ARAF KARSH HAMID
Co-Founder / CTO
MetaMagic Global Inc., NJ, USA

@arafkarsh

arafkarsh

Building Cloud Native Apps

Microservice Architecture Series

Service Mesh / Istio
Zipkin / Prometheus / Grafana / Kiali
Monitoring / Observability

1

Monitoring
Observability

2

Kubernetes
Auditing

Zipkin
Prometheus
Grafana / Kiali

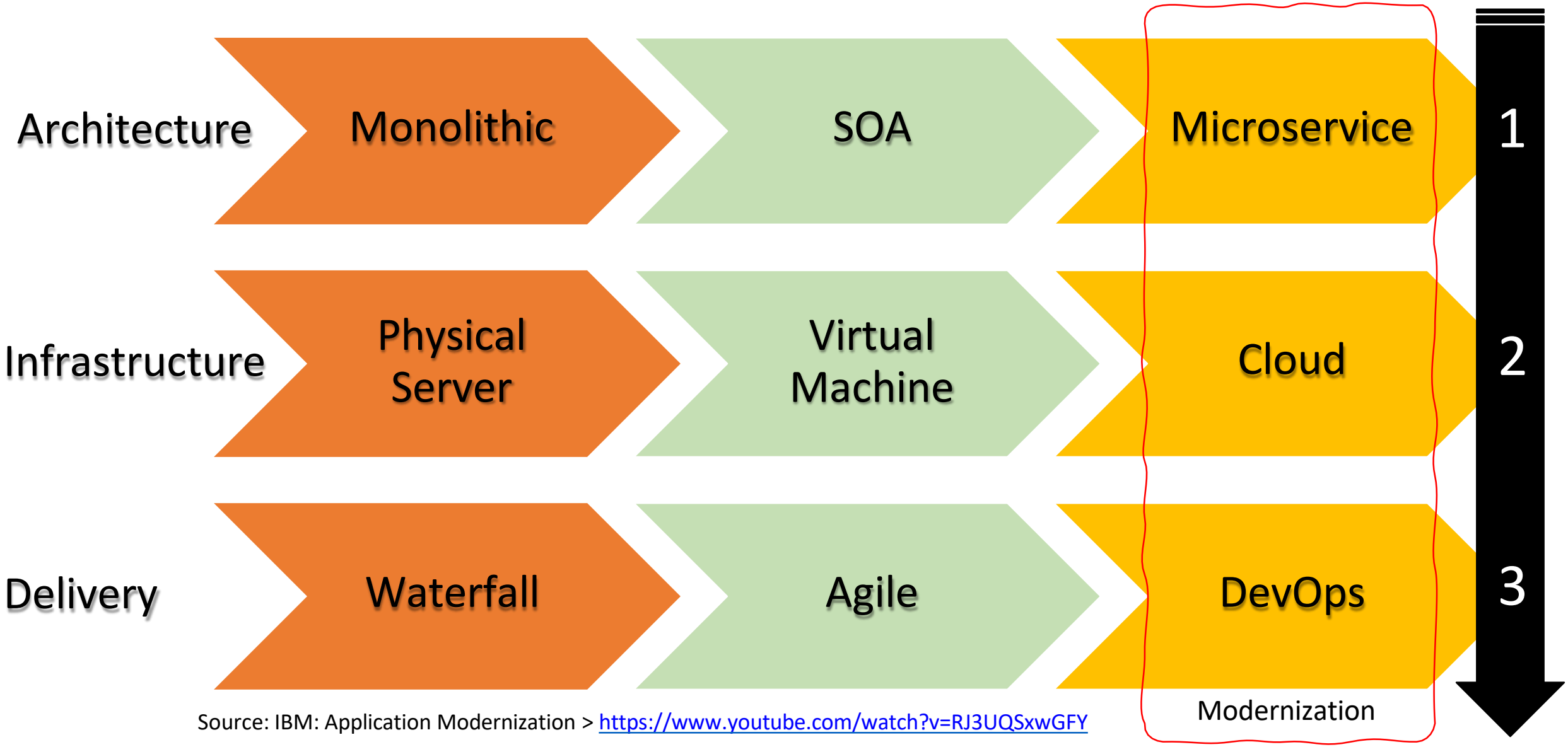
3

ML / AI

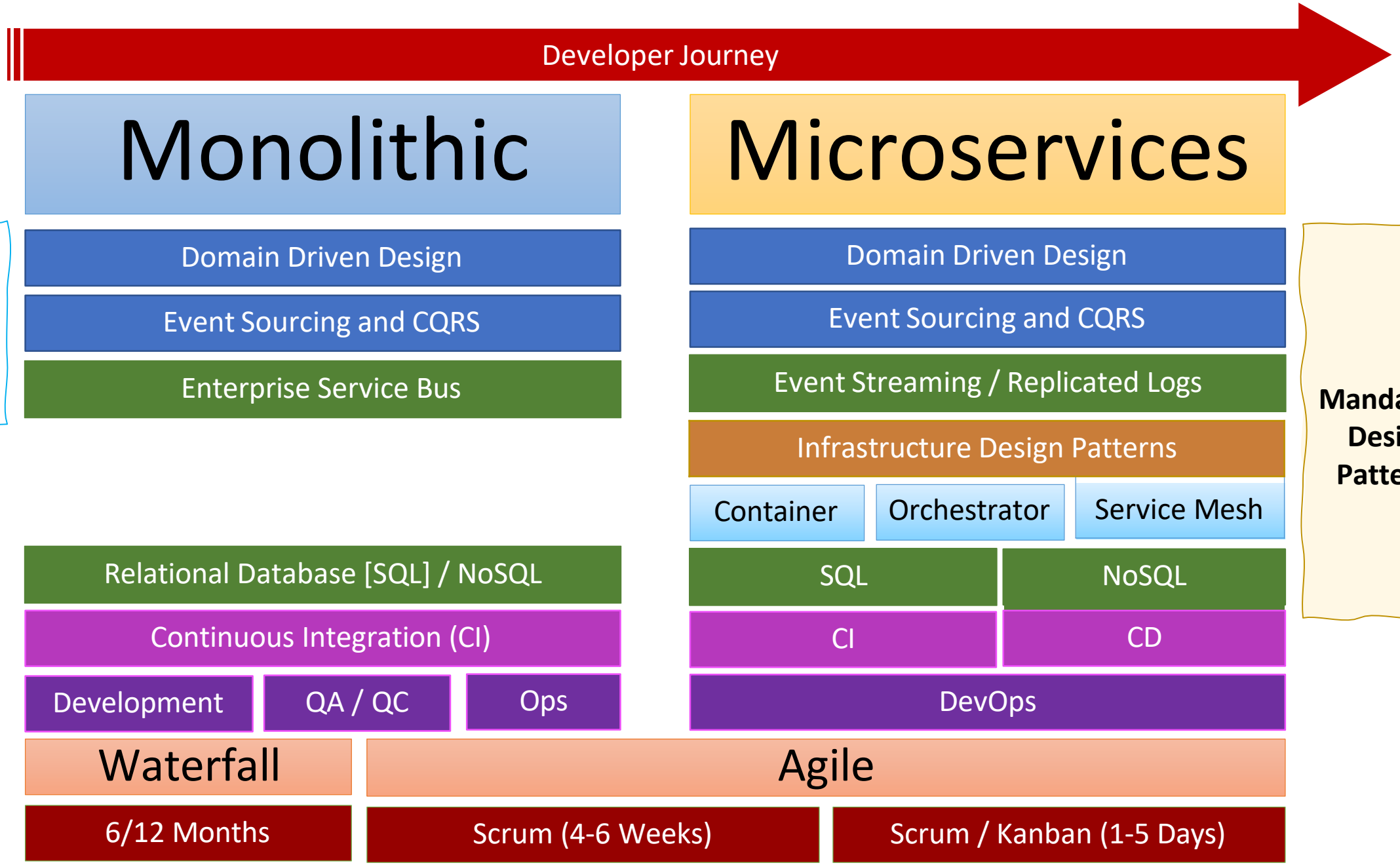
4

Slides are color coded based on the topic colors.

Application Modernization – 3 Transformations



Source: IBM: Application Modernization > https://www.youtube.com/watch?v=RJ3UQSxwGFY


Developer Journey
Monolithic
Microservices
Optional Design Patterns
Domain Driven Design
Event Sourcing and CQRS
Enterprise Service Bus
Domain Driven Design
Event Sourcing and CQRS
Event Streaming / Replicated Logs
Infrastructure Design Patterns
Container
Orchestrator
Service Mesh
Mandatory Design Patterns
Relational Database [SQL] / NoSQL
SQL
NoSQL
Continuous Integration (CI)
CI
CD
Development
QA / QC
Ops
DevOps
Waterfall
Agile
6/12 Months
Scrum (4-6 Weeks)
Scrum / Kanban (1-5 Days)

Monitoring & Observability

- Challenges in Monitoring
- Monitoring Vs. Observability
- ML / AI – based Analytics

Challenges in Monitoring

Blind Spot	Container / Pod Disposability increases Portability and Scalability – However, this creates blind spots in Monitoring.
Need to Record	Portability of inter-dependent components creates an increased need to maintain and record telemetry data with traceability to ensure Observability.
Visualization	The scale and complexity introduced by the Containers and Container Orchestration good tools to Visualize and Analyze the data generated.
Don't Leave DevOps in Dark	Application performance is Critical for Ops Team as Containers can be scaled up and down in lightning speed.

