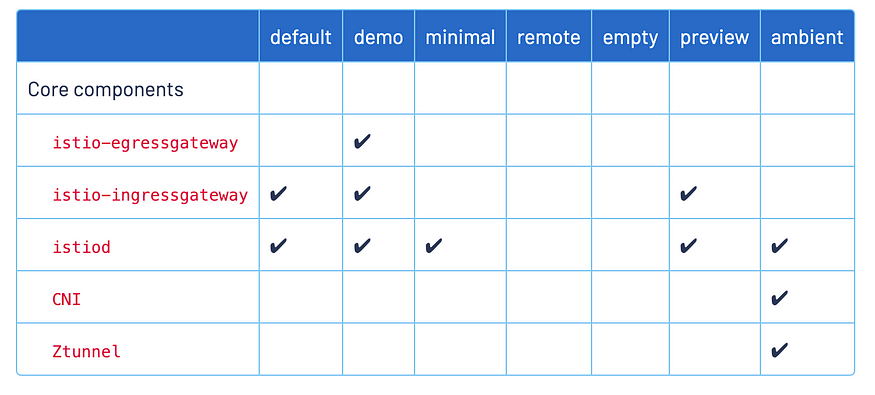
Follow these steps to get started with Istio:

1. Download and install Istio using istioctl with demo profile
2. Verify istio installation, version 1.20.1
3. Deploy the sample application, a bookinfoapplication
4. Deploy gateway, open the sample application to outside traffic
5. View the dashboard, [Kiali](https://istio.io/latest/docs/ops/integrations/kiali/)
6. Visualizing Metrics with [Grafana](https://istio.io/latest/docs/ops/integrations/grafana/#option-1-quick-start)

**Prerequisites:**

1. [Download and Install the Istio release](https://istio.io/latest/docs/setup/getting-started/#download). In this guide, I am using the version client version: 1.20.1, control plane version: 1.20.1
2. A Kubernetes cluster, in this tutorial, I am using an [AKS cluster](https://learn.microsoft.com/en-us/azure/aks/learn/quick-kubernetes-deploy-portal?tabs=azure-cli) server version: v1.27.7
3. Client tool, [kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl-macos/) client version v1.26.1

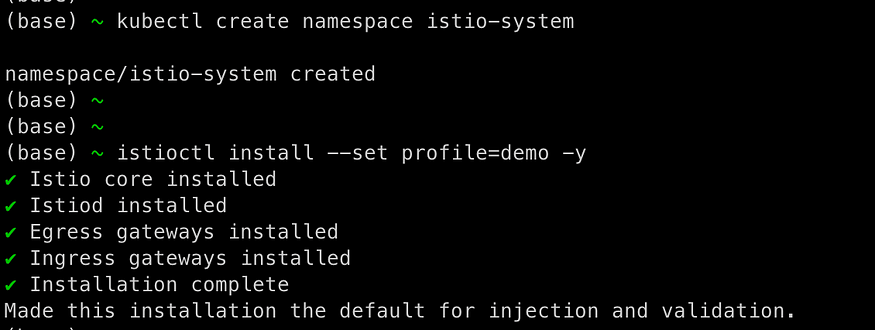
Many built-in configuration profiles can be used when [installing Istio](https://istio.io/latest/docs/setup/install/istioctl/). Each profile contains a different istio components see details below:



**Step 1: Install**istio**with demo profile**

I am using macOS. Open a terminal, and run the following command to install istio:

kubectl create namespace istio-system  
  
istioctl install --set profile=demo -y  
  
kubectl label namespace default istio-injection=enabled



**Step 2: Verify Installation**

Run the following command to verify the installation, you can see the output message to confirm the installation is successful.

