Department of Computing

Bachelor of Science (Hons) in Software Development

Cloud Data Centres - Y4

Lab 5

Student name: Sebastian Firsaev

Student Number: C00263348

Lecturer: Dr Lei Shi

Table of Contents

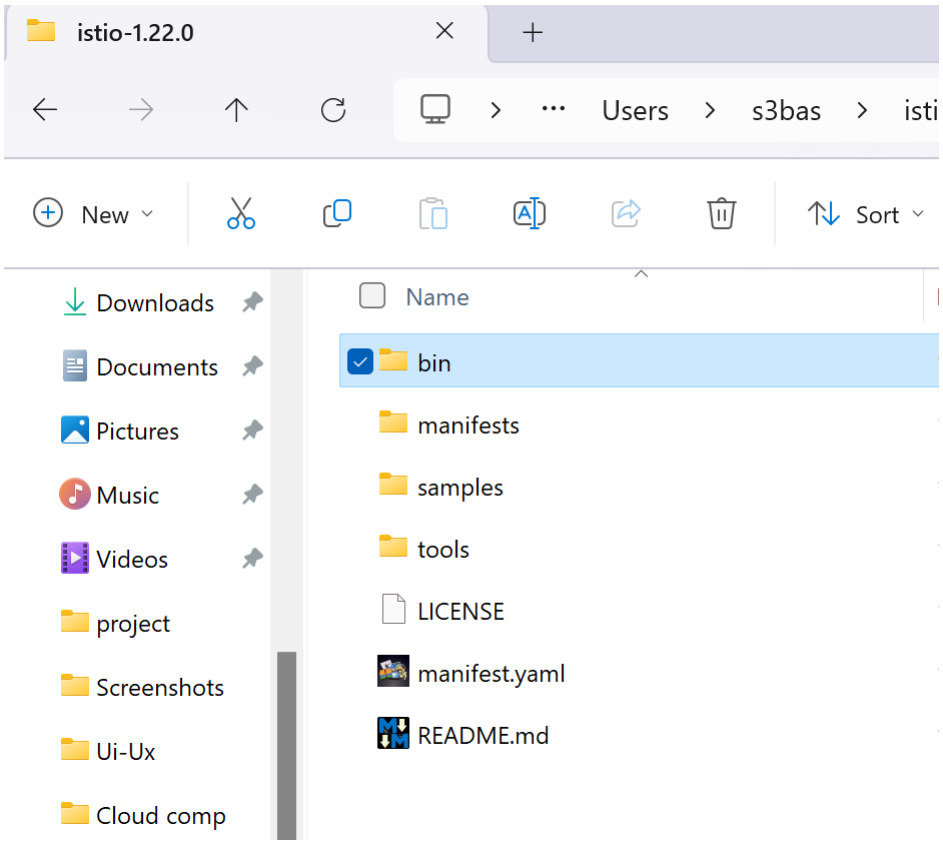
Introduction:

This lab report documents my experience with deploying and evaluating Istio 1.22, a service mesh solution tailored for microservices management in Kubernetes environments. Throughout this report, I will detail the steps involved in setting up Istio, deploying a sample application, and assessing its core functionalities, including traffic management, security enforcement, observability, and extensibility.

By following the provided instructions and conducting hands-on tasks, I aim to gain practical experience with Istio and understand its capabilities in real-world scenarios. This report serves as a record of my journey in exploring Istio's features and evaluating its potential benefits for enhancing the management and performance of microservices in Kubernetes environments.

Getting started with Istio

Downloading Istio



Setting up Istio path variable:

```
PS C:\Users\s3bas\istio-1.22.0> $env:PATH += ";C:\Users\s3bas\istio-1.22.0\bin"
PS C:\Users\s3bas\istio-1.22.0> 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
  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
  upgrade              Upgrade Istio control plane in-place
```

Installing Istio:

```
PS C:\Users\s3bas\istio-1.22.0> 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.
PS C:\Users\s3bas\istio-1.22.0>
```

Adding a namespace label:

```
PS C:\Users\s3bas\istio-1.22.0> kubectl label namespace default istio-injection=enabled
namespace/default labeled
```

Deploying Bookinfo sample application:

```
PS C:\Users\s3bas\istio-1.22.0> 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 services:

