



链滴

宝塔部署最新版 Solo 博客

作者: [Jireh](#)

原文链接: <https://ld246.com/article/1605842717376>

来源网站: [链滴](#)

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

宝塔部署最新版Solo博客,并每天自动更新升级。

基础环境

- 宝塔
- Mysql
- Docker

以上为基础环境，怎么搭建基础环境就不再赘述了

新建站点、数据库

添加网站

域名:

备注:

根目录:

FTP:

数据库:

数据库设置: 用户名 密码 创建站点的同时，为站点创建一个对应的数据库帐户，方便不同站点使用不同数据库。

程序类型:

PHP版本:

网站分类:

在网站中新建站点，填入域名，选择创建数据库。



配置计划任务



在计划任务中，添加新的计划任务，选择Shell脚本，执行周期按照自己的需求更改，我这里是每天然后在脚本内容中填入**如下配置**：

```
#!/bin/bash
```

```
#  
# Solo docker 更新重启脚本  
#  
# 1. 请注意修改参数  
# 2. 可将该脚本加入 crontab，每日凌晨运行来实现自动更新  
#
```

```
restart_solo(){  
    docker stop solo  
    docker rm solo  
    docker run --detach --name solo --network=host \  
    --env RUNTIME_DB="MYSQL" \  
    --env JDBC_USERNAME="you mysql user" \  
    --env JDBC_PASSWORD="you mysql user password" \  
    --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=https --server_host=you domain  
}
```

```

update_solo(){
  echo "Pulling Solo's image"
  isUpdate=$(docker pull b3log/solo | grep "Downloaded")
  if [[ -z $isUpdate ]]
  then
    echo "Solo is up to date"
  else
    restart_solo >> /dev/null 2>&1
    echo "Restarted Solo"
  fi
}

# 检查当前容器状态，如果状态正常进行升级操作，否则重新进行部署
update_and_test_service(){
  isCrash=$(docker ps | grep "b3log/solo")
  if [[ -z $isCrash ]]
  then
    echo "Solo's status is unexpected, trying to restart it"
    docker pull b3log/solo
    restart_solo
    sleep 5
    isSecondCrash=$(docker ps | grep "b3log/solo")
    if [[ -z $isSecondCrash ]]
    then
      echo "Failed to restart Solo, please check logs via 'docker logs solo'"
    fi
  else
    update_solo
  fi
}

update_and_test_service

```

注意修改以上配置中的数据库相关配置为自己的配置，端口8080如果占用，请修改成其他的，server host中填入自己的域名

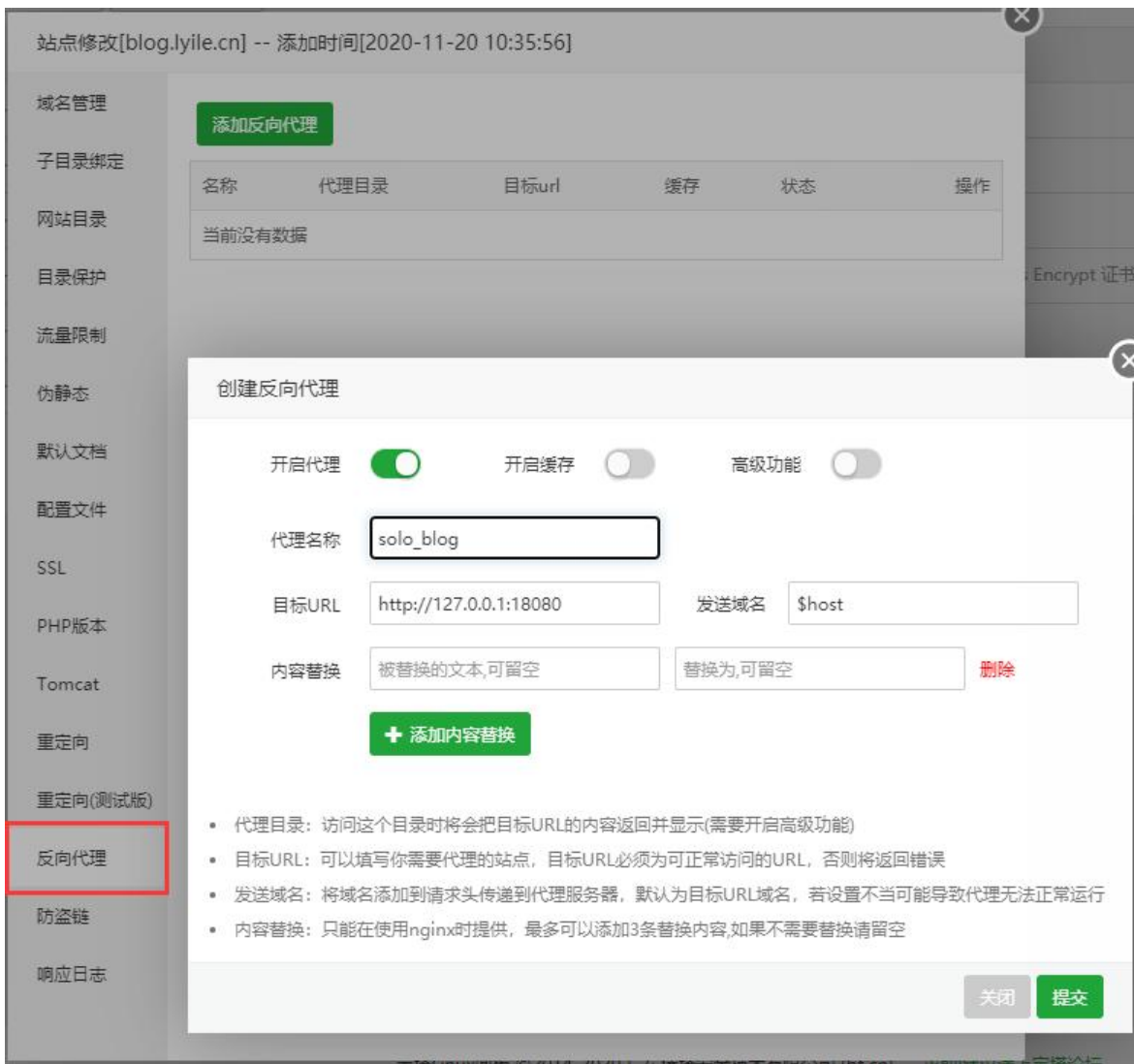
配置Nginx

配置HTTPS/SSL



配置反代

如果需要配置https的，首先配置https证书，否则会有冲突。



新建反向代理，按照以上示例填写。

完整视频教程

<https://www.bilibili.com/video/BV1uK4y1j7gi/>