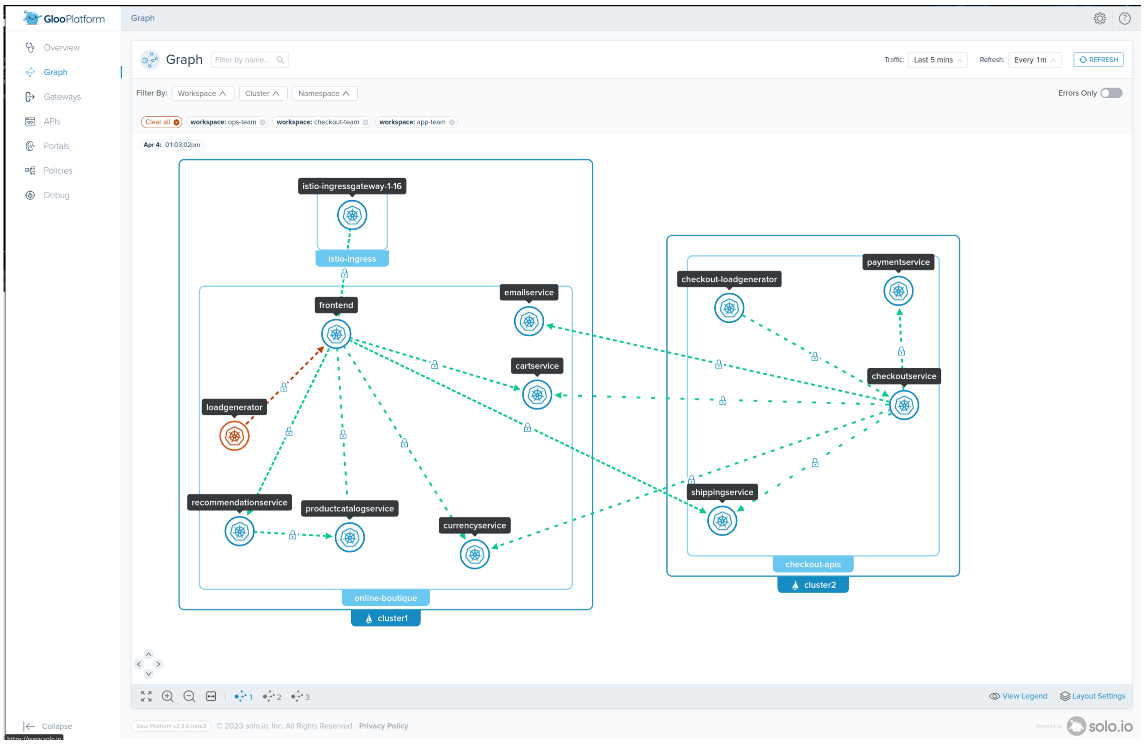
Lab 11 - Observability

Gloo Platform offers the best in class observability for communication between your services. By aggregating logging, tracing, and metrics, Gloo Platform combines telemetry from all environments to give you a complete picture of how your platform is performing.



Links:

