《Service Mesh 实战》

扫码试看/订阅

《Service Mesh 实战》视频课程

3.12 使用 Kiali 观测你的微服务应用

微服务架构可视化的重要性

- 痛点
  - 服务间依赖关系错综复杂
  - 问题排查困难，扯皮甩锅时有发生
- 优势
  - 梳理服务的交互关系
  - 了解应用的行为与状态

什么是 Kiali

- 官方定义：
  - Istio 的可观察性控制台
  - 通过服务拓扑帮助你理解服务网格的结构
  - 提供网格的健康状态视图
  - 具有服务网格配置功能
- 名字含义
- Istio 作为宿主

Kiali 的功能

Topology

See your services communicate

Health

Quickly identify issues

Metrics

Chart Istio and App performance

Configuration

See and edit Istio custom resource YAML

Tracing

Follow requests with Jaeger Distributed Tracing

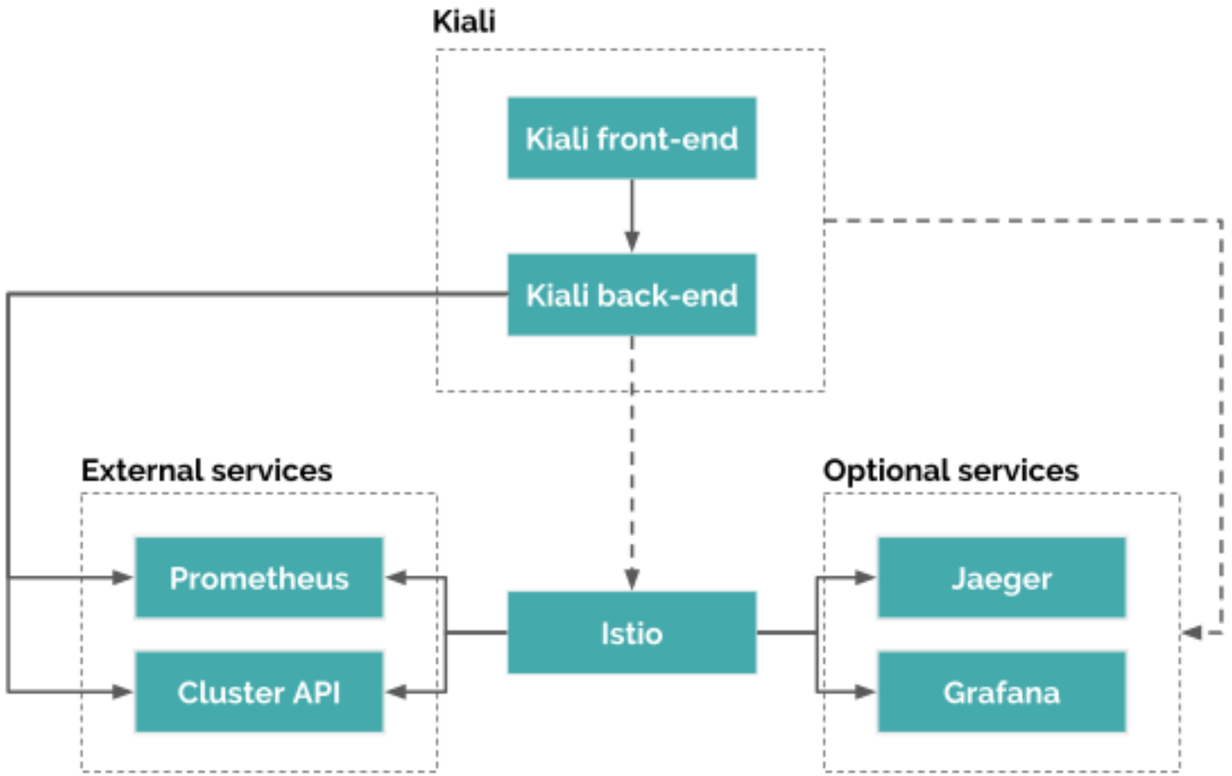
Validations

Detect advanced misconfigurations

Wizards

Easily configure Istio routing

Kiali 架构


Kiali
Kiali front-end
Kiali back-end
External services
Prometheus
Cluster API
Istio
Optional services
Jaeger
Grafana

演示

- 基本页面
  - 总体拓扑
  - 服务信息
  - 工作负载信息
- 配置验证
- 向导功能

课后练习

- 使用 Kiali 查看你的 Istio 网格

3.13 使用 Prometheus 收集指标

Prometheus 简介

Dimensional data

Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

Powerful queries

PromQL allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

Efficient storage

Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

Simple operation

Each server is independent for reliability, relying only on local storage. Written in Go, all binaries are statically linked and easy to deploy.

