[AWS Black Belt Online Seminar]
AWS IoT Events

サービスカットシリーズ

Prototyping Solutions Architect 市川 純
2020/9/16

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive

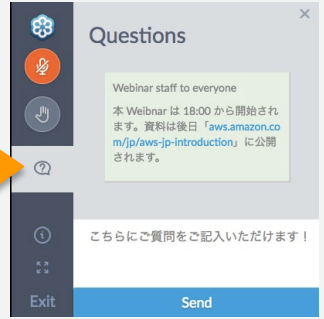
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
  お答えできませんのでご了承下さい

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2020年9月16日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(http://aws.amazon.com)にてご確認ください。

- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます。

- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at http://aws.amazon.com/agreement/. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

自己紹介

名前：

市川 純（いちかわ じゅん）

所属：

アマゾン ウェブ サービス ジャパン株式会社

デジタルトランスフォーメーション本部

プロトタイピング ソリューション アーキテクト

担当：

IoTに関係するプロトタイピングやお客様の支援

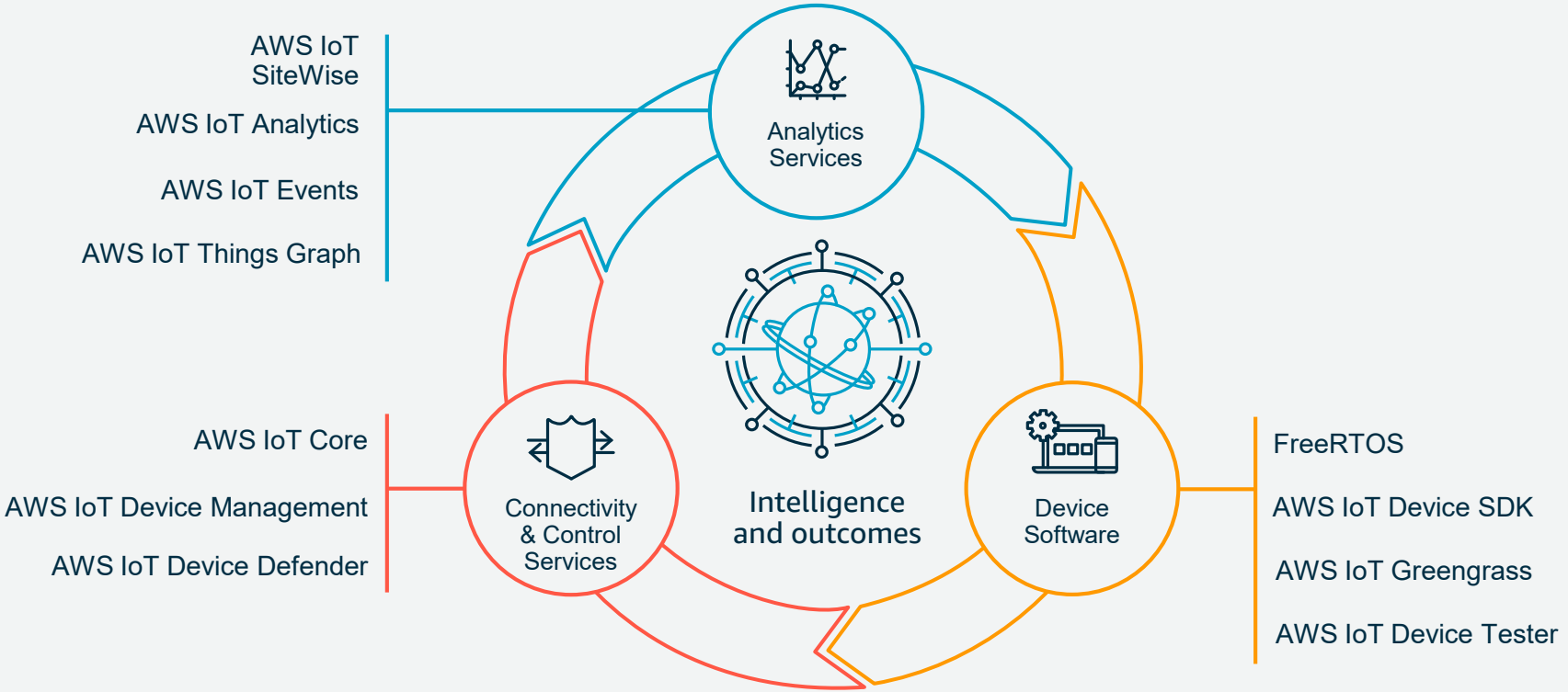
aws

本日のアジェンダ

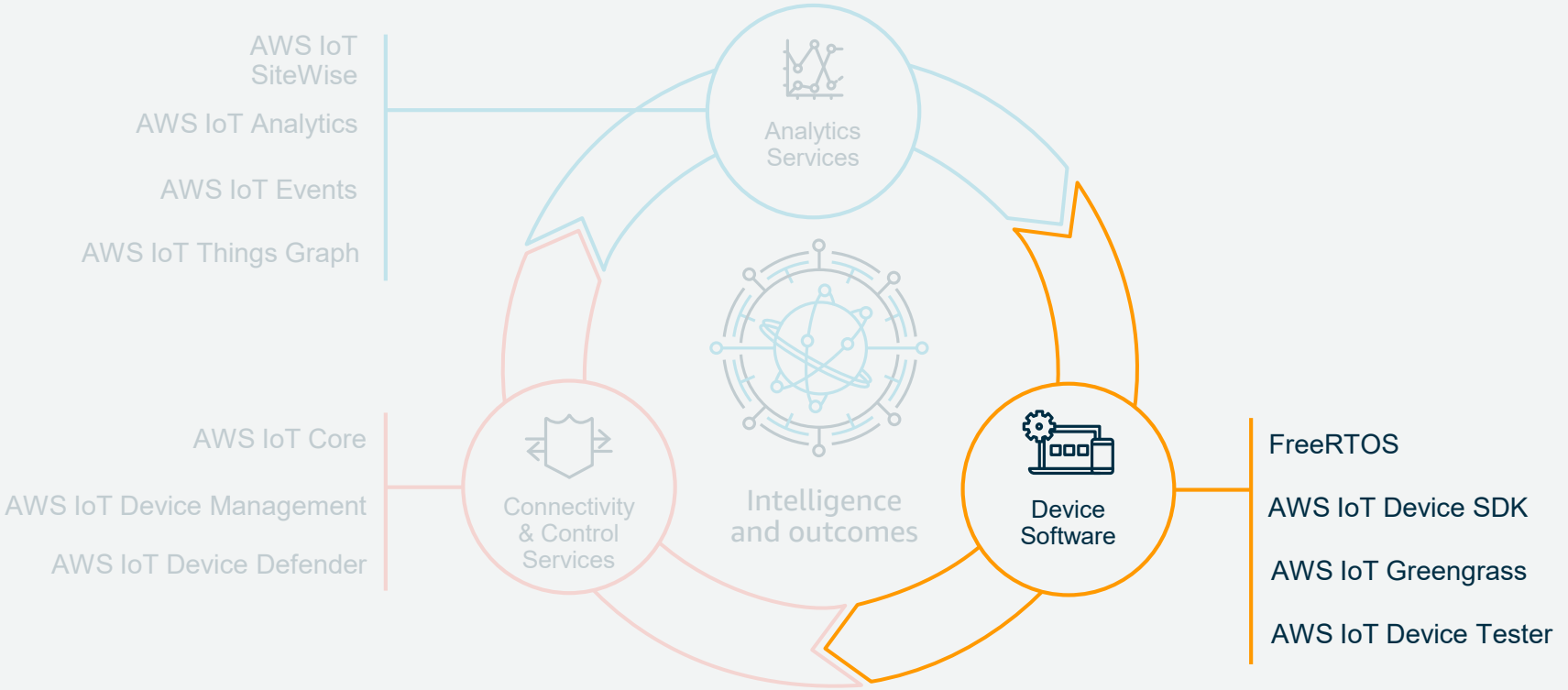
- AWS IoTサービスの紹介
- AWS IoT Eventsについて
- AWS IoT Eventsを使ったユースケースの紹介
- AWS IoT Eventsの始め方
- まとめ
- Q&A

