Connectivity Guide

# Kepware MQTT Agent and Google Cloud IoT Core

March 2020
Ref. 1.001

# Table of Contents


1. Configure Google Cloud Platform ..... 3
   - 1.1 Create a New Project ..... 3
   - 1.2 Create a Registry ..... 4
   - 1.3 Create Certificates ..... 5
   - 1.4 Create Device ..... 6
2. Configure MQTT Client in KEPServerEX ..... 7
   - 2.1 Create JWT ..... 7
   - 2.2 Configure MQTT Client ..... 7
3. Review Data Flow on Google Cloud ..... 8
   - 3.1 Subscribe to Topic ..... 8
   - 3.2 Activate Cloud Shell ..... 9

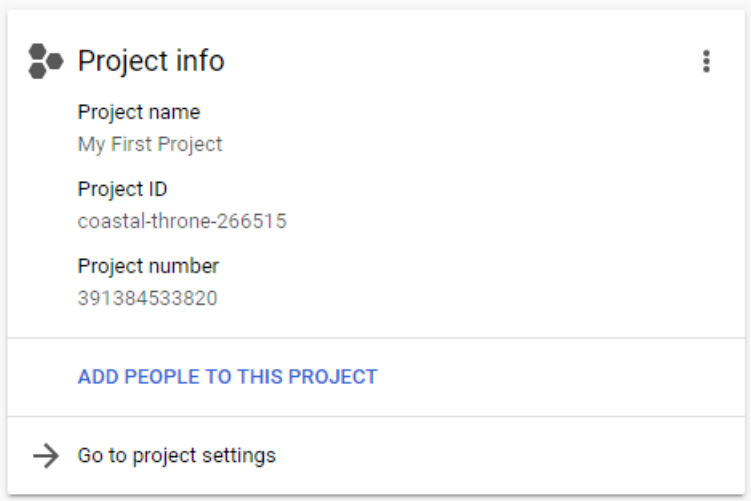

 

This document instructs users how to connect KEPServerEX® MQTT Agent and Google Cloud IoT Core to manage and deliver data. These instructions assume the user is registered with Google Cloud, and that OpenSSL is available to create self-signed certificates.

# 1. Configure Google Cloud Platform

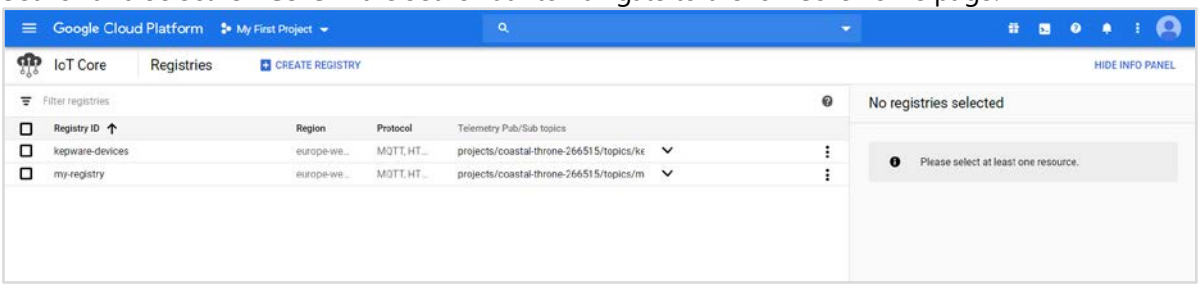
## 1.1 Create a New Project

Once registered to Google Cloud, an initial project is created on the account. Use this project or create a new project for this exercise. Note the Project ID, which will be required later.



