

CentOS 7 安装 MySQL

环境

- CentOS 7.1 (64-bit system)
- MySQL 5.6.24

CentOS 安装

参考: <http://www.waylau.com/centos-7-installation-and-configuration/>

依赖

MySQL 依赖 libaio, 所以先要安装 libaio

```
yum search libaio # 检索相关信息 yum install libaio # 安装依赖包
```

成功安装, 提示如下:

```
[root@bogon /]# yum install libaio
已加载插件: fastestmirrorLoadingmirrorspeedsfromcachedhostfile
* base: mirrors.yun-idc.com * extras: mirrors.163.com * updates: mirrors.163.com
软件包 libaio-0.3.109-12.el7.x86_64 已安装并且是最新版本无须任何处理
```

检查 MySQL 是否已安装

```
yum list installed | grep mysql
```

如果有, 就先全部卸载, 命令如下:

```
yum -y remove mysql-libs.x86_64
```

若有多个依赖文件则依次卸载。当结果显示为 Complete! 即卸载完毕。

下载

下载 MySQL Yum Repository

地址为 <http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm>

执行

```
wget http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm
```

如果提示-bash: wget: 未找到命令, 请先执行 yum install wget 安装 wget

安装

添加 MySQL Yum Repository

添加 MySQL Yum Repository 到你的系统 repository 列表中, 执行

```
yum localinstall mysql-community-release-el7-5.noarch.rpm
```

显示

```
[root@bogon software]# yum localinstall mysql-community-release-el7-5.noarch.rpm
已加载插件: fastestmirror
正在检查 mysql-community-release-el7-5.noarch.rpm: mysql-community-release-el7-5.noarch.rpm 将被安装
正在解决依赖关系--> 正在检查事务---> 软件包 mysql-community-release.noarch.0.el7-5 将被安装--> 解决依赖关系完成依赖关系解决
```

```
===== Package
架构    版本    源
mysql-community-release
noarch el7-5 /mysql-community-release-el7-5.noarch.rpm
=====正在安装:
```

5.noarch 4.3k事务概要

=====安装 1 软件包

总计: 4.3k安装大小: 4.3kIs this ok [y/d/N]

提示是否 OK , 输入 y

Is this ok [y/d/N]: yDownloading packages:Running transaction checkRunning transaction

testTransaction test succeededRunning transaction 正在安装 : mysql-community-release-el7-

5.noarch 1/1 验证中 : mysql-community-release-el7-5.noarch

1/1已安装: mysql-community-release.noarch 0:el7-5完毕!

提示“完成!”, 则说明 源添加成功。

验证下是否添加成功

```
yum repolist enabled | grep"mysql.*-community.*"
```

可以看到下面内:

```
[root@bogon software]# yum repolist enabled | grep "mysql.*-community.*"mysql-connectors-
community/x86_64 MySQL Connectors Community 1mysql-tools-community/x86_64
MySQL Tools Community 1mysql56-community/x86_64 MySQL 5.6
Community Server 13
```

选择要启用 MySQL 版本

查看 MySQL 版本, 执行

```
yum repolist all | grep mysql
```

可以看到 5.5, 5.7 版本是默认禁用的, 因为现在最新的稳定版是 5.6

```
[root@bogon software]# yum repolist all | grep mysqlmysql-connectors-community/x86_64 MySQL
Connectors Community 启用: 14mysql-connectors-community-source MySQL Connectors
Community - Sourc 禁用mysql-tools-community/x86_64 MySQL Tools Community 启
用: 17mysql-tools-community-source MySQL Tools Community - Source 禁用mysql55-
community/x86_64 MySQL 5.5 Community Server 禁用mysql55-community-source
MySQL 5.5 Community Server - Sourc 禁用mysql56-community/x86_64 MySQL 5.6 Community
Server 启用: 139mysql56-community-source MySQL 5.6 Community Server - Sourc
禁用mysql57-community-dmr/x86_64 MySQL 5.7 Community Server Develop 禁用mysql57-community-
dmr-source MySQL 5.7 Community Server Develop 禁用
```

可以通过类似下面的语句来启动某些版本

```
yum-config-manager --disable mysql56-communityyum-config-manager --enable mysql57-community-
dmr
```

或者通过修改 /etc/yum.repos.d/mysql-community.repo 文件

```
# Enable to use MySQL 5.6[mysql56-community]name=MySQL 5.6 Community
Serverbaseurl=http://repo.mysql.com/yum/mysql-5.6-
community/el/7/$basearch/enabled=1gpgcheck=1gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

其中 enabled=0 是指禁用, enabled=1 指启用。

注意: 任何时候, 只能启用一个版本。

执行

