参考：<http://blog.csdn.net/sungsasong/article/details/76278361>

[**最新版本Mysql 5.7.19三种安装方式手册**](http://blog.csdn.net/sungsasong/article/details/76278361)

标签： [mysql](http://www.csdn.net/tag/mysql)[mysql安装](http://www.csdn.net/tag/mysql%e5%ae%89%e8%a3%85)

2017-07-28 19:36 834人阅读 [评论](http://blog.csdn.net/sungsasong/article/details/76278361#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/sungsasong/article/details/76278361#report)

 分类：

mysql（1）

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)[+]](http://blog.csdn.net/sungsasong/article/details/76278361)

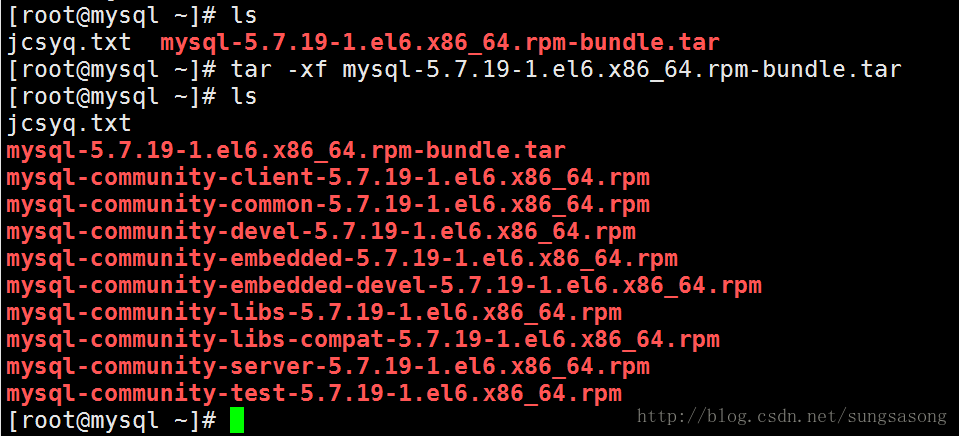
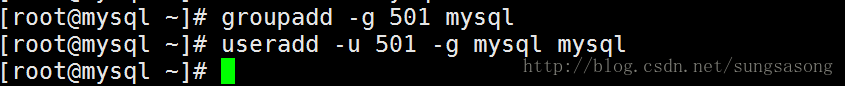
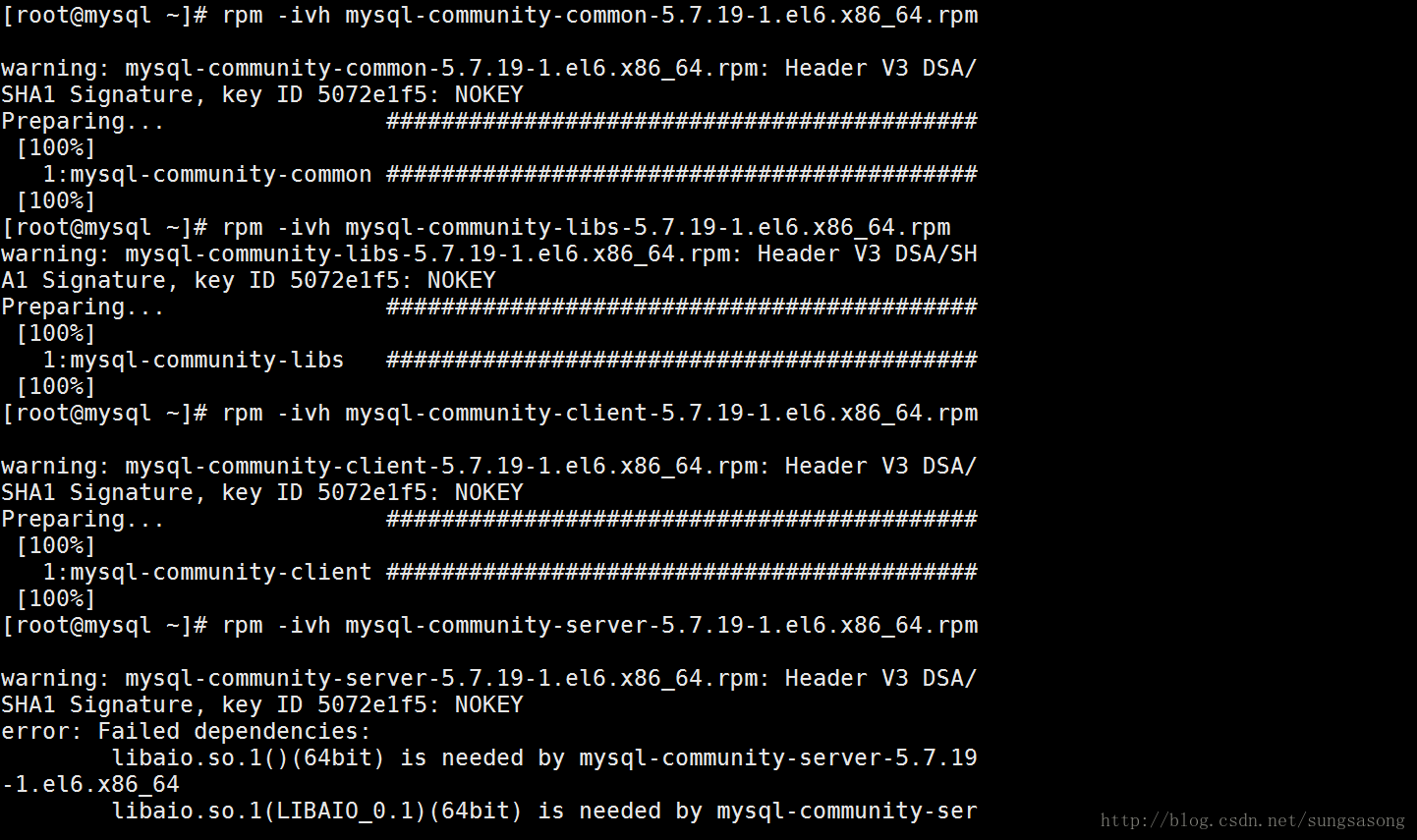
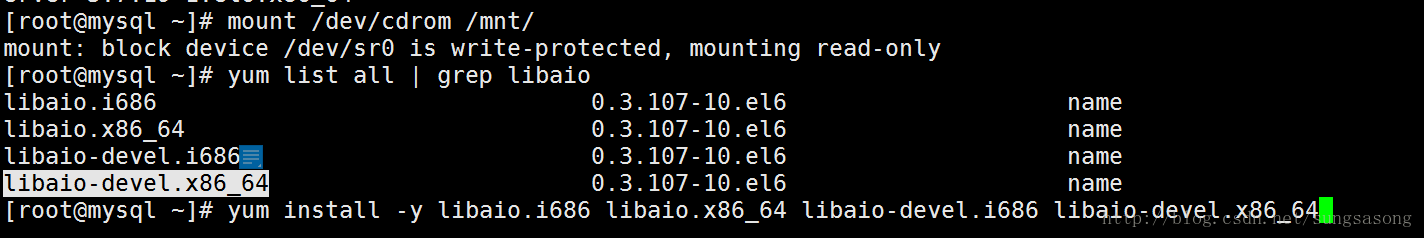
\*\*

