



Grafana's Open Source products



Sebastian Schubert

Solutions Engineer

- *30y Firefighter - 26y in Monitoring & Observability*
- *Thinks one bullet point is enough for an intro*



<https://www.linkedin.com/in/bastischubert>

About Grafana Labs



Open source is at the heart of what we do



Employ 91% of the Loki team members, including project founders



Employ 89% of Grafana team members, including project founders



Employ 100% of the Tempo team members, including project founders



Employ 100% of the Mimir team members, including project founders



Employ 100% of Pyroscope team members, including the project founders



Employ 100% of k6 team members, including the project founders



Employ 44% of the Prometheus team members



The leading contributors to the Graphite project



Employ contributors, including a Governance Committee member



Employ 100% of OnCall team members, including the project founders



Employ 100% of Faro team members, including the project founders

1,200+

Employees across
40+ countries

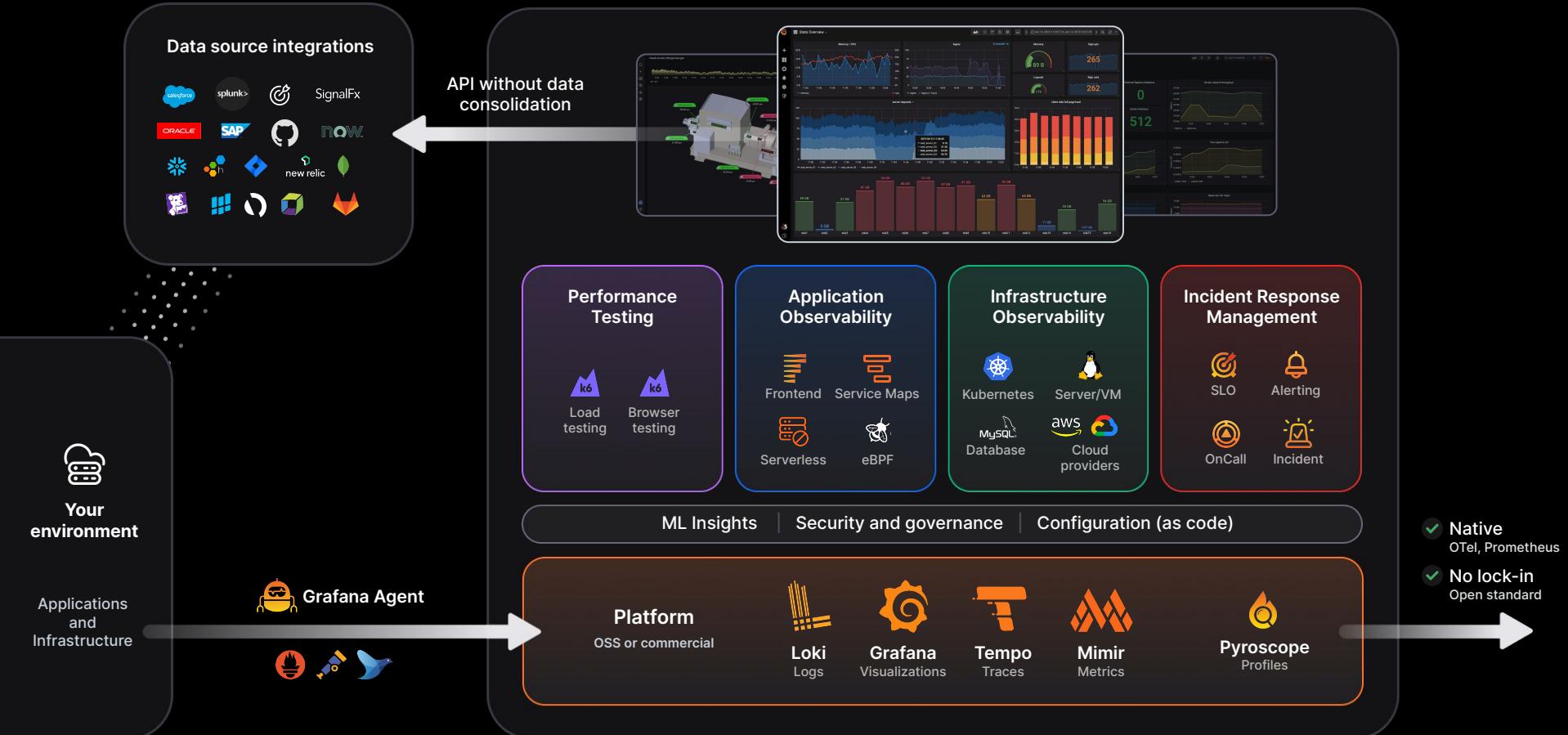
1M+

Instances across
Grafana Cloud and OSS

20M+

Users across OSS
and Cloud Free tier

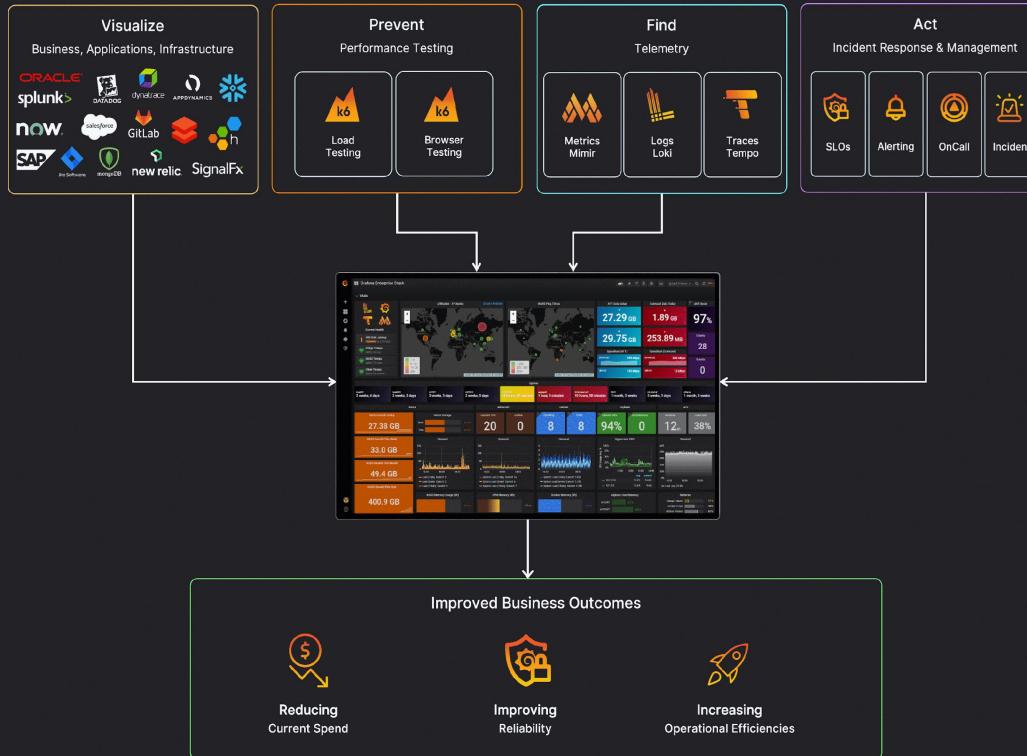
Your open and composable observability stack





