

# 使用docker来部署nevinee/jd的jd签到脚本

写这个教程是为了之后重装系统时候，可以用来参考一下，以防自己忘了。现在网上还是挺少这个详细的教程的。本教程仅供交流参考，如果使用本教程搭建服务之后出现封号什么的，不要来找我。是你自己的责任。

下面开始

nevinee/jd的脚本发布页是：<https://hub.docker.com/r/nevinee/jd> <<https://hub.docker.com/r/nevinee/jd>> 这个脚本的原作者是lxx0301大佬，因为某些人为了自己的利益导致这位大佬的仓库被迫转为私有仓库。而nevinee是少数经lxx0301大佬授权可以发布在自己仓库的大佬之一。本教程主要还是以lxx0301大佬的脚本为例，来介绍jd签到脚本的安装步骤，以及安装控制面板，因为lxx0301大佬自己的脚本是不带控制面板的。我安装时候也是网上找了好久也没找到教程，最后是群里一位大佬帮忙解决的。

这里演示用openwrt系统内的docker为例，其他系统应该都差不多。

如果你的openwrt没有安装的话，自行百度安装

1、确认你系统的docker已经打开了



2、使用ssh工具连接你安装docker的机器



3、鼠标右键复制以下全部命令，然后在命令框内鼠标右键点一下黏贴命令

```
docker run -dit \  
-v /root/jd/config:/jd/config \  
-v /root/jd/jd/log:/jd/log \  
-v /root/jd/own:/jd/own \  
-v /root/jd/scripts:/jd/scripts \  
-e ENABLE_HANGUP=true \  
-e ENABLE_TG_BOT=true \  
-e ENABLE_WEB_PANEL=true \  
-p 5678:5678 \  

```

```
--name jd \  
--hostname jd \  
--restart always \  
nevinee/jd:v4-bot
```

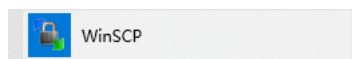
```
> --hostname jd \  
> --restart always \  
> nevinee/jd:v4-bot  
  
Unable to find image 'nevinee/jd:v4-bot' locally  
  
v4-bot: Pulling from nevinee/jd  
535256e01bcb: Extracting [=====] 2.727MB/2.727MB  
854e953d7d35: Download complete  
15d93b3801df: Download complete  
0e67d37e7e22: Download complete  
72ebee66b215: Download complete  
375170a2e3b0: Downloading [==>] 1.407MB/27.5MB  
375170a2e3b0: Downloading [=====] 4.787MB/27.5MB  
375170a2e3b0: Download complete  
46b685054e55: Download complete  
[
```

等待它镜像拉去完成，如果提示错误多试几次，还是不行就自行百度吧!

