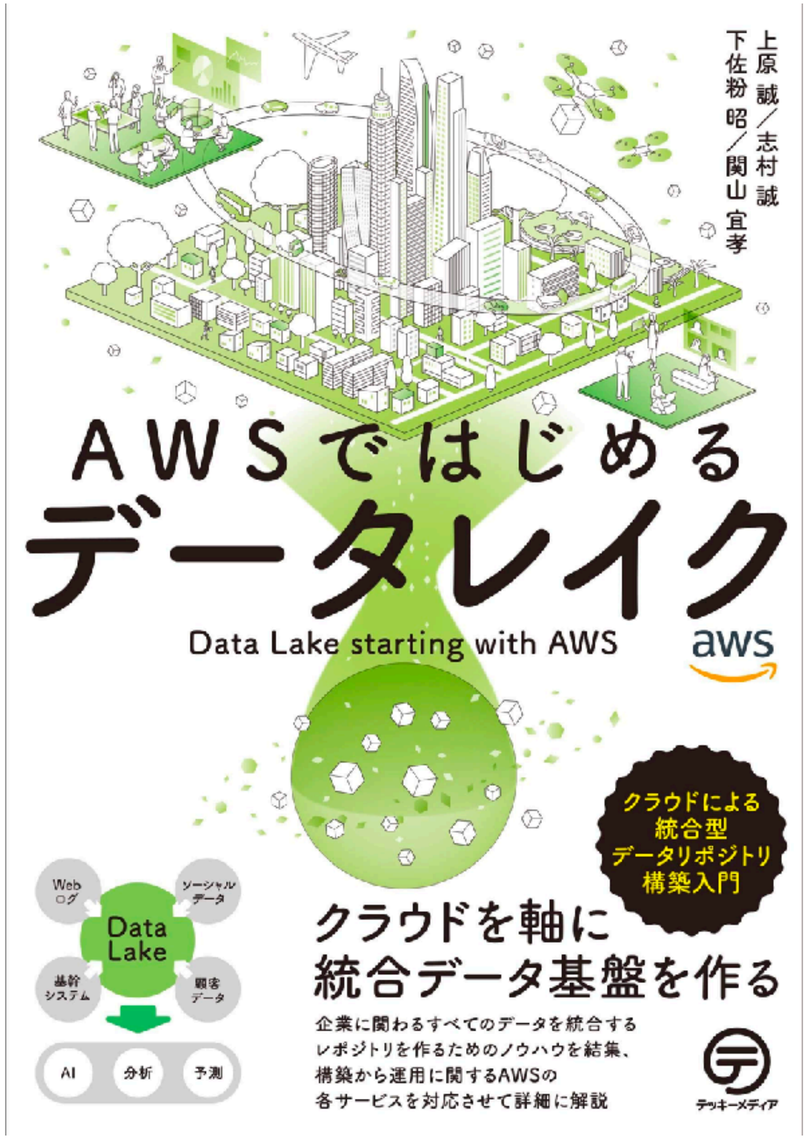
『AWSではじめるデータレイク』まとめ



この書籍を学習する個人的な目的

・クラウド上でのデータ基盤について学びたい

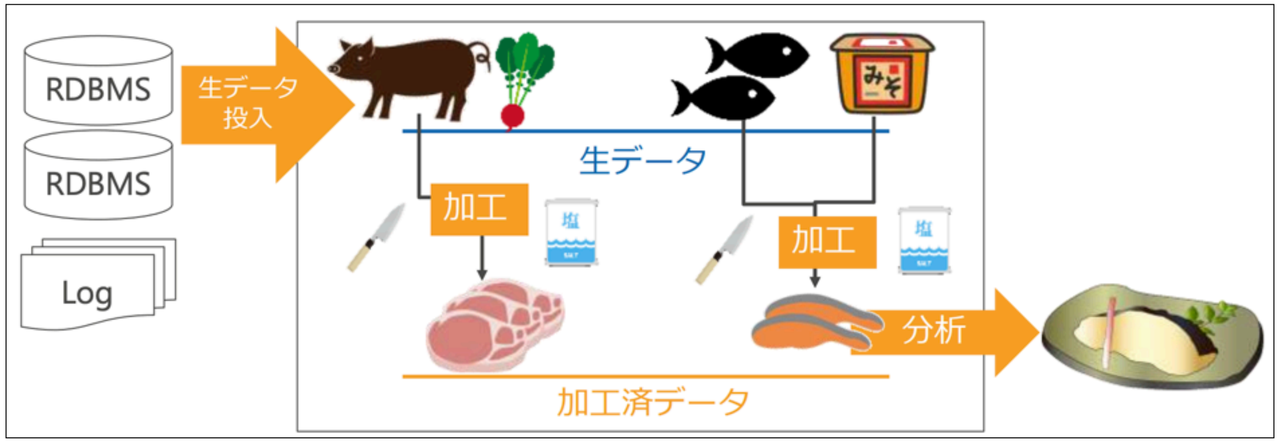
・データ処理の流れを掴みたい

・各種AWSのサービスについて手を動かして学びたい

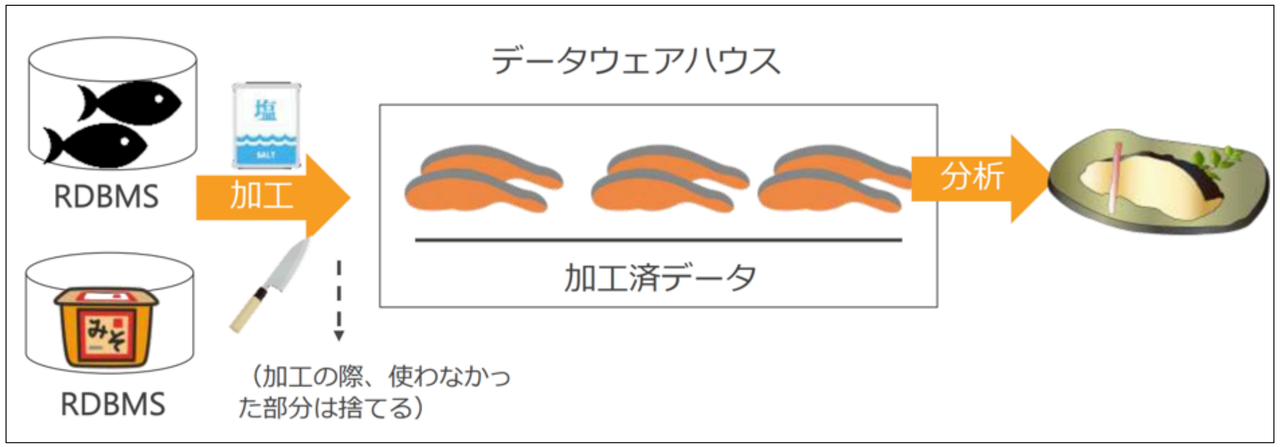
目次

データレイクについて

- **データレイク：**
