

任务背景

由于现有业务架构已不能满足当前业务需求，在保证数据完整的前提下，现需要将原有==数据库迁移==到另外一台单独的服务器上，在保证原有服务正常的情况下，将原有LAMP环境中==mysql数据库版本5.6.25升级==为5.6.35。

任务要求

1. 在业务维护时间段对数据库进行==全量备份==
2. 将MySQL数据库从原有服务器迁移到新服务器上并且==升级版本为5.6.35==

课程目标

- 能够通过官方网站获取相应的MySQL软件包及相关手册
- ==掌握MySQL的glibc软件包的安装==
- ==掌握MySQL的源码包的安装==
- 能够对MySQL数据库管理员==root密码进行设置，修改==
- 熟练使用客户端工具mysql连接数据库

涉及知识点

1. 官方下载mysql软件包（根据需求）
2. MySQL数据库的安装部署
3. MySQL数据库管理员用户root密码的设置及修改
4. mysql客户端工具的使用
5. 在任何的场景或者环境下能够安装并且启动（常见的启动报错排查—>思路）

理论储备

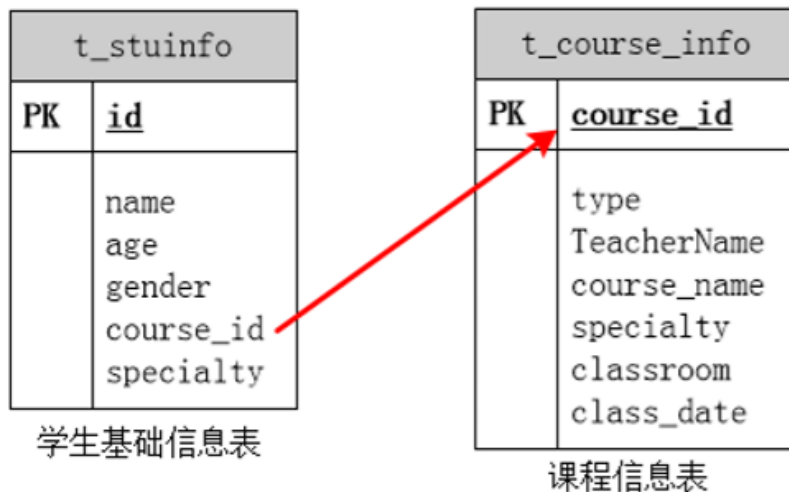
一、MySQL概述

1. 数据库的分类

1.1 关系型数据库

我们常说的关系型数据库，RDBMS（relational database management system），既关系型数据库管理系统；它是指建立在==关系模型==上的数据库系统，关系模型是指用==二维表==的形式表示==实体==和实体间联系的数据模型。比如：

我们需要将学生的信息（姓名、性别、年龄、学号等）保存到数据库中，其中==学生就是一个实体==，==学生的信息就是实体的属性==。这些数据不是直接放到数据库中，而是有组织有纪律的存放在不同的二维表中，并且实体（学生）与实体（学生）之间的关系可以设置。



关系型数据库，是指采用了关系模型来组织数据的数据库。简单来说，关系模型指的就是二维表格模型。

- 常见的关系型数据库

==oracle、mysql==、DB2 (IBM) 、Sybase、==SQL server== (Microsoft微软) 、IBM Informix

特点：容易理解、方便使用、数据一致、读写实时等；

关系型数据库的**最大特点就是==事务的一致性==**：所以也在对事物一致性的维护中有很大的开销。

什么是事务：

事务由一条或者多条sql语句组成，在事务中的操作，这些sql语句要么都成功执行，要么都不执行，这就是一个事务。

==举例说明：==

1. 登陆ATM机，输入密码；
2. 连接数据库，验证密码；
3. 验证成功，获得用户信息，比如存款余额等；
4. 用户输入需要转账的金额，按下确认键；
5. ==从后台数据库中减掉用户账户上的对应金额（update语句）；==
6. ==从后台数据库中给对方账户增加相应的金额（update语句）；==
7. 确认，退卡，走人；

事务特点 (ACID)：

- 原子性(Atomicity):

事务中的全部操作在数据库中是不可分割的，要么==全部完成==，要么==均不执行==。

- 一致性(Consistency):

指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态，不能破坏关系数据的完整性以及业务逻辑上的一致性。

- 隔离性(Isolation):

一个事务的执行不受其他事务的干扰，事务执行的中间结果对其他事务必须是透明的。隔离性是当多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。

- 持久性(Durability):

持久性是指一个事务一旦被提交了, 那么对数据库中的数据的改变就是永久性的, 即便是在数据库系统遇到故障的情况下也不会丢失提交事务的操作。

注意:

==在关系型数据库中, 导致性能欠佳的最主要原因是多表的关联查询, 以及复杂的数据分析类型的复杂SQL报表查询。==

- 应用场景

1、对数据的一致性要求较高; 2、对读写的实时性要求较高; 3、典型的银行及支付系统类;

1.2 非关系型数据库

非关系型数据库, 又被称为NoSQL (Not Only SQL), 意为不仅仅是SQL (Structured QueryLanguage, 结构化查询语言)

- 常见的非关系型数据库

MangoDB、Redis、Membase等

- 应用场景

1、对数据库性能要求较高; 2、需要灵活性更强的IT系统; 3、不需要高度的数据一致性;

2. MySQL数据库介绍

MySQL是由MySQL创始人和主要开发人在瑞典创办的一个mysql AB 公司。 2008年1月16号 MySQL被Sun公司收购。 2009年04月20日Oracle收购Sun 公司, MySQL 转入Oracle 门下。 Oracle 对MySQL版本重新进行了划分, 分成了社区版和企业版, 企业版是需要收费的, 当然收费的就会提供更多的功能。

社区版: MySQL Community Edition (==GPL==)

- 1、可以看做是企业版的“广泛体验版 (小白鼠版)”, 未经各个专有系统平台的压力测试和性能测试
- 2、基于GPL协议发布, 可以随意下载使用
- 3、没有任何官方技术支持服务

企业版: MySQL Enterprise Edition(==commercial==)

- 1、提供了比较全面的高级功能、管理工具及技术支持
- 2、安全性、稳定性、可扩展性比较好

集群版: MySQL Cluster CGE(==commercial==)

扩充: 关于版本的命名方式如何定义?

