Content

[**Traffic Shifting** 2](#_Toc218028950)

[**Request Routing** 4](#_Toc218028951)

[**Fault Injection** 7](#_Toc218028952)

[Fix bug: transfer all traffics to reviews:v3 9](#_Toc218028953)

[Injecting an HTTP abort fault 10](#_Toc218028954)

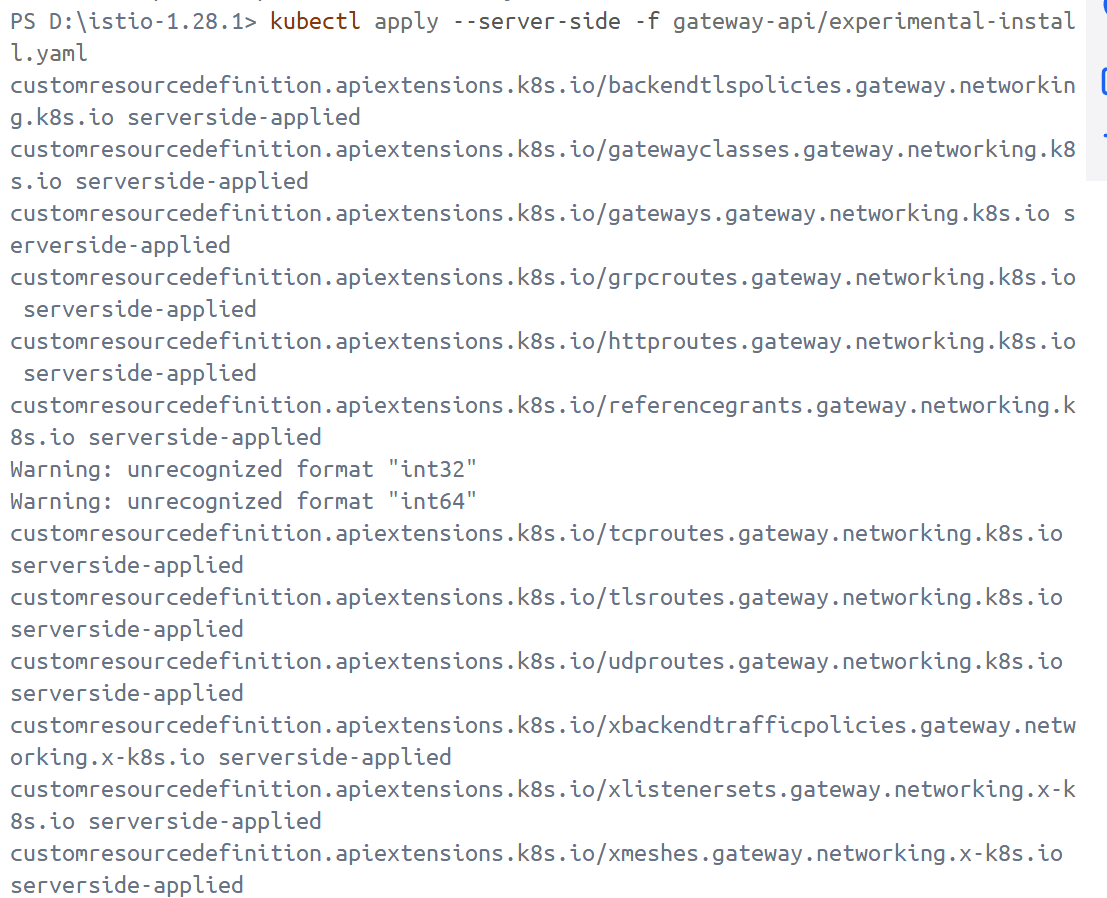
[Clean up 11](#_Toc218028955)

[**Circuit Breaking** 12](#_Toc218028956)

# **Traffic Shifting**

1. Preparation: install Kubernetes Gateway API CRDs

*kubectl apply --server-side -f* [*https://github.com/kubernetes-sigs/gateway-api/releases/download/v1.4.0/experimental-install.yaml*](https://github.com/kubernetes-sigs/gateway-api/releases/download/v1.4.0/experimental-install.yaml)



1. Apply weight-based routing
   1. Route all traffic to the v1 version

*kubectl apply -f samples/bookinfo/gateway-api/route-reviews-v1.yaml*

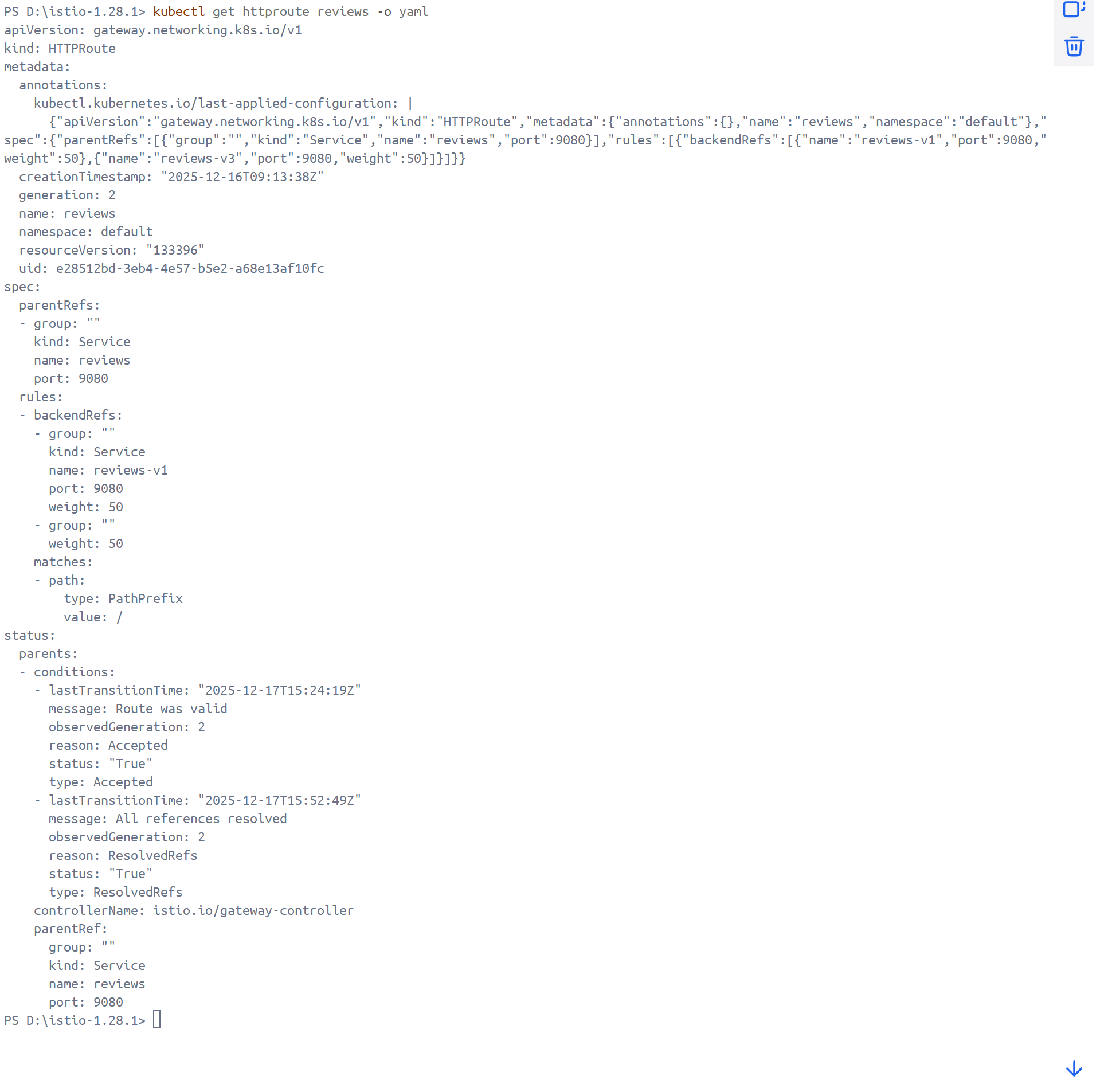
* 1. Transfer 50% of the traffic from reviews:v1 to reviews:v3

