



KubeCon



CloudNativeCon

Europe 2023



TiKV



KubeCon



CloudNativeCon

Europe 2023

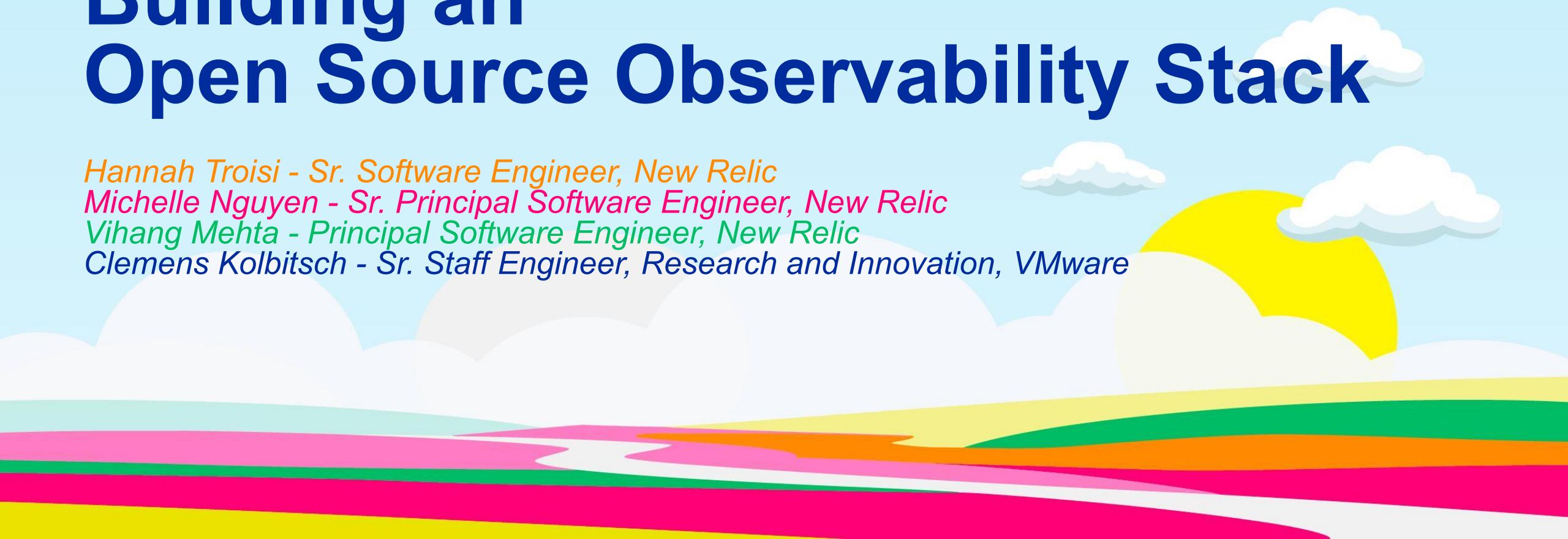
Building an Open Source Observability Stack

Hannah Troisi - Sr. Software Engineer, New Relic

Michelle Nguyen - Sr. Principal Software Engineer, New Relic

Vihang Mehta - Principal Software Engineer, New Relic

Clemens Kolbitsch - Sr. Staff Engineer, Research and Innovation, VMware



Welcome!



Hannah Troisi
Sr. Software
Engineer
New Relic



Michelle Nguyen
Sr. Principal
Software Engineer
New Relic



Vihang Mehta
Principal Software
Engineer
New Relic



**Clemens
Kolbitsch**
Sr. Staff Engineer
VMware



What is your greatest challenge in observability?

