目录

[旧 2](#_Toc533790529)

[安装 2](#_Toc533790530)

[本地域名解析配置 3](#_Toc533790531)

[基本配置 3](#_Toc533790532)

[使用 4](#_Toc533790533)

[查看配置文件位置 5](#_Toc533790534)

[启动 5](#_Toc533790535)

[重启 5](#_Toc533790536)

[关闭 5](#_Toc533790537)

[基础 6](#_Toc533790538)

[流程 6](#_Toc533790539)

[模块 7](#_Toc533790540)

[进程 8](#_Toc533790541)

[变量 8](#_Toc533790542)

[$http\_ 8](#_Toc533790543)

[常用官方变量 8](#_Toc533790544)

[配置 9](#_Toc533790545)

[规则 9](#_Toc533790546)

[常用配置项 10](#_Toc533790547)

[调试进程和定位 10](#_Toc533790548)

[正常运行配置 11](#_Toc533790549)

[优化 12](#_Toc533790550)

[事件配置类 12](#_Toc533790551)

[模块-http 12](#_Toc533790552)

[基本配置 13](#_Toc533790553)

[server 13](#_Toc533790554)

[文件路径 14](#_Toc533790555)

[index 15](#_Toc533790556)

[error\_page 16](#_Toc533790557)

[网络连接相关 16](#_Toc533790558)

[客户端限制 17](#_Toc533790559)

[缓存 18](#_Toc533790560)

[变量 18](#_Toc533790561)

[负载均衡 18](#_Toc533790562)

[反向代理相关 19](#_Toc533790563)

[指令 21](#_Toc533790564)

[其他 21](#_Toc533790565)

[正则 21](#_Toc533790566)

[rewrite 21](#_Toc533790567)

[常见错误 22](#_Toc533790568)

[13: Permission denied 22](#_Toc533790569)

# 旧

## 安装

前置软件

gcc编译器，gcc-c++编译器，(现在的服务器一般都有了)

yum install -y gcc

yum install -y gcc-c++

pcre: perl兼容正则表达式工具。

yum install -y pcre pcre-devel

zlib

yum install -y zlib zlib-devel

openssl开发库(需手动安装)

yum install -y openssl openssl-devel

epel(里面有大量第三方包，nginx不在yum默认的包里)

sudo yum install epel-release

安装

sudo yum install nginx

启动

sudo systemctl start nginx

设置系统启动时启动

sudo systemctl enable nginx

## 本地域名解析配置

利用nginx进行反向代理的时候，我们会配置proxy\_pass。在启动nginx的时候，会报

nginx: [emerg] host not found in upstream "yq.object.com" in /usr/local/nginx/conf/vhost/yq.nginx.com.conf:19 这个错误。

其实nginx配置语法上没有错误的,只是系统无法解析这个域名,所以报错.  
解决办法就是添加dns到/etc/resolv.conf 或者是/etc/hosts,让其能够解析到IP。具体步骤如下：

vim /etc/hosts

修改hosts文件，在hosts文件里面加上一句

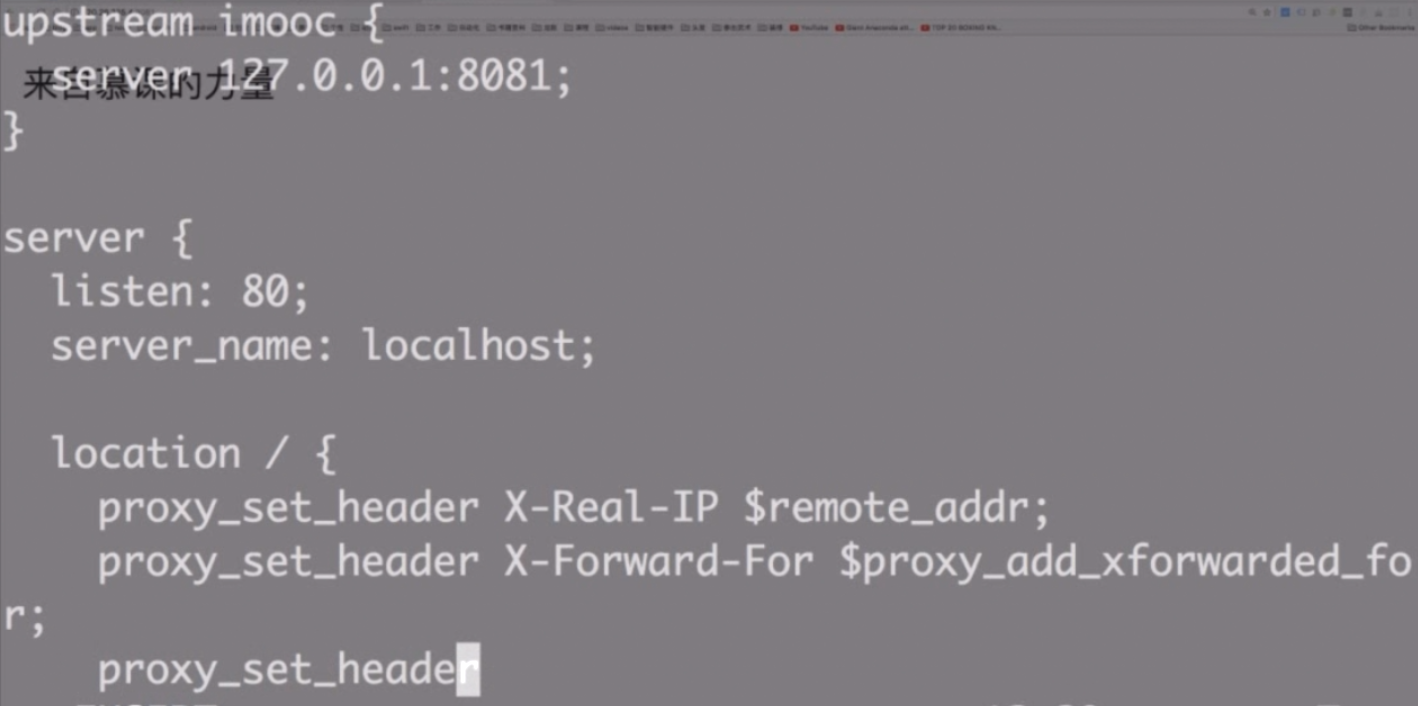
127.0.0.1       localhost.localdomain   yq.object.com

## 基本配置

linux: cd ect/nginx/conf.d

mac: usr/loacl/etc/nginx

sudo nginx –t 测试





nginx.conf里修改加载conf.d里面所有文件

http {

include servers/\* # 目录

include conf.d # 文件

}

关键: proxy\_pass 比如proxy\_pass <http://127.0.0.1:3000/> ，实际就是代理到这个端口去处理。（对应的程序运行在这个端口。）