AWS IoT サービスの紹介

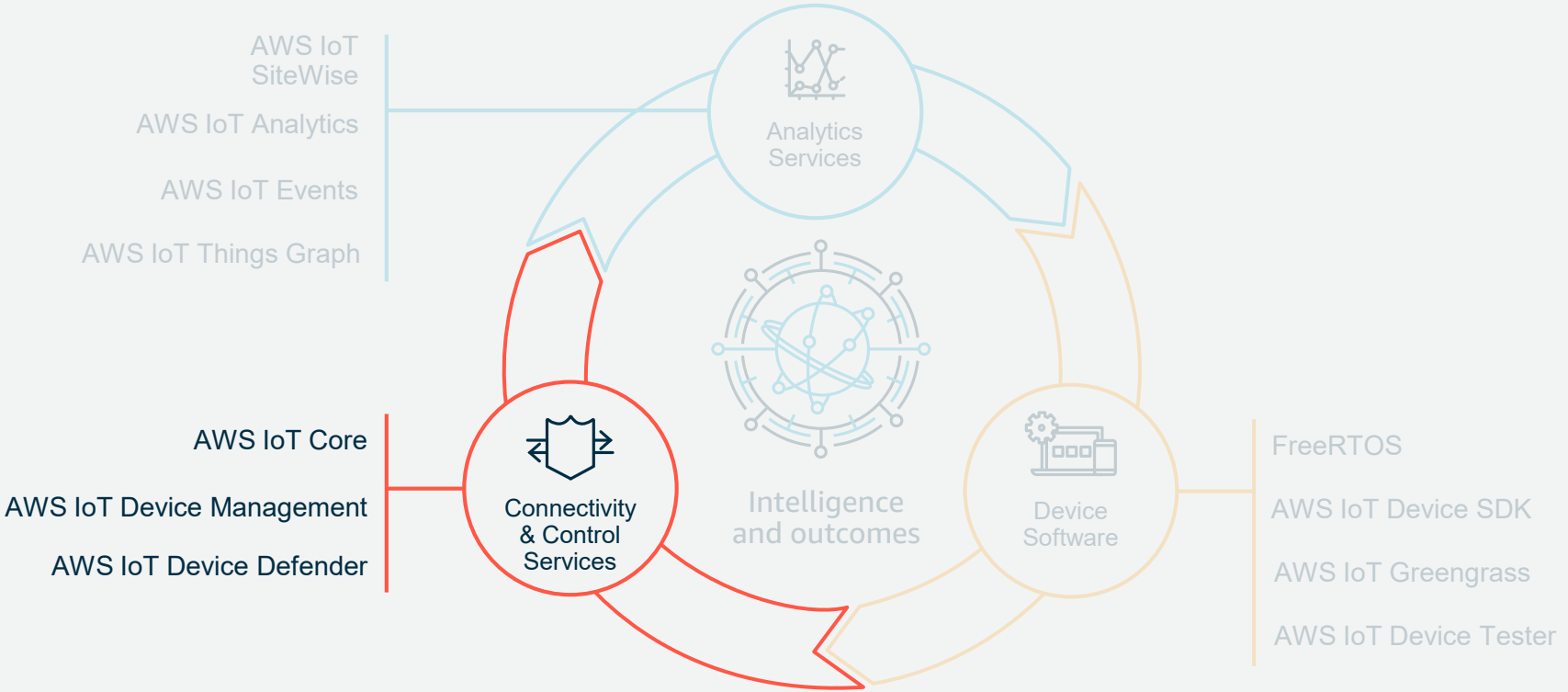
AWS IoTのサービス



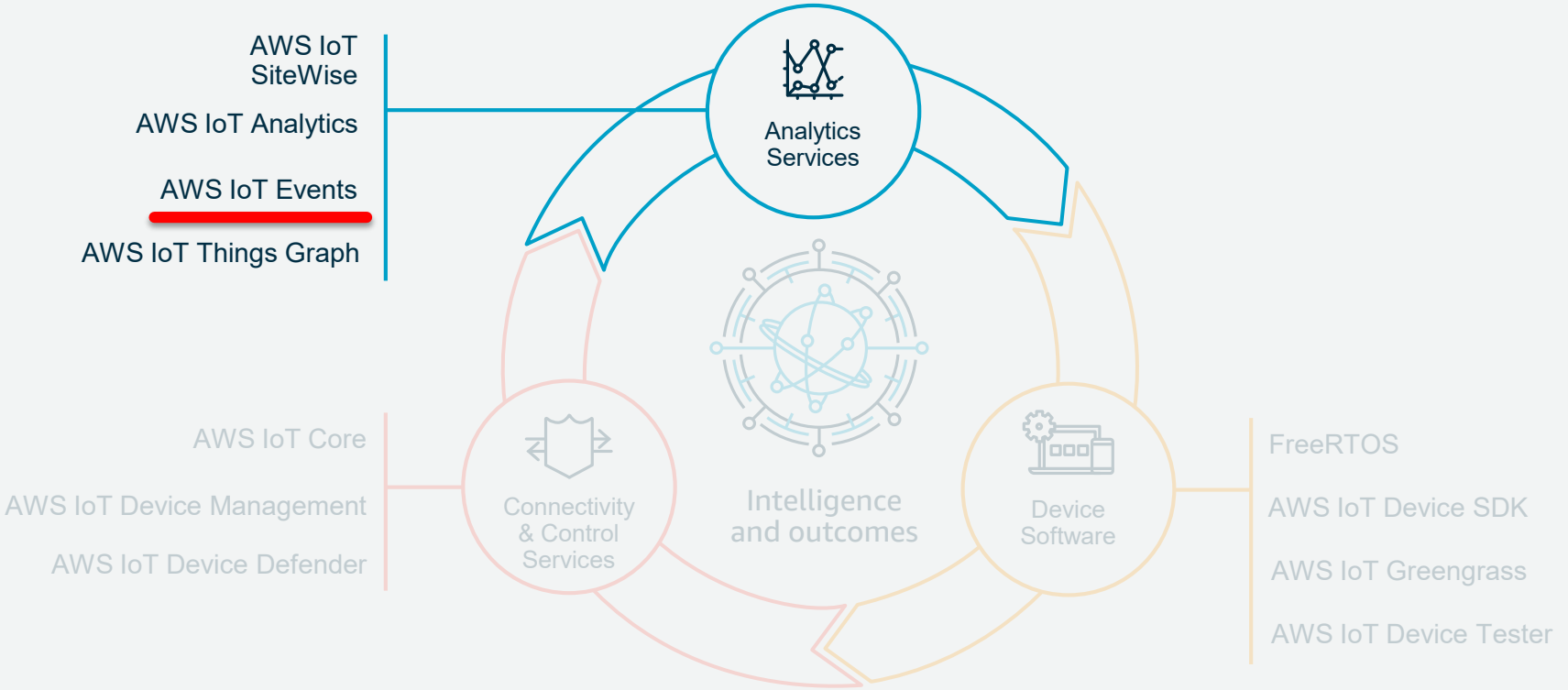
AWS IoTのサービス



AWS IoTのサービス



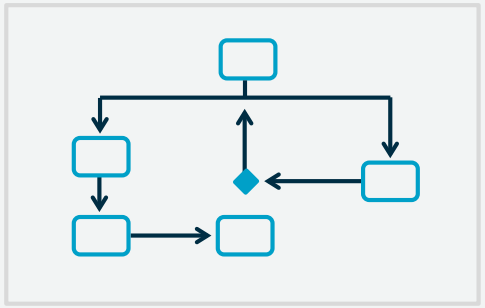
AWS IoTのサービス



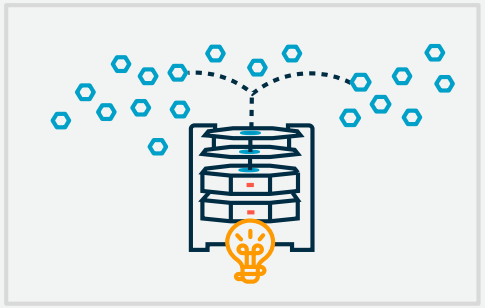
AWS IoT Eventsについて

AWS IoT Events

AWS IoT Eventsを使用すると、機器および一連のデバイスからのデータを継続的に監視し、イベントが発生したときに適切な対応をトリガーできます



複数の条件を評価しながら
デバイスやシステムの状態を