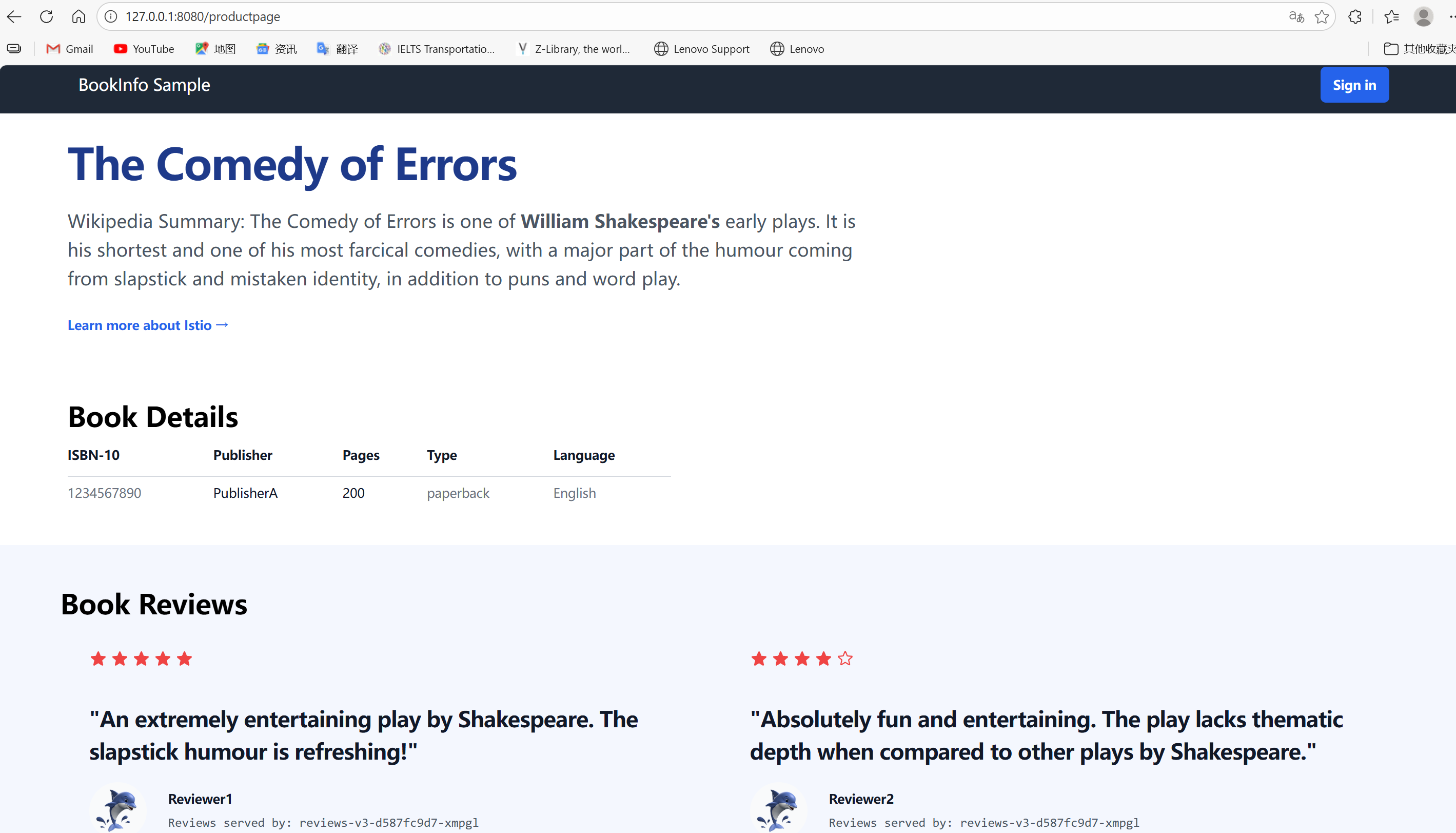
*kubectl apply -f samples/bookinfo/gateway-api/route-reviews-50-v3.yaml*



* 1. Wait a few seconds for the new rules to propagate and then confirm the rule was replaced

*kubectl get httproute reviews -o yaml*





* 1. Route 100% of the traffic to reviews:v3

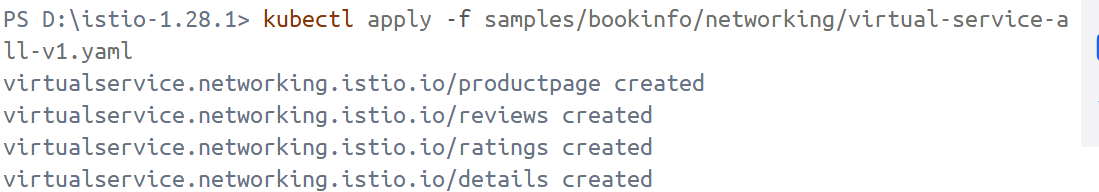
*kubectl apply -f samples/bookinfo/gateway-api/route-reviews-v3.yaml*