## 使用

mac下记得sudo

-v, -V查看版本。-V更详细，包括编译器版本和配置参数。

### 查看配置文件位置

sudo nginx -t

-t: 表示不运行。

### 启动

sudo nginx

使用默认配置。

nginx -c /path/to/nginx.conf

-p:指定安装目录（区分nginx执行脚本的位置，也就是说执行脚本可以不在安装目录里）。 默认：usr/local/ect/nginx/

-g: 临时指定全局配置项（不能与.conf的默认配置项冲突，否则无法启动，使用了以后调其他命令比如关闭时也需要带上，保持配置一致）

### 重启

nginx -s reload

### 关闭

nginx -s stop :快速停止nginx

quit ：完整有序的停止nginx

其他的停止nginx 方式：

ps -ef | grep nginx： 查看nginx进程号。

kill -QUIT 进程号 ：从容停止Nginx

kill -TERM 进程号 ：快速停止Nginx

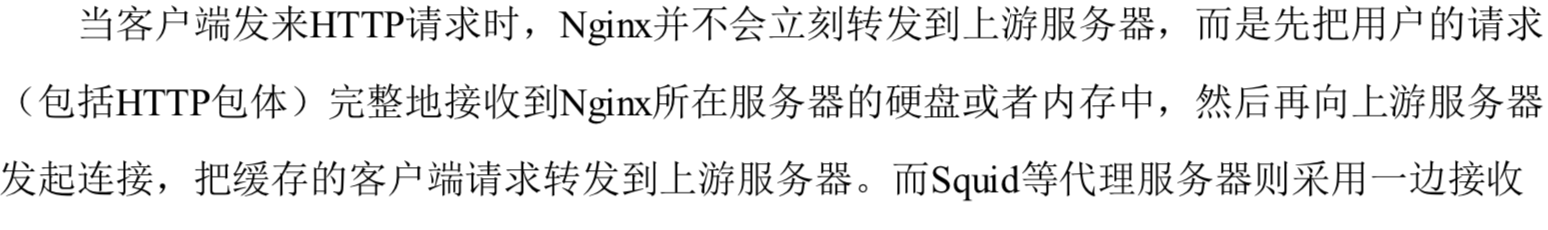
kill -9 nginx进程号 ：强制停止Nginx

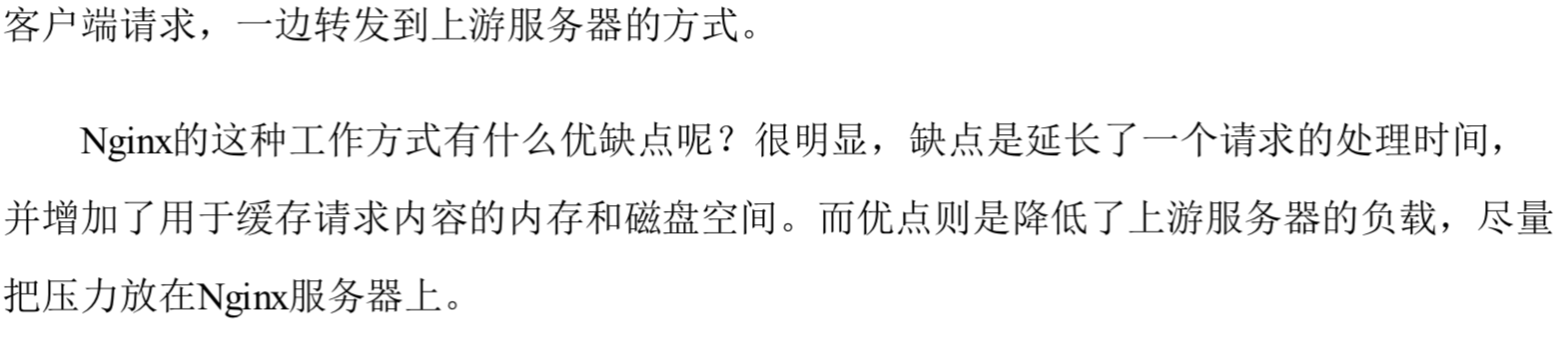
也可以不用手动查看进程号

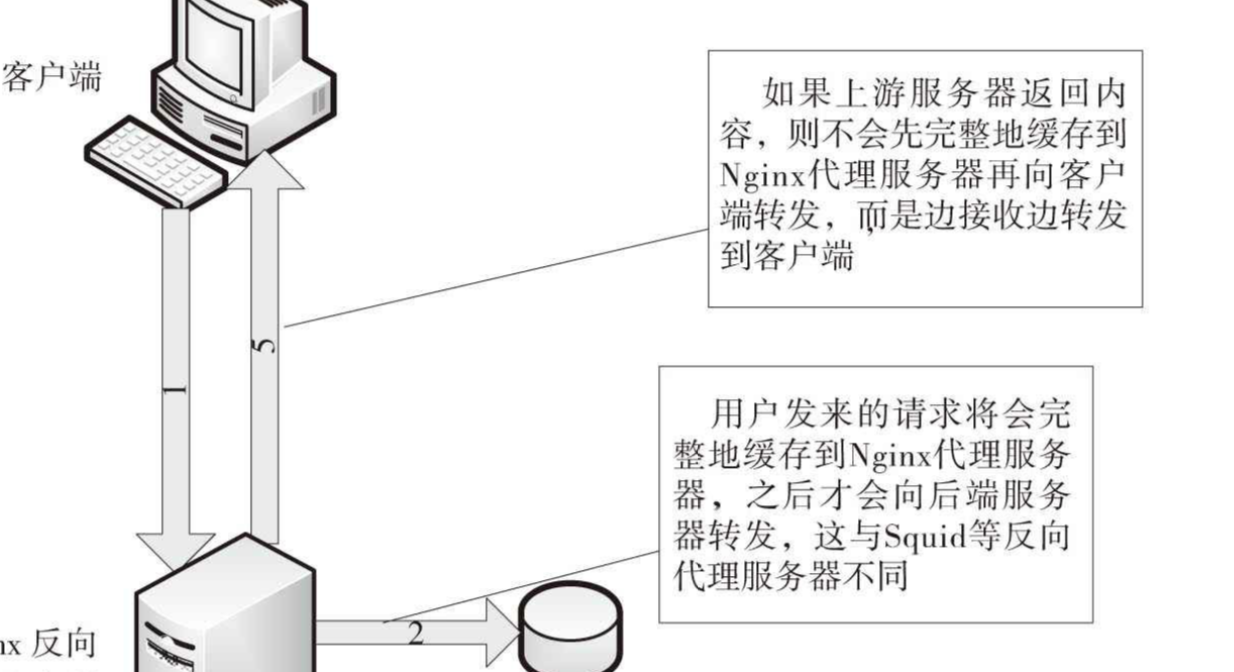
kill -QUIT `cat /usr/local/nginx/nginx.pid`（保存进程号的文件）

