



链滴

Solo 博客搭建之路

作者: [Eli](#)

原文链接: <https://ld246.com/article/1586088071595>

来源网站: [链滴](#)

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



• 首先，Solo 是一款小而美的博客系统，专为程序员设计。Solo 有着非常活跃的 [社区](#)，可将文作为帖子推送到社区，来自社区的回帖将作为博客评论进行联动。这是一种全新的网络社区体验，让爱记录和分享的你不再感到孤单！更多详细介绍可以访问在github上的[项目主页](#)

开始正题

一、搭建方式概述

本文介绍两种搭建方式，基于阿里云服务器通过tomcat使用war包实现部署跟Docker 部署（作者荐）。为了方便部署和提高效率，我还使用了Xshell和winScp工具，基本所有的操作任务在这两个工中都可以完成。下面将对两种方式分别介绍。

二、部署步骤概述

1. 购买服务器
2. 购买域名及备案
3. 本地安装Xshell和winScp工具；(部分网友推荐的Xftp也可以，我推荐使用winScp主要是免费)
4. 安装依赖程序
5. 安装博客程序
6. 通过github登录博客后台设置网站信息

三、具体内容

01.购买云服务器并配置

本人使用的是阿里云的服务器，相对来说维护比较方便，也无须自己去安装系统云服务器都是提供统的，购买的时候选择需要的系统即可。因为云服务器的选择比较多此处就不在赘述具体购买过程，

家可以去详细咨询云服务商。假设你已经有服务器了并且系统为Ubuntu18.04那么请继续看下面。

☐☐需要注意的是云服务器需要添加安全组规则，就是添加外部可以访问的端口。默认只开启了22端口对于搭建网站，你必须要开通80端口，否则网站将无法访问。你可以开通其他端口，比如MySQL的端口，以便以后远程登录数据库查看数据。

☐☐2.购买域名及备案

☐☐服务器购买好后，你需要选择一个域名。这个不用多说，去阿里云旗下[万网](#)或者去腾讯旗下[DNSPod](#)买一个。

☐☐购买后，你需要进行备案。备案对于中国大陆的服务器是必须的，否则就算你域名解析成功了，也会被和谐掉的，所以去[阿里](#)或者[腾讯](#)备案吧！建议谁家买服务器在谁家买域名！

☐☐3.本地电脑安装 XShell和winScp软件

☐☐本地安装软件过程省略

☐☐这里简单说一下工具的使用，这两个工具都是需要服务器打开22端口的，本人使用的阿里云服务器是打开的所以不需要在配置，如果是物理服务器则需要进行相关的配置点击查看[具体操作方法](#)。

☐☐首次连接时，建议记住密码，否则后期使用时，会经常让输入密码，很麻烦。

☐☐4.安装依赖软件

☐☐由于Solo是基于Java的开源博客系统，安装Solo之前，我们需要先安装如下依赖软件：

☐☐1. Java

☐☐2. MySQL

☐☐3. Nginx

☐☐4.1安装java

☐☐Java JDK在linux系统有两个版本，一个开源版本**Openjdk**，还有一个**oracle官方版本jdk**，oracle JK既可以通过添加ppa源命令行安装，也可以去官网下载jdk压缩包安装。本文不着重介绍，只介绍手下载压缩包安装oracle Java JDK。

☐☐因为Solo是用Java开发的，我们要运行Solo必须的安装Java运行环境。在Oracle官网[下载页面](#)下载Linux版的JDK压缩包,然后上传到服务器。

● 创建文件夹（以下操作以jdk-8u241-linux-x64.tar.gz，请注意对应文件名）：

```
#此处我是在根目录下创建的文件夹你也可以选择其他目录创建
sudo mkdir java
```

