



链滴

Solo 博客搭建

作者: [adongs](#)

原文链接: <https://ld246.com/article/1563518678911>

来源网站: [链滴](#)

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

服务器

1. 申请云服务器(这里笔者选择腾讯云,使用学生套餐便宜)



The screenshot shows a promotional page for Tencent Cloud's 'Cloud Server Discount Package'. It highlights a special offer of 10 RMB/month (down from 116.5 RMB/month) for a 6-month period. The package includes 1 core, 2GB RAM, 1M bandwidth, and 50GB free object storage. There are two '前往管理' (Go to management) buttons for the server and storage options.

2. 选择系统(这里笔者建议使用自己熟悉的系统,如果你不熟悉linux可以跟随笔者选择的系统)



3. 创建云服务器登录方式选择密码方式(方便上传文件)

4. 配置创建好的云服务器安全组,配置如下



The screenshot displays the configuration for a security group, divided into 'Inbound Rules' (入站规则) and 'Outbound Rules' (出站规则). The Inbound Rules table lists four rules for allowing traffic on ports 8080, 80, 443, and 22. The Outbound Rules table shows a single rule allowing all traffic.

来源	协议端口	策略	备注	操作
<input type="checkbox"/> 0.0.0.0/0	TCP:8080	允许	-	编辑 插入 删除
<input type="checkbox"/> 0.0.0.0/0	TCP:80	允许	放通Web服务HTTP (80), 如 Apache, Nginx	编辑 插入 删除
<input type="checkbox"/> 0.0.0.0/0	TCP:443	允许	放通Web服务HTTPS (443), 如 Apache, N...	编辑 插入 删除
<input type="checkbox"/> 0.0.0.0/0	TCP:22	允许	放通Linux SSH登录	编辑 插入 删除

目标	协议端口	策略	备注	操作
<input type="checkbox"/> 0.0.0.0/0	ALL	允许	一键放通出站规则	编辑 插入 删除

申请域名

1. 查询域名



2. 购买域名(购买完域名,需要备案才能解析和访问)
3. 备案域名(备案周期在30天左右)

安装服务

1. 登录服务器(mac直接打开终端,Windows系统需要安装软件Secure CRT)

```
ssh -p 22 系统用户名@公网ip  
# 例如  
ssh -p 22 ubuntu@10.22.33.49
```

2. 安装docker

```
sudo apt-get install docker.io
```

3. 安装nginx

```
sudo apt-get install nginx
```

4. 安装mycli(命令行操作mysql的智能提示工具)

```
sudo apt-get install mycli
```

5. 安装vim

```
sudo apt-get install vim
```

6. 使用docker安装solo和mysql

```
# 下载mysql 镜像  
sudo docker pull mysql  
# 启动镜像 MYSQL_ROOT_PASSWORD=mysql的root密码  
docker run -d --name mysql-5.7 --network=host -v /var/lib/mysql:/var/lib/mysql -e MYSQL_  
OOT_PASSWORD=root mysql:5.7 --character-set-server=utf8mb4 --collation-server=utf8mb  
_general_ci  
# 登录mysql  
mycli  
# 如果登录提示如下错误
```

```
Traceback (most recent call last):
  File "/usr/bin/mycli", line 9, in <module>
    load_entry_point('mycli==1.5.2', 'console_scripts', 'console')()
  File "/usr/lib/python3/dist-packages/click/core.py", line 716, in __call__
    return self.main(*args, **kwargs)
  File "/usr/lib/python3/dist-packages/click/core.py", line 675, in main
    _verify_python3_env()
  File "/usr/lib/python3/dist-packages/click/_unicodefun.py", line 119, in _verify_python3_env
    'mitigation steps.' + extra)
RuntimeError: Click will abort further execution because Python 3 was configured to use ASCII as encoding for the environment. Either run this under Python 2 or consult http://click.pocoo.org/python3/ for mitigation steps.

This system supports the C.UTF-8 locale which is recommended.
You might be able to resolve your issue by exporting the
following environment variables:

    export LC_ALL=C.UTF-8
    export LANG=C.UTF-8
```

```
# 执行如下即可
export LC_ALL=C.UTF-8
export LANG=C.UTF-8
# 创建数据库
CREATE DATABASE solo DEFAULT CHARACTER SET utf8mb4 DEFAULT COLLATE utf8mb4_geral_ci
# 退出mycli工具
quit
```

```
# 下载solo博客镜像
sudo docker pull b3log/solo
```

```
# 启动solo镜像
# JDBC_PASSWORD(mysql的密码)
# listen_port(启动端口)
# server_scheme(访问协议)
# server_host(请求的地址)
# server_port(请求的端口)
docker run --detach --name solo --network=host \
  --env RUNTIME_DB="MYSQL" \
  --env JDBC_USERNAME="root" \
  --env JDBC_PASSWORD="root" \
  --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=adongs.com --server_port=80
# 查看启动结果
sudo docker ps
# 结果如下就表示启动成功(STATUS 都为UP表示启动成功)
```

```
ubuntu@VM-0-7-ubuntu:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f13e78072a3f	b3log/solo	"java -cp WEB-INF/li..."	3 hours ago	Up 3 hours		solo
4d34d9eb9b32	mysql:5.7	"docker-entrypoint.s..."	23 hours ago	Up 23 hours		mysql-5.7

配置域名和代理

1. 申请SSL(笔者在腾讯云申请的免费SSL, [第三方免费SSL](#))

1、【重要通知】由于CA机构和证书代理商策略调整，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）。若您的业务因此次调整受做，建议您购买泛域名SSL证书。
 2、近期由于赛门铁克SSL证书颁发系统升级，域名敏感词库进行更新调整，可能导致部分原来可正常颁发的DV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，若您忘记密码则需要重新申请证书

所属项目

[下一步](#)

2. 下载SSL到本地,并解压

