

# [Istio] Traffic Mgmt & Canary 배포

Istio 의 Improved 한 트래픽 관리를 활용한 카나리 배포(Canary Deployment)와 Request Header 정보 기반의 라우팅을 실습한다.

## Instruction

---

## Traffic Mgmt & Canary 배포

### Istio Tutorial 셋업

- Git repository 에서 Tutorial 리소스 가져오기

```
cd /home/project
git clone https://github.com/redhat-developer-demos/istio-tutorial
cd istio-tutorial
```

### 네임스페이스 생성

```
kubectl create namespace tutorial
```

### Customer Service 배포 생성확인

```
kubectl apply -f <(istioctl kube-inject -f
customer/kubernetes/Deployment.yml) -n tutorial
kubectl describe pod/[CUSTOMER POD Obj] -n tutorial
kubectl create -f customer/kubernetes/Service.yml -n tutorial
```

### Istio Gateway 설치 및 Customer 서비스 라우팅(VirtualService) 설정

```
cat customer/kubernetes/Gateway.yml
kubectl create -f customer/kubernetes/Gateway.yml -n tutorial
```

### Istio-IngressGateway 를 통한 Customer 서비스 확인

```
kubectl get service/istio-ingressgateway -n istio-system
```

- 해당 EXTERNAL-IP 가 Istio Gateway 주소
- Customer 서비스 호출

```
http://(istio-ingressgateway IP)/customer
```

### Preference, Recommendation-v1 Service 배포

```
kubectl apply -f <(istioctl kube-inject -f
preference/kubernetes/Deployment.yml) -n tutorial
kubectl create -f preference/kubernetes/Service.yml -n tutorial
```

```
kubectl apply -f <(istioctl kube-inject -f
recommendation/kubernetes/Deployment.yml) -n tutorial
kubectl create -f recommendation/kubernetes/Service.yml -n tutorial
```

## Simple Routing

`pwd` 로 현 위치가 `/istio-tutorial/` 인지 확인

`recommendation` 서비스 추가 배포: `v2`

```
kubectl apply -f <(istioctl kube-inject -f
recommendation/kubernetes/Deployment-v2.yml) -n tutorial
```

서비스 호출

- 브라우저에서 **Customer** 서비스(<http://Istio-Gateway-IP/customer>) 호출
- **F5(새로고침)**을 **10** 회 이상 클릭하여 다수의 요청 생성
- **Routing** 결과 확인 - **Kiali(External-IP:20001)**
- 접속 서비스의 **v2** 의 **replica** 를 **2** 로 설정

```
kubectl scale --replicas=2 deployment/recommendation-v2 -n tutorial
kubectl get po -n tutorial
```

- **Customer** 서비스를 **10** 회 이상 **F5(새로고침)**하여 서비스 호출
- **Routing** 결과 확인 - **Kiali(External-IP:20001)** 접속

## Advanced Routing with Istio

- 정책(VirtualService, DestinationRule) 설정 확인

```
kubectl get VirtualService -n tutorial -o yaml
kubectl get DestinationRule -n tutorial -o yaml
```

- 사용자 선호도에 따른 추천 서비스 라우팅 정책 설정
- **version-2** 로 **100%** 라우팅

```
kubectl create -f istiofiles/destination-rule-recommendation-v1-
v2.yml -n tutorial
```

```
kubectl create -f istiofiles/virtual-service-recommendation-v2.yml -
n tutorial
```

- 정책 설정 확인

```
kubectl get VirtualService -n tutorial -o yaml
kubectl get DestinationRule -n tutorial -o yaml
```

- 서비스 확인  
브라우저에서 **Customer** 서비스(<http://Istio-Gateway-IP/customer>) 호출  
**Kiali(External-IP:20001)**, **Jaeger(External-IP:80)** 에서 모니터링

## 가중치 기반 스마트 라우팅

- **recommendation** 서비스 v1의 가중치를 100으로 변경

```
kubectl replace -f istiofiles/virtual-service-recommendation-v1.yaml -n tutorial
```

서비스 호출 및 Kiali(Externl-IP:20001)에서 모니터링

**VirtualService** 삭제 시, **Round-Robin** 방식으로 동작

```
kubectl delete -f istiofiles/virtual-service-recommendation-v1.yaml -n tutorial
```

**Canary** 라우팅 비율별 배포 정책 예시

(90 : 10)

```
kubectl apply -f istiofiles/virtual-service-recommendation-v1_and_v2.yaml -n tutorial
```

(75 : 25)

```
kubectl replace -f istiofiles/virtual-service-recommendation-v1_and_v2_75_25.yaml -n tutorial
```

## Header 기반 라우팅(브라우저 유형별)

- **Firefox** 브라우저로 접속 시, v2로 라우팅되도록 설정

```
kubectl apply -f istiofiles/destination-rule-recommendation-v1-v2.yaml -n tutorial
```

```
kubectl apply -f istiofiles/virtual-service-firefox-recommendation-v2.yaml -n tutorial
```

- **Firefox** 브라우저와 다른 브라우저에서 접속 확인 **Browser** 환경이 지원되지 않을 경우

```
curl -A Safari Externl-IP:8080
```

```
curl -A Firefox Externl-IP:8080
```

삭제

```
kubectl delete dr recommendation -n tutorial
```

```
kubectl delete vs recommendation -n tutorial
```

---

## CheckPoints

1. 모든 요구사항을 만족하는가 ☐