ISTIO-Service Mesh

[Install istio: 1](#_Toc73168291)

[Automatically inject Envoy Sidecar proxies to default namespace 1](#_Toc73168292)

[Deploying book-info app: 1](#_Toc73168293)

[Verify everything working correctly or not: 1](#_Toc73168294)

[Apply Gateway opening app from outside: 1](#_Toc73168295)

[Finding Ingress IP and Port: 1](#_Toc73168296)

[Setting up Environment Variables: 1](#_Toc73168297)

[Access the App from Browser (Distributing V1/V2/V3 Apps-Round Robin Method): 1](#_Toc73168298)

[Kiali Dashboard: 1](#_Toc73168299)

[Install: 1](#_Toc73168300)

[Accessing Kiali Dashboard: 1](#_Toc73168301)

[Kiali Dashboard UI: 1](#_Toc73168302)

[Request Routing: 1](#_Toc73168303)

[Apply Default Destination Rules: 1](#_Toc73168304)

[Apply Virtual Service (All Requests Routes to V1): 1](#_Toc73168305)

[With Multiple Request Same Page Display 1](#_Toc73168306)

[Route User Based Identity: 1](#_Toc73168307)

[Accessing Page (User:Jason Version-2 Displays) 1](#_Toc73168308)

[Accessing Page (Default User. V1 Displays) 1](#_Toc73168309)

[Cleanup: 1](#_Toc73168310)

[Fault Injection: 1](#_Toc73168311)

[Applying Fault Injection 1](#_Toc73168312)

[User-1 Accessing (Delaying 6 Seconds) 1](#_Toc73168313)

[Default User Accessing (No Delay) 1](#_Toc73168314)

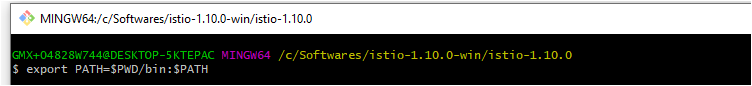
[Aborting Injection of HTTP fault 1](#_Toc73168315)

[All request no delay 1](#_Toc73168316)

# Download istio:

* Download Istio Installer Zip.
* Unzip the folder and copy it on C:/ Drive
* Go Inside the folder home and run the command on linux bash to add IstioCtl client to your path.

**$ export PATH=$PWD/bin:$PATH**



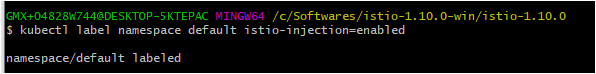
# Install istio:

**$ istioctl install --set profile=demo -y**



# Automatically inject Envoy Sidecar proxies to default namespace

**$ kubectl label namespace default istio-injection=enabled**



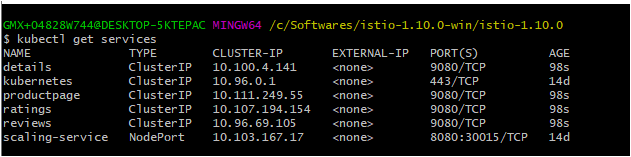
# Deploying book-info app:

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

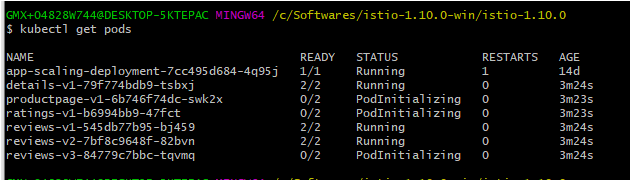
# 

# Check all services are deployed well

**$ kubectl get services**

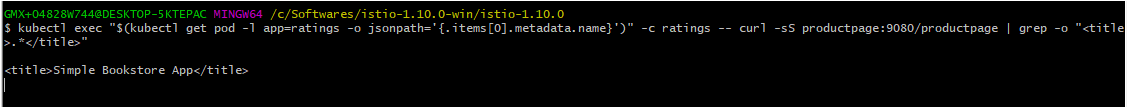


**$ kubectl get pods**



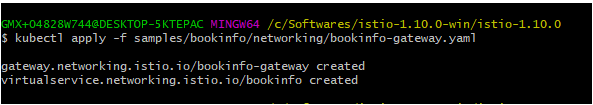
# Verify everything working correctly or not:

**$ kubectl exec "$(kubectl get pod -l app=ratings -o jsonpath='{.items[0].metadata.name}')" -c ratings -- curl -sS productpage:9080/productpage | grep -o "<title>.\*</title>"**



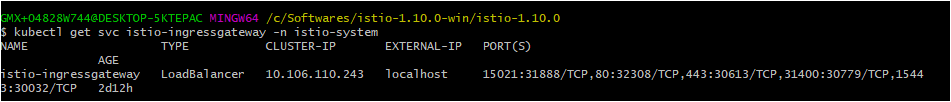
# Apply Gateway opening app from outside:

