# Nginx\_02

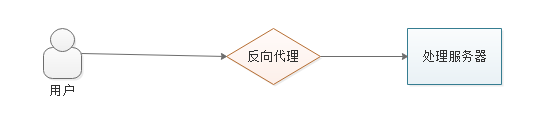
# 今日目标3

* 能够了解负载均衡的原理与意义
* 能够解释session访问失效的原因
* 能够理解session共享的原理

# 一、反向代理

## 1.1、概述

反向代理是作用在服务器端的，对于用户的一个请求，会转发到后端处理服务器中来处理该具体请求



## 1.2、配置语法

# 设置代理后端服务器的协议和地址

语法: proxy\_pass URL;

默认值: —

上下文: location, if in location, limit\_except

例：<http://ip>:端口

# 允许重新定义或者添加发往后端服务器的请求头

语法: proxy\_set\_header field value;

默认值:

proxy\_set\_header Host $http\_host;

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

proxy\_set\_header X-Real-IP $remote\_addr;

上下文: http, server, location

# 超时时间设置

语法: proxy\_connect\_timeout time;

默认值: 秒

proxy\_connect\_timeout 60;

上下文: http, server, location

语法: proxy\_send\_timeout time;

默认值: 秒

proxy\_send\_timeout 60;

上下文: http, server, location

语法: proxy\_read\_timeout time;

默认值: 秒

proxy\_read\_timeout 60;

上下文: http, server, location

## 1.3、实操作

### 1.3.1、反向代理服务器



代理服务器中进行配置

server {

listen 80;

server\_name php.1314000.cn;

location / {

# 注意先后顺序

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

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

proxy\_pass http://127.0.0.1:8080;

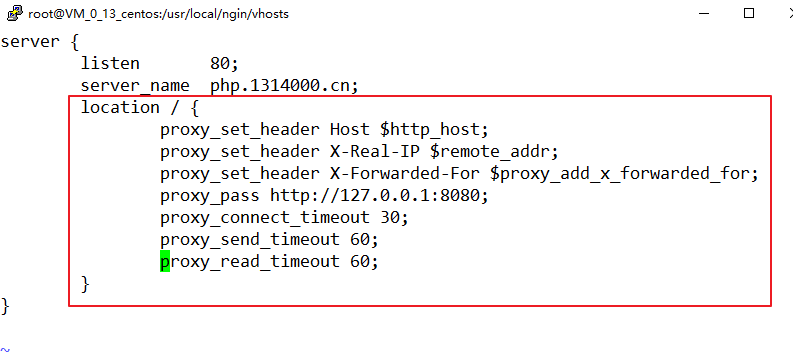
proxy\_connect\_timeout 30;

proxy\_send\_timeout 60;

proxy\_read\_timeout 60;

}

}



web服务器

server {

listen 8080;

server\_name php.1314000.cn;

autoindex on;

autoindex\_exact\_size on;

autoindex\_localtime on;

root /data/php.1314000.cn;

access\_log /usr/local/ngin/logs/success.log main;

location / {

index index.html index.htm index.php;

}

location ~\* \.(jpg|jpeg|gif|bmp|png|ico|svg|mp4|mp3){

expires 1d; # 缓存1天

valid\_referers none blocked \*.1314000.cn 1314000.cn;

if ($invalid\_referer) {

return 403;

}

}

location ~ \.php$ {

fastcgi\_pass 127.0.0.1:9000;

