



# [AWS Black Belt Online Seminar]

## Amazon EventBridge

サービスカットシリーズ

Solutions Architect

野上 恭平

2020/1/22

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>



# 自己紹介

## 野上恭平

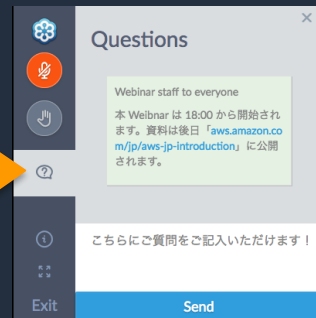
- 所属
  - アマゾン ウェブ サービス ジャパン株式会社  
技術統括本部
  - ソリューションアーキテクト
- 好きなサービス
  - AWS Lambda
  - IAM
  - CloudFormation



# 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1月22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おります。日本居住者のお客様が東京リージョンを使用する場合、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.

# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

## 3. クロスアカウント連携

## 4. 直近のアップデート

## 5. 料金、制限

## 6. まとめ

# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

## 3. クロスアカウント連携

## 4. 直近のアップデート

## 5. 料金、制限

## 6. まとめ

# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

## 3. クロスアカウント連携

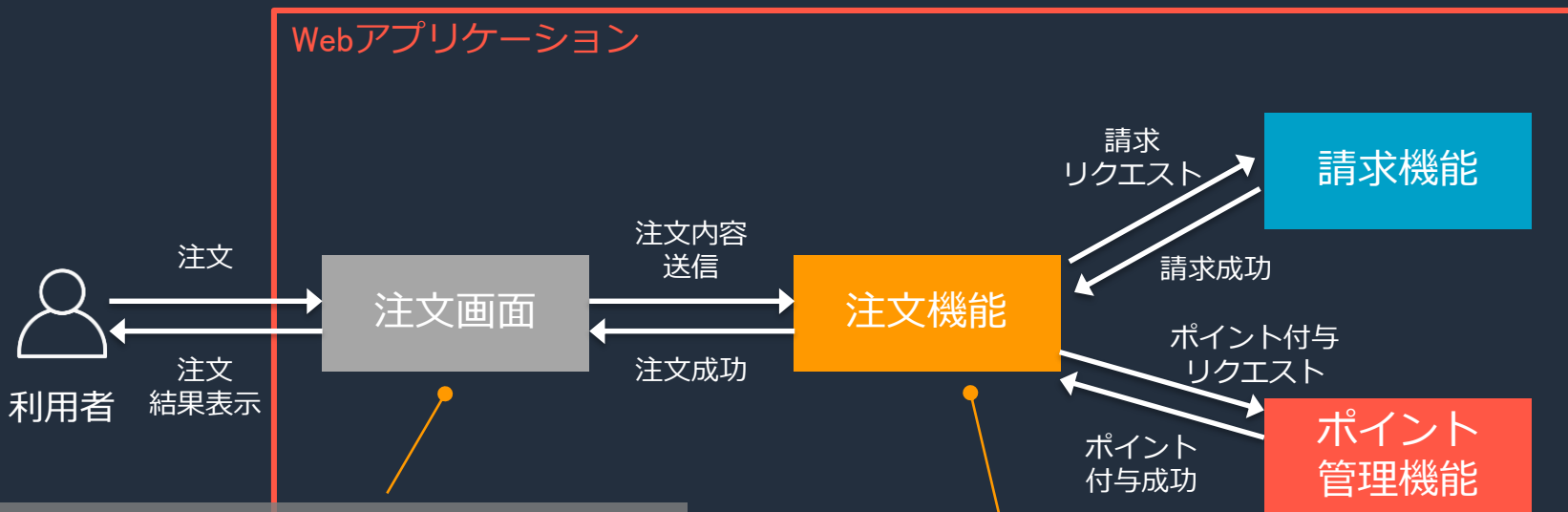
## 4. 直近のアップデート

## 5. 料金、制限

## 6. まとめ

# ECサイトの例

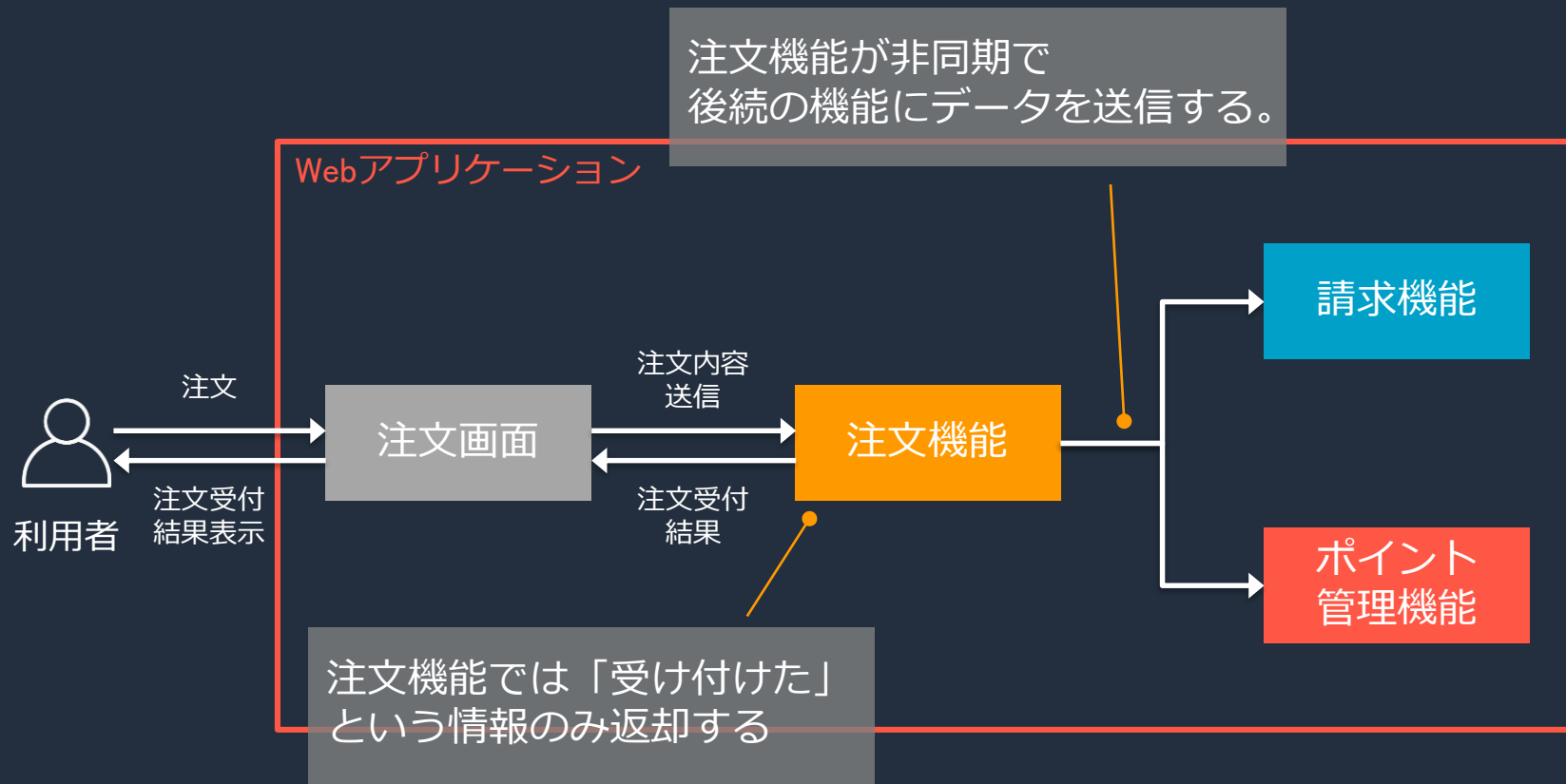
## イベント = 状態の変化



**同期処理**であるため、すべての処理が完了するまで利用者の画面が表示待ちになる

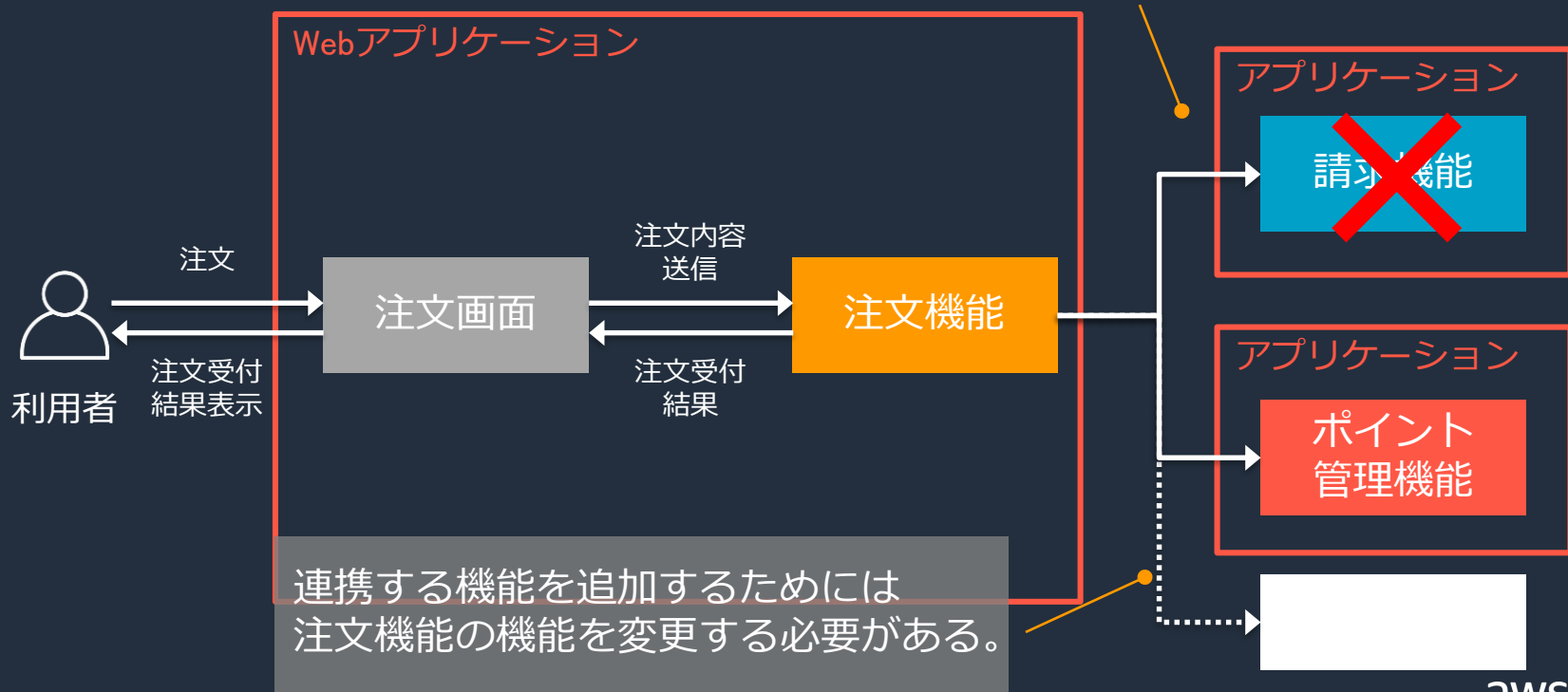
「注文の受付」と「請求」「ポイント付与」を**非同期処理**することで、表示待ちを改善

# ECサイトの例（非同期処理で実現）



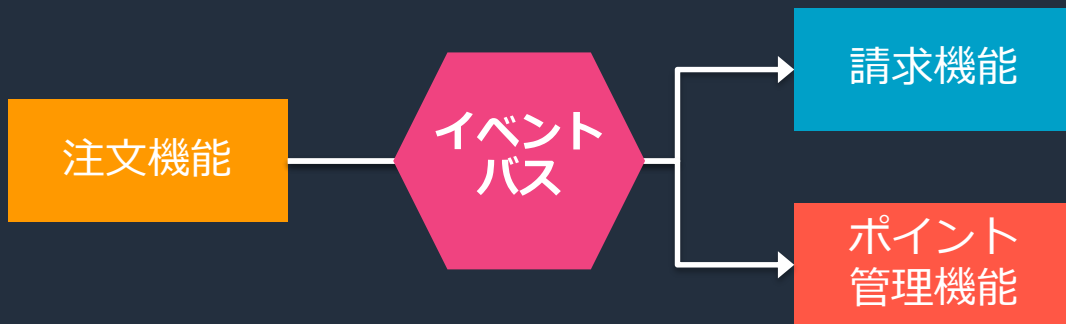
# ECサイトの例（アプリケーションを分割する）

多数の注文を受けると後続で処理できなくなり、処理に失敗する。



# イベントバスの導入

メッセージの送受信を管理するイベントバスを導入する。  
イベントバスは**Push型**で**非同期**で**一度に単一のメッセージを処理**する。  
送信者（Publisher）が受信者（Subscriber）を意識しない。





# Amazon EventBridge

AWSサービス、カスタムアプリケーション、  
SaaSアプリケーションのための  
サーバーレスイベントバスサービス

- 統合に必要な「point-to-point」の実装を取り除く
- シンプルなプログラミングモデルを提供する
- 多数のAWS サービスとSaaSアプリケーションを接続する
- フルマネージドでサーバーレスなイベントバス

# お客様へもたらす価値

イベント駆動  
アーキテクチャを  
容易に構築

イベントの送信元・送信先がお互いの実装を意識する必要が無い。

実装量の削減

データの取り込み  
フィルタ、配信が  
実装なしに実現可能

運用負荷の  
低減

イベント連携用の  
サーバーが不要。  
自動でスケールし、イ  
ベント数での従量課金

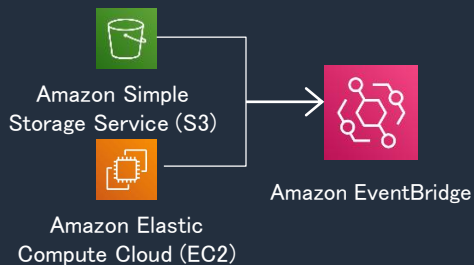
SaaS 由来の  
データの利用

AWS サービスや SaaSア  
プリケーションの  
イベントを利用して  
ワークフローを実行

# イベントバス

EventBridgeはイベントバスでイベントを受け取る。

デフォルト

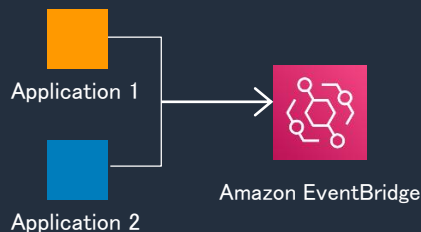


AWSサービス

例) EC2インスタンスが停止した

AWSが発行する  
システムのイベント

カスタム



独自のアプリケーション

例) 注文機能が注文を受け付けた

お客様/SaaSが発行する  
業務的なイベント

パートナー

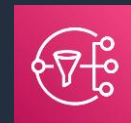
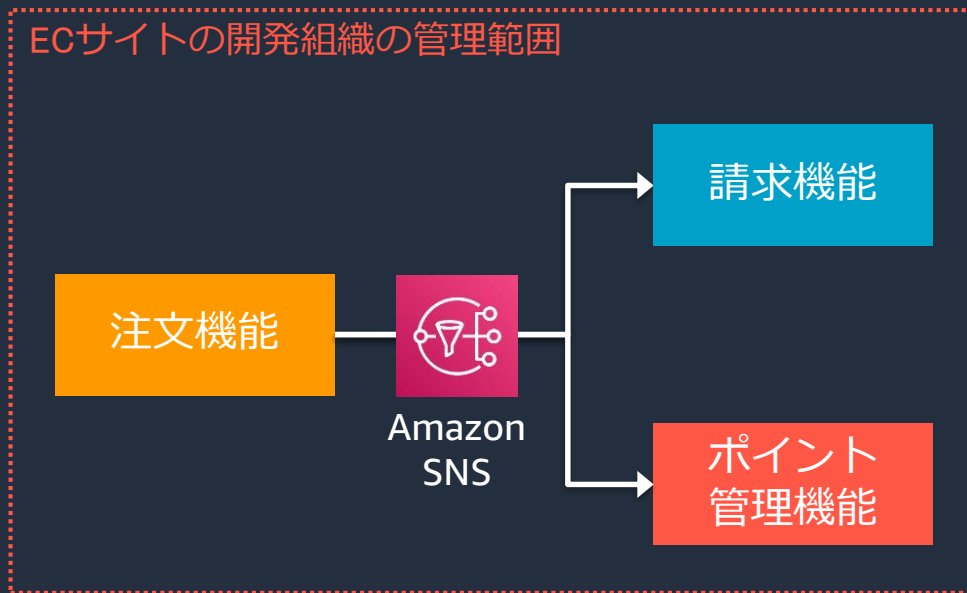


