Reminder about HTTP KeepAlive

In early version of the HTTP protocol, clients used to send each request over a new TCP connection to the server, getting content from the server through this connection and finally closing it. This method works well when web pages have a few objects. More objects means more time to wait for each TCP connection setup and close.

- **HTTP 1.0** introduced the header **Connection: Keepalive**: Clients and servers sent each other this header in order to tell the other side to keep the connection opened. *By default, there was no keepalive.*
- **HTTP 1.1** considers every connection to be kept alive. If one doesnt want the connection to stay opened, the client or the server has to send the header: **Connection: Close**.

By using HTTP Keepalive, youre going to reduce the web pages load time.

ALOHA and HTTP KeepAlive

In the **ALOHA GUI**, on the **LB Admin** tab, you can choose the **Connection Mode** in the **HTTP** area. The different options are listed below, with their usage:

- **tunnel**: **Connection** header is not modified, HTTP headers are analyzed for the first request of the connection only, body is ignored. **HTTP KeepAlive** is validated directly between the client and the server, the **ALOHA** will only forward packets in both ways. The following options are set in **HAProxy**'s configuration:
  - no option httpclose
  - no option http-server-close
  - no option forceclose

  Notes:
  - the only cookie based persistence compatible with **tunnel** mode is the cookie insertion method
  - **tunnel** mode hardly works with content switching
  - **tunnel** mode is mandatory if your application requires NTLM authentication

- **passive-close**: **HTTP KeepAlive** disabled on both the client and server side, HTTP headers are analyzed, body is ignored. TCP connection will remain opened and unused if the server or the client does not explicitly close it or until timeout client or server expires. The following options are set in **HAProxy**'s configuration:
  - option httpclose
  - no option http-server-close
  - no option forceclose

  Notes:
  - **passive-close** mode is compatible with all cookie persistence methods
  - **passive-close** works with content switching
  - **passive-close** mode is not compatible with NTLM authentication

- **forced-close**: **HTTP KeepAlive** disabled on both the client and server side, HTTP headers are analyzed, body is ignored. TCP connection will be closed be the **ALOHA**. This option implicitly enables HAProxy's **option httpclose**. The following options are set in **HAProxy**'s configuration:
  - no option httpclose
  - no option http-server-close
  - option forceclose

  Notes:
  - **forced-close** mode is compatible with all cookie persistence methods
  - **forced-close** works with content switching
  - **forced-close** mode is not compatible with NTLM authentication

- **server-close**: **HTTP KeepAlive** allowed on the client side and disabled on the server, HTTP headers are analyzed, body is scanned. The following options are set in **HAProxy**'s configuration:
  - no option httpclose
  - option http-server-close
  - no option forceclose

  Notes:
  - **server-close** mode is compatible with all cookie persistence methods
  - **server-close** works with content switching
  - **server-close** mode is not compatible with NTLM authentication