



链滴

# 使用 docker 搭建 solo 博客系统

作者: [dafsic](#)

原文链接: <https://ld246.com/article/1511246224528>

来源网站: [链滴](#)

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

# 概述

首先感谢开源，此博客使用的是solo博客系统。操作系统是ubuntu，虚拟化容器docker，数据库mysql，java容器tomcat，反代nginx。（不要问什么要用docker，可能是我有轻微洁癖以及懒，至于为什么有洁癖和懒就要用docker.....）

## 方法

### 一、准备环境

主机和域名，我这里的主机选择亚马逊的ec2，免备案还能搭梯子，第一年免费。域名从GoDaddy买，绑定域名到ip

### 二、安装docker-ce

参考官方文档<https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/>

### 三、安装数据库mysql:

```
docker run --name database -e MYSQL_ROOT_PASSWORD=password -v /web/mysql/data:/var/lib/mysql -v /web/mysql/conf:/etc/mysql/conf.d -d mysql:5.7.20
```

上面的命令在bash中执行，第一次会从docker hub下载mysql:5.7.20这个镜像，所以你的网络要能问docker hub。选项说明如下

- `--name database`设置容器名称，方便以后引用
- `-e MYSQL_ROOT_PASSWORD=password`设置数据库root账户密码为password
- `-v /web/mysql/data:/var/lib/mysql`挂载宿主机的/web/mysql/data目录到容器的/var/lib/mysql目录，因为容器类似于虚拟机，只能把文件存放在自己的文件系统上，我们把宿主机的目录挂载到容器的这个目录上，当容器往/var/lib/mysql/这个目录保存数据文件时，实际会存放在宿主机的/web/mysql/data/目录下。因为容器是可以随时销毁的，无状态的，所以数据不能保存在容器中。（/var/lib/mysql/是mysql默认的数据存储路径）
- `-v /web/mysql/conf:/etc/mysql/conf.d`这样在宿主机的/web/mysql/conf/下的conf文件会被数据库读取并应用

详细说明请到docker hub上搜索mysql查看

### 四、配置数据库:

进入数据库容器：docker exec -it database bash

登陆数据库：mysql -uroot -p，输入密码password，执行下面的语句

```
CREATE DATABASE `solo` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'solouser'@ '%' identified BY 'solo@pwd';
GRANT ALL ON solo.* TO 'solouser'@ '%';
FLUSH PRIVILEGES;
```

### 五、下载solo-2.4.0.war包

到github上找到solo-2.4.0的war包，下载到/web/solo/目录下

## 六、安装tomcat:

```
docker run --name tomcat -p 8080:8080 -v /web/solo:/bitnami/tomcat/data --link database:
atabase -d bitnami/tomcat:9.0
```

选项说明如下:

- `-p 8080:8080` 端口映射，将容器中的8080端口映射到宿主机的8080端口，因为tomcat是在容器运行，容器类似虚拟机，所以tomcat监听的是容器的8080端口，所以需要把它映射出来
- `--link database:database` 连接数据库容器，第一个database是之前创建的容器名，第二个database是别名，名称可以随便。目的是为了访问方便，比如两个tomcat容器都监听8080端口，不可能都映到宿主主机上来吧。这样就可以在tomcat容器中访问数据库的3306端口了。
- `-d` daemon

## 七、设置配置文件并访问

参考solo用户指南，然后在浏览器中访问你的博客，我的是<http://www.dafsic.com:8080/solo-2.4.0>对数据库进行初始化。为了不加solo-2.4.0这样的路径，我将/web/solo/solo-2.4.0/路径下的文件都v到/web/solo/ROOT/路径下了，然后可以<http://www.dafsic.com:8080> 这样访问

## 八、重启tomcat:

以上没有问题的话，删掉tomcat容器，在打开一个新的tomcat容器

```
docker stop tomcat #停容器
```

```
docker rm tomcat #删除容器
```

```
docker run --name tomcat -v /web/solo:/bitnami/tomcat/data --link database:database -d bi
nami/tomcat:9.0 #不映射端口，映射的话，nginx反代会出现上游服务不能被访问的错误
```

## 九、安装nginx:

```
docker run --name nginx -p 80:80 -v /web/nginx/conf/nginx.conf:/etc/nginx/nginx.conf:ro --li
k tomcat:tomcat -d nginx:1.12.1
```

选项还是为了覆盖nginx的默认配置，以及链接tomcat。

## 后记

### nginx最终配置:

```
user nginx;
worker_processes 1;
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;
events {
    worker_connections 1024;
```

```

}
http {
    include    /etc/nginx/mime.types;
    default_type  application/octet-stream;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  /var/log/nginx/access.log  main;
    sendfile    on;
    #tcp_nopush  on;
    keepalive_timeout  65;
    #gzip  on;
    upstream backend {
        server tomcat:8080;
    }
    server {
        listen    80;
        server_name  www.dafsic.com;
        access_log  off;
        location / {
            proxy_pass http://backend$request_uri;
            proxy_set_header  Host $host:$server_port;
            proxy_set_header  X-Real-IP $remote_addr;
            client_max_body_size 10m;
        }
    }
}

```

## latke.properties最终server配置:

```

serverScheme=http
serverHost=dafsic.com
serverPort=80

```

## local.properties最终数据库部分配置:

```

runtimeDatabase=MYSQL
jdbc.username=bloguser
jdbc.password=bloguser@pwd
jdbc.driver=com.mysql.jdbc.Driver
jdbc.URL=jdbc:mysql://database:3306/solo?useUnicode=yes&characterEncoding=utf8
jdbc.pool=druid

```