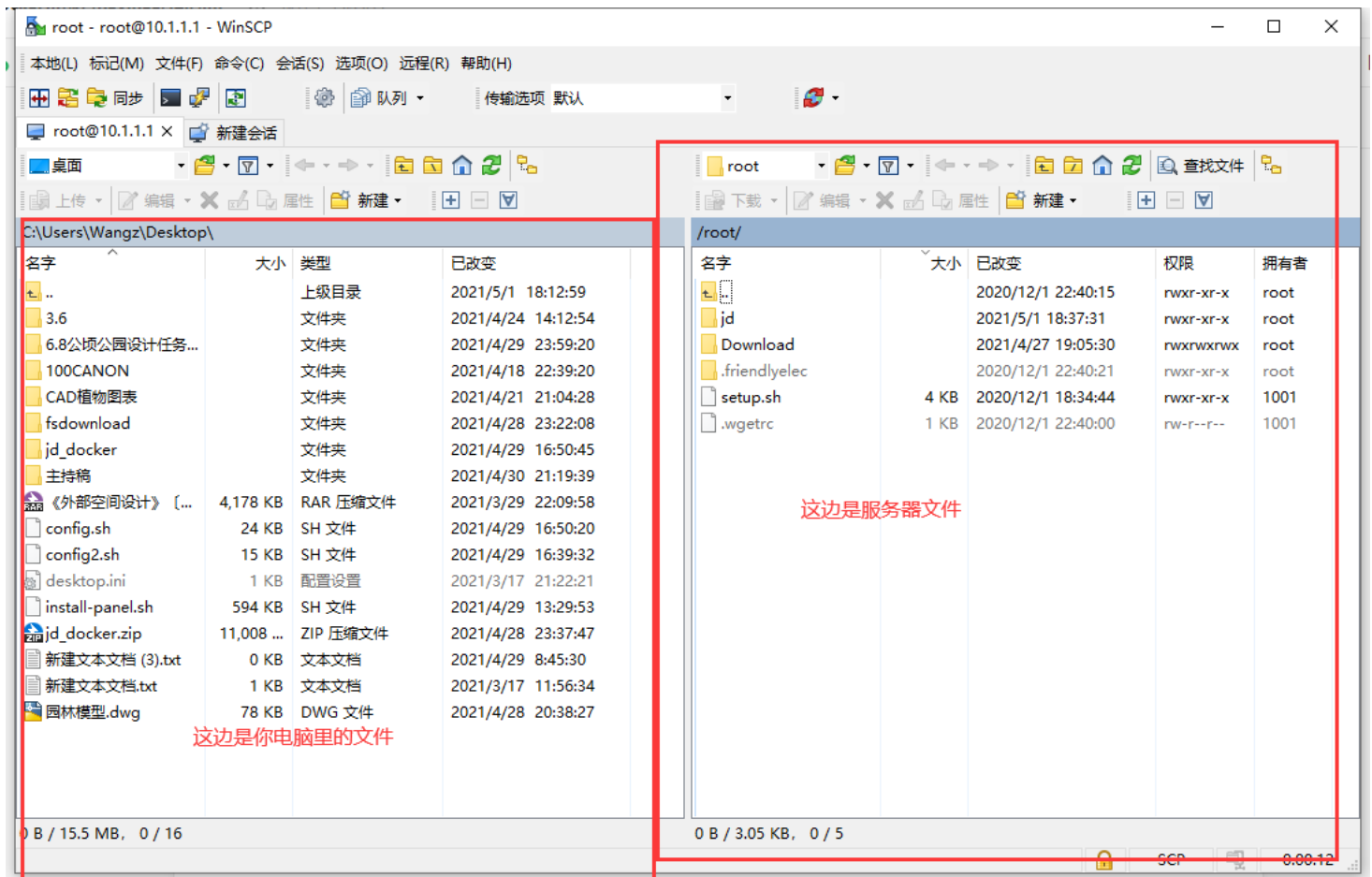
```
> -p 5678:5678 \  
> --name jd \  
> --hostname jd \  
> --restart always \  
> nevinee/jd:v4-bot  
-ash: docker run -dit -v /root/jd/config:/jd/config -v /root/jd/jd/log:/jd/log -v /root/jd/own:/jd/own -v /root/jd/scripts:/jd/scripts -e ENABLE_HANGUP=true -e ENABLE_TG_BOT=true -e ENABLE_WEB_PANEL=true -p 5678:5678 --name jd --hostname jd --restart always nevinee/jd:v4-bot: not found  
root@xiaoming:~# docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               N  
AMES  
1e3ac701e5cf        nevinee/jd:v4-bot  "docker-entrypoint.sh"  56 minutes ago     Up 48 minutes      0.0.0.0:5678->5678/tcp  j  
d  
root@xiaoming:~# docker ps -a  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               N  
AMES  
1e3ac701e5cf        nevinee/jd:v4-bot  "docker-entrypoint.sh"  56 minutes ago     Up 48 minutes      0.0.0.0:5678->5678/tcp  j  
d  
root@xiaoming:~#
```

