http（超文本传输协议）：

web服务器：

简单点说就是我们如果要浏览一个网页的话，基本上所有的网站都使用的是http协议来进行数据传输的！至于怎么样传输，我们做为运维来说就没有必要去深究了，那是做html前端开发人员要去考虑的事情！

survey.netcraft.net 这个是统计全球web服务器使用率的一个网站！

根据排行结果显示，目前最主流的web服务器为：

Apache

Microsoft

Sun

nginx 等等

我们这里主要讲的是Apache：www.apache.org 这个是apache的官方网址。

# curl -I www.apache.org

HTTP/1.1 200 OK

Date: Tue, 01 Nov 2016 13:02:28 GMT

Server: Apache/2.4.7 (Ubuntu) 这里可以看到该网站使用的是什么web服务器

Last-Modified: Tue, 01 Nov 2016 12:10:17 GMT

ETag: "d21e-5403c352b2012"

Accept-Ranges: bytes

Content-Length: 53790

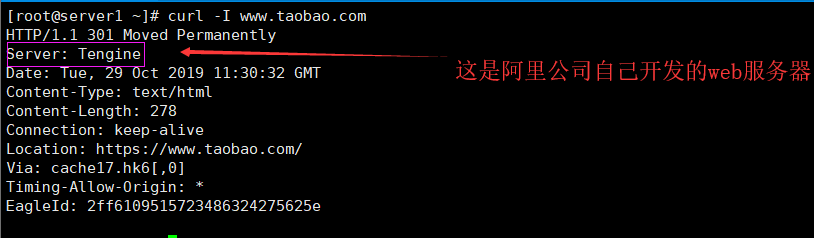
Vary: Accept-Encoding

Cache-Control: max-age=3600

Expires: Tue, 01 Nov 2016 14:02:28 GMT

Content-Type: text/html

# curl -I www.taobao.com

Set-Cookie: thw=cn; Path=/; Domain=.taobao.com; Expires=Wed, 01-Nov-17 13:01:54 GMT;

Strict-Transport-Security: max-age=31536000

GWS(谷哥) BWS(百度) SWS(搜狐） Tegine（阿里) (都是在apache或nginx的基础上二次开发的）

中间件：

websphere tomcat jboos weblogic

==================================================

网站分为两类：

1.静态网站

2.动态网站

(关于静态网站和动态网站的解析：

程序是否在服务器端运行，这个是判断网页属不属于动态网页的重要标志。

在服务器端运行的程序、网页、组件，属于动态网页，它们会随不同客户、不同时间，返回不同的网页，例如ASP、PHP、JSP、ASP.net、CGI等。

运行于客户端的程序、网页、插件、组件，属于静态网页，例如html页、Flash、JavaScript、VBScript等等，它们是永远不变的。

在网站设计中，纯粹HTML格式的网页通常被称为“静态网页”，早期的网站一般都是由静态网页制作的。静态网页的网址形式通常为：www.example.com/eg/eg.htm，也就是以.htm、.html等为后后缀的。在HTML格式的网页上，也可以出现各种动态的效果，如.GIF格式的动画、FLASH、滚动字母等，这些“动态效果”只是视觉上的，与我说常说的动态网页是不同的概念。静态网页是实实在在保存在服务器上的文件，每个网页都是一个独立的文件等。

动态网页，与网页上的各种动画、滚动字幕等视觉上的“动态效果”没有直接关系，动态网页也可以是纯文字内容的，也可以是包含各种动画的内容，这些只是网页具体内容的表现形式，无论网页是否具有动态效果，采用动态网站技术生成的网页都称为动态网页

动态的是需要服务器解释的。

静态的放到什么地方都能运行的。

你静态的拷贝的别的地方 也能看到。

动态的就不行。)

