THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
JAPAN

Introduction to service mesh with Istio and Kiali

Alissa Bonas

mikeyteva

THE LINUX FOUNDATION

Evolution of application architecture

•••

How did we get to service mesh?

Monolith application

Single unit of executable

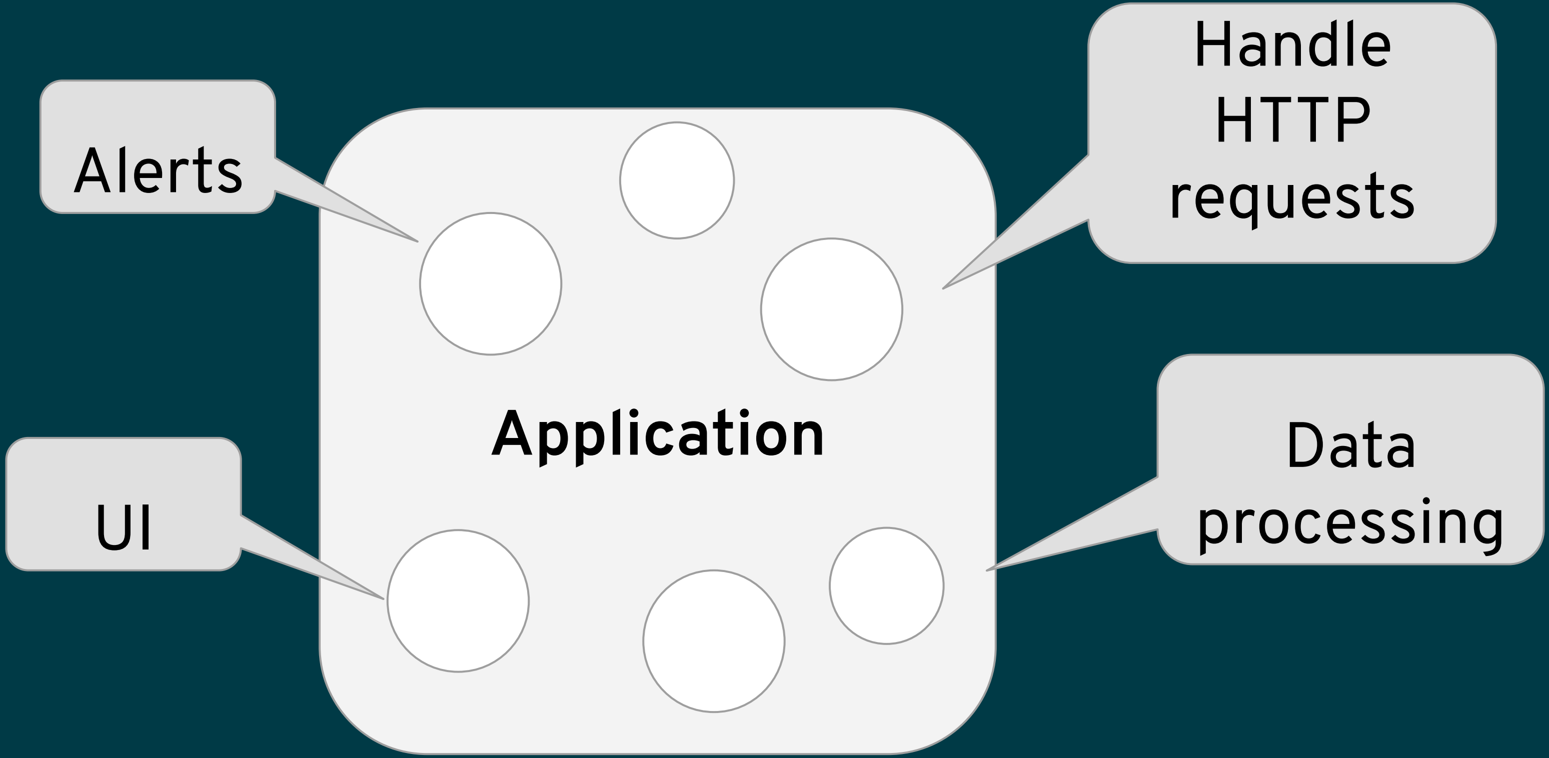
=

Application

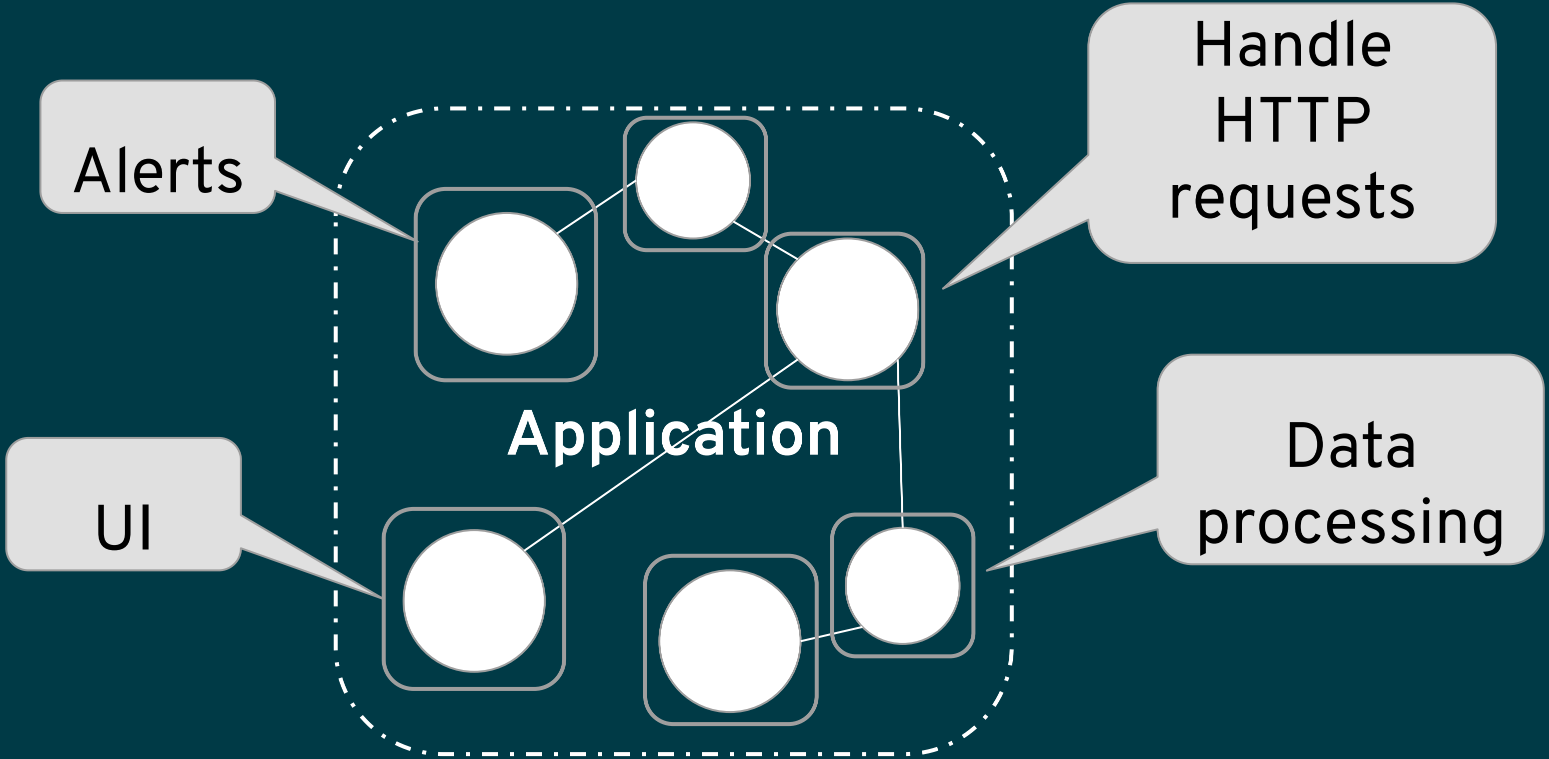
=

Single process

Application modules



Multiple processes

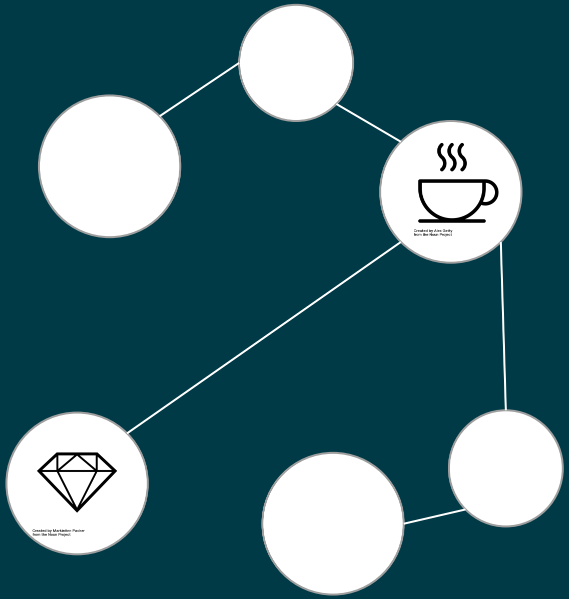


Microservices

Language agnostic

Scaled separately

Upgraded separately



A shift in Application Packaging and Runtime

Containerizing an app

Run multiple containers

Orchestrate containers

- Run many containers on multiple hosts
- Scale - manage several instances (replicas) of the same container
- Manage a container based environment

Container orchestration platforms

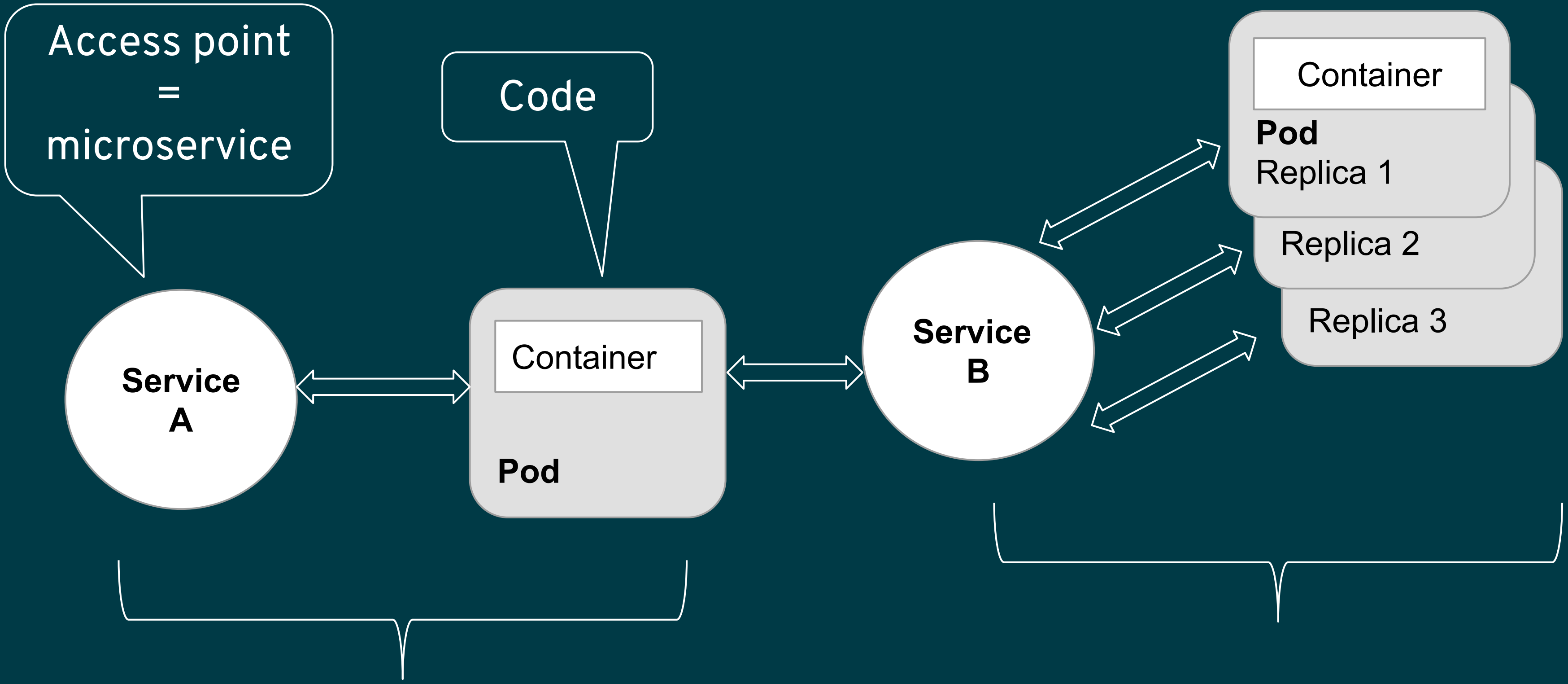
Kubernetes

OKD
(Openshift)

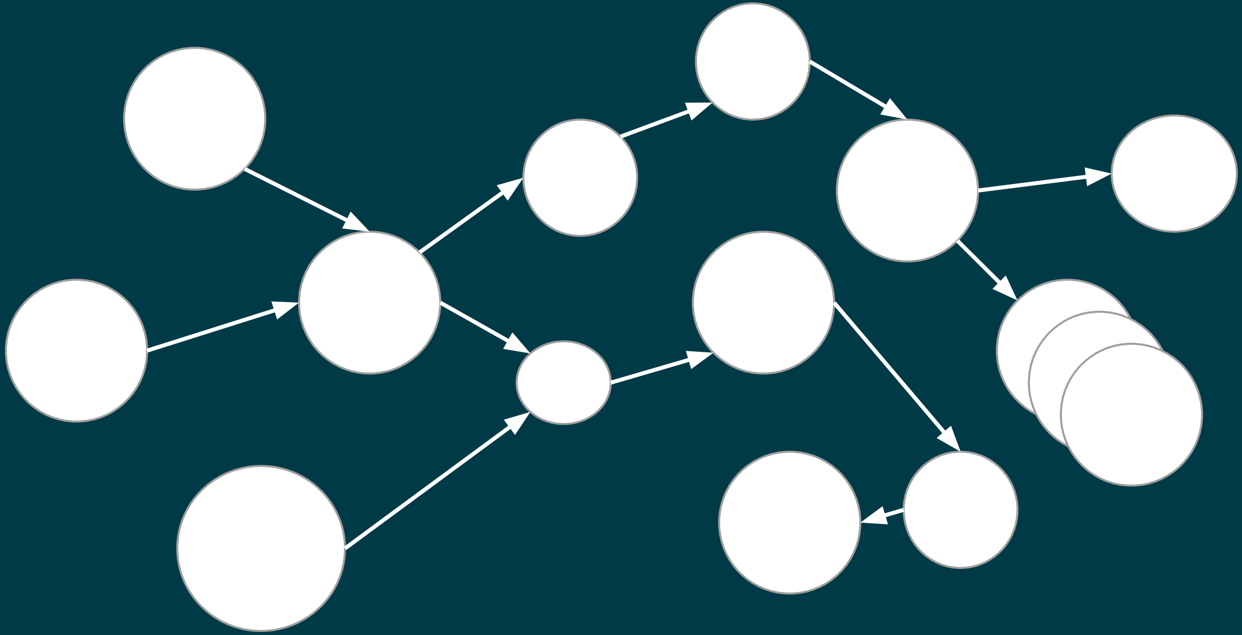
Kubernetes building blocks (some...)

