



链滴

备份数据库到 github

作者: [mnizht](#)

原文链接: <https://ld246.com/article/1560305047308>

来源网站: [链滴](#)

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

由于之前用的国外vps布的solo，前段时间封了几个ip，导致数据solo访问不了，数据也取不到了。以想着得能够把数据备份一下，以防什么时候服务器再死掉。。。

所以网上搜了备份数据库的脚本，然后又加了几句，实现了备份数据库到github。

[脚本来源](#)

新建脚本文件

```
vi /opt/mysql-backup.sh
```

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

```
[]  
# 以下配置信息请自己修改  
[]  
mysql_user="USER" #MySQL备份用户  
mysql_password="PASSWORD" #MySQL备份用户的密码  
mysql_host="localhost"  
mysql_port="3306"  
mysql_charset="utf8" #MySQL编码  
backup_db_arr=("db1" "db2") #要备份的数据库名称，多个用空格分开隔开 如("db1" "db2" "db3")  
  
backup_location=/opt/mysql #备份数据存放位置，末尾请不要带"/",此项可以保持默认，程序会自  
创建文件夹  
expire_backup_delete="ON" #是否开启过期备份删除 ON为开启 OFF为关闭  
expire_days=3 #过期时间天数 默认为三天，此项只有在expire_backup_delete开启时有效  
[]  
# 本行开始以下不需要修改  
backup_time=`date +%Y%m%d%H%M` #定义备份详细时间  
backup_Ymd=`date +%Y-%m-%d` #定义备份目录中的年月日时间  
backup_3ago=`date -d '3 days ago' +%Y-%m-%d` #3天之前的日期  
backup_dir=$backup_location/$backup_Ymd #备份文件夹全路径  
welcome_msg="Welcome to use MySQL backup tools!" #欢迎语  
[]  
# 判断MYSQL是否启动,mysql没有启动则备份退出  
mysql_ps=`ps -ef |grep mysql |wc -l`  
mysql_listen=`netstat -an |grep LISTEN |grep $mysql_port|wc -l`  
if [ [$mysql_ps == 0] -o [$mysql_listen == 0] ]; then  
    echo "ERROR:MySQL is not running! backup stop!"  
    exit  
else  
    echo $welcome_msg  
fi  
[]  
# 连接到mysql数据库，无法连接则备份退出  
mysql -h$mysql_host -P$mysql_port -u$mysql_user -p$mysql_password <<end  
use mysql;  
select host,user from user where user='root' and host='localhost';  
exit  
end  
[]  
flag=`echo $?`
```

```

if [ $flag != "0" ]; then
    echo "ERROR:Can't connect mysql server! backup stop!"
    exit
else
    echo "MySQL connect ok! Please wait....."
[]
# 判断有没有定义备份的数据库，如果定义则开始备份，否则退出备份
if [ "$backup_db_arr" != "" ];then
#dbnames=$(cut -d ';' -f1-5 $backup_database)
#echo "arr is (${backup_db_arr[@]})"
for dbname in ${backup_db_arr[@]}
do
echo "database $dbname backup start..."
[]
mkdir -p $backup_dir
mysqldump -h$mysql_host -P$mysql_port -u$mysql_user -p$mysql_password $dbname --def
ult-character-set=$mysql_charset | gzip > $backup_dir/$dbname-$backup_time.sql.gz
[]
flag=`echo $?`
if [ $flag == "0" ];then
echo "database $dbname success backup to $backup_dir/$dbname-$backup_time.sql.gz"
else
echo "database $dbname backup fail!"
fi
[]
done
else
echo "ERROR:No database to backup! backup stop"
exit
fi
[]
# 如果开启了删除过期备份，则进行删除操作
if [ "$expire_backup_delete" == "ON" -a "$backup_location" != "" ];then
#`find $backup_location/ -type d -o -type f -ctime +$expire_days -exec rm -rf {} \;`
[]
#find $backup_location/ -type d -mtime +$expire_days | xargs rm -rf
#因为之后加了git，所以此处的筛选条件多加一条按名称匹配，否则会把git的配置文件删掉，导致gi
出错
find $backup_location/ -type d -mtime +$expire_days -name "2019*" | xargs rm -rf

echo "Expired backup data delete complete!"
fi
echo "All database backup success! Thank you!"
exit
fi

chmod 700 mysql-backup.sh
# 脚本目前是查询数据库数据备份到本地。创建完成后可以先手动运行一下看看是否成功。
./mysql-backup.sh
# 执行成功后会在 backup_location 参数指定的文件夹下创建出今天日期的文件夹和文件。（名为20
9-06-11的文件夹，名为 solo-201906110951.sql.gz 的文件）

```

配置git

```
//安装git
yum -y install git
//进入备份文件夹，创建本地仓库
cd /usr/local/mysql/backup
git init
//直接将用户名和密码写入url，操作简单，但是不太安全
git remote set-url origin http://yourname:password@github.com/yourname/repository.git
//设置git 配置
git config --global user.name "your name"
git config --global user.email "your email"
[]
配置完后先手动git提交一下看看配置是否有问题
[]

#脚本文件追加git操作
git_commit="push backup"
echo "start publish..."
#切换文件夹
cd /usr/local/mysql/mysqlBackup
git add -A
git commit -m $git_commit
git push
echo "database push success"
[]
```

设置定时任务

```
//直接编辑系统定时任务文件，创建定时任务
vim /etc/crontab
[]
根据提示设置好时间，写上运行脚本语句
0 17 * * * root /user/local/mysql-backup.sh
（此处设置的时间应注意服务器上的时区，不是中国时区的可以更改服务器时区，也可以设置时间的候算好时间差，比如我的服务器是0时区，这里设置的17 就应该是中国时间的每天凌晨1点）
```