Source: A Beginners guide to Kubernetes Monitoring by Splunk

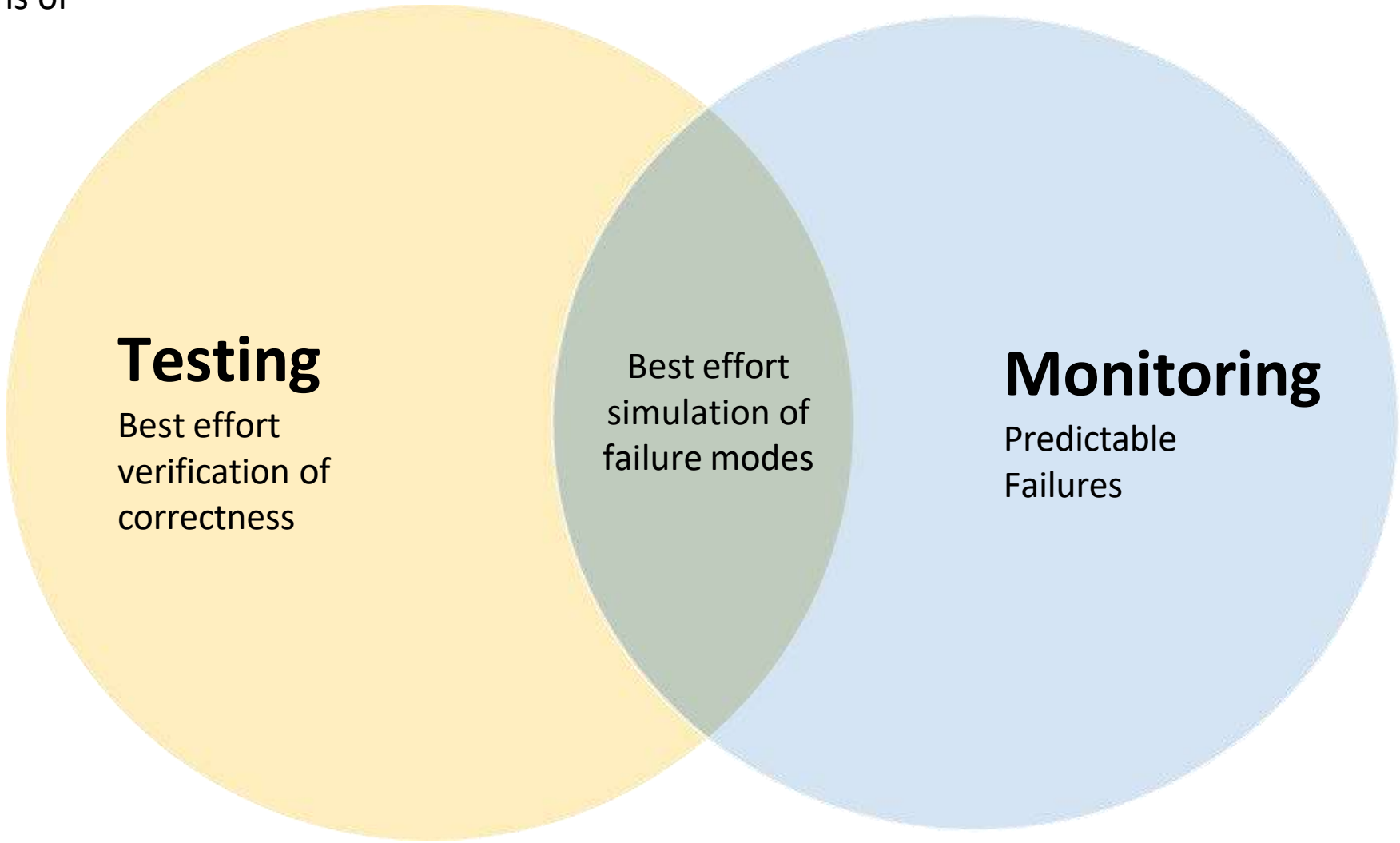
Monitoring Vs. Observability

	Monitoring	Observability
1	Says whether the System is Working or Not	Why its not working
2	Collects Metrics and Logs from a System	Actionable Insights gained from the Metrics
3	Failure Centric	Overall Behavior of the System
4	Is “the How” of something you do	Is ”The Process” of something you have
5	I monitor you	You make yourself observable

Source: A Beginners guide to Observability by Splunk

Observability

All possible permutations of full and partial failure



Source: A Beginners guide to Observability by Splunk

Benefits of Observability

1. Better understanding of complex microservices communication and end-user usage patterns
2. Helps in faster troubleshooting and shorter MTTR (Mean Time To Recovery)
3. Better understanding of incidents
4. Better uptime and performance
5. Happier customers and more revenue

Source: A Beginners guide to Observability by Splunk

Pillars of Observability

Logs/events

Immutable records of discrete events that happen over time

Metrics

Numbers describing a particular process or activity measured over intervals of time

Traces

Data that shows, for each invocation of each downstream service, which instance was called, which method within that instance was invoked, how the request performed, and what the results were

Source: A Beginners guide to Observability by Splunk

Events / Logs

Event Sources

- System and Server logs (syslog)
- Firewall and IDS/IPS logs
- Container / Pod Logs
- Application / Service / Database logs (log4j, log4net, Apache, MySQL, AWS)

Metrics

Metric Sources

- Infrastructure Metrics (Node, K8s)
- System Metrics (CPU, Memory, Disk)
- Service Metrics (Envoy Proxy)
- Network Metrics (Packets, Bytes)
- Business metrics (revenue, customer sign-ups, bounce rate, cart abandonment)
- UI Metrics (Google Analytics, Digital Experience Management)

Traces

- Specific parts of a user's journey are collected into traces, showing
- Which services were invoked,
- Which containers/hosts/instances they were running on, and
- what the results of each call were.

Kubernetes Auditing

- Auditing
- Audit Stages
- Audit Policy
- Audit Example

Kubernetes Auditing

Auditing Provides logs on what's happening within the cluster.
Scope and Levels of details are configurable

Forensics review of the Kubernetes logs shows the following

- What happened?
- When did it happen?
- Who initiated it?
- On what did it happen?
- Where was it observed?
- From where was it initiated?
- To where was it going?

Source: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

Kubernetes Audit Stages

Request Received	The stage for events generated as soon as the audit handler receives the request.
Response Started	Once the response headers are sent, but before the response body is sent.
Response Completed	The response body has been completed and no more bytes will be sent.
Panic	Events generated when a panic occurred.

Source: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

Lifecycle of an Audit Event

Kubernetes Audit Policy

None	Don't log events that match this rule.
MetaData	Log request metadata (requesting user, timestamp, resource, verb, etc.) but not request or response body.
Request	Log event metadata and request body but not response body. This does not apply for non-resource requests.
Request Response	Log event metadata, request and response bodies. This does not apply for non-resource requests.

Source: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

Kubernetes Audit Policy Example

Audit-Policy-Example.yaml

```yaml
apiVersion: audit.k8s.io/v1 # This is required.
kind: Policy
# Don't generate audit events for all requests in RequestReceived stage.
omitStages:
  - "RequestReceived"
rules:
  # Log pod changes at RequestResponse level
  - level: RequestResponse
    resources:
    - group: ""
      # Resource "pods" doesn't match requests to any subresource of pods,
      # which is consistent with the RBAC policy.
      resources: ["pods"]
  # Log "pods/log", "pods/status" at Metadata level
  - level: Metadata
    resources:
    - group: ""
      resources: ["pods/log", "pods/status"]

  # Don't log requests to a configmap called "controller-leader"
  - level: None
    resources:
    - group: ""
      resources: ["configmaps"]
      resourceNames: ["controller-leader"]

  # Don't log watch requests by the "system:kube-proxy" on endpoints or services
  - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
    - group: "" # core API group
      resources: ["endpoints", "services"]

  # Don't log authenticated requests to certain non-resource URL paths.
  - level: None
    userGroups: ["system:authenticated"]
    nonResourceURLs:
    - "/api*" # Wildcard matching.
    - "/version"
```

