



链滴

从零开始安装 solo 博客

作者: [InkDP](#)

原文链接: <https://ld246.com/article/1565021959471>

来源网站: [链滴](#)

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



之前也写过类似的帖子，但是由于那时自己的各种原因写的不是特别好，所以今天写一个聚合帖，记从购买服务器到安装solo然后通过nginx反向代理，最后升级https的全过程。此贴献给完全无基础的，所以废话较多，见谅

1. 购买服务器

首先你得有自己的服务器，有的话就跳过。服务商可选的有很多，比如：[阿里云](#)、[腾讯云](#)、[百度云](#)、有一些香港的服务商以及国外的（有特殊需求的可以考虑下）。腾讯和阿里对于新用户以及学生都有大的优惠，配置的话如果只挂solo，买最低配1G1核1M即可。国内的几家都可以关注下，不定时会很好的优惠活动。我比较推荐[阿里云](#)，前段时间刚买了一台3年才668多，不知道活动结束后没有。阿里购买服务器时会要求你安装系统（不知道可不可以不选），推荐选择centos。

2. 购买域名(可不买)

建议还是买一个域名，直接通过IP访问的话不是特别好。购买域名时不要盯着 [.com](#)，[.cn](#)这种比较热的域名，往往很贵。还有尽量选择可备案的域名，否则就会像[鼠鼠在碎觉](#)一样只能挂载444端口上运。可通过[域名.信息](#)查看可备案域名。如果你服务器买着国外的话似乎就不用备案。

3. 域名解析与备案

服务器和域名购买完后需要将域名解析到服务器，有些服务商可能不支持跨服务商解析，腾讯云域名以解析阿里云服务器。解析过程大概需要10分钟。解析完成后如下图所示：

域名	解析状态 ①	解析类型	最后操作时间	操作
jinjiah.com	正常解析	免费套餐	2019-07-18 17:42:22	解析 升级套餐 更多 *

如果不备案的话，80端口与443端口大概率会被封。所以备案还是需要的，备案的过程有点麻烦需要心。大致步骤如下：



接入备案很快，几天就完事了。

4. 为服务器安装系统

给服务器安装系统不用像你给自己装系统那样麻烦，很方便，由于我使用的centos，所以推荐安装centos，以免出现不必要的错误。后续操作噓设置服务器安全组，开发如下端口：**80、8080、3306、443、22**。具体左右我会在下面一一说明，**22用来远程连接服务器使用**，windows用户可下载xshell远程链服务器。

5. 安装docker

直接使用yum安装，简单快捷

```
#安装 Docker
yum -y install docker
```

```
#启动 Docker 后台服务
service docker start
```

```
#测试运行 hello-world
docker run hello-world
```

出现hello world 就证明安装正常了

```
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/get-started/
```

6. 安装Mysql

版本随意，我这里选择的5.6，你可以选择更高版本的，这个没关系，不影响使用

```
# 安装mysql:5.6,直接docker run 他会自动去官方镜像下载
# MYSQL_ROOT_PASSWORD=你的数据库密码
```

```
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql:5.6
```

```
# 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
```

7. 安装solo

直接运行以下命令

```
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=UTF-8&useSSL=false&serverTimezone=UTC" \  
--rm \  
b3log/solo --listen_port=8080 --server_scheme=http --server_host=www.jinjianh.com
```

上面的命令建议手敲，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，暂时使用http，后面我们会换成https
- `--server_host=www.jinjianh.com` 你的域名，如果你没有域名可以写ip地址
- `--rm`因为这个容器后面要删掉，带上rm会省很多事。

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，存在这表示启动功，直接访问你的域名加:8080即可访问你的博客，[金戈博客--http://www.jinjianh.com:8080](http://www.jinjianh.com:8080)

如果你尚在备案中，你可以收藏本帖，后面等备案通过了在研究后面的部分。

如果你不想使用nginx也不想升级https，那么你可以先执行 `docker stop solo`，然后将上面 `--listen_ort=8080`的 `8080`换成 `80`，然后去掉 `--rm`，再执行一次就ok。

