# SSL configuration with self signed certificate with loopback connection setup

**Subtasks :**
1. Create a keystore using GSKit.
2. Create a self-signed certificate.
3. Extract the certificate for distribution to clients.
4. Configure TLS support for the Db2 server.

```
---------------------------------------------------------
```

**Task 1 : Creating a keystore with GSKit**

```
---------------------------------------------------------
[db2inst1@server1 ssl]$ db2 get dbm cfg | grep -i ssl
SSL server keydb file                (SSL_SVR_KEYDB) =
SSL server stash file                (SSL_SVR_STASH) =
SSL server certificate label          (SSL_SVR_LABEL) =
SSL service name                     (SSL_SVCENAME) =-
SSL cipher specs                  (SSL_CIPHERSPECS) =
SSL versions                      (SSL_VERSIONS) =
SSL client keydb file               (SSL_CLNT_KEYDB) =
SSL client stash file              (SSL_CLNT_STASH) =

[db2inst1@server1 ssl]$ hostname
server1.fyre.ibm.com
[db2inst1@server1 ssl]$ gsk8capicmd_64 -keydb -create -db server.p12 -pw myServerPassw0rdpw0 -stash -pqc false
[db2inst1@server1 ssl]$
***************************************************************************
****************************************************************
----------------------------------------------------------------------
```

**Task2 : Creating a self-signed certificate with GSKit**

```
----------------------------------------------------------------------
[db2inst1@server1 ssl]$ gsk8capicmd_64 -cert -create -db server.p12 -stashed -label myselfsigned -dn "CN=server1.fyre.ibm.com" -size 2048 -sigalg SHA256_WITH_RSA
[db2inst1@server1 ssl]$
[db2inst1@server1 ssl]$ ls -ltr
total 8
-rw------- 1 db2inst1 db2iadm1  193 Feb 21 06:14 server.sth
-rw------- 1 db2inst1 db2iadm1 2948 Feb 21 06:16 server.p12
[db2inst1@server1 ssl]$

***************************************************************************
*****************************************************************
-------------------------------------------------------------------------
```

**Task3 : Distributing a self-signed certificate to your Db2 clients**

```
-------------------------------------------------------------------------
[db2inst1@server1 ssl]$ gsk8capicmd_64 -cert -extract -db server.p12 -stashed -label myselfsigned -target myselfsigned.crt -format ascii
[db2inst1@server1 ssl]$
```

```
[db2inst1@server1 ssl]$ ls -ltr
total 12
-rw------- 1 db2inst1 db2iadm1  193 Feb 21 06:14 server.sth
-rw------- 1 db2inst1 db2iadm1 2948 Feb 21 06:16 server.p12
-rw-r--r-- 1 db2inst1 db2iadm1 1099 Feb 21 06:19 myselfsigned.crt

[db2inst1@server1 ssl]$ cat myselfsigned.crt
-----BEGIN CERTIFICATE-----
MIIC/jCCAeagAwIBAgIIdPs8YffaDsAwDQYJKoZIhvcNAQELBQAwHTEbMBkGA1UE
AxMSa2VtcDEuZnlyZS5pYm0uY29tMB4XDTI0MDIyMDE0MTY0NFoXDTI1MDIyMDE0
MTY0NFowHTEbMBkGA1UEAxMSa2VtcDEuZnlyZS5pYm0uY29tMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxkcSHZuWuCI+0ukpdR2DmilXVRji7mM+AJfk
DFF3k4ChmmPuEUF8XVerj+PeynuW5VR6IbobUOKEdmwDZ0y0cef0d9Gg/T40I+SE
USQn+CUKZfrF8l8XbljAKlBFBHvdmhTtnCvDFaddkSlMZS55AprtDUbaYiYwjywu
kFJfYwA2aGnwvi3XUcw/9eJet0uxHJenFdWJpD4214Bm/K3otpjZZD8BcWsm9MCt
X5VlUrz24I8MqYgu62oC9kiVLEbyKzfkj/uCACko00d/G/Fjf1ydwifUJk2ys6C7
oSGbfK2ljjNdTkPrt21fNvtV8sL8J1NiRDH+5/RoAmwPNKT6IwIDAQABo0IwQDAd
BgNVHQ4EFgQUNBmAA6F29aLkvCy/gW7lYwveBhIwHwYDVR0jBBgwFoAUNBmAA6F2
9aLkvCy/gW7lYwveBhIwDQYJKoZIhvcNAQELBQADggEBABBKt3fMRpC36qb3EbKz
HU8fWHX2/aOczL9iwpOJDiP0zpTpjcByF8j5xRhurtIjSY432WkqnB1xMRxlDoH6
d41Dj/SrHag5OXeEW6ndESwbBQqn4AkWhYK9Gxn6xod0S2wAnpbsRK1mngPIBh48
WnDZJMkmfepzLlhGh8RBh/e47P7aHKQpJ1YIER9BQhKYwG/SjM0lndDkna2C4beh
GoM506MI/aCGrsvpf5oSM8P6Lk1CFHJAn803qtkNFSFrk675fmJUYvfHEFtf7cjH
U5F8j3nQqXLp+ZBhAuUs4rSBtvx0z+nBYUv08d6lur9J+z14s1KcsY6XNeYXr0KF
DRQ=
-----END CERTIFICATE-----
```

