データレイク on AWS
AWS Glue編

AWS Glue

データカタログとETLの２つのコンポーネント

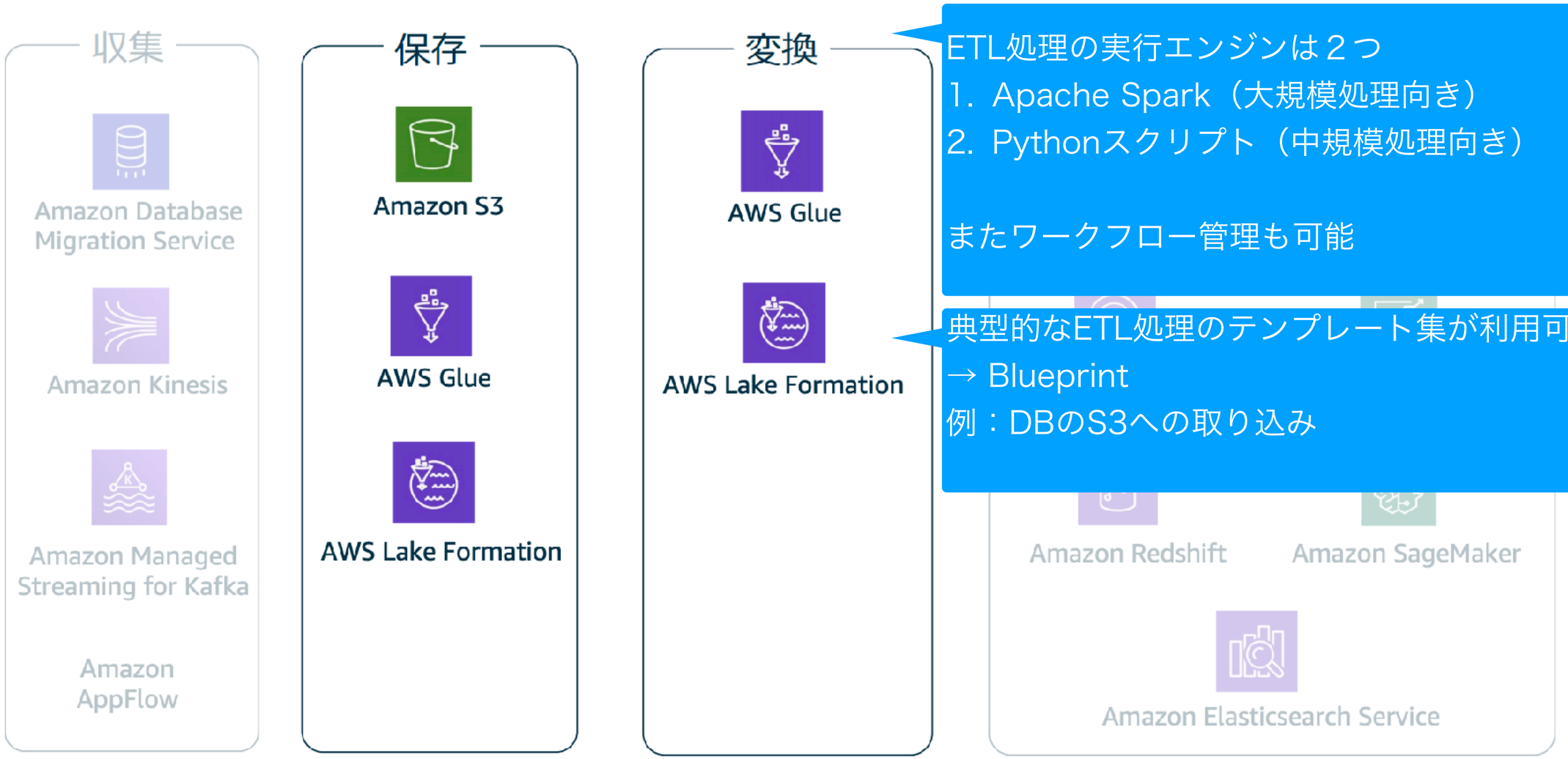
データカタログ：

AWS か JDBC 準拠のソースに保存されたデータを指定するだけでデータ検索が行われ、テーブル定義やスキーマなどの関連するメタデータが AWS Glue データカタログに保存される。

ETL：

Scala または Python で Apache Spark ETL コードを生成しこのコードを使用して、ソースからのデータ抽出、ターゲットのスキーマに合わせたデータ変換、ターゲットへのロードを行う。

