

TaiShan 服务器

Mysql 环境搭建指导书 - 中标麒麟 7.0

文档版本

01

发布日期

2020-7-28



鲲鹏创新中心旗舰店

目 录

| | |
|------------------------|----------|
| 1 软件介绍..... | 3 |
| 2 环境配置..... | 4 |
| 2.1 检查编译环境..... | 4 |
| 2.2 检查运行环境..... | 5 |
| 2.3 升级 cmake..... | 5 |
| 3 编译 mysql..... | 7 |
| 3.1 编译准备..... | 7 |
| 3.2 源码编译..... | 7 |
| 4 验证 mysql..... | 8 |
| 4.1 配置文件设置..... | 8 |
| 4.2 关闭防火墙..... | 9 |
| 4.3 mysql 部署..... | 10 |

1 软件介绍

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)之一。

2 环境配置

| 类别 | 子项 | 版本 |
|------|-------|-------|
| 硬件 | CPU | 鲲鹏920 |
| OS | 中标麒麟 | 7.0 |
| 编译环境 | Gcc | 7.3.0 |
| | Glibc | 2.27 |

gcc7.3.0 安装路径/usr/local/gcc-7.3.0；glibc2.27 安装路径/usr/local/glibc-2.27；已配置环境变量 PATH，使用 which gcc 可见使用/usr/local/gcc-7.3.0/bin/gcc，使用 which ldd 可见使用/usr/local/glibc-2.27/bin/ldd。

2.1 检查编译环境

步骤 1 查看编译器 gcc、g++版本：

```
gcc -v
[root@localhost cmake-3.5.2]# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/local/gcc-7.3.0/libexec/gcc/aarch64-linux/7.3.0/lto-wrapper
Target: aarch64-linux
Configured with: ./configure --prefix=/usr/local/gcc-7.3.0 --enable-languages=c,c++,fortran --enable-shared --enable-linker-build-id --without-included-gettext --enable-threads=posix --disable-multilib --disable-nls --disable-libsanitizer --disable-browser-plugin --enable-checking=release --build=aarch64-linux
Thread model: posix
gcc version 7.3.0 (GCC)

g++ -v
```

```
[root@localhost cmake-3.5.2]# g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/local/gcc-7.3.0/libexec/gcc/aarch64-linux/7.3.0/lto-wrapper
Target: aarch64-linux
Configured with: ./configure --prefix=/usr/local/gcc-7.3.0 --enable-languages=c,c++,fortran --enable-shared --enable-linker-build-id --without-included-gettext --enable-threads=posix --disable-multilib --disable-nls --disable-lsanitizer --disable-browser-plugin --enable-checking=release --build=aarch64-linux
Thread model: posix
gcc version 7.3.0 (GCC)
```

2.2 检查运行环境

步骤 1 查看 C++运行库版本(cmake 3.5.2 需要运行库版本 3.4.21)

```
strings /lib64/libstdc++.so.6 | grep ^GLIBCXX
[ root@localhost cmake-3.5.2]# strings /lib64/libstdc++.so.6 | grep GLIBCXX
GLIBCXX_3.4
GLIBCXX_3.4.1
GLIBCXX_3.4.2
GLIBCXX_3.4.3
GLIBCXX_3.4.4
GLIBCXX_3.4.5
GLIBCXX_3.4.6
GLIBCXX_3.4.7
GLIBCXX_3.4.8
GLIBCXX_3.4.9
GLIBCXX_3.4.10
GLIBCXX_3.4.11
GLIBCXX_3.4.12
GLIBCXX_3.4.13
GLIBCXX_3.4.14
GLIBCXX_3.4.15
GLIBCXX_3.4.16
GLIBCXX_3.4.17
GLIBCXX_3.4.18
GLIBCXX_3.4.19
GLIBCXX_DEBUG_MESSAGE_LENGTH
```

可见运行库支持到 GLIBCXX_3.4.19，需要配置使用 gcc7.3.0 自带的新运行库，才能满足 cmake 需要 (GLIBCXX_3.4.21)。

步骤 2 配置软链接到/lib64

```
ln -s -f /usr/local/gcc-7.3.0/lib64/libstdc++.so.6 /lib64/libstdc++.so.6
```

