

## 抜粋版

# はじめてのAWSを使った運行管理開発キット

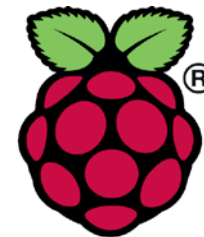
～GPS端末からWebアプリまでトータルで提供、従量制でコスト削減～



AWS IoT Core



Amazon Location Service



Raspberry Pi

スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

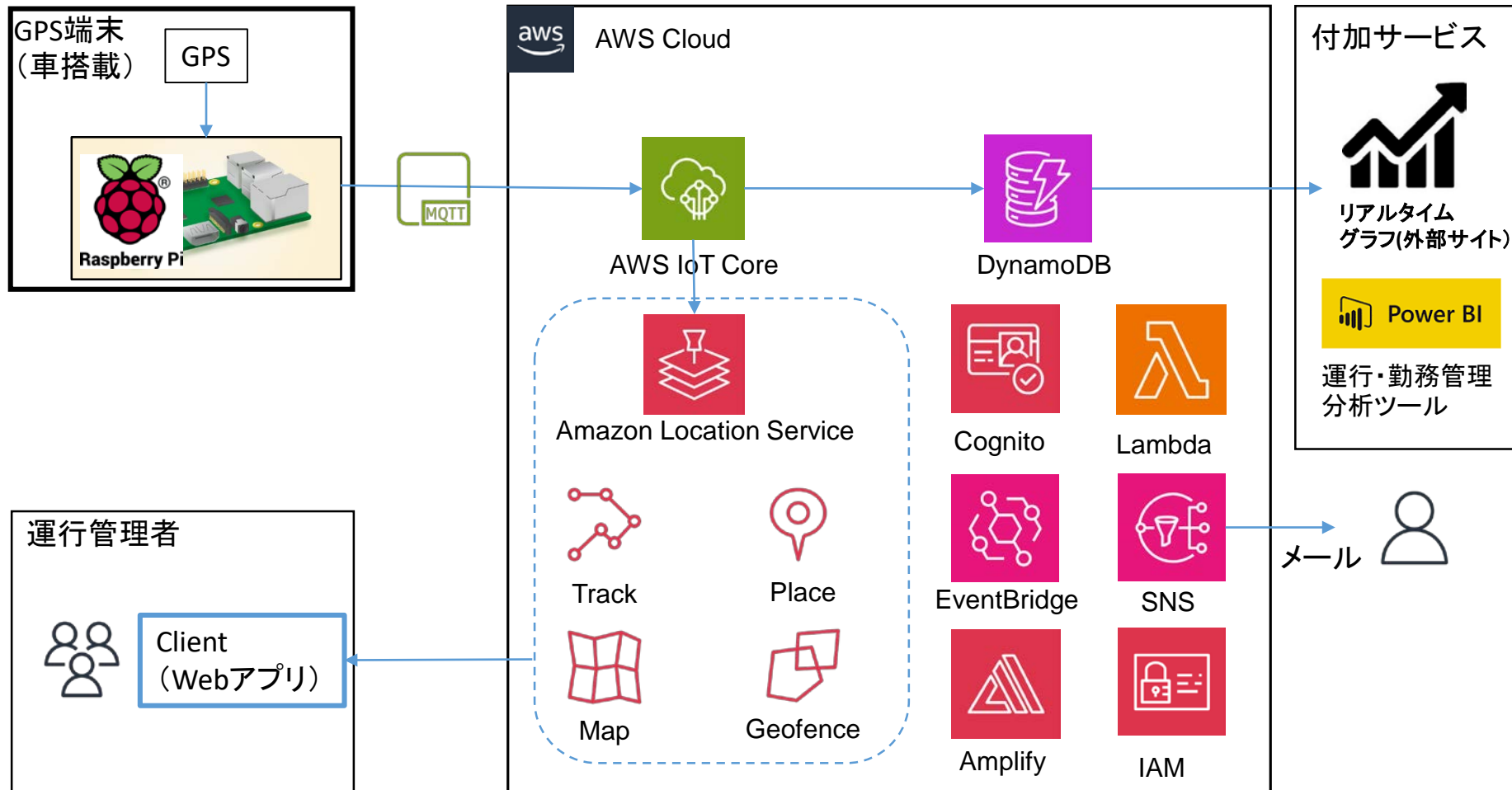
[sales1@spectrum-tech.co.jp](mailto:sales1@spectrum-tech.co.jp)

# 運行管理開発キット 目次

運行管理キット 全体像	<a href="#"><u>3</u></a>
GPS端末 概要	<a href="#"><u>4</u></a>
GPS端末 仕様	<a href="#"><u>5</u></a>
クライアント端末 概要	<a href="#"><u>6</u></a>
アプリ 一覧	<a href="#"><u>7</u></a>
1. 設定手順 全体像	<a href="#"><u>8</u></a>
事前準備	
2. 設定手順	<a href="#"><u>9</u></a>
① AWS IoT coreのモノ設定	<a href="#"><u>10</u></a>
② GPS端末設定	<a href="#"><u>22</u></a>
③ AWS IoT coreのルール設定	<a href="#"><u>32</u></a>
④ AWS IoT側での受信確認	<a href="#"><u>39</u></a>
⑤ クライアントPCにAWS Locationアプリ設定(Amplifyなど使用)	<a href="#"><u>42</u></a>
A) openlayers-amplify	<a href="#"><u>45</u></a>
B) maplibre-js-react-iot-asset-tracking	<a href="#"><u>51</u></a>
C) amplify-ui-geo-explore	<a href="#"><u>74</u></a>
⑥ GPS端末から正規データによる確認	<a href="#"><u>86</u></a>
⑦ 地図関係の各種機能(ジオフェンス、ルート、インデックス等)	<a href="#"><u>88</u></a>
3. その他	
⑧ 外部サイトによるデータ受信確認	<a href="#"><u>99</u></a>
⑨ 運行・勤務管理ツールによる分析	<a href="#"><u>105</u></a>
4. 運行管理のポイント	<a href="#"><u>109</u></a>

抜粋版のため本文  
と一致しません

# 運行管理開発キット 全体像



注: 運行管理者用clientPCはお客様に提供して頂きます。他のハード、ソフト、設定マニュアルは提供します。

# 運行管理開発キット GPS端末 概要

品名	項目	内容	備考
Raspberry Pi 4 model B	CPU	1.5GHz 64bit クアッドコアCortex-A72	
	GPU	デュアルコア VideoCore VI® 500MHz	
	メモリ	4GB RAM	
	OS	Raspbian bullseye(Debianベース)	
	インターフェース	2.4/5GHz WiFi(802.11 bgnac), Bluetooth 5.0, BLE, 1G ether, USB 2.0x2, USB 3.0x2, micro HDMIx2, microSD カード, 40 GPIO pin	モバイルルータ、テザリングでインターネットで接続
	電源／消費電力	Micro USB Type C 3.0A	
	サイズ	85x56x18mm	
GPS		UBX-M8030, GPS(米国)、QZSS(日本,みちびき3機対応)、GLONASS(ロシア)、Galileo(欧州)、BeiDou(中国) USB対応、1.5mケーブル、60g、精度:2m	マグネット、粘着シート付き
付属品		内容	備考
ケース		透明、白、黒から選択できます。	
microSD 32GB		Raspbian OS。お客様が設定するものは必要最低限のパスワード設定、WiFi設定になります。	
プログラム		運行管理用mqttプログラム(python3.9.2)	著作権は弊社に属します。コピー厳禁

USB電源ケーブル、HDMIケーブルは付属しておりません。

別途オプション品を購入ください



# 運行管理開発キット GPS端末 仕様

項番	端末電源	通信回線	GPS受信	GPSデータ送信 (MQTT)	内容	備考
1	○	○	○	○	10秒毎に送信	
2	○	×	○	○:回復後送信	10秒毎にGPS端末側でデータを保持し、通信回復後、データ送信。	
3	○	○	×	×	GPS受信後に送信、受信できない時間は、データ未送信	
4	○	×	×	×	GPS受信後にデータ送信システムとしては、動作中	
5	×			×	電源回復後、自動起動	

