

1.工具使用： canal从入门到实战

1.安装前准备

1.1 操作系统(cenos7.9 x86)

1.2 工具版本选择

2 安装环境搭建

2.1 MySQL安装

- (1) 检查是否已经安装过mysql, 执行命令
- (2) 安装mysql
- (3) 启动mysql服务器
- (4) 开放远程连接, 测试应用

2.1 canal安装

- (1) git下载canal,选择最新release版本 1.1.5
- (2) 数据库配置
- (3) canal的工作原理
- (4) canal配置
- (5) 启动canal, 测试
- (6) canal client测试

1.安装前准备

1.1 操作系统(cenos7.9 x86)

```
1  #查看
2  /etc/redhat-release文件
3  cat /etc/redhat-releaseCentOS Linux release 7.9.2009 (Core)
4  #查看/proc/version文件
5  # proc 为process的缩写，里面存放与内核相关的文件
6  cat /proc/version
7  Linux version 3.10.0-1160.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org)
   (gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC) )#1 SMP Mon Oct 19
   16:18:59 UTC 2020
8  #使用uname -a 命令
9  uname -a
10 Linux localhost.localdomain 3.10.0-1160.el7.x86_64 #1 SMP Mon Oct 19
    16:18:59 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

1.2 工具版本选择

mysql: MySQL-5.7

rocketMQ:rocketmq-all-4.9.3

canal版本:canal.deployer-1.1.5.tar.gz

2 安装环境搭建

2.1 MySQL安装

参考文档: <https://zhuanlan.zhihu.com/p/87069388>

(1) 检查是否已经安装过mysql，执行命令

```
1  [root@localhost software]# rpm -qa | grep mysql[root@localhost software]#
```

从结果来看，当前机器没有安装mysql。可以直接安装MySQL，如果需要卸载旧版，操作如下：

Shell | 复制代码

```
1 [root@localhost /]#rpm -e --nodeps mysql-libs-5.1.73-5.el6_6.x86_64
2 #再次执行查询命令，查看是否删除
3 [root@localhost /]# rpm -qa | grep mysql
4 #查询所有Mysql对应的文件夹
5 [root@localhost /]
6 # whereis mysqlmysql:
7 /usr/bin/mysql
8 /usr/include/mysql
9 [root@localhost lib]
10 # find / -name mysql/data/mysql/data/mysql/mysql
11 #删除相关目录或文件[root@localhost /]
12 # rm -rf /usr/bin/mysql /usr/include/mysql /data/mysql /data/mysql/mysql
13 #验证是否删除完毕
14 [root@localhost /]# whereis mysql
15 mysql:
16 [root@localhost /]# find / -name mysql
17 [root@localhost /]#
```

(2) 安装mysql

从官网下载用于Linux的Mysql安装包

```
1  wget https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.24-linux-
   glibc2.12-x86_64.tar.gz
2  --2022-04-25 23:11:17--
3  https://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.24-linux-
   glibc2.12-x86_64.tar.gzResolving dev.mysql.com (dev.mysql.com)...
   137.254.60.11Connecting to dev.mysql.com
   (dev.mysql.com)|137.254.60.11|:443...
4  connected.HTTP request sent, awaiting response... 302 FoundLocation:
   https://cdn.mysql.com//archives/mysql-5.7/mysql-5.7.24-linux-glibc2.12-
   x86_64.tar.gz [following]--2022-04-25 23:11:18--
5  https://cdn.mysql.com//archives/mysql-5.7/mysql-5.7.24-linux-glibc2.12-
   x86_64.tar.gzResolving cdn.mysql.com (cdn.mysql.com)...
6  23.56.1.19Connecting to cdn.mysql.com (cdn.mysql.com)|23.56.1.19|:443...
7  connected.HTTP request sent, awaiting response... 200 OKLength: 644930593
   (615M) [application/x-tar-gz]Saving to: 'mysql-5.7.24-linux-glibc2.12-
   x86_64.tar.gz'100%[=====] in
   6m0s2022-04-25 23:17:19 (1.71 MB/s)
8  - 'mysql-5.7.24-linux-glibc2.12-x86_64.tar.gz' saved
   [644930593/644930593]
```

解压，赋权限，切换用户到anchu

```
1  #解压当前目录
2  [root@localhost software]# pwd/home/anchu/software
3  [root@localhost software]# ls canal
4  canal.deployer-1.1.5.tar.gz  canal.example-1.1.5.tar.gz  mysql-5.7.24-
linux-glibc2.12-x86_64.tar.gz
5  [root@localhost software]# tar xzvf mysql-5.7.24-linux-glibc2.12-
x86_64.tar.gz
6  #赋权给anchu
7  [root@localhost software]# chown -R anchu:anchu mysql-5.7.24-linux-
glibc2.12-x86_64
8  #切换到anchu
9  [root@localhost software]# su anchu
10 #重命名目录为 mysql-5.7.24
11 [anchu@localhost software]$ mv mysql-5.7.24-linux-glibc2.12-x86_64 mysql-
5.7.24
12 [anchu@localhost software]$
13 [anchu@localhost software]$ ls canal
14 canal.deployer-1.1.5.tar.gz  canal.example-1.1.5.tar.gz  mysql-5.7.24
mysql-5.7.24-linux-glibc2.12-x86_64.tar.gz
15 #设置mysql环境变量
16 [anchu@localhost software]$ cd mysql-5.7.24/
17 [anchu@localhost mysql-5.7.24]$ cd bin
18 [anchu@localhost bin]$ pwd/home/anchu/software/mysql-5.7.24/bin
19 [anchu@localhost bin]$ vi ~/.bash_profile
20 [anchu@localhost bin]$ ls
21 innochecksum      myisam_ftdump  my_print_defaults  mysqlbinlog
      mysql_config      mysqld-debug  mysqldump          mysqlimport
mysqlpump          mysqlslap      mysql_tzinfo_to_sql
pererror          resolve_stack_dump  lz4_decompress  myisamlog      mysql
      mysqlcheck      mysql_config_editor  mysqld_multi
mysqldumpslow  mysql_install_db  mysql_secure_installation
mysql_ssl_rsa_setup  mysql_upgrade      replace
zlib_decompressmyisamchk      myisampack      mysqladmin
mysql_client_test_embedded  mysqld          mysqld_safe
mysql_embedded  mysql_plugin      mysqlshow
mysqltest_embedded  mysqlxtest      resolveip
22 [anchu@localhost bin]$ pwd/home/anchu/software/mysql-5.7.24/bin
23 [anchu@localhost bin]$ echo 'MYSQL_HOME=/home/anchu/software/mysql-
5.7.24/' >> ~/.bash_profile
24 [anchu@localhost bin]$ echo 'PATH=$PATH:$MYSQL_HOME/bin'
>> ~/.bash_profile
25 [anchu@localhost bin]$ echo 'export PATH' >> ~/.bash_profile
26 [anchu@localhost bin]$ source ~/.bash_profile
27 [anchu@localhost bin]$ mysql --versionmysql
```

28 Ver 14.14 Distrib 5.7.24, for linux-glibc2.12 (x86_64) using EditLine wrapper

创建数据目录初始化mysql,务必记住初始化输出日志末尾的密码（数据库管理员临时密码）