Grafana

Grafana Labs – Open & Composable Observability Stack

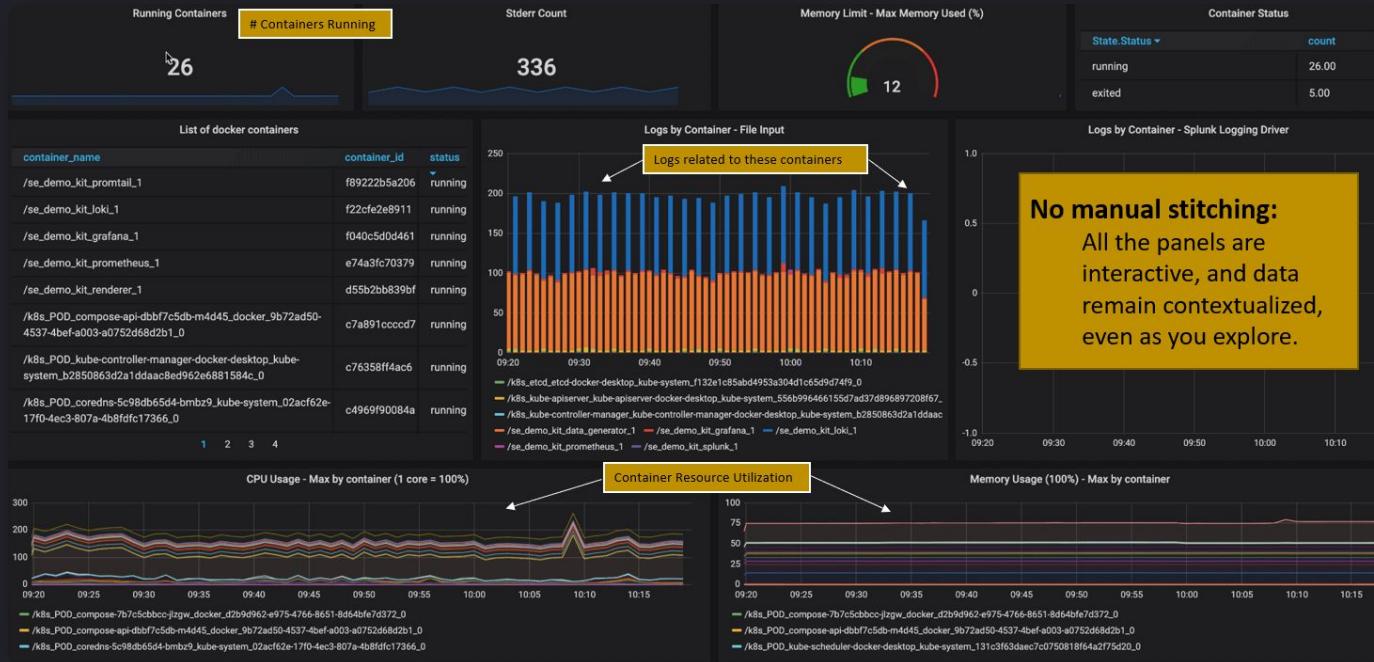


Grafana

- 25 out of the box Visualizations
 - >100 more available as plugins
- 150+ Datasources (incl. Enterprise)
- Available as OSS in a usable way



With Grafana: Unified Metrics, Logs, & Traces



Customer Example: Before Grafana

The screenshot displays a web browser window with several tabs open, each showing different monitoring data. Red arrows point to the tabs for Splunk, Zabbix, Open Source Search, Datadog, and New Relic Metrics & Traces.

- Splunk Docker Overview:** Shows a summary of running containers (30), stderr count (266), memory usage (0.602%), and a list of docker containers.
- Zabbix :: The Enterprise-Class:** Shows a chart of logs by container - File Input over time (11:55 PM Tue Apr 21 2020 to 12:15 AM Wed Apr 22).
- Open Source Search: The Creative:** Shows a chart of logs by container - Splunk Log.
- Datadog: Log In:** Shows a terminal window titled "docker-compose up" displaying log output for data_generator_1.
- New Relic Metrics & Traces:** Shows a terminal window titled "docker-compose up" displaying log output for data_generator_1.





Grafana Correlations

Correlate your data, no matter where it lives.

Who?

Users with related data in more than one data source.

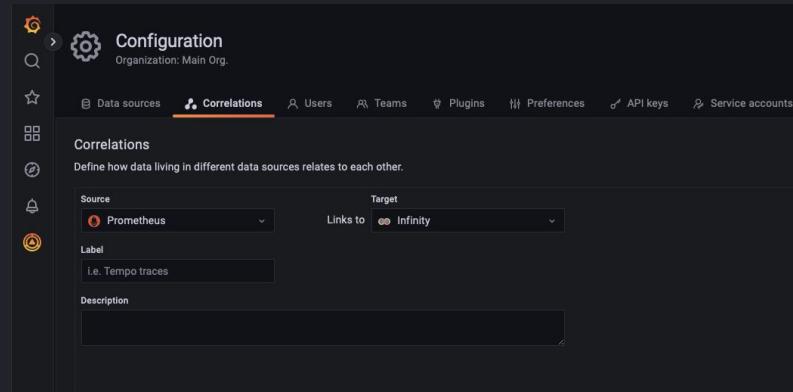
Why?

Bringing together all the signals and context for the systems you care about enables you to truly observe those systems.

Unfortunately, those signals and context often live in different data sources. Grafana Glue brings all that data together and enables you to explore correlated and contextualized information.

What?

- ✓ Define the relationships between your data in Grafana
- ✓ Use Grafana to pivot between different data sources using those relationships. For example, go from Prometheus metrics to Splunk logs for the same application.



New onboarding experience - Revamped, easier Panel Edit side pane

Improve information architecture and user experience for new users

Who?

All users creating or editing panels

Why?

