifies the address of the least busy Presentation Server hosting the application, sends an IMA ping to ensure that the server is alive, and returns it back to the XML Broker. The XML Broker relays the location of the least busy Presentation Server to the Web Interface using the XML Service.
7. The Web Interface generates a customized ICA file necessary to launch the application and delivers it to the user via the web browser.
8. The ICA file is executed by the client and the user initiates an ICA connection with the target Presentation Server hosting the application.

For more information on Citrix Presentation Server, see **www.citrix.com/English/ps2/products/product.asp?contentID=186**

Note

On February 11, 2008, Citrix changed the name of its Presentation Server product line to XenApp. This document is written with the assumption that you are familiar with the Citrix Presentation Server software. For more information on configuring the product, consult the appropriate documentation

Appendix D: Backing up and restoring the BIG-IP LTM system

We recommend saving your BIG-IP LTM configuration before you begin this configuration. When you save the BIG-IP configuration, it collects the following critical data and compress it into a single User Configuration Set (UCS) file:

- BIG-IP configuration files
- BIG-IP license and passwords
- SSL certificates
- SSH keys

Saving and restoring the BIG-IP configuration

The Configuration Management screen allows you to save and restore all configuration files that you may edit to configure a BIG-IP system. These configuration files are called a User Configuration Set (UCS). The Configuration Management screen contains sections for saving and restoring a configuration. The list boxes in these sections display only files in the **/usr/local/ucs** directory. If you want to save or restore files from another directory, you must type the full path in the box.

To save the BIG-IP configuration using the Configuration utility

1. In the navigation pane, click **System Admin**.
   The User Administration screen displays.
2. Click the Configuration Management tab.
   The Configuration Management screen displays.
3. In the **Save Current Configuration** section, type the path where you want your configuration file saved or choose a path from the list box. If no path is specified, the BIG-IP saves files to **/usr/local/ucs**. The BIG-IP appends the extension.ucs to file names without it.
4. Click the **Save** button to save the configuration file.

To restore a BIG-IP configuration

1. In the navigation pane, click **System Admin**.
   The User Administration screen displays.
2. Click the Configuration Management tab.
   The Configuration Management screen displays.
3. In the **Restore a Configuration** section, choose the configuration file you want to restore from the list box, or type the path where your configuration files were saved.

4. Click the **Restore** button.
   To check the status of the restoration, click the **View Log** button. You should wait a few moments for the log file to start generating before you click **View Log**. Repeated clicking of this button will update your screen with the most current log file information until the restoration is complete.