```
▼ Shell | 复制代码
1 [anchu@localhost mysql-5.7.24]$ pwd/home/anchu/software/mysql-5.7.24
2 [anchu@localhost mysql-5.7.24]$ mkdir data
3 [anchu@localhost mysql-5.7.24]$ cd data
4 [anchu@localhost data]$ pwd/home/anchu/software/mysql-5.7.24/data
5
6 [anchu@localhost mysql-5.7.24]$ mysqld --initialize --user=anchu --
  datadir=/home/anchu/software/mysql-5.7.24/data --
  basedir=/home/anchu/software/mysql-5.7.24/
7 2022-04-26T07:26:58.470872Z 0
8 [Warning] Changed limits: max_open_files: 1024 (requested 5000)2022-04-
  26T07:26:58.471169Z 0 [Warning] Changed limits: table_open_cache: 431
  (requested 2000)2022-04-26T07:26:58.471516Z 0 [Warning] TIMESTAMP with
  implicit DEFAULT value is deprecated. Please use --
  explicit_defaults_for_timestamp server option (see documentation for more
  details).2022-04-26T07:26:59.096129Z 0 [Warning] InnoDB: New log files
  created, LSN=457902022-04-26T07:26:59.272923Z 0 [Warning] InnoDB:
  Creating foreign key constraint system tables.2022-04-26T07:26:59.346943Z
  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:
  4253d31b-c532-11ec-9f60-000c29924945.2022-04-26T07:26:59.358828Z 0
  [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed'
  cannot be opened.2022-04-26T07:26:59.381263Z 1 [Note] A temporary
  password is generated for root@localhost: )wR#VWfgd0O<
```

由上可知：密码为)wR#VWfgd0O<

编辑配置文件my.cnf，添加配置如下：

```
1 [anchu@localhost bin]$ su root
2 [root@localhost bin]# vi /etc/my.cnf
3 [mysqld]
4 #datadir=/var/lib/mysql
5 #socket=/var/lib/mysql/mysql.sockdata
6 dir=/home/anchu/software/mysql-5.7.24/
7 datasocket=/home/anchu/software/mysql-5.7.24/data/mysql.sock
8 port=3306
9 sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
10 max_connections=400
11 innodb_file_per_table=1#表名大小写不明感, 敏感为1
12 #lower_case_table_names=0 表名存储为给定的大小和比较是区分大小写的
13 #lower_case_table_names = 1 表名存储在磁盘是小写的, 但是比较的时候是不区分大小写
14 #lower_case_table_names=2 表名存储为给定的大小写但是比较的时候是小写的
15 #unix,linux下lower_case_table_names默认值为 0 .Windows下默认值是 1 .Mac OS X
16 下默认值是 2
17 lower_case_table_names=1
18 # Disabling symbolic-links is recommended to prevent assorted security
19 riskssymbolic-links=0
20 # Settings user and group are ignored when systemd is used.# If you need
21 to run mysqld under a different user or group,
22 # customize your systemd unit file for mariadb according to the
23 # instructions in http://fedoraproject.org/wiki/Systemd
24 [mysqld_safe]
25 log-error=/home/anchu/software/mysql-5.7.24/data/mariadb.log
26 pid-file=/home/anchu/software/mysql-5.7.24/data/mariadb.pid
27 ## include all files from the config directory#!includedir /etc/my.cnf.d
28 [root@localhost bin]# cp /etc/my.cnf /home/anchu/software/mysql-5.7.24/
29 [root@localhost bin]# chown anchu:anchu /home/anchu/software/mysql-
30 5.7.24/my.cnf
```

(3) 启动mysql服务器

```
1 [anchu@localhost support-files]$ su anchuPassword:
2 [anchu@localhost support-files]$ pwd/home/anchu/software/mysql-
5.7.24/support-files
3 [anchu@localhost support-files]$ mysql.server startbash: mysql.server:
command not found...
4 [anchu@localhost support-files]$ lsmagic mysqld_multi.server mysql-log-
rotate mysql.server
5 [anchu@localhost support-files]$ ./mysql.server start
6 ./mysql.server: line 239: my_print_defaults: command not
found./mysql.server: line 259: cd: /usr/local/mysql: No such file or
directoryStarting MySQL ERROR! Couldn't find MySQL server
(/usr/local/mysql/bin/mysqld_safe)
7 #通过mysqld_safe启动
8 [anchu@localhost mysql-5.7.24]$ /bin/sh /home/anchu/software/mysql-
5.7.24/bin/mysqld_safe --defaults-file=/home/anchu/software/mysql-
5.7.24/my.cnf 2>&1 > /dev/null &
9 [1] 81651
10 #查看进程
11 [anchu@localhost mysql-5.7.24]$ ps -ef|grep mysql
12 anchu      81651  80040  0 19:44 pts/1    00:00:00 /bin/sh
/home/anchu/software/mysql-5.7.24/bin/mysqld_safe --defaults-
file=/home/anchu/software/mysql-5.7.24/my.cnf
13 anchu      81841  81651 11 19:44 pts/1    00:00:00
/home/anchu/software/mysql-5.7.24/bin/mysqld --defaults-
file=/home/anchu/software/mysql-5.7.24/my.cnf --
basedir=/home/anchu/software/mysql-5.7.24 --
datadir=/home/anchu/software/mysql-5.7.24/data --plugin-
dir=/home/anchu/software/mysql-5.7.24/lib/plugin --log-
error=/home/anchu/software/mysql-5.7.24/data/mariadb.log --pid-
file=/home/anchu/software/mysql-5.7.24/data/mariadb.pid --
socket=/home/anchu/software/mysql-5.7.24/data/mysql.sock --port=3306
14
15 [anchu@localhost support-files]$ ps -ef|grep mysqld
16 anchu      81651  80040  0 19:44 pts/1    00:00:00 /bin/sh
/home/anchu/software/mysql-5.7.24/bin/mysqld_safe --defaults-
file=/home/anchu/software/mysql-5.7.24/my.cnf
17 anchu      81841  81651  0 19:44 pts/1    00:00:00
/home/anchu/software/mysql-5.7.24/bin/mysqld --defaults-
file=/home/anchu/software/mysql-5.7.24/my.cnf --
basedir=/home/anchu/software/mysql-5.7.24 --
datadir=/home/anchu/software/mysql-5.7.24/data --plugin-
dir=/home/anchu/software/mysql-5.7.24/lib/plugin --log-
error=/home/anchu/software/mysql-5.7.24/data/mariadb.log --pid-
file=/home/anchu/software/mysql-5.7.24/data/mariadb.pid --
socket=/home/anchu/software/mysql-5.7.24/data/mysql.sock --port=3306
```


登录mysql，修改密码(密码为步骤5生成的临时密码) 123456

▼ Shell | 复制代码

```
1 [anchu@localhost support-files]$ mysql -u root -P 3306 -h 127.0.0.1 -p
2 Enter password: )wR#VWfgd00
3 <Welcome to the MySQL monitor.  Commands end with ; or \g. Your MySQL
  connection id is 13 Server version: 5.7.24 Copyright (c) 2000, 2018, 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.
4 mysql> set password for root@localhost = password('123456');
5 Query OK, 0 rows affected, 1 warning (0.03 sec)
6 mysql> exit
7 #重新登陆验证密码
8 [anchu@localhost support-files]$ mysql -u root -P 3306 -h 127.0.0.1 -
  pEnter password: 123456
```

(4) 开放远程连接，测试应用

```
1  mysql>use mysql;^C
2  mysql> use anchu;
3  ERROR 1049 (42000): Unknown database 'anchu'
4  mysql> use mysql;
5  Reading table information for completion of table and column namesYou can
   turn off this feature to get a quicker startup with -ADatabase changed
6
7  mysql> update user set user.Host='%' where user.User='root';
8  Query OK, 1 row affected (0.03 sec)Rows matched: 1  Changed: 1  Warnings:
   0
9  mysql> flush privileges;
10 Query OK, 0 rows affected (0.00 sec)
11
12 mysql> exit
13
14 #测试
15
16 [anchu@localhost support-files]$ mysql -u root -P 3306 -h 192.168.120.110
   -p
17 Enter password:
18
19 Welcome to the MySQL monitor.  Commands end with ; or \g.Your MySQL
   connection id is 17Server version: 5.7.24 MySQL Community Server
   (GPL)Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights
   reserved.Oracle is a registered trademark of Oracle Corporation and/or
   itsaffiliates. Other names may be trademarks of their respectiveowners.
20 Type 'help;' or '\h' for help. Type '\c' to clear the current input
   statement.
21
22 mysql> show tables;
23 ERROR 1046 (3D000): No database selected
24
25 mysql> create database test;
26 Query OK, 1 row affected (0.00 sec)
27
28 mysql> connect test;
29 Connection id: 20
30 Current database: test
31
32 mysql> show tables;
33 Empty set (0.01 sec)
34 mysql> create table test(id int(10),name varchar(10));
35 Query OK, 0 rows affected (0.17 sec)
36
37 mysql> show tables;
```