```
yum repolist enabled | grep mysql
```

查看当前的启动的 MySQL 版本

```
[root@bogon software]# yum repolist enabled | grep mysqlmysql-connectors-community/x86_64
MySQL Connectors Community 14mysql-tools-community/x86_64 MySQL Tools
Community 17mysql56-community/x86_64 MySQL 5.6 Community
Server 139
```

本例, 我们启用的是 5.6 版本。

通过 Yum 来安装 MySQL

执行

```
yum install mysql-community-server
```

Yum 会自动处理 MySQL 与其他组件的依赖关系:

```
[root@bogon software]# yum install mysql-community-server已加载插件: fastestmirrorLoading
mirror speeds from cached hostfile * base: mirrors.yun-idc.com * extras: mirrors.163.com *
updates: mirrors.163.com正在解决依赖关系--> 正在检查事务---> 软件包 mysql-community-
server.x86_64.0.5.6.24-3.el7 将被 安装--> 正在处理依赖关系 mysql-community-common(x86-64) =
5.6.24-3.el7, 它被软件包 m
ysql-community-server-5.6.24-3.el7.x86_64 需要--> 正在处理依赖关系 mysql-community-client(x86-64)
= 5.6.24-3.el7, 它被软件包 m
ysql-community-server-5.6.24-3.el7.x86_64 需要--> 正在处理依赖关系 perl(warnings), 它被软件包
mysql-community-server-5.6.24-3.
el7.x86_64 需要--> 正在处理依赖关系 perl(strict), 它被软件包 mysql-community-server-5.6.24-3.el
7.x86_64 需要--> 正在处理依赖关系 perl(if), 它被软件包 mysql-community-server-5.6.24-3.el7.x8
6_64 需要--> 正在处理依赖关系 perl(Sys::Hostname), 它被软件包 mysql-community-server-5.6.
24-3.el7.x86_64 需要--> 正在处理依赖关系 perl(POSIX), 它被软件包 mysql-community-server-5.6.24-
3.el7
.x86_64 需要--> 正在处理依赖关系 perl(Getopt::Long), 它被软件包 mysql-community-server-5.6.2
4-3.el7.x86_64 需要--> 正在处理依赖关系 perl(File::Temp), 它被软件包 mysql-community-server-5.6.24-
3.el7.x86_64 需要--> 正在处理依赖关系 perl(File::Spec), 它被软件包 mysql-community-server-5.6.24-
3.el7.x86_64 需要--> 正在处理依赖关系 perl(File::Path), 它被软件包 mysql-community-server-5.6.24-
3.el7.x86_64 需要--> 正在处理依赖关系 perl(File::Copy), 它被软件包 mysql-community-server-5.6.24-
3.el7.x86_64 需要--> 正在处理依赖关系 perl(File::Basename), 它被软件包 mysql-community-server-5.6
.24-3.el7.x86_64 需要--> 正在处理依赖关系 perl(Fcntl), 它被软件包 mysql-community-server-5.6.24-
3.el7
.x86_64 需要--> 正在处理依赖关系 perl(Data::Dumper), 它被软件包 mysql-community-server-5.6.2
4-3.el7.x86_64 需要--> 正在处理依赖关系 perl(DBI), 它被软件包 mysql-community-server-5.6.24-3.el7.x
86_64 需要--> 正在处理依赖关系 net-tools, 它被软件包 mysql-community-server-5.6.24-3.el7.x
86_64 需要--> 正在处理依赖关系 /usr/bin/perl, 它被软件包 mysql-community-server-5.6.24-3.e
17.x86_64 需要--> 正在检查事务---> 软件包 mysql-community-client.x86_64.0.5.6.24-3.el7 将被 安装--
> 正在处理依赖关系 mysql-community-libs(x86-64) = 5.6.24-3.el7, 它被软件包 mys
ql-community-client-5.6.24-3.el7.x86_64 需要--> 正在处理依赖关系 perl(Exporter), 它被软件包 mysql-
community-client-5.6.24-3.
el7.x86_64 需要---> 软件包 mysql-community-common.x86_64.0.5.6.24-3.el7 将被 安装---> 软件包 net-
tools.x86_64.0.2.0-0.17.20131004git.el7 将被 安装---> 软件包 perl.x86_64.4.5.16.3-285.el7 将被 安
装--> 正在处理依赖关系 perl-libs = 4:5.16.3-285.el7, 它被软件包 4:perl-5.16.3-285.
el7.x86_64 需要--> 正在处理依赖关系 perl(Socket) >= 1.3, 它被软件包 4:perl-5.16.3-285.el7.x86_6
4 需要--> 正在处理依赖关系 perl(Scalar::Util) >= 1.10, 它被软件包 4:perl-5.16.3-285.el
7.x86_64 需要--> 正在处理依赖关系 perl-macros, 它被软件包 4:perl-5.16.3-285.el7.x86_64 需要--> 正在
处理依赖关系 perl-libs, 它被软件包 4:perl-5.16.3-285.el7.x86_64 需要--> 正在处理依赖关系
perl(threads::shared), 它被软件包 4:perl-5.16.3-285.el7.x86
_64 需要--> 正在处理依赖关系 perl(threads), 它被软件包 4:perl-5.16.3-285.el7.x86_64 需要--> 正在处理
依赖关系 perl(constant), 它被软件包 4:perl-5.16.3-285.el7.x86_64 需
要--> 正在处理依赖关系 perl(Time::Local), 它被软件包 4:perl-5.16.3-285.el7.x86_64
需要--> 正在处理依赖关系 perl(Time::HiRes), 它被软件包 4:perl-5.16.3-285.el7.x86_64
需要--> 正在处理依赖关系 perl(Storable), 它被软件包 4:perl-5.16.3-285.el7.x86_64 需
要--> 正在处理依赖关系 perl(Socket), 它被软件包 4:perl-5.16.3-285.el7.x86_64 需要--> 正在处理依赖关系
perl(Scalar::Util), 它被软件包 4:perl-5.16.3-285.el7.x86_64
需要--> 正在处理依赖关系 perl(Pod::Simple::XHTML), 它被软件包 4:perl-5.16.3-285.el7.
x86_64 需要--> 正在处理依赖关系 perl(Pod::Simple::Search), 它被软件包 4:perl-5.16.3-285.el7
.x86_64 需要--> 正在处理依赖关系 perl(Filter::Util::Call), 它被软件包 4:perl-5.16.3-285.el7.
x86_64 需要--> 正在处理依赖关系 perl(Carp), 它被软件包 4:perl-5.16.3-285.el7.x86_64 需要--> 正在处理
依赖关系 libperl.so() (64bit), 它被软件包 4:perl-5.16.3-285.el7.x86_6
4 需要---> 软件包 perl-DBI.x86_64.0.1.627-4.el7 将被 安装--> 正在处理依赖关系 perl(RPC::PlServer)
```

