Product Price Analysis App

Christopher Calixte, Liam Daly, Tyler Martin, Alexis Nunez
GitHub: https://github.com/TM826651/csc468-01_project

Purpose

Our team will achieve this product through a CI/CD pipeline to handle the web scraping, data cleansing and saving to a database, and the presentation of the data through an interactive UI. Containers using a combination of Linux and Python will be utilized for web scraping and cleaning, while a lightweight Linux distribution node will handle the database management; Another node will handle the web server for the UI. By using CI/CD pipeline, we can automate the deployment of containers building the web scraping to scale and increase speed by each container performing its own data cleansing.

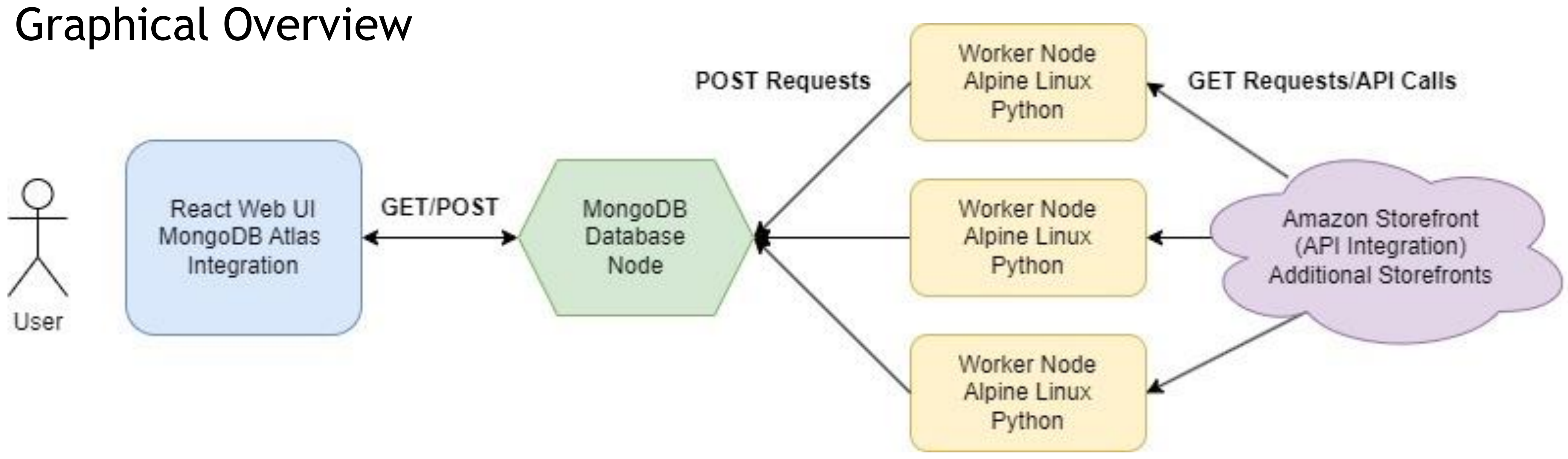
Stakeholder Value

- Save the user time
- Save the user money
- Gain insight into product price fluctuations to determine future trends

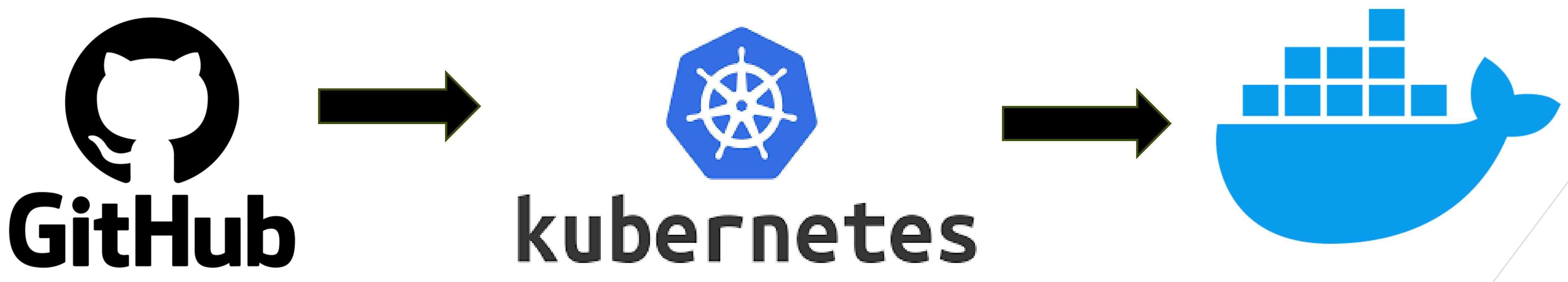
Architectural Design

- Alpine Linux webscraper node detects price change or pulls price at irregular intervals. This will be handled through either comparison or using Python sleep at randomized intervals to avoid bot detection by the host storefront.
- Data is pulled down to worker node. This will be performed using a Python script which utilizes a web scraping library.
- Data is parsed for price and time. Parsing will be done with a Python script to pull the price from the data, depending on the storefront.
- Worker node sends POST with cleaned information to MongoDB database server
- MongoDB writes data to database taking into account product name, product price, and time of capture.
- User opens web UI (webserver written in React)
- Webserver sends GET to database server to load pertinent data during query
- Data is loaded into UI with information for user. This frontend will provide all needed information with a visually pleasing UX, as well as a suggestion on the ideal time to purchase the product in question.

Graphical Overview



Deployment


GitHub
kubernetes