



链滴

## 2-MySQL 安装和基本使用

作者: [Carey](#)

原文链接: <https://ld246.com/article/1612342441734>

来源网站: 链滴

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



## 2 MySQL安装和基本使用

### 2.1 MySQL 介绍

#### 2.1.1 MySQL 历史

1979年: TcX公司 Monty Widenius, Unireg

1996年: 发布MySQL1.0, Solaris版本, Linux版本

1999年: MySQL AB公司, 瑞典

2003年: MySQL 5.0版本, 提供视图、存储过程等功能

2008年: Sun公司以10亿美元收购MySQL

2009年: Oracle公司以 75 亿美元收购 sun 公司

2009年: Monty成立MariaDB

#### 2.2.2 MySQL系列

##### 2.2.2.1 MySQL 的三大主要分支

- mysql
- mariadb
- percona Server

## 2.2.2.2 官方网址

<https://www.mysql.com/>

<http://mariadb.org/>

<https://www.percona.com>

## 2.2.2.3 官方文档

<https://dev.mysql.com/doc/>

<https://mariadb.com/kb/en/>

<https://www.percona.com/software/mysql-database/percona-server>

## 2.2.2.4 版本演变

MySQL: 5.1 --> 5.5 --> 5.6 --> 5.7 --> 8.0

MariaDB: 5.1 --> 5.5 --> 10.0 --> 10.1 --> 10.2 --> 10.3 --> 10.4 --> 10.5

MySQL被Sun收购后,搞了个过渡的6.0版本,没多久就下线了,后来被Oracle收购后,终于迎来了像的5.6版本,之后就是5.7、8.0版本。由于6.0版本号已被用过,7.x系列版本专用于NDB Cluster,因新版本号从8.0开始。

## 2.2.3 MySQL的特性

- 开源免费
- 插件式存储引擎: 也称为“表类型”, 存储管理器有多种实现版本, 功能和特性可能均略有差别; 户可根据需要灵活选择,MySQL5.5.5开始InnoDB引擎是MySQL默认引擎

MyISAM ==> Aria

InnoDB ==> XtraDB

- 单进程, 多线程

#判断多线程

```
[09:57:23 root@centos8 ~]#grep -i threads /proc/1074/status
```

```
Threads: 39
```

- 诸多扩展和新特性
- 提供了较多测试组件

## 2.2 MySQL 安装方式介绍和快速安装

### 2.2.1 安装方式介绍

- 程序包管理器管理的程序包

- 源代码编译安装
- 二进制格式的程序包：展开至特定路径，并经过简单配置后即可使用

## 2.2.2 RPM包安装MySQL

CentOS 安装光盘

项目官方: <https://downloads.mariadb.org/mariadb/repositories/>

国内镜像: <https://mirrors.tuna.tsinghua.edu.cn/mariadb/yum/>

<https://mirrors.tuna.tsinghua.edu.cn/mysql/yum/>

### CentOS 8: 安装光盘直接提供

- mysql-server: 8.0
- mariadb-server : 10.3.17

### CentOS 7: 安装光盘直接提供

- mariadb-server: 5.5 服务器包
- mariadb 客户端工具包

### CentOS 6:

- mysql-server: 5.1 服务器包
- mysql 客户端工具包

### 范例: CentOS 7 安装MySQL5.7

```
[10:54:25 root@centos7 ~]#cat /etc/yum.repos.d/mysql57.repo
[mysql57]
name=mysql57
baseurl=https://mirrors.tuna.tsinghua.edu.cn/mysql/yum/mysql-5.7-community-el7-x86_64/
gpgcheck=0
[10:55:08 root@centos7 ~]#yum list | grep mysql57
mysql-community-client.i686          5.7.33-1.el7          mysql57
mysql-community-client.x86_64       5.7.33-1.el7          mysql57
mysql-community-common.i686         5.7.33-1.el7          mysql57
mysql-community-common.x86_64       5.7.33-1.el7          mysql57
mysql-community-devel.i686          5.7.33-1.el7          mysql57
mysql-community-devel.x86_64        5.7.33-1.el7          mysql57
mysql-community-embedded.i686       5.7.33-1.el7          mysql57
mysql-community-embedded.x86_64     5.7.33-1.el7          mysql57
mysql-community-embedded-compat.i686 5.7.33-1.el7          mysql57
mysql-community-embedded-compat.x86_64 5.7.33-1.el7          mysql57
mysql-community-embedded-devel.i686  5.7.33-1.el7          mysql57
mysql-community-embedded-devel.x86_64 5.7.33-1.el7          mysql57
mysql-community-libs.i686           5.7.33-1.el7          mysql57
mysql-community-libs.x86_64         5.7.33-1.el7          mysql57
mysql-community-libs-compat.i686     5.7.33-1.el7          mysql57
mysql-community-libs-compat.x86_64   5.7.33-1.el7          mysql57
```

```

mysql-community-server.x86_64      5.7.33-1.el7      mysql57
mysql-community-test.x86_64      5.7.33-1.el7      mysql57
[10:55:17 root@centos7 ~]#yum install mysql-community-server -y
[11:00:53 root@centos7 ~]#systemctl enable --now mysqld
[11:04:08 root@centos7 ~]#ss -ntl
State      Recv-Q Send-Q      Local Address:Port      Peer Address:Port
LISTEN     0      128          *:111                    *.*
LISTEN     0      128          *:22                     *.*
LISTEN     0      100         127.0.0.1:25            *.*
LISTEN     0      80           [::]:3306                [::]:*
LISTEN     0      128          [::]:111                 [::]:*
LISTEN     0      128          [::]:22                  [::]:*
LISTEN     0      100         [::1]:25                 [::]:*
[11:04:26 root@centos7 ~]#mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
[11:05:15 root@centos7 ~]#grep password /var/log/mysqld.log
2021-01-28T03:04:05.376619Z 1 [Note] A temporary password is generated for root@localho
t: j9eNwN#IsToD
2021-01-28T03:05:15.225574Z 2 [Note] Access denied for user 'root'@'localhost' (using pass
ord: NO)
#修改初始密码方法1
[11:37:14 root@centos7 ~]#mysql -uroot -p'j9eNwN#IsToD'
mysql> alter user root@'localhost' identified by 'Zhangzhuo@0705';
Query OK, 0 rows affected (0.00 sec)
mysql> status
#修改初始密码方法2
[11:38:28 root@centos7 ~]#mysqladmin -uroot -p'j9eNwN#IsToD' password 'Zhangzhuo@07
5'