GPS端末は、起動時通信回線(約2分以内)が必要になります。起動後は、通信回線が断となってもシステムとして動作します。  
基本操作は、電源接続するだけです

GPSは、QZSS(みちびき), GPS/SBASを捕捉するため、端末起動後、約2分でGPSデータの送信を始めます。

# 運行管理開発キット クライアント端末 概要

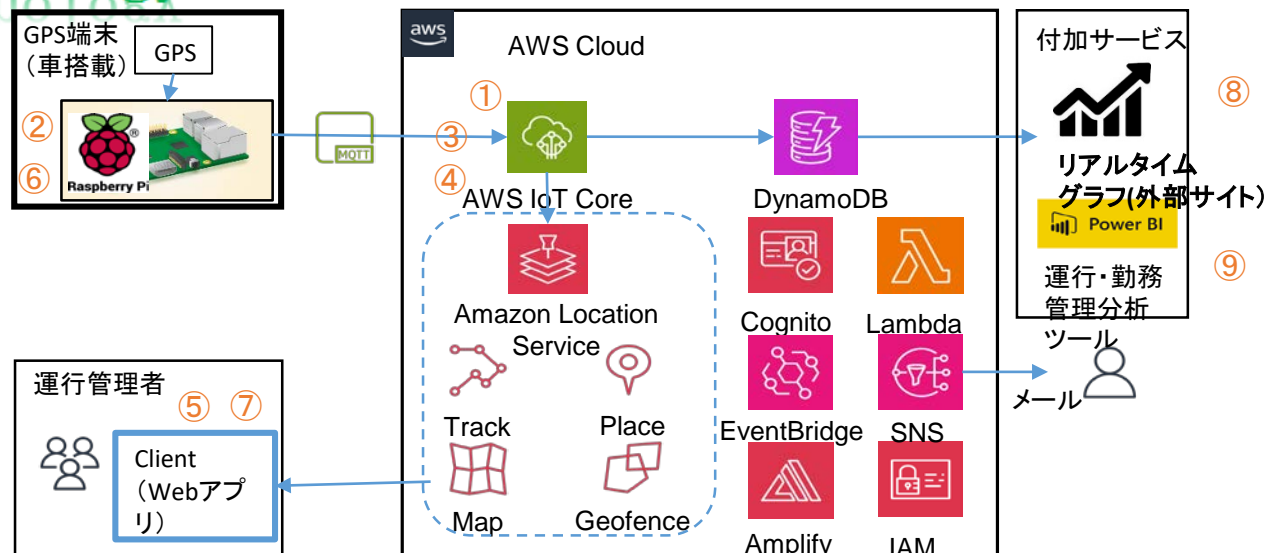
運行管理者用クライアント端末に必要なハード、ソフトです。

区分	項目	バージョン	備考
ハード	i5, 8GB メモリ、250GB HDD 以上	osは、windows又は Linux	お客様準備
ソフト	Node.js	V18.x以上	
	npm	V8.x以上	
	aws cli	V2.x以上	
	Amplify cli	V9.x以上	
	Visual studio code	V1.8以上	

# 運行管理開発キット アプリ 一覧

項番	区分	アプリ名	AWSサービス名	内容	頻度	備考
1	基本機能	トラッキング	AWS location service/tracker	車などに搭載したGPS情報からリアルタイムに自動追尾。精度は、2m。履歴はdynamoDBに蓄積	10秒毎に更新	
2		ジオフェンス	AWS location service/geofence	地図上に設定したエリアに対象物体が入出力した時点でアラーム(メール)通知	入出力時点	
3		ルート計算	AWS location service/route	地図で、出発地と目的地を設定して、最適ルートを計算表示	都度	
4		場所検索	AWS location service/place index	検索したい場所、企業などを入力し、その場所を明示。緯度経度からの逆引き、カテゴリ(ホテル、飛行場)による検索も可能	都度	
5		地図	AWS location service/map	通常の地図、夜間、衛星などの地図を選択可能。日本語対応	設定時	日本語対応地図を提供します。
6		メール等通知	AWS SNS	メール、SMSなどで通知	イベント発生時	
7	付加機能	速度表示	AWS location service/tracker	10秒毎に進んだ距離に基づき、速度(km/h)を表示	10秒毎	
8		ドライバ表示	AWS location service/tracker	ドライバをお客様で自由に設定	GPS端末毎	
9		リアルタイムモニタ		GPS情報がDynamoDBに入力されたかを外部Webサイトで確認	リアルタイム	
10		運行・勤務管理分析ツール		運行・勤務管理を分析するツールpowerBI(マイクロソフト、無償)でサンプル版を提供	バッチ処理	dynamoDBから抽出したcsvをそのままインポート可能
11	応用機能	GNSS-RTK		RTK(Real Time Kinematic)をHATで搭載し、精度:20cmで位置情報を提供	別途	弊社でカスタマイズして提供
12		センサ情報		温湿度、9軸センサ(加速度、ジャイロ、地磁気)、I2Cで接続できるセンサ類	別途	弊社でカスタマイズして提供
13		カメラ		ライブカメラ等により提供	別途	弊社でカスタマイズして提供

## 設定手順全体像



### 1. 事前準備

- AWS、Javascriptの基礎スキルが必要になります。AWSのアカウント設定は終了していること。
- GPS端末用のテザリング、モバイルルータなどのインターネット回線準備
- クライアントPCの準備 (i5, 8GB メモリ、250GB HDD, windows又はlinux)

### 2. 設定手順は、以下のとおり実施します。

- ① AWS IoT coreのモノ設定
- ② GPS端末設定
- ③ AWS IoT coreのルール設定
- ④ AWS IoT側での受信確認
- ⑤ クライアントPCにAWS Locationアプリ設定 (Amplifyなど使用)。
- ⑥ GPS端末から正規データによる確認
- ⑦ 地図関係の各種機能(ジオフェンス、ルート、インデックス等)

### 3. その他

- ⑧ 外部サイトによるデータ受信確認
- ⑨ 運行・勤務管理ツールによる分析



# 運行管理開発

## 2. 設定手順

## 運行管理開発

## ①. AWS IoT coreモノ設定

## • AWSトップ画面

- 利用する場合に、地域を意識して設定してください。地域毎に料金が変わったり、利用できるサービスが限定されている場合があります。以下ドキュメント参照
- 基本的にどのリージョンでも設定可能。弊社例は、ohio:us-east-2になります。
- Iot coreとlocationなどは同じregionになります。

## ①. AWS IoT coreモノ設定



AWS IoT Core

- モノの登録1
  - 管理>モノの作成を押します。

The screenshot shows the AWS IoT Core console interface. The left sidebar contains navigation links for 'モノ' (Things), 'モノのグループ' (Thing Groups), 'モノのタイプ' (Thing Types), 'フリーメトリクス' (Free Metrics), 'Greengrass デバイス' (Greengrass Devices), 'リモートアクション' (Remote Actions), 'メッセージのルーティング' (Message Routing), '保持されたメッセージ' (Stored Messages), 'セキュリティ' (Security), 'イントロダクション' (Introduction), '証明書' (Certificates), 'ポリシー' (Policies), '認証機関' (Authentication), 'ロールエイリアス' (Role Aliases), 'オーソライザー' (Authorizers), '監査' (Auditing), '検出' (Detection), and 'アラーム' (Alarms). The main content area displays the 'モノ (16)' page, which includes a description of IoT Things, a search bar, and a table of existing things. The 'モノを作成' button is highlighted with a red box.

名前	モノのタイプ
pi4	-
0123539ac9de3ef901	-
test0506	-
helloworld_publisher2	-
jetson2_Core	-
gg_switch	-
gg_trafficlight	-
helloworld_subscriber	-
helloworld_publisher	-
jetson_Core	-
stm32	-
pi3	-
pi2	basic

## ①. AWS IoT coreモノ設定



AWS IoT Core

## • エンドポイント

- pi側に設定するエンドポイント名です。
- 設定>エンドポイント

The screenshot shows the AWS IoT Core console interface. On the left, the navigation menu is visible with '設定' (Settings) highlighted. The main content area is titled 'AWS IoT > 設定' (Settings). Under '設定 情報' (Settings Information), the 'デバイスデータエンドポイント' (Device Data Endpoint) section is expanded, showing the endpoint 'a2' and the URL 'j-ats.iot.us-east-2.amazonaws.com'. Below this, the 'ドメイン設定' (Domain Settings) section is visible, showing a table with columns for Name, Domain Name, Status, Service Type, and Update Date. The table is currently empty, with a message indicating that domain settings are not yet configured. At the bottom, the 'ログ' (Logs) section is visible, showing a message about CloudWatch Logs.