```
→ adongs.com ll
total 8
drwxr-xr-x 5 yudong staff 160B 8 12 15:04 Apache
drwxr-xr-x 4 yudong staff 128B 8 12 15:04 IIS
drwxr-xr-x 4 yudong staff 128B 8 12 15:04 Nginx
drwxr-xr-x 4 yudong staff 128B 8 12 15:04 Tomcat
-rwxr-xr-x@ 1 yudong staff 996B 7 18 13:16 adongs.com.csr
```

3. 上传SSL到云服务器

scp -r SSL文件目录 系统用户名@公网i:/home/系统用户名

4. 配置nginx

#登录云服务器,执行如下

sudo vim /etc/nginx/sites-enabled/default

#将文本里面的内容全删除,拷贝如下文本内容

#修改server_name为自己的域名

#修改ssl_certificate和ssl_certificate_key的文件路径

#修改proxy_pass为自己的域名和solo启动端口

```
server {
    listen 443 ssl;
    server_name www.xxx.com,xxx.com;
    ssl_certificate /etc/nginx/ssl/xxx.crt;
    ssl_certificate_key /etc/nginx/ssl/xxx.key;

    location / {
        proxy_pass http://www.xxx.com:8080;
    }
}
```

```
server {
    listen 80;
    listen [::]:80;
    server_name www.xxx.com,xxx.com;
    return 301 https://$host$request_uri;
}
```

保存并退出后重启nginx

nginx -s reload

调试

1. 打开自己的域名如下



配置自动更新

1. 创更新脚本

docker-restart-solo.sh

```
#
# Solo docker Update and restart scripts
#
# The script is added to the crontab
#
# 注意要修改配置
# JDBC_PASSWORD(mysql的密码)
# JDBC_URL(mysql连接地址)
# listen_port(启动端口)
# server_scheme(访问协议)
# server_host(请求的地址)
# server_port(请求的端口)
#
docker pull b3log/solo
docker stop solo
docker rm solo
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" \
```

```
b3log/solo --listen_port=8080 --server_scheme=https --server_host=adongs.com --server_port=80
```

2.修改脚本权限

```
#最后是你的脚本绝对路径  
sudo chmod 744 /opt/docker-restart-solo.sh
```

3.加入crontab

```
# ubuntu默认没有开启cron日志记录,需要开启  
# 修改rsyslog  
sudo vim /etc/rsyslog.d/50-default.conf  
#将cron前面的注释符去掉  
cron.* /var/log/cron.log  
#编辑crontab  
crontab -e  
#在最后一行加入如下内容(最后是你脚本的绝对路径),每天凌晨3点执行脚本  
* 3 * * * /opt/docker-restart-solo.sh  
# control + X 退出,选择Y保存,查看定时任务  
crontab -l
```

相关资料

1. [solo地址](#)
2. [mysql镜像说明](#)
3. [腾讯云学生购买](#)

感谢

B3log 开源社区

问题

安装有什么问题请留言