==================================

安装httpd服务：

# yum install httpd -y

检查是否安装好：

# rpm -qa |grep httpd

httpd-2.2.15-47.el6.centos.4.x86\_64

httpd-devel-2.2.15-47.el6.centos.4.x86\_64

httpd-manual-2.2.15-47.el6.centos.4.noarch

httpd-tools-2.2.15-47.el6.centos.4.x86\_64

启动httpd服务：

# systemctl restart httpd

设置开机自启动：

# systemctl enable httpd

http://httpd.apache.org/docs/2.4/configuring.html --apache中文手册

httpd服务配置文件解析：

# vim /etc/httpd/conf/httpd.conf --apache主配置文件，主要关注以下几个

serverRoot "/etc/httpd" --配置文件目录

Listen 80 --监听端口

Listen 192.168.1.70:80 --只有192.168.1.70网段能访问apache

Include conf.d/\*.conf --表示conf.d/下的\*.conf配置文件生效

User apache

Group apache --以什么身份运行

DocumentRoot "/var/www/html" --网站家目录

DirectoryIndex index.html --定义主页文件支持的格式

# ls /etc/httpd/conf.d/welcome.conf --test页面的配置文件

# systemctl start httpd

使用浏览器：输入自己的IP，或者127.0.0.1回环地址会看到apache的测试页面

=====================================================

注意：进行下面的操作前，请先将iptables（防火墙）规则清空，并把selinux关闭后重启系统！

实践例题：

建立一个主页文件：

# echo 'main page' > /var/www/html/index.html

完成后，打开浏览器输入http://IP 如果能看到 main page 就说明apache已经搭建完成

主页文件默认是没有的 需要我们手动建立才会有，支持的主页文件名称类型不限，主要取决于你自己去定义

/etc/httpd/conf.d/welcome.conf 这个是默认的测试主页

# rm -rf /etc/httpd/conf.d/welcome.conf 将这个文件删除后在去试试看

把主页文件写成html标签的格式如下：

# vim /var/www/html/index.html

<html>

<head>

<title>笔记站点</title>

</head>

<body>

<center><h1>欢迎来我的网站!@\_@</h1></center>

</body>

</html>

写完后保存退出，再去浏览器试试看！

=========================================

例二：修改网站家目录

# vim /etc/httpd/conf/httpd.conf

119： DocumentRoot "/www" --修改网站家目录为/www

131： <Directory "/www"> --把这个也对应的修改为/www

修改完成后重新装载httpd服务，相当于刷新不是重启

# systemctl reload httpd

# mkdir /www 建立httpd服务的根目录

# echo 'new main page' > /www/index.html 网站根目录改变后我们新建的网站主页文件就要写到新指定的网站根目录中

上面都修改完成后请到浏览器进行测试！

需要重启apache

再次提醒：注意！一定不要开启selinux，否则换新的家目录的话，会造成报目录不存在的错误

=========================================

例三：修改主页类型或者主页名

DirectoryIndex index.html index.html.var --定义主页文件支持的格式

在配置文件中的这句话就是定义限制了我们httpd服务的主页文件的支持！

那么我们对其进行一点小修改！

# vim /etc/httpd/conf/httpd.conf

DirectoryIndex index.php

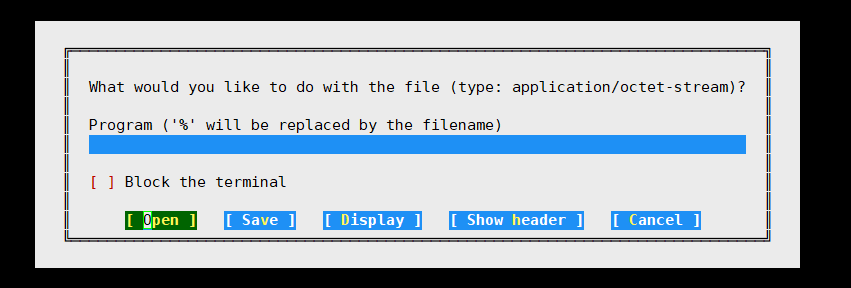
修改后保存退出，在使用浏览器去测试，发现没有返回主页信息，因为上面的意思为:先找网站家目录下的index.php，不存在话，再找index.html.var，都不存在，就表示找不到主页

