



# AWSのテクノロジーを用いた データの利活用について

アマゾン ウェブ サービス ジャパン

中島 丈博

2022/3/24

# 自己紹介



名前：中島 丈博（なかじま たけひろ）

所属：アマゾン ウェブ サービス ジャパン  
ソリューションアーキテクト

役割：製薬業界のお客様を中心に  
クラウド利用に関する技術支援

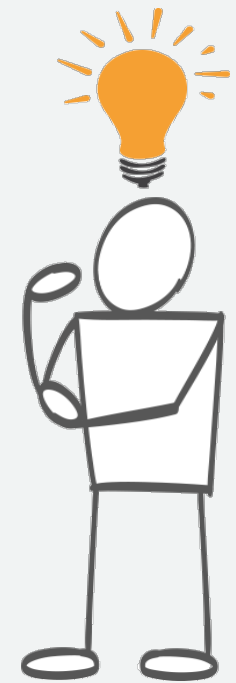
- ・ご要望の具現化
- ・課題の解決に向けたご支援
- ・製薬業界のクラウド利用事例のご紹介

# アジェンダ

- コマーシャル領域における課題
- データ利活用のためのプラットフォーム
  - データレイク
  - データの加工・変換 (ETL)
  - 分析
  - 可視化
  - 機械学習 (ML)
  - エンゲージ
- まとめ

# このセッションにおける想定参加者について

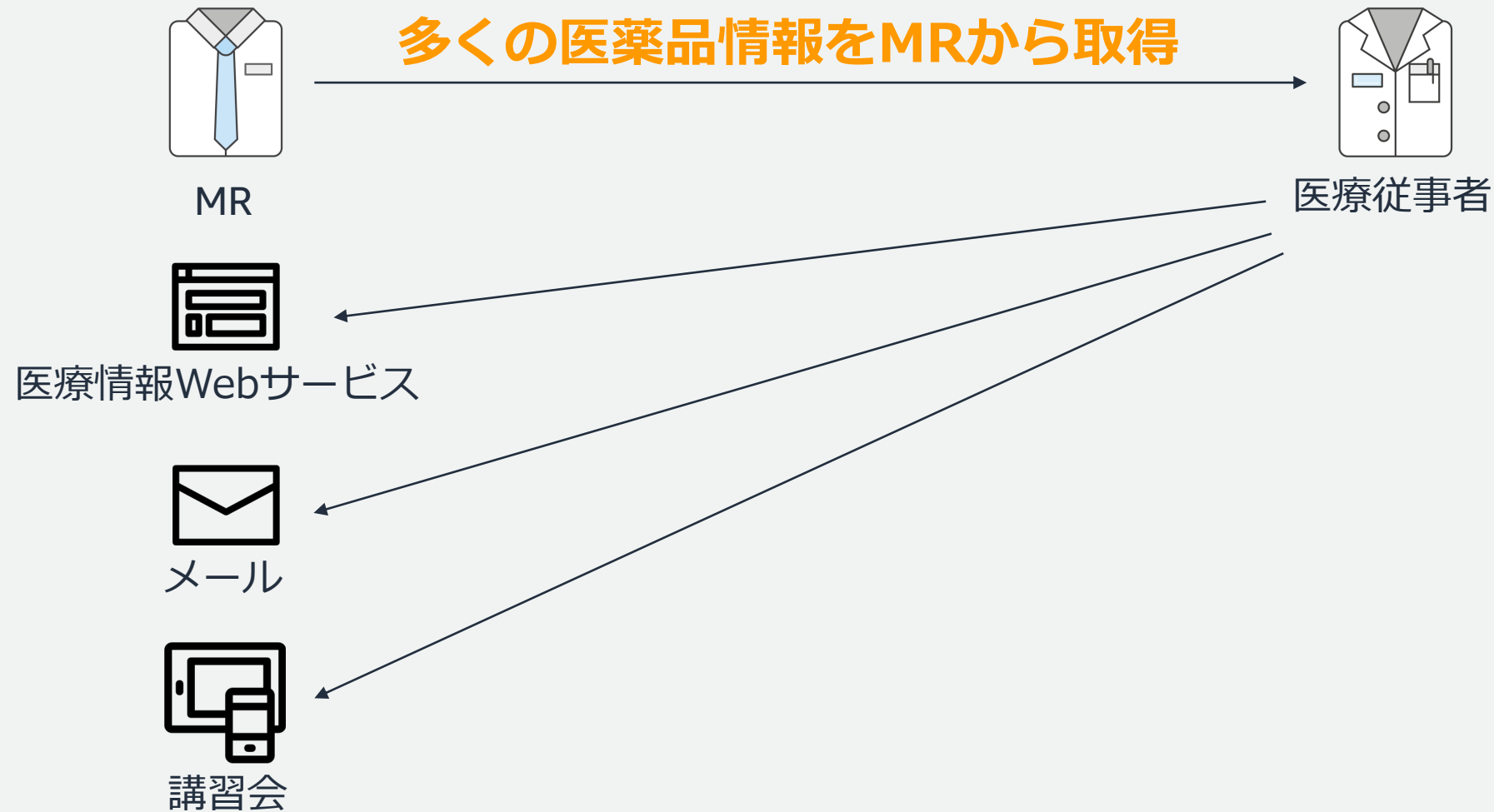
本セミナーは製薬業界の商業領域に関連するお客様が対象であることから、各サービスの機能をどう活用できるかについて、サマリ形式でお伝え致します



# コマース領域における課題

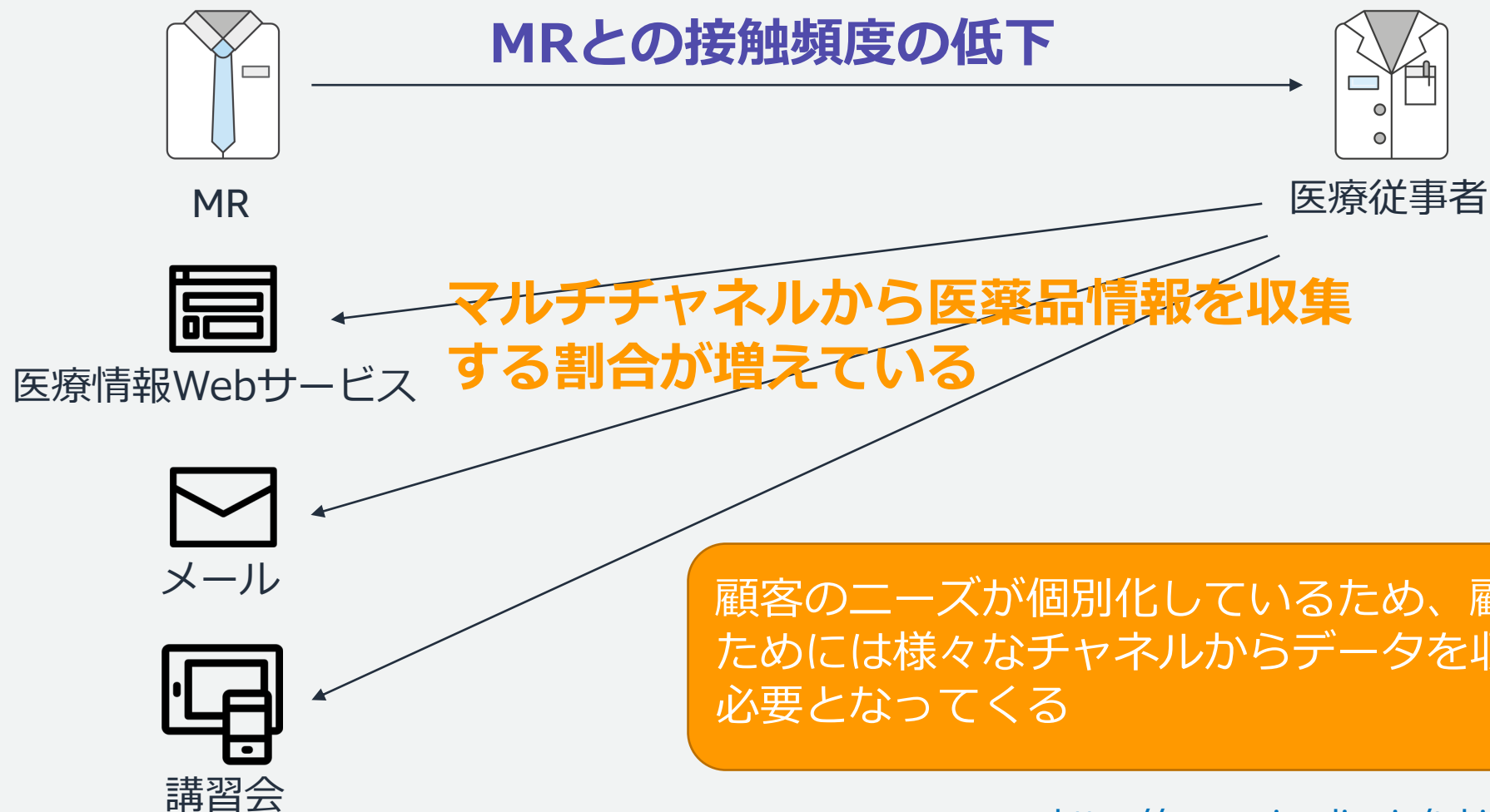
# コマース領域における課題

これまでの顧客（医療従事者）の情報ソース



# コマース領域における課題

現在の顧客（医療従事者）の情報ソース



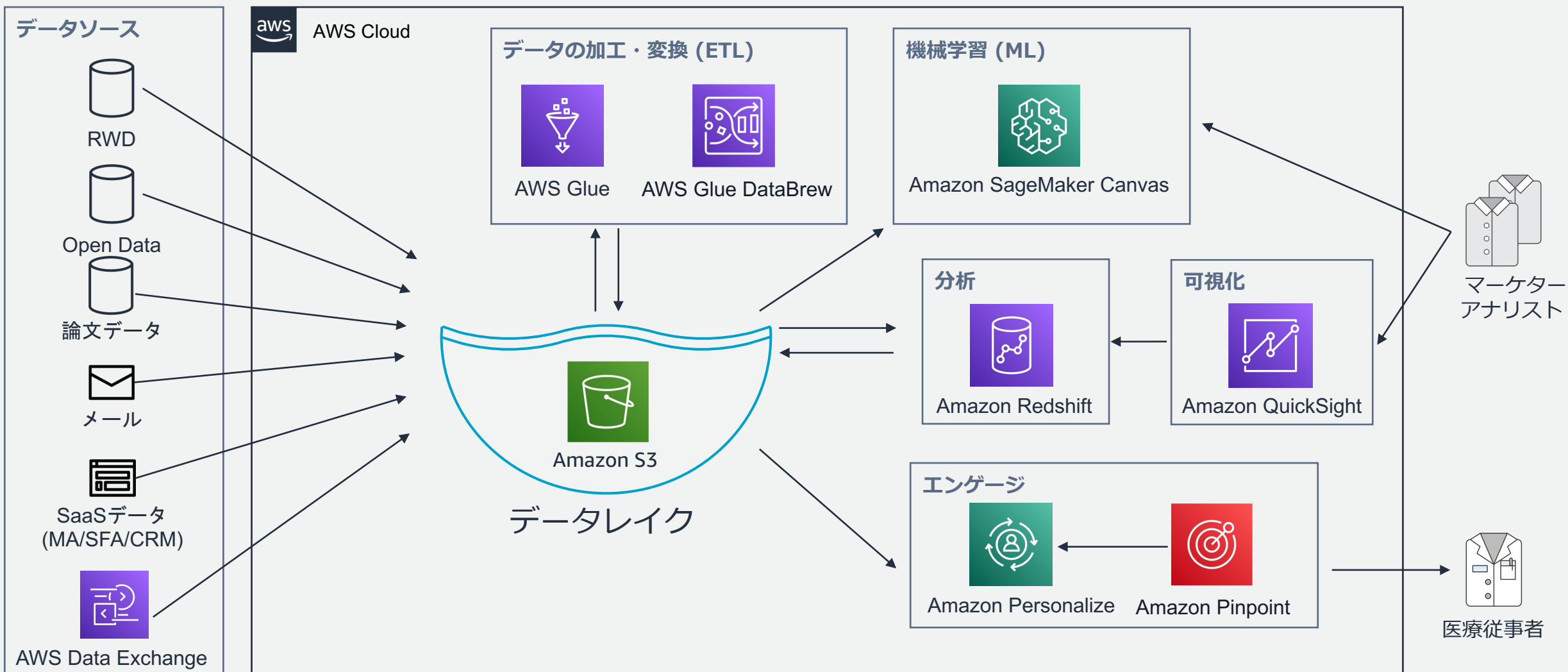
<https://www.mixonline.jp/tabid55.html?artid=70512>