# **Request Routing**

1. Route to version 1
   1. Apply virtual services that will route all traffic to v1 of each microservice

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



* 1. Display the defined routes

*kubectl get virtualservices -o yaml*



1. Test the new routing configuration

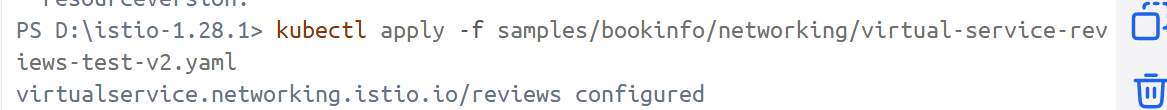
Notice that the reviews part of the page displays with no rating stars, no matter how many times we refresh. This is because we configured Istio to route all traffic for the reviews service to the version reviews:v1 and this version of the service does not access the star ratings service.

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

1. Route based on user identity
   1. Enable user-based routing

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



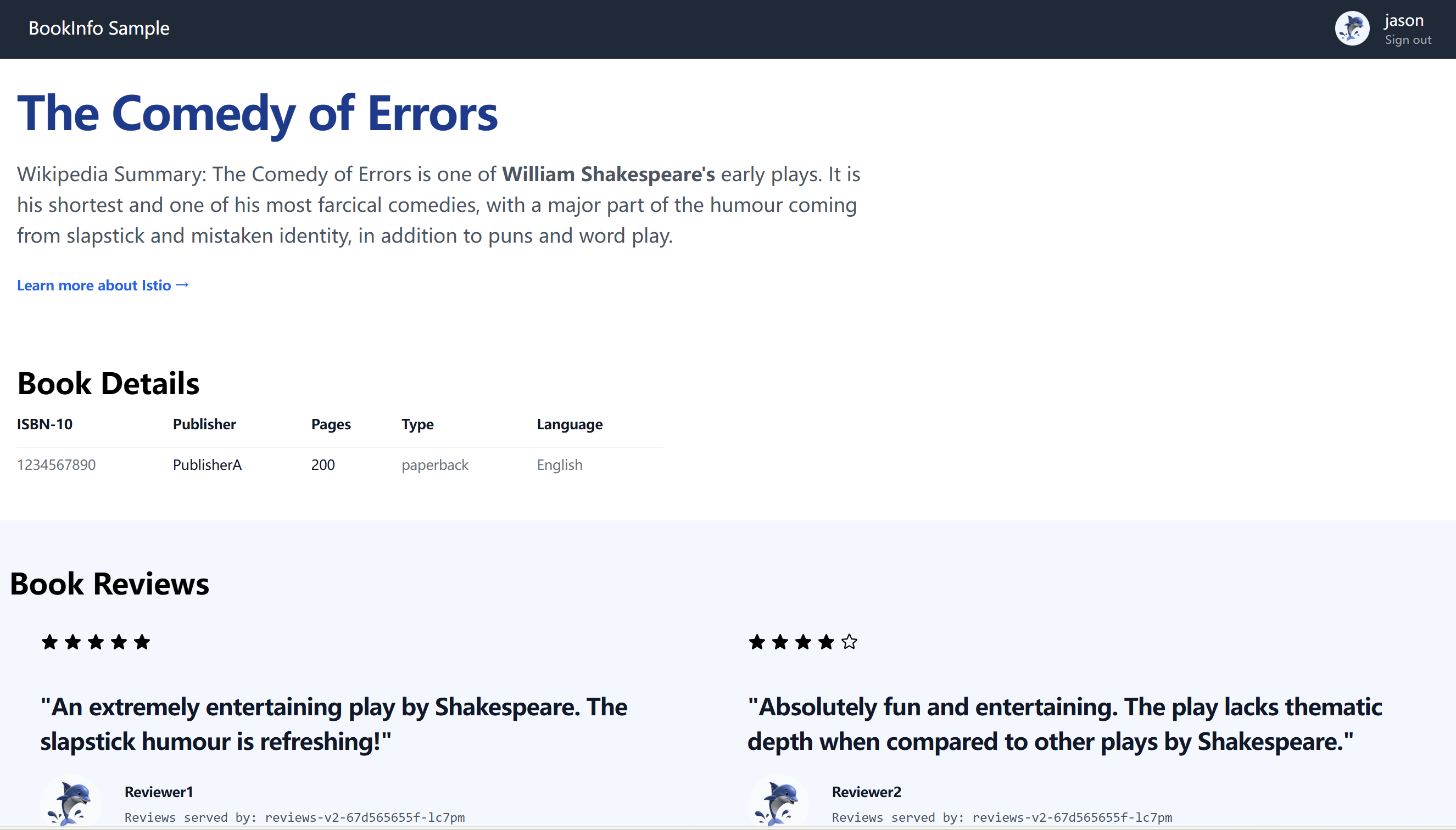
* 1. Confirm the rule is created

*kubectl get virtualservice reviews -o yaml*

文本

AI 生成的内容可能不正确。

* 1. On the /productpage of the Bookinfo app, log in as user jason. The star ratings appear next to each review.



* 1. Log in as another user and the stars are gone. This is because traffic is routed to reviews:v1 for all users except Jason.

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

# **Fault Injection**

1. Injecting an HTTP delay fault
   1. Create a fault injection rule to delay traffic coming from the test user jason

*kubectl apply -f* [*samples/bookinfo/networking/virtual-service-ratings-test-delay.yaml*](https://raw.githubusercontent.com/istio/istio/release-1.28/samples/bookinfo/networking/virtual-service-ratings-test-delay.yaml)

