Authenticate With JWT

1. Create connected app in Salesforce
   1. API (Enable OAuth Settings)
      1. Enable OAuth Settings
      2. Callback URL : <http://localhost:1717/OauthRedirect>
      3. Use digital signatures and upload digital certificate.
         1. For the digital certificate, you can create self-signed certificate.
      4. Exportable Private Key : True
   2. Add these OAuth scopes:
      1. Access and manage your data (API)
      2. Perform requests on your behalf at any time (refresh\_token, offline\_access)
      3. Provide access to your data via the Web (web)
   3. OAuth Policies : Admin approved users are pre-authorized
   4. Rest Keep Default.
2. Create a Permission set and deployment user.
   1. *Create a Permission set*
   2. *Add permission to the above connected app*
   3. *Assign this permission set to the deployment user (who wants JWT authentication)*
3. Change connected app OAuth policy
   1. *Go to manage connected app*
   2. *Edit policy and change OAuth policy to*
      1. Admin approved users are pre-authorized
   3. Verify in the Permission set section of the connected app is having the permission set added in step 2.
4. Create Self Signed Certificate and private key outside salesforce
   1. Use below command to create private key and certificate
      1. *openssl req -newkey rsa:2048 -nodes -keyout my\_store.key -x509 -days 1825 -out certificate.pem (here 1825 is the validity of the certificate in days)*
      2. *keytool -importcert -file certificate.pem -keystore keystore.jks -alias mycertificate -storetype jks*
5. Create Self Signed certificate in salesforce and generate private Key
   1. *Export private key in keystore from salesforce self-signed certificate. (Remember password) This will give you .jks file. (e.g. mycert.jks)*
   2. *After placing the .jks file in a directory, open the path with command prompt and run command to generate .p12 file (e.g. keystore.p12)*
      1. *keytool -importkeystore -srckeystore mycert.jks -destkeystore keystore.p12 -deststoretype PKCS12*
      2. *Provide destination keystore password and confirm.*
      3. *Provide destination keystore password (from step 4a)*
   3. *Now use below command in openssl to generate .key file (install openssl and set path in environment variable for bin library e.g. C:\Program Files\OpenSSL-Win64\bin)*
   4. *Now run command openssl and it should look like OpenSSL>*
   5. *Now run below command to generate private key file ‘.key’ from ‘.p12’ file*
   6. *This will ask password to import. Provide the one which you provided for 4.b.ii*
      1. *pkcs12 -in keystore.p12 -nocerts -nodes -out my\_store.key (from .p12 location)*

*or*

* + 1. *pkcs12 -in "C:\Users\devvratgoyal\Development\Self Signed Certificate and Private Key\keystore.p12" -nocerts -nodes -out "C:\Users\devvratgoyal\Development\Self Signed Certificate and Private Key\my\_store1.key" (from any location)*
    2. *This will produce the .key file which you can use for JWT authentication*

1. Authenticate with JWT
   1. Copy clientid from connected app step 1
   2. Copy directory path to the private key with file name generated at step 4.f.iii
   3. Copy user name of the deployment user (which has permission set assigned from step 2)
   4. Execute the below command for JWT Authentication where instance url is the target salesforce url.
   5. sfdx force:auth:jwt:grant --clientid <clientid> --jwtkeyfile <private key file including path> --username <deployment username> --instanceurl <salesforce instance>
      1. Example: sfdx force:auth:jwt:grant --clientid 3MVG9rKhT8ocoxGlOR.Bb0ZbUz1Noos4ZxnM2sWUCR.Xo3RjNtjlyNFmR3e109O1SOmf5x00Ps2KFbw30wOuz --jwtkeyfile C:\Users\devvratgoyal\Development\my\_store.key --username [devvrat.goyal@mizuhogroup.com.obdev2](mailto:devvrat.goyal@mizuhogroup.com.obdev2) --instanceurl [https://test.salesforce.com](https://test.salesforce.com/)
   6. This should return you following successfully message

Successfully authorized devvrat.goyal@mizuhogroup.com.obdev2 with org ID 00D250000009A0vEAE

1. Remove other certificate from keystore
   1. If you want to import the self-signed certificate from keystore (.jks) file to another environment, you may want to remove other self-signed certificate than the intended one to import using below command,
   2. keytool -delete -alias <certificate name> -keystore <jks file>
   3. keytool -delete -alias mobile\_ios\_app\_prod -keystore 00D250000009A0v.jks