Case Study - A/B Test Infrastructure

For the position "Cloud Engineer" (R0001891)

Hello and welcome to your case study! We are curious to see your brain in action and have prepared some tasks for you to solve. Our hiring team is looking forward to your results!

Purpose

This challenge is designed to give you a glimpse of the activities you would be tackling in your potential future team.

The tasks in this case study are inspired by and very similar to those you need to deal with every day. This means that if you have fun solving this case study you are most likely on the right track!

As you will notice, not every single detail necessary to come up with a decision is included in the tasks. We are mostly interested in understanding your process of thought, so make your own assumptions when you feel it is necessary and do not forget to write them down in your answer. We're looking forward to getting your response and hope you can derive some learnings and joy from doing this assignment.

Deadline for Submission

You should not need more than 3-4 hours to solve this case study. Please make sure that you submit your results within 7 business days after receiving. We would like to ask you to also grand us the same time to evaluate your results after submission.

Delivery Format

Please submit your results via the link provided in the email from your recruiter. There you can upload your files in a zip archive.

The Challenge

Introduction and Background Information

As a Cloud Engineer in our Hotel Search backend team, you will work on our cloud-based infrastructure serving millions of requests every day.

This does not mean we are looking for someone whose aim is to avoid downtime at all cost only, but also to bring up their ideas how to shape our architecture in the future and who is motivated to support our application engineers implementing new features and services.

At trivago, we have a culture of better trying and failing than never trying something new. This means we're also doing a lot of testing. In the sense of determining the best solution for our product Split or A/B testing is an essential piece of this.

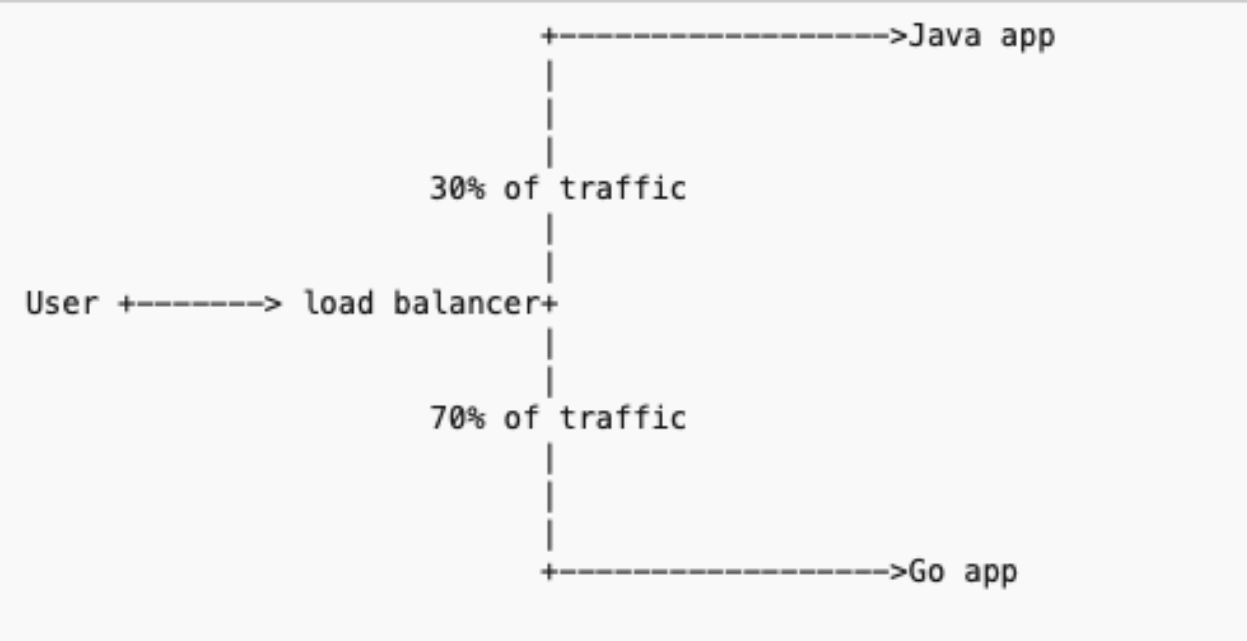
This assignment has been designed to give you a glimpse of some of the basic challenges you will face in this role. There is no perfect answer; we are interested in how you process your ideas and structure your response.

You will find two binaries, one Go based application, one Java-based application. Both are providing an HTTP service with the same endpoints:

Route	Description
/	A static site. Should **not** appear in the final setup as it is but redirect to **/hotels**.
/hotels	JSON object containing hotel search results
/health	Exposes the health status of the application
/ready	Readiness probe
/metrics	Exposes metrics of the application

Your Task

Your challenge will be to provide a load balancer setup like the following:



The traffic distribution should be as follows: 70% of the requests are going to the application written in Go, 30% of the requests are going to the application written in Java.

The setup has to be containerized and available as Docker Compose File or Kubernetes Manifests (whichever you're feeling more comfortable with).

Please also add a README file where you explain your chosen architecture (plain text or Markdown). It also provides you the space to add anything you think we should also know about the setup.

Ensure that the final zip archive contains all the resources we need to build and start the setup **locally**. We hope you have fun with this, thank you and Good luck.

Additional information:

- The applications were built using Go 1.12 (provided as Linux binary) and Java 11
- Both applications are binding to all interfaces on port 8080
- You don't have to upload the built Docker images to a registry, we will build them locally