Red Hat
OpenShift

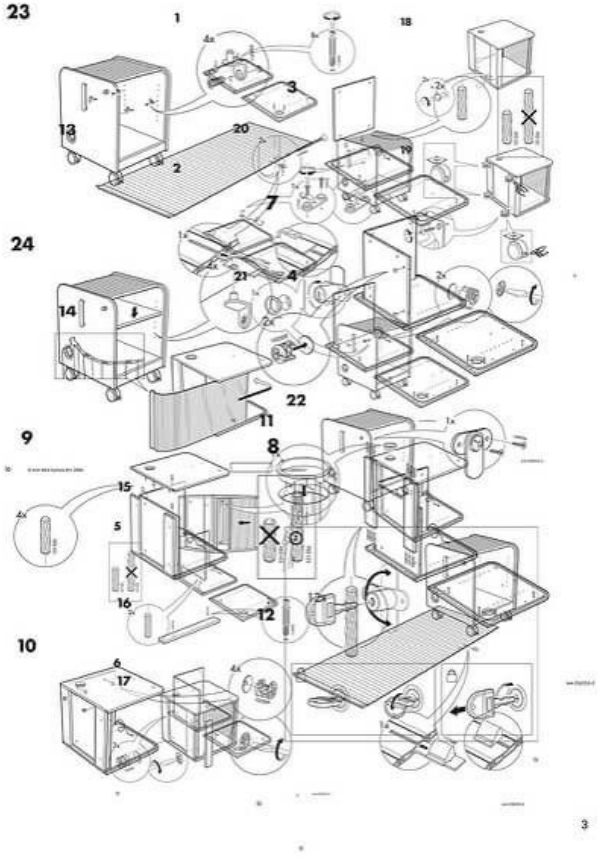
OpenShift Service Mesh

It was supposed to be simple, but then it was *not?*

Building Furniture

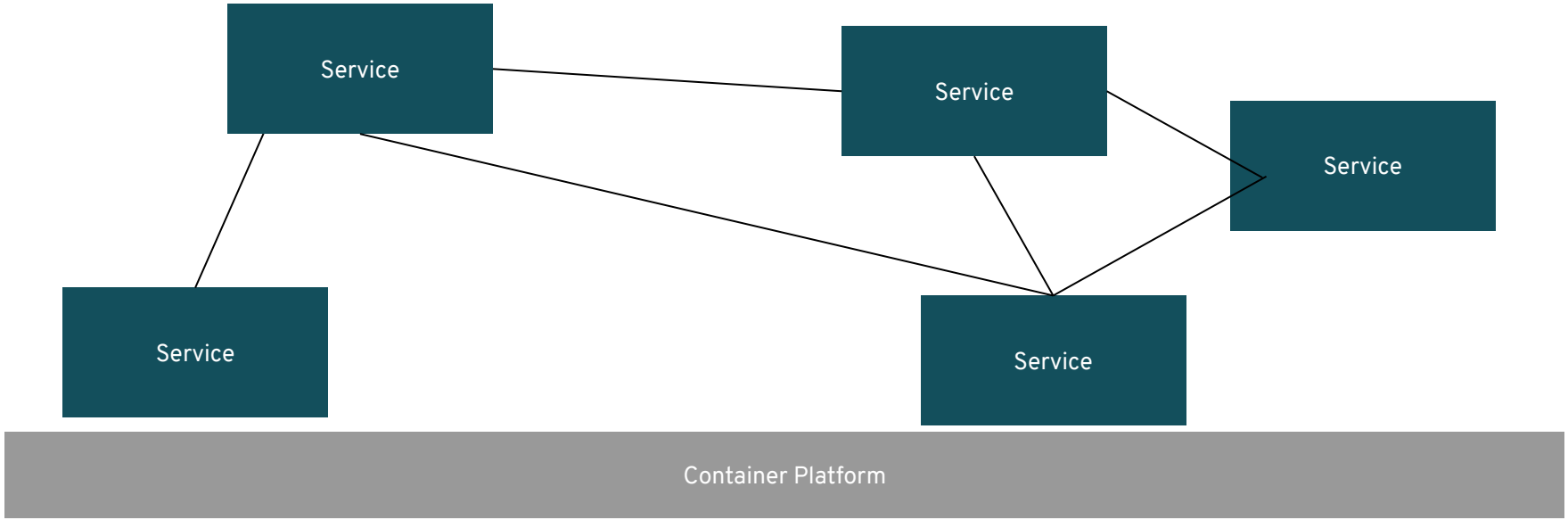
Red Hat

...Remember that one time you thought it would be simple?

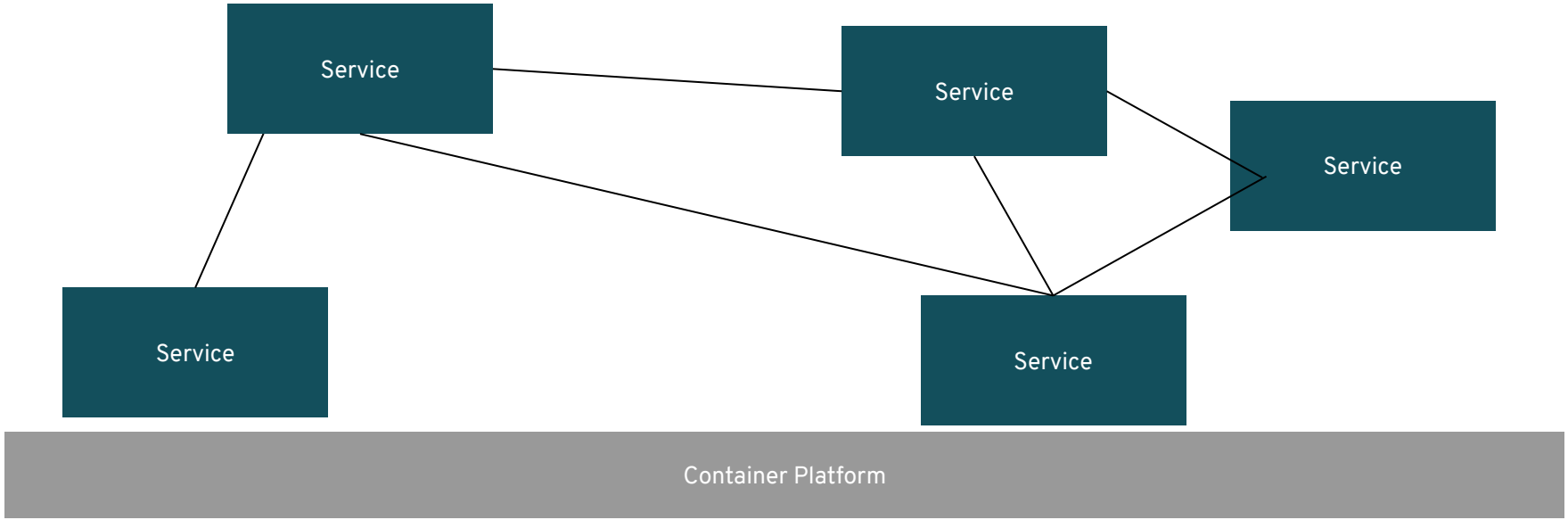


Red Hat

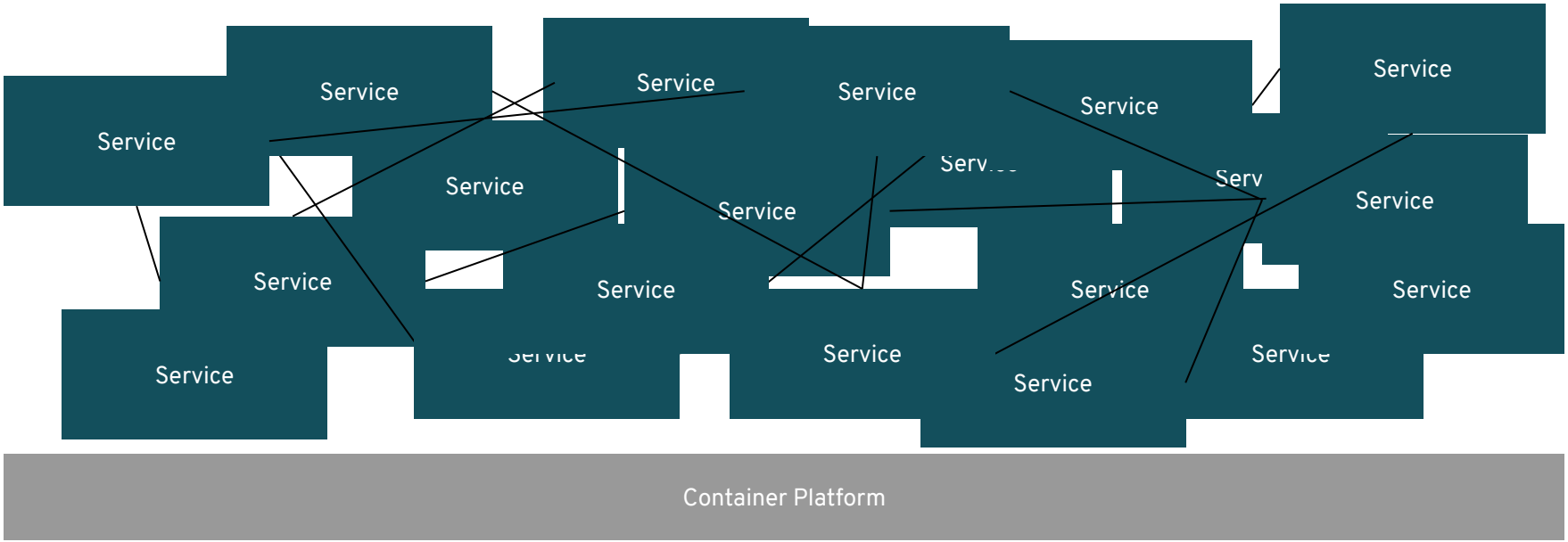
Microservices!