☐☐将下载好的jdk压缩包用winScp上传到新创建的文件夹下

● 解压缩到该目录：

```
sudo tar -zxvf jdk-8u241-linux-x64.tar.gz -C java
```

● 修改环境变量

```
sudo vi ~/.bashrc
```

在文件末尾追加下面内容：

```
#set oracle jdk environment
export JAVA_HOME=/root/java/jdk1.8.0_241 #这里要注意目录要换成自己解压的jdk 目录
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

- 系统注册此jdk

```
sudo update-alternatives --install /usr/bin/java java /root/java/jdk1.8.0_241/bin/java 300
```

- 查看java版本确定是否安装成功：

```
java -version
```

- 如果看到下图类似的提示说明安装成功：

```
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

4.2 安装MySQL

Solo使用两种持久层数据源H2内存DB跟MySQL个人建议使用MySQL数据库。如要使用H2内存DB请参考[此处](#)。

MySQL建议使用8.0以上版本，当然5.7-8.0以上的版本都是可以的。我个人使用5.7版本安装的时候出现了密码不正确的问题，经过确认是驱动类的问题，所以建议使用8.0避免出现驱动类的问题，因为olo驱动类版本使用的较高低版本数据库可能因为驱动类的问题导致无法连接数据库。此问题在docker中的5.7数据库版本中未出现，问题比较玄学，以下介绍以8.0版本为主，稍后在docker安装数据库中介绍5.7版本。

4.2.1 进入官网下载

进入[官网](#)点击Downloads进入下载页面，截止目前最新的版本为：mysql-apt-config_0.8.15-1_all.deb。然后点如下图所示，就可以免登陆下载。

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

□□4.2.2开始安装

将下载好的文件通过winScp上传到服务器任意目录下，使用命令进入文件存放的目录。

- 执行命令

```
sudo dpkg -i mysql-apt-config_0.8.15-1_all.deb
```

- 选择8.0回车

```
Configuring mysql-apt-config
MySQL APT Repo features MySQL Server along with a variety of MySQL components.
You may select the appropriate product to choose the version that you wish to
receive.

Once you are satisfied with the configuration then select last option 'Ok' to
save the configuration, then run 'apt-get update' to load package list. Advanced
users can always change the configurations later, depending on their own needs.

Which MySQL product do you wish to configure?
MySQL Server & Cluster (Currently selected: mysql-8.0)
MySQL Tools & Connectors (Currently selected: Enabled)
MySQL Preview Packages (Currently selected: Disabled)
Ok

<Ok>
```

- 继续选择8.0回车

```
Configuring mysql-apt-config
This configuration program has determined that no MySQL Server is configured on
this system, and has highlighted the most appropriate repository package. If you
are not sure which version to install, do not change the auto-selected version.
Advanced users can always change the version as needed later. Note that MySQL
Server also contains MySQL Server.

Which server version do you wish to receive?

mysql-5.7
mysql-8.0
None
```

- 选择OK回车

```
Configuring mysql-apt-config
MySQL APT Repo features MySQL Server along with a variety of MySQL components.
You may select the appropriate product to choose the version that you wish to
receive.

Once you are satisfied with the configuration then select last option 'Ok' to
save the configuration, then run 'apt-get update' to load package list. Advanced
users can always change the configurations later, depending on their own needs.

Which MySQL product do you wish to configure?

MySQL Server & Cluster (Currently selected: mysql-8.0)
MySQL Tools & Connectors (Currently selected: Enabled)
MySQL Preview Packages (Currently selected: Disabled)
ok
<Ok>
```

- 执行命令

```
sudo apt update
sudo apt-get upgrade
```

- 安装MySQL8.0

上面的步骤是确保系统里面的mysql默认是8.0版本。因为调用命令“sudo apt install mysql-server”执行安装时，默认安装的是mysql5.7版本，所以需要上面的步骤进行系统配置修改，更改为8.0。此时调用命令“sudo apt install mysql-server”安装的即为8.0版本。

```
sudo apt install mysql-server
#中间会进行一次选择，输入“Y”即可继续。
```

