Grafana Labs

Observability 101

From theory to practice 🚀

3 December 2022

Lukasz Gut
Senior Software Engineer

Demystifying observability

Three pillars of observability

Incident response and management

Observability in practice

The future and beyond

Demystifying O11y...

Telemetry

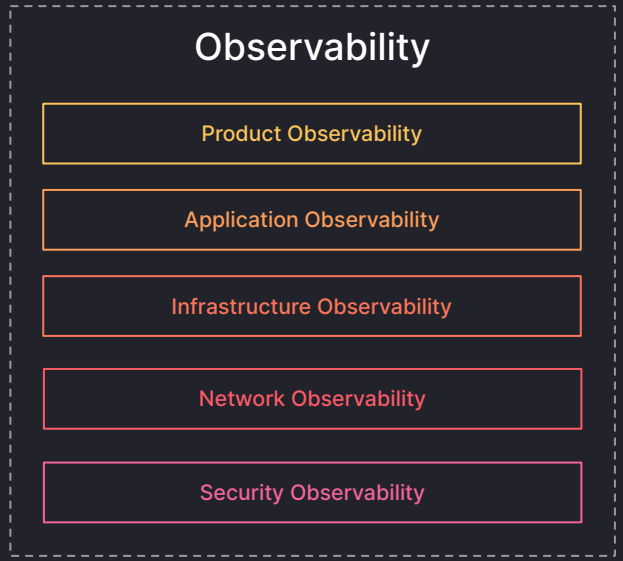
Collect & Store

