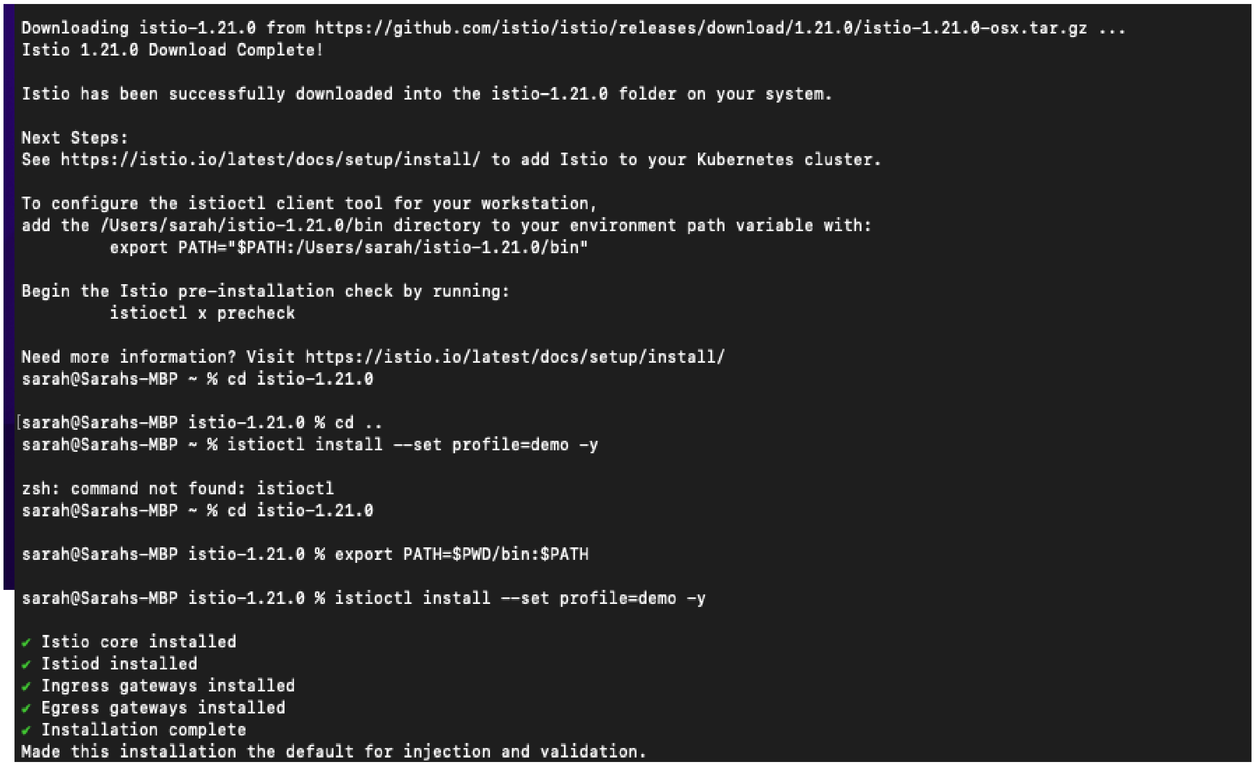
# Lab 5  (Istio Setup)

