



链滴

(超级详细) 使用 GoAccess 分析 Nginx 日志的安装和配置

作者: [Leif160519](#)

原文链接: <https://ld246.com/article/1568098765125>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>

官网地址: https://goaccess.io/

官方 github 地址: https://github.com/goaccess

官方 demo 地址: https://rt.goaccess.io/?20190828082924

</p>

<h2 id="一-前言">一、前言</h2>

<p>GoAccess 在 GitHub 的源码库上已经给出多种安装方式, 源码编译, apt-get, docker 等, 挑一种自己喜欢的就可以了。这里需要说明的是, 果想要使用某些特性如地理位置, ssl 连接等的话, 则需要编译安装, 可以先将依赖装好, 再编译安装即可。</p>

<h2 id="二-安装">二、安装</h2>

<h3 id="1-centos编译安装">1.centos 编译安装</h3>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">#安装依赖</span></span><span class="highlight-line"><span class="highlight-cl">yum -y install gcc make GeolIP-devel ncurses-devel</span></span><span class="highlight-line"><span class="highlight-cl">wget https://tar.goaccess.io/goaccess-1.3.tar.gz</span></span><span class="highlight-line"><span class="highlight-cl">tar -xzf goaccess-1.3.tar.gz</span></span><span class="highlight-line"><span class="highlight-cl">cd goaccess-1.3/</span></span><span class="highlight-line"><span class="highlight-cl">./configure --enable-utf8 --enable-geoip=legacy</span></span><span class="highlight-line"><span class="highlight-cl">make</span></span><span class="highlight-line"><span class="highlight-cl">make install</span></span></code></pre>
```

<blockquote>
<p>执行 configure 配置, 根据需要选用配置项, 可以通过 <code>./configure --help</code> 看可选参数。</p>
</blockquote>

<p>配置选项</p>

<blockquote>
<p>可以使用多个选项配置 GoAccess。有关配置选项的完整最新列表, 请运行 <code>./configure -help</code></p>
</blockquote>

<p><code>--enable-debug</code>: 使用调试符号进行编译并关闭编译器优化。</p>

<p><code>--enable-utf8</code>: 编译具有广泛的字符支持。</p>

<p><code>--enable-geoip=<legacy|mmdb></code>: 使用 GeoLocation 支持进行编译 MaxMind 的 GeolIP 是必需的。<code>legacy</code> 将使用原始的 GeolIP 数据库。<code>mmdb</code> 将使用增强的 GeolIP2 数据库。</p>

-
<p><code>--enable-tcb=<memhash|btree></code>: 与 Tokyo Cabinet 存储支持一起译。<code>memhash</code> 将利用 Tokyo Cabinet 的内存中哈希数据库。<code>btree</code> 将利用 Tokyo Cabinet 的磁盘 B + Tree 数据库。</p>

<p><code>--disable-zlib</code>: 在 B + Tree 数据库上禁用 zlib 压缩。</p>

<p><code>--disable-bzip</code>: 禁用 B + Tree 数据库上的 bzip2 压缩。</p>

<p><code>--with-getline</code>: 动态扩展行缓冲区以解析整行请求, 而不是使用 4096 的固定大小缓冲区。</p>

<p><code>--with-openssl</code>: 使用 OpenSSL 编译 GoAccess, 支持其 WebSocket 服务。</p>

<blockquote>
<p>不需要走 https 的话可以去掉 --with-openssl 选项</p>
</blockquote>
<h3 id="2-centos包管理器安装">2.centos 包管理器安装</h3>
<p>如果使用 <code>yum install goaccess</code> 直接安装, 需要先安装 epel 源。</p>
<pre><code class="highlight-chroma">sudo yum install -y epel-release</code></pre>
<h4 id="安装排错">安装排错</h4>
<p>由于官网并未提供安装之前的依赖包, 故安装之后会出现如下错误

FAQ: </p>
<p>根据错误提示解决依赖关系: </p>
<p>configure: error:</p>
<p>Missing development files for the GeolIP library</p>
<p>安装需要的软件包</p>
<pre><code class="highlight-chroma">yum install -y GeolIP-devel</code></pre>
<p>再次运行预编译命令并根据错误提示解决依赖关系: </p>
<p>configure: error:</p>
<p>Missing development libraries for ncursesw</p>
<p>安装需要的软件包: </p>
<pre><code class="highlight-chroma">yum install -y ncurses-devel</code></pre>
<p>附上各个平台依赖包列表

</p>
<h3 id="3-ubuntu包管理器安装">3.ubuntu 包管理器安装</h3>
<pre><code class="highlight-chroma">apt-get install goaccess</code></pre>
<blockquote>
<p>上述方法安装的 goaccess 只是 1.2 版本的, 故需要升级, 方法见下</p>

</blockquote>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> echo "deb https://deb.goaccess.io/ $(lsb_release -cs) main" | sudo tee -a /etc/apt/sources.ist.d/goaccess.list
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> wget -O - https://eb.goaccess.io/gnugpg.key | sudo apt-key add -
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> sudo apt-get update
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> sudo apt-get install goaccess
```

```
</span> </span> </code> </pre>
```