输入docker ps -a可以查看当前已经部署的容器，可以看到jd签到脚本已经安装好了。

4、然后用winSCP这个软件登入到openwrt里，这个软件是用来给服务器等机器来传输管理文件用的，这里用它来传输一个脚本，用来安装控制面板

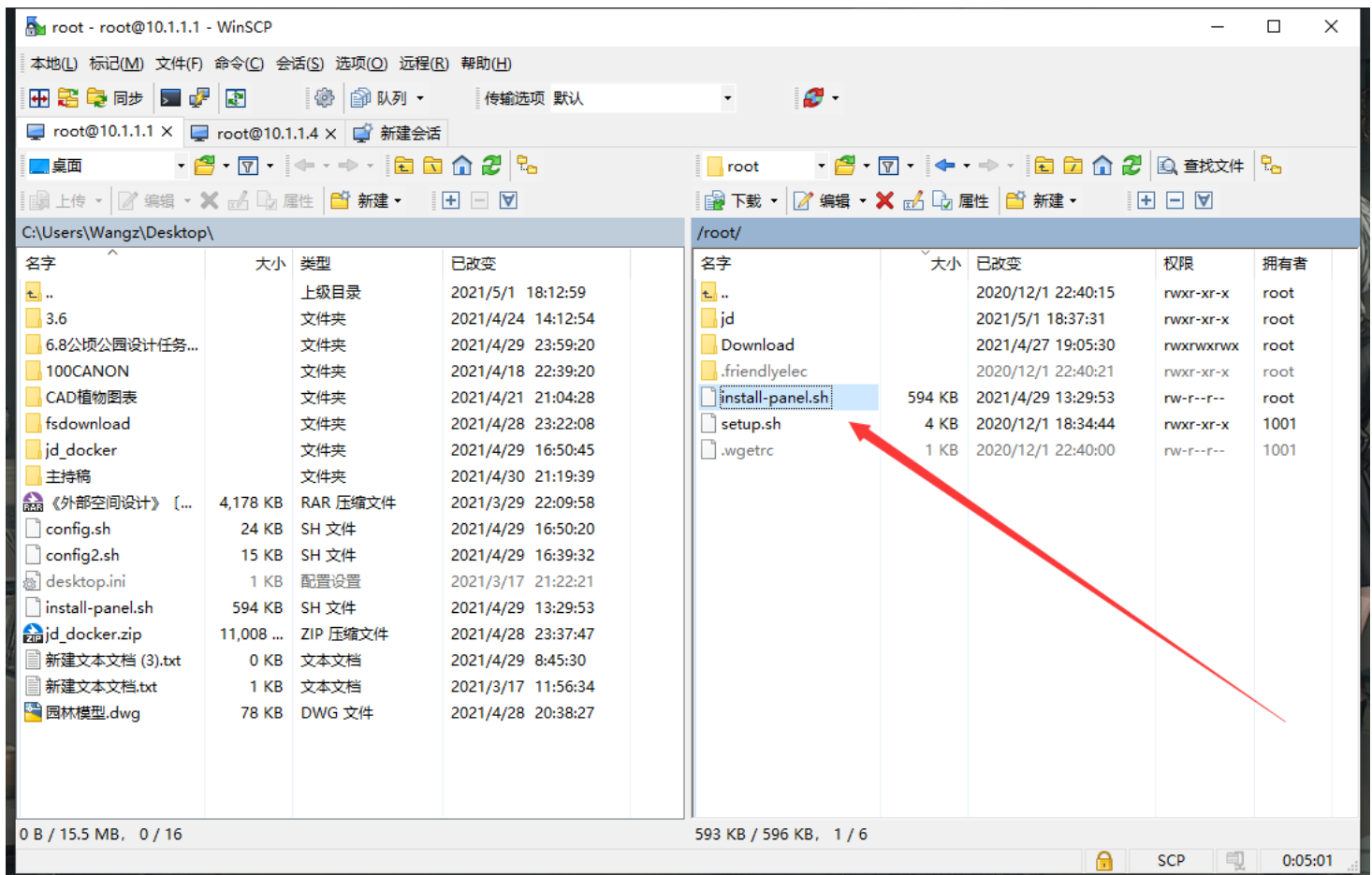


如果你没有用过这个软件的话安装好打开应该是只有一个新建站点。你只要点击新建站点，文件协议选择SCP，然后输入你的openwrt系统ssh的用户名和密码登录就好了



可以看到openwrt里面有一个叫jd的文件夹，这个就是我们jd签到脚本的配置文件。

5、然后把脚本文件上传到有root这个文件夹（一般默认在root这个文件夹）



6、然后我们执行以下命令

```
docker cp /root/install-panel.sh jd:/jd
```

这条命令的意思是把install-panel.sh这个文件复制到容器内部。如果你的文件不是在这个地方，那你就自己百度Linux复制文件命令怎么用，命令前面加docker。

```
> -p 5678:5678 \
> --name jd \
> --hostname jd \
> --restart always \
> nevinee/jd:v4-bot
-ash: docker run -dit -v /root/jd/config:/jd/config -v /root/jd/jd/log:/jd/log -v /root/jd/own:/jd/own -v /root/jd/scripts:/jd/scripts -e ENABLE_HANGUP=true -e ENABLE_TG_BOT=true -e ENABLE_WEB_PANEL=true -p 5678:5678 --name jd --hostname jd --restart always nevinee/jd:v4-bot: not found
root@xiaoming:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1e3ac701e5cf       nevinee/jd:v4-bot  "docker-entrypoint.sh" 56 minutes ago      Up 48 minutes      0.0.0.0:5678->5678/tcp  jd
root@xiaoming:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1e3ac701e5cf       nevinee/jd:v4-bot  "docker-entrypoint.sh" 56 minutes ago      Up 48 minutes      0.0.0.0:5678->5678/tcp  jd
root@xiaoming:~# docker cp /root/install-panel.sh jd:/jd
```

7、接着我们再执行以下命令，进入容器内部。

docker exec -it jd /bin/bash

```
> --hostname jd \
> --restart always \
> nevinee/jd:v4-bot
-ash: docker run -dit -v /root/jd/config:/jd/config -v /root/jd/jd/log:/jd/log -v /root/jd/own:/jd/own -v /root/jd/scripts:/jd/scripts -e ENABLE_HANGUP=true -e ENABLE_TG_BOT=true -e ENABLE_WEB_PANEL=true -p 5678:5678 --name jd --hostname jd --restart always nevinee/jd:v4-bot: not found