By Sarah Martin
Student ID: C00257967

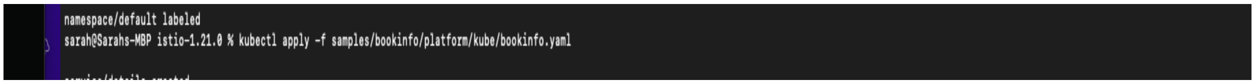
# Download Istio

I began with installing the Istio release and added the istioctl client to my path (MacOs). I then installed the demo configuration profile.



```
Downloading istio-1.21.0 from https://github.com/istio/istio/releases/download/1.21.0/istio-1.21.0-osx.tar.gz ...
Istio 1.21.0 Download Complete!

Istio has been successfully downloaded into the istio-1.21.0 folder on your system.

Next Steps:
See https://istio.io/latest/docs/setup/install/ to add Istio to your Kubernetes cluster.

To configure the istioctl client tool for your workstation,
add the /Users/sarah/istio-1.21.0/bin directory to your environment path variable with:
         export PATH="$PATH:/Users/sarah/istio-1.21.0/bin"

Begin the Istio pre-installation check by running:
         istioctl x precheck

Need more information? Visit https://istio.io/latest/docs/setup/install/
sarah@Sarahs-MBP ~ % cd istio-1.21.0

sarah@Sarahs-MBP istio-1.21.0 % cd ..
sarah@Sarahs-MBP ~ % istioctl install --set profile=demo -y

zsh: command not found: istioctl
sarah@Sarahs-MBP ~ % cd istio-1.21.0

sarah@Sarahs-MBP istio-1.21.0 % export PATH=$PWD/bin:$PATH

sarah@Sarahs-MBP istio-1.21.0 % istioctl install --set profile=demo -y

✔ Istio core installed
✔ Istiod installed
✔ Ingress gateways installed
✔ Egress gateways installed
✔ Installation complete
Made this installation the default for injection and validation.
```

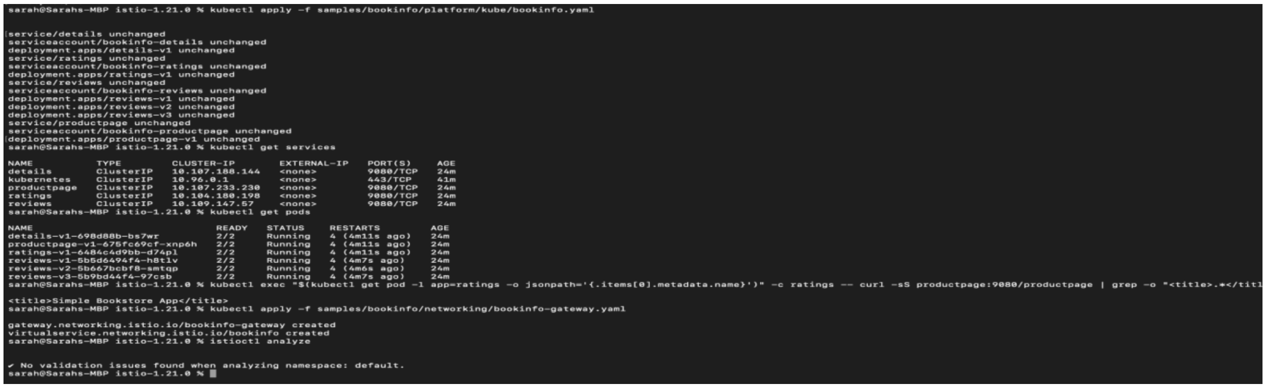
I added a namespace label to Istio.



```
namespace/default labeled
sarah@Sarahs-MBP istio-1.21.0 % kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml

service/details created
```

