ALOHA Load-Balancer

API objects

Document version:	v2.0
API version concerned:	v2.0
Aloha version concerned:	v4.2.3
Last update date:	24th November, 2011

[Sélectionnez la date]

Sommaire

L7 module objects 5
Farm object 5
balance 5
protocol 5
log 6
log_format 6
http_connection_mode 6
http_pretend_keepalive 6
http_xff_header_insert 7
http_cookie 7
http_cookie_name 7
http_cookie_mode 7
http_cookie_nocache 8
check-interval 8
check_rise 9
check_fall 9
check_port 9
check_timeout 9
adv_check 9
adv_check_http_uri 10
adv_check_http_method 10
queued_timeout 10
connect_timeout 10
connect_retries 10
connect_failure_redispatch 11
connect_source 11
connect_transparent 11
server_inactivity_timeout 11
Server object 12

address 12
port 12
max_connections 12
weight 12
http_cookie_id 13
sorry 13
check 13
maintenance 13
Service object 14
protocol 14
log 14
log_format 14
log_ignore_null 15
http_connection_mode 15
http_pretend_keepalive 15
client_inactivity_timeout 16
http_request_timeout 16
http_keepalive_timeout 16
max_connections 16
default_farm 16
Listener object 17
port 17
address 17
ssl 17
ssl_certificate 17
transparent 18
L4 module objects 18
Farm object 18
balance 18
mode 19

persistence 19
persistance_timeout 19
persistance_netmask 19
service_address 20
service_port 20
service_protocol 20
check-interval 20
check_port 20
check_timeout 21
check_source 21
adv_check 21
adv_check_http_uri 21
adv_check_http_status_code 21
Server object. 22
address 22
port 22
weight 22
sorry 22
check 23

L7 module objects

Farm object

A farm is a real server container. Contained servers are supposed to have the same role and to deliver the same service. Farm attributes and options define session behavior on the real servers side of the load balancer: how to check, dispatch connections, connect, forward data and maintain sessions to real servers.

Farms are identified by a name, allowed chars are alpha numerics dot, dash and underscore.

The available attributes for a farm are listed below.

balance

Type: **enum**

Defines the load balancing algorithm to dispatch new connections on real servers

Values:

- **roundrobin**: Each server is used in turn, according to their weights
- **least-connections**: The new connection is affected to the server with the lowest number of connections
- **hash-uri**: The left part of the URI (before the question mark) is hashed and divided by the total weight of the running servers. The result designates which server will receive the request
- **hash-source**: The source IP address is hashed and divided by the total weight of the running servers to designate which server will receive the request. This ensures that the same client IP address will always reach the same server as long as no server goes down or up

Default: roundrobin is used.

protocol

Type: **enum**

Defines the protocol analyser used by load balander. Some features on HTTP protocol require to set analyser on **http**.

Values:

- **tcp**: Layer4 protocol TCP analyser.
- **http**: Layer7 protocol HTTP analyser.

Default: Layer4 protocol TCP analyser.

log

Type: **enum**

Enable logging on farm events.

Values:

- **enabled**: logs are enabled.

Default: log are disabled.

log_format

Type: **enum**

Defines the log format used if logs are enabled.

Values:

- **tcp**: Advanced tcp log format
- **http**: Advanced http log format (fall back on tcp mode if protocol not set to http)
- **clf**: Use common log format defined by apache (fall back on tcp mode if protocol not set to http)

Default: basic log format.

http_connection_mode

Type: **enum**

Defines http keepalive behavior (ignored if **protocol** is not set to **http**).

Values:

- **tunnel**: Connection header is left untouched and body is ignored.
- **passive-close**: Connection header is changed and body is ignored
- **server-close**: Connection header set, body scanned, client-side keep-alive is made possible regardless of server-side capabilities
- **forced-close**: Connection header set, body scanned, connection closed

Default: Connection header is left untouched and body is ignored.

http_pretend_keepalive

Type: **enum**

Enable load balancer keepalive announce to the server regardless keepalive status (ignored if **protocol** is not set on **http**).