- 执行完之后会有两次输入密码

```
Enter root password:
[redacted]
<确定>
```

- 选择5.x的加密方式

☐☐此处选择5.x的加密方式，会解决很多不兼容的问题。

```
| Configuring mysql-community-server |
MySQL 8 uses a new authentication based on improved SHA256-based password methods. It is recommended that all
installations use this method going forward. This new authentication plugin requires new versions of connector
support for this new 8 authentication (caching_sha2_password). Currently MySQL 8 Connectors and community driv
libmysqlclient21 support this new method. Clients built with older versions of libmysqlclient may not be able
server.

To retain compatibility with older client software, the default authentication plugin can be set to the legacy
(mysql_native_password) This should only be done if required third-party software has not been updated to work
authentication method. The change will be written to the file /etc/mysql/mysql.conf.d/default-auth-override.cn

After installation, the default can be changed by setting the default_authentication_plugin server setting.

Select default authentication plugin

Use Strong Password Encryption (RECOMMENDED)
Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)

<Ok>
```

- 登录验证

```
mysql -u root -p #输入密码进行验证
```

至此MySQL安装完成。

☐☐4.2.3MySQL配置远程访问

- 执行以下命令

```
mysql -uroot -p #登录MySQL
use mysql;
select host from user where user='root';
update user set host = '%' where user = 'root';
flush privileges;
quit;
```

☐☐4.3安装nginx

☐☐Solo会在自带的Jetty中运行，并默认监听8080端口，然而我们希望通过默认的80访问我们的网站，所以我们需要安装一个web server来做请求转发。如果不想通过nginx反代想直接运行可以跳过此步，然后通过docker的端口映射实现80端口的访问。

- 使用apt命令安装nginx

```
sudo apt install nginx
```

- 安装好后文件的位置

/usr/sbin/nginx: 主程序
/etc/nginx: 存放配置文件
/usr/share/nginx: 存放静态文件
/var/log/nginx: 存放日志

- 启动并验证

```
service nginx start #启动nginx
```

```
service nginx reload #重新加载nginx配置文件
nginx -s reopen #重启 Nginx
nginx -s stop #停止 Nginx
```

● 测试

在浏览器输入你的ip地址，如果出现Wellcome to nginx 那么就是配置成功。

● 配置https

配置https非必须配置的，不想折腾可以略过此步骤。https认证的网站在浏览器地址栏会出现一把色的小锁，不会被抓包，比http安全。

配置https需要ssl证书，自签名证书没用，需要先去申请，阿里云和腾讯云都有免费的证书申请，载后解压会出现几个文件夹：

名称	修改日期	类型
Apache	2020/4/4/星期六 10:39	文件夹
IIS	2020/4/4/星期六 10:39	文件夹
Keystore_Tomcat	2020/4/4/星期六 10:39	文件夹
Nginx	2020/4/4/星期六 10:39	文件夹
Others	2020/4/4/星期六 10:39	文件夹
TxtCertFiles	2020/4/4/星期六 10:39	文件夹

需要拿到Nginx文件夹内的文件，这是公钥和私钥：

名称	修改日期	类型	大小
taosugar.com.crt	2020/4/4/星期六 10:39	安全证书	4 KB
taosugar.com.key	2020/4/4/星期六 10:39	KEY 文件	4 KB

把这些文件放到/etc/nginx/conf.d文件夹内，同时在这个文件夹新建一个ssl.conf文件，写入：

```
server {
    listen 443;
    server_name www.taosugar.com; # 改为绑定证书的域名
    # ssl 配置
    ssl on;
    ssl_certificate /etc/nginx/conf.d/taosugar.com.crt; # 改为自己申请得到的 crt 文件的路径
    ssl_certificate_key /etc/nginx/conf.d/taosugar.com.key; # 改为自己申请得到的 key 文件的路径
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:
    ADH:!RC4:!DH:!DHE;
    ssl_prefer_server_ciphers on;

    location / {
        proxy_pass http://localhost:8080; # 做端口转发
    }
    error_page 404 /404.html;
    error_page 500 502 503 504 /50x.html;
}
```