Precise alerting

Alerts are defined based on Prometheus's flexible PromQL and maintain dimensional information. An alertmanager handles notifications and silencing.

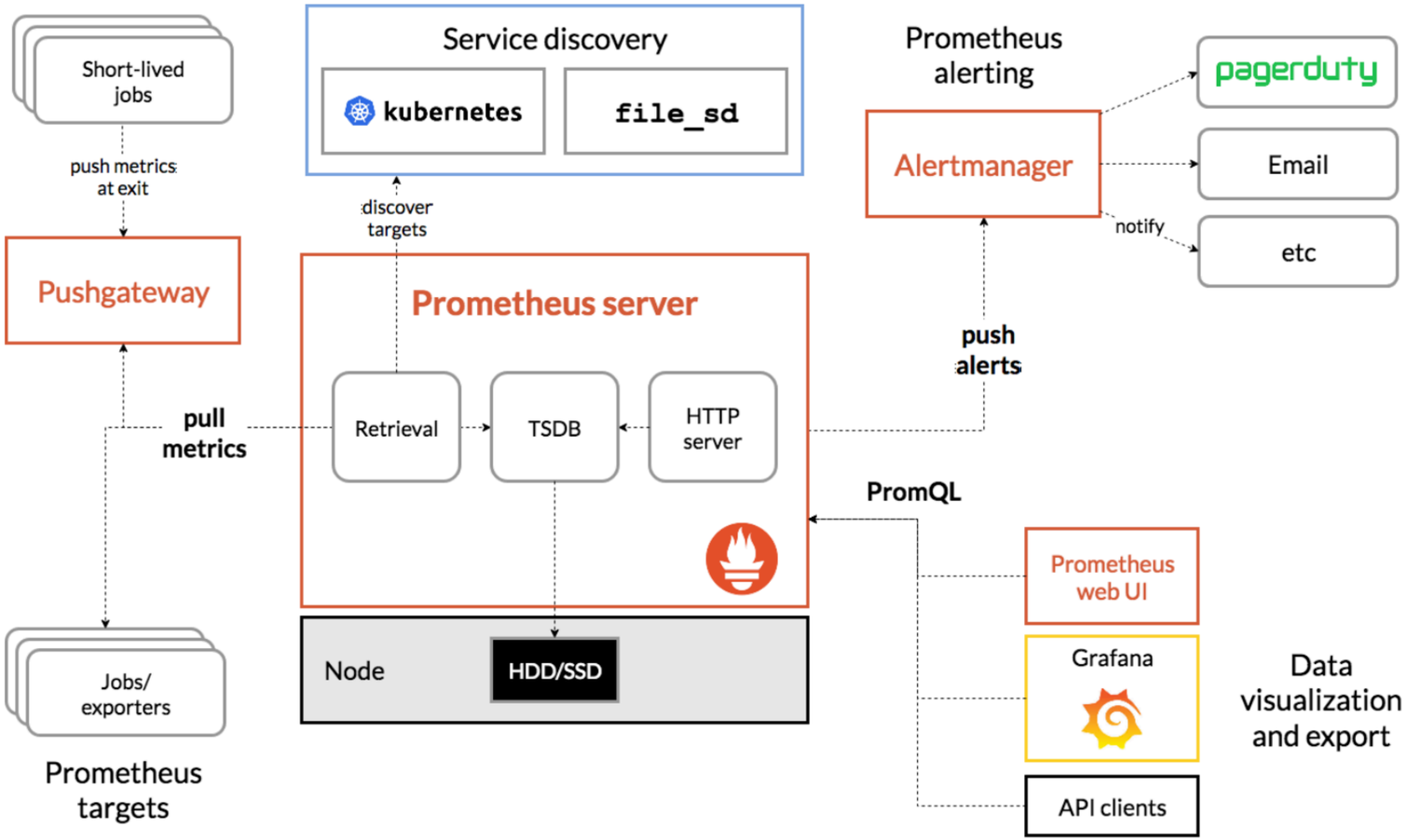
Many client libraries

Client libraries allow easy instrumentation of services. Over ten languages are supported already and custom libraries are easy to implement.

