



# Amazon CloudWatch Evidently

## AWS Black Belt Online Seminar

日平 大樹 (Taiki Hibira)

Technical Account Manager

2023 / 11

# AWS Black Belt Online Seminarとは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
- <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
- <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>

ご感想は X (Twitter) へ！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

- 本資料では資料作成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。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://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には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)

# 自己紹介

名前：日平 大樹 (Hibira Taiki)

所属：アマゾン ウェブ サービス ジャパン  
技術支援本部  
テクニカルアカウントマネージャー



経歴：HRシステム や 不動産業務 SaaS の開発

好きなAWSサービス：Amazon CloudWatch シリーズ

# 本セミナーの対象者

- 機能フラグを使い、一部のユーザーのみ新機能を利用できるようにすることで、安全にローンチしたい方
- A/B テストを実施し、より効果的な機能を定量的に判断したい方

# アジェンダ

1. Amazon CloudWatch の全体像と Application Monitoring
2. Amazon CloudWatch Evidently とは
3. Amazon CloudWatch Evidently - プロジェクト
4. Amazon CloudWatch Evidently - 機能
5. Amazon CloudWatch Evidently - 起動（機能フラグ）
6. Amazon CloudWatch Evidently - 実験（A/Bテスト）
7. Amazon CloudWatch Evidently の連携サービス
8. Amazon CloudWatch Evidently の注意事項、料金
9. まとめ

# Amazon CloudWatch の全体像と Application Monitoring

# エンドユーザーエクスペリエンスの重要性

アプリケーションのエンドユーザー（またはデジタル）エクスペリエンスが悪いと、評判、収益、定着率などのビジネスに悪影響を及ぼします。

**70%** のカスタマーが、ページ速度が「オンライン小売業者からの購入意欲に影響する」と答えています。

[Unbounce report 2019](#)

エンドツーエンドのアプリケーションパフォーマンスを監視する必要があり、統合されたオブザーバビリティツールが必要です。



# エンドユーザーエクスペリエンスを提供するには

優れた 体験を  
一貫して 提供

- 観測** ..... ユーザーが見ているものを確認
- 断続的な問題を把握
  - 問題がいつ発生したかを確認

- 調査** ..... デバッグの可視化
- パフォーマンスの問題を発見
  - 分散アプリケーションの対応

- 対応** .... フィードバックサイクルの短縮
- 顧客への影響を最小限に抑える
  - 問題の修正/ロールバックの実施

- 検証** ..... 新機能のリスクを軽減
- 監視機能付きの制御されたローンチ
  - 実験してベストを選択する

# エンドユーザーエクスペリエンスを提供するには

## 観測

..... ユーザーが見ているものを確認

- 断続的な問題を把握
- 問題がいつ発生したかを確認

優

従来のメトリクスやログの監視では不十分

発見

対応

目

- 顧客への影響を最小限に抑える
- 問題の修正/ロールバックの実施

## 検証

..... 新機能のリスクを軽減

- 監視機能付きの制御されたローンチ
- 実験してベストを選択する

# Amazon CloudWatchの全体像

## Application Monitoring



[外形監視]

CW Synthetics

[リアルユーザーモニタリング]

CW RUM

[フィーチャーフラグA/Bテスト]

CW Evidently

[インターネット監視]

CW Internet Monitor

[トレース]

CW ServiceLens

## Infrastructure



メトリクスを送信

ログを送信

イベントに応じた  
ターゲットによる処理



[イベント]

Amazon EventBridge/  
Amazon EventBridge  
Scheduler

[メトリクス]  
CW Metrics

ログの可視化

[ログ]  
CW Logs

[アラーム]  
CW Alarms

メトリクスに応じた  
アクション [リアルタイムメトリクス分析]

メトリクスの  
可視化  
CW Metrics Insight  
[タグベースの視覚化]  
CW Metrics Explorer

[メトリクスストリーム]  
CW Metrics Stream  
Amazon Kinesis  
Data Firehose

[ログ分析]  
CW Logs Insights

[ダッシュボードに統合]  
CW Dashboard



パートナー  
サービス、  
S3/Redshift

## Insights



[構造化ログによるメトリクス]

CW Container Insights / Contributor Insights

[Lambda拡張機能によるメトリクス]

Lambda Insights

[アプリケーションコンポーネントのメトリクス]

Application Insights

※CW = CloudWatch

# Application Monitoring サービス

## Amazon CloudWatch Internet Monitor

- アプリケーションとエンドユーザー間のインターネットの問題を可視化

## Amazon CloudWatch ServiceLens

- トレース、メトリクス、ログ、アラーム、および他のリソースヘルス情報を統合し、サービスとアプリケーションの監視性を強化

## Amazon CloudWatch Synthetics (canaries)

- アプリケーションのエンドポイントを継続的に監視

## Amazon CloudWatch Real-user monitoring (RUM)

- Web アプリケーションのパフォーマンスに関するクライアント側のデータをニアリアルタイムで収集

## Amazon CloudWatch Evidently

- A/B テストを実施し、エビデンスとデータに基づいて機能の設計を決定可能

# Amazon CloudWatch Evidentlyとは

# ビジネスを安全に加速

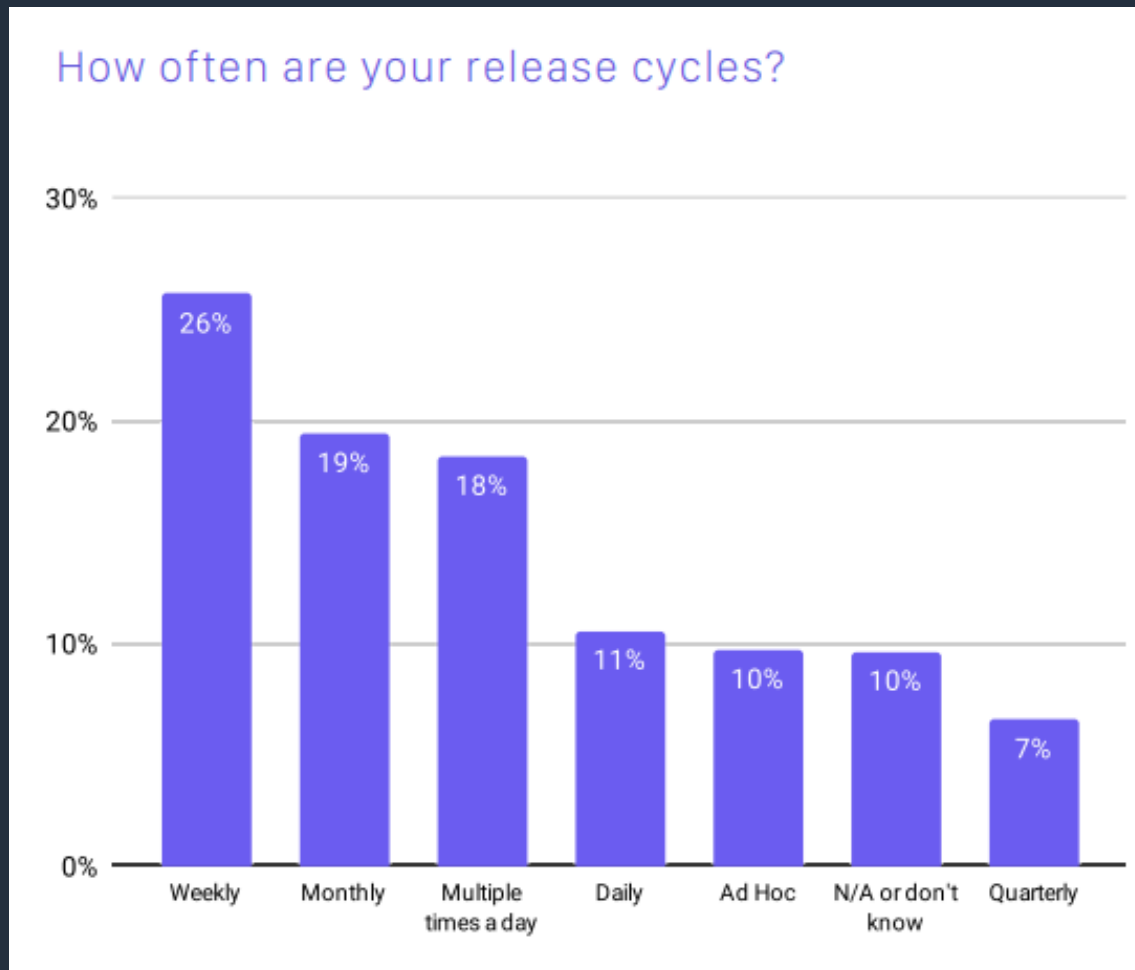
機能は本当に優れていますか？ ローンチしても安全ですか？

イノベーションが加速し、より迅速なローンチ・リリースが求められている

ローンチによる意図しない結果

- 予期しないユーザー行動
- パフォーマンスの低下

ユーザー体験の低下リスクが増加



*Courtesy of CNCF Cloud Native Survey 2020*

# ローンチリスクの管理

新機能がユーザー体験を低下させてないか

- 影響範囲のコントロール
- レスポンスタイムの計測
- エラー率の把握
- 離脱率の把握



トラフィック制御や  
監視

新機能は本当に優れているのか

- 定量的・統計上の評価
- ユーザー体験の監視



機能フラグ や A/B テスト

# Amazon CloudWatch Evidently

Amazon CloudWatch Evidently はアプリケーション開発者が一般向けの機能をリリースする前に、実験を行い、リスクを軽減し、意図しない結果を特定できるようにします。



