Serverless Amazon ETL Pipelines

Author: Ahmed Nadeem

Email: ahmedprog2003@gmail.com

Date: March 4, 2025

Table of Contents

1. Pipelines

1.1. Task 1: News ETL Pipeline

Goal:

- To fetch technology news from a News API, validate the data, remove duplicate articles, and store the cleaned JSON data in an S3 bucket.

Key Services:

- **AWS Lambda (news_etl_lambda):** Implements the ETL logic to fetch, validate, deduplicate, and load the data.
- **Amazon EventBridge:** Schedules the Lambda function to run periodically (e.g., every 6 hours).
- **AWS Systems Manager Parameter Store:** Securely stores the News API key.
- **Amazon SNS:** Sends email alerts if any error occurs during processing.
- **Amazon S3 (news-etl-bucket):** The target bucket where cleaned JSON data (e.g., under raw/news_data.json) is stored.

1.2. Task 2: CSV ETL Pipeline

Goal:

- To process CSV files (e.g., daily city temperatures) uploaded to a staging area. The pipeline performs schema validation, handles missing values and duplicate records, and finally stores the cleaned CSV in a processed folder.

Key Services:

- **AWS Lambda (file_etl_lambda):** Processes the CSV files and applies transformation logic.
- **S3 Event Notifications:** Automatically triggers the Lambda function when a CSV file is uploaded.
- **Amazon SNS:** Sends email alerts on processing errors.
- **Amazon S3 (csv-etl-bucket):** Contains two logical folders: raw/ for incoming CSV files and processed/ for the cleaned output.

---

2. Infrastructure Setup

2.1. S3 Buckets

News ETL:

- **Bucket:** news-etl-bucket
- **Purpose:** Stores the cleaned JSON data fetched from the News API (e.g., at raw/news_data.json).

CSV ETL:

- **Bucket:** csv-etl-bucket
- **Folders:**
    - **raw/:** Where raw CSV files are uploaded (staging area).
    - **processed/:** Where cleaned and transformed CSV files are stored.

2.2. AWS Lambda Functions

news_etl_lambda:

- **Language:** Python (e.g., Python 3.12)
- **Functionality:**
    - Retrieves the News API key from the Parameter Store.
    - Calls the News API to fetch tech news.
    - Validates the data and removes duplicates based on article titles.
    - Uploads the cleaned JSON data to the news-etl-bucket.
    - Send SNS alerts if errors occur.

file_etl_lambda:

- **Language:** Python (e.g., Python 3.12)
- **Functionality:**
    - Triggered by an S3 event when a CSV file is uploaded to the raw/ folder.
    - Reads the CSV file and validates its schema (e.g., checks for columns like city, date, temperature).
    - Handles missing values, converts date fields, removes duplicate rows, and optionally sorts data.
    - Saves the transformed CSV to the processed/ folder in csv-etl-bucket.
    - Send SNS alerts if any error occurs.

2.3. Scheduling & Triggering

EventBridge:

- Schedules news_etl_lambda on a recurring basis (e.g., every 6 hours).

S3 Event Notifications:

- Configured to automatically trigger file_etl_lambda when a CSV file is uploaded to the raw/ folder of csv-etl-bucket.

2.4. IAM Roles & Permissions

news_etl_role:

- **Attached Policies:**
  - **AmazonS3FullAccess:** To enable reading and writing to S3.
  - **AmazonSNSFullAccess:** To allow sending SNS alerts.
  - **AmazonSSMReadOnlyAccess:** To retrieve parameters from the Parameter Store.
  - **CloudWatchLogsFullAccess:** To log events and errors.

file_etl_role:

- Similar permissions to news_etl_role (S3, SNS, CloudWatch Logs) to support the CSV ETL processing.

2.5. Additional Components

AWS Systems Manager Parameter Store:

- Stores sensitive configuration data like the News API key securely.

Amazon SNS:

- Configured to send email alerts for failures in both ETL pipelines.

---

3. Lambda Function Code

3.1. News ETL Lambda (news_etl_lambda)

Steps:

1. **Get API Key:** Fetches the News API key from the Parameter Store.
2. **Fetch News:** Calls the News API to retrieve technology news articles.
3. **Deduplicate:** Removes duplicate articles based on the title.
4. **Upload Data:** Writes the cleaned JSON data to S3 (e.g., news-etl-bucket/raw/news_data.json).
5. **Error Handling:** Sends an SNS alert if any error occurs.

Key Points:

- **Security:** Uses secure parameter retrieval.
- **Idempotency:** Deduplication logic ensures repeated processing does not produce duplicates.
- **Logging:** Uses CloudWatch for monitoring and debugging.

3.2. CSV ETL Lambda (file_etl_lambda)

Steps:

1. **Trigger:** Automatically invoked by S3 event when a CSV file is uploaded to the raw folder.
2. **Extract:** Reads the CSV file from the S3 bucket.
3. **Transform:**
    - **Schema Validation:** Checks for required columns (e.g., city, date, temperature).
    - **Missing Value Handling:** Drops rows with missing values in critical columns.
    - **Date Conversion:** Converts the date column to a proper datetime format, dropping rows with invalid dates.
    - **Deduplication:** Removes duplicate rows.
    - **Sorting:** Optionally sorts the data (e.g., by date).
4. **Load:** Writes the cleaned CSV to the processed folder in the S3 bucket.
5. **Error Handling:** Publishes an SNS alert if errors occur.

Key Points:

- **Extensive Data Validation:** Ensures data integrity before loading.

- **Organized Key Management:** Uses helper functions to ensure files are correctly placed in the processed/ folder.
- **Monitoring:** Logs events and errors via CloudWatch and SNS alerts.