Many integrations

Existing exporters allow bridging of third-party data into Prometheus. Examples: system statistics, as well as Docker, HAProxy, StatsD, and JMX metrics.

Prometheus 架构



任务：收集指标

- 任务说明
  - 通过 Prometheus 收集指标并查看指标数据
- 任务目标
  - 学会用 Prometheus 查看指标数据
  - 了解 Istio 1.5 中新的遥测方法 （Telemetry V2）

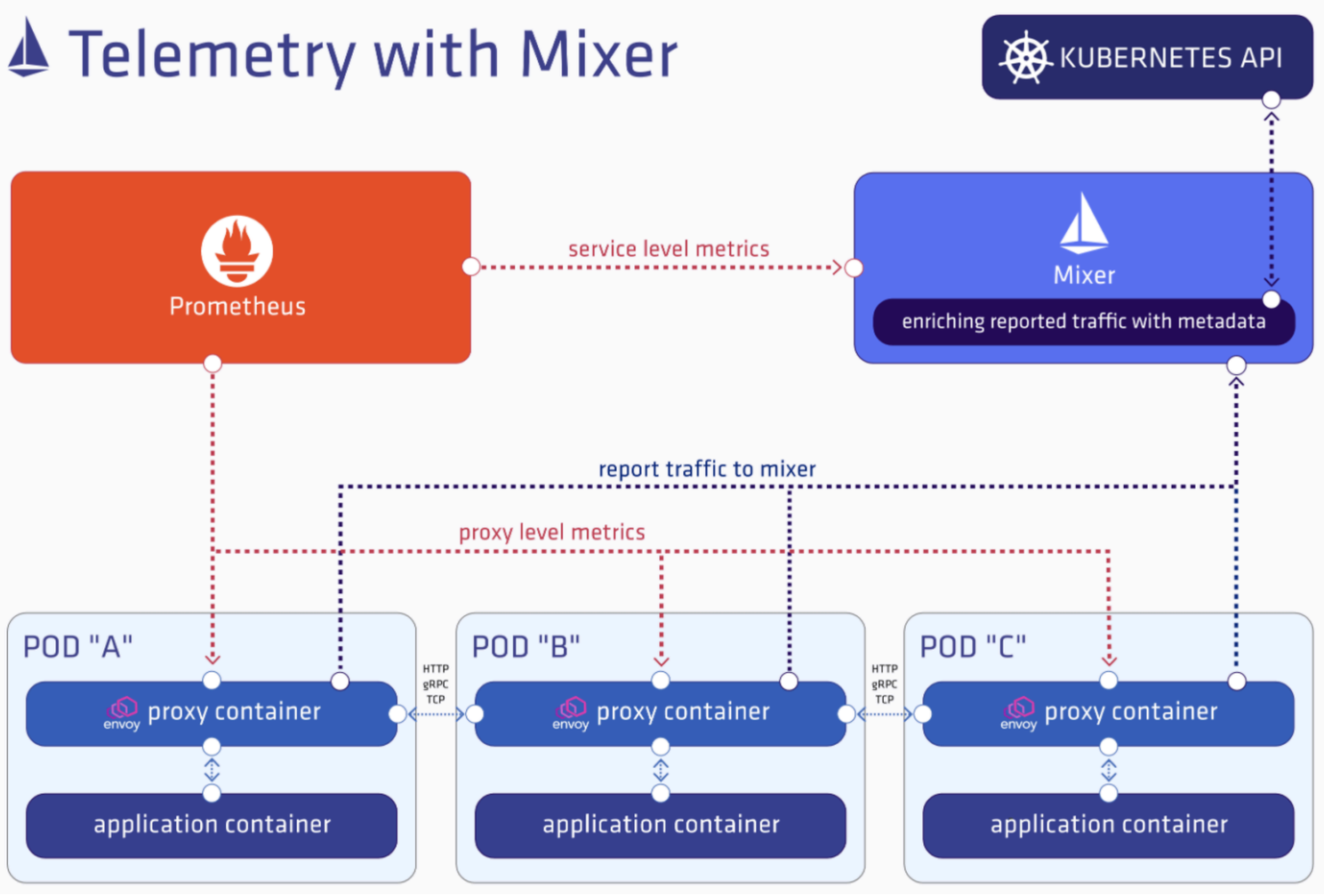
演示

- 查看服务指标
- 查看 Envoy 代理指标
