

# AN12403

EdgeLock™ SE05x for secure connection to IBM Watson IoT

Rev. 1.4 — 7 December 2020

Application note

535113

## Document information

Information	Content
Keywords	EdgeLock SE05x, Watson IoT Core, Secure cloud onboarding
Abstract	This application note describes how to leverage the EdgeLock SE05x ease of use configuration for secure cloud onboarding to the Watson IoT cloud platform. It provides detailed instructions to run the software example provided as part of the support package using an OM-SE050ARD and an MCU board.



## Revision history

---

### Revision history

Revision number	Date	Description
1.0	2019-06-24	First document release
1.1	2019-10-15	Added EdgeLock SE05x ease of use configuration
1.2	2019-11-22	Updated project import from SDK instead of CMake.
1.3	2020-01-20	Fixed broken links.
1.4	2020-12-07	Updated to latest template and fixed broken URLs

## Abbreviations

Table 1. Abbreviations

Acronym	Description
OEM	Original Equipment Manufacturer

## 1 EdgeLock SE05x ease of use configuration

The IoT device identity should be unique, verifiable and trustworthy so that device registration attempts and any data uploaded to Watson IoT can be trusted by the OEM. Watson IoT verifies the device identity using PKI cryptography. This authentication scheme requires that the associated private key remains secret and hidden from users, software or malicious attackers during the product's lifecycle

The EdgeLock SE05x security IC is designed to provide a tamper-resistant platform to safely store keys and credentials needed for device authentication and device onboarding to cloud service platforms such as Watson IoT. Using the EdgeLock SE05x security IC, OEMs can safely connect their devices to Watson IoT without writing security code or exposing credentials or keys.

However, key generation and injection into security ICs can introduce vulnerabilities if not done properly. Manual provisioning can lead to errors and is difficult to scale when more devices are needed. Also, to ensure keys are kept safe, injection should take place in a trusted environment, in a facility with security features like tightly controlled access, careful personnel screening, and secure IT systems that protect against cyberattacks and theft of credentials among others.

In order to allow OEMs to get rid of the complexity of key management and to offload the cost of ownership of a PKI infrastructure, the EdgeLock SE05x is offered pre-provisioned for ease of use. This means that OEMs are not required to program additional credentials and can leverage the EdgeLock SE05x ease of use configuration for most of the use cases, including for secure cloud onboarding of their devices to Watson IoT.

**Note:** For more information about the EdgeLock SE05x ease of use configuration, please refer to [AN12436 - SE050 configurations](#).

## 2 Leveraging EdgeLock SE05x ease of use configuration for Watson IoT

Watson IoT uses X.509 certificate-based attestation mechanisms for verifying the device authenticity during a registration attempt. This authentication scheme requires a certificate chain of trust, from the OEM's CA certificate to the OEM's device certificate as well as their associated private key. In addition, Watson IoT uses unique device identifiers to whitelist connection attempts. For this reason, OEMs are required to add the device ID during its registration to the platform.

The EdgeLock SE05x is offered off-the-shelf pre-provisioned so that OEMs are not required to program any additional credentials to onboard their devices to Watson IoT. On the one hand, EdgeLock SE05x provides a tamper-resistant platform to safely store the private key of the device digital certificate used for authentication and registration to Watson IoT platform. On the other hand, the device UID can be read out from the EdgeLock SE05x (e.g. at manufacturing time) and white-listed on the Watson IoT platform.

Figure 1 illustrates the device registration flow to Watson IoT leveraging the EdgeLock SE05x ease of use configuration:

1. NXP delivers to the OEM's device manufacturer a quantity of EdgeLock SE05x security ICs based on a purchase order. The EdgeLock SE05x samples come pre-provisioned with die-individual credentials.
2. The OEM's device manufacturer assembles the EdgeLock SE05x security ICs and deploys the software into the final IoT devices. It also needs to take care to read out the device ID key from the EdgeLock SE05x samples.
3. The OEM, as the system operator of the Watson IoT account, needs to upload the NXP certificate chain of trust and register every device by uploading its device ID. The certificate chain of trust can be obtained from NXP and the device ID can be obtained from EdgeLock SE05x security ICs.
4. The OEM ships IoT devices to end customers.
5. IoT devices boot up and automatically connect to Watson IoT service using the private key of the device digital certificate inside EdgeLock SE05x ease of use configuration.

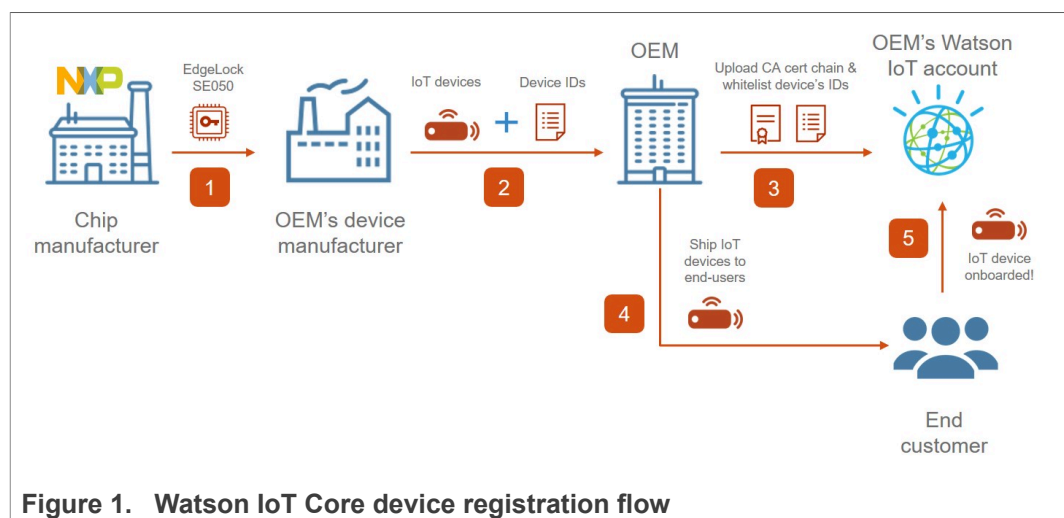


Figure 1. Watson IoT Core device registration flow

**Disclaimer:** The described device registration flow spans across multiple roles given the various entities involved. How each role is mapped in the registration flow might be scenario-dependent for each OEM.

### 3 Running the Watson IoT device onboarding demo example

The Watson IoT demo example included in the EdgeLock SE05x Plug & Trust Middleware is an illustrative software example that showcases how to leverage EdgeLock SE05x security IC to set up trusted connections to Watson IoT cloud.

This section explains the steps required to run the Watson IoT demo leveraging the **EdgeLock SE05x ease of use configuration**. We also use the FRDM-K64F board as an example, but the same steps can be replicated using any the MCU/ MPUs supported by the EdgeLock SE05x Plug & Trust Middleware.

On the other hand, if you prefer to generate and inject your own credentials in EdgeLock SE05x for the Watson IoT demo, please refer to [Section 4](#). It explains how to use the provisioning scripts included as part of EdgeLock SE05x Plug & Trust Middleware for that purpose.


**Note:** *The Watson IoT device onboarding procedure described in this section and the Watson IoT demo example are provided only for evaluation purposes. Therefore, the subsequent procedure must be adapted and adjusted accordingly for a commercial deployment.*

#### 3.1 Hardware required

This guide provides detailed instructions to the Watson IoT project example using the hardware described below. However, you could use other MCU / MPU boards supported by EdgeLock SE05x Plug & Trust Middleware for this purpose as well.


1. OM-SE050ARD development kit:

Table 2. OM-SE050ARD development kit details

Part number	12NC	Content	Picture
<a href="#">OM-SE050ARD</a>	935383282598	EdgeLock SE050 development board	

2. FRDM-K64F board:

Table 3. FRDM-K64F details

Part number	12NC	Content	Picture
<a href="#">FRDM-64F</a>	935326293598	Freedom development platform for Kinetis K64, K63 and K24 MCUs	

### 3.2 Cloud connectivity certificated used for Watson IoT device onboarding

As part of the EdgeLock SE05x ease of use configuration, a set of die-individual credentials are injected in each EdgeLock SE05x security IC. OEMs can directly use these pre-injected credentials to securely onboard their devices to Watson IoT, without the need to program any additional keys.

[Table 4](#) shows the ECC256 key pair and certificate we will use to run the Watson IoT device onboarding demo example. This ECC256 key pair and certificate has been selected as an example, for a complete detail of the EdgeLock SE05x ease of use configuration, refer to [AN12436 - SE050 configurations](#).

**Table 4. Cloud connectivity certificate used for Watson IoT device onboarding**

Key name and type	Certificate	Usage policy	Erasable by customer	Identifier
Cloud connection key 0, ECC256, Die Individual	Cloud Connectivity Certificate 0, ECC signed	Default	Yes	Key: 0xF0000100 Cert: 0xF0000101

### 3.3 Read EdgeLock SE05x device ID

Watson IoT platform only accepts connection attempts from white-listed devices. For this reason, OEMs are required to add device ID during its registration process into the platform. This section explains how to read the device ID from the EdgeLock SE05x using the tools included in EdgeLock SE05x Plug & Trust Middleware.

This section explains how to read out the device ID from the EdgeLock SE05x using a FRDM-K64F board as a host platform.

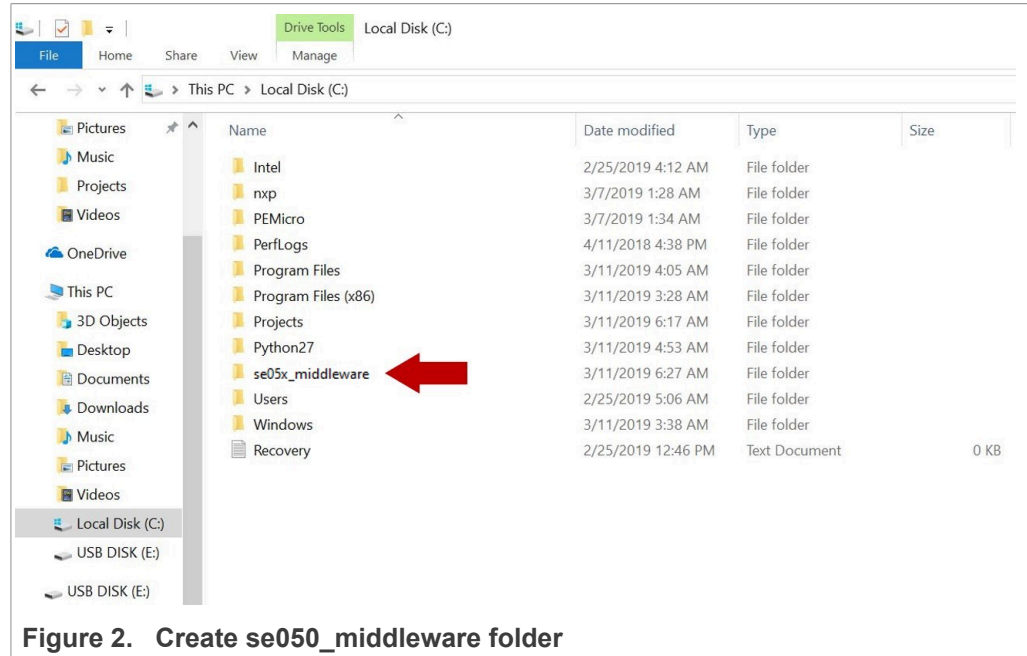
**Note:** Check [AN12396- Quick start guide to Kinetis K64](#) for detailed instructions on how to bring up the FRDM-K64F board.

#### 3.3.1 Download EdgeLock SE05x Plug & Trust Middleware

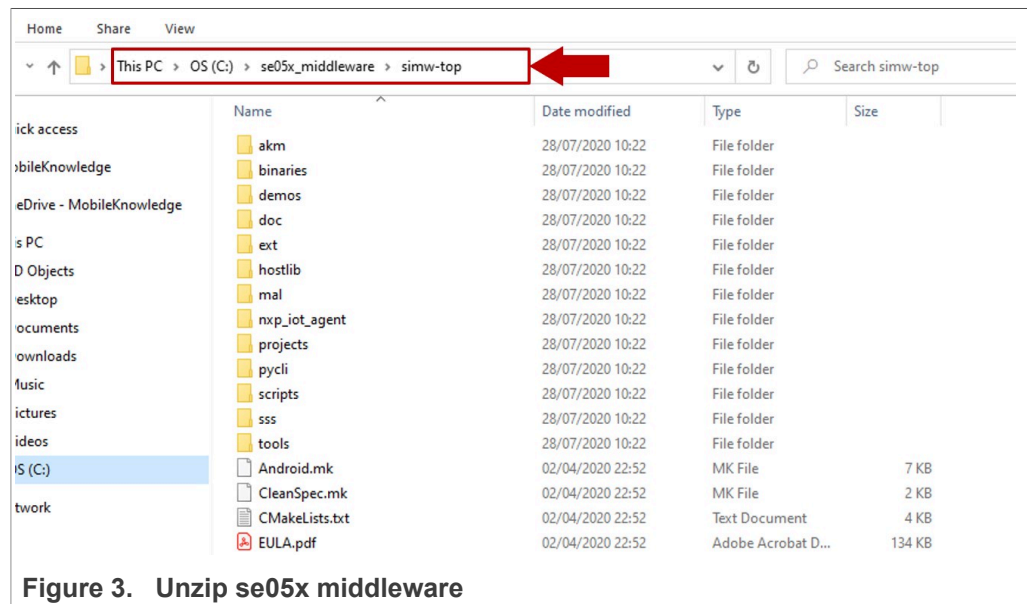
Follow these steps to download the EdgeLock SE05x Plug & Trust Middleware in your local machine:

1. Download EdgeLock SE05x Plug & Trust Middleware from the [NXP website](#)

2. Create a folder called **se05x\_middleware** in C: directory as shown in [Figure 2](#):



3. Unzip the EdgeLock SE05x Plug & Trust Middleware inside the **se05x\_middleware** folder. After unzipping, you will see a folder called **simw-top** created. The contents of the **simw-top** directory should look as they appear in [Figure 3](#):



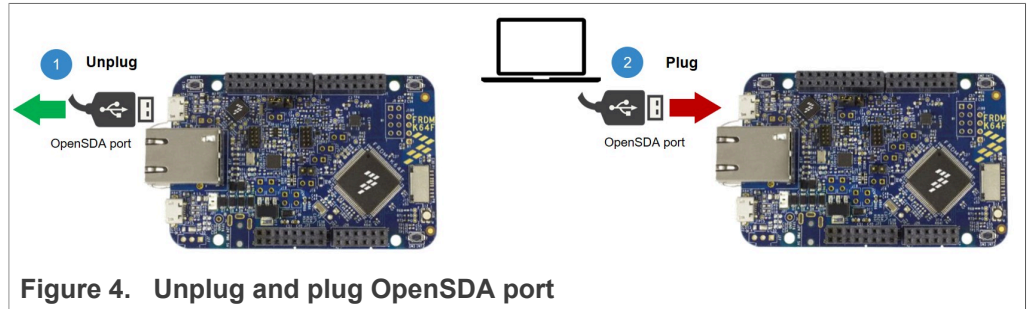
**Note:** It is recommended to keep *se05x\_middleware* with the **shortest** path possible and **without spaces** in it. This avoids some issues that could appear when building the middleware if the path contains spaces.



