Lab7

Download Istio

1. Go to the [Istio release](#) page to download the installation file:

istio-1.27.1-win.zip		2025/9/30 13:5
istioctl-1.27.1-win-amd64.zip		2025/9/30 13:4

2. Unzip:

bin	2025/9/3 21:41	文件夹
manifests	2025/9/3 21:41	文件夹
samples	2025/9/3 21:41	文件夹
tools	2025/9/3 21:41	文件夹
LICENSE	2025/9/3 21:41	文件
manifest.yaml	2025/9/3 21:41	YAML 文
README.md	2025/9/3 21:41	MD 文件

3. Start up Kubernets from minikube:

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1\bin> minikube start
😄 Microsoft Windows 11 Home China 10.0.26100.6584 Build 26100.6584 上的 minikube v1.37.0
✨ 根据现有的配置文件使用 docker 驱动程序
👍 在集群中 "minikube" 启动节点 "minikube" primary control-plane
🚜 正在拉取基础镜像 v0.0.48 ...
🔄 正在为"minikube"重启现有的 docker container ...
❗ 从 Minikube 的 container 内部连接到 https://registry.cn-hangzhou.aliyuncs.com/google_containers/ 失败
   要获取新的外部镜像，可能需要配置代理：https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🐳 正在 Docker 28.4.0 中准备 Kubernetes v1.34.0…
🔎 正在验证 Kubernetes 组件...
   ▪ 正在使用镜像 registry.cn-hangzhou.aliyuncs.com/google_containers/storage-provisioner:v5
   ▪ 正在使用镜像 docker.io/kubernetesui/metrics-scraper:v1.0.8
   ▪ 正在使用镜像 docker.io/kubernetesui/dashboard:v2.7.0
💡 某些仪表板功能需要 metrics-server 插件。要启用所有功能，请运行：

       minikube addons enable metrics-server

🌟 启用插件：default-storageclass, storage-provisioner, dashboard
❗ C:\Program Files\Docker\Docker\resources\bin\kubectl.exe 的版本为 1.32.2，可能与 Kubernetes 1.34.0 不兼容
。
   ▪ 想要使用 kubectl v1.34.0 吗? 尝试使用 'minikube kubectl -- get pods -A' 命令
🏄 完成！kubectl 现在已配置，默认使用"minikube"集群和"default"命名空间
```

Install Istio

1. Install Istio:

bin/istioctl install -f samples/bookinfo/demo-profile-no-gateways.yaml -y

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> bin/istioctl install -f samples/bookinfo/demo-profile-no-gatewa
ys.yaml -y
                |
               /|\
              / | \
             /  |  \
            /   |   \
           /||  |
          / ||  |
         /  ||  |
        /   ||  |
       /____||__|_____
      _____
        _____/
         _____/
          _____/

  ✔Istio core installed ⛵
  ✔Istiod installed 🧊
  ✔Installation complete
```

2. Add a namespace label to instruct Istio to automatically inject Envoy sidecar proxies
    when you deploy your application later:

    kubectl label namespace default istio-injection=enabled

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl label namespace default istio-injection=enabled
namespace/default labeled
```

Install the Kubernetes Gateway API CRDs

1. Install the Kubernetes Gateway API CRDs:

    kubectl get crd gateways.gateway.networking.k8s.io

    kubectl kustomize "github.com/kubernetes-sigs/gateway-api/config/crd?ref=v1.3.0"
    | kubectl apply -f -

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl kustomize "github.com/kubernetes-sigs/gateway-api/config/crd?ref=
v1.3.0" | kubectl apply -f -
Warning: unrecognized format "int64"
customresourcedefinition.apiextensions.k8s.io/gatewayclasses.gateway.networking.k8s.io created
Warning: unrecognized format "int32"
customresourcedefinition.apiextensions.k8s.io/gateways.gateway.networking.k8s.io created
customresourcedefinition.apiextensions.k8s.io/grpcroutes.gateway.networking.k8s.io created
customresourcedefinition.apiextensions.k8s.io/httproutes.gateway.networking.k8s.io created
customresourcedefinition.apiextensions.k8s.io/referencegrants.gateway.networking.k8s.io created
```

