|  |
| --- |
| 文章标题  ngnix |
| **作者：**  **归档：学习笔记** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文 |
| **格式说明：**  蓝色字体：注释  黄色背景：重要  绿色背景：注意 |

****

**老男孩教育教学核心思想6重：重目标、重思路、重方法、重实践、重习惯、重总结**

**学无止境，老男孩教育成就你人生的起点！**

课程回顾：

01. HTTP协议原理（请求报文--请求行 请求头部 空行 请求主体 响应报文---起始行 响应头部 空行 响应主体）

02. HTTP请求报文里面：请求方法 get post

03. HTTP响应报文里面：200 ok 301 永久跳转 302 临时跳转 404 not found 403 客户端访问禁止 502 坏的网关

04. www.baidu.com/asdhoasdhaosd/sadasd www.baidu.com/ URL /asdhoasdhaosd/sadasd URI

05. 静态页面（客户端解析的页面---便于被搜索引擎收录） 动态页面（服务端解析的页面--实现人机交互---数据库配合）

将动态页面--->静态页面 伪静态页面

06. 三种网站的度量方法：IP PV UV(cookie session)

07. 网站并发是什么---（单位时间内，网站可以处理的请求数）

# nginx介绍

### nginx是什么

Nginx（“engine x”）是一个开源的，支持高性能、高并发的WWW服务器和代理服务软件

Nginx因具有高并发（特别是静态资源）、占用系统资源少等特性，且功能丰富而逐渐流行起来。

Nginx可以运行在UNIX、Linux、BSD、Mac OS X、Solaris，以及Microsoft Windows等操作系统中。

当前流行的Nginx Web组合被称为LNMP或LEMP

L==Linux N/E==Nginx M==MySQL P==php 网站的代码程序 ===搭建出一个完整的网站、

### nginx软件特性说明

1）. 支持高并发：能支持几万并发连接（特别是静态小文件业务环境）

2）. 资源消耗少：在3万并发连接下，开启10个Nginx线程消耗的内存不到200MB

3）. 可以做HTTP反向代理及加速缓存、即负载均衡功能（4层以及7层），

内置对RS节点服务器健康检查功能，这相当于专业的Haproxy软件或LVS(4层)的功能。

4) . 具备Squid等专业缓存软件等的缓存功能。（memcache/redis）

5）. 支持异步网络I/O事件模型epoll（Linux 2.6+） ?

