

# 基于 CentOS 7 的 MySQL 搭建与配置

修订记录:

版本号	修订人	修订日期	修订内容

# 目 录

一、软件版本 .....	1
二、删除 MariaDB .....	1
三、yum 方式安装 .....	1
3.1 下载存储库 .....	1
3.2 修改安装设置 .....	1
3.3 安装 MySQL .....	2
3.4 启动 MySQL .....	2
3.5 进行安全设置 .....	2
四、编译安装 .....	2
4.1 添加用户 .....	2
4.2 创建所需文件（夹） .....	2
4.3 安装依赖包 .....	2
4.4 主备安装包 .....	3
4.5 安装 MySQL .....	3
4.7 配置环境变量 .....	3
4.8 编辑配置文件 .....	3
4.9 配置开机启动 .....	3
4.10 初始化 MySQL .....	3
4.11 进行安全设置 .....	3
五、主备复制 .....	4
5.1 修改主设备配置文件 .....	4
5.2 创建同步用户 .....	4
5.3 获取二进制日志坐标 .....	4
5.4 修改从设备配置文件 .....	4
5.5 在从设备上设置主配置 .....	4
5.6 启动复制 .....	5

一、软件版本

MySQL 版本：5.7.25

二、删除 MariaDB

新装系统中往往会 MariaDB 的相关组件。为保证后续正常安装、配置及使用，在安装 MySQL 前需要先删除它们。使用命令：

```
yum remove -y mariadb*
```

三、yum 方式安装

3.1 下载存储库

到 MySQL 官网上下载存储库，目前使用的是 mysql80-community-release-el7-1.noarch.rpm，下载后安装它，使用命令：

```
yum localinstall -y mysql80-community-release-el7-1.noarch.rpm
```

3.2 修改安装设置

默认情况下会安装 MySQL8.0，因此需要在安装前进行修改。

方式一：

修改/etc/yum.repos.d/mysql-community.repo 文件，找到如下内容，修改 enabled 的数值，0 是不安装，1 是安装。

原文	<pre>[mysql57-community] name=MySQL 5.7 Community Server baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/\$basearch/ enabled=0 gpgcheck=1 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql  [mysql80-community] name=MySQL 8.0 Community Server baseurl=http://repo.mysql.com/yum/mysql-8.0-community/el/7/\$basearch/ enabled=1 gpgcheck=1 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql</pre>
改后	<pre>[mysql57-community] name=MySQL 5.7 Community Server baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/\$basearch/ enabled=1 gpgcheck=1 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql  [mysql80-community] name=MySQL 8.0 Community Server baseurl=http://repo.mysql.com/yum/mysql-8.0-community/el/7/\$basearch/ enabled=0 gpgcheck=1 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql</pre>

方式二：

用 yum-config-manager 命令来修改，需要提前安装 yum-utils。使用命令：

```
yum install -y yum-utils
```

```
yum-config-manager --enable mysql57-community
yum-config-manager --disable mysql80-community
```

### 3.3 安装 MySQL

此时可以进行安装，安装的版本就是刚刚修改的版本，使用命令：

```
yum install -y mysql-server
```

### 3.4 启动 MySQL

安装完成后即可启动 MySQL，使用命令：

```
systemctl enable mysqld
systemctl start mysqld
```

### 3.5 进行安全设置

MySQL 5.7 及更高版本在启动后，会自动产生一个临时密码，我们首先需要获取这个密码，才能进行后续操作。使用命令：

```
grep root@localhost /var/log/mysqld.log
```

在显示结果行尾找到临时密码，然后进行安全设置。使用命令：

```
mysql_secure_installation
```

此时需要输入 root 的密码，也就是刚刚查到的临时密码。然后 MySQL 会要求我们重新设置密码，密码有长度及复杂度要求。设置完成后会有以下内容需要设置：

```
修改 root 密码? (“是” 按 y 或 Y, “否” 按其他键)
Change the password for root ? (Press y|Y for Yes, any other key for No) :
移除匿名用户? (“是” 按 y 或 Y, “否” 按其他键)
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
不允许 root 用户远程登陆? (“是” 按 y 或 Y, “否” 按其他键)
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
删除测试数据库及其访问? (“是” 按 y 或 Y, “否” 按其他键)
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
现在重载权限表? (“是” 按 y 或 Y, “否” 按其他键)
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
```

## 四、编译安装

### 4.1 添加用户

首先添加用户，使用命令

```
groupadd mysql
useradd -g mysql -s /bin/false -M mysql
```

### 4.2 创建所需文件（夹）

创建后续需要的日志文件、数据文件夹等，使用命令：

```
touch /var/log/mysqld.log
chown mysql.mysql /var/log/mysqld.log
mkdir -p /data/mydata
chown -R mysql.mysql /data/mydata
mkdir -p /usr/local/mysql
chown -R mysql.mysql /usr/local/mysql
mkdir -p /var/run/mysqld/
chown -R mysql.mysql /var/run/mysqld/
```

