Amazon QuickSight RoadShow Deep Dive③

データセットパラメータでクエリーを最適化し、CloudWatch で測定

坂下　和香奈

QuickSight Solutions Architect
アマゾンウェブサービスジャパン合同会社

Agenda

- パラメータとは
- データセットパラメータ機能
- ユースケース
- デモ

1) データセットパラメータを使用してデータセット・分析を作成
2) CloudWatchによるダッシュボードロード時間の確認

- 制限事項

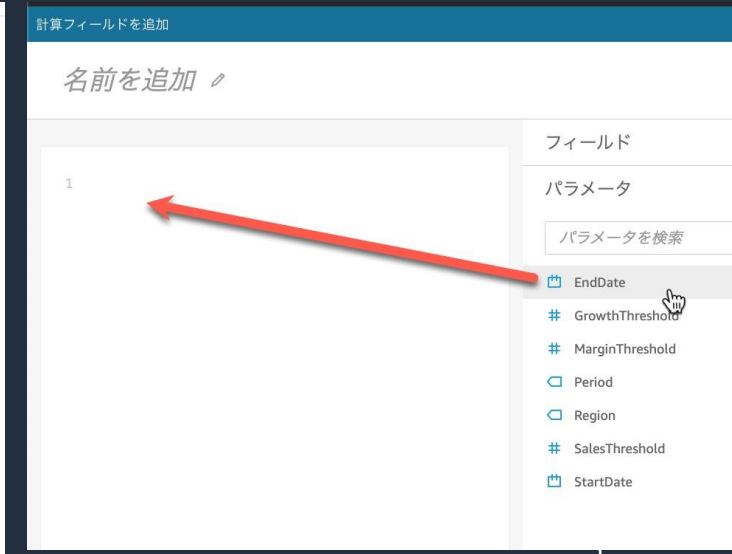
パラメータとは

- アクションまたはオブジェクトにより使用される値を転送できる名前付きの変数
- インタクティブなダッシュボードの作成を可能に
- パラメータを使用するには、以下で参照設定可

  - コントロール、フィルタ、アクション

  - 計算フィールド、ナラティブ、タイトル/サブタイトル内

- ダッシュボード間の連携にも利用可



データセットパラメータ機能とは

ダイレクトクエリーで大規模なデータを扱う場合、動的にスライスされたデータのみを取得するようにクエリーを最適化し、データソース側のスキャン量を制御したい

- 特に多くのテーブルをJoinしたクエリー等において、クエリーを最適化でき、フィルター制御のパフォーマンス向上が期待できる
- データセットの所有者がデータセット内にパラメータを作成。それを分析のフィルターコントロールで利用できる

