# Istio Traffic Management Lab Report

**Student name:Du Xilei**

**Student number:202383930004**

**Date:10.15.2025**

## 1. Experiment Overview

This lab demonstrates Istio's traffic management capabilities, including traffic shifting, request routing, fault injection, and circuit breaking. Each section follows the official Istio documentation and includes corresponding kubectl commands, observations, and screenshots.

## 2. Environment Setup

Commands executed:

minikube start

istioctl install --set profile=demo -y

kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml

kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml

kubectl get services

kubectl get pods

kubectl get svc -n istio-system

kubectl get gateway -n default

kubectl get virtualservice -n default

## 3. Traffic Shifting

Official guide: https://istio.io/latest/docs/tasks/traffic-management/traffic-shifting/

Objective: Demonstrate progressive traffic migration between versions using VirtualService and DestinationRule.

Commands executed:

kubectl apply -f samples/bookinfo/networking/virtual-service-all-v1.yaml

kubectl apply -f samples/bookinfo/networking/destination-rule-all.yaml

## 4. Request Routing

Official guide: https://istio.io/latest/docs/tasks/traffic-management/request-routing/

Objective: Route user-specific requests to different service versions (e.g., routing ‘jason’ to reviews:v2).

Commands executed:

kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-test-v2.yaml

## 5. Fault Injection

Official guide: https://istio.io/latest/docs/tasks/traffic-management/fault-injection/

Objective: Simulate network delays or aborts to observe failure responses.

Commands executed:

kubectl apply -f samples/bookinfo/networking/virtual-service-ratings-test-delay.yaml

kubectl apply -f samples/bookinfo/networking/virtual-service-ratings-test-abort.yaml

## 

## 6. Circuit Breaking

Official guide: https://istio.io/latest/docs/tasks/traffic-management/circuit-breaking/

Objective: Configure connection pool and outlier detection to demonstrate resilience.

Commands executed:

kubectl apply -f samples/bookinfo/networking/destination-rule-reviews.yaml

kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-cb.yaml

## 

## 7. Conclusion

Through this lab, we have practiced key Istio traffic management techniques, including traffic shifting, request routing, fault injection, and circuit breaking. These methods allow fine-grained control over microservice traffic flows and improve system reliability.

