

HAProxy 配置

global

参数是进程级的，通常和操作系统（OS）相关。这些参数一般只设置一次，如果配置无误，就不需要再次配置进行修改

defaults

配置默认参数的，这些参数可以被利用配置到 frontend, backend, listen 组件

frontend

接收请求的前端虚拟节点，Frontend 可以根据规则直接指定具体使用后端的 backend(可动态选择)。

backend

后端服务集群的配置，是真实的服务器，一个 Backend 对应一个或者多个实体服务器。

listen

Frontend 和 Backend 的组合物。

以上 5 个组件不是必选的，可以根据需要选择部分作为配置。

详细的配置信息，请查阅官方文档

1.3 版本

<http://haproxy.1wt.eu/download/1.3/doc/configuration.txt>

1.4 版本

<http://haproxy.1wt.eu/download/1.4/doc/configuration.txt>

组件配置如下：

global

```
#全局的日志配置 其中日志级别是[err warning info debug]
#local0 是日志设备，必须为如下 24 种标准 syslog 设备的一种：
#kern user mail daemon auth syslog lpr news
#uucp cron auth2 ftp ntp audit alert cron2
#local0 local1 local2 local3 local4 local5 local6 local7
#但是之前在/etc/syslog.conf 文件中定义的是 local0 所以
#这里也是用 local0
#如果日志无法写到 haproxy.log，解决方案如下
#vi /etc/syslog.conf
#添加 local3.* /var/log/haproxy.log
#添加 local0.* /var/log/haproxy.log
#vi /etc/sysconfig/syslog
#把 SYSLOGD_OPTIONS="-m 0" 改成 SYSLOGD_OPTIONS="-r -m 0"
#重启 syslogd: /etc/init.d/syslog restart
log 127.0.0.1 local0 info #[err warning info debug]
#最大连接数
```

```
maxconn 4096
#用户
user admin
#组
group admin
#使 HAProxy 进程进入后台运行。这是推荐的运行模式
daemon
#创建 4 个进程进入 daemon 模式运行。此参数要求将运行模式设置为“daemon”
nbproc 4
#将所有进程的 pid 写入文件启动进程的用户必须有权限访问此文件。
pidfile /usr/local/haproxy/haproxy.pid
```

defaults

```
#默认的模式 mode { tcp|http|health }, tcp 是 4 层, http 是 7 层, health 只会返回 OK
mode http
#采用 http 日志格式
option httplog
#三次连接失败就认为是服务器不可用, 也可以通过后面设置
retries 3
如果 cookie 写入了 serverId 而客户端不会刷新 cookie,
#当 serverId 对应的服务器挂掉后, 强制定向到其他健康的服务器
option redispatch
#当服务器负载很高的时候, 自动结束掉当前队列处理比较久的链接
option abortonclose
#默认的最大连接数
maxconn 4096
#连接超时
timeout 5000
#客户端超时
clitimeout 30000
#服务器超时
srvtimeout 30000
#心跳检测超时
timeout check 2000
#注: 一些参数值为时间, 比如说 timeout。时间值通常单位为毫秒(ms), 但是也可以通过
加#后缀, 来使用其他的单位。
#- us : microseconds. 1 microsecond = 1/1000000 second
#- ms : milliseconds. 1 millisecond = 1/1000 second. This is the default.
#- s : seconds. 1s = 1000ms
#- m : minutes. 1m = 60s = 60000ms
#- h : hours. 1h = 60m = 3600s = 3600000ms
#- d : days. 1d = 24h = 1440m = 86400s = 86400000ms
```

#####统计页面配置#####

listen admin_stats

```
#监听端口
bind 0.0.0.0:1080
#http 的 7 层模式
mode http
#日志设置
log 127.0.0.1 local0 err #[err warning info debug]
#统计页面自动刷新时间
stats refresh 30s

#统计页面 url
stats uri /admin?stats
#统计页面密码框上提示文本
stats realm Gemini\ Haproxy
#统计页面用户名和密码设置
stats auth admin:admin
stats auth admin1:admin1
#隐藏统计页面上 HAProxy 的版本信息
stats hide-version
#####网站检测 listen 定义#####
```

