# Part 1 MySQL源码安装

## 1.安装依赖包

yum -y install gcc-c++ ncurses-devel cmake make perl gcc autoconf automake zlib libxml libgcrypt libtool bison

## 2.安装boost库

wget http://nchc.dl.sourceforge.net/project/boost/boost/1.59.0/boost\_1\_59\_0.tar.gz

tar zxvf boost\_1\_59\_0.tar.gz

mv boost\_1\_59\_0 /usr/local/boost

## 3.添加MySQL用户

useradd mysql

## 4.下载mysql源码包并安装

tar -zvxf mysql-5.7.19.tar.gz

cd mysql-5.7.19

cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DMYSQL\_DATADIR=/usr/local/mysql/data -DDEFAULT\_CHARSET=utf8mb4 -DDEFAULT\_COLLATION=utf8mb4\_general\_ci -DMYSQL\_TCP\_PORT=3307 -DMYSQL\_USER=mysql -DENABLED\_LOCAL\_INFILE=1 -DEXTRA\_CHARSETS=all -DWITH\_PARTITION\_STORAGE\_ENGINE=1 -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_BOOST=/usr/local/boost

参数说明：

-DCMAKE\_INSTALL\_PREFI：安装路径

-DMYSQL\_DATADIR：数据路径

-DDEFAULT\_CHARSET：数据库默认字符编码

-DDEFAULT\_COLLATION：默认排序规则

-DENABLED\_LOCAL\_INFILE：允许从本文件导入数据

-DEXTRA\_CHARSETS：安装所有字符集

-DWITH\_INNOBASE\_STORAGE\_ENGINE：安装innodb存储引擎

-DWITH\_PARTITION\_STORAGE\_ENGINE：安装分区存储引擎，支持分区表

-DWITH\_BOOST：boost源码路径

make -j `grep processor /proc/cpuinfo | wc -l`

make install

## 5.初始化数据库实例

/usr/local/mysql/bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql/3307 --datadir=/usr/local/mysql/3307/data

注：初始化后会生成默认密码，请记录下来

2017-09-13T06:01:51.212773Z 1 [Note] A temporary password is generated for root@localhost: je(Ilr<&Y3r&

## 6.创建配置文件

cd /usr/local/mysql/3307

[root@iZuf6h90t5awxwr3uqp6f0Z 3307]# vi my.cnf

[client]

port=3307

socket=/usr/local/mysql/3307/mysql.sock

default-character-set = utf8mb4

[mysql]

default-character-set = utf8mb4

[mysqld]

port=3307

socket=/usr/local/mysql/3307/mysql.sock

character-set-client-handshake = FALSE

character-set-server=utf8mb4

collation-server=utf8mb4\_general\_ci

init\_connect='SET NAMES utf8mb4'

skip-external-locking

skip-name-resolve

user=mysql

basedir=/usr/local/mysql/3307

datadir=/usr/local/mysql/3307/data

log-error=/usr/local/mysql/3307/logs/mysql\_error.log

pid-file=/usr/local/mysql/3307/mysql.pid

open\_files\_limit=10240

max\_connections=1024

max\_allowed\_packet=128M

lower\_case\_table\_names=1

default-storage-engine=INNODB

innodb\_file\_per\_table=1

long\_query\_time=2

slow-query-log=on

slow-query-log-file=/usr/local/mysql/3307/logs/mysql\_slow.log

[mysqld\_safe]

log-error=/usr/local/mysql/3307/logs/mysqld.log

pid-file=/usr/local/mysql/3307/mysqld.pid

如果logs不存在，要先创建出来

增加mysql用户的权限：

cd /usr/local/mysql/3307

chown -R mysql .

## 7.启动数据库

/usr/local/mysql/bin/mysqld\_safe --defaults-file=/usr/local/mysql/3307/my.cnf &

## 8.访问数据库

[root@localhost mysql]# mysql -uroot -p -P3307 -S/usr/local/mysql/3307/mysql.sock

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 3

Server version: 5.7.19-log

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> alter user 'root'@'localhost' identified by '123456';

# Part 2 MySQL主从配置

## 安装3307数据库，参考Part1

## 安装3308数据库，参考Part1-5，6，7，8

## 修改my.cnf

在3307主服务器上，添加，然后重启：

[mysqld]

log-bin=/usr/local/mysql/3307/mysql-bin #启动二进制文件

server-id=1 #服务器ID

binlog-do-db = test #同步的数据库

binlog-ignore-db = mysql #忽略数据库

在3308从服务器上添加，然后重启：

[mysqld]

log-bin=/usr/local/mysql/3308/mysql-bin #启动二进制文件

server-id=10 #服务器ID

binlog-do-db = test #同步的数据库

binlog-ignore-db = mysql #忽略数据库

## 在3307主服务器中添加一个backup的账号，并授权给从服务器

create user 'backup'@'127.0.0.1' identified by 'backup';

grant replication slave on \*.\* to 'backup'@'127.0.0.1' identified by 'backup';