50% Engineers and teams using multiple tools

40% Shortage of skills

36% Organization — presence of process and team silos

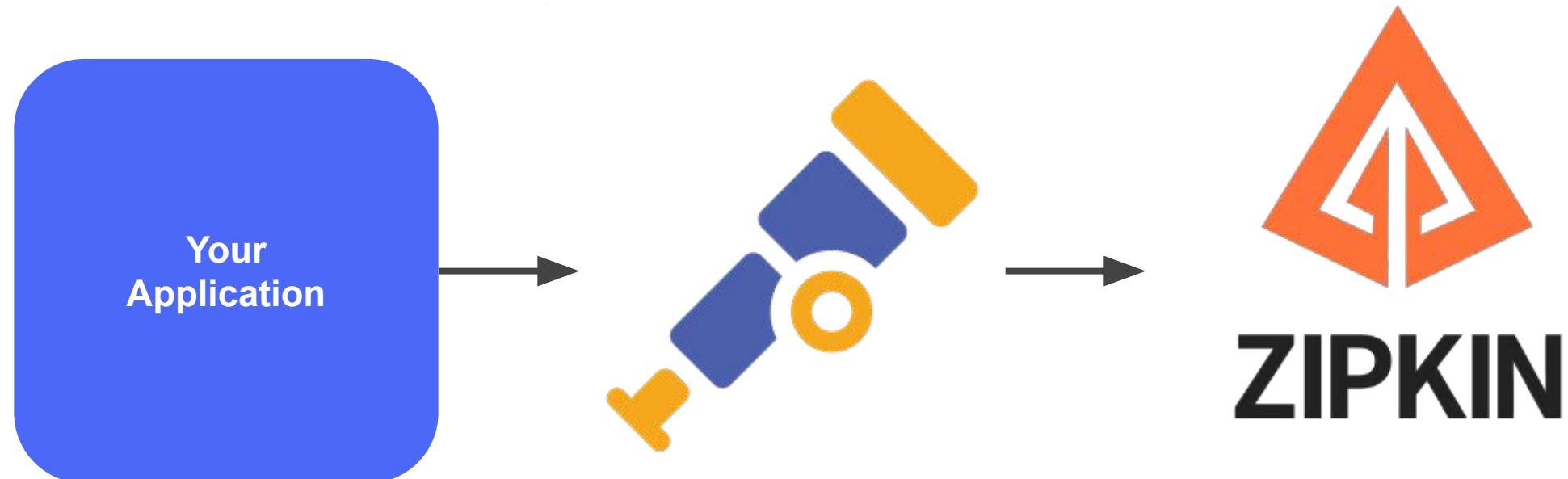
35% Lack of resources

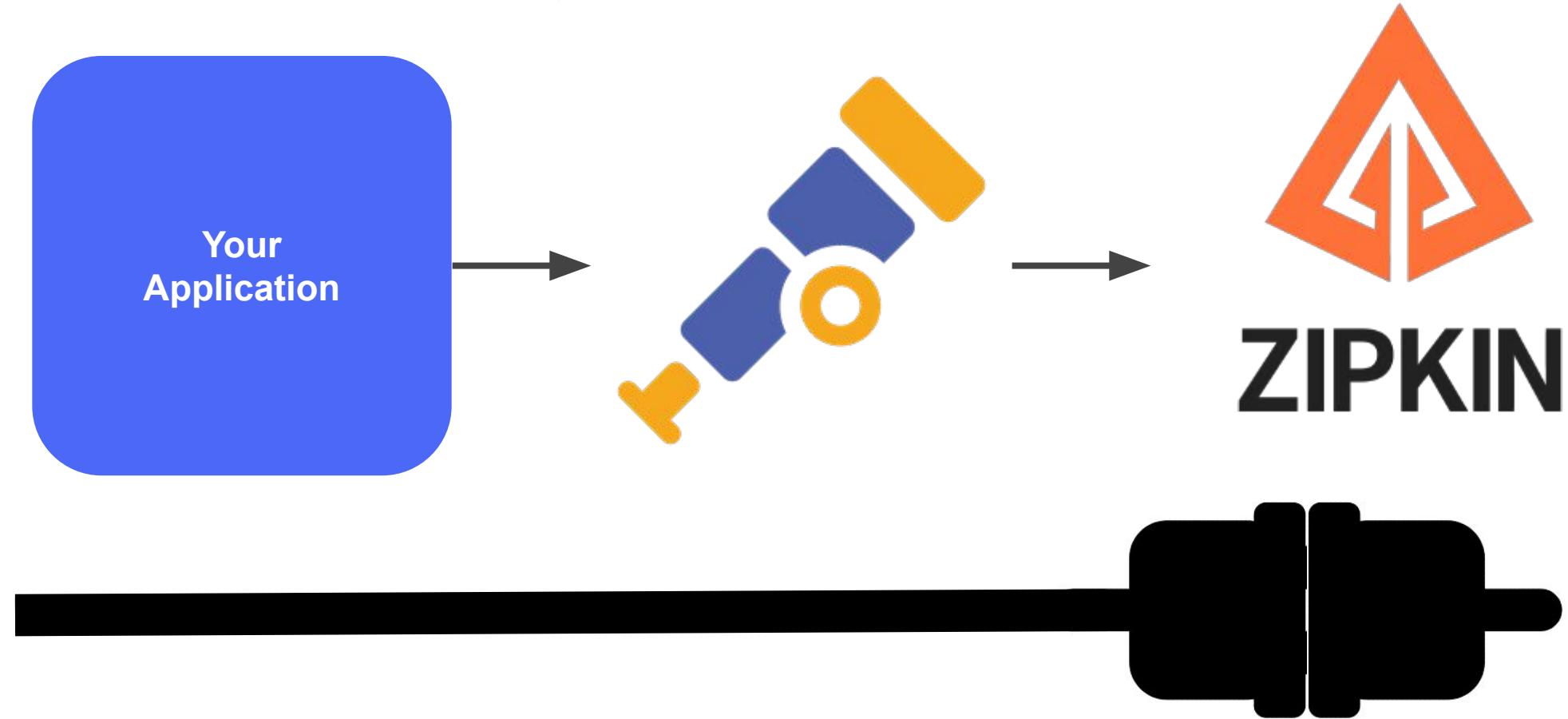
35% Complexity

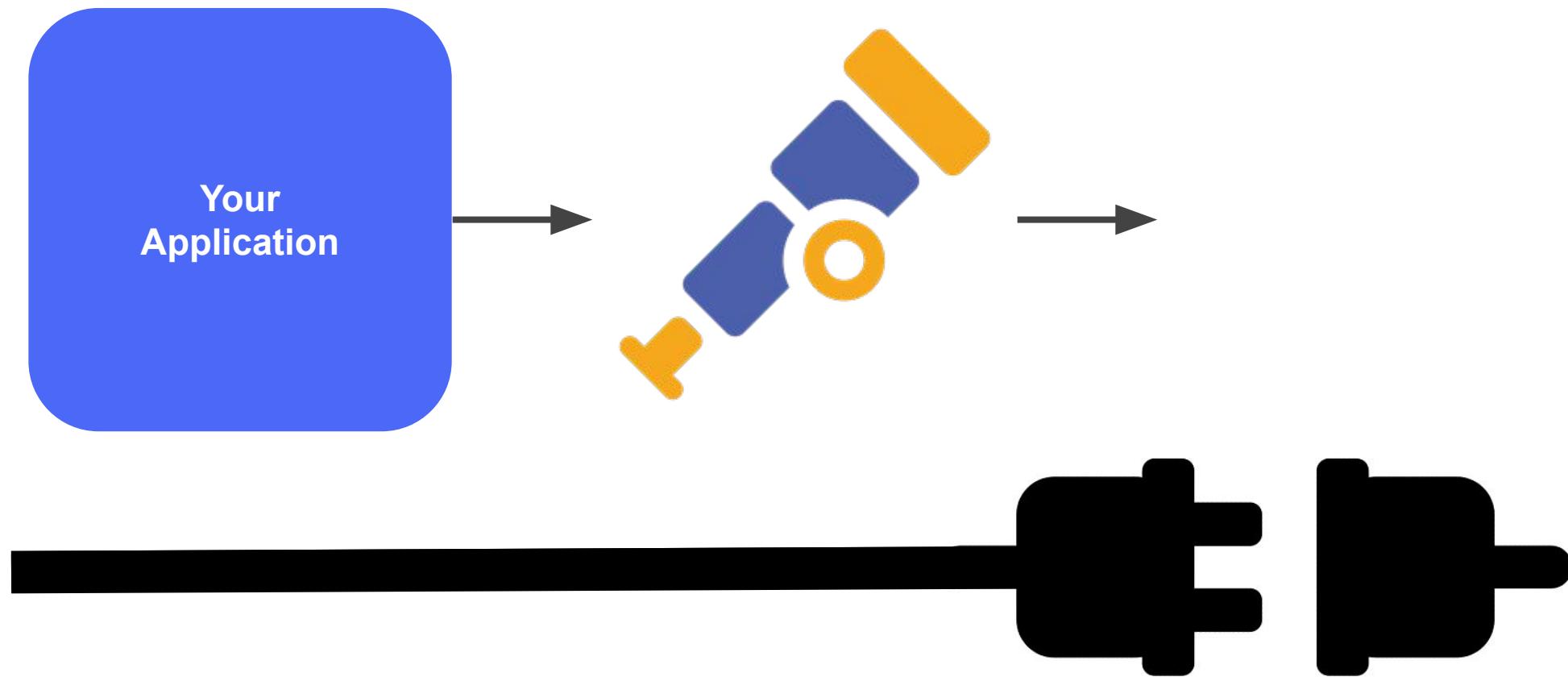
34% No strategy

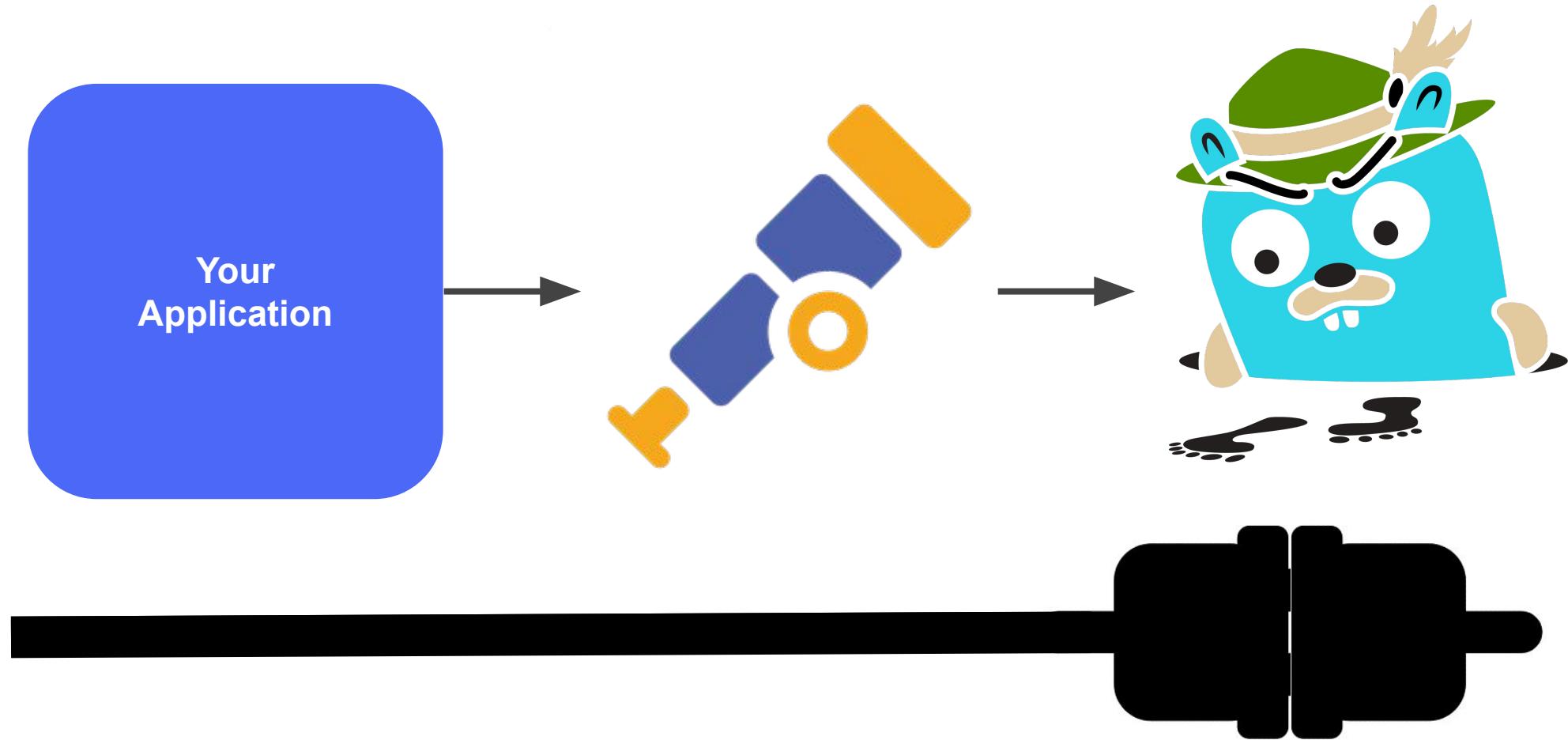
25% Lack of understanding of the benefits

24% Lack of leadership from management









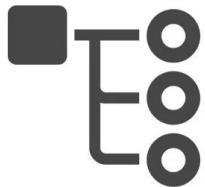
Metrics, Logs, Traces



Log: immutable, timestamped record of discrete events



Metric: numeric representation of data measured over intervals of time

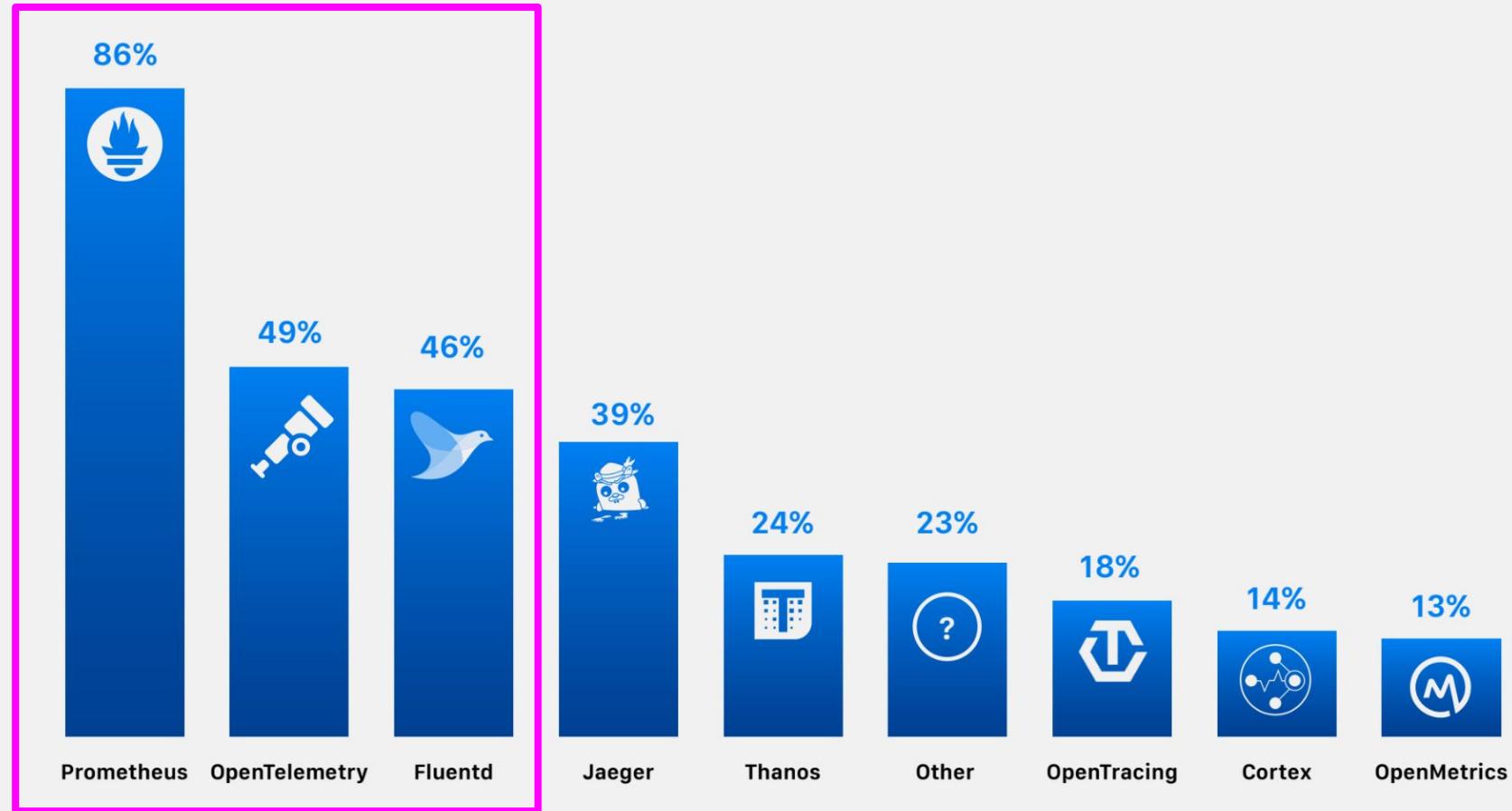


Trace: represents a series of related events as a request flows through a distributed system

Span: a single event within a trace, such as a database call or HTTP request from one pod/service to another

Kubernetes Observability

Which, if any, of the following projects do you use for observability?



Kubernetes Observability



fluentbit

CNCF GRADUATED PROJECT

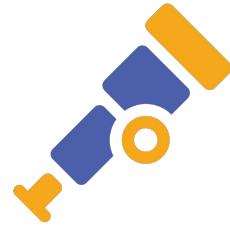
Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Kubernetes Observability



fluentbit

CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Why is logging challenging in Kubernetes?

- Pods are ephemeral
- Diverse log formats
- Lack of centralization
- Performance

Why is logging challenging in Kubernetes?

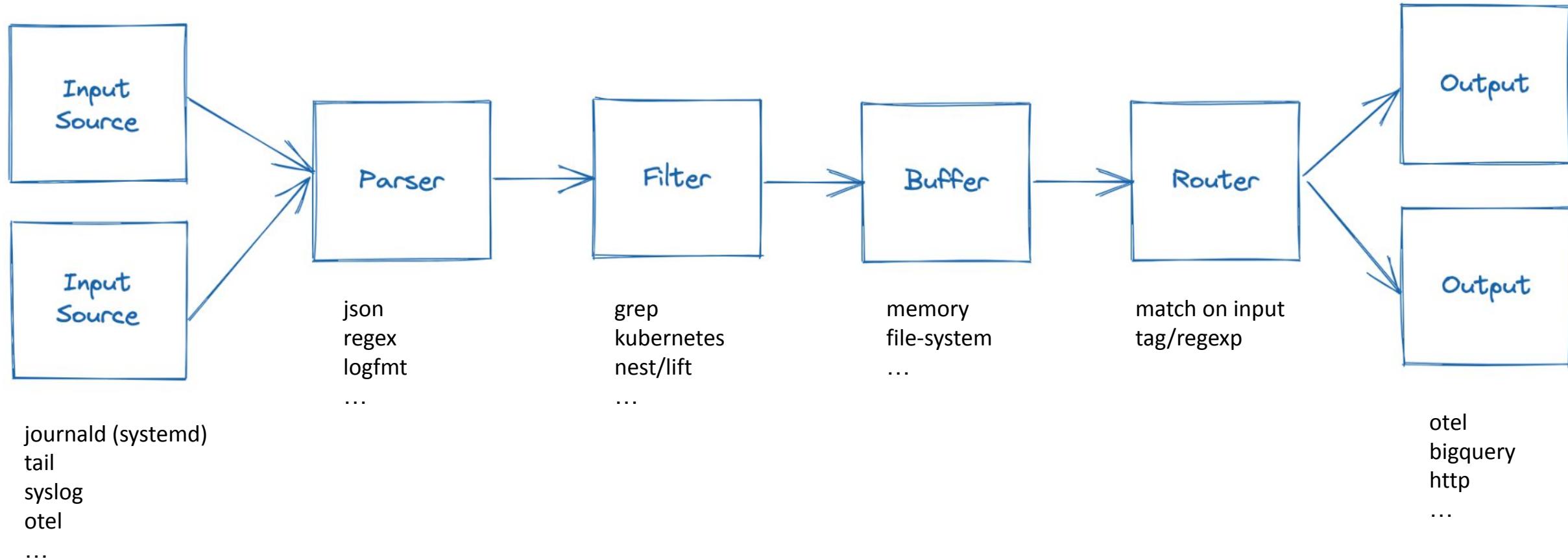
- Pods are ephemeral
- Diverse log formats
- Lack of centralization
- Performance



Why is logging challenging in Kubernetes?

