windows10 python3.7 OpenSSL 1.1.1a 20 Nov 2018

1,make sure that you prepared self-signed CA and Signer CA using scripts

<https://github.com/kmwebnet/ECC608-Provision/tree/master/scripts>

make sure not to set same CN root-ca.crt and signer-ca.crt, or it causes unexpected error.

Results as follows:

2020/03/06 12:19 595 root-ca.crt

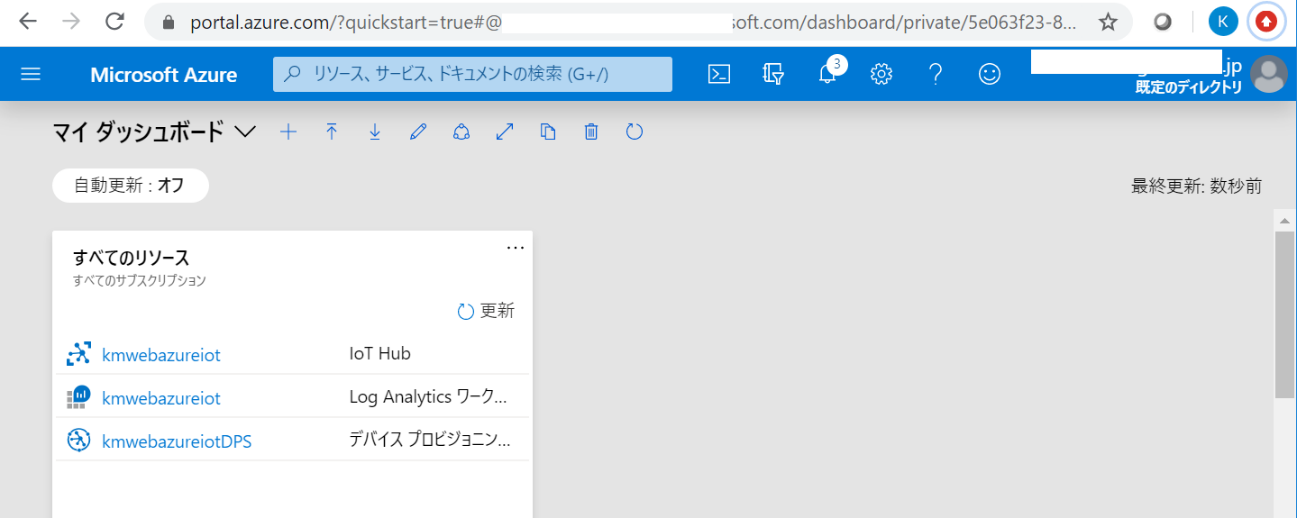
2020/03/06 12:19 241 root-ca.key

2020/03/06 12:20 672 signer-ca.crt

2020/03/06 12:19 241 signer-ca.key

you can get 4 ralated files.

2,create Azure IoT Hub and Azure IoT DPS Service in your Azure account.