# 基础

linux2.6及以上才支持epoll。select或者

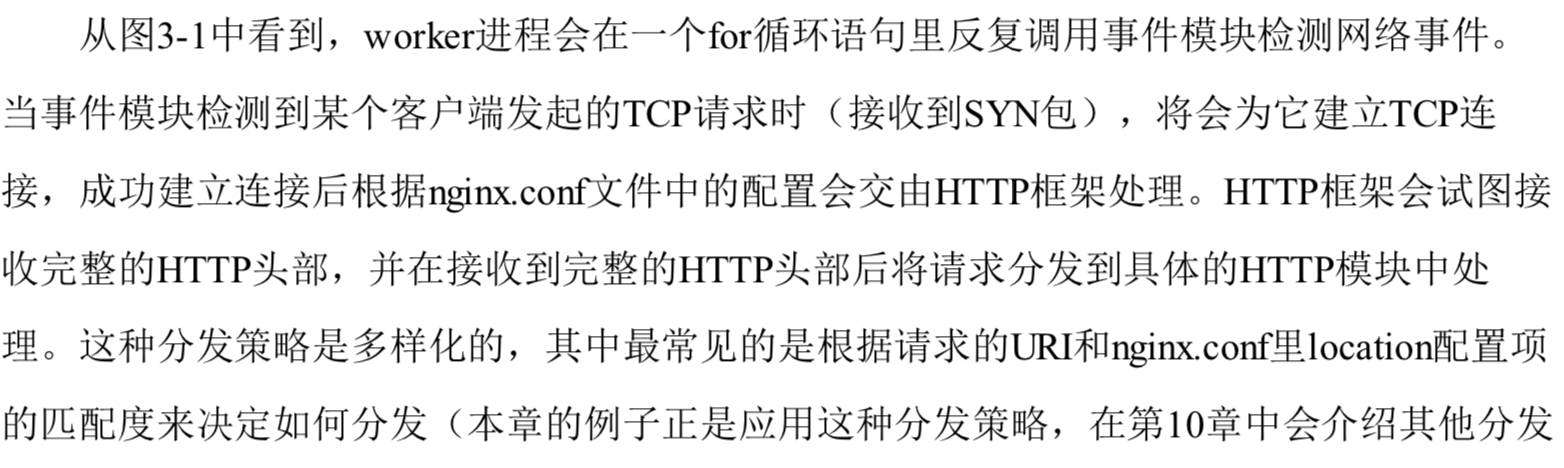


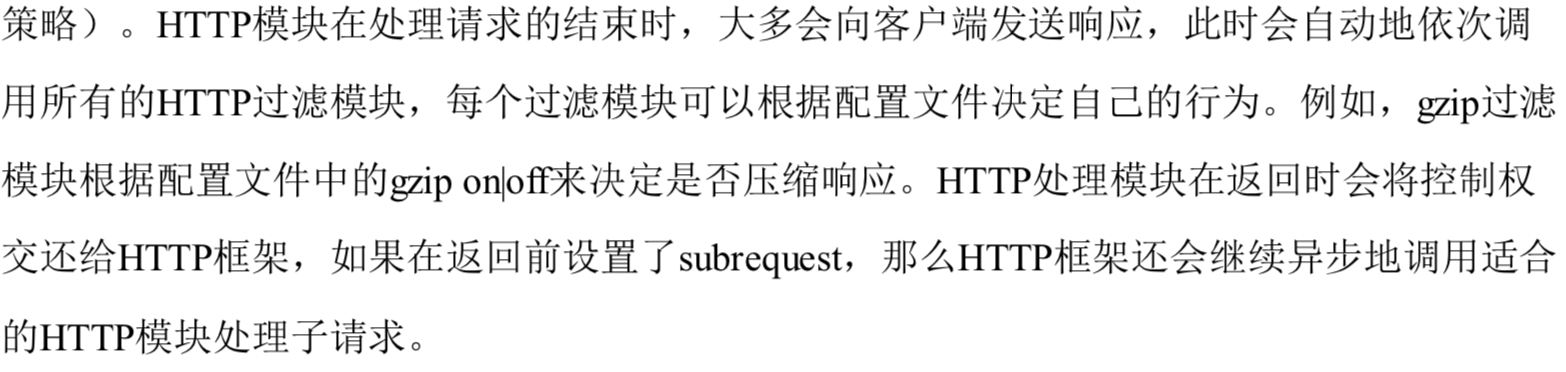




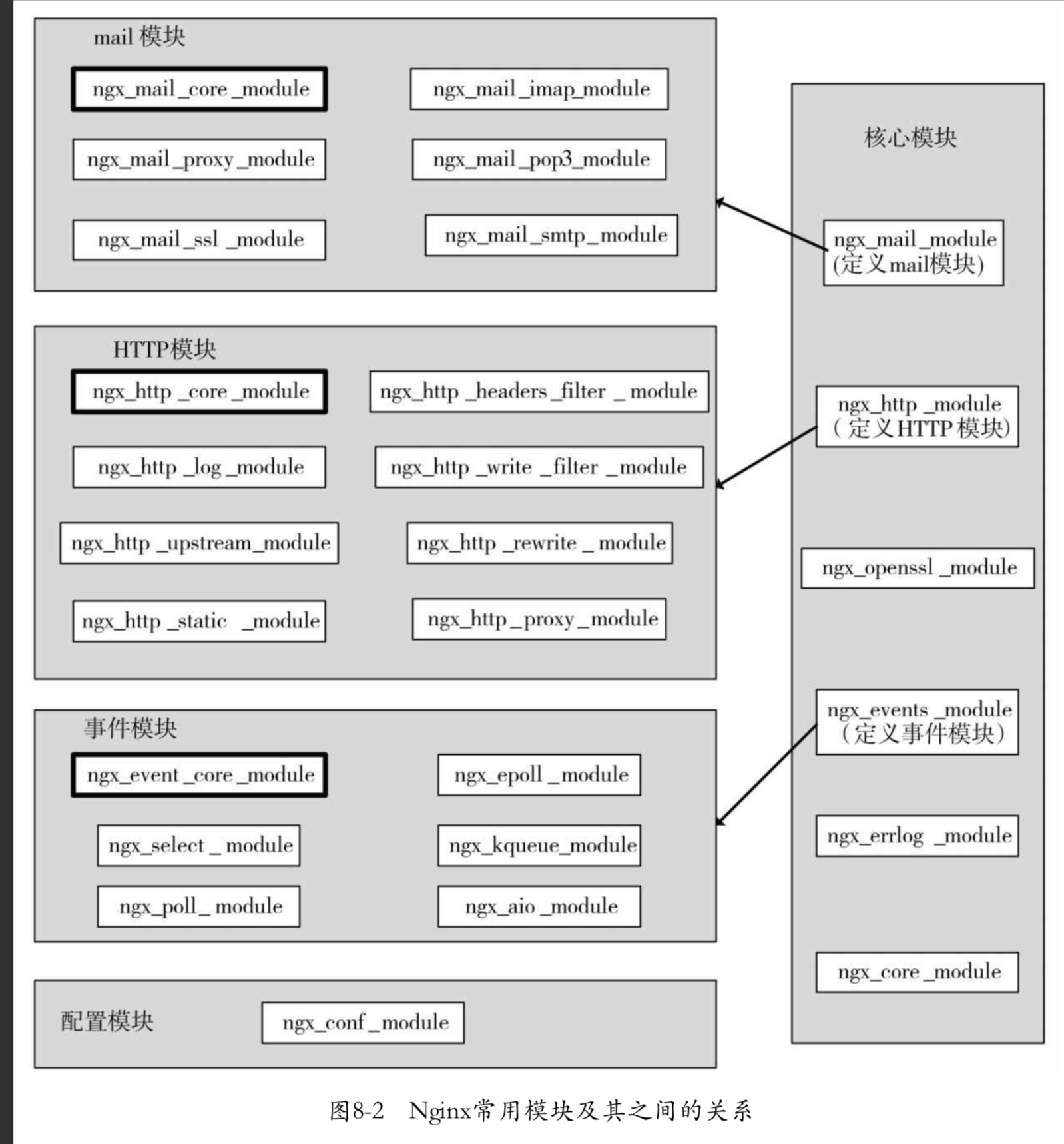
nginx和服务器之间走内网，速度快，缩短服务器保持连接的时间。

## 流程

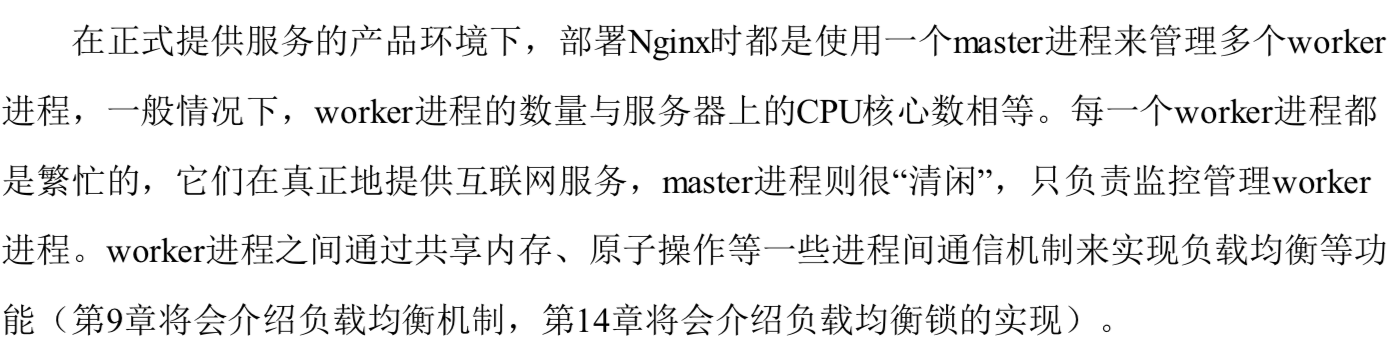




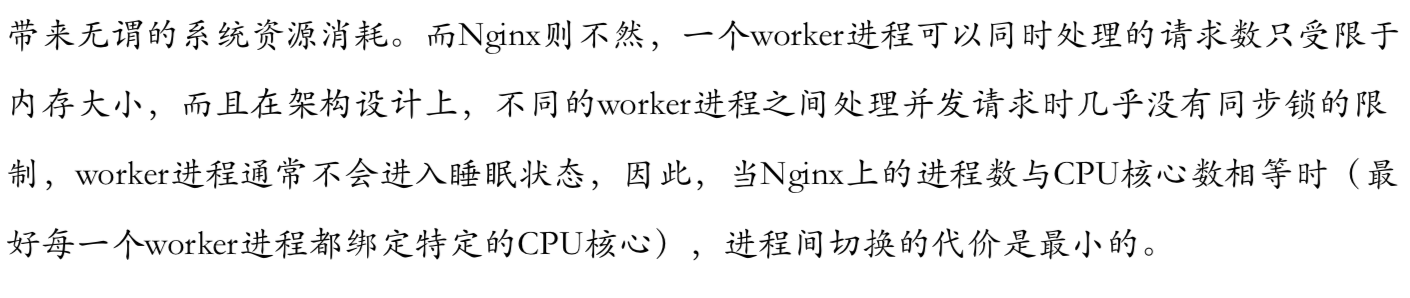
## 模块



## 进程



apache一个线程同时只能处理一个请求，所以在高并发的情况下，需要开大量的进程，进程切换导致资源消耗。



每个worker进程都是单线程进程

## 变量

### $http\_

所有请求头里面的key再nginx里面都可以通过小写和下划线来让nginx读取。例如header里面的Host就能转成$http\_host, user\_agent可以转成$http\_user\_agent。

### 常用官方变量

$host

host变量的值按照如下优先级获得：

请求行中的host.

请求头中的Host头部.

与一条请求匹配的server name.

$reomote\_addr

假设中间没有任何代理，那么网站的web服务器（Nginx，Apache等）就会把remote\_addr设为你的机器IP，如果你用了某个代理，那么你的浏览器会先访问这个代理，然后再由这个代理转发到网站，这样web服务器就会把remote\_addr设为这台代理机器的IP。

# 配置

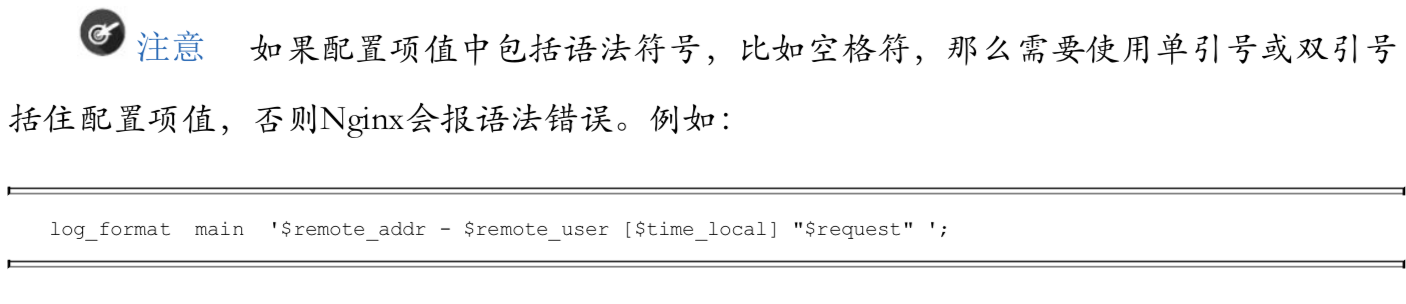
## 规则

以配置块为单位，可以嵌套，内层配置块直接继承外层配置块

配置块名 {

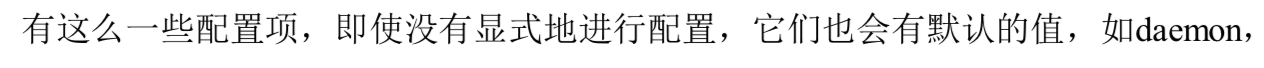
配置项名 配置项值;

