



# Amazon EventBridge の 便利な機能 + 新機能

「SaaS + サーバーレス/EventBridge」で開発加速化、運用作業軽減

28th Oct 2020

Kensuke Shimokawa



@\_kensh

# Who am I?

## Name

Kensuke Shimokawa

## Company

Amazon Web Services Japan K.K.

## Role

Serverless Specialist Solutions Architect



@\_kensh

# 本セッションは

## 想定される聴講者

- AWSサーバーレス をご利用されている事業者、デベロッパー
- SaaSを今後利用したビジネスを考えている方
- サーバーレスサービス構築に興味がある方

## ゴール

- システムでのAmazon EventBridgeの使い所を理解していただく

# Table of contents

- SaaSを利用した連携での課題とその解決
  - Event の取り扱い、管理どうする?
  - 開発環境で簡単にSchemaを使いたい
  - SchemaをEventから発見したい
  - Event Schemaの検証をしたい
  - Event送信の信頼性を高めたい
- まとめ

# SaaSを利用した連携での課題

## Event & SaaS

### よくある課題

- Event の取り扱い、管理どうする?
- 開発環境で簡単にSchemaを使いたい
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- SaaS との連携の作り込み
- 利用するSaaSが増えていく
- 連携のためのクレデンシャル管理

## Event & SaaS

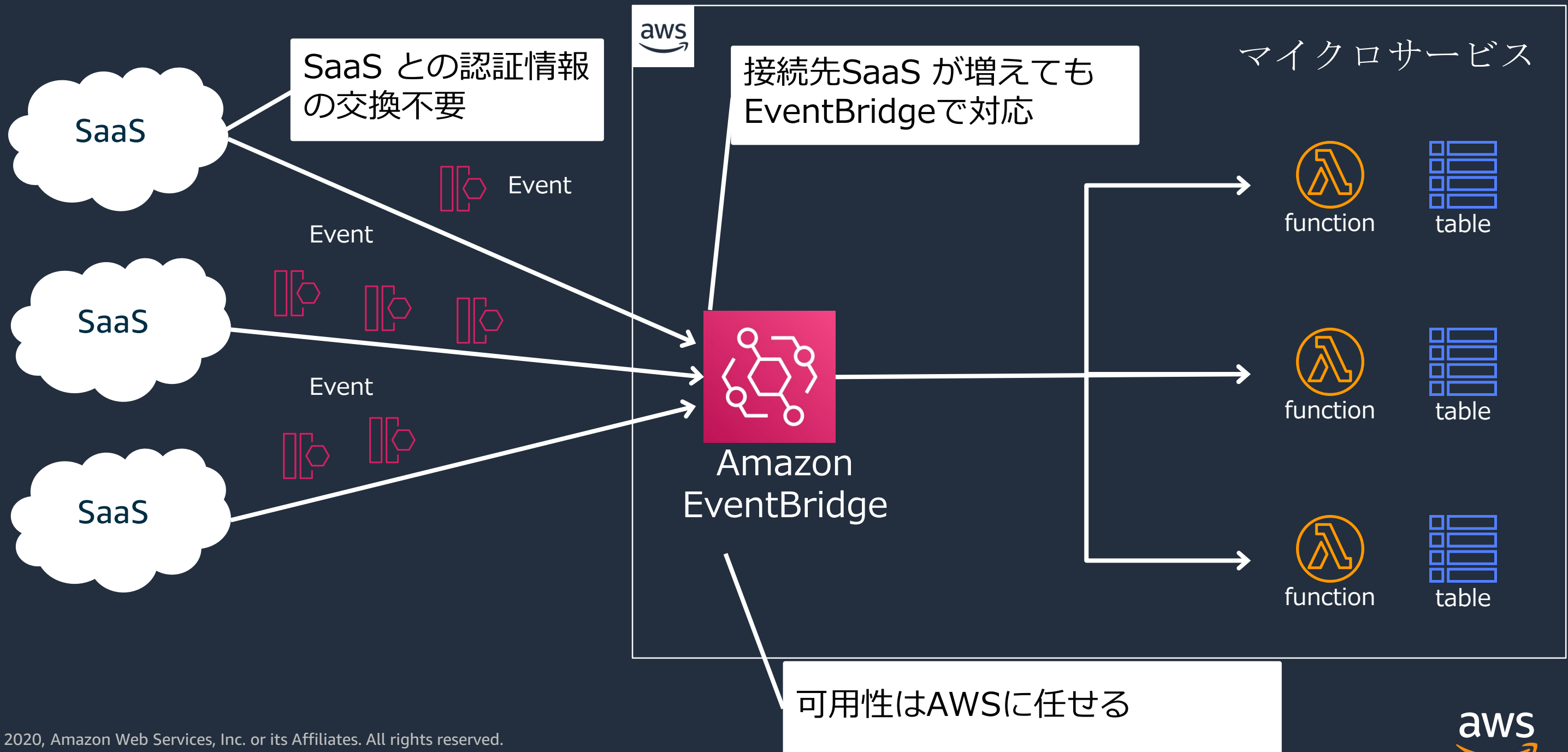
### よくある課題

- Event の取り扱い、管理どうする?
- 開発環境で簡単にSchemaを使いたい
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- SaaS との連携の作り込み
- 利用するSaaSが増えていく
- 連携のためのクレデンシャル管理

# SaaS と連携する際のポイント





# SaaSを利用した連携での課題例

## Event & SaaS

### よくある課題

- Event の取り扱い、管理どうする?
- 開発環境で簡単にSchemaを使いたい
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# SaaSを利用した連携での課題例

## Event & SaaS

### よくある課題

- **Event** の取り扱い、管理どうする?
- 開発環境で簡単にSchemaを使いたい
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# Amazon EventBridge Schema Registry

- **Schema Registry**
  - EventBridge (CloudWatch Events) でやり取りされる様々なデータのスキーマを、コンソールからまとめて公開
    - e.g. EC2 インスタンスの状態変化通知のスキーマ

aws.ec2@EC2InstanceStateChangeNotification

```
Version 1 Created on Dec 1, 2019, 09:11 AM GMT+9
Action ▼
8 "components": {
9   "schemas": {
10    "AWSEvent": {
11     "type": "object",
12     "required": ["detail-type", "resources", "id", "source", "time", "detail", "region", "version", "account"],
13     "x-amazon-events-detail-type": "EC2 Instance State-change Notification",
14     "x-amazon-events-source": "aws.ec2",
15     "properties": {
16      "detail": {
17       "$ref": "#/components/schemas/EC2InstanceStateChangeNotification"
18      },
19      "detail-type": {
20       "type": "string"
21      },
22      "resources": {
```

