

APN Blog

Coarse Location Tracking with AnyNet Secure SIM and AWS IoT

Appendix III – Create IoT Rule to Send Data to AWS IoT Analytics

Follow the steps below to create an IoT rule and add actions to republish the message and send data to IoT Analytics.

1. Create a new rule filter using following values:

a. **Attribute**

```
state.reported.anynet.metadata.msisdn,
state.reported.anynet.metadata.location.lat,
state.reported.anynet.metadata.location.lon, version,
timestamp
```

b. **Topic filter:** *We'll use the shadow accepted topic*

```
$aws/things/<Your Eseye Thing Name>/shadow/update/accepted
```

c. **Condition:**

```
state.reported.anynet.status = "Provisioned"
```

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

Description

Message source

Indicate the source of the messages you want to process with this rule.

Using SQL version [?](#)

2016-03-23

Rule query statement

```
SELECT state.reported.anynet.metadata.msisdn, state.reported.anynet.metadata.location.lat,
state.reported.anynet.metadata.location.lon, version, timestamp FROM
'$aws/things/EseyeDemoThing/shadow/update/accepted' WHERE state.reported.anynet.status =
'Provisioned'
```

Attribute [?](#)

```
state.reported.anynet.metadata.msisdn, state.reported.anynet.metadata.location.lat, state.reported.anynet.metadata.location.lon, version,
timestamp
```

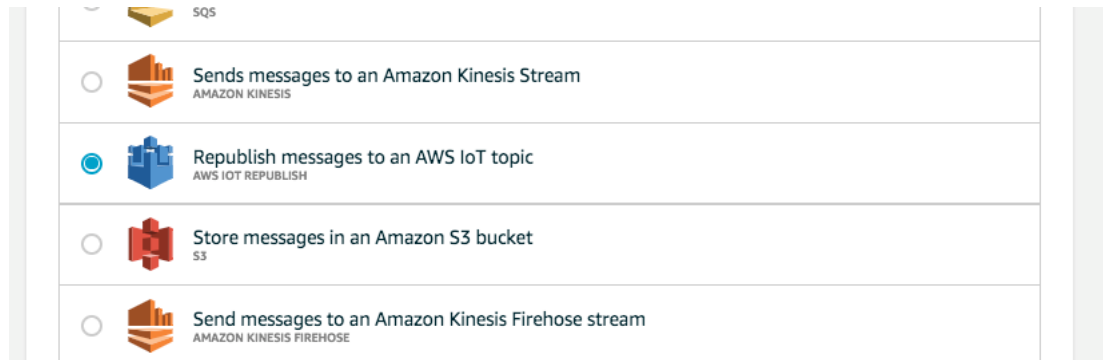
Topic filter [?](#)

```
$aws/things/EseyeDemoThing/shadow/update/accepted
```

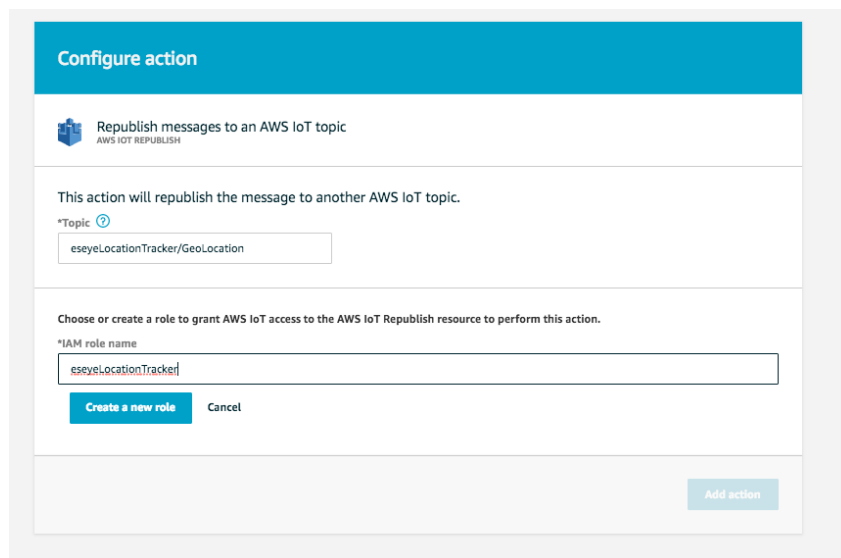
Condition [?](#)

```
state.reported.anynet.status = "Provisioned"
```