}



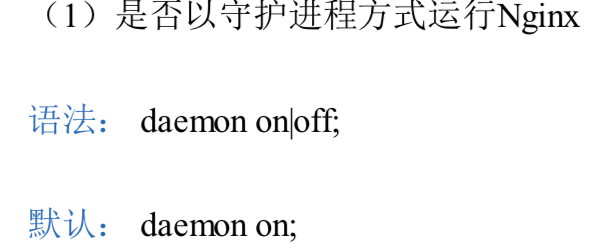
# 注释

变量只有某些模块支持，并不是通用的。（模块自己提供的，遵循一些命名规则）



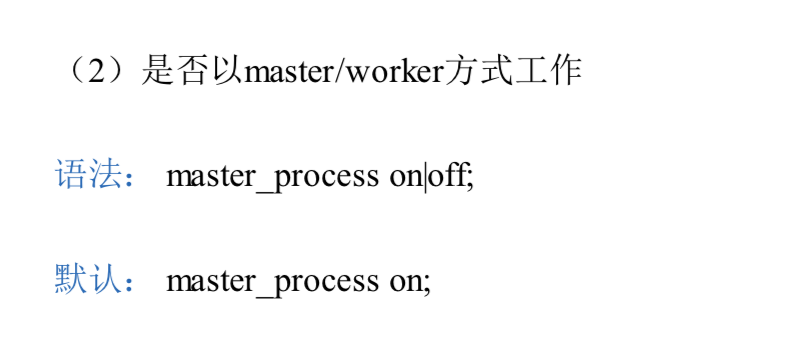
## 常用配置项

### 调试进程和定位

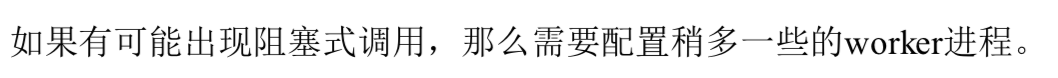


#### 工作方式

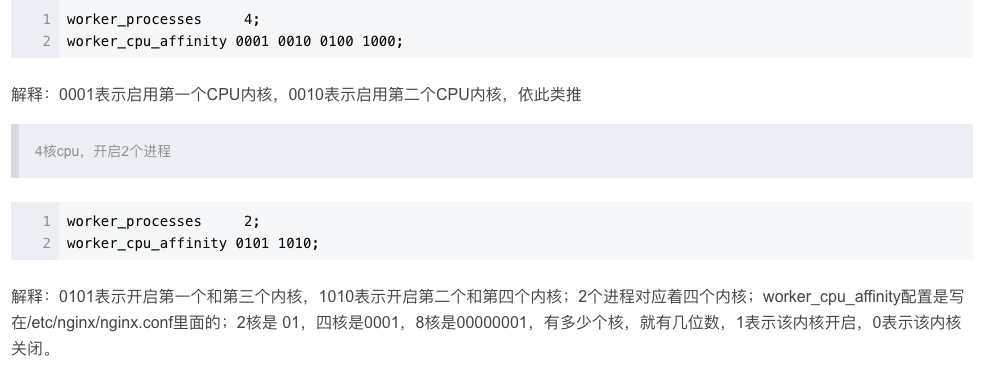
https://www.cnblogs.com/felixzh/p/6283816.html



worker\_processes auto; work\_process数量



worker\_cpu\_affinity auto; 启用哪些cpu内核，默认是没有开启利用多核cpu的配置的。

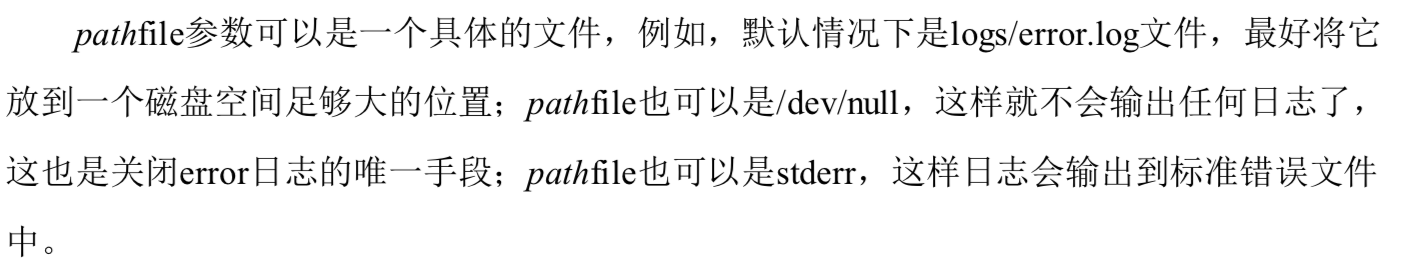


worker\_rlimit\_nofile 20480; 可用文件描述符(文件句柄)数量

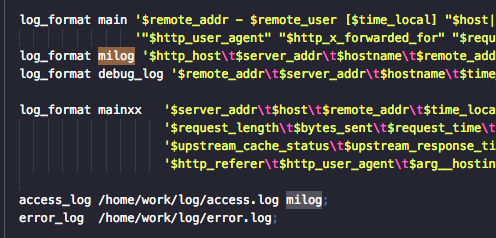
worker\_connections：这个属性是指单个工作进程可以允许同时建立外部连接的数量。无论这个连接是外部主动建立的，还是内部建立的。这里需要注意的是，一个工作进程建立一个连接后，进程将打开一个文件副本。所以这个数量还受操作系统设定的，进程最大可打开的文件数有关。