步骤 3 验证运行库版本，可见增加了对高版本 GLIBCXX 的支持。

```
GLIBCXX_3.4.17
GLIBCXX_3.4.18
GLIBCXX_3.4.19
GLIBCXX_3.4.20
GLIBCXX_3.4.21
GLIBCXX_3.4.22
GLIBCXX_3.4.23
GLIBCXX_3.4.24
```

2.3 升级 cmake

centos7.6自带cmake2.8.12.2版本过低，不满足mysql编译要求，需要升级到3.4.3版本以上。

步骤 1 检查 cmake 版本

```
cmake --version  
[root@localhost home]# cmake --version  
cmake version 2.8.12.2
```

可见 cmake 版本过低，需要升级。

步骤 2 下载高版本 cmake 源码包并编译安装：

```
wget https://cmake.org/files/v3.5/cmake-3.5.2.tar.gz  
tar -zxf cmake-3.5.2.tar.gz  
cd cmake-3.5.2  
./bootstrap && make && make install
```

步骤 3 清除系统的 hash，否则可能引用到旧版本 cmake。

```
hash -r
```

步骤 4 验证更新的 cmake 版本

```
cmake --version  
[root@localhost ~]# cmake --version  
cmake version 3.5.2  
  
CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

可见更新后的 cmake 版本为 3.5.2。

3 编译 mysql

3.1 编译准备

步骤 1 安装依赖包

```
yum install -y ncurses-devel libatomic
```

步骤 2 获取 mysql 源码并解压:

```
wget https://downloads.mysql.com/archives/get/p/23/file/mysql-boost-5.7.26.tar.gz  
tar -zxf mysql-boost-5.7.26.tar.gz
```

步骤 3 mysql 不允许在源码根目录下编译，因此创建单独的目录存放编译结果:

```
cd ./mysql-5.7.26  
mkdir build
```

3.2 源码编译

步骤 1 进入准备好的编译目录:

```
cd build
```

步骤 2 创建 Makefile:

```
cmake ../ -DCMAKE_C_COMPILER=/usr/local/gcc-7.3.0/bin/gcc -
DCMAKE_CXX_COMPILER=/usr/local/gcc-7.3.0/bin/g++ -
DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DWITH_BOOST=../boost
-- Configuring done
-- Generating done
-- Build files have been written to: /home/mysql/mysql-5.7.27/build
```

步骤 3 编译:

在工具源码目录下执行 `make -j 128`

```
[100%] Built target sql
Scanning dependencies of target pfs_connect_attr-t
Scanning dependencies of target mysqld
[100%] Building CXX object sql/CMakeFiles/mysqld.dir/main.cc.o
[100%] Linking CXX executable mysqld
[100%] Building CXX object storage/perfschema/unittest/CMakeFiles/pfs_connect_attr-t.dir/pfs_connect_attr-t.cc.o
[100%] Building C object storage/perfschema/unittest/CMakeFiles/pfs_connect_attr-t.dir/_/_/_mysys/string.c.o
[100%] Building CXX object storage/perfschema/unittest/CMakeFiles/pfs_connect_attr-t.dir/_/_/_sql/sql_builtin.cc.o
[100%] Built target mysql_embedded
[100%] Linking CXX executable pfs_connect_attr-t
[100%] Built target mysqltest_embedded
[100%] Linking CXX executable mysql_client_test_embedded
[100%] Built target mysqld
[100%] Built target pfs_connect_attr-t
[100%] Built target mysql_client_test_embedded
```

步骤 4 安装:

```
make install
```

在 `/usr/local/mysql` 下可见编译好的 mysql。

4 验证 mysql

4.1 配置文件设置

步骤 1 `/etc/my.cnf` 为 mysql 的配置文件，操作系统自带 mariadb 配置文件，修改后可以给安装的 mysql 使用，如果不存在，也可参考下文手动创建。

```
vim /etc/my.cnf
```

默认 `my.cnf` 配置文件内容如下:


```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

在 datadir 上添加 basedir=/usr/local/mysql

修改 datadir=/usr/local/mysql/datadir

修改 socket=/usr/local/mysql/mysql.sock

注释 [mysqld_safe]

修改 log-error=/usr/local/mysql/mysql.log

修改 pid-file=/usr/local/mysql/mysql.pid

添加 [client] (用于本地登录验证)

