Mysql使用SSL连接

发布时间：2020-03-12 来源：未知 点击:64 次

最近项目中用到了SSL连接，记录一下，环境为windows10，Mysql版本为5.6

查看是否支持 SSL

首先在 MySQL 上执行如下命令, 查询是否 MySQL 支持 SSL:

mysql> SHOW VARIABLES LIKE 'have\_ssl';

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

| Variable\_name | Value |

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

| have\_ssl | YES |

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

1 row in set (0.02 sec)

当 have\_ssl 为 YES 时, 表示此时 MySQL 服务已经支持 SSL 了. 如果是 DESABLE, 则需要在启动 MySQL 服务时, 使能 SSL 功能.

使用 OpenSSL 创建 SSL 证书和私钥

* 根据自己的操作系统下载Win(xx)OpenSSL安装
* 新建一个目录用于存放生成的证书和私钥

//生成一个 CA 私钥

openssl genrsa 2048 > cert/ca-key.pem

//使用私钥生成一个新的数字证书,执行这个命令时, 会需要填写一些问题, 随便填写就可以了

openssl req -sha1 -new -x509 -nodes -days 3650 -key ./cert/ca-key.pem > cert/ca-cert.pem

//创建服务器端RSA 私钥和数字证书,这个命令会生成一个新的私钥(server-key.pem), 同时会使用这个新私钥来生成一个证书请求文件(server-req.pem).

//这个命令同样需要回答几个问题, 随便填写即可. 不过需要注意的是, A challenge password 这一项需要为空

openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout cert/server-key.pem > cert/server-req.pem

//将生成的私钥转换为 RSA 私钥文件格式

openssl rsa -in cert/server-key.pem -out cert/server-key.pem

//使用原先生成的 CA 证书来生成一个服务器端的数字证书

openssl x509 -sha1 -req -in cert/server-req.pem -days 3650 -CA cert/ca-cert.pem -CAkey cert/ca-key.pem -set\_serial 01 > cert/server-cert.pem

//创建客户端的 RSA 私钥和数字证书

openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout cert/client-key.pem > cert/client-req.pem

//将生成的私钥转换为 RSA 私钥文件格式

openssl rsa -in cert/client-key.pem -out cert/client-key.pem

//为客户端创建一个数字证书

openssl x509 -sha1 -req -in cert/client-req.pem -days 3650 -CA cert/ca-cert.pem -CAkey cert/ca-key.pem -set\_serial 01 > cert/client-cert.pem

SSL 配置

在前面的步骤中, 我们已经生成了8个文件, 分别是:

* ca-cert.pem: CA 证书, 用于生成服务器端/客户端的数字证书.
* ca-key.pem: CA 私钥, 用于生成服务器端/客户端的数字证书.
* server-key.pem: 服务器端的 RSA 私钥
* server-req.pem: 服务器端的证书请求文件, 用于生成服务器端的数字证书.
* server-cert.pem: 服务器端的数字证书.
* client-key.pem: 客户端的 RSA 私钥
* client-req.pem: 客户端的证书请求文件, 用于生成客户端的数字证书.
* client-cert.pem: 客户端的数字证书.

接下来我们就需要分别配置服务器端和客户端

* 服务器端配置  
  服务器端需要用到三个文件, 分别是: CA 证书, 服务器端的 RSA 私钥, 服务器端的数字证书, 我们需要在 [mysqld] 配置域下添加如下内容:

[mysqld]

ssl-ca=/etc/mysql/ca-cert.pem

ssl-cert=/etc/mysql/server-cert.pem

ssl-key=/etc/mysql/server-key.pem

接着我们还可以更改 bind-address, 使 MySQL 服务可以接收来自所有 ip 地址的客户端, 即:

bind-address = \*

当配置好后, 我们需要重启 MySQL 服务  
最后一步, 我们添加一个需要使用 SSL 才可以登录的帐号, 来验证一下我们所配置的 SSL 是否生效:

GRANT ALL PRIVILEGES ON \*.\* TO 'ssl\_test'@'%' IDENTIFIED BY 'ssl\_test' REQUIRE SSL;

FLUSH PRIVILEGES;

当配置好后, 使用 root 登录 MySQL

mysql --ssl-ca="D:/Program Files/OpenSSL-Win64/bin/cert/ca-cert.pem" --ssl-cert="D:/Program Files/OpenSSL-Win64/bin/cert/client-cert.pem" --ssl-key="D:/Program Files/OpenSSL-Win64/bin/cert/client-key.pem" -u coisini -p

当连接成功后, 我们执行如下指令

\s

执行 show variables like '%ssl%' 语句会有如下输出:

mysql> show variables like '%ssl%';

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

| Variable\_name | Value |

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

| have\_openssl | YES |

| have\_ssl | YES |

| ssl\_ca | ca.pem |

| ssl\_capath | |

| ssl\_cert | server-cert.pem |

| ssl\_cipher | |

| ssl\_crl | |

| ssl\_crlpath | |

| ssl\_key | server-key.pem |

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

9 rows in set (0.01 sec)

JAVA配置

* 使用该命令生成java使用SSL连接所需的文件：

keytool -importcert -alias MySQLCACert -file "D:\Program Files\OpenSSL-Win64\bin\cert\ca-cert.pem" -keystore truststore -storepass 密码

* 将生成的文件配置系统环境变量

名：JAVA\_OPTS

值：-Djavax.net.ssl.trustStore="上一步中生成文件的本地路径" -Djavax.net.ssl.trustStorePassword="密码"

* JDBC配置连接

##jdbc.properties:

yxaq.dz=jdbc:mysql://127.0.0.1:3306/yxaqgl?verifyServerCertificate=true&useSSL=true&requireSSL=true