  - 生データを一元的に保存
  - 加工済みデータも保存



- **データウェアハウス：**
  - 加工後のデータを保存
  - 目的に特化したデータしか保存して
    いないため新しいニーズへの対応が困難

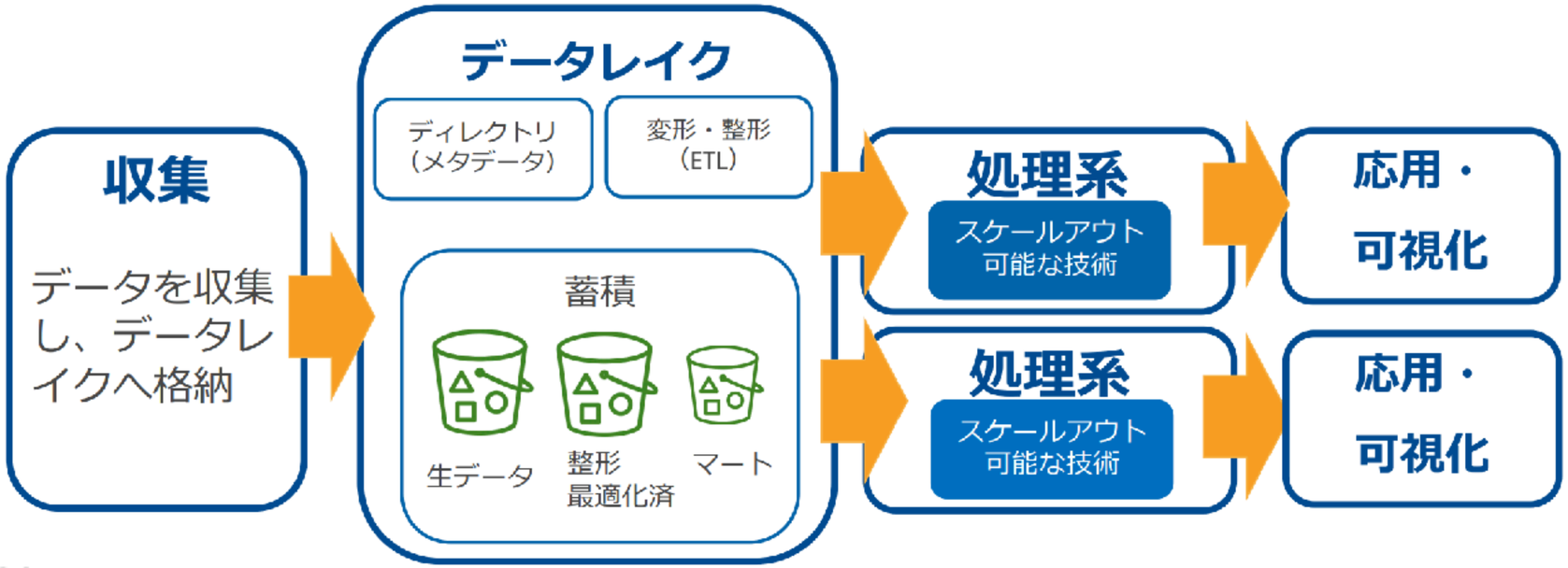


https://aws.amazon.com/jp/blogs/news/data-lake-starting-with-aws-book/

データレイクを中心にした構成

・データの蓄積とデータの処理系が分離

