Grafana + CLOUD NATIVE COMPUTING FOUNDATION

APAC Meetup

Hybrid meetup with the Grafana Prometheus Squad

Richi Hartmann
Director of Community
Grafana Labs

Harish Madhavan
CNCF Singapore Host

Ganesh Vernekar
Sr. Software Engineer
Prometheus squad
Grafana Labs

Tuesday, 11th April
In-Person in Singapore | 5:30PM SGT

Virtual via Zoom
3:30PM IST | 6:00PM SGT | 8:00PM AEST

Grafana Labs

LGTM Stack

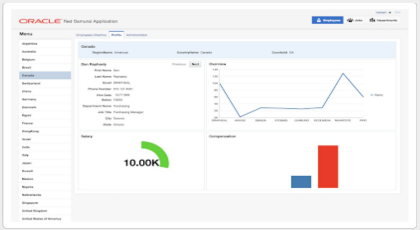
Or: Philosophy of Observability

Richard “RichiH” Hartmann

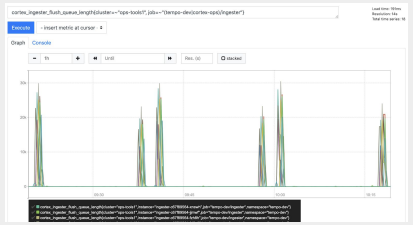

GOLD to go
The gold ATM
www.gold-to-go.com

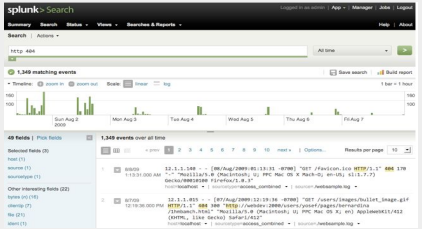
Today's reality:
Disparate systems. Disparate data.

ORACLE



APPDYNAMICS





Grafana

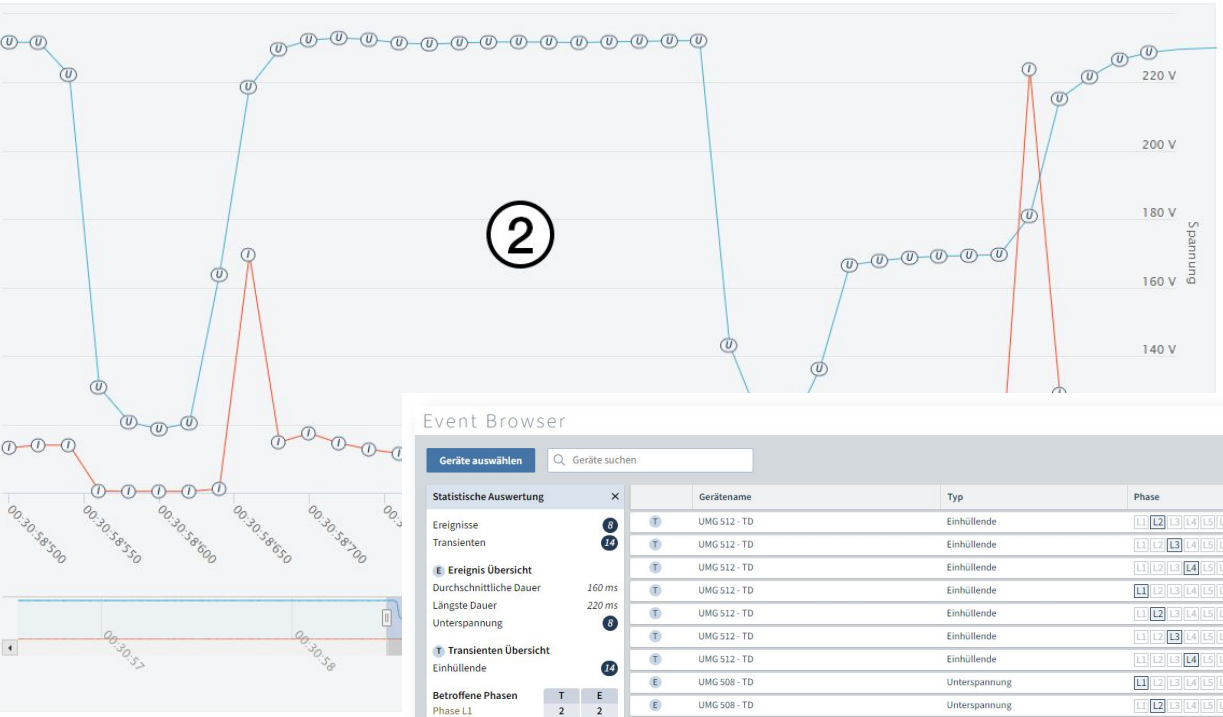
splunk>

elasticsearch

Back to the basics

Let’s rethink this

How humanity deals with data



Event Browser

Geräte auswählen | Geräte suchen | 19.12.2019 - 19.12.2019 00:00 - 23:59

Statistische Auswertung

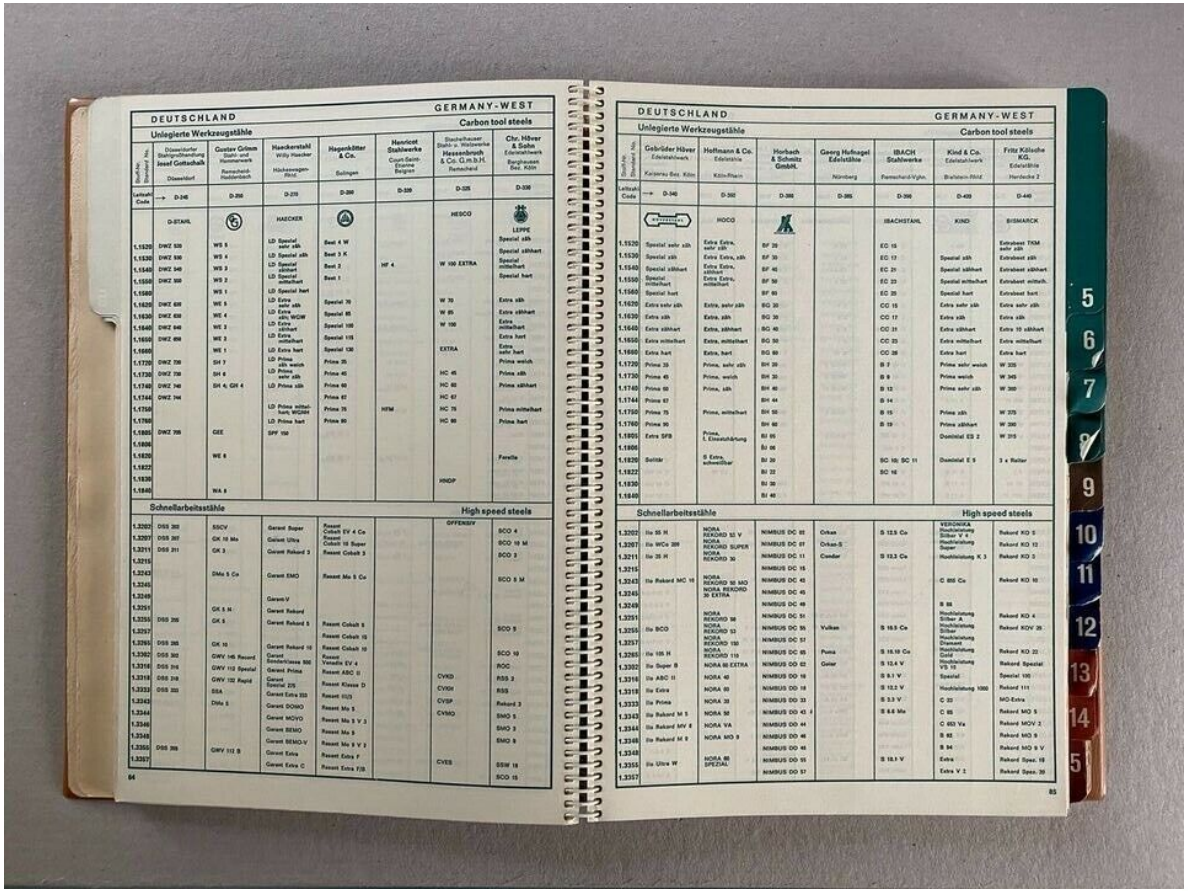
Ereignisse	8
Transienten	14
Ereignis Übersicht	
Durchschnittliche Dauer	160 ms
Längste Dauer	220 ms
Unterspannung	8
Transienten Übersicht	
Einhüllende	14

