What's inside:

Configuring the BIG-IP LTM v11 for Oracle Database and RAC

Welcome to the F5 deployment guide for the BIG-IP Local Traffic Manager (LTM) and Oracle Databases, including Real Application Cluster (RAC) Databases. This guide describes how to configure the BIG-IP LTM for Oracle Database servers when you are looking to create optimized connections between the Oracle Database clients and servers.

Oracle RAC helps enterprises create greater levels of database performance, reliability, and scalability. When the BIG-IP device is added as a full TCP proxy in the Oracle Net environment, network performance, reliability, and faster client failover can be achieved. The BIG-IP device can monitor the RAC nodes, reduce TCP overhead on the servers, present a single Virtual IP Address for all clients, and provide Primary to Standby cluster failover connection management. Using these two technologies together can provide enterprise class database connection services for mission critical applications.

The appendices at the end of this guide contain optional ways to configure the BIG-IP for Instance name and Service name switching, as well as using iControl for light Fast Application Notification (FAN) integration.

For more information on Oracle RAC, see *http://www.oracle.com/technetwork/database/clustering/overview/index.html*.

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

Products and versions

Product	Version
BIG-IP LTM	11.x (11.1 or later recommended)
Oracle Database 11g Release 2	11.2.0.2.0

Important: *Make sure you are using the most recent version of this deployment guide, available at http://www.f5.com/pdf/deployment-guides/oracle-rac-database-ltmv11-dg.pdf.*

ORACLE
DATABASE
READY

Prerequisites and configuration notes

The following are general prerequisites and configuration notes for this guide:

➤ You must have an existing Oracle Database implementation.

➤ This guide is written for BIG-IP LTM v11 and later. If you are using v10.x, see *http://www.f5.com/pdf/deployment-guides/oracle-rac-database-dg.pdf*.

➤ This Deployment Guide contains configuration procedures for configuring the F5 devices only. For information on how to configure Oracle devices, refer to the appropriate Oracle documentation.

➤ We strongly recommend creating a new Oracle database user account specifically for use in the BIG-IP health monitor.

➤ Briefly review the basic configuration tasks and the few pieces of information, such as IP addresses, that you should gather in preparation for completing the F5 device configuration.

Configuration example

In this deployment guide, we use an example of two RAC clusters, with three nodes in each cluster. The first cluster is the RAC Primary, and second is the RAC Standby. The BIG-IP is located between the Oracle Net clients (application servers), and the Oracle Net servers (RAC nodes) as a full TCP proxy.

The clients connect to the BIG-IP system's virtual IP Address, and the BIG-IP system connects to the pool of RAC servers. The BIG-IP monitors each RAC node individually, using an SQL query to determine the health of each RAC node.

The BIG-IP load balancing pool uses an advanced feature, called Priority Groups, to direct traffic only to the RAC Primary nodes. If all of the RAC Primary nodes are unavailable for any reason, the BIG-IP directs the traffic to the RAC Standby nodes - the client side connections are properly reset and reconnected in the case of a RAC Primary to Standby transition event.

See the appendices for additional deployment options, such as Instance name and Service name switching using iRules, as well as light FAN integration.
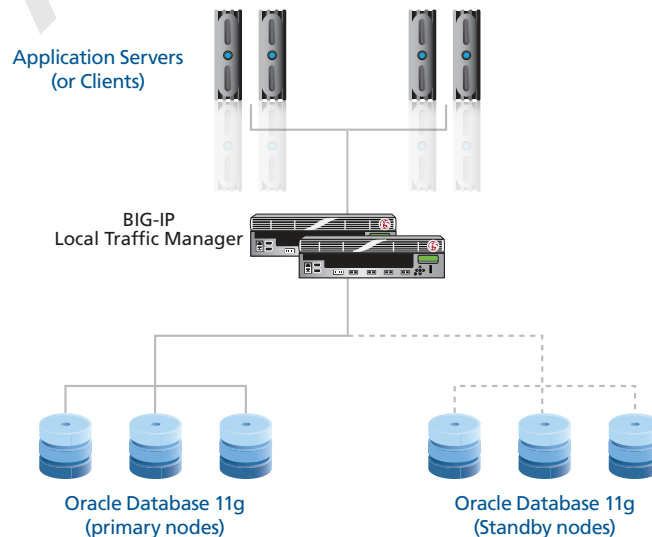

Figure 1: *Logical configuration example*

Configuring the BIG-IP LTM for Oracle Database

In this section, we configure the BIG-IP LTM for Oracle Database servers.