# echo 'php main page' > /www/index.php

在测试，发现又能找到了！

# elinks 192.168.224.11 --elinks是一个文本模式的web浏览器，如果报命令找不到请使用下面的命令将其安装上：

# yum install -y elinks\*



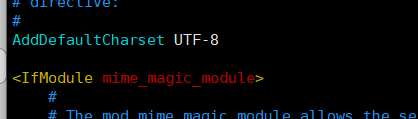
=========================================

例四：修改语言集和字符集

# vim /etc/httpd/conf/httpd.conf

DefaultLanguage zh-CN --将默认语言集设置为简体中文zh-CN

AddDefaultCharset UTF-8 --字符编码，如果中文的话，有可能需要改为gb2312或者gbk,因你的网站文件的默认编码而异



常见的可以显示中文的字符集有:gbk,gb2312,gb18030,utf-8

--扩展：

windows默认的字符集是gb2312

linux默认的字符集是utf-8

----------------------------------

关于apache的标签，容器（访问控制）

标签类型：

directory （目录)

files(文件）

location （url）

<Directory "/www"> --对/www目录进行的控制

Options Indexes FollowSymLinks --允许列表，符号链接

AllowOverride None --不使用.htaccess控制

Order allow,deny --指定的是先允许，后拒绝

Allow from all --针对/www目录的访问权限的控制，这两句代表允许所有访问

</Directory> --容器结束

容器内控制选项详解：

indexes --指当找不到默认的主页文件时，就把此目录下的文件或者目录以列表形式显示出来

FollowSymlinks --允许符号链接，扩展了网站根目录，允许链接到网站根目录以外

allowoverride none --可以把none改为all来允许.htaccess控制

order allow,deny --指定的是先允许，后拒绝

allow from all --指定允许所有

例五：把网站的文件修改成目录列表的形式

直接在网站根目录创建一个index.html命名的目录即可

# mkdir /var/www/html/index.html

或者直接将/etc/httpd/conf.d/welcome.conf 文件删除或重命名即可

# rm -rf /etc/httpd/conf.d/welcome.conf

然后使用浏览器访问你的IP就可以看到效果

=========================================

例六：访问权限控制 --注意这些访问权限控制是针对容器的

Order allow,deny --先允许后拒绝

allow from all --允许所有

Order allow,deny

deny from all --拒绝所有

Order allow,deny

allow from all

deny from 10.0.0.2 --允许所有，但拒绝此IP

Order deny,allow \\ 注意这里的写法，这里是先拒绝后允许

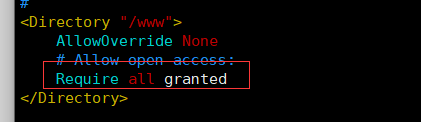
deny from all

allow from 10.0.0.2 --拒绝所有，但允许此IP

新版本apache

Require all granted --允许所有

Require all denied --拒绝所有



Require ip 10.0.0.2 --允许单个IP

Require ip 10.0.0.0/24 --允许某个网段

Require ip 10 192.168 1.1.1 --允许10、192.168、1.1.1开头的IP

Require host local --以主机名的形式限制，只允许本地访问，其它都拒绝

<RequireAll>

Require all granted

Require not ip 10.0.0.2 --允许所有，但拒绝此IP（此方法必须写入专用容器）

</RequireAll>

练习：在你的家目录/www目录下新建一个bbs目录，也就是/www/bbs，再在/www/bbs里建立一个index.html测试文件，使用directory目录容器控制这个子目录允许自己访问，其它都拒绝