```

>= 0.2001, 它被软件包 perl-DBI-1.627-4.
e17.x86_64 需要--> 正在处理依赖关系 perl(RPC::PlClient) >= 0.2000, 它被软件包 perl-DBI-1.627-4.
e17.x86_64 需要---> 软件包 perl-Data-Dumper.x86_64.0.2.145-3.e17 将被 安装---> 软件包 perl-File-
Path.noarch.0.2.09-2.e17 将被 安装---> 软件包 perl-File-Temp.noarch.0.0.23.01-3.e17 将被 安装--->
软件包 perl-Getopt-Long.noarch.0.2.40-2.e17 将被 安装--> 正在处理依赖关系 perl(Pod::Usage) >=
1.14, 它被软件包 perl-Getopt-Long-2.40-
2.e17.noarch 需要--> 正在处理依赖关系 perl(Text::ParseWords), 它被软件包 perl-Getopt-Long-2.40-2.
e17.noarch 需要---> 软件包 perl-PathTools.x86_64.0.3.40-5.e17 将被 安装--> 正在检查事务---> 软件包
mariadb-libs.x86_64.1.5.5.41-2.e17_0 将被 取代---> 软件包 mysql-community-libs.x86_64.0.5.6.24-
3.e17 将被 舍弃---> 软件包 perl-Carp.noarch.0.1.26-244.e17 将被 安装---> 软件包 perl-
Exporter.noarch.0.5.68-3.e17 将被 安装---> 软件包 perl-Filter.x86_64.0.1.49-3.e17 将被 安装--->
软件包 perl-PlRPC.noarch.0.0.2020-14.e17 将被 安装--> 正在处理依赖关系 perl(Net::Daemon) >= 0.13,
它被软件包 perl-PlRPC-0.2020-14.
e17.noarch 需要--> 正在处理依赖关系 perl(Net::Daemon::Test), 它被软件包 perl-PlRPC-0.2020-14.e1
7.noarch 需要--> 正在处理依赖关系 perl(Net::Daemon::Log), 它被软件包 perl-PlRPC-0.2020-14.e17
.noarch 需要--> 正在处理依赖关系 perl(Compress::Zlib), 它被软件包 perl-PlRPC-0.2020-14.e17.n
oarch 需要---> 软件包 perl-Pod-Simple.noarch.1.3.28-4.e17 将被 安装--> 正在处理依赖关系
perl(Pod::Escapes) >= 1.04, 它被软件包 1:perl-Pod-Simple-3.
28-4.e17.noarch 需要--> 正在处理依赖关系 perl(Encode), 它被软件包 1:perl-Pod-Simple-3.28-
4.e17.noarc
h 需要---> 软件包 perl-Pod-Usage.noarch.0.1.63-3.e17 将被 安装--> 正在处理依赖关系 perl(Pod::Text)
>= 3.15, 它被软件包 perl-Pod-Usage-1.63-3.e
17.noarch 需要--> 正在处理依赖关系 perl-Pod-Perldoc, 它被软件包 perl-Pod-Usage-1.63-3.e17.noar
ch 需要---> 软件包 perl-Scalar-List-Utils.x86_64.0.1.27-248.e17 将被 安装---> 软件包 perl-
Socket.x86_64.0.2.010-3.e17 将被 安装---> 软件包 perl-Storable.x86_64.0.2.45-3.e17 将被 安装--->
软件包 perl-Text-ParseWords.noarch.0.3.29-4.e17 将被 安装---> 软件包 perl-Time-
HiRes.x86_64.4.1.9725-3.e17 将被 安装---> 软件包 perl-Time-Local.noarch.0.1.2300-2.e17 将被 安装-
--> 软件包 perl-constant.noarch.0.1.27-2.e17 将被 安装---> 软件包 perl-libs.x86_64.4.5.16.3-
285.e17 将被 安装---> 软件包 perl-macros.x86_64.4.5.16.3-285.e17 将被 安装---> 软件包 perl-
threads.x86_64.0.1.87-4.e17 将被 安装---> 软件包 perl-threads-shared.x86_64.0.1.43-6.e17 将被 安
装--> 正在检查事务---> 软件包 perl-Encode.x86_64.0.2.51-7.e17 将被 安装---> 软件包 perl-IO-
Compress.noarch.0.2.061-2.e17 将被 安装--> 正在处理依赖关系 perl(Compress::Raw::Zlib) >= 2.061, 它
被软件包 perl-IO-Comp
ress-2.061-2.e17.noarch 需要--> 正在处理依赖关系 perl(Compress::Raw::Bzip2) >= 2.061, 它被软件包
perl-IO-Com
press-2.061-2.e17.noarch 需要---> 软件包 perl-Net-Daemon.noarch.0.0.48-5.e17 将被 安装---> 软件包
perl-Pod-Escapes.noarch.1.1.04-285.e17 将被 安装---> 软件包 perl-Pod-Perldoc.noarch.0.3.20-4.e17
将被 安装--> 正在处理依赖关系 perl(parent), 它被软件包 perl-Pod-Perldoc-3.20-4.e17.noarch
需要--> 正在处理依赖关系 perl(HTTP::Tiny), 它被软件包 perl-Pod-Perldoc-3.20-4.e17.no
arch 需要---> 软件包 perl-podlators.noarch.0.2.5.1-3.e17 将被 安装--> 正在检查事务---> 软件包 perl-
Compress-Raw-Bzip2.x86_64.0.2.061-3.e17 将被 安装---> 软件包 perl-Compress-Raw-
