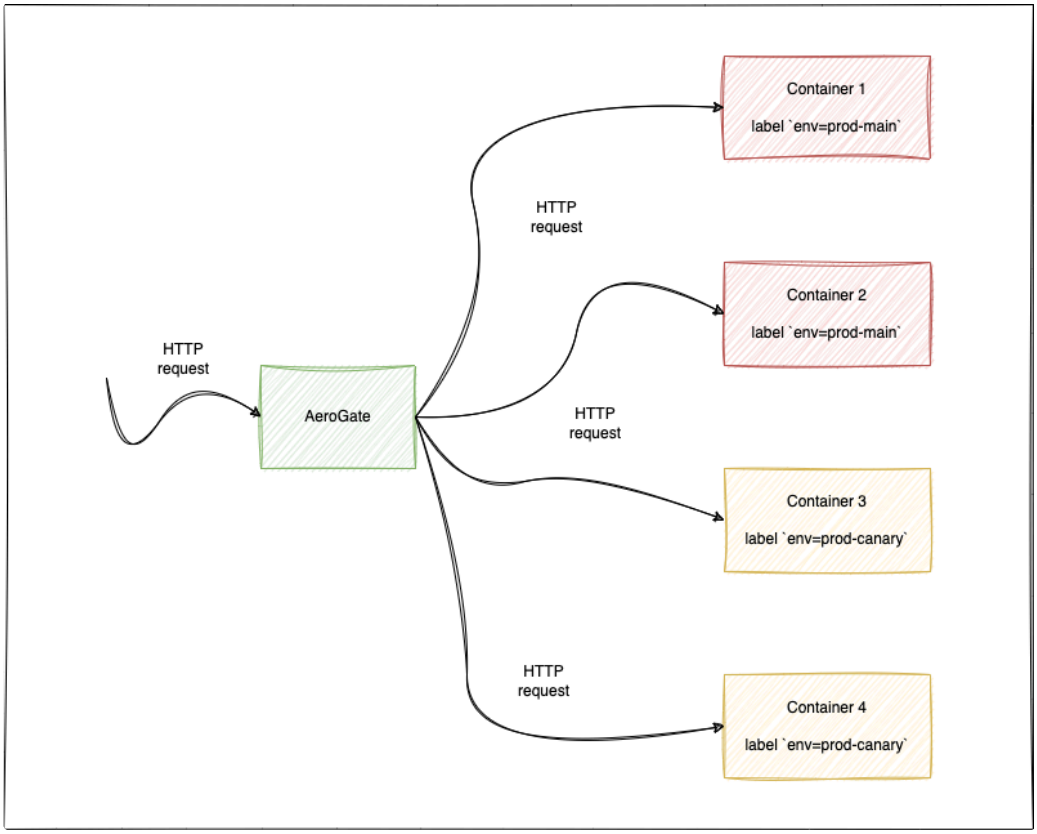
Interview Assignment



Problem Statement

Develop an application named `AeroGate` to run as a Docker container, actively listening on port `8080`, and handling HTTP requests directed to the `/cluster-info` endpoint.

Upon receiving requests, `AeroGate` will efficiently route them to backend services, also deployed as Docker containers. This routing is determined by a specific label, `'env'`, attached to the backend Docker containers. Each backend service Docker container must be labeled appropriately with a distinct environment identifier, such as `env=prod-canary`.

The routing configuration is managed through the environment variable `ROUTE_LABEL` within the `AeroGate` container. For example, setting `ROUTE_LABEL` to `'prod-canary'` will instruct `AeroGate` to direct requests exclusively to backend containers labeled with `env=prod-canary`.

Guidelines:

- Backend service containers can run any HTTP application. For simplicity, they can host an Nginx server with a custom webpage, allowing differentiation based on `curl` output.
- The backend HTTP application can listen on any port as long as requests are correctly routed.
- You're free to choose any programming language for developing `AeroGate`.
- `AeroGate` must be containerized and run as a Docker container alongside the backend services.
- The entire stack, including `AeroGate` and backend services, should be deployable using `docker-compose`.