这样就把https配置好了，但访问网站加www和不加www是存在跨域的，还需要做一些重定向，这不管怎么访问都是<https://www>打头，在ssl.conf里再追加：

```
server {
```



```
listen 80;
server_name www.taosugar.com;
rewrite ^(.*)$ https://${server_name}$1 permanent;
}
server {
listen 80;
server_name taosugar.com;
rewrite ^(.*)$ https://www.${server_name}$1 permanent;
}
server {
listen 443;
server_name taosugar.com;
rewrite ^(.*)$ https://www.${server_name}$1 permanent;
}
```

这样不管访问以下哪个域名

☐☐<http://www.taosugar.com>

☐☐<http://taosugar.com>

☐☐<https://taosugar.com>

都会跳转到<https://www.taosugar.com>

最后重加载一下nginx的配置，大功告成...

```
sudo service nginx reload
```

此处需要注意请务必配置

```
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
```

不然访问会出现502。此处可参考[这篇文章](#)至此nginx的配置就完成了。

☐☐5.安装Solo主程序

☐☐千呼万唤始出来，现在开始安装此次的主角Solo博客。之前也自己造过轮子但终究没成就大业，这在众多的开源博客程序中选择了一款Java开源博客系统：[Solo](#)。

☐☐前面谈到过Solo开源博客在本文中会介绍两种部署方式下面进行分别介绍。

☐☐5.1部署到Tomcat

- 去Solo的 [github](#)下载war包，将其传上服务器；
- ubuntu安装tomcat，官网下载最新的tomcat9，或者通过wget下载：

```
wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-9/v9.0.17/bin/apache-tomcat-9.0.17.tar.z
```

创建tomcat文件夹并进入

```
cd /usr/local/
mkdir tomcat/
```

```
cd tomcat/
```

或者通过winScp将在官网下载的压缩包文件上传到服务器刚刚创建的Tomcat文件夹。

- 安装Tomcat

```
cd /usr/local/tomcat/  
tar -zxvf apache-tomcat-9.0.17.tar.gz
```

安装包会被解压到/usr/local/tomcat/apache-tomcat-9.0.17,执行命令将目录重命名为 tomcat9

```
mv /usr/local/tomcat/apache-tomcat-9.0.17 /usr/local/tomcat/tomcat9
```

- 开启 Tomcat 服务

```
cd /usr/local/tomcat/tomcat/bin/  
./startup.sh
```

- 验证 Tomcat 是否安装成功

☐☐用本地浏览器访问tomcat务器 IP:8080/ 如果能出现熟悉的 Tomcat 主页, 就表示安装成功了, 如是云服务器则需要去服务器的控制台去开启8080端口, 如果属于本地物理服务器需要在防火墙开放800端口。执行以下命令开放防火墙端口:

```
#一般情况下, ubuntu安装好的时候, iptables会被安装上, 如果没有的话先安装  
sudo apt-get install iptables  
#添加开放端口  
# sudo iptables -I INPUT -p tcp --dport [端口号] -j ACCEPT  
sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
# 临时保存配置, 重启后失效  
sudo iptables-save  
#安装 iptables-persistent工具, 持久化开放端口配置  
sudo apt-get install iptables-persistent  
sudo netfilter-persistent save  
sudo netfilter-persistent reload
```

☐☐Tomcat 服务默认开机不自启, 系统重启后需手动开启服务。如需要开机自启 Tomcat 服务请自行娘、Google或者参考[此篇文章](#)。

- 在Tomcat中运行系统

☐☐在tomcat9目录下的webapps/ROOT 是Tomcat 容器的网站根目录, 也就是我们熟悉的 Tomcat 页文件的目录, 将在Solo的github主页下载的压缩包先在本地进行解压然后通过winScp上传到 wena ps 目录下。