版本	说明
α (Alpha) 版	内测版，内部交流或者专业测试人员测试用。Bug较多，普通用户最好不要安装。
β (Beta) 版	公测版，专业爱好者大规模测试用，存在一些缺陷，该版本也不适合一般用户安装。
γ (Gamma) 版	相当成熟的测试版，与即将发行的正式版相差无几。
==Final==	正式版本
Free	自由版本
==Release==	发行版本
==Standard==	标准版本
==Mini==	迷你精简版本,只有最基本的功能
Upgrade	升级版本
==GA(GenerallyAvailable)==	开发团队认为该版本是稳定版,可以在较为关键的场合使用。
Retail	零售版

3. MySQL的获取

官网：www.mysql.com

4. MySQL的安装方式

4.1 二进制方式安装

- RPM版本

命名：MySQL-server-5.6.31-1.el7.x86_64.rpm，需要在特定linux版本下安装。

- ==基于glibc版本==

命名：

MySQL-server-5.6.31-1.linux_glibc2.5.x86_64.rpm

==mysql-5.6.30-linux-glibc2.5-x86_64.tar.gz==

依赖¹库，可以安装在通用的Linux系统下

优点：

安装和维护都比较方便，不需要编译。

缺点：

可定制性差，可移植性差，一个系统上只能安装一个mysql，不灵活。（rpm包方式）

glibc方式安装比rpm灵活：安装路径可以选择，事先已经编译好，所以可定制性相对差（相对源码包）

4.2 源代码编译安装

命名: mysql-5.6.31.tar.gz, 通用的Linux下都可以编译安装。

优点:

可定制性强 (安装可以根据用户的需求, 只安装所需要的功能)

缺点:

安装复杂, 所需要的时间比二进制的安装要长得多

二、MySQL数据库的安装

1. rpm包安装

1.1 安装须知

- MySQL官方软件包介绍

通过MySQL官网下载相应的软件rpm包, 此次以5.6.25为例:

MySQL-5.6.25-1.el7.x86_64.rpm-bundle.tar

解压后得到以下软件包:

MySQL-server-5.6.25-1.el7.x86_64.rpm	服务端
MySQL-client-5.6.25-1.el7.x86_64.rpm	客户端工具
MySQL-devel-5.6.25-1.el7.x86_64.rpm	开发工具包 (库文件)
MySQL-shared-5.6.25-1.el7.x86_64.rpm	共享组件
MySQL-embedded-5.6.25-1.el7.x86_64.rpm	嵌入式工具包
MySQL-shared-compat-5.6.25-1.el7.x86_64.rpm	兼容库
MySQL-test-5.6.25-1.el7.x86_64.rpm	测试套件

- 默认安装目录

```
/usr/sbin/mysqld
```

- 默认数据目录

```
/var/lib/mysql
```

- 默认端口号

端口: 3306

- 默认socket文件存放路径

```
/var/lib/mysql/mysql.sock
```

1.2 安装MySQL数据库

我的环境:

MySQL相关软件存放路径:

/soft/installer

mysql安装软件包相关目录

rpm包安装MySQL步骤:

1. 解压到指定路径

```
# tar -xf MySQL-5.6.25-1.el7.x86_64.rpm-bundle.tar -C /mysql25_rpm
```

2. 安装MySQL数据库

```
# rpm -ivh MySQL-server-5.6.25-1.el7.x86_64.rpm MySQL-client-5.6.25-1.el7.x86_64.rpm
```

注意:

```
Preparing... ##### [100%]  
file /usr/share/mysql/charsets/README from install of MySQL-server-5.6.25-1.el7.x86_64  
conflicts with file from package mariadb-libs-1:5.5.52-1.el7.x86_64
```

错误原因:

centos7默认安装的mariadb-libs软件包和现在安装的MySQL-server冲突导致。

解决:

卸载mariadb-libs软件包后再次安装。

3. 安装完毕后查看输出信息

...

A RANDOM PASSWORD HAS BEEN SET FOR THE MySQL root USER !
You will find that password in '/root/.mysql_secret'.

You must change that password on your first connect,
no other statement but 'SET PASSWORD' will be accepted.
See the manual for the semantics of the 'password expired' flag.

Also, the account for the anonymous user has been removed.

In addition, you can run:

```
/usr/bin/mysql_secure_installation
```

which will also give you the option of removing the test database.
This is strongly recommended for production servers.

See the manual for more instructions.

Please report any problems at <http://bugs.mysql.com/>

The latest information about MySQL is available on the web at

<http://www.mysql.com>

Support MySQL by buying support/licenses at <http://shop.mysql.com>

New default config file was created as /usr/my.cnf and
will be used by default by the server when you start it.

You may edit this file to change server settings

4. 后续配置

注意：在初始化数据库之前，必须先启动mysql数据库

```
[root@MissHou ~]# service mysql start           //启动数据库
Starting MySQL.. SUCCESS!
[root@MissHou ~]# netstat -nlt|grep mysql        //查看3306端口是否监听
tcp6      0      0 :::3306          :::*             LISTEN        4248/mysqld
[root@MissHou ~]# /usr/bin/mysql_secure_installation //数据库安全配置
...
```

5. 测试登录

```
[root@MissHou ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.6.25 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 its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

1.3 MySQL客户端工具

客户端工具1: mysql

mysql: the MySQL command-line tool //mysql命令行工具，一般用来连接访问mysql数据库

```
[root@MissHou ~]# which mysql
/usr/bin/mysql
[root@MissHou ~]# rpm -qf /usr/bin/mysql
MySQL-client-5.6.25-1.el7.x86_64
```

用法:

mysql [options] [db_name]

参数:

- u, --user=name: 指定登录用户名
- p, --password=指定登录密码(注意是小写p), 一定要放到最后面
- h, --host=name: 指定数据库的主机地址
- P, --port=xxx: 指定数据库的端口号(大写P)
- S, --socket=name: 指定socket文件
- e, --execute=name: 使用非交互式操作(在shell终端执行sql语句)

举例:

```
[root@MissHou ~]# mysql -uroot -e 'show databases;' -p
```

Enter password: 这里密码不显示

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema|
+-----+
```

默认数据库说明:

information_schema库:

1. 对象信息数据库, 提供对数据库元数据的访问, 有关MySQL服务器的信息, 例如数据库或表的名称, 列的数据类型或访问权限等。
2. 在INFORMATION_SCHEMA中, 有数个只读表。它们实际上是视图, 而不是基本表, 因此你将无法看到与之相关的任何文件。

视图: 是一个虚表, 即视图所对应的数据不进行实际存储, 数据库中只存储视图的定义, 在对视图的数据进行操作时, 系统根据视图的定义去操作与视图相关联的基本表。

mysql库:

1. 该mysql数据库是系统数据库。它包含存储MySQL服务器运行时所需的信息的表。比如权限表、对象信息表、日志系统表、时区系统表、优化器系统表、杂项系统表等。
2. 不可以删除, 也不要轻易修改这个数据库里面的表。

performance_schema库:

1. MySQL5.5开始新增一个数据库, 主要用于收集数据库服务器性能; 并且库里表的存储引擎均为PERFORMANCE_SCHEMA, 而用户是不能创建存储引擎为PERFORMANCE_SCHEMA的表。
2. MySQL5.5默认是关闭的, 需要手动开启, 在配置文件里添加:

```
[mysqld]
performance_schema=ON
```

从MySQL5.6开始, 默认打开

```
mysql> show variables like '%performance%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| performance_schema | ON   |
...

