



---

---

# Load Balancing BEA WebLogic Servers with F5 Networks BIG-IP v9

---

---

- Introducing BIG-IP load balancing for BEA WebLogic Server
- Configuring the BIG-IP for load balancing WebLogic Servers

---

# Introducing BIG-IP load balancing for BEA WebLogic Server

F5 Networks and BEA® systems have created a highly effective way to direct traffic for WebLogic Server™ with the BIG-IP application traffic manager. BEA WebLogic Server is the number one application server on the market, and the core of today's most reliable enterprise applications. F5 Networks BIG-IP system is a secure, highly available and scalable application traffic management device. This strong interoperability and integration provides a solution that delivers unparalleled load balancing functionality for those deploying services and applications on the WebLogic Enterprise Platform™.

This Deployment Guide has been updated to include a number of ways to optimize BEA WebLogic deployments using version 9 of the BIG-IP system. Configuration procedures for these features (such as Intelligent Compression and Fast Cache) are included in this guide, but are marked as optional. For general information on these features and the interoperability of F5 products and BEA WebLogic Server, see the BEA WebLogic **Application Ready Network Guide**.

## Prerequisites and configuration notes

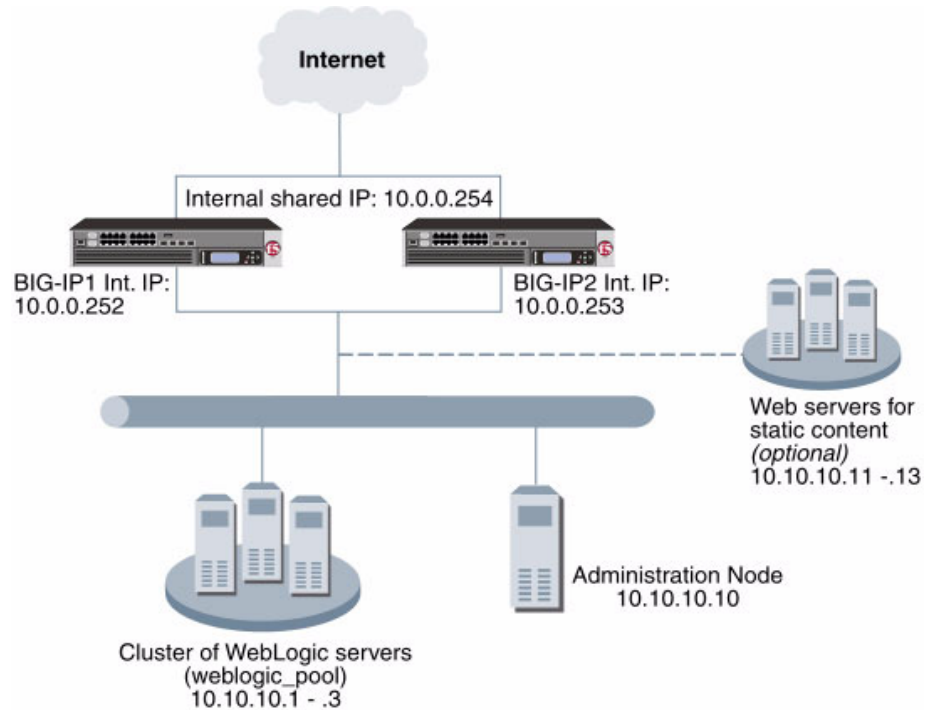
All of the procedures in this Deployment Guide are performed on the BIG-IP system. The configuration described in this document assumes that your WebLogic servers are already in a clustered environment. For information on how to configure a WebLogic server cluster, see the *BEA WebLogic Server: Using WebLogic Server Clusters* guide, available on the BEA web site at <http://www.bea.com>.

- ◆ The WebLogic server should be running version 5.1 or later. This Deployment Guide has been tested with BEA WebLogic version 8.1.
- ◆ For this Deployment Guide, the BIG-IP system must be running version 9.0 or later. For deploying WebLogic Servers with BIG-IP versions 4.x, see <http://www.f5.com/pdf/deployment-guides/boa-bigip45-dg.pdf>.
  - For certain *optional* optimization features, the appropriate module must be licensed (such as compression and caching) and in some cases (like caching) the BIG-IP system must be running version 9.0.5 or later.

