

Linux 下使用 crontab 定时任务对 MySQL 数据库备份并将备份文件发送至邮箱

作者: [flowerunbeaten](#)

原文链接: <https://ld246.com/article/1644982632888>

来源网站: [链滴](#)

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



1.安装crontab

依次执行以下命令

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

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
```

设置开机自启动

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

2.mutt、msmtp安装

CentOs

```
# 更新所有的软件包  
yum update -y  
# 安装mutt、msmtp
```

```
yum install mutt msmtplib -y
```

Debian

```
apt update -y  
apt install mutt msmtplib -y
```

3.配置.muttrc、.msmtplib文件

进入用户根目录

```
cd /root/
```

新建.muttrc文件

```
vim .muttrc
```

.muttrc内容

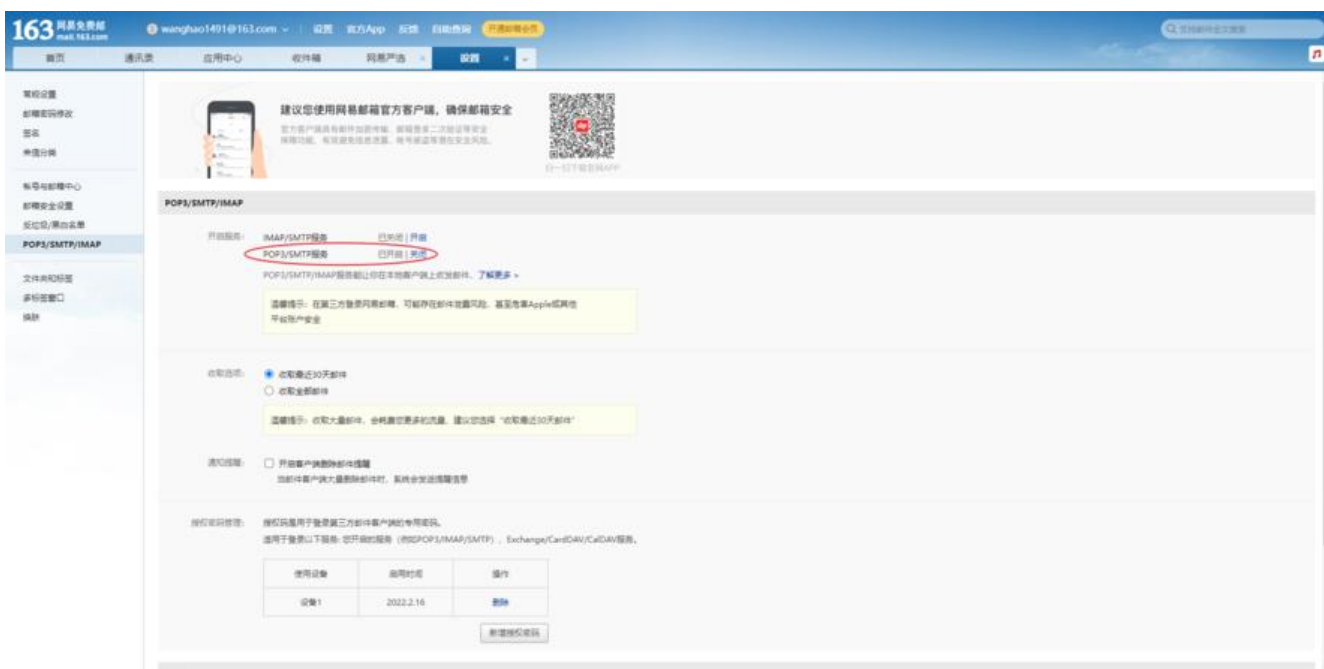
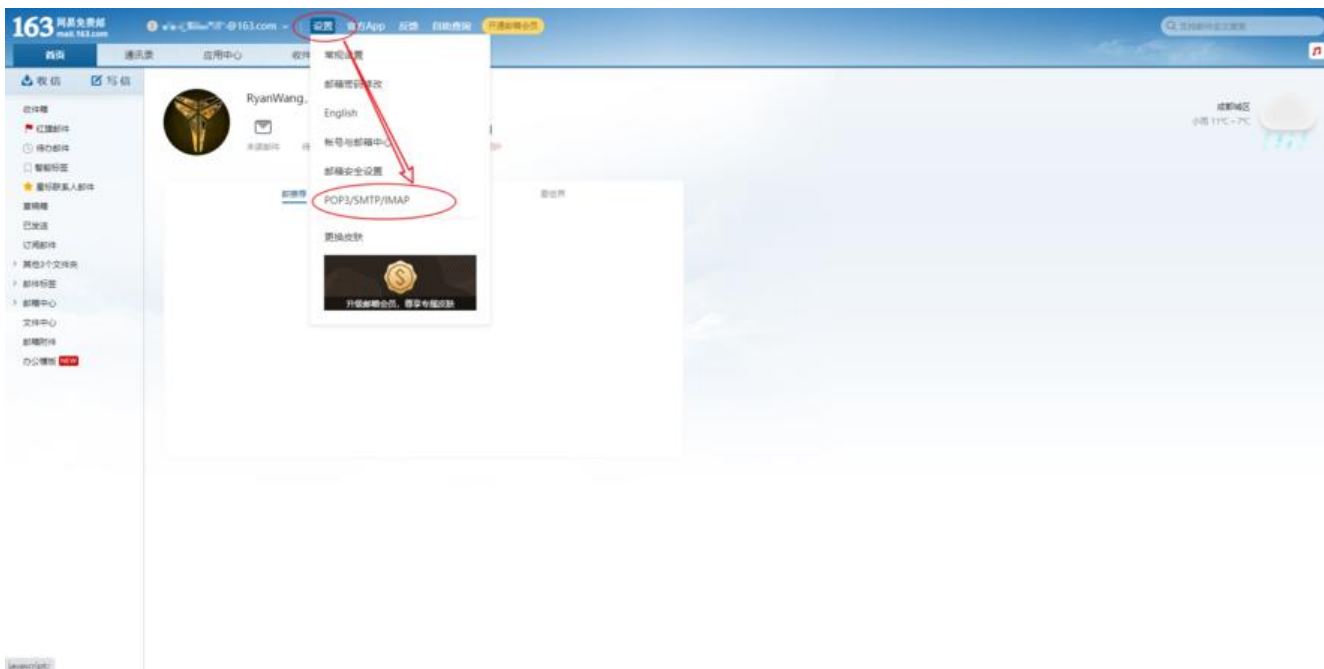
```
set sendmail="/usr/bin/msmtplib"  
set use_from=yes  
#发件人名字  
set realname="Ryan Wang"  
#发件人地址  
set from=xxxxxxxx@163.com  
set envelope_from=yes  
  