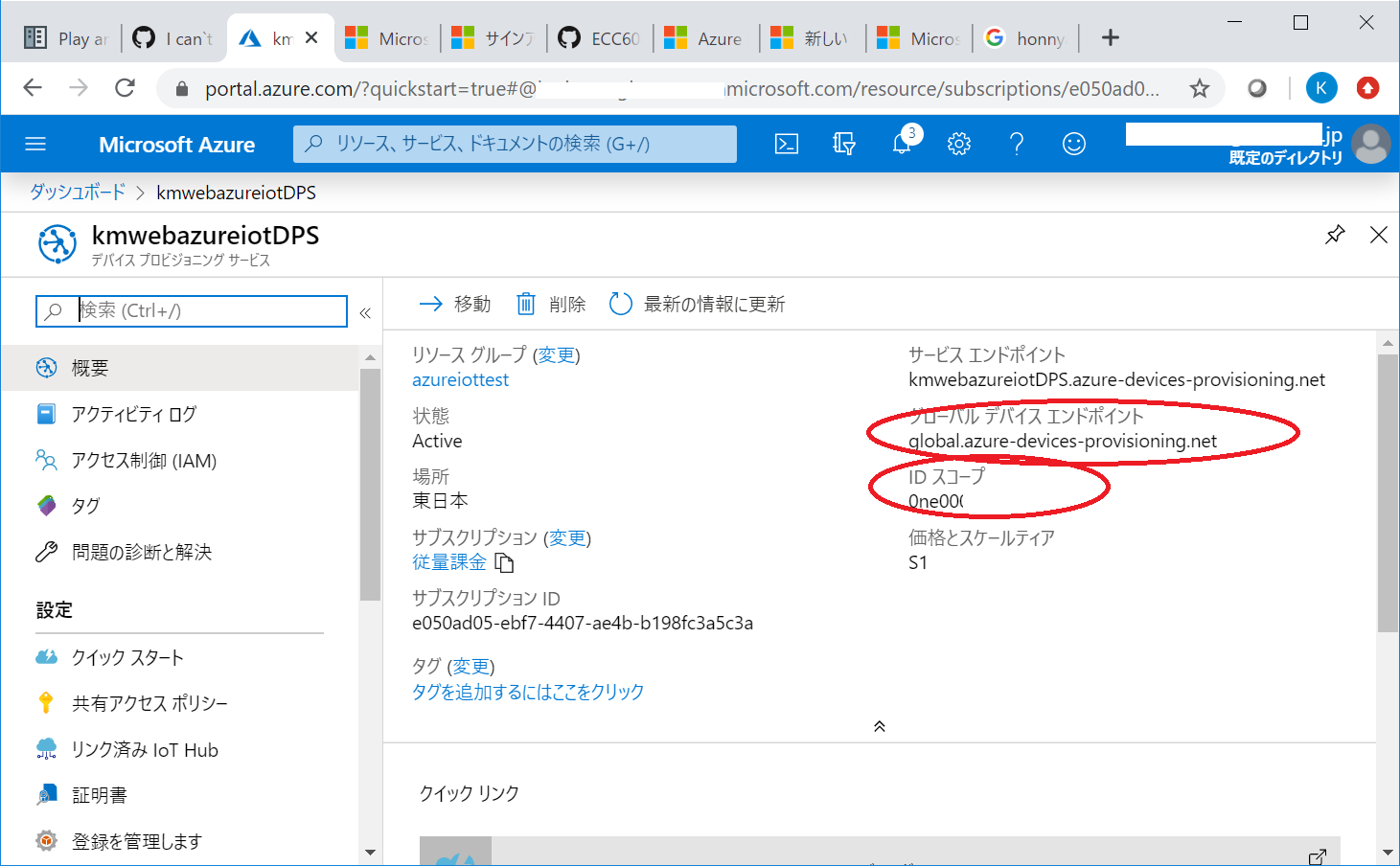
you can make sure them as follows if you pinned to your dashboard.

3,setup your Azure IoT DPS service

you need to record some items as highlighted below to execute latter steps.

