HAProxy® Configuration Reference Card

2020-03-02 - Use under Creative Commons License (CC BY-SA)
For version 2.2-dev3

General configuration format (Configuration Manual §2)

As of version 2.2-dev3, configuration lines are words separated by spaces except if preceded by a backslash or enclosed within single or double-quote. Environment variables (${VAR} or $VAR) are resolved within double-quote. Sharp (#) outside quotes mark the end of the line. Backslash (\) before any delimiter outside single quote copy the next character verbatim..

```
global
  [global directives]
defaults
  [options to next defaults section]
frontend <FE_NAME>                # front half of a proxy
  [<frontend-options>]
  bind [<address>]:<port_range>[,...] [<bind-params>]*
  mode {tcp|http|heath}
  acl <aclname> <criterion> [<flags>] [<op>] <value> ...
  tcp-request connection <action> [{if|unless} ...]
  tcp-request session <action> [{if|unless} <condition>]
  tcp-request inspect-delay <timeout>
  tcp-request content <action> [{if|unless} <condition>]
  http-request <action> [options…] [{if|unless} <condition>]
  use_backend <backend> [{if|unless} <condition>]
  tcp-response content <action> [{if|unless} <condition>]
  http-response <action> [options…] [{if|unless} <condition>]
backend <BE_NAME>                 # backend half of a proxy
  [<backend-options>]
  mode {tcp|http|heath}
  acl <aclname> <criterion> [<flags>] [<op>] <value> ...
  stick-table [<sticktable-params>]*
  tcp-request content <action> [{if|unless} <condition>]
  http-request <action> [options…] [{if|unless} <condition>]
  use-server <server> [{if|unless} <condition>]
  server [<address>] [<server-params>]*
  tcp-response inspect-delay <timeout>
  tcp-response content <action> [{if|unless} <condition>]
  http-response <action> [options…] [{if|unless} <condition>]
  http-after-response <action> [options...] [{if|unless} ...]
  stick match <pattern> [table <table>] [{if|unless} <condition>]
  stick on <pattern> [table <table>] [{if|unless} <condition>]
  stick store-request <pattern> [table <table>] [{if|unless} ...]
  stick store-response <pattern> [table <table>] [{if|unless} ...]
listen <NAME>                     Front + backend of a proxy
  [<frontend-options>|<backend-options>]
mailers <NAME>                    Notification mailers definition
  [<mailers-options]              Configuration Manual §3.6
peers <NAME>                      Peers definition to synchronise stick tables
  [<peers-options>]
program <name>                    External program definition
  [<program-options]              Configuration Manual §3.7
resolver <NAME>                   Dynamic DNS
  [<resolver-options]
userlist <NAME>                   Username, password and group for auth
  group <groupname> [users <user>, ...]
  user <username> {password|insecure-password} <password> ...
http-errors <NAME>
  errorfile <code> <filename>
```

Global options (Configuration Manual §3)

```
<address>: ipv4, ipv6, dgram-socket, fd@<number>, stdout, stderr
ca-base <dir>
chroot <jail-dir>
crt-base <dir>
daemon
quiet
description <text>
gid|group <number|groupname>
hard-stop-after <time>
log <address> [log-params]*
lua-load <file>
maxconn <number>
nbproc|nbthread <number>
pidfile <pidfile>
stats socket [<address:port>|<path>] [<bind-params>]*
tune.bufsize <number>             (default: 16384)
uid|user <number>|<username>
ulimit-n <number>
```

Bind (<bind-params>) (Configuration Manual §5.1)

Bind can be multiple with specific process directive

```
<process-set>: all, odd, event, num.start[-[num.end]]
<thread-set>: all, odd, event, num.start[-[num.end]]
<cpu-set>: all, odd, event, num.start[-[num.end]]
alpn <protocols>
ciphers <ciphers>
crt <pem-file>                    Certs followed by keys in PEM
crt-list <filename>               By SNI certs file
defer-accept
expose-fd listeners
{force|no}-sslv3
{force|no}-tls{v10,v11,v12,v13}
interface <interface>
name <name>
process <process-set>[/<thread-set>]
ssl | transparent | v4v6 | v6only
verify [none|optional|required]
```