set charset="utf-8"  
set send_charset="utf-8"  
set content_type = "text/html\;charset=utf-8"
```

新建.msmtplib文件

```
vim .msmtplib
```

获取网易邮箱授权码

- 1.登录 [网易邮箱](#)
- 2.进入设置界面



- 3.开启POP3/SMTP服务，根据要求发送短信即可，然后网易会发送一个邮箱授权码（.msmtprc文会使用到）

.msmtprc内容

- 1 account default
- 2 #代理邮箱服务器，这里以163邮箱为例
- 3 host smtp.163.com
- 4 #端口
- 5 port 465
- 6 #同上发件人地址
- 7 from xxxxxxxx@163.com
- 8 tls_starttls off
- 9 tls on
- 10 tls_certcheck off

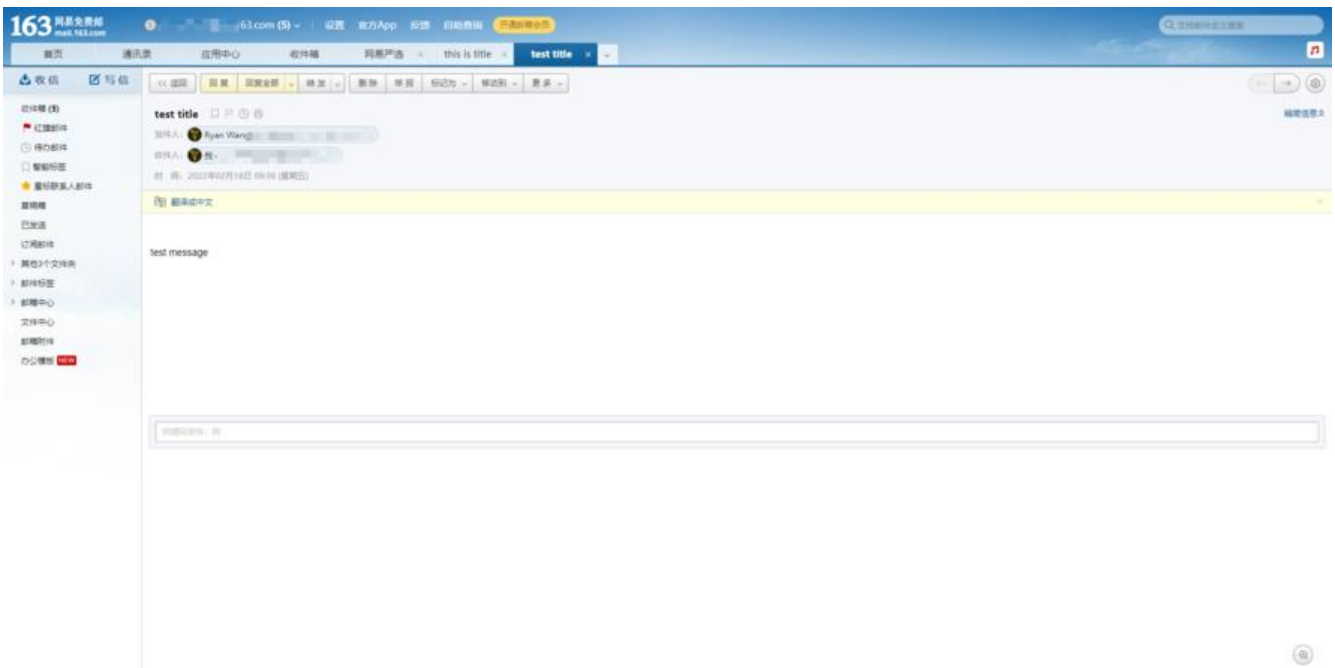
11 #验证方式
12 auth login
13 #用户名
14 user xxxxxxxxx@163.com
15 #邮箱授权码（网易提供邮箱授权码）
16 password Sxxxxxxx

设置.msmtprc文件权限

chmod 600 .msmtprc

测试发送邮件

echo "test message"|mutt -s "test title" xxxxxxxxx@163.com



4.MySQL数据库备份脚本文件

新建一个sh备份文件

cd /home/cron.d

vim mysql-backup-solo.sh

mysql-backup-solo.sh内容如下

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

```
#定义需要备份的数据库名
```

```
dbname=solo
```

```
#数据库备份文件存放位置
```

```
dbpath=/home/backup
```

```
#定义备份数据库时使用的用户名和密码
```

```
dbuser=username
```

```
dbpasswd=password
```

```

#数据库备份的路径
backuppath=${dbpath}/mysql/${dbname}

#日志文件目录
logpath=${dbpath}/log/${dbname}

#数据库备份日志文件存储的路径
logfile=${logpath}/mysql.log

#数据库IP地址
dbhost=localhost

#以当前的时间作为备份的数据库命名。
dumpfile=${dbname}_${date +%y%m%d%H%M}

#这个函数用来备份数据库
back_db()
{
    #将备份的时间、数据库名存入日志
    echo "-----"$(date +%Y-%m-%d%t%A%t%T)" Beginning database "${dbname}" backup-----
    --" >> ${logfile}

    #备份数据库，如果有错误信息也记入日志。(此处为MySQL自带的备份数据位置)
    /usr/local/mysql/bin/mysqldump -h${dbhost} -u${dbuser} -p${dbpasswd} ${dbname} > ${ba
    kuppath}/${dumpfile}.sql 2>> ${logfile}

    #开始压缩数据文件
    echo $(date +%Y-%m-%d%t%A%t%T)" Beginning zip ${backuppath}/${dumpfile}.sql" >> ${l
    gfile}

    #将备份数据库文件库压成ZIP文件，并删除先前的SQL文件。如果有错误信息也记入日志。
    cd ${backuppath} && tar -zcvf ${dumpfile}.tar.gz ${dumpfile}.sql && rm ${backuppath}/${d
    mpfile}.sql 2>> ${logfile}

