aws

でのサーバーレス ETL の開始方法 AWS Glue

# AWS 規範ガイダンス

# AWS 規範ガイダンス: でのサーバーレス ETL の開始方法 AWS Glue

# Table of Contents

# AWS Glue でのサーバーレス ETL 入門

Dheer Toprani と Adnan Alvee、Amazon Web Services (AWS)

2024 年 3 月 (ドキュメント履歴)

Amazon Web Services (AWS) クラウド上で、AWS Glue は完全に管理されたサーバーレス環境であり、スケールの大きなデータの抽出、変換、ロード (ETL) を行うことができます。AWS Glue を使えば、データを分類し、クリーニングし、リッチ化し、さまざまなデータストアやデータストリームに費用対効果の高い方法で確実に移動させることができます。

AWS Glue はサーバーレスなので、サーバーのプロビジョニングや管理を心配する必要はありません。AWS Glue では、使用したリソースに対してのみ支払いが発生し、必要に応じて規模を拡大したり縮小したりできます。

AWS Glue のコンポーネントは次のとおりです。

- AWS Glue ETL – AWS Glue ETL は、あるソースから別のソースへデータを抽出、変換、ロードするためのバッチおよびストリーミングオプションを提供します。
- AWS Glue Data Catalog – Data Catalog は、すべてのデータ資産のメタデータを整理するための中央リポジトリです。Data Catalog は、データ分析サービス全体でデータ資産を検索、発見、共有できる統合インターフェイスを提供します。
- AWS Glue DataBrew – DataBrew は、データを視覚的に探索、クリーニング、変換できるコーディング不要のデータ準備ツールです。250 種類以上のあらかじめ用意された変換の中から選択して、コードを記述せずにデータ準備タスクを自動化できます。

このガイドでは、AWS Glue がどのように機能するのか、どのように使い始めることができるのかなど、 を大まかに紹介します。自動化、モニタリング、他の AWS サービスとの統合など、AWS Glue ジョブをオーサリングする前に知っておくべき重要なコンセプトをカバーしています。次のステップセクションでは、AWS Glue でコードをすばやく記述できるようになります。すでに AWS Glue の使用経験がある場合は、ベストプラクティスのセクションが知識のギャップを埋めるのに役立ちます。このガイドを読み終える頃には、AWS Glue を効果的に使い始めるために必要な知識とリソースを身につけることができます。

# AWS Glue ETL

AWS Glue ETL は、さまざまなソースからデータを抽出し、ビジネスニーズに合わせてデータを変換し、選択した宛先にロードすることをサポートしています。このサービスは Apache Spark エンジンを使用してビッグデータのワークロードをワーカーノード全体に分散し、インメモリ処理によるより高速な変換を可能にします。

AWS Glue は、Amazon Simple Storage Service (Amazon S3)、Amazon DynamoDB、Amazon Relational Database Service (Amazon RDS) などさまざまなデータソースをサポートしています。サポートされているデータソースの詳細については、AWS Glue の「ETL の接続タイプとオプション」を参照してください。

## AWS Glue でのオーサリング

AWS Glue には経験やユースケースに応じて、ETL ジョブをオーサリングする方法が複数用意されています。

- Python シェルジョブは、Python で記述された基本的な ETL スクリプトを実行するために設計されています。これらのジョブは 1 台のマシンで実行されるため、小規模または中規模のデータセットに適しています。
- Apache Spark ジョブは、 Python または Scala のいずれかで作成できます。これらのジョブは Spark を使用し多数のワーカーノードにわたってワークロードを水平方向にスケールするため、大規模なデータセットや複雑な変換を処理できます。
- AWS Glue ストリーミング ETL は、Apache Spark の構造化ストリーミングエンジンを使用して、Exactly-once を使用してマイクロバッチジョブ内のストリーミングデータを変換します。AWS Glue ストリーミングジョブは Python または Scala で作成できます。
- AWS Glue Studio は、Apache Spark プログラミングに不慣れな開発者が Spark ベースの ETL にアクセスできるようにするための視覚的なボックスと矢印スタイルのインターフェースです。

