Load Balancing Overview

https://campus.barracuda.com/doc/4259968/

> The Load Balancing feature is available only in the Barracuda Web Application Firewall 460 or higher.

A load balancer is a networking device that distributes traffic across multiple backend servers in order to improve website response times. The Barracuda Web Application Firewall can act as a stand-alone load balancer or work in conjunction with other load balancers. Situated in front of backend servers, it distributes incoming traffic across the servers using the configured algorithm. The Barracuda Web Application Firewall supports load balancing of all types of applications. Load balancing ensures that subsequent requests from the same IP address will be routed to the same backend server as the initial request. This guarantee of persistence requires an awareness of server health so subsequent requests are not routed to a server that is no longer responding. The Barracuda Web Application Firewall can monitor server health by tracking server responses to actual requests and marking the server as out-of-service when errors exceed a user-configured threshold. In addition, the Barracuda Web Application Firewall can perform out-of-band health checks, requests created and sent to a server at configured time intervals to verify its health.

The Barracuda Web Application Firewall includes the following load-balancing features:

- Distributes traffic requests among backend servers according to a user-configured algorithm.
- Automatically identifies server status to ensure appropriate traffic routing.
- Adds and removes servers without interrupting network traffic.
- Provides persistence support that allows a user to maintain connection integrity between client and web service.
- Provides for configuration of a backup server, used only when all other servers being load balanced are out-of-service.

Load balancing can be configured at two levels:

- General (all TCP traffic - Layer 4)
- Content rule (HTTP traffic only - Layer 7)

The general policy is configured for a service and applies to all TCP requests to the service, whereas the content rule policy applies to HTTP requests matching the configured content rule only. The general and content rule configuration procedures are identical. There are three steps to configure load balancing on a Barracuda Web Application Firewall:

- Configure the load balancing algorithm and other general parameters.
- Configure a persistence method to maintain the integrity of stored state information.

- Configure a failover method to handle requests for a server which is down.

General load balancing, routing requests to backend servers based on a user-configured algorithm, is configured for a service. From **BASIC > Services**, choose **Edit** under Options for the service. Choose the algorithm, persistence method, and failover method. The algorithm determines where the first request from a source IP address is routed. Future requests from the same client will be routed to the same server according to the configured persistence method. Failover method applies only when the persisted server is “out-of-service”. For detailed instructions to configure load balancing, see online help.

Monitoring the Health of the Server

Load balancing distributes requests to servers, sending subsequent requests from the same client to the same backend server. To prevent requests from being sent to an unresponsive server, the health of all backend servers must be monitored. The Barracuda Web Application Firewall monitors server health in three ways: by using In-Band, Out-of-Band, and Application Layer health checks. In-Band and Application Layer health checks can only change a server status to out-of-service from an online state, but Out-of-Band health checks, which perform periodic tests of all servers, allow a server state to change from out-of-service to online when the health checks succeed.

In-Band and Application Layer health checks are performed only if the parameter **Enable OOB Health Checks** is set to *Yes* for Out-of-Band health checks. This prevents servers from being marked out-of-service indefinitely. Disabling Out-of-Band health checks disables monitoring server health.

For detailed configuration instructions, see the online help by clicking **Edit** for the server on the **BASIC > Services** page.

In-Band Health Checks

In-Band health checks monitor a server’s connections and response to user traffic. The In-Band health check policy specifies Layer 4 and Layer 7 error thresholds. The server connections and responses are monitored for errors. When error counts exceed configured thresholds, the server is marked out-of-service.

Servers marked out-of-service no longer receive requests. Traffic is routed to other load balanced servers if possible. When no healthy server is available to serve a request, an error response is sent to the client.

In-Band monitoring is enabled by default, and default parameters are provided. The settings can be modified if desired. In-Band monitoring is disabled if Out-of-Band health checks are disabled.

Out-of-Band Health Checks

The Barracuda Web Application Firewall also monitors server health by sending requests at configured intervals that are independent of incoming traffic. Out-of-Band health checks are performed in addition to user traffic connections. The Out-of-Band health check parameters specify Layer 4 and Layer 7 server monitoring.

If a server health check fails, the server is marked as out-of-service. Out-of-service servers continue to be sent data based on the Out-of-Band health check configuration. Therefore, when a health check succeeds, the server's status reverts to in-service. An out-of-service server can only be restored to service by using Out-of-Band health checks because In-Band checks require user traffic to be sent to the server, and user traffic is not sent to an out-of-service server.

