# 1、基本命令

## start neginx

## ./nginx -c conf/nginx.conf

## 1、关闭Nginx

### 1、停止：

### nginx -s stop

### 2、有序退出

### nginx -s quit

## 2、当配置信息修改，需要重新载入这些配置时使用此命令。

### nginx -s reload

## 3、重启Nginx

### nginx -s reopen

## 4.检测配置文件

### nginx -t

## 5、平滑重启

**第二步：平滑重启**

### **kill -HUP nginx进程号**

**kill -HUP ''/var/run/nginx.pid"**

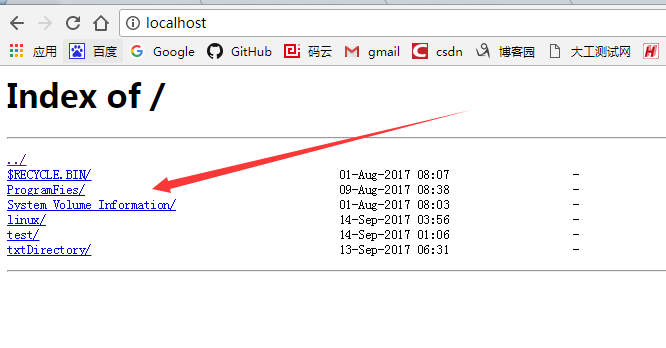
当nginx接收到HUP信号时，它会尝试先解析配置文件（如果指定文件，就使用指定的，否则使用默认的），如果成功，就应用新的配置文件（例如：重新打开日志文件或监听的套接字），之后，nginx运行新的工作进程并从容关闭旧的工作进程，通知工作进程关闭监听套接字，但是继续为当前连接的客户提供服务，所有客户端的服务完成后，旧的工作进程就关闭，如果新的配置文件应用失败，nginx继续使用之前的配置进行工作。

# 2、开启浏览功能

### 1、autoindex on; #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启

|  |
| --- |
| server {  listen 80;  server\_name localhost; #监听tomcat8080  root D:/;  index index.html index.htm;  #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启  autoindex on; |

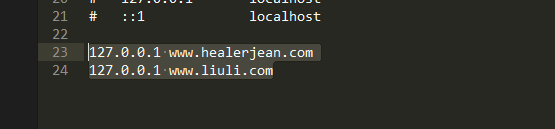
### 2、测试 成功



# 3、ngin配置站点[www.liuli.com](http://www.liuli.com)

## 1、首先需要将自定义的站点域名配置到host文件中，并且能够ping通

### 1、这里我输入了 [www.healerjean.com](http://www.healerjean.com) [www.liuli.com](http://www.liuli.com)

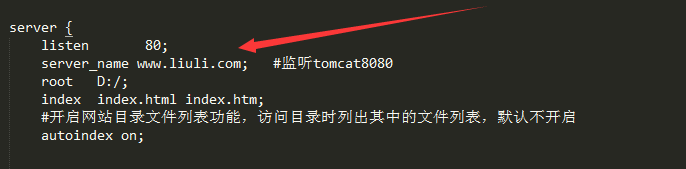


### 2、ping [www.liuli.com](http://www.liuli.com)

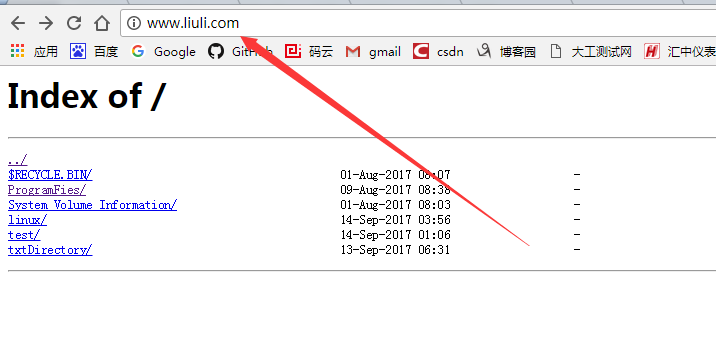


## 2、配置server参数 server\_name

|  |
| --- |
| server {  listen 80;  server\_name www.liuli.com;  root D:/;  index index.html index.htm;  #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启  autoindex on; |