* 1. Confirm the rule was created

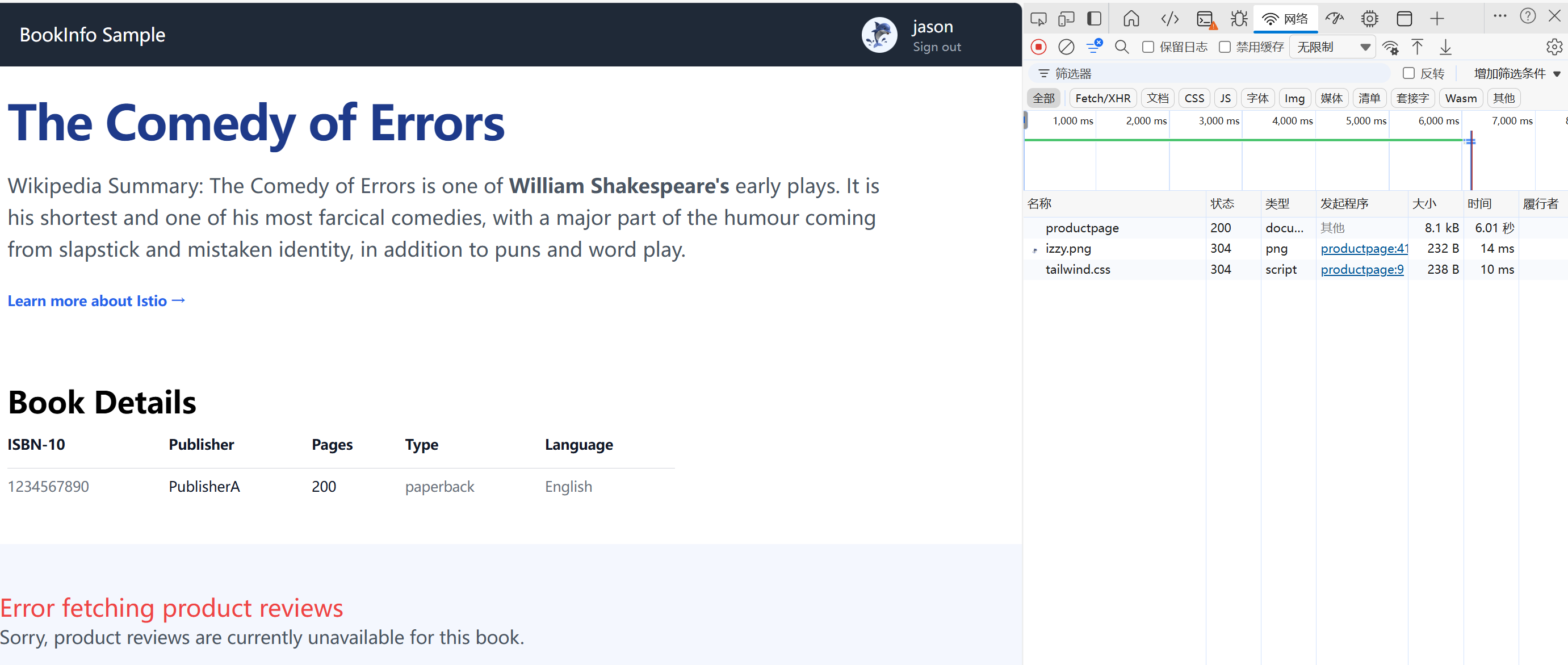
*kubectl get virtualservice ratings -o yaml*



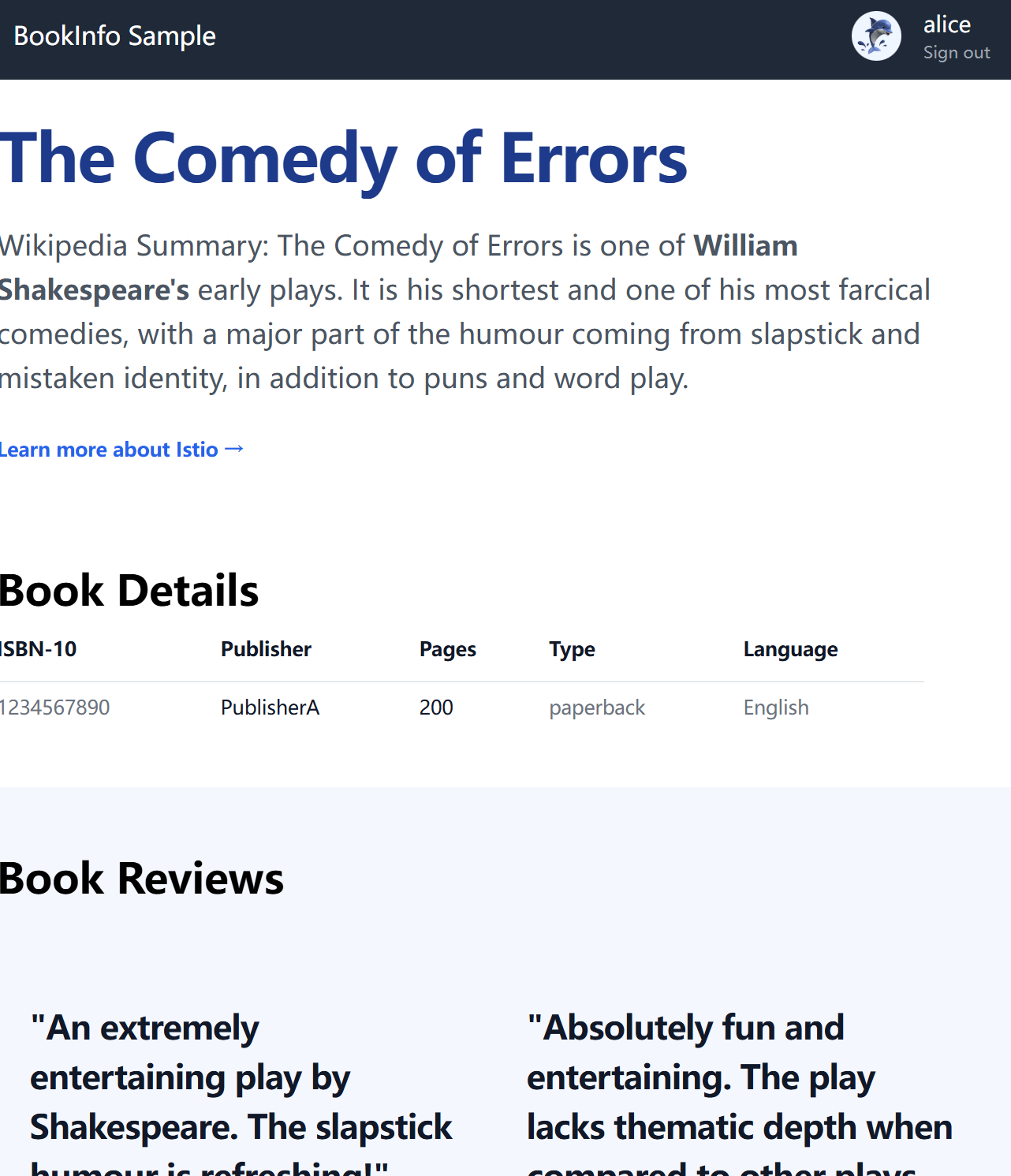
1. Testing the delay configuration

On the /productpage web page, log in as user Jason. There is a problem that the Reviews section displays an error message: “Sorry, product reviews are currently unavailable for this book.”