#### log

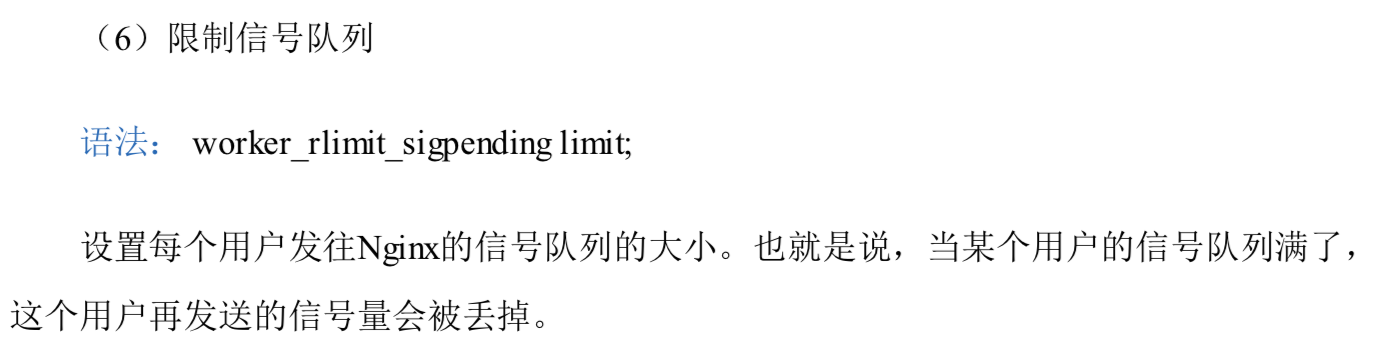
##### error\_log



##### access\_log

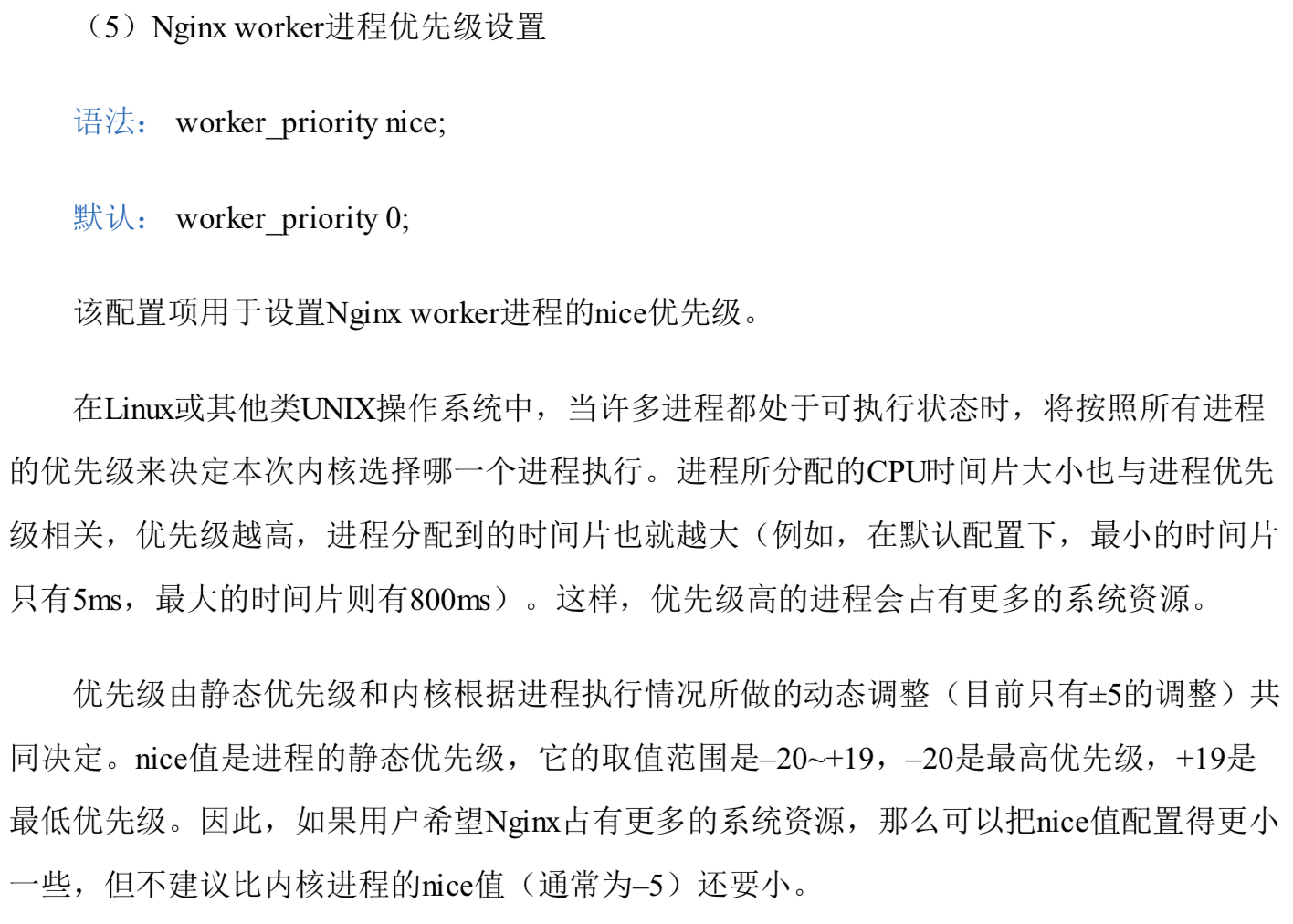


### 正常运行配置



### 优化

#### worker\_priority



### 事件配置类

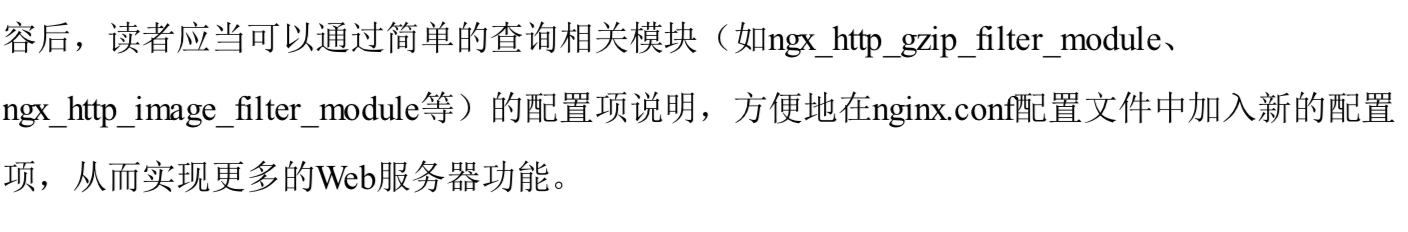
#### accept\_mutext

# 模块-http

ngx\_http\_core\_module

ngx\_http\_index\_module

ngx\_http\_proxy\_module



## 基本配置

一般没特别说明的属性可用于server,location,http模块。

### server

每个server块对应一个虚拟主机。（即可以匹配一个主机域名）

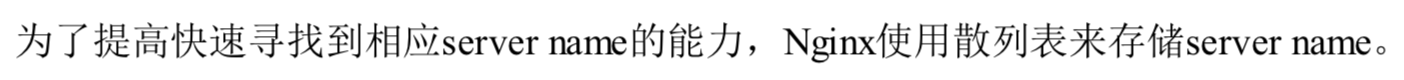
listen

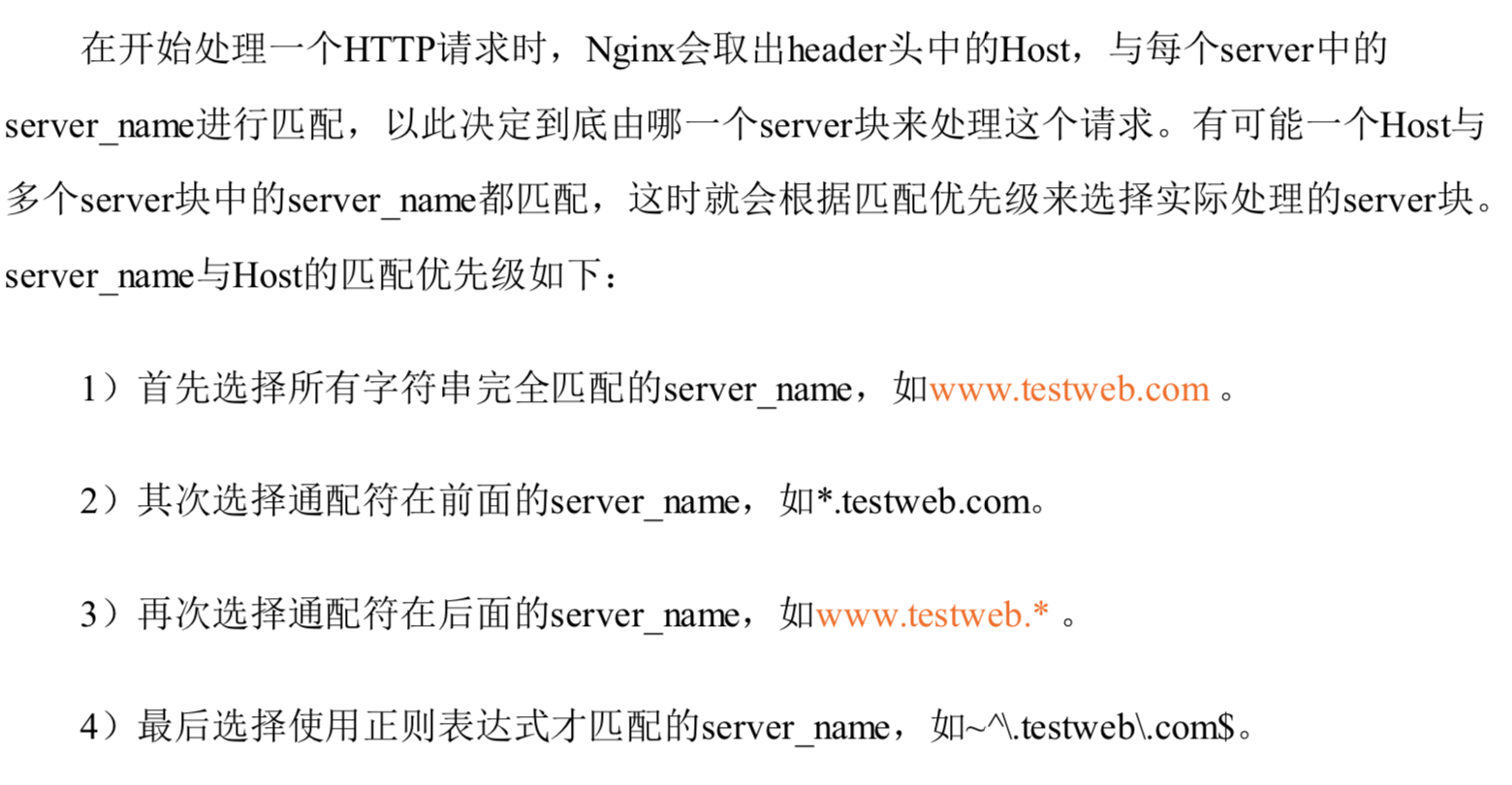


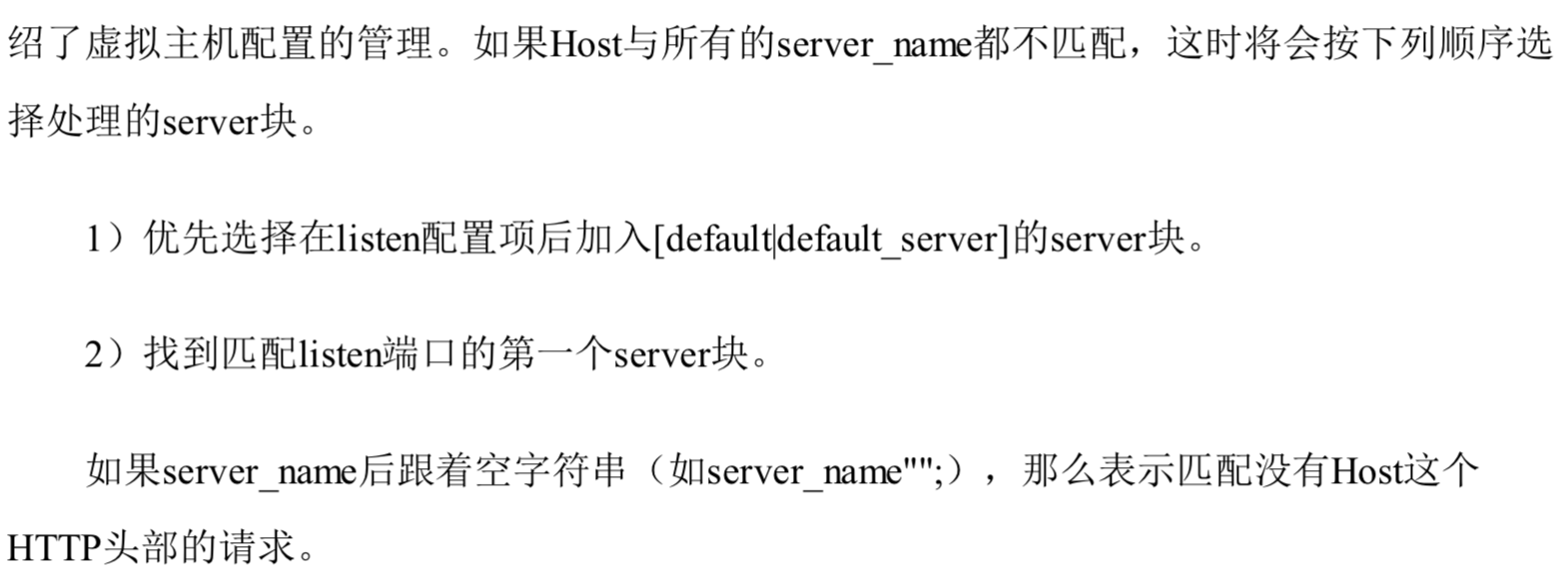
server\_name

server\_name name [... ]

默认hostname







server\_name\_in\_redirect on|off

locaciton

location [=|~|~\*|^~|@] /uri/ { ... }

=: 完全匹配。默认： 正则式匹配

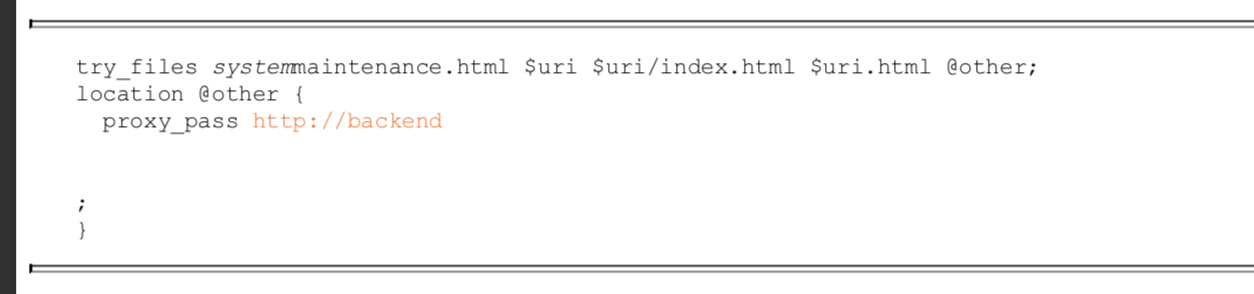
~ 大小写敏感

~\* 忽略大小写

^~ 匹配具体路径的前半部分： /img => /imgone

@ 仅处理内部重定向过来的请求，不处理外面的请求

有多个匹配时，被第一个location处理。



### location

常用header:

