



HAProxy® Troubleshooting Reference Card

2020-03-02 - Use under Creative Commons License (CC BY-SA)

For version 2.2-dev3

Stats field names (Management Guide §9)

show stat typed desc

act	[BS]	number of active servers (backend), server is active (server)
addr	[LS]	address:port or "unix".
agent_code	[S]	numeric code reported by agent if any (unused for now)
agent_desc	[S]	short human-readable description of agent_status
agent_duration	[S]	time in ms taken to finish last check
agent_fall	[S]	agent's "fall" parameter, normally 1
agent_health	[S]	agent's health parameter, between 0 and rise+fall-1
agent_rise	[S]	agent's "rise" parameter, normally 1
agent_status	[S]	status of last agent check
algo	[B]	load balancing algorithm
bck	[BS]	number of backup servers (B), server is backup (S)
bin	[LFBS]	bytes in
bout	[LFBS]	bytes out
btot	[BS]	total number of times a server was selected
cache_hits	[FB]	cumulative number of cache hits
cache_lookups	[LB]	cumulative number of cache lookups
check_code	[S]	layer5-7 code, if available
check_desc	[S]	short human-readable description of check_status
check_duration	[S]	time in ms took to finish last health check
check_fall	[S]	server's "fall" parameter used by checks
check_health	[S]	server's health check value between 0 and rise+fall-1
check_rise	[S]	server's "rise" parameter used by checks
check_status	[S]	status of last health check
chkdown	[BS]	number of UP->DOWN transitions
chkfail	[BS]	number of failed checks
cli_abrt	[BS]	number of data transfers aborted by the client
comp_byp	[FB]	bytes that bypassed the HTTP compressor
comp_in	[FB]	number of HTTP response bytes fed to the compressor
comp_out	[FB]	number of HTTP response bytes emitted by the compressor
comp_rsp	[BS]	number of HTTP responses that were compressed
conn_rate	[F]	number of connections over the last elapsed second
conn_rate_max	[F]	highest known conn_rate
conn_tot	[F]	cumulative number of connections
connect	[BS]	cumulative number of connection establishment attempts
cookie	[BS]	server's cookie value or backend's cookie name
ctime	[BS]	average connect time in ms over the 1024 last requests
ctime_max	[BS]	the maximum observed connect time in ms
dcon	[LF]	requests denied by "tcp-request connection" rules
downtime	[BS]	total downtime in seconds
dreq	[LFBS]	requests denied because of security concerns
dresp	[LFBS]	responses denied because of security concerns
dses	[LF]	requests denied by "tcp-request session" rules
econ	[BS]	errors while trying to connect to a backend server
ereq	[LF]	request errors
eresp	[BS]	response errors. srv_abrt will be counted here also
hanafail	[S]	failed health checks details
hrsp_other	[FBS]	http responses with other codes (protocol error)
hrsp_Yxx	[FBS]	http responses with Yxx code (Y: [1, 2, 3, 4, 5])
iid	[LFBS]	unique proxy id
intercepted	[FB]	cum. number of intercepted requests (monitor, stats)
last_agt	[BS]	last agent check contents or textual error
last_chk	[S]	last health check contents or textual error
lastchg	[BS]	number of seconds since the last UP->DOWN transition
lastsess	[S]	seconds since last session assigned to server/backend
mode	[LFBS]	proxy mode (tcp, http, health, unknown)
pid	[LFBS]	process id (0 for first instance, 1 for second, ...)
pxname	[LFBS]	proxy name
qcur	[BS]	current queued requests
qlimit	[S]	configured maxqueue for the server
qmax	[BS]	max value of qcur
qtime	[BS]	average queue time in ms over the 1024 last requests
qtime_max	[BS]	the maximum observed queue time in ms
rate	[FBS]	number of sessions per second over last elapsed second
rate_lim	[F]	configured limit on new sessions per second

rate_max	[FBS]	max number of new sessions per second
req_rate	[F]	HTTP requests per second over last elapsed second
req_rate_max	[F]	max number of HTTP requests per second observed
req_tot	[FB]	total number of HTTP requests received
reuse	[BS]	cumulative number of connection reuses
rtime	[BS]	average response time in ms over the 1024 last requests
rtime_max	[BS]	the maximum observed response time in ms (0 for TCP)
scur	[LFBS]	current sessions
sid	[LS]	server id (unique inside a proxy)
slim	[LFBS]	configured session limit
smax	[LFBS]	max sessions
src_ilim	[S]	limit on the number of available idle connections
srv_abrt	[BS]	number of data transfers aborted by the server
srv_icur	[S]	current number of idle connections available for reuse
status	[LFBS]	status (UP/DOWN/NOLB/MAINT/MAINT)
stot	[LFBS]	cumulative number of sessions
svname	[LFBS]	service name
throttle	[S]	current throttle percentage for the active server
tracked	[S]	id of proxy/server if tracking is enabled
ttime	[S]	avg total session time in ms over the 1024 last requests
ttime_max	[BS]	the maximum observed total session time in ms
type	[LFBS]	0=frontend, 1=backend, 2=server, 3=socket/listener
weight	[BS]	total weight (backend), server weight (server)
wredis	[BS]	number of times a request was redispatched to another server
wretr	[BS]	number of times a connection to a server was retried
wrew	[LFBS]	cumulative number of failed header rewriting warnings