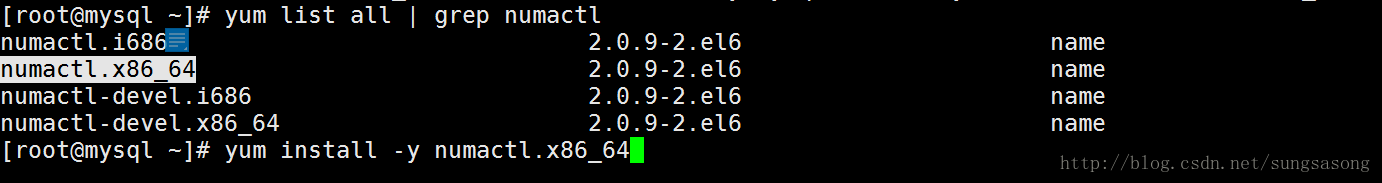
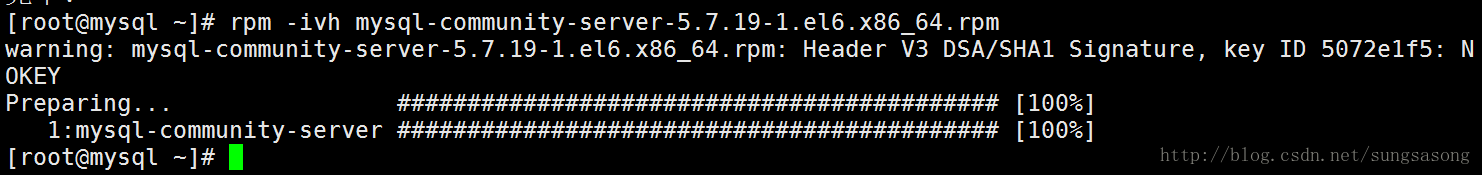
环境准备

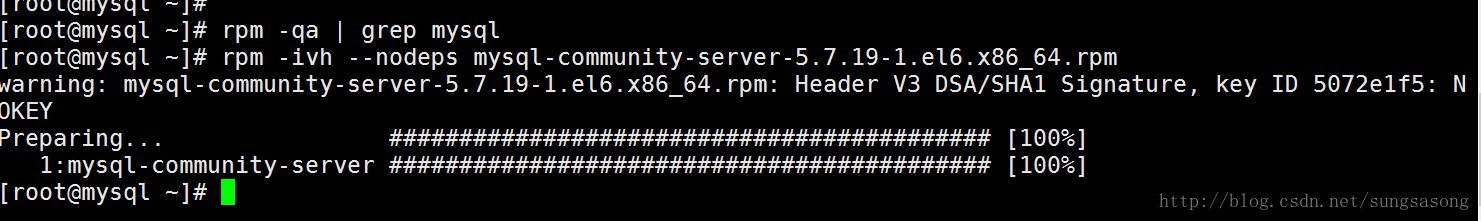
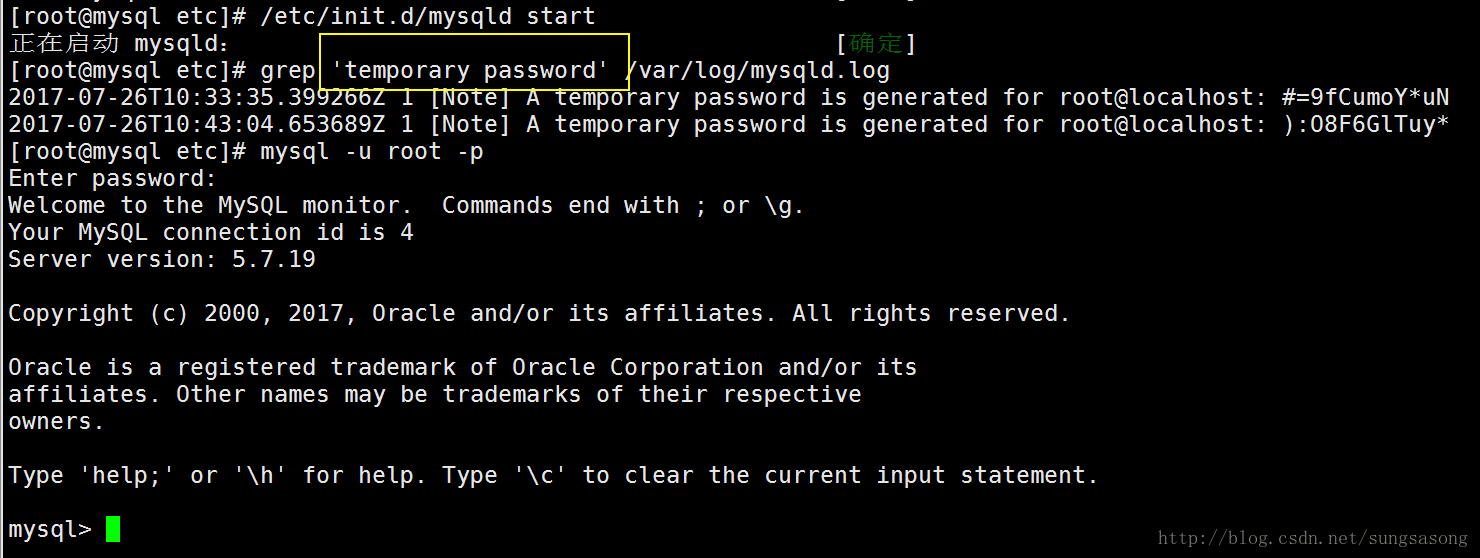
[**操作系统**](http://lib.csdn.net/base/operatingsystem)：CentOS   
软件：[**MySQL**](http://lib.csdn.net/base/mysql)-5.7.19   
\*\*   
\*\*

一、RPM 方式安装

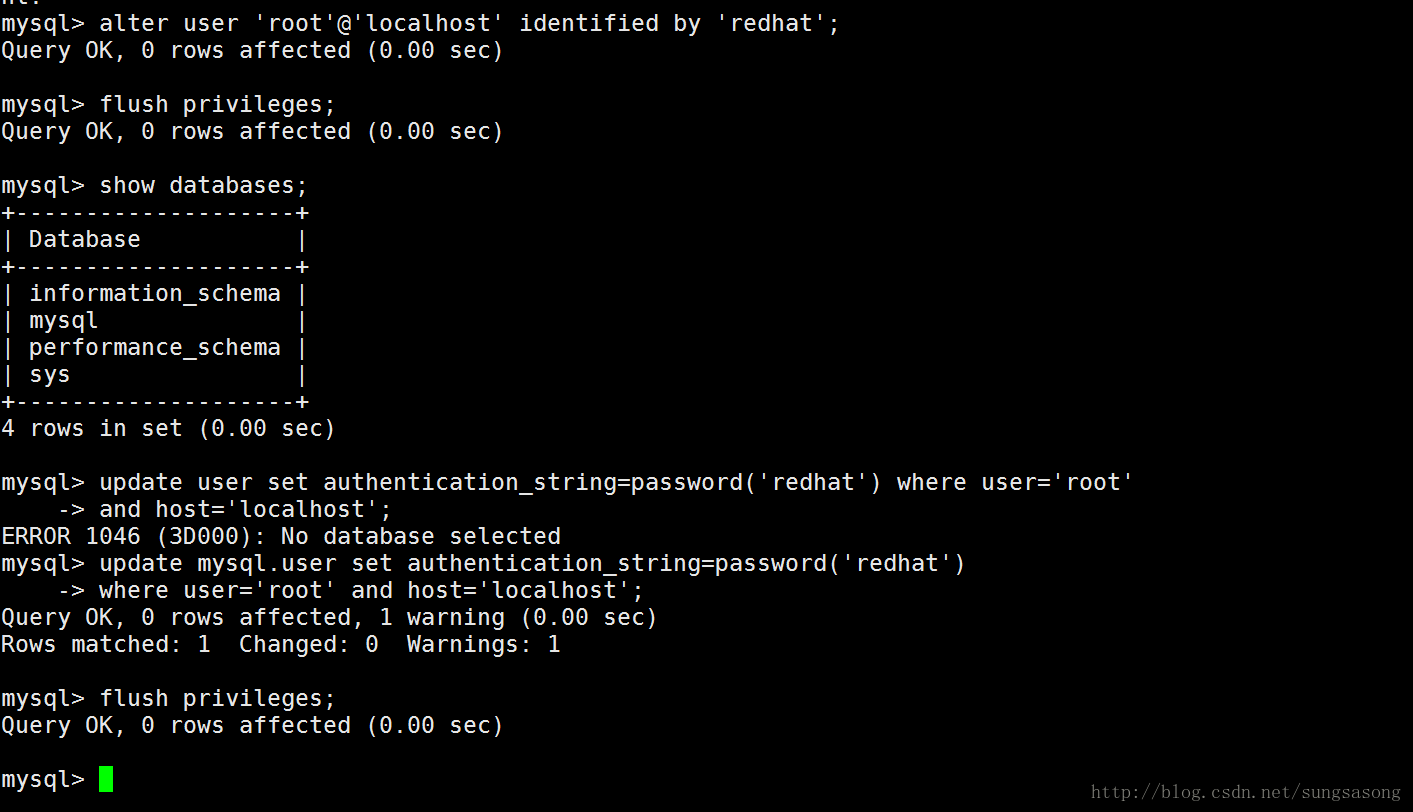
进入官方网站：[https://www.mysql.com](https://www.mysql.com/)   
注册账号

