MySQL学习笔记（Day007：多实例下/SSL）

MySQL学习

一. 多实例安装 – 多版本

1. [mysqld_multi]标签

- **`[mysqld_multi]` 是否需要配置**
  从操作演示来看，在 `my.cnf` (*老师给的模板配置*)上直接配置 `[mysqld1]` 、 `[mysqld2]` 等实例标签，而 不配置`[mysqld_multi]` ,使用 `mysqld_multi start 1` 也是 可以启动 数据库实例的，但是没有 `mysqld_safe` 的守护进程。所以该标签 需要配置

2. 停止mysql实例

- **`multi_admin`用户 的作用**
  通过官方文档中我们看到， `'multi_admin'@'localhost'` 这个用户主要的作用是用来 关闭 数据库实例，因为文档中只授权了 `SHUTDOWN` 权限。所以在 `[mysqld_multi]` 标签下，我们需要配置 `user` 和 `password` **(注意 `5.7.9` 中是 `pass`)**来进行关闭数据库实例。
- **`[client]` 标签**
  从操作演示来看，老师并没有在 `[mysqld_multi]` 下配置 `user` 和 `password` ，但是仍然可以关闭数据库，原因是因为 `/root/.my.cnf` 中存在了 `[client]` 标签。该标签下的用户 `user = root` 有关闭数据库实例的权限，因此可以关闭数据库。

> 如果在 `[client]` 和 `[mysqld_multi]` 标签中同时存在 `user` 和 `password` , 则在关闭数据库实例中会使用 `[mysqld_multi]` 中的 `user` 去关闭。
> (存在精确匹配的标签，则优先使用精确匹配标签下的配置项)