## 查询主数据库状态，并记下FILE及Position的值

mysql> show master status;

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

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

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

| mysql-bin.000002 | 702 | | | |

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

1 row in set (0.00 sec)

## 在3308从服务器上设置主服务器

mysql> change master to master\_host='127.0.0.1',master\_user='backup',master\_port=3307,master\_password='backup',master\_log\_file='mysql-bin.000002',master\_log\_pos=702;

Query OK, 0 rows affected, 2 warnings (0.01 sec)

启动slave同步

mysql> start slave;

Query OK, 0 rows affected (0.00 sec)

## 检查主从的状态

如果您看到Slave\_IO\_Running和Slave\_SQL\_Running均为Yes，则主从复制连接正常。

mysql> show slave status\G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 127.0.0.1

Master\_User: backup

Master\_Port: 3307

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000002

Read\_Master\_Log\_Pos: 702

Relay\_Log\_File: localhost-relay-bin.000002

Relay\_Log\_Pos: 320

Relay\_Master\_Log\_File: mysql-bin.000002

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB:

Replicate\_Ignore\_DB:

Replicate\_Do\_Table:

Replicate\_Ignore\_Table:

Replicate\_Wild\_Do\_Table:

Replicate\_Wild\_Ignore\_Table:

Last\_Errno: 0

Last\_Error:

Skip\_Counter: 0

Exec\_Master\_Log\_Pos: 702

Relay\_Log\_Space: 531

Until\_Condition: None

Until\_Log\_File:

Until\_Log\_Pos: 0

Master\_SSL\_Allowed: No

Master\_SSL\_CA\_File:

Master\_SSL\_CA\_Path:

Master\_SSL\_Cert:

Master\_SSL\_Cipher:

Master\_SSL\_Key:

Seconds\_Behind\_Master: 0

Master\_SSL\_Verify\_Server\_Cert: No

Last\_IO\_Errno: 0

Last\_IO\_Error:

Last\_SQL\_Errno: 0

Last\_SQL\_Error:

Replicate\_Ignore\_Server\_Ids:

Master\_Server\_Id: 1

Master\_UUID: 092d9a9c-9849-11e7-a302-00505680267e

Master\_Info\_File: /usr/local/mysql/3308/data/master.info

SQL\_Delay: 0

SQL\_Remaining\_Delay: NULL

Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for more updates

Master\_Retry\_Count: 86400

Master\_Bind:

Last\_IO\_Error\_Timestamp:

Last\_SQL\_Error\_Timestamp:

Master\_SSL\_Crl:

Master\_SSL\_Crlpath:

Retrieved\_Gtid\_Set:

Executed\_Gtid\_Set:

Auto\_Position: 0

Replicate\_Rewrite\_DB:

Channel\_Name:

Master\_TLS\_Version:

1 row in set (0.00 sec)

# Part 3 MySQL双主配置

## 1.安装3309，3310数据库，3310为主，3310为从。

现在想把3307和3309做一个双主同步

## 2.修改my.cnf，重启

在3307上：

#主－主形式需要多添加的部分

log-slave-updates

sync\_binlog = 1

auto\_increment\_offset = 1

auto\_increment\_increment = 2

replicate-do-db = test

replicate-ignore-db = mysql

在3309上：

#主－主形式需要多添加的部分

log-slave-updates

sync\_binlog = 1

auto\_increment\_offset = 2

auto\_increment\_increment = 2

replicate-do-db = test

replicate-ignore-db = mysql

## 3.查看配置情况

在3307上：

mysql> show master status \G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: mysql-bin.000004

Position: 154

Binlog\_Do\_DB: test

Binlog\_Ignore\_DB: mysql

Executed\_Gtid\_Set:

1 row in set (0.00 sec)

在3309上：

mysql> show master status \G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: mysql-bin.000004

Position: 154

Binlog\_Do\_DB: test

Binlog\_Ignore\_DB: mysql

Executed\_Gtid\_Set:

1 row in set (0.00 sec)

## 4.设置同步

3307上：

stop slave;

change master to master\_host='127.0.01',master\_port=3309,master\_user='backup', master\_password='backup', master\_log\_file='mysql-bin.000004', master\_log\_pos=154;

//修改当前的Master的值,因为是互为备份，所以3307的master为3309，3309设置复制的用户名为backup,密码是backup。

start slave; #开始salve，开始同步

show slave status; #检测slave状态,如果Slave\_IO\_Running: Yes Slave\_SQL\_Running: Yes Seconds\_Behind\_Master: 0就OK了

3309上：

stop slave;

change master to master\_host='127.0.01',master\_port=3307,master\_user='backup', master\_password='backup', master\_log\_file='mysql-bin.000004', master\_log\_pos=154;

start slave;

# 参考

<http://blog.csdn.net/goldenfish1919/article/details/50251225>

<http://blog.csdn.net/xuejiazhi/article/details/8941156>

<http://blog.csdn.net/qq394829044/article/details/53203645>