Kubernetes Native Monitoring

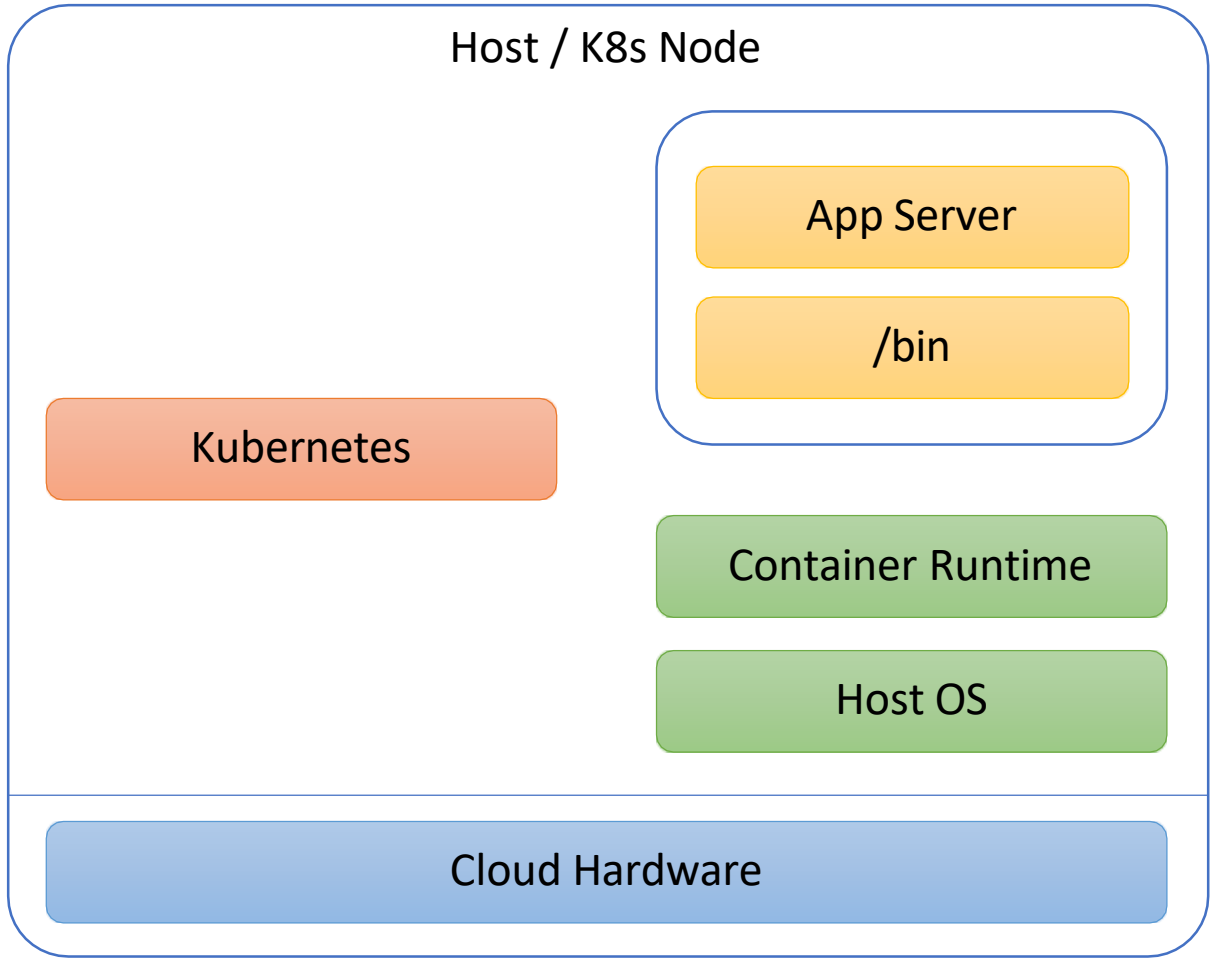
Application Logs (L7 Logs)

Container / Pod Logs
- Process
- System Calls
- Network Logs
- File System Logs

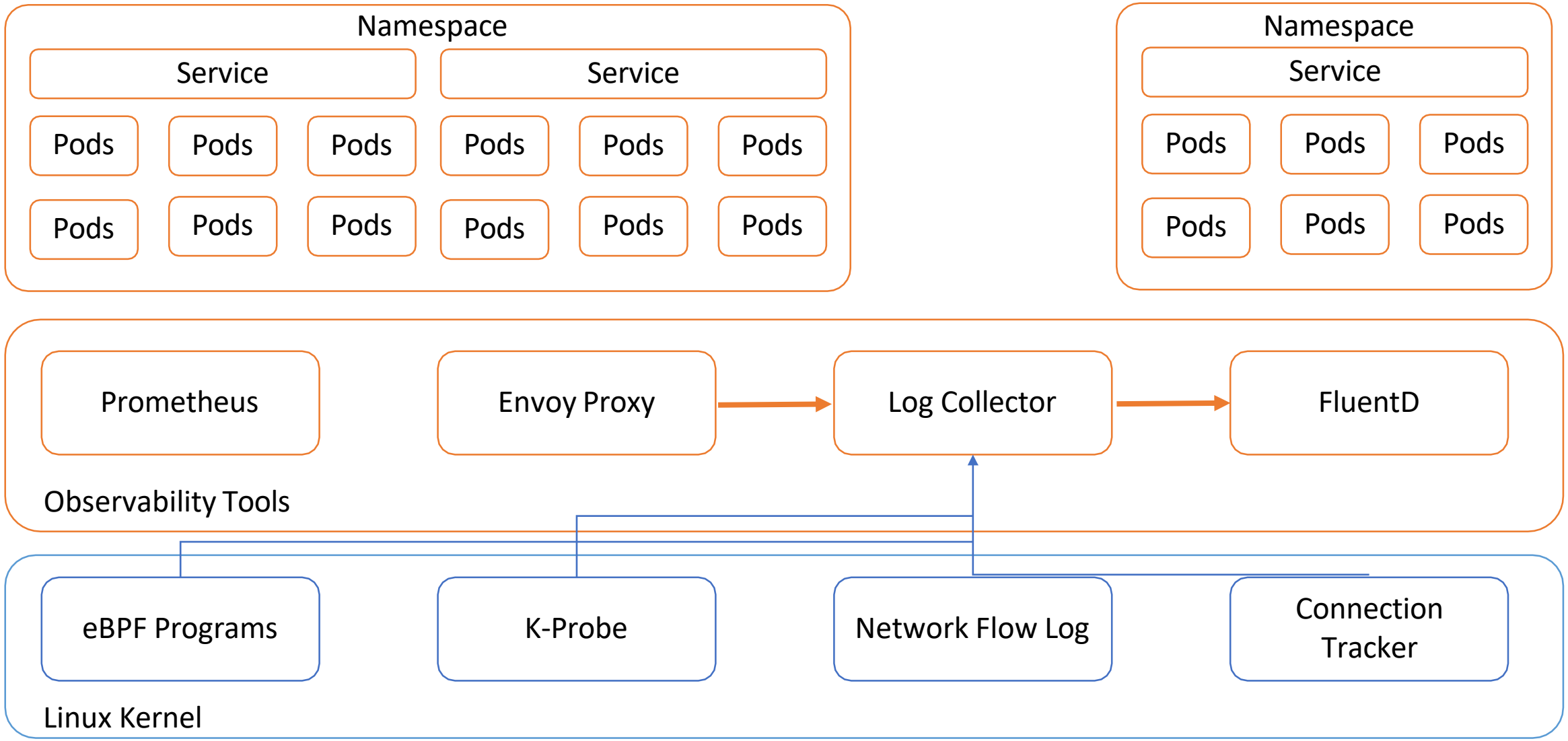
Kubernetes Logs
- Network Flow Logs
- Audit Logs
- DNS Logs

Host OS Logs
- SSH Logs
- OS Audit Logs

Cloud Infra Logs



Kubernetes Node



Data Collection

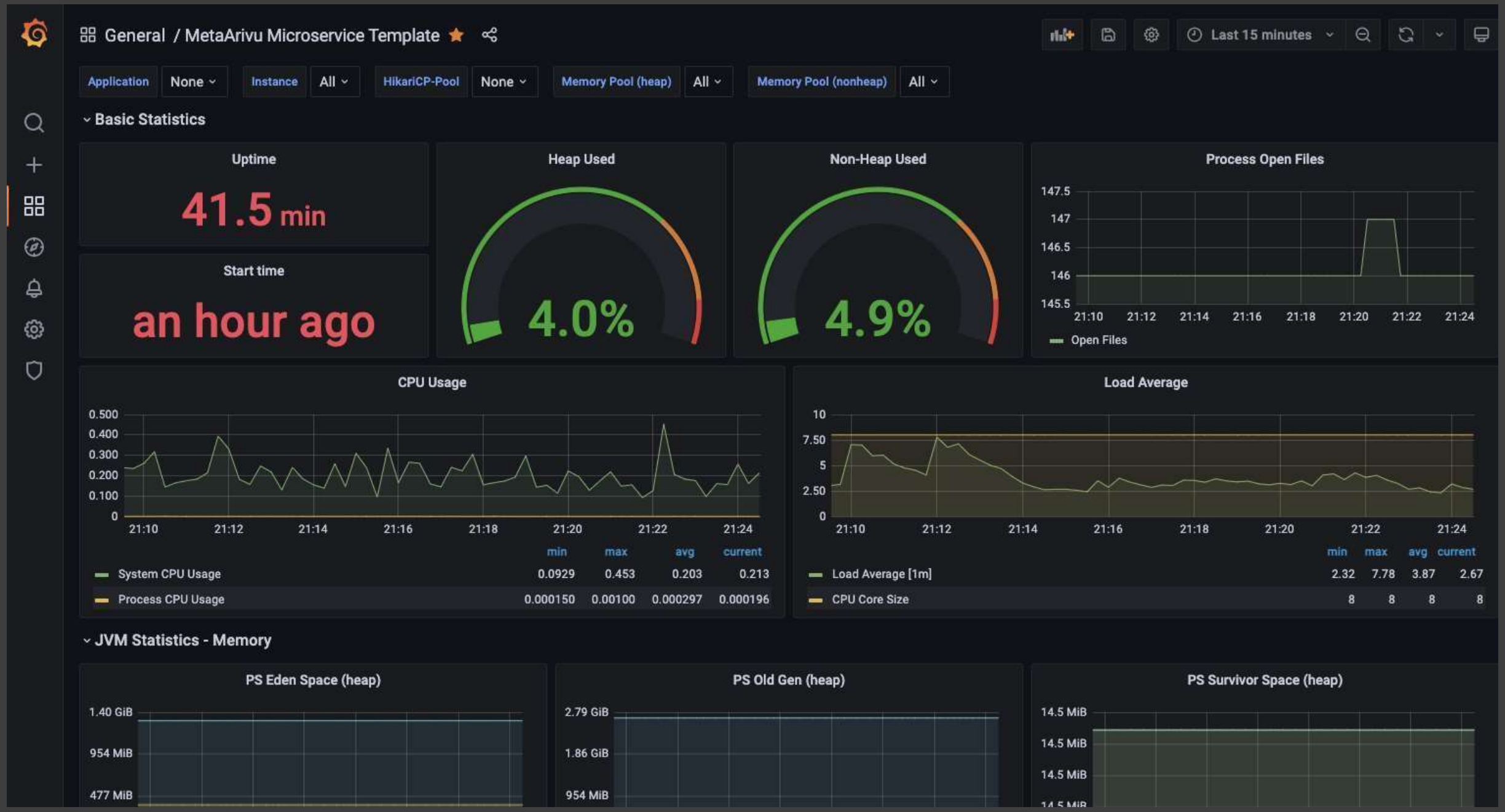
K-Probe	Source IP Address, Source Port, Destination IP Address, Destination Port, Protocol
NF Log	Adds Bytes and Packets count for the above five attributes for a connection
Log Collector	Adds Kubernetes Meta Data to the above data like Namespace, Service, Pod etc..
Prometheus	Collects metrics, System, Service metrics