# コマース領域における課題

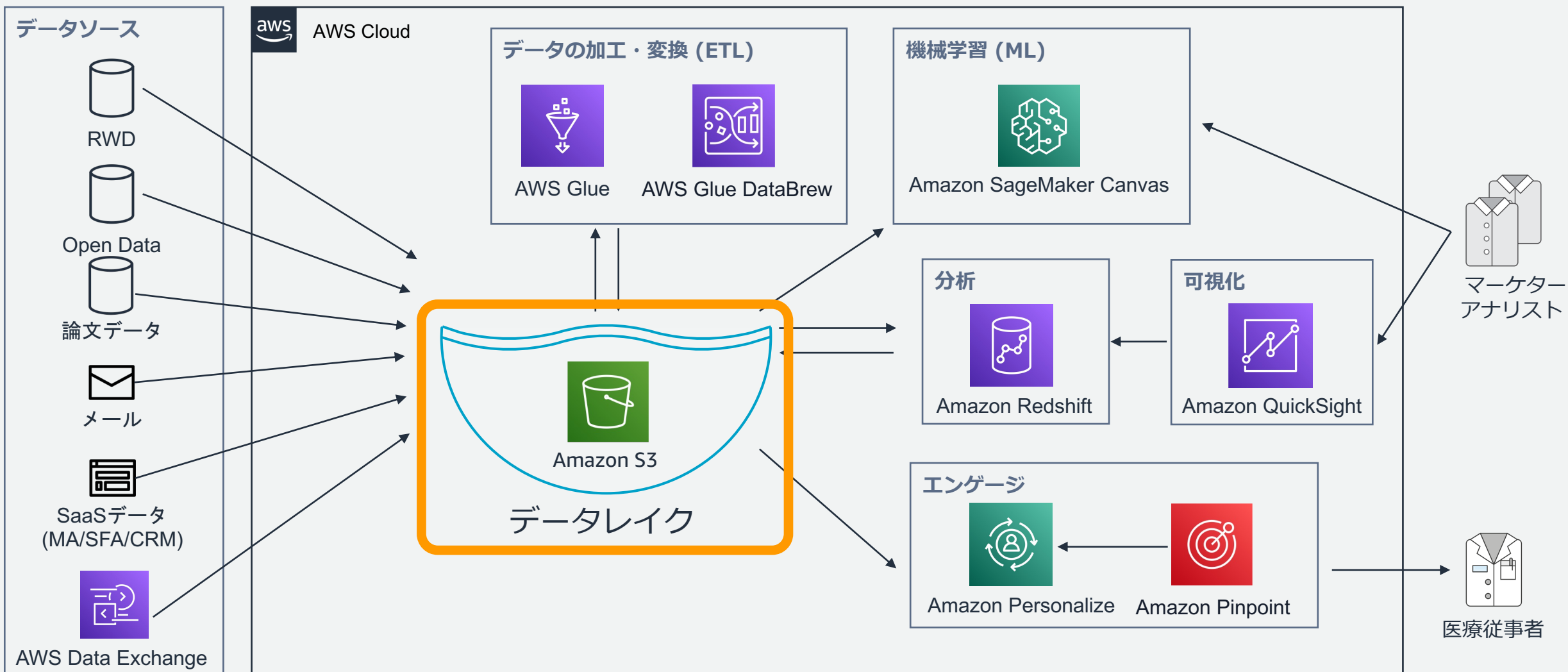
現在の顧客（医療従事者）の情報ソース



# コマース領域での利用を想定したデータレイク構成案



# コマース領域での利用を想定したデータレイク構成案



# データ利活用のためのプラットフォーム データレイク

# データレイクとは

すべてのデータをそのままの形式で一箇所に  
集めて保存可能なリポジトリ（貯蔵庫）



# なぜデータレイクが必要なのか

## 一般的なデータ分析の流れ



様々なデータを目的に応じて加工・整理したデータをデータウェアハウスに格納してBIツール等で可視化する

→この分析方法には課題も存在する

# 一般的なデータ分析の課題を料理に例えると…

- データウェアハウスは目的に合わせた**加工済のデータ**を保存  
例) 西京焼きのため、魚をさばいて、味噌に漬け込んだ形で保存してある



# データウェアハウスの課題を料理に例えると…

- データウェアハウスは目的に合わせた**加工済のデータ**を保存  
例) 西京焼きのため、魚をさばいて、味噌に漬け込んだ形で保存してある



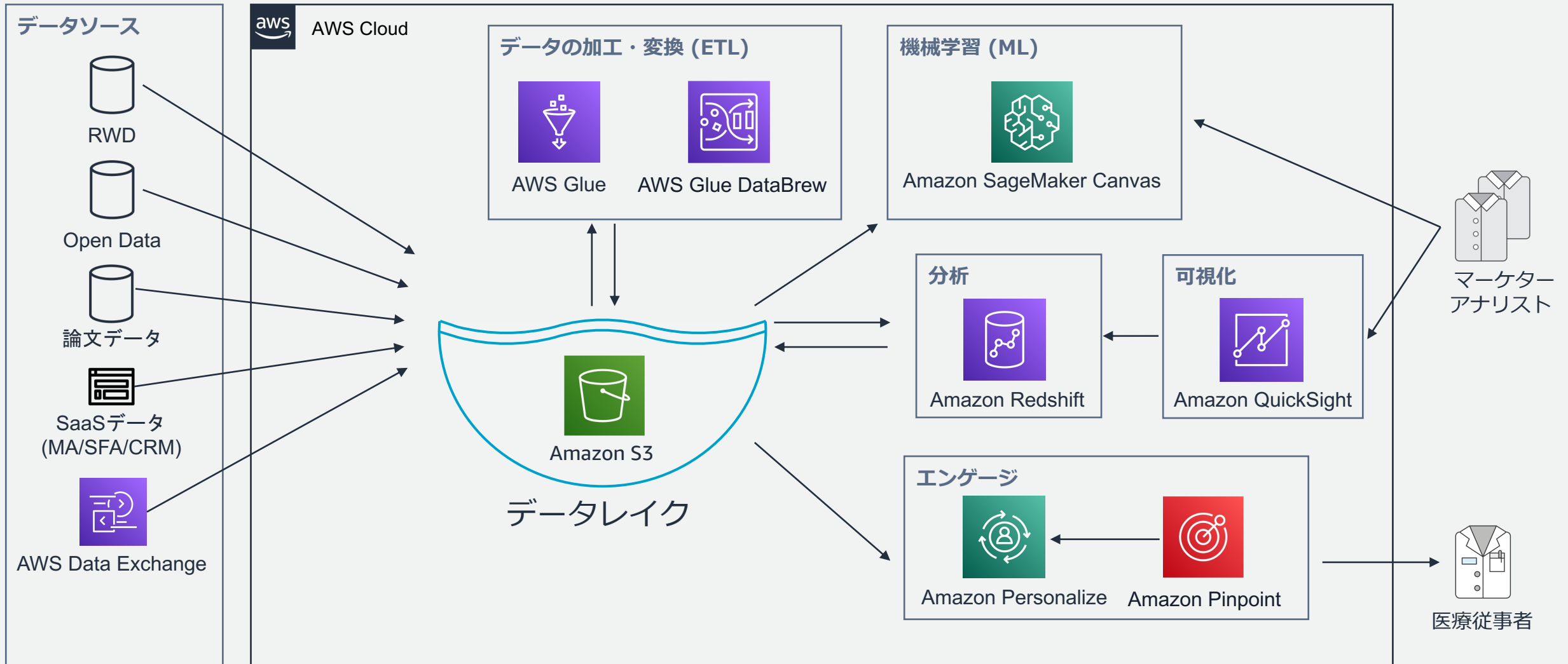
後から刺し身を作りたくなっても、生魚は加工済のため刺し身にすることが出来ない

# データレイクによる解決

- データを「生」のまま保存し、将来のニーズに備える
- 加工したデータもデータレイクに保存
- 分析処理はデータウェアハウス等、データレイクの外で実現



# コマース領域での利用を想定したデータレイク構成案



# Amazon S3

- Amazon Simple Storage Service (S3)は、ユーザがデータを安全に、容量制限なく、データ保存が可能な、クラウド時代のオブジェクトストレージ
- Amazon S3 Glacier Deep Archiveは、安全性とコスト効率を重視したアーカイブ向けストレージ



# S3 ストレージクラス (2021/11/29まで)



S3 標準



S3 Intelligent-Tiering



S3標準  
- 低頻度アクセス



S3 1 ゾーン  
- 低頻度アクセス



S3 Glacier



S3 Glacier  
Deep Archive

## Frequent

- アクティブ、頻繁にアクセスするデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.023/GB~

## Access frequency

- 変化するアクセスパターンのデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.023~
- \$0.002/GB
- オブジェクト毎の管理料金
- 最低保持期限

- 低頻度アクセスデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.019/GB~
- GB毎の取り出し料金
- 最低保持期限
- 最小オブジェクトサイズ

- 再作成可能な低頻度アクセスデータ
- ミリ秒アクセス
- 1 AZ
- \$0.0152/GB~
- GB毎の取り出し料金
- 最低保持期限
- 最小オブジェクトサイズ

## Archive

- アーカイブデータ
- 分~時間アクセス
- $\geq 3$  AZ
- \$0.005/GB~
- GB毎の取り出し料金
- 最低保持期限

- アーカイブデータ
- 時間アクセス
- $\geq 3$  AZ
- \$0.002/GB~
- GB毎の取り出し料金
- 最低保持期限

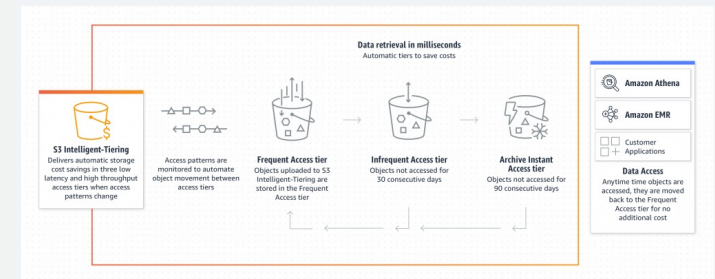
# Amazon S3 Glacier Instant Retrieval

New

即座にデータを取り出すことができる、アーカイブ用途のAmazon S3の新しいストレージクラス

- TBクラスからPBクラスのデータを長期保存しつつ、必要な場合は即座に呼び出す必要がある場合に向けたサービス
- S3 Standardと同じスループットと、同じミリ秒単位の待ち時間でアーカイブデータにアクセスできる
- メディアコンテンツのアーカイブ、ユーザ生成コンテンツ、医療画像、ゲノミクスなどの用途に最適
- データへのアクセスが四半期に1度だと仮定しS3 Standard IAと比較するとストレージコストが最大68%安価になる
- 全てのAWSリージョンでご利用可能

Storage class	Designed for
<input type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access
<input type="radio"/> Intelligent-Tiering	Data with changing or unknown access patterns
<input type="radio"/> Standard-IA	Infrequently accessed data (once a month) with milliseconds access
<input type="radio"/> One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access
<input checked="" type="radio"/> Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds
<input type="radio"/> Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours
<input type="radio"/> Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours



# S3 ストレージクラス (現在)



S3 標準



S3 Intelligent-Tiering



S3標準  
- 低頻度アクセス



S3 1 ザーン  
- 低頻度アクセス



S3 Glacier  
Instant Retrieval



S3 Glacier  
Flexible Retrieval



S3 Glacier  
Deep Archive

- アーカイブデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.005/GB~
- GB毎の取り出し料金
- 最低保持期限

## Frequent

- アクティブ、頻繁にアクセスするデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.023/GB~

## Access frequency

- 変化するアクセスパターンのデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.023~
- \$0.002/GB
- オブジェクト毎の管理料金
- 最低保持期限

- 低頻度アクセスデータ
- ミリ秒アクセス
- $\geq 3$  AZ
- \$0.0138/GB~
- GB毎の取り出し料金
- 最低保持期限
- 最小オブジェクトサイズ

- 再作成可能な低頻度アクセスデータ
- ミリ秒アクセス
- 1 AZ
- \$0.011/GB~
- GB毎の取り出し料金
- 最低保持期限
- 最小オブジェクトサイズ

## Archive

- アーカイブデータ
- 分~時間アクセス
- $\geq 3$  AZ
- \$0.0045/GB~
- GB毎の取り出し料金
- 最低保持期限

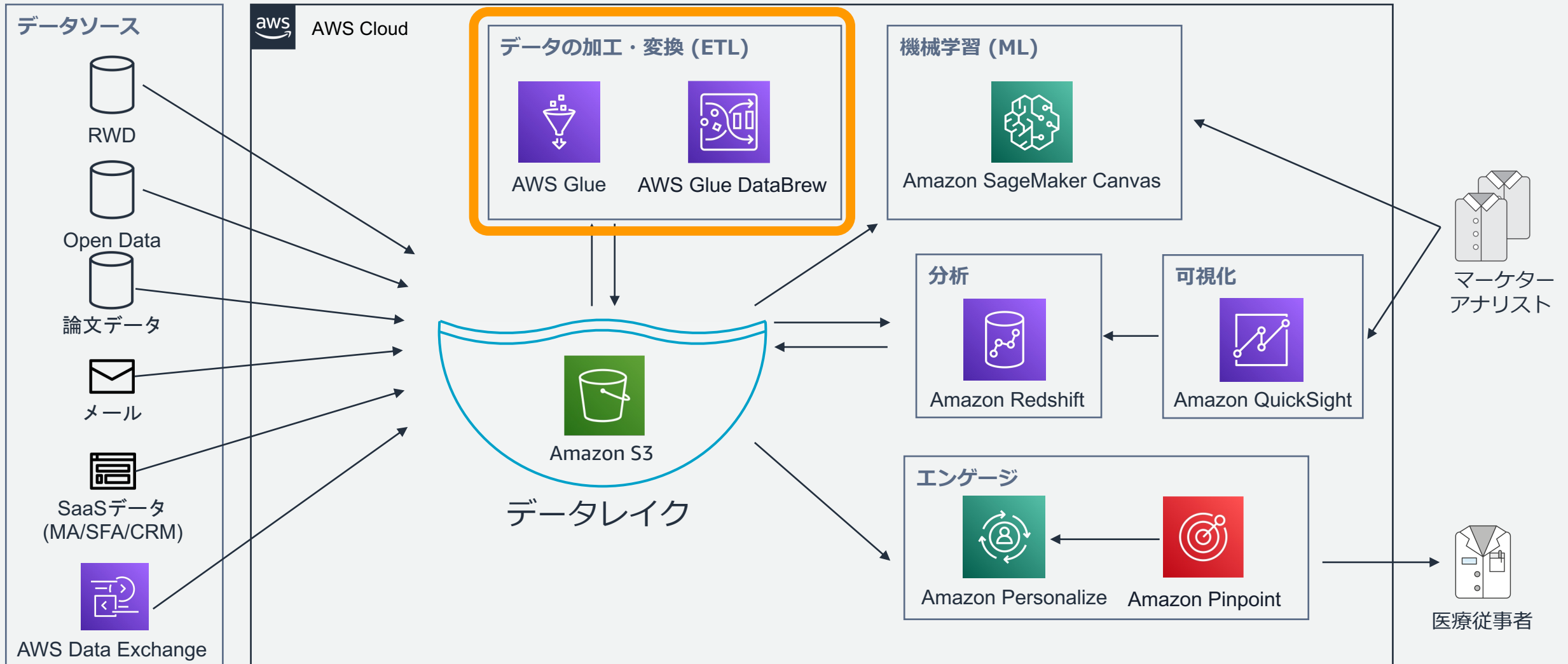
- アーカイブデータ
- 時間アクセス
- $\geq 3$  AZ
- \$0.002/GB~
- GB毎の取り出し料金
- 最低保持期限



