Kuma

Multi-Cluster & Multi-Cloud Service Mesh

Marco
PALLADINO

CO-FOUNDER & CTO AT KONG

Kuma

Kuma, open source service mesh project started by Kong in September 2019 and donated to CNCF.

PAST 6 MONTHS GROWTH *

130%
Increase in number of deployed Kuma nodes

320%
Increase in number of data planes powered by Kuma

580%
Increase in number of virtual meshes powered by Kuma

* As of June 2020

Matt Klein
@mattklein123

Excited to see an @EnvoyProxy based service mesh now part of the @CloudNativeFdn. 🎉 (Especially one focused on simplicity 😌)

> **Aghi** @sonicaghi · Jun 30
> First step for a true cloud connectivity neutrality! 🚀🙏 Many months of Kong hard work and collaboration to get the @EnvoyProxy universal control plane @KumaMesh to join the @CloudNativeFdn family. Congrats @subnetmarco and team! #servicemesh that works.
> twitter.com/TechCrunch/sta...

Kelsey Hightower ✔
@kelseyhightower

The Kuma project demonstrates the best way to build a control plane that runs on Kubernetes.

Design your control plane as if Kubernetes does not exist. Then add support for Kubernetes as a storage engine and layer CRDs on top of your existing API.

TELUS

"Kuma reduces complexity and accelerates service reliability with an Envoy-based Service Mesh"

**Luca Maraschi,** Chief Architect at Telus Digital

Sabre

"Knowing Kuma can support us moving as quickly as we want on the cloud migration front relieves a lot of pressure. Using Kuma helps my team move fast without compromising on metrics or stability. It removes a lot of our traffic concerns and allows our teams to focus on building business logic instead of managing the network"
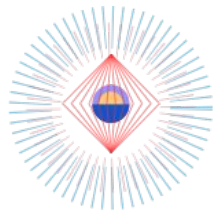
Thomas Ellis, Principal SRE at Sabre

Kuma Overview

Why do you need a Service Mesh?
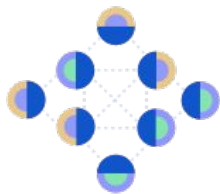
Ensure service connectivity, discovery and traffic reliability

Intelligently route traffic across any platform and any cloud to meet expectations and SLAs

Achieve Zero-Trust Security

Restrict access and encrypt all traffic by default to only complete transactions when identity is verified

Gain Global Traffic Observability

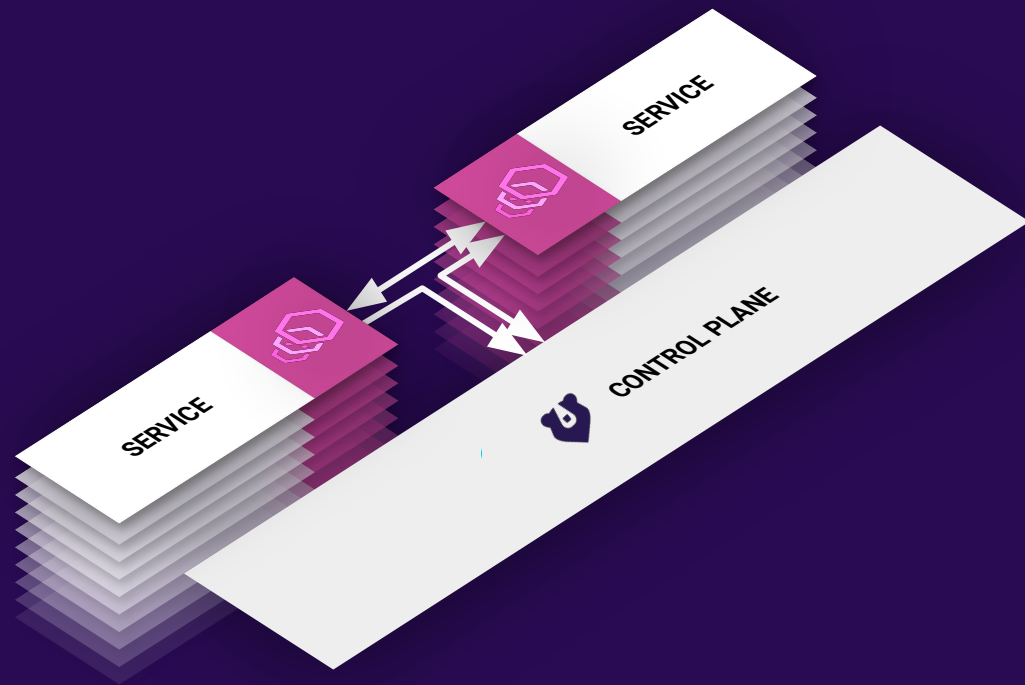Gain a detailed understanding of service behavior to increase application reliability and the efficiency of teams

