

# AWI: Configuration

The AWS Injector module provides the ability to push Tag data to an Kinesis Streams and DynamoDB Database endpoints. The settings configuration for this module are located under the Configure tab of the Ignition Gateway web UI.

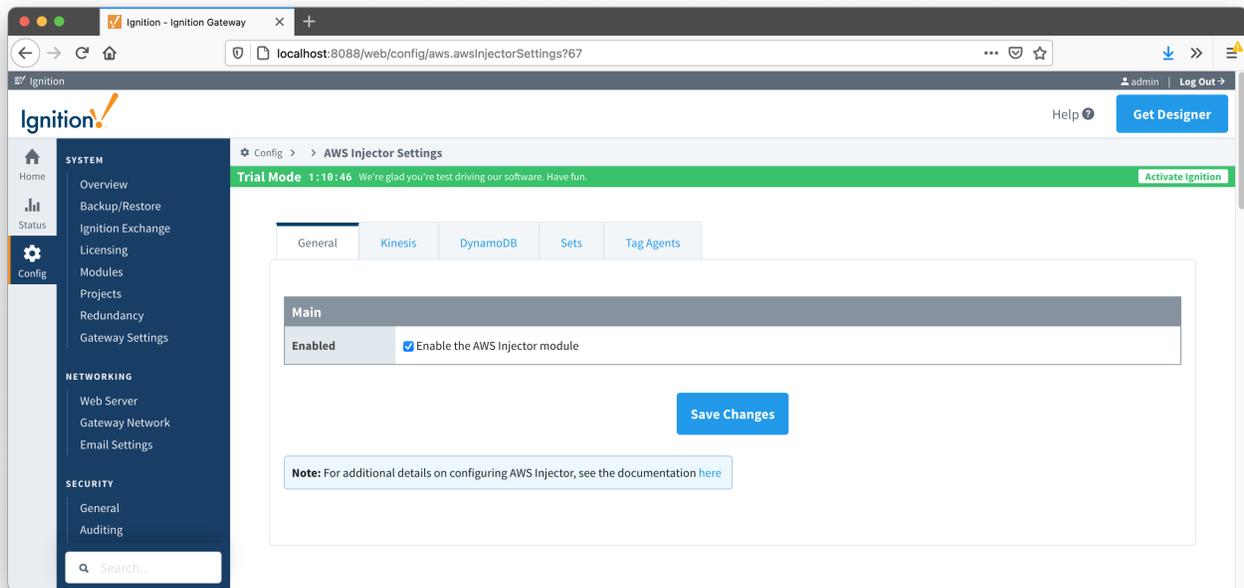
Once in the AWS Injector Settings configuration section there are five tabs: [General](#), [Kinesis](#), [DynamoDB](#), [Sets](#), and [Tag Agents](#).

- ✓ Prior to release 4.0.22, Disk-Backed History stores are located here `~yourIgnitionInstance\user-lib\cls\data\h2`
- From release 4.0.22, Disk-Backed History store are located here `~yourIgnitionInstance\data\modules\com.cirrus-link\com.cirruslink.injector.aws.gateway\h2`
- From release 4.0.25, the base path for the database location of the Disk-Backed History store is configurable.
  - The default base path for Linux is `~yourIgnitionInstance\data\modules` and the database will be included in the Ignition GWBK
  - The default location for Windows Linux is `~yourIgnitionInstance\user-lib\modules` and the database will not be included in the Ignition GWBK
  - The database file will be created in this directory under the base path `com.cirrus-link\com.cirruslink.injector.aws.gateway\h2`

Each of these tabs is described in detail in the following sections.

## General

The first tab contains the general settings for the AWS Injector module.



