https://blog.csdn.net/a639735331/article/details/81045631

**ngnix反向代理+负载均衡+session共享**

2018年07月14日 17:25:14 [郝文龙](https://me.csdn.net/a639735331) 阅读数：67 标签： [ngnix](http://so.csdn.net/so/search/s.do?q=ngnix&t=blog) 更多

个人分类： [nginx](https://blog.csdn.net/a639735331/article/category/7396309)

Ngnix

**1.ngnix安装**

一、准备ngnix安装环境

1.安装gcc环境，本环境是ngnix源码编译所依赖的环境

yum install gcc-c++

2.nginx的http模块使用pcre来解析正则表达式

yum install -y pcre pcre-devel

3.nginx使用zlib对http包的内容进行解压或者压缩

yum install -y zlib zlib-devel

4.OpenSSL是一个强大的安全套接字层密码库

yum install -y openssl openssl-devel

二、编译安装

将nginx-1.8.0.tar.gz拷贝至linux服务器

1.解压到指定目录，解压后的文件夹为 nginx-1.8.0

tar -zxvf nginx-1.8.0.tar.gz

2.进入nginx-1.8.0

3.执行以下命令进行参数配置

|  |
| --- |
| ./configure \    --prefix=/usr/local/nginx \    --pid-path=/var/run/nginx/nginx.pid \    --lock-path=/var/lock/nginx.lock \    --error-log-path=/var/log/nginx/error.log \    --http-log-path=/var/log/nginx/access.log \    --with-http\_gzip\_static\_module \    --http-client-body-temp-path=/var/temp/nginx/client \    --http-proxy-temp-path=/var/temp/nginx/proxy \    --http-fastcgi-temp-path=/var/temp/nginx/fastcgi \    --http-uwsgi-temp-path=/var/temp/nginx/uwsgi \    --http-scgi-temp-path=/var/temp/nginx/scgi |

其中 /usr/local/nginx为nginx安装目录，上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录

4.编译安装，执行以下2个命令

make

make install

注：命令是在nginx-1.8.0文件夹下执行的

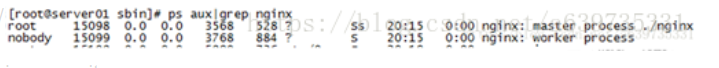
5.启动ngnix

cd /usr/local/nginx/sbin/

./nginx

6.查看linux进程

ps aux | grep nginx



15098是nginx主进程的进程id，15099是nginx工作进程的进程id

7.停止ngnix

cd /usr/local/nginx/sbin

./nginx -s quit

8.重启ngnix,既先停止在启动

./nginx -s quit

./nginx

9.测试

在测试前首先开启linux的 80，81端口

/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT

/sbin/iptables -I INPUT -p tcp --dport 81 -j ACCEPT

否则物理主机无法通过浏览器访问nginx



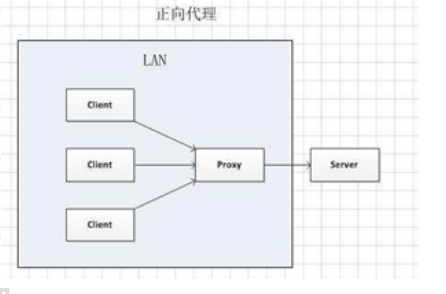
**2.ngnix反向代理**

来自网络的一段描述

**2.1正向代理**

比如你现在缺钱，想找马云爸爸去借钱，可想而知人家可能鸟都不鸟你，到最后碰一鼻子灰借不到钱。不过你认识你家隔壁老王，而老王认识马云同志，而且关系还很好。这时候你托老王去找马云借钱，当然这事最后成了，你从马云那里借到了500万！这时候马云并不知道钱是你借的，只知道这钱是老王借的。最后由老王把钱转交给你。在这里，老王就充当了一个重要的角色：代理。

此时的代理，就是我们常说的正向代理。代理客户端去请求服务器，隐藏了真实客户端，服务器并不知道真实的客户端是谁。正向代理应用最广泛的莫过于现在的某些“科学上网工具”，你访问不了谷歌、Facebook的时候，你可以在国外搭建一台代理服务器，代理你访问，代理服务器再把请求到的数据转交给你，你就可以看到内容了。