- Pod - a group of one or more containers, with shared storage/network
- Deployment - manages pod definition and defines replicas of pods
- Service - an abstraction, an access point to a set of Pods
    - Sometimes called a **microservice**

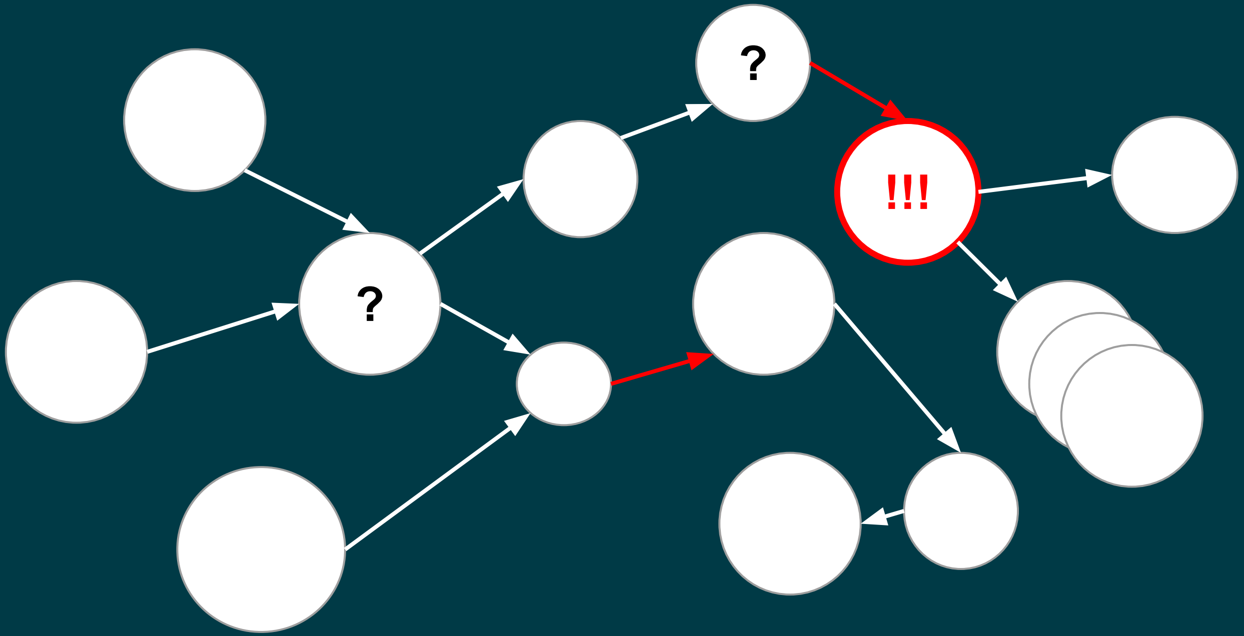
Microservices - the Kubernetes way



High Complexity



Multiple points of failure



Challenges

- How are the requests routed between services?
- How do I detect failures and downtime?
- How to upgrade and test new versions of a service?
- Securing the communication

Service mesh to the rescue

What is a service mesh

- Infrastructure/framework that handles communication between services
- Often implemented as network proxies deployed alongside the microservices

Istio - Ιστίο

Open source service mesh

The dry facts

- Started in May 2017
- Means “sail” in Greek
- Developed in Go

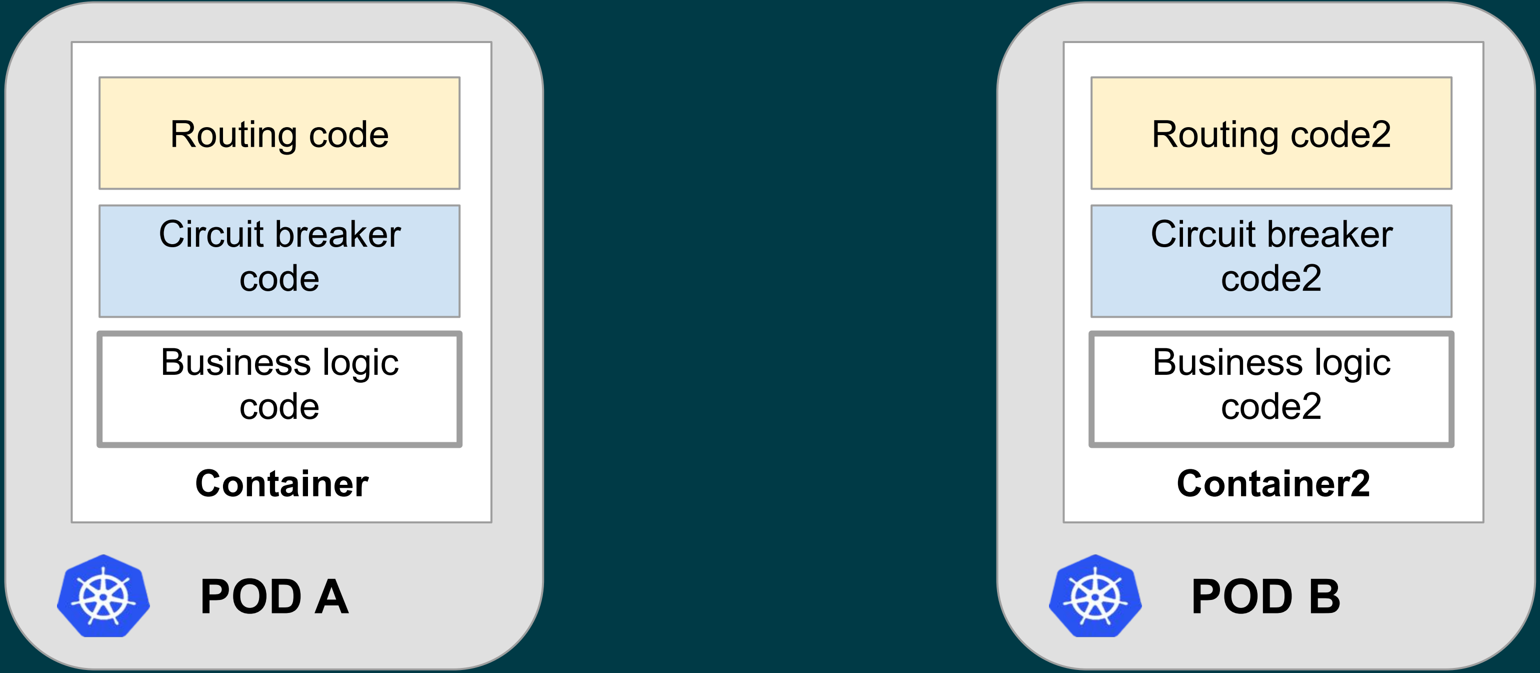
Istio features

- Load balancing (HTTP, gRPC, TCP...)
- Traffic control (routing rules, retries, timeouts, fault injection, mirroring)
- Secure service-to-service communication
- Access controls (authorization)
- Metrics and traces for traffic

Important Terminology

- Workload - anything owning/controlling pods (like a Deployment) or the pods themselves
- Service - a **microservice**
- Application - *label* “app” on a pod/service
- Version - *label* “version” on a pod/service

Before Istio



Routing code

Circuit breaker code

Business logic code

Container

POD A

Routing code2

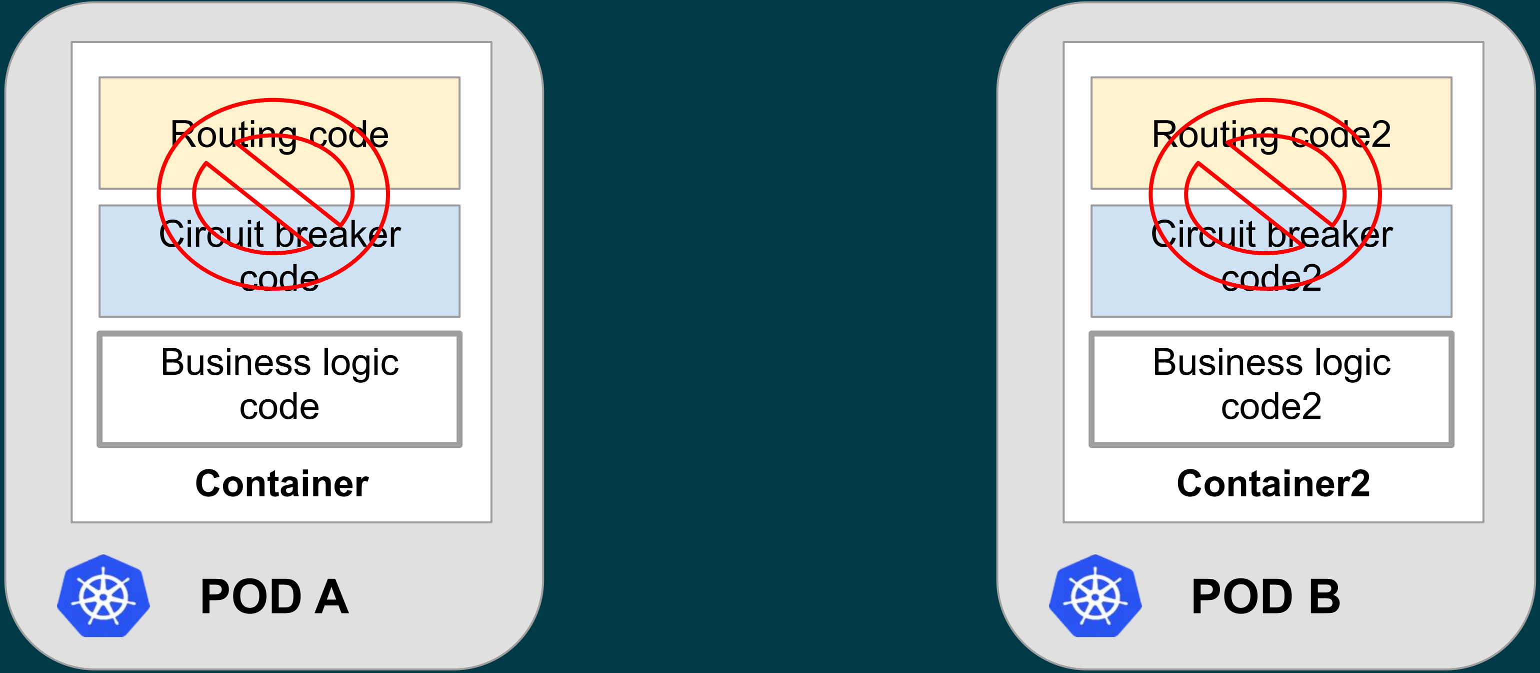
Circuit breaker code2

Business logic code2

Container2

POD B

Istio



Routing code
Circuit breaker code
Business logic code
Container
POD A
Routing code2
Circuit breaker code2
Business logic code2
Container2
POD B

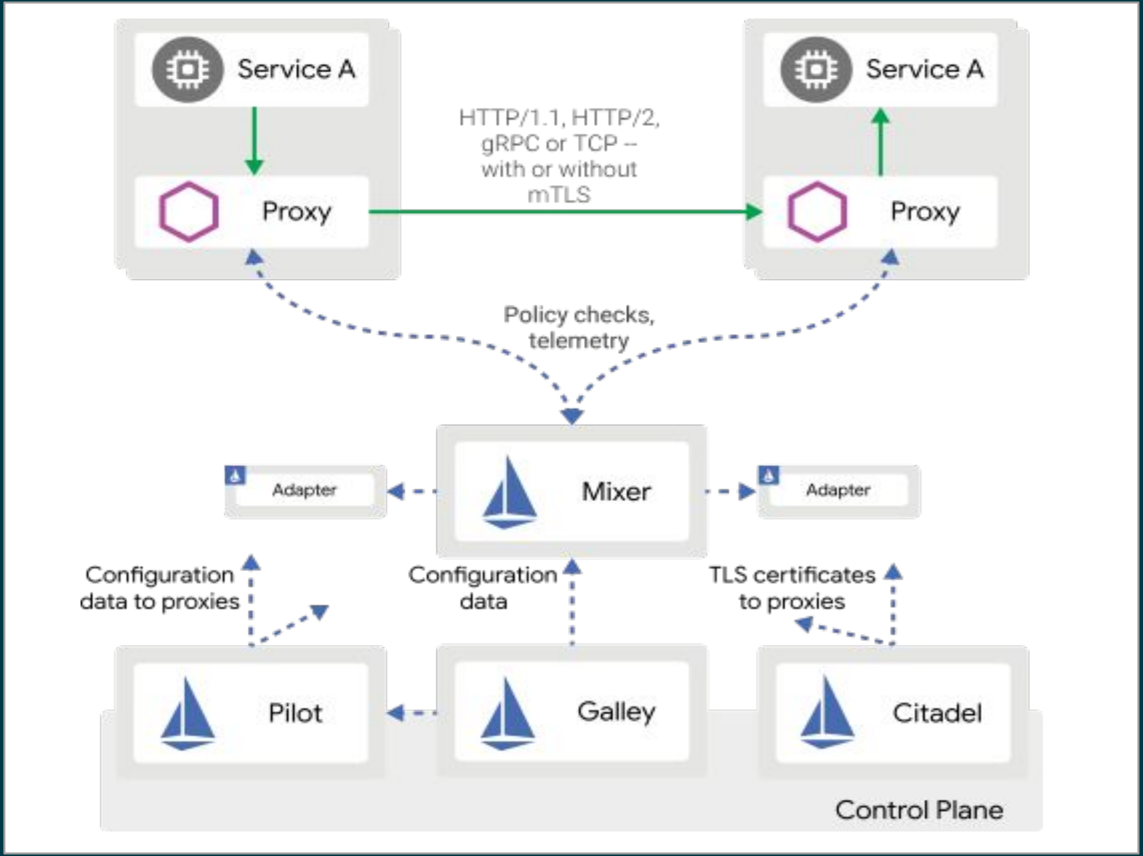
Sidecar Proxy

- A proxy is deployed in a container next to each instance of microservice (inside a pod)
- Container name: istio-proxy
- It is **transparent** to application code
- Envoy open source proxy is currently used

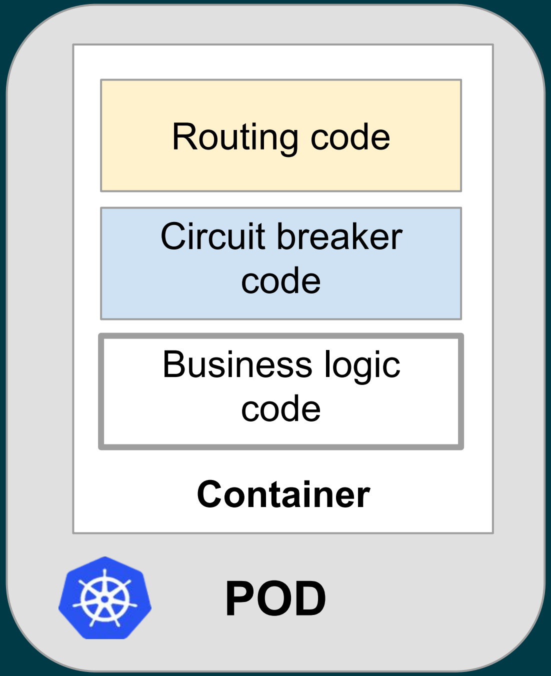
How is the sidecar injected?