デバイスデータエンドポイント 情報

デバイスは、アカウントのデバイスデータエンドポイントを使用して AWS に接続できます。

各モノには、このエンドポイントで使用可能な REST API があります。MQTT クライアントと [AWS IoT デバイス SDK](#) もこのエンドポイントを使用します。

エンドポイント

a2 j-ats.iot.us-east-2.amazonaws.com

セキュリティポリシーを選択 情報

TLS バージョンやサポートされている暗号スイートなどの TLS 設定をカスタマイズするには、セキュリティポリシーを選択します。

IoTSecurity 16\_01

セキュリティポリシーを比較

ドメイン設定 情報

ドメイン設定を作成して、デバイスの AWS IoT Core への移行、アプリケーションインフラストラクチャの AWS IoT Core への移行、ブランドアイデンティティの維持などのタスクを簡素化できます。

アクション ドメイン設定を作成

名前	ドメイン名	ステータス	サービスタイプ	更新日
ドメイン設定がありません ドメイン設定がありません。				

ドメイン設定を作成

ログ 情報

CloudWatch Logs に有益な情報を記録するために、AWS IoT ログ記録を管理できます。

ログの管理

デバイスからのメッセージがメッセージブローカーとルールエンジンを通過すると、AWS IoT はトラブルシューティングに役立つプロセスイベントをログに記録します。

ロール

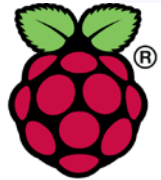


## 運行管理開発

### ②. GPS端末設定 接続構成



Raspberry Pi 4B



Raspberry Pi

microHDMI

USB

モニタ

WiFiにより、テザリ  
ング、モバイル  
ルータ等に接続

インター  
ネット

GPSをusb端子に接続



USB電源 (typeC) : 5V3A(PCの端子は2.1Aで動作しない)

## 運行管理開発

## ②. GPS端末設定

## 1. Raspberry Pi起動

## ① マイクロSDカードを挿入



## ② 電源ケーブル接続



- ・Raspberry Piの裏面からマイクロSDカードを挿入します。
- ・Raspbian OS(Bullseye系列)と必要なモジュール、ドライバをインストール済です。

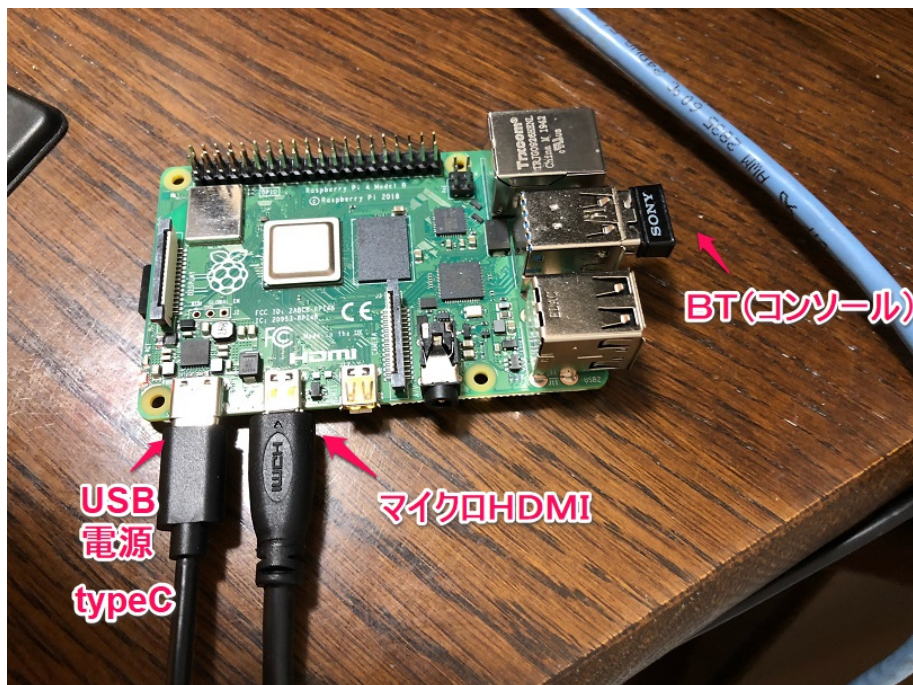
- ・Raspberry PiのマイクロUSB端子に電源コードを挿入します。
- ・電源コードは付属しておりません。必要な場合は、オプションで販売しております。5V3A必要なため、AC100Vから変換器経由で給電してください。**パソコンのusbの場合起動しないことがあります。**

## ②. GPS端末設定

## 2. モニター、キーボード接続

Raspberry Piとモニタ(HDMI接続)、マウス、キーボード(USB接続)を接続し、初期設定を行います。

ノートPCの場合は、  
表示されません



## ① モニタ接続

- モニター(TV、PCでHDMI端子のあるもの)を準備します。
- マイクロHDMIケーブルにより、Raspberry Piとモニタを接続します。

## ② マウス、キーボード接続

- デスクトップPCに接続しているマウス、キーボードをUSBでRaspberry Piと接続します。
- マウス、キーボードがBluetoothで接続されている場合は、本体のBluetooth USBを外してRaspberry Piと接続します。