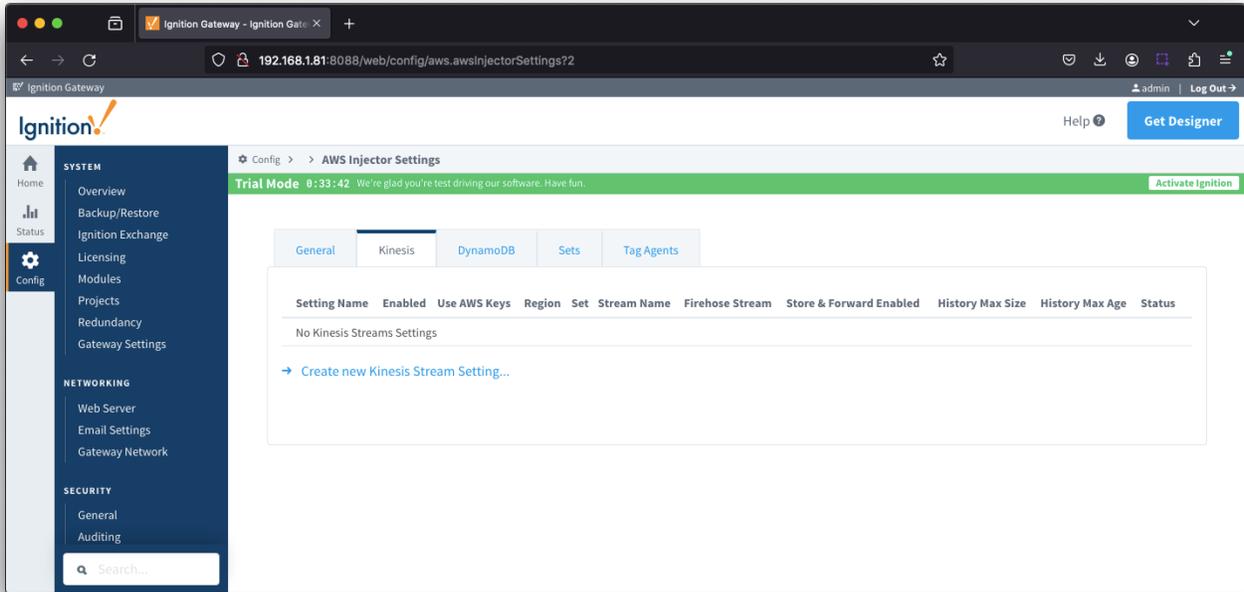
The general configurations options available on this tab are listed below:

### Main

- **Enabled**
  - Sets whether the module is enabled or disabled. If disabled, the Tag Agents will not run and no data will be pushed to any configured endpoints.

## Kinesis Streams

The next tab is the list of AWS Kinesis Stream endpoints that the module should connect to to push Tag data.



One or more AWS Kinesis Stream endpoints can be configured on this tab. The configuration options for an AWS Kinesis Stream connection are listed below.

## Main

- **Setting Name**
  - This is a friendly name of the AWS Kinesis Stream used to easily identify it. This must also be unique.
- **Enabled**
  - Whether to enable/disable this setting
- **Use AWS Keys (added 4.0.22)**
  - Whether to use AWS access and secret keys or use an AWS EC2 instance with an AWS IAM role
- **AWS Access Key**
  - The AWS Access Key ID.
  - Example: AKIAIOSFODNN7EXAMPLE
  - Leave blank if 'Use AWS Keys' is unchecked
- **Password/AWS Secret Key**
  - The AWS Secret Access Key.
  - Example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  - Leave blank if 'Use AWS Keys' is unchecked
- **AWS Session Tokens**
  - Whether to use AWS session tokens
- **Role ARN**
  - The AWS role ARN
- **Session Duration**
  - The AWS session duration
- **Session Name**
  - The AWS session name
- **Region**
  - The AWS region.
- **Set**
  - The Set to associate this AWS Kinesis Stream connection with.
- **Stream Name**
  - The name of the AWS Kinesis Stream.
- **Firehose Stream**
  - Whether this setting is for a Kinesis Data Firehose delivery stream

 If using a Firehose stream, its source must be "Direct PUT or other sources"

## Store & Forward

- **Store & Forward Enabled**
  - Whether to enable Store & Forward capabilities for this stream

- **Store & Forward Type**
  - The type of the Store & Forward mechanism options: In\_Memory and Disk\_Backed (available in release 4.0.17 and higher)
  - Data stored with an In\_Memory Store & Forward will not be persisted across a module configuration change, module disable/enable, module restart or power loss
  - Data stored with a Disk\_Backed Store & Forward will persist across a module configuration change, module disable/enable, module restart or power loss
- **Message Capacity - deprecated in 4.0.19**
  - The maximum number of messages to store before dropping the oldest historical messages
- **History Max Size**
  - The maximum number of megabytes history can use before dropping the data
  - In\_Memory Store & Forward will use the Ignition Java Heap memory
- **Max History Age**
  - The maximum number of minutes to store history before dropping the data
- **Flush Period**
  - The period of time to wait (in milliseconds) between sending when flushing messages

