

CONFIGURATION GUIDE

How to Connect
dataFEED OPC Suite to AWS IoT



Table of Contents

1. Preliminary Remarks 1

2. Configure *AWS IoT* Thing for Data Exchange With Softing Gateways..... 1

3. Configure ***dataFEED OPC Suite*** 8

4. Test MQTT Connection and Data Exchange 17



1. Preliminary Remarks

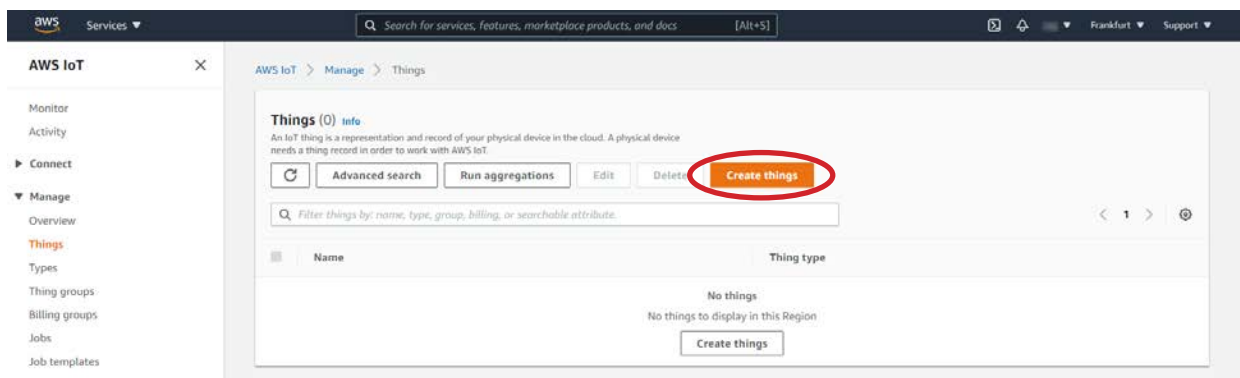
This configuration guide describes how to configure ***dataFEED OPC Suite*** as *AWS IoT* device (“thing”) and thus to transfer shopfloor data to the *AWS IoT* cloud using the MQTT standard.

NOTES:

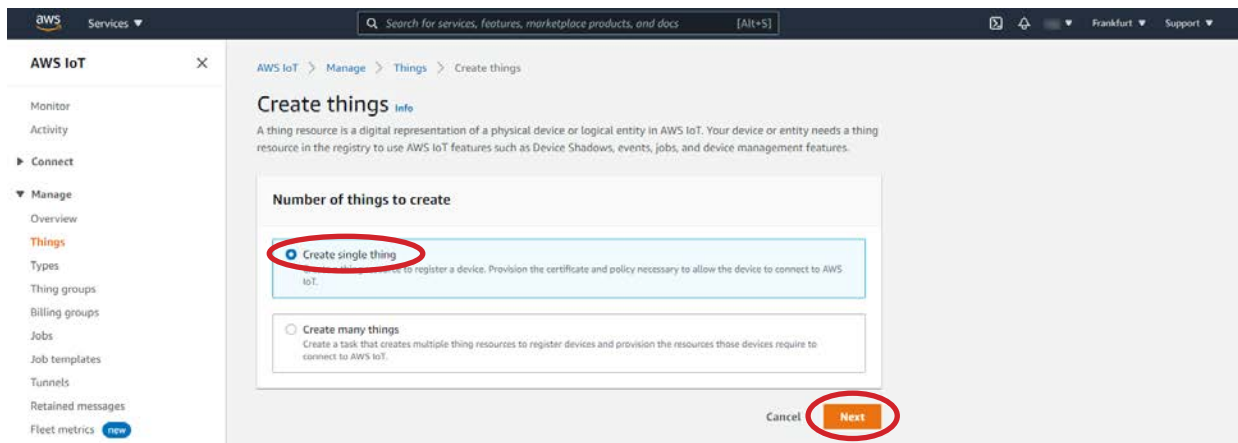
- This document is based on the the *AWS IoT* Developer Guide
<https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>
It uses the *new AWS IoT console experience* as available in December 2021.
- Additional ***dataFEED OPC Suite*** information can be found at the according product web pages.
dataFEED OPC UA Suite Extended: <https://industrial.softing.com/products/opc-opc-ua-software-platform/opc-server-middleware/datafeed-opc-suite-extended.html>
dataFEED OPC UA Suite Base: <https://industrial.softing.com/products/opc-opc-ua-software-platform/opc-server-middleware/datafeed-opc-suite-base.html>

2. Configure *AWS IoT* Thing for Data Exchange With ***dataFEED OPC Suite***

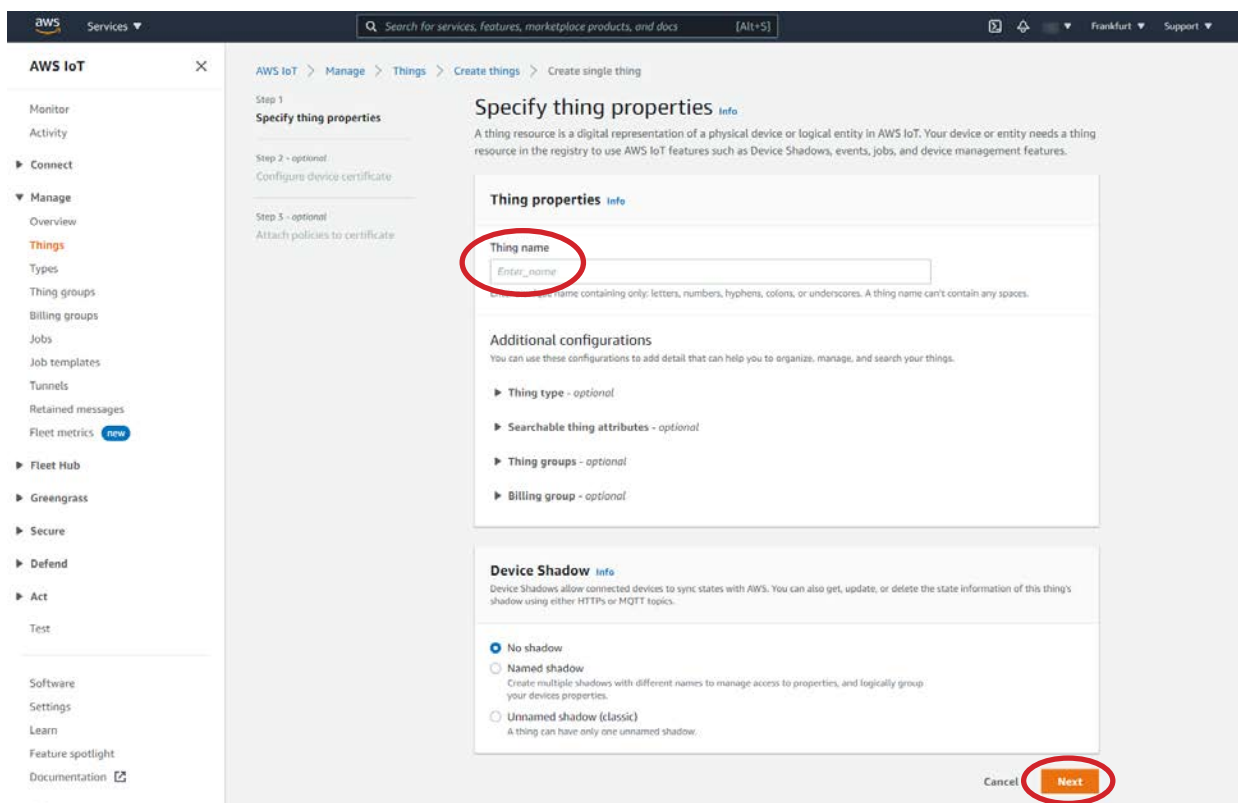
- Set up an *AWS* account, if not yet available
This is done by signing up for an *AWS* account, creating a user and granting the required permissions.
- Open *AWS IoT* console in Internet Browser using URL <https://eu-central-1.console.aws.amazon.com/iot/home> and sign in
- Navigate to *Manage/Things*