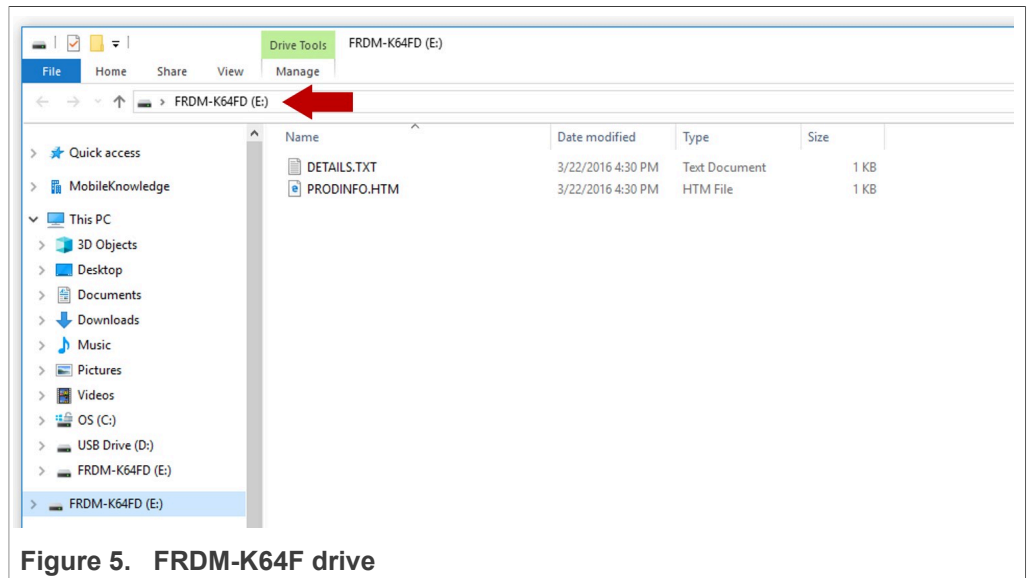
### 3.3.2 Flash FRDM-K64F with VCOM software

The VCOM software allows the FRDM-K64F board to be used as a bridge between the Windows machine and the EdgeLock SE05x and enables the execution of the EdgeLock SE05x `sscli` tool and other utilities from the laptop. To flash the VCOM software into the FRDM-K64F, follow these steps:

1. Unplug and plug again the USB cable to the openSDA USB port as shown in [Figure 4](#):



2. When you plug the board, your laptop should recognize the board as an external drive as shown in [Figure 5](#):



3. Flash the VCOM software to FRDM-K64F. The VCOM software binary can be found in the EdgeLock SE05x Plug & Trust Middleware package, inside the `simw-top\binaries` folder as shown in [Figure 6](#):

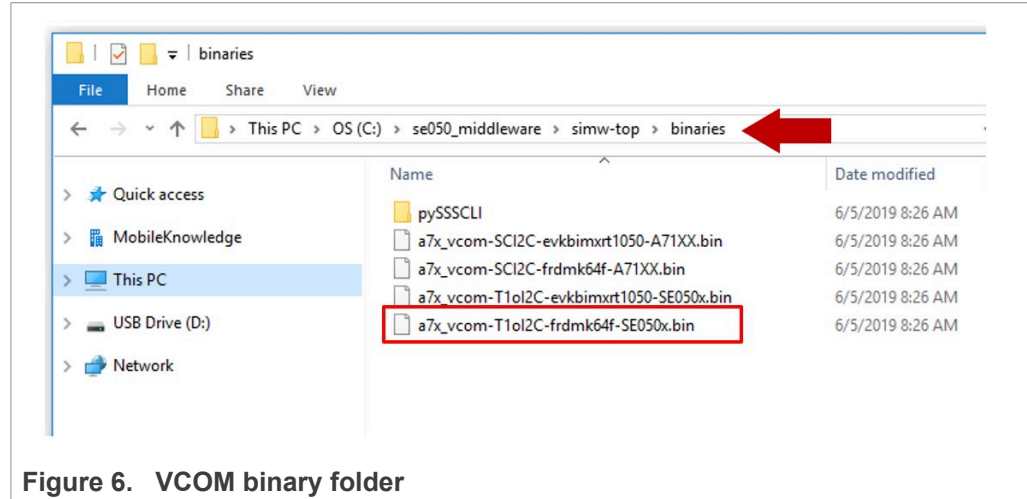


Figure 6. VCOM binary folder

4. Drag and drop or copy and paste the `a7x_vcom-T1oI2C-frdmk64f-SE050x.bin` file into the FRDM-K64F drive from your computer file explorer as shown in [Figure 7](#):

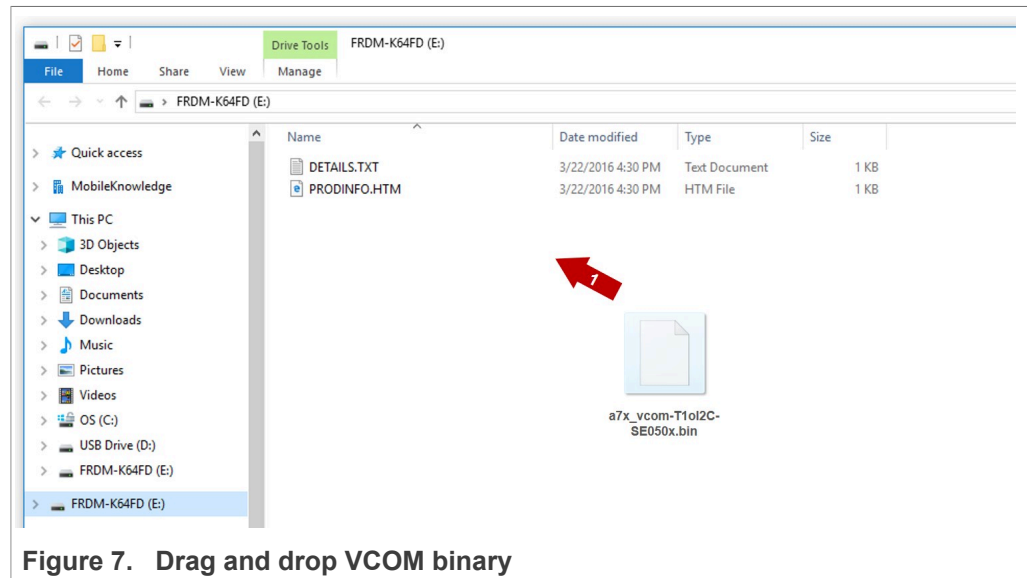


Figure 7. Drag and drop VCOM binary

5. The serial and VCOM ports should be recognized by your Device Manager. To check that the ports are recognized, follow the steps indicated in [Figure 8](#):
  - a. Unplug the USB cable from the OpenSDA USB port.
  - b. Plug the USB cable to the OpenSDA USB port.
  - c. Check that the serial port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as *USB Serial Device (COM7)* but this naming might change depending on your computer. Therefore, it is important that you

identify which device is recognized at the moment you plug the SDA USB port to the computer.

- d. Plug the USB cable to the K64F USB port.
- e. Check that the VCOM port is recognized in the category **Ports (COM & LPT)**. In this document, it is recognized as *Virtual Com Port (COM8)* but this naming might change depending on your computer (e.g. It could also appear named as *USB Serial Device*). Therefore, it is important that you identify which device is recognized at the moment you plug the K64F USB port to the computer.

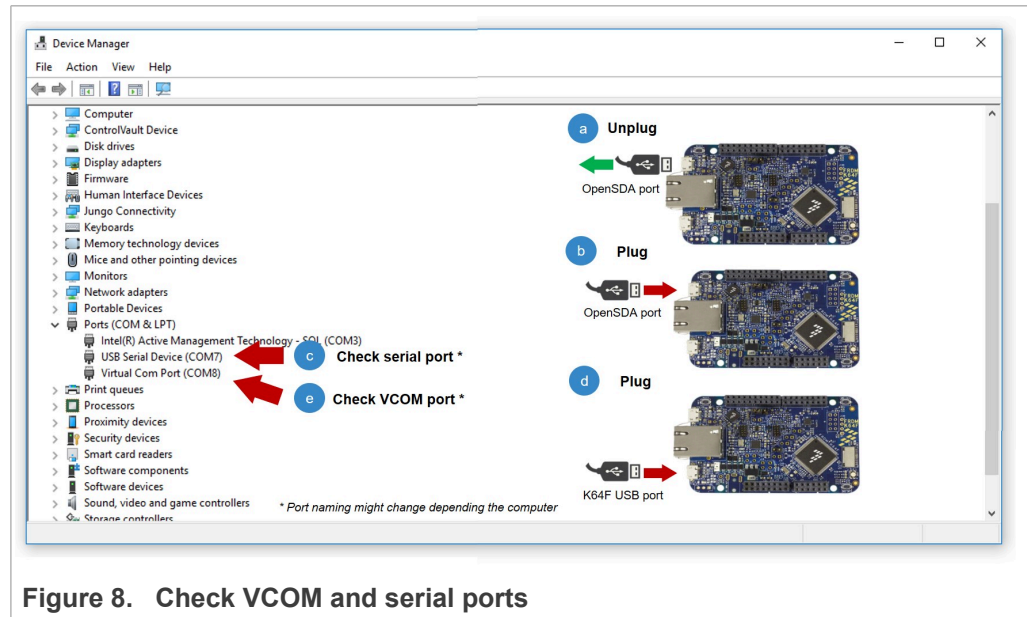


Figure 8. Check VCOM and serial ports

**Note:** Please note that it is possible that either of the two COM ports is not detected when using low-quality or charge-only USB cables.

### 3.3.3 Read EdgeLock SE05x device ID

To read EdgeLock SE05x device ID:

1. Mount OM-SE050ARD on top of the FRDM-K64F. Then, connect FRDM-K64F OpenSDA port and K64F port to your laptop as shown in [Figure 9](#)

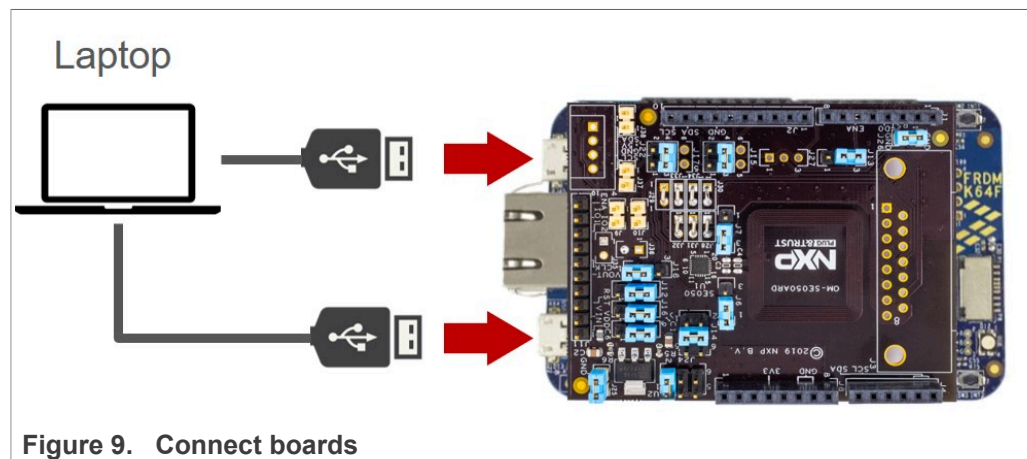


Figure 9. Connect boards

2. Start the `ssscli` tool by sending the commands shown in [Figure 10](#):
  - a. Go to `simw-top\binaries\pySSCLI` folder:  
Send: `>cd se050_middleware\simw-top\binaries\pySSCLI`
  - b. Check your VCOM port number in your Device Manager. Open the connection using the `ssscli`:  
Send: `>ssscli connect se050 vcom <COM_NUMBER>`
  - c. Send the reset command:  
Send: `>ssscli se05x reset`

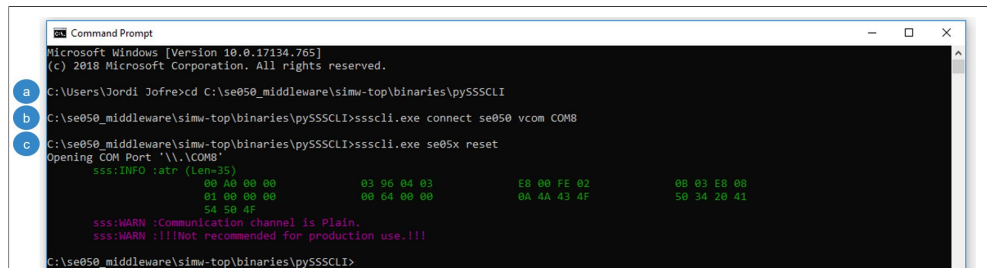


Figure 10. Start ssscli tool

**Note:** If you see the following message: `WARNING:sss.connect:Session already open, close current session first` as shown in [Figure 11](#), it means that you have a session open. To close it, send: (1) `> ssscli disconnect` and then send once again (2) `>ssscli connect se050 vcom <COM_NUMBER>` and later (3) `>ssscli se05x reset`.

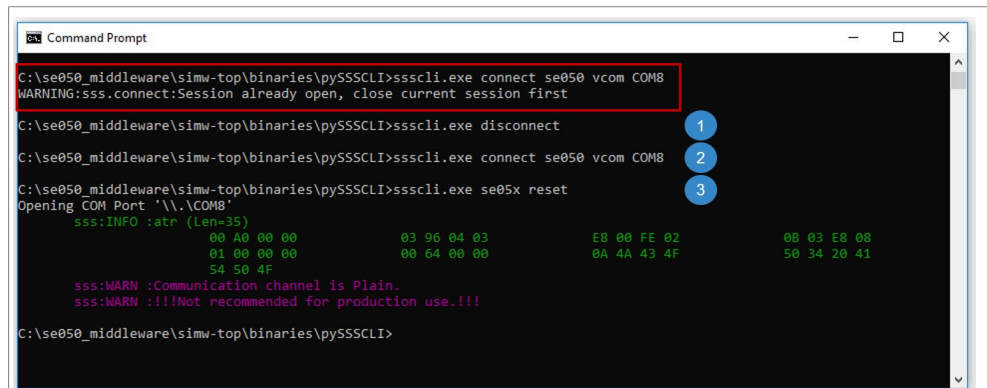


Figure 11. Close an already opened session

3. Read the EdgeLock SE05x device ID as shown in [Figure 12](#):
  - (1) Send: >ssscli se05x uid
  - (2) Copy the EdgeLock SE05x device ID in a text file for later use.

```

C:\se050_middleware\simw-top\binaries\pySSCLI>ssscli se05x uid
Opening COM Port '\\.\COM69'
sss:INFO :atr (Len=35)
          00 A0 00 00          03 96 04 03          E8 00 FE 02          08 03 E8 08
          01 00 00 00          00 64 00 00          0A 4A 43 4F          50 34 28 41
          54 50 4F
sss:WARN :Communication channel is Plain.
sss:WARN :!!!Not recommended for production use!!!
INFO:sss.se05x:04005001c02edce5c8c39e04250559550000
INFO:sss.se05x:Unique ID: 04005001c02edce5c8c39e04250559550000
    
```

Figure 12. Retrieve SE050 device UID

4. Close the connection.
  - ([Figure 13](#)) Send: >ssscli disconnect

```

C:\se050_middleware\simw-top\binaries\pySSCLI>ssscli.exe disconnect
C:\se050_middleware\simw-top\binaries\pySSCLI>
    
```

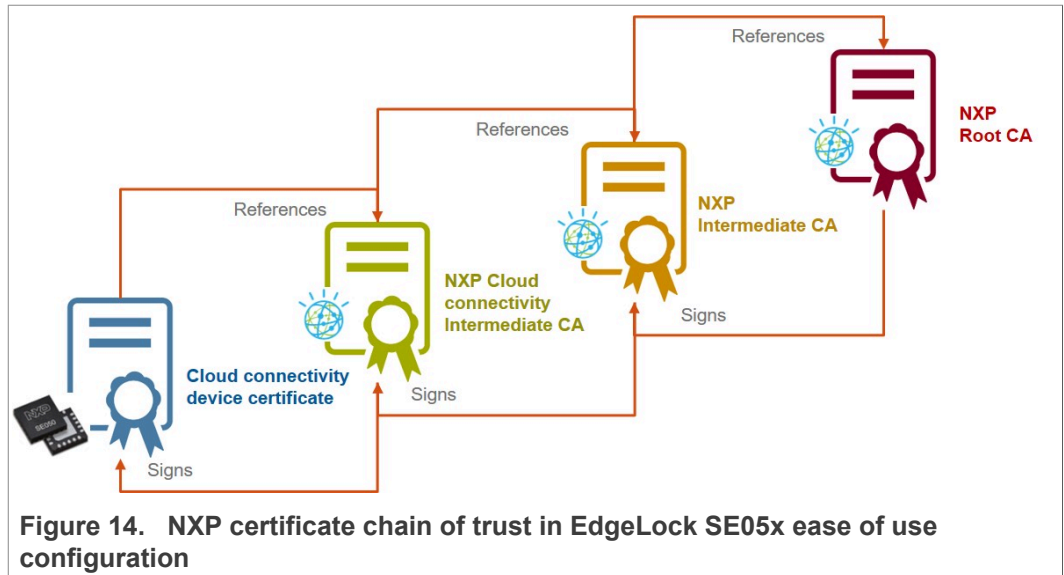
Figure 13. Disconnect ssscli

If you have completed this section, go to [Section 3.4](#).

### 3.4 Obtain NXP certificate chain of trust

During the Watson IoT onboarding process, device are required to present its client certificate to authenticate to the Watson IoT backend. However, to validate the client certificate, it is first necessary to upload to Watson the whole certificate chain of trust, from the root CA certificate to the device certificate.

In our case, the NXP certificate chain of trust in EdgeLock SE05x ease of use configuration consists of one root CA certificate and two intermediate CA certificates as shown in [Figure 14](#)

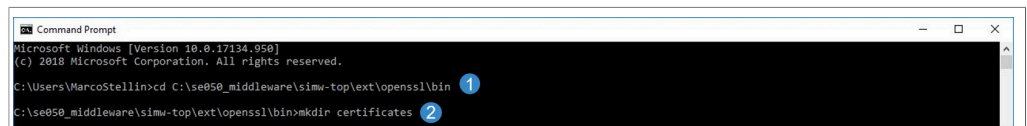


This section explains how to obtain the NXP certificate chain of trust and how to convert them to a format accepted by Watson using the OpenSSL toolset.

### 3.4.1 Download and convert the CA certificates

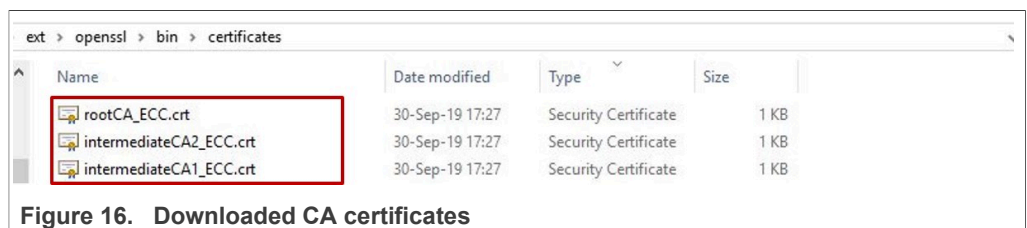
To obtain the NXP certificate chain of trust:

1. Open a command prompt
2. Navigate to the location of the OpenSSL binary in the middleware folder and create a new folder to store the NXP CA certificates, as shown in [Figure 15](#):
  - (1) Send > cd C:\se050\_middleware\simw-top\ext\openssl\bin
  - (2) Send > mkdir certificates



**Figure 15. Create a folder to store the CA certificates**

3. Open the following links with a browser to download the required NXP CA certificates and save them in the previously created `certificates` folder, as shown in [Figure 16](#)
  - (1) [Root ECC CA certificate](#). Save it as `rootCA_ECC.crt`.
  - (2) [Intermediate ECC CA certificate 1](#). Save it as `intermediateCA1_ECC.crt`.
  - (3) [Intermediate ECC CA certificate 2](#). Save it as `intermediateCA2_ECC.crt`.