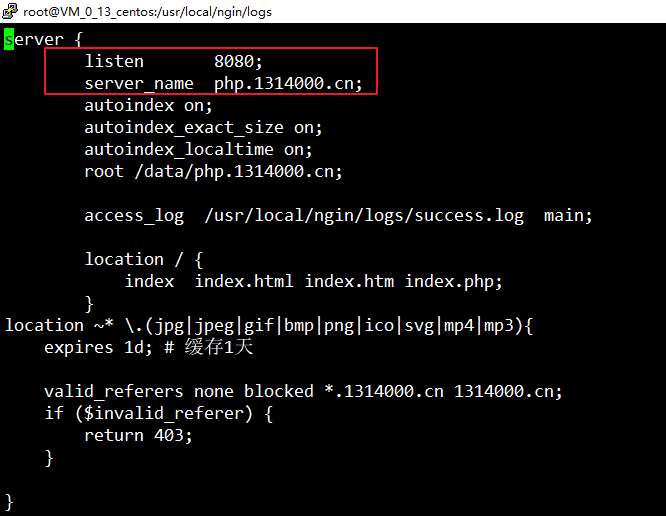
#fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

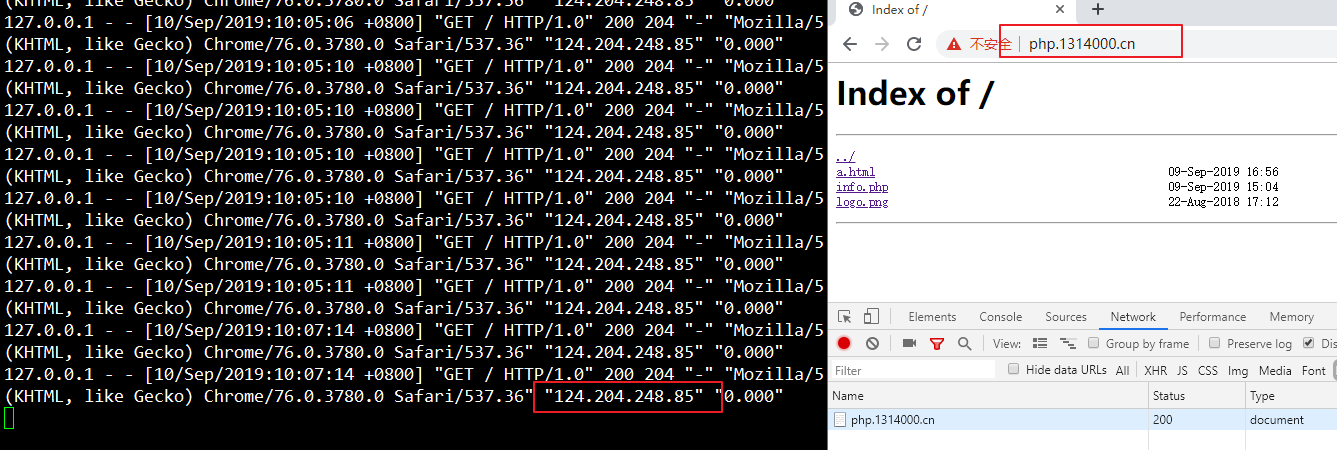
include fastcgi\_params;

}

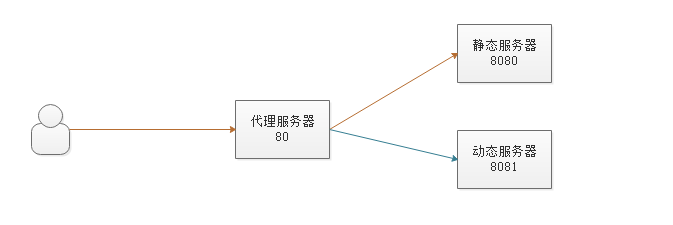
}



效果，在web服务器中看到来源的IP



### 1.3.2、动静分离



代理服务器

server {

listen 80;

server\_name php.1314000.cn;

location / {

# 重写

rewrite ^(.\*)$ http://$http\_host/a.html redirect;

}

# 静态服务器

location ~ \.(html|htm|css|js|png|jpg|gif|jpeg|icon)$ {

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

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

proxy\_pass http://127.0.0.1:8080;

proxy\_connect\_timeout 30;

proxy\_send\_timeout 60;

proxy\_read\_timeout 60;

}

# 动态服务器

location ~ \.php$ {

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

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

proxy\_pass http://127.0.0.1:8081;

