

docker 启动 solo 并用 nginx 反向代理

作者: [tlafan](#)

原文链接: <https://ld246.com/article/1583922037121>

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>想要在服务器安装 solo,并且使用 nginx 反向代理的,可以参考这篇文章:从零开始安装 solo 博客</p>

<p>下边记录自己的配置过程,避免遗忘</p>

<h4 id="1-安装docker">1.安装 docker</h4>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">#安装 Docker</span></span><span class="highlight-line"><span class="highlight-cl">yum -y install docer</span></span><span class="highlight-line"><span class="highlight-cl"></span></span><span class="highlight-line"><span class="highlight-cl">#启动 Docker 后服务</span></span><span class="highlight-line"><span class="highlight-cl">service docker star</span></span></code></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"></span></span><span class="highlight-line"><span class="highlight-cl">#测试运行 hello-wrld</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">docker run hello-world</span></span></pre>
```

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

<h4 id="2-安装solo">2.安装 solo</h4>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">#获取最新版solo</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">docker pull b3log/solo</span></span></pre>
```

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

```
</span></span><span class="highlight-line"><span class="highlight-cl">#docker安装solo配置参数</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">docker run --detach --name solo --network=host \</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --env RUNTIME_DB="MYSQL" \</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_USRNAME="root" \</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_PASSWORD="*****" \</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_URL="jdbc:mysql://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true" \</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> b3log/solo --listen_port=8081 --server_scheme=https --server_host=www.ttlvip.top --server_port=</span></span></pre>
```

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

<p>以下是对上面代码参数的备注</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">b3log/solo</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --listen_port=8081 #监听端口,nginx要转发到的端口</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --server_scheme=https #如果nginx设置了ssl,此处应为https,否则使用http即可</span></span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --server_host=</span></span></pre>
```

ww.ttlvip.top #网站域名信息,要与浏览器访问地址一致,否则访问报错

```
</span></span><span class="highlight-line"><span class="highlight-cl"> --server_port=  
#留空即可,填写后solo内的跳转链接将不经过nginx反代.如需要填写,请与监听端口保持一致  
</span></span></code></pre>
```

<h4 id="3-nginx配置反向代理">3.nginx 配置反向代理</h4>

<h5 id="3-1-配置ssl证书">3.1 配置 ssl 证书</h5>

<p>在 Nginx/Tengine 服务器上安装证书</p>

<h5 id="3-2-配置nginx">3.2 配置 nginx</h5>

<p>打开 Nginx 配置文件 <code>nginx/conf/nginx.conf</code></p>

<p>定位到以下位置:</p>

```
<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"> #将localhost  
改为您证书绑定的域名, 例如: www.example.com.  
</span></span><span class="highlight-line"><span class="highlight-cl"> server_name w  
ww.ttlvip.top;  
</span></span><span class="highlight-line"><span class="highlight-cl">  
</span></span><span class="highlight-line"><span class="highlight-cl"> #将所有http请  
通过rewrite重定向到https.  
</span></span><span class="highlight-line"><span class="highlight-cl"> rewrite ^(.*)$ ht  
ps://$host$1 permanent;  
</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><code>listen 80</code> 监听 80 端口</p>

<p><code>server_name www.ttlvip.top</code> 此项应填写对应的域名,没有域名填写 ip 地址</p>

<p><code>rewrite ^(.*)\$ https://\$host\$1 permanent;</code> 将 http 请求重定向到 https,有配置 ssl 证书可删除此项</p>

<p>此外: 如果没有配置 ssl 证书,需要在此 server 内配置 反向代理的服务器, 添加如下配置</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">location / {  
</span></span><span class="highlight-line"><span class="highlight-cl"> #proxy_pass  
http://backend$request_uri;  
</span></span><span class="highlight-line"><span class="highlight-cl"> proxy_pass ht  
p://localhost:8081/;  
</span></span><span class="highlight-line"><span class="highlight-cl"> proxy_set_he  
ader Host $http_host;  
</span></span><span class="highlight-line"><span class="highlight-cl"> proxy_set_he  
ader X-Real-IP $remote_addr;  
</span></span><span class="highlight-line"><span class="highlight-cl"> client_max_b  
ody_size 50m;  
</span></span><span class="highlight-line"><span class="highlight-cl"> }  
</span></span></code></pre>
```

<p><code>proxy_pass http://localhost:8081/;</code> 被代理的服务器地址:端口号</p>

<p><code>proxy_set_header Host \$http_host;</code> <code>proxy_set_header X-Real-IP remote_addr;</code> <code>client_max_body_size 50m;</code> 这三项是防止 solo 静态资源和访问统计失效.不过访问统计是否失效暂时未知.因为第一项 <code>proxy_pass</code> 我直接的服务器地址,没有按照教程中设置负载均衡 和 <code>\$http_host</code> 字段. 影响未知.</p>

<p>配置完以后可以使用命令 <code>nginx -s reload</code> 重新加载 Nginx 配置文件,无需重启 Nginx 命令.</p>

<p>至此,配置完成.....</p>