

## SignalFire Ranger – Ignition Integration Guide

The SignalFire Ranger is an Internet of Things (IoT) cellular LTE-M telemetry node capable of communicating to a variety SCADA host software using [Message Queuing Telemetry Transport \(MQTT\)](#) and [Sparkplug](#) protocol. This guide will walkthrough the steps necessary in connecting the SignalFire Ranger to [Ignition](#) from [Inductive Automation](#).

### Setup MQTT Broker

The SignalFire Ranger connects to an MQTT broker to publish its measurement data. Ignition must also connect to the same MQTT broker to subscribe to the measurements published by the Ranger and other nodes. This requires having access to your own MQTT broker and creating credentials for the Ignition server and the Ranger node to use to connect and login to the broker.

These MQTT broker credentials include:

- Hostname or IP Address (required)
- Port number (required)
- Username/Password (optional)
- TLS Settings (optional): CA certificate for host validation, Client certificate and key

The links below contain some options for setting up your own MQTT broker.

[CloudMQTT](#) – Hosted MQTT brokers

[Mosquitto](#), [EMQ X](#), [RabbitMQ](#) – Open-source MQTT brokers for Windows, Mac, or Linux

[HiveMQ](#), [Chariot](#), [MQTT Distributor](#) – Enterprise MQTT brokers

## Install Ignition

Follow the steps in this [Ignition installation guide](#) to install Ignition on Windows, Mac, or Linux.

After Ignition is installed and the commissioning process is completed, confirm that you can login to the Ignition Gateway webpage.