Betroffene Phasen	T	E
Phase L1	2	2
Phase L2	4	2
Phase L3	4	2
Phase L4	4	2

	Gerätename	Typ	Phase	Start	Ende	Dauer	Wert	
T	UMG 512 - TD	Einhüllende	L2	19.12.2019 01:15:40'663	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L3	19.12.2019 01:15:40'663	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L4	19.12.2019 01:15:40'663	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L1	19.12.2019 01:15:39'345	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L2	19.12.2019 01:15:39'345	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L3	19.12.2019 01:15:39'345	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L4	19.12.2019 01:15:39'345	---	---	---	Analysieren
E	UMG 508 - TD	Unterspannung	L1	19.12.2019 00:30:59'979	19.12.2019 00:30:59'199	220 ms	117.836 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L2	19.12.2019 00:30:59'979	19.12.2019 00:30:59'199	220 ms	117.806 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L3	19.12.2019 00:30:59'979	19.12.2019 00:30:59'199	220 ms	117.825 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L4	19.12.2019 00:30:59'979	19.12.2019 00:30:59'199	220 ms	117.830 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L1	19.12.2019 00:30:58'559	19.12.2019 00:30:58'659	100 ms	118.640 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L2	19.12.2019 00:30:58'559	19.12.2019 00:30:58'659	100 ms	118.612 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L3	19.12.2019 00:30:58'559	19.12.2019 00:30:58'659	100 ms	118.622 V (MIN)	Analysieren
E	UMG 508 - TD	Unterspannung	L4	19.12.2019 00:30:58'559	19.12.2019 00:30:58'659	100 ms	118.631 V (MIN)	Analysieren
T	UMG 512 - TD	Einhüllende	L2	19.12.2019 00:28:40'565	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L3	19.12.2019 00:28:40'565	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L4	19.12.2019 00:28:40'565	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L1	19.12.2019 00:28:40'130	---	---	---	Analysieren
T	UMG 512 - TD	Einhüllende	L2	19.12.2019 00:28:40'130	---	---	---	Analysieren

20 | 1 2

Grafana Labs



DEUTSCHLAND — GERMANY-WEST

Unlegierte Werkzeugstähle — Carbon tool steels

Stahlnr. / Standard No.	Düsseldorfer Stahlgroßhandlung Josef Gottschalk, Düsseldorf	Custav Grimm Stahl- und Kunststoffwerk, Remscheid-Hasten/Bonn	Haeckerstahl Willy Haecker, Hückeswagen-Rhld.	Hagenköttter & Co., Solingen	Henricot Stahlwerke Court-Saint-Etienne, Belgien	Stahlschmiede Stahl u. Walzwerke Hessenbruch & Co. G.m.b.H., Remscheid	Chr. Höver & Sohn Edelstahlwerk, Burghausen bei Köln
Leitzahl / Code	→ D-345	D-350	D-370	D-390	D-330	D-335	D-330
	D-STAHL		HAECKER			HESCO	LEPPE
1.1520	DWZ 520	WS 5	LD Special sehr zäh	Best 4 W			Special zäh
1.1530	DWZ 530	WS 4	LD Special zäh	Best 3 K			Special zähhart
1.1540	DWZ 540	WS 3	LD Special zähhart	Best 2	HF 4	W 100 EXTRA	Special mittelhart
1.1550	DWZ 550	WS 2	LD Special mittelhart	Best 1			Special hart
1.1560		WS 1	LD Special hart				
1.1620	DWZ 620	WE 5	LD Extra sehr zäh	Special 70		W 70	Extra zäh
1.1630	DWZ 630	WE 4	LD Extra zäh, WCW	Special 85		W 85	Extra zähhart
1.1640	DWZ 640	WE 3	LD Extra zähhart	Special 100		W 100	Extra mittelhart
1.1650	DWZ 650	WE 2	LD Extra mittelhart	Special 115			Extra hart
1.1660		WE 1	LD Extra hart	Special 130		EXTRA	Extra sehr hart
1.1720	DWZ 720	SH 7	LD Prima zäh weich	Prima 35			Prima weich
1.1730	DWZ 730	SH 6	LD Prima sehr zäh	Prima 45		HC 45	Prima zäh
1.1740	DWZ 740	SH 4, GH 4	LD Prima zäh	Prima 60		HC 60	Prima zähhart
1.1744	DWZ 744			Prima 67		HC 67	
1.1750			LD Prima mittelhart, WQ5M	Prima 75	HFM	HC 75	Prima mittelhart
1.1760			LD Prima hart	Prima 90		HC 85	Prima hart
1.1805	DWZ 705	GEE	SPF 150				
1.1806							
1.1820		WE 6					Feralle
1.1822							
1.1830						HNDP	
1.1840		WA 8					