```
PS C:\Users\s3bas\istio-1.22.0> kubectl get services
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
details          ClusterIP   10.98.25.67      <none>        9080/TCP         2m58s
hello-minikube   NodePort    10.99.179.172    <none>        8080:31368/TCP   3h23m
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP          106d
productpage      ClusterIP   10.111.241.214   <none>        9080/TCP         2m57s
ratings          ClusterIP   10.101.232.85    <none>        9080/TCP         2m58s
reviews          ClusterIP   10.104.168.252   <none>        9080/TCP         2m57s
PS C:\Users\s3bas\istio-1.22.0>
```

Getting Pods, As each pod becomes ready, the Istio sidecar will be deployed along with it:

```
PS C:\Users\s3bas\istio-1.22.0> kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
details-v1-cf74bb974-ghj7p        2/2     Running   0          117s
productpage-v1-87d54dd59-6n2qg    2/2     Running   0          90s
ratings-v1-7c4bbf97db-p5ztr       2/2     Running   0          78s
reviews-v1-5fd6d4f8f8-b5jgr       2/2     Running   0          64s
reviews-v2-6f9b55c5db-k4vbw       2/2     Running   0          42s
reviews-v3-7d99fd7978-fxlvq       2/2     Running   0          29s
PS C:\Users\s3bas\istio-1.22.0>
```

Verify everything is working correctly up to this point:

```
PS C:\Users\s3bas\istio-1.22.0>  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>"

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

Opening the application to outside traffic

Associating this application with the Istio gateway:

```
PS C:\Users\s3bas\istio-1.22.0> kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml
gateway.networking.istio.io/bookinfo-gateway created
virtualservice.networking.istio.io/bookinfo created
PS C:\Users\s3bas\istio-1.22.0>
```

Ensuring that there are no issues with the configuration:

```
PS C:\Users\s3bas\istio-1.22.0> istioctl analyze
Info [IST0118] (Service default/hello-minikube) Port name  (port: 8080, targetPort: 8080) doesn't follow the naming conv
ention of Istio port.
PS C:\Users\s3bas\istio-1.22.0>
```

Cleaning old service:

```
PS C:\Users\s3bas\istio-1.22.0> kubectl delete service hello-minikube
service "hello-minikube" deleted
```

No issues:

```
PS C:\Users\s3bas\istio-1.22.0> istioctl analyze

✔No validation issues found when analyzing namespace: default.
PS C:\Users\s3bas\istio-1.22.0>
```

Determining the ingress IP and ports

starting a Minikube tunnel that sends traffic to Istio Ingress Gateway:

```
PS C:\Users\s3bas\istio-1.22.0> minikube tunnel
W0517 12:30:42.472374   19716 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\s3bas\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\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 accessible ...
```

Setting the ingress host and ports:

Ingress Host:

```
PS C:\Users\s3bas\istio-1.22.0> $INGRESS_HOST = kubectl -n istio-system get service istio-ingressgateway -o jsonpath="{.
status.loadBalancer.ingress[0].ip}"
PS C:\Users\s3bas\istio-1.22.0> $env:INGRESS_hOST = $INGRESS_HOST
```

Ensuring an IP address and ports were successfully assigned to each environment variable:

```
PS C:\Users\s3bas\istio-1.22.0> echo "$INGRESS_HOST"
127.0.0.1
```

```
PS C:\Users\s3bas\istio-1.22.0> echo "$INGRESS_PORT"
80
```

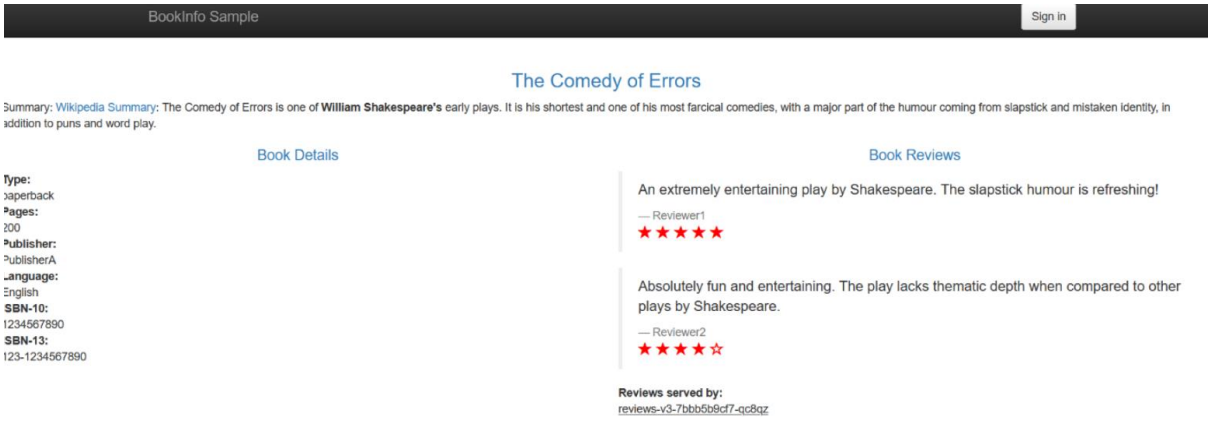
```
PS C:\Users\s3bas\istio-1.22.0> echo "$SECURE_INGRESS_PORT"
443
```

