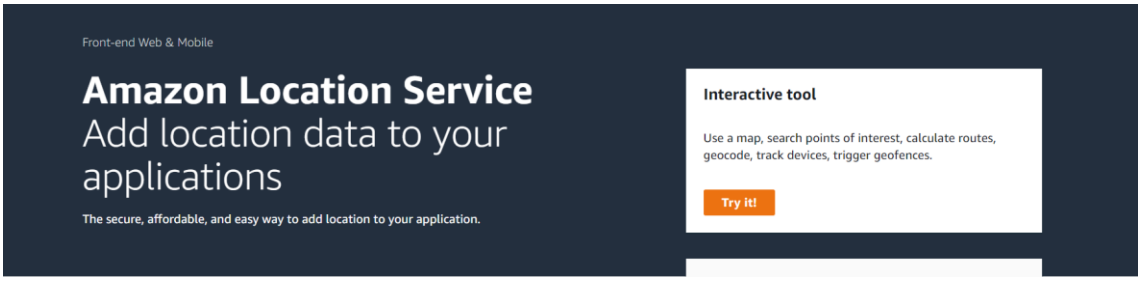
Amazon Location Service ハンズオン

2021/07/14

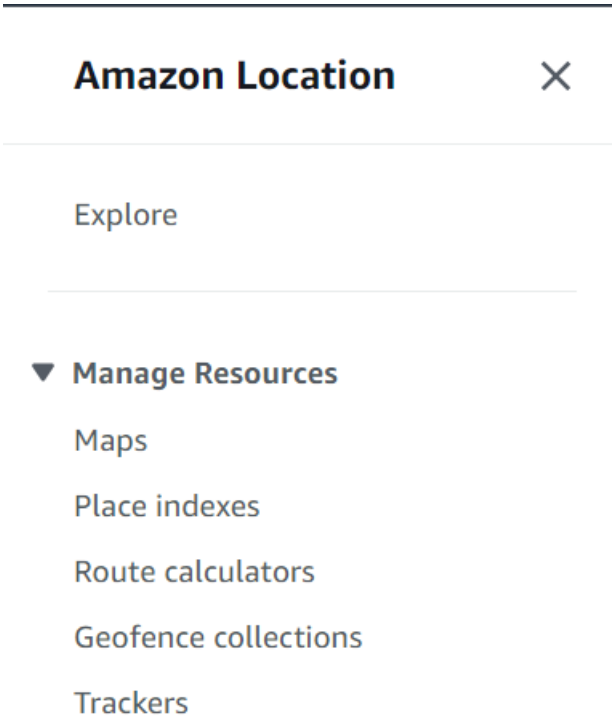
シニアエバンジェリスト　亀田治伸

【地図情報の作成】

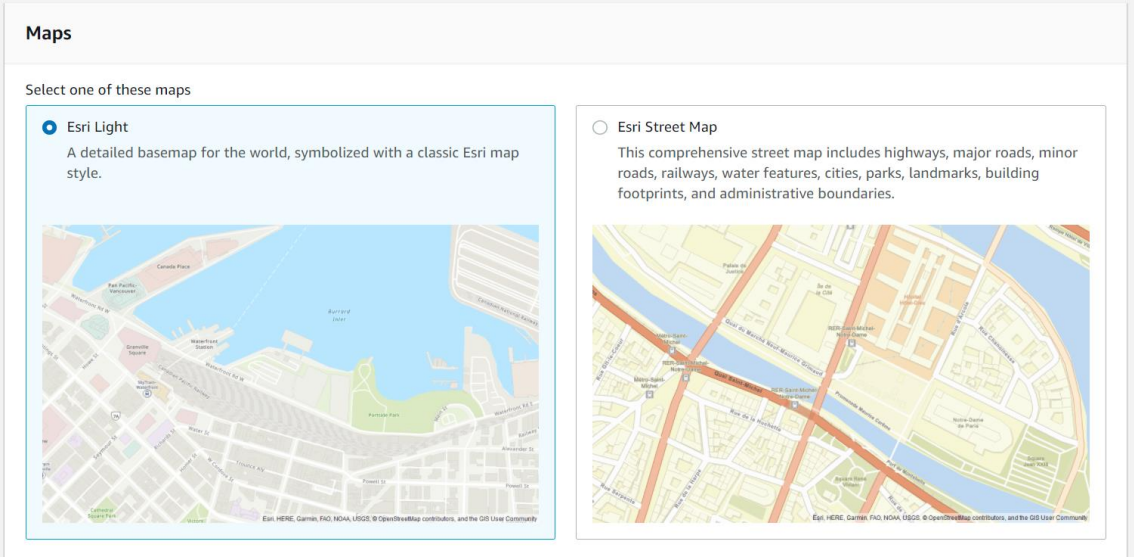
1. マネージメントコンソールで Location Service にアクセスします



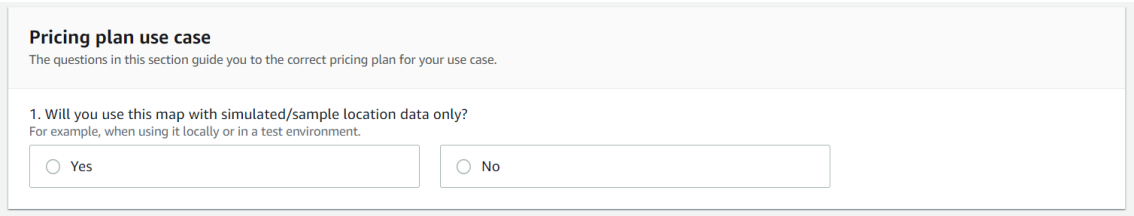
2. 左上、三本線（三）をクリックし、[Maps]をおします



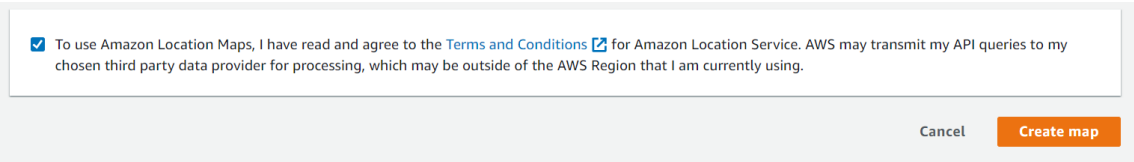
3. [Create map]ボタンをおします
4. 適当な名前をつけます。この名前は後で使うのでメモっておいてください
5. 利用する Map を選びます。種別はなんでも OK です



