GETTING STARTED WITH MQTT IN *groov* PRODUCTS

Form 2350-221129—November 2022

OPTO 22
Your Edge in Automation.™

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

Getting Started with MQTT in *groov* Products
Form 2350-221129—November 2022

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or newer are guaranteed for life. This lifetime warranty excludes reed relay modules, *groov* and SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. Refer to Opto 22 form 1042 for complete warranty information.

Opto 22
Your Edge in Automation.

Table of Contents

1: Getting Started with MQTT

INTRODUCTION

Opto 22's *groov* EPIC® edge programmable industrial controllers and *groov* RIO® edge I/O modules give you many ways to use MQTT data communications—via *groov* Manage, Node-RED, and Ignition Edge from Inductive Automation®. These three methods are all MQTT clients built into your *groov* product. With EPIC or RIO firmware versions R3.4 and higher, you can use multiple methods to send data to multiple locations simultaneously. For example, you can set up multiple brokers and use both strings & Sparkplug payloads.

The goal of this guide is to help you understand which approach or approaches are best for you and to help you get started fast. If you already know which approach you'll take, you can jump to the specific how-to section that best describes the application you have in mind. If you're still deciding on your approach, continue reading for a perspective on the general trade-offs between the available clients.

If you are a *groov* product user but you're on the fence about whether MQTT is for you, we've also included a chapter to help you build your own proof-of-concept system.

In this Guide

This guide includes:

Chapter 1: Getting Started with MQTT—This chapter, which includes how-to steps for setting up an MQTT broker if you don't have one, enabling public tag access, configuring simple data collection, and configuring high-reliability SCADA/IIoT using MQTT

Chapter 2: Security and Fault Tolerance—Configuring system security and additional options for fault tolerance in communications

Chapter 3: Build a Proof of Concept—Steps to build a proof of concept

Appendix A: MQTT Brokers—Help for choosing an MQTT broker if you don't already have one

For Help

Related Documents

groov product user's guides are available under Help in *groov* Manage, and the most recent versions are available on our website. Follow the links below or go to www.opto22.com and search on the form number.

Guide name	Part numbers	Contents	Form #
groov EPIC User's Guide	GRV-EPIC-PR1 GRV-EPIC-PR2	Installing and using a *groov* EPIC system; using Node-RED, MQTT, and Ignition or Ignition Edge	2267

Guide name	Part numbers	Contents	Form #
groov RIO Universal I/O User's Guide	GRV-R7-MM1001-10 GRV-R7-MM2001-10	Installing and using a *groov* RIO universal I/O module; using Node-RED, MQTT, and Ignition or Ignition Edge (MM2 only)	2324
groov RIO Energy Monitoring Unit User's Guide	GRV-R7-I1VAPM-3	Installing and using a *groov* RIO EMU energy monitoring unit; using Node-RED and MQTT	2372
Guide to Networking groov Products	GRV-EPIC-PR1 GRV-EPIC-PR2 GRV-R7-MM1001-10 GRV-R7-MM2001-10 GRV-R7-I1VAPM-3	Using *groov* products in your network and over the internet	2161

OptoForums

OptoForums focused on *groov* products and their tools are available 24 hours a day, 7 days a week, so you can get advice from experienced *groov* product users:

- *groov* EPIC Forum
- *groov* RIO Forum
- Node-RED Forum
- Ignition Edge Forum

Product Support

If you can't find the help you need in this guide or in the product user's guides, contact Opto 22 Product Support. Product Support is free.

Phone: 800-TEK-OPTO
(800-835-6786 toll-free in the U.S. and Canada)
951-695-3080
Monday through Friday,
7 a.m. to 5 p.m. Pacific Time

Email: support@opto22.com

Opto 22 website: www.opto22.com

GENERAL PREREQUISITES

This guide is best for people who have some background knowledge of MQTT and want to try it out on their Opto 22 hardware. In addition, the following are required to make use of the examples in this guide:

- Administrator access to a *groov* RIO or *groov* EPIC device (If you don't have access to one, contact Opto 22 for a demo. For steps to log in or complete initial setup, see the user's guide for your *groov* product.)
- The basic information to connect to an MQTT broker, such as:
  - Broker URL and port number
  - Username
  - Password

If you don't have access to a broker or aren't sure which one is right for you, take a look at Appendix A: MQTT Brokers and "Setting up an MQTT broker (server)" on page 4.

A word on terminology: Although MQTT's publish-subscribe architecture is different from the typical client-server architecture, the terms *client* and *server* are still used to describe the relationship between devices

or software communicating via MQTT. The MQTT broker functions as a server. The devices or software that publish or subscribe to data function as clients. *groov* Manage, Node-RED, and Ignition Edge can all act as MQTT clients running on a *groov* EPIC or *groov* RIO.

Comparing *groov* MQTT clients

MQTT and *groov* devices provide basic network and data protections like device-originating connections, authentication, encryption, and certification for all MQTT clients available on the device. When determining which approach is right for a given application, consider items like network size, engineering effort, fault tolerance, and budget. See the table below for considerations.

You can use both strings and Sparkplug B payloads for all methods, although Node-RED requires some extra work for Sparkplug data. If you rely on high-quality data for operations, auditing, or historization, Sparkplug payloads give you the guarantee of state awareness. Sparkplug also unlocks additional fault tolerance features and makes it easier to work with larger data sets.

In general, both *groov* Manage and Node-RED let you get started quickly without adding licensing costs. Ignition Edge provides more integration and fault tolerance options, but it has some additional licensing costs.

	***groov* Manage**	**Node-RED**	**Ignition/Ignition Edge**
Platform	GRV-EPIC-PR1 GRV-EPIC-PR2 GRV-R7-MM1001-10 GRV-R7-MM2001-10 GRV-R7-I1VAPM-3	GRV-EPIC-PR1 GRV-EPIC-PR2 GRV-R7-MM1001-10 GRV-R7-MM2001-10 GRV-R&-I1VAPM-3	GRV-EPIC-PR1[1] GRV-EPIC-PR2[1] GRV-R7-MM2001-10
Tag count	< 1000	< 50	1000+
Payload	Strings, Sparkplug B	Strings, JSON, Sparkplug B	Strings, JSON, Sparkplug B
Data sources	*EPIC:* Local I/O, PID loops, I/O features, Scratch Pad values, and PAC Control[2] tags *RIO:* Local I/O, PID loops, I/O features, and Scratch Pad values	Local I/O, OptoMMP addresses, PAC Control[2], databases, web services, OPC, and more	PAC Control[2], databases, legacy PLCs, devices, or OPC[3]
Subscribe to other publishers' topics	No	Yes	Yes, with MQTT Engine module
Security	User authentication SSL/TLS encryption Certificate management	User authentication SSL/TLS encryption Certificate management	User authentication SSL/TLS encryption Certificate management
Historization	With strings, none. With Sparkplug, volatile storage (limited to disk space[4])	None	Non-volatile storage (optional)
Failover	Multiple brokers With Sparkplug, primary host designation	None	Multiple brokers Multiple failover groups Multiple clients Primary host designation
Additional Cost	$0	$0	$800[5] +$200[6] w/ MQTT Engine

1 For Ignition/Ignition Edge 8, choose GRV-EPIC-PR2 for additional memory.
2 *groov* EPIC only
3 Other data sources are possible through the use of additional Ignition modules.
4 GRV-EPIC-PR2 and GRV-R7-MM2001-10 have greater disk capacity than other models.
5 GROOV-LIC-EDGE current list pricing as of 6/10/22.
6 Special pricing when purchased with Ignition Edge license; all Ignition features are free to try with unrestricted functionality.

String vs. Sparkplug B payloads

In addition to providing multiple client options, *groov* devices give you the option of working with two different payload formats: strings and Sparkplug B. How do you know which is right for you?

For simple data collection and control, string payloads are easy to use and flexible, although they involve some engineering effort that limits scalability and interoperability. For example, you'll need to manage the topic structure for each device or application client and give instructions on how to interpret the data you are publishing, which can be difficult if the network is large or heterogeneous. Similarly, while MQTT includes features to monitor client connection status, they require manually configuring status messages for each client, increasing engineering overhead. This kind of administrative overhead is one reason we recommend keeping a low tag count when working with basic string payloads.

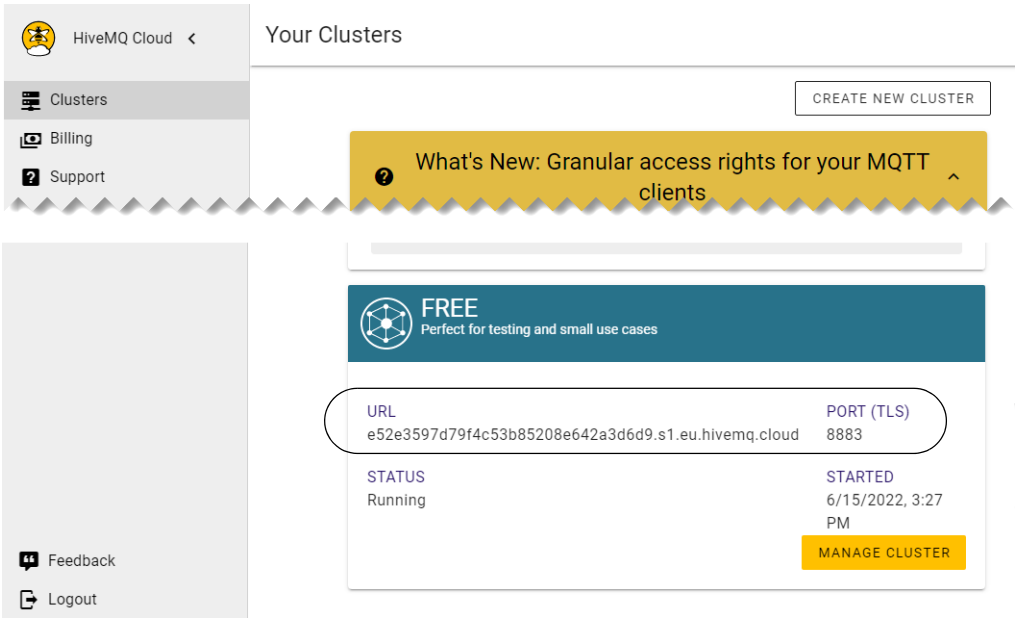
In contrast, Sparkplug payloads make it easier to bring MQTT publishers and subscribers online, and to work with devices from different vendors, so we recommend it if you're looking to grow and manage a large network. And if state-awareness is critical for your application, you'll definitely want to look at Sparkplug, because it eliminates the administrative overhead that MQTT normally requires to make these features work for you.

A general consideration when working with Sparkplug B, however, is the need for compatible clients. Since MQTT is data-agnostic, most brokers can handle Sparkplug B traffic alongside string payloads (though at some point broker performance limits may be reached). However, it's not just the broker: any device or application clients that want to use your Sparkplug data must also support Sparkplug.