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



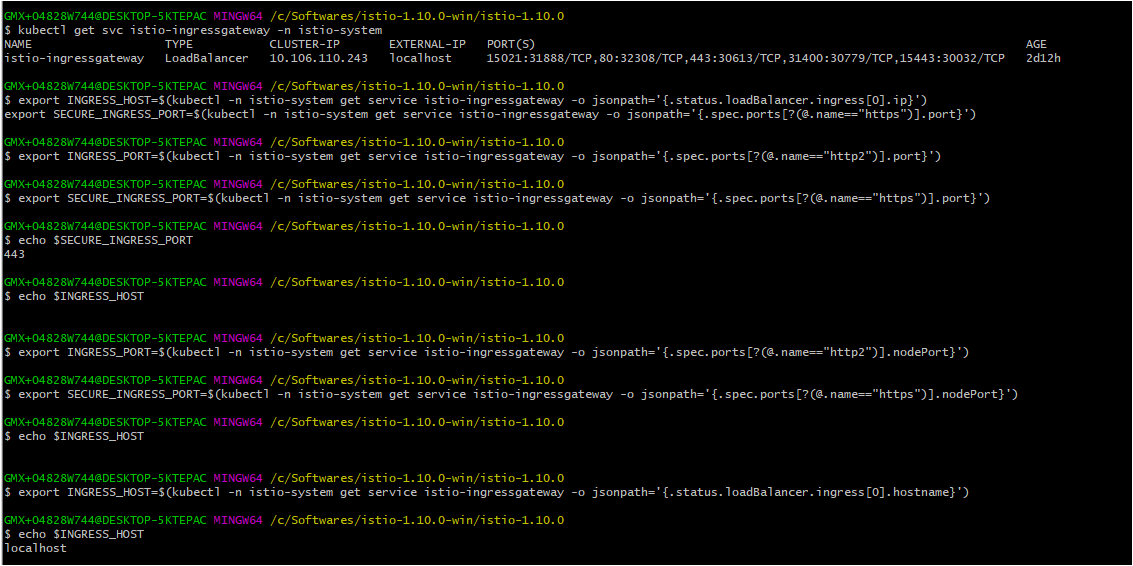
# Finding Ingress IP and Port:

