

Web前端自学笔记-发布

如何让其他用户在任何情况下访问html文件？

(内容可能存在纰漏，在后续的学习过程中会不断进行修改和调整)

1. 首先需要公网服务器。

- 注册腾讯云账号，选择云服务器，按需购买。
- 点击登录按钮。

配置	主IP地址	实例计费模式 ▾	网络计费模式	
1 核 2 GB 1 Mbps 系统盘：普通云硬盘 网络：Default-VPC	129.204.207.46 (公) 172.16.0.10 (内)	包年包月 2019-06-03 16:19到期	按带宽包年包月计费	登录 续费 更多 ▾

- 验证身份后，选择一种方式进行登录。这里腾讯云推荐使用Webshell

登录Linux实例

×

浏览器 WebShell 方式登录

支持复制粘贴、中文输入法，需要开启 SSH 端口 (TCP:22)

[立即登录](#)

标准登录方式

• 使用 Windows 系统的电脑

- 1、下载安装 Windows SSH 和 Telnet 客户端工具 Putty。 [下载Putty](#)
- 2、用户名： **root**，Host： **129.204.207.46**
- 3、按照 Putty 使用帮助进行登录。 [Putty 密码方式使用帮助](#)

• 使用 Linux/Mac OS X 系统的电脑（使用密码登录）

- 1、打开 SSH 客户端（Mac可使用系统自带的终端）
- 2、输入 **ssh -q -l root -p 22 129.204.207.46**
- 3、输入 CVM 实例密码进行登录。

• 使用 Linux/Mac OS X 系统的电脑（使用密钥登录）

- 1、打开 SSH 客户端（Mac可使用系统自带的终端）。
- 2、查找您实例关联的 SSH 密钥文件本地存放地址。
- 3、您的密钥必须不公开可见，SSH 才能工作。请使用此命令：**chmod 400 [密钥文件路径]**。
- 4、输入命令：**ssh [-i 密钥文件路径] root@129.204.207.46**。

- 输入密码

登录实例



密码登录

密钥登录

实例IP 129.204.207.46

端口

用户名

登录密码

注意：

请确认安全组中来源为[Webshell代理IP](#)的远程登录端口（如SSH默认22端口）已经放通，[详细说明](#)。

当登录卡顿时建议您优先关注机器的cpu，内存运行情况，订阅[云监控](#)在主机异常时通过短信等多种方式通知您，[详情说明](#)。

腾讯云不会保存您的实例密码或密钥，请妥善保管谨防丢失。

确定

取消

- 现在我们已经成功进入Linux系统终端界面

清理终端

```
* Socket connection established *
Last login: Mon Mar  4 12:23:43 2019 from 119.29.96.147
[root@VM_0_10_centos ~]#
```

2. 下面我们进行的是服务器配置

- 首先我们要了解，NGINX服务需要的前提环境：gcc-c++、pcre、zlib、openssl

- 安装命令：

gcc: yum install gcc-c++

安装pcre库

1. 进入local文件夹cd /usr/local
2. 下载压缩包: wget
ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.39.tar.gz
3. 解压缩tar -zxvf pcre-8.39.tar.gz
4. 进入解压后的文件夹cd pcre-8.39
5. 执行安装 ./configure 然后make 然后make install

安装zlib库

1. 进入local文件夹cd /usr/local
2. 下载wget http://zlib.net/zlib-1.2.11.tar.gz
3. 解压缩tar -zxvf zlib-1.2.11.tar.gz
4. 进入解压后文件夹cd zlib-1.2.11
5. 执行安装 ./configure 然后make 然后make install

安装OpenSSL

1. 进入local文件夹cd /usr/local
2. 下载wget https://www.openssl.org/source/openssl-1.0.1t.tar.gz
3. 解压缩tar -zxvf openssl-1.0.1t.tar.gz
4. 进入解压缩文件夹cd openssl-1.0.1t
5. 执行安装 ./config 然后make 然后make install

安装nginx

1. 进入local文件夹cd /usr/local
2. 下载wget https://nginx.org/download/nginx-1.10.1.tar.gz
3. 解压缩tar -zxvf nginx-1.10.1.tar.gz
4. 进入解压缩文件夹cd nginx-1.10.1
5. 执行安装 ./configure 然后make 然后make install