CLI (Management Guide §9.3)

```

ex: global.stats socket /var/run/haproxy.sock mode 600 level admin
$ socat /var/run/haproxy.sock readline
$ nc -U /var/run/haproxy.sock
<type> : IPv4, IPv6, integer, string
<operator> : eq, ne, le, ge, lt, gt
<trace-level> : user, proto, state, data, developer
<trace-criterion> : backend, connection, frontend, listener,
                    nothing, server, session, thread
help
prompt
quit
expert-mode [on|off]
operator
user
set timeout_cli <delay>

```

```

show acl [<acl>]
show backend
show activity
show cli sockets
show cache
show env [<name>]
show errors [<iid>|<proxy>] [request|response]
show events [<sink>] [-w] [-n]
show fd [<fd>]
show info [typed|json] [desc]
show map [<map>]
show peers [<peers>]
show pools
show profiling
show servers state [<backend>]
show sess [<id>]
show stat [{<iid>|<proxy>} <type> <sid>] [typed|json] [desc]
show schema json
show resolvers [<resolvers>]
show table [<name>] [{data.<type> <operator> <value>}|key <key>}]
show threads
show tls-keys [<id>|*]
show trace [<source>]
get map <map> <value>
get acl <acl> <value>
get weight <backend>/<server>

```

```

add acl <acl> <pattern>
add map <map> <key> <value>
del acl <acl> [ <key> | #<ref> ]
del map <map> [ <key> | #<ref> ]
set maxconn frontend <frontend> <value>
set maxconn server <backend>/<server> <value>
set maxconn global <maxconn>
set rate-limit connections global <value>
set rate-limit http-compression global <value>
set rate-limit sessions global <value>
set rate-limit ssl-sessions global <value>
set server <backend>/<server> addr {<ip4>|<ip6>} [port <port>]
set server <backend>/<server> agent [ up | down ]
set server <backend>/<server> agent-addr <addr>
set server <backend>/<server> agent-send <value>
set server <backend>/<server> health [ up | stopping | down ]
set server <backend>/<server> check-port <port>
set server <backend>/<server> state [ready|drain|maint]
set server <backend>/<server> weight <weight>[%]
set server <backend>/<server> fqdn <FQDN>
set weight <backend>/<server> <weight>[%]
set ssl ocp-sp-response <response|payload>
set ssl tls-key <id> <tlskey>

```

```

clear counters [all]
clear acl <acl>
clear map <map>
clear table <table> [[data.<type> <operator> <value>]|key <key>]]
enable|disable agent <backend>/<server>
enable|disable dynamic-cookie-backend <backend>
enable|disable frontend <frontend>
enable|disable health <backend>/<frontend>
enable|disable server <backend>/<server>
set dynamic-cookie-key backend <backend> <value>
set map <map> [ <key> | #<ref> ] <value>
set severity-output [none|number|string]
set table <table> key <key> [data.<type> <value>]*
shutdown frontend <frontend>
shutdown session <id>
shutdown sessions server <backend>/<server>

```

```

debug dev <command> [args]*
set profiling { tasks } { auto | on | off }
trace
trace 0
trace <source> event [ [+|-!]<name> ]
trace <source> level [<trace-level>]
trace <source> lock [<trace-criterion>]
trace <source> { pause | start | stop } [ [+|-!]event ]
trace <source> sink [<sink>]
trace <source> verbosity [<level>]

```

Master CLI (Management Guide §9.4)

```

prompt
show proc
reload
@ID | @!PID switch to process cli
@ID | @!PID command execute command to @ID or @!PID

```

Timing (Configuration Manual §8.4)

Ta	(http) total active time for the request (=TR+Tw+Tc+Tr+Td)
Tc	time to establish connection to the server
Td	data transmission time
Th	total time to accept tcp connection and handshakes (ssl)
Ti	(http) idle time before first packet
Tq	total time to get the client request (=Th+Ti+TR)
TR	(http) total time to get the client request after handshakes
Tr	(http) server response time
Tt	total session duration time (=Tq+Tw+Tc+Tr+Td)
Tw	time waiting

Session state (Configuration Manual §8.5)

C	session unexpectedly aborted by the client
S	session unexpectedly aborted or refused by the server
P	session prematurely aborted by the proxy
L	locally processed and not passed by the server
R	resource on the proxy exhausted
I	internal error
D	session killed by proxy, when server goes down
U	session to backup server killed when others go up
K	actively killed by admin on proxy
c	client timeout
s	server timeout
-	normal session completion
R	proxy waiting for complete REQUEST from the client
Q	proxy waiting in the queue for a slot
C	proxy waiting for connection to established
H	proxy waiting for response headers from server
D	session in DATA phase
L	proxy transmit LAST data to client while server already finished
T	tarpit
-	normal session after end of data transfer
CC	client aborted before session to server
CD	client aborted during data transfer
cD	client timeout during data transfer
CH	client aborted while waiting for server response
cH	client timeout during POST data from client
CQ	client aborted during queue
CR	client aborted before sending full HTTP request
cR	client timeout before sending full HTTP request
CT	client aborted because of tarpit
LR	intercept by proxy cause of redirect or stats
SC	the connection was refused by something
sC	connection timeout before been completed
SD	connection died with error during DATA
sD	connection timeout during DATA
SH	server aborted before all headers was sent
sH	server timeout before all headers was sent
sQ	session queued too long
PC	proxy refused because of limits
PD	proxy blocked incorrect caused by invalid chunk
PH	proxy blocked response caused by invalid or security reason
PR	proxy blocked request caused by invalid or deny
PT	proxy blocked request and has tarpit
RC	a local resource has been exhausted