☐☐程序上传完成之后, 我们需要先将项目中的数据库连接地址跟密码改成自己的否则会运行失败。我可以在本地的解压的目录 solo下找到 local.properties配置文件, 然后右键用记事本打开, 此处我使用的是notpad++, 打开之后找到 MySQL runtime如下图所示:

```
2 # Author: Liang Ding  
3 #  
4  
5 ##### MySQL runtime #####  
6 runtimeDatabase=MYSQL  
7 jdbc.username=root  
8 jdbc.password=123456  
9 jdbc.driver=com.mysql.cj.jdbc.Driver  
0 jdbc.URL=jdbc:mysql://localhost:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true  
1  
2 ##### H2 runtime #####  
3 #runtimeDatabase=H2  
4 #jdbc.username=root  
5 #jdbc.password=  
6 #jdbc.driver=org.h2.Driver  
7 #jdbc.URL=jdbc:h2:~/solo_h2/db;MODE=MYSQL  
8
```

将 `jdbc.username` 设置成你数据库的用户名 `jdbc.password` 设置成你数据库的密码。修改完之后我在同级目录下再找到 `latke.properties` 按照实际情况进行修改，如下图所示：

```
2 # Author: Liang Ding
3 #
4 #
5 ##### MySQL runtime #####
6 runtimeDatabase=MySQL
7 jdbc.username=root
8 jdbc.password=123456
9 jdbc.driver=com.mysql.cj.jdbc.Driver
0 jdbc.URL=jdbc:mysql://localhost:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true
1
2 ##### H2 runtime #####
3 #runtimeDatabase=H2
4 #jdbc.username=root
5 #jdbc.password=
6 #jdbc.driver=org.h2.Driver
7 #jdbc.URL=jdbc:h2:~/solo_h2/db:MODE=MYSQL
8
```

否则会出现配置错误的页面，具体可以看[这里](#)

配置完成之后进入 tomcat 的 bin 目录下重新启动 tomcat 服务，然后我们就可以通过访问 `ip:8080/sol` 就能访问到博客的主页了，如果你想直接设置 `ip:8080` 访问的话，那么接下在我们要把 Tomcat 的访问目录换为 solo，最简单的办法就是将原来的 ROOT 文件夹删除或者重命名，再将 solo 文件夹重命名为 ROOT，这样我们通过服务器 `ip:8080` 就能访问 Solo 了。如果想通过不加端口的方式访问就需要修改 omcat 的配置文件。找到 tomcat 的 conf 目录下，找到 `service.xml` 文件如下图所示：

名称	修改日期	类型	大小
Catalina	2019/2/13/星期三 22...	文件夹	
catalina.policy	2018/6/29/星期五 10...	POLICY 文件	14 KB
catalina.properties	2018/6/29/星期五 10...	PROPERTIES 文件	8 KB
context.xml	2018/6/29/星期五 10...	XML 文档	2 KB
logging.properties	2018/6/29/星期五 10...	PROPERTIES 文件	4 KB
server.xml	2019/4/27/星期六 19...	XML 文档	7 KB
tomcat-users.xml	2018/6/29/星期五 10...	XML 文档	3 KB
tomcat-users.xsd	2018/6/29/星期五 10...	XSD 文件	3 KB
web.xml	2018/6/29/星期五 10...	XML 文档	171 KB

邮件用记事本或者使用 notepad++ 来打开找到下图所示的代码：

```
APR (HTTP/1.1) CONNECTOR: /docs/api.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
```

将 `port` 改为 80 即可，然后重启 tomcat 就可以直接通过域名访问不需要加端口，或者使用上面 nginx 置，安装 nginx 使用 nginx 反向代理。

- 验证 Solo 博客是否部署成功

先打开 Tomcat 服务，然后通过本地浏览器访问服务器 IP:808 如果配置了域名修改了端口那就直通过域名访问，没有配置域名就直接通过 IP 访问。

```
#运行命令开启服务
cd /usr/local/tomcat/tomcat9/bin/
```