    #将压缩后的文件名存入日志。
    echo "backup file name:"${dumpfile}".tar.gz" >> ${logfile}
    echo -e "-----"$(date +%Y-%m-%d%t%A%t%T)" Ending database "${dbname}" backup----
    --\n" >> ${logfile}
    #发送邮件
    tail -n 50 ${logfile} | mutt -s "Db BackUp" xxxxxxxx@163.com -a ${backuppath}/${dumpfile}.
    ar.gz
    echo "tail -n 50 ${logfile} | mutt -s \"Db BackUp\" xxxxxxxx@163.com -a ${backuppath}/${d
    mpfile}.tar.gz"

}

#这个函数用来删除30天前的数据，注意，这个脚本会删除当前目录下所有的早于30天前的文件
rm_oldfile()
{
    #查找出当前目录下30天前生成的文件，并将之删除
    find ${backuppath} -type f -mtime +30 -exec rm {} \;
}

```

```
if [ ! -d "${backuppath}" ]; then
    echo "创建文件夹${backuppath}"
    mkdir -p ${backuppath}
fi
```

```
if [ ! -d "${logpath}" ]; then
    echo "创建文件夹${logpath}"
    mkdir -p ${logpath}
fi
```

```
#切换到数据库备份的目录。如果不做这个操作，压缩文件时有可能会错误
cd ${backuppath}
```

```
#运行备份数据函数
back_db
```

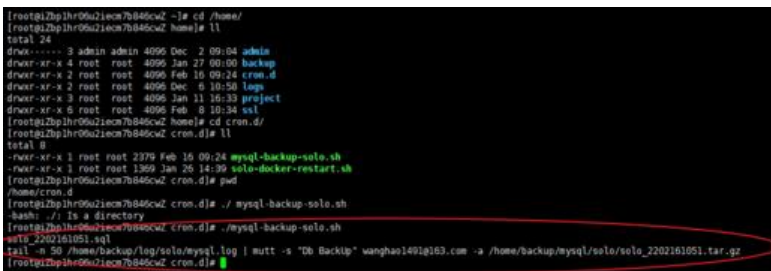
```
#运行删除文件函数
rm_oldfile
```

添加该文件的执行权限

```
chmod +x /home/cron.d/mysql-backup-solo.sh
```

测试脚本

```
./mysql-backup-solo.sh
```



```
[root@izbp1hr06u21ccm7b846cuz ~]# cd /home/
[root@izbp1hr06u21ccm7b846cuz home]# ll
total 24
drwx----- 3 admin admin 4096 Dec 2 09:04 admin
drwxr-xr-x 4 root root 4096 Jan 27 09:00 backup
drwxr-xr-x 2 root root 4096 Feb 16 09:24 cron.d
drwxr-xr-x 2 root root 4096 Dec 6 10:58 logs
drwxr-xr-x 3 root root 4096 Jan 11 16:33 project
drwxr-xr-x 6 root root 4096 Feb 3 10:34 sql
[root@izbp1hr06u21ccm7b846cuz home]# cd cron.d/
[root@izbp1hr06u21ccm7b846cuz cron.d]# ll
total 8
-rwxr-xr-x 1 root root 2379 Feb 16 09:24 mysql-backup-solo.sh
-rwxr-xr-x 1 root root 1369 Jan 26 14:30 solo-docker-restart.sh
[root@izbp1hr06u21ccm7b846cuz cron.d]# pwd
/home/cron.d
[root@izbp1hr06u21ccm7b846cuz cron.d]# ./mysql-backup-solo.sh
-bash: ./: is a directory
[root@izbp1hr06u21ccm7b846cuz cron.d]# ./mysql-backup-solo.sh
wget 2202161051.sql
tail -n 50 /home/backup/log/solo/mysql_log | mutt -s "DB Backup" wanghao1401@163.com -s /home/backup/mysql/solo/solo_2202161051.tar.gz
[root@izbp1hr06u21ccm7b846cuz cron.d]#
```

如图表示测试成功，可以在网易邮箱和服务器备份文件目录查看有一条备份数据

5.编写定时任务

执行以下命令

```
crontab -e
```

在编辑器中输入以下内容

```
#每天0时0分执行该命令  
0 0 * * * /home/cron.d/mysql-backup-solo.sh
```

然后保存退出即可

使用命令查看定时任务是否存在

```
crontab -l
```

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

如图表示成功，每天的0时0分就会执行定时任务通过脚本备份数据库并将备份文件发送至邮箱了。

crontab参数说明

*	*	*	*	*	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