---

安全な起動（機能フラグ・ダークローンチ）

意図しない結果を低減



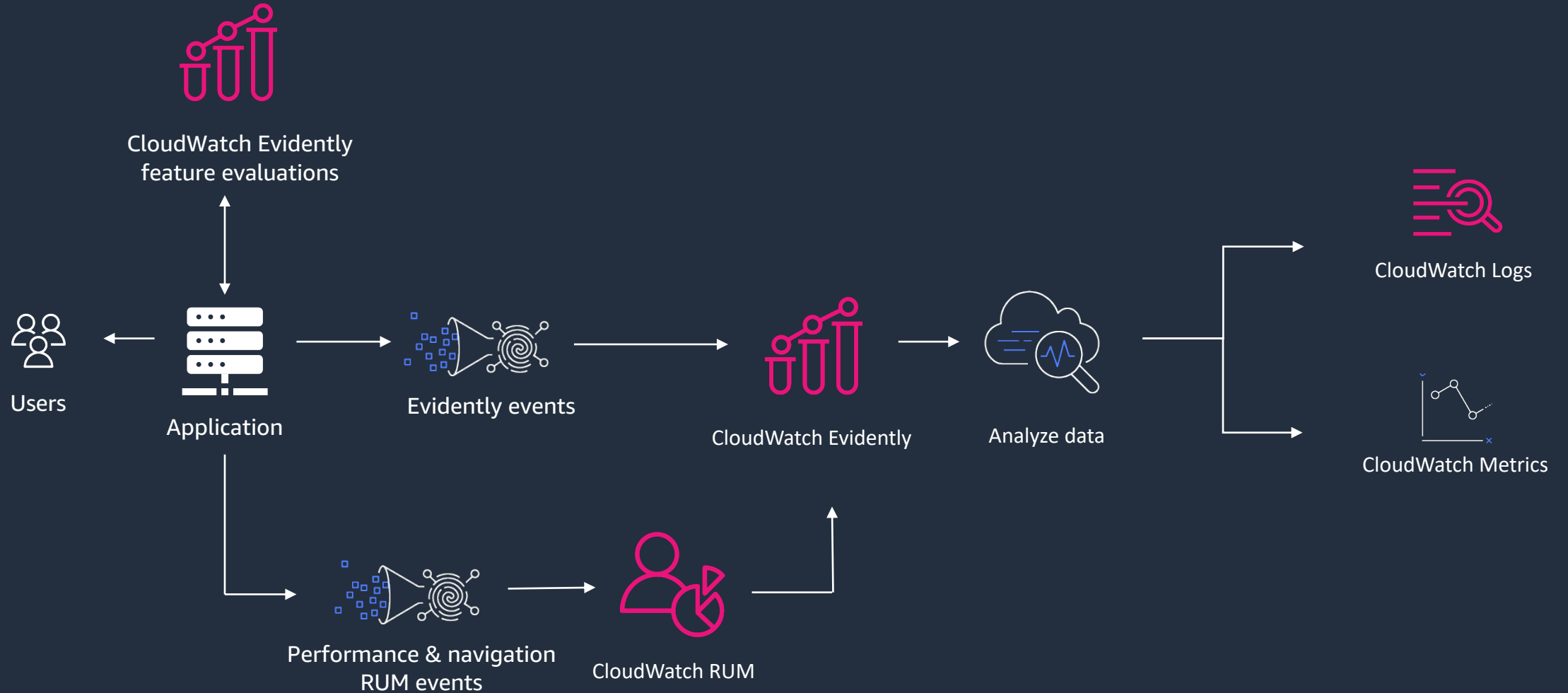
---

実験（A/Bテスト）

データに基づいた判断



# Amazon CloudWatch Evidently - 構成例

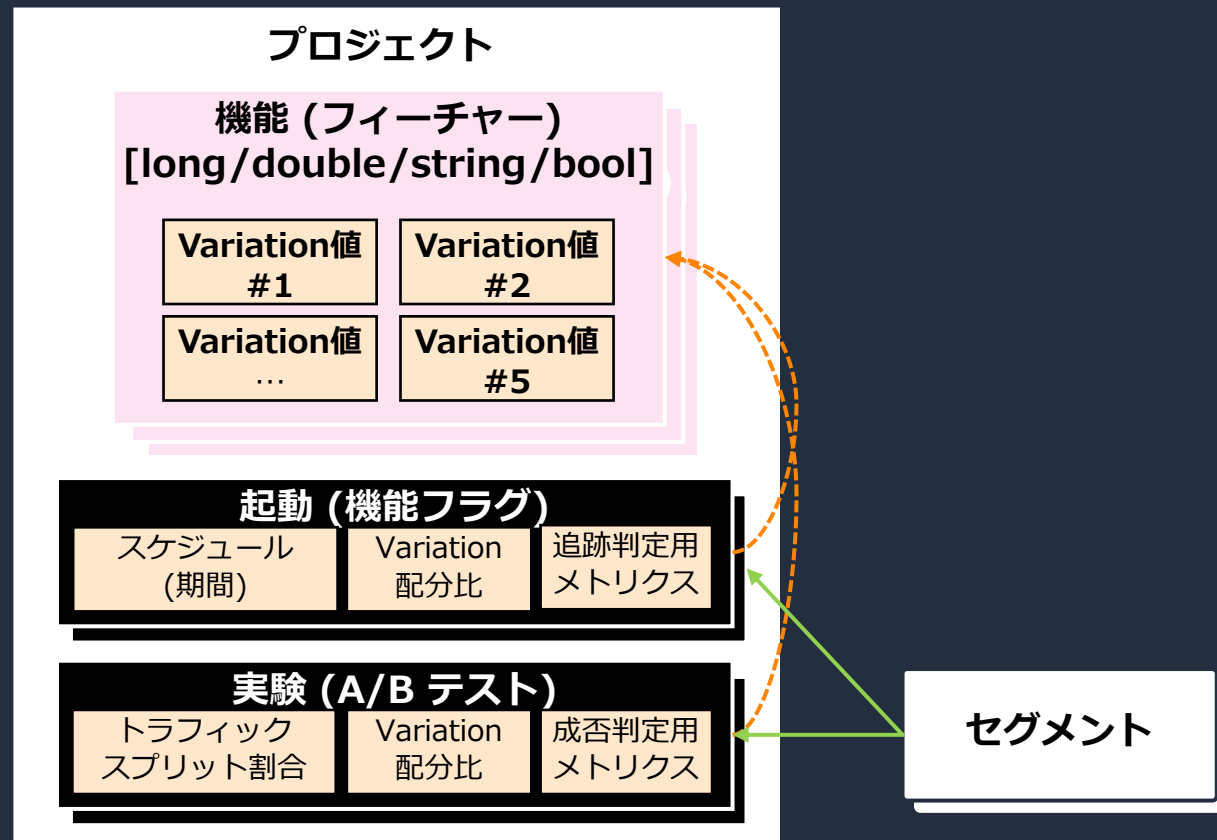


# Amazon CloudWatch Evidently - プロジェクト

# Amazon CloudWatch Evidently – プロジェクト



## CloudWatch Evidently



プロジェクトは、リソースを論理的にグループ化したものです。

CloudWatch Evidently を開始するには、「機能」の「起動」または「実験」に対して、「プロジェクト」を作成します。

### プロジェクトの詳細

プロジェクト名

プロジェクトを識別する簡単な方法

プロジェクト名

名前は、127 文字までの文字、数字、ハイフン、またはアンダースコアで構成され、スペースを含めることはできません。

プロジェクトの詳細 - オプション

プロジェクトの説明

# Amazon CloudWatch Evidently – 評価イベントストレージ

## 評価イベントストレージ 情報

収集する Evidently 評価イベントを Amazon S3 または CloudWatch Logs に保存するかどうかを選択します。評価イベントを保存すると、これらのイベントを独自に分析できます。

- S3**  
評価イベントのコピーをアカウントの Amazon S3 バケットに送信します。
- CloudWatch Logs**  
Evidently は評価イベントのコピーを CloudWatch Logs のアカウントに保存し、CloudWatch Logs Insights でクエリを実行できるようにします。
- 評価イベントを保存しないでください  
評価イベントを保存していない場合でも、Evidently はそれらを使用してメトリクスやその他の実験データを生成します。これらのデータは、Evidently で表示できません。

## S3 の場所

データを保存する S3 の場所を選択します。既存の S3 バケットを選択するか、新しい S3 バケットを作成します。 [🔗](#)

## S3 URI

🔍 s3://bucket/prefix/object

表示 [🔗](#)

S3 を参照

Evidently イベントには次の 2 つのタイプがあります。

## • 評価イベント

評価イベントは、ユーザーに表示される機能のバリエーションに関連しています。

Evidently が収集する評価イベントを Amazon S3 または CloudWatch Logs に保存するかどうかを選択します。

## • カスタムイベント

カスタムイベントは、クリックやチェックアウトなどのユーザーアクションからメトリクスを生成するために使用されます。

保存する場合は、アプリケーションコードを変更して、Evidently 以外のストレージを選択して送信する必要があります。

# Amazon CloudWatch Evidently – セグメント

## セグメントパターン 情報

```
1  {
2    "Location": [
3      "Japan"
4    ],
5    "$or": [
6      {
7        "Browser": [
8          "Safari"
9        ]
10     },
11     {
12       "Device": [
13         "Tablet"
14       ]
15     }
16   ]
17 }
```

セグメントとは、1 つまたは複数の特徴を共有する、ユーザーのグループの一部です。

例) Chrome ブラウザのユーザー、ヨーロッパのユーザー、ヨーロッパの Firefox ブラウザのユーザー、50歳以上、など

起動や実験でセグメントを使用すると、異なるユーザーのグループに対して異なるトラフィック分割を定義できます。

# Amazon CloudWatch Evidently - 機能

# Amazon CloudWatch Evidently - 機能（フィーチャー）

CloudWatch Evidently の「機能」は、起動(フィーチャーフラグ)する機能や実験(A/Bテスト)する機能を表しています。  
起動をスケジュールしたり、実験を開始する前に、機能を Evidently API でコード化して実装する必要があります。

## 機能の詳細

プロジェクト: myloTApp-myEvidentlyProject

---

**機能名**  
機能を識別する簡単な方法

名前は、127 文字までの小文字、数字、ハイフン、ピリオド、またはアンダースコアで構成でき、スペースを含めることはできません。

**機能の説明 - オプション**

# Amazon CloudWatch Evidently - 機能のバリエーション

**機能のバリエーション** [情報](#)  
機能のバリエーションと取得できる値を追加します。

バリエーションタイプ [情報](#)  
長整数値

バリエーション名	値 <a href="#">情報</a>	デフォルト <a href="#">情報</a>	
<b>V1</b> Default	0	<input checked="" type="radio"/>	<a href="#">削除</a>
<b>V2</b> Variation1	1	<input type="radio"/>	<a href="#">削除</a>
<b>V3</b> Variation2	2	<input type="radio"/>	<a href="#">削除</a>
<b>V4</b> Variation3	3	<input type="radio"/>	<a href="#">削除</a>
<b>V5</b> Variation4	4	<input type="radio"/>	<a href="#">削除</a>