4.ubuntu编译安装

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> #安装依赖库
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> apt-get install libursesw5-dev libssl-dev libgeoip-dev
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> #安装编译依赖的包
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> apt-get install gcc make
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> wget https://tar.goaccess.io/goaccess-1.3.tar.gz
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> tar -xzvf goaccess-1.3.tar.gz
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> cd goaccess-1.3/
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ./configure --enable-utf8 --enable-geoip=legacy
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> make
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> make install
```

```
</span> </span> </code> </pre>
```

三、检测 goaccess 版本

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> root@ubuntu:~# goaccess --version
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> GoAccess - 1.3.
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> For more details visit: http://goaccess.io
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Copyright (C) 200-2016 by Gerardo Orellana
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Build configure arguments:
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> --enable-utf8
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> --enable-geoip=legacy
```

```
</span> </span> </code> </pre>
```

若添加了 openssl 编译参数的话, 效果如下:

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> [root@ubuntu]~# goaccess --version
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> GoAccess - 1.3.
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> For more details visit: http://goaccess.io
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Copyright (C) 200-2016 by Gerardo Orellana
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Build configure arguments:
```

uments:

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --enable-utf8
</span></span><span class="highlight-line"><span class="highlight-cl"> --enable-geoip=
egacy
</span></span><span class="highlight-line"><span class="highlight-cl"> --with-openssl
</span></span></code></pre>
```

四、配置

1.首先机器上得有nginx环境

2.goaccess配置

选择一个路径存放 goaccess 的配置文件，这里我选择 `/usr/local/src/goaccess` 目录

新建 goaccess.conf，内容如下：

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">time-format %H:%M:%S
</span></span><span class="highlight-line"><span class="highlight-cl">date-format %d/
b/%Y
</span></span><span class="highlight-line"><span class="highlight-cl">log-format %h %^
%d:%t %^] "%r" %s %b "%R" "%u"
</span></span><span class="highlight-line"><span class="highlight-cl">real-time-html tru
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">port &lt;port&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">ssl-cert &lt;cert.cr
&gt;
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">ssl-key &lt;priv.ke
&gt;
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">ws-url wss://&lt;y
ur-domain&gt;:&lt;port&gt;
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">output /usr/local/
rc/goaccess/index.html
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">log-file /usr/local/
rc/goaccess/goaccess.log
```

```
</span></span></code></pre>
```

参数解释

-

- real-time-html 用来使用实时刷新特性；

- port 是用来和浏览器通信的，选一个没被占用的就行（别忘了在防火墙里开启端口）；

- 如果你不走 https 的话，ssl-cert, ssl-key, ws-url 都不是必需的；

- output: 存放 goaccess 的站点目录；

- log-file:goaccess 的日志文件，后来发现运行后里面都是空的，暂时不知道为啥；

需要注意的几点

三个 format 的设置要与 nginx 的设置一致，当然如果你像我这样根本没改过 nginx 的日志格式的话用这个就行了；

nginx 中的日志格式如下：

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">log_format main '$remote_addr - $remote_user [$time_local] "$request" '
</span></span><span class="highlight-line"><span class="highlight-cl">'$status $body_b
tes_sent "$http_referer" '
</span></span><span class="highlight-line"><span class="highlight-cl">'"$http_userAgen
"$http_x_forwarded_for";
</span></span></code></pre>
```

