**由浅入深学习Apache httpd原理与配置**

原贴：<http://jackcui.blog.51cto.com/11877206/1858989>

**文章防盗链地址：**[**http://jackcui.blog.51cto.com/**](http://jackcui.blog.51cto.com/)

**一、apache简介**：

    Apache HTTPD又可以简称为httpd或者Apache，它是Internet使用最广泛的web服务器之一，使用Apache提供的web服务器是由守护进程httpd，通过http协议进行文本传输，默认使用80端口的明文传输方式，当然，后来，为了保证数据的安全和可靠性，又添加了443的加密传输的方式，Apache提供的服务器又被称为：补丁服务器，原因很简单，它是一款高度模块化的软件，想要给它添加相应的功能只需添加相应的模块，让其Apache主程序加载相应的模块，不需要的模块也可以不用加载，保证了Apache的简洁，轻便，高效性，当出现大量访问一个服务器是可以使用多种复用模式，保证了服务器能快速回应客户端的请求，如MPM，端口复用技术。

**二、一次完整的http请求**

**1. http请求和响应过程：**

（1）建立连接：客户端向服务器建立连接，发送报文，包含相应的请求资源的方法和客户端能支持的协议，编码等数据。

（2）服务器决定是否接受请求。

（3）如果服务器同意建立连接时便要处理请求，其中包括访问资源，访问资源时需要相应的映射机制，把客户端的URL转化为本地目录下相应的文件

（4）服务器访问到相应的资源后构建响应报文

（5）发送响应报文，报文中包含相应的状态码，和数据报文

（6）服务器记录日志

（7）客户端接收数据

附示例：

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8 | [root@177cnode1 ~]# telnet vhost2.jack.com 80  Trying 192.168.20.128...  Connected to vhost2.jack.com.  Escape character is '^]'.  GET /index.html http/1.1  ====>使用GET方法请求服务器端的主页  host:192.168.20.128 ===>host字段不能缺少，但可以是空值，指定服务器的ip地址或域名也可以指定  端口号，请求报文以一个空行结尾，最后键入回车（两次回车）一般浏览器发送请求时会包括能接受的  编码方式，报文长度等 | |
| 1  2  3  4  5  6  7  8  9  10  11 | HTTP/1.1 200 OK   ===>回应报文的状态码  Date: Tue, 04 Oct 2016 03:48:47 GMT   ===>回应报文的其它字段  Server: Apache/2.4.6 (CentOS)  Last-Modified: Sun, 02 Oct 2016 08:41:47 GMT  ETag: "16-53dddcc3d4d35"  Accept-Ranges: bytes  Content-Length: 22  Content-Type: text/html; charset=UTF-8    this is vhost2's page    ===>网页内容  [root@177cnode1 ~]# |

**2. http两种连接方式：**

（1）短连接：非保持连接

（2）长连接：保持连接。

数量限制：多少个资源

时间限制：最长可以保持长连接多长时间

附示例：

  在相应的目录下面建立以.conf结尾的文件内容如下keepalive.conf文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | [root@177cnode1 conf.d]# pwd  /etc/httpd/conf.d  [root@177cnode1 conf.d]# cat keepalive.conf  keepalive on  keepalivetimeout 180  MaxKeepAliveRequests 1000 |

   上面配置中第一行启用了keepalive 功能，第二行设置建立连接后多少秒后断开连接，第三项设置了最大保持alive状态的请求数量为1000个

**3.  多种服务器状态码：**

  在服务器回应客户端时会回应相应的状态码，不同的状态码具有不同的含义

1xx：信息类

2xx：成功类

3xx：重定向类

4xx：客户端错误类

5xx：服务端错误类

**4. 多种客户端请求方法：**

GET、HEAD、POST、PUT、DELETE、OPTION、TRACE …

附：IANA机构对socket端口分配：

1-1024：众所周知端口，永久分配给固定的应用程序使用，属于特权端口，只有root有权使用

1024-4195：注册端口，要求略宽松，分配给某程序注册使用

41952-65535：客户端程序使用的随机端口，动态端口，又叫私有端口或随机端口

三、http的特性：

高度模块化：core module +modules

**1. 多路处理模块MPM：multipath processing modules**

MPM模块在httpd-2.4中是动态共享模块的，没有编译如主程序当中，httpd-2. 2中是静态编译入主程序当中的。在这些模型中，默认使用第一个prefork模型，第二个模型因为出错不以排查，因此使用较少，在第三个模型当中因为是比较新的功能，只有在httpd-2.4之后的版本才有的功能，所以使用较少，因为在企业使用时稳定才是王道，绝非功能越新越好