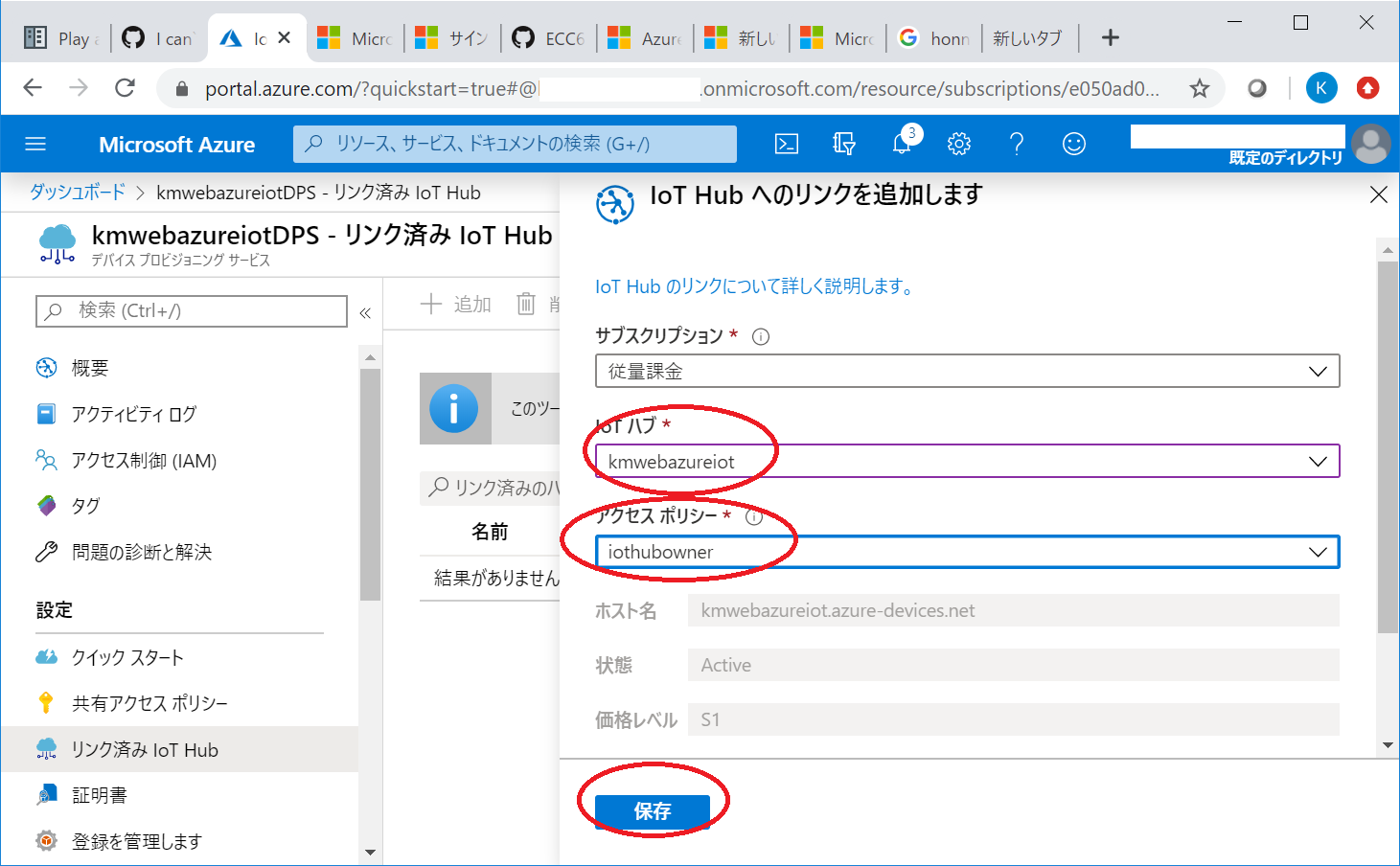
a, global device endpoint URL(This seems to be common to all users)

b, ID Scope



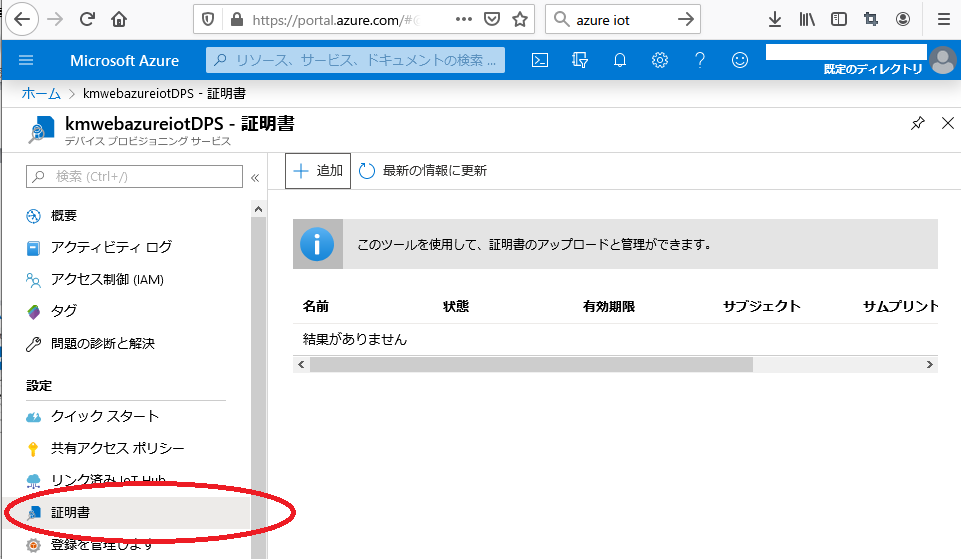
make sure your Azure DPS is linked to your Azure IoT by checking “Linked Hub” menu.