<Directory "/www/bbs">

order deny,allow

deny from all

allow from 10.0.0.2

</Directory>

或

<Directory "/www/bbs">

require host localhost --这种写法自己访问自己也只能使用127.0.0.1的地址访问，可以换成自己的IP

</Directory>

Directory容器不能嵌套在其它目录容器里，所以这里需要重新定义一个独立的容器，并且写成绝对路径

=========================================

例七.files 容器 --针对文件的控制

<Directory "/www">

Options Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

<files "1"> --表示2这个客户端不能访问/www/1,而且如果要控制/www/bbs/1这个文件，不能在这里写成<files "bbs/1">，需要在/www/bbs这个目录的directory容器里再嵌套files标签来控制

order allow,deny

allow from all

deny from 10.0.0.2

</files>

</Directory>

-------------------------------------

<files "/www/1">

order allow,deny

deny from all

</files> --这种写法没有生效，所以要files容器嵌套到目录容器内部，要上面的那种写法

<files "1">

order allow,deny

deny from all --整个文档目录都生效

</files>

--下面这段是apache的默认的一个files容器的配置示例，表示将apache家目录中所有.ht开头的文件都隐藏

<Files ".ht\*">

Require all denied

</Files>

注意：Files容器对配置的目录以及子目录起作用，对上一级目录没有影响

Files容器不在任何一个目录，将会对整个数据目录起作用

=========================================

例八：location 容器（对URL进行控制）

<Location "/bbs">

order allow,deny

deny from all

</Location>

--这是对http://IP/bbs 这种url来进行权限控制

=========================================

例九：对网站目录使用apache的基本验证（使用用户名和密码进行验证）功能

1.

# vim /www/.htaccess 　　--对哪个目录进行限制，就在哪个目录下建立此文件

authname "please input your username and password! @\_@"

authtype basic

authuserfile /etc/httpd/userpasswd

require valid-user

2.

# htpasswd -c /etc/httpd/userpasswd aaa　　--创建此文件，并加入一个用户，自定义密码，注意此用户与系统普通用户无关

New password:

Re-type new password:

Adding password for user aaa

# htpasswd /etc/httpd/userpasswd bbb　　--再增加一个用户，文件已经存在，无需在创建，所以不需-c参数

New password:

Re-type new password:

Adding password for user bbb

# cat /etc/httpd/userpasswd

aaa:JscuRCtWV0Sh.

bbb:C6wrx4pFco/lk --密码为密文

3.

# vim /etc/httpd/conf/httpd.conf

<Directory "/www">

Options Indexes FollowSymLinks

AllowOverride all --改成all，支持.htaccess加密

Order allow,deny

Allow from all

</Directory>

4.

# systemctl restart httpd

http://IP 访问时，需要验证

再次访问需要清空浏览器缓存才需要验证

=========================================

例十:

apache的alias跳转（可以将不是网站家目录的目录加入进来）

# vim /etc/httpd/conf/httpd.conf

Alias /test "/test"

<Directory "/www">

Options Indexes MultiViews FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

可以做Alias跳转，也可以做软链接

并且需要将下面的内容注释掉才能正常访问，因为apache被限制禁止直接访问根目录下的内容

<Directory />

AllowOverride none

# Require all denied # 注释该行

</Directory>

新版本apache为了安全考虑，已经不支持这样做了，只能通过软连接的形式实现

# ln -s /test /www/test

=========================================

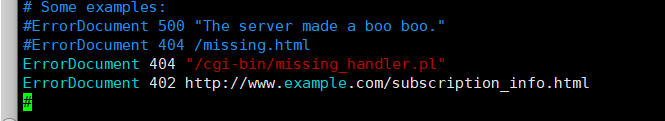
例十一：

错误页面

# ls /var/www/error/404.jpg --准备一个图片放到这个位置

# vim /etc/httpd/conf/httpd.conf --修改主配置文件，指向你的这个图片路径

ErrorDocument 404 /error/404.jpg



# systemctl restart httpd

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

虚拟主机 --用apache或nginx就可以做

一台服务器跑多个web服务

VPS virtual private server 虚拟专用服务器 --使用虚拟化技术来做

云服务器

虚拟主机(用web服务器来做)－－》 VPS (虚拟化) －－》 云服务器 (云计算)