- 检查nginx是否安装成功: /usr/local/nginx/sbin/nginx -t -c
/usr/local/nginx/conf/nginx.conf

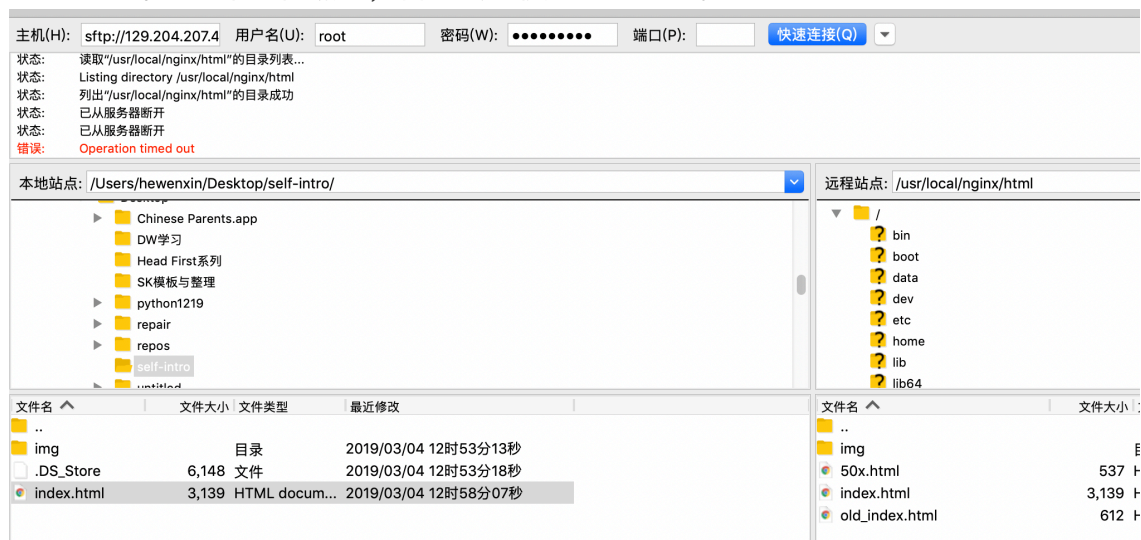
- 启动nginx： `/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf`
- 查看进程： `ps -ef|grep nginx`，看到有master process和worker process进程，说明成功了
- 此时输入网址，显示如下界面：



3. 编写index.html文件

4. 把文件放到服务器中

这里我对命令行不是很熟悉，首先尝试使用了filezilla。



连接至云服务器，分别选择目标文件夹，进行拖拽即可完成上传任务。

文件命名为index.html直接拖拽入对应文件夹，图片统一收集到img文件夹中，再进行拖拽。（这里需要注意的是html文件中图片url的统一）

更新：Linux命令行的使用

涉及到的命令：

cd

ls
mv
root@

更新：Linux常用命令行的积累

1. cd命令

这是一个非常基本，也是大家经常需要使用的命令，它用于切换当前目录，它的参数是要切换到的目录的路径，可以是绝对路径，也可以是相对路径。如：

cd /root/Documents # 切换到目录/root/Documents

cd ./path # 切换到当前目录下的path目录中，“.”表示当前目录

cd ../path # 切换到上层目录中的path目录中，“..”表示上一层目录

2. ls命令

这是一个非常有用的查看文件与目录的命令，list之意，它的参数非常多，下面就列出一些我常用的参数吧，如下：

-l：列出长数据串，包含文件的属性与权限数据等

-a：列出全部的文件，连同隐藏文件（开头为.的文件）一起列出来（常用）

-d：仅列出目录本身，而不是列出目录的文件数据

-h：将文件容量以较易读的方式（GB，kB等）列出来

-R：连同子目录的内容一起列出（递归列出），等于该目录下的所有文件都会显示出来

注：这些参数也可以组合使用，下面举两个例子：

ls -l #以长数据串的形式列出当前目录下的数据文件和目录

ls -lR #以长数据串的形式列出当前目录下的所有文件

3. grep命令

该命令常用于分析一行的信息，若当中有我们所需要的信息，就将该行显示出来，该命令通常与管道命令一起使用，用于对一些命令的输出进行筛选加工等等，它的简单语法为

grep [-acinv] [--color=auto] '查找字符串' filename

它的常用参数如下：

-a：将binary文件以text文件的方式查找数据

-c：计算找到‘查找字符串’的次数

-i：忽略大小写的区别，即把大小写视为相同

-v：反向选择，即显示出没有‘查找字符串’内容的那一行

#例如：

#取出文件/etc/man.config中包含MANPATH的行，并把找到的关键字加上颜色

```
grep --color=auto 'MANPATH' /etc/man.config
```