Host:

在http1.1协议里，是一个必须设置的头部。

X-Real-IP ：

$remote\_addr

自定义头部字段，通常被 HTTP 代理用来表示与它产生 TCP 连接的设备 IP。所以一般设置成上一级的ip。

X-Forwarded-For：

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for

X-Forwarded-For 是一个 HTTP 扩展头部。HTTP/1.1（RFC 2616）协议并没有对它的定义，它最开始是由 Squid 这个缓存代理软件引入，用来表示 HTTP 请求端真实 IP。如今它已经成为事实上的标准，被各大 HTTP 代理、负载均衡等转发服务广泛使用，并被写入 RFC 7239（Forwarded HTTP Extension）标准之中。

$proxy\_add\_x\_forwarded\_for：假如接收到的报文已经含有X-FORWARDED-FOR首部，那么nginx将在后面添加上$remote\_addr（也就是发送者的ip地址在最后。

所以通过这样设置，每一级代理的ip都能被记录下来，方便后端服务操作。

### 文件路径

root



这样会导致单页面的非/路径找不到资源。

location / {

root /home/work/app/;

index index.html;

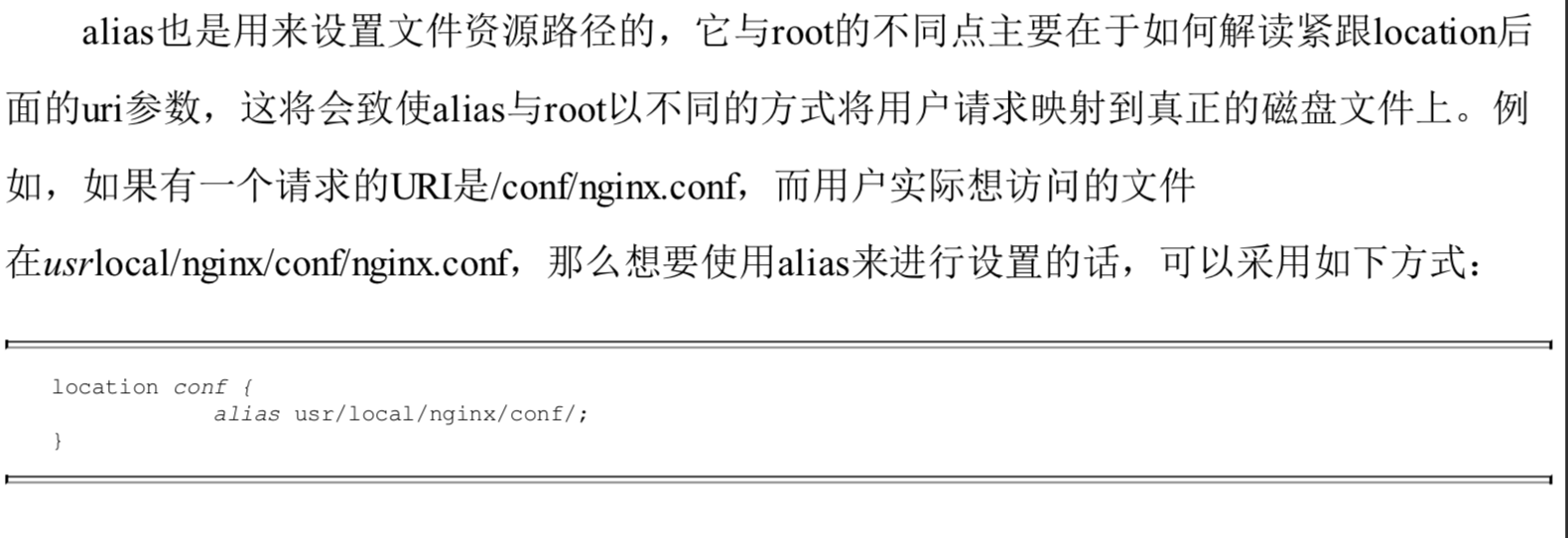
if (!-e $request\_filename) {

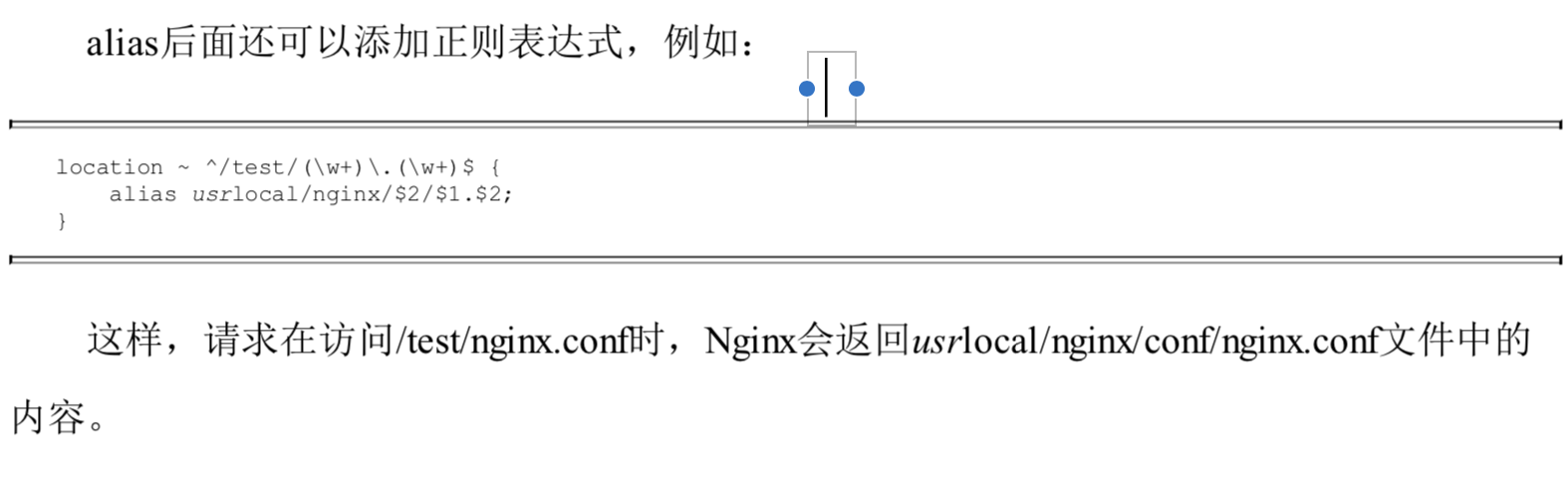
rewrite "^.\*" /index.html break;