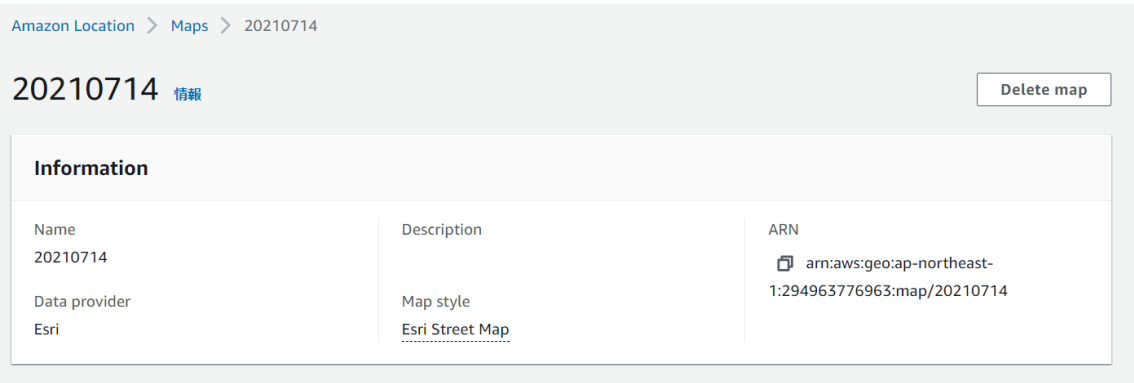
6. [Pricing plan use case]で[Yes]をえらびます



7. 以下にチェックをつけて[Create map]をおします



8. ARN を後で使うのでコピーしておきます



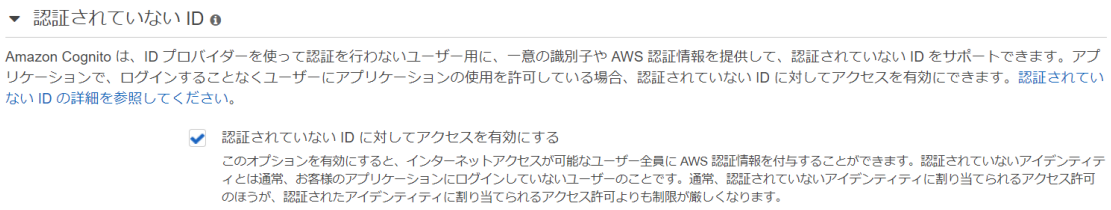
【認証基盤の作成】

以上で地図ができました。地図が埋め込まれた HTML にアクセスするユーザーに地図へアクセスするための認証基盤を作ります。(一時的に IAM ロールを引き渡す仕組

みです)

9. Cognito のマネージメントコンソールにアクセスします
10. [ID プールの管理]をおします
11. [新しい ID プールの作成]をおします
12. 適当な名前を以下につけます

13. [認証されていない ID に対してアクセスを有効にする]にチェックを付けます。これによるログイン不要で IAM ロール権限が　HTML にアクセスした人に一時的に引き渡されるようになります。



14. [プールの作成]をおします
15. [許可]をおします
16. 以下のプールの ID をメモっておきます

17. IAM ロールのマネージメントコンソールにアクセスします
18. 先程作成した cognitoID プールの名前で検索すると 2 つ IAM ロールができています。認証済ユーザー用と非認証一般ユーザー様です

19. このハンズオンでは誰でもアクセスできる HTML 環境を作りますので、[Unauth]の方を選びクリックします
20. [ポリシーをアタッチします]をおします
21. [ポリシーの作成]おします。(ブラウザ別タブが起動します)
22. JSON タブを選び、コピペ用コマンド集の 1 番をコピーします
23. “Resource”の部分を先程作成した map の ARN に置き換えます

24. [次のステップ:タグ]をおし、次の画面で[次のステップ:確認]をおします
25. 適当名前をつけ、[ポリシーの作成]をおします
26. 先程の IAM ロール設定のタブに戻り、先程作成したポリシーを検索から探して、チェックを付けます。出てこない場合、右上の丸い矢印マークをおすと出てきます

