How to Configure Softing Gateways as AWS IoT Device
How to Configure Softing dataFEED Gateways as AWS IoT Device

Preliminary Remarks

This Configuration Manual describes how to configure the Softing gateway *uaGate SI, uaGate MB* or *edgeGate* as AWS IoT device ("Thing"). The specific gateway used is referred to in this manual as *gateway*. It is based on the AWS Developer Guide [https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html](https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html).

Additional information about the Softing products can be found at the following web pages:


AWS Preparation Steps

1. **Register Gateway as Device in the Registry**

Follow the instructions given at [https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html](https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html).

   ![Create a Thing](image)

   In step 4. define unique *gateway* name, e.g. by using its serial number.

2. **Create and Activate Certificate for Gateway**

Follow the instructions given at [https://docs.aws.amazon.com/iot/latest/developerguide/create-device-certificate.html](https://docs.aws.amazon.com/iot/latest/developerguide/create-device-certificate.html).

In step 2. download certificate and key files. These files need to be uploaded as MQTT Client certificate into the *gateway* in a later step.
3. **Create AWS IoT Policy for MQTT Client Certificate**

Follow the instructions given at https://docs.aws.amazon.com/iot/latest/developerguide/create-iot-policy.html.

Define **gateway** permission for **iot:Connect** and **iot:Publish** actions.

**NOTE:**
For some unknown reasons the attachment of policies as described in step 3. does not work. Instead, it looks like the policy must be created before attaching it to a certificate. However, the certificate dialog does not trigger the creation of a policy.

It is possible to create and attach a policy **after** the creation of a certificate by following these steps:

- Return to **AWS IoT Console** main screen and navigate to **Secure/Policies**

- Press **Create a policy** button
- Define policy name
- Define **gateway** permission for **iot:Connect** action
• Enter user-defined client ID in Resource ARN field
  It is recommended to use the gateway name here. Remember this client ID for a later step.

• Press Add Statement button
• Define gateway permission for iot:Publish action
• Define user-specific ARN topic. Remember this topic for a later step.

  NOTE:
  The placeholder “*” allows publishing to any topic.
• Press *Create* button

4. **Attach AWS IoT Policy to Gateway**
   • Return to *AWS IoT Console* main screen and navigate to *Manage/Things* page
   • Click on device name of *gateway*
• Click on **Security**
• Click on hash value of created device certificate

Select **Attach policy** in **Actions** menu

Attach created policy to certificate

Attach policies to certificate(s)

Policies will be attached to the following certificate(s):
2c1696c35411d66fae3efb6086f929e15363f37730888f07d0b6d47cd45e325e

Choose one or more policies

1 policy selected
5. **Determine AWS MQTT Broker Address**
   - Return to *AWS IoT Console* main screen and navigate to *Settings*
   - MQTT Broker address can be found in *Custom Endpoint* field
   
   **NOTE:**
   The address looks similar to “123456789-ats.iot.eu-central-1.amazonaws.com”

### Gateway Configuration Steps

6. **Configure MQTT Broker**
   - In *Internet Browser* connect to configuration webpage of *gateway*
   - Login as *Administrator* or *ITAdmin*
   - Navigate to *MQTT Broker Configuration* page

   ![Gateway Configuration Steps Image]

   - Set **URI kind** to `ssl://`
   - Set **Hostname** field by *AWS* MQTT Broker address
     (see section 5, *Determine AWS MQTT Broker Address*)
   - Set **Port** field to “8883”
   - Set **Client ID** field to defined gateway name
   - Enable MQTT **Clean Session** flag
   
   **NOTE:**
   AWS Broker disconnects, if flag is not set
   (see [https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html](https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html))
   - Set **Authentication** to “anonymous”
   - Set **Enable MQTT** checkbox
   - Press **Save** button
7. **Define MQTT Client Certificate**
   - Navigate to *MQTT Client Certificate* page
   - Define MQTT Client certificate as created in section 2, *Create and Activate Certificate for Gateway*
   - Set *Use MQTT Client Certificate* checkbox
   - Press *Save* button
   - Press *Apply Pending Changes* button

8. **Verify Connection**
   - Navigate to *Gateway Status* page for verifying that connection has been established
9. **Select MQTT Topics**