root@xiaoming:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1e3ac701e5cf       nevinee/jd:v4-bot  "docker-entrypoint.sh" 56 minutes ago      Up 48 minutes      0.0.0.0:5678->5678/tcp  jd
root@xiaoming:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1e3ac701e5cf       nevinee/jd:v4-bot  "docker-entrypoint.sh" 56 minutes ago      Up 48 minutes      0.0.0.0:5678->5678/tcp  jd
root@xiaoming:~# docker cp /root/install-panel.sh jd:/jd
root@xiaoming:~# docker exec -it jd /bin/bash
root@jd:/jd $
```

8、然后我们输入ls命令来查看我们有没有把这个文件传到容器内部

```
-ash: docker run -dit -v /root/jd/config:/jd/config -v /root/jd/jd/log:/jd/log -v /root/jd/own:/jd/own -v /root/jd/scripts:/jd/scripts -e ENABLE_HANGUP=true -e ENABLE_TG_BOT=true -e ENABLE_WEB_PANEL=true -p 5678:5678 --name jd --hostname jd --restart always nevinee/jd:v4-bot: not found
root@xiaoming:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1e3ac701e5cf       nevinee/jd:v4-bot  "docker-entrypoint.sh" 56 minutes ago      Up 48 minutes      0.0.0.0:5678->5678/tcp  jd
root@xiaoming:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1e3ac701e5cf       nevinee/jd:v4-bot  "docker-entrypoint.sh" 56 minutes ago      Up 48 minutes      0.0.0.0:5678->5678/tcp  jd
root@xiaoming:~# docker cp /root/install-panel.sh jd:/jd
root@xiaoming:~# docker exec -it jd /bin/bash
root@jd:/jd $ ls
config  install-panel.sh  jcode.sh  jlog.sh  jtask.sh  log      notify.js  own      sample
docker  jbot              jcsv.sh   jshare.sh  jup.sh    mtask.sh  otask.sh   readme.md  scripts
root@jd:/jd $
```