A server marked out-of service will revert to in-service as soon as Out-of-Band health checks succeed. These checks are performed at configured intervals (by default 10 seconds).

Application Layer Health Check

An Application Layer health check sends an HTTP request to verify the server is responding correctly. A correct response verifies the server is healthy. Otherwise, the server is marked as out-of-service. The Application Layer health check settings specify the HTTP request type (URL, method, headers), and healthy response (status code, match content string).

Configuring a Backup Server

An optional backup web server can be configured to be used only when all other load balanced servers fail. For detailed instructions refer to online help.

Content Rule

A website can be further partitioned based on content in the HTTP requests by creating a content rule. A content rule is a collection of one or more rules that specify a pattern in the URL or header fields of the request. For requests matching the rule, the configured content rule policies are applied. Content rules allow management of HTTP traffic flow for a web application.

Configuring a content rule requires the following steps:

- Create the content rule for the target web service.
- Add one or more rules to define match criteria for this content rule.
- Configure the policies to apply to matching requests.

You can configure settings for a content rule that apply to three traffic management techniques:

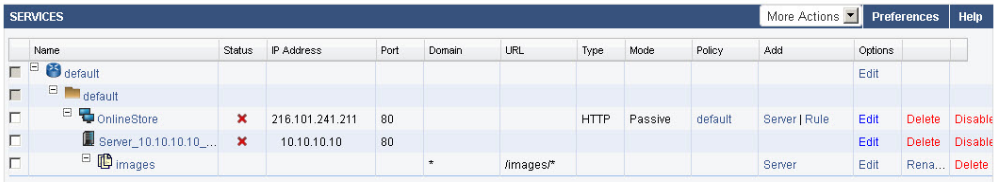
- **Load Balancing** (only in Proxy mode) - Sets the load balancing policy for the content rule. By default, the parent web service's load balancing policy is copied into the content rule. Load balancing is tied to a server group, and the content rule configuration specifies which server group to use. This allows distribution of requests based on the content type.
- **Caching** - Sets caching policy for the content rule (refer to Configuring Caching and Compression). This allows selective caching based on the content type.
- **Compression** - Sets compression policy for the content rule. This improves the response time for clients accessing the web service by compressing web pages with the specific content type.

Rules are evaluated based on a key comprised of the URL, host, and an optional extended match rule in specified sequential order. In most cases, only host and URL are used to specify a rule. The Barracuda Web Application Firewall optimizes the search for the most common case by implementing a parallel search algorithm on all rules. The matching is determined by a best-match algorithm where the best match is the rule with the longest matching host and URL keys. For more information, see Extended Match and Condition Expressions.

Example: Content Rule for Images

Assume that requests to a service are normally served by servers S1, S2, and S3. To direct all requests for image content from the images directory in the web server to a different set of servers (such as S4 and S5), do the following:

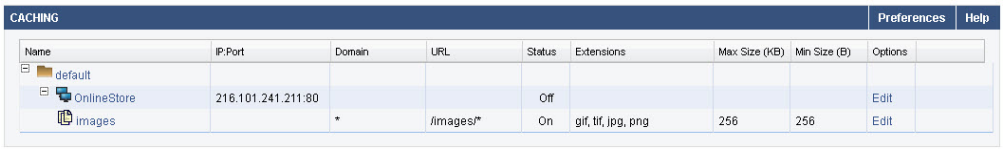
- Use **Rule** for the service on the **BASIC > SERVICES > Services** section to create a content rule for requests matching the URL */images/** as shown below: (For detailed instructions on creating a content rule, refer to online help).



- Add one or more servers for this content rule (S4 and S5, in this case) using the **Server** option

under **Add**. Adding servers for content rules is similar to adding servers for a service. Any future requests matching */images/** will be now directed to one of the servers added to this content rule (S4, S5) instead of being sent to the servers associated with the parent service (S1, S2, S3).
- By default, the load balancing policy of the parent service is inherited by newly added content groups. To customize the load balancing policy used by the servers for this content rule, edit the content rule.

To configure caching for a content rule, create caching rules specifying file extensions and size restrictions for objects that should be cached on the Barracuda Web Application Firewall. These objects will be retrieved from the cache directly to serve future requests, rather than fetching the object content from the backend servers.



Create compression rules for a content rule, specifying what response content should be compressed by the Barracuda Web Application Firewall to improve available network bandwidth. For more information on configuring compression, see Configuring Caching and Compression.

Figures