Information Technology Institute

Yelp CaseStudy

Unlocking Insights: Analyzing Yelp's Businesses and Reviews in North America

A Graduation Project Submitted in Fulfillment of
Scholarship Requirements in

Data Engineering

Prepared By

Ahmed Osama Talaat

Abdallah Amr Abdelaziez

Alaa Rashad Saad

Yasmeen Mohammed Mohammed

Hasnaa Magdi Saad

Mohamed Malek Mahmoud

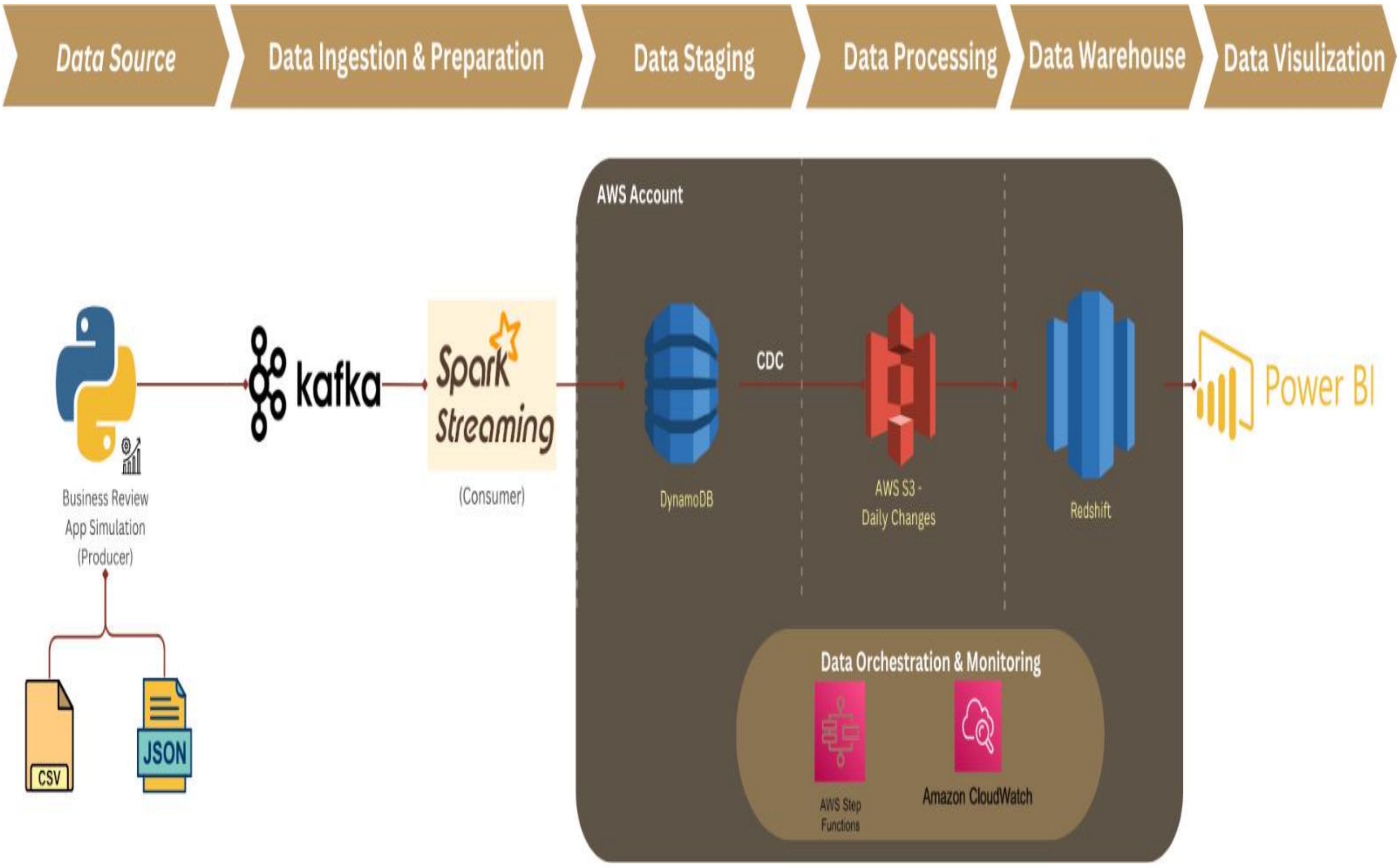
Introduction:

Our big data project centered around processing and analyzing Yelp business review data. The project integrated Python, Kafka, AWS services (DynamoDB, S3, Redshift), PySpark, AWS Lambda, and Power BI to create an end-to-end data pipeline for real-time streaming, change data capture (CDC), daily batch processing, and data visualization.

