SOLUTION DOCUMENT

BASIC CHARACTERISTIC

- Microservice based solution using GoLang
- RPC as the primary messaging pattern
- Consul is a service mesh solution
- ElasticSearch as a Database
- GoMicro as the basic framework for the microservice architecture

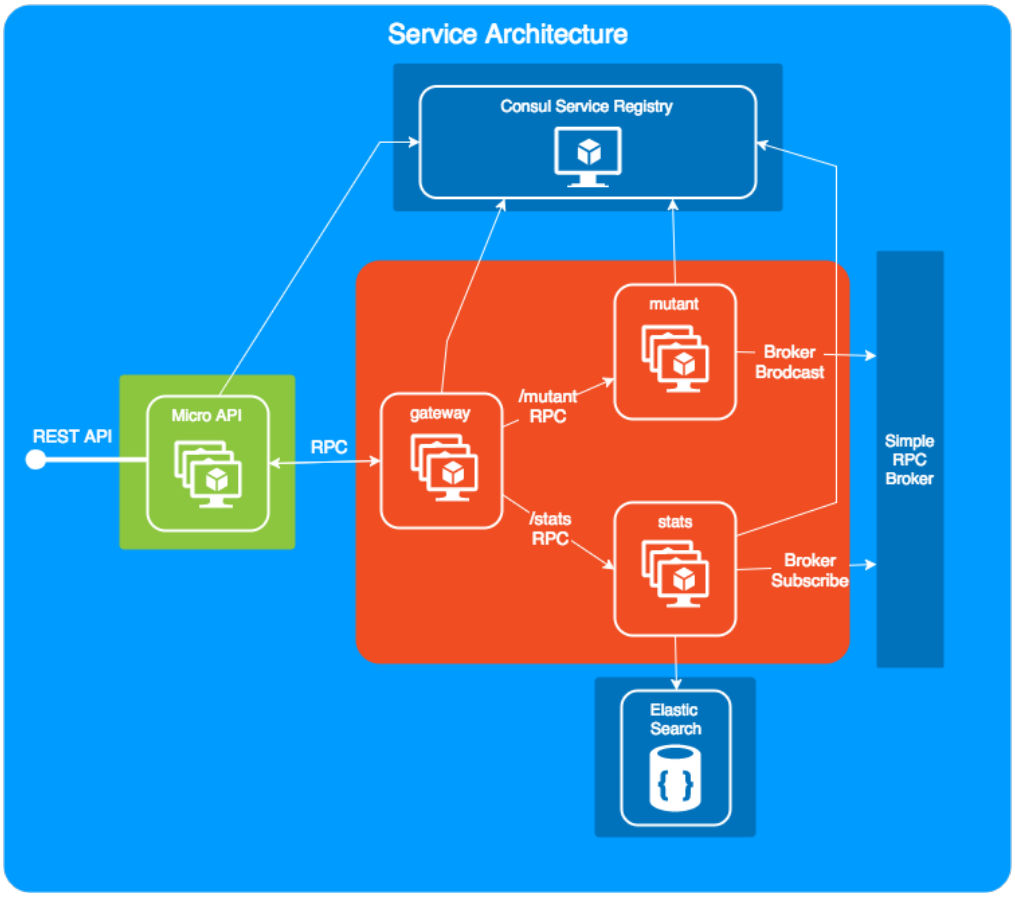
WHY GOLANG? SOURCE

Microservices are supported by just about all languages, after all, microservices are a concept rather than a specific framework or tool. That being said, some languages are better suited and, or have better support for microservices than others. One language with great support is Golang.

Golang is very light-weight, very fast, and has a fantastic support for concurrency, which is a powerful capability when running across several machines and cores.

Go also contains a very powerful standard library for writing web services.

ARCHITECTURE



GATEWAY MICROSERVICE

Gateway microservice will be in charge of registering the APIs into Micro API and forward the requests. see https://github.com/micro/micro/tree/master/api

MUTANT MICROSERVICE

This microservice will contain the DNA Analysis algorithm. It will analyze the DNA data and produce a result. The result is published in the Message broker.