- Exploying the Gloo UI
- Prometheus
- Gloo Platform Tracing/Metrics/Logs

Gloo Platform UI

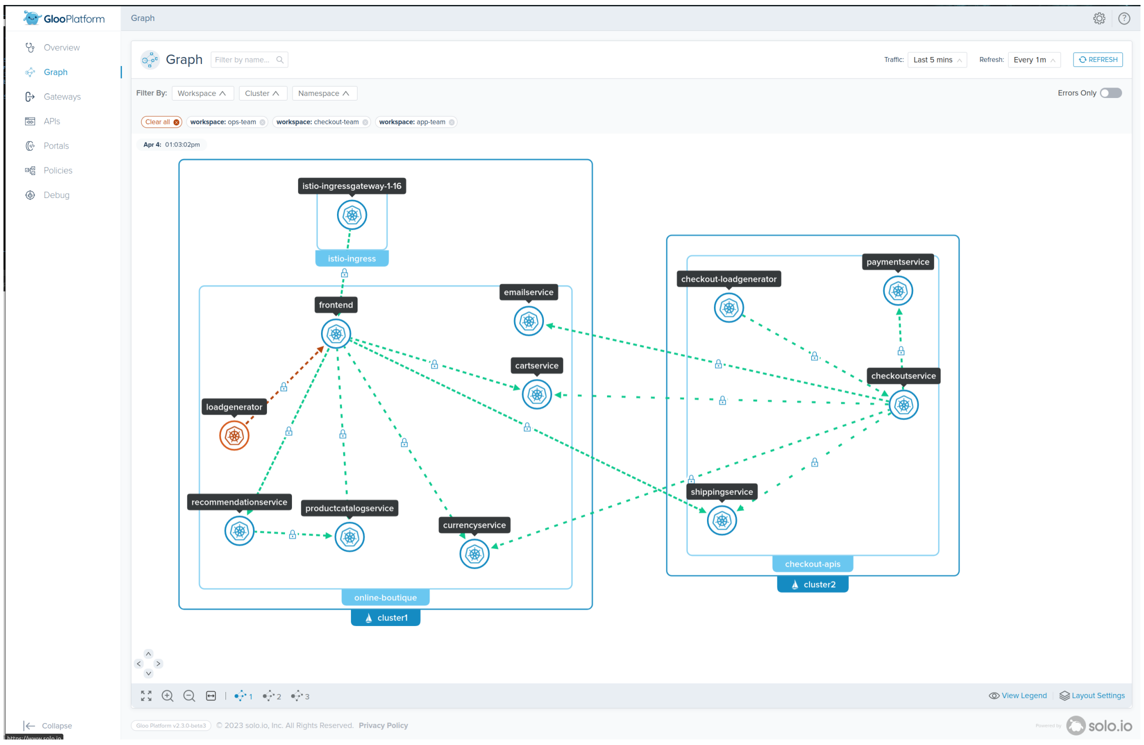
The Gloo UI is automatically installed in the Gloo management cluster. Let's explore some of the key features that you have access to when using the Gloo UI:

- **Gloo Platform overview: With the Gloo UI:** you can view information about your Gloo Platform environment, such as the number of clusters that are registered with the Gloo management server and the Istio version that is deployed to them. You can also review your workspace settings and which Gloo resources you import and export to other workspaces.
- **Verify service mesh configurations:** The Gloo Mesh UI lets you quickly find important information about your service mesh setup, the participating clusters, and the Gloo Mesh resources that you applied to the service mesh. For example, you can review your workspace settings and which Gloo Mesh resources you import and export to other workspaces. You can also view your gateway and listener configurations and traffic policies that you applied to your service mesh routes.
- **Drill into apps and services:** Review what services can communicate with other services in the mesh, the policies that are applied before traffic is sent to a service, and how traffic between services is secured.

- **Visualize and monitor service mesh health metrics:** With the built-in Prometheus integration, the Gloo Mesh UI has access to detailed Kubernetes cluster and service mesh metrics, such as the node's CPU and memory capacity, unresponsive nodes or nodes with degraded traffic, and mTLS settings for services in the mesh. For more information about the Prometheus integration in Gloo Mesh, see Prometheus.
- **OIDC authentication and authorization support:** Set up authentication and authorization (AuthN/AuthZ) for the Gloo Mesh UI by using OpenID Connect (OIDC) and Kubernetes role-based access control (RBAC). The Gloo Mesh UI supports OpenID Connect (OIDC) authentication from common providers such as Google, Okta, and Auth0.