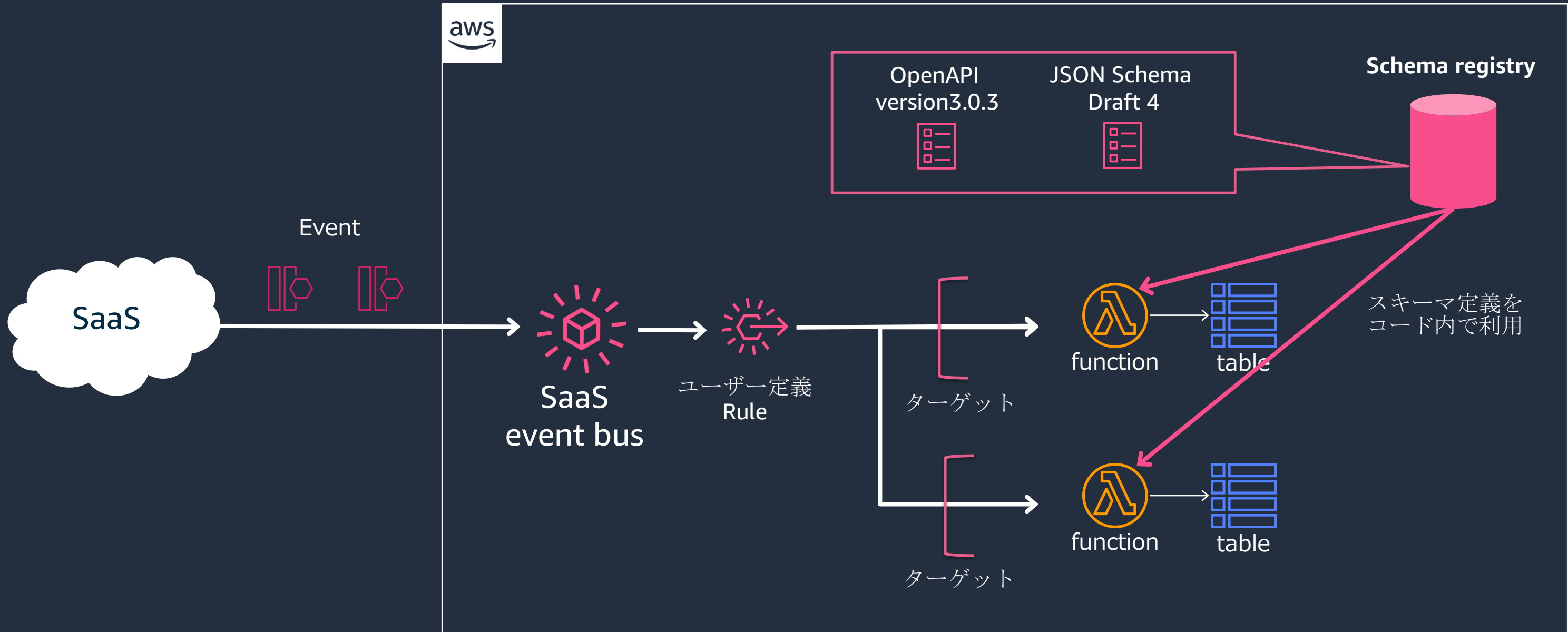
プロパティの一覧とそれぞれの型 (String や独自の型など)

```
Version 1 Created on Dec 1, 2019, 09:11 AM GMT+9
48 },
49 "EC2InstanceStateChangeNotification": {
50  "type": "object",
51  "required": ["instance-id", "state"],
52  "properties": {
53   "instance-id": {
54    "type": "string"
55   },
56   "state": {
57    "type": "string"
58   }
59  }
60 }
```

独自の型の定義も確認できる

<https://aws.amazon.com/jp/blogs/news/new-amazon-eventbridge-schema-registry-is-now-generally-available/>

# Schema Registry (OpenAPI / JSON Schema)



## Event & SaaS

### よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- 開発環境で簡単にSchemaを使いたい
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

## Event & SaaS

### よくある課題

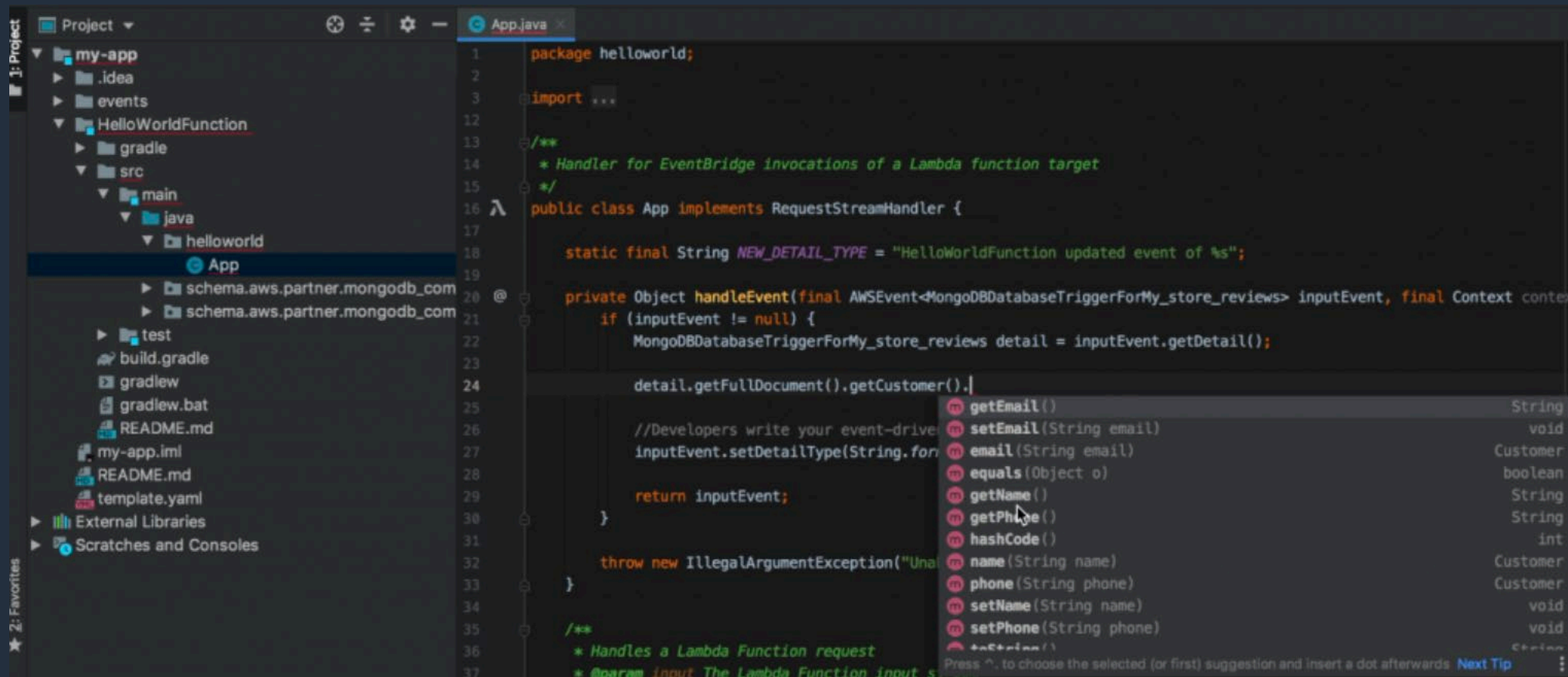
- ~~Eventの取り扱い、管理どうする?~~
- **開発環境で簡単にSchemaを使いたい**
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# Schema Code Bindings

