Documenting: Getting Started with Istio

Setup:

Starting Minikube with extra resources for Istio

```
PS C:\Users\c_bro> minikube start --cpus 4 --memory 4096
W0506 16:39:44.863913    2348 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\c_bro\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.
😄  minikube v1.32.0 on Microsoft Windows 11 Home 10.0.22631.3447 Build 22631.3447
✨  Using the docker driver based on existing profile
❗  You cannot change the memory size for an existing minikube cluster. Please first delete the cluster.
❗  You cannot change the CPUs for an existing minikube cluster. Please first delete the cluster.
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
```

Downloading Istio

```
PS C:\Users\c_bro\Downloads> cd .\istio-1.21.2\
PS C:\Users\c_bro\Downloads\istio-1.21.2> pwd

Path
----
C:\Users\c_bro\Downloads\istio-1.21.2
```

Setting up an Istio path variable:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> $env:PATH += ";C:\Users\c_bro\Downloads\istio-1.21.2\bin"
```

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> istioctl
Istio configuration command line utility for service operators to
debug and diagnose their Istio mesh.

Usage:
  istioctl [command]

Available Commands:
  admin                Manage control plane (istiod) configuration
  analyze              Analyze Istio configuration and print validation messages
  authz                (authz is experimental. Use `istioctl experimental authz`)
  bug-report           Cluster information and log capture support tool.
  completion           Generate the autocompletion script for the specified shell
  create-remote-secret Create a secret with credentials to allow Istio to access remote Kubernetes apiservers
  dashboard            Access to Istio web UIs
  experimental         Experimental commands that may be modified or deprecated
  help                 Help about any command
  install              Applies an Istio manifest, installing or reconfiguring Istio on a cluster.
  kube-inject          Inject Istio sidecar into Kubernetes pod resources
  manifest             Commands related to Istio manifests
  operator             Commands related to Istio operator controller.
  profile              Commands related to Istio configuration profiles
  proxy-config         Retrieve information about proxy configuration from Envoy [kube only]
  proxy-status         Retrieves the synchronization status of each Envoy in the mesh [kube only]
  remote-clusters      Lists the remote clusters each istiod instance is connected to.
  tag                  Command group used to interact with revision tags
  uninstall            Uninstall Istio from a cluster
```

Installing Istio:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> istioctl install --set profile=demo -y
✔Istio core installed
✔Istiod installed
✔Egress gateways installed
✔Ingress gateways installed
✔Installation complete
ade this installation the default for injection and validation.
```

Adding a namespace label:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl label namespace default istio-injection=enabled
namespace/default labeled
```

Deploying the Bookinfo sample application

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl apply -f 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
```

Getting the services:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl get services
NAME          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
details       ClusterIP   10.106.42.144    <none>        9080/TCP   76s
kubernetes    ClusterIP   10.96.0.1        <none>        443/TCP    78d
productpage   ClusterIP   10.111.194.25    <none>        9080/TCP   75s
ratings       ClusterIP   10.110.28.173    <none>        9080/TCP   76s
reviews       ClusterIP   10.101.233.100   <none>        9080/TCP   76s
```

Getting the Pods:

Each pod becomes ready, with an Istio sidecar deployed along with it.

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl get pods
NAME                              READY   STATUS            RESTARTS      AGE
details-v1-698d88b-gcqhs          2/2     Running           0             2m6s
hello-node-ccf4b9788-pgkjk        1/1     Running           3 (80m ago)   78d
productpage-v1-675fc69cf-mksln    0/2     PodInitializing   0             2m5s
ratings-v1-6484c4d9bb-xpr5f       2/2     Running           0             2m6s
reviews-v1-5b5d6494f4-7m5w8       2/2     Running           0             2m6s
reviews-v2-5b667bcbf8-64cfm       2/2     Running           0             2m5s
reviews-v3-5b9bd44f4-d2z4x        0/2     PodInitializing   0             2m5s
```

Verifying everything is working correctly by running the command below to see if the app is running inside the cluster and serving HTML pages by checking for the page title in the response:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl exec (kubectl get pod -l app=ratings -o jsonpath="
{.items[0].metadata.name}") -c ratings -- curl -sS productpage:9080/productpage | Select-String -Pat
tern "<title>.*</title>"

    <title>Simple Bookstore App</title>
```

Opening the application to outside traffic

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl apply -f samples/bookinfo/networking/bookinfo-gate
way.yaml
gateway.networking.istio.io/bookinfo-gateway created
virtualservice.networking.istio.io/bookinfo created
```

Ensuring that there are no issues with the configuration:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> istioctl analyze
Warning [IST0103] (Pod default/hello-node-ccf4b9788-pgkjk) The pod default/hello-node-ccf4b9788-pgkj
k is missing the Istio proxy. This can often be resolved by restarting or redeploying the workload.
```