## Configuration example

Using the configuration in this guide, the BIG-IP system is optimally configured to load balance traffic to BEA WebLogic servers. Figure 1.1 shows a typical configuration with a redundant pair of BIG-IP devices, a

cluster of WebLogic servers, and a WebLogic administration node. Figure 1.1 also shows an optional pool of Web servers which host static content for traffic that does not need to be sent to the WebLogic application servers.



*Figure 1.1 Example Configuration*

◆ **Note**

*For the rest of this document, we use the IP addresses shown in Figure 1.1 in our examples.*

## Configuring the BIG-IP for load balancing WebLogic Servers

To configure the BIG-IP product to load balance WebLogic Servers, you need to complete the following tasks:

- *Creating the HTTP health monitor*
- *Creating the pool*
- *Creating profiles*
- *Creating the virtual server*
- *Configuring an optional rule to send static content to the Web servers*

---

## Creating the HTTP health monitor

The first step is to set up health monitors for the WebLogic devices. This procedure is optional, but very strongly recommended. We configure the health monitors first in version 9.0 and later, as health monitors are now associated during the pool configuration.

For this configuration, we use an HTTP Extended Content Verification (ECV) monitor, which checks nodes (IP address and port combinations), and can be configured to use **send** and **recv** statements in an attempt to retrieve explicit content from nodes.

### To configure a health monitor from the BIG-IP 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 **bea\_http\_monitor**.
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 Interval and Timeout. We recommend at least a 1:3 +1 ratio between the interval and the timeout (for example, the default setting has an interval of **5** and an timeout of **16**). In our example, we use a **Interval** of **30** and a **Timeout** of **91**.
6. In the Send String and Receive Rule sections, you can add a Send String and Receive Rule specific to the device being checked.

**Important Note:**

When using the **GET** send string, you must end the string by including the HTTP protocol at the end of the statement. Use the following syntax:

**GET <fully qualified path name> HTTP/1.0**

For example:

**GET /www/support/customer\_info\_form.html HTTP/1.0**

The screenshot shows the 'New Monitor...' configuration window. The 'General Properties' section includes:

- Name: bea\_http\_monitor
- Type: HTTP
- Import Settings: http

The 'Configuration' section is set to 'Basic' and includes:

- Interval: 30 seconds
- Timeout: 91 seconds
- Send String: GET /
- Receive String: (empty)
- User Name: (empty)
- Password: (empty)
- Reverse: No
- Transparent: No

Buttons at the bottom are 'Cancel', 'Repeat', and 'Finished'.

*Figure 1.2 Creating the HTTP Monitor*

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

## Creating the pool

The first step is to define a load balancing pool for the WebLogic servers. A BIG-IP pool is a set of devices grouped together to receive traffic according to a load balancing method. This pool uses the monitor you just created.

