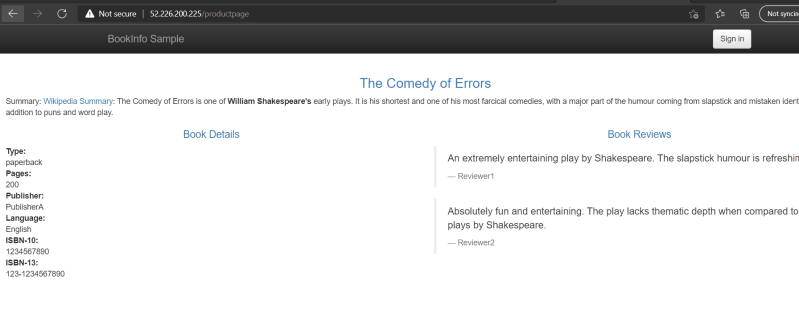
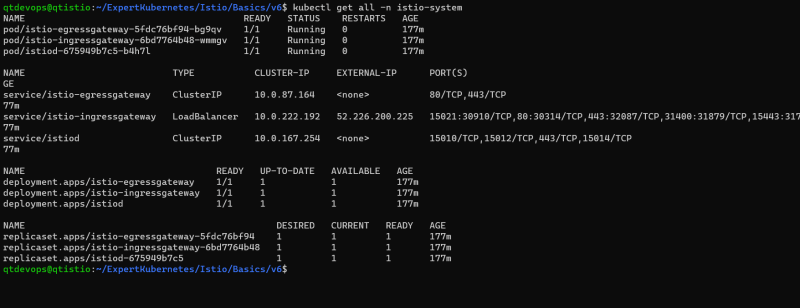
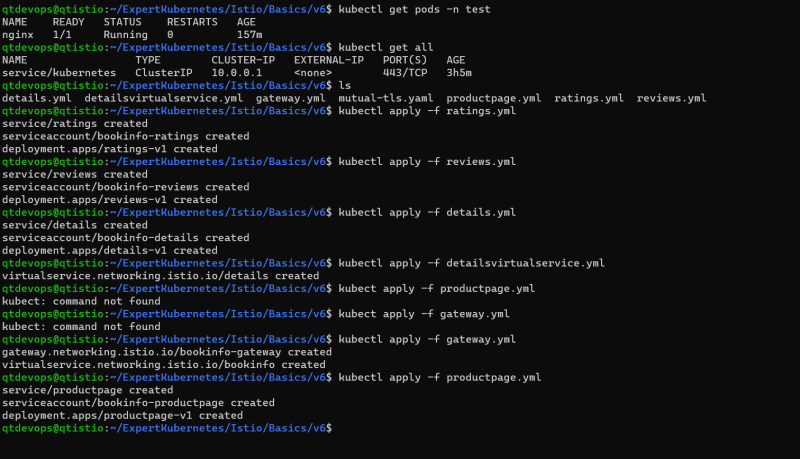
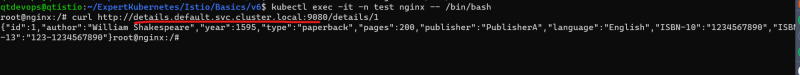
**Mutual TLS**

* Create a nginx pod in test namespace
* Created a book info application in default namespace 
* Now lets try to login into nginx pod in the test namespace and send the curl request to details page 
* Now lets enable mutual tls across all the resources in the default namespace

---

apiVersion: security.istio.io/v1beta1

kind: PeerAuthentication

metadata:

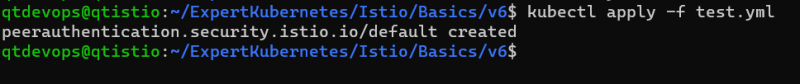
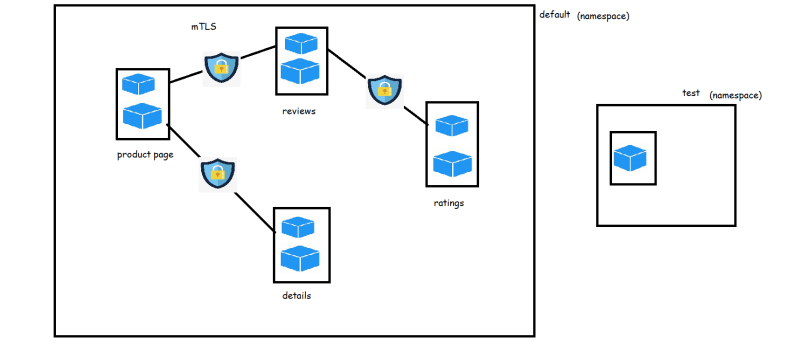
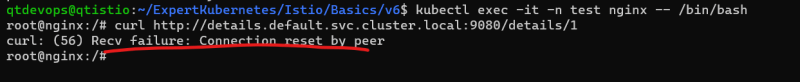
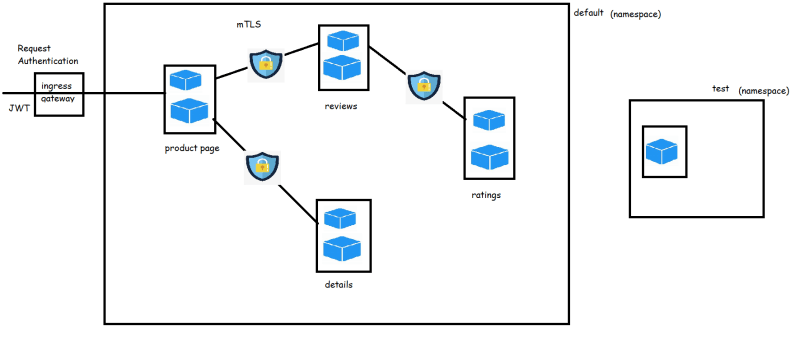
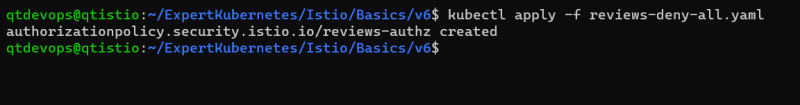
name: default

namespace: default

spec:

mtls:

mode: STRICT

* Apply the above yaml 
* Now try to login into nginx and send the curl request 
* For securing external request use the combination of RequestAuthentication & JWT (Json Web Tokens) 
* If we need to setup only selected services to access our service we can setup Authorization policy by making use of service accounts
* We have create a review authorization policy to deny all the requests [refer](https://github.com/asquarezone/ExpertKubernetes/commit/c41588b69c35f280662e0213070ba54730145272) 
* The yaml so far is

---

apiVersion: security.istio.io/v1beta1

kind: AuthorizationPolicy

metadata:

name: reviews-authz

namespace: default

spec:

selector:

matchLabels:

app: reviews

* Now if you access product page, it will not be able to show reviews
* Now lets change yaml to allow access from product page. <https://github.com/asquarezone/ExpertKubernetes/commit/885c18a1eefc5abbbad527583f3b9a00a8bef9ba> for the changes

---

apiVersion: security.istio.io/v1beta1

kind: AuthorizationPolicy

metadata:

name: reviews-authz

namespace: default

spec:

selector:

matchLabels:

app: reviews

rules:

- from:

- source:

principals: ["cluster.local/ns/default/sa/bookinfo-productpage"]