SaaS

例) SaaSのCRMで顧客情報が更新された

# 機能間の連携

機能間のデータ連携方法の一例として、Amazon Simple Notification Service(SNS)を利用する。



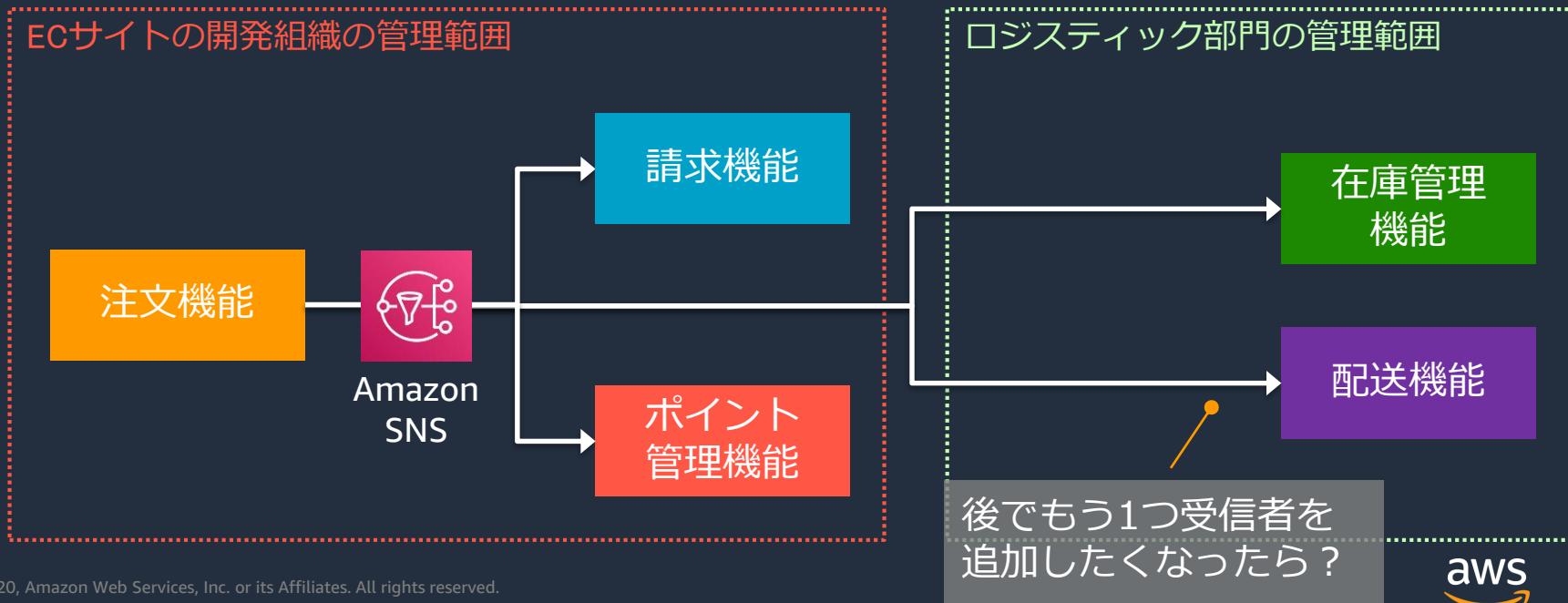
Amazon  
SNS

- Push型でメッセージを配信
- Lambda/SQS/HTTP(S)/Email への連携
- 数百万もの大量のイベントを低レイテンシーで配信可能

# 組織をまたがったイベント連携

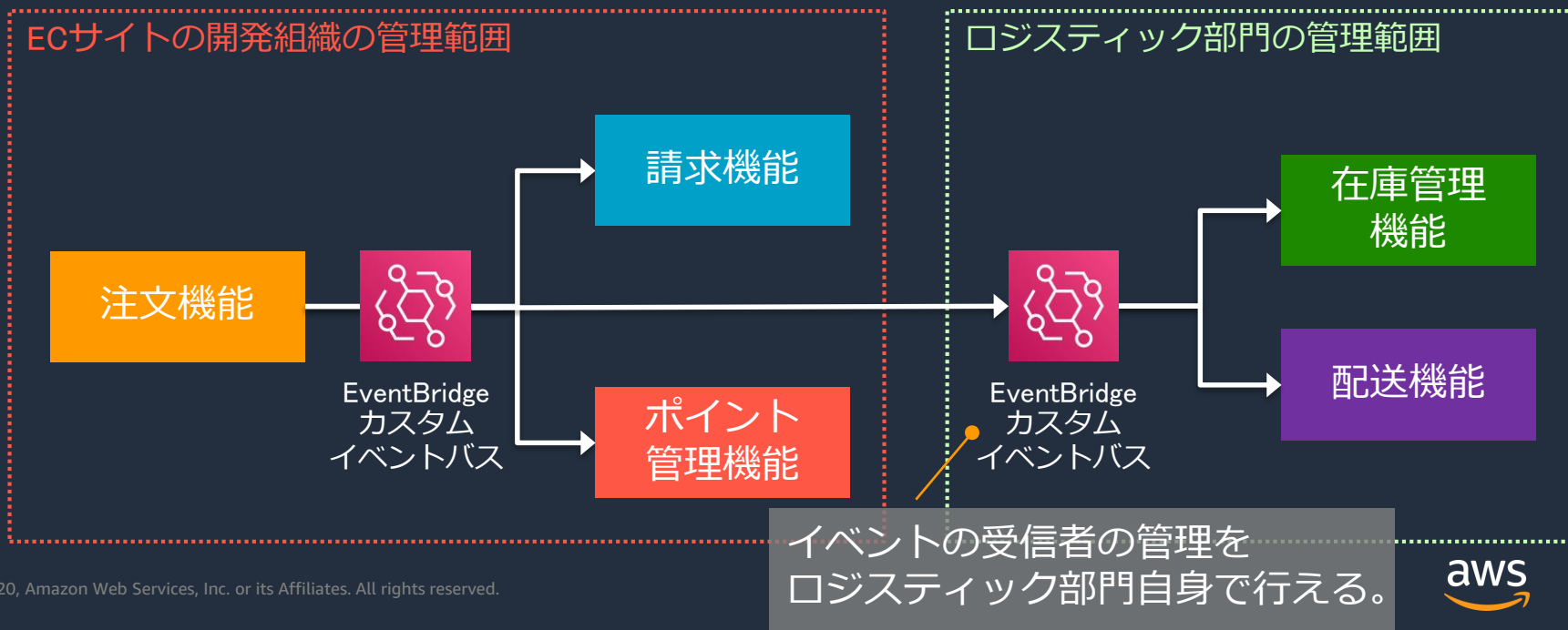
Q. ロジスティック部門から注文時のデータをリアルタイムに連携して欲しいと言われたら？

A. SNSの受信者に追加すれば実現可能。



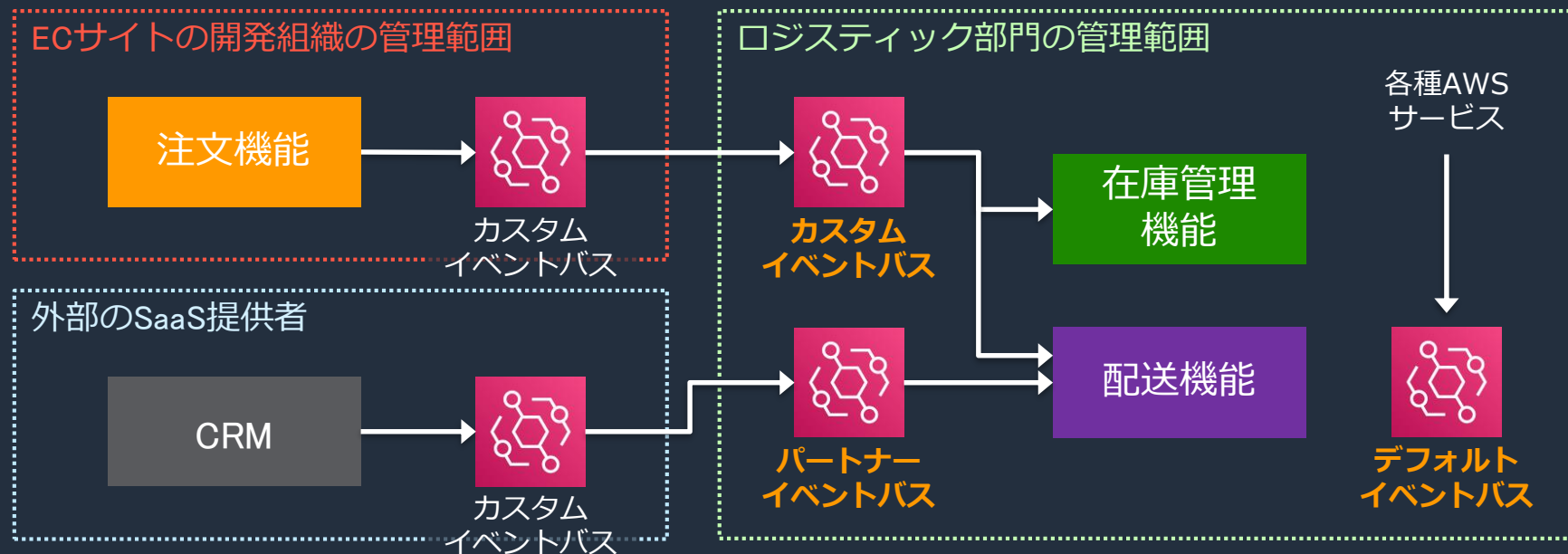
# 組織をまたがったイベント連携

イベントバスを利用して組織間の連携を行えば、他の組織でイベントの受信者が増えても送信側は意識する必要がない。



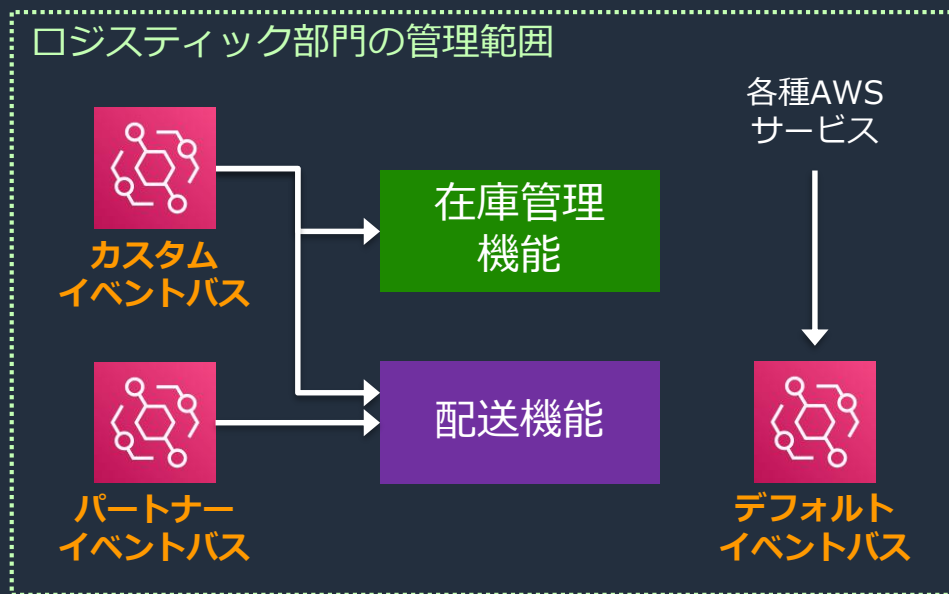
# イベントバスの使い分け

各AWSサービスからのイベントは**デフォルトイベントバス**、  
他のアプリケーションからのデータには**カスタムイベントバス**、  
SaaSからのデータには**パートナーイベントバス**を用いる



# EventBridgeとCloudWatch Eventsとの違い

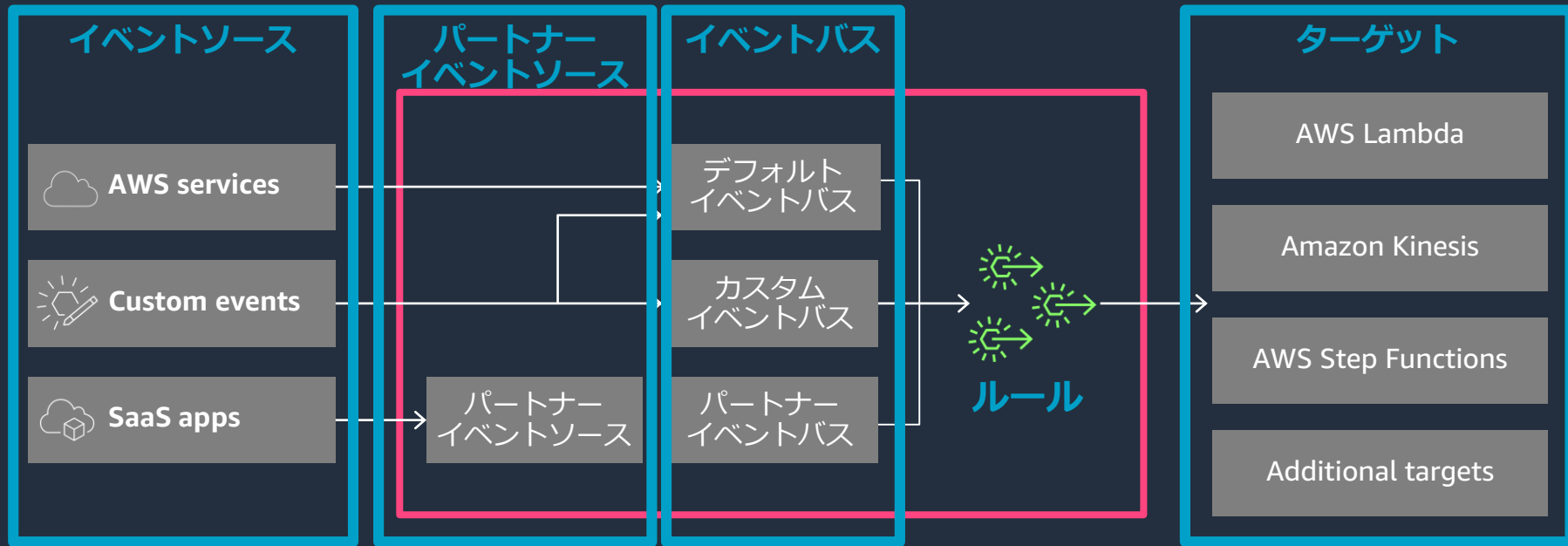
- EventBridgeは Amazon CloudWatch Eventsから派生・独立したサービスで、デフォルトイベントバスはその延長線上にある機能。
  - CloudWatch EventsのAPIやマネジメントコンソールはそのまま利用可能。
- カスタムイベントバスやSaaS連携の機能を追加し、アプリケーションのイベントをより活用していただくため、モニタリング視点のCloudWatchファミリーから独立した。