Deploy the sample application

1. Deploy the Bookinfo sample application:

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

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> 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
```

    check:

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl get services
NAME          TYPE          CLUSTER-IP       EXTERNAL-IP      PORT(S)          AGE
details       ClusterIP     10.111.234.8     <none>           9080/TCP         54s
hello-node    LoadBalancer  10.97.241.151    <pending>        8080:32635/TCP   12d
kubernetes    ClusterIP     10.96.0.1        <none>           443/TCP          12d
productpage   ClusterIP     10.98.235.106    <none>           9080/TCP         53s
ratings       ClusterIP     10.106.84.3      <none>           9080/TCP         54s
reviews       ClusterIP     10.105.253.186   <none>           9080/TCP         54s
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl get pods
NAME                         READY   STATUS          RESTARTS        AGE
details-v1-77d6bd5675-n4zrp  1/2     PodInitializing 0               62s
hello-node-6c9b5f4b59-rpwgd  1/1     Running         2 (7m25s ago)   12d
productpage-v1-bb87ff47b-r75r5 0/2   Init:0/2        0               61s
ratings-v1-8589f64b4c-sntz4  0/2     Init:0/2        0               62s
reviews-v1-8cf7b9cc5-scgpl   0/2     Init:0/2        0               61s
reviews-v2-67d565655f-4c6kt  0/2     Init:0/2        0               61s
reviews-v3-d587fc9d7-w92jc   0/2     Init:0/2        0               61s
```

2. Validate that the app is running inside the cluster 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 | Select-String -Pattern "<title>.*</title>" | ForEach-Object { $_.Matches.Value }

```
PS C:\study\云计算导论\Labs\Lab7> kubectl exec (kubectl get pod -l app=rating
s -o jsonpath='{.items[0].metadata.name}') -c ratings -- curl -sS productpage
:9080/productpage | Select-String -Pattern "<title>.*</title>" | ForEach-Obje
ct { $_.Matches.Value }
<title>Simple Bookstore App</title>
```

Open the application to outside traffic

1. Create a Kubernetes Gateway for the Bookinfo application:

kubectl apply -f samples/bookinfo/gateway-api/bookinfo-gateway.yaml

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl apply -f samples/bookinfo/gateway-api/bookinfo-gateway.yaml
gateway.gateway.networking.k8s.io/bookinfo-gateway unchanged
httproute.gateway.networking.k8s.io/bookinfo configured
```

2. Change the service type to ClusterIP by annotating the gateway:

kubectl annotate gateway bookinfo-gateway

networking.istio.io/service-type=ClusterIP --namespace=default

```
PS C:\study\云计算导论\Labs\Lab7> kubectl annotate gateway bookinfo-gateway n
etworking.istio.io/service-type=ClusterIP --namespace=default
gateway.gateway.networking.k8s.io/bookinfo-gateway annotated
```

3. To check the status of the gateway, run:

kubectl get gateway