```

客户端工具2:

```
mysqladmin: client for administering a MySQL server //mysql数据库的客户端管理工具
[root@MissHou ~]# which mysqladmin
/usr/bin/mysqladmin
[root@MissHou ~]# rpm -qf /usr/bin/mysqladmin
MySQL-client-5.6.25-1.el7.x86_64
```

用法:

```
mysqladmin [options] command [command-options] [command [command-options]] ...
[root@MissHou ~]# mysqladmin --help 获取帮助
```

常用参数:

```
-h, --host=name 连接主机
-p, --password 密码
-P, --port=# 连接端口
```


-S, --socket=name 指定socket文件
-u, --user=name 指定用户名

常用命令:

password [new-password]	更改密码
reload	Reload grant tables 刷新授权表
shutdown	Take server down 停止mysql服务
status	Gives a short status message from the server 简短查看数据库状态信息
start-slave	Start slave 启动slave
stop-slave	Stop slave 停止slave
variables	Prints variables available 打印可用变量
version	Get version info from server 查看当前mysql数据库的版本信息

举例:

```
[root@MissHou ~]# mysqladmin shutdown -p //关闭数据库
Enter password:

[root@MissHou ~]# mysqladmin password 'newpass' -p //默认修改数据库管理员root密码
Enter password:

[root@MissHou ~]# mysqladmin password 123 -uroot -h localhost -p //也可以指定用户和主机
Enter password:

[root@MissHou ~]# mysqladmin status -p //简短查看数据库的状态信息
Enter password:
Uptime: 788 Threads: 1 Questions: 18 Slow queries: 0 Opens: 67 Flush tables: 1 Open
tables: 60 Queries per second avg: 0.022

[root@MissHou ~]# mysqladmin proc status -p
Enter password:
+---+-----+-----+-----+-----+-----+-----+-----+
| Id | User | Host      | db | Command | Time | State | Info          |
+---+-----+-----+-----+-----+-----+-----+-----+
| 8  | root | localhost |    | Query   | 0    | init  | show processlist |
+---+-----+-----+-----+-----+-----+-----+
Uptime: 799 Threads: 1 Questions: 21 Slow queries: 0 Opens: 67 Flush tables: 1 Open
tables: 60 Queries per second avg: 0.026

[root@MissHou ~]# mysqladmin proc extended-status -p //查看扩展信息, 比如变量
[root@MissHou ~]# mysqladmin create db01 -p //创建数据库
[root@MissHou ~]# mysqladmin drop db01 -p //删除数据库
```

2. glibc包安装

2.1 安装须知

- 软件包介绍

```
mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz //通用linux下的二进制包，已经编译好
解压后的文件：
[root@MissHou mysql-5.6.35-linux-glibc2.5-x86_64]# ls
bin  COPYING  data  docs  include  lib  man  mysql-test  README  scripts  share  sql-bench
support-files
```

mysql_glibc1

- 默认相关选项

默认安装路径：/usr/local/mysql
默认数据目录：/usr/local/mysql/data
默认端口：3306
默认socket文件存放路径：/tmp/mysql.sock

2.2 安装MySQL数据库

参考官当：[MySQL-glibc安装手册](#)

需求：

```
install_dir:/data/mysql35
data_dir:/data/mysql35/data
port:3307
socket:$install_dir/mysql35.sock

1. 创建相应的数据目录和用户
[root@mysql01 installer]# mkdir /data/mysql35 -p
[root@mysql01 installer]# id mysql
id: mysql: no such user
[root@mysql01 installer]# useradd -r -s /bin/false mysql
[root@mysql01 installer]# id mysql
uid=997(mysql) gid=995(mysql) groups=995(mysql)

2. 拷贝相应的数据文件到指定的安装目录里
1) 下载glibc的软件包
2) 解压glibc的软件包
[root@mysql01 installer]# ls mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz
mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz
[root@mysql01 installer]# tar xf mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz
3) 进入到解压目录里拷贝数据文件到安装目录
[root@mysql01 installer]# cd mysql-5.6.35-linux-glibc2.5-x86_64/
[root@mysql01 mysql-5.6.35-linux-glibc2.5-x86_64]# ls
bin  COPYING  data  docs  include  lib  man  mysql-test  README  scripts  share  sql-bench
support-files
[root@mysql01 mysql-5.6.35-linux-glibc2.5-x86_64]# cp -a ./ * /data/mysql35/
```

说明：将数据文件拷贝到mysql的安装路径后，说明mysql数据库已经安装完成！！

3. 手动的初始化数据库（重点）

1) 进入到mysql的安装目录里

```
# cd /data/mysql35/
```

2) 更改mysql安装目录的权限

```
# chown -R mysql. /data/mysql35/
```

```
# ll -d /data/mysql35/
```

```
drwxr-xr-x. 13 mysql mysql 191 Dec  4 12:18 /data/mysql35/
```

3) 初始化数据库

清空环境：

确保当前系统没有/etc/my.cnf文件（my.cnf 文件里有默认安装路径）

```
# rm -f /etc/my.cnf
```

```
# scripts/mysql_install_db --user=mysql --basedir=/data/mysql35 --datadir=/data/mysql35/data
```

初始化完毕后，检查数据目录里有没有文件：

```
[root@mysql01 mysql35]# ll data/
```

```
total 110600
```

```
-rw-rw----. 1 mysql mysql 12582912 Dec  4 12:24 ibdata1
```

```
-rw-rw----. 1 mysql mysql 50331648 Dec  4 12:24 ib_logfile0
```

```
-rw-rw----. 1 mysql mysql 50331648 Dec  4 12:24 ib_logfile1
```

```
drwx-----. 2 mysql mysql    4096 Dec  4 12:24 mysql
```

```
drwx-----. 2 mysql mysql    4096 Dec  4 12:24 performance_schema
```

```
drwxr-xr-x. 2 mysql mysql     20 Dec  4 12:15 test
```

4. 启动数据库

使用service方式启动mysql数据库：

```
[root@mysql01 mysql35]# cp support-files/mysql.server /etc/init.d/mysql35
```

```
[root@mysql01 mysql35]# ll /etc/init.d/mysql35
```

```
-rwxr-xr-x. 1 root root 10875 Jul 31 14:51 /etc/init.d/mysql35
```

```
[root@mysql01 mysql35]# service mysql35 start
```

```
Unit mysql35.service could not be found.
```

```
/etc/init.d/mysql35: line 276: cd: /usr/local/mysql: No such file or directory
```

```
Starting MySQL ERROR! Couldn't find MySQL server (/usr/local/mysql/bin/mysqld_safe)
```

```
[root@mysql01 mysql35]# vim /etc/init.d/mysql35
```