Values:

- **enabled**: keepalive announces are enabled.
- **disabled**: keepalive announces are disabled.

Default: keepalive announce are disabled.

http_xff_header_insert

Type: **enum**

Enable the X-Forwarded-For header insertion to requests filled using client source ip (ignored if **protocol** not set to **http**).

Values:

- **enabled**: X-Forwarded-For header insertion is enabled.

Default: X-Forwarded-For header insertion is disabled.

http_cookie

Enable session affinity using cookie analyse (ignored if **protocol** is not set to **http**).

- **enabled**: cookie analyse is enable (require **http_cookie_name**, **http_cookie_mode** and **http_cookie_nocache** configured)

Default: no cookie affinity.

http_cookie_name

Type: **string**

Defines the name of the cookie used for server affinity. On incoming connections, request is forwarded to the server whose the http_cookie_id server attribut match current cookie value. Settable if **http_cookie** is set. Settable and mandatory if **http_cookie** is set.

Default: no cookie affinity.

http_cookie_mode

Type: **enum**

Defines affinity cookie analyse/insertion behavior. Settable if **http_cookie** is set. Settable and mandatory if **http_cookie** is set.

Values:

- **passive**: cookie is analysed on incoming request to choose server but load balancer does not perform any insertion update or deletion on the cookie or set-cookie
- **passive-silent**: cookie is analysed on incoming request to choose server and set-cookie is removed from response if not needed.
- **reset**: cookie is analysed on incoming request to choose server and set-cookie value is rewrited in response if present.
- **set**: cookie is analysed on incoming request to choose server and set-cookie value is rewrited if present or inserted in response if needed.
- **set-silent**: cookie is analysed on incoming request to choose server and set-cookie value is rewrited if present, inserted in response if needed or removed if not needed.
- **session-prefix**: cookie is analysed on incoming request to choose server whose http_cookie_id prefix match. set-cookie value is prefixed using http_cookie_id server attribute value in response.
- **insert-only-silent**: cookie is analysed on incoming request to choose server and set-cookie value is leaved untouched if present, inserted in response if needed or removed if not needed.
- **passive-session-prefix**: cookie is analysed on incoming request to choose server whose http_cookie_id prefix match.

Default: no cookie affinity.

http_cookie_nocache

Type: **enum**

Enable nocache flag on inserted cookie to prevent proxy caching of the cookie. Settable if **http_cookie** is set. Settable and mandatory if **http_cookie** is set.

Values:

- **enabled**: nocache flag insertion on cookie is enabled.
- **disabled**: nocache flag insertion on cookie is disabled.

Default: no cookie affinity.

check-interval

Type: **integer**

Defines interval in seconds between two consecutive server health checks.

Default: 2.

check_rise

Type: **integer**

Defines count of consecutive successful health checks to consider server operational.

Default: 2.

check_fall

Type: **integer**

Defines count of consecutive unsuccessful health checks to consider server not operational.

Default: 3.

check_port

Type: **integer**

Defines destination TCP port used to perform health check. Set to 0 to use server port.

Default: server port is used.

check_timeout

Type: **integer**

Defines additional health check timeout in seconds, but only after a connection has been already established. Set to 0 for set timeout to **check-interval**.

Default: timeout is set to *check-interval*.

adv_check

Type: **enum**

Defines advanced check method.

Values:

- **http**: Use an http request health check for server testing, If http status code 2xx or 3xx is returned, check is successful (require **adv_check_http_method** and **adv_check_http_uri** configured).

- **ssl**: Use SSLv3 client hello health checks for server testing, If a SSLv3 server hello is retured, check is successful.

Default: no advanced check a tcp connect is preformed.

adv_check_http_uri

Type: **string**

Defines requested URI on health check. Settable and mandatory if *adv_check* is set to *http*.

Default: advanced check not set to **http**.

adv_check_http_method

Type: **string**

Defines HTTP request method used on health check. Default is HEAD. Settable and mandatory if *adv_check* is set to *http*.