3.nginx 配置

这里我是用的 80 映射的 7890 端口，你可以根据实际情况选择端口号

http 方式

```

<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">user nginx;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">worker_processes
uto;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">error_log /var/log
nginx/error.log;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">pid /run/nginx.pid

</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">include /usr/share
nginx/modules/*.conf;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">events {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    worker_connect
ons 1024;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">}
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">http {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    log_format ma
n '$remote_addr - $remote_user [$time_local] "$request" '
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        '$sta
us $body_bytes_sent "$http_referer" '
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        "$ht
p_user_agent" "$http_x_forwarded_for";
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    access_log /var
log/nginx/access.log main;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    sendfile
n;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    tcp_nopush
on;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    tcp_nodelay
on;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    keepalive_time
ut 65;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    types_hash_ma
_size 2048;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    include
etc/nginx/mime.types;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    default_type
application/octet-stream;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    include /etc/ng
nx/conf.d/*.conf;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    upstream goac
ess {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        server 127.0.0
1:7890;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    server {

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> listen 80
default_server;
</span></span><span class="highlight-line"><span class="highlight-cl"> listen [:::
0 default_server;
</span></span><span class="highlight-line"><span class="highlight-cl"> server_name
goaccess;
</span></span><span class="highlight-line"><span class="highlight-cl"> #root /u
r/share/nginx/html;
</span></span><span class="highlight-line"><span class="highlight-cl"> root /us
/local/src/goaccess;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> # Load confi
uration files for the default server block.
</span></span><span class="highlight-line"><span class="highlight-cl"># include /etc
nginx/default.d/*.conf;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> location / {
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> error_page 4
4 /404.html;
</span></span><span class="highlight-line"><span class="highlight-cl"> location =
40x.html {
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> error_page 5
0 502 503 504 /50x.html;
</span></span><span class="highlight-line"><span class="highlight-cl"> location =
50x.html {
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>更新 nginx 配置: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">nginx -s reload
</span></span></code></pre>
<blockquote>
<p>注意: 若网页打开 <code>403</code> 报错的话, 请查看站点目录是否有足够的权限访问, 者将上面的 <code>nginx.conf</code> 文件中 <code>user</code> 从 <code>nginx</code> 改为 <code>root</code></p>
</blockquote>
<h4 id="https方式">https 方式</h4>
<p>在 443 配置下添加如下内容: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">location /goaccess {
</span></span><span class="highlight-line"><span class="highlight-cl"> root /usr/local
src;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>别忘了在防火墙中添加 <code>7890</code> 端口, 若是云服务器得在安全策略中开通 <code>tcp:7890</code></p>

```

4.运行

```
goaccess --config-file /usr/local/src/goaccess/goaccess.conf /var/log/nginx/access.log
```


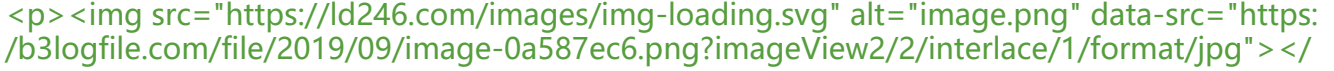

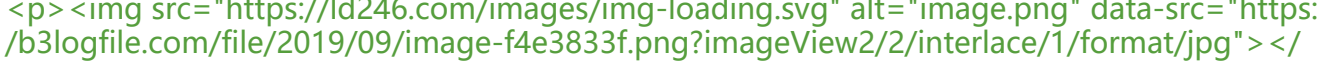
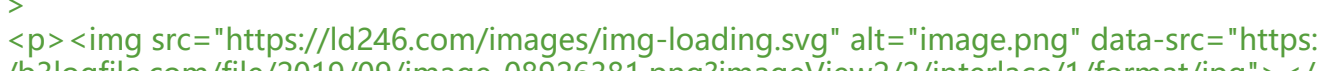
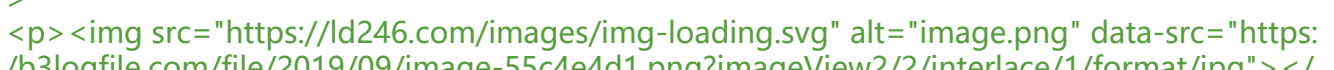
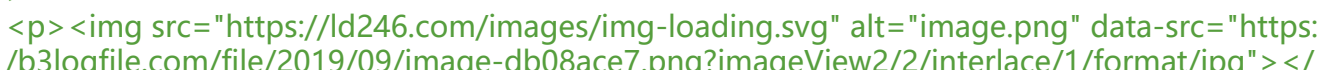