# データ利活用のためのプラットフォーム

## データの加工・変換 (ETL)

# コマース領域での利用を想定したデータレイク構成案



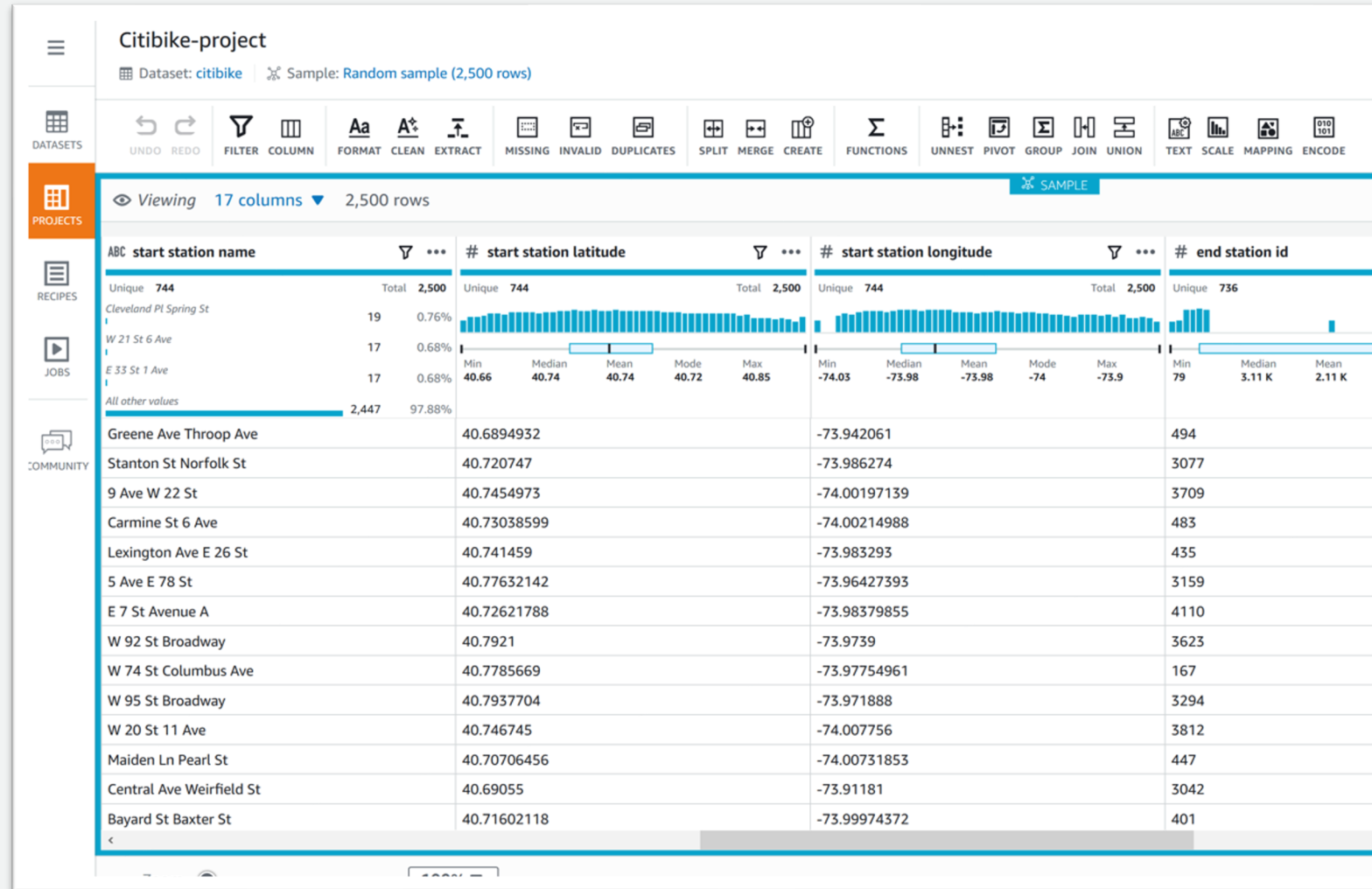
# AWS Glue DataBrew



# AWS Glue DataBrew

## 高度なデータ準備機能をノンコーディングで利用可能

- プロファイリングによるデータの品質の評価
- データのクリーンアップと正規化
- データリネージの視覚化
- 保存された変換手順の再利用やスケジュール実行



# AWS Glue DataBrew

## データのフィルタリング

The screenshot displays the AWS Glue DataBrew console interface for a project named 'chess-project'. The main view shows a data table with columns: # turns, ソース (Source), ABC victory\_status, ABC winner, and ABC increment\_code. A filter is applied to the 'victory\_status' column, showing only rows where the status is 'draw'. The filter panel on the right shows the source column 'victory\_status', the filter condition 'draw', and a summary of the selected values.

# turns	ソース	ABC victory_status	ABC winner	ABC increment_code
44	resign	black	black	15+0
40	outoftime	black	black	10+0
105	outoftime	white	white	10+0
50	resign	black	black	15+0
144	outoftime	draw	draw	10+0
81	resign	black	black	10+0
69	resign	black	black	15+0
77	mate	white	white	7+8
158	mate	black	black	6+6
107	draw	draw	draw	15+5
117	mate	white	white	15+0
44	resign	white	white	10+0
40	resign	black	black	15+0
59	resign	white	white	15+5
110	resign	black	black	10+0
54	resign	black	black	10+0
69	resign	black	black	10+0

**フィルタ値**

ソース列: victory\_status

フィルタ条件: draw

一意の値 (1): draw (1個の選択した値を表示)

一意の値	回数	割合
resign	27	69%
outoftime	6	15%
mate	5	12%
draw	1	2%

# AWS Glue DataBrew

## データの結合

airbnb-listings | データセット: airbnb-listings | サンプル: 最初の n 個のサンプル (500 個の行) | ジョブを作成 | 系統 | アクション

### 結合

ステップ 1  
データセットを選択

ステップ 2  
結合の詳細を指定

結合タイプを選択

- 内部結合  
テーブル A とテーブル B の結合条件を満たすすべての行を選択します。
- 左結合  
テーブル A からすべての行と、テーブル B から結合条件を満たす行を選択します。
- 右結合  
テーブル B からすべての行と、テーブル A から結合条件を満たす行を選択します。
- 外部結合  
結合条件に関係なく、テーブル A とテーブル B からすべての行を選択します。
- 左除外結合  
テーブル A から、結合条件を満たす行を除くすべての行を選択します。
- 右除外結合  
テーブル B から、結合条件を満たす行を除くすべての行を選択します。
- 外部除外結合  
テーブル A とテーブル B から、結合条件を満たす行を除くすべての行を選択します。

結合キー

テーブル A (このプロジェクト)  
airbnb-listings  
# id

テーブル B  
airbnb-reviews  
# listing\_id

別の結合キーを追加する

列リスト | 結合されたテーブルのプレビュー

列リスト  
結合に含める列を選択します

列を検索

<input checked="" type="checkbox"/>	ソーステーブル	列名
<input checked="" type="checkbox"/>	テーブル A	# id
<input checked="" type="checkbox"/>	テーブル A	ABC name
<input checked="" type="checkbox"/>	テーブル A	# host_id
<input checked="" type="checkbox"/>	テーブル A	ABC host_name
<input checked="" type="checkbox"/>	テーブル A	ABC neighbourhood_group
<input checked="" type="checkbox"/>	テーブル A	ABC neighbourhood
<input checked="" type="checkbox"/>	テーブル A	ABC lat_lon
<input checked="" type="checkbox"/>	テーブル A	ABC room_type
<input checked="" type="checkbox"/>	テーブル A	# price

# AWS Glue DataBrew

## 欠損値の補完

The screenshot displays the AWS Glue DataBrew interface. The main area shows a data preview with columns: **chembl\_id**, **updated\_on**, and **updated\_by**. The **updated\_on** column contains several null values. The sidebar on the right, titled "欠落した値" (Missing Values), provides options for handling these values:

- ソース列 (Source Column): updated\_on
- 欠落した値アクション (Missing Value Action):
  - 欠落した値がある行を削除する (Delete rows with missing values)
  - 空の値で埋める (Fill with empty values)
  - null で埋める (Fill with null)
  - 最後の有効な値で埋める (Fill with last non-null value)
  - 最も頻繁な値で埋める (Fill with most frequent value)
  - カスタム値で埋める (Fill with custom value)
  - 数値集計で埋める (Fill with numerical aggregation)
- カスタム値 (Custom Value): 2021/02/17
- 変換を適用 (Apply Transformation):
  - すべての行 (500 行) (All rows (500 rows)) - 変換はデータセット内のすべての行に適用されます (Transformation is applied to all rows in the dataset)
  - フィルタリングされた行 - 適用された 0 フィルタ (500) (Rows filtered - 0 filters applied (500)) - 変換はグリッド内のフィルタリングされた行に適用されます (Transformation is applied to filtered rows in the grid)

At the bottom of the sidebar, there are buttons for "キャンセル" (Cancel) and "適用" (Apply).

# AWS Glue DataBrew

## 関数を使った新たな列の作成

The screenshot displays the AWS Glue DataBrew interface for a job named "sales-data-ett". The main view shows a data table with columns "Date" and "Target Close". A search menu is open, listing various functions such as "日付関数" (Date functions), "集計関数" (Aggregation functions), and "テキスト関数" (Text functions). The "日付関数" menu is expanded, showing options like "DATEDIFF", "DATEADD", "DATEFORMAT", etc. The "DATEDIFF" function is highlighted, and its configuration panel is open on the right. The configuration panel shows the function name "DATEDIFF", a description, and settings for "値 1" (Value 1) and "値 2" (Value 2), both set to "ソース列を選択" (Select source column). The unit is set to "日" (Day), and the resulting column name is "Date\_DATEDIFF".

ABC Date	ABC Target Close
一意 266 10/17/2011	一意 286 10/4/2011
7/21/2011	11/16/2011
9/29/2011	11/27/2011
その他のすべての値 484	その他のすべての値 483
1/2/2011	2/2/2011
1/3/2011	4/9/2011
1/6/2011	5/4/2011
1/11/2011	2/14/2011
1/11/2011	1/31/2011
1/12/2011	3/16/2011
1/13/2011	2/18/2011
1/16/2011	3/6/2011
1/19/2011	3/3/2011
1/25/2011	2/27/2011
1/26/2011	2/23/2011
1/29/2011	2/24/2011
1/30/2011	2/27/2011
1/31/2011	7/24/2011
2/3/2011	4/30/2011
2/3/2011	2/3/2011
2/5/2011	2/5/2011
2/6/2011	6/29/2011
2/13/2011	4/1/2011
2/15/2011	3/25/2011
2/15/2011	2/15/2011

**関数選択メニュー:**

- 数学関数
- 集計関数
- テキスト関数
- 日付関数
- ウィンドウ関数
- ウェブ関数
- その他の関数

**日付関数メニュー:**

- 日付
- TIME
- DATETIME
- DATEADD
- DATEDIFF (2つのソース列間の日付単位の差。)
- DATEFORMAT
- MONTH
- MONTHNAME
- YEAR
- DAY
- WEEKDAY
- WEEKNUMBER
- HOUR
- MINUTE
- SECOND
- UNIXTIME
- UNIXTIMEFORMAT
- NOW
- TODAY

**列作成パネル (DATEDIFF):**

- 関数に基づく
- 関数を選択 (選択した関数に基づいて列を作成する)
- DATEDIFF
- DATEDIFF: ソース列 1 または値 1 とソース列 2 または値 2 の間の日付単位 (年、月、日) の差を新しい列で返します。
- 値 1:  カスタム値を入力  ソース列を選択
- 値 2:  カスタム値を入力  ソース列を選択
- 単位: 日
- 宛先列: Date\_DATEDIFF

# AWS Glue DataBrew

## ジョブとデータリネージ

- データ変換処理をレシピとして保存し、レシピをもとにジョブを作成可能
- オンデマンドかスケジュールによる実行
- ジョブの実行状況も確認可能
- 各アイコンをクリックすることで、詳細情報の確認も可能

The screenshot shows the AWS Glue DataBrew console interface for creating a job. The main page is titled "ジョブを作成" (Create Job) and includes a sidebar with navigation icons for DataBrew, Data Sets, Projects, Recipes, Jobs, and Recent Jobs. The main content area is divided into several sections:

- ジョブの詳細** (Job Details): A form to enter the job name and ID. The job name is "chess-winner-summary".
- ジョブタイプ** (Job Type): A section for selecting the job type. Two options are visible: "レシピジョブを作成" (Create Recipe Job) and "プロフィールジョブを作成する" (Create Profile Job).
- ジョブ入力** (Job Input): A section for selecting the data set and recipe to use for the job. The "データセット" (Data Set) section shows "chess-games" selected. The "レシピを選択" (Select Recipe) section shows "chess-project-recipe" selected.

A modal window titled "スケジュールを追加" (Add Schedule) is open, showing a form to configure a schedule. The schedule name is "everyday". The execution frequency is set to "毎日" (Daily) with a frequency of "1" (hourly) and a start time of "00:00". The modal also displays a preview of the schedule for the next 5 occurrences on 2022年3月18日.

# AWS Glue DataBrew

## ジョブとデータリネージ

- データのインプットからアウトプットまでの流れを可視化
- ジョブの実行状況も確認可能
- 各アイコンをクリックすることで詳細情報の確認も可能

The screenshot displays the AWS Glue DataBrew console interface for a job named 'chess-winner-summary'. The breadcrumb navigation shows 'DataBrew > ジョブ > chess-winner-summary'. The main content area shows a workflow diagram with the following components:

- S3**: chess-games.XLSX (4.6 MB)
- データセット**: chess-games (4.6 MB)
- ジョブ**: chess-winner-summary (実行中 出力)
- S3**: s3://aaaa-bucket/databrew-o... CSV

Below the workflow, the 'レシピ' (Recipe) icon is highlighted, and a modal window titled 'chess-project-recipe' is open, showing the following details:

レシビ名	レシビバージョン
chess-project-recipe	2.0

発行済みバージョン: 3時間前 / ojunquepe-lsengard (2021年2月16日, 6:24:48 午後)

バージョンの説明: Second version of the recipe.

レシビのステップ (7)

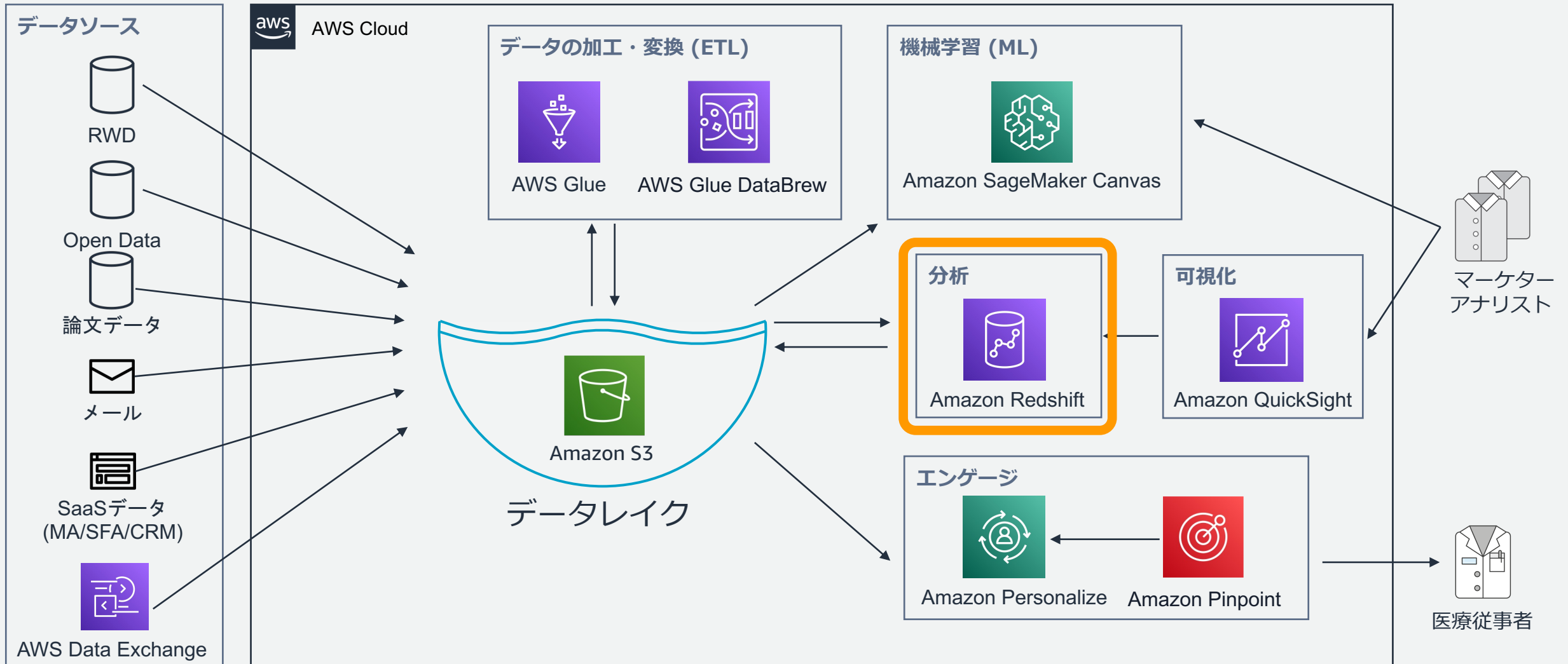
- フィルタ値 by white\_rating
- フィルタ値 by black\_rating
- グループ化 winner, victory\_status and 作成 winner\_count 開始位置 COUNT(winner)
- フィルタ値 by victory\_status
- テキストを置き換える mate と checkmate in victory\_status
- テキストを置き換える resign と other player resigned in victory\_status
- テキストを置き換える outoftime と time ran out in victory\_status