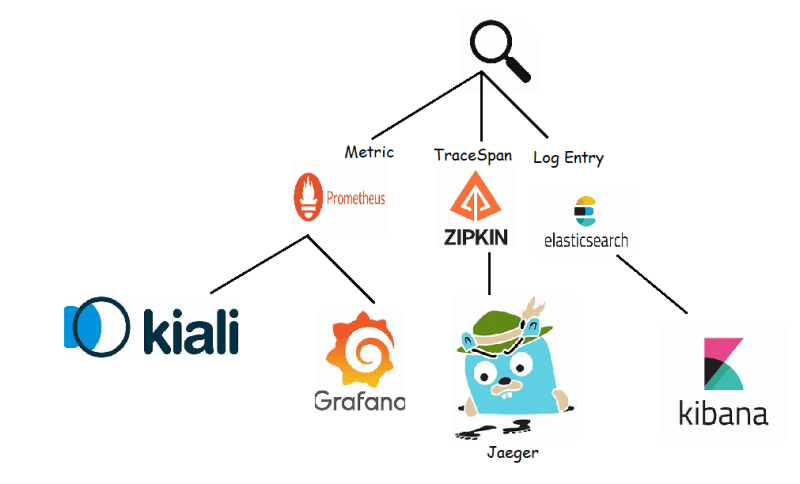
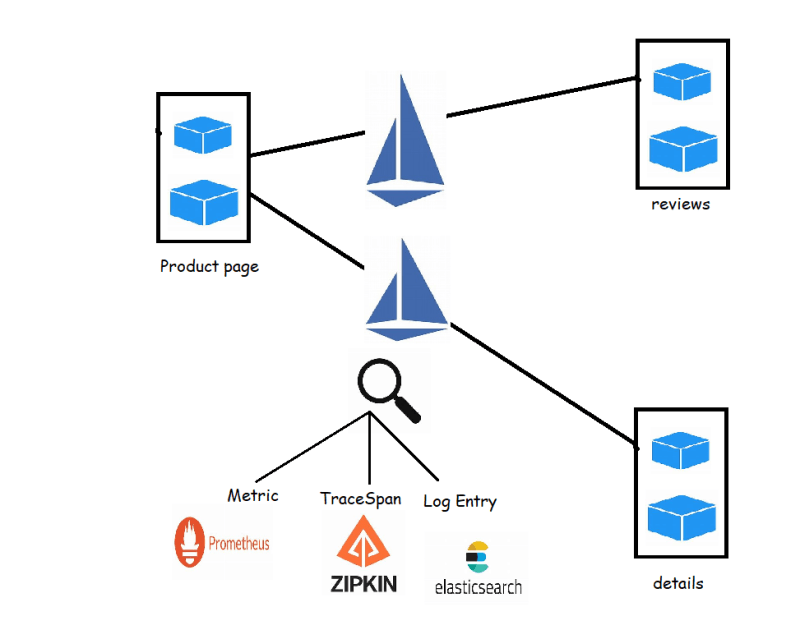
to:

- operation:

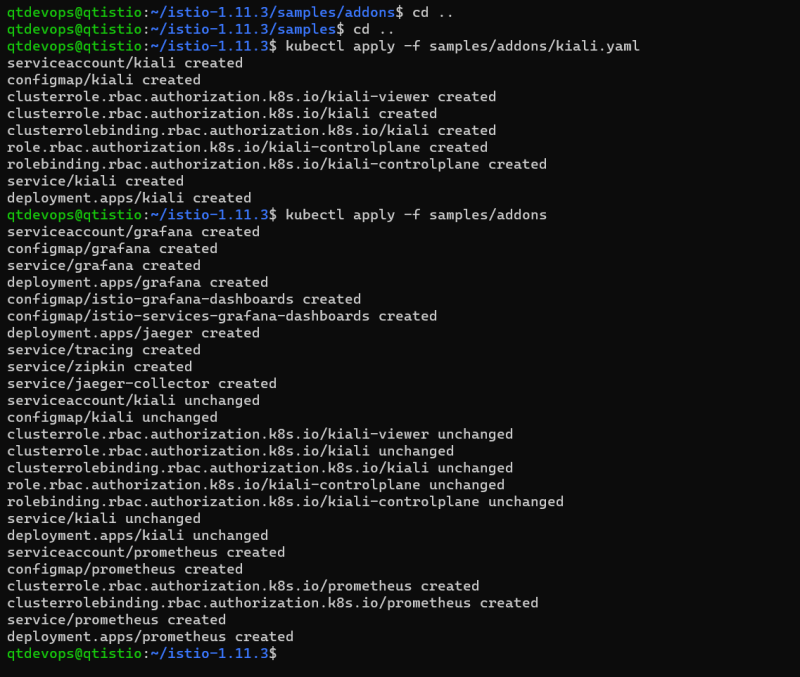
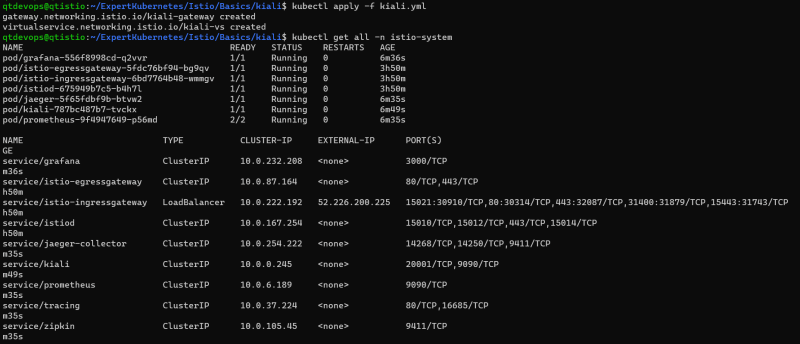
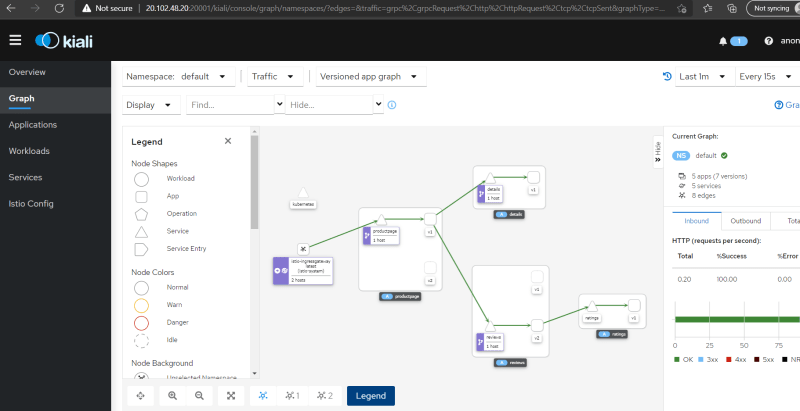
methods: ["GET"]

* Now if you access product page, it will be able to show reviews

**Observability Using Service Mesh**

* From Istio Service mesh we can view logs, metrics and visualizations as istio supports metric adapters which can be connected by multiple popular opensource tools like Prometheus, Kiali, Jaeger 

**Visualize the Service Mesh using Kiali**

* Lets try to deploy the Blue Green Deployment <https://github.com/asquarezone/ExpertKubernetes/tree/master/Istio/Basics/v4>
* Now Navigate to istio directory and execute the command <https://istio.io/latest/docs/setup/getting-started/#dashboard> 
* <https://github.com/asquarezone/ExpertKubernetes/commit/a7e657da8aa0bd6cd5f862207cadacd7205e61ff> for the kiali gateway 
* From your node where you are running istio try to use istioctl dashboard kiali --address 0.0.0.0
* Now try to use your application and look at graph of kiali 
* Lets use fortio docker container to add the artificial load to our application
* Now try to create a docker container anywhere with the following command to generate artificial load

docker container run `

--add-host "bookinfo.com:52.226.200.225" `

fortio/fortio load -c 32 -qps 25 -t 5m http://bookinfo.com/productpage

docker container run `

--add-host "test.bookinfo.com:52.226.200.225" `

fortio/fortio load -c 32 -qps 25 -t 5m http://test.bookinfo.com/productpage