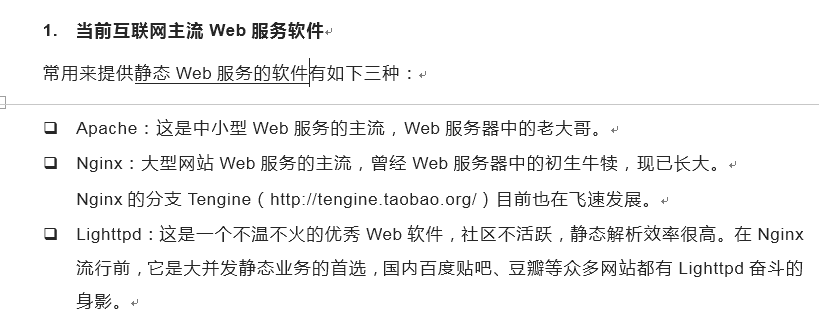
### nginx软件有哪些应用

1) 作为Web服务软件

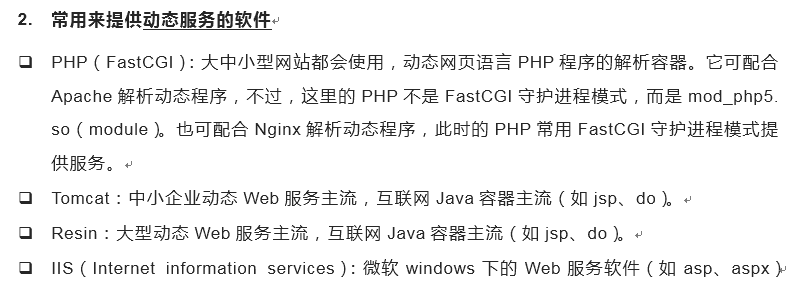
2) 反向代理或负载均衡服务

3) 前端业务数据缓存服务（扩展redis缓存知识后）

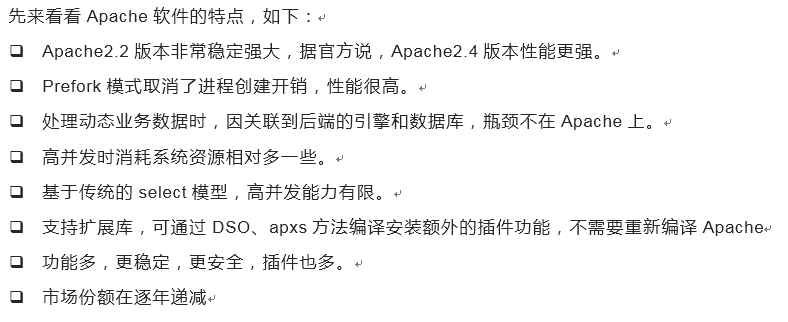
### 提供静态web服务的软件



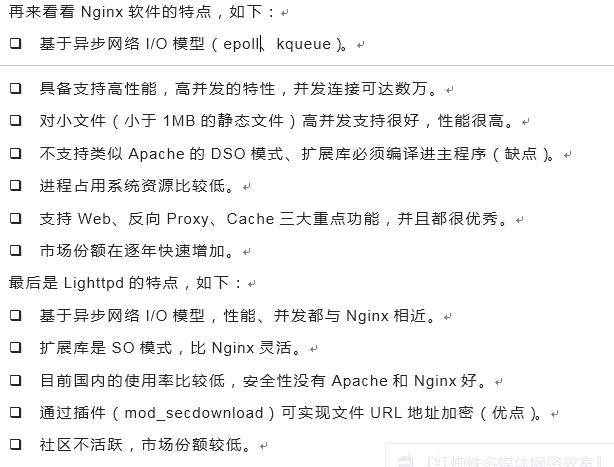
### 常用来提供动态服务的软件



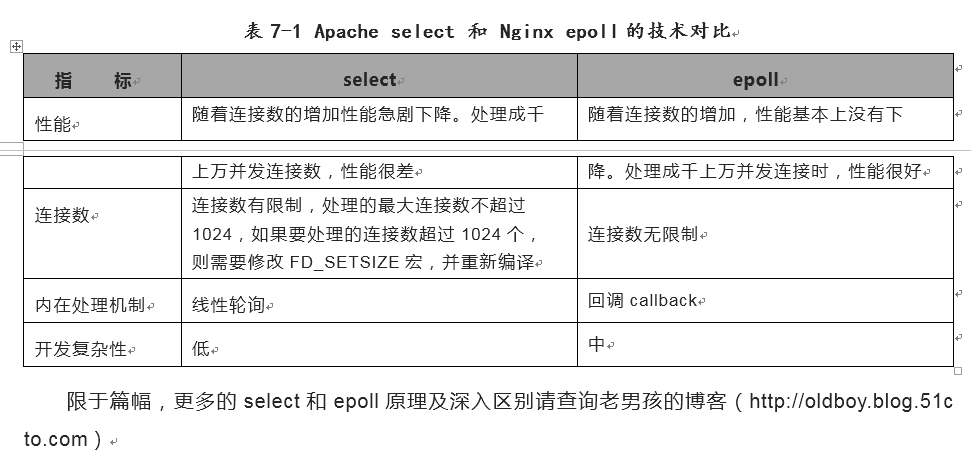
#### Apache软件特点



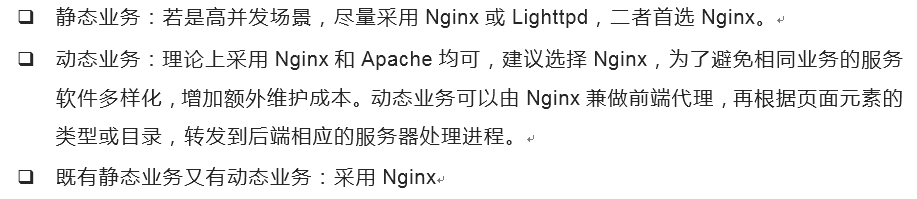
#### Nginx 软件特点



#### Apache select 和 Nainx epoll 的技术对比



#### 首选Nainx



# Nainx的编译安装

## 检查软件安装的系统环境

[root@web01 conf]# cat /etc/redhat-release

CentOS release 6.9 (Final)

[root@web01 conf]# uname -r

2.6.32-696.el6.x86\_64

## 安装nginx的依赖包（pcre-devel openssl-devel）

yum install -y pcre-devel openssl-devel

## 下载nginx软件---1.10.2 复制链接地址（统一位置进行下载）

mkdir -p /home/oldboy/tools 创建一个文件用来存放下载的软件

cd /home/oldboy/tools

wget -q http://nginx.org/download/nginx-1.10.3.tar.gz

## 编译安装软件步骤

### 解压要编译安装的软件(解压软件---配置（./configure）

#### 解压

tar xf nginx-1.10.3.tar.gz

[root@web01 ~]# ll tools/

total 4

drwxr-xr-x 9 1001 1001 4096 Sep 11 15:56 nginx-1.10.3

#### 切换目录

cd nginx-1.10.3

ls (里面的内容就是源代码（config readme安装说明）---默认编译会安装到/usr/local目录)

#### 创建web服务程序www用户

useradd -s /sbin/nologin -M www

[root@web01 nginx-1.10.3]# id www

uid=500(www) gid=500(www) groups=500(www)

#### 安装

./configure --prefix=/application/nginx-1.10.3 --user=www --group=www --with-http\_stub\_status\_module --with-http\_ssl\_module (编译参数说明后续补充说明)

编译安装

make

make install

软件编译过程中，利用返回值，判断操作是否正确

echo $?

0

#### 安装完成一个软件要做一个软链接

ln -s /application/nginx-1.10.3 /application/nginx

[root@web01 nginx-1.10.3]# cd /application/

[root@web01 application]# ll

total 4

lrwxrwxrwx 1 root root 26 Sep 11 16:10 nginx -> /application/nginx-1.10.3/

drwxr-xr-x 11 root root 4096 Sep 11 16:16 nginx-1.10.3

### 启动nginx软件程序进行测试

#### 启动

/applicaton/nginx/sbin/nginx

#### 查看端口

[root@web01 application]# lsof -i:80

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

nginx 4633 root 6u IPv4 19980 0t0 TCP \*:http (LISTEN)

nginx 4634 www 6u IPv4 19980 0t0 TCP \*:http (LISTEN)

#### 浏览器访问本机IP

至此软件安装完毕：

# nginx软件的编译安装常见错误说明

## nginx软件安装过程中遇到的问题

### 软件依赖包未正确安装问题---PCRE依赖包没有安装

./configure: error: the HTTP rewrite module requires the PCRE library.

You can either disable the module by using --without-http\_rewrite\_module

option, or install the PCRE library into the system, or build the PCRE library

statically from the source with nginx by using --with-pcre=<path> option.

解决方法：yum install pcre pcre-devel -y

### 软件依赖包未正确安装问题---OPENSSL依赖包没有安装

./configure: error: SSL modules require the OpenSSL library.

You can either do not enable the modules, or install the OpenSSL library

into the system, or build the OpenSSL library statically from the source

with nginx by using --with-openssl=<path> option.

解决方法：yum install openssl openssl-devel -y

## nginx软件启动过程中遇到的问题

### nginx软件重复启动产生的错误信息

[root@web01 nginx-1.10.2]# /application/nginx/sbin/nginx

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] still could not bind()

解决方法：nginx软件已经启动无需反复启动，如果需要重新启动需要停止nginx进程或者用reload方式进行重启

# nginx 启动方法及配置文件

## nginx命令说明

### nginx 启动方法

[root@web01 application]# /application/nginx/sbin/nginx

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] still could not bind()

### 4.3 nginx停止方法

[root@web01 application]# /application/nginx/sbin/nginx -s stop

[root@web01 application]# ps -ef|grep nginx

root 4283 1773 0 21:52 pts/0 00:00:00 grep --color=auto nginx

[root@web01 application]# netstat -lntup|grep 80

[root@web01 application]# /application/nginx/sbin/nginx

### nginx 重启方法（平滑）

[root@web01 application]# /application/nginx/sbin/nginx -s reload

### nginx 配置文件语法检查

[root@web01 application]# /application/nginx/sbin/nginx -t

nginx: the configuration file /application/nginx-1.10.3/conf/nginx.conf syntax is ok

nginx: configuration file /application/nginx-1.10.3/conf/nginx.conf test is successful