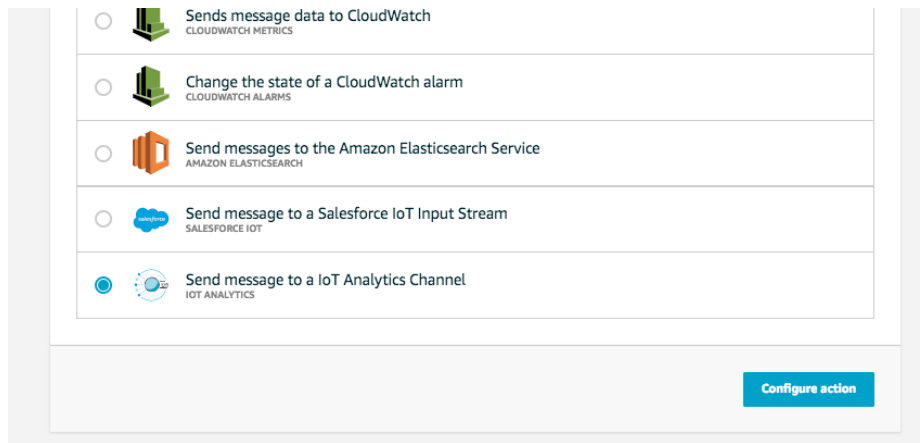
2. Add action to republish message to AWS IoT:



3. Create a new topic "eseyeLocationTracker/GeoLocation" and click on "Create a new Role":

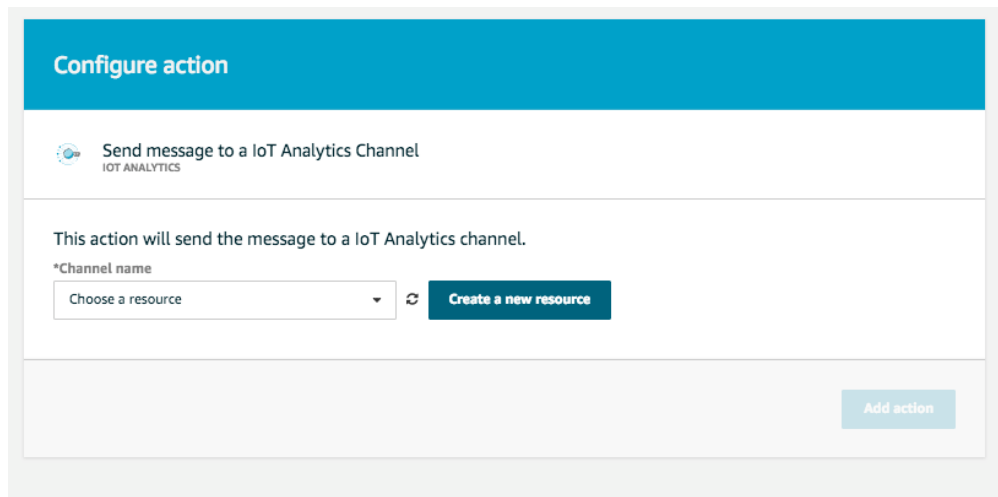


4. Click on "Add action" button and select "Send message to IoT Analytics Channel":

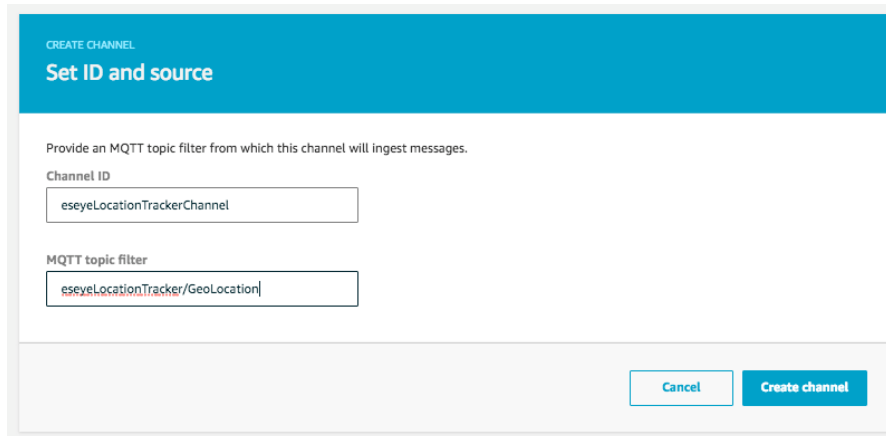


5. Set up AWS IoT Analytics

a. Click on "Create a new resource" button:



- b. Provide a unique name to the Channel ID, and then paste the topic you created to republish the message:



CREATE CHANNEL

Set ID and source

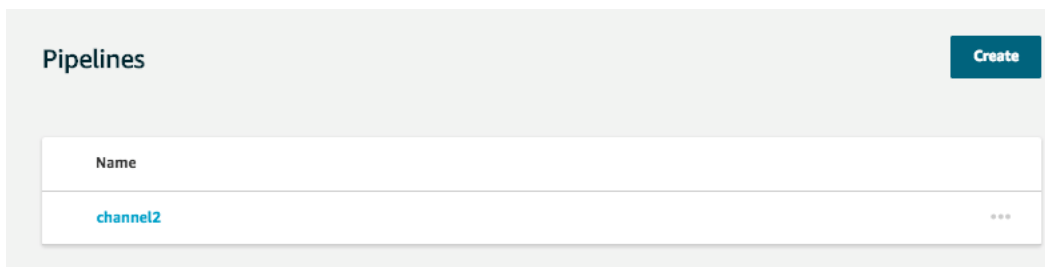
Provide an MQTT topic filter from which this channel will ingest messages.

Channel ID

MQTT topic filter

Cancel Create channel

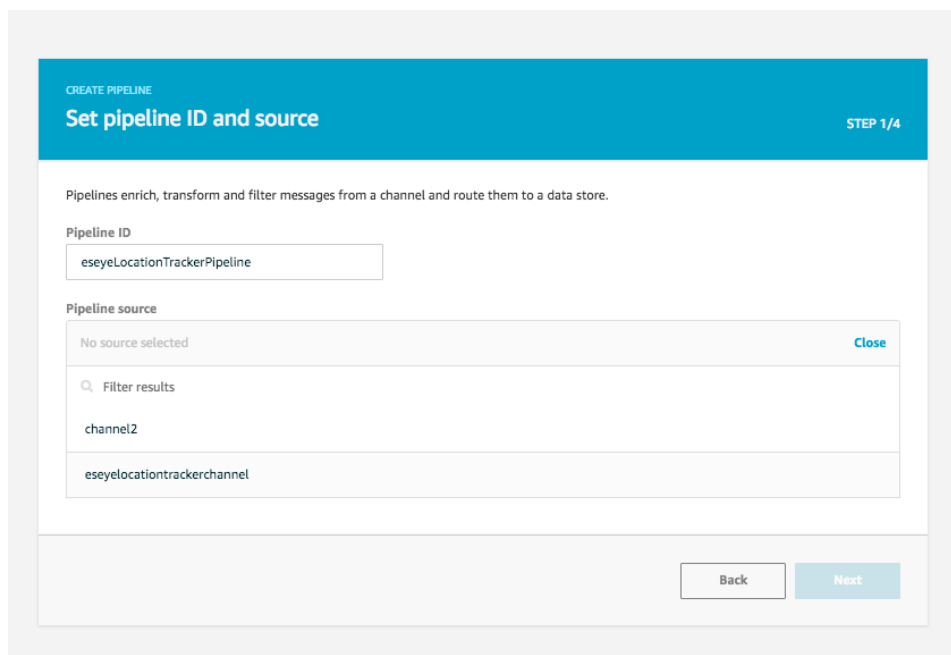
- c. Click on create button on the top:



Pipelines Create

Name
channel2 ...

- d. Provide unique name for the pipeline and select source from dropdown.



CREATE PIPELINE

Set pipeline ID and source

STEP 1/4

Pipelines enrich, transform and filter messages from a channel and route them to a data store.

