Microservice

Design and Implementation of a High level Stock Analytics Terminal

Problem Solved

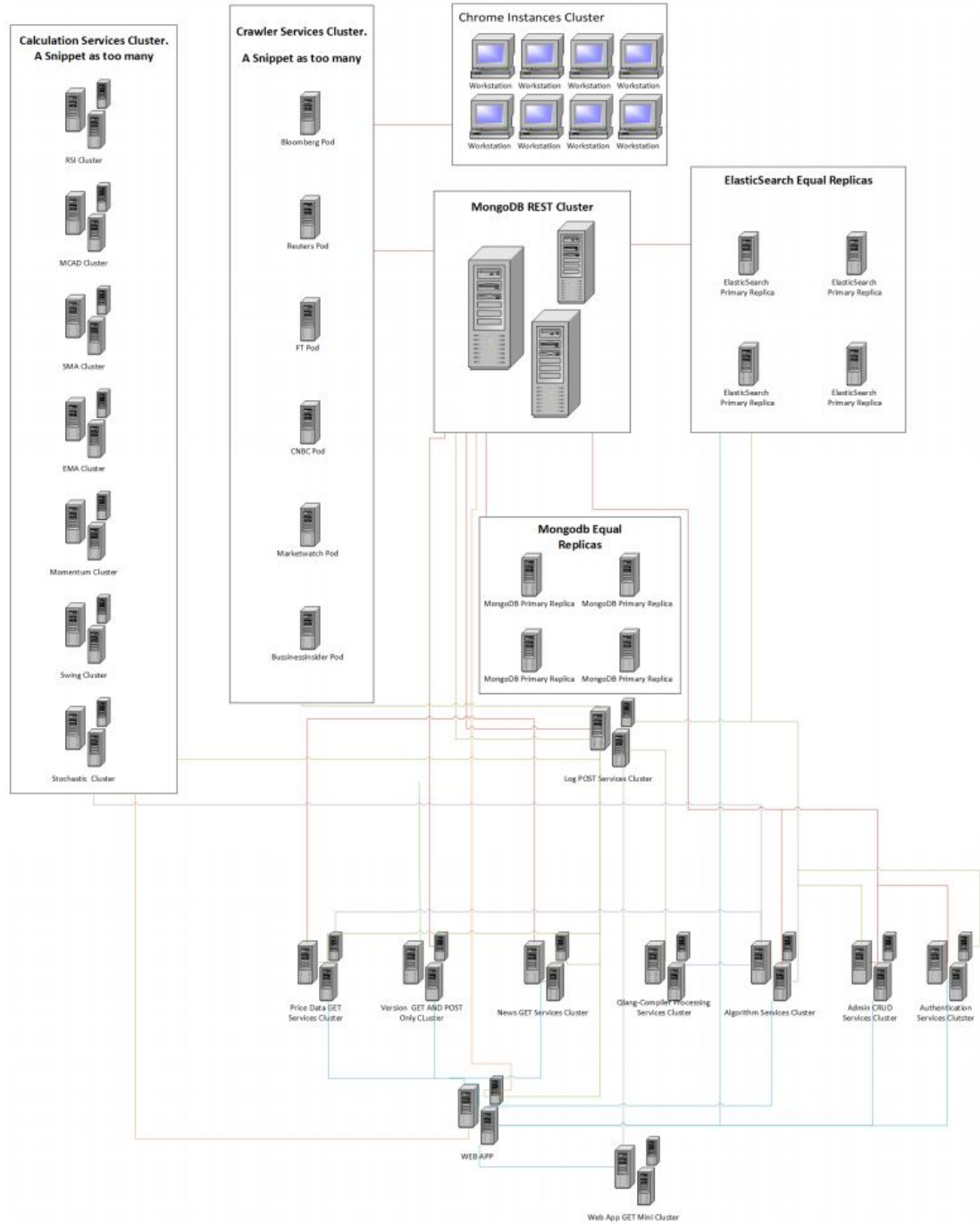
The lack of a Stock Trading Analytics backtesting terminal that allows analytics on realistic data, meaning tick data and fundamental data for the layman trader or analyst without coding acumen.

- Most platforms really are just toy implementations with daily data without fundamental data using python which demands coding experience and is unable to handle large sets of data. It would be like using drag and drop programming langauges to code large scale bussines applicatoins.

Microsevices/ APIs Created

1. Watchlist GET Service
2. Watchlist POST Service
3. Compiler Service
4. Fundamental Service
5. Price Service
6. Mongo Service
7. RSI Service
8. Bollinger Service
9. SMA Service
10. EMA Service
11. MACD Service

Architecture Design



Implemented Extra Features

- DNS server and Router
- Kubernetes - UI/Metrics Collection/ Auto Heal/ Auto Scale and Replication
- Docker Swarm
- Docker
- C++ + Boost Library/CMake
- Restheart/Mongo Microservice
- Websocket Server
- Programming Language Design
- Compiler in C/C++
- Regexe
- Angular 7/ SASS
- Big Data - 120 GB - high perfromance code in c++ for cleaning, db exporting, CSV importing
- Thomson Reuters Tick Data Crawling
- Linux Server Administration - Ubuntu Headless
- Windows HyperV VM Administration

Personal Reflection

- Personally Felt could have done better given more time
- Though i feel that i have exceeded my own expectations despite my fear of writing the compiler
- I proposed this project without knowing about compilers for big data or docker or kubernetes or dns servers
- Felt this subject was actually really exciting, it ignited my spark for devops and server administration as well as low level coding in c++ and AngularJS Development.
- I am most proud about 4 things. 120 Pages Report, Devops, Compiler Journey and Big Data in C++

Thank You Dr Chia