3. 多实例安装 – 多版本

- **环境说明**
  - `mysqld1` – MySQL 5.7.9
  - `mysqld2` – MySQL 5.7.9
  - `mysqld3` – MySQL 5.6.27
  - `mysqld4` – MySQL 5.6.27
- **配置文件**

```
[client]
user = root
password = 123

[mysqld_multi]  # 这里使用了client标签中的user，故这里不再定义user
mysqld = /usr/local/mysql/bin/mysqld_safe
log = /var/log/mysqld_multi.log

[mysqld1]
server-id = 11
datadir = /data1
basedir = /usr/local/mysql   # basedir定义使用了5.7的mysql版本
port = 3307
socket = /tmp/mysql.sock1

[mysqld2]
server-id = 22
datadir = /data2
basedir = /usr/local/mysql
port = 3308
socket = /tmp/mysql.sock2

[mysqld3]
server-id = 33
datadir = /data3
basedir = /usr/local/mysql56  # basedir定义了使用5.6的mysql版本
port = 3309
socket = /tmp/mysql.sock3
plugin_dir=/usr/local/mysql56/lib/plugin  # plugin 目录也变了

#这里无需特别配置mysqld，可以继承使用[mysqld_multi]中的配置，然后根据basedir找到对应的mysqld

[mysqld4]
server-id = 44
datadir = /data4
basedir = /usr/local/mysql56
port = 3310
socket = /tmp/mysql.sock4
plugin_dir=/usr/local/mysql56/lib/plugin

#--------------以下参数是老师的模板，只是将个别size调小-----------
[mysqld]
########basic settings########
server-id = 100
port = 3306
user = mysql
bind_address = 0.0.0.0
#autocommit = 0
character_set_server=utf8mb4
skip_name_resolve = 1
max_connections = 800
max_connect_errors = 1000
datadir = /data/mysql_data
transaction_isolation = READ-COMMITTED
explicit_defaults_for_timestamp = 1
join_buffer_size = 134217728
tmp_table_size = 67108864
tmpdir = /tmp
max_allowed_packet = 16777216
sql_mode = "STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION,NO_ZERO_DATE,NO_ZERO_IN_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER"
interactive_timeout = 1800
wait_timeout = 1800
read_buffer_size = 16777216
read_rnd_buffer_size = 33554432
sort_buffer_size = 33554432
########log settings########
log_error = error.log
slow_query_log = 1
slow_query_log_file = slow.log
log_queries_not_using_indexes = 1
log_slow_admin_statements = 1
log_slow_slave_statements = 1
log_throttle_queries_not_using_indexes = 10
expire_logs_days = 90
long_query_time = 2
min_examined_row_limit = 100
########replication settings########
master_info_repository = TABLE
relay_log_info_repository = TABLE
log_bin = bin.log
sync_binlog = 1
gtid_mode = on
enforce_gtid_consistency = 1
log_slave_updates
binlog_format = row
relay_log = relay.log
relay_log_recovery = 1
binlog_gtid_simple_recovery = 1
slave_skip_errors = ddl_exist_errors
########innodb settings########
innodb_page_size = 8192
innodb_buffer_pool_size = 1G    # 该参数减小到1G
innodb_buffer_pool_instances = 8
innodb_buffer_pool_load_at_startup = 1
innodb_buffer_pool_dump_at_shutdown = 1
innodb_lru_scan_depth = 2000
innodb_lock_wait_timeout = 5
innodb_io_capacity = 4000
innodb_io_capacity_max = 8000
innodb_flush_method = O_DIRECT
innodb_file_format = Barracuda
innodb_file_format_max = Barracuda
#innodb_log_group_home_dir = /redolog/
#innodb_undo_directory = /undolog/
innodb_undo_logs = 128
innodb_undo_tablespaces = 3
innodb_flush_neighbors = 1
innodb_log_file_size = 128M  # 该参数减小到 128M
innodb_log_buffer_size = 16777216
innodb_purge_threads = 4
innodb_large_prefix = 1
innodb_thread_concurrency = 64
innodb_print_all_deadlocks = 1
innodb_strict_mode = 1
innodb_sort_buffer_size = 67108864
########semi sync replication settings########
plugin_dir=/usr/local/mysql/lib/plugin
#plugin_load = "rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=semisync_slave.so"
loose_rpl_semi_sync_master_enabled = 1
loose_rpl_semi_sync_slave_enabled = 1
loose_rpl_semi_sync_master_timeout = 5000

[mysqld-5.7]
innodb_buffer_pool_dump_pct = 40
innodb_page_cleaners = 4
innodb_undo_log_truncate = 1
innodb_max_undo_log_size = 1G   # 该参数减小到1G
innodb_purge_rseg_truncate_frequency = 128
binlog_gtid_simple_recovery=1
log_timestamps=system
transaction_write_set_extraction=MURMUR32
show_compatibility_56=on
```

> **注意MySQL5.6.27的 `plugin_dir` 的路径**
>
> 配置说明：
> 1：配置的标签顺序没有关系，不会影响最终配置的有效性。
> 2：同类型标签中的配置项会合并，形成一个大的配置项
> 2： 匹配度高 的标签中的配置项的 值 ，会 覆盖 掉 匹配度低 的标签中的配置项的 值
>
> **[mysqld `N`]中的配置项会和[mysqld]中的配置项进行合并，并且[mysqld `N`]中已有的配置项的值，会覆盖掉[mysqld]中的配置项的值,如 `datadir` , `port` 等**