**Figure 16. Downloaded CA certificates**



4. The downloaded certificates are in DER format, which is currently not supported by Watson. To convert them to PEM format using OpenSSL, send the commands shown in [Figure 17](#):

```
(1) Send > openssl x509 -in certificates\rootCA_ECC.crt -inform der -outform pem -out certificates\rootCA_ECC.pem
(2) Send > openssl x509 -in certificates\intermediateCA1_ECC.crt -inform der -outform pem -out certificates\intermediateCA1_ECC.pem
(3) Send > openssl x509 -in certificates\intermediateCA2_ECC.crt -inform der -outform pem -out certificates\intermediateCA2_ECC.pem
```

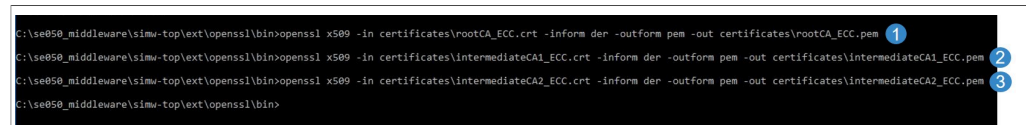


Figure 17. Convert ECC CA certificates

5. The certificates folder should now contain the required certificates in PEM format as shown in [Figure 18](#):

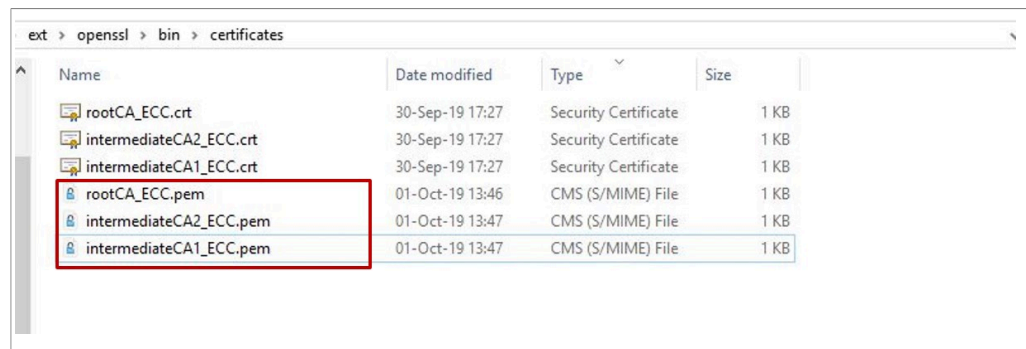


Figure 18. Folder containing the CA certificates in PEM format

If you have completed this section, go to [Section 3.5](#).

### 3.5 Prepare Watson IoT platform

This section describes how to get started with Watson IoT using the built-in web console dashboard. The Watson IoT dashboard is a web service offering a friendly user interface. This chapter describes how to::

- Create a free Watson IoT account.
- Register a CA certificate.
- Configure the connection security policy.
- Register your device or gateway.

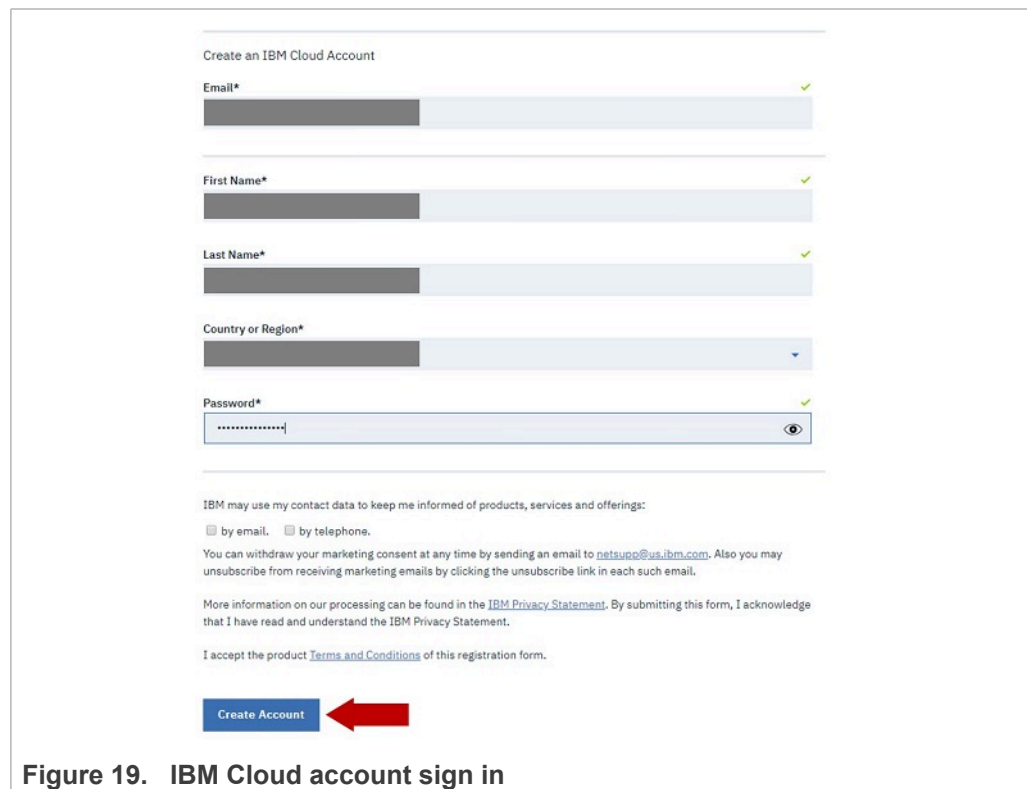
Watson IoT offers APIs, sample apps and client libraries for developing and managing Watson IoT applications. For the sake of simplicity, this application note only uses the Watson IoT using the built-in web console. For details on other Watson IoT developmen tools, refer to the [Watson documentation](#).

### 3.5.1 Create a Watson IoT instance

This section describes how to create an instance of the Watson IoT Platform service in your IBM account. IBM offers a free plan called *Lite account* that allows OEMs to start building apps and explore services tagged with the *Lite* tag. This account does not require a credit card and never expires. To create an Watson IoT account:

**Note:** *If you already have a Watson IoT account, you can sign in and jump to [Section 3.5.2](#)*

1. Go to <https://cloud.ibm.com/registration> , fill in the form with your account details and click **Create Account** button ([Figure 19](#))



Create an IBM Cloud Account

Email\*  ✓

First Name\*  ✓

Last Name\*  ✓

Country or Region\*  ▼

Password\*  ✓

IBM may use my contact data to keep me informed of products, services and offerings:  
 by email.  by telephone.

You can withdraw your marketing consent at any time by sending an email to [netssupp@us.ibm.com](mailto:netssupp@us.ibm.com). Also you may unsubscribe from receiving marketing emails by clicking the unsubscribe link in each such email.

More information on our processing can be found in the [IBM Privacy Statement](#). By submitting this form, I acknowledge that I have read and understand the IBM Privacy Statement.

I accept the product [Terms and Conditions](#) of this registration form.

←

Figure 19. IBM Cloud account sign in



2. Confirm your account using a link sent to your email. Then, go to <https://cloud.ibm.com/login> and sign in your account as shown in [Figure 20](#):

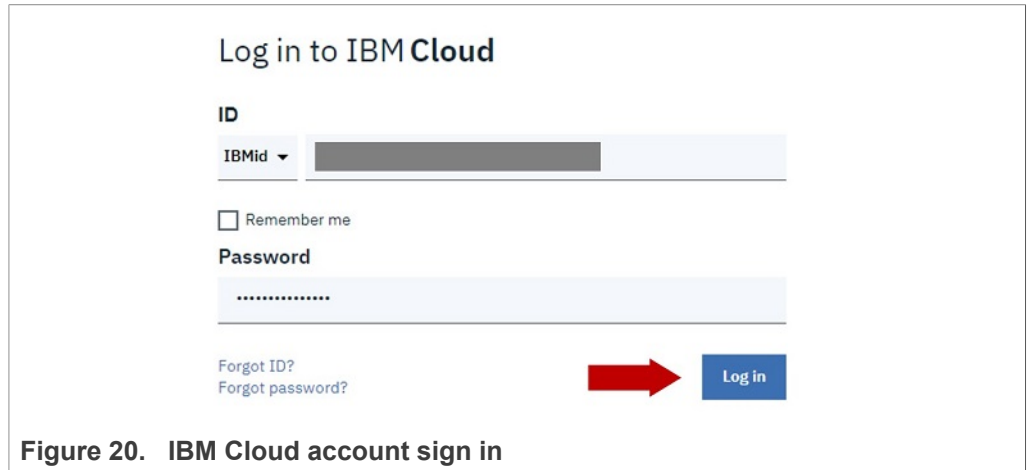


Figure 20. IBM Cloud account sign in

3. After signing in, you will land in IBM cloud dashboard. To access Watson IoT platform, select *Catalog* , then select *Internet of Things* and click *Internet of Things Platform* as shown in [Figure 21](#):

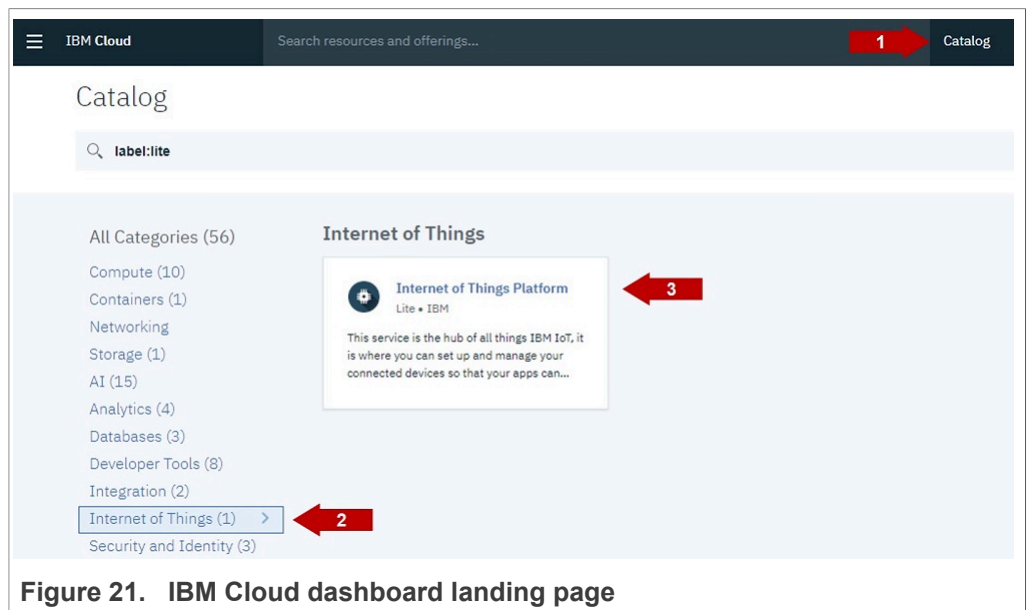


Figure 21. IBM Cloud dashboard landing page

- 4. To create a Watson IoT instance, write a *Service name*, *Choose a region* and click **Create** as shown in [Figure 22](#):

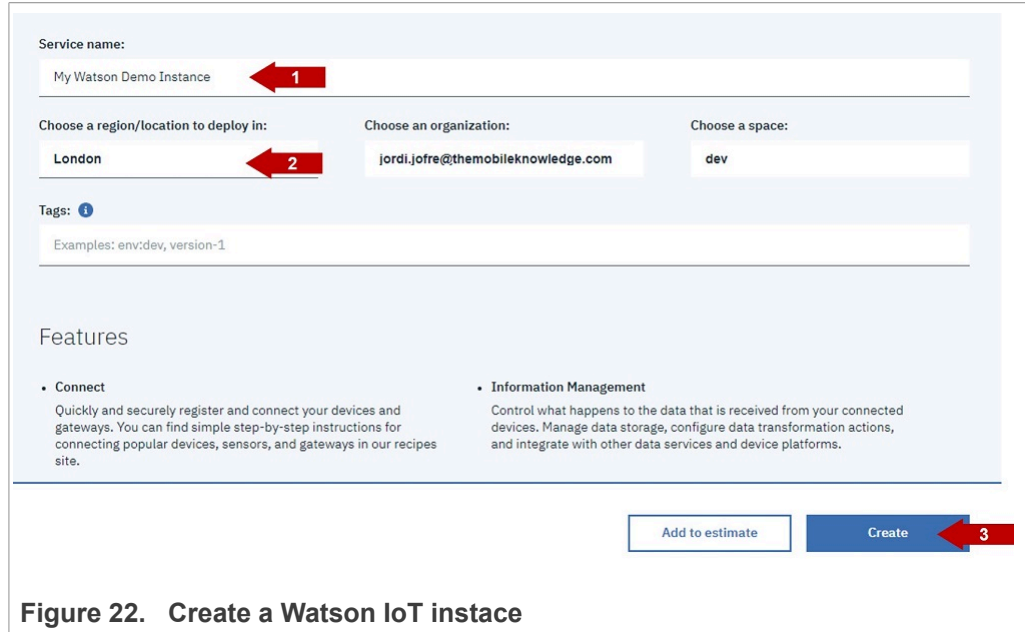


Figure 22. Create a Watson IoT instance

### 3.5.2 Register certificate chain of trust for device authentication

Watson IoT uses certificates for device authentication. Any devices that do not have valid signed certificates are denied access and cannot communicate with Watson IoT servers. CA certificates enable the OEM to recognize the client certificates on trusted devices so that they can connect to Watson IoT infrastructure. To get started with Watson IoT:

- 1. Click the **Launch** button as shown in [Figure 23](#):

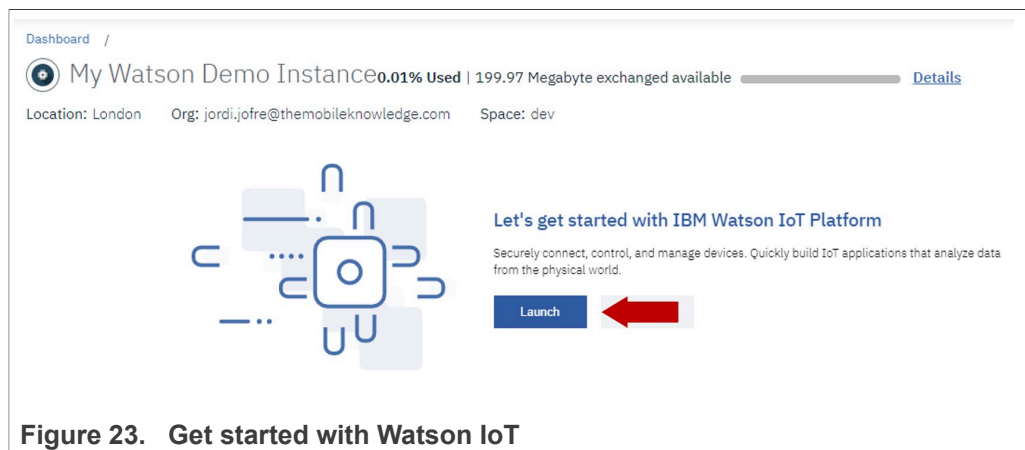
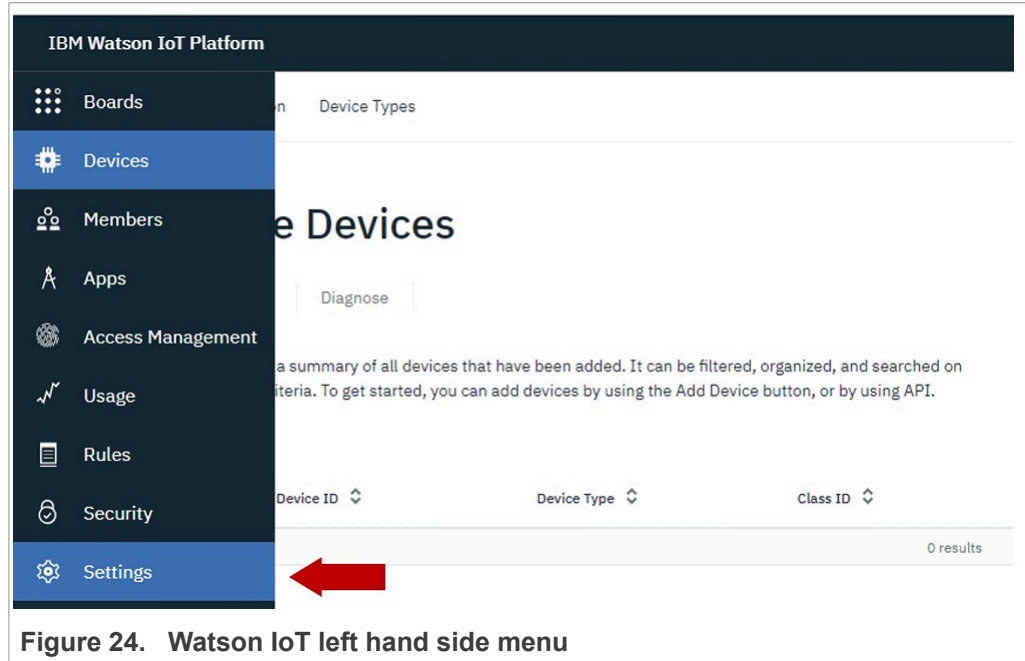
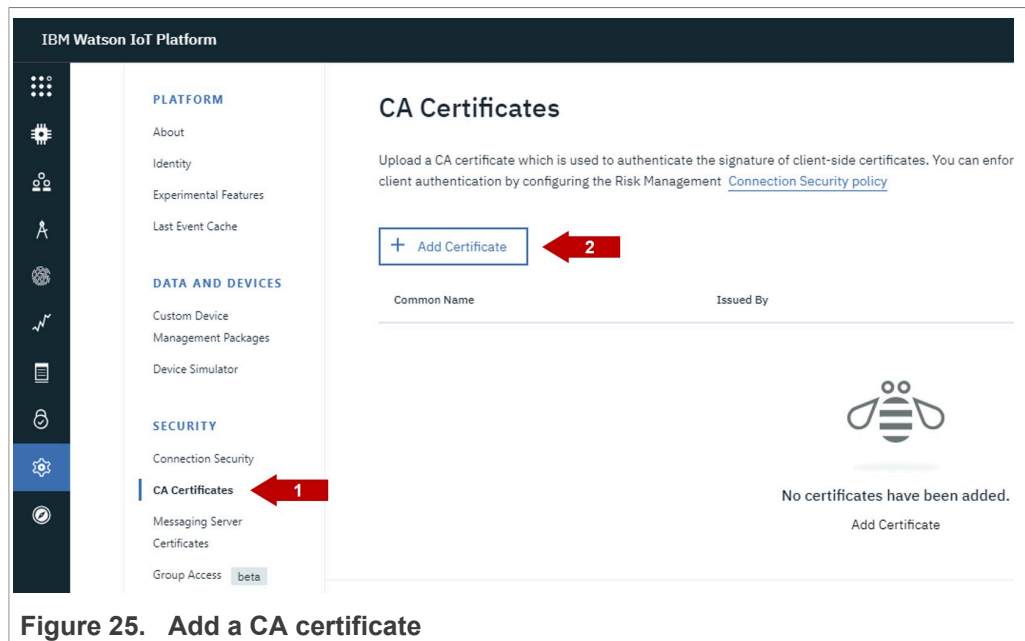


