Observability in Keycloak:

Where does it hurt?

CLOUD NATIVE
COMPUTING FOUNDATION

$ whoami

Martin Bartoš

Senior Software Engineer

KEYCLOAK
Cloud Native team

IBM

@mabartos

@mabartos98

What is **Observability**?

Observability

- **Observability** is the ability to understand the **internal state** or condition of a system based on **knowledge** of its **external outputs** (specifically its **telemetry**)

Observability

- **Observability** is the ability to understand the **internal state** or condition of a system based on **knowledge** of its **external outputs** (specifically its **telemetry**)

Observability

- **Observability** is the ability to understand the **internal state** or condition of a system based on **knowledge** of its **external outputs** (specifically its **telemetry**)

  => **Observability** means being able to **monitor, trace, and understand** what **Keycloak is doing** in real-time

Observability

- **Observability** is the ability to understand the **internal state** or condition of a system based on **knowledge** of its **external outputs** (specifically its **telemetry**)

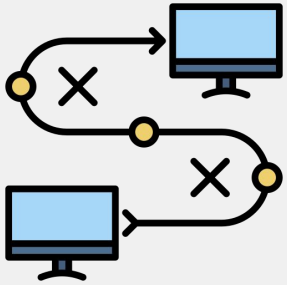
  => **Observability** means being able to **monitor, trace, and understand** what **Keycloak is doing** in real-time

- **Keycloak** emits telemetry data about its **behavior**, **performance** and **interactions** with other systems

What Keycloak exposes?

Logs

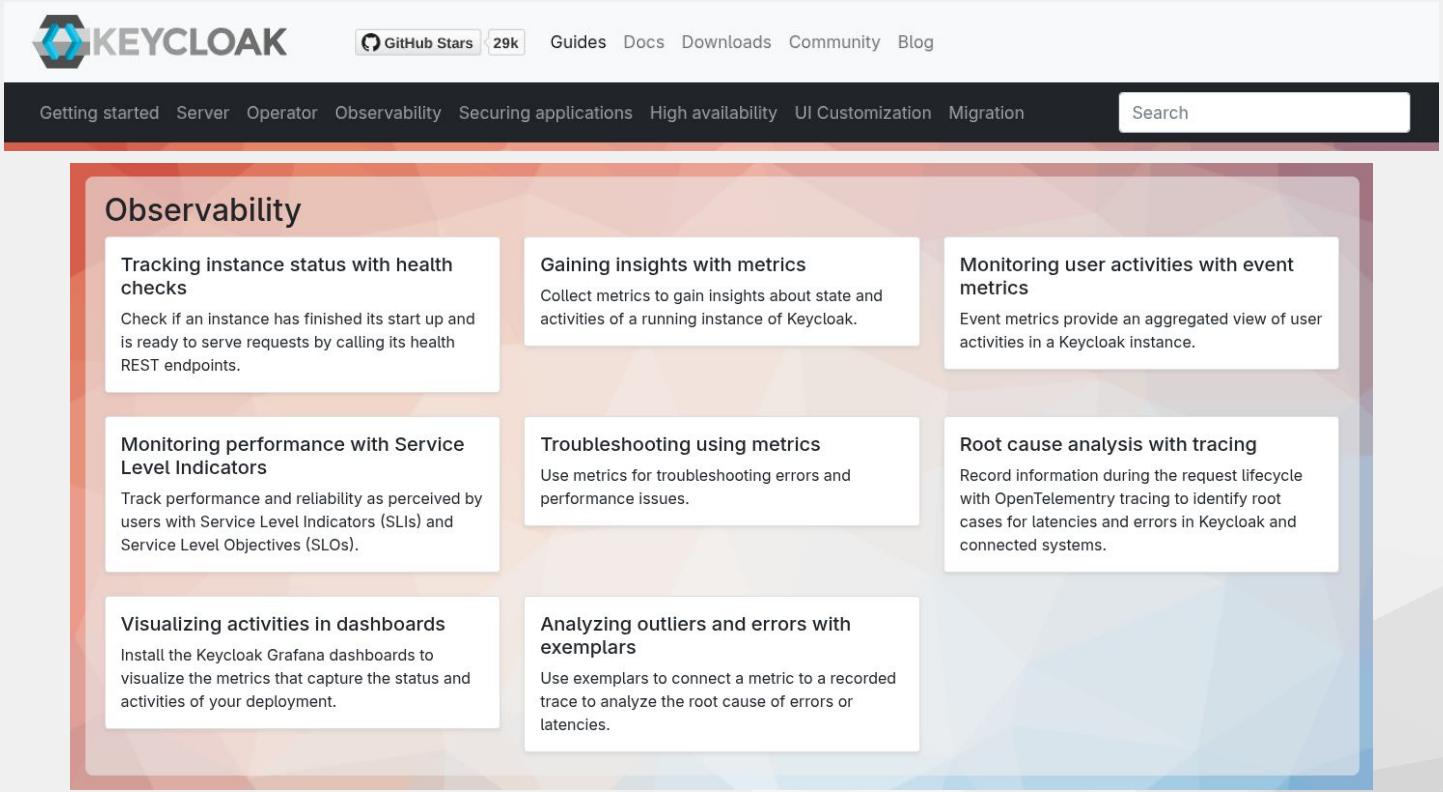
Metrics



Traces

Observability guides

www.keycloak.org



Logs

Logs

- **Logs** are **records of events** happening in a system

```
2025-08-20 10:44:19,982 WARN  [org.keycloak.events] (executor-thread-2) type="LOGIN_ERROR", realmId="5fa
-5e1f-437d-92b3-9195e6c10ae3", ipAddress="127.0.0.1", error="invalid_user_credentials", auth_method="ope
-226c-45d1-9d2c-63980881cb1c", username="admin"
```

Logs

- **Logs** are **records of events** happening in a system
- Keycloak **exports logs** to different sources (console, file, syslog)

```
2025-08-20 10:44:19,982 WARN  [org.keycloak.events] (executor-thread-2) type="LOGIN_ERROR", realmId="5fa
-5e1f-437d-92b3-9195e6c10ae3", ipAddress="127.0.0.1", error="invalid_user_credentials", auth_method="ope
-226c-45d1-9d2c-63980881cb1c", username="admin"
```

Logs

- **Logs** are **records of events** happening in a system
- Keycloak **exports logs** to different sources (console, file, syslog)
- Can be **exported as JSON** (*also Elastic Common Schema support*)

```
2025-08-20 10:44:19,982 WARN  [org.keycloak.events] (executor-thread-2) type="LOGIN_ERROR", realmId="5fa
-5e1f-437d-92b3-9195e6c10ae3", ipAddress="127.0.0.1", error="invalid_user_credentials", auth_method="ope
-226c-45d1-9d2c-63980881cb1c", username="admin"
```

Logs

- **Logs** are **records of events** happening in a system
- Keycloak **exports logs** to different sources (console, file, syslog)
- Can be **exported as JSON** (*also Elastic Common Schema support*)
- **Asynchronous logging** support for higher throughput and lower latency

```
2025-08-20 10:44:19,982 WARN  [org.keycloak.events] (executor-thread-2) type="LOGIN_ERROR", realmId="5fa
-5e1f-437d-92b3-9195e6c10ae3", ipAddress="127.0.0.1", error="invalid_user_credentials", auth_method="ope
-226c-45d1-9d2c-63980881cb1c", username="admin"
```

Metrics

Metrics

- **Metrics** are **quantitative measurements** that track the behavior and performance of an application

Metrics

- **Metrics** are **quantitative measurements** that track the behavior and performance of an application
- Keycloak **gathers various metrics**

List of Keycloak key metrics

- Self-provided metrics
- JVM metrics
- Database Metrics
- HTTP metrics
- Single site metrics (without external Infinispan)
  - Clustering metrics
  - Embedded Infinispan metrics for single site deployments
- Multiple sites metrics (as described in Multi-site deployments)
  - Embedded Infinispan metrics for multi-site deployments
  - External Infinispan metrics

Metrics

- **Metrics** are **quantitative measurements** that track the behavior and performance of an application
- Keycloak **gathers various metrics**
- Keycloak uses **Micrometer** to expose metrics on the `/metrics` **endpoint**

