

# GETTING STARTED WITH MQTT IN GROOV PRODUCTS

Form 2350-201013–October 2020

**OPTO 22**  
The Edge of 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](http://www.opto22.com)

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

Getting Started with MQTT in *groov* Products  
Form 2350-201013—October 2020

Copyright © 2020 Opto 22.  
All rights reserved.

Printed in the United States of America.

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.

---

Wired+Wireless controllers and brains are licensed under one or more of the following patents: U.S. Patent No(s). 5282222, RE37802, 6963617; Canadian Patent No. 2064975; European Patent No. 1142245; French Patent No. 1142245; British Patent No. 1142245; Japanese Patent No. 2002535925A; German Patent No. 60011224.

Opto 22 FactoryFloor, *groov*, *groov* EPIC, *groov* RIO, mobile made simple, The Edge of Automation, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, *groov* Server, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, *mistic*, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDataLink, OptoDisplay, OptoEMU, OptoEMU Sensor, OptoEMU Server, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, PAC Control, PAC Display, PAC Manager, PAC Project, PAC Project Basic, PAC Project Professional, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC System, SNAP Simple I/O, SNAP Ultimate I/O, and Wired+Wireless are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C++, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric, licensed to the Modbus Organization, Inc. Wiegand is a registered trademark of Sensor Engineering Corporation. Allen-Bradley, CompactLogix, ControlLogix, MicroLogix, SLC, and RSLogix are either registered trademarks or trademarks of Rockwell Automation. CIP and EtherNet/IP are trademarks of ODVA. Raspberry Pi is a trademark of the Raspberry Pi Foundation. The registered trademark Ignition by Inductive Automation® is owned by Inductive Automation and is registered in the United States and may be pending or registered in other countries. CODESYS® is a registered trademark of 3S-Smart Software Solutions GmbH.

*groov* includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org>)

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Opto 22  
The Edge of Automation.



# Table of Contents

<b>Chapter 1: Getting Started with MQTT</b> .....	<b>1</b>
Introduction .....	1
In this Guide .....	1
For Help .....	1
Related Documents .....	1
OptoForums .....	2
Product Support .....	2
General prerequisites .....	2
Comparing <i>groov</i> MQTT clients .....	2
String vs. Sparkplug B payloads .....	3
Enabling public tag access .....	4
Exposing I/O data in <i>groov</i> Manage .....	4
Exposing I/O and tag data in PAC Control ( <i>groov</i> EPIC only) .....	4
Configuring simple data collection and control .....	5
Option 1: <i>groov</i> Manage with string payloads .....	5
Option 2: Node-RED .....	7
Configuring high-reliability SCADA/IIoT .....	9
Option 1: <i>groov</i> Manage with Sparkplug B payloads .....	9
Option 2: Ignition Edge ( <i>groov</i> EPIC only) .....	10
Option 3: Node-RED .....	12
<b>Chapter 2: Fault Tolerance and Security</b> .....	<b>15</b>
Configuring additional fault tolerance options .....	15
Failover connections .....	15
<i>groov</i> Manage .....	15
Ignition Edge .....	15
Node-RED .....	16
Primary host ID .....	16
<i>groov</i> Manage .....	16
Ignition Edge .....	17
Store-and-forward history .....	17
<i>groov</i> Manage .....	17
Ignition Edge ( <i>groov</i> EPIC only) .....	18
Enabling MQTT security .....	19
For all clients .....	19

Security in <i>groov</i> Manage .....	19
Security in Ignition Edge .....	20
Security in Node-RED .....	20
<b>Chapter 3: Build a Proof of Concept .....</b>	<b>23</b>
Building a quick proof of concept .....	23
<b>Appendix A: MQTT Brokers .....</b>	<b>25</b>
Selecting an MQTT broker .....	25

# 1: Getting Started with MQTT

## INTRODUCTION

Opto 22's *groov* EPIC® edge programmable industrial controller and *groov* RIO® edge I/O give you many ways to use MQTT data communications—via *groov* Manage, Node-RED, or Ignition Edge from Inductive Automation®. The goal of this guide is to help you understand which approach is best for you and to help you get started fast.

If you already know which approach you'll take, you can jump to the how-to section that best describes the application you have in mind. If you're still thinking about your approach, continue reading for a perspective on the general trade-offs between the available options.

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 enabling public tag access, configuring simple data collection, and configuring high-reliability SCADA/IIoT using MQTT

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

**Chapter 3: Build a Proof of Concept**—Steps to build a simple 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](http://www.opto22.com) and search on the form number.

Guide name	Contents	Form #
<a href="#">groov EPIC User's Guide</a>	Installing and using a <i>groov</i> EPIC system; using Node-RED, MQTT, and Ignition or Ignition Edge	2267
<a href="#">groov RIO User's Guide</a>	Installing and using a <i>groov</i> RIO module; using Node-RED and MQTT	2324
<a href="#">Guide to Networking groov Products</a>	Using <i>groov</i> products in your network and over the internet	2161

## GENERAL PREREQUISITES

### 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

*NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.*

**Fax:** 951-695-3017

**Email:** [support@opto22.com](mailto:support@opto22.com)

**Opto 22 website:** [www.opto22.com](http://www.opto22.com)

## GENERAL PREREQUISITES

This guidebook 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, [reach out to Opto 22 for a demo](#). For steps to log in or complete initial setup, see the [groov EPIC User's Guide](#) or the [groov RIO User's Guide](#).)
- 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](#) or at [Chapter 3: Build a Proof of Concept](#), a walkthrough that will help you set up a complete MQTT system quickly.

### Comparing *groov* MQTT clients

When determining which approach is right for a given application, there are a few key points to consider, like network size, engineering effort, fault tolerance, and budget. The table below summarizes details of these and other points.

In general, both *groov* Manage and Node-RED let you get started quickly with publishing a small amount of data and do not require any additional licensing costs. With either one, you can take advantage of the basic network and data protections that MQTT and *groov* devices provide, like device-originating connections, authentication, encryption, and certification.