判断するロジックを構築

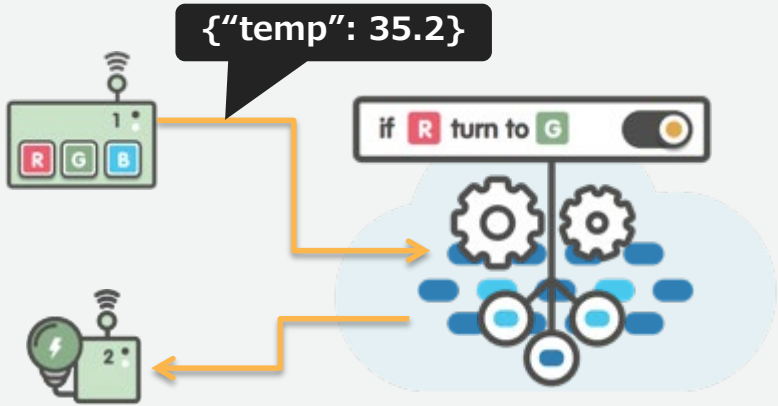


何千ものセンサーや他のソース
のデータからイベントを検出

イベントの結果
最適なアクションを実行

IoT Coreのルールエンジンを利用した場合

- SQL文を使ったトピックのフィルタ
- オプションの**WHERE句で条件を記述**することが可能
- JSONをサポート

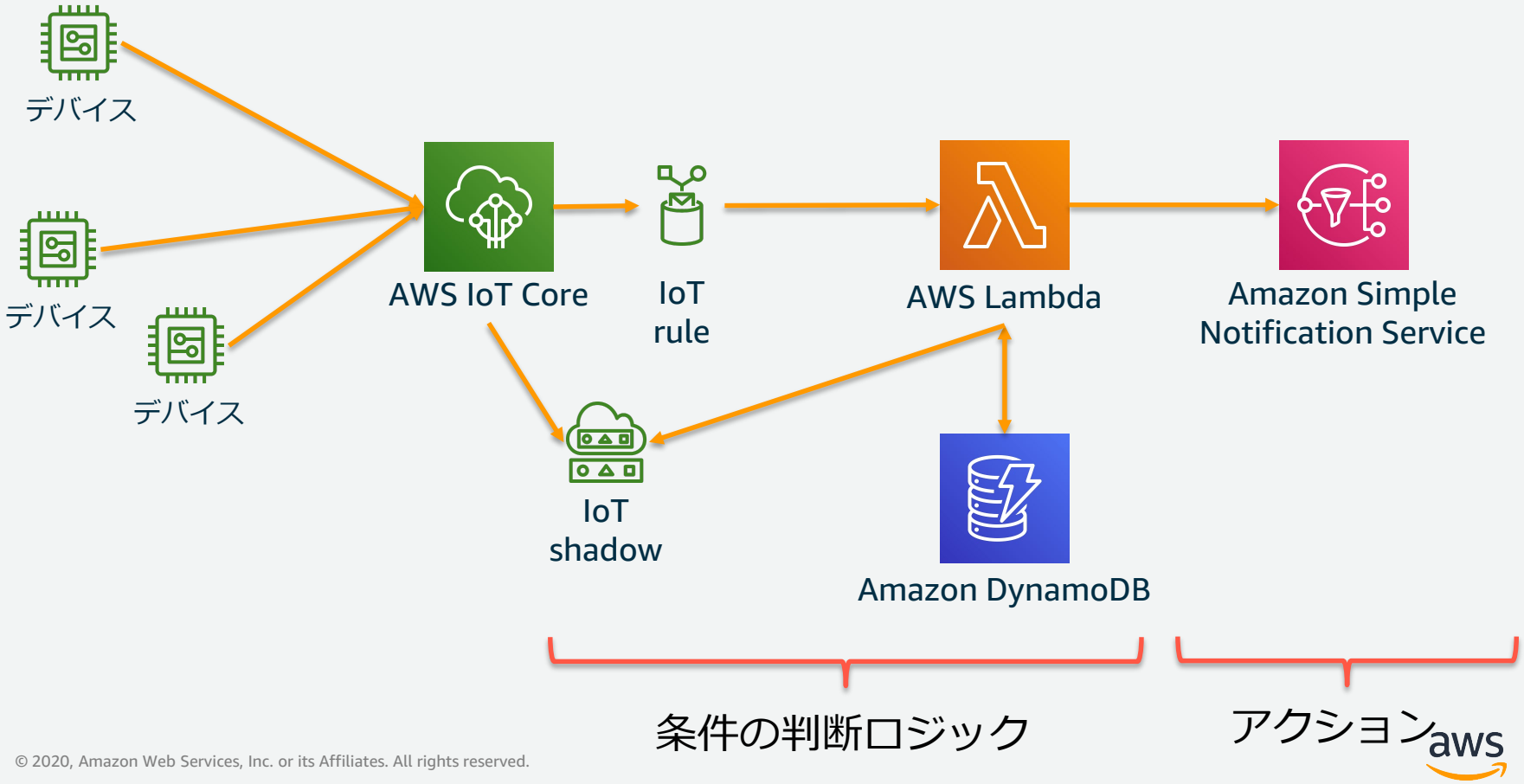


- トピックのフィルタの例

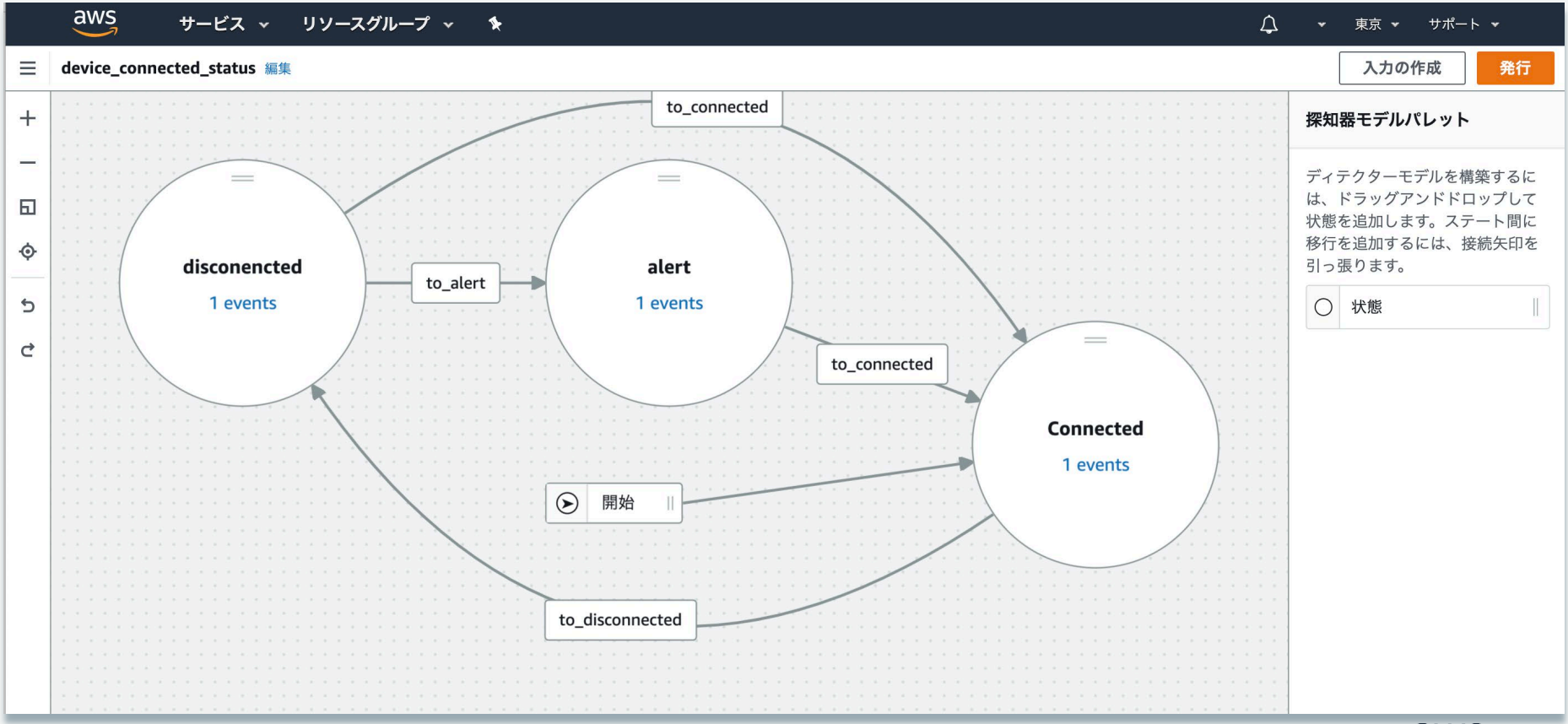
```
SELECT *
FROM ‘data/meguro_building/1f/+/sensor’
WHERE temp > 35.0
```