Schnellarbeitsstähle — High speed steels

1.3202	DSS 202	SSCV	Garant Super	Rasant Cobalt EV 4 Co		OFFENSIV	SCO 4
1.3207	DSS 207	GK 10 Mo	Garant Ultra	Rasant Cobalt 10 Super			SCO 10 M
1.3211	DSS 211	GK 3	Garant Rekord 3	Rasant Cobalt 3			SCO 3
1.3215							
1.3243		DMo 5 Co	Garant EMO	Rasant Mo 5 Co			SCO 5 M
1.3245							
1.3249			Garant V				
1.3251		GK 5 Ni	Garant Rekord				
1.3255	DSS 255	GK 5	Garant Rekord 5	Rasant Cobalt 5			SCO 5
1.3257				Rasant Cobalt 15			
1.3265	DSS 265	GK 10	Garant Rekord 10	Rasant Cobalt 10			SCO 10
1.3302	DSS 302	GWV 145 Record	Garant Sonderklasse 500	Rasant Vanadia EV 4			ROC
1.3310	DSS 310	GWV 112 Special	Garant Prima	Rasant ABC II		CVKD	RSS 2
1.3318	DSS 318	GWV 122 Rapid	Garant Special 215	Rasant Klasse D		CVDH	RSS
1.3333	DSS 333	SSA	Garant Extra 333	Rasant HS5		CVSP	Rekord 3
1.3343		DMo 5	Garant DOMO	Rasant Mo 5		CVMO	SMO 5
1.3344			Garant MOVO	Rasant Mo 5 V 2			SMO 3
1.3348			Garant REMO	Rasant Mo 9			SMO 8
1.3355	DSS 355	GWV 112 S	Garant REMO-V	Rasant Mo 9 V 2			
1.3357			Garant Extra	Rasant Extra F		CVES	SSW 18
			Garant Extra C	Rasant Extra F28			SCO 15

64

DEUTSCHLAND — GERMANY-WEST

Unlegierte Werkzeugstähle — Carbon tool steels

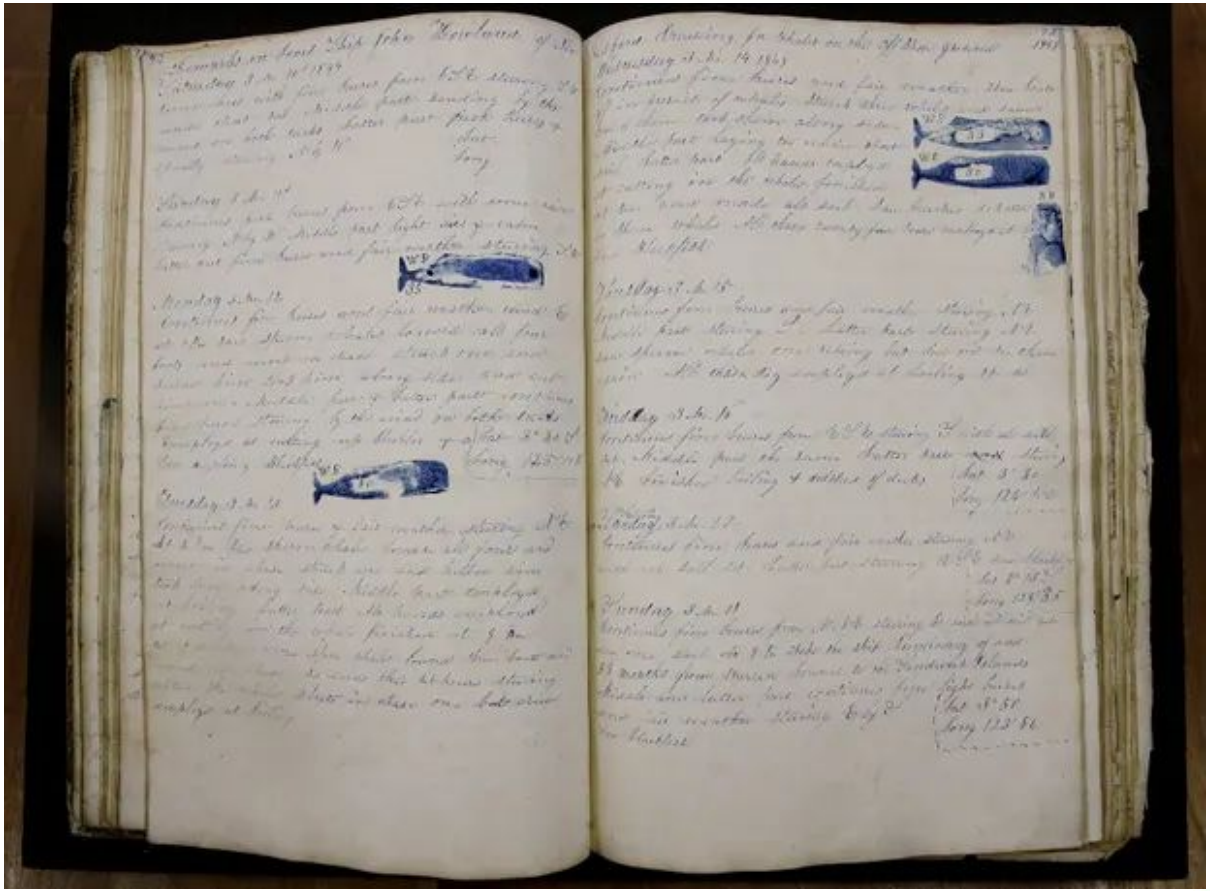
Stahlnr. / Standard No.	Gebrüder Höver Edelstahlwerk, Kaiserau bez. Köln	Hoffmann & Co. Edelstahle, Köln-Rhein	Horbach & Schmitz GmbH,	Georg Hofnagel Edelstähle, Nürnberg	IBACH Stahlwerke, Remscheid-Vgbe.	Kind & Co. Edelstahlwerk, Bretten-Rhld.	Fritz Kölsche KG. Edelstähle Hardecke 2
Leitzahl / Code	→ D-340	D-362	D-366	D-365	D-360	D-400	D-440
	HÖVER	HOCO			IBACHSTAHL	KIND	BISMARCK
1.1520	Special sehr zäh	Extra Extra, sehr zäh	BF 35		EC 15		Extrabest TKM sehr zäh
1.1530	Special zäh	Extra Extra, zäh	BF 30		EC 17	Special zäh	Extrabest zäh
1.1540	Special zähhart	Extra Extra, zähhart	BF 40		EC 21	Special zähhart	Extrabest zähhart
1.1550	Special mittelhart	Extra Extra, mittelhart	BF 50		EC 33	Special mittelhart	Extrabest mittelh.
1.1560	Special hart		BF 60		EC 35	Special hart	Extrabest hart
1.1620	Extra sehr zäh	Extra, sehr zäh	BG 30		CC 15	Extra sehr zäh	Extra sehr zäh
1.1630	Extra zäh	Extra, zäh	BG 35		CC 17	Extra zäh	Extra zäh
1.1640	Extra zähhart	Extra, zähhart	BG 40		CC 21	Extra zähhart	Extra 10 zähhart
1.1650	Extra mittelhart	Extra, mittelhart	BG 50		CC 33	Extra mittelhart	Extra mittelhart
1.1660	Extra hart	Extra, hart	BG 60		CC 38	Extra hart	Extra hart
1.1720	Prima 35	Prima, sehr zäh	BH 35		B 7	Prima sehr weich	W 335
1.1730	Prima 45	Prima, weich	BH 35		B 9	Prima weich	W 345
1.1740	Prima 60	Prima, zäh	BH 40		B 13	Prima sehr zäh	W 360
1.1744	Prima 67		BH 44		B 14		
1.1750	Prima 75	Prima, mittelhart	BH 50		B 15	Prima zäh	W 370
1.1760	Prima 90		BH 60		B 19	Prima zähhart	W 390
1.1805	Extra SFB	Prima, f. Einsatzhärtung	BJ 05			Dominiat ES 2	W 315
1.1806			BJ 06				
1.1820	Solida	S Extra, schweißbar	BJ 20		SC 10; SC 11	Dominiat E 9	3 x Reiter
1.1822			BJ 22		SC 16		
1.1830			BJ 30				
1.1840			BJ 40				