# 本日の内容

1. EventBridge とは？
  - イベントバス導入の背景
  - **アーキテクチャ**
  - 構成要素
  - アクセス制御
2. SaaS との連携
  - SaaSアプリケーションとの連携案
  - 対応しているSaaSアプリケーション
  - SaaSアプリケーションとの連携方法
  - EventBridge 活用事例
  - EventBridgeパートナーになるには
3. クロスアカウント連携
4. 直近のアップデート
5. 料金、制限
6. まとめ

# EventBridgeのアーキテクチャ



# EventBridgeのアーキテクチャ

イベントソースから  
イベントが送信される

## イベントソース

AWS services

Custom events

SaaS apps

デフォルト  
イベントバス

カスタム  
イベントバス

パートナー  
イベントソース

パートナー  
イベントバス



AWS Lambda

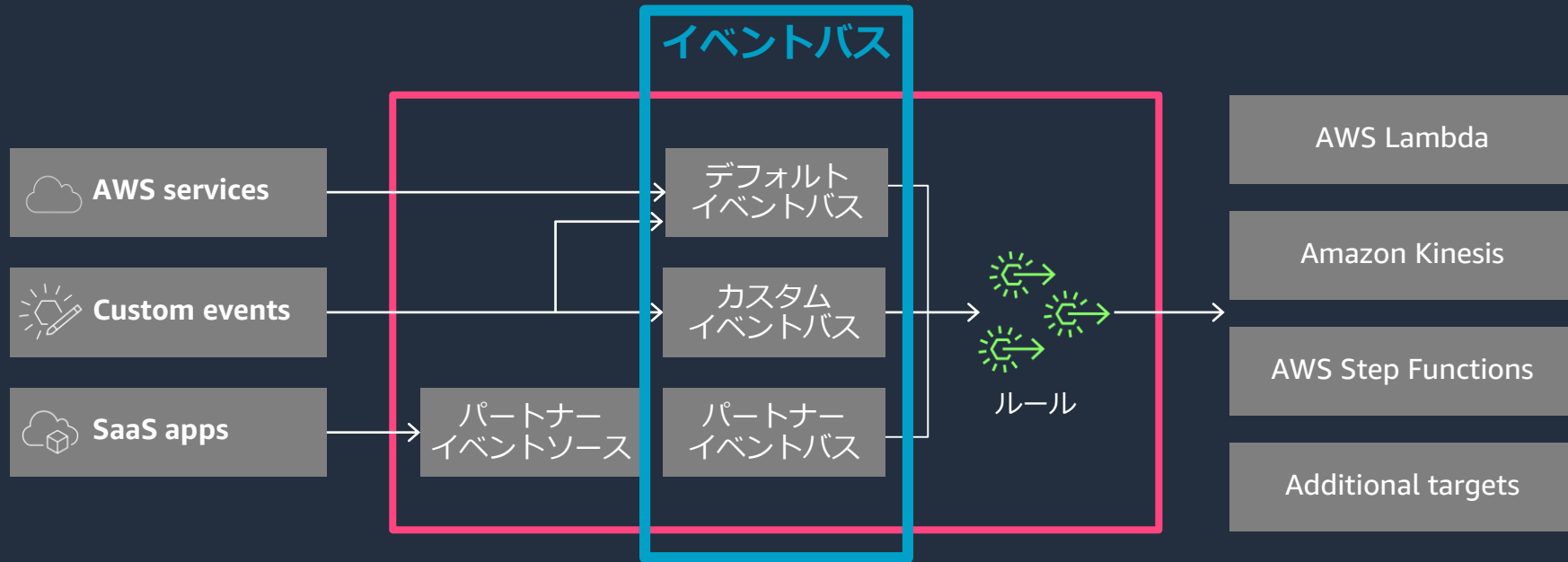
Amazon Kinesis

AWS Step Functions

Additional targets

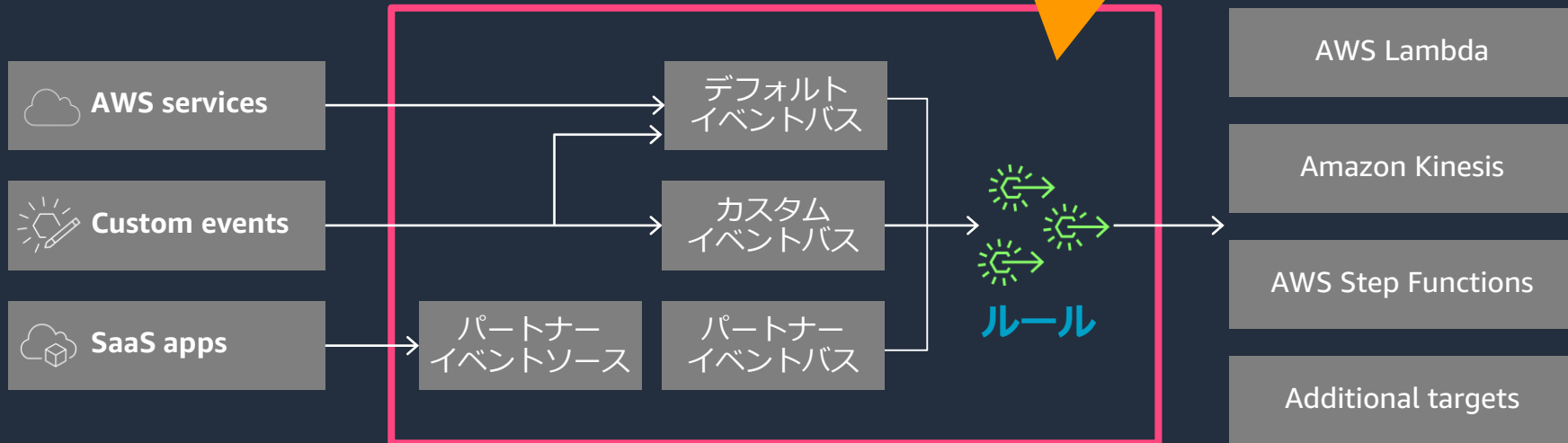
# EventBridgeのアーキテクチャ

イベントバスが  
イベントを受け付ける



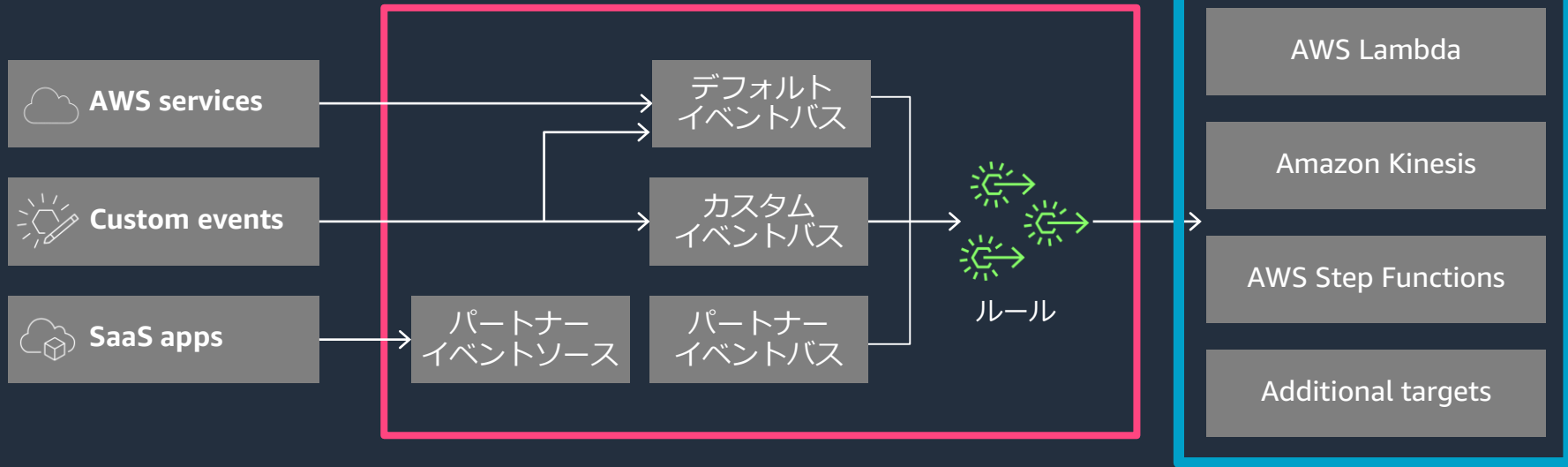
# EventBridgeのアーキテクチャ

後続処理に送信するイベントを選択する



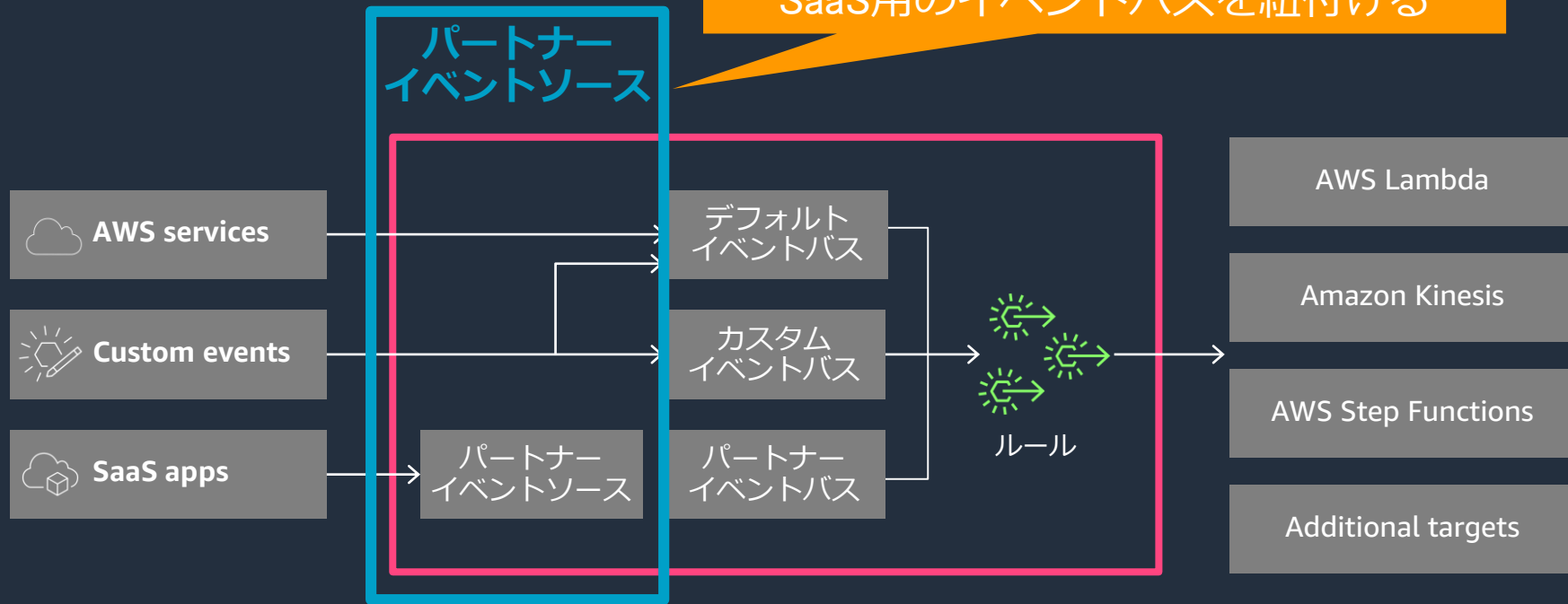
# EventBridgeのアーキテクチャ

ターゲットにイベントが送信され、  
各サービスでイベントが処理される



# EventBridgeのアーキテクチャ

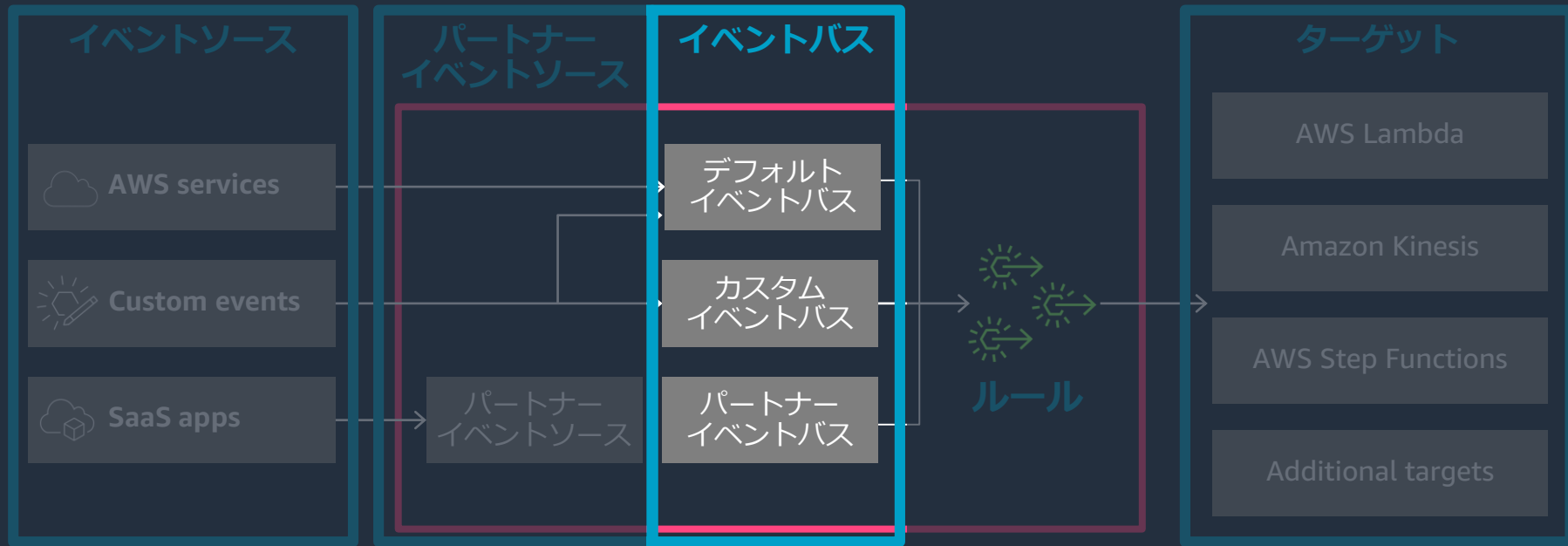
SaaSからのイベントを受信する場合は  
パートナーイベントソースと  
SaaS用のイベントバスを紐付ける