[downloads](https://www.mysql.com/downloads/)   
选择[**mysql**](http://lib.csdn.net/base/mysql) Community Edition (GPL)>>[Community (GPL) Downloads](https://dev.mysql.com/downloads/) >>MySQL Community Server (GPL)>>[download](https://dev.mysql.com/downloads/mysql/)   
操作系统：Red Hat Enterprise & [**Oracle**](http://lib.csdn.net/base/oracle) [**Linux**](http://lib.csdn.net/base/linux)   
操作系统版本：Redhat 64   
\*\*   
**1、上传Mysql rpm 包到**[**linux**](http://lib.csdn.net/base/linux)**服务器并解压**   
   
**2、创建Mysql用户和用户组**   
   
**3、安装Mysql**   
   
**发现出错了，有两个依赖包没有安装，因此需要安装此依赖包，依赖包的为libaio和numactl，安装即可**   


   
再次安装服务端：   


说明：这里rpm如果不采用–nodeps必须将上面的四个包依次进行安装，否则会提示依赖解决。   
**只安装服务端，不安装通用包和libs包**   
   
**4、启动Mysql并登录到Mysql**[**数据库**](http://lib.csdn.net/base/mysql)   
   
**说明：由于第一次启动Mysql，并且如果没有在/etc/my.cnf中添加参数skip-grant-tables参数的话，那么是无法登录到Mysql数据库的，这里有两种解决三种解决方法：**   
**第一种，mysql初次初始化后，在/var/log/mysqld.log中提供了一个可以用于登录的临时密码，使用这个密码，可以登录到数据库，如上图**  
**第二种，在/etc/my.cnf中添加skip-grant-tables参数，不过需要重新启动数据库才可以生效。**   
**第三种，直接使用命令，/etc/init.d/mysqld –skip-grant-tables，然后重新打开一个窗口，如果不想重新打开一个窗口，那么在末尾添加一个**

&

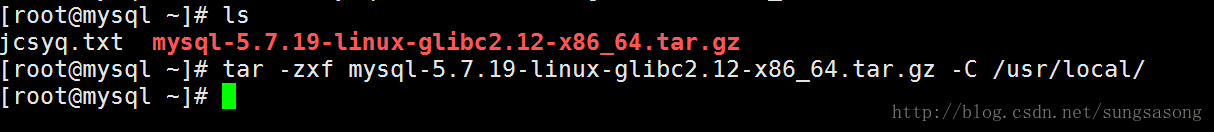
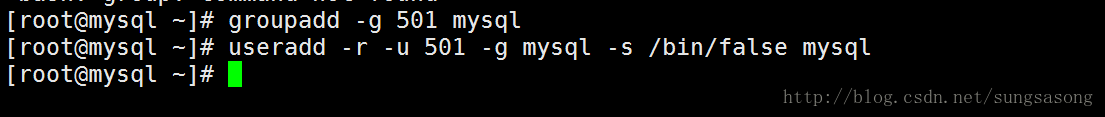
**符号，让其在后台运行。**   
**5、初次使用修改密码**   


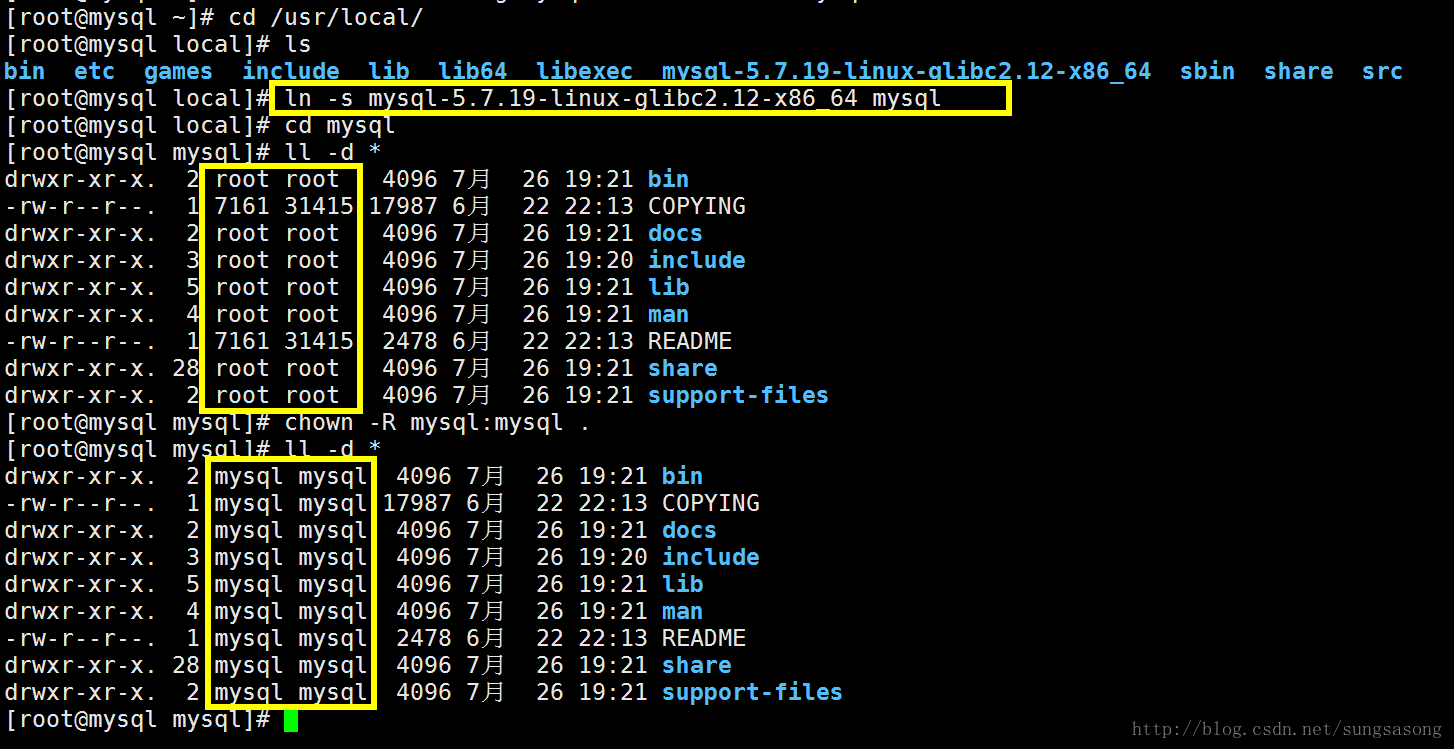
**说明：在以前的版本中，Mysql对于密码的修改，是在mysql库中的user表中有一列为password列，但是从msyql 5.7版本开始，**[**oracle**](http://lib.csdn.net/base/oracle)**对mysql搞了一点事情，它把原有的user表中的column password取消掉了，使用authentication\_string来替代了，所以，现在如果使用update修改表user中的密码必须使用update user set authentication\_string=password(‘your\_password’) where user=’user\_name’ and host=’servcie\_host\_ip’；而不是原来的update user set password了，所以在这里Oracle做了一些改动。并且，mysql从5.6开始，开始强化密码，要求密码中必须使用至少有一个大写字母和小写字母，数字和特殊字符，并且长度不能少于8位，这在windows图形界面安装下非常明显，但是在Linux下，我为了方便记忆，并没有遵守Mysql的约定，所以请引起注意。**