List of Keycloak key metrics

- Self-provided metrics
- JVM metrics
- Database Metrics
- HTTP metrics
- Single site metrics (without external Infinispan)
  - Clustering metrics
  - Embedded Infinispan metrics for single site deployments
- Multiple sites metrics (as described in Multi-site deployments)
  - Embedded Infinispan metrics for multi-site deployments
  - External Infinispan metrics

Metrics

- **Metrics** are **quantitative measurements** that track the behavior and performance of an application
- Keycloak **gathers various metrics**
- Keycloak uses **Micrometer** to expose metrics on the `/metrics` **endpoint**
- Endpoint `/metrics` lives on the **management port** (port 9000)

List of Keycloak key metrics

- Self-provided metrics
- JVM metrics
- Database Metrics
- HTTP metrics
- Single site metrics (without external Infinispan)
  - Clustering metrics
  - Embedded Infinispan metrics for single site deployments
- Multiple sites metrics (as described in Multi-site deployments)
  - Embedded Infinispan metrics for multi-site deployments
  - External Infinispan metrics

Metrics - **response**

Metrics - response

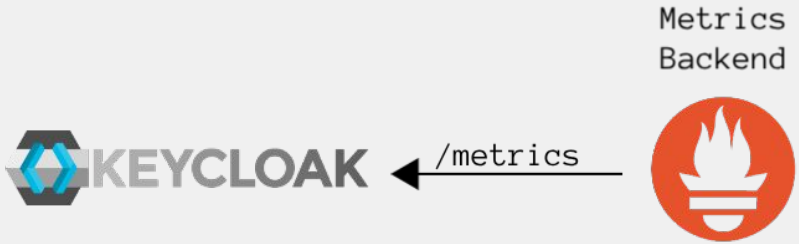
- The response from the metrics endpoint uses `application/openmetrics-text` content type

```
# HELP base_gc_total Displays the total number of collections that have occurred. T
# TYPE base_gc_total counter
base_gc_total{name="G1 Young Generation",} 14.0
# HELP jvm_memory_usage_after_gc_percent The percentage of long-lived heap pool use
# TYPE jvm_memory_usage_after_gc_percent gauge
jvm_memory_usage_after_gc_percent{area="heap",pool="long-lived",} 0.0
# HELP jvm_threads_peak_threads The peak live thread count since the Java virtual m
# TYPE jvm_threads_peak_threads gauge
jvm_threads_peak_threads 113.0
# HELP agroal_active_count Number of active connections. These connections are in u
# TYPE agroal_active_count gauge
agroal_active_count{datasource="default",} 0.0
# HELP base_memory_maxHeap_bytes Displays the maximum amount of memory, in bytes, t
# TYPE base_memory_maxHeap_bytes gauge
base_memory_maxHeap_bytes 1.6781410304E10
# HELP process_start_time_seconds Start time of the process since unix epoch.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.675188449054E9
# HELP system_load_average_1m The sum of the number of runnable entities queued to
# TYPE system_load_average_1m gauge
system_load_average_1m 4.005859375
```

Metrics - **components**

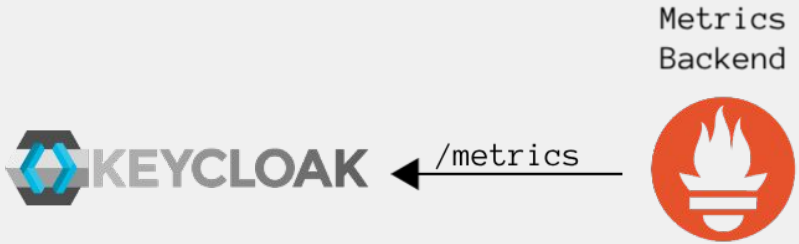
Metrics - components

- **Metrics backend** (*e.g. Prometheus*) **scrapes metrics** from the `/metrics` endpoint



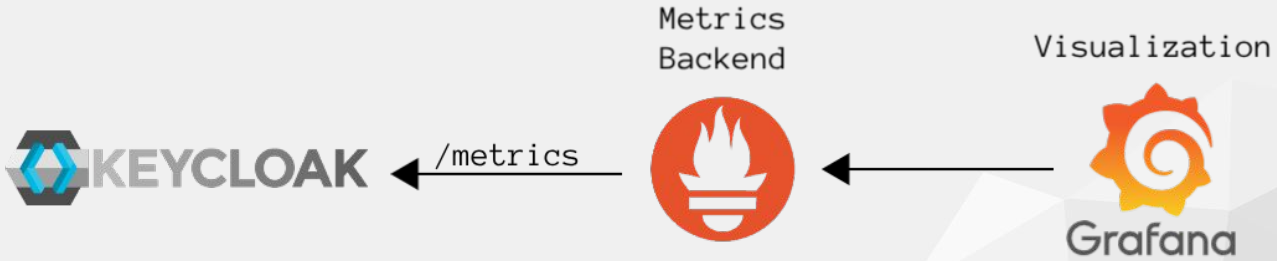
Metrics - components

- **Metrics backend** (*e.g. Prometheus*) **scrapes metrics** from the `/metrics` endpoint
- Metrics backend **collects**, **stores,** and provides **querying** on your metrics



Metrics - components

- **Metrics backend** (*e.g. Prometheus*) **scrapes metrics** from the `/metrics` endpoint
- Metrics backend **collects**, **stores,** and provides **querying** on your metrics
- Metrics can be **visualized** in a visualization platform *(e.g. Grafana)* and provide some **meaningful representation** in form of **dashboards**



Metrics - **Keycloak dashboards**

Metrics - Keycloak dashboards

- Keycloak provides its own Grafana dashboards for **troubleshooting** and **capacity planning**

Metrics - Keycloak dashboards

- Keycloak provides its own Grafana dashboards for **troubleshooting** and **capacity planning**
- **Troubleshooting** - contains information about deployment status with graphs for JVM, DB, HTTP, JDBC caching + basic SLOs

Metrics - Keycloak dashboards

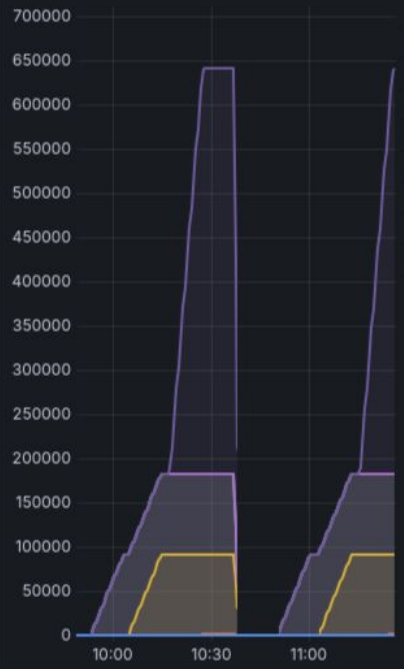
- Keycloak provides its own Grafana dashboards for **troubleshooting** and **capacity planning**
- **Troubleshooting** - contains information about deployment status with graphs for JVM, DB, HTTP, JDBC caching + basic SLOs
- **Capacity planning** - useful when estimating the load handled by a Keycloak deployment

HTTP Metrics

Number of requests in 5 minutes interval

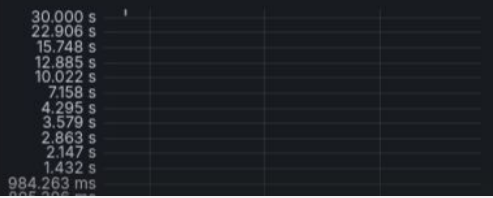
Name	Max	Mean
keycloak-0	250	125
keycloak-1	250	126
keycloak-2	250	126

Total number of requests per URI and outcome



Name	Last *
/realms/{realm}/protocol/{protocol}/token - SUCCESS	640500
/realms/{realm}/login-actions/authenticate - REDIRECTION	183000
/realms/{realm}/protocol/{protocol}/auth - SUCCESS	183000
/realms/{realm}/protocol/{protocol}/logout - REDIRECTION	91500
/health/ready - SUCCESS	765
/health/live - SUCCESS	762
/metrics - SUCCESS	507
/health/started - SUCCESS	3

All requests with processing time



Total number of requests per URI and outcome in 5 minutes interval

Name	Max
/realms/{realm}/{extension}/status-completed - CLIENT_ERROR	0.400
/realms/{realm}/login-actions/authenticate - REDIRECTION	150
/realms/{realm}/protocol/{protocol}/logout - REDIRECTION	150
/health/live - SUCCESS	0.623
/health/ready - SUCCESS	0.623