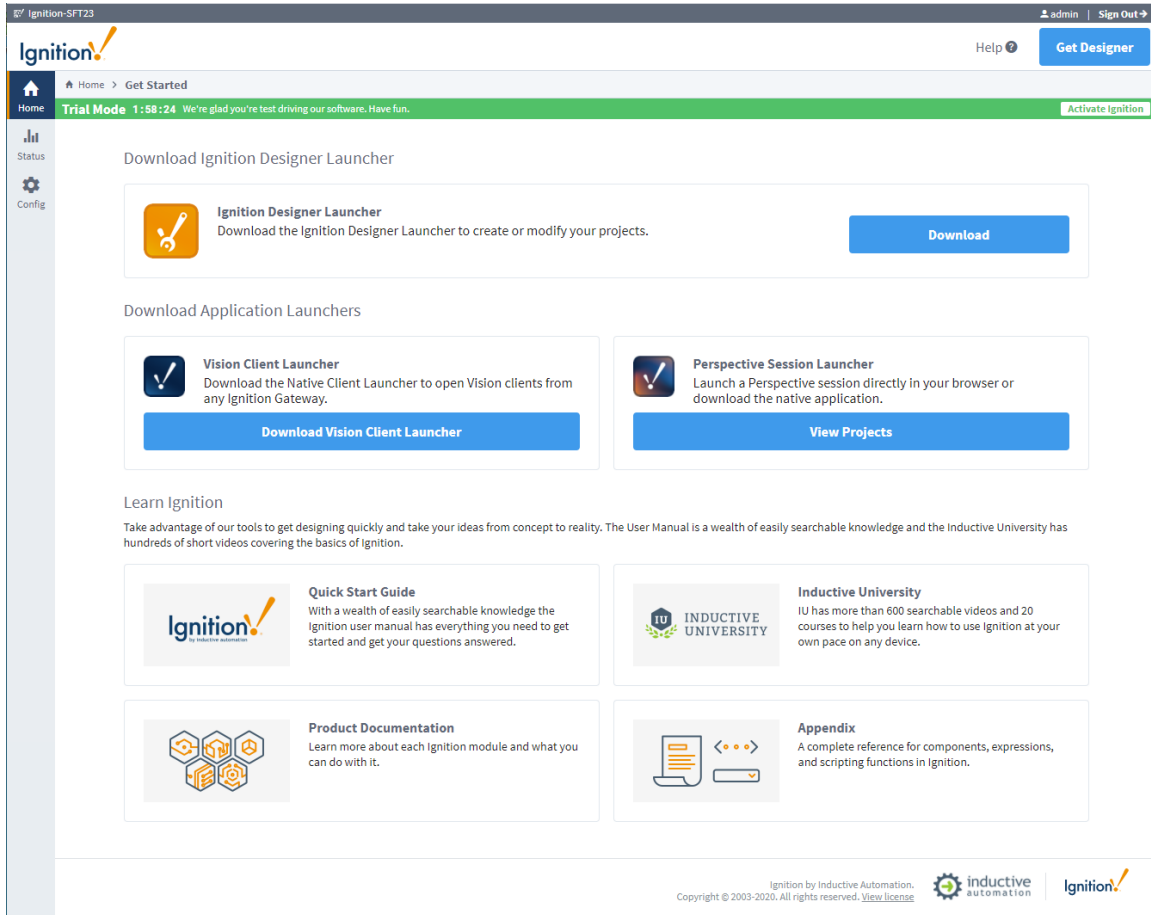


Figure 1. Ignition Gateway webpage

## Install MQTT Engine Ignition Module

Ignition uses the MQTT Engine module, a third-party module from Cirrus-Link, to connect to an MQTT broker and act as a Sparkplug SCADA host for Sparkplug Edge of Network (EoN) nodes publishing data to the broker.

[Download](#) and install the Cirrus-Link MQTT Engine Module for your version of