### nginx 查看软件编译安装配置参数信息

[root@web01 application]# /application/nginx/sbin/nginx -V

nginx version: nginx/1.10.3

built by gcc 4.4.7 20120313 (Red Hat 4.4.7-18) (GCC)

built with OpenSSL 1.0.1e-fips 11 Feb 2013

TLS SNI support enabled

configure arguments: --prefix=/application/nginx-1.10.3 --user=www --group=www --with-http\_stub\_status\_module --with-http\_ssl\_module

### 查看可以使用的参数

[root@web01 application]# /application/nginx/sbin/nginx -h

nginx version: nginx/1.10.3

Usage: nginx [-?hvVtTq] [-s signal] [-c filename] [-p prefix] [-g directives]

Options:

-?,-h : this help

-v : show version and exit

-V : show version and configure options then exit

-t : test configuration and exit

-T : test configuration, dump it and exit

-q : suppress non-error messages during configuration testing

-s signal : send signal to a master process: stop, quit, reopen, reload

-p prefix : set prefix path (default: /application/nginx-1.10.3/)

-c filename : set configuration file (default: conf/nginx.conf)

-g directives : set global directives out of configuration file

## nginx 配置文件

### nginx软件目录结构说明

[root@web01 nginx]# ll

total 36

drwxr-xr-x 2 root root 4096 Sep 8 21:22 conf ---配置文件保存目录

drwxr-xr-x 2 root root 4096 Sep 8 21:22 html ---站点目录

drwxr-xr-x 2 root root 4096 Sep 8 21:52 logs ---nginx服务相关日志文件保存目录（错误日志 访问日志）

drwxr-xr-x 2 root root 4096 Sep 8 21:22 sbin ---服务命令保存目录（只有一个nginx命令文件）

### 精简化nginx.conf主配置文件内容

egrep -v "#|^$" nginx.conf.default >nginx.conf

### 默认nginx程序配置信息

# vim nginx.conf

1 worker\_processes 1;

2 events {

3 worker\_connections 1024;

4 }

5 http {

6 include mime.types;

7 default\_type application/octet-stream;

8 sendfile on;

9 keepalive\_timeout 65;

10 server {

11 listen 80;

12 server\_name localhost;

13 location / {

14 root html;

15 index index.html index.htm;

16 }

17 error\_page 500 502 503 504 /50x.html;

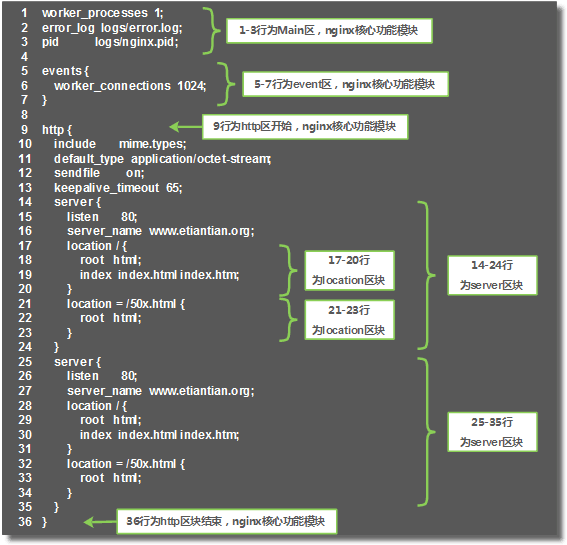
18 location = /50x.html {

19 root html;

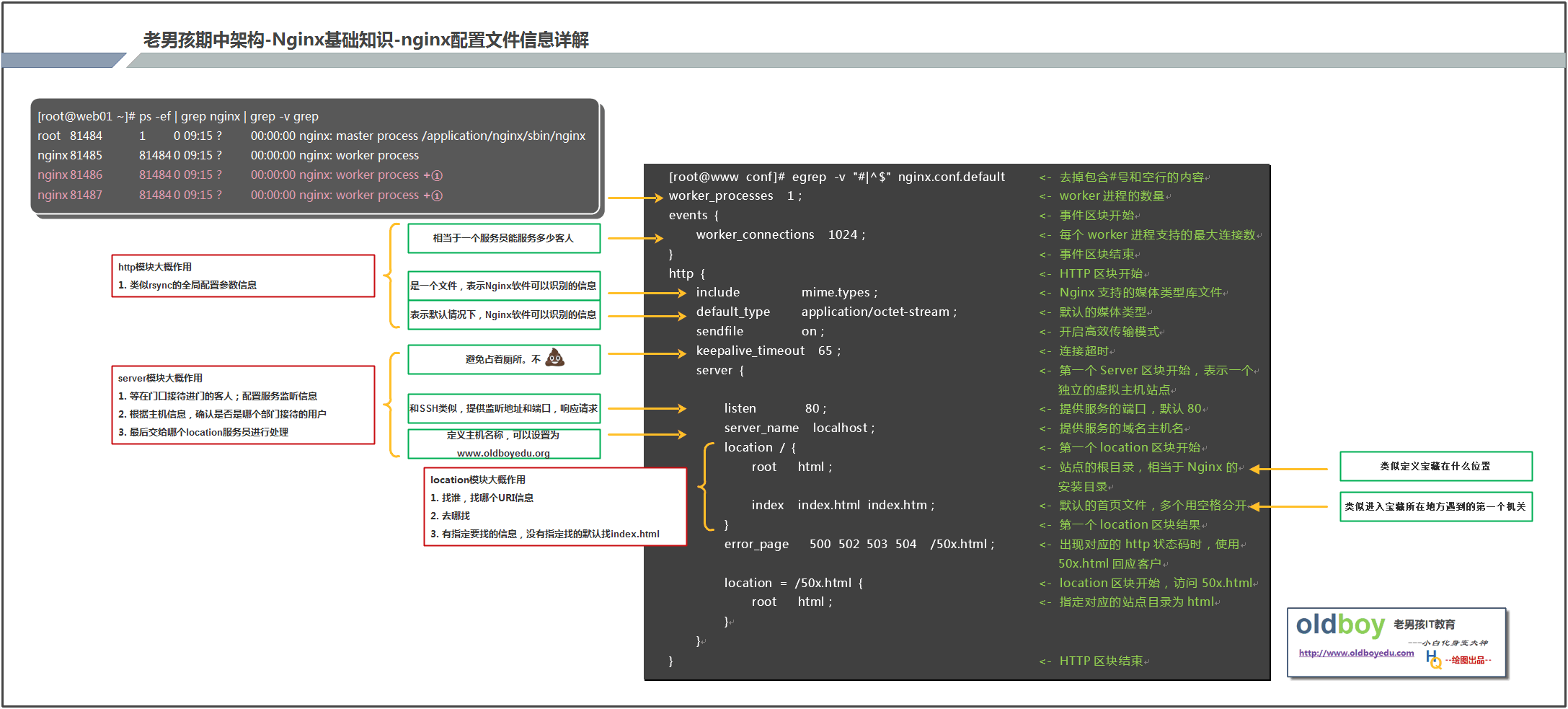
20 }

21 }

22 }



### nginx配置文件内容



# 模拟搭建网站

## 搭建静态网页

### 第一步：编辑nginx配置文件

worker\_processes 3;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm;