- Press *Create things* button



- Select *Create single thing* radio button
- Press *Next* button

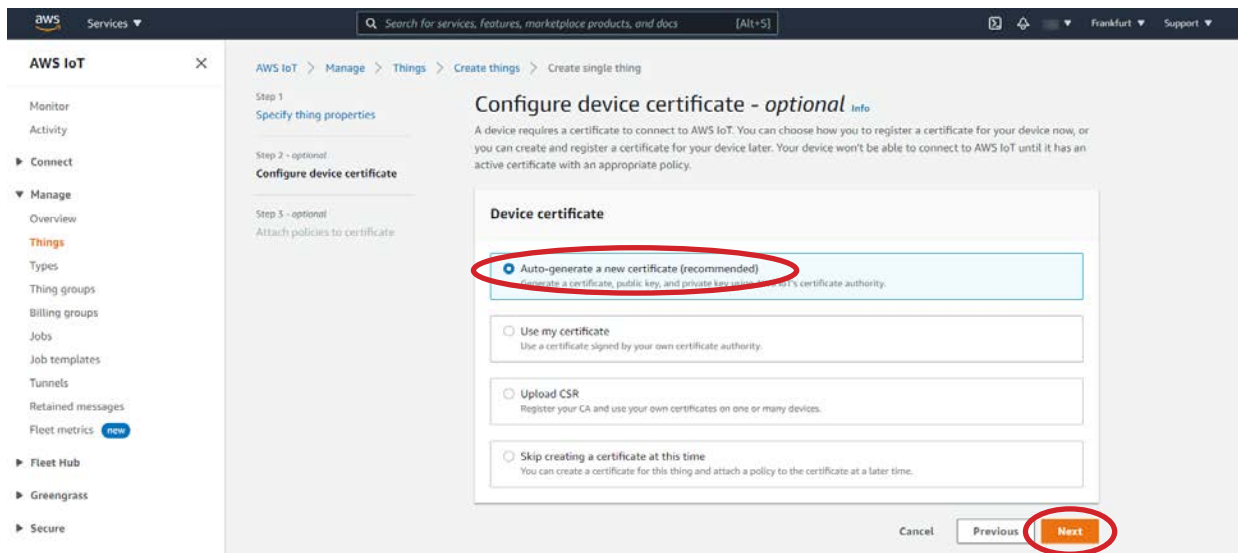


- Enter unique name for accessing Softing gateways in *Thing name* field

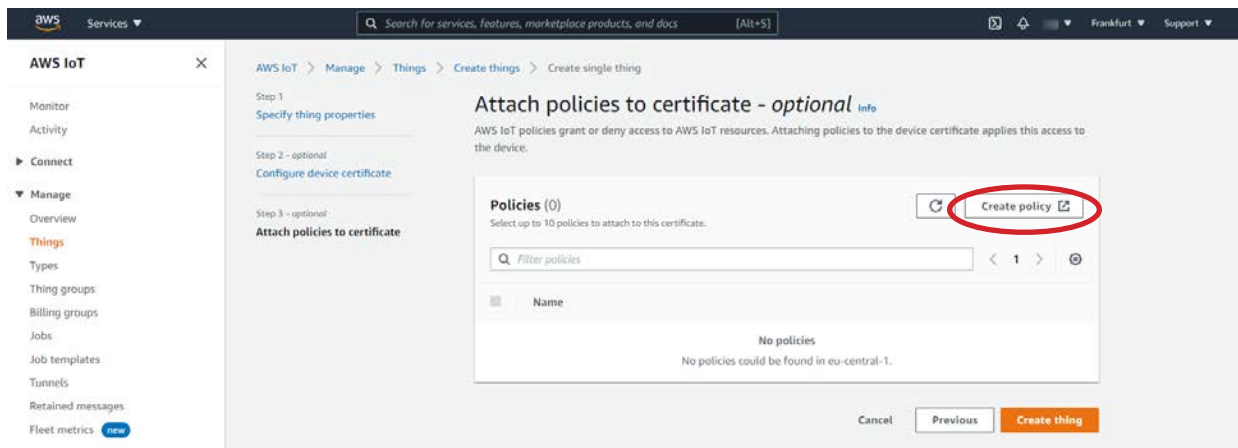
NOTE:

The defined thing name can be used in ***dataFEED OPC Suite*** as *Client ID* when configuring the *MQTT Broker* data destination.

- Press *Next* button



- Select *Auto-generate a new certificate (recommended)* radio button
- Press *Next* button



- Press *Create policy* button

- Enter unique policy name in *Name* field
- Enter action of first policy statement in *Action* field and resource ARN of first policy statement in *Resource ARN* field

NOTE:

While defining the action as *iot:** and the resource ARN as *** grants maximum access capabilities, it is more secure to define the policy statement and the policy resource ARN according to the specific application needs.

- Activate *Allow* checkbox in *Effect* section of statement
- Add additional statements as required
- Press *Create* button once all policy statements are defined

- Press *Create thing* button

Download certificates and keys

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate
You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
816d251bc2418b4abe46082...7ee3fb-te.pem.crt

Deactivate certificate Download

Key files
The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

This is the only time you can download the key files for this certificate.

Public key file
816d251bc2418b4abe46082...47ee3fb-public.pem.key

Download

Private key file
816d251bc2418b4abe46082...7ee3fb-private.pem.key

Download

Root CA certificates
Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1

Download

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

Download

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

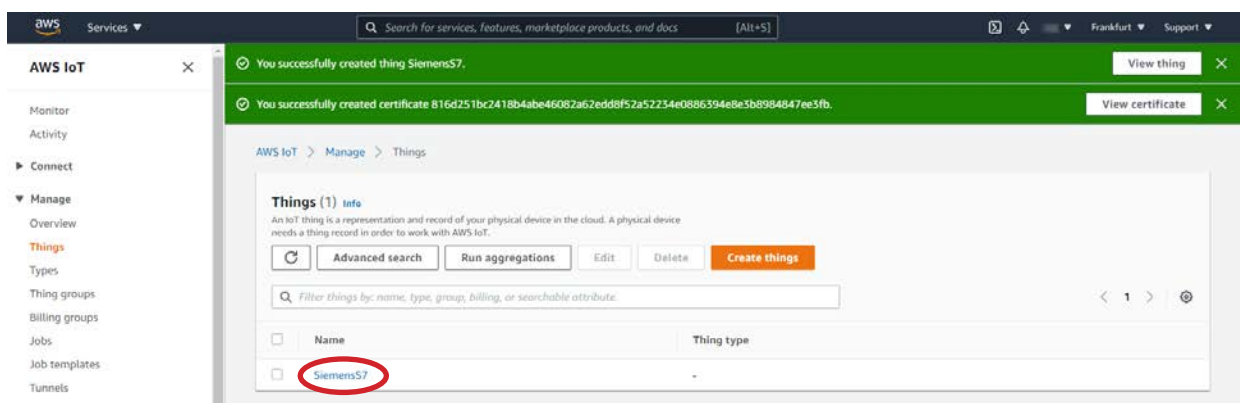
Done

- Press *Download* button for downloading device certificate file
- Press *Download* button for downloading public key file
- Press *Download* button for downloading private key file