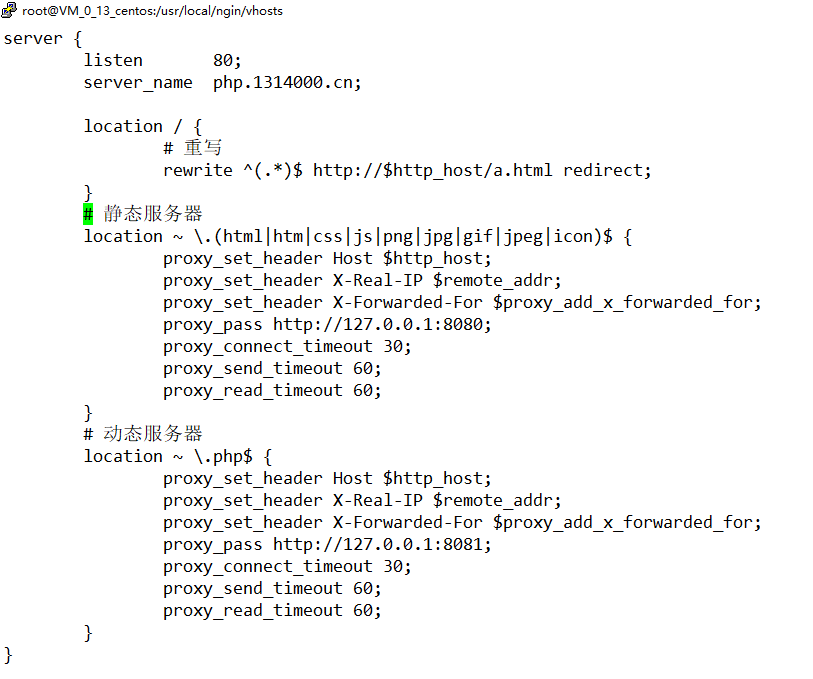
proxy\_connect\_timeout 30;

proxy\_send\_timeout 60;

proxy\_read\_timeout 60;

}

}



静态服务器

server {

listen 8080;

server\_name php.1314000.cn;

autoindex on;

autoindex\_exact\_size on;

autoindex\_localtime on;

root /data/php.1314000.cn;

access\_log /usr/local/ngin/logs/success.log main;

location / {

index index.html index.htm a.html;

}

location ~\* \.(jpg|jpeg|gif|bmp|png|ico|svg|mp4|mp3){

expires 1d; # 缓存1天

valid\_referers none blocked \*.1314000.cn 1314000.cn;

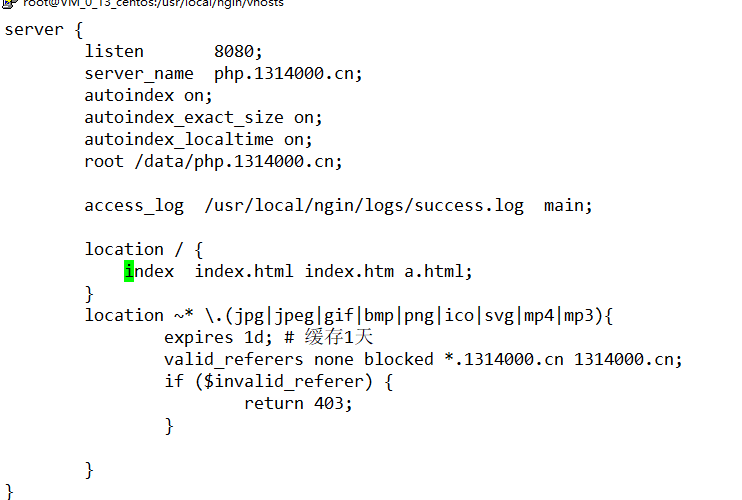
if ($invalid\_referer) {

return 403;

}

}

}



动态服务器

server {

listen 8081;

server\_name php.1314000.cn;

autoindex on;

autoindex\_exact\_size on;

autoindex\_localtime on;

root /data/web8081;

access\_log /usr/local/ngin/logs/success.log main;

location / {

index index.html index.htm index.php;