JVM memory used vs committed

Name	Last *	Min	Max
keycloak-0 - used	544 MiB	253 MiB	1.27 GiB
keycloak-1 - used	690 MiB	252 MiB	1.27 GiB
keycloak-2 - used	651 MiB	259 MiB	1.26 GiB
keycloak-0 - committed	762 MiB	368 MiB	1.36 GiB
keycloak-1 - committed	1.36 GiB	371 MiB	1.37 GiB
keycloak-2 - committed	1.39 GiB	374 MiB	1.39 GiB

Average GC time

keycloak-0 - G1 Young Generation
2.90 ms

keycloak-1 - G1 Young Generation
3.77 ms

keycloak-2 - G1 Young Generation
3.50 ms

Changes in average GC times in 5 minutes interval

Name	Max	M
keycloak-0 - CodeCache GC Threshold	0 s	
keycloak-1 - CodeCache GC Threshold	100.0 µs	690
keycloak-2 - CodeCache GC Threshold	0 s	
keycloak-0 - G1 Evacuation Pause	7.97 ms	2.42
keycloak-1 - G1 Evacuation Pause	14.6 ms	2.04
keycloak-2 - G1 Evacuation Pause	7.73 ms	1.90
keycloak-0 - G1 Humongous Allocation	0 s	

Number of GC events in 5 minutes interval

Maximum GC time and cause

JVM GC CPU overhead %

Metrics - **SLI/SLO**

Metrics - **SLI** (Service Level Indicator)

=> A **measurement** of how well a service is **performing**

Metrics - **SLI** (Service Level Indicator)

=> A **measurement** of how well a service is **performing**

Latency

Metrics - **SLI** (Service Level Indicator)

=> A **measurement** of how well a service is **performing**

Latency

Error rate

Metrics - **SLI** (Service Level Indicator)

=> A **measurement** of how well a service is **performing**

Latency

Error rate

Throughput

Metrics - **SLI** (Service Level Indicator)

=> A **measurement** of how well a service is **performing**

Latency

Error rate

Throughput

Uptime

Metrics - **SLO** (Service Level Objective)

=> A **target** for that **measurement** (SLI)

Metrics - **SLO** (Service Level Objective)

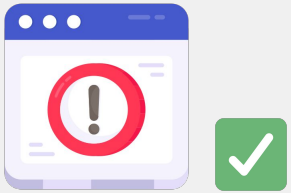
=> A **target** for that **measurement** (SLI)

“99% requests
complete under 300ms
within 30 days”

Metrics - **SLO** (Service Level Objective)

=> A **target** for that **measurement** (SLI)

“99% requests complete under 300ms within 30 days”

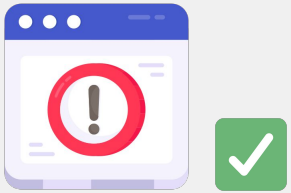


“99.9% requests succeed per year”

Metrics - **SLO** (Service Level Objective)

=> A **target** for that **measurement** (SLI)

“99% requests complete under 300ms within 30 days”



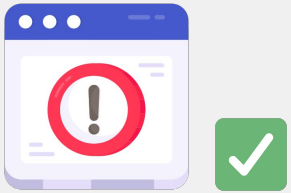
“99.9% requests succeed per year”

“Handles 500req/s 95% of the time within a month”

Metrics - **SLO** (Service Level Objective)

=> A **target** for that **measurement** (SLI)

“99% requests complete under 300ms within 30 days”



“99.9% requests succeed per year”

“Handles 500req/s 95% of the time within a month”

“Available 99.99% of the time per month”

Metrics - SLI/SLO

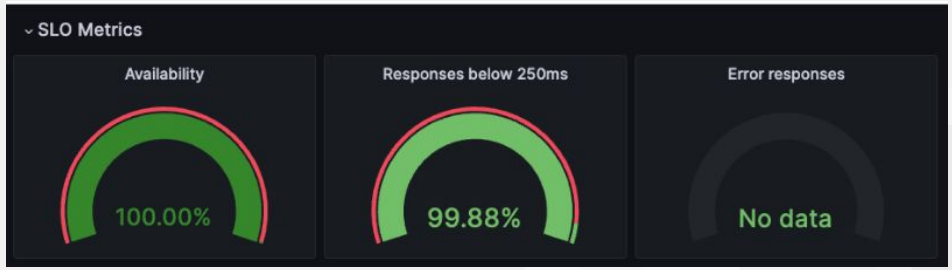
- **SLI/SLO** concepts help to **set expectations** with stakeholders and service owners

Metrics - SLI/SLO

- **SLI/SLO** concepts help to **set expectations** with stakeholders and service owners
- **Enable meaningful alerting** by focusing on **user impact** instead of raw metrics

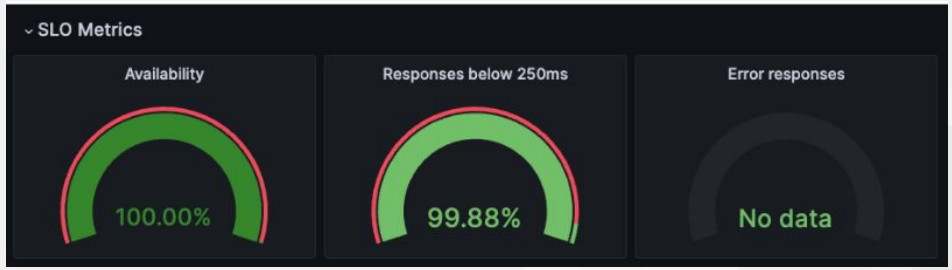
Metrics - SLI/SLO

- **SLI/SLO** concepts help to **set expectations** with stakeholders and service owners
- **Enable meaningful alerting** by focusing on **user impact** instead of raw metrics
- **Keycloak Grafana dashboards** contain graphs for some **fundamental SLOs**

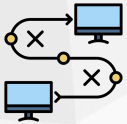


Metrics - SLI/SLO

- **SLI/SLO** concepts help to **set expectations** with stakeholders and service owners
- **Enable meaningful alerting** by focusing on **user impact** instead of raw metrics
- **Keycloak Grafana dashboards** contain graphs for some **fundamental SLOs**
- Monitoring performance with SLI guide

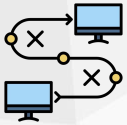


Traces



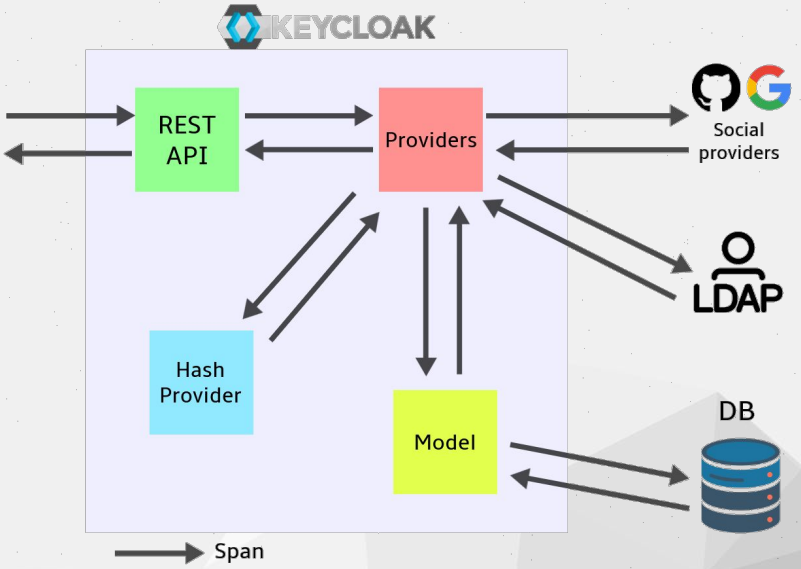
Traces

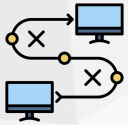
- **Trace** = path of a request through the system