- 查看控制平面指标

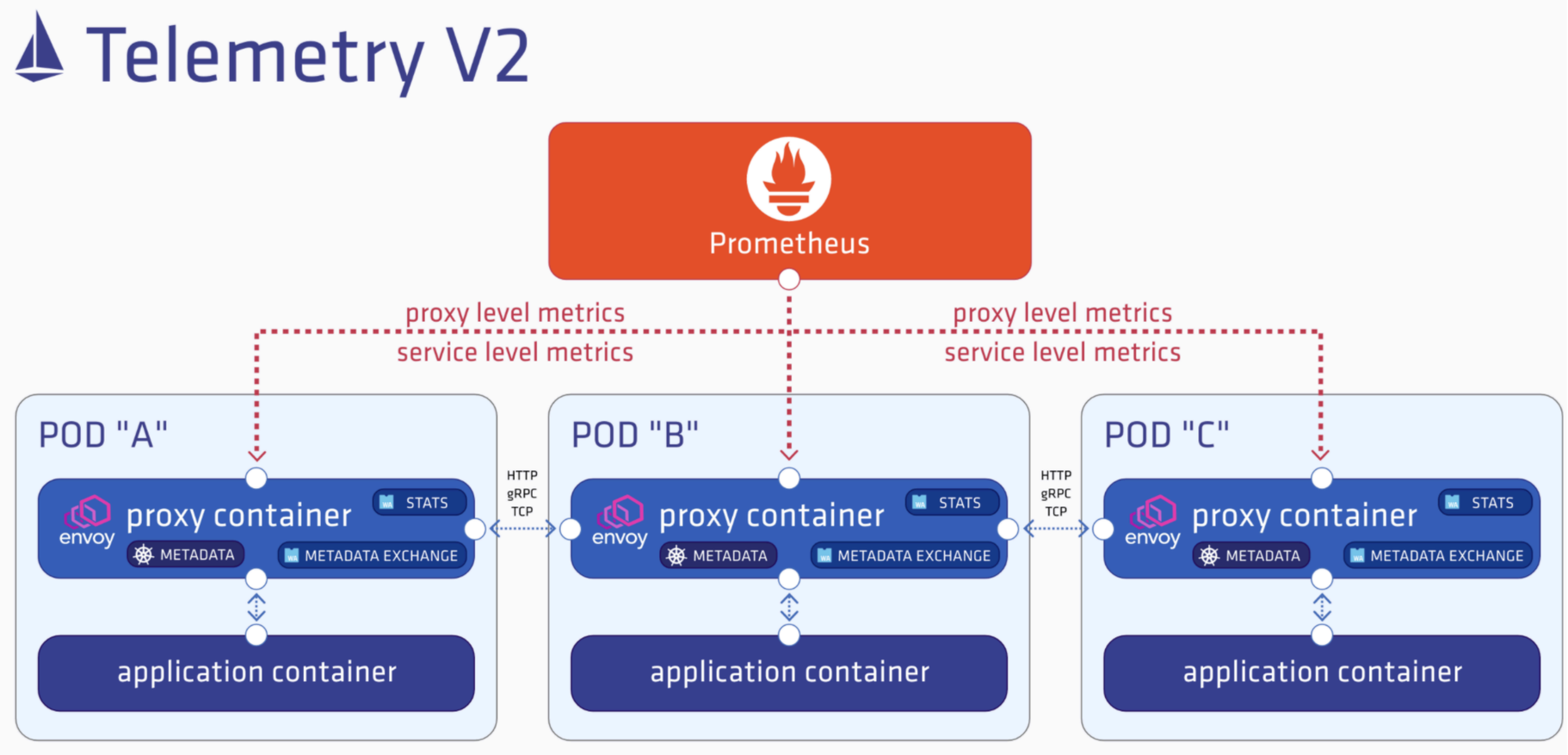
Istio 1.5 的遥测指标

- 请求数（istio_requests_total）
- 请求时长（istio_request_duration_milliseconds）
- 请求大小（istio_request_bytes）
- 响应大小（istio_response_bytes）
- TCP 发送字节数（istio_tcp_sent_bytes_total）
- TCP 接受字节数（istio_tcp_received_bytes_total）
- TCP 连接打开数（istio_tcp_connections_opened_total）
- TCP 连接关闭数（istio_tcp_connections_closed_total）

