

D70 - Getting Started Guide for AWS IoT Core

- 1 - Document information
- 2 - Overview
- 3 - Hardware Description
- 4 - Set up your Development Environment
- 5 - Set up your hardware
- 6 - Setup your AWS account and Permissions
- 7 - Connect your gateway
- 8 - Provision your gateway
- 9 - Exploit your Socomec data
 - a. Create the AWS Timestream database and table
 - b. Forward your data from IoT Core to the Timestream database
 - c. Check your configuration
 - d. Install Grafana (hosted)
 - Grafana SQL commands samples
 - Plot temperature for all BLE temperature sensors found
 - Plot distribution of received messages count per device
 - Plot BLE magnetic sensor counter value:
- 10 - Troubleshooting
- Annex
 - 1 - Dashboard demo
 - 2 - Socomec data types description
 - 3 - Alternate provisioning methods
 - a) Gateway generates certificate
 - b) Gateway generates CSR, signed by AWS
 - c) Gateway generates CSR, custom CA signs the certificate

Glossary

- *Thing*: Device in AWS IoT Core
- AWS: Amazon Web Services
- DQP: Device Qualification Program (AWS program)
- Webview: Embedded web server for configuration and monitoring

1 - Document information

This document provides instructions for connecting the SOCOMEK DIRIS Digiware M/D gateway to AWS IoT Core :

- Register and configure your *Thing* in AWS IoT Core
- Provision your device
- Configure AWS IoT Core connectivity on the device

Prerequisites:

- IP address
- DNS
- Outgoing port 8883 open
- SNTP Configured

2 - Overview

The DIRIS Digiware D-50 (and D-70) display is a master on the Digiware bus and acts as a gateway interface to communicate measurements over RS485 and Ethernet.

DIRIS Digiware M-50 and M-70 act as the Digiware system interface and communication gateway centralizing measurements from DIRIS Digiware modules and communicating data over Ethernet.

3 - Hardware Description

Product user manual and datasheet can be found here:

- For D50/D70: <https://www.socomec.us/en-us/reference/48290203>
- For M50/M70: <https://www.socomec.us/en-us/reference/48290204>

4 - Set up your Development Environment

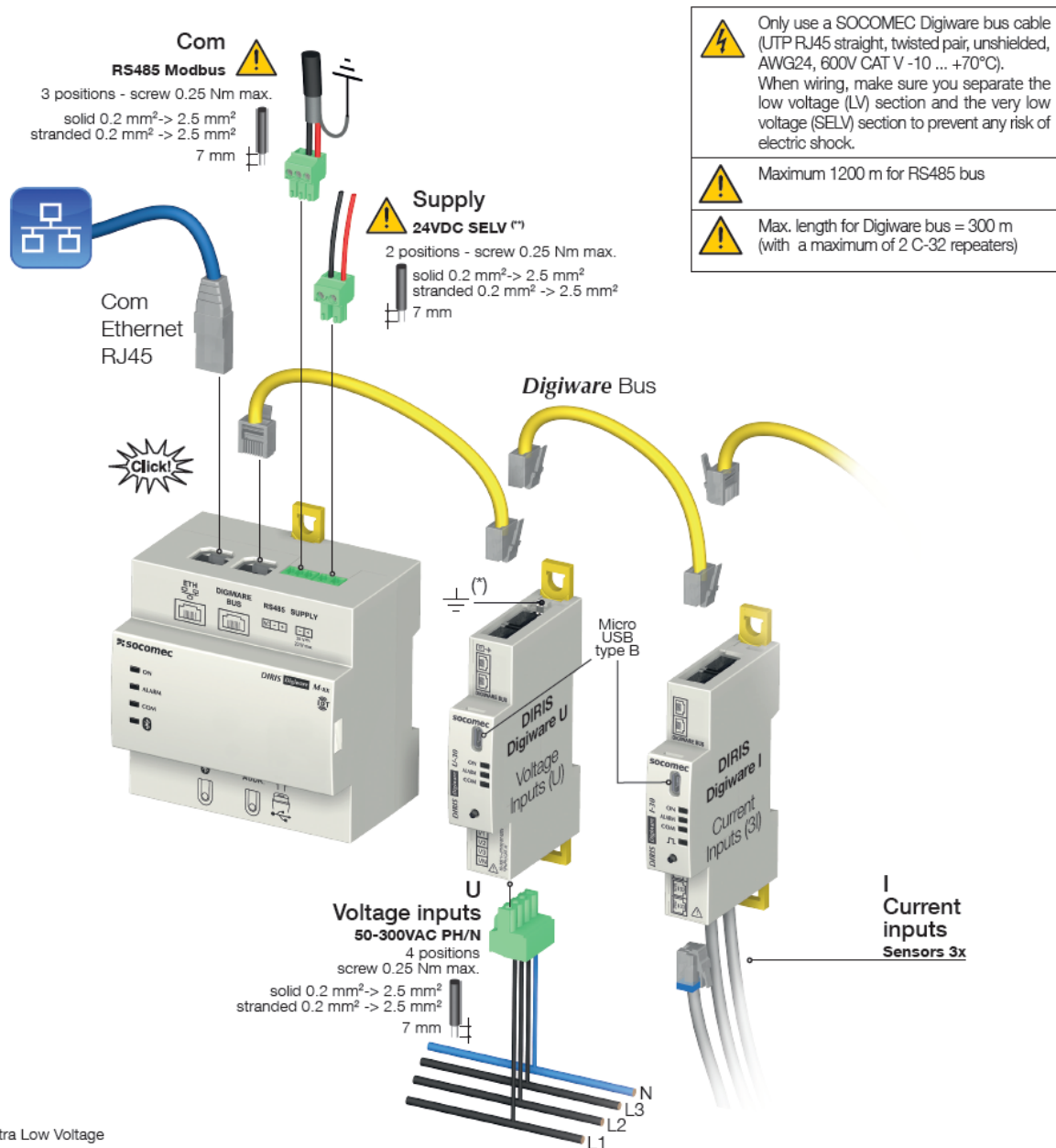
i The DIRIS Digiware M/D gateways come with firmware natively compatible with AWS connectivity. No need to compile source code or libraries, as configuration is easily done through the integrated web server.

5 - Set up your hardware

Prior connecting your gateway to AWS, you will need to install the gateway in your installation. Following resources will help you in this task:

- User manual of your device, which can be found at [socomec.com](https://www.socomec.com)

i Before enabling the AWS platform in your gateway, make sure all devices are properly connected and configured. You can verify this using the diagnostic page of the integrated web server.



6 - Setup your AWS account and Permissions

Refer to the instructions at [Set up your AWS Account](#). Follow the steps outlined in these sections to create your account and a user and get started:

- Sign up for an AWS account
- Create an user and grant permissions.
- Open the AWS IoT console

7 - Connect your gateway

In order to connect your gateway to AWS, you will have to configure the AWS connection using the embedded webserver "Webview".

Please refer to the product datasheet in order to configure your "Cybersecurity" Webview account.

There are 3 ways to provision certificates in your gateway.

