# トリリオン/LTE-Mリーフを利用した省電力,データ送信プラットフォームの試作検討

2021年7月16日

KDDI株式会社 サービス技術部

# クラウド導入ガイド

## １．概要

本ドキュメントLTE-Mリーフによるセンサーノードからのデータを可視化するためのクラウド設定方法を記載したものである

## ２．AWSの設定

AWSに対して実行ユーザ登録、IoT core, TimeStream, Lambdaの設定を行う

### 2-1. AWSリージョンの変更

TimeStreamが北米リージョンのみ利用可能であるためリージョン設定を行う
本資料では米国西部リージョンであるOregon(us-west-2)で設定を行っている

### 2-2. IAMユーザの追加、ポリシーの設定

ユーザー名： 任意 (例:ltem-aws-0528a)
アクセスの種類: [プログラムによるアクセス]

[ポリシーの作成]
ポリシー名: 任意(例:ltem-aws-policy-0528a)
ツールを使って入力すると不要ポリシーが大量に入るためJSON入力で設定を行う
Resource行にある数値部分をIAMユーザーの ARNにある数値列に差し替え
us-west-2の部分はIoT coreを設置するリージョン名に差し替え
(us-east-1, us-east-2, us-west-1, us-west-2)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iot:Publish",
            "Resource": "arn:aws:iot:us-west-2:XXXXXXXXXXXX:topic/*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "timestream:*",
```

```
            "Resource": "*"
        }
    ]
}
```

IAMの設定が完了したらアクセスID,シークレットアクセスキーを取得する
取得した内容は各設定で使うのでメモ①②しておく
ユーザー: (例:ltem-aws-0528a)
アクセスキー ID①: (例:AKIAZB5CYZQ6WXJKEEXC)
シークレットアクセスキー②: (例:1kj5B+k7uYptKPZ5Ls/vNd032OYvdO9EjJfD/Qfe)

## 2-3. IoT Coreの設定

検索窓に"IoT"と入力しAWS IoT coreを選択する
AWS IoTの画面に切り替わったら左側にある設定リンクを押す

この画面の「デバイスデータエンドポイント」項目の文字列をメモ③しておく
例) acaf2z16zr5pa-ats.iot.us-west-2.amazonaws.com

## 2-4. TimeStreamの設定

検索窓に"TimeStream"と入力しAmazon Timestreamを選択する

[Create database]ボタンを押して新規データベースを作成する
Database configuration: Standard database
Name:  任意(例:LTE-M-DB0528)、入力したらメモ⑤しておく
Encryption: デフォルト

次に[Create table]ボタンを押してデータテーブルを作成する

Database name: 先ほど作成したデータベース名(メモ⑤) Table name: 任意(例:LTE-M-TBL0528)、入力したらメモ⑥しておく [Create table]ボタンを押して作成完了

データの保存期限を指定
例としてメモリ上に1日間、ストレージ保存7日間を指定

Memory store retention: "1" Day(s)
Magnetic store retention "7" Day(s)
[Create table]ボタンを押して作成完了

## 2-5. Lambda関数の登録

検索窓に"lambda"と入力しAWS Lambdaを選択する
[関数の作成]ボタンを押して関数登録を行う
作成オプション：「一から作成」
関数名: 任意(lambda_function0523)、入力したらメモ⑦しておく
ランタイム: Python 3.8
[関数の作成]ボタンを押すとコード作成画面に移行する

コードソースのツリー上にあるlambda_function.pyをダブルクリックすると編集画面にあるので添付されているソースコードを張り付ける
lambda_function0523.py

張り付けが終わったら次の部分を修正する

