

APIデータを取得してLineに送ってみる。

できたらAWSLambdaも使ってみるよ

1. 自己紹介

- 下地泰寛@31歳
- 現職：クラスメソッドのエンジニア(2020年1月~現在)
- 前職：工場の作業員（自称SE）
- 趣味：アマプラでアニメや映画を見ること

アニメ：暗殺教室、シャーマンキング、進撃の巨人、テンスラ、キングダム

ドラマ,映画：テセウスの船、バクマン、ミックス、ちょっと今から仕事辞めてくる

- 漫画：ベイビーステップを再度読みました
- pybegiとの関わり

2017年ごろからイベントに参加させてもらっています

これまで3回ほど登壇させてもらっています(転職するために追い込んで頑張ってました)

1. PyBegi18_NLP
2. PyBegi19_Docker
3. PyBegi24_TensorFlow

2. クラスメソッド(CM)について

どんな会社？：IT会社です！

- 社員全員でブログ書いてます [DevelopersIO](#)
- [AWS総合支援\(クラスメソッドメンバーズ\)](#)
- [LINEサービス総合支援](#)
- [データ分析環境構築支援](#)(←私の所属部門です！)

[データ統合基盤「CSアナリティクス」](#)の保守担当しております。

働き方

- 完全リモートワーク（入社して1年半以上経ちましたが1ヶ月も出社してません）
- コアタイムなしのフレックス(でも会議などあるので基本的に日中出勤しています)
- 那覇にオフィスもあります！(現在、那覇オフィス所属は7名おります)

- 私のポジション：データ分析基盤構築エンジニア(主にSQL書いてます。AWSとPythonは少々)とても働きやすい反面、成果重視なので厳しい面もあります(まー仕事ですもんね)程よいプレッシャーを感じAWSの認定資格も7つ取得できました！