istioctl verify-install



**Step 3: Deploy the sample application**

**Deploy the**Bookinfo sample application istio/release-1.20by command :

kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.20/samples/bookinfo/platform/kube/bookinfo.yaml  
  
service/details created  
serviceaccount/bookinfo-details created  
deployment.apps/details-v1 created  
service/ratings created  
serviceaccount/bookinfo-ratings created  
deployment.apps/ratings-v1 created  
service/reviews created  
serviceaccount/bookinfo-reviews created  
deployment.apps/reviews-v1 created  
deployment.apps/reviews-v2 created  
deployment.apps/reviews-v3 created  
service/productpage created  
serviceaccount/bookinfo-productpage created  
deployment.apps/productpage-v1 created

**Verify** sample application deployment:

kubectl get services  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
details ClusterIP 10.0.85.238 <none> 9080/TCP 61s  
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 2d9h  
productpage ClusterIP 10.0.96.182 <none> 9080/TCP 60s  
ratings ClusterIP 10.0.116.34 <none> 9080/TCP 60s  
reviews ClusterIP 10.0.149.80 <none> 9080/TCP 60s  
  
kubectl get pods  
NAME READY STATUS RESTARTS AGE  
details-v1-5f4d584748-f96sv 2/2 Running 0 45s  
productpage-v1-564d4686f-czzl4 2/2 Running 0 44s  
ratings-v1-686ccfb5d8-6pf6d 2/2 Running 0 45s  
reviews-v1-86896b7648-l76tw 2/2 Running 0 45s  
reviews-v2-b7dcd98fb-bkkdf 2/2 Running 0 45s  
reviews-v3-5c5cc7b6d-zzlzq 2/2 Running 0 45s

Verify everything is working correctly up to this point. Run this command to see if the app is running inside the cluster and serving HTML pages by checking for the page title in the response:

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>

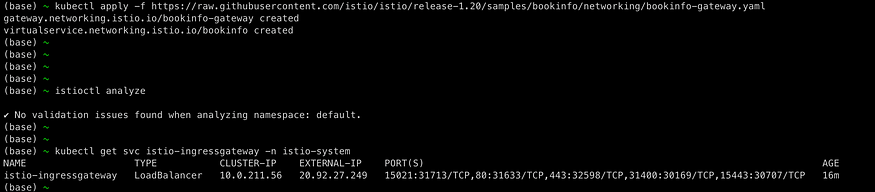
**Step 4: Deploy the Virtual Service, Gateway, and Open the application to outside traffic.**

Associate this application with the Istio gateway

kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.20/samples/bookinfo/networking/bookinfo-gateway.yaml

Ensure that there are no issues with the configuration:

(base) ~ istioctl analyze  
  
✔ No validation issues found when analyzing namespace: default.



Determining the ingress IP and ports

kubectl get svc istio-ingressgateway -n istio-system

Set the ingress IP and ports

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}')

Set GATEWAY\_URL

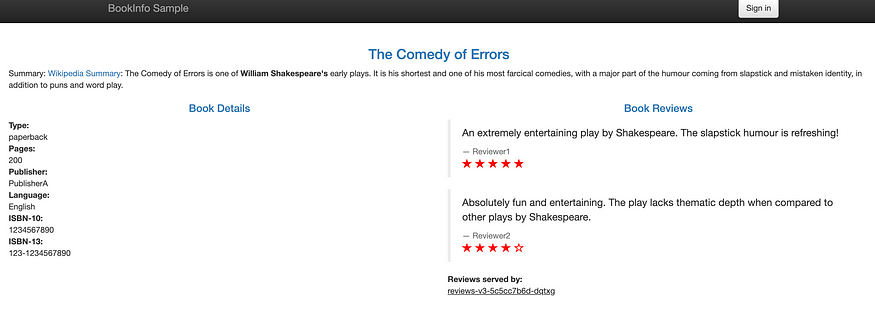
export GATEWAY\_URL=$INGRESS\_HOST:$INGRESS\_PORT

Ensure an IP address and port were successfully assigned to the environment variable:

echo "$GATEWAY\_URL"

