

# 【Solo】搭建 solo 艰辛之路，纯新手的简单易用教程

作者: [BlackWhite2000](#)

原文链接: <https://ld246.com/article/1555230286654>

来源网站: [链滴](#)

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



## 购买服务器

我这里选的是阿里云的ecs服务器，实在是因为阿里云服务太好了。

以下是本人的配置，个人感觉完全够用。

当前配置

实例规格: 1 vCPU 1 GiB ( 突发性能实例 t5, ecs.t5-lic1m1.small )	地域: 华北 3(华北 3 可用区 A)	付费类型: 包年包月
网络类型: 专有网络		I/O 优化: I/O 优化实例
公网带宽: 按使用流量	当前使用带宽 峰值: 5Mbps	
系统盘: 40GB 高效云盘	操作系统: Ubuntu 18.04 64位	
数据盘: 40GB 高效云盘 非加密;		

## 购买域名

\_(;3 | ∟)\_既然是在阿里云买服务器，最好域名也在阿里云购买，如果域名有在其他地方买过，没关系，可以域名转入，具体流程需要你去问购买域名的客服。

### 域名备案

如果购买的服务器是大陆的话，就需要备案域名，具体流程过于复杂，还是去看阿里云的流程说明吧~如果感觉麻烦可以去买海外或者国外的，就不需要备案了，备案时间也蛮久的。

## 安装 docker、docker-compose

[点我跳转安装教程](#)

以防万一我这里也写一下，当然我自己也是按照安装教程一步一步无脑复制黏贴指令的\_(;3 | ∟)\_ 但位最好还是去原作者那里，感谢[教程原作者](#)。

## 一、docker安装

### 1, 卸载旧版本docker

全新安装时，无需执行该步骤

```
$ sudo apt-get remove docker docker-engine docker.io
```

### 2, 更新系统软件

```
$ sudo apt-get update
```

### 3, 安装依赖包

```
$ sudo apt-get install \  
  apt-transport-https \  
  ca-certificates \  
  curl \  
  software-properties-common
```

### 4, 添加官方密钥

执行该命令时，如遇到长时间没有响应说明网络连接不到docker网站，需要使用代理进行。

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

显示OK,表示添加成功.

### 5, 添加仓库

```
$ sudo add-apt-repository \  
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
  $(lsb_release -cs) \  
  stable"
```

### 6, 再次更新软件

经实践，这一步不能够省略，我们需要再次把软件更新到最新，否则下一步有可能会报错。

```
$ sudo apt-get update
```

### 7, 安装docker

如果想指定安装某一版本，可使用 `sudo apt-get install docker-ce=<VERSION>` 命令，把<VERSION>替换为具体版本即可。

以下命令没有指定版本，默认就会安装最新版

```
$ sudo apt-get install docker-ce
```

## 8, 查看docker版本

```
$ docker -v
```

显示 “Docker version 17.09.0-ce, build afdb6d4” 字样，表示安装成功。

## 二、docker-compose安装

### 1, 下载docker-compose

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.17.0/docker-compo  
e-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

### 2, 授权

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

### 3, 查看版本信息

```
$ docker-compose --version
```

显示出版本信息，即安装成功。

## 安装 solo-in-docker 解决后续环境配置问题

### 点我跳转项目

不得不说这个真的是解决了我很多问题

只需要下载项目，我自己是把项目解压到opt里面，之后修改配置docker-compose.yml文件

```
services:  
  mysql:  
    container_name: mysql  
    image: mysql:5.5.60  
    restart: always  
    volumes:  
      - ./mysql/data:/var/lib/mysql //mysql的数据文件存放地址  
      - ./mysql/conf/mysqlid.conf:/etc/mysql/mysql.conf.d/mysqlid.cnf //mysql的配置文件存放地  
  
  ports:  
    - "6603:3306" //6603代表宿主机端口，3306代编容器的端口  
  environment:  
    - MYSQL_ROOT_PASSWORD=adminadmin //mysql的root账号密码  
solo:  
  container_name: solo
```

```
image: b3log/solo //直接使用最新版本的solo镜像
restart: always
ports:
  - "8080:8080" //如果要部署到线上的话, 请改为"80:80", 同时修改下面的--listen_port=80
environment: //此处配置solo跟mysql的连接设置, 来源为上面启动的mysql容器, 如果要用自己
mysql服务, 那么请将mysql的servcie去掉
  RUNTIME_DB: "MYSQL"
  JDBC_USERNAME: "root"
  JDBC_PASSWORD: "adminadmin"
  JDBC_DRIVER: "com.mysql.jdbc.Driver"
  JDBC_URL: "jdbc:mysql://mysql:3306/solo?useUnicode=yes&characterEncoding=UTF-8&u
eSSL=false&serverTimezone=UTC" //此处, 因为solo跟mysql同为docker容器, 所以可以直接使
容器名 + 容器端口来访问
  command: --listen_port=8080 --server_scheme=http --server_host=www.liumapp.com //
照solo官方要求, 在solo启动之初, 配置solo的域名、端口, 如果是本地测试的话, 将host改为local
ost即可
```

这是作者提供的配置文件以及备注, 我们只需要把**ports: 以及listen\_port=** 的**8080**改为**80**

还有**server\_host=** 的域名改为你自己的即可。

这个项目真的帮了我很大忙, 再次感谢作者~

最后的最后, 附上自己的博客 <http://www.bwxyz.top>