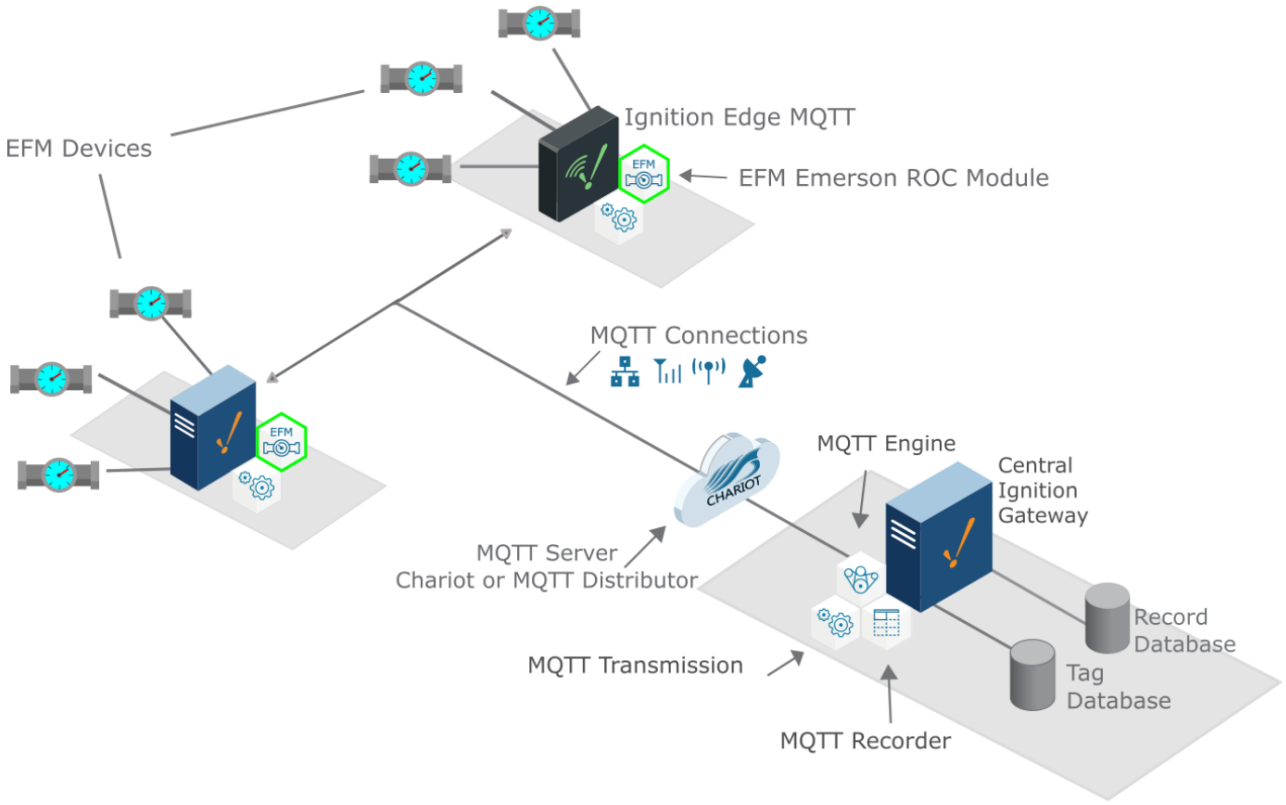
Sending ROC Events to a Central Ignition Gateway

Prerequisites

- Installing Ignition
- Installing the following MQTT Modules on two Ignition systems
    - Ignition System 1 (Central Ignition Gateway)
        - MQTT Distributor
        - MQTT Engine
        - MQTT Recorder
    - Ignition System 2 (Remote/Edge Ignition Gateway)
        - MQTT Transmission
        - EFM Emerson ROC driver module

Overview

The EFM Emerson ROC module is capable of polling events from a ROC device based on a specified polling rate. With MQTT Transmission, these events can be published as Sparkplug records to an MQTT server. Any client subscribed on Sparkplug RECORD messages can receive these objects. In addition, MQTT Engine when combined with MQTT Recorder can also receive these messages and store these objects in a configured Ignition database. The following drawing shows the general architecture used to do this. This tutorial outlines the process of getting events to the central Ignition gateway.