Verify external access, Confirm that the Bookinfo application is accessible from outside by viewing the Bookinfo product page using a browser.

echo "http://$GATEWAY\_URL/productpage"



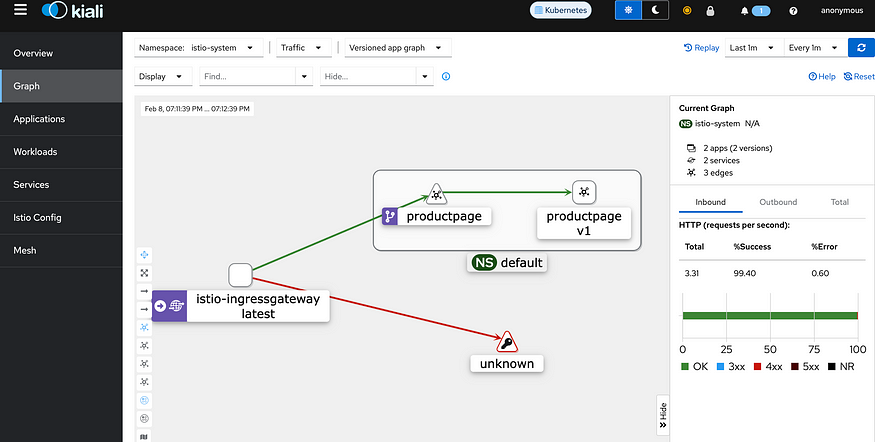
**Step 5: View the Kiali dashboard**

1. **Install** Kiali**,**Prometheus**,**and the other add-ons**and wait for them to be deployed.**

kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.20/samples/addons/kiali.yaml  
kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.20/samples/addons/prometheus.yaml  
kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.20/samples/addons/grafana.yaml

2. Access the Kiali dashboard.

istioctl dashboard kiali



**To see trace data, you must send requests to your service. Run the following script to send some requests:**

for i in $(seq 1 100); do curl -s -o /dev/null "http://$GATEWAY\_URL/productpage"; done;

**Step 6: Visualizing Metrics with Grafana**

Run the command to run the Grafana dashboard. Then Visit**the mesh dashboard** <http://localhost:3000/d/G8wLrJIZk/istio-mesh-dashboard> in your web browser.

istioctl dashboard grafana

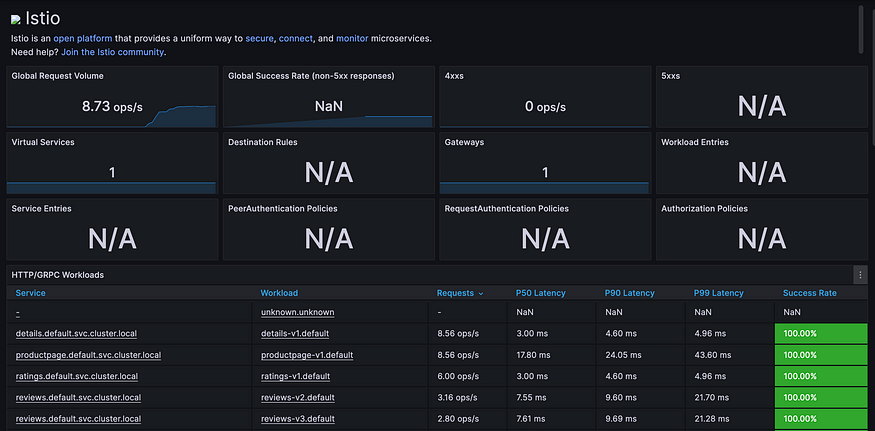
Send traffic to the mesh.