Buttons: レシビを表示, 閉じる

# データ利活用のためのプラットフォーム 分析



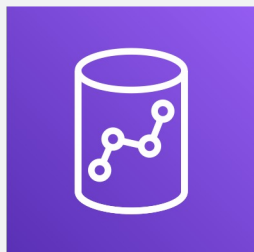
# コマース領域での利用を想定したデータレイク構成案



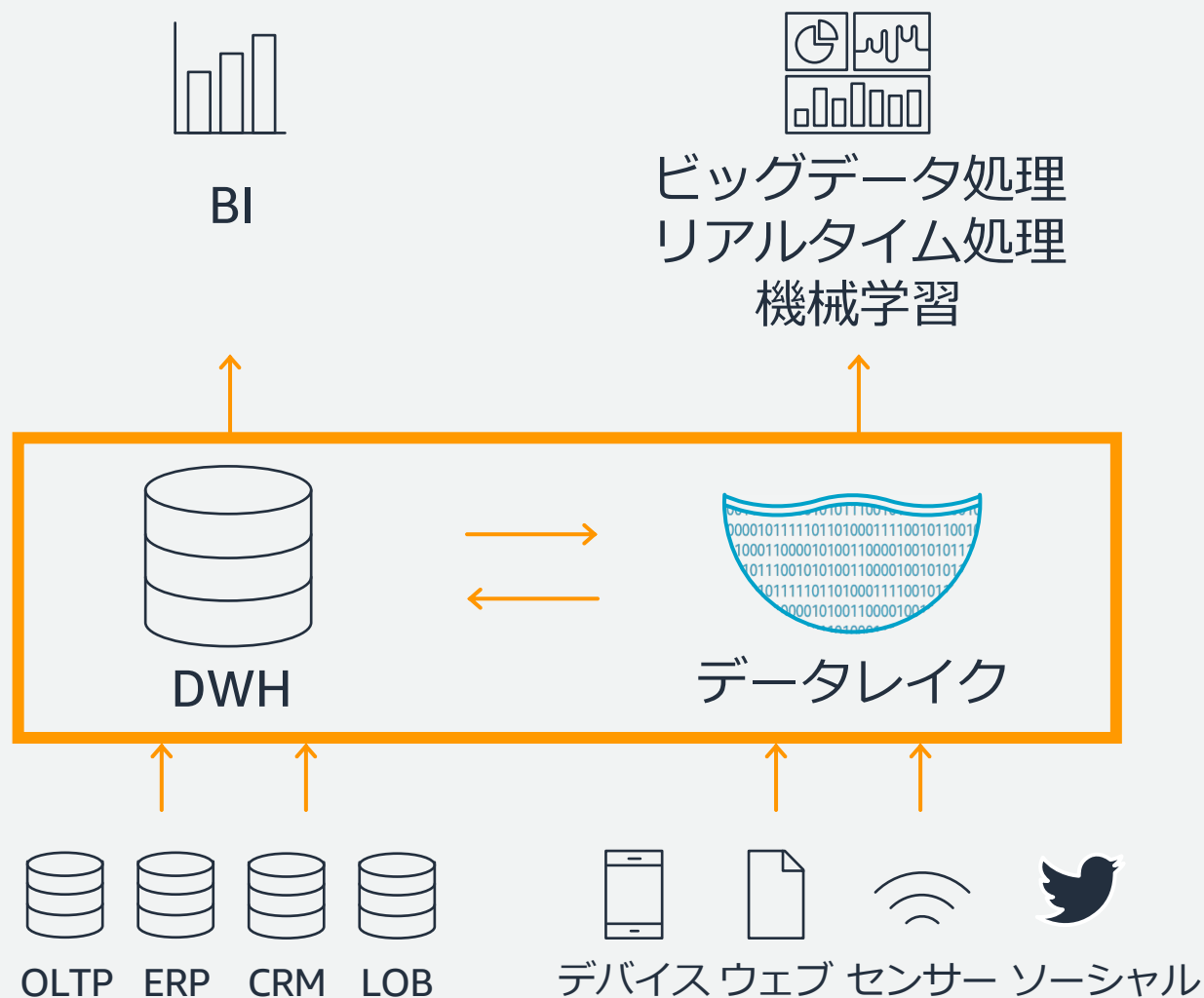
# Amazon Redshift



# Amazon Redshift



高速なスケラブルで  
費用対効果の高い  
データウェアハウス  
およびデータレイク  
分析マネージドサービス



# Amazon Redshift のユースケース

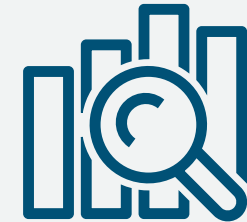
主に大容量データを高速に集計・分析する必要があるワークロードに活用可能なデータウェアハウス



経営ダッシュボード



定型レポートニング



アドホック分析



ETL/バッチ

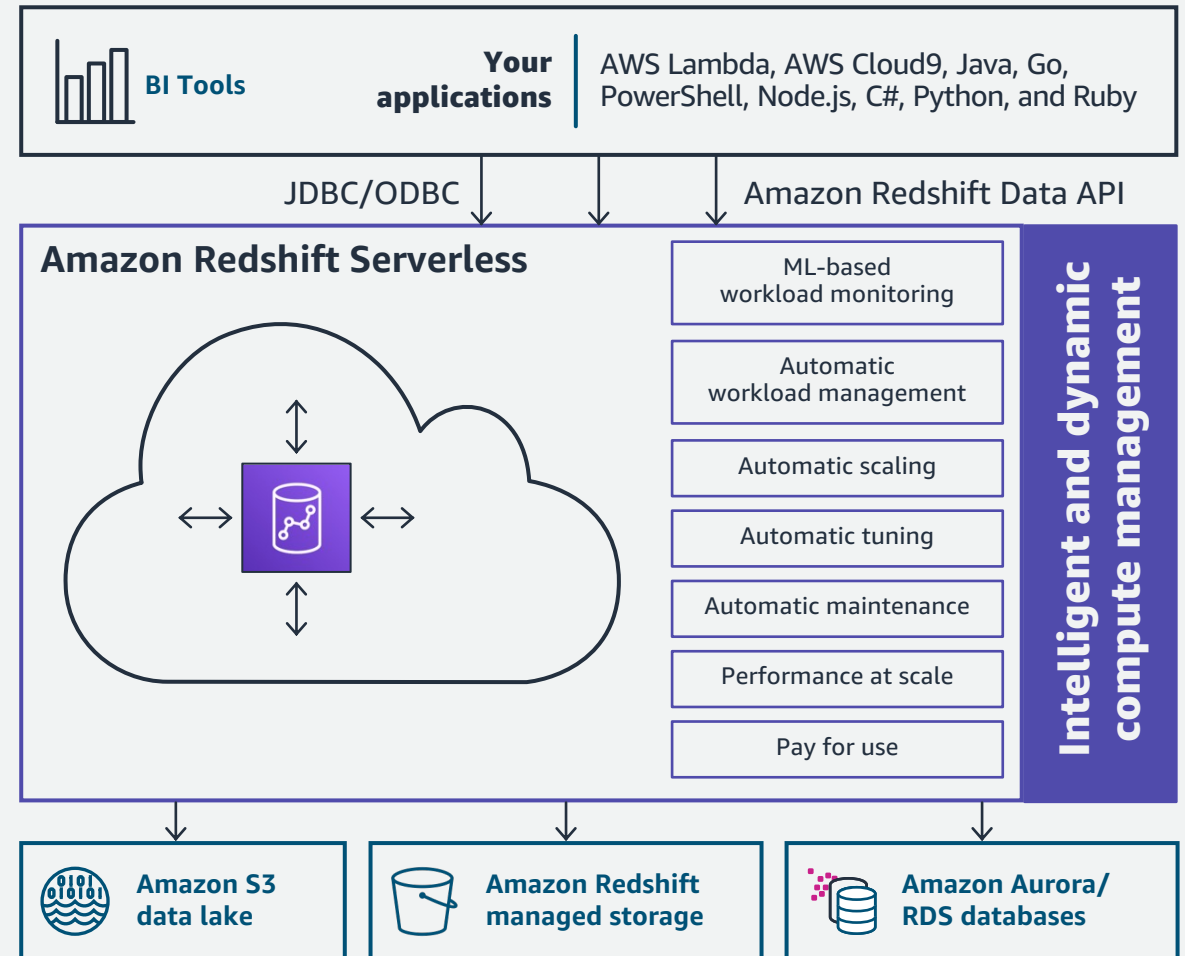


機械学習の前処理

# Amazon Redshift Serverless

New (Public Preview)

- クラスターのセットアップや管理が不要なサーバーレスサービス
- サービス使用中は秒単位で課金  
アイドル状態の場合は課金なし
- コンピューティングのニーズを予測するのが難しい場合に最適
  - 変動するワークロード
  - アイドル時間のある定期的なワークロード
  - アドホック分析のニーズ
  - テスト環境や開発環境



# Amazon Redshift Serverless

データレイクや基幹システムから直接データをブラウズ、クエリ実行が可能

The screenshot displays the Amazon Redshift query editor v2 interface. The left sidebar shows the database structure for 'sales\_db', including tables like 'inventory\_ods', 'store\_sales\_ods', and 'store\_sales\_old'. The main editor area shows a federated query example with the following SQL code:

```
6
7 -- QUERY POSTGRES ODS DATA
8 SELECT *
9 FROM ext_postgres_ods.inventory_ods;
10
11 -- COMBINE POSTGRES ODS DATA with LOCAL, WH DATA in REDSHIFT SERVERLESS
12 SELECT d_year as year,
13        ('Q' || d_qoy) as quarter,
14        i_category as item_category,
15        count(*) as orders
16 FROM
17 ext_postgres_ods.store_sales_ods s, date_dim d, item i
18 WHERE s.ss_sold_date_sk = d.d_date_sk
19        and s.ss_item_sk = i.i_item_sk
20        and s.ss_transaction_dt > current_date -1
21 GROUP BY 1,2,3
22 ORDER BY 4 desc;
23
```

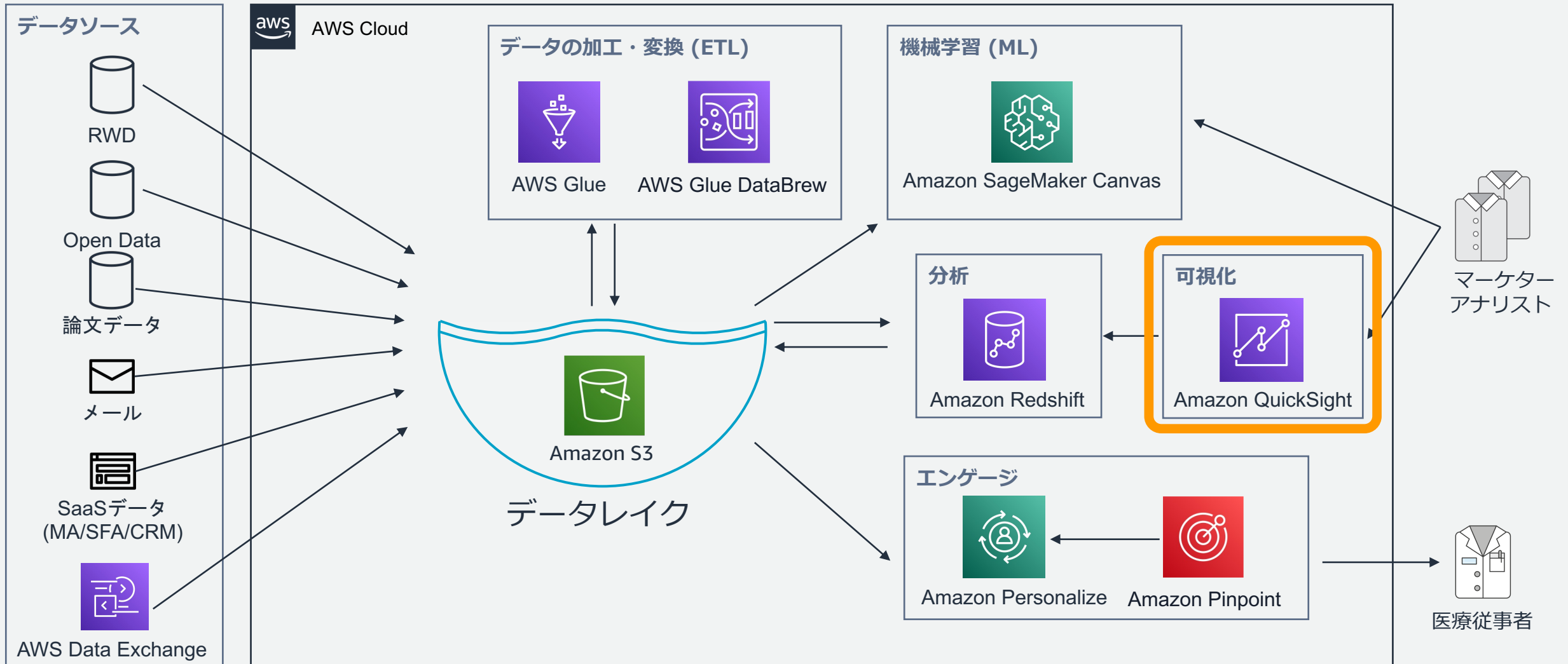
The results are displayed in a table with 11 rows and 4 columns: year, quarter, item\_category, and orders.

year	quarter	item_category	orders
2021	Q3	Sports	26
2021	Q3	Home	25
2021	Q3	Men	25
2021	Q3	Women	22
2021	Q3	Electronics	21
2021	Q3	Children	20
2021	Q3	Jewelry	20
2021	Q3	Music	19
2021	Q3	Shoes	16