***************************************************************************
****************************************************

------------------------------------------------------------

**Task 4 : Configuring TLS support on a Db2 server**

------------------------------------------------------------

```
[db2inst1@server1 ssl]$ db2 update dbm cfg using SSL_SVR_KEYDB /home/db2inst1/ssl/server.p12
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[db2inst1@server1 ssl]$ db2 update dbm cfg using SSL_SVR_STASH /home/db2inst1/ssl/server.sth
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[db2inst1@server1 ssl]$ db2 update dbm cfg using SSL_SVR_LABEL myselfsigned
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[db2inst1@server1 ssl]$ db2 update dbm cfg using SSL_SVCENAME 20026
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[db2inst1@server1 ssl]$
[db2inst1@server1 ssl]$ db2 update dbm cfg using SSL_VERSIONS TLSV12,TLSV13
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
```

successfully.

```
====================

SSL server keydb file              (SSL_SVR_KEYDB) = /home/db2inst1/ssl/server.p12
SSL server stash file              (SSL_SVR_STASH) = /home/db2inst1/ssl/server.sth
SSL server certificate label       (SSL_SVR_LABEL) = myselfsigned
SSL service name                   (SSL_SVCENAME) = 20026
SSL cipher specs                   (SSL_CIPHERSPECS) =
SSL versions                       (SSL_VERSIONS) = TLSV12,TLSV13
SSL client keydb file              (SSL_CLNT_KEYDB) =
SSL client stash file              (SSL_CLNT_STASH) =

======================
```

Verified port

```
[db2inst1@server1 ssl]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026           0.0.0.0:*               LISTEN
```

****************************************************************************
***************************************************
--------------------------------------------------

**Configuring client**

--------------------------------------------------

https://www.ibm.com/docs/en/db2/11.5?topic=clients-configuring-tls-using-keystore-non-java-client

Copied certificate to other path

```
[db2inst1@server1 ssl]$ cp myselfsigned.crt /home/db2inst1/client
[db2inst1@server1 ssl]$
```

****************************

Create Keystore and add certificate

```
[db2inst1@server1 client]$ gsk8capicmd_64 -keydb -create -db "client.p12" -pw "myClientPassw0rdpw0" -stash
[db2inst1@server1 client]$
[db2inst1@server1 client]$ gsk8capicmd_64 -cert -add -db "client.p12" -stashed -label "myServerCert" -file "myselfsigned.crt" -format ascii
[db2inst1@server1 client]$
[db2inst1@server1 client]$
[db2inst1@server1 client]$ ls -ltr
total 12
-rw-r--r-- 1 db2inst1 db2iadm1 1099 Feb 21 06:36 myselfsigned.crt
-rw------- 1 db2inst1 db2iadm1  193 Feb 21 06:38 client.sth
-rw------- 1 db2inst1 db2iadm1 1334 Feb 21 06:38 client.p12
[db2inst1@server1 client]$
[db2inst1@server1 client]$ pwd
/home/db2inst1/client
```

```
---------------------------------------------------
Updating client parameters
----------------------------------------------------
[db2inst1@server1 client]$ db2 update dbm cfg using SSL_CLNT_KEYDB /home/db2inst1/client/client.p12
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
[db2inst1@server1 client]$
[db2inst1@server1 client]$ db2 update dbm cfg using SSL_CLNT_STASH /home/db2inst1/client/client.sth
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.

*****************************
[db2inst1@server1 client]$ db2 get dbm cfg | grep -i ssl
 SSL server keydb file                   (SSL_SVR_KEYDB) = /home/db2inst1/ssl/server.p12
 SSL server stash file                   (SSL_SVR_STASH) = /home/db2inst1/ssl/server.sth
 SSL server certificate label            (SSL_SVR_LABEL) = myselfsigned
 SSL service name                         (SSL_SVCENAME) = 20026
 SSL cipher specs                      (SSL_CIPHERSPECS) =
 SSL versions                             (SSL_VERSIONS) = TLSV12,TLSV13
 SSL client keydb file                  (SSL_CLNT_KEYDB) = /home/db2inst1/client/client.p12
 SSL client stash file                  (SSL_CLNT_STASH) = /home/db2inst1/client/client.sth
```

```
[db2inst1@server1 client]$ gsk8capicmd_64 -cert -details -label "myselfsigned" -db /home/db2inst1/ssl/server.p12 -stashed
Label : myselfsigned
Key Size : 2048
Version : X509 V3
Serial : 74fb3c61f7da0ec0
Issuer : CN=server1.fyre.ibm.com
Subject : CN=server1.fyre.ibm.com
Not Before : February 20, 2024 6:16:44 AM PST

Not After : February 20, 2025 6:16:44 AM PST

Public Key
30 82 01 22 30 0D 06 09 2A 86 48 86 F7 0D 01 01
01 05 00 03 82 01 0F 00 30 82 01 0A 02 82 01 01
```

```
[db2inst1@server1 client]$ gsk8capicmd_64 -cert -list -db /home/db2inst1/client/client.p12 -stashed
Certificates found
* default, - personal, ! trusted, # secret key
!       myServerCert
```