Ignition, and follow the instructions in this [Ignition Module installation guide](#) to install the module. This [video walkthrough](#) may be helpful as well.

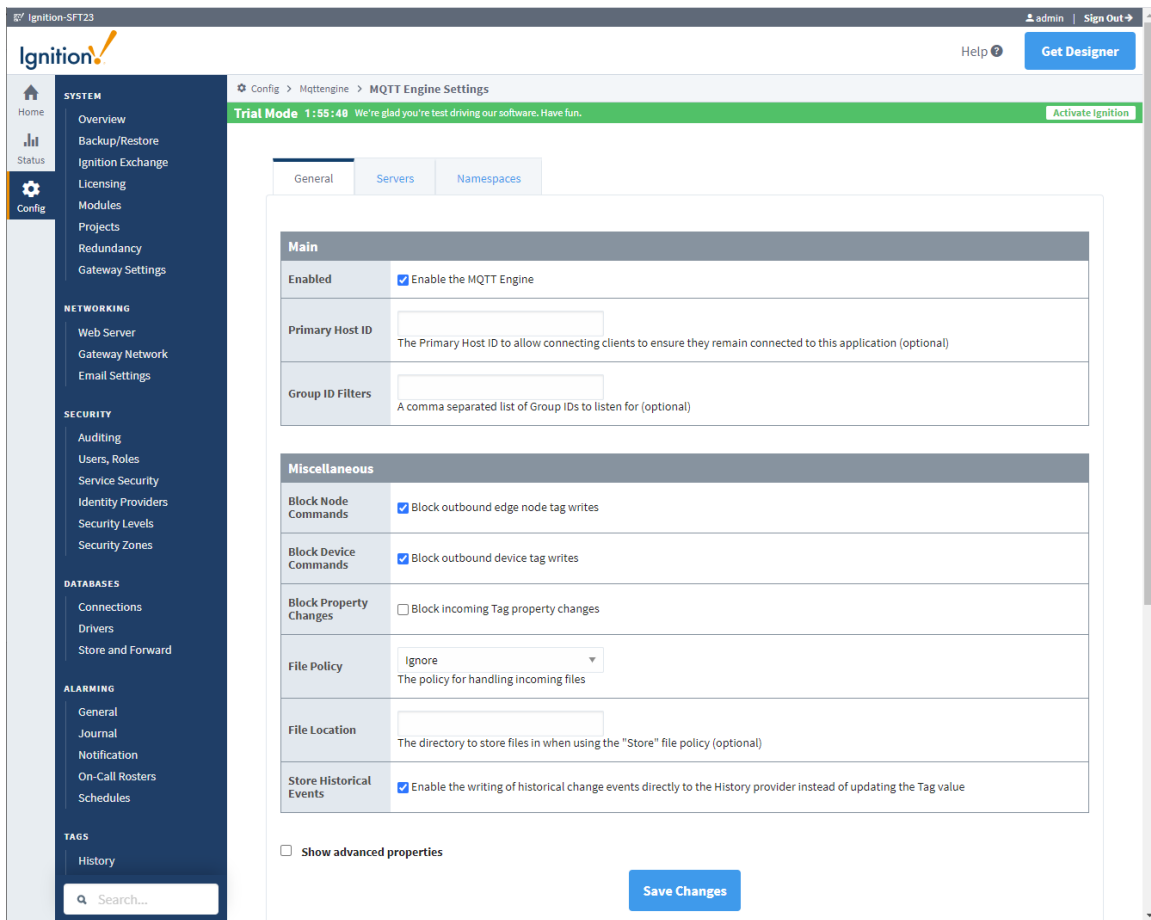


Figure 2. Ignition MQTT Engine Module Settings

## Configure MQTT Engine

The MQTT Engine Module must be configured with credentials to connect to your MQTT broker and start receiving data from Ranger nodes.

