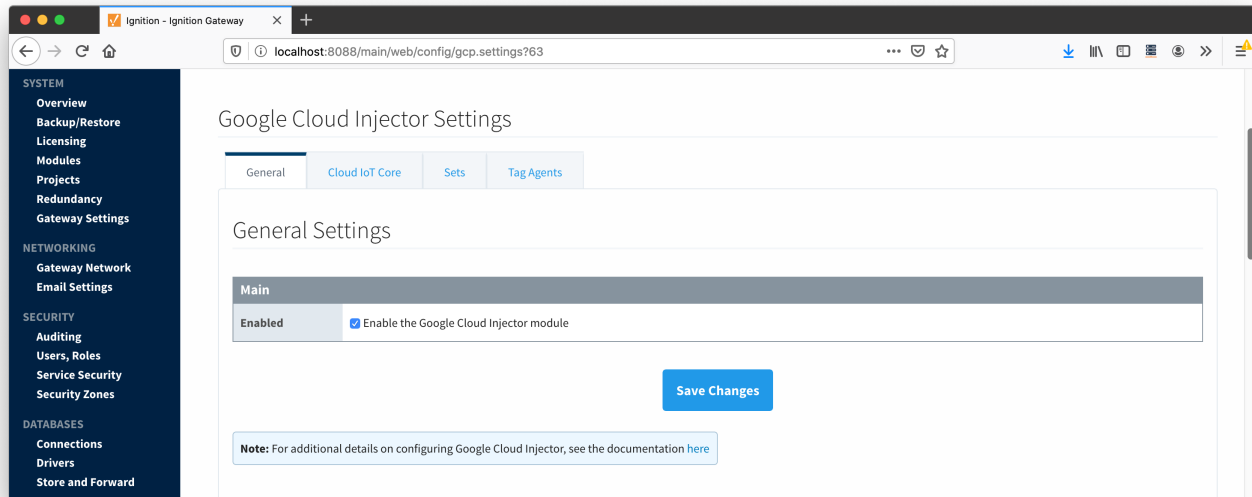


GCI: Configuration

The Google Cloud Injector module provides the ability to push Tag data to a Cloud IoT Core. The settings configuration for this module are located under the Configure tab of the Ignition Gateway web UI. Once in the configuration section there are four tabs: General, Cloud IoT Core, Sets, and Tag Agents. Each of these tabs is described in detail in the following sections.

General

The first tab contains the general settings for the Google Cloud Injector module.



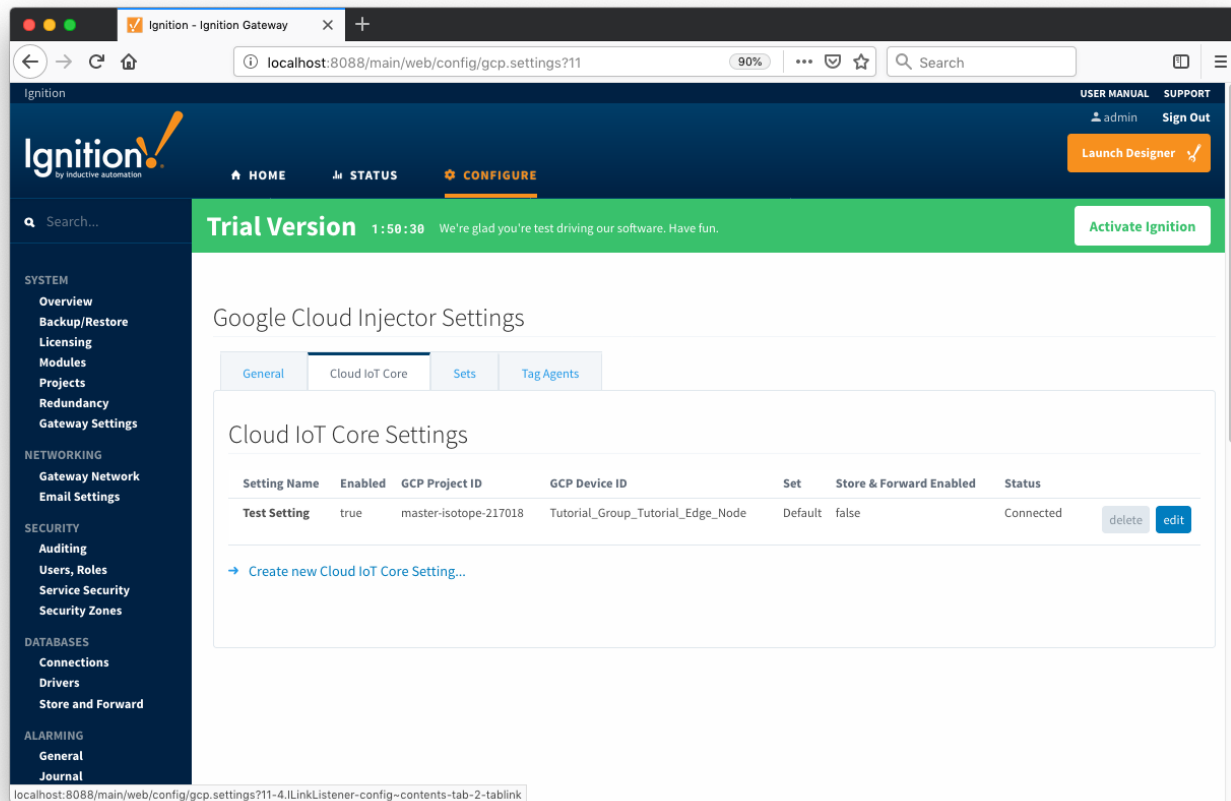
The general configurations options available on this tab are listed below:

Main

- **Enabled**
 - Sets whether the module is enabled or disabled. If disabled, the Tag Agents will not run and now data will be pushed to any configured endpoints.

Cloud IoT Core

The next tab is the list of Cloud IoT Core endpoints that the module should connect to to push tag data.



One or more Cloud IoT Core endpoints can be configured on this tab. The configuration options for a Cloud IoT Core connection are listed below.

Main

- **Setting Name**
 - This is a friendly name of the Cloud IoT Core used to easily identify it. This must also be unique.
- **Enabled**
 - Whether or not this Injector Setting is enabled to push data to the endpoint
- **GCP Project ID**
 - The Google Cloud Platform project under which this device is provisioned
- **GCP Cloud Region**
 - The Google Cloud Platform cloud region under which this device is provisioned
- **GCP Registry ID**
 - The Google Cloud Platform registry ID under which this device is provisioned
- **GCP Device ID**
 - The Google Cloud Platform device ID under which this device is provisioned
- **Private Key File**
 - The private key file associated with the public key provisioned for the device in Google Cloud Platform
- **Algorithm**
 - The algorithm type used by the key file
- **Set**
 - The Set this Cloud IoT Core Setting is associated with

Store & Forward

- **Store & Forward Enabled**
 - Whether to enable Store & Forward capabilities for this endpoint
- **Store & Forward Type**
 - The type of the Store & Forward mechanism
- **Message Capacity**
 - The Maximum number of messages to store before dropping the oldest historical messages
- **Flush Period**
 - The period of time to wait (in milliseconds) between sending when flushing messages

Advanced

- **Keep Alive**
 - The MQTT Keep Alive timeout for the MQTT Connection
- **Max Message Size**
 - The maximum message size allowed for an MQTT message before the message will be broken up into smaller messages

Clicking on the "Create new Cloud IoT Core Setting..." link will bring up the following form for adding a new Cloud IoT Core endpoint.

The screenshot shows the Ignition Gateway web interface at localhost:8088/main/web/config/gcp.settings?12. The left sidebar contains a navigation menu with categories: SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, TAGS, OPC-UA SERVER, OPC CONNECTIONS, MOBILE, ENTERPRISE ADMINISTRATION, SEQUENTIAL FUNCTION CHARTS, and GOOGLE CLOUD INJECTOR. The main content area is titled 'Google Cloud Injector Settings' and has tabs for 'General', 'Cloud IoT Core', 'Sets', and 'Tag Agents'. The 'Cloud IoT Core' tab is active, showing a 'New Cloud IoT Core Setting' form.