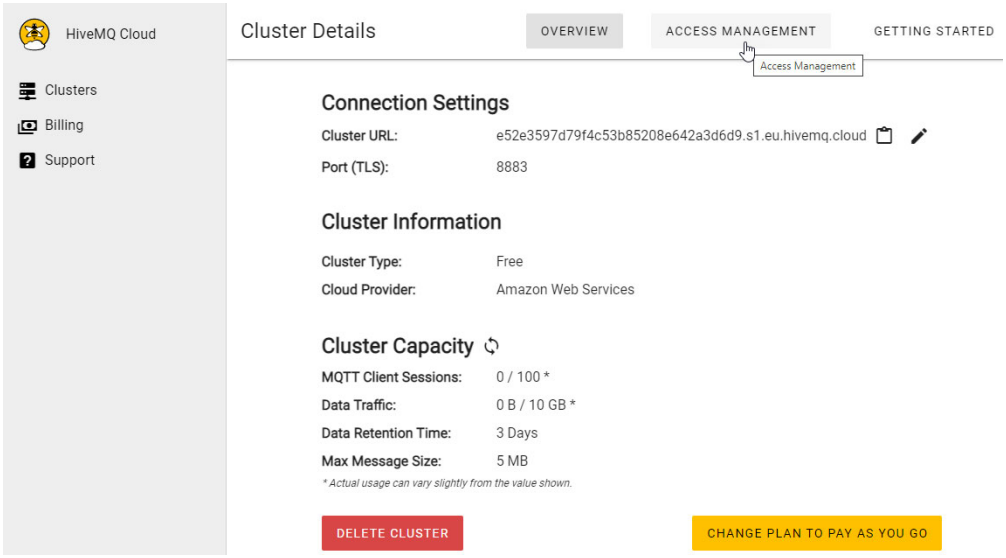
SETTING UP AN MQTT BROKER (SERVER)

If you already have an on-premises or cloud-based MQTT server, you can skip this section. If you don't have one, you can set up a free HiveMQ MQTT cloud broker using these steps.

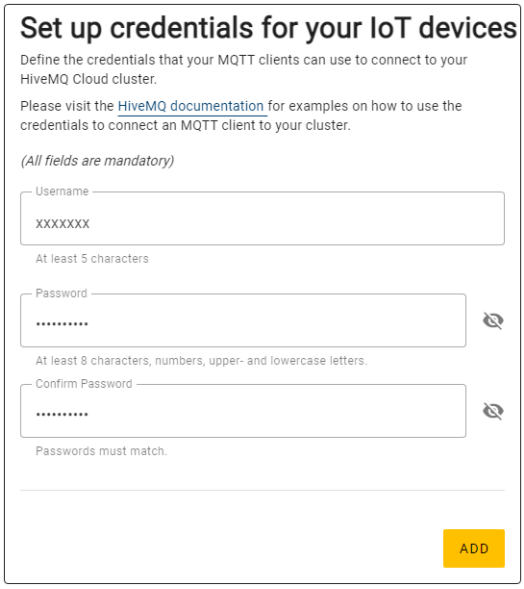
1. Go to hivemq.com/mqtt-cloud-broker. Click Sign up now.
2. Enter a username and password and remember them (you'll need them later). Follow instructions to create an account.
3. When the HiveMQ console appears, scroll down to the FREE section. Notice that the broker is running.



4. Click Manage Cluster to see details about your MQTT cloud broker.



5. Click the Access Management tab at top.



6. Enter a username and password that your MQTT clients will use to connect to this cloud broker. Remember these credentials! Click Add.
7. Click the Overview tab to return to the details page. Keep this page handy so you can copy the URL when you configure your MQTT broker/server in *groov* Manage, Node-RED, and Ignition Edge.

If you need help with HiveMQ, choose Support > HiveMQCloud > Quick Start.

Testing the HiveMQ broker

For testing and debugging communications with your HiveMQ Cloud Edition broker, you can use a quick desktop tool called mqtt.fx. It is a free download and available for Windows and MacOS.

ENABLING PUBLIC TAG ACCESS

In order for the MQTT clients on *groov* EPIC or *groov* RIO to publish your I/O and PAC Control tag data, you need to expose that data publicly to allow it to be read by other services.

Exception: If you are using Node-RED with groov I/O nodes only (`node-red-contrib-groov-io`*), you do not need to enable public tag access.*

Topic paths and wildcards

Once you have exposed the data you want to have publicly available, subscribers (but not publishers) can include a wildcard in MQTT topic paths to shape the data the subscriber receives. Wildcards work the same in strings and Sparkplug. The two available wildcards are the multilevel number sign (#) and the single-level plus sign (+).

Suppose these topic paths are available from publishers, with data at each level:

```
opto22/manufacturing/line1
opto22/manufacturing/line2
opto22/shipping/line1
opto22/shipping/line2
```

Here's how a subscriber might get the data it needs:

- To subscribe to data in the topic `opto22/manufacturing/line2`, a subscriber would use that exact topic path.
- To subscribe to data at any number of levels within a topic, a subscriber can use the multilevel wildcard #. For example, the path `opto22/manufacturing/#` subscribes to opto22/manufacturing, opto22/manufacturing/line1, and opto22/manufacturing/line2. The wildcard can only be used as the last character in the topic path, and it must be the only character at that level.
- To subscribe to data at one topic level, a subscriber can use the single-level wildcard +. For example, the path `opto22/+` subscribes to opto22/manufacturing and opto22/shipping but not to opto22/, opto22/manufacturing/line1, opto22/shipping/line1, and so on. This wildcard can be used in the middle of the topic path, too. For example, `opto22/+/line1` subscribes to opto22/manufacturing/line1 and opto22/shipping/line1.

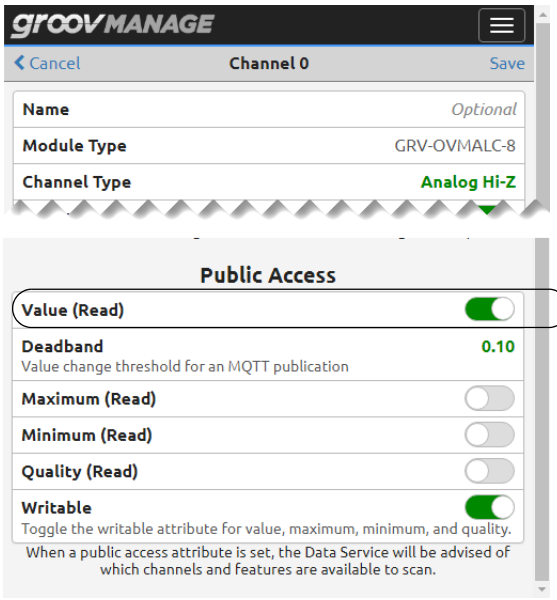
Keep in mind:

- A subscriber can include a wildcard in the topic path; a publisher's topic path cannot include a wildcard.
- If a topic allows writes, subscribers cannot write using a wildcard; the topic path must include a specific output tag to write to.

Exposing I/O data in *groov* Manage

Enabling public access in *groov* Manage lets I/O data be read by the *groov* Manage MQTT service (when the Device Type is set to OptoMMP) and by Node-RED nodes or other applications capable of using OptoMMP.

1. *groov* EPIC: From the Home page, navigate to I/O > Module # > Channel # > Configure.
   groov RIO: From the Home page, navigate to I/O Channels > Channel # > Configure.
2. Under Public Access, enable Value (Read) for each I/O point that you want to publish.

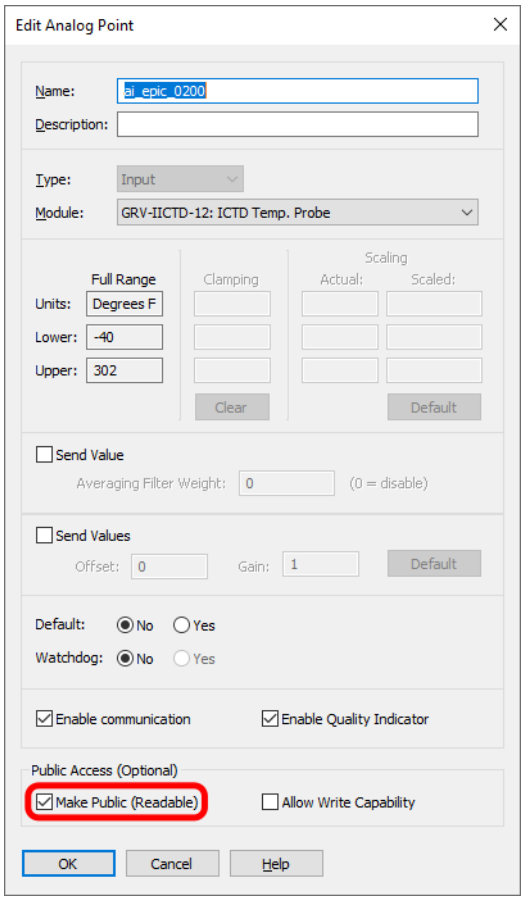


Exposing I/O and tag data in PAC Control (*groov* EPIC only)

If your groov EPIC runs a PAC Control strategy, use PAC Control to expose I/O or variable tags from your strategy in these ways:

- To Ignition Edge
- To the *groov* Manage MQTT service (when the Device Type is set to Controller)
- By the PAC Control nodes (`node-red-contrib-pac`) in Node-RED

To enable access, in PAC Control, open the edit dialog box of each I/O point or variable to be published, and check Make Public (Readable). See example below. Then save and run your control strategy.



CONFIGURING SIMPLE DATA COLLECTION AND CONTROL

Let's say you have a small sensor network or equipment group with a low number of I/O. How do you quickly take that data and publish it from your *groov* device using MQTT? As usual, we give you options. Here are two: *groov* Manage with strings (below) and Node-RED (see page 11).

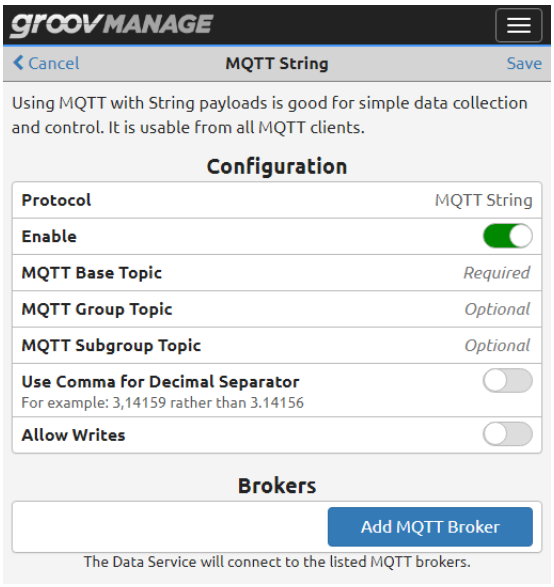
NOTE: To publish tags from a PAC Control strategy running on your groov EPIC, you'll need to be familiar with I/O and tag configuration and how to create and run a control strategy. If you need help, check out our online training for PAC Control.

Option 1: *groov* Manage with string payloads

The option for plain text (string) payloads is available in *groov* RIO and in *groov* EPIC firmware release 1.4.2 and higher. It's the simplest way to get started with MQTT. You can do it in just a few steps.

