## 安装mysql

## centos7安装mysql（yum）

* + [centos7安装mysqlyum](http://blog.csdn.net/whatlookingfor/article/details/52382472#centos7安装mysqlyum)
    - [安装环境](http://blog.csdn.net/whatlookingfor/article/details/52382472#安装环境)
    - [依赖安装](http://blog.csdn.net/whatlookingfor/article/details/52382472#依赖安装)
    - [检查mysql是否已安装](http://blog.csdn.net/whatlookingfor/article/details/52382472#检查mysql是否已安装)
    - [安装](http://blog.csdn.net/whatlookingfor/article/details/52382472#安装)
      * [添加mysql yum respository](http://blog.csdn.net/whatlookingfor/article/details/52382472#添加mysql-yum-respository)
      * [验证是否添加成功](http://blog.csdn.net/whatlookingfor/article/details/52382472#验证是否添加成功)
      * [选择要启用的mysql版本](http://blog.csdn.net/whatlookingfor/article/details/52382472#选择要启用的mysql版本)
      * [通过Yum安装mysql](http://blog.csdn.net/whatlookingfor/article/details/52382472#通过yum安装mysql)
    - [启动和关闭 MySQL Server](http://blog.csdn.net/whatlookingfor/article/details/52382472#启动和关闭-mysql-server)
      * [启动 MySQL Server](http://blog.csdn.net/whatlookingfor/article/details/52382472#启动-mysql-server)
      * [查看 MySQL Server 状态](http://blog.csdn.net/whatlookingfor/article/details/52382472#查看-mysql-server-状态)
      * [关闭 MySQL Server](http://blog.csdn.net/whatlookingfor/article/details/52382472#关闭-mysql-server)
      * [测试是否安装成功](http://blog.csdn.net/whatlookingfor/article/details/52382472#测试是否安装成功)
    - [防火墙设置](http://blog.csdn.net/whatlookingfor/article/details/52382472#防火墙设置)
      * [方式1iptablesCentOS 7x版本之前用法不推荐](http://blog.csdn.net/whatlookingfor/article/details/52382472#方式1iptablescentos-7x版本之前用法不推荐)
    - [方式2firewall-cmd推荐](http://blog.csdn.net/whatlookingfor/article/details/52382472#方式2firewall-cmd推荐)
    - [mysql安全设置](http://blog.csdn.net/whatlookingfor/article/details/52382472#mysql安全设置)
    - [远程访问设置](http://blog.csdn.net/whatlookingfor/article/details/52382472#远程访问设置)
    - [更改数据存放目录](http://blog.csdn.net/whatlookingfor/article/details/52382472#更改数据存放目录)
      * [创建数据存放目录](http://blog.csdn.net/whatlookingfor/article/details/52382472#创建数据存放目录)
      * [把 MySQL 服务进程停掉](http://blog.csdn.net/whatlookingfor/article/details/52382472#把-mysql-服务进程停掉)
      * [移动数据到数据存放目录](http://blog.csdn.net/whatlookingfor/article/details/52382472#移动数据到数据存放目录)
    - [开机自起](http://blog.csdn.net/whatlookingfor/article/details/52382472#开机自起)
    - [设置字符集](http://blog.csdn.net/whatlookingfor/article/details/52382472#设置字符集)
    - [其他常用配置配置](http://blog.csdn.net/whatlookingfor/article/details/52382472#其他常用配置配置)
    - [备份还原](http://blog.csdn.net/whatlookingfor/article/details/52382472#备份还原)

### 安装环境

* centos7
* MySQL 5.6.24

### 依赖安装

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

* 1
* 2

成功安装,提示如下:

[root@bogon /]# yum install libaio

已加载插件：fastestmirror

Loading mirror speeds from cached hostfile

\* 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

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

### 检查mysql是否已安装

yum list installed | grep mysql   
如果已经安装,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。

### 安装

#### 添加mysql yum respository

添加 MySQL Yum Repository 到你的系统 repository 列表中，执行   
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm   
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

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

......

已安装:

mysql-community-release.noarch 0:el7-5

完毕！

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14

提示“完成！”，则说明 源添加成功。

#### 验证是否添加成功

yum repolist enabled | grep "mysql.\*-community.\*"   
可以看到下面内容:

[root@bogon software]# yum repolist enabled | grep "mysql.\*-community.\*"

mysql-connectors-community/x86\_64 MySQL Connectors Community 1

mysql-tools-community/x86\_64 MySQL Tools Community 1

mysql56-community/x86\_64 MySQL 5.6 Community Server 13

* 1
* 2
* 3
* 4

#### 选择要启用的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 mysql

mysql-connectors-community/x86\_64 MySQL Connectors Community 启用: 14

mysql-connectors-community-source MySQL Connectors Community - Sourc 禁用

mysql-tools-community/x86\_64 MySQL Tools Community 启用: 17

mysql-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: 139

mysql56-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

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

可以通过类似下面的语句来启动某些版本

yum-config-manager --disable mysql56-community

yum-config-manager --enable mysql57-community-dmr

* 1
* 2
* 3
* 4

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 Server

baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/7/$basearch/

enabled=1

gpgcheck=1

gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

* 1
* 2
* 3
* 4
* 5
* 6
* 7

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 mysql

mysql-connectors-community/x86\_64 MySQL Connectors Community 14

mysql-tools-community/x86\_64 MySQL Tools Community 17

mysql56-community/x86\_64 MySQL 5.6 Community Server 139

* 1
* 2
* 3
* 4

#### 通过Yum安装mysql

执行命令   
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：fastestmirror

Loading 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

......