#把ls -l的输出中包含字母file（不区分大小写）的内容输出

```
ls -l | grep -i file
```

4. find命令

find是一个基于查找的功能非常强大的命令，相对而言，它的使用也相对较为复杂，参数也比较多，所以在这里将给把它们分类列出，它的基本语法如下：

```
find [PATH] [option] [action]
```

#与时间有关的参数：

-mtime n：n为数字，意思为在n天之前的“一天内”被更改过的文件；

-mtime +n：列出在n天之前（不含n天本身）被更改过的文件名；

-mtime -n：列出在n天之内（含n天本身）被更改过的文件名；

-newer file：列出比file还要新的文件名

#例如：

```
find /root -mtime 0 # 在当前目录下查找今天之内有改动的文件
```

#与用户或用户组名有关的参数：

-user name：列出文件所有者为name的文件

-group name：列出文件所属用户组为name的文件

-uid n：列出文件所有者为用户ID为n的文件

-gid n：列出文件所属用户组为用户组ID为n的文件

#例如：

```
find /home/ljianhui -user ljianhui # 在目录/home/ljianhui中找出所有者为ljianhui的文件
```

#与文件权限及名称有关的参数：

-name filename：找出文件名为filename的文件

-size [+ -]SIZE：找出比SIZE还要大（+）或小（-）的文件

-type TYPE：查找文件的类型为TYPE的文件，TYPE的值主要有：一般文件（f）、设备文件（b、c）、

目录（d）、连接文件（l）、socket（s）、FIFO管道文件（p）；

-perm mode：查找文件权限刚好等于mode的文件，mode用数字表示，如0755；

-perm -mode：查找文件权限必须要全部包括mode权限的文件，mode用数字表示

-perm +mode：查找文件权限包含任一mode的权限的文件，mode用数字表示

#例如：

```
find / -name passwd # 查找文件名为passwd的文件
```

```
find . -perm 0755 # 查找当前目录中文件权限的0755的文件
```

```
find . -size +12k # 查找当前目录中大于12KB的文件，注意c表示byte
```

5. cp命令

该命令用于复制文件，copy之意，它还可以把多个文件一次性地复制到一个目录下，它的常用参数如下：

- a：将文件的特性一起复制
- p：连同文件的属性一起复制，而非使用默认方式，与-a相似，常用于备份
- i：若目标文件已经存在时，在覆盖时会先询问操作的进行
- r：递归持续复制，用于目录的复制行为
- u：目标文件与源文件有差异时才会复制

例如：

`cp -a file1 file2` #连同文件的所有特性把文件file1复制成文件file2

`cp file1 file2 file3 dir` #把文件file1、file2、file3复制到目录dir中

6. mv命令

该命令用于移动文件、目录或更名，move之意，它的常用参数如下：

- f：force强制的意思，如果目标文件已经存在，不会询问而直接覆盖
- i：若目标文件已经存在，就会询问是否覆盖
- u：若目标文件已经存在，且比目标文件新，才会更新

注：该命令可以把一个文件或多个文件一次移动一个文件夹中，但是最后一个目标文件一定要是“目录”。

例如：

`mv file1 file2 file3 dir` # 把文件file1、file2、file3移动到目录dir中

`mv file1 file2` # 把文件file1重命名为file2

7. rm命令

该命令用于删除文件或目录，remove之间，它的常用参数如下：

- f：就是force的意思，忽略不存在的文件，不会出现警告消息
- i：互动模式，在删除前会询问用户是否操作
- r：递归删除，最常用于目录删除，它是一个非常危险的参数