# Deploy the sample application

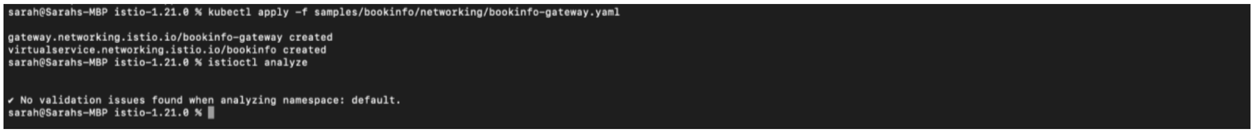
I then deployed the Bookinfo sample application. I did a get services command and a get pods command. I then verified that everything was working correctly by running the Kubectl exec command.



```
sarah@Sarahs-MBP istio-1.21.0 % kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml


service/details unchanged
serviceaccount/bookinfo-details unchanged
deployment.apps/details-v1 unchanged
service/ratings unchanged
serviceaccount/bookinfo-ratings unchanged
deployment.apps/ratings-v1 unchanged
service/reviews unchanged
serviceaccount/bookinfo-reviews unchanged
deployment.apps/reviews-v1 unchanged
deployment.apps/reviews-v2 unchanged
deployment.apps/reviews-v3 unchanged
service/productpage unchanged
serviceaccount/bookinfo-productpage unchanged
deployment.apps/productpage-v1 unchanged
sarah@Sarahs-MBP istio-1.21.0 % kubectl get services

NAME          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
details       ClusterIP   10.107.188.144   <none>        9080/TCP   24m
kubernetes    ClusterIP   10.96.0.1        <none>        443/TCP    41m
productpage   ClusterIP   10.107.233.230   <none>        9080/TCP   24m
ratings       ClusterIP   10.104.180.198   <none>        9080/TCP   24m
reviews       ClusterIP   10.109.147.57    <none>        9080/TCP   24m
sarah@Sarahs-MBP istio-1.21.0 % kubectl get pods

NAME                             READY   STATUS    RESTARTS        AGE
details-v1-698d88b-bs7wr         2/2     Running   4 (4m11s ago)   24m
productpage-v1-675fc69cf-xnp6h   2/2     Running   4 (4m11s ago)   24m
ratings-v1-6484c4d9bb-d74pl      2/2     Running   4 (4m11s ago)   24m
reviews-v1-5b5d6494f4-h8tlv      2/2     Running   4 (4m7s ago)    24m
reviews-v2-5b667bcbf8-smtqp      2/2     Running   4 (4m6s ago)    24m
reviews-v3-5b9bd44f4-97csb       2/2     Running   4 (4m7s ago)    24m
sarah@Sarahs-MBP istio-1.21.0 % kubectl exec "$(kubectl get pod -l app=ratings -o jsonpath='{.items[0].metadata.name}')" -c ratings -- curl -sS productpage:9080/productpage | grep -o "<title>.*</titl

<title>Simple Bookstore App</title>
sarah@Sarahs-MBP istio-1.21.0 % kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml

gateway.networking.istio.io/bookinfo-gateway created
virtualservice.networking.istio.io/bookinfo created
sarah@Sarahs-MBP istio-1.21.0 % istioctl analyze


✔ No validation issues found when analyzing namespace: default.
sarah@Sarahs-MBP istio-1.21.0 %
```

# Open the Application to outside traffic

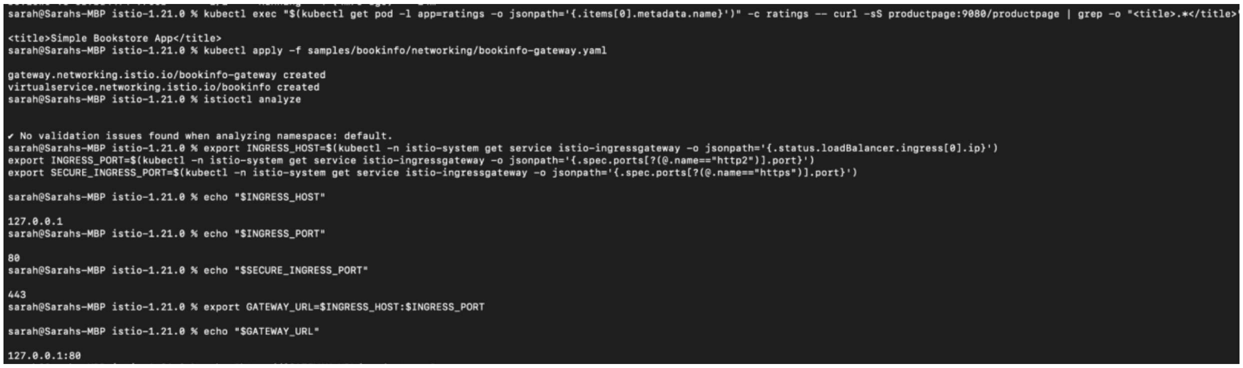
I associated the application with the Istio gateway. I then did the “istioctl analyze” command to make sure there was no configuration issues.



```
sarah@Sarahs-MBP istio-1.21.0 % kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml

gateway.networking.istio.io/bookinfo-gateway created
virtualservice.networking.istio.io/bookinfo created
sarah@Sarahs-MBP istio-1.21.0 % istioctl analyze


✔ No validation issues found when analyzing namespace: default.
sarah@Sarahs-MBP istio-1.21.0 %
```

