Carnegie Mellon University Africa

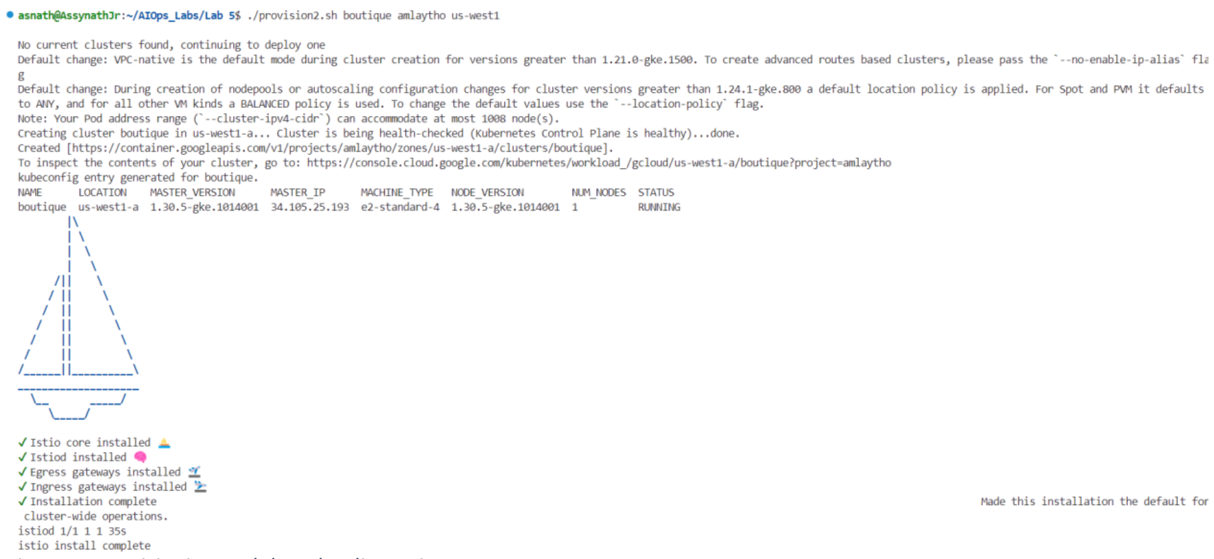
04-800K: AIOps: Continuous and Automated IT and AI Monitoring

Assynath Thompson Mlay, amlaytho

MSIT 2025

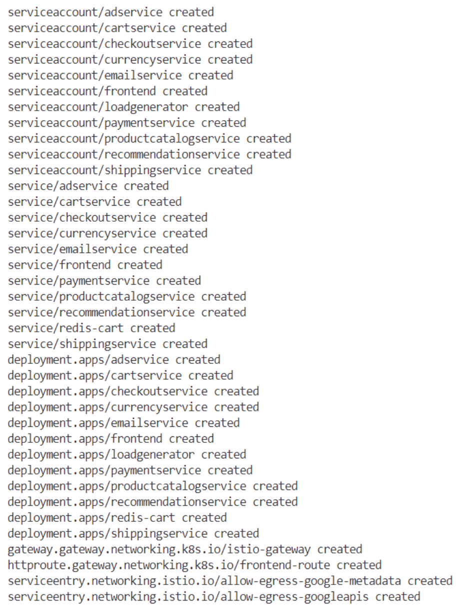
Lab 5: Capturing application metrics with Istio service mesh.

Lab Preparations:



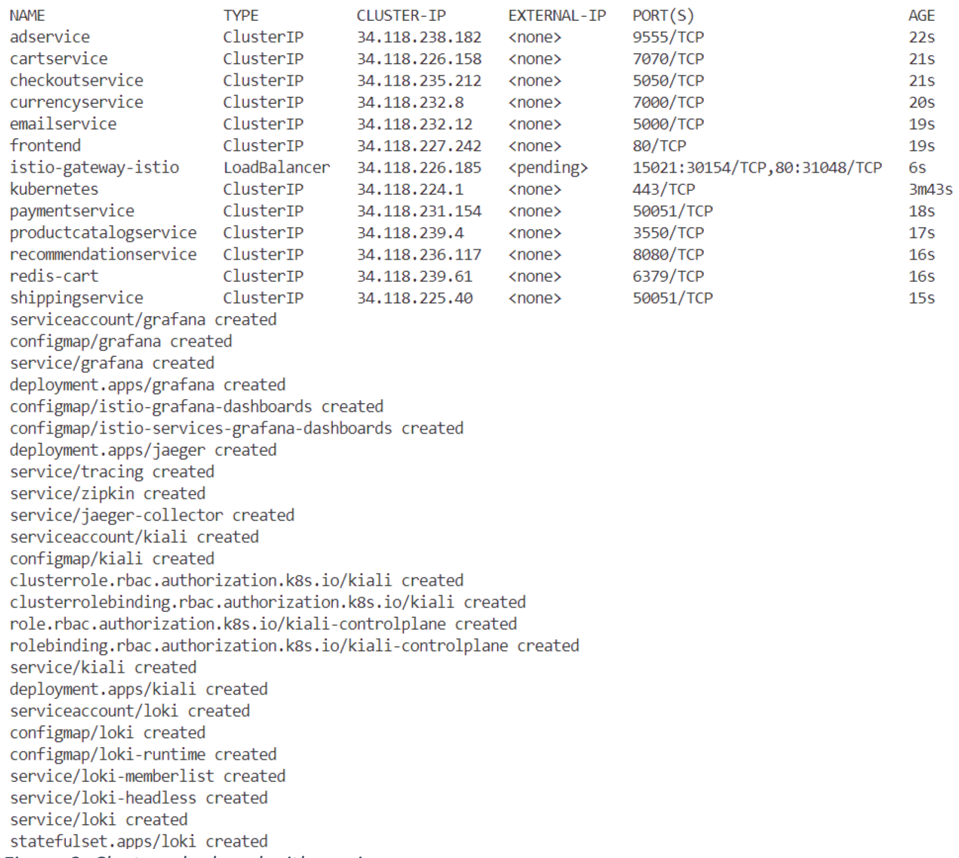
```
● asnath@AssynathJr:~/AIOps_Labs/Lab 5$ ./provision2.sh boutique amlaytho us-west1

  No current clusters found, continuing to deploy one
  Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based clusters, please pass the `--no-enable-ip-alias` fla
  g
  Default change: During creation of nodepools or autoscaling configuration changes for cluster versions greater than 1.24.1-gke.800 a default location policy is applied. For Spot and PVM it defaults
  to ANY, and for all other VM kinds a BALANCED policy is used. To change the default values use the `--location-policy` flag.
  Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
  Creating cluster boutique in us-west1-a... Cluster is being health-checked (Kubernetes Control Plane is healthy)...done.
  Created [https://container.googleapis.com/v1/projects/amlaytho/zones/us-west1-a/clusters/boutique].
  To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-west1-a/boutique?project=amlaytho
  kubeconfig entry generated for boutique.
  NAME      LOCATION    MASTER_VERSION      MASTER_IP      MACHINE_TYPE   NODE_VERSION        NUM_NODES  STATUS
  boutique  us-west1-a  1.30.5-gke.1014001  34.105.25.193  e2-standard-4  1.30.5-gke.1014001  1          RUNNING
          |\
          | \
          |  \
          |   \
        /||    \
       / ||     \
      /  ||      \
     /   ||       \
    /    ||        \
   /     ||         \
  /______||__________\
  ____________________
    \__       _____/
       \_____/

  ✓ Istio core installed ⛵
  ✓ Istiod installed 🧠
  ✓ Egress gateways installed 🛫
  ✓ Ingress gateways installed 🛬
  ✓ Installation complete                                                                                                             Made this installation the default for
   cluster-wide operations.
  istiod 1/1 1 1 35s
  istio install complete
```

Figure 1: Provisioning and downloading Istio



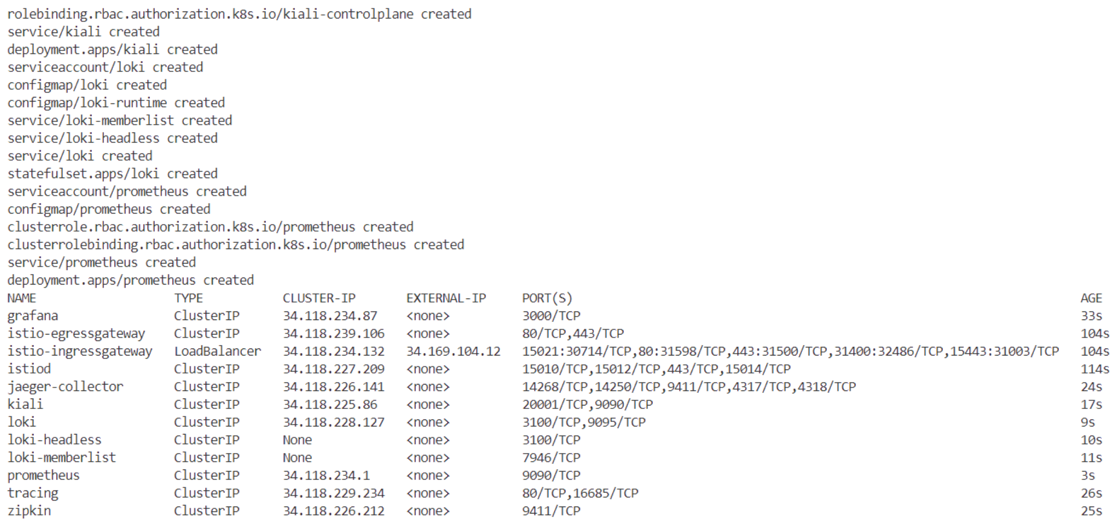
```
serviceaccount/adservice created
serviceaccount/cartservice created
serviceaccount/checkoutservice created
serviceaccount/currencyservice created
serviceaccount/emailservice created
serviceaccount/frontend created
serviceaccount/loadgenerator created
serviceaccount/paymentservice created
serviceaccount/productcatalogservice created
serviceaccount/recommendationservice created
serviceaccount/shippingservice created
service/adservice created
service/cartservice created
service/checkoutservice created
service/currencyservice created
service/emailservice created
service/frontend created
service/paymentservice created
service/productcatalogservice created
service/recommendationservice created
service/redis-cart created
service/shippingservice created
deployment.apps/adservice created
deployment.apps/cartservice created
deployment.apps/checkoutservice created
deployment.apps/currencyservice created
deployment.apps/emailservice created
deployment.apps/frontend created
deployment.apps/loadgenerator created
deployment.apps/paymentservice created
deployment.apps/productcatalogservice created
deployment.apps/recommendationservice created
deployment.apps/redis-cart created
deployment.apps/shippingservice created
gateway.gateway.networking.k8s.io/istio-gateway created
httproute.gateway.networking.k8s.io/frontend-route created
serviceentry.networking.istio.io/allow-egress-google-metadata created
serviceentry.networking.istio.io/allow-egress-googleapis created
```