Pipeline ID

Pipeline source

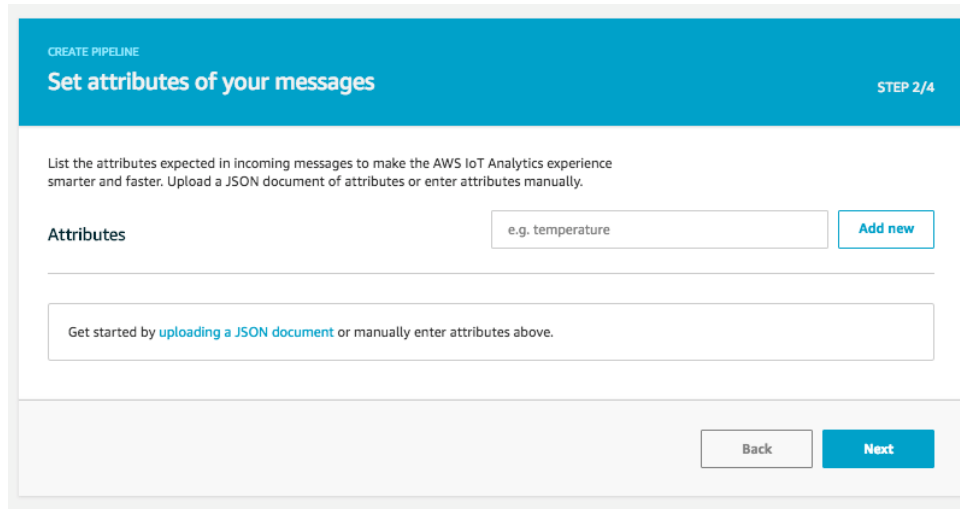
No source selected Close

Filter results

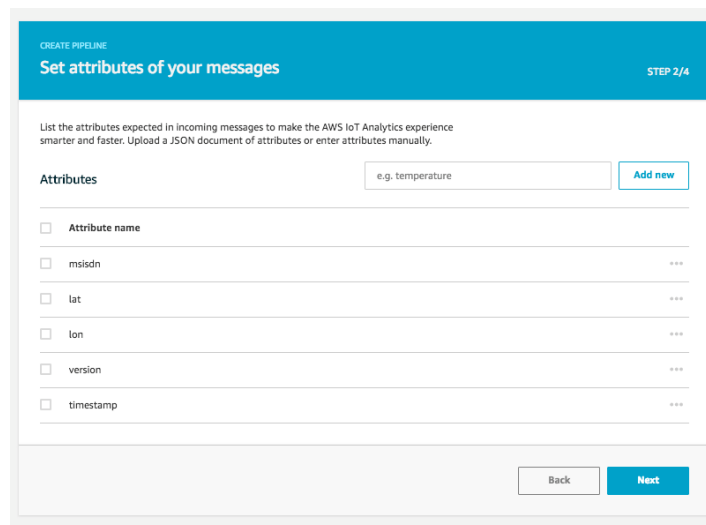
- channel2
- eseyelocationtrackerchannel

Back Next

- e. Download the file below and save it in a local folder `eseye.json`.
- f. Click on uploading a JSON document and select the file you just downloaded:

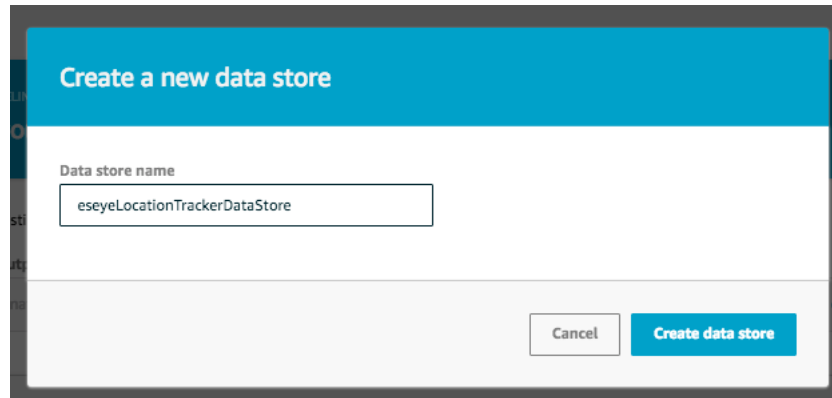


- g. Once uploaded, the console will parse the JSON and populate the screen:



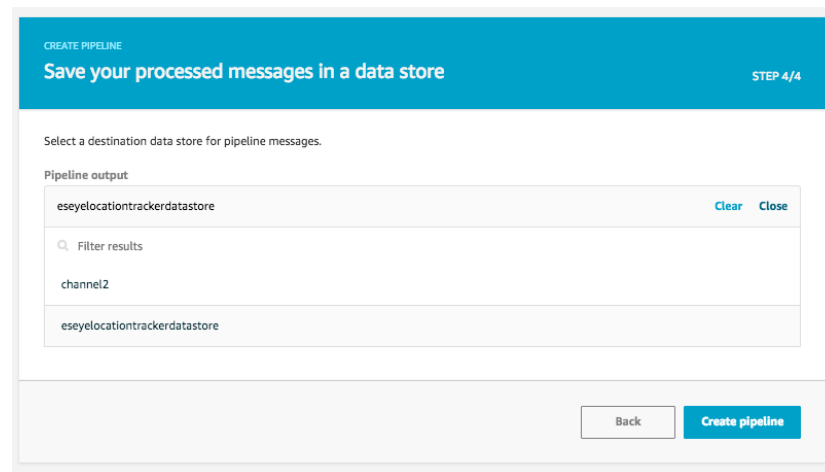
- h. Click next.
- i. Click on "Create new data store."

j. Give a unique name:



The screenshot shows a form titled "Create a new data store". It has a text input field for "Data store name" containing the text "eseyeLocationTrackerDataStore". At the bottom right, there are two buttons: "Cancel" and "Create data store".

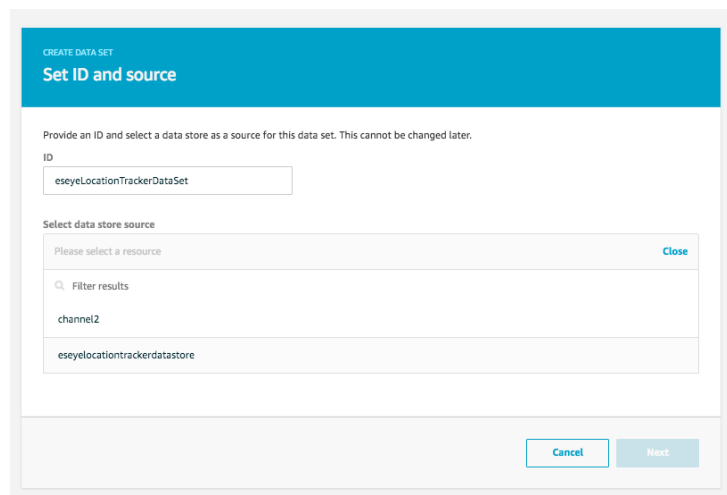
k. Select the data store you just created:



The screenshot shows a step in a pipeline creation process titled "Save your processed messages in a data store". It asks to "Select a destination data store for pipeline messages." Under "Pipeline output", there is a list of data stores. The first item is "eseyelocationtrackerdatastore" with "Clear" and "Close" buttons. Below it is a search bar "Filter results" and another item "channel2". At the bottom, there are "Back" and "Create pipeline" buttons.

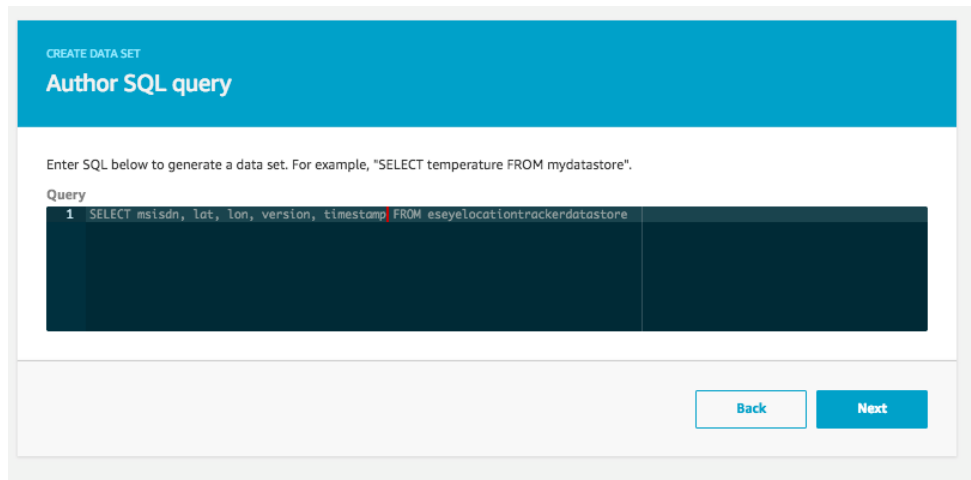
l. Click the create button.

