How to setup a SSL configuration in DB2 and in JDBC toolkit

This document describes a step by step procedure to setup a DB2 SSL configuration and use the trustStore keys in streamsx.jdbc toolkit.

The IBM Db2 database system supports SSL encryption. More details in : https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.admin.sec.doc/doc/t0070301.l

At first, we must create SSL keys and change the DB2 configuration.

Add SSL_SVCENAME (SSL port) to services

Add the SSL service port to the /etc/services .

The SSL_SVCENAME is this example **db2ssl** .

login as root add the following line at the end of file /etc/services .

```
db2ssl    50005/tcp
```

In our example the SSL port is **50005** and the db2 instance owner is **db2inst1** .

```
vi /etc/services
cat /etc/services | grep db2
db2c_db2inst1    50000/tcp
b2j_db2inst1     55000/tcp
db2ssl           50005/tcp
```

//

Global Security Kit

The IBM® Global Security Kit **(GSKit)** supports the use of the SSL protocol to protect DB2® client server communications over the network. login as db2 instance owner and add **gskit** libraries and programs to your path

```
su - db2inst1
```

add the following lines into your db2inst1 (instance owner) .bashrc file.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/db2inst1/sqllib/gskit/bin
```

```
export PATH=$PATH:/home/db2inst1/sqllib/gskit/bin
```

and

```
source ~/.bashrc
```

Create SSL keys and update the db2 configuration parameters.

The following sample describes how to create the SSL keys. You can adapt the shell script with your password and your DN information. Create a shell script in your db2 instance owner (db2inst1) home directory.

vi createSSL.sh

And add the following lines in file createSSL.sh.

```
#!/bin/bash
HOSTNAME=`hostname`
SSL_PORT=50005
SSLPASSWORD="sslPassw0rd"

echo "DB2 server hostname = $HOSTNAME   ssl port=$SSL_PORT ssl password =$SSLPASSWORD"

rm -rf ssl
mkdir ssl
cd ssl
echo "------------------------------------ create keys ------------------------"

gsk8capicmd_64 -keydb -create -db "key.kdb" -pw "$SSLPASSWORD" -stash
gsk8capicmd_64 -cert -create -db "key.kdb" -pw "$SSLPASSWORD" -label "SSLLabel" -dn "CN=$HOST
gsk8capicmd_64 -cert -extract -db "key.kdb" -pw  "$SSLPASSWORD" -label "SSLLabel" -target "ke
ls -al

echo "----------------------- update db2 configuration  ------------------------"


db2 update dbm cfg using SSL_SVR_KEYDB /home/db2inst1/ssl/key.kdb
db2 update dbm cfg using SSL_SVR_STASH /home/db2inst1/ssl/key.sth
db2 update dbm cfg using SSL_SVR_LABEL SSLLabel
db2 update dbm cfg using SSL_SVCENAME db2ssl
db2 update dbm cfg using SSL_VERSIONS TLSV12
db2 get database manager configuration |grep SSL
db2set DB2COMM=SSL,TCPIP

echo "-------------------------- restart db2 ----------------------------------"

db2stop force
db2start

echo "-------------------------- create certificate --------------------------"
openssl s_client -connect $HOSTNAME:$SSL_PORT -servername  $HOSTNAME < /dev/null > cert.txt
cat cert.txt | openssl x509   -trustout > db2cert.pem
echo "-------------------------create keystore via keytool --------------------"

cat db2cert.pem

keytool -noprompt -import -file db2cert.pem -keystore keystore.jks -alias ltsdb2 -storepass $
ls -al keystore.jks
```

Save the file createSSL.sh and make it executable

```
chmod +x createSSL.sh
./createSSL.sh
```

It creates keys, changes the DB2 configuration, **stops the database** and start it again .

//

Check the ports

```
sudo lsof -i |grep db2
db2sysc   16568 db2inst1    8u  IPv4  53077    0t0  TCP *:db2c_db2inst1 (LISTEN)
db2sysc   16568 db2inst1    9u  IPv4  53078    0t0  TCP *:db2ssl (LISTEN)
```

//

Transfer the storkey to your client

copy keystore.jks to your client in opt/ directory of your JDBC project.

copy the latest DB2 jdbc drive jar file **db2jcc4.jar** also into opt/ directory Link for DB2 JDBC drivers http://www-01.ibm.com/support/docview.wss?uid=swg21363866

```
opt/keystore.jks
opt/db2jcc4.jar
```

//

SPL example for JDBCRun operator

Befor you compile this spl file you have to adapt the db2-server-name and the passwords.

```
/**
* This SPL application demonstrates a JDBCRun operator to use SSL keys
* It creates an encrypted connection to DB2 server, using IBM DB2 SSL.
* And print the DB2 instance name
*/

namespace application ;

use com.ibm.streamsx.jdbc::* ;
use com.ibm.streamsx.jdbc.types::* ;

composite JDBCSSl
{
  param
    expression<rstring> $jdbcDriverLib : getSubmissionTimeValue("jdbcDriverLib", "opt/db2jcc4
    expression<rstring> $jdbcClassName : getSubmissionTimeValue("jdbcClassName", "com.ibm.db2
    expression<rstring> $jdbcUrl : getSubmissionTimeValue("jdbcUrl", "jdbc:db2://<your-db2-se
    expression<rstring> $jdbcUser: getSubmissionTimeValue("jdbcUser", "db2inst1" );
    expression<rstring> $jdbcPassword: getSubmissionTimeValue("jdbcPassword", "db2Passw0rd" )
    expression<rstring> $trustStore : getSubmissionTimeValue("trustStore", "opt/keystore.jks"
    expression<rstring> $trustStorePassword : getSubmissionTimeValue("trustStorePassword", "s

  graph
```

```
        stream<int32 count,rstring statement>  createSelect = Custom() {
         logic
          onProcess: {
             for (int32 i in range(5)) {
                 submit({ count=i, statement = "select INST_NAME from SYSIBMADM.ENV_INST_INFO"
                 }
                     // prevent that final punct is sent
                     while (true) {
                     block(1.0);
                 }
             }
         }

    /**
    * The printStatement is a Custom operator
    *  It prints the output createSelect
    */
    () as printStatement = Custom(createSelect)
    {
        logic
            onTuple createSelect : printStringLn("statement " + (rstring)count + " " + statem
    }


    /**
    * The getInstanceName is a JDBCRun operator
    * It connects to the DB2 server via trusted SSL
    * and get the instance name of running DB2 server
    */
    stream<rstring INST_NAME> getInstanceName = JDBCRun(createSelect)
    {
        param
            jdbcDriverLib      : $jdbcDriverLib ;
            jdbcClassName      : $jdbcClassName ;
            jdbcUrl            : $jdbcUrl ;
            jdbcUser           : $jdbcUser ;
            jdbcPassword       : $jdbcPassword ;
            statement          : "select INST_NAME from SYSIBMADM.ENV_INST_INFO" ;
            sslConnection      : true ;

            // trustStore specifies the path to the trustStore key.
            trustStore         : $trustStore ;

            // trustStorePassword specifies the password for the trustStore given by the trus
            // The sslConnection parameter must be set to true.
            trustStorePassword : $trustStorePassword;
    }

    /**
    * The printInstanceName is a Custom operator
    * It prints the SQL results from getInstanceName
    */
    () as printInstanceName = Custom(getInstanceName)
    {
        logic
            onTuple getInstanceName : printStringLn("DB2 Insatnce Name " + INST_NAME) ;
    }

}
```