Nginx均衡负载-url\_hash实践

温馨提示：

此实验得先编译安装upstream\_hash，详见《Nginx均衡负载upstream\_hash的编译安装》 。

1.首先准备三台服务器

45.77.168.38

198.13.38.179

47.92.7.48

2.分别安装Nginx

安装细节，请参考：《LINUX-Centos平台关于Nginx的编译安装》

3.选择均衡负载器

47.92.7.48

4.配置均衡负载器配置文件

user nginx;

worker\_processes 1;

error\_log /var/log/nginx/error.log warn;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

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

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

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

access\_log /var/log/nginx/access.log main;

sendfile on;

#tcp\_nopush on;

keepalive\_timeout 65;

#gzip on;

##------------LDR 2018-01-08------------##

upstream 47.92.7.48 {

server 45.77.168.38;

server 198.13.38.179;

hash $request\_uri;

}

##------------LDR 2018-01-08------------##

server {

listen 80;

server\_name 47.92.7.48;

charset utf-8;

location / {

root html;

index index.html index.htm;

##------------LDR 2018-01-08------------##

proxy\_pass http://47.92.7.48;

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

client\_max\_body\_size 100m;

##------------LDR 2018-01-08------------##

}

location ~ ^/(WEB-INF)/ {

deny all;

}

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

location = /50x.html {

root /var/www/html/;

}

}

}

注意：

1）.需要注意这两段代码的位置

------------------------------------------------

http {

upstream 47.92.7.48 {

server 45.77.168.38;

server 198.13.38.179;

hash $request\_uri;

}

http->server->location / {

proxy\_pass http://47.92.7.48;

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

client\_max\_body\_size 100m;

------------------------------------------------

2）.注意分清web服务器和均衡负载器

------------------------------------------------

47.92.7.48 #均衡负载器

45.77.168.38 #web服务器

198.13.38.179 #web服务器

------------------------------------------------

5.配制完毕后，记得重启Nginx服务器

cd /usr/local/nginx/sbin

./nginx -s reload

6.访问47.92.7.48

结果：页面一直显示的179！



附注：

1. 按访问url的hash结果来分配请求，使每个url定向到同一个后端服务器，后端服务器为缓存时比较有效。

2.在upstream中加入hash语句，server语句中不能写入weight等其他的参数。