1. On the Main tab, expand **Local Traffic**.
2. Click **Pools**.  
The Pool screen opens.

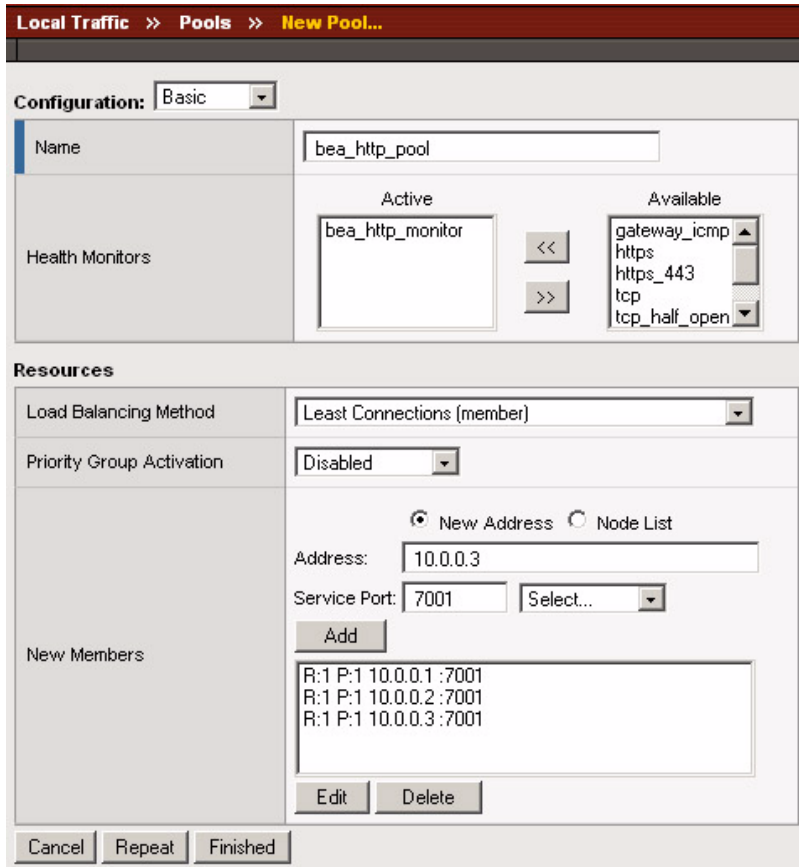
- 
3. In the upper right portion of the screen, click the **Create** button. The New Pool screen opens.

*Note: For more (optional) pool configuration settings, from the Configuration list, select **Advanced**. Configure these settings as applicable for your network.*

4. In the **Name** box, type a name for your pool.  
In our example, we use **bea\_http\_pool**.
5. In the **Health Monitors** section, select the name of the monitor you created in the *Creating the HTTP health monitor* section, and click the Add (<<) button. In our example, we select **bea\_http\_monitor**.
6. 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)**.
7. In this pool, we leave the Priority Group Activation **Disabled**.
8. In the New Members section, make sure the **New Address** option button is selected.

*Tip: As this is the second pool you are creating, you can click the **Node List** option button, then select the web server IP Addresses from the list, so you do not have to type them again.*

9. In the **Address** box, add the first server to the pool. In our example, we type **10.0.0.1**
10. In the **Service Port** box, type the **7001**, the default service for WebLogic.
11. Click the **Add** button to add the member to the list.
12. Repeat steps 9-11 for each server you want to add to the pool.  
In our example, we repeat these steps three times for the remaining servers, **10.0.0.2** and **.3** (see Figure 1.31.3).
13. Click the **Finished** button.



*Figure 1.3 Adding the bea\_http\_pool in the BIG-IP Configuration utility*

14. *Optional:* If your configuration includes web servers to serve static content, repeat the procedure above to create a new pool for the web servers, naming the pool something descriptive, like **bea\_images**. Later in this configuration, you will create a **rule** that sends the static content to this image pool. For information on how to configure this rule, see the *Configuring an optional rule to send static content to the Web servers* section at the end of this guide.

## Creating profiles

BIG-IP version 9.0 and later use profiles. 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. Using profiles enhances your control over managing network traffic, and makes traffic-management tasks easier and more efficient.

---

Although it is possible to use the default profiles, we strongly recommend you create new profiles based on the default parent profiles. Creating new profiles allows you to easily modify the profile settings specific to this deployment, and ensures you do not accidentally overwrite the default profile.

For this configuration, we create two new profiles: an HTTP profile and a cookie persistence profile.

There is an optional profile (TCP) and additional configuration steps if you want to optimize the BIG-IP for WebLogic deployments. These optional portions of the configuration will be clearly marked with *Optional Optimization*:

## Creating an HTTP profile

The first new profile we create is an HTTP profile. In our example, we leave all the options at their default settings, unless you are configuring the optimized deployment. The HTTP profile is where the optional Intelligent Compression and Fast Cache configuration options are located (modules which must be licensed on your BIG-IP system). If you have not licensed the modules, you will not see the options described in the procedures. Fast Cache is only available in version 9.0.5 and later.