# 本日の内容

1. EventBridge とは？
  - イベントバス導入の背景
  - アーキテクチャ
  - **構成要素**
  - アクセス制御
2. SaaS との連携
  - SaaSアプリケーションとの連携案
  - 対応しているSaaSアプリケーション
  - SaaSアプリケーションとの連携方法
  - EventBridge 活用事例
  - EventBridgeパートナーになるには
3. クロスアカウント連携
4. 直近のアップデート
5. 料金、制限
6. まとめ

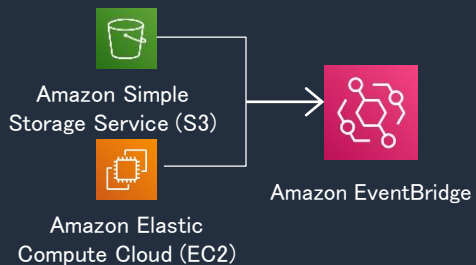
# EventBridgeのアーキテクチャ



# イベントバス

EventBridgeはイベントバスでイベントを受け取る。

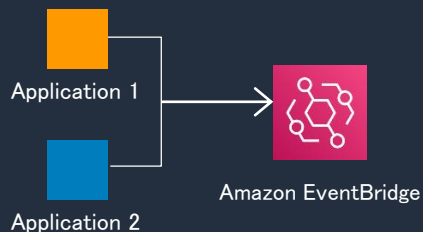
## デフォルト



### AWSサービス

例) EC2インスタンスが停止した

## カスタム



### 独自のアプリケーション

例) 注文機能が注文を受け付けた

## パートナー



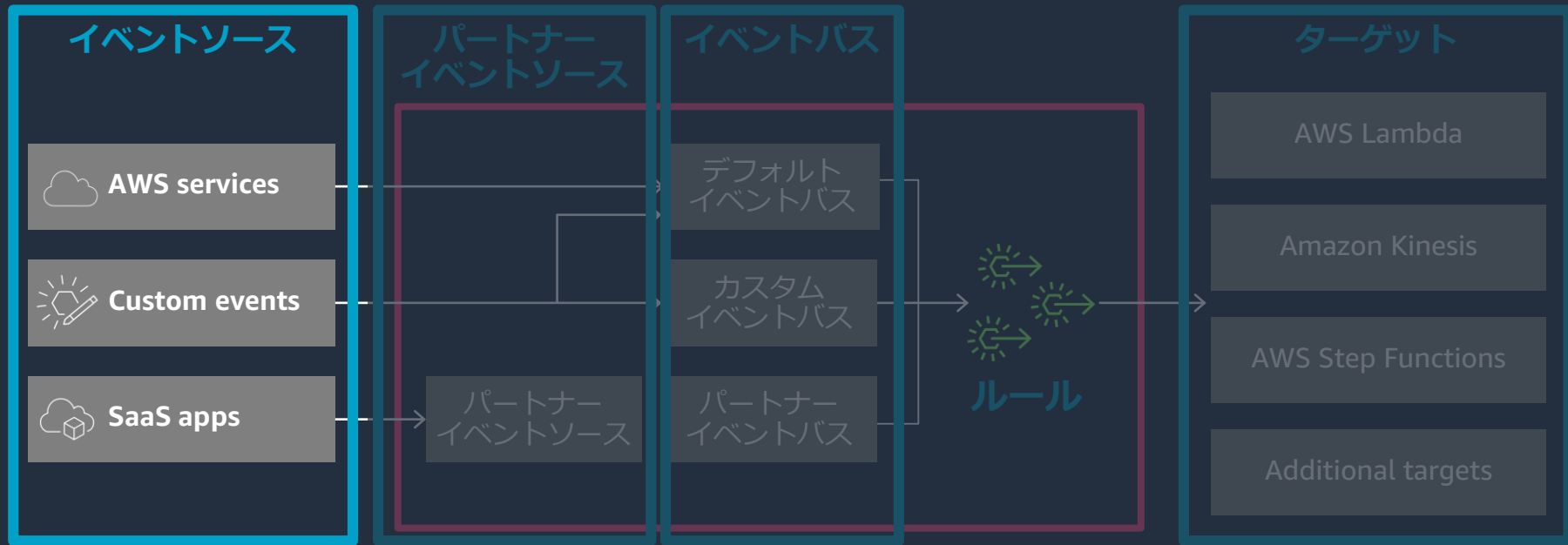
### SaaS

例) SaaSのCRMで顧客情報が更新された

AWSが発行する  
**システムのイベント**

お客様/SaaSが発行する  
**業務的なイベント**

# EventBridgeのアーキテクチャ



# イベント

- イベントは状態の変化を示し、JSONオブジェクトで表される。

	送信元	送信先のイベントバス	送信されるタイミング
AWSサービスのイベント	各AWSサービス	デフォルトのイベントバス	AWSサービスの状態が変わったとき(*1)
	各AWSサービス (CloudTrail経由)		AWSサービスのAPIを呼び出したとき(*2)
カスタムイベント	ユーザーまたはアプリケーション	送信時に指定する(*3)	EventBridgeのPutEvents APIを呼び出したとき
SaaSイベント	SaaSアプリケーション	各SaaSごとに作成されるパートナーイベントバス	送信元のSaaSアプリケーションによって異なる

(\*1) イベントが送信されるサービスは下記参照。

[https://docs.aws.amazon.com/ja\\_jp/eventbridge/latest/userguide/event-types.html](https://docs.aws.amazon.com/ja_jp/eventbridge/latest/userguide/event-types.html)

(\*2) API呼び出しとイベントバスへイベントが送信されるまでにタイムラグがある。

(\*3) 未指定の場合はデフォルトイベントバスに送信される

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



# AWSサービスが送信するイベント

- すべてデフォルトのイベントバスに送信される。
- EventBridgeで未サポートのサービスはAWSのリソースの操作ログを取得する AWS CloudTrail 経由でイベントを取得可能。

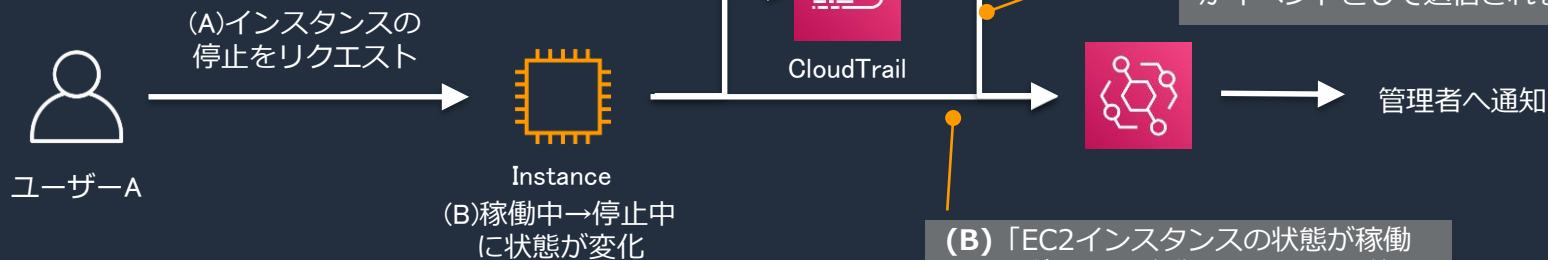
EventBridgeに直接送信されるイベント

AWSリソースの状態変化を起点にイベントが発生する。

CloudTrail経由で送信されるイベント

ユーザーのAWSリソースに対する操作 (=API呼び出し) を起点にイベントが発生する。

例) EC2インスタンスの停止を検出して管理者へ通知する例



# EventBridgeにイベントを送信するサービスと機能

- AWS Batch
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline
- AWS Config
- EBS
- Amazon EC2
  - Auto Scaling
  - spot instanceの中断
  - instanceの状態変更
- Amazon ECS
- Elemental MediaConvert
- Elemental MediaPackage
- Elemental MediaStore
- Amazon EMR
- Amazon GameLift
- AWS Glue
- AWS Ground Station
- Amazon GuardDuty
- AWS Health
- AWS KMS
- Amazon Macie
- AWS OpsWorks スタック
- Amazon SageMaker
- AWS Security Hub
- AWS Server Migration Service
- AWS Systems Manager
  - 設定コンプライアンス
  - メンテナンスウィンドウ
  - パラメータストア
- AWS Step Functions
- AWS リソースのタグ変更
- AWS Trusted Advisor
- Amazon WorkSpaces
- Amazon CloudWatch Alarm
- CloudTrail
- EventBridge
  - スケジュールされたイベント

# カスタムイベント

- ユーザーのPutEvents APIの呼び出しによって送信される。
  - CLIではaws events put-eventsコマンドで送信（下部記載）
  - SDKでは各言語の実装による。
- 送信先のイベントバスは送信時に選択可能。

```
$ cat entries.json
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ ¥"key1¥": ¥"value1¥" }",
    "DetailType": "myDetailType",
    "resources": [
      "resource1"
    ],
    "EventBusName": "myEventBus",
  }
]

$ aws events put-events --entries file:///entries.json
```

送信先のイベントバス名を指定

# SaaSイベント

- SaaSによるイベントは各SaaS専用のイベントバスに送信される。
- 送信されるイベントの種類やタイミングは、各SaaSアプリケーションの仕様による。

# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```



イベントの  
メタデータ



ペイロード

# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```

## 管理用データ

EventBridgeが自動設定。  
ユーザーによる変更不可。

# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```

## タイムスタンプ

明示的な指定が可能。  
省略時にはAPIの呼び出し時刻が付与される

# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```

## イベントの送信元

送信元のアプリケーション/サービスを表す。  
AWSサービスのイベントの場合、  
aws.ec2のようなサービス名が入る。

"aws."のprefixはAWSによって予約されているため使用不可。

# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```

## イベントの種類

イベントの種類を表す。

sourceとdetail-typeの組み合わせでdetailのスキーマが決定する。

# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```

## リソース

イベントが関連するリソース。  
AWSサービスが送信するイベントの場合、  
ARN(Amazon Resource Name)の配列が  
含まれる。

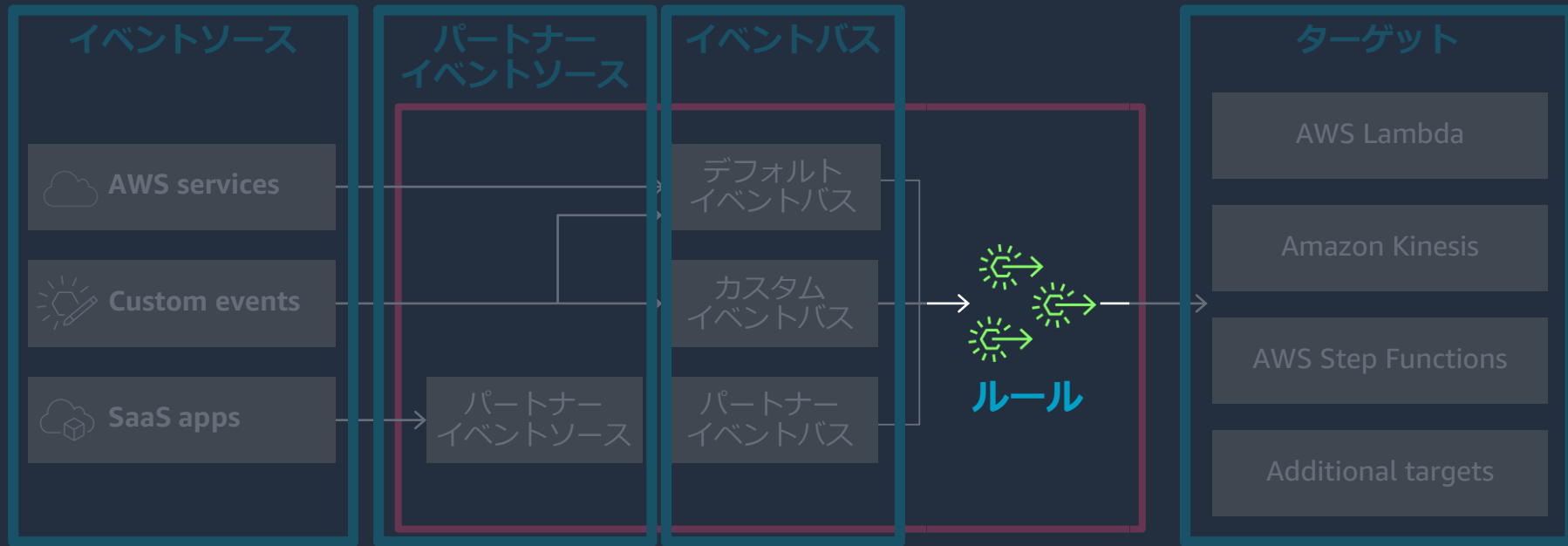
# イベントの構造

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-c4a0-74f106398c4e",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "order-service",  
  "detail-type": "New Order",  
  "resources": [],  
  "detail": {  
    "orderId": "cfb2ae566f9b",  
    "customerId": "C12345",  
    ...  
  }  
}
```

## 詳細

イベントの詳細、内容を表す。

# EventBridgeのアーキテクチャ



# ルール

- イベントバスで受信したイベントのうちどのイベントをターゲットに送信するかを定義する規則。

どんなイベントを送信するか？

- イベントパターンによる定義
- スケジュール式による定義

どのイベントバスを対象とするか？

どのターゲットに送信するか？

**パターンを定義**

イベントパターンを構築またはカスタマイズするか、スケジュールを設定してターゲットを呼び出します。

イベントパターン 情報  
イベントに一致するパターンを作成

スケジュール 情報  
スケジュールに基づいてターゲットを呼び出す

**イベントバスを選択**

このルールのイベントバスを選択します。

AWS のデフォルトのイベントバス

カスタムイベントバスまたはパーティイベントバス

選択したイベントバスでルールを有効にする

**ターゲットを選択**

イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに (1 ルール当たり 5 個のターゲットに制限) 呼び出すターゲットを選択します。

ターゲット 削除

イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに (1 ルール当たり 5 個のターゲットに制限) 呼び出すターゲットを選択します。

Lambda 関数

機能

関数を選択

▶ バージョン/エイリアスを設定

▶ 入力の設定

ターゲットの追加

# ルールのイベントパターン

- イベント同様のJSONオブジェクト形式で記述する。
- AWSサービスのイベントで代表的なものは定義済み。マネジメントコンソールから選択するだけで利用可能。
- テスト用APIを提供しており、CLIでイベントパターンのテストが可能。

```
$ aws events test-event-pattern ¥  
  --event <イベントのJSON文字列> ¥  
  --event-pattern <イベントパターンのJSON文字列>
```

- イベントパターンがイベントとマッチする条件
  - パターンに指定されているすべてのフィールド名を含む。
  - 入れ子構造を含む場合は、同一階層のすべてのフィールド名を含む。
  - パターンで指定されていないフィールドは無視される。
  - 値は配列形式で記述し、配列の要素のいずれかと一致する。
  - マッチングは大文字小文字を区別して行われる。
  - 数字は文字列表現レベルでマッチングされる。(300, 300.0, 3.0e2は全て区別)

# イベントパターンがイベントとマッチする例