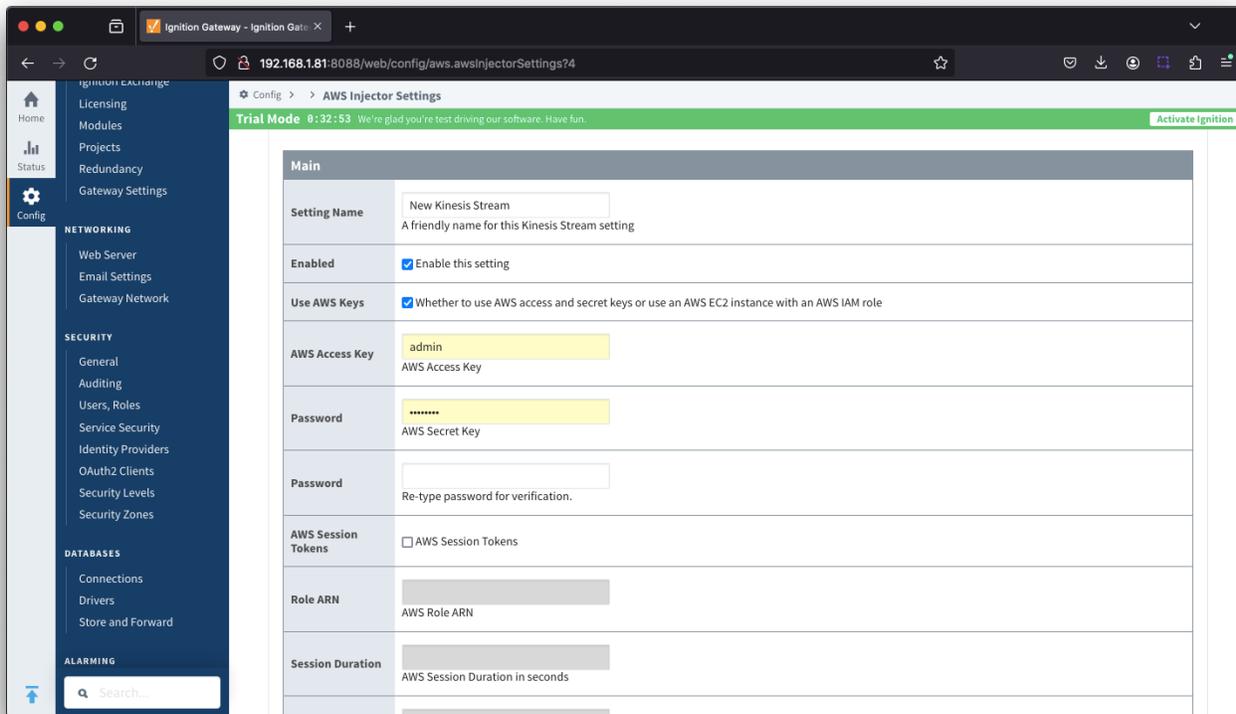
## Advanced

- **Partition Key Count**
  - The number of partition keys to use for inserts
- **Partition Key**
  - An optional partition key pattern that supports token substitution for the following message fields: \${group}, \${edgenode}, \${device}. This should only be configured if explicitly defined partition keys are necessary when inserting into the Kinesis Stream.
- **Max Message Size**
  - The maximum message size in bytes to send. Generally, this should match the max message size allowed by AWS Kinesis.