データレイクのコンポーネントに対応する AWS サービス



ETL処理の実行エンジンは２つ
1. Apache Spark（大規模処理向き）
2. Pythonスクリプト（中規模処理向き）

またワークフロー管理も可能

典型的なETL処理のテンプレート集が利用可能
→ Blueprint
例：DBのS3への取り込み

aws

- https://aws.amazon.com/jp/blogs/news/data-lake-starting-with-aws-book/

Glueデータカタログ

Glueクローラでスキーマを作成

・Glueクローラを使うことで各種データソースをクローロングしてメタ情報を自動で推定し、カタログに登録できる

・クローラの実行：オンデマンドor定期的なスケジュールorイベント時にトリガーで実行可能

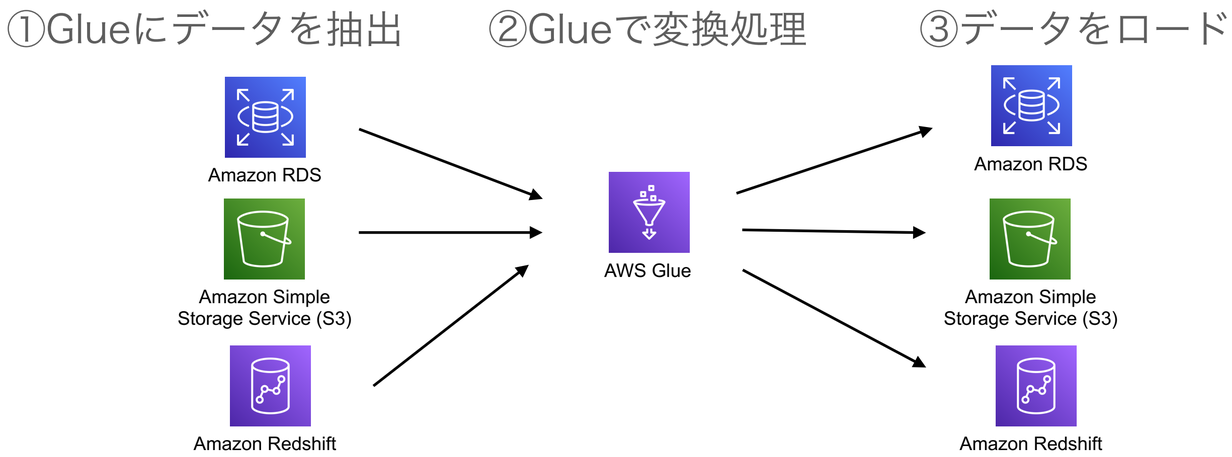
ETLサービス：Glueジョブ

並列分散処理でETL処理を行う

・Apache SparkとPythonスクリプトの２つに対応

・コードの自動生成やカスタムコードの実行が可能

・任意のタイミングでGlueジョブやGlueクローラを実行するワークフローを形成する機能も含む



ETL処理の例
データの圧縮と列指向フォーマットへの変換

処理内容

事前にS3にTSVファイルがアップロードされている

このファイルをParquetに変換し処理結果をS3に出力する

Glueジョブの作成と実行

1. IAMで、S3への書き込み許可を与えるIAMポリシーをGlueジョブが使うIAMロールにアタッチする
2. Glueで[ジョブの追加]しIAMロール、Type（Spark or Python）、データソースを選択
3. [データターゲットでテーブルを作成する]で形式に[Parquet]を選択
4. [ジョブを保存してスクリプトを編集する]→Sparkのコードが生成されるので[ジョブの実行]
5. ジョブが正常完了すれば指定の場所に結果が出力されているはず

編集可能！

☝Parquetに変換したことでAthenaでクエリを実行すると変換前に比べてスキャン容量が大幅に削減される！

２種類のジョブの使い分け

Spark \Pythonスクリプト

Spark：

並列処理を簡単にスケールできる

Python Shell：

入力データが小さい場合は簡潔にジョブを記述しクイックにETL

☝10GB以上ならSpark、100MB未満ならPython、その間に関してはユースケースを元に実測して判断する

今後学習したいこと

Apache Spark

・巨大なデータに対して高速に分散処理を行うオープンソースのフレームワーク。JavaやScala、Pythonなどいろいろなプログラミング言語のAPIが用意されている

・クラスタ上のデータをSQLで処理できる「Spark SQL」や、機械学習のための「MLlib」、グラフ処理のための「GraphX」、ストリーミング処理のための「Spark Streaming」など、便利なコンポーネントが付属

https://www.atmarkit.co.jp/ait/articles/1608/24/news014.html