THE UNIVERSAL SERVICE MESH
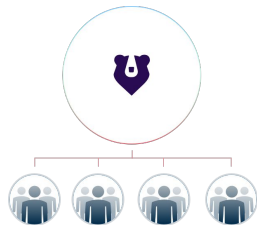
K8s and VMs, single and multi-zone

Built for the enterprise architect

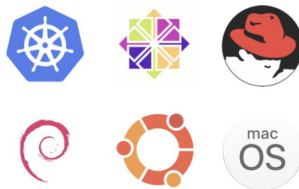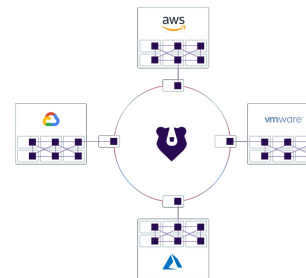Easy to install, use and scale

Vendor Neutral

Why Kuma?

Multi-Mesh And
Easy To Use & Scale

Intelligently route traffic across any platform and any cloud to meet expectations and SLAs

Universal (K8s + VMs),
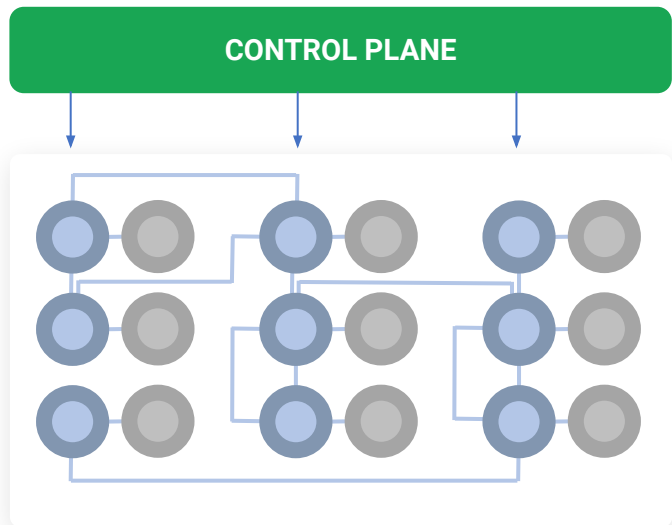### Attribute-Based Policies & More

Restrict access and encrypt all traffic by default to only complete transactions when identity is verified

Built-in Multi Zone
Connectivity

Out of the box connectivity across multi-cluster, multi-cloud and multi-platform deployments across the world.

Ensure Service Connectivity, Discovery and Traffic Reliability
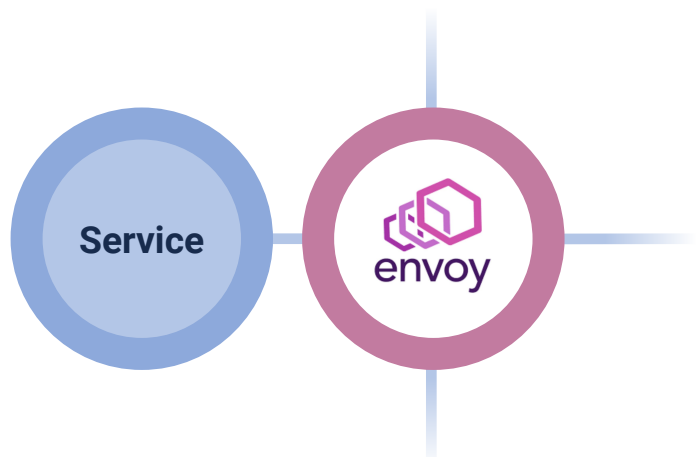
CONTROL PLANE

Unified control plane

Eliminate management complexity
Consistently apply policies to services and reducing the risk of misconfigurations that cause transactions to drop

Increase developer productivity
Instantly add out-of-box policies that eliminate the need to build network functionality into each service