Microservices! Simple!



Do microservices they said! It'll be easy they said!



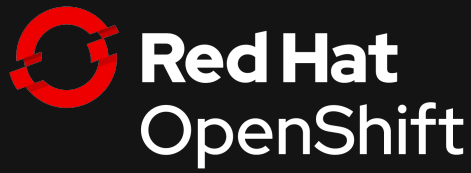
The Microservices Challenge

- Architecture Complexity
- Distributed Computing Problems Multiply
- Polyglot Architecture
- Diagnostic Mess

Addressing the Microservices Challenge

- Defining the problem: distributed computing challenges
- Service Mesh Architecture
- The OpenShift Service Mesh
- Feature deep dive
- Extending Security to the API layer

Distributed Computing Challenges



DISTRIBUTED COMPUTING CHALLENGES

Fallacies of Distributed Computing

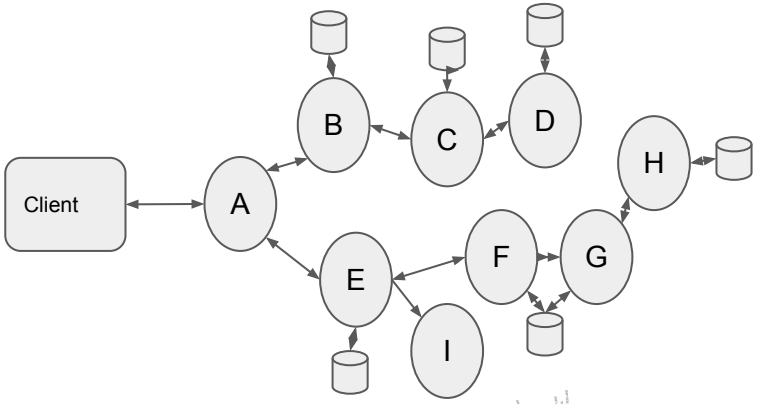
- The network is reliable.
- Latency is zero.
- Bandwidth is infinite.
- The network is secure.
- Topology doesn't change.
- There is one administrator.
- Transport cost is zero.
- The network is homogeneous.

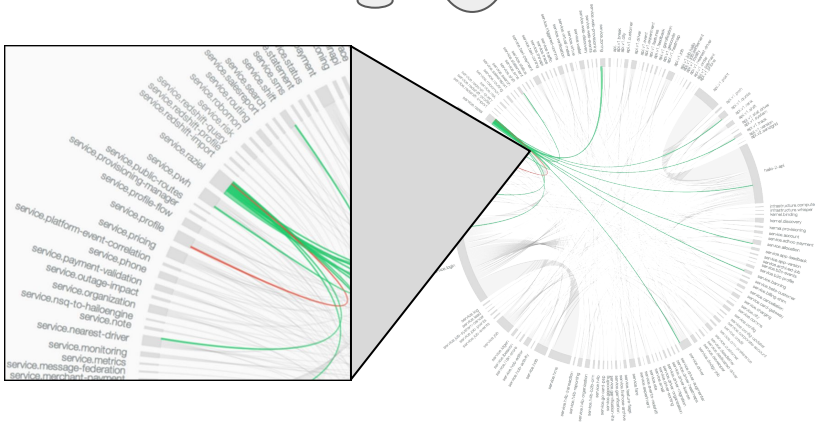
wikipedia.org/wiki/Fallacies_of_distributed_computing

MICROSERVICES ARE HARD

Because applications must deal with...

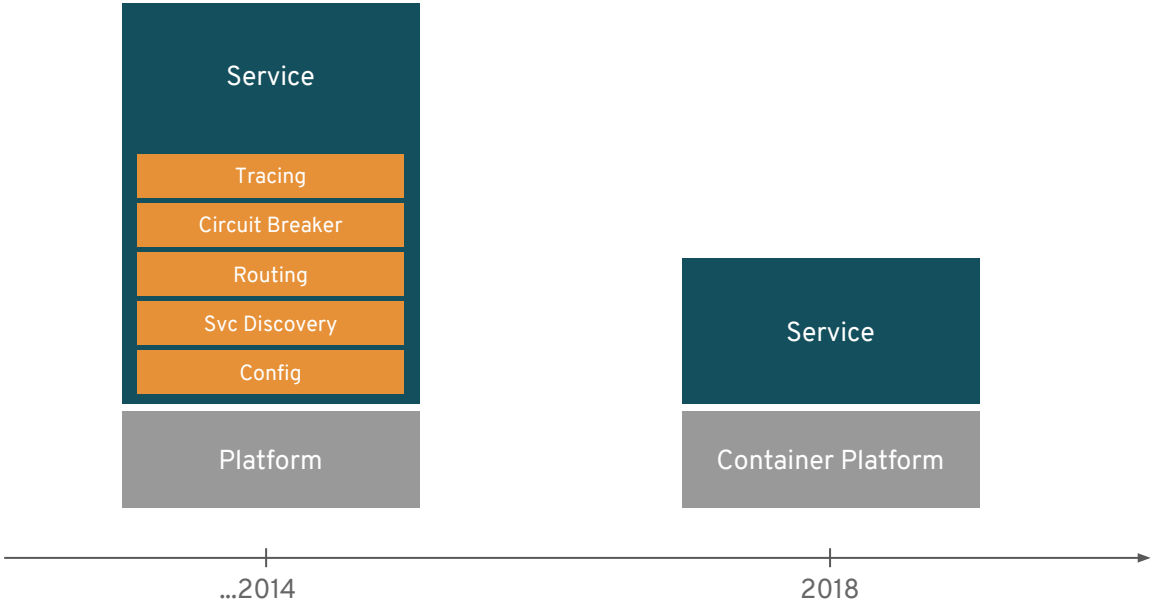
- Unpredictable failures
- End-to-end application correctness
- System degradation
- Topology changes
- Elastic/ephemeral/transient resources
- Distributed logs
- The fallacies of distributed computing





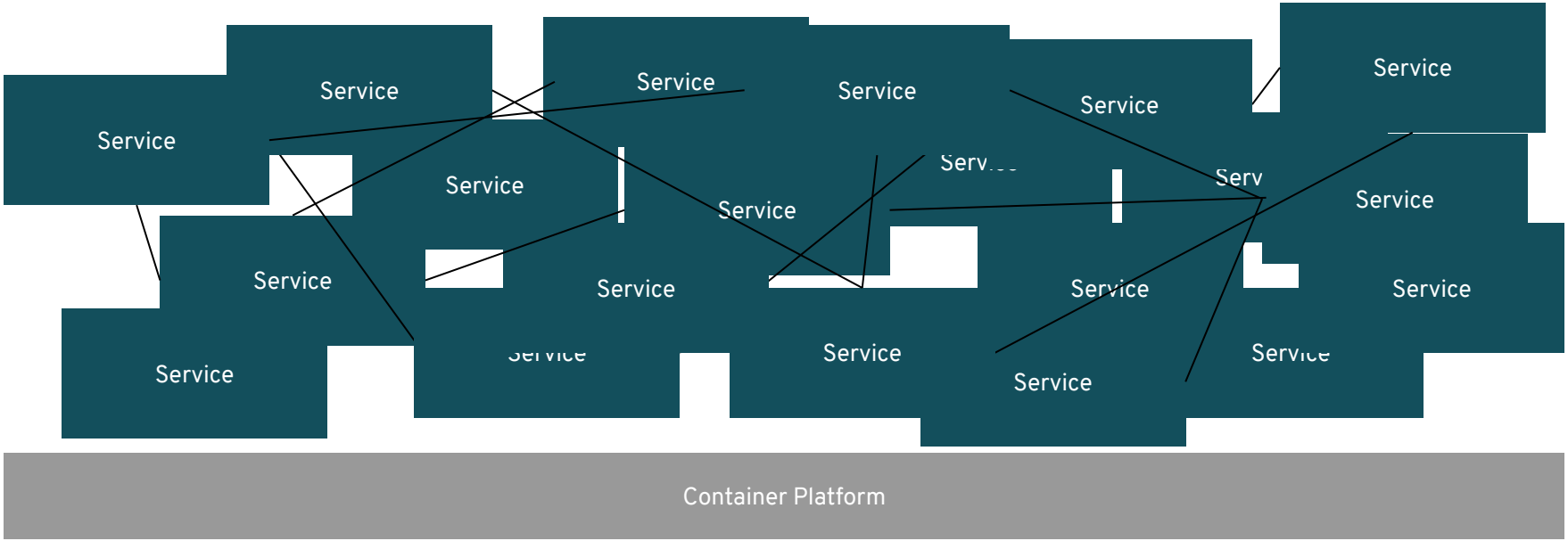
Red Hat

Microservices Evolution



Red Hat

SUPER Simple...