Monitoring

Visualize

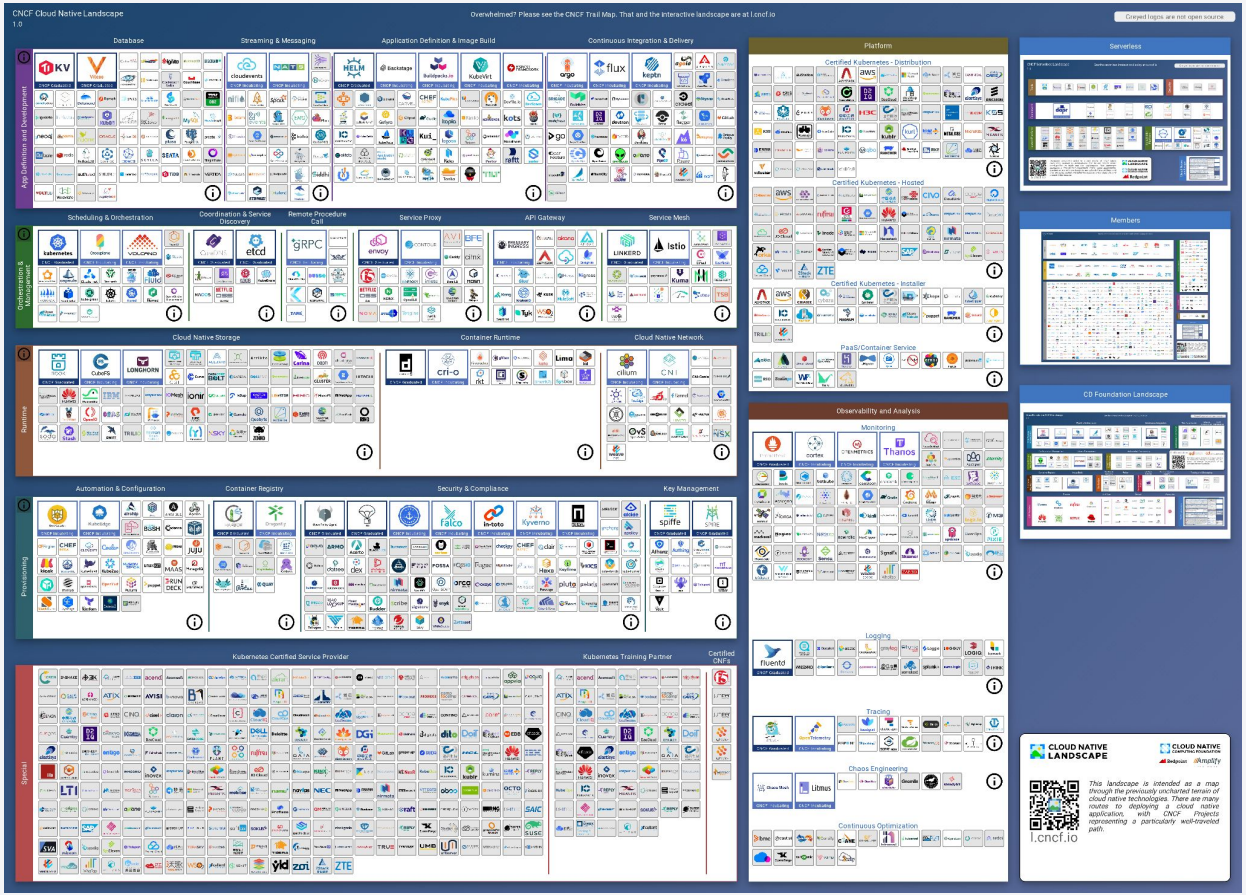
Observability

Understand

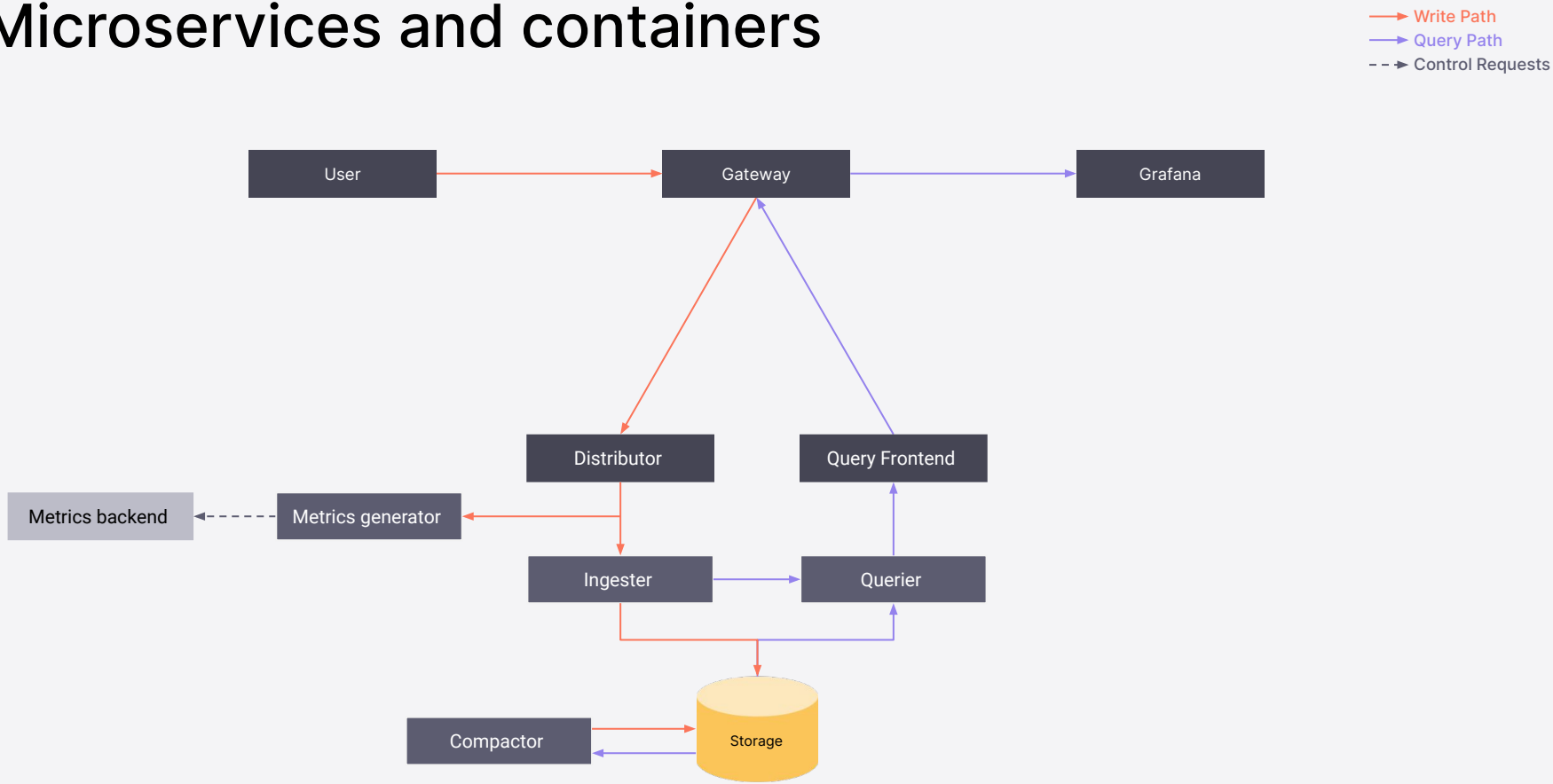

Observability
Product Observability
Application Observability
Infrastructure Observability
Network Observability
Security Observability

Why do we care?

Observability
is
hard


CNCF Cloud Native Landscape
1.0
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
Grayed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Certified CNFs
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Continuous Optimization
Serverless
Members
CD Foundation Landscape
CLOUD NATIVE LANDSCAPE
CLOUD NATIVE COMPUTING FOUNDATION
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io