If your application requires something more resilient, configuring *groov* Manage for Sparkplug payloads gives you the guarantee of state awareness, important if you rely on high-quality data for operations, auditing, or historization. (With some extra work, Node-RED can work with Sparkplug data also.) Sparkplug also unlocks additional fault tolerance features and makes it easier to work with larger data sets.

Ignition Edge can handle even larger data sets and provides more integration and fault tolerance options, but it does impose some additional licensing costs.

	<i>groov</i> Manage w/ string payloads	Node-RED	<i>groov</i> Manage w/ Sparkplug B payloads	Ignition Edge
Platform	<i>groov</i> EPIC <i>groov</i> RIO	<i>groov</i> EPIC <i>groov</i> RIO	<i>groov</i> EPIC <i>groov</i> RIO	<i>groov</i> EPIC
Tag count	< 50	< 50	<1000	1000+
Payload	Strings	Strings, JSON, Sparkplug B	Sparkplug B	Strings, JSON, Sparkplug B
Data source	Local I/O, PAC Control™ <sup>1</sup>	Local I/O, PAC Control <sup>1</sup> , databases, web services, OPC, and more	Local I/O, PAC Control <sup>1</sup>	PAC Control, databases, legacy PLCs, devices, or OPC <sup>2</sup>
Subscribe to other publishers' topics	No	Yes	No	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	User authentication SSL/TLS encryption Certificate management
Historization	None	None	Volatile storage Max 65K records	Non-volatile storage (optional) 100K records or 1 week
Failover	Multiple brokers	None	Multiple brokers Primary host designation	Multiple brokers Multiple failover groups Multiple clients Primary host designation
Additional Cost	\$0	\$0	\$0	\$800 <sup>3</sup> +\$200 <sup>4</sup> w/ MQTT Engine

<sup>1</sup> *groov* EPIC only  
<sup>2</sup> Other data sources are possible through the use of additional Ignition modules.  
<sup>3</sup> [GROOV-LIC-EDGE](#) current list pricing as of 6/10/20.  
<sup>4</sup> 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, it requires 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 when 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 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, any broker can handle Sparkplug B traffic alongside string payloads without an issue. However, any device or application clients that want to use your Sparkplug data will need to support Sparkplug as well.

## ENABLING PUBLIC TAG ACCESS

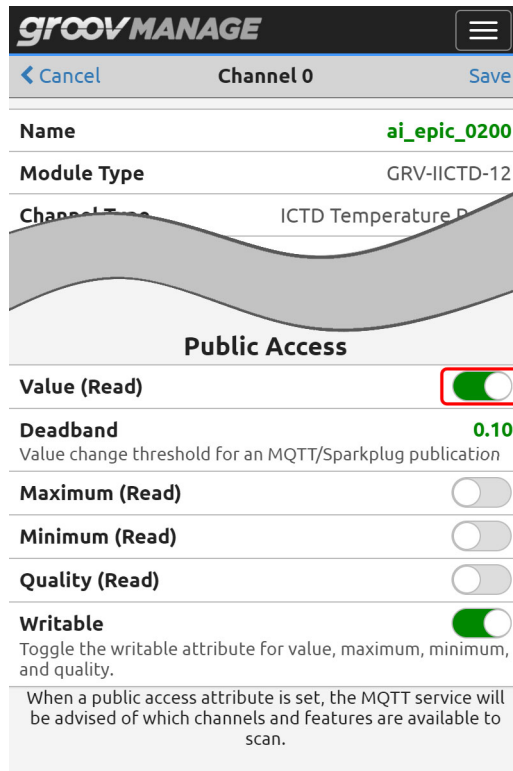
In order for the MQTT clients on *groov* EPIC or *groov* RIO to publish your I/O and tag data, you need to allow that data to be read by other services.

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.

### 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 > 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)

Use PAC Control to expose I/O or variable tags from a control strategy running on your *groov* EPIC processor to Ignition Edge, the *groov* Manage MQTT service (when the Device Type is set to Controller), or by the PAC Control nodes (*node-red-contrib-pac*) in Node-RED.



To enable access, open the edit dialog box of each I/O point or variable to be published, and check Make Public (Readable). 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 (fewer than 50 tags or so). How do you quickly take that data and publish it from your *groov* device using MQTT? As usual, we give you a couple of options: *groov* Manage or Node-RED.

*NOTE: If you want 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](#).*

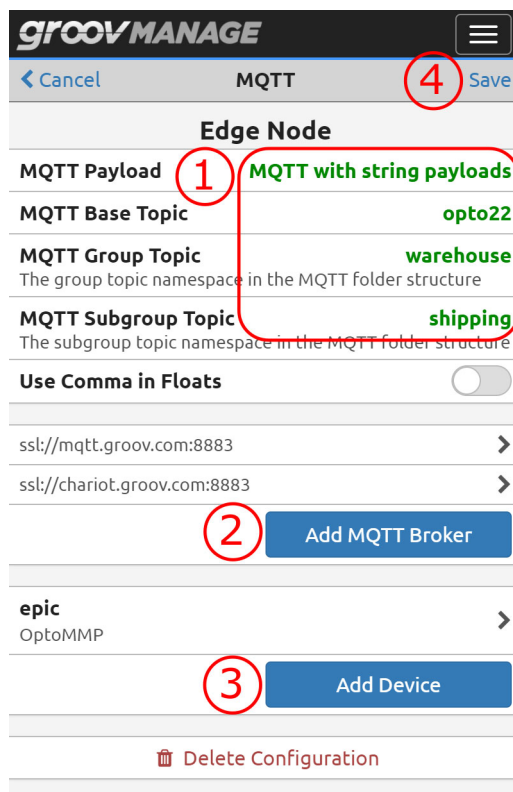
### Option 1: *groov* Manage with string payloads

The option for plain text (string) payloads is available in *groov* RIO and in *groov* EPIC firmware release 1.4.2+. It is, by far, the simplest way to get started with MQTT.