- Code Bindings
  - スキーマ定義を言語ごとにダウンロード可能
  - Java 8、Python 3.6、TypeScript 3 の 3 種類に対応
  - アプリケーション開発の際に、ダウンロードしたスキーマ定義をインポートしておくことで、コード補完やコンパイル時のエラー検出に役立てることができる



The screenshot shows an IDE window with a project structure on the left and a Java file named 'App.java' open in the center. The project structure includes a 'my-app' directory with sub-directories like 'events', 'HelloWorldFunction', 'gradle', 'src', 'main', 'java', and 'helloworld'. The 'App.java' file contains the following code:

```
1 package helloworld;
2
3 import ...
4
5
6
7
8
9
10
11
12
13 /**
14  * Handler for EventBridge invocations of a Lambda function target
15  */
16 public class App implements RequestStreamHandler {
17
18     static final String NEW_DETAIL_TYPE = "HelloWorldFunction updated event of %s";
19
20     private Object handleEvent(final AWSEvent<MongoDBDatabaseTriggerForMy_store_reviews> inputEvent, final Context context) {
21         if (inputEvent != null) {
22             MongoDBDatabaseTriggerForMy_store_reviews detail = inputEvent.getDetail();
23
24             detail.getFullDocument().getCustomer().|
25
26             //Developers write your event-driven
27             inputEvent.setDetailType(String.for
28
29             return inputEvent;
30         }
31
32         throw new IllegalArgumentException("Unal
33     }
34
35     /**
36     * Handles a Lambda Function request
37     * @param input The Lambda Function input s
```

A popup menu is visible over the code, listing methods from the 'Customer' class:

- getEmail() String
- setEmail(String email) void
- email(String email) Customer
- equals(Object o) boolean
- getName() String
- getPhone() String
- hashCode() int
- name(String name) Customer
- phone(String phone) Customer
- setName(String name) void
- setPhone(String phone) void
- +toString() String

## Event & SaaS

### よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- ~~開発環境で簡単にSchemaを使いたい~~
- SchemaをEventから発見したい
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~



# SaaSを利用した連携での課題例

## Event & SaaS

### よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- ~~開発環境で簡単にSchemaを使いたい~~
- **SchemaをEventから発見したい**
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# Amazon EventBridge Schema Discovery

- Schema Discovery

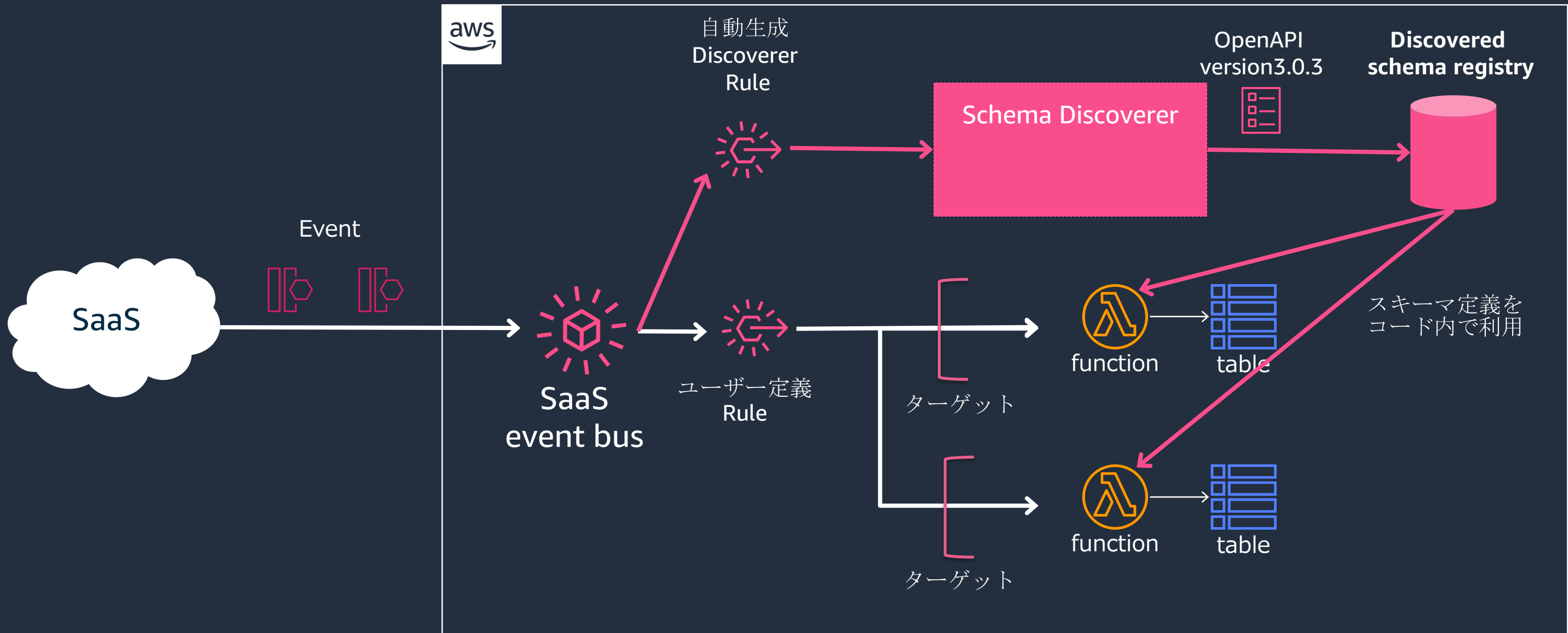
- SaaS から EventBridge へ送信されるデータについては、EventBridge へ流入するデータから **OpenAPI形式** のスキーマを自動生成
  - e.g. MongoDB Atlas (MongoDB 社にて提供されるマネージドサービス) へデータの CRUD 処理が行われた場合のスキーマ

```
aws.partner-mongodb.com-stitch.trigger-5dee176ba96a577ca57e00a8@MongoDB...  
Version 4 Created on Dec 9, 2019, 07:46 PM GMT+9 Version 4 ▼ Save as a new version Action  
9 "schemas": {  
0   "AWSEvent": {  
1     "type": "object",  
2     "required": ["detail-type", "resources", "detail", "id", "source", "time", "region", "version", "account"],  
3     "x-amazon-events-detail-type": "MongoDB Database Trigger for sample_airbnb.listingsAndReviews",  
4     "x-amazon-events-source": "aws.partner/mongodb.com/stitch.trigger/5dee176ba96a577ca57e00a8",  
5     "properties": {  
6       "detail": {  
7         "$ref": "#/components/schemas/MongoDBDatabaseTriggerForSample_airbnb.listingsAndReviews"  
8       },  
9       "account": {  
0         "type": "string"  
1       },  
2       "detail-type": {  
3         "type": "string"  
4       }  
5     }  
6   }  
7 }  
8 }  
9 }  
0 }  
1 }  
2 }  
3 }
```