8. 安装nginx

安装nginx前，我们现在本地建立几个文件，用于存放nginx的配置文件等

```
# 切换到服务器根目录
```

```
cd /
# 创建主目录
mkdir dockerData
# 创建文件
mkdir dockerData/nginx dockerData/nginx/conf dockerData/nginx/logs dockerData/nginx/w
w dockerData/nginx/ssl
```

上面的 `dockerData` 可以换成自己喜欢的名字

- `dockerData/nginx` 用于存放docker下nginx自定义文件
- `dockerData/nginx/conf` 存放nginx配置文件
- `dockerData/nginx/log` 存放nginx日志文件
- `dockerData/nginx/www` 存放nginx访问的资源文件
- `dockerData/nginx/ssl` 存放ssl证书

启动nginx

```
docker run --name nginx -p 80:80 -d --rm nginx
```

如果你没有备案，可以将上面的 `80:80` 换成 `8081:80`，因为这个东西一会儿也要删掉，所以加上 `--rm` 数，命令执行玩后通过 `docker ps` 查看nginx是否在运行，在运行的情况下访问你的域名加端口号是否正常安装，`80` 直接省略。如下表示访问成功

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

导出配置文件

- `docker cp nginx:/etc/nginx/nginx.conf /dockerData/nginx/conf/nginx.conf` 导出配置文件 `nginx.conf`
 - `docker cp nginx:/etc/nginx/conf.d /dockerData/nginx/conf/conf.d` 导出配置为你 `nginx.conf`
- 执行 `docker stop nginx`，会自动删除现在的nginx容器，然后执行如下命令重新启动一个nginx容器

```
docker run -d -p 80:80 --name nginx \
-v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \
-v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d \
-v /dockerData/nginx/www:/usr/share/nginx/html \
-v /dockerData/nginx/logs:/var/log/nginx nginx
```

- `-v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf` \ 挂载配置文件 `nginx.conf`
- `-v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d` 挂载配置文件 `default.conf`
- `-v /dockerData/nginx/www:/usr/share/nginx/html` 挂载项目文件
- `-v /dockerData/nginx/logs:/var/log/nginx` 挂载配置文件

访问你的域名，你会发现报错了



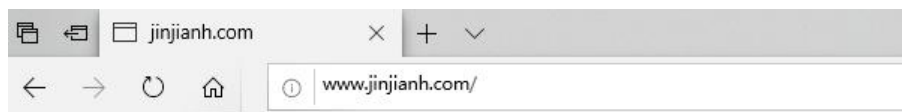
这时我们可以前往 `/dockerData/nginx/logs` 下查看日志文件

```
2019/08/05 14:57:54 [error] 6#6: *3 directory index of "/usr/share/nginx/html/" is forbidden, client: 121.32.33.217, server: localhost, request: "GET / HTTP/1.1", host: "www.jinjianh.com"
```

因为 `/usr/share/nginx/html/` 被挂载到了服务器上面的 `/dockerData/nginx/www` 目录下，原来的迎页面在 `dockerData/nginx/www` 是没有的，所以就报错了，这里我们随便建一个。

```
# 打开项目文件
cd /dockerData/nginx/www
# 使用vim 创建并编辑文件
vim index.html
# 此时我们会进入vim界面，按 i 插入，然后输入
<h1>Hello Docker-Nginx</h1>
# 输入完后，按 esc，然后输入 :wq
```

再次访问我们的域名就可以看到我们刚刚写的 `h1` 标签内容



Hello Docker-Nginx

9. 申请ssl证书，将http升级为https(可跳过)

`https` 想比与 `http` 来说，最核心的内容就是多了一个ssl证书，证书是可以免费申请的。

腾讯云

访问 [SSL证书选购](#) 申请

1. 【重要通知】由于CA机构和证书代理策略调整，从2018年1月1日起，同一主域最多只能申请20张亚洲诚信品牌免费型DV版SSL证书（一级域名及其子域名均属于同一主域，例如 domain.com, ssl.domain.com, ssl.ssl.domain.com 都属于同一主域），之前已颁发的证书在有效期内使用不受影响（注：即将到期的证书需要18年1月1日以后重新申请时，会受到上述策略的限制），感谢您的业务因此次调整受扰，建议您购买泛域名型SSL证书。