27. [ポリシーのアタッチ]をおします。
以上で、非認証一般ユーザーでも地図にアクセスできるようになる権限が作成されま

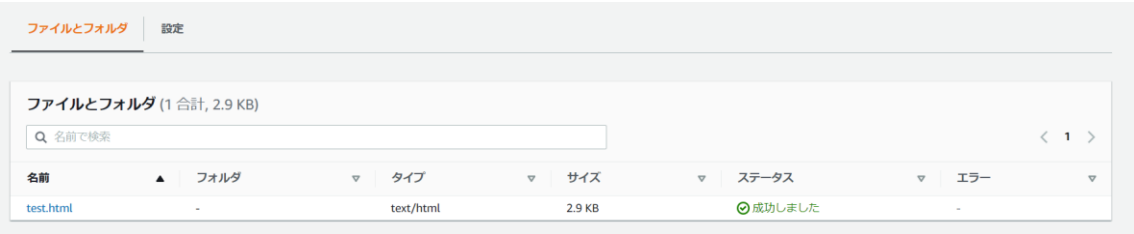
した。

【地図埋め込み HTML の作成とホスティング】

先程作成した地図を HTML に埋め込み、一般ユーザーがアクセスできるように設定し S3 にホスティング。

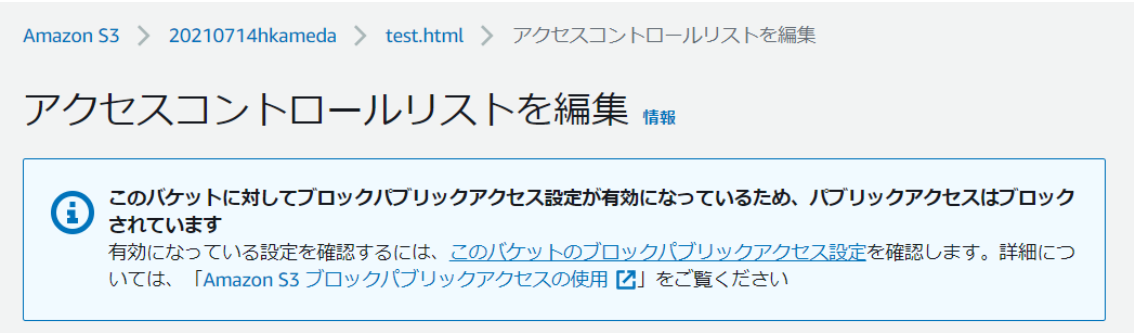
28. S3 のマネージメントコンソールに移動します
29. [バケットを作成]をおします
30. 適当な名前を入力し、[バケットを作成]をおします
31. GitHub から test.html をダウンロードしてエディタで開きます
32. identityPoolId に先程作成した Cognito プール ID の ID をコピペします。同様に const mapName に先程作成した地図の名前をいれます。(ARN ではなく地図の名前です)
33. ファイルを保存し、先程作成した S3 バケットのアップロードします。マネージメントコンソールでバケット名をクリックすると以下の画面が表示されます。[アップロード]ボタンをおしてください

34. [ファイルの追加]をおし、ファイルが選択出来たら[アップロード]ボタンをおします
35. アップされたファイルをクリックします

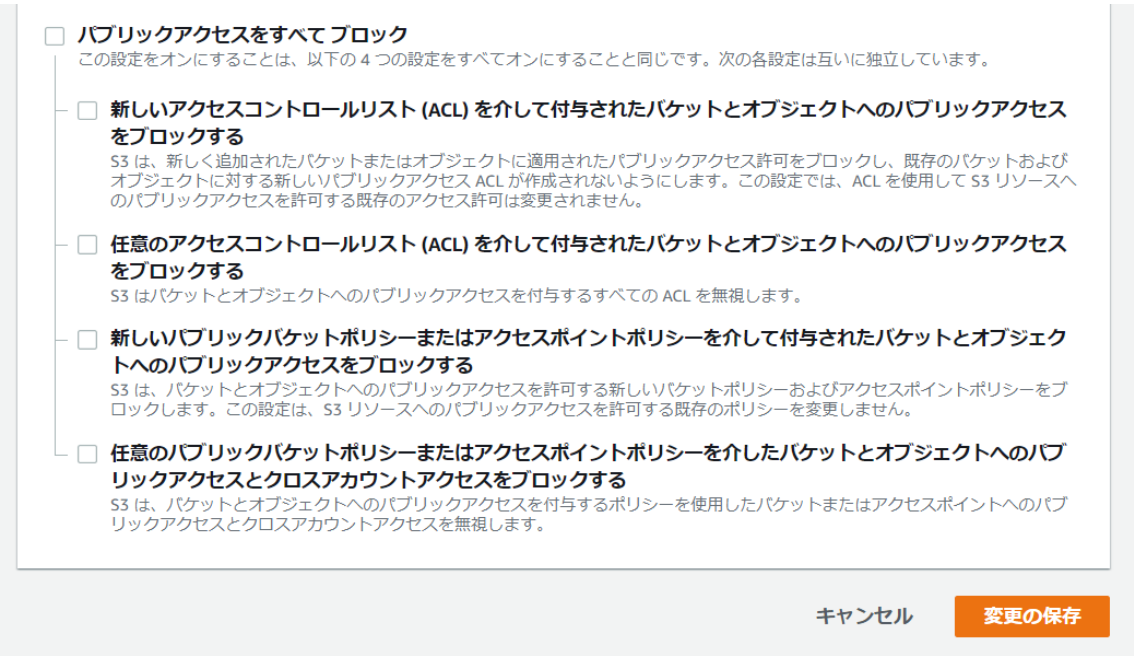


36. [アクセス許可]のタブをおします

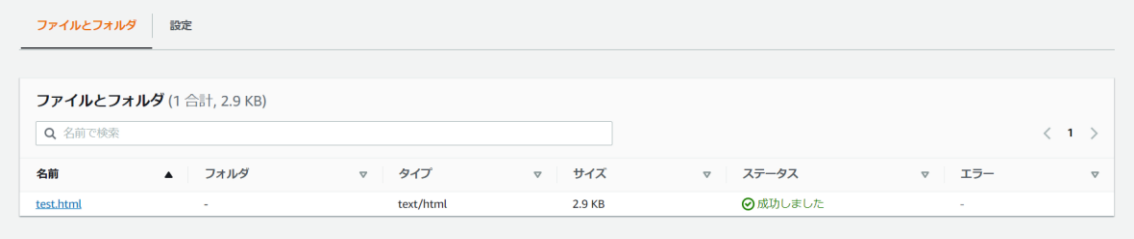
37. [編集]をおします
38. 以下の画面の[このバケットのブロックパブリックアクセス設定]をクリックします



39. [ブロックパブリックアクセス（バケット設定)]の[編集]ボタンをおします
40. 以下の様にチェックを外して[変更の保存]をおします



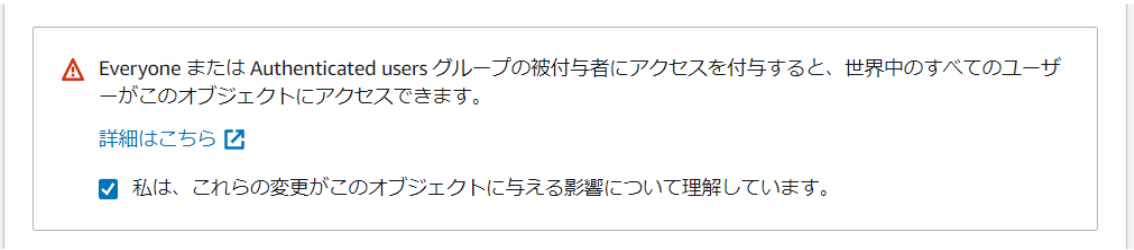
41. 次の画面で[確認]と入力し[確認]ボタンをおします
42. 再度バケットの中にある test.html をクリックします