Kubernetes Metrics Server

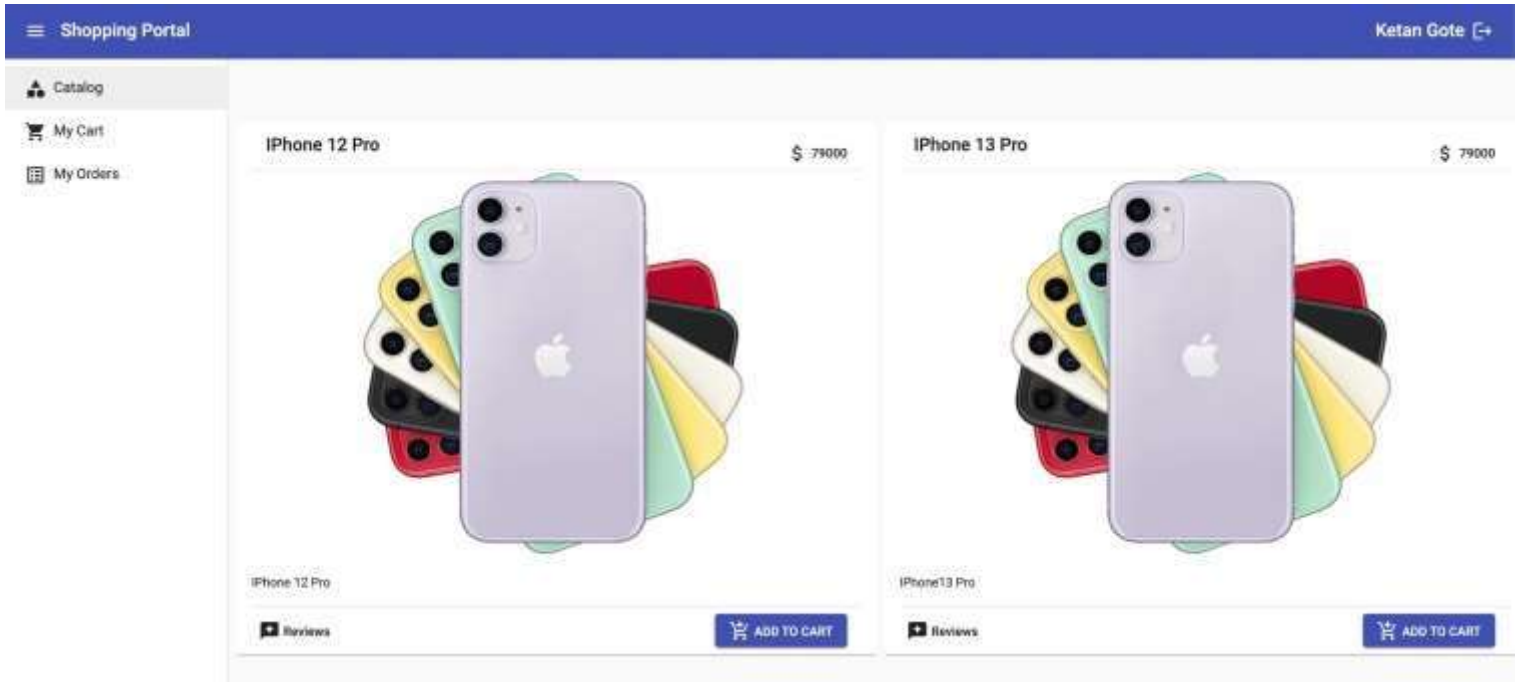
- Metrics Server is a cluster-wide aggregator of resource usage data.
- CPU is reported as the average usage, in CPU cores, over a period of time.
- Memory is reported as the working set, in bytes, at the instant the metric was collected.

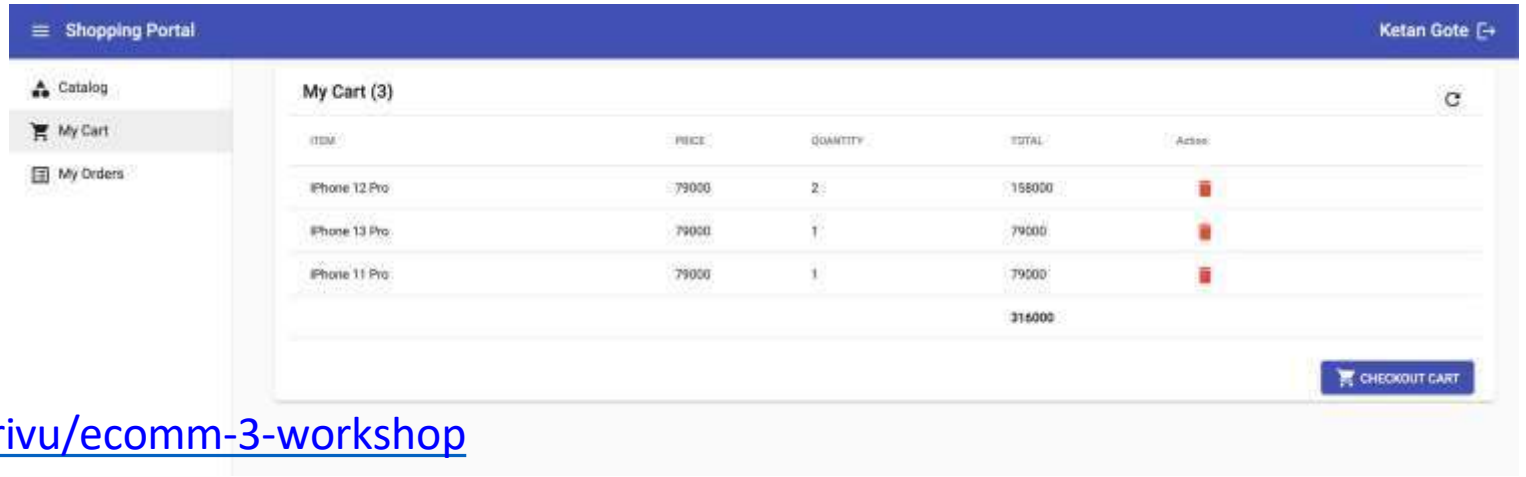
Source: https://kubernetes.io/docs/tasks/debug-application-cluster/resource-metrics-pipeline/

Zipkin / Prometheus / Grafana / Kiali


General / MetaArivu Microservice Template
Last 15 minutes
Application None
Instance All
HikariCP-Pool None
Memory Pool (heap) All
Memory Pool (nonheap) All
Basic Statistics
Uptime
41.5 min
Start time
an hour ago
Heap Used
4.0%
Non-Heap Used
4.9%
Process Open Files
147.5
147
146.5
146
145.5
21:10 21:12 21:14 21:16 21:18 21:20 21:22 21:24
Open Files
CPU Usage
0.500
0.400
0.300
0.200
0.100
0
21:10 21:12 21:14 21:16 21:18 21:20 21:22 21:24
min max avg current
System CPU Usage 0.0929 0.453 0.203 0.213
Process CPU Usage 0.000150 0.00100 0.000297 0.000196
Load Average
10
7.50
5
2.50
0
21:10 21:12 21:14 21:16 21:18 21:20 21:22 21:24
min max avg current
Load Average [1m] 2.32 7.78 3.87 2.67
CPU Core Size 8 8 8 8
JVM Statistics - Memory
PS Eden Space (heap)
1.40 GiB
954 MiB
477 MiB
PS Old Gen (heap)
2.79 GiB
1.86 GiB
954 MiB
PS Survivor Space (heap)
14.5 MiB
14.5 MiB
14.5 MiB

