

haproxy 日志详解

官网的文档

<http://cbonte.github.io/haproxy-dconv/configuration-1.4.html#8>

haproxy日志级别有 tcp跟http 2种 现在来讲例子吧 因为我线上没tcp的日志 我就用官网的来举例吧 （相当于翻译下）

```
>>> Feb  6 12:12:56 localhost \
      haproxy[14387]: 10.0.1.2:33313 [06/Feb/2009:12:12:51.443] fnt \
      bck/srv1 0/0/5007 212 -- 0/0/0/0/3 0/0
```

Field	Format	Extract from the example above
1	process_name '[' pid ':'	haproxy[14387]:
2	client_ip ':' client_port	10.0.1.2:33313
3	'[' accept_date '']	[06/Feb/2009:12:12:51.443]
4	frontend_name	fnt
5	backend_name '/' server_name	bck/srv1
6	Tw '/' Tc '/' Tt*	0/0/5007
7	bytes_read*	212
8	termination_state	--
9	actconn '/' feconn '/' beconn '/' srv_conn '/' retries*	0/0/0/0/3
10	srv_queue '/' backend_queue	0/0

Feb 6 12:12:56 这个不用多讲吧

localhost 这个大家也知道

haproxy[14387] haproxy pid

10.0.1.2:33313 请求源地址 源端口

[06/Feb/2009:12:12:51.443] haproxy 接受请求的时间

fnt 定义的frontend 名称

bck/srv1 backend名称/后端server名称

0/0/5007 Tw/Tc/Tt 时间 #时间都是ms级别的

Tw 是指在请求队列中等待的总时间（T代表总共 w代表waiting） 如果是-1的话 代表客服端主动断开请求

Tc 是指用来连接后端server tcp建联总时间（T代表总共 c代表connect） 如果是-1的话 代表客服端在tcp建

联的时候断开请求

Tt 是指haproxy完成整个请求的总时间（包括接受请求 向后端发起 接收后端的内容 和重试的所有时间）
(T代表总共 t也代表总共)

212 日志记录时 所以经过haproxy 传输的字节数

-- haproxy终止状态 （不知道是干啥的）

0/0/0/0/3 actconn/feconn/beconn/srv_conn/retries

actconn 是指会话记录过程中进程并发连接数（好像跟系统文件描述符相关）

feconn 是指会话记录过程中frontend中的并发连接数（frontend 级别的）

beconn 是指会话记录过程中backend中已经建立握手的并发连接数（backend 级别的）

srv_conn 是指会话记录过程中活跃的并发连接数

retries是指会话记录过程中重试后端server的次数

0/0 srv_queue/backend_queue

srv_queue 处理改请求时 server端的队列正在处理的连接数

backend_queue 处理改请求是 backend队列正在处理的连接数

TCP格式的日志 就先搞到这里吧 下面用我们搞搞 http格式的吧 我去找我们线上的一个日志吧

先说下我们haproxy的日志过滤配置 下面会提到

康叔	hui.fan		
4 客服 495	IMS2.0升级通知2013/06/27		14:44
capture request	header Host	len 40	
capture request	header User-Agent	len 16	
capture request	header Referer	len 40	14:37

日志格式如下：

```
Jun 26 20:55:18 localhost.localdomain haproxy[6255]: 112.3.150.26:19538 [1372251318.135/1372251318.267] http-in fc_backend/CMN-NJ-S-307 0/0/0/111/132 302 436 - - - - - 9670/9670/7878/1496/0 0/0 {sz.photo.store.qq.com|Mozilla/4.0 (com|http://user.qzone.qq.com/2644932027/info)} "GET /rurl2=ce4f79c2e7d3420f44fea1b5364e68ca708e21d984980ab51cc47bddd33bea68df2ca116003656e9b06b9f535b485527b73ce5420f16ce1672878028ba568cc37e28a77e78211dae164a0a9b3f2f1acd9d0f3516 HTTP/1.1"
```