## Advanced Store & Forward

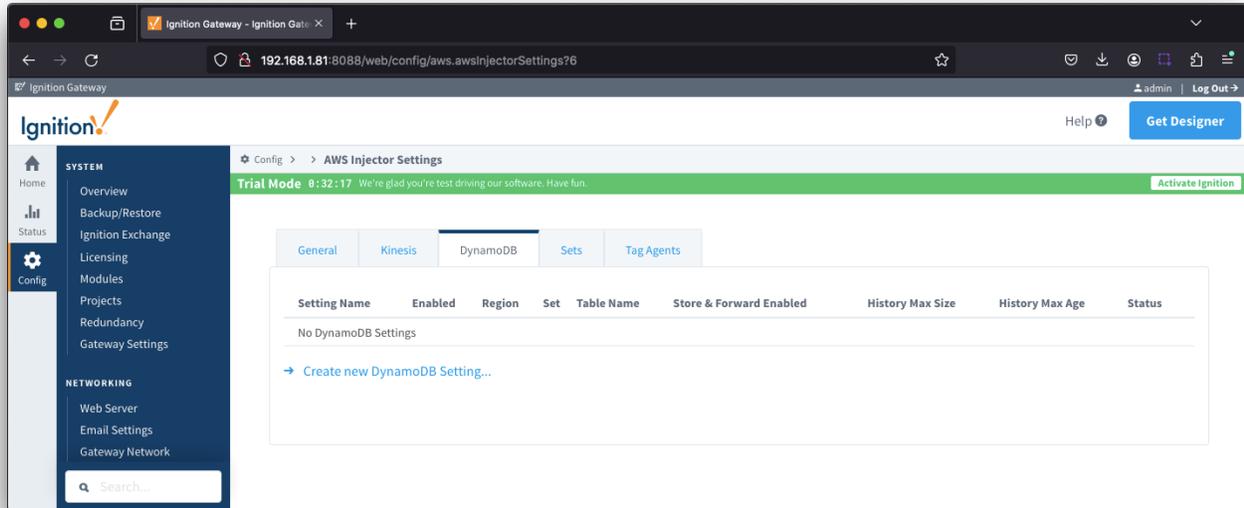
- **H2 Database Directory - added in 4.0.25**
  - Directory to store the H2 Database in. Applicable for Disk-backed history store only
  - The default base path for Linux is `~yourIgnitionInstance\data\modules` and the database will be included in the Ignition GWBK
  - The default location for Windows Linux is `~yourIgnitionInstance\user-lib\modules` and the database will not be included in the Ignition GWBK
  - The database file will be created in this directory under the base path `com.cirrus-link\com.cirruslink.injector.aws.gateway\h2`
- **H2 Database Port**
  - TCP Port to connect to H2 Database for Disk\_Backed Store & Forward

Clicking on the "Create new Kinesis Stream Setting..." link will bring up the following form for adding a new AWS Kinesis Stream endpoint.



DynamoDB Databases

The third tab is the list of AWS DynamoDB Database endpoints that the module should connect to to push Tag data.



One or more AWS DynamoDB Database endpoints can be configured on this tab. The configuration options for an AWS DynamoDB Database connection are listed below.

## Main

- **Setting Name**
  - This is a friendly name of the AWS Kinesis Stream used to easily identify it. This must also be unique.
- **Enabled**
  - Whether to enable/disable this setting
- **AWS Access Key**
  - The AWS Access Key ID.
  - Example: AKIAIOSFODNN7EXAMPLE
- **Password/AWS Secret Key**
  - The AWS Secret Access Key.
  - Example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
- **Region**
  - The AWS region.
- **Set**
  - The Set to associate this AWS DynamoDB Database connection with.
- **Table Name**
  - The name of the AWS DynamoDB Database table.

## Store & Forward

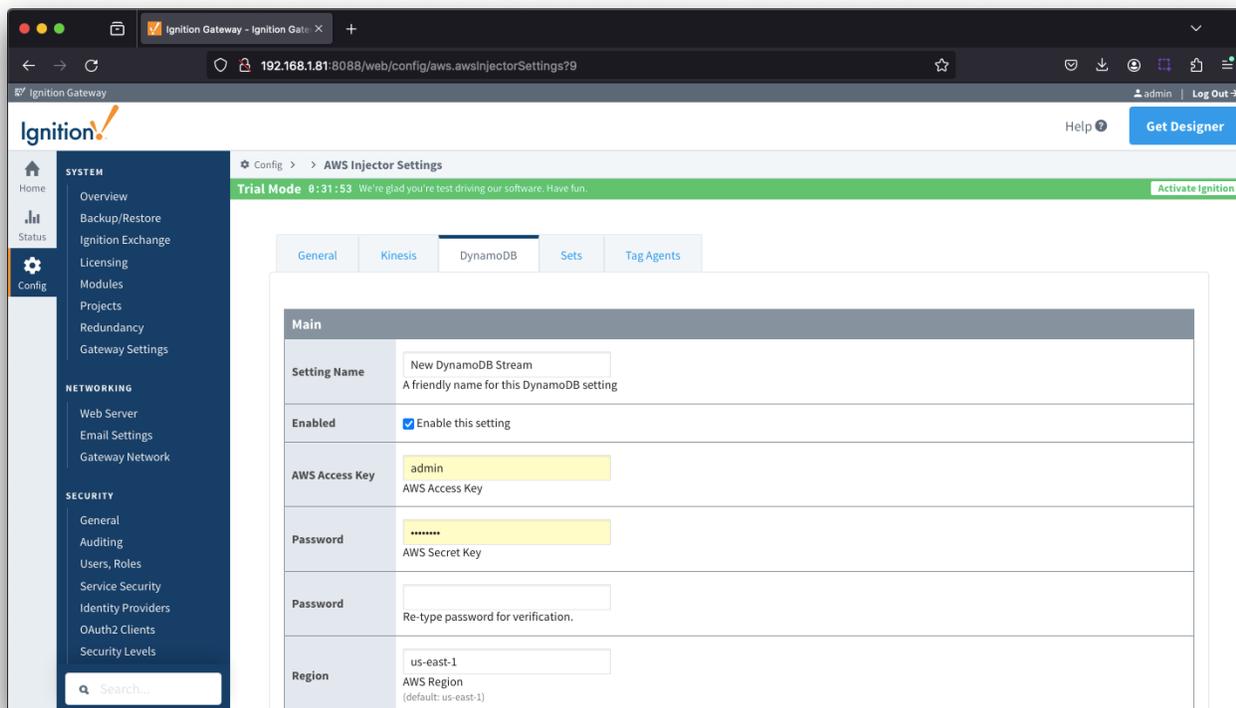
- **Store & Forward Enabled**
  - Whether to enable Store & Forward capabilities for this stream
- **Store & Forward Type**
  - The type of the Store & Forward mechanism options: In\_Memory and Disk\_Backed (available in release 4.0.17 and higher)
  - Data stored with an In\_Memory Store & Forward will not be persisted across a module configuration change, module disable/enable, module restart or power loss.
  - Data stored with a Disk\_Backed Store & Forward will persist across a module configuration change, module disable/enable, module restart or power loss.
- **Message Capacity - deprecated in 4.0.19**
  - The maximum number of messages to store before dropping the oldest historical messages
- **History Max Size**
  - The maximum number of megabytes history can use before dropping the data
  - In\_Memory Store & Forward will use the Ignition Java Heap memory
- **History Max Age**
  - The maximum number of minutes to store history before dropping the data
- **Flush Period**
  - The period of time to wait (in milliseconds) between sending when flushing messages