Traces

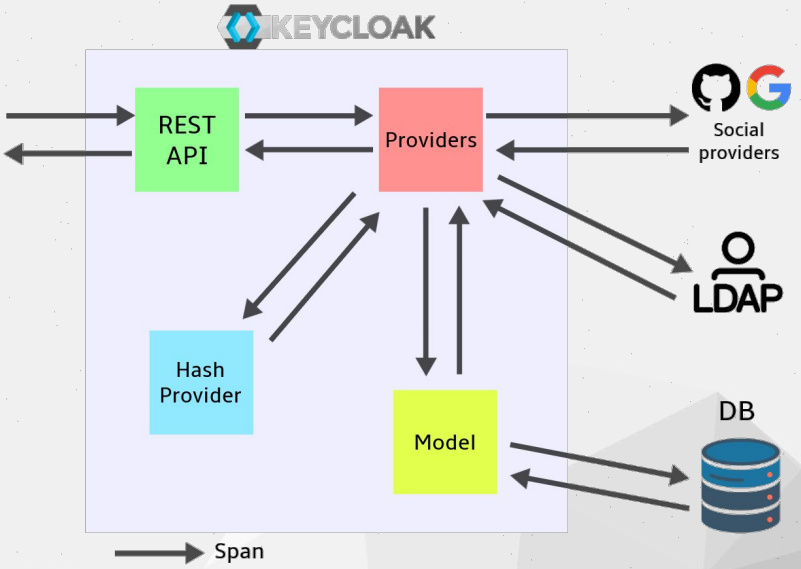
- **Trace** = path of a request through the system

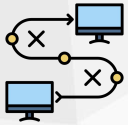




Traces

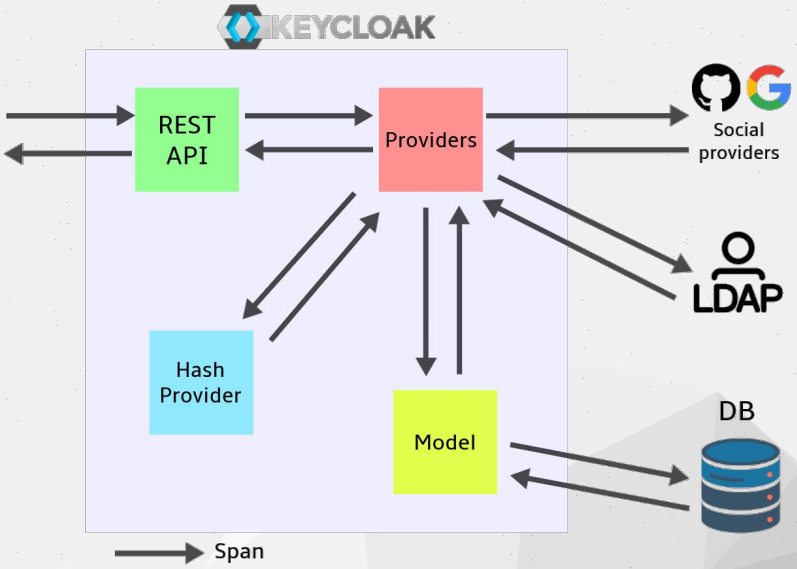
- **Trace** = path of a request through the system
- Benefits:
  - Find **performance bottlenecks**

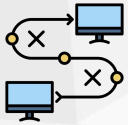




Traces

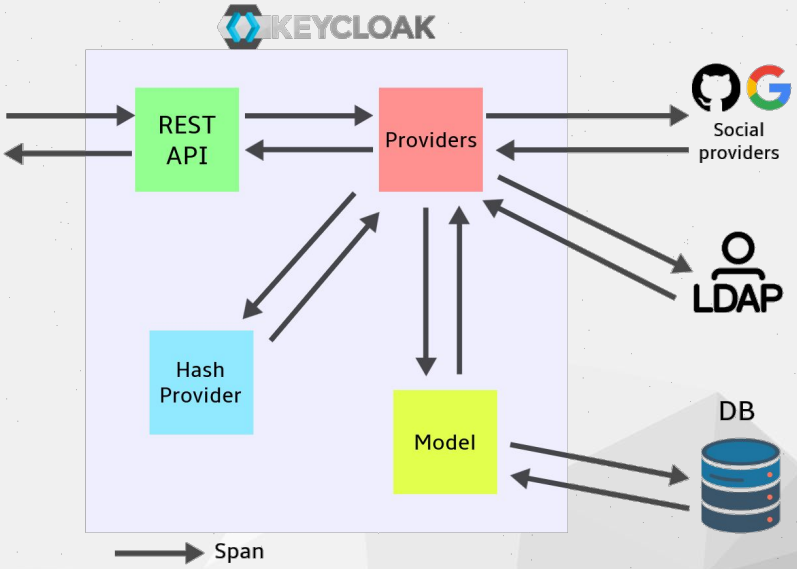
- **Trace** = path of a request through the system
- Benefits:
  - Find **performance bottlenecks**
  - Understand **system behavior**

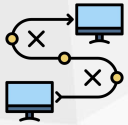




Traces

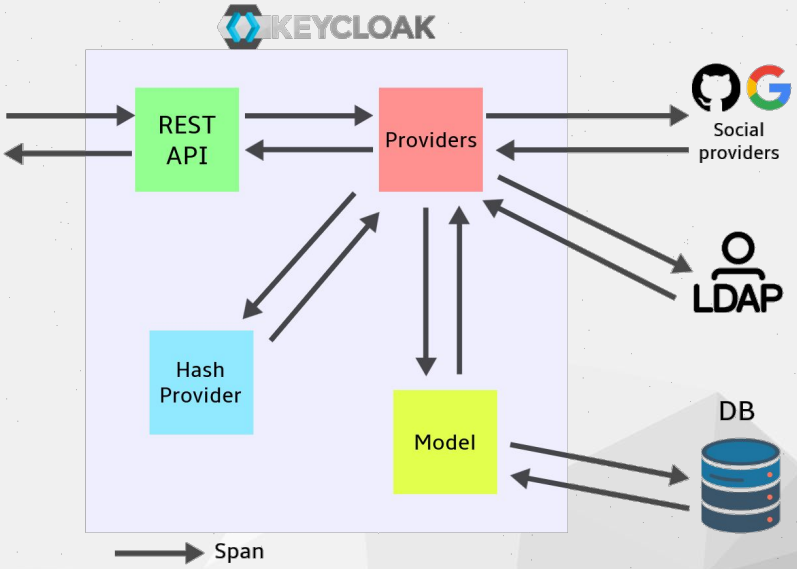
- **Trace** = path of a request through the system
- Benefits:
  - Find **performance bottlenecks**
  - Understand **system behavior**
  - **Diagnose** and **troubleshooting**