Fixing the Issue with the old Kubernetes pod:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl delete pod hello-node-ccf4b9788-pgkjk -n default
pod "hello-node-ccf4b9788-pgkjk" deleted
```

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl get pod -n default
NAME                              READY   STATUS    RESTARTS   AGE
details-v1-698d88b-gcqhs          2/2     Running   0          11m
hello-node-ccf4b9788-fk5gd        2/2     Running   0          41s
productpage-v1-675fc69cf-mksln    2/2     Running   0          11m
ratings-v1-6484c4d9bb-xpr5f       2/2     Running   0          11m
reviews-v1-5b5d6494f4-7m5w8       2/2     Running   0          11m
reviews-v2-5b667bcbf8-64cfm       2/2     Running   0          11m
reviews-v3-5b9bd44f4-d2z4x        2/2     Running   0          11m
```

```
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  81s   default-scheduler  Successfully assigned default/hello-node-ccf4b9788-fk5
gd to minikube
```

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> istioctl analyze

✔No validation issues found when analyzing namespace: default.
```

Determining the Ingress IP and ports

Starting a Minikube tunnel to send traffic to Istio Ingress Gateway:

```
PS C:\Users\c_bro> minikube tunnel
W0506 18:13:40.171452   21072 main.go:291] Unable to resolve the current Docker CLI context "default
": context "default": context not found: open C:\Users\c_bro\.docker\contexts\meta\37a8eec1ce19687d1
32fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot find the path specified
.
✅  Tunnel successfully started

📌  NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be acc
essible ...

❗  Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more in
formation, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-window
s-requires-root-permission
🏃  Starting tunnel for service istio-ingressgateway.
```

Setting up the Ingress host and ports:

Ingress Host:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> $INGRESS_HOST = kubectl -n istio-system get service istio-
ingressgateway -o jsonpath="{.status.loadBalancer.ingress[0].ip}"
PS C:\Users\c_bro\Downloads\istio-1.21.2> $env:INGRESS_HOST = $INGRESS_HOST
```

Test:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> echo "$INGRESS_HOST"
127.0.0.1
```

Ingress Port:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> $INGRESS_PORT = kubectl -n istio-system get service istio-
ingressgateway -o jsonpath="{.spec.ports[?(@.name=='http2')].port}"
PS C:\Users\c_bro\Downloads\istio-1.21.2> $env:INGRESS_PORT = $INGRESS_PORT
```

Test:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> echo "$INGRESS_PORT"
80
```

Secure Ingress Port:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> $SECURE_INGRESS_PORT = kubectl -n istio-system get service
 istio-ingressgateway -o jsonpath="{.spec.ports[?(@.name=='https')].port}"
PS C:\Users\c_bro\Downloads\istio-1.21.2> $env:SECURE_INGRESS_PORT = $SECURE_INGRESS_PORT
```

Test:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> echo "$SECURE_INGRESS_PORT"
443
```

Setting Gateway_URL:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> $GATEWAY_URL = "$env:INGRESS_HOST:$env:INGRESS_PORT"
PS C:\Users\c_bro\Downloads\istio-1.21.2> $env:GATEWAY_URL = $GATEWAY_URL
```

Fixing Gateway:

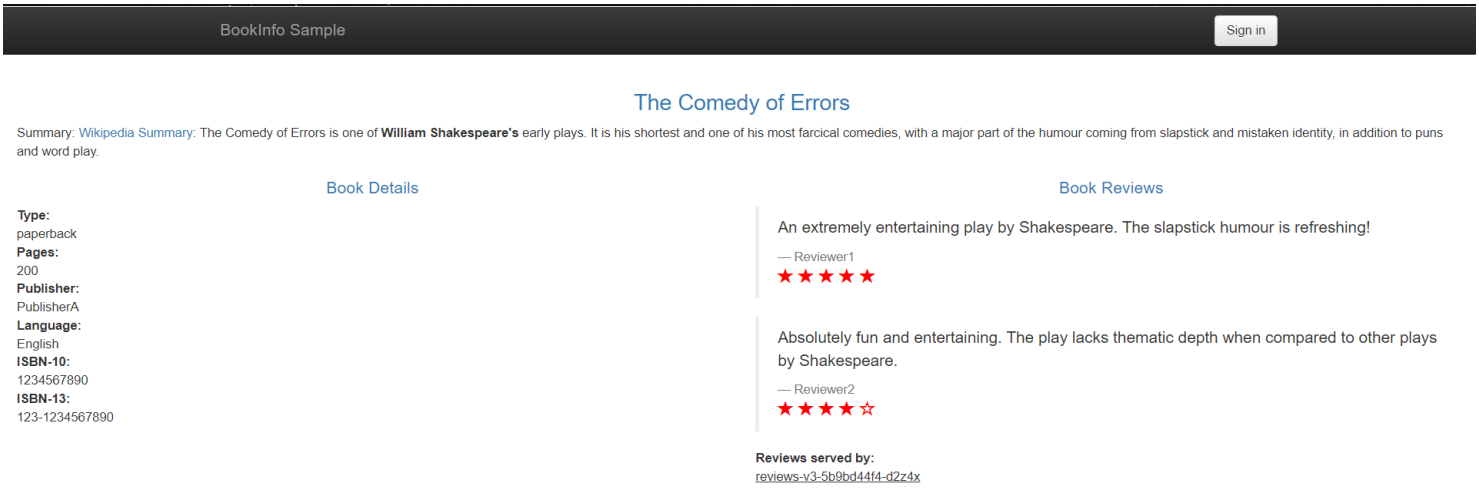
```
PS C:\Users\c_bro\Downloads\istio-1.21.2> $GATEWAY_URL = $env:INGRESS_HOST + ":" + $env:INGRESS_PORT
```

Re-Testing Gateway:

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> echo "$GATEWAY_URL"
127.0.0.1:80
```

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> echo "http://$GATEWAY_URL/productpage"
http://127.0.0.1:80/productpage
```

Verifying external access for Bookinfo:



Viewing the dashboard

Deploying the Kiali dashboard:

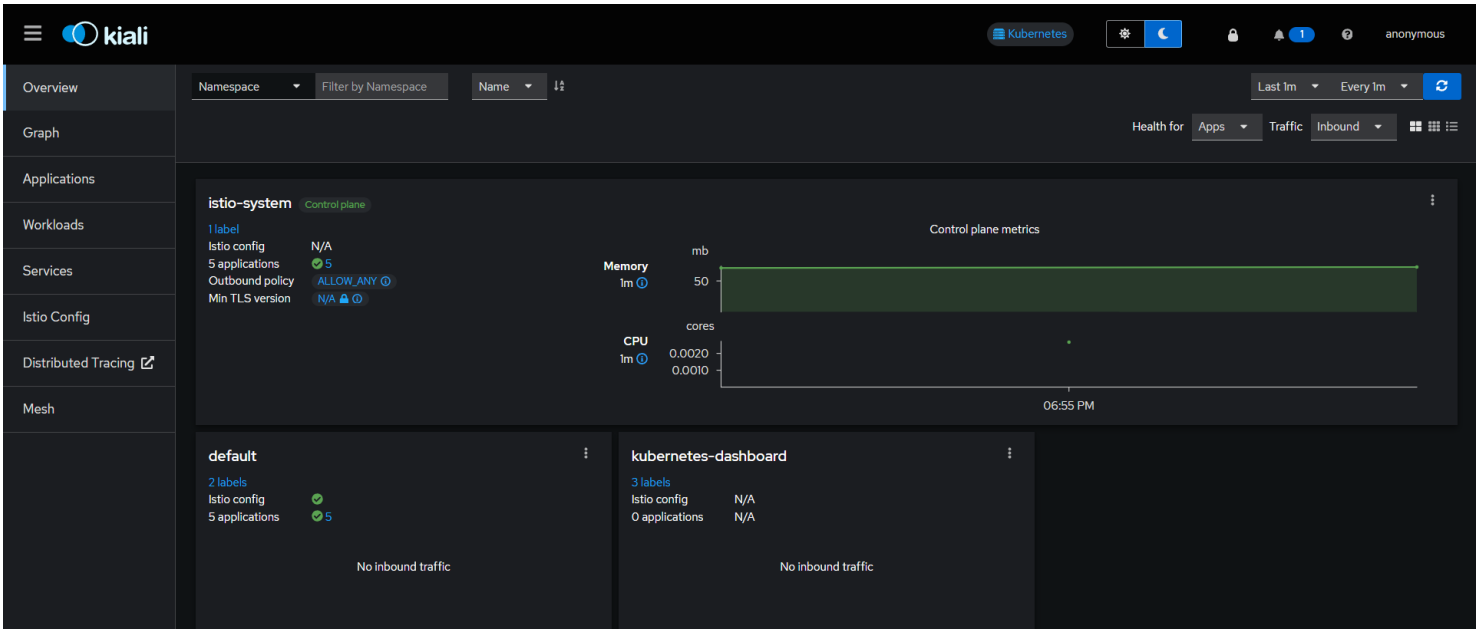
Applying the addons

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl apply -f samples/addons
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
```

Deploying Kiali