Powered by Envoy Proxy

Envoy-based proxies

Fast by design
Lightweight proxy removes bloat from each service

Reduce performance overhead
Client-side load balancing eliminates hops to a centralized load balancer
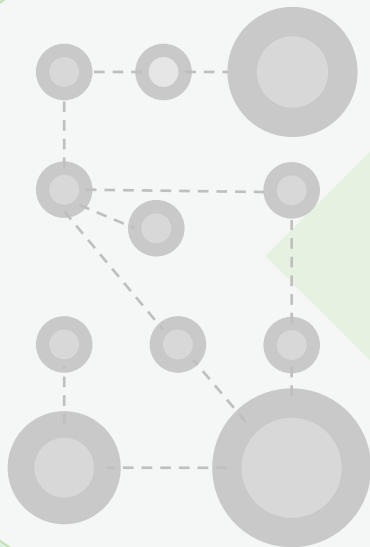
Utilize system resources efficiently
Route traffic to services with spare capacity

Maximize up-time
Monitor the health of services to intelligently retry or route traffic, and seamlessly scales up and down

Achieve Zero-Trust Security

Achieve zero-trust by design
Automatically provide mTLS encryption and identity across every single API, microservice and database

Inject compliance
Fine-grained traffic policies ensure appropriate connectivity and data privacy for every single API, microservice and database

Streamline security responses
Provide the Central IT team with control to rapidly deploy critical security patches across all networks

Gain Global Traffic Observability

Natively integrate with Prometheus for auto-discovery and metrics collection, and then use Grafana dashboards to monitor performance and ensure service mesh health
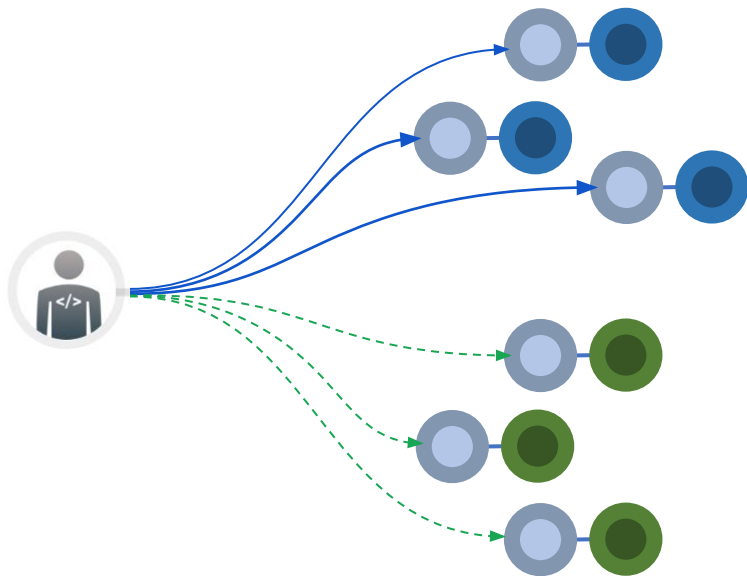
Inject distributed tracing into each service
Monitor and troubleshoot microservices behavior, without introducing any dependencies to the existing code base

Gain a detailed understanding of your service behavior
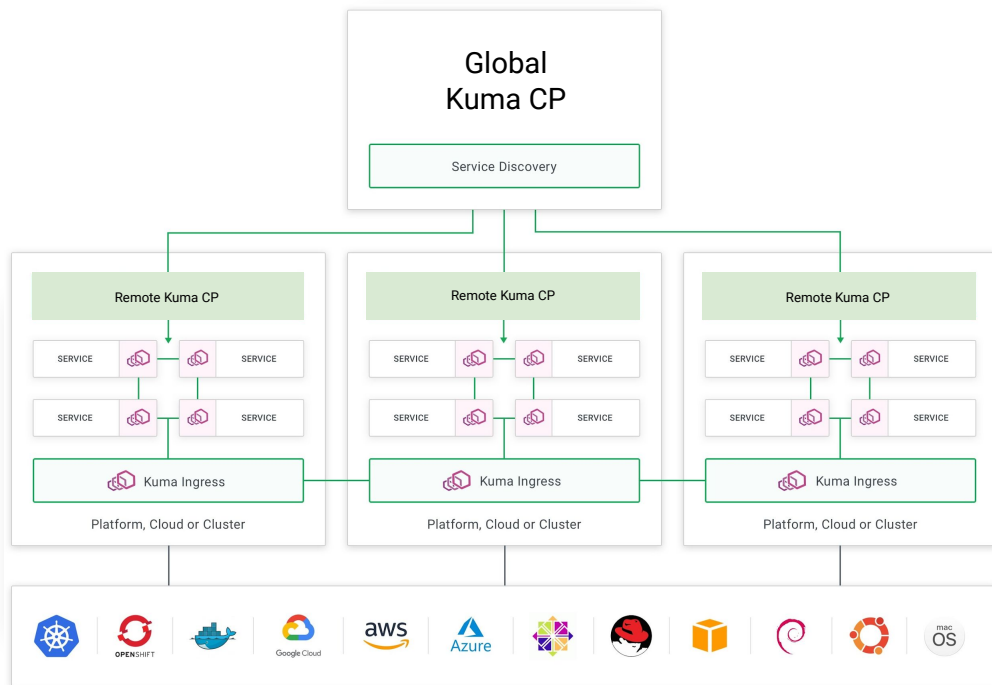Improve the efficiency of your team and accelerate the delivery of performance improvements