例如：

`rm -i file` # 删除文件file，在删除之前会询问是否进行该操作

`rm -fr dir` # 强制删除目录dir中的所有文件

8. ps命令

该命令用于将某个时间点的进程运行情况选取下来并输出，process之意，它的常用参数如下：

- A：所有的进程均显示出来
- a：不与terminal有关的所有进程

- u：有效用户的相关进程
- x：一般与a参数一起使用，可列出较完整的信息
- l：较长，较详细地将PID的信息列出

其实我们只要记住ps一般使用的命令参数搭配即可，它们并不多，如下：

```
ps aux # 查看系统所有的进程数据
ps ax # 查看不与terminal有关的所有进程
ps -lA # 查看系统所有的进程数据
ps axjf # 查看连同一部分进程树状态
```

9. kill命令

该命令用于向某个工作（%jobnumber）或者是某个PID（数字）传送一个信号，它通常与ps和jobs命令一起使用，它的基本语法如下：

```
kill -signal PID
```

signal的常用参数如下：

注：最前面的数字为信号的代号，使用时可以用代号代替相应的信号。

- 1：SIGHUP，启动被终止的进程
- 2：SIGINT，相当于输入ctrl+c，中断一个程序的进行
- 9：SIGKILL，强制中断一个进程的进程
- 15：SIGTERM，以正常的结束进程方式来终止进程
- 17：SIGSTOP，相当于输入ctrl+z，暂停一个进程的进程

例如：

#以正常的结束进程方式来终于第一个后台工作，可用jobs命令查看后台中的第一个工作进程

```
kill -SIGTERM %1
```

#重新改动进程ID为PID的进程，PID可用ps命令通过管道命令加上grep命令进行筛选获得

```
kill -SIGHUP PID
```

10. killall命令

该命令用于向一个命令启动的进程发送一个信号，它的一般语法如下：

```
killall [-ile] [command name]
```

它的参数如下：

- i：交互式的意思，若需要删除时，会询问用户
- e：表示后面接的command name要一致，但command name不能超过15个字符
- l：命令名称忽略大小写

#例如：

```
killall -SIGHUP syslogd # 重新启动syslogd
```


11. file命令

该命令用于判断接在file命令后的文件的基本数据，因为在Linux下文件的类型并不是以后缀为分的，所以这个命令对我们来说就很有用了，它的用法非常简单，基本语法如下：

```
file filename
```

#例如：

```
file ./test
```

12. tar命令

该命令用于对文件进行打包，默认情况并不会压缩，如果指定了相应的参数，它还会调用相应的压缩程序（如gzip和bzip等）进行压缩和解压。它的常用参数如下：

-c：新建打包文件

-t：查看打包文件的内容含有哪些文件名

-x：解打包或解压缩的功能，可以搭配-C（大写）指定解压的目录，注意-c,-t,-x不能同时出现在同一条命令中

-j：通过bzip2的支持进行压缩/解压缩

-z：通过gzip的支持进行压缩/解压缩

-v：在压缩/解压缩过程中，将正在处理的文件名显示出来

-f filename：filename为要处理的文件

-C dir：指定压缩/解压缩的目录dir

上面的解说可以已经让你晕过去了，但是通常我们只需要记住下面三条命令即可：

压缩：tar -jcv -f filename.tar.bz2 要被处理的文件或目录名称

查询：tar -jtv -f filename.tar.bz2

解压：tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

注：文件名并不定要以后缀tar.bz2结尾，这里主要是为了说明使用的压缩程序为bzip2

13. cat命令

该命令用于查看文本文件的内容，后接要查看的文件名，通常可用管道与more和less一起使用，从而可以一页页地查看数据。例如：

```
cat text | less # 查看text文件中的内容
```

#注：这条命令也可以使用less text来代替

14. chgrp命令

该命令用于改变文件所属用户组，它的使用非常简单，它的基本用法如下：

```
chgrp [-R] dirname/filename
```

-R：进行递归的持续对所有文件和子目录更改

#例如：

chgrp users -R ./dir # 递归地把dir目录下中的所有文件和子目录下所有文件的用户组修改为users

15. chown命令

该命令用于改变文件的所有者，与chgrp命令的使用方法相同，只是修改的文件属性不同，不再详述。

16. chmod命令

该命令用于改变文件的权限，一般的用法如下：

chmod [-R] xyz 文件或目录

-R：进行递归的持续更改，即连同子目录下的所有文件都会更改

同时，chmod还可以使用u (user)、g (group)、o (other)、a (all) 和+ (加入)、- (删除)、= (设置) 跟rwx搭配来对文件的权限进行更改。

