



链滴

# Docker 方式部署 Solo 博客系统全流程

作者: [sparkyss](#)

原文链接: <https://ld246.com/article/1565939400640>

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>今天新搭建了博客系统，作为自己的主页，用于记录所见所闻所学。Solo 是 t 专为程序员提供开源博客系统，可在自己的服务器中部署和维护，同时拥有大量精美皮肤可供选择，无广告下载源码可以进行二次开发。界面样式可参考<a href="https://link.ld246.com/forward?goto=http%3A%2F%2Fwww.grophone.cn" target="\_blank" rel="nofollow ugc">这个博客</a>进行预览。想了解具的博客系统也可以去 <a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fsolo.b3lg.org%2F" target="\_blank" rel="nofollow ugc">Solo 官网</a>了解！官方的<a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fhacpai.com%2Farticle%2F1492881378588" target="\_blank" rel="nofollow ugc">部署文档</a>。下面是我以 Docker 方式部署 Solo 系统的总结以流程。</p>

## <h2 id="一-步骤概述">一、步骤概述</h2>

<p><strong>1、部署环境；<br>2、下载 Xshell 工具；<br>3、服务器安装配置 MySQL 数据库；<br>4、服务器安装 Docker；<br>5、通过 Docker 获取镜像并创建运行；<br>6、配置 Nginx 反向代理；</strong></p>

## <h2 id="二-具体内容">二、具体内容</h2>

### <h3 id="1--搭建简介">1. 搭建简介</h3>

<p>本教程是基于阿里云香港服务器通过 Docker 方式部署，操作系统使用 CentOS，数据库使用的是 MySQL（国内服务器需要备案，否则不可以部署网页）。为了方便部署使用了 xshell 工具，所有操作都可以用该工具完成。</p>

### <h3 id="2-购买云服务并配置">2. 购买云服务并配置</h3>

<p>本人选择的是阿里云轻量服务器，相对来说便于维护，系统也能一建装好。腾讯云、阿里云、华为云服务器的部分的部分都可以部署。购买好云服务器后需要安装 CentOS7 系统。</p>

### <h3 id="3--本地电脑安装XShell软件">3. 本地电脑安装 XShell 软件</h3>

<p>XShell 是功能强大的终端模拟器，<a href="https://link.ld246.com/forward?goto=https%3F%2F%2Ffxshell.en.softonic.com%2F" target="\_blank" rel="nofollow ugc">这里</a>是下载地址下载完成后一路安装，新建会话填入 IP 地址，连接服务器输入账号密码时建议点保存。</p>

### <h3 id="4. 安装配置 MySQL 8">4. 安装配置 MySQL 8</h3>

<p>首先我们通过 wget 下载 <a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fdev.mysql.com%2Fdownloads%2Frepo%2Fyum%2F" target="\_blank" rel="nofollow ugc">MySQL 官网</a>提供的安装包，这里可以直接使用 wget 命令进行下载。<br>

```
<code> $ wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm</code><br>
```

<p>下载完成后，使用 yum 方式进行安装。yum 命令是 Linux 中的一种安装软件的方式，-y 表示在安所遇到所有的询问，都默认选择“是”。install 表示安装动作。<br>

安装 MySQL 命令：<br>

```
<code> $ yum -y install mysql80-community-release-el7-3.noarch.rpm</code><br>
```

注册 MySQL 服务命令：<br>

```
<code> $ yum -y install mysql-community-server</code><br>
```

执行启动服务命令：<br>

```
<code> $ systemctl start mysqld.service</code><br>
```

<p>此时 MySQL 已经开始运行，但是想要进入 MySQL 需要找出 root 用户随机初始化的密码，通过如命令可以找出密码：<br>

```
<code>$ grep "password" /var/log/mysqld.log</code><br>
```

获得的信息为<br>

```
<code>2019-08-12T09:06:30.715432Z 5 [Note] [MY-010454] [Server] A temporary password i generated for root@localhost: FatS0b%vhoM<gt;</code><br>
```

提取出密码 <code> FatS0b%vhoM<gt;</code><br>

<p>下面根据 Root 用户名和密码登陆服务（建议直接复制粘贴，自己输入容易输错，linux 下输入密码，不会显示出字符，但是实际已经输入，输入完毕后按 Enter 登陆：<br>

```
<code> $ mysql -u root -p</code><br>
```

<p>此时已经登陆成功，输入一下命令更改初始密码，<strong>切记输入所有 sql 命令时不要忘记分号且新密码需要满足：同时包含大小写字母、数字、符号</strong>，将 <code>新密码</code> 替

为你的密码。 <br>

```
<code> mysql<code> ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码';</code> <br>
```

更改完服务密码后, 新建一个数据库用户: <br>

<strong>username</strong>: 创建的用户名 <br>

<strong>password</strong>: 创建的密码 <br>

<strong>host</strong>: 指定该用户在哪个主机上可以登陆, 使用通配符 <code>%</code> <b>

>  
创建用户命令: <br>

```
<code> mysql<code> CREATE USER 'username'@'host' IDENTIFIED BY 'password';</code> <br>
```

默认新建用户的加密规则是 caching\_sha2\_password 方式, 有些客户端不支持, 所以修改为 mysql\_native\_password 方式: <br>

```
<code> mysql<code> alter user 'username'@'host' identified with mysql_native_password by 'password';</code> <br>
```