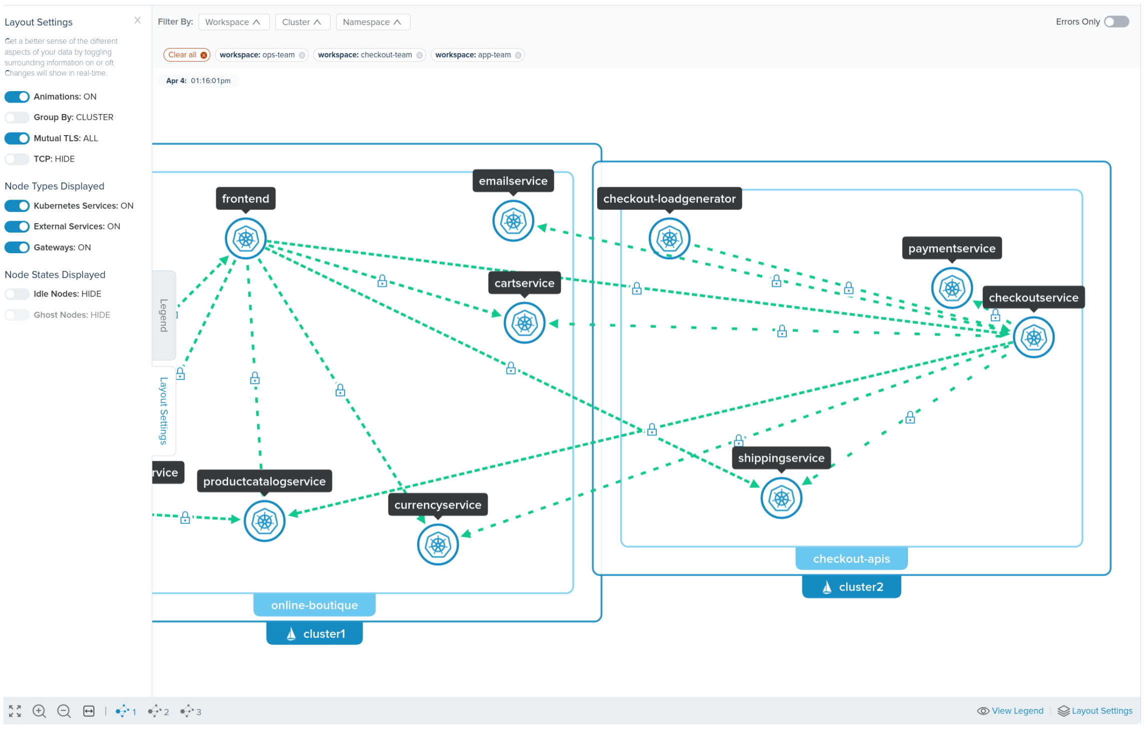
For a detailed overview of what information you can find in the Gloo UI, see Explore the Gloo UI.

- Navigate to Gloo Mesh Graph



```
kubectl port-forward svc/gloo-mesh-ui 8090:8090 --context management -n gloo-mesh
echo "Gloo UI: http://localhost:8090"
```

- Click `Workspaces` and select all workspaces
- Click `Layout Settings` in the bottom right of the screen and update `Group By:` to `CLUSTER`


Layout Settings

Get a better sense of the different aspects of your data by toggling surrounding information on or off. Changes will show in real-time.