- Manually
- Automatically injected to pod on creation
    - *kubectl label namespace default istio-injection=enabled*
    - Mutating Admission Webhook is used for sidecar injection
    - Actually... 2 containers are injected: istio-init and istio-proxy

Istio architecture



Sidecar Proxy in Istio and Kubernetes



Routing code

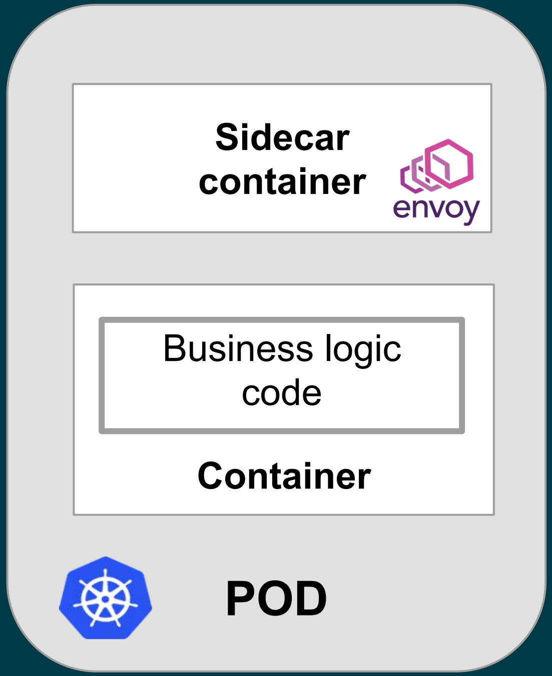
Circuit breaker code

Business logic code

Container

POD

Before Istio, no sidecar



Sidecar container

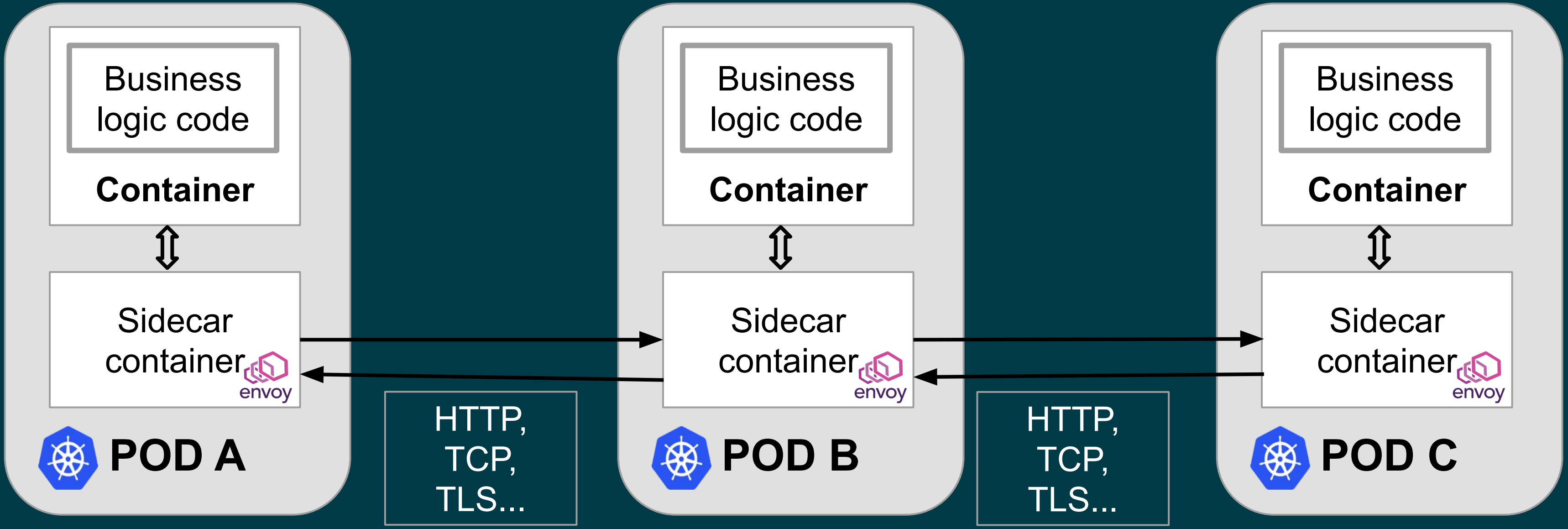
Business logic code

Container

POD

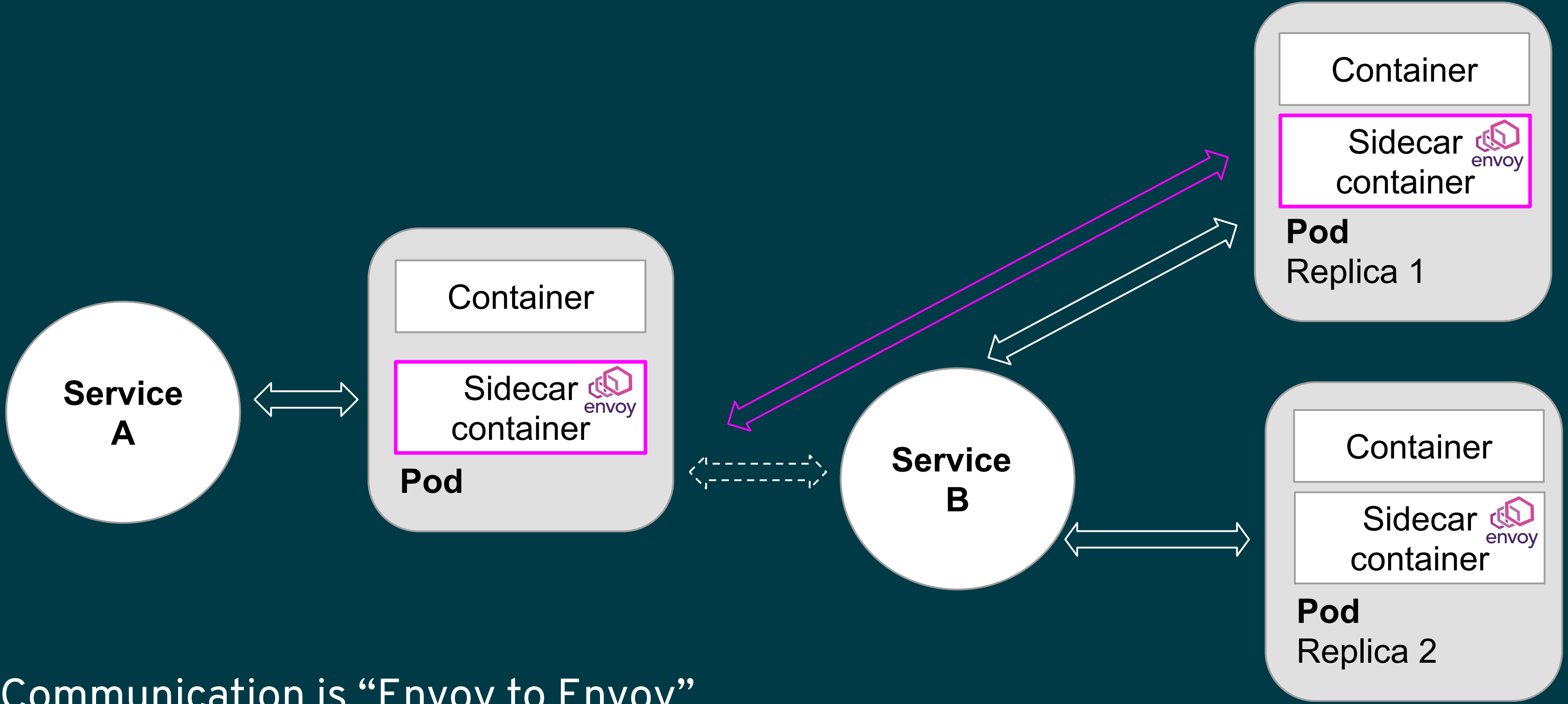
With sidecar

With Istio - sidecar intercepts all traffic



Configuration is transparent to the services and not part of the code

Istio routing in Kubernetes

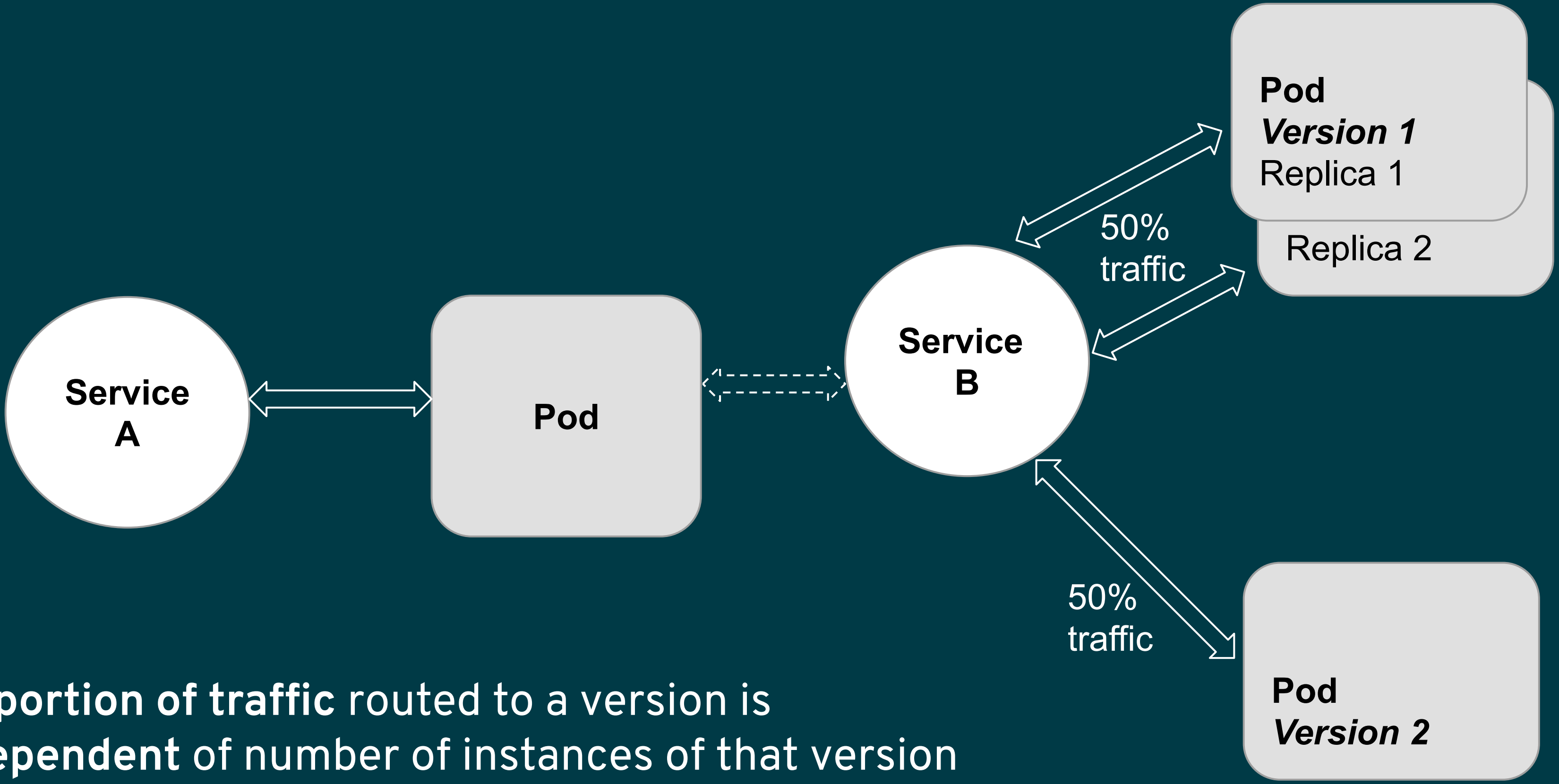


Communication is “Envoy to Envoy”
bypassing the Kubernetes Service

Different routing scenarios

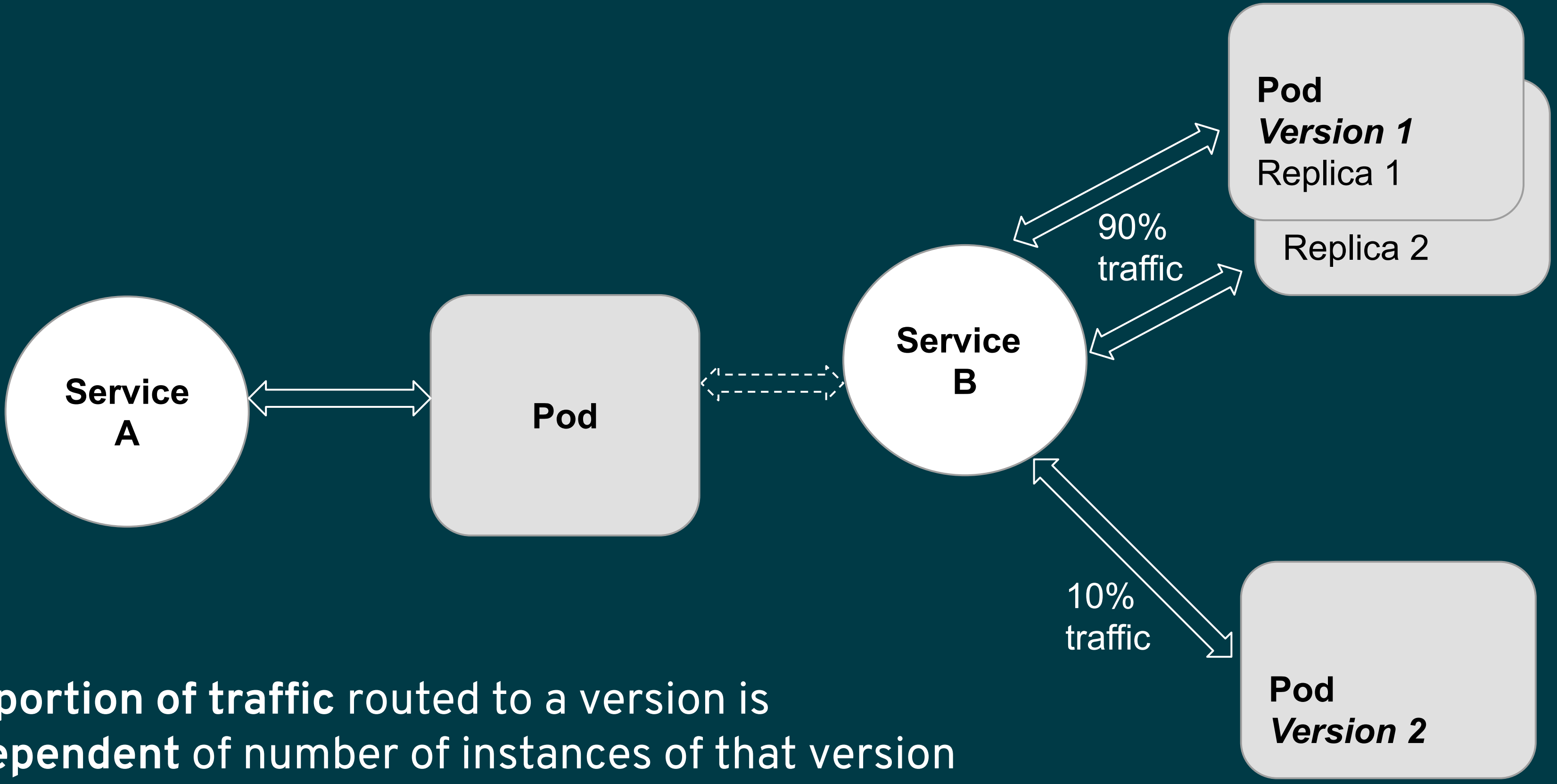
- A/B testing
- Traffic shifting
    - Canary deployment (an example of traffic shifting)
- Mirroring traffic

