

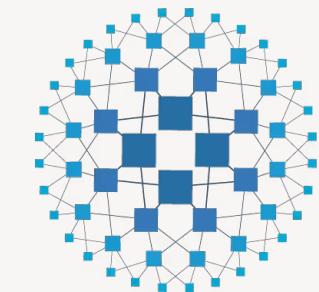


HAProxy



"High Availability Proxy"

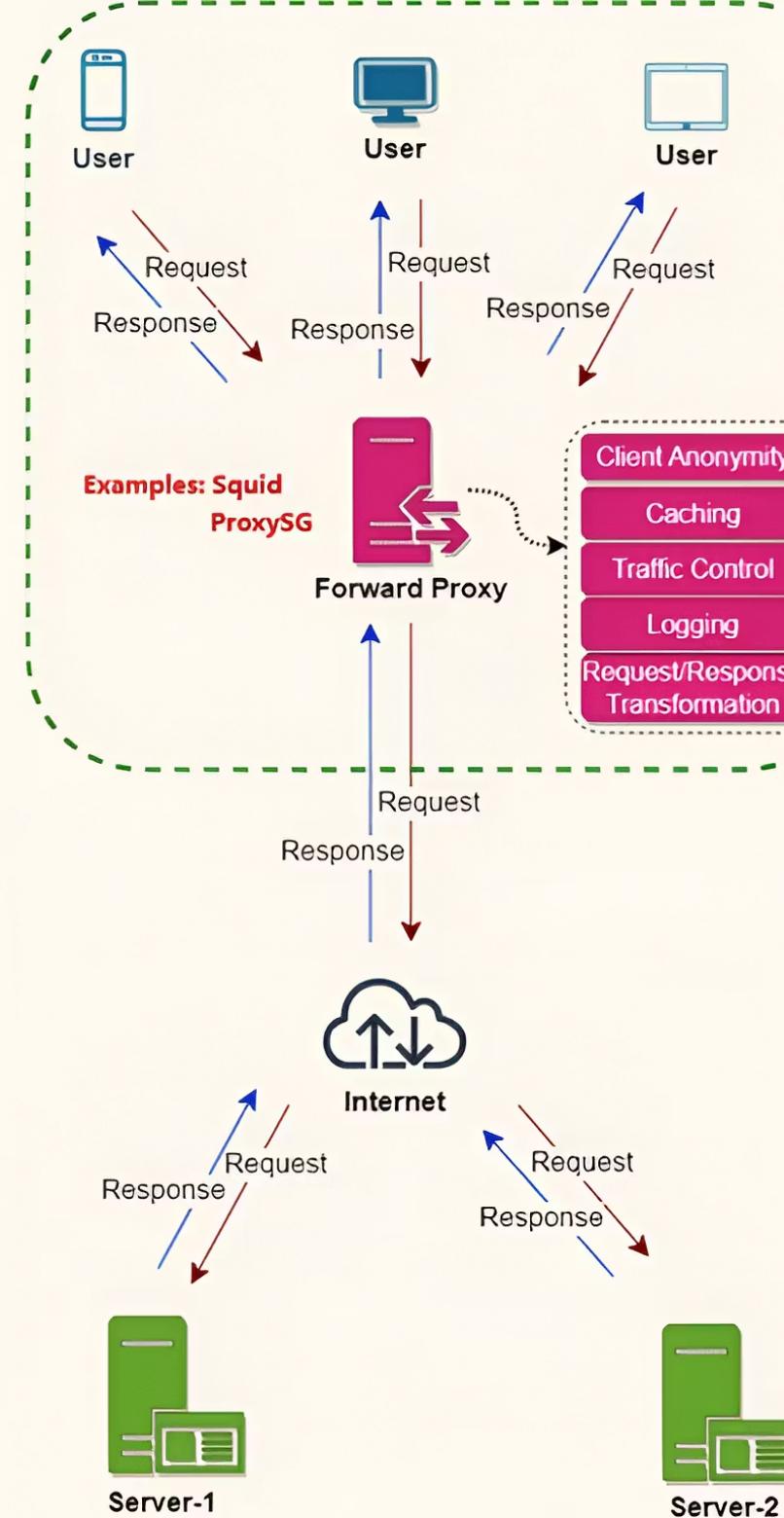
- Reverse Proxy
- Load Balancer
- Layer 4 and 7
- Content Switching