kubectl get svc istio-ingressgateway -n istio-system

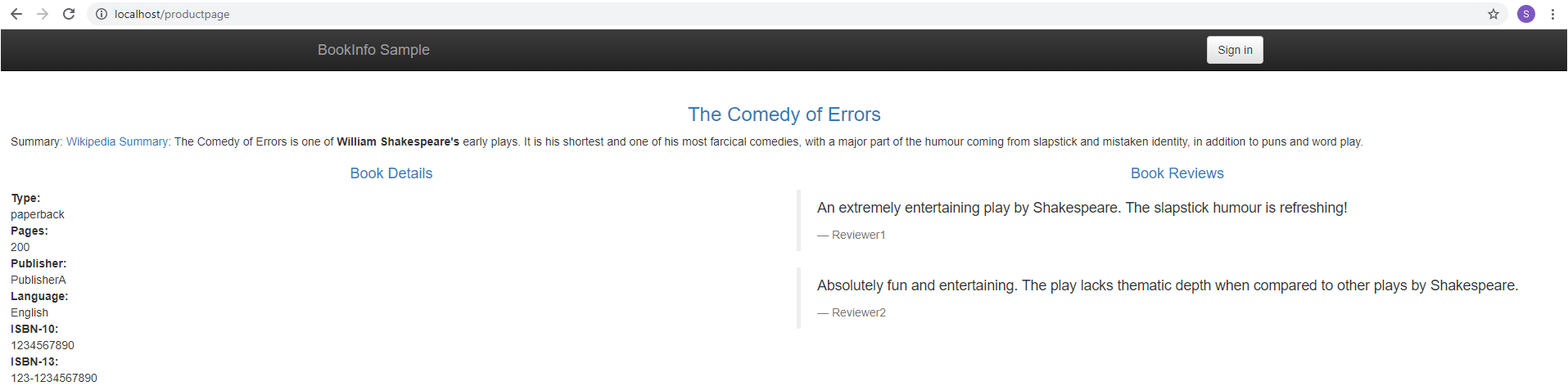


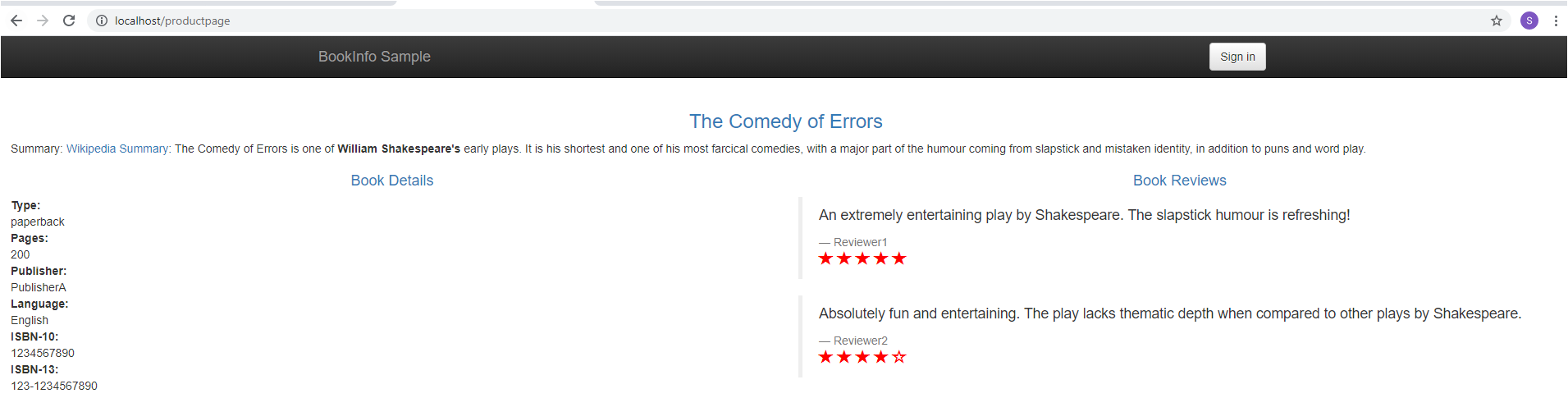
# Setting up Environment Variables:

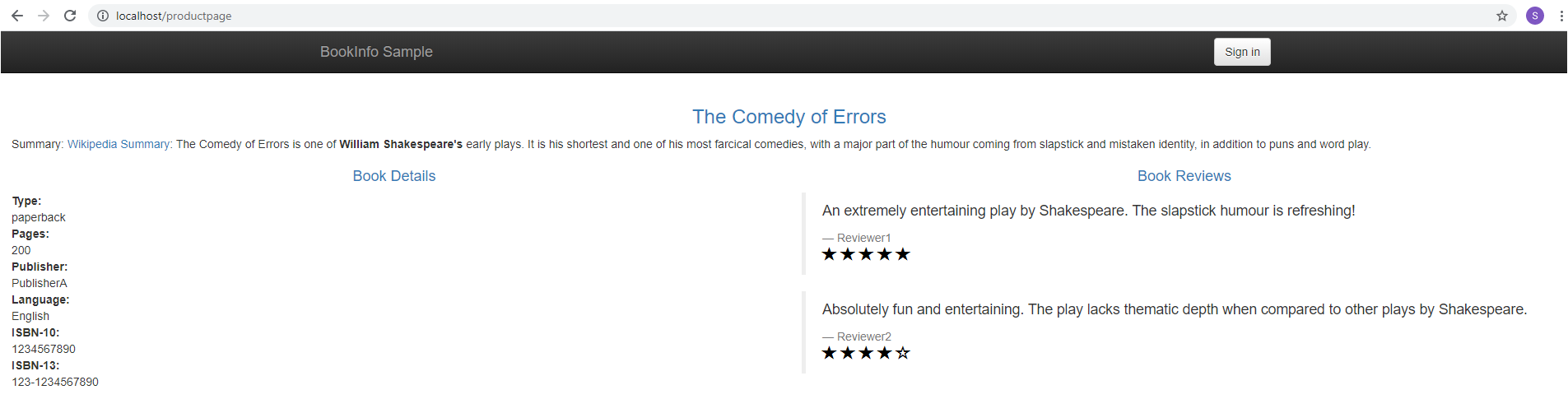
* export INGRESS\_HOST=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
* export INGRESS\_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="http2")].port}')
* export SECURE\_INGRESS\_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="https")].port}')
* echo $SECURE\_INGRESS\_PORT
* echo $INGRESS\_HOST
* export INGRESS\_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="http2")].nodePort}')
* export SECURE\_INGRESS\_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="https")].nodePort}')
* echo $INGRESS\_HOST
* export INGRESS\_HOST=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.status.loadBalancer.ingress[0].hostname}')