9、接着我们输入下面命令来运行这个脚本安装控制面板

bash install-panel.sh

当显示以下画面时候就是表示面板安装成功。用户名是：admin 密码是：adminadmin

在你的浏览器里输入：你路由器或者服务器ip加端口5678

```
[PM2][WARN] Applications server not running, starting...
[PM2] App [server] launched (1 instances)
```

id	name	namespace	version	mode	pid	uptime	u	status	cpu	mem	user	watching
1	server	default	1.0.0	fork	693	0s	0	online	0%	28.4mb	root	enabled
0	ttyd	default	N/A	fork	683	2s	0	online	0%	16.6mb	root	disabled

控制面板启动成功，如未修改，则初始用户名和密码为：admin/adminadmin...

root@jd:/jd \$

### 控制面板

提交

10、浏览器打开输入用户名和密码后就可以登录了，这个就是京东脚本的控制面板了，然后跟着注释去编辑内容就可以了。主要需要填的是Cookie和通知，Cookie可以通过点击扫码获取Cookie。用你的京东app扫码，它会自动替换，如果没有自动替换就自己手动填到Cookie里就可以了。假设你有多个京东账户要签到话，就在Cookie2下面加上Cookie3=""、Cookie=""等等，再把扫码获取到的Cookie信息填到引号内。

保存 切换换行 扫码获取 Cookie ☒ 扫码后自动替换

55 7 93 1.0 1.1 22

---

只要用微信扫码关注，然后点一对一推送就可以看到你的token，这个是我们用来推送通知的

## pushplus（推送加）是什么？

pushplus(推送加)集成了微信、企业微信、钉钉、短信、邮件等实时信息推送平台  
你只需要调用简单的API，即可帮助你迅速完成消息的推送，使用简单方便  
我们的所做的一切只是为了让推送变的更简单



### 你的token

 [更换token](#)

⚠ token很重要，请不要告诉别人！

### 试一试

消息标题	<input type="text" value="请输入标题"/>
消息内容	<input type="text" value="请输入内容"/>
消息模板	<div>html</div>
发送渠道	<div>微信公众号</div>
<a href="#">发送消息</a>	

2、然后 在config.sh里第二区域加上一行，这个就是表示用pushplus来推送。把你在pushplus获取到的token填到引号内。然后点击保存，这样你的jd签到脚本基本就部署完了。

```
export PUSH_PLUS_TOKEN=""
```







### 手动执行脚本

切换执行

跳转底部

列出脚本

更新脚本文件

删除旧日志

获取互助码

导出互助码

查看进程

查看挂机日志

重启挂机程序

重置用户名密码

jtask

jd\_bean\_change.js

now

提交

### 运行日志

```
15
16 2021-05-02日过期京豆：50
17
18 2021-05-03日过期京豆：50
19
20 2021-05-04日过期京豆：0
21
22 2021-05-05日过期京豆：0
23
24 2021-05-06日过期京豆：0
25
26 2021-05-07日过期京豆：0
27
28
29 =====系统通知=====
30 京东资产变动通知
31 账号1
32 昨日收入：100京豆
33 昨日支出：-100京豆
34 当前京豆： (今日将过期50)京豆
35 当前总红包：2.73(今日总过期0.30)元
36 京喜红包：1.56(今日将过期0.21)元
37 极速版红包：0.01(今日将过期0.00)元
38 京东红包：1.16(今日将过期0.09)元
39 健康红包：0.00(今日将过期0.00)元
40
41
42 您未提供server酱的SCKEY，取消微信推送消息通知
43
44 push+发送一对一通知消息完成。
45
46 您未提供Bark的APP推送BARK_PUSH，取消Bark推送消息通知
47
48 您未提供telegram机器人推送所需的TG_BOT_TOKEN和TG_USER_ID，取消telegram推送消息通知
49
50 您未提供钉钉机器人推送所需的DD_BOT_TOKEN或者DD_BOT_SECRET，取消钉钉推送消息通知
51
52 您未提供企业微信机器人推送所需的QYWX_KEY，取消企业微信推送消息通知
53
54 您未提供企业微信应用消息推送所需的QYWX_AM，取消企业微信应用消息推送消息通知
55
56 您未提供iGot的推送IGOT_PUSH_KEY，取消iGot推送消息通知
57
58
59 📢 京东资产变动通知，结束! ⌚ 2.969 秒
60
61 |
```