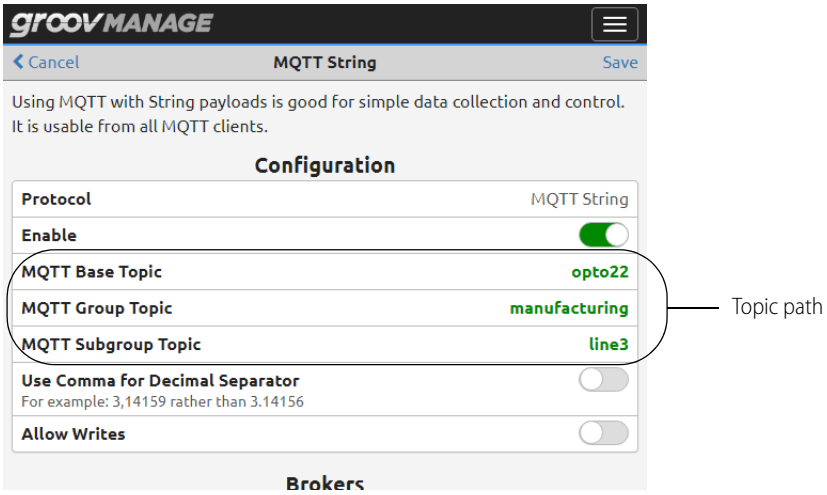
You must already have an MQTT broker (see "Setting up an MQTT broker (server)" on page 4) and have made at least one data point public (for *groov* RIO, see "Exposing I/O data in groov Manage" on page 6; for PAC Control running on a *groov* EPIC, see "Exposing I/O and tag data in PAC Control (groov EPIC only)" on page 7).

1. Log into your *groov* EPIC or *groov* RIO and choose Data Service.
2. In the MQTT section, choose Add MQTT string.

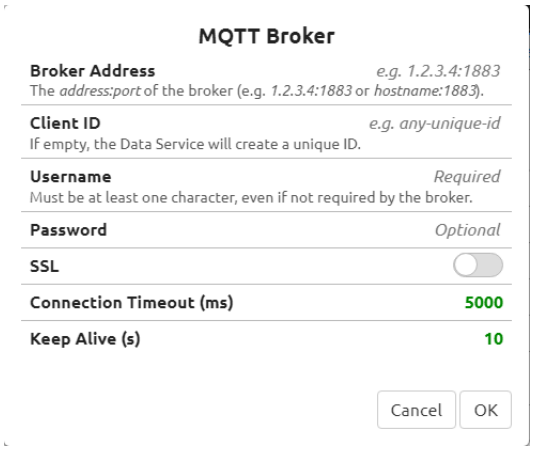


3. Enter an appropriate topic path (Base Topic, and optionally Group Topic and Subgroup Topic).
   For MQTT with strings, the topic path typically reflects location, site, and work cell, to describe where the data is coming from. For example:

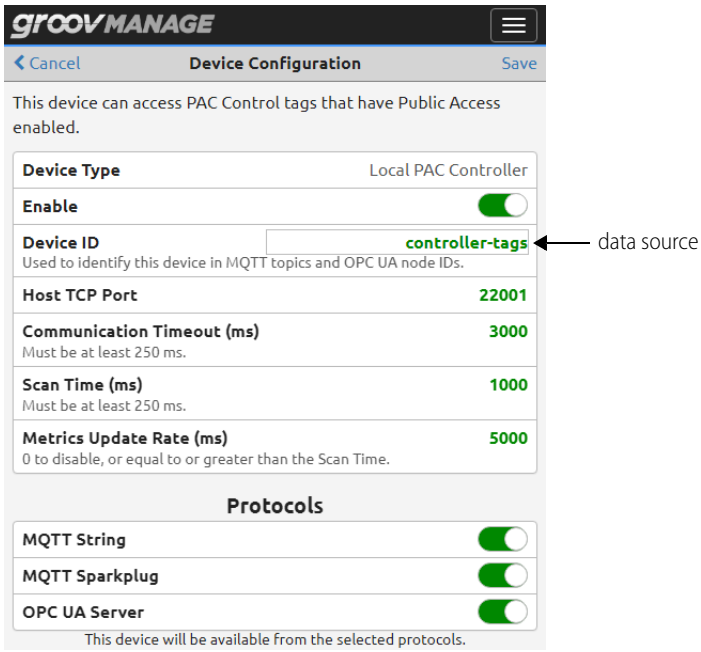
Topic path element	Example
Base Topic	`opto22`
Group Topic	`manufacturing`
Subgroup Topic	`line3`



4. Click Add MQTT Broker.



5. Enter the broker's URL and security details (if needed; see "Security in *groov Manage*" on page 22) to point the client to your MQTT broker. Click OK.
6. Back in the MQTT String window, click Save (upper right).
7. In the Data Service window, scroll down to Scanned Devices.
   - For a *groov* RIO, choose Add Local I/O System.
   - For a *groov* EPIC, choose one:
     - To publish tags from your PAC Control strategy, choose Add Local PAC Controller (used in this example).
     - To publish I/O data directly from *groov* Manage, choose Add Local I/O System.
8. In the Device ID field, enter the data source, for example `controller-tags` or `local-io`.



Notice the Protocols section. By default, all protocols (MQTT String, MQTT Sparkplug, and OPC UA Server) are enabled. You can disable the ones you're not using or leave them enabled for the future.

9. Click Save (upper right).

Voilà! Your broker will pick up your tags as new MQTT topics, and from there, your back-end applications can become subscribers using one of these paths:

```
<Base Topic>/<Group Topic>/<Subgroup Topic>/<Device Topic>/<Tag>
<Base Topic>/<Group Topic>/<Subgroup Topic>/<Device Topic>/#
```

Remember that subscribers can specify an individual tag to subscribe to (first path above), or use the # wildcard at the end to receive all public data from that point in the path (second path).

In our example, the path might look like this:

```
opto22/manufacturing/line3/controller-tags/#
```

Subscribers that use the wildcard would receive all public tags available from manufacturing/line3/controller-tags.

NOTE: If you choose to allow writes to this topic, your subscribers cannot use a wildcard. Writes must include a specific output tag to write to:

```
<Base Topic>/write/<Group Topic>/<Subgroup Topic>/<Device Topic>/<Output Tag>
```

In our example, the write path might be:

```
opto22/write/manufacturing/line3/controller-tags/stack_light_red
```

Other data sources. This example showed configuration for PAC Control. When you're in the Data Service window, scroll down to the Public Access section and notice that from there, you can configure other I/O channels, PID loops, and Scratch Pad areas of the memory map for use with MQTT.

For more information on using *groov* Manage for MQTT, see the user's guide for your *groov* device.

Option 2: Node-RED

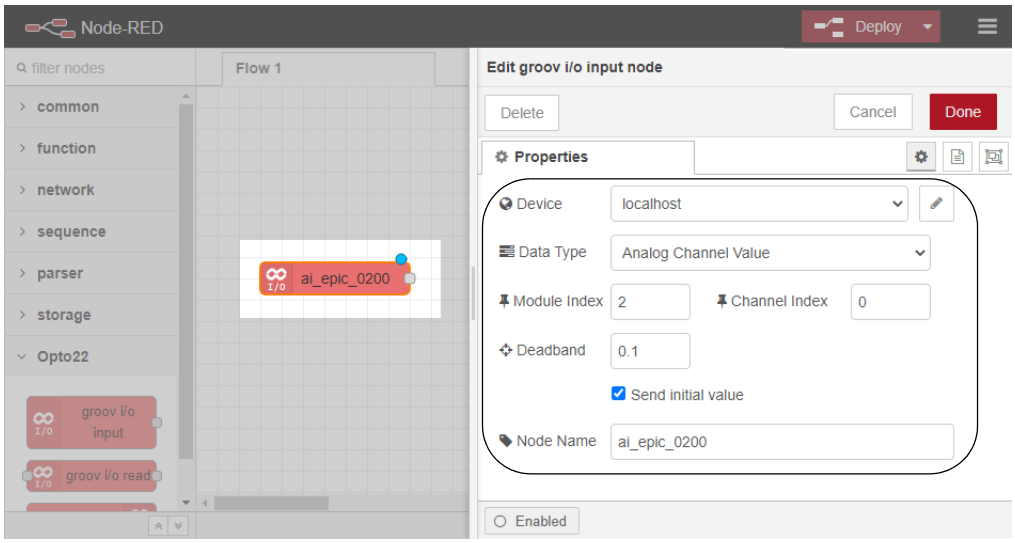
While *groov* Manage is great for straightforward tag publishing, for more flexibility and granularity, consider Node-RED. It has several advantages for basic publishing.

First, Node-RED makes it easy to combine tags with data from other sources, like web services. Node-RED can also subscribe to topics published by other MQTT clients, where *groov* Manage can only detect updates communicated on its own topics. In addition, Node-RED lets you publish JSON-formatted data, commonly used with cloud IoT platforms and other services. Finally, Node-RED lets you configure quality of service (QoS) levels and last will and testament (LWT) messages for each topic, if you want to take advantage of these features.

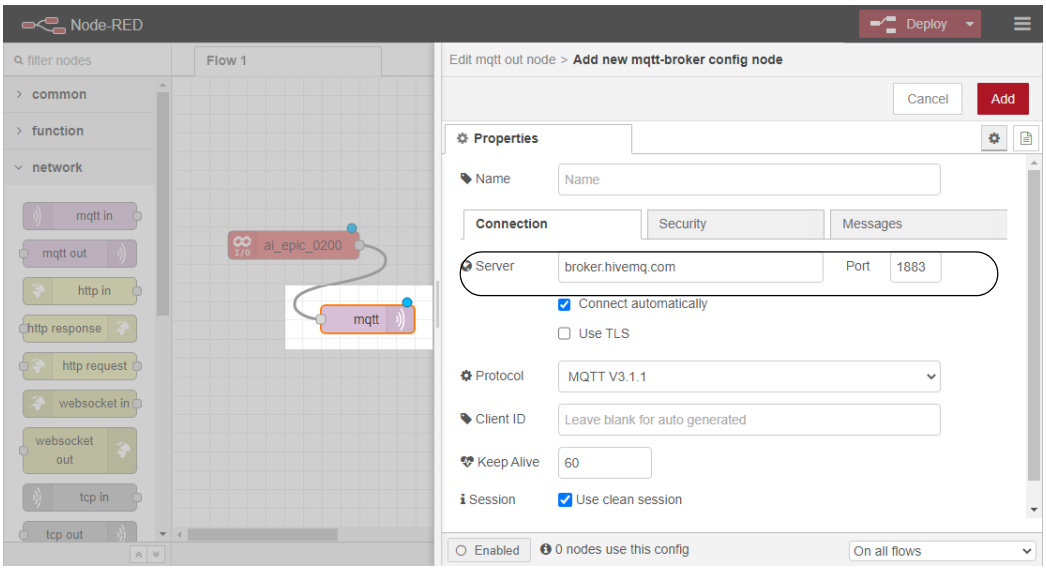
Prerequisite: You should be familiar with the basics of the Node-RED environment and how to build, configure, and deploy flows. If not, check out our online training first or review the Node-RED information in the user's guide for your *groov* device.