You can do it in just a few steps:

1. Log into your *groov* EPIC or *groov* RIO, and navigate to MQTT > Configuration:
  - a. Set MQTT Payload to MQTT with string payloads.
  - b. Configure an appropriate topic path (see an example in the image on the next page).

2. Configure a broker connection:
  - a. Click Add MQTT Broker, and enter the appropriate URL and security details (if needed; see “Security in groov Manage” on page 19) to point the client to your MQTT broker.
  - b. Click OK.
3. Configure your device profile:
  - a. Click Add Device.
  - b. (*groov* EPIC only) Select the appropriate Device Type:
    - Controller, if you want to publish tags from your PAC Control strategy
    - OptoMMP, if you want to publish I/O directly from *groov* Manage
  - c. Enter an MQTT Device Topic name for your *groov* device (for example, epic).
  - d. Click OK.
4. Back on the MQTT page, save your configuration.



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

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

In our example, the path would look like this:

opto22/warehouse/shipping/epic/#

For more information on using *groov* Manage for MQTT, see Chapter 8 in the *groov EPIC User's Guide* or Chapter 9 in the *groov RIO User's Guide*.

## Option 2: Node-RED

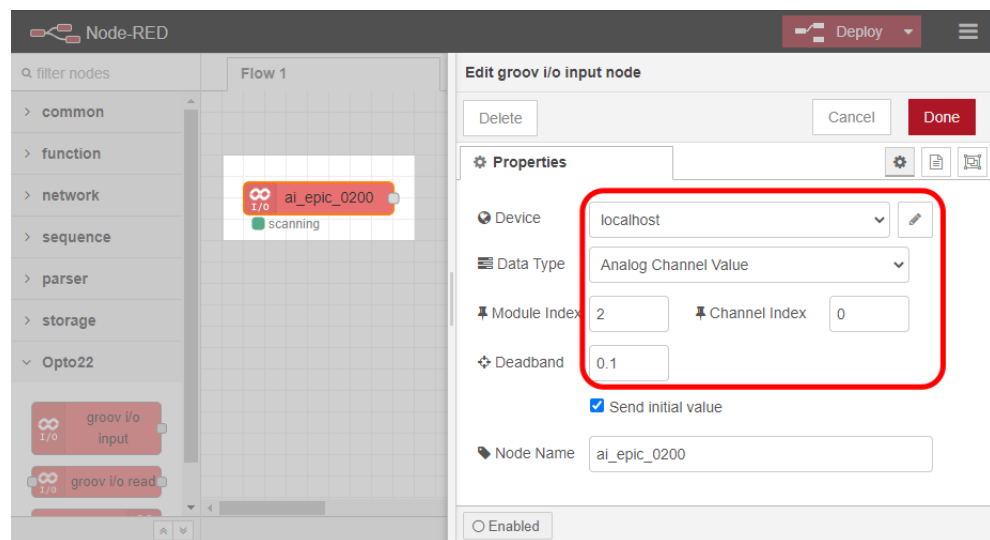
While *groov* Manage is great for straightforward tag publishing, for more flexibility and granularity, consider Node-RED. It has several advantages for basic publishing.

First, Node-RED makes it easy to combine tags with data from other sources, like OPC or 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. And in addition to plain strings, 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.

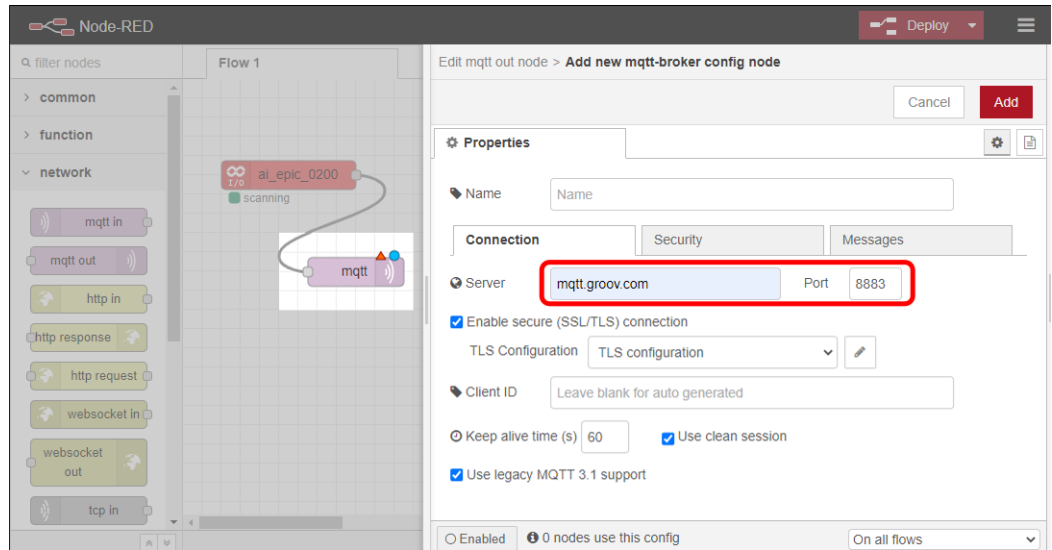
**Prerequisite:** You should be familiar with the basics of the Node-RED environment and how to build, configure, and deploy flows. If not, check out our [online training](#) first or review the beginning of Chapter 14 of the *groov EPIC User's Guide* or Chapter 10 of the *groov RIO User's Guide*.