以上错误原因是因为默认的安装路径为:/usr/local/mysql;但是实际上我是/data/mysql35;所以需要修改脚本文件里的mysql的安装路径。

```
[root@mysql01 mysql35]# service mysql35 start
```

```
Unit mysql35.service could not be found.
```

```
Starting MySQL.Logging to '/data/mysql35/data/mysql01.itcast.cc.err'.
```

```
180731 14:59:26 mysqld_safe The file /usr/local/mysql/bin/mysqld
does not exist or is not executable. Please cd to the mysql installation
directory and restart this script from there as follows:
```

```
./bin/mysqld_safe&
```

See <http://dev.mysql.com/doc/mysql/en/mysqld-safe.html> for more information
ERROR! The server quit without updating PID file (/data/mysql35/data/mysql01.itcast.cc.pid).

解决: service mysql35启动的方式会调用mysqld_safe脚本去启动mysqld程序, 如果启动不成功则需要修改、/data/mysql35/bin/mysqld_safe文件里的内容, 将/usr/local/mysql修改成/data/mysql35

5. 测试验证

```
[root@mysql01 bin]# service mysql35 start
Unit mysql35.service could not be found.
Starting MySQL. SUCCESS!
[root@mysql01 bin]# ps -ef|grep mysql
root      6755      1  0 15:40 pts/0    00:00:00 /bin/sh /data/mysql35/bin/mysqld_safe --
datadir=/data/mysql35/data --pid-file=/data/mysql35/data/mysql01.itcast.cc.pid
mysql     6852    6755  3 15:40 pts/0    00:00:00 /data/mysql35/bin/mysqld --
basedir=/data/mysql35 --datadir=/data/mysql35/data --plugin-dir=/data/mysql35/lib/plugin --
user=mysql --log-error=/data/mysql35/data/mysql01.itcast.cc.err --pid-
file=/data/mysql35/data/mysql01.itcast.cc.pid
```

启动错误:

```
2018-07-31 16:27:40 7385 [Note] Server socket created on IP: '::'.
2018-07-31 16:27:40 7385 [ERROR] Can't start server: Bind on TCP/IP port: Address already in use
2018-07-31 16:27:40 7385 [ERROR] Do you already have another mysqld server running on port: 3306
?
2018-07-31 16:27:40 7385 [ERROR] Aborting
```

原因: 端口号占用, 冲突

解决: 不同的mysql数据库实例来指定不同的端口号

详细步骤参考:

说明: 由于之前已经安装了rpm包的mysql数据库, 所以mysql用户和mysql组已经存在, 故不需再创建。

1. 根据当前环境, 解压安装包到指定路径

```
[root@db01 installer]# pwd
/soft/installer
[root@db01 installer]# mkdir mysql35
[root@MissHou mysql]# tar -xf mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz -C mysql35
[root@MissHou mysql]# cd mysql35
[root@MissHou mysql35]# ls
mysql-5.6.35-linux-glibc2.5-x86_64
```

2. 将解压后的目录里所有文件拷贝到安装路径(默认/usr/local/mysql)

如果没有该目录需要事先创建:

```
[root@MissHou mysql35]# mkdir /usr/local/mysql
将解压目录里的所有文件拷贝到/usr/local/mysql目录中:
[root@MissHou mysql35]# cp -a mysql-5.6.35-linux-glibc2.5-x86_64/* /usr/local/mysql/
```

3. 修改相应目录的权限

```
[root@MissHou mysql35]# chown -R mysql. /usr/local/mysql/
```

4. 初始化数据库

```
[root@MissHou mysql]# pwd
```

```
/usr/local/mysql
```

```
[root@MissHou mysql]# scripts/mysql_install_db --user=mysql
```

验证:

```
[root@MissHou mysql]# ls data/
```

```
ibdata1  ib_logfile0  ib_logfile1  mysql  performance_schema  test
```

注意:

MySQL 5.6.8, mysql_install_db creates a default option file named my.cnf in the base installation directory.

5. 启动数据库

注意: 由于之前已经安装了rpm版本的mysql数据库, 端口默认都是3306, 所以先停止之前的mysql数据库

```
[root@MissHou mysql]# bin/mysqld_safe --user=mysql & 放到后台启动
```

6. 查看mysql是否启动成功

```
[root@MissHou mysql]# netstat -nlp|grep mysqld
```

```
tcp6      0      0 :::3306          :::*              LISTEN
4987/./bin/mysqld
unix  2      [ ACC ]      STREAM      LISTENING      73291      4987/./bin/mysqld      /tmp/mysql.sock
```

7. 拷贝启动脚本到/etc/init.d/下 [可选项]

```
[root@MissHou mysql]# cp support-files/mysql.server /etc/init.d/mysql35
```

8. 使用service命令来管理mysql服务

```
[root@MissHou mysql]# service mysql35 stop
```

```
[root@MissHou mysql]# service mysql35 start
```

扩展:

如何将mysql服务交给systemd来管理?

```
[Unit]
```

```
Description=MySQL Community Server
```

```
After=network.target
```

```
After=syslog.target
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
Alias=mysql.service
```

```
[Service]
```

```
User=mysql
```

```
Group=mysql
```

```
# Execute pre and post scripts as root
```

```
PermissionsStartOnly=true
```

```
# Needed to create system tables etc.
```

```
#ExecStartPre=/usr/bin/mysql-systemd-start pre
```

```
# Start main service
```

```
ExecStart=/usr/local/mysql/bin/mysqld_safe
```

```
# Don't signal startup success before a ping works
```

```
#ExecStartPost=/usr/bin/mysql-systemd-start post
```

```
# Give up if ping don't get an answer
TimeoutSec=600

Restart=always
PrivateTmp=false
```

2.3 启动多实例

- 问题描述:

二进制的rpm包和glibc包安装mysql后, 默认端口都是3306, 如果在同一台主机上使用以上2种方式安装多个mysql实例后, 默认情况下只能启动一个mysql实例。

- 单机如何启动多实例?

- 了解MySQL的启动过程

1. 命令启动数据库:

service mysql **start** ——>mysqld_safe——>mysqld daemon (执行mysqld二进制命令来启动服务)
或者

mysqld_safe [options] & ——>mysqld daemon (执行mysqld二进制命令来启动服务)

2. 启动数据库读取相应配置文件顺序:

注意: 启动方式不一样, 读取顺序不一样!!!