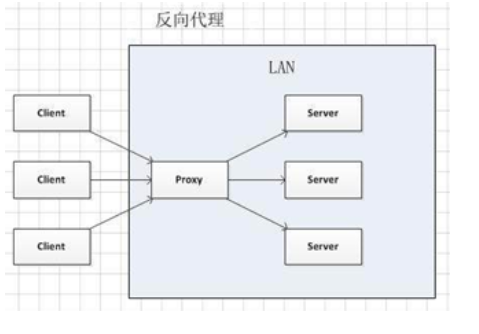


**2.2.反向代理**

比如你现在很无聊，想找人聊天，这时候你拨通了联通客服10010电话，联通的总机可能随机给你分配一个闲置的客服给你接通。这时候你如愿以偿的和客服聊了起来，问了问她目前有没有结婚、有没有对象、家住哪里、她的微信号、她的手机号。。。

此时联通总机充当的角色就是反向代理，你只知道和客服接通并聊了起来，具体为什么会接通这个客服MM，怎么接通的，你并不知道。

反向代理隐藏了真正的服务端，就像你每天使用百度的时候，只知道敲打www.baidu.com就可以打开百度搜索页面，但背后成千上万台百度服务器具体是哪一台为我们服务的，我们并不知道。我们只知道这个代理服务器，它会把我们的请求转发到真实为我们服务的那台服务器那里去



一般常用Nginx来做反向代理服务器，它的性能非常好

**2.3.反向代理配置**

1：安装两个tomcat，分别运行在8080和8081端口

2：启动两个tomcat。

3：反向代理服务器的配置

注：nginx在配置服务器时，端口或者域名只要有一个不同，nginx都认为是2个服务器

4.修改配置文件/usr/local/nginx/conf/nginx.conf

复制server节点,将root改成proxy\_pass,并作以下配置

|  |
| --- |
| upstream tomcat1 {           server 192.168.0.105:8080;  }  server {      listen       80;      server\_name  www.lq1.com;        location / {          proxy\_pass   http://tomcat1;          index  index.html index.htm;      }    }  upstream tomcat2 {           server 192.168.0.105:8081;  }  server {      listen       80;      server\_name  www.lq2.com;        location / {          proxy\_pass   http://tomcat2;          index  index.html index.htm;      }  } |

5、nginx重新加载配置文件   ./nginx -s reload

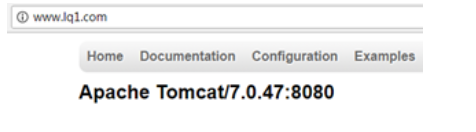
6、配置域名

在本地的hosts文件中添加域名和ip的映射关系（跳过DNS解析直接根据域名访问指定IP）

192.168.0.105 www.lq1.com

192.168.0.105 [www.lq2.com](http://www.lq2.com/)

7.浏览器分别访问2个域名



**3.ngnix负载均衡**

如果www.lq2.com是由多个服务器提供服务(集群)，需要把负载分配到不同的服务器处理，则需要进行负载均衡配置

重新赋值一份tomcat修改端口为8082

|  |
| --- |
| upstream tomcat2 {  server 192.168.0.105:8081;  server 192.168.0.105:8082;  } |

可以根据服务器的实际情况调整服务器权重。权重越高分配的请求越多，权重越低，请求越少。默认是都是1

|  |
| --- |
| upstream tomcat2 {  server 192.168.0.105:8081;  server 192.168.0.105:8082 weight=2;  } |

**4.session共享**

**1.ip\_hash**

nginx中的ip\_hash技术能够将某个ip的请求定向到同一台后端，这样一来这个ip下的某个客户端和某个后端就能建立起稳固的session

ip\_hash是在upstream配置中定义的：

|  |
| --- |
| upstream tomcat2 {  server 192.168.0.105:8081;  server 192.168.0.105:8082 weight=2;;  ip\_hash;  } |

**2. 将session从系统中独立出来**

目前主流做法是利用redis作为session管理的实现，因为redis访问极其快速