在 [client] 下添加 socket=/usr/local/mysql/mysql.sock

修改后的 my.cnf 如图

```
[mysqld]
basedir=/usr/local/mysql
datadir=/usr/local/mysql/data
socket=/usr/local/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
# symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

# [mysqld_safe]
log-error=/usr/local/mysql/mysql.log
pid-file=/usr/local/mysql/mysql.pid

#
# include all files from the config directory
#
# !includedir /etc/my.cnf.d

[client]
socket=/usr/local/mysql/mysql.sock
```

4.2 关闭防火墙

```
systemctl stop firewalld.service
```

```
systemctl disable firewalld.service
```

4.3 mysql 部署

步骤 1 为 mysql 创建独立用户和用户组：

```
groupadd mysql
useradd -g mysql mysql
```

步骤 2 创建数据目录：

```
mkdir -p /usr/local/mysql/datadir
```

步骤 3 创建临时目录：

```
mkdir -p /usr/local/tmp/
```

步骤 4 修改 mysql 文件夹拥有者：

```
chown -R mysql:mysql /usr/local/mysql
```

步骤 5 配置环境变量，在/etc/profile 文件末尾加入：

```
export PATH=/usr/local/mysql/bin:$PATH
```

步骤 6 并使环境变量配置生效：

```
source /etc/profile
```

步骤 7 初始化 mysql 数据库：

```
mysqld --initialize --user=mysql
执行后可在/usr/local/mysql/datadir 看到数据库文件。
cat mysql.log
```

```
[root@localhost etc]# cd /usr/local/mysql
[root@localhost mysql]# cat mysql.log
2020-03-16T07:08:06.561046Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2020-03-16T07:08:06.853587Z 0 [Warning] InnoDB: New log files created, LSN=45790
2020-03-16T07:08:06.898542Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2020-03-16T07:08:06.955839Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: e2e16814-6754-11ea-841b-20283eb8f99e.
2020-03-16T07:08:06.956312Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2020-03-16T07:08:06.957267Z 1 [Note] A temporary password is generated for root@localhost: !%*nfm;q2lR
```

在/usr/local/mysql/mysql.log 中可见临时密码，登录时需提供。

步骤 8 配置 mysql 服务：

```
cp support-files/mysql.server /etc/init.d/mysql.server
chkconfig mysql.server on
```

步骤 9 启动 mysql 服务进行验证：

```
service mysql.server start
```

```

[root@localhost mysql]# service mysql.server start
Starting MySQL. SUCCESS!

```

```
ps -ef | grep mysqld
```

```

[root@localhost ~]# ps -ef | grep mysqld
root      48882      1   0 15:26 pts/0    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --datadir=/usr/local/mysql/data --pid-file=/usr/local/mysql/mysq
l.pid
mysql     48236  48882   1 15:26 pts/0    00:00:00 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --plugin-dir=/usr/
local/mysql/lib/plugin --user=mysql --log-error=/usr/local/mysql/mysql.log --pid-file=/usr/local/mysql/mysql.pid --socket=/usr/local/mysql/mysql.sock
root      48572  44816   0 15:26 pts/1    00:00:00 grep --color=auto mysqld

```

可见 mysqld 和 mysqld_safe 进程运行。

步骤 10 使用初始账户和先前记录的初始密码进行登录，可正常登入数据库：

```
mysql -uroot -p[临时密码]
```

```

[root@localhost mysql]# mysql -uroot -p'!%+nifm;g2lR'
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.27

Copyright (c) 2000, 2019, 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>

```

修改初始密码：

```
ALTER USER 'root'@localhost IDENTIFIED BY 'Huawei12#$';
```

```

mysql> ALTER USER 'root'@localhost IDENTIFIED BY 'Huawei12#$' ;
Query OK, 0 rows affected (0.00 sec)

```

```
quit;
```

退出 mysql。

尝试使用新密码登录：

```

[root@localhost mysql]# mysql -uroot -p'Huawei12#$'
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.27 Source distribution

Copyright (c) 2000, 2019, 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.