**prefork：**多进程模型，也是默认类型，采用预派生子进程方式，用单独的子进程来处理不同的请求，进程之间彼此独立。两级架构，主进程一个子进程若干个。

**几个常用选项（指令后面的#代表相应数量的数字）：**

这些选项直接在主配置文件中修改（httpd-2.2，包含2.2之前），在httpd-2.4之后的版本中mpm的配置文件（/etc/httpd/conf.modules/00-mpm.conf）中编写即可

ServerLimit # 服务器允许配置的进程数上限，在其它两种模型中因为是多级架构，所以和线程子进程的乘积等有关系

StartServers #  httpd服务在启动时要启动的子进程数量

MinspareServers # 在服务器启动StartsServer指定的子进程书量后每隔一秒创建一定数量的进程（是指数型的增加，第一次1个，第二次两个，第三次4个，一直增加到32个每秒后稳定增加），直到增加到MinSpareServer指定的数量

MaxSpareServers # 这个值是设置系统最大空闲进程数量，每当大于这个数量的值时，系统会kill掉一部分子进程，但是当这个值设置的小于MinSpareServer的值时，系统会自动将该值设置为最小空闲进程数量加一

MaxConnectionsPerchild # 这个值代表每个子进程最大能处理多少个连接之后将被系统kill掉（因为进程在使用过程当中可能有数据错误或无法以外的内存泄露等）如果设置为0则代表永不kill，这个值从httpd-2.4之后才开始叫这个名字

MaxRequestWorkers #最大处理并发请求数量的限制2.4之后的版本才有的指令，意义等同于MaxClient

MaxClient # MaxClients设定的是 Apache可以并发处理的请求，是对Apache性能影响最大的参数。其缺省值150是远远不够的，如果请求总数已达到这个值（可通过ps -ef|grep http|wc -l来确认），那么后面的请求就要排队，直到某个已处理请求完毕。这就是系统资源还剩下很多而HTTP访问却很慢的主要原因。虽然理论上这个值越大，可以处理的请求就越多，但Apache默认的限制不能大于256。ServerLimit指令无须重编译Apache就可以加大MaxClients。这个指令是httpd-2.2的

**worker：**多线程模型，为每个请求分配一个线程，三级架构，一个主进程，若干子进程，每个子进程管理若干个线程

ServerLimit

StartServers

MinSpareThreads

MaxSpareThreads

MaxRequestWorkers 应该设置为0，因为此时它已经不在相应用户请求，0表示不加限制，不kill

        ThreadsPerChild

**event：**事件驱动模型，每个进程处理多个请求有多个进程

ThreadsPerChild

MaxRequestWorkers

        AsyncRequestWorkerFactor

**3. httpd的功能特性：**

虚拟主机

反向代理

负载均衡

CGI：common Gateway Interface

…

**4. httpd的版本：**

httpd-1.3

httpd-2.0

httpd-2.2

httpd-2.4

在centos6.8中系统版本为httpd-2.2，centos7.0之后的版本自带的httpd版本为httpd-2.4

**四、程序环境：**

**1. 主程序：/usr/sbin/httpd**

**2. 通过正则表达式将注释行和空行去掉，下面是httpd-2.4版本配置文件示例：**

[root@7cnode2 ~]# grep -v -E -e"^[[:space:]]\*#" -e "^[[:space:]]\*$" httpd.conf

在此需要注意的是该配置文件中每一配置分为两部分，配置指令和值两部分，配置指令不分区大小写，值部分除了路径名是要区分大小写，其它部分也是不区分大小写的！

ServerRoot "/etc/httpd"  配置httpd使用哪个位置中使用相对路径查找配置文件，如指定错误日志时写到的就是一个相对路径，相对与此处的目录查找错误日志文件存放目录