#例如：

chmod 0755 file # 把file的文件权限改变为-rxwr-xr-x

chmod g+w file # 向file的文件权限中加入用户组可写权限

17. vi命令

Vi 在任何 Linux/Unix 发行安装上都存在。因此，我们有必要了解一些基本的命令。Vi 有两个模式：命令模式和插入模式。使用 [ESC] 键可进入命令模式，使用 i 键可进入插入模式。如果你迷失了，可在命令模式下键入 : help。编辑器 nano 和 pico 通常也都可用，而且更容易(IMHO)使用。

:w newfilename 保存文件为 newfilename

:wq or :x 保存并退出

:q! 退出但不保存

/string 向前查找 string

?string 向后查找 string

n 同方向重复上一次搜索命令

N 反方向重复上一次搜索命令

{ 光标移至段落结尾

} 光标移至段落开头

1G 光标移至文件的第一行首

nG 光标移至文件的第 n 行首

G 光标移至文件的最后一行首

:%s/OLD/NEW/g 替换所有查找到的 OLD 为 NEW

dd 删除当前行

D 删除光标到当前行末尾的字符

dw 删除单词

x 删除字符

u 回复上一次操作

U 回复所有此行的更改

18. vim命令

该命令主要用于文本编辑，它接一个或多个文件名作为参数，如果文件存在就打开，如果文件不存在就以该文件名创建一个文件。vim是一个非常好用的文本编辑器，它里面有很多非常好用的命令，在这里不再多说。后期单介绍它。

19. gcc命令

对于一个用Linux开发C程序的人来说，这个命令就非常重要了，它用于把C语言的源程序文件，编译成可执行程序，由于g++的很多参数跟它非常相似，所以这里只介绍gcc的参数，它的常用参数如下：