```
{  
  "version": "0",  
  "id": "adeacade-c34c-ce58-(trim)",  
  "account": "123456789012",  
  "region": "us-east-1",  
  "time": "2019-12-02T21:46:19Z",  
  "source": "aws.ecr",  
  "detail-type": "ECR Image Action",  
  "resources": [],  
  "detail": {  
    "result": "SUCCESS",  
    "repository-name": "app",  
    "image-digest": "sha256:(trim)",  
    "action-type": "PUSH",  
    "image-tag": "latest"  
  }  
}
```

```
{  
  "source": ["aws.ecr"],  
  "detail-type": ["ECR Image Action"],  
  "detail": {  
    "result": ["SUCCESS"],  
    "repository-name": ["app"],  
    "action-type": ["PUSH"],  
    "image-tag": ["latest"]  
  }  
}
```

イベントパターンのすべてのフィールド名を含む

入れ子構造を含む場合、同一階層のフィールド名を含む

値は配列形式で記述し、配列の要素のいずれかと一致する。  
大文字小文字も含めてフィールドの値が一致する。

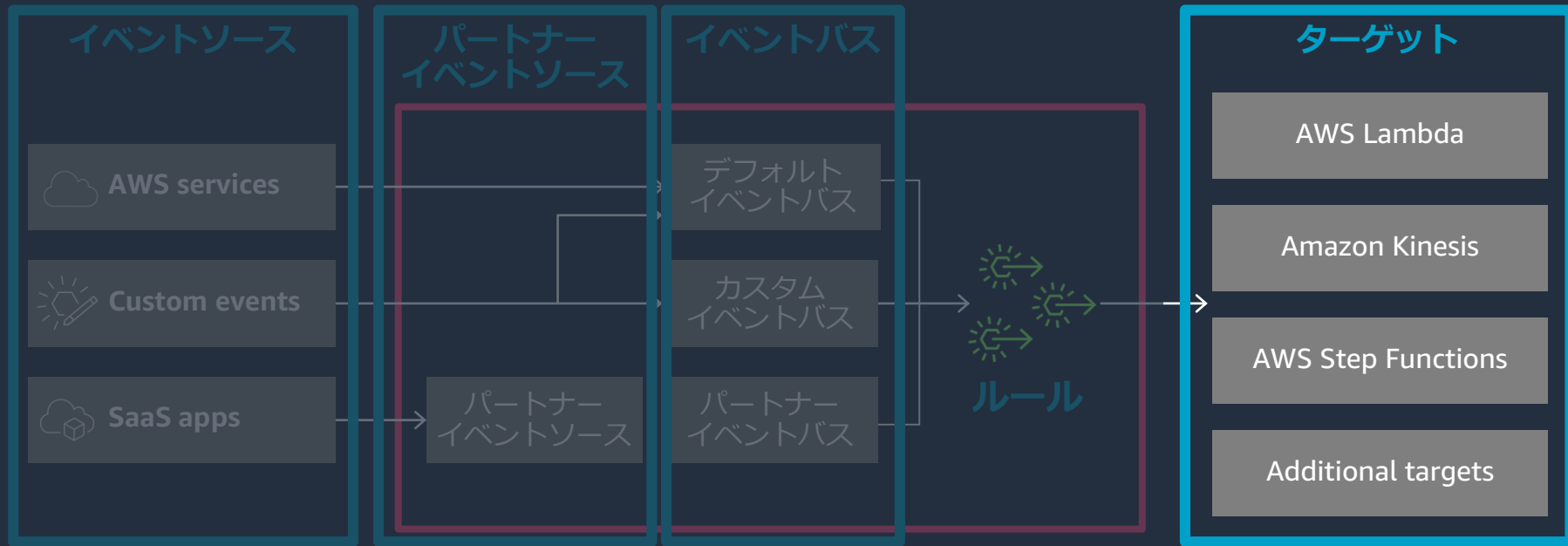
# ルールのスケジュール形式

スケジュールに基づいて自己をトリガーするイベントを作成できる。  
注意：**タイムゾーンはUTC**で、**最小精度は分**での実行が可能。

	cron式	rate式
説明	cron形式で記述する。	実行頻度を指定する。
設定値の形式	cron([分] [時間] [日] [月] [曜日] [年])	rate([値] [単位])
設定例	毎日午後1時(UTC)に実行する cron(0 13 * * ? *)	5分ごとに実行する rate(5 minutes)

13:00:00 から  
13:01:00 までに実行される

# EventBridgeのアーキテクチャ



# ターゲット

- イベントの送信先となり、イベントを処理するもの。
- 1つのルールあたり5つまでのターゲットに送信可能。
  - ターゲットのリソースは同じリージョンに存在する必要がある。
- 送信先のサービスによってパラメータが異なる。
- 一部のサービスでは受信したイベントのJSONを抽出したり、メッセージを変換して送信可能。

### ターゲットを選択

イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに (1 ルール当たり 5 個のターゲットに制限) 呼び出すターゲットを選択します。

ターゲット 削除  
イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに (1 ルール当たり 5 個のターゲットに制限) 呼び出すターゲットを選択します。

Lambda 関数 ▼

機能  
関数を選択 ▼

▶ バージョン/エイリアスを設定

▼ 入力の設定

一致したイベント [情報](#)

一致したイベントの一部 [情報](#)

定数 (JSON テキスト) [情報](#)

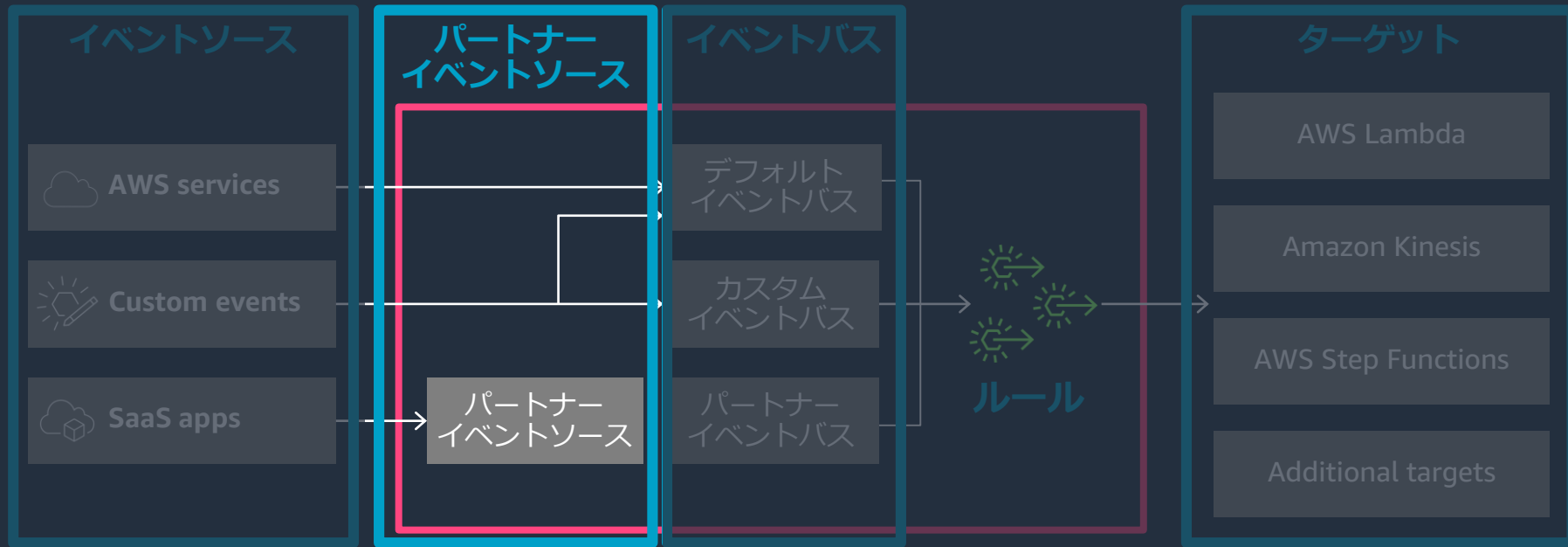
入力トランスフォーマー [情報](#)

ターゲットの追加

# ターゲットとして指定できるもの

- Lambda関数
- EC2インスタンス
- Kinesis Data Streamsのストリーム
- Kinesis Data Firehoseの配信ストリーム
- CloudWatch Logsのロググループ
- ECSタスク
- Systems Manager Run Command
- Systems Manager Automation
- AWS Batchジョブ
- AWS Step Functionsステートマシン
- AWS CodePipelineのパイプライン
- AWS CodeBuildプロジェクト
- Inspector の評価テンプレート
- SNS トピック
- SQS キュー
- 組み込みターゲット
  - EC2 CreateSnapshot API call
  - EC2 RebootInstances API call
  - EC2 StopInstances API call
  - EC2 TerminateInstances API call
- 別のAWS アカウントのイベントバス  
(同一アカウント間の送信やイベントの折返しは不可)

# EventBridgeのアーキテクチャ



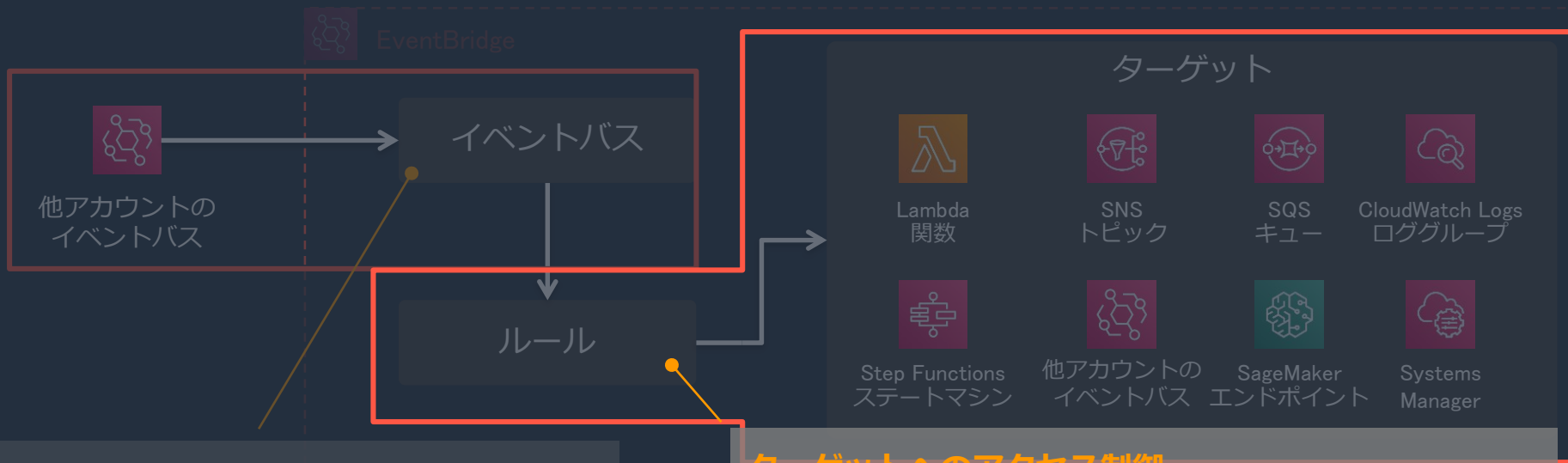
# パートナーイベントソース

- SaaSアプリケーションからのイベント送信を受け付けるためのリソース。
- 各SaaSアプリケーションの管理画面から作成でき、イベントバスにひもづけることで、イベントの連携が開始する。

# 本日の内容

1. EventBridge とは？
  - イベントバス導入の背景
  - アーキテクチャ
  - 構成要素
  - **アクセス制御**
2. SaaS との連携
  - SaaSアプリケーションとの連携案
  - 対応しているSaaSアプリケーション
  - SaaSアプリケーションとの連携方法
  - EventBridge 活用事例
  - EventBridgeパートナーになるには
3. クロスアカウント連携
4. 直近のアップデート
5. 料金、制限
6. まとめ

# アクセス制御の違い



## イベントバスのアクセス制御

他のアカウント、または、AWS Organizations における同一の組織からのイベント送信をイベントバスのポリシーで許可する。

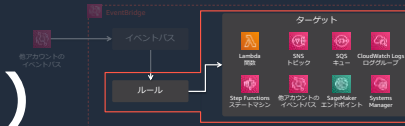
## ターゲットへのアクセス制御

- AWS Lambda, Amazon SNS, Amazon Simple Queue Service(SQS), Amazon CloudWatch Logsはの4つのサービスのリソースは、リソースのポリシーで許可する。
- それ以外のサービスはルールにIAMロールを割り当て、IAM ポリシーで許可する。