Open the *Developer Tools* menu and open the Network tab. It can be seen that the page actually loads in about 6 seconds.



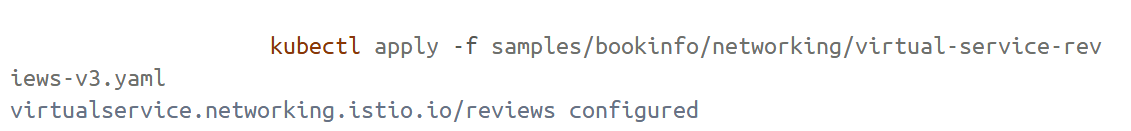
While logging in as any other user, there is no problem.

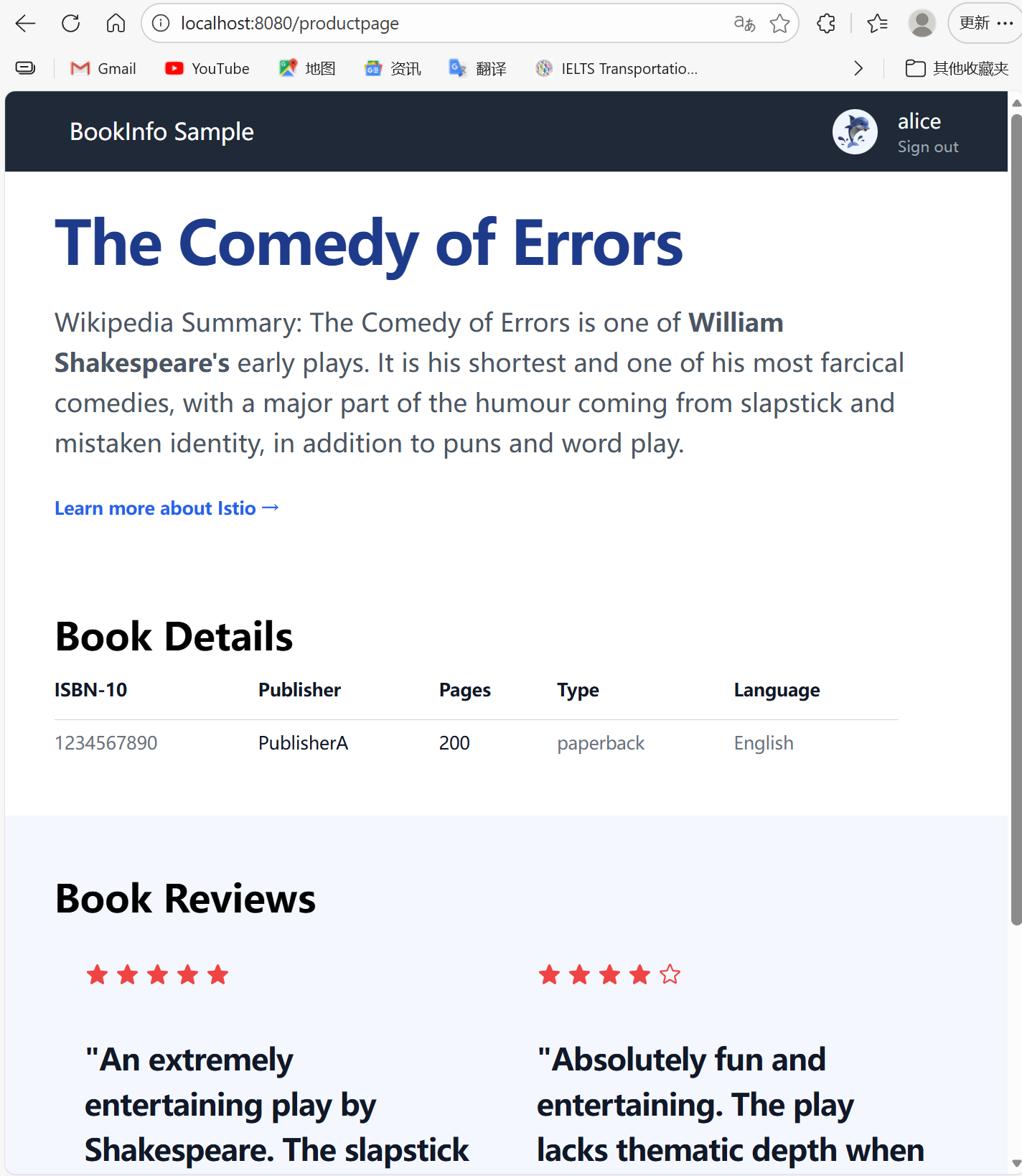


Reason: As expected, the 7s delay I introduced doesn’t affect the reviews service because the timeout between the reviews and ratings service is hard-coded at 10s. However, there is also a hard-coded timeout between the productpage and the reviews service, coded as 3s + 1 retry for 6s total. As a result, the productpage call to reviews times out prematurely and throws an error after 6s.

## Fix bug: transfer all traffics to reviews:v3

*kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-v3.yaml*

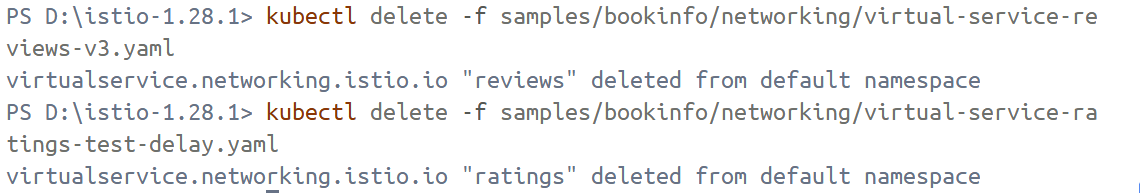




Then delete the following rules and add necessary rules to make preparation for another way.