listen site_status

```
bind 0.0.0.0:1081
mode http
log 127.0.0.1 local0 err #[err warning info debug]
#网站健康检测 URL，用来检测 HAProxy 管理的网站是否可以用，正常返回 200，不正常返回
500
monitor-uri /site_status
#定义网站 down 时的策略
#当挂在负载均衡上的指定 backend 的中有效机器数小于 1 台时返回 true
acl site_dead nbsrv(denali_server) lt 1
acl site_dead nbsrv(tm_server) lt 1
acl site_dead nbsrv(mms_server) lt 1
#当满足策略的时候返回 500
monitor fail if site_dead
#如果 192.168.0.252 或者 192.168.0.31 这两天机器挂了
#认为网站挂了，这时候返回 500，判断标准是如果 mode 是
#http 返回 200 认为是正常的，如果 mode 是 tcp 认为端口畅通是好的
monitor-net 192.168.0.252/31
```

#####frontend 配置#####

frontend http_80_in

```
#监听端口
bind 0.0.0.0:80
#http 的 7 层模式
mode http
#应用全局的日志配置
log global
#启用 http 的 log
option httplog
#每次请求完毕后主动关闭 http 通道，HA-Proxy 不支持 keep-alive 模式
option httpclose
#如果后端服务器需要获得客户端的真实 IP 需要配置次参数，将可以从 Http Header 中
#获得客户端 IP
option forwardfor
```

#####HAProxy 的日志记录内容配置#####

```
capture request header Host len 40
capture request header Content-Length len 10
capture request header Referer len 200
capture response header Server len 40
capture response header Content-Length len 10
capture response header Cache-Control len 8
```

#####acl 策略定义#####

```
#如果请求的域名满足正则表达式返回 true -i 是忽略大小写
acl denali_policy hdr_reg(host) -i ^(www.xxx.net|...
#如果请求域名满足 trade.xxx.net 返回 true -i 是忽略大小写
acl tm_policy hdr_dom(host) -i trade.xxx.net
##在请求 url 中包含 sip_apiname=, 则此控制策略返回 true, 否则为 false
acl invalid_req url_sub -i sip_apiname=
##在请求 url 中存在 timetask 作为部分地址路径, 则此控制策略返回 true, 否则返回 false
acl timetask_req url_dir -i timetask
#当请求的 header 中 Content-length 等于 0 时返回 true
acl missing_cl hdr_cnt(Content-length) eq 0
```

#####acl 策略匹配相应#####

```
##当请求中 header 中 Content-length 等于 0 阻止请求返回 403
block if missing_cl
##block 表示阻止请求, 返回 403 错误, 当前表示如果不满足策略 invalid_req, 或者满足
```

策略 timetask_req, 则阻止请求。
block if !invalid_req || timetask_req
#当满足 denali_policy 的策略时使用 denali_server 的 backend
use_backend denali_server if denali_policy
#当满足 tm_policy 的策略时使用 tm_server 的 backend
use_backend tm_server if tm_policy
#reqisetbe 关键字定义, 根据定义的关键字选择 backend
reqisetbe ^Host:\ img dynamic
reqisetbe ^[\]*\ /(img|css)/ dynamic
reqisetbe ^[\]*\ /admin/stats stats
#以上都不满足的时候使用默认 mms_server 的 backend
default_backend mms_server
#HAProxy 错误页面设置
errorfile 400 /usr/local/haproxy/errorfiles/400.http
errorfile 403 /usr/local/haproxy/errorfiles/403.http
errorfile 408 /usr/local/haproxy/errorfiles/408.http
errorfile 500 /usr/local/haproxy/errorfiles/500.http
errorfile 502 /usr/local/haproxy/errorfiles/502.http

errorfile 503 /usr/local/haproxy/errorfiles/503.http
errorfile 504 /usr/local/haproxy/errorfiles/504.http

