Wasserstoff Backend Interview Task: Intelligent Load Balancer for Multi-API Management

Task Overview

You are tasked with creating a load balancer that intelligently manages and distributes traffic across multiple unique APIs. The primary goal is to showcase your logical, reasoning, and analytical skills by designing and implementing a sophisticated routing mechanism that efficiently handles various types of requests. This task is divided into two milestones:

1. **Milestone 1: Design and Functionality**
2. **Milestone 2: Queue Management and Analysis**

We are not just looking for a complete solution but also for creativity, thought process, and innovative problem-solving.

Milestone 1: Design and Functionality

Objective

Develop a load balancer that can dynamically route incoming requests to different API endpoints. This should include a combination of simple and complex routing logic to demonstrate your understanding of load balancing concepts.

Requirements

1. **Dynamic Routing:**
    - Implement routing based on:
        - API type (e.g., REST, GraphQL, gRPC)
        - Randomized routing to simulate server load balancing
        - Custom criteria defined by you
2. **Function Simulation:**
    - Create multiple mock API endpoints with varied response times and behaviors.
    - Implement functions that simulate slow and fast responses.
3. **Logging and Metrics:**
    - Capture and log metrics such as request times, endpoint selection, and response times.
    - Provide detailed logging for analysis.
4. **Port Management:**
    - Run the load balancer and mock APIs on different ports to simulate a real-world environment.
    - Ensure clear instructions on how to run the load balancer on specified ports.

Deliverables

1. **Source Code:** Well-documented code with clear inline comments.
2. **Configuration Files:** Any necessary configuration files.
3. **Documentation:** Detailed documentation explaining your design choices, setup, and usage instructions.

Evaluation Criteria

1. **Functionality:** Correct and efficient implementation of dynamic routing. 2. **Creativity:** Innovative approaches to routing and simulation of API responses. 3. **Code Quality:** Well-structured and documented code.

Milestone 2: Queue Management and Analysis

Objective

Extend the load balancer to handle incoming requests using different queuing strategies. Demonstrate how you can manage and analyze request queues to optimize performance.

Requirements

1. **Queue Management:**
   ○ Implement different types of queues (e.g., FIFO, priority-based, round-robin). ○ Demonstrate how requests are distributed to these queues based on defined criteria.
2. **Request Handling:**
   ○ Develop a mechanism to handle and process requests from these queues. ○ Show how different queuing strategies affect performance and load distribution. 3. **Logging and Analysis:**
   ○ Provide detailed metrics for each queuing strategy.
   ○ Analyze the performance of each strategy and document your findings. ○ Console log the load distribution and analyze which server picks up routes based on a weightage algorithm that you design.

Deliverables

1. **Source Code:** Well-documented code with clear inline comments.
2. **Configuration Files:** Any necessary configuration files.
3. **Documentation:** Detailed documentation explaining your design choices, setup, usage instructions, and analysis of different queuing strategies.
4. **Video Screengrab:**
   ○ A screencast video explaining your code and demonstrating the running application.
   ○ Your face should be visible in a corner of the video to ensure authenticity.

    - Clearly explain the results and any challenges faced.

Bonus Points

- **Test Scenarios:** Create and document test scenarios that showcase the performance and reliability of your load balancer.
- **Additional Features:** Implement any additional features that enhance the functionality or performance of the load balancer.
- **Innovative Thoughts:** Demonstrate any creative approaches or solutions beyond the basic requirements.

Submission Guidelines

1. **GitHub Repository:** Create a GitHub repository and push your code, configuration files, and documentation.
    - Submit the Postman Collection Link, deploy URL, and GitHub Link to confirm your submission.
    - Name the repository as follows: *{CandidateName}*`/wasserstoff/BackendTask`.
    - Host your project publicly and provide a link to a Vercel or another publicly hosted version of your project.
2. **Video Screengrab:** Upload the video screengrab to a platform of your choice (e.g., YouTube, Vimeo) and provide the link in the GitHub repository.
3. **Submission Deadline:** Ensure your submission is completed within 96 hours from receiving this task.

Instructions for Candidates

- Attempt the task to the best of your skill and knowledge.
- Focus on showcasing your thought process, problem-solving abilities, and commitment.
- Completing the task is less important than demonstrating your approach and creativity.

Evaluation Criteria

1. **Design and Implementation:** Quality and efficiency of your design and implementation.
2. **Creativity and Innovation:** Your ability to think outside the box and provide innovative solutions.
3. **Analytical Skills:** Depth of your analysis and understanding of the problem. 4. **Presentation:** Clarity and thoroughness of your documentation and video presentation.

Tech Stack

We prioritize candidates with experience in Node.js or the MERN stack (MongoDB, Express.js, React.js, Node.js). However, strong coding abilities in other languages will still be considered.

Here are a few key points that we will consider in addition to technical skills:

- **Understanding of the Task:** Demonstrated comprehension of the project requirements and the ability to articulate your approach to solving problems.
- **Technical Skills:** Proficiency in Node.js or the MERN stack is preferred. However, strong coding abilities in other languages will still be considered.
- **Additional Skills:** Any additional relevant skills or experiences that enhance your application, even if not specifically mentioned in the job description.
- **Problem-Solving Abilities:** Your ability to think critically and solve complex problems effectively.
- **Communication and Collaboration:** Strong communication skills and the ability to work well in a team, especially in a remote environment.
- **Project Experience:** Prior experience with relevant projects, whether personal, academic, or professional.