aws

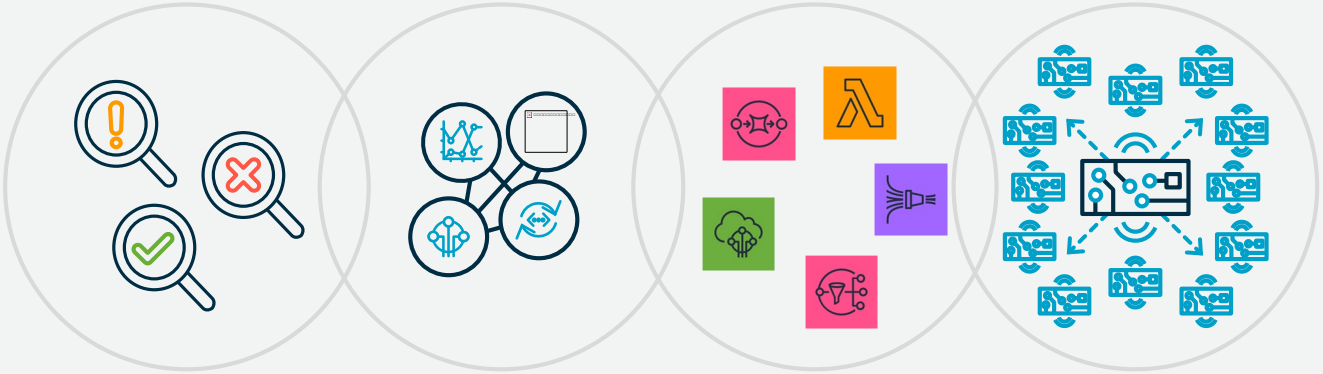
IoT Coreのルールエンジンを利用した場合



AWS IoT Eventsの編集画面



AWS IoT Eventsの特徴

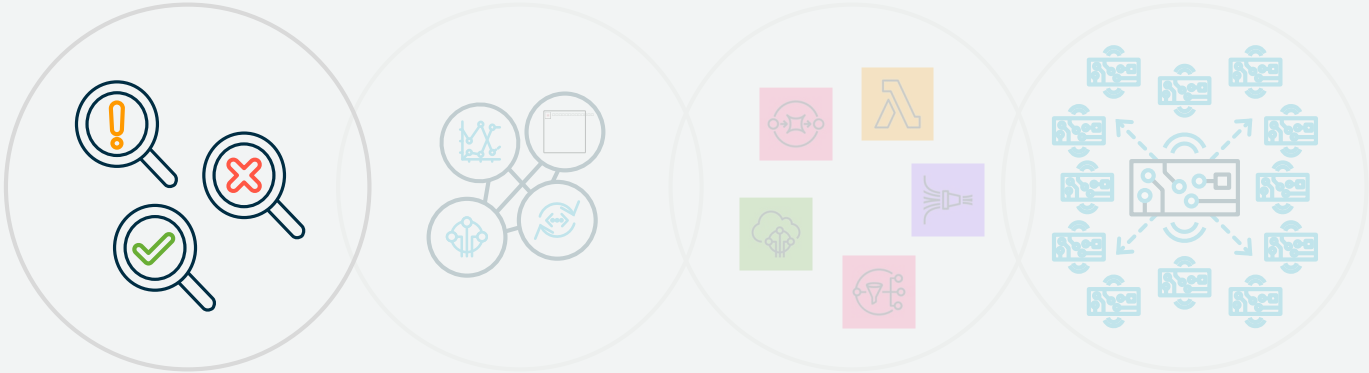


複数のソースから入力を受け入れ

簡単な論理式を使って複雑なイベントのパターンを認識

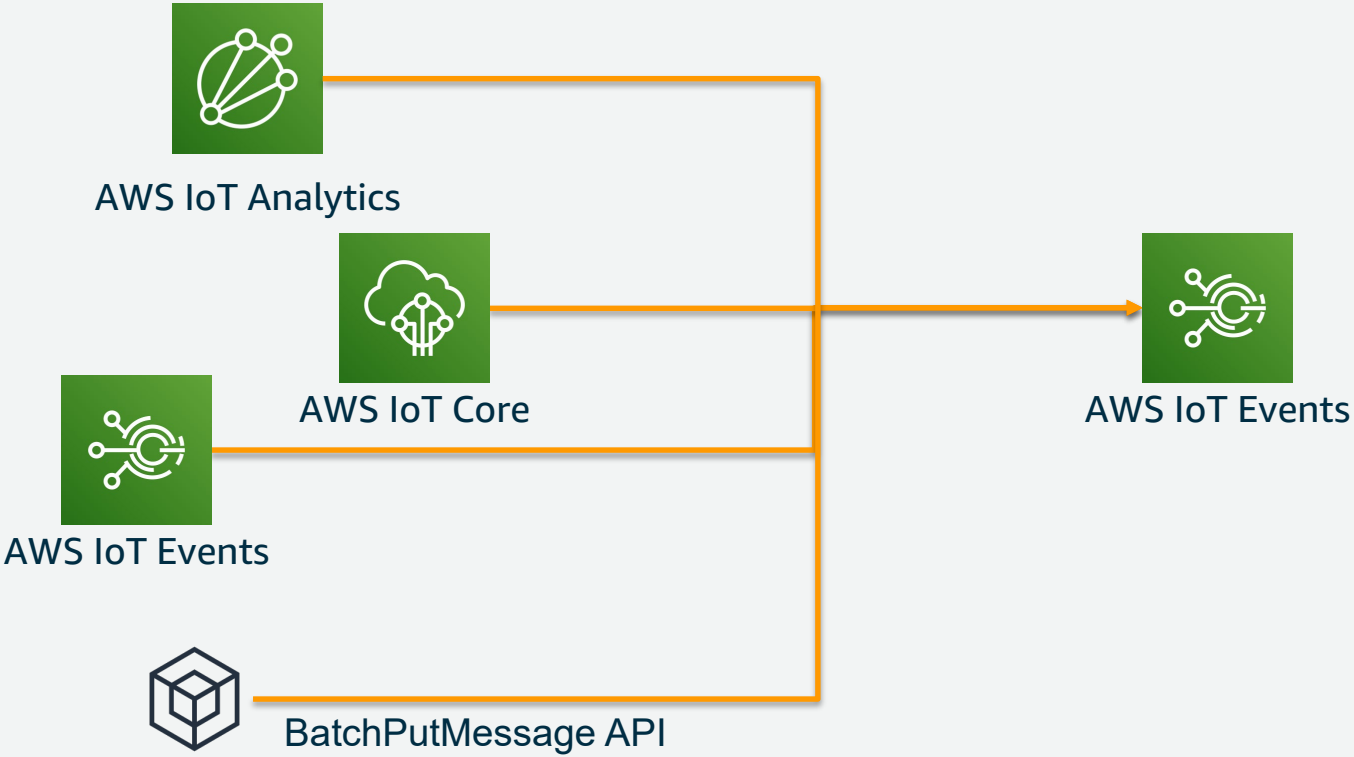
イベントに基づいたアクションの実行

スケーラビリティ



複数のソースから入力を受け入れ

複数のソースから入力を受けいれ



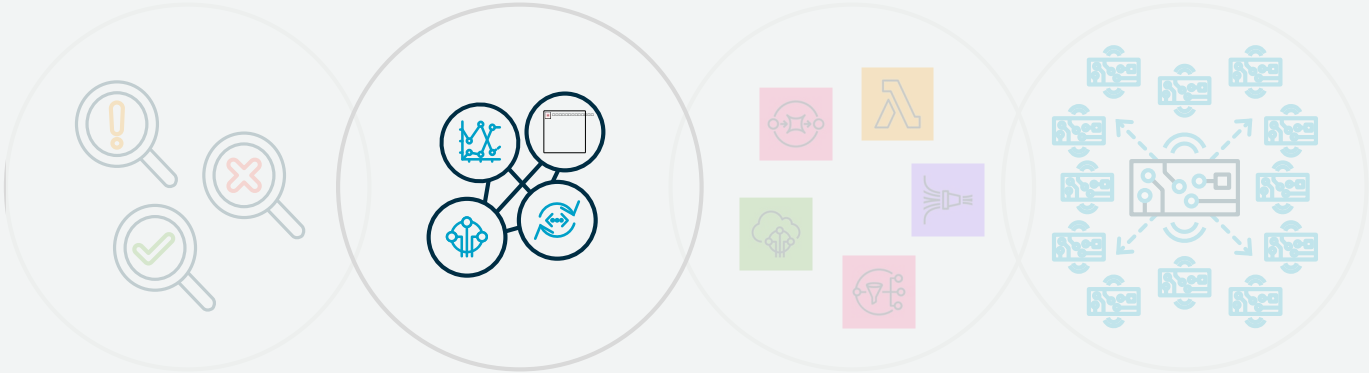
入力データのフォーマットはIoT Events側で定義

- 入力データはJSON形式をサポート

```
{
  "motorid": "Fulton-A32",
  "sensorData": {
    "pressure": 23,
    "temperature": 47
  }
}
```

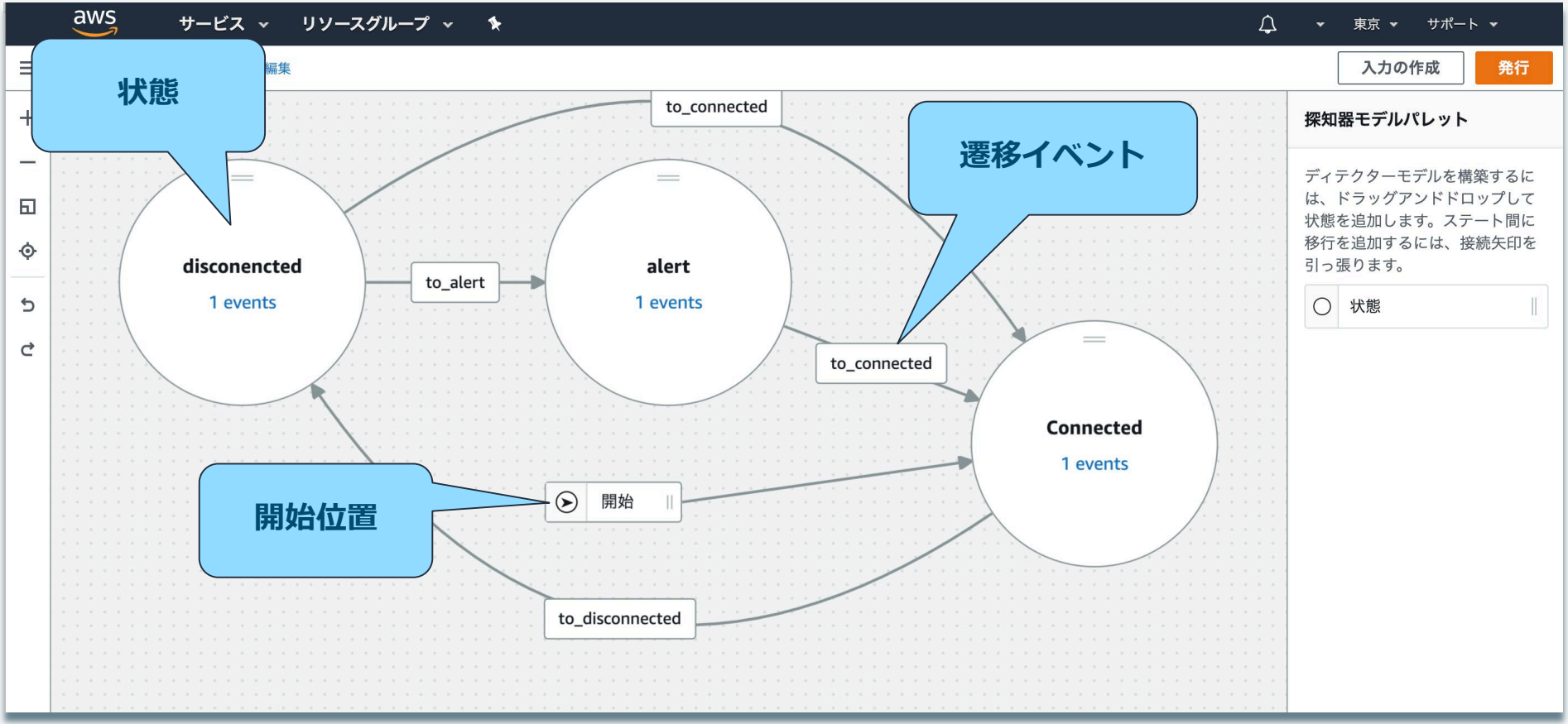
- IoT Eventsの探知器モデルからは、このデータを以下のよう参照します
  - $input.インプット名.motorid
  - $input.インプット名.sensorData.pressure
  - $input.インプット名.sensorData.temperature