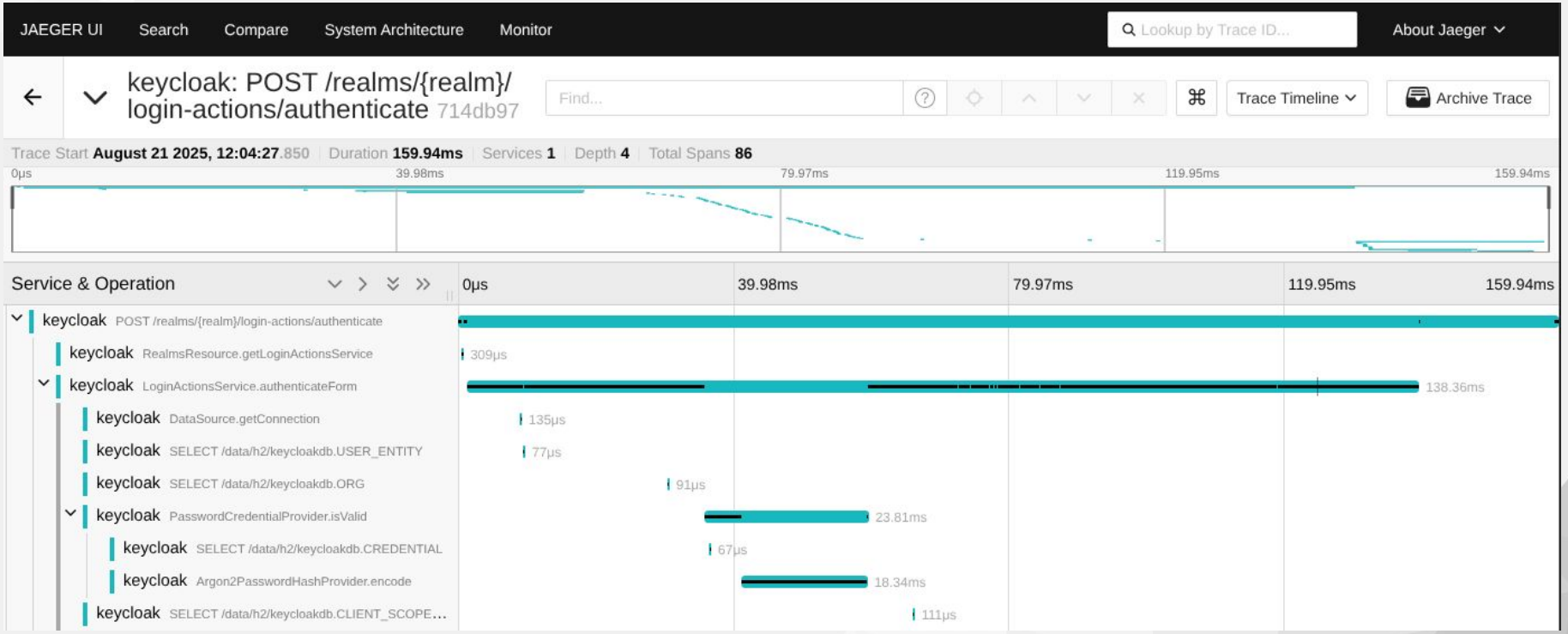


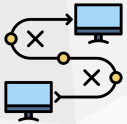
Traces

- **Trace** = path of a request through the system
- Benefits:
  - Find **performance bottlenecks**
  - Understand **system behavior**
  - **Diagnose** and **troubleshooting**
  - Real-time **monitoring**

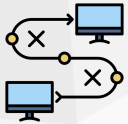


Traces


JAEGER UI
Search
Compare
System Architecture
Monitor
Lookup by Trace ID...
About Jaeger
keycloak: POST /realms/{realm}/ login-actions/authenticate 714db97
Find...
Trace Timeline
Archive Trace
Trace Start August 21 2025, 12:04:27.850 | Duration 159.94ms | Services 1 | Depth 4 | Total Spans 86
0µs
39.98ms
79.97ms
119.95ms
159.94ms
Service & Operation
keycloak POST /realms/{realm}/login-actions/authenticate
keycloak RealmsResource.getLoginActionsService
309µs
keycloak LoginActionsService.authenticateForm
138.36ms
keycloak DataSource.getConnection
135µs
keycloak SELECT /data/h2/keycloakdb.USER_ENTITY
77µs
keycloak SELECT /data/h2/keycloakdb.ORG
91µs
keycloak PasswordCredentialProvider.isValid
23.81ms
keycloak SELECT /data/h2/keycloakdb.CREDENTIAL
67µs
keycloak Argon2PasswordHashProvider.encode
18.34ms
keycloak SELECT /data/h2/keycloakdb.CLIENT_SCOPE...
111µs

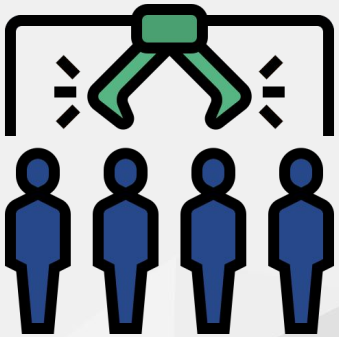


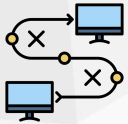
Traces - **Sampling**



Traces - Sampling

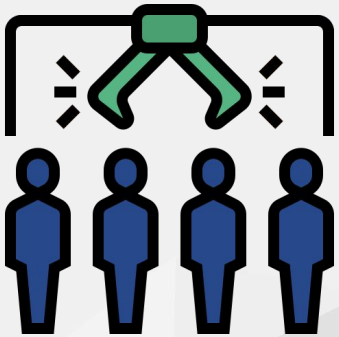
- You **do not** always **need 100% of your traces**

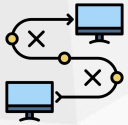




Traces - Sampling

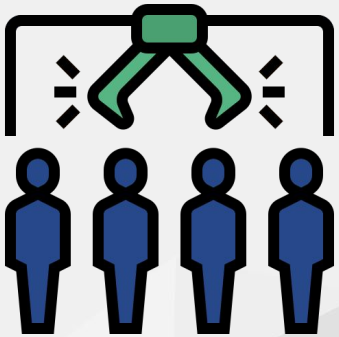
- You **do not** always **need 100% of your traces**
- **Storing all traces = check your huge bill for data storing** + performance

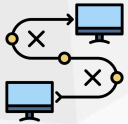




Traces - Sampling

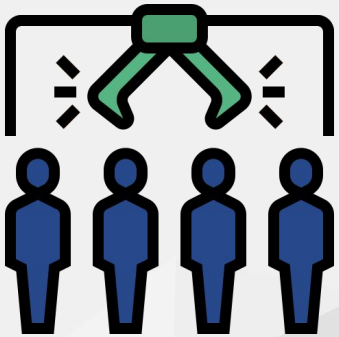
- You **do not** always **need 100% of your traces**
- **Storing all traces = check your huge bill for data storing** + performance
- **Ratio-based** sampler is the default

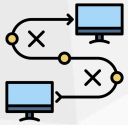




Traces - Sampling

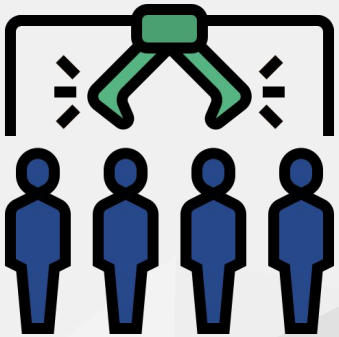
- You **do not** always **need 100% of your traces**
- **Storing all traces = check your huge bill for data storing** + performance
- **Ratio-based** sampler is the default
- i.e. You might store only **10% of all traces**





Traces - Sampling

- You **do not** always **need 100% of your traces**
- **Storing all traces = check your huge bill for data storing** + performance
- **Ratio-based** sampler is the default
- i.e. You might store only **10% of all traces**



Keycloak Guide: [Root cause analysis with tracing]

What's new?

Keycloak 26.3/26.4

KEYCLOAK

Logs - news

- **MDC (Mapped Diagnostic Context)** (26.4) - add **additional context** information to **log records**

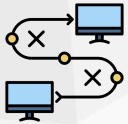
```
2025-06-20 14:13:01,772 {kc.clientId=security-admin-console, kc.realm=master} INFO ...
```

- **HTTP Access Logs** (26.4) - record details of **incoming HTTP requests**
- **Asynchronous logging** (26.3) - handle log records in a separate thread
- **OpenTelemetry Logs** (26.4?) - send logs to OTel Collector

Metrics - news

- **OpenTelemetry Metrics** (26.4?) - will be probably achieved via the Micrometer to OpenTelemetry bridge Quarkus extension

