Istio Workshop

DigIO

Istio Workshop

Deliver secure, observable and resilient microservices at scale using the Istio service mesh.

This workshop will cover:

Istio deployment and sample microservice solution overview.

Mesh architecture walkthrough, typical deployment topology and component responsibilities.

Custom resource definitions and how to combine them.

Agent-less observability with metrics, traces and logs.

Endpoint authentication and fine-grained access control.

Advanced routing and service resiliency.

Participants will leave with answers to the following questions:

What is it, and when should I use it?
Gain an understanding of how Istio works, and the problems it can be applied to.

How do I use it?
Implement common use cases in hands on labs, and become familiar with the tools used to implement them.

What do I do next?
Leave with a working environment and sample applications to support further self-paced learning.

Audience:

Developers or platform engineers who wish to understand the benefits that Istio provides on top of standard Kubernetes, and typical deployment patterns for managing microservice applications.

No prior understanding of Istio is required, however a working knowledge of Kubernetes in a local environment is required.

Duration:

2 days

Number of attendees:

8-10 per session

DigIO

Istio Workshop

Contents

DigIO

Istio Workshop

Overview

This training aims to provide a crash course in Kubernetes and the fundamental primitives available to us, enabling the capabilities of Service Mesh technologies like Istio.

We go through the most pertinent aspects of Kubernetes which will give us sufficient context to understand how Istio works and how we can enable application deployment and lifecycle management.

The key benefits of Istio are demonstrated through sophisticated traffic steering and observability capabilities, with enhanced security through authentication (JWT, mTLS) and authorization (RBAC).

Given the time available to cover such a large problem space, we'll skip certain parts of the training which can be covered in a self-paced fashion. Made possible via the distribution of the training material in a self-contained git repository and deployed into a running Kubernetes cluster locally.

Prerequisites

Admin control of machine to install dependencies.

Install Docker-for-desktop available here.

Run make bootstrap this will install dependencies and validate the level of access on your given machine (admin permissions required).

Run make build.

Istio Workshop

Workshop outline

Istio - Service Meshing Deployments   30-45 minutes

We cover the key aspects of Kubernetes which enable us to understand what information Istio is using as part of a services deployment to facilitate the integration of a service into the "Service Mesh".

We cover how Kubernetes's service discovery capability is used by Istio to enable service-to-service networking within Istio and how the [Ingress-->Service-->Endpoints] constructs translated into the Istio world.

Workshops   30 minutes

Deploy a "simple" version of the full-stack application. Which utilises Nginx-ingress, acting as a more "primitive" ingress mechanism and the associated observability we have available.

Outcomes

Understand what information is important to Istio and how this is used to piece together Service Mesh connectivity. A working understanding of the difference between an Ingress controller and a Service Mesh, especially in the context of observability capability.**Istio - Architecture   30 minutes**

Istio control-plane components and the role played in managing the overall Service Mesh.

Outcomes

Understand the different control-plane components of Istio, and how they work together to manage the underlying Envoy proxy. Where Envoy makes up the core data-plane component which enables connectivity.

DigIO

Istio Workshop

Workshop outline

Istio - Custom Resource Definitions (Resources) 1-1.5 hours

Explore the available Istio API including: DestinationRule, ServiceEntry, VIrtualService, Gateway. And how these primitives are used to construct a Service Mesh topology.

Workshops 1.5-2.5 hours

Deploy fullstack application into an Istio Service Mesh

Traffic Control - Basic

Traffic Control - Advanced [Headers, Canary]

Service Resiliency [CIrcuit Breaking, Retries]

Rate Limiting

Outcomes

Understand the basic resources we use in Istio, and how these pieces together to form a Service Mesh topology. Demonstrate an understanding of the available policies/configuration that can be used to manage the steering of traffic, and how this can be used to enable canary & Blue/Green releases of applications.

Understand how Istio can be used to improve service "Resiliency" - improving fault tolerance, faulty release candidates, and service availability

Istio - Security 30 minutes

Understanding the end-to-end enabling of mTLS within the Istio Service Mesh. Exploring the involved control plane components, and their different interactions.

Workshops 1-1.5 hours

Enabling Authentication (mTLS).

End-user Authentication (JWT).

Granular access control for microservices utilising Role Based Access Control (RBAC).

Outcomes

Understand the available security features of Istio, and how they can be employed to provide next-generation security capabilities with incredible granularity.

Demonstrate an understanding for why we would want to push this capability into the Service Mesh, and how this simplifies application development and platform operation.

DigIO

Istio Workshop

Workshop outline

Istio - Distributed Tracing and Observability 30 minutes

Overview of Distributed Tracing and Observability, what they are and how Istio is enabling the generation, collection, and presentation of this information.

Workshops 30 minutes

Istio Telemetry (Kiali, Grafana, Jaeger).

Outcomes

Understand what observability is, and how it can be used to facilitate rapid root-cause analysis, triage of production issues, understanding of microservice operation. Capable of demonstrating understanding of how this visibility can facilitate the improvement of microservice development and speed in which services can be deployed and issues resolved.