

Nginx

题目难度：★★★

知识点标签：Nginx

学习时长：20分钟

题目描述

Nginx正向代理与反向代理

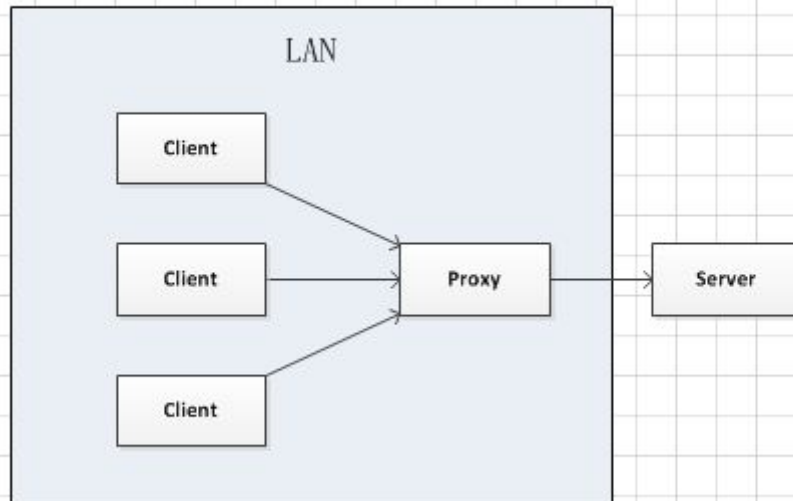
介绍

实践中客户端无法直接跟服务端发起请求的时候，我们就需要代理服务。代理可以实现客户端与服务端之间的通信,我们的Nginx也可以实现相应的代理服务。代理分为正向代理和反向代理。

