

Solo 通过脚本实现定时对软件更新和重启

作者: [flowerunbeaten](#)

原文链接: <https://ld246.com/article/1643186655586>

来源网站: [链滴](#)

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



准备工作

1.安装contab

依次执行以下命令

```
yum -y install vixie-cron  
yum -y install crontabs
```

查看crontab相关命令

```
//启动服务  
service crond start  
//关闭服务  
service crond stop  
//重启服务  
service crond restart  
//重新载入配置  
service crond reload  
//查看crontab服务状态  
service crond status
```

设置开机自启

```
chkconfig --level 345 crond on
```

2.准备脚本文件

脚本文件地址: <https://github.com/88250/solo/blob/master/scripts/docker-restart.sh>

```

#!/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="你自己的数据库名称" \
    --env JDBC_PASSWORD="你自己的数据库密码" \
    --env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \
    --env JDBC_URL="jdbc:mysql://localhost:3306/solo?useUnicode=yes&characterEncoding=
TF-8&useSSL=false&serverTimezone=UTC" \
    b3log/solo --listen_port=9630 --server_scheme=https --server_host=www.flowerunbea
en.top
}

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:

--listen port=9630 这里的端口号需要和你自己的Nginx反向代理的配置文件的端口号对应上, 不然动的时候会访问不了

--server_scheme=https 这里具体采用http还是https也是需要看你自己的Nginx反向代理配置文件否配置了ssl

--server_host=www.flowerunbeaten.top 这里也需要和Nginx反向代理配置文件对应上

把脚本文件上传到服务器的一个文件夹下, 上传之后需要添加该脚本的执行权限

```
chmod +x /home/cron.d/solo-docker-restart.sh
```

执行该脚本

```
./solo-docker-restart.sh
```

```
[root@iZbp1hr06u2iecm7b846cwZ cron.d]# ./solo-docker-restart.sh
Pulling Solo's image
Solo is up to date
[root@iZbp1hr06u2iecm7b846cwZ cron.d]# █
```

如图表示成功了

如果报错bash: ./a.sh: /bin/bash^M: bad interpreter: No such file or directory的话有可能是该脚的文件格式有问题。

解决办法

使用vim命令进入脚本文件

```
vim solo-docker-restart.sh
```

在文件底部输入:set ff如果显示是fileformat=dos则输入:set fileformat=unix后执行:x或者:wq保存改

然后再次执行./solo-docker-restart.sh看是否成功

3.编写一条定时任务

执行crontab -e命令, 在编辑器中输入如下命令

```
0 0 * * * /home/cron.d/solo-docker-restart.sh (每天0时0分执行该命令)
```

```
1 0 0 * * * /home/cron.d/solo-docker-restart.sh
2 0 0 * * * /home/cron.d/mysql-backup-solo.sh
```

然后保存退出即可

使用crontab -l查看定时任务是否存在

```
[root@iZbp1hr06u2iecm7b846cwZ cron.d]# crontab -l
0 0 * * * /home/cron.d/solo-docker-restart.sh
0 0 * * * /home/cron.d/mysql-backup-solo.sh
[root@iZbp1hr06u2iecm7b846cwZ cron.d]# █
```

至此每天的0时0分就会执行定时任务通过脚本检查软件更新和重启了

crontab的文件格式

| | | | | | |
|-------------------------------|----|--------------------|----|---------------|-------|
| * | * | * | * | * | comma |
| d | | | | | |
| 分钟 | 小时 | 几号 | 月份 | 星期几 | |
| 要执行的命令 | | | | | |
| 取值范围 (0-59之间的任何整数) | | 取值范围 (0-23之间的任何整数) | | | |
| 值范围 (1-31之间的任何整数) | | 取值范围 (1-12之间的任何整数) | | | |
| 值范围 (0-7之间的任何整数, 这里的0或7代表星期天) | | | | 系统的命令或自己编写的命令 | |