Tiingo API Data Pipeline

Source	Raw	Staging	Warehousing	User

Raw
- Python script performs API call to Tiingo endpoints to gather information on all stocks in the S&P500.
- Utilizes scaped tickers to know which companies to pull
- Raw data is loaded into database hosted on AWS

Staging
- Reads from the raw_prices table.
- Selects relevant price and volume columns (date, open, high, low, close, volume).
- Keeps the original symbol identifier.
- Creates the stg_tiingo_api view.

Warehousing
- For dim_date: Reads distinct dates from stg_tiingo_api to populate the calendar dimension. Generates date_id.
- For fact_price: Reads prices and volume from stg_tiingo_api. Joins with dim_date and dim_symbol to link to dimension tables using date_id and symbol_id. Performs final type casting. Creates the fact_price table.

Tiingo_API_Extract_Load_Python.py

stg_tiingo_api.sql

dim_date.sql
dim_symbol.sql
fact_price.sql

Tiingo API

AWS MySQL
Shema: sql_project
Table(s): raw_prices

AWS MySQL
Shema: sql_project
Views: stg_tiingo_api

AWS MySQL
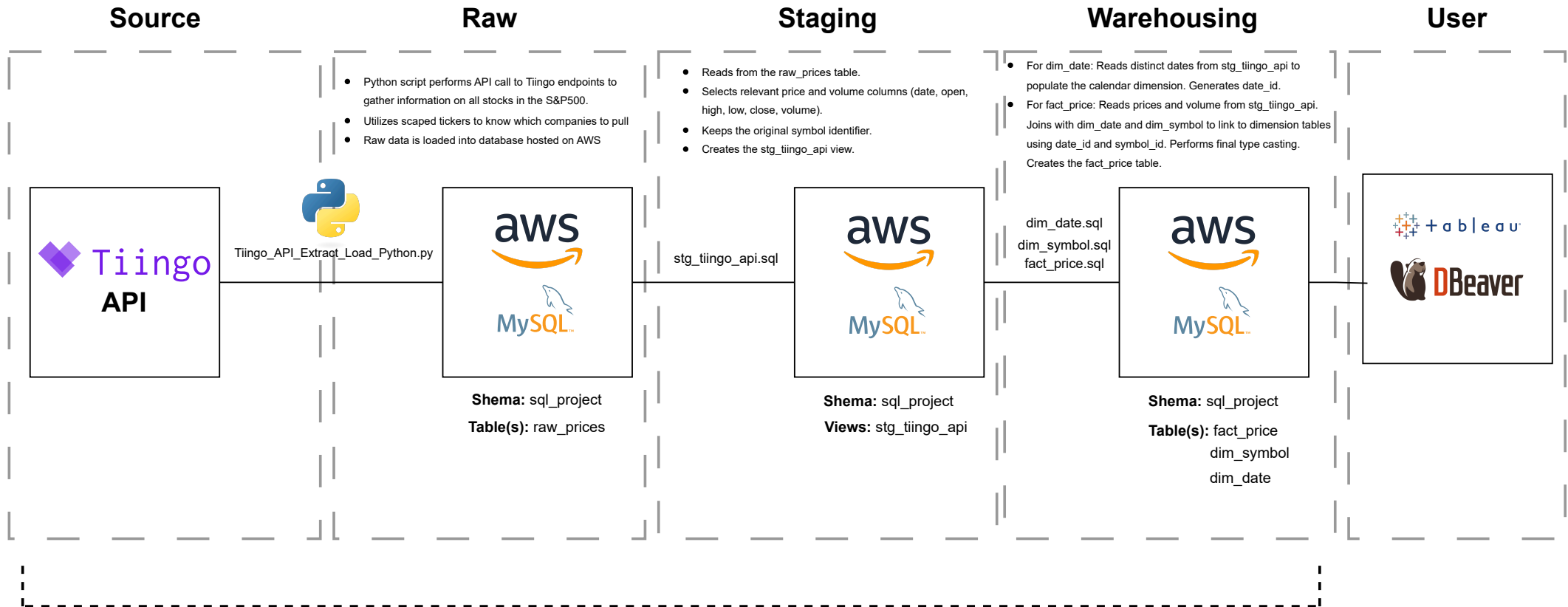**Shema:** sql_project
Table(s): fact_price
dim_symbol
dim_date

tableau / DBeaver

GitHub Actions

dbt_pipeline.yml

tiingo_api_pipeline.yml

wikipedia_web_scrape_pipeline.yml

- tiingo_api_pipeline.yml:
  - Runs on a schedule (or manually).
  - Executes Python script to fetch latest stock prices from Tiingo API.
  - Loads fetched price data into the raw_prices table in MySQL.
- wikipedia_web_scrape_pipeline.yml:
  - Runs on a schedule (or manually).
  - Executes Python script to scrape the S&P 500 company list from Wikipedia.
  - Loads scraped company data into the raw_wikipedia_sp500 table in MySQL.
- dbt_pipeline.yml:
  - Runs on a schedule (likely after data ingestion) or manually.
  - Executes dbt run command.
  - Transforms data from raw tables -> staging views -> warehouse tables within MySQL.