研究学习

正则、匹配：

-i是忽略大小写

1. hdr\_reg(host)

acl host\_xman\_test hdr\_reg(host) –I ^f\.[a-z]+.(xman|ufo).xbox.com.cn

匹配对应的请求的主机头。其中“.”是转义符，匹配如yuanfang.ufo.xbox.com.cn

1. hdr\_dom(host)

acl tm\_policy hdr\_dom(host) -i www.inbank.com

请求域名满足www.inbank.com

1. url\_sub

acl invalid\_req url\_sub -i sip\_apiname=

在请求url中包含sip\_apiname=

1. url\_dir

acl timetask\_req url\_dir -i timetask

在请求url中存在timetask作为部分地址路径

1. hdr\_cnt(Content-length)

acl missing\_cl hdr\_cnt(Content-length) eq 0

当请求的header中Content-length等于0时

1. path\_beg

acl url\_image path\_beg -i /uploads/user/avatar/13

1. path\_end

acl url\_static path\_end -i .jpg .gif .png .css .js

1. hdr\_beg(host)

acl is\_example hdr\_dom(host) -i example.com

1. block if !invalid\_req || timetask\_req

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。

1. block if missing\_cl

当匹配到missing\_cl则阻止请求返回403

haproxy用法

haproxy -c -f haproxy.cfg #对haproxy 语法做检查

haproxy -f /etc/haproxy.cfg -sf `cat /var/run/haproxy.pid` # reload

Haproxy启动参数

-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列:

负载均衡算法

1.roundrobin：基于权重进行的轮叫算法，在服务器的性能分布经较均匀时这是一种最公平的，最合量的算法。

2.static-rr：也是基于权重时行轮叫的算法，不过此算法为静态方法，在运行时调整其服务权重不会生效。

3.source：是基于请求源IP的算法，此算法对请求的源IP时行hash运算，然后将结果与后端服务器的权理总数相除后转发至某台匹配的后端服务器，这种方法可以使用一个客户端IP的请求始终转发到特定的后端服务器。

4.leastconn：此算法会将新的连接请求转发到具有最少连接数目的后端服务器。在会话时间较长的场景中推荐使用此算法。例如数据库负载均衡等。此算法不适合会话较短的环境，如基于http的应用。

5.uri：此算法会对部分或整个URI进行hash运算，再经过与服务器的总权重要除，最后转发到某台匹配的后端服务器上。

6.uri\_param：此算法会椐据URL路径中的参数时行转发，这样可以保证在后端真实服务器数量不变时，同一个用户的请求始终分发到同一台机器上。

7.hdr()：此算法根据httpd头时行转发，如果指定的httpd头名称不存在，则使用roundrobin算法进行策略转发。

cookie：表示充许向cookie插入SERVERID,每台服务器的SERVERID可以下面的server关键字中使用cookie关键字定义。

option httpchk：此选项表示启用HTTP的服务状态检测功能。HAProxy作为一个专业的负载均衡器，它支持对backend部分指定的后端服务节点的健康检查，以保证在后端的backend中某个节点不能服务时，把从frontend端进来的客户端请求分配至backend中其他健康节点上，从而保证整体服务的可用性。

Cookie设置

负载均衡算法为轮询调度，会话保持实现方式为cookie识别，插入cookie

listen web\_proxy \*:80

appsession JSESSIONID len 52 timeout 3h

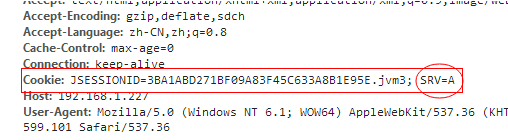
#插入cookie的方式

cookie SRV insert indirect nocache

#后端真实服务

server webA 192.168.1.226:8081 cookie A check

server webB 192.168.1.246:8888 cookie B check



cookie被加入了SRV=A

会话保持的流程

1.客户端首次请求，经过haproxy到web服务端时，web服务端set-cookie并响应到haproxy

2.haproxy在cookie后插入SRV=A，并响应客户端

3.客户端第二次请求，经过haproxy时，haproxy将srv后缀去掉，然后请求服务端

总结

该方案解决的问题

1.负载均衡，并解决web服务的单点故障

2.会话保持

存在的缺点

1.web服务器的session保存存在单点故障，即其中一台web服务器宕机之后，存储在上面的session也会丢失

2.负载均衡服务器存在单点故障

配置实战

禁止以IP访问

ACL: acl host\_ip hdr\_dom(host) -i 192.168.1.12

策略: block if host\_ip

效果



禁止以字符串开头的URL

ACL: acl path\_product path\_reg -i ^/product

策略：block if path\_product

效果：



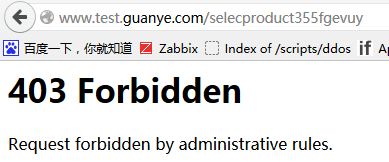
禁止包涵字符串的URL

ACL: acl url\_product url\_sub -i product355

策略：block if path\_product || host\_ip || url\_product

效果：







注意：要deny多个包涵指定字符串的请求，ACL不能写成acl url\_product url\_sub -i (product355|select|hahahahah)，因为此ACL将不会做任何匹配完全无效。