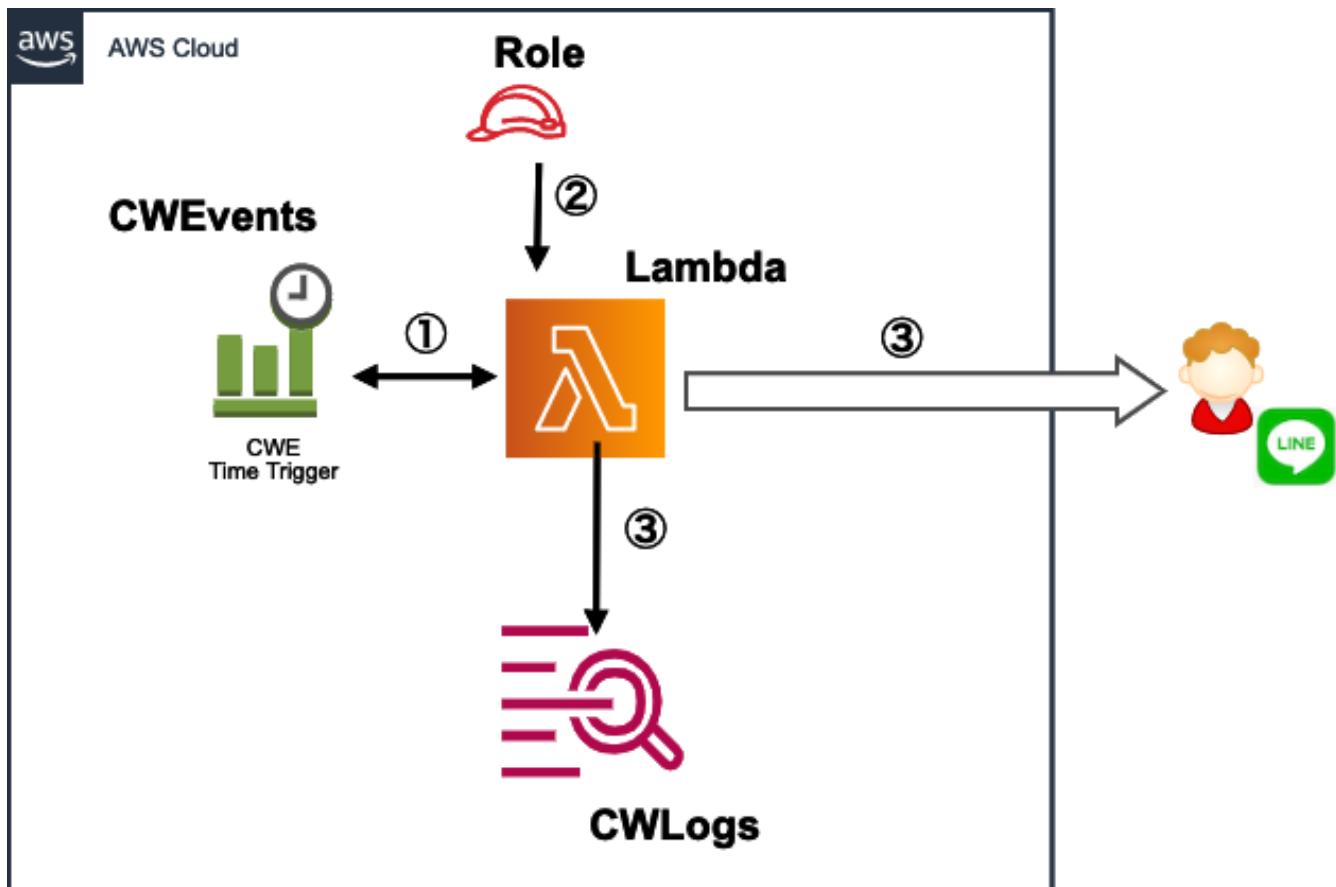
リンクと採用案内

- [classmehotd](#)
- [classmehotd 採用サイト](#)
- [annotation](#)

3. 今回の実装内容

Lambdaで実装する全体像を以下に示します。

定時刻にOpenWeatherAPIから天気情報を取得してLineに通知します



処理の流れ

1. CWEEventsで定時刻にLambdaを起動
2. Roleにより必要な権限を付与
3. Lambdaの処理を実施。実施のログを随時CWLogsに送信

- Lambda: サーバレスアーキテクチャ(ソースコードの実行環境一式提供されるため、ソースコードがあればすぐに実行できる)
- Role: 実行時の権限を付与する機能
- CloudWatchLogs: プログラム実行時のログを記録する
- CloudWatchEvents: 定時刻にLambdaなどを実行する(Cronなどを使用)
- CloudFormation(IaC)を使用してデプロイする(コードでリソースを管理する便利なやつです)

今回使用するファイルを記載します。

- python処理のソースコード: 29_pybegin_python_local.ipynb
- Lambdaを作成するcfnコード: make_lambda_notify_line.yaml

3.1 作業フロー

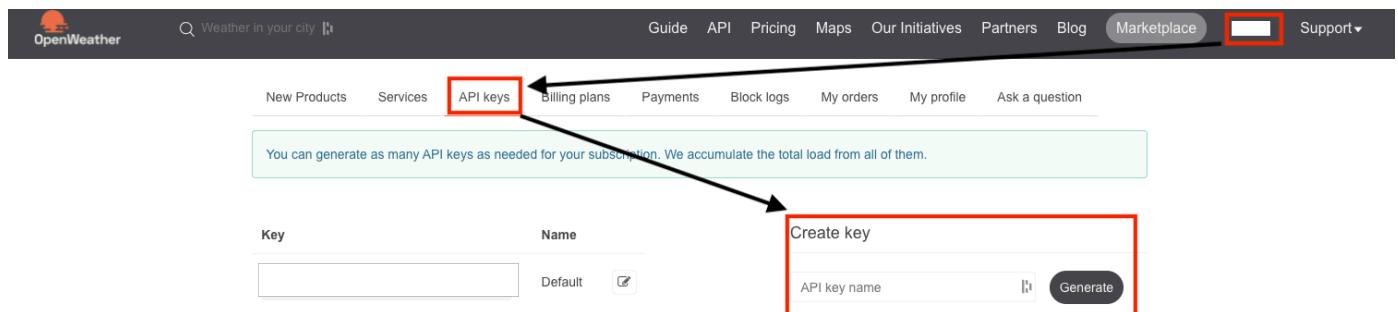
1. OpenWeatherAPI取得
2. OpenWeatherAPIでデータを取得・加工
3. LineNotifyのトークンを取得
4. 2の結果をLineに通知(もしくはSlackに通知)
5. cfнを使用してAWSLambdaを実装してみる

3.2 OpenWeatherAPI取得

参考サイトと図を記載します。

- 公式サイト: [OpenWeather](#)
- 参考URL: [世界の天気API「OpenWeatherMap」の無料APIキー発行・取得、リクエスト方法](#)

アカウント作成を行ったら、図のような流れでAPIを取得します。



Keyが取得できたらOKです。

3.3 LineNotifyのトークン取得

LINE Notifyを使用します。

ラインを使用されている方は、メールアドレスを登録してここから使用できます。

- 公式サイト: [LINE Notify](#)
 - 登録方法: [LINE Notifyをnode.jsで利用する](#)
-

3.4 Slackに通知する

使い方のコマンドは、

- [AWS Lambdaを使ってSlackにメッセージを送る](#)

```
pip install slackweb
```

3.5 cfnを実装してみる

`make_lambda_notify_line.yaml`を開いてください