# Access the App from Browser (Distributing V1/V2/V3 Apps-Round Robin Method):



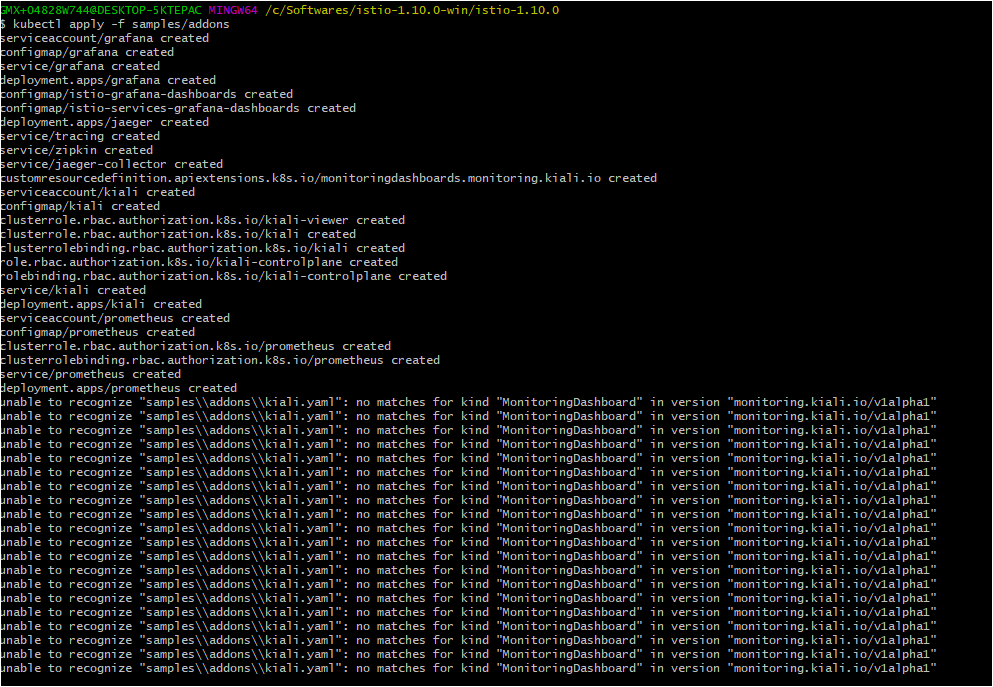




# Kiali Dashboard:

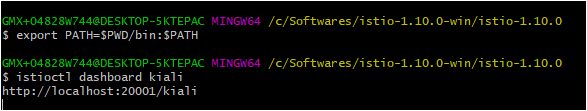
## Install:

$ kubectl apply -f samples/addons

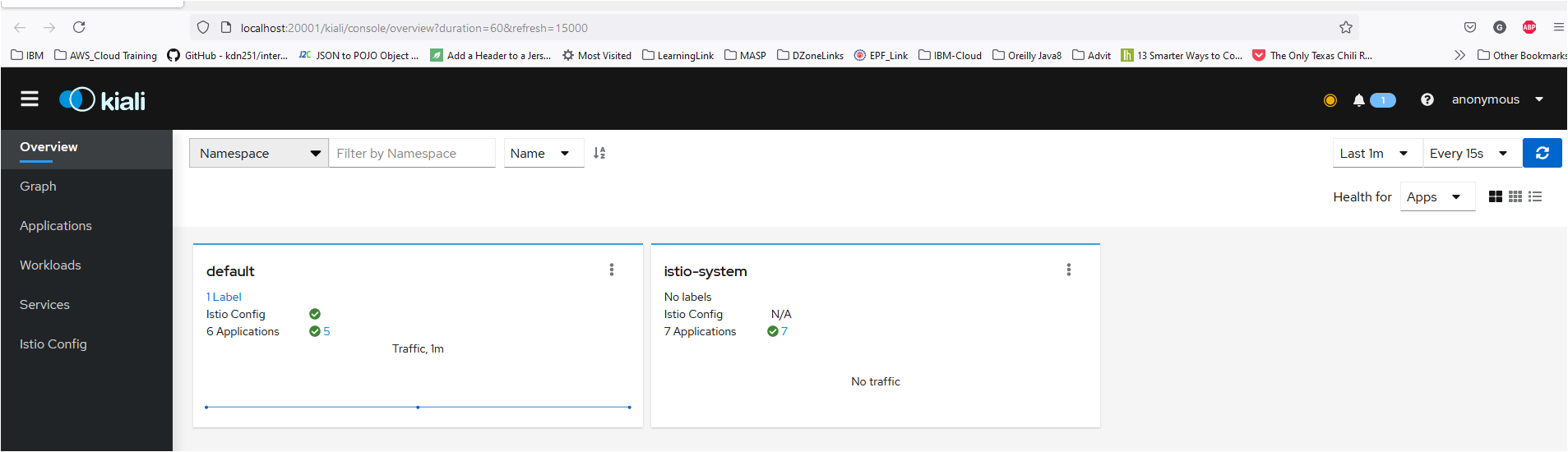


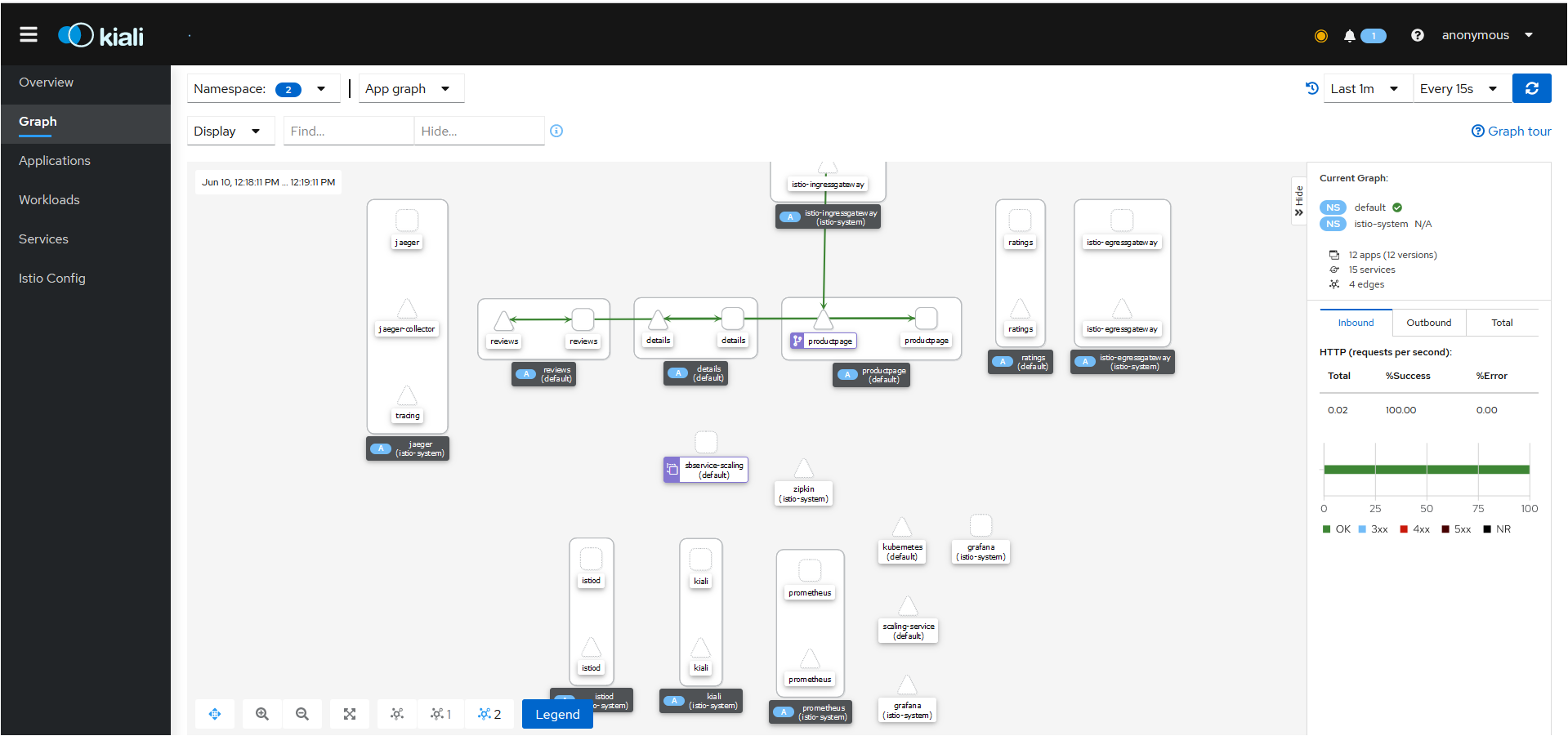
## Accessing Kiali Dashboard:

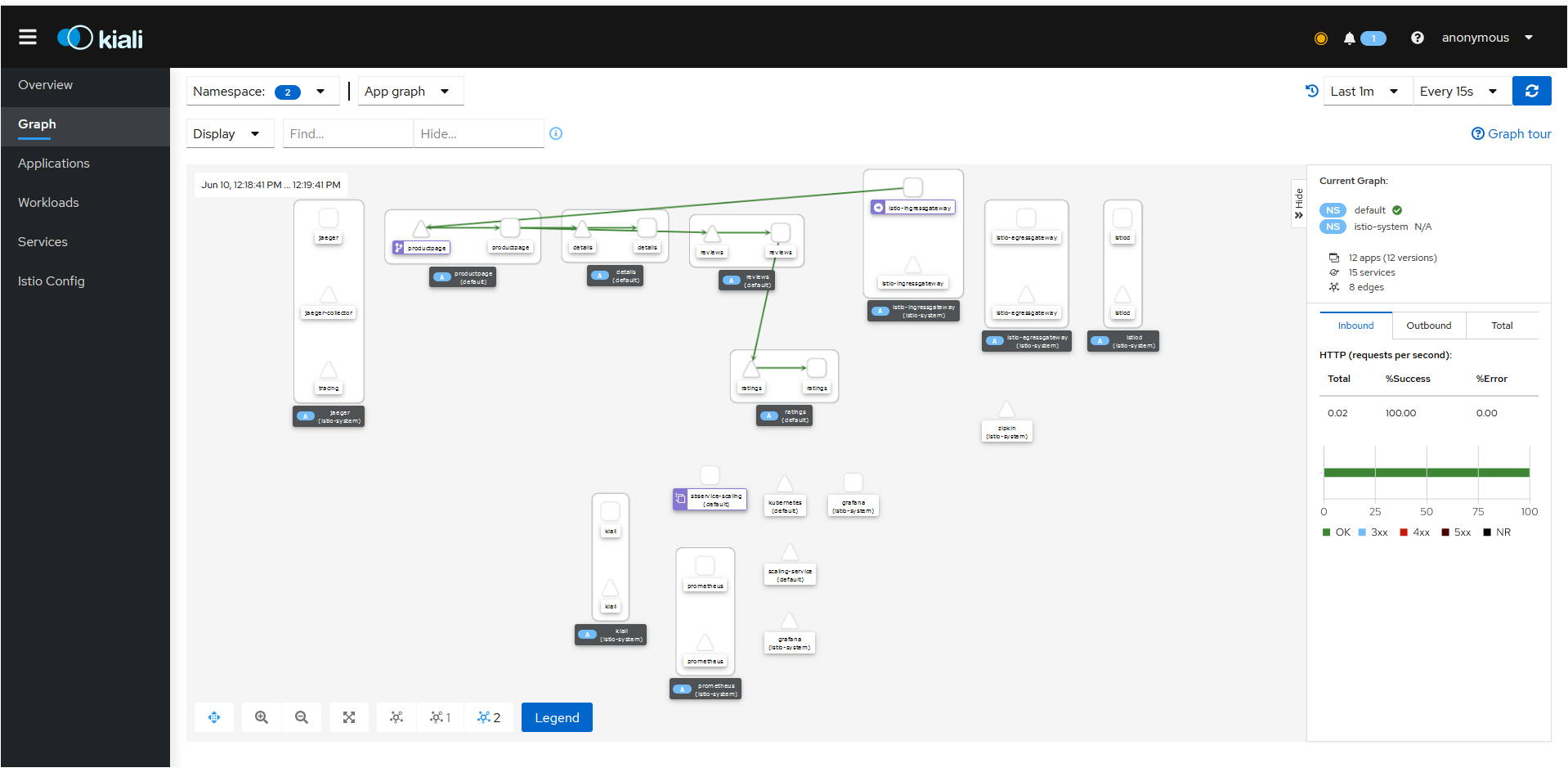
$ istioctl dashboard kiali



## Kiali Dashboard UI:



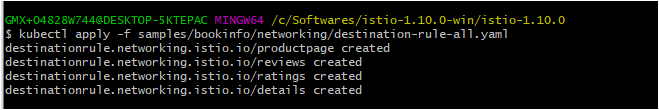




# Request Routing:

## Apply Default Destination Rules:

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

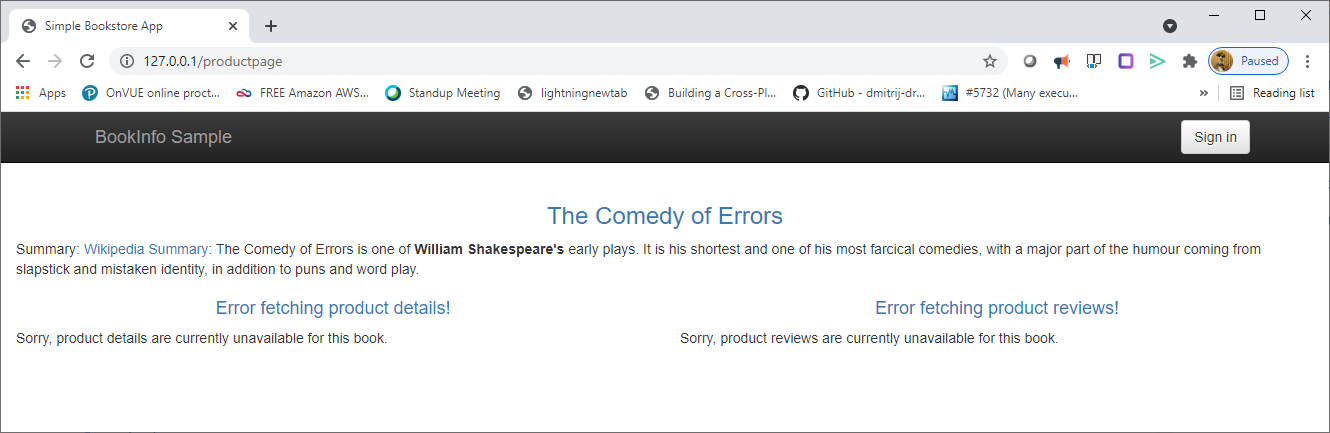


## Apply Virtual Service (All Requests Routes to V1):

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

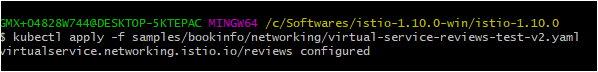