Zlib.x86_64.1.2.061-4.e17 将被 安装---> 软件包 perl-HTTP-Tiny.noarch.0.0.033-3.e17 将被 安装--->
软件包 perl-parent.noarch.1.0.225-244.e17 将被 安装--> 解决依赖关系完成依赖关系解决
===== Package

```

架构	版本	源	大小
=====正在安装:			
mysql-community-libs	x86_64 5.6.24-3.e17	mysql56-community	2.0M 替换
mariadb-libs.x86_64	1:5.5.41-2.e17_0	mysql-community-server	x86_64 5.6.24-3.e17
mysql56-community	58M为依赖而安装:	mysql-community-client	x86_64 5.6.24-3.e17
mysql56-community	19M	mysql-community-common	x86_64 5.6.24-3.e17
community	256k	net-tools	x86_64 2.0-0.17.20131004git.e17 base 304k
perl	x86_64 4:5.16.3-285.e17	base	8.0M perl-Carp

```

noarch 1.26-244.el7          base          19k perl-Compress-Raw-Bzip2
x86_64 2.061-3.el7          base          32k perl-Compress-Raw-Zlib x86_64 1:2.061-
4.el7          base          57k perl-DBI          x86_64 1.627-4.el7
base          802k perl-Data-Dumper      x86_64 2.145-3.el7          base
47k perl-Encode          x86_64 2.51-7.el7          base          1.5M perl-
Exporter          noarch 5.68-3.el7          base          28k perl-File-Path
noarch 2.09-2.el7          base          26k perl-File-Temp          noarch 0.23.01-
3.el7          base          56k perl-Filter          x86_64 1.49-3.el7
base          76k perl-Getopt-Long          noarch 2.40-2.el7          base
56k perl-HTTP-Tiny          noarch 0.033-3.el7          base          38k perl-IO-
Compress          noarch 2.061-2.el7          base          260k perl-Net-Daemon
noarch 0.48-5.el7          base          51k perl-PathTools          x86_64 3.40-
5.el7          base          82k perl-PlRPC          noarch 0.2020-14.el7
base          36k perl-Pod-Escapes          noarch 1:1.04-285.el7          base
50k perl-Pod-Perldoc          noarch 3.20-4.el7          base          87k perl-Pod-
Simple          noarch 1:3.28-4.el7          base          216k perl-Pod-Usage
noarch 1.63-3.el7          base          27k perl-Scalar-List-Utils x86_64 1.27-
248.el7          base          36k perl-Socket          x86_64 2.010-3.el7
base          49k perl-Storable          x86_64 2.45-3.el7          base
77k perl-Text-ParseWords  noarch 3.29-4.el7          base          14k perl-Time-
HiRes          x86_64 4:1.9725-3.el7          base          45k perl-Time-Local
noarch 1.2300-2.el7          base          24k perl-constant          noarch 1.27-
2.el7          base          19k perl-libs          x86_64 4:5.16.3-285.el7
base          687k perl-macros          x86_64 4:5.16.3-285.el7          base
42k perl-parent          noarch 1:0.225-244.el7          base          12k perl-
podlators          noarch 2.5.1-3.el7          base          112k perl-threads
x86_64 1.87-4.el7          base          49k perl-threads-shared  x86_64 1.43-
6.el7          base          39k事务概要

```

=====安装 2 软件包

(+37 依赖软件包)总下载量: 92MIs this ok [y/d/N]:

输入 y 继续下载相关文件;

-----总计