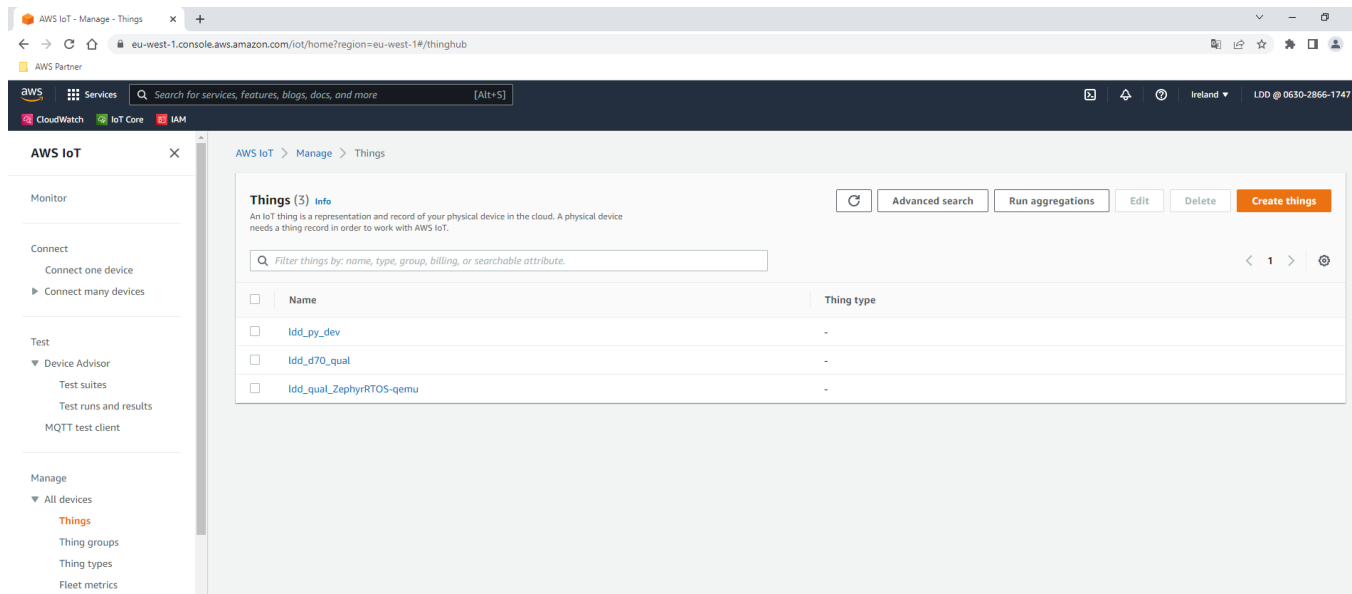
- Allow AWS to generate a private key and a public certificate, then upload these credentials to the gateway (recommended and described below).
- Allow the gateway to generate a private key and a public certificate, then download the public certificate and upload it to AWS when creating your *Thing*.
- Download a generated CSR from the gateway, sign it using AWS CA, and upload the public certificate back to the device. This method has the advantage of keeping the private key protected by not exposing it.

The CSR method also allows you to use a custom CA instead of the default AWS CA. This custom CA should be registered in AWS IoT Core (refer to [this guide](#)). To provision your gateway, download the CSR generated by the gateway, sign the certificate, and upload this certificate back to the gateway. This method offers two advantages:

- The private key remains protected in the gateway's persistent storage and is never exposed
- Using a CA with AWS enables [Just In Time Provisioning \(JITP\)](#), which eliminates the need to manually create your Thing.

Follow the steps outlined in these sections to provision resources for your device, or you can also refer to the official documentation at [Create AWS IoT Resources](#).

1. Select *Thing* tab
2. Click "Create Things"



3. Select "Create single Thing"

AWS IoT - Manage - Things - Create

eu-west-1.console.aws.amazon.com/iot/home?region=eu-west-1#/create/provisioning

AWS Partner

aws Services Search for services, features, blogs, docs, and more [Alt+S]

CloudWatch IoT Core IAM

AWS IoT

Monitor

Connect

Connect one device

Connect many devices

Test

Device Advisor

Test suites

Test runs and results

MQTT test client

Manage

All devices

Things

AWS IoT > Manage > Things > Create things

Create things [Info](#)

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Number of things to create

☒ **Create single thing**
Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.

☐ **Create many things**
Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

Cancel **Next**

4. Choose the *Thing* name, this name cannot be change and will identify the *Thing*.

AWS IoT - Manage - Things - Create things - Create single thing

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Specify thing properties [Info](#)

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties [Info](#)

Thing name

ldd_qual_demo

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

▶ **Thing type** - optional

▶ **Searchable thing attributes** - optional

▶ **Thing groups** - optional

▶ **Billing group** - optional

Device Shadow [Info](#)

Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

☒ **No shadow**

☐ **Named shadow**
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.

☐ **Unnamed shadow (classic)**
A thing can have only one unnamed shadow.

5. Select "Auto-generate a new certificate (recommended)". Other possibilities to select a certificate are discussed below.

This step differ depending on the method you choose to provision your gateway

AWS IoT

Monitor

Connect

Connect one device

Connect many devices

Test

Device Advisor

Test suites

Test runs and results

MQTT test client

Manage

All devices

Things

Thing groups

Thing types

Fleet metrics

AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Configure device certificate - optional [Info](#)

A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

☒ **Auto-generate a new certificate (recommended)**
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

☐ **Use my certificate**
Use a certificate signed by your own certificate authority.

☐ **Upload CSR**
Register your CA and use your own certificates on one or many devices.

☐ **Skip creating a certificate at this time**
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Cancel Previous **Next**

6. Click on "Create policy" and choose a name for the policy

Create a tailored, minimal policy that perfectly meets the Thing's requirements (connect and publish/subscribe to specific topics).

The policy should match the topics the gateway operates on. An incorrect policy may prevent the gateway from connecting to the broker or from publishing/ subscribing.

It is recommended to use the following policy document:

- In "policy document" section click "JSON"
- Paste the following policy document (JSON format)
- Change the region "eu-west-1" with your current AWS region
- Change the account ID "000000000000" with your current account ID

Gateway policy document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:eu-west-1:000000000000:client/${iot:Connection.Thing.ThingName}"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive"
      ],
      "Resource": "arn:aws:iot:eu-west-1:000000000000:topic/${iot:Connection.Thing.ThingName}/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:eu-west-1:000000000000:topicfilter/${iot:Connection.Thing.ThingName}/*"
    }
  ]
}
```

This minimal policy enables the *Thing* to:

- Connect to the broker
- Publish to topics matching the pattern "\${ThingName}/*"

- Subscribe to topics matching the pattern "{ThingName}/*"
- Receive published messages on subscribed topics matching the pattern "{ThingName}/*"



NOTE

The examples in this document are intended only for development environments. All devices in your production fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to Example policies (<https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html>) and Security Best practices (<https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html>).

Policy properties

AWS IoT Core supports named policies so that many identities can reference the same policy document.

Policy name
MyGatewayPolicy

A policy name is an alphanumeric string that can also contain period (.), comma (,), hyphen(-), underscore (_), plus sign (+), equal sign (=), and at sign (@) characters, but no spaces.

Tags - optional

Policy statements **Policy examples**

Policy document info

An AWS IoT policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources.

Builder JSON

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "iot:Connect",
7       "Resource": "arn:aws:iot:eu-west-1:123456789012:client/${iot:Connection.Thing.ThingName}"
8     },
9     {
10      "Effect": "Allow",
11      "Action": "iot:Publish",
12      "Resource": "arn:aws:iot:eu-west-1:123456789012:topic/${iot:Connection.Thing.ThingName}/*"
13    },
14    {
15      "Effect": "Allow",
16      "Action": "iot:Subscribe",
17      "Resource": "arn:aws:iot:eu-west-1:123456789012:topicfilter/${iot:Connection.Thing.ThingName}/*"
18    },
19    {
20      "Effect": "Allow",
21      "Action": "iot:Receive",
22      "Resource": "arn:aws:iot:us-west-1:123456789012:topic/${iot:Connection.Thing.ThingName}/*"
23    }
24  ]
25 }
```

JSON Line 22, Column 54 Errors: 0 Warnings: 0

7. Select the newly created policy for your certificate and click "Create thing".

AWS IoT

Monitor

Connect

- Connect one device
- Connect many devices

Test

- Device Advisor
- MQTT test client

Manage

- All devices
- Greengrass devices
- LPWAN devices
- Remote actions
- Message Routing
- Retained messages
- Security
- Fleet Hub

Device Software

- Billing groups
- Settings
- Learn
- Feature spotlight
- Documentation

☒ New console experience

[Tell us what you think](#)

Policy properties

AWS IoT Core supports named policies so that many identities can reference the same policy document.

Policy name

ldd_qual_demo_minimal

A policy name is an alphanumeric string that can also contain period (.), comma (,), hyphen(-), underscore (_), plus sign (+), equal sign (=), and at sign (@) characters, but no spaces.

Tags - optional

Policy statements | Policy examples

Policy document [info](#)

An AWS IoT policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources.

