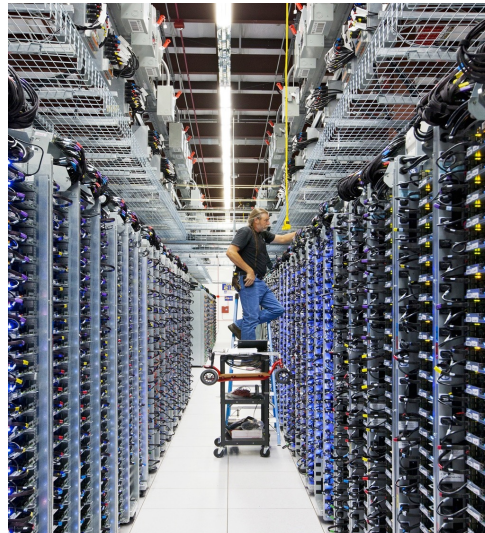


# HAProxy



# HAProxy

- HAProxy (High Availability Proxy) is a load balancer implemented in software
  - Available as Open Source (GPL/LGPL license) (<http://www.haproxy.org>)
  - Available as commercial product (<http://www.haproxy.com>)
    - Also available as appliance: ALOHA
  - Runs on FreeBSD, Linux, OpenBSD and Solaris
- Written by Willy Tarreau in 2000
  - Willy is the maintainer of the Linux 2.4 kernel
  - Lives in Fontenay aux Roses
- Used by high-profile websites: GitHub, Bitbucket, Stack Overflow, Reddit, Tumblr, Twitter



# HAProxy

## Features

- HAProxy can be used for
  - Load balancing on TCP layer and HTTP layer
  - Normalization / filtering of TCP and HTTP traffic
  - HTTP rewriting
  - SSL offloading
  - HTTP compression offloading
  - Traffic regulation
  - Protection against DDoS and service abuse
  - ...

# HAProxy operations

- Health checks

- HAProxy periodically sends probes to servers to check if they are still operational. A probe can be superficial or go deeper:
  - ping to server's IP address (TCP mode)
  - TCP connection to server's HTTP port (TCP mode)
  - HTTP request to server (HTTP mode)
- Based on health checks HAProxy sets a server's state to **UP** or **DOWN**

- Server administrative state

- The administrator can set a server into one of three administrative states
  - **READY** — Server is in normal mode, accepting requests
  - **DRAIN** — Removes server from load balancing, but still allows it to be health-checked and accept new persistent connections.
  - **MAINT** — Disables any traffic to the server as well as any health checks.

## HAProxy load balancing

- HAProxy offer a number of load balancing policies (see “**balance**” keyword in config. manual)
  - **roundrobin** — Each server is used in turns, according to their weights, which can be changed on the fly.
  - **static-rr** — Static round robin, weights cannot be changed on the fly.
  - **leastconn** — The server with the lowest number of connections receives the connection.
  - **first** — The first server with available connection slots receives the connection.
  - **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.
  - **uri** — This algorithm hashes either the left part of the URI (before the question mark) or the whole URI (if the "whole" parameter is present) and divides the hash value by the total weight of the running servers. The result designates which server will receive the request.
  - **url\_param** — The URL parameter specified in argument will be looked up in the query string of each HTTP GET request. If the parameter is found followed by an equal sign ('=') and a value, then the value is hashed and divided by the total weight of the running servers. The result designates which server will receive the request.
  - **hdr(<name>)** — The HTTP header <name> will be looked up in each HTTP request and hashed.
  - **rdp-cookie, rdp-cookie(<name>)** — The RDP cookie <name> (or "mstshash" if omitted) will be looked up and hashed for each incoming TCP request.