m. Select the data store you created:



The screenshot shows a form titled "Set ID and source". It asks to "Provide an ID and select a data store as a source for this data set. This cannot be changed later." There is a text input field for "ID" containing "eseyeLocationTrackerDataSet". Below it, there is a section "Select data store source" with a search bar "Filter results" and a list of data stores. The first item is "eseyelocationtrackerdatastore". At the bottom, there are "Cancel" and "Next" buttons.

- n. Replace * with the following to create the SQL statement: `msisdn, lat, lon, version, timestamp`



CREATE DATA SET

Author SQL query

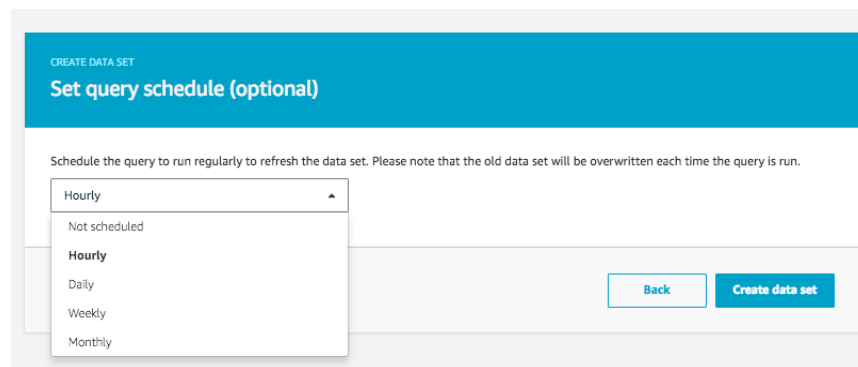
Enter SQL below to generate a data set. For example, "SELECT temperature FROM mydatastore".

Query

```
1 SELECT msisdn, lat, lon, version, timestamp FROM eseyelocationtrackerdatastore
```

Back Next

- o. From the dropdown, select hourly as schedule:



CREATE DATA SET

Set query schedule (optional)

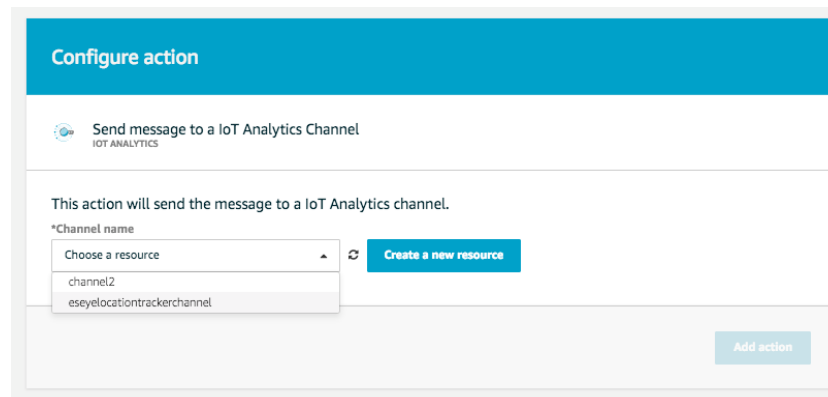
Schedule the query to run regularly to refresh the data set. Please note that the old data set will be overwritten each time the query is run.

Hourly

- Not scheduled
- Hourly**
- Daily
- Weekly
- Monthly

Back Create data set

- p. Let's go back to the rules screen and press the refresh icon, and then choose the channel we just created from the drop down.



Configure action

Send message to a IoT Analytics Channel

This action will send the message to a IoT Analytics channel.