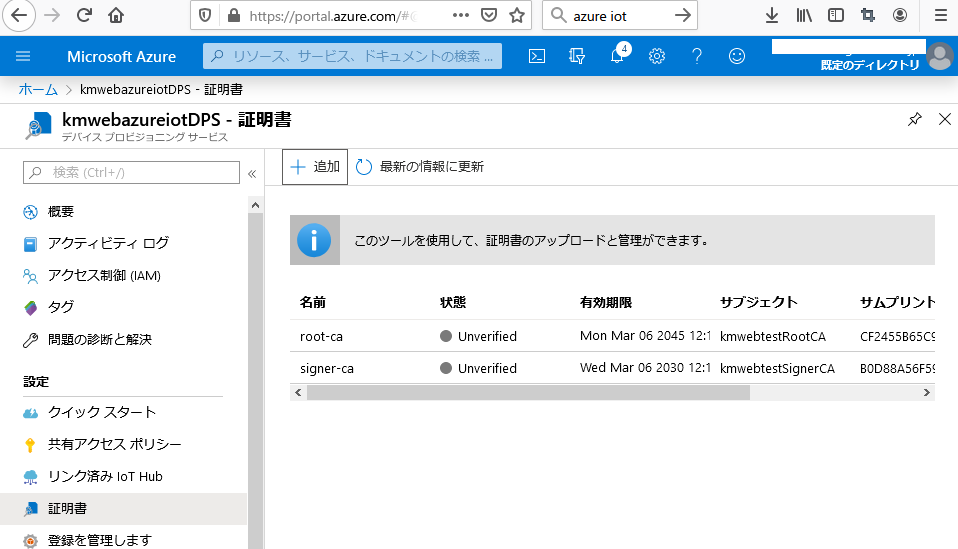
if it's not, you need to set as follows:



4,upload root-ca.crt and signer-ca.crt created above.

at first, rename their “.crt” to “.pem” and then and then Azure will accept them.