Listen 80  监听端口，可以同时监听多个端口不过要注意设置selinux和防火墙，可以重复定义多次，监听多个端口

Include conf.modules.d/\*.conf 加载其它配置文件，类似于C语言的头文件的加载，此处可以使用相对路径也可以使用绝对路径，相对路径是相对于ServerRoot的路径，可以使用glob通配符，此处的配置文件是用来加载DSO（dynamic shared object）模块

User apache

Group apache 程序运行时的用户名和组名，在安装Apache时已经创建了系统账号和组账号，程序启动时是以root身份启动，执行完root特权的所有操作后（例如启动监听80端口，低于1024的端口的监听需要root才有权力执行）会以非特权用户执行程序

ServerAdmin root@localhost 管理员的邮箱地址，当httpd出问题时，联系该邮箱地址可以联系到管理员

ServerName www.jackcui.node2.com:80   ServerName给定了主机名和端口号，主要用于服务器辨识自己是什么名字，如果此处没有配置的话服务器会进行查询/etc/hosts文件，最后查询自己的DNS服务器直到将IP地址反解至主机名如果没有DNS服务器，此处也可以直接写自己的ip地址和端口号，强烈建议此处的配置不要省略，否者重启httpd服务时会很慢，因为会进行DNS反解析，这一个配置的作用主要是用来定义重定向时，服务器区分访问的域名是自己还是别的主机

<Directory />  关于目录的一些配置，有关目录的访问权限等都是在这里定义，可以使用基于URL定义访问权限，但要使用

<Location “”> …<Location>来定义

    AllowOverride none   AllowOverride参数就是指明Apache服务器是否去找.htacess文件作为配置文件，如果设置为none,那么服务器将忽略.htacess文件，如果设置为All,那么所有在.htaccess文件里有的指令都将被重写。对于AllowOverride，还可以对它指定如下一些能被重写的指令类型.

    Require all denied 对目录的授权此处为拒绝所有访问

</Directory>

DocumentRoot"/var/www/html"  定义目录的根位置类似于配置文件的根，不过此处是网页存放的根，在定义Directory目录时，那里的目录同样可以是绝对路径，也可以是相对于此处的路径

<Directory"/var/www">

    AllowOverride None

    Require all granted

</Directory>

<Directory"/var/www/html">

    Options Indexes FollowSymLinks

    AllowOverride None

    Require all granted

</Directory>

<IfModule dir\_module>  此处为仅在加载了模块dir\_module 后才执行，DirectoryIndex配置指令后面的值可以跟多个，先后顺序很关键，值的含义为URL访问时进入目录是应该寻找哪一个文件，多个时按顺序寻找，找不到第一个然后寻找第二个值

    DirectoryIndex index.html

</IfModule>

<Files ".ht\*">  对所有安全相关的敏感文件设置为禁止，如目录下的.htaccess，.htpasswd

    Require all denied

</Files>

ErrorLog"logs/error\_log"  设置错误日志的存放路径，这里是一个软链接，指向/var/log/httpd/的符号链接

LogLevel warn  设置日志级别，仅达到该级别才记录日志

<IfModulelog\_config\_module>

下面定义了多种日志的记录格式，可以人为的修改自己想要定义的格式，LogFormat后面的双引号具体定义了日志的格式，后面有一个名字，在使用该定义好的格式可以使用后面的别名代替，当然也可以在使用时直接用双引号加相应的格式，访问日志也可以在虚拟主机中单独定义

    LogFormat "%h %l %u %t\"%r\" %>s %b \"%{Referer}i\"\"%{User-Agent}i\"" combined

    LogFormat "%h %l %u %t\"%r\" %>s %b" common

    <IfModule logio\_module>

      LogFormat "%h %l %u %t\"%r\" %>s %b \"%{Referer}i\"\"%{User-Agent}i\" %I %O" combinedio

