

配置负载1

http://blog.csdn.net/wig_ww/article/details/49924353

nginx: 192.168.4.72 #nginx安装的位置及apache位置

apache: 192.168.4.69 #要代理的两台服务器

apache: 192.168.4.68

一、安装gcc

```
yum install gcc-c++ -y
```

二、安装nginx所需要的依赖库

```
yum -y install zlib-devel openssl-devel pcre-devel
```

三、如有安装老版本，则卸载。

查看: `find -name nginx`

卸载: `yum remove nginx`

四、下载nginx源码并解压。（源码去官网下载， 以下是官方链接）

```
wget -c http://nginx.org/download/nginx-1.6.2.tar.gz
tar -zxvf nginx-1.6.2.tar.gz
mv nginx-1.6.2 nginx
cd nginx
./configure --user=nginx --group=nginx --prefix=/usr/local/nginx
--with-http_addition_module
--with-http_flv_module --with-http_gzip_static_module
--with-http_realip_module --with-http_ssl_module
--with-http_stub_status_module --with-http_sub_module
--with-http_dav_module
```

注：这里--with开头的选项为nginx自带的模块，需要什么就添加，默认是不安装，

```
make
make install
```

五、建立nginx用户及用户组

```
groupadd -r nginx
```

```
useradd -s /sbin/nologin -g nginx -r nginx
```

cd 到nginx目录，看能否启动

sbin/nginx #如果没有报错，说明ok

再ps一下，ps ax | grep nginx，看到如下，说明ok

```
2537 ?      Ss      0:00 nginx: master process sbin/nginx
2538 ?      S       0:00 nginx: worker process
```

nginx在启动后，会有一个master进程和多个worker进程。master进程主要用来管理worker进程，包含：接收来自外界的信号，向各worker进程发送信号，监控worker进程的运行状态，当worker进程退出后(异常情况下)，会自动重新启动新的worker进程。

查看安装的nginx版本：

```
sbin/nginx -v
```

查看安装的编译选项：

```
sbin/nginx -V
```

六、下面开始配置nginx, 及反向代理，编辑配置文件nginx.conf

```
vim /usr/local/nginx/conf/nginx.conf
```

```
user nginx nginx; #这里是nginx运行的用户
```

```

worker_processes 2;                                #设置nginx服务的worker子进程，比如设为2:
error_log logs/error.log;                          #去掉前面的#, 记录nginx错误日志, 方便检查bug:
pid logs/nginx.pid;                                #nginx的pid位置

events {
    worker_connections 1024;                        #每个进程允许的最多连接数,
}

```

```

http {
    include mime.types;
    default_type application/octet-stream;
    #把下面的#去掉, 这是日志配置:
    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 logs/access.log main;                #日志存放位置
}

```

#这里很关键, 很多小伙伴问我 “负载均衡乍配置, 为啥我配置的不能访问呢”, 这里的upstream就是配置负载均衡的, 当然得两台以上才叫负载均衡, 我这里的ip69和68都是

#用的apache, 也许你们的是tomcat, 没关系, 按这样配置一样可以,

```

upstream proxy_test {
    server 192.168.4.69:80 weight=1;                #如果你要测试, 把这里换成你自己要代理后端的ip
    server 192.168.4.68:80 weight=1;
    #ip_hash;                                       #当负载两台以上用ip来hash解决session的问题, 一台就别hash了。
}

```

这是server段的配置

```

server {
    listen 80;
    server_name www.test.com;                       #要访问的域名, 我这里用的测试域名, 如果有多个, 用逗号分开

    charset utf8;

    location / {
        proxy_pass http://proxy_test;               #这里proxy_test是上面的负载的名称, 映射到代理服务器, 可以是ip加端口,
或url
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

```

保存退出!

nginx平滑重启: `nginx -s reload` #加载刚刚加入的配置。

七、测试:

后端服务器开启, 在192.168.4.69和68的网页文件位置添加测试文件test.html, 内容69上: this is test 69, 68上: this is test 68, 这样方便查看访问到哪台了。

在本地配置好host, 在\Windows\System32\drivers\etc\hosts 用记事本打开, 在最后一行加入: 192.168.4.72
www.test.com

然后使用cmd, `ping www.test.com` 是否能ping通这个192.168.4.72地址, 如果ok, 则继续
打开浏览器用www.test.com去访问后端服务器的文件,

如: `www.test.com/test.html`,

浏览器打开显示有this...说明配置ok了。

然后F5刷新一下, 如果是68和69不停的切换, 说明负载均衡ok了。