先来个官网的接受吧

Field	Format	Extract from the example above
1	process_name '[' pid ']:'	haproxy[14389]:
2	client_ip ':' client_port	10.0.1.2:33317
3	'[' accept_date ']'	[06/Feb/2009:12:14:14.655]
4	frontend_name	http-in
5	backend_name '/' server_name	static/srv1
6	Tq '/' Tw '/' Tc '/' Tr '/' Tt*	10/0/30/69/109
7	status_code	200
8	bytes_read*	2750
9	captured_request_cookie	-
10	captured_response_cookie	-
11	termination_state	----
12	actconn '/' feconn '/' beconn '/' srv_conn '/' retries*	1/1/1/1/0
13	srv_queue '/' backend_queue	0/0
14	'{' captured_request_headers* '}'	{haproxy.1wt.eu}
15	'{' captured_response_headers* '}'	{}
16	''' http_request '''	"GET /index.html HTTP/1.1"

下面开始详解吧

Jun 26 20:55:18 这个你懂的

localhost.localdomain 你也懂的

haproxy[6255]: haproxy pid

112.3.150.26:19538 客户端源地址 源端口

[1372251318.135/1372251318.267] 请求开始时间/请求结束时间 时间戳的格式 你懂的

http-in frontend名称

fc_backend/CMN-NJ-S-307 backend名称/后端server名称

0/0/0/111/132 Tq/Tw/Tc/Tr/Tt （时间跟上面一样 ms级别的）

Tq 等待客户端发送完整的http请求的时间 （-1 表示在接收http响应之前断开连接）

Tw 请求在队列中等待的时间 （-1 表示请求到达队列之前 断开连接）

Tc 与后端server建联的总时间 包括重试时间 （-1 表示请求与后端建联过程中 断开连接）

Tr 等待后端server 发送一个完整http响应总时间 不包括实体内容 （-1 代表在等待发送响应之前 断开连接）

Tt 从接受请求到关闭连接的总时间

302 HTTP响应代码

436 接收到该请求返回的数据大小（包括HTTP头） 单位是字节

- 请求的cookie

官网的用法

```
capture cookie <name> len <length>
```

Capture and log a cookie in the request and in the response.

- 响应的cookie

官网的用法

```
capture cookie <name> len <length>
```

Capture and log a cookie in the request and in the response.

---- 请求终止状态

9670/9670/7878/1496/0 actconn/feconn/beconn/srv_conn/retries

actconn 是指会话记录过程中进程并发连接数（好像跟系统文件描述符相关）

feconn 是指会话记录过程中frontend中的并发连接数（frontend 级别的）

beconn 是指会话记录过程中backend中已经建立握手的并发连接数（backend 级别的）

srv_conn 是指会话记录过程中活跃的并发连接数

retries 是指会话记录过程中重试后端server的次数

0/0 srv_queue/backend_queue

srv_queue 处理改请求时 server端的队列正在处理的连接数

backend_queue 处理改请求是 backend队列正在处理的连接数

```
{sz.photo.store.qq.com|Mozilla/4.0 (com|http://user.qzone.qq.com/2644932027/info} {  
captured_request_headers '}
```

过滤请求中的header头 配置文件已经定义

康叔	hui.fan		
4 客服 495	IMS2.0升级通知2013/06/27		14:44
capture request	header Host	len 40	
capture request	header User-Agent	len 16	
capture request	header Referer	len 40	14:37

host 40字节 User_Agent 16字节 Referer 40字节

官网的用法

```
capture request header <name> len <length>
```

Capture and log the first occurrence of the specified request header.

例子

Example:

```
capture request header Host len 15
capture request header X-Forwarded-For len 15
capture request header Referrer len 15
```

```
"GET /url2=ce4f79c2e7d3420f44fea1b5363ce5420f16ce1672878028ba568cc37e28a77ed9d0f3516
HTTP/1.1" "" http_request ""
```

OK 先写到这里吧 总结下 其实这些东西都是从haproxy官网手册里面 翻译然后总结出来的 我英文不好 如果有啥错误 可以指正下

先吃饭了 下次有时间把haproxy的其他东西也写写吧