## 3、测试 成功 [www.liuli.com](http://www.liuli.com)

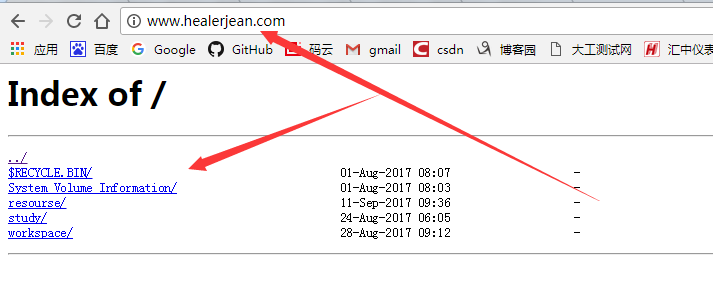


# 4、Nginx配置多个站点 [www.healerjean.com](http://www.healerjean.com)

## 1、host在2中已经添加好了，所以只要配置nginx.conf就可以了，添加一个server配置（这个站点用E盘测试）

|  |
| --- |
| server {  listen 80;  server\_name www.healerjean.com; #监听tomcat8080  root E:/;  index index.html index.htm;  #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启  autoindex on;  }  server {  listen 80;  server\_name www.liuli.com; #监听tomcat8080  root D:/;  index index.html index.htm;  #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启  autoindex on; |

## 2、测试成功 [www.healerjean.com](http://www.healerjean.com)



# 5、[location匹配规则](http://blog.csdn.net/w13770269691/article/details/12029675)

## 1、介绍location匹配字符

#### ~      #波浪线表示执行一个正则匹配，区分大小写

#### ~\*    #表示执行一个正则匹配，不区分大小写

#### ^~    #^~表示普通字符匹配，如果该选项匹配，只匹配该选项，不匹配别的选项，一般用来匹配目录

#### =      #进行普通字符精确匹配

#### @     #"@" 定义一个命名的 location，使用在内部定向时，例如 error\_page, try\_files

## 2、优先级别

#### 1、等号类型（=）的优先级最高。一旦匹配成功，则不再查找其他匹配项。

#### 2、^~类型表达式。一旦匹配成功，则不再查找其他匹配项。

#### 3、正则表达式类型（~ ~\*）的优先级次之。如果有多个location的正则能匹配的话，则使用正则表达式最长的那个。

#### 4、常规字符串匹配类型。按前缀匹配。

## 3、说明

#### 1，普通location和顺序无关、正则location按照顺序来匹配

#### 2、“=”或者"^~" 会终止正则搜索，但是不会终止普通/1231等location搜索。

#### 3、匹配优先级最高的当然是精确匹配。

#### 4、只要匹配到了正则location（不管是什么匹配，也就是说不需要最精确的匹配结果），就会终止继续往下搜索。

#### 5、只有带有 "~" 或者"~\*"前缀的属于**正则location**。其余的都是普通location。

#### 6、@ 这个用的不多，想了解的可以自行百度，有需求的可以在下方提问也行。

## 4、location优先级示例



# 6、自定义日志，以及位置

## 1、首先需要打开日志支持

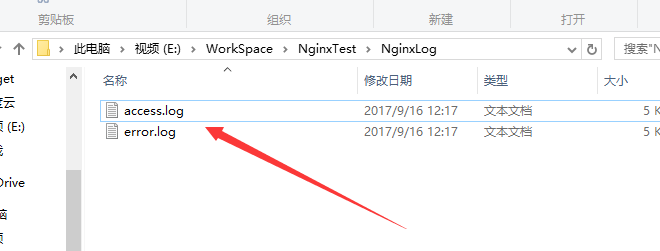
|  |
| --- |
| log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"'; |

## 2、开始自定义，首先需要先创建好文件夹和日志的文件

### 1、mian级别和warn级别

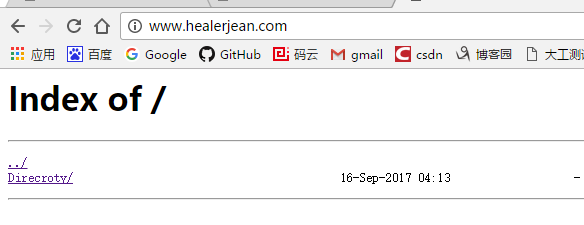
|  |
| --- |
| server {  listen 80;  server\_name www.healerjean.com; #监听tomcat8080  root E:/;  index index.html index.htm;  autoindex on; #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启    access\_log E:/WorkSpace/NginxTest/NginxLog/access.log main;  error\_log E:/WorkSpace/NginxTest/NginxLog/error.log warn;  }} |

