Apache服务器

```
[root@test ~]# yum install httpd -y
[root@test ~]# systemctl enable httpd

[root@test ~]# firewall-cmd --add-service=http
[root@test ~]# firewall-cmd --add-service=http --per
[root@test ~]# firewall-cmd --add-service=https
[root@test ~]# firewall-cmd --add-service=https  --per

[root@test ~]# vim /etc/httpd/conf/httpd.conf  # 主配置文件
```

默认站点主目录：/var/www/html/
ServerRoot "/etc/httpd" 用于指定Apache的运行目录，服务启动之后自动将目录改变为当前目录，在后面使用到的所有相对路径都是想对这个目录下

Listen 80 监听的端口，如有多块网卡，默认监听所有网卡

```
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

对/var/www/html目录的一个权限的设置，options中Indexes表示当网页不存在的时候允许显示目录中的文件，FollowSymLinks是否允许访问符号链接文件。

AllowOverride None表示不允许这个目录下的访问控制文件来改变配置，这也意味着不用查看这个目录下的访问控制文件，修改为：AllowOverride All 表示允许.htaccess（访问控制）。

虚拟目录（页面别名）

将目录以外的内容加入到站点中的办法
（用Alias参数设置虚拟目录和实际目录的对应关系）

```
<IfModule alias_module>
    Alias /log /var/log
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
</IfModule>

<Directory "/var/log">
    Options Indexes MultiViews
    AllowOverride all
    Require all granted
</Directory>

<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

```
[root@example html]# chcon -R --reference=/var/www/html  /var/www/err
```

页面中嵌入脚本

```
557 #
558 ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

在/var/www/cgi-bin目录下可以存放脚本，实现一些动态效果。

```
[root@example cgi-bin]# vim test.perl
```

```
1 #!/usr/bin/perl
2 print "content-Type: text/html\n\n";
3 $now=localtime();
4 print $now,"\n";
```

```
[root@example cgi-bin]# chmod +x test.perl
```

启动用户的个人网站

```
[root@test ~]# vim /etc/httpd/conf.d/userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>
<Directory "/home/*/public_html">
    AllowOverride All
    Options None
    Require method GET POST OPTIONS
</Directory>
[root@test ~]# setsebool -P  httpd_enable_homedirs=1
```

```
[root@test ~]# restorecon -Rv /home/ # 处理安全类型
[root@test ~]# chmod 711 /home/qingqing
[qingqing@test ~]$ mkdir public_html
[qingqing@test ~]$ chmod 755 public_html
客户端访问：http://120.25.79.224/~qingqing/
```

Apache安全配置

提供了多种安全配置：web访问控制、用户登录密码认证、.htaccess文件等

一、访问控制，Apache服务器安全级别最有效的手段之一。

Diretory用于设置与目录相关的参数和指令

```
<Directory 目录的路径>
    目录相关的配置参数和指令
</Directory>
```

a. Allow 指令，用于设置哪些客户端可以访问Apache

Allow from [ALL/全域名/部分域名/IP地址/网络地址/CIDR地址]

全域名：域名对应的客户端，如www.baofeng.com

部分域名：域内的所有客户端如baofeng.com

网络地址：如170.20.17.0/256.356.355.0

Allow 可以指定多个IP地址，不同地址间通过空格进行分隔

b. Deny 拒绝哪些客户端访问Apache

c. Order 指定执行访问规则的先后顺序。

a. Order allow，deny 先执行允许，再执行拒绝 常用于开放所有，拒绝特定的条件

b. Order deny，allow 先执行拒绝，再执行允许 常用于拒绝所有，开放特定的条件

例：<Directory "/var/www/html/"> 拒绝所有，仅允许特定的访问

```
        Order deny,allow
        Deny from all
        Allow from  192.168.24.181
    </Directory>
    <Directory "/var/www/html/"> 开放所有仅拒绝特定的访问
        Order allow,deny
        Allow from  all
        Deny from 192.168.24.181
    </Directory>
```

二、用户认证包括：基本认证和摘要认证。摘要认证比基本认证更安全，但不是所有的浏览器都支持

基本认证：用户访问网页时，弹出对话框，输入用户名，密码

使用基本认证，首先创建保存用户名和密码的认证密码文件。Apache提供htpasswd命令用于创建和修改认证密码文件