## データ処理単位

AWS Glue はデータ処理ユニット (DPU) を使用して ETL ジョブに割り当てられたコンピュートリソースを測定し、コストを計算します。1 つの DPU は 4 基の vCPUs と 16 GB のメモリに相当します。DPU は、その複雑さとデータ量に応じて AWS Glue ジョブに割り当てる必要があります。DPU を適切な量割り当てることで、パフォーマンス要件とコスト制約とのバランスを取ることができます。

AWS Glue には、さまざまなワークロードに最適化されたいくつものワーカータイプが用意されています。

- G.1X または G.2X (ほとんどのデータ変換、結合、クエリ用)
- G.4X または G.8X (より要求の厳しいデータ変換、集約、結合、クエリ用)
- G.025X (少量で散発的なデータストリーム用)
- Standard (AWS Glue バージョン 1.0 以前用、それ以降のバージョンの AWS Glue では非推奨)

## Python シェルを使用する

Python シェルジョブでは、1 DPU を使用して 16 GB のメモリを使用するか、0.0625 DPU を使用して 1 GB のメモリを使用できます。Python シェルは、小規模または中規模のデータセット (約 10 GB まで) を使用する基本的な ETL ジョブを対象としています。

## ワーカータイプの比較

次の表は、Apache Spark 環境を使用するバッチ、ストリーミング、および AWS Glue Studio ETL ワークロードのさまざまな AWS Glue ワーカータイプを示しています。

| | G.1X | G.2X | G.4X | G.8X | G.025X | 規格 |
|---|---|---|---|---|---|---|
| vCPU | 4 | 8 | 16 | 32 | 2 | 4 |
| メモリ | 16 GB | 32 GB | 64 GB | 128 GB | 4 GB | 16 GB |
| ディスク容量 | 64 GB | 128 GB | 256 GB | 512 GB | 64 GB | 50 GB |
| 1 ワーカーあたりのエグゼキューター | 1 | 1 | 1 | 1 | 1 | 2 |
| DPU | 1 | 2 | 4 | 8 | 0.25 | 1 |

AWS Glue バージョン 2.0 以降では、Standard ワーカータイプは推奨されていません。G.025X ワーカータイプは、AWS Glue バージョン 3.0 以降を使用したストリーミングジョブでのみ利用可能です。

# AWS Glue Data Catalog

「AWS Glue Data Catalog」は、さまざまなデータソースのすべてのデータ資産を一元管理するメタデータリポジトリです。データフォーマット、スキーマ、ソースに関する情報を保存してクエリするための統合インターフェースを提供します。AWS Glue ETLジョブが実行されると、このカタログを使用してデータに関する情報を理解し、正しく変換されるようにします。

AWS Glue Data Catalog は次のコンポーネントで構成されています。

- データベースとテーブル
- クローラーおよび分類子
- Connections
- スキーマレジストリ

# AWS Glue データベースとテーブル

AWS Glue Data Catalog はデータベースとテーブルで構成され、メタデータを保存および管理するための論理構造を提供しています。この構造により、「AWS Identity and Access Management (IAM) ポリシー」を使用して、テーブルレベルまたはデータベースレベルでの正確なデータアクセス制御が可能になります。

AWS Glue データベースには多数のテーブルを含めることができ、各テーブルは 1 つのデータベースに関連付ける必要があります。これらのテーブルには実際のデータへの参照が含まれており、AWS Glue サポートされているさまざまなデータソースのいずれかに保存できます。AWS Glue テーブルには、列名、データ型、パーティションキーなどの重要なメタデータも格納されます。

AWS Glue にテーブルを作成する方法はいくつかあります。

- AWS Glue クローラー
- AWS Glue ETL ジョブ
- AWS Glue コンソール
- 「AWS Glue API」での `CreateTable` 操作
- AWS CloudFormation テンプレート
- AWS Cloud Development Kit (AWS CDK)
- 移行された Apache Hive メタストア

