```
####################补充##############################################
    配置主从时从库应指定的内容:
    relay_log = mysql-relay-bin              中继日志的目录地址
    relay_log-index = mysql-relay-bin.index  中继日志的索引文件
    log_slave_update = 1                     防止备份时数据产生一些问题

    添加给指定用户指定权限(8.0):
    grant insert,delete on *.* to  will@'%' with grant option;

    移除一些用户权限,但不移除该用户:
    revoke delete on *.* from 'will'@'%';

    删除用户及其权限:
    drop user 'test1'@'localhost';
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    all privileges 包括一下权限:
    select,
    insert,
    update,
    delete,
    create,
    drop,
    references,
    index,
    alter,
    create temporary tables,
    lock tables,
    execute,
    create view,
    show view,
    create routine,
    alter routine,
    event,
    trigger`
#####################################################################

一.自定义复制对象:
    1. MySQL主要复制启动配置(在从库的配置文件中配置)
      + ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      +   参 数                      作用                 实例
      +  replicate-do-table     指定需要复制的表      replicate-do-table=php45.stu
      +  replicate-ignore-table     指定不复制的表        replicate-ignore-table=test.rep_t1
      +  replicate-do-db        指定不复制的数据库    replicate-do-db=db1
      ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

二.数据库备份概述:
    1.应用场景:现在我们的情况就是在数据库中已经事先就存在了laravel-shop的数据表，并且我们可以往里面添加一些数据作为测试用的，
  而根据主从来说同步的是binlog日志中的信息，但是一般数据库开始的时候并不会直接开启binlog，默认就是off需要修改配置调整为on
    有几种办法来初始化备库或者从其他服务器克隆数据大备库。包括从主库复制数据、从另一台备库克隆数据等等主要的思路就是数据库备份

    2.备份方式:
        1)逻辑备份:又称为导出
            优点:灵活性,版本要求不高,数据量小就可以
            缺点:导入导出速度慢
        2)物理备份(暴力备份)
            直接把使用的数据库全部压缩(连带服务压缩)
            优点:速度快
            缺点:版本要求高,版本尽量一致

    运行方式:
        1)冷备份:数据停止服务运行,停止写的操作,在这个基础上备份
```

```
        优点:数据安全
    2)热备份:不希望服务停机(在线备份)
        备份过程使用binlog


三.数据库备份的使用(跳过了暴力备份)
  1.逻辑备份/冷备份 (mysql自带的mysqldump工具,在bin目录下):
    [root@localhost bin]# mysqldump -uroot -p will > /usr/local/mysql/will.sql
    Enter password:
    [root@localhost bin]#

  实际项目过程:
    1)开启主库的binlog日志配置服务id:
        log_bin=mysql-bin
        server-id=自定义
    2)停止服务/停止写操作,进行数据库备份
    3)备份之后,启动服务,释放锁
    4)从库配置
    5)指定ip,port,user
    6)从库启动配置
    7)还原 同步主库数据



   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    mysql> flush tables with read lock;
    Query OK, 0 rows affected (0.07 sec)

    mysql> delete  from stu;
    ERROR 1223 (HY000): Can't execute the query because you have a conflicting read lock

    mysql> unlock tables;
    Query OK, 0 rows affected (0.00 sec)

    mysql> delete  from stu;
    Query OK, 0 rows affected (0.01 sec)
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    同步数据库步骤:
        1.主库执行 flush tables with read lock 锁住所有表不能写
        2.通过mysqldump做数据库的备份
        3.执行unlock tables释放表锁
        4.从库的数据还原


  2.逻辑备份的另一个工具-----mydumper(通过  多线程方式  对数据进行备份 效率会比mysqldump快近10倍)
   下载地址
   https://launchpad.net/mydumper/+download

  安装:
   1) tar -zvxf mydumper-0.9.1.tar....
   2) cd mydumper-0.9.1
   3) cmake .
   4) make && make install
   5) 检测安装 mydumper -V

   导出:
   -- 全库备份, 注意-u -p 后面要有空格, -o 后面跟的是目录, 不是文件名
   mydumper -u root -p root -P 3306  -o /home/www
   --  -B 指定备份数据库
   mydumper -u root -p root -P 3306 -B php45  -o /home/www

   导入(myloader):
```

```
myloader -u root -p root -P 3306 -B php45 -d /home/www
```