}

}

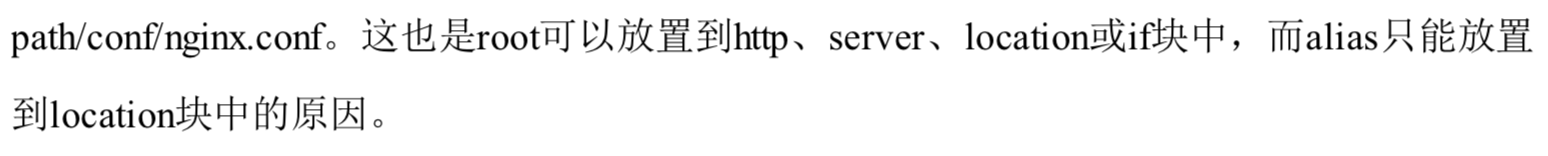
alias





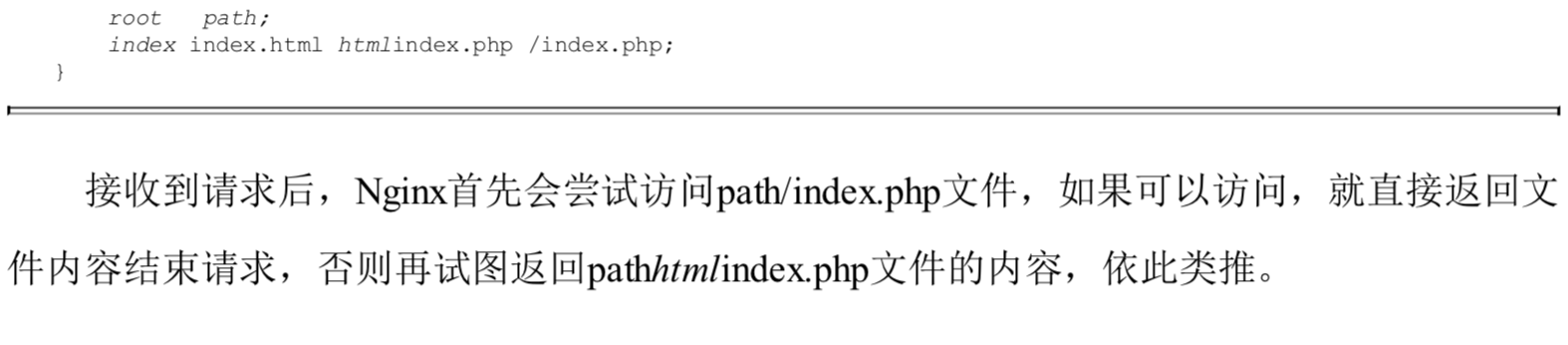
区别

root的路径相当于location的/，alias的路径相当于location的对应路径。



### index





注意这里的location没有跟具体路径，在新的nginx里会报错。

### error\_page

可用于if

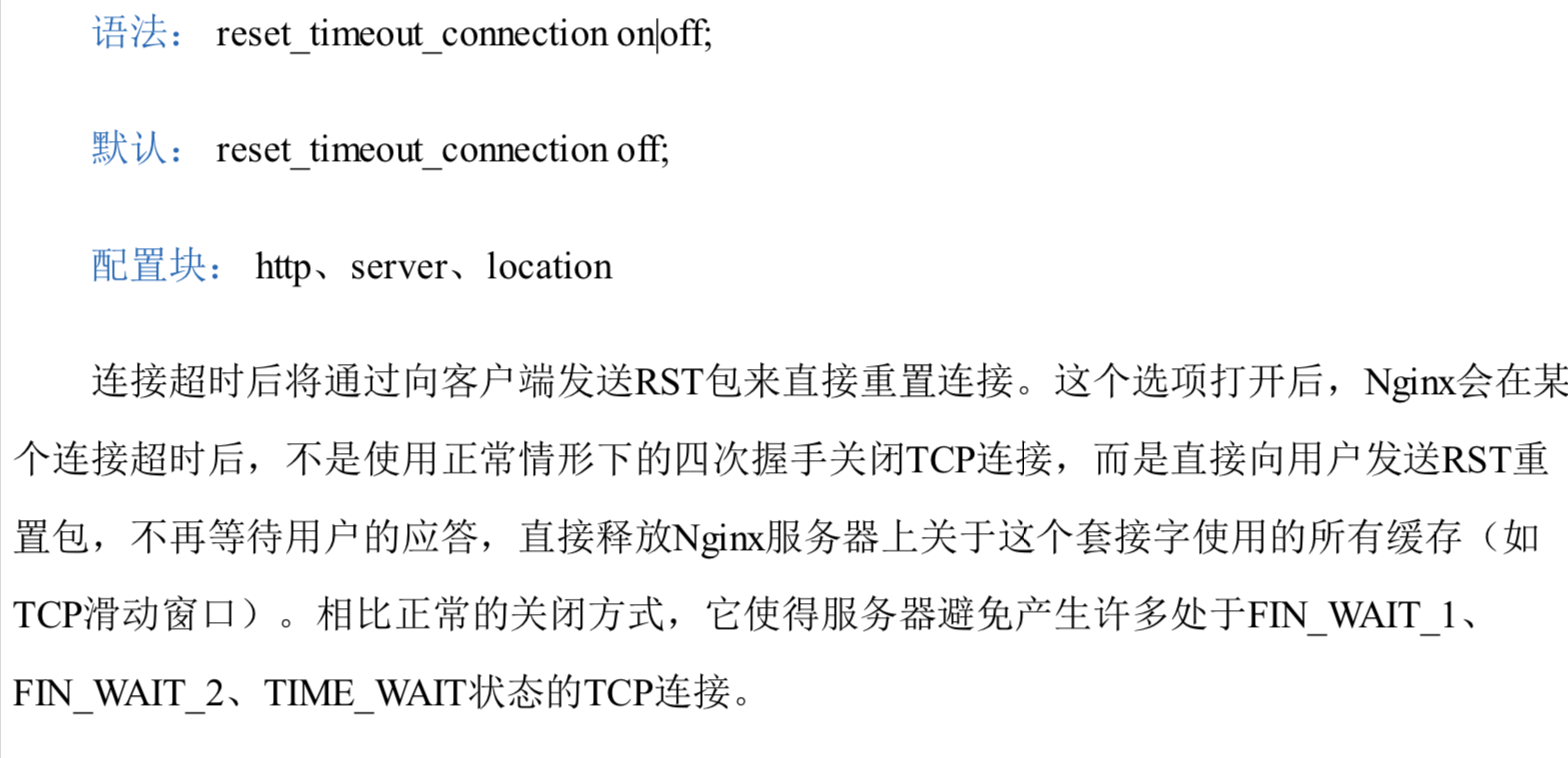
error\_page code [ code... ] [ = | =answer-code ] uri | @named\_location