- Pods are ephemeral
- Diverse log formats
- Lack of centralization
- Performance





Logging



fluentd

vs



fluentbit****

Kubernetes Observability



fluentbit

CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Kubernetes Observability



CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

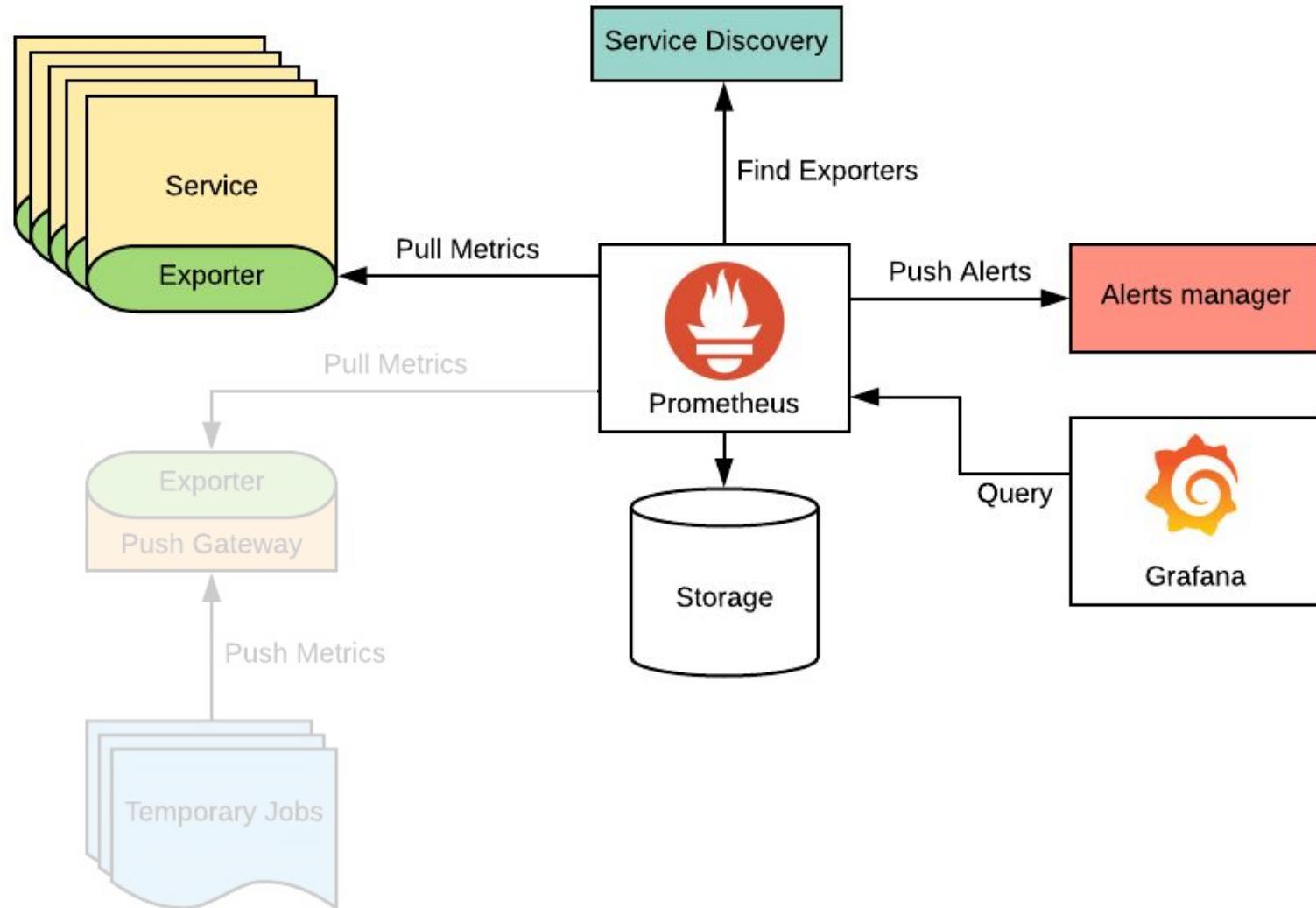
Metrics, Traces,
Application Profiles

What considerations are required for a Kubernetes metrics solution?

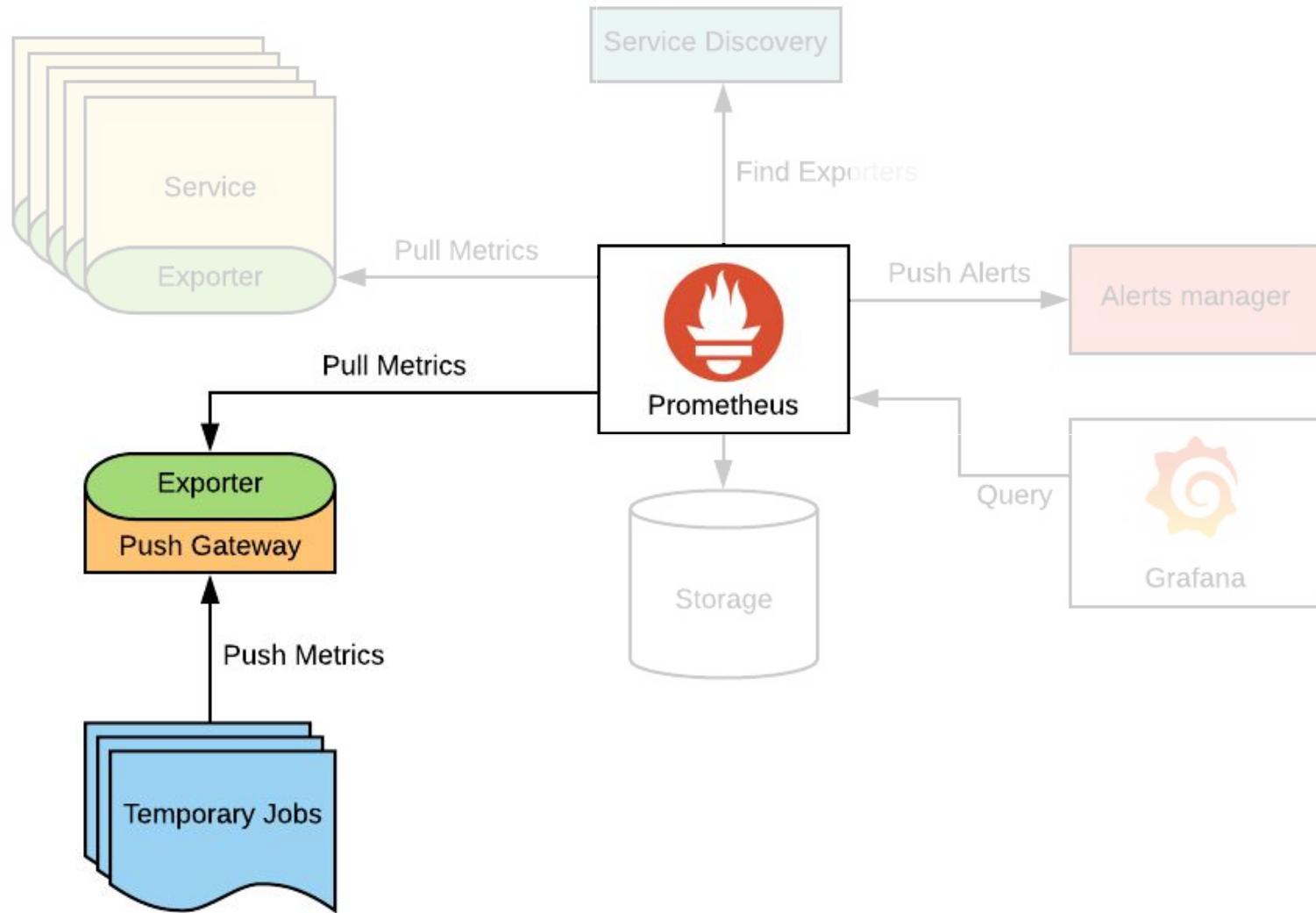
- Pods are ephemeral
- Millions of metrics
- Different metric formats
- Dimensional metrics
- Centralized access

```
# HELP http_requests_total Total number of http api requests
# TYPE http_requests_total counter
http_requests_total{app="shopping-cart", env="prod",
pod="shopping-cart-654b8f56c4-xqxct"} 4633433
```

Prometheus



Prometheus



Kubernetes Observability



CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Kubernetes Observability



CNCF GRADUATED PROJECT

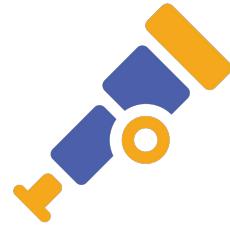
Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Kubernetes Observability



CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs

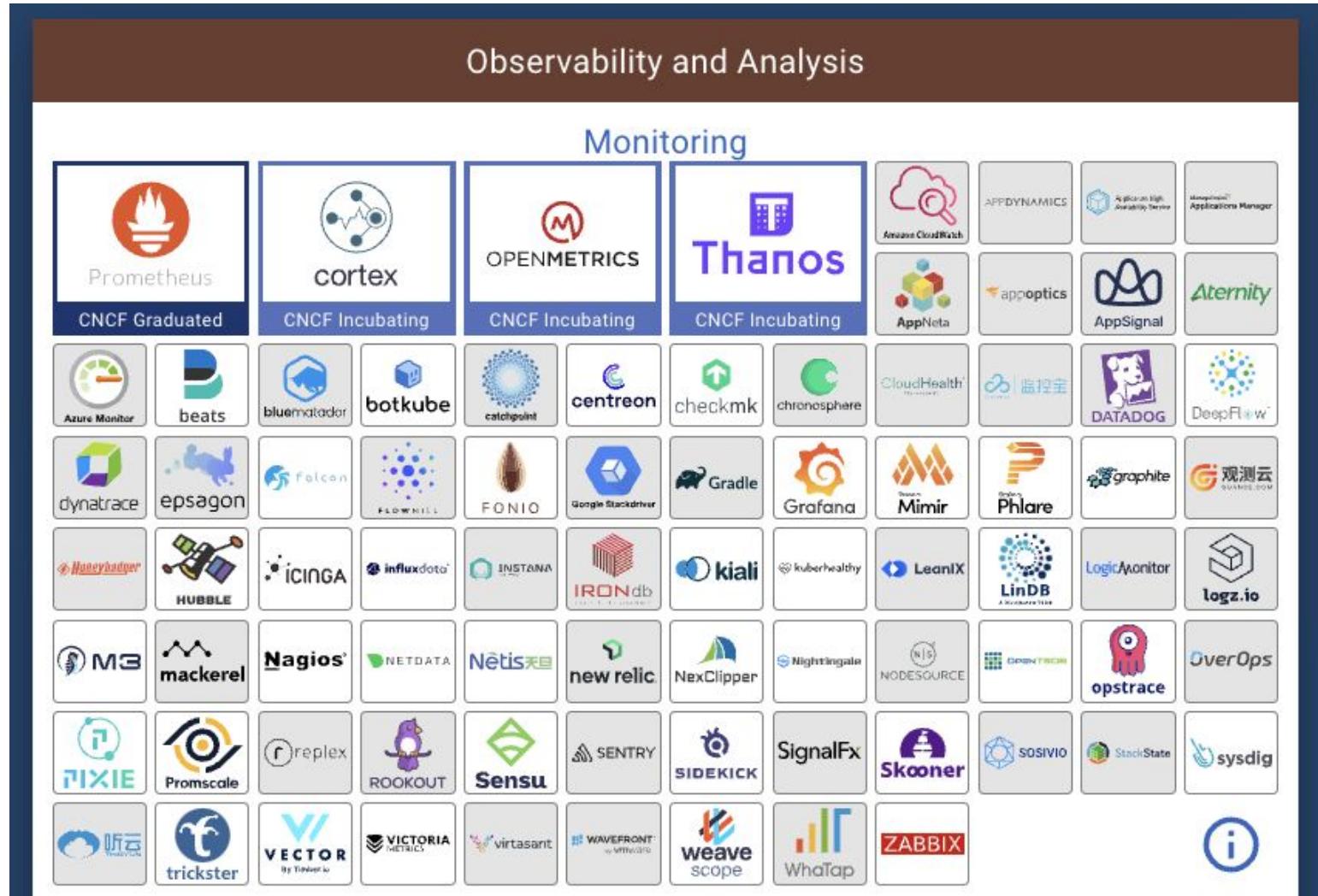


PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Which Observability tool?





