



链滴

使用 mysqldump 结合 git 和 shell 定时备份 Mysql 数据库

作者: [washmore](#)

原文链接: <https://ld246.com/article/1517301455417>

来源网站: [链滴](#)

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

说在前面

今天早些时候突然发现博客无法打开,远程数据库无法访问了,经过咨询VPS供应商客服后得知,因为发我的VPS上面架设了VPN相关设施,被强制封禁了...封禁了...封禁了,我的数据库(☹ ☹ ☹)啊! 最近没备份啊...

施工过程

在漫长的等待答复过程中,我不得不做起了事后诸葛亮,以应对以后可能出现的此类问题

思路

1. 可以使用mysqldump生成备份sql文件;
2. 使用git存储sql文件;
3. 使用gitlab或者免费私人git仓库;(不考虑使用github,因为不提供免费的私人仓库...成本较高)

调研

- 使用mysqldump命令成功生成数据库名_bak.sql文件,查看内容正确;

```
mysqldump -h主机名 -P端口号 -u数据库用户名 -p密码 数据库名 > 数据库名_bak.sql
```

- 打听到码云有免费的git私人仓库 <https://gitee.com>
- 使用git命令成功提交并推送生成的sql备份文件;

```
git add -A
git commit -m "auto-commit:${createAt}"
git push origin master
```

详解

基于以上准备,我决定写一个shell脚本,然后创建linux定时任务,每日自动dump数据库文件并自动上到码云git私人仓库!

首先需要在准备执行shell的机器上安装有mysql服务并能正常使用mysqldump命令,如果没有请自行装;

然后需要有git服务并能正常使用git命令,另外为了方便提交代码以及防止泄密,使用ssh替代账号密码式进行git鉴权!

使用ssh管理git仓库

1. 首先配置基础信息

```
git config --global user.name "washmore" #提交代码的署名
git config --global user.email "admin@washmore.tech" #提交代码的署名邮箱
```

2. 生成ssh公钥

执行下面命令，三次回车即可生成 ssh key:

```
ssh-keygen -t rsa -C "admin@washmore.tech"
```

3. 查看生成的公钥

```
cat ~/.ssh/id_rsa.pub
```

将以上命令打印出屏幕的内容添加到码云<https://gitee.com/profile/sshkeys>。(注意是三段式全部内)

4. 验证配置

```
ssh -T git@gitee.com
```

如果输出>Welcome to Gitee.com, your account!表示配置成功了,没有的话自行排查

5. 其他git基础clone,commit,push等基础操作不再赘述,如有不懂请自行查阅资料

接下来开始编写我们的shell脚本data_back.sh,详情请看每个步骤的注释,shell脚本存放位置请自选

```
#!/bin/bash
##按月创建目录
my_month=`date +%Y%m`
##为sql文件名添加时间戳
my_datetime=`date +%Y-%m-%d-%H:%M:%S`
##设置git仓库根目录(需要提前在码云上创建私有仓库并clone到本地)
project_path=/Users/chenyuqing/IdeaProjects/data_rep
##创建月份子目录(如果已存在会创建失败,忽略)
mkdir ${project_path}/${my_month}
##切换到git仓库目录
cd ${project_path}
##开始使用mysqldump命令备份数据库
##tips:此处需要使用mysqldump的全路径,否则后面使用定时任务时候会发现dump出来的sql文件是
的,具体原因待排查
/usr/local/Cellar/mysql/5.7.21/bin/mysqldump -h数据库所在服务器域名 -P端口 -u数据库账号 -
密码 数据库名 > ${project_path}/${my_month}/数据库名-${my_datetime}.sql
##commit and push
git add -A
git commit -m "auto-commit:${my_datetime}"
git push origin master
```

完成shell脚本后,使用chmod +x data_back.sh命令授权,然后开始创建系统定时任务

使用crontab -e命令开启定时任务的vim模式,添加一条记录

```
30 2 * * * /Users/chenyuqing/IdeaProjects/data_back.sh
```

linux的cron表达式请自行查阅,查看现有的系统定时任务命令crontab -l,由于mysqldump会锁表,大影响数据库的读写性能,因此建议诸位设置为半夜执行定时任务!

MAC OS X系统的童鞋到此结束,linux还需要执行命令重启定时任务服务:

```
systemctl restart crond
```

完成所有工序后，每日凌晨2点30分，自动备份mysql的数据库并推送到码云git仓库。妈妈再也不用担心数据丢失了。

后记

下午被告知数据抢救回来了,仅仅是被要求下架VPN服务,不过本文所记录的方案也会被执行起来,以免后顾之忧,希望能给有此需求的各位童鞋带来一点点帮助