Istio - Circuit Breaker

Istio DestinationRule 설정을 통해, 장애가 감지된 서비스를 서비스 대상에서 일정시간 동안 제외(Pool Ejection)하는 서비스Resiliency 를 실습한다.

Namespace 생성 및 Istio Activation

kubectl create namespace istio-cb-ns
kubectl label namespace istio-cb-ns istio-injection=enabled

Istio가 설정된 네임스페이스에 배송서비스 배포/ 서비스 생성

kubectl create deploy delivery --image=ghcr.io/acmexii/delivery:istio-v1 -n istio-cb-ns
kubectl expose deploy delivery --port=8080 -n istio-cb-ns

```
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl create namespace istio-cb-ns
namespace/istio-cb-ns created
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl label namespace istio-cb-ns istio-injection=enabled
namespace/istio-cb-ns labeled
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl create deploy delivery --image=ghcr.io/acmexii/delivery:istio-v1 -n istio-cb-ns
deployment.apps/delivery created
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl expose deploy delivery --port=8080 -n istio-cb-ns
service/delivery exposed
```

Circuit Breaker 설치

```
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl apply -f - << EOF
>   apiVersion: networking.istio.io/v1alpha3
>   kind: DestinationRule
>   metadata:
>     name: dr-delivery
>     namespace: istio-cb-ns
>   spec:
>     host: delivery
>     trafficPolicy:
>       outlierDetection:
>         consecutive5xxErrors: 1
>         interval: 1s
>         baseEjectionTime: 3m
>         maxEjectionPercent: 100
> EOF
destinationrule.networking.istio.io/dr-delivery created
```

모든 컨테이너가 제외된 경우, 'no healthy upstream' 오류를 리턴

delivery 서비스의 라우팅 대상 컨테이너 목록에서 **1초단위**로 체크하여 **1번이라도 서버 오류**가 발생 시, **3분 동안 라우팅에서 제외**하며, **모든 컨테이너가 제외될 수 있다**.

Circuit Breaker 테스트 환경설정

- 배송서비스의 Replica를 3개로 늘린다.

kubectl scale deploy delivery --replicas=3 -n istio-cb-ns

```
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl scale deploy delivery --replicas=3 -n istio-cb-ns
deployment.apps/delivery scaled
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl get pod -n istio-cb-ns
NAME                        READY   STATUS    RESTARTS   AGE
delivery-67ff6476bb-5shd5   2/2     Running   0          53s
delivery-67ff6476bb-bprnf   2/2     Running   0          53s
delivery-67ff6476bb-ckf9x   2/2     Running   0          4m42s
siege-75d5587bf6-sw7kt      2/2     Running   0          27s
```

- 새 터미널에서 Http Client 컨테이너를 설치하고, 접속한다

kubectl create deploy siege --image=ghcr.io/acmexii/siege-nginx:latest -n istio-cb-ns

kubectl exec -it pod/siege-75d5587bf6-sw7kt -n istio-cb-ns -- /bin/bash

```
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl exec -it pod/siege-75d5587bf6-sw7kt  -n istio-cb-ns -- /bin/bash
Defaulting container name to siege-nginx.
Use 'kubectl describe pod/siege-75d5587bf6-sw7kt -n istio-cb-ns' to see all of the containers in this pod.
```

Circuit Breaker 동작 확인

- Delivery 서비스 호출
- 컨테이너가 정상적으로 서비스 되고 있음을 확인
- 서비스 호출시 마다, 컨테이너이름/ IP(192.168.xx.xx)가 번갈아 출력

```
root@labs-910775232:/home/project/Istio-Timeout-and-Retry# kubectl exec -it pod/siege-75d5587bf6-sw7kt  -n istio-cb-ns -- /bin/bash
Defaulting container name to siege-nginx.
Use 'kubectl describe pod/siege-75d5587bf6-sw7kt -n istio-cb-ns' to see all of the containers in this pod.
root@siege-75d5587bf6-sw7kt:/# http http://delivery:8080/actuator/echo
HTTP/1.1 200 OK
content-length: 40
content-type: text/plain;charset=UTF-8
date: Fri, 18 Mar 2022 08:23:33 GMT
server: envoy
x-envoy-upstream-service-time: 325

delivery-67ff6476bb-5shd5/192.168.78.252

root@siege-75d5587bf6-sw7kt:/# http http://delivery:8080/actuator/echo
HTTP/1.1 200 OK
content-length: 39
content-type: text/plain;charset=UTF-8
date: Fri, 18 Mar 2022 08:23:44 GMT
server: envoy
x-envoy-upstream-service-time: 280

delivery-67ff6476bb-bprnf/192.168.23.79

root@siege-75d5587bf6-sw7kt:/# http http://delivery:8080/actuator/echo
HTTP/1.1 200 OK
content-length: 40
content-type: text/plain;charset=UTF-8
date: Fri, 18 Mar 2022 08:23:46 GMT
server: envoy
x-envoy-upstream-service-time: 19

delivery-67ff6476bb-5shd5/192.168.78.252
```

새로운 터미널에서 마지막에 출력된 Delivery 컨테이너로 접속하여 명시적으로 5xx 오류를 발생 시킨다.

```
# 새로운 터미널 Open
# 3개 중 하나의 컨테이너에 접속
kubectl exec -it pod/[DELIVERY POD객체] -n istio-cb-ns -c delivery -- /bin/sh
```

Delivery 컨테이너로 접속 상태에서

```
# httpie 설치 및 서비스 명시적 다운
apk update
apk add httpie
http PUT http://localhost:8080/actuator/down
```

Siege로 접속한 이전 터미널에서 Delivery 서비스로 접속해 3회 이상 호출해 본다.
3회 이상 호출하는 이유는 명시적으로 down시킨 컨테이너가 호출되도록 위함이다.

```
http GET http://delivery:8080/actuator/health
```

'/actuator/health' URL이 호출될 때, down상태인 컨테이너가 호출을 받게 되면 5xx오류를 리턴 한다.
이 때, 서킷 브레이커가 작동하여 해당 컨테이너를 3분 동안 Eject 한다.

최종, 아래 URL을 통해 3개 중 2개의 컨테이너만 서비스 됨을 확인한다.

```
http http://delivery:8080/actuator/echo
```

Pool Ejection 타임(3') 경과후엔 컨테이너 3개가 모두 동작됨이 확인된다

```
http http://delivery:8080/actuator/echo
```