Yelp Dataset

A trove of reviews, businesses, users.

This dataset is a subset of Yelp's businesses, reviews, and user data. It was originally put together for the Yelp Dataset Challenge which is a chance for students to conduct research or analysis on Yelp's data and share their discoveries. In the most recent dataset you'll find information about businesses across 8 metropolitan areas in the USA and Canada.



Project Pipeline :

1. Data Ingestion:

- Collect data from two sources: CSV and JSON files.

2. Business Review App Simulation (Producer):

- Python-based simulation generating or processing business review data.
- Produces data for further processing.

3. Data Transfer:

- Data sent to Kafka for reliable message-based transfer.

4. Real-time Processing:

- Spark Streaming consumes data from Kafka for real-time or near real-time processing.
- Transformation, filtering, or enrichment of data occurs.

5. Staging Area - DynamoDB (AWS Cloud):

- DynamoDB utilized as a staging area for temporary storage of processed data.
- Serves as an intermediate step for data manipulation or enrichment.

6. Change Data Capture (CDC) in Cloud:

- AWS Step Functions and AWS CloudWatch monitor and manage data changes in the cloud environment.
- Tracks modifications and updates to the data stored in DynamoDB.

7. Daily Data Staging - S3 (AWS Cloud):

- Data stored in S3 as a staging area, possibly organized based on daily batches.
- Acts as an interim storage before further processing or archival purposes.

8. Data Warehousing - Redshift (AWS Cloud):

- Processed and structured data from S3 or DynamoDB is loaded into Redshift.
- Data stored in a format optimized for analytical queries and reporting.

9.Transformation and Querying:

- Further transformations, aggregations, or joining of data within Redshift.
- Enables complex querying for analytics and reporting purposes.

10. Dashboard Creation - Power BI:

- Data from Redshift utilized to create interactive dashboards and reports on Power BI.
- Visualization of key metrics and insights for stakeholders.

11. AWS Services Integration:

- AWS Step Functions used for orchestrating and managing the workflow of the pipeline.
- AWS CloudWatch employed for monitoring various services and tracking performance metrics.

This detailed pipeline encompasses the step-by-step flow of data from ingestion to visualization, emphasizing data processing, storage in different stages, cloud-based services utilization, and eventual visualization for business insights and decision-making.

1. **Data Ingestion:**

The primary objective of this step is to collect data from two different sources - CSV and JSON files. Data ingestion is the foundational step in the pipeline, responsible for acquiring raw data for subsequent processing and analysis.

Details:

- **CSV and JSON Sources:**
    - CSV (Comma-Separated Values) and JSON (JavaScript Object Notation) are common data formats used for storing structured data.
    - CSV files represent tabular data with rows and columns, typically separated by commas or other delimiters.
    - JSON files store data in key-value pairs and hierarchical structures, offering flexibility in representing complex data.
- **Data Collection Process:**
    - In this phase, automated scripts, programs, or processes are employed to retrieve data from the designated CSV and JSON sources.
    - The process involves accessing files stored locally or from remote sources such as databases, APIs, or web services.
- **Data Validation and Cleaning:**