Weighted Routing with Istio - A/B



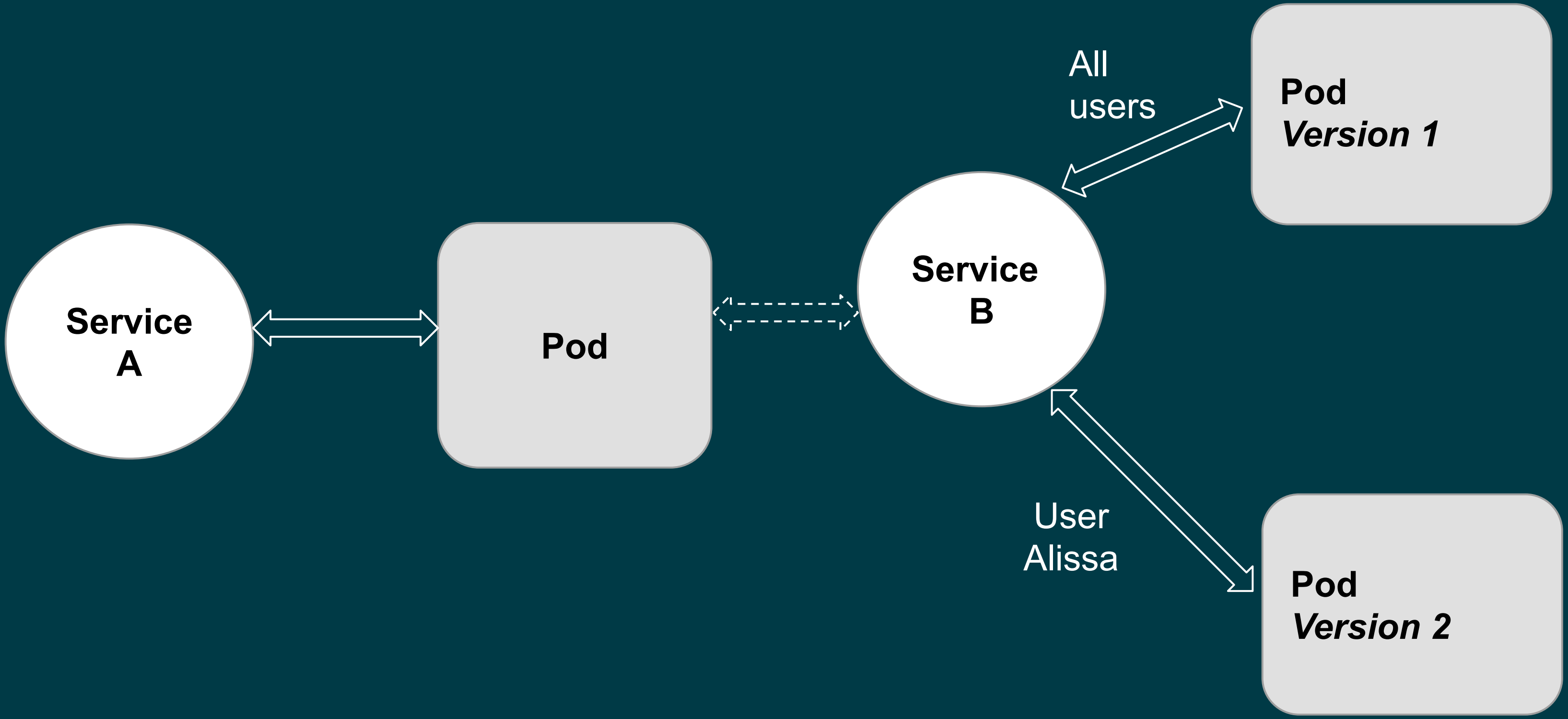
Proportion of traffic routed to a version is **independent** of number of instances of that version

Weighted Routing - Canary

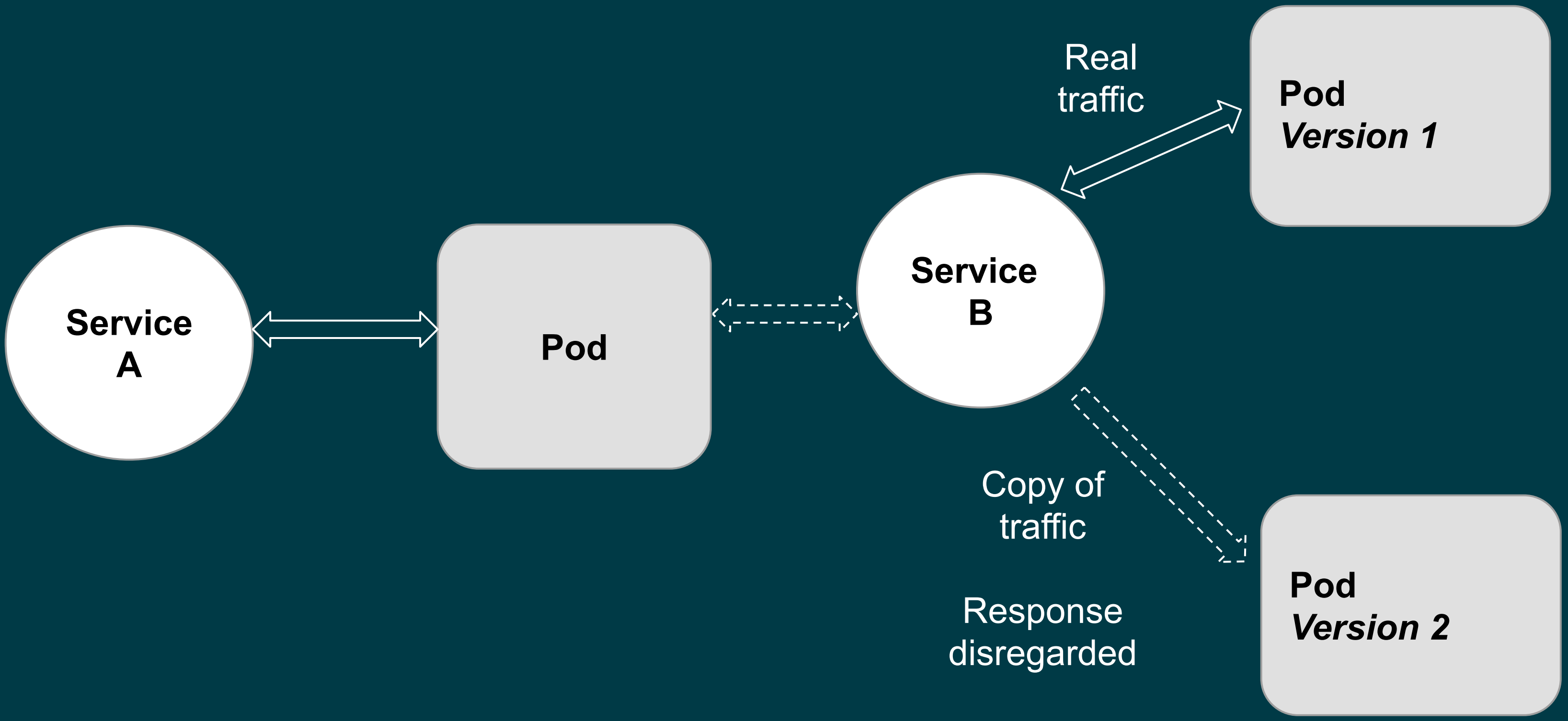


Proportion of traffic routed to a version is **independent** of number of instances of that version

Matching Routing with Istio

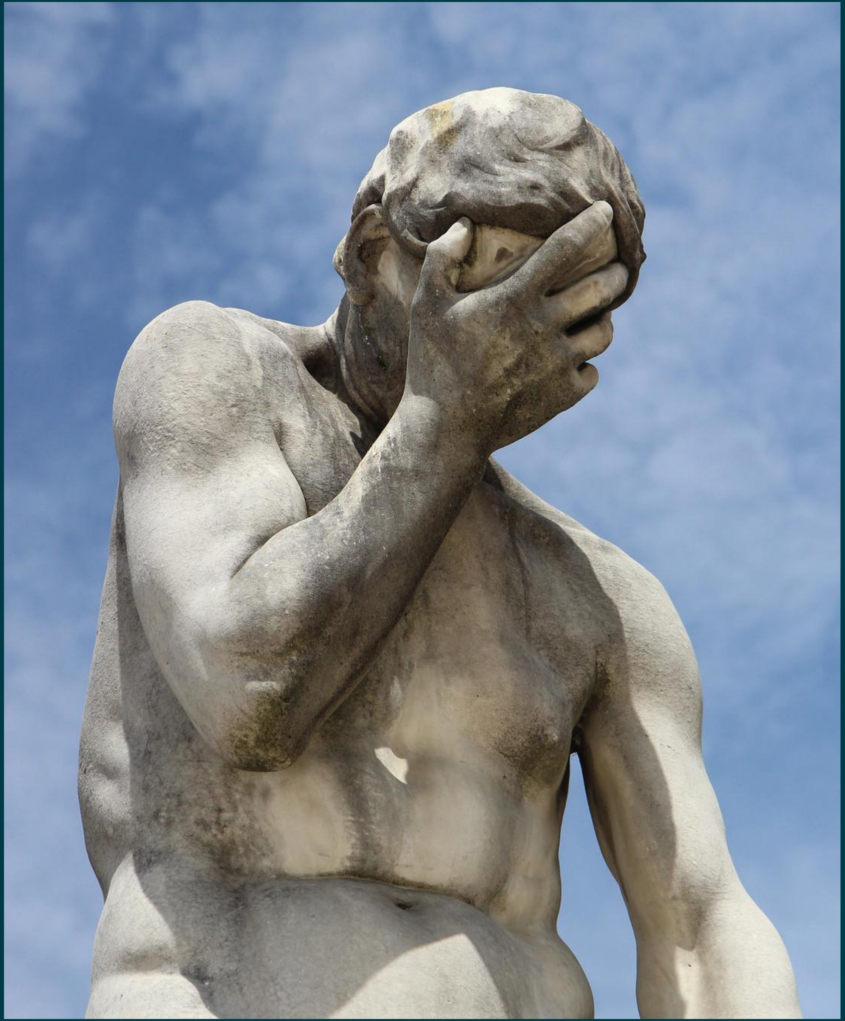


Mirroring traffic



"Anything that can go wrong will go wrong"

(Murphy’s law)