*Channel name

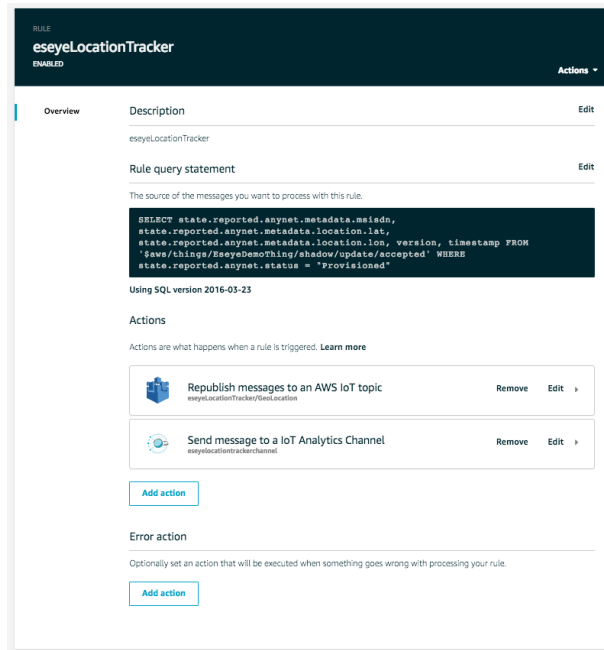
Choose a resource

- channel2
- eseyelocationtrackerchannel

Create a new resource

Add action

6. In the end, it will look like this:



RULE
eseyeLocationTracker
ENABLED

Overview Description Edit

eseyeLocationTracker

Rule query statement Edit



The source of the messages you want to process with this rule:

```
SELECT state.reported.anynet.metadata.msidsn,
state.reported.anynet.metadata.location.lat,
state.reported.anynet.metadata.location.lon, version, timestamp FROM
'save/chassis/eseyeDeviceThing/shadow/update/accepted' WHERE
state.reported.anynet.status = "Provisioned"
```

Using SQL version 2016-03-23

Actions

Actions are what happens when a rule is triggered. [Learn more](#)

- 
Republish messages to an AWS IoT topic
eseyeLocationTracker/GeoLocation
Remove Edit
- 
Send message to a IoT Analytics Channel
eseyeLocationTrackerChannel
Remove Edit

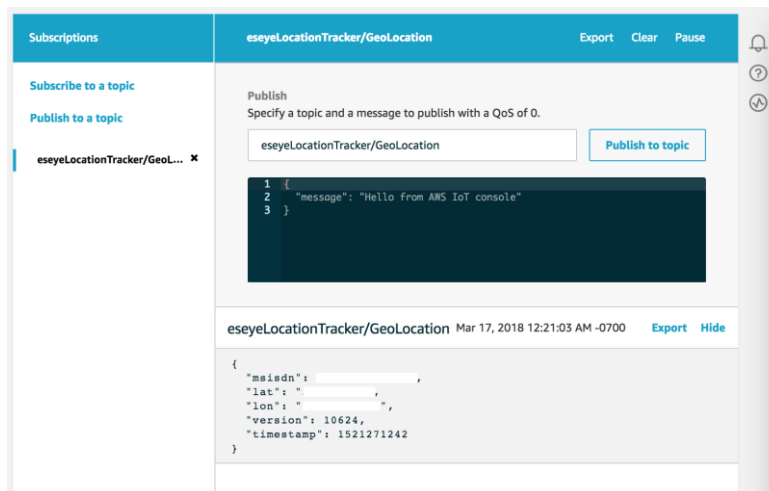
[Add action](#)

Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

[Add action](#)

7. To test trigger Lambda and monitor "eseyeLocationTracker/GeoLocation" topic:



Subscriptions eseyeLocationTracker/GeoLocation Export Clear Pause

[Subscribe to a topic](#)

[Publish to a topic](#)

eseyeLocationTracker/GeoL... ✕

Publish
Specify a topic and a message to publish with a QoS of 0.