Default: advanced check not set to *http*.

queued_timeout

Type: **integer**

Defines the maximum time to wait in the queue for a connection slot to be free. Set to 0 for no timeout.

Default: no timeout.

connect_timeout

Type: **integer**

Defines the maximum time to wait for a connection attempt to a server to succeed. Set to 0 for no timeout.

Default: no timeout.

connect_retries

Type: **integer**

Defines the number of retries to perform on a server after a connection failure. Set to 0 for no retries.

Default: no retries.

connect_failure_redispatch

Type: **enum**

Enabled session redispatch to an other server in case of connection failure.

Values:

- **enabled**: session redipatch is enabled.
- **disabled**: session redipatch is disabled.

Default: no redispatch.

connect_source

Type: **ip**

Force source address to use to connect to servers (require **connect_transparent** configured). Set to 0.0.0.0 to use first address on load balancer outgoing interface.

Default: first address on load balancer outgoing interface is used.

connect_transparent

Type: **enum**

Enable transparent proxy mode: client source address is re-used as source address to connect to server. Settable and mandatory if **connect_source** is set.

Values:

- **enabled**: transparend proxy mode is enabled.
- **disabled**: transparend proxy mode is disabled.

Default: first address on load balancer outgoing interface is used.

server_inactivity_timeout

Type: **integer**

Defines the maximum inactivity time on the server side. Set to 0 for no timeout.

Default: no timeout.

Server object

This is the real server, it delivers the same service than other real servers of the same farm.

Server attributs and options define the way and the conditions to forward requests to this real server and the capacity of the server to process requests comparativly to other server.

Servers are identified by a name and the name of the containing farm, allowed chars are alpha numerics dot, dash and underscore.

address

Type: **ip**

Defines the ip address to forward requests to this server.

Mandatory, no default available.

port

Type: **port**

Defines the port to forward requests to this server Set to 0: incoming connection destination port is reused.

Default: incoming connection destination port is reused.

max_connections

Type: **integer**

Specifies the maximal number of concurrent connections that will be sent to this server. If the number of incoming concurrent requests goes higher than this value, they will be queued, waiting for a connection to be released. This parameter is very important as it can save fragile servers from going down under extreme loads.

Default: unlimited.

weight

Type: **integer**

Used to adjust the server's weight relative to other servers of the farm. All servers will receive a load proportional to their weight relative to the sum of all weights, so the higher the weight, the higher the load. The default weight is 1, and the maximal value is 256. A value of 0 means the server will not participate in load-balancing but will still accept persistent connections. If this parameter is used to distribute the load according to server's capacity, it is recommended to start with values which can both grow and shrink, for instance between 10 and 100 to leave enough room above and below for later adjustments.

Default: 1.

http_cookie_id

string

Defines the identifier used as cookie value to identify this server.

sorry

Type: **enum**

A sorry server is only used when all none sorry servers are not operational (health checks).

Values:

- **enabled**: current server is a sorry server.

Default: server is not a sorry server.

check

Type: **enum**

Enable health checks on current server to determine if this server is operational.

Values:

- **enabled**: current server is checked.

Default: server is not checked.

maintenance

Type: **enum**

This server is disabled for maintenance purpose.

Values:

- **enabled**: current server is disabled.

Default: server is not disabled.

Service object

A Layer 7 service is a set of listener.

Service attributs and options define session behavior on the client side and into the the load balancer: max accepted simultaneous sessions, choice of server farm, timeouts on the client side.

Service are identified by a name, allowed chars are alpha numerics dot, dash and underscore.

protocol

enum

Defines the protocol analyser used by load balander. Some features on HTTP protocol require to set analyser on http.

Values:

- **tcp**: Layer4 protocol TCP
- **http**: Layer7 protocol HTTP

Default: Layer4 protocol TCP.

log

Type: **enum**

Enable logging on farm events.

Values:

- **enabled**: log are enabled.