# AWS Glue クローラおよび分類子

AWS Glue クローラーはデータストアからメタデータを自動的に検出して抽出し、それに応じて AWS Glue Data Catalog データを更新します。クローラーがデータストアに接続して、データのスキーマを推測します。次に、検出したスキーマ情報を使用してデータカタログ内のテーブルを作成または更新します。クローラーは、ファイルベース、およびテーブルベースのデータストアの両方をクロールできます。サポートされているデータストアの詳細については、「クロール可能なデータストア」を参照してください。

クローラーは「分類器」を使用してデータの形式を正確に認識し、処理方法を決定します。デフォルトでは、クローラーは AWS Glue が提供する一般的な「組み込み分類子」のセットを使用しますが、特定のユースケースを処理するカスタム分類子を作成することもできます。

# AWS Glue 接続

AWS Glue「接続」を使用して、AWS Glue がさまざまなデータソースに接続できるようにする接続パラメータを定義できます。接続を追加すると、これらのソースへの接続に必要な構成が一元化され、簡素化されます。

「接続を定義する」ときは、接続タイプ、接続エンドポイント、および必要な認証情報を指定します。接続を定義すると、複数の AWS Glue ジョブやクローラーで再利用できます。AWS Glue による接続を使用することで、ログイン認証情報や仮想プライベートクラウド (VPC) IDなど、同じ接続情報を繰り返し入力する必要性を減らすことができます。

# AWS Glue「 スキーマレジストリ」

「AWS Glue スキーマレジストリ」は、データストリームスキーマを一元的に管理および実施するための場所です。これにより、データプロデューサーと非シリアル化用の異なるシステムで、シリアル化と非シリアル化用のスキーマを共有できます。スキーマを共有することで、これらのシステムは効果的にコミュニケーションをとり、変換中のエラーを回避することができます。

スキーマ・レジストリは、下流のデータ・コンシューマーが上流で行われた変更を確実に処理できるようにします。スキーマの進化をサポートしているため、以前のバージョンのスキーマとの互換性を維持したまま、スキーマを時間の経過とともに変更することができます。

スキーマレジストリは、Amazon Kinesis Data Streams、Firehose、Apache Kafka 用 Amazon マネージドストリーミングなど、多くの AWS サービスと統合されています。使用例と統合については、「AWS Glue スキーマレジストリとの統合」を参照のこと。

# 重要な特徴とコンセプト

## ログ記録とモニタリング

AWS Glue には、ログ記録とモニタリングのオプションがいくつかあります。デフォルトでは、 は Amazon CloudWatch のロググループに`aws-glue`ログ AWS Glue を送信します。これらのログには、開始時刻や終了時刻、構成設定、発生した可能性のあるエラーや警告などの情報が含まれます。

さらに、 AWS Glue Spark ETL ジョブには以下のオプションがあり、高度なモニタリングを有効にする必要があります。

- ジョブメトリクスは、ジョブ固有のメトリクスを 30 秒ごとに CloudWatch AWS Glue の名前空間に報告します。処理されたレコード、入出力データの合計サイズ、ランタイムなど、これらのジョブ固有のメトリクスは、ジョブのパフォーマンスに関する洞察を提供します。ボトルネックや構成を最適化する機会を特定するのに役立ちます。
- 継続ロギングは、リアルタイムの Apache Spark ジョブのログを CloudWatch の `/aws-glue/jobs/logs-v2` ロググループにストリーミングします。リアルタイムログを使用すると、実行中に AWS Glue ジョブを動的にモニタリングできます。
- Spark UI には、各ステージのイベントタイムライン、有向非循環グラフ、ジョブ環境変数など、Spark ジョブに関する情報を表示するための Spark 履歴サーバーの Web インターフェイスが用意されています。永続的な Spark UI イベントログは Amazon S3 に保存され、リアルタイムで使用することも、ジョブの完了後に使用することもできます。
- Job Run Insightsは、一般的な Spark の例外を監視し、根本原因分析を行い、問題を解決するための推奨アクションを提供することで、ジョブのデバッグと最適化を簡素化します。インサイトは CloudWatch に保存されます。