    </IfModule>

下面定义问日志，和使用相应的格式为combined

    CustomLog "logs/access\_log"combined

</IfModule>

<IfModule alias\_module>

ScriptAlias和Alias类似都是在此定义，两者的区别是ScriptAlias是作为服务器的运行文件，而不是发送到客户端的文件

    ScriptAlias /cgi-bin/"/var/www/cgi-bin/"

</IfModule>

<Directory"/var/www/cgi-bin">

    AllowOverride None

    Options None

    Require all granted

</Directory>

<IfModule mime\_module>

mime多媒体英特网邮件扩展，这个模块是用来指定内容元数据，选择HTTP响应的映射模式中的URI或文件的元数据值的内容。如mime-type中的类型有语言，字符集，编码方式

    TypesConfig /etc/mime.types

    AddType application/x-compress .Z

    AddType application/x-gzip .gz .tgz

    AddType text/html .shtml

    AddOutputFilter INCLUDES .shtml

</IfModule>

下面的设置向text/plain和text/html 资源的content-type报头中添加charset部分。

AddDefaultCharset UTF-8

<IfModulemime\_magic\_module>

    MIMEMagicFile conf/magic

</IfModule>

EnableSendfile on

IncludeOptionalconf.d/\*.conf

**3. httpd-2.2版本与httpd-2.4的差别：**

    绝大部分基本是相同的，只不过后者更加趋向于模块化，将主配置文件分割，便于配置和管理下面将说一下具体的差别

（1）ServerTokens指令设置服务器HTTP响应头字段的值。在2.2的版本中有，为了保护服务器免收黑客的攻击，应把服务器回应的消息显示的最不详细为好。

（2）下面是ServerTokens的一些可能的赋值：  
ServerTokens Prod 显示“Server: Apache”  
ServerTokens Major 显示 “Server: Apache/2″  
ServerTokens Minor 显示“Server: Apache/2.2″  
ServerTokens Min 显示“Server: Apache/2.2.17″  
ServerTokens OS 显示 “Server: Apache/2.2.17 (Unix)”  
ServerTokens Full 显示 “Server: Apache/2.2.17 (Unix) PHP/5.3.5″

（3）PidFilerun/httpd.pid 保存httpd运行时的进程id，同样是在2.2版本中有

（4）Keepalive定义，2.2版本放在了主配置文件当中，2.4版本主配置文件没有定义，管理员可以自己定义一个单独配置文件

（5）prefork的模块定义，在2.2版本中可以直接编辑主配置文件定义prefork模块的相关参数，2.4版本的在专门的模块配置文件当中定义

**4. 其它 配置文件：**

  下面的三个文件分别是主配置文件和辅助配置文件，以及模块配置文件，对主配置文件进行分割方便管理，在重启服务或者重新加载配置文件时会一并加载

/etc/httpd/conf/httpd.conf

/etc/httpd/conf.d/\*.conf

/etc/httpd/conf.modules.d/\*.conf

**模块的加载格式为：**

LoadModule  模块名  模块存放路径

UnitFile:/usr/lib/systemd/system/httpd.service   //Unit文件是rhel7之后的版本系统服务脚本启动文件

**模块文件目录：**

**/usr/lib64/httpd/modules/**

**5. 站点主服务器根目录默认：/var/www/html**

**6. 日志文件：/var/log/httpd/**

error\_log 错误日志

access\_log 访问日志

**7. 判断服务正常与否：**

ss -tnlp | grep “:80\>”

systemctl status httpd.service

**五、三种虚拟主机的配置**

**1. 基于IP地址的虚拟主机：**

   在同一台服务器上，有多个IP地址，每一个IP地址负责一台虚拟主机的绑定，每个主机的主机名不一样如www.vhost1.com  www.vhost2.com，使用较少，因为IP地址较为宝贵，而这种虚拟主机需要大量IP地址。

配置示例：

（1）  添加多个供虚拟主机使用的IP地址

|  |  |
| --- | --- |
| 1  2 | [root@cnode6\_8conf.d]# ip a |grep 192  //此时eth2有一个IP地址      inet 192.168.66.142/24 scope global eth2 |