}

error\_page 500 502 503 504 404 /oldboy.jpg;

location = /oldboy.jpg {

root html;

}

}

}

### 第二步：生成站点目录，创建首页文件

cd /application/nginx/html/

mkdir www 创建一个站点目录

cat >>index.html<<EOF

<html>

<meta charset="utf-8">

<head>

<title>老男孩教育39期最牛</title>

</head>

<body>

老男孩教育39期最牛

<table border=1>

<tr> <td>01</td> <td> </td> </tr>

<tr> <td>02</td> <td> </td> </tr>

<tr> <td>03</td> <td> </td> </tr>

</table>

<a href="http://blog.oldboyedu.com">

<img src="oldboy.jpg" />

</a>

</body>

</html>

EOF

### 第三步：重启nginx服务

/application/nginx/sbin/nginx -t

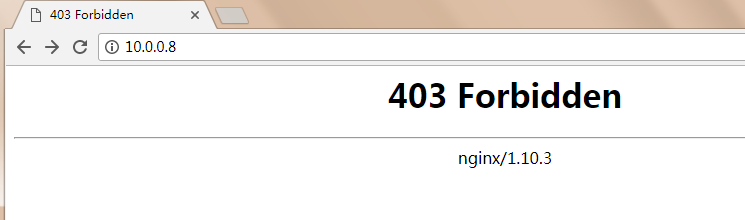
/application/nginx/sbin/nginx -s reload

### 第四步：访问测试

利用浏览器访问测试，如果通过域名访问，需要修改本地hosts文件

## 403错误

### 当站点目录下没有index.html文件时，会报错403



### 解决办法，指定了autoindex on; 参数

#### 当nginx配置文件中指定了autoindex on; 参数；当首页文件不存在时。浏览网页会显示站点目录下所有文件或目录内容

可以被nginx服务处理的文件资源，点击后直接显示文件内容

不可以碑nginx服务处理的文件资源，会通过浏览器直接下载

#### 将首页文件改名或者删除

[root@web01 www]# ll

total 8

-rw-r--r-- 1 root root 0 Sep 12 19:13 index.html.a

-rw-r--r-- 1 root root 296 Sep 12 10:29 index.html.bak

-rw-r--r-- 1 root root 10 Sep 12 14:42 oldboy.txt

#### 查看配置文件中设置

[root@web01 www]# cat /application/nginx/conf/nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm; 默认为index.html文件，但此时www目录下没有，访问网页则会出现错误

autoindex on;

}

}

#### 指定了autoindex on; 参数

[root@web01 www]# cat /application/nginx/conf/nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

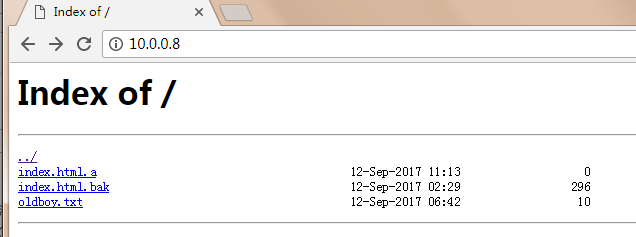
index index.html index.htm;

autoindex on;

}

}

#### 浏览网页会显示站点目录下所有文件或目录内容



# Nginx实践

## nginx配置文件介绍

### 配置文件说明

[root@web02 ~]# cat /application/nginx/conf/nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.etiantian.org; <- 修改虚拟主机域名名称

location / {

root html/www; <- 修改虚拟主机站点目录

index index.html index.htm;

}

}

}

### 创建站点，检查语法，编辑域名

[root@web02 ~]# mkdir /application/nginx/html/www/ -p <- 创建虚拟主机的站点目录

[root@web02 ~]# /application/nginx/sbin/nginx -t <- 检查配置文件语法信息是否正确

[root@web02 ~]# /application/nginx/sbin/nginx -s reload <- 检查语法配置正确后，进行优雅重启nginx服务

[root@web02 ~]# cat /etc/hosts <- 编辑域名解析信息，便于linux服务器通过域名访问主机站点目录下资源信息

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

10.0.0.7 www.etiantian.org bbs.etiantian.org

[root@web02 ~]# curl www.etiantian.org <- 利用curl命令本地检测nginx配置是否成功

进行curl时，报403错误，因为没有首页文件信息

[root@web02 ~]# echo 'web01 www' > /application/nginx/html/www/index.html

<- 在虚拟主机指定的站点目录中创建首页文件 [root@web02 ~]# curl www.etiantian.org <- 利用curl命令本地检测nginx配置是否成功；已经存在首页文件，测试成功

## 进行搭建多个网站(搭建多个虚拟主机，一个网站就是一个虚拟主机（server）)

扩展知识：nginx配置文件编辑技巧

可以利用%在server一对花括号之间进行切换

### 修改配置文件，添加多个server主机标签

[root@web02 ~]# cat /application/nginx/conf/nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name www.etiantian.org; <- 修改虚拟主机域名名称

location / {

root html/www; <- 修改虚拟主机站点目录

index index.html index.htm;

}

}

server {

listen 80;

server\_name bbs.etiantian.org; <- 修改虚拟主机域名名称

location / {

root html/bbs; <- 修改虚拟主机站点目录

index index.html index.htm;

}

}

server {

listen 80;

server\_name blog.etiantian.org; <- 修改虚拟主机域名名称

location / {

root html/blog; <- 修改虚拟主机站点目录

index index.html index.htm;

}

}

}

说明：利用vim复制命令，复制创建多个server主机标签；即x,ycopyz---10,17copy17

nginx配置文件每一行参数结尾都需要输入分号;

### 创建不同网站域名站点目录

[root@web02 ~]# mkdir /application/nginx/html/{www,bbs,blog} -p

### 创建不同站点目录，并利用for循环创建不同站点目录下的首页文件，以及利用for循环进行验证不同站点目录下首页文件内容是否不同

[root@web02 ~]# for name in www bbs blog ;do echo web01 ${name}.etiantian.org >/application/nginx/html/$name/index.html ;done