Sending ROC Events to a Central Ignition Gateway

We must configure a total of five Cirrus Link modules on two different Ignition gateways to get event data flowing from an Emerson ROC device to a central database. These are:

- Central Ignition Gateway
    - MQTT Distributor
    - MQTT Engine
    - MQTT Recorder
- Remote/Edge Ignition Gateway
    - MQTT Transmission
    - EFM Emerson ROC

The configuration of each of these modules is covered below based on the Ignition gateway they're installed on.

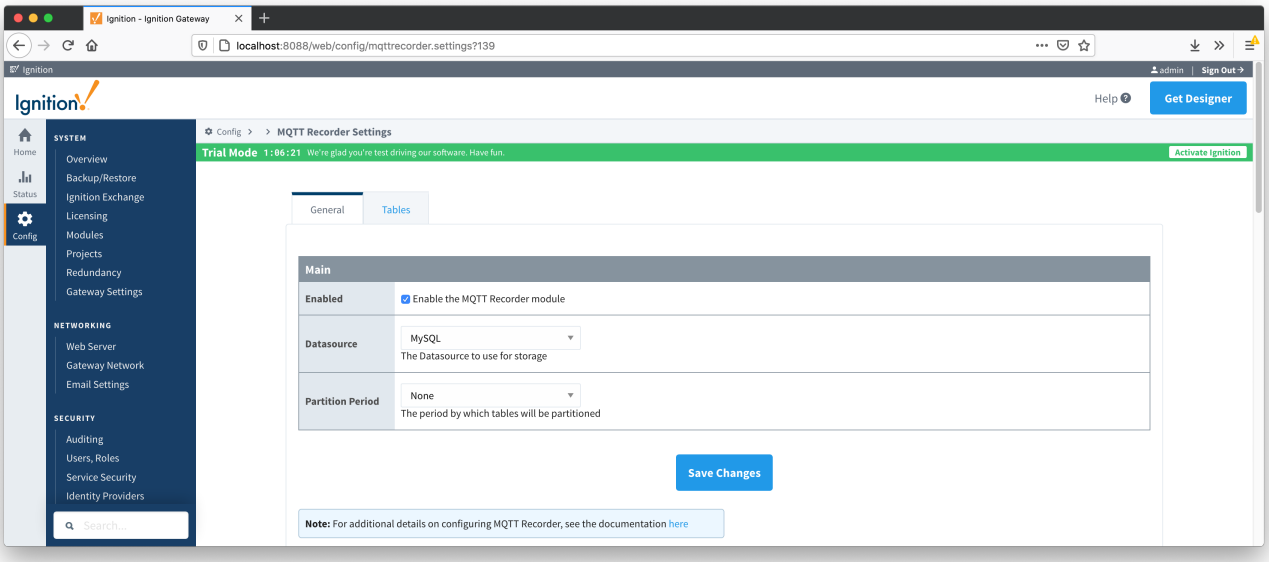
Central Ignition Gateway Setup

MQTT Distributor can be left in its default configuration. Note in a production system you would likely want to set up TLS especially if MQTT connections are using the Internet.

MQTT Engine can also be left in its default configuration.

MQTT Recorder requires that a database be set up in Ignition. That can be done as described in the 'Connect to a Database' section here. Note Ignition supports additional database types. For more detailed information about supported types, take a look at the information provided here. Once a database is set up, MQTT Recorder can be configured. Do so by opening the Ignition Gateway Web UI and browsing to the Configure tab at the top of the screen and then selecting 'MQTT Recorder Settings' as shown in the lower left below.

Once there, select a Datasource as shown in the image below. This drop-down will be populated with any database connections set up in Ignition. Optionally, a Partition Period can be selected to segregated tables by time periods.