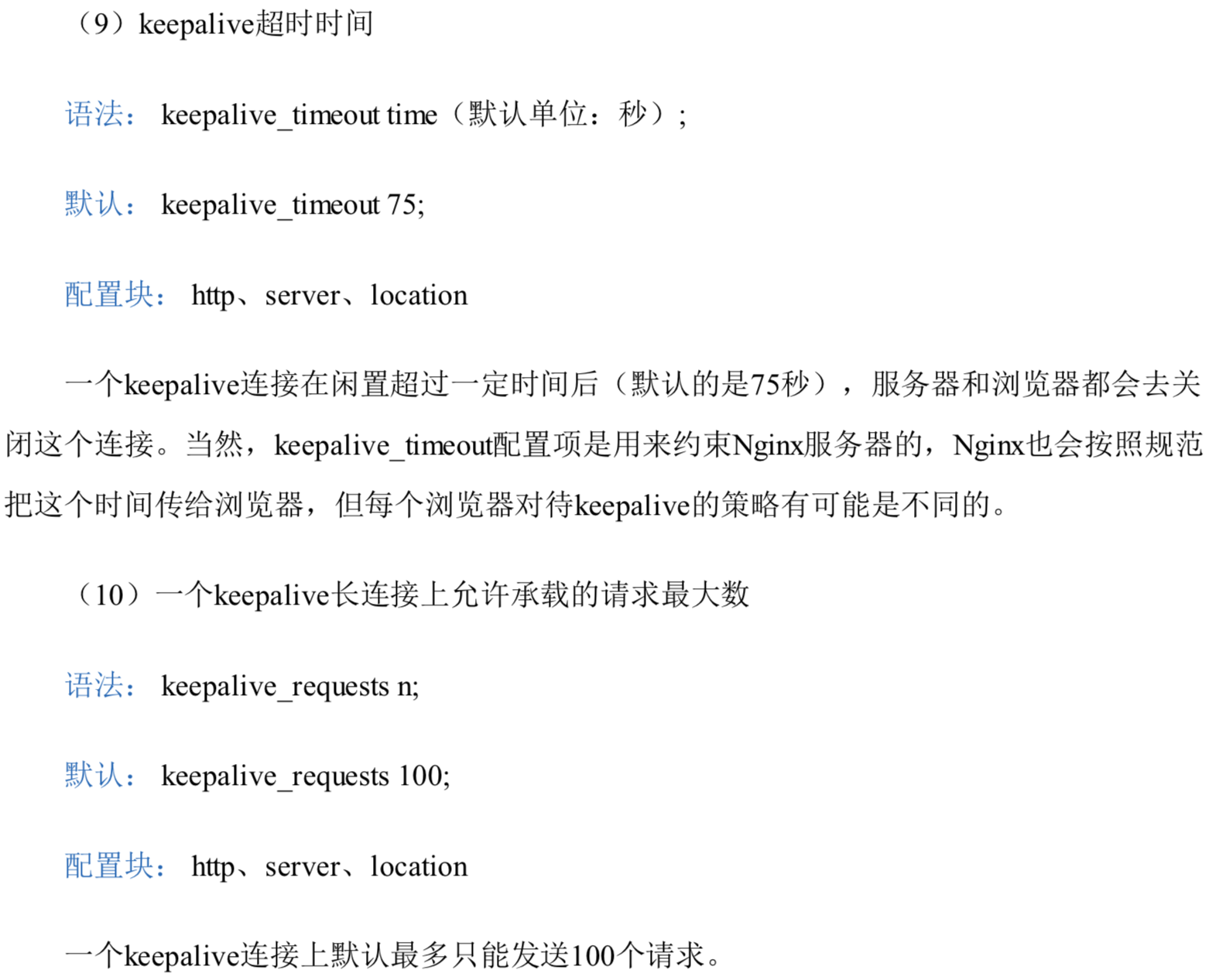
=answer-code：改变状态码

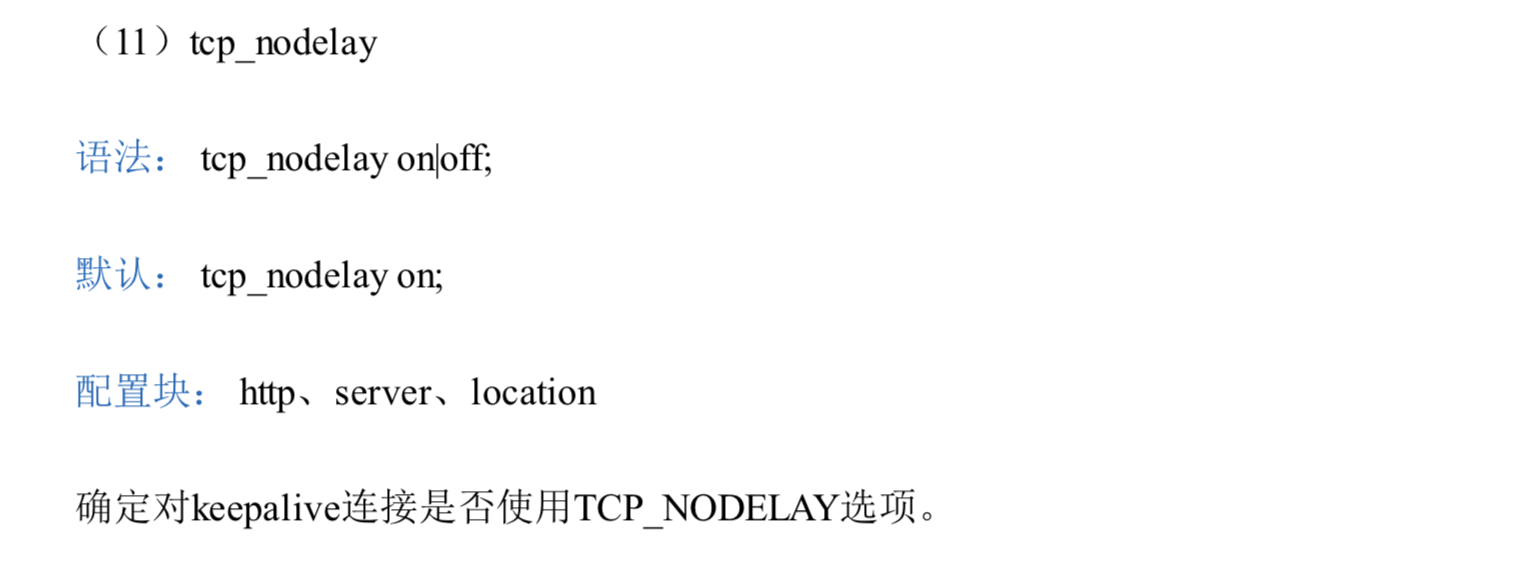
=: 由重定向的服务确定状态码

: 不修改状态码

### 网络连接相关

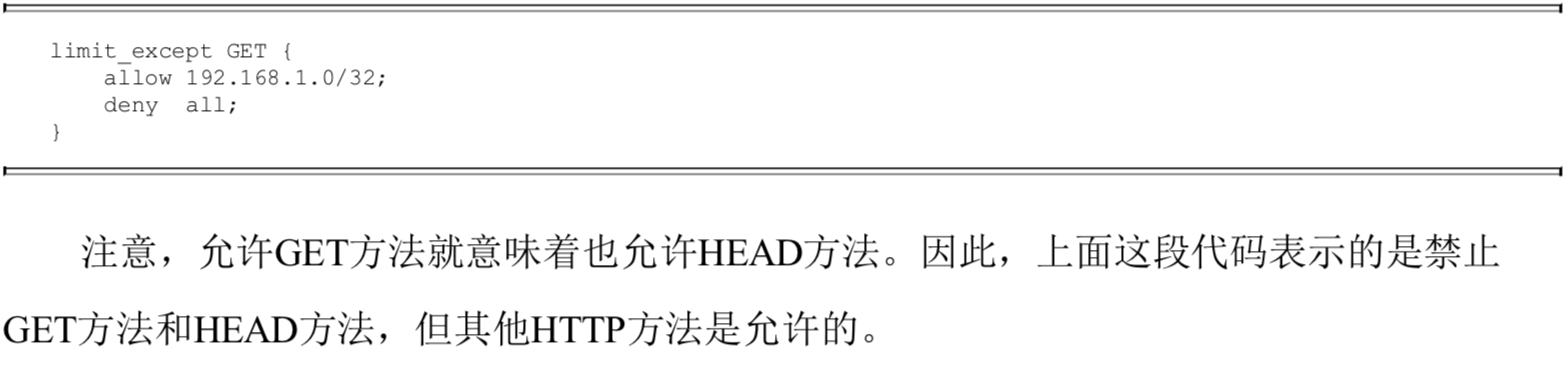


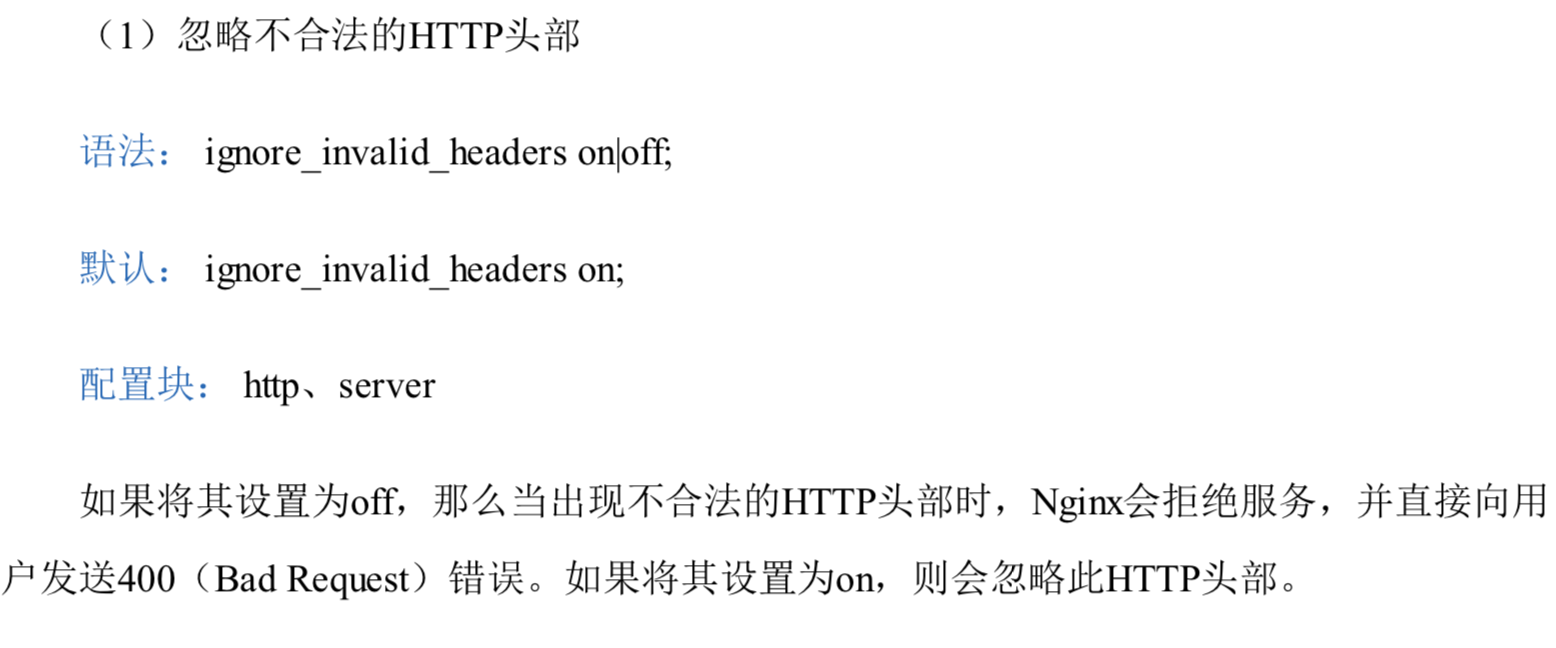




### 客户端限制

请求方法，包体大小，请求限速，是否忽略不合法头部，是否允许下划线，





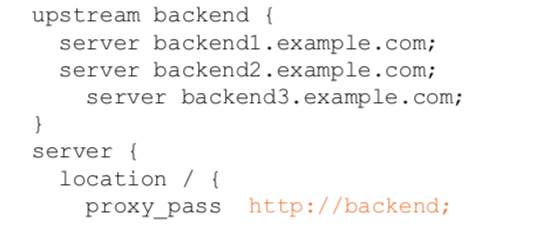
### 缓存

syntax: if\_modified\_since [off|exact|before]

default: if\_modified\_since exact

## 变量

### 负载均衡



也就是说http://backend是一个虚拟的服务，不是真正的后端服务。

server

server xxxx:8080 weight=1 max\_fails=2 fail\_timeout=30s;



access\_log

定义日志格式



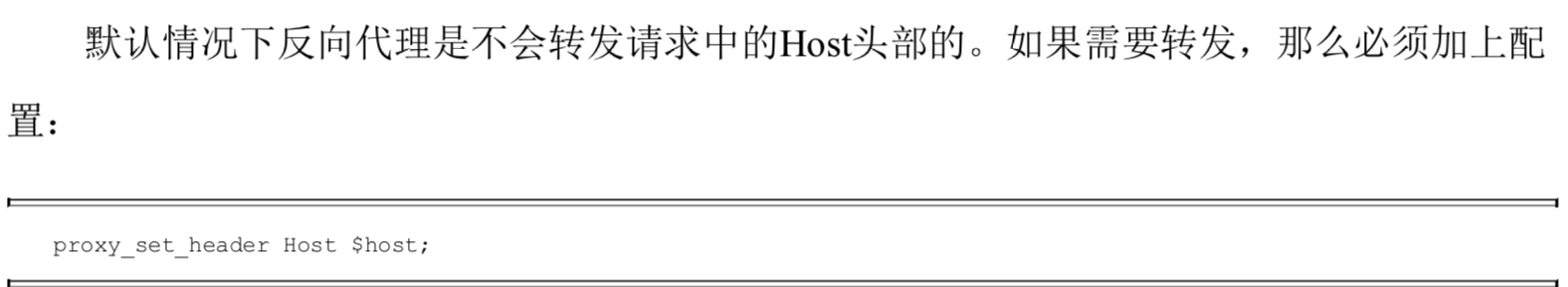
使用规则

access\_log path log\_format

### 反向代理相关

proxy\_pass

可以把http的转发到https上。







proxy\_redirect [ default|off|redirect replacement ]

重定向时，重设location和refresh



proxy\_next\_upstream

## 指令

ngx\_http\_upstream\_module

# 其他

### 正则

### rewrite

看文件路径那一节

location ~ ^/shop {

rewrite ^/shop/(crowdfunding) https://$host/w/$1 redirect;

rewrite ^/shop/(.+) https://$host/$1 redirect;

rewrite ^/shop/ https://$host/main redirect;

# 常见错误

## 13: Permission denied

sudo nginx