Builder JSON

Policy document

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "allow",
6       "Action": "iot:Connect",
7       "Resource": "arn:aws:iot:eu-west-1:123456789012:client/${iot:Connection.Thing.ThingName}"
8     },
9     {
10      "Effect": "allow",
11      "Action": "iot:Publish",
12      "Resource": "arn:aws:iot:eu-west-1:123456789012:topic/${iot:Connection.Thing.ThingName}/*"
13    },
14    {
15      "Effect": "allow",
16      "Action": "iot:Subscribe",
17      "Resource": "arn:aws:iot:eu-west-1:123456789012:topicfilter/${iot:Connection.Thing.ThingName}/*"
18    },
19    {
20      "Effect": "allow",
21      "Action": "iot:Receive",
22      "Resource": "arn:aws:iot:us-west-1:123456789012:topic/${iot:Connection.Thing.ThingName}/*"
23    }
24  ]
25 }
```

JSON Line 22, Column 54 Errors: 0 Warnings: 0

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

8. Download your *Thing* private key and certificate.

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

be3cb2f23d0...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

be3cb2f23d0221808adb8bf...023300b-public.pem.key

[Download](#)

Private key file

be3cb2f23d0221808adb8bf...23300b-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](#)

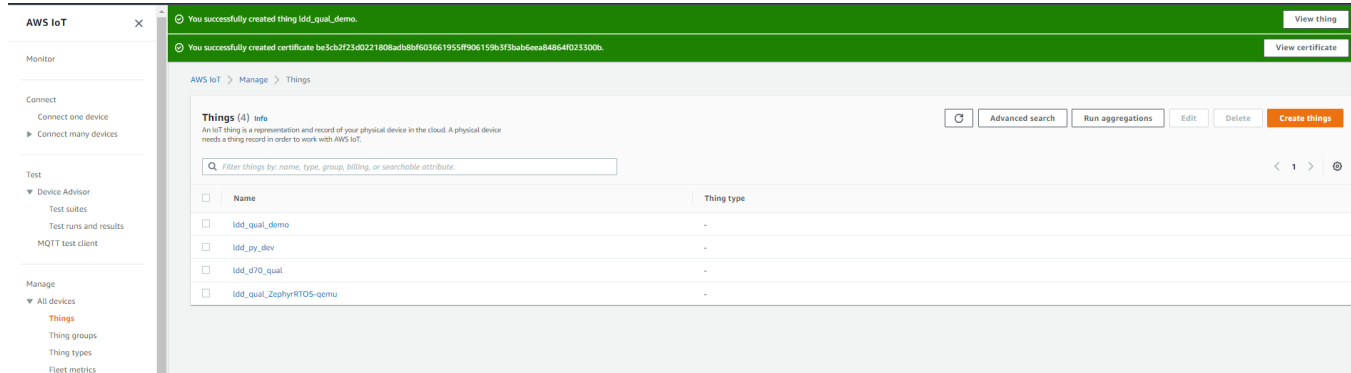


This page is your only chance to download the private key file for your certificate.

 You don't need to manually upload the AWS CA certificate as it is already embedded.

9. Click "done".

At this point you have created the *Thing* and attached to it a certificate. You will find your *Thing* in the list :



8 - Provision your gateway

Once you have a Webview account with necessary privilege and you created the *Thing* and you downloaded the device public certificate and private key. You will need to provision the gateway with these credentials.

To do so, you will need to:

- Connect to Webview application with either Admin or Cyber account.
- Navigate to "Protocol" section, "Cloud" tab
- Enable AWS IoT Core cloud platform
- Fill following information in the form:
 - AWS Endpoint (Endpoint can be found in AWS IoT Core > Settings page)
 - *Thing* name
 - Select "Upload method" in the list
 - Upload the x509 certificate generated by AWS
 - Upload the private key generated by AWS
- Validate your changes

Protocols

Network Configuration Data Push Cloud Field Protocols Email

2023/05/04 13:34

AWS Cloud Connection

Enable Cloud ☒

Certificate management

Endpoint

Thing name

Certificate management

Certificate [Browse](#)

Private Key [Browse](#)

Certificate Signing Request (CSR)

Thing name

[Generate CSR](#)

Synchronisation Status

- Click synchronize

AWS Cloud Connection

Enable Cloud ☒

Certificate management

Endpoint a2urlgdsiqqpee-ats.iot.eu-west-1.amazonaws.com

Thing name ldd_d70_qual

Installed Certificate Information [View Certificate Details](#)

Certificate Signing Request (CSR)

Thing name ldd_d70_qual

[Generate CSR](#)

Synchronisation Status

State **Active**

Last connection 2023/05/04 11:31:01

Synchronised Devices 5/5

[Synchronise](#)

After few seconds, the gateway should connects to AWS and synchronize devices. You should be able to see uplinked MQTT messages in the AWS MQTT Test Client.