************************************************************

**Configuring loopback connection:**

```
db2 catalog tcpip node sslnode remote 1.22.33.456 server 20026 security ssl
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[db2inst1@server1 client]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.

[db2inst1@server1 client]$ db2 catalog db newsamp as sslsamp
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[db2inst1@server1 client]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.

[db2inst1@server1 client]$ db2 uncatalog db newsamp
DB20000I  The UNCATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[db2inst1@server1 client]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.

[db2inst1@server1 client]$ db2 catalog db sslsamp as newsamp at node sslnode
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[db2inst1@server1 client]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.

[db2inst1@server1 client]$ db2stop
02/21/2024 06:59:02     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
[db2inst1@server1 client]$ db2start
02/21/2024 06:59:24     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
```

```
Database 3 entry:

Database alias                       = NEWSAMP
Database name                        = SSLSAMP
Node name                            = SSLNODE
Database release level               = 15.00
Comment                              =
Directory entry type                 = Remote
Catalog database partition number    = -1
Alternate server hostname            =
Alternate server port number         =
Database 5 entry:

Database alias                       = SSLSAMP
Database name                        = NEWSAMP
Local database directory             = /home/db2inst1
Database release level               = 15.00
Comment                              =
Directory entry type                 = Indirect
Catalog database partition number    = 0
Alternate server hostname            =
Alternate server port number         =


***************************************

db2 list node directory

Node Directory

Number of entries in the directory = 1

Node 1 entry:

Node name                      = SSLNODE
Comment                        =
Directory entry type           = LOCAL
Protocol                       = TCPIP
Hostname                       = 1.22.33.456
Service name                   = 20026
Security type                  = SSL

**************************

[db2inst1@server1 client]$ db2set -all
[i] DB2COMM=SSL,TCPIP
[g] DB2SYSTEM=server1.fyre.ibm.com

**************************

[db2inst1@server1 client]$ db2 connect to SSLSAMP

Database Connection Information

Database server        = DB2/LINUXX8664 11.5.8.0
SQL authorization ID   = DB2INST1
Local database alias   = SSLSAMP

[db2inst1@server1 client]$
[db2inst1@server1 client]$ db2 terminate
```

```
DB20000I  The TERMINATE command completed successfully.

[db2inst1@server1 client]$ db2 connect to NEWSAMP
SQL30082N  Security processing failed with reason "3" ("PASSWORD MISSING").
SQLSTATE=08001
[db2inst1@server1 client]$
[db2inst1@server1 client]$ db2 connect to NEWSAMP user db2inst1
Enter current password for db2inst1:

Database Connection Information

Database server        = DB2/LINUXX8664 11.5.8.0
SQL authorization ID   = DB2INST1
Local database alias   = NEWSAMP

[db2inst1@server1 client]$
[db2inst1@server1 client]$
[db2inst1@server1 client]$
[db2inst1@server1 client]$ db2 list applications

Auth Id  Application    Appl.      Application Id                                 DB       # of
Name          Handle                                              Name    Agents
-------- ------------- ---------- -------------------------------------------------------------- -------- -----
DB2INST1 db2bp        24        1.22.33.456.55520.240221150038                        NEWSAMP
1


[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026          0.0.0.0:*              LISTEN
tcp        0      0 1.22.33.456:20026      1.22.33.456:55520      ESTABLISHED
tcp        0      0 1.22.33.456:55520      1.22.33.456:20026      ESTABLISHED
```