Figure 23. Get started with Watson IoT

- In the Watson IoT dashboard, use the following steps to register your CA certificates. First, click **Settings** from the menu on your left hand side as shown in [Figure 24](#):



- In the **Security** section, select **CA Certificates**, and click **Add Certificate** as shown in [Figure 25](#):



- Browse your file system and upload one by one all the `rootCA_ECC.pem` certificates generated in [Section 3.4](#) in the **Add Certificate** window and click **Save**

as shown in [Figure 26](#). This certificate should have been generated in the C : \se050\_middleware\simw-top\ext\openssl\bin\certificates folder:

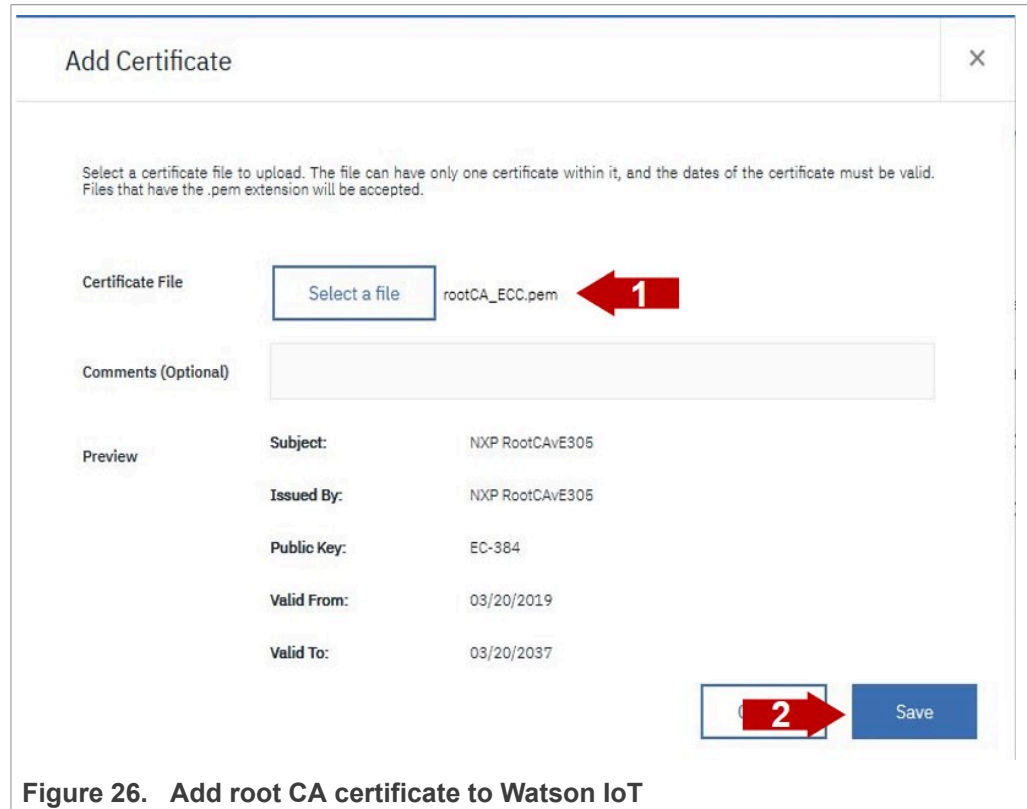
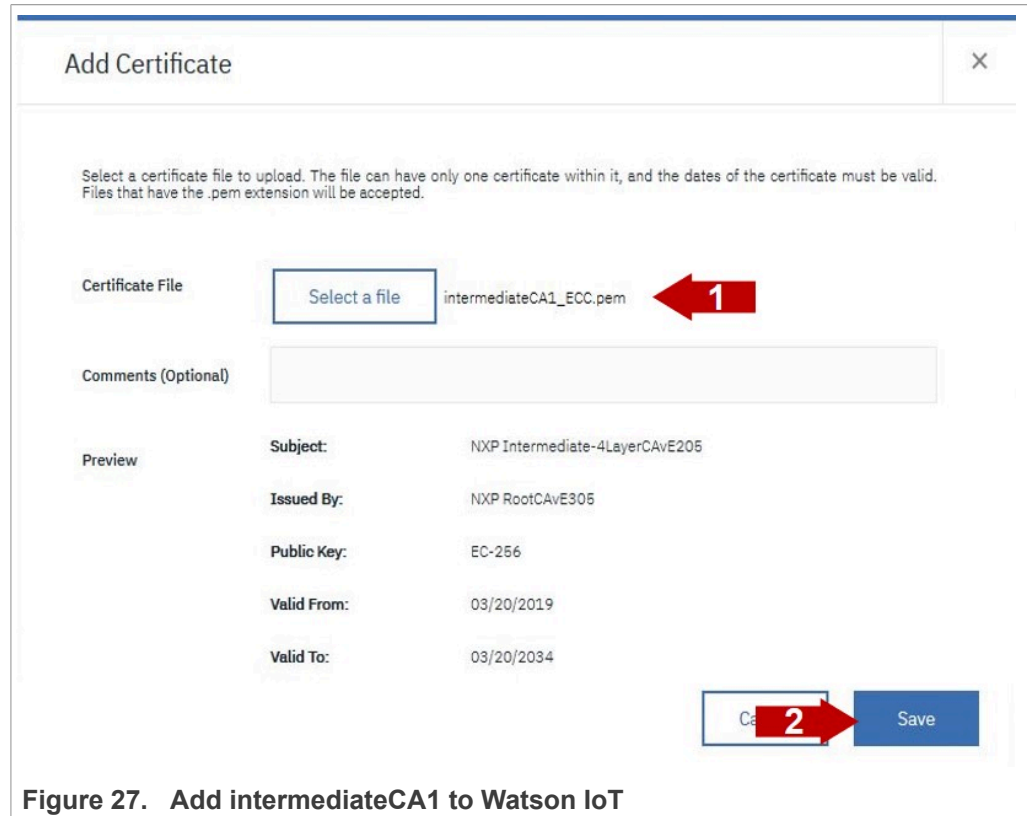


Figure 26. Add root CA certificate to Watson IoT

5. Browse your file system and upload one by one all the intermediateCA1\_ECC.pem certificates generated in [Section 3.4](#) in the **Add Certificate** window and click **Save**

as shown in [Figure 27](#). This certificate should have been generated in the C : \se050\_middleware\simw-top\ext\openssl\bin\certificates folder:



6. Browse your file system and upload one by one all the `intermediateCA2_ECC.pem` certificates generated in [Section 3.4](#) in the **Add Certificate** window and click **Save**

as shown in [Figure 28](#). This certificate should have been generated in the C : \se050\_middleware\simw-top\ext\openssl\bin\certificates folder:

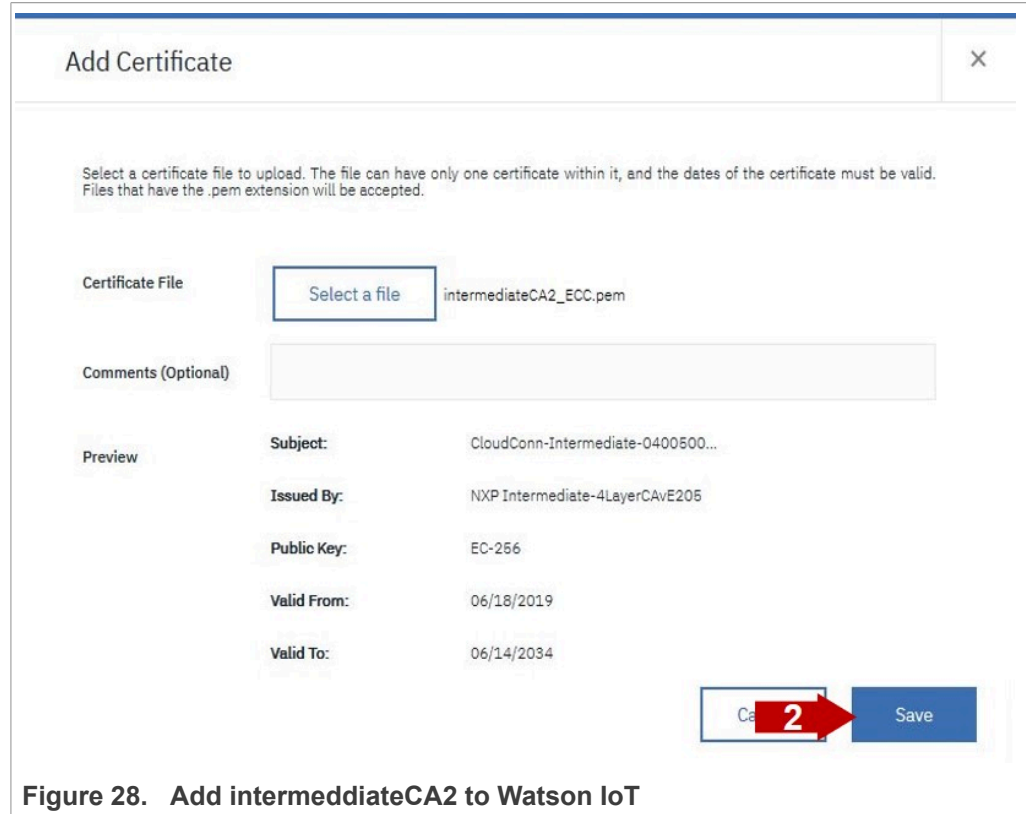


Figure 28. Add intermeddiateCA2 to Watson IoT

7. You should now have three CA certificates as shown in [Figure 29](#):

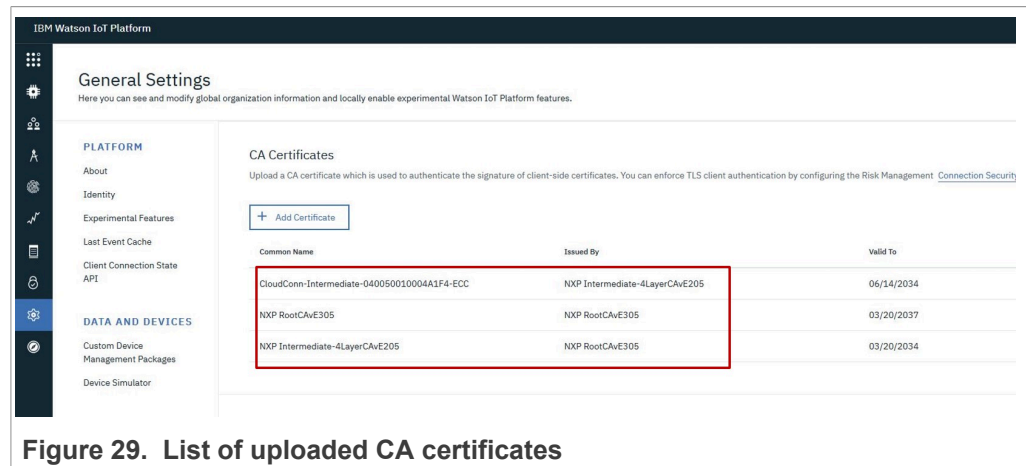


Figure 29. List of uploaded CA certificates

### 3.5.3 Configure connection security policy for advanced security

This section describes how to set *Default Connection Security to TLS with Client Certificate Authentication*. This setting defines the default security level that is applied to all devices. From the menu on your left hand side:

1. Click on *Settings*, navigate to *Connection Security* and click **Open Connection Security Policy** as shown in [Figure 30](#):

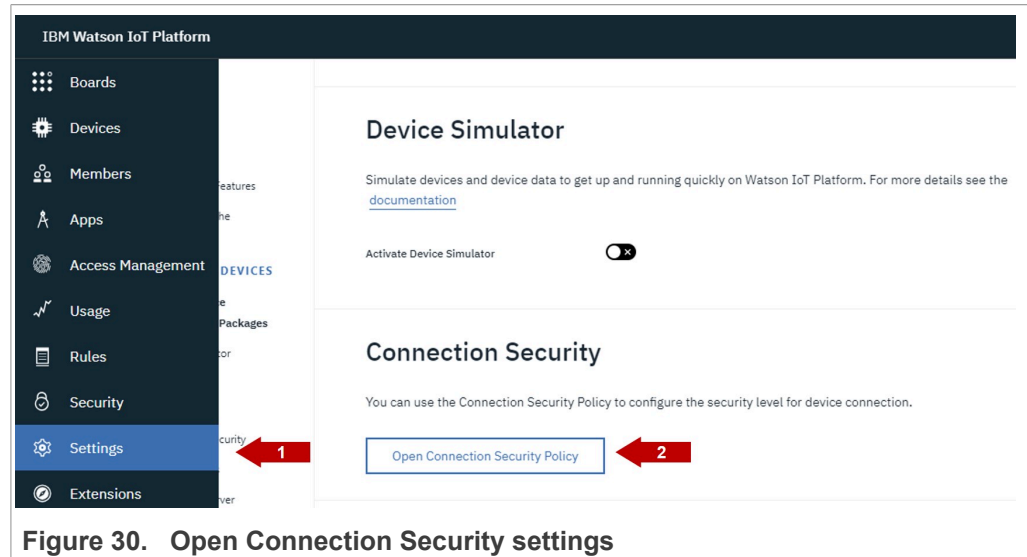


Figure 30. Open Connection Security settings

2. Click on the pencil of *Connection Security* to edit the preferences ([Figure 31](#)):

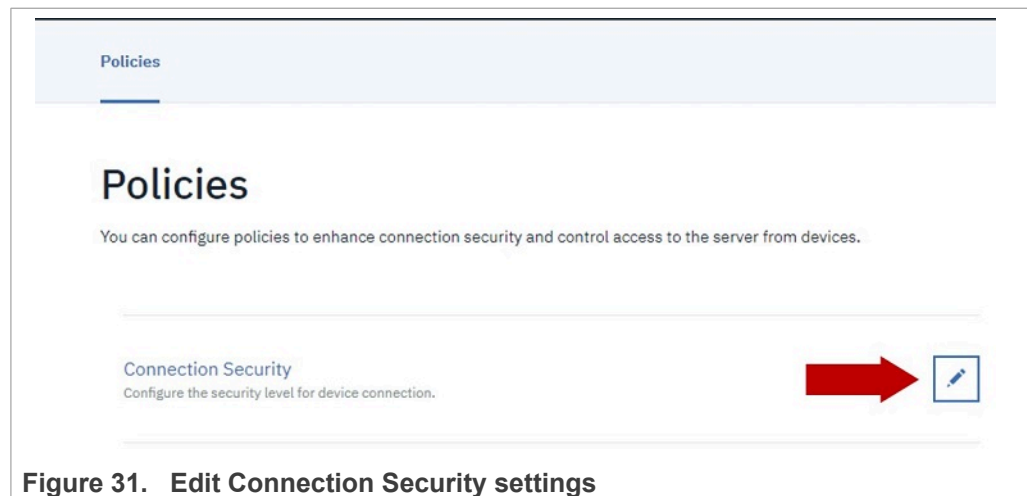


Figure 31. Edit Connection Security settings

3. Under *Security Level*, select from the drop-down list **TLS with Client Certificate Authentication** as a default connection security level. This is the security setting that

will apply to all devices, except for devices having custom connection settings. Click **Refresh Compliance**. to update the rule and finally click **Save** as shown in [Figure 32](#):

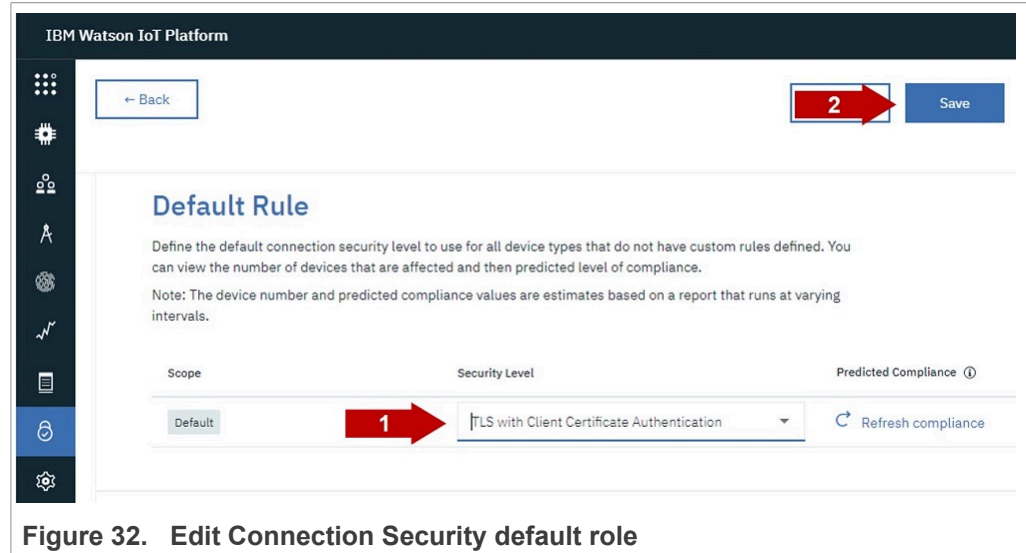


Figure 32. Edit Connection Security default role

### 3.5.4 Register your device or gateway to Watson IoT

Watson IoT allow OEMs to connect what IBM defines as *devices* and *gateways*. On the one hand, a *device* is anything that has a connection to the Internet and has data to send to or receive from the cloud. On the other hand, you can deploy a *gateway* to retrieve and send data to Watson IoT if your devices cannot directly connect to the Internet. *Gateways* have all the functions of a device, but can also publish and subscribe on behalf of the devices connected to them.