Creating the Oracle health monitor

The first task is to configure a health monitor for the Oracle Database. This procedure is optional, but very strongly recommended.

Note →

We assume that you have created a dedicated user in your Oracle Database for monitoring. This user should NOT be a privileged or administrative user, but have the proper permissions for monitoring the Database.

To configure the health monitor

1. On the Main tab, expand **Local Traffic**, and then click **Monitors**.
2. Click the **Create** button.
3. In the **Name** box, type a name. In our example, we type **ora11g-rac-primary_monitor**.
4. From the **Type** list, select **Oracle**.
5. From the **Configuration** list, select **Advanced**.
6. In the **Interval** box, specify the value that you used for your Oracle TCP.CONNECT_TIMEOUT setting. If you did not specify a value for the Oracle setting, leave the Interval value as the default. In our example, we type **10**.
7. In the **Timeout** box, type a timeout that is 3 times the value of the interval +1. In our example, we type 31.
8. In the **Send String** box, type the SQL query you want to perform at regular intervals for monitoring. In our example, we specify the following string:

   SELECT status FROM v$instance

9. In the **Receive String** box, type the response you expect to get back from your SQL query. In our example, we use the following string:

   OPEN

10. In the **User Name** box, specify the user you created specifically for monitoring. In our example, we type **bigip**.
11. In the **Password** box, type the associated password.
12. In the **Connection String** box, type the valid Oracle Net connection string you want to use to connect to your database; this is a free form field.

Note →

*The variables **%node_ip%** and **%node_port%** in the Connection String are dynamically substituted with the IP address and Port number of the Oracle Database instance to which the monitor is connecting.*

In our example, we type the following as a single line:

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=%node_ip%)(PORT=%node_port%))(CONNECT_DATA=(SERVICE_NAME=primary))(SERVER=dedicated))

Where *SERVICE_NAME=primary* specifies which Database service on the RAC system we want to health check.

13. The **Receive Row** and **Receive Column** boxes allow you to handle a SQL query that returns multiple results. If you know what you are looking for is in a specific spot in a multi-line,

multi-column result, this is where you would define it. In our example, we specify a value of 1 for both.

14. Click the **Finished** button. The new monitor is added to the Monitor list.

Creating the pool

The next step is to create a pool on the BIG-IP LTM system for the Oracle devices. A BIG-IP pool is a set of devices grouped together to receive traffic according to a load balancing method.

In this pool, we use the Priority Group feature on the BIG-IP LTM. The Priority Group Activation value specifies the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group. This setting is used in tandem with the Priority Group setting for individual pool members. If the number of available members assigned to the highest priority group drops below the number that you specify in the Priority Group Activation setting, the BIG-IP system distributes traffic to the next highest priority group.

In this case, we use the Priority feature to weight the primary RAC nodes higher than the standby nodes. In this way, the primary nodes will receive the traffic first, however, if all of the primary nodes are down, traffic will flow to the standby nodes.

If you are using the B IG-IP LTM to conduct service switching as described in *Appendix B: Service Name Switching using iRules on page 10*, you create a pool for each database (a database can be a standalone or RAC system of databases).

To create the pool

1. On the Main tab, expand **Local Traffic**, and then click **Pools**.

2. Click the **Create** button. The New Pool screen opens.

3. From the **Configuration** list, select **Advanced**.

4. In the **Name** box, type a name. We use **ora11g-rac-primary-standby_pool**.

5. In the **Health Monitors Available** list, select the monitor you created and then click the Add (**<<**) button. In our example, we select **ora11g-rac-primary_monitor**.