***************************

After termination , No connection , and Port again came in Listen state which was in ESTABLISHED state.

```
[db2inst1@server1 client]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.
[db2inst1@server1 client]$
[db2inst1@server1 client]$ db2 list applications
SQL1611W  No data was returned by Database System Monitor.
[db2inst1@server1 client]$
[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026          0.0.0.0:*              LISTEN
tcp        0      0 1.22.33.456:20026      1.22.33.456:55520      TIME_WAIT
tcp        0      0 1.22.33.456:55566      1.22.33.456:20026      TIME_WAIT
tcp        0      0 1.22.33.456:55568      1.22.33.456:20026      TIME_WAIT
[db2inst1@server1 client]$
[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026          0.0.0.0:*              LISTEN
tcp        0      0 1.22.33.456:20026       1.22.33.456:55520       TIME_WAIT
[db2inst1@server1 client]$
[db2inst1@server1 client]$
[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026           0.0.0.0:*               LISTEN
tcp        0      0 1.22.33.456:20026       1.22.33.456:55520       TIME_WAIT
[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026           0.0.0.0:*               LISTEN
```

***********************************************

One more test with setting db2comm to SSL only , Removed TCPIP

```
[db2inst1@server1 client]$ db2set -all
[i] DB2COMM=SSL

[db2inst1@server1 client]$ db2 connect to SSLSAMP

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.8.0
 SQL authorization ID   = DB2INST1
 Local database alias   = SSLSAMP

[db2inst1@server1 client]$ db2 list applications

Auth Id  Application    Appl.      Application Id                                                 DB       # of
         Name           Handle                                                                    Name     Agents
-------- -------------- ---------- -------------------------------------------------------------- -------- -----
DB2INST1 db2bp          7          *LOCAL.db2inst1.240221152125                                   NEWSAMP
1

[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp        0      0 0.0.0.0:20026           0.0.0.0:*               LISTEN

[db2inst1@server1 client]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.
```

```
[db2inst1@server1 client]$ db2 connect to NEWSAMP user db2inst1
Enter current password for db2inst1:

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.8.0
 SQL authorization ID   = DB2INST1
 Local database alias   = NEWSAMP

[db2inst1@server1 client]$ db2 list applications
```

```
Auth Id  Application    Appl.      Application Id                                     DB       # of
Name          Handle                                                    Name    Agents
-------- -------------- ---------- -------------------------------------------------------------- -------- -----
DB2INST1 db2bp         24        1.22.33.456.55756.240221152306                          NEWSAMP
1

[db2inst1@server1 client]$ netstat -an | grep -i 20026
tcp       0      0 0.0.0.0:20026          0.0.0.0:*              LISTEN
tcp       0      0 1.22.33.456:55756      1.22.33.456:20026      ESTABLISHED
tcp       0      0 1.22.33.456:20026      1.22.33.456:55756      ESTABLISHED
```

Other method:
https://www.ibm.com/support/pages/how-check-if-client-connecting-db2-ssl-or-not

# SSL configuration with CA signed certificate

**We have total 5 tasks here :**

1. Create a keystore using GSKit.
2. Create a certificate signing request (CSR).
3. Add root and intermediate certificates to your keystore.
4. Pull your CA-signed certificate into your keystore.
5. Configure TLS support for the Db2 server.

DB2 installation is being initialized.

```
[testuser@server1 ~]$ db2 get dbm cfg | grep -i ssl
 SSL server keydb file                (SSL_SVR_KEYDB) =
 SSL server stash file               (SSL_SVR_STASH) =
 SSL server certificate label          (SSL_SVR_LABEL) =
 SSL service name                    (SSL_SVCENAME) =
 SSL cipher specs                  (SSL_CIPHERSPECS) =
 SSL versions                      (SSL_VERSIONS) =
 SSL client keydb file               (SSL_CLNT_KEYDB) =
 SSL client stash file              (SSL_CLNT_STASH) =
```

----------------------------------------------------

**Step 1 : Create a keystore using GSKit**

----------------------------------------------------

```
[testuser@server1 ssl]$ gsk8capicmd_64 -keydb -create -db "db2_ssl_keydb.kdb" -pw
"myServerPassw0rdpw0" -type cms -stash
[testuser@server1 ssl]$
[testuser@server1 ssl]$
[testuser@server1 ssl]$ ls -ltr
total 16
-rw------- 1 testuser testuser  88 Feb 29 22:17 db2_ssl_keydb.rdb
-rw------- 1 testuser testuser  88 Feb 29 22:17 db2_ssl_keydb.kdb
-rw------- 1 testuser testuser  88 Feb 29 22:17 db2_ssl_keydb.crl
-rw------- 1 testuser testuser 193 Feb 29 22:17 db2_ssl_keydb.sth
[testuser@server1 ssl]$
```