Watson IoT requires you to register devices before they can connect to Watson servers. First, we need to define the device type, whether we want to register a *device* or a *gateway*. To add a device type:



1. From the menu pane, select **Devices**, then select **Device Types** and click **Add Device Type** as shown in [Figure 33](#):

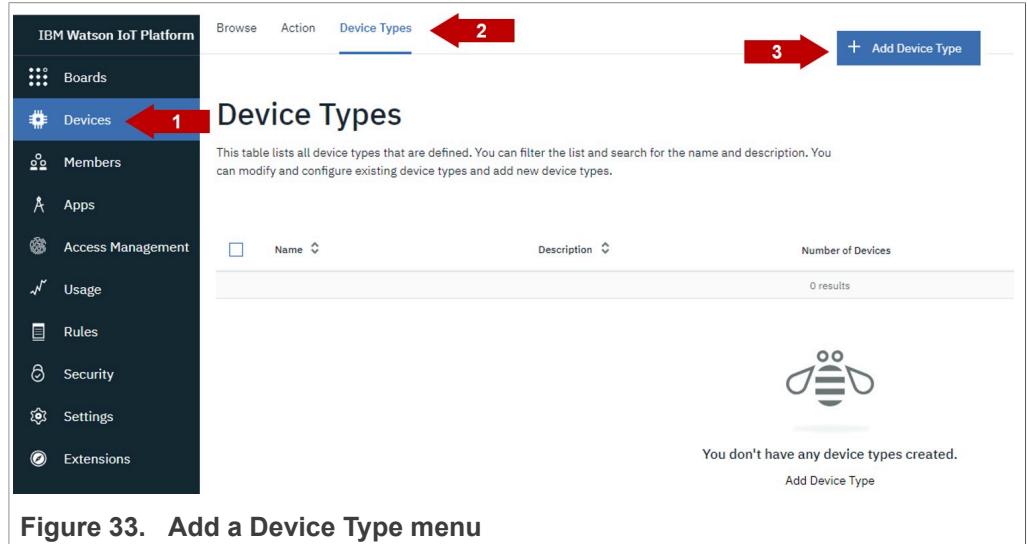


Figure 33. Add a Device Type menu

2. Select either **Device** or **Gateway** type, enter a device type name: **NXP-SE050-EC-D** and optionally, add description for the device type. Click **Next** as shown in [Figure 34](#):

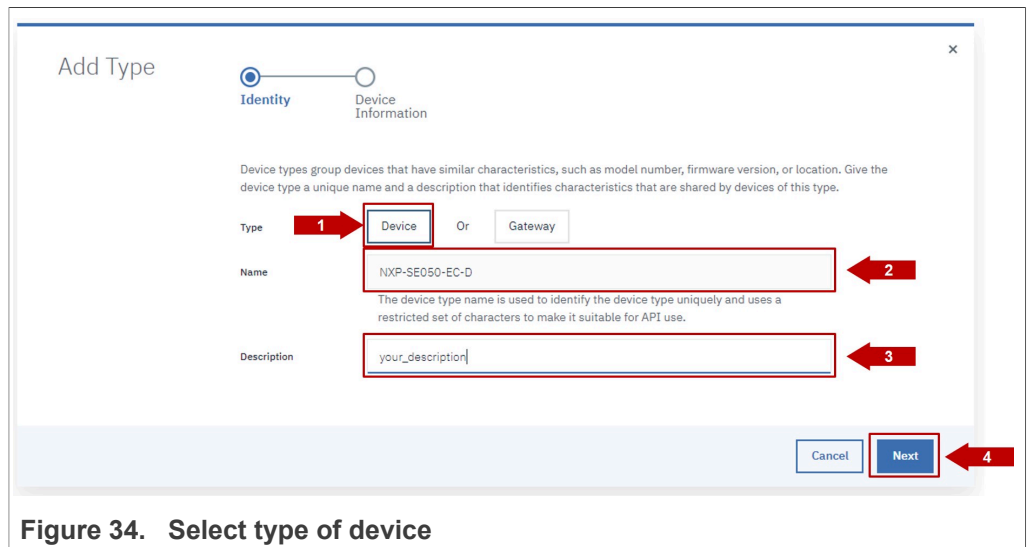


Figure 34. Select type of device

- 3. Optionally, enter device type attributes and metadata and click **Done** as shown in [Figure 35](#):

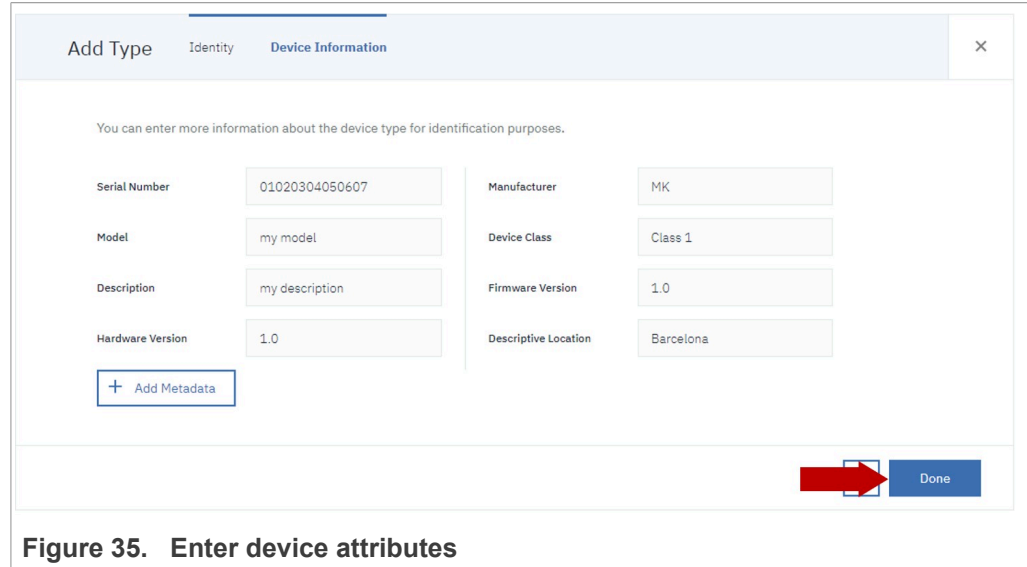


Figure 35. Enter device attributes

- 4. After registering the device type, we need to register the device itself so that it is able to start sending and receiving data. To add a device, from the menu pane, select **Devices**, then click **Add Device** as shown in [Figure 36](#):

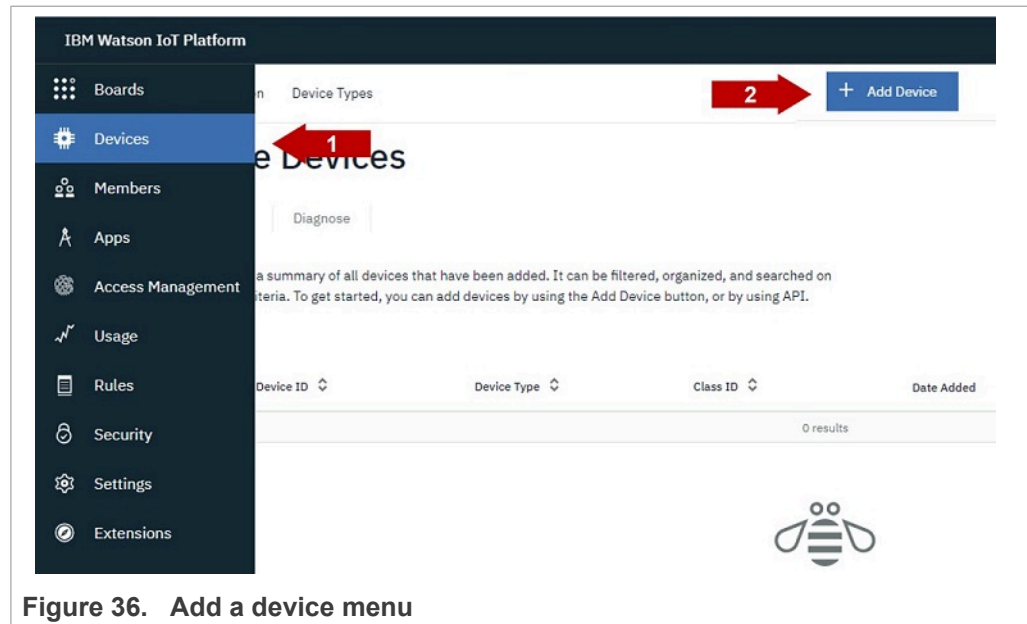
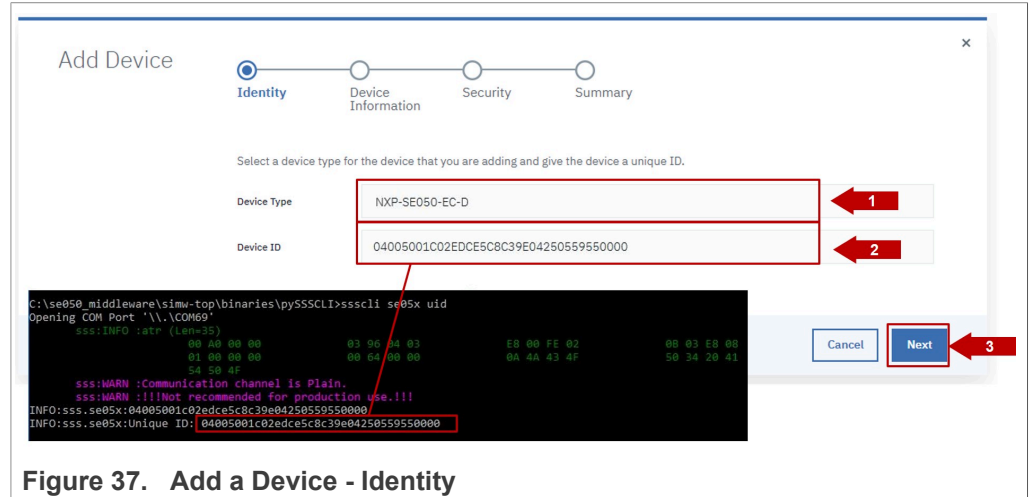


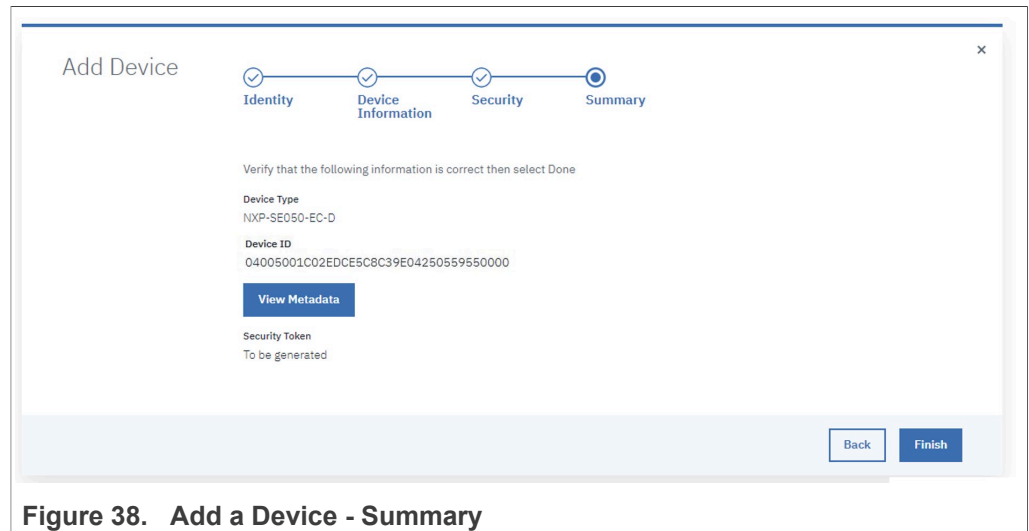
Figure 36. Add a device menu

- 5. In the **Identity** tab, select as **Device Type** the one previously created, in this case *my-device*. Add a **Device ID**, which is used to uniquely identify the device in the Watson

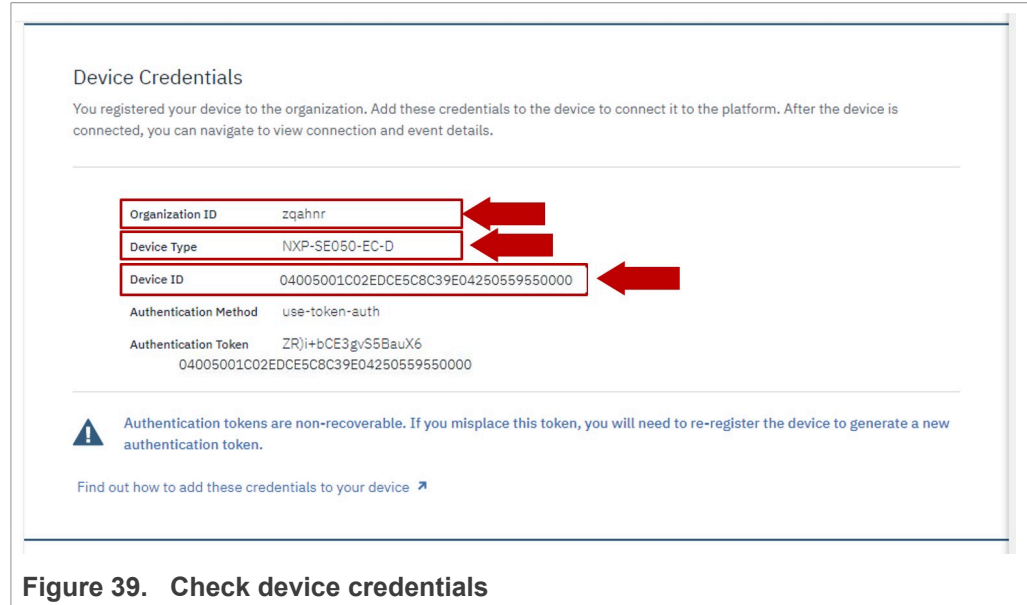
IoT. The device ID is a required parameter for connecting your device to Watson IoT and must be the same obtained in [Section 3.3](#). Click **Next** as shown in [Figure 37](#):



- 6. Optionally, you can add more details in **Device Information** and **Security** tabs. For this guide, we intentionally left these fields blank. After that, check the device summary and click **Done** as shown in [Figure 38](#):



- After the device is registered, Watson IoT delivers the device credentials. Add these credentials to the device to connect it to the platform as shown in [Figure 39](#):



**Figure 39. Check device credentials**

In the next boot up, the device will be able to connect and exchange data with the OEM's Watson application.

### 3.6 Watson IoT project execution

To run the Watson IoT project example, we need to update Watson IoT MCUXpresso project with your Watson IoT account details. This section explains what needs to be updated in Watson IoT MCUXpresso project to add your Watson IoT account details.