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

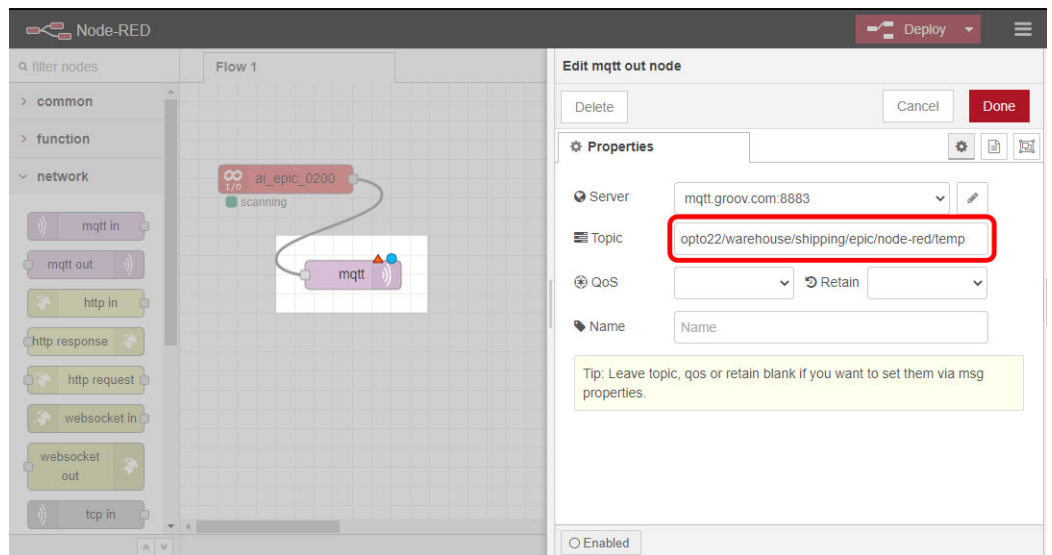
1. Log into your *groov* EPIC or *groov* RIO, navigate to the Node-RED page, and open the Node-RED editor.
2. Install the *groov* I/O nodes (`node-red-contrib-groov-io`) or your data source of choice, add a *groov* I/O input node to the workspace, and configure it to grab the I/O data you want to publish. (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, port, and security details (if needed; see [“Security in Node-RED” on page 20](#)). Click Add when finished.



5. Back in the MQTT node configuration panel, enter your desired topic path and click Done.



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 Chapter 14 of the [groov EPIC User's Guide](#) or Chapter 10 of the [groov RIO User's Guide](#).

## CONFIGURING HIGH-RELIABILITY SCADA/IIOT

Do you need to tap into the scalability and fault tolerance features that Sparkplug B offers? Or maybe your *groov* device needs to publish a large number of tags (up to 1000 or more). 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: Fault Tolerance and Security](#).

### 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. It supports some of the same fault tolerance features as Ignition Edge but without the added cost.

Follow the steps in “[Option 1: \*groov\* Manage with string payloads](#)” on page 5 with two exceptions:

1. Set the MQTT Payload to MQTT with Sparkplug payloads.
2. Enter the elements of your topic path in the Group and Edge Node ID fields.

MQTT	
<b>Edge Node</b>	
MQTT Payload	MQTT with Sparkplug payloads
Group ID <small>The Sparkplug Group ID</small>	opto22
Edge Node ID <small>The Sparkplug Edge Node ID</small>	shipping
Primary Host ID	Optional
Compression	none
ssl://mqtt.groov.com:8883 >	
ssl://chariot.groov.com:8883 >	
<b>Add MQTT Broker</b>	
epic >	
OptoMMP	
<b>Add Device</b>	
<b>Delete Configuration</b>	

*groov* Manage will generate a topic path based on the ID fields in combination with others required by Sparkplug B, as follows:

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

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

```
spBv1.0/opto22/+/shipping/epic/#
```

More details are available in Chapter 8 of the [groov EPIC User's Guide](#) and Chapter 9 of the [groov RIO User's Guide](#).

## Option 2: Ignition Edge (*groov* EPIC only)

Ignition Edge is available only on *groov* EPIC, and 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 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* EPIC on a trial basis. The trial lasts for two hours and can be repeated as often as you wish.

**Prerequisite:** Java installed on your computer (required by Ignition Designer).

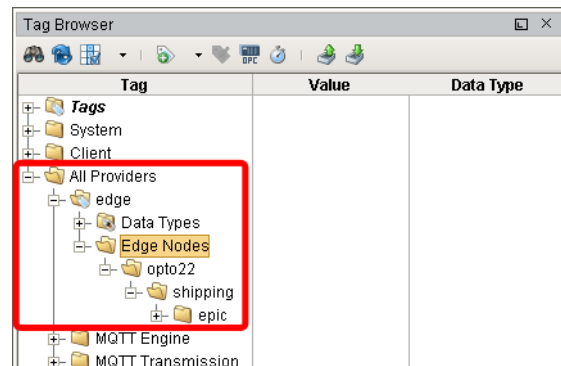
1. Log into your *groov* EPIC, navigate to the Ignition page, and enable Ignition Edge and Ignition Designer. Click Save.
2. Click the link to Open Ignition Edge, and log in using your credentials (default: admin/password).
3. Install *groov* EPIC drivers:
  - a. Navigate to Configure > (System) Modules, 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.
4. Add *groov* EPIC as an OPC UA device:
  - a. Navigate to Configure > (OPC-UA Server) Devices, and click Create new Device.
  - b. Select Opto 22 *groov* EPIC and SNAP PAC. Then click Next.
  - c. Enter a Name and use localhost as the Address. Click Create New Device.
5. Install the MQTT Transmission module:
  - a. In the top navigation, click Configure, and scroll down till you see MQTT. Click Install+.
  - b. On the next page, scroll down to Quarantined modules, and click the install button for MQTT Transmission.
  - c. Follow directions on the screen to accept the license.
  - d. Check "I want to add this certificate to my trusted certificates and install the module." Click the button to install.
6. Add a broker connection to the MQTT Transmission module:
  - a. Navigate to Configure > (MQTT Transmission) Settings
  - b. Select the Servers tab, and click Create new MQTT Server.
  - c. Enter a Name, URL, Server Type, and security details (if needed; see ["Security in Ignition Edge" on page 20](#)). Then click Create New MQTT Server.  
Server Type options:
    - Chariot: Cirrus Link's Chariot MQTT Server
    - MQTT\_Distributor: Ignition MQTT Distributor Module
    - Third\_Party: An MQTT server other than the two above

