



Digi Remote Manager

Integration with Amazon AWS

Tutorial

Revision history

Revision	Date	Description
A	March 2019	Initial release.
B	February 2020	Updated aws_demo.json file to work with latest version of AWS.

Trademarks and copyright

Digi, Digi International, and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

© 2020 Digi International Inc. All rights reserved.

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

Customer support

Gather support information: Before contacting Digi technical support for help, gather the following information:

- ✓ Product name and model
- ✓ Product serial number (s)
- ✓ Firmware version
- ✓ Operating system/browser (if applicable)
- ✓ Logs (from time of reported issue)
- ✓ Trace (if possible)
- ✓ Description of issue
- ✓ Steps to reproduce

Contact Digi technical support: Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Feedback

To provide feedback on this document, email your comments to

techcomm@digicom.com

Include the document title and part number (Tutorial: Digi Remote Manager Integration with Amazon AWS, 90002346 B) in the subject line of your email.

Contents

About this tutorial

Integrate Digi Remote Manager with Amazon AWS

Step 1: Set up an Amazon AWS account	7
Step 2: Set up the sample application for this tutorial	7
Step 3: Identify the invoke URL and API key	10
Step 4: Test the created stack	11
Step 5: Create a Digi Remote Manager monitor	13
Step 6: Test the push monitor	14

About this tutorial

This tutorial demonstrates how to push data from your Digi Remote Manager account to an Amazon AWS IoT analytics channel. The tutorial uses an HTTP push monitor to automatically push alerts and data points uploaded by devices.

The tutorial uses the following components:

- Digi Remote Manager monitor to output data
- AWS API gateway to receive the data
- AWS lambda function to forward the data into the channel
- AWS IoT analytics channel to queue data for processing
- AWS IoT Analytics pipeline to forward data from the channel to a datastore
- AWS IoT Analytics datastore to store the data

Integrate Digi Remote Manager with Amazon AWS

This tutorial demonstrates how to push data from your Digi Remote Manager account to an Amazon AWS IoT analytics channel. The tutorial uses an HTTP push monitor to automatically push alerts and data points uploaded by devices.

Step 1: Set up an Amazon AWS account	7
Step 2: Set up the sample application for this tutorial	7
Step 3: Identify the invoke URL and API key	10
Step 4: Test the created stack	11
Step 5: Create a Digi Remote Manager monitor	13
Step 6: Test the push monitor	14

Step 1: Set up an Amazon AWS account

- Go to <https://aws.amazon.com> and click **Create a Free Account** to set up a free account.

Step 2: Set up the sample application for this tutorial

- Click [here](#) to download the Digi template for a push monitor (**aws_demo.json**).
- Log into your AWS Management Console and go to **Services > Management & Governance > CloudFormation**.
- Click **Create a new stack**.

The screenshot shows the 'Create Stack' wizard in the AWS Management Console. The breadcrumb navigation at the top reads 'CloudFormation > Stacks > Create Stack'. The main heading is 'Create stack'. On the left, there is a vertical navigation menu with four items: 'Select Template' (highlighted with an orange bar), 'Specify Details', 'Options', and 'Review'. The main content area is titled 'Select Template' and contains the following text: 'Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.' Below this text are three options: 1. 'Design a template' with a sub-description 'Use AWS CloudFormation Designer to create or modify an existing template. [Learn more.](#)' and a 'Design template' button. 2. 'Choose a template' with a sub-description 'A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)' and three radio button options: 'Select a sample template' (selected), 'Upload a template to Amazon S3', and 'Specify an Amazon S3 template URL'. The 'Upload a template to Amazon S3' option has a 'Choose File' button and the text 'No file chosen'. The 'Specify an Amazon S3 template URL' option has an empty text input field. At the bottom right of the form, there are 'Cancel' and 'Next' buttons.

- Under **Choose a template**, select **Upload a template to Amazon S3**, and click **Choose File**. Browse to and select the **aws_demo.json** file you downloaded in the previous step.
- Click **Next**.

Create stack

- Select Template
- Specify Details**
- Options
- Review

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

Parameters

ResourceNamePrefix

A string that will be used as a prefix for the names of created resources.