- **Updated guides** on www.keycloak.org

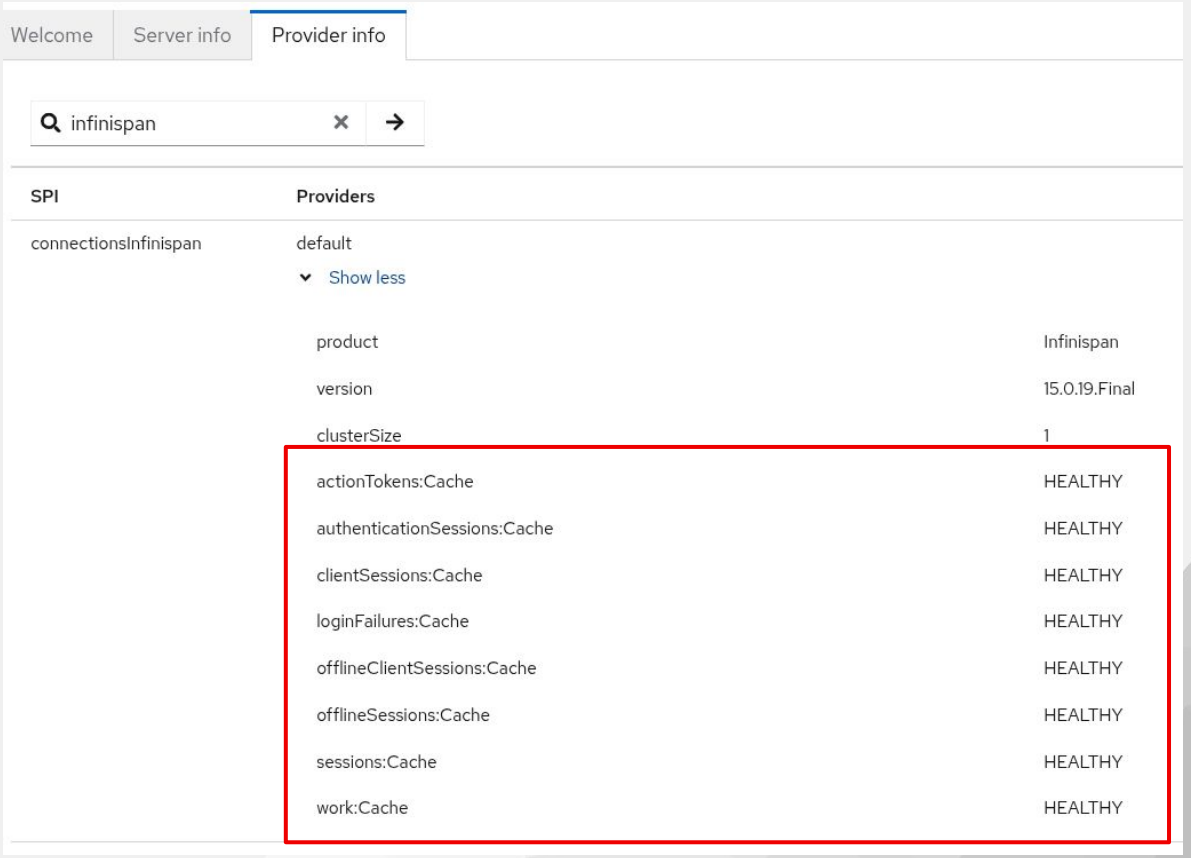


Traces - news

- Polishing and bug fixes
- **Trace JGroups communication** (embedded caches) (26.4)

Caches health

- Admin console shows **health status** of **caches**



OpenTelemetry support

OpenTelemetry support

- Initiative to **support OpenTelemetry** (OTel) for **Logs**, **Metrics**, and **Traces**

OpenTelemetry support

- Initiative to **support OpenTelemetry** (OTel) for **Logs**, **Metrics**, and **Traces**
- **Telemetry data** are sent to a **single place** - OTel Collector

OpenTelemetry support

- Initiative to **support OpenTelemetry** (OTel) for **Logs**, **Metrics**, and **Traces**
- **Telemetry data** are sent to a **single place** - OTel Collector
- **Unified standard** for instrumenting, generating, and exporting **telemetry**

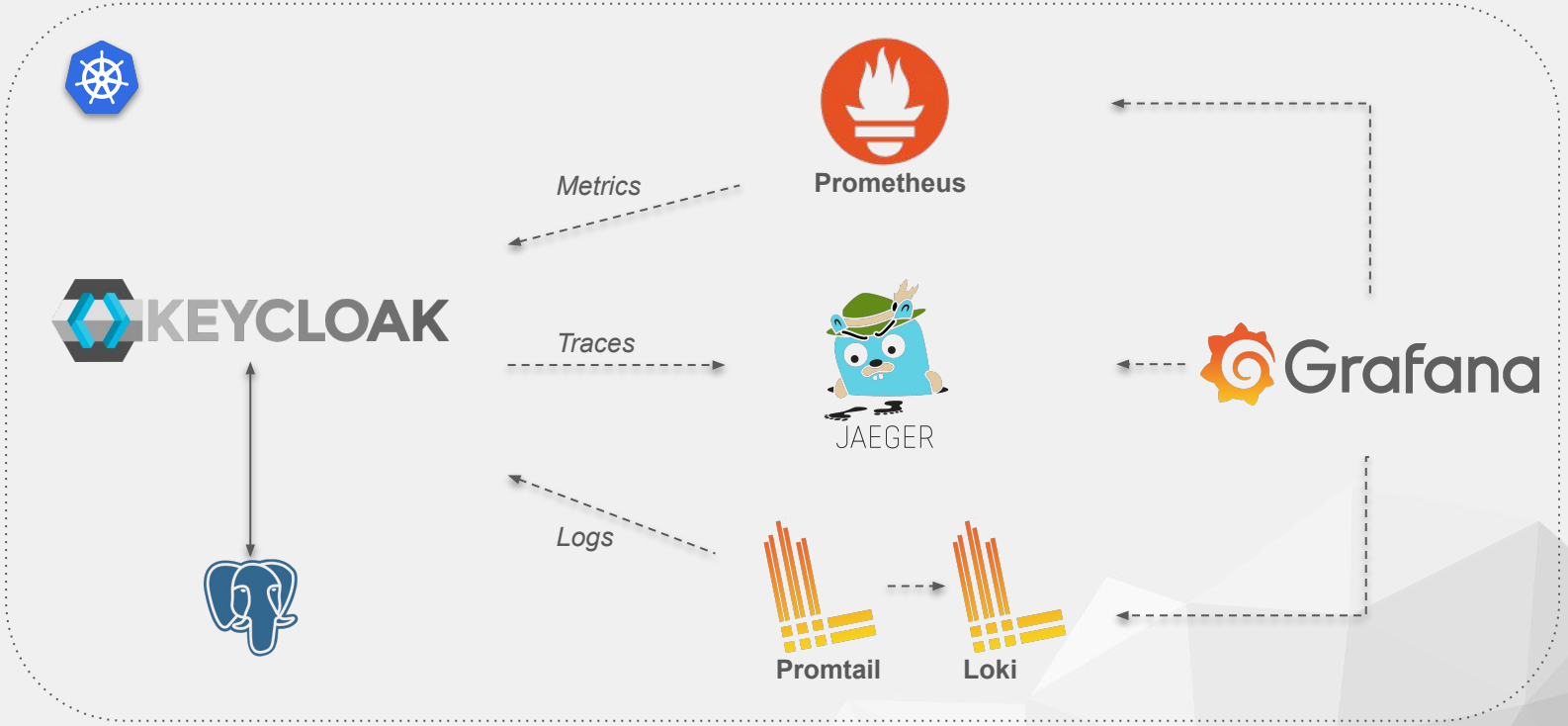
OpenTelemetry support

- Initiative to **support OpenTelemetry** (OTel) for **Logs**, **Metrics**, and **Traces**
- **Telemetry data** are sent to a **single place** - OTel Collector
- **Unified standard** for instrumenting, generating, and exporting **telemetry**
- Probably some new Keycloak options to share configuration of the collector