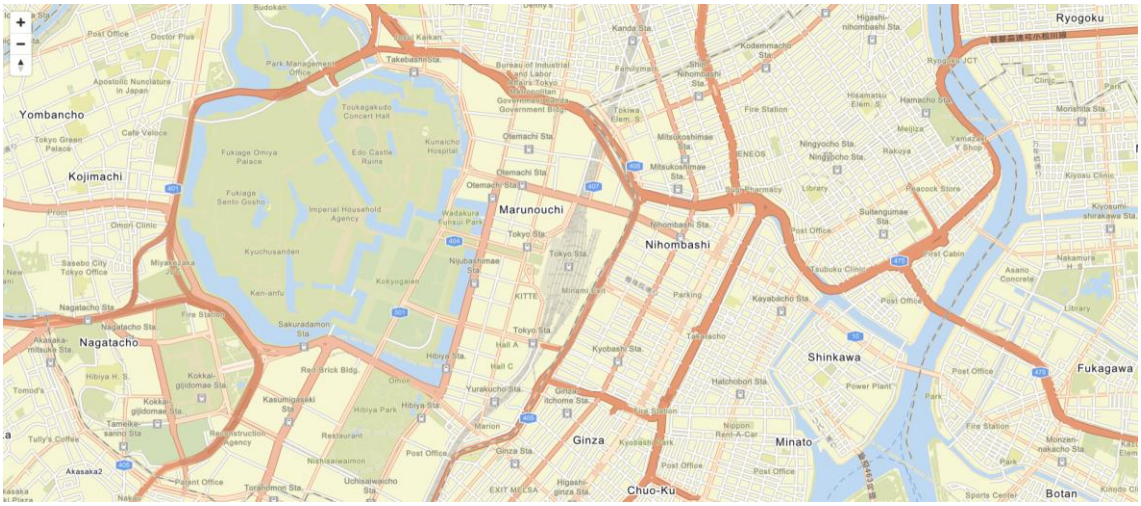
43. [アクセス許可]をタブを選びます

44. [編集]ボタンをおします
45. 全員 (パブリックアクセス)の設定を以下のようにします。

46. 注意事項にチェックをつけ、[変更の保存]をおします



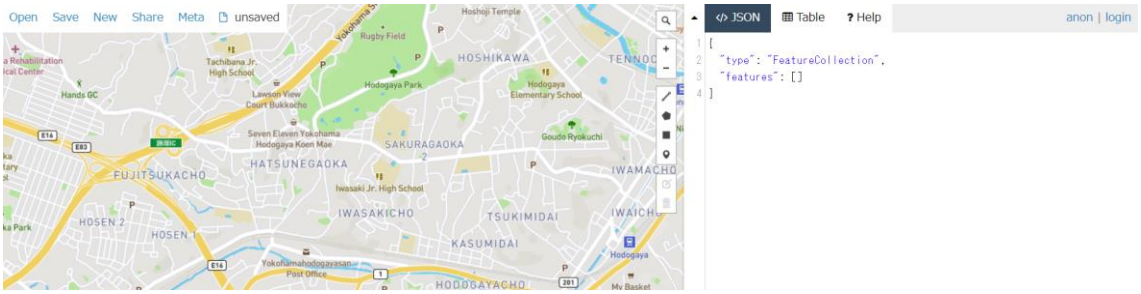
47. 以下のように地図が表示されます



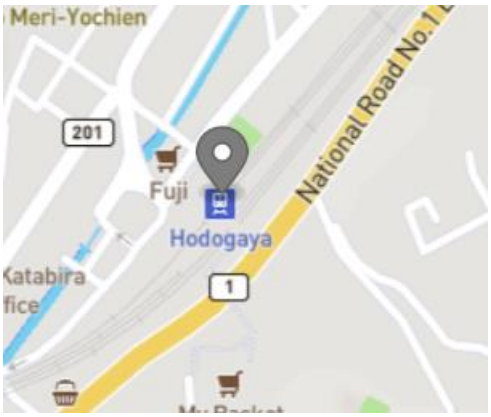
真っ白な画面が出る場合 test.html の修正をミスしていると思われるので、再度確認してみてください。

【Geo Fence 機能の実装】
Location Service には GPS をベースとして任意の区画にデバイスが[Exit]した、もしくは[Enter]したというイベントを取得することができます。
AWS IoT Core 経由でデバイスから GPS 情報が MQTT で挿入され、それを Lambda で Location Service に送信し、イベントを CloudWatch Logs に出力する環境を作ります。