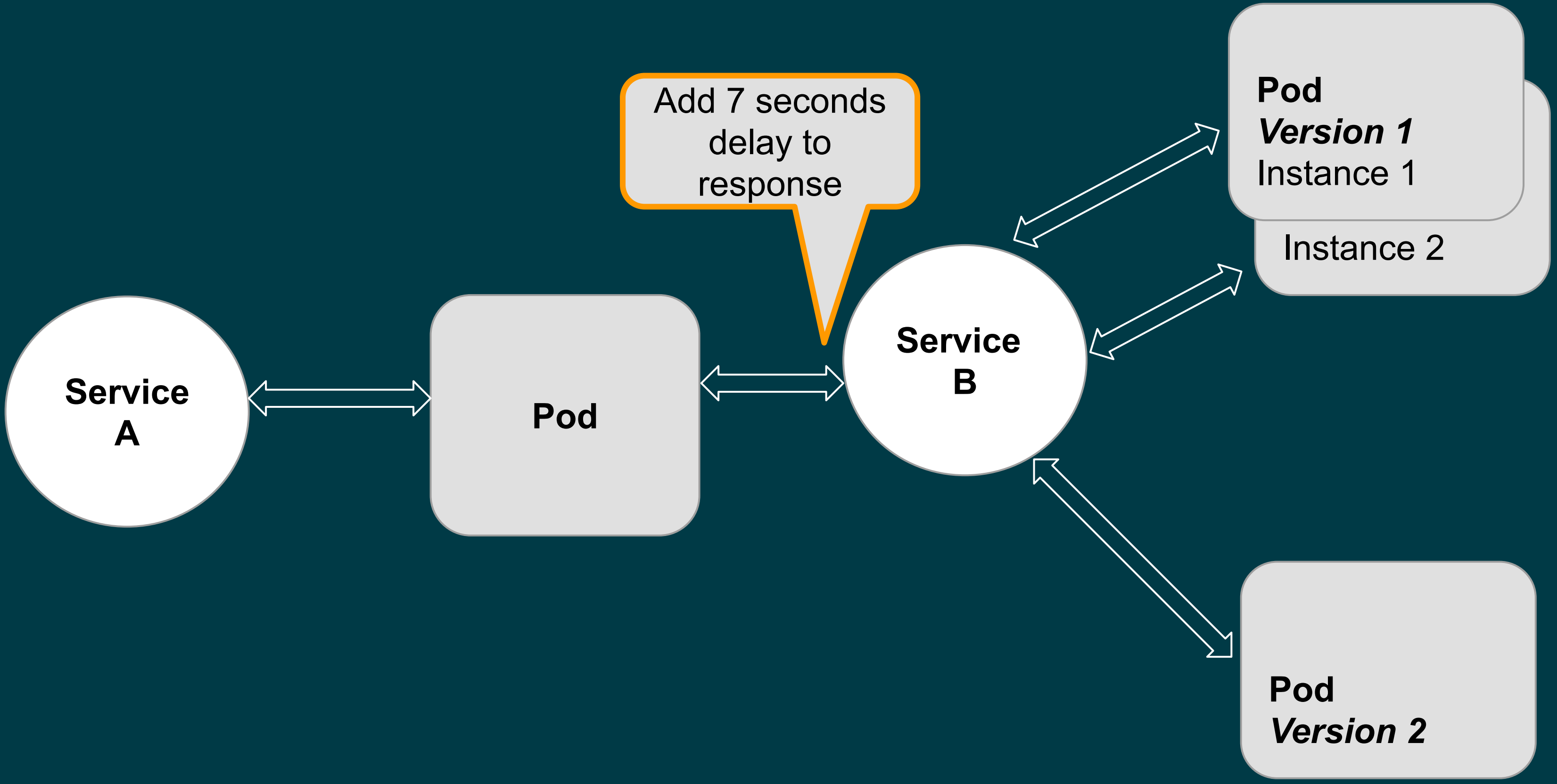
KEEP
CALM
AND
DO
CHAOS
ENGINEERING

Chaos engineering with Istio

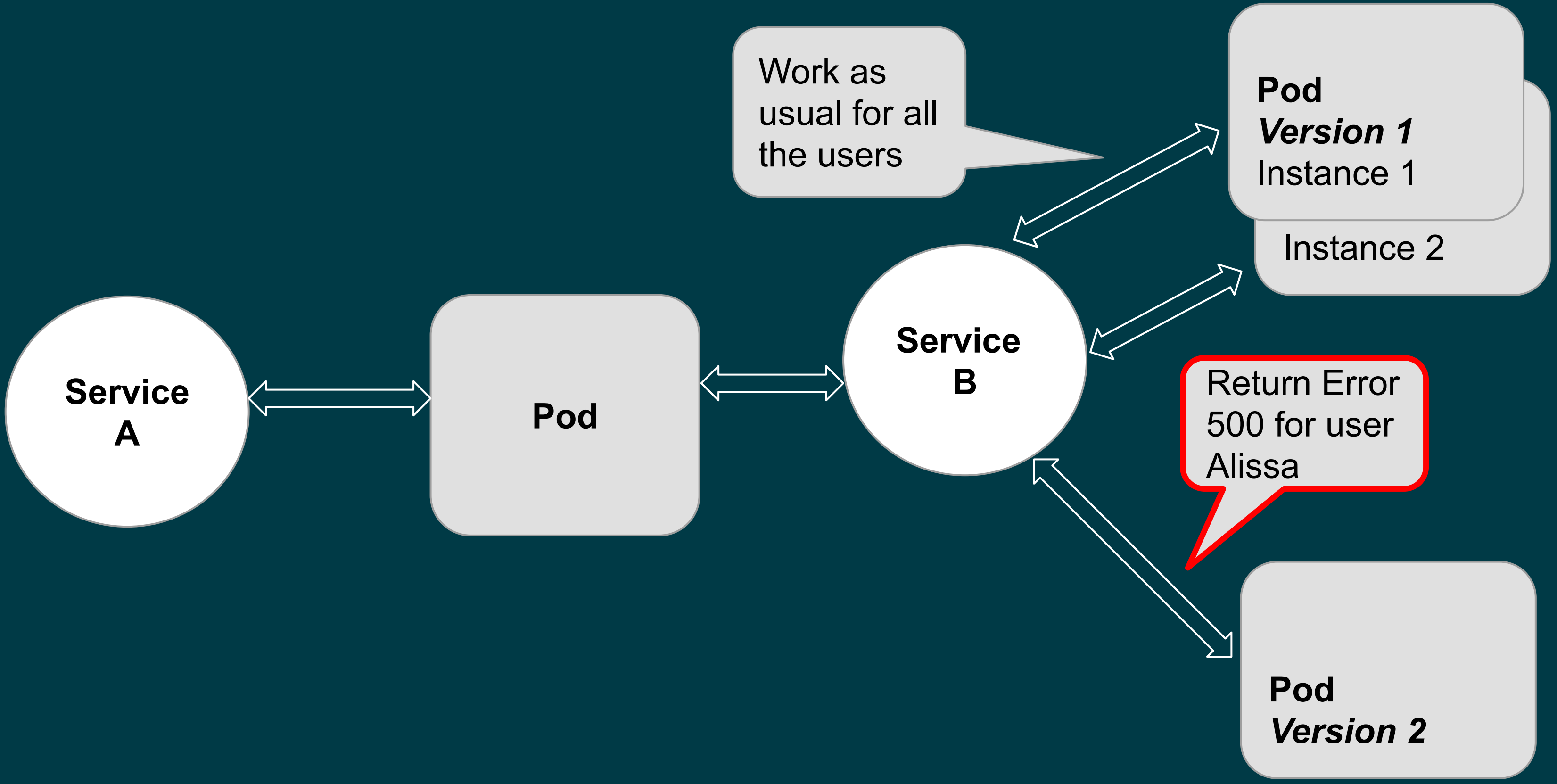
- Inject delays
  - Simulate network latency
  - Simulate an overloaded service

- Define aborts (Inject Errors)
  - Simulate failure in a service (return a predefined HTTP Error)
  - A good alternative for a manual shutdown or “scale to zero”

Inject delay



Inject Error



Work as usual for all the users

Return Error 500 for user Alissa

Circuit breaker

- Set a connection pool to limit connections and requests
- **Example:** “Set a connection pool of 100 connections with no more than 10 req/connection to service A”

Outlier detection

- Classify instances as healthy/unhealthy
- Eject unhealthy instances for a defined timeframe which can be increased over time
- **Example:** “Scan all pods every 5 mins, any instance that fails 7 consecutive times with 5XX error code will be ejected for 15 minutes.”

Authorization and Authentication

- Authentication
    - End user authentication (JSON Web Token (JWT))
    - Service to service authentication (mutual TLS)
        - Permissive mode is possible for flexible migration
- Authorization
    - Can service <A> send <this request> to service <B> ?
    - Roles are visible across namespaces
    - ServiceRole and ServiceRoleBinding

Security

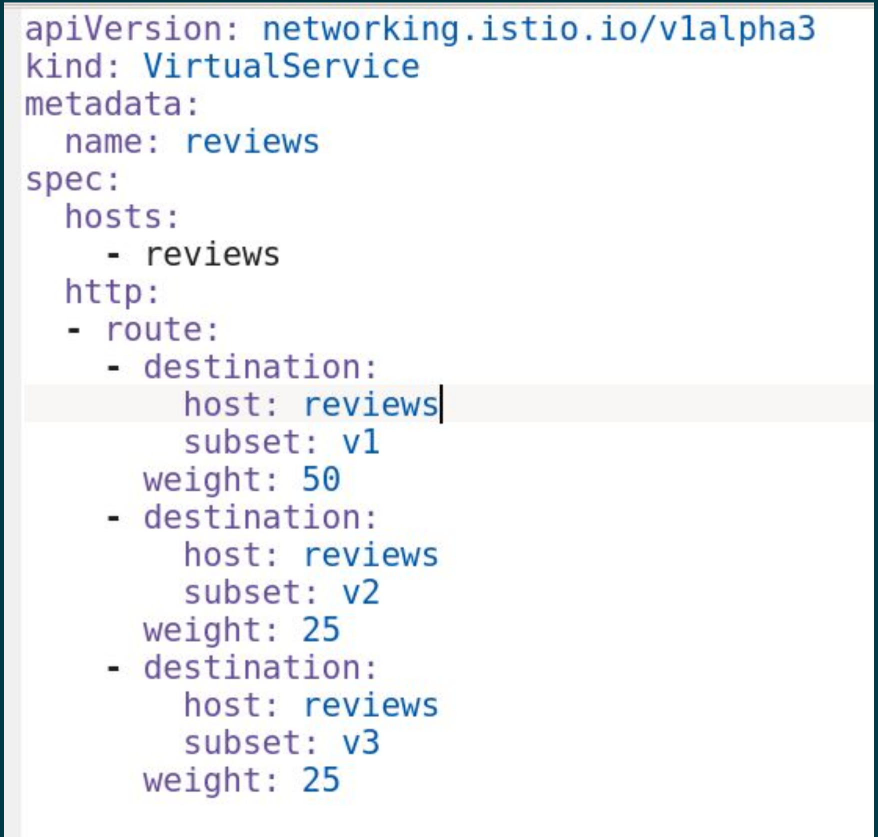
- Defining a Gateway ingress/egress to enable traffic in/out of mesh
- Citadel monitors service accounts creation and creates a certificate for them
    - Certificates only in memory, sent to Envoy via SDS API
- mTLS can be defined on multiple levels
    - Client and server exchange certificates, 2 way
    - All mesh, specific service, etc.

Configuration objects

- VirtualService != Kubernetes service
    - Rules for how requests to a service are routed within service mesh
    - Routing logic, load weighting, chaos injection
- DestinationRule
    - Configures policies to be applied to a request **after** VirtualService routing has occurred
    - Load balancer, circuit breaker
- MeshPolicy, Gateway, ServiceEntry and more…

Configuration Yaml example

All Istio objects are CRD (CustomResource Definition)



```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
      weight: 50
    - destination:
        host: reviews
        subset: v2
      weight: 25
    - destination:
        host: reviews
        subset: v3
      weight: 25
```

New set of challenges

- How many versions exist for service A?
- Is there any traffic **now**?
- Is **routing configured** for service B?
- Is my configuration **valid**?
- Is security **on**?
- Is the app **healthy**?

Kiali - Κιάλι

•••

Open source
Istio service mesh observability

Dry facts

- Started in January 2018
- Means “spyglass” or “monocular” in Greek
- Developed in Go and React

Kiali Features

- Visualize mesh connections and traffic
- Service and application health
- Configure routing via UI
- Validate Istio configurations
- View metrics, traces and logs
- Visualize security configuration

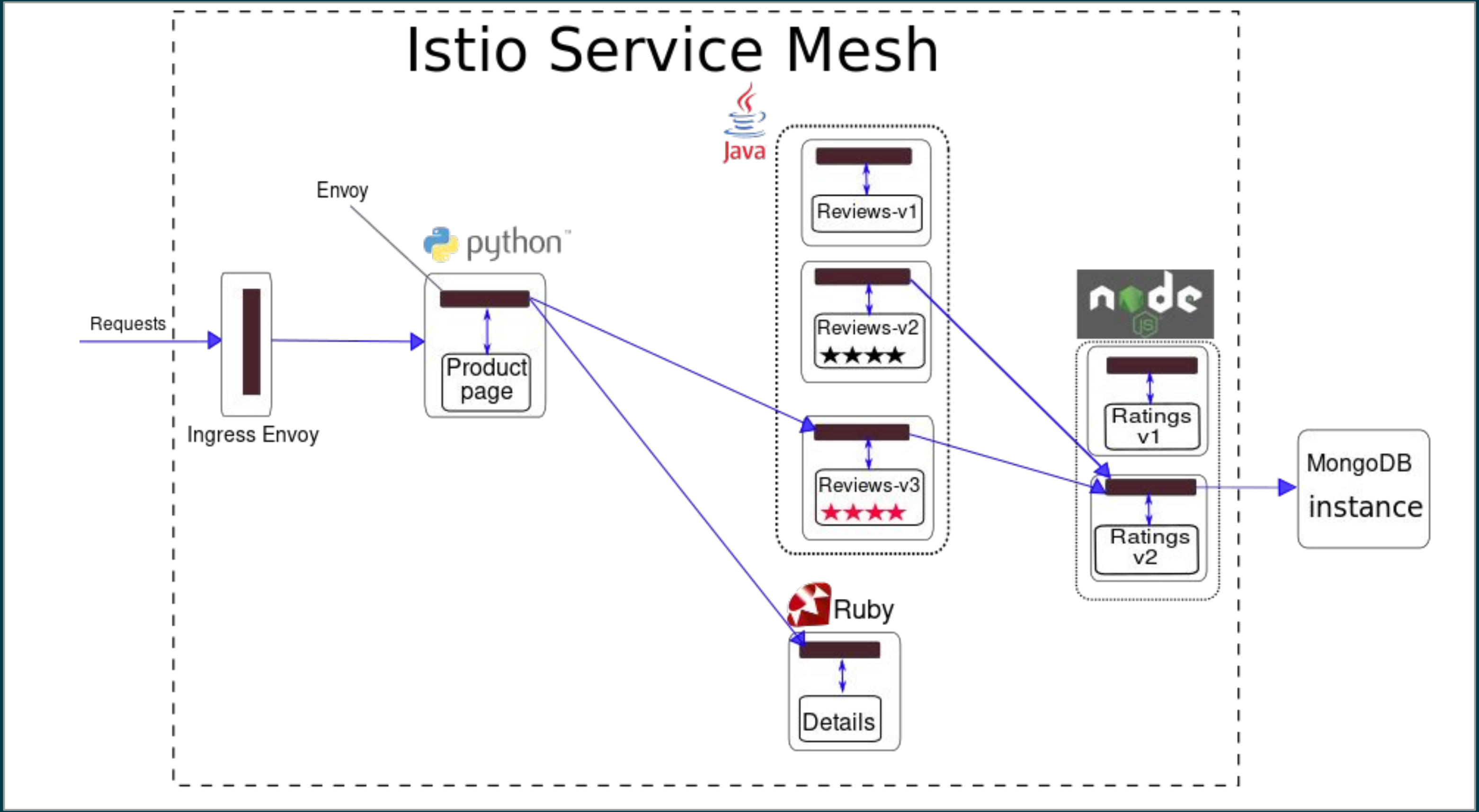
A picture is worth a thousand yamls

Demos based on Bookinfo example

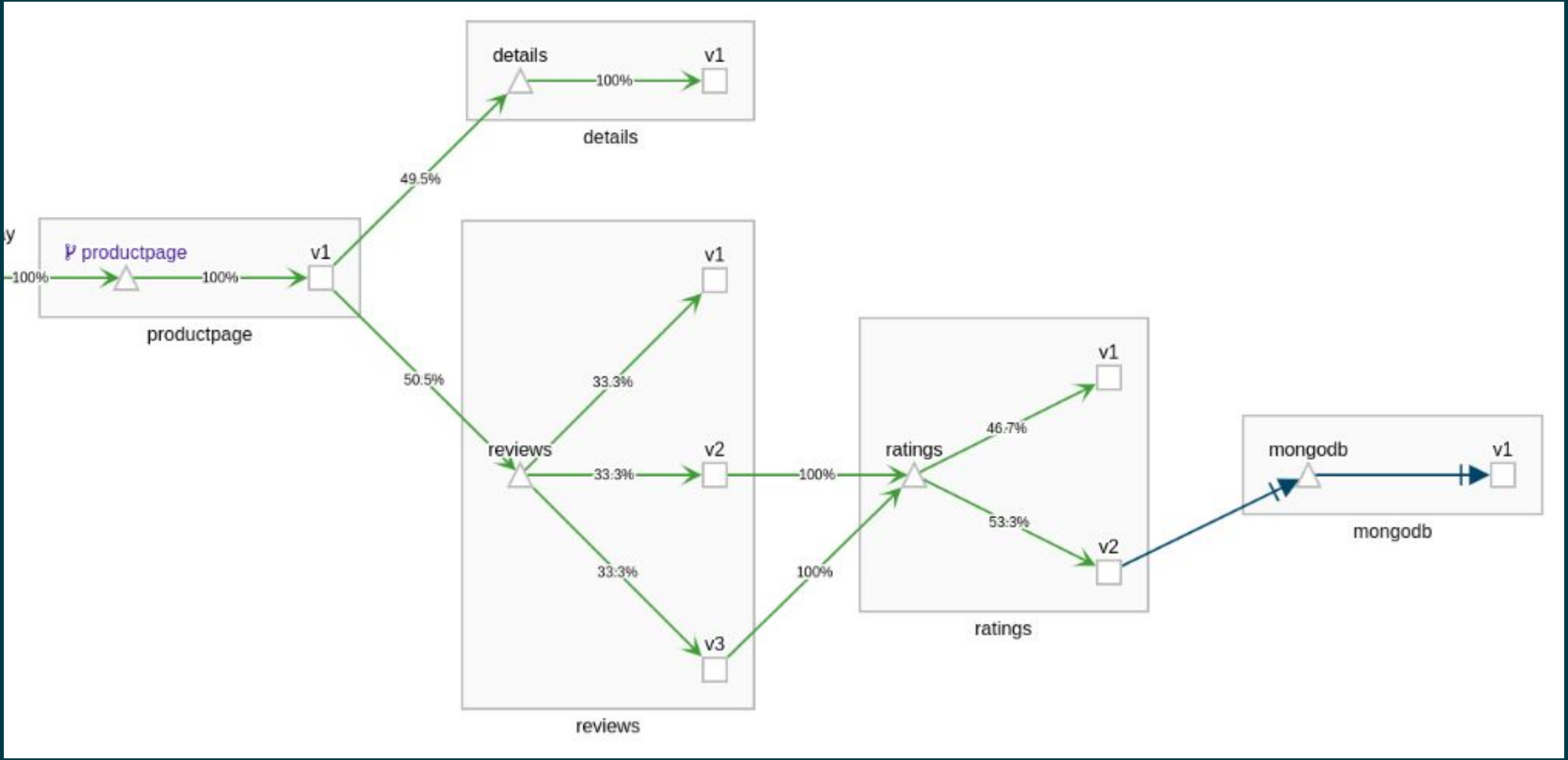
Let’s see Kiali in action

- Mesh visualization
- Fault Injection
- Configuration Validation
- Configure routing rules
- Tracing
- Traffic stats