データセットパタメータ使用方法

データセットレベルで、データセット所有者によって、データ準備画面で作成

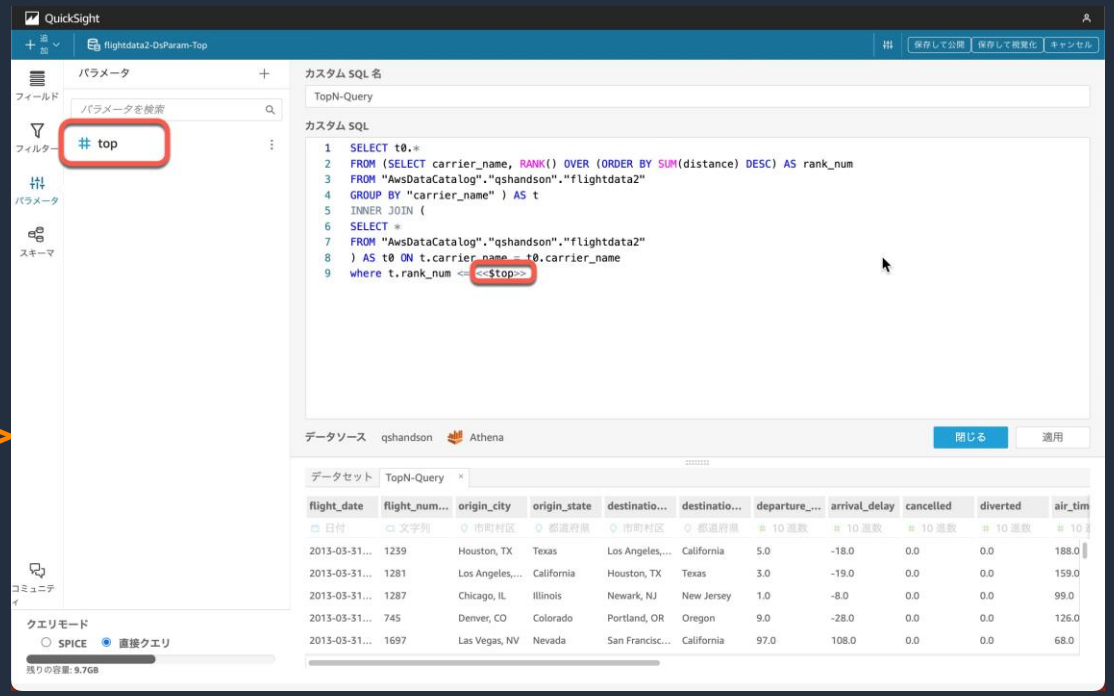
パラメータの使用場所：

1）ダイレクトクエリーモードでのカスタムSQL内

単一値パラメータ：SELECT * FROM all_flights
WHERE origin_state = **<<$State>>**

複数値パラメータ: SELECT * FROM all_flights
WHERE origin_state in（**<<$States>>**

2）ダイレクトクエリー ／SPICEデータセットの
計算フィールド内



分析のパラメータ値をデータセットのパラメータにマッピングすることにより、
可視化時にクエリー内のパラメータを置き換えることができる。

データセットパラメータ使用例①
＜ユーザ選択によるベーシックフィルター＞

```
SELECT *
FROM transactions
WHERE region = <<$RegionName>>
```

データセット所有者は、Where句をカスタムSQLに加える。それにより、Readerが選択した地域（リージョン）によりデータはフィルターされ、スキャンデータ量は大きく軽減される

パラメータは単一文字列として作成されているが、複数値も対応可。

上記のようなシンプルなカスタムSQLをダイレクトクエリー実施する場合は、カスタムSQL＋データセットパタメータではなく、テーブル指定のデータセットを作成し、分析上で地域（リージョン）にてフィルターした方が望ましい（＝QuickSightが同じクエリーを内部で生成しており、効果はあまり期待できない＜詳細は後述＞）

データセットパラメータ使用例②
＜ case when または ifelse で使用 ＞

```
SELECT Region, Country, date, sum(sales)
FROM transactions
WHERE region=
    (Case
    WHEN <<$UserFIRSTNAME>> In
        (select firstname from user where region='region1')
        and <<$UserLASTNAME>> In
        (select lastname from user where region='region1')
        THEN 'region1'
    WHEN <<$UserFIRSTNAME>> In
        (select firstname from user where region='region2')
        and <<$UserLASTNAME>> In
        (select lastname from user where region='region2')
        THEN 'region2'
    ELSE 'region3'
    END)
```

データセット所有者は、２つのパラメータ（名前と苗字）を作成。Readerがダッシュボード上で選択した２つの文字列が、casewhen句で2回参照される

データセットパラメータ使用例③
＜Select句で新しい列を作成＞

```
SELECT Region, Country, date,
        (case
        WHEN <<$RegionName>>='EU'
        THEN sum(sales) * 0.93   --convert US dollar to euro
        WHEN <<$RegionName>>='CAN'
        THEN sum(sales) * 0.78   --convert US dollar to Canadian Dollar
        ELSE sum(sales) -- US dollar
        END
        ) as "Sales"
FROM transactions
WHERE region = <<$RegionName>>
```

ダッシュボード上で選択された地域（リージョン）名に従い、Salesという列が新たに作成される。
地域により、CASE WHEN句を使用して現地の通貨で売上を再計算