## ターゲットへのアクセス制御

# リソースベースポリシー (=ターゲット側で制御)



- LambdaはLambda関数ポリシー、SNSはトピックポリシー、SQSはキューポリシーにアクセス許可を記述する。
- CloudWatch LogsはEventBridgeからのイベント送信がデフォルトで許可されているため利用者による設定は不要
- マネジメントコンソール上でターゲットを追加した場合は自動で権限が付与される。

例) Lambdaのマネジメントコンソール  
正しく権限が付与されているときにサービスやルールが表示される。

▼ Designer

hello-world  
Layers (0)

CloudWatch Events ×

+ トリガーを追加

+ Add destination

CloudWatch Events

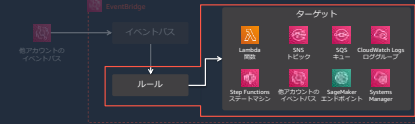
10minutes  有効 削除

arn:aws:events:ap-northeast-1:██████████:rule/10minutes  
イベントバス: default スケジュール式: rate(1 minute)



## ターゲットへのアクセス制御

# IAMポリシー (= IAM側で制御)



- Lambda, SNS, SQS, CloudWatch Logs以外のサービスをターゲットに指定する場合、IAMロールを割り当ててIAMポリシーで許可する。
- マネジメントコンソールからターゲットの追加するときには、必要なサービスの権限を持つIAMロールとIAMポリシーを作成可能。

### ターゲットを選択

イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに (1 ルール当たり 5 個のターゲットに制限) 呼び出すターゲットを選択します。

ターゲット 削除  
イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに (1 ルール当たり 5 個のターゲットに制限) 呼び出すターゲットを選択します。

別の AWS アカウントのイベントバス ▼

イベントバス  
イベントバス ARN を入力する

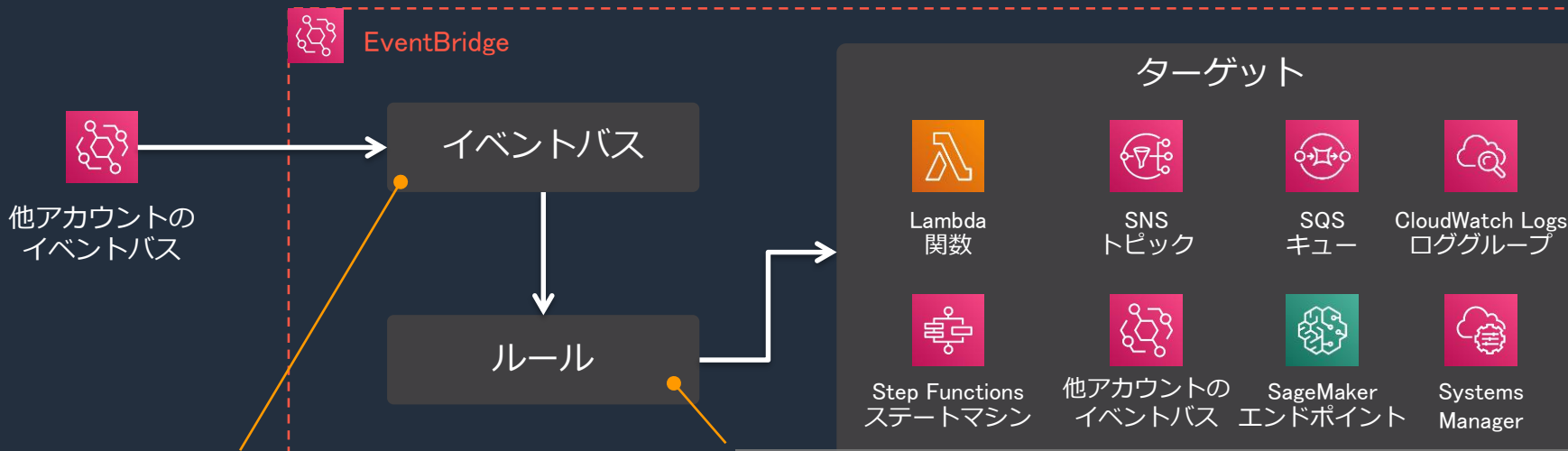
EventBridge には、上記の AWS アカウントのデフォルトイベントバスにイベントを送信するためのアクセス許可が必要です。続行すると、許可したことになります。

この特定のリソースに対して新しいロールを作成する

Amazon\_EventBridge\_Invoke\_Event\_Bus\_848173242

既存のロールの使用

# アクセス制御の違い



## イベントバスのアクセス制御

他のアカウント、または、AWS Organizations における同一の組織からのイベント送信をイベントバスのポリシーで許可する。

## ターゲットへのアクセス制御

- AWS Lambda, Amazon SNS, Amazon Simple Queue Service(SQS), Amazon CloudWatch Logsはの4つのサービスのリソースは、リソースのポリシーで許可する。
- それ以外のサービスはルールにIAMロールを割り当て、IAM ポリシーで許可する。

# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

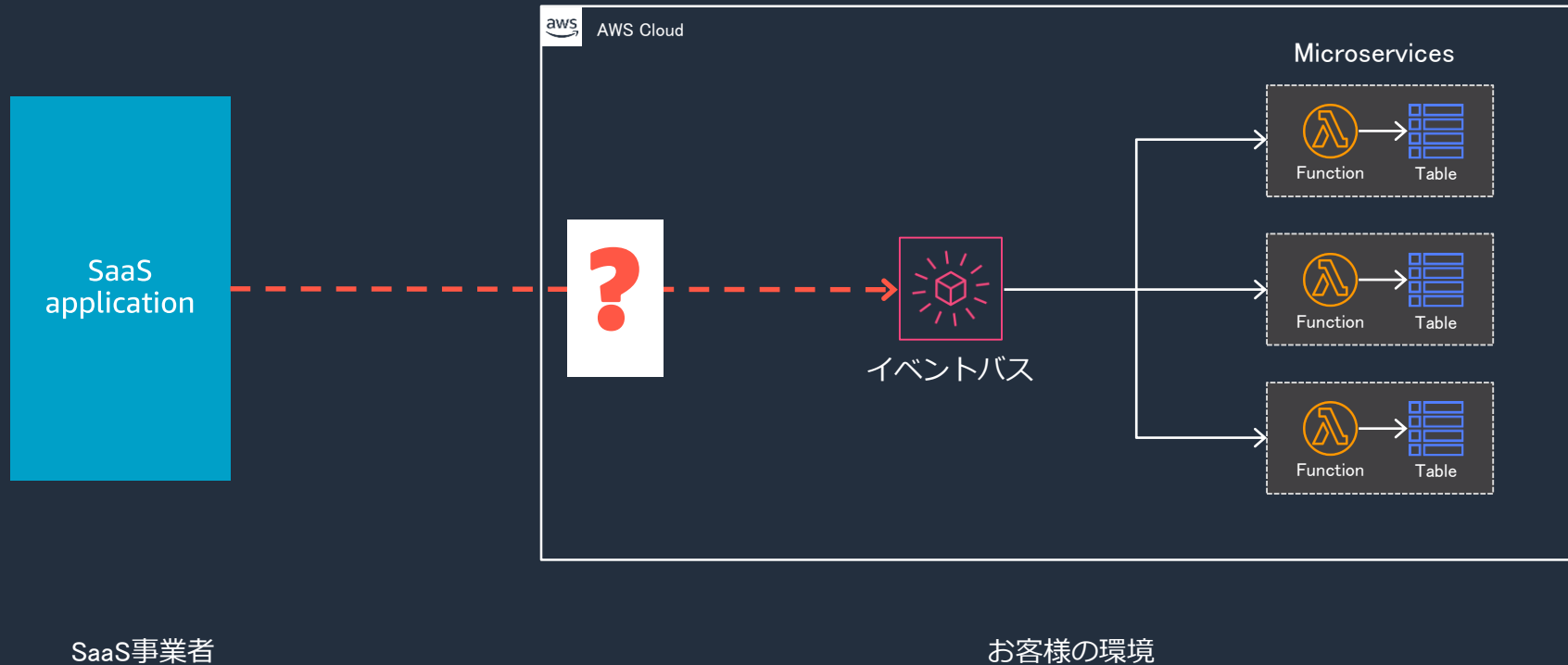
## 3. クロスアカウント連携

## 4. 直近のアップデート

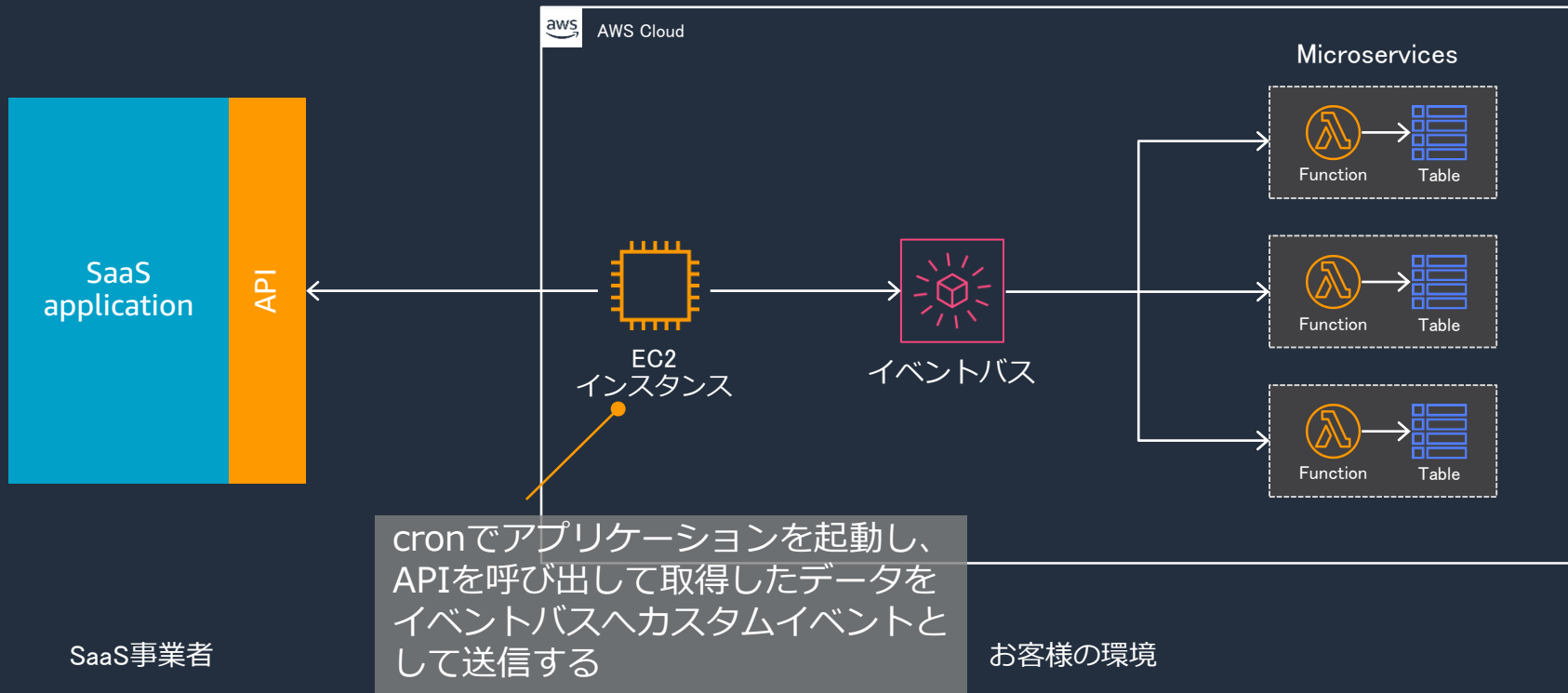
## 5. 料金、制限

## 6. まとめ

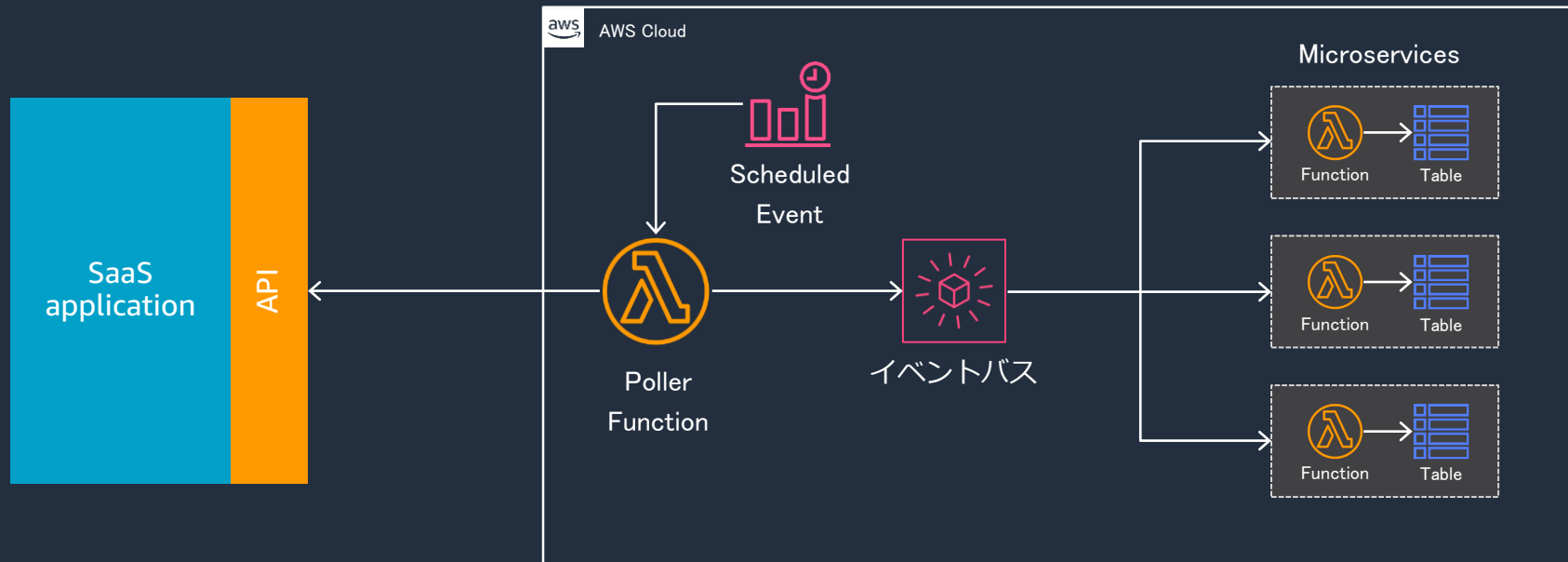
# SaaSアプリケーションとAWSを連携するには？



# 案1：SaaSのAPIをポーリングする（EC2）



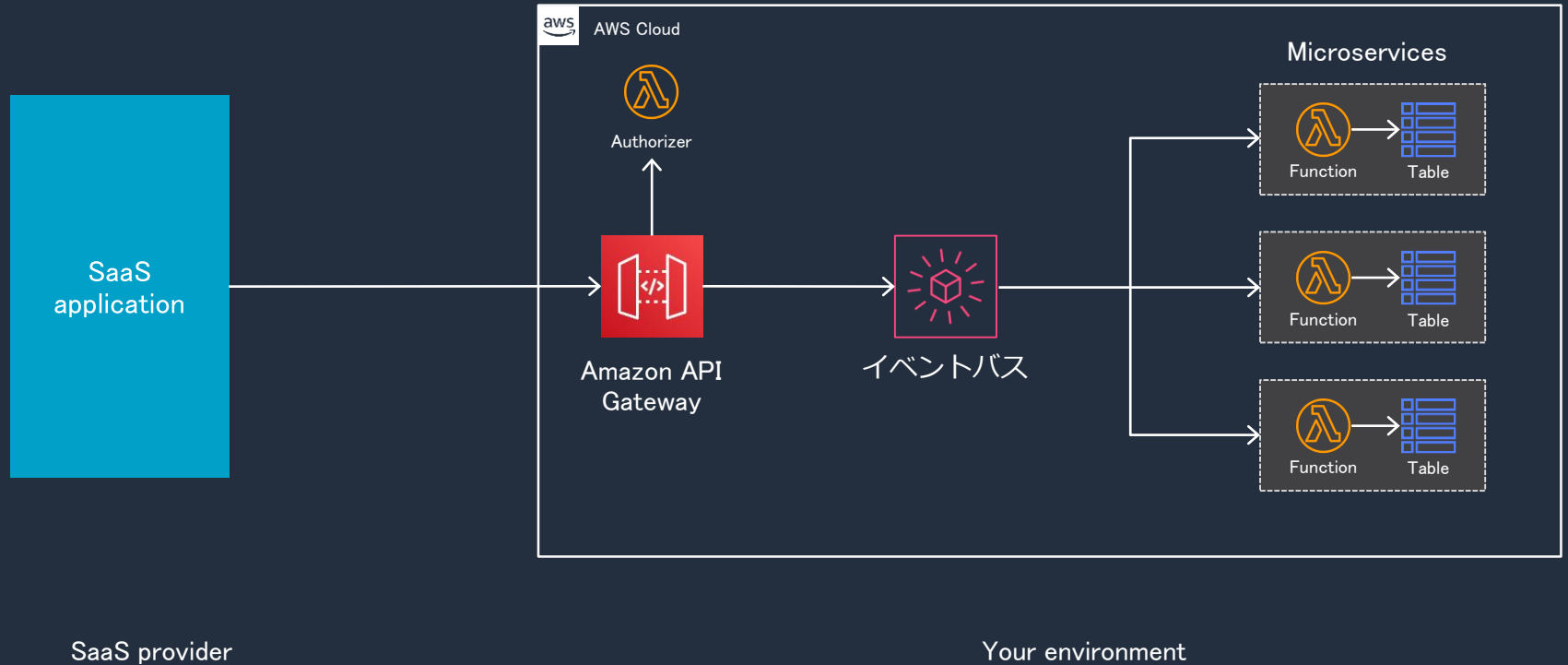
# 案2：SaaSのAPIをポーリングする（Lambda）



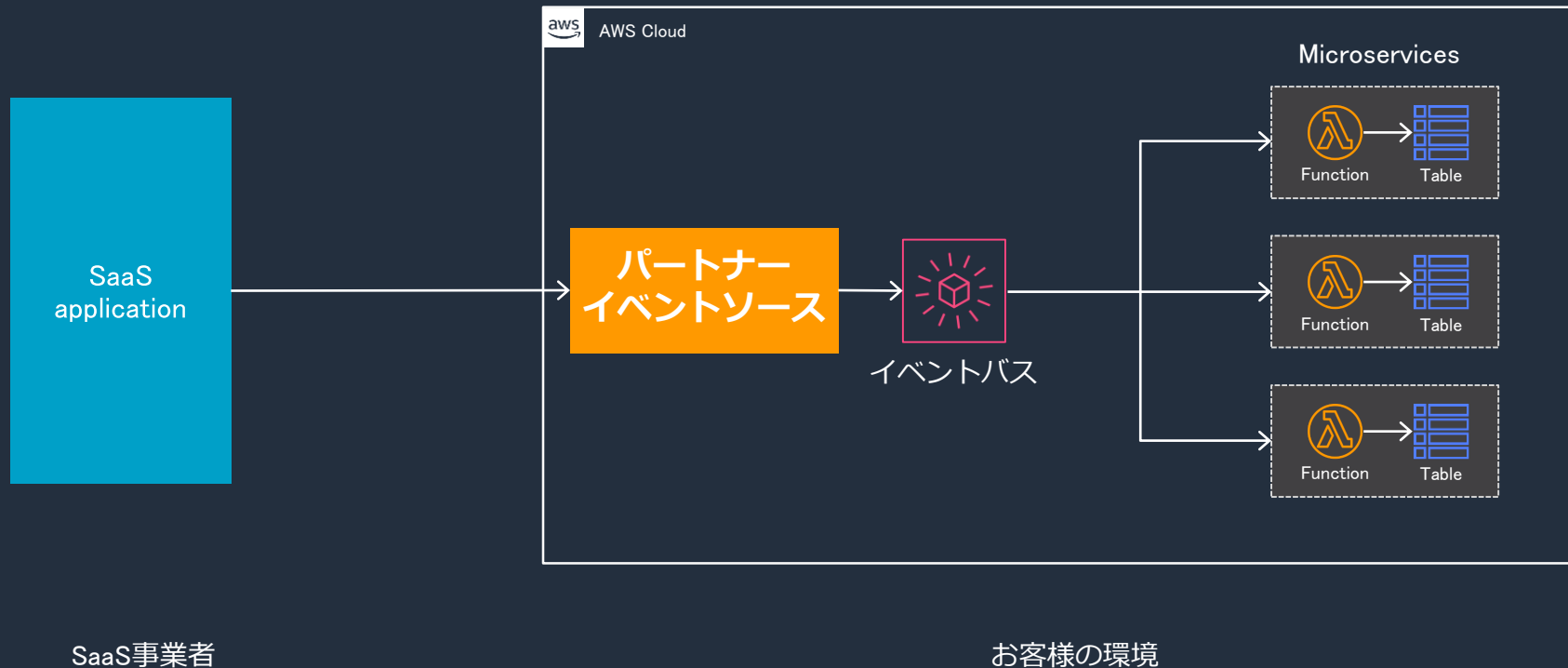
SaaS事業者

お客様の環境

# 案3 : SaaSのWebhook機能でPushしてもらおう



# 案4 : EventBridgeのSaaS連携機能の利用



# 対応しているSaaSアプリケーション

- 24のSaaSサービスに対応（2020年1月22日Webinar実施時点）



PagerDuty



whispir



SignalFx



onelogin



CleverTap



# 日本でも始まる、EventBridge 対応パートナー

NEW



ウェブサイトやモバイルアプリを利用する個々の顧客にあわせた体験を提供する CX プラットフォーム。

EventBridge 統合機能を使用することで、顧客行動イベントを AWS のサービスに送信して、BI、分析、機械学習などに活用可能。



運用中のクラウドやオンプレミスのサーバーにエージェントを1つ入れるだけで、簡単にサーバー管理できる運用監視サービス。

Mackerel によるモニタリングで検出したイベントをトリガーに EventBridge を経由して AWS 上でさまざまなアクションを実行。



Game Server Services

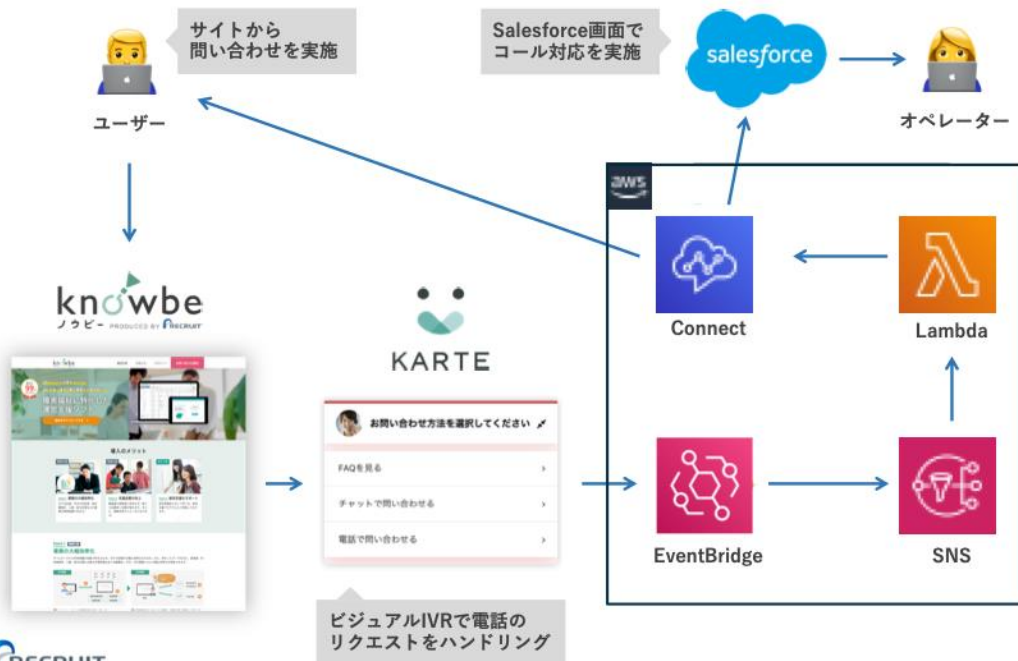
初期費用・運用費用なしで、サーバーアクセス 1回あたり 0.02円  
で使用できるゲームサーバー機能を提供。

EventBridge を使うことで、ゲームイベント（クエストのクリア/  
レベルアップ）と連携したアクションを追加拡張できます。

# EventBridge 活用事例

## Amazon Connect × KARTEで次世代CSプラットフォームを構築へ

### インバンドでの問い合わせ対応



### CX（顧客体験）の向上

従来は分断されていた電話対応をサイトに組み込むことで、問い合わせをサイトに一元化し、顧客の煩わしさを抑制することで、CXを向上する。

### シームレスなサービス連携

マルチクラウドをシームレスに連携することで、サービスのユーザーや社内の担当者が意識することなく、プラットフォームを利用できる。

### 拡張性の高いシステム

各クラウドサービスの機能やプラットフォームから取得されるデータを組み合わせることで、多様な要件に対応することができます。