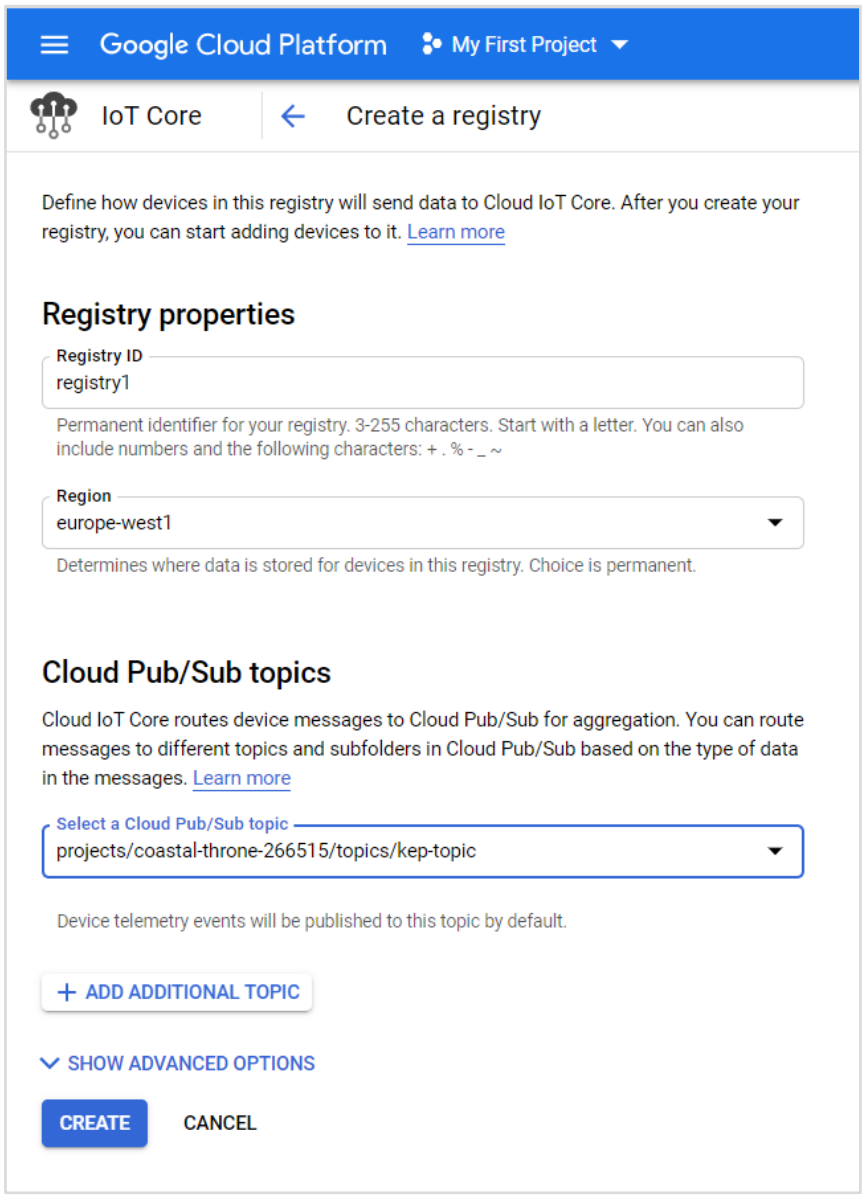

To enable the Cloud IoT Core and Cloud Pub/Sub APIs for the project, access the link below and select the desired project from the dropdown menu:
https://console.cloud.google.com/flows/enableapi?apiid=cloudiot.googleapis.com,pubsub&ga=2.96525144.639288887.1580223926-1334733146.1576681398

Search and select **IoT Core** in the search bar to navigate to the IoT Core home page.



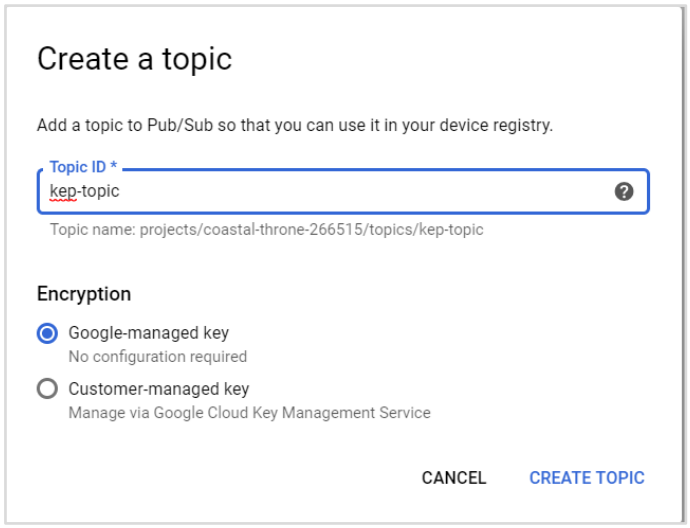
## 1.2 Create a Registry

Create a registry by entering a Registry ID, and selecting a region, and selecting a Cloud Pub/Sub topic.




Review **SHOW ADVANCED OPTIONS** to ensure MQTT and HTTP protocols are selected.

Create a topic to add to the Pub/Sub for the device registry.Enable **Google-managed key** for encryption.