Main

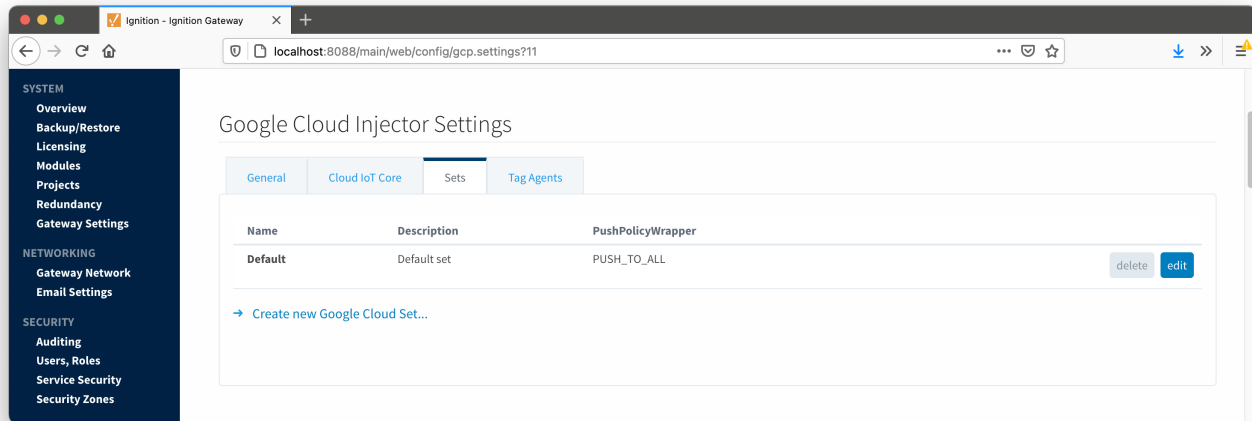
Setting Name		A friendly name for this Cloud IoT Core setting
Enabled	☒ Enable this setting	
GCP Project ID		The GCP project under which this device is provisioned
GCP Cloud Region		The GCP cloud region under which this device is provisioned
GCP Registry ID		The GCP registry ID under which this device is provisioned
GCP Device ID		The GCP ID of this device
Private Key File	Browse... No file selected.	The private key file associated with the public key provisioned for the device in GCP
Algorithm	Choose One	The algorithm type used by the key file
Set	Default	The Set this Cloud IoT Core Setting is associated with

Store & Forward

Store & Forward Enabled	☐ Enable Store and Forward capabilities for this stream
Store & Forward Type	Choose One The Type of this Store & Forward mechanism
Message Capacity	10000 Maximum number of messages to store before dropping oldest historical messages

Sets

The Sets tab contains a list of Google Cloud Sets. Each set represents a grouping of Cloud IoT Core endpoints. When a set is referenced by a Tag Agent the Agent will push Tag data to all Cloud IoT Core endpoints contained within that Set. The Sets are disjoint, meaning that a single Cloud IoT Core endpoint cannot be in more than one set. Out of the box the Google Cloud Injector module will have one "Default" set defined.



Additional Sets can be configured for situations where multiple Tag Agents will need to push to different Cloud IoT Core endpoints. The configuration options for Sets are listed below.