For this example, the following MQTT credentials will be used:

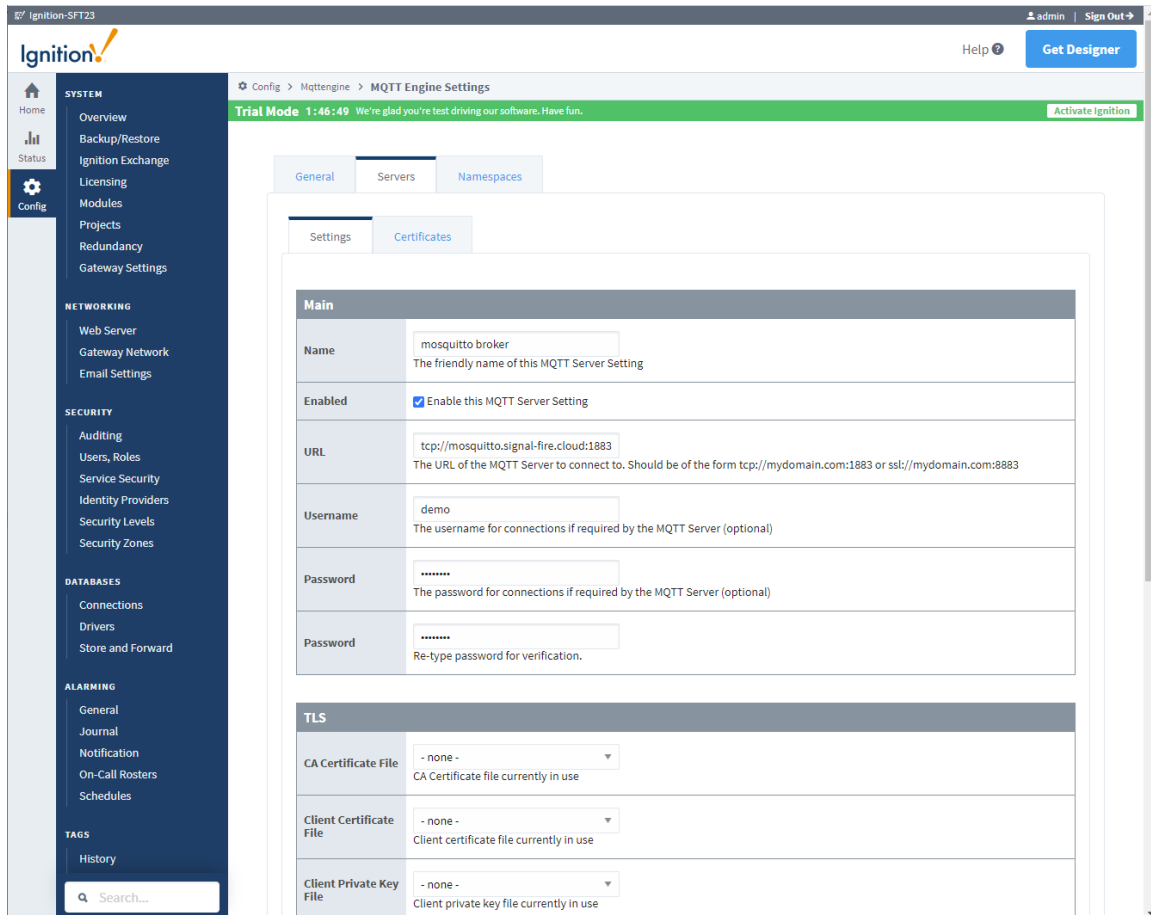
**Hostname:** mosquitto.signal-fire.cloud

**Port:** 1883

**Username:** demo

**Password:** password

First, select the “Servers” tab, and click “Create new MQTT Server Setting...”. Next, give this server connection a name, and enter the URL, Username, Password, and optional TLS credentials. Click the “Create New MQTT Server Setting” button at the bottom of the page to create