### 4.3 安装依赖包

正式编译安装前，安装编译所需的软件或依赖包。使用命令：

```
yum install -y cmake gcc-c++ ncurses-devel openssl-devel
```

#### 4.4 主备安装包

到官网下载对应版本的安装包，本文使用的是 mysql-boost-5.7.25.tar.gz。然后解压下载下来的安装包。使用命令：

```
tar -zxvf mysql-boost-5.7.25.tar.gz
```

#### 4.5 安装 MySQL

编译过程中会产生大量文件，为方便管理，手动新建文件夹。使用命令：

```
cd mysql-5.7.25
mkdir install
cd install
```

然后根据需要进行编译。使用命令：

```
cmake .. \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci \
-DMYSQL_DATADIR=/data/mydata \
-DSYSCONFDIR=/etc \
-DWITH_BOOST=../boost/ \
-DWITH_SSL=system \
-DWITH_ZLIB=system \
-DWITH_ARCHIVE_STORAGE_ENGINE=1 \
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
-DWITH_INNOBASE_STORAGE_ENGINE=1
```

编译完成后安装。使用命令：

```
make && make install
```

#### 4.7 配置环境变量

新建文件/etc/profile.d/mysql.sh，在文件中写入如下内容：

```
MYSQL_HOME=/usr/local/mysql
PATH=$MYSQL_HOME/bin:$PATH
```

完成后更新环境变量。使用命令：

```
source /etc/profile
```

#### 4.8 编辑配置文件

然后修改配置文件/etc/my.cnf，写入如下内容：

```
[mysqld]
datadir=/data/mydata
socket=/tmp/mysql.sock
symbolic-links=0
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

#### 4.9 配置开机启动

使用命令：

```
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
chkconfig --add mysqld
chkconfig mysqld on
```

#### 4.10 初始化 MySQL

使用命令：

```
mysqld --defaults-file=/etc/my.cnf --initialize --user=mysql
```

#### 4.11 进行安全设置

MySQL 5.7 及更高版本在启动后, 会自动产生一个临时密码, 我们首先需要获取这个密码, 才能进行后续操作。  
使用命令:

```
grep root@localhost /var/log/mysqld.log
```

在显示结果行尾找到临时密码, 然后进行安全设置。使用命令:

```
mysql_secure_installation
```

此时需要输入 root 的密码, 也就是刚刚查到的临时密码。然后 MySQL 会要求我们重新设置密码。  
密码设置完成后出现一段文字, 询问是否安装验证密码插件:

原文	翻译
VALIDATE PASSWORD PLUGIN can be used to test passwords and improve security. It checks the strength of password and allows the users to set only those passwords which are secure enough. Would you like to setup VALIDATE PASSWORD plugin?	验证密码插件可用于测试密码和提高安全性。它检查密码的强度, 并允许用户只设置那些足够安全的密码。您想安装验证密码插件吗?

此处根据个人意愿安装 (推荐安装)。之后是与 yum 安装方式相同的安全设置, 参考 2.5 节。

## 五、主备复制

### 5.1 修改主设备配置文件

在/etc/my.cnf 文件中添加如下内容:

```
log-bin=mysql-bin
server-id=1
innodb_flush_log_at_trx_commit=1
sync_binlog=1
```

### 5.2 创建同步用户

首先进入 MySQL 中, 使用命令:

```
mysql -uroot -p
```

进入 MySQL 后创建账户并赋权:

```
create user 'replication_user_name'@'slave_IP' identified by 'replication_password';
grant replication slave on *.* to 'replication_user_name'@'slave_IP';
```

### 5.3 获取二进制日志坐标

首先刷新所有表和阻止写语句:

```
flush tables with read lock;
```

查看信息:

```
show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
mysql-bin.000001	154			

### 5.4 修改从设备配置文件

在/etc/my.cnf 文件中添加如下内容:

```
server-id=2
```

### 5.5 在从设备上设置主配置

进入从设备上的 MySQL 程序

```
change master to
master_host='master_host_name',
master_user='replication_user_name',
master_password='replication_password',
master_log_file='recorded_log_file_name',
master_log_pos=recorded_log_position;
```

## 5.6 启动复制

对从设备的配置完成后，即可开启复制：

```
start slave;
```

然后查看同步状态：

```
show slave status\G;
```

看到下面两行的状态都是 YES 时说明同步成功：

```
Slave_IO_Running: Yes
```

```
Slave_SQL_Running: Yes
```