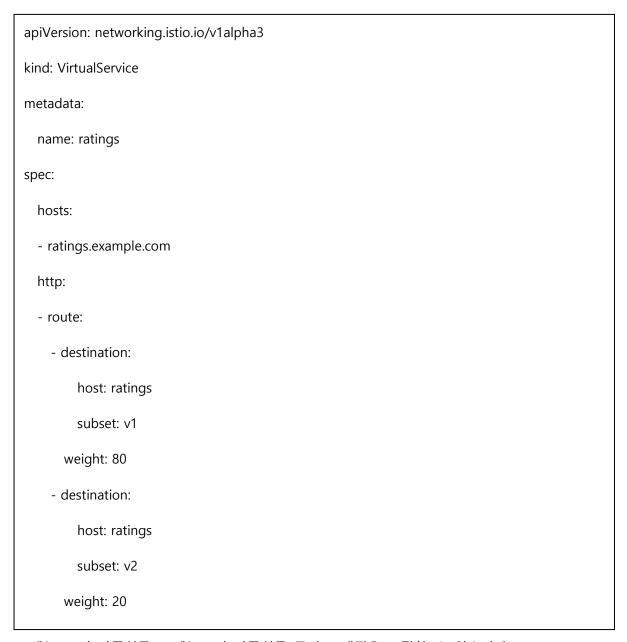
## Assignment #2

(2)는 Istio의 vertualService를 통해 트래픽 배분 비율을 조절할 수 있습니다.



V1에는 80의 가중치를, v2에는20의 가중치를 두어 트래픽을 조절할 수 있습니다.

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings.example.com
  http:
  - route:
    - destination:
         host: ratings
         subset: v1
      weight: 20
    - destination:
         host: ratings
         subset: v2
      weight: 40
    - destination:
         host: ratings
         subset: v3
      weight: 40
```

마찬가지로 v1:v2:v3 를 20:40:40 으로 가중치를 설정해 트래픽을 분산할 수 있습니다.

(3) Istio 의 Fault Injection 을 통해 서비스 간 네트워크 관련 문제를 시뮬레이션 할 수 있습니다.

## **Delay Injection**

apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
name: reviews-delay
spec:
hosts:
- reviews.example.com
http:
- fault:
delay:
percentage:
value: 100
fixedDelay: 5s
route:
- destination:
host: reviews
subset: v1

 Reviews.example.com
 으로
 들어오는
 모든
 요청에
 대해
 100%의
 지연을
 설정하고,
 응답을
 5
 초간

 지연시켜
 Delay Injection 을
 수행하고
 모니터링
 할
 수
 있습니다.

## **Retry Timeout**

apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
name: ratings-timeout
spec:
hosts:
- ratings.example.com
http:
- fault:
abort:
percentage:
value: 100
httpStatus: 503
route:
- destination:
host: ratings
subset: v1

특정 서비스가 일시적으로 응답하지 않을 때를 시뮬레이션할 수 있습니다. Ratings.example.com 으로 들어오는 모든 요청에 대해 100%의 경우 503 오류를 반환해 타임아웃을 유발할 수 있습니다.