ユースケース

1) ダイレクトクエリーのカスタムクエリーを最適化

クエリー作成の自由度が上がるため、データソース側の機能を引き出しやすくなり、より柔軟な処理の実現や、パフォーマンス改善につながります。

2)　分析で利用するデータセットを汎用化

分析で利用しているパラメータを使用した計算フィールドを、データセットレベルに移すことができるようになり、データセットを汎用化することができ、データセットの管理も軽減されます。

3) パラメータの繰り返し利用によるデータセットのメンテナンスの簡素化

パラメータをカスタムクエリーだけでなく、計算フィールドでも利用することにより、メンテナンスを簡素化することができます。

デモ

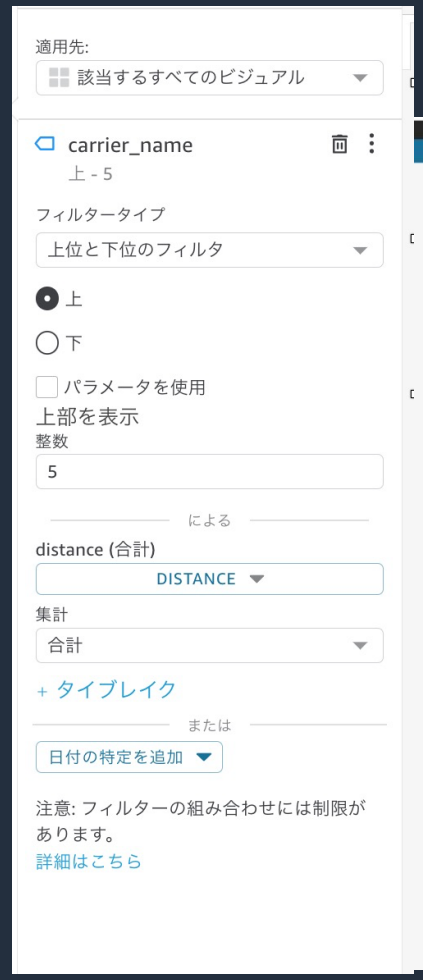
デモ

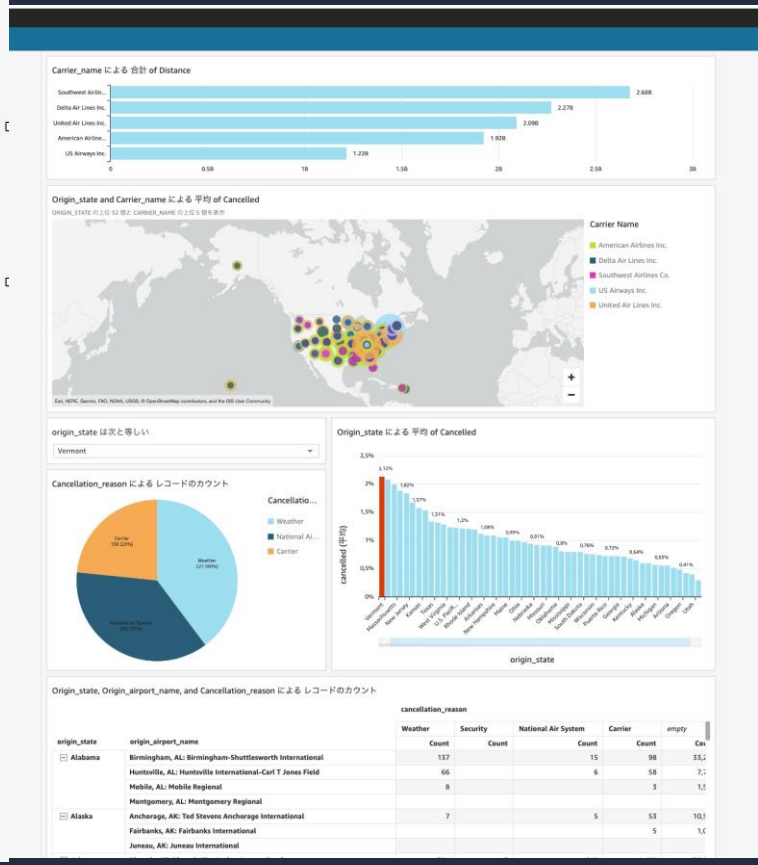
- 米国交通統計局（ https://www.bts.gov/ ）が公開している米国国内線の発着データ(2010-2014)、約2,000万件
- S3 via Athena

