

Introduction to Cloud Computing – Exercise 8

Scope: Istio traffic shifting, Visualizing Metrics with Grafana, Visualizing Mesh

Admission:

The aim of the laboratory is to get acquainted with the redirection of network traffic to individual microservices, visualization of metrics with Grafana and visualization of our network.

To perform the following exercises, you should have a virtual machine with a minicube installed and running, and an example from the previous lab running:

1. Traffic shifting

This task shows how to move traffic from one version of a microservice to another.

A common use case is the gradual migration of traffic from the older version of the microservice to the new one. In Istio, you achieve this goal by configuring a sequence of routing rules that redirect the percentage of traffic from one destination to another.

In this task, you apply a rule that sends 50% of traffic to review:v1 and 50% to review:v3. You will then complete the migration by redirecting 100% of your traffic to review:v3.

The first step is to redirect all network traffic to review:v1. Execute command

```
$ kubectl apply -f samples/bookinfo/networking/virtual-service-all-v1.yaml
```

Open the Bookinfo page in your browser. The URL is `http://$GATEWAY_URL/productpage`, where `$GATEWAY_URL` is the external address of our Bookinfo deployment (from the previous lab).

Note that the part of the review page shows without rating stars, no matter how many times you refresh it. This is because Istio is configured to direct all traffic from to review:v1, and this version of the service does not have access to the star rating service.

Redirect 50% of your traffic from review:v1 to review:v3 now. Do

```
$ kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-50-v3.yaml
```

Wait a few seconds for the rule to activate. Confirm that the new rule is running.

```
$ kubectl get virtualservice reviews -o yaml
```

What should the rule that redirects 75% of traffic to review:v3 look like?

Refresh /productpage in your browser, now in about 50% of cases you will see ratings in the form of red stars. This is because the v3 version of the reviews accesses the star rating service, but the v1 version does not.

With the current implementation of the Envoy sidecar, you may need to refresh /productpage multiple times – for example, 15 or more – to see the actual distribution. You can modify your rules to direct 90% of your traffic to version 3 to see red stars more often.

Assuming you decide that review:v3 is stable, you can direct 100% of your traffic to review:v3 using this virtual service:

```
$ kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-v3.yaml
```

Now, when you refresh the /productpage, you will always see book reviews with red star ratings for each review.

Explanation of what happened:

This task migrates traffic from the old to the new review using Istio weighted routing. Note that this is very different from migrating versions using container orchestration deployment features that use their own scaling of traffic to individual instances.

With Istio, you can allow two versions of review to scale up and down independently, without affecting the distribution of traffic between them.

2. Visualizing Metrics with Grafana

In this task, we will use the tools of Grafana and Prometheus. **Familiarize yourself and describe in a few words what Grafana and Prometheus are.**

Istio provides a basic sample installation of Grafana to launch Grafana easily and quickly, bundled with all Istio dashboards already installed:

```
$ kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.13/samples/addons/grafana.yaml
```

Please note that this installation is for demonstration purposes only and is not configured for performance or security.

Install Prometheus then

```
$ kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.13/samples/addons/prometheus.yaml
```

In the same way, the installation is created only for demonstration purposes.

You should also have the Bookinfo app deployed. Verify that *prometheus* is running.

```
$ kubectl -n istio-system get svc prometheus
```

Then verify that Grafana is running

```
$ kubectl -n istio-system get svc grafana
```

If the services are running, you can access the Grafana UI dashboard

```
$ istioctl dashboard grafana
```