```

### 范例：centos7安装Mariadb

```

[11:42:44 root@centos7 ~]#cat /etc/yum.repos.d/mariadb.repo
[mariadb105]
name=mariadb105
baseurl=https://mirrors.tuna.tsinghua.edu.cn/mariadb/yum/10.5/centos/7/x86_64/
gpgcheck=0
[11:47:06 root@centos7 ~]#yum install -y MariaDB-server
[11:47:43 root@centos7 ~]#systemctl enable --now mariadb.service
[11:48:08 root@centos7 ~]#mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.5.8-MariaDB MariaDB Server

```

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
MariaDB [(none)]> status
```

```
-----
mysql Ver 15.1 Distrib 10.5.8-MariaDB, for Linux (x86_64) using readline 5.1
```

```

Connection id:      3
Current database:

```

```
Current user:      root@localhost
SSL:              Not in use
Current pager:    stdout
Using outfile:    "
Using delimiter:  ;
Server:           MariaDB
Server version:   10.5.8-MariaDB MariaDB Server
Protocol version: 10
Connection:       Localhost via UNIX socket
Server characterset: latin1
Db characterset:  latin1
Client characterset: utf8
Conn. characterset: utf8
UNIX socket:      /var/lib/mysql/mysql.sock
Uptime:           18 sec
```

```
Threads: 2 Questions: 4 Slow queries: 0 Opens: 16 Open tables: 10 Queries per second avg: 0.222
```

## 2.3 初始化脚本提高安全性

运行脚本: `mysql_secure_installation`

```
设置数据库管理员root口令
禁止root远程登录
删除anonymous用户帐号
删除test数据库
```

## 2.4 MYSQL组成

### 2.4.1 客户端程序

- `mysql`: 交互式或非交互式的CLI工具
- `mysqldump`: 备份工具, 基于mysql协议向mysqld发起查询请求, 并将查得的所有数据转换成insert等写操作语句保存文本文件中
- `mysqladmin`: 基于mysql协议管理mysqld
- `mysqlimport`: 数据导入工具

MyISAM存储引擎的管理工具:

- `myisamchk`: 检查MyISAM库
- `myisampack`: 打包MyISAM表, 只读

### 2.4.2 服务器端程序

- `mysqld_safe`
- `mysqld`
- `mysqld_multi` 多实例, 示例: `mysqld_multi --example`

## 2.4.3 用户账号

mysql用户账号由两部分组成:

```
'USERNAME'@'HOST'
```

说明:

HOST限制此用户可通过哪些远程主机连接mysql服务器

支持使用通配符:

% 匹配任意长度的任意字符,相当于shell中\*, 示例: 172.16.0.0/255.255.0.0 或 172.16.%.%  
\_ 匹配任意单个字符,相当于shell中?

## 2.4.4 mysql 客户端命令

### 2.4.4.1 mysql 运行命令类型

- 客户端命令: 本地执行, 每个命令都完整形式和简写格式

```
mysql> \h, help  
mysql> \u, use  
mysql> \s, status  
mysql> \!, system
```

- 服务端命令: 通过mysql协议发往服务器执行并取回结果, 命令末尾都必须使用命令结束符号, 默认为分号

#示例:  
mysql>SELECT VERSION();

### 2.4.4.2 mysql 使用模式

- 交互模式
- 脚本模式:

```
mysql -uUSERNAME -pPASSWORD < /path/somefile.sql  
cat /path/somefile.sql | mysql -uUSERNAME -pPASSWORD  
mysql>source /path/from/somefile.sql
```

### 2.4.4.3 mysql命令使用格式

```
mysql [OPTIONS] [database]
```

mysql客户端常用选项:

-A, --no-auto-rehash 禁止补全  
-u, --user= 用户名,默认为root  
-h, --host= 服务器主机,默认为localhost  
-p, --passowrd= 用户密码,建议使用-p,默认为空密码

-P, --port= 服务器端口  
-S, --socket= 指定连接socket文件路径  
-D, --database= 指定默认数据库  
-C, --compress 启用压缩  
-e "SQL " 执行SQL命令  
-V, --version 显示版本  
-v --verbose 显示详细信息  
--print-defaults 获取程序默认使用的配置

登录系统:

```
[11:50:57 root@centos7 ~]#mysql -uroot -p123456 #默认不写空密码
```

运行mysql命令:

```
mysql>use mysql  
mysql> select user();  
mysql>SELECT User,Host>Password FROM user;
```

范例: mysql的配置文件, 修改提示符

```
[12:30:27 root@centos7 ~]#vim /etc/my.cnf  
#添加这行  
[mysql]  
prompt="\\r:\\m:\\s(\\u@\\h) [\\d]>\\_ "  
[12:30:27 root@centos7 ~]#mysql -uroot -p'Zhangzhuo@0705'  
12:31:06(root@localhost) [(none)]>
```

范例: 配置所有MySQL客户端自动登录

