

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 , 可将文章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 全新的网络社区体验, 让热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

>III开始正题

<h3 id="一-搭建方式概述">一、搭建方式概述</h3>

>II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 作者推荐)。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 这两个工具中都可以完成。下面将对两种方式分别介绍。

<h3 id="二-部署步骤概述">二、部署步骤概述</h3>

>III1. 购买服务器

□12. 购买域名及备案 < br >

III3. 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 winS

p 主要是免费) < br >

004. 安装依赖程序 < br >

Ⅲ5. 安装博客程序

□□6. 通过 github 登录博客后台设置网站信息

<h3 id="三-具体内容">三、具体内容</h3>

<h4 id="--1-购买云服务器并配置">III.购买云服务器并配置</h4>

<II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供系统的,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,大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。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。

<h4 id="--2-购买域名及备案">002.购买域名及备案</h4>

<III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 购买一个。

□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! 建议家买服务器在谁家买域名!

<h4 id="--3-本地电脑安装-XShell和winScp软件">III3.本地电脑安装 XShell 和 winScp 软件</h4>III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 默认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 <a href="https:/

```
Id246.com/forward?goto=https%3A%2F%2Fblog.csdn.net%2Fweixin 42739326%2Farticle%2
details%2F82260588<sup>®</sup> target=<sup>"</sup>_blank<sup>®</sup> rel="nofollow ugc">具体操作方法</a>。<br>
III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。
<h4 id="--4-安装依赖软件">004.安装依赖软件</h4>
IDA 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:
>
01. Java
02. MySQL
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
<h5 id="--4-1安装java">III4.1 安装 java</h5>
IJava JDK 在 linux 系统有两个版本,一个开源版本 <strong>Openjdk</strong>,还有一个
strong>oracle 官方版本 jdk</strong>, oracle JDK 既可以通过添加 ppa 源命令行安装,也可以
官网下载 jdk 压缩包安装。本文不着重介绍,只介绍手动下载压缩包安装 oracle Java JDK。<br>
III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 <a href
"https://ld246.com/forward?goto=https%3A%2F%2Fwww.oracle.com%2Fjava%2Ftechnologie
%2Fjavase-downloads.html" target=" blank" rel="nofollow ugc">下载页面</a>下载 Linux 版的
JDK 压缩包,然后上传到服务器。
< u >
创建文件夹(以下操作以 jdk-8u241-linux-x64.tar.gz, 请注意对应文件名): 
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight</pre>
cl"> #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
</span></span></span><span class="highlight-line"><span class="highlight-cl"> sudo mkdir java
</span></span></code>
II将下载好的 jdk 压缩包用 winScp 上传到新创建的文件夹下
< u | >
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight</pre>
cl"> sudo tar -zxvf jdk-8u241-linux-x64.tar.gz -C java
</span></span></code>
<u>
修改环境变量
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight</pre>
cl"> sudo vi ~/.bashrc
</span></span></code>
>III在文件末尾追加下面内容: 
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl"> #set oracle jdk environment
</span></span><span class="highlight-line"><span class="highlight-cl">
                                                                                                export JAVA H
ME=/root/iava/idk1.8.0 241 #这里要注意目录要换成自
                                                                        己解压的idk 目录
</span></span><span class="highlight-line"><span class="highlight-cl">
                                                                                                export JRE HO
E=${JAVA HOME}/jre
</span></span></span><span class="highlight-line"><span class="highlight-cl">
                                                                                                export CLASSP
TH=.:${JAVA HOME}/lib:${JRE HOME}/lib
</span></span><span class="highlight-line"><span class="highlight-cl">
                                                                                               export PATH=$
JAVA_HOME}/bin:$PATH
</span></span></code>
系统注册此 jdk
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight">
cl"> sudo update-alternatives --install /usr/bin/java java /root/java/jdk1.8.0 241/bin/java 3
```

```
0
</span></span></code>
< u >
查看 java 版本确定是否安装成功: 
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight"</pre>
cl"> java -version
</span></span></code>
< u >
如果看到下图类似的提示说明安装成功: <br>
<img src="https://ld246.com/images/img-loading.svg" alt="QQ 截图 20200405150256.png"
ata-src="https://b3logfile.com/file/2020/04/QQ%E6%88%AA%E5%9B%BE20200405150256-d
bab3fe.png?imageView2/2/interlace/1/format/jpg">
<h5 id="--4-2安装MySQL">III4.2 安装 MySQL</h5>
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
2 内存 DB 请<a href="https://ld246.com/forward?goto=https%3A%2F%2Fhacpai.com%2Fartic
e%2F1492881378588%23%25E5%2590%25AF%25E5%258A%25A8%25E5%25AE%25B9%25E
%2599%25A8" target=" blank" rel="nofollow ugc">参考此处</a>。 <br>
ⅢMySQL 建议使用 8.0 以上版本,当然 5.7-8.0 以上的版本都是是可以的。我个人使用 5.7 版本安装
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 Solo 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
装数据库中再介绍 5.7 版本。
<h6 id="--4-2-1进入官网下载">III4.2.1 进入官网下载</h6>
四进入<a href="https://ld246.com/forward?goto=https%3A%2F%2Fdev.mysgl.com%2Fd
wnloads%2Frepo%2Fapt%2F" target=" blank" rel="nofollow ugc">官网</a>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。 <br>
<img src="https://ld246.com/images/img-loading.svg" alt="QQ 截图 20200405154545.png"
ata-src="https://b3logfile.com/file/2020/04/QQ%E6%88%AA%E5%9B%BE20200405154545-5
424b1a.png?imageView2/2/interlace/1/format/jpg">
<h6 id="--4-2-2开始安装">III4.2.2 开始安装</h6>
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。
< u >
,li>执行命令
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight</pre>
cl"> sudo dpkg -i mysql-apt-config 0.8.15-1 all.deb
</span></span></code>
< u >
<strong>洗择 8.0 回车</strong><br>
<img src="https://ld246.com/images/img-loading.svg" alt="QQ 截图 20200405155354.png"
ata-src="https://b3logfile.com/file/2020/04/QQ%E6%88%AA%E5%9B%BE20200405155354-1
15cc25.png?imageView2/2/interlace/1/format/jpg">
<strong>继续选择 8.0 回车</strong>
<img src="https://ld246.com/images/img-loading.svg" alt="QQ 截图 20200405155638.p
g" data-src="https://b3logfile.com/file/2020/04/QQ%E6%88%AA%E5%9B%BE202004051556
8-87d7f544.png?imageView2/2/interlace/1/format/jpg">
< u >
<strong>选择 OK 回车</strong>
<img src="https://ld246.com/images/img-loading.svg" alt="QQ 截图 20200405155802.p
```