# Determine the ingress IP and ports

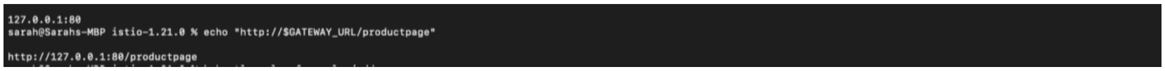
I opened up a new terminal and ran the “minikube tunnel” command so it sends traffic to the Istio Ingress Gateway.  I then set the ingress host and ports and then ensured that they were successfully assigned to each environment variable. I also set the Gateway_URL.



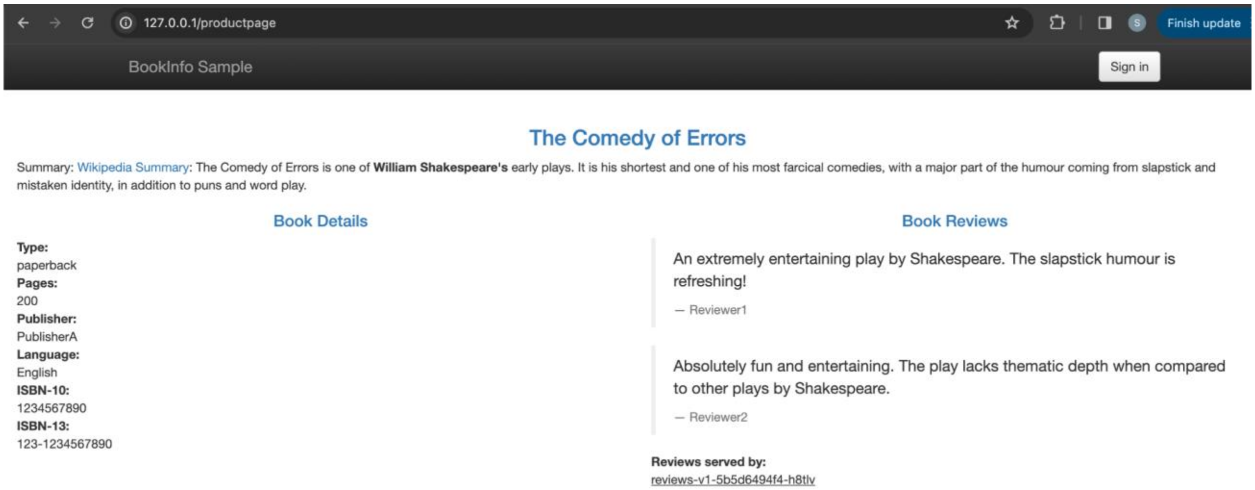
```
sarah@Sarahs-MBP istio-1.21.0 % kubectl exec "$(kubectl get pod -l app=ratings -o jsonpath='{.items[0].metadata.name}')" -c ratings -- curl -sS productpage:9080/productpage | grep -o "<title>.*</title>"

<title>Simple Bookstore App</title>
sarah@Sarahs-MBP istio-1.21.0 % kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml

gateway.networking.istio.io/bookinfo-gateway created
virtualservice.networking.istio.io/bookinfo created
sarah@Sarahs-MBP istio-1.21.0 % istioctl analyze


✔ No validation issues found when analyzing namespace: default.
sarah@Sarahs-MBP istio-1.21.0 % 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}')

sarah@Sarahs-MBP istio-1.21.0 % echo "$INGRESS_HOST"

127.0.0.1
sarah@Sarahs-MBP istio-1.21.0 % echo "$INGRESS_PORT"

80
sarah@Sarahs-MBP istio-1.21.0 % echo "$SECURE_INGRESS_PORT"

443
sarah@Sarahs-MBP istio-1.21.0 % export GATEWAY_URL=$INGRESS_HOST:$INGRESS_PORT

sarah@Sarahs-MBP istio-1.21.0 % echo "$GATEWAY_URL"

127.0.0.1:80
```

