

全局配置

Listen

指定 httpd 监听的 IP 地址或者端口。默认监听所有IP。Listen配置指令可以出现多次。

```
1 Listen 192.168.1.150:80
```

```
1 Listen 80
```

Keep Alive

保持连接特性开启或者关闭，默认开启。当客户端使用Keep-Alive连接时，无论MaxConnectionsPerChild的配置指令值是多少，它都将被计为指令的单个“请求”。

```
1 KeepAlive on|off
```

默认是100。当KeepAlive打开时，MaxKeepAliveRequests指令限制每个连接允许的请求数。如果设置为0，将允许无限制请求。建议将此设置保持在较高的值以实现最大的服务器性能。

```
1 MaxKeepAliveRequests 100
```

默认为5，超时时长。KeepAliveTimeout决定一个KeepAlive的连接能保持多少时间，HTTP 的 Keepalive 方式能减少 TCP 连接数量和网络负载，但是keepalive需要和进程或线程绑定，所以就会消耗过多的系统资源。

ServerRoot

服务器的运行目录。设置此配置指令后，其他配置文件涉及到的路径的相对路径就是基于此路径的。

```
1 ServerRoot "/etc/httpd"
```

DocumentRoot

指定站点文件的目录，修改站点文件目录后，要注意目录的权限，可以使用setfacl命令设置目录权限，使的apache用户可以访问。

```
setfacl -Rm u:apache:rwX DocumentRoot
```

```
1 DocumentRoot "/var/www/html"
```

AddDefaultCharset

设定默认字符集

```
1 AddDefaultCharset On|Off|charset
```

定义默认主页格式

可以定义多种格式，会依次查找最先符合的格式文件予以显示。

```
1 <IfModule dir_module>
2     DirectoryIndex index.html
3 </IfModule>
```

访问属性

格式

```
1 #~代表使用正则表达式匹配目录下的文件
2 <Directory ~ "PATH">
3     [Run-time Configuration Directives] VALUE
4     .
5     .
6     .
7 </Directoty>
```

配置指令

Option

```
1 Option VALUE1 VALUE2 . . .
```

VALUE可能的值。

None：所有都不允许。

All：所有都启用。

Includes：缺少默认页面时，允许将站点目录中的所有文件以列表形式列出来。httpd/2.4默认是关闭此功能的，在Option中增加Includes选项，会发现还是无法显示，修改welcome.conf文件中的Option选项。

ExecCGI：允许 mod_cgi模块执行目录下的CGI脚本。

Includes：允许mod_include模块实现服务器端包含(SSI)。

Multiviews：允许使用mod_negotiation实现内容协商。

SymLinksIfOwnerMatch：允许目录下的链接文件也予以显示，但链接文件属主属组与原始文件属主属组相同。

Require

Require all granted：接受所有访问。

Require all denied：拒绝所有访问。

Require method http-method [http-method] ...：指定允许请求的方法。

Require user userid [userid] ...：设置指定的用户可以访问。

Require group group-name [group-name] ...：这只指定用户组访问。

Require valid-user：配置的所有用户都可以访问。

Require ip 10 172.20 192.168.2：指定IP地址范围可以访问。

用户认证

基于用户/用户组的要配置单独的容器

```
1 #创建账号密码信息文件及用户
2 htpasswd -c -m /etc/httpd/conf/.htpasswd USERNAME
3 #创建用户
4 htpasswd -m /etc/httpd/conf/.htpasswd USERNAME
```

```
1 <Directory "/home/www/image">
2     Options Indexes
3     AllowOverride None
4     AuthName "enter passwd:"
5     AuthType Basic
6     AuthUserFile "/etc/httpd/conf/.htpasswd"
7     #组文件的每一行包含一个组名称，后跟冒号，后跟成员用户
    名，以空格分隔。例：GroupName: user1 user2 user3
8     #AuthGroupFile "/etc/httpd/conf/.htgroup"
9 </Directory>
```

AuthType：认证可用类型None， Basic（由 mod_auth_basic 模块提供）， Digest（由mod_auth_digest模块提供），并 Form（由 mod_auth_form模块提供）。

日志功能

ErrorLog "logs/error_log"：错误日志存放路径

LogLevel warn：错误日志级别

```
1 #定义日志格式
2 <IfModule log_config_module>
3     LogFormat "%h %l %u %t \"%r\" %>s %b \"%%
    {Referer}i\" \"%{User-Agent}i\"" combined
4     LogFormat "%h %l %u %t \"%r\" %>s %b" common
5 </IfModule logio_module>
```

```
1 日志格式:
2 %h      #客户端地址
3 %l      #远程登录名, 通常为-符号
4 %u      #认证时的远程用户名, 没有时为-符号
5 %t      #时间
6 %r      #响应报文的起始行
7 %>s     #响应状态码
8 %b      #响应报文的长度, 单位是字节
9 %{head-name}i    #记录指定请求报文首部的内容(value)
```

路径别名

例如站点文件资源路径在/var/www/html/images/目录, 我们试图通过此路径访问这个路径之外的资源, 就需要定义路径别名。使用路径别名时, 要特别注意, 别名路径的权限问题。

```
1 <IfModule alias_module>
2     alias "/images" "/data/images_1"
3     <Directory "/data/images_1">
4         Require all granted
5     </Directory>
6 </IfModule>
```

虚拟机

一个物理服务器提供多个站点, 中心主机和虚拟主机不能同时存在。使用虚拟主机前, 需要关闭默认中心主机。

```
DocumentRoot "/var/www/html" #关闭中心主机
```

常见的虚拟主机的实现方式: 基于不同的主机名(FQDN); 基于不同的IP。

基于主机名的虚拟机

```
1 <VirtualHost *:80>
2     ServerName www.xuejinwei.cc
3     ServerAlias xuejinwei.cc
4     DocumentRoot "/www/test1"
5 </VirtualHost>
6
7 <VirtualHost *:80>
8     ServerName about.xuejinwei.cc
9     DocumentRoot "/www/about"
10 </VirtualHost>
```

基于IP的虚拟机

```
1 <VirtualHost 172.20.30.40:80>
2     ServerAdmin webmaster@www1.example.com
3     DocumentRoot "/www/vhosts/www1"
4     ServerName www1.example.com
5     ErrorLog "/www/logs/www1/error_log"
6     CustomLog "/www/logs/www1/access_log" combined
7 </VirtualHost>
8
9 <VirtualHost 172.20.30.50:80>
10    ServerAdmin webmaster@www2.example.org
11    DocumentRoot "/www/vhosts/www2"
12    ServerName www2.example.org
13    ErrorLog "/www/logs/www2/error_log"
14    CustomLog "/www/logs/www2/access_log" combined
15 </VirtualHost>
```