\$basedir(mysql安装目录) ——> **\$datadir** (mysql的数据目录) ——>/etc/my.cnf ——> /etc/mysql/my.cnf
——> ~/.my.cnf

3. 读取相应的数据文件和其他物理文件 (日志文件等)

4. 生成socket文件和pid文件并在指定的端口进行监听(可以自定义)

- ==需要注意的地方==

1. mysqld和mysqld_safe启动mysql数据库有什么区别?

- a. 一般情况下, 很少或者几乎不用mysqld命令直接启动mysql数据库, 虽然它可以启动。
- b. mysqld_safe会在启动MySQL服务器后继续监控其运行情况, 并在其死机时重新启动它。
- c. mysqld_safe添加了一些安全特性, 比如当发生错误时重新启动服务器, 并将运行时信息记录到错误日志文件中。
- d. 在Unix系统中推荐使用mysqld_safe启动mysql数据库。

2. 多个my.cnf文件以哪个为准(mysqld)?

- a. 正常情况下, 一个数据库实例对应一个my.cnf文件
- b. 如果同时有不同多个my.cnf文件, 那么mysqld匹配顺序一般为:

\$basedir ——> **\$datadir** ——>/etc/my.cnf——>/etc/mysql/my.cnf——>SYSCONFDIR/my.cnf——>
~/.my.cnf, 如果匹配到, 不会往下继续读。

注意: 安装方式不一样一定需要注意配置文件的存放路径; 尽可能保证一个数据库实例只有一个my.cnf文件。

3. socket文件和pid文件什么时候产生, 什么时候消失?

socket文件和pid文件是在mysql数据库启动后产生, 停止后消失。

- 单机启动多实例的思路

==编写不同的my.cnf文件来指定不同的端口、socket文件! ==

- 配置文件的编写

修改rpm包的mysql数据库的配置文件，指定端口为3307就不会和glibc的默认3306端口冲突：

```
vim /usr/my.cnf
basedir = /usr
datadir = /var/lib/mysql
port = 3307
socket = /var/lib/mysql/mysql.sock
```

=====

```
[mysqld]                //表示启动时给mysqld程序传递相应参数
```

```
basedir=xxx
datadir=xxx
port=xxx
socket=xxx
server_id=xxx
log-error=xxx
pid-file=xxx
```

```
[client]                //表示使用mysql客户端工具连接mysql数据库时传递相应参数
```

```
socket=xxx
port=xxx
password="xxx"
```

```
[mysqld_safe]           //mysqld_safe启动数据库时传递相应参数
```

```
...
```

```
[可以自定义标签]
```

```
....
```

+++++

```
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
```

说明：

mysql通过sql_mode来控制mysql数据库的行为。

NO_ENGINE_SUBSTITUTION：

当像CREATE TABLE或ALTER TABLE这样的语句指定一个被禁用或未编译的存储引擎时，控制自动替换默认存储引擎。

- 1) 如果当前mysql配置了sql_mode=NO_ENGINE_SUBSTITUTION，那么当指定一个被禁用或未编译的存储引擎时直接替换成默认存储引擎InnoDB；
- 2) 如果当前mysql没有配置sql_mode=NO_ENGINE_SUBSTITUTION，那么当发生上面操作时直接报错。

STRICT_TRANS_TABLES表示如果不能将给定的值插入到事务表中，则中止该语句，对于非事务表，如果值出现在单行语句或多行语句的第一行，则中止该语句。

2.4 总结

二进制rpm包和glibc包方式安装的优缺点：

安装方式	优点	缺点
rpm	安装卸载简单	可定制性差
glibc	可定制性相比rpm包灵活些	安装相比rpm包复杂些，需要手动初始化数据库

3. 源码安装

参考官当：[MySQL-Source安装配置手册](#)

源码安装三步曲：配置——>编译——>安装

3.1 安装须知

- 软件包介绍

mysql-5.6.31.tar.gz

//可在任意发行版的Linux下安装

- 默认相关选项

默认安装路径：/usr/local/mysql
默认数据目录：/usr/local/mysql/data
默认端口：3306
默认socket文件存放路径：/tmp/mysql.sock

3.2 安装需求

安装基目录basedir：/mydb/mysql31
数据存放目录datadir：/mydb/mysql31/data
端口号：3308
socket文件存放目录：\$basedir/mysql31.sock
根据需求开启相应功能

3.2 了解配置选项

- 常用配置选项

配置选项	描述	默认值	建议值
CMAKE_INSTALL_PREFIX	安装基目录(basedir)	/usr/local/mysql	根据需求
MYSQL_DATADIR	数据目录(datadir)		根据需求
SYSCONFDIR	默认配置文件my.cnf路径		/etc
MYSQL_TCP_PORT	TCP/IP端口	3306	非默认端口
MYSQL_UNIX_ADDR	套接字socket文件路径	/tmp/mysql.sock	\$basedir/
DEFAULT_CHARSET	默认字符集	latin1	utf8mb4
DEFAULT_COLLATION	默认校验规则	latin1_swedish_ci	utf8mb4_general_ci
WITH_EXTRA_CHARSETS	扩展字符集	all	all
ENABLED_LOCAL_INFILE	是否启用本地加载外部数据文件功能	OFF	建议开启

- 存储引擎相关配置项

说明：

以下选项值均为布尔值，0或1,0代表不编译到服务器中，1代表编译，建议都静态编译到服务器中。其他的存储引擎可以根据实际需求在安装时通过WITH_xxxx_STORAGE_ENGINE=1的方式编译到服务器中。

WITH_INNOBASE_STORAGE_ENGINE

将InnoDB存储引擎插件构建为静态模块编译到服务器中建议编译到服务器中。WITH_PARTITION_STORAGE_ENGINE是否支持分区

WITH_FEDERATED_STORAGE_ENGINE

本地数据库是否可以访问远程mysql数据

WITH_BLACKHOLE_STORAGE_ENGINE

黑洞存储引擎，接收数据，但不存储，直接丢弃

WITH_MYISAM_STORAGE_ENGINE

将MYISAM存储引擎静态编译到服务器中

3.2 安装步骤

- 下载软件包并解压到临时目录

```
[root@misshou mysql]# pwd
/soft/mysql
[root@misshou mysql]# tar -xf mysql-5.6.31.tar.gz -C mysql31
```

- 根据需求进行配置

```
[root@misshou mysql-5.6.31]# cat cmake.sh
cmake . \
```

```
-DCMAKE_INSTALL_PREFIX=/mydb/mysql31 \  
-DMYSQL_DATADIR=/mydb/mysql31/data \  
-DSYSCONFDIR=/etc \  
-DMYSQL_TCP_PORT=3308 \  
-DMYSQL_UNIX_ADDR=/mydb/mysql31/mysql31.sock \  
-DWITH_INNOBASE_STORAGE_ENGINE=1 \  
-DWITH_PARTITION_STORAGE_ENGINE=1 \  
-DWITH_FEDERATED_STORAGE_ENGINE=1 \  
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \  
-DWITH_MYISAM_STORAGE_ENGINE=1 \  
-DENABLED_LOCAL_INFILE=1 \  
-DEXTRA_CHARSETS=all \  
-DDEFAULT_CHARSET=utf8mb4 \  
-DDEFAULT_COLLATION=utf8mb4_general_ci
```