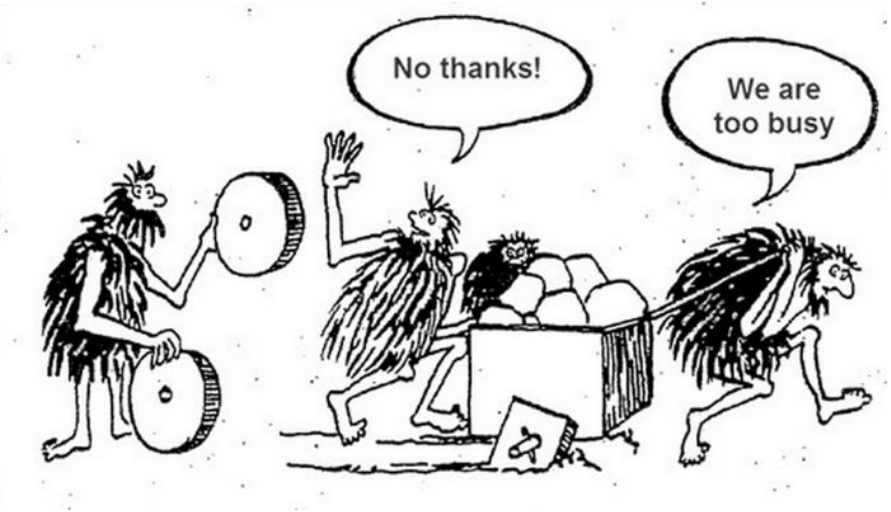


Red Hat

POSSIBLE SOLUTIONS

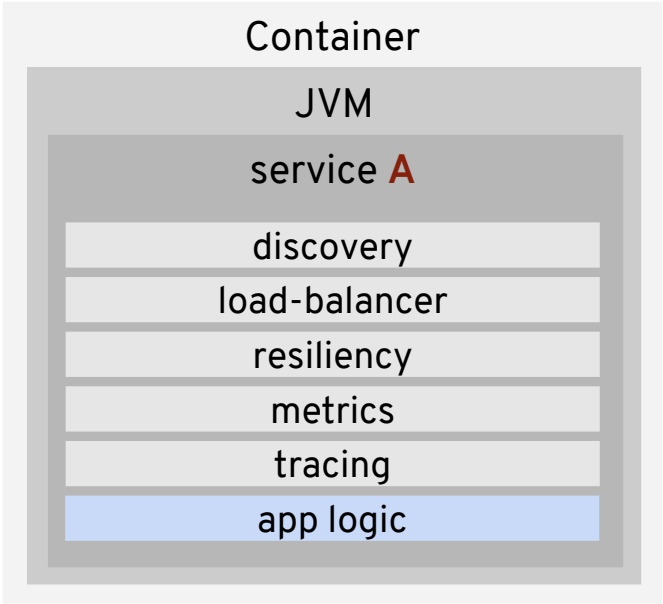
Have your developers do this:

- Circuit Breaking
- Bulkheading
- Timeouts/Retries
- Service Discovery
- Load Balancing
- Traffic Control



Red Hat

NETFLIX OSS



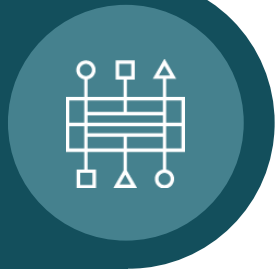
- Hystrix
- Zuul
- Ribbon
- Eureka
- Archaius

Need a library to support each language/framework combination

Red Hat

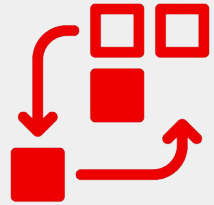
BUT WHAT ABOUT...?

POLYGLOT APPS



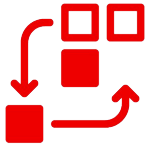
EXISTING APPS

Service Mesh Architecture



Red Hat OpenShift

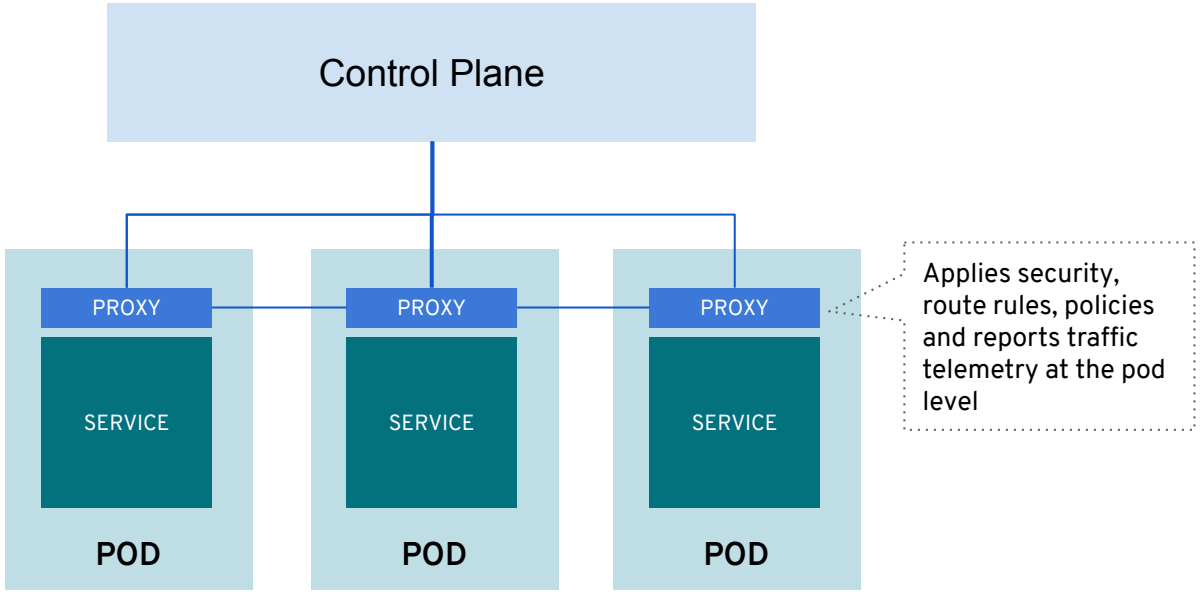
Enter the Service Mesh



Infrastructure layer to help manage service-to-service communication, delivering enhanced security and traffic monitoring for microservices applications.

- Load balancing
- Routing rules
- Service monitoring and logging
- Secure cross-service communications

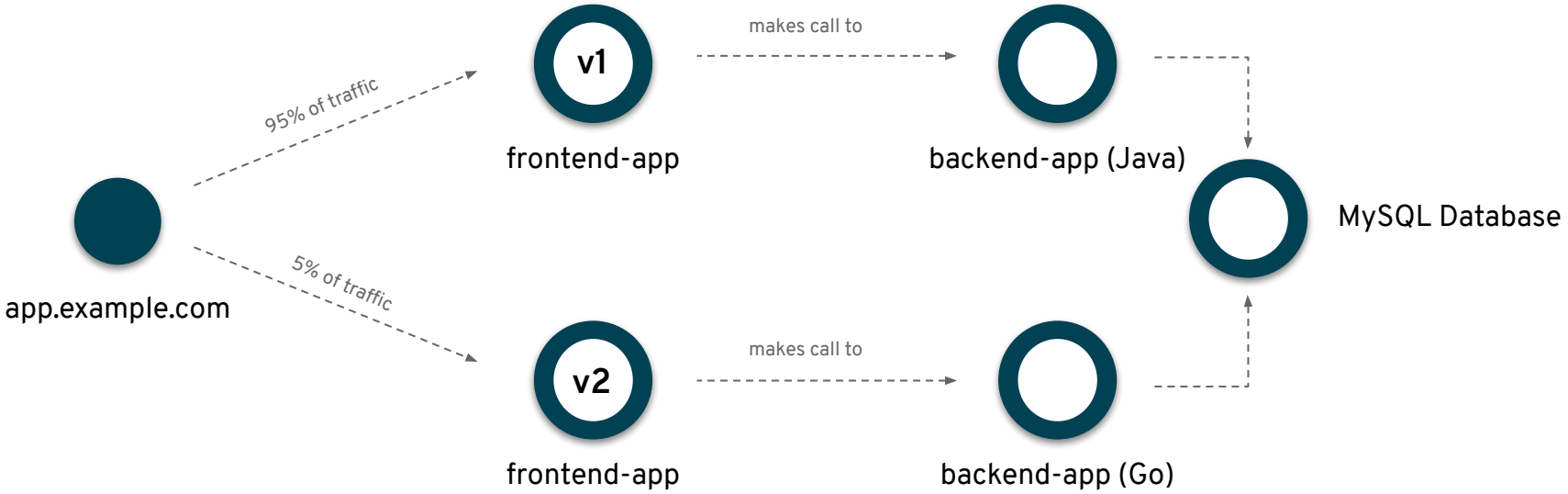
What does the Service Mesh look like?



Red Hat

Simplify with a Service Mesh

Control flow of traffic between application components



Red Hat

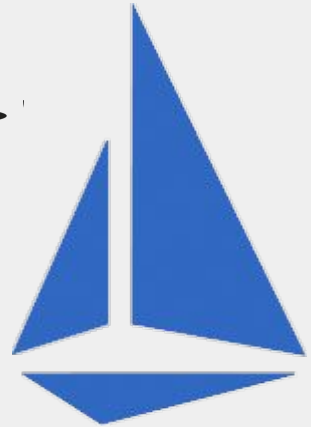
Limitations of a Service Mesh

On its own, the Service Mesh is just the **communication** layer

- Limited measurement functionality
- Limited observation capabilities
- Not a complete set of tools developers need to build and deploy microservices

OpenShift Service Mesh

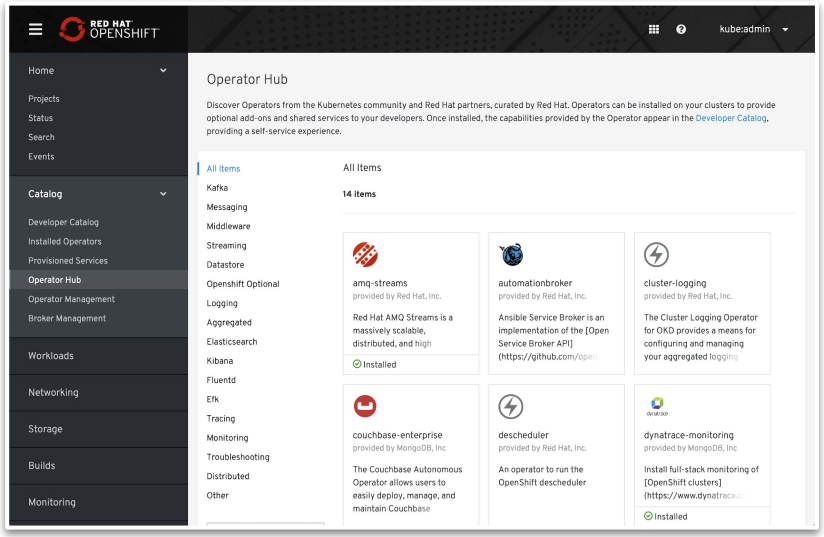
Red Hat OpenShift



Red Hat

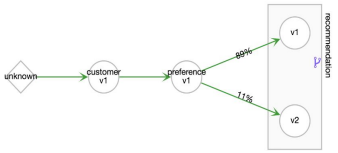
OpenShift Service Mesh

- OpenShift Service Mesh is available and supported at *no additional cost* with OpenShift 4
- OpenShift Service Mesh Operator is found in the OperatorHub menu