# データ利活用のためのプラットフォーム 可視化

# コマース領域での利用を想定したデータレイク構成案



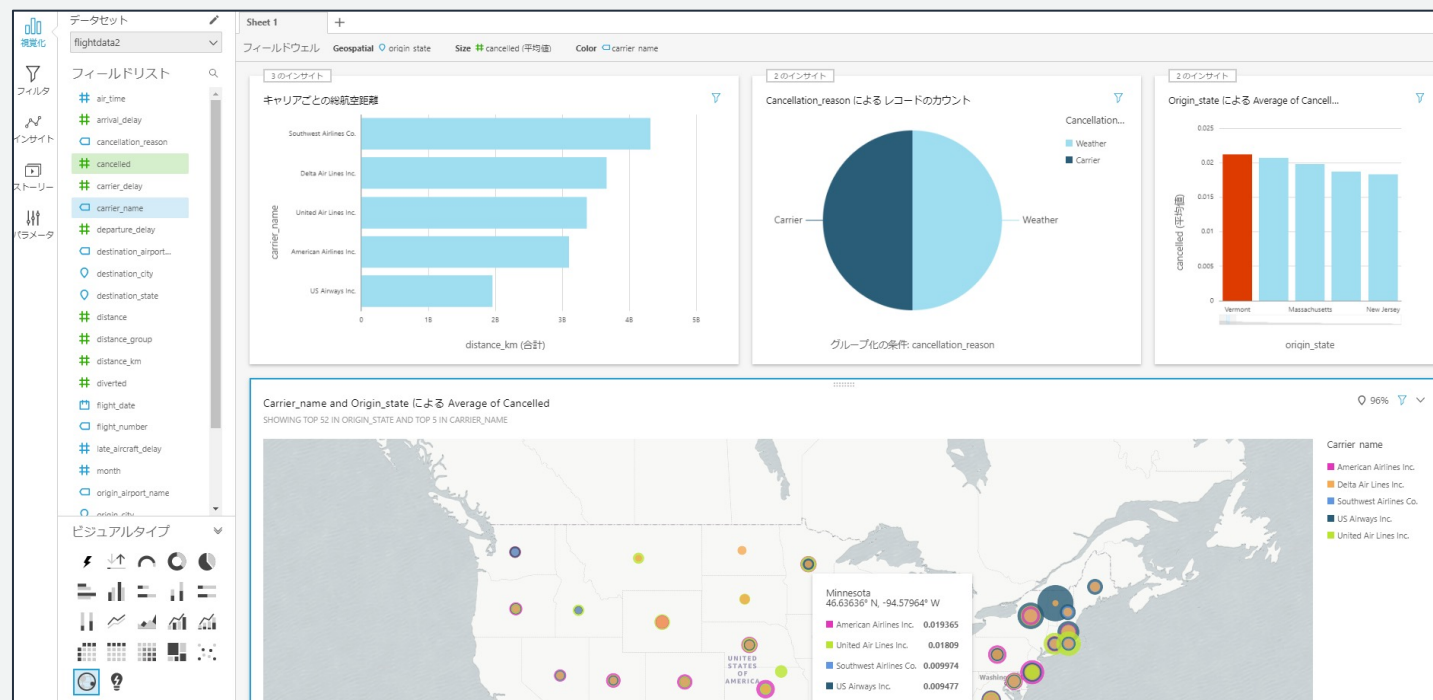
# Amazon QuickSight



# Amazon QuickSight

## サーバーレスで運用不要なブラウザベースのBIツール

- インタラクティブな操作性
  - フィルタ
  - ドリルダウン
  - ズーム等
- 高速な動作
- ブラウザだけで利用可能
- 数クリックで共有
- セッションあたりの料金体系



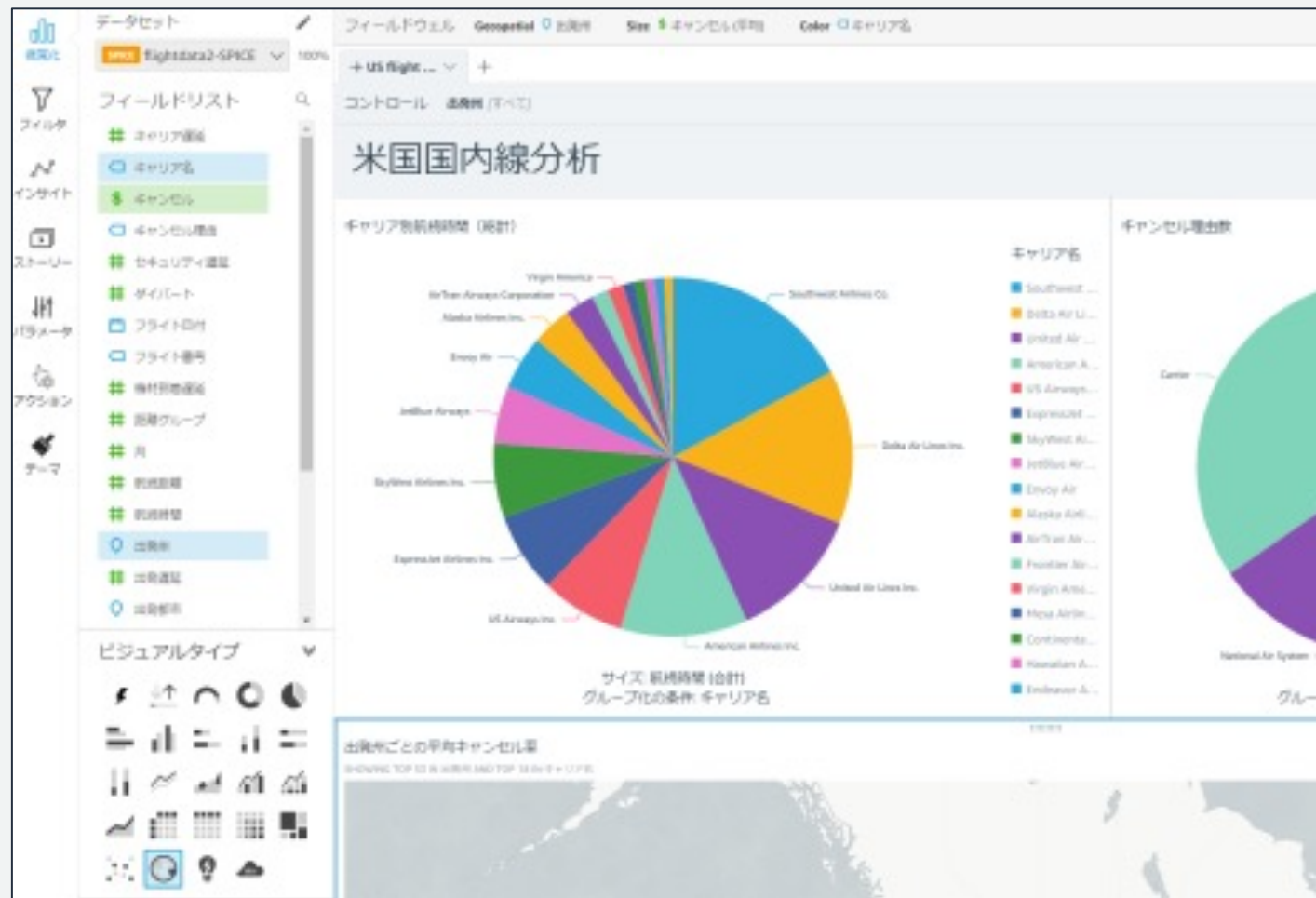
※サポートされるブラウザ [https://docs.aws.amazon.com/ja\\_jp/quicksight/latest/user/supported-browsers.html](https://docs.aws.amazon.com/ja_jp/quicksight/latest/user/supported-browsers.html)


























# 直感的なビジュアライゼーション

## 簡単な操作で可視化を実現

- ビジュアルタイプを選択→フィールドをドラッグ&ドロップ
- フィルタ、ドリルダウン、ユーザ入力等の機能
- データのCSVエクスポート



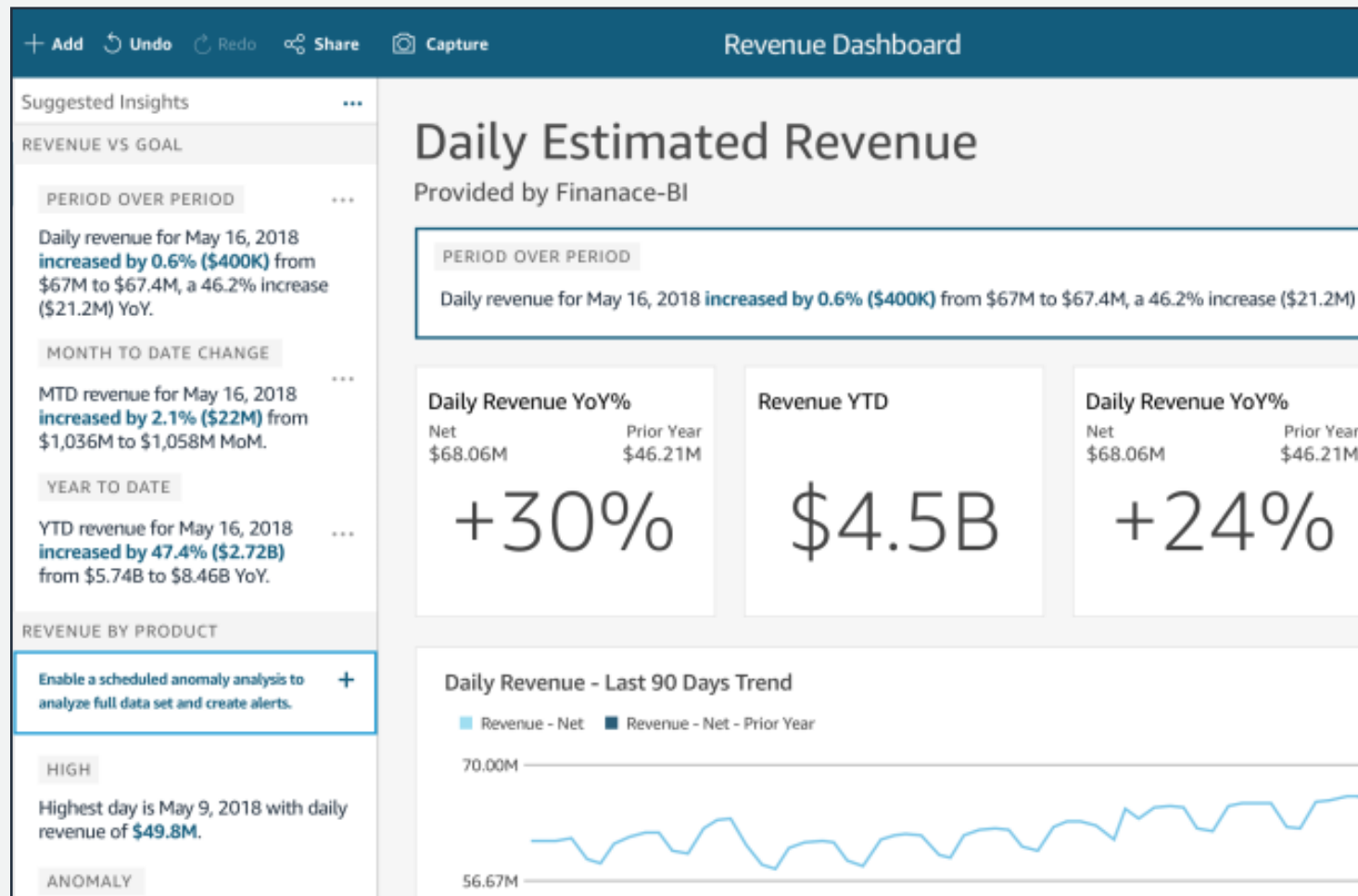
# 多様なデータソースに対応

 <p>Upload a file (.csv, .tsv, .clf, .elf, .xlsx, .json)</p>	 <p>Salesforce Connect to Salesforce</p>	 <p>S3 Analytics</p>	 <p>S3</p>
 <p>Athena</p>	 <p>RDS</p>	 <p>Redshift Auto-discovered</p>	 <p>Redshift Manual connect</p>
 <p>MySQL</p>	 <p>PostgreSQL</p>	 <p>SQL Server</p>	 <p>Aurora</p>
 <p>MariaDB</p>	 <p>Presto</p>	 <p>Spark</p>	 <p>Teradata Provided by Teradata</p>
 <p>Snowflake</p>	 <p>AWS IoT Analytics</p>	 <p>GitHub</p>	 <p>Twitter</p>
 <p>Jira</p>	 <p>ServiceNow</p>	 <p>Adobe Analytics</p>	