```
DATABASE_NAME = "作成したデータベース名(メモ⑤)"
TABLE_NAME = "データベースのテーブル名(メモ⑥)"
```

完了したら[Deploy]ボタンを押して保存する

次にLambda起動のトリガーを登録する
[トリガー を追加]ボタンを押すと設定画面に移行する
トリガーの設定にてAWS IoTを選択

IoT タイプ: カスタム IoT ルール
ルール: 新規ルールの作成
ルール名: 任意(例:lambda_call_iot0528)、入力したらメモ⑦しておく
ルールクエリステートメント: SELECT * FROM 'myTopic/'
[トリガー を追加]ボタンを押して設定を完了させる

Lambdaの実行ロールにTimestreamのアクセス権限を追加
IAM画面のロール画面に移動するとメモ⑦の名前の付いたロールが追加されているのでリンクをクリックする
AWSLambdaBasicExecutionRole...の一つしか登録されていないことを確認し[ポリーシーをアタッチします]ボタンを押す
検索窓で AmazonTimestreamFullAccess を入力し候補にチェックを付けて[ポリシーのアタッチ]ボタンを押す

# 2-5. Grafana(EC2)の設定

## EC2インスタンスの作成

t2-micro でubuntu 18.04をインストールする
初回利用時はSSH鍵(pem)を作成しダウンロードしておく
起動後はインスタンスのIPアドレスをメモしておく

## インスタンスのセキュリティ設定

EC2->セキュリティグループに移動し登録されているセキュリティ設定を開く
デフォルトでは22(SSH)のみ有効となっている

[インバウンドルールを編集]ボタンを押し設定を行う
[ルールを追加]ボタンを押すと設定行が増えるので次の内容を入力する
・タイプ：カスタムTCP
・プロトコル：TCP
・ポート範囲：3000
・ソース： 0.0.0.0/0
・説明：Grafana

入力が終わったら[ルールを保存]ボタンを押す

### EC2にログイン

ubuntuユーザ、SSH鍵(pem)を使ってログインする

### ubuntuのアップデート

次のコマンドでubuntuの更新プログラムを適用する

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

### grafanaのインストール

grafanaをaptパッケージで導入する

```
$ sudo apt-get install -y adduser libfontconfig1
$ wget https://dl.grafana.com/oss/release/grafana_7.5.5_amd64.deb
$ sudo dpkg -i grafana_7.5.5_amd64.deb
```

プラグインをインストール
AWS TimeStreamとGPS位置表示プラグインをインストール

```
$ sudo grafana-cli plugins install grafana-timestream-datasource
$ sudo grafana-cli plugins install pr0ps-trackmap-panel
```

サービス起動

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart grafana-server
$ sudo systemctl status grafana-server
```

### インストール確認

ブラウザからサーバー名:3000でアクセスする
Grafanaのログイン画面が出ればインストール成功
初回ログインは admin/admin のアカウントで行う
ログイン後はadminパスワード変更を求められるので新たにパスワードを設定する

# ３．SORACOMクラウドの設定

SORACOMクラウドでは以下の項目の設定を行う

## 3-1. SIMグループ登録

SORACOMクラウドでは複数のSIMを束ねたSIMグループにてクラウド設定を行うようになっている

・新規にSIMグループを登録
・LTE-Mリーフで使用するSIMをSIMグループに登録する

## 3-2. セキュリティ設定

・SORACOMコンソール右上のボタンを押し「セキュリティ」を選択する
・左側のリストにある「認証情報ストア」を選択
・[認証情報を登録]ボタンを押し次の内容を登録する
　認証情報 ID: 任意(他の設定で用いる認証名として登録される)
　概要: 任意
　種別: AWS 認証情報
　AWS ACCESS KEY ID: AWS IoTアクセスキー ID
　　2-1項で取得したIDを入力(メモ①)
　AWS SECRET ACCESS KEY: シークレットアクセスキー
　　2-1項で取得したアクセスキーを入力(メモ②)

## 3-2. SORACOM Air for Celler設定

次の項目を有効にする
・簡易位置測位機能

## 3-2. Funnel設定

転送先サービス: AWS IoT
転送先 URL : 2-2項で取得したAWS IoT エンドポイント(メモ③)
　例： https://acaf2z16zr5pa-ats.iot.us-west-2.amazonaws.com/myTopic/