Animations: ON

Group By: CLUSTER

Mutual TLS: ALL

TCP: HIDE

Node Types Displayed

Kubernetes Services: ON

External Services: ON

Gateways: ON

Node States Displayed

Idle Nodes: HIDE

Ghost Nodes: HIDE

Filter By: Workspace Cluster Namespace

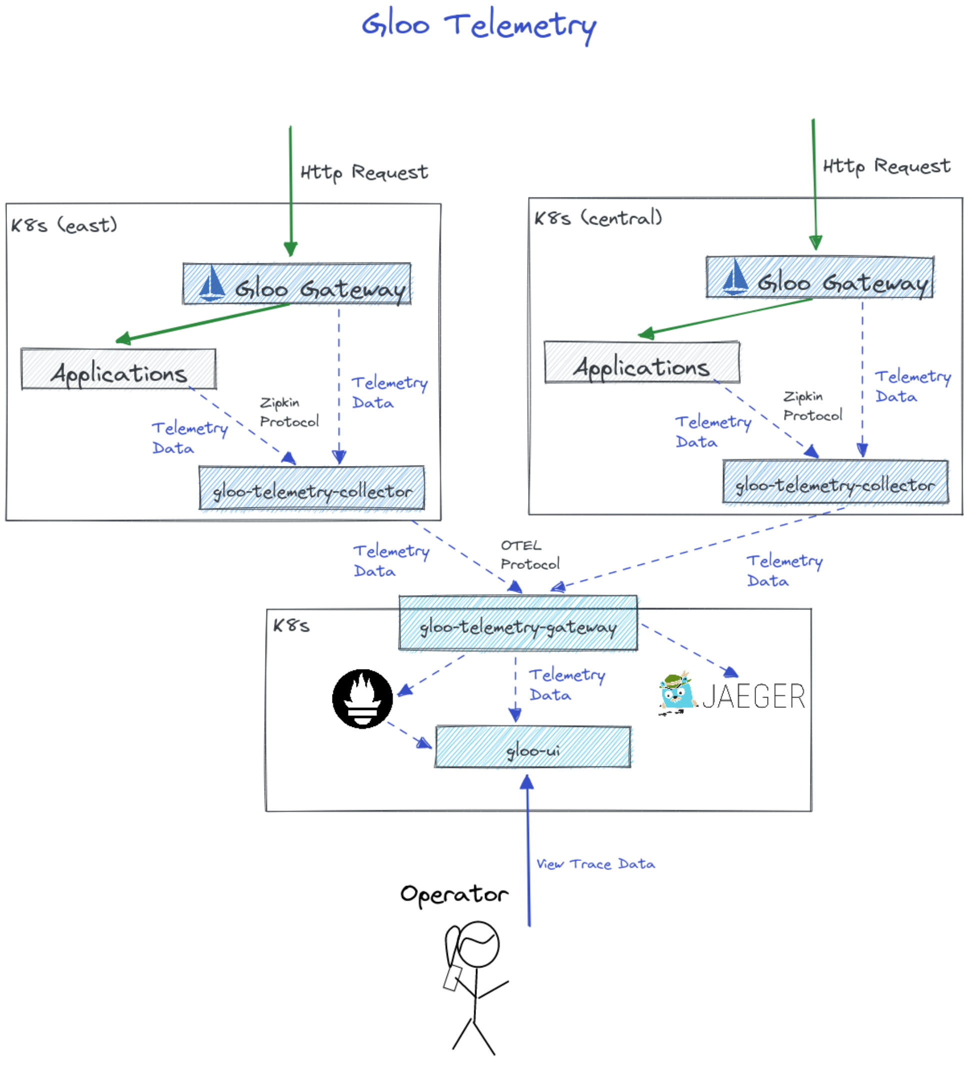
Clear all | workspace: ops-team | workspace: checkout-team | workspace: app-team