**Note:** Before running the Watson IoT demo example, you need to have installed MCUXpresso IDE and FRDM-K64F SDK in your local environment and imported the Watson IoT project example. Check [AN12396- Quick start guide to Kinetis K64](#) for detailed instructions on:

- How to install MCUXpresso.
- How to obtain FRDM-K64F SDK.
- How to import FRDM-K64F project examples.

#### 3.6.1 Download and install the FRDM-K64F SDK

The Watson IoT device onboarding project example is included as part of the FRDM-K64F SDK. Install it to your MCUXpresso workspace as shown in [Figure 40](#):

1. Download the FRDM-K64F SDK, publicly available from the [NXP website](#).
2. Drag and drop the FRDM-K64F SDK zip file in the *Installed SDKs* section in the bottom part of the MCUXpresso IDE.

3. Check that the FRDM-K64F SDK is installed successfully.

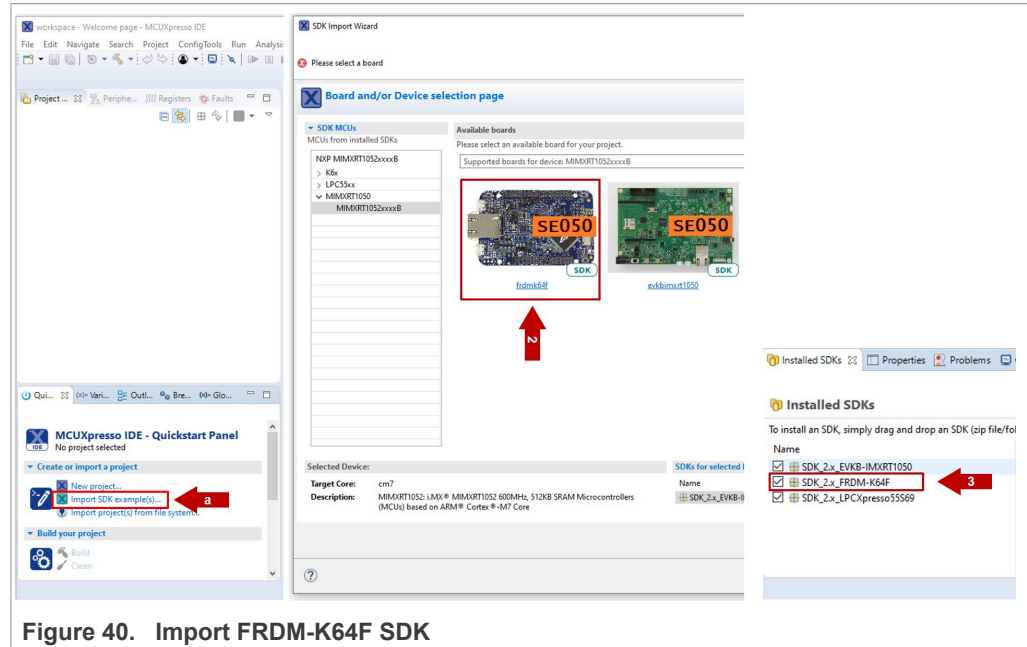


Figure 40. Import FRDM-K64F SDK

### 3.6.2 Import Watson IoT Core project example

The FRDM-K64F SDK includes a project examples called `se_SE050x_cloud_ibm`. Import it to your MCUXpresso workspace as shown in :

1. Click *Import SDK examples* from the MCUXpresso IDE quick start panel.
2. Select `se_SE050x_cloud_ibm` project example and click the *Finish* button.
3. Check the project is now visible in your MCUXpresso workspace

**Note:** For detailed instructions on how to import project examples from FRDM-K64F SDK, check [AN12396 - Quick start guide with Kinetis K64F](#)

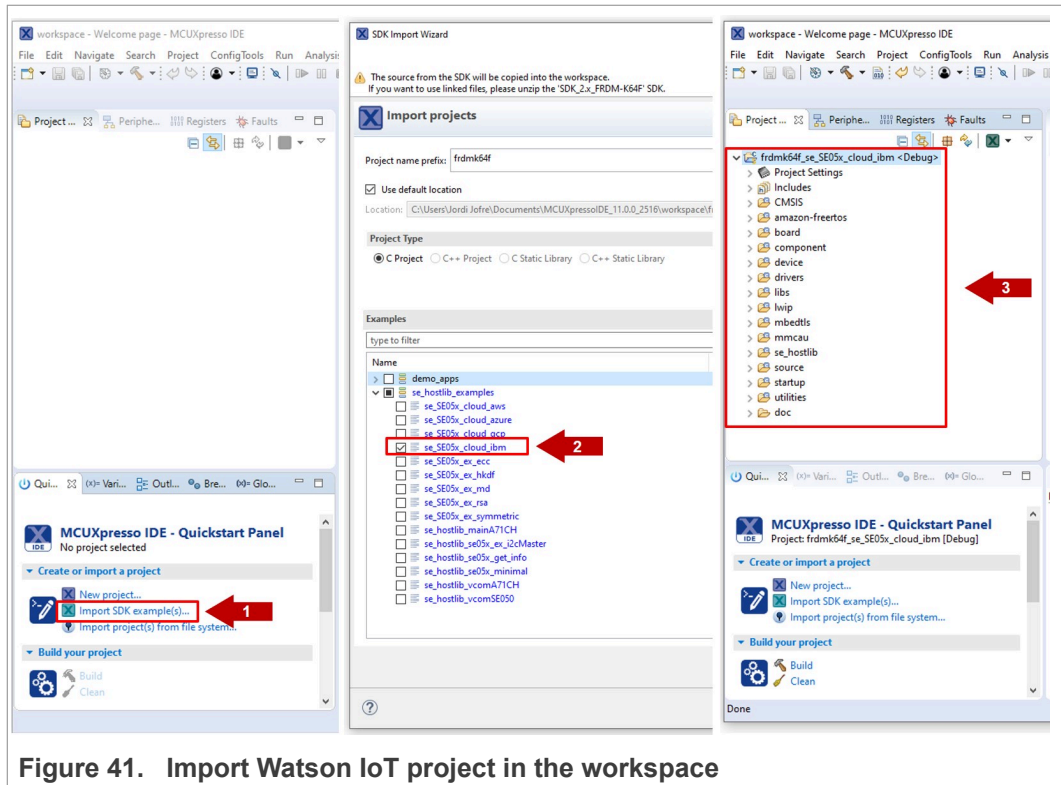


Figure 41. Import Watson IoT project in the workspace

### 3.6.3 Change Watson IoT project account settings

We need to change a few variables in the MCUXpresso Watson IoT demo related with your Watson IoT project account settings. In the MCUXpresso workspace:

EdgeLock™ SE05x for secure connection to IBM Watson IoT

1. Go to frdmk64f\_se\_SE05x\_cloud\_ibm/source folder and open the `ibm_watson_iot_config.h` file as shown in [Figure 42](#):

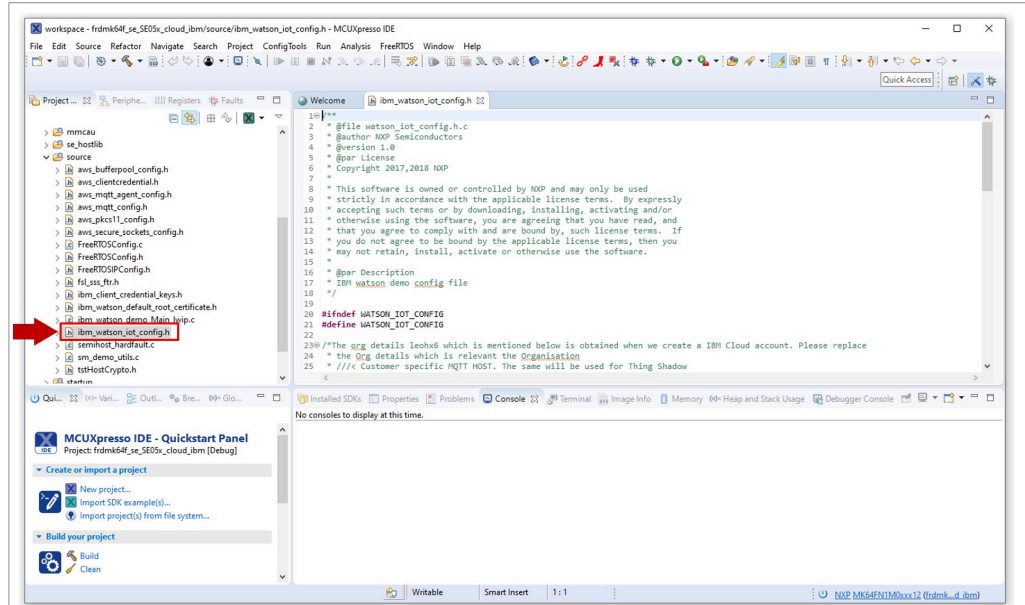


Figure 42. Open `ibm_watson_iot_config.h` file

2. The `#define WATSONIOT_MQTT_BROKER_ENDPOINT` macro has the following format: `organization_id.messaging.internetofthings.ibmcloud.com`. Update the `#define WATSONIOT_MQTT_BROKER_ENDPOINT` macro with the **Organization ID** assigned to your Watson IoT account as shown in [Figure 43](#):

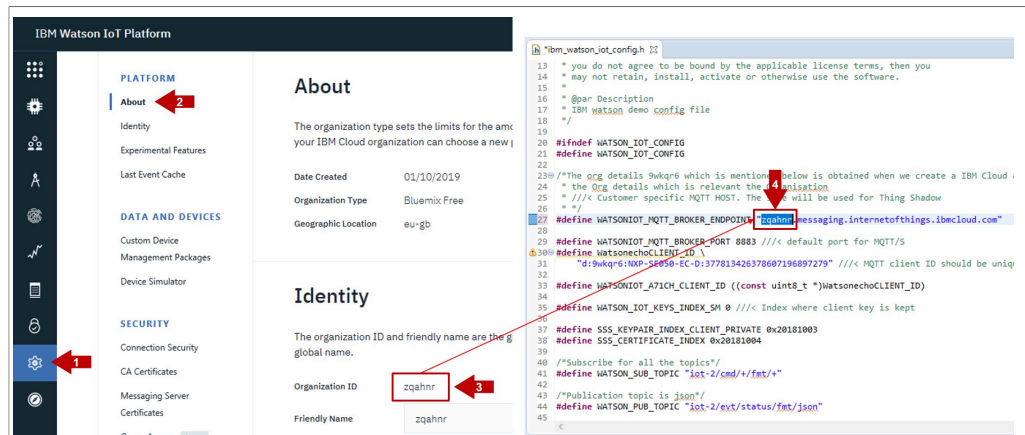


Figure 43. Update your Organization ID

3. The `#define WatsonechoCLIENT_ID` macro has the following format: `d.organization_id:device_type:device_uid`. Update the `#define WatsonechoCLIENT_ID` macro with the **Organization ID** assigned to your Watson



IoT account, the device\_type and the device UID obtained in [Section 3.3](#) as shown in [Figure 44](#):

**Figure 44. Update your device\_type and device\_uid details**

- Replace the #define SSS\_KEYPAIR\_INDEX\_CLIENT\_PRIVATE variable with the ID of the ECC key pair we are using to connect to Watson (0xF0000100) and the #define SSS\_CERTIFICATE\_INDEX with the ID of the associated certificate (0xF0000101) as shown in [Figure 45](#):

**Figure 45. Change SSS\_KEYPAIR\_INDEX\_CLIENT\_PRIVATE and SSS\_CERTIFICATE\_INDEX variables**



### 3.6.4 Run Watson IoT project example

To run the Watson IoT project example, follow these steps:

1. Connect FRDM-K64F OpenSDA port and K64F port to your laptop. and connect the board to Internet using an Ethernet cable as shown in [Figure 46](#):

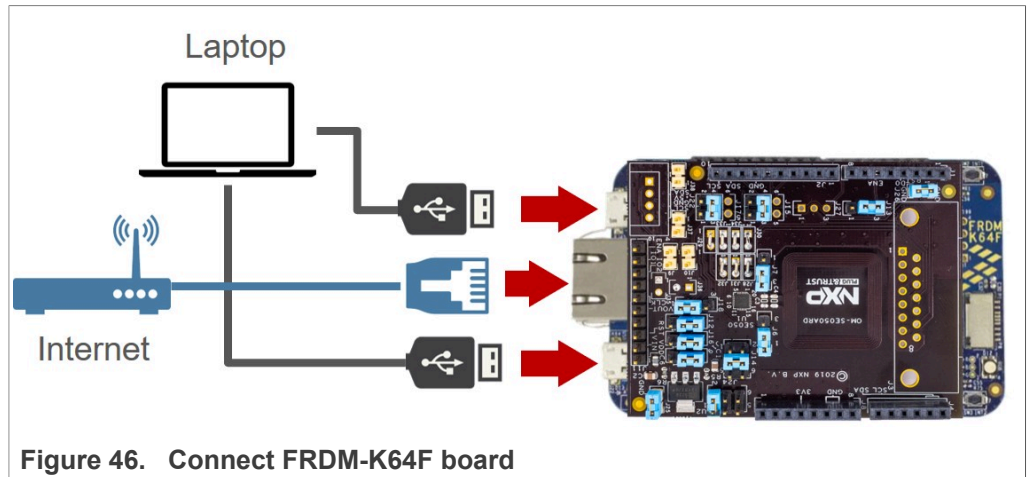


Figure 46. Connect FRDM-K64F board

2. Open TeraTerm, go to Setup > Serial Port and configure the terminal to 115200 baud rate, 8 data bits, no parity and 1 stop bit and click OK as shown in [Figure 47](#):

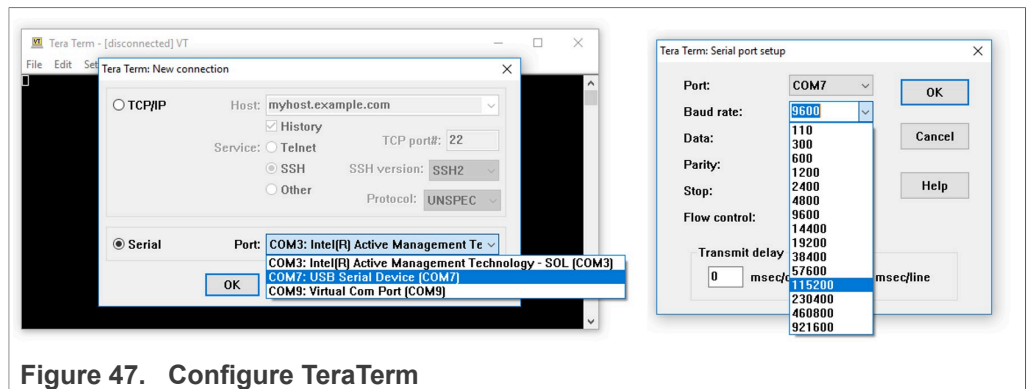


Figure 47. Configure TeraTerm

- 3. Go to the MCUXpresso Quickstart Panel and click **Debug** button, wait a few seconds until the project executes and click on **Resume** to allow the software to continue its execution as shown in [Figure 48](#):

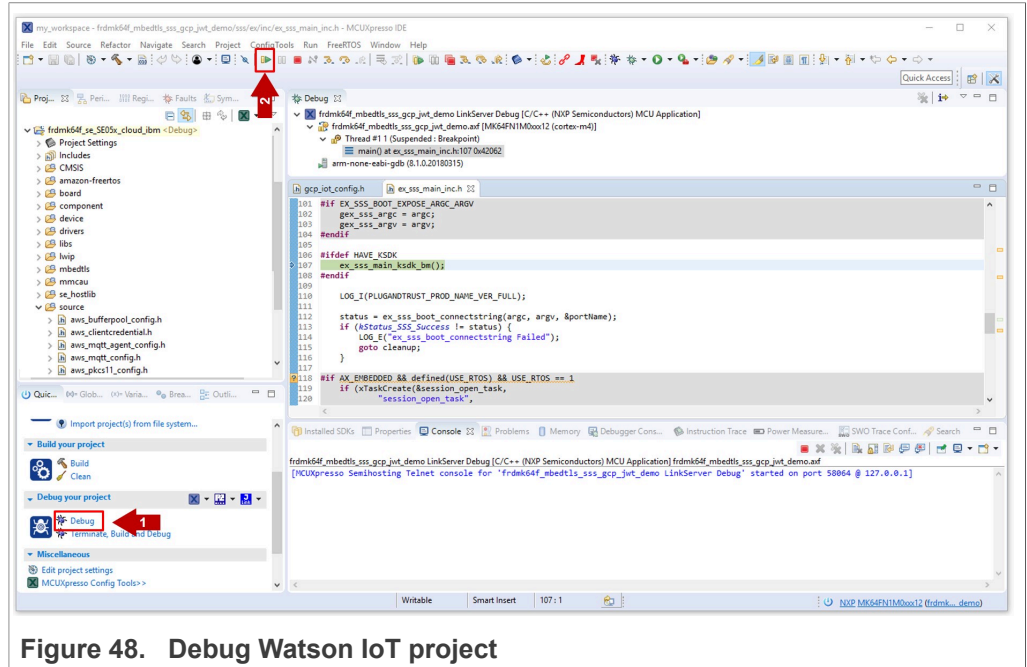


Figure 48. Debug Watson IoT project

- 4. Your device should now be connected to Watson IoT. Check that your device is connected by:
  - a. Checking the TeraTerm logs as shown in [Figure 49](#)