Here is how you would set up basic publishing in Node-RED:

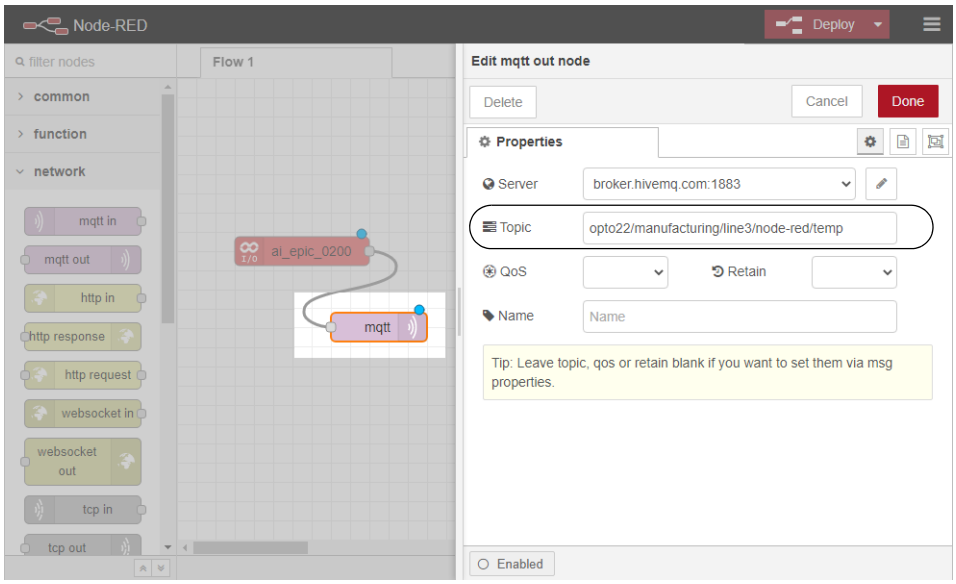
1. Log into your *groov* EPIC or *groov* RIO, navigate to the Node-RED page, and open the Node-RED editor.
2. Install the *groov* I/O nodes (`node-red-contrib-groov-io`) or your data source of choice, add a *groov* I/O input node to the workspace, and configure it to grab the I/O data you want to publish. (For help with configuration, see the *groov* I/O Nodes section on developer.opto22.com.)



3. Add an MQTT Out node to the workspace and connect it to your data node.
4. Double-click the MQTT Out node, click the pencil next to Add new mqtt-broker, and enter your broker URL and port. Click Add when finished.



5. Back in the MQTT node configuration panel, enter your desired topic path and click Done.
   Be careful typing the path, as it is not generated. Some guidelines:
   - Slashes go between path elements; don't start the path with a slash.
   - Use ASCII characters only, and don't include spaces.
   - Remember paths are case-sensitive.



6. Click Deploy to start publishing on the configured topic path.

The process for creating a subscription flow using the MQTT In node is very similar. Take a look at our video on Node-RED and MQTT for details.

For more information on Node-RED, see your *groov* device's user's guide.

CONFIGURING MQTT FOR HIGH-RELIABILITY SCADA/IIOT

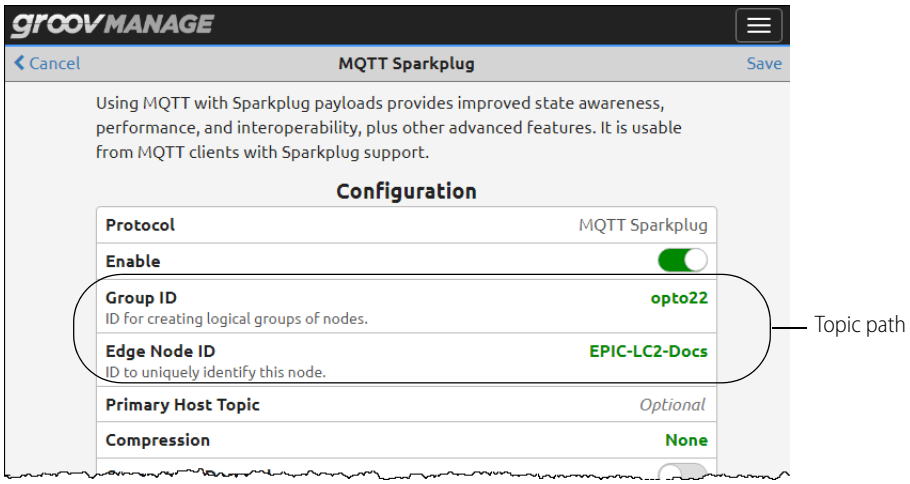
Do you need to tap into the scalability and fault tolerance features that Sparkplug B offers? If so, follow the steps in this section for the most direct Sparkplug B setup path for each *groov* MQTT client: *groov* Manage, Ignition Edge, and Node-RED.

To increase reliability and tap into the advanced features of Sparkplug B, also explore Chapter 2: Security and Fault Tolerance.

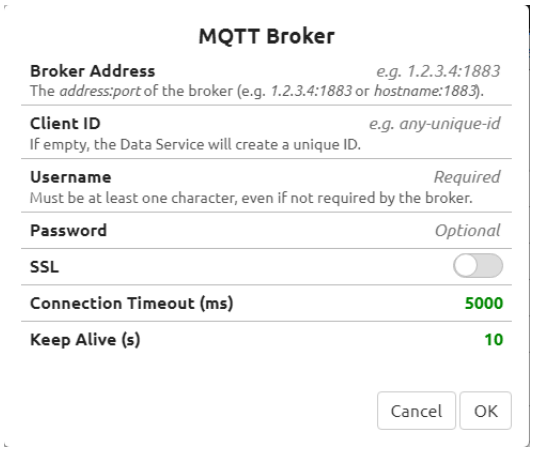
Option 1: *groov* Manage with Sparkplug B payloads

For Sparkplug B publishing on *groov* EPIC or *groov* RIO, *groov* Manage is the quickest way to get started. *groov* Manage supports some of the same fault tolerance features as Ignition Edge.

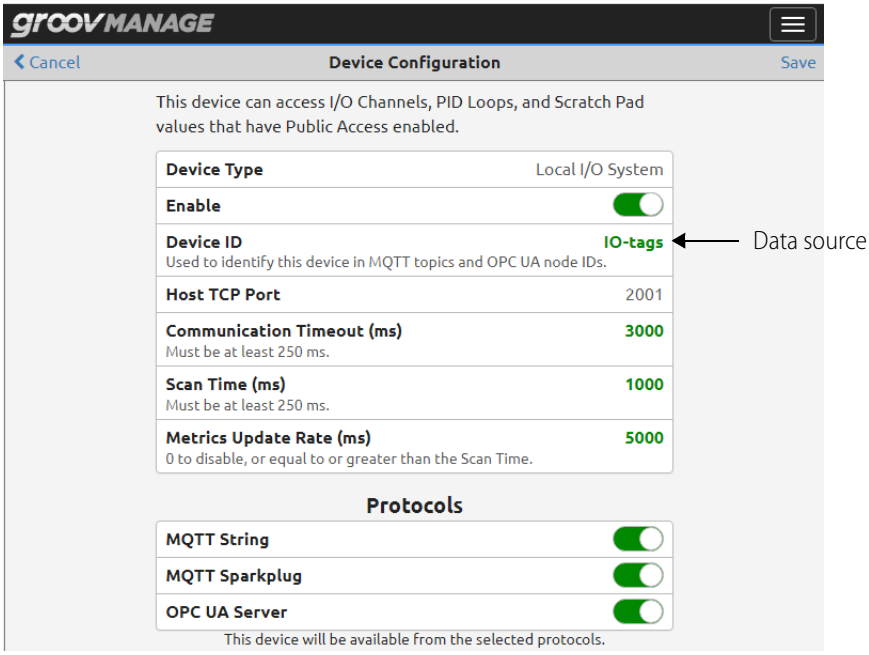
1. Follow the steps in "Option 1: groov Manage with string payloads" on page 8 with two exceptions:
   - In the Data Service window, begin by choosing Add MQTT Sparkplug.
   - Enter the elements of your topic path in the Group and Edge Node ID fields. For MQTT with Sparkplug B, these first two topics can reflect the location and then the EPIC or RIO's hostname.



2. Scroll down and click Add MQTT Broker.



3. Enter the broker's URL and security details (if needed; see "Security in *groov Manage*" on page 22) to point the client to your MQTT broker. Click OK.
4. Back in the MQTT Sparkplug window, click Save (upper right).
5. In the Data Service window, scroll down to Scanned Devices.
   - For a *groov* RIO, choose Add Local I/O System.
   - For a *groov* EPIC, choose one:
     - To publish tags from your PAC Control strategy, choose Add Local PAC Controller.
     - To publish I/O data directly from *groov* Manage, choose Add Local I/O System (used in this example).
6. In the Device ID field, enter the data source, for example controller-tags for PAC Control or IO-tags for Local I/O System.



groov Manage will generate a topic path based on the ID fields in combination with others required by Sparkplug B. Subscribers can use a path with a specific tag, or use a wildcard (#) to subscribe to all public tags at that point on the path:

```
spBv1.0/<Group ID>/+/<Edge Node ID>/<Device ID>/<Tag>
spBv1.0/<Group ID>/+/<Edge Node ID>/<Device ID>/#
```

In this example, MQTT subscribers might use the following topic path:

```
spBv1.0/opto22/+/EPIC-LC2-Docs/IO-tags/#
```

There is much more to the Sparkplug implementation than just subscribing to MQTT topics. Without birth certificates and implementing statefulness, a subscriber won't know tag states that haven't been published or tag properties such as units and scaling. More details are available in the user's guide for your *groov* device.

Option 2: Ignition Edge (*groov* EPICs and GRV-R7-MM2001-10 only)

Ignition Edge is available on *groov* EPIC processors and on the *groov* RIO MM2. Setting it up involves more steps than *groov* Manage. But for a reasonable licensing fee, the platform offers higher performance, a wide range of integration options, and the most complete set of fault tolerance options of any of the *groov* MQTT clients.

In addition to EPIC and RIO MM2 data, if you need to connect to and publish data from legacy systems or PLCs like Allen-Bradley® or Siemens®, or any Modbus®/TCP compatible device, you can follow the same process to add those devices, using Ignition Edge on EPIC as the gateway to your MQTT network. It's one of the primary advantages of the Ignition Edge MQTT client.

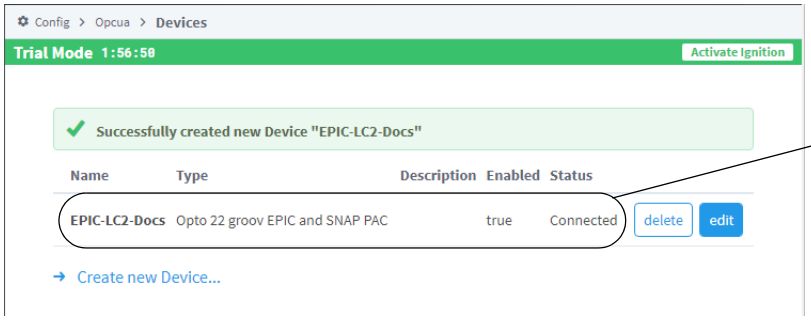
You don't need a license to set up Ignition Edge initially. You can follow all the steps below and see how MQTT works on your *groov* device on a trial basis. The trial lasts for two hours and can be repeated as often as you wish.

Ignition Edge Example

This example uses data from a *groov* EPIC running a PAC Control strategy. To follow these steps, you must have a strategy running in your *groov* EPIC and have already made at least one I/O channel or variable public; see "Exposing I/O and tag data in PAC Control (groov EPIC only)" on page 7. If you follow these steps for

another data source, make sure you have made that data public. For example, if a *groov* RIO MM2 is your data source, see "Exposing I/O data in groov Manage" on page 6.

