# 一、常用的web服务器

**apache**:功能完善，历史悠久

缺点：处理每一个php比较费资源，导致如果高并发时会太耗费服务器资源无法处理更多请求。

**nginx：省资源，省cpu,所以在高并发时能够处理更多的请求，**高端能达到3万到5万的并发量。

# 二、nginx介绍

Nginx（”engine x”）是俄罗斯人编写的一款高性能的HTTP和反向代理服务器。Nginx是Apache服务器不错的替代品，它能够支持高达50000个并发连接数的响应，而内存，cpu等系统资源消耗却非常低，运行非常稳定。

在国内，已经有新浪博客，新浪播客，网易新闻，六间房，56.com.discuz!官方论坛，水木社区，豆瓣，海内SNS，迅雷在线等多家网站使用Nginx作为web服务器或反向代理服务器。

主要优势：省资源，能够处理高并发，运行稳定。

# 三、nginx的特点

**1、它可以高并发连接**，官方测试能够支撑5万并发连接，在实际生产环境中可以支撑2到4万并发连接。

**2、内存消耗少**

Nginx+php(FastCGI)服务器再3万并发连接下，开启的10个Nginx进程消耗150MB内存（15MB\*10=150MB）开启的64个php-cgi进程消耗1280MB内存（20MB\*64=1280MB）

**3、成本低廉**

购买F5 BIG-IP ,NetScaler等硬件负载均衡交换机需要10多万甚至几十万人民币。而Nginx为开源软件，可以免费试用，并且可用于商业用途。

4、其他理由：

（1）配置文件非常简单：通俗易懂，即使非专业管理员也能看懂。

（2）支持 rewrite重写规则：能根据域名、URL的不同，将HTTP请求分到不同的后端服务器群组。

（3）内置的健康检查功能：如果nginx proxy后端的某台服务器宕机了，不会影响前端访问。

（4）节省带宽，支持gzip压缩。

（5）稳定性高：用于反向代理，宕机的概率微乎其微。

（6）支持热部署。在不间断服务的情况下，对软件版本升级。

# 四、支持多个操作系统

FreeBSD 3.x,4.x,5.x,6.x i386; FreeBSD 5.x,6.x amd64;

Linux 2.2,2.4,2.6 i386; Linux 2.6 amd64;

Solaris 8 i386; Solaris 9 i386 and sun4u; Solaris 10 i386;

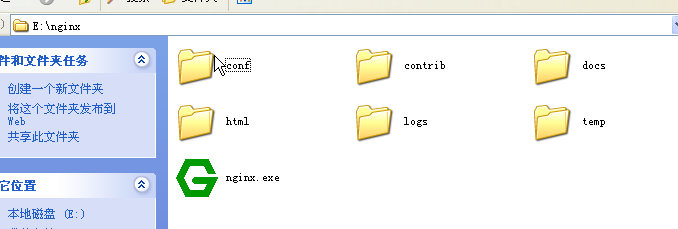
MacOS X （10.4） PPC;

