

AWI: Configuration

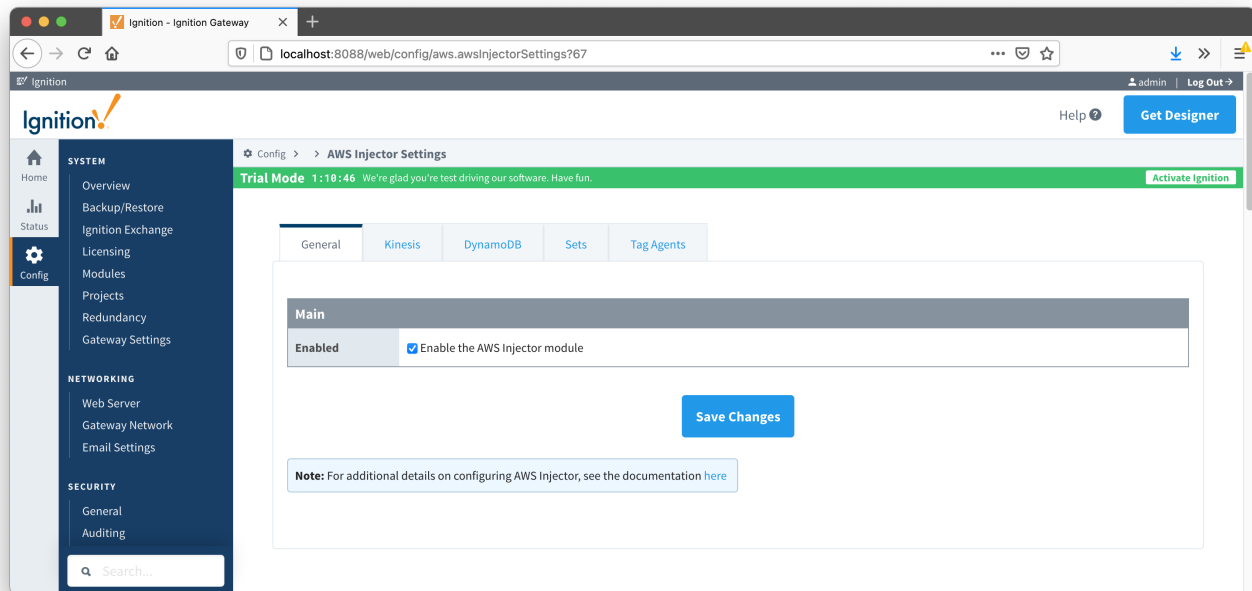
The AWS Injector module provides the ability to push Tag data to an Kinesis Streams and DynamoDB Database endpoints. The settings configuration for this module are located under the Configure tab of the Ignition Gateway web UI.

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

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.



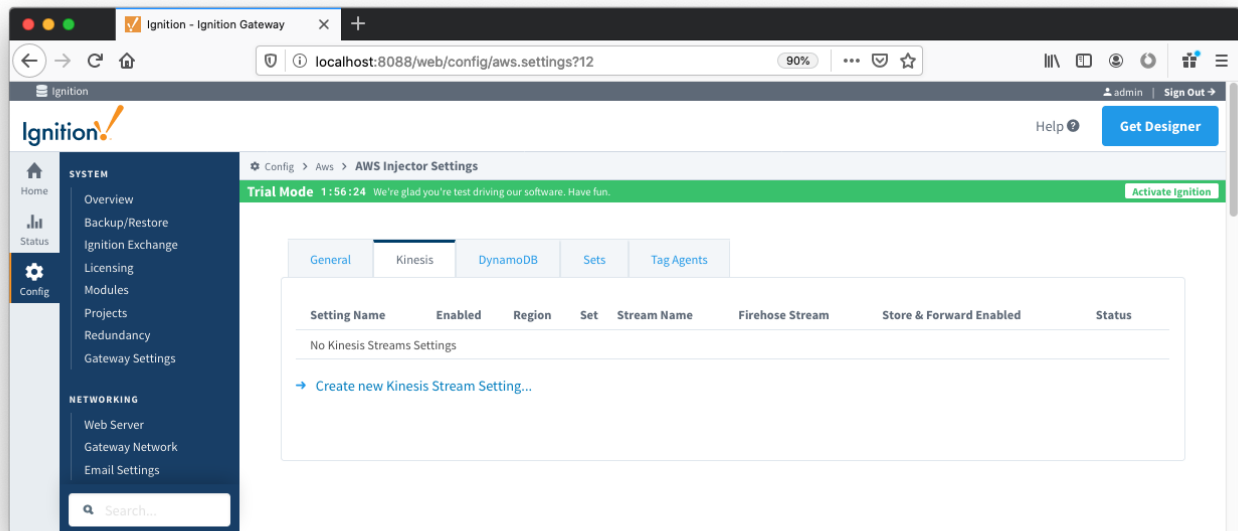
The general configurations options available on this tab are listed below:

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.
- **AWS Access Key**
 - The AWS Access Key ID.
 - Example: AKIAIOSFODNN7EXAMPLE
- **Password/AWS Secret Key**
 - The AWS Secret Access Key.
 - Example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
- **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_Backend (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_Backend 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
- **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.

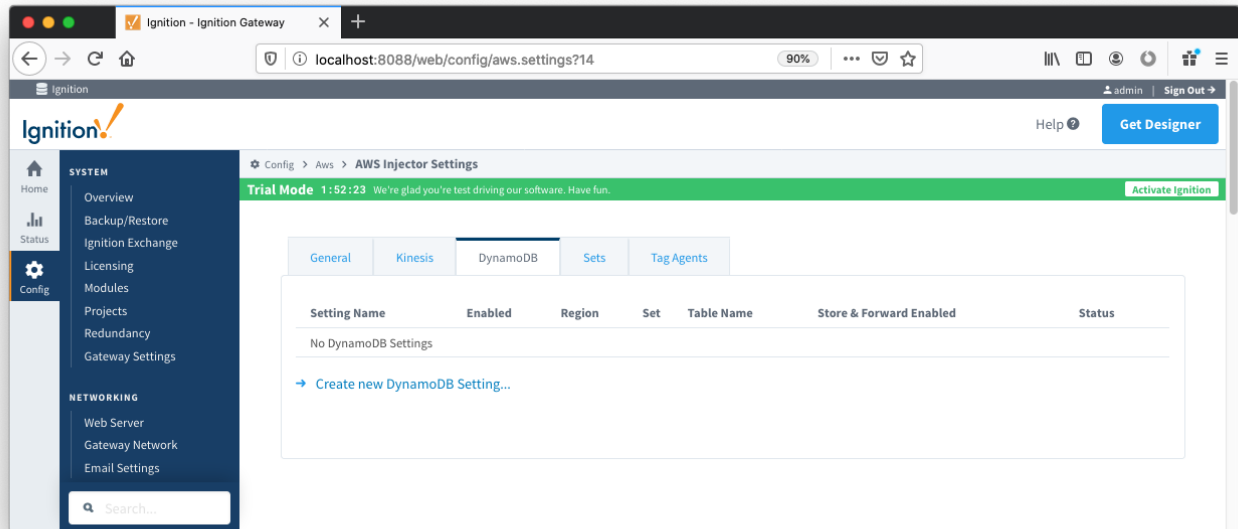
Clicking on the "Create new Kinesis Stream Setting..." link will bring up the following form for adding a new AWS Kinesis Stream endpoint.

The screenshot shows the Ignition Gateway configuration interface. The left sidebar contains a navigation menu with categories: SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, TAGS, OPC CLIENT, and OPC UA. The main content area is titled 'Config > Aws > AWS Injector Settings'. A green banner at the top indicates 'Trial Mode 1:53:05'. Below the banner are tabs for 'General', 'Kinesis', 'DynamoDB', 'Sets', and 'Tag Agents'. The 'Kinesis' tab is active, showing a form for creating a new Kinesis Stream setting. The form includes fields for Setting Name, Enabled (checked), AWS Access Key, Password (masked), AWS Session Tokens (unchecked), Role ARN, Session Duration, Session Name, Region (us-east-1), Set (Default), Stream Name, and Firehose Stream (unchecked).