## ②. GPS端末設定

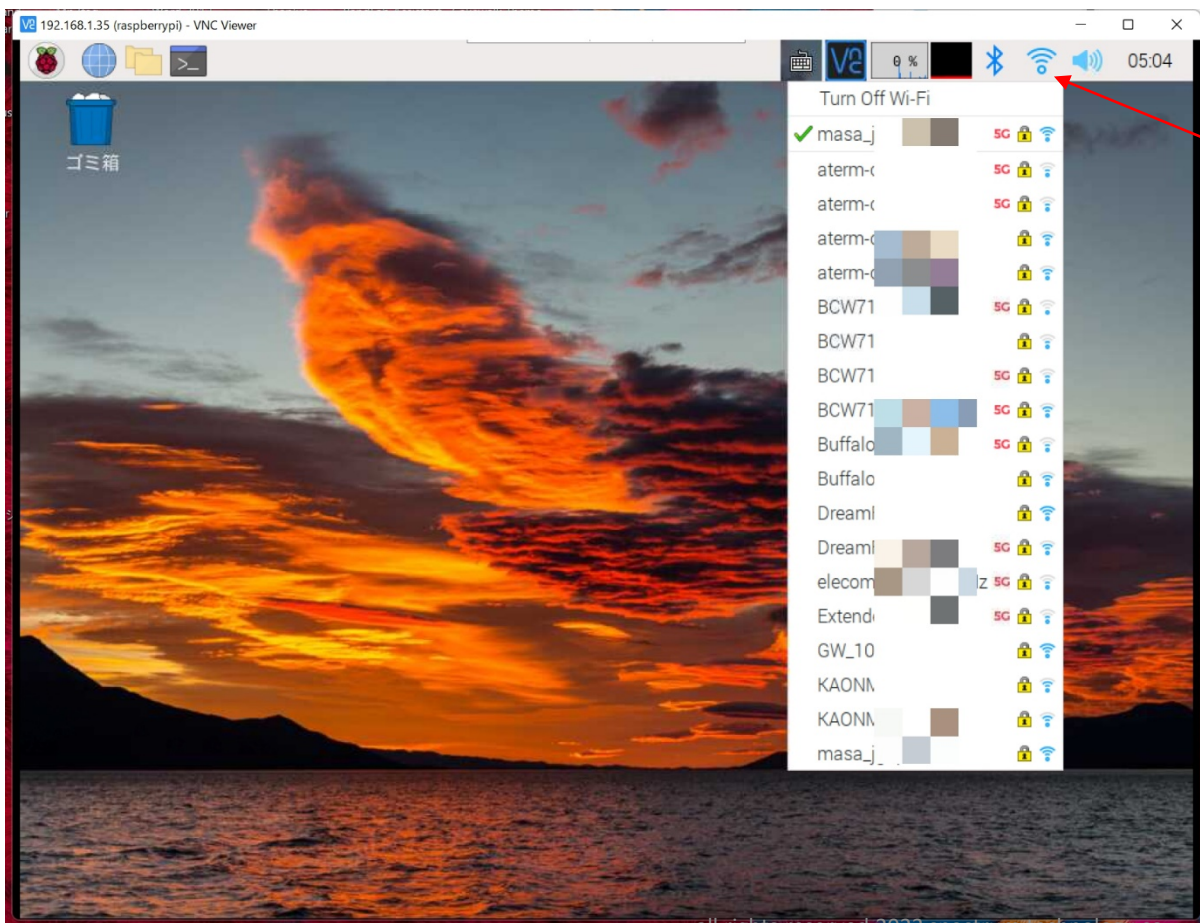
## 3. 初期設定

## (2) ネットワーク設定

Raspberry Piが立ち上がったことを確認し、各種設定を行います。



Raspberry Pi



## ⑤ ネットワーク設定

## • WiFiの場合

- 上段のLANマークをクリックして、接続したい、WiFi(SSID)を選びます。
- パスワードを入力します。
- 次回からは自動で接続されます。

## • LANの場合

- 自動で接続されます。
- IPアドレスは、LAN上のDHCPから自動で取得します。
- IPアドレスをマニュアルで設定することも可能です。

## 運行管理開発

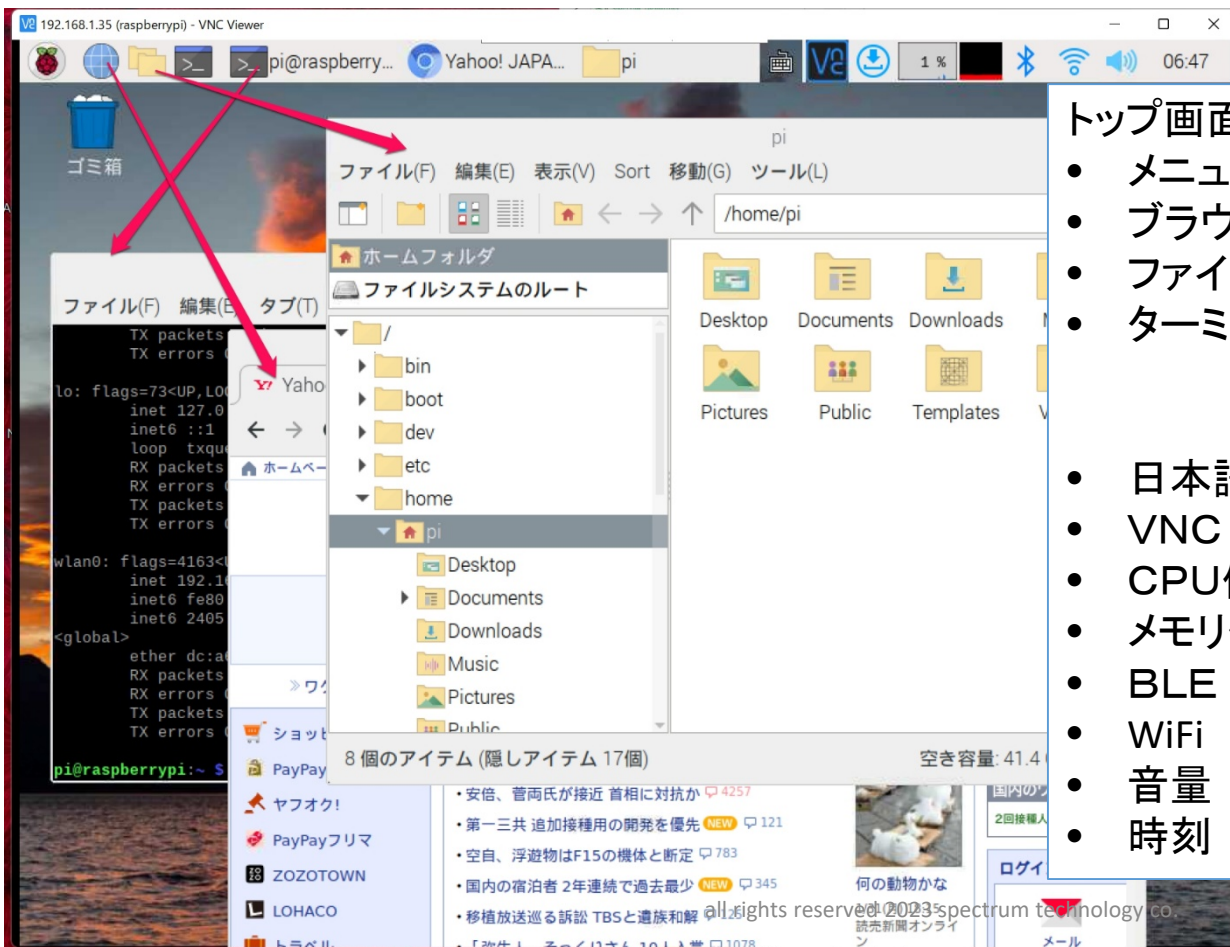
## ②. GPS端末設定



Raspberry Pi

## 3. 初期設定

## (3) 画面説明



トップ画面(上段のタスクバーで選択)

- メニュー
- ブラウザ
- ファイルマネージャ
- ターミナル

- 日本語入力
- VNC
- CPU使用率
- メモリ使用量
- BLE
- WiFi
- 音量
- 時刻

## ③AWS IoT coreのルール設定



AWS IoT Core

## AWS IoTのルール設定

- AWS IoTのコンソールからACT>ルール
- ルール作成を押します。

ルール (9) 情報

ルールを使用すると、モノは他のサービスとインタラクションできます。ルールは分析され、デバイスによって発行されたメッセージに基づいて特定のアクションが実行されます。

リフレッシュ 有効化 無効化 編集 削除 **ルールを作成**

ルールを検索

<input type="checkbox"/>	名前	ステータス	ルールのトピック	作成日
<input type="checkbox"/>	bme280_rule	アクティブ	pi3	November 12, 2019, 08:07:12 (UTC+09:00)
<input type="checkbox"/>	bmp280_rule	アクティブ	pi1	February 02, 2018, 11:07:41 (UTC+09:00)
<input type="checkbox"/>	ct_rule	アクティブ	ct1	December 25, 2018, 16:15:23 (UTC+09:00)
<input type="checkbox"/>	ct2_rule	アクティブ	ct2	December 25, 2018, 17:54:02 (UTC+09:00)
<input type="checkbox"/>	IoTAnalytics_mychannel_ct1	アクティブ	ct1	September 02, 2018, 13:46:39 (UTC+09:00)
<input type="checkbox"/>	myevent_rule	アクティブ	pi3	November 17, 2019, 09:40:26 (UTC+09:00)
<input type="checkbox"/>	myevnet_ccs_rule	アクティブ	pi3	April 03, 2020, 09:21:27 (UTC+09:00)
<input type="checkbox"/>	myiot_analytics_pi_topicrule	アクティブ	ct2	November 12, 2019, 15:29:27 (UTC+09:00)
<input type="checkbox"/>	myiotSNS	アクティブ	topic/mysns	December 19, 2018, 17:02:00 (UTC+09:00)

## 運行管理開発

## ③AWS IoT coreのルール設定



AWS IoT Core

## AWS IoTのルール設定

- ルール作成
- 名前を入力します
- ルールクエリステートメントにSQL文を入力します。

名前: mytrackrule1  
SELECT \* FROM 'track1'

ルールを作成

ルールのプロパティを指定

ルール名: track1\_rule

SQLステートメント: SELECT \* FROM 'track1'