What is OpenTelemetry (OTel)?



What is OpenTelemetry (OTel)?

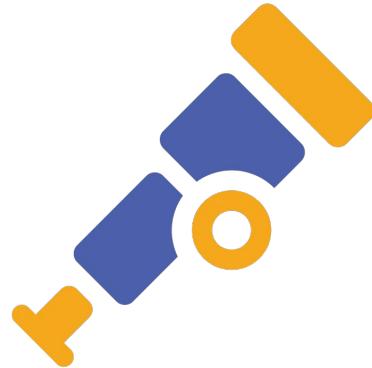
- **A universal format for telemetry data** (metrics, traces, logs)
- **Client libraries to instrument your app** to emit data in the OTel format
- **An API for sending & receiving data in the OTel format** (OTLP)
- **A data collector** (the OTel collector) that can receive, transform, and send data in the OTel format

Metrics Providers



Prometheus

vs



OpenTelemetry

Kubernetes Observability



CNCF GRADUATED PROJECT

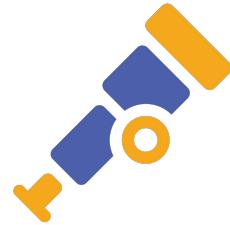
Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Kubernetes Observability



CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Adding Instrumentation Is Tedious

Boilerplate
language-specific
instrumentation, added
manually

```
import (
    "net/http"
    "net/http/httptrace"

    "github.com/opentracing/opentracing-go"
    "github.com/opentracing/opentracing-go/log"
    "golang.org/x/net/context"
)

// We will talk about this later
var tracer opentracing.Tracer

func AskGoogle(ctx context.Context) error {
    // retrieve current Span from Context
    var parentCtx opentracing.SpanContext
    parentSpan := opentracing.SpanFromContext(ctx);
    if parentSpan != nil {
        parentCtx = parentSpan.Context()
    }

    // start a new Span to wrap HTTP request
    span := tracer.StartSpan(
        "ask google",
        opentracing.ChildOf(parentCtx),
    )

    // make sure the Span is finished once we're done
    defer span.Finish()

    // make the Span current in the context
    ctx = opentracing.ContextWithSpan(ctx, span)

    // now prepare the request
    req, err := http.NewRequest("GET", "http://google.com", nil)
    if err != nil {
        return err
    }

    // attach ClientTrace to the Context, and Context to request
    trace := NewClientTrace(span)
    ctx = httptrace.WithClientTrace(ctx, trace)
    req = req.WithContext(ctx)

    // execute the request
    res, err := http.DefaultClient.Do(req)
    if err != nil {
        return err
    }

    // Google home page is not too exciting, so ignore the result
    res.Body.Close()
    return nil
}
```

Actual
Business
Logic

Auto-Telemetry?

What if we could automatically capture telemetry data?

Auto-Telemetry?

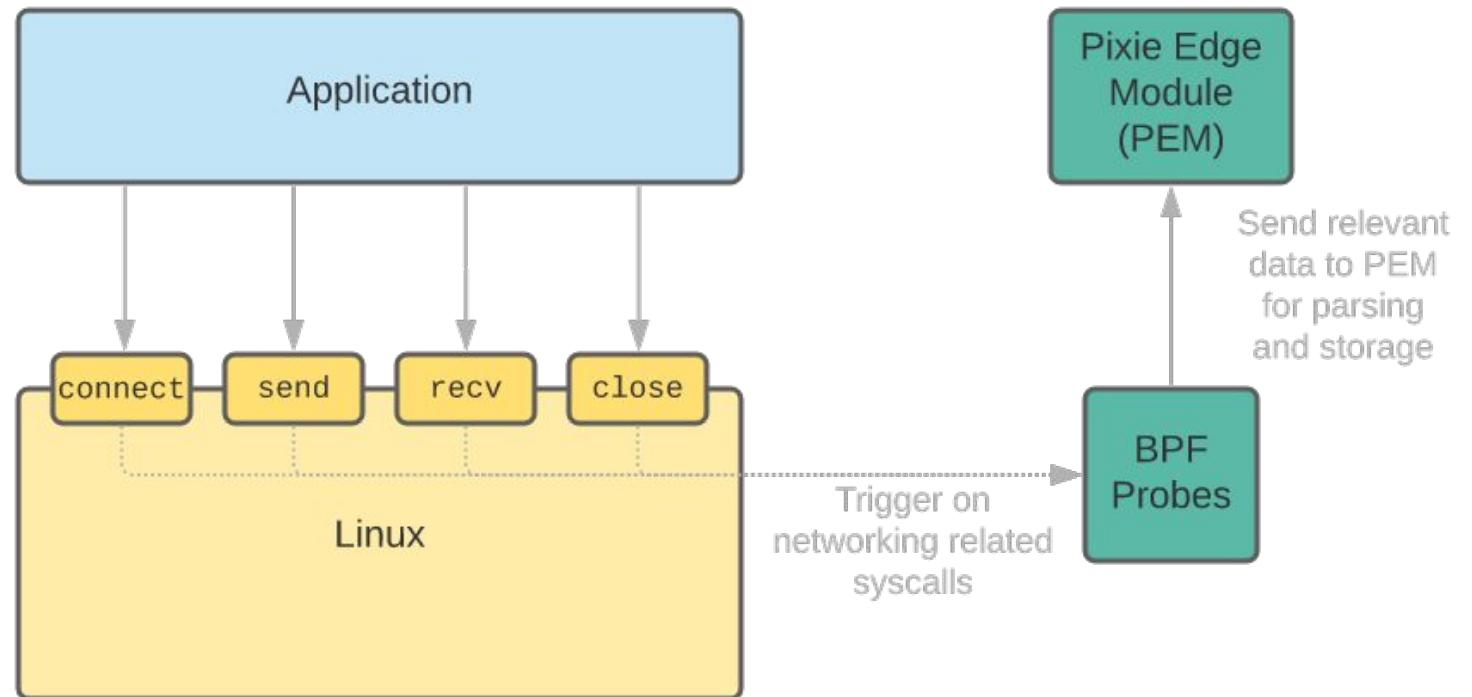
What if we could automatically capture telemetry data?



Pixie Auto-Telemetry

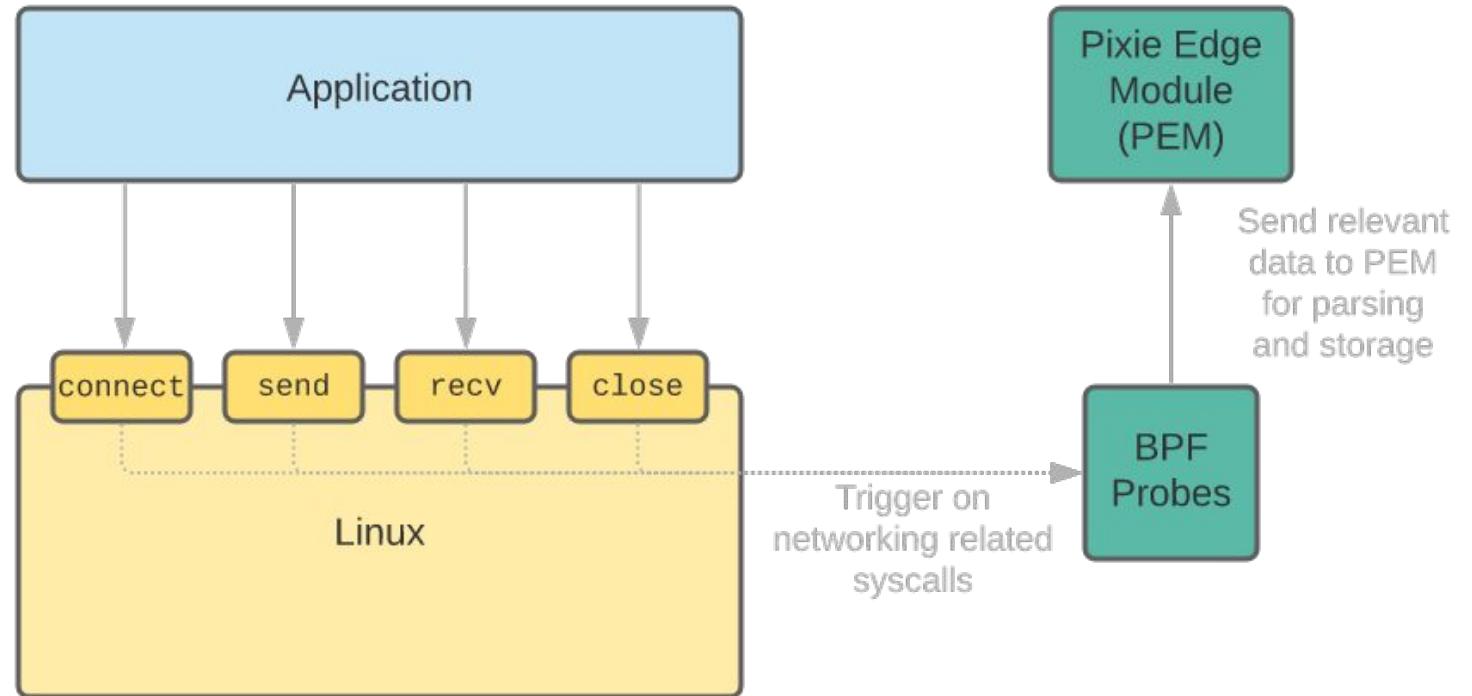


What if we could automatically capture telemetry data?