For the Bookinfo sample, visit http://$GATEWAY\_URL/productpage your web browser or issue the following command:

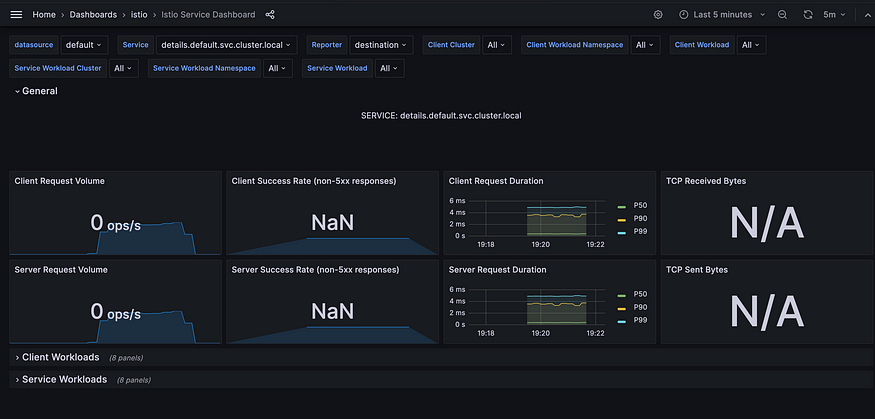
To see trace data, you must send requests to your service. The number of requests depends on Istio’s sampling rate and can be configured using the [Telemetry API](https://istio.io/latest/docs/tasks/observability/telemetry/). With the default sampling rate of 1%, you need to send at least 100 requests before the first trace is visible. To send 100 requests to the productpage service, use the following command:

$ for i in $(seq 1 100); do curl -s -o /dev/null "http://$GATEWAY\_URL/productpage"; done

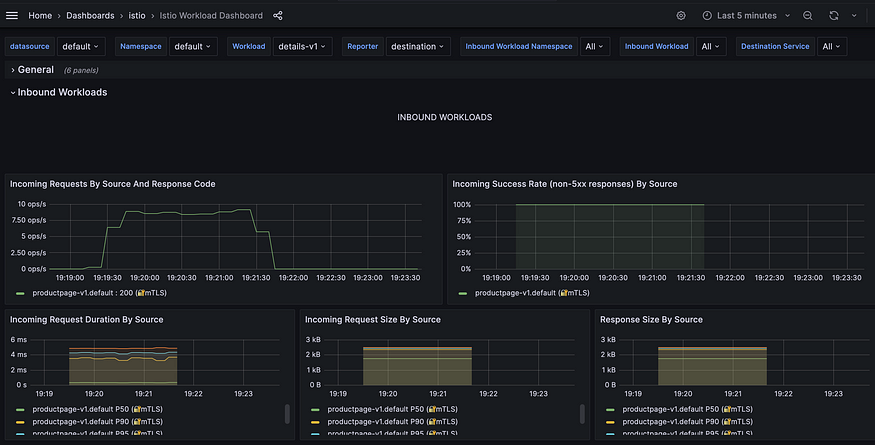
The Istio Dashboard will look similar to:



**Visualize Service Dashboards**: <http://localhost:3000/d/LJ_uJAvmk/istio-service-dashboard?orgId=1&refresh=1m>



**Visualize Workload Dashboards:**



**Cleanup**

if you are not planning any follow-up tasks, remove the Bookinfo sample application and Kiali from your cluster.

1. To remove the Bookinfo application, refer to the [Bookinfo cleanup](https://istio.io/latest/docs/examples/bookinfo/#cleanup) instructions.
2. To remove Kiali from a Kubernetes environment:

kubectl delete -f https://raw.githubusercontent.com/istio/istio/release-1.20/samples/addons/kiali.yaml

**Uninstall Istio**

To completely uninstall Istio from a cluster, run the following command:

istioctl uninstall --purge

The optional --purge flag will remove all Istio resources, including cluster-scoped resources that may be shared with other Istio control planes.

**(Optional) Deleting CRDs installed by Istio**

Deleting CRDs permanently removes any Istio resources you have created in your cluster. To delete Istio CRDs installed in your cluster:

kubectl get crd -oname | grep --color=never 'istio.io' | xargs kubectl delete