# Verify external Access

I ran the following command to get the external address of the Bookinfo application and pasted the url into my web browser.



```
127.0.0.1:80
sarah@Sarahs-MBP istio-1.21.0 % echo "http://$GATEWAY_URL/productpage"

http://127.0.0.1:80/productpage
```



The Comedy of Errors

Summary: Wikipedia Summary: The Comedy of Errors is one of **William Shakespeare's** early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.

Book Details

**Type:**
paperback
**Pages:**
200
**Publisher:**
PublisherA
**Language:**
English
**ISBN-10:**
1234567890
**ISBN-13:**
123-1234567890

Book Reviews

An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!

— Reviewer1

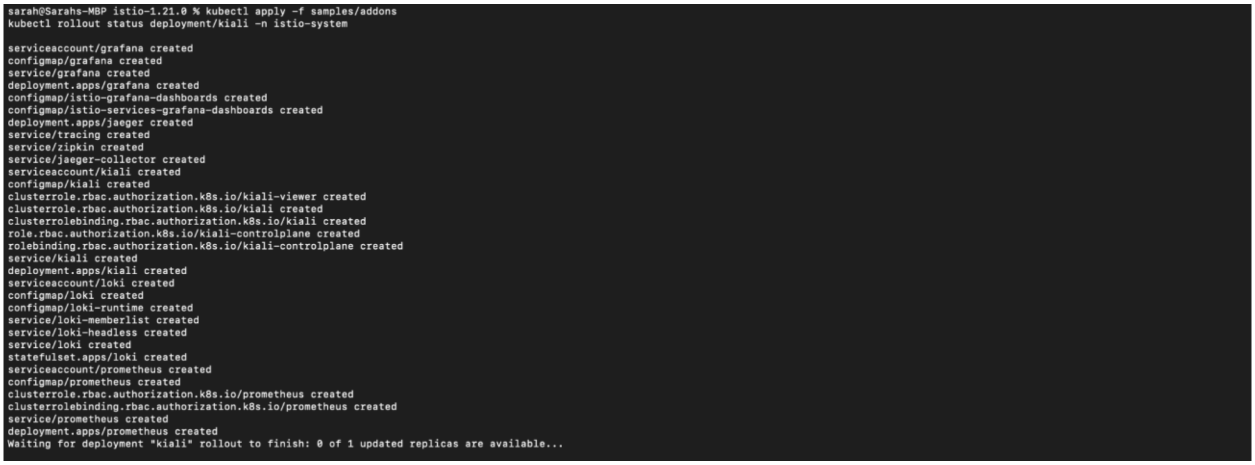
Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.

— Reviewer2

**Reviews served by:**
reviews-v1-5b5d6494f4-h8tlv

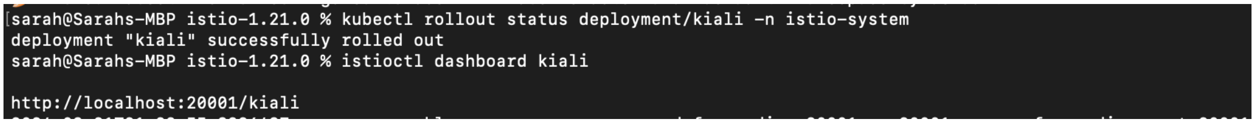
# View the dashboard

I installed the Kiall and other addons and waited for them to be deployed.