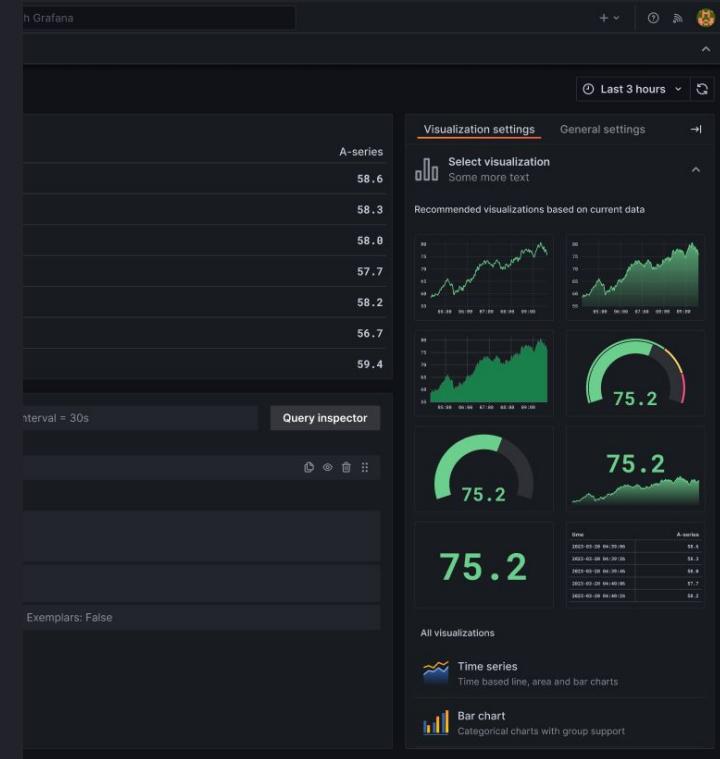
Recent user research has shown that new users are not confident in creating and configuring a visualisation panel that answers their questions.

What?

Revamp specific areas on the panel edit page to improve discoverability and task success. This includes an improved sidebar layout and rethought IA on the top bar options.

Feature highlights

- ✓ New collapsible side pane that unifies Visualization selector and settings as well as general settings.
- ✓ Separation of visualization selection and general settings in the side pane.



Find

Grafana TraceQL Search UI

Find the traces you want without learning TraceQL

Who?

Users looking to unlock the benefits of TraceQL without having to learn yet-another query language.

Why?

Users want to answer questions about their systems using their traces as quickly and easily as possible. Making them learn an entirely new query language to do this can present a steep barrier to entry.

What?

A point-and-click way to build TraceQL queries in Grafana.

Feature highlights

- ✓ See the underlying query being run to build your familiarity with the language
- ✓ Use drop-downs to select the attribute values you want to filter by
- ✓ Configure the default attributes you want to surface as filters
- ✓ Works with for OSS Tempo, Grafana Cloud Traces, and Grafana Enterprise Traces

The screenshot shows the Grafana TraceQL Search UI interface. At the top, there's a dropdown menu with the letter 'A' and the text '(Tempo (tempo-ops))'. Below it is a navigation bar with tabs: 'Query type' (which is selected), 'Search', 'TraceQL', 'JSON File', 'Service Graph', and 'Loki S'. The main area contains four search fields with dropdown menus:

- 'Resource Service Name': set to 'cortex-gateway' with an equals sign operator.
- 'Span Name': set to '/cortex.Ingestor/Push' with an equals sign operator.
- 'Duration': set to '> 1ms' with a greater than operator.
- 'Tags': set to 'resource: namespace: span: kind: client:' with various operators like equals, not equals, and contains.

Below these fields is a code editor window displaying the TraceQL query: `{ resource.service.name="cortex-gateway" && name="/cortex.Ingestor/Push"`.

This screenshot shows another instance of the Grafana TraceQL Search UI. It has a similar layout with a dropdown menu 'A (Tempo (tempo-ops))' and a navigation bar with tabs: 'Query type' (selected), 'Search', 'TraceQL', 'JSON File', 'Service Graph', and 'Loki Search'. The search fields are:

- 'Resource Service Name': set to 'cortex-gateway' with an approximate operator (~).
- 'Span Name': set to 'grafana-com' with an equals sign operator.
- 'Duration': set to '> k6_cloud_insights_forwarder' with a greater than operator.
- 'Tags': set to 'resource: stack_state_service_templatewatcher: span: mimir-continuous-test:' with various operators.



Lightweight integration with VCS

Edit and preview dashboards in VSCode

Who?

Grafana administrators, Platform Engineers and service owners building dashboards.

Why?

There are great advantages to maintaining your dashboards as code, managed using version control, alongside other infrastructure config. However, dashboards are visual and you need a way to validate your updates before committing them.

What?