*kubectl delete -f samples/bookinfo/networking/virtual-service-reviews-v3.yaml*

*kubectl delete -f samples/bookinfo/networking/virtual-service-ratings=test=delay.yaml*



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

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

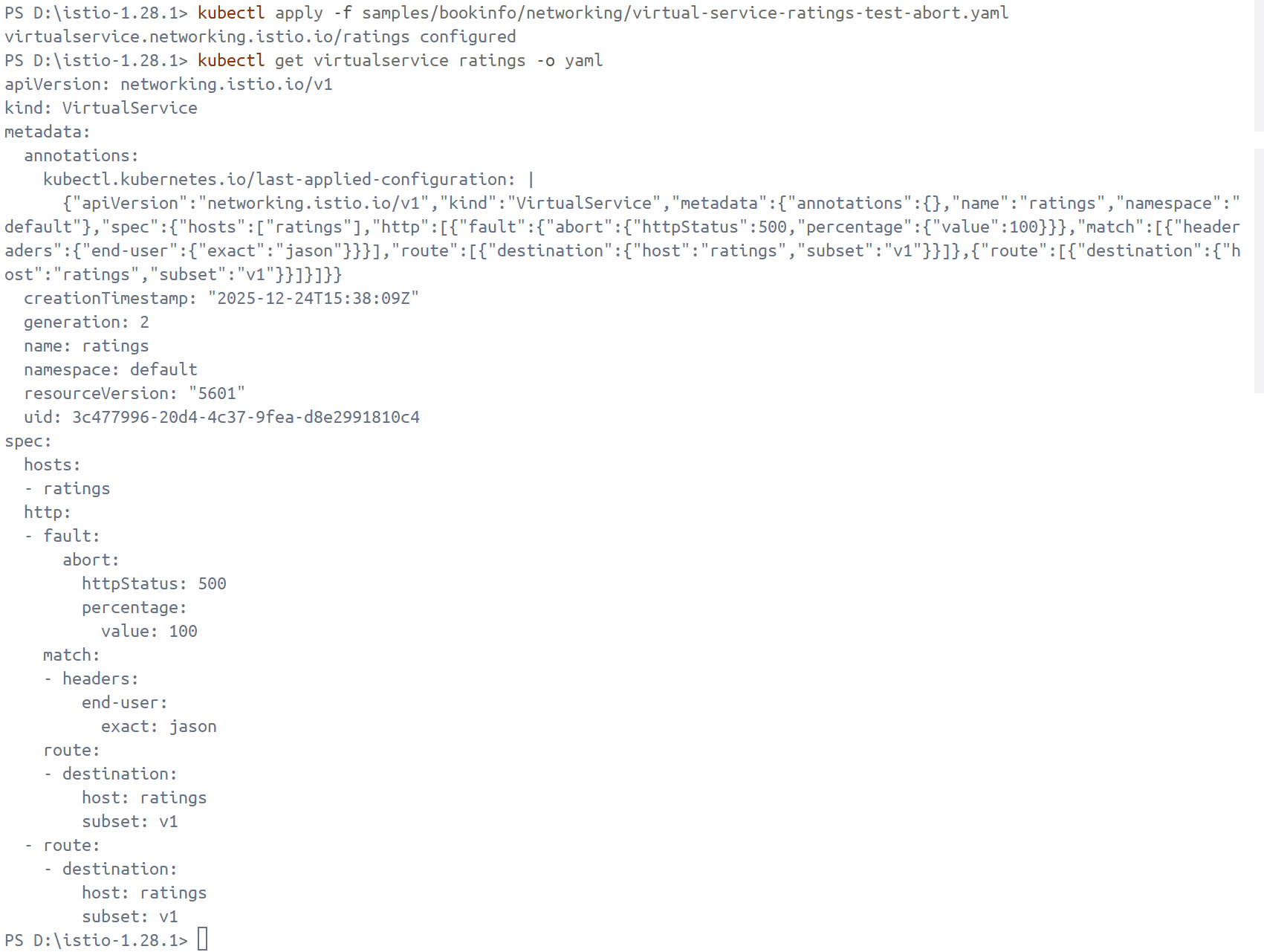
## Injecting an HTTP abort fault

1. Create a fault injection rule to send an HTTP abort for user jason

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

1. Confirm the rule was created

*kubectl get virtualservice ratings -o yaml*

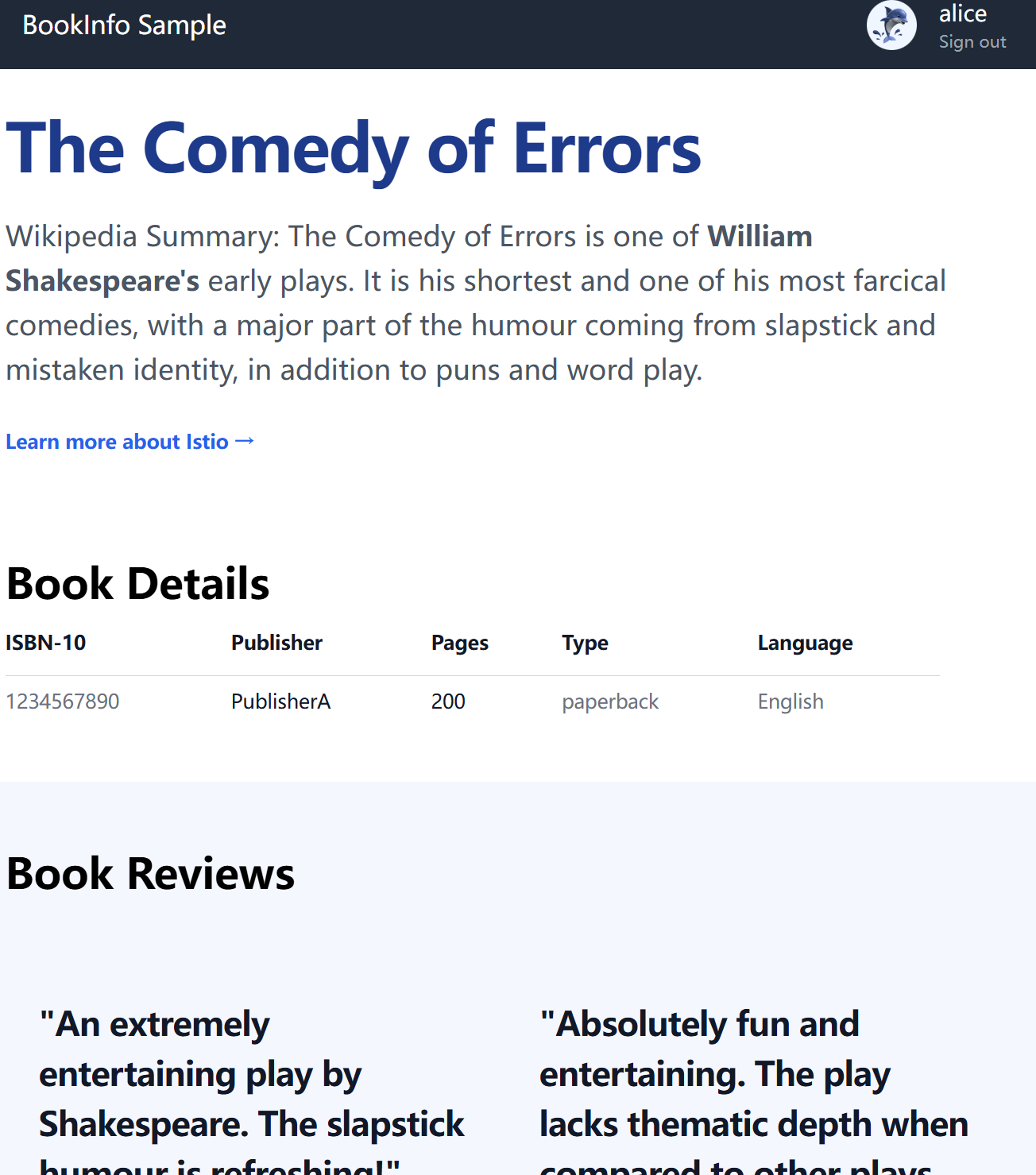


1. Test the abort configuration

On the /productpage, log in as user jason. The page loads immediately and the “Ratings service is currently unavailable” message appears. However, while logging in another user, there is no problem.

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。



## Clean up

Remove the application routing rules

*kubectl delete -f* [*samples/bookinfo/networking/virtual-service-ratings-test-abort.yaml*](https://raw.githubusercontent.com/istio/istio/release-1.28/samples/bookinfo/networking/virtual-service-ratings-test-abort.yaml)

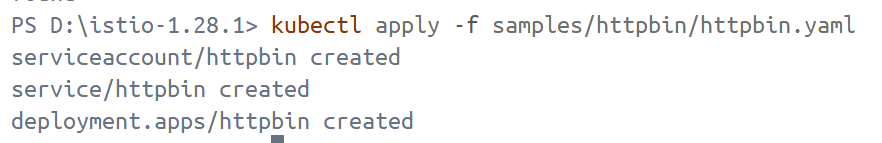
*kubectl delete -f samples/bookinfo/networking/virtual-service-all-v1.yaml*

*kubectl delete -f* [*samples/bookinfo/networking/virtual-service-reviews-test-v2.yaml*](https://raw.githubusercontent.com/istio/istio/release-1.28/samples/bookinfo/networking/virtual-service-reviews-test-v2.yaml)