----------------------------------------------------

**Step 2: Create a certificate signing request (CSR)**

----------------------------------------------------

```
[testuser@server1 ssl]$ gsk8capicmd_64 -certreq -create -db "db2_ssl_keydb.kdb" -pw
"myServerPassw0rdpw0" -label "IBM_CA_signed" -dn "CN=1.22.33.456, O=IBM, L=SJ, ST=CA , C=US "
-file db2_ssl_ibmca_certreq.csr -size 2048 -sigalg SHA512WithRSA
[testuser@server1 ssl]$
[testuser@server1 ssl]$
[testuser@server1 ssl]$ ls -ltr
total 24
-rw------- 1 testuser testuser   88 Feb 29 22:17 db2_ssl_keydb.kdb
```

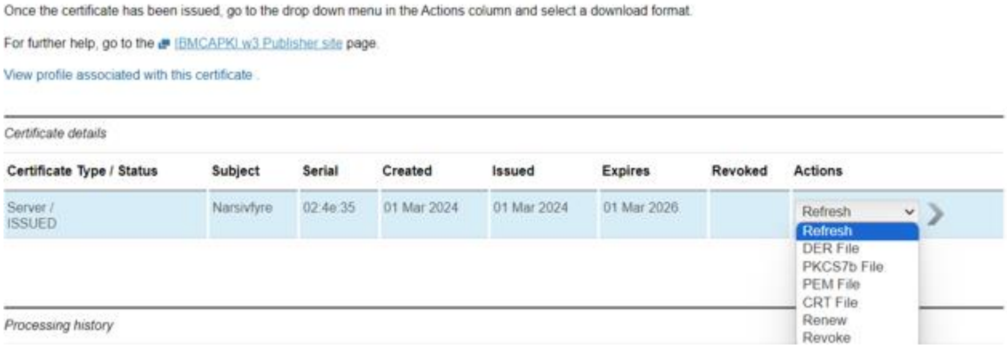
```
-rw------- 1 testuser testuser   88 Feb 29 22:17 db2_ssl_keydb.crl
-rw------- 1 testuser testuser  193 Feb 29 22:17 db2_ssl_keydb.sth
-rw-rw-r-- 1 testuser testuser  976 Feb 29 22:17 db2_ssl_ibmca_certreq.csr
-rw------- 1 testuser testuser 5088 Feb 29 22:17 db2_ssl_keydb.rdb
[testuser@server1 ssl]$
[testuser@server1 ssl]$ cat db2_ssl_ibmca_certreq.csr
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICkjCCAXoCAQAwTTEMMAoGA1UEBhMDVVMgMQwwCgYDVQQIEwNDQSAxCzAJBgNV
BAcTAlNKMQwwCgYDVQQKEwNJQk0xFDASBgNVBAMTCzkuMzAuMjUuMTcyMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArSEb96k8zbyXEFFv/UgodQq1z7Ji
2VUu9QQQ10PjfDNJ4vWktk9l3xK8Vg+NWCDI+C/HwGOABmKNgygbj2PPQM+2m/Ke
6ho2QMDr1mFRcdQH/k/bXq5uA2qWHzH0n9PHTuneKZdBCSdE5DNhK0l/8iBLhmec
SguxqYwPjWcawbVT+K/ns908rXVO07gMgzjt/ex6WuntUxSkgY/aQ33bOkn9JLi0
/fi5yR/mmJTo3jo3FeB6R3Sjbzmqt0FiAsLk/cStXiFUaDvvR1Kb6nuHbqKGV73r
ynEMeJA80WkrcyNh9wsiyWxCTQVLupKetUc7x8H4Oy7mt8ex2yEdihVtAwIDAQAB
oAAwDQYJKoZIhvcNAQENBQADggEBABpstsPQ5OpHGdB2XN3RFVS3PepOXWcKXXOm
iHBElQ0XbeiJx15TQBhzoRNH9jD1Nj5E57l09deRPsaz0jN4NGJDz7W0oJS571Hc
U93WTHRqm0XEqXYvnwhyGas7maAKLBx7HJ4JFB7fDLPn6uCrCj8X8eVmn5MFhR0U
YQlbsGSY0mqV0gwxdCai8bmEKRNpHh3gyjj0V3pmaXudnhKNC9e6QfJoDYxS+7Ep
Kyj2qPhACIC/ygxUWQkehHCqWc6g1c3iCiUhh/wYGrqfFbiTYhpq9aJi6t3SUvWE
P98CiqpEmcAO8m+P0baQVzExvYLF8Luyl1EnqhtpwA2jIECiLjA=
-----END NEW CERTIFICATE REQUEST-----
```

Copied content of csr file in CA site

Here we required 3 files
1) .crt
2) root
3) intermediate

Below are sample steps to download certificate, It will vary depending on the CA.