プロパティの一覧とそれぞれの型  
(String や独自の型など)

AWS のサービスと全く同じ形で、  
パートナーから送信されるデータ  
のスキーマも確認できる

# Schema Discovery (OpenAPI)



# Event & SaaS

## よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- ~~開発環境で簡単にSchemaを使いたい~~
- ~~SchemaをEventから発見したい~~
- Event Schemaの検証をしたい
- Event送信の信頼性を高めたい

## よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# Event & SaaS

## よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- ~~開発環境で簡単にSchemaを使いたい~~
- ~~SchemaをEventから発見したい~~
- **Event Schemaの検証をしたい**
- **Event送信の信頼性を高めたい**

## よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# OpenAPI vs JSON Schema



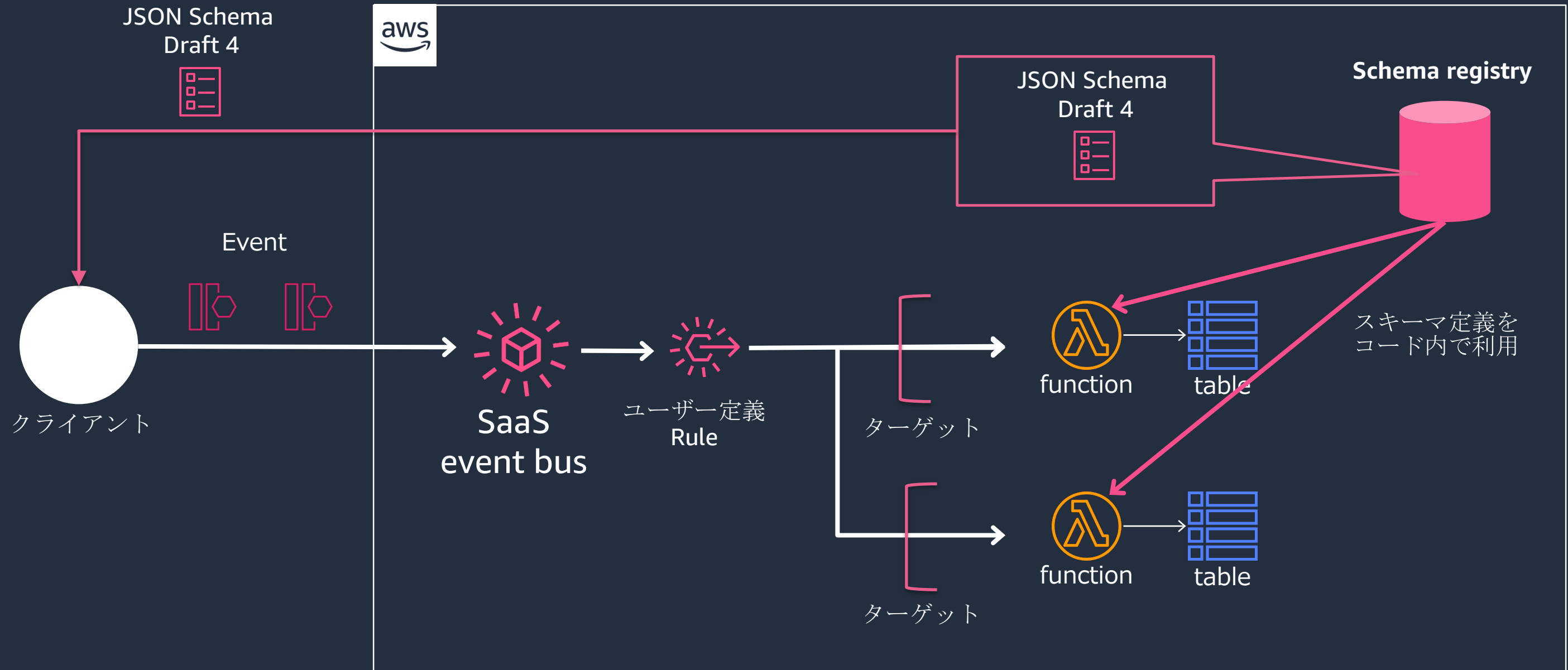
- より高速なビルド時間が期待できる



## JSON Schema

- **\$schema が利用可能**
  - \$schemaは、JSONフラグメントが実際にはJSONスキーマの一部であることを宣言するために使用
- **Hyper-Schema が利用可能**
  - JSON Hyper-Schemaは、HTTPなどのハイパーメディアを介してリモートJSONリソースを処理して、JSONドキュメントに注釈を付けるためのJSONスキーマボキャブラリー
- **実行時のバリデーションが可能**