## Advanced Store & Forward

- **H2 Database Directory - added in 4.0.25**
  - Directory to store the H2 Database in. Applicable for Disk-backed history store only
  - The default base path for Linux is `~yourIgnitionInstance\data/modules` and the database will be included in the Ignition GWBK

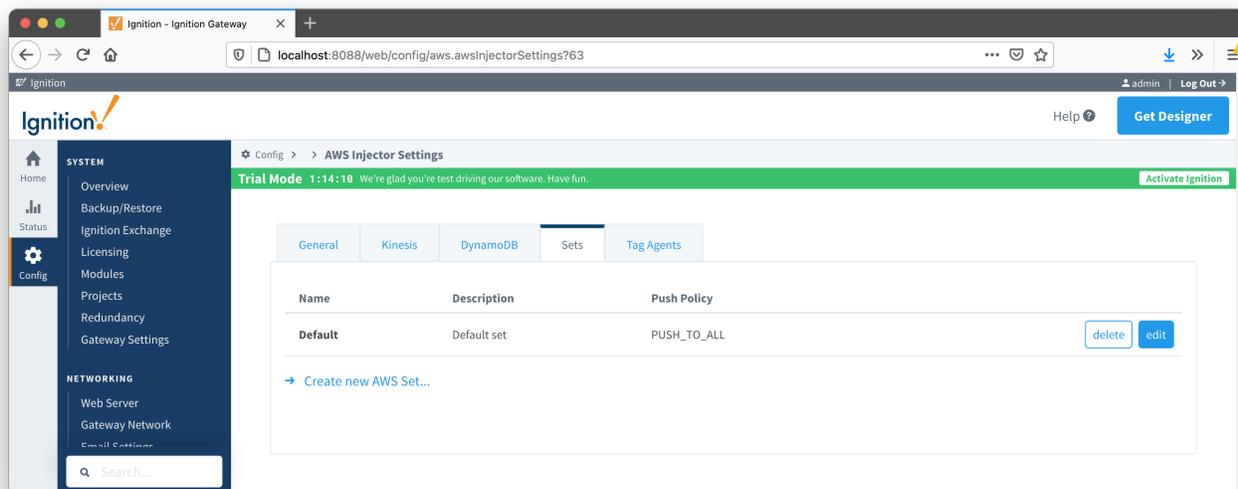
- The default location for Windows Linux is `~\your\IgnitionInstance\user-lib\modules` and the database will not be included in the Ignition GWBK
- The database file will be created in this directory under the base path `com.cirrus-link\com.cirruslink.injector.aws.gateway\h2`
- **H2 Database Port**
  - TCP Port to connect to H2 Database for Disk\_Backed Store & Forward

Clicking on the "Create new DynamoDB Setting..." link will bring up the following form for adding a new AWS DynamoDB Database endpoint.



