SERVICE MESH

JENSON MATHEW

245048

What is a Service mesh?

- A service mesh is a mechanism for managing communications between the various individual services that make up modern applications in a microservice-based system.

- When a service mesh is applied, all inter-service communication is routed through proxies, which can be used to implement networking features such as encryption and load balancing.

Why Do You Need a Service Mesh?

• Once upon a time, programs were developed and deployed as a single application called monolithic architecture, this traditional approach worked fine for simple applications, but it becomes a burden as applications grow complex and the codebase swells.

• That's why modern organizations typically move from monolithic architecture to microservices architecture.But as the number of services grow, we need to ensure connection between services is fast, smooth, and resilient.

What is Istio?

- Istio is an open source service mesh that layers transparently onto existing distributed applications.

- Istio's powerful features provide a uniform and more efficient way to secure, connect, and monitor services.

- Istio is the path to load balancing, service-to-service authentication, and monitoring – with few or no service code changes.

How it Works

- Istio has two components: the data plane and the control plane.

- The data plane is the communication between services. Without a service mesh, the network doesn't understand the traffic being sent over, and can't make any decisions based on what type of traffic it is, or who it is from or to.

- The control plane takes your desired configuration, and its view of the services, and dynamically programs the proxy servers, updating them as the rules or the environment changes.

Installation

- Install istio from official site: https://github.com/istio/istio/releases/tag/1.17.2

- Install kubectl: https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/Install

- eksctl: https://github.com/weaveworks/eksctl/blob/main/README.md#installation

- Install aws cli: https://aws.amazon.com/cli/

Commands

minikube start

aws configure

eksctl create cluster –name tp-cluster-1 –nodes 2 –node-type m5.large --managed –region us-east-1