Figure 2: Microservices deployed



```
NAME                   TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                         AGE
adservice              ClusterIP      34.118.238.182   <none>        9555/TCP                        22s
cartservice            ClusterIP      34.118.226.158   <none>        7070/TCP                        21s
checkoutservice        ClusterIP      34.118.235.212   <none>        5050/TCP                        21s
currencyservice        ClusterIP      34.118.232.8     <none>        7000/TCP                        20s
emailservice           ClusterIP      34.118.232.12    <none>        5000/TCP                        19s
frontend               ClusterIP      34.118.227.242   <none>        80/TCP                          19s
istio-gateway-istio    LoadBalancer   34.118.226.185   <pending>     15021:30154/TCP,80:31048/TCP    6s
kubernetes             ClusterIP      34.118.224.1     <none>        443/TCP                         3m43s
paymentservice         ClusterIP      34.118.231.154   <none>        50051/TCP                       18s
productcatalogservice  ClusterIP      34.118.239.4     <none>        3550/TCP                        17s
recommendationservice  ClusterIP      34.118.236.117   <none>        8080/TCP                        16s
redis-cart             ClusterIP      34.118.239.61    <none>        6379/TCP                        16s
shippingservice        ClusterIP      34.118.225.40    <none>        50051/TCP                       15s
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 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
```

Figure 3: Clusters deployed with services



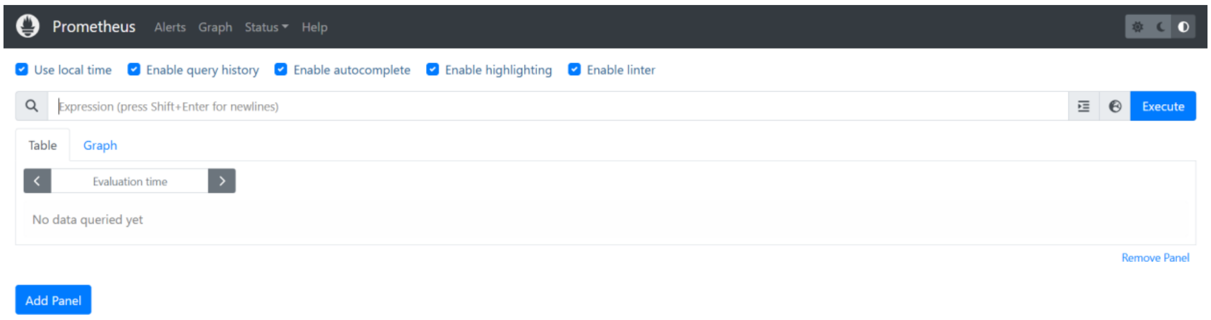
```
rolebinding.rbac.authorization.k8s.io/kiali-controlplane created
service/kiali created
deployment.apps/kiali created
serviceaccount/loki created
configmap/loki created
configmap/loki-runtime created
service/loki-memberlist created
service/loki-headless created
service/loki created
statefulset.apps/loki 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
NAME                   TYPE           CLUSTER-IP       EXTERNAL-IP     PORT(S)                                                                     AGE
grafana                ClusterIP      34.118.234.87    <none>          3000/TCP                                                                    33s
istio-egressgateway    ClusterIP      34.118.239.106   <none>          80/TCP,443/TCP                                                              104s
istio-ingressgateway   LoadBalancer   34.118.234.132   34.169.104.12   15021:30714/TCP,80:31598/TCP,443:31500/TCP,31400:32486/TCP,15443:31003/TCP  104s
istiod                 ClusterIP      34.118.227.209   <none>          15010/TCP,15012/TCP,443/TCP,15014/TCP                                       114s
jaeger-collector       ClusterIP      34.118.226.141   <none>          14268/TCP,14250/TCP,9411/TCP,4317/TCP,4318/TCP                              24s
kiali                  ClusterIP      34.118.225.86    <none>          20001/TCP,9090/TCP                                                          17s
loki                   ClusterIP      34.118.228.127   <none>          3100/TCP,9095/TCP                                                           9s
loki-headless          ClusterIP      None             <none>          3100/TCP                                                                    10s
loki-memberlist        ClusterIP      None             <none>          7946/TCP                                                                    11s
prometheus             ClusterIP      34.118.234.1     <none>          9090/TCP                                                                    3s
tracing                ClusterIP      34.118.229.234   <none>          80/TCP,16685/TCP                                                            26s
zipkin                 ClusterIP      34.118.226.212   <none>          9411/TCP                                                                    25s
```

Figure 4: Other clusters

Lab Task1: Install Prometheus and Grafana for the Boutique Application

```
asnath@AssynathJr:~/AIOps_Labs/Lab 5$ istioctl dashboard prometheus
http://localhost:9090
```

Figure 5: Opening Prometheus Dashboard



Prometheus Alerts Graph Status Help

☑ Use local time ☑ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

Expression (press Shift+Enter for newlines) Execute

Table Graph

Evaluation time

No data queried yet

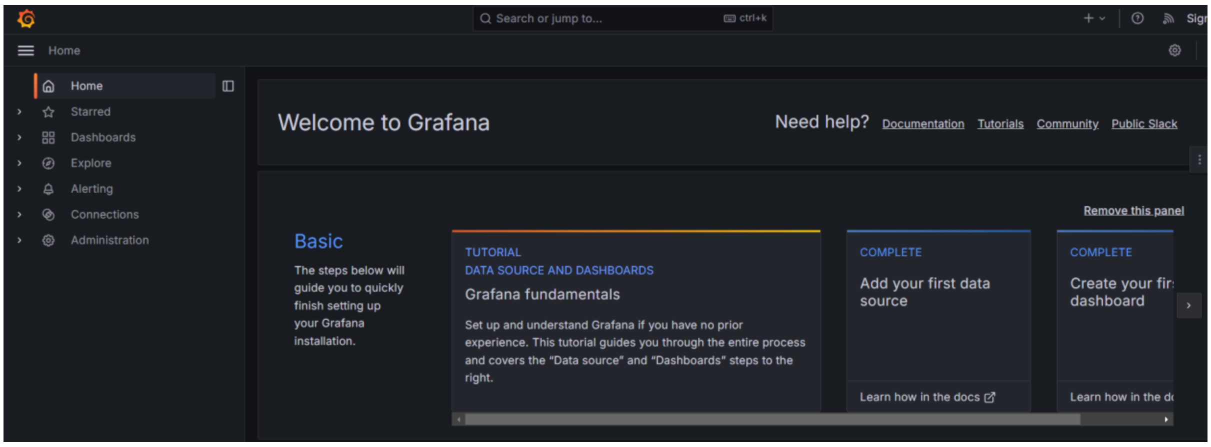
Remove Panel

Add Panel

Figure 6: Opened Prometheus

```
asnath@AssynathJr:~/AIOps_Labs/Lab 5$ istioctl dashboard grafana
http://localhost:3000
```

Figure 7: Opening Grafana Dashboard



Search or jump to... ctrl+k

Home

Home
Starred
Dashboards
Explore
Alerting
Connections
Administration

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

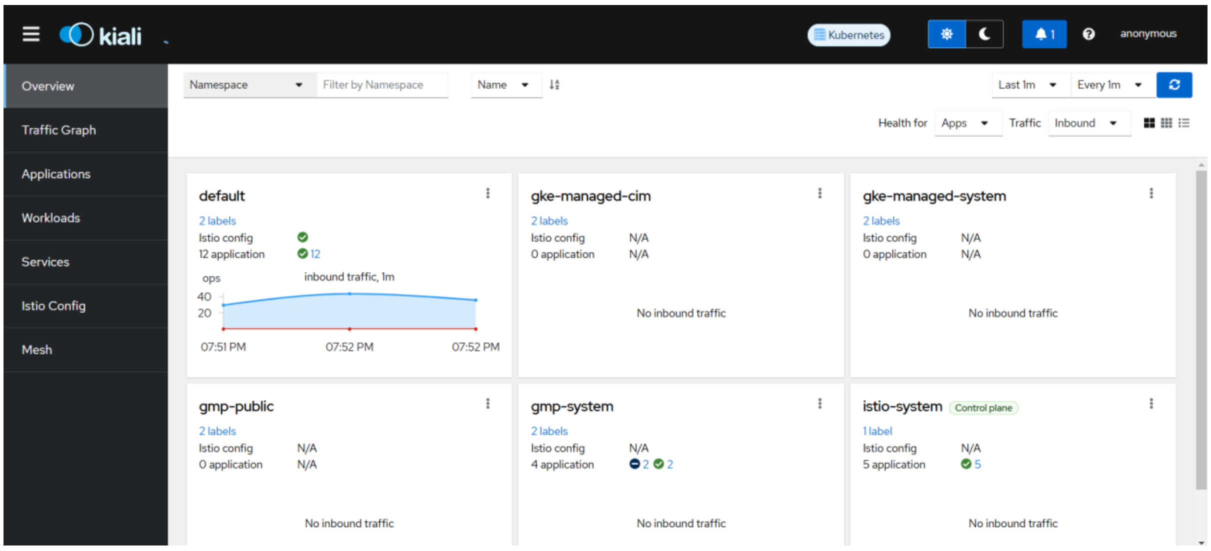
COMPLETE
Add your first data source
Learn how in the docs

COMPLETE
Create your fir dashboard
Learn how in the d

Figure 8: Opening Grafana

```
asnath@AssynathJr:~/AIOps_Labs/Lab 5$ istioctl dashboard kiali
http://localhost:20001/kiali
```

Figure 9: Opening Kiali Dashboard



default
2 labels
Istio config ✓
12 application ✓ 12
inbound traffic, 1m
ops
40
20
07:51 PM 07:52 PM 07:52 PM

gke-managed-cim
2 labels
Istio config N/A
0 application N/A
No inbound traffic

gke-managed-system
2 labels
Istio config N/A
0 application N/A
No inbound traffic

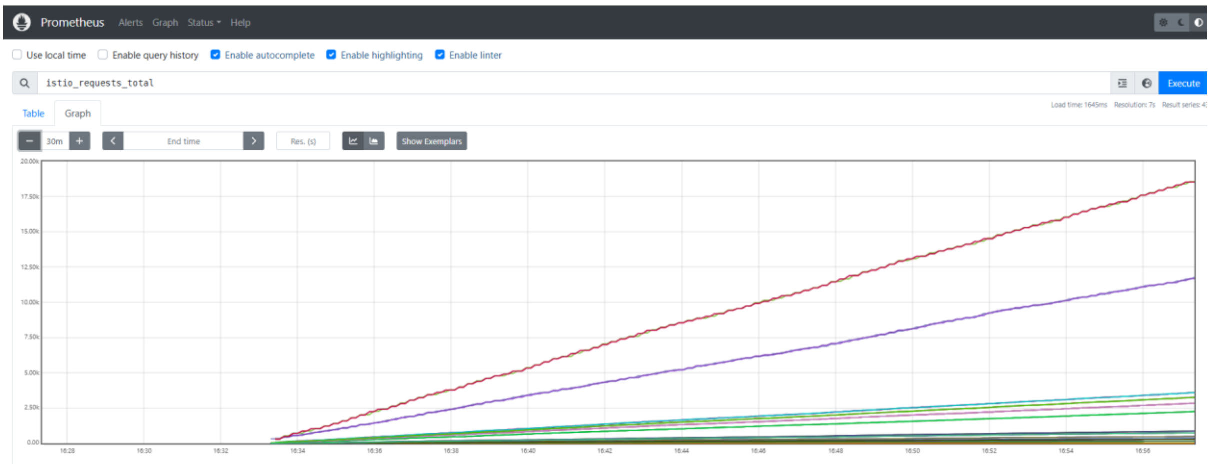
gmp-public
2 labels
Istio config N/A
0 application N/A
No inbound traffic

gmp-system
2 labels
Istio config N/A
4 application ⊖ 2 ✓ 2
No inbound traffic

istio-system Control plane
1 label
Istio config N/A
5 application ✓ 5
No inbound traffic