Red Hat

OpenShift Service Mesh



Jaeger

Secure

Istio

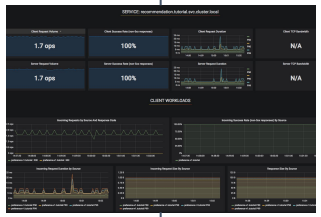
Observe

Connect

Control

Observe

Prometheus



Kiali

Grafana

Red Hat

OpenShift Service Mesh

BENEFITS

- Complete service mesh packaged for ease of use
- Built with key open source projects and integrations
- Extend security through the service mesh into the API layer with with 3scale API management integration

USE CASES

- Adaptive traffic management
- Monitoring and alerting
- Service performance tracing
- Secure communications and API access
- Foundation for Serverless Knative functionality

Enhanced Visualization of Cluster Traffic with Kiali

Visualization of what matters most:

- Application Topology
- Traffic throughput
- Error Rates
- Service Latency
- Service Versioning

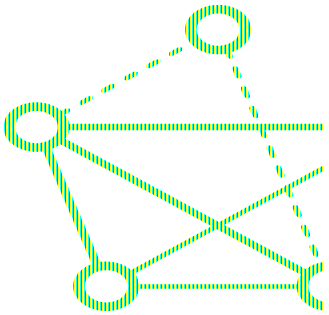
OpenShift Service Mesh Uses

1. Service Mesh visualization
2. Tracing with Jaeger
3. Routing
4. Service versioning
5. Mutual TLS
6. Authentication (e.g. RBAC with JWT)
7. Whitelisting/Blacklisting Auth
8. Retries, timeouts, circuit breakers
9. Rate limiting
10. Fault injection

OpenShift Service Mesh

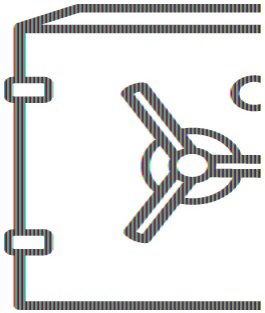
Connect

Control the flow of traffic between services



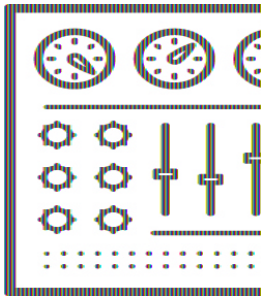
Secure

Application independent security



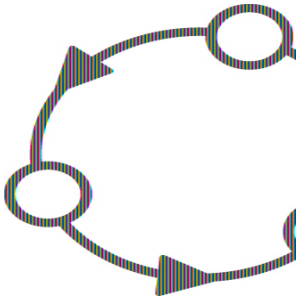
Control

Uniform abstraction for policy control



Observe

Visibility into application deployments



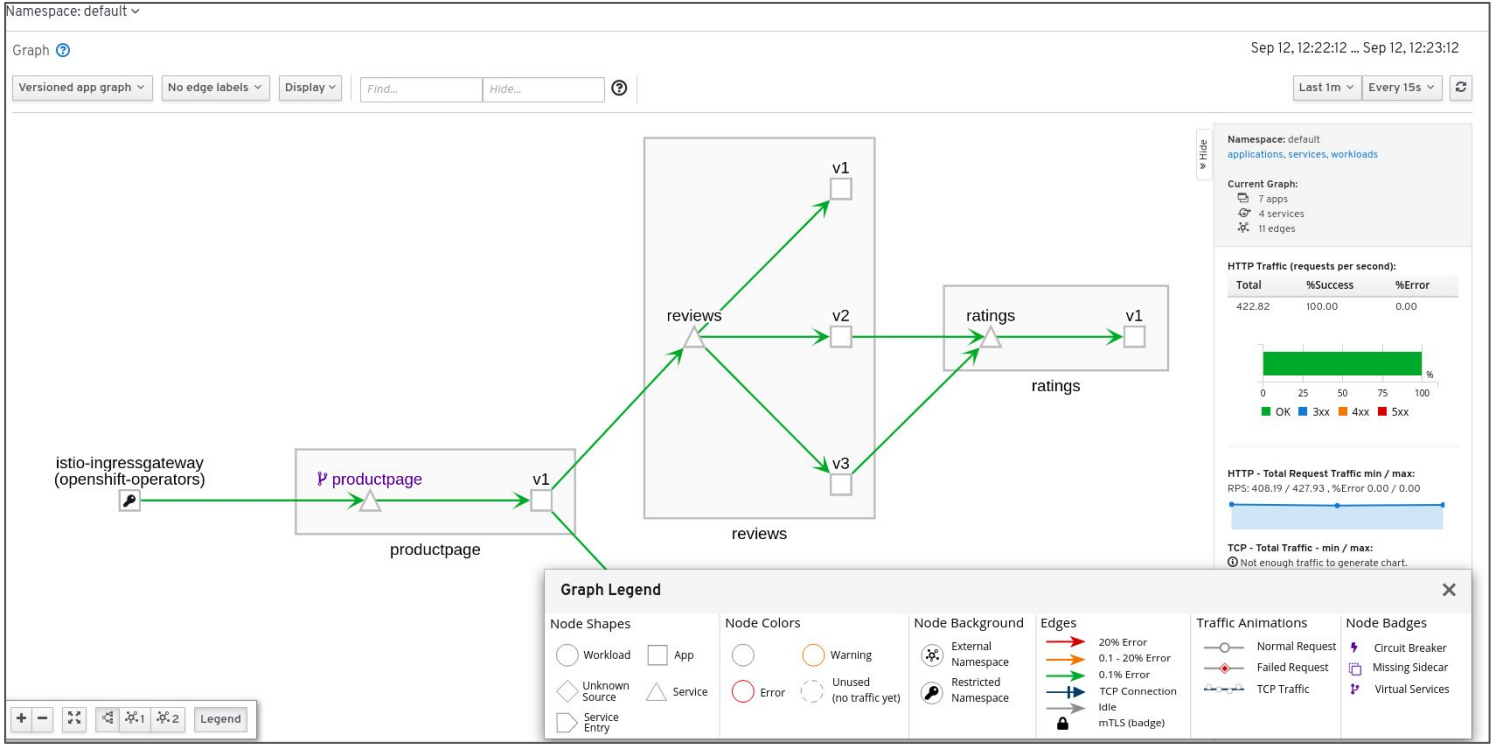
Red Hat

Visualization Deep Dive

Red Hat OpenShift

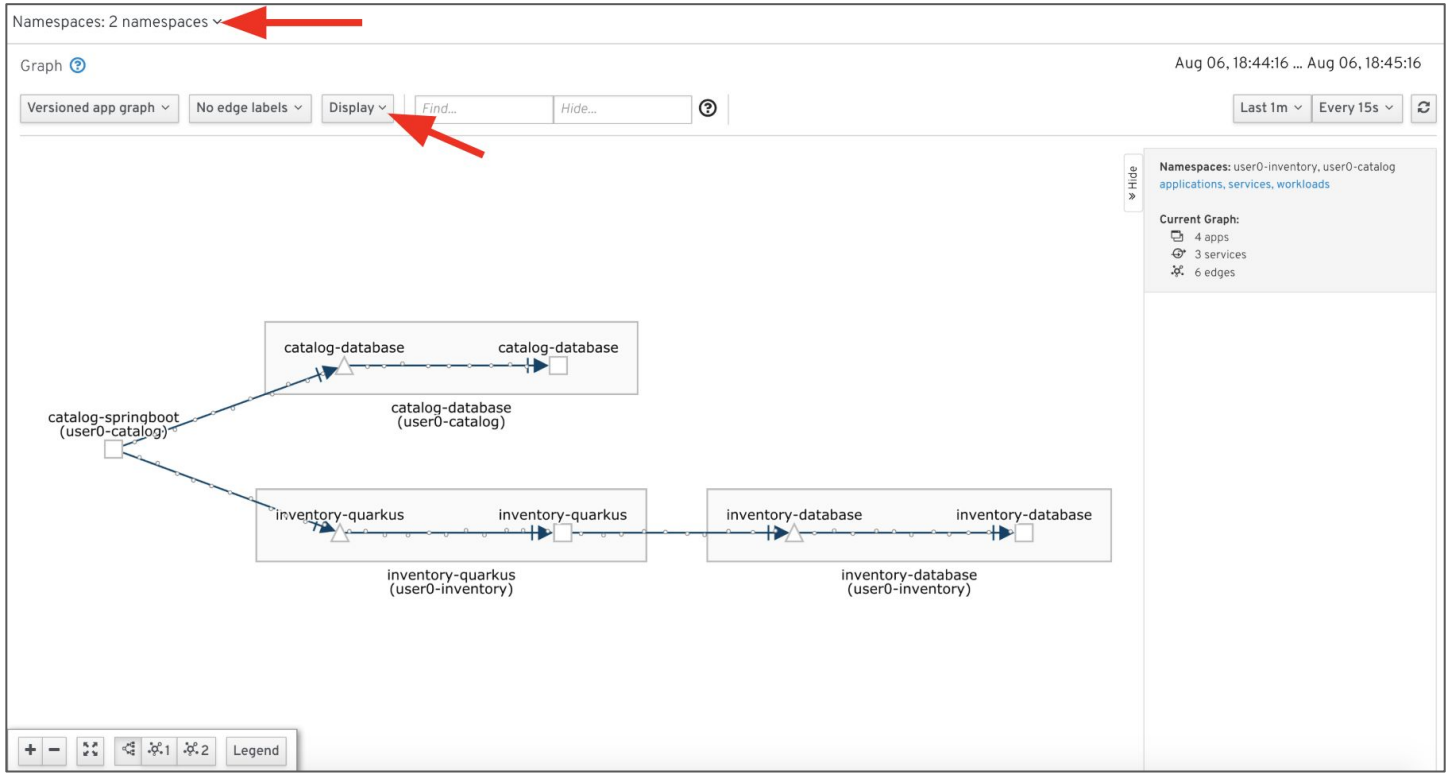
Application Topology & Service Versioning