[root@web02 ~]# for name in www bbs blog ;do cat /application/nginx/html/$name/index.html ;done

### 编辑完多个虚拟主机信息，进行重启nginx服务（检查语法，平滑重启）

[root@web02 ~]# /application/nginx/sbin/nginx -t <- 检查配置文件语法信息是否正确

[root@web02 ~]# /application/nginx/sbin/nginx -s reload <- 检查语法配置正确后，进行优雅重启nginx服务

### 利用curl命令本地检测nginx配置是否成功

[root@web02 ~]# curl [www.etiantian.org](http://www.etiantian.org)

web01 www.etiantian.org

[root@web02 ~]# curl bbs.etiantian.org

web01 bbs.etiantian.org

[root@web02 ~]# curl blog.etiantian.org

web01 blog.etiantian.org

说明：利用curl命令进行测试三个虚拟主机是否已经可以正确访问；需要注意hosts文件中已经对三个域名进行了统一解析

以上配置虚拟主机方式为---基于域名的方式配置虚拟主机

### 总结：基于域名多虚拟主机配置

1. 多个网站的目录 html/{www,bbs,blog}

2. 配置文件 域名配置（server\_name） 网站的站点目录配置（root）

## 虚拟主机配置说明---基于端口配置

理解基于端口虚拟主机概念；主要用于内部网络人员访问虚拟主机使用，不想提供直接对外访问的虚拟主机

### 修改配置文件，修改端口

[root@www conf]# cat nginx.conf

worker\_processes 1；

events {

worker\_connections 1024；

}

http {

include mime.types；

default\_type application/octet-stream；

sendfile on；

keepalive\_timeout 65；

server {

listen 80；

server\_name www.etiantian.org；

location / {

root html/www；

index index.html index.htm；

}

}

server {

listen 81； #←由基于域名的80改为基于端口的81

server\_name bbs.etiantian.org；

location / {

root html/bbs；

index index.html index.htm；

}

}

server {

listen 80；

server\_name blog.etiantian.org；

location / {

root html/blog；

index index.html index.htm；

}

}

}

说明：通过修改不同server标签中的监听端口信息，实现基于端口的方式访问不同虚拟主机；精简化配置：只改动bbs虚拟主机端口，其它主机端口不变

### 检查语法，重启服务

#### 检查语法

[root@web02 ~]# /application/nginx/sbin/nginx -t

#### 重启服务

[root@web02 ~]# /application/nginx/sbin/nginx -s reload

#### 查看修改后的端口

[root@web02 ~]# ss -lntup|grep nginx

### 利用curl命令本地检测nginx配置是否成功；

[root@web02 ~]# curl www.etiantian.org

[root@web02 ~]# curl bbs.etiantian.org

说明：显示结果均为www.etiantian.org域名站点目录下的首页文件内容

#### 利用curl命令-v参数获取访问网站流程

[root@web02 ~]# curl -v bbs.etiantian.org

a. 访问网站时首先需要根据域名解析获取到网站的ip地址，找寻网站的ip地址对应的服务器

b. 访问网站时其次需要根据请求建立连接的目标端口信息，找寻网站的相应服务端口是否存在

c. 访问网站时再次需要根据请求域名信息获悉相应的站点，找寻网站的相应站点目录下的资源信息

d. 访问网站时最后如果ip地址加端口信息都已找到，但没有找到对应的域名信息，会按照默认原则使用第一个虚拟主机作为默认访问的虚拟站点目录

#### 利用curl命令本地检测nginx配置是否成功

[root@web02 ~]# curl http://bbs.etiantian.org:81

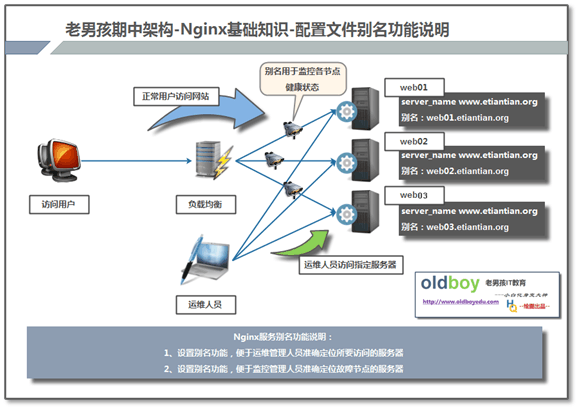
#### 利用curl命令本地检测nginx配置是否成功

[root@web02 ~]# curl http://www.etiantian.org:81

说明：以上信息为在linux系统中进行验证检测基于端口的虚拟主机配置；建议可以利用windos环境再进行相应测试；并且利用抓包工具进行抓包查看

以上配置虚拟主机方式为---基于端口的方式配置虚拟主机(抓包查看基于端口访问过程)