## 

## With Multiple Request Same Page Display

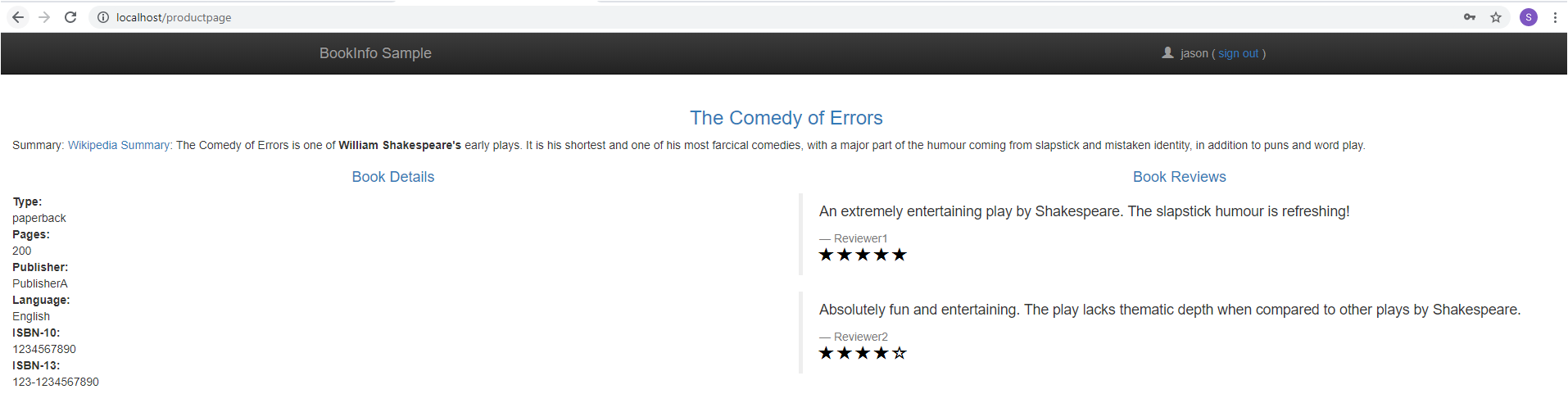


## Route User Based Identity:

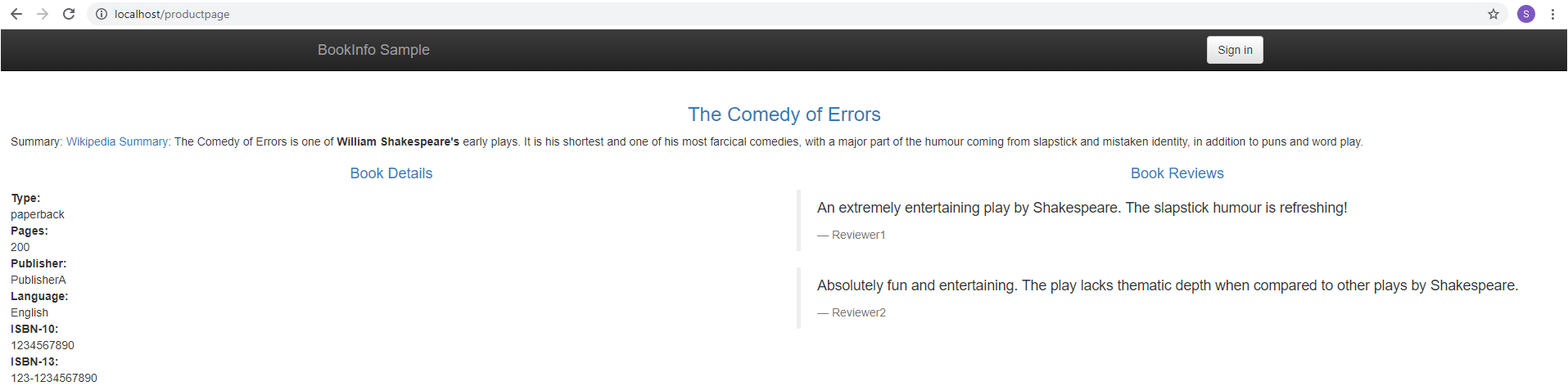
$ kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-test-v2.yaml



## Accessing Page (User:Jason Version-2 Displays)

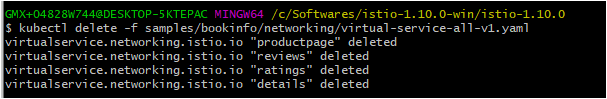


## Accessing Page (Default User. V1 Displays)



## Cleanup:

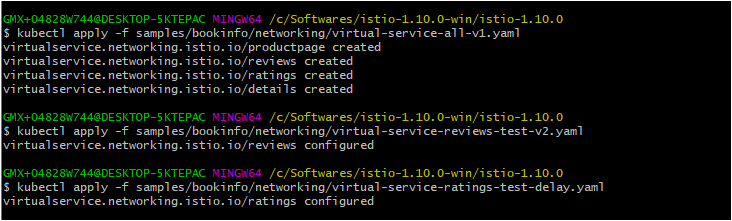
**$ kubectl delete -f samples/bookinfo/networking/virtual-service-all-v1.yaml**