kiali

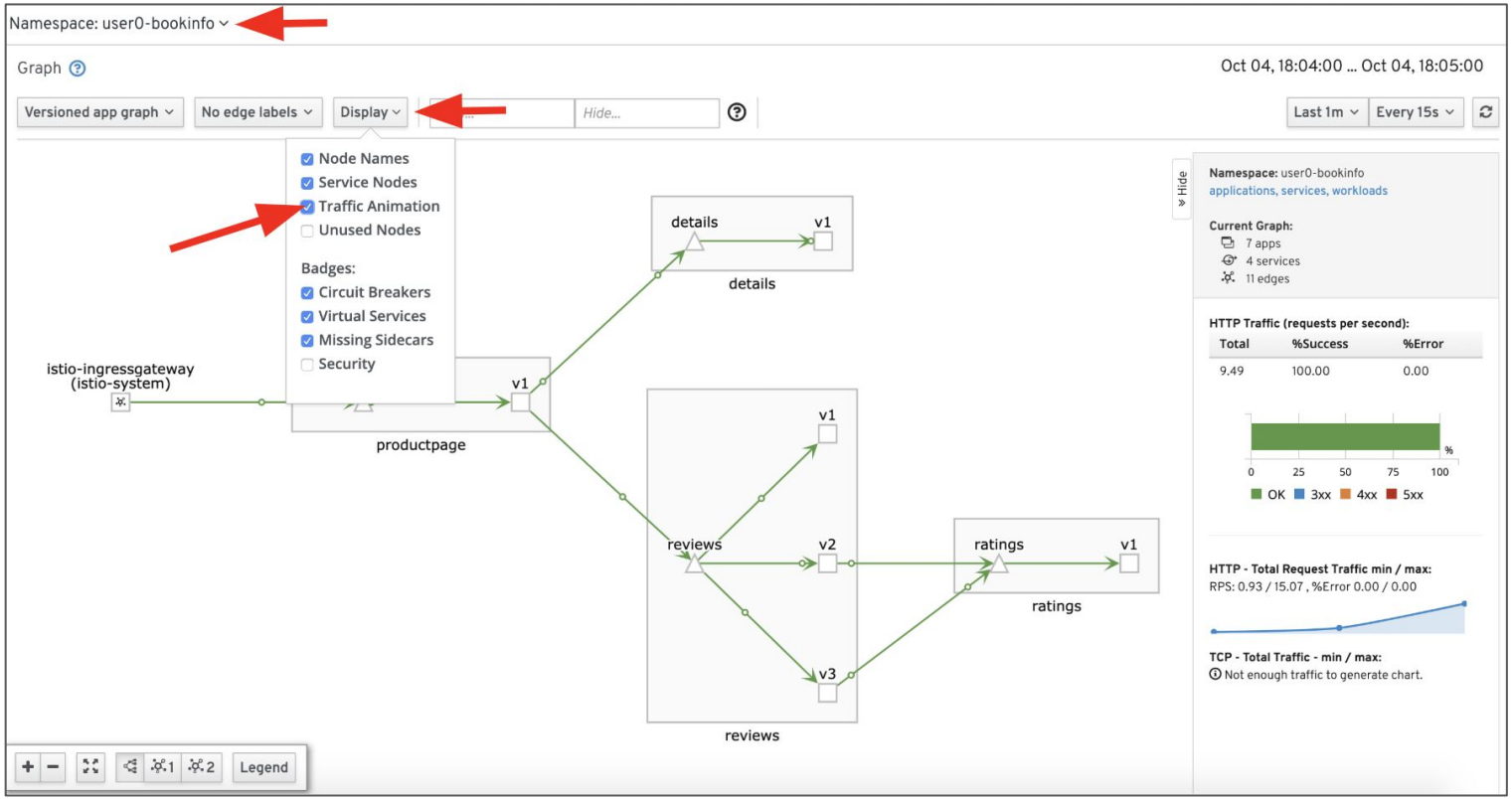


Red Hat

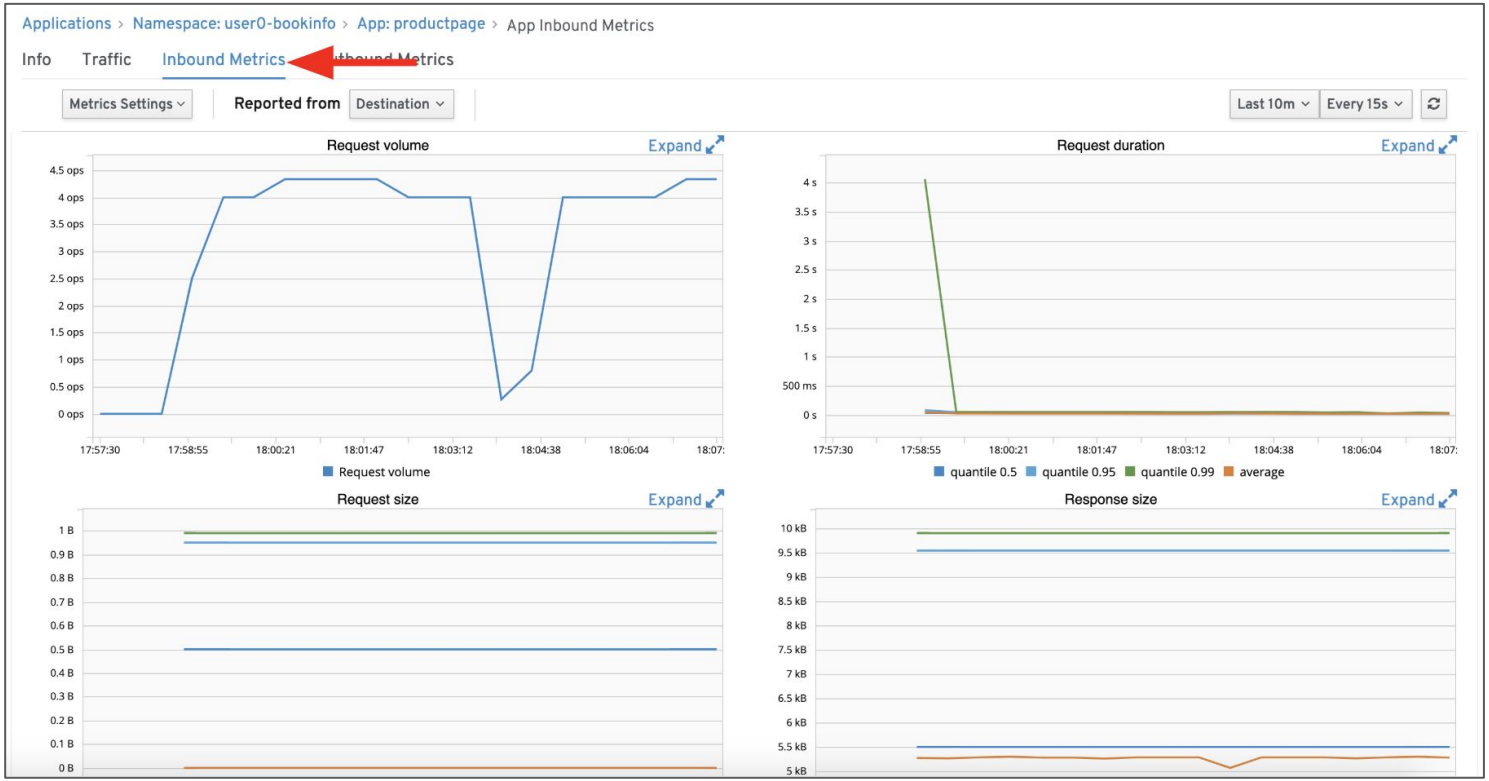
Cross-Application Views



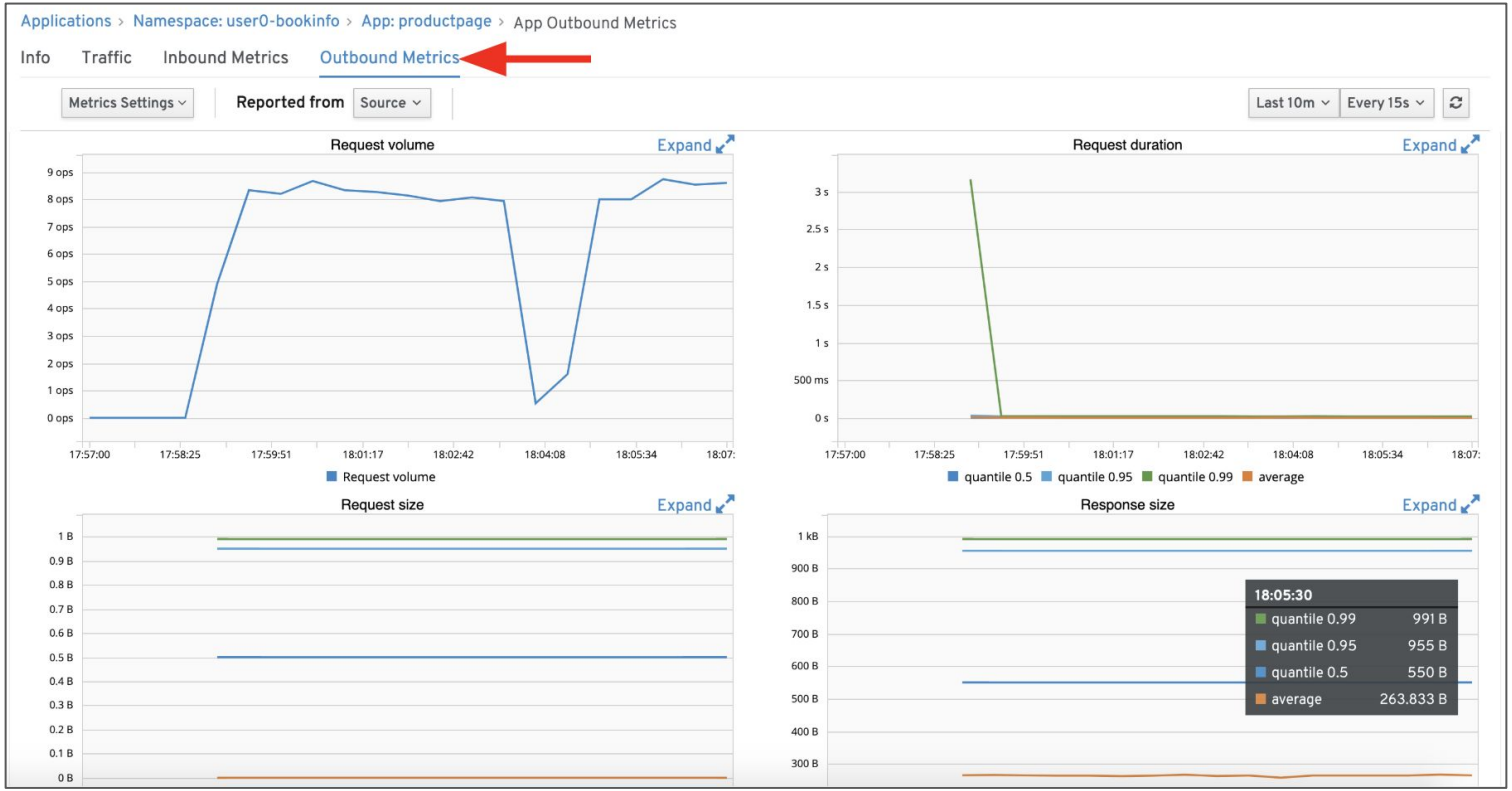
Traffic Throughput



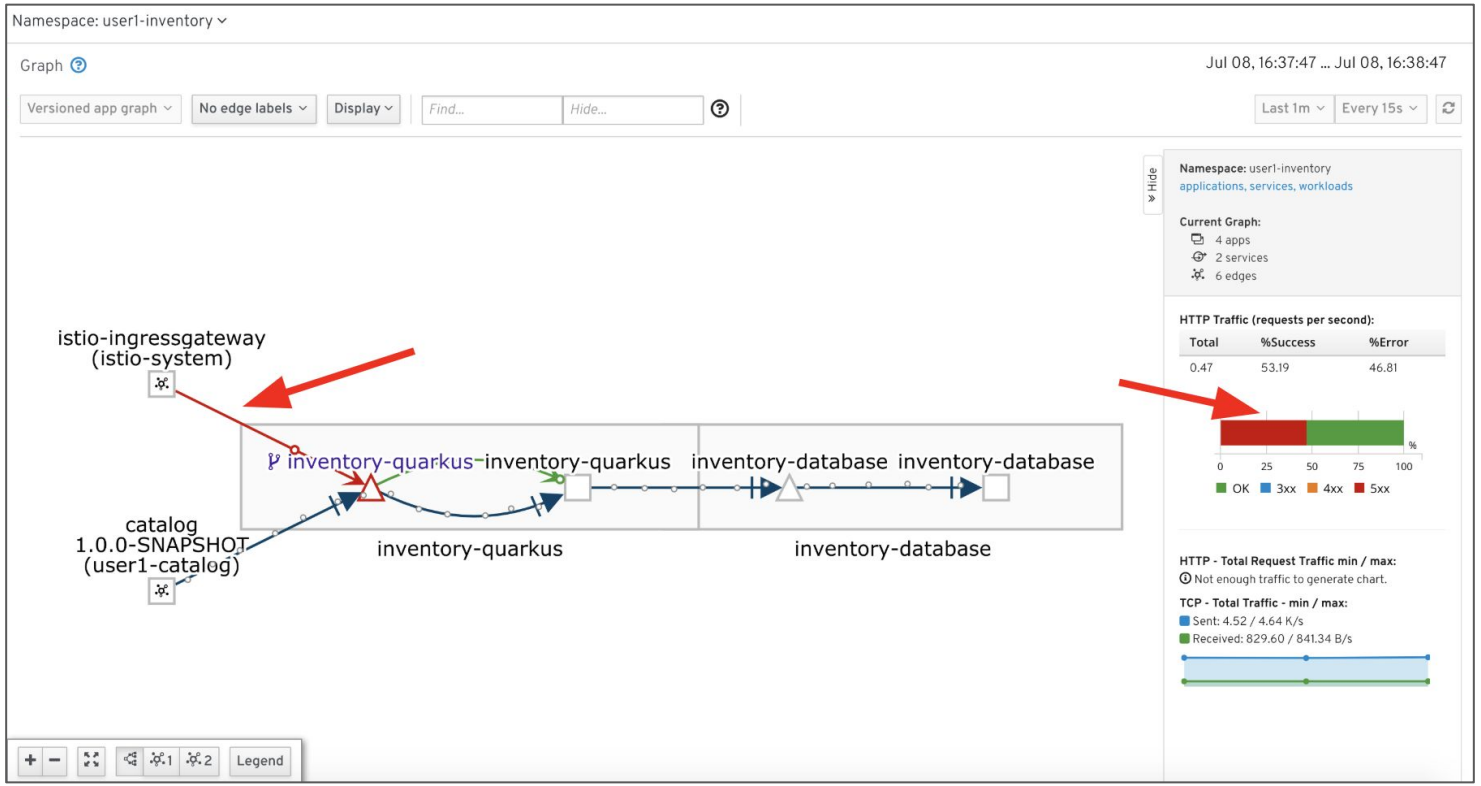
Traffic Throughput Details: Inbound



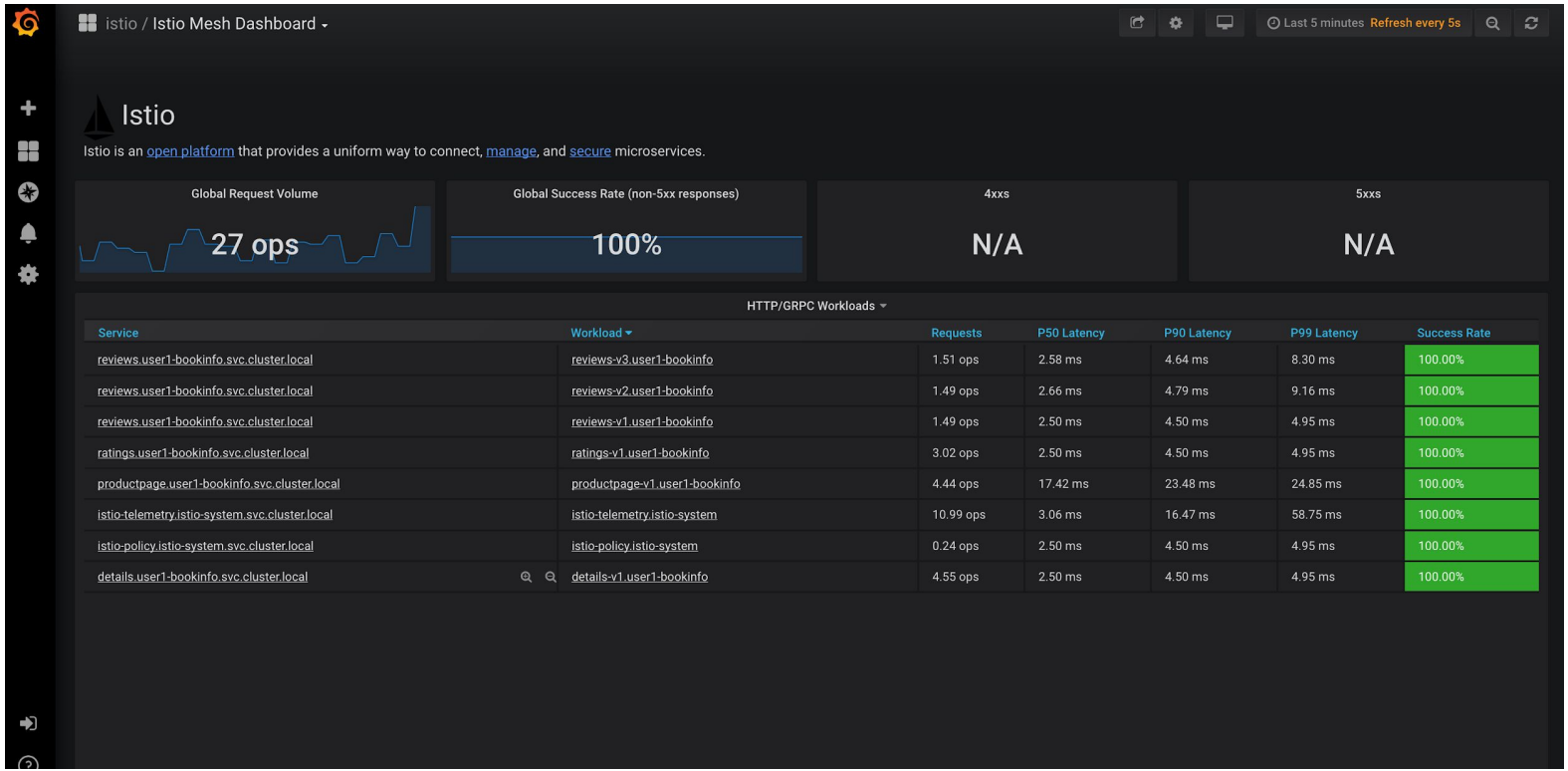
Traffic Throughput Details: Outbound



Errors and Error Rates

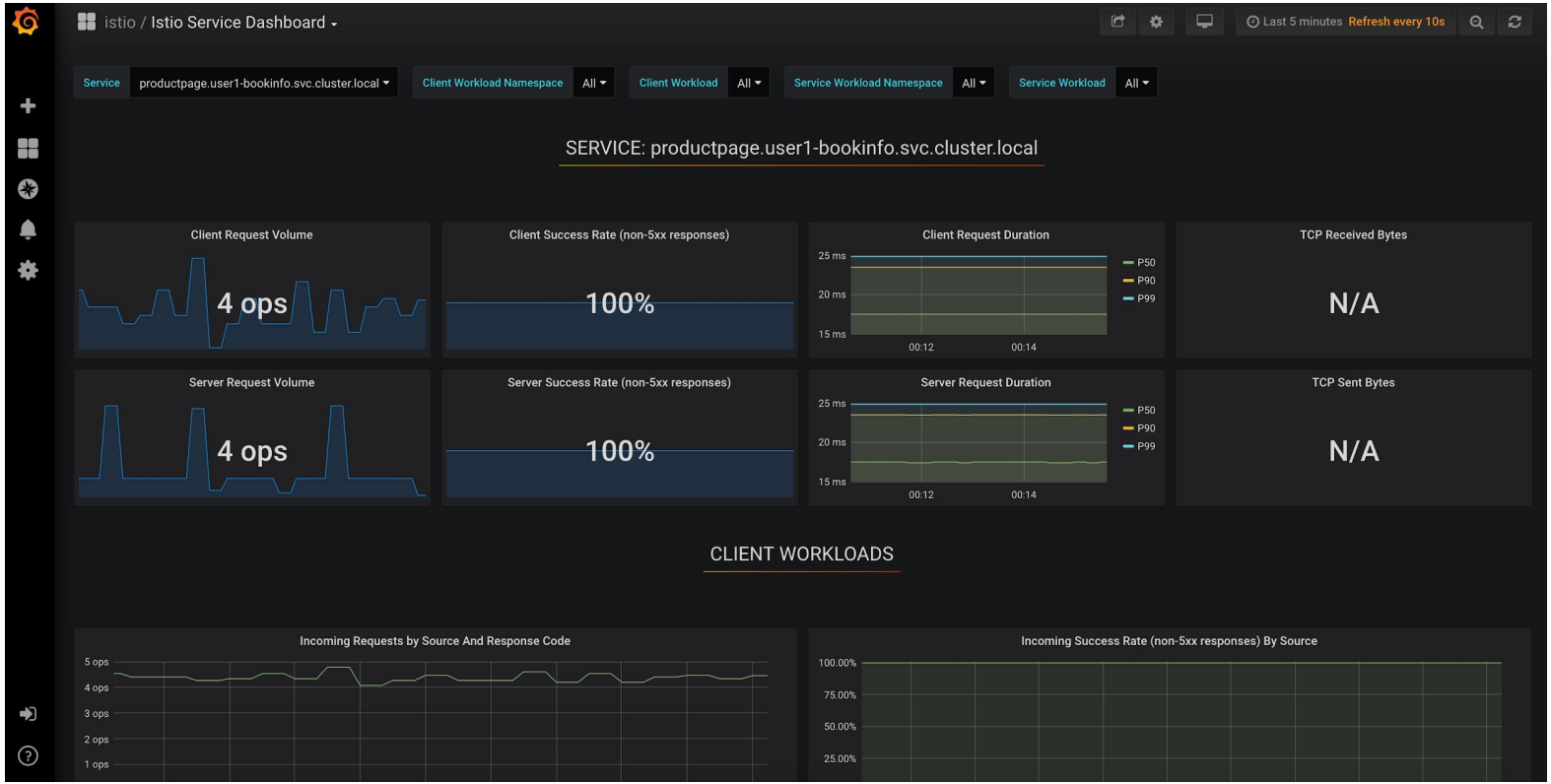


Global Dashboarding



Red Hat

Global Dashboarding



Distributed Tracing with Jaeger

- Discover service relationships and process times, transparent to the services
- Visualize the service execution times across the application
- Identify potential latency issues in each service

Red Hat

Service Mesh Use Cases