```
{
  "name": "Atys P",
  "productId": 2300,
  "netId": "29E021",
  "serialNumber": "99999999999",
  "uuid": "bb9b5f10-4215-11cc-9ac5-393939393939"
}
```

 Additional provisioning methods involving x509 certificates are described in the annex.

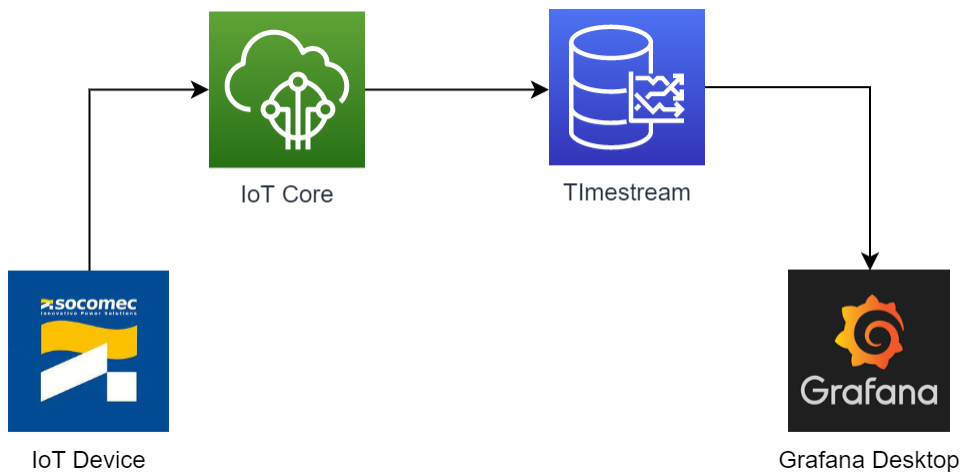
You should be able to view the JSON payloads pushed by the gateway in the *MQTT* test client when subscribing to the topic starting with your *Thing Name*. Congratulations, you have successfully connected your gateway to AWS IoT Core, and you can now process pushed data as needed.

 The Socomec's gateway send the message in JSON format. The complete description is available in the annex bellow.

9 - Exploit your Socomec data

Once IoT Core receives the data published by the gateway you might want to exploit them.

To do this, we will present a simple and straightforward architecture for storing the data in a timeseries database (Timestream) and displaying it in Grafana:

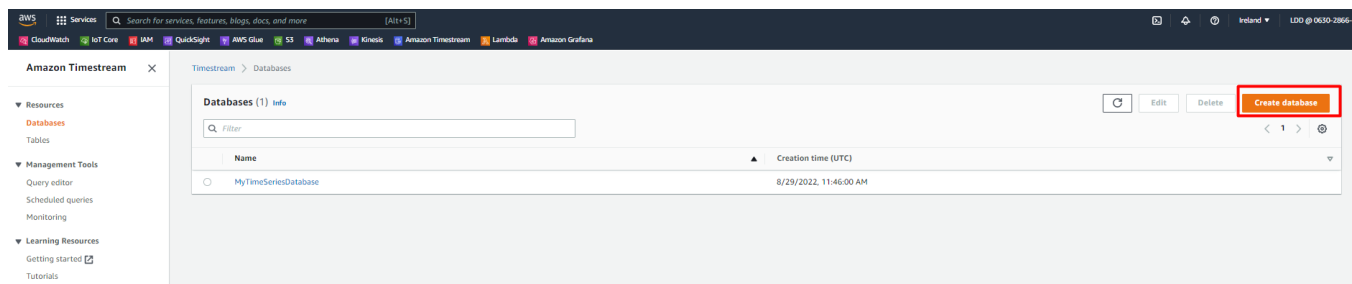


a. Create the AWS Timestream database and table

General documentation about Timestream is available here: [What is Amazon Timestream?](#)

To create a Timestream database:

- Select your region
- Go to the Timestream service console
- Click "Create database"



- Select "Standard database"
- Choose a name for your database

Timestream > Databases > Create database

Create database [Info](#)

Database configuration
Create and configure a database or create a database with sample data to explore Timestream right away.

Choose a configuration

☒ **Standard database**
Create a new database with custom configuration.

☐ **Sample database**
Create a database and populate it with sample data to get started in a single click.

Name
Specify a name that is unique for all Timestream databases in your AWS account in the current Region. You can not change this name once you create it.

MySocomecGatewayDatabase

Must be between 3 and 256 characters long. Must contain letters, digits, dashes, periods or underscores.

- Keep the rest of the configuration unchanged
- Click "Create database"

Your created database should appear in the list:

Amazon Timestream x Successfully created database MySocomecGatewayDatabase.

Timestream > Databases

Databases (2) [Info](#)

Q Filter

Name	Creation time (UTC)
☐ MySocomecGatewayDatabase	9/1/2022, 12:50:46 PM
☐ MyTimeSeriesDatabase	8/29/2022, 11:46:00 AM

Buttons: Refresh, Edit, Delete, Create database

Navigation: < 1 > @

Now we will create a table in this database:

- Go to "Tables" panel
- Click "Create Table"
- Choose from the databases list the one previously created
- Choose a name for your table
 - 1 Day of "Memory store retention"
 - 1 Week of "Magnetic store retention"

Timestream > Tables > Create table

Create table [Info](#)

Table details

Database name
Choose the database where this table will be created.
MySocomecGatewayDatabase

Table name
Specify a table name that is unique within this database. You can not change this name once you create it.
MyTable1

Must be between 3 and 256 characters long. Must contain letters, digits, dashes, periods or underscores.

Data retention [Info](#)
Specify how long your data is retained in each storage tier. Data moves from the memory store to the magnetic store as it ages. Data that exceeds the magnetic store retention will be deleted.

Memory store retention
Specify how long data will be stored in the memory store before it is moved to magnetic store.
1 Day(s)

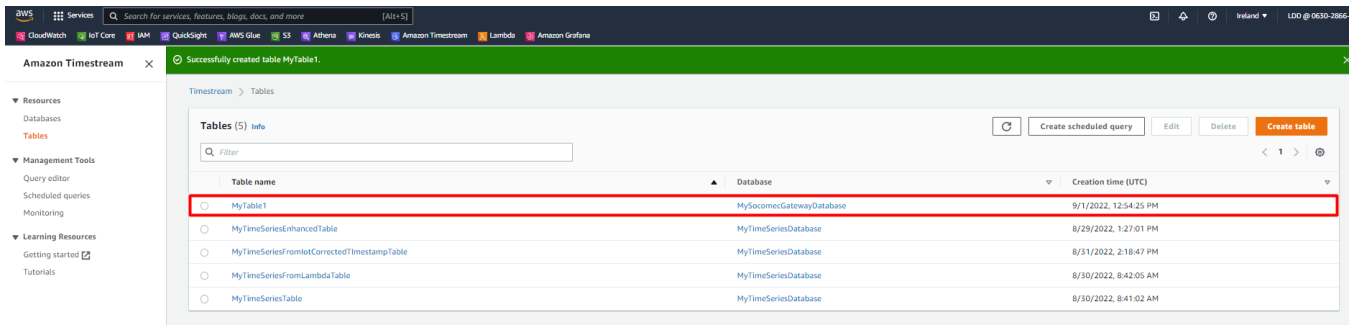
The value must be a number. Minimum: 1 hour, maximum: 12 months.

Magnetic store retention
Specify how long data will be stored in the magnetic store before it is deleted.
1 Week(s)

The value must be a number. Minimum: 1 day, maximum: 200 years.

- Keep the rest of the configuration unchanged
- Finally click "Create table"

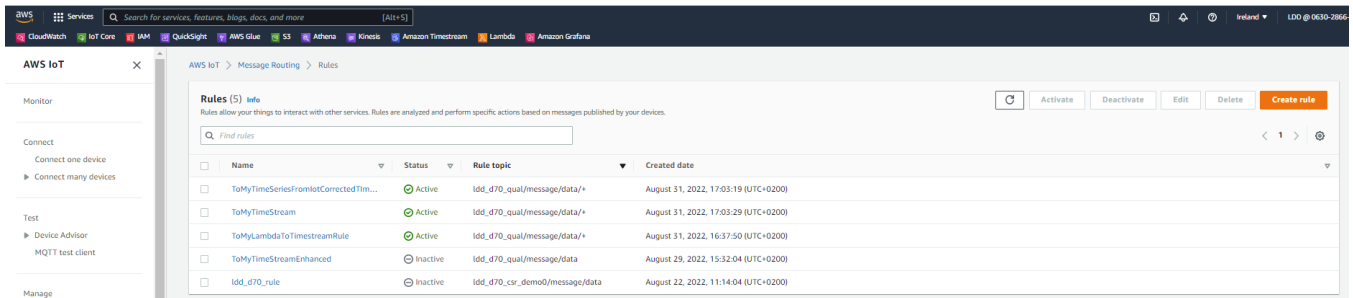
Your table should appear in the list of tables



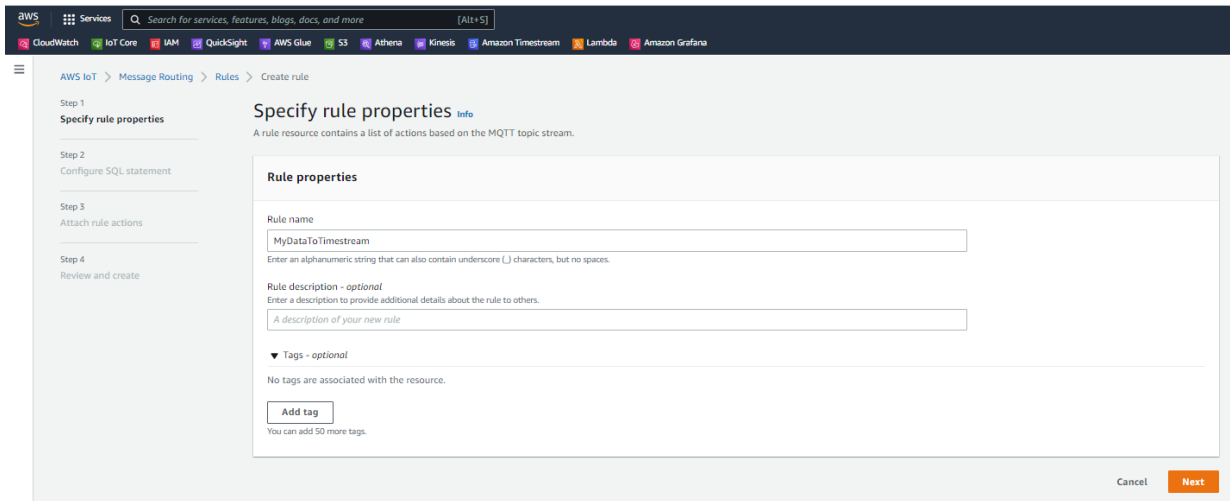
b. Forward your data from IoT Core to the Timestream database

Now we will create a IoT Core rule to forward data collected by IoT Core to the Timestream database, documented here: [Working with other services : AWS IoT Core](#)

- Go to IoT Core
- Go to "Message Routing > Rules" panel
- Click "Create rule"



- Choose a name for your rule
- Click "Next"



- Select SQL version "2015-10-08"



The reason why using an older version of SQL, is because with SQL version "2016-03-23", fractional part of float numbers is wiped out if equals 0. Whereas with the older version, it is kept. This fractional part is required for inserting the measure into the proper (double/bigint) column in the table.

- Copy the following SQL statement