Server (<server-params>) (Configuration Manual §5.2)

Server can be multiple

```
addr <address>
alpn <protocols>
[no-]backup
ca-file <ca-pemfile>
[no-]check
check-alpn <protocols> / check-sni <sni>
[no-]check-ssl
cookie <value>
enabled|disabled
error-limit <count>               (default: 10)
fall <count>                      (default: 3)
inter <delay> / downinter <delay> / fastinter <delay>
maxconn <number> / maxqueue <number>
port <port>
resolvers <id>
slowstart <delay>
source <address> usesrc <usesrc>
  usesrc: {<addr>[:port]|clientip|client|hdr_ip(<hdr>[,occ])}
[no-]ssl
[non-]stick                       Never/Add connections to stick-table
track [<proxy>/]<server>
weight <weight>
```

Proxies (Configuration Manual §4)

D: default, F: frontend, L: listen, B: backend

```
<balance-algorithm>: roundrobin, static-rr, leastconn, first,
  source, uri, url_param, hdr(<name>), random[(<draws>)], ...
<comp-algorithm>: identity, gzip, deflate, raw-deflate
<match>: status, rstatus, string, rstring
<var-name>: {proc|sess|txn|req|res}.<word>
balance <balance-algorithm> [<arguments>]          [DLB]
capture cookie <name> len <length>                 [FL]
capture request header <name> len <length>         [FL]
capture response header <name> len <length>        [FL]
compression algo <comp-algorithm>...               [DFLB]
compression offload                                [DFLB]
compression type <mime-type>...                    [DFLB]
cookie <name> [<cookie-params>]*                   [DFLB]
  [rewrite|insert|prefix] [domain <domain>]*
  [dynamic] [httponly] [indirect] [nocache] [preserve]
  [maxidle <idle>] [maxlife <life>] [postonly] [secure]
default_backend <backend-name>                     [DFL]
default-server [<server-params>*]                  [DLB]
email-alert ...                                    [DFLB]
errorfile <code> <file>                            [DFLB]
errorfiles <name> [<code> ...]                     [DFLB]
hash-type <method> <function> [avalanche]          [DLB]
  <method>: map-based, consistent
  <function>: sdbm, djb2, wt6, crc32
http-check expect [!] <match> <pattern>            [LB]
http-request | http-response  ...                  [FLB]
http-after-response ...                            [FLB]
[no] log global                                    [DFLB]
[no] log <address> [log-options]*                  [DFLB]
[no] option dontlog-normal                         [DFL]
[no] option dontlognull                            [DFL]
[no] option http-keep-alive                        [DFLB]
[no] option http-no-delay                          [DFLB]
[no] option http-pretend-keepalive                 [DLB]
[no] option http-server-close                      [DFLB]
[no] option httpclose                              [DFLB]
[no] option logasap                                [DFL]
[no] option persist                                [DLB]
[no] option redispatch [<interval>]                [DLB]
[no] option transparent                            [DLB]
option forwardfor [except <network>] [header <name>] [if-none]
option httpchk <method> <uri> [<version>]          [DLB]
option httplog [clf]                               [DFL]
option tcp-check                                   [DLB]
option tcplog                                      [DFLB]
server ...                                         [LB]
stats ...                                          [DFLB]
stick ...                                          [FLB]
stick-table ...                                    [FLB]
tcp-check ...                                      [LB]
tcp-response|tcp-request ...                       [FLB]
timeout check <timeout>                            [DLB]
timeout {client|client-fin} <timeout>              [DFL]
timeout connect <timeout>                          [DLB]
timeout http-keep-alive <timeout>                  [DFLB]
timeout http-request <timeout>                     [DFLB]
timeout queue <timeout>                            [DLB]
timeout {server|server-fin} <timeout>              [DLB]
timeout tarpit <timeout>                           [DFLB]
timeout tunnel <timeout>                           [DLB]
use_backend <backend>                              [FL]
use-server ...                                     [LB]
```

Stick Table (<sticktable-params>) (Configuration Manual §4.2)

stick-table type <type> [len <length>] size <size> [expire <expire>] [nopurge] [peers <peersect> [store <data-type>]