6. From the **Action On Service Down** list, select **Reject**.

7. 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 (node)**.

8. From the **Priority Group Activation** list, select **Less Than**.

9. In the **Available Members** box, specify the minimum number of nodes at a particular Priority Level that should be available before the Pool chooses a lower Priority Level. In our example, we type a value of **1**.

10. In the **New Members Address** box, type the IP Address of your first primary RAC node. In our example, we type: **10.133.17.31**.

11. In the **Service Port** box, specify the Port of your first primary RAC node. In our example, we type **1521**.

12. In the **Priority** box, specify a number to assign as a Priority Level for your primary RAC nodes. In our example, we type **100**.

13. Click the **Add** button.

14. Repeat steps 9-12 for each of your primary RAC nodes.

15. Repeat steps 9-12 for each of your standby RAC nodes, ***using a lower Priority Level***. In our example, the standby nodes have a Priority level of **50**.

16. Click **Finished**.

17. *Optional*: If you are configuring service switching as described in *Appendix B: Service Name Switching using iRules on page 10*, repeat this procedure for each database.

Creating TCP profiles

The next task is to create the TCP profiles. A profile is an object that contains user-configurable settings for controlling the behavior of a particular type of network traffic.

Because Oracle client behavior differs depending on software versions, author preference, and so on, we use separate TCP profiles on each side of the BIG-IP to more cleanly handle failover and optimize both sides of the connection.

Although it is possible to use the default profiles, we strongly recommend you create new profiles based on the default parent profiles, even if you do not change any of the settings initially. Creating new profiles allows you to easily modify the profile settings specific to this deployment.

Creating the client-side TCP profile

First we configure the client-side TCP profile. In the following procedure, you need to know a number of values in your Oracle Net settings (formerly Oracle SQL*Net). Determine these values before starting this procedure.

Note ➞ *In our environment, our clients are application servers on a LAN segment, local to our database servers. Therefore, we use the **tcp-lan-optimized** parent profile. Every environment is different, so please the parent TCP profile most suitable to your configuration.*

To create a new TCP profile

1. On the Main tab, expand **Local Traffic**, and then click **Profiles**.

2. On the Menu bar, from the **Protocol** menu, click **TCP**.

3. Click the **Create** button.

4. In the **Name** box, type a name. In our example, we type **ora11g-rac-client-tcp_profile**.

5. From the Parent Profile list, select **tcp-lan-optimized**.

6. In the Settings section, click the **Custom** box above the table. This checks all the boxes and allows you to change the following settings.

7. In the **Maximum Syn Retransmissions** box, type the value of the TCP.CONNECT_TIMEOUT in your Oracle Net settings.

   If you have not changed the value of TCP.CONNECT_TIMEOUT in your Oracle configuration from the default, you do not need to change the value of Maximum Syn Retransmissions in your BIG-IP configuration.

   However, if you set your TCP.CONNECT_TIMEOUT value to the Oracle recommended value of 10 seconds, change the value of **Maximum Syn Retransmissions** to **5**. See the note on the following page.

Note ➡

The wait time between SYN retransmissions on the BIG-IP is exponential. For example, the first retransmission happens after 1 second, the second retransmission after 2 seconds, the third after 4 seconds, the fourth after 8 seconds, etc. In order to not interrupt the Oracle recommended setting of 10 seconds, we recommend a value of 5 (16 seconds) in the BIG-IP configuration.

For more information on configuring the Oracle TCP Connection Timeout, see:
http://download.oracle.com/docs/cd/B28359_01/network.111/b28317/sqlnet.htm#BIIDDACA.

8. In the **Idle Timeout** box, type the value of the SQLNET.EXPIRE_TIME in your Oracle Net settings.

    If you have not changed the value of SQLNET.EXPIRE_TIME in your Oracle configuration from the default, you do not need to change the value of Idle Timeout in your BIG-IP configuration.