### 2、建立目录下的文件access.log 和error.log

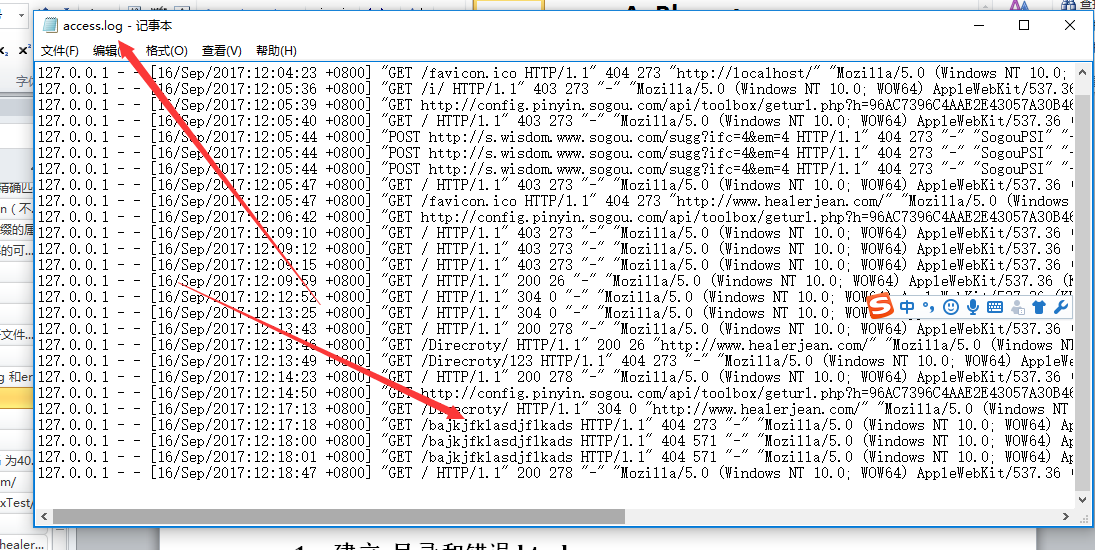


### 3、测试

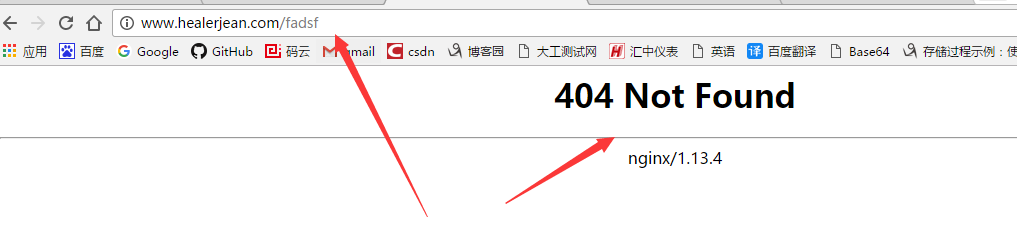
#### 1、正确的路径



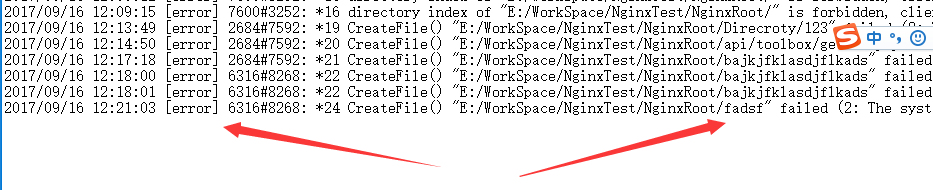
#### 2、正确的日志



#### 1、错误的路径

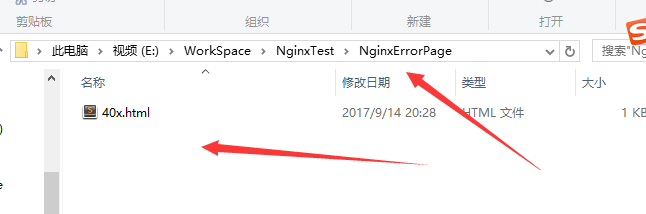


#### 2、错误的日志



# 7、自定义错误页面，以及位置（位置可以随意，不再root目录下）

## 1、建立 目录和错误html

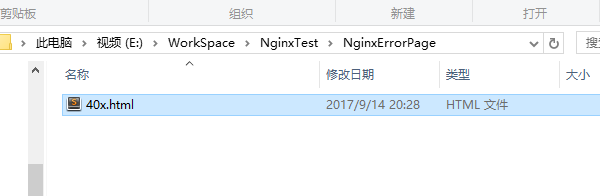


## 2、开始配置，首先需要明确错误码 为403或者404的时候会选择40x.html

|  |
| --- |
| server {  listen 80;  server\_name www.healerjean.com; #监听tomcat8080  root E:/WorkSpace/NginxTest/NginxRoot;  index index.html index.htm;  #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启  autoindex on;  #自定义日志  access\_log E:/WorkSpace/NginxTest/NginxLog/access.log