1. Log into your *groov* device. From Home, choose Ignition, and enable Ignition Edge. Click Save.
2. When Status shows Ignition Edge running, click Open Ignition Edge.
3. Accept the agreement and create your administrator account. Remember these user credentials.
4. In the Configure Ports window, leave the ports at their default values. Click Finish Setup.
5. Click Start Gateway and wait while it starts.
6. When Ignition Edge opens, in the left navigation column, click Config. Log in using the username and password you just created.
7. Install quarantined modules for *groov* and MQTT Transmission:
   a. In the left-hand navigation under System, choose Modules, and scroll down until you see the Opto 22 *groov* EPIC and SNAP PAC Driver module. Click Install.
   b. Confirm that you want to install the module, review the Module License Agreement, check "I accept the terms in the License Agreement," and click Accept License.
   c. Click "I want to add this certificate..." and click Add Certificate and Install Module.
   d. Repeat these steps to install the MQTT Transmission Module.
8. Configure your *groov* EPIC connection:
   a. In the left navigation column, under OPC UA, choose Device Connections. Delete the sample connection shown.
   b. Click Create new Device. Scroll down and select Opto 22 *groov* EPIC and SNAP PAC. Scroll down further and click Next.
   c. Enter a Name, and for Address use `localhost`. Click Create New Device. Wait a few seconds and notice that the device's status is Connected. (if it's not connected, make sure your strategy is running and at least one I/O channel or variable is public.)



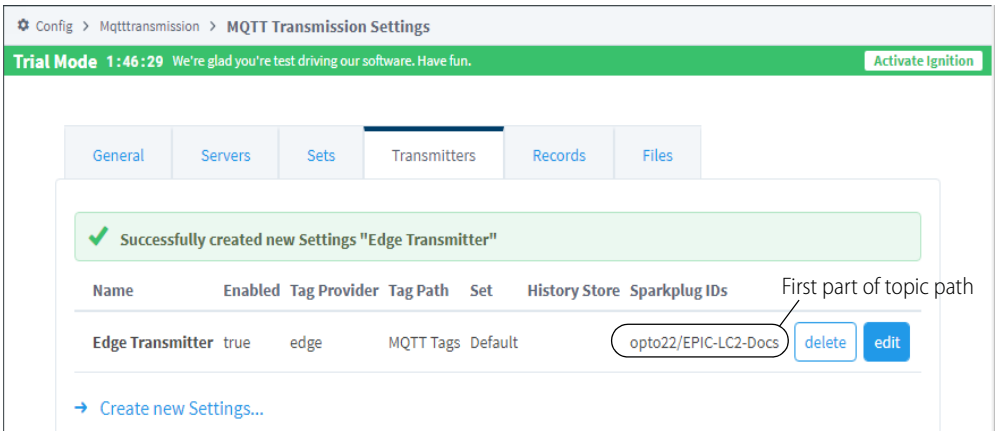
9. Configure the first part of your topic path:
   a. Scroll down in the left navigation column and under MQTT Transmission, choose Settings.
   b. Click the Transmitters tab. Delete the sample transmitter shown.
   c. Click Create new Settings. Enter the following:

      Name: `Edge Transmitter`
      Tag Provider: `edge`
      Tag Path: `MQTT Tags`

   d. Scroll down to Sparkplug Settings and enter the following (if needed, press Tab between each one). Notice that the Edge Node ID is the name of the EPIC device you just configured.

      Group ID: `opto22`
      Edge Node ID: `EPIC-LC2-Docs`
      Device ID: [leave blank]

   e. Click Create New Settings.



10. Add a broker (server) connection to the MQTT Transmission module.

a. Click the Servers tab. Delete the sample server shown.

b. Click Create new MQTT Server. Enter the Name and URL of the broker you are using. Make sure you enter the URL with the correct prefix and port number.

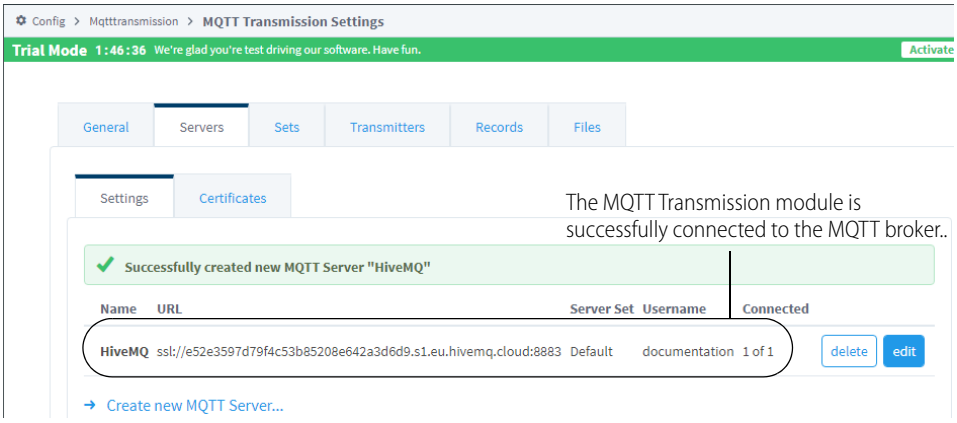
Example: For a secure connection such as the one created in "Setting up an MQTT broker (server)" on page 4, the URL format would be `ssl://<server address>:8883`

c. Enter the username and password you configured for clients when you set up the broker.

For example, if you are using the HiveMQ cloud broker (see page 4), enter the username and password you set up in the Access Management tab.

d. Scroll down and click Create New MQTT Server.

In a few seconds the broker connects to your *groov* device.



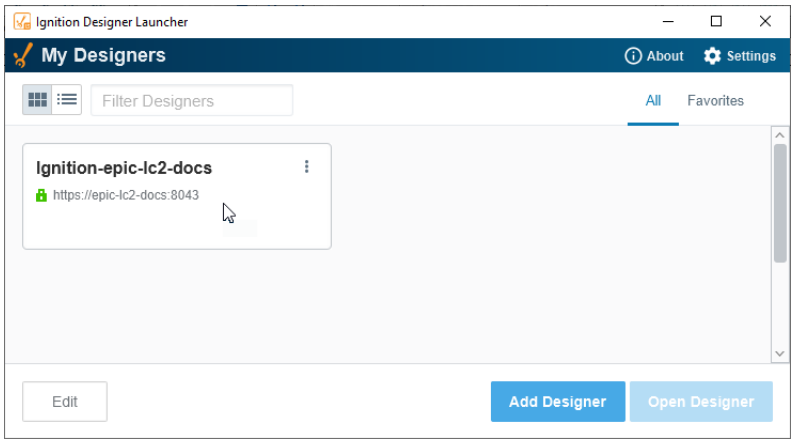
11. Launch Ignition Designer:

a. In the upper right-hand corner of Ignition Edge, click Get Designer. Then cick Download. When the download is complete, open the file and follow setup instructions.

b. When Ignition Designer Launcher opens, click Add Designer. If your *groov* device is in the list that appears, highlight it and click Add Designer. If it doesn't appear, click the Manual tab and enter your device's URL in the format: `https://<hostname>:8043`

c. Follow any instructions to import and trust the certificate on your device.

Your device appears as a Designer:

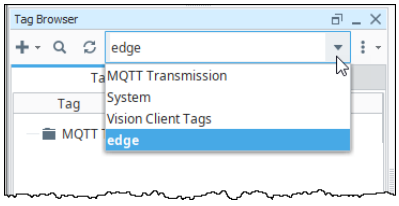


d. Double-click your *groov* device and wait while Ignition Designer is loaded to your computer.

e. Enter your Ignition username and password.

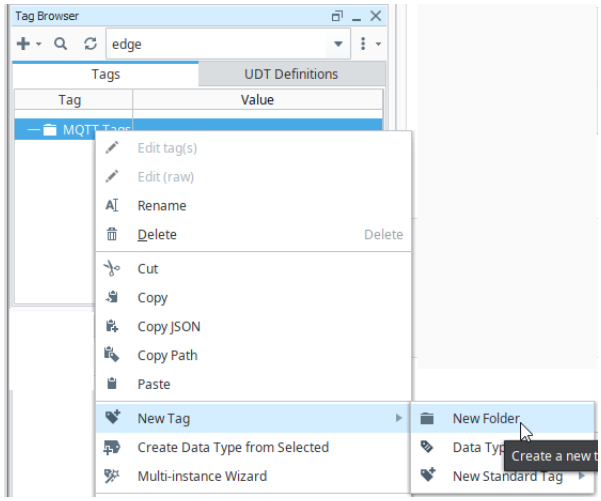
12. When Designer opens, finish the topic path and publish your tags:

a. In the Tag Browser pane (center left), choose `edge` from the dropdown as the tag provider.



b. If a PLC-1 folder appears, right-click it and choose Delete.

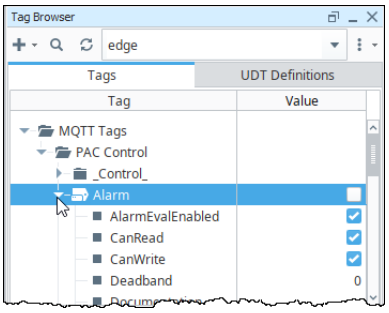
c. Right-click the MQTT Tags folder (the tag path) and choose New Tag > New Folder.



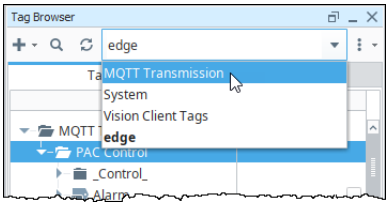
d. Name the folder PAC Control (the device ID—the data source) and click OK. Expand the MQTT Tags folder to show the new folder.

e. Choose View > Panels > OPC Browser. In the panel, expand Devices and your device's folders to see all the public tags. Highlight the tags you want and drag them to the new PAC Control folder.

f. Expand the PAC Control folder. Notice that within a few seconds, values start appearing for the tags. To see tag configuration details, expand the tag.



g. To publish your tags, click the dropdown and choose MQTT Transmission.



h. Expand Transmission Control, click Refresh, and check the box next to it. If a dialog box appears, choose Enable Read/Write Mode.

The checkbox is checked and then cleared when the refresh is completed. Depending on the number of tags, it may be cleared immediately or take several seconds.

13. Repeat step 12 any time you modify the folder structure or add a new tag.

14. Once you have finished adding tags, return to the Ignition page in *groov* Manage and click Save.

This setup produces an MQTT/Sparkplug B topic path according to the folder structure you have created. Subscribers can use a path with a specific tag, or use a wildcard (#) to subscribe to all public tags at that point on the path:

```
spBv1.0/<Group ID>/+/<Edge Node ID>/<Device ID>/<Tag>
spBv1.0/<Group ID>/+/<Edge Node ID>/<Device ID>/#
```

In this example, MQTT subscribers might use the following topic path:

```
spBv1.0/opto22/+/EPIC-LC2-Docs/PAC_Control/#
```

Remember that the advantage of the + wildcard here is that it subscribes to all messages, not just tag data. See "Topic paths and wildcards" on page 6.