Bookinfo example

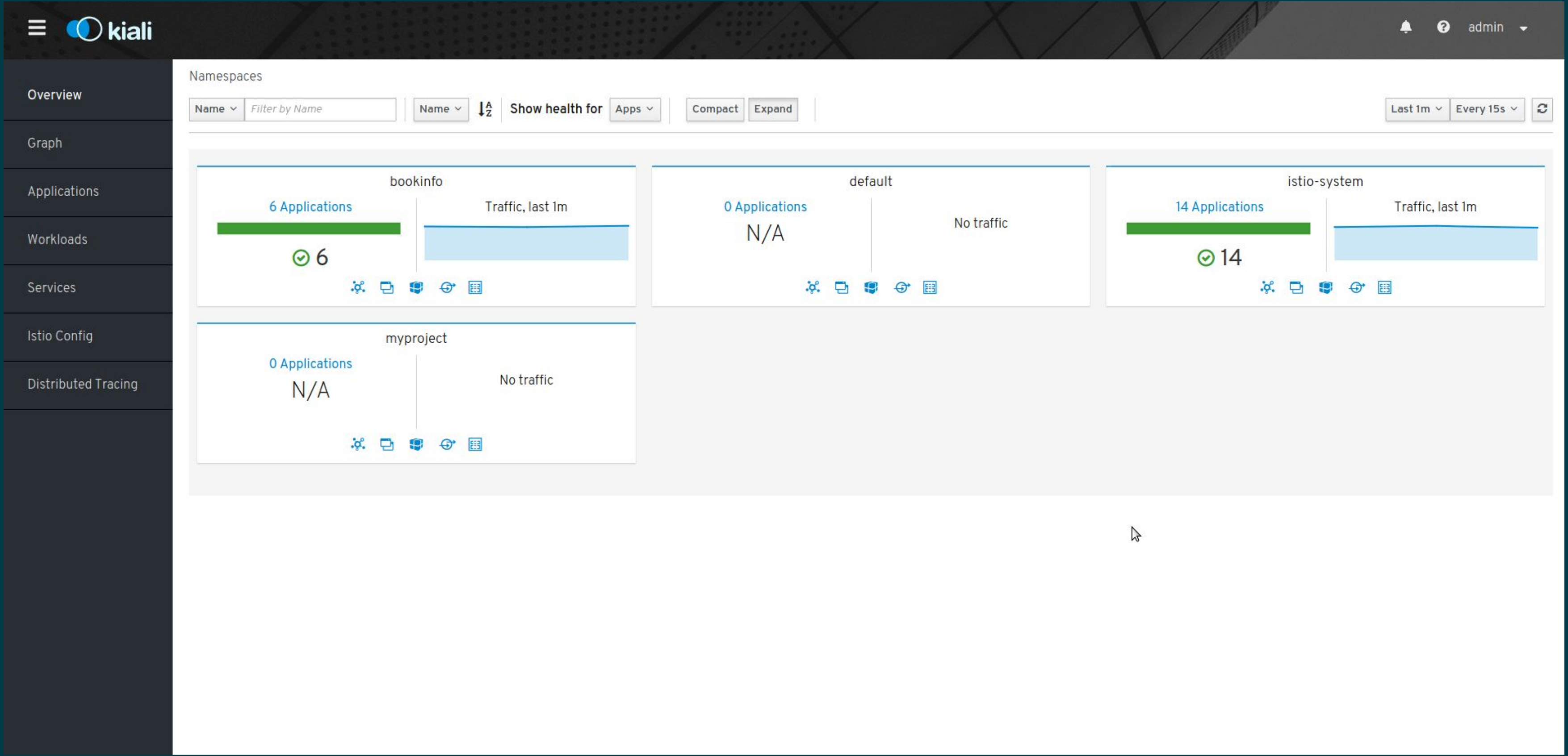


Bookinfo on Kiali



Kiali Features

Overview page



Namespaces

bookinfo

6 Applications

✓ 6

Traffic, last 1m

default

0 Applications

N/A

No traffic

istio-system

14 Applications

✓ 14

Traffic, last 1m

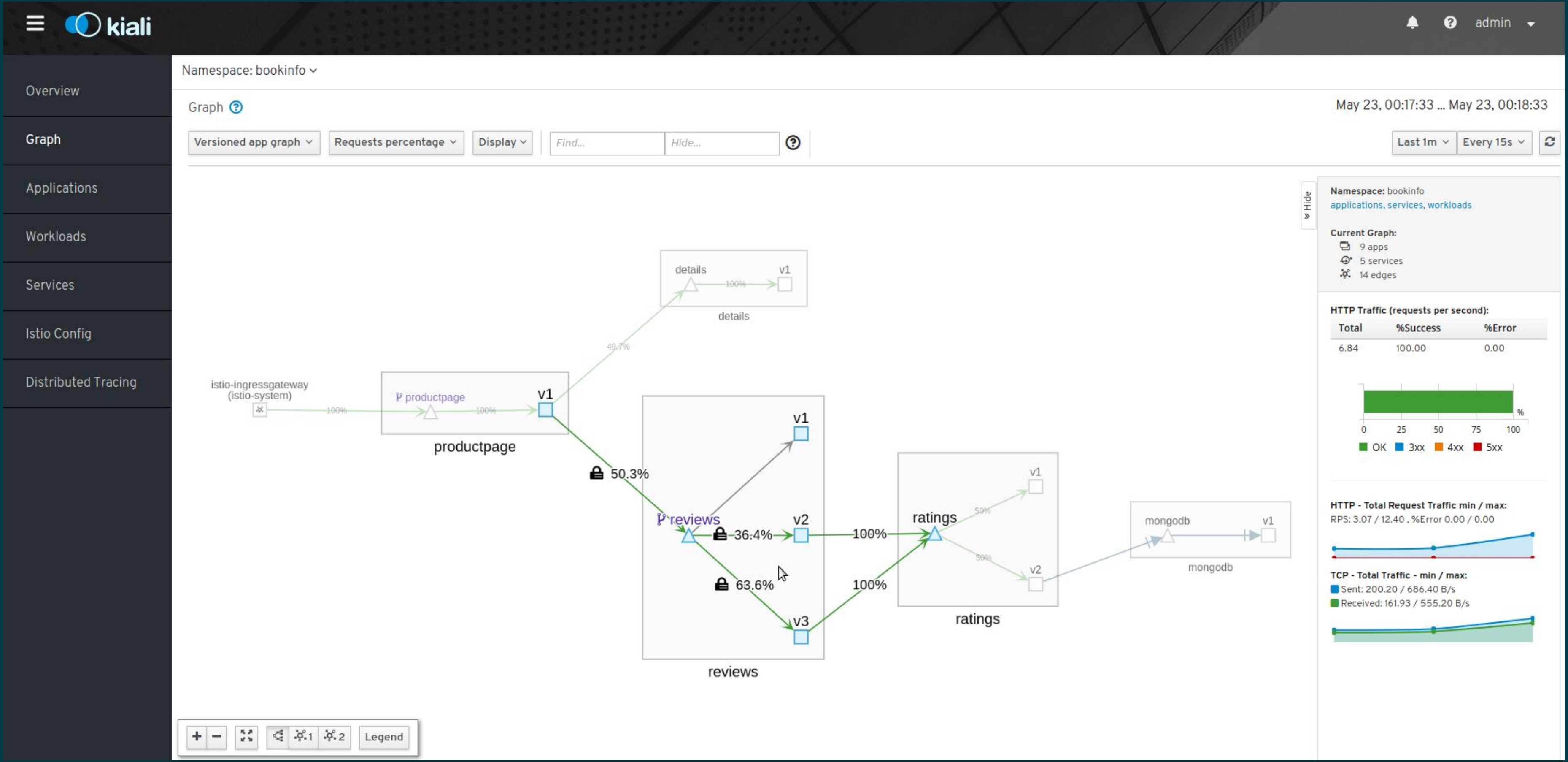
myproject

0 Applications

N/A

No traffic

Mesh Topology Graph



Namespace: bookinfo

May 23, 00:17:33 ... May 23, 00:18:33

Versioned app graph | Requests percentage | Display | Last 1m | Every 15s

Namespace: bookinfo
applications, services, workloads

Current Graph:
9 apps
5 services
14 edges

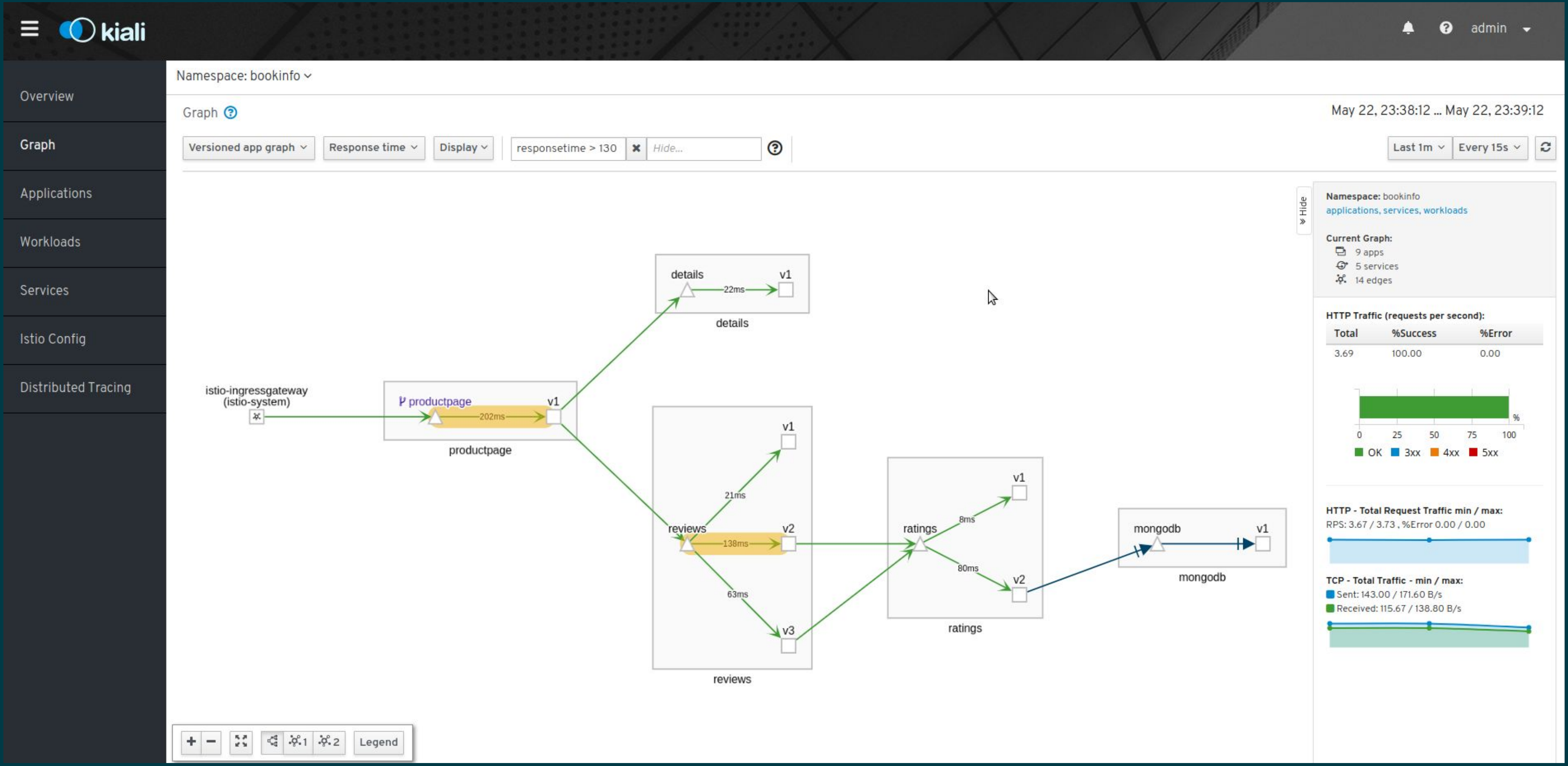
HTTP Traffic (requests per second):

Total	%Success	%Error
6.84	100.00	0.00

HTTP - Total Request Traffic min / max:
RPS: 3.07 / 12.40 , %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
🟦 Sent: 200.20 / 686.40 B/s
🟩 Received: 161.93 / 555.20 B/s

Hide and Seek



Namespace: bookinfo

Graph

May 22, 23:38:12 ... May 22, 23:39:12

Versioned app graph | Response time | Display | responsetime > 130 | Hide... | Last 1m | Every 15s

Namespace: bookinfo
applications, services, workloads

Current Graph:
9 apps
5 services
14 edges

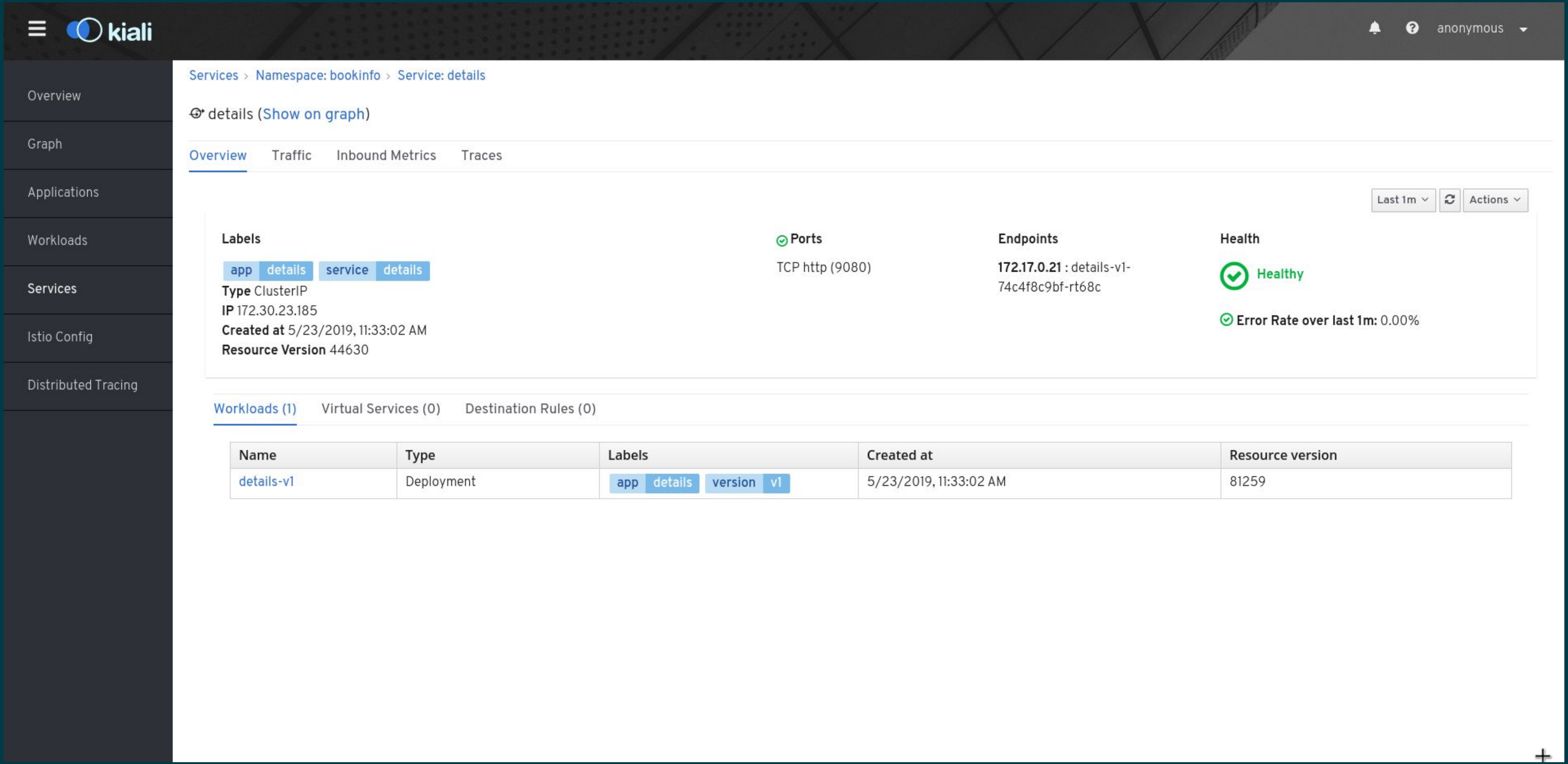
HTTP Traffic (requests per second):