・用途に合わせて適切なツール（サービス）を選択



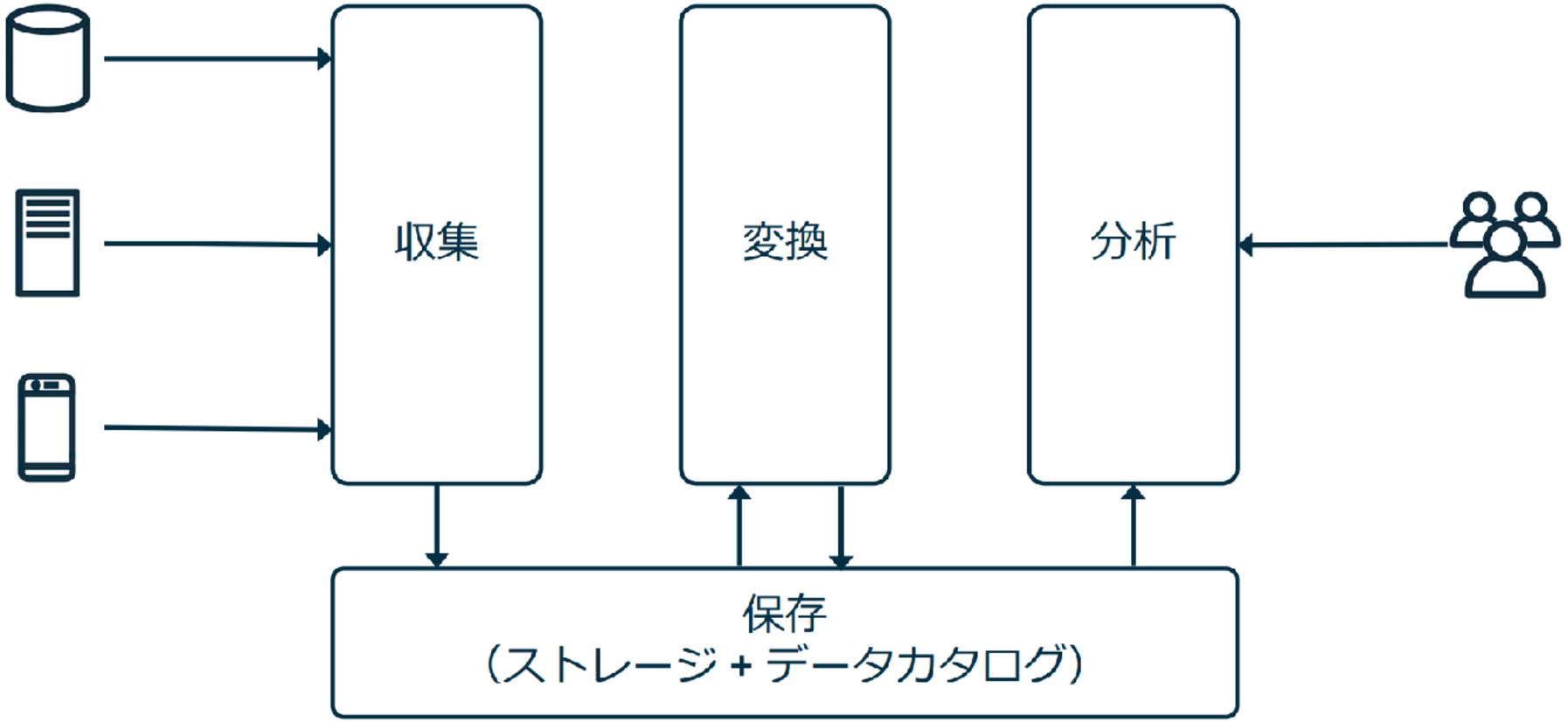
https://aws.amazon.com/jp/blogs/news/data-lake-starting-with-aws-book/

データレイクの全体像

４つのコンポーネント

1. **収集**
2. **保存**：データをストレージに保存し、データカタログ（データの詳細）を管理する
3. **変換**：適切な加工整形処理を行う
4. **分析**：加工整形済みのデータを活用する