    However, if you set your SQLNET.EXPIRE_TIME value to the Oracle recommended value of 10 minutes, change the value of Idle Timeout to a value that is slightly larger, such as 660 (11 minutes).

    For more information on configuring your Oracle Net Expire Timeout, see
    http://download.oracle.com/docs/cd/B28359_01/network.111/b28317/sqlnet.htm#BIIEEGJH

    Setting 'Idle Timeout' to a value that is slightly larger than the Oracle SQLNET.EXPIRE_TIME setting will ensure that the BIG-IP does not prematurely close an idle connection, but will help to cleanup stale connections.

9. In the **Send Buffer** box, type a value that is larger than the Oracle Net RECV_BUF_SIZE setting. The default size for BIG-IP is 64KB, which is sufficient for the Oracle default size of 8KB.

    The Oracle default is specific to the Operating System on which your database is running. For more information on configuring your Oracle socket buffer sizes, see:
    http://download.oracle.com/docs/cd/B28359_01/network.111/b28316/performance.htm#NETAG0142

10. In the **Receive Window** box, type a value that is larger than the Oracle Net SEND_BUF_SIZE setting. The default size for BIG-IP is 64KB, which is sufficient for the Oracle default size of 32KB.

    The Oracle default is specific to the Operating System on which your database is running.

    For more information on configuring your Oracle socket buffer sizes, see:
    http://download.oracle.com/docs/cd/B28359_01/network.111/b28316/performance.htm#NETAG0142

11. In the **Keep Alive Interval** box, type a value that is larger than the value that you specified for **Idle Timeout**. If you do not want the BIG-IP to use Keep Alives on a connection, set the value to **Indefinite**.

12. Click the **Finished** button.

Creating the server-side TCP profile
The next task is to create the server-side TCP profile.

To create the server-side TCP profile, use the procedure Creating the client-side TCP profile, on page 8, with the following exceptions:

- Step 4: Give this profile a unique name. We use **ora11g-rac-server-tcp_profile**.

- Step 7: Do NOT configure Maximum Syn Retransmissions, leave this setting at the default.

Creating the virtual server

Next, we configure a virtual server that references the profiles and pool you created in the preceding procedures.

To create the virtual server

1. On the Main tab, expand **Local Traffic**, and then click **Virtual Servers**.

2. Click the **Create** button.

3. In the **Name** box, type a name. We use **ora11g-rac-primary-standby_vs**.

4. In the Destination section, select the **Host** option button if necessary.

5. In the **Address** box, type the IP address of this virtual server. In our example, we use **10.133.17.121**.

6. In the **Service Port** box, type the Port. In our example, we use **1521**.

7. From the **Configuration** list, select **Advanced**.

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 *Creating the client-side TCP profile on page 5*. In our example, we select **ora11g-rac_client-tcp-profile**.

10. From the **Protocol Profile (Server)** list, select the name of the profile you created in *Creating the server-side TCP profile on page 6*. In our example, we select **ora11g-rac_server-tcp-profile**.

11. From the **SNAT Pool** list, select **Auto Map**.

12. In the Resources section, from the **Default Pool** list, select the pool you created in *Creating the pool on page 4*. In our example, we select **ora11g-rac-primary-standby_pool**.

Important ➞ *If you are configuring Service as described in <u>Appendix B: Service Name Switching using iRules on page 10</u>, do **NOT** select a default pool here. See Appendix B for more information*.

13. From the **Default Persistence Profile** list, select **dest_addr**.

14. From the **Fallback Persistence Profile** list, select **source_addr**.

15. Click the **Finished** button.

This completes the configuration. See the appendices on the following pages for optional, advanced configuration.

Appendix A: Instance name switching using iRules

Some applications and clients, especially those using older libraries, include an INSTANCE_NAME in their Oracle Net connect strings when connecting to an Oracle database. This can be troublesome when trying to provide load balancing and high availability for a RAC system. By default, the Listener on a database usually denies a connection if the client specifies an Instance Name that does not match the name known to it.