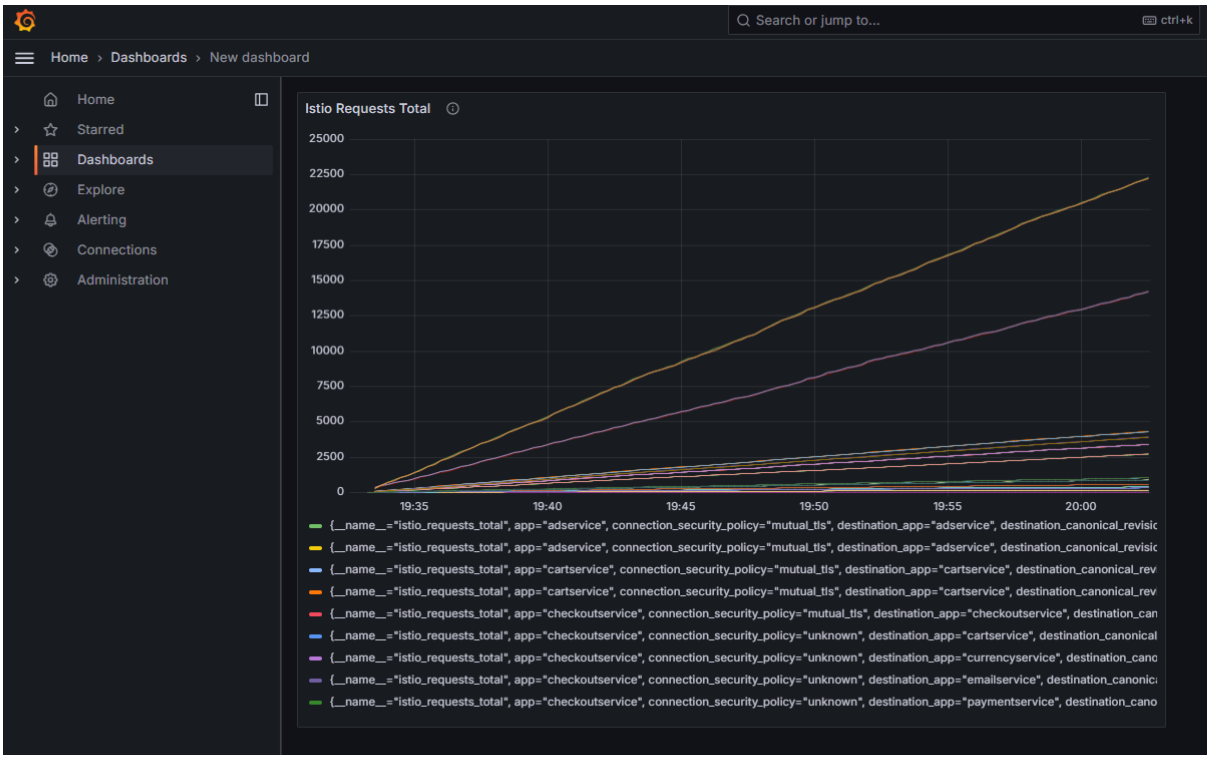
Figure 10: Kiali Dashboard

Runing the “istio_requests_total”:



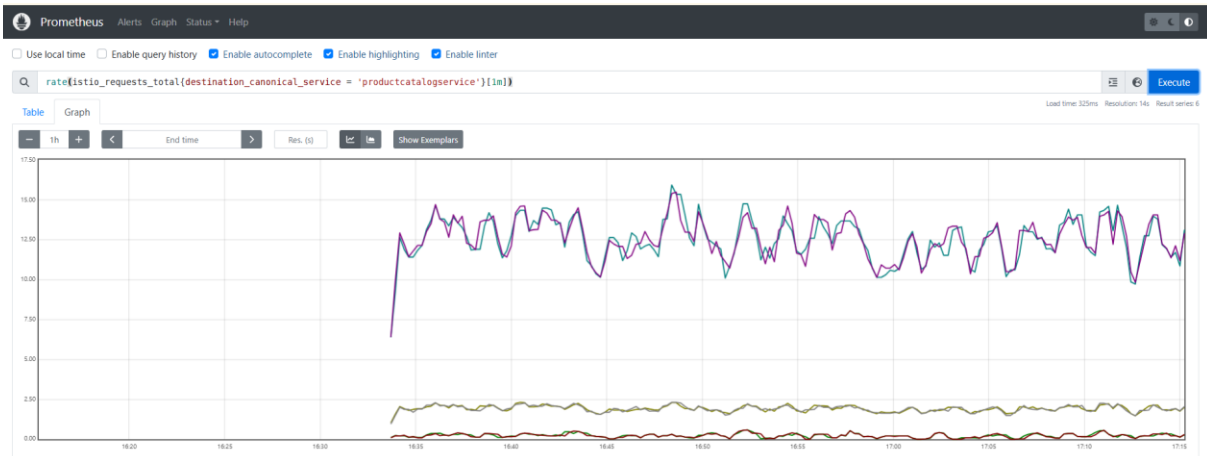
```
istio_requests_total
```

Figure 11: istio_requests_total on Prometheus



```
{__name__="istio_requests_total", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revisic
{__name__="istio_requests_total", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revisic
{__name__="istio_requests_total", app="cartservice", connection_security_policy="mutual_tls", destination_app="cartservice", destination_canonical_rev
{__name__="istio_requests_total", app="cartservice", connection_security_policy="mutual_tls", destination_app="cartservice", destination_canonical_rev
{__name__="istio_requests_total", app="checkoutservice", connection_security_policy="mutual_tls", destination_app="checkoutservice", destination_can
{__name__="istio_requests_total", app="checkoutservice", connection_security_policy="unknown", destination_app="cartservice", destination_canonical
{__name__="istio_requests_total", app="checkoutservice", connection_security_policy="unknown", destination_app="currencyservice", destination_cano
{__name__="istio_requests_total", app="checkoutservice", connection_security_policy="unknown", destination_app="emailservice", destination_canonic
{__name__="istio_requests_total", app="checkoutservice", connection_security_policy="unknown", destination_app="paymentservice", destination_cano
```

Figure 12: Istio Requests Total on Grafana

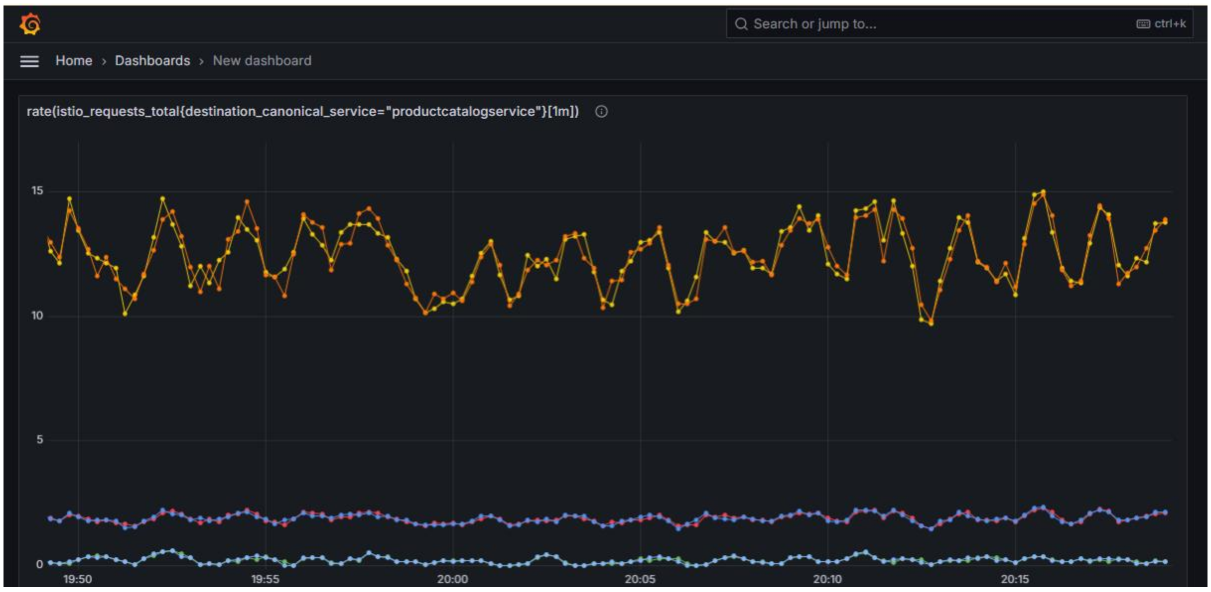


Use local time ☐ Enable query history ☐ Enable autocomplete ☑ Enable highlighting ☑ Enable linter ☑

```
rate(istio_requests_total{destination_canonical_service = 'productcatalogservice'}[1m])
```

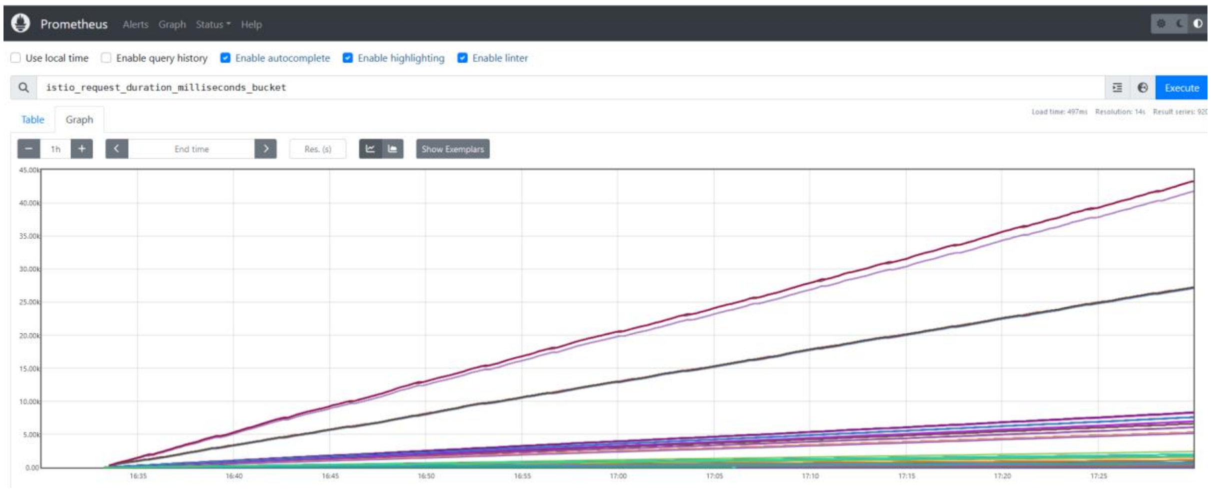
Load time: 325ms Resolution: 14s Result series: 6