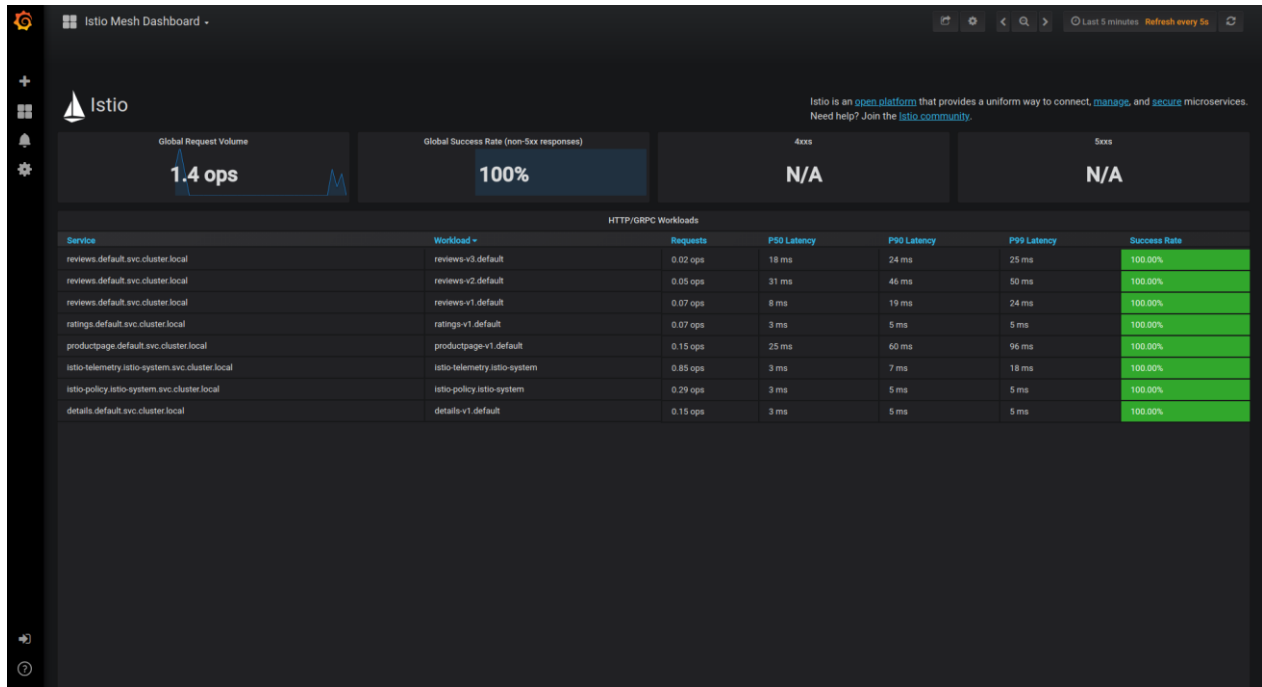
To sample Bookinfo, visit [http://\\$GATEWAY_URL/productpage](http://$GATEWAY_URL/productpage) in a web browser or execute the following command:

To see network traffic, you need to send queries to your service. The number of requests depends on the sampling frequency of Istio. You set this sampling value during the Istio installation. The default sampling rate is 1%. You must send at least 100 requests before the first trace is visible. To send 100 requests to the productpage service, use the following command:

```
$ for i in $(seq 1 100); do curl -s -o /dev/null "http://$GATEWAY_URL/productpage";  
Done
```

Refresh the page a few times (or send a command several times) to generate little traffic.

Take another look at the Istio dashboard. It should reflect the generated traffic. It will look similar to:



This gives you a global view of the grid along with the services and workloads in the grid. You can learn more about services and workloads by navigating to their specific dashboards. Look for dashboards showing individual services and their workloads.

Regarding Grafana desktops:

The Istio dashboard consists of three main sections:

- Series grid summary view. This section provides a global grid summary view and shows the HTTP/gRPC and TCP loads in the grid.
- View of individual services. This section provides request and response metrics for each individual service in the grid (HTTP/gRPC and TCP). It also provides metrics about customer and service workloads for that service.
- Individual workload view: This section provides request and response metrics for each individual workload on the network (HTTP/gRPC and TCP).

3. Visualizing Mesh

In this point, you will visualize the structure of your microservices network using the Kiali tool. Install Kiali with the command:

```
$ kubectl apply -f https://raw.githubusercontent.com/istio/istio/release-1.13/samples/addons/kiali.yaml
```

Kiali is an add-on and uses a web graphical user interface to display graphs of our grid of microservices and Istio configuration objects.

Make sure that you have the Bookinfo sample running.