固定　自分で命名　JSONの階層を”.”で区切る



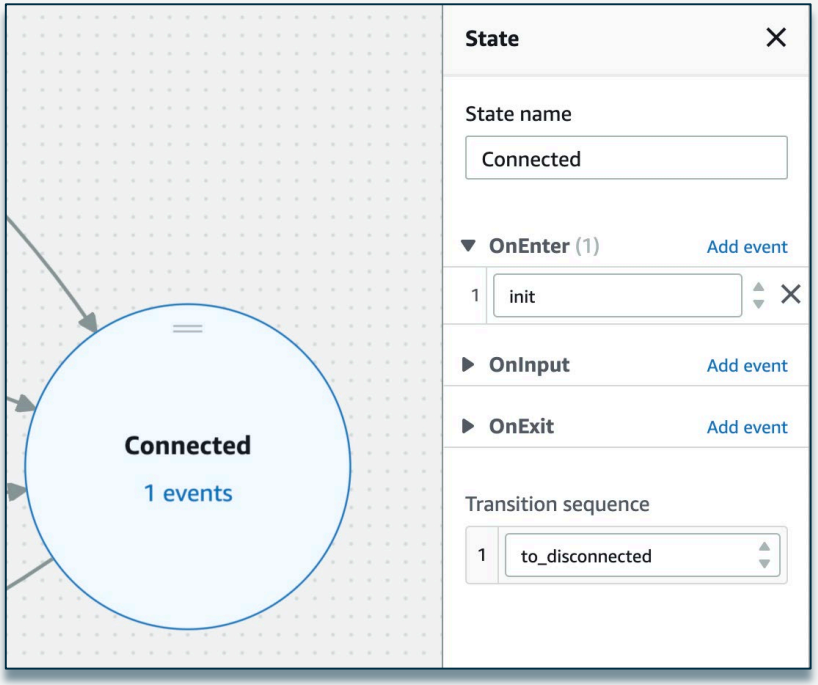
簡単な論理式を使って複雑なイベントのパターンを認識

探知器モデル



状態

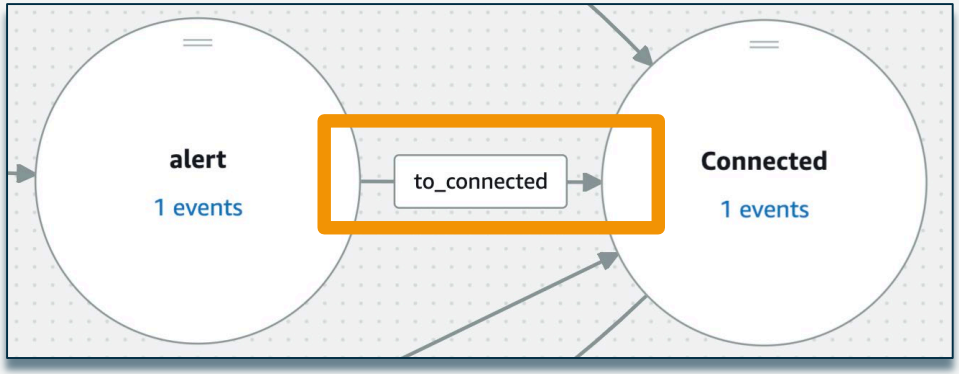
- 3種類のイベントを持つ
  - OnEnter
    - この状態に遷移してきた時に実行
  - OnInput
    - データが入力された時に実行
  - OnExit
    - この状態から抜ける時に実行

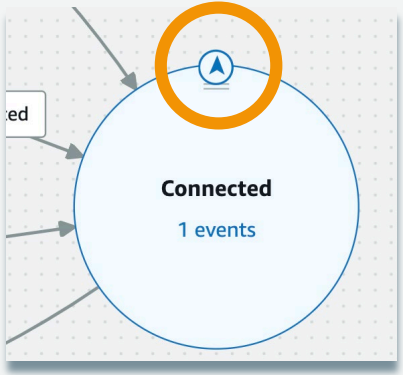


- それぞれのイベントでは条件を記述することが出来、その条件を満たしている時に、Actionを実行することが可能

遷移イベント

- 2つの状態間(Origin State, Destination State)の遷移条件を表します
- 条件を満たすと、Origin StateからDestination Stateに遷移する
- 遷移イベントが発生した際に、アクションを指定することが可能

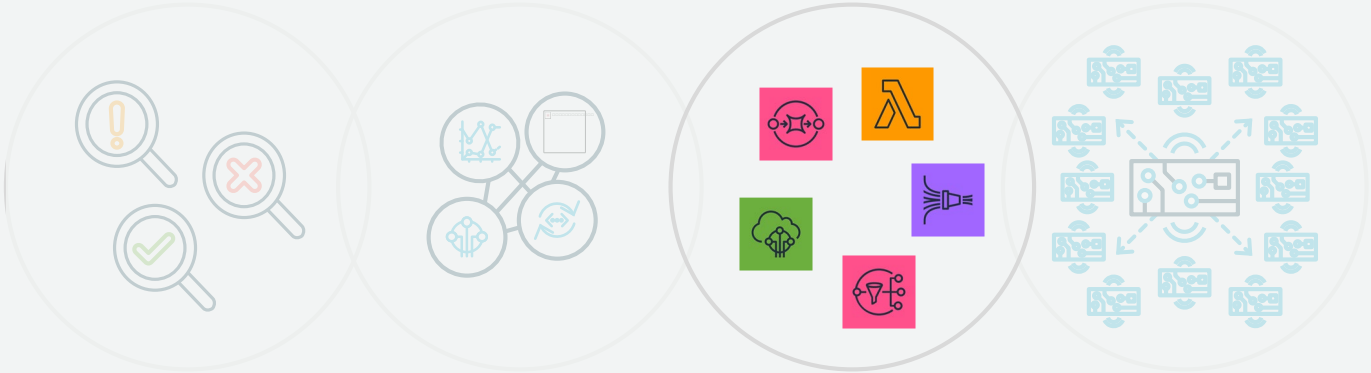




利用可能な条件式

- True, Falseを結果と返す式を指定
- 不等号(<,>, =)、四則演算(+, -, /, *)、論理式(^, |, &, , !,~)が利用可能
- 組み込み関数
    - 型変換(convert)
    - 文字列マッチング(startWith,contains,など)
    - ビット演算(bitor, bitandなど)
    - timeout(タイマーの完了を判断)

https://docs.aws.amazon.com/iotevents/latest/developerguide/expression-syntax.html



イベントに基づいたアクションの実行

利用できるアクション(ビルトイン)

- setTimer
    - タイマーの作成
- resetTimer
    - タイマーのリセット
- clearTimer
    - タイマーの削除
- setVariable
    - 変数の設定

利用できるアクション(AWSサービス連携)


Amazon Simple Notification Service


Amazon Simple Queue Service


Amazon Kinesis Data Firehose


Amazon DynamoDB


AWS IoT Events


AWS IoT SiteWise


AWS IoT Core


AWS Lambda

カスタムペイロード

- アクションを定義する際に、対象のサービスにペイロード (データ)を文字列またはJSONで渡すことができます

- ペイロードで指定できる情報は
    - 固定の文字列
    - $variable で参照できる変数
    - $input で参照できる入力データ

- 置き換えテンプレートなど利用して、SNSに送る情報をデータだけではなく、文章として作ることもできます

https://docs.aws.amazon.com/iotevents/latest/apireference/API_Payload.html

aws

置き換えテンプレート

${}と書くことで置き換えテンプレートを利用できます。変数の参照や、関数の利用など

変数を参照　**変数名**

例1)

- '**${**'Sensor ' + $variable.SensorID**}**’

“Sensor 10” のような変数の値と文字列を組み合わせた値が取得できる

入力を参照　**入力名**　**入力の要素**

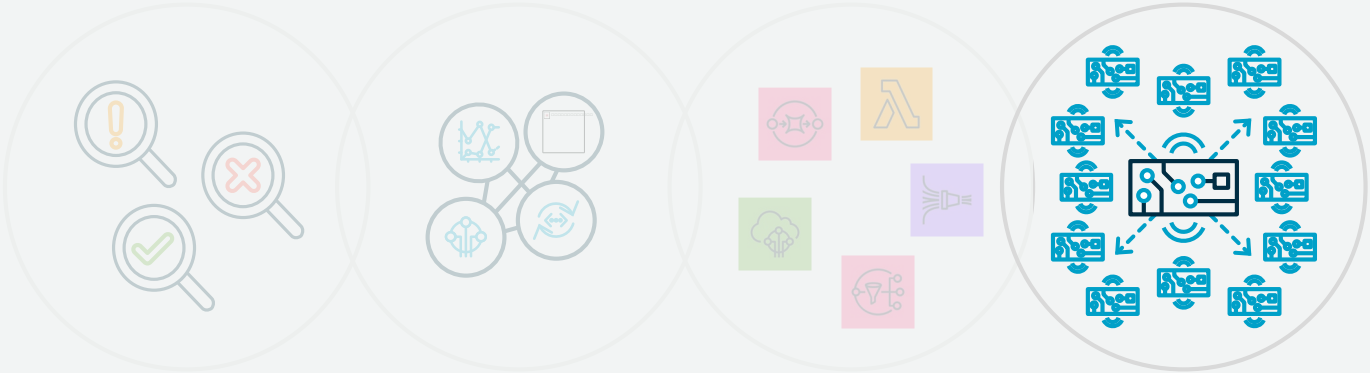
例2)

- 'Sensor 10: **${**$input.input_name.SensorID == 10**}**’

“Sensor 10: true” のように入力データを判定した結果が取得できる

https://docs.aws.amazon.com/iotevents/latest/developerguide/expression-syntax.html#expression-substitution-template