- **安装操作**

```
#
# 准备好数据目录，并初始化安装
#
[root@MyServer ~]> mkdir /data1
[root@MyServer ~]> mkdir /data2
[root@MyServer ~]> mkdir /data3
[root@MyServer ~]> mkdir /data4
[root@MyServer ~]> chown mysql.mysql /data{1..4}
[root@MyServer ~]> mysqld --initialize --user=mysql --datadir=/data1
#
# 这里无输出，临时密码见 /data1/error.log
#
[root@MyServer ~]> mysqld --initialize --user=mysql --datadir=/data2
#
# 这里无输出，临时密码见 /data1/error.log
#
[root@MyServer mysql56]> pwd
/usr/local/mysql56
[root@MyServer mysql56]> scripts/mysql_install_db  --user=mysql --datadir=/data3
#
# 这里有部分信息输出
# 安装后，需要检查error.log 确保没有错误出现
# 注意使用空密码登录后，修改密码
#
[root@MyServer mysql56]> scripts/mysql_install_db  --user=mysql --datadir=/data4
#
# 这里有部分信息输出
# 安装后，需要检查error.log 确保没有错误出现
# 注意使用空密码登录后，修改密码
#
[root@MyServer ~]> cp /usr/local/mysql/support-files/mysqld_multi.server  /etc/init.d/mysqld_multid
# 拷贝启动脚本，方便自启
[root@MyServer ~]> chkconfig mysqld_multid on

[root@MyServer ~]> mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysqld1 is not running
MySQL server from group: mysqld2 is not running
MySQL server from group: mysqld3 is not running
MySQL server from group: mysqld4 is not running

[root@MyServer ~]> mysqld_multi  report
Reporting MySQL servers
MySQL server from group: mysqld1 is running
MySQL server from group: mysqld2 is running
MySQL server from group: mysqld3 is running
MySQL server from group: mysqld4 is running
```

```
[root@MyServer ~]> ps -ef | grep mysqld
root     13859     1  0 22:35 pts/1    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --server-id=11 --datadir=/data1 --basedir=/usr/local/mysql --port=3307 --socket=/tmp/mysql.sock1
root     13865     1  0 22:35 pts/1    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --server-id=22 --datadir=/data2 --basedir=/usr/local/mysql --port=3308 --socket=/tmp/mysql.sock2
root     13872     1  0 22:35 pts/1    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --server-id=33 --datadir=/data3 --basedir=/usr/local/mysql56 --port=3309 --socket=/tmp/mysql.sock3 --plugin_dir=/usr/local/mysql56/lib/plugin
root     13886     1  0 22:35 pts/1    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --server-id=44 --datadir=/data4 --basedir=/usr/local/mysql56 --port=3310 --socket=/tmp/mysql.sock4 --plugin_dir=/usr/local/mysql56/lib/plugin
#
# 上面是mysqld_safe的守护进程
# 下面是实际的mysqld的进程，观察mysqld的路径
# 因为指定了basedir，所以会自动识别mysqld的路径
#
mysql    17783 13859  0 22:35 pts/1    00:00:00 /usr/local/mysql-5.7.9-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql --datadir=/data1 --plugin-dir=/usr/local/mysql/lib/plugin --user=mysql --server-id=11 --log-error=/data1/error.log --pid-file=/data1/MyServer.pid --socket=/tmp/mysql.sock1 --port=3307
mysql    17784 13865  0 22:35 pts/1    00:00:00 /usr/local/mysql-5.7.9-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql --datadir=/data2 --plugin-dir=/usr/local/mysql/lib/plugin --user=mysql --server-id=22 --log-error=/data2/error.log --pid-file=/data2/MyServer.pid --socket=/tmp/mysql.sock2 --port=3308
mysql    17819 13872  0 22:35 pts/1    00:00:00 /usr/local/mysql-5.6.27-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql56 --datadir=/data3 --plugin-dir=/usr/local/mysql56/lib/plugin --user=mysql --server-id=33 --log-error=/data3/error.log --pid-file=/data3/MyServer.pid --socket=/tmp/mysql.sock3 --port=3309
mysql    17824 13886  0 22:35 pts/1    00:00:00 /usr/local/mysql-5.6.27-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql56 --datadir=/data4 --plugin-dir=/usr/local/mysql56/lib/plugin --user=mysql --server-id=44 --log-error=/data4/error.log --pid-file=/data4/MyServer.pid --socket=/tmp/mysql.sock4 --port=3310
root     17988  2657  0 22:44 pts/1    00:00:00 grep mysqld


[root@MyServer ~]> ps -ef | grep mysqld |  grep -v mysqld_safe | grep -v grep | awk '{print $8" "$9}'
/usr/local/mysql-5.7.9-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql
/usr/local/mysql-5.7.9-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql
/usr/local/mysql-5.6.27-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql56
/usr/local/mysql-5.6.27-linux-glibc2.5-x86_64/bin/mysqld --basedir=/usr/local/mysql56
```