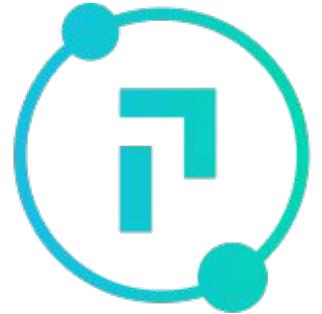




- What if we could automatically capture telemetry data using eBPF?
- Pixie uses **eBPF** to capture:
 - Full-body requests
 - Network metrics
 - Application CPU profiles



Metrics Providers



PIXIE

VS



Prometheus

Kubernetes Observability



CNCF GRADUATED PROJECT

Logging



Prometheus

CNCF GRADUATED PROJECT

Metrics



OpenTelemetry

CNCF INCUBATING PROJECT

Metrics, Distributed
Traces, Logs



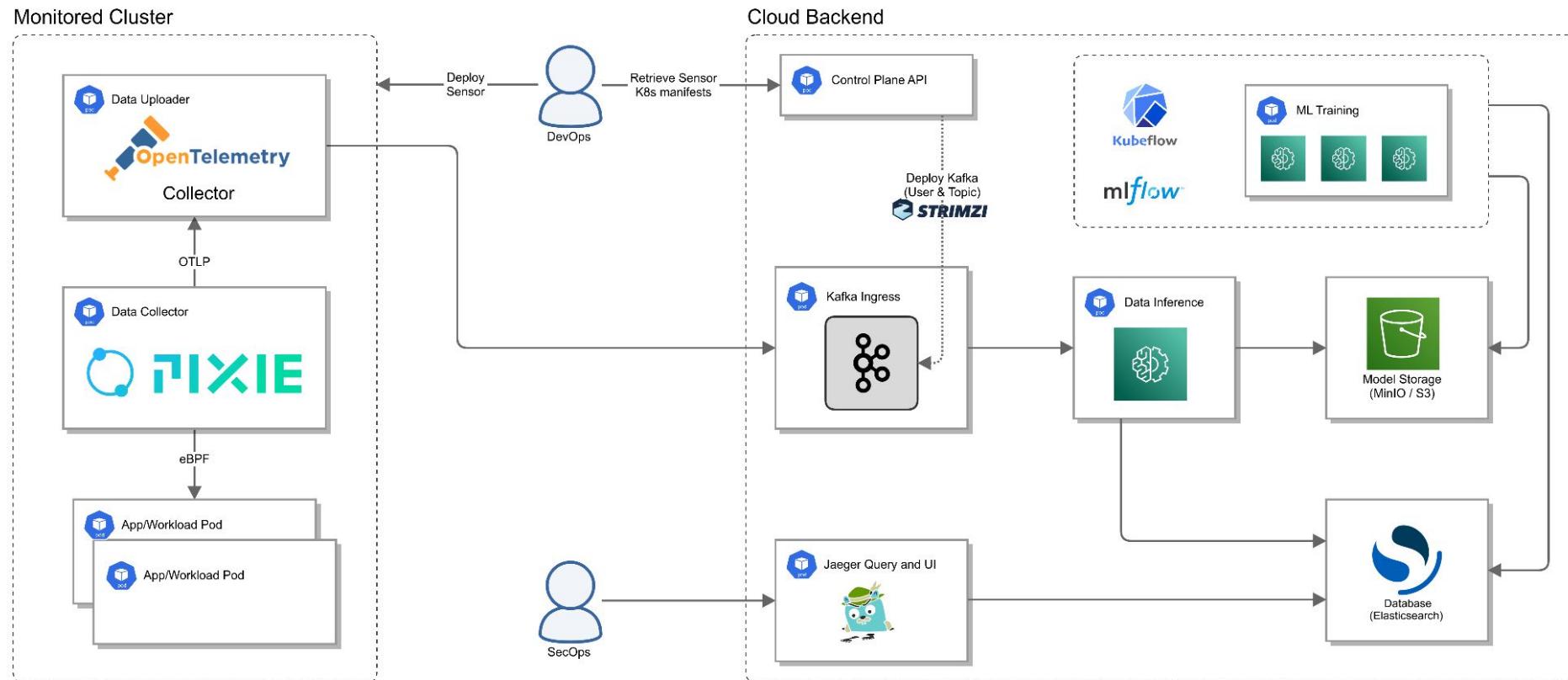
PIXIE

CNCF SANDBOX PROJECT

Metrics, Traces,
Application Profiles

Open Source And Standards

- Integrate with existing workflows: unlock extensive ecosystem of tools, pipelines, SDKs, etc.
- Example use-case:
 - VMware xLabs *Project Trinidad*: anomaly detection for security monitoring
 - Pixie → OpenTelemetry Collector → Kafka → Jaeger & Elasticsearch, Kubeflow & mlflow



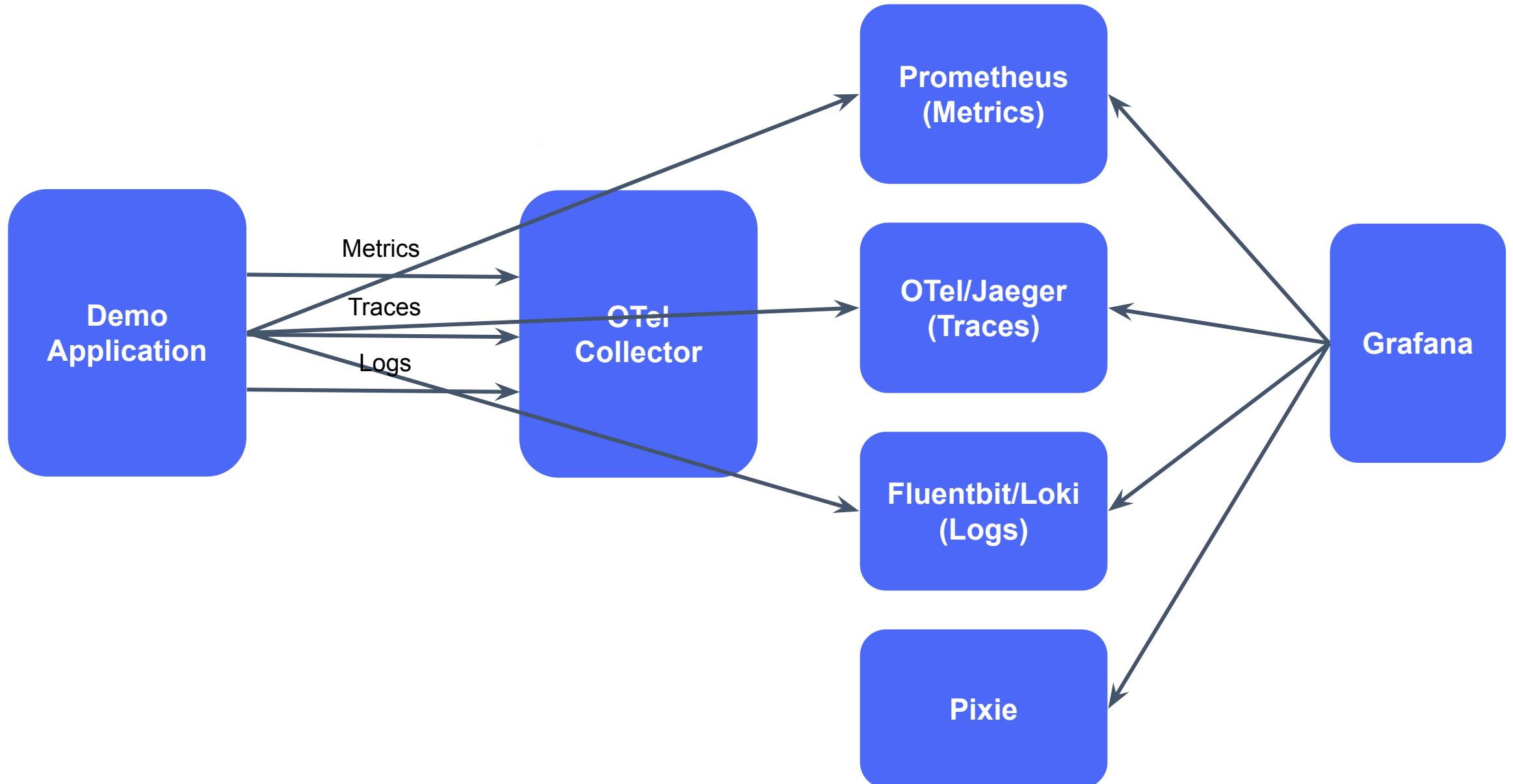
Tutorial - Getting Started

You will first need to setup your Kubernetes cluster, demo apps, and tools.

Instructions are located at:



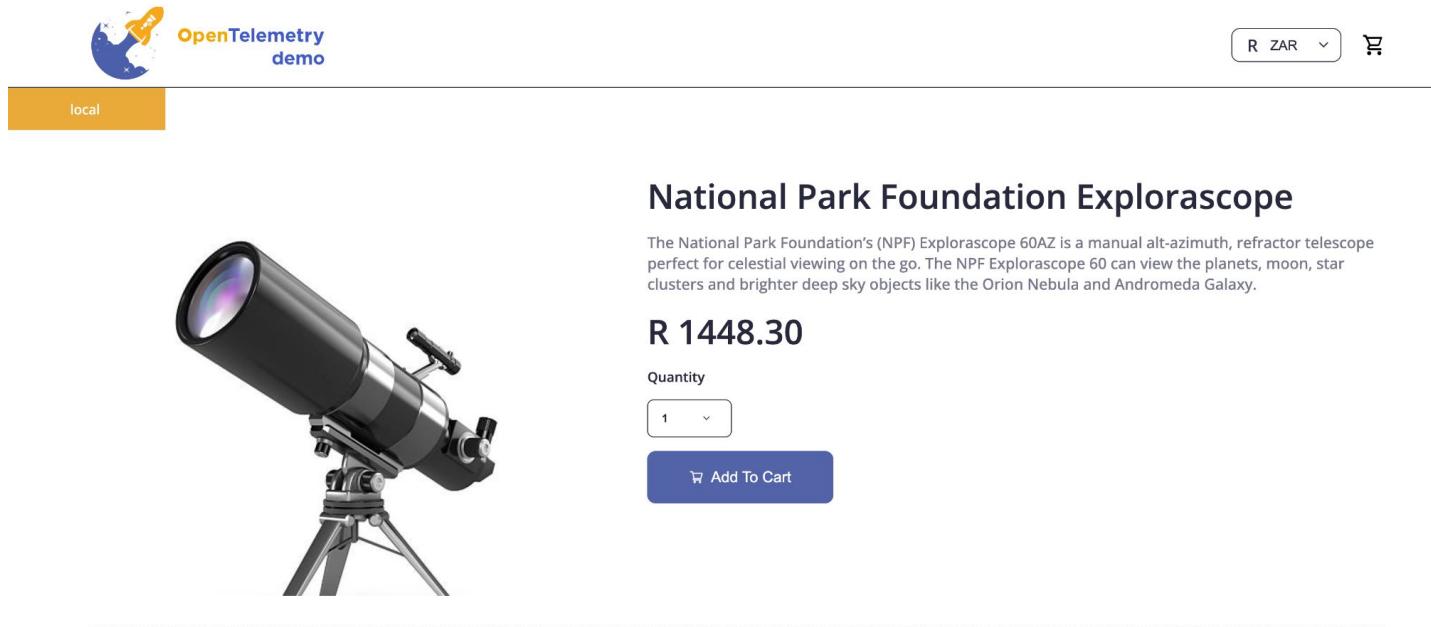
<https://github.com/pixie-labs/opentelemetry-helm-charts>



Tutorial - Demo App

We will be using “OpenTelemetry Demo”, a simple e-commerce website which sells telescopes.