認証情報: 3-2項で登録した認証情報 IDを選択する
送信データ形式: text

## 3.3. Beam設定

本項目はファームウェア設定編にて掲載

# 4． 動作テスト

## 4-1. AWS IoT Coreでの動作確認

AWSコンソールからAWS IoTの画面に移動
左側にあるテストの項目を展開し「MQTTテストクライアント」を開く
トピックのフィルターの枠に"myTopic/#"を入力
[サブスクライブ]ボタンを押して待機する
LTE-Mリーフを起動し緑LEDの点灯2回目でサブスクリプション欄にメッセージが表示されればAWS IoTまでの通信は正常に動作している


```
{
  "operatorId": "OP0081091874",
  "timestamp": 1621911687158,
  "destination": {
    "resourceUrl": "https://a1llzhubvqqis2-ats.iot.us-west-2.amazonaws.com/myTopic/",
    "service": "aws-iot",
```

```
    "provider": "aws"
  },
  "credentialsId": "funnel_aws_iot_key_0503b",
  "payloads": "{\"model\":\"LTE-
M_LeafV3\",\"iccid\":\"8981300025200054523\",\"time\":1621911667,\"battery\":
[3809],\"signalStrength\":[21],\"temp\":[27.06],\"humidity\":[41],\"pressure\":
[999],\"illuminance\":[1959],\"co2\":[400],\"tvoc\":[0],\"pir\":[0],\"axis\":
[-0.012695,0.007813,-1],\"gyro\":[-0.244141,-0.854492,0],\"num_sensor\":11}\u0000",
  "locationQueryResult": "success",
  "imei": "359416090212981",
  "sourceProtocol": "udp",
  "simId": "8981300025200054523",
  "imsi": "440525060026952",
  "location": {
    "lat": 36.xxxxxxx,
    "lon": 140.xxxxxxx
  }
}
```


## 4-2. AWS TimeStreamでの動作確認

AWSコンソールからAmazon Timestreamの画面に移動
左側にあるQuery editorを開く
Choose a database to query.欄で登録したデータベース名を選択
Query 1の枠に次のような文字列を入れる

```
SELECT * FROM "{データベース名}"."{テーブル名}" WHERE time between ago(7d) and now() ORDER
BY time DESC LIMIT 60
```

[実行]ボタンを押す
model,measure_value::double,measure_name,timeの項目が表示されればAWS内のデータ送信は正常に動作している
※timeに表示される時刻はUTC-0の時刻となっているので注意

# ５．可視化設定

ブラウザからgrafanaに再度ログインして可視化の設定を行う

## 5-1 データベース選択

左側の歯車アイコンを押し、展開されたリストから「Data Soureces」を選択する
初期状態では何も表示されていないので [Add data source]ボタンを押す
リストから「Amazon Timestream」を選択する

DataSouce設定画面が表示されるので TimeStreamを設定する
Authentication Provider: Access & secret key
Access Key ID:  AWS IoTアクセスキー ID
　2-1項で取得したIDを入力(メモ①)

Secret Access Key: シークレットアクセスキー:
　2-1項で取得したアクセスキーを入力(メモ②)
Assume Role ARN: 未設定
External ID: 未設定
Default Region: us-west-2
Endpoint: query-cell1.timestream.us-west-2.amazonaws.com

[Save & Test]ボタンを押して「Connection success」が表示されたらデータベース接続は完了

## 5-2 ダッシュボード作成

左側にある□4つのアイコンを押しDashboard->Manageを選択
[New Dashboard]ボタンを押す
Add Panel[Add an empty panel]ボタンを押す
ダミー画面が表示されるのでQueryタブの部分からデータ選択を行う
データベース： Amazon Timestream
Macros:
$**database= Timesteamのデータベース名**
**$**table=登録したテーブル名
$_*measure=データ名*
*次のような命名規則でデータが入っている*
　*{data-name}*{iccid}
data-nameは次のような名称となっている
　asisX,AsisY,AsisZ,battery,co2,gps_alt,gps_lat,gpa_lon,gyroX,gyroY,gyroZ,humidiry,illuminance,
　lte_lat,lte_lon,pressure,signalStrength,temp,tvoc

クエリ行には次のような文字を入力する
このクエリでは現在より1日前までのデータ取得としている
もしそれよりも前から表示を行いたい場合は(1d)の数値を伸ばすようにする
<グラフ化>

```
SELECT measure_name, measure_value::double AS $__measure, time FROM
$__database.$__table WHERE measure_name = '$__measure' AND time between ago(1d) and
now()
```

<座標>
Query A

```
SELECT measure_value::double AS latitude, time FROM $__database.$__table WHERE
measure_name = '$__measure' AND time between ago(1d) and now()
```

Query B (追加)

```
SELECT measure_value::double AS longitude, time FROM $__database.$__table WHERE
measure_name = '$__measure' AND time between ago(1d) and now()
```

## 5-3 可視化選択

右上の「Show options」ボタンを押すとオプション選択が展開される
デフォルトはGraphが選択されている
地図を出す場合は一番下のTrackMapを選択

## 5-4 留意事項

このドキュメント記載時においてgrafanaの可視化で次のような現象が確認されている
TrackMap: latitudeとlongitudeが反転して取り込まれ北極海に座標プロットが表示される
Graph: グラフのラインが過去に向かってプロットされる
いずれも次回更新時には正常表示となる