RetentionPeriod

The number of days to store received messages

Cancel Previous **Next**

- f. Provide stack details:
 - Stack name:** Enter **PushMonitorStack**.
 - ResourceNamePrefix:** Enter a prefix for naming all created resources. The default is **push_monitor_messages_**.
 - RetentionPeriod:** Enter the number of days to store received messages. The default is **90**.
- g. Click **Next**.

CloudFormation > Stacks > Create Stack

Create stack

Select Template
Specify Details
Options
Review

Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. [Learn more.](#)

	Key (127 characters maximum)	Value (255 characters maximum)	
1	<input type="text"/>	<input type="text"/>	<input type="button" value="+"/>

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more.](#)

IAM Role:
Enter role arn:

Rollback Triggers

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more](#)

Monitoring Time: Minutes
Minimum value of 0. Maximum value of 180.

Available triggers remaining: 5

- h. For stack **Options**, click **Next** to accept the default options.
- i. For **Review > Permissions**, click **I acknowledge that AWS CloudFormation might create IAM resources with custom names.**

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::Role]
This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more.](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

- j. Click **Create**.

The following resources are created:

- AWS API gateway named pushMonitorEndpoint containing a single method for handling POST messages. It will be deployed to a stage named prod.
- API key and associated usage plan named api_key and usage_plan respectively.

- AWS lambda function named <ResourceNamePrefix>lambda that will be invoked by the API gateway.
- AWS IoT analytics channel named <ResourceNamePrefix>channel.
- AWS IoT analytics pipeline named <ResourceNamePrefix>pipeline.
- AWS IoT analytics datastore named <ResourceNamePrefix>datastore.
- AWS IoT analytics dataset named <ResourceNamePrefix>example_dataset.
- IAM role named <ResourceNamePrefix>role that the lambda function runs as which has the batchPutMessage permission for the channel.

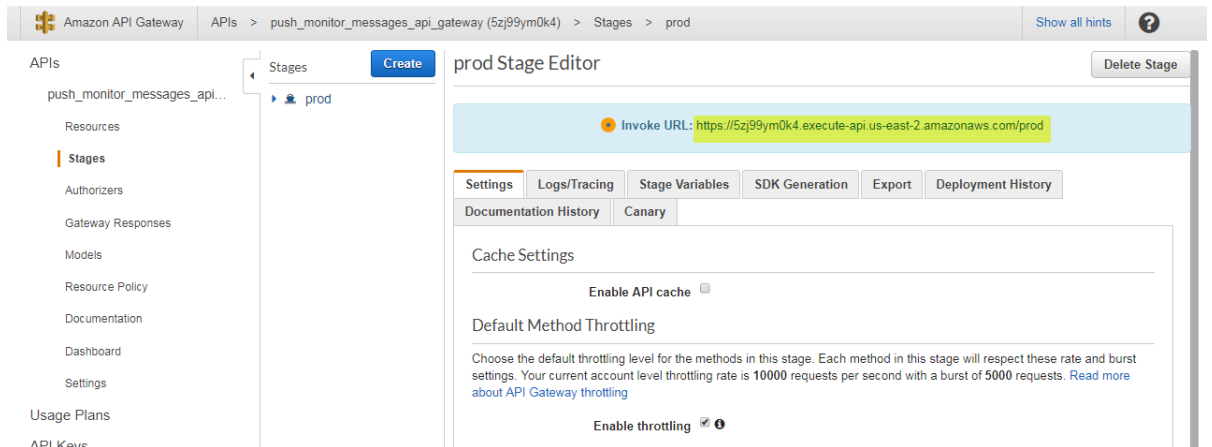
Step 3: Identify the *invoke URL* and *API key*

Once the stack is built, identify the following information for use later in this tutorial:

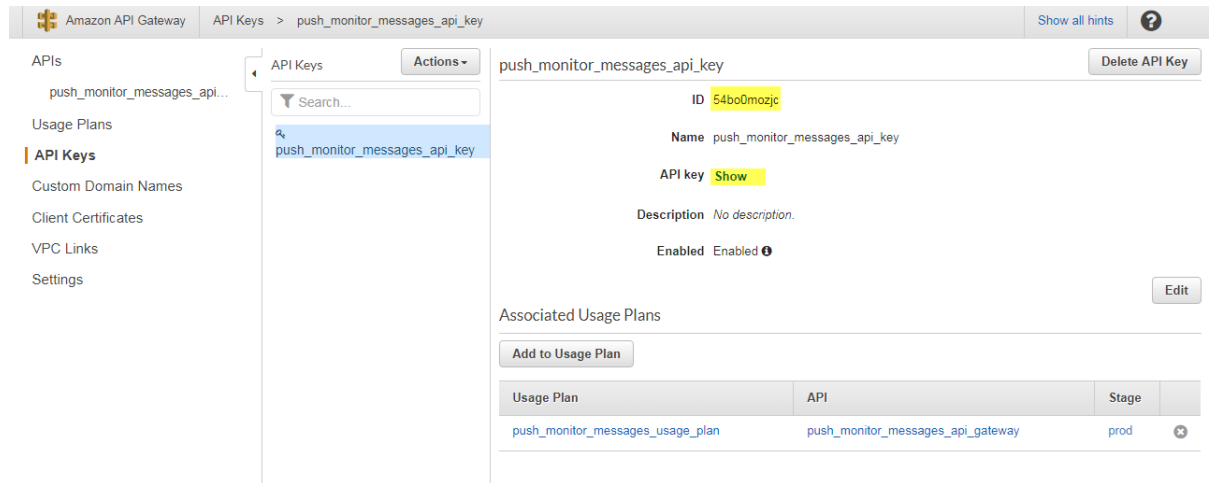
- Invoke URL
 - API key
1. Go to the **API Gateway** console (**Services > API gateway**), and open **push_monitor_messages_api_gateway** settings.



2. Click **Stages > prod** and note the Invoke URL:



3. Click **API keys > push_monitor_messages_api_key** and note the **ID** and **API key**.



Step 4: Test the created stack

To test the stack, create an HTTP client to issue a POST request that simulates how Digi Remote Manager interacts with the created API Gateway.

1. From the command line, use curl to issue a POST request similar to the following.

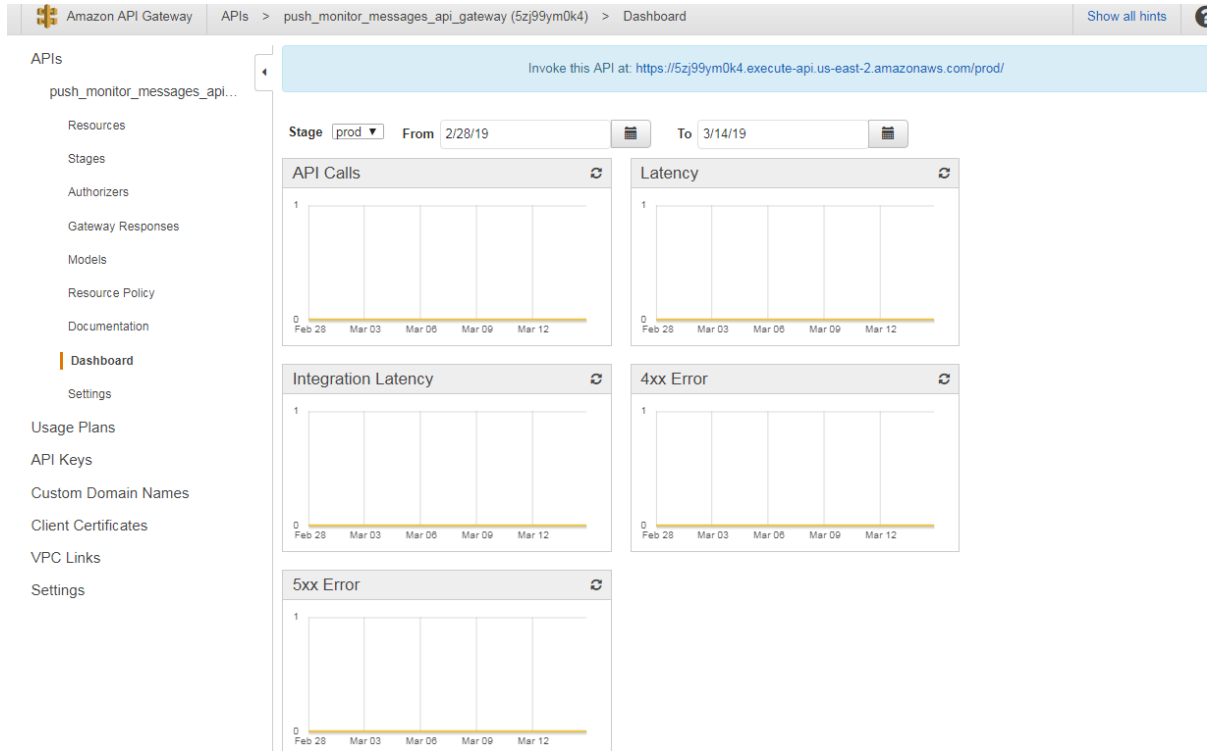
Note Replace the invoke URL and API key highlighted in the sample with the values identified in [Identify the invoke URL and API key](#).

```
curl -X POST \  
-H 'Content-Type: application/json' \  
-H 'x-api-key:APIKEY' \  
--data '{ "Document": { "Msg": {  
  "DataPoint": {  
    "data": 42,  
    "id": "00000000-0000-0000-0000-000000000000",  
    "streamId": "sample/datastream",  
    "streamType": "INTEGER",  
    "timestamp": 1551366000765  
  },  
  "group": "*",  
  "operation": "INSERTION",  
  "timestamp": "2019-02-28T15:00:30.922Z",  
  "topic": "0/DataPoint/awssample/datastream"  
} } }' "https://API-ID.execute-api.REGION.amazonaws.com/prod"
```

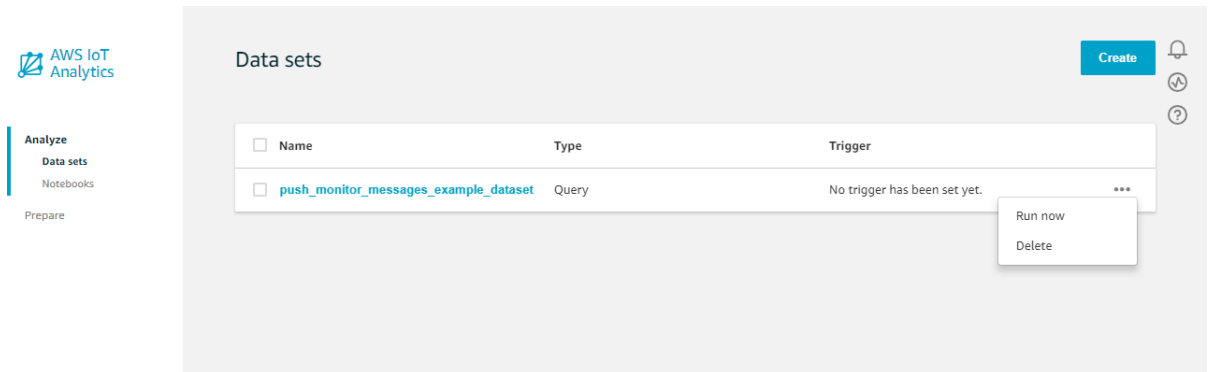
2. In response to the Post request, you should receive a 200 response code and a message with the following body:

```
{"batchPutMessageErrorEntries": []}
```