ベーシックなデータレイクのアーキテクチャ



aws

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

データレイクのコンポーネントに対応する AWS サービス

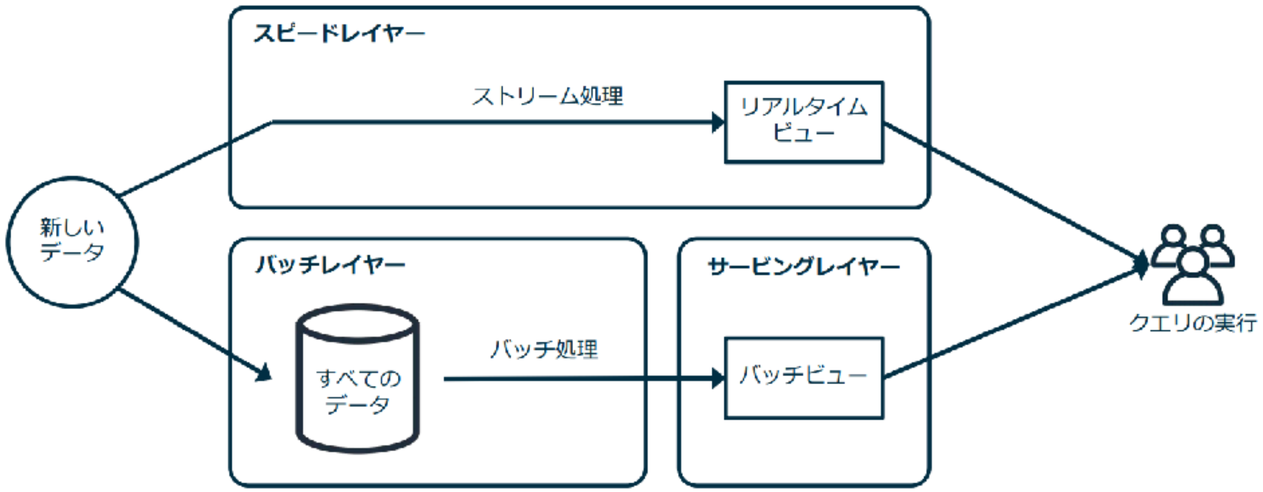
aws

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

ラムダアーキテクチャ

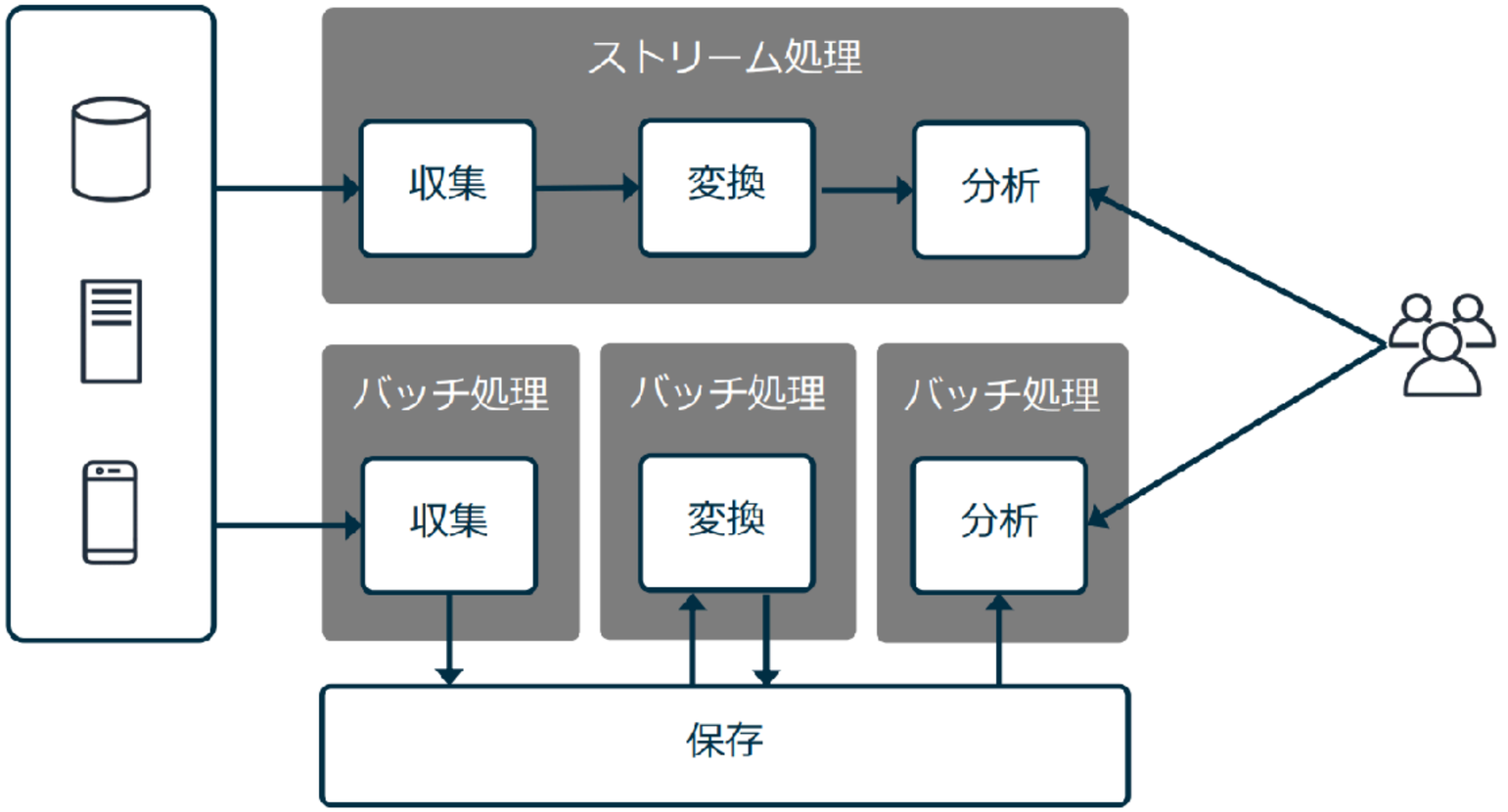
典型的なアーキテクチャの他に考慮すべきアーキテクチャ

1. 大量のデータを保持し定期的な処理を行うバッチレイヤー
2. 新しく入ってきたデータをストリーム処理するスピードレイヤー
3. 両者を組み合わせて参照するサービングレイヤー