`mysql3` 和 `mysql4` 初始状态没有密码，以前可以直接使用 `mysql -S mysql.sock` 登录，而现在登录的时候特别注意，因为我们使用了 `[client]` 标签,登录的时候如果不加 `-p` 参数会默认使用标签下的 `user` 和 `password`，然后导致登录不进去，所以需要使用如下登录方式：

```
shell> mysql -u root -P3309 -S /tmp/mysql.sock3 -p
Enter password: [直接回车]
elcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.6.27-log 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> set password = password("123"); #进行修改密码
```

- **设置login-path**

  设置 `login-path` 主要为了能够简化登录，同时还可以让每个数据库的密码都不同，避免使用[client]下的统一用户名密码

```
[root@MyServer ~]> mysql_config_editor  set -G mysql1 -u root -p -S /tmp/mysql.sock1
[root@MyServer ~]> mysql_config_editor  set -G mysql2 -u root -p -S /tmp/mysql.sock2
[root@MyServer ~]> mysql_config_editor  set -G mysql3 -u root -p -S /tmp/mysql.sock3
[root@MyServer ~]> mysql_config_editor  set -G mysql4 -u root -p -S /tmp/mysql.sock4
# 然后可以使用mysql --login-path=mysql1 这种方式登录


--
-- mysql1
--
mysql> select version();
+-----------+
| version() |
+-----------+
| 5.7.9-log |
+-----------+
1 row in set (0.01 sec)

mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3307  |
+---------------+-------+
1 row in set (0.00 sec)

--
-- mysql2
--
mysql> select version();
+-----------+
| version() |
+-----------+
| 5.7.9-log |
+-----------+
1 row in set (0.00 sec)

mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3308  |
+---------------+-------+
1 row in set (0.00 sec)

--
-- mysql3
--
mysql> select version();
+------------+
| version()  |
+------------+
| 5.6.27-log |  -- mysql 5.6.27
+------------+
1 row in set (0.00 sec)

mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3309  |
+---------------+-------+
1 row in set (0.00 sec)

--
-- mysql4
--
mysql> select version();
+------------+
| version()  |
+------------+
| 5.6.27-log |  -- mysql 5.6.27
+------------+
1 row in set (0.00 sec)

mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3310  |
+---------------+-------+
1 row in set (0.00 sec)
```

二. SSL安装

SSL（Secure Socket Layer）是维护Client - Server之间加密通讯的一套安全协议；

```
mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3307  |
+---------------+-------+
1 row in set (0.00 sec)

mysql> show variables like "%ssl%";
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | DISABLED |  --  SSL被禁止了
| have_ssl      | DISABLED |
| ssl_ca        |          |
| ssl_capath    |          |
| ssl_cert      |          |
| ssl_cipher    |          |
| ssl_crl       |          |
| ssl_crlpath   |          |
| ssl_key       |          |
+---------------+----------+
9 rows in set (0.00 sec)
```

> 经过之前的多实例安装，是没有开启SSL配置的

1. 开启SSL (5.7.9)

- 环境说明
  - 虚拟机1：MyServer； IP：172.18.14.68, MySQL实例1 - mysql1
  - 虚拟机2：MyServer； IP：172.18.14.41, MySQL客户端

> 操作过程中看到的192.168.115.223 是宿主机IP，因为使用KVM虚拟机的NAT功能，所以会被转换

```
#
# 当前虚拟机1 MyServer
#
[root@MyServer mysql]> pwd
/usr/local/mysql
[root@MyServer mysql]> bin/mysql_ssl_rsa_setup  --datadir=/data1 --user=mysql --uid=mysql
# 使用--uid后，就不需要chown mysql.mysql *.pem
[root@MyServer data1]# pwd
/data1
[root@MyServer data1]# ll | grep pem
-rw-------. 1 mysql mysql      1675 Nov 25 23:55 ca-key.pem
-rw-r--r--. 1 mysql mysql      1070 Nov 25 23:55 ca.pem
-rw-r--r--. 1 mysql mysql      1078 Nov 25 23:55 client-cert.pem #客户端证书文件
-rw-------. 1 mysql mysql      1679 Nov 25 23:55 client-key.pem  #客户端私钥文件
-rw-------. 1 mysql mysql      1675 Nov 25 23:55 private_key.pem #用于密钥交换的公钥
-rw-r--r--. 1 mysql mysql       451 Nov 25 23:55 public_key.pem #用户密钥交换的私钥
-rw-r--r--. 1 mysql mysql      1078 Nov 25 23:55 server-cert.pem #服务器端证书文件
-rw-------. 1 mysql mysql      1679 Nov 25 23:55 server-key.pem #服务器端私钥文件
[root@MyServer data1]> mysqld_multi  stop 1
[root@MyServer data1]> mysqld_multi  start 1
```