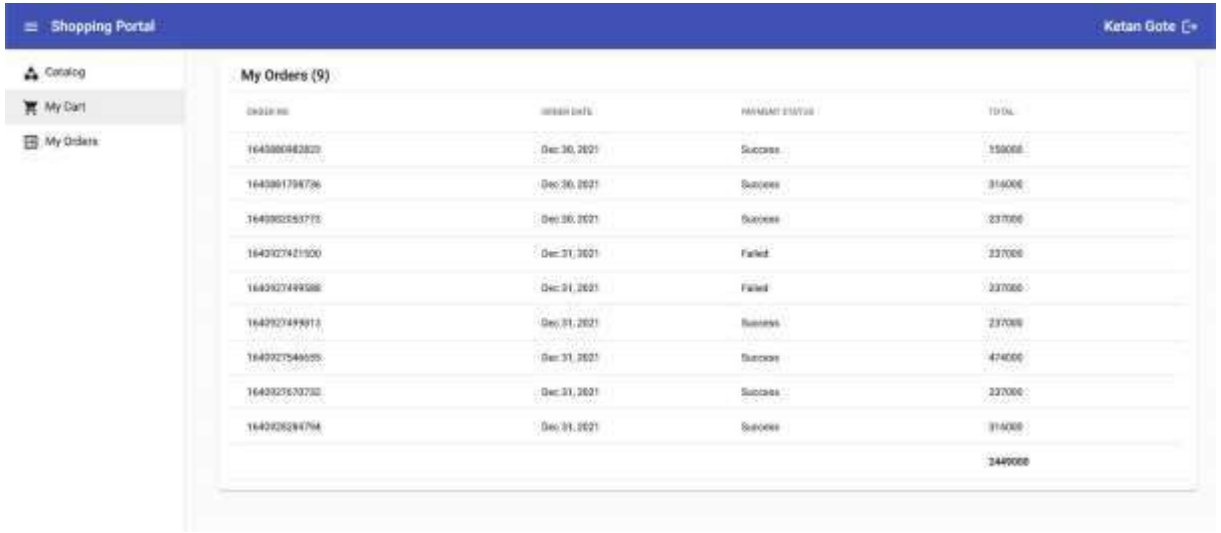
Shopping Portal App





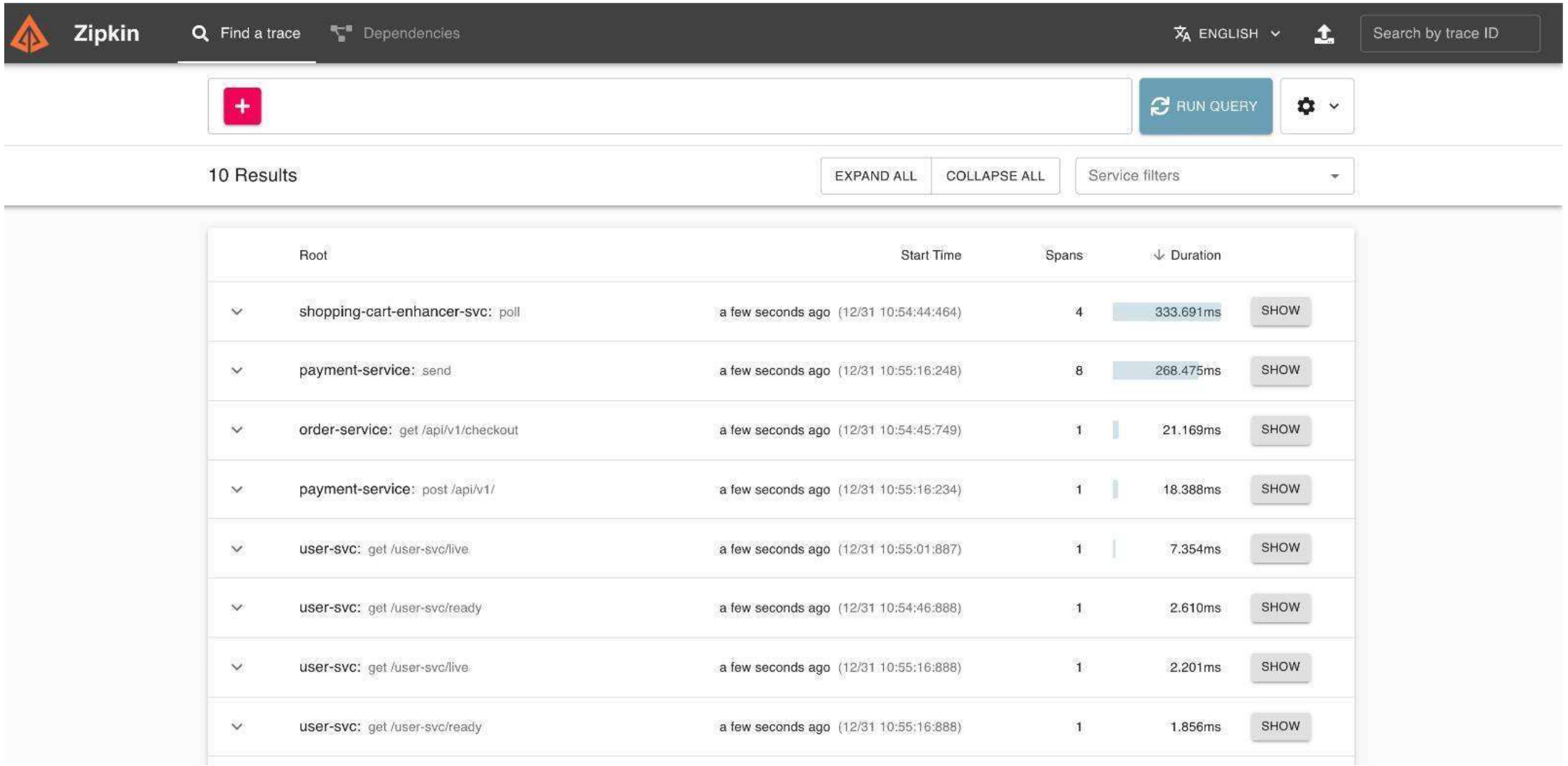
Source: https://github.com/MetaArivu/ecomm-3-workshop

Shopping Portal App



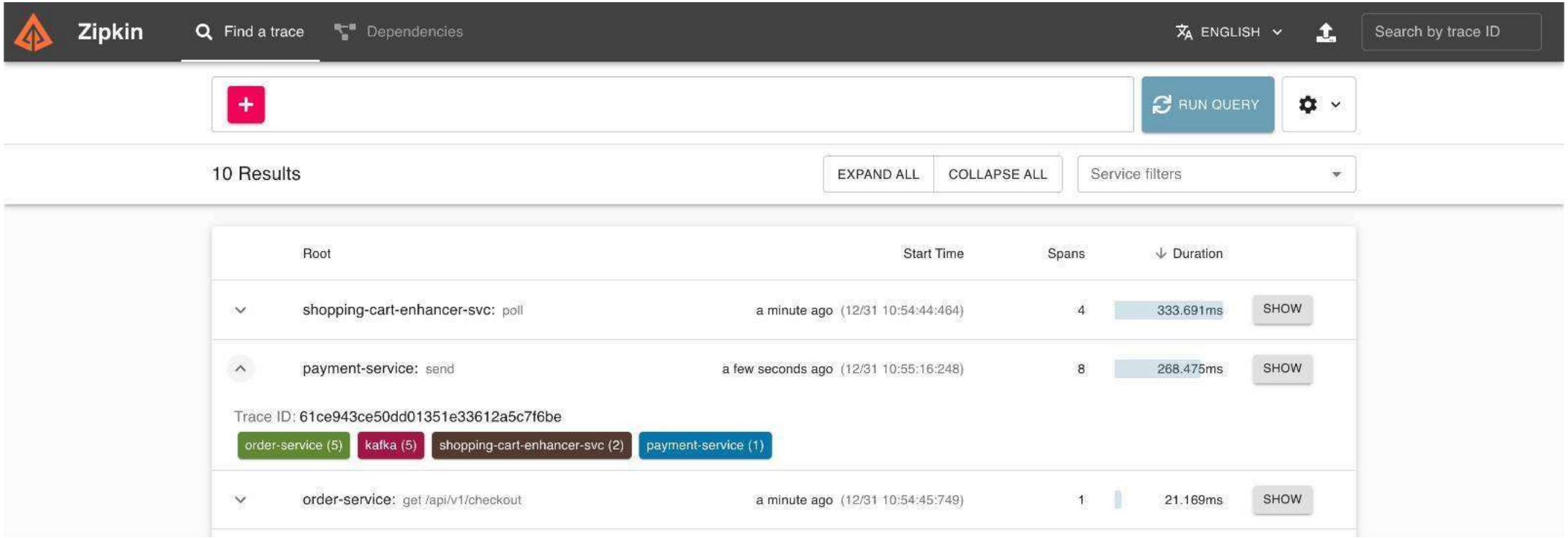
Source: https://github.com/MetaArivu/ecomm-3-workshop