# Schema Validation (JSON Schema)



# Schema Validation (JSON Schema)

```
JSON Schema
{
  "definitions": {
    "person": {
      "id": "#person",
      "type": "object",
      "properties": {
        "id": {
          "type": "integer"
        },
        "name": {
          "pattern": "^[A-Za-z_]+$",
          "type": ["string", "null"]
        },
        "email": {
          "format": "email",
          "type": ["string", "null"]
        },
        "address": {
          "minLength": 3,
          "maxLength": 1000000,
          "type": "string"
        }
      }
    }
  }
},
```

```
Validator
>>> from jsonschema import validate
>>> # A sample schema, like what we'd get from json.load()
>>> schema = {
    "type": "object",
    "properties": {
      "id": {
        "type": "integer"
      },
      "name": {
        "pattern": "^[A-Za-z_]+$",
        "type": ["string", "null"]
      }
    }
  }

>>> # If no exception is raised by validate(), the instance is valid.
>>> validate(
    instance={"name": "Tom", "id": 1001}, schema=schema)
>>> validate(
    instance={"name": "Tom", "id": "Invalid" }, schema=schema, ... )
# doctest: +IGNORE_EXCEPTION_DETAIL
Traceback (most recent call last): ...
ValidationError: 'Invalid' is not of type 'number'
```

<https://json-schema.org/implementations.html#validators>  
<https://github.com/Julian/jsonschema>



# Event & SaaS

## よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- ~~開発環境で簡単にSchemaを使いたい~~
- ~~SchemaをEventから発見したい~~
- ~~Event Schemaの検証をしたい~~
- Event送信の信頼性を高めたい

## よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

# SaaSを利用した連携での課題例

## Event & SaaS

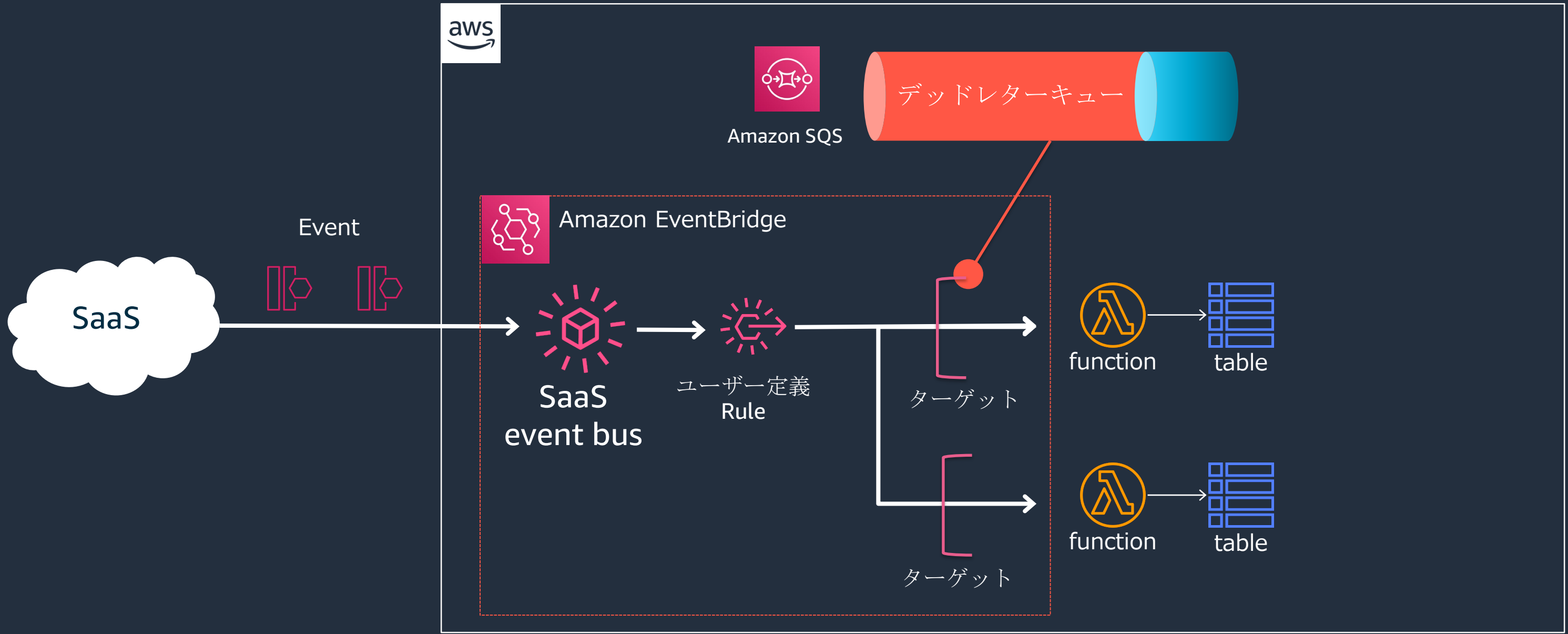
### よくある課題

- ~~Eventの取り扱い、管理どうする?~~
- ~~開発環境で簡単にSchemaを使いたい~~
- ~~SchemaをEventから発見したい~~
- ~~Event Schemaの検証をしたい~~
- **Event送信の信頼性を高めたい**