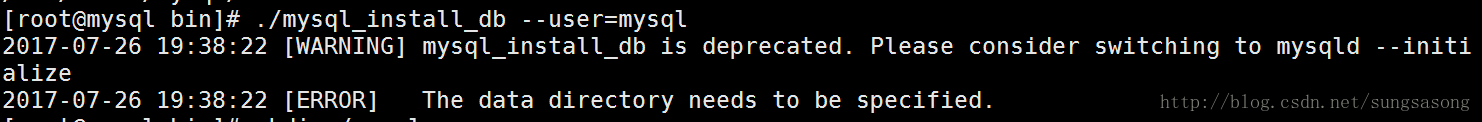
\*\*

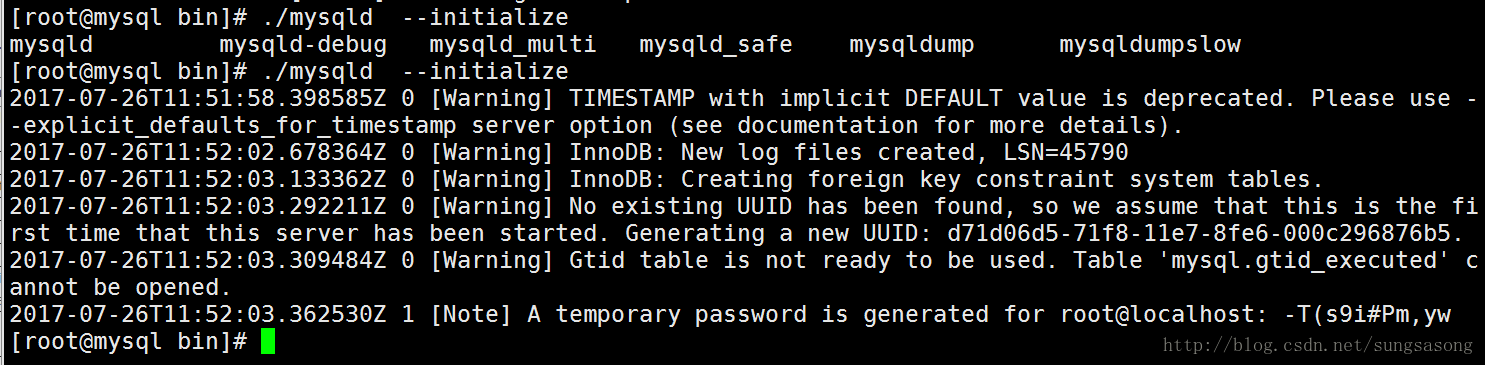
二、二进制安装Mysql

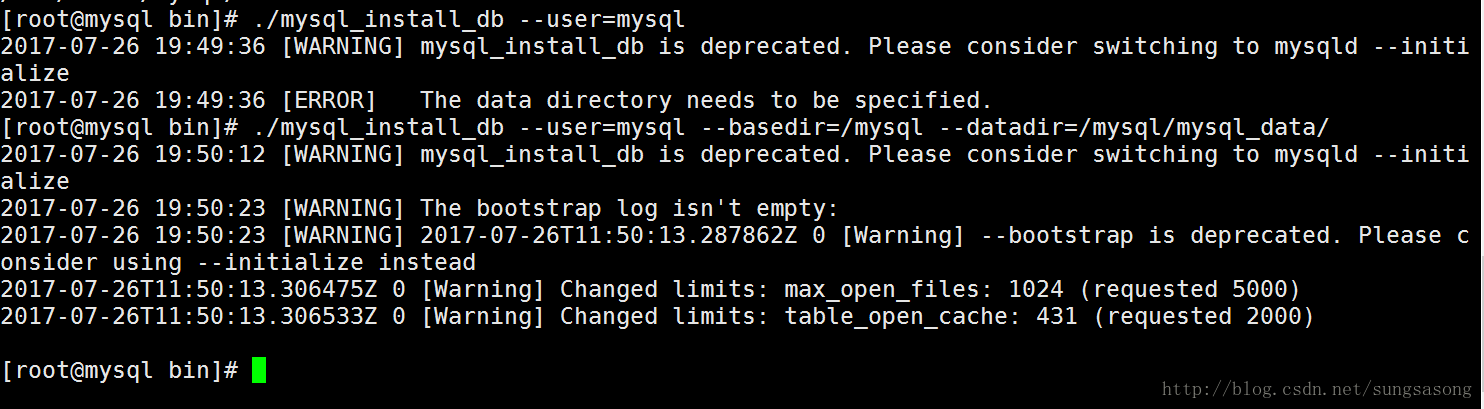
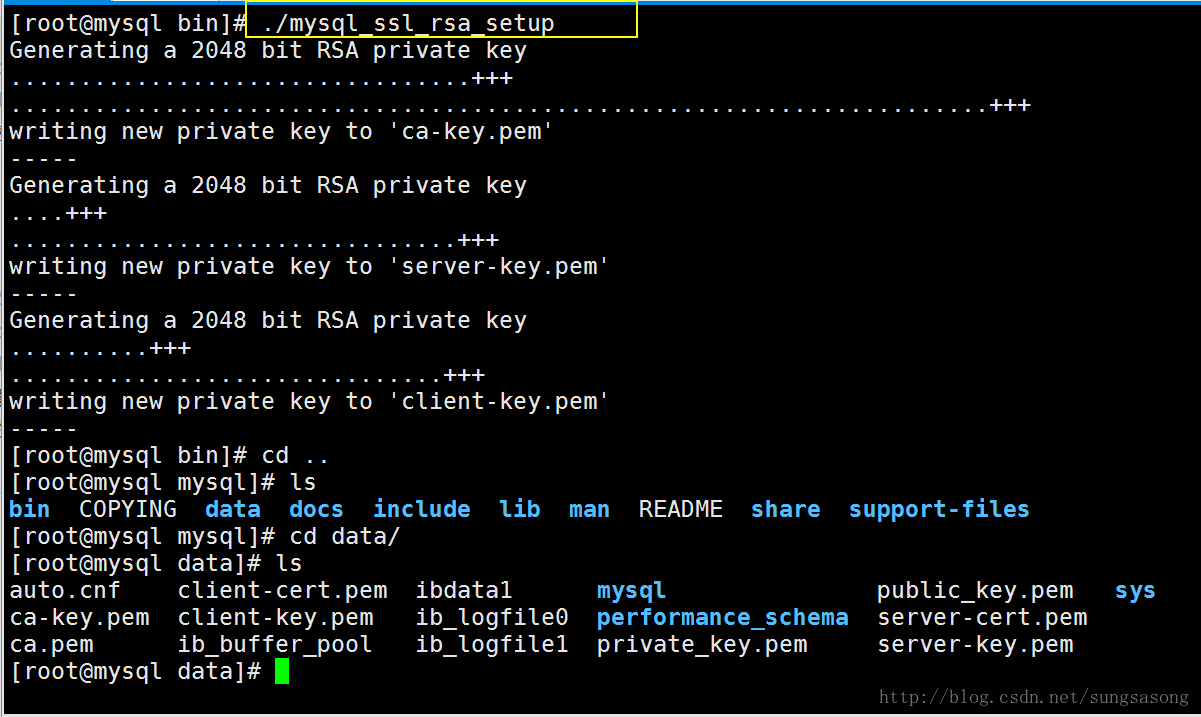
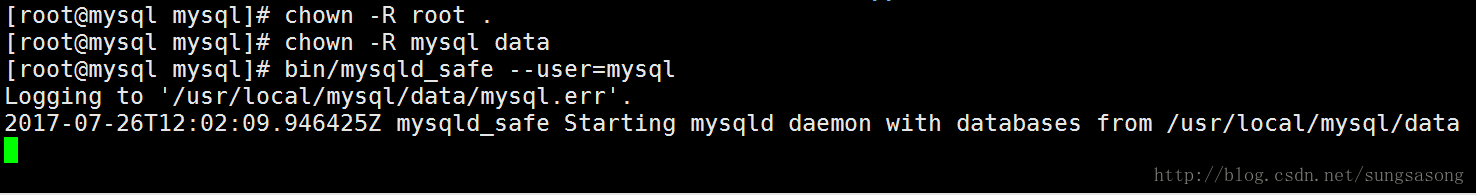
\*\*   
进入官方网站：[https://www.mysql.com](https://www.mysql.com/)   
注册账号