```
PS C:\Users\c_bro\Downloads\istio-1.21.2> kubectl rollout status deployment/kiali -n istio-system
deployment "kiali" successfully rolled out
```

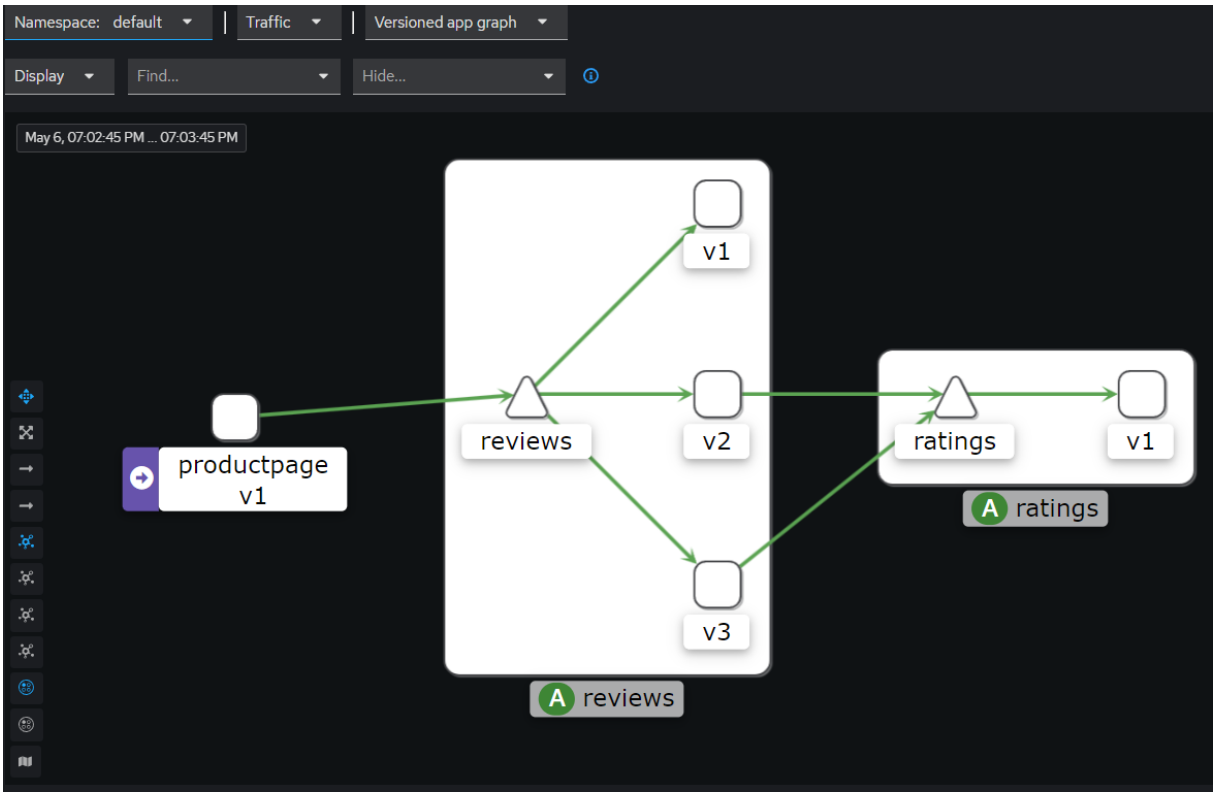
Accessing the Kiali dashboard:



Sending 100 requests to the product page service:

```
PS C:\Users\c_bro> foreach ($i in 1..100) {
>>     Invoke-WebRequest -Uri "http://$env:GATEWAY_URL/productpage" -UseBasicParsing | Out-Null
>> }
```

Kiali Graph:



In this tutorial from Istio's official documentation, I've successfully walked through the steps of getting Istio up and running on a Kubernetes cluster. I began by downloading

and installing Istio, then used the istioctl tool to deploy my first sample application with Istio's service mesh. This included setting up the Istio Ingress Gateway to manage external access to the applications in the mesh. I also explored how to enable and verify automatic sidecar injection, which ensures that Istio's data plane can control and monitor the traffic that flows between my services. Finally, I used istioctl to analyze and ensure that your service mesh configuration was free from errors, and tested the setup by sending requests to your services, showcasing how Istio handles traffic within the cluster.