Main	
Setting Name	New Kinesis Stream A friendly name for this Kinesis Stream setting
Enabled	<input checked="" type="checkbox"/> Enable this setting
AWS Access Key	AKIAIOSFODNNEXAMPLE AWS Access Key
Password	***** AWS Secret Key
Password	***** Re-type password for verification.
AWS Session Tokens	<input type="checkbox"/> AWS Session Tokens
Role ARN	***** AWS Role ARN
Session Duration	***** AWS Session Duration
Session Name	***** AWS Session Name
Region	us-east-1 AWS Region (default: us-east-1)
Set	Default The Set this Kinesis Stream setting is associated with
Stream Name	***** Kinesis Stream Name
Firehose Stream	<input type="checkbox"/> Enables pushing directly into a Kinesis Data Firehose delivery stream (default: false)

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.
- **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.
- **Table Name**
 - The name of the AWS DynamoDB Database table.
- **Set**
 - The Set to associate this AWS DynamoDB Database connection with.

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 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

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

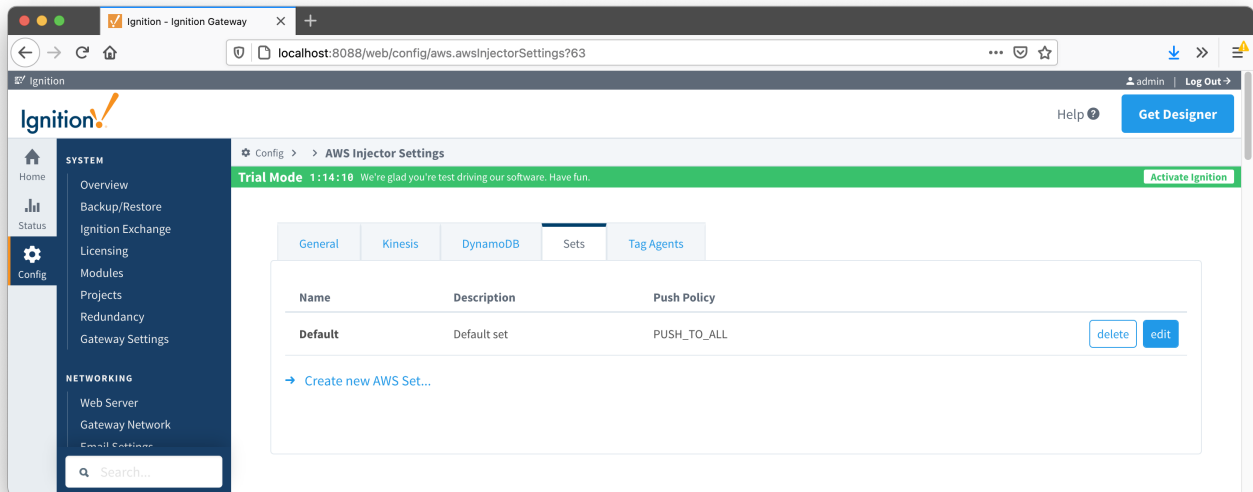
The screenshot displays the Ignition Gateway web interface. The left sidebar contains navigation menus for SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, TAGS, OPC CLIENT, and OPC UA. The main content area is titled 'AWS Injector Settings' and shows a 'Trial Mode' banner. The 'Sets' tab is selected, displaying configuration fields for a new DynamoDB stream. The 'Main' section includes fields for Setting Name, Enabled status, AWS Access Key, Password, Region, Set, and Table Name. The 'Store & Forward' section includes fields for Store & Forward Enabled status, Store & Forward Type, Message Capacity, and Flush Period.