}

location ~ \.php$ {

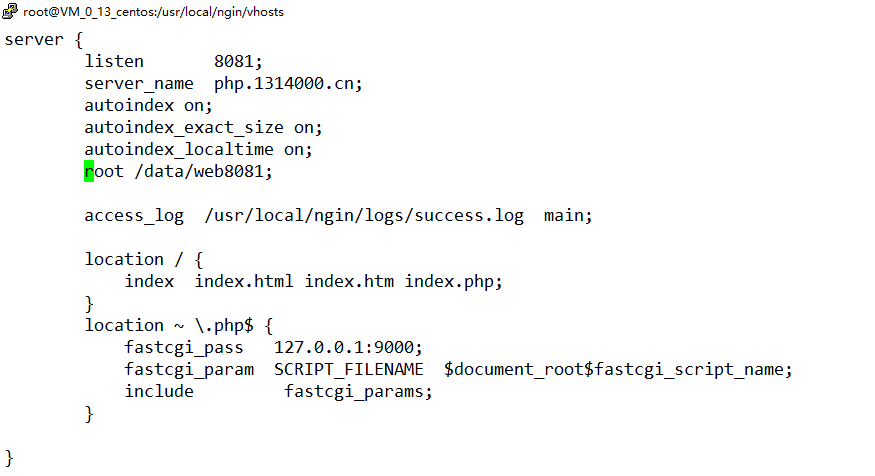
fastcgi\_pass 127.0.0.1:9000;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

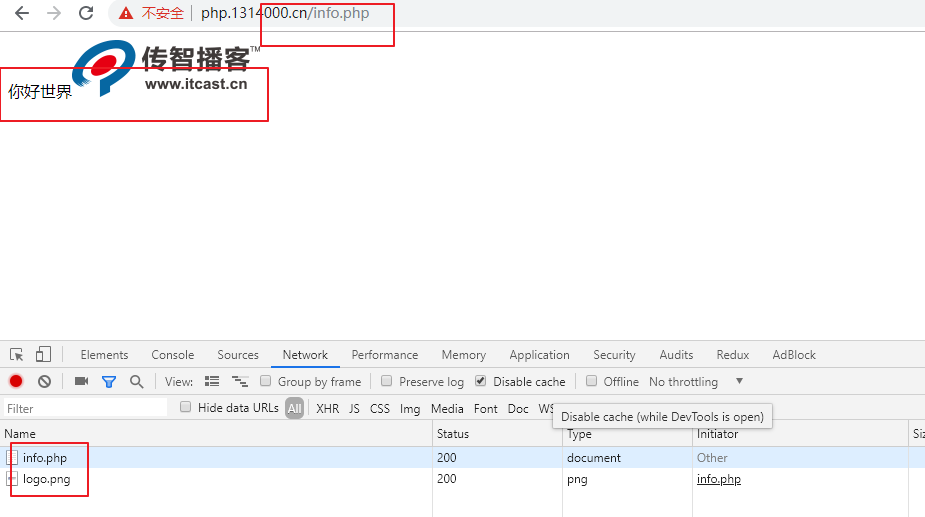
include fastcgi\_params;

}

}



实测效果

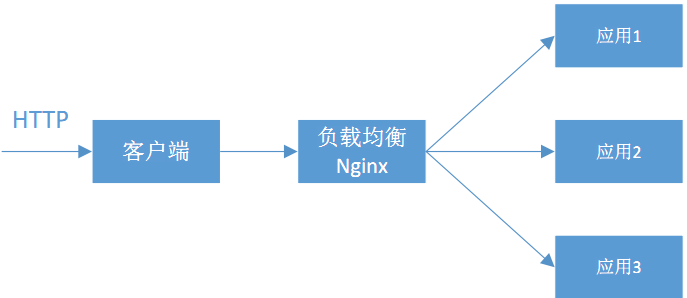


# 二、负载均衡

## 2.1、概述

Nginx是一个配置在七层模型中应用层中的负载均衡服务器。

负载均衡是高可用网络基础架构的的一个关键组成部分，有了负载均衡，我们通常可以将我们的应用服务器部署多台，然后通过负载均衡将用户的请求分发到不同的服务器用来提高网站、应用、数据库或其他服务的性能以及可靠性。