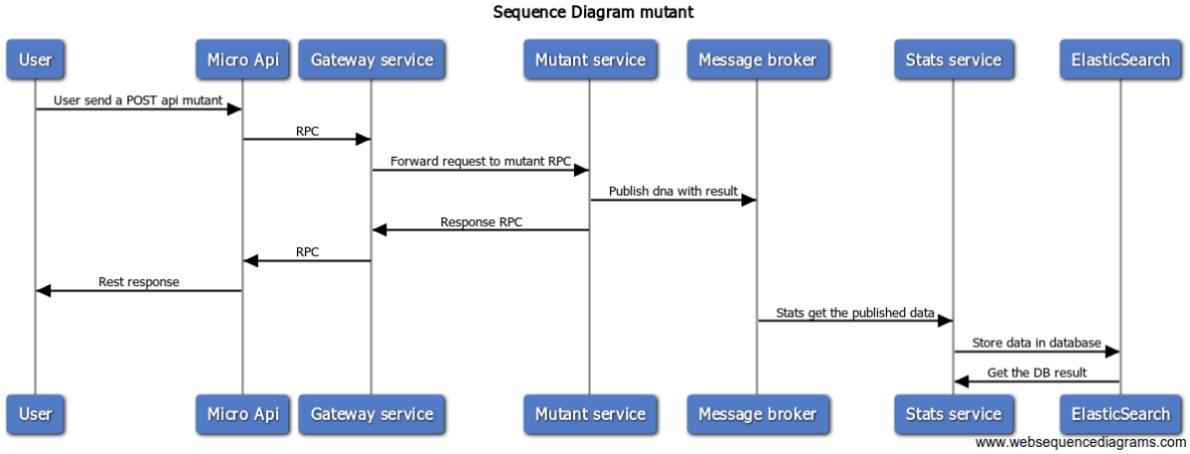
STATS MICROSERVICE

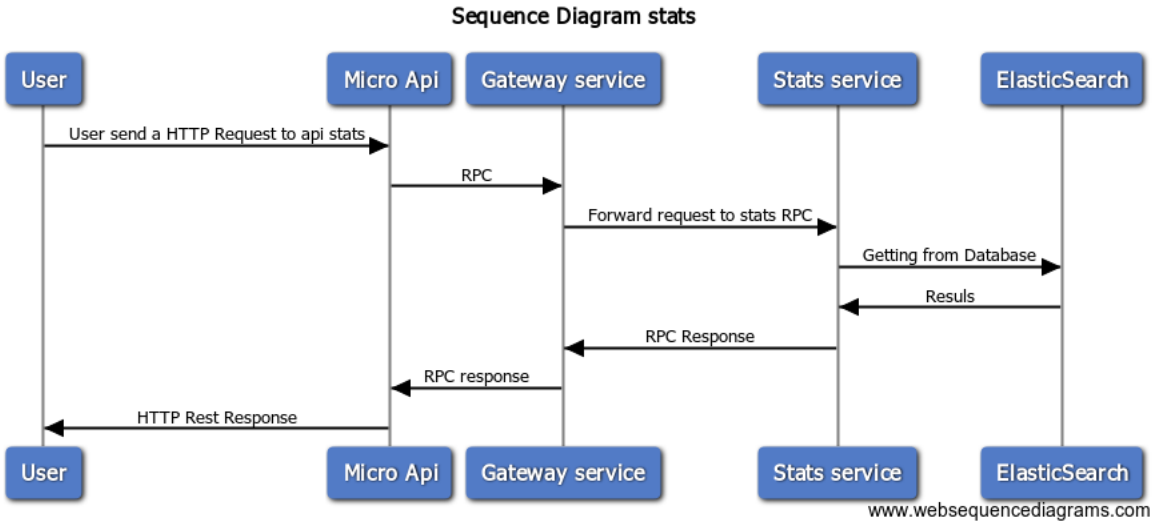
This microservice has two main objectives:

- To subscribe to the mutant results message broker to store it in the DB
- To Return the stats results

SEQUENCE DIAGRAMS

We have two main sequence diagram, one for each service





SOURCE CODE

The source code was organized in two repositories:

https://github.com/rodrigodmd/ml-mutant contains the algorithm to analyze the DNA data. This resources will be consumes as a library from the mutant microservice.