## MQTT Transmission Settings

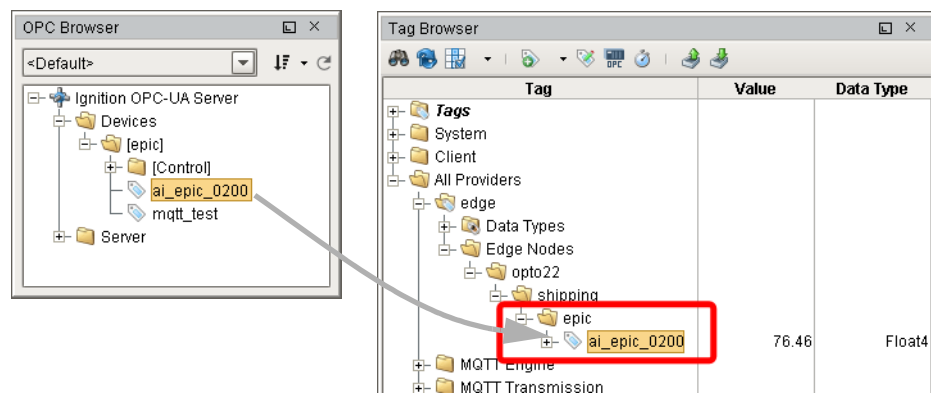
Name	URL	Server Type	Server Set	Username	Certificate Files	Connected	
Opto 22 Chariot Broker	ssl://chariot.groov.com:8883	Chariot	Default		opto22selfca.pem	0 of 0	delete edit
Opto 22 Mosquitto Broker	ssl://mqtt.groov.com:8883	Third_Party	Default		opto22selfca.pem	0 of 0	delete edit

7. Open Ignition Designer, configure your topic path, and publish your tags:
  - a. In the upper right-hand corner of Ignition Edge, click Launch Designer and open the designer.jnlp file when the download is complete.
  - b. Log in with your credentials (default: admin/password).
  - c. In the Tag Browser panel, navigate to All Providers > edge > Edge Nodes.
  - d. Right-click the Edge Nodes folder and select New Folder. Then enter a name to represent the Group ID, and click OK.
  - e. Right-click the new folder and click New Folder again. Then enter a name to represent the Edge Node ID, and click OK.
  - f. Right-click the latest folder and click New Folder again. Then enter a name to represent the Device ID, and click OK.

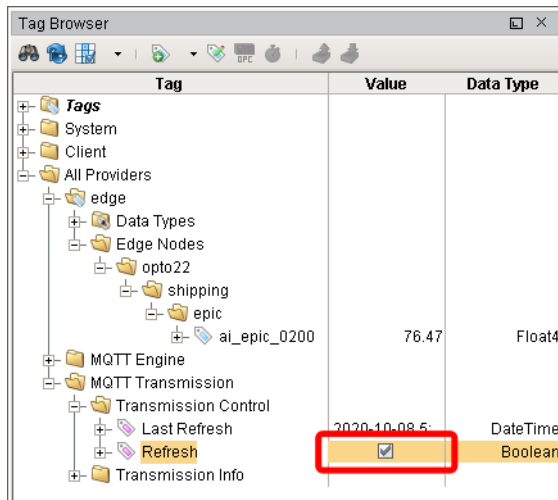
Here's an example of the folder structure:



8. Open the OPC Browser (View > Panels > OPC Browser), open the Devices folder, and drag the desired tags from your EPIC device folder into the Device ID folder you have just created.



- Publish tags by navigating to MQTT Transmission > Transmission Control in the Tag Browser and clicking the checkbox next to the Refresh tag.



A dialog box may appear with the prompt “Change to Read/Write Mode?” If so, click Enable Read/Write Mode.

The checkbox is checked and then immediately cleared when refresh has been completed.

- Repeat [step 9](#) any time you modify the folder structure or add a new tag.
- Once you have finished adding tags, return to the Ignition page in *groov* Manage, disable Ignition Designer, and click Save.

This setup produces an MQTT/Sparkplug B topic path according to the folder structure you have created: `spBv1.0/<Group ID>/+/<Edge Node ID>/<Device ID>/#`

In this example, MQTT subscribers would use the following topic path: `spBv1.0/opto22/+/shipping/epic/#`

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, just reset the timer. For uninterrupted use, you can purchase an Ignition Edge license ([GROOV-LIC-EDGE](#)).

See Chapter 8 of the *groov EPIC User’s Guide* for more information on using Sparkplug B with Ignition Edge.

### Option 3: Node-RED

If you need your *groov* device to only *subscribe* to Sparkplug B 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.

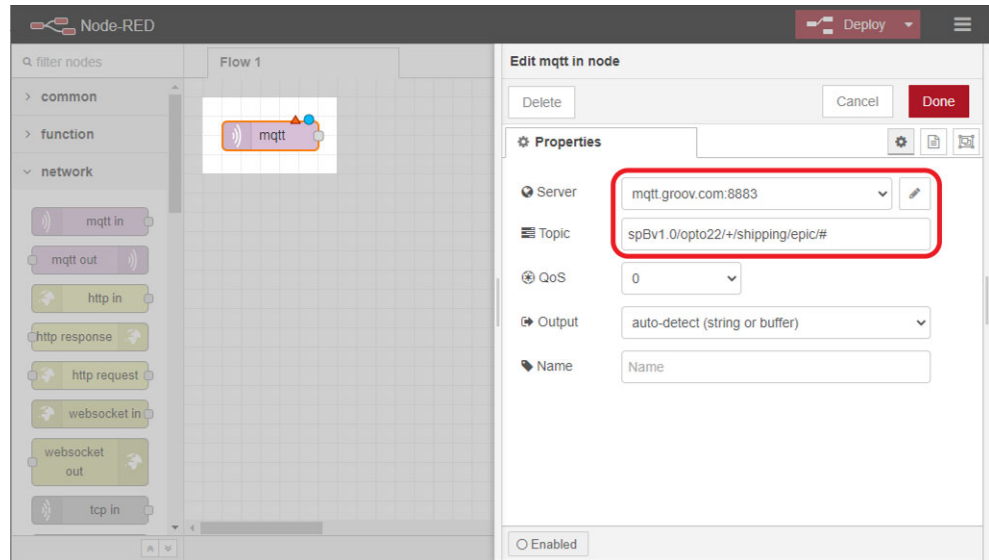
Using this method, Node-RED does not become a Sparkplug-compliant device; it is just a data consumer subscribed to an MQTT broker. But sometimes that’s all you need.