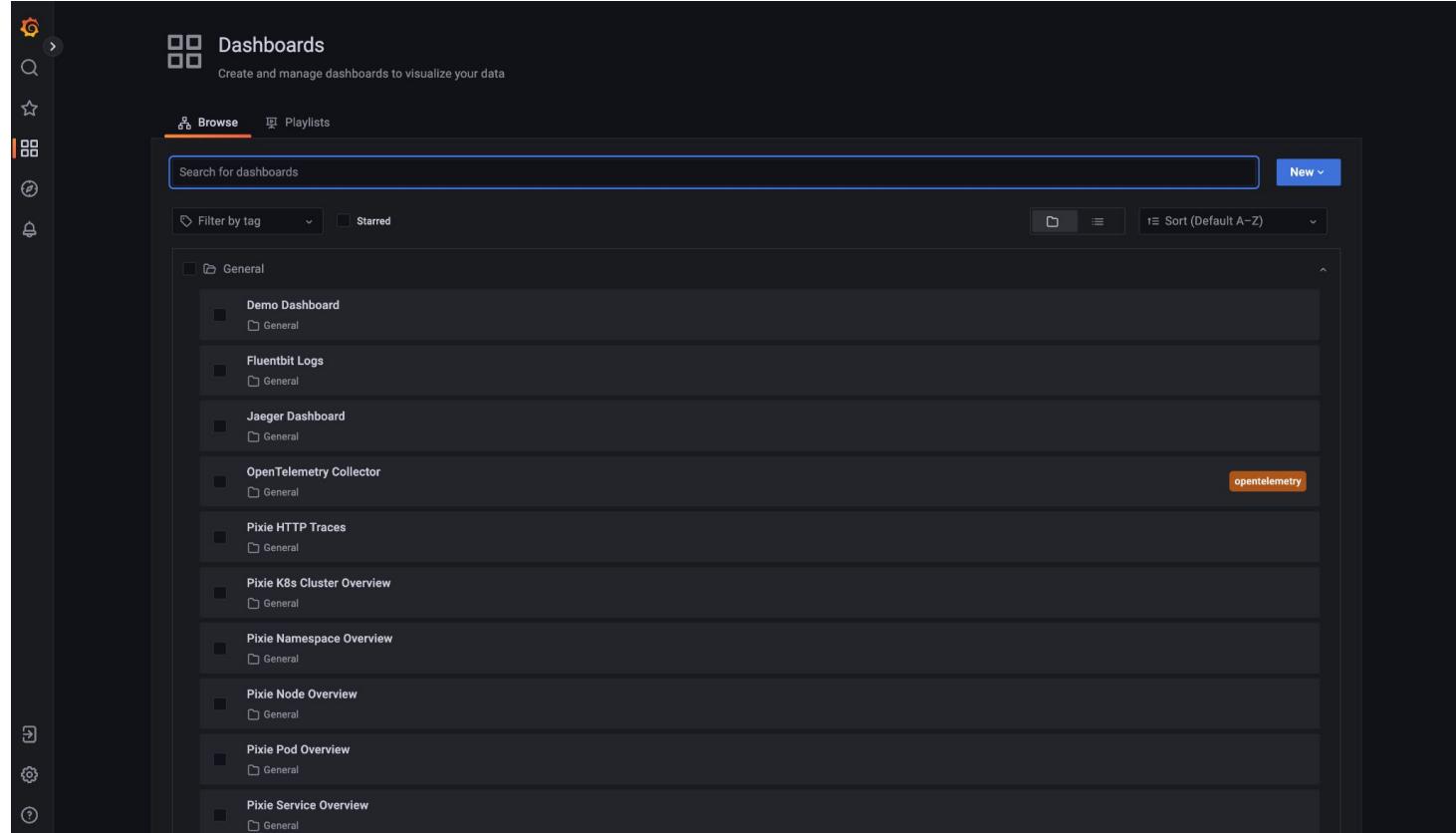
You can access it by running the port-forward and going to <http://localhost:8080/>.



You May Also Like

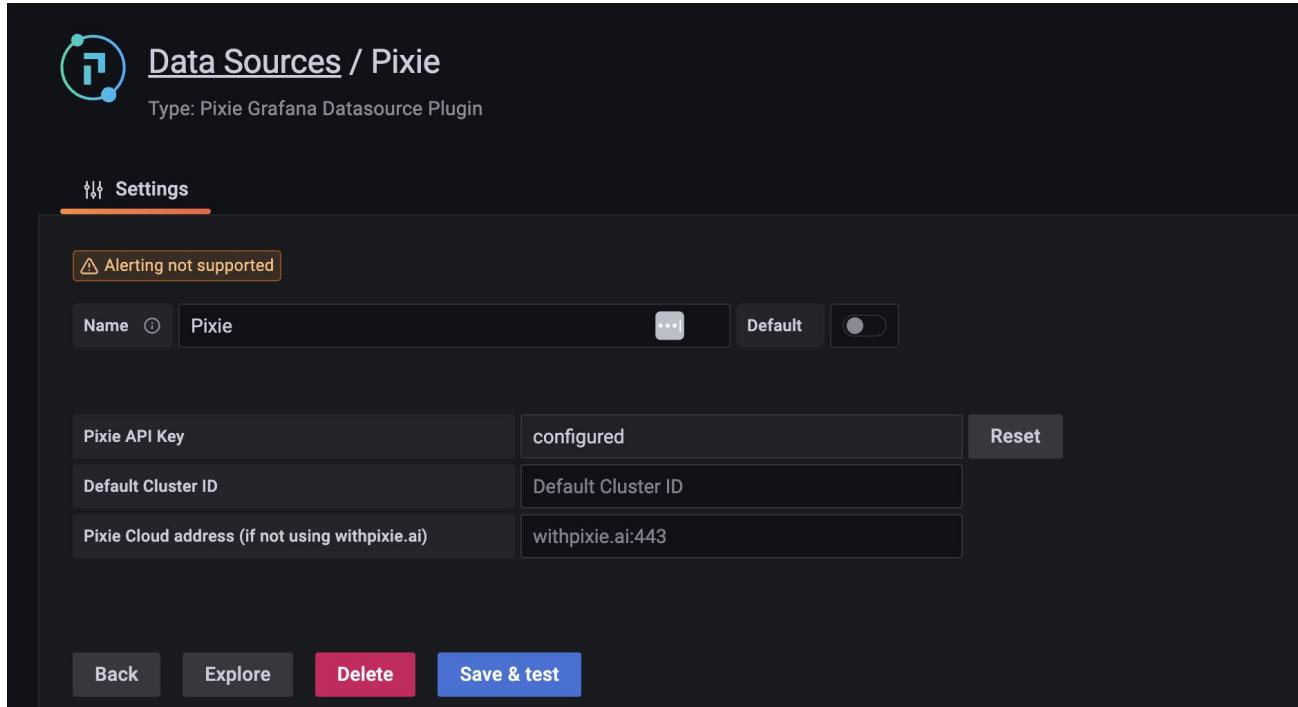


Tutorial - Grafana



Using the OTel collector, we are sending our telemetry data to Grafana.
You can see the Grafana dashboards at: <http://localhost:8080/grafana>.

Tutorial - Grafana



Add your Pixie API key to the Pixie Grafana Datasource:

```
px api-key create --short
```

Copy and paste it into <http://localhost:8080/grafana/datasources/edit/pixie> and click
“Save and Test”

Tutorial - Logs (Fluentbit)

The screenshot shows a log viewer interface with the title "General / Fluentbit Logs". The logs are displayed in a table format with columns for timestamp, log level, file, and message. A red vertical bar highlights the third log entry, which contains an error message about DNS resolution failing for otelcol:4317.

Time	Level	File	Message
2023-04-14 12:56:18	INFO	stdout	"log": "Convert conversion successful\\n", "stream": "stdout", "time": "2023-04-14T19:56:18.650324929Z"
2023-04-14 12:56:18	INFO	stdout	"log": "[Error] File: /opentelemetry-cpp/exporters/otlp/src/otlp_grpc_metric_exporter.cc:74 [OTLP METRIC GRPC Exporter] Export() failed: DNS resolution failed for otelcol:4317: C-ares status is 8", "stream": "stdout", "time": "2023-04-14T19:56:18.65026014Z"
2023-04-14 12:56:18	INFO	stdout	"log": "otel/opentelemetry-collector-contrib@sha256:be76a0ac95d3c1ca39cf15ee5932bf53ea1d0d67e70feb9444a54074de6eb129", "stream": "stdout", "time": "2023-04-14T19:56:18.65026014Z"

Tutorial - Logs (Fluentbit)

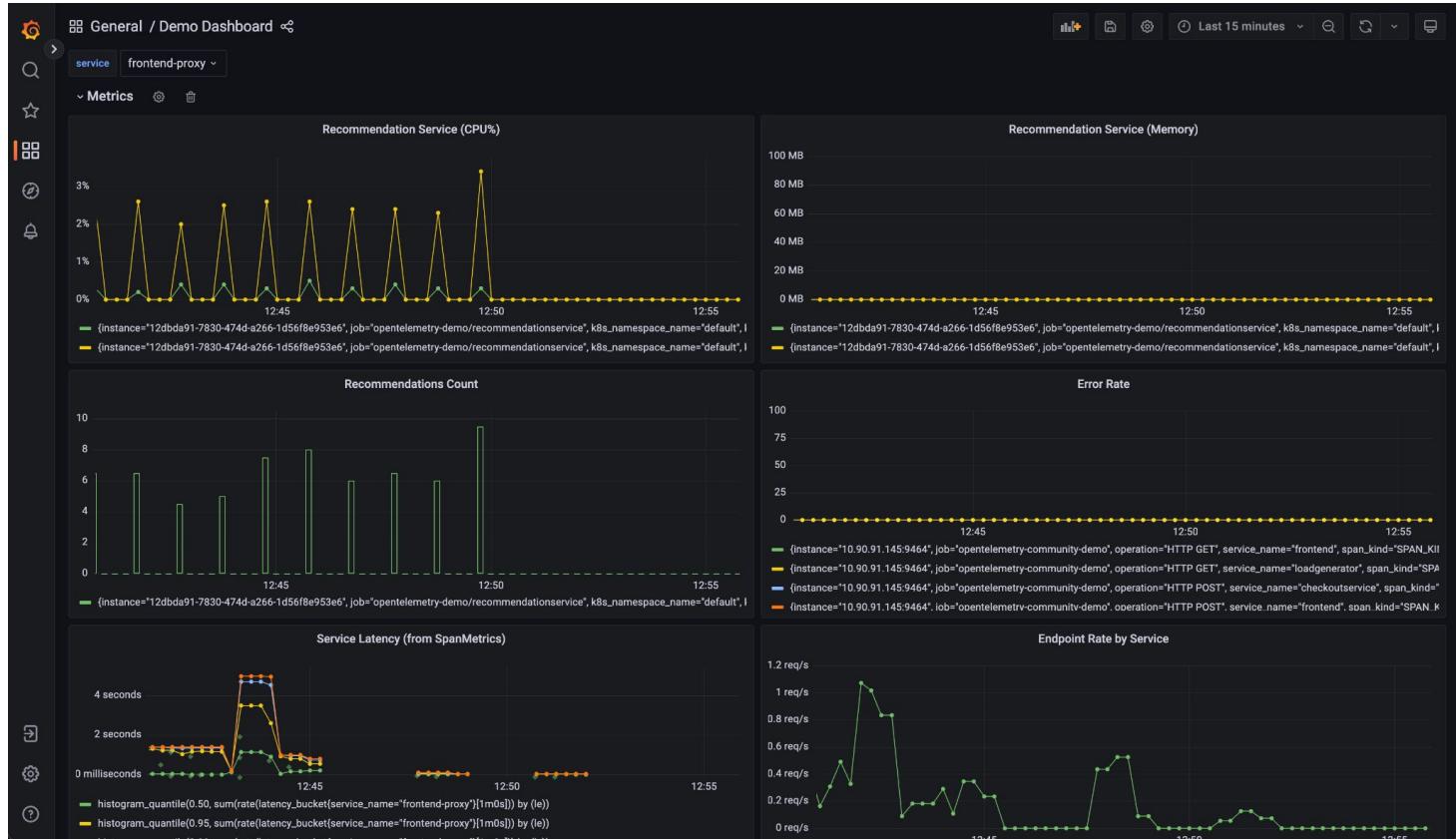
```
[FILTER]
  Name      grep
  Match    kube./*
  Exclude  kubernetes_namespace_name kube-system

[FILTER]
  Name      grep
  Match    kube./*
  Exclude  kubernetes_namespace_name pl

[FILTER]
  Name      grep
  Match    kube./*
  Exclude  kubernetes_container_name loki-canary

[OUTPUT]
  Name      opentelemetry
  Match    kube./*
  Host     my-otel-demo-otelcol
  Port     4318
  Metrics_uri /v1/metrics
  Logs_uri  /v1/logs
  Traces_uri /v1/traces
  Log_response_payload True
  Tls       Off
  Tls.verify Off
# add user-defined labels
add_label   app fluent-bit
add_label   color blue
```

Tutorial - Metrics (Prometheus)



1. Click into “Prometheus Dashboard”.
2. Compare the metrics between the different services by updating the service filter in the top left.

