# 备份相关

数据量少于10G以内使用mydumper,mysqldump 进行备份，其他备份建议xtrabackup

### 1、mysql创建备份用账号

[给MySQL的备份账户添加权限](https://cloud.tencent.com/developer/article/1627000)

<https://www.imooc.com/article/278011>

GRANT SELECT, RELOAD, PROCESS, SUPER, LOCK TABLES, REPLICATION CLIENT ON \*.\* TO 'backup'@'localhost' IDENTIFIED BY '123.com';

### 2、备份脚本

[innobackupex远程备份脚本](https://blog.csdn.net/woshaguayi/article/details/118969866)

### 3 docker-mysql备份

<https://www.djc8.cn/archives/docker-mysql-scheduled-backup.html>

<https://blog.csdn.net/weixin_52270081/article/details/123552094>

#!/bin/bash  
export NOW="$(date +"%Y-%m-%d")"  
export DATA\_DIR=/opt/backup  
docker exec -it mysql\_3306 /bin/bash -c "mysqldump -uroot -p"Xmmtr#!2022.\*" --set-gtid-purged=OFF --master-data=2 -q -c --skip-add-locks -R -E -B xmmtr\_platform > $DATA\_DIR/$NOW.sql"  
#docker exec -it mysql\_3306 /bin/bash -c "mysqldump -uroot -p"Xmmtr#!2022.\*" --set-gtid-purged=OFF --all-databases > $DATA\_DIR/$NOW.sql"  
#docker exec -it mysql\_3306 /bin/bash -c "mysqldump -uroot -p123.com --set-gtid-purged=OFF xmmtr\_platform > $DATA\_DIR/$NOW.sql"  
#docker exec -it mysql\_3306 /bin/bash -c "mysqldump -uxmmtr-platform -p"xmmtr#Attiot2019" --set-gtid-purged=OFF xmmtr\_platform > $DATA\_DIR/$NOW.sql"  
  
#普通用户  
#docker exec -it mysql\_3306 /bin/bash -c "mysqldump -ubackup -p123.com --set-gtid-purged=OFF xmmtr\_platform > $DATA\_DIR/$NOW.sql"

### 4 mysqldump参数

mysqldump -u用户 -p密码 --master-data=2 -c --single-transaction --set-gtid-purged=OFF --skip-add-locks --triggers -R -E -B databaename > back.sql

**全量备份所有库(基于binlog)**

mysqldump \  
--all-databases \  
--set-charset \  
--default-character-set=utf8 \  
--create-options \  
--triggers \  
--routines \  
--events \  
--add-drop-database \  
--add-drop-table \  
--add-drop-trigger \  
--add-locks \  
--disable-keys \  
--extended-insert \  
--set-gtid-purged=OFF \  
--master-data=2 \  
--single-transaction \  
--quick \  
--flush-privileges \  
--flush-logs \  
--skip-opt \  
--result-file=/var/local/mysql/dump.sql \  
--log-error=/var/log/mysql/dump\_error\_log \  
-S /var/local/mysql/mysql.sock \  
-p \

参数详解

--set-charset: 添加'SET NAMES default\_character\_set'到输出文件。默认为打开状态，使用--skip-set-charset关闭选项。

--default-character-set； 设置默认字符集，默认值为utf8

--create-options, -a：在CREATE TABLE语句中包括所有MySQL特性选项。(默认为打开状态)

--add-drop-database：每个数据库创建之前添加drop数据库语句

--add-drop-table：每个数据表创建之前添加drop数据表语句。(默认为打开状态，使用--skip-add-drop-table取消选项)

--add-drop-trigger： 在备份文件中添加、删除相同触发器的 SQL 语句

--add-locks： 在备份数据表前后添加表锁定与解锁 SQL 语句

--disable-keys：对于每个表，用/\*!40000 ALTER TABLE tbl\_name DISABLE KEYS \*/;和/\*!40000 ALTER TABLE tbl\_name ENABLE KEYS \*/;语句引用INSERT语句。这样可以更快地导入dump出来的文件，因为它是在插入所有行后创建索引的。**该选项只适合MyISAM表**，默认为打开状态

--extended-insert -e: 使用具有多个VALUES列的INSERT语法。这样使导出文件更小，并加速导入时的速度。默认为打开状态，使用--skip-extended-insert取消选项。

--set-gtid-purged: 为ON会在导出的备份文件中加入SET @@GLOBAL.GTID\_PURGED和SET @@SESSION.SQL\_LOG\_BIN= 0三种情况：

1. 导出的文件用于本地备份，给成OFF

1. 导出的文件用于从库导入数据，给成ON

1. 导出的文件用于其他不相关节点，给成OFF

--master-data: 该选项将binlog的位置和文件名追加到输出文件中。如果为1，将会输出CHANGE MASTER 命令；如果为2，输出的CHANGE MASTER命令前添加注释信息。该选项将打开--lock-all-tables 选项，除非--single-transaction也被指定（在这种情况下，全局读锁在开始导出时获得很短的时间；其他内容参考下面的--single-transaction选项）。该选项自动关闭--lock-tables选项。如

-- CHANGE MASTER TO MASTER\_LOG\_FILE='mysql-bin.000002', MASTER\_LOG\_POS=1474;

--single-transaction：该选项在导出数据之前提交一个BEGIN SQL语句，BEGIN 不会阻塞任何应用程序且能保证导出时数据库的一致性状态。它只适用于多版本存储引擎，仅InnoDB。本选项和--lock-tables 选项是互斥的，因为LOCK TABLES 会使任何挂起的事务隐含提交。要想导出大表的话，应结合使用--quick 选项。

--quick, -q：不缓冲查询，直接导出到标准输出。默认为打开状态，使用--skip-quick取消该选项。

**--result-file, -r：**直接输出到指定文件中。该选项应该用在使用回车换行对（\\r\\n）换行的系统上（例如：DOS，Windows）。该选项确保只有一行被使用。

[Mysqldump参数大全](https://zhuanlan.zhihu.com/p/370923449)

[mysqldump常用参数](https://www.jianshu.com/p/8a345804193e)

[mysqldump备份参数](https://www.jianshu.com/p/2020ee8d7774)

**--routines, -R** 导出存储过程以及自定义函数

**--events, -E** 导出事件。

**--databases, -B** 导出几个数据库。参数后面所有名字参量都被看作数据库名

**--quick, -q** 不缓冲查询，直接导出到标准输出。默认为打开状态，使用--skip-quick取消该选项。

**--complete-insert, -c** 使用完整的insert语句(包含列名称)。这么做能提高插入效率，但是可能会受到max\_allowed\_packet参数的影响而导致插入失败。

**--flush-privileges** 在导出mysql数据库之后，发出一条FLUSH PRIVILEGES 语句。为了正确恢复，该选项应该用于导出mysql数据库和依赖mysql数据库数据的任何时候

**--flush-logs** 开始导出之前刷新日志。

请注意：假如一次导出多个数据库(使用选项--databases或者--all-databases)，将会逐个数据库刷新日志。除使用--lock-all-tables或者--master-data外。在这种情况下，日志将会被刷新一次，相应的所以表同时被锁定。因此，如果打算同时导出和刷新日志应该使用--lock-all-tables 或者--master-data 和--flush-logs。

**--triggers** 导出触发器。该选项默认启用，用--skip-triggers禁用它。

**--hex-blob** 使用十六进制格式导出二进制字符串字段。如果有二进制数据就必须使用该选项。影响到的字段类型有BINARY、VARBINARY、BLOB。

**--single-transaction：**对事务引擎执行热备