**CentOS下使用AWStats分析Nginx访问日志**

AWStats是在Sourceforge上发展很快的一个基于Perl的WEB日志分析工具。

    它可以统计您站点的如下信息：

* 访问量（UV），访问次数，页面浏览量（PV），点击数，数据流量等
* 精确到每月、每日、每小时的数据
* 访问者国家
* 访问者IP
* Robots/Spiders的统计
* 访客持续时间
* 对不同Files type的统计信息
* Pages-URL的统计
* 访客操作系统浏览器等信息
* 其它信息（搜索关键字等等）

**下面是AWStats分析Nginx日志的操作步骤：**

**一、awstats需要perl的支持,安装perl**

[root@admin ~]# yum install –y perl\*

**二、配置nginx日志格式**

    修改nginx.conf的日志格式，不然awstats无法分析。

|  |
| --- |
| log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  access\_log /var/log/nginx/access.log access; |

    注意，日志格式里的双引号不能漏了且每个参数之间是一个空格分隔，因为只要有细微的差别，awstats就无法读取日志。

    改好后，使nginx重读配置文件：

|  |  |
| --- | --- |
|  | [root@admin ~]#  nginx -t && nginx -s reload |

**三、自动切割nginx日志**

    每天晚上23点59分定时执行一个shell脚本来切割nginx日志。脚本内容如下：

|  |
| --- |
| #!/bin/bash  #  # Filename:    nginxCutLog.sh  # Author:      Qicheng  # Website:     http://qicheng0211.blog.51cto.com/  # Description: 切割nginx日志  # Notes:       设置crontab，每天23点59分定时执行  #  ROOT\_UID=0  if [ "$UID" -ne "$ROOT\_UID" ];then      echo "Error: 必须以root用户运行此程序！"      exit 1  fi    nginx\_logs\_dir="/var/log/nginx"  nginx\_pid\_file="/var/run/nginx.pid"  # 切割后的日志文件名，例如access\_20141022.log  nginx\_log\_today="$nginx\_logs\_dir/access\_`date +%Y%m%d`.log"    [ -f "$nginx\_log\_today" ] && exit 1  mv $nginx\_logs\_dir/access.log $nginx\_log\_today  # 给nginx发送USR1信号，使重新打开新的access.log日志文件  [ -f $nginx\_pid\_file ] && /bin/kill -USR1 $(cat $nginx\_pid\_file) |

    设置crontab：

|  |  |
| --- | --- |
|  | 59 23 \* \* \* /bin/bash /yourscriptpath/nginxCutLog.sh |

**四、安装awstats**

|  |
| --- |
| [root@admin ~]#  wget http://awstats.sourceforge.net/files/awstats-7.0.tar.gz  [root@admin ~]#  tar -zxvf awstats-7.0.tar.gz  [root@admin ~]#  mv awstats-7.0 /usr/local/awstats  [root@admin ~]#  mkdir -p /var/lib/awstats |

**五、配置awstats**

    进入/usr/local/awstats/tools/目录，执行配置脚本awstats\_configure.pl：

|  |
| --- |
| [root@admin ~]#  cd /usr/local/awstats/tools/  [root@admin ~]# sudo chown -R root:root awstats        # 注意权限，否则会影响以后执行  [root@admin ~]# sudo chmod 711 awstats\_configure.pl    #把awstats\_configure.pl  权限修改 与 awstats\_buildstaticpages.pl 相同  [root@admin ~]#  ./awstats\_configure.pl |
|  |



    程序执行结束后，会在/etc/awstats/目录下生成你的配置文件。然后编辑配置文件，修改LogFile参数，跟日志切割脚本中的日志路径对应起来：

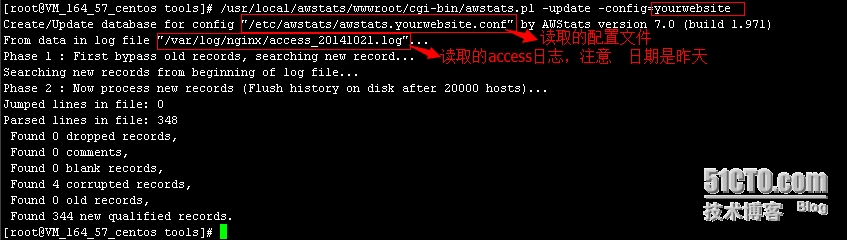
|  |  |
| --- | --- |
|  | LogFile="/var/log/nginx/access\_%YYYY-24%MM-24%DD-24.log" |

    注意，这里日期格式“%YYYY-24%MM-24%DD-24”，是指24小时之前的年月日，也就是昨天的日期。

    测试：

|  |  |
| --- | --- |
|  | [root@admin ~]#  /usr/local/awstats/wwwroot/cgi-bin/awstats.pl -update -config=yourwebsite |

    注意：-config=后面的参数是你在执行awstats\_configure.pl时输入的站点域名。



**六、配置awstats生成静态页面**

    利用awstats的工具将统计的结果生成静态文件：

|  |
| --- |
| [root@admin ~]#  mkdir -p /var/www/awstats  [root@admin ~]#  /usr/local/awstats/tools/awstats\_buildstaticpages.pl -update -config=yourwebsite -lang=cn -dir=/var/www/awstats -awstatsprog=/usr/local/awstats/wwwroot/cgi-bin/awstats.pl |