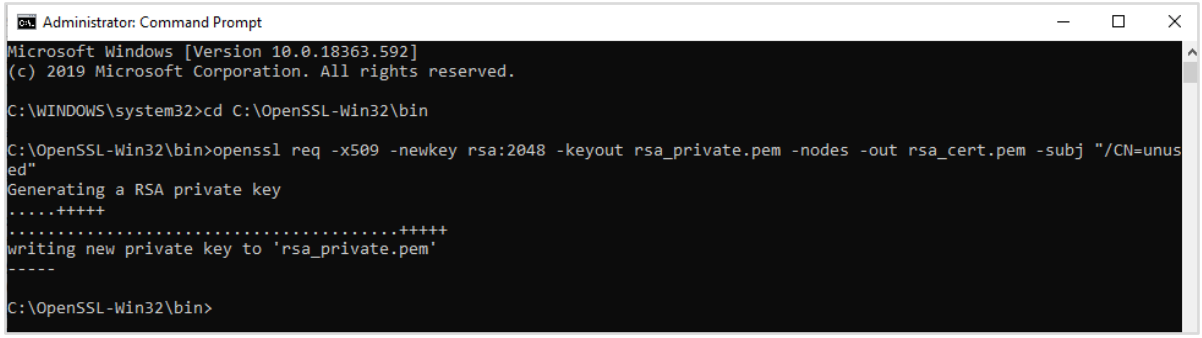



## 1.3 Create Certificates

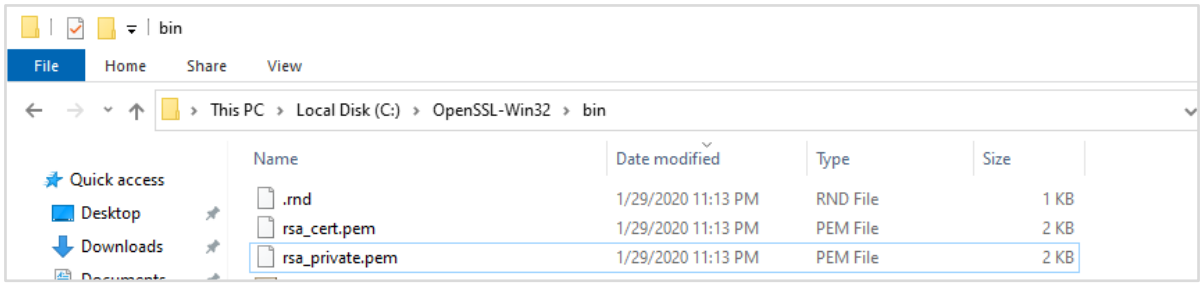
Certificates are required for both device and Java Web Token (JWT) authentication. To create self-signed certificates, download and install OpenSSL: https://www.openssl.org/

Once the OpenSSL is installed, open a command prompt and run the following command to create a certificate and private key:

```
openssl req -x509 -newkey rsa:2048 -keyout rsa_private.pem -nodes -out
rsa_cert.pem -subj "/CN=unused"
```

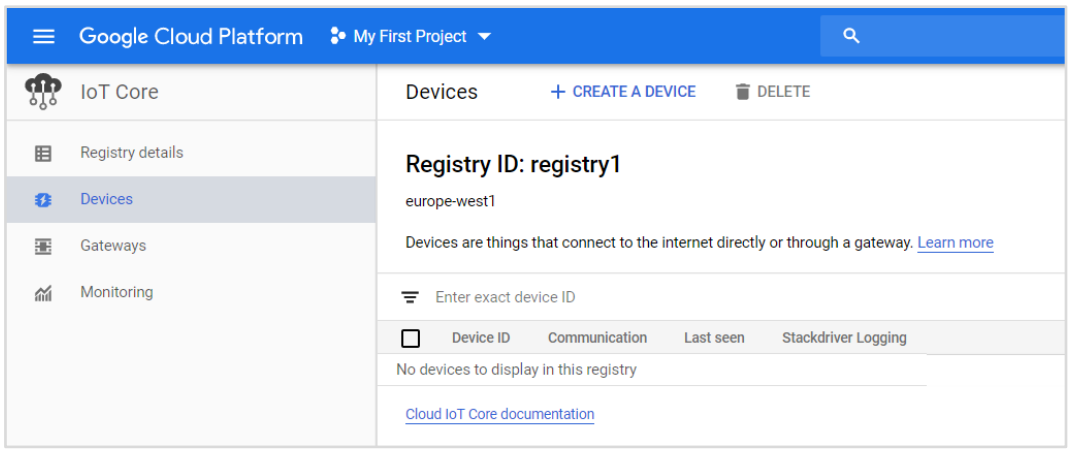



Navigate to **...\OpenSSL-Win32\bin** and open **rsa_cert.pem** with a text editor. Copy the contents to use during Device creation.

