📄 Data Engineering HACKATHON – CASE STUDY DOCUMENT

Data Sources

1. **Yahoo Finance**
   Use the `yfinance` Python library to retrieve OHLCV (Open, High, Low, Close, Volume) data at a **minute-level interval** for S&P 500 symbols. The list of S&P 500 symbols can be obtained from this Wikipedia page.

2. **CoinMarketCap**
   Scrape the **"All Crypto"** table using any suitable Python scraping library (e.g., `BeautifulSoup`, `requests`, or `Selenium`). Limit the extraction to the **top 10 cryptocurrencies** by market cap.

3. **Open Exchange Rates**
   Create an account at Open Exchange Rates and obtain your **App ID**. Use the API key to fetch live foreign exchange data.

Task 1 – Data Acquisition

Implement a serverless ingestion system using AWS Lambda and Amazon EventBridge:

- Configure **AWS Lambda** functions to acquire data from each source (Yahoo Finance, CoinMarketCap, Open Exchange Rates).

- Use **Amazon EventBridge rules** to **trigger each Lambda function every minute**.

- Each Lambda function should fetch the data from its respective source and store it in an **S3 bucket**.

S3 Bucket Configuration

- Bucket naming convention: `data-hackathon-smit-yourname.`

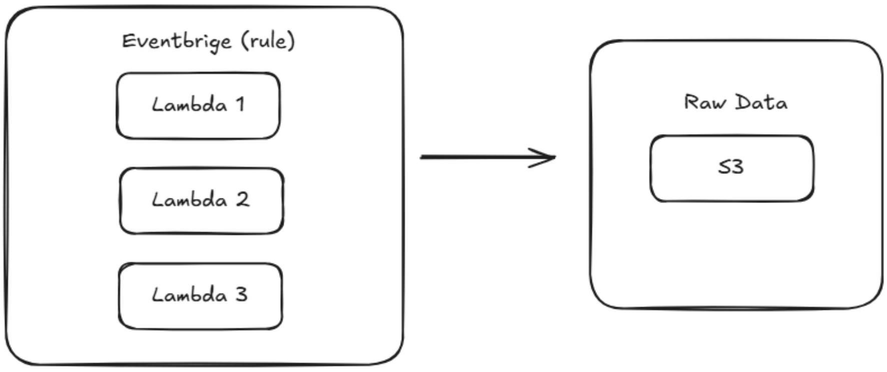
- All data should be stored under the `raw/` folder, organized by date.

- Each file should include a **contract or metadata** about the source, such as timestamp, source name, symbol, and response status.

Example

- `lambda_yahoofinance` runs every minute, fetches minute-level OHLCV data for all S&P 500 symbols, and stores it in:

  `s3://data-hackathon-smit-{yourname}/raw/yahoofinance/YYYY/MM/DD/{HHMM}.{file-format}`



Task 2 – Data Processing

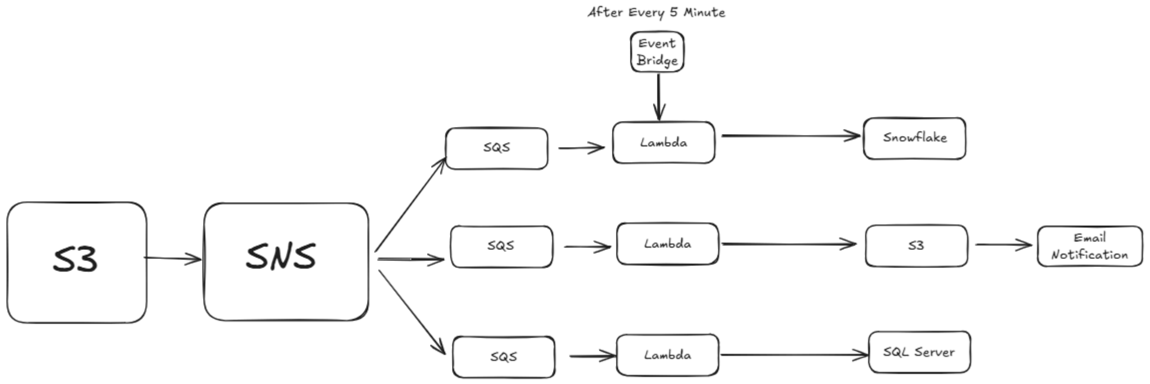
In this task, you will build a real-time data processing pipeline by connecting the S3 bucket (created in Task 1) with Amazon SNS, SQS FIFO queues, and AWS Lambda.

Step-by-Step Instructions:

1. **SNS Integration:**
   - Configure Amazon SNS to listen to new object creation events in your S3 bucket.
   - Use object metadata (contract) added during Task 1 to apply filters in SNS. Based on the source (e.g., Yahoo Finance, CoinMarketCap, Open Exchange Rates), route the event to the appropriate SQS FIFO queue.
2. **SQS FIFO Queues:**
   - Create three separate SQS FIFO queues, one for each data source:
     - `yahoo-finance-queue.fifo`
     - `coinmarketcap-queue.fifo`

        - `openexchangerates-queue.fifo`
    - As files arrive every minute, SNS filters and forwards relevant metadata to these queues.
    - For example, by the 5th minute, each queue should have received five messages corresponding to five processed files.

3. **Lambda Processing:**
    - Deploy three AWS Lambda functions, each specifically designed to handle data from a different source.
    - Use Amazon EventBridge to trigger each Lambda function every minute.
    - Each Lambda function reads messages from its respective SQS queue and processes the data accordingly:
        - Yahoo Finance → Snowflake: Parse the OHLCV data and insert it into a Snowflake data warehouse.
        - CoinMarketCap → S3: Transform and save the scraped crypto data to a processed S3 location.
        - Open Exchange Rates → SQL Server: Extract exchange rates and insert them into an SQL Server database.


After Every 5 Minute
Event Bridge
S3
SNS
SQS
Lambda
Snowflake
SQS
Lambda
S3
Email Notification
SQS
Lambda
SQL Server