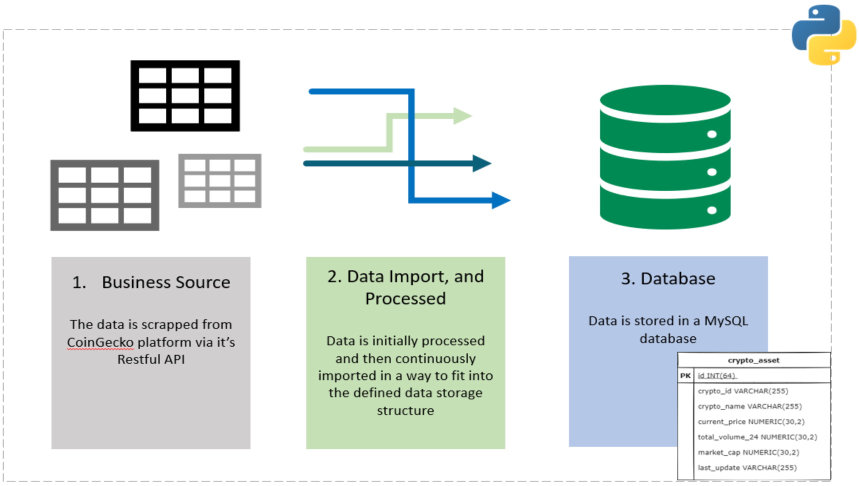
Building a Data pipeline with Batch Processing (Step 0)

I started off by considering the data pipeline as a product that brings data from important business source. This case, the data source is Coin Gecko, the world largest independent cryptocurrency data aggregator with over 6,000 different crypto assets tracked across more than 400 exchanges. The business problem is to collect daily prices, 24h volume, and market cap for all crypto assets listed by business source via a data pipeline built in python.



1. Business Source

The data is scrapped from CoinGecko platform via it's Restful API

2. Data Import, and Processed

Data is initially processed and then continuously imported in a way to fit into the defined data storage structure

3. Database

Data is stored in a MySQL database

crypto_asset	
PK	id INT(64)
	crypto_id VARCHAR(255)
	crypto_name VARCHAR(255)
	current_price NUMERIC(30,2)
	total_volume_24 NUMERIC(30,2)
	market_cap NUMERIC(30,2)
	last_update VARCHAR(255)

Extract data:

I spent some time understanding the Coin Gecko data source, particularly the various Crypto API endpoints. Explored the '/coins/markets' end point which list all supported coins prices, market cap, volume and market related data.

Finally, I wrote a python script that pulls data from Coin Gecko API v3. This script interacts with the API and pulls of the data.

Validate data:

I kept the data that have values and column that is expected and reject any that do not.

Transformation:

At this stage, I perform some slight data transformation, cleaning, and organization. While, checking for duplicate, missing values, and converting null values.

Loading:

Finally, the data extracted from Coin Gecko is loaded into to a MySQL database instance.

Conclusion:

Due to the time constraints, I couldn't incorporate most of the things I would love this pipeline to do. So, I decided to keep it simple. Here are somethings I was looking to incorporate into this project.

Stability and reliability: Develop a way to build a fault tolerant system that recover.

Designing an alert system: To ensure the accuracy of your business source, I believe an alert system that notifies for potential problems with the ETL process is essential.

Growth flexibility: To scale up and down according to the organization changing data needs. A cloud server processing and storage fees help save money and provide such flexibility.

Ensure Accurate logging. I was going to create a way for the data pipeline to create log of new information. But I couldn't get around that on time as had I only the weekend to work on this.

Future consideration:

If I had much more time to work on this project. I will set up workflow orchestration tool like Apache-Airflow. Apache-Airflow helps in scheduling, monitoring workflows and running pipeline.