The screenshot shows the Ignition MQTT Engine Settings configuration page. The page is titled "MQTT Engine Settings" and has a breadcrumb trail: "Config > Mqttengine > MQTT Engine Settings". The page is in "Trial Mode" and shows a "Get Designer" button. The left sidebar contains a navigation menu with categories: SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, and TAGS. The main content area has tabs for "General", "Servers", and "Namespaces". The "Servers" tab is selected, and the "Settings" sub-tab is active. The form contains the following fields:

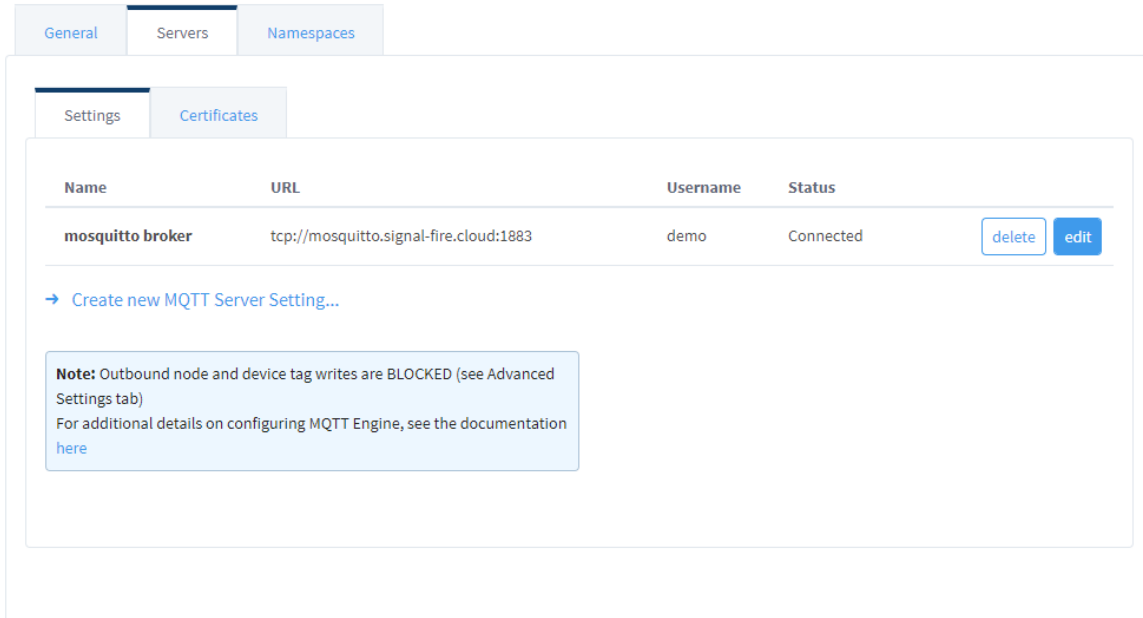
Main	
Name	mosquitto broker <small>The friendly name of this MQTT Server Setting</small>
Enabled	<input checked="" type="checkbox"/> Enable this MQTT Server Setting
URL	tcp://mosquitto.signal-fire.cloud:1883 <small>The URL of the MQTT Server to connect to. Should be of the form tcp://mydomain.com:1883 or ssl://mydomain.com:8883</small>
Username	demo <small>The username for connections if required by the MQTT Server (optional)</small>
Password	..... <small>The password for connections if required by the MQTT Server (optional)</small>
Password	..... <small>Re-type password for verification.</small>

TLS	
CA Certificate File	- none - <small>CA Certificate file currently in use</small>
Client Certificate File	- none - <small>Client certificate file currently in use</small>
Client Private Key File	- none - <small>Client private key file currently in use</small>

Figure 3. Ignition MQTT Server Settings

After adding the MQTT Server Setting, we can confirm that Ignition is connected to the MQTT broker.



