Backend Development Test: API Service for OnFinance AI

Objective

The goal is to assess the candidate's ability to deploy an open-source LLM efficiently using Google Cloud.

Backend API service that interacts with a deployed large language model (LLM) to process questions and return answers, specifically using a Retrieval-Augmented Generation (RAG) model for enhanced response accuracy. This service should demonstrate the candidate's skills in backend development, model deployment, and API creation.

Task Overview

Task: Develop an API Service utilizing LLMs and RAG to deploy a service that take question and provide answer.

Deploy an Open Source LLM using vLLM

- Deploy the Retrieval-Augmented Generation (RAG) model to ensure efficient and accurate inference. The deployment should support real-time API responses and be capable of leveraging the strengths of RAG for question-answering tasks.
- Deploy the model to provide real-time inference capabilities.
- Database should is already provided.
- The deployment must be done on a Google Cloud Platform (GCP) environment provided by OnFinance AI, utilizing resources effectively to handle the expected query load.

API Service Development

- Develop a backend API service that sends questions to the deployed RAG model and returns the model's answers.
- The API should be developed using Golang or Python, adhering to the candidate's expertise aligned with the job requirements.
- Setup Kafka and Google Kubernetes Engine (GKE) for High-Throughput Data Handling
- Configure a data processing pipeline capable of managing 10 concurrent incoming I/O streams, showcasing expertise in scalable system architecture.
- Handle HTTP POST requests, where the request body contains the question in a JSON format, e.g., **{"question": "Is 3M a capital-intensive business based on FY2022 data?"}**.
- Return the answer in a JSON response, e.g., **{"answer": "No, the company is managing its CAPEX and Fixed Assets pretty efficiently, which is evident from below key metrics: CAPEX/Revenue Ratio: 5.1% Fixed assets/Total Assets: 20% Return on Assets= 12.4%"}**.

Deployment and Testing

- Deploy the API service in a containerized environment using Kubernetes on the GCP environment provided by OnFinance AI. This demonstrates the candidate's competency in deploying and managing services in a cloud environment.
- The service must be accessible for live testing via a public endpoint.

Submission Requirements

- **Deployed Service on GCP**: The candidate must deploy the service on the GCP environment provided by OnFinance AI and share the public API URL for live API testing.
- **Code Repository**: Submit the source code via a GitHub repository link, including a README with comprehensive setup and local running instructions. The README should document any development assumptions, as well as detailed steps for deploying the service using Kubernetes on GCP.

Evaluation Criteria

- **Functionality**: The API service accurately processes questions and returns answers using the RAG model, showcasing the model's capabilities.
- **Code Quality**: Source code is well-structured, readable, and adheres to backend development best practices.
- **Cloud Deployment**: Effective containerization and deployment of the service using Kubernetes on the GCP environment provided by OnFinance AI, with clear and reproducible deployment documentation.
- **Model Deployment**: Is able to deploy LLMs in a production environment.
- **Performance**: The service exhibits efficient request handling and minimal latency, ensuring a responsive user experience. **Focus primarily on response speed. and expect you to stream responses for faster response speed. And handle multiple queries parallel.**

Candidates are requested to submit their live service URL and code repository to team@onfinance.in.