シナリオ:

- 航続距離上位5位の航空会社でフィルター
- ダッシュボードを作成済

上位数をパラメータ定義し、カスタムSQLにて上位フィルターしたデータセットを作成





デモ① データセットパラメータを使用した分析を作成

1. 上位数のデータセットパラメータ（整数値）を作成（＄Top）
2. データセットパラメータを使用したカスタムSQL作成

```
SELECT t0.*
FROM (
    SELECT carrier_name, RANK() OVER (ORDER BY SUM(distance) DESC) AS rank_num
    FROM "AwsDataCatalog"."default"."flightdata2"
    GROUP BY "carrier_name" ) AS t
INNER JOIN (
    SELECT *
    FROM "AwsDataCatalog"."default"."flightdata2"
    ) AS t0 ON t.carrier_name = t0.carrier_name
where t.rank_num <= <<$top>>
```

距離数合計で航空会社をランク付け

JOINにて, 必要なデータ項目を取得

Where句で、選択された上位ランクのみを取得

デモ① データセットパラメータを使用して分析を作成

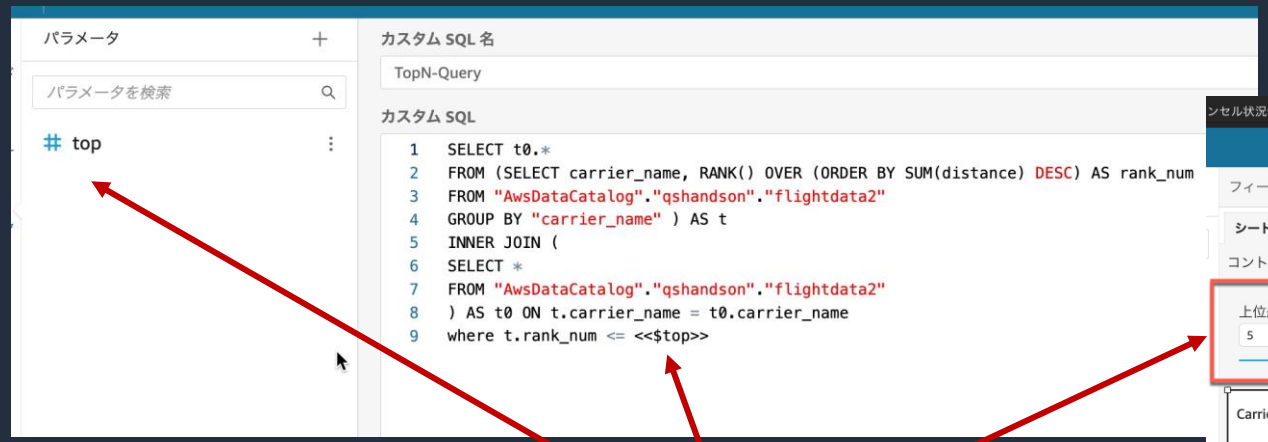
3. 分析画面で、作成したデータセットパラメータを、分析のパラメータにマッピング