```

1.1 MB/s | 92 MB 01:24从 file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql 检索密钥导入 GPG key
0x5072E1F5: 用户ID      : "MySQL Release Engineering build@oss.oracle.com>" 指纹      : a4a9
4068 76fc bd3c 4567 70c8 8c71 8d3b 5072 elf5 软件包      : mysql-community-release-el7-5.noarch
(@/mysql-community-release-el7-5.noarch) 来自      : file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
是否继续? [y/N]:

```

遇到上述提示,输入 y 继续,执行完成会提示“完毕! ”。此时MySQL 安装完成,它包含了 mysql-community-server、mysql-community-client、mysql-community-common、mysql-community-libs 四个包。

执行

```
rpm-qimysql-community-server.x86_64 0:5.6.24-3.el7
```

执行

```
whereis mysql
```

可以看到 MySQL 的安装目录是 /usr/bin/

```
[root@localhost ~]# whereis mysqlmysql: /usr/bin/mysql /usr/lib64/mysql /usr/share/mysql
/usr/share/man/man1/mysql.1.gz
```

启动和关闭 MySQL Server

启动 MySQL Server

```
systemctl start mysqld
```

查看 MySQL Server 状态

```
systemctl status mysqld
```

关闭 MySQL Server

```
systemctl stop mysqld
```

测试是否安装成功

```
mysql
```

可以进入 mysql 命令行界面

```
[root@bogon software]# mysqlWelcome to the MySQL monitor.  Commands end with ; or \g.Your MySQL connection id is2Server version: 5.6.24 MySQL Community Server (GPL)Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.Oracle is a registered trademark of Oracle Corporation and/or itsaffiliates. Other names may be trademarks of their respectiveowners.Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.mysql>
```

防火墙设置

远程访问 MySQL， 需开放默认端口号 3306.

方式1: iptables (CentOS 7.x版本之前用法，不推荐)

打开 iptables 的配置文件:

```
vi /etc/sysconfig/iptables
```

修改

```
*filter:INPUT ACCEPT [0:0]:FORWARD ACCEPT [0:0]:OUTPUT ACCEPT [0:0]-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT-A INPUT -p icmp -j ACCEPT-A INPUT -i lo -j ACCEPT-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT-A INPUT -j REJECT --reject-with icmp-host-prohibited-A FORWARD -j REJECT --reject-with icmp-host-prohibitedCOMMIT
```

在里面加入这2行:

```
-A RH-Firewall-1-INPUT -mstate -state NEW -m tcp -p tcp -dport 3306 -j ACCEPT-A RH-Firewall-1-INPUT -mstate -state NEW -m udp -p udp -dport 3306 -j ACCEPT
```

改为

```
*filter:INPUT ACCEPT [0:0]:FORWARD ACCEPT [0:0]:OUTPUT ACCEPT [0:0]-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT-A INPUT -p icmp -j ACCEPT-A INPUT -i lo -j ACCEPT-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT-A RH-Firewall-1-INPUT -m state -state NEW -m tcp -p tcp -dport 3306 -j ACCEPT-A RH-Firewall-1-INPUT -m state -state NEW -m udp -p udp -dport 3306 -j ACCEPT-A INPUT -j REJECT --reject-with icmp-host-prohibited-A FORWARD -j REJECT --reject-with icmp-host-prohibitedCOMMIT
```

如果该 iptables 配置文件 不存在，先执行 `yum install iptables-services` 安装

执行 iptables 重启生效

```
service iptables restart
```

方式2: firewall-cmd (推荐)

执行

```
firewall-cmd --permanent --zone=public --add-port=3306/tcpfirewall-cmd --permanent --zone=public --add-port=3306/udp
```

这样就开放了相应的端口。

执行