Service mesh enables a number of capabilities and key use cases:

- Traffic Management
- Production Support: Observability
- Production Support: Application Tracing
- Role and auth checks between applications/Lines of Business
- “Zero-trust” Network

Reducing Installation and Management Overhead

Kubernetes Operator model

- Single package Install
- Service Mesh Operator reduces complexity to get running quickly
- Installation, configuration and updates of all components in one place
- Best practices and human operational knowledge baked-in for installation, configuration and upgrades

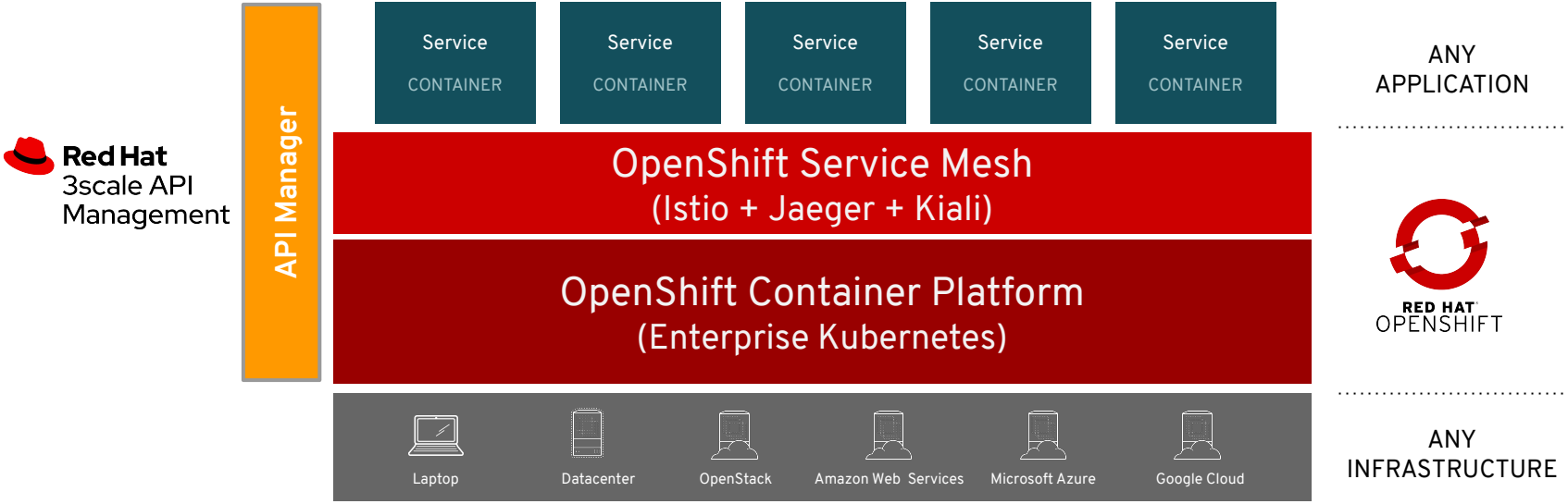
API Management with Service Mesh

- Do you have tens / hundreds of services / APIs?
- Are applications consuming your APIs internal services?
