ALOHA LOAD BALANCER

CHECKING THE HEALTH OF APPLICATION SERVERS

"APPNOTE" #0013 — LIST OF CHECKS IN ALOHA

This application note is intended to list the main checks available in the Aloha solution to be sure of the state of health of real servers and thus to avoid potential malfunctions.

OBJECTIVE

Configure Aloha to control the state of health of application servers in a given server farm.

COMPLEXITY

★☆☆☆☆

VERSIONS CONCERNED

V 3.x and later

CHANGELOG

2012-01-02: Update for ALOHA 5.5 and above

2011-03-28: Initial version

CHECKS AVAILABLE FOR LEVEL 4 LOAD BALANCING

option tcpcheck

```
option tcpcheck  [interval <seconds>] [timeout <seconds>]
                 [source <ip>] [port <int>]
```

Use this test method to check whether a TCP connection is open on the IP addresses of real servers.

With:

- ***interval:*** frequency of checks, in seconds (default: 10 seconds).
- ***timeout:*** period after which a connection attempt not yet completed is considered to have failed (default: 3 seconds).
- ***source:*** forces the IP source which must be used.
- ***port:*** forces the target port (default: the port of the real servers, if it exists).

option httpcheck

```
option httpcheck    [interval <seconds>] [timeout <seconds>] [source <ip>]
                    [port <int>] [uri <uri>] [statuscode <int>]
```

Use this test method to check the performance of an HTTP "GET" request on the IP addresses of real servers.

With:

- ***interval:*** frequency of checks, in seconds (default: 10 seconds).
- ***timeout:*** period after which an attempt with no response from the server is considered to have failed (default: 3 seconds).
- ***source:*** forces the IP source which must be used.
- ***port:*** forces the target port (default: the port of the real servers, if it exists).
- ***uri:*** uri requested (default: the root "/").
- ***statuscode:*** expected status code (default: code 200: OK Request handled successfully).

option arpcheck

```
option arpcheck   [interval <seconds>] [timeout <seconds>]
                  [source <ip>] [iface <name>]
```

Use this test method to send an "arp-whoas" to the IP addresses of real servers.

With:

- ***interval:*** frequency of checks, in seconds (default: 10 seconds).
- ***timeout:*** period after which an attempt with no response from the server is considered to have failed (default: 3 seconds).
- ***source:*** forces the IP source which must be used.
- ***iface:*** name of the network interface.

option icmpcheck

```
option icmpcheck [interval <seconds>] [timeout <seconds>] [source <ip>]
```

Use this test method to send an "icmp echo" to the IP addresses of real servers.

With:

- ***interval:*** frequency of checks, in seconds (default: 10 seconds).
- ***timeout:*** period after which an attempt with no response from the server is considered to have failed (default: 3 seconds).
- ***source:*** forces the IP source which must be used.

CHECKS AVAILABLE FOR LEVEL 7 LOAD BALANCING

IP address and Port

By default, the health check is performed on the server IP address and port configured for the load-balancing. It is possible to overwrite this information with **addr** and **port** parameters:

```
serveur server1 10.0.0.1:443 check port 80
```

In the example above, the load-balancing is done on port 443 but the check is performed on port 80.

Cyphered health check (SSL)

(Aloha 5.5 and above only)

By default, health checks are performed in clear, unless the ALOHA is setup to connect to the server using a cyphered connection.

In some cases, il can be useful to force the cyphering of a check, mainly when load-balancing SSL services in TCP mode combined with a layer 7 health check. This is the role of the **check-ssl** parameter:

```
option httpchk
serveur server1 10.0.0.1:443 check check-ssl
```

check tcp

By default, when no check option is specified, the health check is a TCP connection establishment.

If the **check-ssl** parameter is configured, the check automatically becomes a SSL CONNECT.

option httpcheck

```
option httpchk
option httpchk <uri>
option httpchk <method> <uri>
option httpchk <method> <uri> <version>
```

Use this test method to check the servers' state of health via the HTTP protocol.

If you specify the "httpchk" option, a complete HTTP request is sent once the connection has been established. Responses 2xx or 3xx are considered valid; all others indicate a server failure, including a lack of response.

With:

- ***method:*** an optional HTTP method used for requests. When it is not specified, the "OPTIONS" method is then used, as it requires few server resources and it is easy to filter in logs.

- ***uri:*** uri reference in HTTP requests. The default value is the root "/" which is accessible by default on any server. Any URI is suitable, however, and request strings are authorized.
- ***version:*** an optional string containing the HTTP version. The default value is "HTTP/1.0", but some servers may behave incorrectly when using the HTTP 1.0 protocol. In this case, you can use HTTP/1.1.

IMPORTANT:

The port and interval are specified in the server configuration.

This option does not necessarily require an HTTP backend; it may also run using TCP backends. This distinctive characteristic is useful for checking simple scripts through dedicated ports and using the inetd daemon.

option mysql-check

```
option mysql-check
```

Use this test method to check the servers' "state of health" by initializing a MySQL connection.

This check analyses the packets sent when initializing a connection with a MySQL server. It is a basic but useful test, which does not generate logs on the server. However, it does not check whether the databases are present or consistent, nor does it check access permissions. To do this, you can use an external control with "xinetd", for instance.

option smtpchk

```
option smtpchk
option smtpchk <hello> <domain>
```

Use this test method to check the servers' "state of health" via the SMTP protocol.

With:

- ***hello:*** an optional argument which uses the "HELLO" command. This may be either a "HELO" for SMTP or an "EHLO" for ESMTP.
- ***domain:*** the domain name to present to the server. It can be specified only if the "HELLO" command was specified. "localhost" is used by default.

When you specify "option smtpchk", the health checks will monitor the TCP connections made via an SMTP command. The "HELO localhost" command is used by default. The code returned by the server is then analyzed; if it starts with "2", it will be considered active. All other responses—as well as a lack of response—are considered errors and indicate that the server is not available.

This test is designed to be used with SMTP servers or relays.

option ssl-hello-chk

```
option ssl-hello-chk
```

When certain SSL-based protocols are relayed via HAProxy in TCP mode, you can check whether the server is communicating correctly in SSL, instead of simply checking whether it accepts TCP connections. When the "option ssl-hello-chk" option is set, a "client hello" message in SSLv3 format is sent once the connection is established with the server. The response is analyzed in order to find a corresponding "server hello".

The server is considered active when the server responds correctly to the "hello".