Apr 4: 01:16:01pm

Errors Only

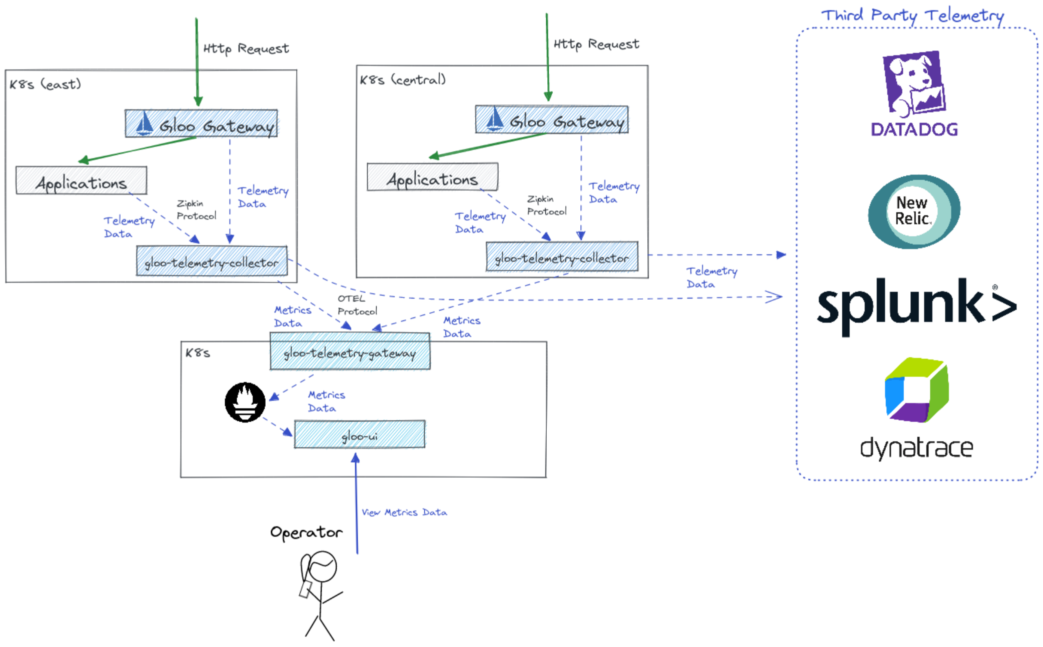
Telemetry

Gloo Platform Telemetry is based on the Open Telemetry standards. Metrics, logs and traces are gathered in cluster by the `Gloo Telemetry Collector`. Where it is shipped from there is up the user. By default, telemetry data will be shipped to the `Gloo Telemetry Gateway` in the `Management Plane` and aggregated with all other telemetry data from other clusters.



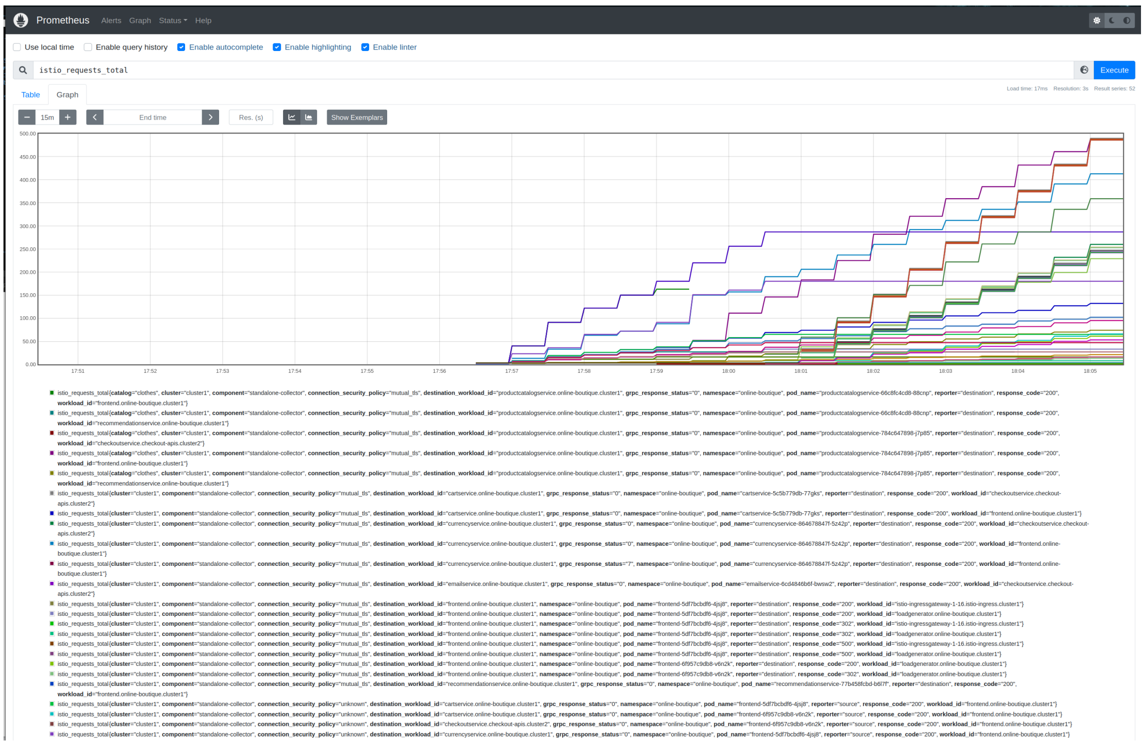
Third Party Support

Gloo Platform can also ship these metrics to you third party telemetry provider like `Datadog` , `Splunk` , `New Relic` , or `Dynatrace` .