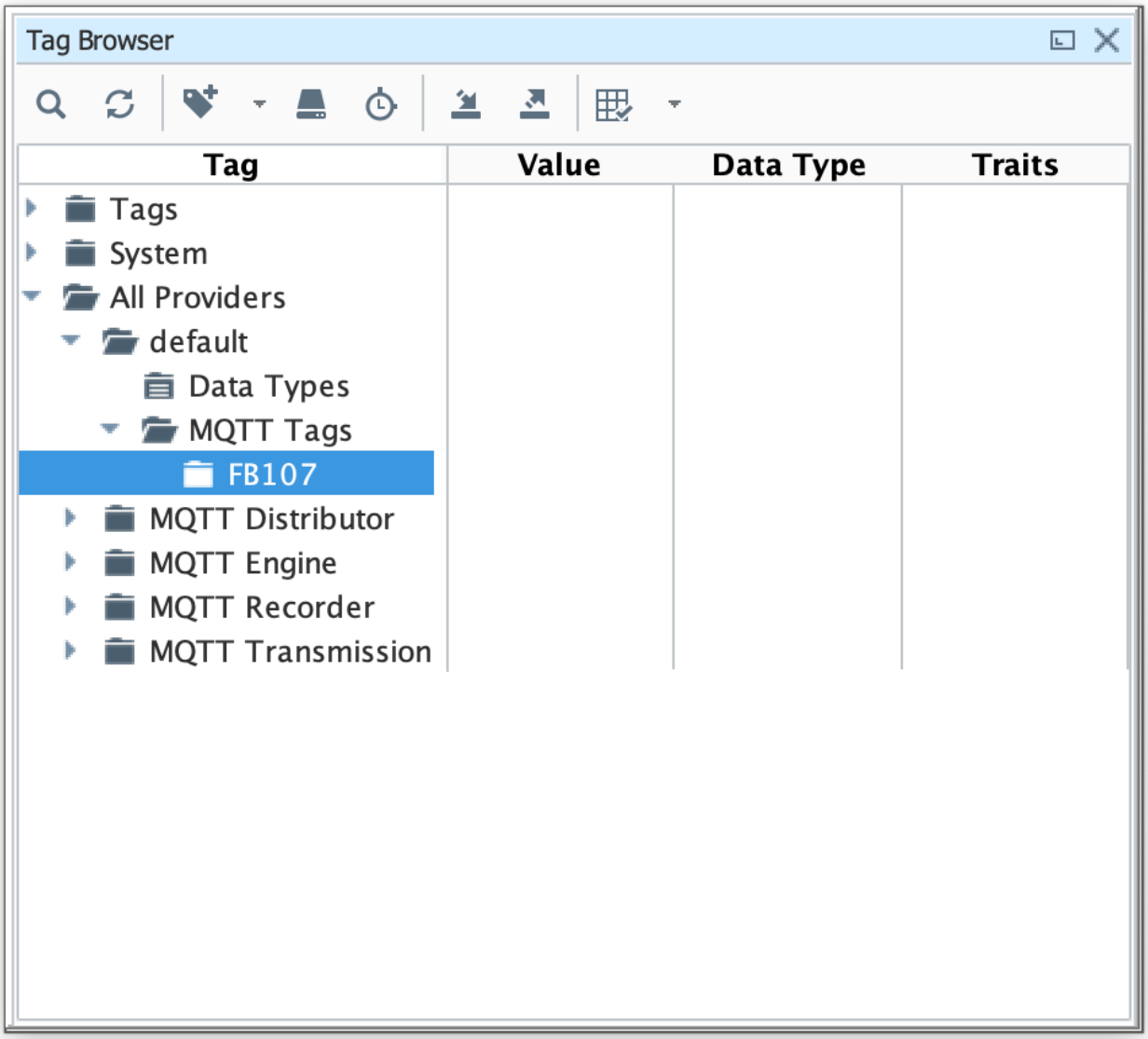


At this point, the Central Ignition Gateway with MQTT Distributor, MQTT Engine, and MQTT Recorder is fully configured and ready to receive MQTT Sparkplug messages from the Remote/Edge Ignition Gateway. MQTT Distributor listens on TCP port 1883 by default for inbound MQTT connections. Make sure the Operating System's Firewall, Antivirus, and Malware protection services allow inbound connections on port 1883/TCP before proceeding.

Remote/Edge Ignition Gateway Setup

With the Central Ignition Gateway ready to receive MQTT/Sparkplug RECORD objects, the EFM Emerson ROC and MQTT Transmission modules can be configured on the Remote/Edge Ignition Gateway.