Main	
Setting Name	New DynamoDB Stream A friendly name for this DynamoDB setting
Enabled	<input checked="" type="checkbox"/> Enable this setting
AWS Access Key	AKIAIOSFODNNEXAMPLE AWS Access Key
Password AWS Secret Key
Password Re-type password for verification.
Region	us-east-1 AWS Region (default: us-east-1)
Set	Default The Set this DynamoDB setting is associated with
Table Name DynamoDB Table Name

Store & Forward	
Store & Forward Enabled	<input type="checkbox"/> 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 (default: 10,000)
Flush Period	100

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.

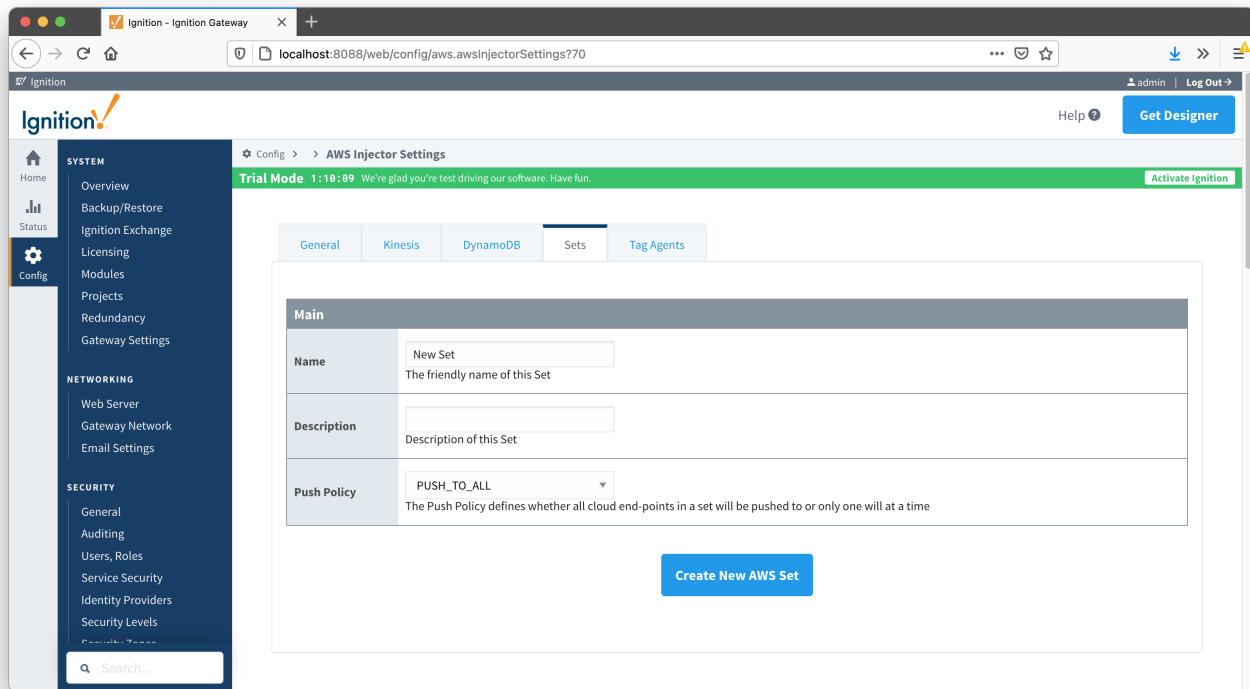


Additional Sets can be configured for situations where multiple Tag Agents will need to push to different AWS endpoints. The configuration options for Sets are listed below.