进行 mysqldump 和 mydumper 的执行时间的对比

```
time mysqldump -uroot -p will php45 > /home/www                   0.9
time  mydumper -u root -p root -P 3306 -B php45  -o /home/www    0.33
```

3.热备份工具(xtrabackup 有版本要求 mysql8.0之前用2.4一下版本)

3.1 xtrabackup手册：https://www.percona.com/doc/percona-xtrabackup/2.4/installation/yum_repo.html

3.2 热备份的方式也是直接复制数据物理文件，和冷备份一样，但热备份可以不停机直接复制，一般用于7×24小时不间断的重要核心业务

3.3 xtrabackup是Percona公司的开源项目，用以实现类似ImnoDB官方的热备份工具ImmoDB Hot Backup的功能，它能非常快速地备份与恢复

3.4 xtrabackup的安装

```
[root@localhost file]$ wget https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.4/binary/redhat/7/x86_64/percona-xtrabackup-24-2.4.4-1.el7.x86_64.rpm

[root@localhost file]$ yum localinstall percona-xtrabackup-24-2.4.4-1.el7.x86_64.rpm

[root@localhost file]# rpm -qa | grep xtrabackup
```

3.5 备份过程:

1)主库中执行备份操作 -> 执行数据库的data文件
    1.1)把主库的文件传递到从库中
2)从库中需要保持-选择合并,选择data为空或者不存在
    2.1)停止从库的服务
    2.2)清空data
3)配置从库的my.cnf-与主库配置尽量保持一致
4)恢复

3.6 Xtrabackup中主要包含两个工具：

· xtrabackup是用于热备份InnoDB及XtraDB表中数据的工具，不能备份其他类型的表，也不能备份数据表结构。
· innobackupex是将xtrabackup进行封装的perl脚本，它提供了备份MyISAM表的能力。由于innobackupex的功能更为全面完善.
所以一般选择innobackupex来进行备份。

~~~~~~

常用选项:

--host   指定主机
--user   指定用户名
--password   指定密码
--port   指定端口
--databases   指定数据库
--incremental   创建增量备份
--incremental-basedir   指定包含完全备份的目录
--incremental-dir   指定包含增量备份的目录
--apply-log   对备份进行预处理操作

一般情况下，在备份完成后，数据尚且不能用于恢复操作，因为备份的数据中可能会包含尚未提交的事务或已经提交但尚未同步至数据文件中的事务。因此，此时数据文件仍处理不一致状态。“准备”的主要作用正是通过回滚未提交的事务及同步已经提交的事务至数据文件也使得数据文件处于一致性状态。

--redo-only   不回滚未提交事务
--copy-back   恢复备份目录

~~~~~~

3.7 开始进行备份操作,进入 主库 中

```
[root@centos ~]# innobackupex --defaults-file=/etc/my.cnf --user=root --password=root --backup /home/www
[root@localhost home]# ll /home/www
```

```
    -rw-r----- 1 root root      425 9月  26 12:43 backup-my.cnf
    drwxr-x--- 2 root root       58 9月  26 12:43 bin
    -rw-r----- 1 root root      361 9月  26 12:43 ib_buffer_pool
    -rw-r----- 1 root root 77594624 9月  26 12:43 ibdata1
    drwxr-x--- 2 root root     4096 9月  26 12:43 laravel@002dshop
    drwxr-x--- 2 root root     4096 9月  26 12:43 mysql
    drwxr-x--- 2 root root     8192 9月  26 12:43 performance_schema
    drwxr-x--- 2 root root     8192 9月  26 12:43 sys
    drwxr-x--- 2 root root       46 9月  26 12:43 test
    -rw-r----- 1 root root       24 9月  26 12:43 xtrabackup_binlog_info
    -rw-r----- 1 root root      113 9月  26 12:43 xtrabackup_checkpoints
    -rw-r----- 1 root root      508 9月  26 12:43 xtrabackup_info
    -rw-r----- 1 root root     2560 9月  26 12:43 xtrabackup_logfile
```