To verify that Kiali is running, call

```
$ kubectl -n istio-system get svc kiali
```

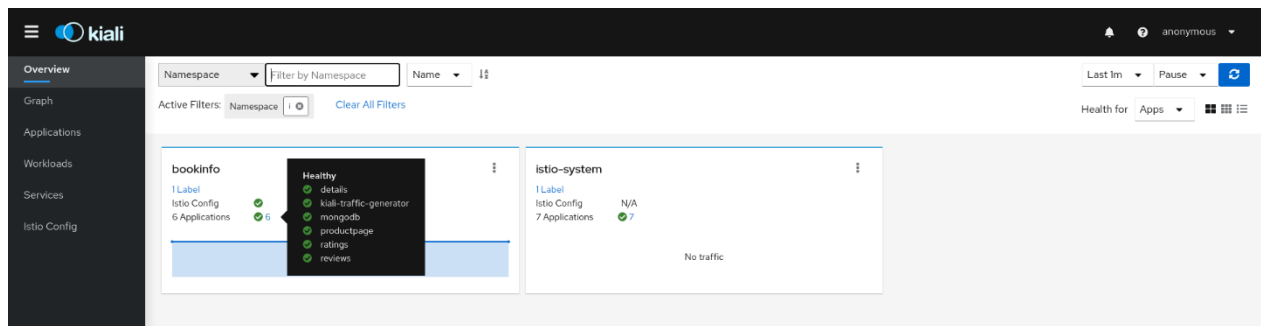
Make sure you know Bookinfo GATEWAY_URL. Trigger traffic by accessing the page in a browser or by calling

```
$ curl http://$GATEWAY_URL/productpage
```

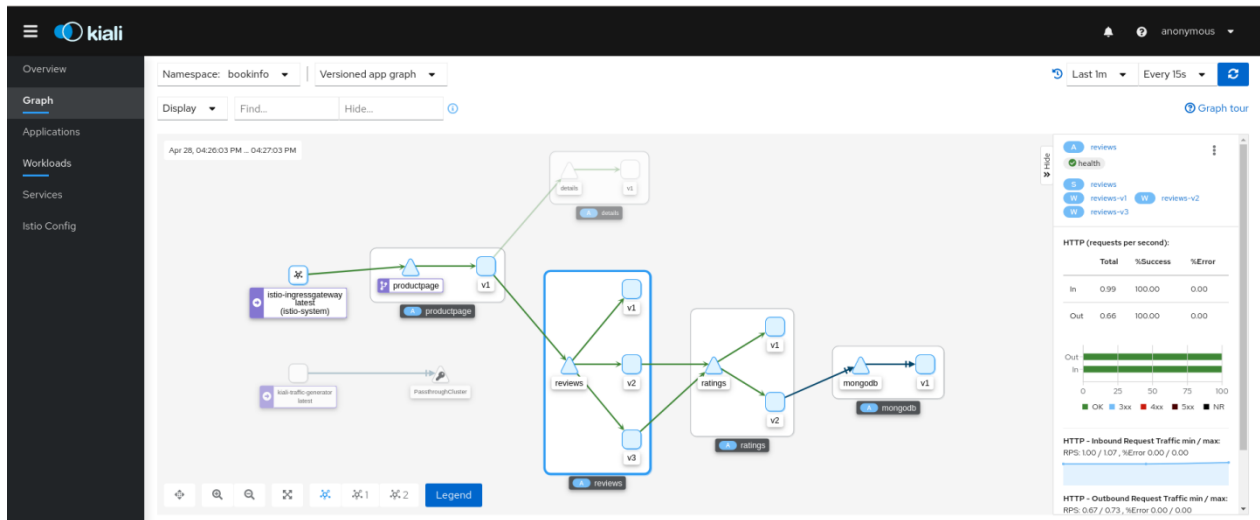
Then run Kiali UI with the help of

```
$ istioctl dashboard kiali
```

View an overview of your microservices grid on the page, which appears immediately after logging in. The Overview page displays all namespaces that have sites in your grid. The following screenshot shows a similar page:



To view the namespace chart, select Graph in the menu on the Bookinfo tab. The menu is located in the upper right corner of the tab and looks like 3 vertical dots. Click on it to see available aboutpcje. The page looks similar to the one shown below:



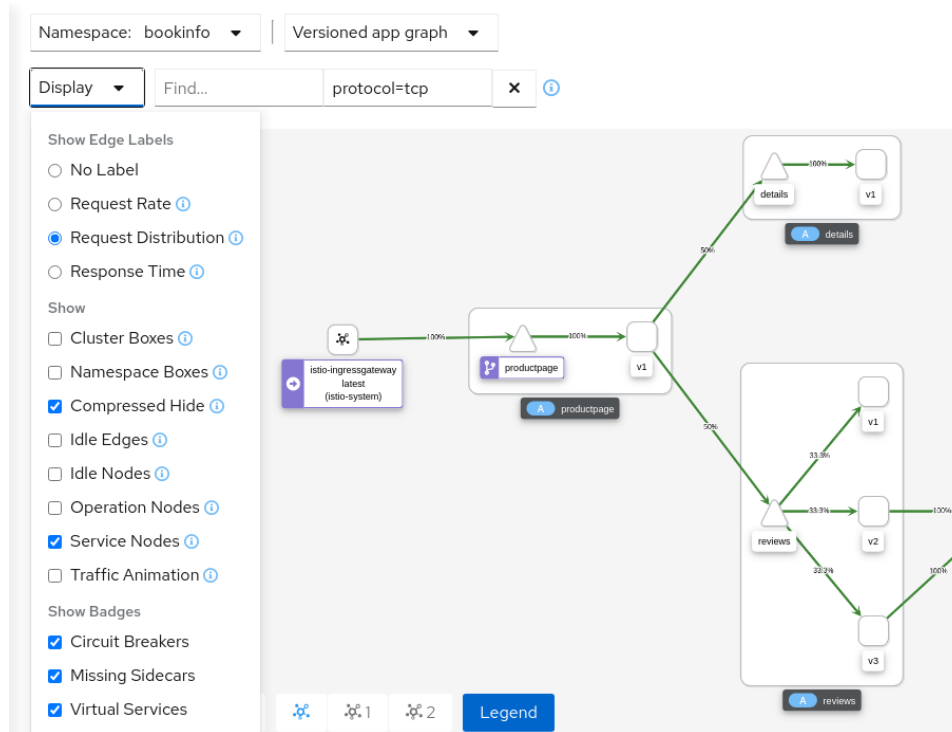
The graph shows the traffic flowing through the network over a period of time. It is generated using Istio telemetry. To view a summary of metrics, select any node or edge in the chart to view the details of the metric in the summary details pane on the right.

To view the service grid using different chart types, select a chart type from the Graph Type drop-down menu. There are several chart types to choose from: App, Versioned App, Workload, Service.

Add views of each chart type to your report

Kiali also offers interference in the Istio configuration, e.g. what we did in the previous part of the laboratory, i.e. traffic redirection.

View the Versioned app graph for bookinfo. Make sure that after expanding Display you have Request Distribution and Service Nodes selected.



We will change the traffic of reviews on the bookinfo chart by clicking the ratings node (triangle). Note that the traffic related to ratings is evenly distributed between two services v1 and v2 (50% of requests are directed to each service).

Click the reviews link in the side panel to go to the site details view. You can also do this by double-clicking the site node and selecting Details from the context menu.

From the Actions drop-down menu, first select Delete Traffic Shifting if available, and then select Traffic Shifting to access the Traffic Change Wizard.

Drag the sliders to specify the percentage of traffic directed to each site. For reviews-v1 set traffic to 10%; for reviews-v2 set traffic to 70% and for reviews-v3 set traffic to 20%

Click Create to apply the new traffic settings. Click Graph in the left navigation bar to return to the bookinfo information chart. Note that the Rating Services node is now marked with a virtual service icon.

Send queries to the bookinfo app. For example, to send one query per second, you can execute this command, if the command does not work install the watch function:

```
$ watch -n 1 curl -o /dev/null -s -w %{http_code} $GATEWAY_URL/productpage
```

After a few minutes, you'll notice that the percentage of your traffic will reflect your new traffic route, thus confirming the fact that your new rule is directing as agreed. **Create your own percentage breakdown and add to the report.**