# SaaSアプリケーションとの連携方法

## 1. パートナーイベントソースの作成

SaaSアプリケーション側の管理画面からAWSアカウントIDとリージョンを指定し、AWS側にパートナーイベントソースを作成する。

## 2. パートナーイベントソースとイベントバスの関連付け

AWSアカウント内に作成されたパートナーイベントソースをパートナーイベントバスをひもづける。

## 3. イベントのルールを作成する

通常のイベント同様ルールを作成する。

# SaaSアプリケーションとの連携方法

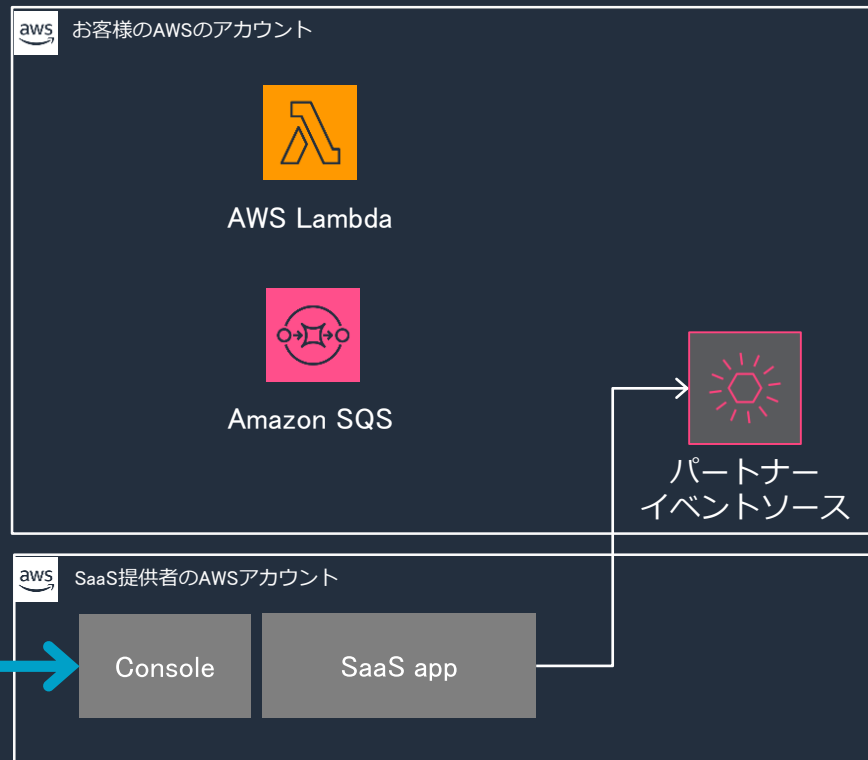
## ① パートナーイベントソースの作成

1. SaaSアプリケーションの管理画面からAWSアカウントIDとリージョンを設定する。

AWS account ID

AWS Region

Submit



## ② パートナーイベントソースとイベントバスの関連付け

### 2. パートナーイベントソースとイベントバスを関連付ける

パートナーイベントソース

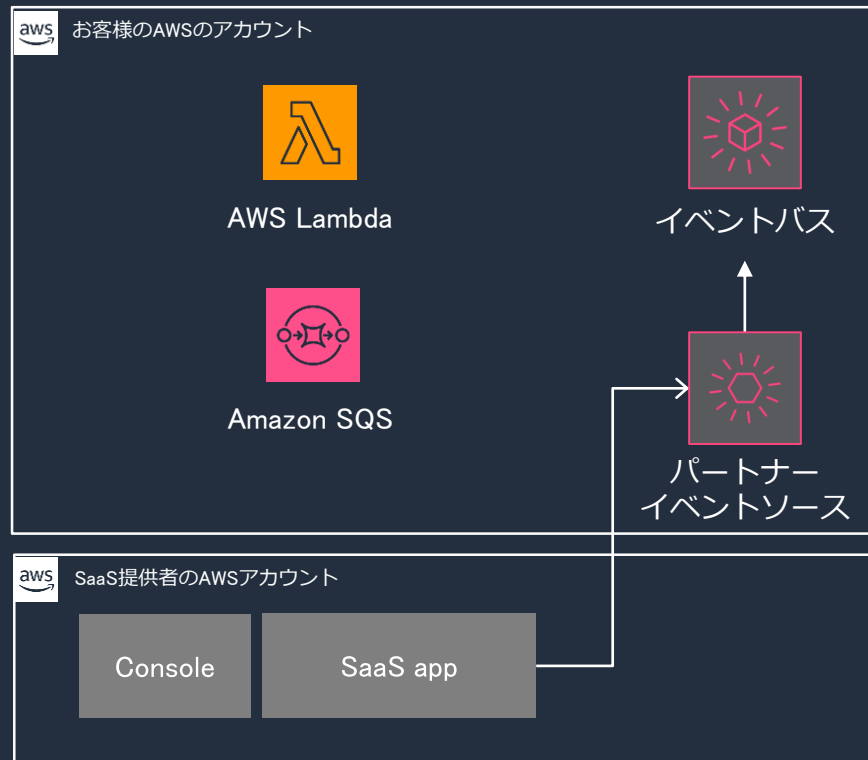
パートナーイベントソース (1) イベントバスと関連付ける

🔍 パートナーイベントソースを検索

< 1 > ⚙️

名前	ステータス	イベントバス
<a href="#">aws.partner/example.com/test-bus</a>	🕒 保留中	関連付けられたイベントバスなし

関連付けをする前は、  
パートナーイベントソースが作成されているが、  
イベントバスへの連携は開始されていない状態。



# SaaSアプリケーションとの連携方法

## ③ ルールの作成

3. ルールを作成し、イベントパターンとターゲットを定義する。

### ルールを作成

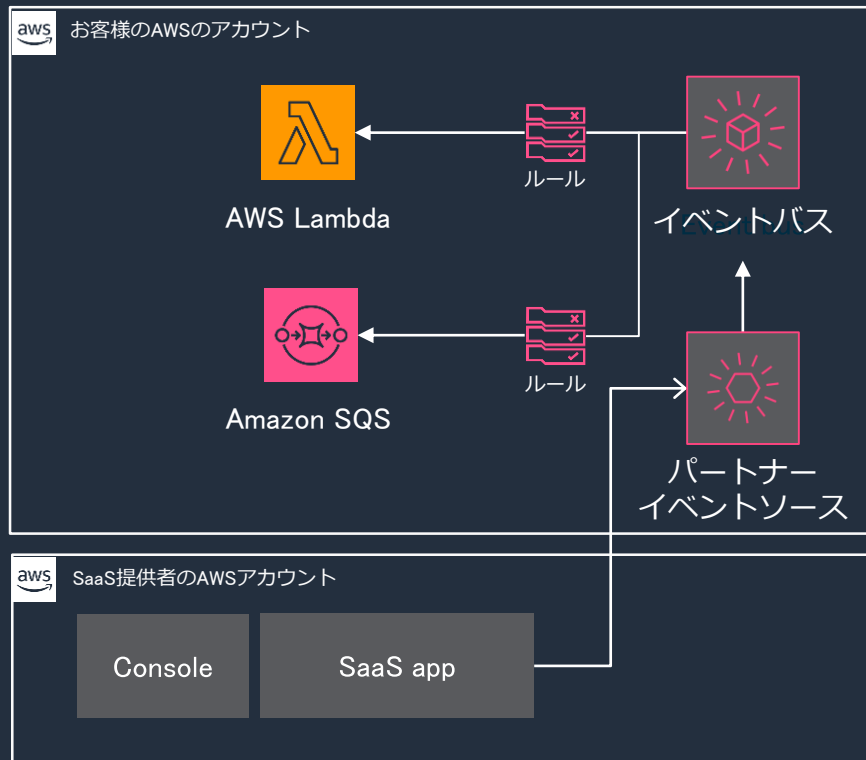
ルールは特定のイベントを監視し、ユーザーの選択した AWS ターゲットへと振り分けます。別の AWS アクションが起きたときに自動的に AWS アクションを実行するルールや、一定のスケジュールを設定して定期的に AWS アクションを実行するルールを作成できます。

#### 名前と説明

名前

小文字/大文字、.(ピリオド)、,(コンマ)、-(ハイフン)、\_(アンダーバー)を含め、最大 64 文字まで使用できます。

説明 - 任意



# Amazon EventBridgeパートナーになるには

For APN Partner

- WebhookやPushベースのイベント送信ができるSaaSアプリケーションを提供している場合、容易にEventBridgeと統合することができます。
- AWS上でSaaSを提供している事業者がAmazon EventBridgeパートナーになる場合は、AWS Partner Networkに登録していただき、弊社の担当者までご連絡ください。

# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

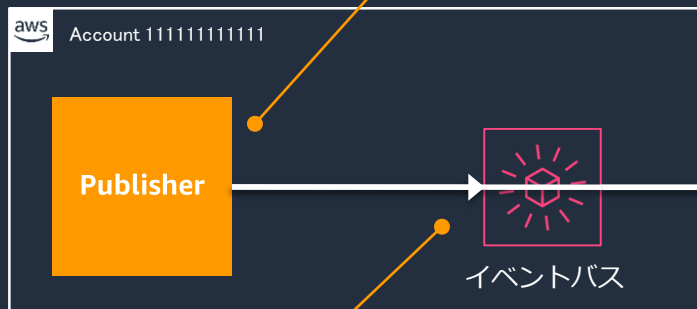
## 3. クロスアカウント連携

- 4. 直近のアップデート
- 5. 料金、制限
- 6. まとめ

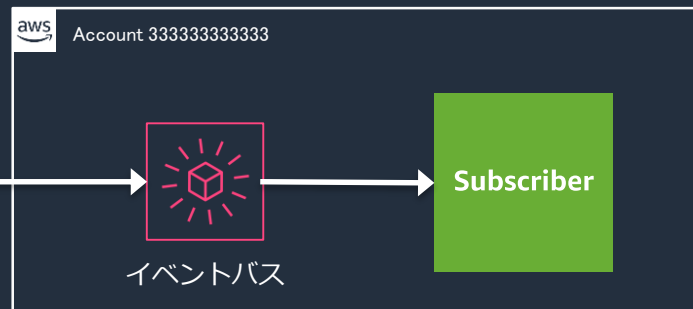
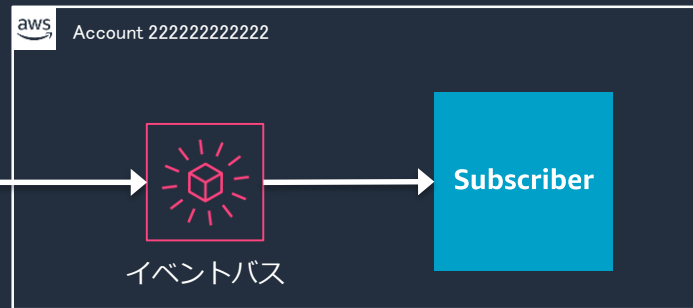
# クロスアカウント連携

複数のAWSアカウントで同じイベントを処理したい場合、イベントバス間で送受信することで依存関係が局所的にできる

Publisherは自身が存在するAWSアカウントのイベントバスのみ意識すればよい。



ルールで他アカウントのイベントバスをターゲットに指定する。  
アカウント間の依存関係がEventBridgeに閉じる。



# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

## 3. クロスアカウント連携

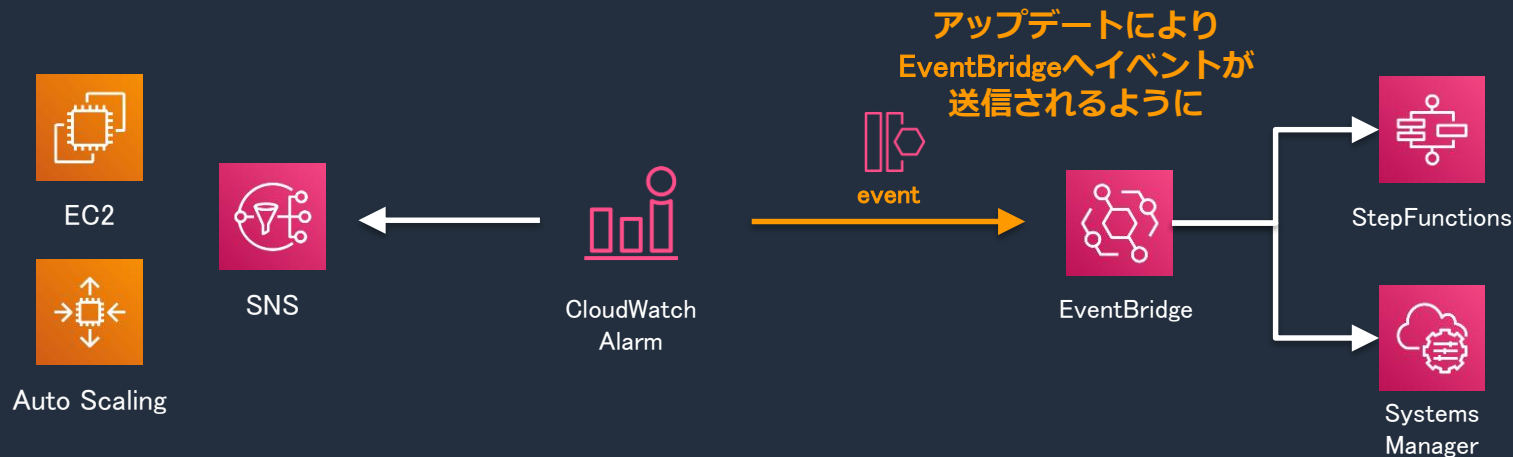
## 4. 直近のアップデート

## 5. 料金、制限

## 6. まとめ

# CloudWatch Alarmのイベントが取得可能に

- CloudWatch Alarmのステータス変化がEventBridgeで取得可能に。
- CloudWatch Alarmの機能では送信先としてEC2、SNS、AutoScalingの操作のみが選択可能。本アップデートによりAlarmのイベントが**実装なしで様々なサービスに連携可能**になった。



# ECR(Elastic Container Registry)のイベントが取得可能に

- ECRのイベントがEventBridgeで取得可能に。
- これまではCloudTrail経由で送信されるイベントのみ取得可能であったため、EventBridgeが受信するまでにタイムラグがあったが、**本アップデートによってECRに対するイベントをリアルタイムで活用可能に。**



# コンテンツベースのフィルタリング

イベントパターンで文字列表現以外の柔軟な記述が可能に。

	概要	イベントパターンの記述例
前方一致 (Prefix Matching)	文字列の前方一致が条件として指定できる。	<pre>{   "time": [ { "prefix": "2017-10-02" } ] }</pre>
例外条件 (Anything-but Matching)	〇〇以外を条件として指定できる。	<pre>{   "detail": {     "state": [ { "anything-but": "running" } ]   } }</pre>
数値比較 (Numeric Matching)	数値の大小や等値の比較ができる。	<pre>{   "detail": {     "c-count": [ { "numeric": [ "&gt;", 0, "&lt;=", 5 ] } ],   } }</pre>
存在性 (Exists Matching)	フィールドが「存在すること」や「存在しないこと」を条件として指定できる。	<pre>{   "detail": {     "c-count": [ { "exists": false } ]   } }</pre>

# Schema Registry & Schema Discovery (Preview)

- Schema Registry
  - EventBridge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参照可能。