- 编译

```
make
```

- 安装

```
make install
```

3.2 后续配置

- 初始化数据库

注意：进入到安装目录里/mydb/mysql31

1. 修改目录权限

```
[root@misshou mysql31]# chown mysql. -R /mydb/mysql31/
```

2. 初始化数据库

```
[root@misshou mysql31]# scripts/mysql_install_db --user=mysql
```

3. 查看数据目录里是否有数据文件

```
[root@misshou mysql31]# ls data/
```

```
ibdata1  ib_logfile0  ib_logfile1  mysql  performance_schema  test
```

- 拷贝启动脚本到/etc/init.d/目录

```
[root@misshou mysql31]# cp support-files/mysql.server /etc/init.d/mysql31
```

- 启动数据库

```
[root@misshou mysql31]# service mysql31 start
```

- 安全配置

```
[root@misshou mysql31]# bin/mysql_secure_installation --user=mysql
```

- 环境变量配置

1. 访问不同实例的数据库

1) 访问glibc的mysql

```
# which mysql
/data/mysql35/bin/mysql
# mysql -uroot -p
```

2) 访问源码的mysql

```
# mysql -S /mydb/mysql31/mysql31.sock -uroot
注意：要指定sock文件路径
```

2. 通过别名直接访问

临时：

```
alias mysql35='/data/mysql35/bin/mysql'
alias mysql31='/mydb/mysql31/bin/mysql'
```

永久：

```
# vim /etc/bashrc
alias mysql35='/data/mysql35/bin/mysql'
alias mysql31='/mydb/mysql31/bin/mysql'
```

```
# source /etc/bashrc
```

直接访问：

```
# mysql35 -uroot -p123
# mysql31
```

- mysql客户端程序连接mysql方式

```
vim /etc/my.cnf
[client]
socket = /xxx
user = root
password = "123"
```

三、设置管理员root密码

1. 设置密码或修改新密码

1. 使用mysqladmin管理工具修改

```
[root@misshou ~]# mysqladmin -uroot password 'pass' //没有密码设置新密码
[root@misshou ~]# mysqladmin -uroot password 'newpass' -p //知道密码，重新修改密码
Enter password:输入老密码
```

2. 进入mysql数据库通过修改权限表来更新密码

```
mysql> update mysql.user set password=password('123') where user='root' and host='localhost';
mysql> flush privileges; //需要刷新权限表
或者
mysql> set password for 'root'@'localhost'=password('111'); //此语句不需要刷新权限表
```

3. 使用安全初始化脚本命令修改

```
[root@misshou ~]# mysql_secure_installation
```

2. 忘记密码重置密码

思路：跳过授权表，直接登录数据库，然后通过sql语句修改密码

方法1：

1. 停止数据库

```
[root@misshou ~]# service mysql stop
```

2. 启动数据库时传递--skip-grant-tables参数

```
[root@misshou ~]# /usr/local/mysql/bin/mysqld_safe --skip-grant-tables --user=mysql &
```

3. 免密码登录数据库修改密码

```
[root@misshou ~]# /usr/local/mysql/bin/mysql
```

```
mysql> update mysql.user set password=password('456') where user='root' and host='localhost';
mysql> flush privileges;
```

或者

```
mysql> set password for root@localhost=password('444');
```

不被允许的

```
mysql> set password for 'root'@'localhost'=password('123');
```

```
ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option so it cannot execute this statement
```

方法2：

1. 修改my.cnf配置文件

```
[mysqld]
```

```
....
```

```
skip-grant-tables
```

2. 重启数据库后免密码登录修改密码

略

3. MySQL5.7版本区别

1. 密码长度和复杂度有要求，如下：

MySQL's validate_password plugin is installed by default. This will require that passwords contain at least one upper case letter, one lower case letter, one digit, and one special character, and that the total password length is at least 8 characters.

2. 修改密码的语句有变化

5.6版本：

```
mysql> update mysql.user set password=password('456') where user='root' and host='localhost';
```

```
mysql> flush privileges;
```

或者

```
mysql> set password for root@localhost=password('444');
```

5.7版本:

1) mysql库的user表中已经没有password这个字段, 所以不能使用update命令更新。

2) mysql> set password for root@localhost=password('444');该语句虽然可以但是在更新版本中会被丢弃, 变成如下:

```
mysql> set password for 'root'@'localhost'='NiHaoshijie@';
```

3) 推荐使用alter user语句, 如下:

```
ALTER USER user IDENTIFIED BY 'auth_string';
```

password5.7

四、扩展补充

1. 字符集与字符编码

==字符: ==字符是各种**文字和符号**的总称, 包括各个国家文字、标点符号、图形符号、数字等。

==字符集: ==字符集是多个**字符的集合**, 字符集种类较多, 每个字符集包含的字符个数不同, 计算机要准确的处理各种字符集文字, 需要进行字符编码, 以便计算机能够识别和存储各种文字。也就是说**字符编码是字符集的实现方式**。

需要注意的是: 有的字符编码和字符集的名称是一致的。

- 常见的字符集:

Unicode:

也叫统一字符集, 它包含了几乎世界上所有的已经发现且需要使用的字符如: 中文、日文、英文、德文等。

ASCII: 早期的计算机系统只能处理英文, 所以ASCII也就成为了计算机的缺省字符集, 包含了英文所需要的所有字符。

GB2312: 中文字符集, 包含ASCII字符集。

GBK: 是GB2312的扩展, 但完整包含了GB2312的所有内容。

GB18030: 是GBK字符集的超集, 常叫大汉字符集, 也叫CJK (Chinese, Japanese, Korea) 字符集, 包含了中、日、韩三国语言中的所有字符。

- 常见的字符编码:

ASCII: 在 ASCII 编码中, 一个英文字母 (不分大小写) 占一个字节的空間, 一个中文汉字占两个字节的空間。

UTF-8(Unicode):

