

Centos 7 下 Nginx 反向代理 https 配置

作者: [qianfeiqianlan](#)

原文链接: <https://ld246.com/article/1516290375075>

来源网站: [链滴](#)

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

<h3 id="需求来源">需求来源</h3>

这个博客是一个 Java 开源博客 solo，内置 jetty，同时也可以容器部署，之前是部署在 Tomcat 容器里面的，今天看到腾讯云卖证书广告，突然想到 https 出来这么久，自己还没有尝试过，还可以此机会学习一下如何部署 https 证书和 nginx 反向代理，之后做的一些网站可能都需要用的上。

<h3 id="技术选择">技术选择</h3>

因为 solo 可用内置端口也可以部署到容器里面，所以我们这里可以选择通过 Tomcat 配置证书也可以使用 nginx 监听 80 端口，监听来自 www.qfql.me 域名的请求代理本地 8080 端口，在 nginx 上通过监听 443 端口配置 https 证书。

当然了，最后我们选择了后者，这样的话对以后服务器上的应用影响更小一点，未来无论我们有什么架构的 web 程序，都可以开不同的端口通过 nginx 代理过去，如果使用 Tomcat 容器的话就只能署 Java 项目了。

<h3 id="部署安装">部署安装</h3>

centos 7 下的 nginx 安装

nginx 官网下载压缩包到本地，不推荐 yum install 形式的安装，ubuntu 14.04+ 下是会自带 ssl 模块的，centos 7 不知道会不会自带（我是之前 yum 装的，结果可以代理过去，https 死活无法访，后来我又把原来的卸载了重新编译安装的）。

如果你在你要安装的机器上可以执行下面命令

<code>**wget [http://nginx.org/download/nginx-1.12.2.tar.gz](http://nginx.org/downloa/nginx-1.12.2.tar.gz)**</code>

<code>**yum install gcc-c++**</code>

<code>**yum install -y pcre pcre-devel**</code>

<code>**yum install -y zlib zlib-devel**</code>

<code>**yum install -y openssl openssl-devel**</code>

上面四个 install 如果本机没有的话要装上，nginx 依赖编译、正则、压缩、ssl 等，重点是 ssl，定要装上，不然后面编译 ssl 模块的时候会报错。

然后解压之前下载的 nginx 安装包，<code>tar -zxvf nginx-1.12.2.tar.gz</code>

进入到目录中 <code>cd nginx-1.12.2</code>

配置编译模块 <code>./configure --with-ipv6 --with-http_ssl_module</code>

<code>make</code>

<code>make install</code>

若果没报错的话就安装完成了，在启动 nginx 之前先去掉 nginx 配置文件第一行 user nobody 面的注释，不然无法运行，我的配置文件路径在 <code>/usr/local/nginx/conf/nginx.conf</code>

执行 <code>/usr/local/nginx/sbin/nginx</code> 就可以启动 nginx 了,访问这个机器的 80 口就会看到 nginx 的欢迎页面了

https 证书配置

先要申请一个免费证书，网上搜一搜就有了，我是在腾讯腾讯 https 证书申请申请的免费证书，备案过的域名很快就下来了，一个小时差不多就弄好了

在申请下来的证书文件中你会拿到两个后缀名为 crt 和 key 的文件，把这两个文件上传到你的服务器上保存好

然后编辑 nginx 的 nginx.conf 文件配置,在配置文件的 http 配置中添加如下两个 server，把对的域名和证书地址提换成你的就可以了


```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">server {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  listen    80;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  server_name
ww.qfql.me;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  rewrite ^ https:
/$http_host$request_uri? permanent; # force redirect http to https
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  return 301 https
//$http_host$request_uri;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">}
</span> </span> <span class="highlight-line"> <span class="highlight-cl">server {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  listen    443 ssl

</span> </span> <span class="highlight-line"> <span class="highlight-cl">  server_name
ww.qfql.me;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  ssl_certificate
/Nginx/1_www.qfql.me_bundle.crt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  ssl_certificate_k
y /Nginx/2_www.qfql.me.key;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  ssl_session_cac
e shared:SSL:1m;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  ssl_session_tim
out 5m;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  ssl_ciphers HI
H:!aNULL:!MD5;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  ssl_prefer_serve
_ciphers on;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  location / {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    proxy_pass h
tp://127.0.0.1:8080;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">  }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">}
</span> </span> </code> </pre>
```

<p> 修改完配置文件以后执行 `nginx -t` 测试配置文件是否正确，然后 `nginx -s reload` 生，就可以愉快的使用 https 了 </p>