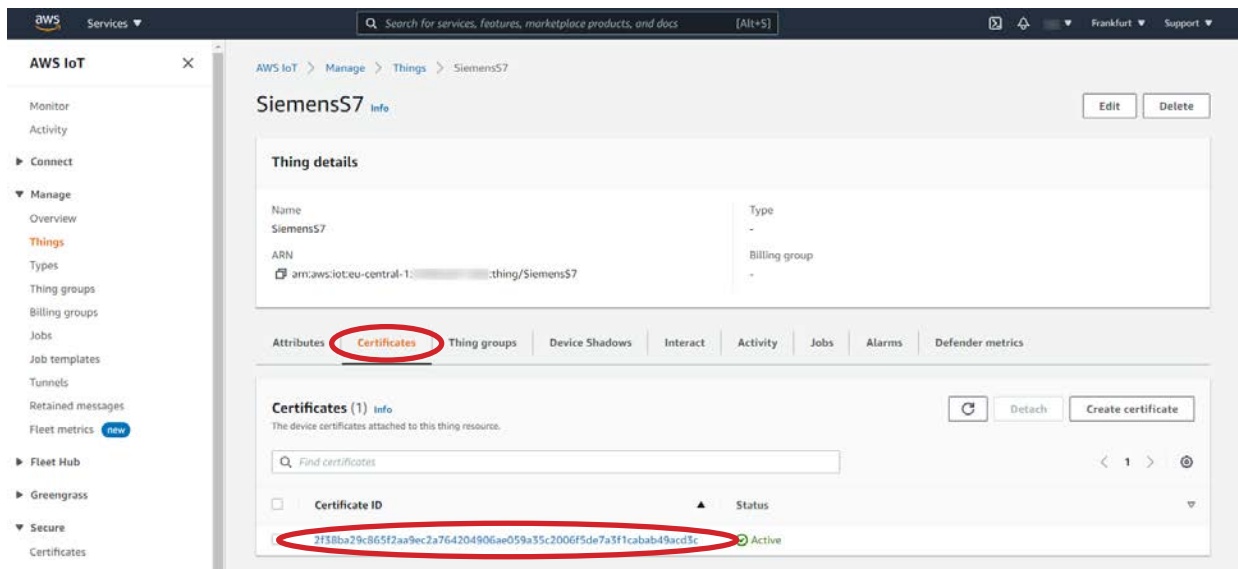
NOTE:

This is the only time you can download the certificate file as well as the public and private key files for the auto-generated certificate. Thus store these at a safe place for later use.

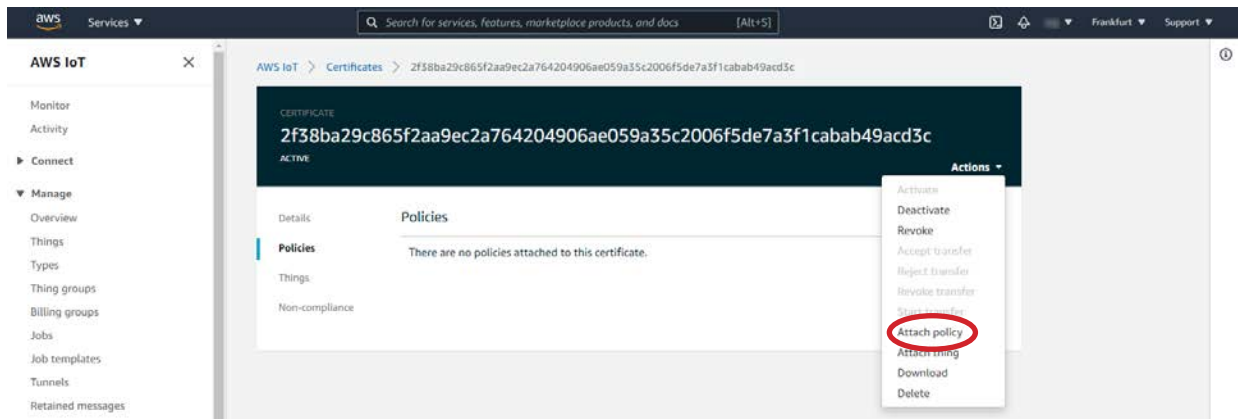
- Press *Done* button



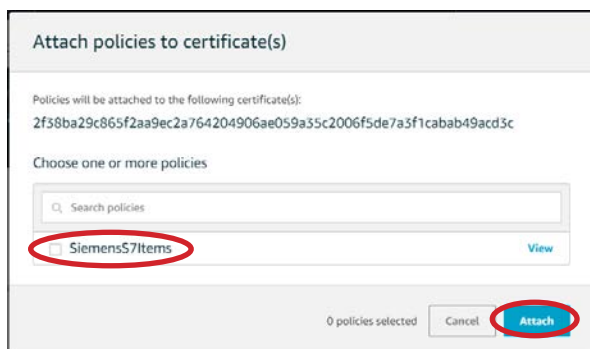
- Click link of created thing



- Click *Certificates* tab
- Click link of auto-generated certificate



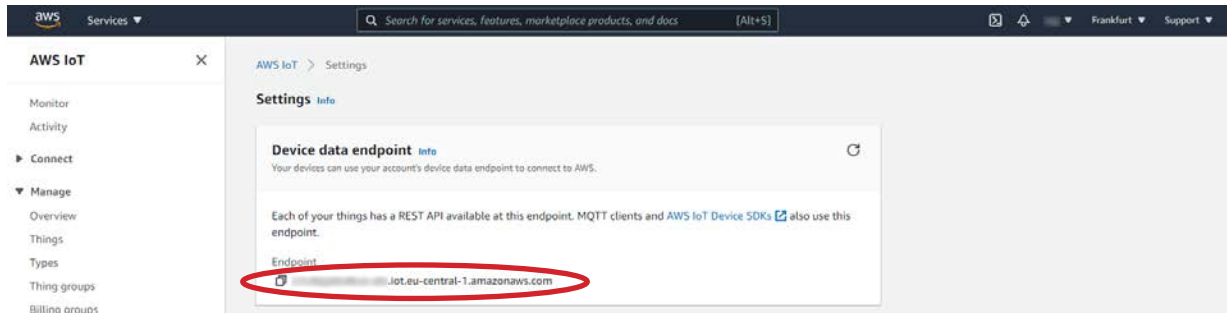
- Open *Actions* menu
- Select *Attach policy* command