Istio 1.5 遥测的变化

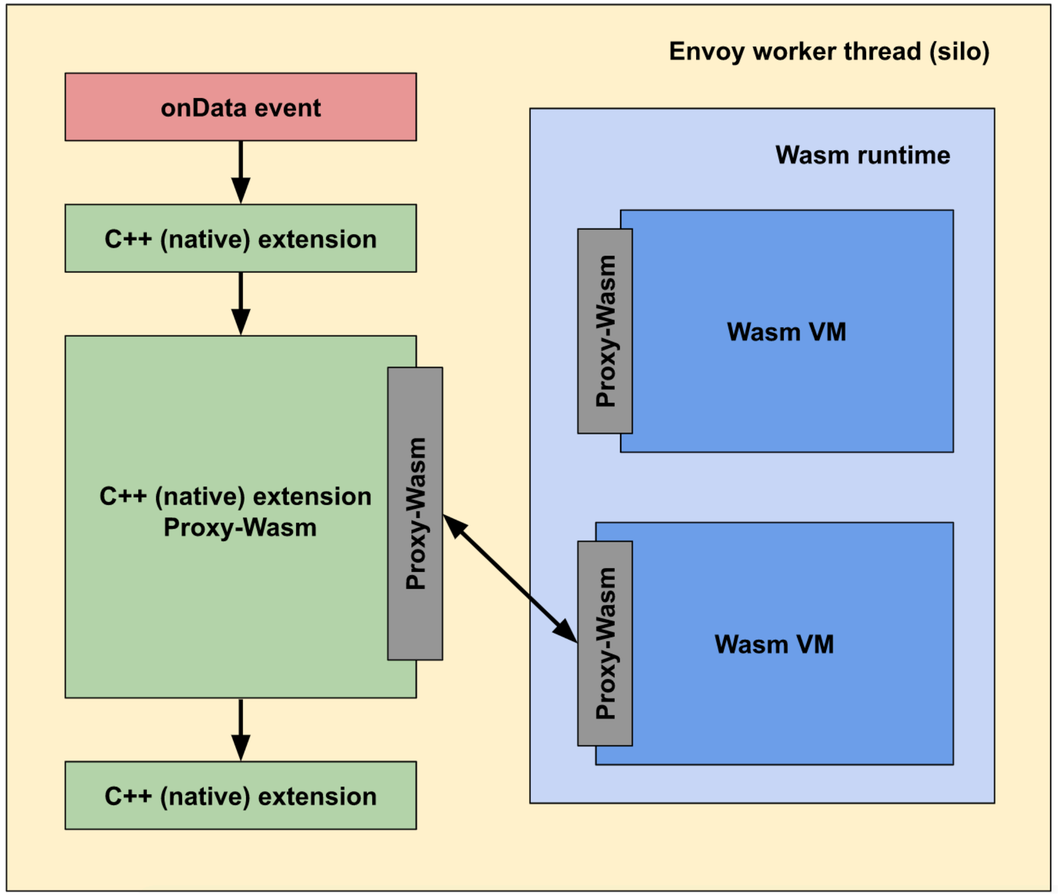


Istio 1.5 遥测的变化



WebAssembly in Envoy

- 解决静态化编译（构建时）的弊端
- 优势：
  - 无需修改 Envoy
  - 避免远程调用
  - 隔离性/安全/多样性
  - 可移植/可维护



课后练习

- 通过访问服务生成指标，并查看数据

3.14 使用 Grafana 查看系统

Grafana 简介

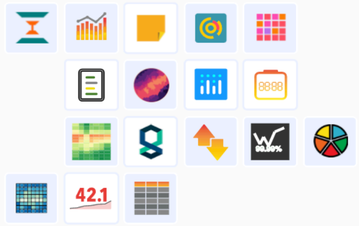
Visualize

From heatmaps to histograms. Graphs to geomaps. Grafana has a plethora of visualization options to help you understand your data, beautifully. Read more.

Alert

Seamlessly define alerts where it makes sense — while you're in the data. Define thresholds visually, and get notified via Slack, PagerDuty, and more. Read more.

Extend

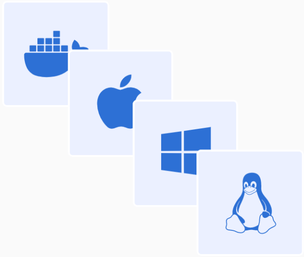