-o：output之意，用于指定生成一个可执行文件的文件名

-c：用于把源文件生成目标文件（.o），并阻止编译器创建一个完整的程序

-I：增加编译时搜索头文件的路径

-L：增加编译时搜索静态连接库的路径

-S：把源文件生成汇编代码文件

-lm：表示标准库的目录中名为libm.a的函数库

-lpthread：连接NPTL实现的线程库

-std=：用于指定把使用的C语言的版本

#例如：

#把源文件test.c按照c99标准编译成可执行程序test

gcc -o test test.c -lm -std=c99

#把源文件test.c转换为相应的汇编程序源文件test.s

gcc -S test.c

20. time命令

该命令用于测算一个命令（即程序）的执行时间。它的使用非常简单，就像平时输入命令一样，不过在命令的前面加入一个time即可，例如：

time ./process

time ps aux

在程序或命令运行结束后，在最后输出了三个时间，它们分别是：

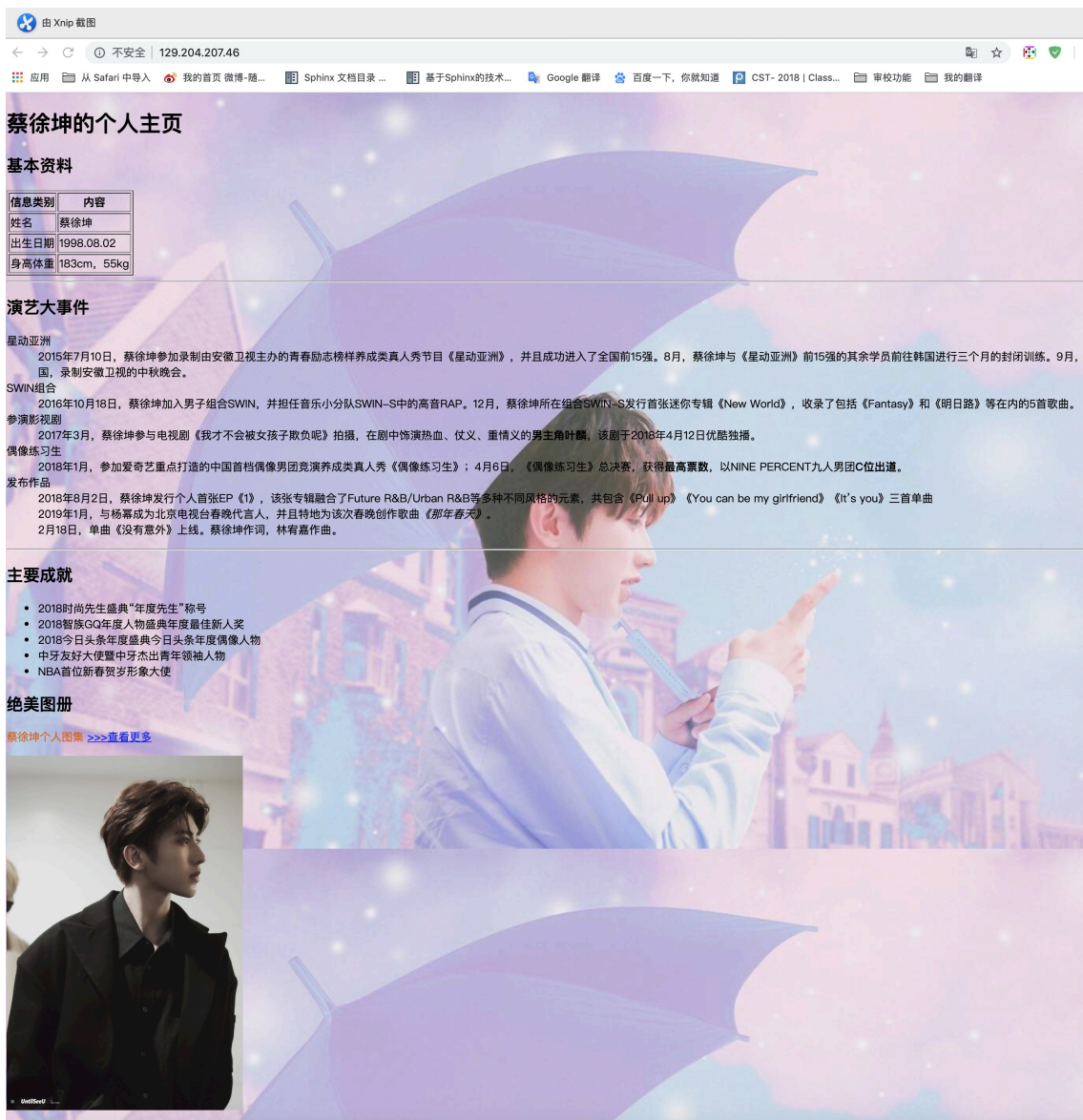
user：用户CPU时间，命令执行完成花费的用户CPU时间，即命令在用户态中执行时间总和；

system：系统CPU时间，命令执行完成花费的系统CPU时间，即命令在核心态中执行时间总和；

real：实际时间，从command命令行开始执行到运行终止的消逝时间；

注：用户CPU时间和系统CPU时间之和为CPU时间，即命令占用CPU执行的时间总和。实际时间要大于CPU时间，因为Linux是多任务操作系统，往往在执行一条命令时，系统还要处理其它任务。另一个需要注意的问题是即使每次执行相同命令，但所花费的时间也是不一样，其花费时间是与系统运行相关的。

5. 接下来，通过浏览器即可使用ip地址访问文件。



附注：服务器的选择

根据查询，网上资料表示，相对于apache服务器，Nginx具有轻量级、抗并发、社区活跃的优点。

相同点：

1. 都是HTTP服务器软件
2. 功能上都采用模块化结构设计
3. 都支持通用的语言接口，如PHP、Perl、Python等
4. 支持正向、反向代理、虚拟主机、URL重写、压缩传输、SSL加密传输

不同点：

5. Apache处理速度很慢，占用很多内存资源
6. 功能上，Apache所有模块支持动静静态编译，Nginx模块都是静态编译的
7. 对Fcgi的支持：Apache支持的很不好，Nginx支持非常好
8. 处理连接方式：Nginx支持epoll,Apache不支持
9. 空间使用上：Nginx安装包仅几百K

来源：https://blog.csdn.net/qq_35216516/article/details/80529371

本次作业选择Nginx服务器。

<http://129.204.207.46/>