证书种类： 企业型(OV) 企业型专业版(OV Pro) 域名型(OV) 域名型免费版(DV) 增强型(EV) 增强型专业版(EV Pro)

域名型加配SSL证书，浏览器上有https提示并有绿锁标记，仅对域名所有权进行验证，快速颁发，较好保护网站数据安全，适合个人、小微企业应用。

证书品牌：TrustAsia
厦门联克亚太白金战略合作伙伴亚洲诚信(TrustAsia)自研证书品牌，由厦门联克证书签发。

域名类型：单域名
仅支持绑定一个一级域名或者子域名，例如 domain.com, ssl.domain.com, ssl.ssl.domain.com 分别为一个域名，注意 domain.com 不包含 ssl.domain.com 等子域名，如果需要支持所有二级或三级域名，请购买通配符证书。

证书年限：1年

所属项目：

总计费用：**0元**

私钥可不填写

1. 【重要通知】由于CA机构和证书代理策略调整，从2018年1月1日起，同一主域最多只能申请20张亚洲诚信品牌免费型DV版SSL证书（一级域名及其子域名均属于同一主域，例如 domain.com, ssl.domain.com, ssl.ssl.domain.com 都属于同一主域），之前已颁发的证书在有效期内使用不受影响（注：即将到期的证书需要18年1月1日以后重新申请时，会受到上述策略的限制），感谢您的业务因此次调整受扰，建议您购买泛域名型SSL证书。
2. 近期由于厦门联克SSL证书颁发系统升级，域名验证环节将进行更新调整，可能导致部分原本可正常颁发的OV版SSL证书，升级后无法重新申请，如遇此情况，建议更换域名申请或购买OV/EV版付费证书，给您带来的不便，敬请谅解。

通用名称*

申请邮箱*

证书备注名

私钥密码
目前暂不支持密码找回，若忘记密码则需要重新申请证书

所属项目：

选择手动DNS验证

手动DNS验证
需您手动为域名添加一条解析记录。 [详细说明](#)

文件验证
需要在域名根目录下创建指定的文件验证域名所有权。 [详细说明](#)

可直接前往 [SSL 证书 域名验证指引 - 操作指南 - 文档中心 - 腾讯云](#)查看

手动 DNS 验证

① 说明:

本 DNS 验证以腾讯云云解析平台为例进行说明。

查看证书详情

1. 登录 [SSL 证书管理控制台](#)。
2. 在“证书列表”页面，选择待查看证书详情的域名型证书 ID，进入“证书详情”页面。如下图所示：

← 证书详情

您的申请信息已提交。请尽快添加DNS解析记录，扫描认证通过后即可颁布证书。

基本信息

ID QI5CMYUE

备注名称 test

状态 待DNS验证
请添加如下解析记录 [操作指引](#)

域名	主机记录	记录类型	记录值
	_dnsauth	TXT	20181227220956.....hj37i4xal8m7uul2a23f

自助诊断 [查询](#)

证书类型 TrustAsia TLS RSA CA(1年)

通用名称 v.....n

提交时间 2018-12-27 10:22:28

3. 在“证书详情”的基本信息中，查看证书状态。
请您到域名对应的[域名解析商](#)处进行解析。您可以通过解析指定的 DNS 记录验证您的域名所有权，指定如“主机记录 > TXT 记录类型 > 记录值”的解析格式。
例如，您的域名对应的[域名解析商](#)在腾讯云，请您登录 [云解析控制台](#) 为申请证书的域名 `www.domain.com` 添加一条记录类型为 TXT 的 DNS 记录，指定的解析格式为“`www.domain.com > TXT > 20181227220956.....hj37i4xal8m7uul2a23f`”。

① 说明:

解析记录填写请按“添加域名 > 域名解析记录”操作。

添加域名

1. 切换至 [云解析控制台](#)，并在“域名解析列表”中，单击【添加解析】。如下图所示：

域名解析列表 全部项目

[进入我的域名列表](#) [用手机管理域名](#)