Setting the GATEWAY_URL:

```
PS C:\Users\s3bas\istio-1.22.0> $GATEWAY_URL = "$env:INGRESS_HOST:$env:INGRESS_PORT"
PS C:\Users\s3bas\istio-1.22.0> $env:GATEWAY_URL = $GATEWAY_URL
```

```
PS C:\Users\s3bas\istio-1.22.0> echo "$GATEWAY_URL"
127.0.0.1:80
```

Verifying external access



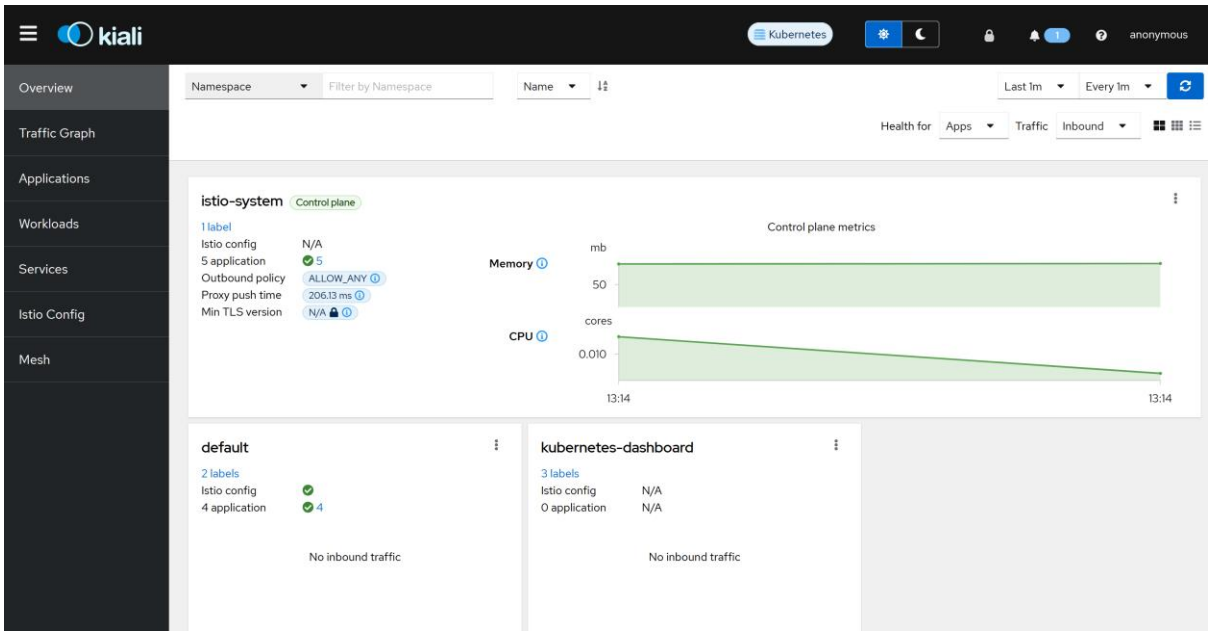
Viewing the dashboard:

Installing Kiali and the other addons

```
PS C:\Users\s3bas\istio-1.22.0> 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
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
```

```
PS C:\Users\s3bas\istio-1.22.0> kubectl rollout status deployment/kiali -n istio-system
deployment "kiali" successfully rolled out
```

Viewing the Kiali dashboard:



Sending a 100 requests to the productpage service and viewing the graph:

```
PS C:\Users\s3bas> foreach ($i in 1..100){Invoke-WebRequest -Uri "http://127.0.0.1/productpage" -UseBasicParsing | Out-N
ull}
PS C:\Users\s3bas>
```