## 删除失效任务通知

通知内容: 成功删除以下失效的定时任务  
(jd\_scripts脚本): jd\_mohe<b...

通知时间: 2021-05-01 19:37:26

备注: 点击查看消息,退订请回复[退订]

查看详情



晚上8:37

## 设备通知



### 京东资产变动通知

通知内容: 账号1: H... 账号2: ...  
昨日支...

通知时间: 2021-05-01 20:37:11

备注: 点击查看消息,退订请回复[退订]

查看详情



≡ pushplus

激活消息

≡ 功能

## 结语

这就是这个教程的全部内容了，已经讲的很详细了，再不会的话那就只能付费用那些二道贩子的一键脚本了。如果你要玩那些小游戏的话，要自己去京东里面挨个把小游戏打开一下。感谢那些大佬的辛苦付出，我们用就用不要到处宣传。毕竟我们这么做也是在薅京东羊毛，万一把京东薅疼了把这个给关了。大家都没得玩了，所以珍惜把。最后，config.sh建议用我给的，通知方式多一些。直接把下载的文件拖到jd文件夹里的config文件内替换，当然在线编辑工具里面全选替换也是可以的，然后执行重启容器命令，等重启完成去浏览器就可以看到更新后的内容。

```
docker restart jd
```

文件链接: [https://pan.baidu.com/s/1XObI9m\\_CdhGdAJUFMZ-HpA](https://pan.baidu.com/s/1XObI9m_CdhGdAJUFMZ-HpA) <[https://pan.baidu.com/s/1XObI9m\\_CdhGdAJUFMZ-HpA](https://pan.baidu.com/s/1XObI9m_CdhGdAJUFMZ-HpA)> 提取码: bxpv

文件链接:<https://drive.google.com/file/d/18enFAbS6KLSPIPWgnr1udw48KeMOCTHx/view?usp=sharing>

<https://drive.google.com/file/d/18enFABs6KLSPIPWgnr1udw48KeMOCTHx/view?usp=sharing>

文件链接:<https://www.lanzous.com/iU8MNopdqtc> <<https://www.lanzous.com/iU8MNopdqtc>> 密码:1m32

(我就不信全部完蛋)