Start by configuring the MQTT Transmission module. Do so by opening Ignition Designer and creating a tag structure similar to what is shown below with [default]MQTT Tags/FB107 folders:



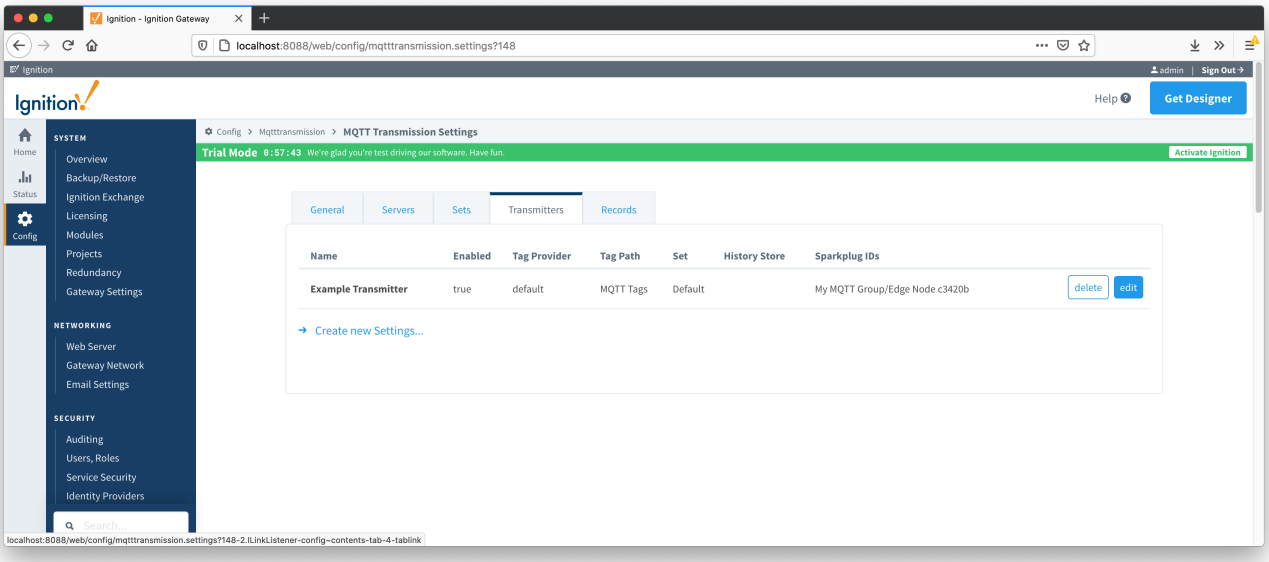
Note this structure is based on usage of the 'Example Transmitter' in MQTT Transmission in Ignition8. So, the directory structure is very important. Note the structure.

- [tag provider]MQTT Tags/[Device ID]/...

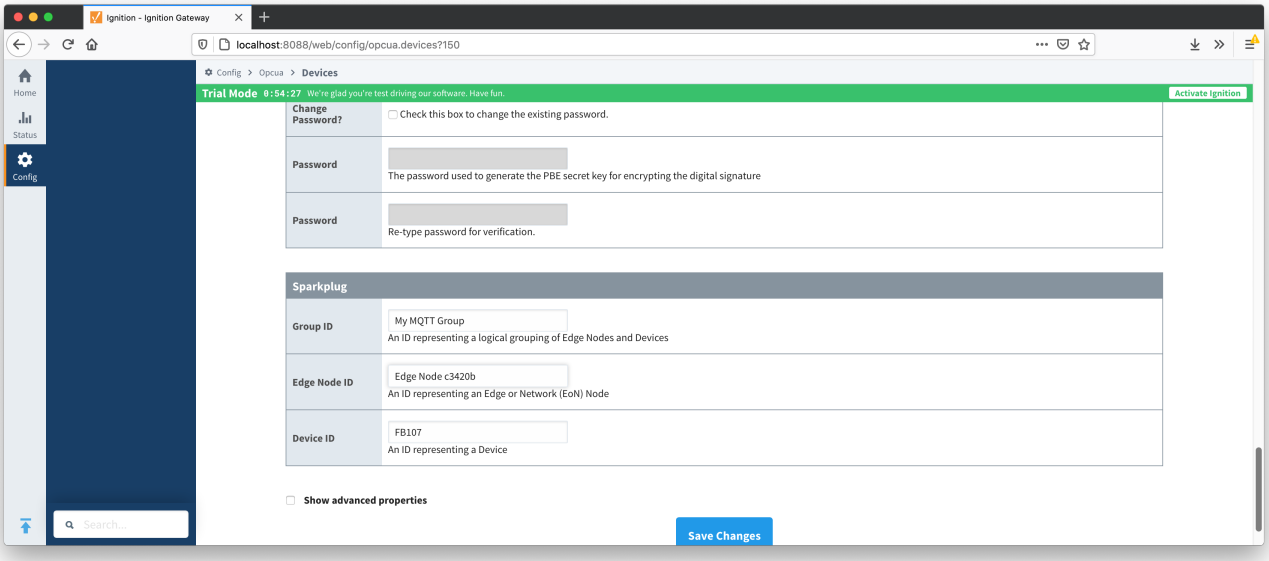
In the example below this implies the following definitions:

- [Device ID] = FB107

Note since we are using the 'Example Transmitter' of Ignition8 we also need to set the Group ID and Edge Node ID in the Example Transmitter. To do so, go to MQTT Transmission Settings on the left Ignition Gateway Web UI navigation panel. Then select the 'Transmitters' tab as shown below:

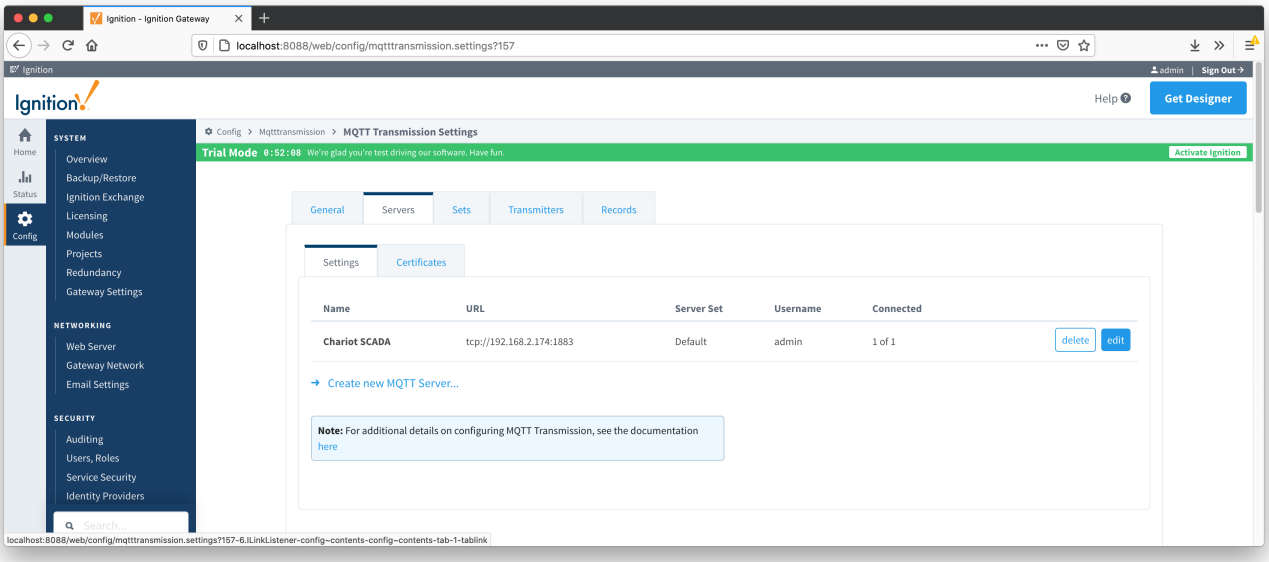


Note that the 'Sparkplug IDs' are 'My MQTT Group/Edge Node c3420b' in this example. These exact values will be used for the EFM Emerson ROC connection Sparkplug parameters to tell the EFM Emerson ROC which MQTT Transmission Transmitter configuration to use and, in turn, which MQTT connection to use to send the alarm data on. So, to match the Transmitter configuration in this example, we must set the Sparkplug IDs in the Emerson ROC driver instance as shown below. Note the Group ID and Sparkplug ID are defined in the MQTT Transmission Transmitter configuration. The Device ID is being picked up from the Ignition default Tag Provider tag tree.



Next the MQTT Transmission server configuration must be modified to point to the Central Ignition Gateway we set up earlier. To do so, in the Ignition Gateway Web UI browse to the Configure tab on the top and then to MQTT Transmission Settings in the lower left as shown below.

In the MQTT Transmission Settings configuration, click the Servers tab. Then click 'edit' on the Chariot SCADA MQTT Server definition. Modify the URL to match the URL of the Central Ignition Gateway. In this example, MQTT Distributor is installed on a Central Ignition Gateway at the IP address of 192.168.2.174. Once the URL is modified to match the configuration, there should be a '1 of 1' in the Connected column as shown below.