## 虚拟主机配置说明---基于ip配置



理解基于ip虚拟主机概念；回顾地址监听概念

### 修改配置文件

[root@www conf]# cat nginx.conf

worker\_processes 1；

events {

worker\_connections 1024；

}

http {

include mime.types；

default\_type application/octet-stream；

sendfile on；

keepalive\_timeout 65；

server {

listen 10.0.0.8:80；

server\_name www.etiantian.org; <- 此处也可以改成对应IP 10.0.0.8

location / {

root html/www；

index index.html index.htm；

}

}

# server {

# listen 10.0.0.9:81；

# server\_name bbs.etiantian.org; <- 此处也可以改成对应IP 10.0.0.9

# location / {

# root html/bbs；

# index index.html index.htm；

# }

# }

# server {

# listen 10.0.0.10:82；

# server\_name blog.etiantian.org; <- 此处也可以改成对应IP 10.0.0.10

# location / {

# root html/blog；

# index index.html index.htm；

# }

# }

# }

说明：通过修改不同server标签中的监听ip信息，实现基于ip的方式访问不同虚拟主机；精简化配置：只配置一个虚拟主机，其它注释

#### 扩展知识说明：vim编辑器使用

ctrl+v 进入可视化模块模式，实现批量编辑操作

按方向键，上下选择多行内容

shift+i 编辑选中的第一行

esc 退出编辑，完成多行内容统一设置

r 直接替换字符信息

x 删除当前光标所在位置字符信息

### 检查，重启过程

#### 检查配置文件语法信息是否正确

[root@web02 ~]# /application/nginx/sbin/nginx -t

#### 检查语法配置正确后，进行优雅重启nginx服务

[root@web02 ~]# /application/nginx/sbin/nginx -s reload

#### 监听地址信息发生变化

[root@web02 ~]# ss -lntup|grep nginx

#### 检查语法配置正确后，进行优雅关闭nginx服务

[root@web02 ~]# /application/nginx/sbin/nginx -s stop

#### 进行启动nginx服务

[root@web02 ~]# /application/nginx/sbin/nginx

#### 监听地址信息发生变化

[root@web02 ~]# ss -lntup|grep nginx

说明：编辑完多个虚拟主机信息，进行重启nginx服务（检查语法，平滑重启）；

和IP地址相关的配置变化，nginx服务需要重新启动生效，不能采用平滑重启方式

提示：更多nginx命令行参数说明：http://nginx.org/en/docs/switches.html

### 利用curl命令本地检测nginx配置是否成功

[root@web02 ~]# curl 10.0.0.8 成功

[root@web02 ~]# curl 172.16.1.8 失败，配置文件中为10.0.0.8

说明：显示结果均为相应站点目录资源信息；而访问非监听地址，没有返回信息

以上配置虚拟主机方式为---基于IP的方式配置虚拟主机

#### 扩展：403错误

01. 客户端访问被禁止了

02. 网站服务指定首页文件不存在了

## Nginx配置虚拟主机的步骤(简介)

### Nginx配置虚拟主机的步骤如下（适合各类虚拟主机类型）：

1) 增加一个完整的server标签段到结尾处。注意，要放在http的结束大括号前，也就是将server标签段放入http标签。

2) 更改server\_name及对应网页的root根目录，如果需要其他参数，可以增加或修改。

3) 创建server\_name域名对应网页的根目录，并且建立测试文件，如果没有index首页，访问会出现403错误。

如果是apache软件，没有首页文件，默认会把站点目录下面的信息显示出来

nginx出403错误解决方式：http://oldboy.blog.51cto.com/2561410/1633952

autoindex on; #<==当找不到首页文件时，会展示目录结构，这个功能一般不要用除非有需求。

PS：显示的目录结构中，有些信息点击就是下载，有的点击就是显示，因为扩展名称不一样

根本在于nginx软件是否能够进行解析

nginx是否解析：

1. html jpg 认识 显示出内容

2. 不认识 不解析 便直接下载

4) 检查Nginx配置文件语法，平滑重启Nginx服务，快速检查启动结果。

5) 在客户端对server\_name处配置的域名做host解析或DNS配置，并检查（ping域名看返回的IP是否正确）。

6) 在Win32浏览器中输入地址访问，或者在Linux客户端做hosts解析，用wget或curl接地址访问。

Nginx虚拟主机的官方帮助网址为：http://Nginx.org/en/docs/http/request\_processing.html

### 扩展知识：浏览器隐身模式设置

谷歌隐身模式：不会留下缓存

· 缓存信息抓包后显示的状态码为304

· nginx里面看缓存304的访问记录，访问nginx日志信息，访问尺寸为0

tail -f /application/nginx/logs/access\_www.log

## nginx排错过程说明

下面介绍客户端排查的思路：(如果curl www.baidu.com结果不是预期的)

### 第一步，在客户端上ping服务器端IP，命令如下。

ping 10.0.0.8 <- 排除物理线路问题影响

### 第二步，在客户端上telnet服务器端IP、端口，命令如下：

telnet 10.0.0.8 80 <- 排除防火墙等得影响

### 第三步，在服务端使用wget命令检测，如下：

wget 10.0.0.8（curl -I 10.0.0.8） <- 模拟用户访问，排除http服务自身问题，根据输出在排错

curl -v/wget -debug

提示信息：以上三步是客户端访问网站异常排查的重要三部曲。

nginx软件如何进行处理请求：参见官方网站资料：http://Nginx.org/en/docs/http/request\_processing.html

# nginx常用功能说明

## 多个主机不同配置文件进行管理(规范化配置文件)

### 创建一个存放目录

[root@web02 ~]# cd /application/nginx/conf/

[root@web02 conf]# mkdir extra

### 将不同站点server标签信息追加到www.conf bbs.conf blog.conf文件中去

[root@web01 extra]# ll

total 16

-rw-r--r-- 1 root root 211 Sep 12 18:38 bbs.conf

-rw-r--r-- 1 root root 187 Sep 12 17:06 blog.conf

-rw-r--r-- 1 root root 145 Sep 12 18:37 status.conf

-rw-r--r-- 1 root root 207 Sep 12 17:00 www.conf

[root@web01 extra]# cat www.conf

server {

listen 80;

server\_name www.etiantian.org;

location / {

root html/www;

index index.html index.htm;

autoindex on;

}

}

[root@web01 extra]# cat bbs.conf

server {

listen 80;

server\_name bbs.etiantian.org;

location / {

root html/bbs;

index index.html index.htm;

autoindex on;

}

}

[root@web01 extra]#

### 修改配置文件内容

[root@web01 conf]# cat nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

include extra/www.conf;

include extra/bbs.conf;

include extra/blog.conf;

include extra/status.conf;

}

### 短域名变为长域名---配置域名别名功能

server\_name www.etiantian.org www1.etiantian.org www2.etiantian.org；

主要功能：便于定位识别相应的nginx服务器集群节点

### 状态模块功能说明

#### nginx -V ---查看编译参数中，是否加载了状态模块信息(--with-http\_stub\_status\_module )

配置nginx服务的状态模块信息

cat >>/application/nginx/conf/extra/status.conf<<EOF

##status

server{

listen 80;

server\_name status.etiantian.org;

location / {

stub\_status on;

access\_log off;

}

}

EOF

sed -i '13 i include extra/status.conf;' nginx.conf

#### 扩展知识：虚拟机端口映射配置

可以对状态模块访问设置限制，只允许部分网段主机可以访问状态模块页面信息

location /nginx\_status {

stub\_status on；

access\_log off；

allow 10.0.0.0/24； #<==设置允许和禁止的IP段访问

deny all； #<==设置允许和禁止的IP段访问

}

## nginx日志功能说明

### 错误日志信息说明

error\_log的默认值为：

#default：error\_log logs/error.log error；

可以放置的标签段为：

#context：main，http，server，location

参考资料：http://nginx.org/en/docs/ngx\_core\_module.html#error\_log 。

说明：nginx官方文档查询信息如何使用，如何根据配置信息获取所在模块目录

### 访问日志信息说明

#### Nginx日志格式中默认的参数配置如下

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"'；

#### Nginx记录日志的默认参数配置如下：

access\_log logs/access.log main；

说明：以下信息类似于inotify命令的使用过程

a. 日志格式信息说明

b. 日志轮询切割说明

/bin/mv /application/nginx/logs/access.log /application/nginx/logs/access\_$(date +%F).log