#使用ip命令添加三个临时IP地址

|  |  |  |
| --- | --- | --- |
| 1  2  3 | [root@cnode6\_8conf.d]# ip addr add 192.168.66.143/24 dev eth2  [root@cnode6\_8conf.d]# ip addr add 192.168.66.144/24 dev eth2  [root@cnode6\_8conf.d]# ip addr add 192.168.66.145/24 dev eth2 | |
| 1  2  3  4 | [root@cnode6\_8conf.d]# ip a | grep 192  //通过查看多了3个IP地址      inet 192.168.66.142/24 scope global eth2      inet 192.168.66.143/24 scope globalsecondary eth2      inet 192.168.66.144/24 scope globalsecondary eth2 |

（2）添加虚拟主机的配置文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | [root@cnode6\_8conf.d]# pwd  /etc/httpd/conf.d  [root@cnode6\_8conf.d]# vim virtual.conf  <VirtualHost 192.168.66.143:80>      ServerName www.vhost1.com      DocumentRoot "/testdir/vhost1"      <Directory"/testdir/vhost1">               AllowOverride none               Allow from all               Order Allow,deny      </Directory>  </VirtualHost>    <VirtualHost 192.168.66.144:80>      ServerName www.vhost2.com      DocumentRoot "/testdir/vhost2"      <Directory"/testdir/vhost2">               AllowOverride none               Allow from all               Order Allow,deny      </Directory>  </VirtualHost> |

（3）修改/etc/hosts文件（此处不是必须的，因为这里没有DNS服务器解析域名，只好修改hosts文件以供测试！）

|  |  |
| --- | --- |
| 1  2  3  4 | [root@cnode6\_8conf.d]# grep "^192" /etc/hosts  168.66.143  www.vhost1.com  168.66.144  www.vhost2.com  168.66.145 www.vhost3.com |

（4）添加相应的目录和文件重启服务测试，添加的目录和文件都因该是配置文件定义的。这里省略这些步骤，测试结果应该为访问相应的域名，会被解析为相应的IP能访问到响应的网页

**2. 基于域名的虚拟主机：**

  在同一台服务器上面，仅有一个IP地址，使用不同的主机名访问不同的网页内容，在虚拟主机块定义上面需要使用NameVirtualHost声明监听的IP地址，使用较多。需要注意在httpd-2.4的版本中不需要使用NameVirtualHost关键字指定监听IP地址和端口号，其余部分没有变化

（1）修改配置文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | root@cnode6\_8conf.d]# pwd  /etc/httpd/conf.d  [root@cnode6\_8conf.d]# vim virtual.conf  NameVirtualHost 192.168.66.142:80 //如果要监听主机所有IP可以使用通配符 \*  <VirtualHost192.168.66.142:80>      ServerNamewww.vhost1.com      DocumentRoot "/testdir/vhost1"      <Directory"/testdir/vhost1">               AllowOverride none               Allow from all               Order Allow,deny      </Directory>  </VirtualHost>    <VirtualHost192.168.66.142:80>      ServerNamewww.vhost2.com      DocumentRoot "/testdir/vhost2"      <Directory"/testdir/vhost2">            AllowOverride none            Allow from all            Order Allow,deny      </Directory>  </VirtualHost> |

（2）修改/etc/hosts文件

|  |  |
| --- | --- |
| 1  2  3 | [root@cnode6\_8 conf.d]# grep 192 /etc/hosts  168.66.143 www.vhost1.com  168.66.143 www.vhost2.com |

（3）测试

 测试时访问不同的域名，虽然是被解析为相同的IP地址，但是能访问到不同的主页

**3. 基于不同端口的虚拟主机：**

  在同一IP，同一主机名下，使用监听不同端口，访问时需要加访问的端口。使用不多，一般用来做内网测试使用

（1）修改配置文件

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | [root@cnode6\_8conf.d]# cat virtual.conf  Listen 8080   //添加监听的端口  <VirtualHost 192.168.66.142:8080>      ServerName www.vhost1.com      DocumentRoot "/testdir/vhost1"      <Directory"/testdir/vhost1">            AllowOverride none            Allow from all            Order Allow,deny      </Directory>  </VirtualHost>    <VirtualHost 192.168.66.142:80>      ServerName www.vhost1.com      DocumentRoot "/testdir/vhost2"      <Directory"/testdir/vhost2">               AllowOverride none               Allow from all               Order Allow,deny      </Directory>  </VirtualHost> |