## オートメーション

AWS Glue には、ETL ジョブを自動化する 2 つの主な方法として、トリガーとワークフローがあります。

AWS Glue トリガー

起動すると、 AWS Glue トリガーは指定されたジョブとクローラを開始します。トリガーは、オンデマンドで、定義済みのスケジュールに基づいて、または特定のイベントに基づいて起動することが

できます。トリガーを使って、依存するジョブとクローラーの連鎖をデザインすることができます。詳細については、AWS Glue トリガーを参照してください。

AWS Glue ワークフロー

より複雑なワークロードの場合、 AWS Glue ワークフローを使用して有向非巡回グラフを作成し、個別の AWS Glue エンティティ (トリガー、クローラ、ジョブ) 間に依存関係を構築できます。ワークフローには、パラメータの共有、進行状況のモニタリング、関連するエンティティ間の問題のトラブルシューティングを行うことができる統合インターフェイスもあります。

AWS Glue ワークフロー内で多くの関連エンティティを設定すると、ますます複雑になる可能性があります。開発者は、データサイエンティストやビジネスアナリストと複雑なデータパイプラインを共有するためのAWS Glue ブループリントを作成することができます。これらのテンプレートを使用すると、 AWS Glue ワークフローを一貫して繰り返し作成し、技術的な詳細を抽象化できます。

AWS Glue ブループリントとワークフローの詳細については、「 でブループリントとワークフローを使用して複雑な ETL アクティビティを実行する AWS Glue」を参照してください。

他の AWS サービスと AWS Glue のジョブのオーケストレーション

その他の自動化オプションについては、、、 AWS Lambda AWS Step Functions Amazon Managed Workflows for Apache Airflow (Amazon MWAA) などの他の AWS サービスと AWS Glue 統合します。

ETL ジョブのオーケストレーション方法の詳細については、 AWS Glue 「」の「オーケストレーション AWS Glue」を参照してください。

## ジョブのブックマーク

のジョブブックマーク AWS Glue は、ETL ジョブの進行状況を追跡するために使用されます。これにより、後続のジョブ実行でデータを再処理する必要がなくなります。ジョブのブックマークを有効にすると、 は処理済みのデータの記録 AWS Glue を保持します。その後、実行するたびに、データソースの新しいデータのみが処理されます。詳細については、「ジョブブックマークを使用して処理されたデータの追跡」を参照してください。

# AWS Glue DataBrew

AWS Glue DataBrew は、データのクリーニング、正規化、変換を行うフルマネージド型のビジュアルデータ準備サービスです。それを使用するたの書き込みコードがないという点で AWS Glue ETL とは異なっています。DataBrew には 250 種類以上の変換が組み込まれており、ポイントアンドクリックによる視覚的なインターフェイスにより、データ変換ジョブを作成および管理できます。

DataBrew は、別のコンソールビューで AWS Glue から使用できます。いくつかの AWS サービスとネイティブに統合されており、さまざまなファイル形式をサポートしています。詳細については、「製品とサービスの統合」を参照してください。

DataBrew は次の 6 つのコアコンセプトに基づいています:

- プロジェクト – DataBrew のデータ準備ワークスペース全体
- データセット – 構造化データまたは半構造化データのコレクション
- レシピ – データ変換ステップのセット。各ステップには多数のアクションを含めることができます
- ジョブ – レシピまたはデータプロファイルジョブを実行するための一連の命令
- データリネージュ – ビジュアルインターフェイスでデータを追跡し、その出所を特定すること
- データプロファイル – データ形式の概要表示

AWS Glue DataBrew は AWS Glue Studio と統合されているため、AWS Glue ETL ジョブとワークフロー内で DataBrew レシピを調整できます。DataBrew レシピは、ジョブのブックマーク、自動再試行、自動スケーリングなどの AWS Glue 機能を活用することもできます。DataBrew を使用開始するには、AWS Glue DataBrew サンプルプロジェクトチュートリアルを使用します。