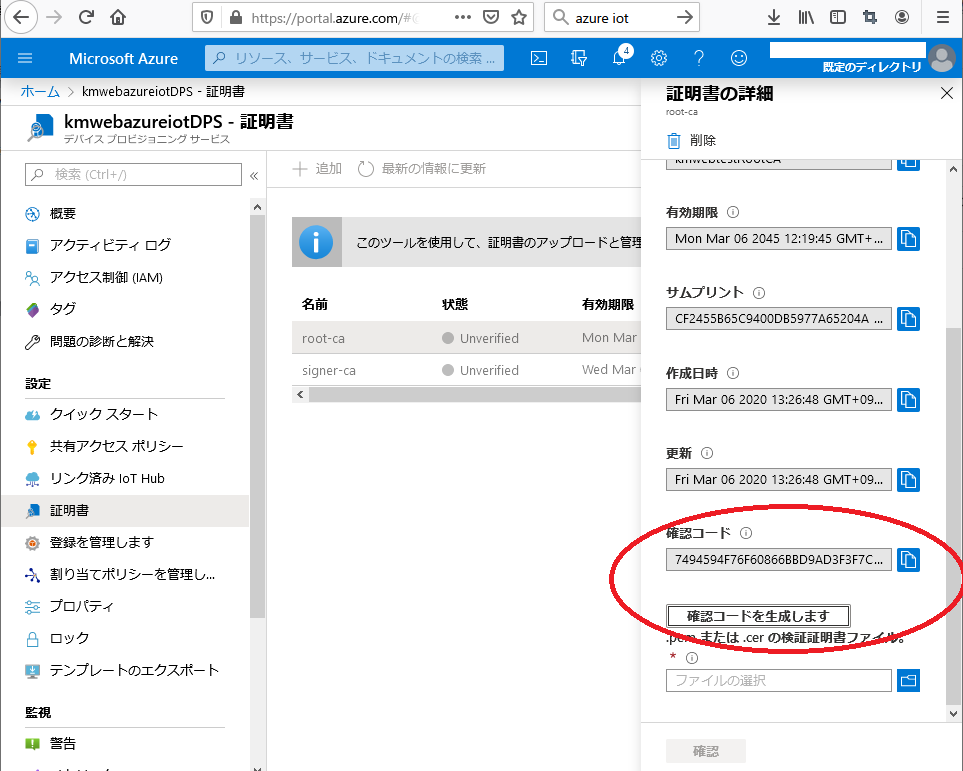




at this point, you can see their status are “Unverified”.

so, you need to turn them “verified” status.

at first, create verify certificate for root-ca.pem.



you can generate verify code in every each of certificate row. you can see the example for root-ca.pem as above.

copy verify code and in command prompt, do some commands and make verify certificate.

openssl ecparam -name prime256v1 -genkey > azuredpsroot.key

openssl req -new -sha256 -key azuredpsroot.key -subj "/C=JP/ST=Tokyo/L=Tokyo/O=kmweb/OU=kmweb/CN=<verify code>" -out azuredpsroot.csr

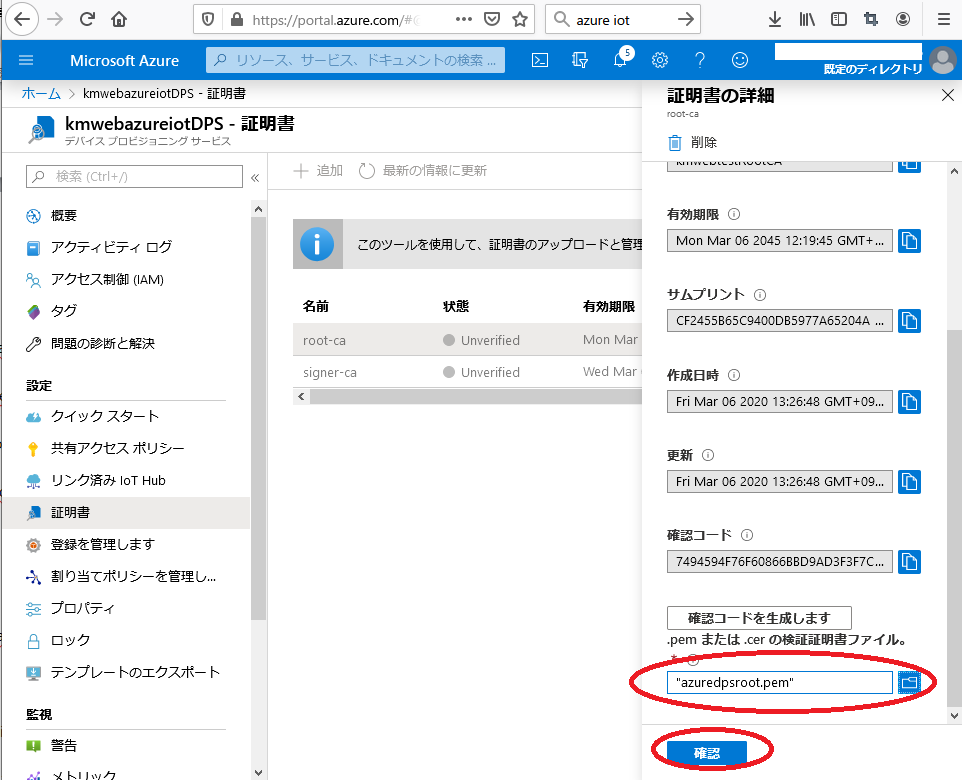
openssl x509 -req -in azuredpsroot.csr -CA root-ca.pem -CAkey root-ca.key -CAcreateserial -out azuredpsroot.pem -days 3650 -sha256

you get 3 additional files as below.