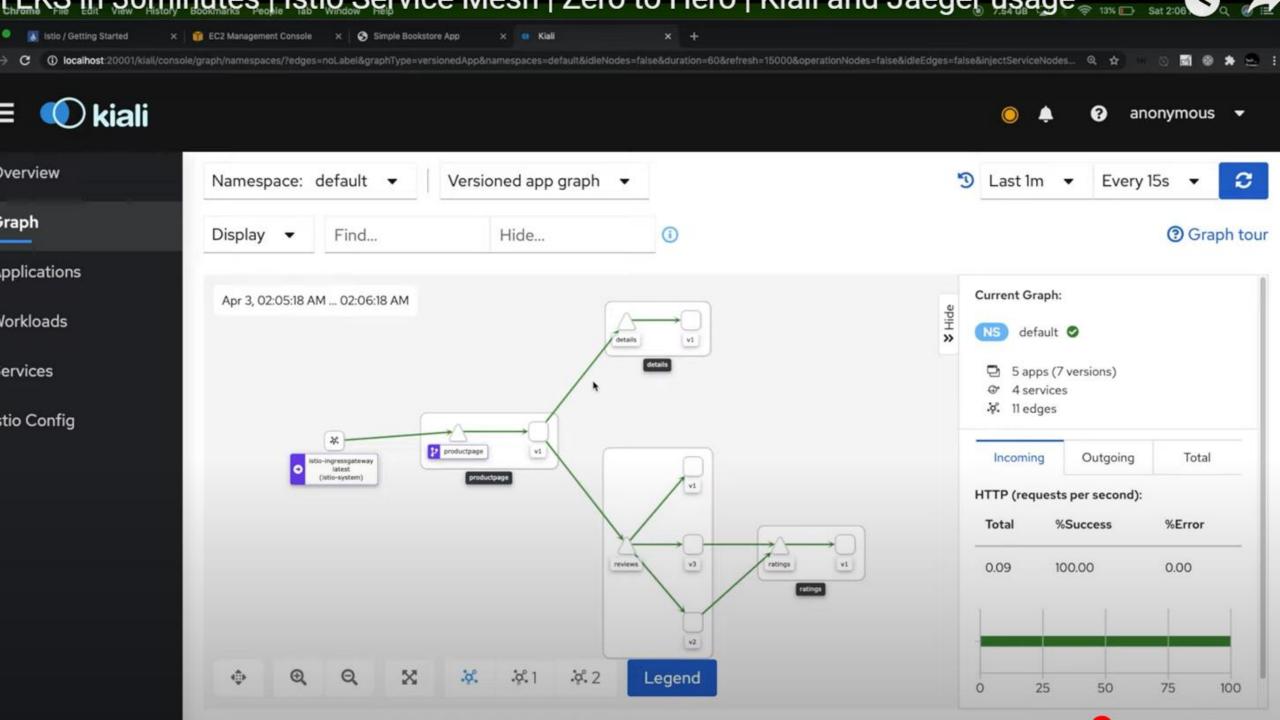kubectl get svc istio-ingressgateway –n istio-system (create load balancer in aws)
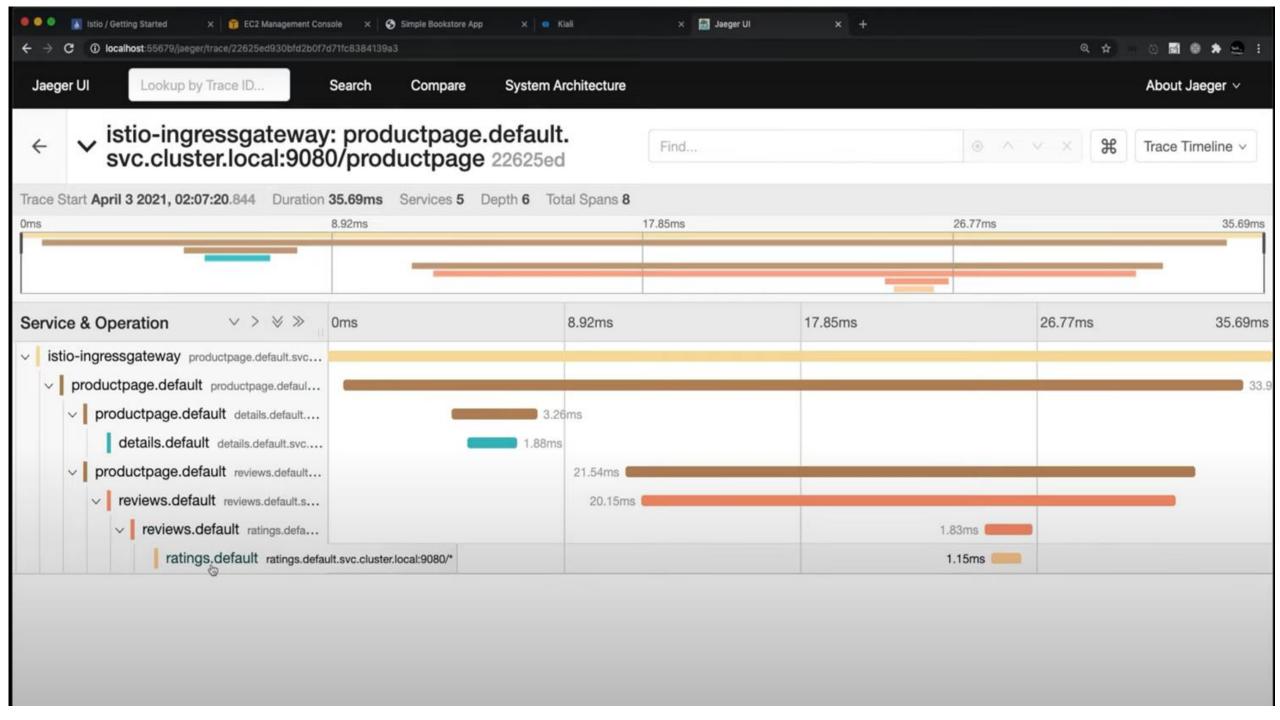

Istioctl dashboard kiali

http://localhost:20001/kiali

Istioctl dashboard jaeger

http://localhost:16686

Chrome File Edit View History Bookmarks People Tab Window Help   7.64 GB ⚡ 13% ▮ Sat 2:06

| Istio / Getting Started | × | EC2 Management Console | × | Simple Bookstore App | × | Kiali | × | + |

localhost:20001/kiali/console/graph/namespaces/?edges=noLabel&graphType=versionedApp&namespaces=default&idleNodes=false&duration=60&refresh=15000&operationNodes=false&idleEdges=false&injectServiceNodes...

kiali

🟡 🔔 ❓ anonymous ▼

Overview

Graph

Applications

Workloads

Services

Istio Config

Namespace: default ▼ | Versioned app graph ▼

🕐 Last 1m ▼   Every 15s ▼   🔄

Display ▼   Find...   Hide...   ⓘ

❓ Graph tour

Apr 3, 02:05:18 AM ... 02:06:18 AM

details
v1

details

productpage
v1

productpage

istio-ingressgateway
latest
(istio-system)

v1

reviews

v3

ratings
v1

ratings

v2

⊹ 🔍 🔍 ⤢ ⚙ ⚙1 ⚙2 **Legend**

Current Graph:

NS default ✅

🗔 5 apps (7 versions)
⚙ 4 services
⚙ 11 edges

| Incoming | Outgoing | Total |

HTTP (requests per second):

Total	%Success	%Error
0.09	100.00	0.00

0   25   50   75   100

References

https://youtu.be/wCJEfqKpUf4

https://istio.io/latest/docs/setup/getting-started/

Thank you