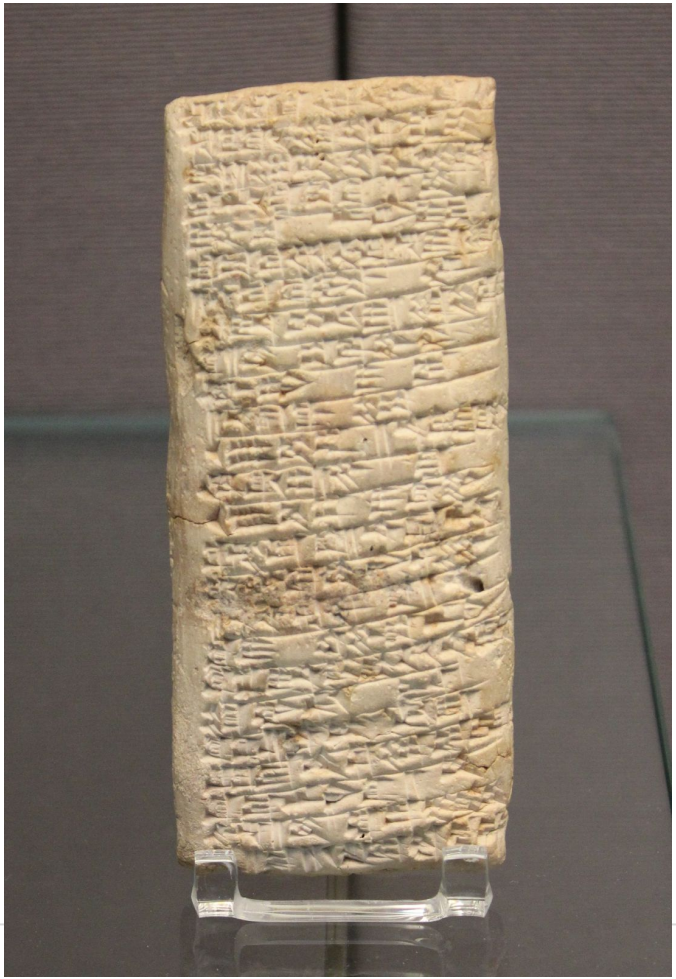
Schnellarbeitsstähle — High speed steels

1.3202	Ila 55 H	NORA REKORD 53 V	NIMBUS DC 05	Orkan	S 12,5 Co	VERONIKA Hochleistung Silber V 5	Rekord KO 5
1.3207	Ila WCo 209	NORA REKORD SUPER	NIMBUS DC 07	Orkan-S		Hochleistung Super	Rekord KO 10
1.3211	Ila 35 H	NORA REKORD 30	NIMBUS DC 11	Condor	S 12,3 Co	Hochleistung K 3	Rekord KO 3
1.3215			NIMBUS DC 15				
1.3243	Ila Rekord MC 10	NORA REKORD 50 MO	NIMBUS DC 45			C 850 Co	Rekord KO 10
1.3245		NORA REKORD 35 EXTRA	NIMBUS DC 45				
1.3249			NIMBUS DC 49			B 86	
1.3251		NORA REKORD 58	NIMBUS DC 51			Hochleistung Silber V 4	Rekord KO 4
1.3255	Ila SCO	NORA REKORD 53	NIMBUS DC 55	Vulkan	S 10,5 Co	Hochleistung Silber	Rekord KOV 25
1.3257		NORA REKORD 150	NIMBUS DC 57			Hochleistung Diamant	
1.3265	Ila 155 H	NORA REKORD 110	NIMBUS DC 65	Puma	S 10,10 Co	Hochleistung Gold	Rekord KO 22
1.3302	Ila Super B	NORA 66 EXTRA	NIMBUS DD 62	Golar	S 13,4 V	Hochleistung VS 10	Rekord Special
1.3310	Ila ABC II	NORA 40	NIMBUS DD 10		S 9,1 V	Special	Special 100
1.3318	Ila Extra	NORA 60	NIMBUS DD 18		S 12,2 V	Hochleistung 1000	Rekord 111
1.3333	Ila Prima	NORA 33	NIMBUS DD 33		S 3,3 V	C 33	MO-Extra
1.3343	Ila Rekord M 5	NORA 50	NIMBUS DD 43		S 6,6 Mo	C 65	Rekord MO 5
1.3344	Ila Rekord MV 8	NORA VA	NIMBUS DD 44			C 653 Va	Rekord MOV 2
1.3348	Ila Rekord M 9	NORA MO 9	NIMBUS DD 48			B 65	Rekord MO 9
1.3348			NIMBUS DD 48			B 94	Rekord MO 9 V
1.3355	Ila Ultra W	NORA 88 SPEZIAL	NIMBUS DD 55		S 18,1 V	Extra	Rekord Spez. 18
1.3357			NIMBUS DD 57			Extra V 2	Rekord Spez. 35

83

5 6 7 8 9 10 11 12 13 14 15





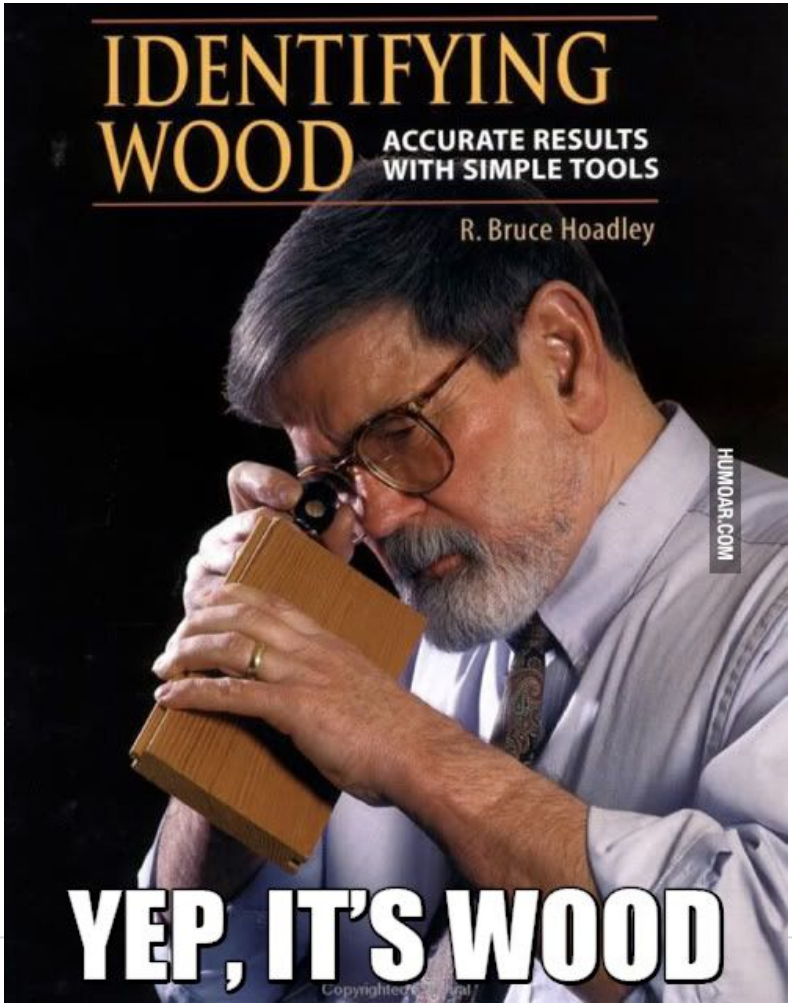
Humanity has optimized detailed accounts into key events into numbers for millenia, again and again