## 2.2、配置语法

# 定义后端服务器列表容器

语法: upstream name { ... }

默认值: —

上下文: http

例：

upstream web {

server 127.0.0.1:8080;

server 127.0.0.1:8081;

}

############# upstream参数说明

weight=number 设定服务器的权重，默认是1

max\_fails=number 允许请求失败的次数

fail\_timeout=时间s 经过max\_fails失败后，服务暂停的时间

down 标记服务器永久不可用

############# 负载均衡算法

轮询 按顺序往后依次选择，直到最后一个，然后循环 默认

加权轮询 weight值越大，分配到的访问概率越高

ip\_hash 每个请求按访问ip的hash结果分配，这样来自相同ip就固定访问一个后端服务器，解决session共享问题

hash hash自定义key一般以$request\_uri来自定义key 1.7.2之后才支持的

# 调用

location / {

proxy\_set\_header Host $http\_host;

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

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_connect\_timeout 60s;

proxy\_send\_timeout 60s;

proxy\_read\_timeout 60s;

proxy\_pass <http://web>;

}

## 2.3、实操作

### 2.3.1、默认轮询机制

在代理服务器中http节点配置好后端服务器列表

# http节点中

upstream web {

server 127.0.0.1:8080;

server 127.0.0.1:8081;

server 127.0.0.1:8082;

}



代理服务器中配置负载均衡代理

server {

listen 80;

server\_name php.1314000.cn;

location / {

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

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

proxy\_connect\_timeout 30;

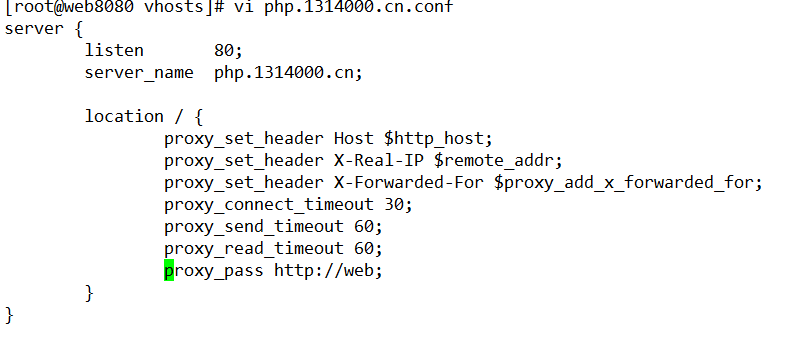
proxy\_send\_timeout 60;

proxy\_read\_timeout 60;

proxy\_pass http://web;

}

}



在后端配置好3台web服务器

server {

listen 8080;

server\_name php.1314000.cn;

autoindex on;

autoindex\_exact\_size on;

autoindex\_localtime on;

root /data/web8080;

access\_log /usr/local/ngin/logs/success\_8080.log main;

location / {

index index.html index.htm index.php;

}

location ~ \.php$ {

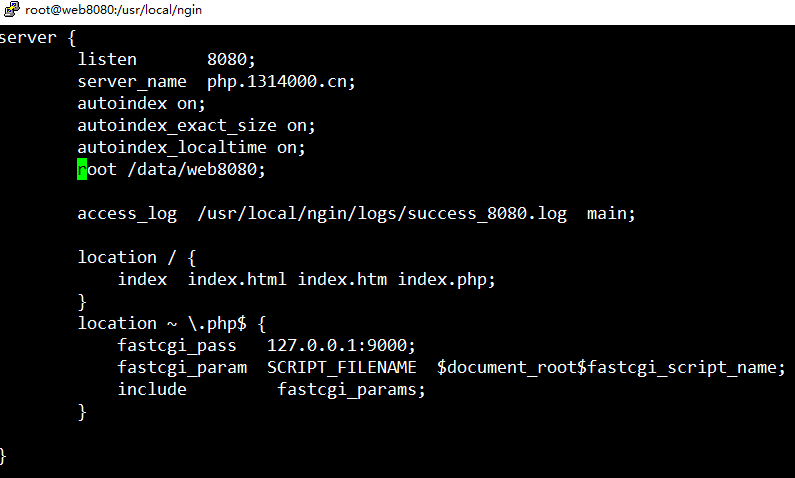
fastcgi\_pass 127.0.0.1:9000;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