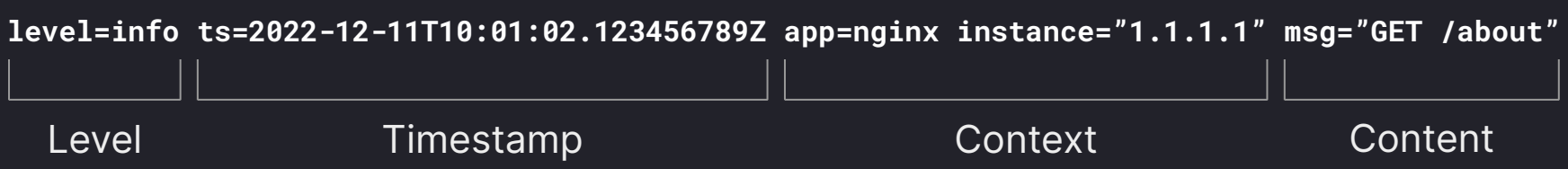
Microservices and containers



Logs

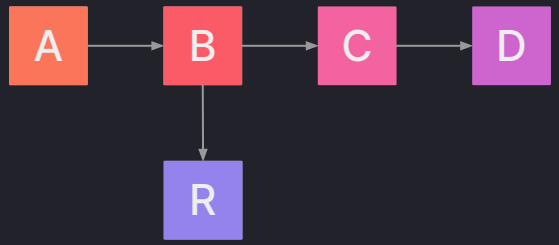
Metrics

Traces

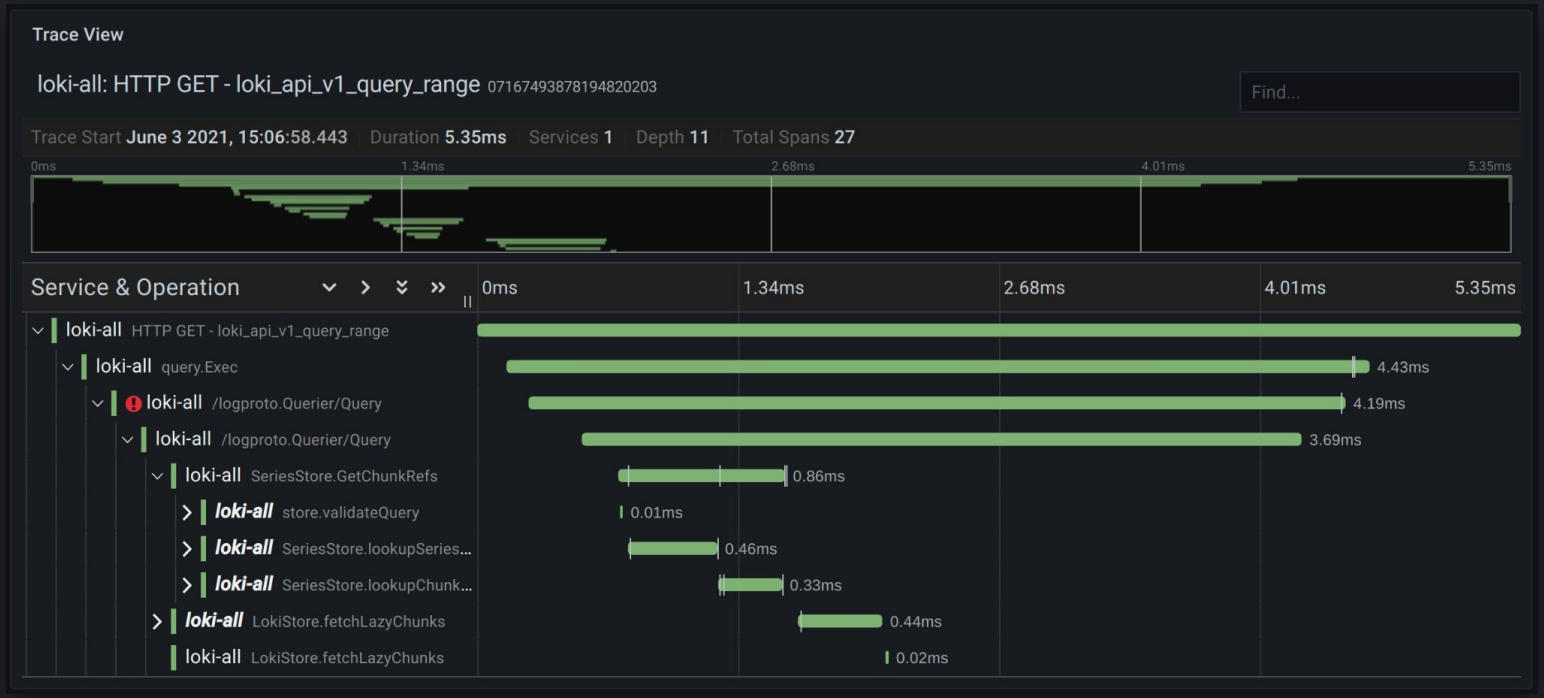

level=info ts=2022-12-11T10:01:02.123456789Z app=nginx instance="1.1.1.1" msg="GET /about"
Level
Timestamp
Context
Content