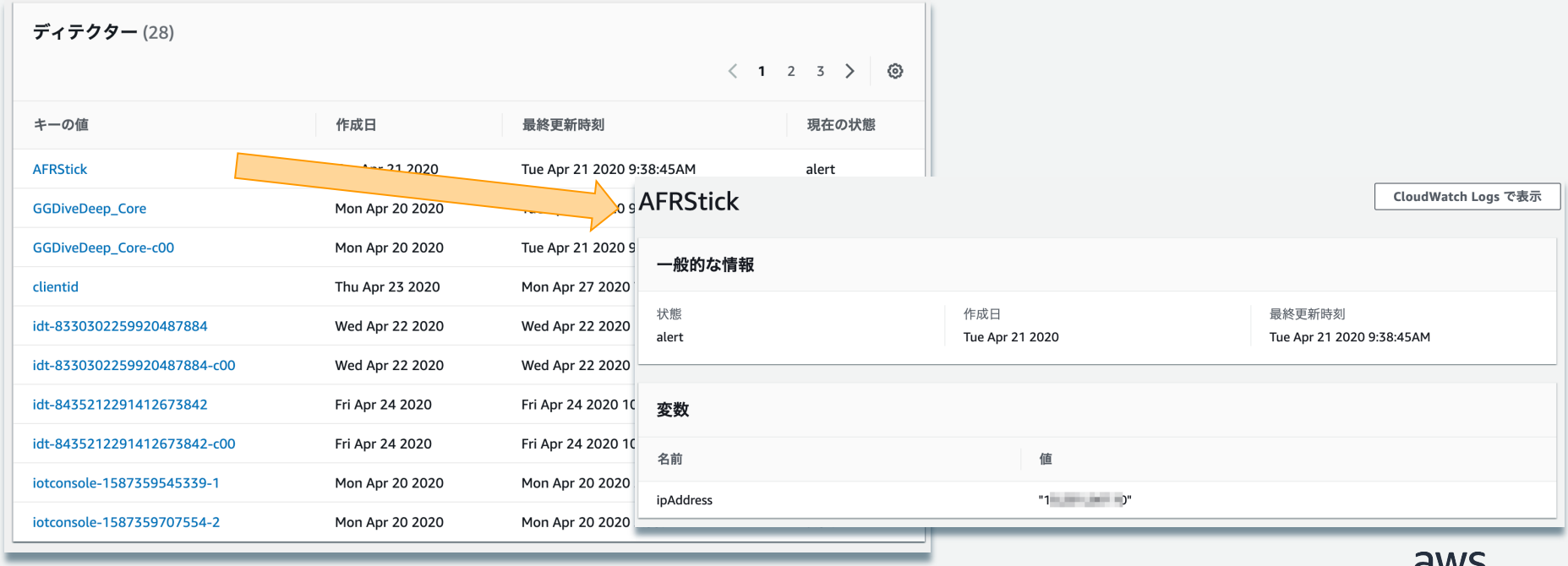
aws

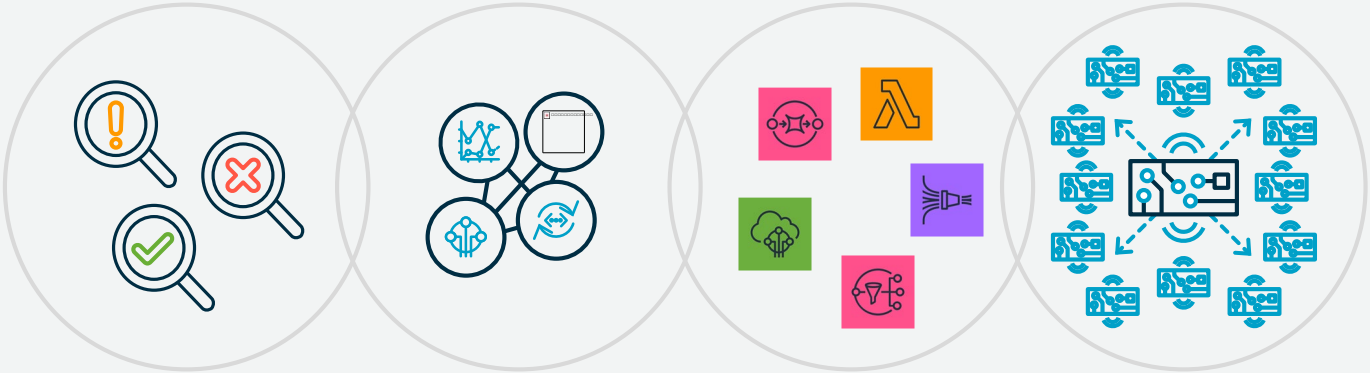


スケーラビリティ

入力データに合わせてスケール

- デバイスごとに状態を参照することが可能(この例は一意のキー毎に探知器モデルを作成)
- 新しいデバイス(キー)が登場すると自動的に状態の管理を開始





探知器モデルについて

探知器の生成メソッド

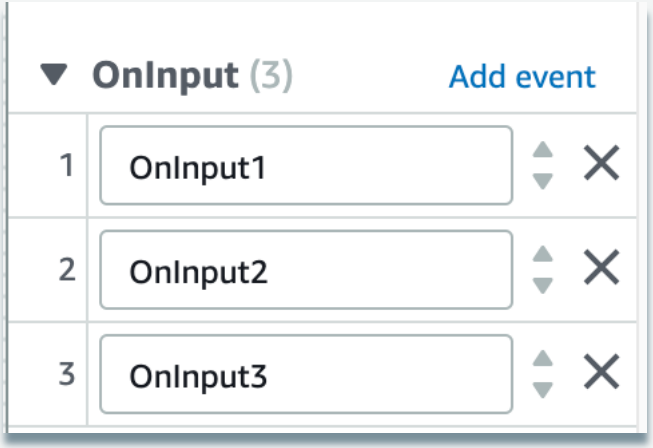
- 一意のキー毎に探知器を作成
    - デバイス毎など、複数のものに対して状態を管理したい時に向いている
- 単一の探知器を作成
    - 複数のデバイス(情報)から構成されるシステムの状態を管理したい時に向いている

探知器モデルの評価方法

- バッチ評価
    - 変数が更新され、すべてのイベント条件が評価された後にのみイベントが実行されます

- シリアル評価
    - 変数が更新され、イベント条件はイベントが定義されている順序で評価されます

探知器の評価方法(例 シリアル)



それぞれのイベントの条件とアクション

OnInput_1
- 条件
  $input.input_name.tempture > 20
- アクション
  変数 tempture に$input.input_name.tempture を設定

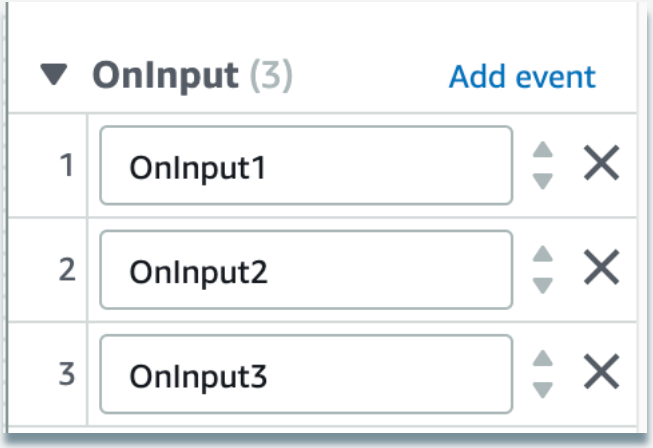
OnInput_2
- 条件
  $variable.tempture > 20
- アクション
  変数 status に文字列"greater than 20"を設定

OnInput_3
- 条件
  $variable.tempture <= 20
- アクション
  変数 status に文字列"lower or equal 20"を設定

入力データ

```
{
  "tempture": 30
}
```

探知器の評価方法(例 バッチ)



それぞれのイベントの条件とアクション

OnInput_1
- 条件
  $input.input_name.tempture > 20
- アクション
  変数 tempture に$input.input_name.tempture を設定

OnInput_2
- 条件
  $variable.tempture > 20
- アクション
  変数 status に文字列"greater than 20"を設定

OnInput_3
- 条件
  $variable.tempture <= 20
- アクション
  変数 status に文字列"lower or equal 20"を設定