### ◆ Note

*The following procedure shows one way to optimize the BEA WebLogic configuration that has been tested in real-world scenarios by F5 using the Gomez Performance Network, and shown to give the greatest improvement. These procedures and the specific values given in some steps should be used as guidelines, modify them as applicable to your configuration.*

### To create a new HTTP profile based on the default HTTP profile

1. On the Main tab, expand **Local Traffic**.
2. Click **Profiles**.  
The HTTP Profiles screen opens.
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 **bea\_http\_profile**.
5. From the **Parent Profile** list, ensure that **HTTP** is selected.

***Optional Optimization:** The following 8 steps are optional and show one way to optimally configure compression on the BIG-IP system. If your configuration does not include compression on the BIG-IP system, skip to Step 14.*

6. In the Settings table, from the **Response Chunking** section, click a check in the Custom box. From the list, select **Unchunk**. This allows for more efficient caching and compression.

7. In the Compression table, from the Compression row, click a check in the Custom box, then select **Enabled** from the list.
8. In the Content list section, we leave the settings at the default level, configure as applicable for your deployment.
9. In the Compression Buffer Size section, click a check in the Custom box. In the **Compression Buffer Size** box, type **131072**.
10. In the gzip Compression Level section, click a check in the Custom box. From the list, select a level of compression suitable to your configuration. For most compression, select **9 - Most Compression (Slowest)**.
11. In the gzip Memory Level section, click a check in the Custom box. From the list, select **16** kilobytes.
12. In the gzip Window size section, click a check in the Custom box. From the list, select **64** kilobytes.
13. In the HTTP/1.0 Requests section, click a check in the Custom box. Click a check in the box to enable HTTP/1.0 requests.

***Optional Optimization:** The following 7 steps are optional and show one way to optimally configure caching on the BIG-IP system. If your configuration does not include Fast Cache on the BIG-IP system, click the **Finished** button.*

14. In the RAM Cache table, click a check in the Custom boxes for all the settings except Maximum Entries, URI Caching, Ignore Headers, and Aging Rate.
15. In the Ram Cache section, select **Enabled** from the list.
16. In the Maximum Cache Size section, type **10** in the box.
17. In the Maximum Age section, type **86400** seconds.
18. In the Minimum Object Size section, type **0**.
19. In the Maximum Object Size section, type **2,000,000** bytes.
20. In the Insert Age Header section, select Disabled from the list.
21. Click the **Finished** button.

For more information on creating or modifying profiles, or applying profiles in general, see the BIG-IP documentation.

## Creating a cookie persistence profile

The next profile we create is a Cookie Persistence profile. We recommend using the default cookie method for this profile (HTTP cookie insert), but modify other settings, such as specifying a cookie expiration, as applicable for your network.

---

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

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

For more information on creating or modifying profiles, or applying profiles in general, see the BIG-IP documentation.

### Creating an optional TCP profile

The BIG-IP system's TCP Express feature set provides a number of enhancements and optimizations to TCP handling that enhance end user experience. Configuring a TCP profile is an *Optional Optimization*, use the following procedure only if applicable to your deployment.

### To create a TCP profile based on the default profile

1. On the Main tab, expand **Local Traffic**.
2. Click **Profiles**.  
The Profiles screen opens.
3. On the menu bar, from the **Protocol** menu, select **TCP**.  
The TCP Profiles screen opens.
4. Click the **Create** button.  
The New TCP Profile screen opens.
5. In the Name box, type a name for the profile. In our example, we type **bea\_tcp**.
6. In the **Parent Profile** list, make sure that **tcp** is selected.
7. In the Configuration table, locate **Proxy Buffer Low**, and click a check in the Custom box on the far right. In the Proxy Buffer Low box, type **131072**.
8. In the **Proxy Buffer High** section, click a check in the Custom box, and in the Proxy Buffer High box, type **131072**.

9. In the **Send Buffer** section, click a check in the Custom box, and in the Send Buffer box, type **65535**.
10. In the **Receive Window** section, click a check in the Custom box, and in the Receive Window box, type **65535**.
11. Click the **Finished** button.  
The new profile appears in the TCP profiles list.