The screenshot shows the Ignition configuration interface for MQTT servers. It has tabs for 'General', 'Servers', and 'Namespaces'. Under 'Servers', there are sub-tabs for 'Settings' and 'Certificates'. A table lists the configured MQTT servers:

Name	URL	Username	Status	
mosquitto broker	tcp://mosquitto.signal-fire.cloud:1883	demo	Connected	<input type="button" value="delete"/> <input type="button" value="edit"/>

Below the table is a link: [→ Create new MQTT Server Setting...](#)

A note box contains the following text: **Note:** Outbound node and device tag writes are BLOCKED (see Advanced Settings tab) For additional details on configuring MQTT Engine, see the documentation [here](#)

Figure 4. Ignition MQTT Server showing Status “Connected”

### MQTT Engine Configuration:

After installing the Cirrus Link MQTT Engine module on your Ignition server, and configuring the module to connect to your MQTT broker, there are a few important configuration settings that need to be considered for allowing the SignalFire Ranger to be added to Ignition’s tag provider as an Edge Node.

#### **Primary Host ID**

Sparkplug protocol allows for SCADA Hosts, such as Ignition, to publish their name, also known as their Host ID, as well as their Online or Offline state, so that Sparkplug EoN nodes, such as the SignalFire Ranger, can detect whether or not one or more SCADA hosts are online. The SignalFire Ranger requires, by default, at least one SCADA Host to be Online in order to publish data to the broker.

If this Ignition server is to be considered the Primary SCADA Host for the Sparkplug EoN nodes connected to the MQTT broker, then it is important to give the server a Primary Host ID. This is simply a name used for EoN nodes to identify this SCADA Host server from other servers, and must be unique across all SCADA Host servers connected to this MQTT broker. In this example, the Primary Host ID is set to “IgnitionTest”.

Main	
Enabled	<input checked="" type="checkbox"/> Enable the MQTT Engine
Primary Host ID	<input type="text" value="IgnitionTest"/> The Primary Host ID to allow connecting clients to ensure they remain connected to this application (optional)
Group ID Filters	<input type="text"/> A comma separated list of Group IDs to listen for (optional)

Figure 5. Ignition MQTT Engine Module set to Primary Host ID “IgnitionTest”

### **Unblock Node and Device Commands**

By default, Ignition’s MQTT Engine Module blocks all commands to Sparkplug EoN node and devices. This effectively makes this Ignition server read-only, without the ability to send commands and write tags on the remote nodes. To allow for such commands, uncheck the “Block outbound edge node tag writes” and “Block outbound device tag writes” checkboxes.

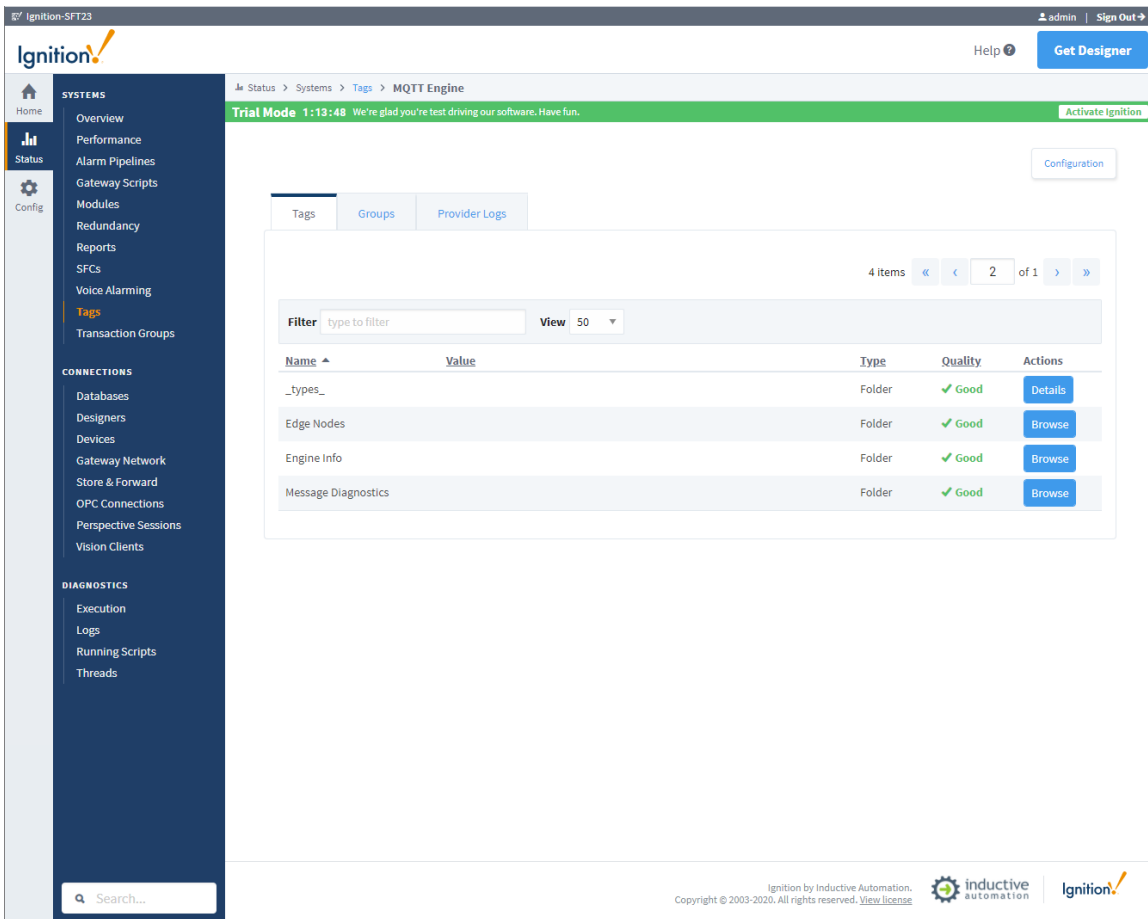
Miscellaneous	
Block Node Commands	<input type="checkbox"/> Block outbound edge node tag writes
Block Device Commands	<input type="checkbox"/> Block outbound device tag writes
Block Property Changes	<input type="checkbox"/> Block incoming Tag property changes
File Policy	<input type="text" value="Ignore"/> The policy for handling incoming files
File Location	<input type="text"/> The directory to store files in when using the "Store" file policy (optional)
Store Historical Events	<input checked="" type="checkbox"/> Enable the writing of historical change events directly to the History provider instead of updating the Tag value