   - データセットパラメータがunmappedパラメータとして表示される

   - マッピングする新しいパラメータを作成し、スライダーコントロールを作成

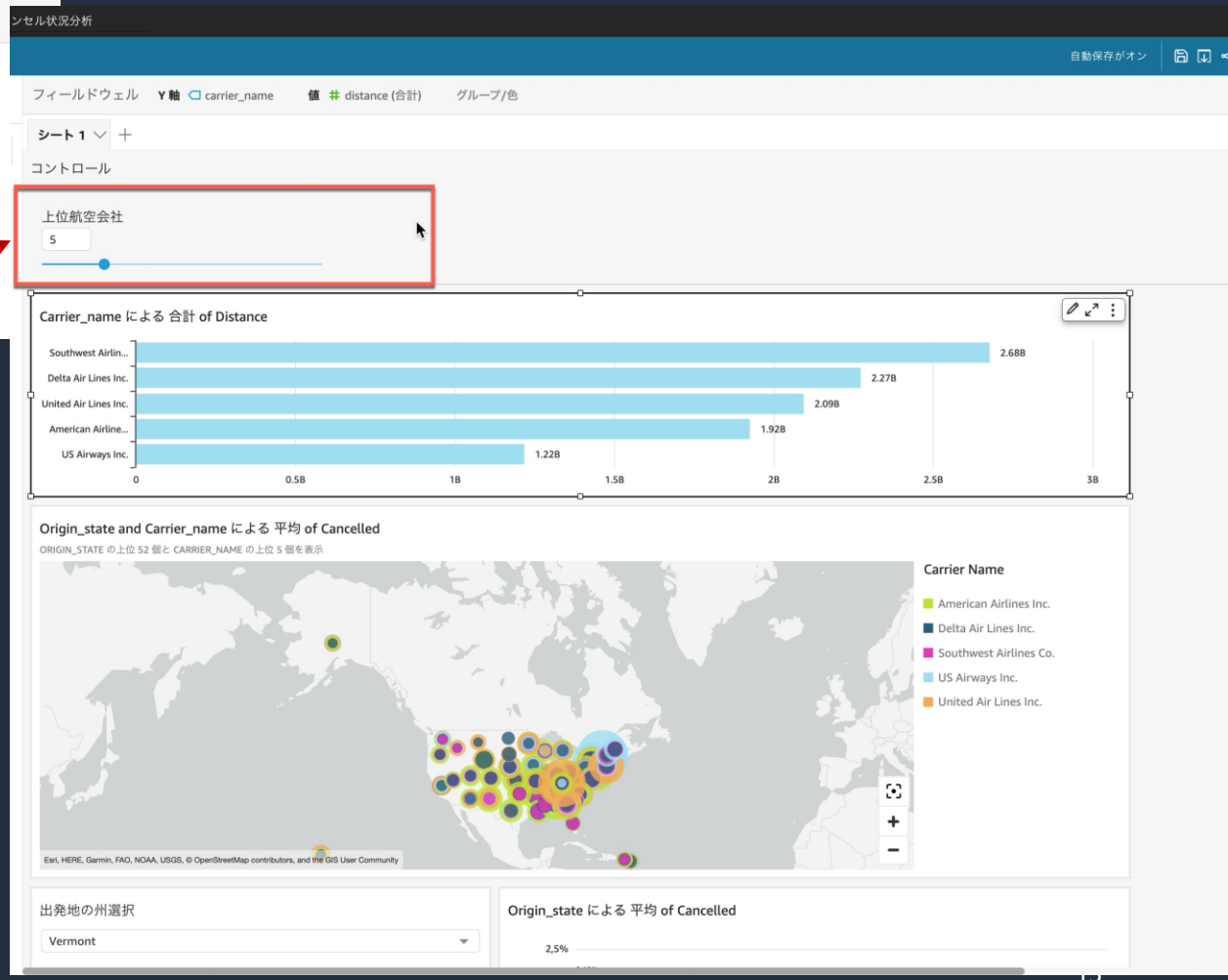
➡ スライダー上で選択した値が、カスタムSQLに渡される

データセットパラメータ＋カスタムSQLで、航続距離数上位5位の航空会社でフィルターしたダッシュボード



データセット

分析



データソース(Athena）へのクエリを確認

```
    COUNT(*) AS "count", DENSE_RANK() OVER (ORDER BY "origin_state" NULLS FIRST,
    "origin_airport_name" NULLS FIRST) AS "$RANK_1", DENSE_RANK() OVER (ORDER BY
    "cancellation_reason" DESC) AS "$RANK_2"
8   FROM (SELECT t0.*
9   FROM (SELECT carrier_name, RANK() OVER (ORDER BY SUM(distance) DESC) AS rank_num
10  FROM "AwsDataCatalog"."default"."flightdata2"
11  GROUP BY "carrier_name" ) AS t
12  INNER JOIN (
13  SELECT *
14  FROM "AwsDataCatalog"."default"."flightdata2"
15  ) AS t0 ON t.carrier_name = t0.carrier_name
16  where t.rank_num <= 5) AS "TopN-Query"
17  GROUP BY "origin_state", "origin_airport_name", "cancellation_reason"
18  ) AS "t"
19  WHERE "$RANK_1" <= 500 AND
```

作成したカスタムクエリー(TopN-Query)が、サブクエリー化され、ビジュアルのクエリーのFROMで利用

上位パラメータの値がデフォルト値に置き換わっている

データソース(Athena）へのクエリを確認

分析フィルターを使用した場合

```
3  SELECT "carrier_name" AS "$f0e87223-03a8-4e64-93c1-ed6191df4011.carrier_name_join_1", RANK() OVER