    注意：-config=后面的参数是你在执行awstats\_configure.pl时输入的站点域名；-dir=是统计结果静态文件的输出目录。

    设置crontab，每天凌晨00:01定时更新静态页面：

|  |  |
| --- | --- |
|  | 1 0 \* \* \* /usr/local/awstats/tools/awstats\_buildstaticpages.pl -update -config=yourwebsite -lang=cn -dir=/var/www/awstats -awstatsprog=/usr/local/awstats/wwwroot/cgi-bin/awstats.pl |

    一般站长都不愿随便让人知道自己站的真实流量，所以要把awstats统计结果页面进行密码保护，这里需要用到apache自带的工具htpasswd：

|  |
| --- |
| [root@admin ~]#  yum -y install httpd-tools  [root@admin ~]#  htpasswd -cd admin.pass admin  New password:  Re-type new password:  Adding password for user admin |

    把生成的密码文件admin.pass放到nginx主配置目录下（/etc/nginx/）。

    修改nginx.conf文件，添加红色字体部分

|  |
| --- |
| user nginx;  worker\_processes 1;  error\_log /var/log/nginx/error.log warn;  pid /var/run/nginx.pid;  events {  worker\_connections 1024;  }  http {  include /etc/nginx/mime.types;  default\_type application/octet-stream;  log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  access\_log /var/log/nginx/access.log access;  sendfile on;  #tcp\_nopush on;  keepalive\_timeout 65;  #gzip on;  upstream puppetmaster {  server 127.0.0.1:8080;  server 127.0.0.1:8090;  }  server {  listen 9000;  location / {  ##设置跳转到puppetmaster负载均衡  proxy\_pass http://puppetmaster;  proxy\_redirect off;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Client-Verify $ssl\_client\_verify;  proxy\_set\_header X-SSL-Subject $ssl\_client\_s\_dn;  proxy\_set\_header X-SSL-Issuer $ssl\_client\_i\_dn;  proxy\_buffer\_size 10m;  proxy\_buffers 1024 10m;  proxy\_busy\_buffers\_size 10m;  proxy\_temp\_file\_write\_size 10m;  proxy\_read\_timeout 120;  }  location ~ ^/awstats/ {  root /var/www;  index index.html;  access\_log off;  error\_log off;  charset gb2312;  auth\_basic "admin";  auth\_basic\_user\_file admin.pass;  }  location ~ ^/icon/ {  root /usr/local/awstats/wwwroot;  index index.html;  access\_log off;  error\_log off;  }  }  } |
|  |

最后使nginx重读配置文件，用浏览器查看统计结果 http://yourhostname:9000/awstats/awstats.yourwebsite.html

日志分析页面示例：



注：

### shell脚本“syntax error:unexpected end of file”解决方案

2011-08-28 12:53:18|  分类： [编程技术](http://renyongjie668.blog.163.com/blog/#m=0&t=1&c=fks_084067084084084065080086081095085080081067081085087) |  标签：[shell](http://renyongjie668.blog.163.com/blog/#m=0&t=3&c=shell)  [linux](http://renyongjie668.blog.163.com/blog/#m=0&t=3&c=linux)  [学习笔记](http://renyongjie668.blog.163.com/blog/#m=0&t=3&c=学习笔记)  [编程](http://renyongjie668.blog.163.com/blog/#m=0&t=3&c=编程)  |举报|字号 订阅

[下载LOFTER客户端](http://www.lofter.com/app?act=qbbkrzydb_20150408_01)

今天在写Shell时，运行时出现了这样的错误。  
git-sync-tree.sh\_temp: line 111: syntax error: unexpected end of file  
  
网上Google了一下，网上都是说从windows下脚本传到Linux上可能会出现这样的问题，是因为Windows和Linux下的行末结束符是不一样的，曾经写过一篇博客：[回车与换行的区别](http://www.51testing.com/?uid-225738-action-viewspace-itemid-213825)    当然，我今天遇到的不是这种情况导致的。  
  
1. 如果确实是这种情况，在windows下写好了Shell 但是在linux下用：  
sh -n [filesName]  检查语法总是出一个错误 syntax error:unexpected end of file  
原因如下:  
dos文件传输到unix系统时,会在每行的结尾多一个^M,在vi的时候,当你用如下命令：  
vi dos.txt  
:set fileformat=unix  
:w  
就会看到这些存在于每行结尾的^M符号，这个就是产生syntax error:unexpected end of file的原因  
解决方案：  
在vi下把这些^M都删除后即可。  
也可以使用Linux下的工具：**dos2unix**也可轻松将一个windows下的文本文件转化为Unix兼容的格式。  
  
2.我遇到的不是这样由于windows和Linux相互拷贝文件而导致的。这个是语法错误嘛，由于我这个shell脚本有点大，看了一阵子也没发现是那句话语法错误了，所以不得不用二分法来查找原因，不断注释一些代码，然后用sh -n test.sh来做语法检查，直到最后找到那一段或者哪一行代码引起的错误。我发现是下面这行代码引起的，你能看出其中的问题吗？^\_^  
[ -d /home/repo/${SPPATH} ] || { mkdir -p /home/repo/${SPPATH}; cd /home/repo/${SPPATH}; git init >> $GITLOG 2>&1 }  
嗯，其实我用花括号{}是想把几个命令组合起来在当前shell中执行，然后我犯了一个语法错误，在最后的一个命令后没有加分号（;）。将这行改为如下即可（添加一个最后的分号）：  
[ -d /home/repo/${SPPATH} ] || { mkdir -p /home/repo/${SPPATH}; cd /home/repo/${SPPATH}; git init >> $GITLOG 2>&1; }  
关于当前shell中执行一组命令，特别要注意的是，**在”{“的右边 和”}“的左边，至少要间隔一个以上的空格，而且每个命令都要以分号(;)作为结尾。**