A Prometheus metric model


General / Service overview
Last 3 hours
logins
Last 20 minutes
logins
logins (-1 hour)
Memory / CPU
cpu
memory
Free disk space
sda1
sda2
sda3
sda5
sda6
sda7
69.9 GB
24.2 GB
20.3 GB
18.1 GB
89.8 GB
73.0 GB
Server Request / s
web_server_01
web_server_02
web_server_03
web_server_04
Free disk space
80.6 GB
sda1
21.9 GB
sda2
54.5 GB
sda3
68.0 GB
sda5
29.7 GB
sda6
4.21 GB
sda7
Server CPU usage
Backend-ops-01
Backend-ops-02
Backend-ops-03
Backend-ops-04
Frontend-web-01
Frontend-web-02
< 10%
10%+
60%+
70%+
client side full page load
avg
upper_25 2.03 ms
upper_50 138 ms
upper_75 612 ms
upper_90 1.20 s
upper_95 1.70 s
Flow readings
time flow1 flow2 flow3
Jun 8th, 2021 11:23:43 50% 50% 50%
Jun 8th, 2021 11:25:43 54.0% 49.7% 44.5%
Jun 8th, 2021 11:27:43 53.0% 49.7% 50.2%
Jun 8th, 2021 11:29:43 49.4% 51.8% 54.6%
Jun 8th, 2021 11:31:43 51.1% 52.7% 57.5%
Jun 8th, 2021 11:33:43 50.2% 56.8% 62.9%
Jun 8th, 2021 11:35:43 53.1% 61.3% 65.0%
Jun 8th, 2021 11:37:43 50.6% 59.7% 68.9%
Jun 8th, 2021 11:39:43 52.4% 66.0% 65.8%
Jun 8th, 2021 11:41:43 51.5% 64.9% 72.5%
Jun 8th, 2021 11:43:43 54.0% 69.0% 79.5%
Jun 8th, 2021 11:45:43 52.8% 75.9% 81.9%
Jun 8th, 2021 11:47:43 57.1% 78.2% 86.8%
Jun 8th, 2021 11:49:43 61.8% 84.3% 85.0%
Jun 8th, 2021 11:51:43 65.2% 79.5% 88.4%
Jun 8th, 2021 11:53:43 68.1% 85.3% 90.4%


A
B
C
D
R


Time
Stack trace
A
B
R
C
D


Trace View
loki-all: HTTP GET - loki_api_v1_query_range 07167493878194820203
Find...
Trace Start June 3 2021, 15:06:58.443 Duration 5.35ms Services 1 Depth 11 Total Spans 27
0ms
1.34ms
2.68ms
4.01ms
5.35ms
Service & Operation
loki-all HTTP GET - loki_api_v1_query_range
loki-all query.Exec
4.43ms
loki-all /logproto.Querier/Query
4.19ms
loki-all /logproto.Querier/Query
3.69ms
loki-all SeriesStore.GetChunkRefs
0.86ms
loki-all store.validateQuery
0.01ms
loki-all SeriesStore.lookupSeries...
0.46ms
loki-all SeriesStore.lookupChunk...
0.33ms
loki-all LokiStore.fetchLazyChunks
0.44ms
loki-all LokiStore.fetchLazyChunks
0.02ms

Observability must be paired with action

IRM = Incident Response & Management

First 60 seconds of an incident

During an incident and after

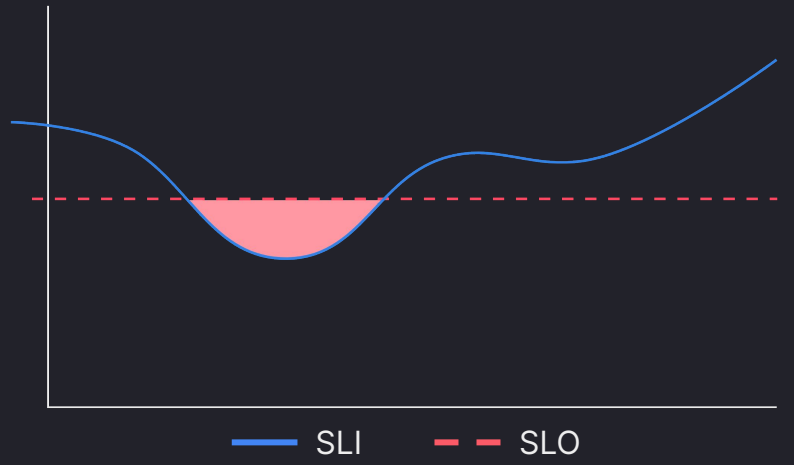
SLAs, SLOs and SLIs

SLA is the agreement between you and your clients or users

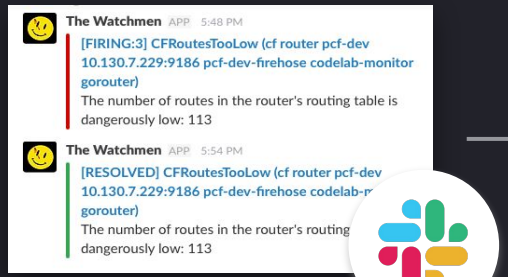
SLOs are the objectives that the service needs to hit