We're bringing Grafana into the developer's workflow with a VSCode plugin where you can make changes and review them directly on your workstation, within seconds of opening a dashboard file, with all the relevant data sources pulled in from your Grafana instance.

Feature highlights

- ✓ Backup and version control through VCS if enabled on file system
- ✓ Easily review a PR: just download a branch and open the dashboard on your workstation
- ✓ Hook up to your Grafana instance to access all your data sources
- ✓ Integrate into your CD pipeline like any other code change



New Developer Portal

One stop shop for Developer related information

Who?

Developers creating community, commercial or private plugins for Grafana or general contributing to the Grafana ecosystem.

Why?

Content for developers has been published to several disparate locations, increasing the barrier to entry for enhancements and extensions to Grafana.

What?

A single portal with a growing set of documentation and resources to help you easily develop and extend Grafana.

<https://grafana.com/developers>

Feature highlights

Initial launch Aug 2023 bringing together:

- ✓ Plugin Tools documentation
- ✓ UI Component Library
- ✓ Scenes
- ✓ Data Plane

The screenshot shows the Grafana Developers portal homepage. At the top, there's a navigation bar with a gear icon and the text "Create Merge Publish". Below this is a dark header with abstract geometric shapes (hexagons and circles) and a 3D cube icon. The main content area features four cards: "Plugin Documentation" (with an icon of a gear and a document), "UI Components" (with an icon of a document and a gear), "Scenes" (with an icon of a document and a camera), and "Data Plane" (with an icon of a document and a gear). Each card has a brief description below its title.





Mimir

Mimir



- For Metrics
- Prometheus -> Cortex -> Grafana Enterprise Metrics -> Mimir
- Scales to more than 1,000,000,000 Active Series
- Blazingly fast query performance
- Hard multi-tenancy, access control, and three-way replication
- Can ingest native OpenTelemetry, DataDog, Graphite, and Influx





Loki

Loki

Loki is a horizontally-scalable, highly-available, multi-tenant log aggregation system inspired by Prometheus.

- For Logs
- Following the same label-based system as Prometheus
 - Only index what you need often, query the rest
 - “Index the labels, query the data”
- Work with logs at scale, without the massive cost
 - Scalable low latency write path
 - Flexible schema on read
- Access logs with the same label sets as metrics
 - Turn logs into metrics, to make it easier & cheaper to work with them



The Loki storage model

2019-12-11T10:01:02.123456789Z {app="nginx", env="dev"} GET /about 1034 Debug "page not found"

Timestamp

with nanosecond precision

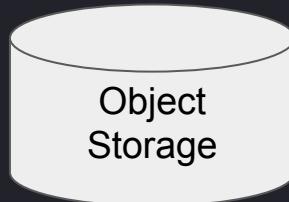
Labels>Selectors

key-value pairs

Content log line

indexed

Aka "the index"



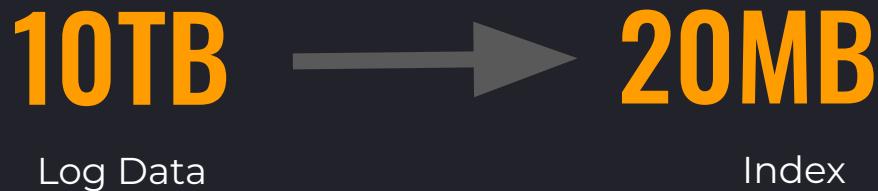
unindexed

Aka "chunks"



Efficient logging

Loki does not index the text of logs. Instead, entries are grouped into streams and indexed with Prometheus-style labels.



Think of it more like a table of contents than an index



Fast queries

1PB → 80TB → 8TB → 1TB/s max

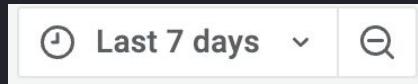
Raw Logs

Label selector

Timeframe

Brute force
search - heavily
parallelized

{ cluster="us-central1",
job=~"nginx*" }



|= “needle in
the haystack”