Figure 13: identifying source/destination services



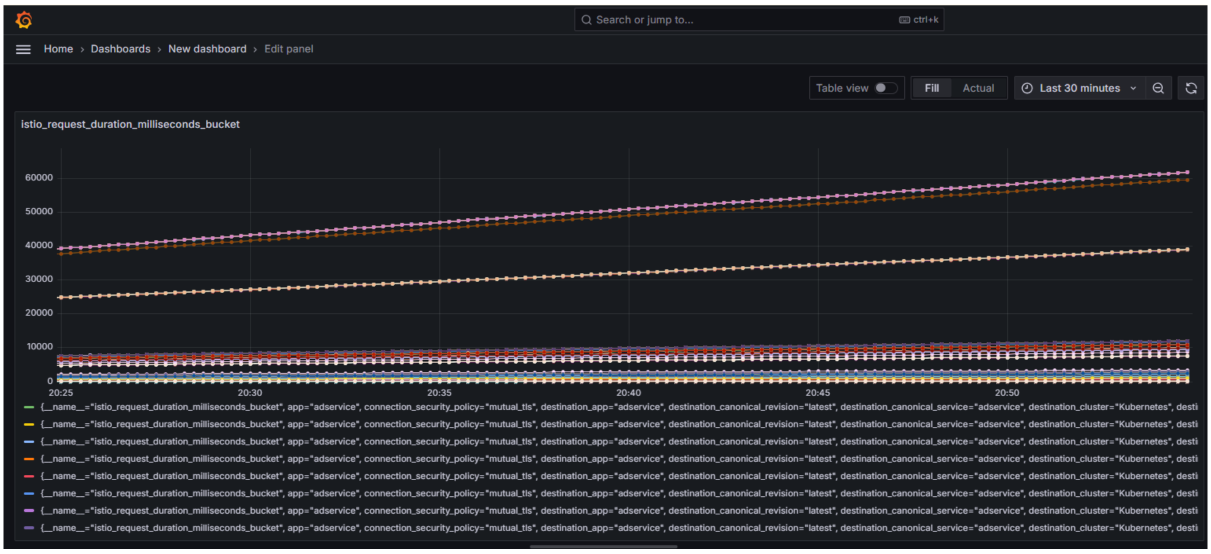
```
rate(istio_requests_total{destination_canonical_service="productcatalogservice"}[1m])
```

Figure 14: finding the destination services on grafana



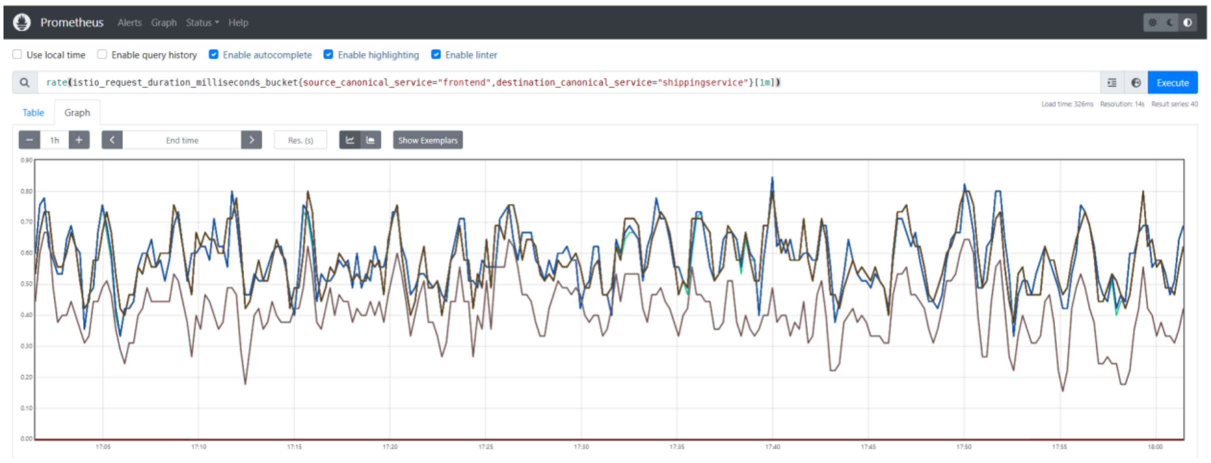
```
istio_request_duration_milliseconds_bucket
```

Figure 15: istio request duration milliseconds bucket


Search or jump to...
ctrl+k
Home › Dashboards › New dashboard › Edit panel
Table view
Fill
Actual
Last 30 minutes
istio_request_duration_milliseconds_bucket

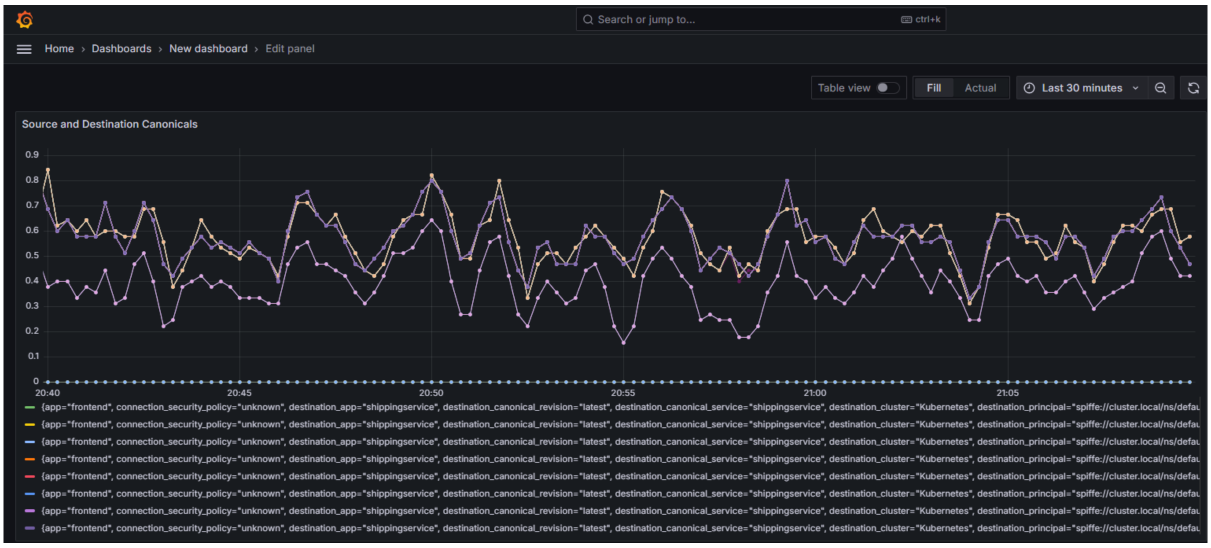
```
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
{__name__="istio_request_duration_milliseconds_bucket", app="adservice", connection_security_policy="mutual_tls", destination_app="adservice", destination_canonical_revision="latest", destination_canonical_service="adservice", destination_cluster="Kubernetes", desti
```

Figure 16: istio request duration milliseconds bucket



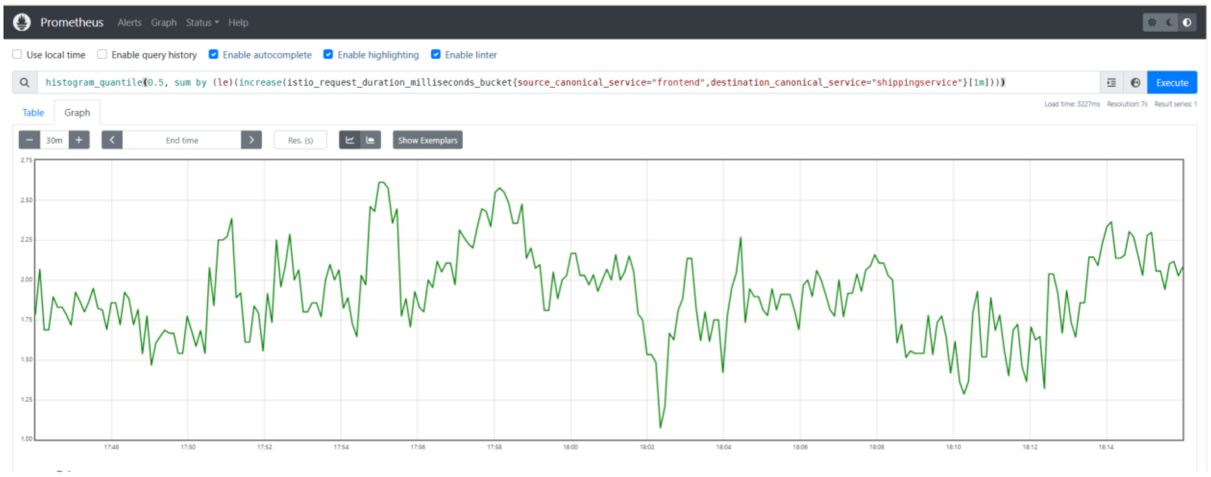
```
rate(istio_request_duration_milliseconds_bucket{source_canonical_service="frontend",destination_canonical_service="shippingservice"}[1m])
```

Figure 17: both source and destination canonical



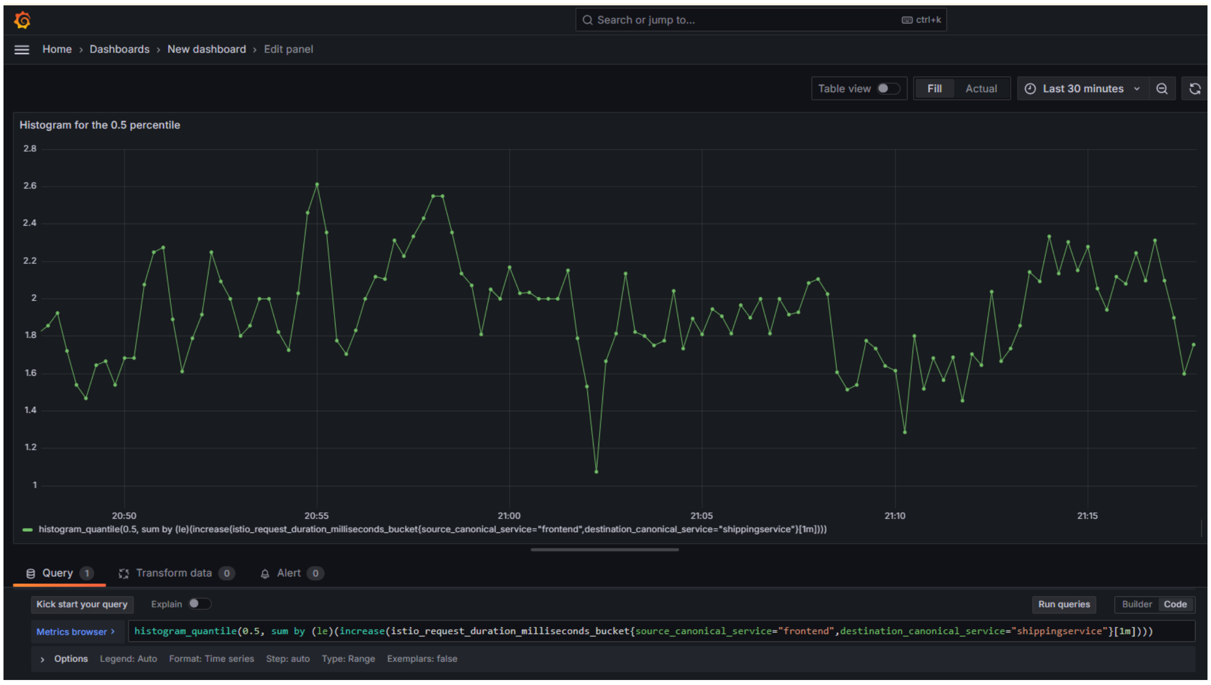
```
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
{app="frontend", connection_security_policy="unknown", destination_app="shippingservice", destination_canonical_revision="latest", destination_canonical_service="shippingservice", destination_cluster="Kubernetes", destination_principal="spiffe://cluster.local/ns/defau
```

Figure 18: both source and destination canonicals on Grafana



```
histogram_quantile(0.5, sum by (le)(increase(istio_request_duration_milliseconds_bucket{source_canonical_service="frontend",destination_canonical_service="shippingservice"}[1m])))
```