g" data-src="https://b3logfile.com/file/2020/04/QQ%E6%88%AA%E5%9B%BE202004051558 2-e080a168.png?imageView2/2/interlace/1/format/jpg"> < u >执行命令 <code class="highlight-chroma"><span class="highlight</pre> cl"> sudo apt update sudo apt-get u grade </code> < u >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-ser er"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. 。此时调用命令 "sudo apt install mysql-server" 安装的即为 8.0 版本。 <code class="highlight-chroma"><span class="highlight</pre> cl"> sudo apt install mysql-server #中间会进行一 选择, 输入"Y"即可继续。 </code> <u>执行完之后会有两次输入密码
 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。
 登录验证 <code class="highlight-chroma"><span class="highlight"</pre> cl"> mysql -u root -p #输入密码进行验证 </code> 至此 MySQL 安装完成。 <h6 id="--4-2-3MySQL配置远程访问">III4.2.3MySQL 配置远程访问</h6> < u >执行以下命令 <code class="highlight-chroma"><span class="highlight</pre> cl"> mysql -uroot -p #登录MySQL use mysal; select host from user where user='root'; update user set host = '%' where user ='root'; flush privileges; quit; </code> <h5 id="--4-3安装nginx">III4.3 安装 nginx</h5> Image <p 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

跳过此步骤, 然后通过 docker 的端口映射实现 80 端口的访问。 < u >使用 apt 命令安装 nginx <code class="highlight-chroma"><span class="highlight</pre> cl"> sudo apt install nginx </code> < u > $\langle | \rangle$ 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 $\langle | \rangle$ 启动并验证 <code class="highlight-chroma"><span class="highlight</pre> cl"> service nginx start #启动nginx service nginx re oad #重新加载nginx配置文件 class="highlight-line"> nginx -s reopen #重启 Nginx nginx -s stop #停止 Nginx </code> < u >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
 III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 绿色的小锁,不会被抓包,比 http 安全。
 III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 下载后解压会出现几个文件夹:

 需要拿到 Nginx 文件夹内的文件,这是公钥和私钥:

 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: </li <code class="highlight-chroma"><span class="highlight</pre> cl">server { listen 443; server name w w.taosugar.com; # 改为绑定证书的域名 # ssl 配置 ssl on: ssl certificate /e