[downloads](https://www.mysql.com/downloads/)   
选择MySQL Community Edition (GPL)>>[Community (GPL) Downloads](https://dev.mysql.com/downloads/) >>MySQL Community Server (GPL)>>[download](https://dev.mysql.com/downloads/mysql/)   
操作系统：Linux\_Generic   
操作系统版本：Linux\_generic x86\_64   
**1、上传msyql二进制包致Linux服务器，并解压到/usr/local目录**   
   
**2、创建Mysql用户和用户组**   
 

**说明: -r 参数表示login -s 指定默认的shell**   
**3、对/usr/local下的mysql解压包做软连接(相当于windows下的快捷方式)并授权**   
   
**4、执行mysql\_install\_db脚本并指定用户**

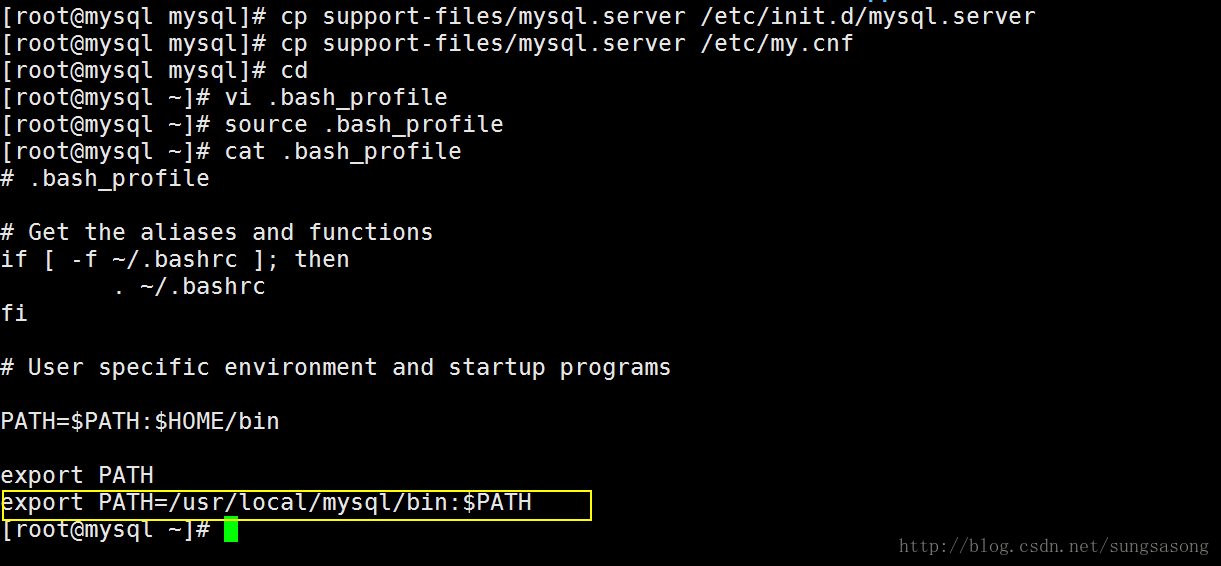
**在这里有必要进行提前说明，在以前执行mysql\_install\_db都是在scripts目录下，但是从Mysql 5.7.5开是，Oracle又搞了一些事情，把scripts拿掉了，直接把mysql\_install\_db放在bin目录下了，并且从 mysqld –initialize这个脚本也是从5.7.6开始加上去的，现在的版本是5.7.19，说明这个脚本加入的时间并不是怎么长**   
   
**注意，上面出错了，第一个只是warning，说明并没有错，只是在这里警告说这样执行mysql\_install\_db命令不是推荐的方式，给出的建议是使用mysqld –initialize 的方式**   
**第二个的确是错误，说，必须要指定存放数据文件的目录**

**首先解决第一个警告**   
   
OK！没有问题，并且和 RPM一样，最后生成了一个临时密码，这个密码用来登陆Mysql，以便于对初始化的用户名的密码进行修改设置。

**解决第二个错误，在这里依然使用mysql\_install\_db –user的方式，既然要求我们指定一个数据文件的目录，那么我们给指定一个不就行了。**  
   
**OK!都是一些警告，这只是**[**测试**](http://lib.csdn.net/base/softwaretest)**环境，并不是生产环境，在生产环境中，为了保证数据库宕机的可能性，一些参数必须要进行设置和指定，特别是性能和基准测试。**   
**5、执行mysql\_ssl\_rsa\_setup脚本进行进行密匙初始化**   
**mysql\_ssl\_rsa\_setup这个脚本也是从mysql 5.7.6以后加入的性特性，旨在增强安全加密连接**   
   
**执行这个脚本后会生成采用RSA加密的三个文件，一个为客户端连接加密安全文件，一个为服务端连接加密安全文件，还有一个为CA认证文件**   
**6、修改目录权限，并执行mysqld\_safe脚本**   
   
**可以在mysql\_safe –user=msyql加一个**

&

**以便于在后台运行**

**7、拷贝配置文件并配置环境变量**   
   
**8、执行自带的mysql\_secure\_installation命令强化安全脚本，这个脚本将会删除匿名用户**   
鉴于时间限制就不上传图片了   
也可以进行手动删除   
执行

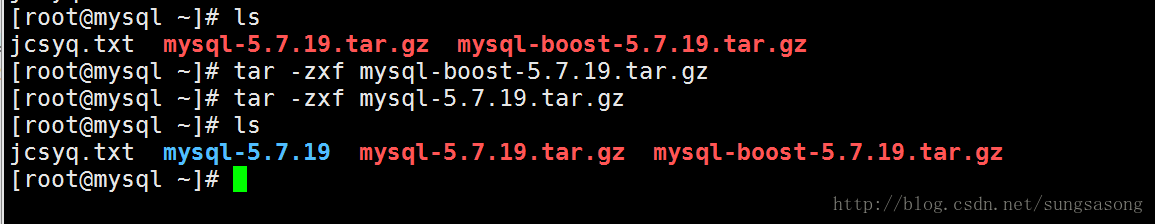
DELETE FROM mysql.user WHERE user='';