[バリエーションを追加](#)  
5 個のバリエーションの制限に達しました

1つの機能には最大 5 つのバリエーションを追加できます。

バリエーションを定義するときは、以下のバリエーションの型を選択する必要があります。

- ブール値
- 整数
- 倍精度浮動小数点数
- 文字列

デフォルトバリエーションは、他のバリエーションが比較対象とするベースラインであり、この機能の起動や実験に加わっていないユーザーに提供されるバリエーションでもあります。



# Amazon CloudWatch Evidently – 機能のコード化

## ▼ サンプルコード

次のサンプルコードを使用して、アプリケーションで機能評価を設定します。SDK をセットアップしていない場合は、[こちらの手順に従います。](#)

JavaScript

コピー

ダウンロード

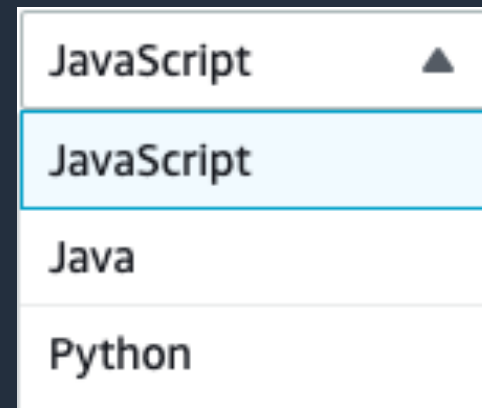
```
1 // Set up credentials here
2
3 // API request structure
4 const evaluateFeatureRequest = {
5   // entityId is a JSON key to store the entity ID in custom event.
6   // The entity ID used in evaluateFeature API needs to be the same as the one used in putProjectE
7   entityId: 'myId',
8   // Name of your feature
9   feature: '',
10  // Name of your project
11  project: "CloudWatchEvidently-BlackBeltOnlineSeminar",
12 };
13
14 // Evaluate feature
15 evidently.evaluateFeature(evaluateFeatureRequest)
16   .promise()
17   .then(response => {
18     // Instrument your code based on evaluate feature API response
19   });
20 });
```

## 予想される Evaluate Feature API レスポンス

```
1 // Example of API response when no rules are being evaluated on the feature
2 // (no experiment, no launch and no overrides)
3 {
4   "details": {},
5   "reason": "DEFAULT",
6   "value": "{
7     "longValue": "0"
8   }",
9   "valueType": "LONG",
10  "variation": "Default",
11 }
```

アプリケーションの機能をコード化し、Evidently API を使用してトリガーをセットアップします。

サンプルコードとして、JavaScript、Java、Python が提供されます。



# Amazon CloudWatch Evidently – 主なAPI

## EvaluateFeature API

特定の 1 つのユーザーセッションに機能のバリエーションを割り当てます(割り当てイベント)。Evidently は、同じ entityId と機能では、必ず同じバリエーションを割り当てます。

[パラメーター]

- **project:** プロジェクト
- **feature:** 機能
- **entityId:** 一意のユーザー (セッション ID、ユーザー ID、ハッシュなど)
- **evaluationContext:** ユーザーに関する追加情報を表す JSON オブジェクト。(オプション)

## PutProjectEvents API

パフォーマンスイベントを Evidently に送信します。これらのイベント(データイベント)は、起動や実験の評価に使用できます。

[パラメーター]

- **project:** プロジェクト
- **events:** Evidently に送信されるパフォーマンスデータを含む JSON 配列
  - **timestamp:** イベントのタイムスタンプ
  - **type:** イベントの種類
    - **aws.evidently.evaluation:** ユーザーに表示される機能のバリエーションを決定する評価イベント
    - **aws.evidently.custom:** クリックやチェックアウトなどのユーザーアクションからメトリクスを生成するカスタムイベント
- **data:** イベントデータ



# Amazon CloudWatch Evidently - オーバーライド

## ▼ オーバーライド - オプション 情報

識別子を追加して、選択した訪問者のバリエーションを事前に決定します。識別子は、UserID、AccountID、またはスペースを含まない英数字の文字列です。

識別子	バリエーション	
<input type="text" value="default_user"/>	Default ▼	<input type="button" value="削除"/>
<input type="text"/>	Variation1 ▼	<input type="button" value="削除"/>

最大でさらに 18 個のオーバーライドを追加できます

特定のユーザーに常に特定のバリエーションが表示されるように指定するには、「オーバーライド」を指定します。

ユーザー ID、アカウント ID、またはその他の識別子を入力することで、そのユーザーに表示されるバリエーションを指定できます。

オーバーライドが割り当てられているユーザーのセッションは、起動または実験メトリクスに寄与しません。

# Amazon CloudWatch Evidently - 評価ルール

## 評価ルール



CloudWatch Evidently コンソールを使用して、機能の評価ルールが機能の現在の起動、実験、およびバリエーションに対象者トラフィックをどのように割り当てているかを確認できます。

- ① オーバーライド に指定したユーザーに、指定したバリエーションが割り当てられます。
- ② 起動で指定したセグメントに対して、指定した割合でバリエーションが割り当てられます。
- ③ 実験で指定したセグメントに対して、指定した割合でバリエーションが割り当てられます。
- ④ 最後まで割り当てられなかったユーザーに対しては、デフォルトのバリエーションが割り当てられます。

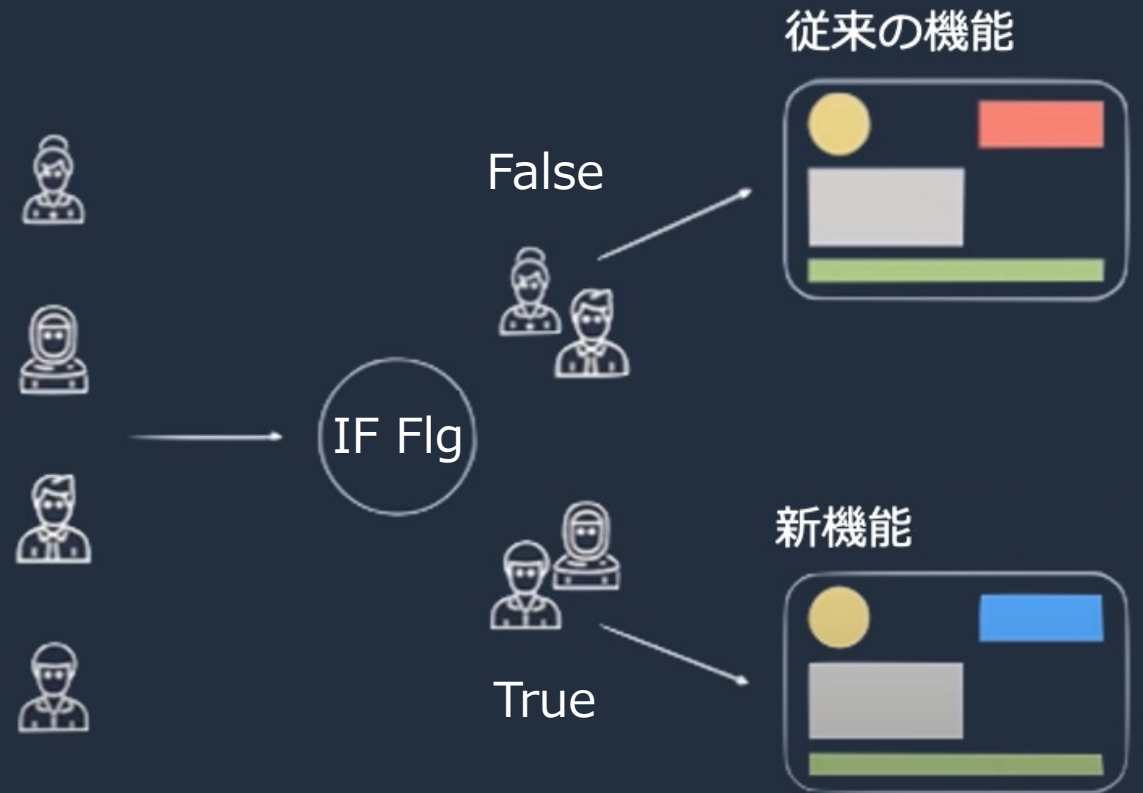
# Amazon CloudWatch Evidently

## - 起動 (機能フラグ・ダークローンチ)

# 機能フラグ（フィーチャーフラグ） ・ ダークローンチとは

機能の有効または無効を  
コードのデプロイなしに行う  
ソフトウェア開発手法

ローンチをモニタリングし、  
意図しない結果の場合は  
すぐに戻すことが可能



# Amazon CloudWatch Evidently - 起動 (機能フラグ・ダークローンチ)



スモールトラフィック

**限定的なリリース**



モニタリングと制御

**異常の視覚化  
機能提供の停止**



ダイヤルアップ/  
ダウントラフィック

**トラフィックの  
コントロール**

# Amazon CloudWatch Evidently - 起動のステップ

機能の選択

起動設定

セグメントオーバーライド

起動スケジュール

メトリクスの指定

アラームの指定

モニタリングと分析



# Amazon CloudWatch Evidently – 機能の選択

機能は「既存の機能から選択」または「新機能を追加」のいずれかを選択します。

既存の機能を使用する場合は、その機能を「機能名」で選択します。

**機能 情報**

起動する機能を追加し、そのバリエーションを定義します。割り当てをセットアップしてトリガーするには、機能をコード化して Evidently API で実装する必要があります。

