 在很多情况下，可能只有一个公网IP地址，但是内部可能有多个服务需要映射出去，如：**www.domain.com  oa.domain.com**并且相关的服务不在同一台服务器上面的时候，那直接做端口映射就无法达到要求了

Server{

Server\_name domain.cn

Proxy\_pass http://192.168.1.200

}

**配置完后,使用nginx –t的命令测试一下,配置是否正确.是否有提示相关的错误.这个服务的路径要以具体的为主. service nginx restart**

Nginx的负载均衡功能和代理功能是经常被用到的

反向代理（Reverse Proxy）方式是指以代理服务器来接受Internet上的连接请求，然后将请求转发给内部网络上的服务器；

Nginx 的优势是在于它的异步阻塞模型，可以通过基于事件的方式同时处理和维护多个请求，而后端只要去做逻辑计算，节约等待时间去处理更多请求。

Nginx提供了两种负载均衡策略：内置策略和扩展策略。内置策略为轮询，加权轮询，Ip hash。扩展策略，就是自己实现一套策略。  
大家可以通过upstream这个配置，写一组被代理的服务器地址，然后配置负载均衡的算法。

热备 <https://www.jianshu.com/p/e98e84a3322f>

当一台服务器发生事故时，才启用第二台服务器给提供服务。  
比如127.0.0.1 挂了，就启动192.168.10.121。

## ginx 负载均衡详解

在文章[Nginx 配置详解](https://www.runoob.com/w3cnote/nginx-setup-intro.html" \t "/home/wqb/Documents\\x/_blank)中我说啦nginx有哪些中负载均衡算法。这一结我就给如何操作配置的给大家做详细说明下。

首先给大家说下upstream这个配置的，这个配置是写一组被代理的服务器地址，然后配置负载均衡的算法。这里的被代理服务器地址有2中写法。

upstream mysvr {

server 192.168.10.121:3333;

server 192.168.10.122:3333;}

server {

....

location ~\*^.+$ {

proxy\_pass http://mysvr; #请求转向mysvr 定义的服务器列表

}}

然后，就来点实战的东西。

1、热备：如果你有2台服务器，当一台服务器发生事故时，才启用第二台服务器给提供服务。服务器处理请求的顺序：AAAAAA突然A挂啦，BBBBBBBBBBBBBB.....

upstream mysvr {

server 127.0.0.1:7878;

server 192.168.10.121:3333 backup; #热备 }

2、轮询：nginx默认就是轮询其权重都默认为1，服务器处理请求的顺序：ABABABABAB....

upstream mysvr {

server 127.0.0.1:7878;

server 192.168.10.121:3333; }

3、加权轮询：跟据配置的权重的大小而分发给不同服务器不同数量的请求。如果不设置，则默认为1。下面服务器的请求顺序为：ABBABBABBABBABB....

upstream mysvr {

server 127.0.0.1:7878 weight=1;

server 192.168.10.121:3333 weight=2;}

4、ip\_hash:nginx会让相同的客户端ip请求相同的服务器。

upstream mysvr {

server 127.0.0.1:7878;

server 192.168.10.121:3333;

ip\_hash;}

5、如果你对上面4种均衡算法不是很理解，可以查看[Nginx 配置详解](https://www.runoob.com/w3cnote/nginx-setup-intro.html" \t "/home/wqb/Documents\\x/_blank)，可能会更加容易理解点。

到这里你是不是感觉nginx的负载均衡配置特别简单与强大，那么还没完，咱们继续哈，这里扯下蛋。

关于nginx负载均衡配置的几个状态参数讲解。

down，表示当前的server暂时不参与负载均衡。

backup，预留的备份机器。当其他所有的非backup机器出现故障或者忙的时候，才会请求backup机器，因此这台机器的压力最轻。

max\_fails，允许请求失败的次数，默认为1。当超过最大次数时，返回proxy\_next\_upstream 模块定义的错误。

fail\_timeout，在经历了max\_fails次失败后，暂停服务的时间。max\_fails可以和fail\_timeout一起使用。

upstream mysvr {

server 127.0.0.1:7878 weight=2 max\_fails=2 fail\_timeout=2;

server 192.168.10.121:3333 weight=1 max\_fails=2 fail\_timeout=1; }

到这里应该可以说nginx的内置负载均衡算法已经没有货啦。如果你像跟多更深入的了解nginx的负载均衡算法，nginx官方提供一些插件大家可以了解下。