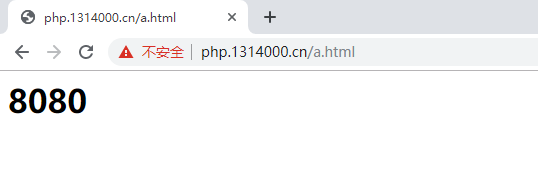
include fastcgi\_params;

}

}



效果



### 2.3.2、加权轮询

在http主节点对upstream配置进行修改，添加权重

upstream web {

server 127.0.0.1:8080 weight=3 max\_fails=3 fail\_timeout=10s;

server 127.0.0.1:8081 weight=1 max\_fails=3 fail\_timeout=10s;

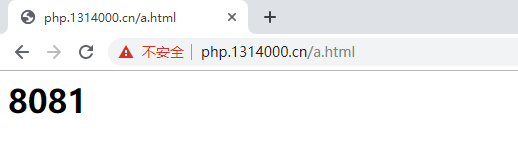
server 127.0.0.1:8082 weight=10 max\_fails=3 fail\_timeout=10s;

}



之前的轮询代理和后台服务保持不变，实现加权轮询。

权重越大，机会越多



### 2.3.3、ip\_hash

同一个IP将会访问同一台服务器，此由可以解决多台服务器集群，session丢失问题解决。

配置ip\_hash只需要在http节点中的upstream服务器列表上面添加一个ip\_hash就可以

upstream web {

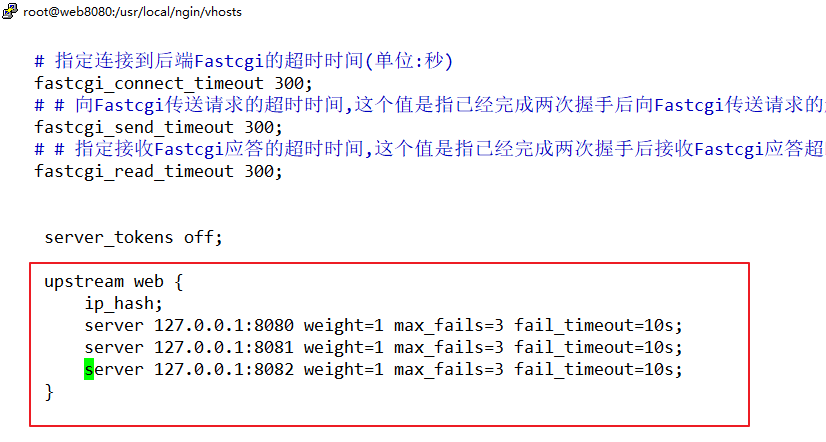
ip\_hash;

server 127.0.0.1:8080 weight=1 max\_fails=3 fail\_timeout=10s;

server 127.0.0.1:8081 weight=1 max\_fails=3 fail\_timeout=10s;

server 127.0.0.1:8082 weight=1 max\_fails=3 fail\_timeout=10s;