Default: logs are disabled.

log_format

Type: **enum**

Defines the log format used if logs are enabled.

Values:

- **tcp**: Advanced tcp log format

- **http**: Advanced http log format (fall back on tcp mode if protocol not set to http)
- **clf**: Use common log format defined by apache (fall back on tcp mode if protocol not set to http)

Default: basic log format.

log_ignore_null

enum

Enabled: no log for open/closed empty connections.

Values:

- **enabled**: no log for connections are ignored
- **disabled**: no log for connections are not ignored

Default: log empty connections .

http_connection_mode

Type: **enum**

Defines http keepalive behavior (ignored if **protocol** is not set to **http**).

Values:

- **tunnel**: Connection header is left untouched and body is ignored.
- **passive-close**: Connection header is changed and body is ignored
- **server-close**: Connection header set, body scanned, client-side keep-alive is made possible regardless of server-side capabilities
- **forced-close**: Connection header set, body scanned, connection closed

Default: Connection header is left untouched and body is ignored.

http_pretend_keepalive

Type: **enum**

Enable load balancer keepalive announce to the server regardless keepalive status (ignored if **protocol** is not set on **http**).

Values:

- **enabled**: keepalive announces are enabled.
- **disabled**: keepalive announces are disabled.

Default: keepalive announce are disabled.

client_inactivity_timeout

integer

Defines the maximum inactivity time on the client side. Set to 0 for no timeout.

Default: no timeout.

http_request_timeout

integer

Defines the maximum time to wait an entire valid HTTP request (no effect if *protocol* is not set to *http*). Set to 0 for no timeout..

Default: no timeout.

http_keepalive_timeout

integer

Defines the maximum time to wait for a new HTTP request to appear in the same session (no effect if *protocol* is not set to *http*). Set to 0 for no timeout.

Default: no timeout.

max_connections

integer

Defines the maximum of simultaneous sessions accepted on this service.

Default: no limit.

default_farm

integer

Defines the default farm used to process requests

Default: if no rules match connections are closed.

Listener object

A listener is part of a service. it defines a port or a couple ip port to listen for incoming tcp connections.

Listener are identified by a name and the name of the containing service, allowed chars are alpha numerics dot, dash and underscore.

port

type: **integer**

Defines the port to listen.

Mandatory, No default available.

address

Type: **ip**

Defines the address to listen. Use 0.0.0.0 for all ipv4 adresses present on load balancer network configuration.

Mandatory, No default available.

ssl

enum

Enable ssl offloading.

Values:

- **enabled**: ssl offloading is enabled.

Default: ssl offloading is disabled.

ssl_certificate

string

Defines the certificate used for ssl offloading. Settable and mandatory if *ssl* is enabled.

Default: ssl offloading is disabled.

transparent

enum

Enabled: incoming connections to listener ip and port are intercept even if ip is not present in load balancer network configuration of the load balancer (ie a router forward connections to the load balancer).

Values:

- **enabled**: transparent accept is enabled.

Default: intercept only connections to listener ip and port present in load balancer network configuration.

L4 module objects

Farm object

A farm is a real server containers. Contained servers are supposed to have the same role and to deliver the same service. Farm attributs and options defines session behavior on the real servers side of the load balancer: how to check real servers, dispatch connections on real servers, forward packets to real servers, maintain session to real servers.

Farms are identified by a name, allowed chars are alpha numerics dot, dash and underscore.

balance

Type: **enum**

Defines the load balancing algorithm to dispatch new connections on real servers

Values:

- **roundrobin**: distributes jobs equally amongst the available real servers, jobs are assigned to real servers proportionally to there real servers' weight.
- **least-connections**: assigns more jobs to servers with fewer jobs
  and relative to the real servers' weight. This is the default.
- **hash-source**: assigns jobs to servers through looking up a statically assigned hash table by their source IP addresses.
- **hash-destination**: assigns jobs to servers through looking up a statically assigned hash table by their destination IP addresses
- **shortest-expected-delay**: assigns an incoming job to the server with the shortest expected delay.
- **never-queue**: assigns an incoming job to an idle server if there is, instead of waiting for a fast one; if all the servers are busy, it adopts the Shortest Expected Delay policy to assign the job.