./startup.sh

本地浏览器访问出现 Solo 的开始界面说明你的 Solo 博客就部署成功了，你就可以登录绑定你的 Gi Hub 账号开始写文章了。

5.2 Docker部署

Docker部署的话是solo博客系统最推荐的方式，而且作者也是希望部署博客的用户都使用这种方式因为对于后续的更新比较方便。使用Docker部署的话主要是以下的步骤：

- java环境配置前面已经谈到过不在赘述。
- 安装Docker

Docker 要求 Ubuntu 系统的内核版本高于 3.10,通过下面的命令查看内核版本安装，本篇文章在 Ubuntu18.04的环境下进行的操作，如果跟我采用的是同样的操作系统或者更高版本的就不需要再确认系统内核是否支持。如果不能确认执行命令 `uname -r`查看系统内核版本。确保你的内核版本要高于3.10 没有的话自己去更新内核。安装docker的话也有两种方式：

- 直接安装，不过这种方式安装的docker可能不是最新版本，不是很推荐，通过命令：`apt-get install docker-io`来安装，通过命令 `service start docker`启动dockers，之后可以敲命令 `docker run hello world`来去官方仓库下载hello-world镜像并运行，如果出现以下页面就表示成功：

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/quickstart/
```

- 通过添加官方源来安装最新版本的docker

(1)、首先卸载旧版本的docker，docker的旧版本名称为：docker、docker-engine 或者 docek-io。如果安装过旧版本的需要先卸载：

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

(2)、安装最新版本的 Docker

最新版本的 Docker 分两个版本，docker-ce(Community Edition)和docker-ee(Enterprise Edition)。CE版本是免费的，如果我们学习或者一般应用，CE足够。我们安装社区版：由于docker安装需要用https，所以需要使 apt 支持 https 的拉取方式。

(3)、安装 https 相关的软件包

```
sudo apt-get update # 先更新一下软件源库信息
sudo apt-get install \
```

```
apt-transport-https \  
ca-certificates \  
curl \  
software-properties-common
```

☐ (4) 、 设置apt仓库地址

☐ 鉴于国内网络问题，强烈建议使用国内地址，添加 Docker 官方apt仓库（使用国外源）执行该命令，如遇到长时间没有响应说明网络连接不到docker网站，需要使用国内的。

#添加 Docker 官方的 GPG 密钥（为了确认所下载软件包的合法性，需要添加软件源的 GPG 密钥

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
#设置稳定版本的apt仓库地址  
sudo add-apt-repository \  
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) \stabl  
"
```

● 添加 阿里云 的apt仓库（使用国内源）

```
curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -  
sudo add-apt-repository \  
"deb [arch=amd64] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \  
$(lsb_release -cs) \  
stable"
```

☐(5) 、 安装 Docker 软件

```
sudo apt-get update  
sudo apt-get install docker-ce # 安装最新版的docker
```

如果要安装指定版本的docker，则使用下面的命令：

```
apt-cache policy docker-ce # 查看可供安装的所有docker版本  
sudo apt-get install docker-ce=18.03.0~ce-0~ubuntu # 安装指定版本的docker
```

☐ 安装完了docker之后，我们下一步需要的是安装数据库了，可能你就会有疑问了，我在服务器上装了mysql啊，怎么又要安装了，这么说吧，服务器上的mysql是服务器上的，而docker中不使用服务器上的mysql，我们需要在docker上从官方仓库拉取mysql下来，这两个地方的mysql是不一样的，当如果你选择用docker的方式安装mysql的话，那么服务器上可以不装mysql，下面就是在docker上安装mysql：（我装的是mysql5.7）

☐ (1) 、 拉取镜像mysql:5.7

```
docker pull mysql:5.7
```

☐ (2) 、 新建目录date,logs,conf用来映射docker上的mysql容器的一些配置，防止mysql容器占用内存过大。

```
mkdir -p /opt/mysql/data /opt/mysql/logs /opt/mysql/conf
```