# ML（機械学習）インサイトの提供

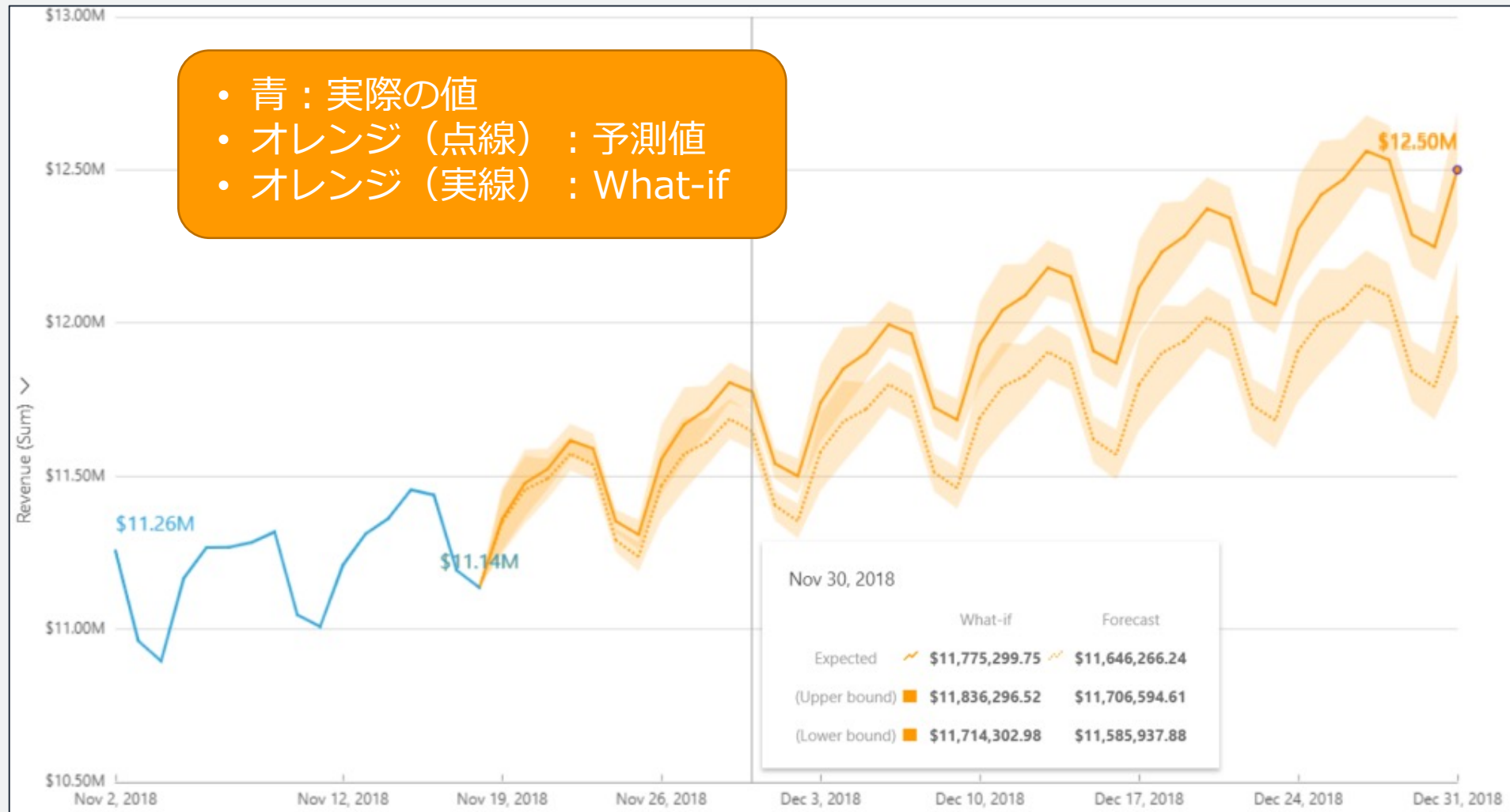
自動的に異常を発見し、多様なインサイトを提供する最新鋭のインサイト機能

- MLによる異常検知
- MLによる予測
- 自然な文章を自動生成
- SageMakerとの連携による予測



# ML（機械学習）インサイトの提供

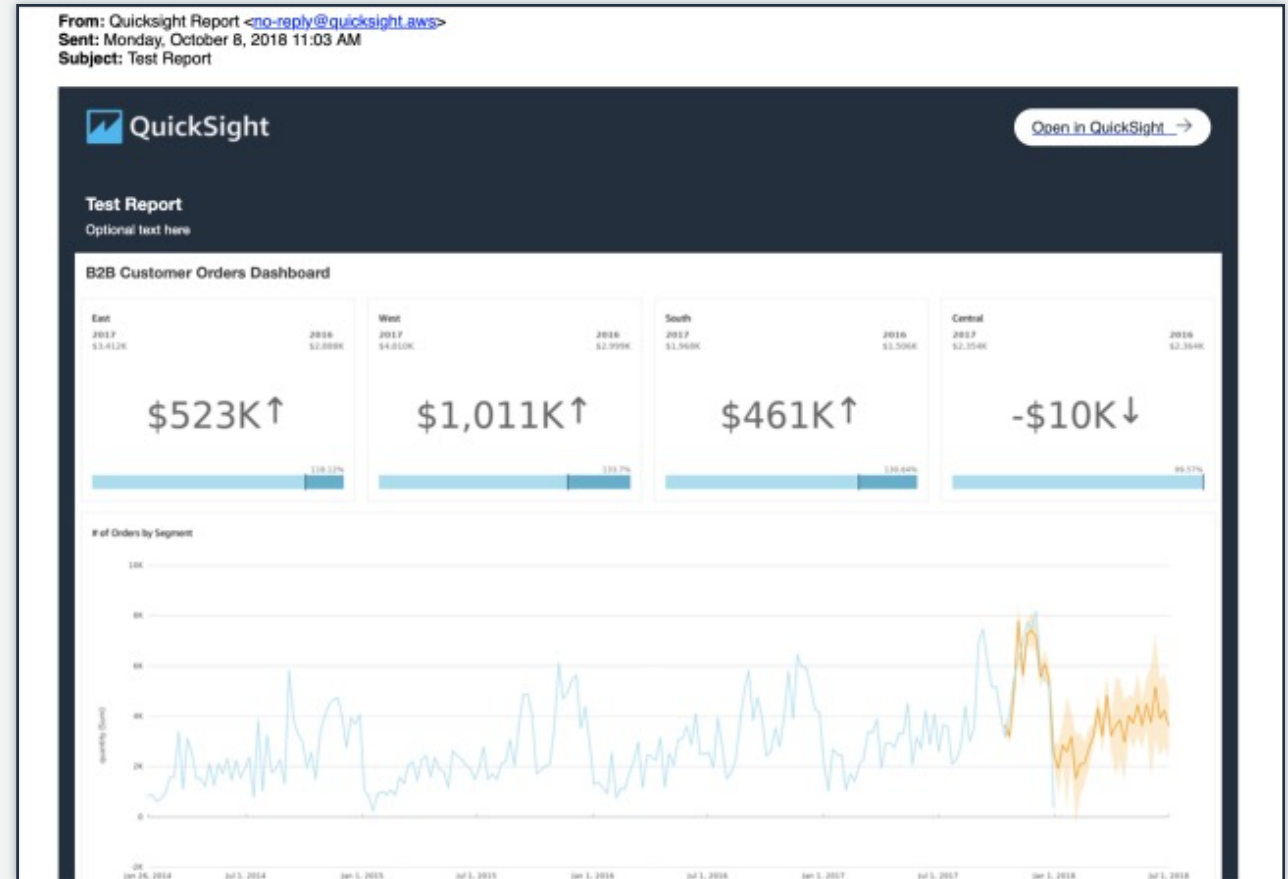
予測とWhat-if分析を利用してターゲットを指定して分析する



# 分析結果がメールボックスに届く

## QuickSightが定期的に分析スナップショットをメール送信

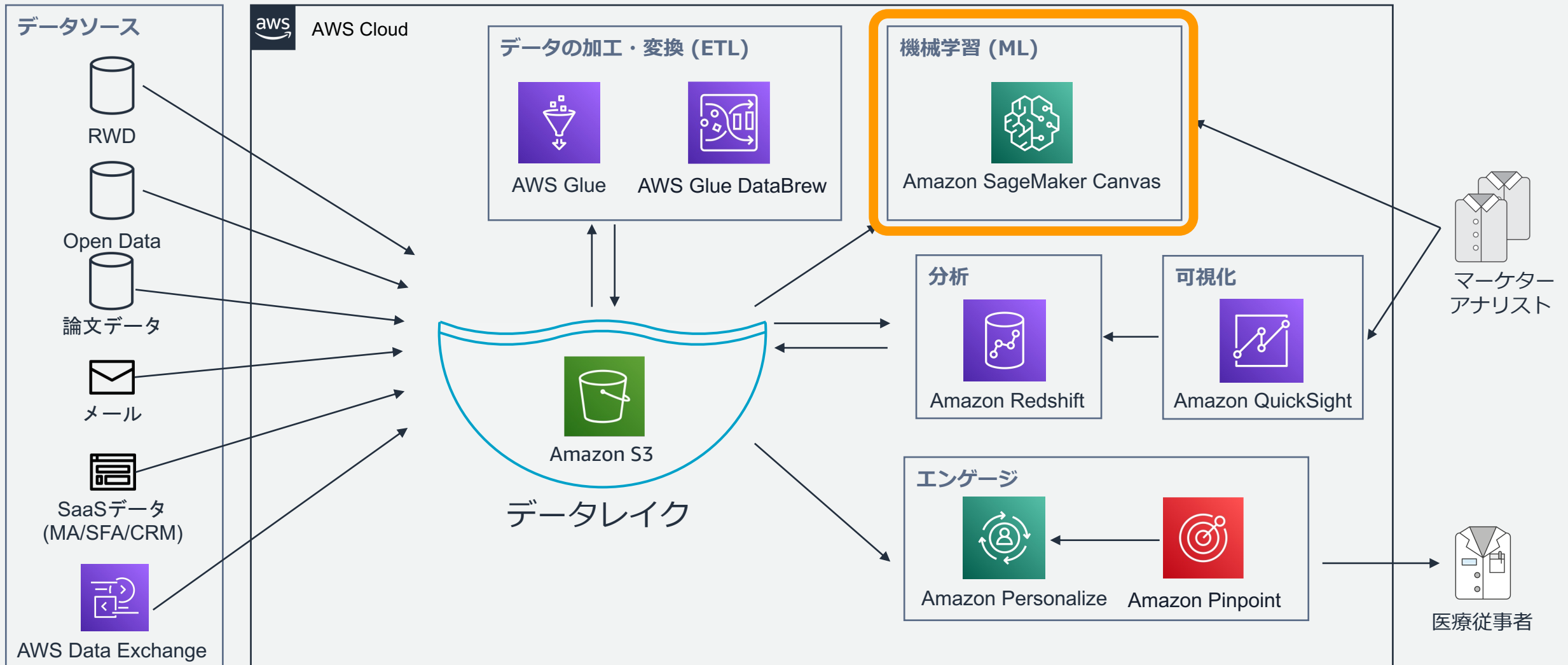
- Eメールレポートを日次、週次等のスケジュールで送信
- 個別のユーザ・グループごとに送信
- 不要レポートをオプトアウト可能  
必要な情報のみにフォーカス
- ペイ・パー・セッション料金



# データ利活用のためのプラットフォーム 機械学習 (ML)



# コマース領域での利用を想定したデータレイク構成案



# コマース領域における機械学習 (ML) のユースケース

1. 売上予測
2. 顧客の離反傾向予測
3. ターゲティング



AWSクラウドに  
データを送信



スケーラブルなビッグデータ  
フレームワークとAmazon Machine Learning  
をプロビジョニング



データの視覚化して製品の品質と  
安全性の問題を予測

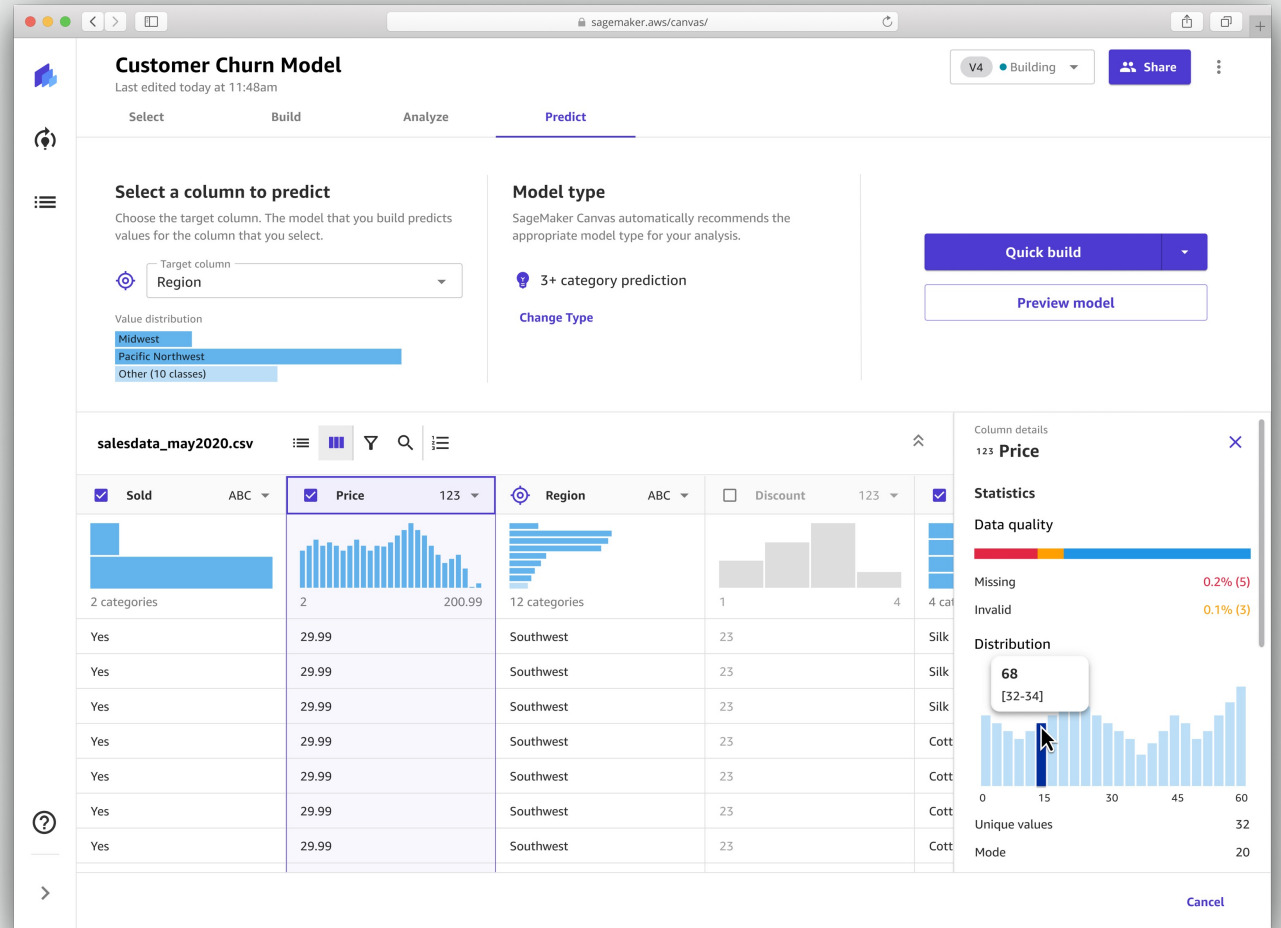
# Amazon SageMaker Canvas



# Amazon SageMaker Canvas

## ビジネスアナリスト向けのノーコードな機械学習サービス

- 視覚的なインターフェイス
- Amazon SageMakerと同じテクノロジーを利用
- コードを記述したりMLの専門知識を必要とせずにMLモデルを構築
- 強力な自動機械学習技術をもとに高品質の予測モデルを作成可能



# Amazon SageMaker Canvas

さまざまなソースからのデータセットを結合することが可能

**Import Data**

Upload S3 Snowflake Crystal 1 Redshift Crystal 1 Add Connection

Connection name Context

Search

database1 database2 database3 database4

schema1 schema2

table1

Autosaved 8/9/21 at 11:34 AM Edit in SQL

table1.csv table2.csv

Import preview Show dropped columns

<input checked="" type="checkbox"/> Sold	ABC	<input type="checkbox"/> Price	123	<input checked="" type="checkbox"/> Region	ABC	<input checked="" type="checkbox"/> Discount	123	<input checked="" type="checkbox"/> Fabric	ABC	<input checked="" type="checkbox"/> Age	123
Yes		29.99		Southwest		23		Cotton		27	
Yes		29.99		Southwest		23		Silk		35	
Yes		29.99		Southwest		23		Silk		32	
Yes		29.99		Southwest		23		Silk		32	
Yes		29.99		Southwest		23		Cotton		30	