```
[12:35:05 root@centos7 ~]#vim /etc/my.cnf  
[client]  
user=root  
password=Zhangzhuo@0705
```

## 2.4.4.4 mysqladmin命令

mysqladmin 命令格式

```
mysqladmin [OPTIONS] command command...
```

范例:

```
#查看mysql服务是否正常, 如果正常提示mysqld is alive  
[12:36:10 root@centos7 ~]#mysqladmin -uroot -p'Zhangzhuo@0705' ping  
mysqladmin: [Warning] Using a password on the command line interface can be insecure.  
mysqld is alive  
#关闭mysql服务, 但mysqladmin命令无法开启  
[12:36:30 root@centos7 ~]#mysqladmin -uroot -p'Zhangzhuo@0705' shutdown  
#创建数据库testdb  
[12:37:39 root@centos7 ~]#mysqladmin -uroot -p'Zhangzhuo@0705' create testdb  
#删除数据库testdb  
[12:38:21 root@centos7 ~]#mysqladmin -uroot -p'Zhangzhuo@0705' drop testdb  
#修改root密码
```



```
[12:39:26 root@centos7 ~]#mysqladmin -uroot -p'Zhangzhuo@0705' password 'Admin@123'  
#日志滚动,生成新文件/var/lib/mysql/mariadb-bin.00000N  
[12:39:33 root@centos7 ~]#mysqladmin -uroot -p'Admin@123' flush-logs
```

## 2.4.4.5 服务器端配置

**服务器端(mysql): 工作特性有多种配置方式**

- 命令行选项:
- 配置文件: 类ini格式,集中式的配置, 能够为mysql的各应用程序提供配置信息

**服务器端配置文件:**

- /etc/my.cnf            #Global选项
- /etc/mysql/my.cnf    #Global选项
- ~/.my.cnf            #User-specific 选项

**配置文件格式:**

- [mysqld]
- [mysqld\_safe]
- [mysqld\_multi]
- [mysql]
- [mysqldump]
- [server]
- [client]

**格式:**

parameter = value

**说明:**

1和- 相同  
1, ON, TRUE意义相同, 0, OFF, FALSE意义相同,无区分大小写

## 2.4.4.6 socket地址

服务器监听的两种 socket 地址:

- ip socket: 监听在tcp的3306端口, 支持远程通信, 侦听3306/tcp端口可以在绑定有一个或全部接IP上
- unix sock: 监听在sock文件上, 仅支持本机通信, 如: /var/lib/mysql/mysql.sock)说明: host为localhost 时自动使用unix sock

## 2.4.4.7 关闭mysqld网络连接

只侦听本地客户端, 所有客户端和服务器的交互都通过一个socket文件实现, socket的配置存放

在/var/lib/mysql/mysql.sock) 可在/etc/my.cnf修改

范例:

```
vim /etc/my.cnf
[mysqld]
skip-networking=1      #表示关闭网络连接1关闭0开启
bind_address=127.0.0.1 #绑定端口
```

## 2.5 通用二进制格式安装 MySQL

### 2.5.1 准备用户

```
[12:48:56 root@centos7 ~]#useradd -r -s /sbin/nologin -d /data/mysql mysql
```

### 2.5.2 准备数据目录，建议使用逻辑卷

```
#可选做，后面的脚本mysql_install_db可自动生成此目录
[12:50:20 root@centos7 ~]#mkdir -p /data/mysql
[12:50:24 root@centos7 ~]#chown mysql: /data/mysql/
```

### 2.5.3 准备二进制程序

```
[13:56:35 root@centos7 ~]#tar xvf mysql-5.6.50-linux-glibc2.12-x86_64.tar.gz -C /usr/local/
[13:56:35 root@centos7 ~]#cd /usr/local/
[13:57:43 root@centos7 local]#ln -sv mysql-5.6.50-linux-glibc2.12-x86_64 mysql
[13:58:14 root@centos7 local]#chown -R root: /usr/local/mysql
```

### 2.5.4 准备配置文件

```
[14:16:18 root@centos7 mysql]#cd /usr/local/mysql
[14:16:20 root@centos7 mysql]#cp -d support-files/my-default.cnf /etc/my.cnf
[14:17:54 root@centos7 mysql]#vim /etc/my.cnf
[mysqld]
datadir = /data/mysql
innodb_file_per_table = on #在mariadb5.5以上版的是默认值，可不加
skip_name_resolve = no    #禁止主机名解析，建议使用
```

### 2.5.5 创建数据库文件

```
[14:21:09 root@centos7 mysql]#./scripts/mysql_install_db --datadir=/data/mysql --user=mysql
```

```
[14:28:12 root@centos7 mysql]#ls /data/mysql/ -l
total 110600
-rw-rw---- 1 mysql mysql 12582912 Jan 28 14:28 ibdata1
-rw-rw---- 1 mysql mysql 50331648 Jan 28 14:28 ib_logfile0
-rw-rw---- 1 mysql mysql 50331648 Jan 28 14:28 ib_logfile1
drwx----- 2 mysql mysql 4096 Jan 28 14:28 mysql
drwx----- 2 mysql mysql 4096 Jan 28 14:28 performance_schema
drwx----- 2 mysql mysql 6 Jan 28 14:27 test
```

## 2.5.6 准备服务脚本，并启动服务

```
[14:28:17 root@centos7 mysql]#cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql
[14:28:58 root@centos7 mysql]#chkconfig --add mysqld
[14:29:22 root@centos7 mysql]#service mysqld start
```