Metrics

Metrics provide important information about the health of the apps in your service mesh. You can use these measures to detect failures, troubleshoot bottlenecks, and to find ways to improve the performance and reliability of your service mesh.



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

```
istio_requests_total
```

Load time: 17ms Resolution: 3s Result series: 52

```
istio_requests_total{catalog="clothes", cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="productcatalogservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="productcatalogservice-66c8fc4cd8-88cnp", reporter="destination", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{catalog="clothes", cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="productcatalogservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="productcatalogservice-66c8fc4cd8-88cnp", reporter="destination", response_code="200", workload_id="recommendationservice.online-boutique.cluster1"}
istio_requests_total{catalog="clothes", cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="productcatalogservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="productcatalogservice-784c647898-j7p85", reporter="destination", response_code="200", workload_id="checkoutservice.checkout-apis.cluster2"}
istio_requests_total{catalog="clothes", cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="productcatalogservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="productcatalogservice-784c647898-j7p85", reporter="destination", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{catalog="clothes", cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="productcatalogservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="productcatalogservice-784c647898-j7p85", reporter="destination", response_code="200", workload_id="recommendationservice.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="cartservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="cartservice-5c5b779db-77gks", reporter="destination", response_code="200", workload_id="checkoutservice.checkout-apis.cluster2"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="cartservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="cartservice-5c5b779db-77gks", reporter="destination", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="currencyservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="currencyservice-864678847f-5z42p", reporter="destination", response_code="200", workload_id="checkoutservice.checkout-apis.cluster2"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="currencyservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="currencyservice-864678847f-5z42p", reporter="destination", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="currencyservice.online-boutique.cluster1", grpc_response_status="7", namespace="online-boutique", pod_name="currencyservice-864678847f-5z42p", reporter="destination", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="emailservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="emailservice-6cd4846b6f-bwsw2", reporter="destination", response_code="200", workload_id="checkoutservice.checkout-apis.cluster2"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="destination", response_code="200", workload_id="istio-ingressgateway-1-16.istio-ingress.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="destination", response_code="200", workload_id="loadgenerator.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="destination", response_code="302", workload_id="istio-ingressgateway-1-16.istio-ingress.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="destination", response_code="302", workload_id="loadgenerator.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="destination", response_code="500", workload_id="istio-ingressgateway-1-16.istio-ingress.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="destination", response_code="500", workload_id="loadgenerator.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-6f957c9db8-v6n2k", reporter="destination", response_code="200", workload_id="loadgenerator.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="frontend.online-boutique.cluster1", namespace="online-boutique", pod_name="frontend-6f957c9db8-v6n2k", reporter="destination", response_code="302", workload_id="loadgenerator.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="mutual_tls", destination_workload_id="recommendationservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="recommendationservice-77b458fcbd-b6l7f", reporter="destination", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="unknown", destination_workload_id="cartservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="source", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="unknown", destination_workload_id="cartservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="frontend-6f957c9db8-v6n2k", reporter="source", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="unknown", destination_workload_id="checkoutservice.checkout-apis.cluster2", grpc_response_status="0", namespace="online-boutique", pod_name="frontend-6f957c9db8-v6n2k", reporter="source", response_code="200", workload_id="frontend.online-boutique.cluster1"}
istio_requests_total{cluster="cluster1", component="standalone-collector", connection_security_policy="unknown", destination_workload_id="currencyservice.online-boutique.cluster1", grpc_response_status="0", namespace="online-boutique", pod_name="frontend-5df7bcbdf6-4jsj8", reporter="source", response_code="200", workload_id="frontend.online-boutique.cluster1"}
```