# ベストプラクティス

AWS Glue を使用して開発するときは、次のベストプラクティスを考慮します。

## まずローカルで開発します。

ETL ジョブの構築にかかるコストと時間を節約するには、まずコードとビジネスロジックをローカルでテストします。シェルと統合開発環境 (IDE) の両方で AWS Glue ETL ジョブをテストするのに役立つ Docker コンテナのセットアップ手順については、ブログポスト「Docker コンテナを使って AWS Glue ジョブをローカルで開発・テストする」を参照してください。

## AWS Glue インタラクティブセッションを利用する

AWS Glue インタラクティブセッションは、PyCharm、IntelliJ、VS Code などのノートブックおよび IDE と統合するオープンソースの Jupyter カーネルと統合できるサーバーレスの Spark バックエンドを提供します。インタラクティブセッションを使用すると、AWS Glue Spark バックエンドと任意の IDE を使用して、実際のデータセットでコードをテストできます。開始するには、「AWS Glue インタラクティブセッション入門」の手順に従ってください。

## パーティショニングを使用して、必要なものを正確にクエリします。

パーティショニングとは、大きなデータセットを特定の列やキーに基づいて小さなパーティションに分割することです。データがパーティショニングされている場合、AWS Glue はデータセット全体をスキャンするのではなく、特定のパーティショニング基準を満たすデータのサブセットに対して選択的スキャンを実行できます。これにより、特に大きなデータセットを扱う場合に、クエリ処理がより速く、より効率的になります。

そのデータに対して実行されるクエリに基づいてデータを分割します。たとえば、ほとんどのクエリが特定の列でフィルタリングされている場合、その列を分割することでクエリ時間を大幅に短縮できます。データのパーティショニングについて詳しくは、「AWS Glue での分割データの処理」を参照してください。

# メモリ管理を最適化する

AWS Glue ETL ジョブはインメモリ処理向けに最適化された Apache Spark エンジン上で実行されるため、メモリ管理は極めて重要です。ブログ記事「AWS Glue におけるメモリ管理の最適化」では、以下のメモリ管理手法について詳しく説明しています。

- AWS Glue の Amazon S3 リスト実装
- グループ化
- 無関係な Amazon S3 パスとストレージクラスを除外する
- スパークと AWS Glue リードパーティショニング
- バルクインサート
- 最適化に取り組む
- PySpark ユーザー定義関数 (UDF)
- インクリメンタル処理

# 効率的なデータストレージ形式を使用する

ETL ジョブを作成する場合、変換されたデータを列ベースのデータ形式で出力することをおすすめします。Apache Parquet や ORC などの列指向データ形式は、ビッグデータの保存や分析によく使用されます。データの移動を最小限に抑え、圧縮率を最大化するように設計されています。これらのフォーマットはまた、クエリ処理中の並列読み取りを増やすために、複数のリーダーにデータを分割することも可能です。

データを圧縮すると、保存されるデータ量が減り、読み取り/書き込み操作のパフォーマンスが向上します。AWS Glue は複数の圧縮形式をネイティブにサポートします。効率的なデータ保存と圧縮について詳しくは、「パフォーマンス効率の高いデータパイプラインの構築」を参照してください。

# 適切なスケーリングタイプを使用する

水平スケール (ワーカー数の変更) や垂直スケール (ワーカータイプの変更) のタイミングを理解することは、ETL ジョブのコストやパフォーマンスに影響を与えかねないため、AWS Glue にとって重要です。

一般に、変換が複雑な ETL ジョブはメモリを大量に消費し、垂直スケーリング (たとえば G.1X から G.2X ワーカータイプへの移行) が必要です。AWS Glue は複数のノードで並列にデータを処理する

ように設計されているため、大量のデータを扱う計算集約的な ETL ジョブの場合は、水平スケーリングを推奨します。

Amazon CloudWatch で AWS Glue ジョブメトリクスを綿密に監視すると、パフォーマンスのボトルネックの原因がメモリ不足かコンピューティング不足かを判断するのに役立ちます。AWS Glue ワーカーの種類とスケーリングの詳細については、「AWS Glue によるApache Spark ジョブのスケーリングとデータ分割のベストプラクティス」を参照してください。