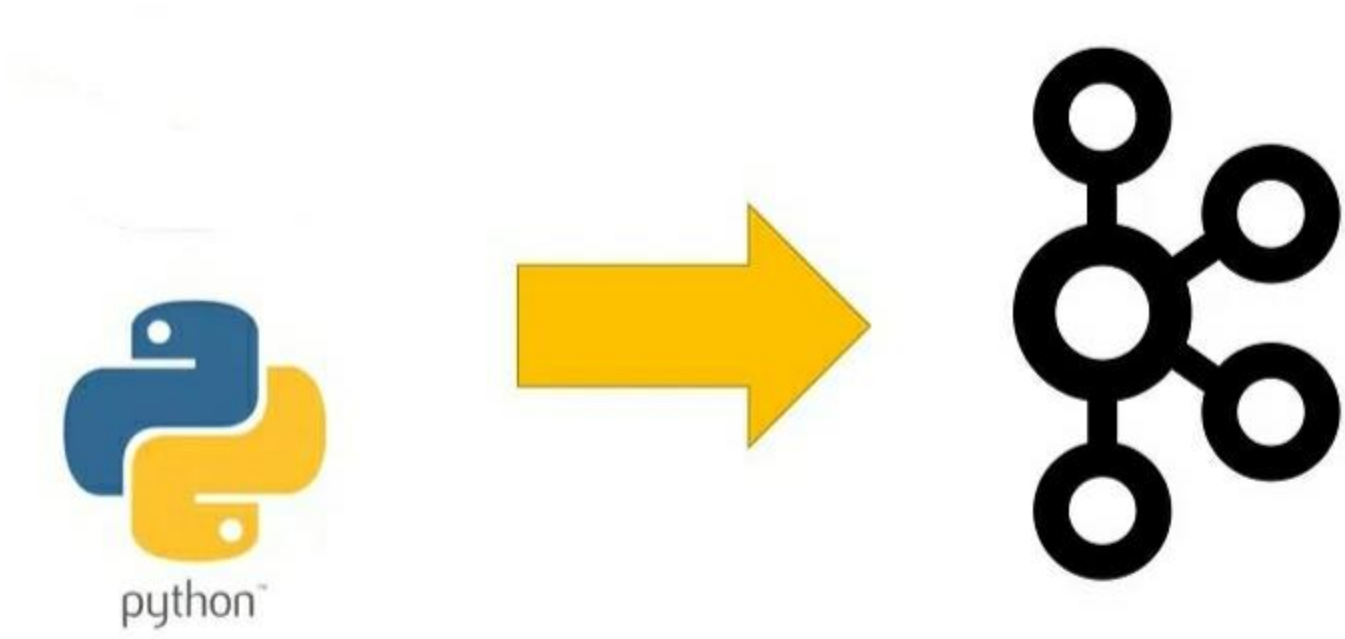
    - Validation checks ensure that the retrieved data adheres to expected formats, schemas, or constraints.
    - Data cleaning involves identifying and handling inconsistencies, missing values, or errors within the datasets.
- **Data Integration and Consolidation:**
    - Once collected, data from both CSV and JSON sources might need to be integrated and consolidated if they contain related or complementary information.
    - Transformation might be necessary to standardize formats or resolve schema differences for unified processing.
- **Metadata Generation:**
    - Metadata, including information about data sources, timestamps, and data characteristics, might be generated or appended to provide context and aid downstream processing.
- **Data Cataloging or Tagging:**
    - To facilitate efficient data management and future accessibility, data might be cataloged or tagged with relevant identifiers, descriptions, or labels.

Importance:

- Data ingestion lays the groundwork for subsequent stages in the pipeline by sourcing the initial raw data required for analysis, transformation, and storage.
- Accurate and efficient data ingestion is critical for ensuring the quality, reliability, and usability of data downstream.

2. Business Review App Simulation (Producer):

This step involves the execution of a Python-based application that simulates the generation or processing of business review data. It acts as the producer of data, generating information relevant to a business review application.



Details:

- **Python Application:**
  - A Python-based program or set of scripts designed to mimic or simulate the behavior of a business review application.
  - It may create artificial but realistic data related to business reviews, ratings, user interactions, or other relevant metrics.
- **Data Generation or Processing:**
  - Generation: Simulates the creation of synthetic data representing reviews, ratings, comments, or other aspects of a business review platform.
  - Processing: In some cases, the Python app might manipulate existing data, perform computations, or apply transformations to simulate real-world scenarios.

- **Emulation of Business Logic:**
    - The application may emulate the logic and behavior typical of a business review platform, considering functionalities like user interactions, review submissions, ratings, etc.
- **Data Quality Considerations:**
    - Ensuring that the simulated data maintains quality standards similar to actual business data to facilitate realistic downstream processing and analysis.
- **Scalability and Performance:**
    - Designing the simulation to handle large volumes of data efficiently to mimic real-world conditions, ensuring scalability and performance of the application.

Importance:

- This step allows for the generation or manipulation of data that resembles real-world scenarios, providing valuable input for subsequent stages of the pipeline.
- The simulated data serves as input for testing, development, and validation of downstream processes without relying on actual live data.