FLUSH PRIVILEGES;

\*\*

三、源码安装

\*\*   
进入官方网站：[https://www.mysql.com](https://www.mysql.com/)   
注册账号

[downloads](https://www.mysql.com/downloads/)   
选择MySQL Community Edition (GPL)>>[Community (GPL) Downloads](https://dev.mysql.com/downloads/) >>MySQL Community Server (GPL)>>[download](https://dev.mysql.com/downloads/mysql/)   
操作系统：Source Code   
操作系统版本:Generic\_Linux   
**1、老套路，将源码包上传之Linux服务器**   
   
**说明：这里有两个包，在5.7以后，必须要有boost这个包**   
**2、安装编译环境所需要的包和一些依赖包**

[root@mysql ~]# yum install -y cmake make gcc gcc-c++ ncurses-devel

已加载插件：fastestmirror

设置安装进程

Loading mirror speeds from cached hostfile

包 1:make-3.81-20.el6.x86\_64 已安装并且是最新版本

解决依赖关系

--> 执行事务检查

---> Package cmake.x86\_64 0:2.8.12.2-4.el6 will be 安装

--> 处理依赖关系 libarchive.so.2()(64bit)，它被软件包 cmake-2.8.12.2-4.el6.x86\_64 需要

---> Package gcc.x86\_64 0:4.4.7-11.el6 will be 安装

--> 处理依赖关系 libgomp = 4.4.7-11.el6，它被软件包 gcc-4.4.7-11.el6.x86\_64 需要

--> 处理依赖关系 cpp = 4.4.7-11.el6，它被软件包 gcc-4.4.7-11.el6.x86\_64 需要

--> 处理依赖关系 glibc-devel >= 2.2.90-12，它被软件包 gcc-4.4.7-11.el6.x86\_64 需要

--> 处理依赖关系 cloog-ppl >= 0.15，它被软件包 gcc-4.4.7-11.el6.x86\_64 需要

--> 处理依赖关系 libgomp.so.1()(64bit)，它被软件包 gcc-4.4.7-11.el6.x86\_64 需要

---> Package gcc-c++.x86\_64 0:4.4.7-11.el6 will be 安装

--> 处理依赖关系 libstdc++-devel = 4.4.7-11.el6，它被软件包 gcc-c++-4.4.7-11.el6.x86\_64 需要

--> 处理依赖关系 libmpfr.so.1()(64bit)，它被软件包 gcc-c++-4.4.7-11.el6.x86\_64 需要

---> Package ncurses-devel.x86\_64 0:5.7-3.20090208.el6 will be 安装

--> 执行事务检查

---> Package cloog-ppl.x86\_64 0:0.15.7-1.2.el6 will be 安装

--> 处理依赖关系 libppl\_c.so.2()(64bit)，它被软件包 cloog-ppl-0.15.7-1.2.el6.x86\_64 需要

--> 处理依赖关系 libppl.so.7()(64bit)，它被软件包 cloog-ppl-0.15.7-1.2.el6.x86\_64 需要

---> Package cpp.x86\_64 0:4.4.7-11.el6 will be 安装

---> Package glibc-devel.x86\_64 0:2.12-1.149.el6 will be 安装

--> 处理依赖关系 glibc-headers = 2.12-1.149.el6，它被软件包 glibc-devel-2.12-1.149.el6.x86\_64 需要

--> 处理依赖关系 glibc-headers，它被软件包 glibc-devel-2.12-1.149.el6.x86\_64 需要

---> Package libarchive.x86\_64 0:2.8.3-4.el6\_2 will be 安装

---> Package libgomp.x86\_64 0:4.4.7-11.el6 will be 安装

---> Package libstdc++-devel.x86\_64 0:4.4.7-11.el6 will be 安装

---> Package mpfr.x86\_64 0:2.4.1-6.el6 will be 安装

--> 执行事务检查

---> Package glibc-headers.x86\_64 0:2.12-1.149.el6 will be 安装

--> 处理依赖关系 kernel-headers >= 2.2.1，它被软件包 glibc-headers-2.12-1.149.el6.x86\_64 需要

--> 处理依赖关系 kernel-headers，它被软件包 glibc-headers-2.12-1.149.el6.x86\_64 需要

---> Package ppl.x86\_64 0:0.10.2-11.el6 will be 安装

--> 执行事务检查

---> Package kernel-headers.x86\_64 0:2.6.32-504.el6 will be 安装

--> 完成依赖关系计算

依赖关系解决

=========================================================================================================

软件包 架构 版本 仓库 大小

=========================================================================================================

正在安装:

cmake x86\_64 2.8.12.2-4.el6 name 8.0 M

gcc x86\_64 4.4.7-11.el6 name 10 M

gcc-c++ x86\_64 4.4.7-11.el6 name 4.7 M

ncurses-devel x86\_64 5.7-3.20090208.el6 name 642 k

为依赖而安装:

cloog-ppl x86\_64 0.15.7-1.2.el6 name 93 k

cpp x86\_64 4.4.7-11.el6 name 3.7 M

glibc-devel x86\_64 2.12-1.149.el6 name 983 k

glibc-headers x86\_64 2.12-1.149.el6 name 611 k

kernel-headers x86\_64 2.6.32-504.el6 name 3.3 M

libarchive x86\_64 2.8.3-4.el6\_2 name 126 k

libgomp x86\_64 4.4.7-11.el6 name 133 k

libstdc++-devel x86\_64 4.4.7-11.el6 name 1.6 M

mpfr x86\_64 2.4.1-6.el6 name 157 k

ppl x86\_64 0.10.2-11.el6 name 1.3 M

事务概要

=========================================================================================================

Install 14 Package(s)

总下载量：35 M

Installed size: 89 M

下载软件包：

---------------------------------------------------------------------------------------------------------

总计 63 MB/s | 35 MB 00:00

运行 rpm\_check\_debug

执行事务测试

事务测试成功

执行事务

正在安装 : mpfr-2.4.1-6.el6.x86\_64 1/14

正在安装 : cpp-4.4.7-11.el6.x86\_64 2/14

正在安装 : kernel-headers-2.6.32-504.el6.x86\_64 3/14

正在安装 : glibc-headers-2.12-1.149.el6.x86\_64 4/14

正在安装 : glibc-devel-2.12-1.149.el6.x86\_64 5/14

正在安装 : libstdc++-devel-4.4.7-11.el6.x86\_64 6/14

正在安装 : libgomp-4.4.7-11.el6.x86\_64 7/14

正在安装 : ppl-0.10.2-11.el6.x86\_64 8/14

正在安装 : cloog-ppl-0.15.7-1.2.el6.x86\_64 9/14

正在安装 : gcc-4.4.7-11.el6.x86\_64 10/14

正在安装 : libarchive-2.8.3-4.el6\_2.x86\_64 11/14

正在安装 : cmake-2.8.12.2-4.el6.x86\_64 12/14

正在安装 : gcc-c++-4.4.7-11.el6.x86\_64 13/14

正在安装 : ncurses-devel-5.7-3.20090208.el6.x86\_64 14/14

Verifying : gcc-4.4.7-11.el6.x86\_64 1/14

Verifying : gcc-c++-4.4.7-11.el6.x86\_64 2/14

Verifying : libarchive-2.8.3-4.el6\_2.x86\_64 3/14

Verifying : ppl-0.10.2-11.el6.x86\_64 4/14

Verifying : glibc-devel-2.12-1.149.el6.x86\_64 5/14

Verifying : mpfr-2.4.1-6.el6.x86\_64 6/14

Verifying : glibc-headers-2.12-1.149.el6.x86\_64 7/14

Verifying : libgomp-4.4.7-11.el6.x86\_64 8/14

Verifying : cmake-2.8.12.2-4.el6.x86\_64 9/14

Verifying : libstdc++-devel-4.4.7-11.el6.x86\_64 10/14

Verifying : cpp-4.4.7-11.el6.x86\_64 11/14

Verifying : kernel-headers-2.6.32-504.el6.x86\_64 12/14

Verifying : ncurses-devel-5.7-3.20090208.el6.x86\_64 13/14

Verifying : cloog-ppl-0.15.7-1.2.el6.x86\_64 14/14

已安装:

cmake.x86\_64 0:2.8.12.2-4.el6 gcc.x86\_64 0:4.4.7-11.el6 gcc-c++.x86\_64 0:4.4.7-11.el6

ncurses-devel.x86\_64 0:5.7-3.20090208.el6

作为依赖被安装:

cloog-ppl.x86\_64 0:0.15.7-1.2.el6 cpp.x86\_64 0:4.4.7-11.el6

glibc-devel.x86\_64 0:2.12-1.149.el6 glibc-headers.x86\_64 0:2.12-1.149.el6

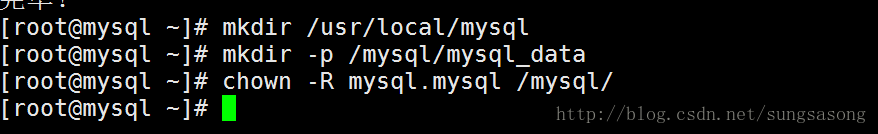
kernel-headers.x86\_64 0:2.6.32-504.el6 libarchive.x86\_64 0:2.8.3-4.el6\_2

libgomp.x86\_64 0:4.4.7-11.el6 libstdc++-devel.x86\_64 0:4.4.7-11.el6

mpfr.x86\_64 0:2.4.1-6.el6 ppl.x86\_64 0:0.10.2-11.el6

完毕！

[root@mysql ~]#

**3、创建必要的目录，并授权**   
   
**4、执行编译命令**

[root@mysql mysql-5.7.19]# cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DMYSQL\_DATADIR=/mysql/mysql\_data/

-- Running cmake version 2.8.12.2

-- Could NOT find Git (missing: GIT\_EXECUTABLE)

-- Configuring with MAX\_INDEXES = 64U

-- The C compiler identification is GNU 4.4.7

-- The CXX compiler identification is GNU 4.4.7

-- Check for working C compiler: /usr/bin/cc

-- Check for working C compiler: /usr/bin/cc -- works

-- Detecting C compiler ABI info

-- Detecting C compiler ABI info - done

-- Check for working CXX compiler: /usr/bin/c++

-- Check for working CXX compiler: /usr/bin/c++ -- works

-- Detecting CXX compiler ABI info

-- Detecting CXX compiler ABI info - done

-- CMAKE\_GENERATOR: Unix Makefiles

-- Looking for SHM\_HUGETLB

-- Looking for SHM\_HUGETLB - found

-- Looking for sys/types.h

-- Looking for sys/types.h - found

-- Looking for stdint.h

-- Looking for stdint.h - found

-- Looking for stddef.h

-- Looking for stddef.h - found

-- Check size of void \*

-- Check size of void \* - done

-- SIZEOF\_VOIDP 8

-- Performing Test HAVE\_C\_SHIFT\_OR\_OPTIMIZATION\_BUG

-- Performing Test HAVE\_C\_SHIFT\_OR\_OPTIMIZATION\_BUG - Failed

-- Performing Test HAVE\_CXX\_SHIFT\_OR\_OPTIMIZATION\_BUG

-- Performing Test HAVE\_CXX\_SHIFT\_OR\_OPTIMIZATION\_BUG - Failed

-- Performing Test HAVE\_C\_FLOATING\_POINT\_FUSED\_MADD

-- Performing Test HAVE\_C\_FLOATING\_POINT\_FUSED\_MADD - Failed

-- Performing Test HAVE\_CXX\_FLOATING\_POINT\_FUSED\_MADD

-- Performing Test HAVE\_CXX\_FLOATING\_POINT\_FUSED\_MADD - Failed

-- Performing Test HAVE\_C\_FP\_CONTRACT\_FLAG

-- Performing Test HAVE\_C\_FP\_CONTRACT\_FLAG - Failed

-- Performing Test HAVE\_CXX\_FP\_CONTRACT\_FLAG

-- Performing Test HAVE\_CXX\_FP\_CONTRACT\_FLAG - Failed

-- MySQL 5.7.19

-- Packaging as: mysql-5.7.19-Linux-x86\_64

-- Looked for boost/version.hpp in and

-- BOOST\_INCLUDE\_DIR BOOST\_INCLUDE\_DIR-NOTFOUND

-- LOCAL\_BOOST\_DIR

-- LOCAL\_BOOST\_ZIP

-- Could not find (the correct version of) boost.

-- MySQL currently requires boost\_1\_59\_0

CMake Error at cmake/boost.cmake:81 (MESSAGE):

You can download it with -DDOWNLOAD\_BOOST=1 -DWITH\_BOOST=<directory>

This CMake script will look for boost in <directory>. If it is not there,

it will download and unpack it (in that directory) for you.

If you are inside a firewall, you may need to use an http proxy:

export http\_proxy=http://example.com:80

Call Stack (most recent call first):

cmake/boost.cmake:238 (COULD\_NOT\_FIND\_BOOST)

CMakeLists.txt:491 (INCLUDE)

-- Configuring incomplete, errors occurred!

See also "/root/mysql-5.7.19/CMakeFiles/CMakeOutput.log".

See also "/root/mysql-5.7.19/CMakeFiles/CMakeError.log".

[root@mysql mysql-5.7.19]#

**执行结果中出现了**

CMake Error at cmake/boost.cmake:81 (MESSAGE):

You can download it with -DDOWNLOAD\_BOOST=1 -DWITH\_BOOST=<directory>

This CMake script will look for boost in <directory>. If it is not there,

it will download and unpack it (in that directory) for you.\*\*

**并且给出了解决方法**

[root@mysql mysql-5.7.19]# cmake -DDOWNLOAD\_BOOST=1 -DWITH\_BOOST=/root/mysql-5.7.19/boost/boost\_1\_59\_0/

-- Running cmake version 2.8.12.2

-- Could NOT find Git (missing: GIT\_EXECUTABLE)

-- Configuring with MAX\_INDEXES = 64U

-- CMAKE\_GENERATOR: Unix Makefiles

-- SIZEOF\_VOIDP 8

-- MySQL 5.7.19

-- Packaging as: mysql-5.7.19-Linux-x86\_64

-- Local boost dir /root/mysql-5.7.19/boost/boost\_1\_59\_0

-- Found /root/mysql-5.7.19/boost/boost\_1\_59\_0/boost/version.hpp

-- BOOST\_VERSION\_NUMBER is #define BOOST\_VERSION 105900

-- BOOST\_INCLUDE\_DIR /root/mysql-5.7.19/boost/boost\_1\_59\_0

-- Performing Test HAVE\_LLVM\_LIBCPP

-- Performing Test HAVE\_LLVM\_LIBCPP - Failed

。。。。。。。。。。

-- Looking for EVFILT\_TIMER - not found

-- Check if the system is big endian

-- Searching 16 bit integer

-- Check size of unsigned short

-- Check size of unsigned short - done

-- Using unsigned short

--。。。。。。。。。。。。

-- Performing Test HAVE\_VISIBILITY\_HIDDEN

-- Performing Test HAVE\_VISIBILITY\_HIDDEN - Success

-- Checking stack direction : -1

-- Looking for include files time.h, sys/time.h

-- Looking for include files time.h, sys/time.h - found

--。。。。。。。。

-- Searching 16 bit integer

-- Using unsigned short

-- Check if the system is big endian - little endian

-- Found ZLIB: zlib (found version "1.2.3")

-- Googletest was not found. gtest-based unit tests will be disabled. You can run cmake . -DENABLE\_DOWNLOADS=1 to automatically download and build required components from source.