main;  error\_log E:/WorkSpace/NginxTest/NginxLog/error.log warn;  #自定义错误页面  error\_page 403 404 /40x.html; #/40x.html 必须加上  location = /40x.html {  root E:/WorkSpace/NginxTest/NginxErrorPage;  }  } |

## 3、测试<http://www.healerjean.com/>

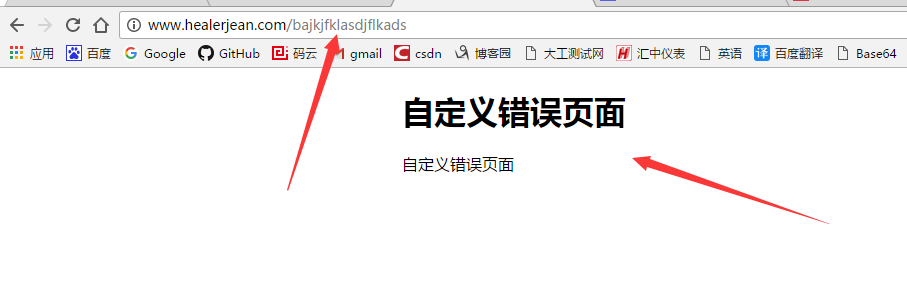
### 1、在目录E:/WorkSpace/NginxTest/NginxErrorPage;创建40.html



### 2、直接输入，会发现有显示

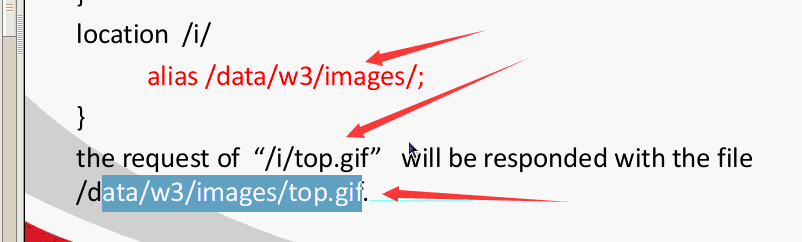


### 3、输入错误的路径<http://www.healerjean.com/bajkjfklasdjflkads>



# 8、自动索引以及别名功能 alias

## 1、访问 /i/top.gif 相当于是 /data/ws/images/top.gif

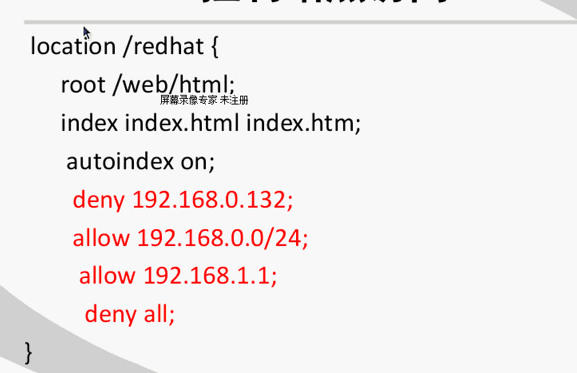


|  |
| --- |
| # 别名功能 alias  location = /i/ {  alias E:/WorkSpace/NginxTest/NginxLog  } |