Here’s the procedure:

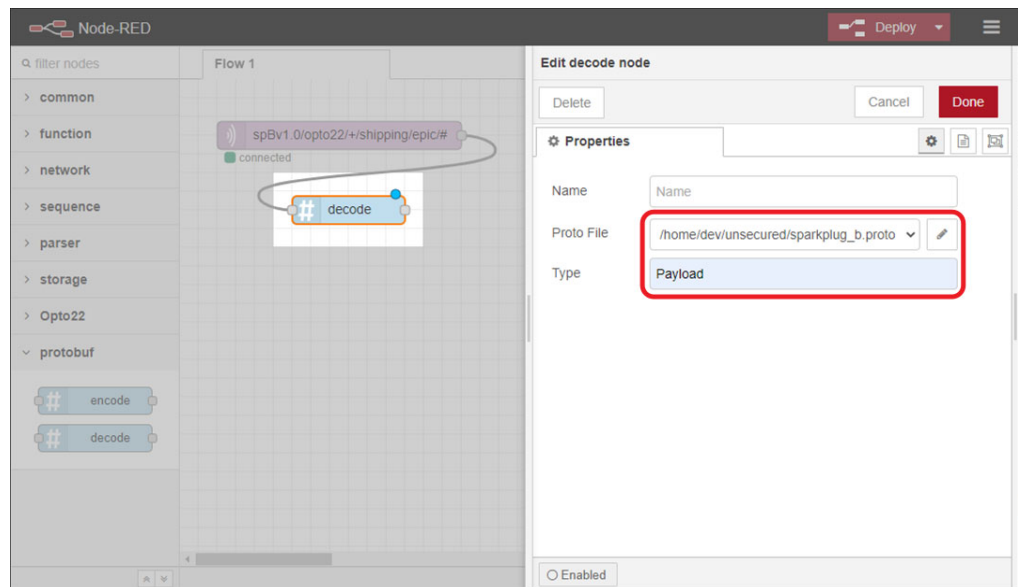
- Log into your *groov* EPIC or *groov* RIO, navigate to the Node-RED page, and open the Node-RED editor.
- Place an MQTT In node, and configure the broker connection, security details (if needed; see “[Security in Node-RED](#)” on page 20), and topic path. The path should use this format:

`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: Fault Tolerance and Security

## 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](#)

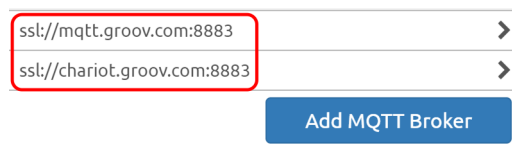
### Failover connections

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 16](#)).

#### *groov* Manage

To set up failover connections, navigate to MQTT > Configuration > Add Broker, and create as many connections as you want.



The screenshot shows a configuration table with two rows of MQTT broker addresses. The first row is 'ssl://mqtt.groov.com:8883' and the second row is 'ssl://chariot.groov.com:8883'. Both rows have a right-pointing chevron icon. Below the table is a blue button labeled 'Add MQTT Broker'.

ssl://mqtt.groov.com:8883	>
ssl://chariot.groov.com:8883	>

Add MQTT Broker

The example above shows a connection to Opto 22's on-site Mosquitto broker as well as a second connection to a Chariot broker hosted on AWS.

#### 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.

### MQTT Transmission Settings

Name	URL	Server Type	Server Set	Username	Certificate Files	Connected		
Opto 22 Chariot Broker	ssl://chariot.groov.com:8883	Chariot	Default		opto22selfca.pem	0 of 0	delete	edit
Opto 22 Mosquitto Broker	ssl://mqtt.groov.com:8883	Third_Party	Default		opto22selfca.pem	0 of 0	delete	edit

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, but only one connection can be configured per node. To add some redundancy, you can create duplicate nodes with connections to different brokers.

#### Primary host ID

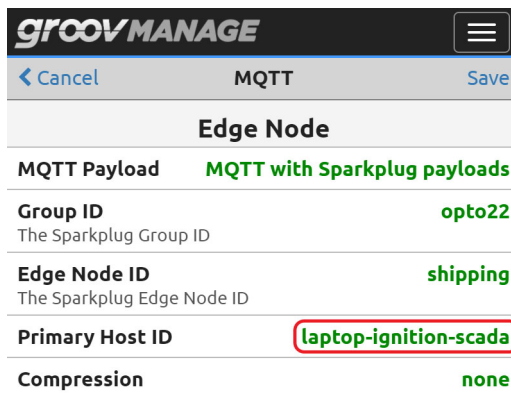
If you've configured multiple broker connections for failover (see page 15) in *groov* Manage or Ignition Edge, you can also set a Primary Host ID. This setting causes the MQTT client to give preference to broker connections that publish an ONLINE status for the specific client ID you enter.

Be aware that 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 the main MQTT configuration screen and set the Primary Host ID 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.

Set it up by editing the desired server set under the (MQTT Transmission) Settings > Sets tab.