说明

- config-file : 指定配置文件, 不写默认使用/etc/goaccess.conf

- /var/log/nginx/access.log : 指定 nginx 的日志

效果

可以访问本站监控地址: [https://leif.fun/goaccess](https://ld246.com/forward?goto=https%3A%2F%2Fleif.fun%2Fgoaccess)



五、设置开机自启

<p>设置 <code>goaccess</code> 开机自启

一般来讲，设置开机自启可以使用 <code>systemctl</code> 或者 <code>Supervisor</code> 自启文件如下。</p>

<p>supervisor</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[program:goaccess]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">command=/usr/local/bin/goaccess --config-file=/usr/local/src/goaccess/goaccess.conf /var/log/nginx/access.lo
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">directory=/usr/local/src/goaccess/
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">user=root
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">stdout_logfile=/var/log/goaccess.log
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">autostart=true
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">autorestart=true
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">redirect_stderr=true
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">stopsignal=QUIT
```

```
</span></span></code></pre>
```

<p>systemctl</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">[Unit]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">Description=GoAccess
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">After=network.target
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">[Service]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">ExecStart=/usr/local/bin/goaccess --config-file=/usr/local/src/goaccess/goaccess.conf /var/log/nginx/access.lo
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">Restart=on-failure
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">[Install]
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">WantedBy=multi-user.target
```

```
</span></span></code></pre>
```

<p>细心的同学会发现，官网提供的示例上包含了平均响应时，总共响应时和最高响应时，但是依据上述教程操作下来看并没有出现以下内容

 那是因为 nginx 中并未配置有关响应时间的日志格式，所以需要在 <code>nginx.conf</code> 中增两行配置：</p> ``` <code class="highlight-chroma">log_format main '$remote_addr - $remote_user [$time_local] "$request" ' ``` ``` '$status $body_bytes_sent "$http_referer" ' ``` ``` "$http_protocol $http_user_agent" "$http_x_forwarded_for" ' ``` ``` '$connection $upstream_addr ' ``` ``` '$upstream_response_time $request_time'; ``` ``` </code></pre> ``` 原文链接: [\(超级详细\) 使用 GoAccess 分析 Nginx 日志的安装和配置](#)