Windows XP，Windows Server 2003和Windows 7等。

# 五、安装

1、下载解压软件到指定的目录，解压后的目录如下：

配置文件的目录



默认的网站根目录

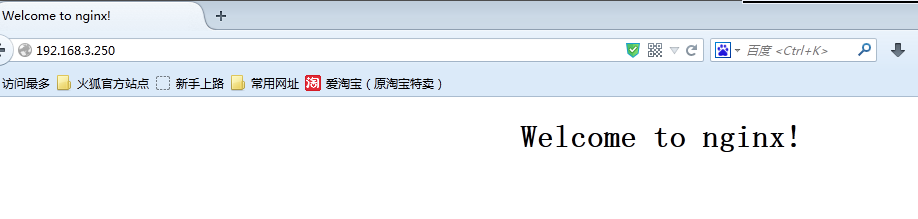
执行命令的文件，比如开启关闭

把该目录拷贝到指定的目录即可完成安装（绿色安装）

2、启动

进入cmd输入start nginx 完成启动。

3、在浏览器输入（在本地输入localhost）出现如下，表明已经成功启动。



# 六、配置虚拟主机

打开nginx.conf配置文件。

里面：

配置文件整体的结构：

**events{**

**}**

**http{**

**server{**

**//配置项**

**}**

**server{**

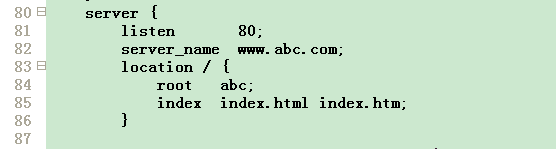
**//配置项**

**}**

**}**

配置步骤：

（1）每一个server{}代码段是一个虚拟主机，我们要配置虚拟主机，添加server{}段即可。

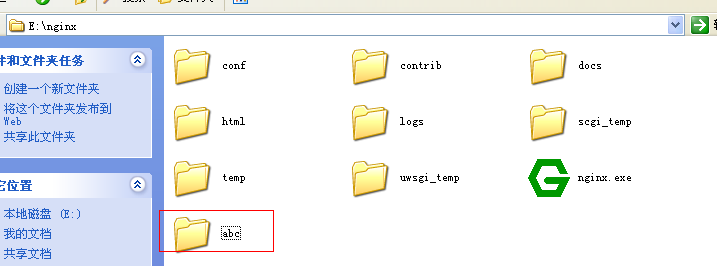


网站的根目录，我们也可以写绝对路径的。如果写相对路径则是和默认的html目录同一级目录。

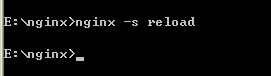
默认 的首页

虚拟主机的域名

（2）根据配置文件，添加网站的目录



（3）要重新加载配置文件。



（4）修改hosts文件，访问虚拟主机

（5）在浏览器中直接输入虚拟主机的域名即可。



nginx的相关的一些命令：

start nginx//启动nginx服务。

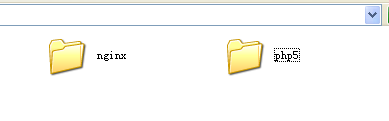
nginx -s stop // 停止nginx  
nginx -s reload // 重新加载配置文件  
nginx -s quit // 退出nginx

# 七、配置支持php

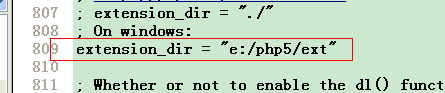
在默认情况下，nginx是无法支持php的。

当 nginx碰到php结尾的文件是，要交给9000端口，由9000端口调用php处理。

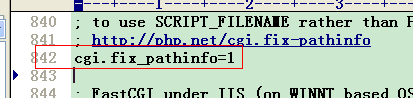
（1）把php的软件拷贝到指定的目录，一般和nginx是同一个目录



（2）打开php的软件，把php.ini-deve改成php.ini文件。并打开配置文件，配置加载扩展的目录

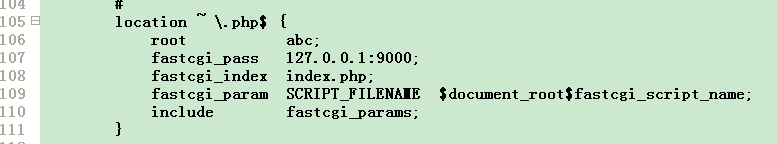


（3）打开该选项：开启9000端口。



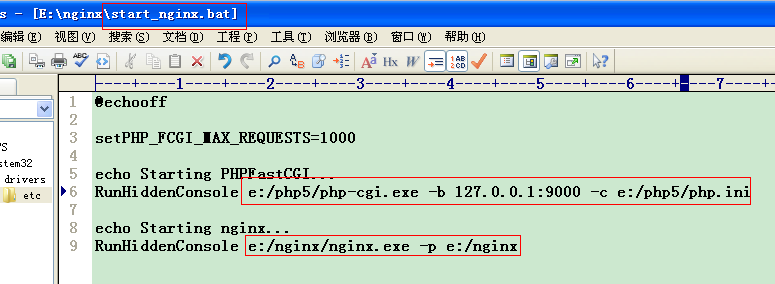
表示碰到php文件是如何处理

**（4）打开nginx.conf配置文件，修改如下配置。**



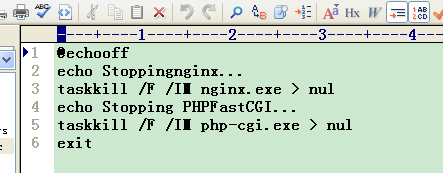
**RunHiddenConsole一个让你的程序隐藏运行的小工具**

启动脚本里面的内容：



该路径要和当前的安装位置一致，不能照抄。

停止脚本里面的内容：



（6）运行，start\_nginx.bat脚本，开始写一个php文件进行测试：



# 八、配置nginx expires缓存功能

## 1、整体配置有效期：

**对于图片，css,js等元素更改机会较少，特别是图片，可以将图片设置在浏览器本地缓存365天，css,js缓存10天，这样可以提高下次打开用户页面加载速度，并节省大量带宽。此功能同apache的expires。这里通过location的功能，将需要缓存的扩展名列出来，然后指定缓存时间：**

