Portfolio Generator/Rebalancer Application - Specification (WIP)

Requirement

Financial Markets are a great source to earn passive income at a rate of interest which is higher then traditional financial instruments like savings account, fixed deposit etc. However, financial markets are volatile and trends in these markets keep changing.

A sure short way to make money in financial markets is to be able to identify momentum in the initial stages itself and then go long on that momentum till a particular exist criteria is met. However, identifying these strategies, generating portfolio stocks, rebalancing them on a timely basis requires a lot of effort so this project aims to automate many of these steps

Solution

The aim of this project is twofold:

1. To identify, read and automate momentum based investing across asset classes - stocks, bonds, currencies, commodities. Implement these strategies and backtest, simulate the results and then using a subset of the initial list of strategies generate porfolio's to invest
2. The 2nd aspect of this project would be implementing a portfolio generator application or portfolio rebalancer application which generates a portfolio based on an underlying model provided by the first module, selecting which would be up to the end user. This module would provide a list of stocks with a weightage and tentative period till which to invest for any investment mode SIP or lumpsum

Overview & Implementation

This project will be implemented in multiple phases.

Phase 1:

1. Ingress - Data ingestion layer
    a. Daily Data
2. ETL - Data transformation layer
    a. open specification for price data defined and data transformed accordingly
3. Backtest Layer
    a. Reads data from db where ETL stores data and applies strategy on top of it to generate backtest results and store them in a DB. As part of backtest result it saves:
        i. Backtesting Period
        ii. Frequency of price
        iii. % Return
        iv. Maximum drawdown
        v. Number of profit making trades
        vi. Number of loss making trades
    b. This layer needs as input all the hyperparameters of a particular strategy
    c. Backtest layer will save its hyperparameters along with test results
4. Simulation Layer
    a. If backtest for a particular strategy is successful, based on certain thresholds then it would be simulated on data the strategy has not seen yet and the results of this test will be saved as simulation results
    b. Simulation layer will pick hyperparameters stored by backtest layer for a particular backtest and run the same strategy on the new dataset
5. Post Trade Monitoring Layer

This application runs via a web application in Phase 1

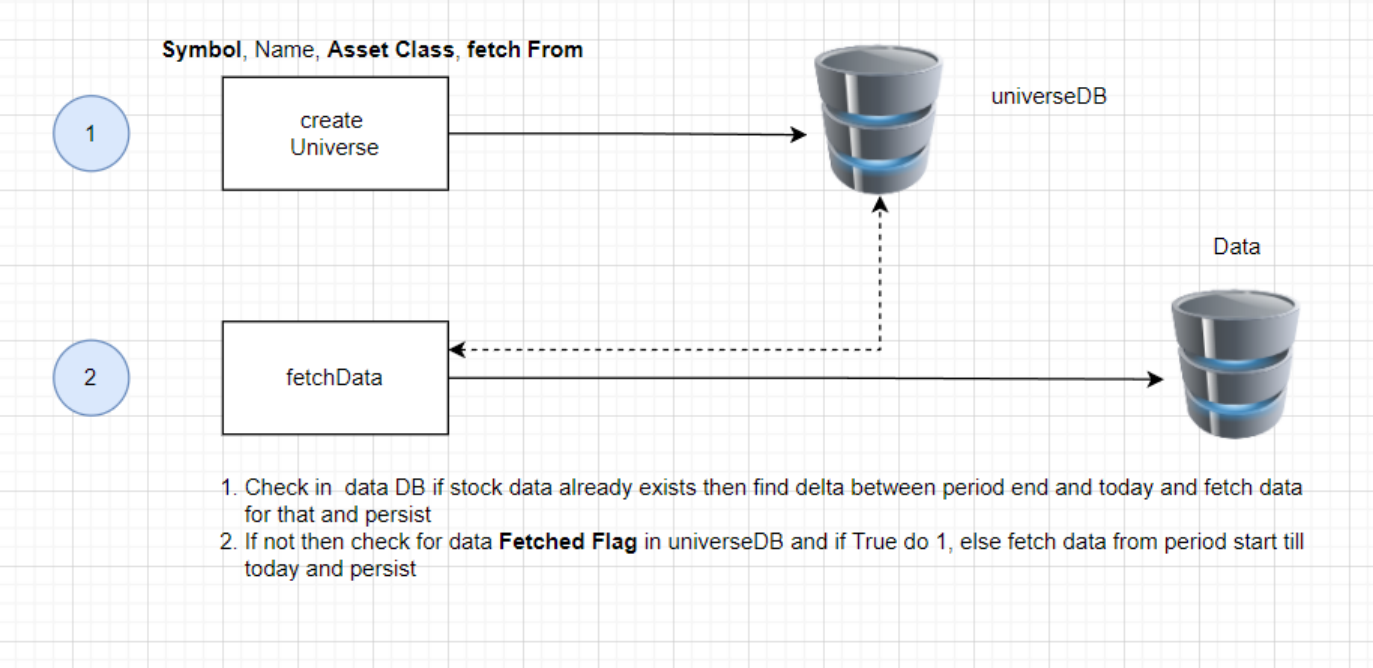
Phase 2:

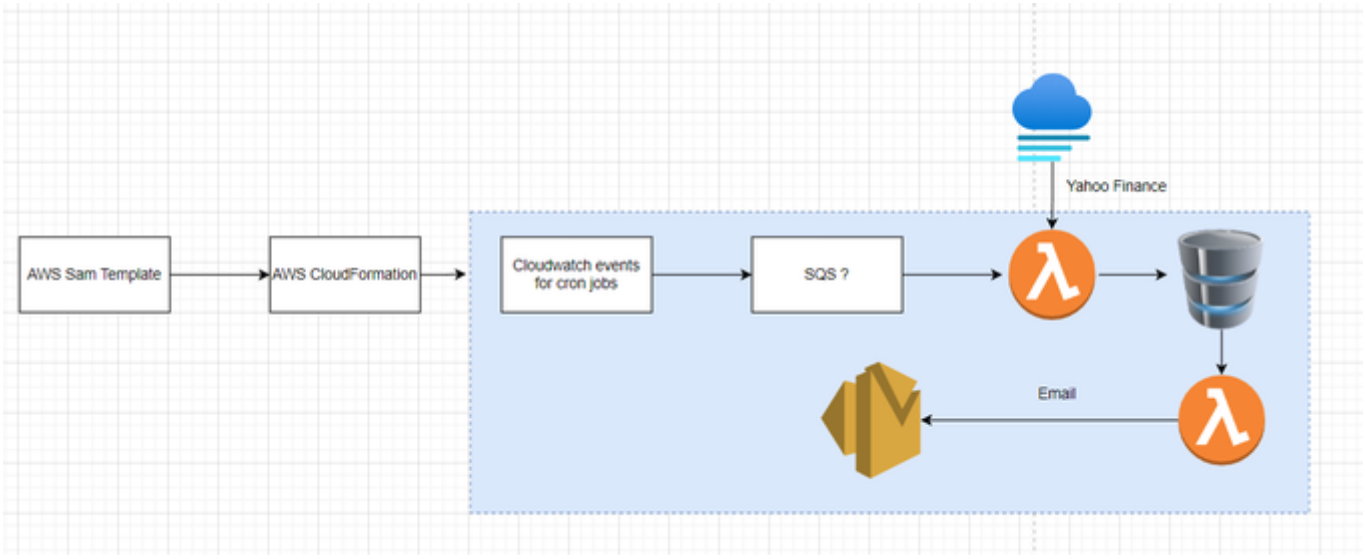
1. Ingress - Data ingestion layer
    a. Intraday Data
    b. Fundamental Data
    c. Qualitative Data

Implementation Flowchart

Database Architecture

Application Architecture





User Journey

1. Open web application

2. Select strategy, select stock universe, select option to ignore backtest and simulation results and view portfolio suggestion along with weightage
3. View backtest results, trigger strategy simulation on untested data
4. View results and generate portfolio allocation

UI Mock Up

User Stories

User Story	Priority	Effort (Hrs)
Setting up AWS, CICD pipeline, other infra pipeline 1. Search what CICD helps in and if its required and if so what is required to set it up 2. Setup CICD pipeline if required; **CICD is not required** 3. Check what is docker, what benefit it provides and if it is required 4. Setup docker if required 5. Converting Project files to lambda and testing them	1	6
Define DB Architecture and detailed product architecture	2	7
Layer to fetch daily data from yahoo finance and store it in S3 - Async API 1. Create async API which takes in hyperparameters to fetch data and store it in a DB a. This would only be required in 1 time b. This is done for daily and minute data c. Hyperparameters would be instrument list, start date, frequency 2. Create cron job which pulls data from DB, identifies delta between latest data stores in DB and today and fetches that data at both daily and minute frequency	1	7
Transformation Layer and addition of metadata and storage in DB - Async API	2	7
Strategy Layer Backtrader API (https://www.backtrader.com/)	1	8
Backtest Layer - Async API Backtrader API (https://www.backtrader.com/)	1	
Simulation Layer - Async API Backtrader API (https://www.backtrader.com/)	1	
Portfolio Generator - Async API	1	
UI Development	2	

APIs

Roadmap