☐ data目录将映射为mysql容器配置的数据文件存放路径

☐ logs目录将映射为mysql容器的日志目录

☐ conf目录里的配置文件将映射为mysql容器的配置文件

☐ (3) 、 运行镜像创建容器

```
cd /opt/mysql
docker run -p 3306:3306 --name mysql -v $PWD/conf:/opt/mysql/conf -v $PWD/logs:/opt
mysql/logs -v PWD/data:/opt/mysql/data -e MYSQL_ROOT_PASSWORD=123456 -d mysql:5.7
16
```

命令说明:

- -p 3306:3306: 将容器的 3306 端口映射到主机的 3306 端口。
- -v \$PWD/conf:/etc/mysql/conf.d: 将主机当前目录下的 conf/my.cnf 挂载到容器的 /etc/mysql/my.cnf。
- -v \$PWD/logs:/logs: 将主机当前目录下的 logs 目录挂载到容器的 /logs。
- -v \$PWD/data:/var/lib/mysql: 将主机当前目录下的 data 目录挂载到容器的 /var/lib/mysql。
- -e MYSQL_ROOT_PASSWORD=123456: 初始化 root 用户的密码。

□□ (4)、查看正在运行的容器

```
#通过命令检查mysql是否安装成功
docker images #这个命令会显示你docker上的所有镜像, 比如现在估计你只有mysql和hello-w
rld
#通过命令检查mysql是否在后台运行
docker ps #可以查看当前在运行的镜像
```

□□ (5)、进入数据库, 创建solo库

```
# docker安装的mysql默认允许远程连接, 可以使用Navicat等软件连接数据库
# 进入容器mysql
docker exec -it mysql bash

# 进入数据库 p后面跟你的密码
mysql -uroot -pXXX

# 创建数据库(数据库名:solo;字符集utf8mb4;排序规则utf8mb4_general_ci)
create database solo DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
# 出现Query OK, 1 row affected (0.00 sec)表示成功
#退出数据库
exit
#退出容器
exit
```

● 部署solo博客执行以下命令

```
docker run --detach --name solo --network=host \
--env RUNTIME_DB="MYSQL" \
--env JDBC_USERNAME="你的用户名" \
--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=你的域名 --server_port
```

上面的命令仔细检查，免得出错，参数说明：

`--env JDBC_PASSWORD= "123456"` 将 123456 换成你的密码

`--listen_port=8080` 监听的端口

`--server_scheme=http` 请求方式，如果使用nginx需要将请求方式换成https

`--server_host=` 你的域名，如果你没有域名可以写 ip 地址

`--server_port=` 这里填你想映射的端口，http的话80和https的443可以不填，但不能注释，否则导致静态资源无法加载。

`--rm`因为这个容器后面要删掉，带上rm 会省很多事，如果不带rm的话，那么后面你要删除这个镜的话，首先你要删除它的容器，然后通过命令删除镜像：

```
docker rm 容器id #删除容器
docker rmi 镜像id #删除镜像
```

镜像id可以通过命令`docker images`查看到。带上了rm的话，那么后面执行命令`docker stop` 镜像就会自动删除镜像，便于要修改配置。

命令成功执行没有报错的话，通过`docker ps`查看执行的容器列表中是否存在 Solo，存在这表示启动成功，直接访问你的域名加:8080 即可访问你的博客。

如果使用nginx的话建议先安装nginx然后再运行Solo博客，避免重复启动关闭Solo。

6. 登录博客后台设置网站信息

最后可以输入你的公网ip访问你的网站了。首次访问时，需要初始化网站。你需要设置你的管理员号，然后开始初始化，Solo会自动在MySQL中建立数据库表。初始化成功后就自动进入你的网站了。

进入Solo后台管理控制台，进入"工具"->“偏好设定”，你可以修改你的网站名称等其他基本网站信息。

就这样，属于你的博客网站就已经基本搭建完成了。

更多关于Solo博客系统的可以访问开源社区[点击了解更多](#)

欢迎一起讨论