- To view the raw metrics

```
kubectl port-forward svc/prometheus-server --context management -n gloo-mesh 9080:80
echo "Prometheus available at http://localhost:9080"
```

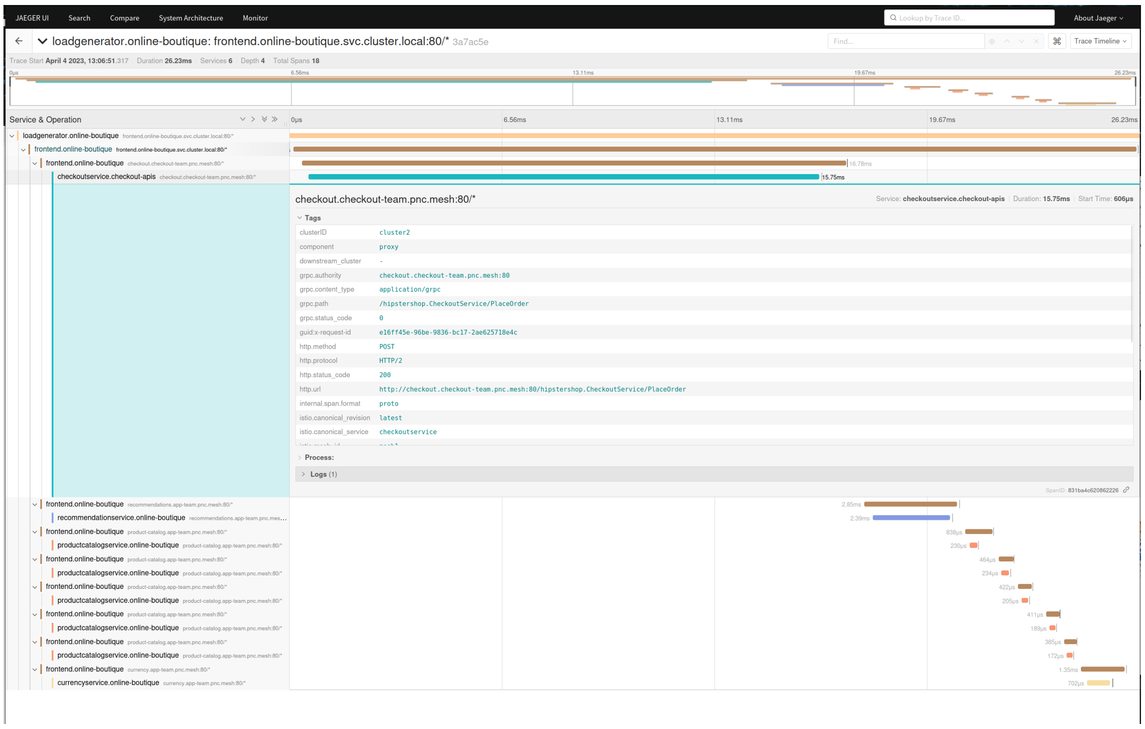
- Search for metrics prefixed with `istio_` , Example `istio_requests_total`
- Here is an example query that lists non 200 codes for each workload

```
sum(rate(istio_requests_total{response_code!="200"}[1m])) by
(workload_id,response_code)
```

Tracing

Sample request traces to monitor the traffic and health of your service mesh.