```
firewall-cmd --reload
```

使最新的防火墙设置规则生效。

MySQL 安全设置

服务器启动后，可以执行

```
mysql_secure_installation;
```

看到如下提示

```
[root@bogon software]# mysql_secure_installationNOTE: RUNNING ALL PARTS OF THIS SCRIPT IS
RECOMMENDED FOR ALL MySQL      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP
CAREFULLY!Inordertologinto MySQL to secure it, we'll need the currentpassword for the root
user.  If you've just installed MySQL, andyou 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):

此时输入 root 原始密码（初始化安装的话为空），接下来，为了安全，MySQL 会提示你重置 root 密码，移除其他用
户账号，禁用 root 远程登录，移除 test 数据库，重新加载 privilege 表格等，你只需输入 y 继续执行即可。
OK, successfully used password, moving on...Setting the root password ensures that nobody can
log into the MySQLroot user without the proper authorisation.Set root password? [Y/n]
yNewpassword:Re-enter newpassword>Passwordupdated successfully!Reloading privilege tables..
... Success!Bydefault, a MySQL installation has an anonymous user, allowing anyone tologinto
MySQL withouthavingto have a useraccount created forthem.  This is intended onlyfor testing,
andto make the installationgo a bit smoother.  You should remove them before moving into
aproduction environment.Remove anonymous users? [Y/n] y ... Success!Normally, root should
only be allowed toconnectfrom'localhost'.  Thisensures that someone cannot guess at the root
passwordfrom the network.Disallow root login remotely? [Y/n] y ... Success!Bydefault, MySQL
comes with a database named 'test' that anyone canaccess.  This is also intended onlyfor
testing, and should be removedbefore moving into a production environment.Remove
testdatabaseandaccessto it? [Y/n] y - Dropping testdatabase...ERROR1008 (HY000) at line 1:
Can't drop database 'test'; database doesn't exist ... Failed!  Notcritical, keep moving... -
Removing privilegesontestdatabase... ... Success!Reloading the privilege tables will ensure
that all changes made so farwill take effect immediately.Reload privilege tablesnow? [Y/n] y
... Success!All done!  If you've completed all of the above steps, your MySQLinstallation
should now be secure.Thanks for using MySQL!Cleaning up...
```

至此，整个 MySQL 安装完成。

远程访问设置

创建一个普通用户 sa，密码是 some_pass

```
CREATEUSER'sa'@'%'IDENTIFIEDBY'some_pass';
```

给这个用户授予 SELECT, INSERT, UPDATE, DELETE 的远程访问的权限，这个账号一般用于提供给实施的系统访问

```
GRANTSELECT,INSERT,UPDATE,DELETEON *.* TO'sa'@'%';
```

创建一个管理员用户 admin 账号，密码是 some_pass

```
CREATEUSER'admin'@'%'IDENTIFIEDBY'some_pass';
```

给这个用户授予所有的远程访问的权限。这个用户主要用于管理整个数据库、备份、还原等操作。

```
GRANT ALL  ON *.* TO'admin'@'%';
```

使授权立刻生效

```
flushprivileges;
```

更改数据存放目录

创建数据存放目录

home 目录下建立 data 目录

```
mkdir /home/data
```

把 MySQL 服务进程停掉

```
mysqladmin -u root -p shutdown
```

`/var/lib/mysql` 整个目录移到 `/home/data`, 执行

```
mv /var/lib/mysql /home/data
```

这样就把 MySQL 的数据文件移动到了 `/home/data/mysql` 下

修改 `/etc/my.cnf` 文件,

```
[mysqld]datadir=/home/data/mysqlsocket=/home/data/mysql/mysql.sock[mysql]socket=/home/data/mysql/mysql.sock
```

修改权限

```
chown -R mysql:mysql /home/data/mysql
```

重启后，如果不能启动 MySQL 服务，执行

```
vi /etc/sysconfig/selinux
```

调整

SELINUX=permissive

保存设置，执行 `reboot` 重启生效

查看 MySQL 服务是否开机启动

```
[root@localhost ~]# systemctl is-enabled mysql.service;echo $?enabled0
```

如果是 enabled 则说明是开机自动，如果不是，执行

```
chkconfig --levels 235 mysqld on
```

一般的，为了支持中文，我们应该讲字符集设为 UTF-8， 执行

```
SHOWVARIABLESLIKE 'character%';
```

查看当前 MySQL 字符集

```
mysql> SHOW VARIABLES LIKE 'character%';+-----+
```

----+ Variable_name	Value		+-----+
-----+ character_set_client utf8			
character_set_connection	utf8	character_set_database	latin1
character_set_filesystem	binary	character_set_results	utf8
character_set_server	latin1	character_set_system	utf8
character_sets_dir	/usr/share/mysqlCharsets/	+-----+	