Total	%Success	%Error
3.69	100.00	0.00

HTTP - Total Request Traffic min / max:
RPS: 3.67 / 3.73 , %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
🟦 Sent: 143.00 / 171.60 B/s
🟩 Received: 115.67 / 138.80 B/s

Details Page



Labels

app details service details

Type ClusterIP

IP 172.30.23.185

Created at 5/23/2019, 11:33:02 AM

Resource Version 44630

Ports

TCP http (9080)

Endpoints

172.17.0.21 : details-v1-74c4f8c9bf-rt68c

Health

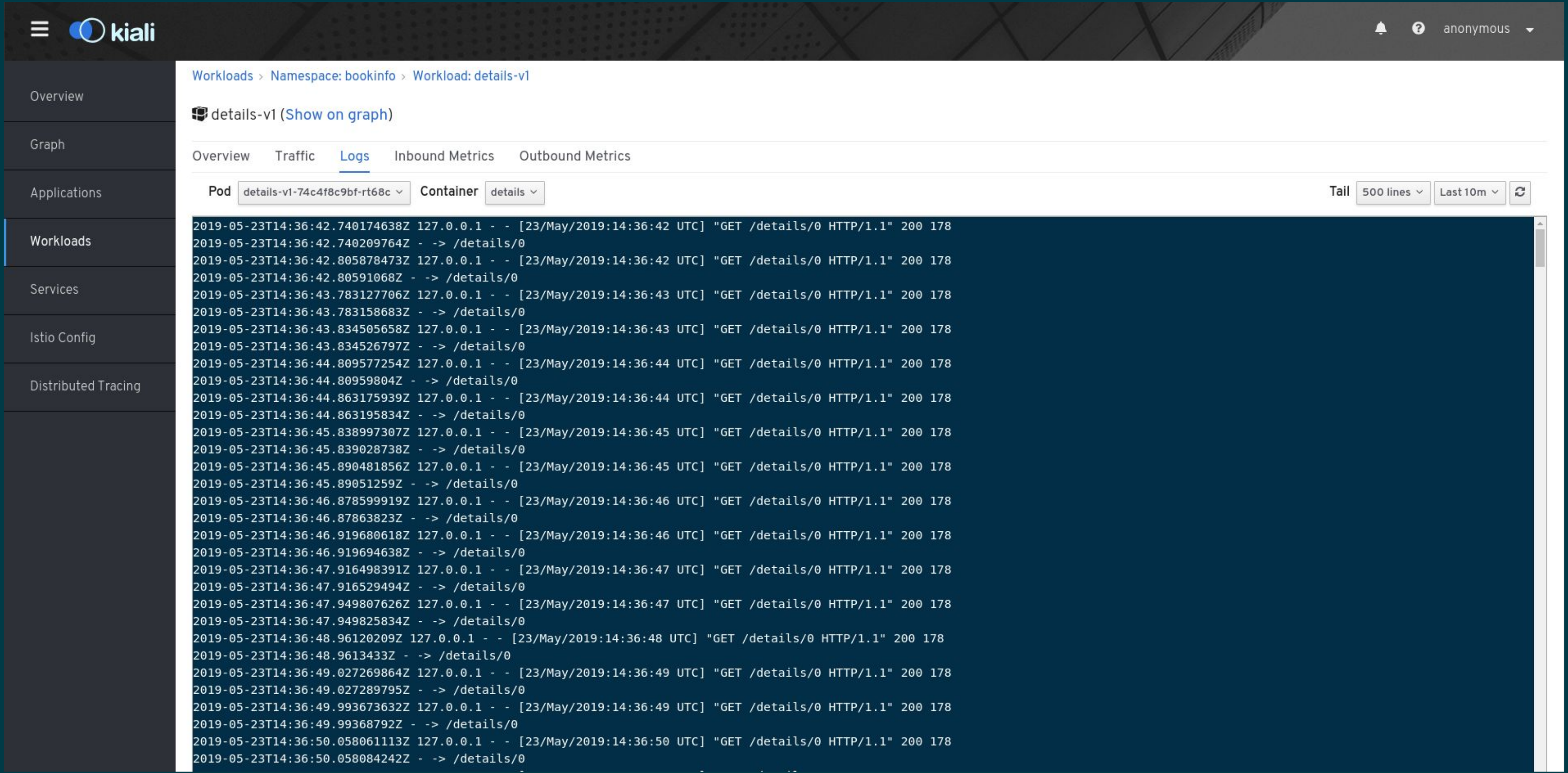
Healthy

Error Rate over last 1m: 0.00%

Workloads (1) Virtual Services (0) Destination Rules (0)

Name	Type	Labels	Created at	Resource version
details-v1	Deployment	app details version v1	5/23/2019, 11:33:02 AM	81259

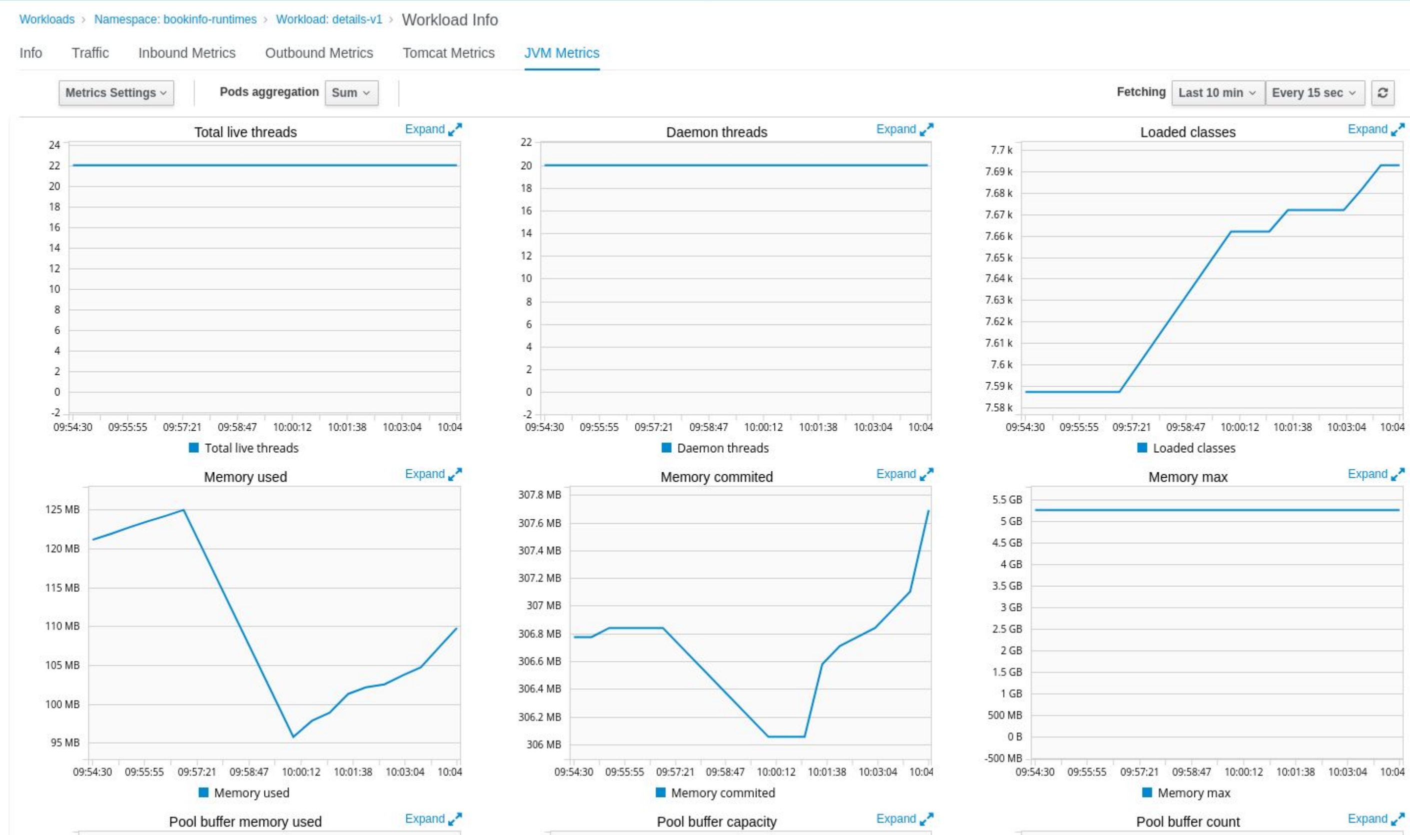
Viewing Logs



Pod: details-v1-74c4f8c9bf-rt68c
Container: details
Tail: 500 lines
Last 10m

```
2019-05-23T14:36:42.740174638Z 127.0.0.1 - - [23/May/2019:14:36:42 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:42.740209764Z - -> /details/0
2019-05-23T14:36:42.805878473Z 127.0.0.1 - - [23/May/2019:14:36:42 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:42.80591068Z - -> /details/0
2019-05-23T14:36:43.783127706Z 127.0.0.1 - - [23/May/2019:14:36:43 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:43.783158683Z - -> /details/0
2019-05-23T14:36:43.834505658Z 127.0.0.1 - - [23/May/2019:14:36:43 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:43.834526797Z - -> /details/0
2019-05-23T14:36:44.809577254Z 127.0.0.1 - - [23/May/2019:14:36:44 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:44.80959804Z - -> /details/0
2019-05-23T14:36:44.863175939Z 127.0.0.1 - - [23/May/2019:14:36:44 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:44.863195834Z - -> /details/0
2019-05-23T14:36:45.838997307Z 127.0.0.1 - - [23/May/2019:14:36:45 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:45.839028738Z - -> /details/0
2019-05-23T14:36:45.890481856Z 127.0.0.1 - - [23/May/2019:14:36:45 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:45.89051259Z - -> /details/0
2019-05-23T14:36:46.878599919Z 127.0.0.1 - - [23/May/2019:14:36:46 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:46.87863823Z - -> /details/0
2019-05-23T14:36:46.919680618Z 127.0.0.1 - - [23/May/2019:14:36:46 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:46.919694638Z - -> /details/0
2019-05-23T14:36:47.916498391Z 127.0.0.1 - - [23/May/2019:14:36:47 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:47.916529494Z - -> /details/0
2019-05-23T14:36:47.949807626Z 127.0.0.1 - - [23/May/2019:14:36:47 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:47.949825834Z - -> /details/0
2019-05-23T14:36:48.96120209Z 127.0.0.1 - - [23/May/2019:14:36:48 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:48.9613433Z - -> /details/0
2019-05-23T14:36:49.027269864Z 127.0.0.1 - - [23/May/2019:14:36:49 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:49.027289795Z - -> /details/0
2019-05-23T14:36:49.993673632Z 127.0.0.1 - - [23/May/2019:14:36:49 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:49.99368792Z - -> /details/0
2019-05-23T14:36:50.058061113Z 127.0.0.1 - - [23/May/2019:14:36:50 UTC] "GET /details/0 HTTP/1.1" 200 178
2019-05-23T14:36:50.058084242Z - -> /details/0
```

Runtime metric dashboards

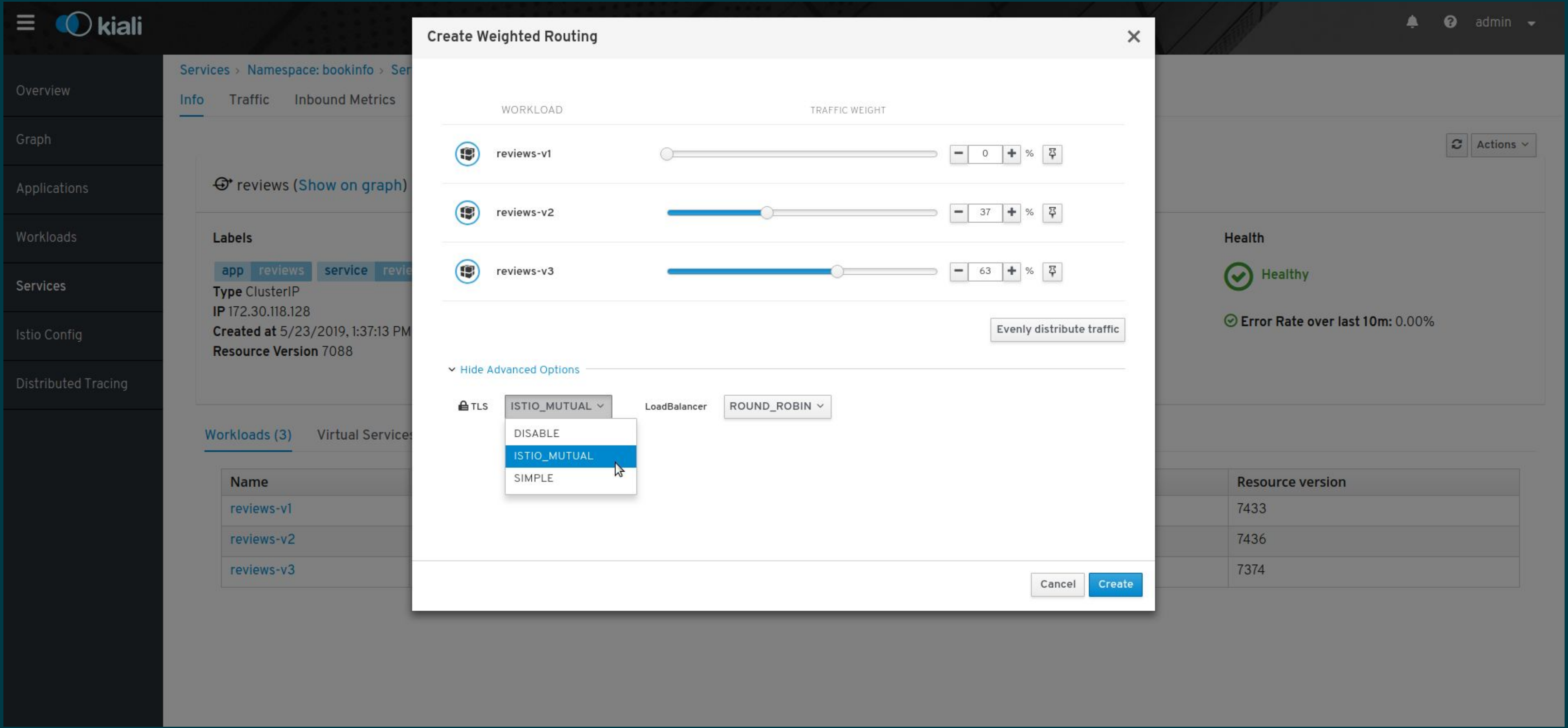


Workloads › Namespace: bookinfo-runtimes › Workload: details-v1 › Workload Info

Info | Traffic | Inbound Metrics | Outbound Metrics | Tomcat Metrics | JVM Metrics

