Starburst

Observing Trino

Subtitle Goes Here

Matt Stephenson
06.13.2024

Agenda

Observing Trino

- OpenMetrics
- OpenTelemetry
- Airlift Logging
- Deployment
- Metrics (Prometheus/Grafana)
- Logging (OpenSearch/Kibana)
- Tracing (Jaeger)

OpenMetrics

- Open standard for pull metrics
- Originally from the Prometheus project
- Complex data structures like historgrams
- Fully supported in OpenTelemetry and other 3rd party tools
- Added support to Airlift
- Instantiate the **`JmxOpenMetricsModule`**
- Relies on JmxUtils
- No configuration necessary
- Exposed on the **`/metrics`** endpoint

OpenTelemetry

- Open standard for Traces, Metrics, and Logs/Events
- Currently we just use it for Traces in Trino
- Protocol for Traces is push via GRPC
- Extensive ecosystem including support for both OSS and 3rd party tools (Jaeger, Datadog, etc)
- Added support to Airlift
- Instantiate the **`OpenTelemetryModule`**
- Uses OSS OpenTelemetry SDK for creating spans and traces in code

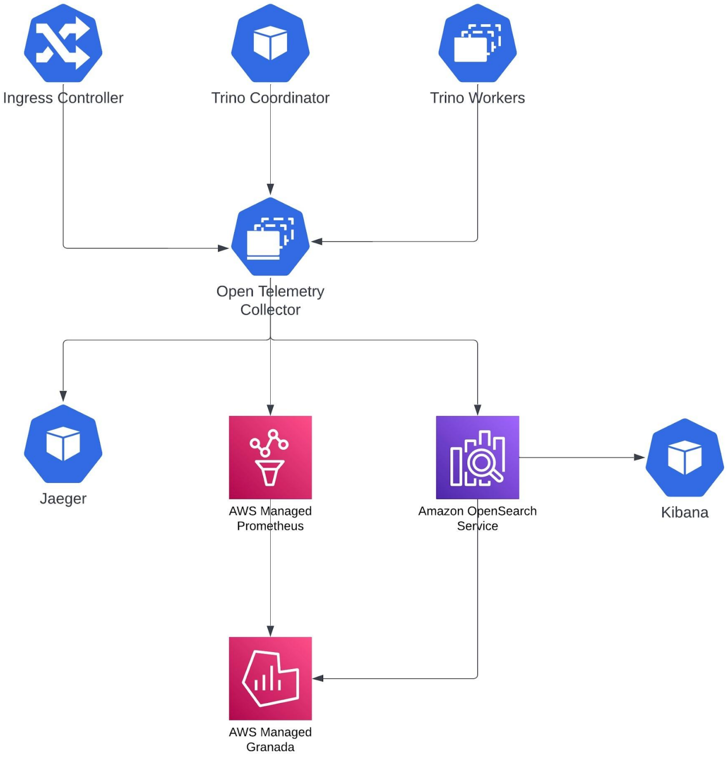
```
tracing.enabled=true

tracing.exporter.endpoint=http://observe.example.com:4317
```

Airlift logging

- Relies on Java Util Logging
- Very simple, straightforward api
- Added support for logging over a TCP channel from Trino or any Airlift application
- Easily integrated with Fluentbit or the OpenTelemetry collector
- Supports static structured logging fields via **`node.annotation-file`**
- Any log file can be set to a TCP channel by specifying **`tcp://<ip>:<port>`**

Kubernetes Deployment



Starburst

Demo

https://github.com/mattstep/pulumi-trino-java

Starburst

Thank you!
Q&A

Matt Stephenson