48. https://geojson.io/　にアクセスします



49. 自分の最寄駅をさがします
50. 駅が見つかったら以下のボタンをおしてピンを落とします

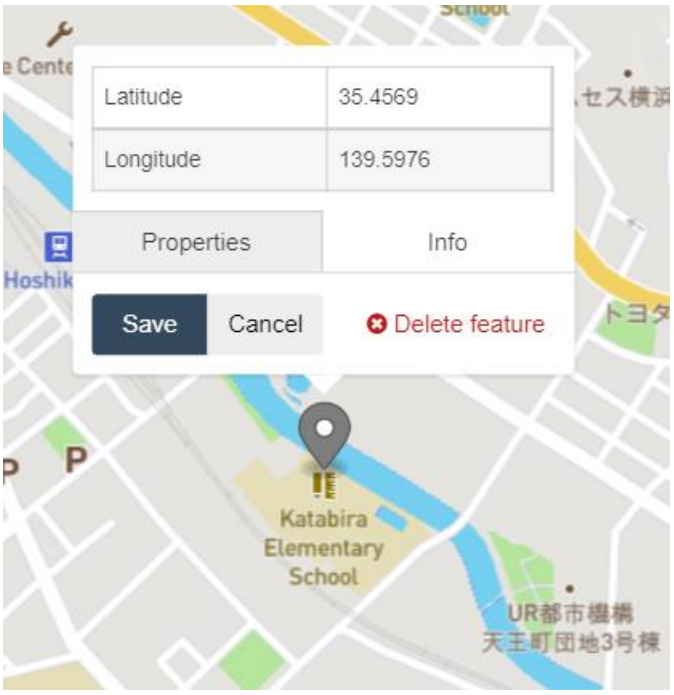


51. 画面右に GPS 情報が出てきますので、coordinates で表示される座標をメモリます

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "type": "Point",
        "coordinates": [
          139.59964513778687,
          35.44680904209397
        ]
      }
    }
  ]
}
```

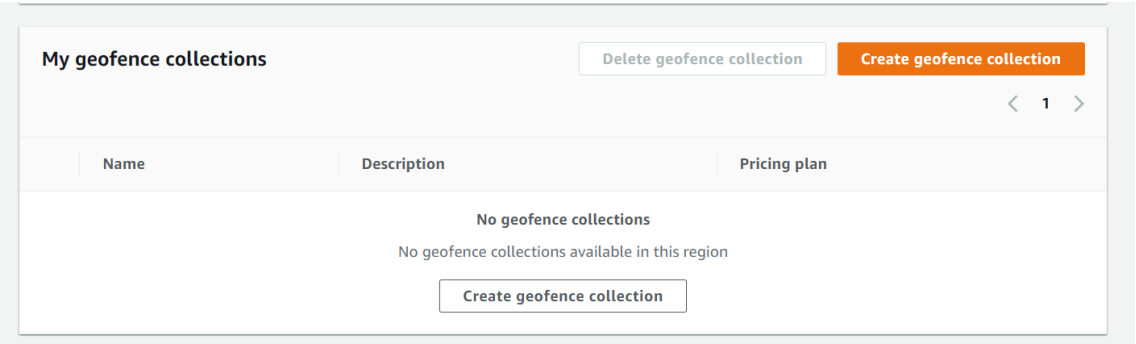
52. 地図上のピンをおして[Delete feature]をクリックし、ピンをけします
53. 今度は 5 角形のマークをおして、駅周辺に領域を作成します

54. 画面右側に表示される JSON をコピーし、[geo.json]として保存してください
55. 次に、領域の外にある場所のどこかでピンを再度落とし、GPS 情報をメモって下さい

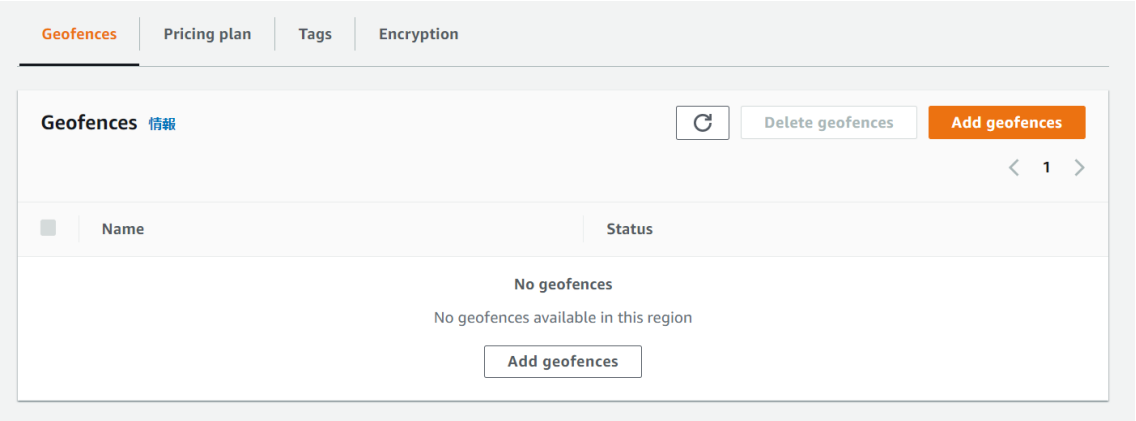


56. Location Service のマネージメントコンソールに戻り左のペインから[Geofence

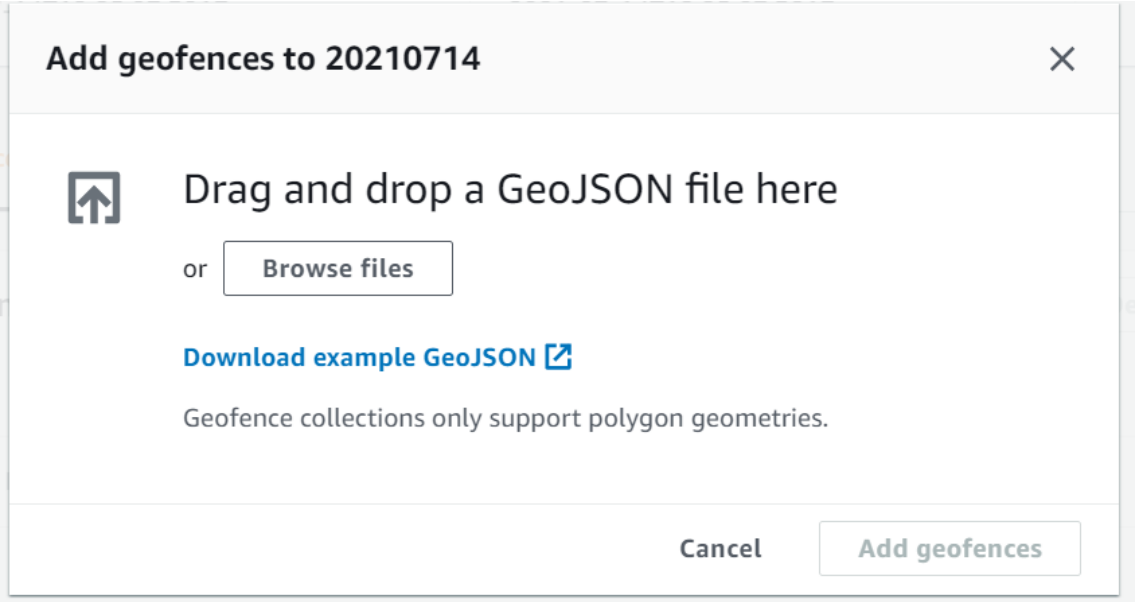
collections]をクリックします



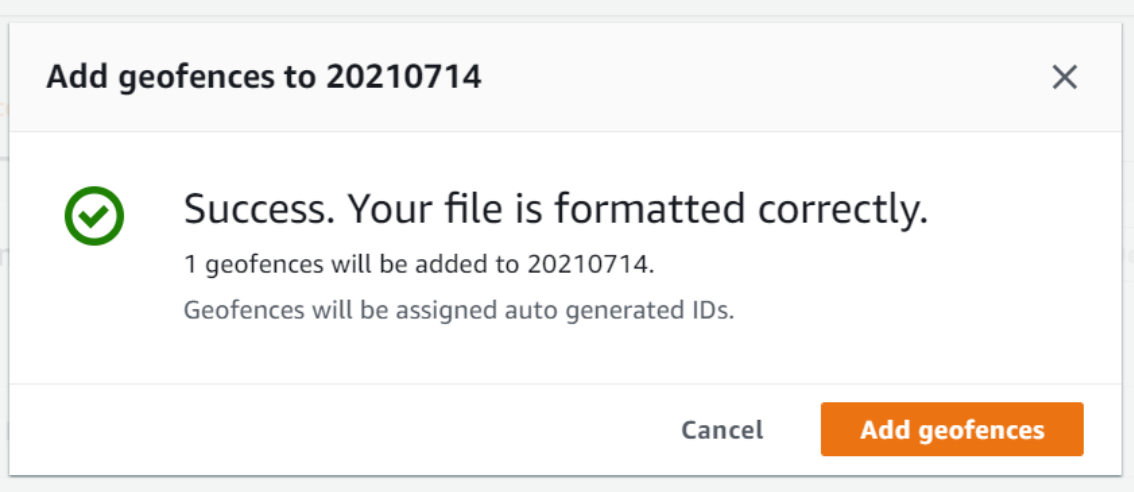
57. [Create geofence collection]をおします
58. 適当な名前をつけ、[Pricing plan use case]で Yes を選び、[Create geofence collection]をおします
59. [Add geofence]をおします