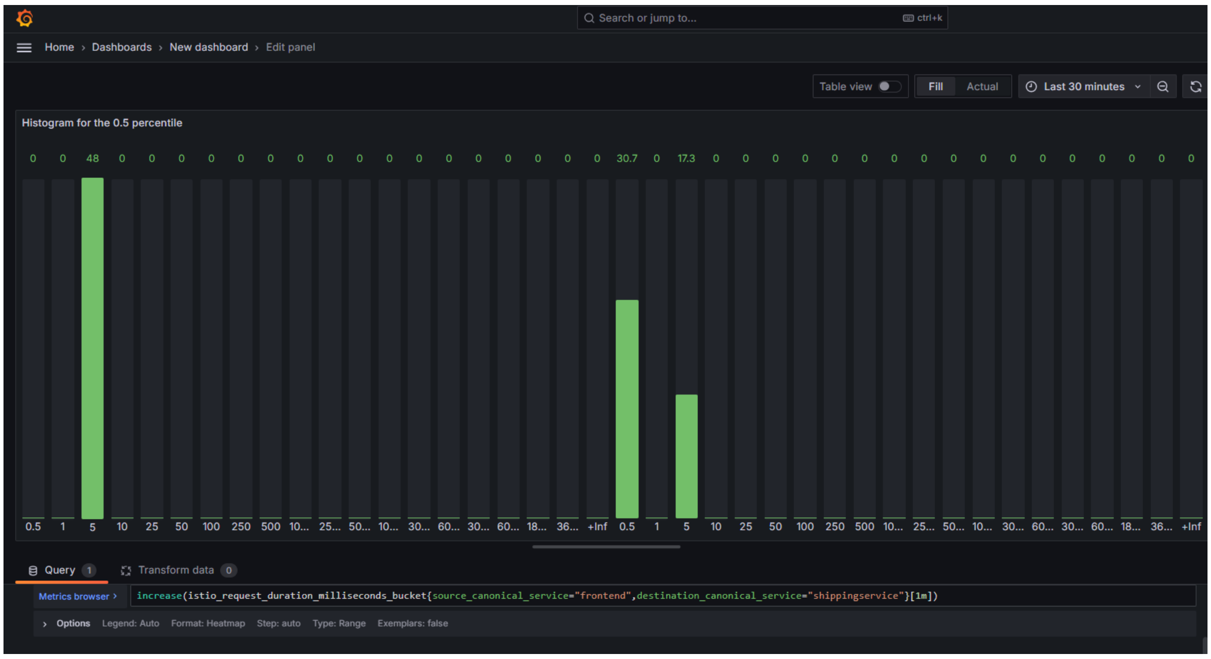
Figure 19: Histogram for the 0.5 percentile on Prometheus



```
histogram_quantile(0.5, sum by (le)(increase(istio_request_duration_milliseconds_bucket{source_canonical_service="frontend",destination_canonical_service="shippingservice"}[1m])))
```

Options Legend: Auto Format: Time series Step: auto Type: Range Exemplars: false

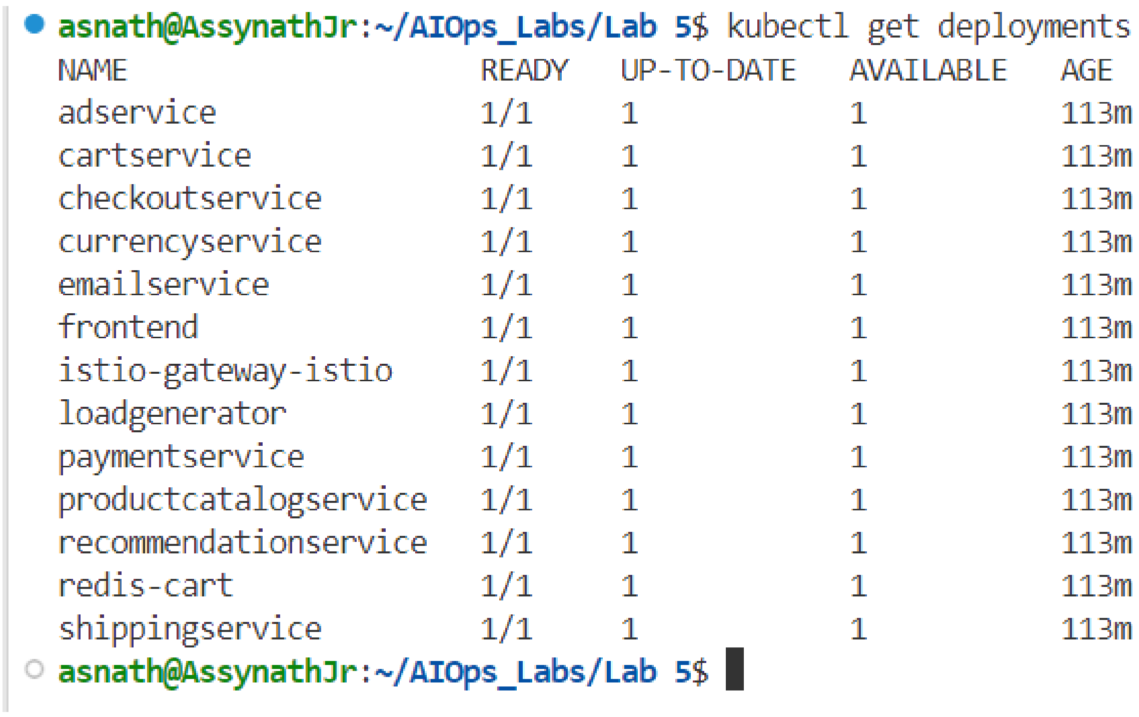
Figure 20: Histogram for the 0.5 percentile on Grafana


Search or jump to...
ctrl+k
Home › Dashboards › New dashboard › Edit panel
Table view
Fill
Actual
Last 30 minutes
Histogram for the 0.5 percentile
0 0 48 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 30.7 0 17.3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.5 1 5 10 25 50 100 250 500 10... 25... 50... 10... 30... 60... 30... 60... 18... 36... +Inf 0.5 1 5 10 25 50 100 250 500 10... 25... 50... 10... 30... 60... 30... 60... 18... 36... +Inf
Query 1
Transform data 0
Metrics browser ›

```
increase(istio_request_duration_milliseconds_bucket{source_canonical_service="frontend",destination_canonical_service="shippingservice"}[1m])
```

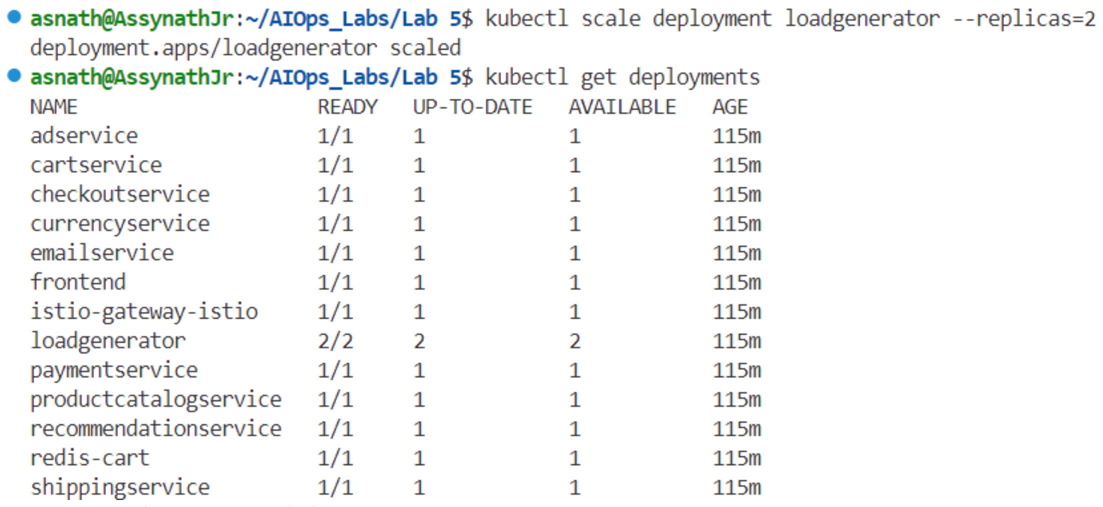
Options Legend: Auto Format: Heatmap Step: auto Type: Range Exemplars: false

Figure 21: Histogram view in Grafana



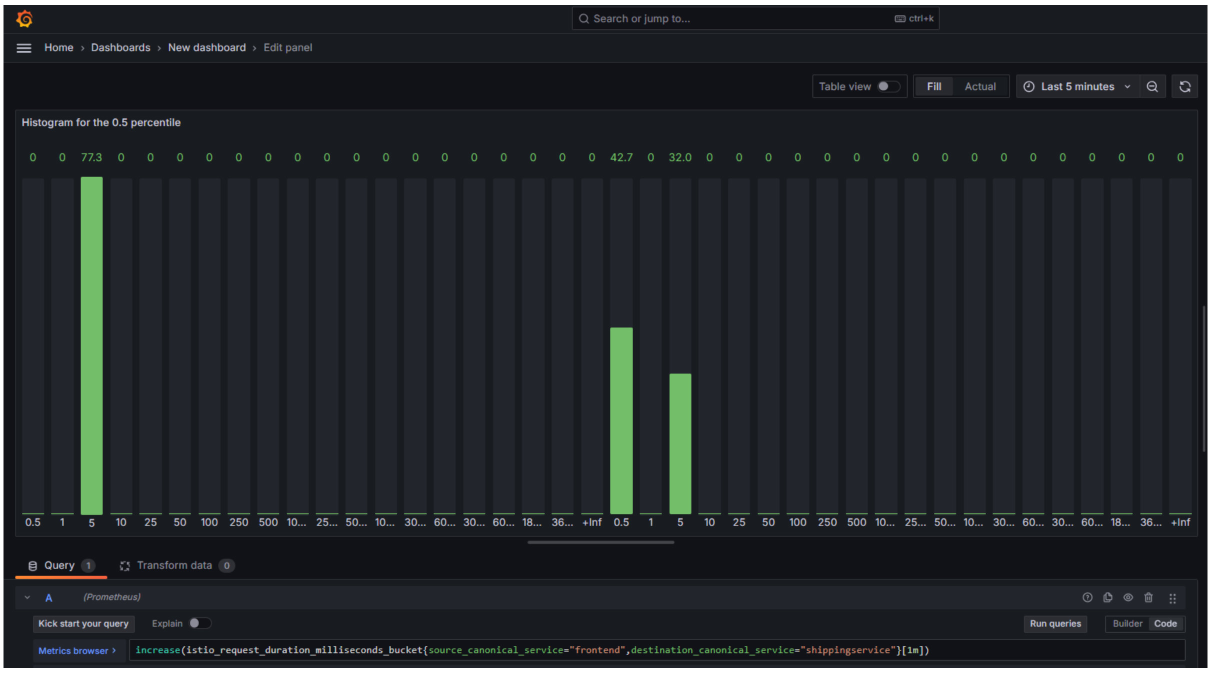
```
● asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl get deployments
  NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
  adservice               1/1     1            1           113m
  cartservice             1/1     1            1           113m
  checkoutservice         1/1     1            1           113m
  currencyservice         1/1     1            1           113m
  emailservice            1/1     1            1           113m
  frontend                1/1     1            1           113m
  istio-gateway-istio     1/1     1            1           113m
  loadgenerator           1/1     1            1           113m
  paymentservice          1/1     1            1           113m
  productcatalogservice   1/1     1            1           113m
  recommendationservice   1/1     1            1           113m
  redis-cart              1/1     1            1           113m
  shippingservice         1/1     1            1           113m
○ asnath@AssynathJr:~/AIOps_Labs/Lab 5$ 
```

Figure 22: Checking the deployments.