工信部要求，在中国大陆地区开展网站服务，必须在服务器提供商处完成域名/网站备案。如您解析指向腾讯云国内服务器，需在腾讯云为网站办理接入备案。了解详情

[添加解析](#) [购买解析套餐](#) [批量操作](#)

<input type="checkbox"/>	域名	解析状态 ①	解析套餐	最后操作时间	操作
<input type="checkbox"/>		域名 DNS 未修改 ①	个人专业版 2019-11-12 到期	2018-11-20 10:30:08	解析 升级套餐 更多

2. 在弹出的“添加解析”窗口中，输入您需要解析域名的主域名，例如“domain.com”。
3. 单击【确定】，完成添加。

添加解析记录

1. 在新添加的域名行中，单击【解析】。如下图所示：



2. 在域名解析页面，单击【添加记录】，显示新记录行。如下图所示：



关键参数信息如下：

- 主机记录：根据 [证书详情](#) 进行填写，例如 "_dnsauth"。
- 记录类型：选择“TXT”。
- 线路类型：选择“默认”。
- 记录值：为系统提供的文本内容，记录值须完整填写。请根据 [证书详情](#) 进行填写。
- TTL (秒)：选择默认值600秒。

3. 单击【保存】。

4. 在弹出的“添加成功”提示框中，单击【我知道了】，记录列表显示新添加的记录信息，即表示添加完成。如下图所示：



① 说明：

证书的域名（例如 `www.domain.com`）TXT 记录值的系统会定时检查，若能检测到并且与指定的值匹配，即可完成域名所有权验证。

前往[证书管理-控制台](#)等待验证通过即可

名称	修改日期	类型	大小
Apache	2019/8/5 23:26	文件夹	
IIS	2019/8/5 23:26	文件夹	
Nginx	2019/8/5 23:26	文件夹	
Tomcat	2019/8/5 23:26	文件夹	
www.jinjianh.com.csr	2019/8/5 23:26	CSR 文件	1 KB

将 nginx下文件上传到 /dockerData/nginx/ssl目录下即可

阿里云

访问云盾证书服务申请，访问后如果有无内容可复制 <https://common-buy.aliyun.com/?spm=517610695662.958455.3.3f9140d55mPzFH&commodityCode=cas#/buy> 打开

走一下支付流程，然后申请

证书	品牌	状态	域名域名	有效期	已部署	操作
Symantec 免费DV SSL	Symantec	已付款		1年		操作

证书申请



填写申请

验证信息

- * 证书绑定域名:
- * 所在地: 中国大陆 / 北京市
- * 域名验证方式: 自动DNS验证 手工DNS验证 文件验证
- * 申请人姓名: 胡钱玲
- * 申请人手机号:
- * 申请人邮箱:
- * CSR生成方式: 系统生成 手动填写

全部品牌

联系我们

下一步

取消



操作流程基本与腾讯云一样，我就不详细说明了

其他平台

免费的证书平台有很多，也可以去别的地方申请

上传证书

一下示例为腾讯云证书，阿里云



验证完后，我们下载证书，解压后得到

名称	修改日期	类型	大小
Apache	2019/8/5 23:26	文件夹	
IIS	2019/8/5 23:26	文件夹	
Nginx	2019/8/5 23:26	文件夹	
Tomcat	2019/8/5 23:26	文件夹	
www.jinjianh.com.csr	2019/8/5 23:26	CSR 文件	1 KB

最后把 Nginx 下的两个文件上传至服务器 `/dockerDat/nginx/ssl` 目录下，别的服务商申请的证书也一样，将最后的ssl证书放到 `/dockerDat/nginx/ssl` 下即可

配置nginx配置文件

```
cd /dockerData/nginx/conf/conf.d
vim default.conf
```

参考我的配置，配置自己的default.conf文件

```
server {
    listen    443;
    server_name localhost;
    ssl on;

    #charset koi8-r;
    #access_log /var/log/nginx/host.access.log main;

    ssl_certificate /ssl/1_www.jinjianh.com_bundle.crt; # ssl 证书目录
    ssl_certificate_key /ssl/2_www.jinjianh.com.key;
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;

    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
    }
    # .....
}
server{
    listen 80;
    server_name localhost;
    rewrite ^(.*) https://$host$1 permanent;
}
# 按esc, 然后输入:wq保持退出
```