Observability & SRE

Or: Buzzwords, and their useful parts

Observability, the buzzword

- Cool new term, almost meaningless by now, what does it mean?
  - Pitfall alert: Cargo culting
  - It's about changing the behaviour, not about changing the name
- "Monitoring" has taken on a meaning of collecting, not using data
  - One extreme: Full text indexing
  - Other extreme: Data lake
- "Observability" is about enabling humans to understand complex systems
  - Ask new questions on the fly
  - Ask **why** it's not working instead of just knowing that it's not


IDENTIFYING WOOD
ACCURATE RESULTS WITH SIMPLE TOOLS
R. Bruce Hoadley
HUMOAR.COM
YEP, IT'S WOOD

[...] observations are [...] approximations to the truth [...] this can be accomplished in no other way than by a suitable combination of more observations than the number absolutely requisite for the determination of the unknown quantities

Carl Friedrich Gauß, 1809

Observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs.

Rudolf Emil Kálmán, 1960

Complexity

- Fake complexity, a.k.a. bad design
  - Can be reduced
- Real, system-inherent complexity
  - Can be moved (monolith vs client-server vs microservices)
  - Must be compartmentalized (service boundaries)
  - Should be distilled meaningfully (observability…)

Services

- What's a service?
  - Compartmentalized complexity, with an interface
  - Different owners/teams
  - Contracts define interfaces
- Why "contract": Shared agreement which MUST NOT be broken
  - Internal and external customers rely on what you build and maintain
- Other common term: layer
  - The Internet would not exist without network layering
  - Enables innovation, parallelizes human engineering
- Other examples: CPUs, hard drive, compute nodes, your lunch

Cloud-native vs client-server vs mainframe vs...

- A mainframe application and a microservices fleet are fundamentally the same
  - You can *move* system-inherent complexity, but...
- Microservices broke up old service and system boundaries
  - Enabling horizontal scalability, arguably at the cost of vertical scalability
- Previous-generation tooling is designed to understand system complexity along existing service boundaries
  - Cloud native tooling is able to deal with this increased complexity
  - NB: This means previous-generation complexity is even easier to observe

SRE, an instantiation of DevOps

- At its core: Align incentives across the org
  - Error budgets allow devs, ops, PMs, etc. to optimize for shared benefits
- Measure it!
  - SLI: Service Level Indicator: What you measure
  - SLO: Service Level Objective: What you need to hit
  - SLA: Service Level Agreement: When you need to pay
- Discern between different SLIs
  - Primary: service-relevant, for alerting
  - Secondary: informational, debugging, might be underlying's primary

Shared understanding

- Everyone uses the same tools & dashboards
  - Shared incentive to invest into tooling
  - Pooling of institutional system knowledge
  - Shared language & understanding of services

Alerting

- Customers care about services being up, not about individual components

Anything currently or imminently impacting customer service must be alerted upon

But nothing(!) else

Prometheus

Prometheus 101

- Inspired by Google's Borgmon
- Time series database
- Rich ecosystem, 1,000s of instrumentations & exporters
- Cloud-native default

Time series

- Time series are recorded values which change over time
- Individual events are usually merged into counters and/or histograms
- Changing values are recorded as gauges
- Typical examples
  - Requests to a webserver (counter)
  - Temperatures in a datacenter (gauge)
  - Service latency (histograms)

Cloud-native default

- Kubernetes =~ Borg
- Prometheus =~ Borgmon
- Google couldn't have run Borg without Borgmon
- Kubernetes & Prometheus are designed and written with each other in mind

Main selling points

- Highly dynamic, built-in service discovery
- No hierarchical model, n-dimensional label set
- PromQL: for processing, graphing, alerting, and export
- Simple operation
- Highly efficient

Super easy to emit, parse & read

```
http_requests_total{env="prod",method="post",code="200"} 1027
http_requests_total{env="prod",method="post",code="400"} 3
http_requests_total{env="prod",method="post",code="500"} 12
http_requests_total{env="prod",method="get",code="200"} 20
http_requests_total{env="test",method="post",code="200"} 372
http_requests_total{env="test",method="post",code="400"} 75
```

PromQL

All partitions in my entire infrastructure with more than 100GB capacity that are not mounted on root?

```
node_filesystem_bytes_total{mountpoint!="/"} / 1e9 > 100
```

```
{device="sda1", mountpoint="/home", instance="10.0.0.1"} 118.8
{device="sda1", mountpoint="/home", instance="10.0.0.2"} 118.8
{device="sdb1", mountpoint="/data", instance="10.0.0.2"} 451.2
{device="xdvc", mountpoint="/mnt", instance="10.0.0.3"} 320.0
```

PromQL

What's the ratio of request errors across all service instances?

```
sum by(path) (rate(http_requests_total{status="500"}[5m])) /
 sum by(path) (rate(http_requests_total[5m]))

 {path="/status"} 0.0039
 {path="/"} 0.0011
 {path="/api/v1/topics/:topic"} 0.087
 {path="/api/v1/topics} 0.0342
```

New features

- Remote Write Receiver (v2.25 (feature flag) v2.33 (stable))
- Trigonometric functions (v2.31)
- Agent mode (v2.32)
- Long term support versions (v2.27)
- Out of order ingestion (v2.39)

Next highlight feature: Native histograms

Prometheus scale

- 1,000,000+ samples/second no problem on current hardware
- ~200,000 samples/second/core
- 16 bytes/sample compressed to 1.36 bytes/sample
- Reliable into the tens of millions of active series