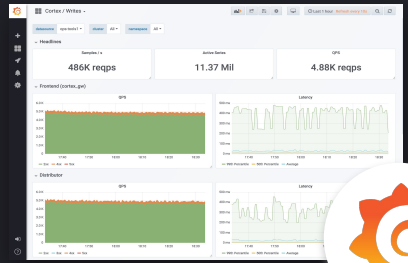
SLIs are the measurements indicating service performance (e.g. service availability)

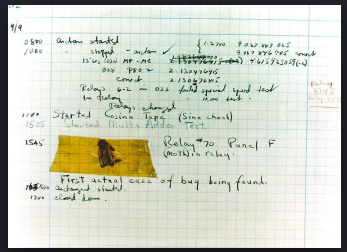
Area is the consumed **error budget**

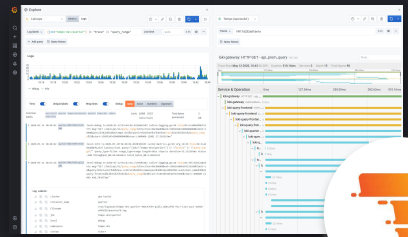


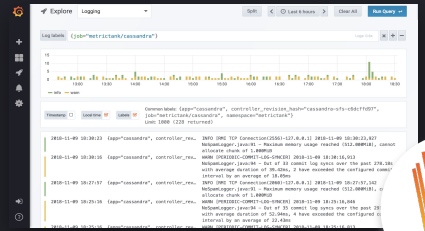
Incident response flow





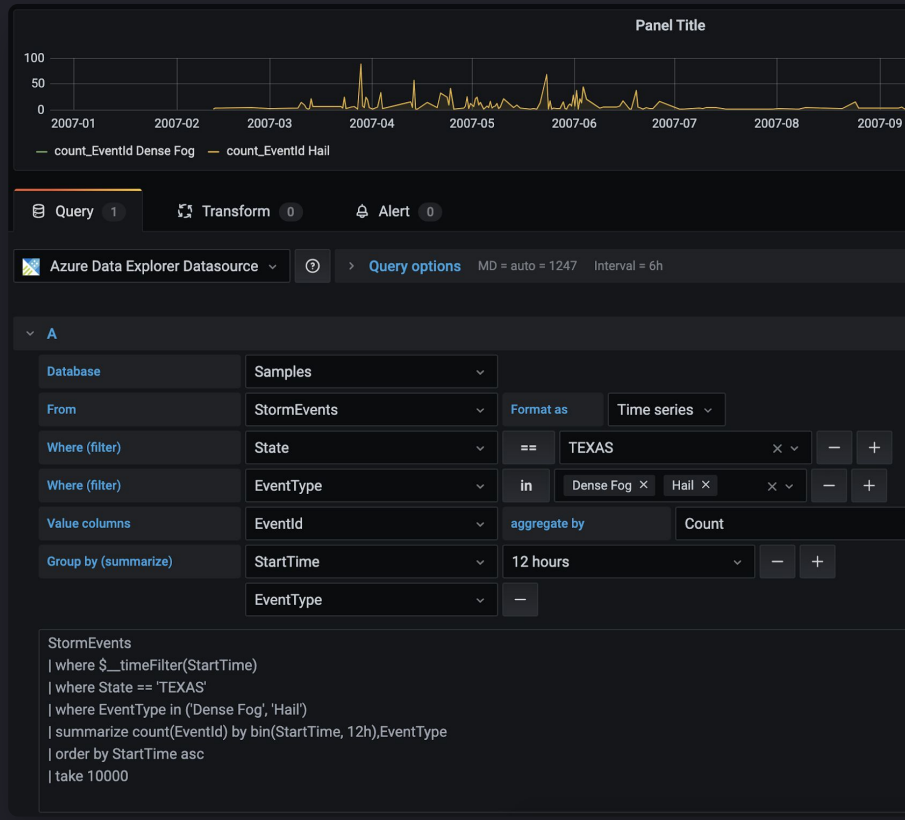






Grafana

- Open-source monitoring platform
- Provides E2E observability capabilities
- Supports heterogeneous data sources



The holy backends

Grafana Loki

Log aggregation system inspired by Prometheus. Supports querying data using LogQL.

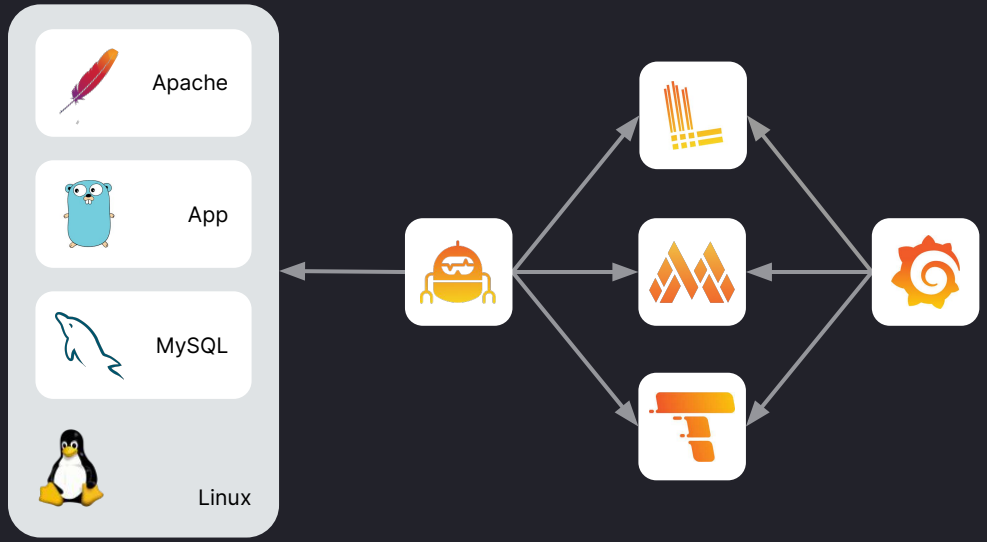
Grafana Mimir

Metrics backend compatible with Prometheus. Supports querying data using PromQL.

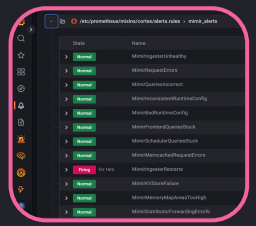
Grafana Tempo

Distributed tracing backend compatible with OpenTelemetry, OpenCensus, Jaeger, Zipkin, and Kafka. Supports querying data using TraceQL.

- Open-source
- Horizontally scalable and highly-available
- Natively support multi-tenancy
- Share similar architecture



4
Dashboards



1
Collection

2
Aggregation

3
Visualization

5
Alerts

6
IRM

What's next?

AI/ML	techniques applied across the entire stack
New platforms	providing observability capabilities, such as eBPF
New signals	such as profiles
Standardization	of platforms, APIs, and frameworks
Streamlined experience	from day 1

Takeaways

1. Observability is hard
2. Logs, metrics and traces form the three pillars of observability
3. The future looks exciting
4. We have only scratched the surface…

Grafana Labs

Thank you

Q&A