



[AWS Black Belt Online Seminar]

AWS Glue DataBrew

Junpei Ozono, Solutions Architect

2021.2.17

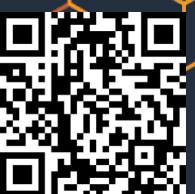
AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>



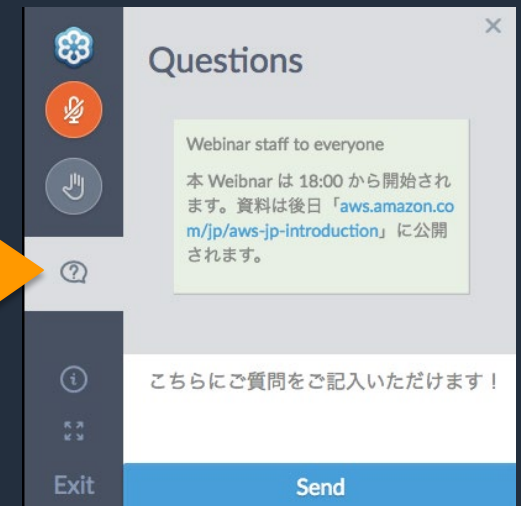
AWS Black Belt Online Seminar とは

「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

自己紹介

大園 純平 (おおその じゅんぺい)

 @jostandard

アマゾン ウェブ サービス ジャパン

アナリティクスソリューションアーキテクト



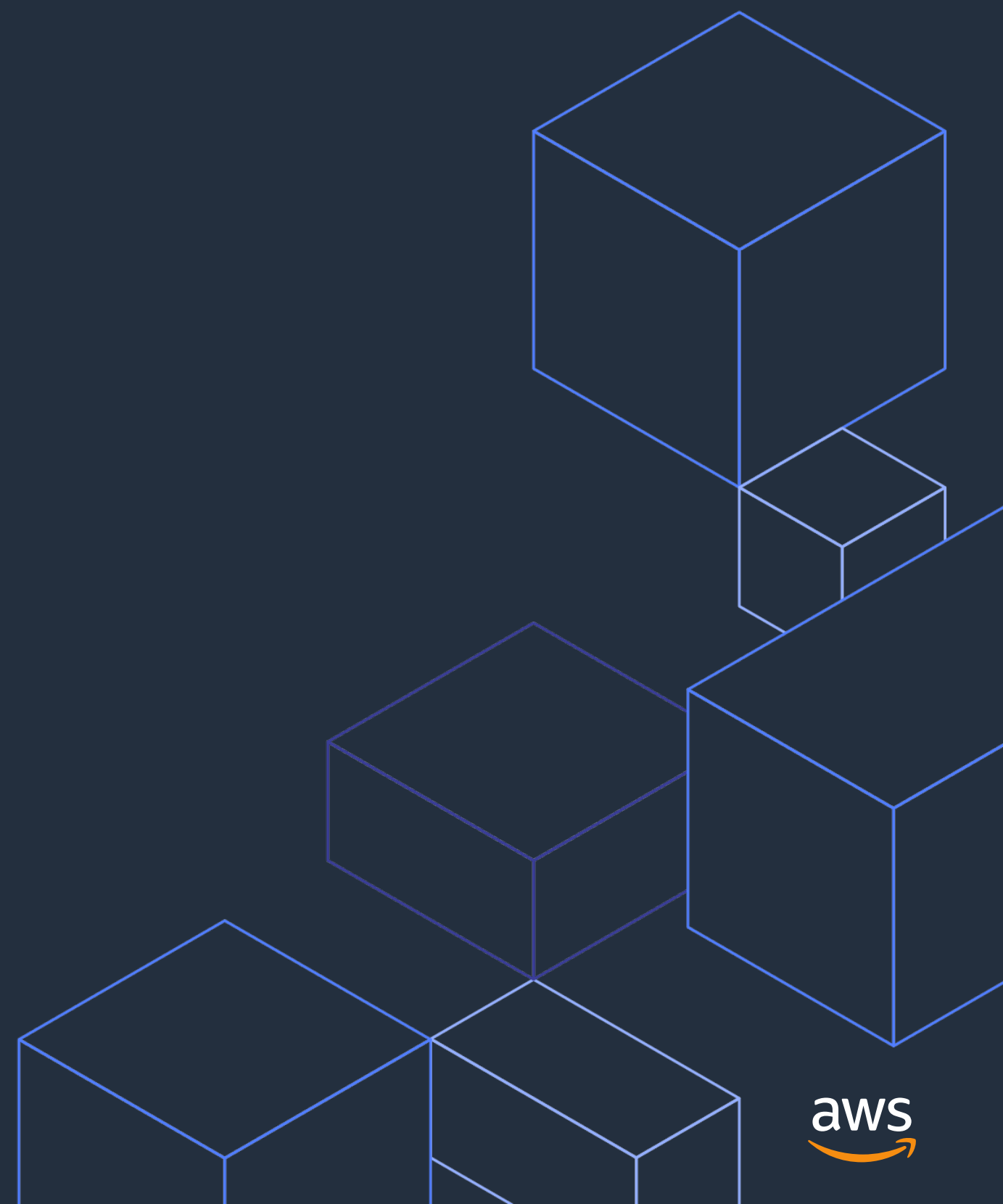
内容についての注意点

- 本資料では2021年2月17日現在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には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.

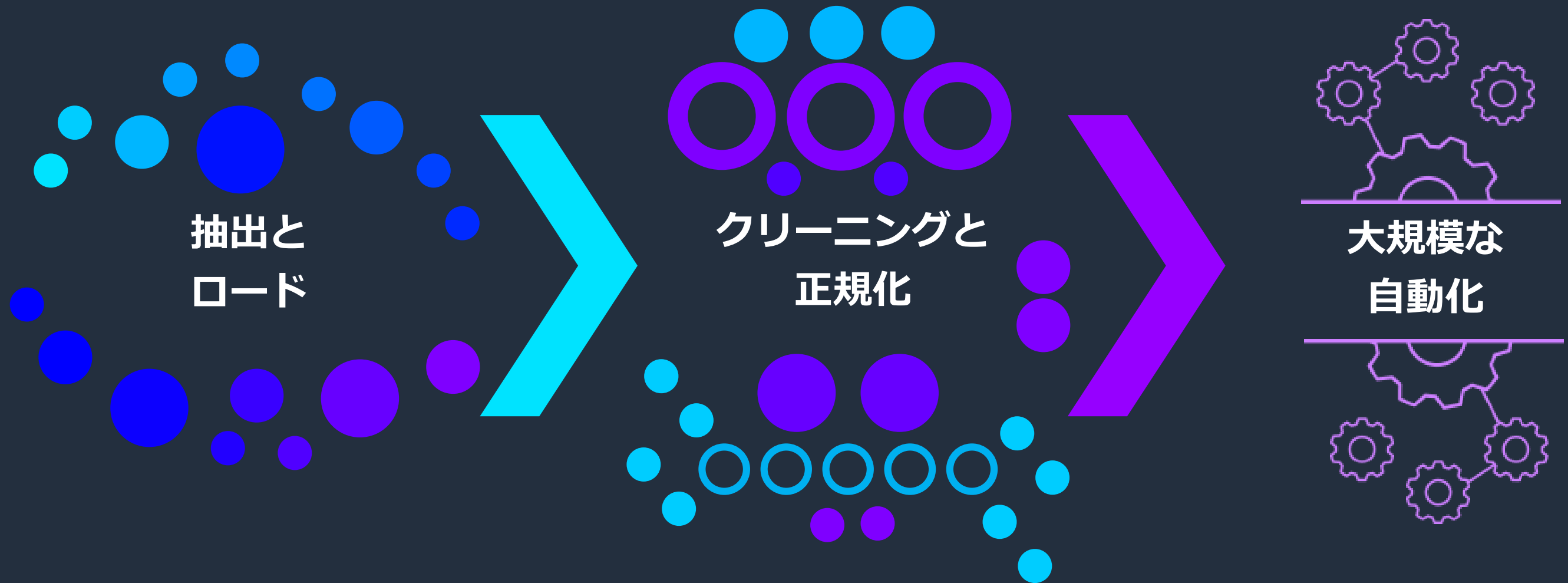
本日のアジェンダ

- データ準備の課題
- AWS Glue DataBrew 概要
- AWS Glue DataBrew の使い方
- AWS Glue DataBrew のユースケース
- AWS Glue DataBrew の料金
- まとめ

データ準備の課題



データ準備 (Data prep) には複雑なタスクを伴う



大規模に活用するためには複雑な ETL パイプラインの実装が必要

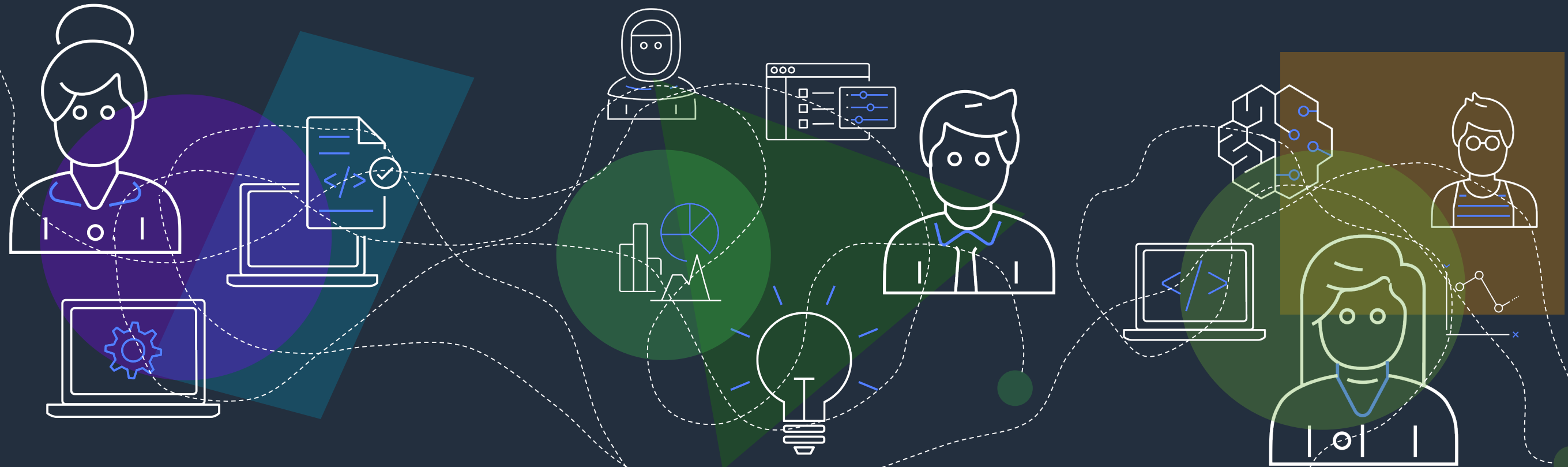
80% の時間がデータ準備に費やされている

データエンジニア

ETL 開発者

データアナリスト

データサイエンティスト



利用ユーザーに合った適切なツールが必要

典型的なデータ準備における課題

手動

繰り返しのワークフローを構築・運用するのは困難
スケールさせるには大規模コーディングが必要

大容量データの移動

組織間やシステム間での繰り返しのデータ移動が発生

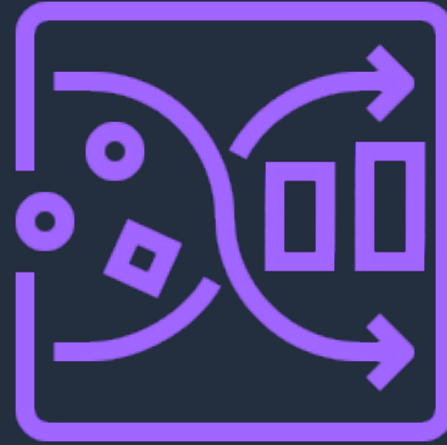
時間がかかる

大規模なデータの抽出、クレンジング、正規化、ロードを
マルチステップで行う必要がある

AWS Glue DataBrew 概要



AWS Glue DataBrew



データのクリーンアップおよび正規化を
最大 80% 高速化するビジュアルデータ準備ツール

データアナリストとデータサイエンティストのためのツール

高度なデータ準備機能をノンコーディングで利用可能



データ品質の理解

データパターンを理解し
異常を検出するために
プロファイリングを行い
データの品質を評価



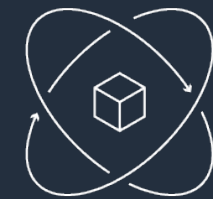
データのクリーンアップと正規化

250 種類以上の組み込みの変換処理から選択し、データの視覚化、クリーニング、正規化を実施



データリネージの視覚化

データソースと
変換手順を視覚化
してトラッキング



自動化

保存された変換手順を
使いまわしたり
自動実行する

AWS Glue DataBrew の使い方



AWS Glue DataBrew 用語の紹介

プロジェクト

データセットのクリーンアップや正規化などの変換に関するステップをまとめたレシピを作成するためのワークスペース

データセット

AWS Glue DataBrew が接続する、フィールド (列) を持つデータの集合

レシピ

データ変換ステップの一連のセット

ジョブ

データセットに対してレシピを適用して変換処理を行うもの (レシピジョブ)

データセットの統計に関するプロファイルを作成するもの (プロファイルジョブ)

AWS Glue DataBrew の使い方



事前準備 (IAM*)

データ変換処理の作成

ジョブの実行

- IAM ユーザー/グループ
- IAM ロール
- IAM ポリシー

- プロジェクトの作成
- データセットへの接続
- レシピの作成

- レシピジョブ
- プロファイルジョブ

* AWS Identity and Access Management

AWS Glue DataBrew の使い方



事前準備 (IAM*)

データ変換処理の作成

ジョブの実行

- IAM ユーザー/グループ
- IAM ロール
- IAM ポリシー

- プロジェクトの作成
- データセットへの接続
- レシピの作成

- レシピジョブ
- プロファイルジョブ

* AWS Identity and Access Management

IAM おさらい

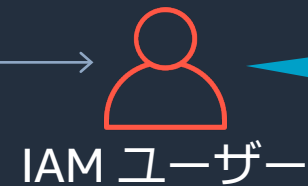
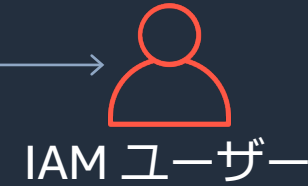
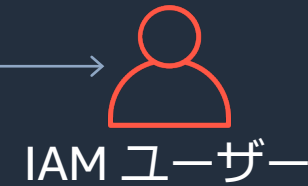
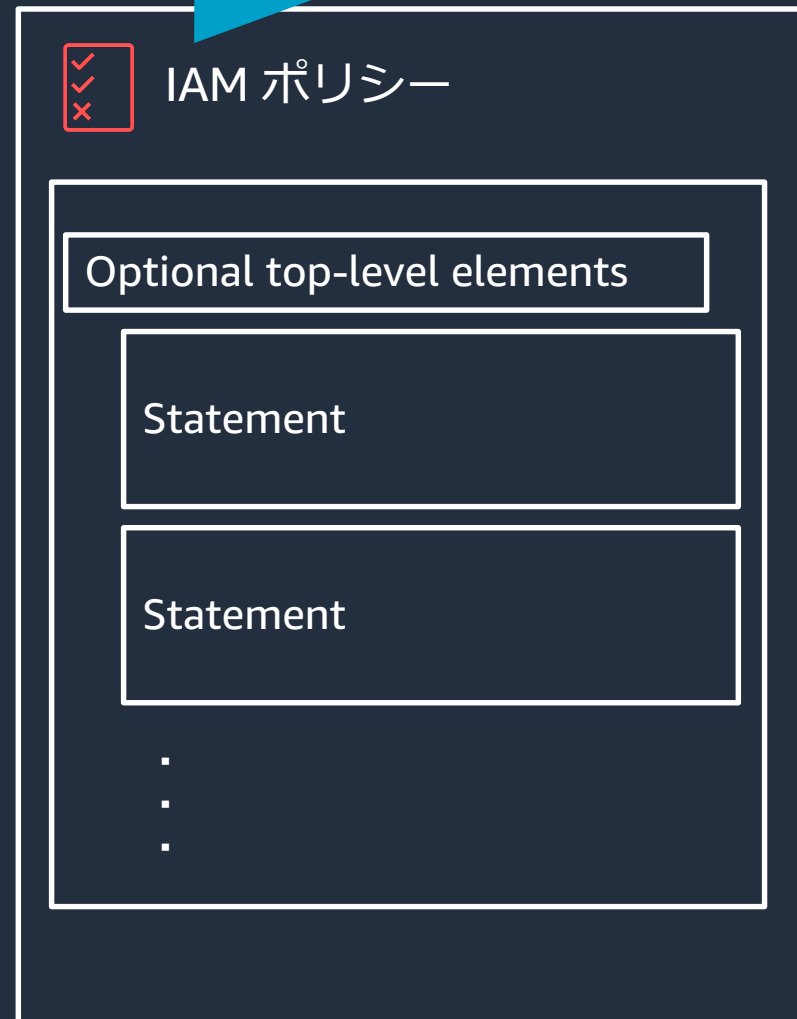
事前準備

データ変換処理の作成

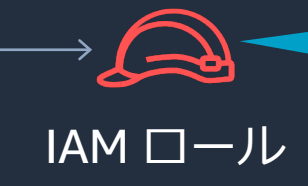
ジョブの実行

どのリソースにどの操作を許可するか権限を定義する

グループ内のユーザに対して特定の権限を付与する



ログインと特定の権限を付与する



特定のユーザや AWS サービスに対して権限を委任する

事前準備として必要なもの

1. AWS Glue DataBrew の利用者が
認証に使用するための IAM ユーザー/グループ
および IAM ユーザー/グループにアタッチする IAM ポリシー
2. AWS Glue DataBrew サービス自体が
他の AWS サービスにアクセスする際に使用する IAM ロール
および IAM ロールにアタッチする IAM ポリシー

IAM ユーザー/グループ準備

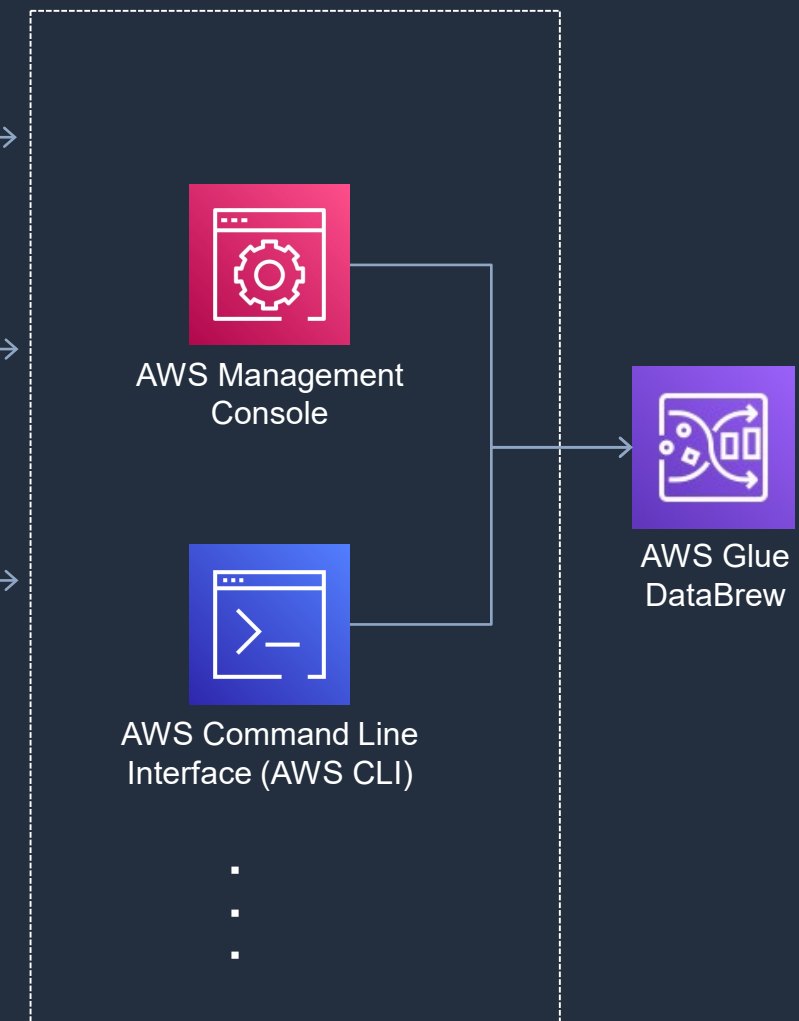
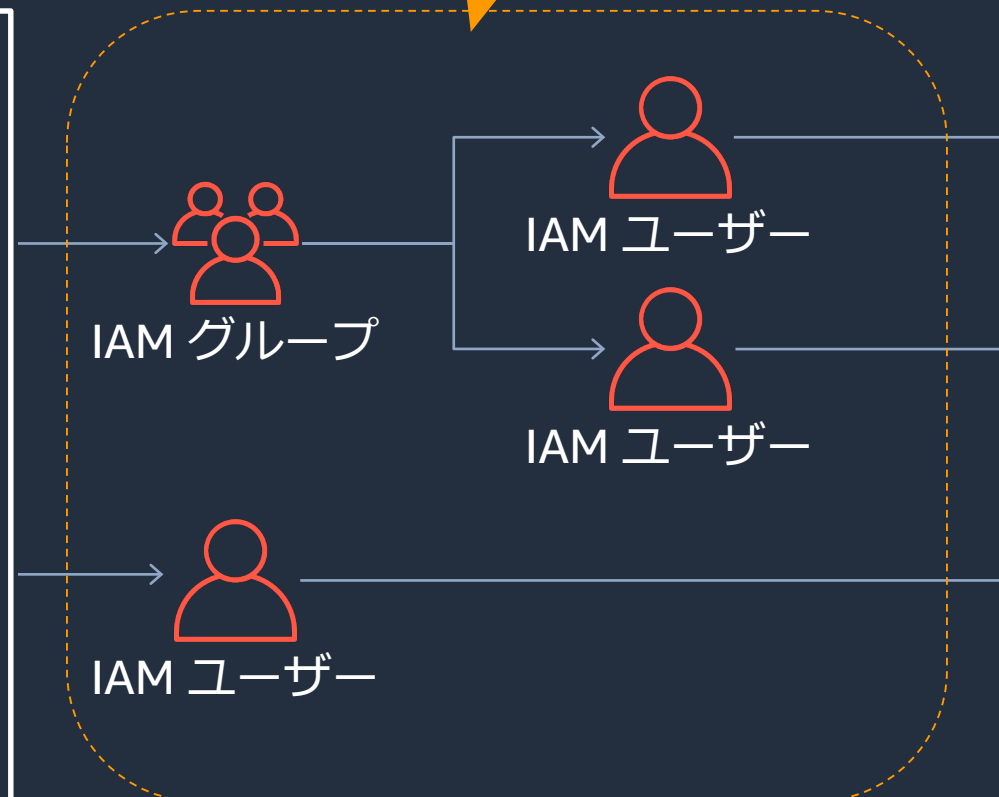
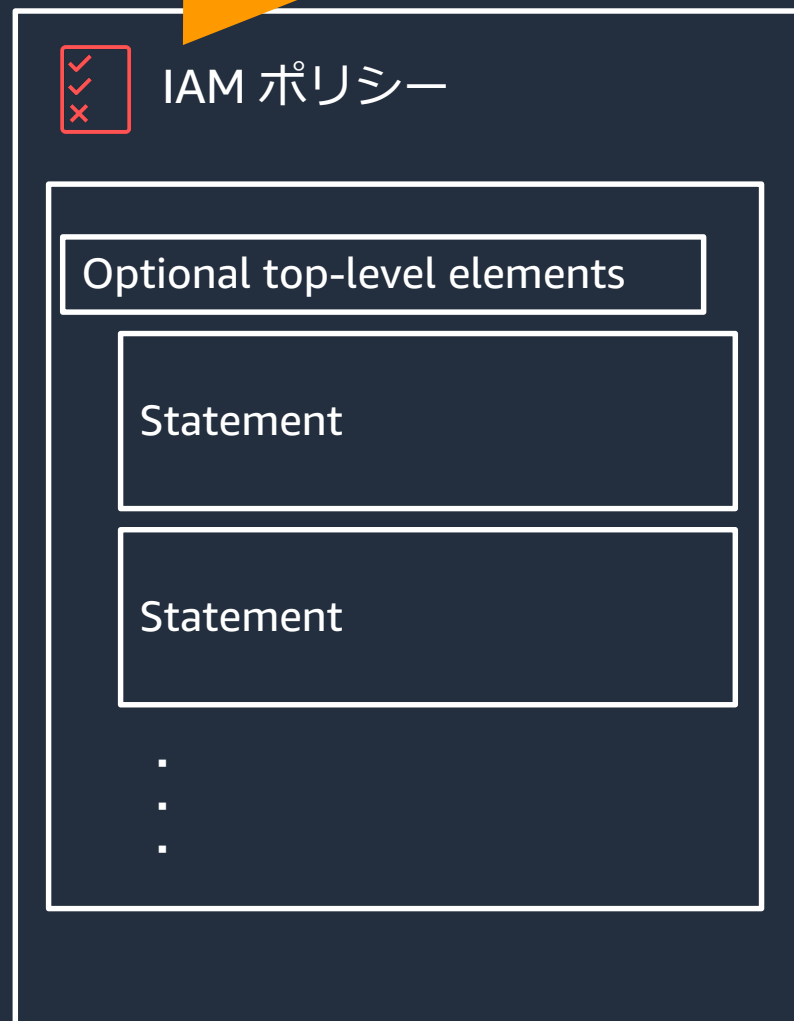
事前準備

データ変換処理の作成

ジョブの実行

1 IAM ユーザー/グループに
アタッチする必要な権限が
揃った IAM ポリシーを作成

2 AWS マネージメントコンソールやコマンドラインインターフェース (CLI) から
AWS Glue DataBrew に接続するための IAM ユーザー/グループを準備(作成)



3 IAM ポリシーを
IAM ユーザー/グループにアタッチ

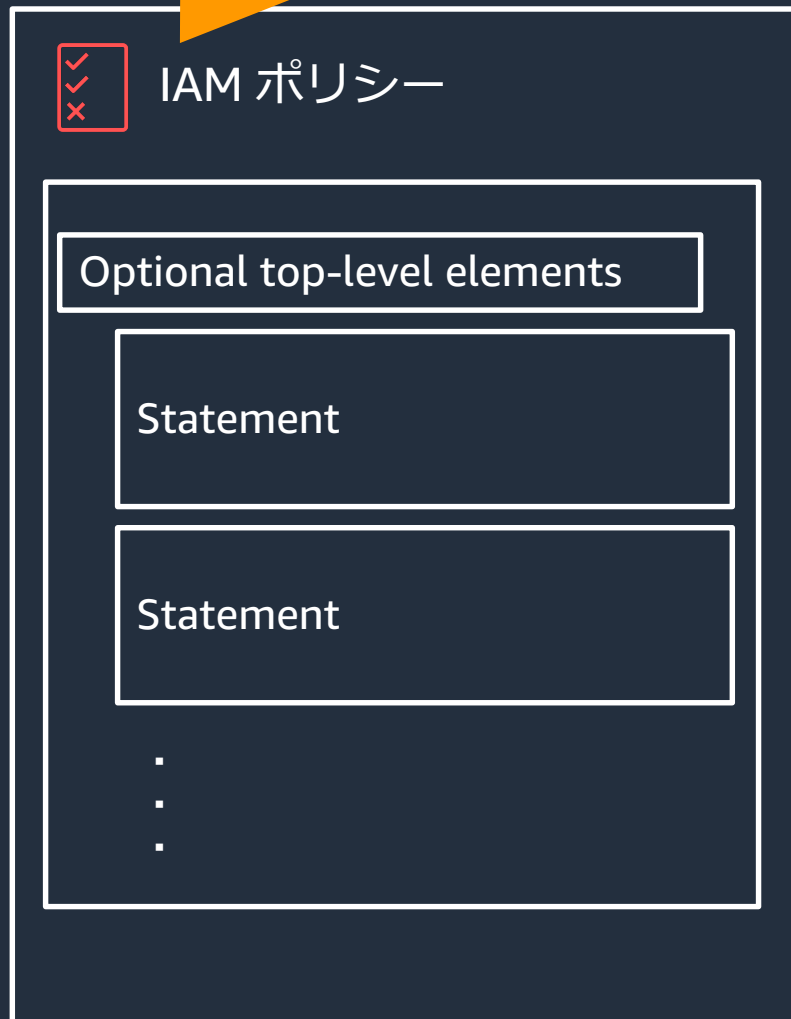
IAM ロール準備

事前準備

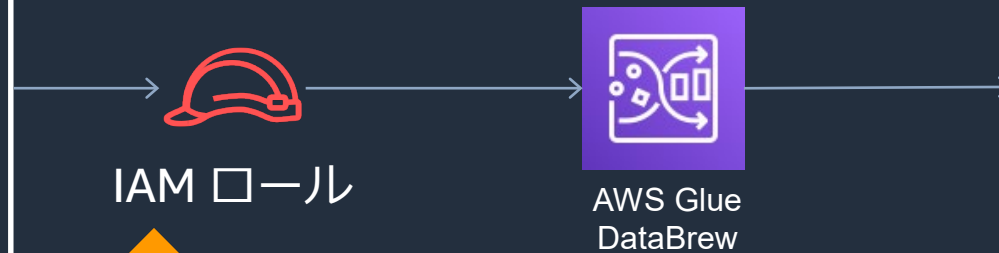
データ変換処理の作成

ジョブの実行

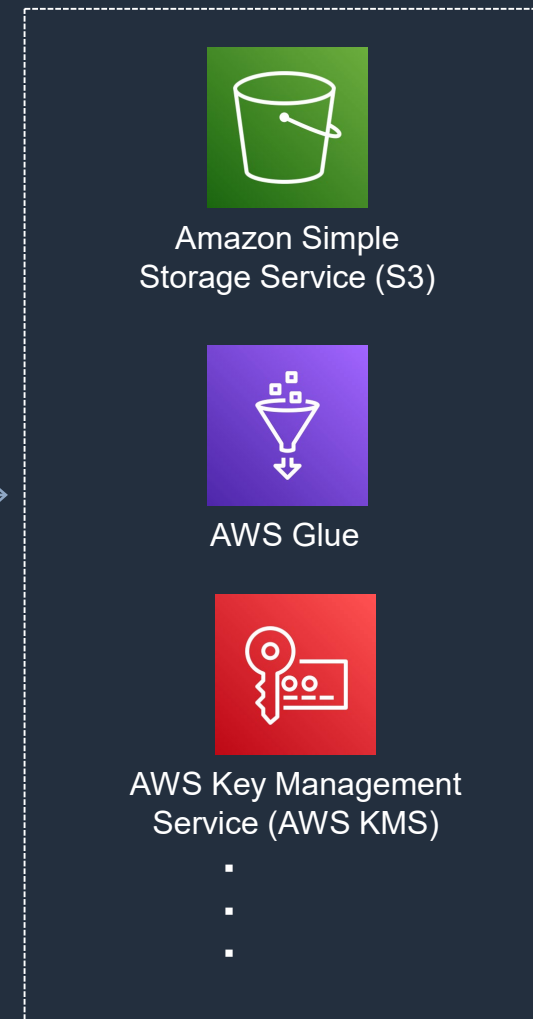
1 IAM ロールに必要な権限が揃った IAM ポリシーを作成



2 AWS Glue DataBrew からアクセスが必要な他の AWS サービスへのアクセス権限を定義した IAM ロールを準備 (作成)



3 IAM ポリシーを IAM ロールにアタッチ



AWS Glue DataBrew の使い方



事前準備 (IAM*)

データ変換処理の作成

ジョブの実行

- IAM ユーザー/グループ
- IAM ロール
- IAM ポリシー

- プロジェクトの作成
- データセットへの接続
- レシピの作成

- レシピジョブ
- プロファイルジョブ

* AWS Identity and Access Management

マネージメントコンソールに AWS Glue DataBrew の操作権限を持った IAM ユーザーでアクセスしてプロジェクトを作成する

アナリティクス

AWS Glue DataBrew データのクリーンアップと正規 化を最大 80% 高速化

AWS Glue DataBrew はビジュアルデータ準備ツールで、コードを記述することなくデータをクリーンアップして正規化し、分析や機械学習 (ML) 用のデータ準備にかかる時間を今日における従来のコードベースのデータ準備と比較して最大 80% 短縮できます。250 を超える事前構築済みの変換から選択して、異常のフィルタリング、標準形式へのデータの変換、無効な値の修正などのデータ準備タスクを自動化できます。すべてのタスクにおいて、コードを記述する必要はありません。

プロジェクトを作成する

データを使用して開始します。

プロジェクトを作成

サンプルデータセットの 1 つを使用して、データの準備と変換についてご覧ください。

サンプルプロジェクトを作成

プロジェクトとは

事前準備

データ変換処理の作成

ジョブの実行

プロジェクトは、特定のデータセットに対する変換ステップを定義する“レシピ”を作成するためのワークスペース

Sample project - 1

データセット: chess-games | サンプル: 最初の n 個のサンプル (39 個の行)

ジョブを作成

データセット | 元に戻す | やり直し | フィルタ | 列 | 形式 | クリーン | EXTRACT | 欠落 | 無効 | 重複 | 分割 | マージ | 作成 | 関数 | ネスト解除 | ビット | グループ | 結合 | ユニオン | テキスト | スケール | マッピング | エンコード

表示 17 個の列 | 39 個の行 | サンプル | グリッド | スキーマ | プロフィール

ABC id	rated	# created_at
一意 39	一意 2	一意 34
合計 39	合計 39	合計 39
15nlkNRf 1 2.56%	true 35 89.74%	
xwl41jna 1 2.56%	false 4 10.26%	
1zRjgcqm 1 2.56%		
その他のすべての値 36 92.31%		
15nlkNRf	true	1504140000000
xwl41jna	true	1503950000000
1zRjgcqm	true	1503800000000
vfnLULoh	true	1503690000000
ut6LIGPI	true	1503630000000
cqrATt9o	true	1503620000000
deKfSOPy	true	1503620000000
vQ1psjxd	true	1503600000000
k1qV5HHz	true	1503550000000
QQ3iIM2V	true	1503460000000

ズーム率 100%

レシピ (2)

Sample recipe - 1
作業バージョン

適用されたステップ2 | すべてクリア

1. フィルタ値 by white_rating
2. フィルタ値 by black_rating

プロジェクト作成手順①

レシピとデータセット

レシピ

- 新しいレシピを作成
- 既存のレシピを編集
- レシピからステップをインポート

データセット

- マイデータセット
- サンプルファイル
- 新しいデータセット

事前準備

データ変換処理の作成

ジョブの実行

DataBrew > プロジェクト > プロジェクトを作成

プロジェクトを作成

プロジェクトの詳細

プロジェクト名

プロジェクト名は 1~255 文字にする必要があります。有効な文字は、英数字 (A~Z、a~z、0~9)、ハイフン (-)、ピリオド (.), およびスペースです。

レシピの詳細

DataBrew のデータクリーンアップステップはレシピとして保存されます。レシピはデフォルトでプロジェクトに接続されます。プロジェクトが関連付けられていない既存のレシピをプロジェクトに適用することもできます。

アタッチされたレシピ

レシピ名

レシピ名は 1~255 文字にする必要があります。有効な文字は、英数字 (A~Z、a~z、0~9)、ハイフン (-)、ピリオド (.), およびスペースです。

レシピからステップをインポートする
既存のレシピからプロジェクトにレシピをインポートします。選択した既存のレシピは編集されません。

データセットを選択

作業するデータセットを選択します

- マイデータセット
インポートされたデータセット
- サンプルファイル
データセットのサンプルファイルを調べる
- 新しいデータセット
新しいデータセットのインポート

プロジェクト作成手順②

データセットの選択

加工/変換したいデータを以下の
中から選択

- ローカルファイル
- Amazon S3 上のファイル
- AWS Glue データカタログ
- AWS Data Exchange

事前準備

データ変換処理の作成

ジョブの実行

新しいデータセットへの接続

↑ ファイルを上ロード

アップロードするファイルを選択します

🔗 ファイルを選択

CSV、TSV、JSON、JSONL、Parquet、または .XLSX 形式で 1 つのファイルを上ロード

S3 送信先を入力

既存の S3 送信先を選択します。選択したファイルはこの S3 パケットにアップロードされます

S3 を参照

形式は s3://bucket/prefix/ です

▼ 追加設定

暗号化

アップロードされたファイルの暗号化を有効にする

SSE-S3 または AWS KMS を使用して、アップロードされたファイルを暗号化します

プロジェクト作成手順③

行サンプリング/アクセス許可

プロジェクト内で操作するデータセットの行数をサンプリング可能

データセットに対するアクセス許可を指定

- 新しい IAM ロールを作成
- 既存の IAM ロールを選択

「事前準備」で IAM ロールを作成した場合はこちらを選択

事前準備

データ変換処理の作成

ジョブの実行

▼ サンプルング - オプション
サンプルのタイプとサイズを選択します

タイプ
最初の n 行

サンプルする行の数を教えてください。

500
 1,000
 2,500
 カスタムサイズ

▶ タグ - オプション
定義して AWS リソースに割り当てることができるメタデータ。各タグは、お客様が定義したキー（名前）とオプションの値で構成されるシンプルなラベルです。タグを使用すると、目的、所有者、環境、またはその他の条件によるリソースの管理、検索、フィルタリングが簡単になります。

アクセス許可
DataBrew needs permission to connect to data on your behalf. Use an IAM role with the [必須ポリシー](#) attached.

ロール名
データに接続するためのアクセス権を持つロールを選択します。最新の更新を表示するには更新します。

AwsGlueDataBrewDataAccessRole

サポートされるデータセット

事前準備

データ変換処理の作成

ジョブの実行

インプットファイルのサポートフォーマットと拡張子

フォーマット	拡張子 (非圧縮) *1	拡張子 (圧縮) *1
CSV *2	.csv	.csv.gz, .csv.snappy, .csv.lz4, .csv.bz2, .csv.deflate
TSV *2	.tsv	.tsv.gz, .tsv.snappy, .tsv.lz4, .tsv.bz2, .tsv.deflate
Microsoft Excel ワークブック	.xlsx	非サポート
JSON	.json	.json.gz, .json.snappy, .json.lz4, .json.bz2, .json.deflate
JSON lines	.jsonl	.jsonl.gz, .jsonl.snappy, .jsonl.lz4, .jsonl.bz2, .jsonl.deflate
Apache Parquet	.parquet	.parquet.gz, .gz.parquet, .parquet.snappy, .snappy.parquet, .parquet.lz4, .lz4.parquet

*1 DataBrew は拡張子でファイルフォーマットを判断するため、必ず上記拡張子を使用する

*2 区切り文字は Comma (,), Colon (:), Semi-colon (;), Pipe (|), Tab (¥t), Caret (^), Space に対応

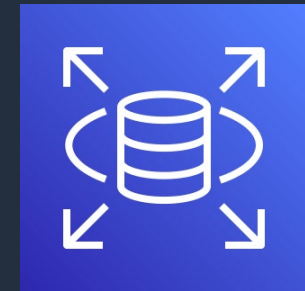
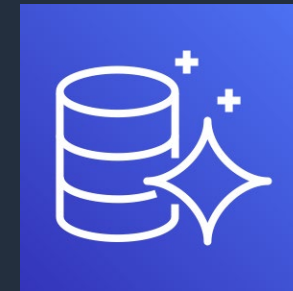
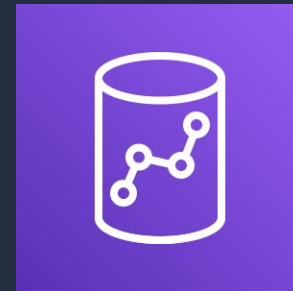
Amazon S3 上の特定ファイル/フォルダを示す「S3 パス」または正規表現を用いた「パラメータ化された S3 パス」を指定可能

例

- ある特定のファイルを指定
`s3://bucket-name/inventory-data.csv`
- ある特定のフォルダ配下にあるすべてのファイルを指定
`s3://bucket-name/folder-name/`
- "2021" を名称に含むフォルダ配下にあるすべてのファイルを指定
`s3://bucket-name/<.*>2021<.*>/`

AWS Glue データカタログ経由で以下サービス上のデータと接続可能

- Amazon Redshift
- Amazon Aurora MySQL
- Amazon Aurora PostgreSQL
- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL



レシピの作成

事前準備

データ変換処理の作成

ジョブの実行

プロジェクト内でインタラクティブに変換イメージを確認しながら、データセットに対する変換ステップのコレクションであるレシピを作成

Sample project - 1 (1)

データセット: chess-games | サンプル: 最初の n 個のサンプル (7 個の行)

元に戻す やり直し フィルタ 列 形式 クリーン EXTRACT 欠落 無効 重複 分割 マージ 作成 関数 ネスト解除 ピボット グループ 結合 ユニオン テキスト スケール マッピング エンコード

ABC winner	ABC victory_status	ABC winner_count
一意 3	一意 3	一意 5
black 3 42.86%	time ran out 3 42.86%	2 28.57%
white 3 42.86%	checkmate 2 28.57%	3 28.57%
draw 1 14.29%	other player resigned 2 28.57%	13 14.29%
		その他のすべての値 2 28.57%
black	checkmate	2
white	checkmate	3
white	other player resigned	13
black	other player resigned	14
black	time ran out	2
draw	time ran out	1
white	time ran out	3

レシピ (7)

Sample recipe - 1 (2)
バージョン 2.0

適用されたステップ 7 | すべてクリア

1. フィルタ値 by white_rating
2. フィルタ値 by black_rating
3. グループ化 winner, victory_status and 作成 winner_count 開始位置 COUNT(winner)
4. フィルタ値 by victory_status
5. テキストを置き換える mate と checkmate in victory_status
6. テキストを置き換える resign と other player resigned in victory_status
7. テキストを置き換える outoftime と time ran out in victory_status

1. 250 種類以上の組み込みの処理から選択して変換ステップを作成

2. 変換ステップが確定したらレシピを発行

- 作成したレシピは編集・削除でき、バージョン管理も可能
- レシピは YAML/JSON でのダウンロード, JSON のアップロードも可能

DataBrew > レシピ

▼ レシピとは

データ変換ステップのコレクションはレシピと呼ばれ、DataBrew プロジェクトで作成または編集され、スタンドアロンエンティティとして発行できます。

発行されたレシピは、データのコンテキストがない変換ステップのみのテンプレートです。ユーザーは、作業中にプロジェクトから複数のバージョンのレシピを発行できます。レシピは、レシピジョブを通じてプロジェクトの他のデータセットにダウンロードまたは適用できます。

レシピ (4)

YAML としてダウンロード JSON 形式でダウンロード このレシピでジョブを作成する アクション ▼ **レシピをアップロード**

🔍 Sample ✕ 1 件の一致 発行済み ▼ < 1 > ⚙️

<input checked="" type="checkbox"/>	レシピ名	バージョンの説明	関連付けられたプロジェクト	発行日	パブリッシャー	タグ
<input checked="" type="checkbox"/>	Sample recipe - 1 発行済みバージョン 2.0	Second version of the recipe.	Sample project - 1			

代表的な変換処理

データのフィルタリング

事前準備

データ変換処理の作成

ジョブの実行

chess-project
ジョブを作成
システム アクション

データセット: chess-games | サンプル: 最初の n 個のサンプル (39 個の行)
2 レシビ

元に戻す やり直し
フィルタ
列
形式 クリーン EXTRACT
欠落 無効 重複
分割 マージ 作成
関数
ネスト解除 ピボット グループ 結合 ユニオン
テキスト スケール マッピング エンコード

表示 17 個の列 39 個の行
ハイライトされているもののみを表示
サンプル
グリッド
スキーマ
プロフィール

turns

一意 31 合計 38

統計	値
Min	34
中央値	69.5
平均値	79.47
Mode	69
Max	178

ソース	ABC victory_status	ABC winner	ABC increment_code
一意	3	合計	38
resign	27	71.05%	
outoftime	6	15.79%	
mate	5	13.16%	
一意	3	合計	38
white	19	50%	
black	18	47.37%	
draw	1	2.63%	
一意	14	合計	38
15+0	11	28.95%	
10+0	11	28.95%	
10+10	3	7.89%	
その他のすべての値	13	34.21%	

フィルタ値

ソース列
大文字と小文字を変更する列の名前

victory_status

フィルタ条件
でない

カスタム値を入力 正規表現値を入力

フィルタ値を入力

draw X

Find

一意の値 (1) 1 個の選択した値を表示

- resign 27 69%
- outoftime 6 15%
- mate 5 12%
- draw 1 2%

データの結合

事前準備

データ変換処理の作成

ジョブの実行

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

システム 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

レシビ

ジョブ

最新機能

コミュニティ

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

テーブル B
airbnb-reviews

id

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

データの集計

事前準備

データ変換処理の作成

ジョブの実行

Sample project - 1

データセット: chess-games | サンプル: 最初の n 個のサンプル (39 個の行)

ジョブを作成

システム アクション

グループ

列リスト

グループ化されたテーブルの集計を含む列を追加する

列名	集計	新しい列名	新しい列タイプ	
ABC winner	グループ化	winner	ABC 文字列	削除
ABC victory_status	グループ化	victory_status	ABC 文字列	削除
ABC winner	Count	winner_count	# Integer	削除

別の列を追加

グループタイプ

- 新しいテーブルとしてグループ化 (既存のすべての列を新しい列で置き換えます)
- 新しい列としてグループ化 (新しい列は既存の列に追加されます)

グループテーブルのプレビュー

ABC winner	ABC victory_status	# winner_count
black	mate	2
draw	draw	1
white	mate	3
white	resign	13
black	resign	14
black	outoftime	2
draw	outoftime	1

キャンセル 終了

欠損値の補完

事前準備

データ変換処理の作成

ジョブの実行

データセット

元に戻す やり直し フィルタ 列 形式 クリーン EXTRACT 欠落 無効 重複 分割 マージ 作成 関数 ネスト解除 ピボット グループ 結合 ユニオン テキスト スケール マッピング エンコード

レシビ

表示 39 個の列 500 個の行 ハイライトされているもののみを表示 サンプル

グリッド スキーマ プロフィール

ABC chembl_id	ソース	プレビュー	ABC updated_by
一意 500 合計 500	一意 312 合計 422	一意 312 合計 500	一意 38 合計 424
CHEMBL1282976 1 0.2%	2021-02-17 00:00:00 78 15.6%	2021-02-17 00:00:00 78 15.6%	autoloader 235 47%
CHEMBL3083642 1 0.2%	2013-02-14 14:00:47 41 8.2%	2013-02-14 14:00:47 41 8.2%	null 76 15.2%
CHEMBL1665725 1 0.2%	2004-10-31 12:48:03 13 2.6%	2004-10-31 12:48:03 13 2.6%	set cell-type from tid 43 8.6%
その他のすべての値 497 99.4%	その他のすべての値 368 73.6%	その他のすべての値 368 73.6%	その他のすべての値 146 29.2%
CHEMBL3878869 null	2021-02-17 00:00:00	2021-02-17 00:00:00	null
CHEMBL917320 2013-02-14 14:00:11	2013-02-14 14:00:11	2013-02-14 14:00:11	set tissue from tid
CHEMBL856031 2010-10-26 17:10:29	2010-10-26 17:10:29	2010-10-26 17:10:29	SQL-abbr-desc
CHEMBL4036079 null	2021-02-17 00:00:00	2021-02-17 00:00:00	null
CHEMBL2156960 2013-04-08 15:03:54	2013-04-08 15:03:54	2013-04-08 15:03:54	set tissue from tid
CHEMBL732336 2013-12-05 14:18:48	2013-12-05 14:18:48	2013-12-05 14:18:48	george_spelling
CHEMBL3788634 2016-12-19 13:30:13	2016-12-19 13:30:13	2016-12-19 13:30:13	autoloader
CHEMBL1274171 2011-04-08 10:46:23	2011-04-08 10:46:23	2011-04-08 10:46:23	SQL_unknown_sp
CHEMBL2400679 2014-01-07 13:09:06	2014-01-07 13:09:06	2014-01-07 13:09:06	autoloader
CHEMBL1931503 2014-02-14 00:00:00	2014-02-14 00:00:00	2014-02-14 00:00:00	AH_assay_types
CHEMBL3630515 2016-05-09 15:29:11	2016-05-09 15:29:11	2016-05-09 15:29:11	autoloader
CHEMBL3404541 2015-09-10 13:00:16	2015-09-10 13:00:16	2015-09-10 13:00:16	autoloader
CHEMBL1817241 2012-02-01 12:46:11	2012-02-01 12:46:11	2012-02-01 12:46:11	autoloader
CHEMBL2434217 2014-01-07 15:51:20	2014-01-07 15:51:20	2014-01-07 15:51:20	autoloader
CHEMBL3881540 null	2021-02-17 00:00:00	2021-02-17 00:00:00	null
CHEMBL2445826 2014-01-07 16:10:50	2014-01-07 16:10:50	2014-01-07 16:10:50	autoloader
CHEMBL4259571 null	2021-02-17 00:00:00	2021-02-17 00:00:00	null
CHEMBL2091654 2012-11-02 16:02:54	2012-11-02 16:02:54	2012-11-02 16:02:54	fix_type
CHEMBL3538847 2015-12-04 14:44:48	2015-12-04 14:44:48	2015-12-04 14:44:48	autoloader
CHEMBL794391 2004-10-31 12:48:03	2004-10-31 12:48:03	2004-10-31 12:48:03	admin

欠落した値

ソース列
欠落した値を持つ列の名前
updated_on

欠落した値アクション
欠落した値に対して実行するアクション

- 欠落した値がある行を削除する
- 空の値で埋める
- null で埋める
- 最後の有効な値で埋める
- 最も頻繁な値で埋める
- カスタム値で埋める
- 数値集計で埋める

カスタム値
2021/02/17

変換を適用

- すべての行 (500 行)
変換はデータセット内のすべての行に適用されます
- フィルタリングされた行 - 適用された 0 フィルタ (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)

表示されるプレビュー

キャンセル 適用

関数を使った新たな列の作成

事前準備

データ変換処理の作成

ジョブの実行

sales-data-etl
ジョブを作成

データセット: sales-data | サンプル: 最初の n 個のサンプル (500 個の行)
システム アクション

元に戻す やり直し
フィルタ
列
形式 クリーン EXTRACT
欠落 無効 重複
分割 マージ 作成

関数
ネスト解除
ピボット
グループ
結合
ユニオン

テキスト
スケール
マッピング
エンコード

表示
16 個の列
▼
500 個の行
□ ハイライトされているもののみを表示

グリッド
スキーマ
プロフィール

ABC	Date	ABC	Target	Close
一意	266	合計	500	一意
	10/17/2011		6	1.2%
	7/21/2011		5	1%
	9/29/2011		5	1%
その他のすべての値		484	96.8%	その他のすべての値
	1/2/2011		2/2/2011	
	1/3/2011		4/9/2011	96.0
	1/6/2011		5/4/2011	118.0
	1/11/2011		2/14/2011	34.0
	1/11/2011		1/31/2011	20.0
	1/12/2011		3/16/2011	63.0
	1/13/2011		2/18/2011	36.0
	1/16/2011		3/6/2011	49.0
	1/19/2011		3/3/2011	43.0
	1/25/2011		2/27/2011	33.0
	1/26/2011		2/23/2011	28.0
	1/29/2011		2/24/2011	26.0
	1/30/2011		2/27/2011	28.0
	1/31/2011		7/24/2011	174.0
	2/3/2011		4/30/2011	86.0
	2/3/2011		2/3/2011	0.0
	2/5/2011		2/5/2011	0.0
	2/6/2011		6/29/2011	143.0
	2/13/2011		4/1/2011	47.0
	2/15/2011		3/25/2011	38.0
	2/15/2011		2/15/2011	0.0

Forecasted Monthly Revenue

合計 500

一意 309

平均値 16.09 K Mode 0 Max 814.8 K

関数を選択

2つのソース列間の日付単位の差。

- 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 >

列を作成

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

Date

値 2

カスタム値を入力 ソース列を選択

Target Close

単位

日

宛先列

抽出された値を使用して作成された列の名前

Date_DATEDIFF

有効な文字は、英数字、アンダースコア、スペースです。

複数列の統合



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

元に戻す やり直し フィルタ 列 形式 クリーン EXTRACT 欠落 無効 重複 分割 マージ 作成 関数 詳細

2 レシビ

表示 15 個の列 | 500 個の行 | ハイライトされているもののみを表示
サンプル
グリッド スキーマ プロフィール

ソース		ソース		プレビュー		プレビュー	
#	latitude	#	longitude	ABC	lat_lon	ABC	room_type
一意	446	一意	442	一意	460	一意	3
合計 500		合計 500		合計 500		合計 500	
Min	35.55	Min	139.27	35.74012, 139.89638	7	Entire home/apt	321
中央値	35.7	中央値	139.71	35.68288, 139.81552	6	Private room	157
平均値	35.7	平均値	139.7	35.70399, 139.59775	4	Shared room	22
Mode	35.74	Mode	139.9	その他のすべての値	483		
Max	35.79	Max	139.9				
				35.71721, 139.82596		Entire home/apt	
				35.73844, 139.76917		Private room	
				35.70865, 139.69681		Entire home/apt	
				35.65833, 139.67153		Private room	
				35.74253, 139.7973		Private room	
				35.69098, 139.70618		Private room	
				35.74409, 139.79895		Private room	
				35.65111, 139.72165		Entire home/apt	
				35.60682, 139.67629		Private room	
				35.7385, 139.85167		Entire home/apt	
				35.70099, 139.74012		Entire home/apt	
				35.62198, 139.69987		Entire home/apt	
				35.72597, 139.7016		Private room	
				35.72607, 139.70363		Private room	
				35.73626, 139.85024		Entire home/apt	

列をマージ

ソース列
マージする順序で 2 つ以上の列を選択します。

- latitude
- longitude

列を追加する

セパレータ - オプション
連結された値は、この

新しい列名
マージ先のターゲット列の名前

lat_lon

有効な文字は、英数字、アンダースコア、スペースです。

変換を適用

- すべての行 (500 行)
変換はデータセット内のすべての行に適用されます
- フィルタリングされた行 - 適用された 0 フィルタ (500/5)
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

フラグ値の作成

事前準備

データ変換処理の作成

ジョブの実行

chess-project
ジョブを作成
システム アクション

データセット: chess-games | サンプル: 最初の n 個のサンプル (39 個の行)
2 レジビ

元に戻す やり直し フィルタ 列 形式 クリーン EXTRACT 欠落 無効 重複 分割 マージ 作成 関数 ネスト解除 ピボット グループ 結合 ユニオン テキスト スケール マッピング エンコード

列にフラグを立てる
関数に基づく

グリッド スキーマ プロフィール

表示 18 個の列 ▼ 39 個の行 ハイライトされているもののみを表示

opening_eco		opening_name		Related to Defense?		# opening_ply	
一意	28	一意	31	一意	2	一意	10
合計	39	合計	39	合計	39	合計	39
B50	4	Sicilian Defense	4	True	32	Min	1
B30	3	French Defense: Knight Variation	2	False	7	中央値	5
C55	2	Ruy Lopez: Morphy Defense Classical Defen...	2			平均値	5.26
その他のすべての値	30	その他のすべての値	31			Mode	4
	76.92%		79.49%			Max	12

列を作成

列オプションを作成

フラグ値 ▼

フラグを設定する値

欠落した値

列の重複する値を削除

行を複製

カスタム値

ソース列

値を抽出する列を選択

opening_name ▼

フラグを設定する値

Defense

文字列値または正規表現を入力

フラグ値

True または False ▼

宛先列

抽出された値を使用して作成された列の名前

Related to Defense?

有効な文字は、英数字、アンダースコア、スペースです。

One-hot エンコーディング

事前準備

データ変換処理の作成

ジョブの実行

chess-project
ジョブを作成

データセット: chess-games | サンプル: 最初の n 個のサンプル (39 個の行)

元に戻す やり直し フィルタ 列 形式 クリーン EXTRACT 欠落 無効 重複 分割 マージ 作成 関数 ネスト解除 ピボット グループ 結合 ユニオン テキスト スケール マッピング エンコード

表示 20 個の列 39 個の行 ハイライトされているもののみを表示

One-Hot-Encode 列

ソース	プレビュー	プレビュー	プレビュー
ABC victory_status	# victory_status_mate	# victory_status_outoftime	# victory_status_resign
一意 4 合計 39	一意 2 合計 39	一意 2 合計 39	一意 2 合計 39
resign 27 69.23%	Min 0 中央値 0 平均値 0.13 Mode 0 Max 1	Min 0 中央値 0 平均値 0.15 Mode 0 Max 1	Min 0 中央値 1 平均値 0.69 Mode 1 Max 1
outoftime 6 15.38%			
mate 5 12.82%			
その他のすべての値 1 2.56%			
resign	0	0	1
resign	0	0	1
resign	0	0	1
outoftime	0	1	0
mate	1	0	0
resign	0	0	1
resign	0	0	1
resign	0	0	1
resign	0	0	1
resign	0	0	1
resign	0	0	1
resign	0	0	1
resign	0	0	1
resign	0	0	1
mate	1	0	0
outoftime	0	1	0
mate	1	0	0
resign	0	0	1
resign	0	0	1
outoftime	0	1	0

One-Hot-Encode 列

「n」の数値列を作成します。ここで、n は選択したカテゴリ変数の一意の値の数です。

ソース列
One-Hot-Encode 列
victory_status

一意の値 (サンプル内) 3 個の値

⚠️ AWS では、パフォーマンスの低下やメモリオーバーフローを避けるため、入力データにより 200 を超える新しい列が生じる可能性がある場合は、この変換を使用しないことを推奨しています。コンソールに表示される値よりも多くの値が列にある場合、データセットが想定よりも多くの列を生成する可能性があります。

追加 3 個の列

変換を適用

- すべての行 (39 行)
変換はデータセット内のすべての行に適用されます
- フィルタリングされた行 - 適用された 1 フィルタ (38/39)
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

[フィルター条件を追加](#)

フィルター victory_status by IS

数値データの正規化

事前準備

データ変換処理の作成

ジョブの実行

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

表示 18 個の列 | 500 個の行 | ハイライトされているもののみを表示
サンプル

m_type	ソース	プレビュー	# num
	# price	# price_normalize	
合計	500	合計 500	
一意	189	一意 181	一意 179
m	270 54%		
e/apt	213 42.6%		
m	17 3.4%		
room	4000	-0.784589686216347	3 225
ome/apt	9800	0.386183301006854	3 304
room	7000	-0.179017451445726	2 227
ome/apt	4000	-0.784589686216347	30 73
room	4266	-0.730895614733352	2 73
ome/apt	7750	-0.0276243927530704	2 156
ome/apt	10000	0.426554783324896	2 267
room	20000	2.44512889922697	1 3
ome/apt	4363	-0.711315445809102	3 87
ome/apt	11514	0.732166904472469	3 16
ome/apt	15000	1.43584184127593	3 90
room	3650	-0.855239780272920	2 69
ome/apt	4500	-0.683660980421244	3 50
room	3800	-0.824961168534389	2 90
room	2600	-1.06719006244264	2 131
room	4500	-0.683660980421244	1 94
room	4067	-0.771065239639803	1 33
room	7000	-0.179017451445726	2 54
ome/apt	7143	-0.150151841588326	1 131

スケーリングまたは正規化

ソース列
正規化を実行する列を選択

price

スケーリングまたは正規化タイプの選択

- 最小-最大正規化
値を [0,1] の範囲に再スケールします。
- 指定された値の間のスケール
2 つの指定された値の範囲に値を再スケールします。
- 平均正規化または標準化
[-1, 1] の範囲内で平均 (μ) が 0、標準偏差 (σ) が 1 であるようにデータを再スケールします。
- Z スコアの正規化または標準化
平均値 (μ) が 0 で、標準偏差 (σ) が 1 であるようにデータ値を直線的にスケールします。異常値を処理するのに最適。

宛先列

price_normalize

変換を適用

- すべての行 (500 行)
変換はデータセット内のすべての行に適用されます
- フィルタリングされた行 - 適用された 1 フィルタ (472/5)
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

[フィルター条件を追加](#)

フィルタ price by LESS_THAN_EQUAL

AWS Glue DataBrew の使い方



事前準備 (IAM*)

データ変換処理の作成

ジョブの実行

- IAM ユーザー/グループ
- IAM ロール
- IAM ポリシー

- プロジェクトの作成
- データセットへの接続
- レシピの作成

- レシピジョブ
- プロファイルジョブ

* AWS Identity and Access Management

AWS Glue DataBrew には 2 種類のジョブがある

• レシピジョブ

- データセットに対してレシピを適用して変換処理を行うもの

• プロファイルジョブ

- データセットの統計に関するプロファイルを作成するもの

ジョブを実行しても対象のデータセットを書き換えることはせず、指定した Amazon S3 上に結果を書き出す

レシピジョブ

レシピジョブの作成手順①

データセットに対してレシピを適用して変換処理

事前準備

データ変換処理の作成

ジョブの実行

ジョブタイプとして
「レシピジョブを作成」を選択

ジョブを実行する対象の

- データセット
- プロジェクト
- レシピ

を選択

DataBrew > ジョブ > ジョブを作成

ジョブを作成

ジョブの詳細

ジョブ名
ジョブの ID
chess-winner-summary

ジョブ名は 1~240 文字にする必要があります。有効な文字は、英数字 (A~Z、a~z、0~9)、ハイフン (-)、ピリオド (.), およびスペースです。

ジョブタイプ
データセットで実行するジョブのタイプ

- レシピジョブを作成**
関連付けられたデータセットの母集団について、関連付けられたレシピから変換を実行します。
- プロファイルジョブを作成する**
データの形状を示す概要と統計を生成します。

ジョブ入力
ジョブの入力データセットとそれに適用されるレシピ。

で実行する

- データセット**
既存または新規の DataBrew データセットでジョブを実行します。
- プロジェクト**
ジョブが関連付けられていないプロジェクトでジョブを実行します。

データセットを選択

chess-games X データセットを参照 新しいデータセットの接続

レシピを選択 レシピバージョン

chess-project-recipe X バージョン 2.0 ▼ レシピを参照

レシピジョブの作成手順②

データセットに対してレシピを適用して変換処理

ジョブの出力結果のアウトプット
ファイルフォーマットや書き出し
先の S3 プレフィックスを指定

パーティションの設定や
ファイルの上書きオプション、
暗号化設定も指定することが可能

事前準備

データ変換処理の作成

ジョブの実行

ジョブ出力設定
ジョブを実行すると、指定したファイルの送信先に出力ファイルが生成されます。

ファイルタイプ: CSV
S3 の場所: s3://aaaa-bucket/databrew-
圧縮: None
区切り記号: カンマ (,)

別の出力を追加する

出力ファイルが大きすぎる場合、出力ファイルはパーティション分割されます。

追加設定 - オプション

列値によるカスタムパーティション
列の一意の値でパーティション化。ファイルはパーティション化され、指定された列の順序に基づいてフォルダパスに保存されます。例: 列 A と列 B でパーティション分割されたファイルは S3 パス s3://出力ファイルパス.../列 A/列 B/ に格納されます。

列名を入力

ファイル出力ストレージ

- ジョブ実行ごとに新しいフォルダを作成する
指定された S3 パスの下に、ジョブ実行ごとに、および各出力ファイルの種類ごとに新しいフォルダが作成されます。出力フォルダとファイル名には、ジョブ名とジョブの実行時間が含まれます。例: s3://bucket/myfolder/jobname_10may20_timestamp/filetype_compression/jobname_10may2020_timestamp_part1.csv
- ジョブ実行ごとに出力ファイルを置き換える
フラット出力ファイルは、指定された S3 パスの下に作成されます。ジョブ実行ごとに、以前の出力ファイルは最新のジョブ実行のファイルに置き換えられます。バケットバージョンニングを有効にして、以前のファイルバージョンを復元できます。例: s3://bucket/myfolder/jobname_part1.csv

暗号化

- ジョブ出力ファイルの暗号化を有効にする
SSE-S3 または AWS KMS を使用してジョブ出力ファイルを暗号化する

サポートされるデータセット

事前準備

データ変換処理の作成

ジョブの実行

アウトプットファイルのサポートフォーマットと拡張子

フォーマット	拡張子 (非圧縮)	拡張子 (圧縮)
CSV *1	.csv	.csv.snappy, .csv.gz, .csv.lz4, csv.bz2, .csv.deflate
Apache Parquet	非サポート	.parquet.snappy, .parquet.gz, .parquet.lz4, .parquet.lzo
AWS Glue Parquet	非サポート	.glue.parquet.snappy
Apache Avro	.avro	.avro.snappy, .avro.gz, .avro.lz4, .avro.bz2, .avro.deflate
Apache Orc	非サポート	.orc.snappy, .orc.lzo, .orc.zlib
XML	.xml	.xml.snappy, .xml.gz, .xml.lz4, .xml.bz2, .xml.deflate
JSON (JSON Lines format only)	.json	.json.snappy, .json.gz, .json.lz4, json.bz2, .json.deflate

*1 区切り文字は Comma (,), Colon (:), Semi-colon (;), Pipe (|), Tab (¥t), Caret (^), Space に対応

レシピジョブの作成手順②

データセットに対してレシピを適用して変換処理

ジョブに割り当てるノード数やタイムアウト、リトライ回数を指定し、パフォーマンスを調整することが可能

- ノード数はデフォルト 5, 最大 149
- 1 ノード 4 vCPUs, 16GB メモリ

データセットに対するアクセス許可を指定

- 新しい IAM ロールを作成
- 既存の IAM ロールを選択

プロジェクト作成時と同じものを選択

事前準備

データ変換処理の作成

ジョブの実行

The screenshot shows the AWS Glue console interface for configuring a job. The left sidebar contains navigation icons for Data Sets, Projects, Recipes, Jobs, and Latest Features. The main content area is titled 'Advanced Job Settings - Options' and includes the following sections:

- ▼ アドバンスドジョブ設定 - オプション**
プロジェクトで実行されるジョブに使用される処理とコンピューティングを制御する設定
- ユニットの最大数**
ジョブの実行時に割り当てることができる DataBrew ノードの最大数を設定します。
Input: 5
- ジョブのタイムアウト (分)**
ジョブがタイムアウトしたときの設定
Input: 2880
- 再試行回数**
失敗時にジョブを再試行する最大回数
Input: 0
- CloudWatch Logs**
 ジョブの Amazon CloudWatch ログを有効にする
このジョブの実行時に Amazon CloudWatch ログの作成を有効にします。 [詳細はこちら](#)
- ▶ 関連付けられたスケジュール - オプション**
最大 2 つのスケジュールを関連付けることで、ジョブを自動化できます。
- ▶ タグ - オプション**
定義して AWS リソースに割り当てることができるメタデータ。各タグは、お客様が定義したキー（名前）とオプションの値で構成されるシンプルなラベルです。タグを使用すると、目的、所有者、環境、またはその他の条件によるリソースの管理、検索、フィルタリングが簡単になります。
- アクセス許可**
DataBrew needs permission to connect to data on your behalf. Use an IAM role with the [必須ポリシー](#) attached.
- ロール名**
データに接続するためのアクセス権を持つロールを選択します。最新の更新を表示するには更新します。
Dropdown: AwsGlueDataBrewDataAccessRole

At the bottom right, there are three buttons: 'キャンセル' (Cancel), 'ジョブを作成' (Create Job), and 'ジョブを作成し実行する' (Create and Run Job).

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

事前準備

データ変換処理の作成

ジョブの実行

The screenshot shows the AWS DataBrew console interface. At the top, there are three navigation arrows: '事前準備' (Preparation), 'データ変換処理の作成' (Creation of data transformation), and 'ジョブの実行' (Job execution). The main content area displays the 'chess-winner-summary' job details. It shows the data set 'chess-games' (4.6 MB) and the recipe 'chess-project-recipe'. A data lineage diagram illustrates the flow: S3 (chess-games.xlsx) -> データセット (chess-games) -> ジョブ (chess-winner-summary) -> S3 (output CSV). A red arrow points from the 'レシピ' (Recipe) icon to a detailed view of the 'chess-project-recipe'.

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	バージョンの説明
3時間前/ojunpei-lsengard	Second version of the recipe.
2021年2月16日, 6:24:48 午後	

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 overtime と time ran out in victory_status



プロファイルジョブ

プロファイルジョブの作成手順①

データセットの統計に関するプロファイルを作成

事前準備

データ変換処理の作成

ジョブの実行

ジョブタイプとして
「プロファイルジョブを作成する」
を選択

ジョブを実行する対象の
データセットを選択

DataBrew > ジョブ > ジョブを作成

ジョブを作成

ジョブの詳細

ジョブ名
ジョブの ID

chess-data-profile

ジョブ名は 1~240 文字にする必要があります。有効な文字は、英数字 (A~Z、a~z、0~9)、ハイフン (-)、ピリオド (.)、およびスペースです。

ジョブタイプ
データセットで実行するジョブのタイプ

- レシピジョブを作成
関連付けられたデータセットの母集団について、関連付けられたレシピから変換を実行します。
- プロファイルジョブを作成する**
データの形状を示す概要と統計を生成します。

ジョブ入力
ジョブの入力データセット。

データセットを選択

chess-games

データセットを参照

新しいデータセットの接続

プロファイルジョブの作成手順②

データセットの統計に関するプロファイルを作成

事前準備

データ変換処理の作成

ジョブの実行

データセットの
サンプリング件数を指定

- 全件
- 件数指定

ジョブの出力結果の
アウトプットファイルの
書き出し先の S3 プレフィックス
を指定

The screenshot shows the AWS Databrew console interface for configuring a job. The left sidebar contains navigation options: データセット, プロジェクト, レシビ, ジョブ, and 最新機能. The main content area is titled 'Job run sample' and includes the following sections:

- Job run sample**: A job can be run on the entire dataset or a custom sample of the dataset.
- Data sample**: Define the scope of the dataset to run the job on.
 - Full dataset
 - Custom sample
- Custom sample configuration**: A text input field contains '20000' with the label 'rows' to its right. Below the field, it states 'Value must be greater than zero'.
- ジョブ出力設定**: ジョブを実行すると、指定したファイルの送信先に出力ファイルが生成されます。
 - ファイルタイプ**: S3 の場所
 - 出力形式**: 形式は s3://bucket/folder/ です
 - JSON**: A text input field contains 's3://aaaa-bucket/databrew-output/' with a '参照' button to its right.
 - 暗号化**: ジョブ出力ファイルの暗号化を有効にする
SSE-S3 または AWS KMS を使用してジョブ出力ファイルを暗号化する

プロファイルジョブの作成手順③

データセットの統計に関するプロファイルを作成

ジョブに割り当てるノード数やタイムアウト、リトライ回数を指定し、パフォーマンスを調整することが可能

- ノード数はデフォルト 5, 最大 149
- 1 ノード 4 vCPUs, 16GB メモリ

データセットに対するアクセス許可を指定

- 新しい IAM ロールを作成
- 既存の IAM ロールを選択

プロジェクト作成時と同じものを選択

事前準備

データ変換処理の作成

ジョブの実行

The screenshot shows the AWS DataBrew console interface for configuring a job. The left sidebar has a 'ジョブ' (Jobs) tab selected. The main content area is titled 'アドバンスドジョブ設定 - オプション' (Advanced Job Settings - Options) and includes several sections:

- ▼ アドバンスドジョブ設定 - オプション**
プロジェクトで実行されるジョブに使用される処理とコンピューティングを制御する設定
- ユニットの最大数**
ジョブの実行時に割り当てることができる DataBrew ノードの最大数を設定します。
Input: 5
- ジョブのタイムアウト (分)**
ジョブがタイムアウトしたときの設定
Input: 2880
- 再試行回数**
失敗時にジョブを再試行する最大回数
Input: 0
- CloudWatch Logs**
 ジョブの Amazon CloudWatch ログを有効にする
このジョブの実行時に Amazon CloudWatch ログの作成を有効にします。 [詳細はこちら](#)
- ▶ 関連付けられたスケジュール - オプション**
最大 2 つのスケジュールを関連付けることで、ジョブを自動化できます。
- ▶ タグ - オプション**
定義して AWS リソースに割り当てることができるメタデータ。各タグは、お客様が定義したキー（名前）とオプションの値で構成されるシンプルなラベルです。タグを使用すると、目的、所有者、環境、またはその他の条件によるリソースの管理、検索、フィルタリングが簡単になります。
- アクセス許可**
DataBrew needs permission to connect to data on your behalf. Use an IAM role with the [必須ポリシー](#) attached.
- ロール名**
データに接続するためのアクセス権を持つロールを選択します。最新の更新を表示するには更新します。
Dropdown: AwsGlueDataBrewDataAccessRole

At the bottom right, there are three buttons: 'キャンセル' (Cancel), 'ジョブを作成' (Create Job), and 'ジョブを作成し実行する' (Create and Run Job).

データプロファイルの生成

事前準備

データ変換処理の作成

ジョブの実行

DataBrew > データセット > chess-games

chess-games S3 chess-games.xlsx 4.6 MB

プロファイルを再実行 このデータセットを使用してプロジェクトを作成する アクション

ジョブの詳細

表示するプロファイルを選択 ジョブの実行 1 | 2021年2月16日, 10:28:50 午後

前回のジョブ実行 成功 5分前前、スケジュールされたジョブの実行はありません
Data profile was run on custom sample of first 20,000 rows of your dataset

表示するプロファイルを選択 ジョブの実行 1 | 2021年2月16日, 10:28:50 午後

列 (17)

検索

すべて (17) ABC 文字列 (9)

ABC id

rated

created_at

creation_date

last_move_at

turns

ABC victory_status

ABC winner

ABC increment_code

ABC white_id

white_rating

ABC black_id

black_rating

ABC moves

ABC opening_eco

ABC opening_name

opening_ply

ABC 文字列 id

データ品質

有効な値 20000 100% 欠落した値 0 0%

データインサイト

濃度 高 95% の行は一意です 19070

欠落 欠落した値がありません 0

値の分散

一意の値 文字列の長さ

一意 19,070 合計 20,000

上位の一意の値

検索

XRuQPSzH	5	<1%
dFQ5D7CS	4	<1%
OgTDO6Av	4	<1%
Q0jogkvi	4	<1%
h0YsGMhj	4	<1%
edYOVb5F	4	<1%
j5KY62y5	4	<1%
dJEtAQp7	4	<1%
t7vvcwqO	4	<1%
UvqLtnPM	4	<1%
その他	19,959	99%

50 個の一意の値をすべて表示する

相関関係

相関係数 (r) は、2 つの変数がどの程度密接に関連しているかを定義します。範囲は -1.0 から +1.0 です。ここで、0 は変数間に関係がないことを意味します。

	created_at	last_move_at	turns	white_rating	black_rating	opening_ply
created_at	0.5	0.5	0.5	0.5	0.5	0.5
last_move_at	0.5	0.5	0.5	0.5	0.5	0.5
turns	0.5	0.5	0.5	0.5	0.5	0.5
white_rating	0.5	0.5	0.5	0.5	0.5	0.5
black_rating	0.5	0.5	0.5	0.5	0.5	0.5
opening_ply	0.5	0.5	0.5	0.5	0.5	0.5

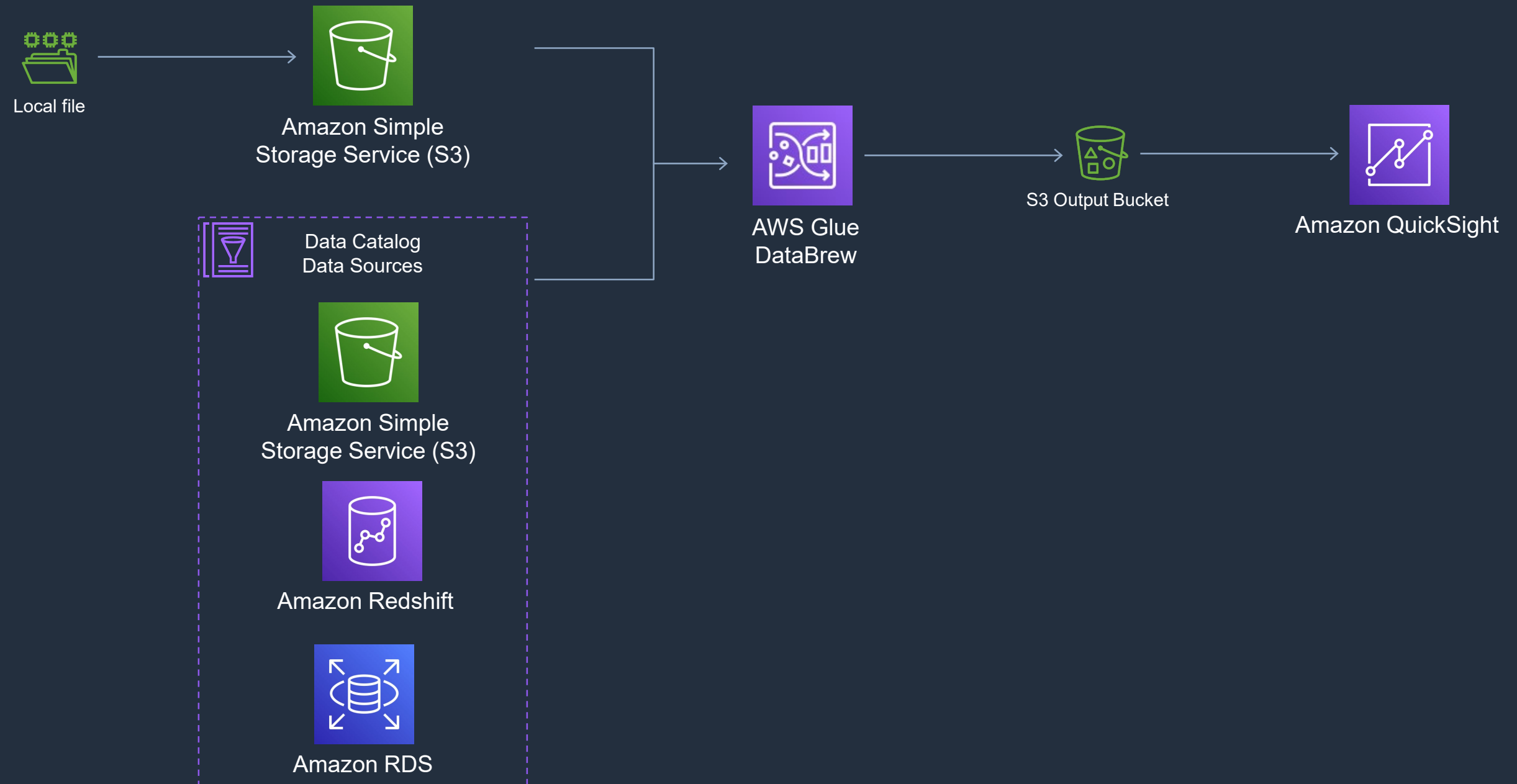
AWS Glue DataBrew のユースケース



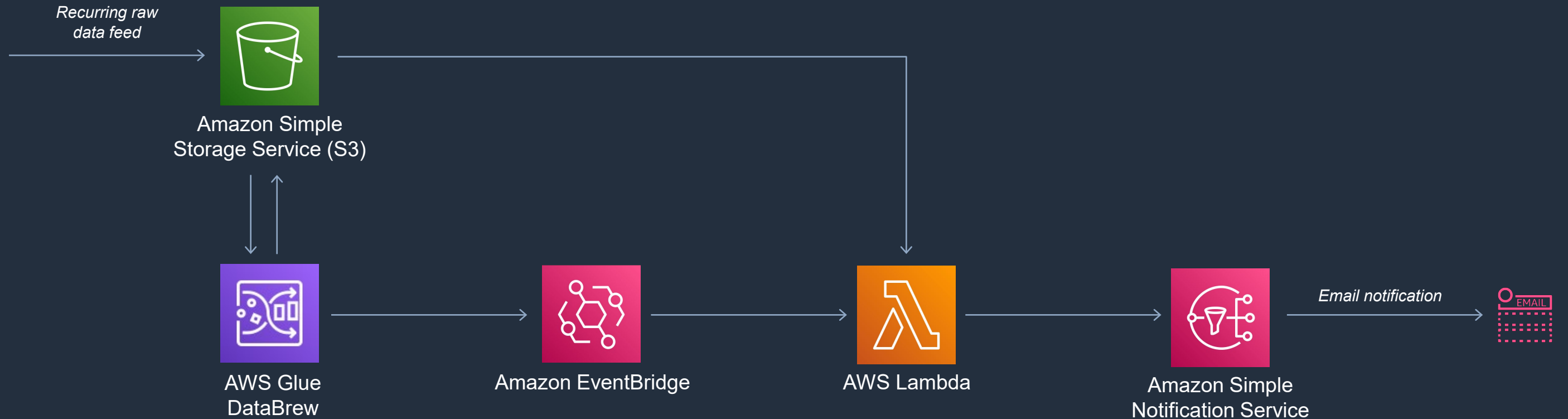
AWS Glue DataBrew のユースケース

1. Amazon Redshift や Amazon RDS など他の AWS サービス内のデータをアドホックに探索して整備し、BI レポート環境を構築する
2. 定常的に生成するデータのプロファイルチェックを自動化し通知する
3. 機械学習モデルを構築するためのデータを準備する
4. Amazon Athena にクエリして Amazon QuickSight で可視化するためのデータを準備するパイプラインをコーディングレスで構築する

1. BI レポートティングのためのアドホックデータ分析

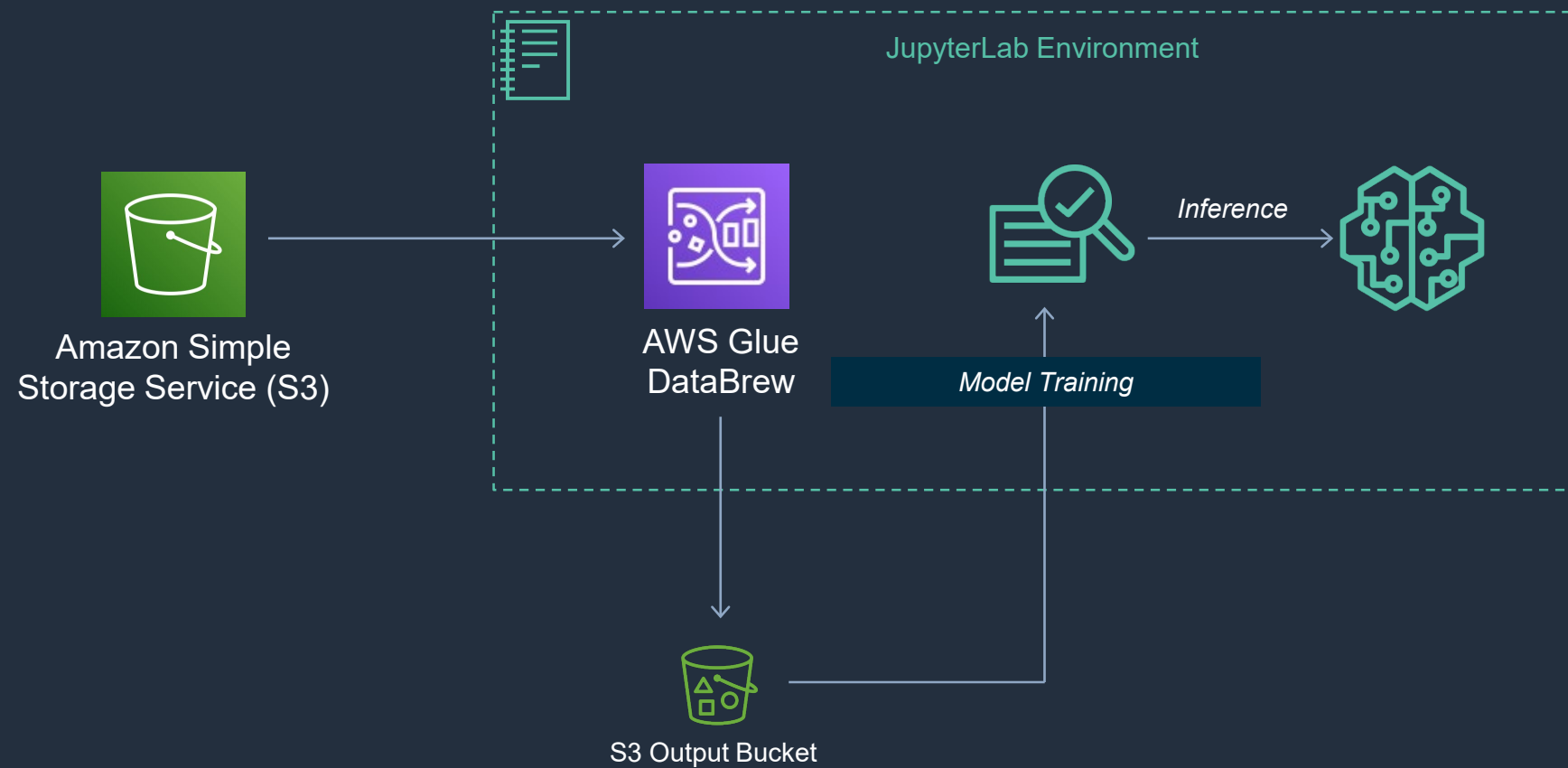


2. AWS Lambda でデータ品質ルールを設定



<https://aws.amazon.com/jp/blogs/big-data/setting-up-automated-data-quality-workflows-and-alerts-using-aws-glue-databrew-and-aws-lambda/>

3. 機械学習のためのデータ前処理

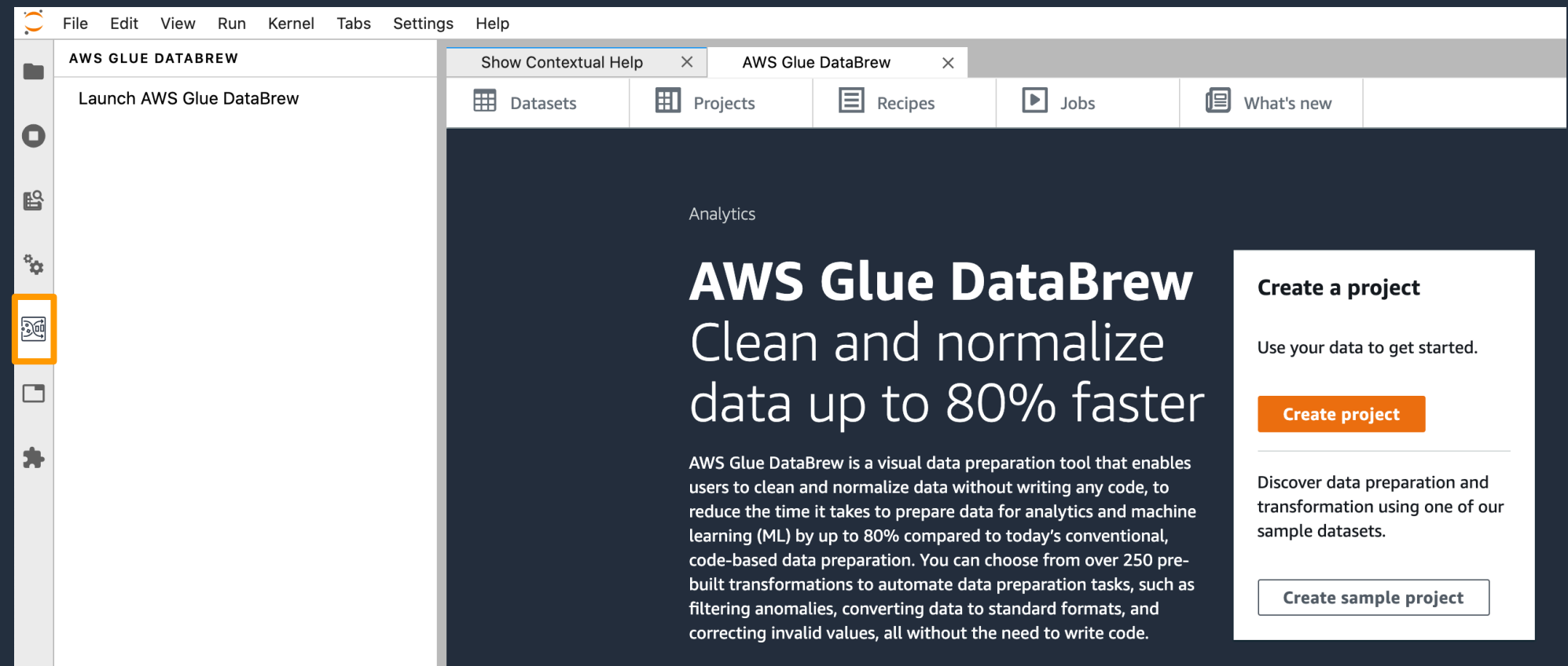


JupyterLab Extension

JupyterLab 経由で AWS Glue DataBrew に接続可能

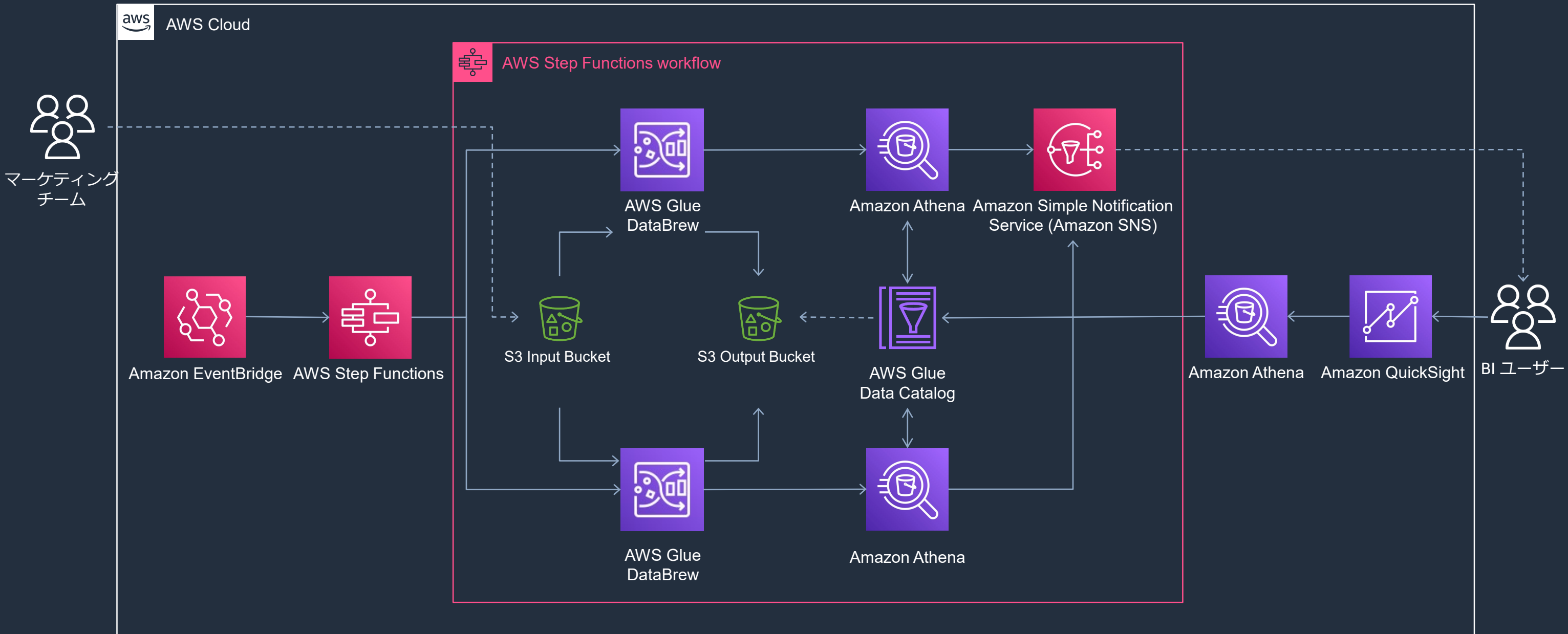
- AWS CLI, AWS Jupyter proxy をセットアップ
- JupyterLab (v.2.2.6 以降) に aws_glue_databrew_jupyter Extension をインストール
- JupyterLab から AWS Glue DataBrew の接続経路はパブリック通信となる

Extension インストールで
JupyterLab から
AWS Glue DataBrew に
直接接続できるようになる



The screenshot shows the JupyterLab interface with the AWS Glue DataBrew extension installed. The left sidebar contains a menu with a highlighted icon for the extension. The main content area displays the AWS Glue DataBrew landing page, which includes a navigation bar with 'Datasets', 'Projects', 'Recipes', 'Jobs', and 'What's new'. The main heading reads 'AWS Glue DataBrew Clean and normalize data up to 80% faster'. Below this, there is a 'Create a project' section with a 'Create project' button and a 'Create sample project' button. The text describes the tool's capabilities: 'AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code, to reduce the time it takes to prepare data for analytics and machine learning (ML) by up to 80% compared to today's conventional, code-based data preparation. You can choose from over 250 pre-built transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values, all without the need to write code.'

4. ワークフロー内でデータ準備をオーケストレーション

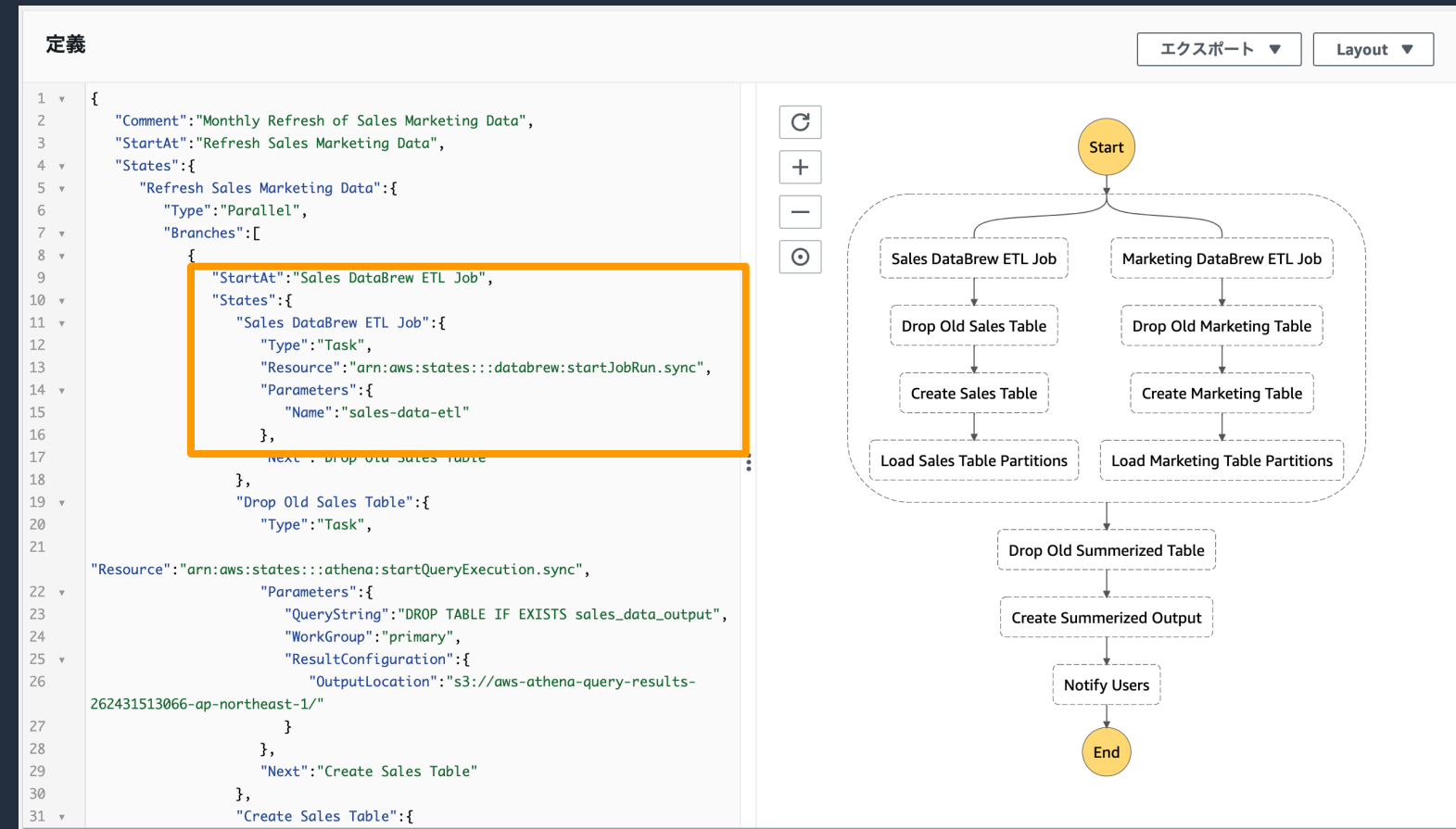


<https://aws.amazon.com/jp/blogs/big-data/orchestrating-an-aws-glue-databrew-job-and-amazon-athena-query-with-aws-step-functions/>

AWS Step Functions との連携

AWS Glue DataBrew ジョブを
AWS Step Functions の
ワークフローに統合可能

データのクリーニングや正規化の
ステップを、分析や機械学習の
ワークフローの一部として
オーケストレーション可能に





ビジネスアナリスト
データサイエンティスト

AWS Glue DataBrew

リッチなビジュアルインターフェース
によりデータを整形・正規化

250 以上の組み込みの変換機能
を選択し、タスクを自動化

データパターンや異常値を把握するための
データプロファイル機能

大規模なデータセットを操作可能



ETL デベロッパー

AWS Glue Studio

コードを記述せずに ETL ジョブを
視覚的にオーサリング

コンソールから数千のジョブを監視

学習コストなしに分散処理を活用

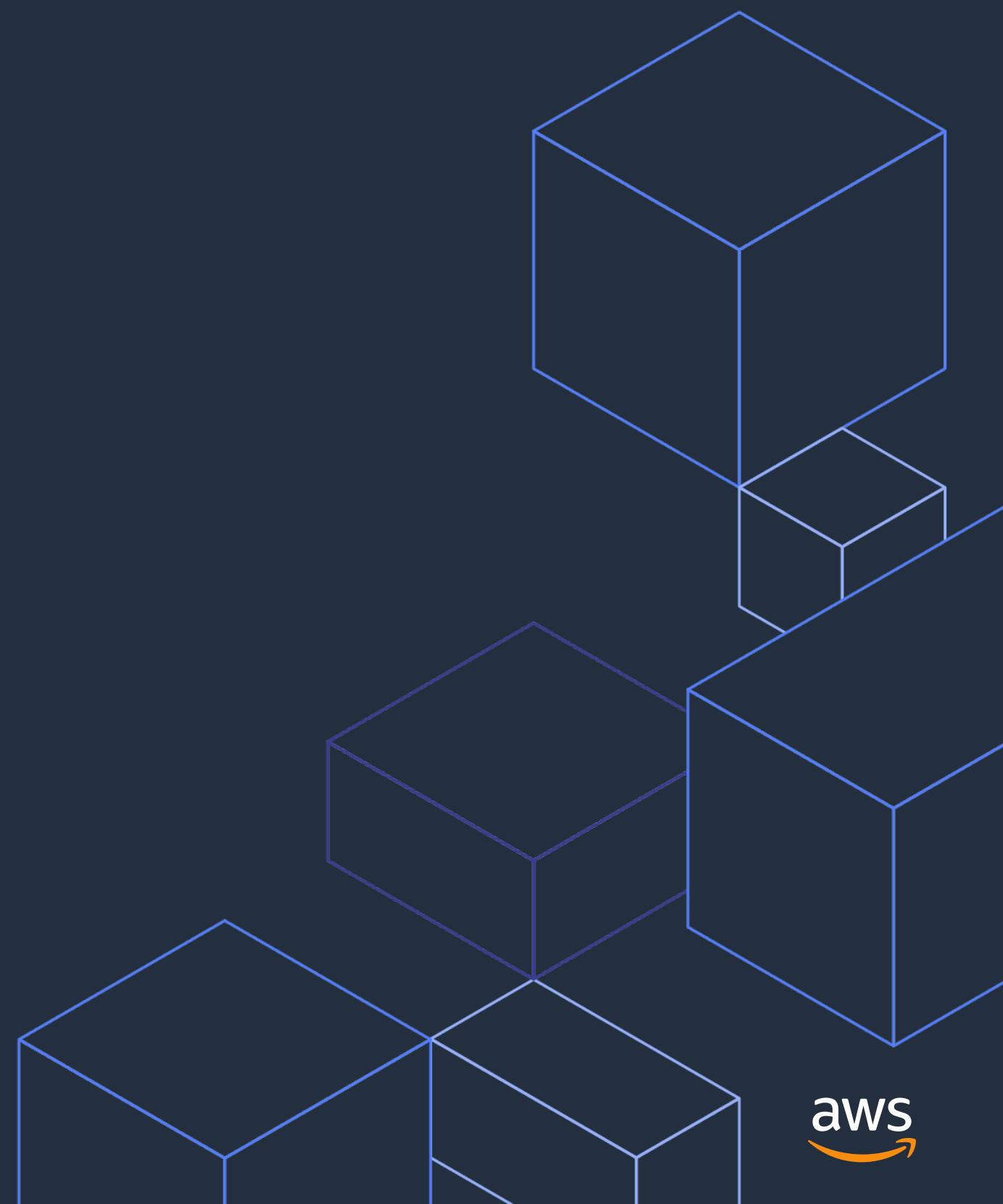
再利用可能なコードを使った高度な変換

AWS Glue DataBrew の料金

AWS Glue DataBrew の料金

- DataBrew ジョブ - **\$0.48/node/hour**
 - ジョブの実行に使用された AWS Glue DataBrew ノードの数に基づいて 1 時間ごとの料金が発生
 - デフォルトでは各ジョブに 5 ノード 割り当てられる
 - 1 ノード 4 vCPUs, 16GB メモリ
- DataBrew インタラクティブセッション - **\$1/30分**
 - DataBrew プロジェクトを開くとセッションが開始され、未操作の時間が続いた場合自動的にサスペンド
 - はじめて DataBrew にアクセスする場合、最初の 40 セッションは無償

まとめ



まとめ

- AWS Glue DataBrew は、データのクリーンアップおよび正規化を最大 80% 高速化するビジュアルデータ準備ツール
- データアナリストやサイエンティストがコーディングを行うことなしに、250 種類以上の組み込みの変換処理を使ってデータを分析に必要な形に簡単に整形することが可能
- アドホックなデータ探索、データの品質チェック、機械学習モデル構築の前処理、データ分析パイプライン構築などさまざまなユースケースに活用することが可能

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
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>