Default: roundrobin

mode

Type: **enum**

Values:

- **gateway**: Use direct routing.
- **nat**: Use destination masquerading (network access translation, or NAT).
- **tunnel**: Use ipip encapsulation (tunneling).

Default: nat

persistence

Type: **enum**

If this option is enabled, multiple requests from a client (same source ip) are redirected to the same real server selected for the first request. This option may be used in conjunction with protocols such as SSL or FTP where it is important that clients consistently connect with the same real server.

- **enabled**: persistence is enabled.

Default: persistence is disabled.

persistance_timeout

Type: **integer**

Value in seconds specify how long sessions are kept persistence load balance table.

```
    persistence <timeout>
```

Default: 300 seconds.

persistance_netmask

Type: **ip**

Specify the granularity of persistance

Default: 255.255.255.255

service_address

Type: **ip**

Defines the service address to intercept to be served by current farm.

Default: no interception.

service_port

Type: **port**

Defines the service port to intercept to be served by current farm. Settable if *service_address* is configured. Set to 0 for intercept all ports.

Default: if *service_address* is set all ports are intercepted.

service_protocol

Type: **enum**

Defines the service protocol (udp or tcp) to intercept to be served by current farm.

Values: * **tcp**: intercept tcp. * **udp**: intercept udp.

Default: if *service_address* is set tcp is intercept.

check-interval

Type: **integer**

Defines interval between two consecutive server health checks in seconds.

Default: 10.

check_port

Type: **integer**

Defines destination TCP port used to perform health check.

Default: server port is used.

check_timeout

optional

Defines additional health check timeout, but only after a connection has been already established. If not set default is check interval.

```
    check_timeout <seconds>
```

Default: *checkinterval*/2

check_source

Type: **integer**

Defines source address used to perform health check.

Default: first load balancer interface address is used.

adv_check

Type: **enum**

Defines advanced check method.

Values:

- **http**: Use an http request health checks for server testing, If http status code 200 is returned, check is successful.
- **tcp**: Perform a tcp connect. if connect accepted, check is successful.
- **icmp**: process an icmp echo to server, if echo reply received, check is successful.

Default: icmp

adv_check_http_uri

Type: **string**

Defines requested URI on health check. Default is / (ignored if *adv_check* not set to *http*).

Default: /

adv_check_http_status_code

Type: **intger** (ignored if *adv_check* is not set to *http*)

Defines the HTTP status code value to consider check successful (ignored if *adv_check* not set to *http*).

Default: 200.

Server object.

This is the real server, it deliver the same service than other real servers of the same farm.

Server attributs and options define the way and the conditions to forward packets to this real server and the capacity of the server to process requests comparativly to other server.

Servers are identified by a name and the name of the containing farm, allowed chars are alpha numerics dot, dash and underscore.

address

Type: **ip**

Defines the ip address to forward requests to this server. *Mandatory*, no default.

port

Type: **port**

Defines the port to forward requests to this server in mode *nat*. or/and port to perform tcp checks if *checkport* not configured. Set to 0 and incoming packets destination port is used.

Default: incoming packets destination port is used.

weight

Type: **integer**

jobs are assigned to real servers proportionally to their real servers' weight. Servers with higher weight receive new jobs first and get more jobs than servers with lower weights. Servers with same weight get an equal distribution of new jobs.

Default: 1

sorry

Type **enum**

If enabled, server is only used when all none sorry servers are not operational (health checks). Only one sorry server is available per L4 farm. A sorry server could not be checked (heath check).

- **enabled**: server is a sorry server.

Default: server is not a sorry server.

check

Type **enum**

If enabled, health checks are processed to determine if this server is operational.

- **enabled**: server is checked.

Default: server is not checked.