入力データ

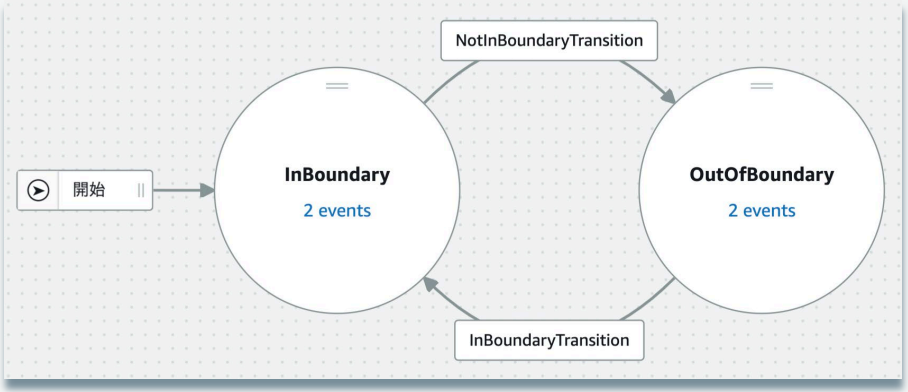
```
{
  "tempture": 30
}
```

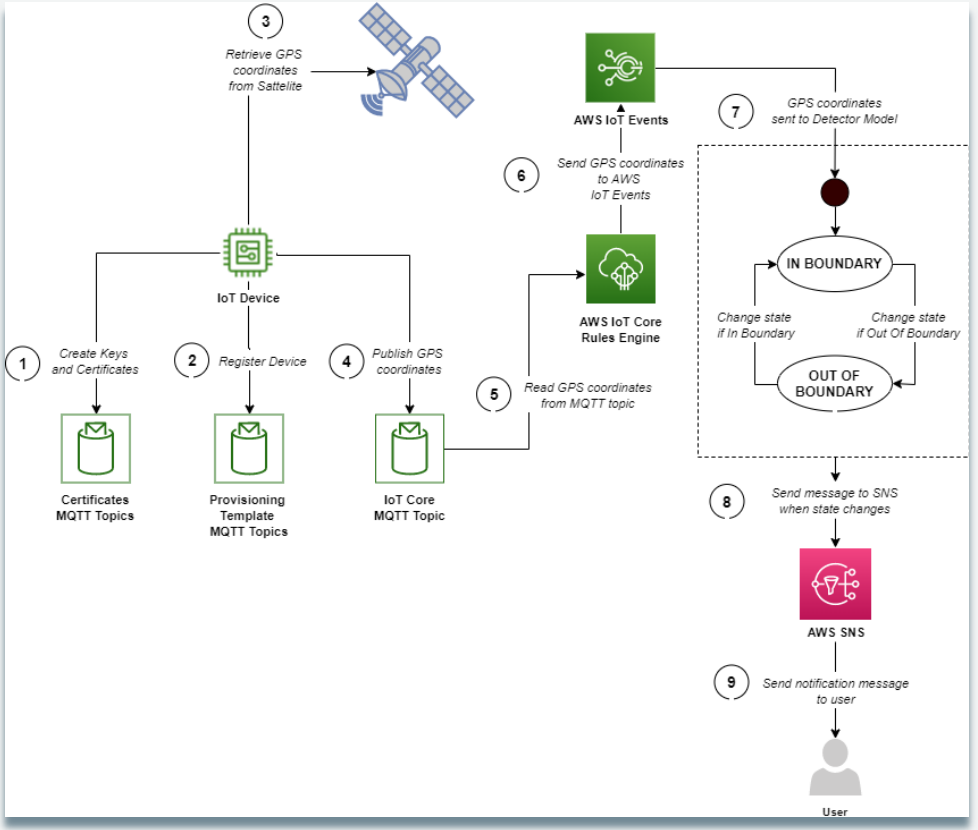
AWS IoT Eventsを使った ユースケースの紹介

1) デバイスの位置情報を監視

https://aws.amazon.com/jp/blogs/news/monitor-iot-device-geolocation-with-aws-iot-events/

- 車両が予想されるジオロケーションの境界から外れるとサポートチームに通知

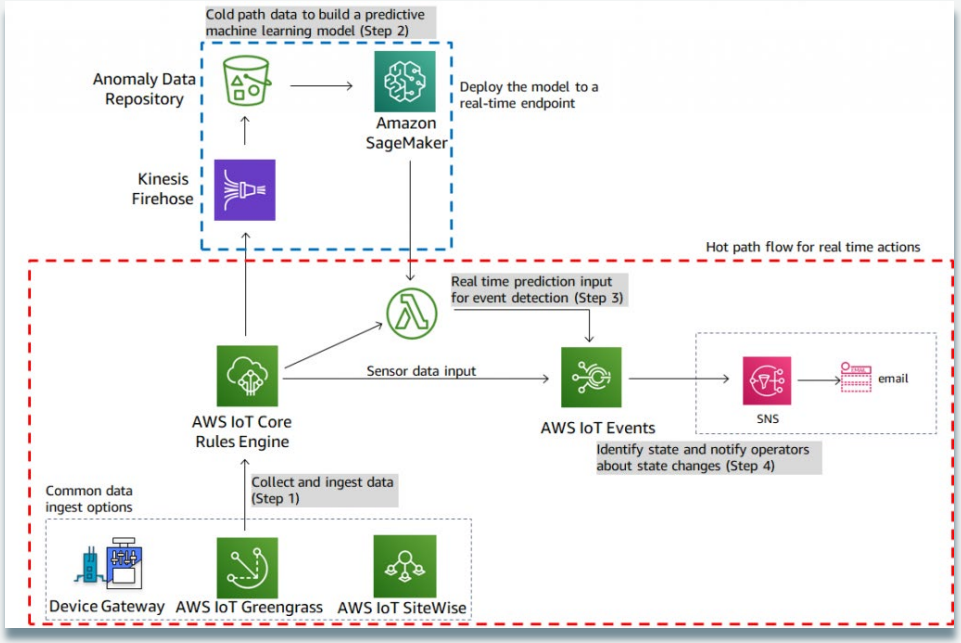




2) AWS IoT EventsとMLを組み合わせた設備の管理

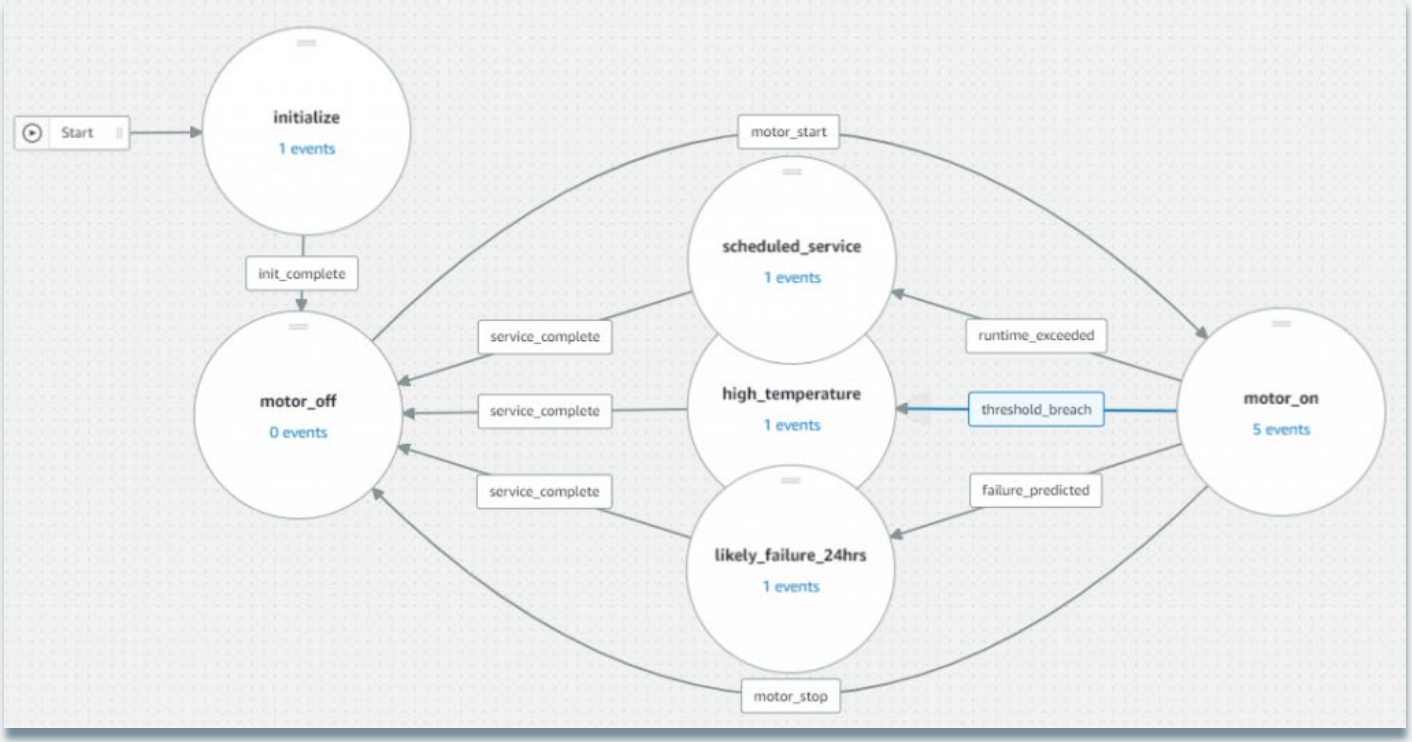
https://aws.amazon.com/jp/blogs/iot/asset-maintenance-with-aws-iot-services-predict-and-respond-to-potential-failures-before-they-impact-your-business/