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

ストリーム処理を組み込んだデータレイクのアーキテクチャ



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

1.データの収集

代表的なデータソースとその収集方法

・ファイル：

コマンドラインツール、ファイル転送ツール（ex. Embulk）

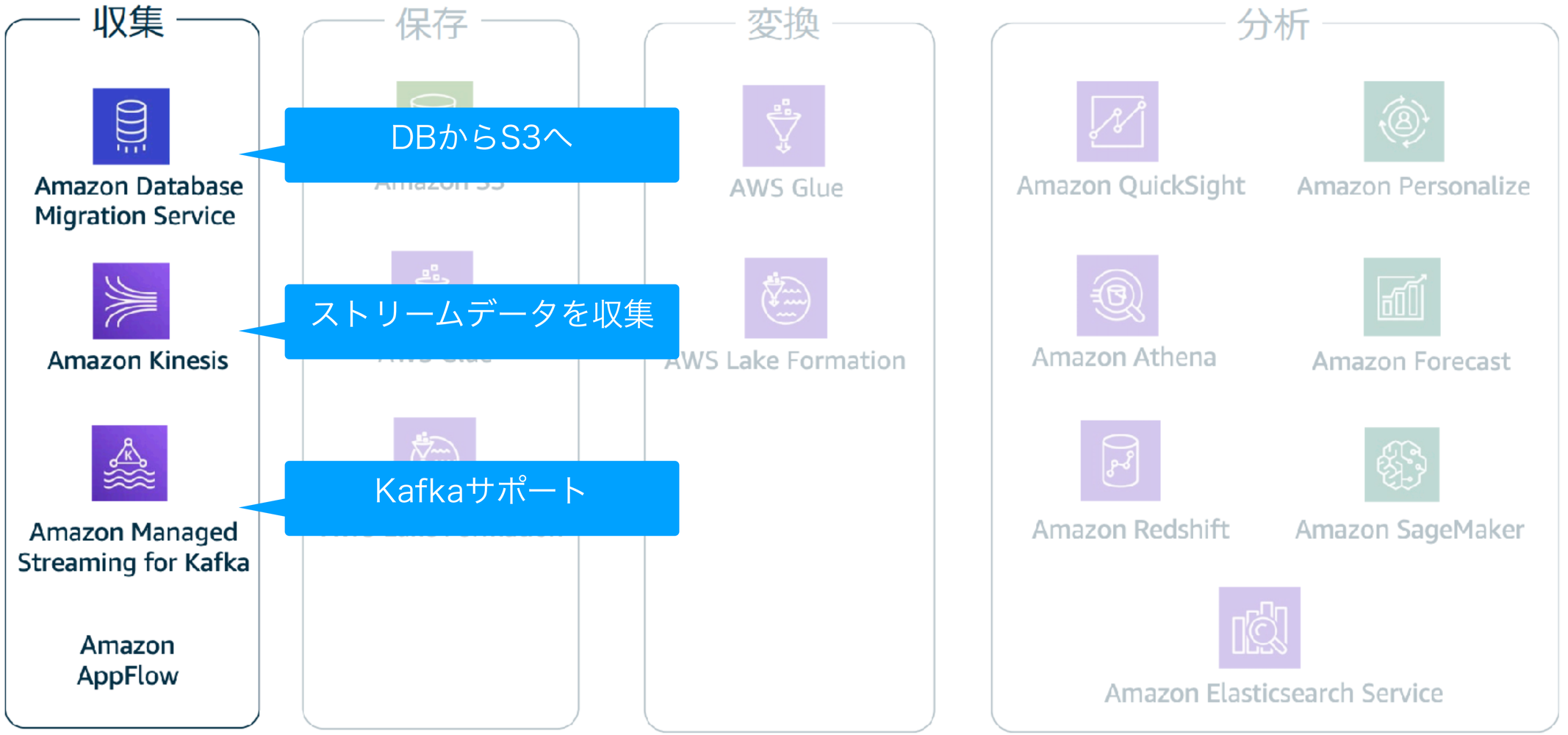
・ストリームデータ：

ログ収集ツール（ex. fluntd）、ストリームデータ処理基盤（ex.　Apache Kafka）を用いた処理

・データベース：

バッチ処理、CDC（Change Data Capture:テーブルへの変更をすぐに処理）

データレイクのコンポーネントに対応する AWS サービス



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

2.データの保存
ストレージに求められること

・耐久性

・可用性（障害への強さ）

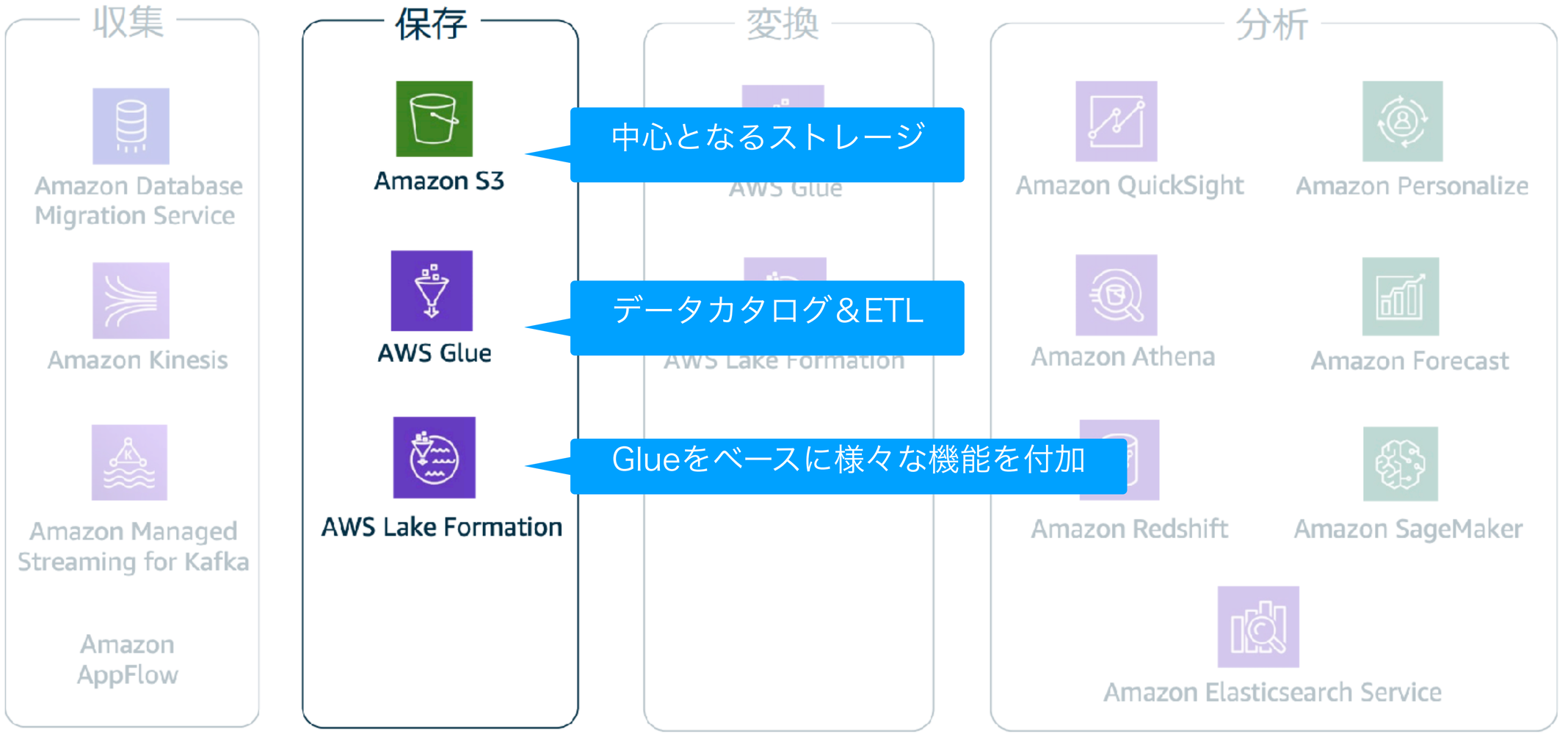
・スケーラビリティ

・データの種類

・コスト

・セキュリティ/権限管理

データレイクのコンポーネントに対応する AWS サービス



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

データカタログ

物理的なデータの場所と、データの構造やアクセス方法の関係性を定義

- **データの発見しやすさ：**