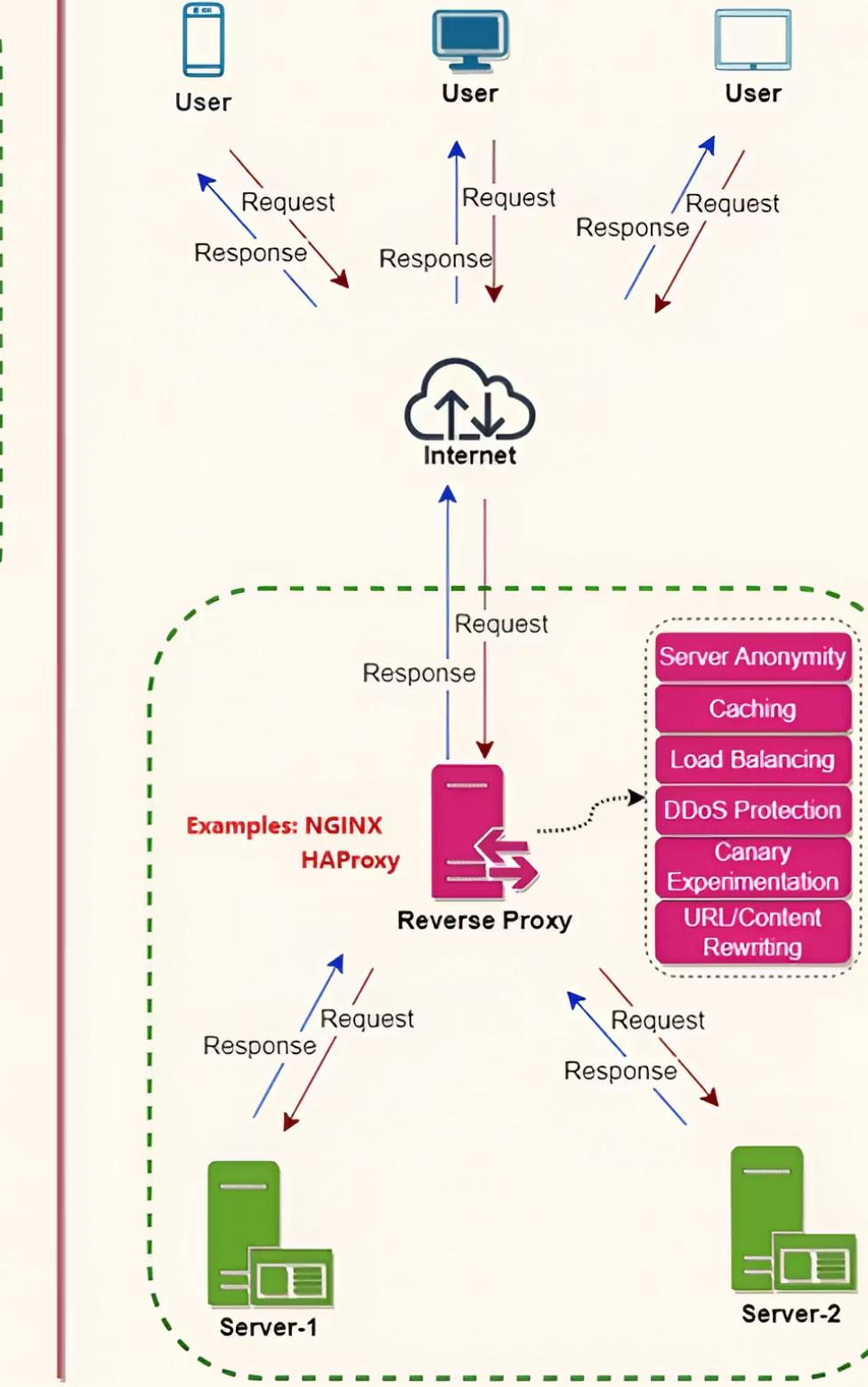
HAPROXY



Forward Proxy



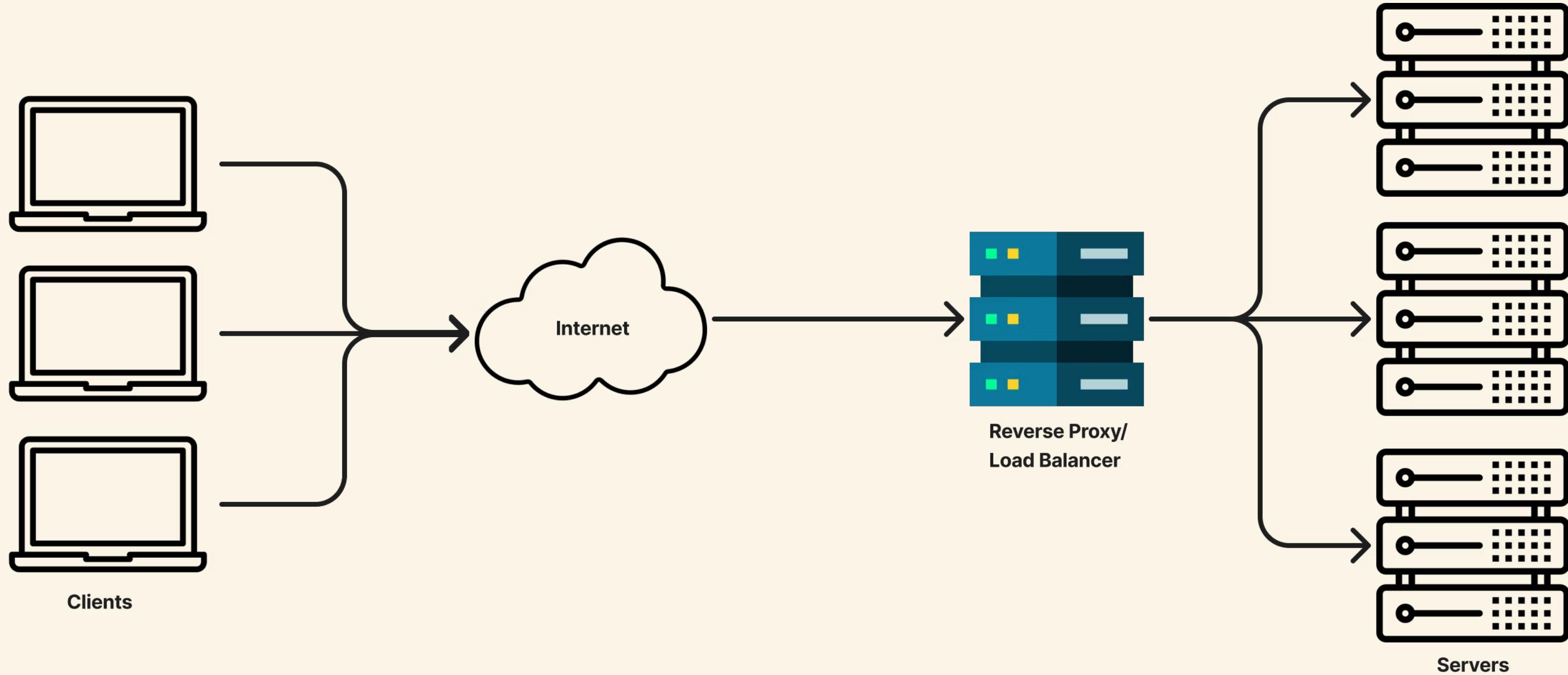
Reverse Proxy



<https://www.designgurus.io/>