```
#如果有对应的service 文件可以执行下面
cp /usr/local/mysql/support-files/systemd/mariadb.service
/usr/lib/systemd/system/
systemctl daemon-reload
systemctl enable --now mariadb
```

## 2.5.7 PATH路径

```
[14:30:50 root@centos7 mysql]#echo 'PATH=/usr/local/mysql/bin:$PATH' >/etc/profile.d/mysql.sh
[14:32:09 root@centos7 mysql]#. /etc/profile.d/mysql.sh
```

## 2.5.8 安全初始化

```
[14:34:03 root@centos7 mysql]#mysql_secure_installation
```

# 2.6 实战案例：一键安装mysql-5.6二进制包的脚本

## 2.6.1 离线安装mysql-5.6二进制包的脚本

```
#!/bin/bash
#
#*****
#Author:zhangzhuo
#QQ: 1191400158
#Date: 2021-01-28
#FileName: install_mysql5.6.sh
#URL: https://www.zhangzhuo.ltd
#Description: The test script
#Copyright (C): 2021 All rights reserved
#*****
NAME=$1
NAME=`echo $NAME | grep -Eo '[^/]+$`
DIRNAME=`echo $NAME | sed -rn 's/(.*)tar.*\1/p`
DIR=/usr/local
DATADIR=/data/mysql

install_mysql_user(){
if id mysql &&>/dev/null;then
echo "MYSQL用户已经存在，无需从新创建！"
else
useradd -r -s /sbin/nologin -d $DATADIR mysql
fi
}
}
```

```

install_mysql_datadir(){
if [ -d $DATADIR ];then
echo "${DATADIR}目录已经存在，无需从新创建！"
else
mkdir -p $DATADIR
chown mysql: $DATADIR
fi
}

install_mysql_my(){
cat /etc/profile.d/mysql.sh
./etc/profile.d/mysql.sh
cp $DIR/mysql/support-files/mysql.server /etc/init.d/mysqld
chkconfig --add mysqld
service mysqld start
else
echo -e "\033[1;31m二进制文件不存在请检查\033[0m";exit
fi
}

install_mysql_user
install_mysql_datadir
install_mysql

```

## 2.6.2 在线安装mysql-5.6二进制包的脚本

```

#!/bin/bash
#
#*****
#Author:zhangzhuo
#QQ: 1191400158
#Date: 2021-01-28
#FileName: install_online_mysql5.6_for_centos.sh
#URL: https://www.zhangzhuo.ltd
#Description: The test script
#Copyright (C): 2021 All rights reserved
#*****
./etc/init.d/functions

DIR=`pwd`
URL=https://repo.huaweicloud.com/mysql/Downloads/MySQL-5.6/mysql-5.6.50-linux-glibc2.
2-x86_64.tar.gz
NAME=`echo $URL | grep -Eo '[^/]+$`
DIRNAME=`echo $NAME | sed -rn 's/(.*)tar.*\1/p`
DATA_DIR=/data/mysql
FULL_NAME=${DIR}/${NAME}

install_mysql_wget(){
rpm -q wget &>/dev/null || yum install -y wget
yum install -y -q libaio perl-Data-Dumper autoconf
if [ -f ${FULL_NAME} ];then
action "安装文件已经存在不需要下载"
else

```

```

action "安装文件不存在开始下载" false
wget $URL || { action "下载失败, 异常退出" false;exit 10; }
fi
}

install_mysql_user(){
if id mysql &>/dev/null;then
action "MYSQL用户已经存在, 无需从新创建! "
else
useradd -r -s /sbin/nologin -d $DATA_DIR mysql
action "mysql用户创建成功"
fi
}

install_mysql_datadir(){
if [ -d $DATA_DIR ];then
action "${DATA_DIR}目录已经存在, 无需从新创建! "
chown mysql: $DATA_DIR
else
mkdir -p $DATA_DIR
chown mysql: $DATA_DIR
fi
}

install_mysql_my(){
cat /dev/null
echo "PATH=/usr/local/mysql/bin: '$PATH' >/etc/profile.d/mysql.sh
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
chkconfig --add mysqld
service mysqld start
source /etc/profile.d/mysql.sh
mysqladmin ping && action "服务启动成功" || action "服务启动失败请检查" false
fi
}

install_mysql_wget
install_mysql_user
install_mysql_datadir
install_mysql

```

## 2.7 实战案例：二进制安装安装MySQL 5.7 和 MySQL8.0

### 2.7.1 安装相关包

```
[18:20:20 root@centos7 ~]#yum install libaio numactl-libs
```

### 2.7.2 用户和组

```
[18:27:30 root@centos7 ~]#useradd -r -s /sbin/nologin -d /data/mysql mysql
[18:46:02 root@centos7 local]#mkdir -p /data/mysql
[18:46:10 root@centos7 local]#chown mysql: /data/mysql
```

### 2.7.3 准备程序文件

```
[18:29:25 root@centos7 ~]#wget https://repo.huaweicloud.com/mysql/Downloads/MySQL-5.7.31-linux-glibc2.12-x86_64.tar.gz
[18:35:03 root@centos7 ~]#tar xf mysql-5.7.31-linux-glibc2.12-x86_64.tar.gz -C /usr/local/
[18:36:36 root@centos7 ~]#cd /usr/local/
[18:36:59 root@centos7 local]#ln -s mysql-5.7.31-linux-glibc2.12-x86_64 mysql
[18:37:37 root@centos7 local]#chown -R root: /usr/local/mysql
```

### 2.7.4 准备环境变量