  どこになにがあるかを把握できるようにする

- **データの活用しやすさ：**

  さまざまなツールからカタログを通してデータにアクセスできるようにする

- **権限管理：**

  必要な権限を必要な人に付与し適切に管理できるようにする

- **監視・監査：**

  不正なアクセスがなかったか、誰がどのようにデータを利用しているか等を管理して、ガバナンスを確保する

- **通知：**

  なにかあった際に通知を飛ばして対応できるようにする

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。

3.データの変換
分析に適した形へ：ETL処理

分析時のビジネス的な最適化

・日付に関わる処理：西暦・和暦など表示形式変更，JSTからUTCへの変換など

・不正な値の処理：Null値や空っぽの値の変換など

・文字列や値の統一：大文字小文字の統一、整数・少数の統一、表記揺れの統一など

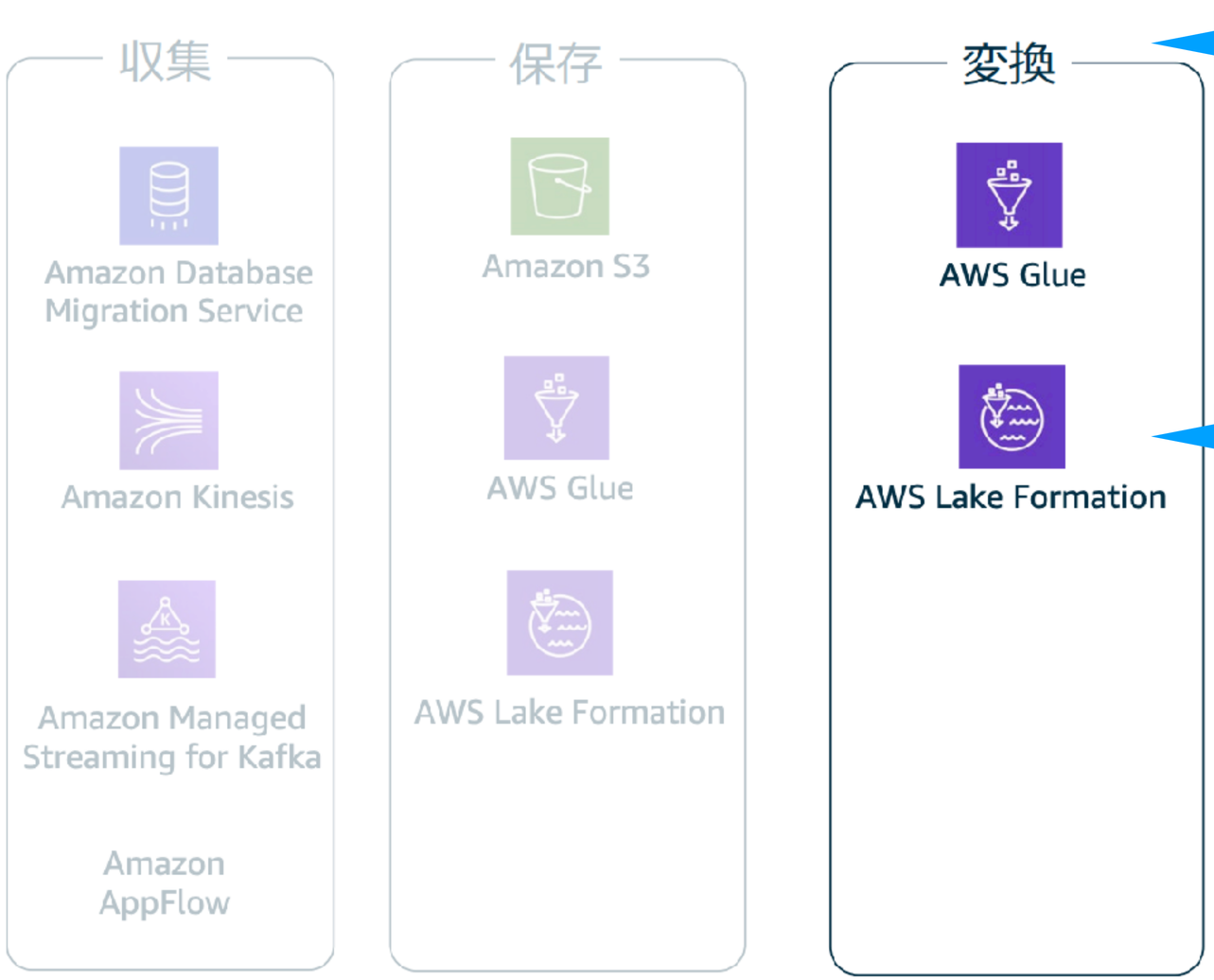
・テーブルの結合処理：取引ログテーブルと取引先テーブルを取引先IDで結合など

分析時のパフォーマンス的な最適化

・ファイルサイズの変換：大量の細かいファイルを、数百MBのファイルに集約など

・ファイルフォーマットの変換：JSON から Parquet/ORC への変換など

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

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