Activate checkbox of previously generated policy

Press *Attach* button

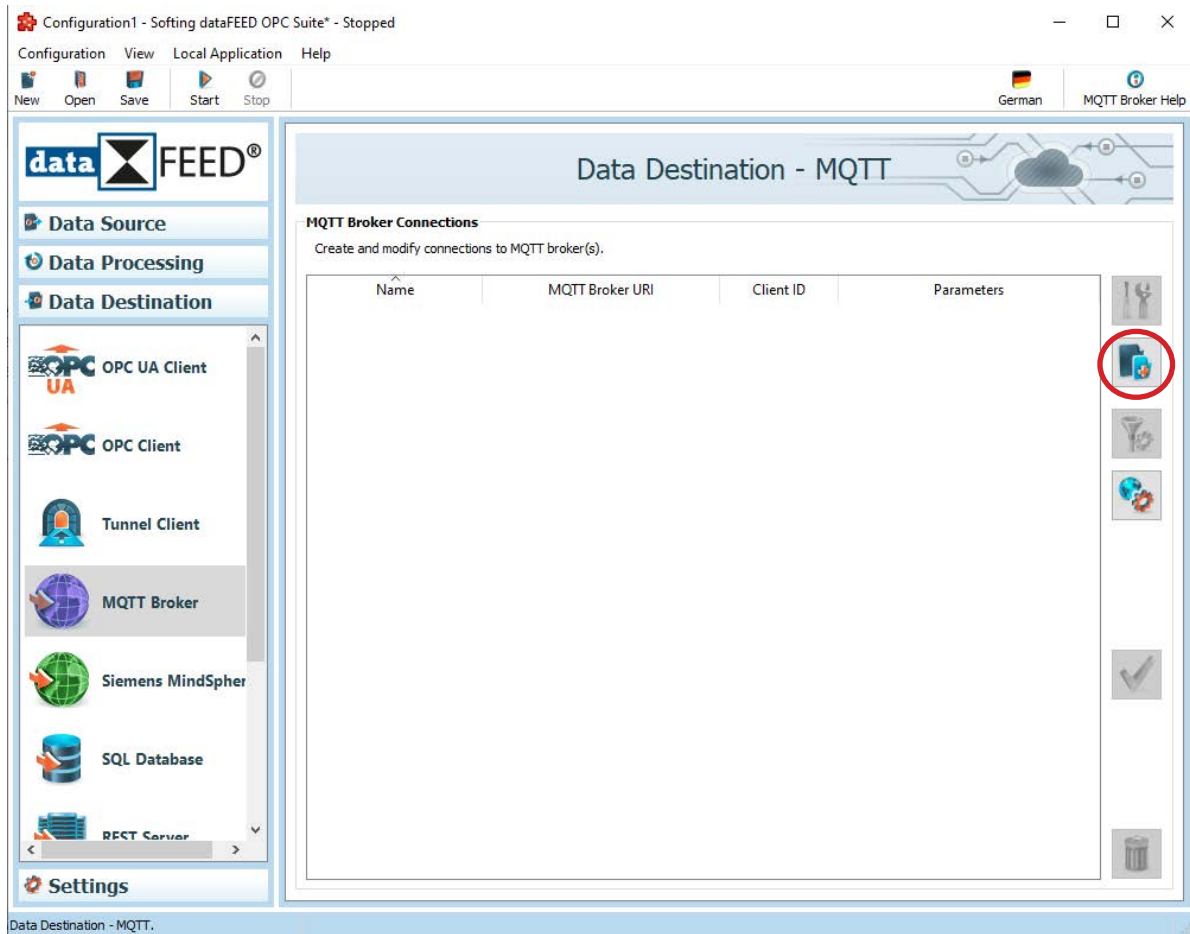
In AWS IoT console navigate to Settings



- Copy *Endpoint* URL for later use

3. Configure dataFEED OPC Suite

- Open **dataFEED OPC Suite** configurator
- Navigate to **Data Destination/MQTT Broker**



- Press  (Add a new data source) button

MQTT Broker Connection Wizard

Connection Settings

On this wizard page the connection settings of the data destination connection to an external MQTT Broker for publishing data are configured.

Connection: MQTT_Broker_Publisher_1

Connection Name
Provide here the connection name which will identify the current connection. The name must be unique throughout the whole configuration.

Connection name: MQTT_Broker_Publisher_1

Client ID

Client ID: b1689fdf-ffa-4161-8ee Generate client ID

Connection State
Specify here if the connection shall be active.

Connection Active: ☒

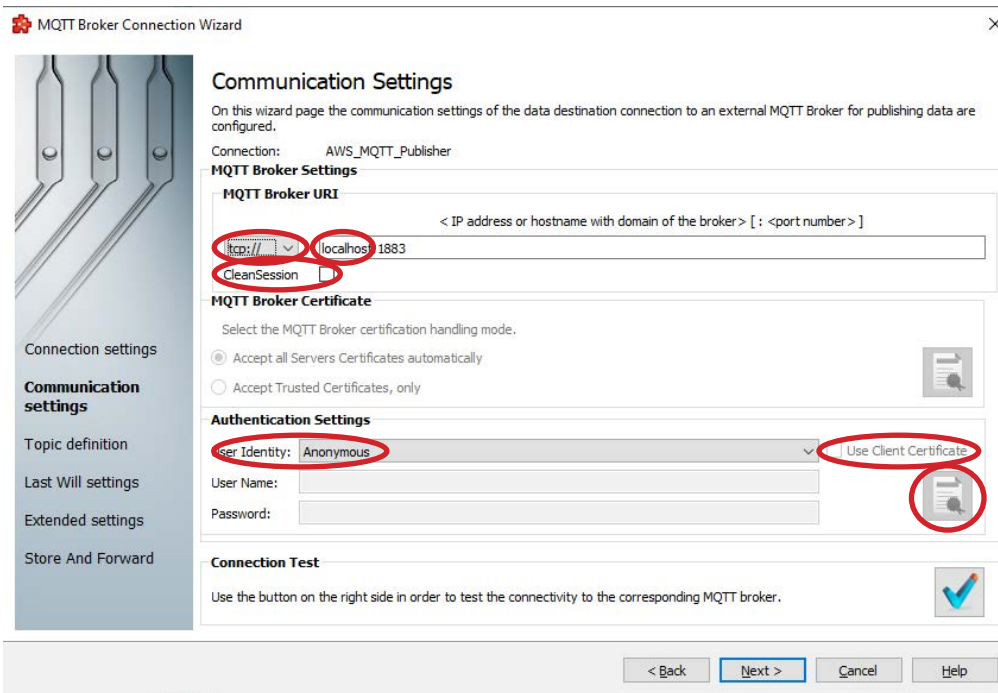
Next > Cancel Help

- Enter unique connection name in *Connection Name* field
- Enter thing name as defined in *AWS IoT* console in *Client ID* field

NOTE:

The client ID has to be allowed in the according *AWS IoT* policy statement.

- Press *Next >* button



MQTT Broker Connection Wizard

Communication Settings

On this wizard page the communication settings of the data destination connection to an external MQTT Broker for publishing data are configured.

Connection: AWS_MQTT_Publisher

MQTT Broker Settings

MQTT Broker URI

< IP address or hostname with domain of the broker > [: <port number>]

ssl:// localhost 1883

☒ CleanSession

MQTT Broker Certificate

Select the MQTT Broker certification handling mode.

☒ Accept all Servers Certificates automatically

☐ Accept Trusted Certificates, only

Authentication Settings

User Identity: Anonymous

☒ Use Client Certificate

User Name:

Password:

Connection Test

Use the button on the right side in order to test the connectivity to the corresponding MQTT broker.


< Back Next > Cancel Help

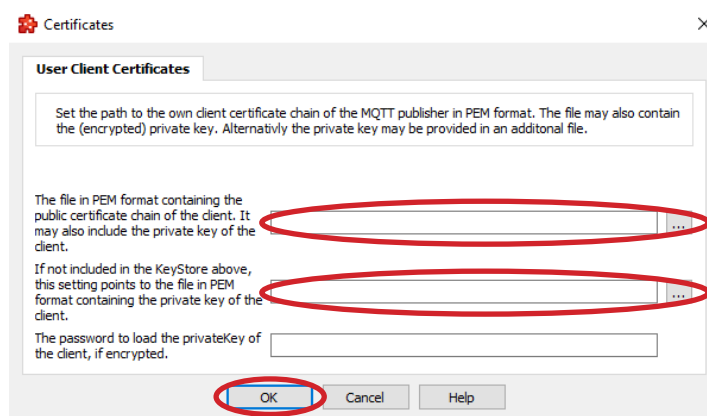
- Select `ssl://` protocol
- Enter copied *AWS IoT* endpoint URL as prefix in
< IP address or hostname with domain of the broker > [: <port number>] field
- Activate *Clean Session* checkbox

NOTE:

AWS IoT closes connection, if clean session flag is not set.

