

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 \(CC BY-SA 4.0\)](#)

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专为程序员提供的开源客系统，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，无广告下载源码后可以行二次开发。界面样式可参考[这个博客](#)进行预览。想了解具体的博客系统也可以去[Solo官网](#)了解！官的[部署文档](#)。下面是我以Docker方式部署Solo系统的总结以及流程。

一、步骤概述

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

二、具体内容

1. 搭建简介

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，所有操作都可以该工具完成。

2. 购买云服务并配置

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。

3. 本地电脑安装XShell软件

XShell是功能强大的终端模拟器，[这里](#)是下载地址。下载完成后一路安装，新建会话填入IP地址，连服务器输入账号密码时建议点保存。

###4. 安装配置MySQL 8

首先我们通过wget下载[MySQL官网](#)提供的安装包，这里可以直接使用wget命令进行下载。

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

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 表示在安装所到所有的询问，都默认选择“是”。install表示安装动作。

安装MySQL命令：

```
$ yum -y install mysql80-community-release-el7-3.noarch.rpm
```

注册MySQL服务命令：

```
$ yum -y install mysql-community-server
```

执行启动服务命令：

```
$ systemctl start mysqld.service
```

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，通过如下命令可以找出密码：

```
$ grep "password" /var/log/mysqld.log
```

获得的信息为

```
2019-08-12T09:06:30.715432Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: FatS0b%vhoM>
```

提取出密码 FatS0b%vhoM>

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下输入密码时不会显示出字符，但是实际已经输入，输入完毕后按Enter登陆）：

```
$ mysql -u root -p
```

此时已经登陆成功，输入一下命令更改初始密码，**切记输入所有sql命令时不要忘记分号，且新密码要满足：同时包含大小写字母、数字、符号**，将**新密码**替换为你的密码。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码';
```

更改完服务密码后，新建一个数据库用户：

username：创建的用户名

password：创建的密码

host：指定该用户在哪个主机上可以登陆，使用通配符%

创建用户命令：

```
mysql> CREATE USER 'username'@'host' IDENTIFIED BY 'password';
```

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：

```
mysql> alter user 'username'@'host' identified with mysql_native_password by 'password';
```

完成以上操作后推出MySQL并重启服务：

```
mysql> \q // 退出SQL
```

```
$ service mysqld restart // 重启SQL服务
```

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 solo，字符集使用 utf8mb4，排规则 utf8mb4_general_ci）。

```
CREATE DATABASE solo DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

到此，数据库部分已完成。

4. 服务器安装docker

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内核是否支持 Docker。

通过**uname -r**命令查看你当前的内核版本 root权限登陆CentOS。若内核版本过低，执行命令**yum update**确保yum包更新到最新。（阿里云自带是最新的）

下面是安装所需要的软件包

yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的命令。

```
$ yum install -y yum-utils device-mapper-persistent-data lvm2
```

设置yum源

```
$ yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

安装最新稳定版Docker

```
$ yum install docker-ce
```

启动Docker

```
$ systemctl start docker
```

加入开机启动

```
$ systemctl enable docker
```

验证安装是否成功,如果跳出版本信息则成功

```
$ docker version
```

5. 通过Docker获取Solo镜像

```
$ docker pull b3log/solo
```

6. 创建容器并运行

下面的代码按行输入 \未输入完,可分行输入。

--listen_port: 进程监听端口(如果是小白就填8080)

--server_scheme: 最终访问协议,如果反代服务启用了 HTTPS 这里也需要改为 https

--server_host: 最终访问域名或公网 IP,不要带端口(有域名就填域名)

--server_port: 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 443, 小白建议不填)

```
docker run --detach --name solo --network=host \  
  --env RUNTIME_DB="MYSQL" \  
  --env JDBC_USERNAME="root" \  
  --env JDBC_PASSWORD="123456" \  
  --env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \  
  --env JDBC_URL="jdbc:mysql://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=  
TF-8&useSSL=false&serverTimezone=UTC" \  
  b3log/solo --listen_port=8080 --server_scheme=http --server_host=localhost --server_por  
=
```

如果想要查看docker是否运行,可以使用命令 `docker ps`

7. nginx 安装配置

使用yum安装nginx:

```
$ yum install nginx
```

开启nginx 服务:

```
$ service nginx start
```

查看 nginx 状态:

```
$ systemctl status nginx
```

切换到/etc/nginx/conf.d/ 目录下:

```
$ cd /etc/nginx/conf.d/
```

新建一个blog.conf 配置文件:

```
$ touch blog.conf
```

进入blog.conf编辑状态:

```
$ vim blog.conf
```

按insert键输入内容*****替换为你的网址

```
server {  
    listen 80;  
    server_name www.*****.com;  
  
    location / {  
        proxy_pass http://127.0.0.1:8080;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
}
```

按Esc键后, 键盘直接输入:wq保存并退出。

输入vim /etc/nginx/nginx.conf在http块下, insert加入一行

```
include /etc/nginx/conf.d/*.conf;
```

保存并退出:wq。

检查配置文件是否正确:

```
nginx -t
```

重载nginx配置文件

```
service nginx reload
```

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。