## Creating the virtual server

The next step in this configuration is to define a virtual server that references the pool and the HTTP and cookie persistence profiles you created in the preceding procedures.

### To create the HTTP virtual server using the Configuration utility

1. On the Main tab, expand **Local Traffic**.
2. Click **Virtual Servers**.  
The Virtual Server screen opens.
3. In the upper right portion of the screen, click the **Create** button.  
The New Virtual Server screen opens.
4. In the **Name** box, type a name for this virtual server. In our example, we type **bea\_http\_vs**.
5. In the **Destination** section, select the **Host** option button.
6. In the **Address** box, type the IP address of this virtual server. In our example, we use **192.168.10.100**.
7. In the **Service Port** box, type the service port, or select it from the list. In our example, we select **HTTP**.

General Properties	
Name	bea_http_vs
Destination	Type: <input checked="" type="radio"/> Host <input type="radio"/> Network Address: 192.168.200.10
Service Port	80 HTTP
State	Enabled

*Figure 1.4 General Properties of the add virtual server page*

---

**Optional Optimization:** If you configured a TCP profile to optimize your configuration, from the Configuration list, select **Advanced**. From the Client Protocol Profile list, select the name of the TCP profile you created in the *Creating an optional TCP profile* section. In our example, we select **bea\_tcp**. Continue with Step 8.

8. In the Configuration section, leave the **Type** list at the default setting: **Standard**.
9. In the HTTP profile section, select the name of the profile you created in the *Creating an HTTP profile* section. In our example, we select **bea\_http\_profile**.  
Configure the rest of the Configuration section as applicable for your environment.

Configuration:	Basic
Type	Standard
Protocol:	TCP
OneConnect Profile	None
HTTP Profile	bea_http_profile
FTP Profile	None
SSL Profile (Client)	None
SSL Profile (Server)	None
VLAN Traffic	All VLANs

**Figure 1.5** Configuration properties of the add virtual server page

10. In the Resources section, from the **Default Pool** list, select the pool you created in the *Creating the pool* section. In our example, we select **bea\_http\_pool**.
11. From the **Default Persistence Profile** list, select the name of the profile you created in the *Creating a cookie persistence profile* section. In our example, we select **bea\_cookie** (see Figure 1.6).

**Note:** The cookie used in cookie persistence Insert mode resides in memory, and is not written to disk.

The screenshot shows the 'Resources' configuration page. It includes a table for iRules, a 'Default Pool' dropdown menu with 'bea\_http\_pool' selected, a 'Default Persistence Profile' dropdown menu with 'bea\_cookie' selected (circled in blue), and a 'Fallback Persistence Profile' dropdown menu with 'None' selected. There are also 'Up' and 'Down' buttons for the iRules list, and 'Cancel', 'Repeat', and 'Finished' buttons at the bottom.

**Figure 1.6** Resources section of the add virtual server page

12. 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 an optional rule to send static content to the Web servers

### ◆ Important

*This section is only necessary if your configuration includes web servers for static content. If you do not have web servers for static content, you do not need to follow the procedures below.*

---

If your configuration includes web servers to serve static content, you must create an iRule on the BIG-IP system that sends the static content to the web servers.

Before you start the following procedure, you must have a pool that contains the web servers. If you have not already created the web server pool, follow the procedure in the *Creating the pool* section, substituting the information from the web servers hosting the static content. In our example, we name the pool **bea\_images**.

◆ **Note**

*In the procedure below, the rule uses the line `[matchclass [HTTP::uri] ends_with $::images]`. `$::images` refers to a predefined class on the BIG-IP system named **images** that includes **.bmp**, **.jpg**, and **.gif** extensions. You can modify this class to include other types of files (such as **.html**), or create a new class that contains the file types applicable to your configuration. For information on how to modify or create a class, see the **BIG-IP Reference Guide**, or the online help.*