# AWS Glue よくある質問のサーバーレス ETL

このセクションでは、 AWS Glue上のサーバーレスETLに関してよく寄せられる質問に対する回答を提供します。

## AWS Glue ジョブに Apache Spark の代わりに Python シェルをいつ使用すべきですか?

基本的な ETL ジョブや、Apache Spark の分散コンピューティング機能を必要としない小さなデータセットがある場合は、Python シェルを使用してください。Apache Spark は、Spark の処理に最適化されている高い処理能力を必要とする、より複雑な ETL ジョブや大規模なデータセットに使用します。

## プロジェクトに推奨される AWS Glue バージョンは何ですか?

通常、 の最新バージョンを使用することをお勧めします AWS Glue。「AWS Glue バージョン」ページには、バージョン間の違いと、Python や Spark のさまざまなバージョンとの互換性が記載されています。

# 次の手順

## AWS Glue 変換について

より効率的なデータ処理のために、 には組み込みの変換関数 AWS Glue が含まれています。この関数は、「Apache Spark」のSQL DataFrameを拡張したDynamicFrameと呼ばれるデータ構造内で、トランスフォームからトランスフォームへと渡されます。DynamicFrame は DataFrame に似ているが、各レコードが自己記述型であるため、最初はスキーマが必要ない点が異なります。

いくつかの AWS Glue PySpark 組み込み関数を理解するには、ブログ記事「 なしで ETL AWS Glue パイプラインをローカルに構築する AWS アカウント」を参照してください。

## 最初の ETL ジョブの作成

以前に ETL ジョブを記述したことがない場合は、3 つの ETL AWS Glue ジョブタイプを使用してデータを Apache Parquet パターンに変換することで開始できます。

ETL ジョブを作成した経験がある場合は、「AWS Glue GitHub の例」 を使用してさらに詳しく調べることができます。

## 料金

料金情報については、「AWS Glue の料金」を参照してください。を使用してAWS 料金見積りツール、さまざまな AWS Glue コンポーネントを使用するための月額コストを見積もることもできます。

# その他のリソース

## リファレンス

- AWS Glue デベロッパーガイド
- AWS Glue DataBrew デベロッパーガイド
- AWS Glue API
- 「AWS Glue ストリーミング ETL」
- AWS Glue Studio
- の Python シェルジョブ AWS Glue
- 「Apache Spark ジョブ」
- AWS Glue PySpark 変換リファレンス
- のデータカタログとクローラ AWS Glue
- AWS Glue スキーマレジストリ
- での口グ記録とモニタリング AWS Glue
- 「AWS Glue バージョン」

## ブログ記事

- Docker コンテナを使用してローカルで AWS Glue ジョブを開発およびテストする
- でのメモリ管理の最適化 AWS Glue
- を使用して Apache Spark ジョブをスケーリングし、データをパーティション化するためのベストプラクティス AWS Glue
- なしで ETL AWS Glue パイプラインをローカルに構築する AWS アカウント
- でパーティション化されたデータを操作する AWS Glue

## 例

- AWS Glue DataBrew サンプルプロジェクト
- AWS Glue インタラクティブセッションの開始方法
- AWS Glue ETL コードサンプルリポジトリ

# パターン

- 「 AWS Glueを使用してAmazon S3からAmazon Redshiftにデータを段階的にロードするETLサービスパイプラインを構築します」
- Infrastructure as Code AWS クラウド を使用してサーバーレスデータレイクを にデプロイして管理する
- データを Apache Parquet に変換するための AWS Glue 3 つの ETL ジョブタイプ

# ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

| 変更 | 説明 | 日付 |
| --- | --- | --- |
| の更新 | AWS Glue ETL セクションにワーカータイプに関する情報を追加しました。 | 2024 年 3 月 13 日 |
| の更新 | ガイド全体でコンテンツが更新されました。 | 2023 年 3 月 15 日 |
| ベストプラクティスを追加しました | パーティショニングを使用して特定のデータをクエリする方法についての情報を追加しました。 | 2021 年 2 月 4 日 |
| 初版発行 | — | 2021 年 1 月 29 日 |

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。