Streamline DevOps to Ship Faster

Deploy faster with zero downtime
Improve release management with built in versioning, feature flagging, canary and blue / green deployments to streamline your CI/CD and DevOps workflows

Start, Secure and Scale with Ease

- Turnkey universal service mesh with built-in multi-zone connectivity

- One click deployment, one click attribute-based policies

- Multi-mesh support for scalability across the organization

Kuma

EASY TO USE

Kubernetes Native (CRDs)

Universal CLI

HTTP API

Built-in GUI

Filter by Mesh:

All Meshes

Meshes › all › Dataplanes ›

Overview
Zones
Meshes

DATAPLANES

All Dataplanes
Ingress
Gateway

POLICIES

Circuit Breakers
Fault Injections
Health Checks
Proxy Templates
Traffic Logs
Traffic Permissions
Traffic Routes
Traffic Traces

Status	Name	Mesh	DP Type	Tags
✓ ● Online	billing1	default	Standard	ENV production KUMA.IO/PROTOCOL http2 KUMA.IO/SERVICE billing KUMA.IO/ZONE aws-east VERSION 3.4.7
View ● Online	billing2	default	Standard	ENV production KUMA.IO/PROTOCOL http2 KUMA.IO/SERVICE billing KUMA.IO/ZONE gcp-west VERSION 3.4.5
View ● Online	cassandra1	default	Standard	ENV production KUMA.IO/PROTOCOL tcp KUMA.IO/SERVICE cassandra KUMA.IO/ZONE aws-east VERSION 4.0
View ● Online	cassandra2	default	Standard	ENV production KUMA.IO/PROTOCOL tcp KUMA.IO/SERVICE cassandra KUMA.IO/ZONE gcp-west VERSION 4.0
View ● Offline	users-api-1	default	Standard	ENV production KUMA.IO/PROTOCOL http KUMA.IO/SERVICE users KUMA.IO/ZONE aws-east VERSION 1.5.1
View ● Online	users-api-2	default	Standard	ENV production KUMA.IO/PROTOCOL http KUMA.IO/SERVICE users KUMA.IO/ZONE aws-east VERSION 1.5.1

Mesh: billing-1

Overview   Certificate Insights   YAML

Entity Overview for billing-1

Universal   Kubernetes

```
type: Dataplane
mesh: default
name: billing-1
networking:
  address: 192.168.0.2
```

POLICY BASED

Security

Traffic Control

Observability

Advanced

Policies

Bundled policies for your service traffic and network configuration.