3. Data Transfer:

This step involves the movement of data generated or processed by the Business Review App Simulation (Producer) to Kafka for reliable message-based transfer.

Details:

- **Kafka as a Message Broker:**
  - Kafka serves as a distributed messaging system designed for high-throughput, fault-tolerant, and scalable data transfer.
  - It operates based on a publish-subscribe model, allowing multiple producers to send data to multiple consumers.
- **Producer-to-Kafka Communication:**
  - The Python-based simulation or producer application sends data to Kafka topics, organizing data streams into logical categories.
  - Data is published to specific Kafka topics for further consumption by downstream components.
- **Reliable Message Delivery:**

    - Kafka ensures reliable message delivery by persisting data to disk and replicating across multiple brokers.
    - Messages are retained for configurable durations, allowing consumers to replay or consume historical data.
- **Scalability and Fault Tolerance:**
    - Kafka's distributed nature allows it to scale horizontally by adding more brokers to handle increased data loads.
    - It is fault-tolerant, ensuring minimal data loss and high availability even in the event of broker failures.
- **Asynchronous Communication:**
    - The decoupling of the producer and consumer allows asynchronous communication, ensuring that the producer can continue producing data at its own pace without waiting for consumers.

Importance:

- Data transfer to Kafka establishes a reliable and scalable communication channel, facilitating the seamless flow of data between components in the pipeline.
- Kafka's features ensure fault tolerance, scalability, and asynchronous communication, crucial for handling high volumes of data.

4. **Real-time Processing:**

In this step, Spark Streaming is utilized to consume and process data from Kafka in real-time or near real-time, enabling continuous computation on streaming data.

Details:

- **Spark Streaming:**
  - Spark Streaming is an extension of the Apache Spark platform designed specifically for handling real-time data streams.
  - It breaks down the continuous data streams into mini-batches for processing, allowing high-throughput and fault-tolerant stream processing.
- **Consuming Data from Kafka:**
  - Spark Streaming subscribes to Kafka topics to consume incoming data streams.
  - It receives data published by producers and processes these streams in micro-batch intervals.
- **Transformation and Processing:**

    - Data received from Kafka is subjected to transformations, such as mapping, filtering, aggregations, or joining with other datasets.
    - Complex computations or analytics can be applied to derive meaningful insights from the streaming data.
- **Windowed Operations:**
    - Spark Streaming supports windowed operations, enabling computations over sliding windows of data within specified time intervals.
    - This facilitates the analysis of data over a window of time, offering insights into trends or patterns.
- **Output to Next Stage or Storage:**
    - Processed data can be directed to storage for further analysis, staging, or onward transmission to subsequent stages in the pipeline.

Importance:

- Spark Streaming enables real-time or near real-time processing of streaming data, allowing for rapid insights and timely decision-making.
- The ability to perform transformations and computations on continuous data streams is vital for various real-time applications and analytics.

5. Staging Area - DynamoDB (AWS Cloud):

DynamoDB, a NoSQL database service provided by AWS, serves as a staging area for temporarily storing processed data before further processing or storage.

Details:

- **NoSQL Staging Area:**
    - DynamoDB is a fully managed NoSQL database offered by AWS, known for its scalability, high performance, and low latency.
    - It allows for flexible and quick storage of structured data without the need for schema definitions.
- **Temporary Data Storage:**
    - Processed data from Spark Streaming or other components of the pipeline is temporarily stored in DynamoDB.
    - It acts as an intermediary stage, holding data before subsequent processing or moving it to a more permanent storage solution.
- **Scalability and Performance:**
    - DynamoDB scales automatically to accommodate varying workloads, ensuring consistent performance as data volumes fluctuate.
    - Its low-latency capabilities make it suitable for applications requiring rapid data access and retrieval.