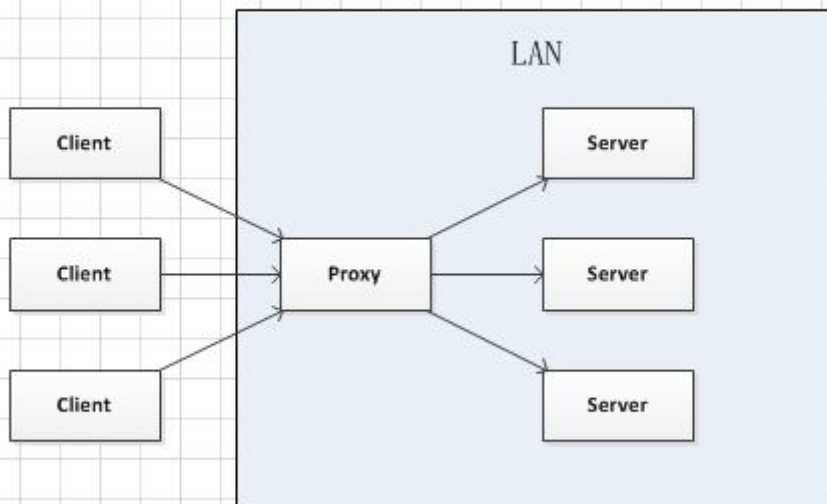
正向代理和反向代理的区别

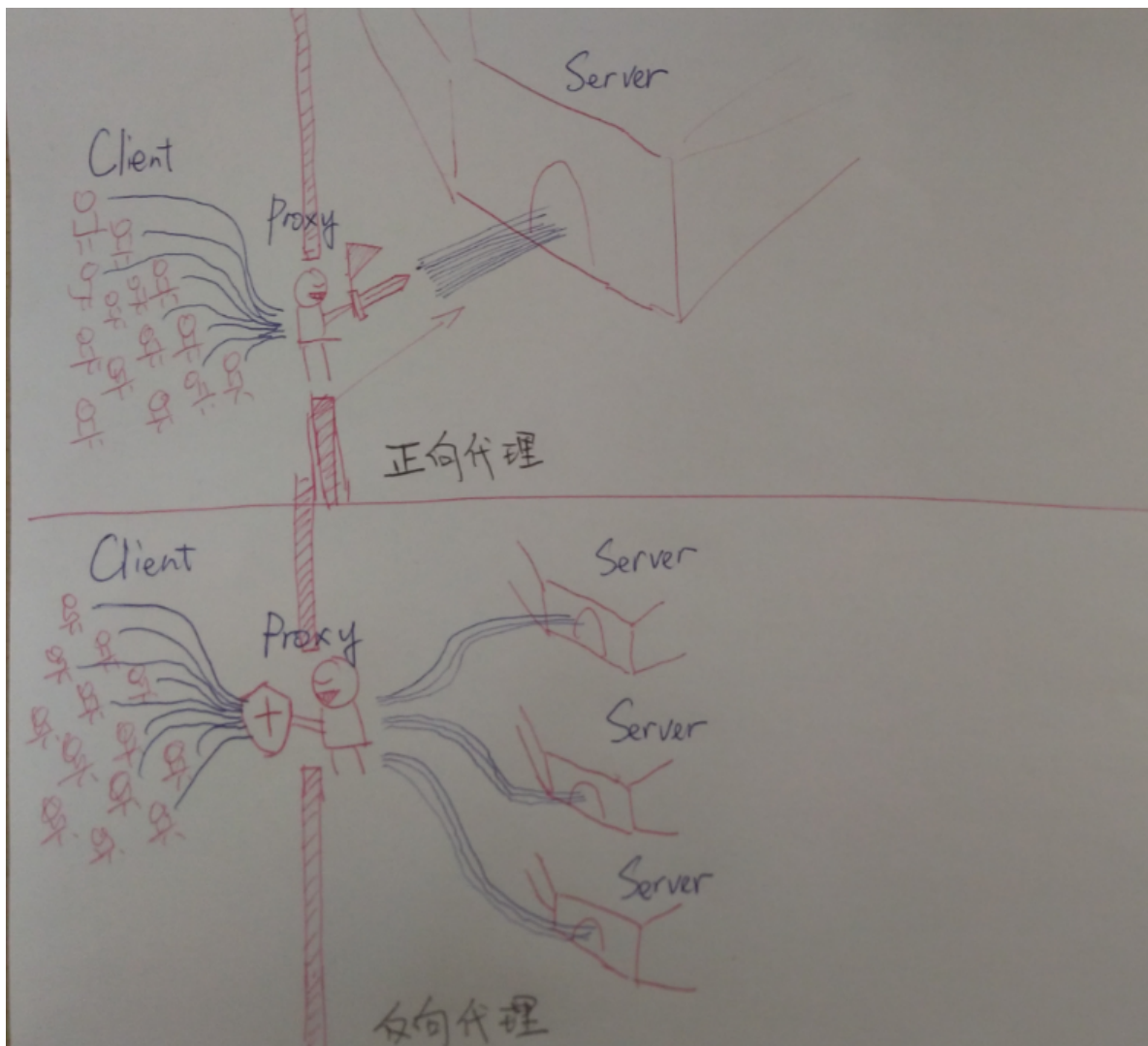
正向代理和反向代理的区别找到两张图可以帮助我们很好的理解：

正向代理



反向代理





正向代理:客户端 <—> 代理 —> 服务端

正向代理简单地打个租房的比方:A(客户端)想租C(服务端)的房子,但是A(客户端)并不认识C(服务端)租不到。

B(代理)认识C(服务端)能租这个房子所以你找了B(代理)帮忙租到了这个房子。这个过程中C(服务端)不认识A(客户端)只认识B(代理)

C(服务端)并不知道A(客户端)租了房子,只知道房子租给了B(代理)。

反向代理:客户端 —>代理 <—> 服务端

反向代理也用一个租房的例子:A(客户端)想租一个房子,B(代理)就把这个房子租给了他。

这时候实际上C(服务端)才是房东。

B(代理)是中介把这个房子租给了A(客户端)。这个过程中A(客户端)并不知道这个房子到底谁才是房东

他都有可能认为这个房子就是B(代理)的由上的例子和图我们可以知道正向代理和反向代理的区别在于代理的对象不一样,正向代理的代理对象是客户端,反向代理的代理对象是服务端。

Nginx代理的配置演示

正向代理很常见,我们的科学上网就是一种正向代理。

我们接下来演示正向代理的这么一个场景。

首先我在我的A服务器的nginx设置访问控制

访问控制之前我访问A下的test.html是这样的:



hello!test.html,tangll add it.

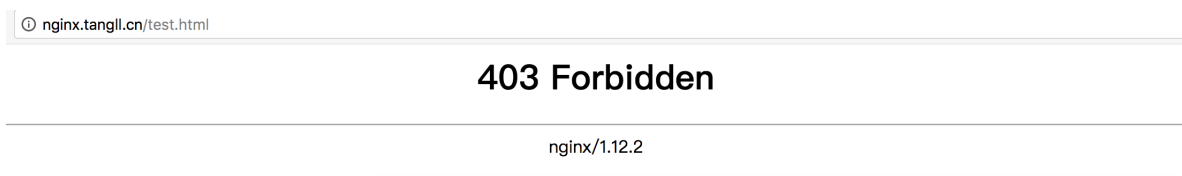
我们打开/etc/nginx/conf.d/default.conf

我们加入这么一个判断语句

如果访问A的IP不是118.126.106.11(我的B服务器)则返回403.

```
location / {
    if ( $remote_addr !~* "^118.126.106.11" ) {
        return 403;
    }
    root /opt/app/demo/html;
    index index.html index.htm;
}
```