> 关于几个pem文件的用途说面，见官方文档，并搜索关键字 private/public key-pair

```
--
-- 当前虚拟机1 MyServer ,当前实例为 mysql1
--
mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3307  |
+---------------+-------+
1 row in set (0.00 sec)

mysql> show variables like "%ssl%";
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| have_openssl  | YES             |  -- 已经支持SSL
| have_ssl      | YES             |
| ssl_ca        | ca.pem          |
| ssl_capath    |                 |
| ssl_cert      | server-cert.pem |  -- 公钥文件
| ssl_cipher    |                 |
| ssl_crl       |                 |
| ssl_crlpath   |                 |
| ssl_key       | server-key.pem  |  -- 私钥文件
+---------------+-----------------+
9 rows in set (0.00 sec)

mysql> \s  -- status
--------------
mysql  Ver 14.14 Distrib 5.7.9, for linux-glibc2.5 (x86_64) using  EditLine wrapper

Connection id:          2
Current database:
Current user:           root@localhost
SSL:                    Not in use  -- 此时本地socket登录，不用SSL
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.9-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8mb4
Db     characterset:    utf8mb4
Client characterset:    utf8
Conn.  characterset:    utf8
UNIX socket:            /tmp/mysql.sock1
Uptime:                 6 min 16 sec

Threads: 1  Questions: 7  Slow queries: 0  Opens: 108  Flush tables: 1  Open tables: 101  Queries per second avg: 0.018
--------------

mysql> create user 'burn'@'%' identified by '123'; -- 创建一个burn@%用户，先不require ssl
Query OK, 0 rows affected (0.02 sec)

mysql> grant all on *.* to 'burn'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> select * from mysql.user where user='burn'\G
*************************** 1. row ***************************
                  Host: %
                  User: burn
           Select_priv: Y
           Insert_priv: Y
           Update_priv: Y
           Delete_priv: Y
           Create_priv: Y
             Drop_priv: Y
           Reload_priv: Y
         Shutdown_priv: Y
          Process_priv: Y
             File_priv: Y
            Grant_priv: N
       References_priv: Y
            Index_priv: Y
            Alter_priv: Y
          Show_db_priv: Y
            Super_priv: Y
 Create_tmp_table_priv: Y
      Lock_tables_priv: Y
          Execute_priv: Y
       Repl_slave_priv: Y
      Repl_client_priv: Y
      Create_view_priv: Y
        Show_view_priv: Y
   Create_routine_priv: Y
    Alter_routine_priv: Y
      Create_user_priv: Y
            Event_priv: Y
          Trigger_priv: Y
Create_tablespace_priv: Y
              ssl_type:         --  此处为空
            ssl_cipher:
           x509_issuer:
          x509_subject:
         max_questions: 0
           max_updates: 0
       max_connections: 0
  max_user_connections: 0
                plugin: mysql_native_password
 authentication_string: *23AE809DDACAF96AF0FD78ED04B6A265E05AA257
      password_expired: N
 password_last_changed: 2015-11-26 09:55:31
     password_lifetime: NULL
        account_locked: N
1 row in set (0.00 sec)
```

```
#
# 当前虚拟机2  MyServer2
#
[root@MyServer2 bin]> ./mysql -u burn -h 172.18.14.68 -P3307 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.9-log 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> \s
--------------
./mysql  Ver 14.14 Distrib 5.7.9, for linux-glibc2.5 (x86_64) using  EditLine wrapper

Connection id:          6