}

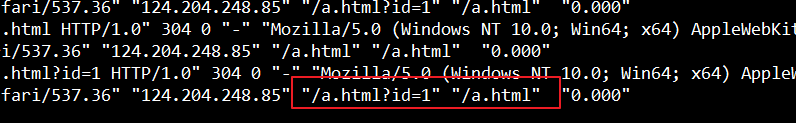


负载均衡配置和后端配置不需要任务修改。

### 2.3.4、hash

此机制是nginx.1.7.2之后提供的.按照自定义key来分配服务器,一般使用$request\_uri/$uri

区别：$request\_uri/$uri



$request\_uri 包含 query参数

$uri 没有包含参数

upstream web {

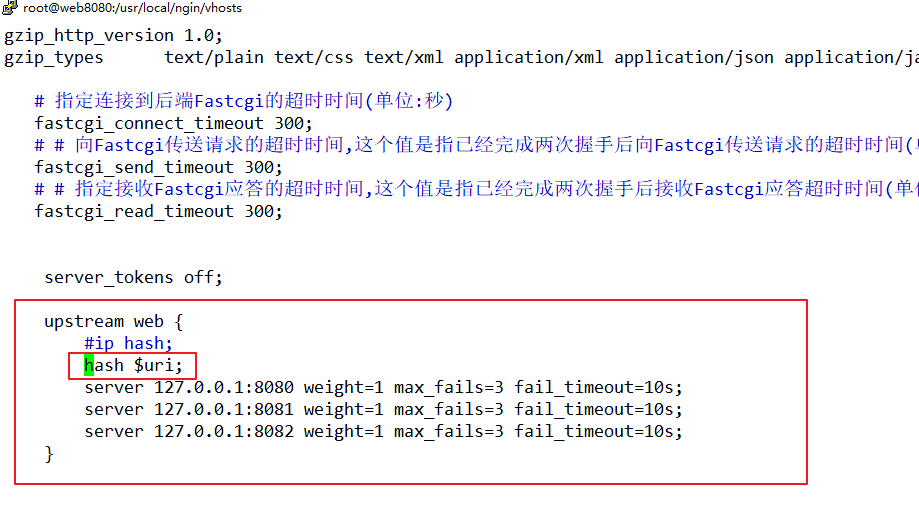
hash $uri;

server 127.0.0.1:8080 weight=1 max\_fails=3 fail\_timeout=10s;

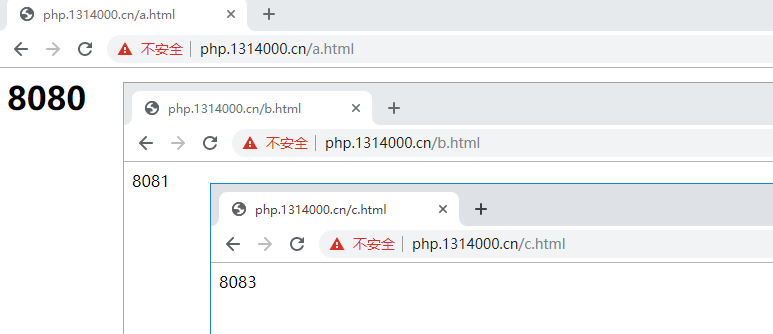
server 127.0.0.1:8081 weight=1 max\_fails=3 fail\_timeout=10s;

server 127.0.0.1:8082 weight=1 max\_fails=3 fail\_timeout=10s;

}



负载均衡配置和后端配置不需要任务修改。



# 三、nginx支持thinkphp

虚拟主机的配置

server {

listen 80;

server\_name php.1314000.cn;

root /data/tp5/public;

access\_log /usr/local/ngin/logs/success\_php1314000cn.log main;

location / {

if (!-e $request\_filename) {

rewrite ^(.\*)$ /index.php?s=/$1 last;

break;

}

index index.html index.htm index.php;

}

# 防攻击

location ~ /thinkphp {

rewrite ^ https://www.baidu.com/ redirect;

break;

}

location ~ \.php$ {

fastcgi\_pass 127.0.0.1:9000;

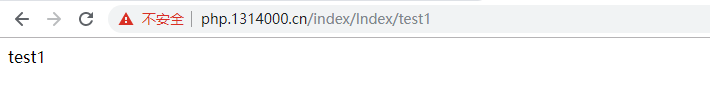
fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

}





网站安全的限制