添加后reload一下nginx再访问test.html:



此时本地我的浏览器就是被限制了,访问不了该资源。

现在我登录上我的B服务器,打开/etc/nginx/conf.d/default.conf

添加resolver和proxy_pass,设置如下:

```
server {
    listen 80;
    server_name localhost nginx.tangll.cn;

    resolver 8.8.8.8;
    location / {
        proxy_pass http://$http_host$request_uri;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }
}
```

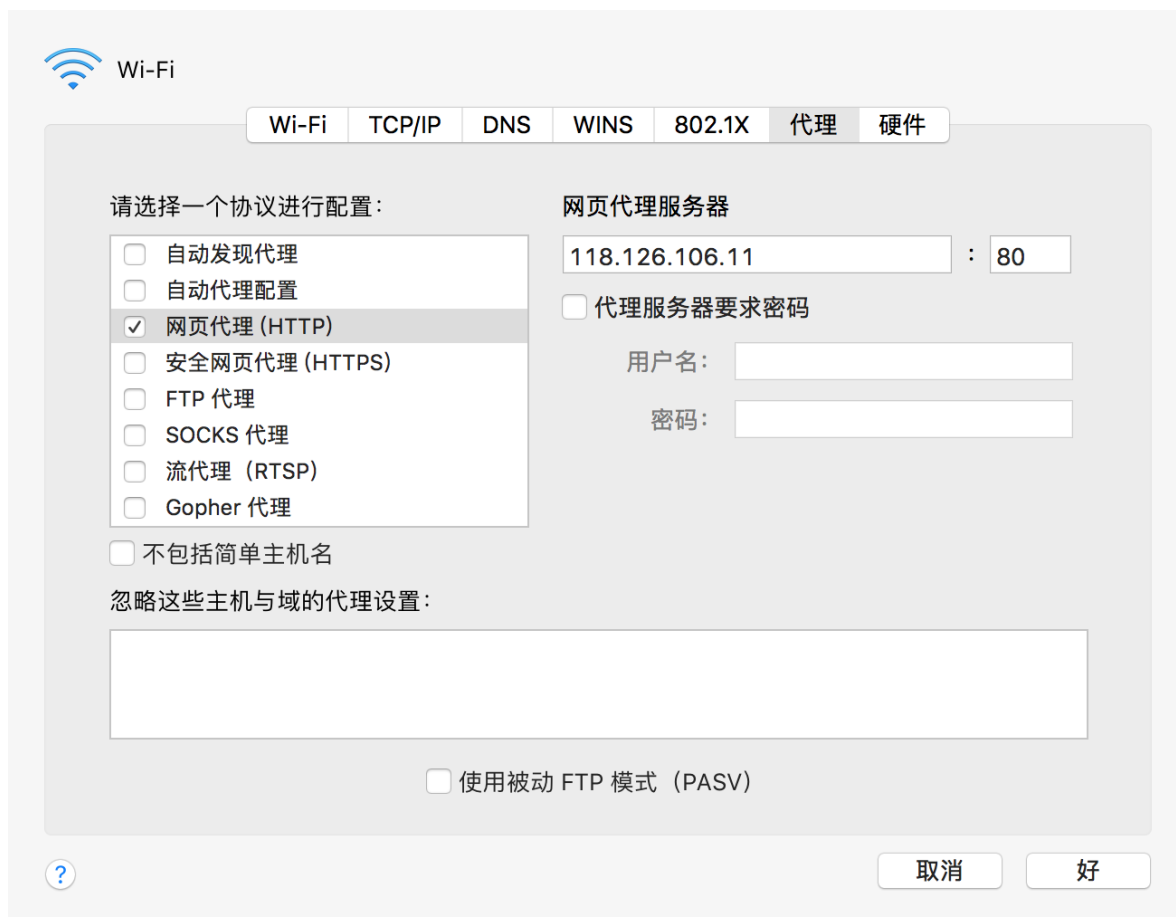
resolver为DNS解析,这里填写的IP为Google提供的免费DNS服务器的IP地址

proxy_pass配置代理转发

至此便是配置了B服务器所有访问根一级的请求全部都代理转发对应到\$http_host\$request_uri去了,\$http_host就是我们要访问的主机名,\$request_uri就是我们后面所加的参数。