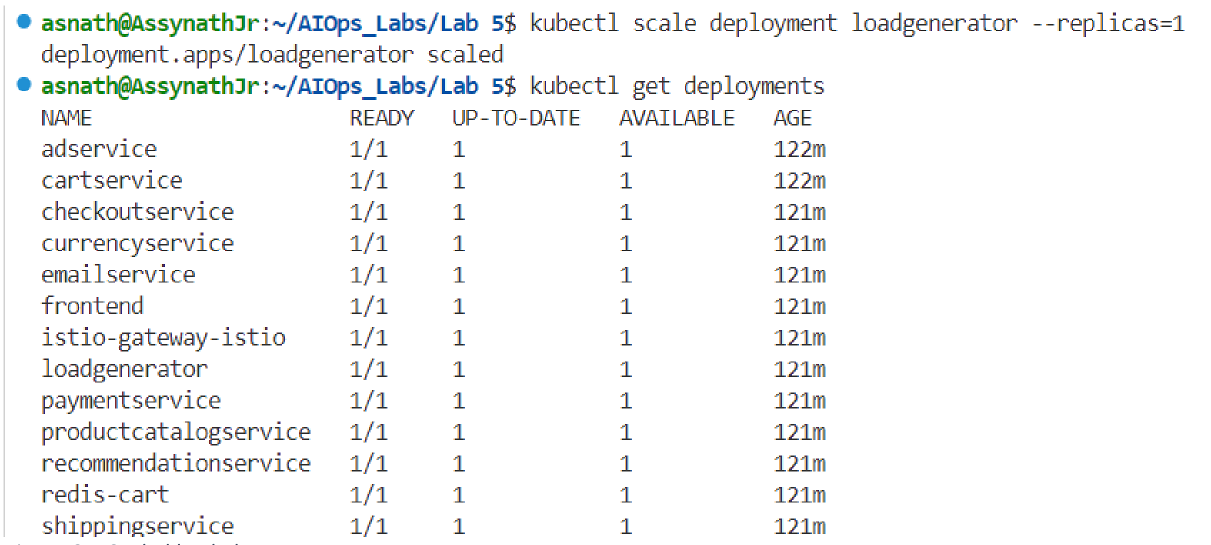
```
● asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl scale deployment loadgenerator --replicas=2
  deployment.apps/loadgenerator scaled
● asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl get deployments
  NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
  adservice               1/1     1            1           115m
  cartservice             1/1     1            1           115m
  checkoutservice         1/1     1            1           115m
  currencyservice         1/1     1            1           115m
  emailservice            1/1     1            1           115m
  frontend                1/1     1            1           115m
  istio-gateway-istio     1/1     1            1           115m
  loadgenerator           2/2     2            2           115m
  paymentservice          1/1     1            1           115m
  productcatalogservice   1/1     1            1           115m
  recommendationservice   1/1     1            1           115m
  redis-cart              1/1     1            1           115m
  shippingservice         1/1     1            1           115m
```

Figure 23: Load generator scaled.


Search or jump to...
ctrl+k
Home › Dashboards › New dashboard › Edit panel
Table view
Fill
Actual
Last 5 minutes
Histogram for the 0.5 percentile
0 0 77.3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 42.7 0 32.0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.5 1 5 10 25 50 100 250 500 10... 25... 50... 10... 30... 60... 30... 60... 18... 36... +Inf 0.5 1 5 10 25 50 100 250 500 10... 25... 50... 10... 30... 60... 30... 60... 18... 36... +Inf
Query 1
Transform data 0
A
(Prometheus)
Kick start your query
Explain
Run queries
Builder
Code
Metrics browser ›

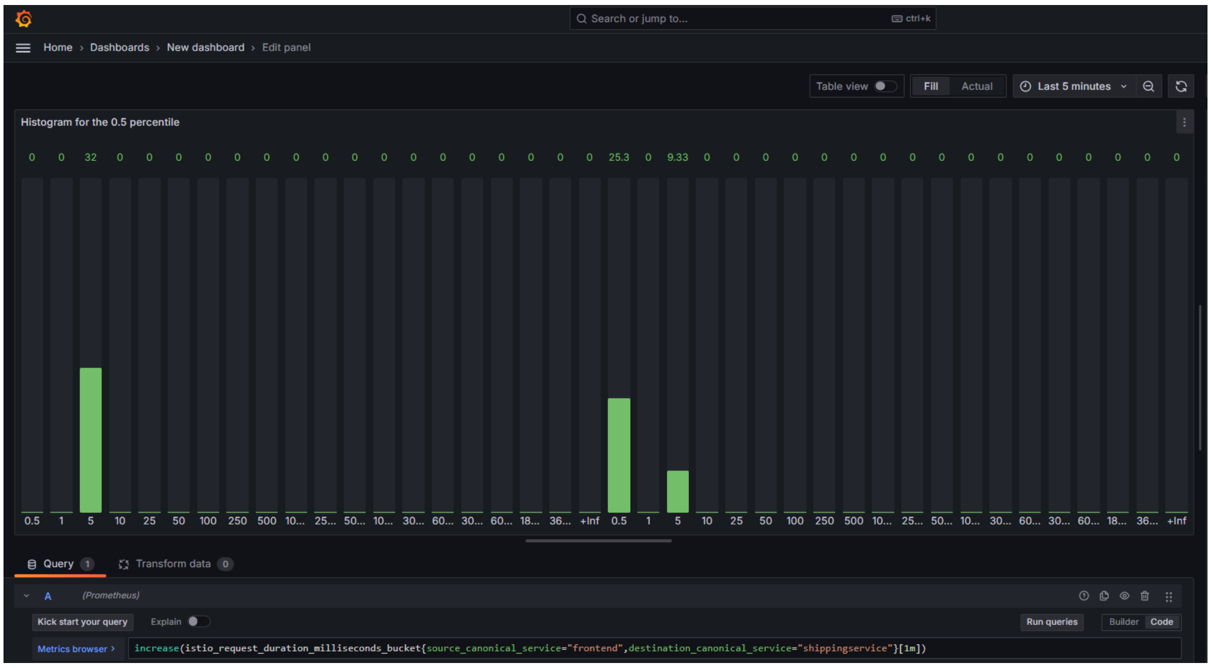
```
increase(istio_request_duration_milliseconds_bucket{source_canonical_service="frontend",destination_canonical_service="shippingservice"}[1m])
```

Figure 24: After scaling



```
● asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl scale deployment loadgenerator --replicas=1
  deployment.apps/loadgenerator scaled
● asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl get deployments
  NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
  adservice               1/1     1            1           122m
  cartservice             1/1     1            1           122m
  checkoutservice         1/1     1            1           121m
  currencyservice         1/1     1            1           121m
  emailservice            1/1     1            1           121m
  frontend                1/1     1            1           121m
  istio-gateway-istio     1/1     1            1           121m
  loadgenerator           1/1     1            1           121m
  paymentservice          1/1     1            1           121m
  productcatalogservice   1/1     1            1           121m
  recommendationservice   1/1     1            1           121m
  redis-cart              1/1     1            1           121m
  shippingservice         1/1     1            1           121m
```

Figure 25: Scaled back down.



```
increase(istio_request_duration_milliseconds_bucket{source_canonical_service="frontend",destination_canonical_service="shippingservice"}[1m])
```

Figure 26: After scaling back down on Grafana

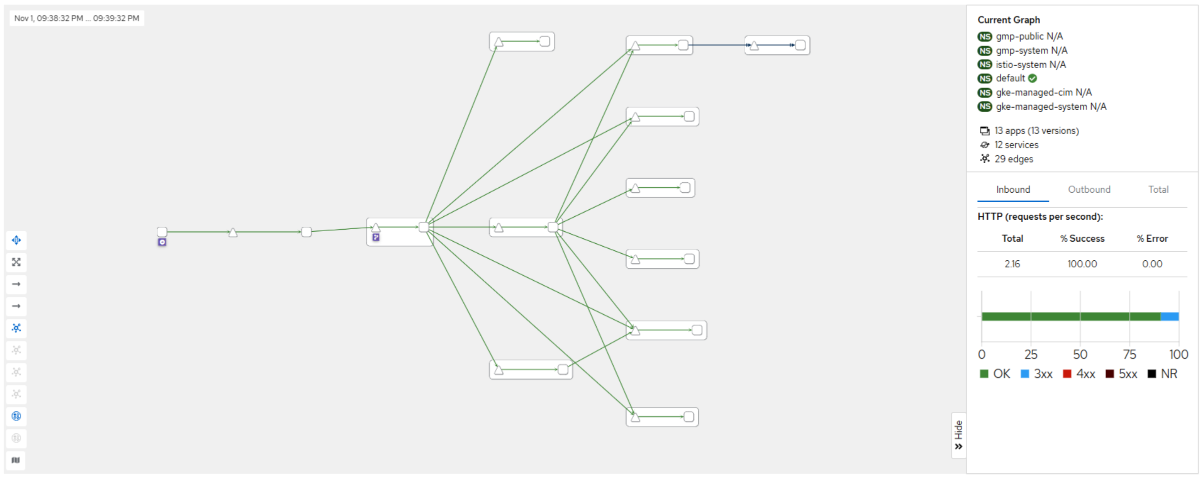
Why do duplicated entries appear for requests between services?

- Duplicates happen because Prometheus tracks each request using multiple labels, like the names and versions of both the sending and receiving services. Each unique combination of these labels creates a new data entry, even if it's the same request. To reduce these duplicates, we can use a function like sum by (le) to combine them, or we can filter by just one label, such as the source or destination, for a single view.

Convert to Incremental Rate (1m) and Reduce Duplicates:

- To make cumulative data show changes per minute, we use rate(metric_name[1m]), which calculates the request rate over each 1 minute. We can focus on a specific label that limits data to one source-to-destination pair to avoid duplicate entries, giving us a clearer view without extra data.
-

Lab Task 2: Install the Kiali cluster visualization dashboard.