Example: One of Grafana Cloud Logs Larger Single Clusters

INGESTION

Per Hour

3TB

A day

60 TB

A month

1.8 PB

Active streams

4400

QUERIES

Per Hour

330 TB

A day

4 PB

A week

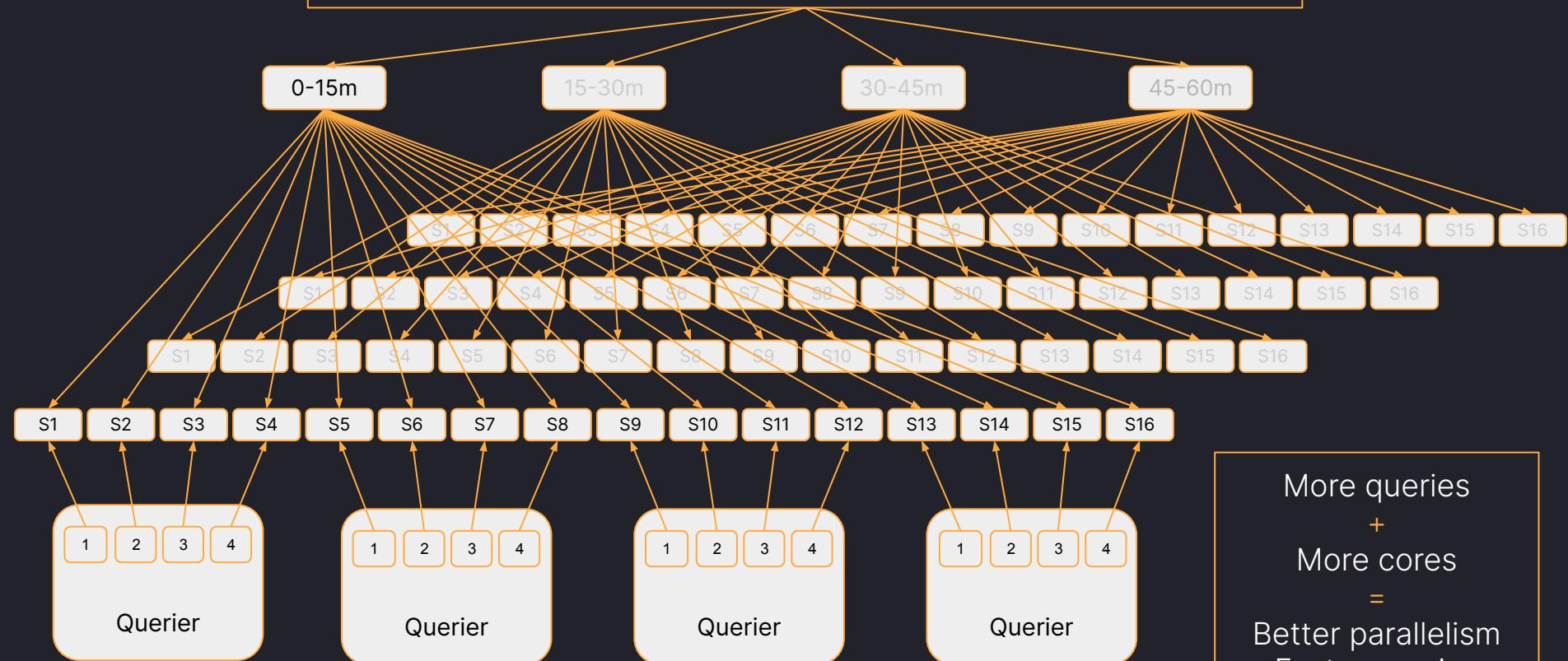
25 PB

Fastest query throughput

1TB/sec



{cluster="us-central1", job="loki-prod/querier"} |= "6114e9e58b14d5f0"



More queries
+
More cores
=

Better parallelism
Faster queries



Loki makes log collection easy



Grafana Agent



Promtail

- Targets discovery for Kubernetes, Syslog, files and more
- Automatically attach labels to your log lines
- Advanced pipeline mechanism for parsing, transforming and filtering your logs
- Build and expose custom metrics from your logs data

But Loki is open.



logstash



Lambda



docker

How to run Loki

0 1 0 0
0 0 0 0
1 0 1 1



Single Binary

- Testing
- Small installations without HA or horizontal scaling needs

SSD or Microservices

- Horizontal scalability
- Separate Read/Write paths
- Large installations



Grafana Cloud

- Batteries included:
 - Multi-tenant
 - Automatic scaling and upgrades
 - RBAC
 - Per tenant QoS



Enterprise Logs

- Self hosted
- Supported by Grafana Labs
- Security out of the box
- Part of the GES stack



Tempo

Tempo

Easy-to-operate, high-scale, and cost-effective distributed tracing system

- For Traces
- Historic problem: Traces require extremely rich metadata for analysis
 - Expensive, slow, and mandates sampling
- Exemplars: Leverage the extracted logs & metrics
 - Exemplars work at Google scale, with the ease of Grafana
 - Native to Prometheus, Cortex, Thanos, and Loki
- Index and search by labelsets available for those who need it
- 100% compatible with OpenTelemetry Tracing, Zipkin, Jaeger





