**一．apt安装**

**1.**  **更新源**

sudo apt install dirmngr

sudo apt-get install software-properties-common

sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8

sudo add-apt-repository **'deb [arch=amd64,i386,ppc64el] http://mirrors.neusoft.edu.cn/mariadb/repo/10.3/ubuntu xenial main'**

sudo apt update

**2.**  **安装数据库环境**

sudo apt install -y mariadb-server mariadb-client

**3.**  **安装过程**

sudo mysql\_secure\_installation

  NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB

         SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

  In order to log into MariaDB to secure it, we'll need the current

  password for the root user.  If you've just installed MariaDB, and

  you haven't set the root password yet, the password will be blank,

  so you should just press enter here.

  Enter current password for root (enter for none): 设置root的密码,暂时不设置，enter

  OK, successfully used password, moving on...

  Setting the root password ensures that nobody can log into the MariaDB

  root user without the proper authorisation.

  You already have a root password set, so you can safely answer 'n'.

  Change the root password? [Y/n] n 是否改变密码:n不改变

    ... skipping.

  By default, a MariaDB installation has an anonymous user, allowing anyone

  to log into MariaDB without having to have a user account created for

  them.  This is intended only for testing, and to make the installation

  go a bit smoother.  You should remove them before moving into a

  production environment.

  Remove anonymous users? [Y/n] y 是否移除测试用户：y是的

    ... Success!

  Normally, root should only be allowed to connect from 'localhost'.  This

  ensures that someone cannot guess at the root password from the network.

  Disallow root login remotely? [Y/n] y 是否允许root远程登录：y允许

    ... Success!

  By default, MariaDB comes with a database named 'test' that anyone can

  access.  This is also intended only for testing, and should be removed

  before moving into a production environment.

  Remove test database and access to it? [Y/n] y 删掉测试数据库test:y删掉

    - Dropping test database...

    ... Success!

    - Removing privileges on test database...

    ... Success!

  Reloading the privilege tables will ensure that all changes made so far

  will take effect immediately.

  Reload privilege tables now? [Y/n] y 重新加载权限表：y重新加载

    ... Success!

  Cleaning up...

  All done!  If you've completed all of the above steps, your MariaDB

  installation should now be secure.

  Thanks for using MariaDB

**4.**  **查看服务状态指令**

sudo systemctl status mysql

**5.**  **设置mariadb随系统服务启动**

  sudo update-rc.d  mysql defaults

**6.**  **修改mariadb配置文件，监听外网访问**

vim /etc/mysql/my.cnf

#bind-address        = 127.0.0.1  #注释掉这一行

**7.**  **重启数据库服务，使配置生效**

sudo systemctl restart mysql

**8.**  **设置root远程登录密码**

mysql -u root

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'password' WITH GRANT OPTION;

**二.** **主从配置**

**1.**  **修改配置文件**

**A服务器**

vim /etc/mysql/my.cnf

在配置文件中添加下面代码

[mysqld]

server-id=1

binlog-ignore-db = mysql

binlog-ignore-db = information\_schema

log-bin=master-bin

relay-log=relay-bin

sync-binlog = 1

character-set-server=utf8

**B服务器**

vim /etc/mysql/my.cnf

[mysqld]

server-id=2

replicate-ignore-db = mysql

replicate-ignore-db = information\_schema

relay-log=relay-bin

log-bin=master-bin

sync-binlog = 1

character-set-server=utf8

**2.**  **重启两个服务器**

sudo systemctl restart mysql

**3.**  **清空A、B的同步日志**

如果之前有开启过同步，需要先停止同步，stop slave

mysql -u root(表示在mysql的命令行中执行)

reset master;

reset slave;

**4.**  **创建同步用户**

A:给B创建用户

mysql -u root

grant replication slave on \*.\* to 'slaveuser'@'xxx.xxx.xxx.xxx(B的ip)' identified by 'slaveuser';

B:给A创建用户

mysql -u root

grant replication slave on \*.\* to 'slaveuser'@'xxx.xxx.xxx.xxx(A的ip)' identified by 'slaveuser';

**5.**  **A和B锁表**

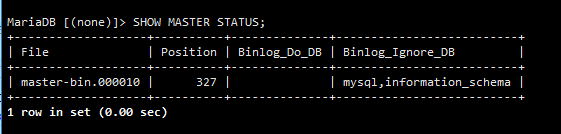
mysql -u root

FLUSH TABLES WITH READ LOCK;

**6.**  **查看A和B二进制文件**

mysql -u root

SHOW MASTER STATUS;



**7.**  **修改A和B的配置**

按照步骤6的查询结果修改配置

**A数据库**

mysql -u root

CHANGE MASTER TO MASTER\_HOST='xxx.xxx.xxx.xxx(B的ip)',MASTER\_USER='slaveuser', MASTER\_PASSWORD='slaveuser',MASTER\_LOG\_FILE='master-bin.000002',MASTER\_LOG\_POS=245;

**B数据库**

mysql -u root

CHANGE MASTER TO MASTER\_HOST='xxx.xxx.xxx.xxx(A的ip)',MASTER\_USER='slaveuser', MASTER\_PASSWORD='slaveuser',MASTER\_LOG\_FILE='master-bin.000002',MASTER\_LOG\_POS=245;

**8.**  **A和B开启同步**

mysql -u root

start slave;

**9.**  **解锁**

mysql -u root

UNLOCK TABLES;

**10.查看A和B同步状态**

mysql -u root

SHOW SLAVE STATUS\G;