Zipkin



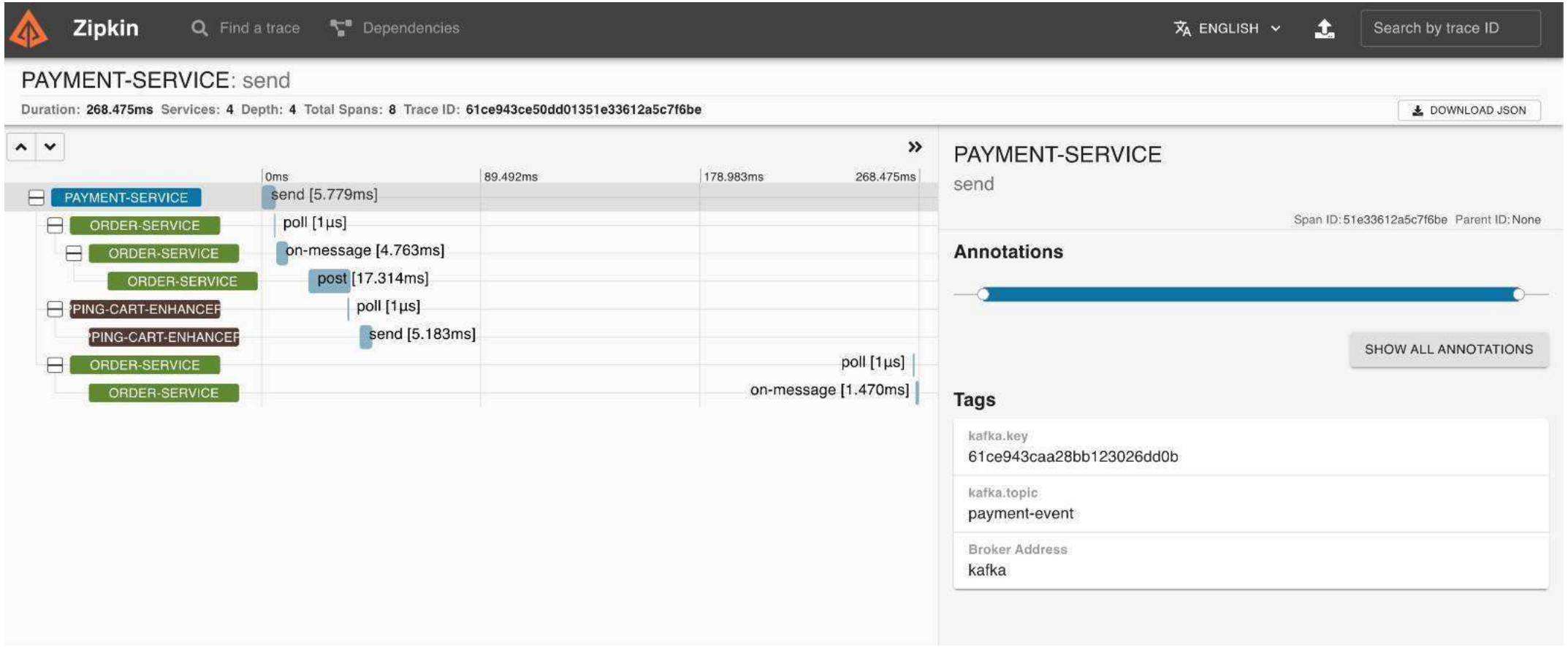
Source: https://github.com/MetaArivu/ecomm-3-workshop

Zipkin Traces



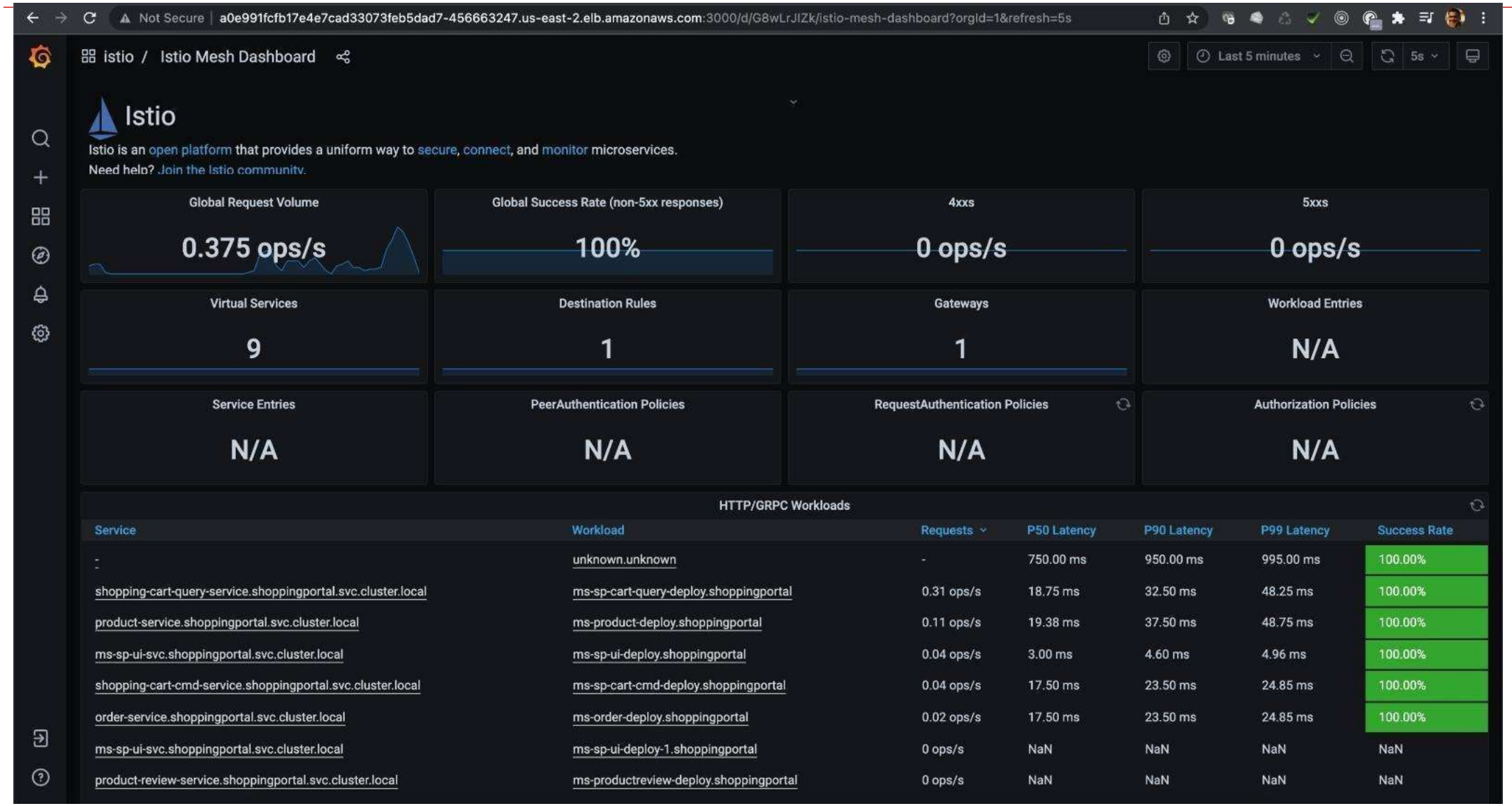
Source: https://github.com/MetaArivu/ecomm-3-workshop

Zipkin Traces



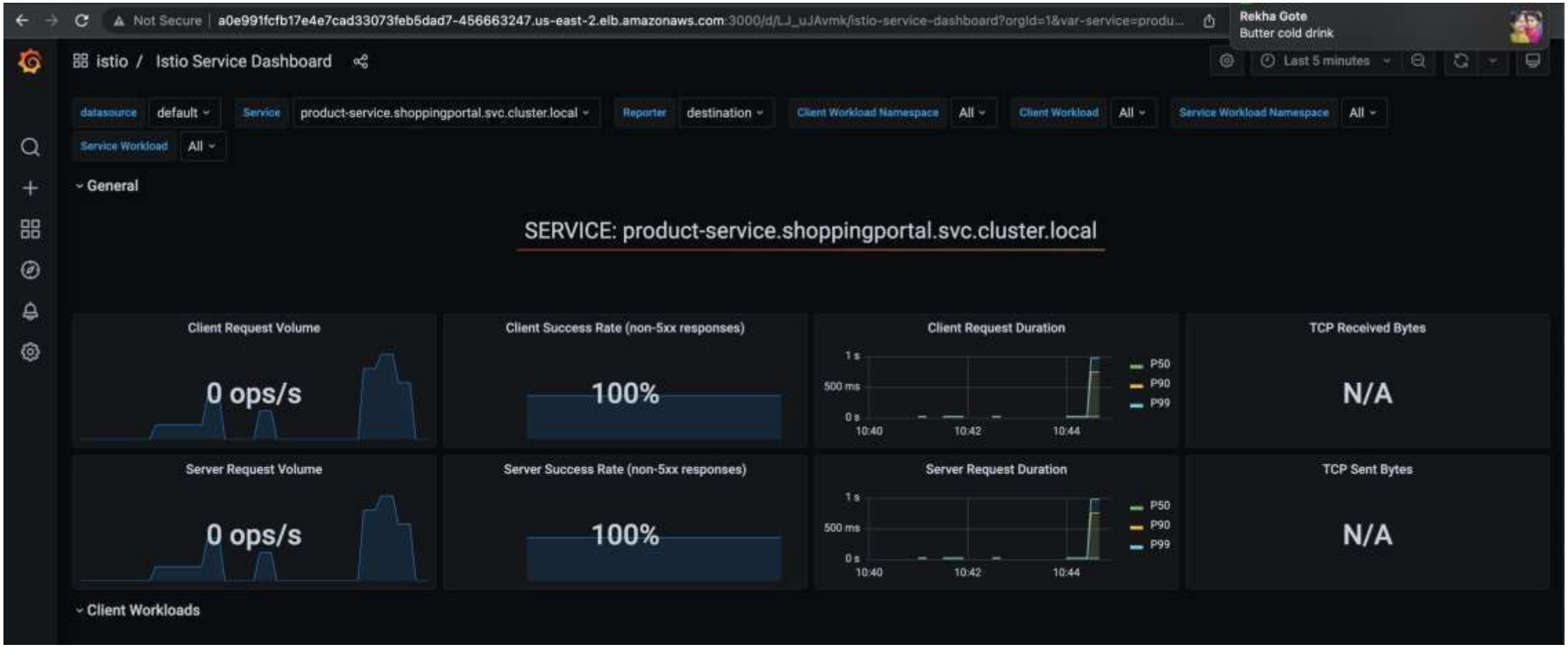
Source: https://github.com/MetaArivu/ecomm-3-workshop