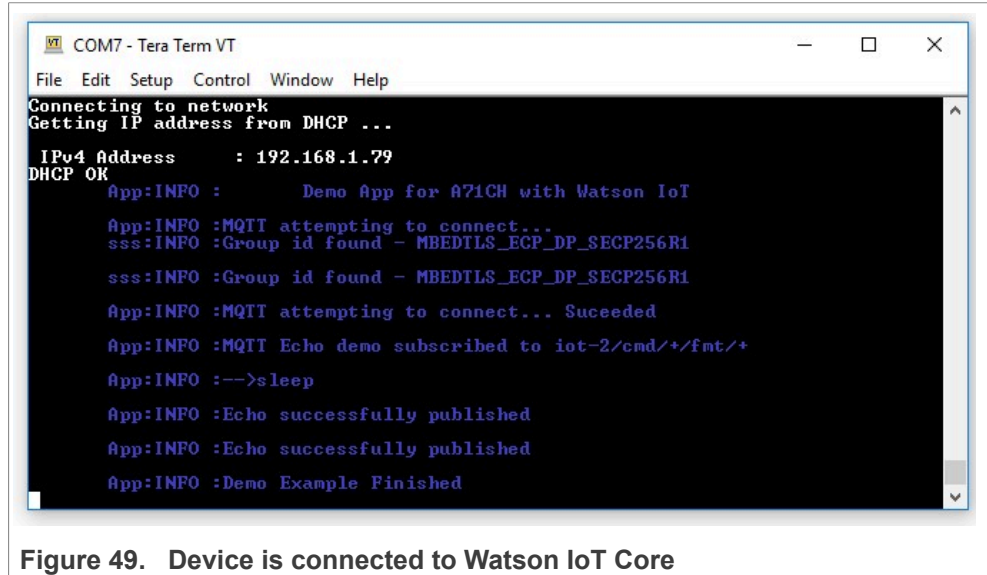


Figure 49. Device is connected to Watson IoT Core

- b. Checking the last time the device was seen in the Watson IoT dashboard as shown in [Figure 50](#)

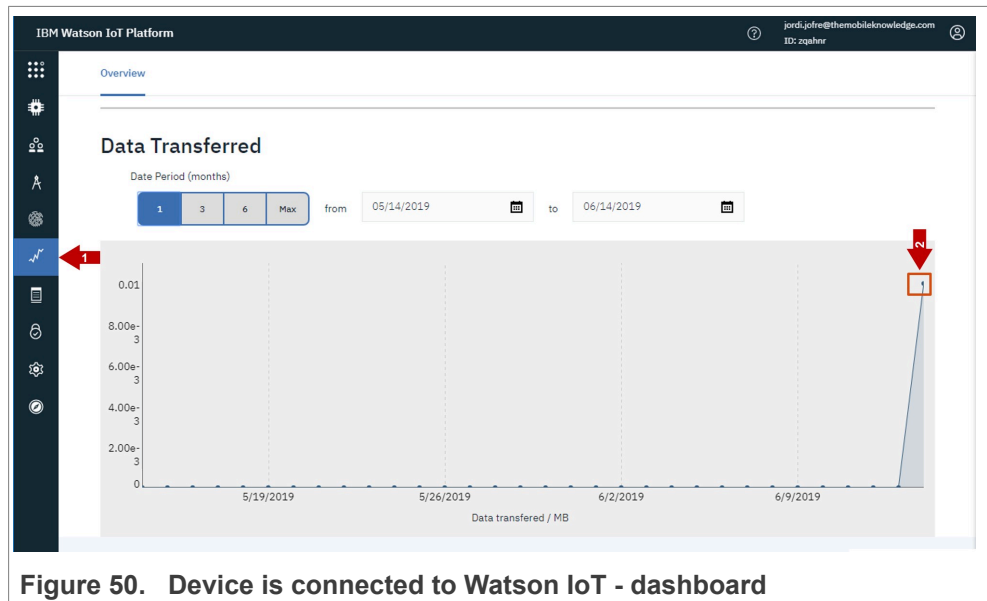


Figure 50. Device is connected to Watson IoT - dashboard

## 4 Appendix: Key provisioning with EdgeLock SE05x Plug & Trust Middleware provisioning scripts

This section explains how to generate and inject your own credentials in EdgeLock SE05x using the provisioning scripts included as part of EdgeLock SE05x Plug & Trust Middleware. Please, use this procedure only if you prefer to generate your own keys instead of leveraging the EdgeLock SE05x ease of use configuration.

**Note:** Before the key provisioning using FRDM-K64F, you need to have installed the EdgeLock SE05x Plug & Trust Middleware in your local environment. Check [AN12396-Quick start guide to Kinetis K64](#) for detailed instructions on how to install the EdgeLock SE05x Plug & Trust Middleware in your local environment.

### 4.1 Flash FRDM-K64F with VCOM software

The VCOM software allows the FRDM-K64F board to be used as a bridge between the Windows machine and the EdgeLock SE05x and enables the execution of the EdgeLock SE05x `sscli` tool and other utilities from the laptop. To flash the VCOM software into the FRDM-K64F, follow these steps:

1. Unplug and plug again the USB cable to the openSDA USB port as shown in [Figure 51](#):

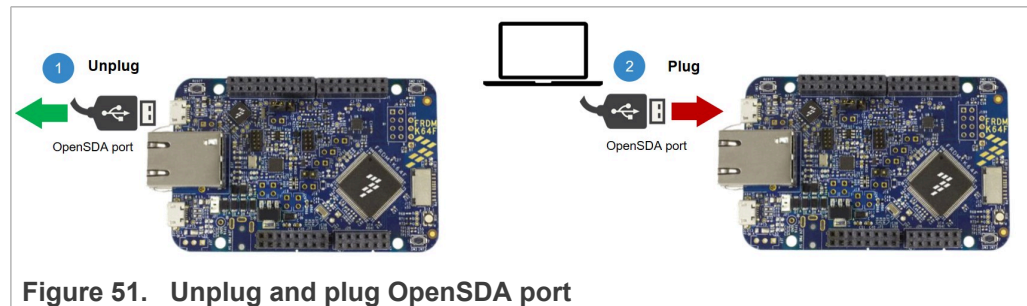


Figure 51. Unplug and plug OpenSDA port

- 2. When you plug the board, your laptop should recognize the board as an external drive as shown in [Figure 52](#):

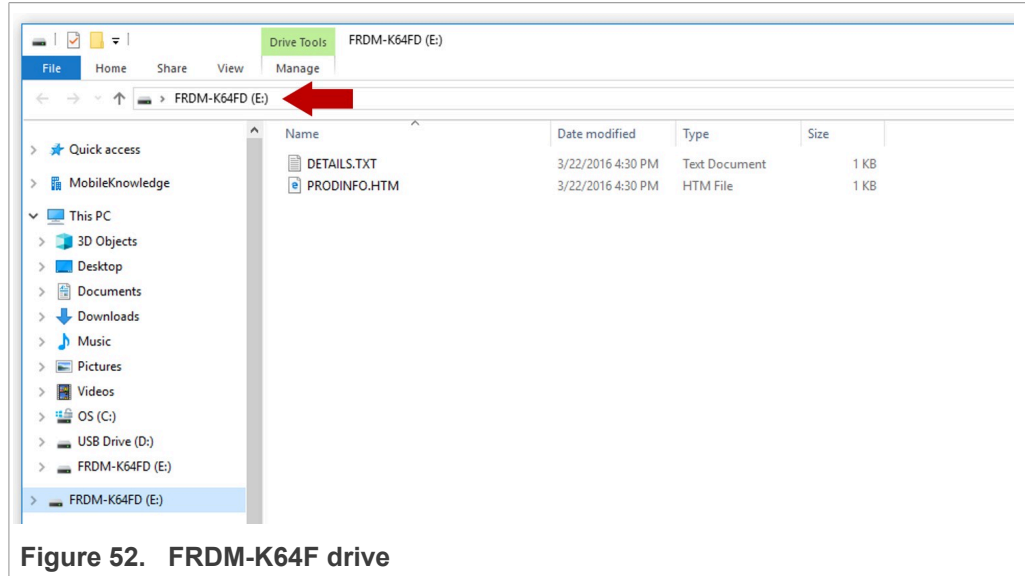


Figure 52. FRDM-K64F drive

- 3. Flash the VCOM software to FRDM-K64F. The VCOM software binary can be found in the EdgeLock SE05x Plug & Trust Middleware package, inside the `simw-top` \binaries folder as shown in [Figure 53](#):

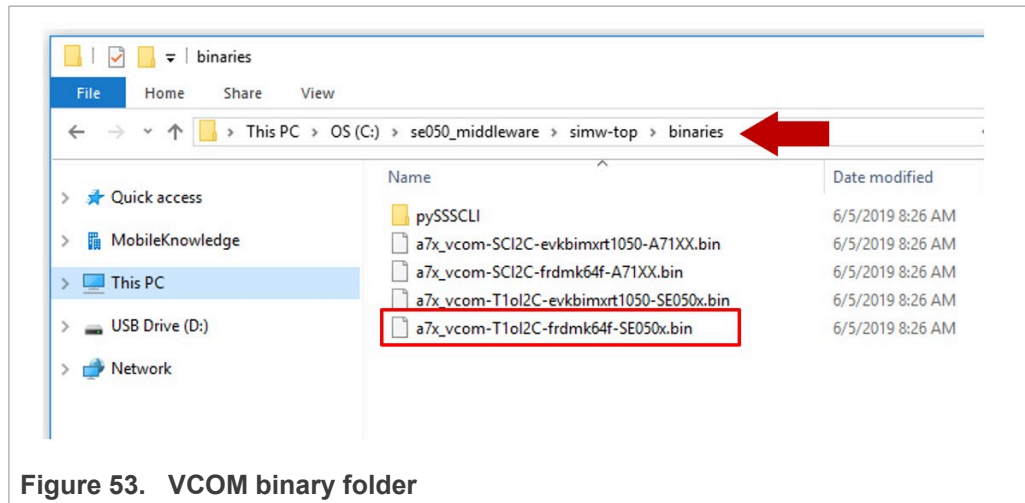


Figure 53. VCOM binary folder

4. Drag and drop or copy and paste the `a7x_vcom-T1oI2C-frdmk64f-SE050x.bin` file into the FRDM-K64F drive from your computer file explorer as shown in [Figure 54](#):

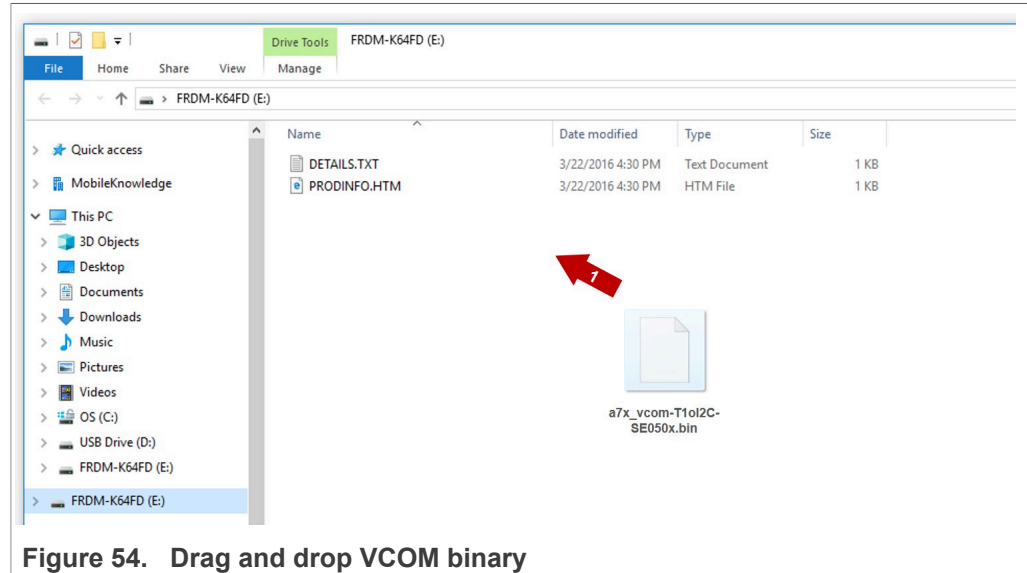


Figure 54. Drag and drop VCOM binary

5. The serial and VCOM ports should be recognized by your Device Manager. To check that the ports are recognized, follow the steps indicated in [Figure 55](#):
  - a. Unplug the USB cable from the OpenSDA USB port.
  - b. Plug the USB cable to the OpenSDA USB port.
  - c. Check that the serial port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as *USB Serial Device (COM7)* but this naming might change depending on your computer. Therefore, it is important that you identify which device is recognized at the moment you plug the SDA USB port to the computer.
  - d. Plug the USB cable to the K64F USB port.
  - e. Check that the VCOM port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as *Virtual Com Port (COM8)* but this naming might change depending on your computer (e.g. It could also appear named as

USB Serial Device). Therefore, it is important that you identify which device is recognized at the moment you plug the K64F USB port to the computer.

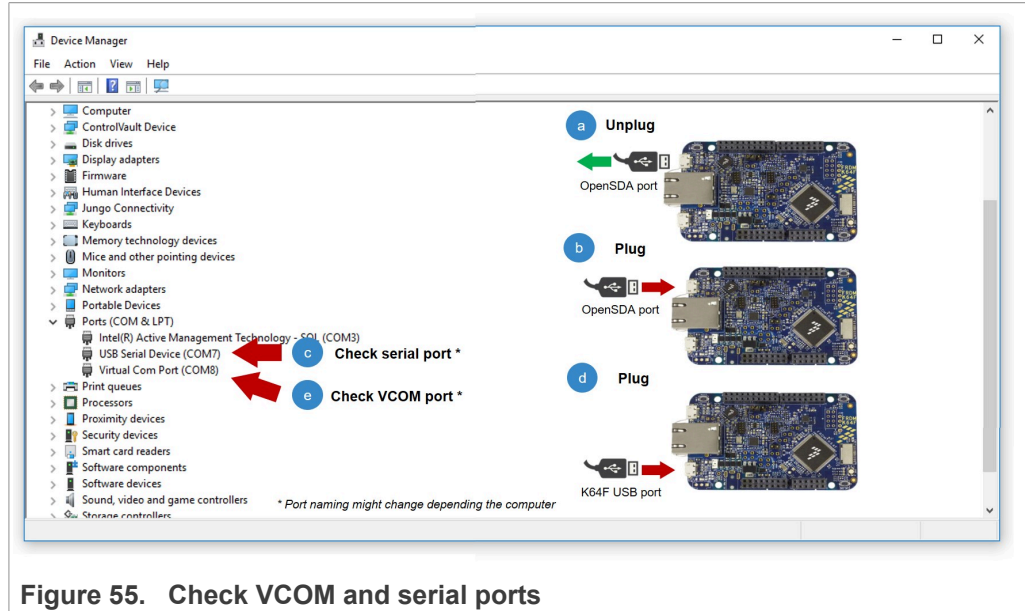


Figure 55. Check VCOM and serial ports

**Note:** Please note that it is possible that either of the two COM ports is not detected when using low-quality or charge-only USB cables.