简单的说至此就是相当于配置好了我们请求了B服务器,B服务器再去请求我们所请求的地址。

那么接下来我们来看一下结果,我们在本地配置好代理,我这里是mac系统,可以从网络设置中选择高级,然后选择代理



填入我们B服务器的IP,然后我们来看一下代理是否成功。
我们登录

<http://www.ip138.com/>

可以看到此时我们的IP地址已经为B服务器的IP,说明代理成功。

www.ip138.com iP查询(搜索iP地址的地理位置)

您的IP是: [118.126.106.11] 来自: 广东省广州市 腾讯集团

然后我们再来访问一下test.html:



结果证明,此时的客户端已经可以成功访问A服务器的资源。以上就是正向代理的一个场景演示,这个过程中可以知道,我们客户端是想要A的资源,但是A的资源只有B能拿到,便让B代理去帮助我们访问A的资源。整个过程A只知道B拿了他的资源,并不知道客户端拿到。

反向代理配置场景演示

反向代理的演示更为简单一些。

首先在/etc/nginx/conf.d/下新建一个test.conf:server {

```
listen    8080;
```

```
server_name localhost nginx.tangll.cn;
```

```
location / {
```

```
    root /opt/app/demo/html;
```

```
    index index.html index.htm;
```

```
}
```

```
error_page 500 502 503 504 404 /50x.html;
```

```
location = /50x.html {
```

```
    root /usr/share/nginx/html;
```

```
}
```

```
}
```

可以看到我server里listen的是8080端口,但是我的服务器本身不对外开放8080端口,只开放了80端口。

所以我们此时访问test.html结果是访问不到的:



然后我们打开我们的/etc/nginx/conf.d/default.conf

添加proxy_pass设置如下:server {

```
listen    80;
```

```
server_name localhost nginx.tangll.cn;
```

```
location / {
```

```
    root /usr/share/nginx/html;
```

```
    index index.html index.htm;
```

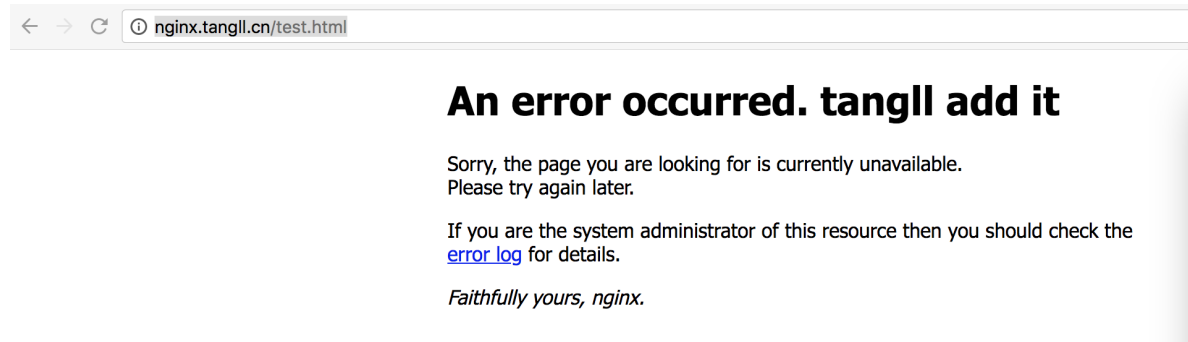
```
}
```

```
\#设置代理
\#location ~ /test.html$ {
\#    proxy_pass http://127.0.0.1:8080;
\#}
```

```
error_page 500 502 503 504 404 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}
}
```

我们设置当匹配test.html结尾的URL时就去代理访问本机的8080端口

为了对比我们先注释掉,然后直接80端口访问一下test.html:



可以看到此时返回的404。

这时候取消注释我们reload一下nginx然后用80端口访问test.html



hello!test.html,tangll add it.

此时便可访问8080端口配置的资源。

以上便是完成了一个反向代理的演示,这个过程中我们可以知道,客户端想要访问的是test.html,但是test.html实际上是8080端口下配置的,中间经过了代理才能拿到。也就是说客户端并不知道中间经历了什么代理过程,只有服务端知道。客户端只知道他拿到了test.html也就是8080端口下配置的资源内容。

总结

- 由上的打比方和演示例子可以体会到正向代理与反向代理的区别和Nginx正向代理和反向代理的简单配置。正向代理和反向代理的区别上边也说过在于代理的对象不一样,正向代理的代理对象是客户端,反向代理的代理对象是服务端。

最后一句话总结此文就是

代理服务器站在客户端那边就是正向代理,

代理服务器站在原始服务器那边就是反向代理,

Nginx通过proxy_pass可以设置代理服务。