4.データの分析・活用

データ活用の全体像

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

１）BI（データの集計・分析・可視化そしてレポーティング）

AWSサービス：Amazon QuickSight

サーバーレス：

AWS側で管理のため運用の手間が少ない

様々なデータソースへ対応：

AWSサービス、ファイル、SaaS など 20 以上に対応

豊富な分析機能：

多様なビジュアル機能に加え，機械学習ベースの分析機能（時系列予測や異常検知など）も

2）SQLによるアドホック/探索的な分析

AWSサービス：Amazon Athena

・S3上のデータに対して直接 SQLクエリを投入できる

・クエリエンジンとしてPrestoを使用し、並列分散処理による高速なクエリ実行が可能

・スキャンした量に応じた従量課金、クエリを投げなければ課金なし

☝Apache ParquetやApache ORCというデータフォーマットを使うと、使う列のデータだけを選択的にスキャンすることができスキャン量を削減できる

☝頻繁に使われる列を元にパーティションと呼ばれる塊に分けることでさらにスキャン量を削減できる

3）データウェアハウスによる複雑/定常的なSQL

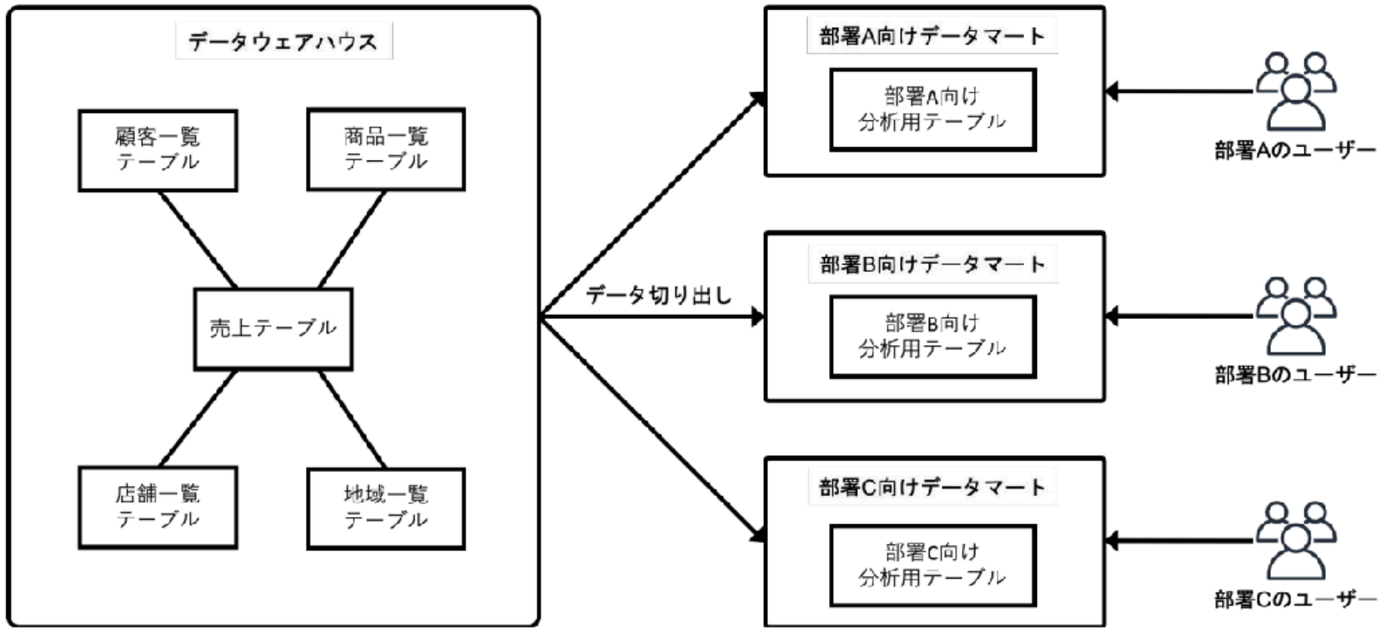
定期的に重たい複雑な処理を行う際はDWHを活用

同一クエリの定期実行：業務 KPI を日次・週次で更新

高いパフォーマンス / SLA：業務の核となるため，高いサービスレベルが求められる

スケーラビリティ：増え続けるデータに対応し続ける必要