Metrics Settings ⌄ | Pods aggregation: Sum ⌄ | Fetching: Last 10 min ⌄ | Every 15 sec ⌄

Weighted Routing



reviews (Show on graph)

Labels

app reviews service revie

Type ClusterIP
IP 172.30.118.128
Created at 5/23/2019, 1:37:13 PM
Resource Version 7088

Health

Healthy

Error Rate over last 10m: 0.00%

Workloads (3) Virtual Services

Name	Resource version
reviews-v1	7433
reviews-v2	7436
reviews-v3	7374

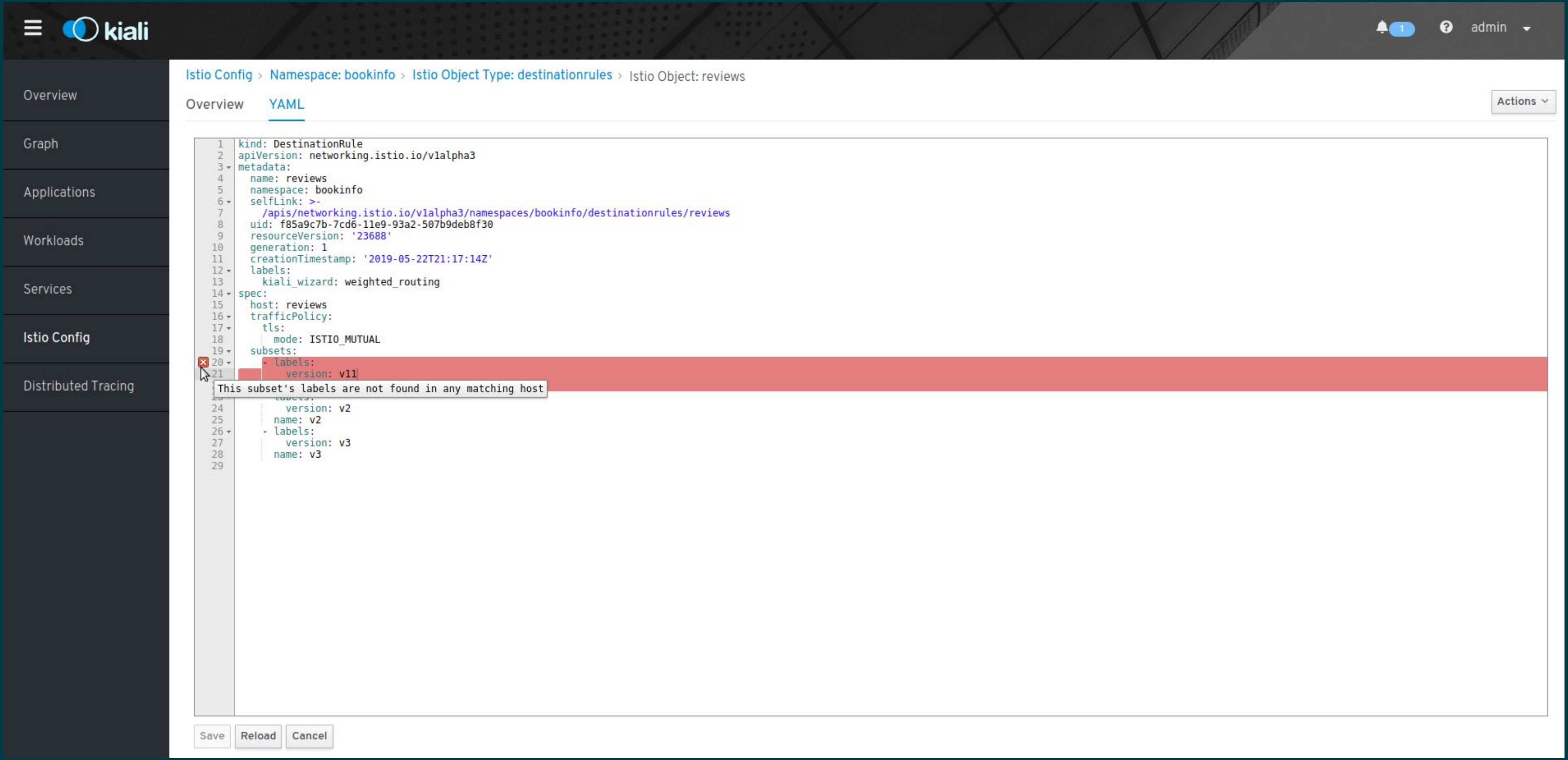
Create Weighted Routing

WORKLOAD	TRAFFIC WEIGHT
reviews-v1	0 %
reviews-v2	37 %
reviews-v3	63 %

TLS ISTIO_MUTUAL

LoadBalancer ROUND_ROBIN

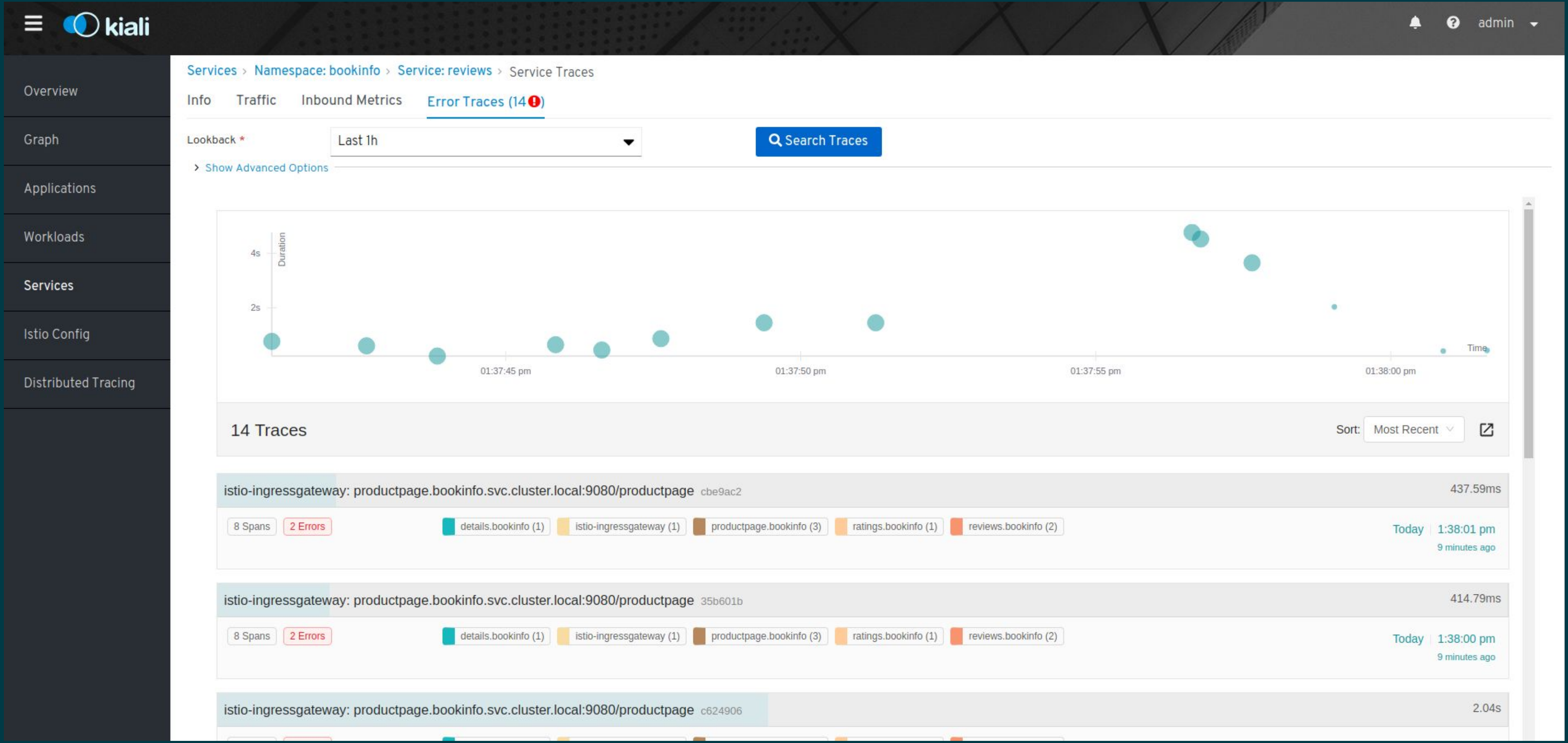
Configuration validations



```yaml
kind: DestinationRule
apiVersion: networking.istio.io/v1alpha3
metadata:
  name: reviews
  namespace: bookinfo
  selfLink: >-
    /apis/networking.istio.io/v1alpha3/namespaces/bookinfo/destinationrules/reviews
  uid: f85a9c7b-7cd6-11e9-93a2-507b9deb8f30
  resourceVersion: '23688'
  generation: 1
  creationTimestamp: '2019-05-22T21:17:14Z'
  labels:
    kiali_wizard: weighted_routing
spec:
  host: reviews
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
  subsets:
    - labels:
        version: v11
      labels:
        version: v2
      name: v2
    - labels:
        version: v3
      name: v3
```

```
This subset's labels are not found in any matching host
```

Tracing (integration with Jaeger)



Lookback * Last 1h

14 Traces

Sort: Most Recent

istio-ingressgateway: productpage.bookinfo.svc.cluster.local:9080/productpage cbe9ac2 437.59ms

8 Spans | 2 Errors | details.bookinfo (1) | istio-ingressgateway (1) | productpage.bookinfo (3) | ratings.bookinfo (1) | reviews.bookinfo (2)

Today | 1:38:01 pm
9 minutes ago

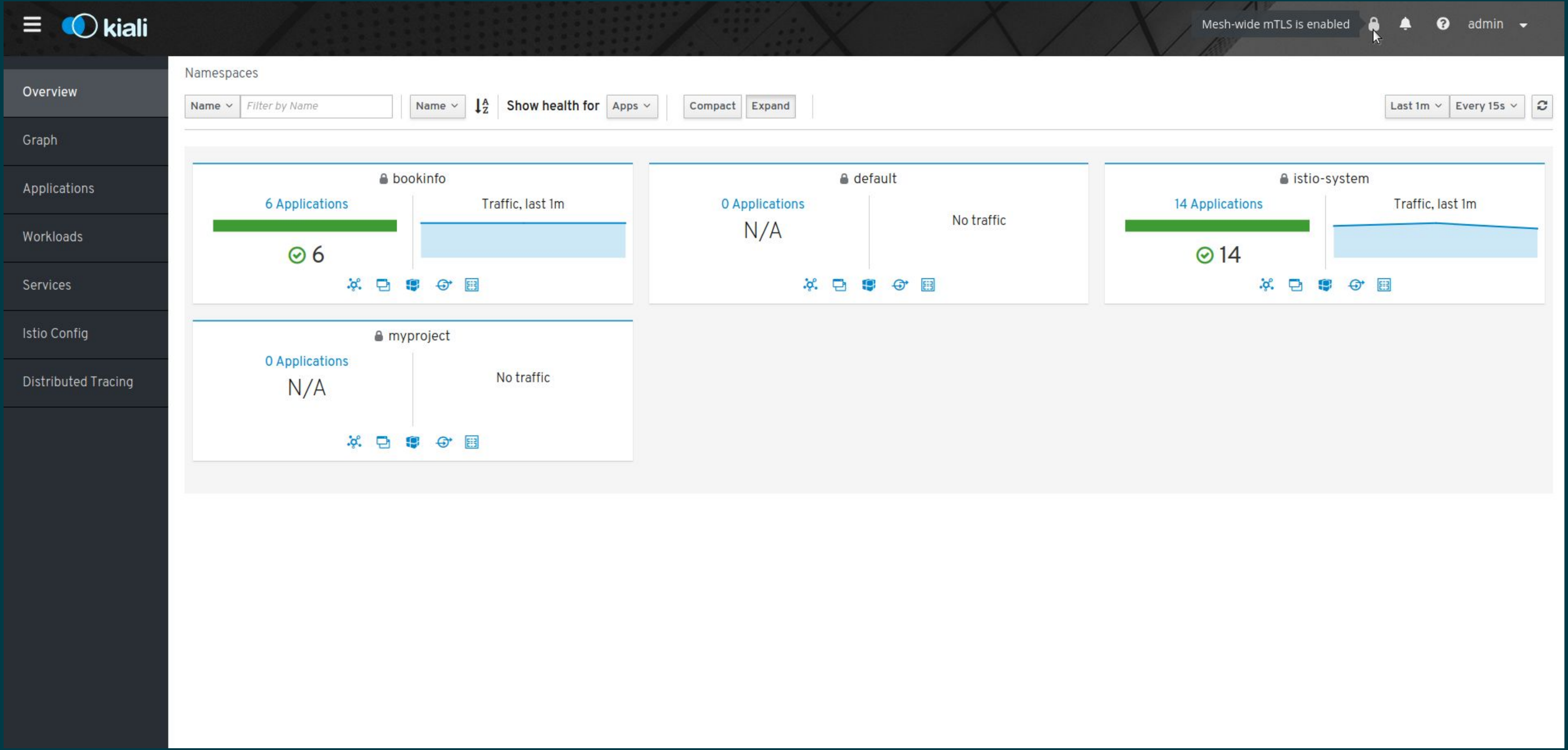
istio-ingressgateway: productpage.bookinfo.svc.cluster.local:9080/productpage 35b601b 414.79ms

8 Spans | 2 Errors | details.bookinfo (1) | istio-ingressgateway (1) | productpage.bookinfo (3) | ratings.bookinfo (1) | reviews.bookinfo (2)

Today | 1:38:00 pm
9 minutes ago

istio-ingressgateway: productpage.bookinfo.svc.cluster.local:9080/productpage c624906 2.04s

Visualizing security



Connect with the community

Kiali.io

Istio.io

KialiProject

IstioMesh

github.com/kiali

github.com/istio

Icon credits

- Twitter by Lubos Volkov, the Noun Project
- Light Bulb by artworkbean, the Noun Project
- Magnifying Glass by Musket from the Noun Project
- Questions by Rediffusion from the Noun Project
- Mug by Alex Getty from the Noun Project
- Diamond by MarkieAnn Packer from the Noun Project
- Box by Cornelius Danger from the Noun Project

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
JAPAN

Thank you!

mikeyteva

Introduction to service mesh with Istio and Kiali

Alissa Bonas

mikeyteva