For example, we have a 3-node RAC system hosting a database named **orcl**. In most cases, the individual nodes in the RAC system have an Instance Name derived from the Database Unique Name (such as **orcl1**, **orcl2**, and **orcl3**). The application is configured to connect to the BIG-IP virtual server in front of the RAC system. You need to make sure that if the connect string states to connect to instance **orcl2**, the connection is actually sent to the node that hosts that instance.
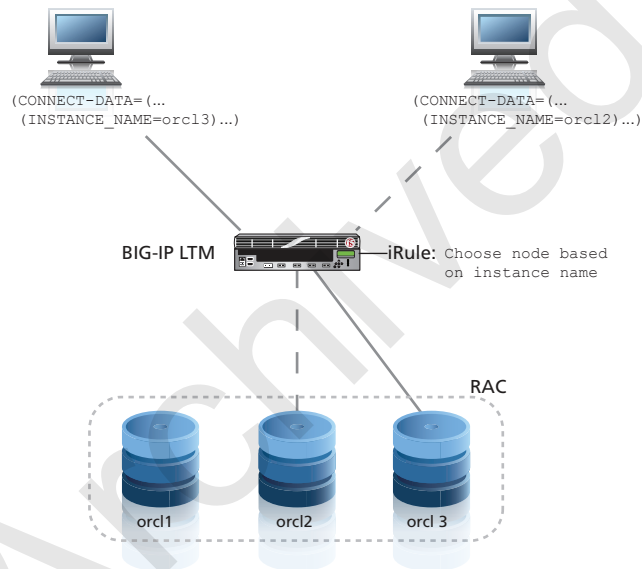
Logical configuration example of Instance Name switching

In this Appendix, we show how you can optionally add an iRule to send traffic to different Oracle servers based on the Instance name. We assume you have already configured the BIG-IP LTM as described in this guide. The only additional requirement for trying out this scenario is to attach an iRule to perform the actual INSTANCE_NAME switching.

Note ➔ *In some installations, depending on how your Listener is configured on your RAC nodes, it may be required to rewrite the HOSTNAME in the connect string in order for the RAC node to accept the connection. This ability is included in the iRule, but is turned off by default. In certain situations, you may need to enable this functionality.*

Creating the iRule

The first task is to create the iRule on the BIG-IP system. Because of the length of the iRule, we do not include the text in this guide. First, download the iRule from the following location: *http://www.f5.com/solutions/resources/deployment-guides/files/instance-switching-irule.txt* and then use the following procedure to create the iRule.

To create the iRule

1. Open the following text file and copy the contents
   http://www.f5.com/solutions/resources/deployment-guides/files/instance-switching-irule.txt

2. From the Main tab of the BIG-IP, expand **Local Traffic**, and then click **iRules**.

3. Click the **Create** button.

4. In the **Name** box, type a name. In our example, we type **instance-switching**.

5. In the **Definition** section, paste the iRule you copied in step 1.

6. Modify the **switch_map** array in the CLIENT_ACCEPTED event in the iRule to properly map your known Oracle Net Service Instance Names to the node where the associated instance resides.

Important ➞ *If the client tries to connect to a instance name that doesn't exist in the **switch_map**, the BIG-IP LTM closes the connection and logs a warning.*

7. Click **Finished**.

Next, you modify the virtual server you created in Creating the virtual server, on page 10 to use the iRule you just created.

**To modify the virtual server to use the iRule**

1. On the Main tab, expand **Local Traffic**, and then click **Virtual Servers**.

2. From the **Virtual Server** list, click the name of the virtual server you created in *Creating the virtual server on page 7*. In our example, we click **ora11g-rac-primary-standby_vs**.

3. On the Menu bar, click **Resources**. The Resources screen for the Virtual Server opens.

4. In the **iRules** section, click the **Manage** button.

5. From the **Available** list, select the name of the iRule you created in the preceding procedure, and then click the Add (**<<**) button to move it to the **Enabled** box. In our example, we select **instance-switching**.