<type>: ip, ipv6, integer, string, binary

<data-type>:

server_id	
gpc0 / gpc0_rate(<period>)	1er Gen. Purpose Counter
gpc1 / gpc1_rate(<period>)	2nd Gen. Purpose Counter
conn_cnt / conn_cur / conn_rate(<period>)	Connections
sess_cnt(<period>) / sess_rate(<period>)	Sessions
http_req_cnt / http_req_rate(<period>)	HTTP Requests
http_err_cnt / http_err_rate(<period>)	HTTP Errors
bytes_in_cnt / bytes_in_rate(<period>)	Bytes In
bytes_out_cnt / bytes_out_rate(<period>)	Bytes Out

Peers (Configuration Manual §3.5)

```
bind [<address>]:<port_range>
peer <peername> <ip>:<port> [<params>*]
server <peername> [<ip>:<port>] [<params>*]
table <tablename> type <type> size <size> [...]
  <type>: ip, integer, string [len <len>], binary [len <len>]
  ...: [nopurge] [store <data-type>] [expire <expire>]
```

ACL (Configuration Manual §7)

```
acl <aclname> <criterion> [<flags>] [<operator>] [<value>] …
<condition>: [!]<acl> [[!]<acl>]* [or [!]<acl> [[!]<acl>]*]
```

<acl> are explicitly declared or anonymous with <SP>{<SP><expr><SP>}<SP> form.

<acl> extract a sample, apply modifications on it and compare to a set of constant patterns using a specific matching method.

<type>: boolean, integer, ip, string, binary

<flags>:

-i	ignore case for subseq -f entries
-f <file>	load from file
-m <meth>	pattern meth. (found, bool, int, ip, bin, len, str, sub, reg, beg, end, dir, dom)
-n	forbid the DNS resolutions
-M	map file defined with -f
-u <id>	force unique id of ACL
--	end of flags

<operator>: eq, ge, gt, le, lt

Warning : regex and case insensitive are linearly evaluated !

Actions (Configuration Manual §4 / http-request / http-response)

Q: request, R: response, S: stats

```
{add|del}-header <name> <fmt>                          [QR]
allow | deny [deny_status <status>]                    [QRS]
auth [realm <realm>]                                   [QS]
capture <sample> [ len <length> | id <slot> ]          [Q]
redirect <rule> | silent-drop                          [QR]
reject | tarpit                                       [Q]
replace-header <name> <match-regex> <replace-fmt>      [QR]
replace-uri <match-regex> <replace-fmt>                [Q]
replace-value <name> <match-regex> <replace-fmt>       [QR]
return [status <code>] [content-type <type>]           [QR]
  [{default-errorfiles|errorfile <file>|errorfiles <name>|
    file <file>|lf-file <file>|string <str>|lf-string <fmt>}]
    [hdr <name> <fmt>]*
set-header <name> <fmt>                                [QR]
{set-uri|set-path|set-query} <fmt>                     [Q]
set-status <status> [reason <string>]                  [R]
```

Internal samples (Configuration Manuel §7.3.2)

```
avg_queue([<backend>])
{be_conn|fe_conn}([<backend>])
{be_conn_free|fe_conn_free}([<backend>])
{be_sess_rate|fe_sess_rate}([<backend>])
bin(<hex>) / bool(<bool>) / int(<integer>)
ipv4(<ipv4>) / ipv6(<ipv6>) / str(<string>)
connslots([<backend>])
cpu_calls                    number of calls by request
cpu_ns_avg|cpu_ns_tot        avg|total nanosecs spent in req.
date([<offset>],[{s|ms|us}])
fe_req_rate([<frontend>])
meth(<method>)
nbsrv([<backend>]) / queue([<backend>])
rand([<range>]) / uuid([<version>])
{srv_conn|srv_conn_free}([<backend>/]<server>)
srv_is_up([<backend>/]<server>)
srv_queue([<backend>/]<server>)
srv_sess_rate([<backend>/]<server>)
{table_avl|table_cnt}([<table>])
var(<var-name>) / env(<name>)
```

L4 samples (Configuration Manual §7.3.3)

```
be_id / be_name / fe_name / srv_name
src / dst / dst_port
src_is_local / dst_is_local
```