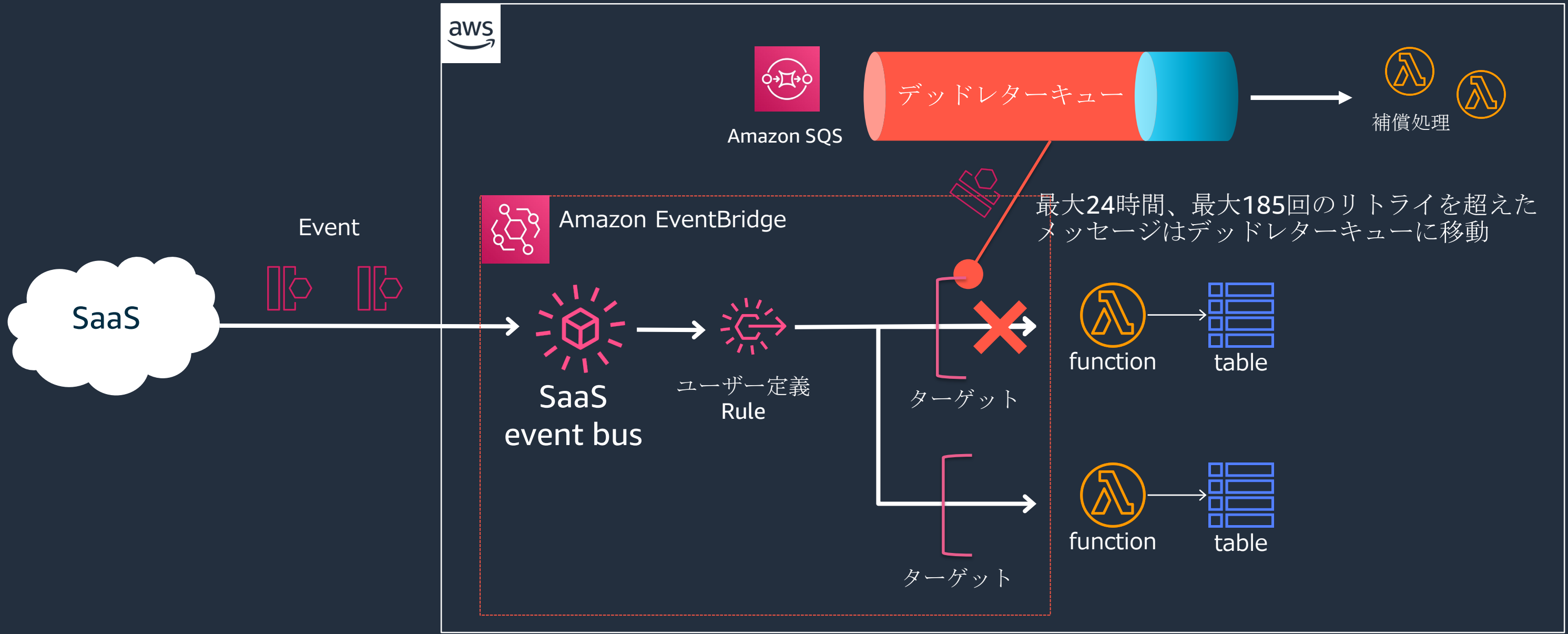
### よくある課題

- ~~SaaSとの連携の作り込み~~
- ~~利用するSaaSが増えていく~~
- ~~連携のためのクレデンシャル管理~~

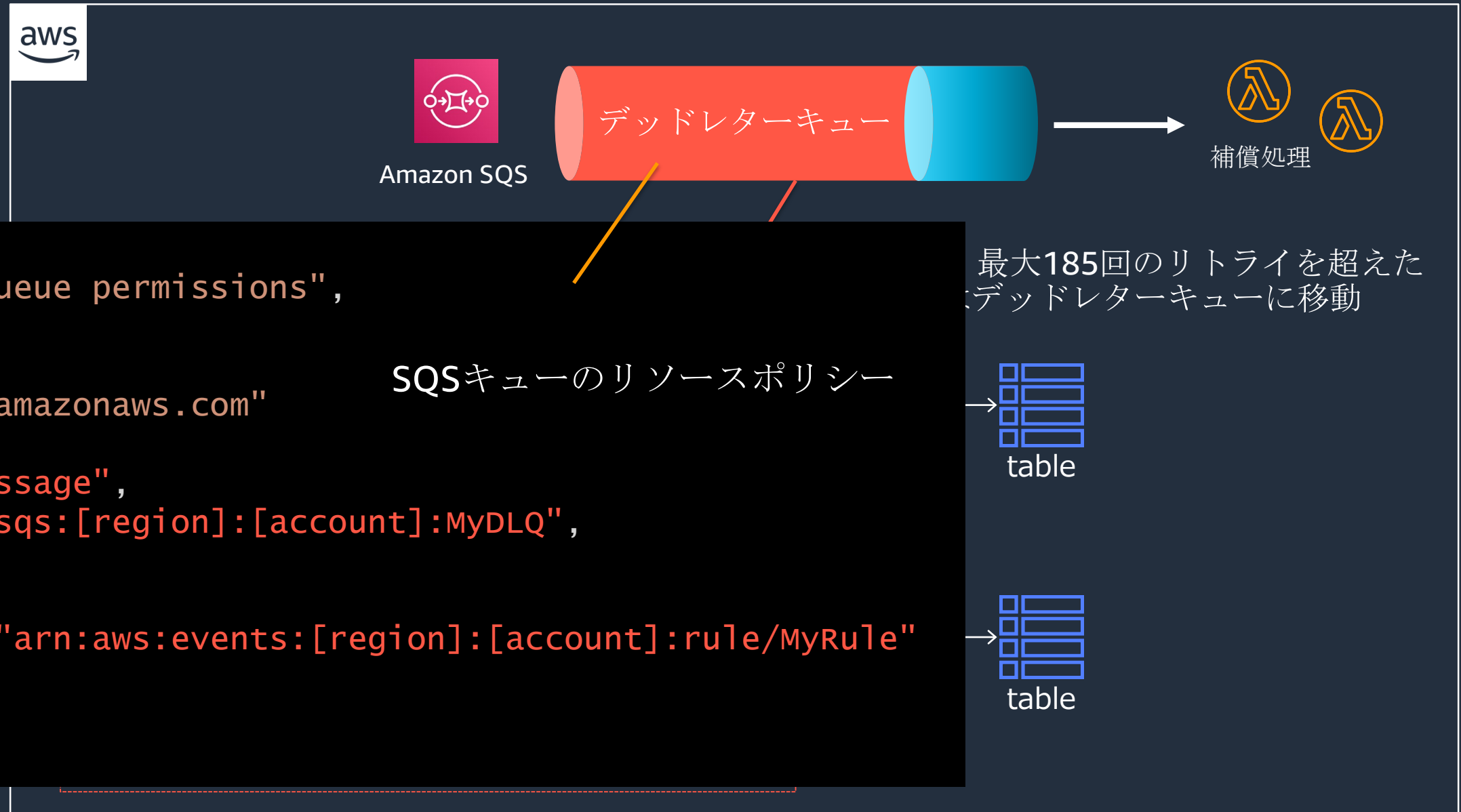
# EventBridgeで信頼性を高める



# EventBridgeで信頼性を高める



# EventBridgeで信頼性を高める



# 再試行ポリシーとデッドレターキューの設定

## ▼ 再試行ポリシーとデッドレターキュー

### ポリシーを再試行 [情報](#)

EventBridge は、デフォルトで最大 **24 時間** 失敗した呼び出しの再試行を試みます。イベントの最大経過時間と再試行する最大回数は以下で指定できます。