C/nginx/conf.d/taosugar.com.crt; # 成为自己中頃侍到的 Crt 义件的路径	
	ssl_certificate_k
y /etc/nginx/conf.d/taosugar.com.key; # 改为自己申请得到的 key 文件的路径 	ssl_session_tim
out 5m;	
 Sv1 TLSv1.1 TLSv1.2;	ssl_protocols T
 HE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!A	ssl_ciphers EC \DH:!RC4:!DH:!D
	ssl prefer serve
ciphers on;	
	
	location / {
lass="highlight-line"> 	proxy_pass ht
p://localhost:8080; # 做端口转发	
	}
 404 html:	error_page 404
	error_page 500
<pre>SU2_SU3_SU4_/SUX.IIIIII, </pre>	
m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——些重
向. 这样不管怎么访问都是 <a href="https://ld246.com/forward?goto=https%3</td><td>A%2F%2Fwww.t</td></tr><tr><td>osugar.com" rel="nofollow ugc" target=" blank">https://www 打头, 在	ssl.conf 里再追
:	
<pre><code class="highlight-chroma"><span< pre=""></span<></code></pre>	class="highlight
cl">server {	
	listen 80;
 	listen 80; server_name w
 w.taosugar.com;	listen 80; server_name w
 w.taosugar.com; ps://\${server_pame}\$1_permapent;	listen 80; server_name w rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; }	listen 80; server_name w rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } }	listen 80; server_name w rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } < < <td>listen 80; server_name w rewrite ^(.*)\$ ht erver {</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver {
 w.taosugar.com; ps://\${server_name}\$1 permanent; } < < <td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta
 w.taosugar.com; ps://\${server_name}\$1 permanent; } } } se se se se sugar.com:	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta
 w.taosugar.com; ps://\${server_name}\$1 permanent; } <td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } } } se se sugar.com; <td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } se se sugar.com; <td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } sugar.com; sugar.com; sugar.com; sigar.com; 	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } se <span class="highlight-c</td><td>listen 80;
server_name w
rewrite ^(.*)\$ ht
erver {
listen 80;
server_name ta
rewrite ^(.*)\$ ht
erver {
listen 443;</td></tr><tr><td> w.taosugar.com; ps://\${server_name}\$1 permanent; } se se sugar.com; ps://www.\${server_name}\$1 permanent; syan> syan> syan> se <	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta
 w.taosugar.com; ps://\${server_name}\$1 permanent; } sugar.com; <span class="hi</td><td>listen 80;
server_name w
rewrite ^(.*)\$ ht
erver {
listen 80;
server_name ta
rewrite ^(.*)\$ ht
erver {
listen 443;
server_name ta</td></tr><tr><td> w.taosugar.com; ps://\${server_name}\$1 permanent; } < > > >< <td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; sugar.com; <span <="" class="highlight-cl" td=""><td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } sugar.com; > <span class="h</td><td>listen 80;
server_name w
rewrite ^(.*)\$ ht
erver {
listen 80;
server_name ta
rewrite ^(.*)\$ ht
erver {
listen 443;
server_name ta
rewrite ^(.*)\$ ht</td></tr><tr><td> class=" highlight-line"=""> w.taosugar.com; ps://\${server_name}\$1 permanent; <td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht
<td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht</td>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht
 w.taosugar.com; ps://\${server_name}\$1 permanent; } se sugar.com; ps://www.\${server_name}\$1 permanent; } <s< td=""><td>listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht server { .*)\$ ht</td></s<>	listen 80; server_name w rewrite ^(.*)\$ ht erver { listen 80; server_name ta rewrite ^(.*)\$ ht erver { listen 443; server_name ta rewrite ^(.*)\$ ht server { .*)\$ ht

III <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.taosugar.com" targe

="_blank" rel="nofollow ugc">http://taosugar.com

III 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...

<code class="highlight-chroma"> sudo service nginx reload

</code>

比处需要注意请务必配置

<code class="highlight-chroma"> proxy_set_header Host \$http_host;

 proxy_set_head r X-Real-IP \$remote_addr;

</code>

<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。

<ID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,这次在众多的开源博客程序中选择了一款 Java 开源博客系统: Solo</a .