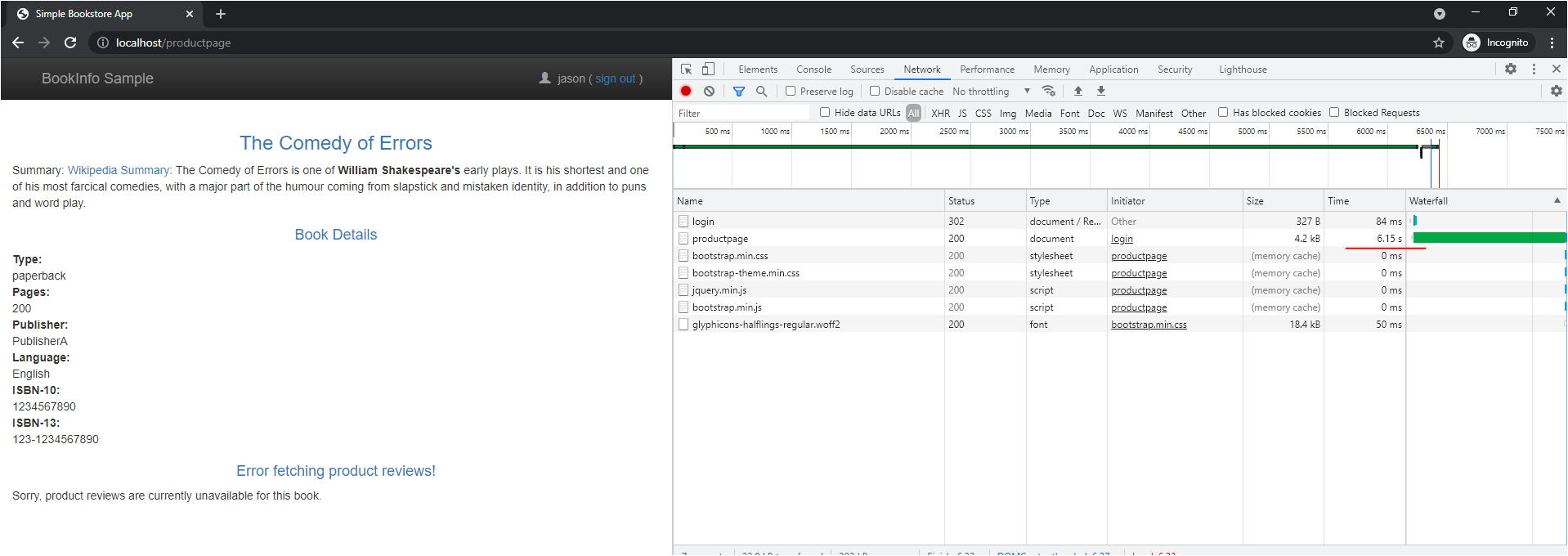
# Fault Injection:

## Applying Fault Injection

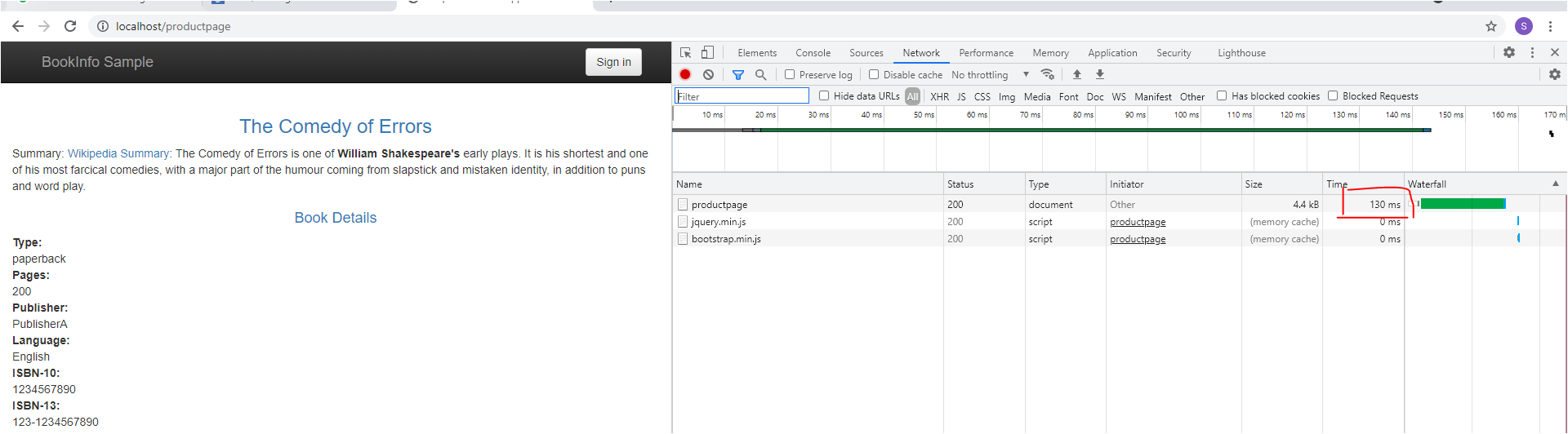
* **$ kubectl delete -f samples/bookinfo/networking/virtual-service-all-v1.yaml**
* kubectl apply -f samples/bookinfo/networking/virtual-service-all-v1.yaml
* kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-test-v2.yaml
* kubectl apply -f samples/bookinfo/networking/virtual-service-ratings-test-delay.yaml



## User-1 Accessing (Delaying 6 Seconds)

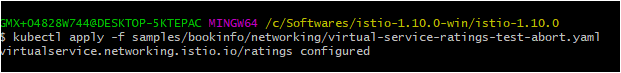


## Default User Accessing (No Delay)



## Aborting Injection of HTTP fault

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



## All request no delay

