

Crontab 定时任务更新 SOLO 以及备份数据库

作者: [boolean-dev](#)

原文链接: <https://ld246.com/article/1566895195265>

来源网站: [链滴](#)

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



Linux定时执行任务Crontab

1. 安装Crontab

1.1 安装命令

vixie-cron软件包是cron的主程序
yum install vixie-cron

crontabs软件包是用来安装、卸装、或列举用来驱动 cron 守护进程的表格的程序
yum install crontabs

1.2 启动命令

启动服务
/sbin/service crond start
关闭服务
/sbin/service crond stop
#重启服务
/sbin/service crond restart
#重新载入配置
/sbin/service crond reload

1.3 查看相关状态

查看服务状态
service crond status
手动启动服务

service crond start

1.4 开机自启

chkconfig --level 35 crond on

2. crontab使用

2.1 定时器规则

以下是一条定时器的规则

```
# 每天0时0分执行`/usr/home/mysql.sh`的文件，并将日志写入`/usr/home/mysql.log`  
0 0 * * * /usr/home/mysql.sh >> /usr/home/mysql.log
```

crontab 文件的格式：

```
{minute} {hour} {day-of-month} {month} {day-of-week} {full-path-to-shell-script}
```

- minute: 分钟，区间为0-59
- hour: 小时，区间为0-23
- day-of-month: 每月的日，区间为0-31；
- month: 区间为1-12；1是1月，12是12月；
- day-of-week: 区间为0-6；周日是0。

除了数字还有以下几个特殊的符号需要特殊说明：

- *: 代表所有的取值范围内的数字；
- /: 代表每的意思，“*/5”表示每5个单位；
- -: 代表从某个数字到某个数字；
- ,: 分开几个离散的数字

以下是部分定时器示例

```
# 每五分钟执行  
*/5 * * * * /usr/home/mysql.sh  
# 每小时执行  
0 * * * * /usr/home/mysql.sh  
# 每天执行  
0 0 * * * /usr/home/mysql.sh  
# 每周执行  
0 0 * * 0 /usr/home/mysql.sh  
# 每月执行  
0 0 1 * * /usr/home/mysql.sh  
# 每年执行  
0 0 1 1 * /usr/home/mysql.sh  
  
# 每天早上6点  
0 6 * * * /usr/home/mysql.sh
```

```
# 每两个小时
0 */2 * * * /usr/home/mysql.sh

# 晚上11点到早上8点之间每两个小时
0 23-7/2 * * * /usr/home/mysql.sh

# 1月1日早上4点
0 4 1 1 * /usr/home/mysql.sh
```

3. 定时更新博客

3.1 编写更新博客shell

```
#!/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
#

docker pull b3log/solo
docker stop solo
docker rm solo
docker run --detach --name solo --restart=on-failure:10 --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/blog?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC" \
b3log/solo --listen_port=8081 --server_scheme=https --server_host=blog.booleandev.xyz -
server_port=
```

在添加定时任务之前，可提前执行下该文件，看看是否能够更新成功

如果没有执行权限 则执行以下命令 `chmod 777 文件名`

3.2 将shell添加到crontab定时任务中

```
# 打开crontab定时器文件
crontab -e

# 我的shell文件时放置于`/usr/local/soft/crontab/solo.sh`
# 将日志文件写于/usr/local/soft/crontab/solo.log
0 0 * * 0 /usr/local/soft/crontab/solo.sh >> /usr/local/soft/crontab/solo.log

# 等到时间，会自动执行，就可以看到solo.log文件内容
```

4. 定时备份mysql数据库

4.1 编写shell文件

```
#!/bin/bash

#获取当前时间
date_now=$(date +%Y%m%d-%H%M%S)
# 备份文件夹目录
backUpFolder=/usr/local/soft/backup/mysql/blog
username="root"
password="123456"
db_name="solo"
#定义备份文件名
fileName="${db_name}_${date_now}.sql"
#定义备份文件目录
backUpFileName="${backUpFolder}/${fileName}"
echo "starting backup mysql ${db_name} at ${date_now}."
/usr/bin/mysqldump -u${username} -p${password} --lock-all-tables --flush-logs ${db_name}
> ${backUpFileName}
#进入到备份文件目录
cd ${backUpFolder}
#压缩备份文件
tar zcvf ${fileName}.tar.gz ${fileName}
```

4.2 将shell添加到crontab定时任务中

```
# 打开crontab定时器文件
crontab -e

# 我的shell文件时放置于`/usr/local/soft/crontab/solo.sh`
# 将日志文件写于/usr/local/soft/crontab/solo.log
0 0 * * * /usr/local/soft/crontab/mysql.sh >> /usr/local/soft/crontab/mysql.log

# 等到时间，会自动执行，就可以看到mysql.log文件内容
```

5. 参考链接

[果冻想 的Linux定时执行任务Crontab](#)

[Galaxy' Home的CentOS安装crontab及使用方法](#)