Out of the box, you'll have unrestricted access to all of the features of Ignition Edge on a two-hour timer. At the end of two hours, you can reset the timer using the button in Ignition Gateway at the top of the page. For uninterrupted use, you can purchase an Ignition Edge license (GROOV-LIC-EDGE for Ignition Edge version 7 or GROOV-LIC-EDGE8 for Ignition Edge version 8).

See your *groov* device's user's guide for more information on using Sparkplug B with Ignition Edge.

Option 3: Node-RED

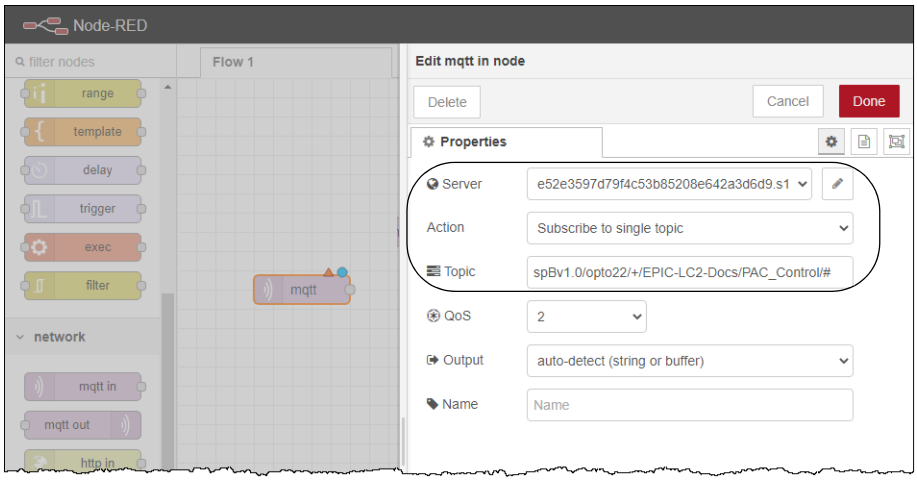
If you don't need your *groov* device to *publish* Sparkplug B data, but only need to *subscribe* to that data, it's possible to read in and decode Sparkplug messages directly in a data flow. Besides being free to use, Node-RED has the advantage of easily mashing up data from other sources and even creating dashboards or a light UI as well.

Using this method, Node-RED does not become a Sparkplug-compliant device; it is just a data consumer subscribed to an MQTT broker. But sometimes that's all you need.

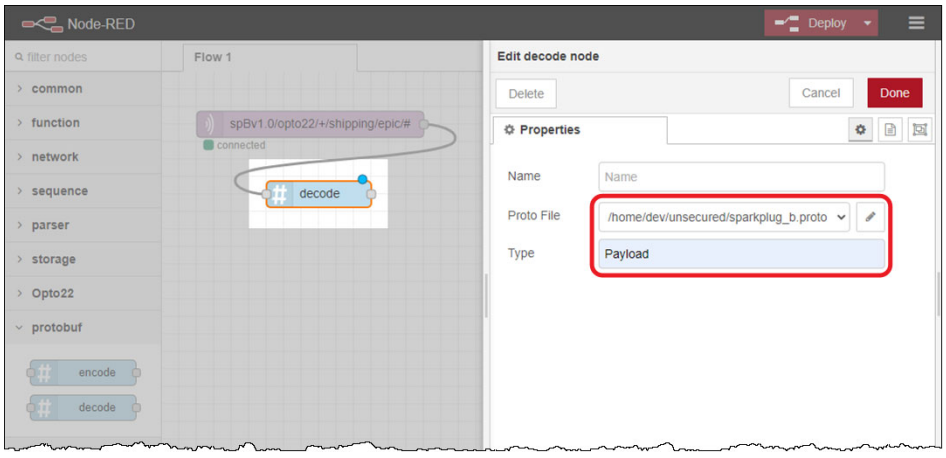
Here's the procedure:

1. Log into your *groov* EPIC or *groov* RIO, navigate to the Node-RED page, and open the Node-RED editor.
2. Place an MQTT In node. Double-click it and configure the broker connection, security details (if needed; see "Security in Node-RED" on page 23), and topic path. The path should use this format:

   ```
   spBv1.0/<Group ID>/+/<Edge Node ID>/<Device ID>/#
   ```



3. Download the sparkplug_b.proto file from the Eclipse Tahu Project on GitHub, and place it in the unsecured file section of your *groov* device.
4. Install the protocol buffer nodes (`node-red-contrib-protobuf`), add the Decode node to the workspace, and connect it to the MQTT In node.
5. Configure the Decode node by setting Proto File to the .proto file path and setting Type to Payload.



6. Connect the output of the Decode node to wherever you want to send the subscribed data, and click Deploy to start receiving data.

Publishing Sparkplug payloads with Node-RED is also possible but is much more involved. Check out this guide published by Cirrus Link if you are seriously considering that option:
https://docs.chariot.io/display/CLD79/B%3A+Example+Node-RED+Client

2: Security and Fault Tolerance

ENABLING MQTT SECURITY

Cybersecurity is a top concern for industrial IoT, so *groov* devices are designed to help you create a secure foundation for MQTT communication.

All MQTT clients on *groov* EPIC and *groov* RIO allow you to authenticate broker connections, to apply certificates of trust, and to enable encrypted communications. Opto 22 strongly recommends that you set up authentication for access to your broker, apply security certificates, and use encryption. Some of the examples in this book won't work unless you use these features.

For all clients

The following sections give the specifics of each MQTT client related to enabling security, but for all clients you need to make sure the appropriate protocol and port designations are shown in your broker URL to indicate you are using an encrypted connection.

TCP port 1883 is registered with IANA for MQTT *unsecure* connections; port 8883 is registered for secure *TLS encrypted* connections. See the user's guide for your *groov* device for more information on security features.

Encryption and authentication are two key concepts for understanding security. *Encryption* encodes data so that only a device or software with the encryption key can decode it. *Authentication* verifies to the two communicating devices or software that the other is actually who it says it is. For a good discussion of how encryption works, see our blog post on Encryption and Certificates.

Secure communication between a *groov* MQTT client and an MQTT broker begins like this:

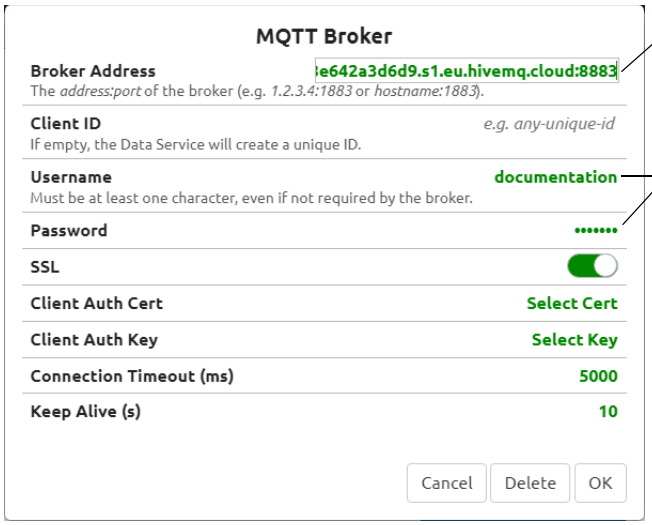
1. The *groov* client connects to the broker/server and requests a secure connection.
2. The server sends its encryption certificate information, which the *groov* device (or Ignition Edge, if it is the client) verifies in its Trust Store.

   The *groov* device's Trust Store (and Ignition Edge's Trust Store) already contain certificates for most cloud services, so they are easily verified. If you have set up your own MQTT broker, however, you may need to upload an encryption certificate for it to the Trust Store.
3. Once the server's encryption certificate is verified, the client and server complete the encrypted connection.
4. Authentication is next. The client now knows that the broker/server is valid, but it has to prove that it's a legitimate client. It does so in one of two ways: by providing a *username and password* that the server recognizes, or by providing a *client authorization certificate and client authorization key*.

   The broker determines which authentication method is used. HiveMQ, for example, uses a username and password. Some large cloud-based services like Amazon Web Services (AWS) and Microsoft Azure use the client authorization certificate and key. You need to know which authentication method your broker requires before you can configure the broker in your *groov* device.

Security in *groov* Manage

1. If you need to add an encryption certificate for a broker you have set up (this is rare; see "For all clients" on page 21), get it from the broker and then choose Home > Security > Certificate Trust Store. Click Add/Update and locate the certificate file.
2. If your broker requires a client authorization certificate and key, get them from the broker and then choose Home > Security > Certificate Trust Store. Click Add/Update and locate the certificate-key files.
3. When you add an MQTT broker, in addition to entering the correct broker URL, enable the SSL option and add the user credentials the broker requires:
   - If the broker requires a username and password, enter them (shown in the example below).
   - If the broker requires a client authorization certificate and key, enter a character in the Username field but leave the Password field blank. Click Select Cert and Select Key to choose the client authorization.



DO NOT enter ssl:// at the beginning of the broker's address. (The SSL toggle below takes care of this information.)
DO enter the port at the end.

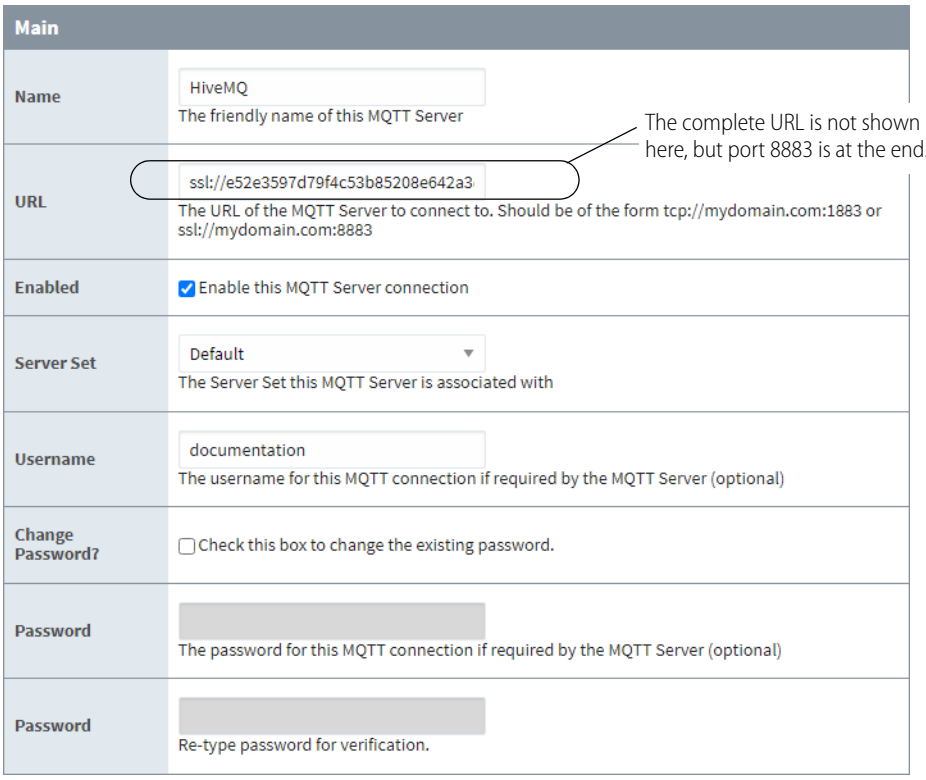
If the broker requires a username and password for authentication, enter the ones you set up in the broker. If the broker does not require a username and password, you must still enter at least one character in the Username field (but no password).

