



链滴

Docker 中部署基于 Ngrok 的内网穿透服务

作者: [Gouzhong1223](#)

原文链接: <https://ld246.com/article/1582460612314>

来源网站: [链滴](#)

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

<p></p>

<h2 id="为什么要内网穿透">为什么要内网穿透</h2>

<p>我们在做微信开发的时候，避免不了的就是验证自己的服务器 URL</p>

<p></p>

<p>并且，这个 URL 是只能填写域名的，不能够填写 IP 地址，但是，我们在开发调试期间，不可能真的要把自己的服务打包到云服务器上，然后再用域名解的方式解析到自己的云服务器。这是非常不友好的一种方式，姑且不说部署测试环境花的时间，光是程调试就会遇到很多猝不及防的问题。所以我们就需要做一下内网穿透，将一个公网域名映射到自己本地环境上面来，这样就大大简化了我们调试的步骤。</p>

<p>PS：文末有惊喜！！</p>

<h2 id="常见的内网穿透">常见的内网穿透</h2>

<p>既然内网穿透用处这么大，那肯定就会有互联网公司会推出便携式内网穿透的服务，常见的便携内网穿透就有 natapp,sunny ngrok,以及花生壳(:thumbsdown:),当然，既然是便携式内网穿透，我肯定是需要支付一定的 Money 才能有好的体验，虽然有免费的，比如 natapp，它的免费套餐一次重新启动都会更换一个域名，端口也会随机改变，着很显然不符合我们做微信开发的业务需求。句话说完，要想在便携式内网穿透获得良好的体验，这肯定是需要我们付出一点代价的哈哈哈哈哈。</p>

<h2 id="屌丝逆袭-做自己的内网穿透服务器">屌丝逆袭-做自己的内网穿透服务器</h2>

<p>既然便携式的要花钱，那我们就自己部署一个内网穿透的服务</p>

<h3 id="我们需要准备的东西有">我们需要准备的东西有</h3>

一个带公网 IP 的服务器

一个已经备案的域名(二级域名最好，顶级域名拿来做了这个浪费了哈哈哈哈哈)

<p>我这里用的是华为的一台 1 核 2G 的服务器，域名的话，我就用 <code>ngrok.gouzhong1223.om</code> 来做演示</p>

<p></p>

<p>建议大家都拿一些小的，或者是闲置的服务器来部署这个，毕竟这个不是我们的主线业务，我的一台服务器也是闲置的，就只是部署了一个私有云项目(如本文最后的资源链接)。</p>

<h3 id="准备工作">准备工作</h3>

<p>我们来到自己的域名控制面板，解析两个 A 记录到我们的服务器上面来</p>

<p></p>

<p>这里我解析了一个泛域名到服务器上是为了方便以后拓展，意思就是类似于 <code>xxx.ngrok.gouzhong1223.com</code> 的域名我们以后也能进行穿透。</p>

<h3 id="开始正式搭建">开始正式搭建</h3>

<p>本次搭建的环境是依赖了一个 Docker 镜像的，点击这里跳转到镜像仓库。</p>

<p></p>

<p>我们将镜像拉倒宿主机上面来</p>

<p></p>

<p>完成之后输入命令 <code>docker images</code>，出现这个就说明我们拉取成功了！</p>

<p></p>

<p>我们直接运行下面这个命令</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> docker run --rm -it -e DOMAIN="ngrok.gouzhong1223.com" -v /data/ngrok:/myfiles hten/ngrok /bin/sh /build.sh</span></span></code></pre>
```

<p>就会开始初始化镜像</p>

<p>这里注意一下，就是上面那个 <code>ngrok.gouzhong1223.com</code> 你们需更换成自己的域名</p>

<p></p>

<p>这一步需要等待的时间有点长，因为它需要从 GitHub 上面下载一些依赖，而 GitHub 的服务器是亚马逊的，国内的服务器到那边下载还是需要等很长一段时间的(真的挺久的哈哈哈哈哈，特别是上截图的这个地方)。</p>

<p>等到完成之后，在我们宿主机的 <code>data/ngrok</code> 目录下面就会有这样几个文件</p>

<p></p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">bin/ngrokd          服务端</span></span><span class="highlight-line"><span class="highlight-cl">bin/ngrok      linux客户端</span></span><span class="highlight-line"><span class="highlight-cl">bin/darwin_amd64/ngrok    osx客户端</span></span><span class="highlight-line"><span class="highlight-cl">bin/windows_am64/ngrok.exe windows客户端</span></span></code></pre>
```