Discover hundreds of dashboards and plugins in the official library. Thanks to the passion and momentum of our community, new ones are added every week.

Unify

Bring your data together to get better context. Grafana supports dozens of databases, natively. Mix them together in the same Dashboard. Read more.

Open



Grafana gives you options. It's completely open source, and backed by a vibrant community. Use Hosted Grafana, or easily install on any platform.

Collaborate

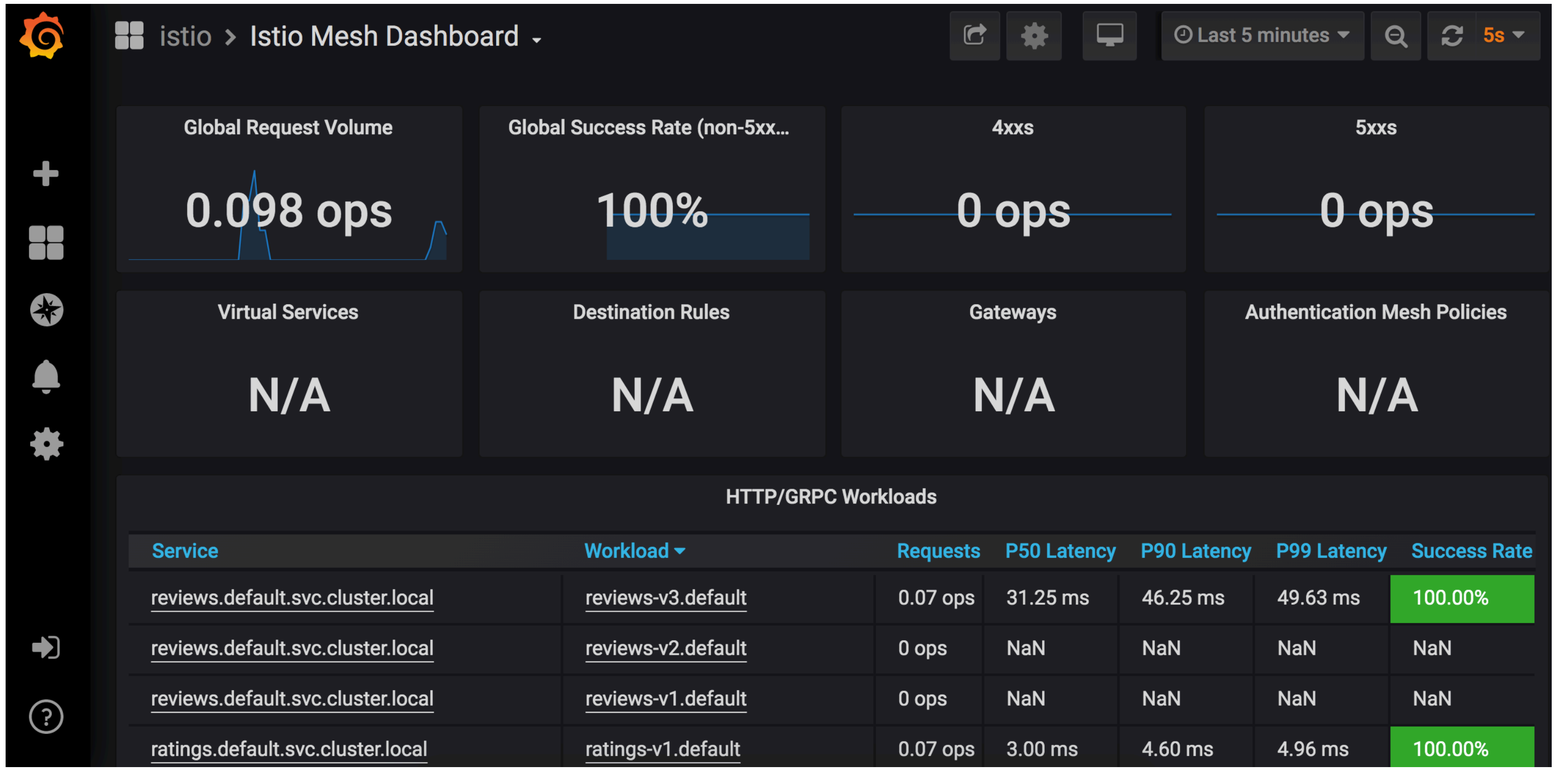
Bring everyone together, and share data and dashboards across teams. Grafana empowers users, and helps foster a data driven culture.