## 4.2 Running Watson IoT key provisioning script

To run the Watson IoT provisioning script, follow these steps:

1. Open a command prompt and go to the `C:\se050_middleware\simw-top\binaries\pySSSCLI` as shown in [Figure 56](#):  
Send `>cd C:\se050_middleware\simw-top\binaries\pySSSCLI`

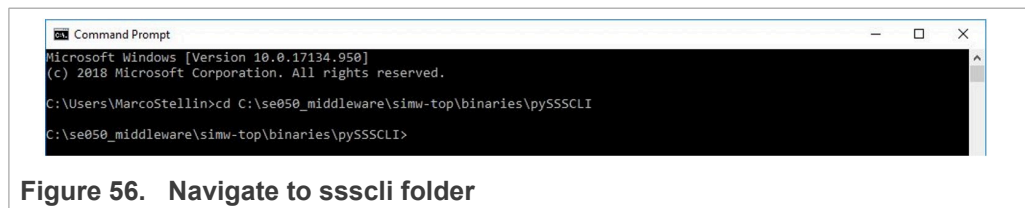


Figure 56. Navigate to ssscli folder



- Run the Provision\_IBM.exe executable as shown in [Figure 57](#):  
Send > Provision\_IBM.exe <VCOM\_NUMBER>

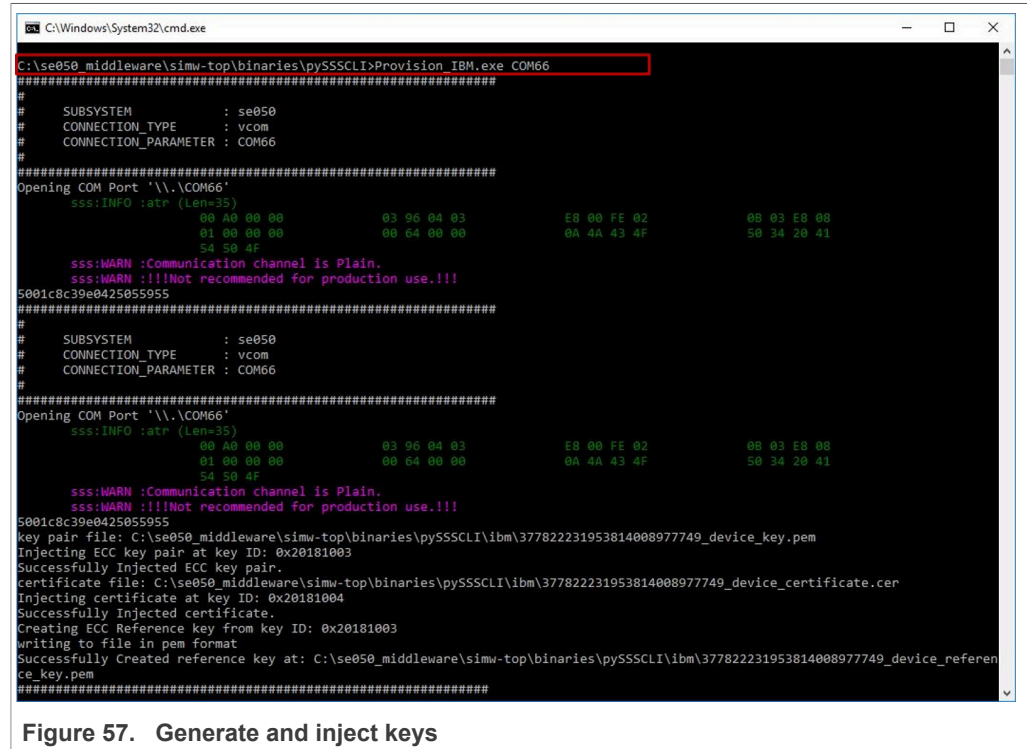


Figure 57. Generate and inject keys

- The key pair and the certificates should have been injected in the EdgeLock SE05x and a copy of the credentials should have been created in the `ibm` folder as shown in [Figure 58](#):

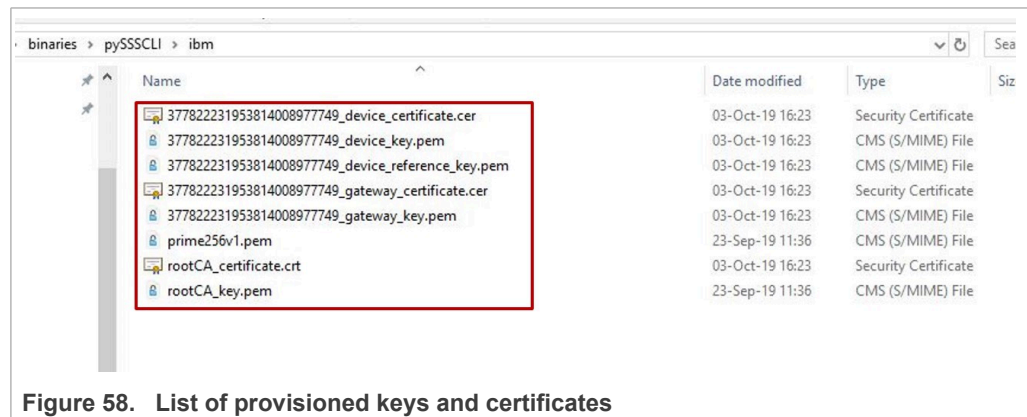
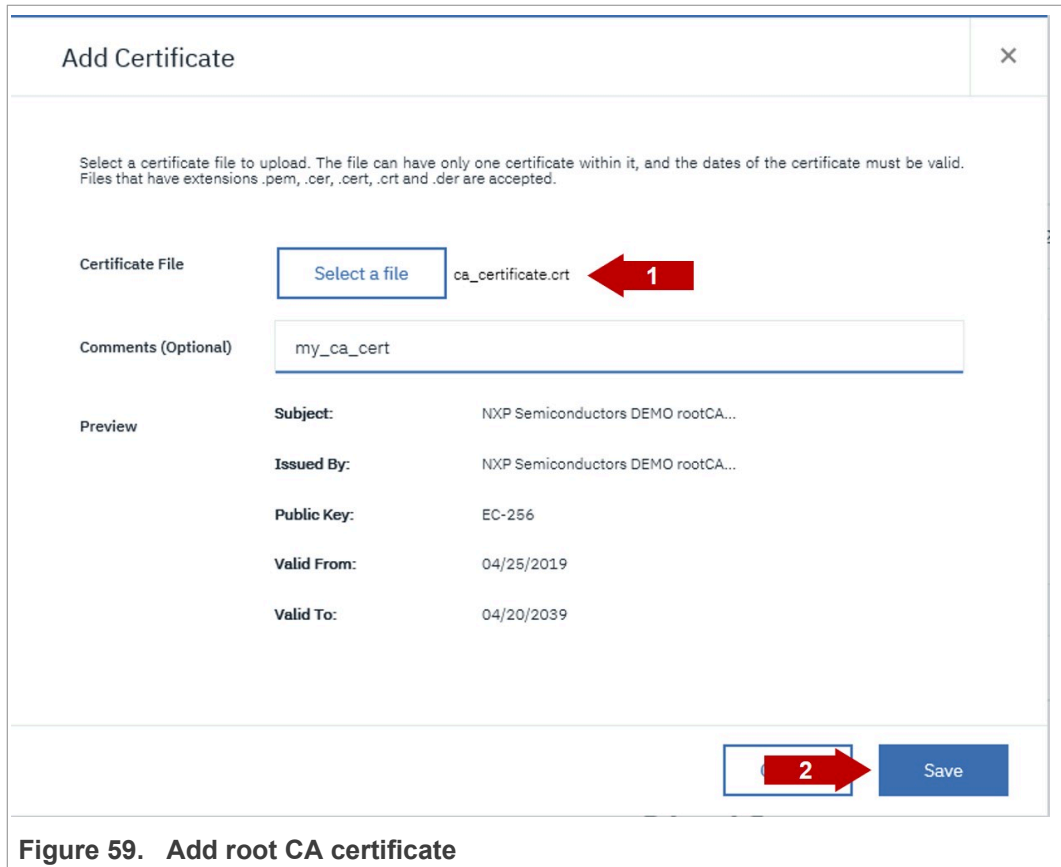


Figure 58. List of provisioned keys and certificates

### 4.3 Register CA certificate for device authentication

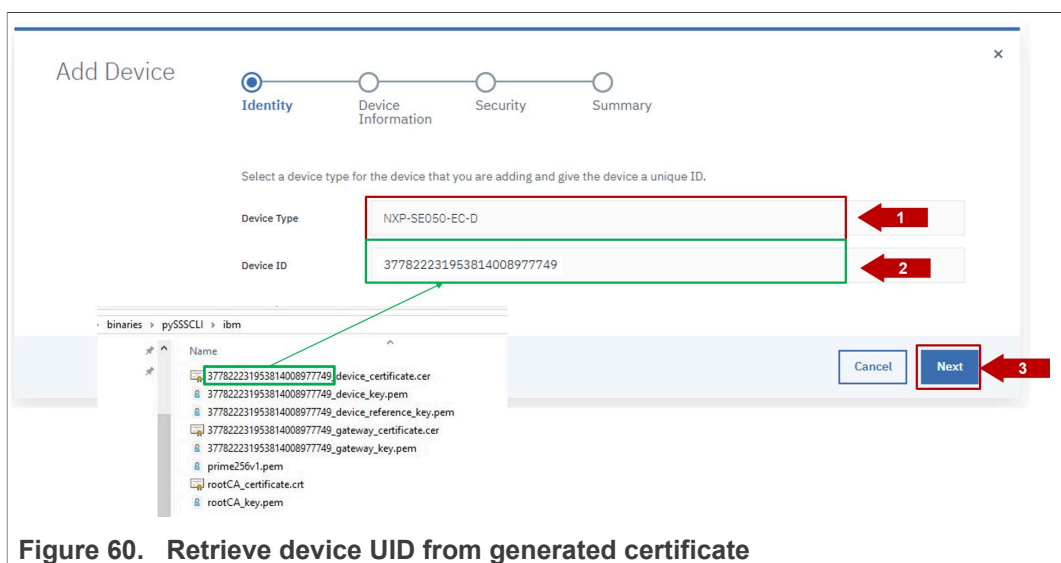
Upload the `rootCA_certificate.crt` in your Watson IoT account. In the Security section, **select CA Certificates**, and click **Add Certificate** window and click **Save** as shown in [Figure 59](#)





#### 4.4 Register your device or gateway to Watson IoT

After injecting the credentials, go to [Section 3.5](#). When asked to upload the CA certificate, select the rootCA\_certificate.crt that should have been generated in C:\se050\_middleware\simw-top\pycli\Provisioning\ibm. The device UID can be obtained as shown in [Figure 60](#):



### 4.5 Change Watson IoT project settings

Finally, go to [Running Watson IoT demos](#). Replace the #define SSS\_KEYPAIR\_INDEX\_CLIENT\_PRIVATE variable with the ID of the injected key pair (0x20181003) and the #define SSS\_CERTIFICATE\_INDEX with the ID of the injected certificate (0x20181004) as shown in [Figure 61](#) :

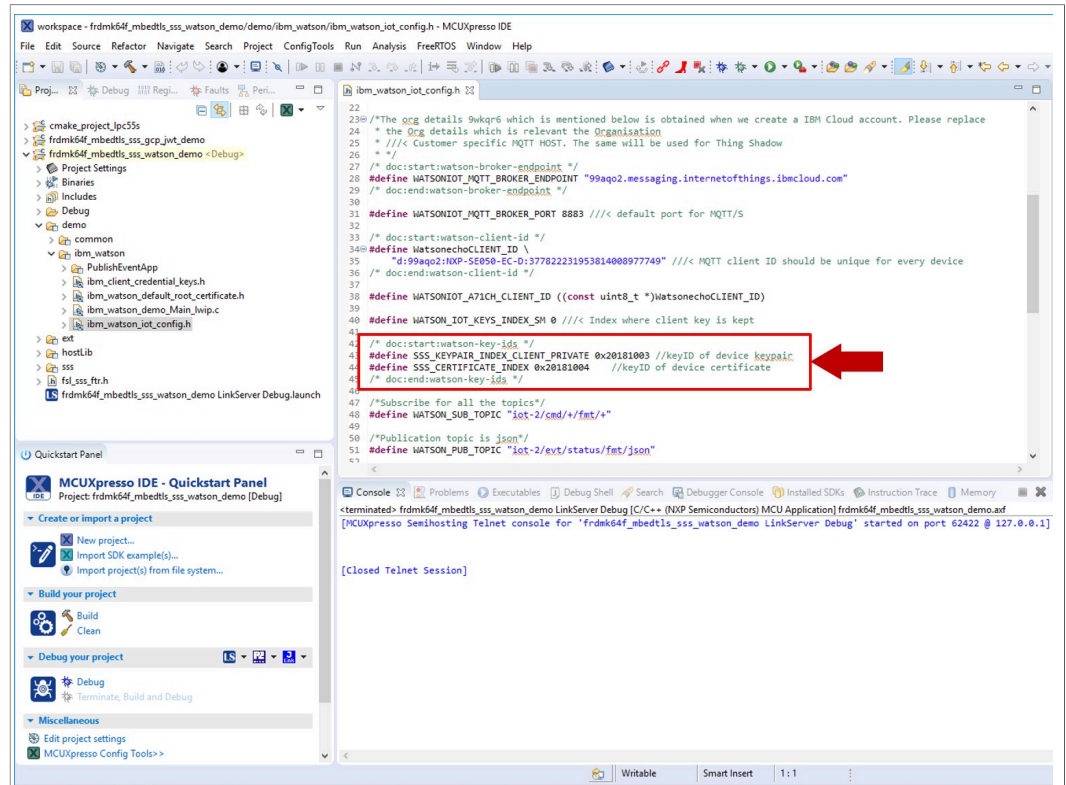


Figure 61. Set SSS\_INDEX\_CLIENT\_PRIVATE and SSS\_CERTIFICATE\_INDEX with injected keys and certificates IDs

## 5 Legal information

### 5.1 Definitions

**Draft** — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

### 5.2 Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based

on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Evaluation products** — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out of the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

**Translations** — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

**Security** — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

### 5.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

Tables

Tab. 1.	Abbreviations .....	3	Tab. 4.	Cloud connectivity certificate used for Watson IoT device onboarding .....	7
Tab. 2.	OM-SE050ARD development kit details .....	6			
Tab. 3.	FRDM-K64F details .....	6			

Figures

Fig. 1.	Watson IoT Core device registration flow	5	Fig. 34.	Select type of device	25
Fig. 2.	Create se050_middleware folder	8	Fig. 35.	Enter device attributes	26
Fig. 3.	Unzip se05x middleware	8	Fig. 36.	Add a device menu	26
Fig. 4.	Unplug and plug OpenSDA port	9	Fig. 37.	Add a Device - Identity	27
Fig. 5.	FRDM-K64F drive	9	Fig. 38.	Add a Device - Summary	27
Fig. 6.	VCOM binary folder	10	Fig. 39.	Check device credentials	28
Fig. 7.	Drag and drop VCOM binary	10	Fig. 40.	Import FRDM-K64F SDK	29
Fig. 8.	Check VCOM and serial ports	11	Fig. 41.	Import Watson IoT project in the workspace	30
Fig. 9.	Connect boards	11	Fig. 42.	Open ibm_watson_iot_config.h file	31
Fig. 10.	Start ssscli tool	12	Fig. 43.	Update your Organization ID	31
Fig. 11.	Close an already opened session	12	Fig. 44.	Update your device_type and device_uid details	32
Fig. 12.	Retrieve SE050 device UID	13	Fig. 45.	Change SSS_KEYPAIR_INDEX_CLIENT_PRIVATE and SSS_CERTIFICATE_INDEX variables	32
Fig. 13.	Disconnect ssscli	13	Fig. 46.	Connect FRDM-K64F board	33
Fig. 14.	NXP certificate chain of trust in EdgeLock SE05x ease of use configuration	14	Fig. 47.	Configure TeraTerm	33
Fig. 15.	Create a folder to store the CA certificates	14	Fig. 48.	Debug Watson IoT project	34
Fig. 16.	Downloaded CA certificates	14	Fig. 49.	Device is connected to Watson IoT Core	35
Fig. 17.	Convert ECC CA certificates	15	Fig. 50.	Device is connected to Watson IoT - dashboard	35
Fig. 18.	Folder containing the CA certificates in PEM format	15	Fig. 51.	Unplug and plug OpenSDA port	36
Fig. 19.	IBM Cloud account sign in	16	Fig. 52.	FRDM-K64F drive	37
Fig. 20.	IBM Cloud account sign in	17	Fig. 53.	VCOM binary folder	37
Fig. 21.	IBM Cloud dashboard landing page	17	Fig. 54.	Drag and drop VCOM binary	38
Fig. 22.	Create a Watson IoT instace	18	Fig. 55.	Check VCOM and serial ports	39
Fig. 23.	Get started with Watson IoT	18	Fig. 56.	Navigate to ssscli folder	39
Fig. 24.	Watson IoT left hand side menu	19	Fig. 57.	Generate and inject keys	40
Fig. 25.	Add a CA certificate	19	Fig. 58.	List of provisioned keys and certificates	40
Fig. 26.	Add root CA certificate to Watson IoT	20	Fig. 59.	Add root CA certificate	41
Fig. 27.	Add intermediateCA1 to Watson IoT	21	Fig. 60.	Retrieve device UID from generated certificate	41
Fig. 28.	Add intermeddiateCA2 to Watson IoT	22	Fig. 61.	Set SSS_INDEX_CLIENT_PRIVATE and SSS_CERTIFICATE_INDEX with injected keys and certificates IDs	42
Fig. 29.	List of uploaded CA certificates	22			
Fig. 30.	Open Connection Security settings	23			
Fig. 31.	Edit Connection Security settings	23			
Fig. 32.	Edit Connection Security default role	24			
Fig. 33.	Add a Device Type menu	25			

## Contents

---

<b>1</b>	<b>EdgeLock SE05x ease of use configuration .....</b>	<b>4</b>
<b>2</b>	<b>Leveraging EdgeLock SE05x ease of use configuration for Watson IoT .....</b>	<b>5</b>
<b>3</b>	<b>Running the Watson IoT device onboarding demo example .....</b>	<b>6</b>
3.1	Hardware required .....	6
3.2	Cloud connectivity certificated used for Watson IoT device onboarding .....	7
3.3	Read EdgeLock SE05x device ID .....	7
3.3.1	Download EdgeLock SE05x Plug & Trust Middleware .....	7
3.3.2	Flash FRDM-K64F with VCOM software .....	9
3.3.3	Read EdgeLock SE05x device ID .....	11
3.4	Obtain NXP certificate chain of trust .....	13
3.4.1	Download and convert the CA certificates .....	14
3.5	Prepare Watson IoT platform .....	15
3.5.1	Create a Watson IoT instance .....	16
3.5.2	Register certificate chain of trust for device authentication .....	18
3.5.3	Configure connection security policy for advanced security .....	22
3.5.4	Register your device or gateway to Watson IoT .....	24
3.6	Watson IoT project execution .....	28
3.6.1	Download and install the FRDM-K64F SDK .....	28
3.6.2	Import Watson IoT Core project example .....	29
3.6.3	Change Watson IoT project account settings .....	30
3.6.4	Run Watson IoT project example .....	33
<b>4</b>	<b>Appendix: Key provisioning with EdgeLock SE05x Plug &amp; Trust Middleware provisioning scripts .....</b>	<b>36</b>
4.1	Flash FRDM-K64F with VCOM software .....	36
4.2	Running Watson IoT key provisioning script .....	39
4.3	Register CA certificate for device authentication .....	40
4.4	Register your device or gateway to Watson IoT .....	41
4.5	Change Watson IoT project settings .....	42
<b>5</b>	<b>Legal information .....</b>	<b>43</b>

---

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

---

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 7 December 2020

Document identifier: AN12403

Document number: 535113