A Quick Introduction to Istio, Prometheus, and Flagger

This lesson gives an introduction to Istio, Prometheus and Flagger.

We'll cover the following

Istio

Istio is a service mesh that runs on top of Kubernetes. Quickly after the project was created, it became one of the most commonly used in Kubernetes. It allows traffic management that enables us to control the flow of traffic and other advanced networking such as:

- *point to point security*
- *policy enforcement*
- *automated tracing*
- *monitoring*
- *logging*

We could write a full book about **Istio**. Instead, we'll focus on the traffic shifting and metric gathering capabilities of **Istio** and how we can use those to enable **Canary** deployments.

Prometheus

We can configure **Istio** to expose

metrics that can be pulled by specialized tools. **Prometheus** is a

natural choice, given that it is the most commonly used application for *pulling*, *storing*, and querying *metrics*.

Its format for defining metrics can be considered the de-facto standard in Kubernetes. It stores time-series data that can be queried using its query language **PromQL**. We just need to make sure that **Istio** and **Prometheus** are integrated.

Flagger

Flagger is a project sponsored by **Weaveworks**. It can use service mesh solutions like **Istio**, **Linkerd**, **Gloo** (which we used with **Knative**), and quite a few others. Together with a service mesh, we can use **Flagger** to automate deployments and rollback using a few different strategies. Even though the focus right now is canary deployments, you should be able to adapt the examples that follow into other strategies as well. To make things easier, **Flagger** even offers a **Grafana** dashboard to monitor the deployment progress.

🔍 Please note that we could have used **Gloo** instead of **Istio**, just as we did in the Using Jenkins X To Define And Run Serverless Deployments chapter. But, I thought that this would be an excellent opportunity to introduce you to

Istio. Also, you should be aware that none of the tools are the focus of the book and that the main goal is to show you one possible implementation of canary deployments. Once you understand the logic behind it, you should be able to switch to whichever toolset you prefer.

> 🔍 This book is dedicated to continuous delivery with Jenkins X. All the other tools we use are chosen mostly to demonstrate integrations with Jenkins X. We are not providing an in-depth analysis of those tools beyond their usefulness to continuous delivery.

In the next lesson, we will be taking a look at how to install **Istio**, **Prometheus**, and **Flagger**.