- Navigate to *MQTT Topic Selection* page

- Select PLC items to be published
10. Define MQTT Topic Settings

- Navigate to **MQTT Topic Settings** page

- If user-specific ARN topic has been defined (see section 3, *Create AWS IoT Policy for MQTT Client Certificate*):
  - Do **not** define **MQTT Root Topic (Topic Prefix)**
  - Set **Hierarchy** to **Suppressed PLC symbols**
  - Enter defined unique gateway name from section 1, *Register Gateway as Device in the Registry, MQTT Suffix Topic** field

- If placeholder "*" has been defined as ARN topic (see section 3, *Create AWS IoT Policy for MQTT Client Certificate*):
  - Enter user-defined root topic in **MQTT Root Topic (Topic Prefix)** field
  - Select **Hierarchy** radio button as needed
  - Enter user-defined suffix topic in **MQTT Suffix Topic** field

- Do not activate **Enable MQTT Retain** checkbox

**NOTE:**
AWS Broker closes connection, if retain flag is set for MQTT Publish message (see https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html)

- Set **Collect PLC values in one MQTT message** checkbox for reducing the amount of MQTT messages sent from the gateway to the AWS cloud

- Press **Save** button

- Press **Apply Pending Changes** button
dataFEED OPC Suite Configuration Steps

11. Define MQTT Broker Connection

- Start *dataFEED OPC Suite Configurator*
- Navigate to *Data Destination/MQTT Broker* page
• Navigate to **Connection Settings** page (page 1) of **MQTT Broker Connection Wizard**

![MQTT Broker Connection Wizard](image)

**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 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-url]

**Client ID**

Client ID: [beedPubSub]

**Connection State**

Specify here if the connection shall be active.

Connection Active: [ ]

• Define MQTT Broker **Connection Name** and unique **Client ID**

**NOTE:**

Device ID has to be allowed in iot:Connect policy

(see section **AWS Preparation Steps**)

---

**Configuration Manual**

---

**data FEED®**
- Navigate to **Communication Settings** page (page 2) of **MQTT Broker Connection Wizard**

  - Set **MQTT Broker URI** to `ssl://
  - Enter **AWS MQTT broker address** as prefix in `< IP address or hostname with domain of the broker > [ : <port number> ]` field
  - Enter port number “8883” as postfix in `< IP address or hostname with domain of the broker > [ : <port number> ]` field
  - Enable MQTT **Clean Session** flag

**NOTE:**
AWS Broker disconnects, if flag is not set (see [https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html](https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html))

- Set **Authentication Settings/User Identity** to **Anonymous**
- Activate **Use Client Certificate** checkbox
- Select the downloaded certificate “*-certificate.pem” and private key “*-private.pem.key”

**NOTE:**
Before the private key can be selected the certificate file extension has to be changed to “.pem”
MQTT Topic definition

On this wizard page the address space for the current connection can be defined.

Connection: MQTT-Broker-aws

MQTT Topics

Add the topic you want to insert into the namespace of the local application. The topics can be defined manually or imported from an external file. Please define the item mapping into MQTT topics within the configuration of the Exchange function. The configuration of the Exchange function can be found within Data Processing.

MQTT-Broker-aws

[Tree view of MQTT topics]

☐ Time Zone Format

Specify the time zone for the timestamp expression used within requests.

Time zone: UTC
• Navigate to *MQTT Topic definition* page (page 3) of *MQTT Broker Connection Wizard*

![MQTT Topic Properties](image)

• Add topics
  
  **NOTE:**
  Topics have to be allowed in iot:Publish policy
  (see section *AWS Preparation Steps*)

• Deactivate *Retain* flag

• Press *Save* button
• Navigate to **Data Processing/Exchange** page

![Data Processing/Exchange page screenshot](image)

• Add exchange task for copying data source values to MQTT topic

12. **Finish dataFEED OPC Suite Configuration**

• **dataFEED OPC Suite** configuration is finished
• Save configuration and start **dataFEED OPC Suite**
13. Monitor Values in AWS

- Return to **AWS IoT Console** main screen and navigate to **Test**
- Subscribe to all topics using the wildcard “#”
- Received MQTT messages look different, depending on topic format settings and options for the collected PLC values

MQTT client for single value

MQTT client for multi value