## Sets

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

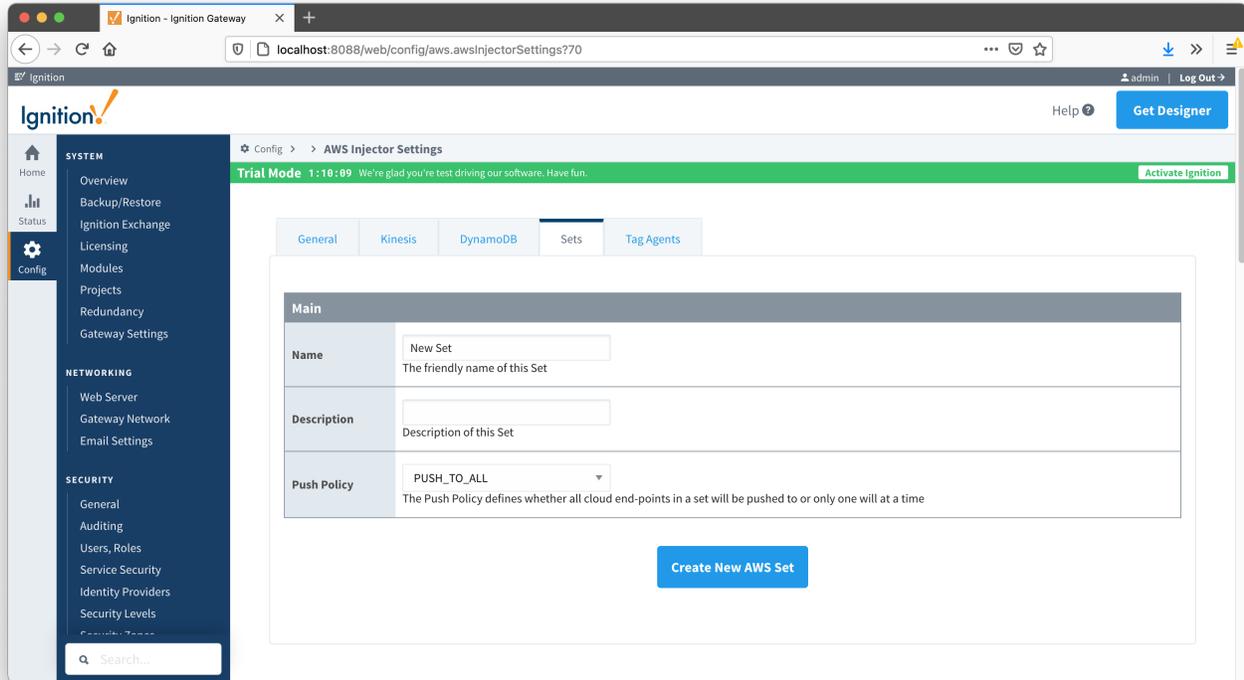


Additional Sets can be configured for situations where multiple Tag Agents will need to push to different AWS endpoints. The configuration options for Sets are listed below.

## 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 AWS Set...' link will bring up the following form to add a new Set.