Mimir

Mimir

- For **M**etrics
- 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

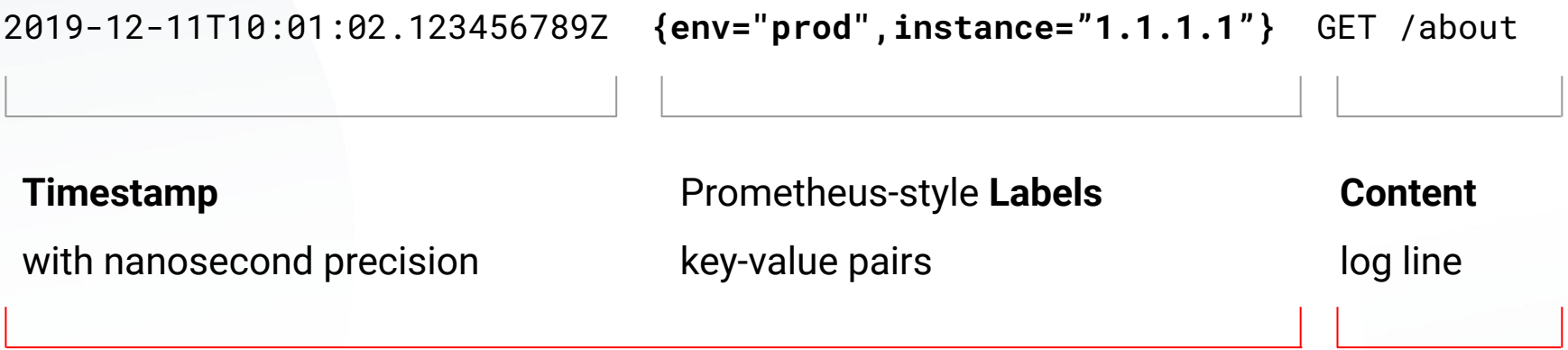
Mimir @ Grafana

- 1,000,000,000 Active Series - in one cluster
- 1,500 machines
- 7,000 CPU cores
- 30 TiB RAM

Loki

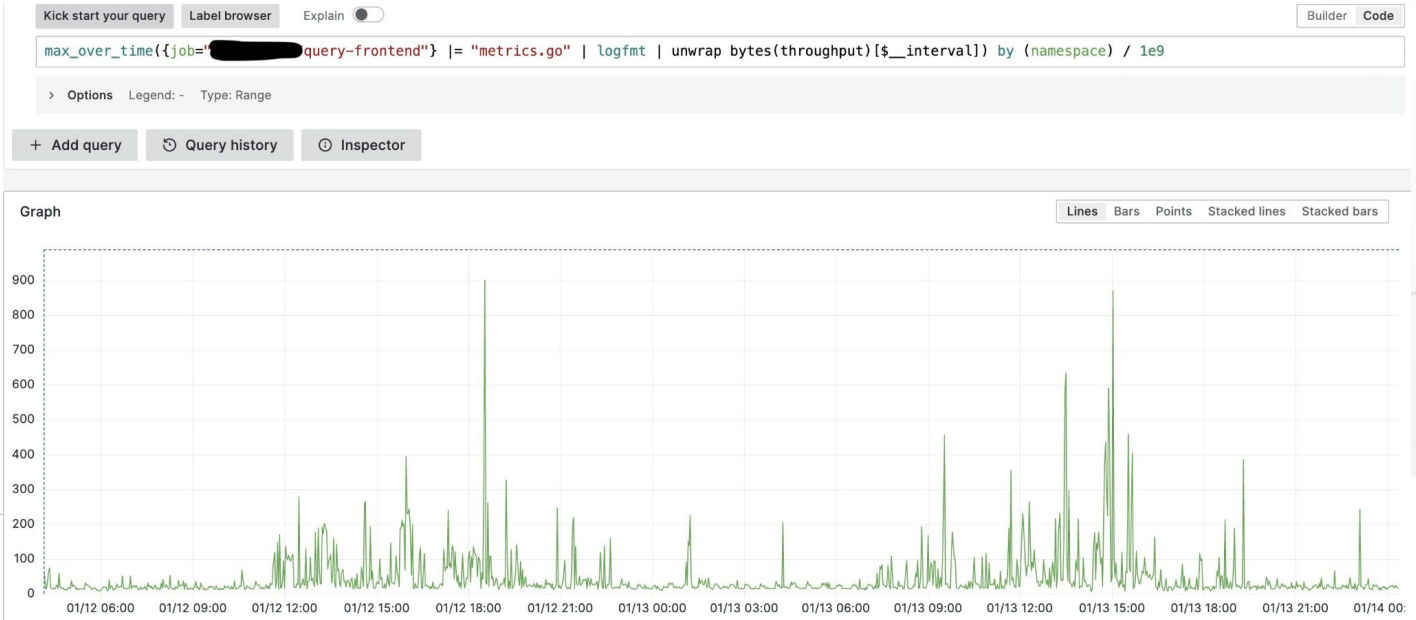
Loki 101

- For **L**ogs
- 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


2019-12-11T10:01:02.123456789Z
{env="prod",instance="1.1.1.1"}
GET /about
Timestamp
with nanosecond precision
Prometheus-style Labels
key-value pairs
Content
log line
indexed
unindexed

Loki @ Grafana Labs

- Largest user cluster (as of 2023-01): 180 TiB per day
- Queries regularly peak at **900GB/s**
  - Query 10TB in 12 seconds, including complex processing of result sets



Tempo

Tempo

- For **T**races
- 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

Tempo @ Grafana Labs (2023-04)

- 1,500,000 samples per second @ 450 MiB/s
  - 560 MiB/s peak
- 14-day retention @ 3 copies stored
- Latencies:
  - p99 - 2.5s
  - p90 - 2.3s
  - p50 - 1.6s