HAProxy response code

The error 4xx and 5xx codes above may be customized

200	access to stats page, and when replying to monitoring requests
400	for an invalid or too large request
401	when an authentication is required to perform the action (stats)
403	when a request is forbidden by a "http-request deny" rule
408	when the request timeout strikes before the request is complete
500	when haproxy encounters an unrecoverable internal error
502	deny response or empty, invalid or incomplete response by server
503	monitor fail or no server was available to handle the request
504	when the response timeout strikes before the server responds
30x	redirection, depending on the configured code

Command Line & System (Management Guide §3)

```
$ haproxy -f /etc/haproxy.cfg -D \  
-p /var/run/haproxy.pid -sf $(cat /var/run/haproxy.pid)  
-f CFG          define configuration file  
-C PATH        change directory  
-c            configuration check  
-D            daemon mode  
-d[x]         debug (x as options)  
-L NAME       local peer name  
-N LIMIT      default frontend maxconn  
-m LIMIT      memory limit to LIMIT  
-n LIMIT      maxconn limit to LIMIT  
-q           quiet  
-S BIND[, . . .] master-worker bind options  
-sf <pid> . . . process to send SIGUSR1  
-st <pid> . . . process to send SIGTERM  
-V           verbose  
-v[v]       version [more verbose]  
-W          master-worker mode  
-Ws        master-worker with systemd  
-x SOCKET  socket migration
```

Tricks & Tuning (Management Guide §5-7)

```
$ kill -USR1 <PID>      stop listening  
$ kill -USR2 <PID>      restart master worker  
$ kill -TERM <PID>     with "-st"  
$ kill -QUIT <PID>     flush pools  
$ ss                   list all system sockets (new)  
$ netstat -anp         list all system sockets (old)  
$ lsof                 list all open files to find fd  
$ ulimit -n           monitor number of file descriptors  
${VAR} or $VAR within double quoted strings in config files use environment value  
● HAPROXY_LOCALPEER  
● HAPROXY_CFGFILES  
● HAPROXY_MWORKS  
● HAPROXY_CLI  
● HAPROXY_MASTER_CLI
```

halog Usage

```
Usage: halog [-h|--help] for long help  
halog [-q] [-c] [-m <lines>]  
{-cc|-gt|-pct|-st|-tc|-srv|-u|-uc|-ue|-ua|-ut|-uao|-uto|-uba|-ubt|-ic}  
[-s <skip>] [-e|-E] [-H] [-rt|-RT <time>] [-ad <delay>] [-ac <count>]  
[-v] [-Q|-QS] [-tcn|-TCN <termcode>] [-hs|-HS [min][:max]] ]  
[ -time [min][:max]] ] < log
```

Input filters (several filters may be combined) :

-H	only match lines containing HTTP logs (ignore TCP)
-E	only match lines without any error (no 5xx status)
-e	only match lines with errors (status 5xx or negative)
-rt -RT <time>	only match response times larger smaller than <time>
-Q -QS	only match queued requests (any queue server queue)
-tcn -TCN <code>	only match requests with/without termination code <code>
-hs -HS <[min][:][max]>	only match requests with HTTP status codes within/not within min..max. Any of them may be omitted. Exact code is checked for if no ':' is specified.
-time <[min][:][max]>	only match requests recorded between timestamps. Any of them may be omitted.

Modifiers

-v	invert the input filtering condition
-q	don't report errors/warnings
-m <lines>	limit output to the first <lines> lines
-s <skip_n_fields>	skip n fields from the beginning of a line (default 5) you can also use -n to start from earlier then field 5

Output filters - only one may be used at a time

-c	only report the number of lines that would have been printed
-pct	output connect and response times percentiles
-st	output number of requests per HTTP status code
-cc	output number of requests per cookie code (2 chars)
-tc	output number of requests per termination code (2 chars)
-srv	output statistics per server (time, requests, errors)
-u*	output statistics per URL (time, requests, errors)

Additional characters indicate the output sorting key :

-u	: by URL, -uc : request count, -ue : error count
-ua	: average response time, -ut : average total time
-uao, -uto	: average times computed on valid ('OK') requests
-uba, -ubt	: average bytes returned, total bytes returned

HAProxy® is a registered trademark of HAProxy Technologies - <https://www.haproxy.com/trademark-policy/>. Thanks to Willy Tarreau, Thierry Fournier and many others for their feedback and suggestions.