If the broker requires a client authentication certificate and key, click Select Cert and Select Key and locate those files.

4. When finished, click OK.

Security in Ignition Edge

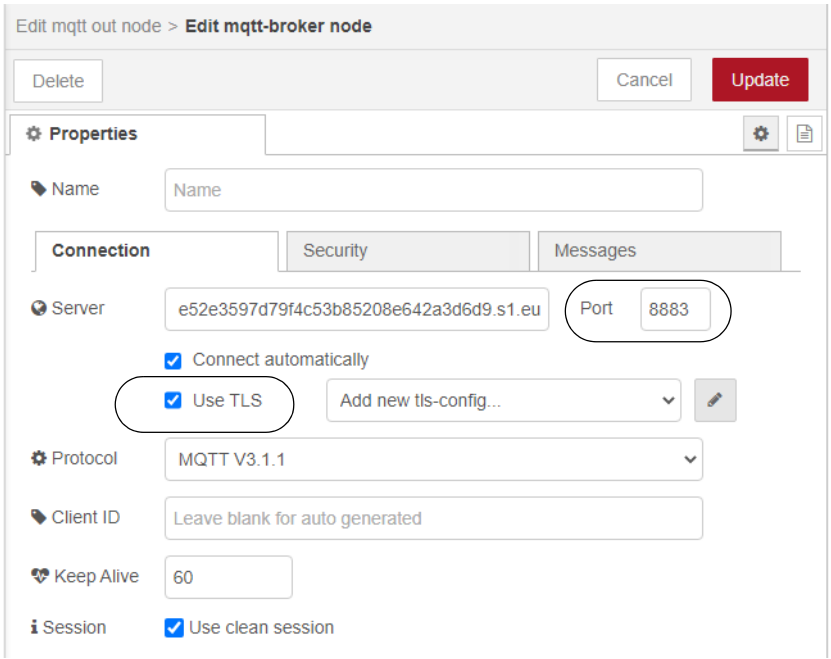
1. If you need to add an encryption certificate for a broker you have set up (this is rare; see "For all clients" on page 21), get it from the broker and then upload the server's public security certificate:
   a. In Ignition, choose MQTT Transmission > Settings.
   b. Click the Servers tab, then click the Certificates tab.
   c. Click Create new Certificate and enter the required information to upload the certificate.
2. If your broker requires a client authorization certificate and key, get them from the broker and then upload them to Ignition Edge:
   a. In Ignition, choose MQTT Transmission > Settings.
   b. Click the Servers tab, then click the Certificates tab.
   c. Click Create new Certificate and enter the required information to upload the certificate.
3. When you add a broker connection, specify the appropriate protocol and port in the URL. Then:
   a. If your broker requires a username and password, enter the username you set up in the broker.
   b. If your broker requires a client authorization certificate and key, do not enter a username. Instead, scroll down to the TLS section and choose the Client Certificate File and Client Private Key File from the dropdown list.



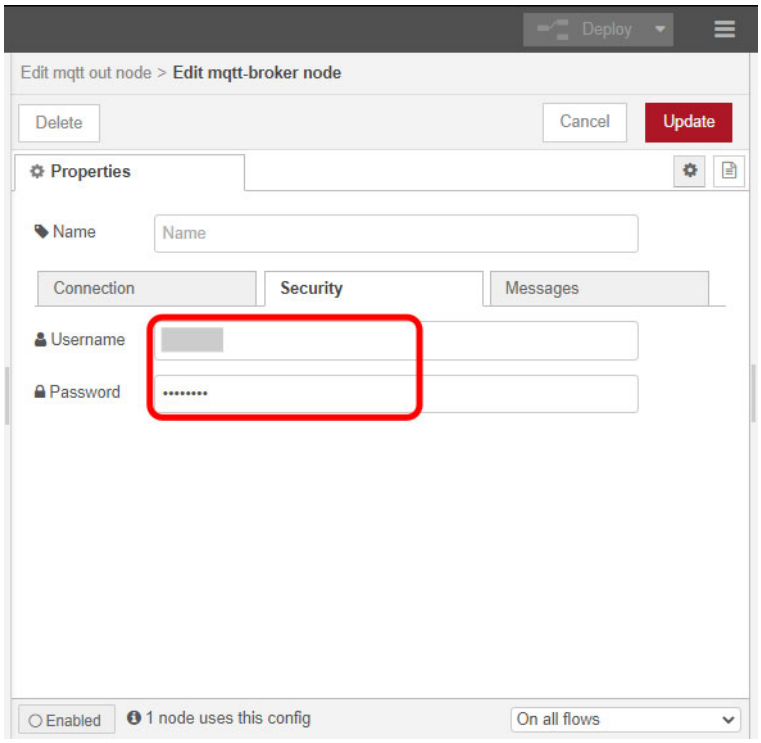
4. At the bottom of the page, click Save Changes.

Security in Node-RED

1. When editing the MQTT broker details, be sure to specify TLS port 8883. Then check the box to Use TLS.



2. If you need to add an encryption certificate for a broker you have set up (this is rare; see "For all clients" on page 21), get it from the broker. Click the pencil next to Add new tls-config and then, in the TLS configuration panel, upload it.
3. If your broker requires a client authorization certificate and key, click the pencil next to Add new tls-config and then, in the TLS configuration panel, upload them.
4. If your broker requires a username and password, click the Security tab. Enter the username and password you set up in the broker.



5. Click Update when done.

CONFIGURING ADDITIONAL FAULT TOLERANCE OPTIONS

groov EPIC and *groov* RIO allow you to take full advantage of MQTT's built-in reliability features. If you are using Sparkplug B, you will have even more options. This section explains how to configure these features in each MQTT client. Features include:

- Failover connections
- Primary host ID
- Store-and-forward history

Failover connections

A basic fault tolerance measure is to set up multiple MQTT brokers and configure your *groov* device with connections to each.

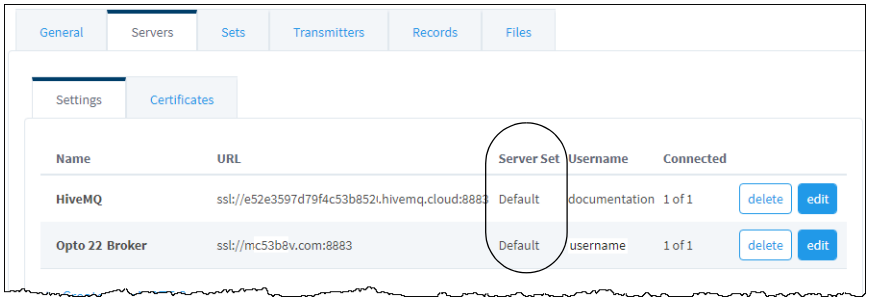
NOTE: By default, the MQTT client connects to the first available broker and switches round-robin if it loses connection. However, this behavior changes if you configure the Primary Host ID property (see page 25).

groov Manage

To set up failover connections, from the Home page choose Data Service > MQTT (string or Sparkplug) > Add MQTT Broker, and create as many connections as you want.

Ignition Edge

Add new server connections from the MQTT Transmission > Settings > Servers tab. Any broker connections that share the same Server Set property become part of an automatic failover group.



Ignition Edge allows you to create additional failover groups under the MQTT Transmission > Settings > Sets tab.

For advanced scenarios where you might group MQTT clients around specific brokers, you can create additional MQTT clients, each with its own server set and history store, by defining custom transmitter settings under MQTT Transmission > Settings > Transmitters > Custom > Create new Settings.

Node-RED

In Node-RED, broker connections are configured individually for each MQTT node. Only one connection per node can be set at a time, but you can dynamically change connections without redeploying by using the *connect* action and additional message properties with the broker details. See the bottom of the MQTT node help tab for instructions.

To add some redundancy to your project, you can create multiple, duplicate nodes with connections to different brokers.

Primary host ID

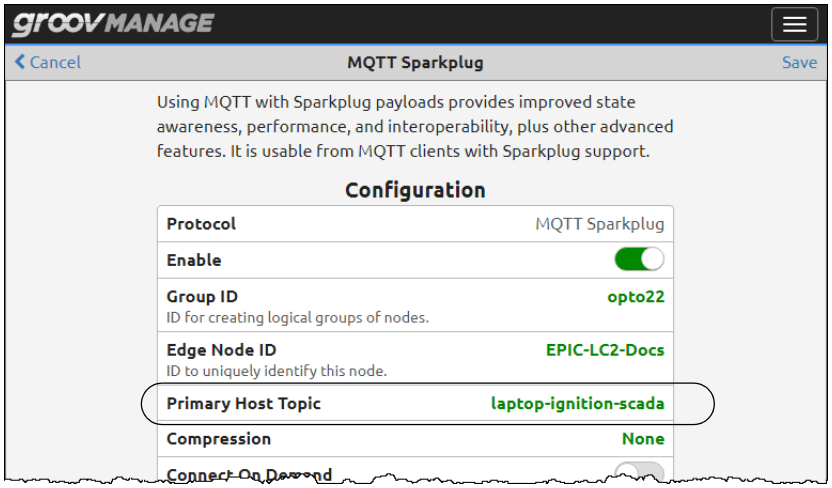
If you've configured multiple broker connections for failover (see page 24) in *groov* Manage or Ignition Edge, you can also set a Primary Host ID. This setting causes the MQTT client to give preference to broker connections that publish an ONLINE status for the specific client ID you enter.

Opto 22 strongly recommends using a Primary Host ID, because it's the only way an edge node knows if the Primary Host Application is online. If none of the available broker connections is publishing an ONLINE status for that ID, the MQTT client will continue to rotate through available brokers without staying connected.

In these examples, an Ignition SCADA server running on a local laptop (not Ignition Edge, but full Ignition), has a client ID of `laptop-ignition-scada`.

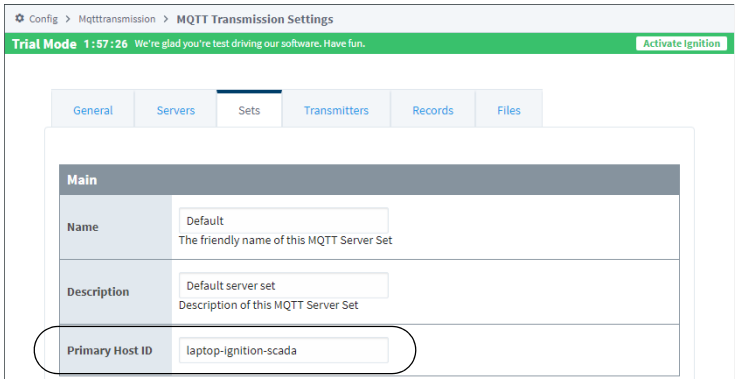
groov Manage

To designate a primary host application, go to Data Service > MQTT Sparkplug Configuration and set the Primary Host Topic to the MQTT client ID of your target application.



Ignition Edge

A broker failover group, called a Server Set in Ignition Edge, can have a designated Primary Host ID. To set it up, choose MQTT Transmission > Settings > Sets tab and click Edit next to the set you want. Enter the ID and save.