/application/nginx/sbin/nginx -s reload

编写脚本，利用定时任务进行执行，实现日志切割的目的

扩展自学：logrotate工具使用(系统自带日志切割工具)

### 日志参数说明

#### 取出某条日志

10.0.0.1 - - [09/Sep/2017:02:32:42 +0800] "GET / HTTP/1.1" 304 0 "-"

"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"

#### 分析参数

'$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

参数说明

$remote\_addr 10.0.0.1 --- 访问端（客户端）IP地址信息

- -

$remote\_user - --- 显示远程访问者用户信息

[$time\_local] [09/Sep/2017:02:32:42 +0800] --- 显示访问时间信息

$request GET / HTTP/1.1 --- 请求行信息

$status 304 --- 状态码信息

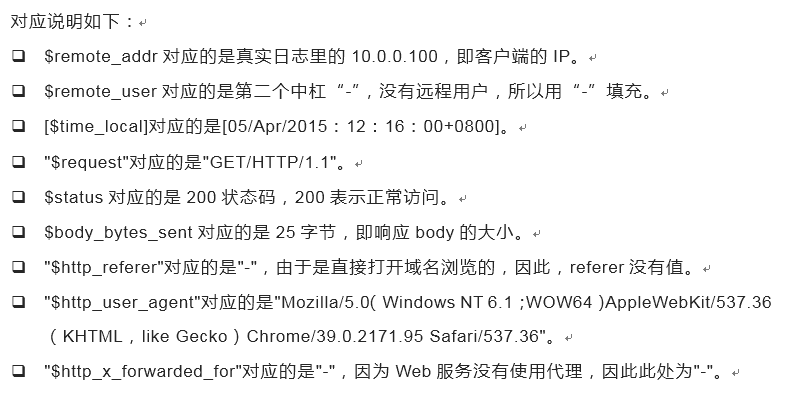
$body\_bytes\_sent 0 --- 响应报文主体内容大小

$http\_referer -

$http\_user\_agent Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36

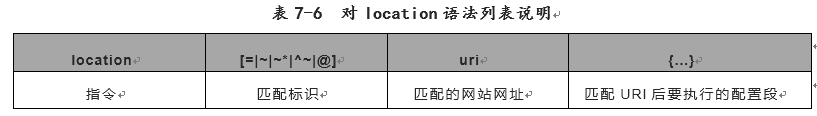
--- 客户端浏览网页工具信息

$http\_x\_forwarded\_for -



## nginx的location作用说明

### location表示位置的概念，类似于if，即满足什么条件，就做什么



官方location配置信息说明：

location = / {

[ configuration A ]

}

location / {

[ configuration B ]

}

location /documents/ {

[ configuration C ]

}

location ^~ /images/ {

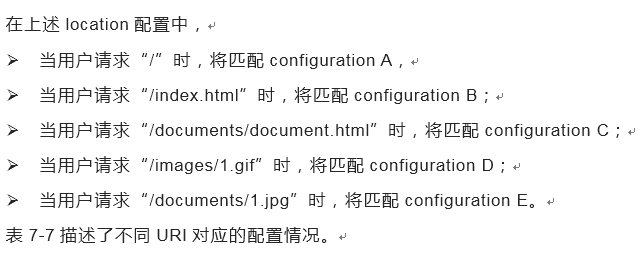
[ configuration D ]

}

location ~\* \.（gif|jpg|jpeg）$ {

[ configuration E ]

}



### 测试说明

server {

listen 80;

server\_name www.etiantian.org etiantian.org;

root html/www;

location / {

return 401;

}

location = / {

return 402;

}

location /documents/ {

return 403;

}

location ^~ /images/ {

return 404;

}

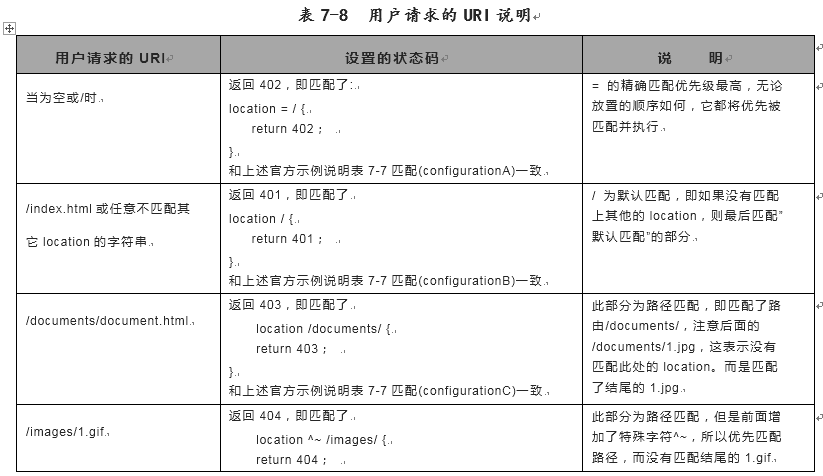
location ~\* \.（gif|jpg|jpeg）$ {

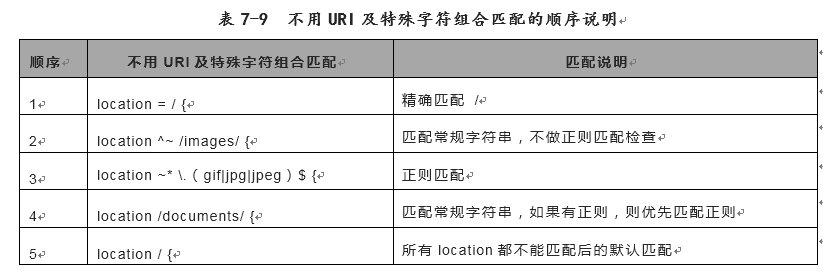
return 500;

