

SSH反向通道暴露内网主机80端口——作为Nginx的upstream后端

作者: [unhappydepig](#)

原文链接: <https://ld246.com/article/1469760064017>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h5 id="背景">背景</h5>

<p>在开发拇指动力项目的时候遇到了微信JS接口问题。微信开发的时候，需要提供一个已经备案的域名才能调用api。这里假设我的域名是domain.cm，我把weixin.domain.com解释到一个公网IP。我的项目就部署到weixin.domain.com上面，微信入的URL就是http://weixin.domain.com/gateway.php。现在有个问题，我的项目有一些bug， 而不是肉眼能看出来的，我需要debug。但是微信发送给公众账号的消息是发送到http://weixin.domain.com/gateway.php，即发到送到我的服务器上。我在内网做开发的主机没法收到请求。没法debug 于是就有了如下法子。</p>

<h5 id="概念">概念</h5>

SSH反向通道

<p>SSH反向通道可以将公网服务器的某个端口映射到内网主机的某个端口上来。关于SSH反向通道 以参考下面两篇文章，</p>

<p>SSH隧道技术简介：端口转发&a p;SOCKS代理</p>

<p>实战 SSH 端 转发</p>

Nginx反向代理和负载均衡

<p>Nginx的反向代理可以把接收到的客户端请求代理到后端的处理器，得到结果后再把结果响应给 户端。而负载均衡就是上面提到的处理器的一个提供方式，一般是几个处理器并列。一个死了，另一 顶上。</p>

<h5 id="配置">配置</h5>

<p>我的部署方案是这样的，在www.domain.com上面部署了一个Nginx服务器，一个8989端口的p p-fpm。我在内网做开发的主机上部署了一个9090端口的php-fpm。</p>

<p>现在我的目标是让nginx做前端，而两个php-fpm做后端的负载均衡。而且当内网的开发主机开 的时候，要求nginx把请求发送到开发主机上来。当内网的开发主机不开启的时候，nginx就把请求代 到公网服务器上的php-fpm上，即8989端口。</p>

<p>这里要做两件事，</p>

让内网的9090端口暴露出去，以便Nginx能把请求代理到这个端口来。我的做法是用ssh的反向 道，如下

<pre><code class="hljs css">> ssh -f -N -q -R 9090:localhost :80 sshadmin@weixin.<spa

```
class="hljs-keyword">domain</span>.<span class="hljs-keyword">com</span>
```

> 修改weixin.domain.com上面的sshd配置文件/etc/ssh/sshd_config，在最后面添加 Gateway ports yes。 **这一点非常重要，是为了让9090端口的监听地不局限于127.0.0.1**

> * -f *表示后台运行。

```
</code></pre>
```

<p>经过刚才的配置，在浏览器上就可以访问weixin.domain.com:9090。 </p>

```
<ol>
```

```
<li>配置Nginx，很简单，两个配置： </li>
```

```
</ol>
```

```
<pre><code class="hljs markdown"><span class="hljs-quote">&gt; 添加一个均衡负载器，**意下面的*backup*关键字**。 </span>
```

```
<span class="hljs-code"> ``
```

```
</span></code></pre>
```

```
<p>upstream weixin { server weixin.domain.com:8989 backup; server weixin.domain.com:9090; }</p>
```

```
<pre><code class="hljs javascript"><span class="hljs-string">`</span><span class="hljs-string">`
```

> 添加一个命名虚拟主机

```
`</span><span class="hljs-string"> `</span>
```

```
</code></pre>
```

```
<p>server { listen 80; server_name weixin.domain.com; location / { proxy_pass &nbsp;<a href="http://weixin/">http://weixin</a>; } } ``</p>
```

<p>到这里所有的配置就完成了。现在我在浏览器访问weixin.domain.com的时候，Nginx就会默认请求代理到weixin.domain.com:9090，然后通过ssh reverse tunnel来到我的开发机器上了，而当的开发机器关机的时候，请求就交给了weixin.domain.com:8989。这样就实现了开发生产两不误。 </p>

<p>转自: http://my.oschina.net/atanl/blog/375997 </p>