Store-and-forward history

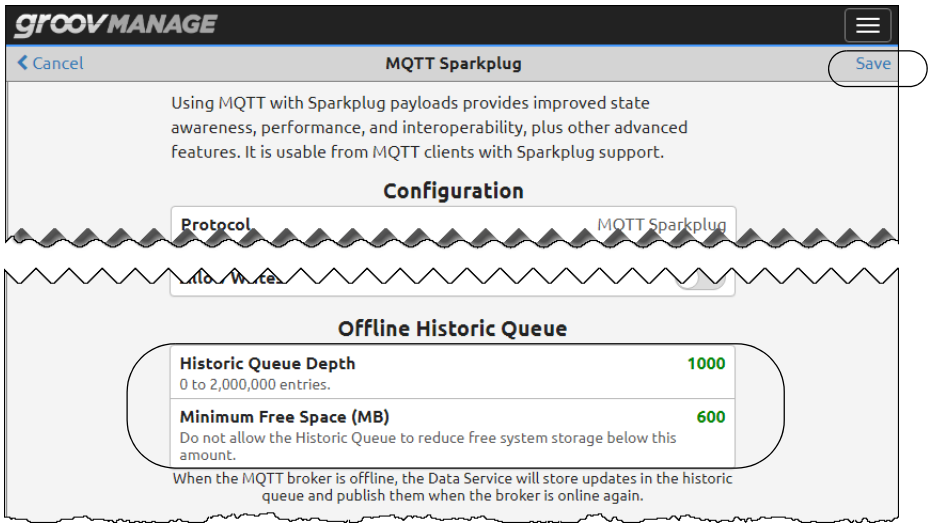
With a primary host ID set, *groov* devices can also store MQTT records temporarily when they lose connection to the primary host. A lost connection might be caused by a broker failure or a primary host failure. When the connection is restored, the *groov* device then forwards these records to the primary application.

To get the full advantage of store-and-forward history, you'll need to be communicating with an in-network MQTT subscriber that is watching for these historical records. Both the Ignition MQTT Engine module and the Canary Labs MQTT Sparkplug B Data Collector can do this.

You can set up store-and-forward history in *groov* Manage for your *groov* device. For *groov* EPIC and GRV-R7-MM2001-10 only, you can also set up store-and-forward history in Ignition Edge.

groov Manage

To enable historization, go to Data Service > MQTT Sparkplug Configuration. Scroll down to the Offline Historic Queue section and enter the values you want. Then scroll up and click Save (top right).



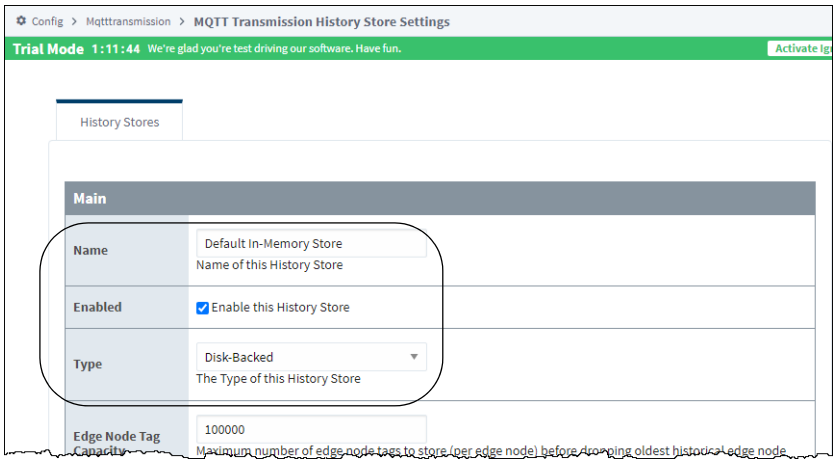
Note that records are stored to disk and will be lost in the event of a power cycle or power failure.

Additional performance details on the Historic Queue can be found in the user's guide for your *groov* device.

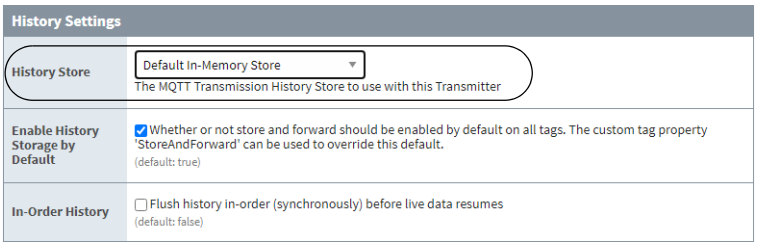
Ignition Edge (*groov* EPIC and GRV-R7-MM2001-10 only)

groov EPIC's Ignition Edge client offers larger capacity and a non-volatile storage option. If you select the disk-backed option, you get the additional benefit of your *groov* device's power-fail-safe file system and industrial SSD (solid state drive).

1. To enable historization, navigate to Config > MQTT Transmission > History and edit the Default In-Memory Store. Select Enable this History Store and change Type to Disk-Backed (if desired).



2. (Optional) Adjust tag capacities here if you want.
3. Click Save Changes.
4. Once done, choose MQTT Transmission > Settings > Transmitters tab. Scroll down to History Settings and set History Store to the name of the store you just enabled ("Default Store" in this example).

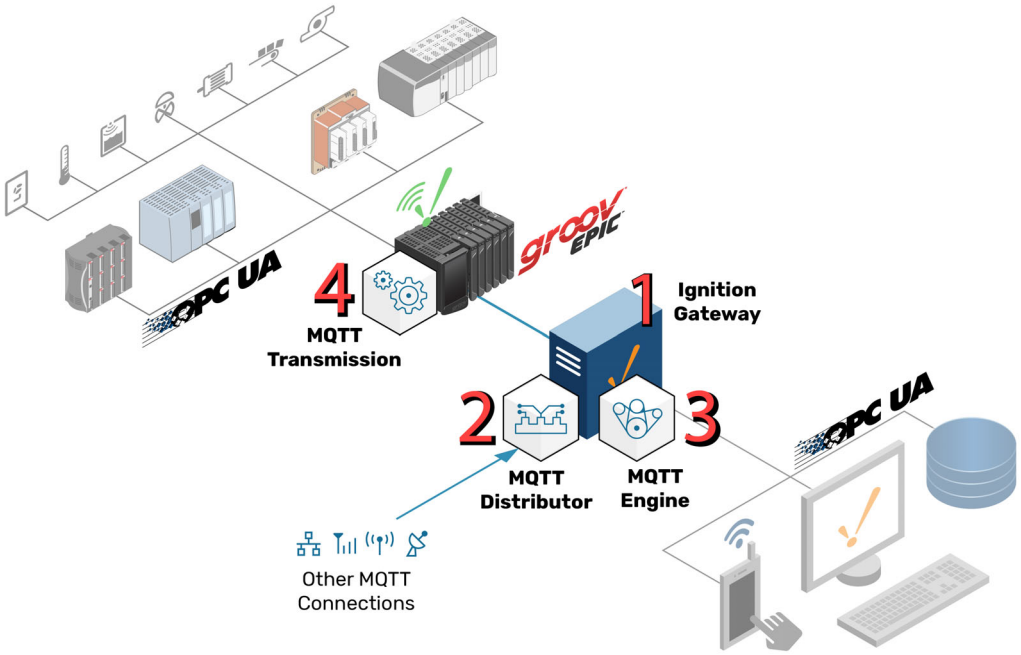


5. Click Save Changes.

3: Build a Proof of Concept

BUILDING A QUICK PROOF OF CONCEPT

A starter MQTT network using *groov* EPIC and Ignition



Here's our go-to formula for setting up an MQTT network that provides a full set of integration options and can be fully tested at no additional cost:

1. Download and install Ignition to a local computer (Linux, Mac, Windows). It takes only a few minutes, and Ignition is fully functional in two-hour increments.
2. Within Ignition, install the Cirrus Link MQTT Distributor module (Modules > Install or Upgrade a Module). You now have a scalable MQTT broker.
3. Next, install the Cirrus Link MQTT Engine module. Non-MQTT applications can now access your MQTT data through Ignition's OPC UA Server.
4. Finally, use one of many integration options to turn your *groov* device into an MQTT gateway for your legacy systems:
   - On your *groov* EPIC or RIO model GRV-R7-MM2001-10, enable the Cirrus Link MQTT Transmission module in Ignition Edge (see "Option 2: Ignition Edge (groov EPICs and GRV-R7-MM2001-10 only)"

on page 15) and install OPC drivers for Allen-Bradley, Siemens, and Modbus/TCP devices; or leverage integration kits for PAC Control to access a variety of serial protocols.

- On other *groov* RIO models, connect wired I/O signals and serial devices, then publish them through *groov* Manage or Node-RED.

Ready to go, and you haven't spent a dime! Experiment as much as you want, and when you're ready for production, all you have to do is license the system (for Ignition Edge on *groov* EPIC, just purchase GROOV-LIC-EDGE; for other Ignition components, contact Inductive Automation). Or, once you are satisfied that MQTT can do what you need, you can opt for no-cost components like HiveMQ's cloud broker and *groov* Manage client, instead.

A: MQTT Brokers

SELECTING AN MQTT BROKER

The focus of this guide is on understanding how to connect your *groov* device to an MQTT infrastructure, but field devices are only one part of the equation. The MQTT server is also a critical component, so here are a few recommended options. All of them are field proven, with many real-world users, and all support user authentication, certificate-based TLS encryption, and access control lists (ACLs).

Eclipse Mosquitto	Lightweight, fast, and free. A good option if you're comfortable on the command line.
	Cost: Free, open-source
	Capacity: Successfully tested with 100,000 clients.
	Requirements: Compatible with Windows, Mac, and Linux. ~120 KB executable; 3 MB RAM is enough to support 1000 clients.
	Technical support: Community support only
HiveMQ	This fully managed, cloud-based broker is 100% compliant with the MQTT specification and offers an easy way to get started and then scale up.
	Cost: Free up to 100 MQTT devices, up to 10 GB data.* $0.10/month per device to 1000 devices and $0.15 /month per GB to 100 GB. Contact HiveMQ for cost above these limits.
	Capacity: See Cost.
	Requirements: None; cloud-based
	Technical support: Community support for free version; basic support for paid; 24/7 support available for dedicated server
MQTT Distributor	A broker used in conjunction with Ignition, running on a PC, server, or in the cloud. Ignition is designed for industrial applications. This broker is easy to set up and boasts features for scalability and redundancy as well as integration with other features of the Ignition platform.
	Cost: $2,950/$4,250**, or free on two-hour timer
	Capacity: Available in 50- and 250-client versions.
	Requirements: Ignition or Ignition Edge platform. Ignition requires a dual-core processor and 4 GB RAM.
	Technical support: Free email and forum support from Cirrus Link. 90-day free post-sales telephone support. Long-term support contracts available.

Chariot V2 MQTT Server	Designed by the co-inventors of MQTT and Sparkplug B for industrial end-users, Chariot is a functional replacement for MQTT Distributor for larger deployments and for non-Ignition users.
	Cost: $7,950***, or free on two-hour timer
	Capacity: Unlimited clients
	Requirements: See installation details on Cirrus Link's website. Can be installed as a virtual machine on a private network or in the cloud via Amazon AWS EC2
	Technical support: Free phone and email support

* Price as of 6/24/22
** Price as of 7/20/20.
*** Price as of 7/20/20.