# **Circuit Breaking**

1. Preparation: deploy the httpbin service

*kubectl apply -f* [*samples/httpbin/httpbin.yaml*](https://raw.githubusercontent.com/istio/istio/release-1.28/samples/httpbin/httpbin.yaml)



1. Configurating the circuit breaker
   1. Create a destination rule to apply circuit breaking settings when calling the httpbin service

*@"*

*apiVersion: networking.istio.io/v1*

*kind: DestinationRule*

*metadata:*

*name: httpbin*

*spec:*

*host: httpbin*

*trafficPolicy:*

*connectionPool:*

*tcp:*

*maxConnections: 1*

*http:*

*http1MaxPendingRequests: 1*

*maxRequestsPerConnection: 1*

*outlierDetection:*

*consecutive5xxErrors: 1*

*interval: 1s*

*baseEjectionTime: 3m*

*maxEjectionPercent: 100*

*"@ | kubectl apply -f –*



* 1. Verify the destination rule was created correctly

*kubectl get destinationrule httpbin -o yaml*

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

1. Add a client
   1. Deploy the fortio service

*kubectl apply -f samples/httpbin/sample-client/fortio-deploy.yaml*



3.2. Log in to the client pod and use the fortio tool to call httpbin. Pass in curl to indicate that you just want to make one call

$FORTIO\_POD = kubectl get pods -l app=fortio -o jsonpath="{.items[0].metadata.name}"

kubectl exec $FORTIO\_POD -c fortio -- `

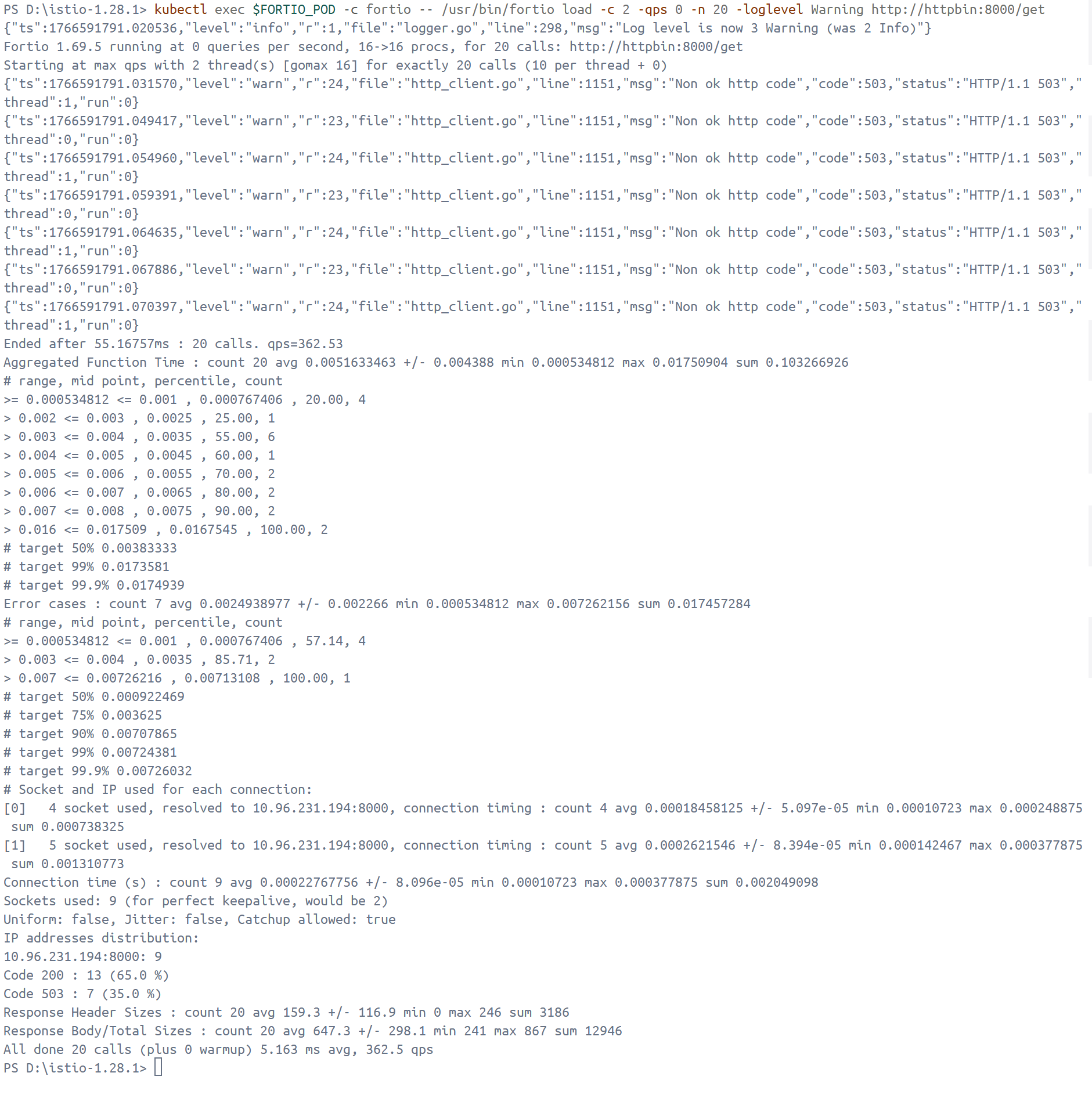
/usr/bin/fortio curl -quiet <http://httpbin:8000/get>