### イベントの最大有効期間 - オプション

未処理のイベントを保持する最大時間数。デフォルト値は 24 時間です。

### 再試行回数 - オプション

エラーが発生した後、ターゲットへのイベントの送信を再試行する最大回数。デフォルト値は 185 回です。

### デッドレターキュー [情報](#)

未処理のイベントはスタンダード SQS キューに送信することができます

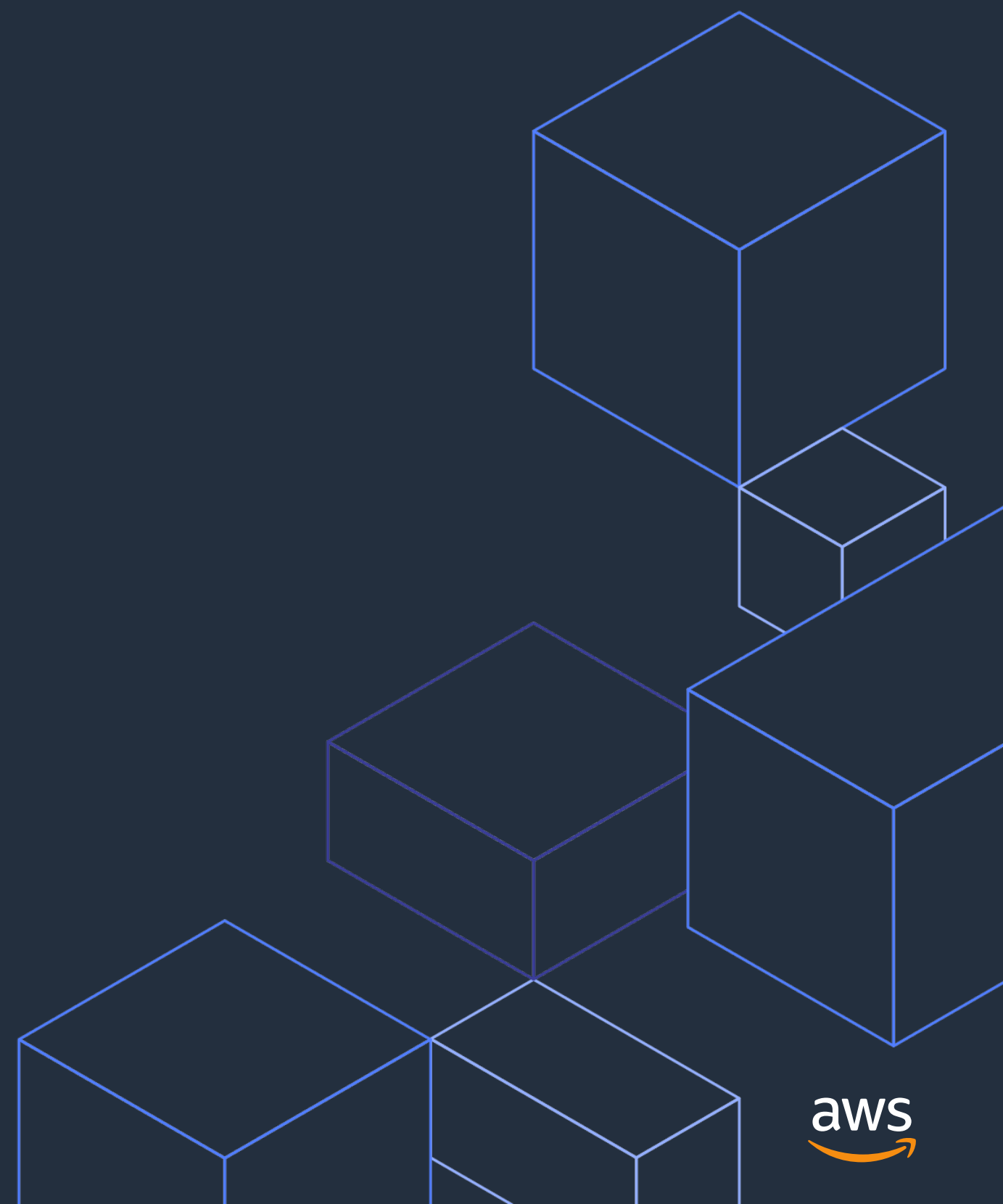
- なし
- デッドレターキューとして使用する、現在の AWS アカウントの Amazon SQS キューを選択