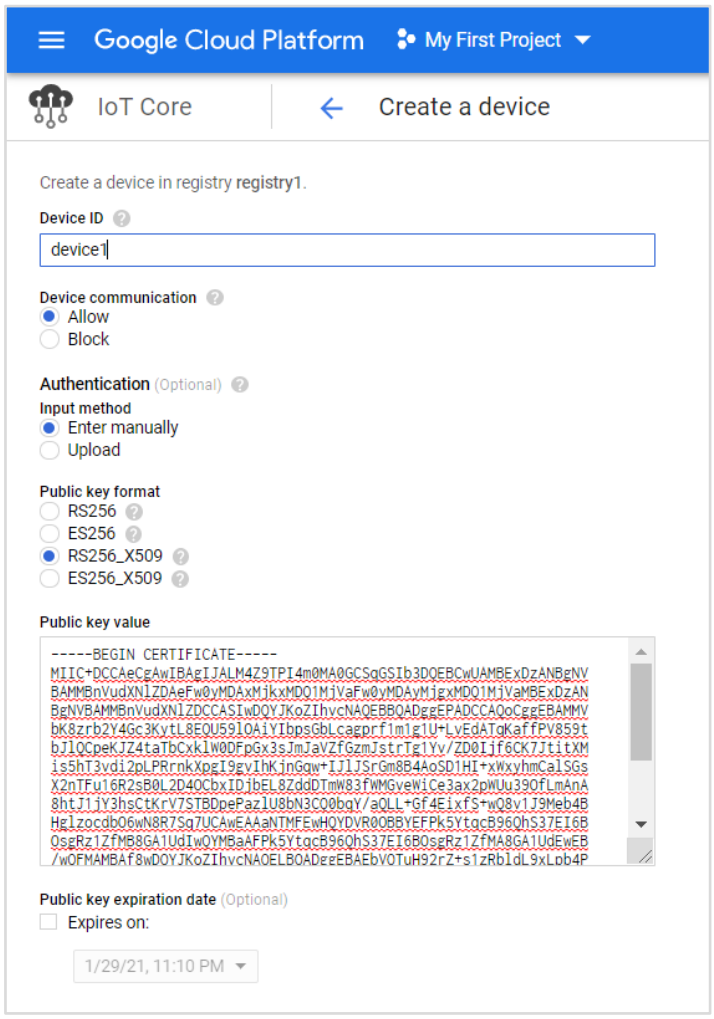


## 1.4 Create Device

Return to the Google Cloud Platform and select **+ CREATE A DEVICE** in the **Devices** section.




Enter a Device ID, and select **RS256_X509** as the public key format. Do not adjust the default device communication and authentication settings. Paste the certificate and private key text in the **Public key value** field, then click **CREATE**.




This completes the Google Cloud Platform configuration.

# 2. Configure MQTT Client in KEPServerEX

## 2.1 Create JWT

For the MQTT Client, a JSON Web Token (JWT) is required.

- *For JWT creation examples, refer to the following Google Cloud documentation: https://cloud.google.com/iot/docs/how-tos/credentials/jwts*

## 2.2 Configure MQTT Client

- *For more information, see the IoT Gateway Manual.*

In KEPServerEX, click **Add Agent...** under the **IoT Gateway** plugin.

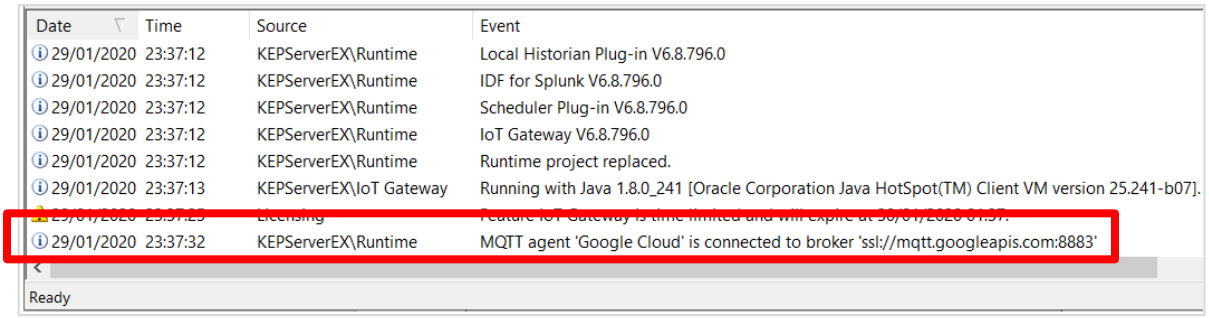
Enter a name and select **MQTT Client** as the agent type.