```
PS C:\study\云计算导论\Labs\Lab7> kubectl get gateway
NAME             CLASS    ADDRESS
 PROGRAMMED      AGE
bookinfo-gateway  istio    bookinfo-gateway-istio.default.svc.cluster.local
 True            19h
```

Access the application

1. You will connect to the Bookinfo productpage service through the gateway you just provisioned. To access the gateway, you need to use the kubectl port-forward command:

   kubectl port-forward svc/bookinfo-gateway-istio 8080:80

   ```
   PS C:\study\云计算导论\Labs\Lab7> kubectl port-forward svc/bookinfo-gateway-i
   stio 8080:80
   Forwarding from 127.0.0.1:8080 -> 80
   Forwarding from [::1]:8080 -> 80
   ```

2. Open your browser and navigate to http://localhost:8080/productpage to view the Bookinfo application:
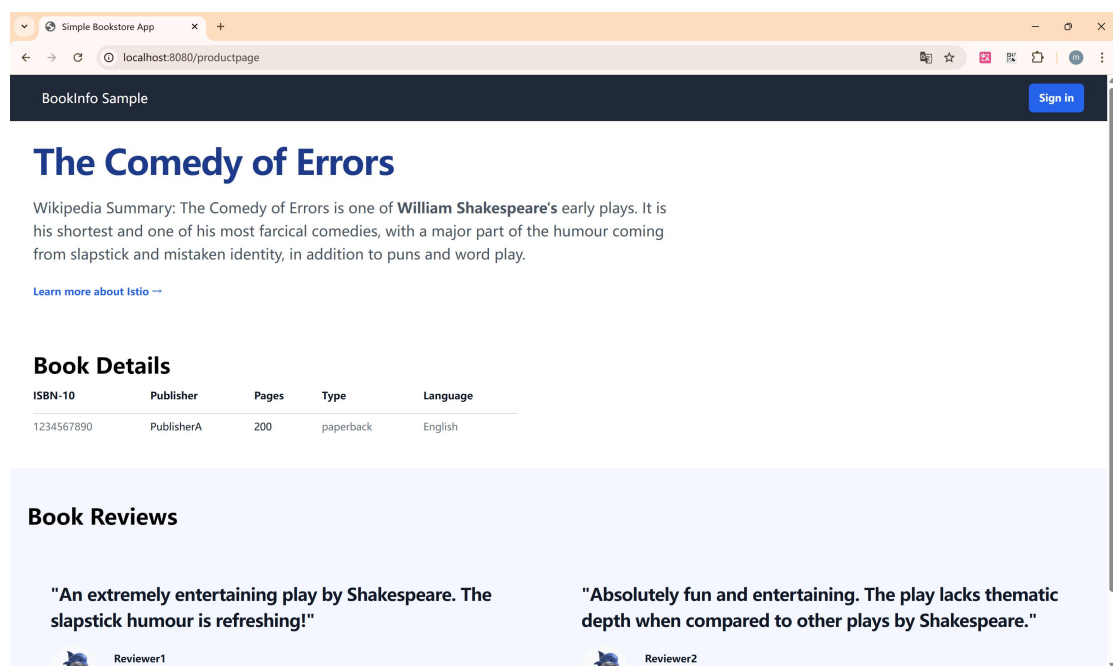
View the dashboard

1. Install Kiali and the other addons and wait for them to be deployed:

   kubectl apply -f samples/addons

   kubectl rollout status deployment/kiali -n istio-system