```
[18:38:55 root@centos7 local]#echo 'PATH=/usr/local/mysql/bin:$PATH' >/etc/profile.d/mysql.sh
[18:39:08 root@centos7 local]#. /etc/profile.d/mysql.sh
```

### 2.7.5 准备配置文件

```
[18:43:53 root@centos7 local]#grep -Ev '^#|$\`' /etc/my.cnf
[mysqld]
datadir=/data/mysql
skip_name_resolve=1
socket=/data/mysql/mysql.sock
log-error=/var/log/mysql.log
[client]
socket=/data/mysql/mysql.sock
!includedir /etc/my.cnf.d
[18:51:01 root@centos7 local]#touch /var/log/mysql.log
[18:51:16 root@centos7 local]#chown mysql: /var/log/mysql.log
```

### 2.7.6 生成数据库文件,并提取root密码

```
[18:46:25 root@centos7 local]#mysqld --initialize --user=mysql --datadir=/data/mysql
[18:52:20 root@centos7 local]#grep password /var/log/mysql.log
2021-01-28T10:51:39.397031Z 1 [Note] A temporary password is generated for root@localhost: ggtorsg(H6wj)
```

### 2.7.7 准备服务脚本和启动

```
[19:17:02 root@centos7 ~]#cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
[19:17:27 root@centos7 ~]#chkconfig --add mysqld
[19:27:59 root@centos7 ~]#service mysqld start
```

### 2.7.8 修改口令

```
[19:28:59 root@centos7 ~]#mysqladmin -uroot -p'gEG?>*c=E8oL' password 123456
```

### 2.7.9 测试登录

```
[19:53:56 root@centos7 ~]#mysql -uroot -p123456
```

## 2.8 实战案例：一键安装MySQL5.7 和 MySQL8.0 二进制的脚本

### 2.8.1 在线安装脚本

```
#!/bin/bash
#
#*****
#Author:zhangzhuo
#QQ: 1191400158
#Date: 2021-01-28
#FileName: install_mysql5.7-8.0.sh
#URL: https://www.zhangzhuo.ltd
#Description: The test script
#Copyright (C): 2021 All rights reserved
#*****
./etc/init.d/functions

DIR=`pwd`
URL=https://repo.huaweicloud.com/mysql/Downloads/MySQL-5.7/mysql-5.7.31-linux-glibc2.2-x86_64.tar.gz
NAME=`echo $URL | grep -Eo '[^/]+$`
DIRNAME=`echo $NAME | sed -rn 's/(.*)tar.*\1/p`
DATA_DIR=/data/mysql
FULL_NAME=${DIR}/${NAME}
PASSWORD=123456

install_mysql_wget(){
rpm -q wget &>/dev/null || yum install -y wget
yum install -y -q libaio perl-Data-Dumper autoconf
if [ -f ${FULL_NAME} ];then
action "安装文件已经存在不需要下载"
else
action "安装文件不存在开始下载" false
wget $URL || { action "下载失败, 异常退出" false;exit 10; }
fi
}

install_mysql_user(){
if id mysql &>/dev/null;then
action "MYSQL用户已经存在, 无需从新创建! "
else
useradd -r -s /sbin/nologin -d $DATA_DIR mysql
action "mysql用户创建成功"
fi
}

install_mysql_datadir(){
if [ -d $DATA_DIR ];then
action "${DATA_DIR}目录已经存在, 无需从新创建! "
chown mysql: $DATA_DIR
else
mkdir -p $DATA_DIR
```

```

chown mysql: $DATA_DIR
action "${DATA_DIR}目录创建完成!"
fi
}

install_mysql_my(){
cat /etc/profile.d/mysql.sh
source /etc/profile.d/mysql.sh
mysqld --initialize --user=mysql --datadir=/data/mysql &>/dev/null
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
chkconfig --add mysqld
service mysqld start && action "服务启动完成" || action "服务启动失败" false
mysql_set_passwd
fi
}

install_mysql_wget
install_mysql_user
install_mysql_datadir
install_mysql

```

## 2.9 源码编译安装 MySQL 5.6

建议：内存4G以上,CPU 核数越多越好

### 2.9.1 安装相关依赖包

```

yum -y install gcc gcc-c++ cmake bison bison-devel zlib-devel libcurl-devel libarchive-devel
oost-devel ncurses-devel gnutls-devel libxml2-devel openssl-devel libevent-devel libaio-dev
l perl-Data-Dumper

```

### 2.9.2 做准备用户和数据目录

```

useradd -r -s /sbin/nologin -d /data/mysql mysql

```

### 2.9.3 准备数据库目录

```

mkdir /data/mysql
chown mysql.mysql /data/mysql

```

### 2.9.4 源码编译安装

编译安装说明

利用cmake编译,而利用传统方法, cmake的重要特性之一是其独立于源码(out-of-source)的编译功, 即编译工作可以在另一个指定的目录中而非源码目录中进行, 这可以保证源码目录不受任何一次编的影响, 因此在同一个源码树上可以进行多次不同的编译, 如针对于不同平台编译

编译选项:<https://dev.mysql.com/doc/refman/5.7/en/source-configuration-options.html>

#### 2.9.4.1 下载并解压缩源码包



## 2.9.4.2 源码编译安装mariadb