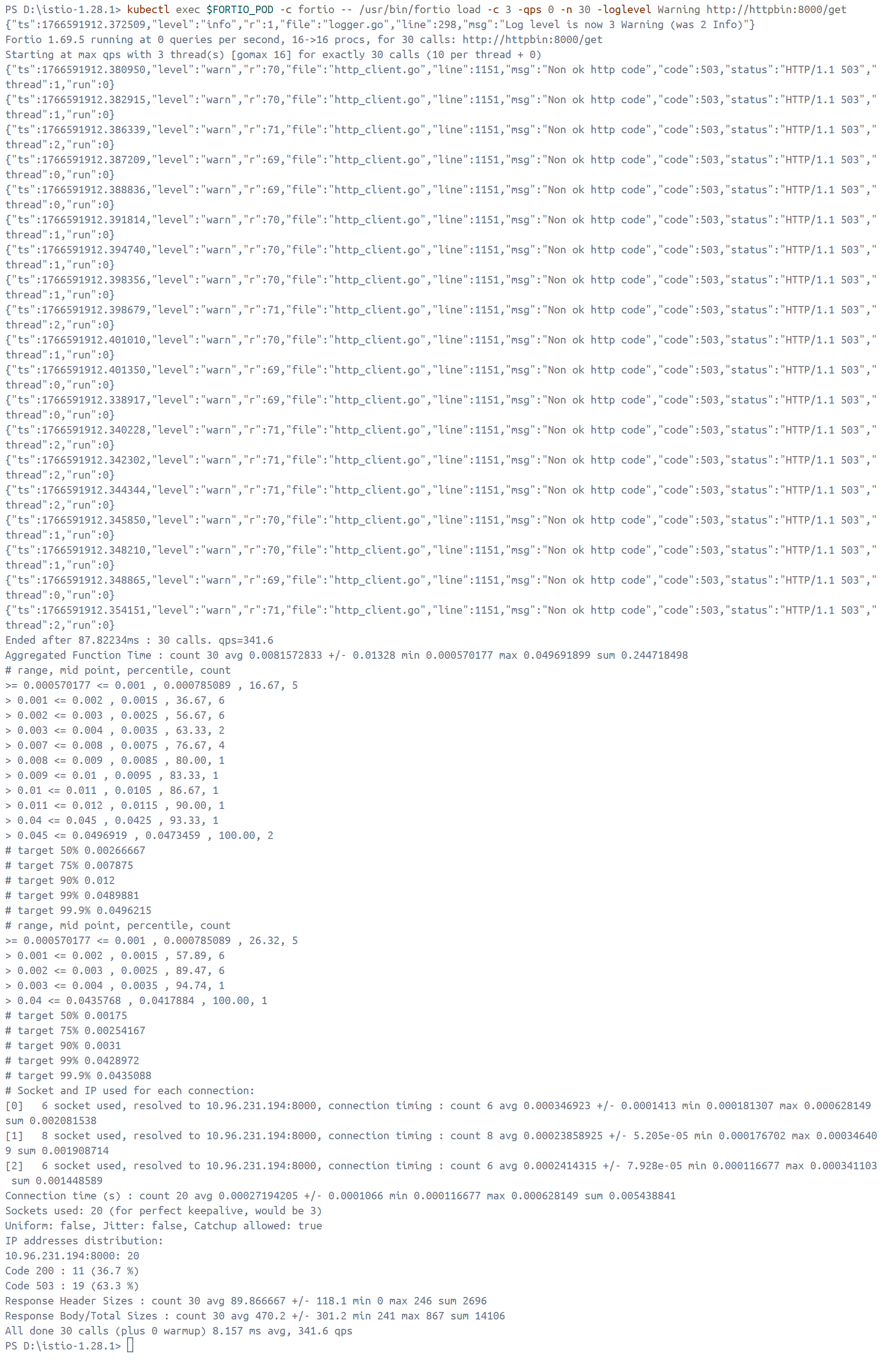
1. Tripping the circuit breaker
   1. Call the service with two concurrent connections (-c 2) and send 20 requests (-n 20)

*kubectl exec $FORTIO\_POD -c fortio -- /usr/bin/fortio load -c 2 -qps 0 -n 20 -loglevel Warning http://httpbin:8000/get*



* 1. Bring the number of concurrent connections up to 3

*kubectl exec $FORTIO\_POD -c fortio -- /usr/bin/fortio load -c 3 -qps 0 -n 30 -loglevel Warning http://httpbin:8000/get*



Only 36.7% of the requests succeeded and the rest were trapped by circuit breaking:

* 1. Query the istio-proxy stats to see more

*kubectl exec $FORTIO\_POD -c istio-proxy -- `*

*pilot-agent request GET stats |*

*Select-String httpbin |*

*Select-String pending*



It shows 26 for the upstream\_rq\_pending\_overflow value which means 26 calls so far have been flagged for circuit breaking.

1. Clean up
   1. Remove the rules:

*kubectl delete destinationrule httpbin*

* 1. Shutdown the httpbin service and client:

*kubectl delete -f samples/httpbin/sample-client/fortio-deploy.yaml*

*kubectl delete -f samples/httpbin/httpbin.yaml*

手机屏幕截图

AI 生成的内容可能不正确。