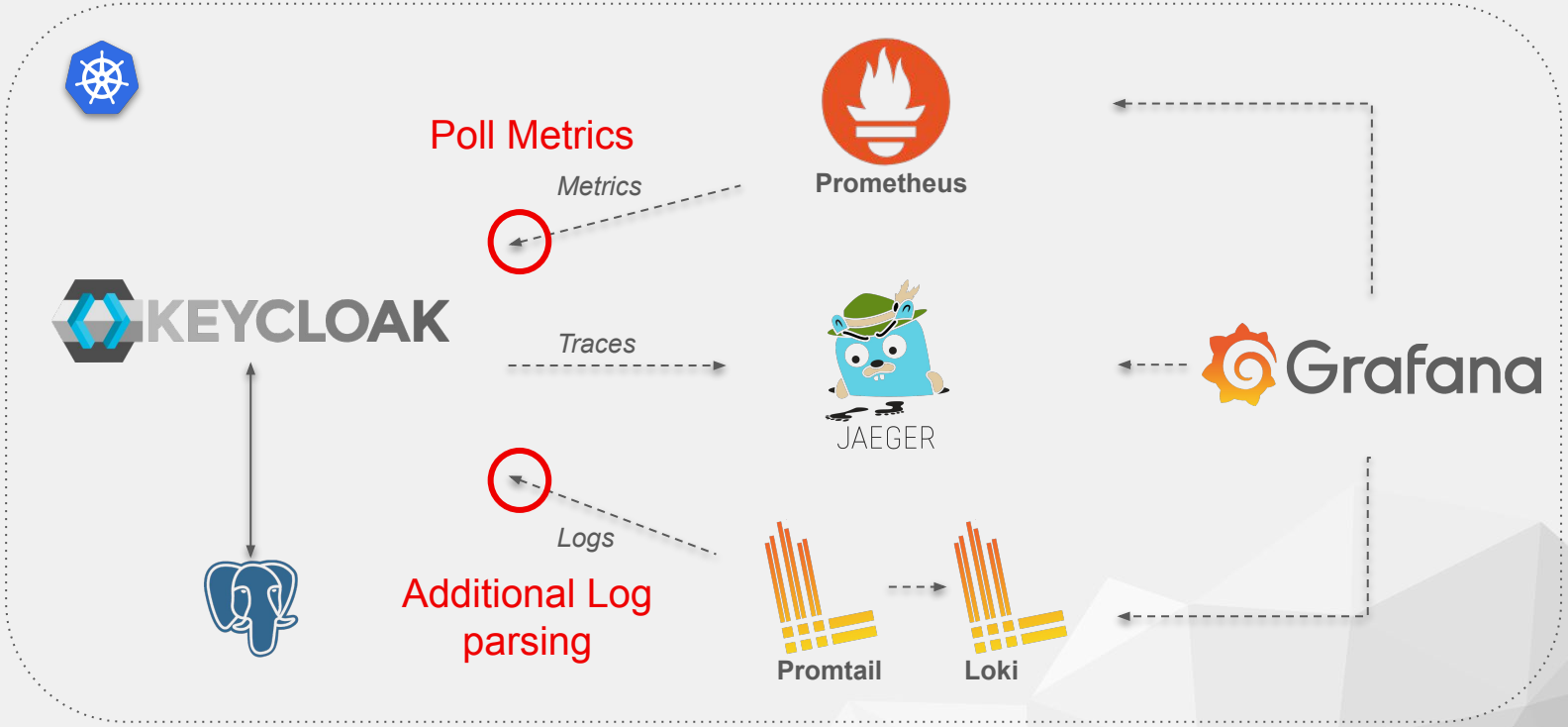
OpenTelemetry support

- Initiative to **support OpenTelemetry** (OTel) for **Logs**, **Metrics**, and **Traces**
- **Telemetry data** are sent to a **single place** - OTel Collector
- **Unified standard** for instrumenting, generating, and exporting **telemetry**
- Probably some new Keycloak options to share configuration of the collector
- **Preview** support for **Logs** and **Metrics** hopefully in **26.4**

