

[Istio] Circuit Breaker

Istio DestinationRule 객체를 통해 샘플 서비스에 대한 Circuit Breaker 를 적용하고, 부하를 발생시켜 정상 동작하는지 테스트한다.

Instruction

Istio - Circuit Breaker

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

Namespace 생성 및 Istio Activation

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

- 이스티오가 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
```

Circuit Breaker 설치

```
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
```

- delivery 서비스의 라우팅 대상 컨테이너 목록에서 1 초단위로 체크하여 1 번이라도 서버 오류가 발생 시, 3 분동안 라우팅에서 제외하며, 모든 컨테이너가 제외될 수 있다.
- 모든 컨테이너가 제외된 경우, 'no healthy upstream' 오류를 리턴한다.

Circuit Breaker 테스트 환경설정

- 배송서비스의 Replica 를 3 개로 늘린다.

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

- Terminal 을 2 개 더 준비한다.
- 배송 컨테이너 Pod 의 IP 를 확인한다.

```
kubectl get pod -o wide -n istio-cb-ns
```

- 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 POD 객체] -n istio-cb-ns -- /bin/bash
```

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

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

- 새로운 터미널에서 마지막에 출력된 Delivery 컨테이너로 접속하여 명시적으로 5xx 오류를 발생 시킨다. (컨테이너 ID 메모해 두기)

```
# 새로운 터미널 Open
```

```
# 3 개 중 하나의 컨테이너에 접속
```

```
kubectl exec -it pod/[DELIVERY POD 객체] -n istio-cb-ns -c delivery - /bin/sh
```

```
#
```

```
# httpie 설치 및 서비스 명시적 다운
```

```
apk update
```

```
apk add httpie
```

```
http PUT http://localhost:8080/actuator/down
```

- Siege 로 접속한 이전 터미널에서 Delivery 서비스로 접속해 3 회 이상 호출해 본다.
- 3 회 이상 호출하는 이유는 명시적으로 down 시킨 컨테이너가 호출되도록 위함이다.

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

- '/actuator/health' URL 이 호출될 때, down 상태인 컨테이너가 호출을 받게되면 5xx 오류를 리턴한다.
- 이 때, 서킷브레이커가 작동하여 해당 컨테이너를 3 분동안 Eject 한다.

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

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

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

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

Kiali 를 통한 Circuit Breaker 확인

- 모니터링 시스템(Kiali) : EXTERNAL-IP:20001 에서 배송서비스에 **Circuit Breaker** 발생 확인

Clear Istio

```
kubect1 delete ns tutorial istio-cb-ns  
istioc1 x uninstall --purge
```

CheckPoints

1. 모든 요구사항을 만족하는가 ☐