UM-D CIS-578 Challenge 2

Nate Pierce

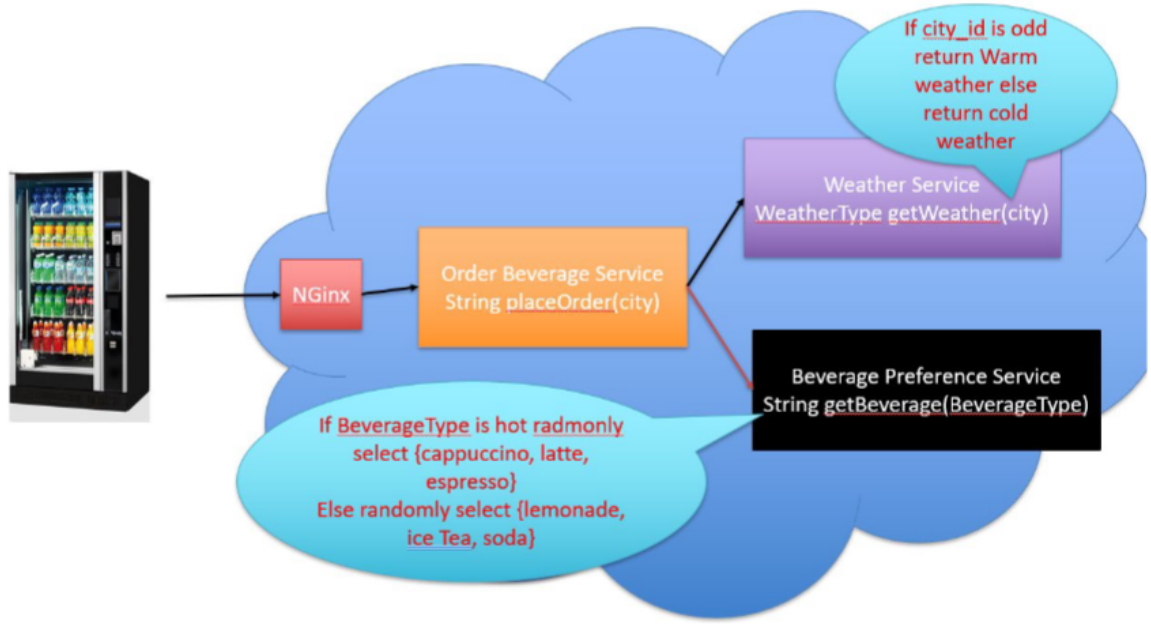
Overview

This challenge implements different services through Apache Thrift. The services are: order beverage, beverage preference, and weather. The client machine interfacing with the user/customer is a “vending machine” abstraction written in C++. An nginx reverse proxy handles communication between the vending machine and the order beverage service. From here, an RPC determines the weather type based on the unique city ID. The callee in this case is on the weather service server. Then, also in order beverage, another RPC is invoked to the beverage preference service. The weather type is used to randomly select a type of beverage, based on whether the weather is hot or cold.

This challenge required creating a brand new service (beverage preference service) following the model. Then, the getBeverage function was implemented, and the placeOrder function updated to make use of this new service and function.

The project was completed on a single node ubuntu instance hosted on cloudlabs.

The demo/script/generate_request.sh script doesn’t appear to output anything near random results. Inspection of the results show almost no variation. However, with a server running local, you can run curl -d "city_id="$i http://127.0.0.1:8080/ and get more satisfying results by running the command slowly. I suspect that this is because of how often the srand(time(NULL)) function I used updates the random seed -- not frequently enough to allow the results from the script to appear actually random for each line of the bash script.



Github Repo

[Jormogundr/vending-machine]

Docker Image

[nathancp93/vending-machine-microservices]

Output

See the output.txt in either the github repo or the attached tar.gz

Screenshot