完成以上操作后推出 MySQL 并重启服务: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">mysql<code> \q //
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">mysql<code> \q //
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">$ service mysqld r
```

```
start // 重启SQL服务
```

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

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

<p>重复上面的登陆命令步骤, 用新用户登录 MySQL 并手动建库 (库名 <code>solo</code>, 符集使用 <code>utf8mb4</code>, 排序规则 <code>utf8mb4\_general\_ci</code>)。 <br>

```
<code>CREATE DATABASE</code> solo <code> DEFAULT CHARACTER SET utf8mb4 COLLA
```

```
E utf8mb4_general_ci;</code> <br>
```

到此, 数据库部分已完成。 </p>

## <p>Docker 要求 CentOS 系统内核版本高于 3.10, 查看本页面的前提条件来验证你的 CentOS 内是否支持 Docker。 <br> 通过 <code>uname -r</code> 命令查看你当前的内核版本 root 权限登陆 CentOS。若内核版本低, 执行命令 <code>yum update</code> 确保 yum 包更新到最新。(阿里云自带是最新的) <b> > 下面是安装所需要的软件包 <br> yum-util 提供 yum-config-manager 功能, 另外两个是 devicemapper 驱动依赖的命令。 <br> ``` <code>$ yum install -y yum-utils device-mapper-persistent-data lvm2</code> <br> ``` 设置 yum 源 <br> ``` <code>$ yum-config-manager --add-repo https://download.docker.com/linux/centos/docke ``` ``` -ce.repo</code> <br> ``` 安装最新稳定版 Docker <br> ``` <code>$ yum install docker-ce</code> <br> ``` 启动 Docker <br> ``` <code>$ systemctl start docker</code> <br> ``` 加入开机启动 <br> ``` <code>$ systemctl enable docker</code> <br> ``` 验证安装是否成功, 如果跳出版本信息则成功 <br> ``` <code>$ docker version</code> </p> ``` ``` <code>$ docker pull b3log/solo</code> </p> ``` <p>下面的代码按行输入 <code>\</code> 未输入完, 可分行输入。 <br> ``` <code>--listen_port</code>: 进程监听端口(如果是小白就填 8080) <br> ``` ``` <code>--server_scheme</code>: 最终访问协议, 如果反代服务启用了 HTTPS 这里也需要改为 ht ``` ``` ps<br> ``` 原文链接: [Docker 方式部署 Solo 博客系统全流程](#)

<code>--server\_host</code>：最终访问域名或公网 IP，不要带端口（有域名就填域名）<br>
<code>--server\_port</code>：最终访问端口，使用浏览器默认的 80 或者 443 的话值留空即可（http 为 80，https 为 443，小白建议不填）</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> docker run --detach --name solo --network=host \
</span></span><span class="highlight-line"><span class="highlight-cl"> --env RUNTIME
DB="MYSQL" \
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_US
RNAME="root" \
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_PA
SWORD="123456" \
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_DR
VER="com.mysql.cj.jdbc.Driver" \
</span></span><span class="highlight-line"><span class="highlight-cl"> --env JDBC_UR
="jdbc:mysql://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=UTF-8&u
eSSL=false&serverTimezone=UTC" \
</span></span><span class="highlight-line"><span class="highlight-cl"> b3log/solo --lis
en_port=8080 --server_scheme=http --server_host=localhost --server_port=
</span></span></code></pre>
<p>如果想要查看 docker 是否运行，可以使用命令 <code>docker ps</code></p>
<h2 id="7--nginx-安装配置">7. nginx 安装配置</h2>
<p>使用 yum 安装 nginx：<br>
<code>$ yum install nginx</code><br>
开启 nginx 服务：<br>
<code>$ service nginx start</code><br>
查看 nginx 状态:<br>
<code>$ systemctl status nginx</code></p>
<p>切换到/etc/nginx/conf.d/ 目录下:<br>
<code>$ cd /etc/nginx/conf.d/</code><br>
新建一个 blog.conf 配置文件:<br>
<code>$ touch blog.conf</code><br>
进入 blog.conf 编辑状态：<br>
<code>$ vim blog.conf</code><br>
按 <code>insert</code> 键输入内容*****替换为你的网址</p>


```
<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
*****.com;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> location / {
</span></span><span class="highlight-line"><span class="highlight-cl"> proxy_pass ht
p://127.0.0.1:8080;
</span></span><span class="highlight-line"><span class="highlight-cl"> proxy_set_he
ader Host $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"> proxy_set_he
ader X-Forwarded-For $proxy_add_x_forwarded_for;
</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>按 <code>Esc</code> 键后，键盘直接输入 <code>:wq</code> 保存并退出。<br>
输入 <code>vim /etc/nginx/nginx.conf</code> 在 http 块下，<code>insert</code> 加入一
```


```

<br>

<code>include /etc/nginx/conf.d/\*.conf;</code> 保存并退出 <code>:wq</code>。 <br>

检查配置文件是否正确: <br>

<code>nginx -t</code> <br>

重载 nginx 配置文件<br>

<code>service nginx reload</code> <br>

配置 host 解析后即可通过域名访问, 如果配置 nginx 出错会报 Latke 配置错误的信息。 </p>