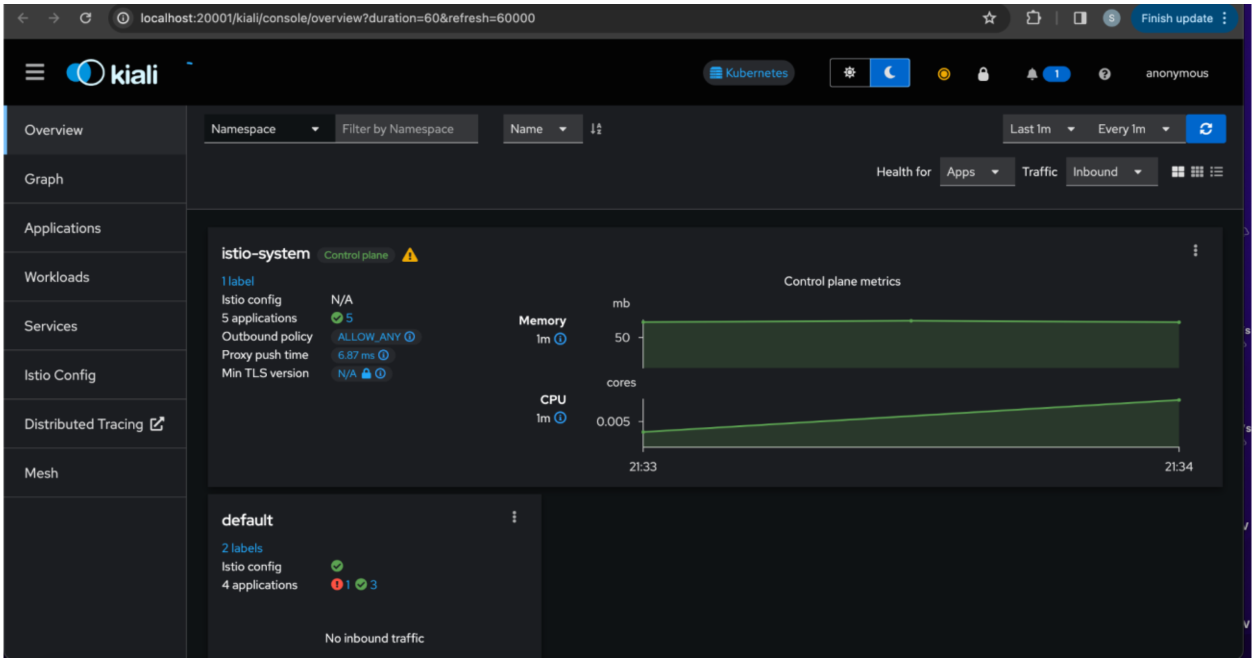


```
sarah@Sarahs-MBP istio-1.21.0 % kubectl apply -f samples/addons
kubectl rollout status deployment/kiali -n istio-system

serviceaccount/grafana created
configmap/grafana created
service/grafana created
deployment.apps/grafana created
configmap/istio-grafana-dashboards created
configmap/istio-services-grafana-dashboards created
deployment.apps/jaeger created
service/tracing created
service/zipkin created
service/jaeger-collector created
serviceaccount/kiali created
configmap/kiali created
clusterrole.rbac.authorization.k8s.io/kiali-viewer created
clusterrole.rbac.authorization.k8s.io/kiali created
clusterrolebinding.rbac.authorization.k8s.io/kiali created
role.rbac.authorization.k8s.io/kiali-controlplane created
rolebinding.rbac.authorization.k8s.io/kiali-controlplane created
service/kiali created
deployment.apps/kiali created
serviceaccount/loki created
configmap/loki created
configmap/loki-runtime created
service/loki-memberlist created
service/loki-headless created
service/loki created
statefulset.apps/loki created
serviceaccount/prometheus created
configmap/prometheus created
clusterrole.rbac.authorization.k8s.io/prometheus created
clusterrolebinding.rbac.authorization.k8s.io/prometheus created
service/prometheus created
deployment.apps/prometheus created
Waiting for deployment "kiali" rollout to finish: 0 of 1 updated replicas are available...
```