Current Graph
NS gmp-public N/A
NS gmp-system N/A
NS istio-system N/A
NS default ✓
NS gke-managed-cim N/A
NS gke-managed-system N/A

13 apps (13 versions)
12 services
29 edges

Inbound | Outbound | Total

HTTP (requests per second):

Total	% Success	% Error
2.16	100.00	0.00

Figure 27: Visualizing on Kiali

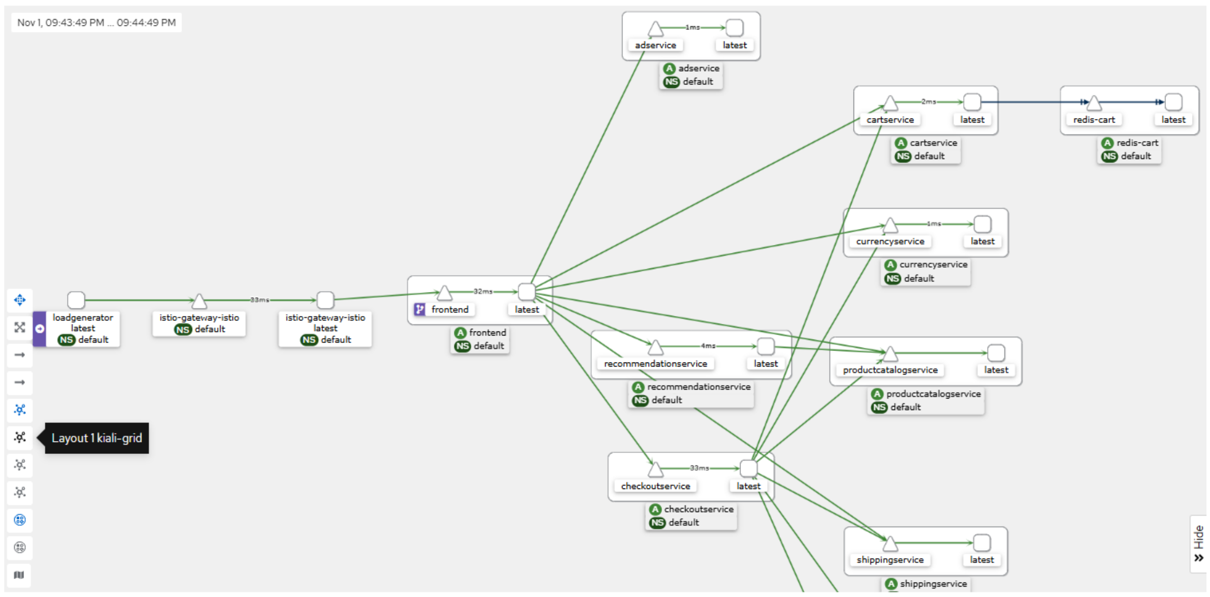
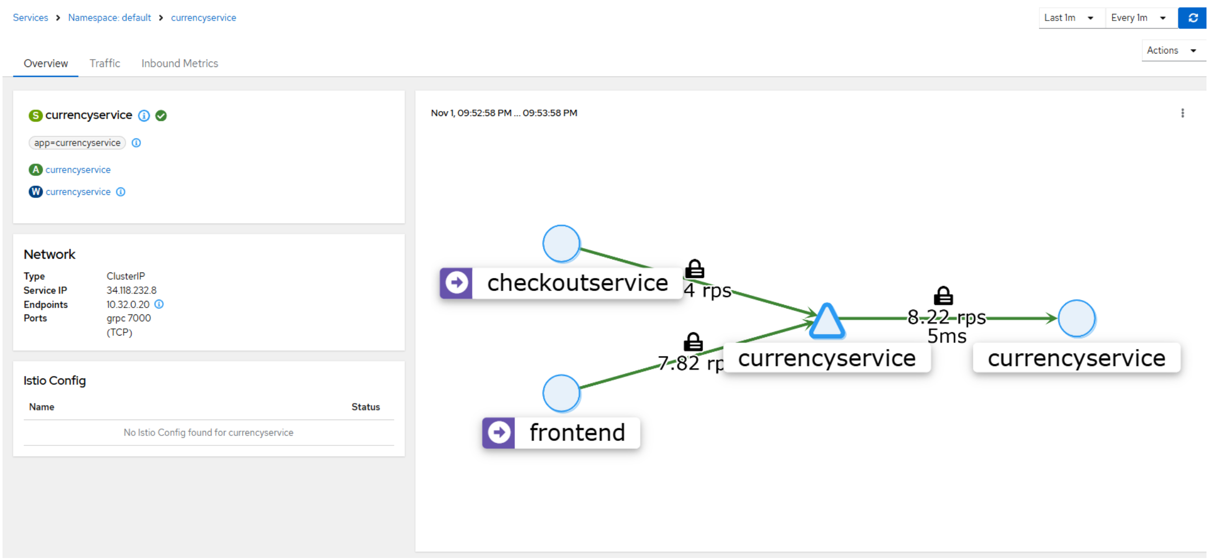


Figure 28: Zoomed in Visualization


Services > Namespace: default > currencyservice

Last 1m · Every 1m · Actions

Overview · Traffic · Inbound Metrics

currencyservice

app=currencyservice

currencyservice

currencyservice

Network

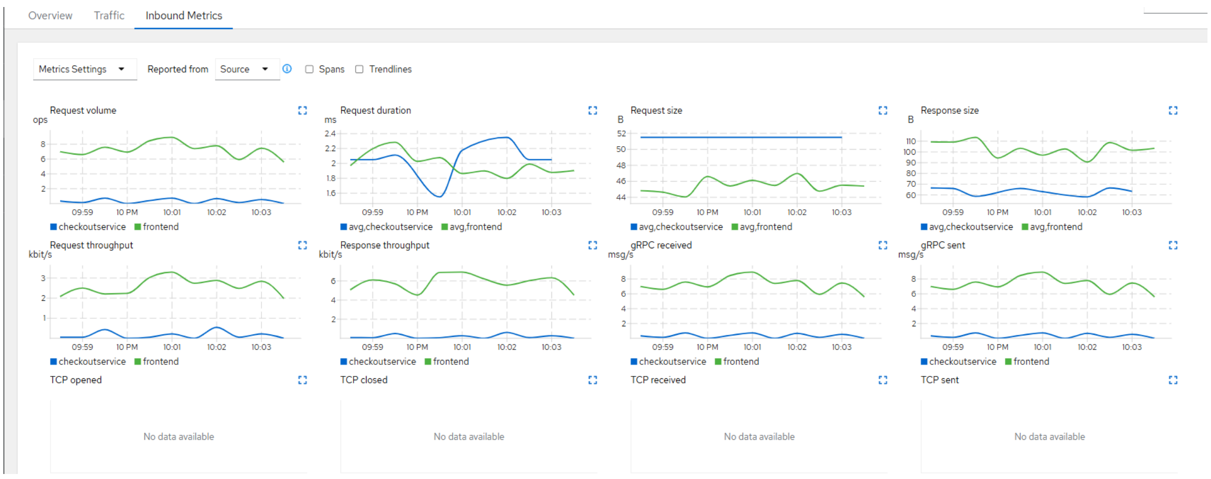
Type	ClusterIP
Service IP	34.118.232.8
Endpoints	10.32.0.20
Ports	grpc 7000 (TCP)

Istio Config

Name	Status
No Istio Config found for currencyservice	

Nov 1, 09:52:58 PM ... 09:53:58 PM

Figure 29: Currency Service


Overview · Traffic · Inbound Metrics

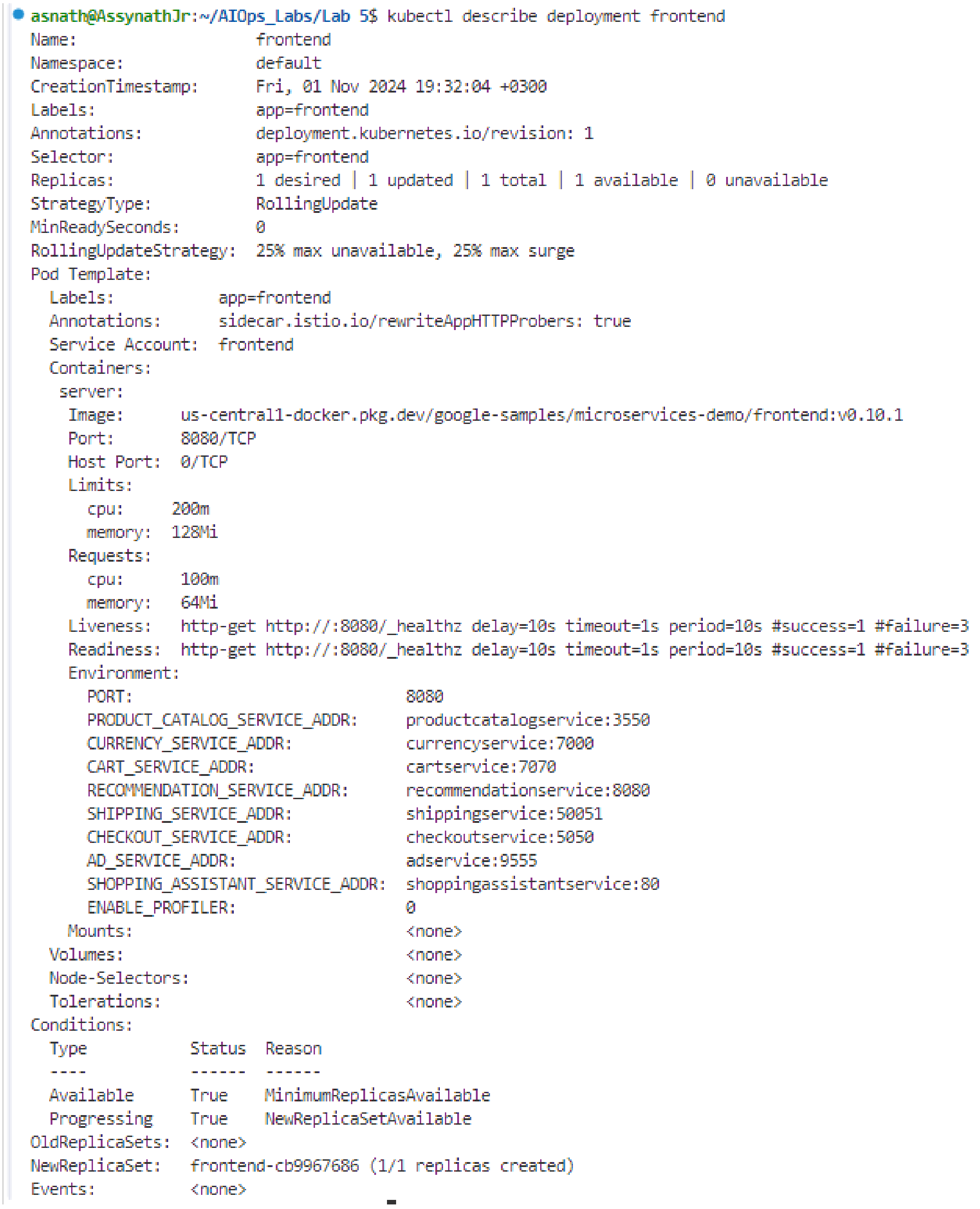
Metrics Settings · Reported from Source · Spans · Trendlines

Figure 30: inbound metrics for currency service

Impact of Fault Injection Observed in Kiali:

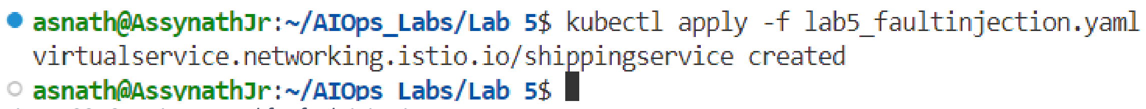
- After adding a fault injection, we can see delays in Kiali by looking at the Inbound Metrics for `shippingservice`. Requests coming from `frontend` show longer response times, while requests from other services, like `checkoutservice`, don't have delays. This shows that the fault injection only affected requests from `frontend`, as intended.