Main

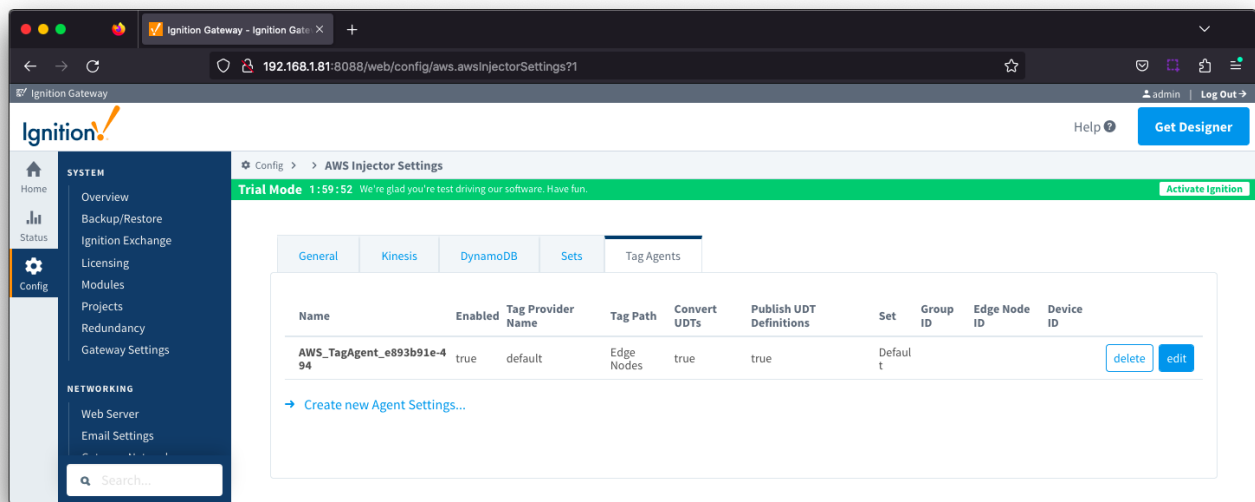
- **Name**
 - This is the friendly name of the set used to easily identify it.
- **Description**
 - This is a friendly description of the set.
- **Push Policy**
 - This defines which endpoints to push to. If PUSH_TO_ALL is selected, every endpoint that is part of this set will receive all messages. If 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.



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.

Tag Settings

- **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 data for all tags associated with the Agent every 'Tag Pacing Period'
- **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. 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 used to either hard-code these IDs, or leave them blank so that the Agent will scan and discover them based on the tag tree layout. For more information see the [Cloud Injector Tag Agents and Tag Trees](#) document.

Advanced

- **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.

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

Ignition Gateway - Ignition Gate

192.168.1.81:8088/web/config/aws.awsInjectorSettings?7

admin | Log Out

Ignition

Help

Get Designer

Home

Status

Config

SYSTEM

Overview

Backup/Restore

Ignition Exchange

Licensing

Modules

Projects

Redundancy

Gateway Settings

NETWORKING

Web Server

Email Settings

Gateway Network

SECURITY

General

Auditing

Users, Roles

Service Security

Identity Providers

OAuth2 Clients

Security Levels

Security Zones

DATABASES

Connections

Drivers

Store and Forward

ALARMING

General

Journal

Notification

On-Call Rosters

Schedules

TAGS

History

Realtime

OPC CLIENT

OPC Connections

OPC Quick Client

OPC UA

Device Connections

Config > AWS Injector Settings

Trial Mode 1:56:33

Activate Ignition

General

Kinesis

DynamoDB

Sets

Tag Agents

Agent Settings

Name

AWS_TagAgent_e893b91e-494

A unique name for the Tag Agent

Enabled

☒ Enable Tag Agent

(default: true)

Tag Provider Name

default

The Name of the tag provider

Tag Path

Edge Nodes

A path to the root folder where the tag tree starts (optional)

Push Trigger

EVENT_DRIVEN

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.

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

Sparkplug Settings

Group ID

An ID representing a logical grouping of Edge Nodes and Devices (optional)

Edge Node ID

An ID representing an Edge or Network (EoN) Node (optional)

Security
Server Settings

BACNET

🔍 Search...

Device ID

An ID representing a Device (optional)

☐ Show advanced properties