```
SELECT measures.* FROM 'ldd_d70_qual/message/data/*'
```

- Click Next

Step 1
Specify rule properties

Step 2
Configure SQL statement

Step 3
Attach rule actions

Step 4
Review and create

Configure SQL statement [Info](#)

Add a simplified SQL syntax to filter messages received on an MQTT topic and push the data elsewhere.

SQL statement

SQL version
The version of the SQL rules engine to use when evaluating the rule.

2016-03-23

SQL statement
Enter an SQL statement using the following: SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT deviceid, measures.* FROM 'idd_d78_qual/message/data/*'
```

SQL Line 1, Column 1

Cancel Previous **Next**

- Select "Timestream table" as Action 1
- Select your Timestream database from the list
- Select your Timestream table from the list
- Add a first dimension, with following configuration
 - Dimension name: "device_id"
 - Dimension value: "\${topic(4)}"
- Configure the timestamp source
 - Timestamp value: "\${utctimestamp}"
 - Select "SECONDS" as timestamp unit
- Configure the IAM role:
 - Click "Create new role"
 - Choose a name for your role
 - Click "Create"

Successfully created role MyDataToTimeStreamRole

AWS IoT > Message Routing > Rules > Create rule

Step 1: Specify rule properties

Step 2: Configure SQL statement

Step 3: Attach rule actions

Step 4: Review and create

Attach rule actions

An action routes data to a specific AWS service.

SQL statement

SELECT deviceId, measures.* FROM '1dd_g70_qual/message/data/*'

Rule actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1

TimeStream table

Write a message into a TimeStream table

Database name: MySoconecGatewayDatabase

Table name: MyTable1

Dimensions

Each record contains an array of dimensions (minimum 1). Dimensions represent the metadata attributes of a time series data point.

Dimensions name: device_id

Dimension value: \${topic(4)}

Timestamp value - optional: \${utctime}

Timestamp unit: SECONDS

IAM role: MyDataToTimeStreamRole

Choose a role to grant AWS IoT access to your endpoint.

AWS IoT will automatically create a policy with a prefix of "aws-iot-rule" under your IAM role selected.

Add rule action

- Keep the rest of the configuration unchanged and click "Next"
- Review your rule configuration
- Then click "Create"

AWS Grafana Console | PSM workspace dashboard - Am | New folder | AWS IoT | What is Amazon TimeStream? - | AWS IoT - Message Routing - Ru | Google

eu-west-1.console.aws.amazon.com/iot/home?region=eu-west-1#/create/rule

Admin | Misc | Azure | AWS | Agora | Workspace | Current | AWS Doc to finish 3...

Successfully created role MyDataToTimeStreamRole

Step 1: Rule properties

Rule properties

Name: MyDataToTimeStream

Description: -

Step 2: SQL statement

SQL statement

SQL version: 2018-03-23

SQL query: SELECT deviceId, measures.* FROM '1dd_g70_qual/message/data/*'

Step 3: Rule actions

Actions

TimeStream table

Write a message into a TimeStream table

Database name: MySoconecGatewayDatabase

Table name: MyTable1

Dimensions

- 1
 - name: device_id
 - value: \${topic(4)}

Timestamp value - optional

- value: \${utctime}
- unit: SECONDS

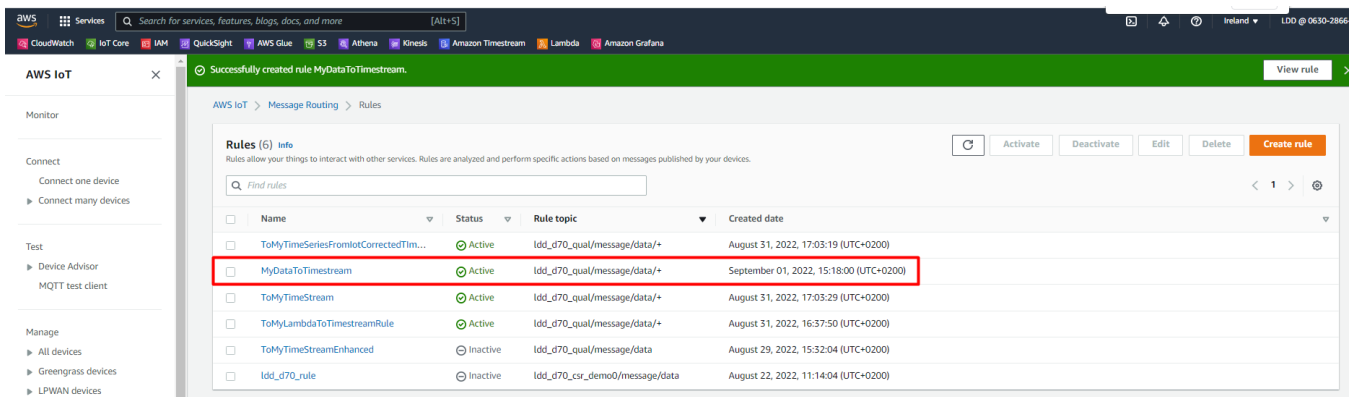
IAM role: arn:aws:iam::063028961747:role/service-role/MyDataToTimeStreamRole

Error action

No error action

Cancel | Previous | Create

Your rule should be listed in the following table:



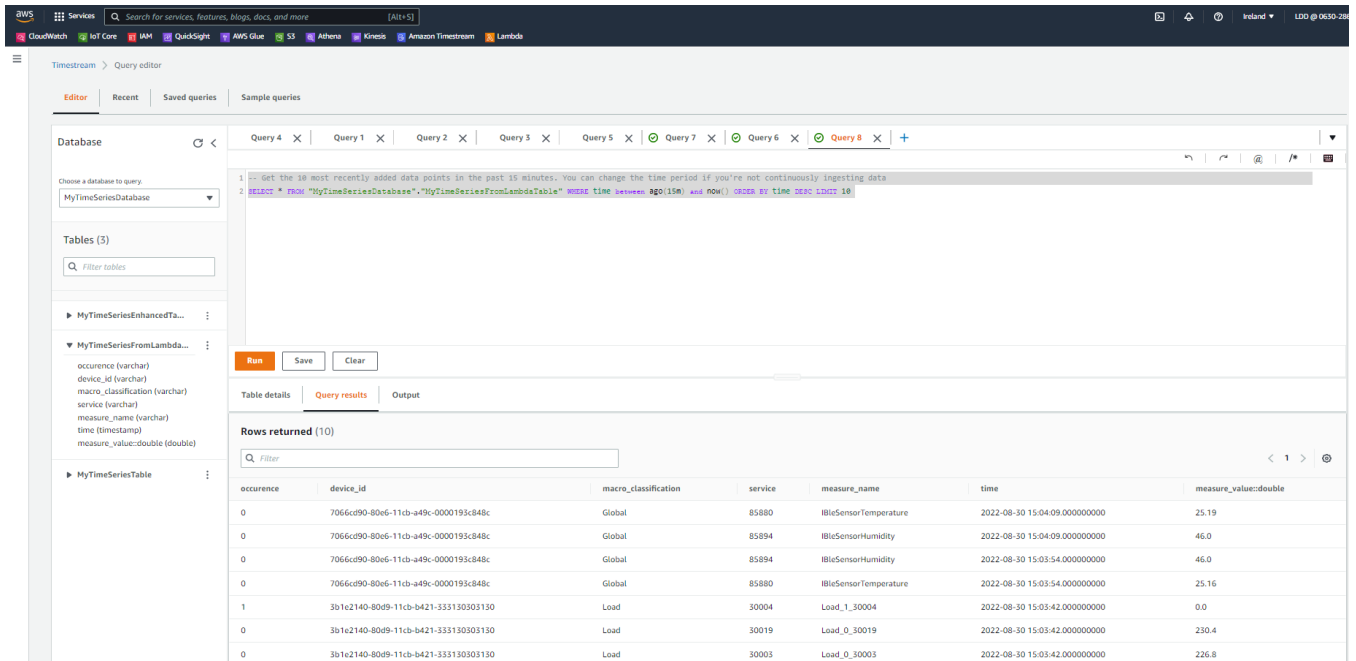
c. Check your configuration

Once you create the Timestream table and the IoT Core rule. If your gateway is connected to AWS, you should see your table being filled with the data.