Lab Task 3: Inject selective delay faults to the Boutique shipping service.



```
asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl describe deployment frontend
Name:                   frontend
Namespace:              default
CreationTimestamp:      Fri, 01 Nov 2024 19:32:04 +0300
Labels:                 app=frontend
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=frontend
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:           app=frontend
  Annotations:      sidecar.istio.io/rewriteAppHTTPProbers: true
  Service Account:  frontend
  Containers:
   server:
    Image:      us-central1-docker.pkg.dev/google-samples/microservices-demo/frontend:v0.10.1
    Port:       8080/TCP
    Host Port:  0/TCP
    Limits:
      cpu:     200m
      memory:  128Mi
    Requests:
      cpu:      100m
      memory:   64Mi
    Liveness:   http-get http://:8080/_healthz delay=10s timeout=1s period=10s #success=1 #failure=3
    Readiness:  http-get http://:8080/_healthz delay=10s timeout=1s period=10s #success=1 #failure=3
    Environment:
      PORT:                              8080
      PRODUCT_CATALOG_SERVICE_ADDR:      productcatalogservice:3550
      CURRENCY_SERVICE_ADDR:             currencyservice:7000
      CART_SERVICE_ADDR:                 cartservice:7070
      RECOMMENDATION_SERVICE_ADDR:       recommendationservice:8080
      SHIPPING_SERVICE_ADDR:             shippingservice:50051
      CHECKOUT_SERVICE_ADDR:             checkoutservice:5050
      AD_SERVICE_ADDR:                   adservice:9555
      SHOPPING_ASSISTANT_SERVICE_ADDR:   shoppingassistantservice:80
      ENABLE_PROFILER:                   0
    Mounts:                              <none>
  Volumes:                               <none>
  Node-Selectors:                        <none>
  Tolerations:                           <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   frontend-cb9967686 (1/1 replicas created)
Events:          <none>
```

Figure 31: Kubectl getting the description of the frontend.



```
asnath@AssynathJr:~/AIOps_Labs/Lab 5$ kubectl apply -f lab5_faultinjection.yaml
virtualservice.networking.istio.io/shippingservice created
asnath@AssynathJr:~/AIOps_Labs/Lab 5$
```

Figure 32: Creating a yaml for fault injection

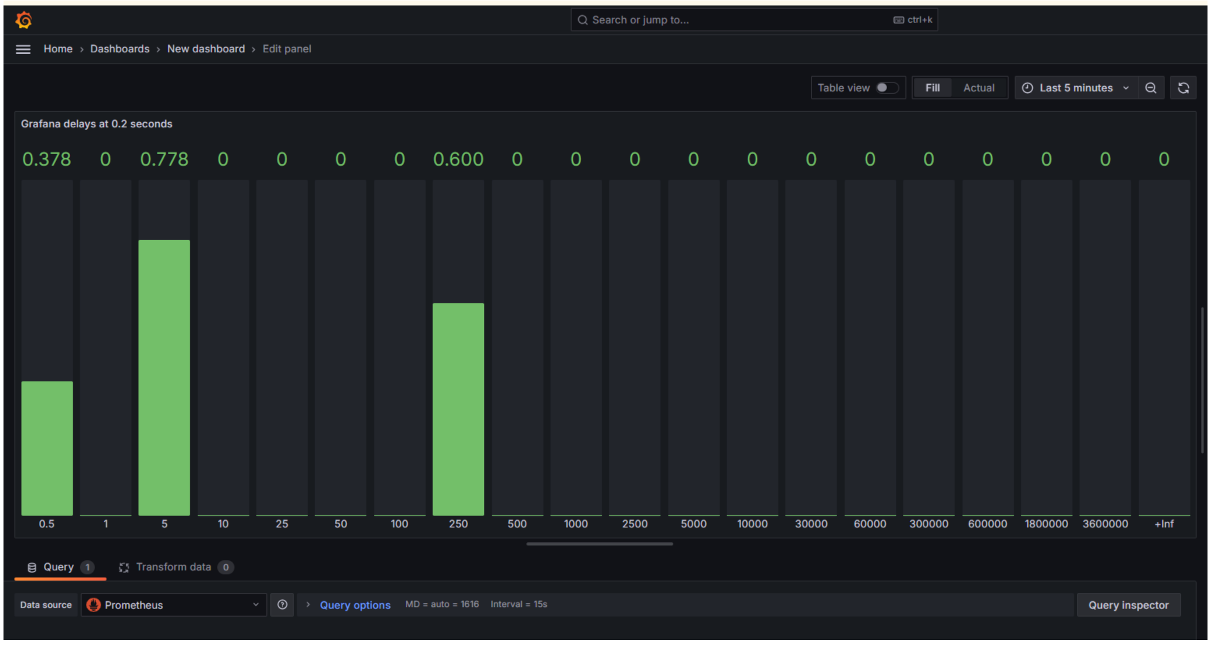


Figure 33: Grafana delays by 0.2 seconds

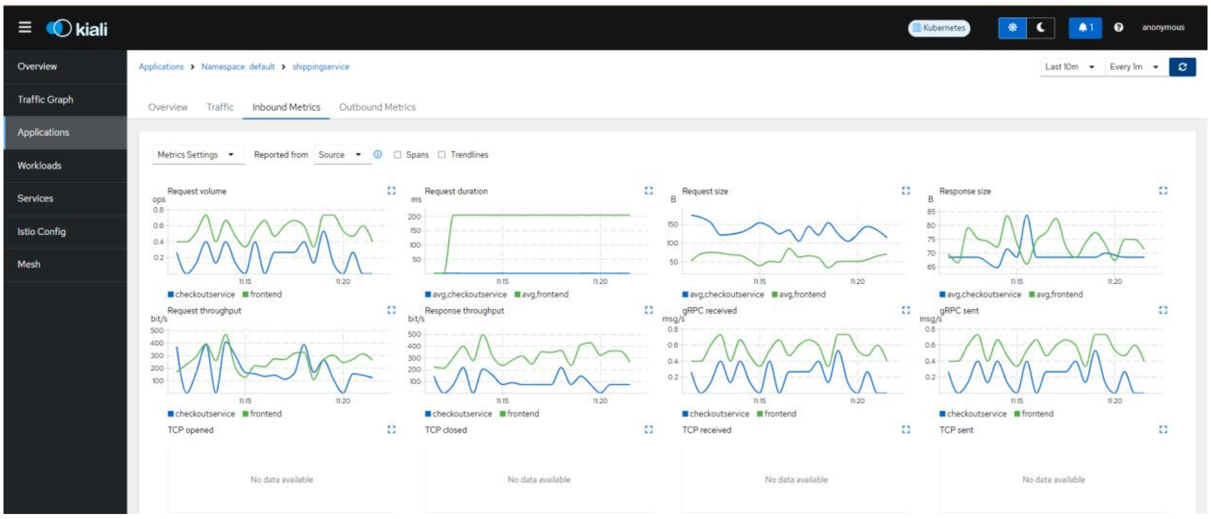
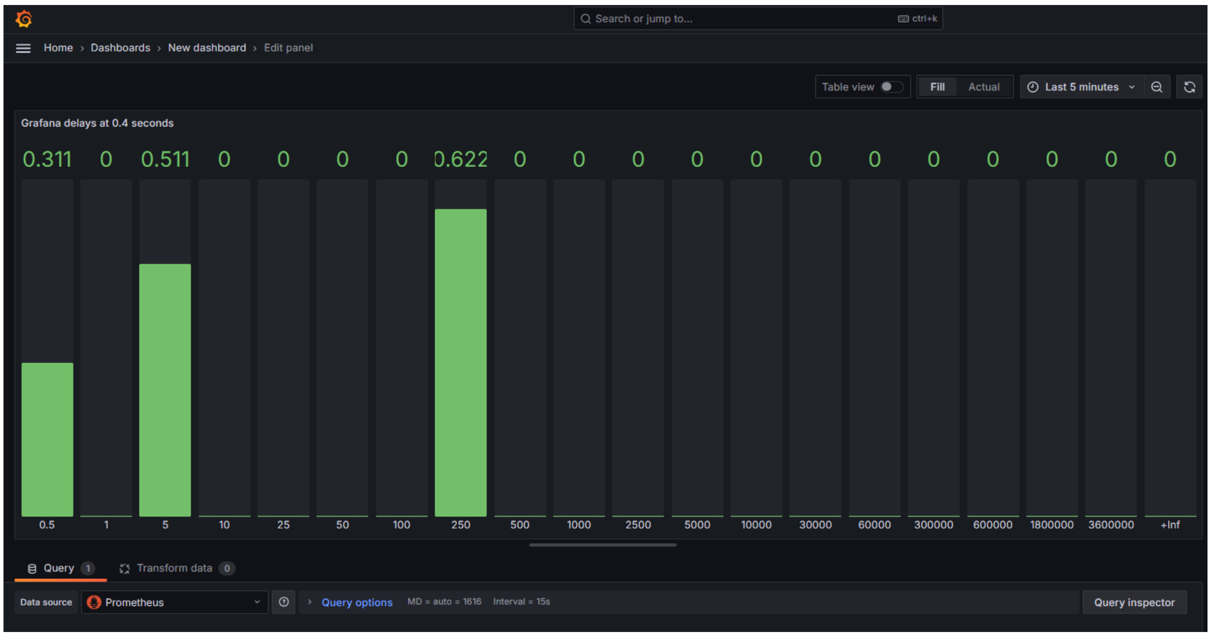


Figure 34: Kiali delays



Search or jump to...
ctrl+k
Home › Dashboards › New dashboard › Edit panel
Table view
Fill
Actual
Last 5 minutes
Grafana delays at 0.4 seconds
0.311 0 0.511 0 0 0 0 0.622 0 0 0 0 0 0 0 0 0 0 0 0
0.5 1 5 10 25 50 100 250 500 1000 2500 5000 10000 30000 60000 300000 600000 1800000 3600000 +Inf
Query 1
Transform data 0
Data source
Prometheus
Query options
MD = auto = 1616 Interval = 15s
Query inspector

Figure 35: Grafana delays at 0.4

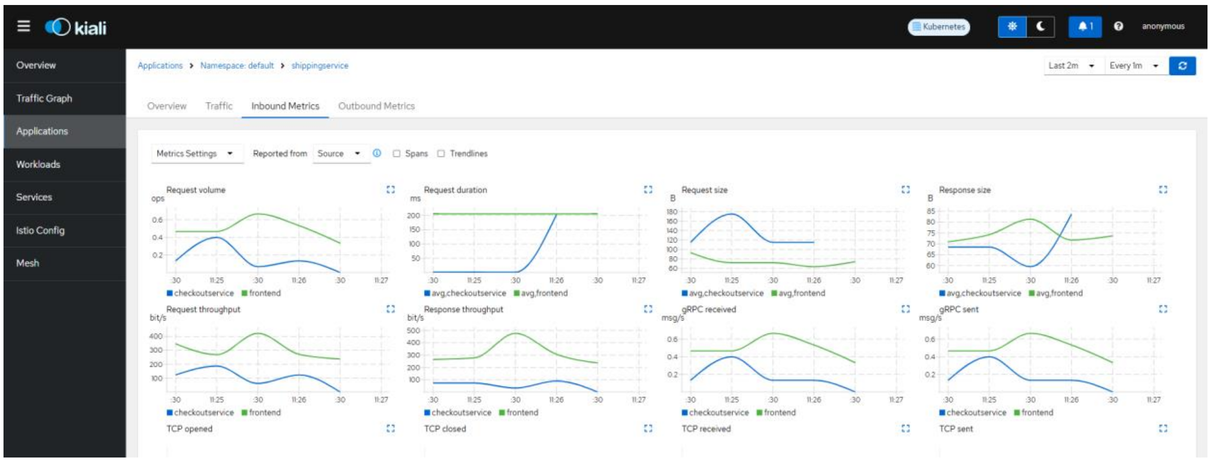


Figure 36: 0.4 delays on Kiali. Where the request duration jumped to meet the frontend service.

Grafana Histogram with Both Delays:

- In Grafana, the histogram for `shippingservice` shows two peaks: one at 0.2 seconds for delays from `frontend` and another at 0.4 seconds for delays from `checkoutservice` on specific requests like "GetQuote." Using `sum by (le)` groups these delay data points into one histogram, so we don't see duplicates from Istio's added delay labels.

Where is the Delay Imposed (Kiali)?

- In Kiali, looking at the Source metrics for `shippingservice` shows that the delay affects the whole path from `frontend` to `shippingservice`. This means that Istio applies the delay at the network level before it reaches the application code, as we can see by comparing the Source and Destination metrics.