# HAProxy

## Configuration file

- **global** section
  - Sets process-wide parameters
  - Often OS-specific
- **defaults** section
  - Sets default parameters for all other sections following its declaration
  - Options for logging go here
- **frontend** section
  - Describes a **set of listening sockets** accepting client connections
- **backend** section
  - Describes a **set of servers** to which the proxy will connect to forward incoming connections
  - Load balancing policies go here
  - Sticky session policies go here

```
# Simple configuration for an HTTP proxy
# listening on port 80 on all interfaces and
# forwarding requests to a single backend
# "servers" with a single server "server1"
# listening on 127.0.0.1:8000
global
    daemon
    maxconn 256

defaults
    mode http
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

frontend http-in
    bind *:80
    default_backend servers

backend servers
    server server1 127.0.0.1:8000 maxconn 32
```

Source: <https://cbonte.github.io/haproxy-dconv/configuration-1.5.html>

# HAProxy

## Statistics report

- HAProxy provides a web-based statistics report that is useful to inspect the server state

## HAProxy

### Statistics Report for pid 15

#### > General process information

pid = 15 (process #1, nbproc = 1)  
 uptime = 0d 9h05m00s  
 system limits: memmax = unlimited; ulimit-n = 4044  
 maxsock = 4044; maxconn = 2000; maxpipes = 0  
 current conns = 2; current pipes = 0/0; conn rate = 1/sec  
 Running tasks: 1/9; idle = 100 %

<span style="background-color: #90EE90; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> active UP	<span style="background-color: #ADD8E6; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> backup UP
<span style="background-color: #FFD700; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> active UP, going down	<span style="background-color: #800080; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> backup UP, going down
<span style="background-color: #FFA500; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> active DOWN, going up	<span style="background-color: #FF69B4; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> backup DOWN, going up
<span style="background-color: #FF6347; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> active or backup DOWN	<span style="background-color: #D3D3D3; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> not checked
<span style="background-color: #FF8C00; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> active or backup DOWN for maintenance (MAINT)	
<span style="background-color: #0000FF; border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> active or backup SOFT STOPPED for maintenance	

Display option:

- Scope :
- [Hide 'DOWN' servers](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [Primary site](#)
- [Updates \(v1.5\)](#)
- [Online manual](#)

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

#### stats

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				1	1	-	1	1	2 000	1			0	0	0	0	0						OPEN								
Backend	0	0		0	0		0	0	200	0	0	0s	0	0	0	0	0	0	0	0	0	0	9h5m UP		0	0	0		0		

#### localnodes

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	2	-	1	3	2 000	24			392 906	793 767	0	0	0						OPEN								

#### nodes

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
s1	0	0	-	0	50		0	1	-	1 004	209	4s	197 360	397 980		0		0	0	0	0	0	9h5m UP	L7OK/200 in 3ms	1	Y	-	0	0	0s	-
s2	0	0	-	0	50		0	1	-	1 000	208	2h19m	195 546	395 787		0		0	0	0	0	0	9h5m UP	L7OK/200 in 3ms	1	Y	-	0	0	0s	-
Backend	0	0		0	100		0	2	200	2 004	417	4s	392 906	793 767	0	0		0	0	0	0	0	9h5m UP		2	2	0		0	0s	

# HAProxy

## Statistics report

- The statistics report can be configured in the **listen** section of the configuration file

File haproxy.cfg:

```
listen stats *:1936
    stats enable
    stats uri /
    stats hide-version
```



# HAProxy

## Command line interface

- HAProxy provides a command line interface for the administrator, connecting to the running process (“stats socket”). Allows administrator to
  - perform operations such as
    - Change a server’s address, weight and status (e.g., put a server in drain mode or maintenance mode)
    - Apply connection limits and rate limits on the fly to frontends
    - Disable a specific frontend to release a listening port
  - retrieve troubleshooting information
    - Consult statistics
    - Dump stickiness tables
    - Dump client-side and server-side connections
    - Dump captured errors
- The stats socket can be defined in the **global** section of the configuration file.

### File haproxy.cfg:

```
global
    stats socket ipv4@0.0.0.0:9999 level admin
```

### Connecting to the stats socket using socat tool:

```
$ socat - tcp:localhost:9999
prompt
> show info
Name: HAProxy
Version: 1.5.14
Release_date: 2015/07/02
Nbproc: 1
Process_num: 1
[...]
```

# HAProxy

## Documentation

- Starter guide  
<http://www.haproxy.org/download/1.6/doc/intro.txt>
- Configuration file  
<https://cbonte.github.io/haproxy-dconv/configuration-1.5.html#2>
- Command-line interface (CLI)  
<https://cbonte.github.io/haproxy-dconv/configuration-1.5.html#9.2>
  - Commands to change servers' administrative state  
<https://cbonte.github.io/haproxy-dconv/configuration-1.5.html#9.2-set%20server>