```
[root@centos6 ~]# htpasswd  -c /var/www/html/users sam  # 在/var/www/html/目录下创建名为users的认证密码文件，并在其中添加用户sam
```

创建认证密码文件后，需要对配置文件进行修改。用户认证是在http.conf配置文件中的<Directory>段中设置

```
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    AuthType Basic # AuthType参数：用于设置认证的方式
    AuthName  "security_info" # 设置受保护领域的名称
    AuthUserFile "/var/www/html/users" # 用于设置认证密码文件的位置
    Require user sam  # 用于指定哪些用户可以对目录进行访问
    Order deny,allow
    Deny from all
    Allow from 192.168.88.191
</Directory>
```

三、访问控制文件

.htaccess 可以覆盖http.conf文件中的配置，但只能设置对目录的访问和用户认证。可以有多个.htaccess 文件，每个.htaccess 文件的作用范围仅限于该文件所存放的目录以及该目录下的所有子目录。.htaccess 文件修改配置后不要重启Apache服务

```
386 #
387 AccessFileName .htaccess 定义网页访问验证的文件
388
```

```
317 <Directory "/var/www/html">
318
319     Options Indexes FollowSymLinks
320     AllowOverride AuthConfig
321     Order deny,allow
322     Allow from all
323
324 </Directory>
```

通过认证配置文件来访问

```
[root@test html]# vim .htaccess
AuthName "please input user and password"
AuthType "Basic"
AuthUserFile "/var/www/html/users"
Require user sam
```

虚拟主机

2019年7月26日　　莫宇剑♡符菁 21:15

将一台物理服务器虚拟成多台web服务器。
虚拟主机服务方案：

1. 基于IP的虚拟主机服务
2. 基于主机名的虚拟主机服务
3. 基于端口的虚拟主机服务

```
[root@test ~]# yum install httpd -y
[root@test ~]# rpm -ql httpd |grep vhost
[root@test ~]# cp /usr/share/doc/httpd-2.4.6/httpd-vhosts.conf /etc/httpd/conf/ # 复制模板
```

基于主机名的虚拟主机服务：当客户程序向web服务器发送请求时，客户想要访问的主机名通过请求头中的Host：语句传递给服务器。

```
[root@test ~]# vim /etc/httpd/conf.d/httpd-vhosts.conf
#NameVirtualHost 192.168.88.185
<VirtualHost *:80>
    DocumentRoot "/var/www/html/host1"
    ServerName host1.test.com
</VirtualHost>
<VirtualHost *:80>
    DocumentRoot "/var/www/html/host2"
    ServerName host2.test.com
</VirtualHost>
[root@test ~]# apachectl -t
Syntax OK # 检查配置文件，如果出现Syntax OK，则配置文件正确
```

基于端口的虚拟主机服务

```
[root@test ~]# vim /etc/httpd/conf/httpd.conf # 在apache主配置文件中加入新的监听端口
#Listen 12.34.56.78:80
Listen 80
Listen 8080
[root@test ~]# vim /etc/httpd/conf.d/h.conf
<VirtualHost *:80>
    DocumentRoot "/var/www/html/host1"
    ServerName host1.test.com
</VirtualHost>
<VirtualHost *:8080>
    DocumentRoot "/var/www/html/host2"
    ServerName host2.test.com
</VirtualHost>
[root@test ~]# systemctl restart httpd
防火墙放行端口
[root@test ~]# firewall-cmd --add-port=8080/tcp
[root@test ~]# firewall-cmd --add-port=8080/tcp --permanent
[root@test ~]# semanage port --add  -t http_port_t -p tcp 8080    selinux上添加apache信任端口
```

基于IP的虚拟主机服务：根据用户请求的目的IP来判定用户请求的是哪个虚拟主机的服务

在服务器上创建多个IP地址

```
[root@test ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE="Ethernet"
BOOTPROTO="static"
NAME="ens33"
DEVICE="ens33"
ONBOOT="yes"
IPADDR0=192.168.88.200
IPADDR1=192.168.88.201
NETMASK=255.255.255.0
GATEWAY=192.168.88.2
DNS2=114.114.114.114

[root@test ~]# ifconfig ens33:2 192.168.88.202 netmask 255.255.255.0 up
[root@test ~]# ip addr add 192.168.88.202/24 dev ens33
[root@test ~]# ip addr show dev ens33
```