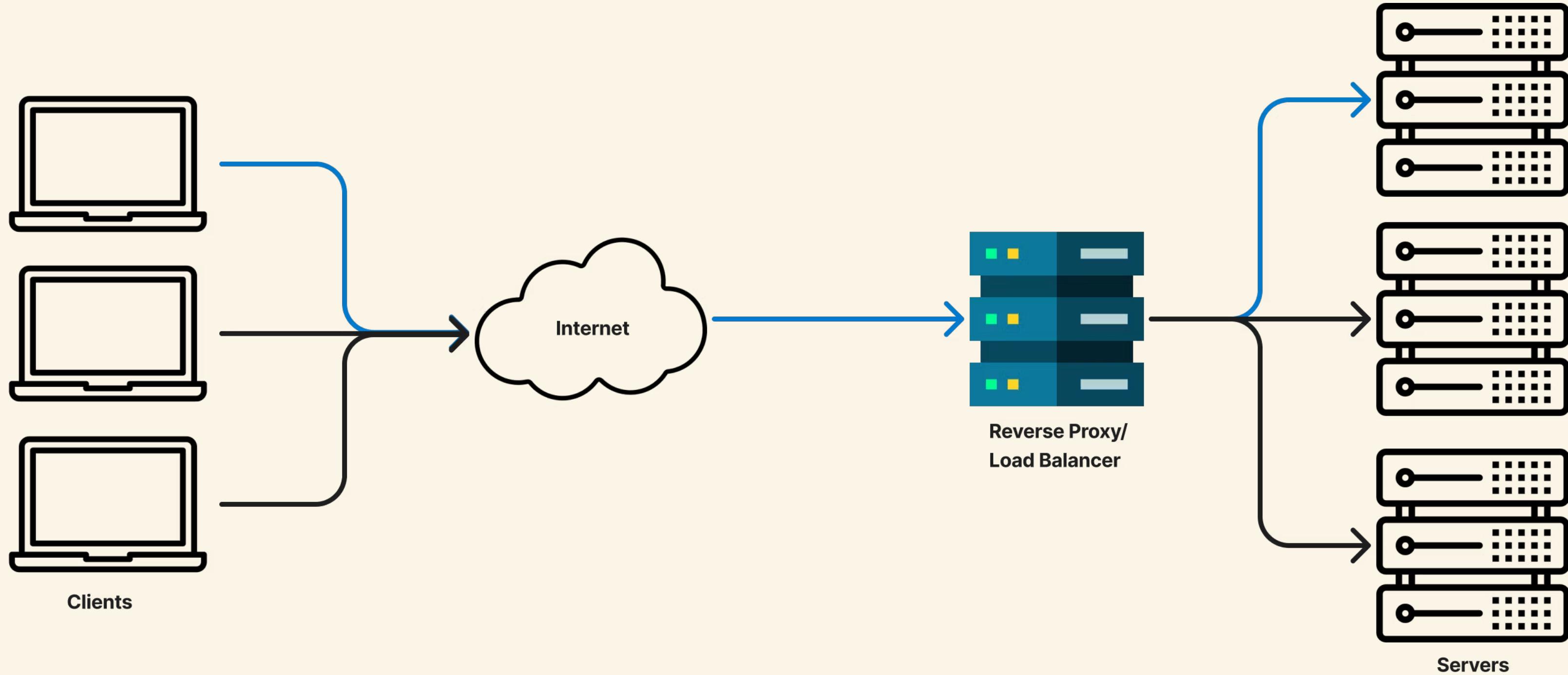
Load Balancer

Reliability, Availability



Load Balancer

Reliability, Availability

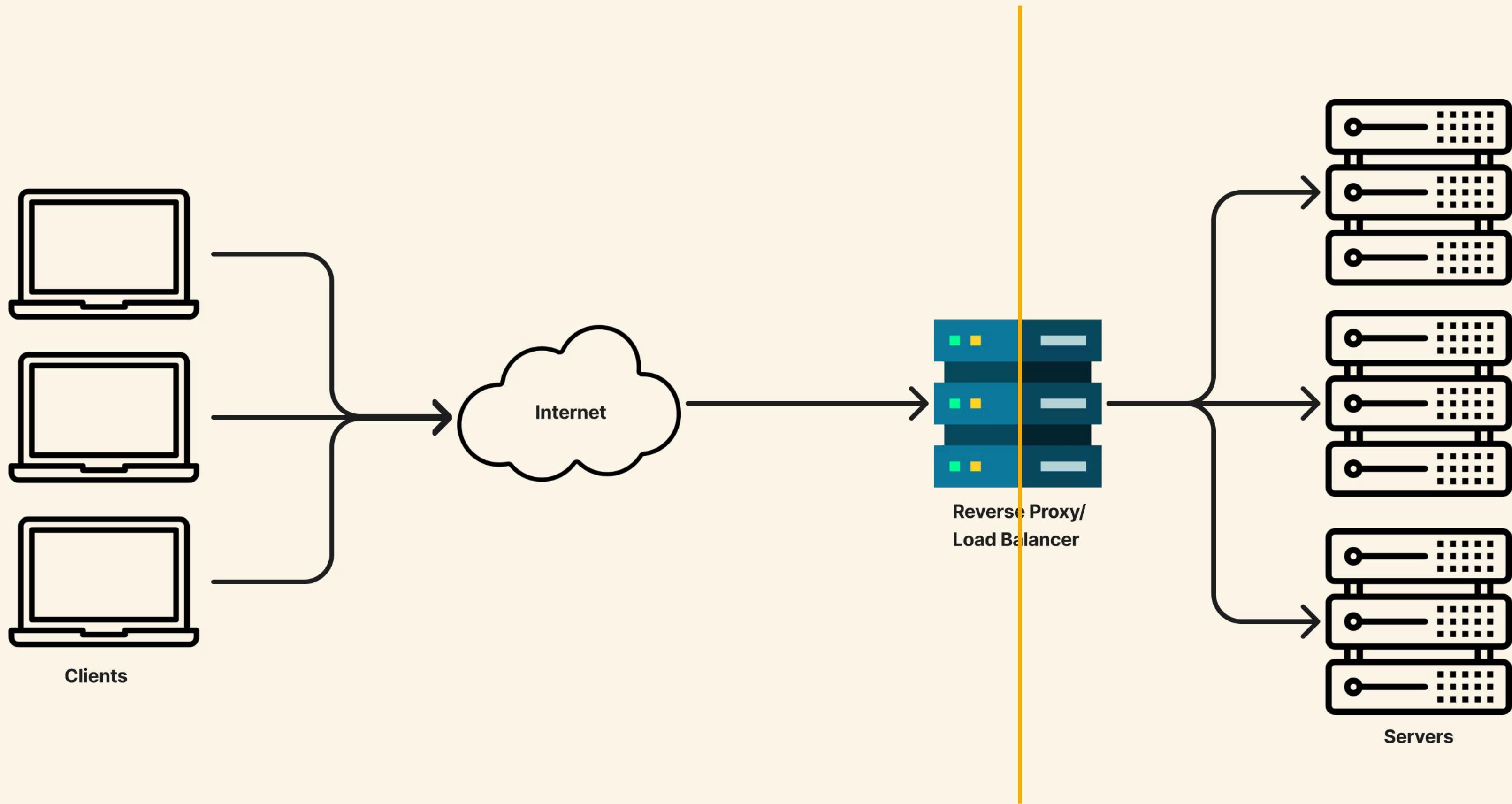


HAProxy

- 2001; Willy Tarreau
- HAProxy Technologies
- Linux, BSD, Solaris, macOS
- C
- Open Source

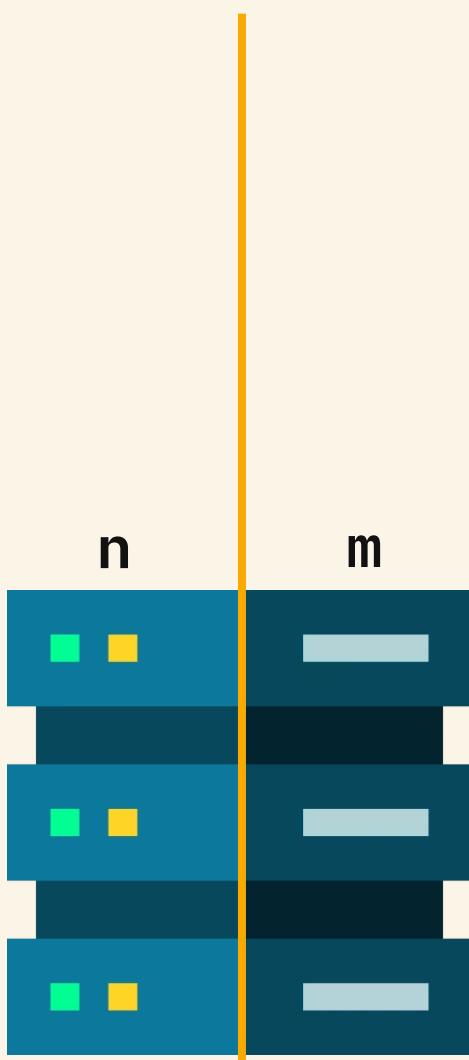


<https://www.haproxy.com/blog/author/wtarreau>



Frontend

- Listen on **specified ports**
- Process & forward **rule-based** to backend servers



Backend

- Connect to **servers** (usually internal network)
- Forward responses

Frontend

- Layer 4 vs. 7 Proxying
- Content switching through Access Control Lists (ACL)
- Rate Limiting, HTTP Normalizing



Backend

- Layer 4 vs. 7 Proxying
- Balancing Algorithms
- Health Checks & Timeouts
- Session stickiness

- Logging
- Statistics
- Server Protection
- Threaded workers

Layer 4 vs. Layer 7

- TCP: IP & Port => no **inspection**, **passthrough**
 - **Faster**
=> Use cases:
 - Ultra-Low latency
 - L4-based connections, e.g. to Postgres, MySQL Databases
 - Don't want to decrypt at LB level
 - HTTP & TLS (**is terminated**)
 - **Slightly slower**, but servers do not have to **terminate TLS**
 - Advanced routing (e.g. ACLs) => MS
 - Sticky sessions, e.g. cookies
 - Modify HTTP Headers, Paths, ...
 - Basic caching
- => Standard use case for "normal" web servers

Layer 4 vs. Layer 7

- TCP: IP & Port => no inspection, passthrough
- Faster
 - => Use cases:
 - Ultra-Low latency
 - L4-based connections, e.g. to Postgres, MySQL Databases
 - Don't want to decrypt at LB level
- HTTP & TLS (is terminated)
- Slightly slower, but servers do not have to terminate TLS
- Advanced routing (e.g. ACLs) => MS
- Sticky sessions, e.g. cookies
- Modify HTTP Headers, Paths, ...
- Basic caching
 - => Standard use case for "normal" web servers

Frontend

- Layer 4 vs 7 Proxying
- Content Switching through Access Control Lists (ACL)
- Rate Limiting, HTTP Normalizing



Backend

- Layer 4 vs 7 Proxying
- **Balancing Algorithms**
- Health Checks & Timeouts
- Session stickiness

- Logging
- Statistics
- Server Protection
- Threaded workers



Balancing Algorithms

RoundRobin

leastconn

Source IP

Frontend

- Layer 4 vs 7 Proxying
- Content Switching through Access Control Lists (ACL)
- Rate Limiting, HTTP Normalizing



Backend

- Layer 4 vs 7 Proxying
- Balancing Algorithms
- Health Checks & Timeouts
- Session stickiness

- Logging
- Statistics
- Server Protection
- Threaded workers

Frontend

- Layer 4 vs 7 Proxying
- Content Switching through Access Control Lists (ACL)
- Rate Limiting, HTTP Normalizing



Backend

- Layer 4 vs 7 Proxying
- Balancing Algorithms
- Health Checks & Timeouts
- Session stickiness

- Logging
- Statistics
- Server Protection
- Threaded workers

Frontend

- Layer 4 vs 7 Proxying
- Content Switching through Access Control Lists (ACL)
- Rate Limiting, HTTP Normalizing



Backend

- Layer 4 vs 7 Proxying
- Balancing Algorithms
- Health Checks & Timeouts
- Session stickiness

- Logging
- Statistics
- Server Protection
- Threaded workers

Performance

Scenario	Avg. RPS	Latency with proxy	Latency standalone
L7/HTTP	~2.075 Mio	~0.54 ms	~0.16 ms
L7/HTTPS	~2.00 Mio	~0.57 ms	~0.16 ms

- April 2021
- HAProxy 2.4
- AWS Graviton2 (ARM Neoverse N1), 64 vCPUs at 2.5 GHz base
- 100 Gbps ENA
- Ubuntu 20.04 LTS

<https://www.haproxy.com/blog/haproxy-forwards-over-2-million-http-requests-per-second-on-a-single-aws-arm-instance>



Live Showcase

Clone Showcase & Slides



<https://github.com/sam01de/haproxy-showcase#>

HAProxy vs. nginx

- Specialized reverse proxy/load balancer
- TCP/HTTP/HTTPS
- Configure complex setups
- Granular healthchecks
- High loads

=> Only reverse proxy/load balancer
=> Very high loads
=> Complex setups, rules,
healthchecks

- Web-Server & reverse proxy/load balancer
- Deliver static content
- Advanced caching setups
- TCP/UDP/HTTP/HTTPS

=> Easy setups
=> Need web server capabilities
=> Need extensive caching



Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

[Create a presentation \(It's free\)](#)