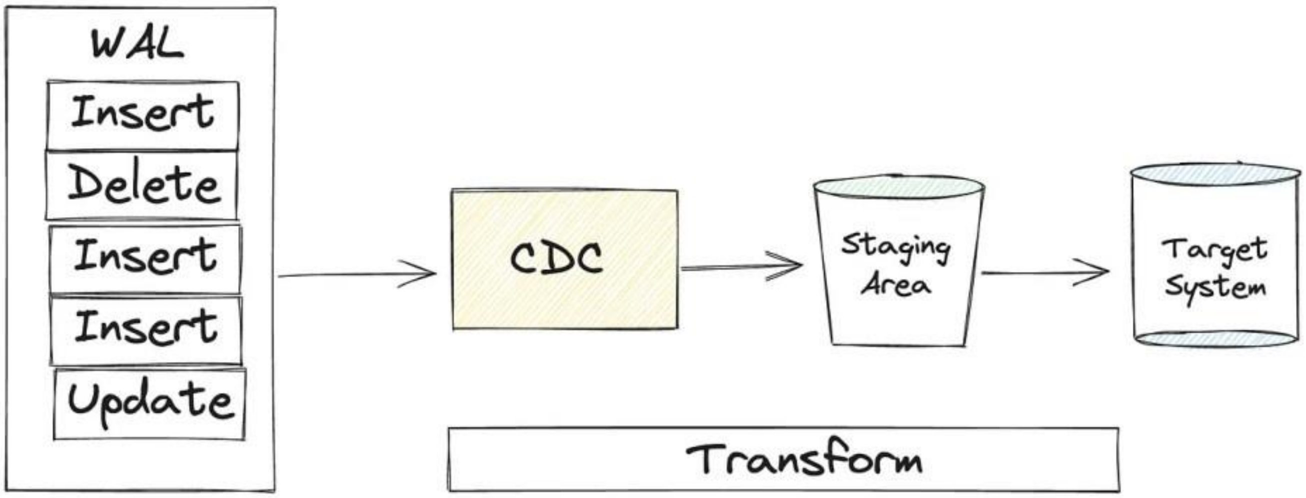
- **AWS Integration and Security:**
    - DynamoDB seamlessly integrates with other AWS services, facilitating easy data transfer and integration within the AWS ecosystem.
    - AWS security features ensure data encryption, access control, and compliance with security best practices.
- **Flexible Data Models:**
    - DynamoDB's flexible data model allows for the storage of varying data types and structures, accommodating diverse data formats received from previous processing stages.

Importance:

- DynamoDB serves as a temporary holding area for processed data, providing agility and flexibility in managing data before it undergoes further transformations or moves to permanent storage.
- Its scalability and performance capabilities ensure efficient handling of varying workloads and data volumes.

6. Change Data Capture (CDC) in Cloud:

This step involves the use of AWS Step Functions and AWS CloudWatch to monitor and manage data changes in the cloud environment, specifically targeting DynamoDB.

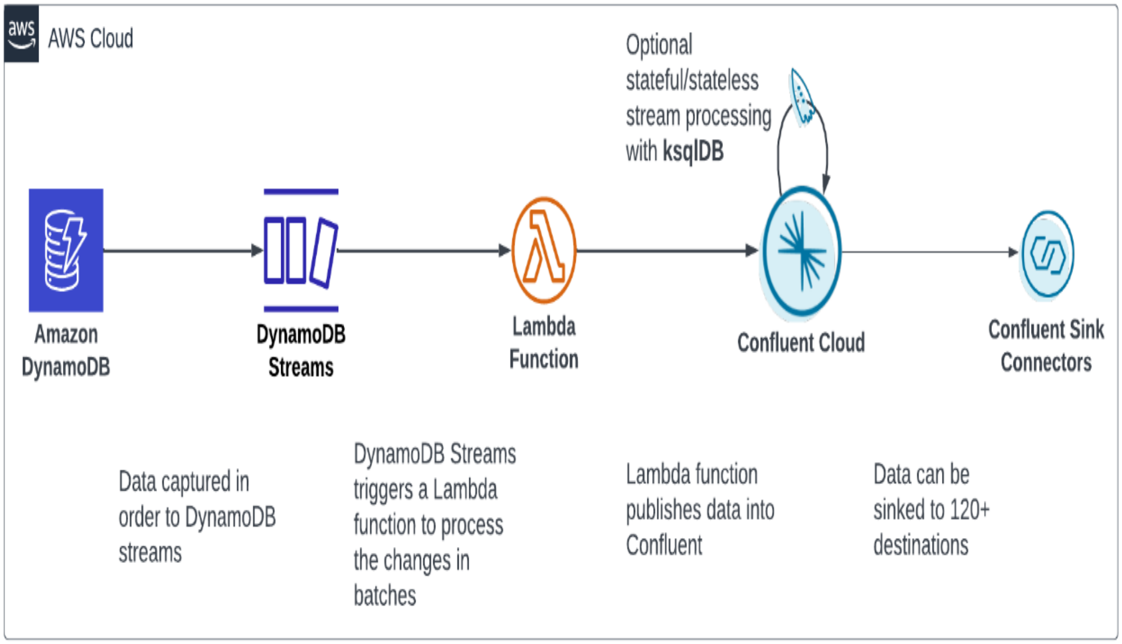


Details:

- **AWS Step Functions:**
  - Step Functions are used to coordinate and automate the workflow of the pipeline, including tracking changes in DynamoDB.
  - They enable the creation of state machines to define and execute multi-step workflows, including CDC processes.
- **Change Data Capture (CDC):**
  - CDC mechanisms track changes made to the data stored in DynamoDB tables, capturing modifications such as inserts, updates, or deletions.
  - These changes are recorded in a way that allows applications or processes to identify what data has been modified and how.
- **AWS CloudWatch Integration:**
  - CloudWatch is utilized for monitoring and logging purposes, allowing the tracking of DynamoDB activity and performance metrics.

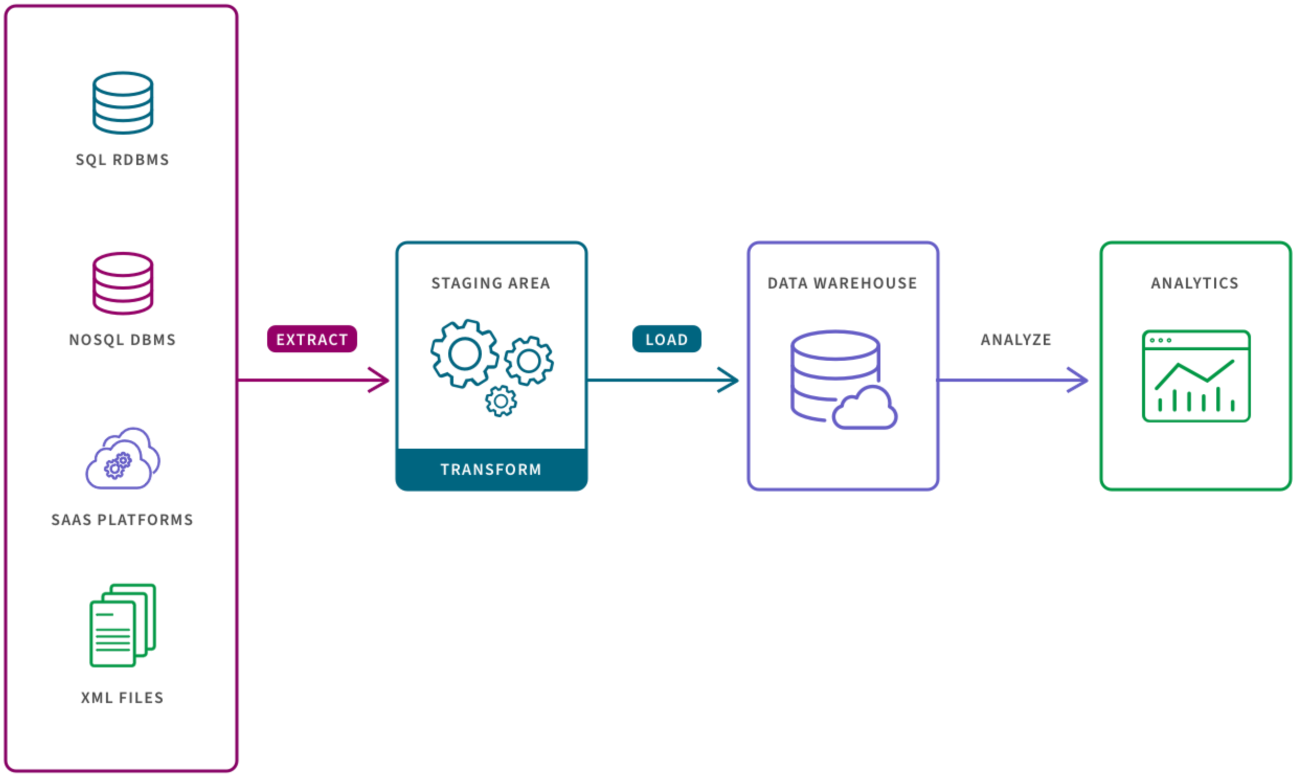
  - It provides insights into resource utilization, system-wide events, and operational health, including metrics related to DynamoDB activities.