I then accessed the dashboard to view it.



```
sarah@Sarahs-MBP istio-1.21.0 % kubectl rollout status deployment/kiali -n istio-system
deployment "kiali" successfully rolled out
sarah@Sarahs-MBP istio-1.21.0 % istioctl dashboard kiali

http://localhost:20001/kiali
```



Health for Apps Traffic Inbound

istio-system Control plane

1 label
Istio config N/A
5 applications ✓ 5
Outbound policy ALLOW_ANY
Proxy push time 6.87 ms
Min TLS version N/A

Memory 1m

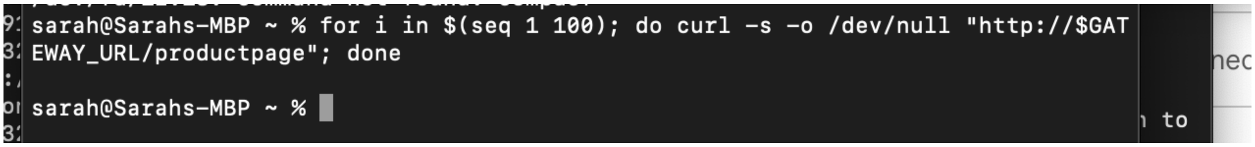
CPU 1m

default

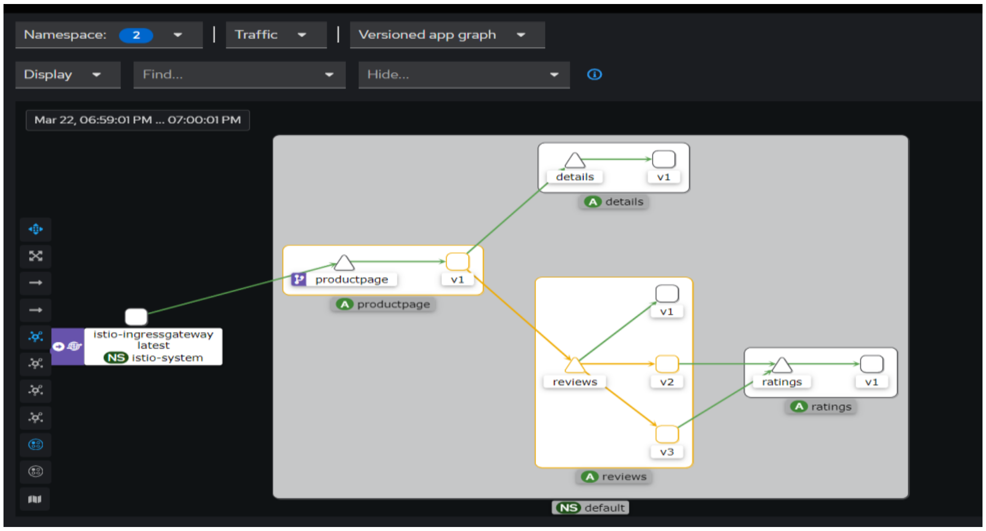
2 labels
Istio config ✓
4 applications ❗1 ✓3

No inbound traffic

I ran this command to view the graph.



```
sarah@Sarahs-MBP ~ % for i in $(seq 1 100); do curl -s -o /dev/null "http://$GAT
EWAY_URL/productpage"; done

sarah@Sarahs-MBP ~ % 
```



Namespace: 2 Traffic Versioned app graph

Mar 22, 06:59:01 PM ... 07:00:01 PM