Istio Dashboard

- Mesh Dashboard：查看应用（服务）数据
  - 网格数据总览
  - 服务视图
  - 工作负载视图
- Performance Dashboard：查看 Istio 自身（各组件）数据
  - Istio 系统总览
  - 各组件负载情况

任务：指标可视化

- 使用 Grafana 查看指标生成的 Istio Dashboard



演示

- 确保集群中已安装 Grafana
  - --set values.grafana.enabled=true
- 启动 Grafana
- 查看 Dashboard
- 通过 bookinfo 应用产生指标数据

课后练习

- 使用 Grafana 查看 Istio 的指标数据

3.15 如何获取 Envoy 的日志并进行调试

任务：查看 Envoy 日志

- 通过查看 Envoy 日志了解流量信息
- 学会通过分析日志调试流量异常

演示

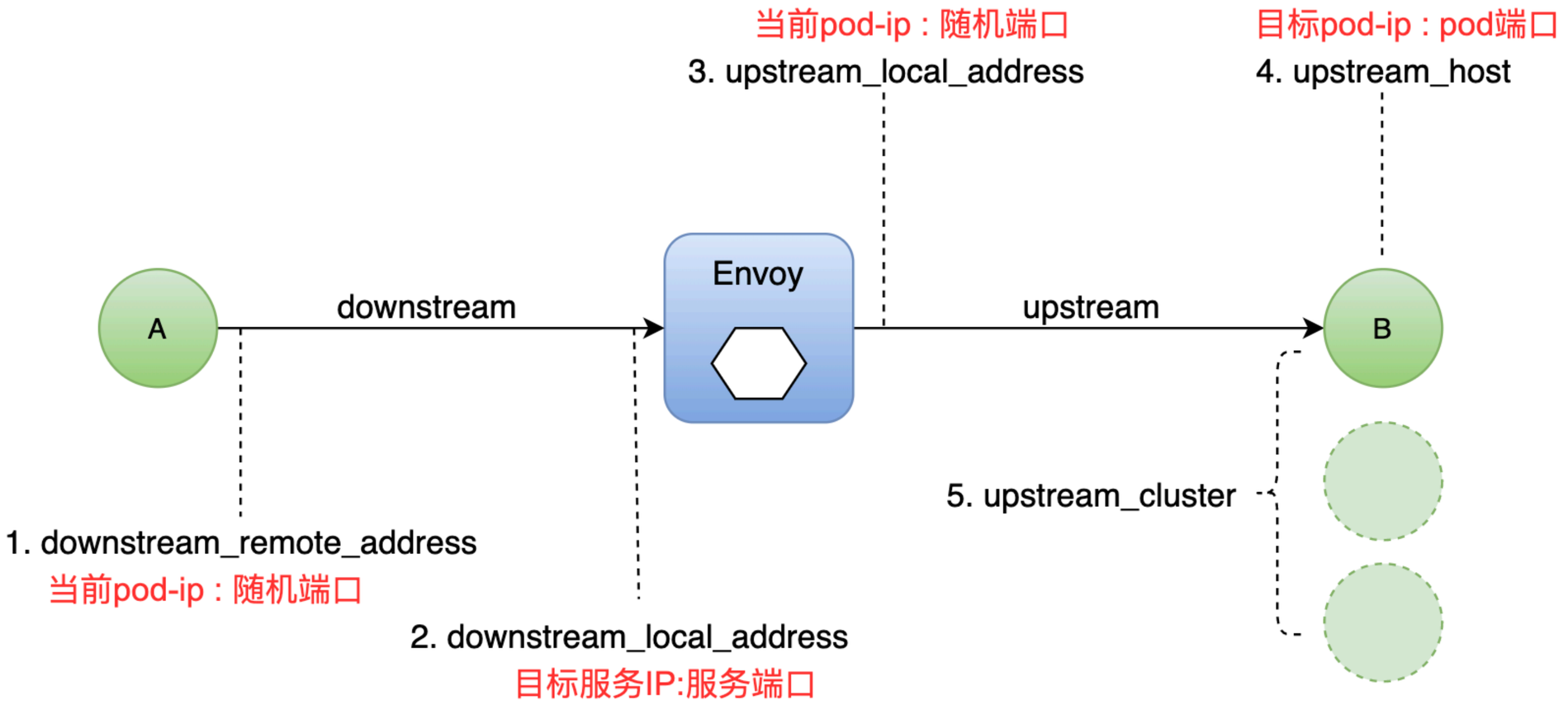
- 确认 Envoy 日志配置开启
  - --set values.global.proxy.accessLogFile="/dev/stdout"
- 访问服务产生流量输出
- 查看 Envoy (istio-proxy) 日志输出

日志项分析

JSON

- **bytes_sent :** 178
- **upstream_cluster :** outbound|9080||details.default.svc.cluster.local
- **downstream_remote_address :** 10.1.0.18:53912
- **authority :** details:9080
- **path :** /details/0
- **protocol :** HTTP/1.1
- **upstream_service_time :** 6
- **upstream_local_address :** 10.1.0.18:44928
- **duration :** 6
- **upstream_transport_failure_reason :** -
- **route_name :** default
- **downstream_local_address :** 10.106.253.3:9080