機能

既存の機能から選択  
既に追加されている機能から選択します。

新しい機能を追加  
新しい機能とその定義が機能リストに追加されます。

機能名

LaunchFeature ▼

バリエーションタイプ

ブール値

バリエーション

<b>V1</b> Default <span>デフォルト</span>	<b>V2</b> NewFeature
値 false	値 true

ルール

デフォルト: **V1** Default

トラフィック: 100%

# Amazon CloudWatch Evidently – 起動設定

## 起動設定 情報

起動トラフィックと起動計画をセットアップします。

### オプション

**今すぐ起動を開始**  
すぐに起動を開始します。

**起動をスケジュール**  
後の日時に起動をスケジュールするか、  
複数のステップをスケジュールした計画  
をセットアップします。

### トラフィックの設定

設定されたセグメントオーバーライドに一致しないエンティティは、  
下部に指定されたトラフィック分散が発生します。

セグメントオーバーライドを追加

### バリエーション

V1 Default **デフォルト**

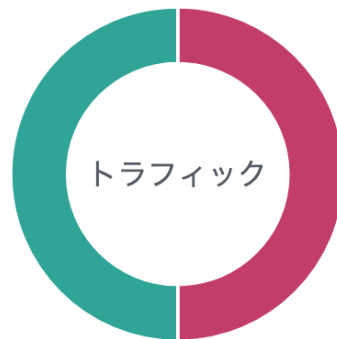
### トラフィックの割合

50 %  含む

V2 NewFeature

50 %  含む

**i** トラフィックの 0% は未割り当てであり、次のルールで使用できます。



## オプション

起動をすぐに開始するか、後で開始するようにスケジュールするかを選択します。

## トラフィックの設定

セグメントオーバーライドに一致しないエンティティのトラフィック分割を指定します。

# Amazon CloudWatch Evidently – セグメントオーバーライド

## セグメントオーバーライド

特定のセグメントに一致するすべてのユーザーに適用する起動設定を追加します。複数のセグメントに属するユーザーは、このリスト内で最初に一致するセグメント設定に従って評価されます。

セグメント

Safari-or-Tablet

セグメントオーバーライドを削除

バリエーション

バリエーション	デフォルト	トラフィックの割合
V1	Default <b>デフォルト</b>	0 %
V2	NewFeature	100 %

ⓘ トラフィックの 0% は未割り当てであり、次のルールで使用できます。

セグメントオーバーライドを追加

セグメントオーバーライドは最大 5 個追加できます。

キャンセル **確認**

セグメントオーバーライドでは、特定のセグメントに一致するすべてのユーザーに適用する起動設定を追加します。

複数のセグメントに属するユーザーは、このリスト内で最初に一致するセグメント設定に従って評価されます。



# Amazon CloudWatch Evidently – 起動スケジュール

ステップ日付: 2023/09/30 | ステップ時間 (JST): 10:00 | セグメントオーバーライドを追加

バリエーション: V1 Default (デフォルト) | V2 NewFeature

トラフィックの割合: V1 67% | V2 33%

③ トラフィックの0%は未割り当てであり、次のルールで使用できます。

トラフィック

---

ステップ日付: 2023/09/30 | ステップ時間 (JST): 11:00 | セグメントオーバーライドを追加

バリエーション: V1 Default (デフォルト) | V2 NewFeature

トラフィックの割合: V1 33% | V2 67%

③ トラフィックの0%は未割り当てであり、次のルールで使用できます。

トラフィック

---

ステップ日付: 2023/09/30 | ステップ時間 (JST): 12:00 | セグメントオーバーライドを追加

バリエーション: V1 Default (デフォルト) | V2 NewFeature

トラフィックの割合: V1 0% | V2 100%

③ トラフィックの0%は未割り当てであり、次のルールで使用できます。

トラフィック

起動をスケジュールする場合は、複数のステップを起動に追加できます。

各ステップでは、バリエーションの提供に異なる割合を使用できます。

1つの起動には、最大5つのステップを含めることができます。

# Amazon CloudWatch Evidently – メトリクスの指定(オプション)

▼ **メトリクス - オプション** 情報

機能のパフォーマンスをモニタリングするために起動中に追跡されるメトリクス。メトリクスを計測するには、[詳細はこちら](#) 

メトリクスソース  
追跡するメトリクスのメトリクスソースを選択します。

カスタムメトリクス ▼ 削除

メトリクス名

メトリクスルール 情報  
イベントに適用するルールを追加して、メトリクスを生成します。ログ記録 API を計測していない場合は、[こちらの手順に従ってください](#)

エンティティ ID 情報    値キー 情報    単位 - オプション 情報

カスタムイベントパターンの作成 - オプション  
メトリクスの記録方法を定義する JSON ブロックを作成します。詳細については、「」を参照してください。 [Amazon EventBridge イベントパターン](#)。 

メトリクスを追加

最大でさらに 2 個のメトリクスを追加できます。

起動中に機能のパフォーマンスを追跡する場合は、メトリクスを追加します。

メトリクスは、CloudWatch RUM メトリクスまたはカスタムメトリクスのいずれかを使用できます。

カスタムメトリクスにおけるエンティティ ID はカスタムイベントにエンティティ ID を保存する JSON キーです。

エンティティ ID は EvaluateFeature API や PutProjectEvents API で使用される entityId と同じである必要があります。

# Amazon CloudWatch Evidently – アラームの指定(オプション)

## ▼ CloudWatch アラームを関連付ける - オプション 情報

アラームの名前を指定して、既存の CloudWatch アラームをこの起動に関連付けます。アラームを関連付けるため、Evidently は識別されたアラームに、このプロジェクト `evidently:EvidentlySample:alarm` のタグとこの起動 `evidently:EvidentlySample::alarm` のタグでタグ付けします。

Evidently が、この起動リソースで以下に識別されるアラームリソースにタグ付けできるようにします。

アラーム名

Synthetics-Alarm-Evidently-Demo-Service

削除

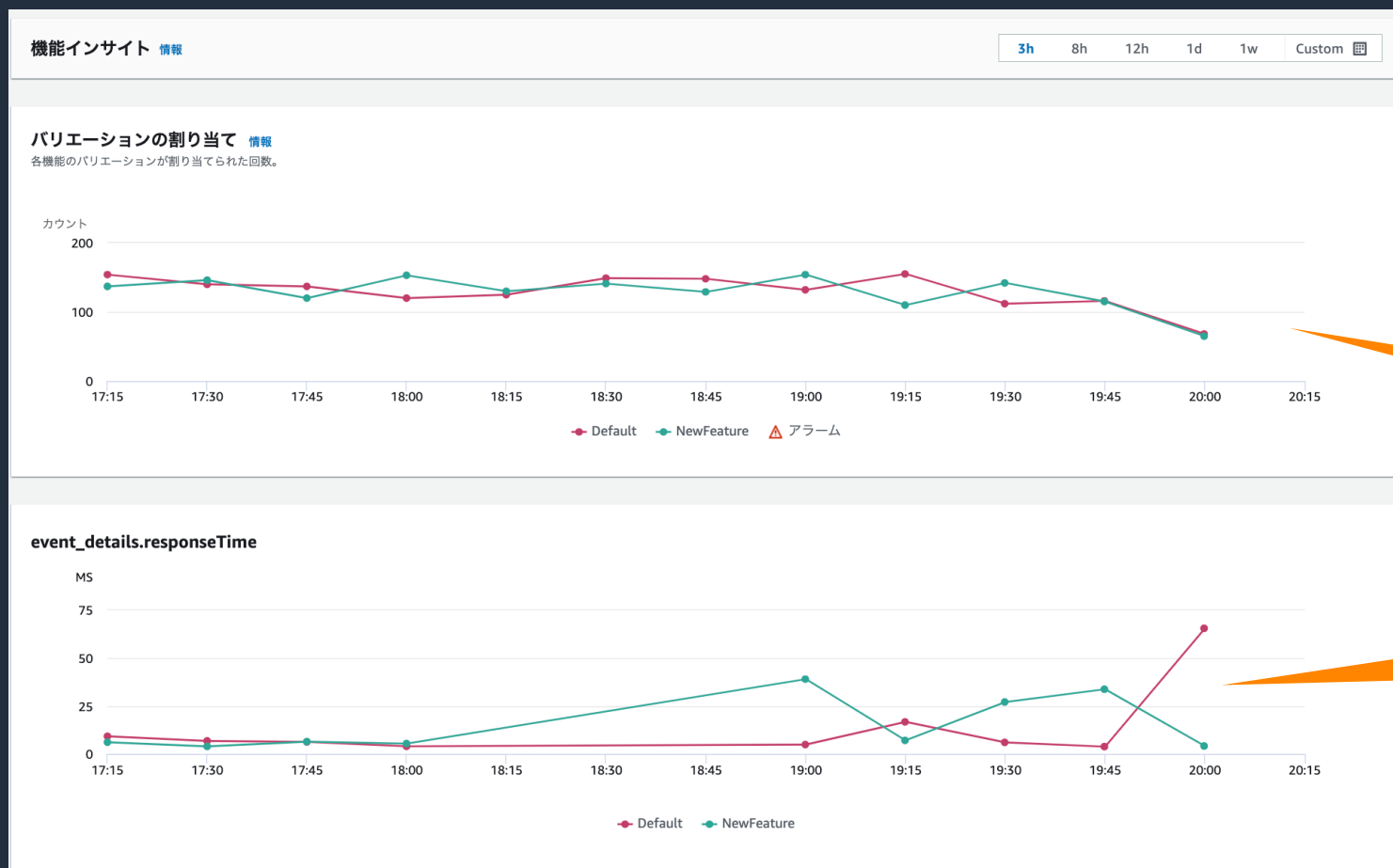
アラームを関連付け

最大でさらに 4 個のアラームを関連付けることができます

起動には 既存の CloudWatch アラームをこの起動に関連付けることができます。

起動にアラームを関連付ける場合、CloudWatch Evidently がコンソールの起動情報に正しいアラームを表示できるようにするために タグをアラームに追加する必要があります。

# Amazon CloudWatch Evidently – モニタリングと分析



各バリエーションに割り当てられたユーザーセッションの数を時間の経過とともに確認し、起動での各バリエーションのパフォーマンスメトリクスを表示できます。

各バリエーションに割り当てられたユーザーセッション数

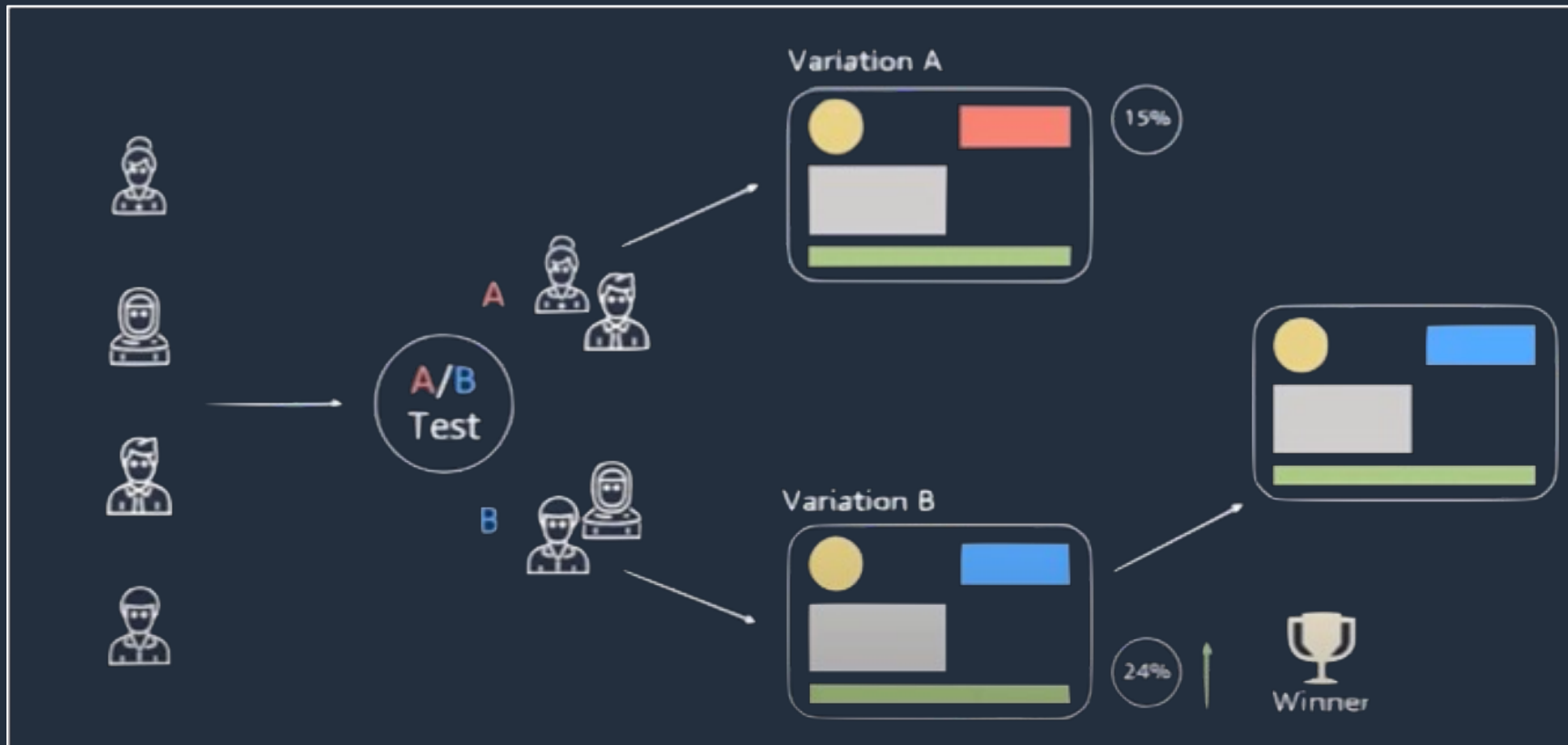
指定したメトリクス(オプション)

# Amazon CloudWatch Evidently - 実験 (A/B テスト)



# A/B テストとは

複数のバリエーションの応答を比較することで、  
どちらがより効果的かを定量的に判断



# Amazon CloudWatch Evidently - 実験 (A/B テスト)



定量的な結果

**統計手法を利用**



リアルタイムの分析

**分析のための時間を短縮**



ローンチ方法の選択

**起動を使用してトラフィック  
のダイヤルアップを自動化**

# Amazon CloudWatch Evidently - 実験のステップ

機能の選択

対象者の指定

メトリクスの指定

アラームの指定

実験の開始

実験結果の評価

# Amazon CloudWatch Evidently – 機能の選択

「既存の機能から選択」または「新機能を追加」のいずれかを選択します。

既存の機能を使用する場合は、その機能を「機能名」で選択します。

**機能 情報**  
起動する機能を追加し、そのバリエーションを定義します。割り当てをセットアップしてトリガーするには、機能をコード化して Evidently API で実装する必要があります。

機能

既存の機能から選択  
既に追加されている機能から選択

新しい機能を追加  
新しい機能とその定義が機能リストに追加されます。

機能名  
VariationFeature ▼

バリエーションタイプ  
長整数値

バリエーション

V1 Default デフォルト 値 0	V2 Variation1 値 1
V3 Variation2 値 2	V4 Variation3 値 3
V5 Variation4 値 -1	

# Amazon CloudWatch Evidently – 対象者の指定

## 対象者 情報

バリエーションをテストする参加可能な対象者の割合を定義します。

### セグメントを選択

適用するセグメント定義を選択します。

すべてのトラフィック

### 実験のトラフィック分割

実験に参加する、利用可能な訪問者の割合

100 %

0~100% の範囲内

### 各バリエーションのトラフィック比率

#### バリエーション

V1 Default **デフォルト**

V2 Variation1

V3 Variation2

V4 Variation3

V5 Variation4

#### トラフィック比率

20 %   含む

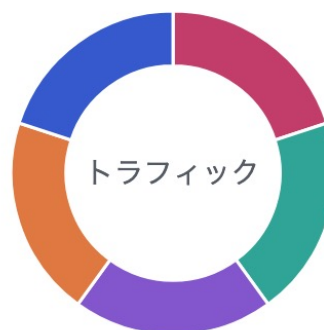
20 %   含む

20 %   含む

20 %   含む

20 %   含む

均等に分散



この実験を、そのセグメントに適合するユーザーのみに適用する場合は、セグメントを選択します。

「実験のトラフィック分割」では、実験に参加するトラフィックの割合を指定できます。

各バリエーションのトラフィック比率では、各バリエーションに割り当てるトラフィックの割合を指定できます。

# Amazon CloudWatch Evidently – メトリクスの指定

**メトリクス** [情報](#)

実験の成功を判断するために追跡するメトリクス。メトリクスが利用できない場合は、[計測メトリクス](#)

メトリクスソース  
追跡するメトリクスのメトリクスソースを選択します。

カスタムメトリクス ▼ 削除

メトリクス名 目標

PageLoadTime 減少 ▼

メトリクスルール [情報](#)  
イベントに適用するルールを追加して、メトリクスを生成します。ログ記録 API を計測していない場合は、[こちらの手順に従ってください](#)

エンティティ ID [情報](#) 値キー [情報](#) 単位 - オプション [情報](#)

userDetails.entityId details.pageLoadTime ms

カスタムイベントパターンの作成 - オプション  
メトリクスの記録方法を定義する JSON ブロックを作成します。詳細については、「」を参照してください。 [Amazon EventBridge イベントパターン](#)。

実験中のバリエーションを評価するために使用するメトリクスを選択します。メトリクスのソースでは、CloudWatch RUM メトリクスか、カスタムメトリクスを選択します。

「目標」では、メトリクスの値が大きい方が良いバリエーションである場合、「増加」を選択し、値が小さい方が良いバリエーションである場合、「減少」を選択します。

# Amazon CloudWatch Evidently – アラームの指定(オプション)

## ▼ CloudWatch アラーム - オプション 情報

必要に応じて、実験メトリクスのアラームを作成できます。作成すると、バリエーションごとにアラームが作成されます。

メトリクス名	アラームの状態	しきい値	期間全体の平均	
PageLoadTime ▼	より大きい ▼	1	5分 ▼	削除
<input type="button" value="新しいアラームを追加"/>				

## ▶ アラームの通知を設定する

メトリクスが定義したレベルに達したときに通知を受け取る場所を選択する

作成

実験でアラームを利用すると、各バリエーションとデフォルトのバリエーションの結果の差が指定したしきい値より大きいかどうかをモニターリングできます。


## CloudWatch アラーム (5)

編集

アラームの状態 ▲	アラーム名 ▼	メトリクス名 ▼	SNS トピック
OK	Evidently-Alarm-CloudWatchEvidently-BlackBeltOnlineSeminar-VariationTest-1_Default	PageLoadTime	-
OK	Evidently-Alarm-CloudWatchEvidently-BlackBeltOnlineSeminar-VariationTest-1_Variation1	PageLoadTime	-
OK	Evidently-Alarm-CloudWatchEvidently-BlackBeltOnlineSeminar-VariationTest-1_Variation2	PageLoadTime	-
OK	Evidently-Alarm-CloudWatchEvidently-BlackBeltOnlineSeminar-VariationTest-1_Variation3	PageLoadTime	-
OK	Evidently-Alarm-CloudWatchEvidently-BlackBeltOnlineSeminar-VariationTest-1_Variation4	PageLoadTime	-

実験でアラームを作成すると、デフォルトのバリエーションではないバリエーションごとに1つずつアラームが作成されます。

# Amazon CloudWatch Evidently – 実験の開始



実験: VariationTest

実験をキャンセル 実験トラフィックを変更 アクション ▲

実験を管理

実験を開始

実験を編集

実験を削除

その他

起動を作成

セットアップ  
機能: [VariationFeature](#)  
2 個のメトリクス、5 個のバリエーション

スケジュール  
まだ開始されていません  
[Start experiment](#)

アラームの実験  
アラームなし

セグメント  
セグメントなし

ステータス  
作成済

実験を作成すると、作成済みステータスで実験が生成されます。

作成した実験を選択し、実験を開始を行うことで、すぐに実験を開始できます。

実験は指定した終了日時に、分析が停止しますが、トラフィックは継続します。



実験を開始

実験の終了日と終了時刻を定義し、開始します。

▶ 実験のセットアップ

実験終了をスケジュール

実験の終了日時を定義します。スケジュールされた終了時刻に分析は停止しますが、トラフィックは継続します。実験はすぐに開始されます。結果は、実験の開始から 63 日間利用できます。