 <br/

Ⅲ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。

<h5 id="--5-1部署到Tomcat">115.1 部署到 Tomcat</h5>

< Solo 的 github 下载 war 包,将其传上服务器; /li>

ubuntu 安装 tomcat, 官网下载最新的 tomcat9, 或者通过 wget 下载:

<code>wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-9/v9.0.17/bin/apache-tomcat-9. .17.tar.gz</code>

创建 tomcat 文件夹并进入

<code class="highlight-chroma"> cd /usr/local/

</

<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。</p>

安装 Tomcat

<code class="highlight-chroma"> cd /usr/local/tomcat/

 tar -zxvf apache tomcat-9.0.17.tar.gz

</code>

>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 tomcat

<code class="highlight-chroma"> mv /usr/local/tomcat/apache-tomcat-9.0.17 /usr/local/tomcat/tomcat9 </code>

开启 Tomcat 服务 <code class="highlight-chroma"> cl" > cd /usr/local/tomcat/tomcat/bin/ ./startup.sh </code> < u >>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 080 端口。执行以下命令开放防火墙端口: <code class="highlight-chroma"><span class="highlight</pre> cl"> #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 in tall iptables #添加开放端口 # sudo iptables I INPUT -p tcp --dport [端口号] -j ACCEPT sudo iptables -I NPUT -p tcp --dport 80 -j ACCEPT #临时保存配置 重启后失效 sudo iptables-s ve #安装 iptablesersistent工具,持久化开放端口配置 sudo apt-get in tall iptables-persistent sudo netfilter-p rsistent save sudo netfilter-p rsistent reload </code> ITomcat 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 行度娘、Google 或者参考 此篇文章。 <u>在 Tomcat 中运行系统
 III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 主页文件的目录,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 wenapps 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 <code>solo</code> 下找到 <code>local.properties</code> 配置文 ,然后右键用记事本打开,此处我使用的是 notpad++,打开之后找到 <code>MySQL runtime</c de> 如下图所示:

 将 <code>jdbc.username</code> 设置成你数据库的用户名 <code>jdbc.password</code> 设 成你数据库的密码。修改完之后我们在同级目录下再找到 < code>latke.properties</code> 按照实 情况进行修改,如下图所示:
 <img src="https://ld246.com/images/img-loading.svg" alt="QQ 图片 20200405182958.png" ata-src="https://b3logfile.com/file/2020/04/QQ%E5%9B%BE%E7%89%8720200405182958-c

3e09b3.png?imageView2/2/interlace/1/format/jpg">

<否则会出现配置错误的页面,具体可以看这

III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 solo 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 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 要修改 tomcat 的配置文件。找到 tomcat 的 conf 目录下,找到 service.xml 文件如下图所示:

邮件用记事本或者使用 notpad++ 来打开找到下图所示的代码:

将 <code>port</code> 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 配置,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。

<code class="highlight-chroma"> #运行命令开启服务

class="highlight-line"> cd /usr/local/t mcat/tomcat9/bin/

</

<II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 GitHub 账号开始写文章了。

<h5 id="--5-2Dcoker部署">105.2Dcoker 部署</h5>

>IIDocker 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 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 Uuntu18.04 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统内核是否支持。如果不能确认执行命令 <code>uname -r</code> 查看系统内核版本。确保你的核版本要高于 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: <code>ap -get install docker-io</code> 来安装,通过命令 <code>service start docker</code> 启动 doc ers,之后可以敲命令 <code>docker run hello-world</code> 来去官方仓库下载 hello-world 镜 并运行,如果出现以下页面就表示成功:

<ing src="https://ld246.com/images/img-loading.svg" alt="QQ 截图 20200405191259.png" ata-src="https://b3logfile.com/file/2020/04/QQ%E6%88%AA%E5%9B%BE20200405191259-d fbf40f.png?imageView2/2/interlace/1/format/jpg">

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 doce r-io。如果安装过旧版本的需要先卸载:

<code>sudo apt-get remove docker docker-engine docker.io</code>

(2) 、安装最新版本的 Docker < br>

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 使用 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 <code class="highlight-chroma"><span class="highlight</pre> sudo apt-get update # 先更新一下软件源库信息 cl"> sudo apt-get in tall \ apt-transport-h tps \ ca-certificates \ curl \ software-prope ties-common </code> >III (4) 、 设置 apt 仓库地址
 III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 时,如遇到长时间没有响应说明网络连接不到 docker 网站,需要使用国内的。 <code class="highlight-chroma"> cl"> #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 钥) curl -fsSL https //download.docker.com/linux/ubuntu/gpg | sudo apt-key add - #设置稳定版本 apt仓库地址 sudo add-apt-r pository \ "deb [arch=a d64] https://download.docker.com/linux/ubuntu \ \$(lsb release -cs) \stable" </code> <u>添加 阿里云 的 apt 仓库 (使用国内源) <code class="highlight-chroma"> curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add cl"> sudo add-ap -repository \ "deb [arc =amd64] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \ class="highlight-line"> \$(lsb rele se -cs) \ stable" </code> >III(5)、安装 Docker 软件 <code class="highlight-chroma"><span class="highlight</pre> cl"> sudo apt-get update sudo apt-get in tall docker-ce # 安装最新版的docker </code> 如果要安装指定版本的 docker,则使用下面的命令: <code class="highlight-chroma"> cl"> apt-cache policy docker-ce # 查看可供安装的所有docker版本 sudo apt-get in

tall docker-ce=18.03.0~ce-0~ubuntu # 安装指定版本的docker </code> III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 上安装了 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 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 不一样的,当然如果你选择用 docker 的方式安装 mysql 的话,那么服务器上可以不装 mysql,下 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 < br> □ (2) 、新建自录 date,logs,conf 用来映射 docker 上的 mysql 容器的一些配置, 防止 mysql 容器 用内存过大。
 <code>mkdir -p /opt/mysql/data /opt/mysql/logs /opt/mysql/conf</code>
 IIIdata 目录将映射为 mysql 容器配置的数据文件存放路径 < br> □logs 目录将映射为 mysql 容器的日志目录 < br> IIIconf 目录里的配置文件将映射为 mysql 容器的配置文件 < br> □□ (3) 、运行镜像创建容器 <code class="highlight-chroma"><span class="highlight</pre> cl"> cd /opt/mysql docker run -p 3 06:3306 --name mysql -v \$PWD/conf: /opt/mysql/conf -v \$PWD/logs:/opt/mysql/logs -v PW /data:/opt/mysql/data -e MYSQL ROOT PASSWORD=123456 -d mysql:5.7.16 </code> >命令说明: <code class="highlight-chroma"><span class="highlight</pre> cl">□ -p 3306:3306: 将容器的 3306 端口映射到主机的 3306 端口。 III -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。 logs:将主机当前目录下的 logs 目录挂载到容器的 /logs。 00 -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。 00 -e MYSQL R OT PASSWORD=123456: 初始化 root 用户的密码。 </code> >III (4) 、查看正在运行的容器 <code class="highlight-chroma"><span class="highlight</pre> cl"> #通过命令检查mysql是否安装成功 class="highlight-line"> 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mysgl和hello-world #通过命令检查 vsal是否在后台运行 docker ps 可以查看当前在运行的镜像 </code> >III (5)、进入数据库, 创建 solo 库 <code class="highlight-chroma"><span class="highlight"</pre> cl"> # docker安装的mysgl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mysg class="highlight-line"> docker exec -it

mysgl bash # 进入数据库 p 面跟你的密码 mysql -uroot -XXX # 创建数据库(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 </code> < u >部署 solo 博客执行以下命令 <code class="highlight-chroma"><span class="highlight</pre> cl"> docker run --detach --name solo --network=host \ --env RUNTIME DB="MYSQL" \ --env JDBC US RNAME="你的用户名" \ --env JDBC PA SWORD="123456" \ --env JDBC DR VER="com.mysql.cj.jdbc.Driver" \ --env JDBC UR ="jdbc:mysql://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=UTF-8&u eSSL=false&serverTimezone=UTC" \ b3log/solo --lis en port=8080 --server scheme=http --server host=你的域名 --server port= </code> 上面的命令仔细检查,免得出错,参数说明: >II--env JDBC PASSWORD= "123456" 将 123456 换成你的密码
 III--listen port=8080 监听的端口
 III--server scheme=http 请求方式,如果使用 nginx 需要将请求方式换成 https
 III--server host= 你的域名, 如果你没有域名可以写 ip 地址 < br> III--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, 则会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 像的话,首先你要删除它的容器,然后通过命令删除镜像: <code class="highlight-chroma"><span class="highlight</pre> cl"> docker rm 容器id #删除容器 docker rmi 镜像 d #删除镜像 </code> □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 st p镜像名就会自动删除镜像,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

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。<h4 id="--6--登录博客后台设置网站信息">□06. 登录博客后台设置网站信息</h4>

<III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 员帐号,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 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站信息。

□□就这样,属于你的博客网站就已经基本搭建完成了。

>III更多关于 Solo 博客系统的可以访问开源社区点击了解更

<code class="highlight-chroma">欢迎一起讨论

</code>