```
-----+8 rows inset (0.00 sec)
```

可以看到默认服务器的字符器是 latin1，对中文不友好。

修改 `/etc/my.cnf` 文件，添加字符集的设置

```
[mysqld]character set server = utf8[mysql]default-character-set = utf8
```

重启 MySQL，可以看到字符集已经修改了

```
mysql> SHOW VARIABLES LIKE 'character%' -> ;+-----+-----+-----+-----+
-----+| Variable_name          | Value                                     |+-----+-----+-----+-----+
--+-----+-----+-----+| character_set_client      | utf8                                     ||
character_set_connection | utf8                                     || character_set_database    | utf8
|| character_set_filesystem | binary                                 || character_set_results    | utf8
|| character_set_server     | utf8                                  || character_set_system     | utf8
|| character_sets_dir       | /usr/share/mysql/charsets/ |+-----+-----+-----+-----+
-----+8 rows inset (0.00 sec)
```

其他常用配置配置

调整 MySQL 运行参数，修改 /etc/my.cnf 文件，常用配置如下：

```
[mysqld]    basedir          = path                # 使用给定目录作为根目录(安装目录)。datadir          = path
# 从给定目录读取数据库文件。pid-file          = filename          # 为mysqld程序指定一个存放进程ID的文件(仅适用于
UNIX/Linux系统);socket = /tmp/mysql.sock        # 为MySQL客户程序与服务器之间的本地通信指定一个套接字文件
(Linux下默认是/var/lib/mysql/mysql.sock文件)port                = 3306          # 指定MySQL侦听的端口
key_buffer      = 384M          # key_buffer是用于索引块的缓冲区大小，增加它可得到更好处理的索引(对所有读和
多重写)。                                索引块是缓冲的并且被所有的线程共享，key_buffer的大小视内存大小而
定。table_cache      = 512          # 为所有线程打开表的数量。增加该值能增加mysqld要求的文件描述符的数量。可
以避免频繁的打开数据表产生的开销sort_buffer_size = 2M          # 每个需要进行排序的线程分配该大小的一个缓冲
区。增加这值加速ORDER BY或GROUP BY操作。                                注意：该参数对应的分配内存是每连
接独占！如果有100个连接，那么实际分配的总共排序缓冲区大小为100×6=600MBread_buffer_size = 2M          # 读
查询操作所能使用的缓冲区大小。和sort_buffer_size一样，该参数对应的分配内存也是每连接独享。query_cache_size
= 32M          # 指定MySQL查询结果缓冲区的大小read_rnd_buffer_size = 8M # 改参数在使用行指针排序之后，
随机读用的。myisam_sort_buffer_size =64M # MyISAM表发生变化时重新排序所需的缓冲thread_concurrency
= 8 # 最大并发线程数，取值为服务器逻辑CPU数量×2，如果CPU支持H.T超线程，再×2thread_cache      = 8
# #缓存可重用的线程数skip-locking          # 避免MySQL的外部锁定，减少出错几率增强稳定性。
[mysqldump]max_allowed_packet      =16M # 服务器和客户端之间最大能发送的可能信息包
[myisamchk]key_buffer      = 256Msort_buffer = 256Mread_buffer = 2Mwrite_buffer = 2M
```

其他可选参数：

back_log = 384

指定MySQL可能的连接数量。当MySQL主线程在很短时间内接收到非常多的连接请求，该参数生效，主线程花费很短时间检查连接并且启动一个新线程。back_log参数的值指出在MySQL暂时停止响应新请求之前的短时间内多少个请求可以被存在堆栈中。如果系统在一个短时间内有很多连接，则需要增大该参数的值，该参数值指定到来的TCP/IP连接的侦听队列的大小。试图设定back_log高于你的操作系统的限制将是无效的。默认值为50。对于Linux系统推荐设置为小于512的整数。

max_connections = n

MySQL服务器同时处理的数据库连接的最大数量(默认设置是100)。超过限制后会报 Too many connections 错误

key_buffer_size = n

用来存放索引区块的RMA值(默认设置是8M)，增加它可得到更好处理的索引(对所有读和多重写)

record_buffer:

每个进行一个顺序扫描的线程为其扫描的每张表分配这个大小的一个缓冲区。如果你做很多顺序扫描，你可能想要增加该值。默认数值是131072(128K)

wait_timeout:

服务器在关闭它之前在一个连接上等待行动的秒数。

interactive_timeout:

服务器在关闭它前在一个交互连接上等待行动的秒数。一个交互的客户被定义为对 mysql_real_connect()使用 CLIENT_INTERACTIVE 选项的客户。默认数值是28800，可以把它改为3600。

skip-name-resolve

禁止MySQL对外部连接进行DNS解析，使用这一选项可以消除MySQL进行DNS解析的时间。但需要注意，如果开启该选项，则所有远程主机连接授权都要使用IP地址方式，否则MySQL将无法正确处理连接请求！

log-slow-queries = slow.log

记录慢查询，然后对慢查询一一优化

skip-innodb

skip-bdb

关闭不需要的表类型，如果你需要，就不要加上这个

备份、还原

方法1:命令行

备份