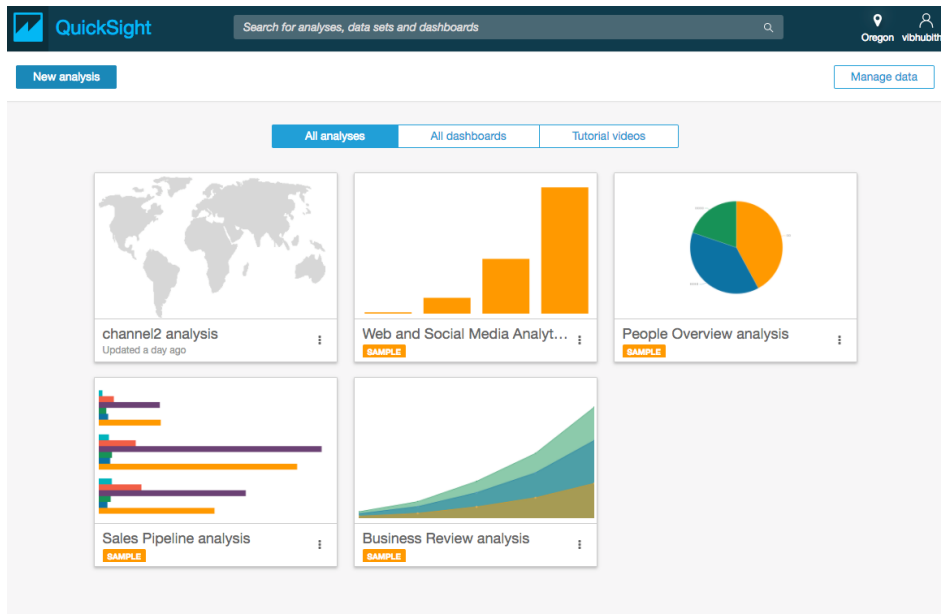
eseyeLocationTracker/GeoLocation Publish to topic

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

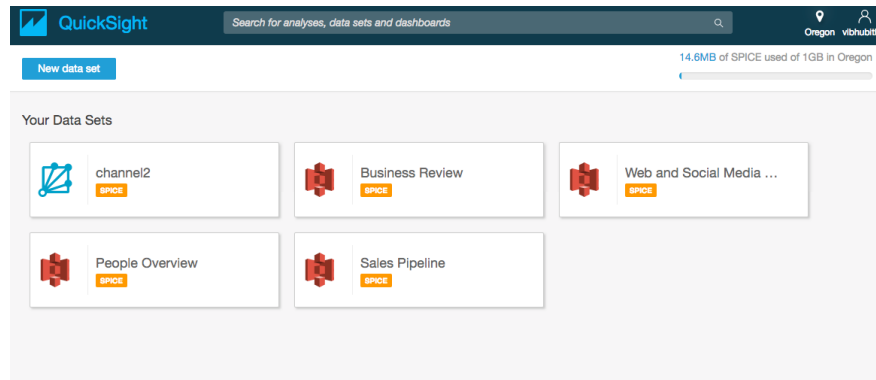
eseyeLocationTracker/GeoLocation Mar 17, 2018 12:21:03 AM -0700 Export Hide

```
{
  "msidsn": " ",
  "lat": " ",
  "lon": " ",
  "version": 10624,
  "timestamp": 1521271242
}
```

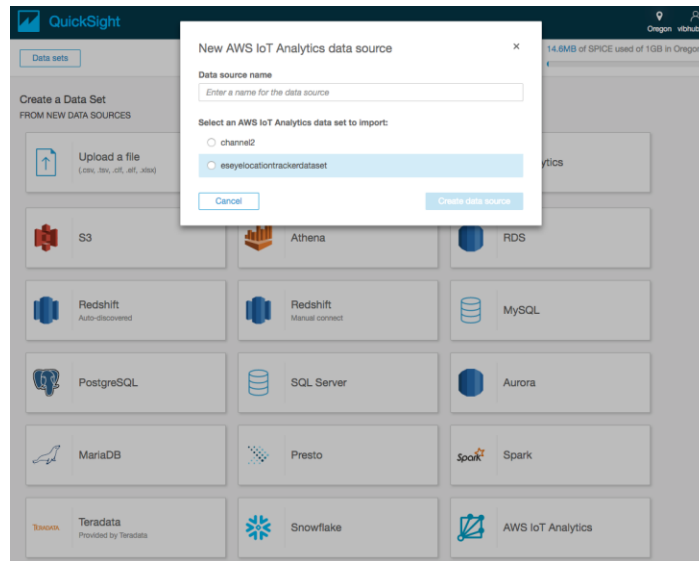

8. Connect Amazon QuickSight to AWS IoT Analytics dataset to visualize the data
 - a. Click on "New Analysis" button:



- b. Click on "New data set":



- c. Select the dataset created as part of AWS IoT Analytics and click "Create data source":



- d. Once the data import completes, click the visualize button:

- e. Finally, lick on the geospatial icon at the bottom of the screen and add latitude and longitude:

