

## CloudWatch Evidently ワークショップ

2022/07/01

シニアエバンジェリスト 亀田

すべての作業はオレゴンで行ってください。

1. IAM のマネージメントコンソールに移動します
2. 左ペインから[ユーザー]をクリックします
3. [ユーザーを追加]をおします
4. ユーザー名に[evidently]と入力します
5. [アクセスキー - プログラムによるアクセス]にチェックをつけます
6. [次のステップ:アクセス権限]を押します
7. [既存のポリシーを直接アタッチ]を選び[AmazonCloudWatchEvidentlyFullAccess]を選択します

### ▼ アクセス許可の設定

ポリシー名	タイプ	次として使用
<input checked="" type="checkbox"/> AmazonCloudWatchEvidentlyFullAccess	AWS による管理	なし

8. [次のステップ:タグ]を押します
9. [次のステップ:確認]、[ユーザーの作成]と順に押します
10. csv をダウンロードしておきます
11. Cloud9 のマネージメントコンソールに移動します
12. [Create environment]をおします
13. 適当な名前をつけ、全てデフォルトのまま[Next step]をおします。VPC はデフォルトが自動で選択されていますが、存在しない場合、パブリックサブネットがある VPC を指定してください。
14. [Create environment]をおします
15. しばらく待つとコンソールが利用可能となりますので、以下のコマンドを実行します  
nvm install 14.17.5
16. 以下のコマンドを入力します  
wget <https://evidently-sample-application.s3.us-west->

[2.amazonaws.com/evidently-sample-shopping-app.zip](https://2.amazonaws.com/evidently-sample-shopping-app.zip)

17. Zip を解凍します

```
unzip evidently-sample-shopping-app.zip
```

18. /src/App.jsx の中身を git 上の App.jsx に置き換えます。(Cloud9 上で開いてコピーしたのち保存です)

この Web サイトではアクセスするたびに 50%の確率で、20%オフセール中であることと、割引後の価格が表示されます。ユーザーがオリジナルサイト、値引き後サイト双方で操作を行うたび、以下の値が Evidently に送られます。

```
[{"timestamp": 1637368646.468, "type": "aws.evidently.custom", "data":
{"details":{"pageLoadTime":2058.002058},"userDetails":{"userId":"1637368644430","sessionId":"1637368644430"}}}]
```

これにより、値引き前と後でユーザーの行動の違いを知るためのデータが蓄積されます。

19. [config.js]を開き、先程作成した IAM ユーザーのクレデンシャルを入力し保存します

```
credential: {
  accessKeyId: "ACCESS_KEY_ID_HERE",
  secretAccessKey: "SECRET_ACCESS_KEY_HERE"
}
```

20. ブラウザ別タブで CloudWatch マネージメントコンソールへ移動します

21. 左ペインから Evidently をクリックします

#### ▼ アプリケーションのモニタリング

ServiceLens マップ

リソースの正常性

Synthetics Canaries

Evidently

RUM

22. [プロジェクトを作成]をおします

23. 名前に[EvidentlySampleApp]と入力します

24. 以下で評価イベントを保存しないように設定します

**評価イベントストレージ 情報**

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。Evidently 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 で表示できます。

25. [プロジェクトの作成]をおします

26. 以下の出来上がったプロジェクトをクリックします

プロジェクト (1)					プロジェクトアクション	プロジェクトを作成
<input type="text" value="プロジェクト名または値でプロジェクトをフィルタリング"/>					< 1 >	⚙️
プロジェクト名	実験	機能	起動	作成日		
<input type="radio"/> EvidentlySampleApp	0	0	0	Jul 01 2022 10:28:41 AM JST		

27. [機能を追加]をおします

28. 名前に[showDiscount]と設定します

29. [機能のバリエーション]を以下のようにセットします

**バリエーションタイプ 情報**

ルール値

バリエーション名	値	情報	デフォルト	情報	
<input type="text" value="v1 disable"/>	<input checked="" type="checkbox"/> 偽		<input checked="" type="radio"/>		<input type="button" value="削除"/>
<input type="text" value="v2 enable"/>	<input checked="" type="checkbox"/> 真		<input type="radio"/>		<input type="button" value="削除"/>

最大でさらに 3 個のバリエーションを追加できます

30. サンプルコードでコードスニペットが出てきます。(この手順ではすでに App.jsx に記載済ですので省略します)

31. [機能を追加]をおします

32. [実験]のタブを選択し[実験の作成]をおします

33. 実験名に[pageLoadTime]と入力します

34. 機能が以下の通り設定されていることを確認します

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

showDiscount ▼

バリエーションタイプ

ブール値

バリエーション

V1 disable

デフォルト

値

false

V2 enable

値

true

対象者 情報

バリエーションをテストする対象を定義します。

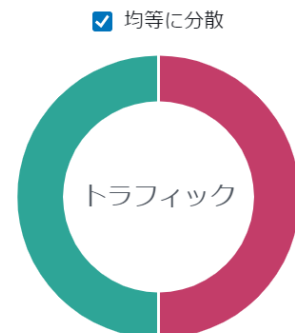
実験のトラフィック分割  
実験に参加する、利用可能な訪問者の割合

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	トラフィック比率	
V1 disable デフォルト	50 %	<input checked="" type="checkbox"/> 含む
V2 enable	50 %	<input checked="" type="checkbox"/> 含む



35. メトリクス項目のメトリクス名に[pageLoadTime]と入力します
36. 目標を[減少]に設定します。これは実験の目標値として、少ない方がいいことを表します。このテストでは、20%値引きがユーザーに与える影響と組み合わせページ読み込み時間が与える影響も取得します
37. 以下の情報をそれぞれセットします
  - エンティティ ID : UserDetails.userId
  - 値キー : details.pageLoadTime
  - 単位 : ms

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 ▼

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.userId details.pageLoadTime ms

38. [実験を作成]をおします

39. 右上ドロップダウンから[実験を開始]を選びます

**実験: pageLoadTime** Cancel experiment 実験トラフィックを変更 アクション ▲

セットアップ 機能: <a href="#">showDiscount</a> 1 メトリクス、2 バリエーション	 スケジュール まだ開始されていません <a href="#">Start experiment</a>	アラームの実験 アラームなし	ステータス ③ 作成済み
---	--	-------------------	-----------------

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

40. [実験を開始]をおします。以下の通り実験が開始されます

**実験: pageLoadTime** Cancel experiment 実験トラフィックを変更 アクション ▼

セットアップ 機能: <a href="#">showDiscount</a> 1 メトリクス、2 バリエーション	 スケジュール Jul 01, 2022 - Jul 08, 2022 残り 6 / 6 日	アラームの実験 アラームなし	ステータス <b>LIVE</b>
---	---	-------------------	----------------------

41. Cloud9 コンソールに戻り以下のコマンドを順に実行します

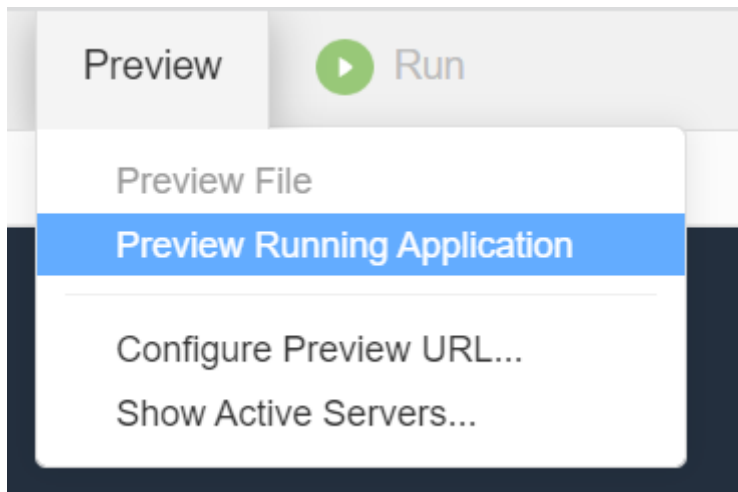
```
cd evidently-sample-shopping-app
```

```
npm install -f && npm start
```

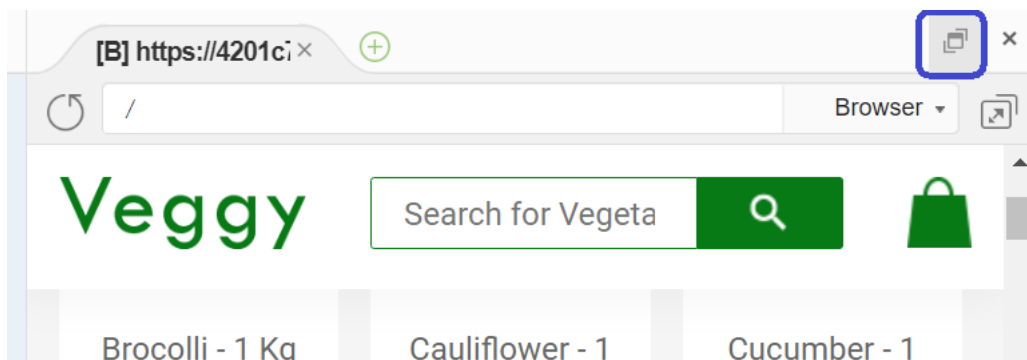
以下が表示されれば起動は完了です

```
Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.
```

42. Cloud9 上部メニューバーで[Preview]→[Preview Running Application]をおします



43. ページが表示されます。青枠の部分をクリックすると画面表示が大きくなります



44. ブラウザのリロードを繰り返すと値引き後、値引き前が両方表示されます。(通常はユーザーごとに固定となりますが、このサンプルでは意図的に同じユーザーでもテストでいるようにしています) カートに品物を入れ購入などを行うと以下のデータが Evidently でできます。(それぞれ値引き前、値引き後最低 1 回ずつ購入の最後まで行ってください。ユーザーは適当なパスワードとメールアドレスでログインできますので、新規作成は不要です)

経時的なイベントのカウント

さまざまな実験バリエーションの経時的なイベント数

経時的なイベントのカウント ▼



バリエーション名 ▼	イベント数 ⓘ ▼	合計値 ⓘ ▼	Average ⓘ ▼	改善 (95% CI) ⓘ ▼
<b>v1</b> DISABLE (デフォルト) 比率 - 50%	1	158 ms	0 ms	- デフォルト -
<b>v2</b> ENABLE 比率 - 50%	2	655 ms	0 ms	- N/A -