Kahoot!

Data K!rew

Data Engineer Challenge

Instructions

- ➔ Use whatever programming language/tool you are most comfortable with.
- ➔ The assessment is designed to get progressively harder. Do not feel you have to answer all the questions if you get stuck.
- ➔ Make it clear in your answers if you have made any assumptions about the data/problems.

What to return

- ➔ A GitHub, GitLab or some other online Git repository with:
  - ◆ All source code, relevant files or written answers to questions
  - ◆ Instructions for setting up and running your solution

The purpose of this challenge is to create a proof of concept application that can ingest streams from the public Coinbase market data feeds and output some insights. You do not need any prior knowledge about cryptocurrency or financial markets to solve this challenge.

The Kahoot! finance team has identified some key questions that if answered, would enable them to reduce their cryptocurrency risk.

Challenge

Part 1

Coinbase operates a cryptocurrency exchange platform that enables users to buy and sell cryptocurrency. The *Coinbase real-time market data feeds* provide streams of data covering the cryptocurrency exchange markets on the Coinbase platform.

One of the Coinbase data feeds provide *a snapshot of the current order book and all subsequent updates to the order book* for a given currency exchange, referred to as a *product* (e.g. Bitcoin for US dollars or Ethereum for Euros). The order book represents all available supply and demand for the product. The buyers represent the demand for the product or *bids* in trading terminology, and the sellers represent the supply of the product or *asks*.

Whenever someone bids for a quantity of the product at the same or higher price as someone who asks, a match will be made by the exchange and the trade goes through. Both the ask and bid will then be removed from the order book. So in a market with many transactions the order book changes all the time.

The order book at any given time for the product *Bitcoin for US dollars* can be represented with the following schema:

Product: BTC-USD		
Column	**Data type**	**Description**
side	string	Either “bid” or “ask”. If you are selling Bitcoin you are “asking”, and if you are buying Bitcoin you are “bidding”.
price_level	float	The price (in US dollars) buyers and sellers are willing to accept.
quantity	float	The amount of Bitcoins currently available at this price level.

Create a proof-of-concept for a tool that allows the user to specify a product and receive every five seconds an updated output of some metrics about these trades.

The specific insights the tool should provide in its output are:

- ➔ **What is the current highest bid and lowest ask, and at which quantities?**
- ➔ **What is the biggest difference between the highest bid and lowest ask we have seen so far?**
- ➔ **The mid-price is the average price of the highest bid and the lowest ask. I.e. the mid-point between those two prices. What is the average mid-price in the last 1, 5 and 15 minutes?**
- ➔ **What is the forecasted mid-price in 60 seconds?**
- ➔ **The forecasting error is the absolute difference between the predicted price and the observed price once it arrives. What is the average forecasting error on previous 60 second predictions in the last 1, 5 and 15 minutes?**

This is not intended to be a fully production ready application. We are not expecting any fancy GUI — command line is fine, or the ability to change the currency trade after the application is started. Also we are not expecting you to create your own forecasting algorithm, there are plenty of open source libraries that are good enough for this proof of concept.

Depending on how you solve the case, you might want to sign up with an account at Coinbase to get an API key with access to the feeds requiring authentication, but the case is also solvable without signing up for access. Coinbase provides several feeds that provide the data necessary to solve this case, some of them require authentication and some of them don't.

Part 2

Design a public cloud data architecture capable of handling the use case from part 1. The architecture should cover everything from data ingestion to end-users ingesting insights. It does not however, need to be able to provide real-time insights to the end-users, it only needs to be able to ingest the real-time data from the market data feeds.

You are expected to come up with a diagram of your architecture, along with a short summary of the choices you have made.

You are free to use any public cloud vendor to support your architecture.