6. Click the **Update** button.

This completes the configuration in this appendix.

Appendix B: Service Name Switching using iRules

In many enterprises, there are applications that need access to various databases which can be located in different areas of the network. It can be difficult to maintain which databases reside in which location, and to keep all databases current in the configuration of each application.

In the scenario presented in this appendix, all applications use the BIG-IP virtual server IP address as the host for all databases. This virtual server includes an iRule that checks which SERVICE_NAME is present in the Oracle Net connect string, and then switches the connection to the correct pool containing the nodes on which the database resides.
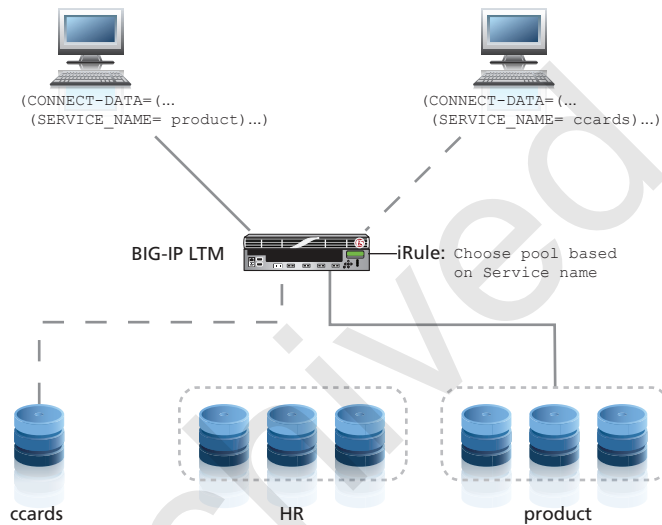
Logical configuration example of Instance Name switching

We assume you have used this guide to configure the BIG-IP LTM monitors, TCP profiles, pools containing the nodes (RAC or standalone) for each database, and virtual server. As noted in the virtual server configuration section, you should not assign the pool to this virtual server. If you do attach a pool to the virtual server, the LTM could send SQL traffic to a database that it is not intended to receive it (in the event that there is no SERVICE_NAME match).

If you did configure the virtual server with a pool, go back into the virtual server configuration and remove the default pool from the virtual server.

Note →

In some installations, depending on how your Listener is configured on your RAC nodes, it may be required to remove the INSTANCE_NAME in the connect string in order for the RAC node to accept the connection. This ability is included in the iRule, but is turned off by default. In certain situations, you may need to enable this functionality.

Creating the iRule

The first task is to create the iRule on the BIG-IP system. Because of the length of the iRule, we do not include the text in this guide. First, download the iRule from the following location:
http://www.f5.com/solutions/resources/deployment-guides/files/service-name-switching-irule.txt
and then use the following procedure to create the iRule.

To create the iRule

1. Open the following text file and copy the contents
   http://www.f5.com/solutions/resources/deployment-guides/files/service-name-switching-irule.txt

2. From the Main tab of the BIG-IP system, expand **Local Traffic** and then click **iRules**.

3. Click the **Create** button.

4. In the **Name** box, type a name. In our example, we type **service-name-switching**.

5. In the **Definition** section, paste the iRule you copied in step 1.

6. Modify the **switch_map** array in the CLIENT_ACCEPTED event in the iRule to properly map your known Oracle Net Service Names to the pool where the associated database resides.

   *If the client tries to connect to a service name that doesn't exist in the **switch_map**, the BIG-IP LTM closes the connection and logs a warning.*

7. Click **Finished**.

Next, you modify the virtual server you created in Creating the virtual server, on page 10 to use the iRule you just created.

To modify the virtual server to use the iRule

1. On the Main tab, expand **Local Traffic**, and then click **Virtual Servers**.

2. From the **Virtual Server** list, click the name of the virtual server you created in *Creating the virtual server on page 7*. In our example, we click **ora11g-rac-primary-standby_vs**.