Grafana Beyla

Beyla - eBPF application auto-instrumentation

- Zero code change auto instrumentation
- Leverages eBPF technology
- GA since November 2023
 - <https://github.com/grafana/beyla>
- Currently support for ootb RED Metrics



Faro



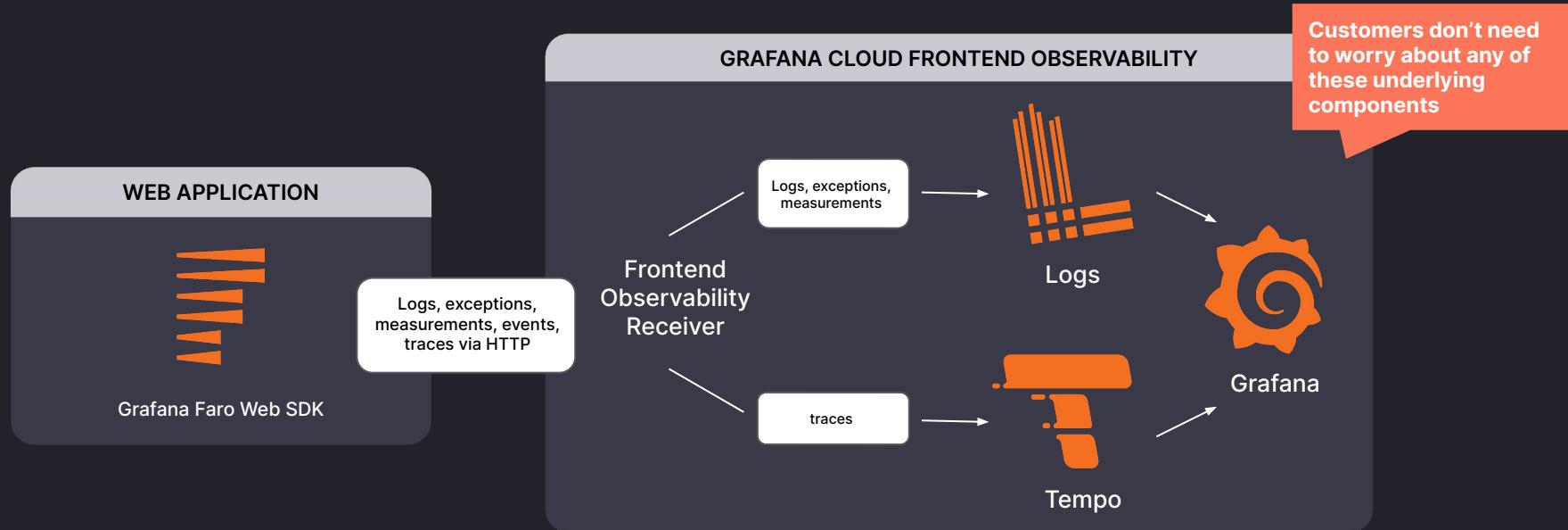
Faro

- For Frontend/Browser Monitoring - **RealUserMonitoring**
- Providing visibility into the performance experience of real users interacting with your application
 - Expensive, slow, and mandates sampling
- RUM tools bridge the gap between application performance and the impacts on User Experience
- Understand how your optimizations are improving the UX



How Frontend Observability works under the hood

- The **Grafana Faro Web SDK** collects telemetry & forwards it to the **collector endpoint in Grafana Cloud**
- The **Grafana Cloud Collector Endpoint** processes and forwards telemetry to the appropriate **logs and traces backends in Grafana Cloud**
- The **Grafana Cloud Frontend Observability application** provides out-of-box dashboards and curated workflow





Grafana
Pyroscope

Pyroscope

- For Profiling
- We announced the acquisition March 15th
 - <https://github.com/grafana/pyroscope>
- Profiles
 - “How much CPU & RAM am I spending in what areas of the code?”
 - “...and how does this change over time?”
- Go: pprof
- Java: <https://github.com/grafana/JPProf>





Grafana
k6

Modern teams are shifting performance testing left

	OLD WAY	MODERN WAY
Release frequency	Quarterly or biannually	Weekly
Who is responsible for performance testing	QA	QA Engineers/SDETs, Developers, SREs
What to test	High-risk components, user stories	AND microservices, infrastructure, unexpected failures
When and how testing is done	Manually before releases	Automatically as part of CI/CD

But legacy testing tools have not kept pace...