配置设置

定时设置

对比工具

自定义脚本

手动执行

终端

日志查询

查看脚本

修改密码

退出

在线编辑工具

保存

切换换行

扫码获取 Cookie

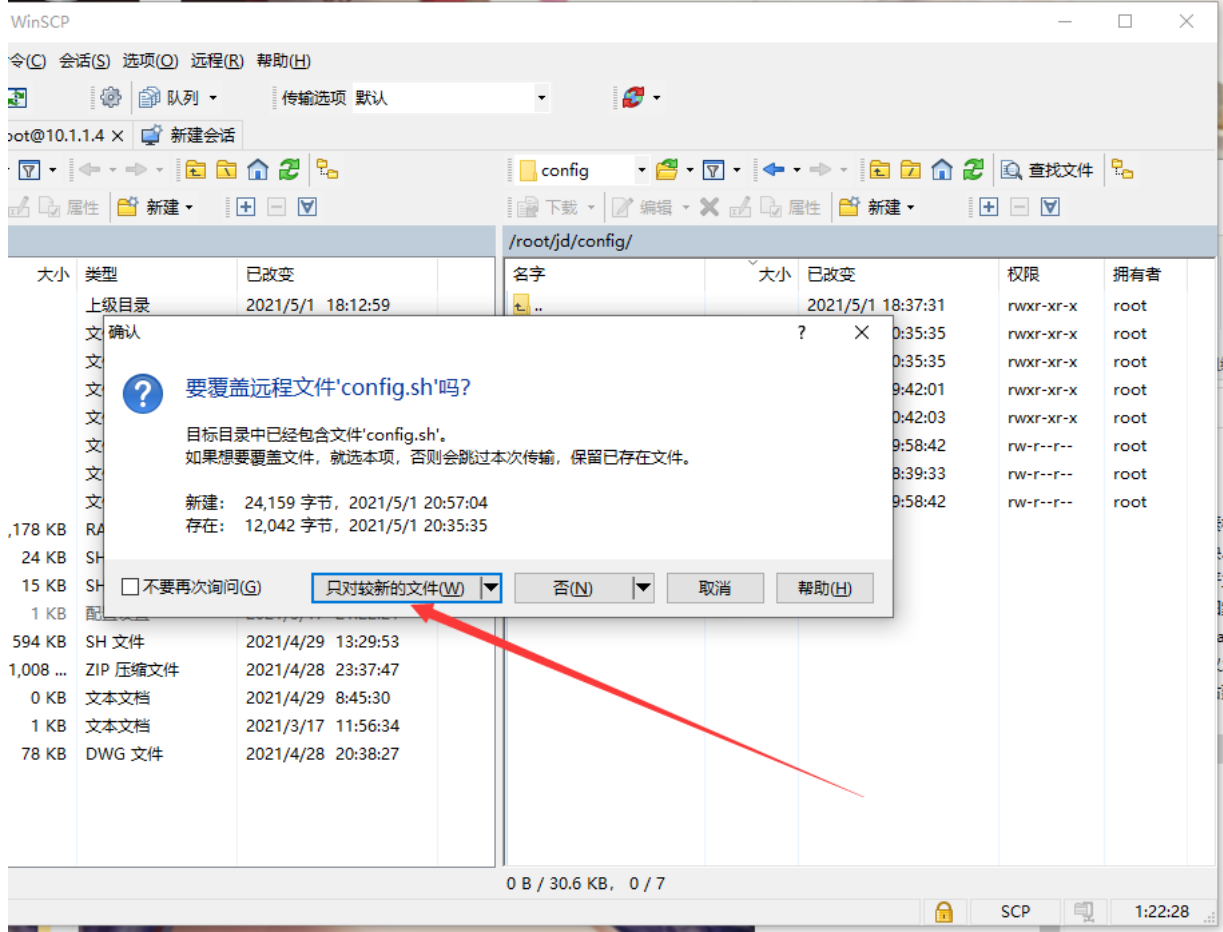
☒ 扫码后自动替换

config.sh

```

17 # 商品名称 export 商品名称 为商品名称 export
18 goodPageSize="350"    ## 商品取关数量
19 zshopPageSize="350"   ## 店铺取关数量
20 jdUnsubscribeStopGood="" ## 遇到此商品不再取关此商品以及它后面的商品，需去商品详情页长按拷贝商品信息
21 jdUnsubscribeStopShop="" ## 遇到此店铺不再取关此店铺以及它后面的店铺，请从头开始输入店铺名称
22
23 ##### 第一区域：jd_scripts脚本特有变量填写区域（需要shell转换的） #####
24
25
26
27 ##### 第二区域：jd_scripts特有变量填写区域（不需要shell转换的） #####
28
29 # 请在本区域补充其他你需要用到的变量，export 变量名“变量值”，或：export 变量名=‘变量值’
30 # 其他变量详见：https://gitee.com/lxk0301/jd_docker/blob/master/githubAction.md
31 # 该链接中称ID_COOKIE、JD_BEAN_SIGN_STOP_NOTIFY、JD_BEAN_SIGN_NOTIFY_SIMPLE、UN_SUBSCRIBES这四个变量以及所有互助码类变量外，其他所有变量请在本区域自行补充
32
33 ##### 定义通知TOKEN（选项） #####
34 # 想通过什么渠道收取通知，就填入对应渠道的值
35 # 1. ServerChan，教程：https://sc.ftqq.com/3.version
36 export PUSH_KEY=""
37
38 # 2. BARK，教程（看BARK_PUSH和BARK_SOUND的说明）：https://github.com/lxk0301/jd_scripts/blob/master/githubAction.md
39 export BARK_PUSH=""
40 export BARK_SOUND=""
41
42 # 3. Telegram，如需使用，TG_BOT_TOKEN和TG_USER_ID必须同时赋值，教程：https://gitee.com/lxk0301/jd_scripts/blob/master/githubAction.md
43 export TG_BOT_TOKEN=""
44 export TG_USER_ID=""
45
46 # 4. 钉钉，教程（看DD_BOT_TOKEN和DD_BOT_SECRET部分）：https://gitee.com/lxk0301/jd_scripts/blob/master/githubAction.md
47 export DD_BOT_TOKEN=""
48 export DD_BOT_SECRET=""
49
50 # 5. iGot聚合推送，支持多种方式推送，填写iGot的推送key，教程：https://gitee.com/lxk0301/jd_scripts/blob/master/githubAction.md