Select Certificates tab → select Server label from the list.
On the next screen select “CRT File” from Action dropdown and click on arrow “>” to download it.



Once the certificate has been issued, go to the drop down menu in the Actions column and select a download format.

For further help, go to the IBMCAPKI w3 Publisher site page.

View profile associated with this certificate .

Certificate details

| Certificate Type / Status | Subject | Serial | Created | Issued | Expires | Revoked | Actions |
|---|---|---|---|---|---|---|---|
| Server /<br>ISSUED | Narsivfyre | 02:4e:35 | 01 Mar 2024 | 01 Mar 2024 | 01 Mar 2026 |  | Refresh |

Processing history

Copied all 3 files cert.crt , carootcert.der and caintermediatecert.der

```
[testuser@server1 ssl]$ ls -ltr
```

```
total 36
-rw------- 1 testuser testuser   88 Feb 29 22:17 db2_ssl_keydb.kdb
-rw------- 1 testuser testuser   88 Feb 29 22:17 db2_ssl_keydb.crl
-rw------- 1 testuser testuser  193 Feb 29 22:17 db2_ssl_keydb.sth
-rw-rw-r-- 1 testuser testuser  976 Feb 29 22:17 db2_ssl_ibmca_certreq.csr
-rw------- 1 testuser testuser 5088 Feb 29 22:17 db2_ssl_keydb.rdb
-rwxr-xr-x 1 testuser testuser 1289 Feb 29 22:22 cert.crt
-rwxr-xr-x 1 testuser testuser 1001 Feb 29 22:25 carootcert.der
-rwxr-xr-x 1 testuser testuser 1294 Feb 29 22:31 caintermediatecert.der
[testuser@server1 ssl]$
```

--------------------------------------------------------------------------------

**Step 3 : Add root and intermediate certificates to your keystore.**

--------------------------------------------------------------------------------

https://www.ibm.com/docs/en/db2/11.5?topic=certificate-adding-root-intermediate-certificates

```
[testuser@server1 ssl]$ gsk8capicmd_64 -cert -add -db db2_ssl_keydb.kdb -stashed -file
carootcert.der -label MyRootCA
[testuser@server1 ssl]$
[testuser@server1 ssl]$ gsk8capicmd_64 -cert -add -db db2_ssl_keydb.kdb -stashed -file
caintermediatecert.der -label MyIntermediateCA
[testuser@server1 ssl]$
[testuser@server1 ssl]$ ls -ltr
total 44
-rw------- 1 testuser testuser    88 Feb 29 22:17 db2_ssl_keydb.crl
-rw------- 1 testuser testuser   193 Feb 29 22:17 db2_ssl_keydb.sth
-rw-rw-r-- 1 testuser testuser   976 Feb 29 22:17 db2_ssl_ibmca_certreq.csr
-rw------- 1 testuser testuser  5088 Feb 29 22:17 db2_ssl_keydb.rdb
-rwxr-xr-x 1 testuser testuser  1289 Feb 29 22:22 cert.crt
-rwxr-xr-x 1 testuser testuser  1001 Feb 29 22:25 carootcert.der
-rwxr-xr-x 1 testuser testuser  1294 Feb 29 22:31 caintermediatecert.der
-rw------- 1 testuser testuser 10088 Feb 29 22:42 db2_ssl_keydb.kdb
```

----------------------------------------------------------------

**Step 4 : Pull your CA-signed certificate into your keystore.**

----------------------------------------------------------------

https://www.ibm.com/docs/en/db2/11.5?topic=certificate-pulling-ca-signed-into-keystore

```
[testuser@server1 ssl]$ gsk8capicmd_64 -cert -receive -db db2_ssl_keydb.kdb -stashed -file cert.crt
[testuser@server1 ssl]$
[testuser@server1 ssl]$ ls -ltr
total 44
-rw------- 1 testuser testuser    88 Feb 29 22:17 db2_ssl_keydb.crl
-rw------- 1 testuser testuser   193 Feb 29 22:17 db2_ssl_keydb.sth
-rw-rw-r-- 1 testuser testuser   976 Feb 29 22:17 db2_ssl_ibmca_certreq.csr
-rwxr-xr-x 1 testuser testuser  1289 Feb 29 22:22 cert.crt
-rwxr-xr-x 1 testuser testuser  1001 Feb 29 22:25 carootcert.der
-rwxr-xr-x 1 testuser testuser  1294 Feb 29 22:31 caintermediatecert.der
-rw------- 1 testuser testuser 15088 Feb 29 22:43 db2_ssl_keydb.kdb
```

```
-rw------- 1 testuser testuser    88 Feb 29 22:43 db2_ssl_keydb.rdb
[testuser@server1 ssl]$
```

