Faculty of Engineering & Applied Science

SOFE 4790U
Distributed Systems

Lab 1: Deploying a request splitting ambassador and a load balancer with Kubernetes

Group#: 14

First Name	Last Name	Student Number
Rodaba	Ebadi	100708585

Objective:

1. Learn how to configure and run a request splitter using Nginx.
2. Be familiar with the ConfigMap tool in Kubernetes.
3. Learn how to use the curl command for requesting an HTTP method.
4. Learn how to configure load balancer services
5. Get Familiar with load balancing pattern

Part 2: https://drive.google.com/file/d/1ybVfTfrgN7MbmNgSzSAIYEjam9bMGoV6/view

Part 3: https://drive.google.com/file/d/1rZQ4BKstJ8HC8e6eDJfQITx8Cn-4wV87/view

Autoscaler Deployment:
https://drive.google.com/file/d/1M6tr-upciV4ICzht10dTFddjqEo_z88F/view

Discussion:
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?

The problem for the pattern given in Part 1 is that the users are to be updated when services are modified like when they are added or removed and if the user requires multiple services or instances. The solution for this problem would be to disparate services, same service instances, and same service multiple versions. Going back to discovering issues without having an effect on clients is a solution. Clients can also add, divide, and reorder services without change. Requirements would include active-active deployment in order to improve latency and also to increase availability of services. The three main problems for the pattern given in part 1 are that the client has to be updated when multiple versions of the same service exit, multiple disparate services, and multiple instances of the same service.

1. Multiple versions of same service: Blue-green deployment is an operation involving a distribution plan where services are deployed with different versions within the same service.
2. Multiple disparate services: In order to separate services on an app into two or more, the application's code must be modified in the client's end and the services end.
3. Multiple instances of same service: On some occurrences, multiple instances of same service must be running in the same part of the system

Solution: The solution for the problem mentioned above would be to use Layer 7 application routing. This solution guarantees that any request routes to the right instance while placing a gateway ahead of the set of services, apps and deployments.

Requirements:

- Layer 7 of the application is mandatory (is accessible on IP)
- Ensure there is no cascade failure

- Backend services have restricted access/limited access from public network
- Making sure that the design of the system matches availability needs
- Ensuring there is access to multiple versions of the service during the same period

Having access to multiple versions of the same service was shown during part 2 of this lab. Two versions of the same service were created, the web-deployment.yaml was created first, followed by the experiment-deployment.yaml file.

Auto Scaling: provides an automated approach for users to increase/decrease allocated networking or memory resources. It is implemented for virtualization and users are able to define a instance with a capacity