Continue through the configuration wizard to set the following parameters:

| Property | Value |
|---|---|
| URL | **ssl://mqtt.googleapis.com:8883** |
| Topic | **/devices/<DEVICE_ID>/events** |
| Client ID | **projects/<PROJECT_ID>/locations/</REGION>/registries/<REGISTRY_ID>/devices/<DEVICE_ID>** |
| Username | **<USERNAME>** |
| Password | **JWT** |

- **Note**: A username is required when setting a password. For this scenario, any username is acceptable.

Add tags to the MQTT Agent. Once added, a message on Event Log should confirm a connection to Google Cloud IoT Core.

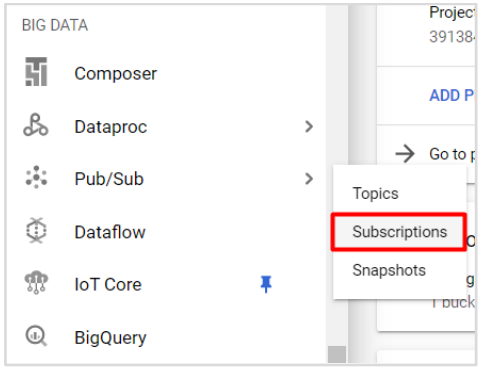


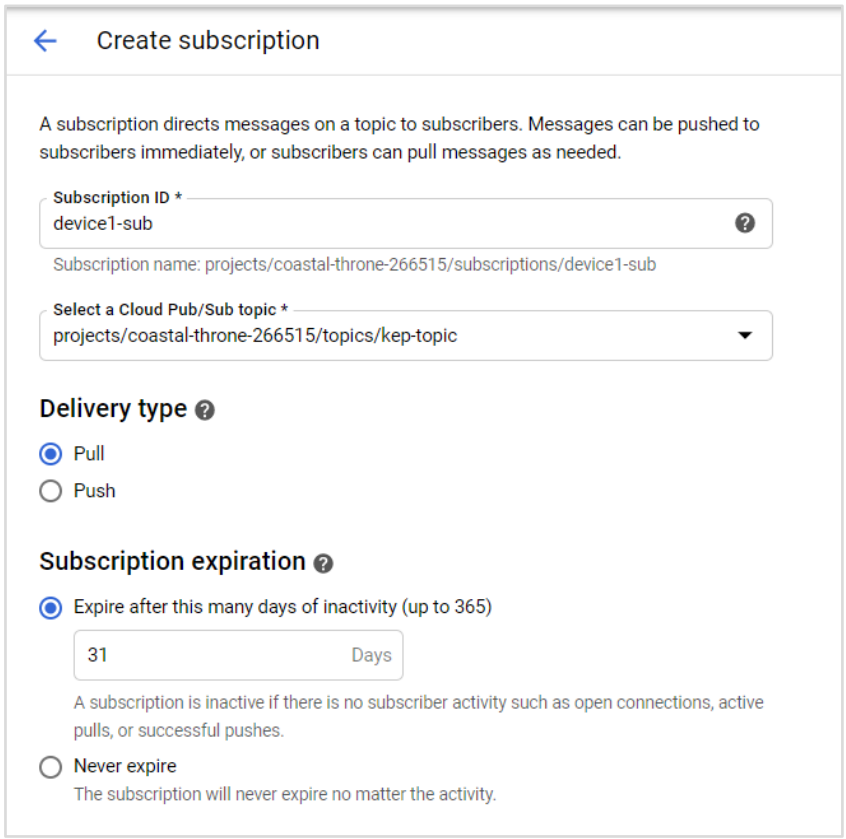
# 3. Review Data Flow on Google Cloud

## 3.1 Subscribe to Topic

To view the data flow on Google Cloud, create a subscription for the topic from the registry. On Google Cloud, navigate to **Pub/Sub | Subscriptions** in the project.




Choose the appropriate Cloud Pub/Sub topic from the dropdown menu (see Step 1.2), and set delivery type to **Pull**.



## 3.2 Activate Cloud Shell

In the Google Cloud Console, click **Activate Cloud Shell**.

In Cloud Shell, run the following command:

```
gcloud pubsub subscriptions pull --auto-ack
projects/<PROJECT_ID>/subscriptions/<SUBSCRIPTION_ID>
```

Under **DATA**, the MQTT Client should deliver the JSON payload. In this example, the *Simulation Examples.Functions.Ramp1* tag is published by the MQTT Agent.