If we notice  .kdb timestamp is changing after add and receive command

Certificate list , We could see all 3 labels here :

```
[testuser@server1 ssl]$ gsk8capicmd_64 -cert -list -db /home/testuser/ssl/db2_ssl_keydb.kdb -
stashed
Certificates found
* default, - personal, ! trusted, # secret key
!       MyRootCA
!       MyIntermediateCA
-       IBM_CA_signed
```

---------------------------------------------------------------------

**Step 5 :Configure TLS support for the Db2 server.**

---------------------------------------------------------------------

```
[testuser@server1 ssl]$ db2 update dbm cfg using SSL_SVR_KEYDB
/home/testuser/ssl/db2_ssl_keydb.kdb
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[testuser@server1 ssl]$ db2 update dbm cfg using SSL_SVR_STASH
/home/testuser/ssl/db2_ssl_keydb.sth
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[testuser@server1 ssl]$ db2 update dbm cfg using SSL_SVR_LABEL IBM_CA_signed
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[testuser@server1 ssl]$ db2 update dbm cfg using SSL_SVCENAME  20033
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[testuser@server1 ssl]$ db2 update dbm cfg using SSL_VERSIONS TLSV12,TLSV13
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[testuser@server1 ssl]$ db2set DB2COMM=SSL,TCPIP
```

```
[testuser@server1 DIAG0000]$ db2 get dbm cfg | grep -i ssl
 SSL server keydb file                   (SSL_SVR_KEYDB) = /home/testuser/ssl/db2_ssl_keydb.kdb
 SSL server stash file                   (SSL_SVR_STASH) = /home/testuser/ssl/db2_ssl_keydb.sth
 SSL server certificate label            (SSL_SVR_LABEL) = IBM_CA_signed
 SSL service name                         (SSL_SVCENAME) = 20033
 SSL cipher specs                      (SSL_CIPHERSPECS) =
 SSL versions                             (SSL_VERSIONS) = TLSV12,TLSV13
 SSL client keydb file                  (SSL_CLNT_KEYDB) =
 SSL client stash file                  (SSL_CLNT_STASH) =
[testuser@server1 DIAG0000]$
```

```
[testuser@server1 DIAG0000]$ db2set -all
[i] DB2COMM=SSL,TCPIP
[g] DB2SYSTEM=server1.fyre.ibm.com
```

So Here SSL configuration at Server side is completed ;

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

## Client configuration :

Created new directory with clienSSL and copied certifcates, We require IBM CA Root Certificate (carootcert.der).

```
[testuser@server1 ssl]$ cp carootcert.der /home/testuser/clientssl
[testuser@server1 ssl]$

[testuser@server1 clientssl]$ ls -ltr
total 12
-rwxr-xr-x 1 testuser testuser 1001 Feb 29 23:33 carootcert.der
```

------------------------------------------------------------------------

**Step1:GSKit to create a key database**

------------------------------------------------------------------------

```
[testuser@server1 clientssl]$ gsk8capicmd_64 -keydb -create -db
"/home/testuser/clientssl/ibmca.kdb" -pw "myServerPassw0rdpw0" -stash
[testuser@server1 clientssl]$
[testuser@server1 clientssl]$ ls -ltr
total 28
-rwxr-xr-x 1 testuser testuser 1001 Feb 29 23:33 carootcert.der
-rw------- 1 testuser testuser   88 Feb 29 23:34 ibmca.kdb
-rw------- 1 testuser testuser   88 Feb 29 23:34 ibmca.rdb
-rw------- 1 testuser testuser   88 Feb 29 23:34 ibmca.crl
-rw------- 1 testuser testuser  193 Feb 29 23:34 ibmca.sth
```

--------------------------------------------------------------------------------

**Step 2 : Updating Parameters**

--------------------------------------------------------------------------------

```
[testuser@server1 clientssl]$ db2 update dbm cfg using SSL_CLNT_KEYDB
/home/testuser/clientssl/ibmca.kdb
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[testuser@server1 clientssl]$ db2 update dbm cfg using SSL_CLNT_STASH
/home/testuser/clientssl/ibmca.sth
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
```

--------------------------------------------------------------------------------------

**Step 3: Importing root certificate**

--------------------------------------------------------------------------------------