- **Real-time Monitoring and Alerts:**
  - CloudWatch facilitates real-time monitoring and triggers alerts or notifications based on predefined thresholds or conditions, ensuring timely responses to critical events or issues.
- **Automated Workflows and Responses:**
  - Step Functions combined with CloudWatch enable the setup of automated responses or workflows triggered by specific data changes, facilitating event-driven actions based on CDC.


AWS Cloud
Optional stateful/stateless stream processing with **ksqlDB**
Amazon DynamoDB
DynamoDB Streams
Lambda Function
Confluent Cloud
Confluent Sink Connectors
Data captured in order to DynamoDB streams
DynamoDB Streams triggers a Lambda function to process the changes in batches
Lambda function publishes data into Confluent
Data can be sinked to 120+ destinations

Importance:

- CDC in the cloud, facilitated by Step Functions and CloudWatch, provides real-time tracking and monitoring of data changes in DynamoDB.
- It enables timely responses to data modifications and facilitates automated workflows or alerts based on defined conditions.



7. Daily Data Staging - S3 (AWS Cloud):

In this step, Amazon S3 (Simple Storage Service) is utilized as a staging area to store daily data batches, serving as an interim storage before further processing or analysis.

Details:

- **Object Storage in S3:**
    - Amazon S3 is a highly scalable, durable, and secure object storage service offered by AWS.
    - It provides a simple web services interface to store and retrieve data, supporting a wide variety of use cases.
- **Organizing Data by Daily Batches:**
    - Data processed or captured from the CDC process or DynamoDB may be organized into daily batches.
    - This partitioning by time intervals allows for structured management and retrieval of data for specific periods.
- **Staging for Further Processing:**
    - S3 serves as a staging area, holding data temporarily before it undergoes additional processing or transformation.
    - It acts as an intermediary stage before the data moves to a more permanent storage or processing solution.
- **Durability and Scalability:**
    - S3 offers high durability with data replicated across multiple availability zones, ensuring data resilience.
    - Its scalable nature allows it to accommodate varying data volumes and access patterns.
- **AWS Integration and Access Controls:**
    - Integration with other AWS services allows seamless data transfer and integration within the AWS ecosystem.

- S3's access controls enable defining granular permissions and security policies to manage data access.

Importance:

- S3 serves as a reliable and scalable staging area for daily data batches, enabling organized storage and facilitating subsequent processing or analysis.
- Its durability, scalability, and integration capabilities make it a suitable solution for temporary data storage in the pipeline.

8. Data Warehousing - Redshift (AWS Cloud):

This step involves using Amazon Redshift as a data warehouse solution to store processed and structured data from the staging area (such as S3) for analytical querying and reporting.

Details:

- **Amazon Redshift:**
    - Redshift is a fully managed, petabyte-scale data warehouse service provided by AWS.
    - It is optimized for online analytical processing (OLAP) and designed to handle large-scale data sets for analytics.
- **Loading Data into Redshift:**
    - Processed and structured data from the staging area (e.g., S3) is loaded into Redshift for storage and analysis.
    - Various methods such as COPY commands or AWS Glue can be used for data ingestion into Redshift.
- **Columnar Storage and Compression:**
    - Redshift utilizes columnar storage and compression techniques to optimize query performance and reduce storage costs.
    - It efficiently stores and retrieves data by organizing data into columns instead of rows.
- **Querying and Analytics:**
    - Redshift allows complex SQL queries and analytics operations on large datasets, facilitating ad-hoc analysis and data exploration.

    - It supports business intelligence tools and applications for generating reports and visualizations.
- **Scalability and Concurrency:**
    - Redshift is scalable, allowing for the addition of nodes to handle increased workloads and concurrent queries.
    - It offers high performance even with multiple users querying the data simultaneously.

Importance:

- Redshift serves as a centralized repository for structured data, optimized for analytical querying and reporting purposes.
- Its capabilities in handling large datasets and performing complex analytics make it a valuable asset for deriving insights from data.

Transformation and Querying:

This step involves further transformation, data processing, and querying operations within the Redshift data warehouse environment to derive insights and facilitate advanced analytics.