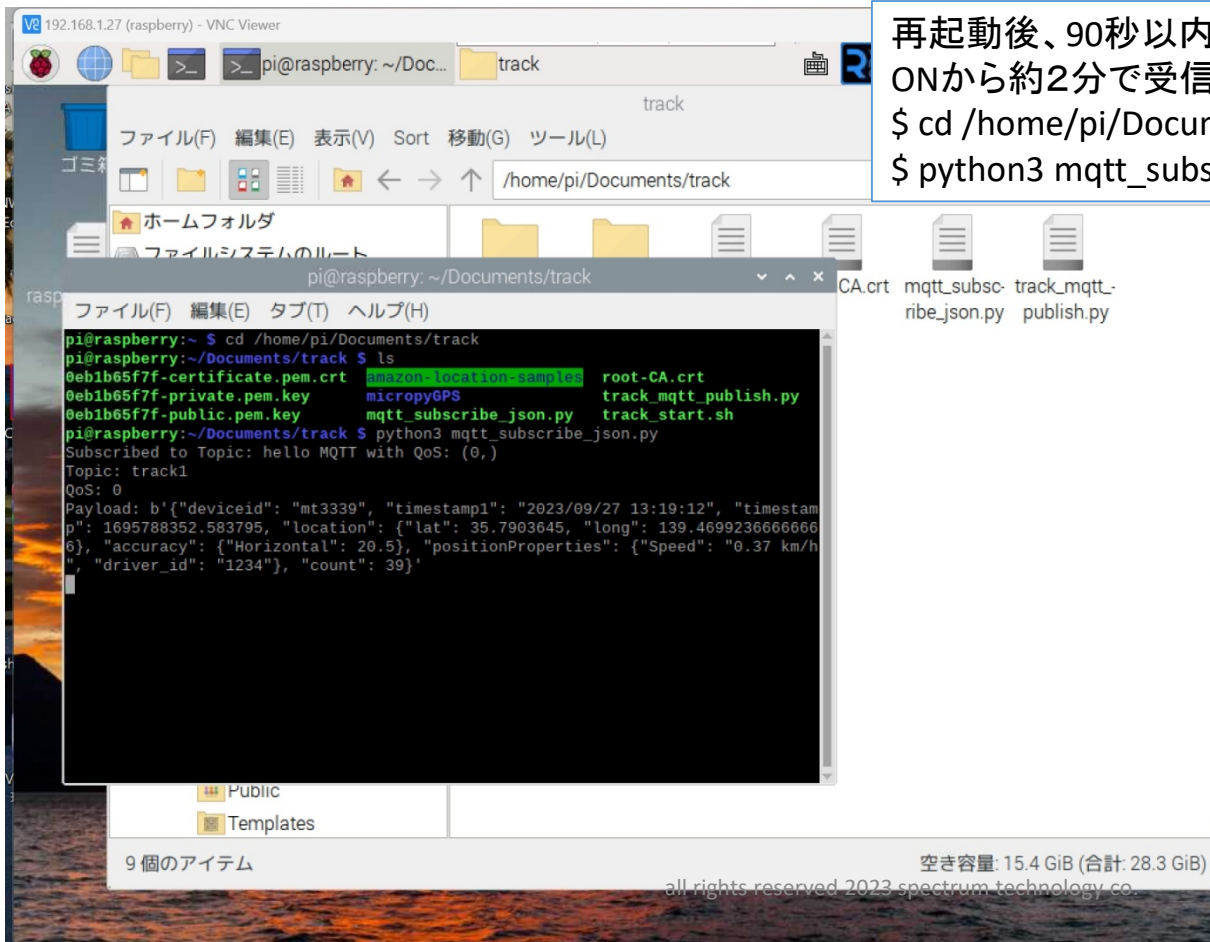
pi側のトピックの登録をtrack1で実施した場合。



## ④. AWS IoT側での受信確認

### AWS IoT折り返し試験

- Piのコマンド画面からsubscriberの確認
- subscribe側にデータが受信できていればOk



```
pi@raspberrypi: ~/Documents/track
$ cd /home/pi/Documents/track1
$ python3 mqtt_subscribe_json.py

Subscribed to Topic: hello MQTT with QoS: (0,)
Topic: track1
QoS: 0
Payload: b'{"deviceid": "mt3339", "timestamp": "2023/09/27 13:19:12", "timestamp": "1695788352.583795", "location": {"lat": 35.7903645, "long": 139.46992366666666}, "accuracy": {"Horizontal": 20.5}, "positionProperties": {"Speed": "0.37 km/h", "driver_id": "1234"}, "count": 39}'
```

再起動後、90秒以内に回線接続、GPSを受信状態。電源ONから約2分で受信できます。

```
$ cd /home/pi/Documents/track1
```

```
$ python3 mqtt_subscribe_json.py 受信側を先に立ち上げ
```

#### 受信できない場合の理由

- 証明書が合っていない
- AWS IoT core側のトピックがtrack1になっていない





## ④. AWS IoT側での受信確認

AWS IoT coreのテスト画面で確認

- Piの電源を入れて、前頁で動作確認後
- MQTT テストクライアントでトピックスを”track1”と入力し、サブスクライブ
- 下にデータ受信できれば、iot coreまでの通信は良好

トピックをサブスクライブする トピックに公開する

トピックのフィルター 情報  
トピックフィルタは、サブスクライブするトピックを記述します。トピックフィルタには、MQTT ワイルドカード文字を含めることができます。

track1

追加設定

サブスクライブ

サブスクリプション track1 一時停止 クリア エクスポート

track1

メッセージペイロード

```
{
  "message": "Hello from AWS IoT console"
}
```

追加設定

実行

track1 September 27, 2023, 13:15:41 (UT)

```
{
  "deviceId": "mt3339",
  "timestamp": "2023/09/27 13:15:52",
  "timestamp": 1695788152.339176,
  "location": {
    "lat": 35.790423,
    "long": 139.46983983333334
  },
  "accuracy": {
    "Horizontal": 20.5
  },
  "positionProperties": {
    "Speed": "0.07 km/h",
    "driver_id": "1234"
  },
  "count": 19
}
```

受信データ

all rights reserved 2023 spectrum technology co.



## 運行管理開発

## ⑤クライアントPCにAWS Locationアプリ設定

## 2. 事前準備

- [https://docs.aws.amazon.com/ja\\_jp/cli/latest/userguide/getting-started-install.html](https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/getting-started-install.html)
- iamからアクセスkeyなどを取得
- 端末からC:>aws configureと入力し設定
- Access keyなどは各自入力してください。
- >aws -hで内容が出れば使用可能です。

```
コマンド プロンプト
Microsoft Windows [Version 10.0.22621.2361]
(c) Microsoft Corporation. All rights reserved.

C:\Users\skyri>aws configure
AWS Access Key ID [*****EM5Z]:
AWS Secret Access Key [*****th+2]:
Default region name [us-east-2]:
Default output format [json]:

C:\Users\skyri>aws -h
usage: aws [-h] [-i profile PROFILE] [--debug]

options:
  -h, --help            show this help message and exit
  --profile PROFILE
  --debug
```

## ⑤クライアントPCにAWS Locationアプリ設定

## 入カコマンド

- npm install -g @aws-amplify/cli
- npm install
- amplify configure
- amplify init
- amplify push



Amazon Location Service

## 3. locationアプリ設定

## A) openlayers-amplify

- <https://github.com/aws-samples/amazon-location-samples/tree/main/openlayers-amplify>
- インストール
  - コマンドプロンプトを管理者で開く
  - 使用するディレクトリに移動
    - npm install -g @aws-amplify/cli
    - npm install

```
Microsoft Windows [Version 10.0.22621.2361]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32\F:
F:\>cd F:\技術関係\2m\gps_logger\amazon-location-samples\openlayers-amplify2
F:\技術関係\2m\gps_logger\amazon-location-samples\openlayers-amplify2>npm install -g @aws-amplify/cli
changed 26 packages in 1m
7 packages are looking for funding
  run `npm fund` for details

F:\技術関係\2m\gps_logger\amazon-location-samples\openlayers-amplify2>npm install
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm WARN deprecated @babel/plugin-proposal-optional-catch-binding@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-catch-binding instead.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm WARN deprecated @babel/plugin-proposal-object-rest-spread@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-object-rest-spread instead.
npm WARN deprecated @babel/plugin-proposal-async-generator-functions@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-async-generator-functions instead.
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uglify-es@3.9.0: support for ECMAScript is superseded by 'uglify-js' as of v3.13.0
added 1309 packages, and audited 1310 packages in 44s
68 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

F:\技術関係\2m\gps_logger\amazon-location-samples\openlayers-amplify2>
```