51 export IGOI_PUSH_KEY=""
52
53 # 6. Push Plus，微信扫码登录后一对一推送或一对多推送，参考文档：http://pushplus.hxtrip.com/
54 # 其中PUSH_PLUS_USER是一对多推送的“群组标识”（一对多推送下面->您的群组(如无则新建)->群组编码)注：(1、需订阅者扫描二维码 2、如果您是创建群组所赠人，也需点击“查看二维码”扫描绑定，否则不能接受群组消息)
55 export PUSH_PLUS_TOKEN=""
56 export PUSH_PLUS_USER=""
57
58 # 7. 企业微信机器人消息推送 webook 后面的 key，文档：https://work.weixin.qq.com/api/doc/90000/90136/91770
59 export QYWX_KEY=""
60
61 # 8. 企业微信应用消息推送的值，文档：https://work.weixin.qq.com/api/doc/90000/90136/90236
62 # 依次填上corpId的值,corpsecret的值,touser的值,agentId.media_id的值，注意用,号隔开，例如："wwoef56746d9adwers,B-791548lnrXEB6_Bwifx3kStuJr9vFEPKAbh6WERQ_mingcheng,1000001,2COXgJH2UifHERF2zxrtU0KsQ9Kx"
63 export QYWX_AM=""
64
65 #export PUSH_PLUS_TOKEN=""
66 #export PUSH_PLUS_USER=""
67 # export JD_USER_AGENT="jdapp;android;9.5.0;10.5653336693738366-7316732666536673;network/UNKNOWN,model/Redmi Note 8 Pro;addressid/4330097245;aid/e63f878f7a7bf5f7,oid/feeb78ca74813dal;osVer/29;appBuild"
68 # export QYWX_KEY="adcb27d1-28c4-4abb-8e36-d01849aee53f"
69 # export QYWX_AM="wwe5abaf787bf6eece6,rEjzdcPPFOyzBwMSRq6RyuZ3v1aSQUZICq01Bfquc,@all,1000002,2Y6JZ154W4g0iZItuWK4-vQrDbUb-QtrPFaXLmLCHR-r7q14r1TbZfHz1-OoXzX"
70
71 # export JD_BEAN_SIGN_STOP="" ## 京东多合一签到 # jd-bean-sign 是自定义签到签到，单位毫秒，默认分批并发无延迟，延迟作用于两个签到接口，如填入延迟则切换顺序签到(耗时较长)，如需填写建议输入数字1，详见

```



要覆盖远程文件'config.sh'吗?

目标目录中已经包含文件'config.sh'。  
如果想要覆盖文件，就选本项，否则会跳过本次传输，保留已存在文件。

新建: 24,159 字节, 2021/5/1 20:57:04  
存在: 12,042 字节, 2021/5/1 20:35:35

☐ 不要再次询问(G)

☒ 只对较新的文件(W)

☐ 否(N)