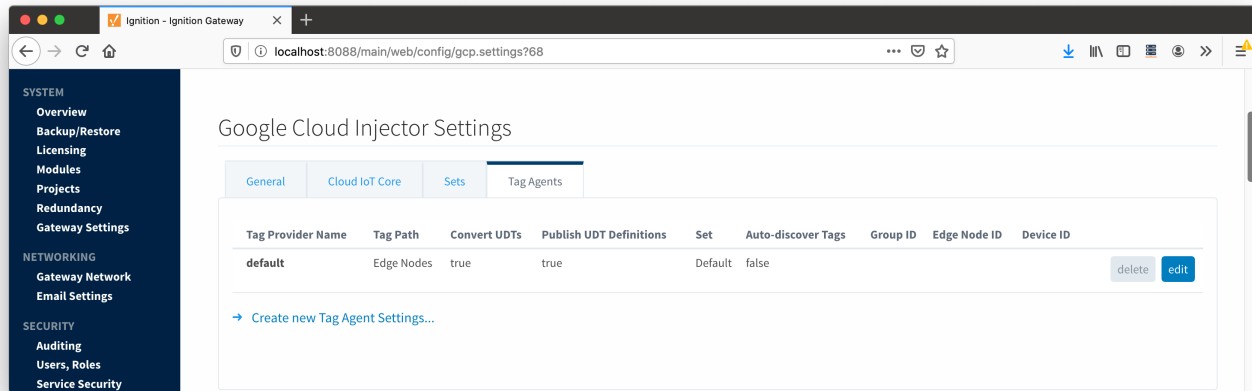
Main

- **Name**
 - This is the friendly name of the set used to easily identify it.
- **Description**
 - This is a friendly description of the set.
- **Push Policy**
 - This defines which endpoints to push to. If PUSH_TO_ALL is selected, every endpoint that is part of this set will receive all messages. If PUSH_TO_ANY is selected, only one of the endpoints that is part of this set will receive any given message. PUSH_TO_ANY is useful when adding endpoint configurations to increase the throughput of the Injector.

Clicking on the 'Create new Google Cloud Set...' link will bring up the following form to add a new Set.

Tag Agents

Tag Agents are the workers within Google Cloud Injector that monitor tag events, convert them to a JSON representation, and push them to one or more Cloud IoT Core endpoints. Out of the box the Google Cloud Injector module will have one "default" Tag Agent defined.



Tag Agents are configured to point to a single folder. All Tags within that folder will be monitored by the Tag Agent.

Agent Settings

- **Tag Provider Name**
 - The name of the Tag provider containing the tags.
- **Tag Path**
 - An optional folder path under the Tag provider where the root folder of the Tags can be found.
- **Tag Pacing Period**
 - The buffer period, in milliseconds, that Tag events will be aggregated into a single payload before pushing.
- **Convert UDTs**
 - Whether to convert UDT members to normal Tags before publishing. If enabled the Tags representing the UDT member will retain their member path prefixed by the UDT Instance name.
 - **Publish UDT Definitions**
 - This can only be set if 'Convert UDTs' is false
 - Whether or not to publish UDT definitions in the NBIRTH messages
- **Set**
 - The Set of Cloud IoT Core endpoints that the Tag Agent will push to.
- **Auto-discover Tags**
 - Whether newly added tags should be dynamically scanned and their values pushed. This field is disabled by default. It should remain disabled while manually editing tags and/or their configurations. It should only typically be enabled in systems where tags are created in real time.

Sparkplug Settings

- **Group ID**
 - An ID representing a logical grouping of MQTT Edge Of Network (EoN) Nodes and Devices into the infrastructure.
- **Edge Node ID**
 - An ID that uniquely identifies the MQTT Edge Of Network (EoN) Node within the infrastructure.
- **Device ID**
 - An optional ID that uniquely identifies a Device within the infrastructure.

The Sparkplug settings are optional and allow for an additional customization of how the Tag Agent scans and discovers tag within the specified Tag Path. Here is a brief description of how the Agent scans/discovers folders based on the different combinations of potential Sparkplug Settings.

- If all three IDs are left blank the Agent will assume the following folder structures follow the Tag Path:
 - `<groupFolder>/<edgeNodeFolder>/<deviceFolder>/<tags>`
 - `<groupFolder>/<edgeNodeFolder>/<tags>`
- If only the Group ID is specified the Agent will assume the following folder structure follows the Tag Path:
 - `<edgeNodeFolder>/<deviceFolder>/<tags>`
 - `<edgeNodeFolder>/<tags>`
- If the Group ID and the Edge Node ID are specified the Agent will assume the following folder structure follows the Tag Path:
 - `<deviceFolder>/<tags>`
 - `<tags>`
- If the Group ID, Edge Node ID, and the Device ID are specified the Agent will assume the following folder structure follows the Tag Path:
 - `<tags>`

As you can see, the Sparkplug settings can be defined by setting these IDs in this configuration page, or leave them blank so that the Agent will scan and discover them based on the tag tree layout.

Clicking on the 'Create new Tag Agent Settings..' link will bring up the following form to add a new Tag Agent.