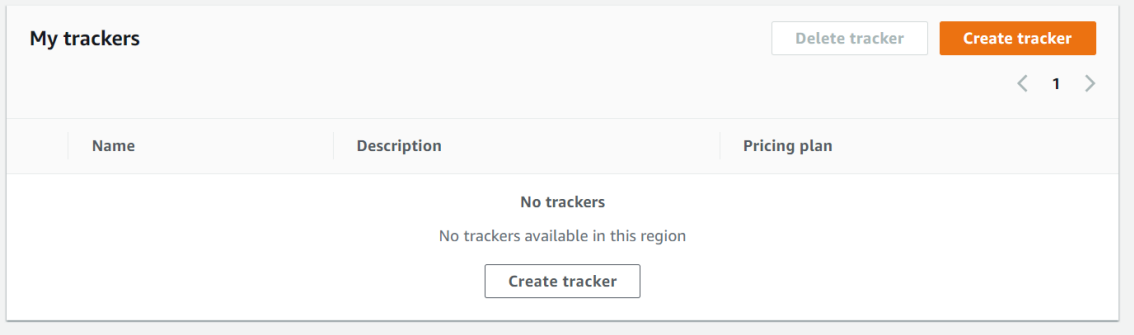
60. 先程作成した geo.json をアップロードします



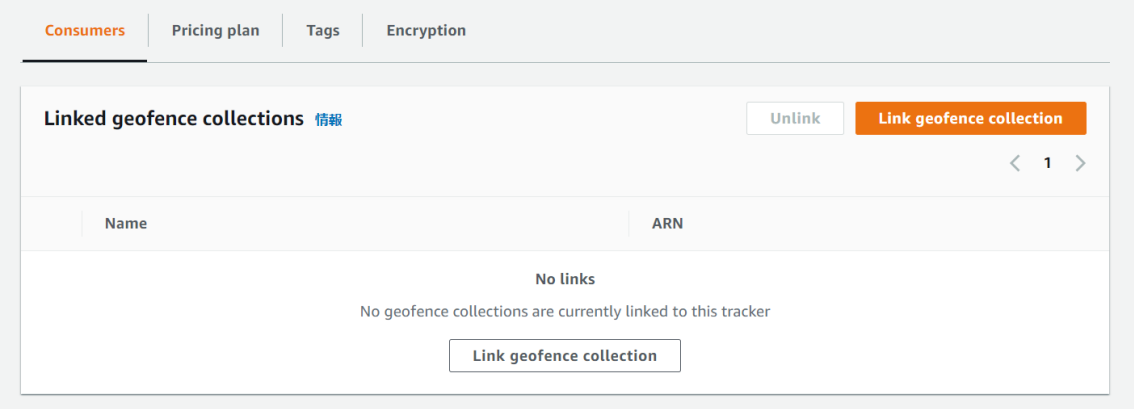
61. ファイルが正しければ以下が表示されますので[Add geofences]をおします



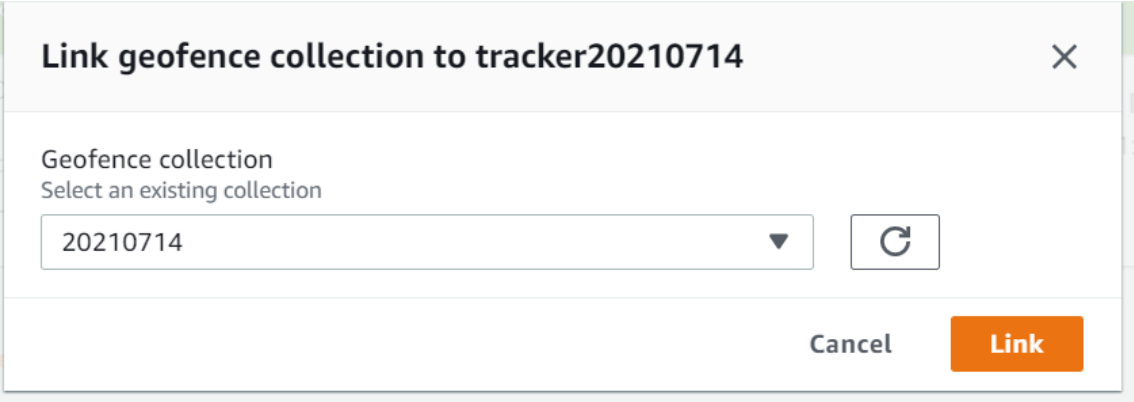
62. これで最寄駅を周辺とした領域を Location Service が認識しました。次に GPS デバイスと連携させるために、左ペインから[Trackers]をクリックします



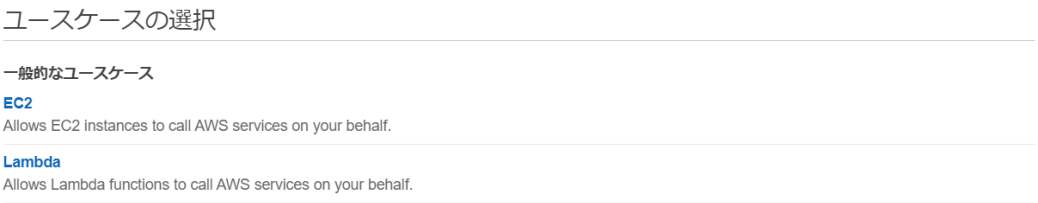
63. [Create trackers]をおします
64. 適当な名前をつけ、[Pricing plan use case]で Yes を選び、[Create tracker]をおします
65. [Link geofence collection]をおします



66. 先程作成した geofence を選んで[Link]をおします



67. これで特定の Tracker が特定の Geo fence と連携するようになりました。Tracker には複数の Geo fence を連携させることも可能です。これからの手順では Tracker に対して AWS IoT Core から MQTT ベースで GPS 情報が送られるように設定をおこないます。
68. まず Lambda 関数にアタッチする IAM ロールを作成するために、IAM のマネージメントコンソールにアクセスします
69. 左のペインから[ロール]をクリックします
70. [ロールを作成]をおします
71. 以下で Lambda を選び、[次のステップ:アクセス権限]をおします

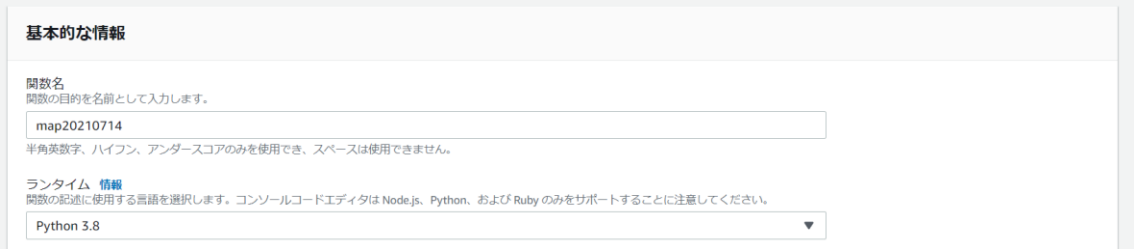


72. [Administrator Access]を選び、[次のステップ:タグ]をおします