Grafana Pyroscope

Pyroscope

- For **P**rofiling
- 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

Data (and cost) savings

Logs to metrics

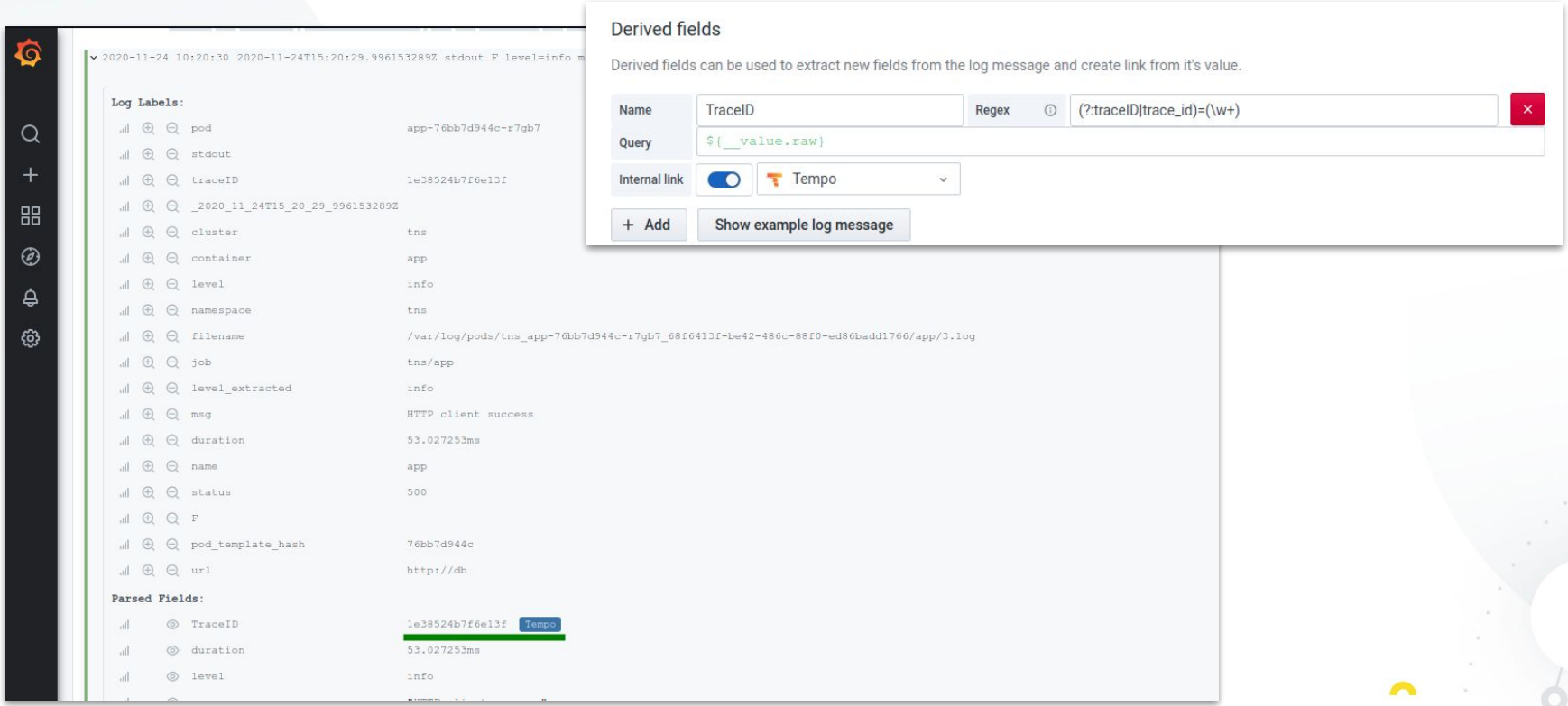
- Full text indexing: 10 TiB logs → ~20 TiB index
- Loki: 10 TiB logs → ~200 MiB index

- Logs @ Grafana ~600 Byte average per line
- Metrics ~1.36 Byte per metric sample

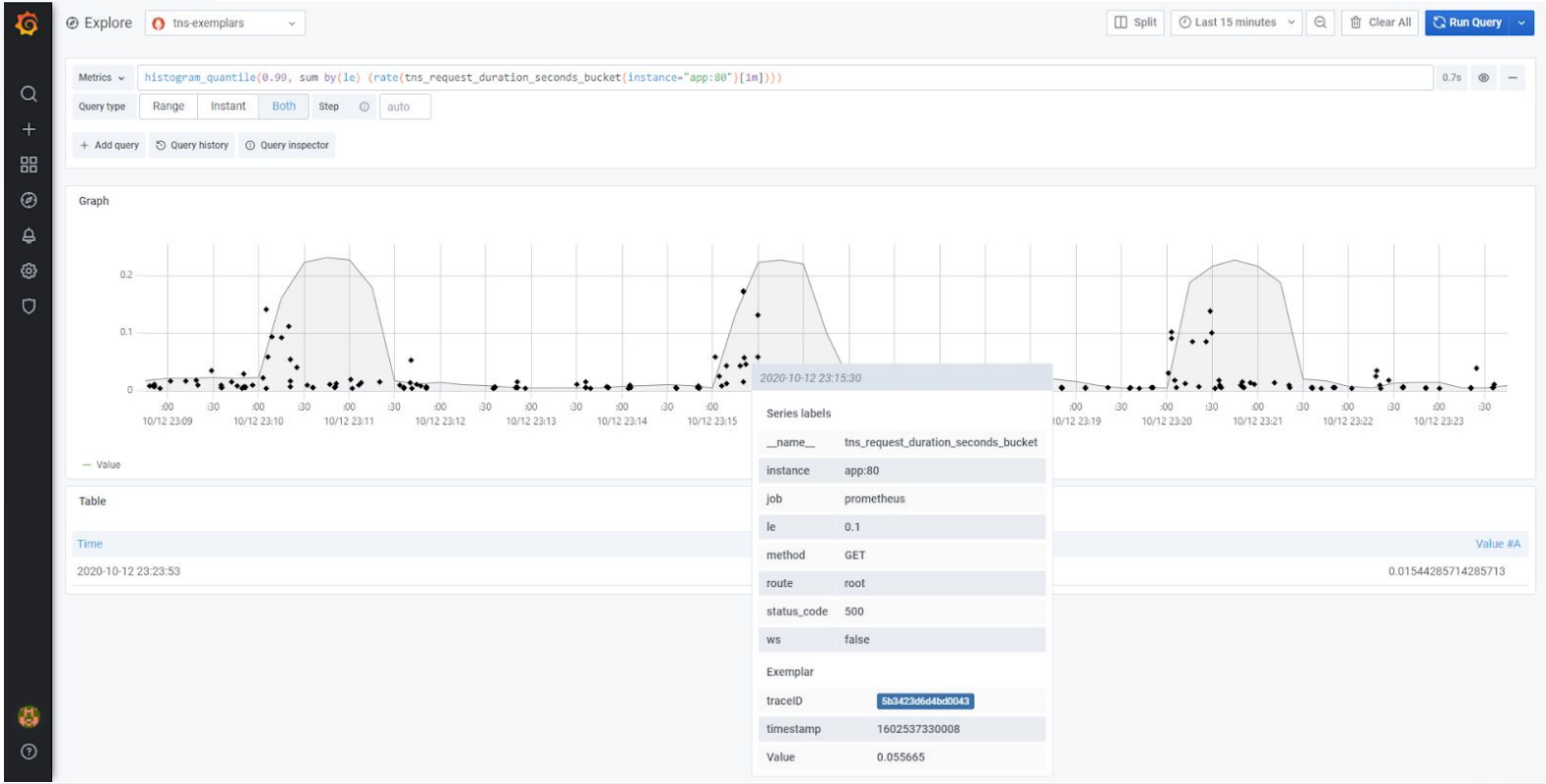
→ 99.8% reduction in storage size for first log line
~100% for every follow-up log line