- **user_agent :** Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.113 Safari/537.36
- **response_code :** 200
- **response_flags :** -
- **start_time :** 2020-04-25T13:49:00.523Z
- **method :** GET
- **request_id :** 02f7bfbb-0351-972c-a2ff-4bf1d21ec091
- **upstream_host :** 10.1.0.13:9080
- **x_forwarded_for :** -
- **requested_server_name :** -
- **bytes_received :** 0
- **istio_policy_status :** -

Envoy 流量五元组


当前pod-ip：随机端口
3. upstream_local_address
目标pod-ip：pod端口
4. upstream_host
Envoy
A
downstream
upstream
B
5. upstream_cluster
1. downstream_remote_address
当前pod-ip：随机端口
2. downstream_local_address
目标服务IP:服务端口

调试关键字段：RESPONSE_FLAGS

- UH：upstream cluster 中没有健康的 host，503
- UF：upstream 连接失败，503
- UO：upstream overflow（熔断）
- NR：没有路由配置，404
- URX：请求被拒绝因为限流或最大连接次数
- …… ……

Envoy 日志配置项

配置项	说明
global.proxy.accessLogFile	日志输出文件，空为关闭输出
global.proxy.accessLogEncoding	日志编码格式：JSON、TEXT
global.proxy.accessLogFormat	配置显示在日志中的字段，空为默认格式
global.proxy.logLevel	日志级别，空为 warning，可选 trace|debug|info|warning|error|critical|off

课后练习

- 尝试修改日志项，并进行日志分析

扫码试看/订阅

《Service Mesh 实战》视频课程