```
aws.ec2@EC2InstanceStateChangeNotification

バージョン 1 作成者 2019年12月1日 9:11 JST   アクション ▼   コードバインディングのダウンロード

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)

```
バージョン 1 作成者 2019年12月1日 9:11 JST   アクシ

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  }
61  }
62  }
```

# Schema Registry & Schema Discovery (Preview)

- Schema Discovery
  - SaaSイベントやカスタムイベントがイベントバスに送信されると、スキーマ情報を自動生成可能に。
  - (例) DatadogのAlertが発生した際に送信されるイベントのスキーマ

aws.partner-datadog.com-example@DatadogAlertNotification

バージョン 2 作成者 2020年1月20日 17:46 JST

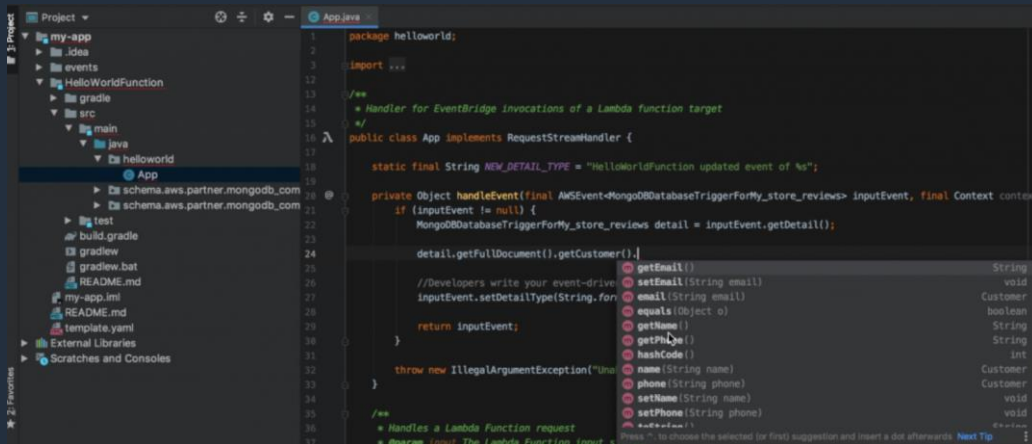
バージョン 2 ▾ 新しいバージョンとして保存 アクション ▾ コードバインディングのダウンロード

```
1 {
2   "openapi": "3.0.0",
3   "info": {
4     "version": "1.0.0",
5     "title": "DatadogAlertNotification"
6   },
7   "paths": {},
8   "components": {
9     "schemas": {
10      "AWSEvent": {
11        "type": "object",
12        "required": ["detail-type", "resources", "detail", "id", "source", "time", "region", "version", "account"],
13        "x-amazon-events-detail-type": "Datadog Alert Notification",
14        "x-amazon-events-source": "aws.partner/datadog.com/nogamincho",
15        "properties": {
```

AWSのサービスと全く同じ形で、SaaSのイベントやカスタムイベントのスキーマも確認できる

# Schema Registry & Schema Discovery (Preview)

- **Code Bindings** – スキーマ定義を言語ごとにダウンロード可能
  - AWS Toolkit for IntelliJ および Visual Studio Codeを用いてスキーマ定義を取得可能。
  - Java 8+、Python 3.6+、TypeScript 3 の 3 種類に対応
  - 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



# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

## 3. クロスアカウント連携

## 4. 直近のアップデート

## 5. 料金、制限

## 6. まとめ

# Amazon EventBridgeの料金

2020年1月22日時点の東京リージョンでの利用料金

課金対象		料金
受信した イベント	AWSサービスのイベント	無料
	カスタムイベント	100万件あたり1.00USD
	SaaSイベント	100万件あたり1.00USD
	クロスアカウントイベント	100万件あたり1.00USD
スキーマレジストリ (Preview)		無料
スキーマディスカバリ (Preview)		1ヶ月あたり500万件まで無料 それ以降は100万件あたり0.10USD

# Amazon EventBridgeの制限

上限の対象	上限値	上限緩和の可否
イベントの送信	毎秒 400 リクエスト	○
リクエストの最大サイズ	1リクエストあたり256KB	○
エントリ数	1リクエストあたり10エントリ	—
ターゲットの呼び出し	毎秒 750 回	○
イベントバス数	1リージョンあたり100	—
ルール数	1イベントバスあたり300個	○
イベントパターン	2048文字	—
ターゲット数	1ルールあたり5個	—

# 本日の内容

## 1. EventBridge とは？

- イベントバス導入の背景
- アーキテクチャ
- 構成要素
- アクセス制御

## 2. SaaS との連携

- SaaSアプリケーションとの連携案
- 対応しているSaaSアプリケーション
- SaaSアプリケーションとの連携方法
- EventBridge 活用事例
- EventBridgeパートナーになるには

## 3. クロスアカウント連携

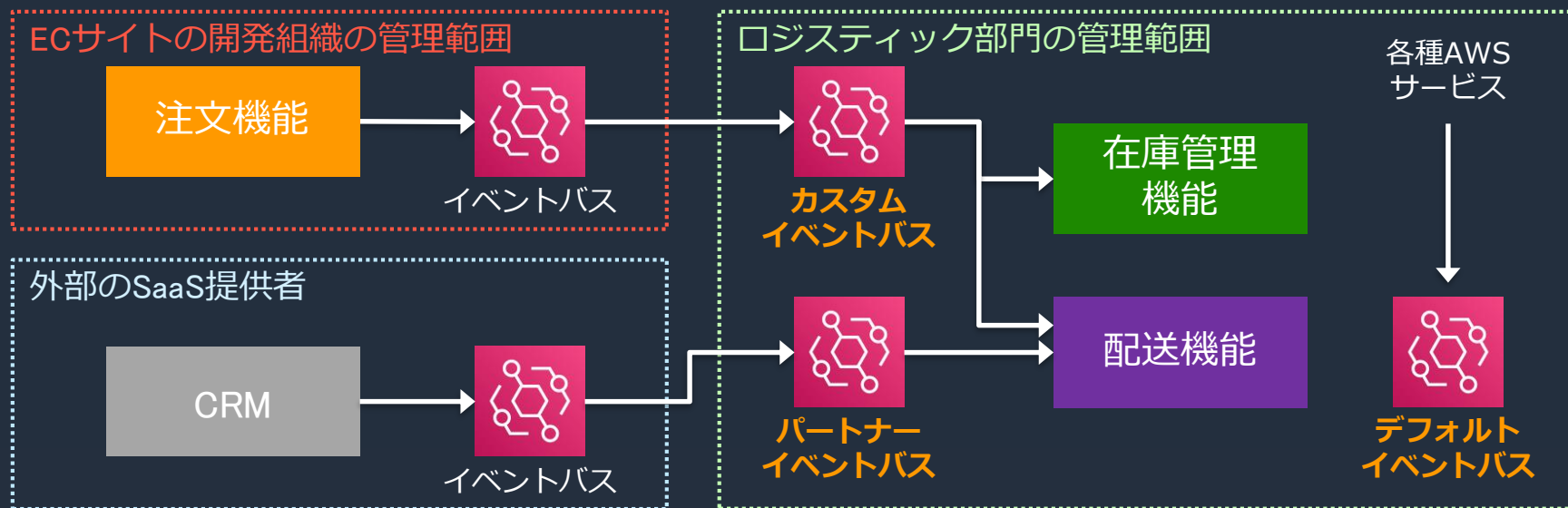
## 4. 直近のアップデート

## 5. 料金、制限

## 6. まとめ

# まとめ

- イベントを利用したアーキテクチャの構築が容易になる。
- サーバー管理不要なイベントバスが利用でき、実装なしにAWSサービスへイベントが連携できる。
- SaaSアプリケーションとのイベントの連携が容易になり、これまで取り扱えなかったSaaSのデータを利用できる。



# 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



日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼

コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

## AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#)

[AWS 初心者向け »](#)

[業種・ソリューション別資料 »](#)

[サービス別資料 »](#)

<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>