Previewing the first 100 rows Close Import data



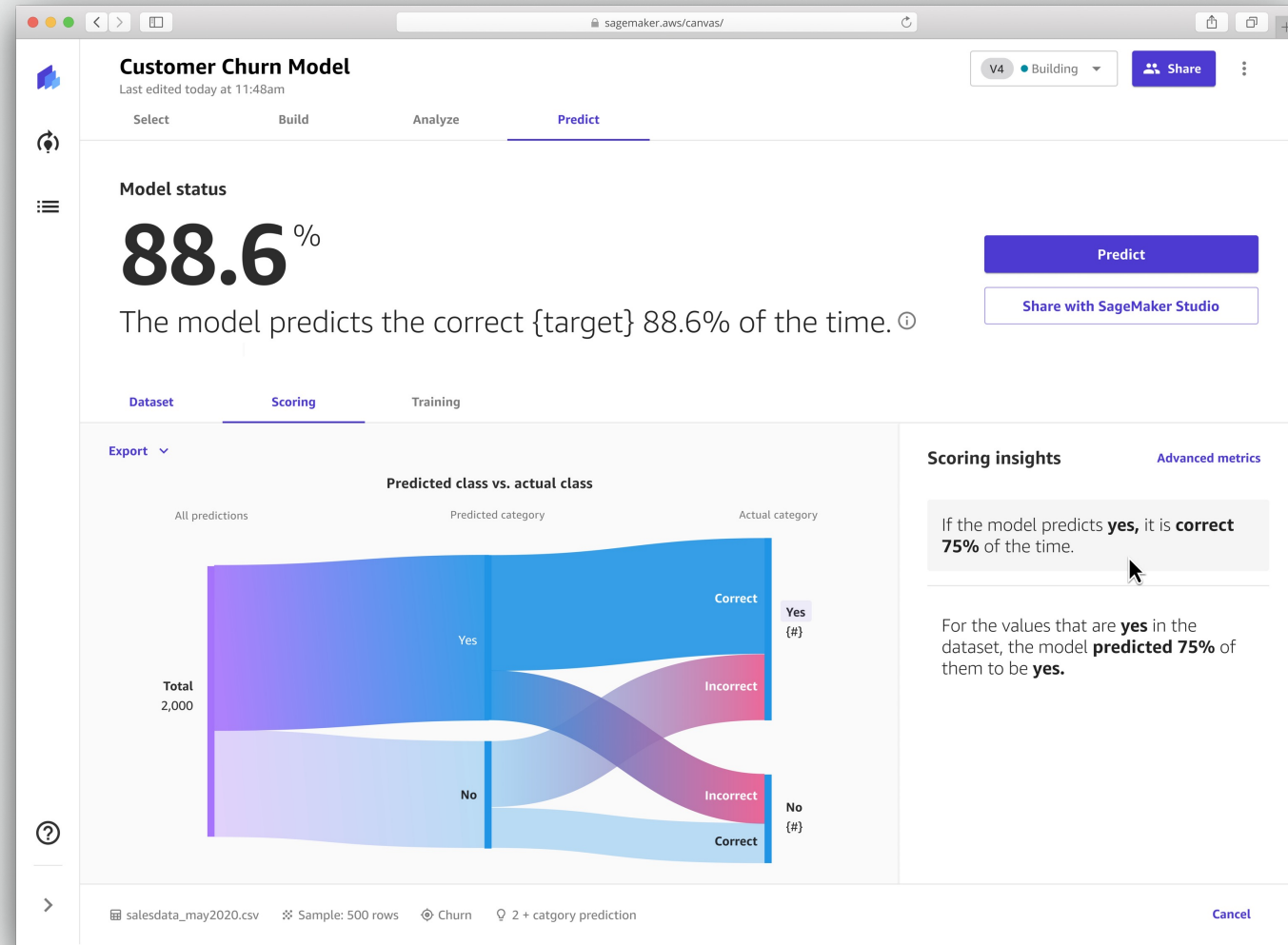
# Amazon SageMaker Canvas

データセットに対する正確な ML モデルを自動的に構築する

The screenshot displays the Amazon SageMaker Canvas interface for a 'Customer Churn Model'. The main window is in the 'Predict' phase. A 'Model type' dialog box is open, showing two options: '3+ category prediction' (recommended) and 'Numeric prediction'. The background interface shows the 'Select a column to predict' section with 'Region' selected as the target column. The 'Value distribution' for 'Region' is shown as a bar chart with categories: Midwest, Pacific Northwest, and Other (10 classes). Below this, a table of data is visible with columns for 'Sold', 'Price', and 'Region'. The 'Price' column is highlighted, and its 'Statistics' are shown on the right, including 'Data quality' (Missing: 0.2% (5), Invalid: 0.1% (3)) and 'Distribution' (Unique values: 32, Mode: 20). A bar chart for 'Price' shows a peak at 68 with a range of [32-34].

# Amazon SageMaker Canvas

## 数分で最初の ML モデルを生成



# Amazon SageMaker Canvas

作成したモデルをデータサイエンティストと共有可能

The screenshot displays the Amazon SageMaker Canvas interface for a 'Customer Churn Model'. The model's performance is shown as 88.6% accuracy. A 'Share with SageMaker Studio' dialog box is open, allowing the user to select model versions to share. The dialog shows four versions (V2, V4, V5, V8) all marked as 'Ready' and selected. A link is generated: <https://canvas.com...customer-churn-model>. The background shows a prediction chart and a confusion matrix.

# Amazon SageMaker Canvas

## 予測の実施

The screenshot displays the Amazon SageMaker Canvas interface for a 'Customer Churn Model'. The model is in the 'Predict' stage, and the user is viewing the 'Predict target values' section. The interface includes a table of feature values, a 'Predict target values' section, and a 'Churn prediction' bar chart.

**Predict target values**

Batch prediction | Single prediction

Modify values to calculate target column in real time.

Filter columns

Column	Column impact	Value	Reset all to average
Contract	61.3%	Two year	
OnlineSecurity		Month-to-month	
TechSupport		One year	
InternetService		Fiber optic	
PaymentMethod		Electronic check	
OnlineBackup		No	
DeviceProtection		Yes	
MonthlyCharges		104.8	
PaperlessBilling		Yes	

**Churn prediction**

Copy

No

Average probability | New probability

No 71.5%

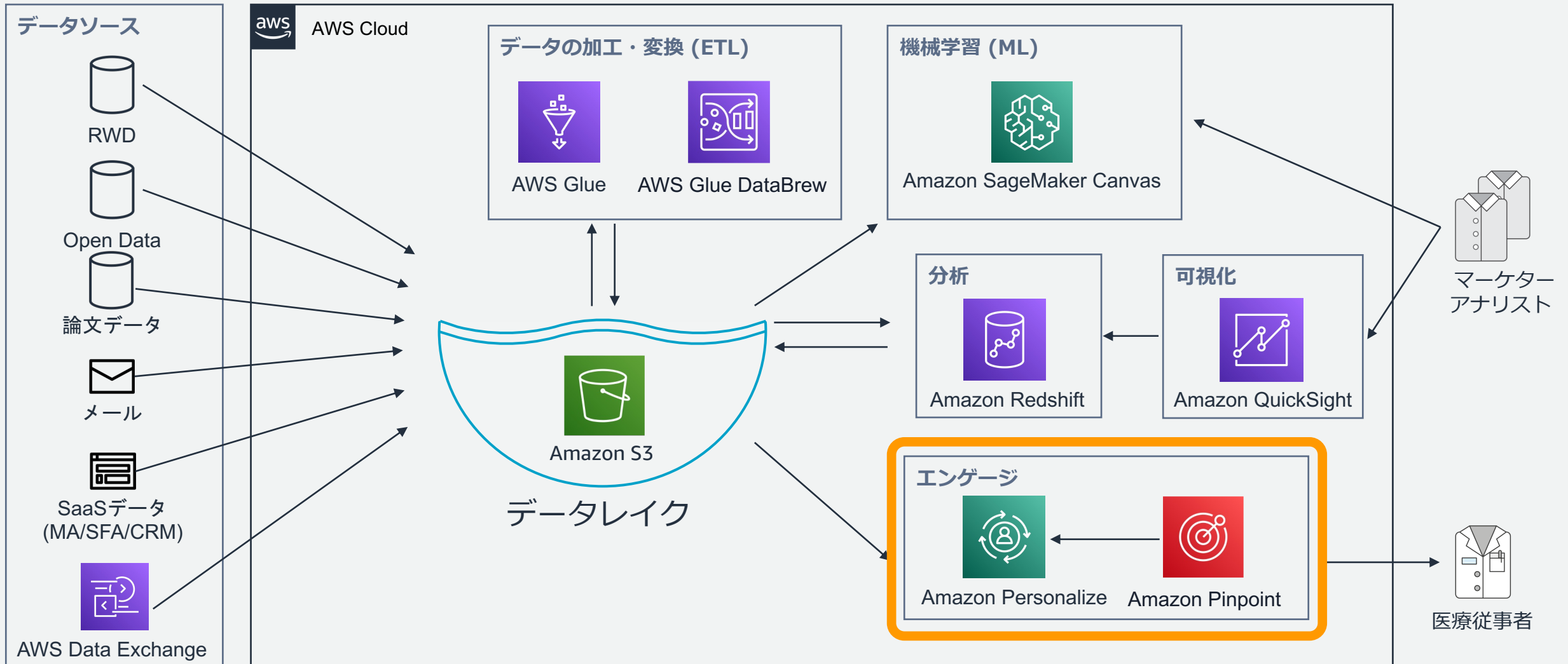
Yes 28.5%

Close | Download

# データ利活用のためのプラットフォーム エンゲージ



# コマース領域での利用を想定したデータレイク構成案



# Amazon Pinpoint



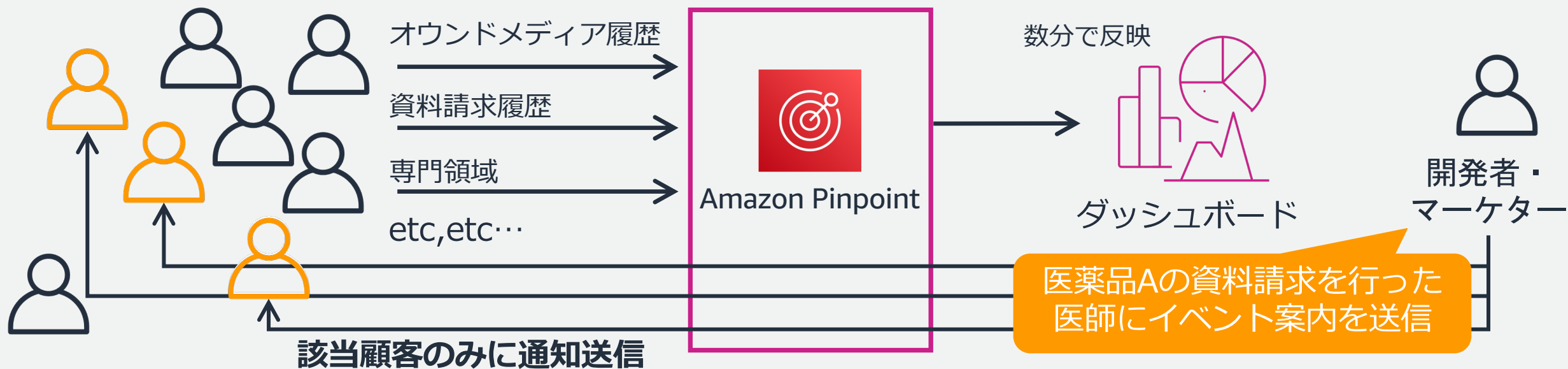
# Amazon Pinpoint

## AWSによるカスタマーエンゲージメントプラットフォーム

- ユーザ行動の解析と可視化
- ターゲティング（セグメンテーション）通知
- モバイルプッシュ/Email/SMS/Voice etc
- 配信数、開封率の把握などの分析

### イベント・属性収集

### 可視化・分析

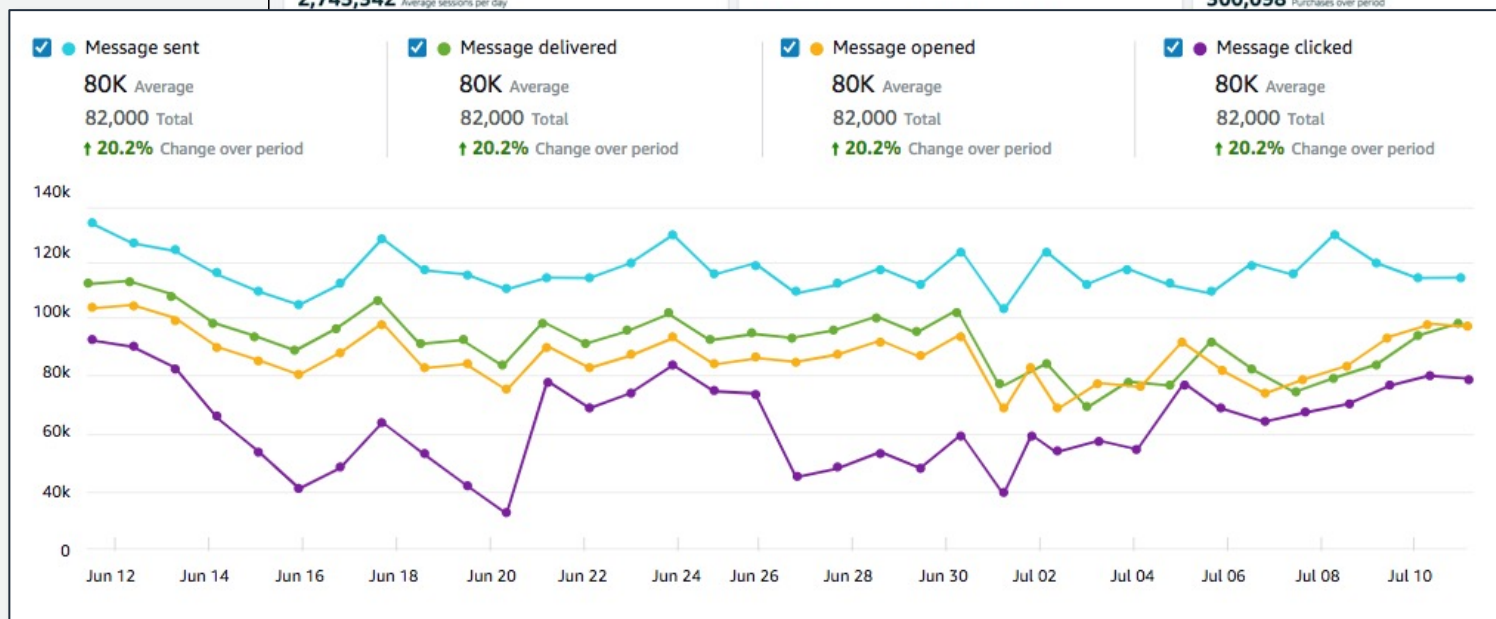
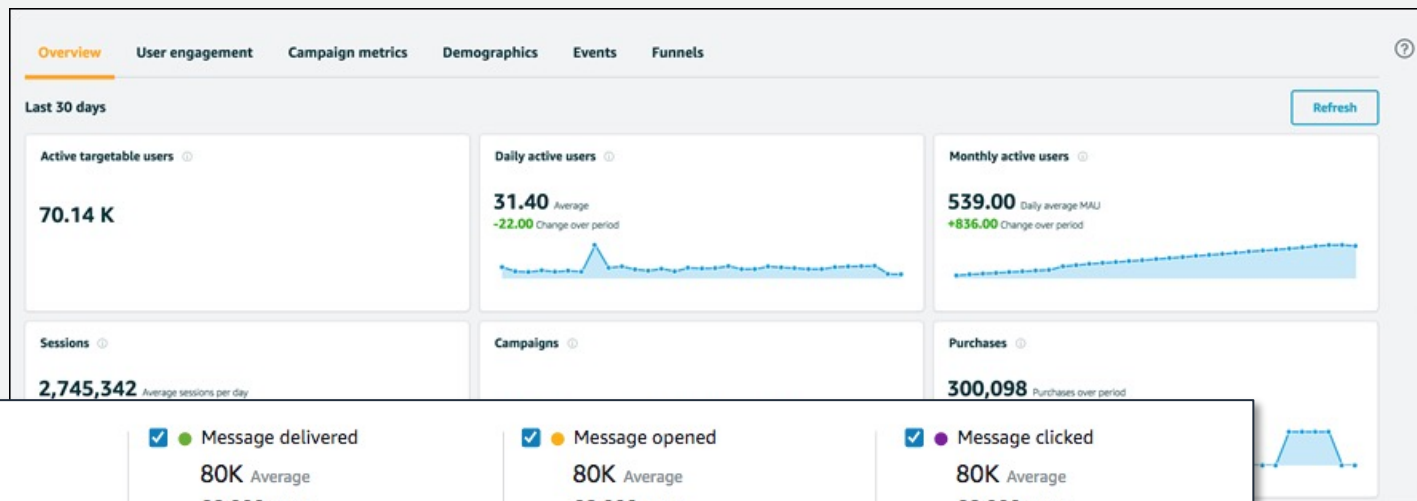