Tutorial - Metrics (Prometheus)

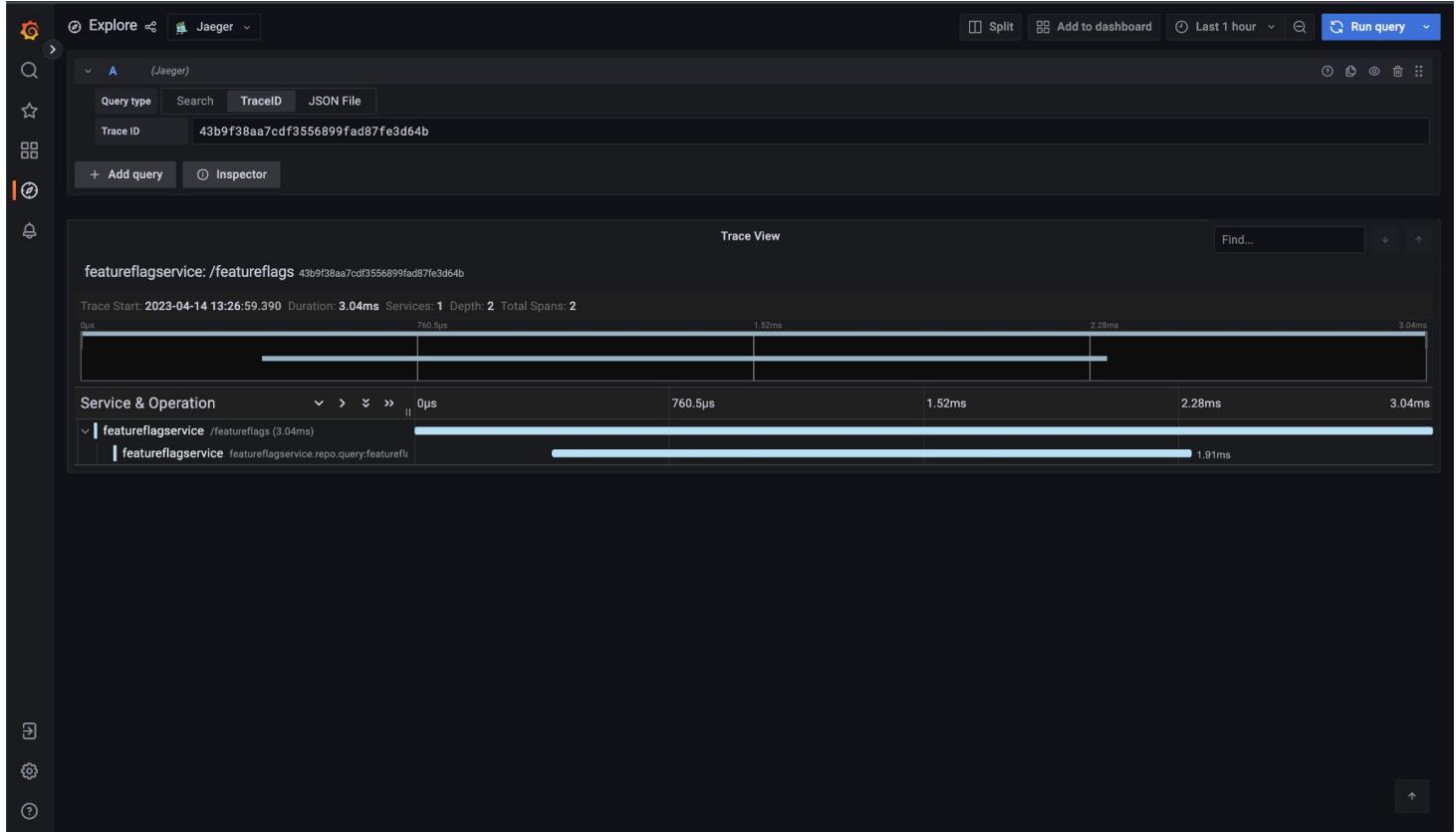
```
class RecommendationService(demo_pb2_grpc.RecommendationServiceServicer):
    def ListRecommendations(self, request, context):
        prod_list = get_product_list(request.product_ids)
        span = trace.get_current_span()
        span.set_attribute("app.products_recommended.count", len(prod_list))
        logger.info(f"Receive ListRecommendations for product ids:{prod_list}")

        # build and return response
        response = demo_pb2.ListRecommendationsResponse()
        response.product_ids.extend(prod_list)

        # Collect metrics for this service
        rec_svc_metrics["app_recommendations_counter"].add(len(prod_list), {'recommendation.type': 'catalog'})

    return response
```

Tutorial - Traces (OTel)



Look through the traces—how can you find out which product is resulting in an error?

Tutorial - Traces (OTel)

```
func (p *productCatalog) GetProduct(ctx context.Context, req *pb.GetProductRequest) (*pb.Product, error) {
    span := trace.SpanFromContext(ctx)
    span.SetAttributes(
        attribute.String("app.product.id", req.Id),
    )

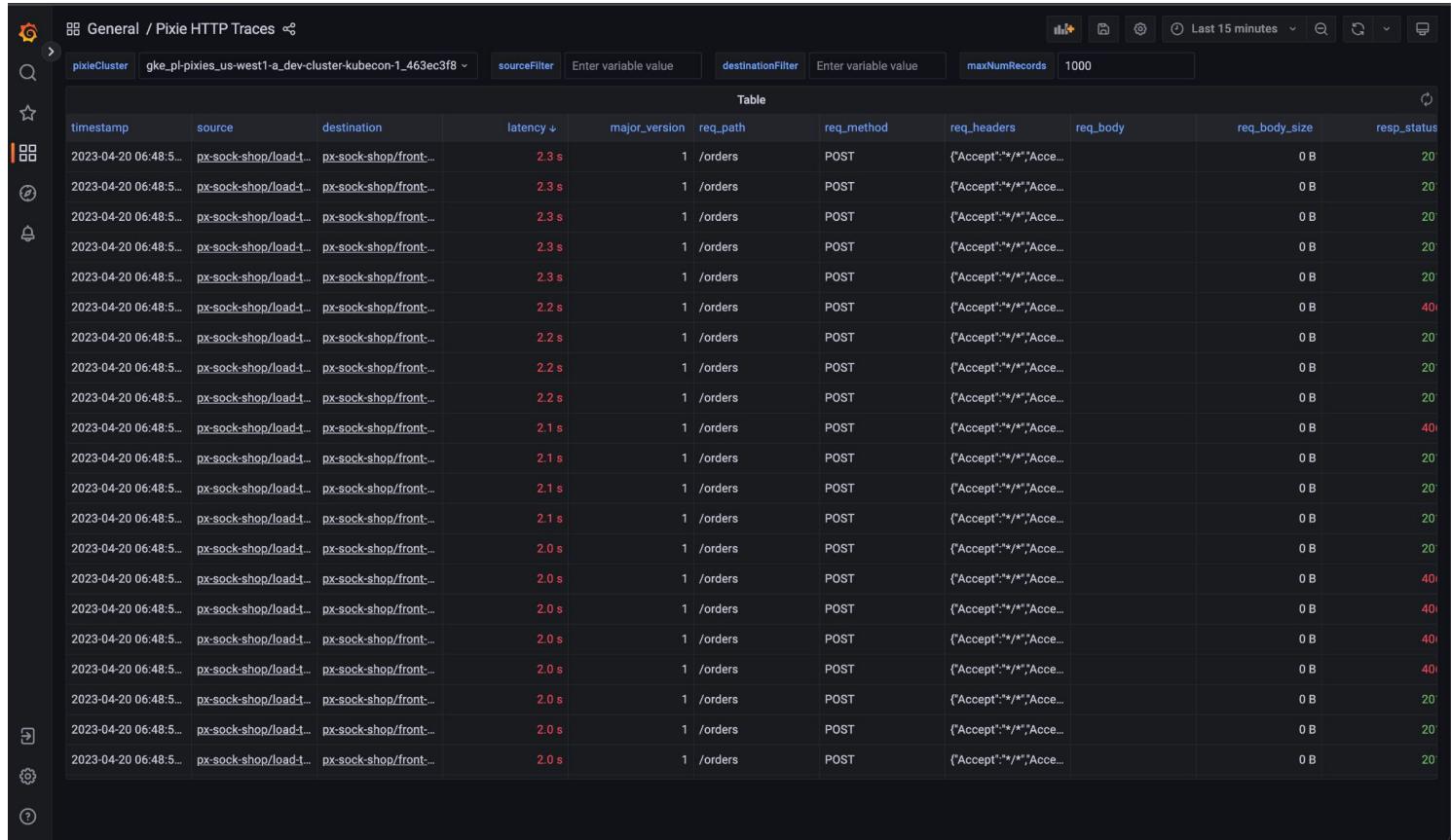
    // GetProduct will fail on a specific product when feature flag is enabled
    if p.checkProductFailure(ctx, req.Id) {
        msg := fmt.Sprintf("Error: ProductCatalogService Fail Feature Flag Enabled")
        span.SetStatus(otelcodes.Error, msg)
        span.AddEvent(msg)
        return nil, status.Errorf(codes.Internal, msg)
    }

    var found *pb.Product
    for _, product := range catalog {
        if req.Id == product.Id {
            found = product
            break
        }
    }

    if found == nil {
        msg := fmt.Sprintf("Product Not Found: %s", req.Id)
        span.SetStatus(otelcodes.Error, msg)
        span.AddEvent(msg)
        return nil, status.Errorf(codes.NotFound, msg)
    }

    msg := fmt.Sprintf("Product Found - ID: %s, Name: %s", req.Id, found.Name)
    span.AddEvent(msg)
    span.SetAttributes(
        attribute.String("app.product.name", found.Name),
    )
    return found, nil
}
```

Tutorial - Pixie



The screenshot shows a detailed view of network traffic captured by Pixie. The interface includes a header with cluster selection (pixieCluster, gke_p1-pixies_us-west1-a_dev-cluster-kubecon-1_463ec3f8), search/filtering fields, and a timestamp range selector (Last 15 minutes). The main area is a table titled "Table" with columns: timestamp, source, destination, latency (sorted), major_version, req_path, req_method, req_headers, req_body, req_body_size, and resp_status. The data consists of approximately 20 rows of POST requests from 'px-sock-shop/load-t...' to 'px-sock-shop/front...' with a latency of 2.0-2.3 seconds, major_version 1, req_path '/orders', and various response status codes (200, 401).

timestamp	source	destination	latency ↓	major_version	req_path	req_method	req_headers	req_body	req_body_size	resp_status
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.3 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.3 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.3 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.3 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.3 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.2 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	401
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.2 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.2 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.2 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.1 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	401
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.1 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.1 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.1 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	401
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	401
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	401
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	401
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20
2023-04-20 06:48:5...	px-sock-shop/load-t...	px-sock-shop/front...	2.0 s	1	/orders	POST	{"Accept": "*/*", "Acce...		0 B	20

Look at Pixie's HTTP Traces dashboard. Can you see the GetProduct requests from earlier?