```
cmake . -DCMAKE_INSTALL_PREFIX=/apps/mysql -DMYSQL_DATADIR=/data/mysql/ -DSYSC
NFDIR=/etc/ -DMYSQL_USER=mysql -DWITH_INNOBASE_STORAGE_ENGINE=1 -DWITH_ARC
HIVE_STORAGE_ENGINE=1 -DWITH_BLACKHOLE_STORAGE_ENGINE=1 -DWITH_PARTITION_S
ORAGE_ENGINE=1 -DWITHOUT_MROONGA_STORAGE_ENGINE=1 -DWITH_DEBUG=0 -DWIT
_READLINE=1 -DWITH_SSL=system -DWITH_ZLIB=system -DWITH_LIBWRAP=0 -DENABLED
LOCAL_INFILE=1 -DMYSQL_UNIX_ADDR=/data/mysql/mysql.sock -DDEFAULT_CHARSET=utf8
DDEFAULT_COLLATION=utf8_general_ci
[21:17:56 root@centos7 mysql-5.6.49]#make
[21:20:36 root@centos7 mysql-5.6.49]#make install
```

提示：如果出错，执行rm -f CMakeCache.txt

## 2.9.5 准备环境变量

```
[21:22:01 root@centos7 mysql-5.6.49]#echo 'PATH=/apps/mysql/bin:$PATH' >/etc/profile.d
mysql.sh
[21:22:08 root@centos7 mysql-5.6.49]#source /etc/profile.d/mysql.sh
```

## 2.9.6 生成数据库文件

```
[21:22:29 root@centos7 mysql]#scripts/mysql_install_db --datadir=/data/mysql --user=mysql
```

## 2.9.7 准备配置文件

```
[21:23:44 root@centos7 mysql]#cp -b /apps/mysql/support-files/my-default.cnf /etc/my.cnf
```

## 2.9.8 准备启动脚本,并启动服务

```
[21:24:46 root@centos7 mysql]#cp /apps/mysql/support-files/mysql.server /etc/init.d/mysqld
[21:25:18 root@centos7 mysql]#chkconfig --add mysqld
[21:25:30 root@centos7 mysql]#service mysqld start
Starting MySQL.Logging to '/data/mysql/centos7.err'.
. SUCCESS!
```

## 2.9.9 安全初始化

```
[21:26:52 root@centos7 mysql]#mysql_secure_installation
```

## 2.10 基于 dockcer 容器创建MySQL

范例:

```
[14:39:24 root@centos7 ~]#docker run --name mysql -d -p 3306:3306 -e MYSQL_ROOT_PAS
WORD=123456 mysql:8.0
```

## 2.11 MySQL多实例

## 2.11.1 多实例介绍

- 什么是数据库多实例

- MySQL多实例就是在同一台服务器上同时开启多个不同的服务端口（如：3306、3307等），同时运行多个MySQL服务进程，这些服务进程通过不同的Socket监听不同的服务端口来提供服务。多实例可能是MySQL的不同版本，也可能是MySQL的同一版本实现

- 多实例的好处

- 可有效利用服务器资源。当单个服务器资源有剩余时，可以充分利用剩余资源提供更多的服务且可以实现资源的逻辑隔离节约服务器资源。例如公司服务器资源紧张，但是数据库又需要各自尽量立的服务，并且还需要到主从复制等技术，多实例就是最佳选择

- 多实例弊端

- 存在资源互相抢占的问题。比如：当某个数据库实例并发很高或者SQL查询慢时，整个实例会消耗大量的CPU、磁盘I/O等资源，导致服务器上面其他的数据库实例在提供服务的质量也会下降，所以具的需求要根据自己的实际情况而定。

## 2.11.2 实战案例 1: CentOS 8 实现 MySQL 8.0 二进制安装的多实例

本案例适用于以版本

mysql-8.0.23-linux-glibc2.12-x86\_64.tar.xz  
mysql-5.7.33-linux-glibc2.12-x86\_64.tar.gz

### 实战目标

CentOS8 二进制安装MySQL8.0,并实现三个实例

### 环境说明

一台系统CentOS 8.X主机

### 前提准备

关闭SELinux  
关闭防火墙  
时间同步

### 2.11.2.1 实现步骤

#### 下载MySQL二进制文件并解压缩

```
[14:48:49 root@centos8 ~]##wget http://mirrors.163.com/mysql/Downloads/MySQL-8.0/mysql-8.0.23-linux-glibc2.12-x86_64.tar
[14:49:55 root@centos8 ~]#tar -xvf mysql-8.0.21-linux-glibc2.12-x86_64.tar -C /usr/local/
[14:50:06 root@centos8 ~]#cd /usr/local/
[14:50:11 root@centos8 local]#ln -sv mysql-8.0.21-linux-glibc2.12-x86_64 mysql
[14:50:46 root@centos8 local]##ll mysql/
total 408
drwxr-xr-x 2 7161 31415 4096 Jun 17 2020 bin
drwxr-xr-x 2 7161 31415 55 Jun 17 2020 docs
```

```
drwxr-xr-x 3 7161 31415 282 Jun 17 2020 include
drwxr-xr-x 6 7161 31415 201 Jun 17 2020 lib
-rw-r--r-- 1 7161 31415 404759 Jun 17 2020 LICENSE
drwxr-xr-x 4 7161 31415 30 Jun 17 2020 man
-rw-r--r-- 1 7161 31415 687 Jun 17 2020 README
drwxr-xr-x 28 7161 31415 4096 Jun 17 2020 share
drwxr-xr-x 2 7161 31415 77 Jun 17 2020 support-files
```

## 创建用户和组配置权限

```
[14:50:58 root@centos8 local]#useradd -r -s /sbin/nologin mysql
[14:51:45 root@centos8 local]#chown -R mysql: /usr/local/mysql/
[14:52:17 root@centos8 local]#ll /usr/local/mysql/
total 408
drwxr-xr-x 2 mysql mysql 4096 Jun 17 2020 bin
drwxr-xr-x 2 mysql mysql 55 Jun 17 2020 docs
drwxr-xr-x 3 mysql mysql 282 Jun 17 2020 include
drwxr-xr-x 6 mysql mysql 201 Jun 17 2020 lib
-rw-r--r-- 1 mysql mysql 404759 Jun 17 2020 LICENSE
drwxr-xr-x 4 mysql mysql 30 Jun 17 2020 man
-rw-r--r-- 1 mysql mysql 687 Jun 17 2020 README
drwxr-xr-x 28 mysql mysql 4096 Jun 17 2020 share
drwxr-xr-x 2 mysql mysql 77 Jun 17 2020 support-files
```

## 配置环境变量