Import the IBM CA root certificate (as downloaded in step 1) into the GSKit key database:

```
[testuser@server1 clientssl]$ gsk8capicmd_64 -cert -add -db "/home/testuser/clientssl/ibmca.kdb" -
pw "myServerPassw0rdpw0" -label "IBMRoot" -file "/home/testuser/clientssl/carootcert.der" -
format binary
[testuser@server1 clientssl]$
[testuser@server1 clientssl]$ ls -ltr
total 32
-rwxr-xr-x 1 testuser testuser 1001 Feb 29 23:33 carootcert.der
-rw------- 1 testuser testuser   88 Feb 29 23:34 ibmca.rdb
-rw------- 1 testuser testuser   88 Feb 29 23:34 ibmca.crl
-rw------- 1 testuser testuser  193 Feb 29 23:34 ibmca.sth
-rw------- 1 testuser testuser 5088 Feb 29 23:38 ibmca.kdb
```

----------------------------------------------------------------------------------

**Step 4 : Test Connection :**

----------------------------------------------------------------------------------

```
[testuser@server1 clientssl]$ db2 list db directory

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = TESTSSL
 Database name                        = TESTSSL
 Local database directory             = /home/testuser
 Database release level               = 15.00
 Comment                              =
 Directory entry type                 = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
 Alternate server port number         =

[testuser@server1 clientssl]$ db2 connect to testssl

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.8.0
 SQL authorization ID   = TESTUSER
 Local database alias   = TESTSSL


[testuser@server1 clientssl]$ netstat -an | grep -i 20033
tcp        0      0 0.0.0.0:20033           0.0.0.0:*               LISTEN


[testuser@server1 clientssl]$ db2 list node directory
SQL1027N  The node directory cannot be found.
```

```
[testuser@server1 clientssl]$ db2 catalog tcpip node sslca remote server1.fyre.ibm.com server
20033 security ssl
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[testuser@server1 clientssl]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.
[testuser@server1 clientssl]$
[testuser@server1 clientssl]$ db2 list node directory

 Node Directory

 Number of entries in the directory = 1

Node 1 entry:

 Node name                      = SSLCA
 Comment                        =
 Directory entry type           = LOCAL
 Protocol                       = TCPIP
 Hostname                       = server1.fyre.ibm.com
 Service name                   = 20033
 Security type                  = SSL
```

```
[testuser@server1 ~]$ db2 catalog db TESTSSL as NEWSSL at node SSLCA
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[testuser@server1 ~]$ db2 terminate
DB20000I  The TERMINATE command completed successfully.
[testuser@server1 ~]$
[testuser@server1 ~]$ db2 list db directory

 System Database Directory

 Number of entries in the directory = 2

Database 1 entry:

 Database alias                       = NEWSSL
 Database name                        = TESTSSL
 Node name                            = SSLCA
 Database release level               = 15.00
 Comment                              =
 Directory entry type                 = Remote
 Catalog database partition number    = -1
 Alternate server hostname            =
 Alternate server port number        =

Database 2 entry:

 Database alias                     = TESTSSL
 Database name                      = TESTSSL
 Local database directory           = /home/testuser
 Database release level             = 15.00
 Comment                            =
 Directory entry type               = Indirect
 Catalog database partition number    = 0
 Alternate server hostname          =
 Alternate server port number        =

[testuser@server1 ~]$ db2 connect to TESTSSL

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.8.0
 SQL authorization ID   = TESTUSER
 Local database alias   = TESTSSL

[testuser@server1 ~]$
[testuser@server1 ~]$ db2 list applications

Auth Id  Application    Appl.      Application Id                                          DB       # of
         Name           Handle                                                             Name     Agents
-------- -------------- ---------- -------------------------------------------------------------- -------- -----
TESTUSER db2bp          8          *LOCAL.testuser.240301080123                            TESTSSL  1



[testuser@server1 ~]$ db2 connect to NEWSSL user testuser
Enter current password for testuser:

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.8.0
 SQL authorization ID   = TESTUSER
 Local database alias   = NEWSSL

[testuser@server1 ~]$ db2 list applications

Auth Id  Application    Appl.      Application Id                                          DB       # of
         Name           Handle                                                             Name     Agents
-------- -------------- ---------- -------------------------------------------------------------- -------- -----
TESTUSER db2bp          25         10.11.88.103.60436.240301080248                          TESTSSL  1

[testuser@server1 ~]$ netstat -an | grep -i 20033
tcp        0      0 0.0.0.0:20033           0.0.0.0:*               LISTEN
tcp        0      0 10.11.88.103:60436      10.11.88.103:20033      ESTABLISHED
tcp        0      0 10.11.88.103:20033      10.11.88.103:60436      ESTABLISHED
```