(see <https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html>)

- Select *Anonymous* in *User Identity* field
- Activate *User Client Certificate* checkbox
- Press  (*Import user client certificates*) button



Certificates

User Client Certificates

Set the path to the own client certificate chain of the MQTT publisher in PEM format. The file may also contain the (encrypted) private key. Alternatively the private key may be provided in an additional file.

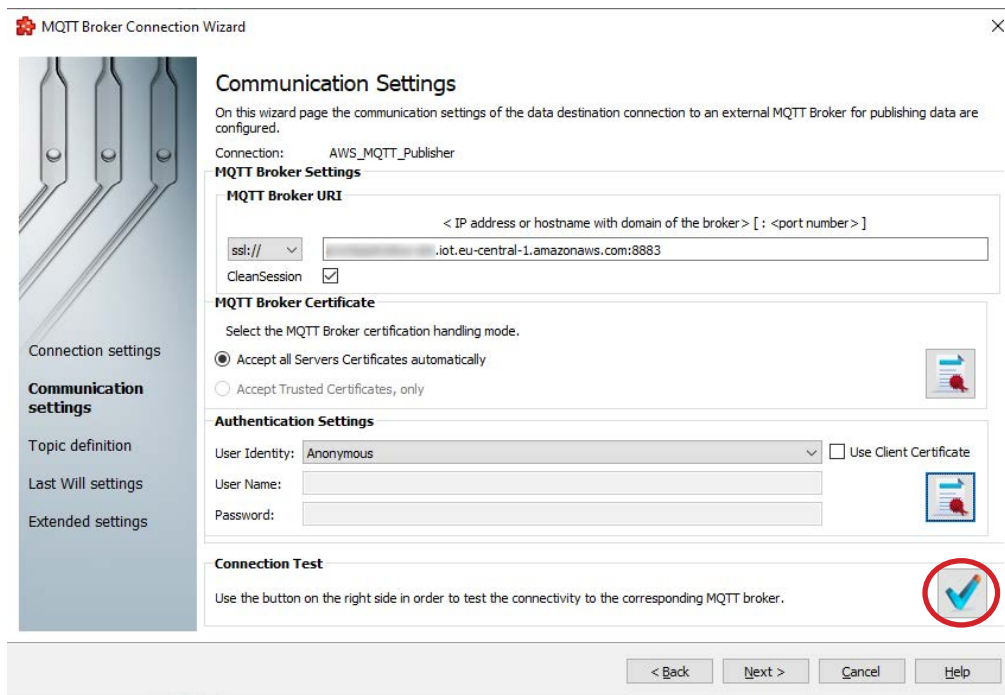
The file in PEM format containing the public certificate chain of the client. It may also include the private key of the client.

If not included in the KeyStore above, this setting points to the file in PEM format containing the private key of the client.

The password to load the privateKey of the client, if encrypted.

OK Cancel Help

- Select downloaded *AWS IoT* device certificate file
- Select downloaded *AWS IoT* public key file
- Press *OK* button



MQTT Broker Connection Wizard

Communication Settings

On this wizard page the communication settings of the data destination connection to an external MQTT Broker for publishing data are configured.

Connection: AWS_MQTT_Publisher

MQTT Broker Settings

MQTT Broker URI

< IP address or hostname with domain of the broker> [: <port number>]

ssl:// .iot.eu-central-1.amazonaws.com:8883

CleanSession ☒

MQTT Broker Certificate

Select the MQTT Broker certification handling mode.

☒ Accept all Servers Certificates automatically

☐ Accept Trusted Certificates, only

Authentication Settings

User Identity: Anonymous ☐ Use Client Certificate


User Name:

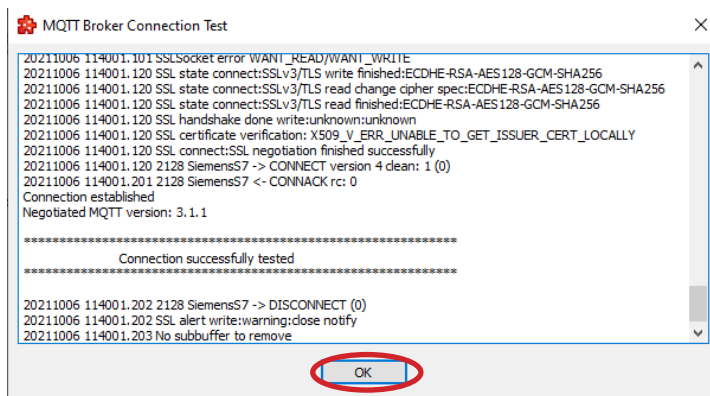
Password:

Connection Test

Use the button on the right side in order to test the connectivity to the corresponding MQTT broker.

< Back Next > Cancel Help

- Press  (Connection test for the selected data destination) button



MQTT Broker Connection Test

```

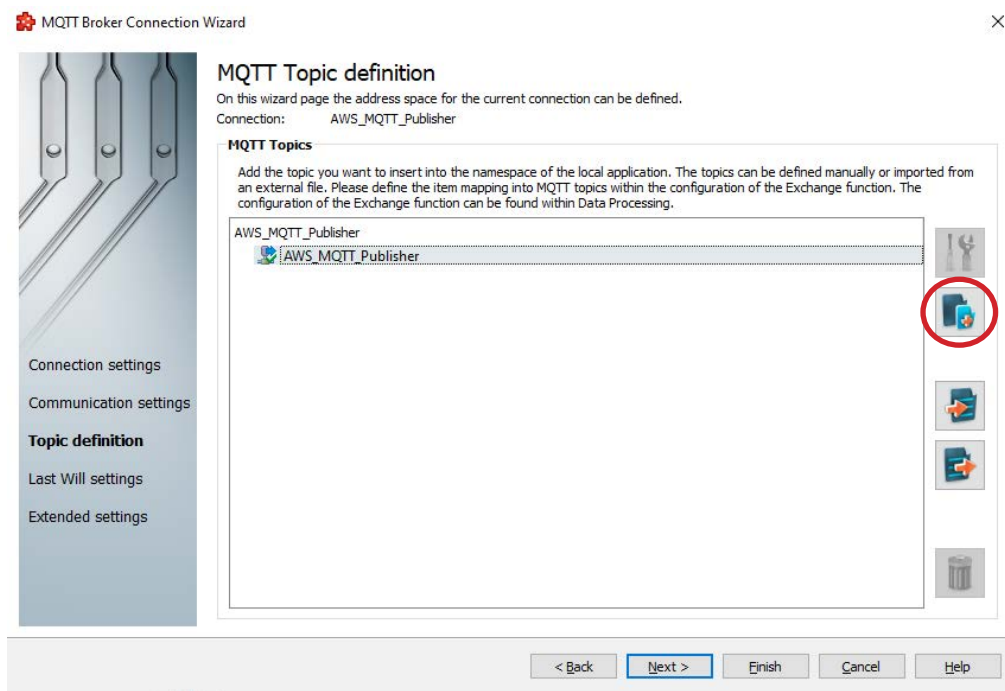
20211006 114001.101 SSLSocket error WANT_READ/WANT_WRITE
20211006 114001.120 SSL state connect:SSLv3/TLS write finished:ECDSA-RSA-AES128-GCM-SHA256
20211006 114001.120 SSL state connect:SSLv3/TLS read change cipher spec:ECDSA-RSA-AES128-GCM-SHA256
20211006 114001.120 SSL state connect:SSLv3/TLS read finished:ECDSA-RSA-AES128-GCM-SHA256
20211006 114001.120 SSL handshake done write:unknown:unknown
20211006 114001.120 SSL certificate verification: X509_V_ERR_UNABLE_TO_GET_ISSUER_CERT_LOCALLY
20211006 114001.120 SSL connect:SSL negotiation finished successfully
20211006 114001.120 2128 SiemensS7 -> CONNECT version 4 clean: 1 (0)
20211006 114001.201 2128 SiemensS7 -> CONNACK rc: 0
Connection established
Negotiated MQTT version: 3.1.1

*****
Connection successfully tested
*****

20211006 114001.202 2128 SiemensS7 -> DISCONNECT (0)
20211006 114001.202 SSL alert write:warning:close notify
20211006 114001.203 No subbuffer to remove
  