Go to the Timestream Query editor

- Select your database
- Select your Table
- Click the three dots
- Click "Preview data"
- Run query

You should get something like this:



d. Install Grafana (hosted)

In the following steps, we will use AWS Amazon Grafana service, which is a simple and convenient way to create dashboards accessible worldwide. But you are free to use the desktop version of Grafana or another dashboard tool .

- Goto to AWS Grafana Console
- Click "Create Workspace"
- Choose a name
- Click "Next"

Amazon Managed Grafana

Amazon Grafana > Workspaces > Create new workspace

Step 1: Specify workspace details

Step 2: Configure settings

Step 3: Service managed permission settings

Step 4: Review and create

Service managed permission settings

IAM permission access settings

Select how you would like to specify account access.

☒ Current account
Use Grafana to monitor resources in your current account.

☐ Organization
Use Grafana to monitor resources in your Organizational Units (OUs).

Data sources and notification channels - optional

Data sources

Selecting an AWS data source below creates an IAM role that enables Amazon Grafana access to those resources in your current account. It does not set up the selected service as a data source. Note that some resources must be tagged `GrafanaDataSource` to be accessible.

☒	Data source name
☐	AWS IoT SiteWise
☐	AWS X-Ray
☐	Amazon CloudWatch
☐	Amazon OpenSearch Service
☐	Amazon Managed Service for Prometheus
☒	Amazon TimeStream
☐	Amazon Redshift
☐	Amazon Athena

Notification channels

Selecting a notification channel below creates an IAM role that enables Amazon Grafana access to those resources in your current account. It does not set up the selected service as a notification channel. Note that when selecting "Amazon SNS", only SNS topics that start with "grafana" will be accessible.

☐	Notification channel name
☐	Amazon SNS

Cancel Previous Next

- Review your Grafana workspace
- Then click "Create"
- Your workspace should have in the home list
- From there click on the Grafana workspace URL
- You will have to sign in

General / Home

Welcome to Amazon Managed Grafana

Need help? [Documentation](#) [Tutorials](#) [Community](#)

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE
Add your first data source

Learn how in the docs

COMPLETE
Create your first dashboard

Learn how in the docs

Dashboards

Starred dashboards

PSM workspace dashboard

Recently viewed dashboards

PSM workspace dashboard

Latest from the blog

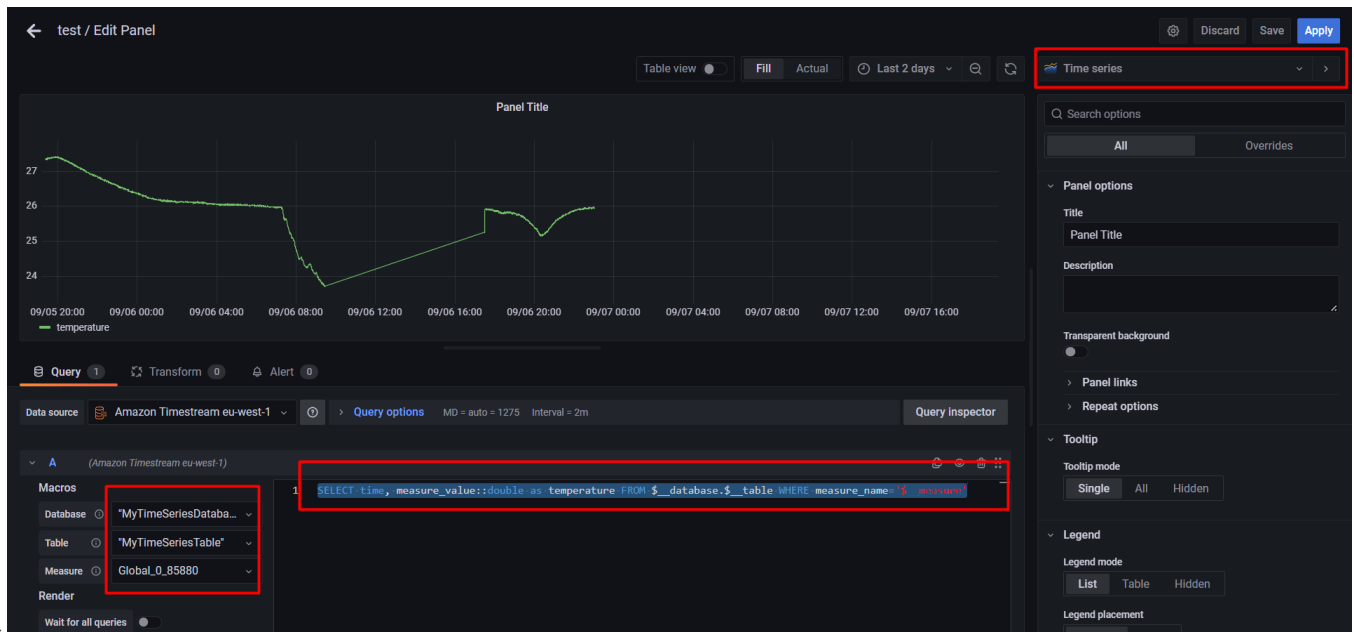
Sep 07
New in Grafana Mimir: Introducing out-of-order sample ingestion
Traditionally the Prometheus TSDB only accepts in-order samples that are less than one hour old, discarding everything else. Several use cases, however, need out-of-order support. For example: IoT devices waking up asynchronously and writing metrics. Complex metric delivery architectures using message buses like Kafka with randomized sharding and delay due to congestion. Standalone Prometheus instances isolated from a network connection for some time trying to push old samples. As Prometheus continues to be adopted in new fields and industries, the limitation of in-order sampling poses an increasingly hard problem.

Sep 06
HLRE Learning with Grafana Labs Senior Engineering Manager *Mirko Krutz*

- Once you have access to the Grafana dashboard, click on the plus "+" symbol, then "Create a dashboard"
- Click "Add panel"
- Select time series
- Copy the following SQL request

```
SELECT time, measure_value::double as temperature FROM $__database.$__table WHERE measure_name='$__measure'
```

- Select your AWS Timestream data source
- Select the database
- Select the table
- Select the measure



- Click Apply
- Exit and save your dashboard

Grafana SQL commands samples

- Data source: "Amazon Timestream"
- Select your database and table for "\$__database" and "\$__table".

Plot temperature for all BLE temperature sensors found

Select measure ("\$_measure") : "Global_0_85880":

```
SELECT device_id, CREATE_TIME_SERIES(time, measure_value::double) as temperature
FROM $__database.$__table
WHERE $__timeFilter AND measure_name='$_measure'
GROUP BY device_id
ORDER BY device_id
```

Plot distribution of received messages count per device

```
SELECT device_id, COUNT(*) as message_counts
FROM $__database.$__table
WHERE $__timeFilter
GROUP BY device_id
ORDER BY device_id
```

Plot BLE magnetic sensor counter value:

Select measure "Global_0_85895":

```
SELECT time, measure_value::bigint
FROM $__database.$__table
WHERE measure_name='$_measure' AND $__timeFilter
```

Plot measured voltage 1 from U30 device:

Select measure "Global_O_110228"

```
SELECT device_id, CREATE_TIME_SERIES(time, measure_value::double) as "V1"
FROM $__database.$__table
WHERE $__timeFilter
AND measure_name='$_measure'
AND $__timeFilter
GROUP BY device_id
ORDER BY device_id
```

10 - Troubleshooting

- If you're experiencing any issues or need further assistance, reach us at <https://www.socomec.co.uk/en-gb/technical-support>.
- AWS Policy: More documentation about Policies at [AWS IoT Core policies](#)
- Certificates: More information about x509 certificate at <https://docs.aws.amazon.com/iot/latest/developerguide/x509-client-certs.html>

Annex

1 - Dashboard demo



2 - Socomec data types description

Socomec data (called "service") are labelled depending on their multiplicity and type. Following table describe all services that are send to AWS. Devices which uses a given service are listed in the "products" column. If a service is not related to specific product, the cell is empty.