### MQTT Transmission Settings

MQTT Transmission Settings			
General Servers Sets Transmitters Records			
Name	Description	Primary Host ID	
Default	Default server set	laptop-ignition-scada	delete edit

## Store-and-forward history

With a primary host ID set, *groov* devices can also store MQTT records temporarily when they lose connection to the broker, and then forward these records to the primary application when the connection is restored.

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. Currently, there are two applications that can do this: the Ignition MQTT Engine module and the Canary Labs MQTT Sparkplug B Data Collector.

You can set up store-and-forward history in *groov* Manage or, for *groov* EPIC only, in Ignition Edge.

### *groov* Manage

To enable historization, navigate to MQTT > Configuration > Device and enter a number into the Historic Queue field.

**Device**

<b>Device Type</b>	<b>Controller</b>
<b>Device ID</b>	<b>epic</b>
<small>The Sparkplug Device ID</small>	
<b>Host TCP Port</b>	<b>22001</b>
<b>CommTimeout(ms)</b>	<b>5000</b>
<b>Scan Time (ms)</b>	<b>1000</b>
<b>Additional Host Tasks</b>	<i>Optional</i>
<b>Historic Queue</b>	<b>1000</b>

We recommend a queue size of 3,600 to strike a balance between storage and performance, but you could go much higher, depending on the rate of data production.

*Note that records are stored in RAM and will be lost in the event of a power failure.*

Additional performance details on the Historic Queue can be found in Chapter 8 of the [groov EPIC User's Guide](#) and Chapter 9 of the [groov RIO User's Guide](#).

Ignition Edge (*groov* EPIC 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 *groov* EPIC’s power-fail-safe file system and industrial SSD (solid state drive).

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

MQTT Transmission History Store Settings

History Stores

### Edit Transmission History Store

Main	
Name	Default Store Name of this History Store
Enabled	<input checked="" type="checkbox"/> Enable this History Store
Type	Disk-Backed The type of this History Store

2. Click Save Changes.
3. (Optional) Adjust the history capacity here if you want.  
The default setting buffers up to 100,000 records for 1 week.
4. Once done, navigate to the (MQTT Transmission) Settings menu and select the Transmitters tab. Scroll down and set History Store to the name of the store you just enabled (“Default Store” in this example).

MQTT Transmission Settings

General Servers Sets Transmitters Records

Default Custom

### Configuration

Name	Default Transmitter Name for the Default Transmitter
History Store	Default Store The MQTT Transmission History Store to use with the Default Transmitter

5. Click Save Changes.

## 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 specify a username and password to authenticate broker connections, to apply certificates of trust, and to enable encrypted communications.

### For all clients

The following sections give the specifics of each MQTT client related to enabling security, but for all clients you need to input the appropriate protocol and port designations in your broker URL to indicate you are using an encrypted connection. TCP ports 8883 and 1883 are registered with IANA for MQTT TLS and non-TLS communication, respectively.

For example:

Replace unencrypted URL...	with encrypted URL
<code>tcp://mqtt.groov.com:1883</code>	<code>ssl://mqtt.groov.com:8883</code>

See the [groov EPIC User's Guide](#) and [groov RIO User's Guide](#) for more information on security features, including certificate creation and management.

### Security in *groov* Manage

1. When adding an MQTT broker, in addition to entering the correct broker URL and user credentials, also enable the SSL option and select the appropriate CA certificate.

The screenshot shows the 'MQTT Broker' configuration dialog. The 'Broker URL' field contains 'ssl://mqtt.groov.com:8883'. The 'Client ID' field has a placeholder 'e.g. any\_unique\_id'. The 'Username' field is empty. The 'Password' field is masked with dots. The 'SSL' toggle is turned on. The 'CA Certificate' dropdown is set to 'opto22selfca.pem'. The 'ConnTimeout (ms)' is set to '5000' and 'Keep Alive (s)' is set to '10'. At the bottom are 'Cancel', 'Delete', and 'OK' buttons.

2. To upload a security certificate to your *groov* device, navigate to the Security > Client SSL page, click Add/Update, and locate the appropriate file. Once this is done, the certificate appears as an option in the MQTT Broker > Select Certificate dialog box.

## Security in Ignition Edge

1. When adding a broker connection, specify the appropriate protocol and port in the URL, enter a username, and check the box next to Change Password. Then enter the appropriate password twice.

Main	
Name	<input type="text" value="Opto 22 Mosquitto Broker"/> The friendly name of this MQTT Server
URL	<input type="text" value="ssl://mqtt.groov.com:8883"/> The URL of the MQTT Server to connect to. Should be of the form tcp://mydomain.com:1883 or ssl://mydomain.com:8883
Server Type	<input type="text" value="Third_Party"/> The type of MQTT Server to connect to (default: MQTT_Distributor)
Server Set	<input type="text" value="Default"/> The Server Set this MQTT Server is associated with
Username	<input type="text"/> The username for this MQTT connection if required by the MQTT Server (optional)
Change Password?	<input checked="" type="checkbox"/> Check this box to change the existing password.
Password	<input type="password" value="*****"/> The password for this MQTT connection if required by the MQTT Server (optional)
Password	<input type="password" value="*****"/> Re-type password for verification.