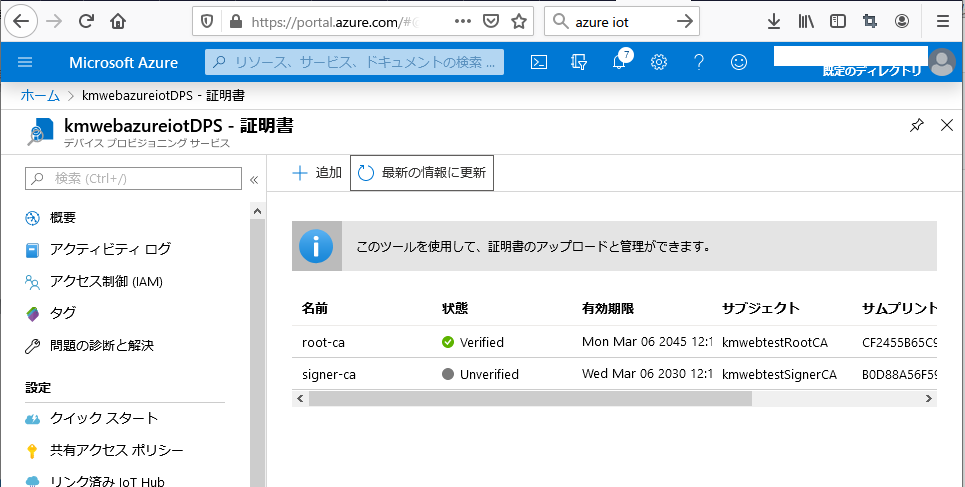
2020/03/06 13:40 526 azuredpsroot.csr

2020/03/06 13:40 310 azuredpsroot.key

2020/03/06 13:41 660 azuredpsroot.pem



upload azuredpsroot.pem as verify certificate.



verify root-ca.pem succeded as below.

you also need to veryfy signer-ca.pem with same steps.

openssl ecparam -name prime256v1 -genkey > azuredpssigner.key

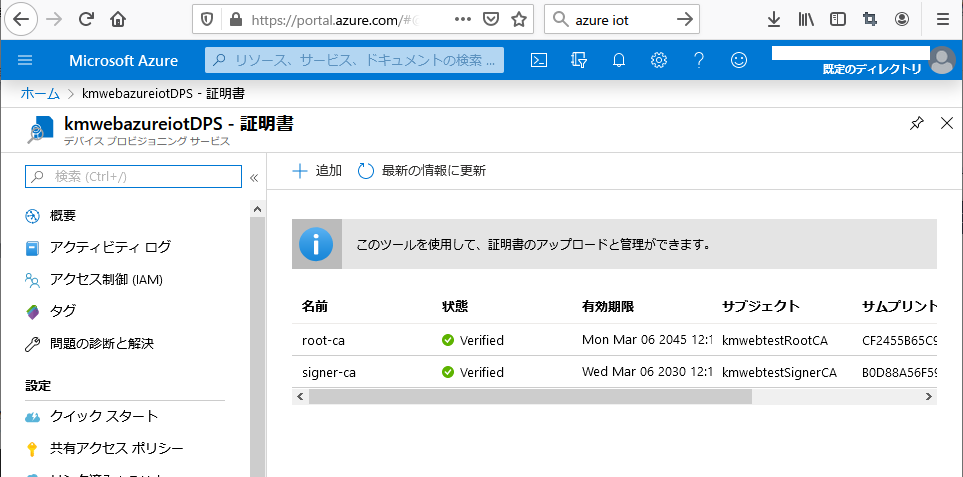
openssl req -new -sha256 -key azuredpssigner.key -subj "/C=JP/ST=Tokyo/L=Tokyo/O=kmweb/OU=kmweb/CN=<verify code>" -out azuredpssigner.csr

openssl x509 -req -in azuredpssigner.csr -CA signer-ca.pem -CAkey signer-ca.key -CAcreateserial -out azuredpssigner.pem -days 3650 -sha256