終了日 2023/09/24

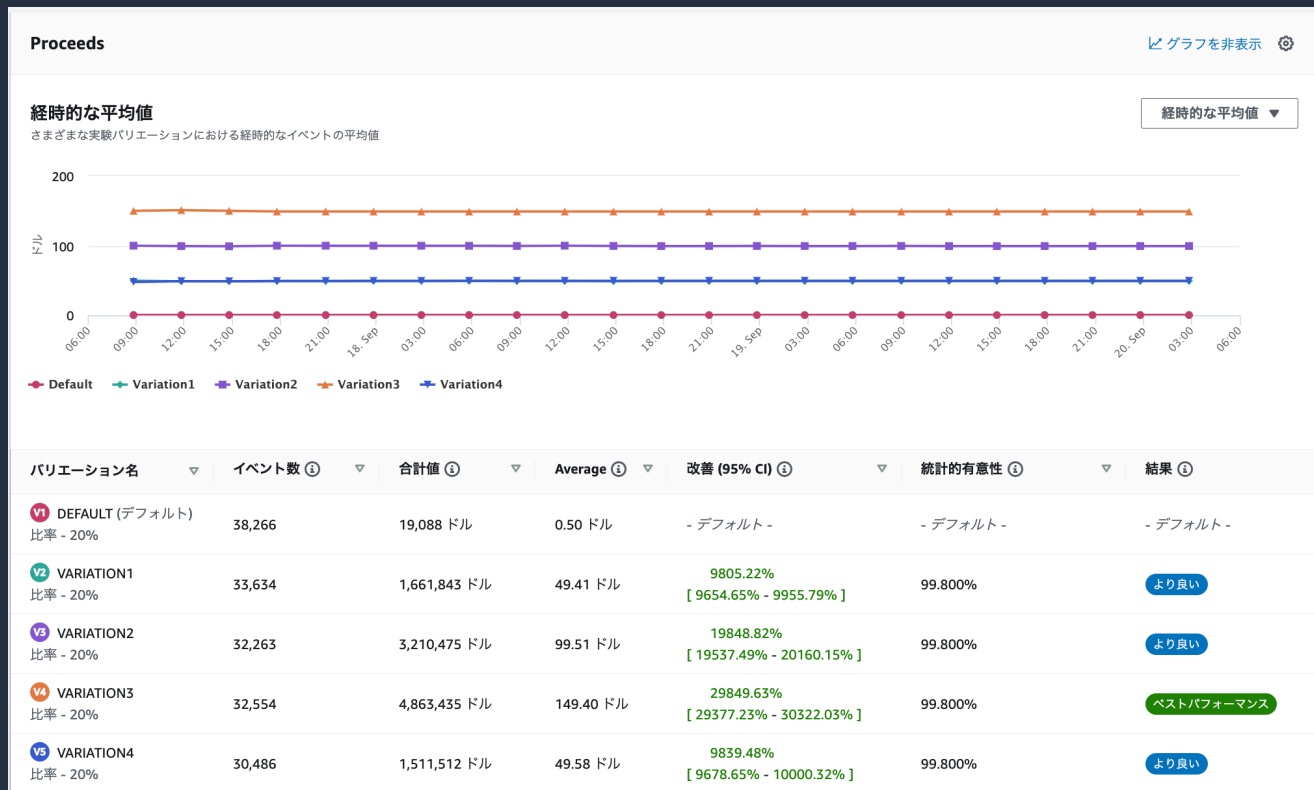
終了時間 (JST) 09:50

期間: 7 日、0 時間、6 分

キャンセル 実験を開始



# Amazon CloudWatch Evidently – 実験結果



実験の進行中および完了後に、実験の統計結果を確認できます。

実験結果は、実験の開始から 63 日後まで確認できます。

各バリエーションに少なくとも 100 個のイベントが発生するまで、統計結果は表示されません。

Evidently では、実験の最後に追加のオフライン p 値分析が実行されます。

オフライン p 値分析では、実験中に使用された p 値では統計的有意性が得られない場合に、統計的有意性を検出できます。

列	説明
イベント数	各バリエーションで記録されたユーザーセッションイベントの数
合計値	これまでのメトリクス値の合計
Average	合計値/これまでのイベント数 (またはこれまでの合計値とイベント数の比率)

# Amazon CloudWatch Evidently – 実験結果 (推論タイプ)



統計的有意性とベイズ推論を使用することで、観測された差が有意であるか、それとも偶然によるものなのかを知ることができます。

Evidently は次の点を考慮して、観測された差が有意であるかどうかを判断します。

- 差はどの程度か
- テストに含まれるサンプル数
- データの配布方法

# Amazon CloudWatch Evidently – 実験結果（統計的有意性）

Evidently は逐次テストを使用するため、頻度論的統計の一般的な落とし穴であるピーキングの問題を回避することができます。そのため、結果に既に有意な意味がある場合は、予定時間前に実験を中止することができます。結果ページには次の情報が表示されます。

列	説明
改善（95% CI）	選択したメトリクスについての各バリエーションとデフォルトのバリエーションとの差の 95% 信頼区間の上限および下限。信頼区間の上限と下限間のギャップが小さいほど、推定値の信頼性が高い。
統計的有意性	選択したメトリクスについて、デフォルトのバリエーションと他の各バリエーションとの差の比較。デフォルトとバリエーションの間に観測された差がランダム性によるものではないことがどの程度確実かを測定する。
結果	起動決定を支援するため、頻度分析に基づいて以下の結果の推奨を行います。 <ol style="list-style-type: none"><li>1. 決定的ではない（統計的有意性が 95% 未満）</li><li>2. より良い（デフォルトよりも良い）</li><li>3. より悪い（デフォルトよりも悪い）</li><li>4. ベストパフォーマンス</li></ol>

# Amazon CloudWatch Evidently – 実験結果（ベイズ推論）

ベイズ推論では、テスト済みのバリエーションの平均がデフォルトのバリエーションの平均よりも大きい小さいかの確率を計算することができます。Evidently は実験終了日まで待機してから、ベイズ推論の結果を計算します。結果ページには次の情報が表示されます。

列	説明
増加確率	テスト済みのバリエーションにおけるメトリクスの平均が、デフォルトのバリエーションの平均よりも少なくとも 3% 大きい確率
減少確率	テスト済みのバリエーションにおけるメトリクスの平均が、デフォルトのバリエーションの平均よりも少なくとも 3% 小さい確率
変化がない確率	テスト済みのバリエーションにおけるメトリクスの平均が、デフォルトのバリエーションの平均の $\pm 3\%$ 以内にある確率
結果	ローンチ決定を支援するため、ベイズ分析に基づいて、以下の結果の推奨を行います。 条件を満たさない場合、結果は表示されません。 1. より良い（デフォルトよりも良い） 2. より悪い（デフォルトよりも悪い）

# Amazon CloudWatch Evidently – 起動を作成

実験結果から有益な情報を把握できた場合、有益だった機能についてローンチ（起動）することができます。

## 実験: VariationTest

実験をキャンセル      実験トラフィックを変更      アクション ▲

<p>セットアップ 機能: <a href="#">VariationFeature</a>  2 個のメトリクス、5 個のバリエーション</p>	<p>39% completed</p>	<p>スケジュール Sep 17, 2023 - Sep 24, 2023 残り 4 / 6 日</p>	<p>アラームの実験 アラームなし セグメント セグメントなし</p>	<p>ステータス <b>● LIVE</b></p>	<p>実験を管理 実験を開始 実験を編集 実験を削除</p> <p>その他 <b>起動を作成</b></p>
--	--------------------------	--	---	--------------------------------	--

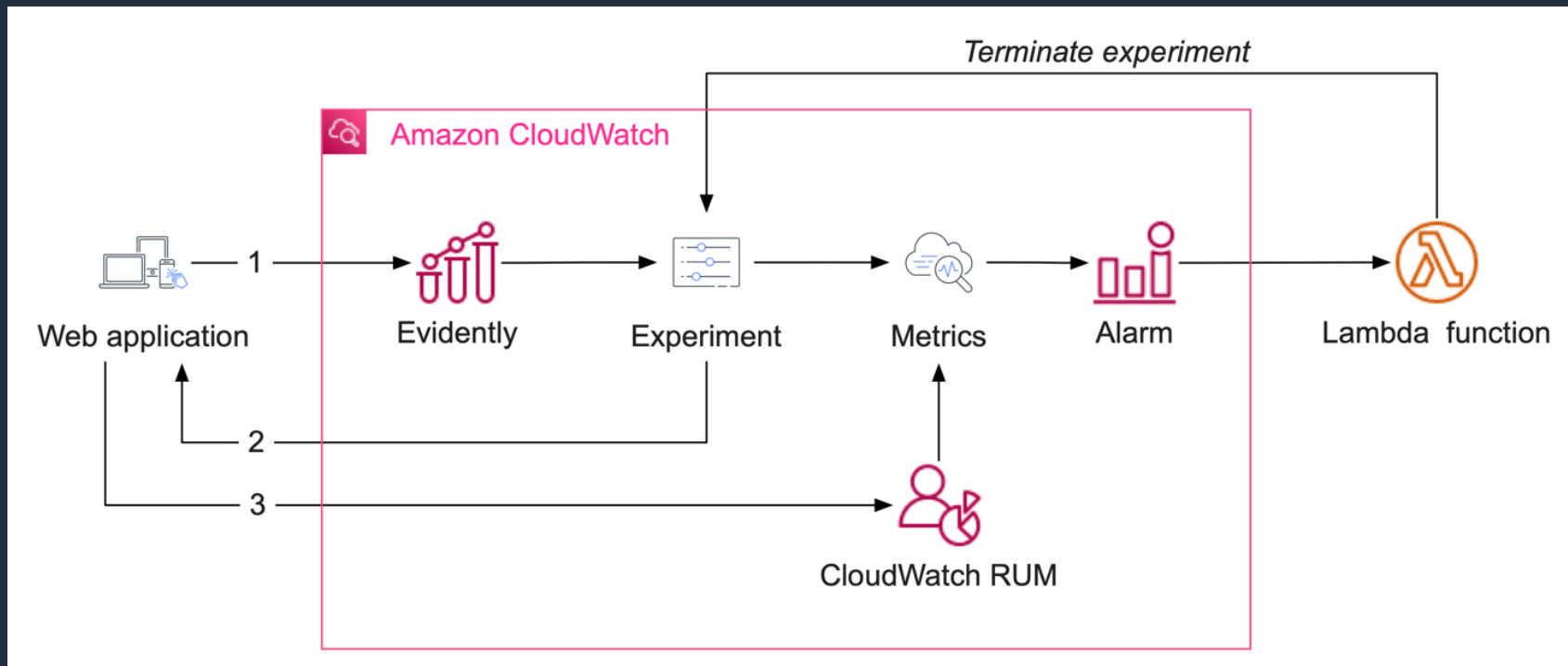
# Amazon CloudWatch Evidently の 連携サービス

# CloudWatch Evidently と CloudWatch RUM

Amazon CloudWatch RUM は、ウェブアプリケーションのエクスペリエンスの特定、理解、および改善に役立つインサイトを提供するメトリクスの収集をサポートします。