<p>格式新增过后，goaccess 配置文件中有关日志格式的部分也需要更改，但是有的同学并不知道置中的每个参数代表什么意义，这里我具体贴出来</p>

<blockquote>

<p>log-format 与 access.log 的 log_format 格式对应，每个参数以空格或者制表符分割</p>

</blockquote>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">%t 匹配time-format格式的时间字段
</span></span><span class="highlight-line"><span class="highlight-cl">%d 匹配date-format格式的日期字段
</span></span><span class="highlight-line"><span class="highlight-cl">%h host(客户端地址，包括ipv4和ipv6)
</span></span><span class="highlight-line"><span class="highlight-cl">%r 来自客户端的请求行
</span></span><span class="highlight-line"><span class="highlight-cl">%m 请求的方法
</span></span><span class="highlight-line"><span class="highlight-cl">%U URL路径
</span></span><span class="highlight-line"><span class="highlight-cl">%H 请求协议
</span></span><span class="highlight-line"><span class="highlight-cl">%s 服务器响应的状态码
</span></span><span class="highlight-line"><span class="highlight-cl">%b 服务器返回的内容大小
</span></span><span class="highlight-line"><span class="highlight-cl">%R HTTP请求头的Referer字段
</span></span><span class="highlight-line"><span class="highlight-cl">%u 用户代理的HTTP请求报头
</span></span><span class="highlight-line"><span class="highlight-cl">%D 请求所花费的时间，单位微秒
</span></span><span class="highlight-line"><span class="highlight-cl">%T 请求所花费的时间，单位秒
</span></span><span class="highlight-line"><span class="highlight-cl">%^ 忽略这一字段
</span></span></code></pre>
```

<p>但是，这样的方法太过复杂了，我们可以使用 nginx2goaccess.sh 脚本将 nginx 日志格式格式为 goaccess 能识别的日志格式，nginx2goaccess.sh 脚本内容在 https://github.com/stockrt/nginx2goaccess/blob/master/nginx2goaccess.sh </p>

<p>nginx2goaccess.sh</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">log_format="$1"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"># Usage
</span></span><span class="highlight-line"><span class="highlight-cl">if [[ -z "$log_format" ]]; then
</span></span><span class="highlight-line"><span class="highlight-cl">    echo "Usage: $0 '&lt;log_format&gt;'"
</span></span><span class="highlight-line"><span class="highlight-cl">    exit 1
</span></span><span class="highlight-line"><span class="highlight-cl">fi
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"># Variables map
</span></span><span class="highlight-line"><span class="highlight-cl">conversion_table=
time_local,%d:%t_%^
</span></span><span class="highlight-line"><span class="highlight-cl">host,%v
</span></span><span class="highlight-line"><span class="highlight-cl">http_host,%v
</span></span><span class="highlight-line"><span class="highlight-cl">remote_addr,%h
</span></span><span class="highlight-line"><span class="highlight-cl">request_time,%T
</span></span></code></pre>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">request_method,
m
</span></span><span class="highlight-line"><span class="highlight-cl">request_uri,%U
</span></span><span class="highlight-line"><span class="highlight-cl">server_protocol,
H
</span></span><span class="highlight-line"><span class="highlight-cl">request,%r
</span></span><span class="highlight-line"><span class="highlight-cl">status,%s
</span></span><span class="highlight-line"><span class="highlight-cl">body_bytes_sent,
b
</span></span><span class="highlight-line"><span class="highlight-cl">bytes_sent,%b
</span></span><span class="highlight-line"><span class="highlight-cl">http_referer,%R
</span></span><span class="highlight-line"><span class="highlight-cl">http_user_agent,
u"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"># Conversion
</span></span><span class="highlight-line"><span class="highlight-cl">for item in $conve
sion_table; do
</span></span><span class="highlight-line"><span class="highlight-cl">    nginx_var=${it
m%%,*}
</span></span><span class="highlight-line"><span class="highlight-cl">    goaccess_var=$
item##*,}
</span></span><span class="highlight-line"><span class="highlight-cl">    goaccess_var=$
goaccess_var//_}
</span></span><span class="highlight-line"><span class="highlight-cl">    log_format=${{
g_format/\${$nginx_var}/$goaccess_var}
</span></span><span class="highlight-line"><span class="highlight-cl">    log_format=${{
g_format/\${$nginx_var}/$goaccess_var}
</span></span><span class="highlight-line"><span class="highlight-cl">done
</span></span><span class="highlight-line"><span class="highlight-cl">log_format=$(ech
"$log_format" | sed 's/${[a-z]*}/%^/g')
</span></span><span class="highlight-line"><span class="highlight-cl">log_format=$(ech
"$log_format" | sed 's/${[a-z]*}/%^/g')
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"># Config output
</span></span><span class="highlight-line"><span class="highlight-cl">echo "
</span></span><span class="highlight-line"><span class="highlight-cl">- Generated goac
ess config:
</span></span><span class="highlight-line"><span class="highlight-cl">time-format %T
</span></span><span class="highlight-line"><span class="highlight-cl">date-format %d/
b/%Y
</span></span><span class="highlight-line"><span class="highlight-cl">log_format $log_f
rmat
</span></span><span class="highlight-line"><span class="highlight-cl">"
</span></span></code></pre>
<p>使用如下方法获取日志格式：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">sh nginx2goaccess.sh '&lt;log_format&gt;' #log_format为你nginx.conf中配置的日志格式
</span></span></code></pre>
<p>如：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">bash nginx2goaccess.sh '$remote_addr - $remote_user [$time_local] "$request" $status $
ody_bytes_sent "$http_referer" "$http_user_agent" "$http_x_forwarded_for" $connection $ups
ream_addr $upstream_response_time $request_time'
</span></span></code></pre>

```

<p>会得到三个格式</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> time-format %T
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> date-format %d/
b/%Y
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> log_format %h -
^ [%d:%t %^] "%r" %s %b "%R" "%u" "%^" %^ %^ %^ %T
</span> </span> </code> </pre>
```

<p>将上述格式填写到 goaccess 配置文件中即可</p>

<p>先重启 <code>nginx</code> 之后再重启 <code>goaccess</code> 即可

 </p>

<h2 id="七-参考">七、参考</h2>

<p>https://www.jianshu.com/p/3b147fdbac8

https://www.jianshu.com/p/b134995ae16c

https://blog.51cto.com/13594742/2406665

https://www.cnblogs.com/erbiao/p/221671.html </p>