......

中途遇到提示,按照提示安装   
执行   
rpm -qi mysql-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 mysql

mysql: /usr/bin/mysql /usr/lib64/mysql /usr/share/mysql /usr/share/man/man1/mysql.1.gz

* 1
* 2

### 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

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

添加

-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

* 1
* 2

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/tcp

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

* 1
* 2

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;   
  
根据提示进行选择。

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   
CREATE USER 'sa'@'%' IDENTIFIED BY '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   
GRANT SELECT,INSERT,UPDATE,DELETE ON \*.\* TO 'sa'@'%';   
创建一个管理员用户 admin 账号 ，密码是 some\_pass   
CREATE USER 'admin'@'%' IDENTIFIED BY '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   
flush privileges;

### 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

如果 MySQL 是启动的，要先关闭   
mysqladmin -u root -p shutdown

#### 移动数据到数据存放目录

/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/mysql

socket=/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

### 开机自起

查看 MySQL 服务是否开机启动

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

enabled

0

如果是 enabled 则说明是开机自动，如果不是，执行   
chkconfig --levels 235 mysqld on

### 设置字符集

一般的，为了支持中文，我们应该讲字符集设为 UTF-8， 执行   
SHOW VARIABLES LIKE '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/mysql/charsets/ |

+--------------------------+----------------------------+

8 rows in set (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

### 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MsSQL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=600MB

read\_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×2

thread\_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 = 256M

sort\_buffer = 256M

read\_buffer = 2M

write\_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：

这里写代码片 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将无法正常处理连接请求！

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.sq

## 重置密码

Login mysql -h localhost -u root -p

## 授权允许用户远程登录

Mysql为了安全性，在默认情况下用户只允许在本地登录，可是在有此情况下，还是需要使用用户进行远程连接，因此为了使其可以远程需要进行如下操作：

一、允许root用户在任何地方进行远程登录，并具有所有库任何操作权限，

具体操作如下：

在本机先使用root用户登录mysql： mysql -u root -p"youpassword" 进行授权操作：

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

重载授权表：

FLUSH PRIVILEGES;

退出mysql数据库：

exit

二、允许root用户在一个特定的IP进行远程登录，并具有所有库任何操作权限，具体操作如下： 在本机先使用root用户登录mysql： mysql -u root -p"youpassword" 进行授权操作： GRANT ALL PRIVILEGES ON \*.\* TO [root@"172.16.16.152](mailto:root@%22172.16.16.152)" IDENTIFIED BY "youpassword" WITH GRANT OPTION; 重载授权表： FLUSH PRIVILEGES; 退出mysql数据库： exit

三、允许root用户在一个特定的IP进行远程登录，并具有所有库特定操作权限，具体操作如下： 在本机先使用root用户登录mysql： mysql -u root -p"youpassword" 进行授权操作： GRANT select，insert，update，delete ON \*.\* TO [root@"172.16.16.152](mailto:root@%22172.16.16.152)" IDENTIFIED BY "youpassword"; 重载授权表： FLUSH PRIVILEGES; 退出mysql数据库： exit

四、删除用户授权，需要使用REVOKE命令，具体命令格式为： REVOKE privileges ON 数据库[.表名] FROM user-name; 具体实例，先在本机登录mysql: mysql -u root -p"youpassword" 进行授权操作： GRANT select，insert，update，delete ON TEST-DB TO [test-user@"172.16.16.152](mailto:test-user@%22172.16.16.152)" IDENTIFIED BY "youpassword"; 再进行删除授权操作： REVOKE all on TEST-DB from test-user; \*\*\*\*注：该操作只是清除了用户对于TEST-DB的相关授权权限，但是这个“test-user”这个用户还是存在。 最后从用户表内清除用户： DELETE FROM user WHERE user="test-user"; 重载授权表： FLUSH PRIVILEGES; 退出mysql数据库： exit

五、MYSQL权限详细分类： 全局管理权限： FILE: 在MySQL服务器上读写文件。 PROCESS: 显示或杀死属于其它用户的服务线程。 RELOAD: 重载访问控制表，刷新日志等。 SHUTDOWN: 关闭MySQL服务。 数据库/数据表/数据列权限： ALTER: 修改已存在的数据表(例如增加/删除列)和索引。 CREATE: 建立新的数据库或数据表。 DELETE: 删除表的记录。 DROP: 删除数据表或数据库。 INDEX: 建立或删除索引。 INSERT: 增加表的记录。 SELECT: 显示/搜索表的记录。 UPDATE: 修改表中已存在的记录。 特别的权限： ALL: 允许做任何事(和root一样)。 USAGE: 只允许登录--其它什么也不允许做。