```

OK

- Press OK button
- Press Next > button at Communication Settings page



- Press  (*Add a new item*) button to add all items to be used in the MQTT connection with AWS IoT

Topic Properties

Topic Class
Specify the basic class for the current topic. Class Node can have any type of children. Class Tag has a value to publish but no children.
Topic class: Node

Topic Identity
Specify the topic name and properties of the current topic.

Name:

Publish Format: JSON string with Item Identifier, value, timestamp and quality

User defined format string:
`{ "ItemID": "@ITEMID@", "timestamp": "@TIME@", "value": "@VALUE@", "quality": "@QUALITY@" }`

Example for value 1234:
`{ "ItemID": "S7.TestDB.Int", "timestamp": "2016-01-13T 14:07:34:963", "value": "1234", "quality": "good" }`

Retain: ☐

QoS Setting: QoS0: At most once

OK Cancel Help

- Enter item name in *Name* field

NOTE:

The item name has to be allowed in the according *AWS IoT* policy statement.

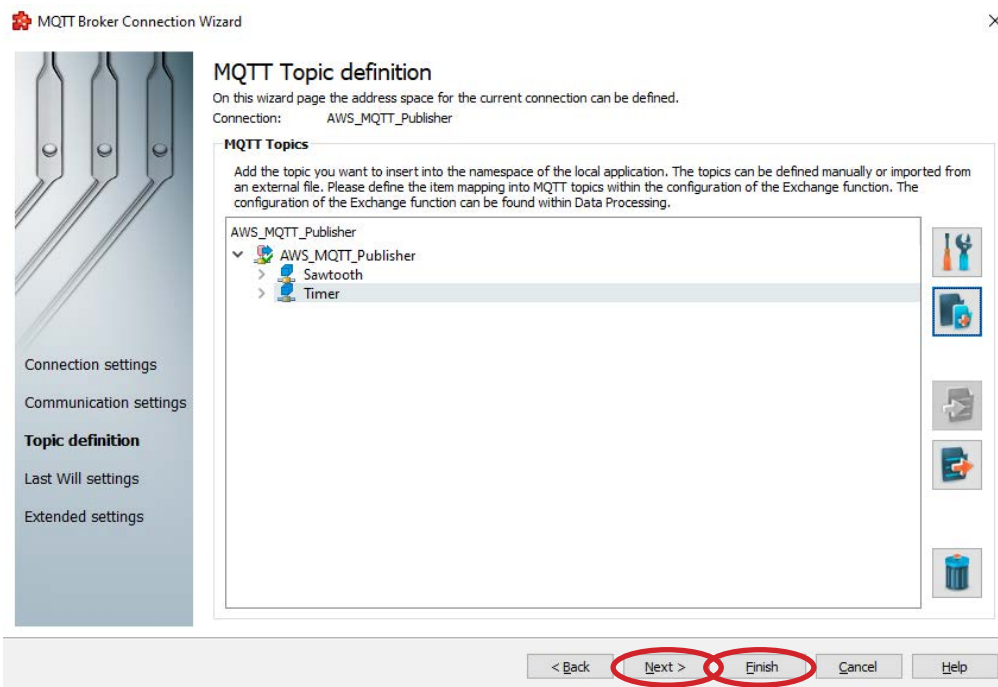
- Select *JSON string* in *Publish Format* field
- Deactivate *Retain* checkbox

NOTE:

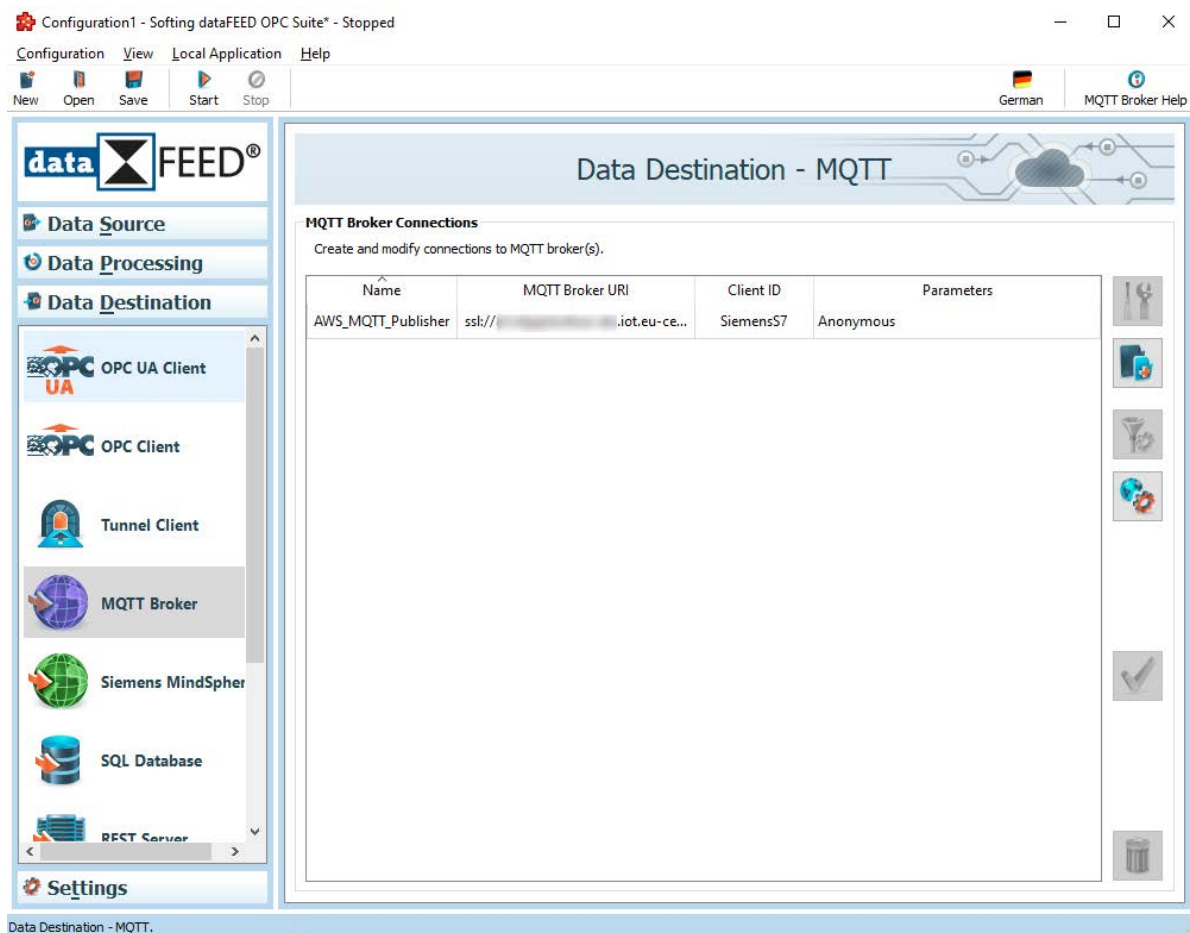
AWS IoT closes connection, if retain flag is set for MQTT Publish message.