### To create a rule for static content using the Configuration utility

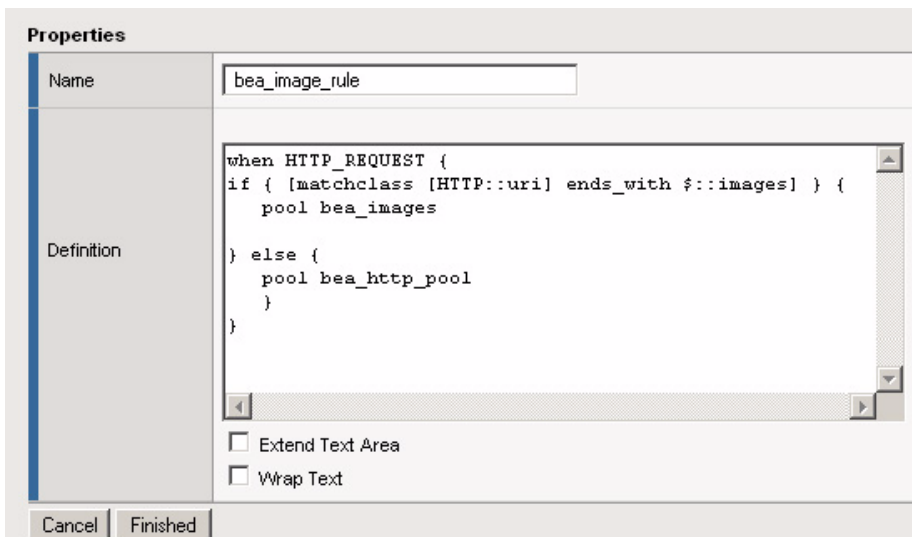
1. On the Main tab, expand **Local Traffic**.
2. Click **iRules**.  
The iRules screen opens.
3. In the upper right portion of the screen, click the **Create** button.  
The New iRules screen opens.
4. In the **Name** box, type a name for this rule. In our example, we use **bea\_image\_rule**.
5. In the **Definition** box, type the complete text of the rule. Use the following rule syntax:

```
when HTTP_REQUEST {  
  if { [matchclass [HTTP::uri] ends_with $::images] } {  
    pool <image pool name>  
  
  } else {  
    pool <http pool name>  
  }  
}
```

In our example, we type:

```
when HTTP_REQUEST {  
  if { [matchclass [HTTP::uri] ends_with $::images] } {  
    pool bea_images  
  
  } else {  
    pool bea_http_pool  
  }  
}
```

}



**Figure 1.7** iRule to send static content to the images pool

6. Click **Finished**.

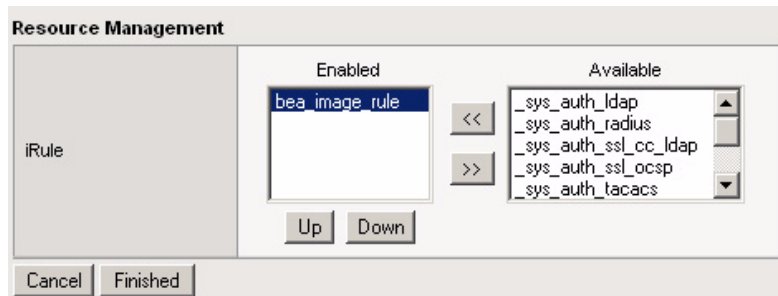
## Changing the virtual server to use the rule

After you have completed the rule, you must the modify the virtual server you created in the *Creating the virtual server* section to use the iRule you just made.

### To change the virtual server to use the rule

1. On the Main tab, expand **Local Traffic**.
2. Click **Virtual Servers**.  
The Virtual Server screen opens.
3. Click the name virtual server you created in the *Creating the virtual server* section. In our example, we click **bea\_http\_vs**.  
The Virtual Server Properties screen opens.
4. On the Menu bar, click **Resources**.  
The Resources screen for the virtual server opens.
5. In the iRules section, click the **Manage** button.  
The Resource Management screen opens.
6. From the **Available** list, select the name of the iRule you created in the preceding procedure, and click the Add (<<) button to move the rule to the **Enabled** list. In our example, we select **bea\_image\_rule**. (see Figure 1.8)

- 
7. Click the **Finished** button.  
The virtual server is now configured to use the iRule.



*Figure 1.8 Associating the iRule with the virtual server*

## 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.