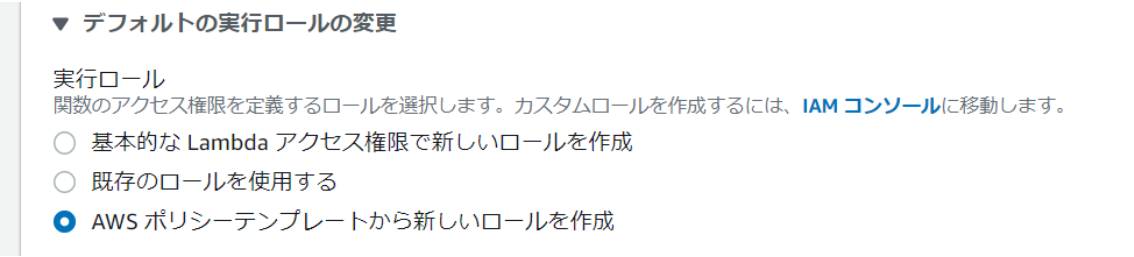
注意：本来は Location Service のみのアクセス権でいいのですが、2021 年 7 月時点でビルトインの Location Service 用ポリシーが要されておらず、そちらの作成は手順が煩雑となるため、このハンズオンでは割愛します。商用環境における Admin 権限はお勧めしませんので、詳しくは以下を参考にしてください。

https://docs.aws.amazon.com/ja_jp/location/latest/developerguide/security-iam.html

73. 次の画面ではそのまま[次のステップ:確認]をおします
74. 適当な名前をロールにつけ[ロールの作成]をおします
75. 次に Lambda 関数を作成します。この関数は AWS IoT Core から上がってきた GPS 情報を Location Service に渡す役割をにないます。Lambda のマネージメントコンソールにアクセスします
76. [関数の作成]をおします
77. 適当な名前を関数につけ、ランタイムには Python3.8 を選びます

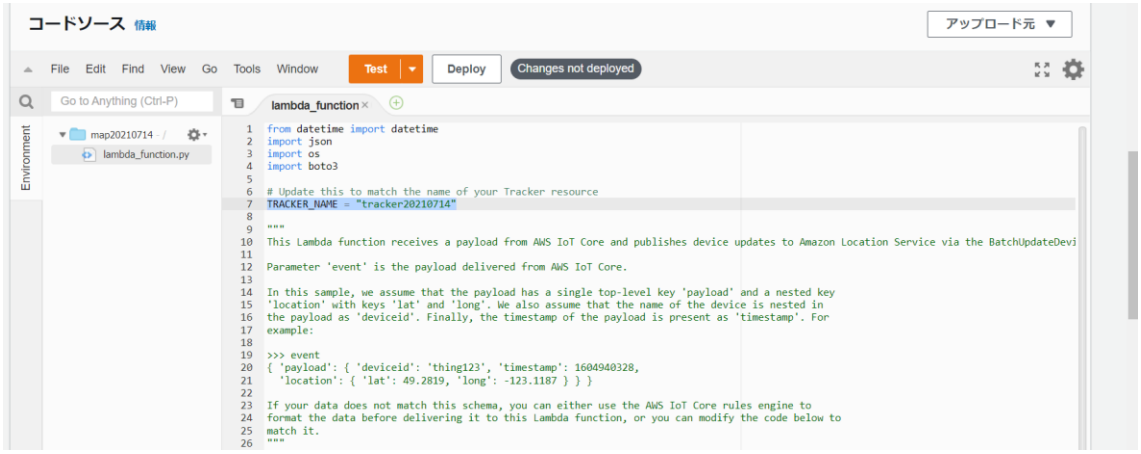


78. [デフォルトの実行ロールの変更]をクリックし、[既存のロールを使用する]を選びます



79. 先程作成した IAM ロールを選びます、[関数の作成]をおします

80. GitHub から lambda.txt をダウンロードし、その中身をコードに貼り付けます
81. TRACKER_NAME の部分を先程作成した Tracker の名前に置き換えます

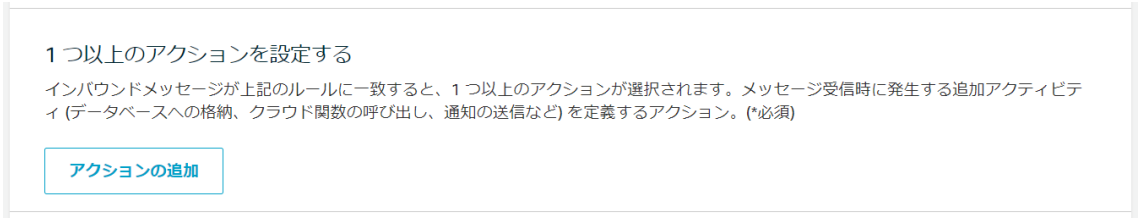


82. [Deploy]をおします
83. 次に IoT Core のマネージメントコンソールにアクセスします
84. 左ペインから[ACT]→[ルール]をクリックします

85. [作成]をおします
86. 適当な名前をつけ、[ルールクエリステートメント]を以下で置換します