```

可见使用新密码可以正常登录数据库。

```
shutdown;
```

关闭数据库。

```
exit;
```

退出 mysql。

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

mysql> exit;
Bye
```

```
ps -ef | grep mysqld
```

可见不再有 mysql 进程。

```
[root@localhost ~]# ps -ef | grep mysqld
root      55563  44816  0 15:32 pts/1    00:00:00 grep --color=auto mysqld
```

5 FAQ

5.1 internal compiler error: in final_scan_insn

```
/home/mysql-5.7.27/storage/innobase/srv/srv0srv.cc:2875:1: internal compiler error: in final_scan_insn, at final.c:2897
Please submit a full bug report,
with preprocessed source if appropriate.
See <http://bugzilla.redhat.com/bugzilla> for instructions.
/home/mysql-5.7.27/storage/innobase/row/row0mysql.cc: In function 'dberr_t row_drop_table_for_mysql(const char*, trx_t*, bool, bool, dict_table_t*)':
/home/mysql-5.7.27/storage/innobase/row/row0mysql.cc:4910:1: error: could not split insn
```

编译使用的gcc、g++版本过低，参考3.4 cmake编译配置进行设置，使用新版的gcc。

5.2 GLIBCXX_3.4.21 not found

```
[root@localhost mysql]# mysqld --user=mysql
mysqld: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by mysqld)
mysqld: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.20' not found (required by mysqld)
```

系统C++运行环境过低，需要更新/lib64/libstdc++.so.6软链接到新版本运行环境，参考2.2 检查运行环境。

5.3 libatomic.so.1: Cannot open shared object file

```
[root@localhost mysql]# service mysql.server start
/usr/local/mysql/bin/my_print_defaults: error while loading shared libraries: libatomic.so.1: cannot open shared object file: No such file or directory
Starting MySQL: /usr/local/mysql/bin/my_print_defaults: error while loading shared libraries: libatomic.so.1: cannot open shared object file: No such file or directory
/usr/local/mysql/bin/my_print_defaults: error while loading shared libraries: libatomic.so.1: cannot open shared object file: No such file or directory
Logging to '/usr/local/mysql/data/localhost.localdomain.err'
ERROR! The server quit without updating PID file (/usr/local/mysql/data/localhost.localdomain.pid).
```

系统缺少依赖包libatomic，参考4.1 编译准备进行安装。

5.4 Can't connect to local MySQL server

```
[root@localhost build]# /usr/local/mysql/bin/mysql -uroot -pHuawei12#$
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)
```

检查mysqld是否正在运行，如果不在运行，需要先启动mysql服务service mysql.server start。如果mysqld正常运行，参考5.1 配置文件设置，配置[client]部分。

5.5 error: 'prctl' was not declared in this scope

```
[100%] Building CXX object sql/CMakeFiles/sql.dir/mysqld_thd_manager.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/protocol_callback.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/signal_handler.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/sql_audit.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/sql_client.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/srv_session.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/srv_session_info_service.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/srv_session_service.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/auth/sha2_password_common.cc.o
[100%] Building CXX object sql/CMakeFiles/sql.dir/mysqld_daemon.cc.o
/home/mysql-5.7.28/sql/mysqld.cc: In function 'passwd* check_user(const char*)':
/home/mysql-5.7.28/sql/mysqld.cc:1565:12: error: 'prctl' was not declared in this scope
    (void) prctl(PR_SET_DUMPABLE, 1);
           ^~~~~
Scanning dependencies of target mysql_embedded
[100%] Building CXX object libmysqld/examples/CMakeFiles/mysql_embedded.dir/__/client/completion_hash.cc.o
Scanning dependencies of target mysql_client_test_embedded
[100%] Building C object libmysqld/examples/CMakeFiles/mysql_client_test_embedded.dir/__/testclients/mysq
[100%] Building CXX object libmysqld/examples/CMakeFiles/mysql_embedded.dir/__/client/mysql.cc.o
make[2]: *** [sql/CMakeFiles/sql.dir/mysqld.cc.o] Error 1
make[2]: *** Waiting for unfinished jobs....
[100%] Building CXX object libmysqld/examples/CMakeFiles/mysql_embedded.dir/__/client/readline.cc.o
[100%] Built target mysqltest_embedded
[100%] Linking CXX executable mysql_embedded
[100%] Built target mysql_embedded
[100%] Linking CXX executable mysql_client_test_embedded
[100%] Built target mysql_client_test_embedded
make[1]: *** [sql/CMakeFiles/sql.dir/all] Error 2
make: *** [all] Error 2
[root@ecs-gz-mysql5 mysql-5.7.28]#
```

这是mysql的官方bug，未在测试时出现，但有复现可能。

vi /home/mysql-5.7.26/sql/mysqld.cc

在文件首行添加以下内容

```
#include<sys/prctl.h>
```

官网bug说明

<https://bugs.mysql.com/bug.php?id=9064>