===================================

STEPS TO GENERATE TEST CERTIFICATES

===================================

1. CA key and certficate

========================

(Generate the CA key)

$ openssl genrsa -out ca-key.pem 2048

(Generate a self-signed certificate for the CA)

$ openssl req -new -x509 -nodes -sha256 -days 3650 -key ca-key.pem -out ca-cert.pem

(...)

Country Name (2 letter code) []:US

State or Province Name (full name) []:California

Locality Name (eg, city) []:Redwood Shores

Organization Name (eg, company) []:Oracle

Organizational Unit Name (eg, section) []:MySQL

Common Name (e.g. server FQDN or YOUR name) []:MySQL Connector/J CA

Email Address []:mysql@oracle.com

2. Server key and certificate

=============================

(Generate the server key)

$ openssl genrsa -out server-key.pem 2048

(Generate a certificate signing request for the server)

$ openssl req -new -key server-key.pem -out server-csr.pem

(Or, if the Subject Alternative Name field is required for testing - \*used together with below sign command\*)

$ openssl req -new -key server-key.pem -out server-csr.pem \

-reqexts SAN \

-config <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\n\nsubjectAltName=@alt\_names\n\n[alt\_names]\nDNS.1=bug99767.mysql.san1.tst\nDNS.2=\*.mysql.san2.tst\nDNS.3=bug\*.mysql.san3.tst\nDNS.4=\*99767.mysql.san4.tst\nDNS.5=bug99767.\*.san5.tst\nDNS.6=bug99767.\*\nDNS.7=\*\nIP.1=9.9.7.67\nIP.2=99.7.6.7"))

(...)

Country Name (2 letter code) []:US

State or Province Name (full name) []:California

Locality Name (eg, city) []:Redwood Shores

Organization Name (eg, company) []:Oracle

Organizational Unit Name (eg, section) []:MySQL

Common Name (e.g. server FQDN or YOUR name) []:MySQL Connector/J Server

Email Address []:mysql@oracle.com

(...)

A challenge password []:

An optional company name []:

(Sign the server certificate signing request)

$ openssl x509 -req -in server-csr.pem -CA ca-cert.pem -CAkey ca-key.pem -set\_serial 01 -days 3650 -sha256 -out server-cert.pem

(Or, if the Subject Alternative Name field is required for testing - \*used together with above request command\*)

$ openssl x509 -req -in server-csr.pem -CA ca-cert.pem -CAkey ca-key.pem -set\_serial 01 -days 3650 -sha256 -out server-cert.pem \

-extfile <(printf "subjectAltName=@alt\_names\n\n[alt\_names]\nDNS.1=bug99767.mysql.san1.tst\nDNS.2=\*.mysql.san2.tst\nDNS.3=bug\*.mysql.san3.tst\nDNS.4=\*99767.mysql.san4.tst\nDNS.5=bug99767.\*.san5.tst\nDNS.6=bug99767.\*\nDNS.7=\*\nIP.1=9.9.7.67\nIP.2=99.7.6.7")

(OPTIONAL: Delete the certificate signing request file)

$ rm server-csr.pem

(OPTIONAL: Verify the server certificate)

$ openssl verify -CAfile ca-cert.pem server-cert.pem

3. Client key and certificate

=============================

(Generate the client key)

$ openssl genrsa -out client-key.pem 2048

(Generate a certificate signing request for the client)

$ openssl req -new -key client-key.pem -out client-csr.pem

(...)

Country Name (2 letter code) []:US

State or Province Name (full name) []:California

Locality Name (eg, city) []:Redwood Shores

Organization Name (eg, company) []:Oracle

Organizational Unit Name (eg, section) []:MySQL

Common Name (e.g. server FQDN or YOUR name) []:MySQL Connector/J Client

Email Address []:mysql@oracle.com

(...)

A challenge password []:

An optional company name []:

(Sign the client certificate signing request)

$ openssl x509 -req -in client-csr.pem -CA ca-cert.pem -CAkey ca-key.pem -set\_serial 01 -days 3650 -sha256 -out client-cert.pem

(OPTIONAL: Delete the certificate signing request file)

$ rm client-csr.pem

(OPTIONAL: Verify the client certificate)

$ openssl verify -CAfile ca-cert.pem client-cert.pem

4. CA truststore

================

(Create a truststore containing the CA certificate)

$ keytool -importcert -alias mysqlcacert -file ca-cert.pem -keystore ca-truststore -storepass password

Trust this certificate? [no]: yes

(OPTIONAL: List the contents of the truststore)

$ keytool -list -keystore ca-truststore -storepass password

5. Client key and certificate keystore

======================================

(Convert client key to pkcs12 format)

$ openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -name "mysqlclient" -passout pass:password -out client-keystore.p12

(Create a keystore containing the client key)

$ keytool -importkeystore -srckeystore client-keystore.p12 -srcstoretype pkcs12 -srcstorepass password -destkeystore client-keystore -deststoretype JKS -deststorepass password

(OPTIONAL: Delete the client key in pkcs12 format)

$ rm client-keystore.p12

(OPTIONAL: List the contents of the client keystore)

$ keytool -list -keystore client-keystore -storepass password

==========================

RUN SERVER WITH TEST CERTS

==========================

Add to my.conf:

[mysqld]

ssl-key = "/path/server-key.pem"

ssl-cert = "/path/server-cert.pem"

ssl-ca = "/path/ca-cert.pem"