不重要的部分我省略了，可根据自己服务器配置做出相应调整

由于我们现在用的nginx容器并未监听443端口，所以需要删除现在的容器，重新启动一个新的nginx器

```
docker stop nginx # 停止容器
docker rm nginx # 删除容器
# 启动新的
docker run -d -p 80:80 -p 443:443 --name nginx \
-v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \
-v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d \
-v /dockerData/nginx/ssl:/ssl/ \
-v /dockerData/nginx/www:/usr/share/nginx/html \
-v /dockerData/nginx/logs:/var/log/nginx nginx
```

- **-p 443:443** 监听443端口
- **-v /dockerData/nginx/ssl:/ssl/** 挂载ssl证书目录

访问查看，一切正常



Hello Docker-Nginx

10. 将solo通过nginx方向代理实现https访问

让solo还是跑在8080端口上，通过nginx代理到443端口即可，由于我们上面启动solo时添加了 `--rm` 数，只需要 `docker stop solo`即可自动删除solo容器，然后我们重新启动一个solo容器

```
docker run --detach --name solo --network=host \  
--env RUNTIME_DB="MYSQL" \  
--env JDBC_USERNAME="root" \  
--env JDBC_PASSWORD="123123" \  
--env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \  
--env JDBC_URL="jdbc:mysql://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC" \  
b3log/solo --listen_port=8080 --server_scheme=https --server_host=www.jinjianh.com --server_port=
```

- `--server_scheme=http`换成 `--server_scheme=https`即可
- `--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

然后我们去配置nginx配置文件，实现nginx反向代理

```
cd /dockerData/nginx/conf/conf.d  
vim default.conf  
location / {  
    proxy_pass http://backend$request_uri;  
    proxy_set_header Host $http_host;  
    proxy_set_header X-Real-IP $remote_addr;  
    #root /usr/share/nginx/html;  
    #index index.html index.htm;  
}
```