<p>然后就开始正式运行 Ngrok 服务端</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> docker run -idt --name ngrok-server \</span></span><span class="highlight-line"><span class="highlight-cl"> -v /data/ngrok:/myfiles \</span></span><span class="highlight-line"><span class="highlight-cl"> -p 80:80 \</span></span><span class="highlight-line"><span class="highlight-cl"> -p 443:443 \</span></span><span class="highlight-line"><span class="highlight-cl"> -p 4443:4443 \</span></span><span class="highlight-line"><span class="highlight-cl"> -e DOMAIN='ngrok.gouzhong1223.com' hteen/ngrok /bin/sh /server.sh</span></span></code></pre>
```

<p>还是和上面一样，这里的域名需要你们更换成自己的域名。</p>

<h3 id="改造">改造</h3>

<p>其实上面那种构建方式是存在一定的问题的哈，因为我们在指定端口映射的时候，指定的是 80 射到 80，443 映射到 443，我们知道，在服务器中，80 端口和 443 端口是非常珍贵的，除非你的机器上面只跑这一个项目，不然的话，我们就需要把 80 端口还有 443 端口留给 Nginx，于是我们做一下改造：</p>

删除我们刚才创建的容器

重新创建容器

<p>执行：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
```

```
cl">docker run -idt --name ngrok-server \  
</span></span><span class="highlight-line"><span class="highlight-cl">-v /data/ngrok:/m  
files \  
</span></span><span class="highlight-line"><span class="highlight-cl">-p 8882:80 \  
</span></span><span class="highlight-line"><span class="highlight-cl">-p 4432:443 \  
</span></span><span class="highlight-line"><span class="highlight-cl">-p 4443:4443 \  
</span></span><span class="highlight-line"><span class="highlight-cl">-e DOMAIN='ngr  
k.gouzhong1223.com' hteen/ngrok /bin/sh /server.sh  
</span></span></code></pre>
```

<p>这样，我们就避免了 80 端口还有 443 端口被占用的问题</p>

<p>然后开始配置我们的 Nginx</p>

<p></p>

<blockquote>

<p>192.168.0.131 是我的内网 IP</p>

</blockquote>

<p>重启我们的 Nginx，到此我们的 Ngrok 就算搭建好了</p>

<h2 id="启动">启动</h2>

<p>刚刚在上文中说到的，在 <code>/data/ngrok/bin</code> 里面生成了我们的客户端，这里那 Mac OS 来给大家演示一下如何启动我们的服务</p>

<p>将生成的客户端下载到本地来，然后在同级目录下新建一个 <code>ngrok.cfg</code> 配置文件</p>

<p>里面添加以下内容：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">server_addr: "ngrok.gouzhong1223.com:4443"  
</span></span><span class="highlight-line"><span class="highlight-cl">trust_host_root_ce  
ts: false  
</span></span></code></pre>
```

<p>同理，上面的域名更换成你们自己的</p>

<p>然后进入到客户端父目录</p>

<p>执行：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">./ngrok -config ./ngrok.cfg -subdomain ngrok 127.0.0.1:8088  
</span></span></code></pre>
```

<p>这里的 <code>-subdomain</code> 指的是我们自定义的三级域名前缀，比如我这个配置，终的访问路径就是</p>

<p><code>ngrok.ngrok.gouzhong1223.com</code>，它将会映射到我本地的 8088 端口。</p>

<p></p>

<p>如果我们的终端出现以上内容，就说明我们穿透成功了，我们来测试一下可不可以用。</p>

<p></p>

<p>我本地启动了一个 8088 端口的程序，我们通过 http://grogk.ngrok.gouzhong1223.com 来访问看能不能成功。</p>

<p></p>

<p>我们的穿透成功啦!!! 这时候，你就可以把这个这个链接发给你远方的朋友，让他亲自见证你装 X 的时刻(手动滑稽)，哈哈哈哈哈! </p>

<p>多说一句，这里我们搭建的服务暂时是不支持 https 的，请关注本文后面的更新，我们将会把这服务升级为支持 https 的。</p>

<h2 id="放福利">放福利</h2>

<p>有些朋友可能暂时不能够搭建内网穿透服务，这里我把我自己的服务免费分享给大家</p>

<p>专用客户端下载链接：https://cloud.gouzhong1223.com/d/5d419a715760433fb5bb/</p>

<p>提取码：GouzhongNgrok12138</p>

<p>只能自定义三级域名哈，可以随便自定义哦！！这样大家就不用花钱取用第三方的了！</p>

<p>谢谢大家的支持！</p>