Distributed tracing helps you track requests across multiple services in your distributed service mesh. Sampling the requests in your mesh can give you insight into request latency, serialization, and parallelism



loadgenerator.online-boutique: frontend.online-boutique.svc.cluster.local:80/* 3a7ac5e

Trace Start **April 4 2023, 13:06:51**.317 | Duration **26.23ms** | Services **6** | Depth **4** | Total Spans **18**

```
loadgenerator.online-boutique  frontend.online-boutique.svc.cluster.local:80/*
  frontend.online-boutique  frontend.online-boutique.svc.cluster.local:80/*
    frontend.online-boutique  checkout.checkout-team.pnc.mesh:80/*
      checkoutservice.checkout-apis  checkout.checkout-team.pnc.mesh:80/*
    frontend.online-boutique  recommendations.app-team.pnc.mesh:80/*
      recommendationservice.online-boutique  recommendations.app-team.pnc.mes...
    frontend.online-boutique  product-catalog.app-team.pnc.mesh:80/*
      productcatalogservice.online-boutique  product-catalog.app-team.pnc.mesh:80/*
    frontend.online-boutique  product-catalog.app-team.pnc.mesh:80/*
      productcatalogservice.online-boutique  product-catalog.app-team.pnc.mesh:80/*
    frontend.online-boutique  product-catalog.app-team.pnc.mesh:80/*
      productcatalogservice.online-boutique  product-catalog.app-team.pnc.mesh:80/*
    frontend.online-boutique  product-catalog.app-team.pnc.mesh:80/*
      productcatalogservice.online-boutique  product-catalog.app-team.pnc.mesh:80/*
    frontend.online-boutique  product-catalog.app-team.pnc.mesh:80/*
      productcatalogservice.online-boutique  product-catalog.app-team.pnc.mesh:80/*
    frontend.online-boutique  currency.app-team.pnc.mesh:80/*
      currencyservice.online-boutique  currency.app-team.pnc.mesh:80/*
```

checkout.checkout-team.pnc.mesh:80/*

Service: **checkoutservice.checkout-apis** | Duration: **15.75ms** | Start Time: **606μs**

Tags

clusterID	cluster2
component	proxy
downstream_cluster	-
grpc.authority	checkout.checkout-team.pnc.mesh:80
grpc.content_type	application/grpc
grpc.path	/hipstershop.CheckoutService/PlaceOrder
grpc.status_code	0
guid:x-request-id	e16ff45e-96be-9836-bc17-2ae625718e4c
http.method	POST
http.protocol	HTTP/2
http.status_code	200
http.url	http://checkout.checkout-team.pnc.mesh:80/hipstershop.CheckoutService/PlaceOrder
internal.span.format	proto
istio.canonical_revision	latest
istio.canonical_service	checkoutservice

Process:

Logs (1)

SpanID: 831ba4c620862226

- Deploy Jaeger to the `management` cluster

```
helm repo add jaegertracing https://jaegertracing.github.io/helm-charts
helm repo update

helm upgrade -i jaeger jaegertracing/jaeger \
  --kube-context management \
  --version 0.71.7 \
  --namespace gloo-mesh \
  -f data/jaeger-values.yaml
```

- Create a service for Istio to send metrics

```
kubectl apply --context cluster-1 -f data/tracing-service.yaml
kubectl apply --context cluster-2 -f data/tracing-service.yaml
```

- Enable Tracing within each cluster

```
kubectl apply --context cluster-1 -f - <<EOF
apiVersion: telemetry.istio.io/v1alpha1
kind: Telemetry
metadata:
  name: default
  namespace: istio-system
spec:
  tracing:
  - providers:
      - name: zipkincustom
    randomSamplingPercentage: 100
    disableSpanReporting: false
  accessLogging:
  - providers:
    - name: envoyOtelAls
EOF
kubectl apply --context cluster-2 -f - <<EOF
apiVersion: telemetry.istio.io/v1alpha1
kind: Telemetry
metadata:
  name: default
  namespace: istio-system
spec:
  tracing:
  - providers:
      - name: zipkincustom
    randomSamplingPercentage: 100
    disableSpanReporting: false
  accessLogging:
  - providers:
    - name: envoyOtelAls
EOF
```

- Open Jaeger and observe tracing

```
kubectl port-forward svc/jaeger-query --context management -n gloo-mesh 9081:16686
echo "Jaeger available at http://localhost:9081"
```

- Open browser at http://localhost:9081

- Select service `frontend.online-boutique` and then click `Find Traces`