拷贝到从库中

```
[root@localhost home]#scp -r /home/www/ slave1@192.168.100.140:/home/www
```

然后呢进入 从库 中

```
[root@localhost slave1]# ll /home/www
    -rw-r----- 1 root root      425 9月  26 12:43 backup-my.cnf
    drwxr-x--- 2 root root       58 9月  26 12:43 bin
    -rw-r----- 1 root root      361 9月  26 12:43 ib_buffer_pool
    -rw-r----- 1 root root 77594624 9月  26 12:43 ibdata1
    drwxr-x--- 2 root root     4096 9月  26 12:43 laravel@002dshop
    drwxr-x--- 2 root root     4096 9月  26 12:43 mysql
    drwxr-x--- 2 root root     8192 9月  26 12:43 performance_schema
    drwxr-x--- 2 root root     8192 9月  26 12:43 sys
    drwxr-x--- 2 root root       46 9月  26 12:43 test
    -rw-r----- 1 root root       24 9月  26 12:43 xtrabackup_binlog_info
    -rw-r----- 1 root root      113 9月  26 12:43 xtrabackup_checkpoints
    -rw-r----- 1 root root      508 9月  26 12:43 xtrabackup_info
    -rw-r----- 1 root root     2560 9月  26 12:43 xtrabackup_logfile
```

进项data目录的清空(这里我们选择重命名)

```
[root@localhost slave1]# mv /usr/local/mysql/data /usr/local/mysql/data2
```

进行内容的拷贝

```
[root@localhost server]# innobackupex --defaults-file=/usr/local/mysql/etc/my.cnf --copy-back
/home/www/2020-05-01_12-43-34/
[root@localhost server]#chown -R mysql:mysql /usr/local/mysql/data
```

重启mysql服务

进入检测数据库即可

3.8对于热备份实现解释

innobackupex启动后，会先fork一个进程，用于启动xtrabackup，然后等待xtrabackup备份ibd数据文件；

xtrabackup在备份innoDB数据是，有2种线程：redo拷贝线程和ibd数据拷贝线程。xtrabackup进程开始执行后，会启动一个redo拷贝的线程，用于从最新的checkpoint点开始顺序拷贝redo.log；再启动ibd数据拷贝线程，进行拷贝ibd数据。这里是先启动redo拷贝线程的。在此阶段，innobackupex进行处于等待状态（等待文件被创建）

xtrabackup拷贝完成ibd数据文件后，会通知innobackupex（通过创建文件），同时xtrabackup进入等待状态（redo线程依旧在拷贝redo.log）

innobackupex收到xtrabackup通知后哦，执行FLUSH TABLES WITH READ LOCK（FTWRL），取得一致性位点，然后开始备份非InnoDB文件（如frm、MYD、MYI、CSV、opt、par等格式的文件），在拷贝非InnoDB文件的过程当中，数据库处于全局只读状态。

当innobackup拷贝完所有的非InnoDB文件后，会通知xtrabackup，通知完成后，进入等待状态；

xtrabackup收到innobackupex备份完成的通知后，会停止redo拷贝线程，然后通知innobackupex，redo.log文件拷贝完成；

innobackupex收到redo.log备份完成后，就进行解锁操作，执行：UNLOCK TABLES；

最后innbackupex和xtrabackup进程各自释放资源，写备份元数据信息等，innobackupex等xtrabackup子进程结束后退出。