UTF-8编码是在互联网上使用最广的一种Unicode字符集的实现方式。其他实现方式还包括UTF-16和UTF-32, 不过在互联网上基本不用。

UTF-8编码最大的一个特点, 就是它是一种变长的编码方式。它可以使用1~4个字节表示一个符号, 根据不同的符号而变化字节长度。

UTF-8编码中, 一个英文字符等于一个字节, 一个中文 (含繁体) 等于三个字节。中文标点占三个字节, 英文标点占一个字节。

GBK/GB2312(中文)和GB18030(CJK): 国家编码, 通用型不如UTF-8, 每个字符均占2个字节。

- MySQL中字符集的建议

1. 非常肯定只有中文终端用户时, 可选择gbk/gb2312字符集, 否则选用utf8mb4字符集。

2. 为了方便数据迁移、以及多种终端展示, 最好使用utf8mb4。

3. 在mysql中utf8最多存放3个字节, 而非实际的1-4个字节, 为了不必要的麻烦建议utf8mb4。

4. 字符无需区分大小写时，采用默认的==xxx_general_ci==校验集，否则选择==xxx_bin==校验集(生产环境中，尽量不要修改校验集)。

- 根据字符集查看校验集

```
mysql> show collation where charset = 'utf8mb4';
```

2. systemd管理mysql

==注意：以下mysqld.service文件内容适用于mysql 5.7版本以后（参考）==

```
[root@misshou ~]#vim /usr/lib/systemd/system/mysql.service
[Unit] #该小节包含与单元类型无关的通用信息
#描述信息
Description=MySQL Server
#帮助手册
Documentation=man:mysql(8)
Documentation=http://dev.mysql.com/doc/refman/en/using-systemd.html
#定义unit启动顺序，After表示当前unit晚于以下两个单元启动
After=network.target
After=syslog.target
[Install] #该小节包含单元的启用信息，systemctl的enable与disable命令在启用/停用单元时才会使用此小节。
WantedBy=multi-user.target
#依赖multi-user.target单元，执行systemctl enable 启用命令之后，将会建立一个指向该单元文件的软链接/etc/systemd/system/multi-user.target.wants/mysql.service，表示将mysql.service包含到 multi-user.target目标中，这样，当启动multi-user.target目标时，将会自动启动mysql.service服务。 同时，systemctl disable 命令将会删除这个软连接。

[Service]
User=mysql
Group=mysql
#设置进程的启动类型， 必须设为 simple, forking, oneshot, dbus, notify, idle之一。
如果设为forking，那么表示ExecStart=xxx进程将会在启动过程中使用fork()系统调用。 这是传统UNIX守护进程的经典做法。也就是当所有的通信渠道都已建好、启动亦已成功之后，父进程将会退出，而子进程将作为该服务的主进程继续运行。 对于此种进程，建议同时设置 PIDFile=xxx选项，以帮助systemd准确定位该服务的主进程，进而加快后继单元的启动速度。
如果设置为simple表示ExecStart=xxx进程就是该服务的主进程。
Type=forking
PIDFile=/var/run/mysqld/mysqld.pid
# Disable service start and stop timeout logic of systemd for mysqld service.
#设置该服务允许的最大启动时长。 如果守护进程未能在限定的时长内发出"启动完毕"的信号，那么该服务将被视为启动失败，并会被关闭。 如果未指定时间单位，那么将视为以秒为单位。 例如设为"20"等价于设为"20s"。 设为"infinity" 则表示永不超时。 当 Type=oneshot 时， 默认值为 "infinity" (永不超时)， 否则默认值等于DefaultTimeoutStartSec= 的值(参见 systemd-system.conf(5) 手册)。
TimeoutSec=0
# Execute pre and post scripts as root
#设为yes表示所有与权限相关的执行选项(例如 User= 之类的选项),仅对 ExecStart= 中的程序有效，而对 ExecStartPre=, ExecStartPost=, ExecReload=, ExecStop=, ExecStopPost= 中的程序无效。 默认值 no表示所有与权限相关的执行选项，对所有 Exec*= 系列选项中的程序都有效。
PermissionsStartOnly=true
# Needed to create system tables

#设置在执行ExecStart=之前执行的命令行
```

```

ExecStartPre=/usr/bin/mysqld_pre_systemd
# Start main service
#在启动该服务时需要执行的命令行(命令+参数)。
ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD_OPTS
# Use this to switch malloc implementation
#设置进程的环境变量,文件必须用绝对路径表示(可以包含通配符)。但可在路径前加上"-"前缀表示忽略不存在的文件。
可以多次使用此选项,以从多个不同的文件中读取设置。若设为空,则表示清空所有先前已经从文件中读取的环境变量。
EnvironmentFile=/etc/sysconfig/mysql
# Sets open_files_limit
#进程打开文件的数量(文件描述符的数量), ulimit -n
LimitNOFILE = 5000
#当服务进程正常退出、异常退出、被杀死、超时的时候,是否重新启动该服务。on-failure表示仅在服务进程异常退出时重启
Restart=on-failure
#可以设为一系列以空格分隔的数字退出码或信号名称,当进程的退出码或收到的信号与此处的设置匹配时,无论Restart是如何设置的,该服务都将无条件的禁止重新启动。这里1表示当进程的退出码为1和SIGHUP信号时不会导致该服务被自动重启。
RestartPreventExitStatus=1
#设为true表示在进程的文件系统名字空间中挂载私有的/tmp与/var/tmp目录,也就是不与名字空间外的其他进程共享临时目录。这样做会增加进程的临时文件安全性,但同时也让进程之间无法通过/tmp或/var/tmp目录进行通信。同时,当服务停止之后,所有先前在临时目录中创建的文件都将被删除。默认是false。注意,因为此选项的设置有可能在实际无法落实(例如挂载名字空间不可用),所以,在编写单元文件的时候,不应该将单元的安全依赖于此选项必然生效的假定。
PrivateTmp=false

```

==以下文件内容,适用于mysql5.6版本(参考): ==

```

[Unit]
Description=MySQL Server
Documentation=man:mysqld(8)
Documentation=http://dev.mysql.com/doc/refman/en/using-systemd.html
After=network.target
After=syslog.target
[Install]
WantedBy=multi-user.target
[Service]
User=mysql
Group=mysql
Type=simple //跟5.7不同
# Disable service start and stop timeout logic of systemd for mysql service.
TimeoutSec=0
# Execute pre and post scripts as root
PermissionsStartOnly=true
# Needed to create system tables
#ExecStartPre=/usr/local/mysql/bin/mysqld_pre_systemd
# Start main service
#以下启动方式也可以使用mysqld_safe程序,助于5.6主程序不支持--daemonize选项
ExecStart=/usr/local/mysql/bin/mysqld --defaults-file=/etc/my.cnf
# Use this to switch malloc implementation

EnvironmentFile=/etc/sysconfig/mysql

```

```
# Sets open_files_limit
LimitNOFILE = 5000
Restart=on-failure
PrivateTmp=false
RestartPreventExitStatus=1
```

五、启动报错总结

mysql启动报错：

```
[root@mysql01 mysql]# service mysql35 start
Unit mysql35.service could not be found.
Starting MySQL. ERROR! The server quit without updating PID file
(/data/mysql35/data/mysql01.itcast.cc.pid).
```