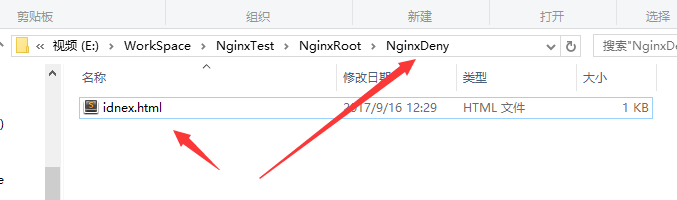
# 9、控制站点访问 allow deny

### 编写location

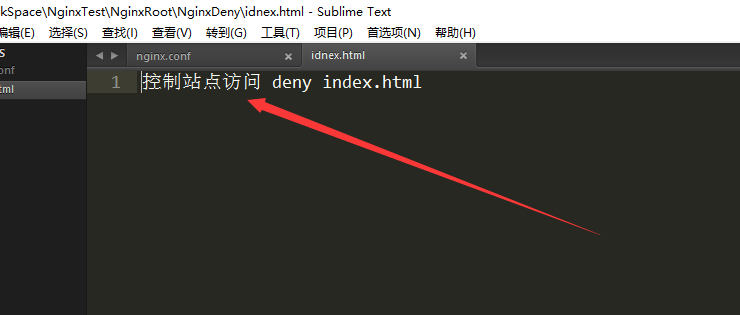
|  |
| --- |
| server {  listen 80;  server\_name www.healerjean.com; #监听tomcat8080  root E:/WorkSpace/NginxTest/NginxRoot;  index index.html index.htm;  #开启网站目录文件列表功能，访问目录时列出其中的文件列表，默认不开启  autoindex on;  #自定义日志  access\_log E:/WorkSpace/NginxTest/NginxLog/access.log main;  error\_log E:/WorkSpace/NginxTest/NginxLog/error.log warn;  #自定义错误页面  error\_page 403 404 /40x.html;  location = /40x.html {  root E:/WorkSpace/NginxTest/NginxErrorPage;  }  # 别名功能 alias  location = /i/ {  alias E:/WorkSpace/NginxTest/NginxRoot;  }  # 控制站点访问  location /NginxDeny {  deny 127.0.0.1;  }  } |



### 1、在root目录下创建文件夹NginxDeny



### 2、编写限制ip的index.html



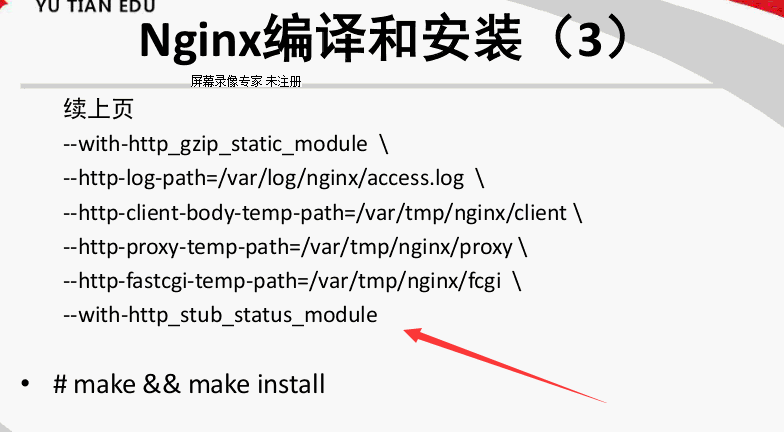
### 3、测试，访问 <http://www.healerjean.com/NginxDeny/> 报错，进入40.html





# 10、Nginx 状态检查

## 准备条件 安装的时候添加过状态检查模块

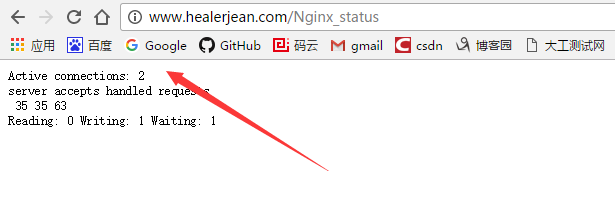


## 1、stub\_status on

|  |
| --- |
| # 状态检查  location /Nginx\_status {  stub\_status on;  access\_log off;  } |

## 2、测试,使用火狐和谷歌浏览器分别访问 [www.healerjean.com](http://www.healerjean.com) (显示连接数为2)

#### <http://www.healerjean.com/Nginx_status>



# 11、文件结果