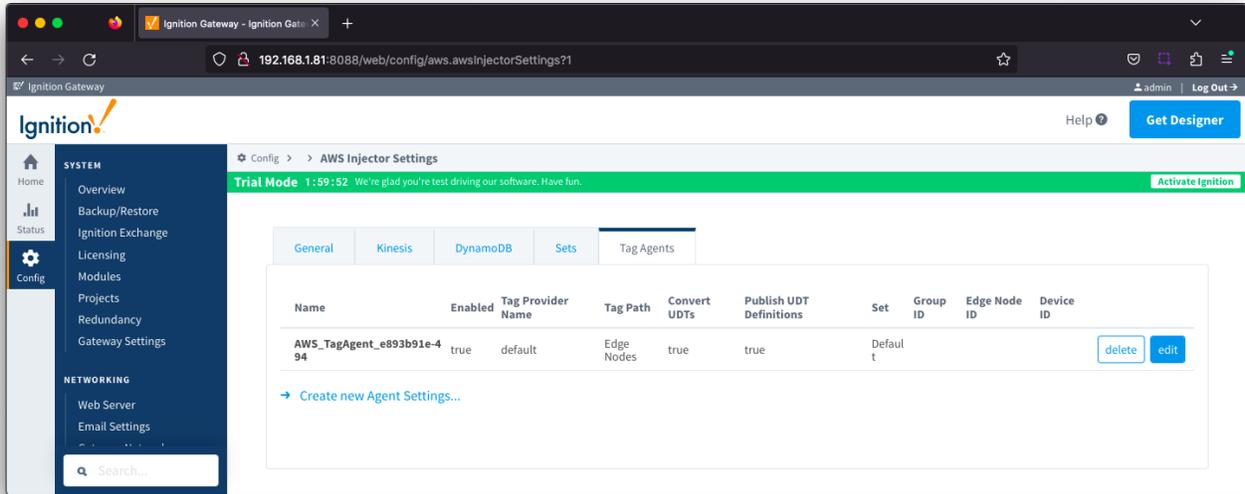
The screenshot shows the Ignition Gateway web interface. The browser address bar indicates the URL is localhost:8088/web/config/aws.injectorSettings?70. The page title is 'AWS Injector Settings'. A green banner at the top indicates 'Trial Mode 1:10:09' and includes a 'Get Designer' button. The main content area is titled 'Main' and contains a form with the following fields:

- Name:** A text input field containing 'New Set'. Below the input is the label 'The friendly name of this Set'.
- Description:** A text input field. Below the input is the label 'Description of this Set'.
- Push Policy:** A dropdown menu with 'PUSH\_TO\_ALL' selected. Below the dropdown is the label 'The Push Policy defines whether all cloud end-points in a set will be pushed to or only one will at a time'.

At the bottom of the form is a blue button labeled 'Create New AWS Set'.

## Tag Agents

Tag Agents are the workers within AWS Injector that monitor tag events, convert them to a [JSON representation](#), and push them to one or more AWS endpoints. Out of the box the AWS 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.

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

## Tag Settings

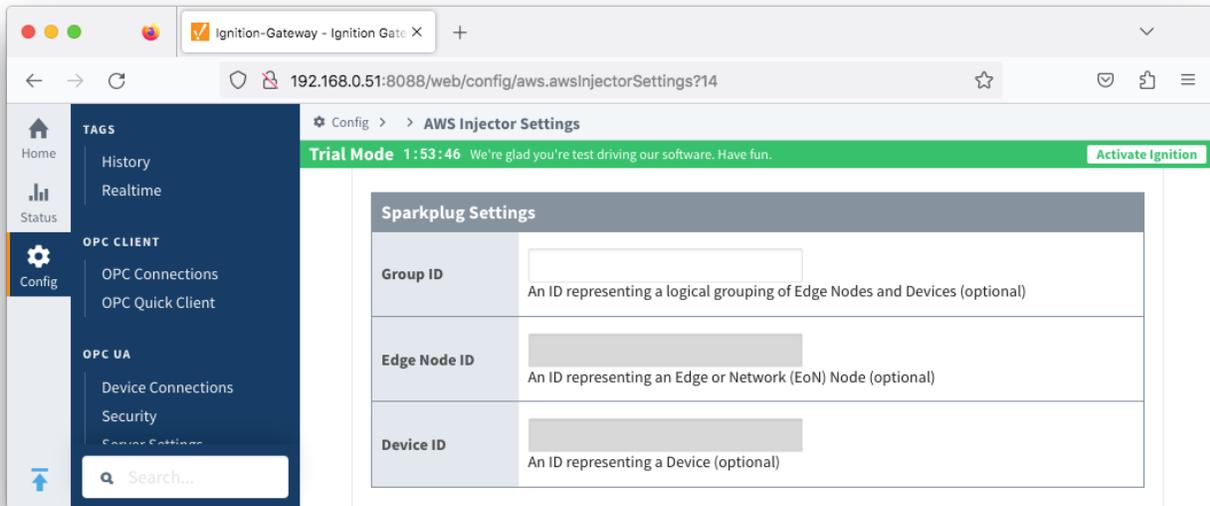
The screenshot shows the 'AWS Injector Settings' configuration page in the Ignition Gateway. The page is in 'Trial Mode' and features a sidebar with navigation options. The main content area is titled 'Agent Settings' and contains a form with the following fields:

- Name:** A unique name for the Tag Agent
- Enabled:**  Enable Tag Agent (default: true)
- Tag Provider Name:** The Name of the tag provider
- Tag Path:** A path to the root folder where the tag tree starts (optional)
- Push Trigger:** EVENT\_DRIVEN (default). The trigger to use when pushing data. EVENT\_DRIVEN (default) means to push on change events. PERIODIC means to push all data on a periodic basis. HYBRID will push change events as EVENT\_DRIVEN does but also push non-changed tags on the same tag pacing period.
- Tag Pacing Period:** 1000. If using an EVENT\_DRIVEN Push trigger, the waiting period in milliseconds after an initial tag change event before pushing all changed tags. If using PERIODIC, the number of milliseconds to wait between pushing all data. (default: 1,000)
- Convert UDTs:**  Converts UDT members to normal Tags
- Publish UDT Definitions:**  Publish UDT Definitions in BIRTH
- Optimize UDTs:**  Optimizes UDT payload sizes in NDATA and DDATA payloads
- Set:** Default. The Set this Agent is associated with

- **Name**
  - A unique name for the tag agent.
- **Enabled**
  - Sets whether the Tag Agent is enabled or disabled. If disabled, the Tag Agent will not run and no data will be pushed to any configured endpoints.
- **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.
- **Push Trigger**
  - Defines what triggers a push to the cloud endpoint
    - EVENT\_DRIVEN (default) - when a tag change event occurs, and no pending push exists, tag events will be aggregated for the 'Tag Pacing Period' before being pushed
    - PERIODIC - will push current data value for all tags associated with the Agent every 'Tag Pacing Period'.
      - With this option, only one event per tag will be sent - this may result in data loss if tags are changing more frequently than the 'tag pacing period'
    - HYBRID - will push change events as EVENT\_DRIVEN but will also push non changed events on the same tag pacing period.
  - Review [How the MQTT Injector Push Trigger affects data published](#) for examples
- **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 push the UDT Definitions in the the NBIRTH messages
- **Optimize UDTs**
  - This can only be set if 'Convert UDTs' is false
  - Whether or not to 'convert UDTs' only for DATA messages.
- **Set**
  - The Set of AWS endpoints that the Tag Agent will push to.

## Sparkplug Settings

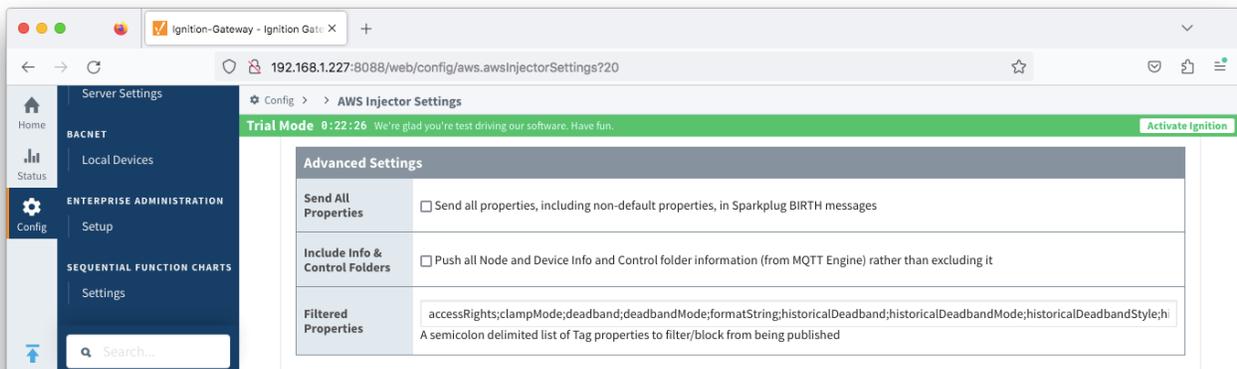


- **Group ID**
  - An ID representing a logical grouping of MQTT Edge Of Network (EoN) Nodes and Devices into the infrastructure.
- **Edge Node ID**
  - An ID that uniquely identifies the MQTT Edge Of Network (EoN) Node within the infrastructure.
- **Device ID**
  - An optional ID that uniquely identifies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

✔ For more information see the [Cloud Injector Tag Agents and Tag Trees](#) tutorial.

## Advanced



- **Send All Properties - added in 4.0.25**
  - Send all properties, including default properties, in Sparkplug BIRTH messages

- **Include Info & Control Folders - added in 4.0.30**
  - Push all Node and Device Info and Control folder tags from MQTT Engine
- **Filtered Properties**
  - A semicolon delimited list of Tag properties to filter/block from being published. These should typically not be modified unless there is an explicit requirement that a specific property is needed to be added or removed from the default.