Faculty of Engineering and Applied Science

SOFE 4790U: Distributed Computing

Fall 2022

Dr. Mohamed El-darieby
TA: Ghadeer Abdelkader

Lab 2: Deploying a request splitting ambassador and a load balancer with Kubernetes

Lab Group 20

Adris Azimi (100704571),

Deepak Thangella (100695929),

Harris Athwal (100750935),

Krystian Rusin (100704636)

Discussion.

Q. **Summarize the problem, the solution, and the requirements for the pattern given in part 1. Which of these requirements can be achieved by the procedures shown in parts 2 and 3?**

Problem

The problem is that whenever a client interacts with and consumes multiple services, the client must have to be updated whenever services are added or removed. This is a very strenuous task as more time and resources are required to update the client. There are three main situations in which client updates would be required whenever a change to the service is made: multiple disparate services, multiple service instances, and multiple service versions.

Multiple disparate services

If an API that the client interacts with changes, the user must be updated to those changes as well. If a service is refactored into multiple separate services, both the server and client code has to be updated as well.

Multiple service instances

Through running multiple instances of the service, every time the instances are required to meet the demand of consumption through load balancing, the client has to be updated.

Multiple service versions

Every time new versions of a service are deployed along with their already existing previous versions, the client has to be updated because of the changes to the percentage of traffic that is being routed to the new version and existing endpoint.

Solution

A possible solution to this problem is to place gateways in front of the services and then use application layer 7 routing to route the request to the application instances. With this solution, the client only needs to know and communicate with a single endpoint, which is the gateway that acts as a medium and allows changes to be made to the services. Therefore updates to the client are not necessary due to this.

Requirements

The requirements for the pattern given in part 1 consist of the gateway service being properly designed to meet availability requirements and has good enough performance to deal with heavy loads and can be easily scalable depending on your service needs. The gateway routing level 7 is required, based on IP, port, header, or URL. This pattern is best used when client services need to be accessed behind a single endpoint gateway, consume services in multiple regions and multiple instances, route requests from external endpoints to internal endpoints, and clients can run multiple versions of the service at the same time.

The requirements that can be achieved by the procedures shown in part 2 and part 3 include using a gateway routing level 7 such as azure instance used and nginx as the router. Another requirement utilized and achieved is exposing ports to cluster virtual IP addresses in order to route requests from external endpoints to internal endpoints.

Design

Autoscaling in GKE

In GKE, autoscaling is achieved by implementing a Horizontal Pod Autoscaler (HPA). HPA automatically scales the workload to match demand, specifically it assigns more pods when more demand is increased.

Autoscaling is specifically achieved by the help of a load balancer which is responsible for shifting resources when demand is increased and new instances are created by the autoscaler. Furthermore, a request splitter differs from an autoscaler in the sense that it is a static resource manager. A resource splitter has its rules pre-defined and cant be changed while the system is running, which is in direct contrast to an autoscaler which dynamically creates new pods based on present demand.

Deliverables

Group Submission Deliverable

Part 2 and Part 3 Results

https://youtu.be/MdtRr0oYhmg

Design part using autoscaling with GKE

https://drive.google.com/drive/folders/1i4x0Bz2oGSHUoB7PWhLupvLqnj0C_Yrx?usp=sharing

Individual Work

Adris Azimi:

https://drive.google.com/drive/folders/1i4x0Bz2oGSHUoB7PWhLupvLqnj0C_Yrx?usp=sharing

Deepak Thangella:

https://drive.google.com/drive/folders/1_Pwfy-Rjfo-jenTq6EKbpwcLLlHGae9C?usp=sharing

Harris Athwal:

https://drive.google.com/drive/folders/1oMPbTTHE9tcmkP_ihQAQ9EzTdy3k55M9?usp=sharing

Krystian Rusin:

https://youtu.be/MdtRr0oYhmg