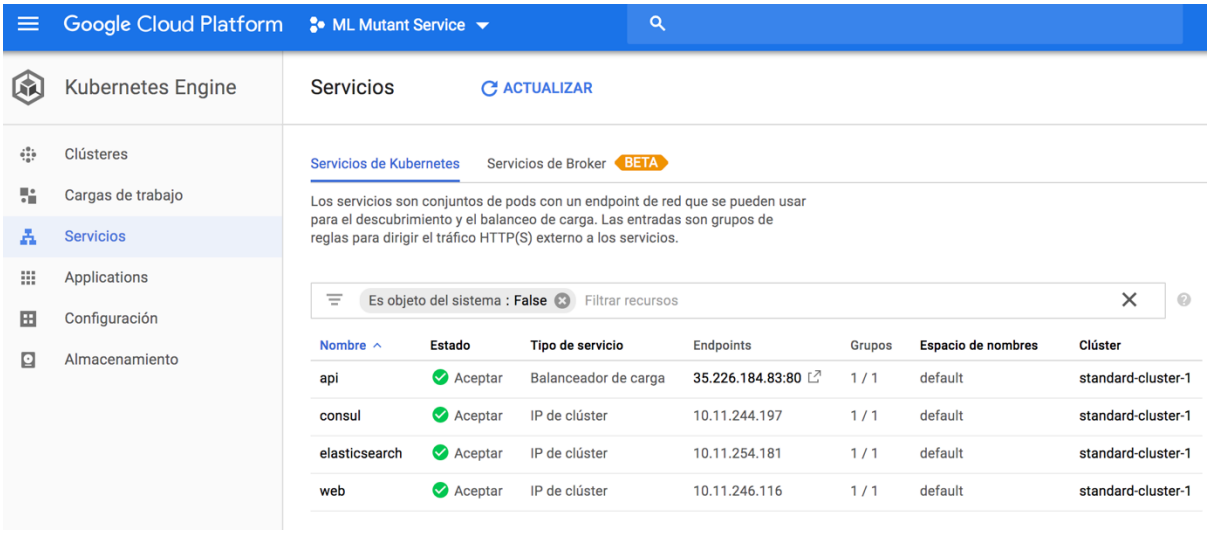
https://github.com/rodrigodmd/ml-mutant-srv Contains:

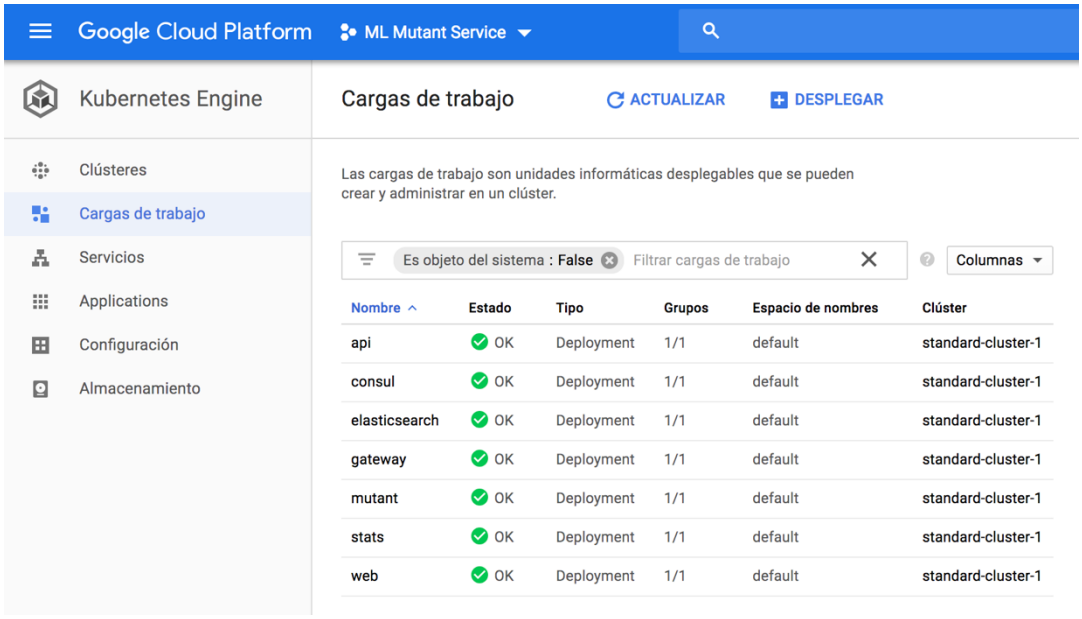
- the source code for the microservices. Will consume ml-mutant library as a dependency.
- docker image scripts (build, run and publish)
- Deployment scripts

DEPLOYED ENVIRONMENT

Deployment done in Google Kubernetes and configured to auto scale depending on the CPU usage

Base API url: http://35.226.184.83/api





EXAMPLES

MUTANT DNA

```
curl --header "Content-Type: application/json" \
  --request POST \
  --data '{ "dna": ["ATGCAA","CAGTTT","TTATTT","AGAAGG","CCACTA","TCACTG"] }}' \
          http://35.226.184.83/api/mutant
```

HUMAN DNA

```
curl --header "Content-Type: application/json" \
  --request POST \
  --data '{ "dna": ["ATGC", "CAGT", "TATT", "AGAA"] }}' \
          http://35.226.184.83/api/mutant
```

INVALID DNA STRUCTURE

```
curl --header "Content-Type: application/json" \
  --request POST \
  --data '{ "dna": ["ATGCAA","invalid","TTATTT","AGAAGG","CCACTA","TCACTG"] }}' \
          http://35.226.184.83/api/mutant
```

GET STATS

```
curl --header "Content-Type: application/json" \
  --request GET \
          http://35.226.184.83/api/stats
```