CloudWatch Evidently はこのアプリケーションをモニターリングするように RUM をセットアップしている場合に使用できます。

Amazon CloudWatch RUM についての詳細は [Amazon CloudWatch RUM 【AWS Black Belt】](#) をご覧ください。





# CloudWatch Evidently と CloudWatch RUM

## CloudWatch Evidently と CloudWatch RUM の連携における注意事項

- RUM アプリケーションモニター をセットアップして、拡張メトリクス、カスタムメトリクス、またはその両方を CloudWatch Evidently に送信することができます。
  - **カスタムメトリクス** – カスタムメトリクスはユーザーが定義するメトリクスです。カスタムメトリクスは、CloudWatch と CloudWatch Evidently の両方に送信できます。
  - **拡張メトリクス** – デフォルトの CloudWatch RUM 拡張メトリクスを CloudWatch Evidently に送信し、Evidently で使用できます。
- CloudWatch RUM メトリクス を CloudWatch Evidently に連携するためには、CloudWatch RUM イベントの **sessionId** を EvaluateFeature API の **entityId** パラメータ として使用する必要があります。



# CloudWatch Evidently で 利用可能な RUM メトリクス

- CloudWatch Evidently で 利用可能な デフォルト の CloudWatch RUM 拡張メトリクス

イベントタイプ	メトリクス名
ウェブバイタル	累積レイアウトシフトイベント
	最初の入力遅延イベント
	最大のコンテンツフルペイントイベント
パフォーマンス リソースイベント	Duration
	Transfer Size
パフォーマンス ナビゲーション イベント	Duration
	Unload Event Start
	Prompt For Unload
	Redirect Start
	Redirect Time
	Worker Start
	Worker Time
	Domain Lookup Start
Dns	

イベントタイプ	メトリクス名
パフォーマンス ナビゲーション イベント	Connect Start
	Connect
	Secure Connection Start
	Tls Time
	Request Start
	Time To First Byte
	Response Start
	Response Time
	Dom Interactive
	Dom Processing Time
	Dom Content Loaded Event Start
	Dom Content Loaded
	Load Event Start
	Load Event Time

# CloudWatch Evidently で RUM カスタムメトリクスを利用

CloudWatch RUM カスタムメトリクスを Evidently に送信するには、コンソールではなく、AWS API または AWS CLI を使用する必要があります。

1. [PutRumMetricsDestination](#) API ([put-rum-metrics-destination](#) CLI コマンド) で CloudWatch Evidently への送信先(Destination) を作成
2. [BatchCreateRumMetricDefinitions](#) API ([batch-create-rum-metric-definitions](#) CLI コマンド) で CloudWatch Evidently への送信する カスタムメトリクス を作成
3. [PutRumEvents](#) API ([recordEvent](#) スクリプト) でカスタムイベントを CloudWatch RUM へ送信
4. CloudWatch Evidently でカスタムメトリクスを指定

The screenshot displays the AWS Evidently console interface. On the left, a table lists metrics, with 'PageVisit' selected. The 'メトリクスルール' (Metric Rule) section is expanded, showing a configuration for a custom metric. The configuration is as follows:

メトリクス (1)	メトリクスソース	詳細	メトリクスルール
メトリクス 1 PageVisit 目標: 増加	CloudWatch RUM アプリケーションモニター: RUMAppMonitor	イベントタイプ: Custom Metrics ページ ID: /evidently 単位: Count	ルールを表示

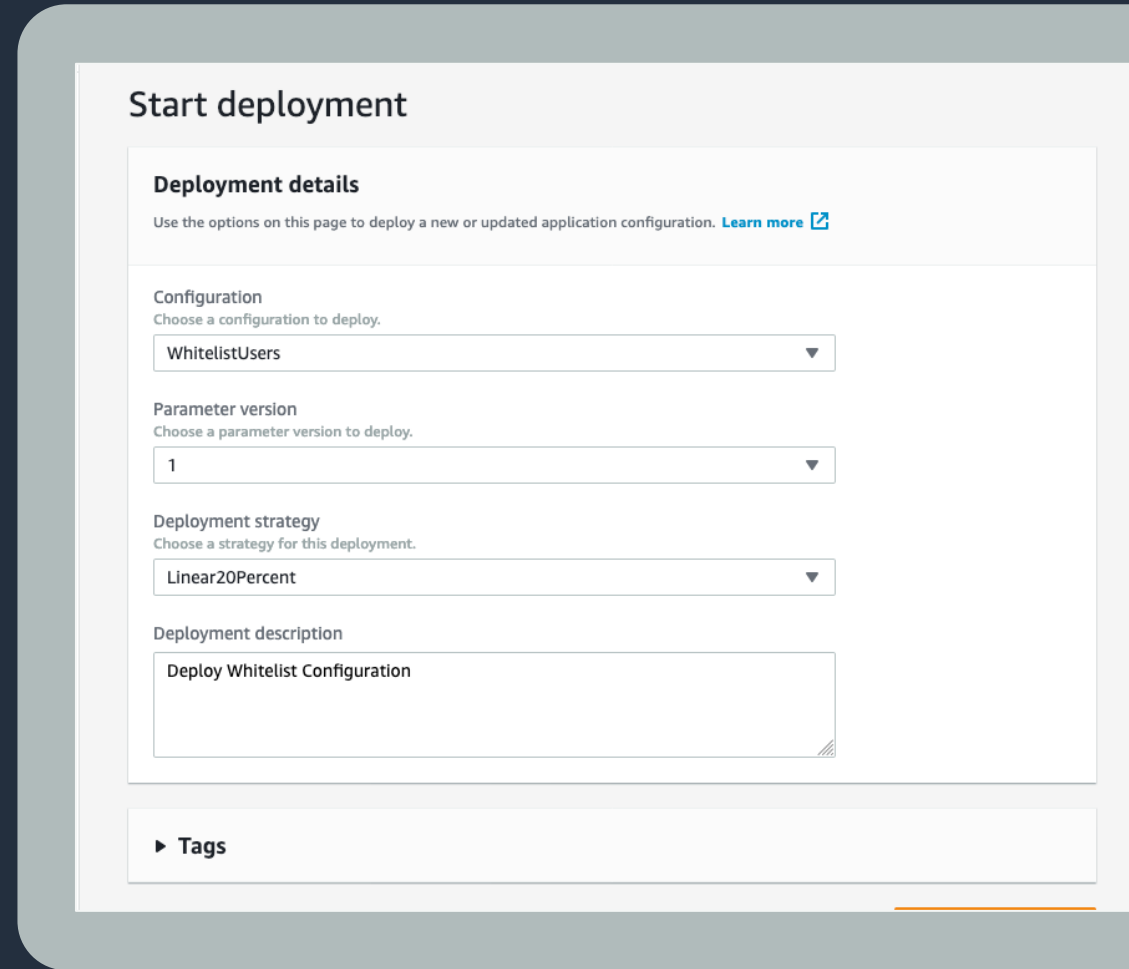
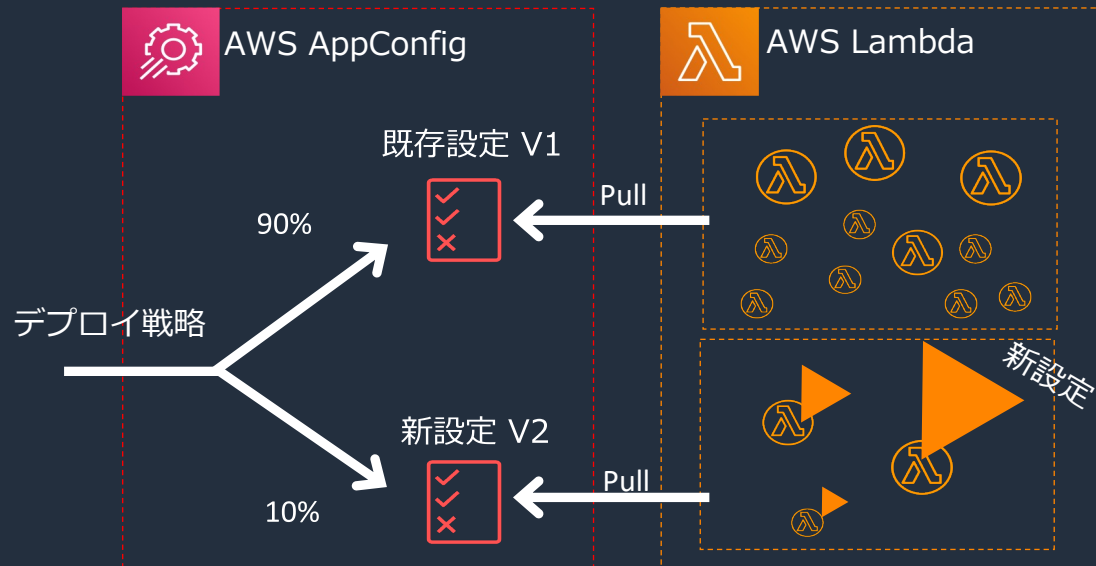
```
1 {
2   "entityIdKey": "$.user_details.sessionId",
3   "valueKey": "$.event_details.visit_count",
4   "eventPattern": {
5     "application_name": [
6       "AWSRailsSample"
7     ],
8     "page_id": [
9       "/evidently"
10    ],
11    "event_type": [
12      "my_custom_event.rum.custom_metrics"
13    ],
14    "metric_unit": [
15      "Count"
16    ]
17  }
18 }
```