（2）修改/etc/hosts文件

|  |  |
| --- | --- |
| 1  2 | [root@cnode6\_8 conf.d]# grep 192 /etc/hosts  168.66.142 www.vhost1.com |

  测试要注意默认的端口可以不加，但是非80的端口访问时要手动添加，在访问相同的域名，不同的端口的地址时，同样可以得到不同的网页

**六、目录选项指令和站点其他页面定义**

**1.目录中常用的指令：**

（1）   options：用于定义资源的展示方式；后跟以空白字符分割的“选项”列表

（2）   AllowOverRide：httpd允许在网页文档的各目录下使用隐藏文件.htaccess来自个子的访问控制；此指令定义哪些指令可以再.htaccess中定义

（3）   Require：定义访问控制的用户、组或者IP等的访问

**2.站点主页面文件设置（主配置文件中）：**

DirectoryIndex  filename1 filename2 ...

在DirectoryIndex后面跟多个文件，在访问时每一个目录时默认的页面文件名字，当第一个文件没有找到时，去找第二个文件，文件顺序很重要！

**3.路径别名定义（主配置文件的别名定义模块中定义）**

Alias  /URL/ "/PATH/TO/SOME\_DIR/"

给一个Url定义一个路径别名通过URL可以映射到本地文件系统的目录

**4.status页面**

用于显示当前主机运行状态，子进程或线程运行快照

<Location /status>

       SetHandler server-status

       Require all granted

</Location>

ExtendedStatus{On|Off} 扩展状态设置，建议关闭，否则影响服务器性能，默认关闭的，没有这一行设置也行，这一行设置是全局设置，不能在虚拟主机中设置。

**七、用户认证和组认证**

   http协议的认证由http协议的实现，对自身进行认证。http的认证可以是明文形式或者是摘要算法形式。在认证时，服务器响应吗为401，拒绝客户端请求，并需要客户端填入账号和密码后进行认证。服务器做认证的账号是虚拟账号，不是服务器系统的账号，可以存放在特定的文件系统中，或者数据库中，连接数据库需要相应的数据库模块为httpd提供接口等支持。表单认证由服务器的应用程序进行认证

**1.http协议的用户认证**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <VirtualHost 192.168.66.142:80>  ServerName www.vhost.com  DocumentRoot"/vhosts/www1"      <Directory "/vhosts/www1">         Options None         AllowOverride None         Require all granted      </DIrectory>      <Directory"/vhosts/www1/admin">         OptionsNone         AllowOverrideNone         AuthTypebasic         AuthName"Admin Area, Enter your name/password"         AuthUserFile"/etc/httpd/conf/.htpasswd"         Requirevalid-user      </Directory>  </VirtualHost> |

使用命令生成文件：

认证文件：.htpasswd

htpasswd[OPTIONS]  /PATH/TO/HTPASSWD\_FILEUSERNAME [PASSWORD]

 -c：创建文件，没有密码文件时需要加-c，再次添加时不需要

 -m：使用md5

 -s：Force SHA encryption of the password(insecure)

如需创建jackcui的认证密码：

|  |  |
| --- | --- |
| 1  2  3  4 | [root@cnode6\_8 conf.d]# htpasswd -c -s/var/www/html/.htpasswd jackcui  New password:   //输入该用于的密码  Re-type new password:   //再次确认密码  Adding password for user jackcui  //提示添加密码成功 |

**2.http协议的组认证**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <Directory"/vhosts/www1/admin">      Options None      AllowOverride None      AuthType basic      AuthName "Admin Area, Enter yourname/password"      AuthUserFile"/etc/httpd/conf/.htpasswd"      AuthGroupFIle"/etc/httpd/conf/.htgroup"      Require group  </Directory> |