---

4. Data Lineage & Architecture Diagrams

4.1. Data Lineage Diagrams

Task 1 (News ETL):

- **Flow:**
  - **News API → AWS Lambda (news_etl_lambda)**
    - Performs validation and deduplication
      → Target S3 Bucket (news-etl-bucket/raw/news_data.json)
  - (Optional) **SNS Alerts** on errors.

Task 2 (CSV ETL):

- **Flow:**
  - **Raw CSV Files** (S3: csv-etl-bucket/raw/) **→ AWS Lambda (file_etl_lambda)**
    - Performs schema validation, missing value handling, deduplication, and transformation
      → Processed CSV Files (S3: csv-etl-bucket/processed/)
  - (Optional) **SNS Alerts** on errors.

4.2. Data Architecture Drawings

Components to Include:

- **AWS Lambda** (for both news_etl_lambda and file_etl_lambda)
- **Amazon S3 Buckets** (with raw and processed folders)
- **Amazon EventBridge** (for scheduling news_etl_lambda)
- **Amazon SNS** (for failure alerts)
- **AWS Systems Manager Parameter Store** (for secure parameter storage)

Flow: Visually represent how data moves from ingestion (News API or raw CSV upload) through Lambda processing (validation, deduplication, transformation) to final storage in S3.

Deliverable: Create separate diagrams for Task 1 and Task 2 (e.g., news_architecture_ahmednadeem_ahmedprog2003@gmail.com.pdf and file_architecture_ahmednadeem_ahmedprog2003@gmail.com.pdf).

---

5. Failure Analysis Report

Topics Covered:

5.1. Concurrent Updates

- **S3 Versioning:**
  - Each S3 write creates a new version, ensuring that simultaneous writes do not overwrite existing data.
- **Idempotent Design:**
  - Deduplication and key-based file naming minimize duplicate processing.
- **Partition-Level Locking (Conceptual):**
  - Data partitioning (e.g., using date-based folders) segregates writes, reducing conflicts.

5.2. Data Corruption

- **Schema Validation:**
  - The pipelines check for required columns and validate data types before processing.
- **Quarantine Mechanism:**
  - Invalid or corrupt data is logged and, in a production scenario, could be moved to a quarantine folder for manual review.
- **Error Handling:**
  - Robust logging with CloudWatch and alerts via SNS ensure rapid detection and remediation of data issues.

---

6. Pytest Scripts & Test Results

Unit Tests:

- **For news_etl_lambda:**
  - Test API key retrieval from Parameter Store.
  - Test news fetching and deduplication logic.
  - Test error handling (e.g., simulate API failure and ensure SNS alerts are sent).
- **For file_etl_lambda:**
  - Test CSV reading and schema validation.
  - Test missing value handling and date conversion.
  - Test deduplication and sorting functionality.
  - Test error handling (e.g., simulate a missing CSV file).

Integration Tests:

- Use frameworks like **moto** to simulate AWS services if possible.
- Run tests in a staging environment and record the results.

Deliverables:

- test_news_etl_ahmednadeem_ahmedprog2003@gmail.com.py
- test_file_etl_ahmednadeem_ahmedprog2003@gmail.com.py
- Test results file (e.g., test_results_ahmednadeem_ahmedprog2003@gmail.com.txt or screenshot as PNG).

---

7. Loom Video

Content:

- **Infrastructure Setup:**
  - Walkthrough of the S3 buckets, EventBridge rule, SNS topic, and Parameter Store configuration.
- **Lambda Code Logic:**
  - Detailed explanation of both Lambda functions, including error handling and data processing logic.

- **Testing Process:**
    - Demonstrate manual testing (using test events), review CloudWatch logs, and show SNS alerts.

Deliverable: Record a Loom video (8–12 minutes). Save the video URL in a text file named loom_video_ahmednadeem_ahmedprog2003@gmail.com.txt.

---

8. Final Packaging & Submission

Files to Include:

- **Lambda Code Files:**
    - news_etl_lambda_ahmednadeem_ahmedprog2003@gmail.com.py
    - file_etl_lambda_ahmednadeem_ahmedprog2003@gmail.com.py
- **Test Scripts & Results:**
    - test_news_etl_ahmednadeem_ahmedprog2003@gmail.com.py
    - test_file_etl_ahmednadeem_ahmedprog2003@gmail.com.py
    - test_results_ahmednadeem_ahmedprog2003@gmail.com.txt or PNG screenshot
- **Diagrams:**
    - news_data_lineage_ahmednadeem_ahmedprog2003@gmail.com.pdf/png
    - file_data_lineage_ahmednadeem_ahmedprog2003@gmail.com.pdf/png
    - news_architecture_ahmednadeem_ahmedprog2003@gmail.com.pdf/png
    - file_architecture_ahmednadeem_ahmedprog2003@gmail.com.pdf/png
- **Failure Analysis Report:**
    - failure_analysis_ahmednadeem_ahmedprog2003@gmail.com.pdf/docx
- **Loom Video Link:**
    - loom_video_ahmednadeem_ahmedprog2003@gmail.com.txt

ZIP Naming Convention:

- Package everything into a single ZIP file named: AWS_lambda_ahmednadeem_ahmedprog2003@gmail.com.zip

Code Quality:

- Ensure that all Python code adheres to **PEP 8** style guidelines.
- Include comments and docstrings to explain code functionality.

Conclusion

We have successfully designed and implemented two serverless ETL pipelines on AWS:

- **Task 1:** The News ETL Pipeline fetches and processes news data from a News API, ensuring data integrity via validation and deduplication before storing it in S3.
- **Task 2:** The CSV ETL Pipeline processes uploaded CSV files by validating, cleaning, and deduplicating data before placing it in a processed folder in S3.