完了

# CloudWatch Evidently と AWS AppConfig の連携

## AppConfig - アプリケーション構成を作成、管理、デプロイ

- アプリケーションを実行したまま、アプリケーション構成の変更をデプロイ。
- 製品の発表など、タイムリーな展開が必要な新機能の展開を、機能フラグで実現。
- デプロイ戦略を指定でき、少しずつデプロイが可能。



# CloudWatch Evidently と AWS AppConfig の主な違い

## AWS AppConfig

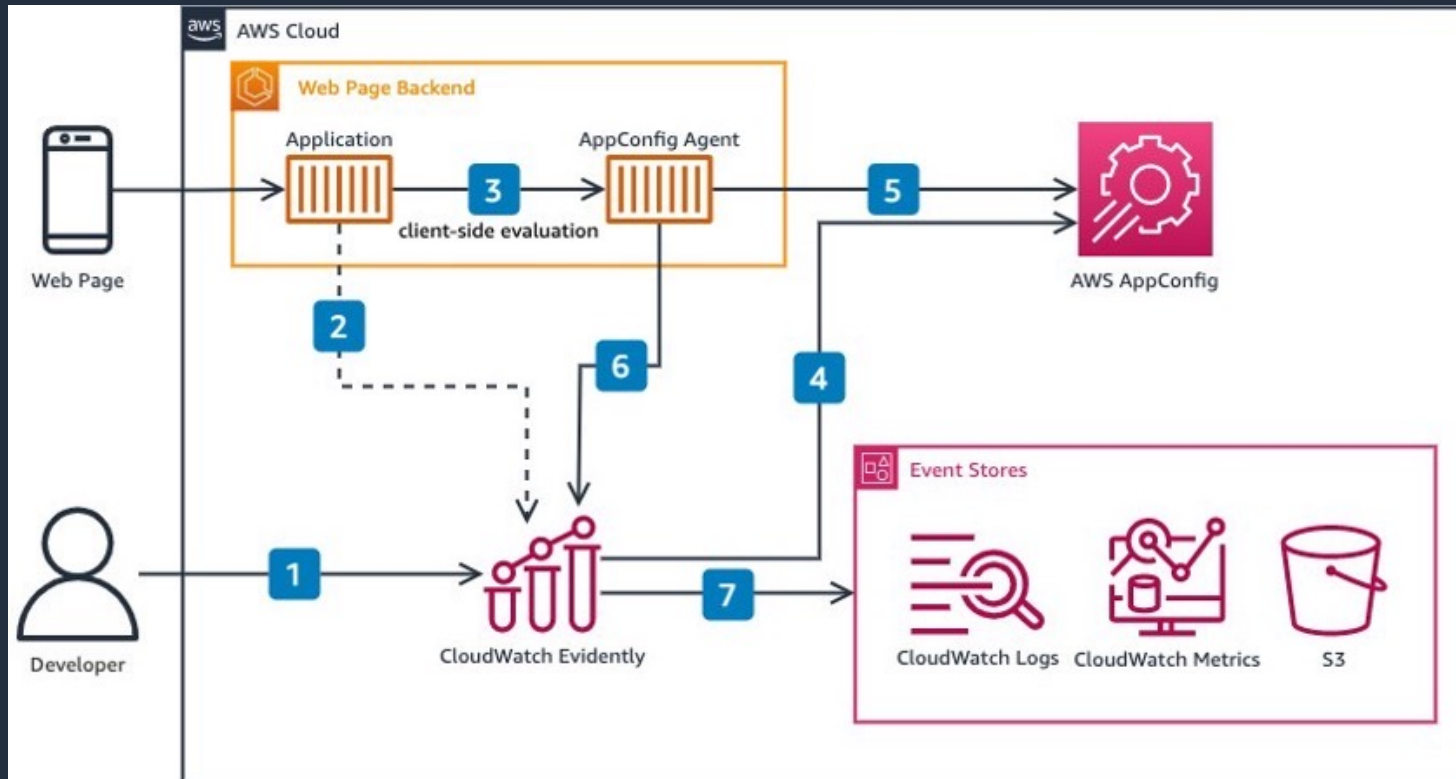
構成の管理  
様々な拡張機能の提供

## CloudWatch Evidently

ユーザー体験モニタリング  
実験(A/Bテスト)の評価

# CloudWatch Evidently と AWS AppConfig の連携

AWS AppConfig を使用 (クライアント側の評価) を使用すると、都度、EvaluateFeature API を呼び出すのではなく、ローカルでバリエーションをユーザーセッションに割り当てることができます。これにより、API コールに伴うレイテンシーと可用性のリスクが軽減されます。より詳しい説明については、[Amazon CloudWatch Evidently による Client-Side Evaluation の紹介](#) をご覧ください。



# Amazon CloudWatch Evidently の注意事項、料金

# Amazon CloudWatch Evidently の注意事項と制限事項

- 利用可能リージョン (2023/11月時点)

- 米国東部 (バージニア北部)
- 米国東部 (オハイオ)
- 米国西部 (オレゴン)
- 欧州 (フランクフルト)
- 欧州 (ストックホルム)
- 欧州 (アイルランド)
- **アジアパシフィック (東京)**
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)

**大阪リージョンは未対応  
(2023/11月時点)**

# Amazon CloudWatch Evidently の注意事項と制限事項

クォータ	デフォルトのクォータ	引き上げリクエスト可能か
プロジェクト数	リージョンごと、アカウントごとに 50	Yes
セグメント数	リージョンごと、アカウントごとに 500	Yes
プロジェクトごとのクォータ	<ul style="list-style-type: none"> <li>機能は合計 100</li> <li>起動は合計 500</li> <li>実行する起動は 50</li> <li>実験は 合計 500</li> <li>実行する実験は 50</li> </ul>	Yes
API クォータ (リージョンごとのすべてのクォータ)	<ul style="list-style-type: none"> <li>PutProjectEvents:                             <ul style="list-style-type: none"> <li>米国東部 (バージニア北部)、米国西部 (オレゴン)、欧州 (アイルランド) では 1000 TPS (1 秒あたりのトランザクション件数)</li> <li><u>その他すべてのリージョンでは 200 TPS</u></li> </ul> </li> <li>EvaluateFeature:                             <ul style="list-style-type: none"> <li>米国東部 (バージニア北部)、米国西部 (オレゴン)、欧州 (アイルランド) では 1000 TPS</li> <li><u>その他すべてのリージョンでは 200 TPS</u></li> </ul> </li> <li>BatchEvaluateFeature: 50 TPS</li> <li>Create、Read、Update、Delete (CRUD) の API: CRUD API 全体で 10 TPS</li> </ul>	Yes

[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/monitoring/CloudWatch-Evidently-quotas.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch-Evidently-quotas.html)



# VPC エンドポイント経由でデータを送受信する場合

**CloudWatch Evidently は専用の VPC エンドポイント (PrivateLink) が用意 (VPCエンドポイントポリシーに対応)**

The screenshot shows the AWS IAM console search results for VPC endpoints. The search filters are set to 'サービス名 = com.amazonaws.ap-northeast-1.evidently' and 'サービス名 = com.amazonaws.ap-northeast-1.evidently-dataplane'. The results table shows two entries, both owned by 'amazon' and of type 'Interface'. The first entry is 'com.amazonaws.ap-northeast-1.evidently' and the second is 'com.amazonaws.ap-northeast-1.evidently-dataplane'. A red dashed box highlights these two entries.

サービス名	所有者	タイプ
○ com.amazonaws.ap-northeast-1.evidently	amazon	Interface
○ com.amazonaws.ap-northeast-1.evidently-dataplane	amazon	Interface

サービス名	説明
com.amazonaws.<region>.evidently	CloudWatch Evidently コントロールプレーンの VPCエンドポイント
com.amazonaws.<region>.evidently-dataplane	CloudWatch Evidently データプレーンの VPCエンドポイント

# Amazon CloudWatch Evidently の料金

クライアントから送信されるイベント数に基づく従量課金 2023年11月時点での利用料金(全リージョン同一)

## Amazon CloudWatch Evidently 料金

### Evidently イベント

100,000イベント  
あたり 5.00 USD / 月

Evidently イベントは、クリックやページビューなどのユーザーアクションから発生する データイベント と、ユーザーに提供する 機能のバリエーション を決定する 割り当てイベント があります。

+

### Evidently 分析ユニット

100 万分析ユニット  
あたり 7.50 USD / 月

Evidently 分析ユニットは、Evidently で作成したルールに基づいて、Evidently イベントから生成されます。つまり、イベントでルールが一致した数です。

+

### (関連サービス利用料)

CloudWatch  
(Metrics/Alarm/...)

+

Cognito

+

AppConfig

+

⋮

<https://aws.amazon.com/jp/cloudwatch/pricing/>



# まとめ

# まとめ

Amazon CloudWatch Evidently を使用して、新機能のパフォーマンスをモニターリングすることで、新機能を完全にローンチする前に、リスクを軽減し、意図しない結果を素早く解消することができます。

また、Amazon CloudWatch Evidently でA/B テストを実行すると、統計的手法を用いて分析し、どのバリエーションが優れているかについて、明確なレコメンデーションを手に入れることができます。

より詳しく学びたい方は、One Observability Workshop もご覧ください。

<https://catalog.workshops.aws/observability/ja-JP/aws-native/app-monitoring/evidently>

# 本資料に関するお問い合わせ・ご感想

技術的な内容に関しましては、有料のAWSサポート窓口へお問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想はTwitterへ！ハッシュタグは以下をご利用ください  
#awsblackbelt

# その他コンテンツのご紹介

ウェビナーなど、AWSのイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWSのソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!