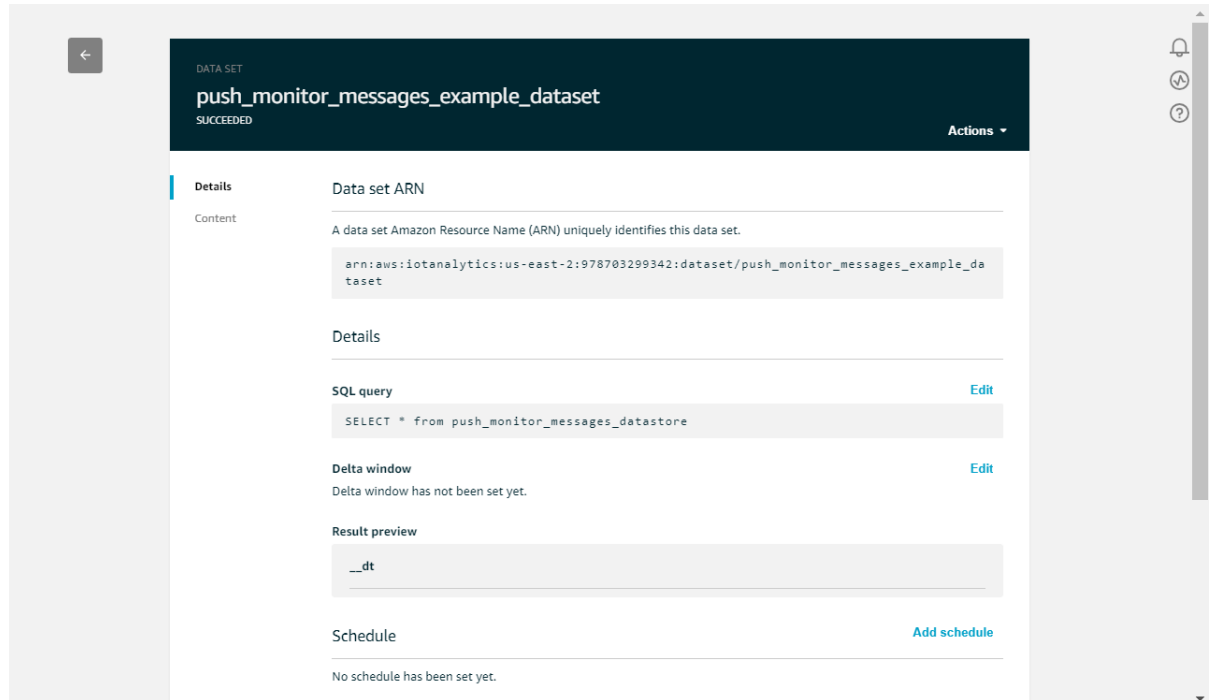
3. On the API Gateway Console dashboard, you should see that the API has been invoked once and did not generate an error response.



- To ensure that the data has been properly stored in the datastore, run the example data set created as part of the stack from the AWS IoT Analytics console.



- View the result:



Step 5: Create a Digi Remote Manager monitor

Create a Digi Remote Manager monitor by posting the monitor definition using the Digi Remote Manager API `/ws/Monitor`.

Specify the following properties:

monTopic—Use the topic `DataPoint/awssample`.

monTransportType—Specify `http`.

monTransportUrl—Specify the **invoke URL** of the API gateway. See [Identify the invoke URL and API key](#).

monTransportHeaders—Specify `x-api-key:API_KEY`; replace `API_KEY` with the API key for the push monitor. See [Identify the invoke URL and API key](#).

monTransportMethod—Specify `POST`.

monFormatType—Specify `json`.

See `ws/Monitor` in the [Digi Remote Manager Programmer Guide](#).

For example:

```
curl -X POST \
-u REMOTE_MANAGER_USERNAME:REMOTE_MANAGER_PASSWORD \
--data '<Monitor>
  <monTopic>DataPoint/awssample</monTopic>
  <monTransportType>http</monTransportType>
  <monTransportUrl>https://API-ID.execute-
api.REGION.amazonaws.com/prod</monTransportUrl>
  <monTransportHeaders>x-api-key:APIKEYVALUE</monTransportHeaders>
  <monTransportMethod>POST</monTransportMethod>
```

```
<monFormatType>json</monFormatType>
</Monitor>' "https://remotemanager.digi.com/ws/Monitor"
```

Step 6: Test the push monitor

Verify the monitor is working properly by adding a data point to the **awssample** data stream by making a POST request to the Digi Remote Manager API **/ws/v1/streams/history**. See [ws/v1/streams](#) in the [Digi Remote Manager Programmer Guide](#).

For example:

```
curl -X POST \
-H 'Content-Type: application/json' \
-u REMOTE_MANAGER_USERNAME:REMOTE_MANAGER_PASSWORD \
--data '{
  "stream_id": "awssample",
  "stream_type": "DOUBLE",
  "value": "42"
}' "https://remotemanager.digi.com/ws/v1/streams/history"
```

Then use the API Gateway dashboard and sample data set to verify the data point is stored. See [Test the created stack](#).