Security

Identity, Encryption and Compliance

Mesh / Multi-Mesh	Mutual TLS (mTLS)	Traffic Permissions

Traffic Control

Routing, Ingress, Failover

Traffic Route	Health Check	Circuit Breaker
Fault Injection	Kong Gateway	

Observability

Metrics, Logs and Traces

Traffic Metrics	Traffic Trace	Traffic Log

Advanced

Envoy configuration and Miscellaneous

Run Anywhere

Service Mesh

Container-based
Microservices

Service Mesh

VM-based
Monolith

VM-based
Monolith
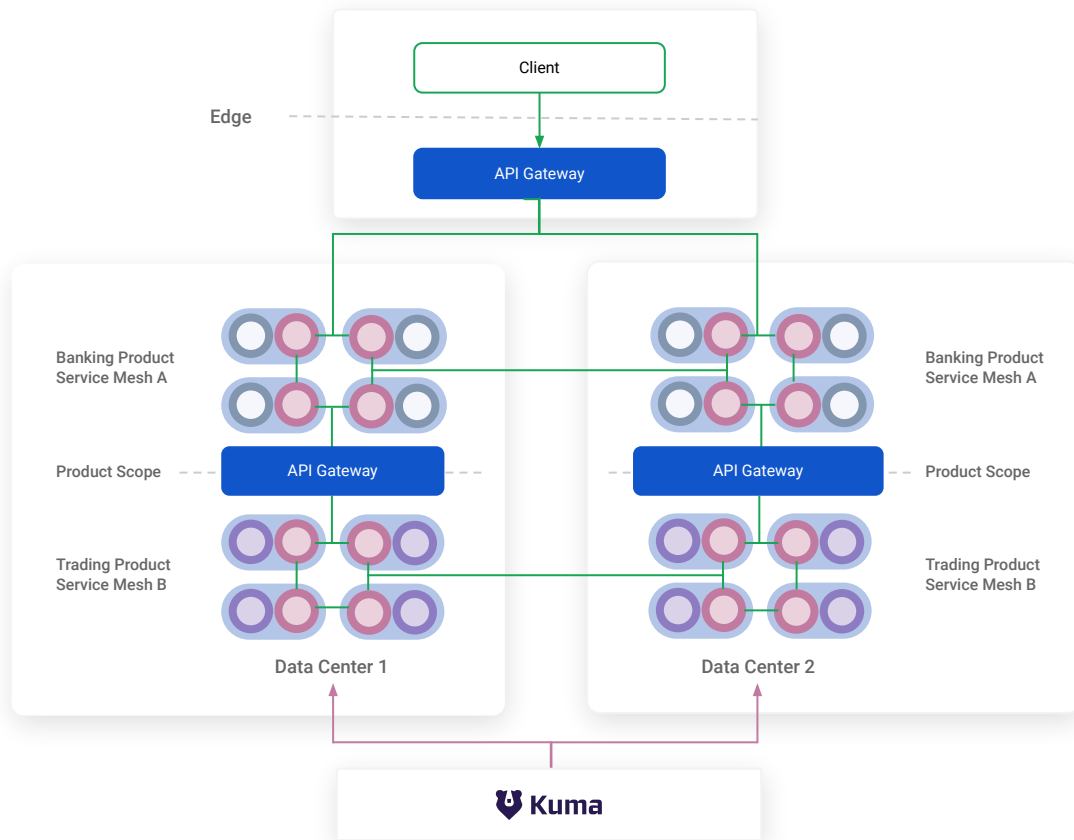
Container-based
Microservices
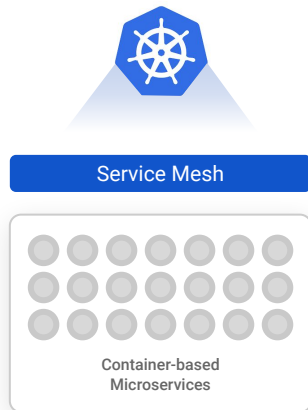
Manage service meshes natively in
Kubernetes using CRDs
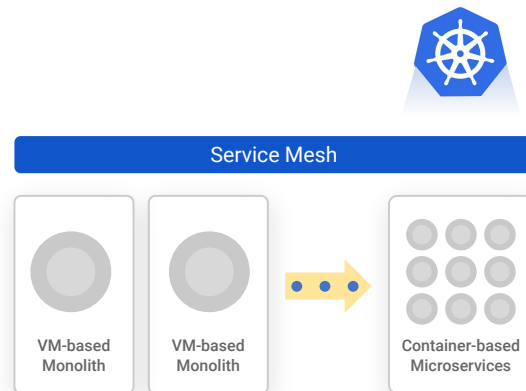
OR start with a service mesh in VM environments
and migrate to Kubernetes at your own pace

Deploy the service mesh across any environment, including multi-cluster, multi-cloud
and multi-platform

Client

Edge

API Gateway

Banking Product
Service Mesh A

Product Scope

API Gateway

Trading Product
Service Mesh B

Data Center 1

Banking Product
Service Mesh A

Product Scope

API Gateway

Trading Product
Service Mesh B

Data Center 2

Kuma

DEMO

KUMA.IO/INSTALL