-- If you are inside a firewall, you may need to use an https proxy: export https\_proxy=http://example.com:80

-- Performing Test HAVE\_MISLEADING\_INDENTATION

-- Performing Test HAVE\_MISLEADING\_INDENTATION - Failed

CMake Warning at cmake/bison.cmake:20 (MESSAGE):

Bison executable not found in PATH

Call Stack (most recent call first):

sql/CMakeLists.txt:548 (INCLUDE)

CMake Warning at cmake/bison.cmake:20 (MESSAGE):

Bison executable not found in PATH

Call Stack (most recent call first):

libmysqld/CMakeLists.txt:187 (INCLUDE)

-- Library mysqlserver depends on OSLIBS -lpthread;m;rt;crypt;dl

-- INSTALL mysqlclient.pc lib/pkgconfig

-- Skipping deb packaging on unsupported platform .

-- CMAKE\_BUILD\_TYPE: RelWithDebInfo

-- COMPILE\_DEFINITIONS: \_GNU\_SOURCE;\_FILE\_OFFSET\_BITS=64;HAVE\_CONFIG\_H;HAVE\_LIBEVENT1

-- CMAKE\_C\_FLAGS: -Wall -Wextra -Wformat-security -Wvla -Wwrite-strings -Wdeclaration-after-statement

-- CMAKE\_CXX\_FLAGS: -Wall -Wextra -Wformat-security -Wvla -Woverloaded-virtual -Wno-unused-parameter

-- CMAKE\_C\_LINK\_FLAGS:

-- CMAKE\_CXX\_LINK\_FLAGS:

-- CMAKE\_C\_FLAGS\_RELWITHDEBINFO: -O3 -g -fabi-version=2 -fno-omit-frame-pointer -fno-strict-aliasing -DDBUG\_OFF

-- CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO: -O3 -g -fabi-version=2 -fno-omit-frame-pointer -fno-strict-aliasing -DDBUG\_OFF

-- Configuring done

-- Generating done

-- Build files have been written to: /root/mysql-5.7.19

[root@mysql mysql-5.7.19]#

OK！完成。重新执行上面的命令

[root@mysql mysql-5.7.19]# cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DMYSQL\_DATADIR=/mysql/mysql\_data/

-- Running cmake version 2.8.12.2

-- Could NOT find Git (missing: GIT\_EXECUTABLE)

-- Configuring with MAX\_INDEXES = 64U

-- CMAKE\_GENERATOR: Unix Makefiles

-- SIZEOF\_VOIDP 8

-- MySQL 5.7.19

-- Packaging as: mysql-5.7.19-Linux-x86\_64

-- Local boost dir /root/mysql-5.7.19/boost/boost\_1\_59\_0

-- Found /root/mysql-5.7.19/boost/boost\_1\_59\_0/boost/version.hpp

-- BOOST\_VERSION\_NUMBER is #define BOOST\_VERSION 105900

-- BOOST\_INCLUDE\_DIR /root/mysql-5.7.19/boost/boost\_1\_59\_0

-- NUMA library missing or required version not available

-- WITH\_PROTOBUF=bundled

-- protobuf version is 2.6

-- You need to set WITH\_CURL. This variable needs to point to curl library.

-- Using cmake version 2.8.12.2

-- Disabling -Wunused-but-set-variable warning for building NDB

-- Disabling -Wstrict-aliasing warning for building NDB

-- Not building NDB

-- Using Boost headers from /root/mysql-5.7.19/boost/boost\_1\_59\_0

-- MYSQLX - Text log of protobuf messages enabled

-- Creating LDAP authentication SASL client library.

-- Required SASL library is missing. Skipping the LDAP SASL client authentication plugin.

-- Library mysqlclient depends on OSLIBS -lpthread;m;rt;dl

-- Googletest was not found. gtest-based unit tests will be disabled. You can run cmake . -DENABLE\_DOWNLOADS=1 to automatically download and build required components from source.

-- If you are inside a firewall, you may need to use an https proxy: export https\_proxy=http://example.com:80

CMake Warning at cmake/bison.cmake:20 (MESSAGE):

Bison executable not found in PATH

Call Stack (most recent call first):

sql/CMakeLists.txt:548 (INCLUDE)

CMake Warning at cmake/bison.cmake:20 (MESSAGE):

Bison executable not found in PATH

Call Stack (most recent call first):

libmysqld/CMakeLists.txt:187 (INCLUDE)

-- Library mysqlserver depends on OSLIBS -lpthread;m;rt;crypt;dl

-- INSTALL mysqlclient.pc lib/pkgconfig

-- Skipping deb packaging on unsupported platform .

-- CMAKE\_BUILD\_TYPE: RelWithDebInfo

-- COMPILE\_DEFINITIONS: \_GNU\_SOURCE;\_FILE\_OFFSET\_BITS=64;HAVE\_CONFIG\_H;HAVE\_LIBEVENT1

-- CMAKE\_C\_FLAGS: -Wall -Wextra -Wformat-security -Wvla -Wwrite-strings -Wdeclaration-after-statement

-- CMAKE\_CXX\_FLAGS: -Wall -Wextra -Wformat-security -Wvla -Woverloaded-virtual -Wno-unused-parameter

-- CMAKE\_C\_LINK\_FLAGS:

-- CMAKE\_CXX\_LINK\_FLAGS:

-- CMAKE\_C\_FLAGS\_RELWITHDEBINFO: -O3 -g -fabi-version=2 -fno-omit-frame-pointer -fno-strict-aliasing -DDBUG\_OFF

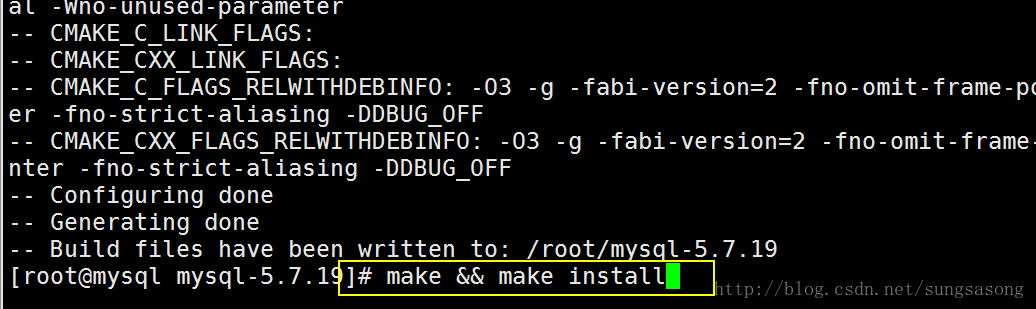
-- CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO: -O3 -g -fabi-version=2 -fno-omit-frame-pointer -fno-strict-aliasing -DDBUG\_OFF

-- Configuring done

-- Generating done

-- Build files have been written to: /root/mysql-5.7.19

[root@mysql mysql-5.7.19]#

完成！   
**5、编译安装**   
   
后续步骤一样，再不做说明，等的时间太长了，等不住了！

**最后记：**   
在企业的生产环境中，追求的是稳定性，官方的二进制版本是经过了充分的验证的，虽然使用源码安装会带来性能方面的提升，但是编码复杂，如果不清楚编译的参数，依然推荐大家使用二进制的方式进行安装。

**完成于2017 07 28**

sss