#####backend 的设置#####

backend mms_server

#http 的 7 层模式
mode http
#负载均衡的方式, roundrobin 平均方式
balance roundrobin
#允许插入 serverid 到 cookie 中, serverid 后面可以定义
cookie SERVERID
#心跳检测的 URL, HTTP/1.1\r\nHost:XXXX, 指定了心跳检测 HTTP 的版本, XXX 为检测时请求
#服务器的 request 中的域名是什么, 这个在应用的检测 URL 对应的功能有对域名依赖的话
需要设置
option httpchk GET /member/login.html HTTP/1.1\r\nHost:xxx.com
#服务器定义, cookie 1 表示 serverid 为 1, check inter 1500 是检测心跳频率
#rise 3 是 3 次正确认为服务器可用, fall 3 是 3 次失败认为服务器不可用, weight 代表
权重
server mms1 10.1.5.134:80 cookie 1 check inter 1500 rise 3 fall 3 weight 1
server mms2 10.1.6.118:80 cookie 2 check inter 1500 rise 3 fall 3 weight 2

backend denali_server

```
mode http
#负载均衡的方式，source 根据客户端 IP 进行哈希的方式
balance source
#但设置了 backup 的时候，默认第一个 backup 会优先，设置 option allbackups 后
#所有备份服务器权重一样
option allbackups
#心跳检测 URL 设置
option httpchk GET /xxx/xxx.html HTTP/1.1\r\nHost:xxx.com
#可以根据机器的性能不同，不使用默认的连接数配置而使用自己的特殊的连接数配置
#如 minconn 10 maxconn 20
server denlail 10.1.5.114:80 minconn 4 maxconn 12 check inter 1500 rise 3 fall 3
server denlai2 10.1.6.104:80 minconn 10 maxconn 20 check inter 1500 rise 3 fall 3
#备份机器配置，正常情况下备机不会使用，当主机的全部服务器都 down 的时候备机会启用
server dnali-back1 10.1.7.114:80 check backup inter 1500 rise 3 fall 3
server dnali-back2 10.1.7.114:80 check backup inter 1500 rise 3 fall 3
```

backend tm_server

```
mode http
#负载均衡的方式，leastconn 根据服务器当前的请求数，取当前请求数最少的服务器
balance leastconn
option httpchk GET /xxx/xxx.htm HTTP/1.1\r\nHost:trade.xxx.com

server tm1 10.1.5.115:80 check inter 1500 rise 3 fall 3
server tm2 10.1.6.105:80 check inter 1500 rise 3 fall 3
#####requisetbe 自定义关键字匹配 backend 部分#####
```

backend dynamic

```
mode http
balance source
option httpchk GET /welcome.html HTTP/1.1\r\nHost:www.xxx.net
server denlail 10.3.5.114:80 check inter 1500 rise 3 fall 3
server denlai2 10.4.6.104:80 check inter 1500 rise 3 fall 3
```

backend stats

```
mode http
balance source
option httpchk GET /welcome.html HTTP/1.1\r\nHost:www.xxx.net
server denlail 10.5.5.114:80 check inter 1500 rise 3 fall 3
server denlai2 10.6.6.104:80 check inter 1500 rise 3 fall 3
```

haproxy 的启动

```
/usr/local/haproxy/sbin/haproxy -f /usr/local/haproxy/haproxy.cfg
```

Haproxy 的重启

```
/usr/local/haproxy/sbin/haproxy -f /usr/local/haproxy/haproxy.cfg -st `cat  
/usr/local/haproxy/haproxy.pid` (不要换行)
```

Haproxy 的停止

```
Killall -9 haproxy
```

启动参数

haproxy -f < 配置文件>

[-n 最大并发连接总数] [-N 每个侦听的最大并发数] [-d] [-D] [-q] [-V] [-c] [-p] [-s] [-l] [-dk]
[-ds] [-de] [-dp] [-db] [-m < 内存限制 M>] [{-sf|-st} pidlist...]

-d 前台，debug 模式

-D daemon 模式启动

-q 安静模式,不输出信息

-V 详细模式

-c 对配置文件进行语法检查

-s 显示统计数据

-l 显示详细统计数据

-dk 不使用 kqueue

-ds 不使用 speculative epoll

-de 不使用 epoll

-dp 不使用 poll

-db 禁用后台模式，程序跑在前台

-sf 程序启动后向 pidlist 里的进程发送 FINISH 信号，这个参数放在命令行的最后

-st 程序启动后向 pidlist 里的进程发送 TERMINATE 信号，这个参数放在命令行的最后