```
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl apply -f samples/addon
s
serviceaccount/grafana unchanged
configmap/grafana unchanged
service/grafana unchanged
deployment.apps/grafana configured
configmap/istio-grafana-dashboards configured
configmap/istio-services-grafana-dashboards configured
deployment.apps/jaeger unchanged
service/tracing unchanged
service/zipkin unchanged
service/jaeger-collector unchanged
serviceaccount/kiali unchanged
configmap/kiali unchanged
clusterrole.rbac.authorization.k8s.io/kiali unchanged
clusterrolebinding.rbac.authorization.k8s.io/kiali unchanged
service/kiali unchanged
deployment.apps/kiali unchanged
serviceaccount/loki unchanged
configmap/loki unchanged
configmap/loki-runtime unchanged
clusterrole.rbac.authorization.k8s.io/loki-clusterrole unchanged
clusterrolebinding.rbac.authorization.k8s.io/loki-clusterrolebinding unchange
d
service/loki-memberlist unchanged
service/loki-headless unchanged
service/loki unchanged
statefulset.apps/loki configured
serviceaccount/prometheus unchanged
configmap/prometheus unchanged
clusterrole.rbac.authorization.k8s.io/prometheus unchanged
clusterrolebinding.rbac.authorization.k8s.io/prometheus unchanged
service/prometheus unchanged
deployment.apps/prometheus configured
PS C:\study\云计算导论\Labs\Lab7\istio-1.27.1> kubectl rollout status deploym
ent/kiali -n istio-system
deployment "kiali" successfully rolled out
```

2. Access the Kiali dashboard:
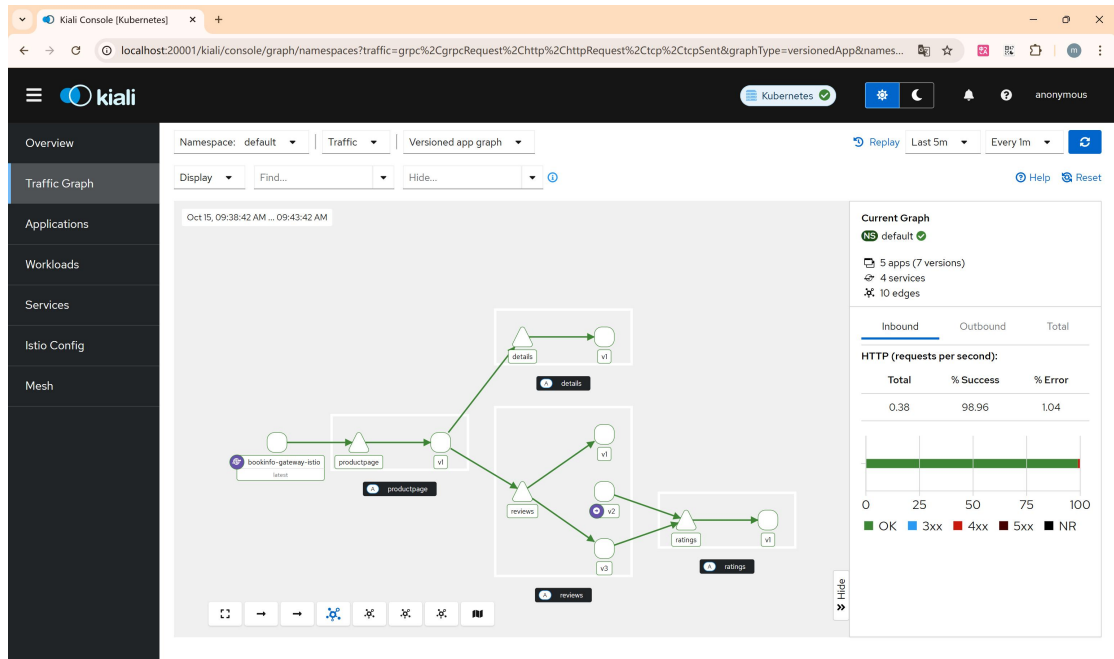
istioctl dashboard kiali
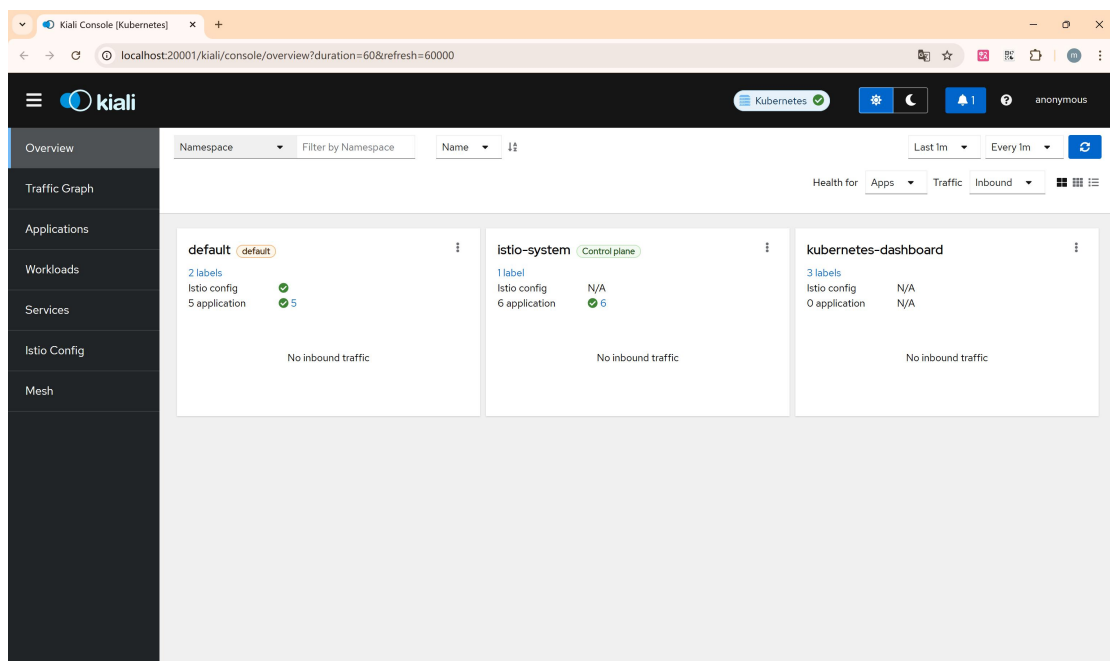
3. Send 100 requests to the productpage service:

```
for ($i = 1; $i -le 100; $i++) {
    Invoke-RestMethod -Uri "http://127.0.0.1:8080/productpage" -ErrorAction
SilentlyContinue | Out-Null
}
```

4. The Kiali dashboard shows an overview of your mesh with the relationships between the services in the Bookinfo sample application:

Cleanup

1. Uninstall Bookinfo sample:

samples/bookinfo/platform/kube/cleanup.sh