使用虚拟化技术（vitural sever 或者 vmware EXS server )把一台服务器分隔成几个服务器，每个都拥有独立的CPU资源，内存资源，磁盘资源，独立IP等

mkdir -p /www/aaa

mkdir -p /www/bbb

echo "aaa main page" > /www/aaa/index.html

echo "bbb main page" > /www/bbb/index.html

例十二：基于ＩＰ的虚拟主机

NameVirtualHost \*:80

<VirtualHost \*:80>

DocumentRoot /www/aaa

ServerName 1.1.1.4

ErrorLog logs/1-error\_log

CustomLog logs/1-access\_log common

</VirtualHost>

<VirtualHost \*:80>

DocumentRoot /www/bbb

ServerName 1.1.1.6

ErrorLog logs/2-error\_log

CustomLog logs/2-access\_log common

</VirtualHost>

# ifconfig ens33:0 1.1.1.6/24 将上面虚拟主机中使用的ip都虚拟出来

# systemctl restart httpd

验证：

http://1.1.1.4

http://1.1.1.6

例十三：基于端口的虚拟主机

Listen 80 --这一句默认就有，不用再加

Listen 8000

<VirtualHost \*:80>

DocumentRoot /www/aaa

ServerName 1.1.1.4

ErrorLog logs/1-error\_log

CustomLog logs/1-access\_log common

</VirtualHost>

<VirtualHost \*:8000>

DocumentRoot /www/bbb

ServerName 1.1.1.4

ErrorLog logs/2-error\_log

CustomLog logs/2-access\_log common

</VirtualHost>

systemctl restart httpd

验证：

http://1.1.1.4/

http://1.1.1.4:8000/

------------------------------------------------------------------

例十四：基于域名的虚拟主机

--需要DNS的cname记录的支持，或者A记录（这里需要使用到DNS服务器的知识，今天就随便讲一下）

aaa.test.com

bbb.test.com

--以下架构是DNS服务器和apache服务器的IP在同一台机器10.0.0.2上，也可以考虑分成两台服务器来做

# yum install bind\* -y

修改域名解析服务器的主配置文件

# vim /etc/named.conf

listen-on port 53 { any; }; --改为any，表示监听所有

allow-query { any; }; --改为any,表示所有人能查询这台服务器

zone "test.com" IN {

type master;

file "data/test.com.zone";

};

配置解析文件

# vim /var/named/data/test.com.zone

$TTL 86400

@ IN SOA test.com. test.com. (

2011032601

120

240

360

86400 )

IN NS 10.0.0.2.

aaa IN A 10.0.0.2 --这里为apache服务器的IP

bbb IN CNAME aaa

# systemctl restart named

客户端修改域名解析服务器为自己搭建的

# vim /etc/resolv.conf

nameserver 1.1.1.4

测试解析是否正常：

# nslookup aaa.test.com

# nslookup bbb.test.com

=========================================

配置基于域名的web服务器

# vim /etc/httpd/conf/httpd.conf

NameVirtualHost \*:80

<VirtualHost \*:80>

DocumentRoot /www/aaa

ServerName aaa.test.com

ErrorLog logs/1-error\_log

CustomLog logs/1-access\_log common

</VirtualHost>

<VirtualHost \*:80>

DocumentRoot /www/bbb

ServerName bbb.test.com

ErrorLog logs/2-error\_log

CustomLog logs/2-access\_log common

</VirtualHost>

# systemctl restart httpd

在客户端浏览器或使用curl命令进行测试：

http://aaa.test.com/

http://bbb.test.com/

-----------------------------------------------------------------------

思考：

1。上面两个域名，那么我直接用IP是能否访问，访问的是哪一个？为什么？

答案：哪个虚拟主机在最前面，直接使用IP访问的就是哪一个

2。配置了虚拟主机后，那么我原来的DocumentRoot里指的原网站家目录是否还可用?

如果不可用，如何解决?

答案：不能用。

除非再为原家目录再建立一个虚拟主机

问题:上面这三种各有什么优缺点?

基于IP:浪费IP

基于端口:访问时需要加端口，客户不可能记端口的

基于域名:比较适合外网直接访问

客户端

｜

｜

路由器

｜

｜

防火墙

|

|

反向代理或缓存

｜

｜

网站