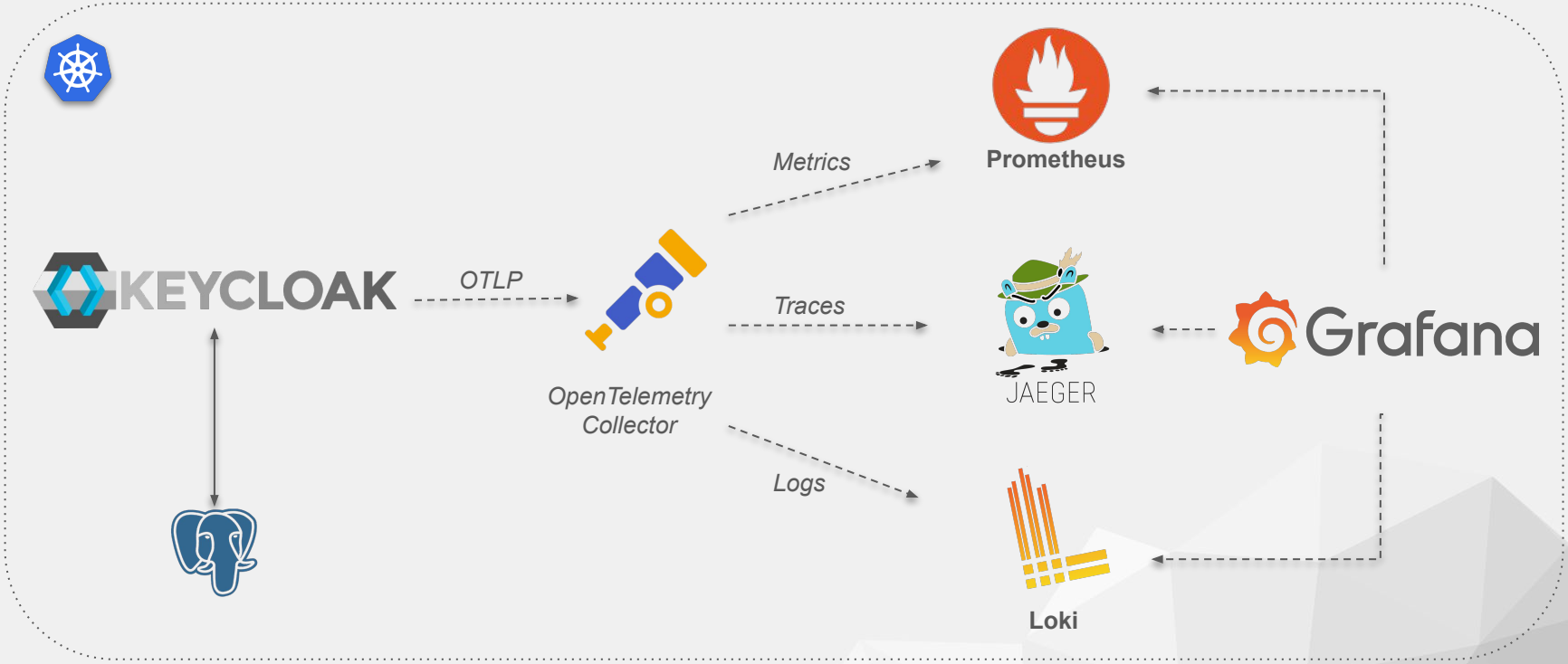
Architecture **without OTel**



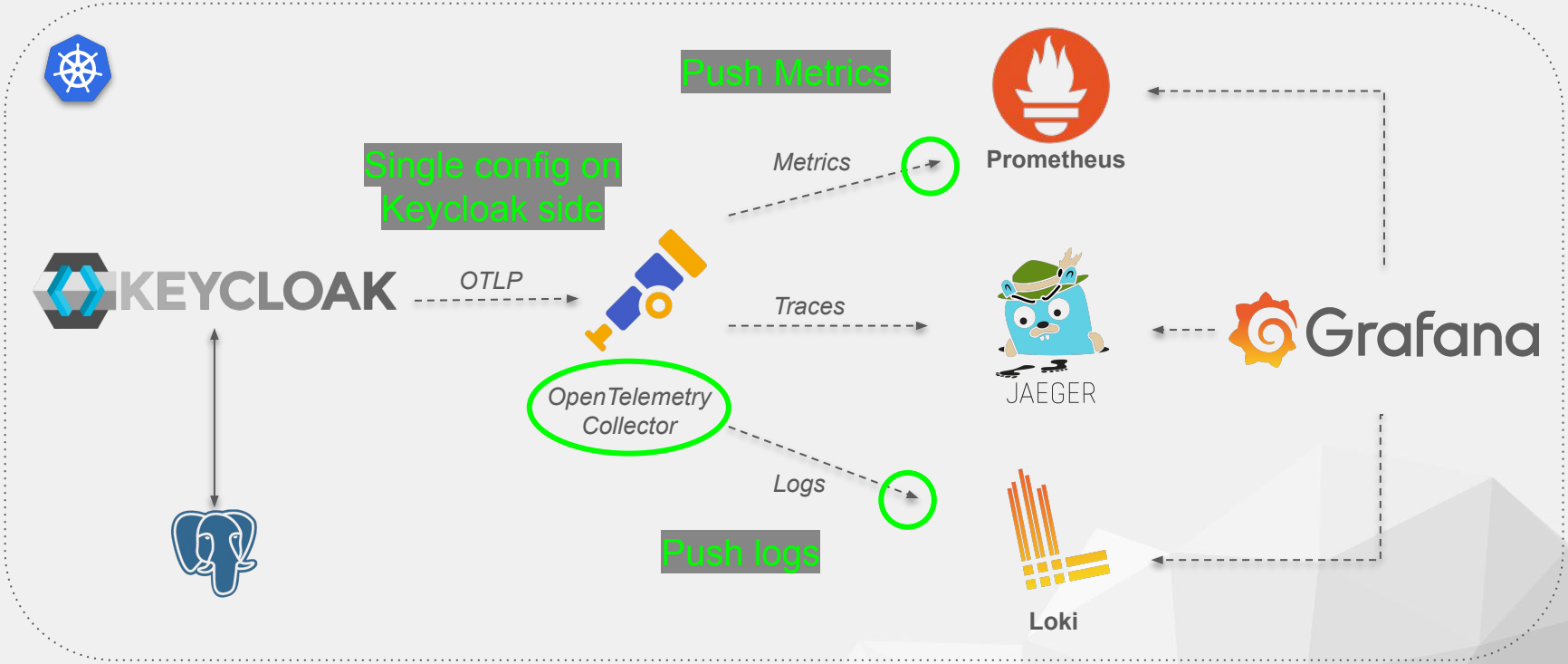
Architecture **without OTel**



Architecture **with OTel**

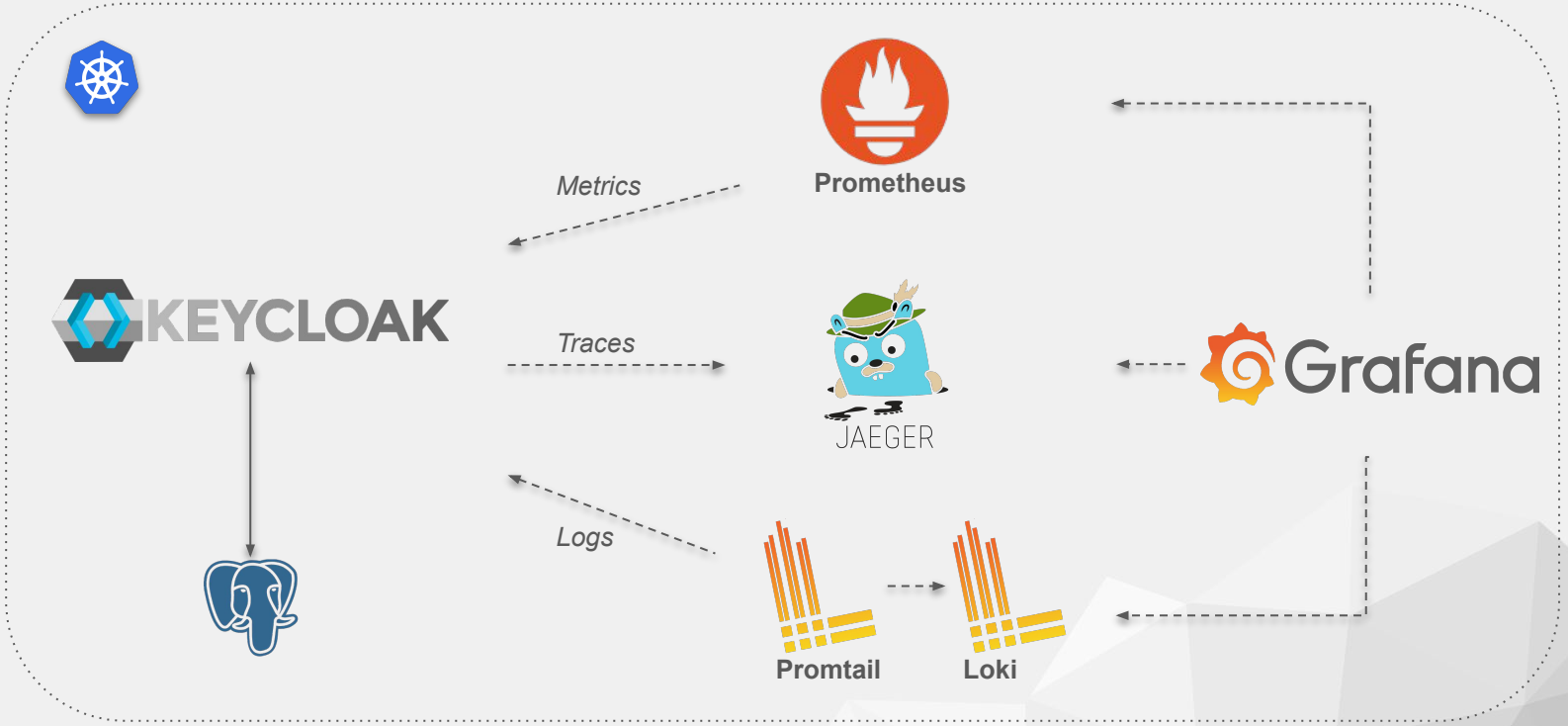


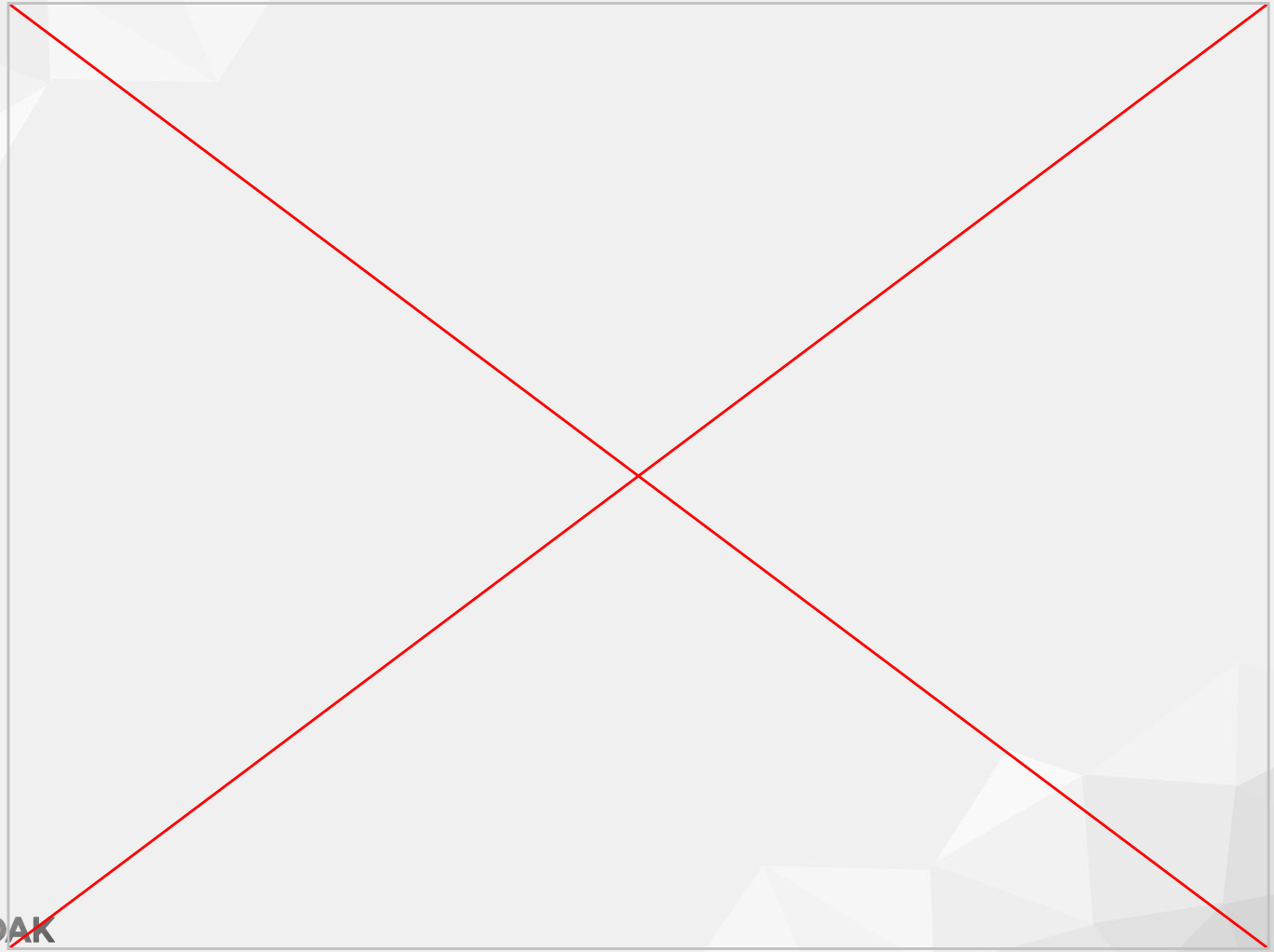
Architecture **with OTel**



Demo

Deployed Architecture





KEYCLOAK

Backup video

Stay in touch

Martin Bartoš

mabartos@ibm.com

@mabartos

@mabartos98

Credits

- *Diagrams: Keycloak contributors*
- *Icons: https://www.flaticon.com/*