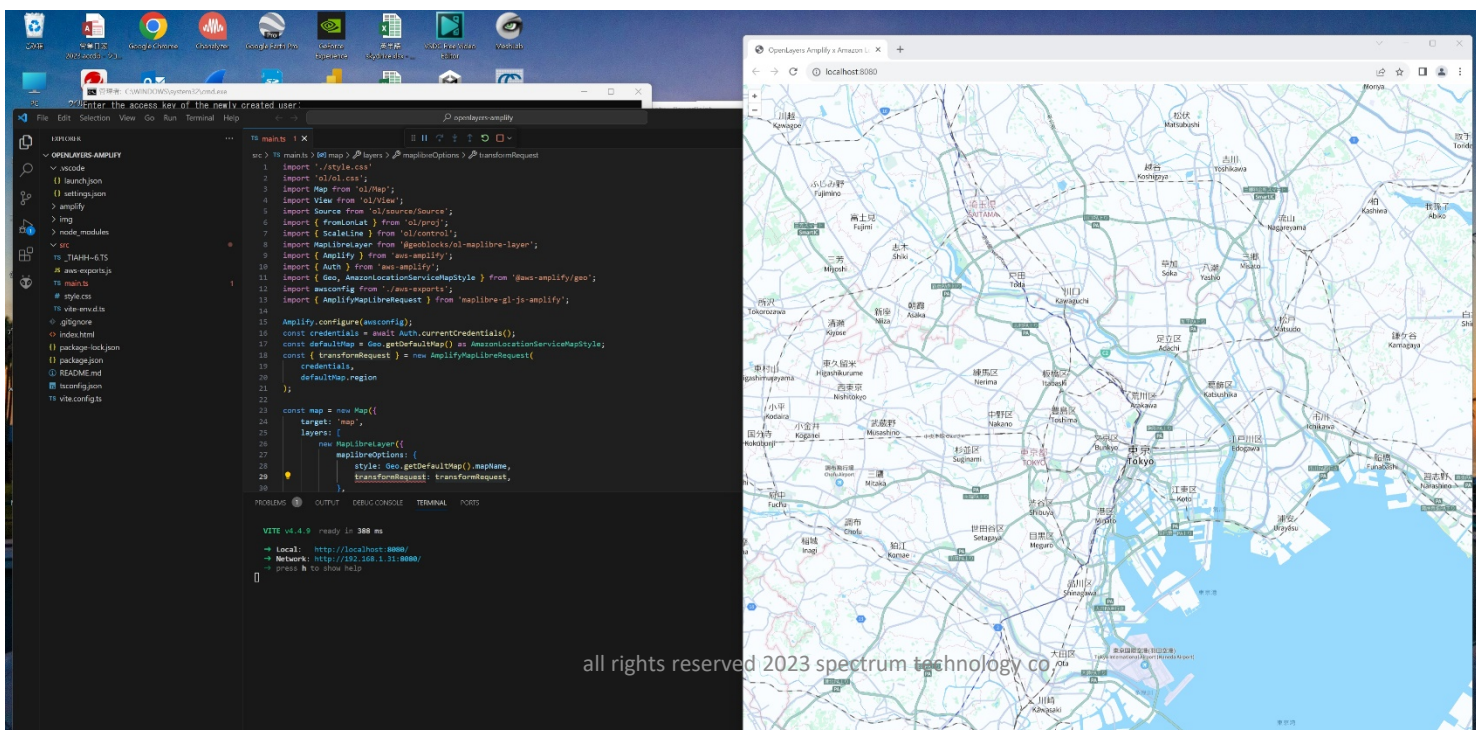
## 運行管理開発

## ⑤クライアントPCにAWS Locationアプリ設定

## 3. locationアプリ設定

## A) openlayers-amplify

- <https://github.com/aws-samples/amazon-location-samples/tree/main/openlayers-amplify>
- 起動
  - npm start
  - Run>without debugでweb(chrome)を選択すると自動表示





## ⑤クライアントPCにAWS Locationアプリ設定

## 入カコマンド

- npm install
- npm install -g @aws-amplify/cli
- amplify configure
- amplify init



Amazon Location Service

## 3. locationアプリ設定

## B) maplibre-js-react-iot-asset-tracking

mqttを使って追尾を行うモデル

<https://github.com/aws-samples/amazon-location-samples/tree/main/maplibre-js-react-iot-asset-tracking>

## インストール

- npm install
- npm install -g @aws-amplify/cli
- amplify configure
- amplify init

基本デフォルト  
でok

```
管理者: C:\WINDOWS\system32\cmd.exe
run npm fund for details
C:\Users\user>npm install
C:\Users\user>npm install -g @aws-amplify/cli
C:\Users\user>amplify configure
Follow these steps to set up access to your AWS account:
1. Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue
2. Specify the AWS Region:
region: us-east-2
Follow the instructions at
https://docs.aws.amazon.com/amplify/cli/latest/userguide/amplify-cli.html
to complete the user creation in the AWS console
https://console.aws.amazon.com/iamv2/home#/users/create
Press Enter to continue
Enter the access key of the newly created user:
? accessKeyId: *****
? secretAccessKey: *****
This would update/create the AWS Profile in your local machine
Profile Name: default
Successfully set up the new user.
C:\Users\user>amplify init
Note: It is recommended to run this command from the root of your app directory
? Enter a name for the project maplibrejsreactiotas
The following configuration will be applied:
Project information
Name: maplibrejsreactiotas
Environment: dev
Default editor: Visual Studio Code
App type: javascript
Javascript framework: react
Source Directory Path: src
Distribution Directory Path: build
Build Command: npm.cmd run-script build
Start Command: npm.cmd run-script start
? Initialize the project with the above configuration? Yes
Using default provider awscloudformation
? Select the authentication method you want to use: AWS profile
For more information on AWS Profiles, see:
https://docs.aws.amazon.com/cli/latest/userguide/cli-config-profiles.html
? Please choose the profile you want to use default
Adding backend environment dev to AWS Amplify app: d32f7b1wgfy5m
Deployment completed.
Deploying root stack maplibrejsreactiotas [=====] 2/4
amplify-maplibrejsreactiotas... AWS::CloudFormation::Stack CREATE_IN_PROGRESS Thu Sep 28 2023 0
AuthRole AWS::IAM::Role CREATE_COMPLETE Thu Sep 28 2023 0
DeploymentBucket AWS::S3::Bucket CREATE_IN_PROGRESS Thu Sep 28 2023 0
UnauthRole AWS::IAM::Role CREATE_COMPLETE Thu Sep 28 2023 0
C:\Users\user>
all rights reserved 2023 spectrum technology co.
You can always opt-in by running "amplify configure --share-project-config-on"
Deployment state saved successfully.
```



## 運行管理開発

## ⑤クライアントPCにAWS Locationアプリ設定

## 3. locationアプリ設定

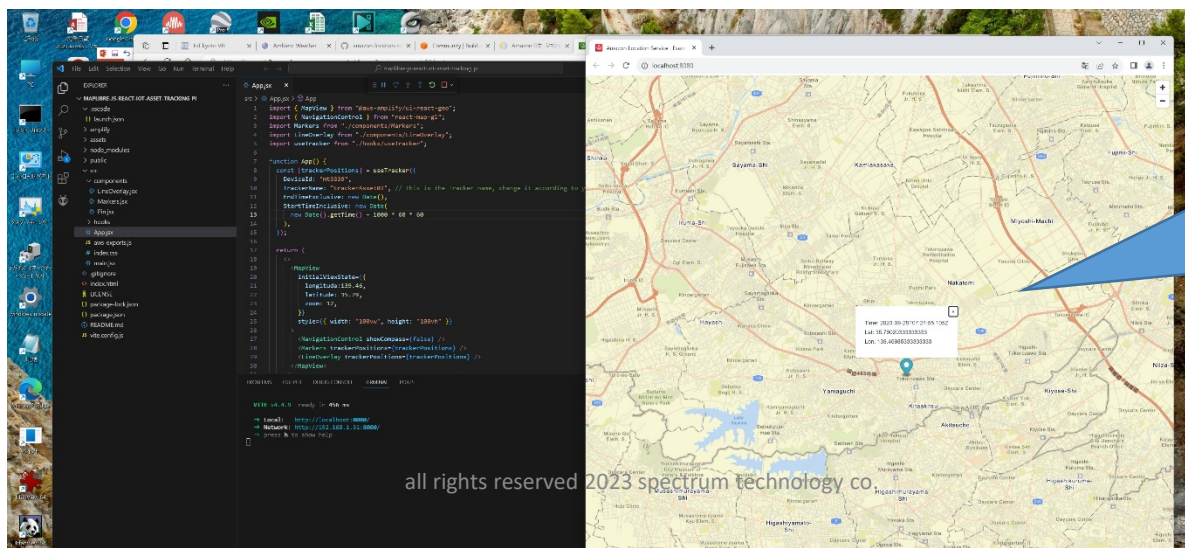
## B) maplibre-js-react-iot-asset-tracking

mqttを使って追尾を行うモデル

<https://github.com/aws-samples/amazon-location-samples/tree/main/maplibre-js-react-iot-asset-tracking>

AssetTrackingRule\_piのテスト

- lot coreのテストクライアント、疑似データを入れてトピックを公開します
- Visual studio code立ち上げ、該当フォルダを開き、new terminal
  - npm start
  - Run>without debugでweb(chrome)を選択すると自動表示