Prometheus / Grafana



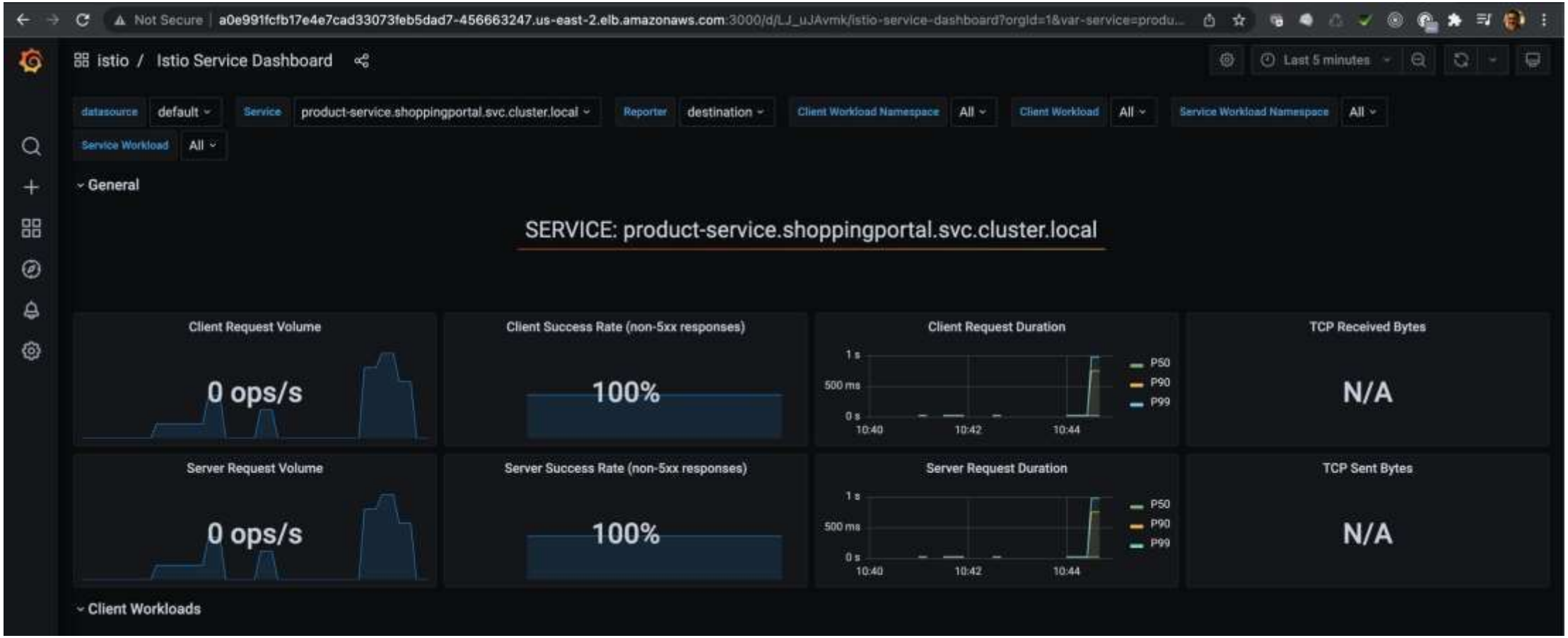
Source: https://github.com/MetaArivu/ecomm-3-workshop

Prometheus / Grafana



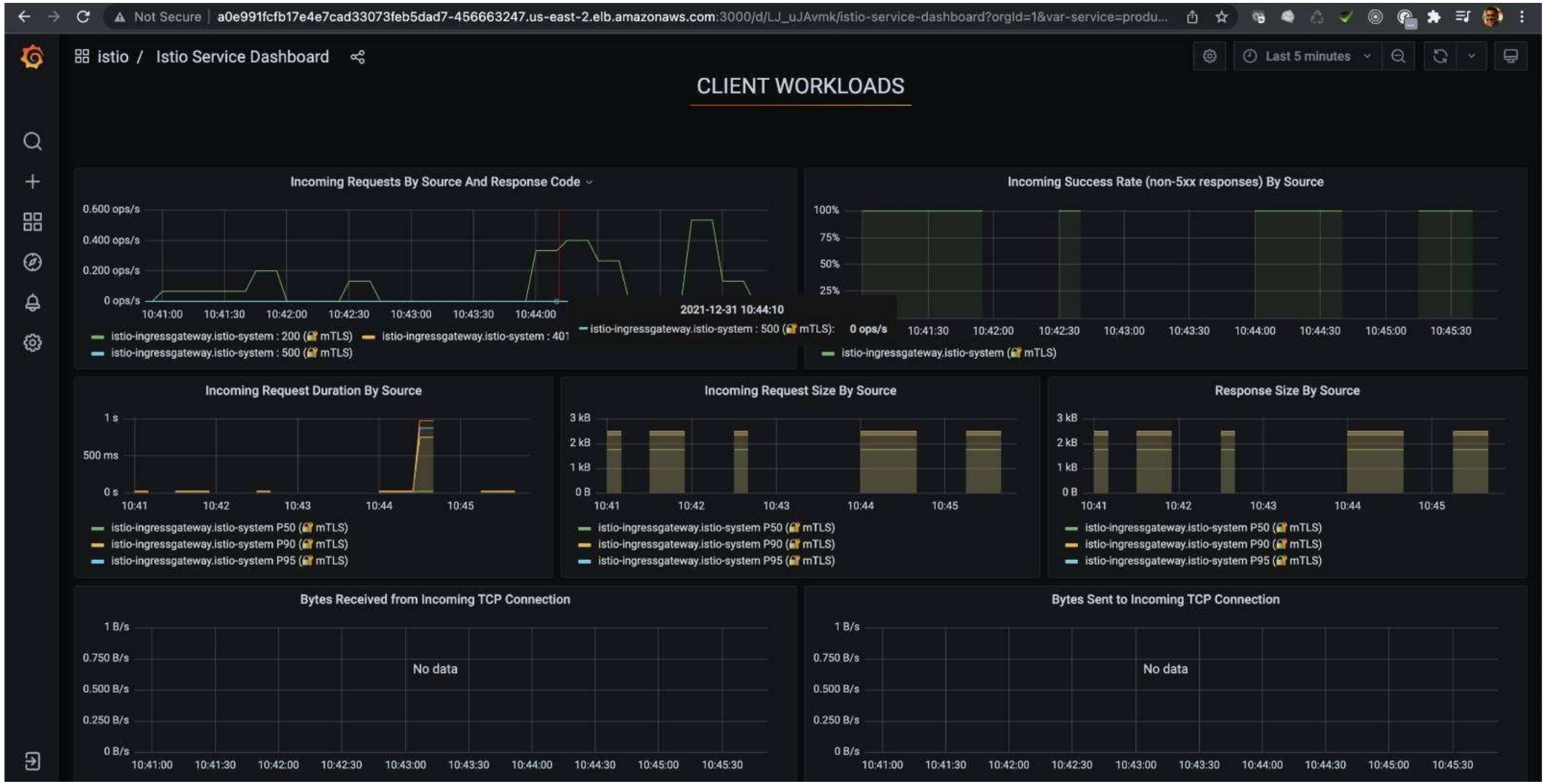
Source: https://github.com/MetaArivu/ecomm-3-workshop