L5 samples (Configuration Manual §7.3.4)

```
ssl_bc / ssl_fc
ssl_c_i_dn([<entry>[,<occ>[,{rfc2253}]]])
ssl_c_s_dn([<entry>[,<occ>[,{rfc2253}]]])
ssl_c_verify
ssl_f_i_dn([<entry>[,<occ>[,{rfc2253}]]])
ssl_f_s_dn([<entry>[,<occ>[,{rfc2253}]]])
ssl_fc_sni
```

L6 samples (Configuration Manual §7.3.5)

```
req.proto_http
wait_end                     predefined: WAIT
```

L7 samples (Configuration Manual §7.3.6)

```
capture.req.hdr(<idx>)
{req|res}.fhdr(<name>[,<occ>])
[req|res}.hdr([<name>[,<occ>]])
method / url / path / query / req.ver / status
http_auth(<userlist>)
urlp([<name>[,<delim>]])
```

Sample converters (Configuration Manual §7.3.1)

Without args brackets can be avoid

```
{add|div|mod|mul|sub}(<value>)
{and|or|xor}(<value>)
{b64dec|base64}()
{bool|neg|not}()
bytes(<offset>[,<length>])
concat([<start>],[<var>],[<end>])
{even|odd}()
field(<index>,<delimiters>[,<count>])
hex()
json([{ascii|utf8|utf8s|utf8p|utf8ps}])
length()
{lower|upper}()
map_<method>_<type>(<map-file>[,<default-value>])
  <method>: str,beg,sub,dir,dom,end,reg,int
  <type>: str, int, ip
nbsrv()
regsub(<regex>,<subst>[,{i|g}*])
set-var(<var-name>)
strcmp(<var>)
table_<data-type>(<table>)   See Stick Tables
```

Resolvers (Configuration Manual §5.3.2)

```
accept_payload_size <size>                    (default: 512)
nameserver <id> <address>:<port>
parse-resolv-conf
hold valid <period>                           (default: 10s)
hold obsolete <period>                        (default: 0s)
host {nx|refused|timeout|other} <period>      (default: 30s)
resolve_retries <count>                       (default: 3)
timeout {resolve|retry} <timeout>             (default: 1s)
```

Logs Options (log-format <string>)

```
[no] log <addr> [len <length>] [format <format>] [sample
<ranges>:<smp_size>] <facility> [<level> [<minlevel>]]
<addr>: {<ip4>|<ip6>}:<port>|<sock>|fd@<n>|stderr|stdout|ring@<n>
<format>: rfc3164|rfc5424|short|raw
<facility>: facility to send with
<level>: filter outgoing message
log-format <format>
```

<format>: %[<obrace><FLAGS><cbrace>]<option>... (ex: %{+Q,+X}ci)

%[<expr>]	samples & converters (ex: %[req.hdr(user-agent)])
%ac	actconn
%b %f %s %ft: backend/frontend/server's name/frontend suffixed by '~" for SSL	
%bc %fc %sc: backend/frontend/server's concurrent connections	
%bi %bp %ci %cp %fi %fp %si %sp: backend/client/frontend/server IP/port	
%Ta %Tc %Th %Ti %Tq %Tr %Tt %Tw: timers (active, connect, handshake, idle, request, response, total, wait-queue)	
%B %U	bytes read/uploaded (from S->C / C->S)
%ci %cp	client IP/port (accepted address)
%CC %CS	captured request/response cookie
%ST	status_code
%t	date-time (with millisecond resolution)
%Td	Td = Tt - (Tq + Tw + Tc + Tr)
%tr	date-time of HTTP request
%ts %tsc	termination_state / with cookie status
%Ts	timestamp
%TR	time to receive the full request from 1st byte
%rc	retries
%ms	accept date milliseconds (left-padded with 0)
%sq %bq	server / backend queue
%hr %hs	captured request / response headers default style (+l for CLF style)
%r	http_request (%H… for details)
%sslc	ssl_ciphers (ex: AES-SHA)
%sslv	ssl_version (ex: TLSv1)
%H	hostname
%pid	PID
%ID	unique-id
%T %Tl	gmt / local date-time
%trg %trl	gmt / local date-time of start of HTTP request
%rt %lc	request counter (HTTP req or TCP session) / frontend log counter