替换上面部分即可

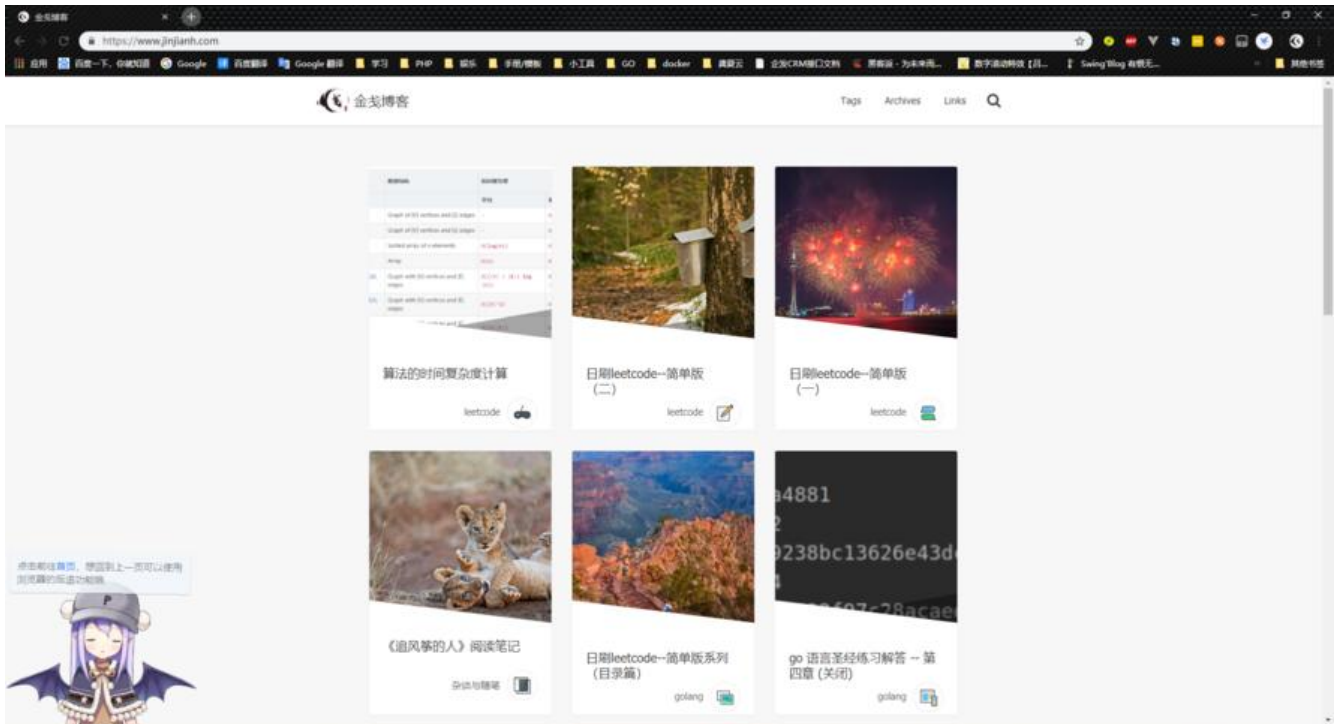
按esc，然后输入:wq保持退出

注意!!! Nginx反代理上面的方式可能出现问题参考Ngin反代

注意!!! Nginx反代理上面的方式可能出现问题参考Ngin反代

注意!!! Nginx反代理上面的方式可能出现问题参考Ngin反代

重启nginx, docker restart nginx



完美

11. 后记

皮肤挂载

- 在你的服务器上创建一个目录用于存放皮肤，比如我的 `/dockerData/solo/skins/`
- 然后将你要挂载的皮肤放到上面那个目录下
- 最后删除当前容器 重新启动一个容器，添加参数 `--v /dockerData/solo/skins/:/opt/solo/skins/` 这个添加时要注意位置，要添加到 `b3log/solo --listen...` 的上面一排
- 使用挂载皮肤时，默认会使用 `Pingsu`,

皮肤推荐

我开源了两款皮肤 `solo-nexmoe`，因为我很懒的原因，`solo-star`没有手机端，所以你可以多挂载一皮肤们比如官方皮肤 `Pinghsu`，如果你没有这款皮肤就会报错，没有请前往 `solo-skins` 下载

数据库占用内存过大优化

由于我们购买的服务器是内存只有1G，然后docker安装的mysql虽然很快，但是实际上占用内存非大，之前服务器在腾讯云的时候就经常挂掉，排查了很久才发现是docker下mysql的问题，迁移到阿云后倒是没出先挂掉的问题，但是服务器内存占用也一直在90%以上，所以我们对mysql容器进行一优化。

由于容器内不能vim，所以我们将mysql的配置文件复制到服务器上改了之后再复制回去，也可以将置文件挂载到服务器上，过程我不多讲，只讲核心部分。

这里注意，如果你要删除容器重新挂载的话，请提前备份mysql数据，不然你就属于删库了
这里注意，如果你要删除容器重新挂载的话，请提前备份mysql数据，不然你就属于删库了
这里注意，如果你要删除容器重新挂载的话，请提前备份mysql数据，不然你就属于删库了
重要的话说三遍

在配置文件 `/etc/mysql/mysql.conf.d/mysqld.cnf`中添加

```
performance_schema_max_table_instances=400  
table_definition_cache=400  
table_open_cache=256
```

从容器中复制到服务器

```
docker cp mysql:/etc/mysql/mysql.conf.d/mysqld.cnf /dockerData/mysql
```

从服务器复制到容器

```
docker cp /dockerData/mysql/mysql.conf.d/mysqld.cnf mysql:/etc/mysql/mysql.conf.d/mysqld.cnf
```

改完之后记得重启mysql, `docker restart mysql`

启用lute

此内容适用于solo3.6.5+

- 启动Lute参考 [Lute HTTP 使用指南](#)
- 在solo启动参数末尾追加 `--lute_http=http://127.0.0.1:8249/--lute_http=http://localhost:8249/-lute_http=`

solo成功启动后在终端输入 `docker logs solo`,日志显示有 `luteAvailable=true`即表示启用lute成功