Service	Products	Unit	Name / Description
Global_0_5204	ATS Loose controller ATys C65 IEC [2402] Atys p [2300]	-	Alarm/Fault Code
Global_0_9000		-	ILoadAvgMeasurementCplx

Global_0_9001		-	ILoadEnergyMeasurementCplx
Load_0_10179 Load_1_10179 Load_2_10179	Diris Digiware I-35 DC [4118]	A	IDCIrmsAvg IDCIrmsAvg IDCIrmsAvg
Load_0_10180 Load_1_10180 Load_2_10180	Diris Digiware I-35 DC [4118]	A	IDCIdcAvg IDCIdcAvg IDCIdcAvg
Load_0_10181 Load_1_10181 Load_2_10181	Diris Digiware I-35 DC [4118]	A	IDCIacAvg IDCIacAvg IDCIacAvg
Load_0_10220 Load_1_10220 Load_2_10220	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	Crest factor : I1 Crest factor : I1 Crest factor : I1
Load_0_10221 Load_1_10221 Load_2_10221	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	Crest factor : I2 Crest factor : I2 Crest factor : I2
Load_0_10222 Load_1_10222 Load_2_10222	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	Crest factor : I3 Crest factor : I3 Crest factor : I3
Load_0_10223 Load_1_10223 Load_2_10223	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	Crest factor : In Crest factor : In Crest factor : In
Global_0_30276	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	IEnergiesTotalCplx
Global_0_45136	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	ILoadMetroAvgVUCplx
Global_0_45137	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	ILoadMetroAvgIPQSCplx
Multi_Fluid_0_65014 Multi_Fluid_1_65014 Multi_Fluid_2_65014	Diris Digiware IO-10 [4114]	-	MFF 1 Total MFF 2 Total MFF 3 Total
Multi_Fluid_0_65015 Multi_Fluid_1_65015 Multi_Fluid_2_65015	Diris Digiware IO-10 [4114]	-	MFF 1 Partial MFF 2 Partial MFF 3 Partial
Load_0_80031 Load_1_80031 Load_2_80031	Diris Digiware I-35 DC [4118]	W	P tot Load 1 P tot Load 2 P tot Load 3
Global_0_85880	Environmental sensor : ELA RHT [30000]	°C	Temperature
Global_0_85894	Environmental sensor : ELA RHT [30000]	%	Humidity
Global_0_85895	Environmental sensor : ELA MAG [30001]	-	IBleSensorMagCount
Global_0_85896	Environmental sensor : ELA MAG [30001]	-	IBleSensorMagPresent
Load_0_100000 Load_1_100000 Load_2_100000	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Ph-N Voltage total harmonic distortion : THD V1 Ph-N Voltage total harmonic distortion : THD V1 Ph-N Voltage total harmonic distortion : THD V1
Load_0_100001 Load_1_100001 Load_2_100001	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Ph-N Voltage total harmonic distortion : THD V2 Ph-N Voltage total harmonic distortion : THD V2 Ph-N Voltage total harmonic distortion : THD V2
Load_0_100002 Load_1_100002 Load_2_100002	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Ph-N Voltage total harmonic distortion : THD V3 Ph-N Voltage total harmonic distortion : THD V3 Ph-N Voltage total harmonic distortion : THD V3
Load_0_100003 Load_1_100003 Load_2_100003	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Ph-Ph Voltage total harmonic distortion : THD U12 Ph-Ph Voltage total harmonic distortion : THD U12 Ph-Ph Voltage total harmonic distortion : THD U12
Load_0_100004 Load_1_100004 Load_2_100004	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Ph-Ph Voltage total harmonic distortion : THD U23 Ph-Ph Voltage total harmonic distortion : THD U23 Ph-Ph Voltage total harmonic distortion : THD U23
Load_0_100005 Load_1_100005 Load_2_100005	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Ph-Ph Voltage total harmonic distortion : THD U31 Ph-Ph Voltage total harmonic distortion : THD U31 Ph-Ph Voltage total harmonic distortion : THD U31
Load_0_100006 Load_1_100006 Load_2_100006	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Curent total harmonic distortion : THD I1 Curent total harmonic distortion : THD I1 Curent total harmonic distortion : THD I1
Load_0_100007 Load_1_100007 Load_2_100007	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Curent total harmonic distortion : THD I2 Curent total harmonic distortion : THD I2 Curent total harmonic distortion : THD I2

Load_0_100008 Load_1_100008 Load_2_100008	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Curent total harmonic distortion : THD I3 Curent total harmonic distortion : THD I3 Curent total harmonic distortion : THD I3
Load_0_100009 Load_1_100009 Load_2_100009	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Curent total harmonic distortion : THD In Curent total harmonic distortion : THD In Curent total harmonic distortion : THD In
Load_0_100056 Load_1_100056 Load_2_100056	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	K-Factor I1 K-Factor I1 K-Factor I1
Load_0_100057 Load_1_100057 Load_2_100057	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	K-Factor I2 K-Factor I2 K-Factor I2
Load_0_100058 Load_1_100058 Load_2_100058	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	K-Factor I3 K-Factor I3 K-Factor I3
Load_0_100059 Load_1_100059 Load_2_100059	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	-	K-Factor In K-Factor In K-Factor In
Load_0_100060 Load_1_100060 Load_2_100060	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	System THD V System THD V System THD V
Load_0_100061 Load_1_100061 Load_2_100061	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	System THD U System THD U System THD U
Load_0_100062 Load_1_100062 Load_2_100062	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	System THD I System THD I System THD I
Global_0_100120	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Total demand distortion : TDD I1
Global_0_100121	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Total demand distortion : TDD I2
Global_0_100122	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Total demand distortion : TDD I3
Global_0_100123	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	Total demand distortion : TDD In
Global_0_100124	DIRIS PMD US Medium level RJ12 [50] DIRIS PMD US Medium level 333mV [51]	%	System TDD I
Global_0_110343	Diris Digiware I-35 DC [4118]	V	ILoadDCVrmsAvg
Global_0_110344	Diris Digiware I-35 DC [4118]	V	ILoadDCVdcAvg
Global_0_110345	Diris Digiware I-35 DC [4118]	V	ILoadDCVacAvg
Global_0_135000	ATS3 Socomec pM [1000] ATS Loose controller ATyS C65 IEC [2402] Atys p [2300]	-	Priority
Global_0_135001	ATS3 Socomec pM [1000]	-	Alarm/Fault Code
Global_0_135030	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Type of Application
Global_0_135038	ATS Loose controller ATyS C65 IEC [2402] ATS3 Socomec pM [1000] Atys Bypass dual line [2601] Atys Bypass single line [2600] Atys p [2300]	-	Operating Mode
Global_0_135039	ATS Loose controller ATyS C65 IEC [2402] ATS3 Socomec pM [1000] Atys Bypass dual line [2601] Atys Bypass single line [2600] Atys p [2300]	-	Switch Position
Global_0_135041	ATS Loose controller ATyS C65 IEC [2402]	-	Source 1 Start Generator relay State
Global_0_135042	ATS Loose controller ATyS C65 IEC [2402] Atys p [2300]	-	Source 2 Start Generator relay State
Global_0_135044	ATS Loose controller ATyS C65 IEC [2402] ATS3 Socomec pM [1000] Atys Bypass dual line [2601] Atys Bypass single line [2600] Atys p [2300]	-	Source 1 State