3. On the Menu bar, click **Resources**. The Resources screen for the Virtual Server opens.

4. In the **iRules** section, click the **Manage** button.

5. From the **Available** list, select the name of the iRule you created in the preceding procedure, and then click the Add (**<<**) button to move it to the **Enabled** box. In our example, we select **service-name-switching**.

6. Click the **Update** button.

This completes the configuration in this appendix.

Appendix C: Integrating with Oracle's Fast Application Notification using iControl

Using the BIG-IP LTM's Oracle health monitor, you can monitor most everything in your database using standard SQL queries. However, there can be scenarios where you do not want traffic sent to a node, but these scenarios are not discernible using results from a SQL query. In an Oracle RAC installation, there are circumstances where a node still accepts SQL queries, but the cluster has decided this particular node is no longer a valid destination. This can be the result of a high load on the node, cluster brownout, or node demotion in the cluster for a variety of reasons.

Oracle Fast Application Notification (FAN) is a system designed for these types of situations. With FAN, all clients connected to a RAC system can be notified when they should no longer be directing traffic to a particular node or cluster for a time.

Unfortunately, not all clients and applications natively support FAN. However the BIG-IP system, using iControl, can be configured to help. In addition to offering load balancing and high availability for Oracle databases, the BIG-IP can perform automatic node and network availability tasks when the RAC system sends out FAN events.

The BIG-IP system handles Oracle FAN events by using a feature built-in to RAC and FAN called server-side callouts. Server-side callouts can be configured to execute an arbitrary script on a RAC node when a particular event is triggered. By having a script on each node that integrates with your BIG-IP using iControl, you can preemptively control the status of every node in a RAC cluster for which your BIG-IP is providing load balancing and high availability. Your clients and/or applications only connect to valid and enabled nodes, even in the event the node monitor is returning true.

For more information on Oracle Fast Application Notification and setting up your RAC system to use server-side callouts, see
http://download.oracle.com/docs/cd/E11882_01/rac.112/e16795/hafeats.htm.

For more on information on F5 iControl, see:
http://devcentral.f5.com/wiki/iControl.HomePage.ashx.

Configuring FAN and iControl integration

To use Oracle's Fast Application Notification via server-side callouts, you need to place the iControl script on each node that you want to integrate with your BIG-IP.

To install and configure the iControl script

1. Download the iControl Perl script from the following location:
    http://www.f5.com/solutions/resources/deployment-guides/files/f5-oracle-fan.zip.

2. Extract the **f5-oracle-fan.pl** file from the archive.

3. Copy the script to your Oracle Database RAC node.

4. Make any necessary modifications in the CONFIGURATION SECTION in the script.

    At minimum, you need to define the iControl **URL**, **username** and **password**.

5. Check to make sure the script runs on your server node without any errors by running it through the Perl syntax checker.

    Use the following command:

    ```
    perl -c f5-oracle-fan.pl
    ```

    The shell should echo back:

`f5-oracle-fan.pl syntax OK`

The following Perl modules are required:

- » MIME::Base64
- » SOAP::Lite
- » Socket
- » Switch

6. Place the script in the following directory: **$GRID_HOME/racg/usrco/**.

Note ➞ *The file needs to be executable in order to run. You may need to change the file permissions to make it executable depending on your system.*

7. Repeat steps 2-5 for each database server node you want to integrate with your BIG-IP.

The script logs activity to /tmp/f5-oracle-fan.log. Check this log file often as it confirms which actions it is taking as to the status of nodes on the BIG-IP, as well as logging FAN Event Messages that it receives but does not act upon. This gives you a starting point for defining filters to handle other events.

Note ➞ *By default, the FAN iControl integration script only handles **NODE UP** and **NODE NODEDOWN** events. If you want the script to handle other events, you need to modify the script and define your own filters, as well as define new logic to handle these new events. FAN Event filters are defined in the **@fan_event_filter** array.*

This completes the configuration.

Document Revision History

Version	Description
1.0	New Deployment Guide for BIG-IP v11