- ピンが表示され、時間が合っていれば正常
- 時間、緯度経度を変更すると履歴が表示されます。

## 運行管理開発

## ⑤クライアントPCにAWS Locationアプリ設定

## 3. locationアプリ設定

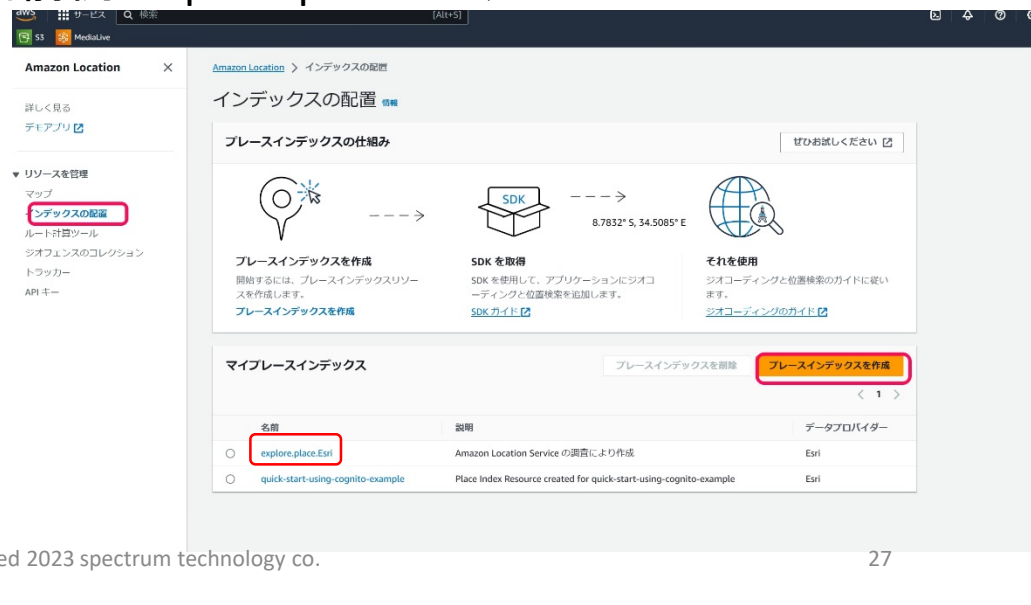
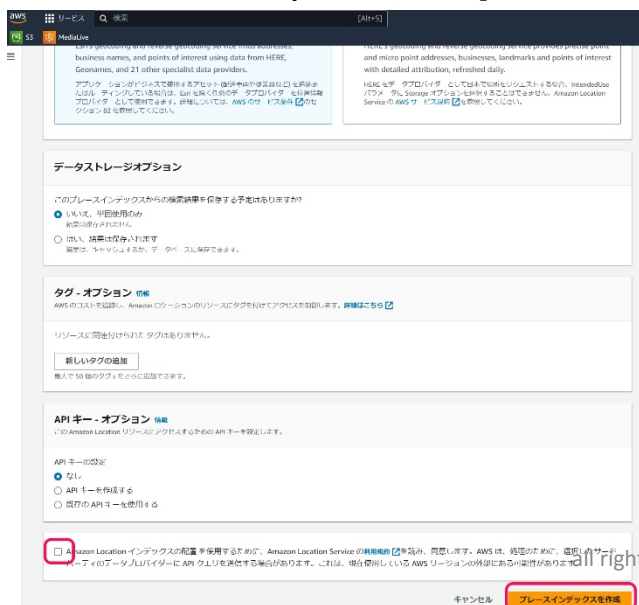
## C) amplify-ui-geo-explore

A)、B) 作成したモデルを統合した最終版

<https://github.com/aws-samples/amazon-location-samples/tree/main/amplify-ui-geo-explore>

## 事前準備

- Aws locationのうち、未設定分を設定
- インデックスの配置(場所名を入れて検索する時に使用)
- 他はデフォルトでok(名前例: explore.place.Esri)





## 28



# 運行管理開発

## ⑦地図関係の各種機能(ジオフェンス、ルート、インデックス等)

C) amplify-ui-geo-explore

A)、B) 作成したモデルを統合した最終版

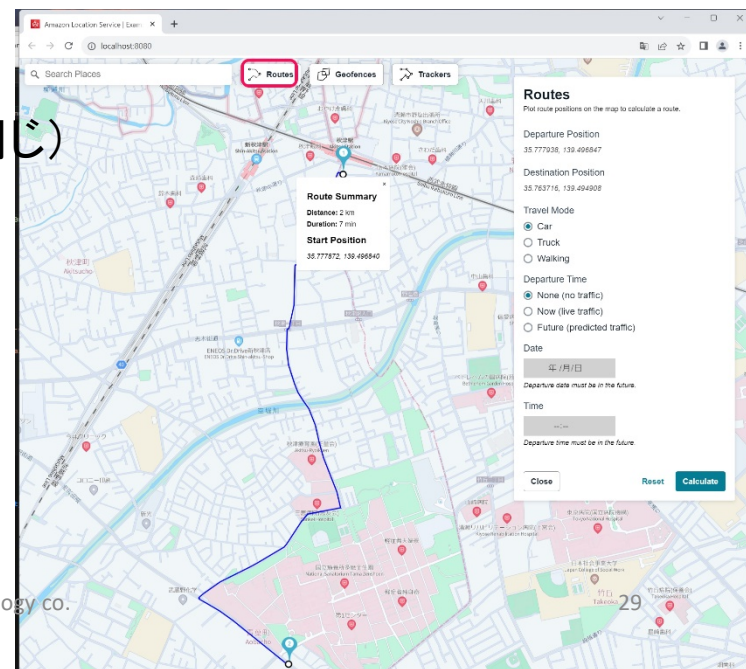
<https://github.com/aws-samples/amazon-location-samples/tree/main/amplify-ui-geo-explore>

ルート

[https://docs.aws.amazon.com/ja\\_jp/location/latest/developerguide/calculating-routes.html](https://docs.aws.amazon.com/ja_jp/location/latest/developerguide/calculating-routes.html)

地図上でプロットした2点間での車のルートを表示

- ルートを選択、地図上で2点をプロット
- 計算を押すとルートと所要時間が出る。
- 出発時間の指定なども可能(googleと同じ)



# 運行管理開発

## ⑦地図関係の各種機能(ジオフェンス、ルート、インデックス等)

C) amplify-ui-geo-explore

A)、B) 作成したモデルを統合した最終版

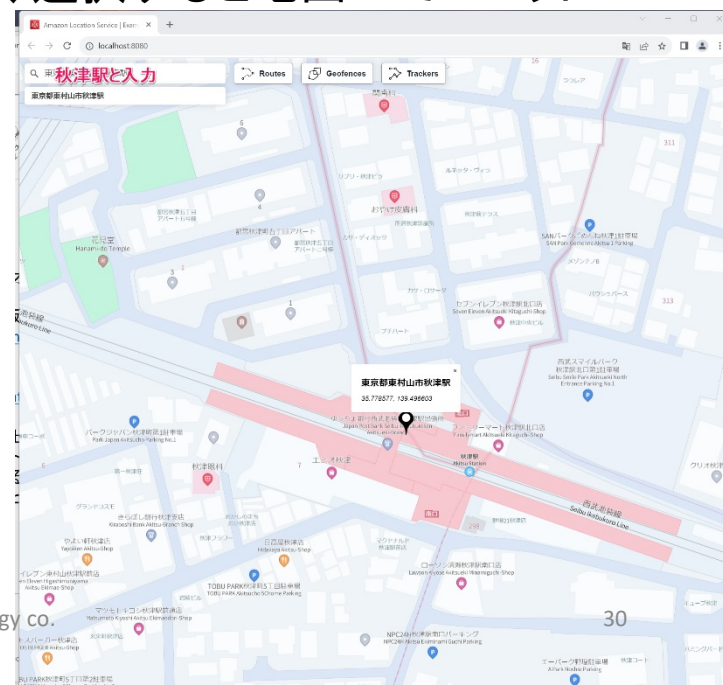
<https://github.com/aws-samples/amazon-location-samples/tree/main/amplify-ui-geo-explore>

インデックス(場所検索)

[https://docs.aws.amazon.com/ja\\_jp/location/latest/developerguide/searching-for-places.html](https://docs.aws.amazon.com/ja_jp/location/latest/developerguide/searching-for-places.html)