```
mysqldump --socket=/home/data/mysql/mysql.sock --single-transaction=TRUE -u root -p emsc > emsc.sql
```

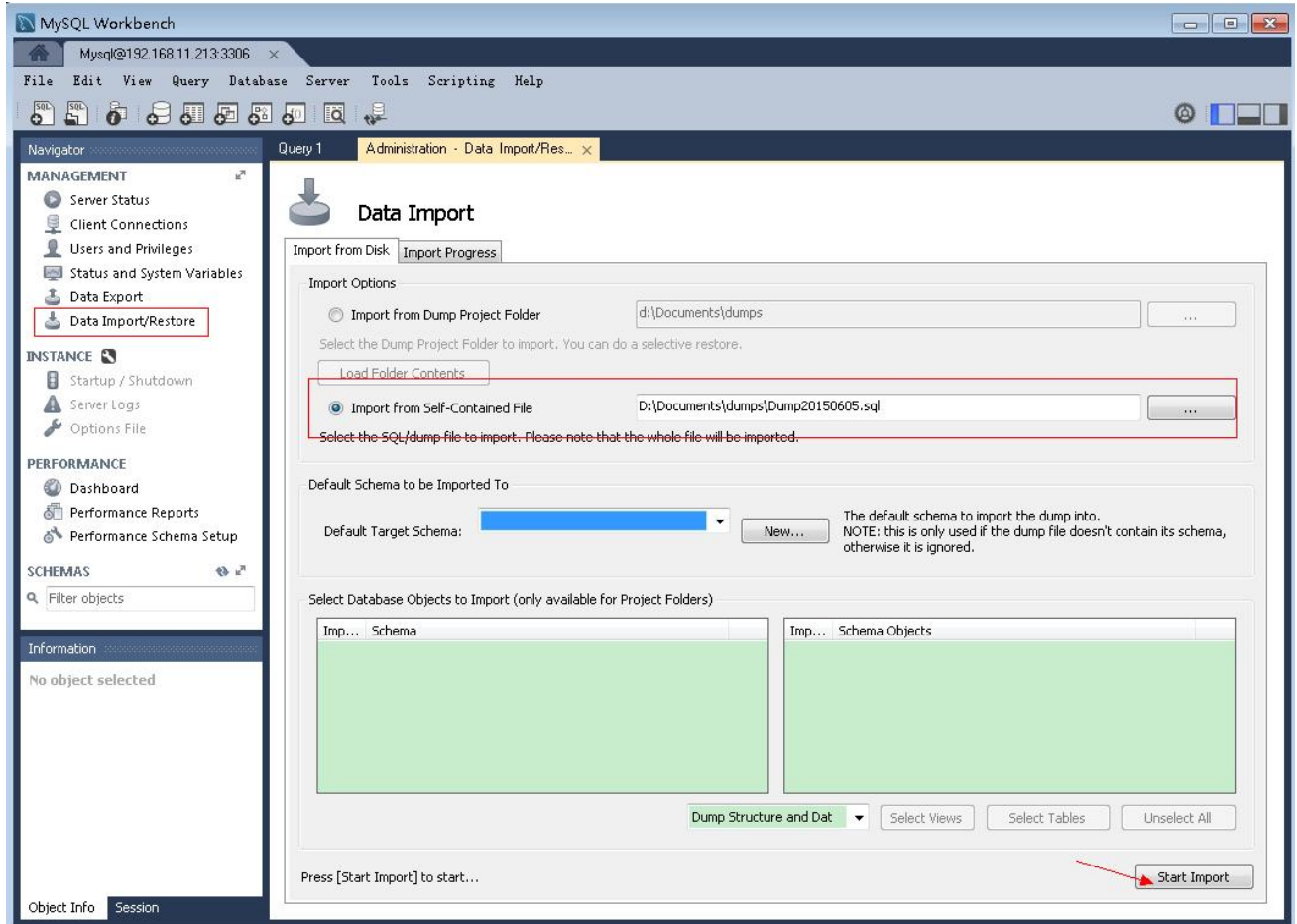
还原

```
mysql --socket=/home/data/mysql/mysql.sock -u root -p emsc < emsc.sql
```

方法2:Workbench

备份

还原



参考

- <http://dev.mysql.com/doc/refman/5.6/en/linux-installation-yum-repo.html>
- <http://dev.mysql.com/doc/refman/5.6/en/mysql-secure-installation.html>
- <http://dev.mysql.com/doc/refman/5.6/en/server-default-configuration-file.html>
- <http://dev.mysql.com/doc/refman/5.6/en/option-files.html>
- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html