Signature ok

subject=C = JP, ST = Tokyo, L = Tokyo, O = kmweb, OU = kmweb, CN = <verify code>

Getting CA Private Key



finished preparation for certificates on Azure DPS.

5,upload root-ca.crt and signer-ca.crt created above on Azure IoT Hub.

you also need two of certificates used above to be set on Azure IoT Hub as well.

commands as follows:

openssl ecparam -name prime256v1 -genkey > azureiotroot.key

openssl req -new -sha256 -key azureiotroot.key -subj "/C=JP/ST=Tokyo/L=Tokyo/O=kmweb/OU=kmweb/CN=<verify code>" -out azureiotroot.csr

openssl x509 -req -in azureiotroot.csr -CA root-ca.pem -CAkey root-ca.key -CAcreateserial -out azureiotroot.pem -days 3650 -sha256

Signature ok

subject=C = JP, ST = Tokyo, L = Tokyo, O = kmweb, OU = kmweb, CN = <verify code>

Getting CA Private Key

openssl ecparam -name prime256v1 -genkey > azureiotsigner.key

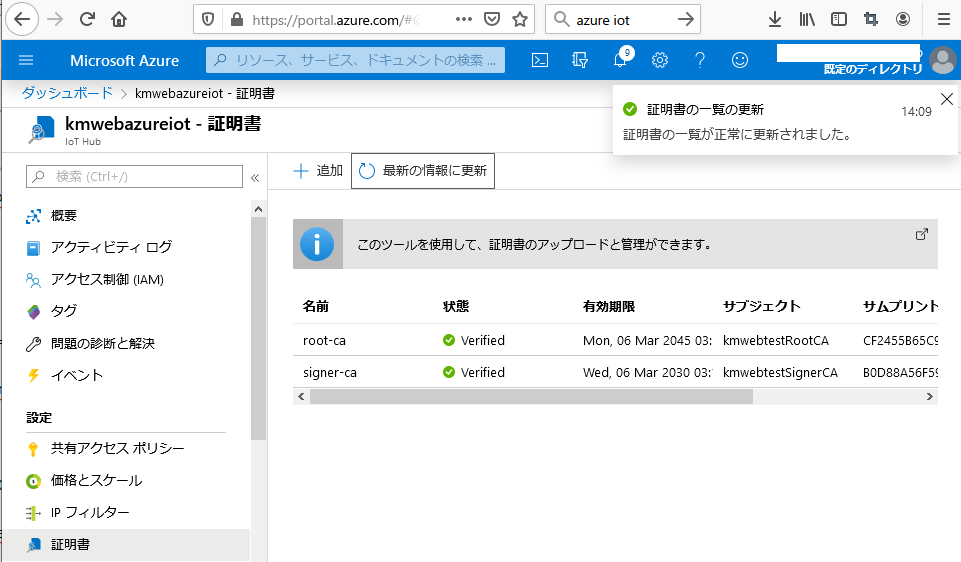
openssl req -new -sha256 -key azureiotsigner.key -subj "/C=JP/ST=Tokyo/L=Tokyo/O=kmweb/OU=kmweb/CN=<verify code>" -out azureiotsigner.csr

openssl x509 -req -in azureiotsigner.csr -CA signer-ca.pem -CAkey signer-ca.key -CAcreateserial -out azureiotsigner.pem -days 3650 -sha256

Signature ok

subject=C = JP, ST = Tokyo, L = Tokyo, O = kmweb, OU = kmweb, CN = <verify code>

Getting CA Private Key



finished preparation for certificates on Azure IoT.