The next step is configuring the EFM Emerson ROC module. This is done as described in the Emerson ROC Configuration manual. In going through the basic setup and configuration for Event configuration the following steps must be performed:

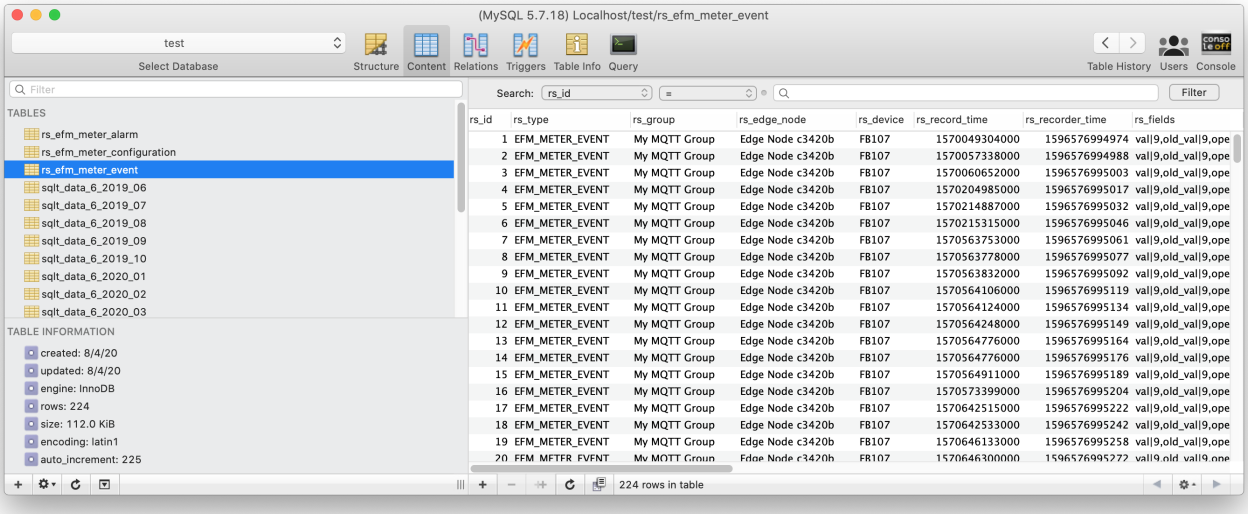
1. Define the global TLP definitions available for all ROC devices in this Ignition instance
2. Upload the Configuration Mappings for all ROC devices in this Ignition instance
3. Create the base device connection to the ROC
    a. The 'Event Scan Rate' in this connection configuration must be greater than zero to tell the driver to poll for event data at the specified rate
    b. Set the Sparkplug Group ID, Edge Node ID, and Device ID that represent this device. These will be used again later in the MQTT Transmission configuration.
4. Specify the subset of global TLP definitions that this specific ROC uses

These steps can be skipped if not configuring the driver to poll for TLP data:

1. Create TLP Template(s) which define groups of TLPs should be polled as a logical group
2. Create TLP Poll Group(s) which specify the logical parameters associated with a given TLP Template
3. Use Ignition designer to pull tags into a tag provider

At this point the EFM Emerson ROC driver is configured and is polling for event data at the rate specified in the EFM Emerson ROC device configuration.

MQTT Transmission is connected to the MQTT Server and as a result MQTT Engine is receiving tag change events. In addition, because an EFM Emerson ROC device has been created and configured with the same Sparkplug Group ID, Edge Node ID, and Device ID, events will also be pushed to the MQTT server as Sparkplug RECORD objects. When event are discovered by the EFM Emerson ROC driver, they will be published to the MQTT server, consumed by MQTT Engine, passed on to MQTT Recorder, and then inserted into the specified database. Below is an view of some event records using a third party database viewing tool.



Additional Resources

- Inductive Automation's Ignition download with free trial
  - https://inductiveautomation.com/downloads/
- Azure Injector download with free trial
  - https://inductiveautomation.com/downloads/third-party-modules
- Questions about this tutorial?
  - Check out the Cirrus Link Forum: https://forum.cirrus-link.com/
  - Contact support: support@cirrus-link.com
- Sales questions
  - Email: sales@cirrus-link.com
  - Phone: +1 (844) 924-7787
- About Cirrus Link
  - https://www.cirrus-link.com/about-us/