k6 OSS

A screenshot of a terminal window titled "k6-tests: k6 run script.js". The window displays performance test results for a script named "script.js". The results include metrics like execution time, output, duration, iterations, and various network and system statistics. A large "JS" logo is overlaid on the bottom left of the terminal window.

```
k6-tests: k6 run script.js
k6-tests --- bash --- 142x33

execution: local
output: -
script: script.js

duration: 1m0s, iterations: -
vus: 100, max: 100

done [====] 1m0s / 1m0s

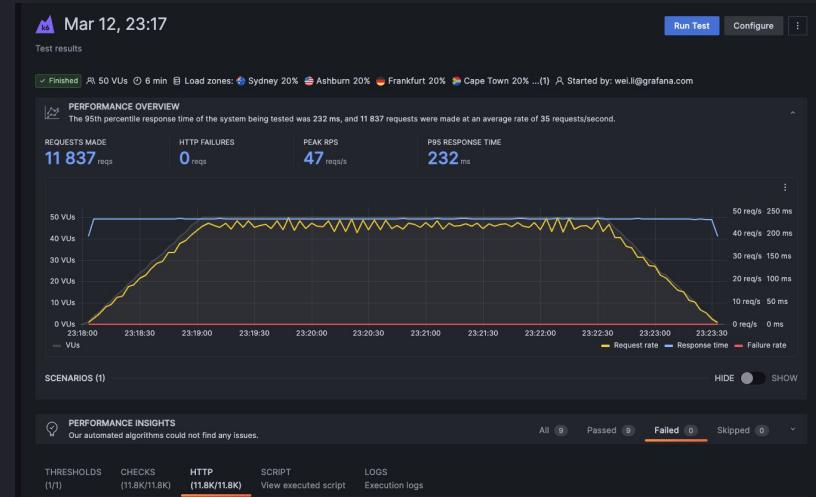
data_received.....: 148 MB 2.5 MB/s
data_sent.....: 1.8 MB 17 kB/s
http_req_blocked.: avg=1.92ms min=1µs med=5µs max=288.72ms p(90)=11µs p(95)=17µs
http_req_connecting.: avg=1.01ms min=1µs med=4µs max=165.44ms p(90)=0µs p(95)=0µs
http_req_duration.: avg=143.14ms min=12.87ms med=165.03ms max=1.16s p(90)=164.2ms p(95)=177.75ms
http_req_receiving.: avg=1.55ms min=1.49µs med=2.11ms max=1.01s p(90)=9.25ms p(95)=1.8ms
http_req_resending.: avg=2.01ms min=1.91µs med=2.42ms max=3.89ms p(90)=20.05ms p(95)=5.35ms
http_req_tls_handshaking.: avg=0.91µs min=0µs med=0µs max=0µs p(90)=0µs p(95)=0µs
http_req_waiting.: avg=137.57ms min=111.44ms med=132.59ms max=589.4ms p(90)=159.95ms p(95)=169.41ms
http_resps.....: 13491 272.84kB/s
iteration_duration.....: avg=45.48ms min=13.05ms med=436.36ms max=1.48s p(90)=464.94ms p(95)=479.66ms
iterations.....: 13410 223.49kB/s
vus.....: 100 min=100 max=100
vus_max.....: 100 min=100 max=100

k6-tests: |
```

JS

Grafana Cloud k6

NEW



- New experience in Grafana Cloud launched on March 21
- Built on the legacy commercial product - "k6 Cloud"
- Performance testing and observability in one platform

Meet our community in the official meetup groups

The screenshot shows the Meetup group page for "Grafana & Friends Munich". The header features the Meetup logo and navigation links for "Neue Gruppe gründen", "Einloggen", and "Registrieren". Below the header, it states "Mitglied von Grafana – 39 Gruppen". The main content area displays the group's name "Grafana & Friends Munich" and its location "München, Deutschland". It also shows "235 Mitglieder · Öffentliche Gruppe" and "Organisiert von Grafana C. und 2 andere". A large yellow banner at the top features the group's name and a screenshot of a Grafana dashboard showing various metrics like CPU, Memory, and Disk usage. Below the banner, there are tabs for "Über", "Events", "Mitglieder", "Fotos", "Diskussionen", and "Mehr", along with a red button "Der Gruppe beitreten" and a three-dot menu icon.

<https://www.meetup.com/grafana-and-friends-munich/>

<https://www.meetup.com/grafana-and-friends-vienna/>



<https://www.meetup.com/grafana-and-friends-berlin/>