}

access\_log logs/access\_www.log main

}

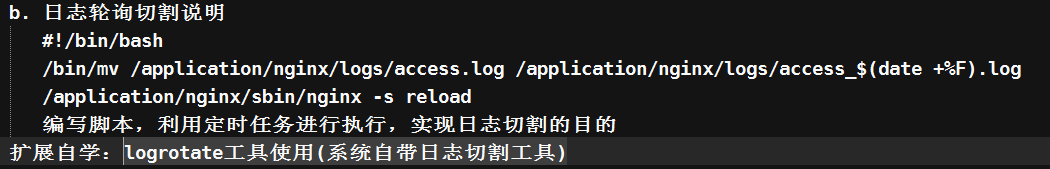




curl -s -o /dev/null -I -w "%{http\_code}\n" http://www.etiantian.org

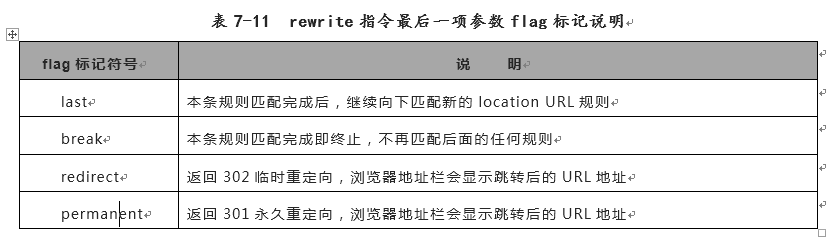
curl -s -o /dev/null -I -w "%{http\_code}\n" <http://www.etiantian.org/index.html>

### 日志切割---日志轮询



#### mv access.log access.log.$(date +%F) nginx -s reload

#### logroate



[root@web01 ~]# ll /var/log/messages\*

-rw------- 1 root root 207385 Mar 24 15:56 /var/log/messages

-rw-------. 1 root root 119421 Dec 27 19:32 /var/log/messages-20161227

-rw-------. 1 root root 654363 Mar 13 16:45 /var/log/messages-20170313

-rw------- 1 root root 204931 Mar 20 15:48 /var/log/messages-20170320

[root@web01 ~]# cat /etc/logrotate.d/

ConsoleKit dracut psacct syslog yum

[root@web01 ~]# cat /etc/logrotate.d/syslog

/var/log/cron

/var/log/maillog

/var/log/messages

/var/log/secure

/var/log/spooler

{

sharedscripts

postrotate

/bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true

endscript

}

## nginx的rewirte作用说明

### nginx 的rwite重写模块说明

http://www.oldboyedu.com/day1&oldboy

http://www.oldboyedu.com/day1-oldboy

rewrite语法格式：rewrite regex replacement [flag] == s#regex#replacement#g

rewrite应用标签：server、location、if

#### 练习一：

etiantian.org >> www.etiantian.org

server {

listen 80;

server\_name www.etiantian.org;

rewrite ^/(.\*) http://www.etiantian.org/$1 permanent;

location / {

root html/www;

index index.html index.htm;

}

access\_log logs/access\_www.log main;

}

说明：以上配置会存在问题

[root@web02 conf]# curl etiantian.org

<html>

<head><title>301 Moved Permanently</title></head>

<body bgcolor="white">

<center><h1>301 Moved Permanently</h1></center>

<hr><center>nginx/1.10.2</center>

</body>

</html>

[root@web02 conf]# curl -L etiantian.org

curl: (47) Maximum (50) redirects followed

#### 命令知识扩充：curl命令参数说明

curl -Lv

-L表示：-L/--location Follow Location: hints (H)；表示追踪访问的过程，跟踪Location信息；示意信息（H）

-v表示： 显示追踪的信息

#### 避免无线跳转的第一种方法：

if ($host ~\* "^etiantian.org$") {

rewrite ^/(.\*) http://www.etiantian.org/$1 permanent;

}

#### 避免无线跳转的第二种方法：

server { <== 添加了一个server标签，在www.etiantian.org标签之上

server\_name etiantian.org;

rewrite ^/(.\*) http://www.etiantian.org/$1 permanent;

}

#### 标记参数说明

301和302状态区别：

Http状态码301和302概念简单区别及企业应用案例

http://oldboy.blog.51cto.com/2561410/1774260

工作中应用场景说明

#### nginx 的rwite重写企业应用场景：

· 可以调整用户浏览的URL，使其看起来更规范，合乎开发及产品人员的需求。

· 为了让搜索引擎收录网站内容，并让用户体验更好，企业会将动态URL地址伪装成静态地址提供服务

· 网站换新域名后，让旧域名的访问跳转到新的域名上，例如：让京东的360buy换成了jd.com。

· 根据特殊变量、目录、客户端的信息进行URL跳转等。

说明：开源软件类似wordpress的，官方都会对伪静态配置进行说明

### nginx 的访问认证

#### 修改nginx的相关配置文件

vim extra/www.conf

location / {

root html/www;

index index.html index.htm;

auth\_basic "oldboy training";

auth\_basic\_user\_file /application/nginx/conf/htpasswd;

}

#### 创建密码认证文件并进行授权

yum install httpd-tools -y

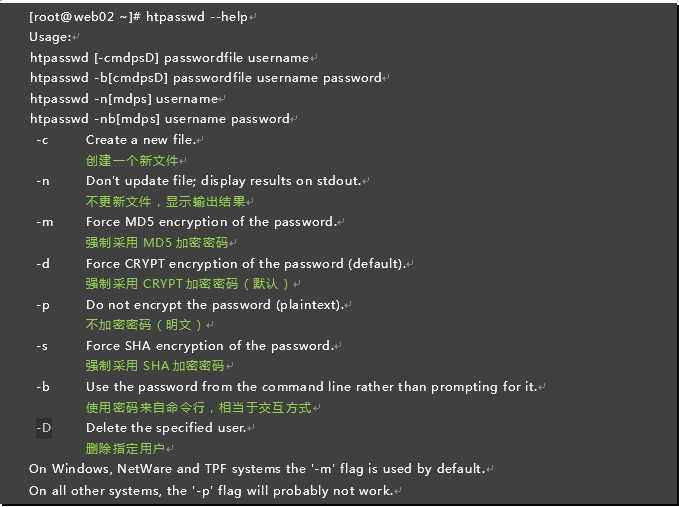
htpasswd -bc /application/nginx/conf/htpasswd oldboy 123456

-c Create a new file.

创建一个新的密码文件

-b Use the password from the command line rather than prompting for it

采用免交互的方式输入用户的密码信息



chmod 400 /application/nginx/conf/htpasswd

chown www /application/nginx/conf/htpasswd

ll /application/nginx/conf/htpasswd

-r-------- 1 nginx root 21 2017-05-14 19:45 /application/nginx/conf/htpasswd

#### 配置结果测试

windows浏览器进行测试

linux系统下进行测试

curl -u

-u/--user <user[:password]> Set server user and password

[root@web01 conf]# curl -u lidao888:123456 www.etiantian.org

web01 www

# 本章回顾