       (ORDER BY SUM("distance") DESC) AS "$RANK_3"
4  FROM "AwsDataCatalog"."default"."flightdata2"
5  GROUP BY "carrier_name"
6  ) AS "t"
7  INNER JOIN (
8  SELECT "origin_state" AS "f0e87223-03a8-4e64-93c1-ed6191df4011.origin_state", "cancelled" AS "f0e87223
       -03a8-4e64-93c1-ed6191df4011.cancelled", "carrier_name" AS "f0e87223-03a8-4e64-93c1-ed6191df4011
       .carrier_name"
9  FROM "AwsDataCatalog"."default"."flightdata2"
10 ) AS "t0" ON "t"."$f0e87223-03a8-4e64-93c1-ed6191df4011.carrier_name_join_1" = "t0"."f0e87223-03a8
       -4e64-93c1-ed6191df4011.carrier_name" OR "t"."$f0e87223-03a8-4e64-93c1-ed6191df4011
       .carrier_name_join_1" IS NULL AND "t0"."f0e87223-03a8-4e64-93c1-ed6191df4011.carrier_name" IS NULL
11 WHERE "t"."$RANK_3" <= 5
12 GROUP BY "t0"."f0e87223-03a8-4e64-93c1-ed6191df4011.origin_state", "t0"."f0e87223-03a8-4e64-93c1
```

サブクエリー化されていないが、基本同じクエリーが生成されている

デモ② CloudWatchでダッシュボードのロード時間を確認する

- CloudWatchにて、QuickSightのメトリックス（SPICEへの取り込み状況、ダッシュボードやビジュアルのパフォーマンス）を参照可能
- 特定のダッシュボードやビジュアル単位にメトリックスを指定し、CloudWatch上でアラーム設定をし監視することができる。