```
SELECT * FROM 'iot/location'
```

87. [1 つ以上のアクションを設定する]の[アクションの追加]をおします



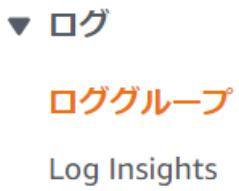
88. [Lambda]を選び、[アクションの設定]をおします

89. 先程作成した関数を選び[アクションの追加]をおします
90. [ルールの作成]をおします

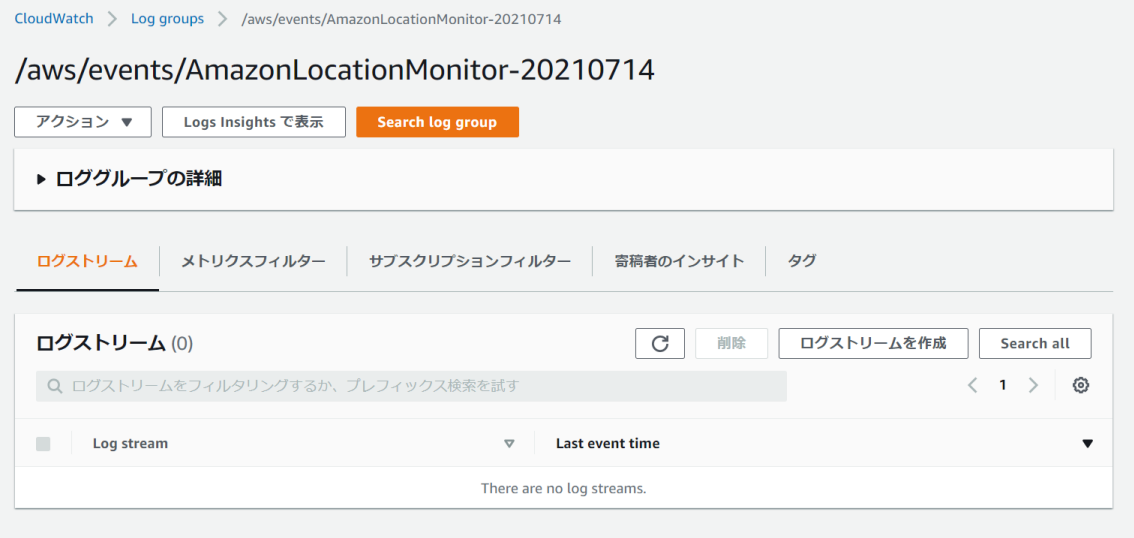
ここまでの手順で、IoT Core から MQTT ベースて上がってきた GPS 情報が Lambda 経由で Location Service の Tracker に渡され、Tracker と Link されている Geo fence 情報と

突合されその結果が CloudWatch Logs に出力されるようになりました。
先程作成した Lambda の関数画面に行くと正しく IoT Core がイベントソースとして設定されていることがわかります

91. テストを行う前に CloudWatch Logs を開きます。CloudWatch マネージメントコンソールの左ペインから[ロググループ]をクリックします



92. [location]で検索すると今日作成したロググループが出てきますので、クリックしておきます



今の時点ではログは何もふくんでいません。この画面は開きっぱなしにしておいてください。

93. いよいよテストです。IoT Core 左ペインから[MQTT テストクライアント]をクリックし、[トピックに公開する]タブを選んでください

94. トピック名は[**iot/location**]と入力します
95. コピペ用コマンド 2 番の内容を別のエディタにコピーし、先程取得した最寄り駅の GPS 情報に置き換えます。置き換えたものを[メッセージペイロード]]に貼り付け、[発行]をおします
96. Location Service はデバイスが Geo fence の中にいることを認識しました。次に timestamp の値を 10 増やし、geo fence の外にある GPS 情報に置き換え、再度メッセージを発行してください
97. ClouudWatch Logs を確認すると 2 つのログがでています
    Enter と EXIT です

おつかれさまでした！ハンズオンは以上です

以下を削除してください

Location Service

- Trackers
- Geofence collection
- Maps

Lambda 関数

S3 バケット

IAM ポリシーとロール

AWS IoT Core

- ルール