Details:

- **Data Transformation:**
    - Additional transformations, aggregations, or data manipulations may occur within Redshift.
    - This step prepares the data for specific analytical queries or reporting requirements.
- **Complex Analytics and Reporting:**
    - Redshift allows the execution of complex SQL queries to perform various analytics tasks.
    - It supports the generation of reports, dashboards, and visualizations using business intelligence tools or SQL-based interfaces.
- **Data Aggregation and Joining:**
    - Aggregating data based on specific criteria or dimensions to extract summarized information.

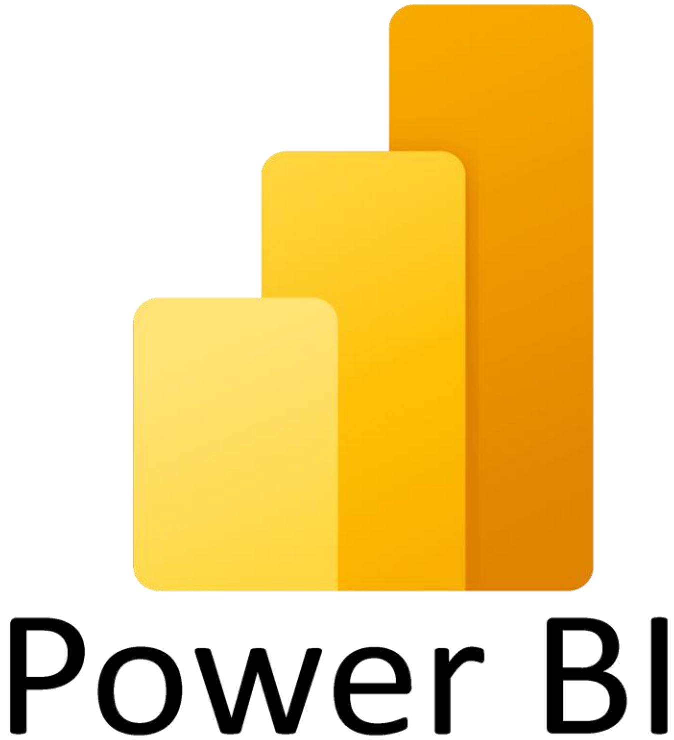
    - Joining multiple datasets within Redshift to combine relevant information for analysis.
- **Performance Optimization:**
    - Designing and optimizing queries to ensure efficient utilization of Redshift's capabilities.
    - Employing techniques such as indexing, query tuning, and workload management to enhance performance.
- **Data Enrichment:**
    - Augmenting or enriching data with additional information to enhance its value for analysis.
    - Combining data from different sources to provide comprehensive insights.

Importance:

- Transformation and querying operations in Redshift enable the extraction of valuable insights from the stored data.
- It supports advanced analytics, reporting, and visualization, empowering stakeholders with actionable information.

9. **Dashboard Creation - Power BI:**

In this step, Power BI is utilized to create interactive dashboards and reports by visualizing the data extracted and processed from Redshift, enabling stakeholders to gain insights and make informed decisions.



Details:

- **Power BI - Business Intelligence Tool:**
    - Power BI is a business analytics tool by Microsoft used for data visualization, exploration, and sharing insights through interactive dashboards and reports.
- **Data Connectivity:**
    - Power BI connects to Redshift as a data source to retrieve processed and analyzed data for visualization.
    - It supports various data connectors, allowing seamless integration with multiple data sources.
- **Dashboard and Report Creation:**
    - Users can create customized dashboards using drag-and-drop features, incorporating visualizations like charts, graphs, maps, and tables.
    - Reports can be designed to present key performance indicators (KPIs) and trends derived from the data.
- **Interactive Data Exploration:**

  - Power BI enables interactive exploration of data, allowing users to drill down into specific data points or apply filters for deeper insights.
  - Users can interact with visualizations to explore data relationships and patterns.
- **Collaboration and Sharing:**
  - Power BI facilitates collaboration by allowing users to share dashboards and reports securely with stakeholders.
  - It supports embedding dashboards into applications or websites for wider accessibility.

Importance:

- Power BI empowers users to visualize and interpret data from Redshift, transforming complex data into easy-to-understand visual representations.
- Interactive dashboards aid in decision-making by providing actionable insights to stakeholders.