(see <https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html>)

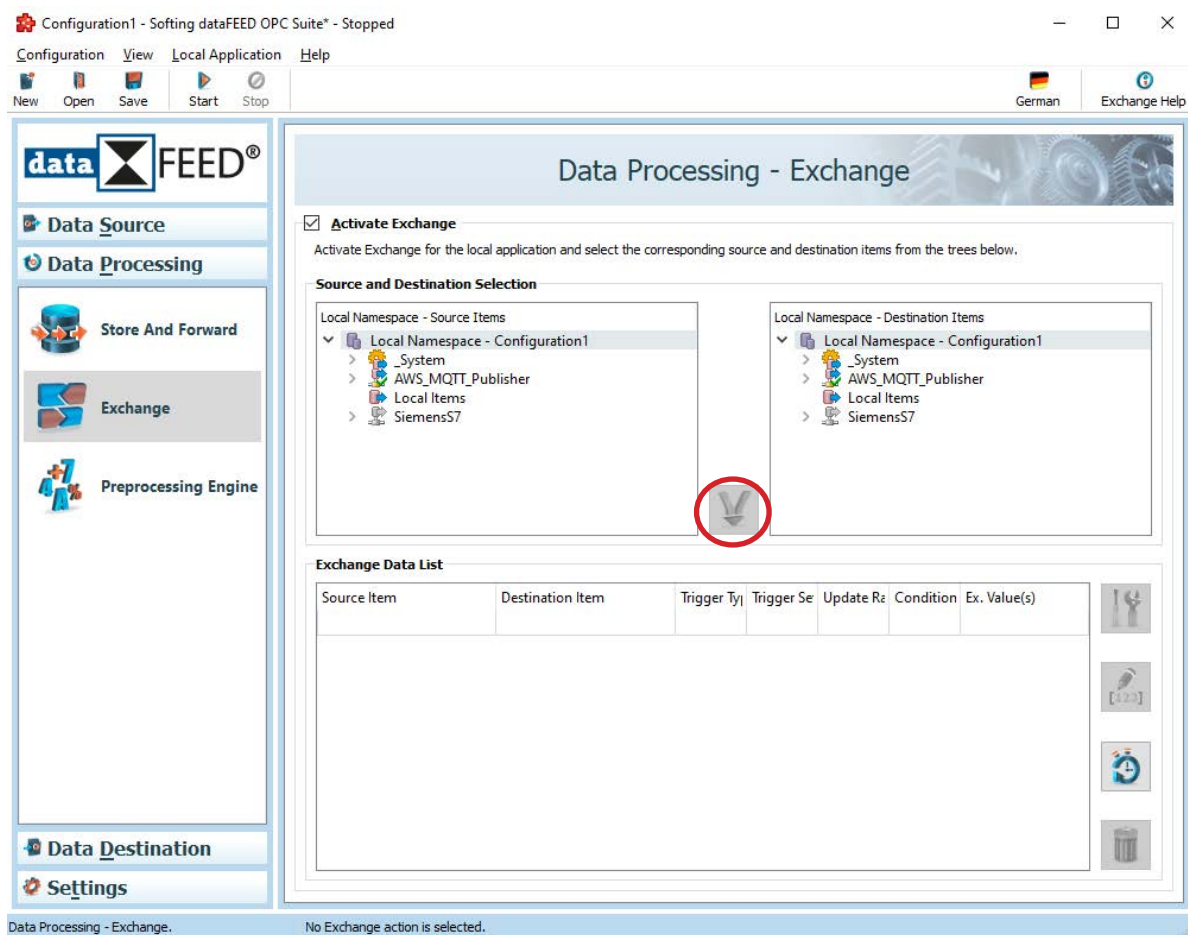
- Press *OK* button




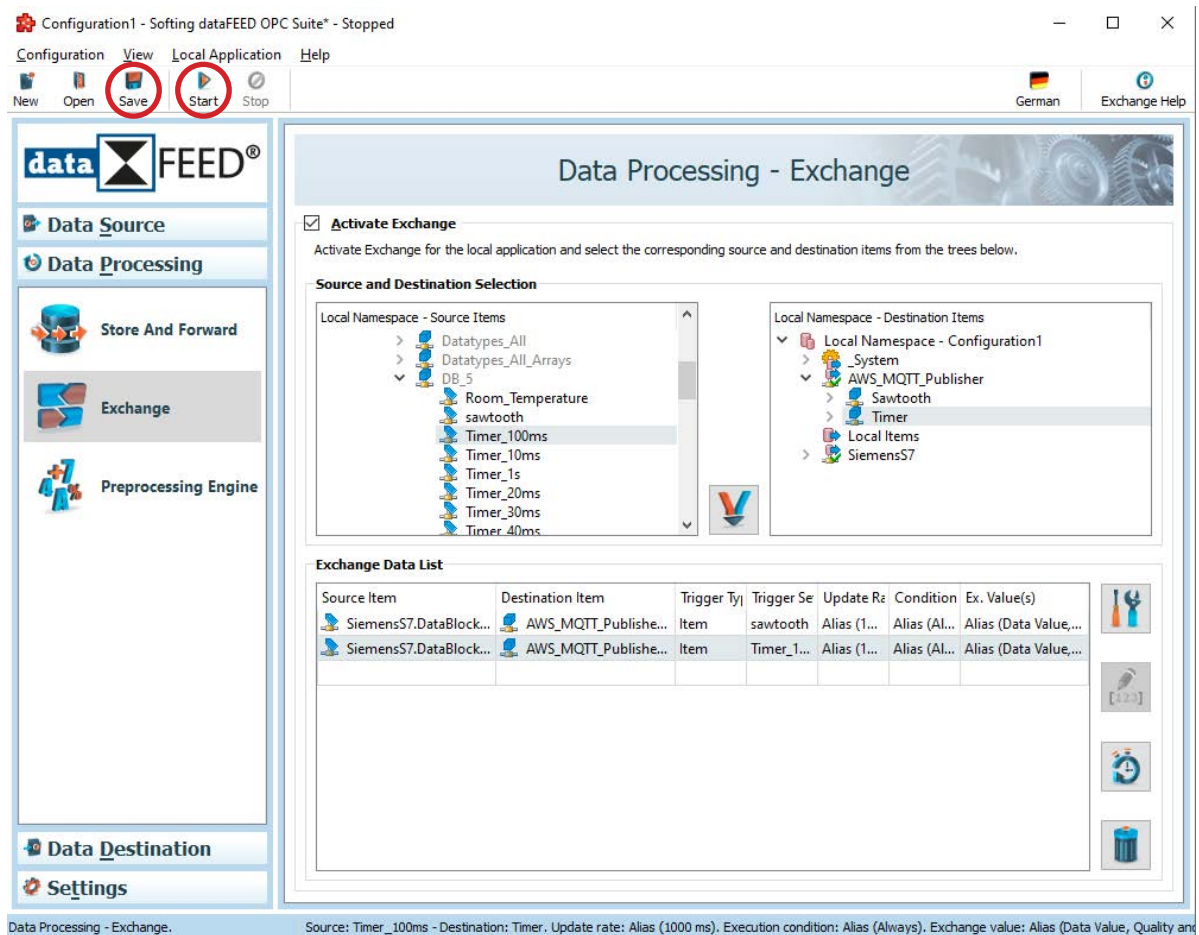
- Once all items are configured press *Next >* button to proceed to the next MQTT Broker settings or press *Finish* button to finish the MQTT Broker configuration.