```
[root@test ~]# ip addr show dev ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
roup default qlen 1000
    link/ether 00:0c:29:ba:d2:e7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.88.200/24 brd 192.168.88.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet 192.168.88.201/24 brd 192.168.88.255 scope global secondary noprefixrou
te ens33
       valid_lft forever preferred_lft forever
    inet 192.168.88.202/24 scope global secondary ens33
       valid_lft forever preferred_lft forever
    inet6 fd15:4ba5:5a2b:1008:20c:29ff:feba:d2e7/64 scope global mngtmpaddr dyna
mic
       valid_lft 86381sec preferred_lft 14381sec
    inet6 fe80::20c:29ff:feba:d2e7/64 scope link
       valid_lft forever preferred_lft forever
```

```
[root@test ~]# vim /etc/httpd/conf.d/httpd-vhosts.conf
<VirtualHost 192.168.88.200:80>
    DocumentRoot "/var/www/html/host1"
    ServerName host1.test.com
</VirtualHost>
<VirtualHost 192.168.88.201:80>
    DocumentRoot "/var/www/html/host2"
    ServerName host2.test.com
</VirtualHost>
```

基于SSL封装的WEB服务器

2019年7月27日　　莫宇剑♡着著 0:03

HTTPS证书获取途径

1、自签名证书被推荐用于测试目的和个人项目。自签名证书，也可以用于服务提供商，不过一般适用于用户互相信任的情形。另外，自签名证书不用花钱购买。
2、证书可以由社区为基础的认证供应商如StartSSL和CACERT办法。这些证书也不需要花钱，但建议为个人项目。
3、对于全球性商业网站，建议从值得信赖的知名证书颁发机构购买证书。这些证书需要花钱，但他们增加了网络服务提供商的信誉。

为了保证使用http协议传输网页信息的安全，我们会使用ssl来封装网页的内容。最常用的应用就是网银和支付宝等。SSL（secure sockets layer）安全套接层，是为网络通信提供安全和数据完整性的一种安全协议。

A和B通信，B为了确保A的身份的可靠性，会进行数字签名。A首先将希望发送的数据进行HMAC，得到一个结果xx，然后A用自己的私钥加密这个xx，得到一个数据TT。A将原始的数据和TT一起发送给B，B将数据也进行HMAC，得到结果yy。并用A的公钥解密TT，解密后得到的xx和自己yy对比，来判断A的身份。数字签名可以解决中间人监听这样的问题，但是因为A的密钥对和A本身并没有直接的联系，所以可能导致的法律等范畴的纠纷。
为了解决数字签名中的密钥对的信任问题，我们采用CA（数字证书认证中心），CA通常都是由可靠的第三方来担任，A或B需要通过自身的可信任信息（身份信息、公司信息、信用信息等）来向CA认证中心提出认证请求，CA通过对发起者的信息确认后对对方的密钥对进行注册，保证在纠纷过程中能够出具可被信任的电子证据。

```
[root@test ~]# yum install mod_ssl -y  # 安装ssl模块
[root@test ~]# vim /etc/httpd/conf.d/ssl.conf     配置ssl模块
```

```
100 SSLCertificateFile /etc/httpd/conf/example.crt
101
102 #   Server Private Key:
103 #   If the key is not combined with the certificate, use this
104 #   directive to point at the key file.  Keep in mind that if
 93 #SSLHonorCipherOrder on
 94
 95 #   Server Certificate:
 96 # Point SSLCertificateFile at a PEM encoded certificate.  If
 97 # the certificate is encrypted, then you will be prompted for a
 98 # pass phrase.  Note that a kill -HUP will prompt again.  A new
 99 # certificate can be generated using the genkey(1) command.
100 SSLCertificateFile /etc/httpd/conf/example.crt
```

```
[root@test /]# cd /etc/pki/tls/certs/    进入证书生成的目录


# 不要 [root@linuxprobe certs]# openssl rsa -in server.key -out server.key
# 不要 [root@linuxprobe certs]# make server.csr
# 不要 [root@linuxprobe certs]# openssl x509 -in server.csr -out server.crt -req -signkey server.key -days
3650


[root@test certs]# make example.crt    生成证书
[root@test certs]# mv example.* /etc/httpd/conf/    将生成的证书和密钥移动到对应的目录下

防火墙放行https
[root@test conf]# firewall-cmd --add-service=https
[root@test conf]# firewall-cmd --add-service=https --permanent
[root@test conf]# systemctl restart httpd.service


https://www.cnblogs.com/mingzhang/p/8949541.html
```