Figure 6. Ignition MQTT Engine Module sett to allow outbound edge node and device tag writes

## Ignition Tags

Ignition’s MQTT Engine Module will automatically handle creating tags for the Sparkplug metrics being published to the MQTT broker by the Sparkplug EoN nodes, such as the SignalFire Ranger. It will also automatically publish tag write commands back to the respective Sparkplug EoN node when the tag is writing to by the Ignition server.

There are several ways to view Ignition’s tag database, including the tag browser built into the Ignition Gateway webpage available from the “Status” section. The tags created from Sparkplug EoN nodes’ published metrics are located in the “Edge Nodes” directory.



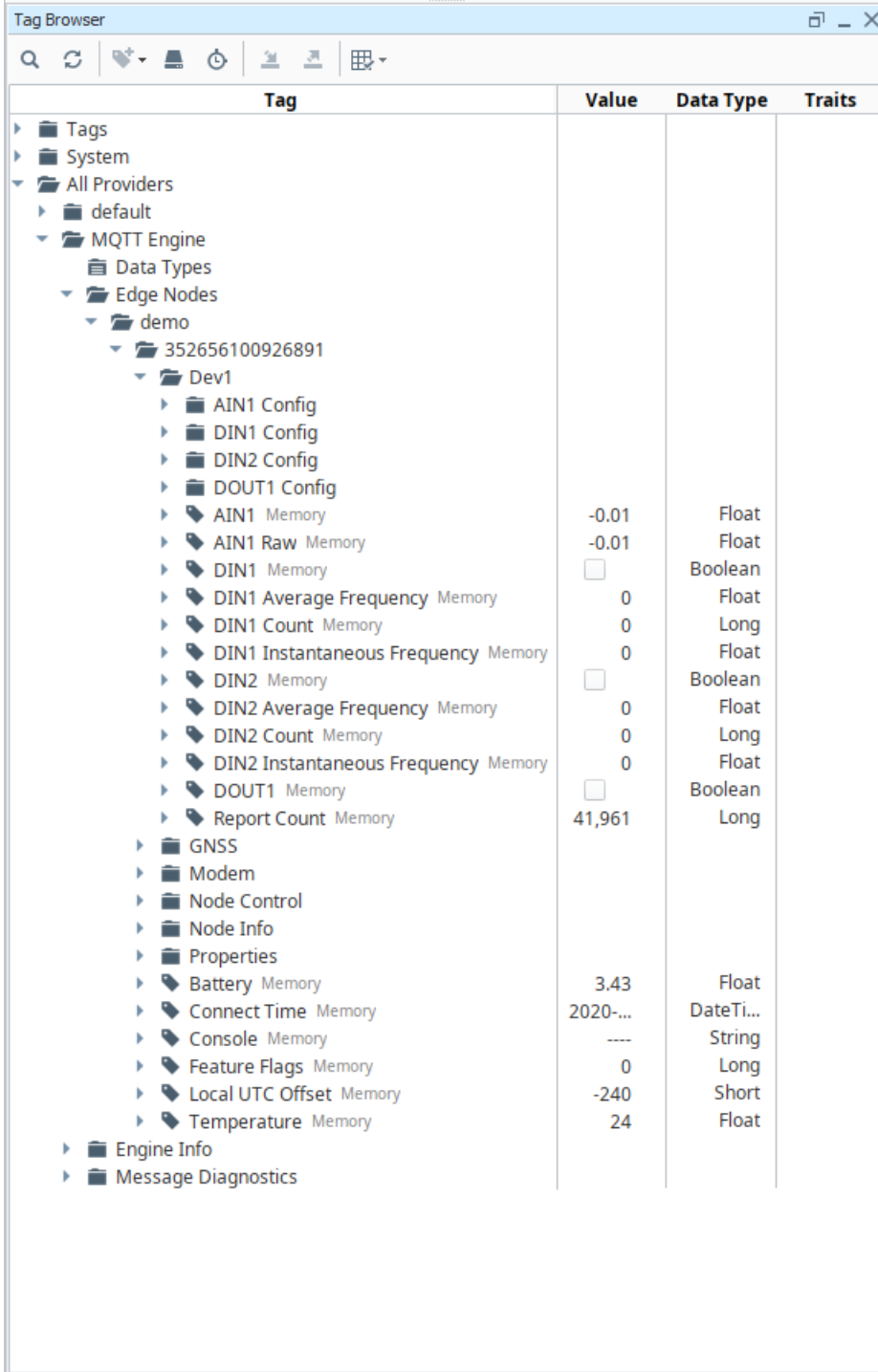
The screenshot shows the Ignition Gateway web interface for the MQTT Engine. The breadcrumb trail is Status > Systems > Tags > MQTT Engine. A green banner at the top indicates 'Trial Mode 1:13:48' and 'We're glad you're test driving our software. Have fun.' with an 'Activate Ignition' button. The left sidebar contains navigation menus for SYSTEMS, CONNECTIONS, and DIAGNOSTICS. The main area has tabs for 'Tags', 'Groups', and 'Provider Logs'. Below the tabs is a table with 4 items, showing a filter input and a 'View 50' dropdown. The table lists the following tags:

Name	Value	Type	Quality	Actions
_types_		Folder	Good	Details
Edge Nodes		Folder	Good	Browse
Engine Info		Folder	Good	Browse
Message Diagnostics		Folder	Good	Browse

At the bottom of the interface, there is a search bar and footer information: 'Ignition by Inductive Automation. Copyright © 2003-2020. All rights reserved. View license' and logos for Inductive Automation and Ignition.

Figure 7. Ignition Gateway webpage tag browser

The Ignition Designer also allows for tag browsing using an expandable tree view.



Tag	Value	Data Type	Traits
Tags			
System			
All Providers			
default			
MQTT Engine			
Data Types			
Edge Nodes			
demo			
352656100926891			
Dev1			
AIN1 Config			
DIN1 Config			
DIN2 Config			
DOUT1 Config			
AIN1 Memory	-0.01	Float	
AIN1 Raw Memory	-0.01	Float	
DIN1 Memory	<input type="checkbox"/>	Boolean	
DIN1 Average Frequency Memory	0	Float	
DIN1 Count Memory	0	Long	
DIN1 Instantaneous Frequency Memory	0	Float	
DIN2 Memory	<input type="checkbox"/>	Boolean	
DIN2 Average Frequency Memory	0	Float	
DIN2 Count Memory	0	Long	
DIN2 Instantaneous Frequency Memory	0	Float	
DOUT1 Memory	<input type="checkbox"/>	Boolean	
Report Count Memory	41,961	Long	
GNSS			
Modem			
Node Control			
Node Info			
Properties			
Battery Memory	3.43	Float	
Connect Time Memory	2020-...	DateTi...	
Console Memory	----	String	
Feature Flags Memory	0	Long	
Local UTC Offset Memory	-240	Short	
Temperature Memory	24	Float	
Engine Info			
Message Diagnostics			

Figure 8. Ignition Designer Tag Browser.