- Navigate to *Data Processing/Exchange*



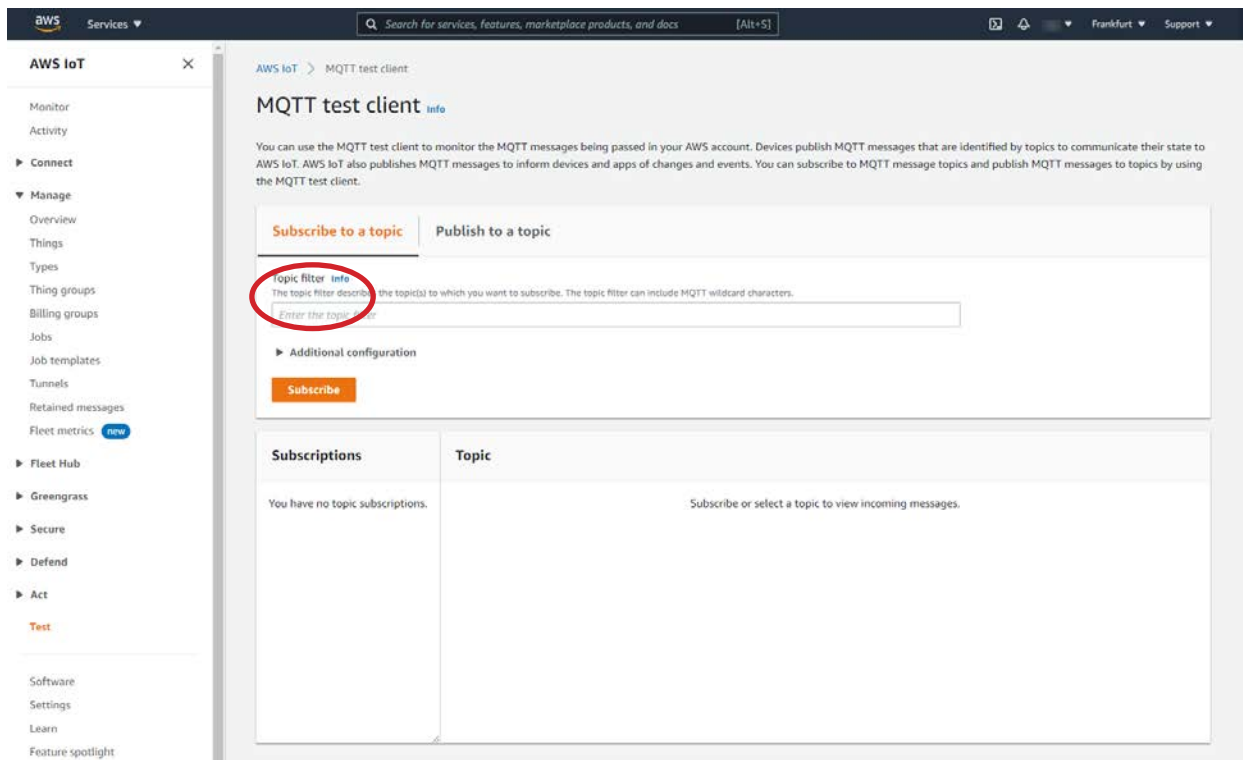
- Define mapping of source items (shopfloor data) to destination items (configured MQTT Broker items) by selecting the appropriate combinations in the *Local Namespace - Source Items* tree and the *Local Namespace - Destination Items* tree and afterwards by pressing the  (Use the selected items as a new Exchange action) button



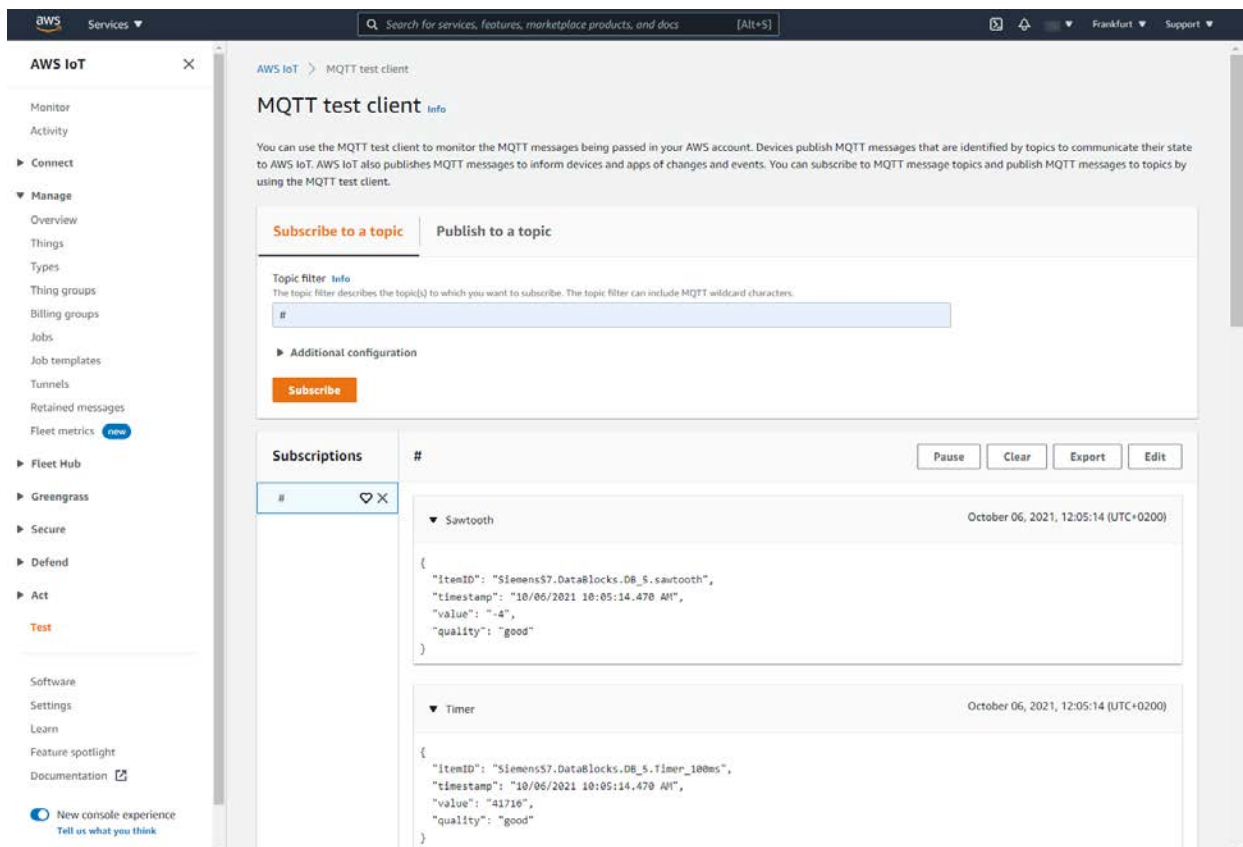
- Press  **Save** (*Save*) button to store **dataFEED OPC Suite** configuration
- Press  **Start** (*Start*) button to start **dataFEED OPC Suite**

4. Test MQTT Connection and Data Exchange

- In AWS IoT console navigate to *Test*



- Subscribe to all configured MQTT topics by entering wildcard # in *Topic filter* field
The received MQTT messages are shown using the defined publish format.



NOTE:

The given URLs have last been checked on Jan 05, 2022.