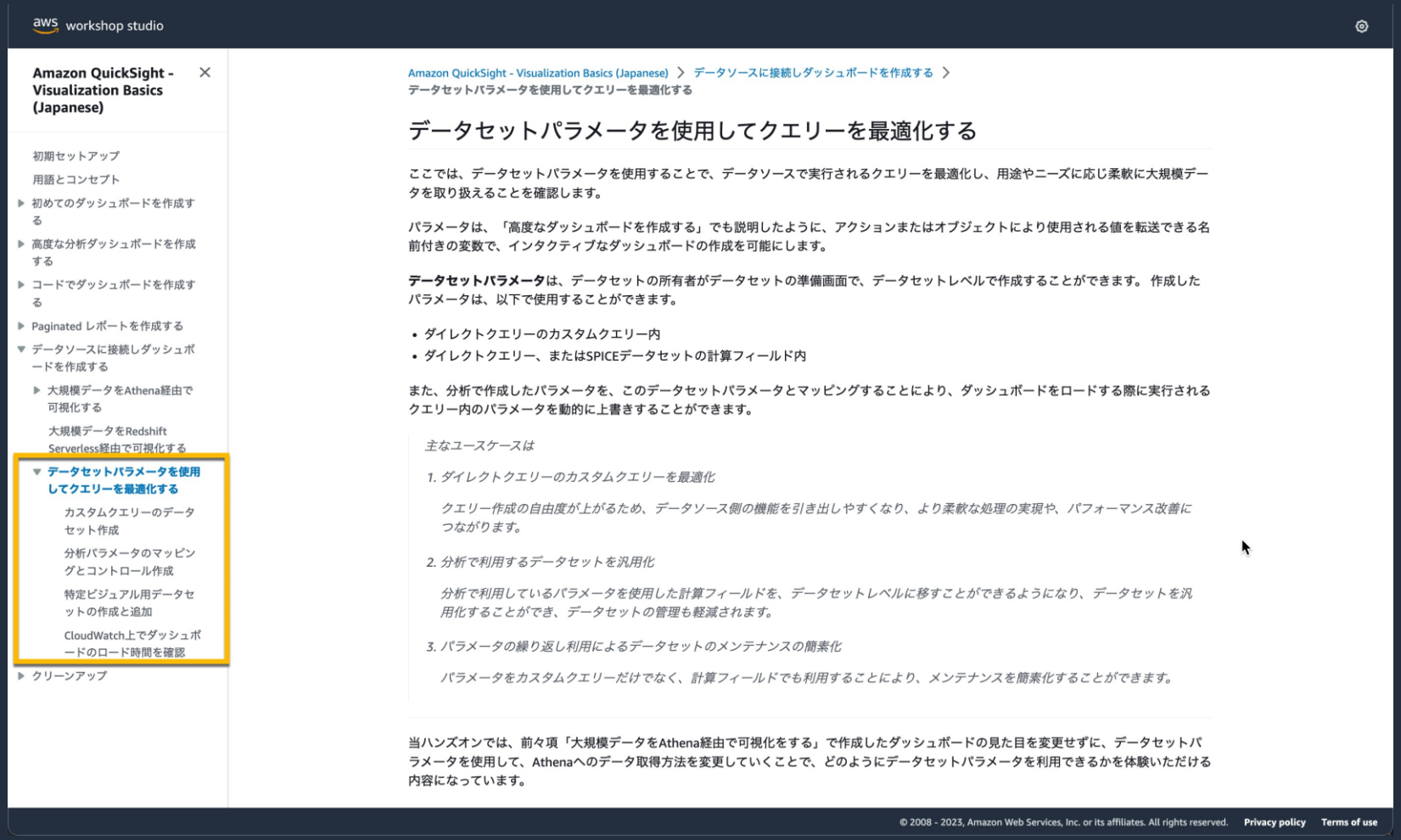
1. 公開した２種類のダッシュボードから、ダッシュボードIDを取得
2. CloudWatch上で、取得したダッシュボードIDの LoadDashbaordTimeのメトリックスを選択し、グラフを追加

データセットパラメータの制限事項

- SPICEデータセットでは利用できない
- 動的デフォルト値設定は、分析のパラメータでのみ利用可能
- データセットパラメータをマッピングした分析のパラメータで 複数値選択のコントロールを使用可能だが、「すべて選択」オプションは利用できない
- データセットパラメータを使用したカスケーディングコントロールはサポート外
- データセットパラメータは分析のフィルターでは使用できず、データセット上のフィルターで使用可（ダイレクトクエリーのみ）
- Eメールでの定期配信をスケジュールするとき、コントロールの選択値がデータセットパラメータの値（フィルター値）としてレポートには反映されない。パラメータのデフォルト値がレポートでは使用される

Call to Action

https://catalog.us-east-1.prod.workshops.aws/workshops/aa601d0b-84c9-4f77-b9a7-5954d8574cd5/ja-JP/7-connecting-to-data-via-direct-query/use-dataset-paramter-feature



Thank you!