Current database:
Current user:           burn@192.168.115.223
SSL:                    Cipher in use is DHE-RSA-AES256-SHA  #已经使用了ssl登录了
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.9-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             172.18.14.68 via TCP/IP
Server characterset:    utf8mb4
Db     characterset:    utf8mb4
Client characterset:    utf8
Conn.  characterset:    utf8
TCP port:               3307
Uptime:                 3 min 6 sec

Threads: 2  Questions: 19  Slow queries: 0  Opens: 109  Flush tables: 1  Open tables: 102  Queries per second avg: 0.102
--------



#
# 当前虚拟机2 MyServer2
# 上面测试中我们没有使用--ssl参数，也是用了ssl登录的，原因如下
#
[root@MyServer2 bin]> ./mysql --help | grep ssl
  --ssl               If set to ON, this option enforces that SSL is
                      server. To disable client SSL capabilities use --ssl=OFF.
                      (Defaults to on; use --skip-ssl to disable.)
                      # 这里说，默认是开启的，可以用--skip-ssl 禁用



#
# 当前虚拟机2 MyServer2
# 禁用ssl登录测试
#
[root@MyServer2 bin]> ./mysql -u burn -h 172.18.14.68 -P3307 -p --skip-ssl  #这里跳过了ssl
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.9-log 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> \s
--------------
./mysql  Ver 14.14 Distrib 5.7.9, for linux-glibc2.5 (x86_64) using  EditLine wrapper

Connection id:          7
Current database:
Current user:           burn@192.168.115.223
SSL:                    Not in use    # 果然就禁用了ssl
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.9-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             172.18.14.68 via TCP/IP
Server characterset:    utf8mb4
Db     characterset:    utf8mb4
Client characterset:    utf8
Conn.  characterset:    utf8
TCP port:               3307
Uptime:                 5 min 50 sec

Threads: 2  Questions: 24  Slow queries: 0  Opens: 109  Flush tables: 1  Open tables: 102  Queries per second avg: 0.068
--------------



--
-- 当前虚拟机1 MyServer, 当前实例mysql1
-- 让用户必须使用ssl
--
mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3307  |
+---------------+-------+
1 row in set (0.00 sec)

mysql> alter user 'burn'@'%' require ssl;
Query OK, 0 rows affected (0.02 sec)



#
# 当前虚拟机2 MyServer2
#
[root@MyServer2 bin]> ./mysql -u burn -h 172.18.14.68 -P3307 -p --skip-ssl
Enter password:
ERROR 1045 (28000): Access denied for user 'burn'@'192.168.115.223' (using password: YES) ## 禁用了SSL就无法登录了
##
[root@MyServer2 bin]> ./mysql -u burn -h 172.18.14.68 -P3307 -p # 默认就启用ssl
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.9-log 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> \s
--------------
./mysql  Ver 14.14 Distrib 5.7.9, for linux-glibc2.5 (x86_64) using  EditLine wrapper

Connection id:          9
Current database:
Current user:           burn@192.168.115.223
SSL:                    Cipher in use is DHE-RSA-AES256-SHA  # 确实启用了
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.9-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             172.18.14.68 via TCP/IP
Server characterset:    utf8mb4
Db     characterset:    utf8mb4
Client characterset:    utf8
Conn.  characterset:    utf8
TCP port:               3307
Uptime:                 14 min 25 sec

Threads: 2  Questions: 32  Slow queries: 0  Opens: 109  Flush tables: 1  Open tables: 102  Queries per second avg: 0.036
--------------
```