**location ~.\*\.(gif|jpg|jpeg|png|bmp)$**

**{**

**expires 365d;**

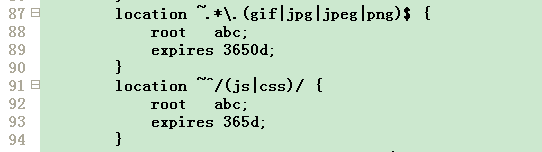
**}**

## 2、根据目录进行判断，添加expires功能

**location ~ ^/(js|css|)/ {**

**expires 360d;**

**}**

****

# 九、压缩配置：

**gzip on;**

**#开启gzip压缩功能**

**gzip\_min\_length 1k;**

**#设置允许压缩的页面最小字节数，页面字节数从header头的content-length中获取。默认值是0,不管页面多大都进行压缩。建议设置成大于1k。如果小于1k可能会越压越大。**

**gzip\_buffers 4 16k;**

**#压缩缓冲区大小。表示申请4个单位为16k的内容作为压缩结果流缓存，默认值是申请与原始数据大小相同的内存空间来存储gzip压缩结果。**

**gzip\_http\_version 1.0;**

**#压缩版本（默认1.1，前端为squid2.5时使用1.0）用于设置识别http协议版本，默认是1.1,目前大部分浏览器已经支持gzip解压，使用默认即可。**

**gzip\_comp\_level 2;**

**#压缩比率。用来指定gzip压缩比，1压缩比量小，处理速度快；9压缩比量大，传输速度快，但处理最慢，也必将消耗cpu资源。**

**gzip\_types text/plain application/x-javascript text/css application/xml;**

**#用来指定压缩的类型，“text/html”类型总是会被压缩。**

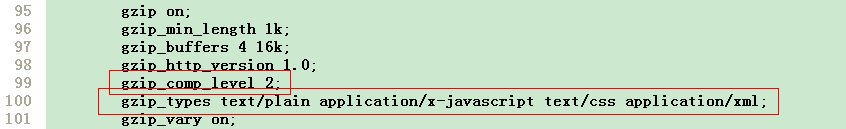
**gzip\_vary on;**

**#vary header支持。该选项可以让前端的缓存服务器缓存经过gzip压缩的页面，例如用squid缓存经过nginx压缩的数据。**

**要注意：需要和不需要压缩的对象**

**（1）大于1k的纯文本文件html,js,css,xml,html.**

**（2）图片，视频等不要压缩，因为不但不会减小，在压缩时消耗cpu和内存资源。**

****

# 十、自动列表目录配置

**经常会看到一些开源软件的下载页面是能够自动列目录的。nginx 的同样可以实现，前提条件是当前目录下不存在用index指令设置的默认首页文件。如果需要在某一虚拟主机的location /{…}目录控制中配置自动列目录，只须加上如下代码。**

**location /{**

**autoindex on;**

**}**

# 十一、错误页面跳转配置

什么是404页面：

如果碰巧网站出了问题，或者用户试图访问一个并不存在的页面时，此时服务器会返回代码404的错误信息，此时对应页面就是404页面。404页面的默认内容和具体的服务器有关。如果后台用的是nginx服务器，那么404页面的内容则为：404 not found

nginx 下如何自定义404页面。

（1）创建自己的404.html页面

（2）更改nginx.conf在http定义区域加入：fastcgi\_intercept\_errors on;

（3）更改nginx.conf中在server区域加入：error\_page 404 /404.html或者error\_page 404 <http://www.XXX.com>

（4）更改后重启nginx，测试nginx.conf正确性：/opt/nginx/sbin/nginx –t

#502等错误可以用同样的方法来配置。

注意事项：

（1）必须要添加：**fastcgi\_intercept\_errors on**;如果这个选项没有设置。即使创建了404.html和配置了error\_page也没有效果。fastcgi\_intercept\_errors语法：fastcgi\_intercept\_errors on|off默认：fastcgi\_intercept\_errors off添加位置：http,server,location默认情况下，nginx不支持自定义404错误页面，只有这个指令被设置为on.nginx才支持将404错误重定向。这里需要注意的是，并不是说设置了fastcgi\_intercept\_errors on,nginx就会将404错误重定向。在nginx中404错误重定向生效的前提是设置了fastcgi\_intercept\_errors on,并且正确的设置了error\_page这个选项（包括语法和对应的404页面）

（2）不要出于省事或者提高首页权重的目的将首页指定为404错误页面，也不要用其他方法跳转到首页。

（3）自定义的404页面必须大于512字节，否则可能会出现ie默认的404页面。

如果配置404错误，则必须添加该选项，才能正常跳转



# 十二、nginx支持thinkphp

把原来支持，php代码替换成如下代码：

location ~ .+\.php($|/) {

set $script $uri;

set $path\_info "/";

if ($uri ~ "^(.+\.php)(/.+)") {

set $script $1;

set $path\_info $2;

}

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php?IF\_REWRITE=1;

#不要照抄，要注意要和conf目录里面的fcgi.conf配置文件名称一致。

include fcgi.conf;

fastcgi\_param PATH\_INFO $path\_info;

fastcgi\_param SCRIPT\_FILENAME $document\_root/$script;

fastcgi\_param SCRIPT\_NAME $script;

}