Tutorial - Pixie

The screenshot shows the Pixie UI interface for a cluster named `gke:kubeconeu-1`. The main view is a table titled "Table" displaying network traffic metrics. The columns include TIME_, SOURCE, DESTIN..., LATENCY, MAJOR..., REQ_PA..., REQ_M..., REQ_HE..., REQ_BO..., REQ_BO..., RESP_S..., RE, and DETAILS. The table lists 15 records of traffic from the `default/my-otel-demo` namespace to the `default/my-otel-demo-product` service. The latency values range from `375 µs` to `534.2 µs`. The response status codes are all `200 OK`. The "DETAILS" column provides a detailed JSON representation of each request, including headers like `baggage`, `content-type`, and `grpc-accept-encoding`, and the response body which includes an error message about failed protobuf parsing.

TIME_	SOURCE	DESTIN...	LATENCY	MAJOR...	REQ_PA...	REQ_M...	REQ_HE...	REQ_BO...	REQ_BO...	RESP_S...	RE	DETAILS
4/20/2023...	default/my...	default/my...	<code>375 µs</code>	2	/oteldemo...	POST	{ :author...	1:"L9ECAV...	17 B	<code>200</code>	OK	<pre>baggage: synthetic_request=true, content-type: application/grpc, grpc-accept-encoding: identity,deflate,gzip, te: trailers, traceparent: 00-db1f37c854173b50781f5 5ee0aed844d-58244c55436c2a23-01, user-agent: grpc-node-js/1.6.7 }, req_body: 1: "OLJCESPC7Z", req_body_size: 17, resp_status: 200, resp_message: OK, resp_headers: { :status: 200, content-type: application/grpc, grpc-message: Error: ProductCatalogService Fail Feature Flag Enabled, grpc-status: 13 }, resp_body: <Failed to parse protobuf>, resp_body_size: 0 }</pre>
4/20/2023...	default/my...	default/my...	<code>682.9 µs</code>	2	/oteldemo...	POST	{ :author...	1:"66VCH...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>383.7 µs</code>	2	/oteldemo...	POST	{ :author...	1:"9SIQT8...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>392.8 µs</code>	2	/oteldemo...	POST	{ :author...	1:"6E9ZZ...	17 B	<code>200</code>	OK	
> 4/20/2023...	default/my...	default/my...	<code>12.6 ms</code>	2	/oteldemo...	POST	{ :author...	1:"OLJCES...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>394.8 µs</code>	2	/oteldemo...	POST	{ :author...	1:"66VCH...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>446 µs</code>	2	/oteldemo...	POST	{ :author...	1:"LS4PSX...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>324.8 µs</code>	2	/oteldemo...	POST	{ :author...	1:"L9ECAV...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>363.9 µs</code>	2	/oteldemo...	POST	{ :author...	1:"ZYFJ3...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>311 µs</code>	2	/oteldemo...	POST	{ :author...	1:"L9ECAV...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>435.1 µs</code>	2	/oteldemo...	POST	{ :author...	1:"ZYFJ3...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>439 µs</code>	2	/oteldemo...	POST	{ :author...	1:"L9ECAV...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>534.2 µs</code>	2	/oteldemo...	POST	{ :author...	1:"ZYFJ3...	17 B	<code>200</code>	OK	
4/20/2023...	default/my...	default/my...	<code>341.3 µs</code>	2	/oteldemo...	POST	{ :author...	1:"ZYFJ3...	5 B	<code>200</code>	OK	

More visualizations and data are available in the Pixie UI!

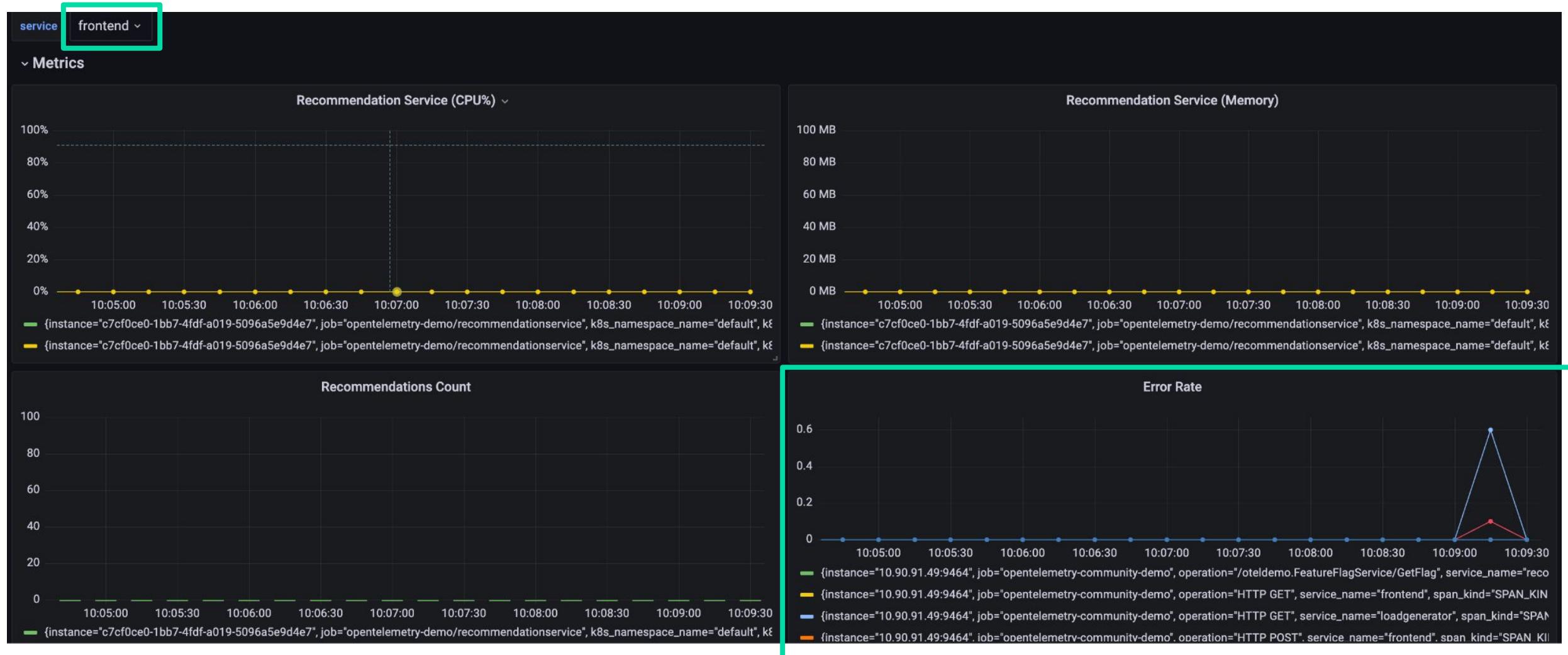
Tutorial

You can go to <http://localhost:8080/feature> to simulate a scenario in your demo app.

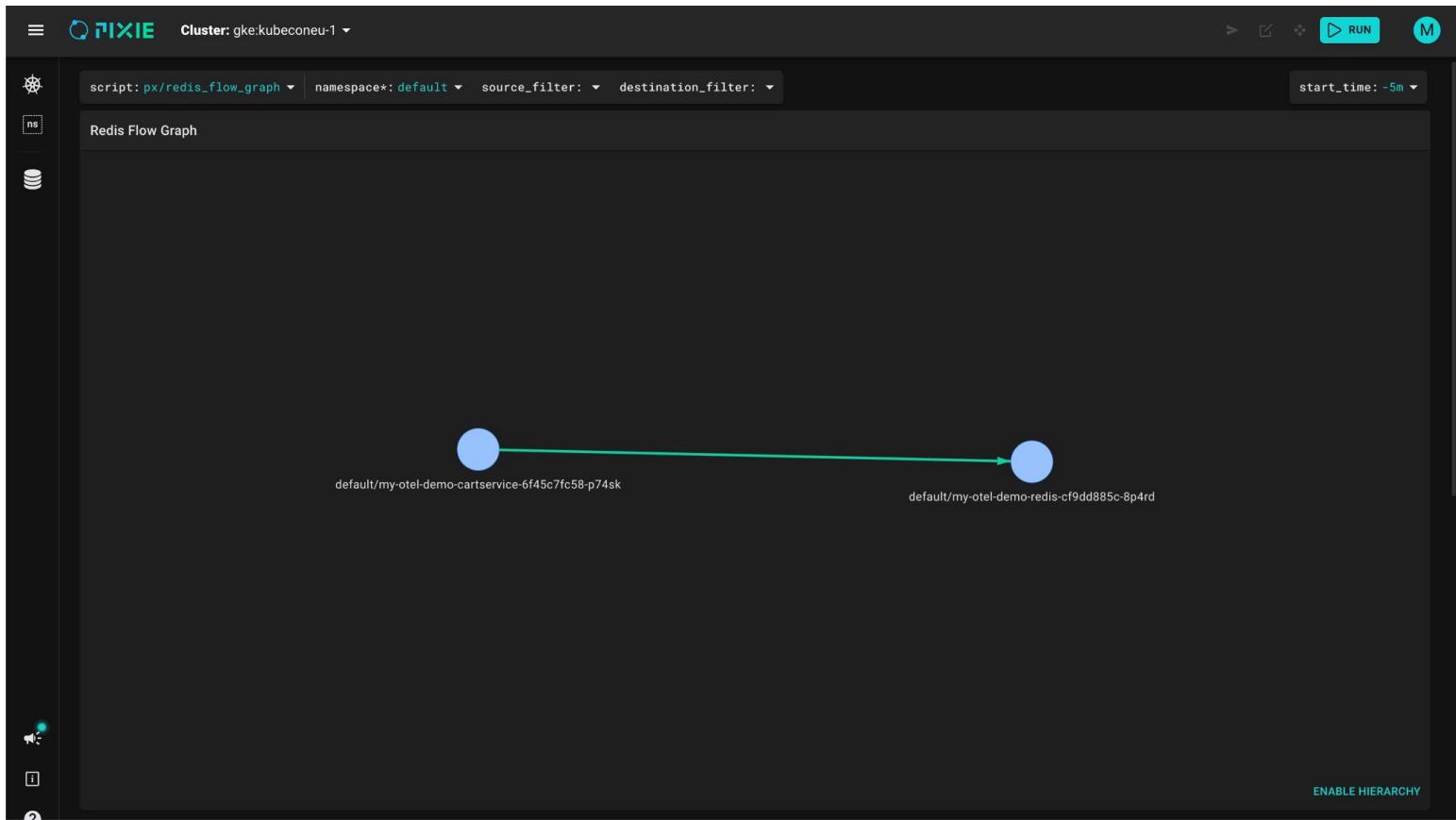
Enable 'productCatalogFailure'.

- How do the dashboards compare before and after you simulated the productCatalogFailure?
- What telemetry data can you use to discover this issue is occurring?

Tutorial



Tutorial - Pixie



Pixie supports many different protocols. Can you find which service makes requests to Redis?

Tutorial - Traces (OTel)

productcatalogservice oteldemo.ProductCatalogService/C 13.6ms

oteldemo.ProductCatalogService/GetProduct Service: productcatalogservice Duration: 13.6ms Start Time: 18.7ms (10:13:17.822) Child Count: 1

> Attributes: app.product.id = OLJCESPC7Z | error = true | internal.span.format = proto | net.peer.ip = 10.90.91.44 | net.peer.port = 37264 | otel.library.name = go.opentelemetry.io/co...
> Resource: container.id = 69fb7077f3a3dd411e9426c4cfda33bdedf9368c71e1773726f64e5fae5827 | host.name = my-otel-demo-productcatalogservice-86bb44c476-m94g5 | k8s...
▼ Events (3)
> 18.7ms: event = message | message.id = 1 | message.type = RECEIVED | message.uncompressed_size = 12
> 32.29ms: event = Error: ProductCatalogService Fail Feature Flag Enabled
> 32.3ms: event = message | message.id = 1 | message.type = SENT | message.uncompressed_size = 58
Log timestamps are relative to the start time of the full trace.
▼ Warnings (1)
invalid parent span IDs=8329039633a9469b; skipping clock skew adjustment

SpanID: 4aaabd105b593053



Please scan the QR Code above
to leave feedback on this session