- Need to package those services into consumable API products?
- Are there different types of customers for the API's? E.g. internal applications, partner applications, etc.
- Do you need a portal where customers can explore available API products and get immediate access to them?

API Management with Service Mesh

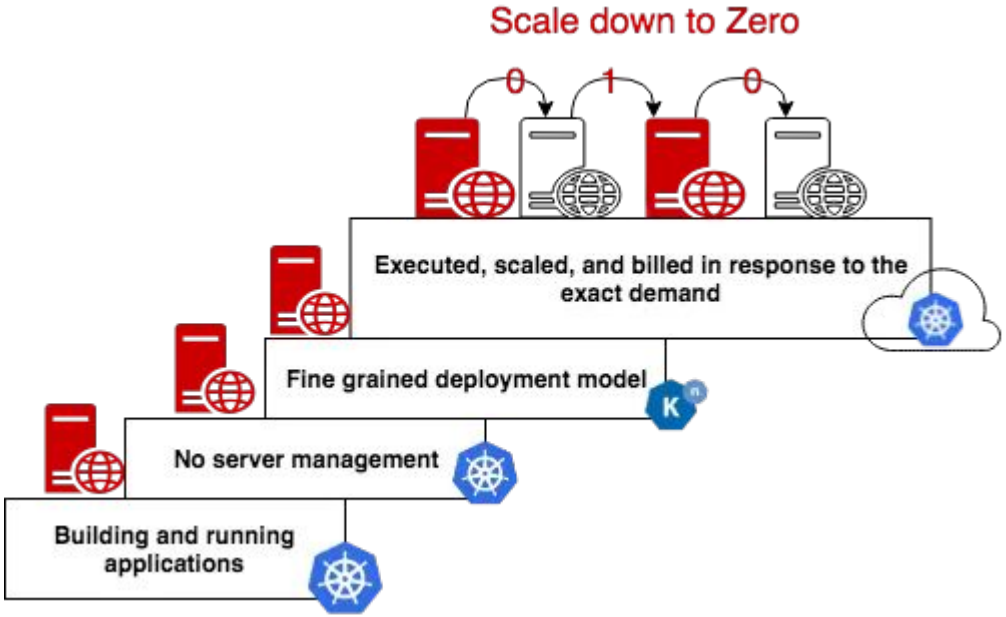
- The 3scale Istio Mixer Adapter gives services exposed within the service mesh API management capabilities.
- Developer access via developer portal and documentation, configuring different types of access for different type of developers, usage analytics, billing and invoicing.
- Quota enforcement, caching, and analytics are available at the ‘API product’ level.