```
38 +-----+ | Tables_in_test |
39 +-----+ | test          | +
40 -----+1 row in set (0.00 sec)
```

至此，mysql安装结束

2.1 canal安装




(1) git下载canal,选择最新release版本 1.1.5

参考文档：

https://blog.csdn.net/A_nonym/article/details/120299939

https://blog.csdn.net/weixin_44224292/article/details/103181040

下载地址：<https://github.com/alibaba/canal/releases>

 canal.admin-1.1.5.tar.gz	36.6 MB
 canal.deployer-1.1.5.tar.gz	57.4 MB
 canal.example-1.1.5.tar.gz	22.1 MB

上传到linux虚拟机，查看解压

```
▼ Shell | 复制代码
1 [anchu@localhost software]$ pwd/home/anchu/software[anchu@localhost
  software]$mkdir canal[anchu@localhost software]$ ls canal canal.example-
  1.1.5.tar.gz mysql-5.7.24-linux-glibc2.12-x86_64.tar.gzcanal.deployer-
  1.1.5.tar.gz mysql-5.7.24
2 [anchu@localhost software]$cd canal
3 [anchu@localhost software]tar xzvf ../canal.deployer-
  1.1.5.tar.gz[anchu@localhost software]tar xzvf ../canal.example-
  1.1.5.tar.gz[anchu@localhost software]tar xzvf ../canal.admin-
  1.1.5.tar.gz[anchu@localhost canal]$ ls bin conf lib logs plugin
```

(2) 数据库配置

查看当前配置，可以看到binlog未开启，需要修改配置开启binlog

Plain Text | 复制代码

```
1 [anchu@localhost support-files]$ mysql -u root -P 3306 -h 192.168.120.110
-D test -pEnter password: Reading table information for completion of
table and column namesYou can turn off this feature to get a quicker
startup with -AWelcome to the MySQL monitor.  Commands end with ; or
\g.Your MySQL connection id is 24Server version: 5.7.24 MySQL Community
Server (GPL)Copyright (c) 2000, 2018, Oracle and/or its affiliates. All
rights reserved.Oracle is a registered trademark of Oracle Corporation
and/or itsaffiliates. Other names may be trademarks of their
respectiveowners.Type 'help;' or '\h' for help. Type '\c' to clear the
current input statement.mysql> show variables like '%log_bin%';+-----
-----+-----+| Variable_name |
Value |+-----+-----+| log_bin
      | OFF   || log_bin_basename |
log_bin_index |      ||
log_bin_trust_function_creators | OFF   || log_bin_use_v1_row_events
      | OFF   || sql_log_bin      | ON    |+-----
-----+-----+6 rows in set (0.03 sec)mysql>
```

修改需要被同步的数据库 /home/anchu/software/mysql-5.7.24/my.cfg配置，有则修改无则添加

Shell | 复制代码

```
1 #启动时，使用的copy到/home/anchu/software/mysql-5.7.24的my.cfg
2 vi /home/anchu/software/mysql-5.7.24/my.cfg
3 [mysqld]
4 log-bin=mysql-bin # 开启 binlog
5 binlog-format=ROW # 选择 ROW 模式
6 server_id=1 # 配置 MySQL replaction 需要定义，不要和 canal 的 slaveId 重复
7 binlog-rows-query-log-events = 1 #查看完整的sql语句
```

重启mysql，使配置生效

```
1 [anchu@localhost support-files]$ jobs[1]
2 + Running /bin/sh /home/anchu/software/mysql-
  5.7.24/bin/mysqld_safe --defaults-file=/home/anchu/software/mysql-
  5.7.24/my.cnf 2>&1 > /dev/null &
3 [anchu@localhost support-files]$
4 [anchu@localhost support-files]$ kill -9 %1
5 [anchu@localhost support-files]$ /home/anchu/software/mysql-
  5.7.24/bin/mysqld_safe --defaults-file=/home/anchu/software/mysql-
  5.7.24/my.cnf 2>&1 > /dev/null
6 &[1] 82914
7 [anchu@localhost support-files]$ ps -ef |grep mysql
8
9 anchu      82914  80040  0 20:47 pts/1    00:00:00 /bin/sh
  /home/anchu/software/mysql-5.7.24/bin/mysqld_safe --defaults-
  file=/home/anchu/software/mysql-5.7.24/my.cnf
10
11 anchu      83155  82914  3 20:47 pts/1    00:00:00
  /home/anchu/software/mysql-5.7.24/bin/mysqld --defaults-
  file=/home/anchu/software/mysql-5.7.24/my.cnf --
  basedir=/home/anchu/software/mysql-5.7.24 --
  datadir=/home/anchu/software/mysql-5.7.24/data --plugin-
  dir=/home/anchu/software/mysql-5.7.24/lib/plugin --log-
  error=/home/anchu/software/mysql-5.7.24/data/mariadb.log --pid-
  file=/home/anchu/software/mysql-5.7.24/data/mariadb.pid --
  socket=/home/anchu/software/mysql-5.7.24/data/mysql.sock --port=3306
```

重新连接mysql查看配置

```

1  [anchu@localhost support-files]$ mysql -u root -P 3306 -h 192.168.120.110
   -D test -p
2  Enter password:
3  Reading table information for completion of table and column namesYou can
   turn off this feature to get a quicker startup with -AWelcome to the
   MySQL monitor.
4  Commands end with ; or \g.
5  Your MySQL connection id is 2Server version: 5.7.24-log
6  MySQL Community Server (GPL)Copyright (c) 2000, 2018, Oracle and/or its
   affiliates. All rights reserved.Oracle is a registered trademark of
   Oracle Corporation and/or itsaffiliates.
7  Other names may be trademarks of their respectiveowners.
8  Type 'help;' or '\h' for help. Type '\c' to clear the current input
   statement.
9
10 mysql> show tables;
11 +-----+ | Tables_in_test |
12 +-----+ | test           |
13 +-----+1 row in set (0.00 sec)
14 mysql> show variables like '%log_bin%';
15 +-----+-----+
   | Variable_name | Value |
   +-----+-----+
16 | log_bin       | ON   |
17 | log_bin_basename | /home/anchu/software/mysql-
   5.7.24/data/mysql-bin |
18 | log_bin_index  | /home/anchu/software/mysql-
   5.7.24/data/mysql-bin.index |
19 | log_bin_trust_function_creators | OFF |
20 | log_bin_use_v1_row_events | OFF |
   | sql_log_bin   | ON   |
   +-----+-----+
   +6 rows in set (0.01
   sec)mysql>

```

查看binlog日志目录及文件

```
1 [anchu@localhost support-files]$ ll /home/anchu/software/mysql-  
5.7.24/data/*bin*  
2  
3 -rw-r-----. 1 anchu anchu 1431 Apr 26 23:23 /home/anchu/software/mysql-  
5.7.24/data/mysql-bin.000001  
4 -rw-r-----. 1 anchu anchu 19 Apr 26 20:47 /home/anchu/software/mysql-  
5.7.24/data/mysql-bin.index
```

创建一个有相关权限的mysql slave账号，用户名canal,密码canal

```
1 # 创建账号  
2 CREATE USER canal IDENTIFIED WITH MYSQL_NATIVE_PASSWORD BY 'canal';  
3 # 给账号赋权限  
4 GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO  
   'canal'@'%';  
5 # 刷新  
6 FLUSH PRIVILEGES;
```

创建和测试canal账号

```
1  mysql> CREATE USER canal IDENTIFIED WITH MYSQL_NATIVE_PASSWORD BY
   'canal';
2  Query OK, 0 rows affected (0.02 sec)
3  mysql> GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO
   'canal'@'%';
4  Query OK, 0 rows affected (0.01 sec)
5  mysql> FLUSH PRIVILEGES;
6  Query OK, 0 rows affected (0.02 sec)
7  mysql> exit
8  Bye
9  [anchu@localhost support-files]$ mysql -u canal -P 3306 -h
   192.168.120.110 -D test -p
10 Enter password:
11 Reading table information for completion of table and column namesYou can
   turn off this feature to get a quicker startup with -AWelcome to the
   MySQL monitor.
12 Commands end with ; or \g.
13 Your MySQL connection id is 3Server version: 5.7.24-log MySQL Community
   Server (GPL)
14 mysql> exit
```

(3) canal的工作原理

- canal 模拟 MySQL slave 的交互协议，伪装自己为 MySQL slave，向 MySQL master 发送dump 协议
- MySQL master 收到 dump 请求，开始推送 binary log 给 slave (即 canal)
- canal 解析 binary log 对象(原始为 byte 流)

(4) canal配置

修改instance 配置文件instance.properties


```

1  [anchu@localhost example]$cd /home/anchu/software/canal/conf/example
2  [anchu@localhost example]$ vi instance.properties
3  #####
4  mysql serverId , v1.0.26+ will autoGen
5  # canal.instance.mysql.slaveId=0
6  # enable gtid use true/false
7  canal.instance.gtidon=false
8  # position info数据库实例地址, 主数据库, 注意端口
   canal.instance.master.address=192.168.120.110:3306
9  canal.instance.master.journal.name=
10 canal.instance.master.position=
11 canal.instance.master.timestamp=
12 canal.instance.master.gtid=
13 # rds oss binlog
14 canal.instance.rds.accesskey=
15 canal.instance.rds.secretkey=
16 canal.instance.rds.instanceId=
17 # table meta tsdb infocanal.instance.tsdb.enable=true
18 #canal.instance.tsdb.url=jdbc:mysql://127.0.0.1:3306/canal_tsdb
19 #canal.instance.tsdb.dbUsername=canal
20 #canal.instance.tsdb.dbPassword=canal
21 #canal.instance.standby.address =
22 #canal.instance.standby.journal.name =
23 #canal.instance.standby.position =
24 #canal.instance.standby.timestamp =
25 #canal.instance.standby.gtid=
26 # username/password 数据库账号密码
27 canal.instance.dbUsername=canal
28 canal.instance.dbPassword=canal
29 canal.instance.connectionCharset = UTF-8
30 # enable druid Decrypt database password
31 canal.instance.enableDruid=false
32 #canal.instance.pwdPublicKey=MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALK4BUxdDltR
   RE5/zXpVEVPUGunvscYFtEip3pmLlhrWpacX7y7GCMo2/JM6LeHmiiNdH1FWgGCpUfircSwlW
   KUCAwEAAQ==
33 # table regex 白名单过滤canal.instance.filter.regex=.*\\.*
34 # table black regexcanal.instance.filter.black.regex=mysql\\.slave_.*
35 # table field
   filter(format:schema1.tableName1:field1/field2,schema2.tableName2:field1/
   field2)#
36 canal.instance.filter.field=test1.t_product:id/subject/keywords,test2.t_c
   ompany:id/name/contact/ch
37 # table field black
   filter(format:schema1.tableName1:field1/field2,schema2.tableName2:field1/
   field2)

```

```
38 #canal.instance.filter.black.field=test1.t_product:subject/product_image,  
    test2.t_company:id/name/contact/ch  
39 # mq config 定义主题  
40 canal.mq.topic=example  
41 # dynamic topic route by schema or table regex  
42 #canal.mq.dynamicTopic=mytest1.user,mytest2\\.*,.*\\.*  
43 #消息分区  
44 canal.mq.partition=0  
45 # hash partition config  
46 #canal.mq.partitionsNum=3  
47 #canal.mq.partitionHash=test.table:id^name,.*\\.*  
48 #canal.mq.dynamicTopicPartitionNum=test.*:4,mycanal:6#####  
#####
```

修改canal 配置文件canal.properties,暂时不用MQ,不用注册中心, 不用admin

```
1 [anchu@localhost conf]$ cd /home/anchu/software/canal/conf
2 [anchu@localhost conf]$ vi canal.properties
3 #主要改动
4 #canal server的唯一标识, 没有实际意义, 但是我们建议同一个cluster上的不同节点, 其ID
  尽可能唯一
5 canal.id =110
6 #canal server因为binding的本地IP地址, 建议使用内网(唯一, 集群可见, consumer可见)
  IP地址, 比如“10.0.1.21”。
7 #此IP主要为canalServer提供TCP服务而使用, 将会被注册到ZK中, Consumer将与此IP建立连
  接。
8 canal.ip =192.168.120.110
9 # register ip to zookeeper
10 canal.register.ip =
11 #cannal server的TCP端口
12 canal.port = 11111
13 canal.metrics.pull.port = 11112
14 # canal instance user/passwd
15 canal.user = canal
16 canal.passwd = canal
17 # canal admin config
18 #canal.admin.manager = 127.0.0.1:8089
19 canal.admin.port = 11110
20 canal.admin.user = admin
21 canal.admin.passwd = 4ACFE3202A5FF5CF467898FC58AAB1D615029441
22 # admin auto register
23 #canal.admin.register.auto = true
24 #canal.admin.register.cluster =
25 #canal.admin.register.name =
26 #zookeeper地址, 可集群
27 canal.zkServers =
28 # flush data to zk
29 canal.zookeeper.flush.period = 1000
30 canal.withoutNetty = false
31 # tcp, kafka, rocketMQ, rabbitMQ
32 canal.serverMode = tcp
33 # flush meta cursor/parse position to file
34 #canal将parse、position数据写入的本地文件目录
35 canal.file.data.dir = ${canal.conf.dir}
36 canal.file.flush.period = 1000
37 ## memory store RingBuffer size, should be Math.pow(2,n)
38 canal.instance.memory.buffer.size = 16384
39 ## memory store RingBuffer used memory unit size , default 1kb
40 canal.instance.memory.buffer.memunit = 1024
41 ## meory store gets mode used MEMSIZE or ITEMSIZE
42 canal.instance.memory.batch.mode = MEMSIZE
```

```
43 canal.instance.memory.rawEntry = true
44 # table meta tsdb info
45 canal.instance.tsdb.enable = true
46 canal.instance.tsdb.dir =
    ${canal.file.data.dir:../conf}/${canal.instance.destination:}
47 canal.instance.tsdb.url =
    jdbc:h2:${canal.instance.tsdb.dir}/h2;CACHE_SIZE=1000;MODE=MYSQL;
48 canal.instance.tsdb.dbUsername = canal
49 canal.instance.tsdb.dbPassword = canal
50 # dump snapshot interval, default 24 hour
51 canal.instance.tsdb.snapshot.interval = 24
52 # purge snapshot expire , default 360 hour(15 days)
53 canal.instance.tsdb.snapshot.expire = 360
```

(5) 启动canal, 测试

```

1  [anchu@localhost bin]$ pwd/home/anchu/software/canal/bin
2  [anchu@localhost bin]$ ./canal-startup.sh
3  cd to /home/anchu/software/canal/bin for workaround relative pathLOG
   CONFIGURATION : /home/anchu/software/canal/bin/./conf/logback.xml
4  #查看进程
5  [anchu@localhost bin]$ ps -ef |grep canal
6  anchu      83236  80040  0 20:51 pts/1    00:00:00 mysql -u canal -P 3306
   -h 192.168.120.110 -D test -p
7  anchu      84725      1  8 23:10 pts/2    00:00:04 /usr/bin/java -server -
   Xms2048m -Xmx3072m -Xmn1024m -XX:SurvivorRatio=2 -
   XX:PermSize=96m.....省略号.....
8  #停止canal
9  [anchu@localhost canal]$ ./bin/canal-stop.sh
10 localhost.localdomain: stopping canal 84725 ... 0ook! cost:1
11 #查看日志
12 [anchu@localhost canal]$ tail -f ../logs/canal/canal_stdout.log
13 2022-04-26 23:20:22.419 [destination = example , address =
   /192.168.120.110:3306 , EventParser] INFO
   c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - connect
   MySqlConnection to /192.168.120.110:3306...
14 2022-04-26 23:20:22.438 [destination = example , address =
   /192.168.120.110:3306 , EventParser] INFO
   c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - handshake
   initialization packet received, prepare the client authentication packet
   to send2022-04-26 23:20:22.439 [destination = example , address =
   /192.168.120.110:3306 , EventParser] INFO
   c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - client
   authentication packet is sent out.
15 2022-04-26 23:20:22.440 [destination = example , address =
   /192.168.120.110:3306 , EventParser] INFO
   c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - disconnect
   MySqlConnection to /192.168.120.110:3306...2022-04-26 23:20:22.441
   [destination = example , address = /192.168.120.110:3306 , EventParser]
   INFO c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - connect
   MySqlConnection to /192.168.120.110:3306...
16 2022-04-26 23:20:22.444 [destination = example , address =
   /192.168.120.110:3306 , EventParser] INFO
   c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - handshake
   initialization packet received, prepare the client authentication packet
   to send
17 2022-04-26 23:20:22.445 [destination = example , address =
   /192.168.120.110:3306 , EventParser] INFO
   c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - client
   authentication packet is sent out.

```

```

18 2022-04-26 23:20:22.540 [destination = example , address =
    /192.168.120.110:3306 , EventParser] INFO
    c.a.otter.canal.parse.inbound.mysql.MysqlConnection - Register slave
    RegisterSlaveCommandPacket[reportHost=192.168.120.110,reportPort=47134,reportUser=canal,reportPasswd=canal,serverId=1778384897,command=21]
19 2022-04-26 23:20:22.545 [destination = example , address =
    /192.168.120.110:3306 , EventParser] INFO
    c.a.otter.canal.parse.inbound.mysql.MysqlConnection - COM_BINLOG_DUMP
    with
    position:BinlogDumpCommandPacket[binlogPosition=4,slaveServerId=1778384897,binlogFileName=mysql-bin.000001,command=18]
20 2022-04-26 23:20:22.546 [MultiStageCoproprocessor-other-example-0] INFO
    com.taobao.tddl.dbsync.binlog.LogEvent - common_header_len= 19,
    number_of_event_types= 38

```

测试

```

▼ Shell | 复制代码
1 # mysql增加数据[anchu@localhost support-files]$ mysql -u root -P 3306 -h
  192.168.120.110 -D test -p
2 Enter password: 123456
3 mysql> insert into test(id,name) values(1,"t1");
4 Query OK, 1 row affected (0.04 sec)
5 mysql>
6 mysql> insert into test(id,name) values(2,"t2");
7 Query OK, 1 row affected (0.02 sec)
8 mysql> \q
9 Bye
10 #查看canal本地同步数据
11 [anchu@localhost example]$ cat ../canal.properties |grep h2
12 canal.instance.tsdb.url =
  jdbc:h2:${canal.instance.tsdb.dir}/h2;CACHE_SIZE=1000;MODE=MYSQL;
13 canal.instance.tsdb.spring.xml = classpath:spring/tsdb/h2-tsdb.xml
14 [anchu@localhost example]$ ll -h ../../conf/example/
15 total 120K
16 -rw-rw-r--. 1 anchu anchu 116K Apr 26 23:21 h2.mv.db
17 -rwxrwxr-x. 1 anchu anchu 2.2K Apr 26 23:19 instance.properties

```

需要注意canal.properties配置文件的几个相对路径配置，方便查看数据

```

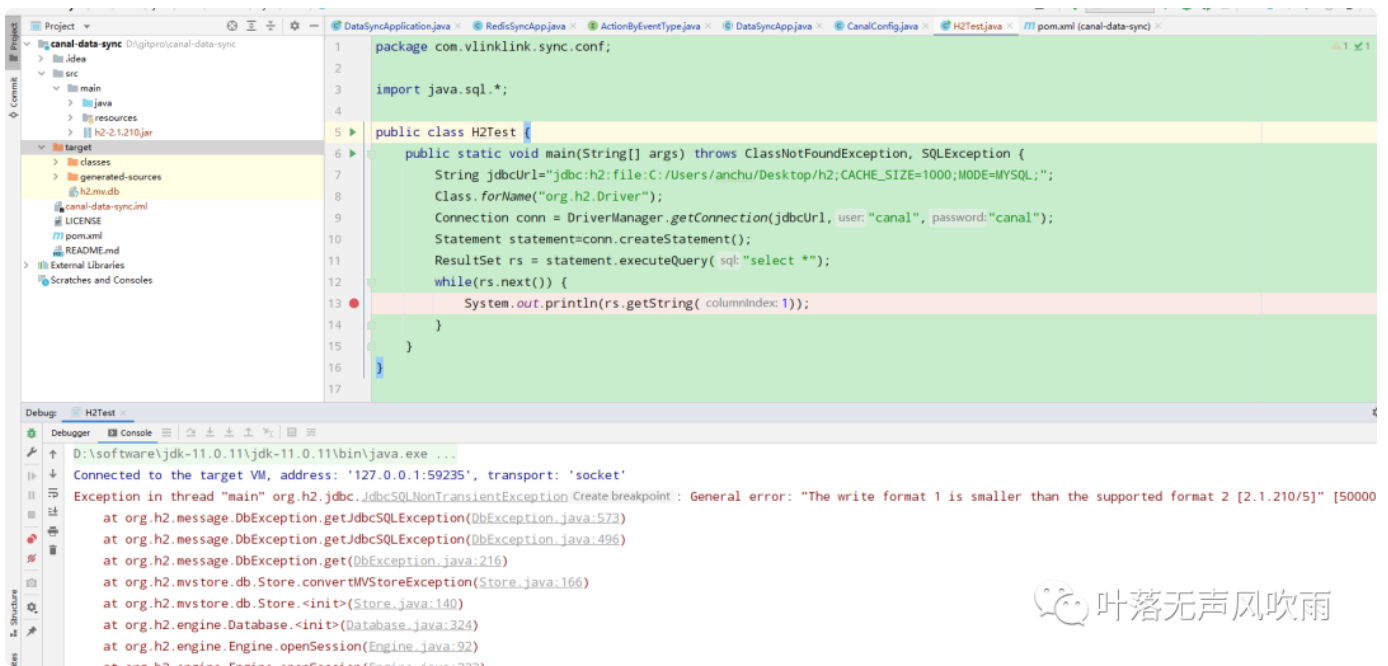
1  # 使用本地文件创建的连接
2  canal.conf.dir = ../conf  #/home/anchu/software/canal/conf
3  canal.file.data.dir= ${canal.conf.dir}  #/home/anchu/software/canal/conf
4  canal.instance.tsdb.dir =
    ${canal.file.data.dir:../conf}/${canal.instance.destination:}
5  # /home/anchu/software/canal/conf/example
6  canal.instance.tsdb.url =
    jdbc:h2:${canal.instance.tsdb.dir}/h2;CACHE_SIZE=1000;MODE=MYSQL;
7  # 及上面的h2文件为什么会在 /home/anchu/software/canal/conf/example/h2.mv.db
    目录, 是从这里配置的

```

可以将文件copy到windows并用dbeaver打开是否有添加的数据。

可以下载客户端去查询 <https://dbschema.com/download.html>

也可以下载驱动jdbc查询如下:



出现异常: General error: "The write format 1 is smaller than the supported format 2 [2.1.210/5]"

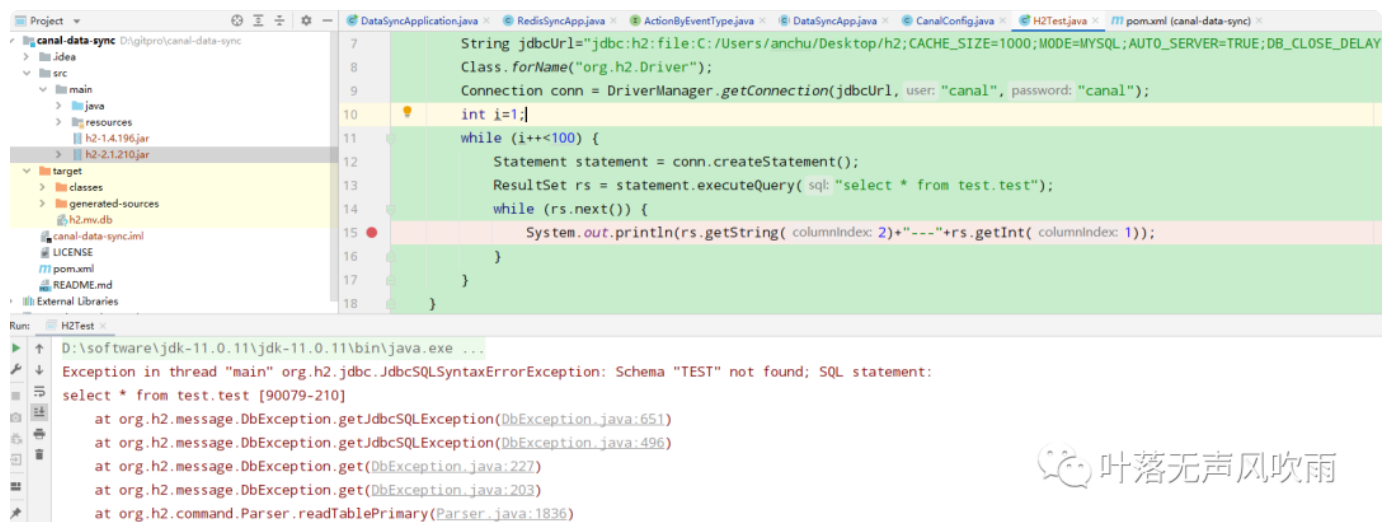
原因canal的h2版本和测试代码的版本不一致, 将canal的h2包copy到测试代码, 再次测试

```
1 [anchu@localhost lib]$ pwd/home/anchu/software/canal/lib
2 [anchu@localhost lib]$ ls |grep h2
3 h2-1.4.196.jar
```

或者查看生成的h2.mv.db文件，查询test和insert关键字，可以看到数据更新信息同步到h2中

[illegible]

发现找不到test表，明明h2文件看到了关键字，却查不到这是为什么呢？



那是因为，canal 使用 tsdb 保存保存表结构 meta 的信息变化，本地使用默认使用的 h2db，可以配置为 mysql

参考文档: <https://blog.csdn.net/wfh45678/article/details/118546668>

由于版本问题，修改canal lib包的h2版本为2.1.210，然后就可以通过h2查看别结构元数据信息


```

1 [anchu@localhost lib]$ mv h2-1.4.196.jar h2-1.4.196.jarold
2 [anchu@localhost lib]$ mv h2-2.1.210.jarold h2-2.1.210.jar
3 [anchu@localhost lib]$ ll |grep h2
4 -rwxrwxr-x. 1 anchu anchu 1821816 Oct 13 2017 h2-1.4.196.jarold
5 -rw-r--r--. 1 anchu anchu 2531600 Jan 17 01:12 h2-2.1.210.jar

```

```

mysql> create table test(id int(10), name varchar(10));
Query OK, 0 rows affected (0.16 sec)

mysql> insert into root.test(id,name) values(2,"t2");
ERROR 1146 (42502): Table 'root.test' doesn't exist
mysql> insert into test(id,name) values(2,"t2");
Query OK, 1 row affected (0.03 sec)

mysql> drop database test;
Query OK, 1 row affected (0.02 sec)

mysql> create database test;
Query OK, 1 row affected (0.01 sec)

mysql> connect test;
Connection id: 9
Current database: test

mysql> create table test(id int(10), name varchar(10));
Query OK, 0 rows affected (0.16 sec)

mysql> insert into test(id,name) values(2,"t2");
Query OK, 1 row affected (0.03 sec)

mysql> select * from test;
+----+-----+
| id | name |
+----+-----+
| 2 | t2 |
+----+-----+
1 row in set (0.00 sec)

mysql>

```

```

2022-04-27 02:18:44.978 [destination = example, address = /192.168.120.110:3306, EventParser] INFO c.alibaba.otter.canal.parse.driver.mysql.MysqlConnector - client authentication packet is sent out.
2022-04-27 02:18:45.092 [destination = example, address = /192.168.120.110:3306, EventParser] INFO c.a.otter.canal.parse.inbound.mysql.l.MysqlConnection - Register slave RegisterSlaveCommandPacket[reportHost=192.168.120.110,reportPort=47508,reportUser=canal,reportPasswd=canal,serverId=1778384897,command=21]
2022-04-27 02:18:45.094 [destination = example, address = /192.168.120.110:3306, EventParser] INFO c.a.otter.canal.parse.inbound.mysql.l.MysqlConnection - COM_BINLOG_DUMP with position:BinlogDumpCommandPacket[binlogPosition=4,slaveServerId=1778384897,binlogFileName=mysql-bin.000001,command=18]
2022-04-27 02:18:45.109 [MultiStageCoprocessor-other-example-0] INFO com.taobao.tddl.dbsync.binlog.LogEvent - common_header_len= 19, number_of_event_types= 38
2022-04-27 02:22:11.173 [MultiStageCoprocessor-other-example-0] INFO com.alibaba.druid.sql.parser.SchemaRepository - replaced table 'test'

```

```

[anchu@localhost example]$ ls
h2.mv.dbold instance.properties
[anchu@localhost example]$ ls
h2.mv.dbold instance.properties
[anchu@localhost example]$
[anchu@localhost example]$
[anchu@localhost example]$
[anchu@localhost example]$
[anchu@localhost example]$
[anchu@localhost example]$
[anchu@localhost example]$

```

id	DESTINATION	BINLOG FILE	BINLOG OFFSET	BINLOG MASTER ID	BINLOG TIMESTAMP	USE SCHEMA	SQL_SCHEMA	SQL_TABLE	SQL TEXT	SQL_TYPE	EXTRA
1	example	mysql-bin.000001	219	1	1,651,051,142,000	test	test	test	create database test	CREATE	[NULL]
2	example	mysql-bin.000001	378	1	1,651,051,161,000	test	test	test	create table test(id int(10), name varchar(10))	CREATE	[NULL]
3	example	mysql-bin.000001	890	1	1,651,051,310,000	test	test	test	drop database test	QUERY	[NULL]
4	example	mysql-bin.000001	1,047	1	1,651,051,316,000	test	test	test	create database test	CREATE	[NULL]
5	example	mysql-bin.000001	1,206	1	1,651,051,331,000	test	test	test	create table test(id int(10), name varchar(10))	CREATE	[NULL]

```

选择 "h2.mv.db" 配置
连接设置
H2 Embedded 连接设置
常规 驱动属性
初始化
Shell 命令
Transactions
常规
元数据
错误处理
SQL 编辑器
JDBC URL: jdbc:h2c:C:\Users\anchu\Desktop\h2.mv.db
路径: C:\Users\anchu\Desktop\h2.mv.db
认证 (Database Native)
用户名: canal
密码: *****
Save password locally
可以在连接参数中使用变量。
驱动名称: H2 Embedded
测试连接...
确定 取消

```

由上图可以以看到，元数据信息变化表，会存储ddl语句相关日志文件及其偏移量。

我们可以借助mysql提供的mysqlbinlog工具进行查看，操作如下：

```

1  mysqlbinlog --base64-output=decode-rows --start-position="219"
   ~/software/mysql-5.7.24/data/mysql-bin.000001
2  /*!50530
3  SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;/*!50003
4  SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;DELIMITER
   /*!*/;
5  # at 219
6  #220427 2:19:02 server id 1  end_log_pos 313 CRC32 0xacaaf5d4  Query
   thread_id=7  exec_time=0  error_code=0SET TIMESTAMP=1651051142/*!*/;
7  SET @@session.pseudo_thread_id=7/*!*/;
8  SET @@session.foreign_key_checks=1,
9  @@session.sql_auto_is_null=0,
10 @@session.unique_checks=1,
11 @@session.autocommit=1/*!*/;
12 SET @@session.sql_mode=1075838976/*!*/;
13 SET @@session.auto_increment_increment=1,
14 @@session.auto_increment_offset=1/*!*/;/*!C utf8 *//*!*/;
15 SET @@session.character_set_client=33,
16 @@session.collation_connection=33,
17 @@session.collation_server=8/*!*/;
18 SET @@session.lc_time_names=0/*!*/;
19 SET @@session.collation_database=DEFAULT/*!*/;create database test/*!*/;
20 # at 313.....

```

如图：可以看到219开始确实紧跟着create database test

```

[anchu@localhost conf]$ mysqlbinlog --base64-output=decode-rows --start-position="219" ~/software/mysql-5.7.24/data/mysql-bin.000001
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 219
#220427 2:19:02 server id 1  end_log_pos 313 CRC32 0xacaaf5d4  Query  thread_id=7  exec_time=0  error_code=0
SET TIMESTAMP=1651051142/*!*/;
SET @@session.pseudo_thread_id=7/*!*/;
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1, @@session.autocommit=1/*!*/;
SET @@session.sql_mode=1075838976/*!*/;
SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*!*/;
/*!C utf8 *//*!*/;
SET @@session.character_set_client=33,@@session.collation_connection=33,@@session.collation_server=8/*!*/;
SET @@session.lc_time_names=0/*!*/;
SET @@session.collation_database=DEFAULT/*!*/;
create database test
/*!*/;
# at 313
#220427 2:19:21 server id 1  end_log_pos 378 CRC32 0x632a7326  Anonymous_GTID  last_committed=1  sequence_number=2  rbr_only=no
SET @@SESSION.GTID_NEXT= 'ANONYMOUS' /*!*/;
# at 378
#220427 2:19:21 server id 1  end_log_pos 500 CRC32 0xb5b9a8f4  Query  thread_id=8  exec_time=0  error_code=0
use `test`/*!*/;

```

叶落无声风吹雨

扩展

binlog日志类型

- statement：记录执行的语句；对于更新而言不需要记录大量的行数据，但是有些信息无法记录比如随机数或者当前时间，所以还需要一些上下文信息。
- row：只记录修改的行，不用上下文，但是数据量可能较大。
- mixed：对于不需要上下文的语句，使用statement，否则使用row，所以会同时存在两种格式；

如何查看binlog

- 如果在mysql客户端内，可以使用show binlog events;这条命令会显示第一个binlog文件内的事件；当然可以指定binlog文件，比如：show binlog events in 'xxx-bin.0000N'。另外也可以使用show binary logs命令查看当前所有binlog文件名。
- 如果不在mysql客户端，可以使用mysql提供的mysqlbinlog命令。比如：mysqlbinlog -v --base64-output=decode-rows --start-position="156" ~/mysql/master/data/mysql-bin.000004
- 简单记一下几个参数的含义：
- -v：将row模式的反解为statement模式，方便查阅；
- --base64-output：如果不加，默认输出是base64格式的，加上这个参数，输出是base64解密后的；
- --start-position：指定开始位置，当然也可以指定结束位置；
- 另外还有一些其他参数也可以指定，比如库名以及表名等。

binlog事件类型

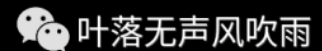
- 总共有3个版本：v1，v2和v4。mysql5以上使用过的v4。这里只记录这几个关键的事件类型：
- QUERY_EVENT：在statement模式下，增删改的语句都会生成该事件；在row模式下，DDL的改动会生成该事件；
- ROTATE_EVENT：新的binlog文件生成时，会记录该事件，内容就是下一个binlog文件的文件名；
- FORMAT_DESCRIPTION_EVENT：每一个binlog文件的起始事件，描述文件属性；
- TABLE_MAP_EVENT：在row模式下会有，每一个更新事件都会先有一个TABLE_MAP_EVENT事件，用于记录表的一些信息。
- WRITE_ROWS_EVENT：在row模式下会有，insert；
- UPDATE_ROWS_EVENT：在row模式下会有，update；
- DELETE_ROWS_EVENT：在row模式下会有，delete

```

# at 890
#220427 2:21:50 server id 1 end_log_pos 982 CRC32 0x3a95745b Query thread_id=8 exec_time=0 error_code=0
SET TIMESTAMP=1651051310/*!*/;
drop database test
/*!*/;
# at 982
#220427 2:21:56 server id 1 end_log_pos 1047 CRC32 0x3cf7ae0 Anonymous_GTID last_committed=4 sequence_number=5 rbr_only=no
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 1047
#220427 2:21:56 server id 1 end_log_pos 1141 CRC32 0x10ad3325 Query thread_id=8 exec_time=0 error_code=0
SET TIMESTAMP=1651051336/*!*/;
create database test
/*!*/;
# at 1141
#220427 2:22:11 server id 1 end_log_pos 1206 CRC32 0xa8d219ea Anonymous_GTID last_committed=5 sequence_number=6 rbr_only=no
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 1206
#220427 2:22:11 server id 1 end_log_pos 1328 CRC32 0x4fa1d021 Query thread_id=9 exec_time=0 error_code=0
use test/*!*/;
SET TIMESTAMP=1651051331/*!*/;
create table test(id int(10), name varchar(10))
/*!*/;
# at 1328
#220427 2:22:32 server id 1 end_log_pos 1393 CRC32 0xefacd90d Anonymous_GTID last_committed=6 sequence_number=7 rbr_only=yes
/*!50718 SET TRANSACTION ISOLATION LEVEL READ COMMITTED'/*!*/;
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 1393
#220427 2:22:32 server id 1 end_log_pos 1465 CRC32 0x9e93ed4e Query thread_id=9 exec_time=0 error_code=0
SET TIMESTAMP=1651051352/*!*/;
BEGIN
/*!*/;
# at 1465
# at 1529
#220427 2:22:32 server id 1 end_log_pos 1579 CRC32 0xada79f6d Table_map: 'test`.`test` mapped to number 121
# at 1579
#220427 2:22:32 server id 1 end_log_pos 1622 CRC32 0xf025e9e3 Write_rows: table id 121 flags: STMT_END_F
# at 1622
#220427 2:22:32 server id 1 end_log_pos 1653 CRC32 0x4b40643f Xid = 282
COMMIT/*!*/;
# at 1653
#220427 19:55:31 server id 1 end_log_pos 1700 CRC32 0xe0f4cc90 Rotate to mysql-bin.000002 pos: 4
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=GOLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
[mysqllocalhost:3306]

```

insert 语句



参考文档：

开源实战 | Canal生产环境常见问题总结与分析

<https://cloud.tencent.com/developer/article/1645881>

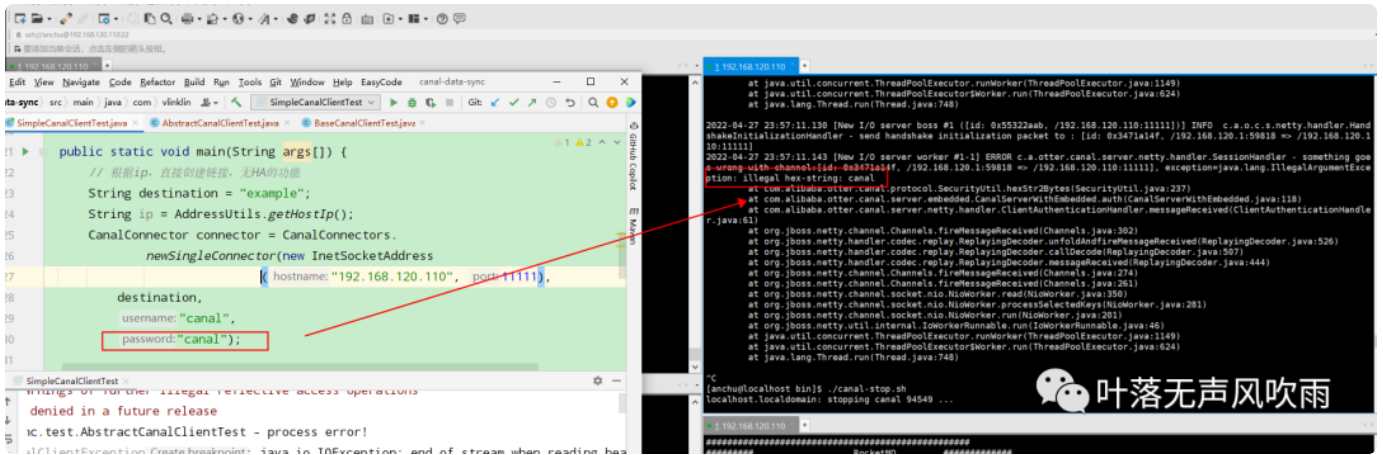
【MySQL（二十一）】binlog 事件

<https://blog.csdn.net/u010900754/article/details/108458028>

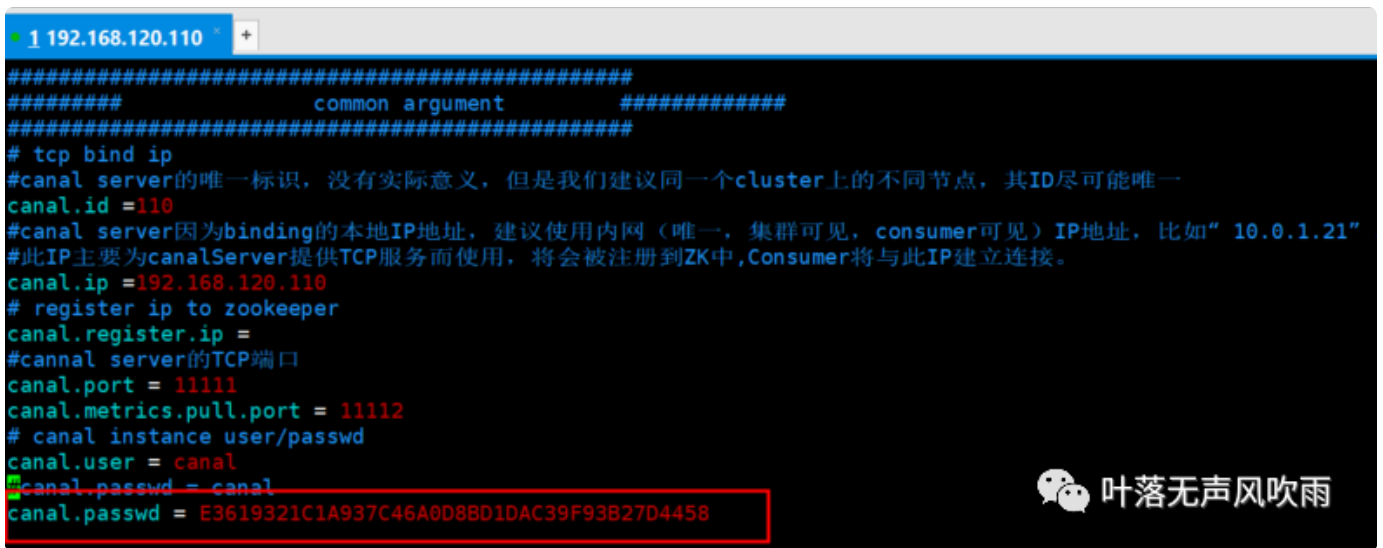
(6) canal client测试

测试消费，canal开源项目地址中有个测试用例，只需要修改ip然后启动

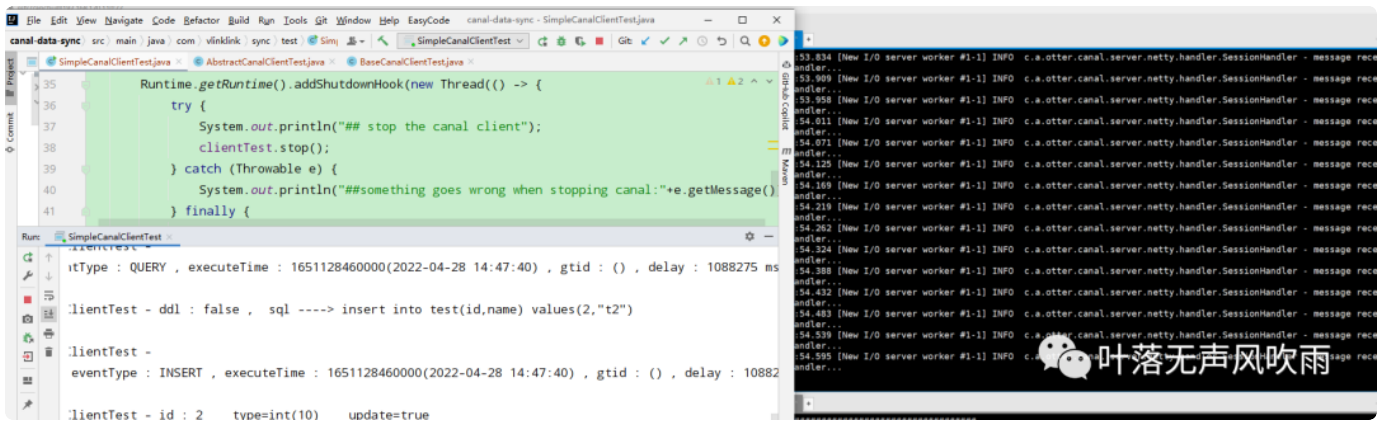
<https://github.com/alibaba/canal/blob/master/example/src/main/java/com/alibaba/otter/canal/example/SimpleCanalClientTest.java>



启动失败，原因查看canal日志，是因为密码校验不能识别canal，所以需要修改canal.properties密码为二进制加密的。



再次启动消费成功，可以从日志中看到增量的sql



```
SimpleCanalClientTest
16:11:09.163 [Thread-0] INFO com.vlinklink.sync.test.AbstractCanalClientTest - ddl : false , sql ----> delete from test
16:11:09.166 [Thread-0] INFO com.vlinklink.sync.test.AbstractCanalClientTest -
----- binlog[mysql-bin.000002:3034] , name[test,test] , eventType : DELETE , executeTime : 1651133469000(2022-04-28 16:11:09) , gtid : () , delay : 163 ms
16:11:09.169 [Thread-0] INFO com.vlinklink.sync.test.AbstractCanalClientTest - id : 3 type=int(10)
16:11:09.169 [Thread-0] INFO com.vlinklink.sync.test.AbstractCanalClientTest - name : t3 tvoe=varchar(10)
16:11:09.171 [Thread-0] INFO
===== binlog[mysql-bin.000002:3034] , name[test,test] , eventType : DELETE , executeTime : 1651133469000(2022-04-28 16:11:09) , gtid : () , delay : 163 ms
16:12:20.531 [Thread-0] INFO com.vlinklink.sync.test.AbstractCanalClientTest -
----- binlog[mysql-bin.000002:3173] , name[test,test] , eventType : ERASE , executeTime : 1651133540000(2022-04-28 16:12:20) , gtid : () , delay : 531 ms
16:12:20.531 [Thread-0] INFO com.vlinklink.sync.test.AbstractCanalClientTest - ddl : true , sql ----> DROP TABLE 'test' /* generated by server */
```

分享一个canal +springboot整合的小项目，git地址：

<https://github.com/zfsndtl/canal-data-sync>