Prometheus / Grafana



Source: https://github.com/MetaArivu/ecomm-3-workshop

Prometheus / Grafana

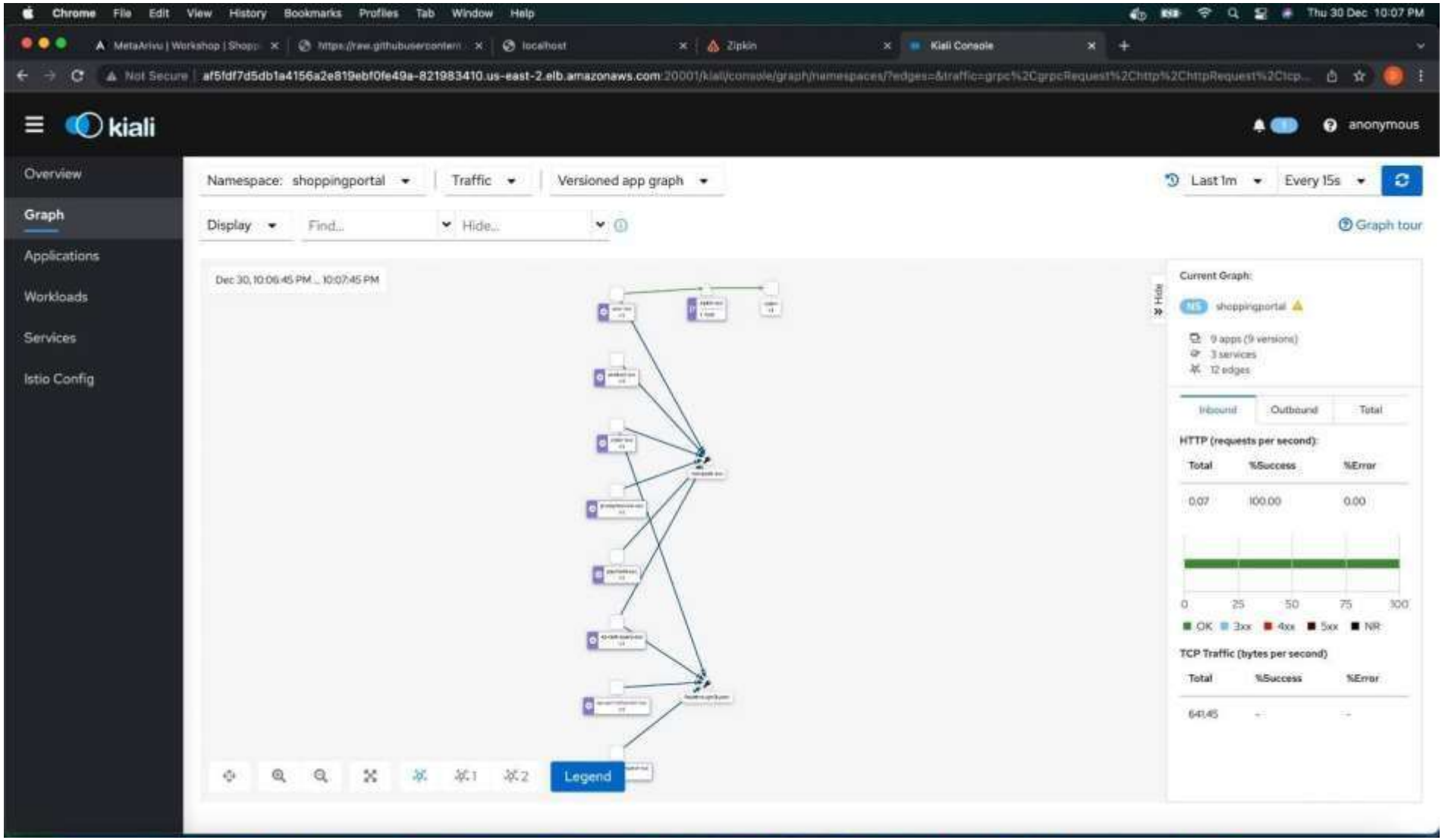


Source: https://github.com/MetaArivu/ecomm-3-workshop

Prometheus / Grafana

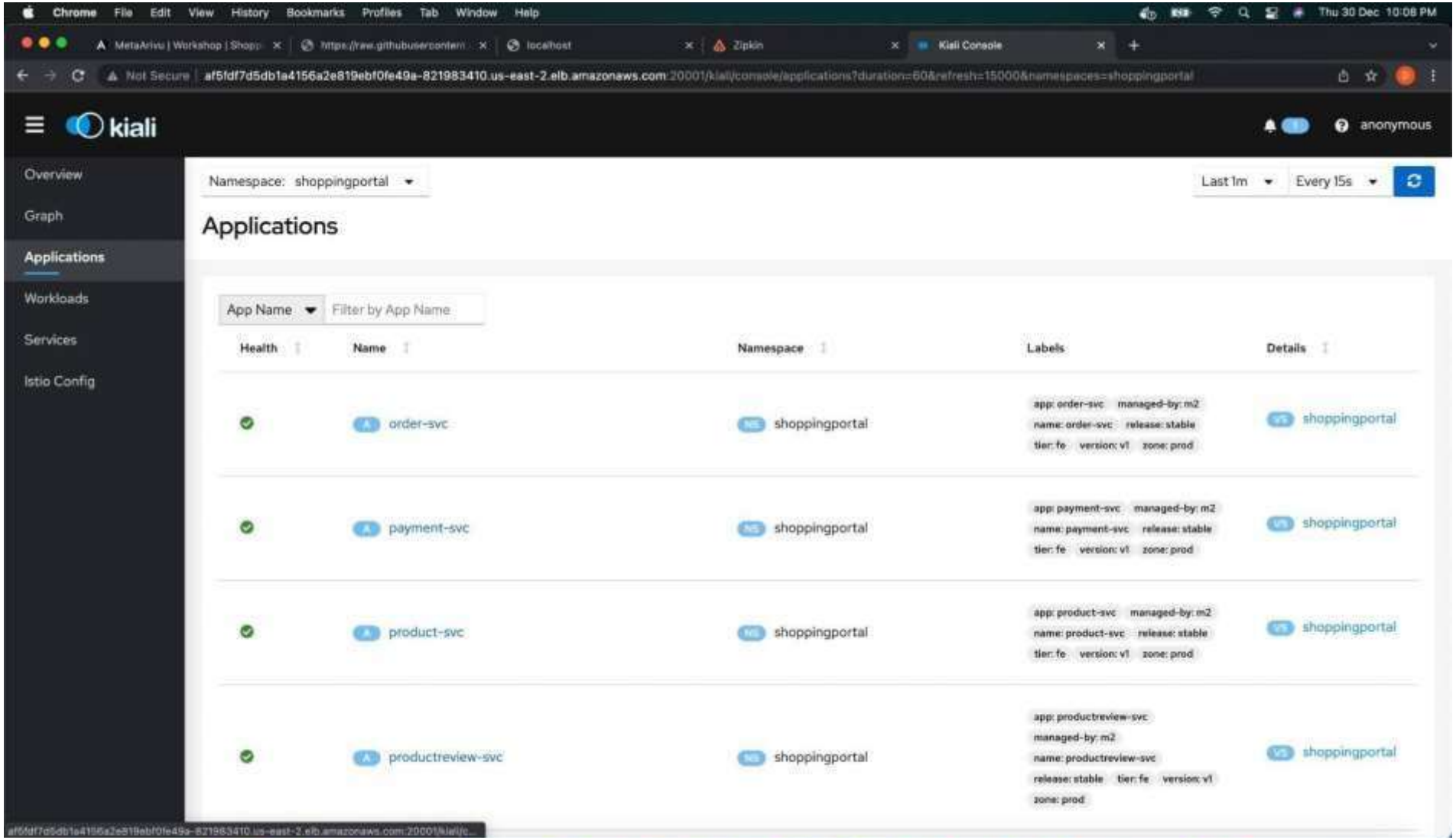
Source: https://github.com/MetaArivu/ecomm-3-workshop

Kiali



Source: https://github.com/MetaArivu/ecomm-3-workshop

Kiali



Source: https://github.com/MetaArivu/ecomm-3-workshop

ML / AI

- Analytics
- Anomalous Events Example

ML/AI Driven Analytics

- **Enrich:** Adding context to events to make them informative and actionable
- **Reduce Duplicate:** Automatically concealing duplicate events to focus on relevant ones and reducing alert storms
- **Reduce False +ve**: Reducing event clutter and false positives with multivariate anomaly detection
- **Filter/Tag/Sort**: Easily sifting through vast amounts of events by filtering, tagging and sorting

Source: A Beginners guide to Observability by Splunk

Anomalous Events

IP Sweep Detection	Pods sending many packets to many destinations
Port Scan Detection	Pods sending packets to One Destination on multiple ports.
HTTP Spike	Service that get too many HTTP inbound Connections
DNS Latency	Too High Latency for DNS Requests
L7 Latency	Pods with Too High Latency for L7 Requests

Source: Kubernetes Security and Observability: Brendan Creane & Amit Gupta

cicd-github-action lines 816

Public

CI / CD Pipeline using Git Actions. Fully automated pipeline with continuous deployment up to the Production Environment. CI / CD Pipeline using Jenkins and Tekton available in other repositories.

Java 2

terraform-quickstart

lines 11.8k Public

Terraform Examples. How to create your cloud infrastructures, examples include AWS VPC, EC2, S3, Route53, ELB, EKS, Apache Web Server, RDS (MySQL), MongoDB, Redis, Apache Solr, Kafka, etc.

3

k8s-quickstart lines 373.5k

Public

Kubernetes in Docker (KinD) Examples with 3 Node clusters and 5 Node Cluster and deploying a Web Server and 2 Tomcats as App Servers and an E-Commerce App with Order, Cart, and Payment Services.

HTML 3

Kafka-quickstart lines 19.8k

Public

Kafka Examples focusing on Producer, Consumer, KStreams, KTable, Global KTable using Spring, Kafka Cluster Setup & Monitoring. Implementing Event Sourcing and CQRS Design Pattern using Kafka

Java 29 12

ms-quickstart lines 5.6k

Public template

Microservice Template gives you a SpringBoot App template with Open API 3 Ex, Actuator, Sleuth, Prometheus, and POM file with Fat and Thin jar file creation and Dockerfile for containerization.

Java 1 2

ms-test-quickstart

lines 63.5k Public

Microservice Test Automation Example for SpringBootTest 2, JUnit 5, Cucumber 6, Selenium 4, Mockito 3, WireMock 2, Pact 4. All the testing frameworks are required to fully automate the testing for ...

Java 1

Source Code: https://github.com/MetaArivu Web Site: https://metarivu.com/ https://pyxida.cloud/

Thank you

DREAM | AUTOMATE | EMPOWER

Araf Karsh Hamid :
India: +91.999.545.8627
http://www.slideshare.net/arafkarsh
https://www.linkedin.com/in/arafkarsh/
https://www.youtube.com/user/arafkarsh/playlists
http://www.arafkarsh.com/

@arafkarsh
arafkarsh