細かな権限管理：機微データを含んだり，複数部署のユーザーが利用したりするため



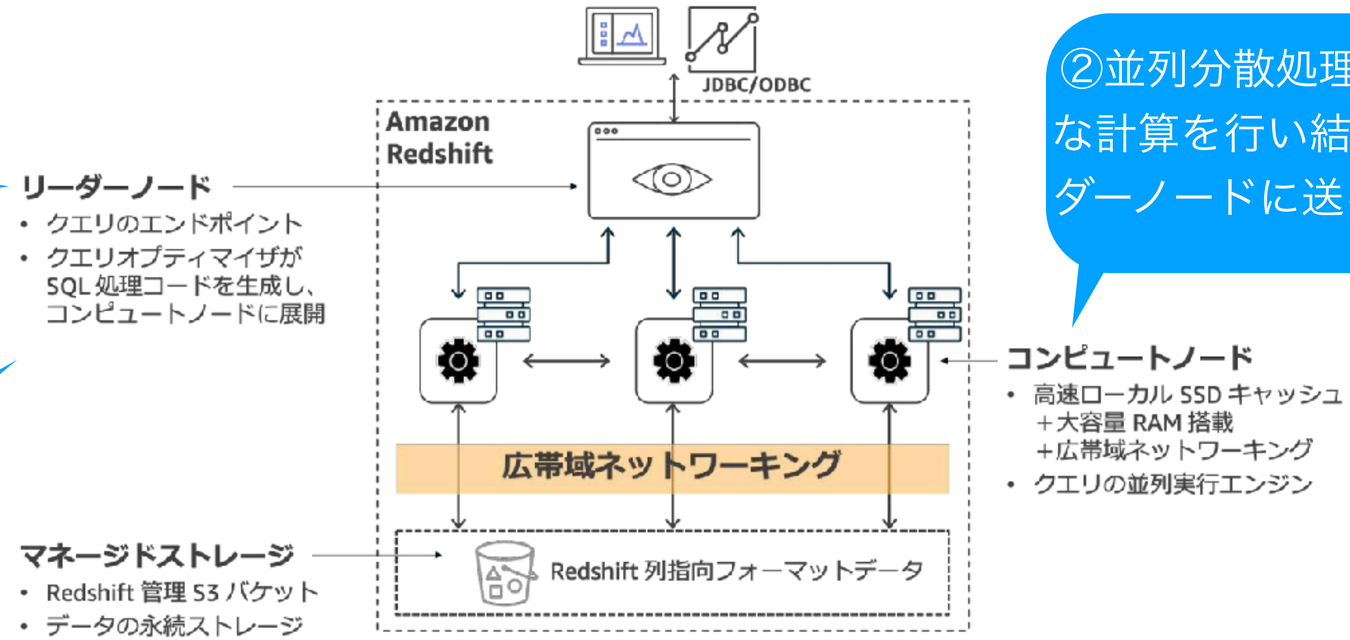
https://aws.amazon.com/jp/blogs/news/data-lake-starting-with-aws-book/

3）データウェアハウスによる複雑/定常的なSQL

AWSサービス：Amazon Redshift

①リーダーノードがユーザーからクエリを受けて複数のコンピュートノードに処理を分割して送る

③結果を集約しユーザーへ返す



②並列分散処理により高速な計算を行い結果をリーダーノードに送る

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

４）Python /Rによる応用的な分析

統計解析や機械学習処理のための複雑な前処理など

Amazon EMR

・複数のマシン上で、Apache HadoopやApache Sparkをベースとした処理を行う

・大規模データを読み込みML用のデータセットを作ったり、Spark MLlibというML用のライブラリが利用可能

４）Python /Rによる応用的な分析

統計解析や機械学習処理のための複雑な前処理など

Amazon SageMaker

・マネージド機械学習サービス

・ノートブックインスタンスと呼ばれるJupyterベースのインタラクティブな開発環境を提供

・直接EMRのSparkクラスターにアクセスして処理を実行したり、RedshiftやAthenaに接続しSQLクエリの実行結果が読み込み可能