- 他の AWS アカウントの Amazon SQS キューをデッドレターキューとして指定

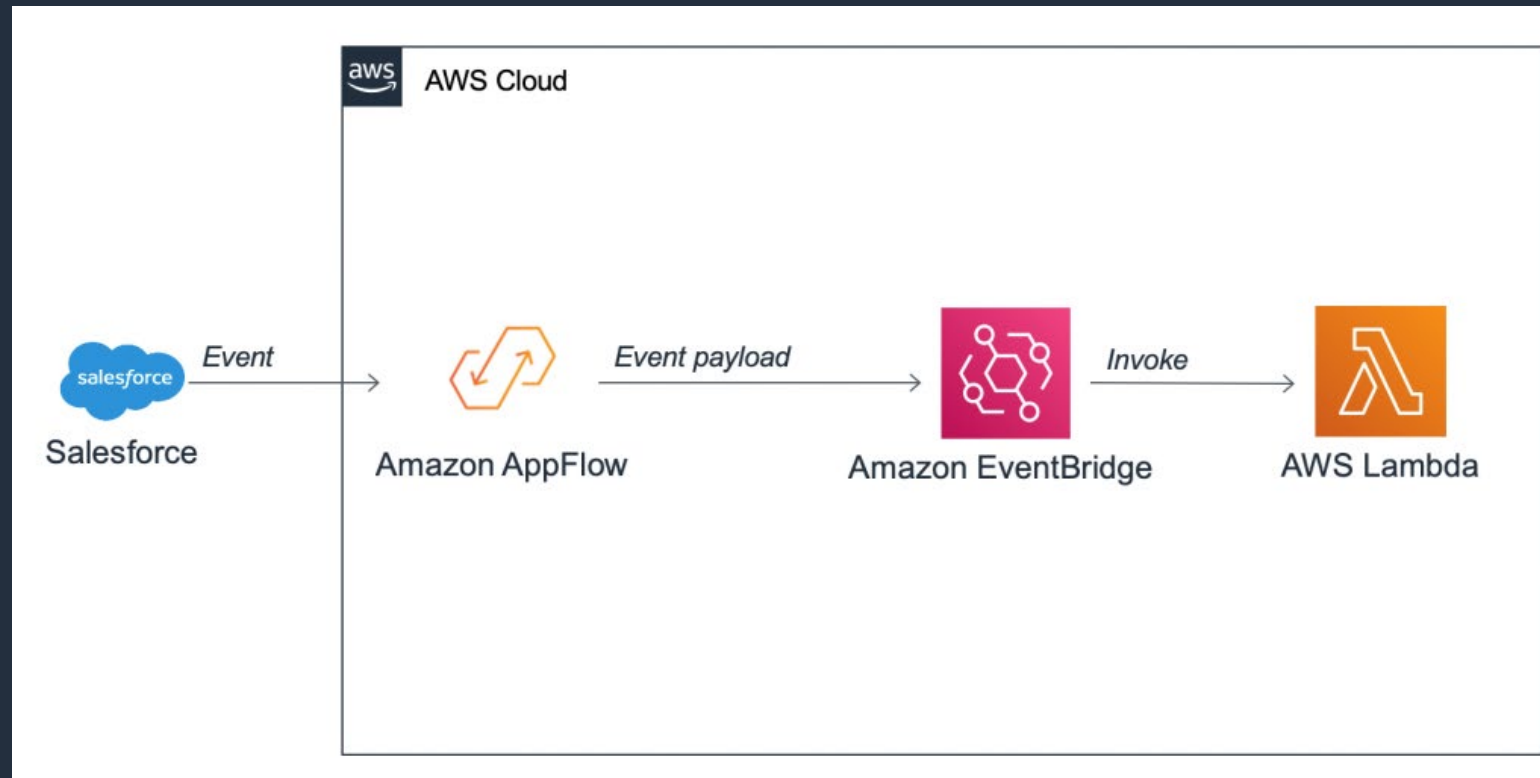
権限不足などはリトライなしで即時デッドレターキューに移動

標準キューが指定可能

# Salesforce統合



# Amazon EventBridgeとAmazon AppFlowによるSalesforce統合



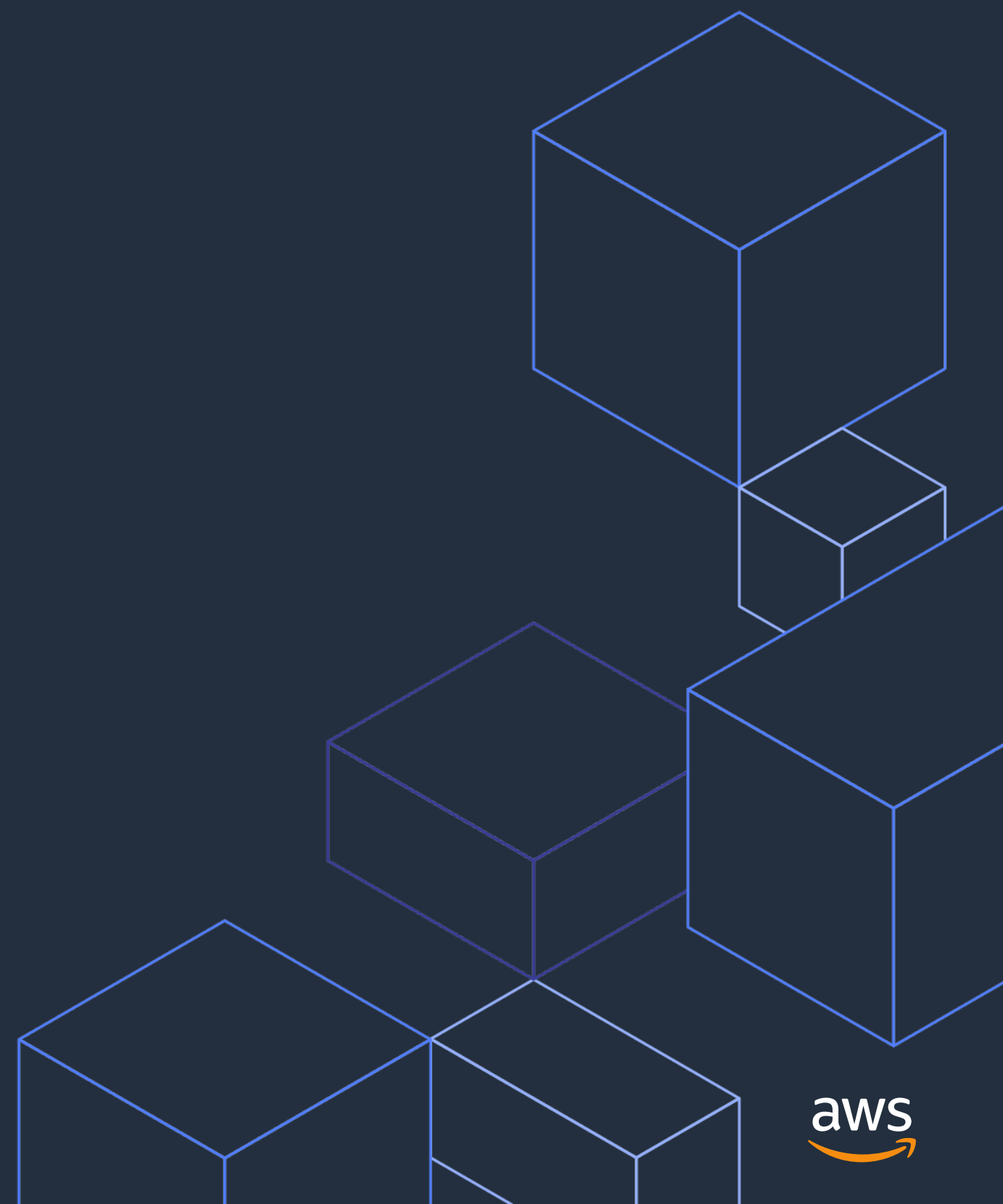
- Amazon EventBridgeとAmazon AppFlowの統合により、顧客はイベント駆動にSalesforceからイベントを受信して処理を記述可能

※Amazon AppFlowは、Salesforce、Marketo、Slack、ServiceNowなどのSaaSアプリケーションと、Amazon S3やAmazon RedshiftなどのAWSサービス間でデータを安全に転送できるようにするフルマネージドサービスです

<https://aws.amazon.com/jp/blogs/compute/building-salesforce-integrations-with-amazon-eventbridge/>



# まとめ



# まとめ

- Amazon EventBridgeと連携サービスを利用してイベントドリブンなシステムを信頼性高く構築することが可能
- 定義されたSchemaをコードの中で利用するためのサポートが充実しており、シンタックスエラーを抑えた高速な開発が可能
- イベントから自動的にスキーマを発見することにより煩わしいマニユアル定義による管理から解放される



**Thank you!**



**@\_kensh**