2. 开启证书认证(5.7.9)

```
--
-- 当前虚拟机1 MyServer，  当前实例 msyql1
--
mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3307  |
+---------------+-------+
1 row in set (0.00 sec)

mysql> create user 'burn_x509'@'%' identified by '123' require x509; -- 启用证书认证
Query OK, 0 rows affected (0.02 sec)

mysql> grant all on *.* to 'burn'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> select * from mysql.user where user='burn_x509'\G
*************************** 1. row ***************************
                  Host: %
                  User: burn_x509
           Select_priv: N
           Insert_priv: N
           Update_priv: N
           Delete_priv: N
           Create_priv: N
             Drop_priv: N
           Reload_priv: N
         Shutdown_priv: N
          Process_priv: N
             File_priv: N
            Grant_priv: N
       References_priv: N
            Index_priv: N
            Alter_priv: N
          Show_db_priv: N
            Super_priv: N
 Create_tmp_table_priv: N
      Lock_tables_priv: N
          Execute_priv: N
       Repl_slave_priv: N
      Repl_client_priv: N
      Create_view_priv: N
        Show_view_priv: N
   Create_routine_priv: N
    Alter_routine_priv: N
      Create_user_priv: N
            Event_priv: N
          Trigger_priv: N
Create_tablespace_priv: N
              ssl_type: X509    -- 使用X509登录
            ssl_cipher:
           x509_issuer:
          x509_subject:
         max_questions: 0
           max_updates: 0
       max_connections: 0
  max_user_connections: 0
                plugin: mysql_native_password
 authentication_string: *23AE809DDACAF96AF0FD78ED04B6A265E05AA257
      password_expired: N
 password_last_changed: 2015-11-26 10:14:43
     password_lifetime: NULL
        account_locked: N
1 row in set (0.00 sec)


#
# 当前虚拟机2 MyServer2
#
[root@MyServer2 bin]> ./mysql -u burn_x509 -h 172.18.14.68 -P3307 -p
Enter password:
ERROR 1045 (28000): Access denied for user 'burn_x509'@'192.168.115.223' (using password: YES)  # 即使默认开启了ssl，也是无法登录的


#
# 当前虚拟机1 MyServer
#
[root@MyServer data1]> pwd
/data1
[root@MyServer data1]> ll | grep pem
-rw-------. 1 mysql mysql      1675 Nov 25 23:55 ca-key.pem
-rw-r--r--. 1 mysql mysql      1070 Nov 25 23:55 ca.pem
-rw-r--r--. 1 mysql mysql      1078 Nov 25 23:55 client-cert.pem
-rw-------. 1 mysql mysql      1679 Nov 25 23:55 client-key.pem
-rw-------. 1 mysql mysql      1675 Nov 25 23:55 private_key.pem
-rw-r--r--. 1 mysql mysql       451 Nov 25 23:55 public_key.pem
-rw-r--r--. 1 mysql mysql      1078 Nov 25 23:55 server-cert.pem
-rw-------. 1 mysql mysql      1679 Nov 25 23:55 server-key.pem
[root@MyServer data1]> scp client-cert.pem  client-key.pem  root@172.18.14.41:~/
The authenticity of host '172.18.14.41 (172.18.14.41)' can't be established.
RSA key fingerprint is 5f:f5:3c:b0:57:79:8d:50:c6:c8:69:b0:90:6e:98:3b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.18.14.41' (RSA) to the list of known hosts.
root@172.18.14.41's password:
client-cert.pem                                                                                   100% 1078     1.1KB/s   00:00
client-key.pem                                                                                    100% 1679     1.6KB/s   00:00


#
# 当前虚拟机2 MyServer2
#
[root@MyServer2 ~]> ll | grep pem
-rw-r--r--.  1 root root       1078 Nov 26 10:22 client-cert.pem
-rw-------.  1 root root       1679 Nov 26 10:22 client-key.pem

[root@MyServer2 ~]> mysql -u burn_x509 -h 172.18.14.68 -P 3307 -p --ssl-cert=./client-cert.pem  --ssl-key=./client-key.pem
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.9-log 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> \s
--------------
mysql  Ver 14.14 Distrib 5.6.27, for linux-glibc2.5 (x86_64) using  EditLine wrapper

Connection id:          12
Current database:
Current user:           burn_x509@192.168.115.223
SSL:                    Cipher in use is DHE-RSA-AES256-SHA # 使用加密方式登录，且通过证书，因为这个用户 require X509
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.7.9-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             172.18.14.68 via TCP/IP
Server characterset:    utf8mb4
Db     characterset:    utf8mb4
Client characterset:    utf8
Conn.  characterset:    utf8
TCP port:               3307
Uptime:                 32 min 15 sec

Threads: 2  Questions: 41  Slow queries: 0  Opens: 114  Flush tables: 1  Open tables: 107  Queries per second avg: 0.021
--------------
```