2. In the TLS section below, click Choose File next to Certificate File Upload, and locate the desired certificate file.

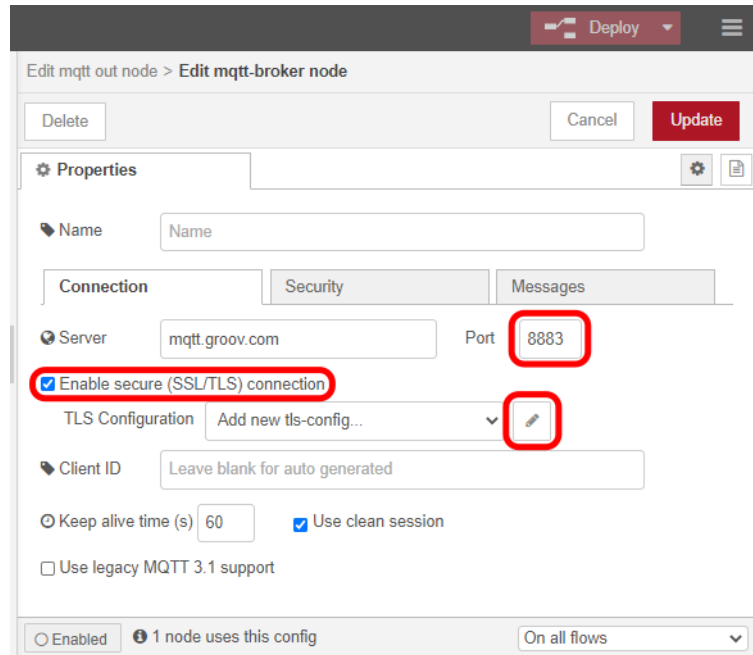
TLS	
Certificate File Upload	<input type="button" value="Choose File"/> No file chosen Files: <input type="text" value="C:\fakepath\opto22selfca.pem"/> Certificate files to upload. This will replace all previously uploaded certificate files. <input type="button" value="Delete"/>
Change Password?	<input type="checkbox"/> Check this box to change the existing password.
Password	<input type="password"/> The password associated with the certificate's private key (optional)
Password	<input type="password"/> Re-type password for verification.

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

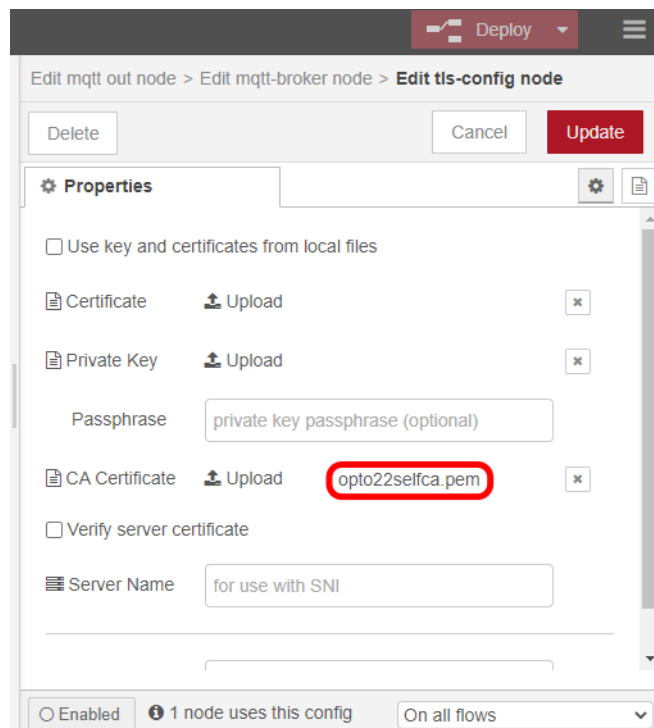
## Security in Node-RED

1. When editing the MQTT broker details, be sure to specify TLS port 8883. Then check the box to Enable secure (SSL/TLS) connection and click the pencil next to Add new tls-config.





2. In the TLS configuration panel, upload the appropriate certificate files.



3. Click Update.
4. To enter user credentials, switch to the Security tab. Enter your username and password.

## ENABLING MQTT SECURITY

The screenshot shows a configuration window titled "Edit mqtt out node > Edit mqtt-broker node". At the top, there are buttons for "Delete", "Cancel", and "Update". Below this is a "Properties" section with a gear icon and a document icon. The "Name" field is labeled "Name". There are three tabs: "Connection", "Security", and "Messages". The "Security" tab is active, showing "Username" and "Password" fields. A red rectangle highlights the "Username" and "Password" fields. At the bottom, there is a status bar with "Enabled", "1 node uses this config", and "On all flows" with a dropdown arrow.

5. Click Update when done.



and Modbus/TCP devices; or leverage [integration kits](#) for PAC Control to access a variety of serial protocols.

- On your *groov* RIO, 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 the Eclipse Mosquitto 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).

<b>Eclipse Mosquitto</b>	<p>Lightweight, fast, and free. A good option if you're comfortable on the command line. Already included in the <i>groov</i> EPIC repository for users who have installed the <a href="#">free Shell access license</a>.</p> <p><b>Cost:</b> Free, open-source</p> <p><b>Capacity:</b> Successfully tested with 100,000 clients.</p> <p><b>Requirements:</b> ~120 KB executable; 3 MB RAM is enough to support 1000 clients.</p> <p><b>Technical support:</b> Community support only</p>
<b>Ignition MQTT Distributor</b>	<p>Designed for industrial applications, Ignition is easy to set up and boasts features for scalability and redundancy as well as integration with other features of the Ignition platform.</p> <p><b>Cost:</b> \$2,950/\$4,250*, or free on two-hour timer</p> <p><b>Capacity:</b> Available in 50- and 250-client versions.</p> <p><b>Requirements:</b> <a href="#">Ignition or Ignition Edge platform</a>. Ignition requires Java, a dual-core processor, and 4 GB RAM.</p> <p><b>Technical support:</b> Free email and forum support. 90-day free post-sales telephone support. <a href="#">Long-term support contracts available</a>.</p>
<b>Chariot MQTT Server</b>	<p>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.</p> <p><b>Cost:</b> \$7,950**, or free on two-hour timer</p> <p><b>Capacity:</b> Unlimited clients</p> <p><b>Requirements:</b> Delivered as an OVF file for native import to VM or cloud computing platforms. Requires a quad-core processor and 8 GB RAM (or equivalent).</p> <p><b>Technical support:</b> Free phone and email support</p>

\* List price as of 7/20/20. Discounted Ignition Edge pricing available through Opto 22.  
\*\* List price as of 7/20/20.