検索窓に、場所名を入力すると候補が出て、選択すると地図上でプロット

- 秋津駅と入力
- リスト表示、選択すると地図上にプロット



# 運行管理開発

## ⑦地図関係の各種機能(ジオフェンス、ルート、インデックス等)

C) amplify-ui-geo-explore

A)、B) 作成したモデルを統合した最終版

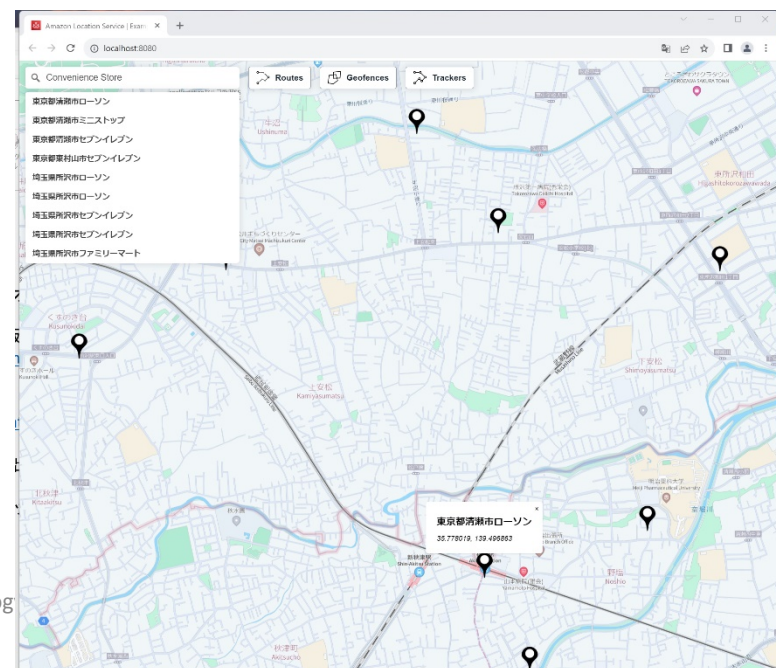
<https://github.com/aws-samples/amazon-location-samples/tree/main/amplify-ui-geo-explore>

インデックス(場所検索、カテゴリ)

[https://docs.aws.amazon.com/ja\\_jp/location/latest/developerguide/category-filtering.html](https://docs.aws.amazon.com/ja_jp/location/latest/developerguide/category-filtering.html)

検索窓に、カテゴリを入力すると候補が出て、地図上に複数表示(英語のみ)

- Convenience Storeと入力
- リスト表示、選択すると地図上にプロット





# 運行管理開発

## ⑦地図関係の各種機能(ジオフェンス、ルート、インデックス等)

C) amplify-ui-geo-explore

A)、B) 作成したモデルを統合した最終版

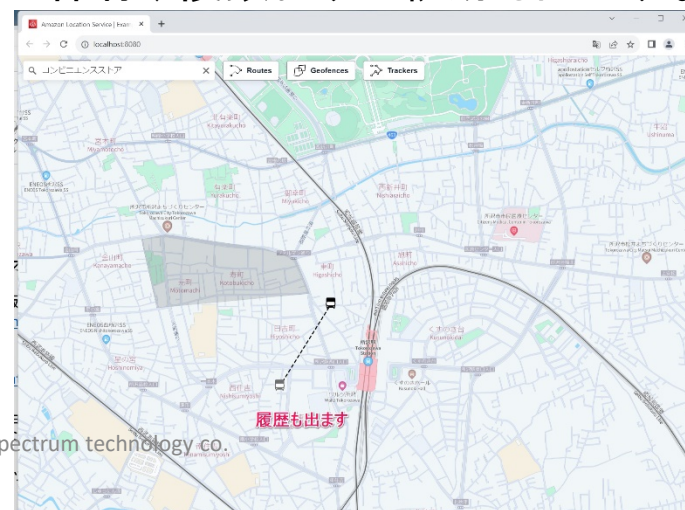
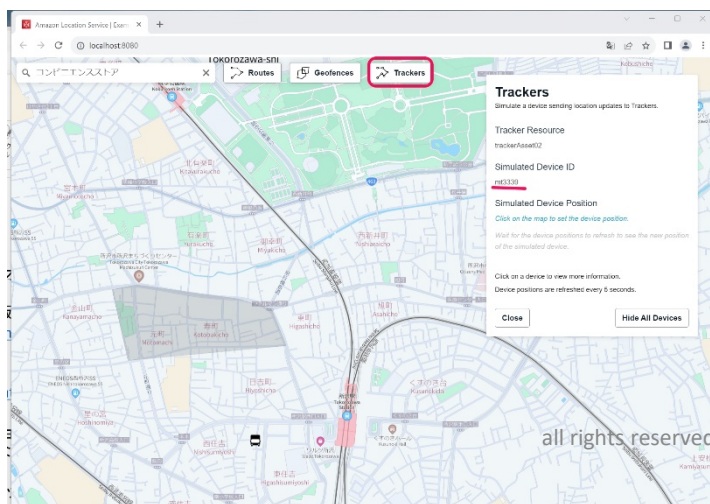
<https://github.com/aws-samples/amazon-location-samples/tree/main/amplify-ui-geo-explore>

トラッカー・シミュレーション

[https://docs.aws.amazon.com/ja\\_jp/location/latest/developerguide/geofence-an-area.html](https://docs.aws.amazon.com/ja_jp/location/latest/developerguide/geofence-an-area.html)

トラッカーを押すと、追尾のシミュレーションができます。実際の履歴にも残るのでテスト期間中にのみ使用または、特別なdeviceidを使用のこと

- トラッカーを押す、右にシミュレーション画面。
- 地図上にプロットすると履歴として保存、複数おすと軌跡も出ます。



## 運行管理開発

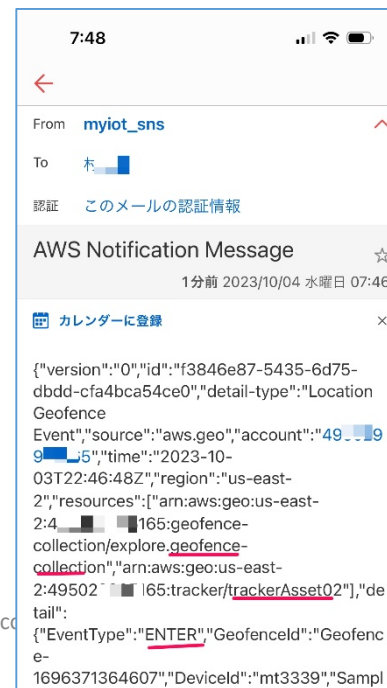
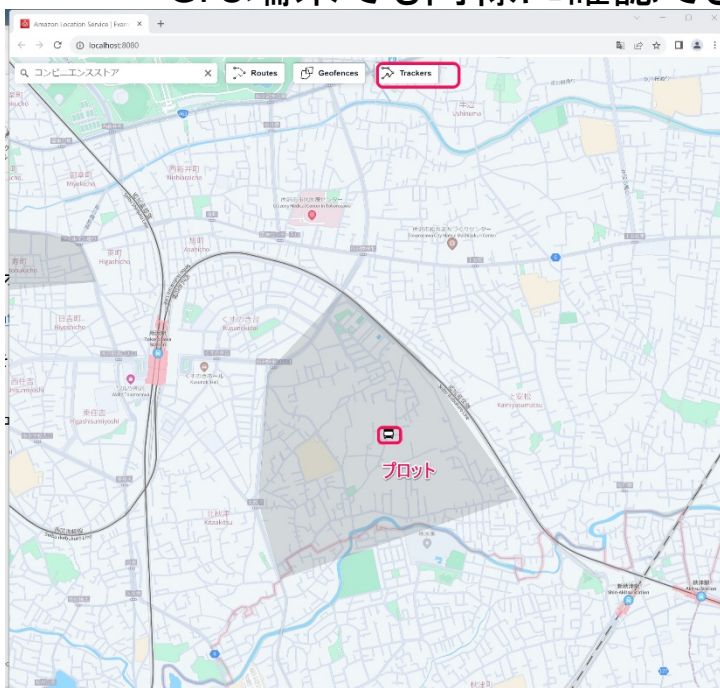
## ⑦地図関係の各種機能(ジオフェンス、ルート、インデックス等)

## ジオフェンス

[https://docs.aws.amazon.com/ja\\_jp/location/latest/developerguide/geofence-an-area.html](https://docs.aws.amazon.com/ja_jp/location/latest/developerguide/geofence-an-area.html)

ジオフェンスを押して、地図上のエリアを選択するとフェンス。そのエリアへの入出力をメールなどで通知可能。

- 確認
  - トラッカシミュレーションでジオフェンス内に車移動させ、メールを確認
  - GPS端末でも同様に確認できます。





# 運行管理開発

## 3. その他

## 運行管理開発

### ⑧外部サイトによるデータ受信確認

#### リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid: mt3339 pi側に設定
- 表示対象をチェックすると自動で表示されます。

#### Countのみ表示



外出先でブラウザで確認できます。

# 運行管理開発

## ⑨運行・勤務管理ツールによる分析

### PowerBIインポート

- Powerbiダウンロード
  - <https://www.microsoft.com/ja-jp/download/details.aspx?id=58494>
  - windowsPCにインストール
- 添付のpowerbiファイルを開く
- データ変換＞データソース設定で各ディレクトリに変更
- ファイル名は、output0925\_2.csvは変えないこと
- 更新でお客様のデータになります。