### ターゲティング通知

# 分析

Amazon Pinpointに保存された様々な分析データから顧客を把握可能

- アプリケーションの使用状況
- 収益
- イベント
- キャンペーン
- デモグラフィック
- ファネル
- トランザクション



# セグメントとキャンペーン

特定の属性をもった顧客グループであるセグメントを作成してキャンペーンを実施

## セグメントについて

- セグメントに属するユーザー情報を含むファイルをインポートしてユーザーセグメントを定義可能
- 事前にインポートした顧客リストからセグメントを作成可能
- 属性と演算子、値を複数追加することで特定の条件に絞ることが可能
- 既存セグメントをファイルにエクスポート可能

### セグメントを作成 [Info](#)

セグメントを構築  
顧客の属性により動的セグメントを作成します。

セグメントをインポート  
特定の受信者リストを含む CSV ファイルまたは JSON ファイルをインポートします。

---

#### セグメントの詳細

名前  
  
名前は 1~64 文字にする必要があります。

---

#### セグメントグループ 1 [Info](#)

セグメントグループには、基本セグメントに適用するフィルターが含まれています。インポートされたセグメントを基本セグメントとして選択した場合は、他のインポートされたセグメントを基本セグメントとして使用したり、セグメントグループを追加したりすることはできません。

基本セグメント [Info](#)

いずれかの対象者を含める  
 すべての対象者を含める

以下の **いずれか** に属する対象者を含めます:

Dynamic

---

#### 基準 - オプション [Info](#)

属性	演算子	値	
<input type="text" value="Company"/>	<input type="text" value="次を含む"/>	<input type="text" value="Q 値を入力"/>	<input type="button" value="削除"/>
		<input type="button" value="ExampleCorp. X"/>	

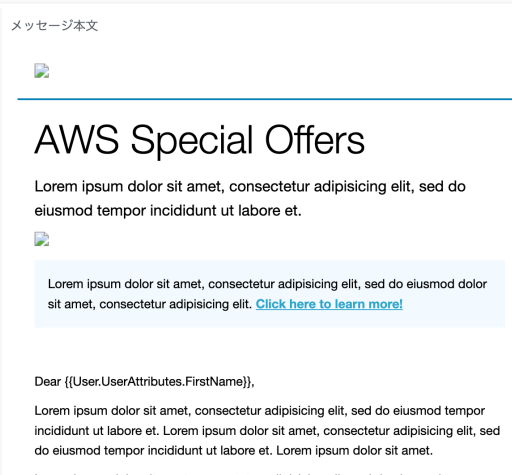
# セグメントとキャンペーン

特定の属性をもった顧客グループであるセグメントを作成してキャンペーンを実施

## キャンペーンについて

- 定義したスケジュールに従ってカスタマイズされたメッセージを送信可能
- メッセージ送信先はモバイルプッシュ、アプリケーション内、Eメール、SMS、カスタムチャネルから選択可能
- 作成したセグメントを指定することで特定条件のユーザーに対してメッセージを送信可能
- 特定のイベントが発生したときにメッセージを送信するようにキャンペーンを設定可能

詳細		
名前 my campaign	最終実行 実行日 March 18th 2022, 04:38 PM, UTC	キャンペーン配信 対象エンドポイント 1
タイプ 標準キャンペーン	結果 🟢 成功	処理済みエンドポイント 1
キャンペーンステータス 完了済み		
スケジュール		
開始日 起動時すぐ	タイムゾーン UTC+00:00 (UTC, GMT, WET)	頻度 起動時すぐ
	現地時間を尊重しますか? はい	
セグメントの詳細		
セグメント名 my Segment	オプトイン済みエンドポイント インポートされたソースセグメントを使用するセグメントの標準サイズを提供します。	ホールドアウト 0%

Eメールコンテンツ	
送信者のEメールアドレス takehina@amazon.co.jp	メッセージ本文
わかりやすい送信者名 - 件名 AWS Special Offers	 <p>AWS Special Offers</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et.</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod dolor sit amet, consectetur adipiscing elit. <a href="#">Click here to learn more!</a></p> <p>Dear {{User.UserAttributes.FirstName}},</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et. Lorem ipsum dolor sit amet.</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p>

# ジャーニー

## ユーザー体験を一連のステップで定義することが可能

- 使いやすいグラフィカルエディタを使用してジャーニーを作成可能
- ジャーニー内から時系列でアクティビティを設定
  - ユーザーへの通知方法
    - モバイルプッシュ、アプリケーション内、Eメール、SMS、カスタムチャネル
  - 分割
    - Yes/No
    - Multivariate
    - Random
- etc...
- ジャーニーの進捗はグラフで確認可能

The screenshot displays the Amazon Journey Builder interface. At the top, it shows 'Sample Journey' in 'DRAFT' status, with an 'アクション' (Action) dropdown and a 'レビュー' (Review) button. Below this, the journey flow is visualized as a series of steps: 'ジャーニーエントリ' (Journey Entry), 'Eメールを送信' (Send Email), 'はい/いいえ分類' (Yes/No Classification), 'Eメールを送信' (Send Email), '待機' (Wait), 'SMSメッセージを送信' (Send SMS Message), 'カスタムチャネル経由で送信' (Send via Custom Channel), 'ホールドアウト' (Holdout), and '終了 Info' (End Info). A summary dashboard on the left shows: 'Currently in journey: 6', 'Completed journey: 14', and 'Lifetime journey entries: 20'. A donut chart indicates 'Currently in journey: 30%'. A bottom right tooltip asks 'ジャーニーはいかがですか?' (How is the journey?).

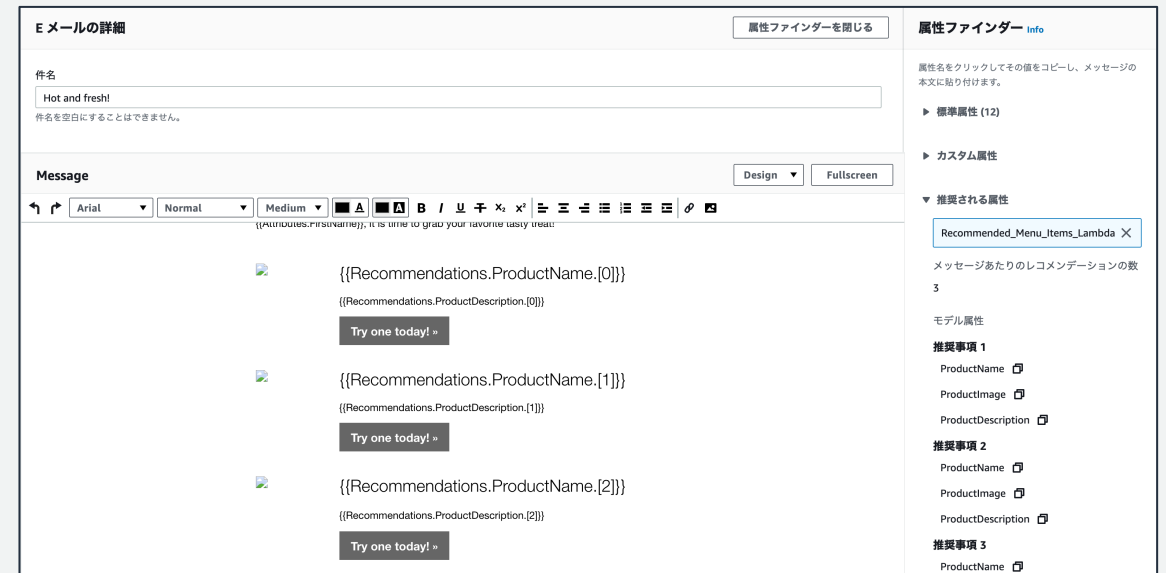
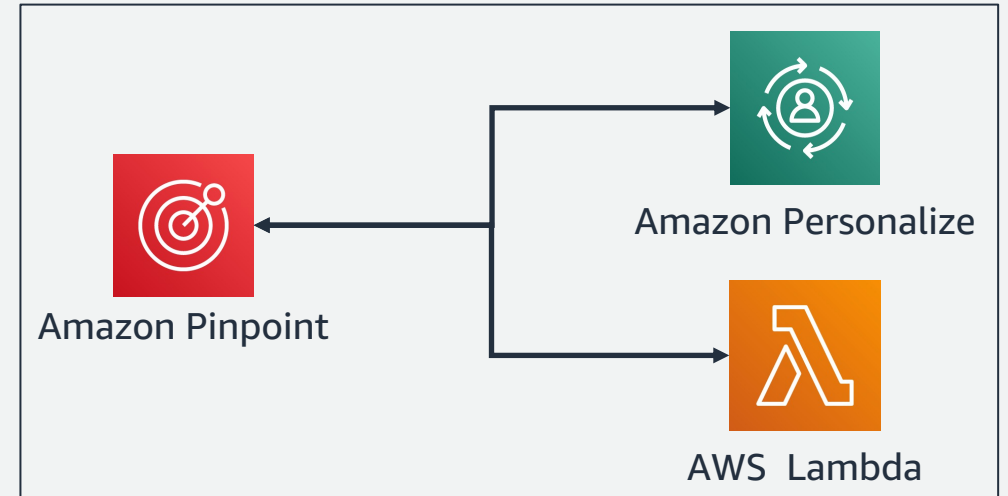
# Amazon Personalize を利用したメッセージのパーソナライズ

## Amazon Personalize とは

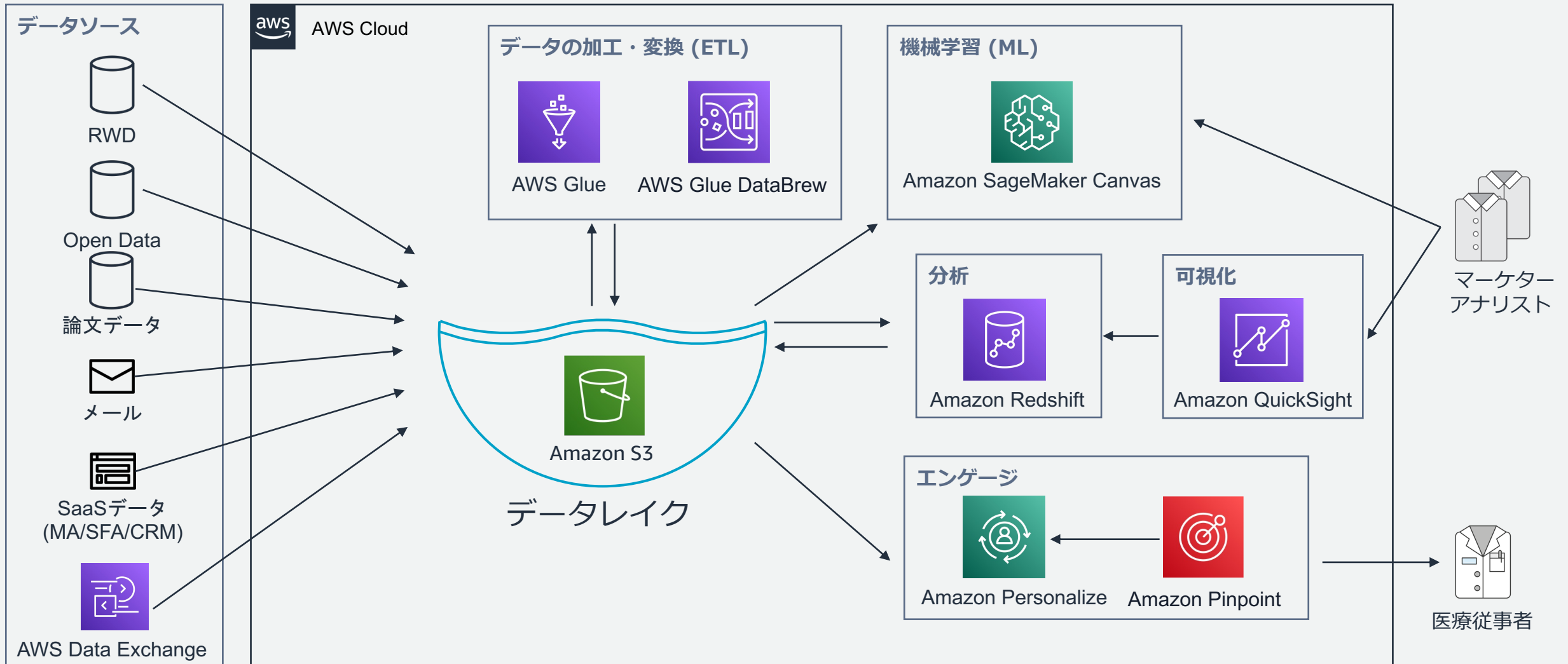
ユーザー向けにパーソナライズされたレコメンデーションを簡単に追加可能な機械学習サービス

## Amazon Pinpoint にパーソナライズ機能を追加するには

- Amazon PinpointとAmazon Personalizeを組み合わせることでユーザーごとのレコメンデーションをメッセージに追加することが可能
- Amazon Personalizeで作成したモデルはAmazon Pinpointのコンソールから指定することで設定可能



# コマース領域での利用を想定したデータレイク構成案



# まとめ



# まとめ

- **個別化した顧客のニーズ**を理解するためには様々なチャネルからデータを**データレイク**に収集して分析が必要
- AWSはデータレイクを中心としたデータ利活用のための**データ変換や分析、可視化、機械学習のサービス**を利用可能
- 収集したデータを活用したカスタマーエンゲージメントを行うためのAWSサービスで**効率的なユーザーとのコミュニケーション**をとることが可能

**ご検討頂く際は是非ご相談ください**



# Thank you!

ご清聴ありがとうございました