```
[14:52:29 root@centos8 local]#echo 'export PATH=/usr/local/mysql/bin:$PATH' >/etc/profile.
/mysql.sh
[14:53:33 root@centos8 local]#. /etc/profile.d/mysql.sh
```

## 创建各实例数据存放的目录

```
[14:53:42 root@centos8 local]#mkdir -p /mysql/{3306,3307,3308}
[14:54:46 root@centos8 local]#chown -R mysql: /mysql/
[14:55:13 root@centos8 local]#ll /mysql/
total 0
drwxr-xr-x 2 mysql mysql 6 Jan 30 14:54 3306
drwxr-xr-x 2 mysql mysql 6 Jan 30 14:54 3307
drwxr-xr-x 2 mysql mysql 6 Jan 30 14:54 3308
[14:55:18 root@centos8 local]#tree /mysql/
/mysql/
├── 3306
├── 3307
└── 3308
```

## 初始化各实例数据库文件

#针对每个实例初始化

```
[14:55:22 root@centos8 local]#mysqld --initialize-insecure --user=mysql --datadir=/mysql/3
06
[14:56:34 root@centos8 local]#mysqld --initialize-insecure --user=mysql --datadir=/mysql/3
07
[14:56:46 root@centos8 local]#mysqld --initialize-insecure --user=mysql --datadir=/mysql/3
08
[14:57:38 root@centos8 local]#tree /mysql/ -d
```

```
/mysql/
├── 3306
│   ├── #innodb_temp
│   ├── mysql
│   ├── performance_schema
│   └── sys
├── 3307
│   ├── #innodb_temp
│   ├── mysql
│   ├── performance_schema
│   └── sys
└── 3308
    ├── #innodb_temp
    ├── mysql
    ├── performance_schema
    └── sys
```

## 准备配置文件/etc/my.cnf

```
[14:57:50 root@centos8 local]#file `which mysqld_multi`
/usr/local/mysql/bin/mysqld_multi: Perl script text executable
[14:59:17 root@centos8 local]#vim /etc/my.cnf
[mysqld_multi]
mysqld=/usr/local/mysql/bin/mysqld_safe
mysqladmin=/usr/local/mysql/bin/mysqladmin
```

```
[mysqld3306]
datadir=/mysql/3306
port=3306
socket=/mysql/3306/mysql3306.sock
pid-file=/mysql/3306/mysql3306.pid
log-error=/mysql/3306/mysql3306.log
```

```
[mysqld3307]
datadir=/mysql/3307
port=3307
socket=/mysql/3307/mysql3307.sock
pid-file=/mysql/3307/mysql3307.pid
log-error=/mysql/3307/mysql3307.log
```

```
[mysqld3308]
datadir=/mysql/3308
port=3308
socket=/mysql/3308/mysql3308.sock
pid-file=/mysql/3308/mysql3308.pid
log-error=/mysql/3308/mysql3308.log
```

## 启动多实例

```
#说明:用 mysqld_multi start N 启动多个实例,
#注意数字N和my.cnf中的[mysqldN]对应, 示例:1-3就是启动[mysqld1]、[mysqld2]、[mysqld3]
置段的MySQL实例
```

```
#启动三个MySQL实例
```

```
[root@centos8 ~]#mysqld_multi start 3306
[root@centos8 ~]#mysqld_multi start 3307
[root@centos8 ~]#mysqld_multi start 3308
```

# 或者用下面命令批量启动多个实例

```
[15:25:47 root@centos8 ~]#mysqld_multi start 3306-3308
```

```
[15:24:45 root@centos8 ~]#ss -ntl
```

State	Recv-Q	Send-Q	Local Address:Port
LISTEN	0	128	0.0.0.0:111
	0.0.0.0:*		
LISTEN	0	128	0.0.0.0:22
	0.0.0.0:*		
LISTEN	0	70	*:33060
	*:*		
LISTEN	0	128	*:3306
	*:*		
LISTEN	0	128	*:3307
	*:*		
LISTEN	0	128	*:3308
	*:*		
LISTEN	0	128	:::111
	:::*		
LISTEN	0	128	:::22
	:::*		

### 关闭多实例

```
[15:25:47 root@centos8 ~]#mysqld_multi stop 3306-3308
```

### 安全加固

#批量修改多个实例root密码

```
[15:28:43 root@centos8 ~]#for i in {3306..3308};do mysqladmin -S /mysql/$i/mysql$i.sock pas word 123456;done
```

#批量验证密码连接

```
[15:28:46 root@centos8 ~]#for i in {3306..3308};do mysqladmin -uroot -p123456 -S /mysql/$i mysql$i.sock ping;done
```

### 配置开机启动多实例

```
[15:29:46 root@centos8 ~]#vim /etc/rc.d/rc.local
```

```
./etc/profile.d/mysql.sh
```

```
mysqld_multi start 3306-3308
```

```
[15:31:18 root@centos8 ~]#chmod +x /etc/rc.d/rc.local
```

rc.local文件中首行必须有shebang机制才可以生效