Distributed Services Platform



Serverless Architecture:

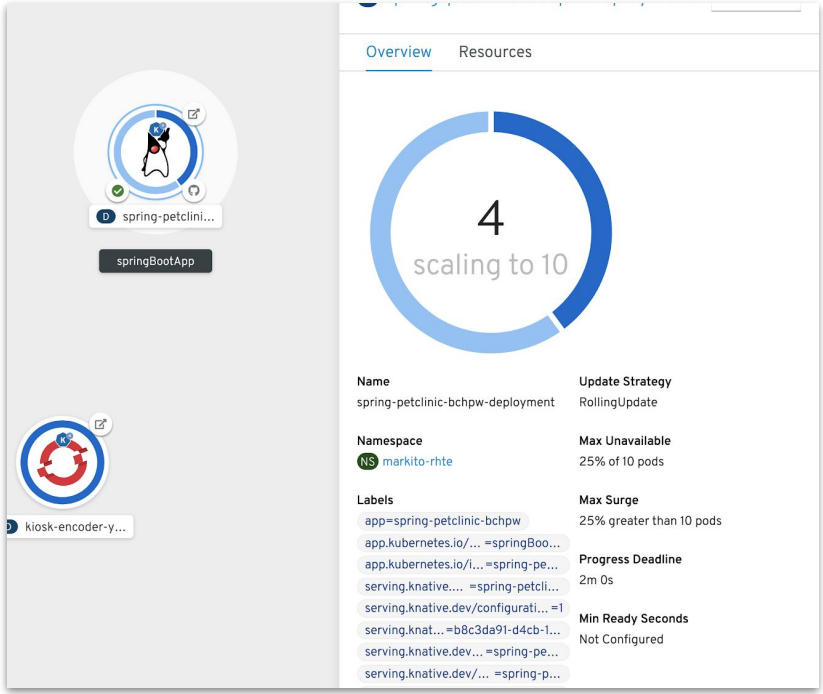
AutoScaling, Event-driven Architecture



Red Hat

Serverless Architecture:

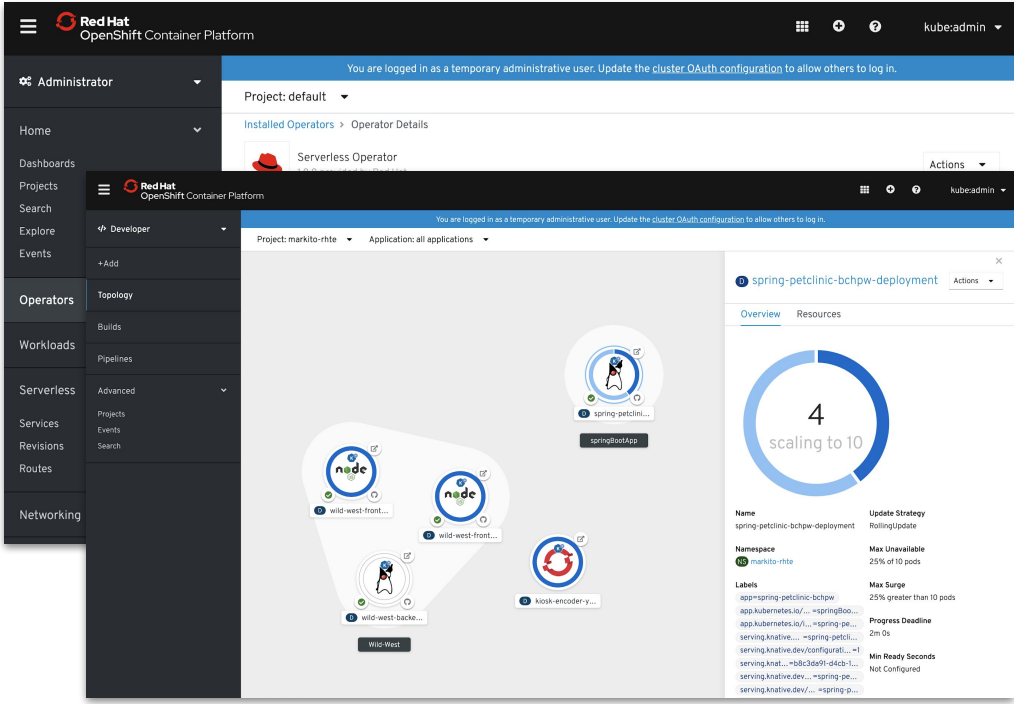
AutoScaling, Event-driven Architecture



OpenShift Serverless: Built on Service Mesh

Key Features

- Familiar to Kubernetes users, native to K8s
- Scale to 0 and autoscale to N based on demand
- Applications *and* functions. Any container workload.
- Powerful eventing model with multiple event sources.
- Operator available via OperatorHub
- Knative v0.7.1 (v1beta1 APIs)
- No vendor lock in

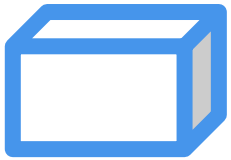


Learn more

https://openshift.com/learn/topics/serverless

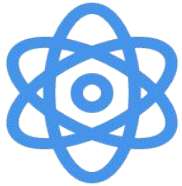
https://redhat-developer-demos.github.io/knative-tutorial

Growth in Application Architecture Choices



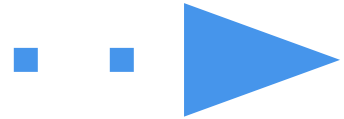
Monolith

Cloud Native



Microservices

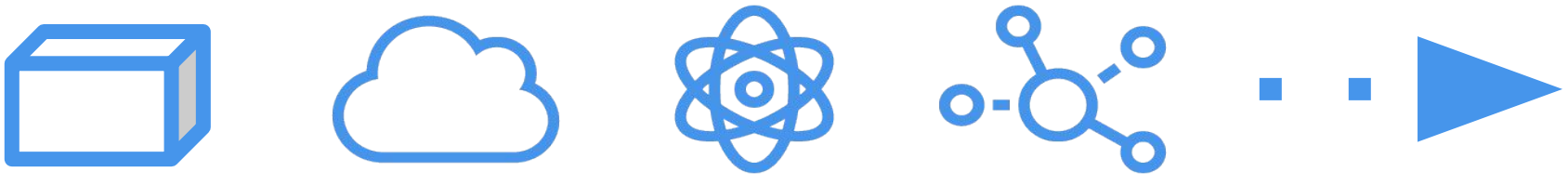
Serverless



Event-Driven Architecture

Red Hat

Complete Platform for Your Architecture Choices



Monolith

Cloud Native

Microservices

Serverless

Event-Driven Architecture

kubernetes

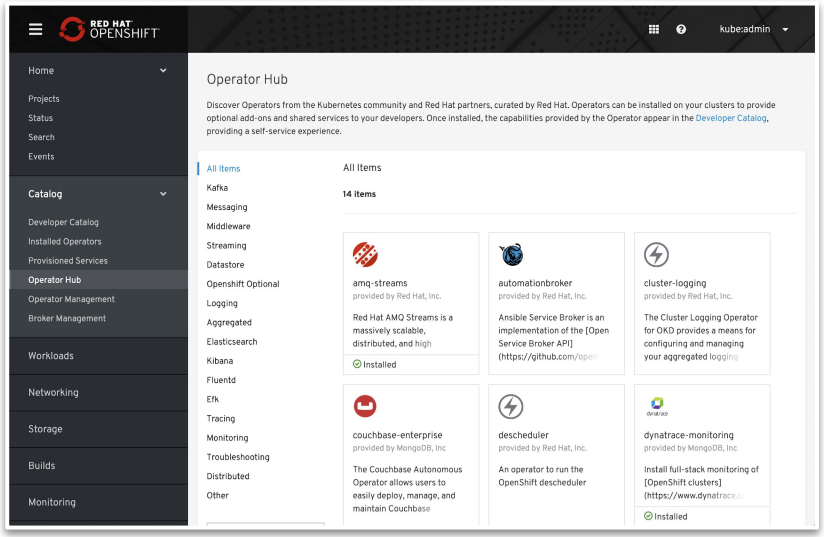
Istio

Knative

Red Hat

OpenShift Service Mesh

- OpenShift Service Mesh is available and supported at no additional cost with OpenShift 4
- OpenShift Service Mesh Operator is found in the OperatorHub menu



Red Hat

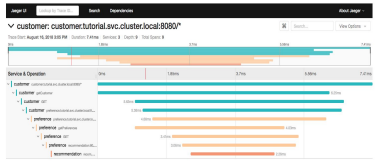
Next Step: Free Hands-On Training

Online Labs: https://learn.openshift.com/servicemesh/

- Istio Architecture
- Microservices deployment into a Service Mesh
- Monitoring, tracing
- Traffic routing
- Fault injection
- Circuit breakers
- Egress security
- Mutual TLS security

Red Hat

Demo and Q&A



Jaeger

Secure

Istio

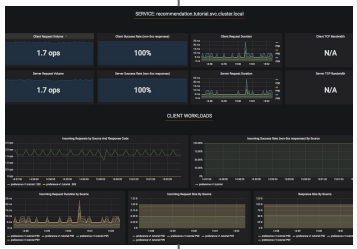
Observe

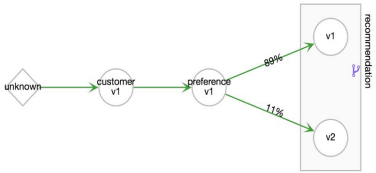
Connect

Control

Prometheus

Observe





kiali

Kiali

Grafana

Red Hat

Next Step: Free Hands-On Training

Online Labs: https://learn.openshift.com/servicemesh/

- Istio Architecture
- Microservices deployment into a Service Mesh
- Monitoring, tracing
- Traffic routing
- Fault injection
- Circuit breakers
- Egress security
- Mutual TLS security

Red Hat

Thank You!

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat

Reference Links

- Product Overview
- Service Mesh Release Announcement
- Product Documentation
- Service Mesh E-book
- Online Service Mesh Training