组账号文件格式：每行一个组定义：

|  |  |
| --- | --- |
| 1 | group\_name: user1 user2 ... |

**八、httpd的访问日志和错误日志：**

   httpd的日志可以定义在主配置文件当中，也可以单独定义在每个虚拟主机中，作用域为定义的虚拟主机的块定义范围之内。主配置文件指向的logs目录是/var/logs/httpd的软链接， 可以自己指定单独的磁盘，提高系统性能。配置要注意文件目录的访问权限

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | [root@7cnode2 conf.d]# cat virtualhost.conf  <VirtualHost 192.168.20.129:80>      ServerName vhost.jack.com      DocumentRoot /vhost/vhost1  <Directory "/vhost/vhost1">      options none      AllowOverride none      require all granted  </Directory>      ErrorLog "/testdir/vhostlog/error.log"      CustomLog "/testdir/vhostlog/custom.log"  common  </VirtualHost> |

**重启服务在相应的目录生成文件，访问之后便可以记录在文件当中**

|  |  |
| --- | --- |
| 1  2  3  4  5 | [root@7cnode2 conf.d]# systemctl restart httpd  [root@7cnode2 conf.d]# curl vhost.jack.com  this is vhost1's test page  [root@7cnode2 conf.d]# tail /testdir/vhostlog/custom.log  192.168.20.129 - - [06/Oct/2016:17:36:12 +0800] "GET / HTTP/1.1" 200 27 |

**其中日志格式也可以自己定义**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | <VirtualHost 192.168.20.129:80>          ServerName vhost.jack.com          DocumentRoot /vhost/vhost1  <Directory "/vhost/vhost1">          options none          AllowOverride none          require host 127.0.0.1   //改为仅本机允许访问，可以测试是否能产生错误日志  </Directory>          LogFormat "%h %l \"%{User-Agent}i\"" myformat  //自己定义的错误日志的格式          ErrorLog "/testdir/vhostlog/error.log"          CustomLog "/testdir/vhostlog/custom.log"  myformat  //定义路径和使用的格式  </VirtualHost>  [root@7cnode2 conf.d]# systemctl restart httpd  [root@7cnode2 conf.d]# cat /testdir/vhostlog/error.log  [Thu Oct 06 17:51:23.474098 2016] [authz\_host:error] [pid 37261] [client 192.168.20.1:65376] AH01753: access check of '127.0.0.1' to /noindex/css/fonts/Light/OpenSans-Light.ttf failed, reason: unable to get the remote host name, referer: http://192.168.20.129/noindex/css/open-sans.css  [Thu Oct 06 17:51:23.474185 2016] [authz\_core:error] [pid 37261] [client 192.168.20.1:65376] AH01630: client denied by server configuration: /vhost/vhost1/noindex, referer: http://192.168.20.129/noindex/css/open-sans.css  [root@7cnode2 conf.d]# cat /testdir/vhostlog/custom.log  192.168.20.129 - - [06/Oct/2016:17:36:12 +0800] "GET / HTTP/1.1" 200 27  192.168.20.129 - "curl/7.29.0"  192.168.20.1 - "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0"  192.168.20.1 - "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0"  [root@7cnode2 conf.d]# |

附：访问日志的格式：

**LogFormat：用来定义日志格式**

**CustomLog：定义访问日志的位置与日志格式。**

**常见的格式：**

%h 远程主机的主机名，默认记录的是IP，HostnameLookups是默认off的，如果改为on则服务器要将客户端的ip反解为域名，然后在正解看是否是对应的IP

%l 用与支持服务器支持验证时，记录远程登录的名字

%u http协议在建立http请求时协议认证质询实现

%t 收到请求的时间，格式为[18/Sep/2011:19:18:28 -0400]

%r 首次请求

%b 包括http报头在内的请求大小

%s status，对于内部重定向的请求这是原始的请求状态，使用%>s 记录最终的状态

%{VARNAME}i 记录以VARNAME变量为首部的字符串的值。如%{Referer}i

更多格式的值参阅：

http://httpd.apache.org/docs/2.4/mod/mod\_log\_config.html#formats