原因1：
mysql用户需要在安装目录里创建sock文件和pid文件，权限不足。

原因2：
端口冲突导致

原因3：
tmp目录权限不足

原因4：
配置文件冲突/etc/my.cnf ../my.cnf

入口就是查看错误日志文件：

```
/var/lib/mysql/hostname.err
/usr/local/mysql/data/hostname.err
....
```

数据目录里的错误日志

任务解决方案

环境：

原有服务器：centos 6.5系统 10.1.1.250 部署LAMP环境，并搭建2个网站

新服务器：centos 7.5系统 10.1.1.5 准备部署MySQL-5.6.35版本

停止原有服务器上的相关服务，以下操作在旧服务器上进行：

IP：10.1.1.250

hostname：lamp.itcast.cc

1. 先停止应用服务apache和数据库

```
[root@lamp ~]# service apache stop
[root@lamp ~]# service mysql25 stop
Shutting down MySQL.. SUCCESS!
```

Centos 7.5系统用来安装最新版本的MySQL数据库，以下操作在新版本上进行：

IP：10.1.1.5

hostname：db01.misshou.cc

1. 下载mysql软件

```
[root@db01 installer]# du -sh mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz
301M    mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz
```

2. 安装规划

安装目录(basedir): /usr/local/mysql

数据库目录(datadir): /usr/local/mysql/data

端口: 3307

socket文件: /usr/local/mysql/mysql.sock

错误日志文件: /usr/local/mysql/data/mysql3307.err

3. 进行安装

1) 解压软件包

```
[root@db01 installer]# tar -xf mysql-5.6.35-linux-glibc2.5-x86_64.tar.gz
```

2) 进入到/usr/local目录里软连接

```
[root@db01 installer]# cd /usr/local/
```

```
[root@db01 local]# ln -s /soft/installer/mysql-5.6.35-linux-glibc2.5-x86_64 mysql
```

3) 初始化数据库

创建mysql用户:

```
[root@db01 mysql]# id mysql
```

id: mysql: no such user

```
[root@db01 mysql]# useradd -r -s /sbin/nologin mysql
```

```
[root@db01 mysql]# id mysql
```

uid=997(mysql) gid=995(mysql) 组=995(mysql)

更改mysql的安装目录的权限:

```
[root@db01 mysql]# chown -R mysql:mysql /usr/local/mysql/
```

```
[root@db01 mysql]# ll -d /usr/local/mysql/
```

```
drwxr-xr-x 13 mysql mysql 191 10月 9 10:26 /usr/local/mysql/
```

初始化数据库:

删除系统默认自带的/etc/my.cnf文件

```
[root@db01 mysql]# scripts/mysql_install_db --user=mysql
```

注意:

如果忘记删除原有/etc/my.cnf文件, 那么可以使用以下命令解决:

记得创建/usr/local/mysql/my.cnf/my.cnf文件

```
[mysqld]
```

basedir=/usr/local/mysql

datadir=/usr/local/mysql/data

port=3307

socket=/usr/local/mysql/mysql.sock

log-error=/usr/local/mysql/data/mysql3307.err

```
[root@db01 mysql]# scripts/mysql_install_db --user=mysql --defaults-file=/usr/local/mysql/my.cnf
```

或者直接修改/etc/my.cnf文件来改变数据目录和安装目录即可

4) 启动数据库

修改配置文件

```
[root@db01 mysql]# sed -ri '/^#|^$/d' my.cnf
```

```
[root@db01 mysql]# cat my.cnf
```

```
[mysqld]
```

```
basedir = /usr/local/mysql
datadir = /usr/local/mysql/data
port = 3307
socket = /usr/local/mysql/mysql.sock
log-error = /usr/local/mysql/data/mysql3307.err
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
启动数据库
```

```
[root@db01 mysql]# cp support-files/mysql.server /etc/init.d/mysql
[root@db01 mysql]# service mysql start
Starting MySQL.Logging to '/usr/local/mysql/data/mysql3307.err'.
SUCCESS!
```

5) 修改数据库密码 略

将原有环境中的mysql数据库拷贝到新服务器中:

1. 先删除新环境中/usr/local/mysql/data/目录下的所有文件

```
rm -rf /usr/local/mysql/data/*
```

2. 使用rsync同步原有数据库的文件到新数据中

```
rsync -av 10.1.1.250:/usr/local/mysql/data/ /usr/local/mysql/data
```

3. 直接启动新的数据库, 并且检查日志文件 (查看是否有选项不兼容)

```
service mysql start
```

4. 使用mysql_upgrade命令检查数据文件的兼容性

```
mysql_upgrade -p123
```

注意:

在实际的环境中, 建议使用-s参数, 不检查业务表, 只检查系统文件

前端应用程序连接远程的mysql数据库, 以下操作在原来的服务器上进行:

10.1.1.250

1. 修改php连接mysql的配置文件

```
[root@lamp ~]# find / -name php.ini
/usr/local/lib/php.ini
[root@lamp ~]# vim /usr/local/lib/php.ini
[MySQL]
mysql.default_port = 3307
mysql.default_host = 10.1.1.5

[MySQLi]
mysql.default_port = 3307
mysql.default_host = 10.1.1.5
```

2. 修改www.myadmin.cc网站所在的数据目录里的配置文件

```
[root@lamp admin]# pwd
/www/admin

[root@lamp admin]# vim config.inc.php
```

```
/* Server parameters */
$cfg['Servers'][$i]['host'] = '10.1.1.5';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

3. 修改www.myblog.net网站所在数据库目录的配置文件

```
[root@lamp myblog]# pwd
/www/myblog
[root@lamp myblog]# vim wp-config.php
...
// ** MySQL 设置 - 具体信息来自您正在使用的主机 ** //
/** WordPress数据库的名称 */
define('DB_NAME', 'myblog');

/** MySQL数据库用户名 */
define('DB_USER', 'root');

/** MySQL数据库密码 */
define('DB_PASSWORD', '123');

/** MySQL主机 */
define('DB_HOST', '10.1.1.5');

/** 创建数据表时默认的文字编码 */
define('DB_CHARSET', 'utf8');
```

4. 新的mysql数据库需要给root@10.1.1.250用户授权

```
mysql> grant all on *.* to 'root'@'10.1.1.250' identified by '123';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

5. 启动web服务

```
[root@lamp ~]# service apache start
[root@lamp ~]# netstat -nltp|grep httpd
tcp        0      0 :::80                :::*                  LISTEN
1376/httpd
```

6. 测试验证

浏览器访问

web服务器上直接访问:

```
[root@lamp ~]# mysql -uroot -h10.1.1.5 -P3307 -p123
```

注解:

1. glibc是GNU发布的libc库，即c运行库;glibc是linux系统中最底层的api，几乎其它任何运行库都会依赖于glibc。[↗](#)