```
#!/bin/bash
```

## 2.12.3 实战案例 2: CentOS 8 实现mariadb的yum安装的多实例

### 实战目的

## CentOS 8 yum安装mariadb-10.3.17并实现三个实例

### 环境要求

一台系统CentOS 8.X主机

### 前提准备

关闭SELinux  
关闭防火墙  
时间同步

### 2.12.3.1 实现步骤

#### 安装mariadb

```
[10:27:54 root@centos8 ~]#yum install -y mariadb-server
```

#### 准备三个实例的目录

```
[10:27:54 root@centos8 ~]#mkdir -pv /mysql/{3306,3307,3308}/{data,etc,socket,log,bin,pid}
[10:28:29 root@centos8 ~]#chown -R mysql:mysql /mysql
[10:28:35 root@centos8 ~]#tree /mysql/
```

```
/mysql/
├── 3306
│   ├── bin
│   ├── data
│   ├── etc
│   ├── log
│   ├── pid
│   └── socket
├── 3307
│   ├── bin
│   ├── data
│   ├── etc
│   ├── log
│   ├── pid
│   └── socket
└── 3308
    ├── bin
    ├── data
    ├── etc
    ├── log
    ├── pid
    └── socket
```

21 directories, 0 files

#### 生成数据库文件

```
[10:30:21 root@centos8 ~]#mysql_install_db --user=mysql --datadir=/mysql/3306/data
[10:30:21 root@centos8 ~]#mysql_install_db --user=mysql --datadir=/mysql/3307/data
[10:30:21 root@centos8 ~]#mysql_install_db --user=mysql --datadir=/mysql/3308/data
```

## 准备配置文件

```
[10:30:28 root@centos8 ~]#vim /mysql/3306/etc/my.cnf
[mysqld]
port=3306
datadir=/mysql/3306/data
socket=/mysql/3306/socket/mysql.sock
log-error=/mysql/3306/log/mysql.log
pid-file=/mysql/3306/pid/mysql.pid
#重复上面步骤设置3307, 3308
[10:34:49 root@centos8 ~]#sed 's/3306/3307/' /mysql/3306/etc/my.cnf >/mysql/3307/etc/my
cnf
[10:35:01 root@centos8 ~]#sed 's/3306/3308/' /mysql/3306/etc/my.cnf >/mysql/3308/etc/my
cnf
```

## 准备启动脚本

```
[10:58:36 root@centos8 ~]#vim /mysql/3306/bin/mysqld
#!/bin/bash
port=3306
mysql_user="root"
mysql_pwd="123456"
cmd_path="/usr/bin"
mysql_basedir="/mysql"
mysql_sock="${mysql_basedir}/${port}/socket/mysql.sock"

function_start_mysql(){
if [ ! -e "$mysql_sock" ];then
printf "Starting MySQL...\n"
${cmd_path}/mysqld_safe --defaults-file=${mysql_basedir}/${port}/etc/my.cnf &>/dev/null &
else
printf "MySQL is running...\n"
exit
fi
}

function_stop_mysql(){
if [ ! -e "$mysql_sock" ];then
printf "MySQL is stopped...\n"
exit
else
printf "Stoping MySQL...\n"
${cmd_path}/mysqladmin -u${mysql_user} -p${mysql_pwd} -S ${mysql_sock} shutdown
fi
}
function_restart_mysql(){
printf "Restarting MySQL...\n"
function_stop_mysql
sleep 2
function_start_mysql
}

case $1 in
start)
```

```
function_start_mysql
;;
stop)
function_stop_mysql
;;
restart)
function_restart_mysql
;;
*)
printf "Usage:${mysql_basedir}/${port}/bin/mysqld {start|stop|restart}\n"
esac
[10:59:22 root@centos8 ~]#chmod +x /mysql/3306/bin/mysqld
#重复上述过程，分别建立3307，3308的启动脚本
```

## 启动服务

```
[11:06:50 root@centos8 ~]#/mysql/3306/bin/mysqld start
[11:06:50 root@centos8 ~]#/mysql/3307/bin/mysqld start
[11:06:50 root@centos8 ~]#/mysql/3308/bin/mysqld start
[11:06:50 root@centos8 ~]#ss -ntl
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	128	0.0.0.0:111	0.0.0.0:*
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*
LISTEN	0	80	*:3306	**
LISTEN	0	80	*:3307	**
LISTEN	0	80	*:3308	**
LISTEN	0	128	:::111	:::*
LISTEN	0	128	2.8.3.4.7 登录实例	:::22

## 登录实例

```
[11:07:51 root@centos8 ~]#mysql -h 127.0.0.1 -P 3308
确认登录的端口
MariaDB [(none)]> show variables like 'port';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| port          | 3308 |
+-----+-----+
1 row in set (0.001 sec)
#关闭数据库，需要手动输入root的密码请把stop中的-p后面的变量删除
[11:09:53 root@centos8 ~]#/mysql/3308/bin/mysqld stop
Stopping MySQL...
Enter password:
```

## 修改root密码

```
#加上root的口令
[11:12:18 root@centos8 ~]#mysqladmin -uroot -S /mysql/3306/socket/mysql.sock password
23456
[11:12:18 root@centos8 ~]#mysqladmin -uroot -S /mysql/3307/socket/mysql.sock password
23456
[11:12:18 root@centos8 ~]#mysqladmin -uroot -S /mysql/3308/socket/mysql.sock password
23456
#或者登录mysql,执行下面也可以
```



```
Mariadb>update mysql.user set password=password("centos") where user='root';  
Mariadb>flush privileges;
```

## 测试连接

```
[11:12:40 root@centos8 ~]#mysql -uroot -p -S /mysql/3306/socket/mysql.sock  
Enter password:  
#提示输入口令才能登录
```