Global_0_135045	ATS Loose controller ATyS C65 IEC [2402] ATS3 Socomec pM [1000] Atys Bypass dual line [2601] Atys Bypass single line [2600] Atys p [2300]	-	Source 2 State
Global_0_135046	ATS Loose controller ATyS C65 IEC [2402] ATS3 Socomec pM [1000] Atys Bypass dual line [2601] Atys Bypass single line [2600] Atys p [2300]	-	Test in progress
Global_0_135048	ATS3 Socomec pM [1000]	-	Cycle Counter
Global_0_135049	ATS3 Socomec pM [1000]	-	Position 1 Manoeuvre counter
Global_0_135050	ATS3 Socomec pM [1000]	-	Position 2 Manoeuvre counter
Global_0_135052	ATS Loose controller ATyS C65 IEC [2402] ATS3 Socomec pM [1000] Atys Bypass dual line [2601] Atys Bypass single line [2600] Atys p [2300]	-	Alarm/Fault summary
Global_0_148015	PMS (Power Management System) [8272]	-	S012: General Alarm
Global_0_148021	PMS (Power Management System) [8272]	-	S005: Inverter ready
Global_0_160007	PMS (Power Management System) [8272]	%	Bank SOC
Global_0_160164	PMS (Power Management System) [8272]	Wh	Nominal energy for 1 string
Global_0_160253	PMS (Power Management System) [8272]	°C	External temperature
Global_0_160606	PMS (Power Management System) [8272]	-	Application - Generic alarm 11
Global_0_160914	PMS (Power Management System) [8272]	W	M016: Grid Active Power (signed)
Global_0_160916	PMS (Power Management System) [8272]	W	M018: ESS Active Power (signed)
Global_0_160920	PMS (Power Management System) [8272]	V	M025: DC voltage on ESS
Global_0_160922	PMS (Power Management System) [8272]	%	M027: Battery State-Of-Charge
Global_0_160923	PMS (Power Management System) [8272]	%	M028: Battery State-Of-Health
Global_0_160924	PMS (Power Management System) [8272]	-	M029: Number of battery racks connected
Global_0_160931	PMS (Power Management System) [8272]	W	Power wind turbines
Global_0_160932	PMS (Power Management System) [8272]	W	Power PV
Global_0_160933	PMS (Power Management System) [8272]	W	Power Genset
Global_0_160944	PMS (Power Management System) [8272]	-	Counter balancing
Global_0_160945	PMS (Power Management System) [8272]	Wh	Battery energy discharged
Global_0_160950	PMS (Power Management System) [8272]	-	Number of Source
Global_0_160951	PMS (Power Management System) [8272]	-	Output type
Global_0_160952	PMS (Power Management System) [8272]	W	Output power
Item_0_160953	PMS (Power Management System) [8272]	-	Type of source
Item_0_160954	PMS (Power Management System) [8272]	-	Status of the contact
Item_0_160955	PMS (Power Management System) [8272]	W	Source nominal power
Item_0_160956	PMS (Power Management System) [8272]	W	Source measured power
Item_0_160957	PMS (Power Management System) [8272]	-	Measured power is available
Global_0_160960	PMS (Power Management System) [8272]	-	Counter System in alarm
Global_0_160961	PMS (Power Management System) [8272]	-	Counter System ON
Global_0_160962	PMS (Power Management System) [8272]	-	Counter PV On
Global_0_160963	PMS (Power Management System) [8272]	-	Counter Genset On
Global_0_160964	PMS (Power Management System) [8272]	%	Deep of Discharge of batteries set
Global_0_160965	PMS (Power Management System) [8272]	-	Counter Wind turbine on
Global_0_165000	PMS (Power Management System) [8272]	s	Delta Counter System in alarm
Global_0_165001	PMS (Power Management System) [8272]	s	Delta Counter System ON
Global_0_165002	PMS (Power Management System) [8272]	s	Delta Counter PV On
Global_0_165003	PMS (Power Management System) [8272]	s	Delta Counter Genset On

Global_0_165004	PMS (Power Management System) [8272]	s	Delta Counter Wind Turbine On
Global_0_165005	PMS (Power Management System) [8272]	s	Delta Counter balancing
Global_0_165006	PMS (Power Management System) [8272]	Wh	Delta Battery energy discharged
Global_0_165010	PMS (Power Management System) [8272]	%	ESS System availability
Global_0_165011	PMS (Power Management System) [8272]	-	Battery cycle number per day
Global_0_165012	PMS (Power Management System) [8272]	%	Battery load rate
Global_0_400013	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Bypass internal temperature
Global_0_400015	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Bypass internal humidity
Global_0_400016	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Bypass source 1 state
Global_0_400017	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Bypass isolated state
Global_0_400018	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Bypass source 2 state
Global_0_400101	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch operating factor alarm
Global_0_400102	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch neutral alarm
Global_0_400103	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch fault 1 alarm
Global_0_400104	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch fault 2 alarm
Global_0_400105	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch alarm 1 alarm
Global_0_400106	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch alarm 2 alarm
Global_0_400107	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch phase rotation 1 alarm
Global_0_400108	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch phase rotation 2 alarm
Global_0_400109	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch capa 1 alarm
Global_0_400110	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch capa 2 alarm
Global_0_400111	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch power 1 alarm
Global_0_400112	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch power 2 alarm
Global_0_400113	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch position 1 alarm
Global_0_400114	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch position 2 alarm
Global_0_400115	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch position 0 alarm
Global_0_400116	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch fault 0 alarm
Global_0_400117	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch unbalanced 1 alarm
Global_0_400118	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch unbalanced 2 alarm
Global_0_400119	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch mainfault alarm
Global_0_400120	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch motor fault alarm
Global_0_400121	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch autoconf failed alarm
Global_0_400122	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch unexpected transfer alarm

Global_0_400123	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch fail to transfer alarm
Global_0_400125	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch max power attempts alarm
Global_0_400126	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch genset failed to start alarm
Global_0_400127	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch external fault alarm
Global_0_400128	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch unknown position alarm
Global_0_400120	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch motor fault alarm
Global_0_400121	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch autoconf failed alarm
Global_0_400122	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch unexpected transfer alarm
Global_0_400123	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch fail to transfer alarm
Global_0_400125	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch max power attempts alarm
Global_0_400126	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch genset failed to start alarm
Global_0_400127	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	Switch external fault alarm
Global_0_400128	Atys Bypass single line [2600] Atys Bypass dual line [2601]	-	

3 - Alternate provisioning methods

a) Gateway generates certificate

If you decide to use the x509 certificate generated by the gateway. Instead of uploading credentials in webview, you shall rather :

- Fill the thing name
- Fill the AWS endpoint
- Select "Auto-generate"
- Click on the "Generate" button

AWS Cloud Connection

Enable Cloud

Certificate management

Endpoint

Thing name

Certificate management

Generate

- Download the generated public certificate
- Register the public certificate to AWS and attach it to your thing (with the proper policy)

b) Gateway generates CSR, signed by AWS

If you decide to use the CSR generated by the gateway:

- Fill the thing name

- Fill the AWS endpoint
- Click the button to generate a CSR, and download it.

Certificate Signing Request (CSR)

Thing name

ldd_d70_qual

Generate CSR

- Give the CSR to AWS, so that it can sign and generate a certificate for your device.
- Download the signed public certificate from AWS
- Upload the certificate to the gateway

AWS Cloud Connection

Enable Cloud

Certificate management

Endpoint

a2urlgdsiqqpee-ats.iot.eu-wes

Thing name

ldd_d70_qual

Certificate management

Upload Signed Certificate

Certificate

None

Browse

c) Gateway generates CSR, custom CA signs the certificate

If you desire to use your own CA to sign the CSR, please refer to following documentation, main steps are:

CA:

- Generate a CA
- Register your CA in AWS

Device (Webview) :

- Fill the thing name
- Fill the AWS endpoint
- Click the button to generate a CSR, and download it.

Certificate Signing Request (CSR)

Thing name

ldd_d70_qual

Generate CSR

- Sign the CSR with your CA
- Upload the signed certificate to the gateway

AWS Cloud Connection

✕

✓

Enable Cloud

Certificate management

Endpoint

i

a2urlgdsiqqpee-ats.iot.eu-wes

Thing name

i

ldd_d70_qual

Certificate management

📄

Upload Signed Certificate

⌵

Certificate

None

Browse

Furthermore, following guide demonstrates how to create a custom CA certificate and register it to AWS to enable JITP : [Set up JITP with AWS IoT Core \(amazon.com\)](#)



In case you generate a certificate or a CSR from the gateway, make sure you have filled the thing name before clicking the button in Webview. Certificate and CSR use the thing name as common name, which is required to authenticate the device from AWS.



When generating a x509 certificate by yourself, subject name is limited in size and should not exceed 128 bytes.