- 工場設備の各種センサーの値を収集
- 一定時間異常な高温やポンプの異常をを検出したら緊急メンテを通知
- 一定時間連続稼働したら計画停止



2) AWS IoT EventsとMLを組み合わせた設備の管理

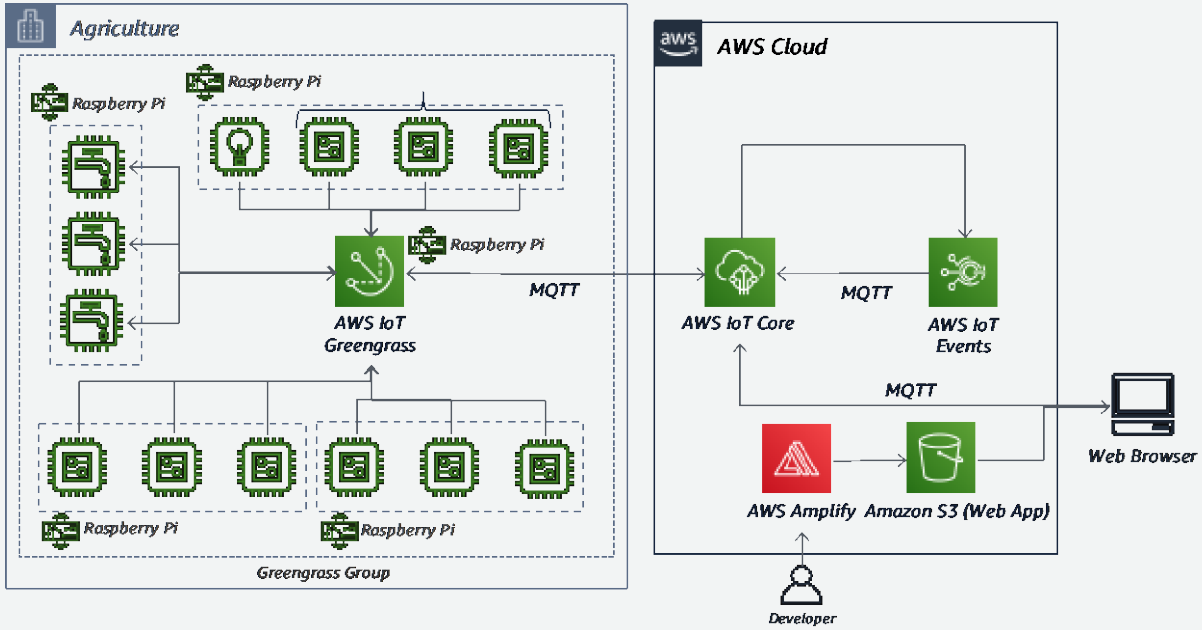
https://aws.amazon.com/jp/blogs/iot/asset-maintenance-with-aws-iot-services-predict-and-respond-to-potential-failures-before-they-impact-your-business/



3) 畑の自動水やり

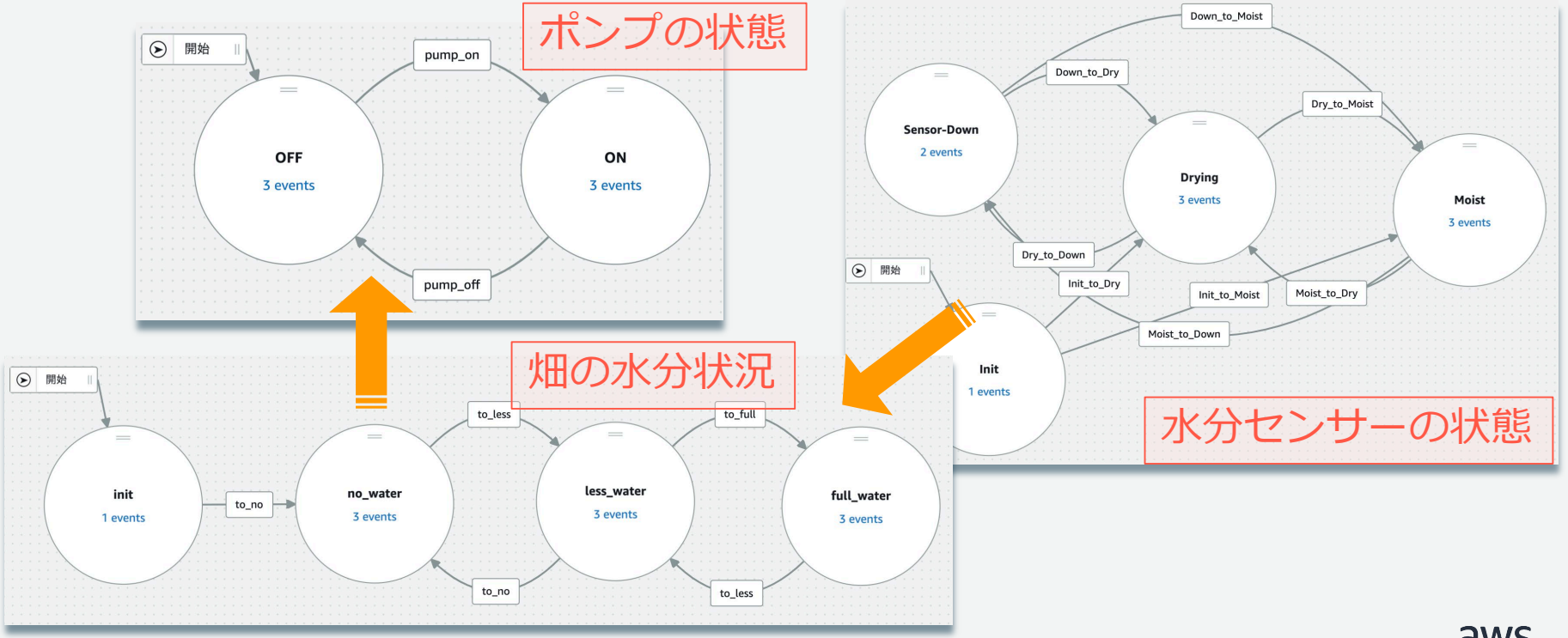
https://github.com/aws-samples/aws-builders-fair-projects/tree/master/reinvent-2019/fully-automated-farm

- 畑の自動水やり機では、水分量の減少を検出するとポンプを起動する
- 複数の探索器を連携



3) 畑の自動水やり

https://github.com/aws-samples/aws-builders-fair-projects/tree/master/reinvent-2019/fully-automated-farm



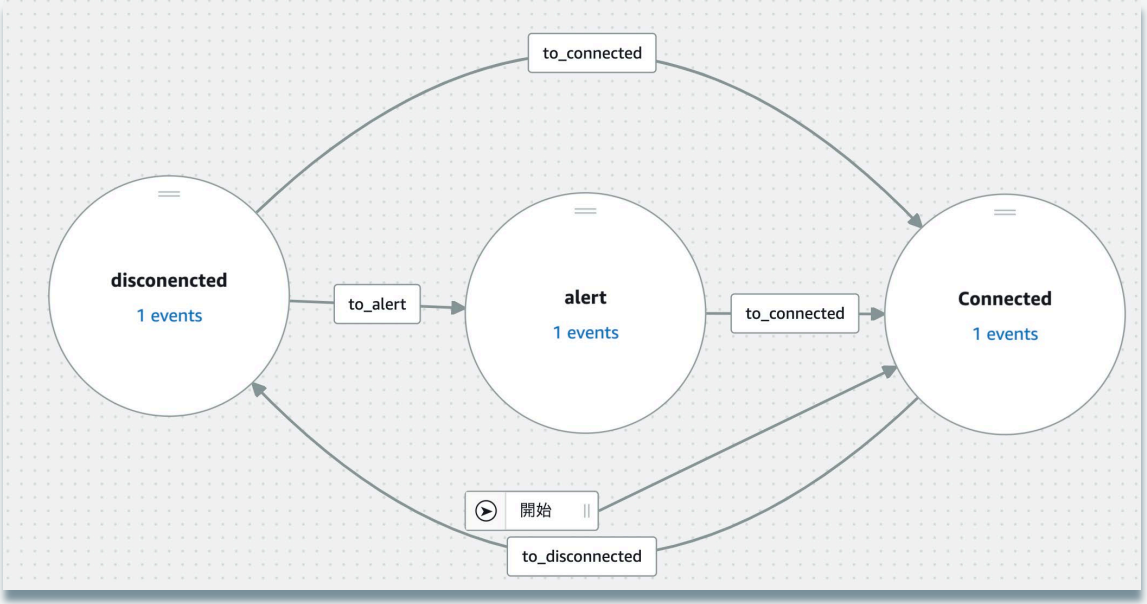
AWS IoT Eventsの始め方

テンプレートを使ってみよう

AWS IoT Eventsワークショップを試してみよう

https://aws-iot-events-for-beginners.workshop.aws/

- AWS IoT Coreのライフサイクルイベントを利用し、コネクションのステータスが一定時間disconnectedとなる場合に通知する仕組みを体験



探知器モデルの開発時のベストプラクティス

- AWS IoT EventsのCloudWatch ログを有効にしましょう
- 定期的にパブリッシュ(発行)を行い、探索器を保存しましょう
- 一定期間探索器を利用しない(データも送らない)場合は、モデルとデータを保存しましょう
    - 利用が無い場合は削除の30日前に通知をし、その後削除されます

https://docs.aws.amazon.com/iotevents/latest/developerguide/best-practices.html

aws

探知器モデルのエクスポートとインポート

料金について

2020/9/16時点 東京リージョンの金額

料金の考え方は、メッセージの評価回数が基本
外部アクションの実行も、評価回数としてカウントする。１つのメッセージ評価に対して、2つのアクション実行は無料

- 最初の1億回のメッセージ評価
    - 100 万回のメッセージ評価あたり 16.50USD
- 次の4億回のメッセージ評価
    - 100 万回のメッセージ評価あたり 11.00USD
- 次の45億回のメッセージ評価
    - 100 万回のメッセージ評価あたり 5.50USD
- 50 億回以上のメッセージ評価
    - 100 万回のメッセージ評価あたり 3.30USD

https://aws.amazon.com/jp/iot-events/pricing/

まとめ

まとめ

- AWS IoT Eventsを利用することで複雑な条件を組み合わせながら、デバイスやシステムの状態を管理する仕組みをコードレスで作成できる

- 複数のサービスからの入力を受付、AWSの他のサービスに対してアクションを起こすことが出来る

- マネージドサービスなので、デバイスが増えてきてもスケールする

Q&A

お答えできなかったご質問については
AWS Japan Blog 「https://aws.amazon.com/jp/blogs/news/」にて
後日掲載します。

AWS の日本語資料の場所「AWS 資料」で検索

https://amzn.to/JPArchive

AWS Well-Architected 個別技術相談会

毎週”W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に対策などを相談することも可能
- 申込みはイベント告知サイトから

(https://aws.amazon.com/jp/about-aws/events/)

AWS イベント で[検索]

ご視聴ありがとうございました

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive