# HAproxy

یک سرویس ایجاد load balancing است. فایل کانفیگ این سرویس پس از نصب در مسیر /etc/haproxy/haproxy.conf قرار میگیرد. این فایل به صورت کلی از ۵ قسمت global، default ، frontend و backend و listen تشکیل شده است.

## قسمت global

در این قسمت هر متغیری که ست شود بر روی تمام frontend و backend ها اعمال می شود مگر اینکه بعدا در همان ماژول این مقدار دوباره ست شود. مواردی که می توان در این قسمت ست کرد به شرح زیر است

* Maxconn: که این متغیر حداکثر تعداد کانکشن را مشخص میکند
* Log : این متغیر تعیین می کند لاگی که توسط این سامانه تولید می شود در همین ماشین (local) باقی بماند یا برای یک syslog سرور بیرونی ارسال شود
* User/group : این دو متغیر تعیین می کنند که چه کسی حق اجرا کردن این برنامه را دارد
* Stats: این متغیر در واقع یک سوکت (runtime api) برای تغییر مقادیر haproxy‌ به صورت دینامیک ایجاد می کند
* Nbproc: تعداد پراسسی که می تواند برای ورکر های haproxy استفاده می شود
* Nbthread : تعداد تردهایی که می تواند برای ورکرهای haproxy استفاده شود.
* Cpu-map : اعلام میکند که ورکر اول روی کدام پردازنده باشد

Cpu-map 1 0

Cpu-map 2 1

در این مدل پراسس اول روی پردازنده صفر و پراسس دوم روی پردازنده ۱ کار کند

## قسمت defaults

* Timeout connect: مدت زمانی که اگر چک اعمال شود و متصل دیده نشود ارتباط قطع شود
* Timeout client: مدت زمانی که اگر کاربر چک اعمال شود و متصل دیده نشود ارتباط قطع شود
* Timeout server: مدت زمانی که اگر سرور چک اعمال شود و متصل دیده نشود ارتباط قطع شود
* Log global : یعنی موضوع لاگ از کانفیگ قسمت global تبعیت کند
* Mode http: مد می تواند از نوع tcp و udp و http باشد

## قسمت frontend

به ازای هر سایتی که از بیرون باید دیده شود می توان یک قسمت frontend در نظر گرفت. در واقع جلوی اسم frontend‌ اسم سایت می آید و شماره پورت آن

Frontend www.example.com

* Bind: اعلام می کند روی کدام IP و کدام پورت از سرور برای این آدرس گوش کن مثلا 10.10.10.10:443
* Default\_backend :اعلام میکند برای این سرور قسمت backend‌ چه نامی دارد مثلا default\_backend: servers
* Use\_backend: در این قسمت می توانیم تعیین کنیم مثلا اگر در قسمت url یک اسمی داشتیم از یک backend‌ استفاده کند و اگر یک اسم دیگر باشد از یک backend دیگر

Use\_backend <name> {path\_beg /<path>}

Use\_backend api {path\_beg /api/}

Use\_backend jpg {path\_beg /jpg/}

Use\_backend gif\_pics {path\_beg /gif/}

## قسمت backend

در این قسمت سرورهای back end و سرویس توزیع بار را مشخص می کنیم.

* Balance: نوع توزیع بار می تواند roundrobin و یا leastconn باشد که در مدل دوم درخواست به سروری ارسال می شود که تعداد کانکشن کمتری را دارد
* Option: در این مدل می توانیم نوع چک کردن زنده بودن سرورهای backend را مشخص کنیم. مثلا httpchk HEAD / اعلام می کند که یک پیام از نوع HEAD برای آدرس / سرور backend ارسال کند و در صورت گرفتن پیام 200 به این معنی است که سرور درست کار می کند.
* Default-server : مقادیر پیش فرض برای متغیرهای سرورها را تعیین می کند. در این مدل می توانیم به جای اینکه برای تمام سرورها این مقادیر را جداگانه تعیین کنیم در صورت امکان مقادیر ثابت را برای تمام آنها اعمال کنیم مثلا

Default-server check maxconn 2000

به این ترتیب عمل چک کردن زنده بودن روی تمام سرور های backend انجام می شود و در ضمن تعداد کانکشن روی سرورها 2000 عدد ست می شود

* Cookie‌: مسیر کوکی را مشخص می کند
* Server: آدرس و پورت سرور backend و سایر پارامترهای آن (در صورتی که در متغیر قبل ست نشده باشد) را می نویسیم

Server <name> <add>:<port> options

Server web1 172.20.20.1:80 cookie check maxconn 2000

Server web2 172.20.20.2:80 cookie check maxconn 20

## قسمت listen

در این قسمت مشخص می کنیم که web ui خود haproxy روی چه پورتی و چگونه بالا بیاید. در جلوی listen اسم مدنظر برای نمایش سایت را مشخص می کنیم.

Listen stats

* Bind: مشخص می کند که روی چه آدرس و چه پورتی stats نمایش داده شود

Bind 0.0.0.0:1936

* Stats auth : نام کاربری و کلمه عبور برای این سرویس را مشخص می کند

Stat auth admin:123

* Stats uri: آدرس این سایت را مشخص می کند

Stat uri /haproxy/stats

* Stats refresh: مدت زمانی که خود به خود صفحه به روز می شود را نشان می دهد

Stat refresh 30s

* Stats enable : این گزینه را فعال میکند
* Log : مشخص می کند که قابلیت از کدام قسمت به ارث ببرد

Log global