Servers and Load Balancing

https://campus.barracuda.com/doc/76284378/

Servers

After Barracuda has filtered network and application attacks, it forwards safe traffic to your application servers. Each application can have one or more servers. Your servers receive users' requests, process them, and return the pages requested by the users. For more information on traffic flow, refer to Understanding Traffic Flow with Barracuda WAF-as-a-Service.

Multiple Servers

If you have only one server defined, Barracuda WAF-as-a-Service will forward all your application traffic to that one server. Having more than one server has several advantages:

- If one of your servers suddenly goes offline – due to a hardware or software problem – Barracuda WAF-as-a-Service can forward all traffic to the other servers, keeping your application available without interruption.
- If your traffic load is too much for one server, Barracuda WAF-as-a-Service can distribute the load between multiple servers, making it more manageable and keeping your application performing well for all users.
- If you have multiple servers in different geographical locations, you can send each user to the server geographically nearest to them, increasing the speed of your application for your users.

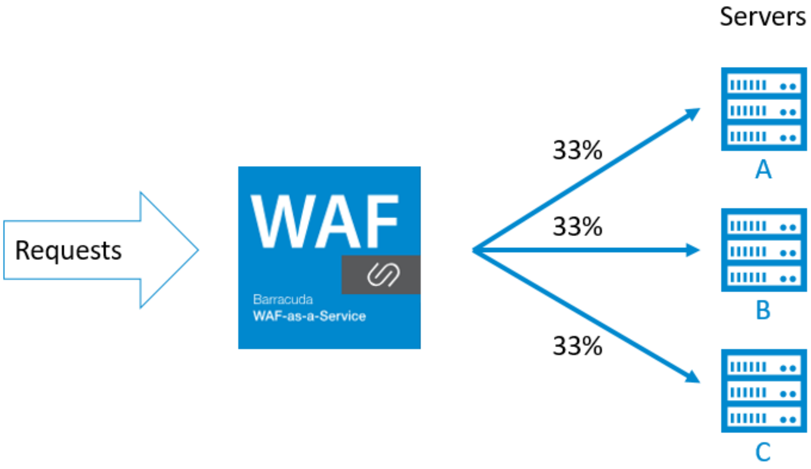
If you define multiple servers, Barracuda WAF-as-a-Service uses Load Balancing to determine which traffic to send to which server.

Load Balancing

Load Balancing is a set of algorithms Barracuda WAF-as-a-Service uses to decide which traffic to send to which server.

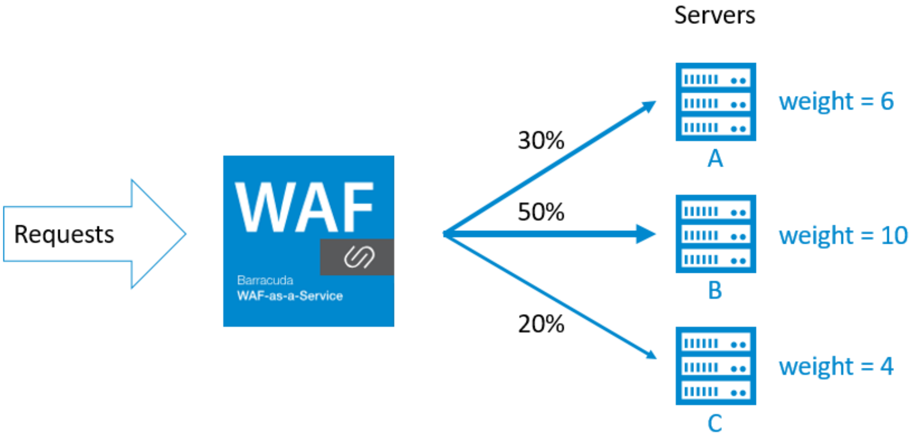
Barracuda WAF-as-a-Service supports the following load balancing algorithms:

- **Round Robin**– Barracuda WAF-as-a-Service sends an equal number of requests to each server. For example, if you define Servers A, B, and C, and 300 requests arrive, approximately 100 requests will be sent to Server A, 100 to Server B, and 100 to Server C. Keep in mind that due to the geographically dispersed nature of Barracuda’s Cloud Scrubbing Centers, the distribution will be approximately equal, but might not be exactly equal.

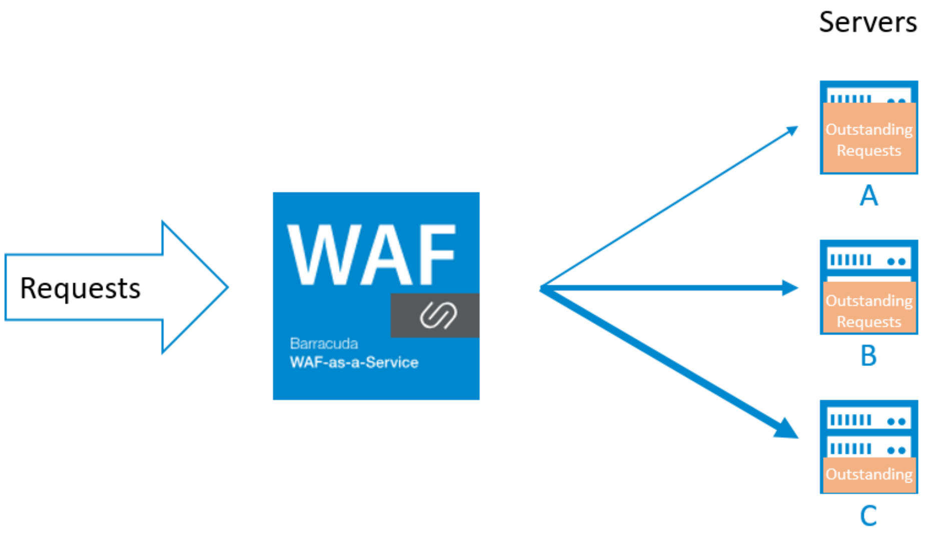


- **Weighted Round Robin**– Barracuda WAF-as-a-Service sends requests to each server, depending on the server's configured weight. For example, the following servers, with their defined weights, will receive appropriate percentages of the requests.

Server	Weight	Approx. Proportion of Requests Sent Based on Weight
A	6	30% (6/20)
B	10	50% (10/20)
C	4	20% (4/20)



- **Least Requests** – Barracuda WAF-as-a-Service will send each request to the server that has the least outstanding requests at the time the new request arrives. In this way, servers that process requests more quickly will receive more requests, and slower servers will receive fewer requests.



- **Geographical** – Barracuda WAF-as-a-Service will send each request to the server that is geographically closest to the Cloud Scrubbing Center that is closest to the requesting user. For example, Server A is located in London, UK, and Server B is located in Boston, USA. Requests from Frankfurt, Germany will be sent to Server A, while requests from Chicago, USA will be sent to Server B.

Persistence Method

After a user has been sent to a particular server, it is sometimes advantageous to have that user continue to be sent to the same server, regardless of the load balancing algorithm. For example, the server that processed the user’s initial request will likely have the user’s information in its cache, allowing it to process future requests from that user more quickly.

Barracuda WAF-as-a-Service supports the following persistence methods:

- **None** – All requests are load balanced according to the algorithm selected. Requests from the same user may be routed to different servers.
- **Source IP** – All requests from a single IP address are always routed to the same server. The first request from an IP is load balanced according to the algorithm selected; future requests are always sent to the same server.
- **Cookie** – All requests bearing a particular cookie are always routed to the same server. Cookie persistence supports two modes:
  - **Cookie Insert** – Barracuda WAF-as-a-Service routes the first request from a user to one of the servers based on the load balancing algorithm. At the same time, it inserts a cookie to identify the client. Subsequent requests from the client include the inserted persistence cookie, identifying the requests so they can be routed to the same server as the first request.

  - **Cookie Passive** – Similar to Cookie Insert, however, Barracuda WAF-as-a-Service does not insert a cookie. Instead, Barracuda WAF-as-a-Service looks for a cookie set by your application servers. If your application servers do not set the cookie indicated, requests from that user continue to be load balanced across all servers.

When defining persistence, you must also specify the Fail Over Method. This determines what happens if a user is assigned to a particular server, but that server then goes out of service or is unreachable. The two Fail Over options are:

- **Load Balance**– (Preferred) The user's requests are load balanced to a new server (and then persisted to the new server according to the Persistence Method selected).
- **Error** – The user receives an error page and cannot proceed. Only use this setting if you absolutely cannot tolerate a user's requests being load balanced to a new server, because doing so will cause the user to be unable to use your application for the time defined in the Persistence Timeout (by default, 10 minutes).

Server Modes

There are two different modes for each server:

- **In Service** – (Default) The server is available to receive traffic.
- **Out of Service** – The server is not available to receive traffic. Existing requests to the server will continue, but new requests will not be sent to the server.
  Only switch a server to Out of Service mode if you want to perform scheduled maintenance on it or if you discover a problem with it.

Server Health

Barracuda WAF-as-a-Service can monitor the health of your servers, and automatically divert traffic away from servers that fail the health check.

Barracuda WAF-as-a-Service performs two types of health checks:

- **Application Layer Health Checks** periodically send a request to your server and ensure that the server responds correctly. You can specify the URL to request from the server, as well as the status code and content you expect to receive back from the server. If the server does not respond back or if the response does not contain the expected status code or content, the system diverts traffic away from that server until it starts responding correctly to this Health Check again.
- **In-Band Health Checks** monitor the performance of your server as it processes normal application traffic. If certain error thresholds are exceeded, the system diverts traffic away from

the server. Traffic is restored to that server once it passes an Application Layer Health Check, described above.

While Barracuda WAF-as-a-Service is diverting traffic away from your server due to a failed health check, you will see "Down" under the "Health" column for that server.

If you disable health checks for a specific server, Barracuda WAF-as-a-Service will send traffic to that server even if it is not operating normally.

Figures