Bringing it together

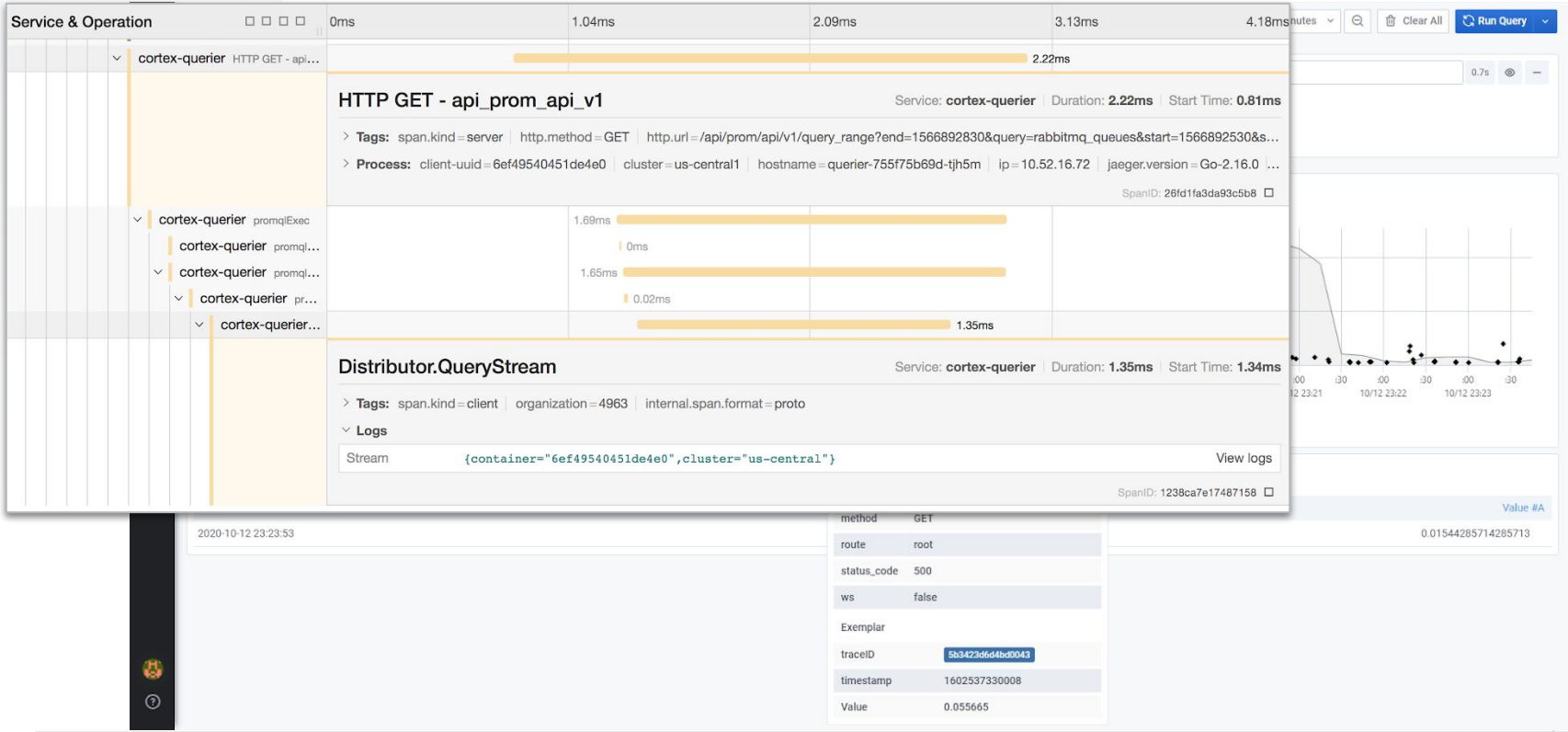
From logs to traces



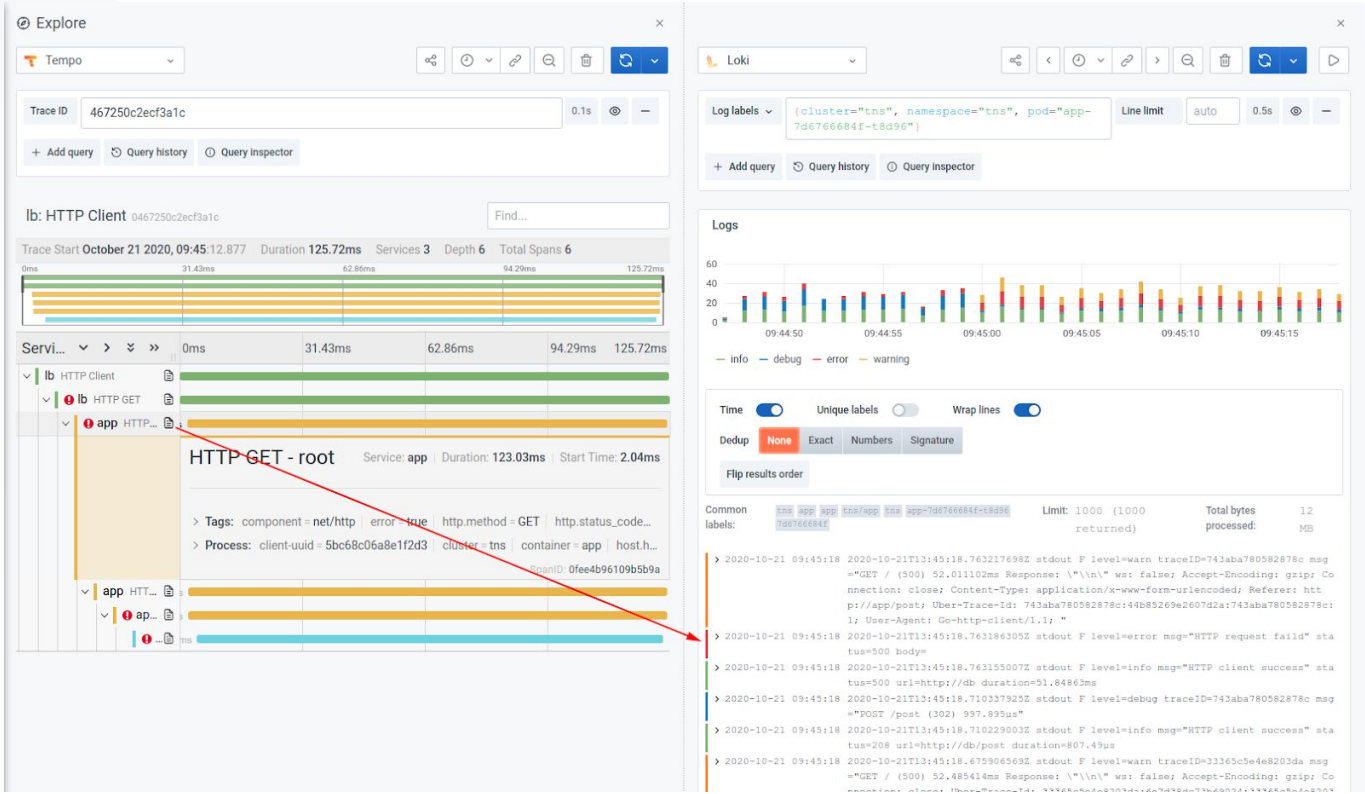
From metrics to traces

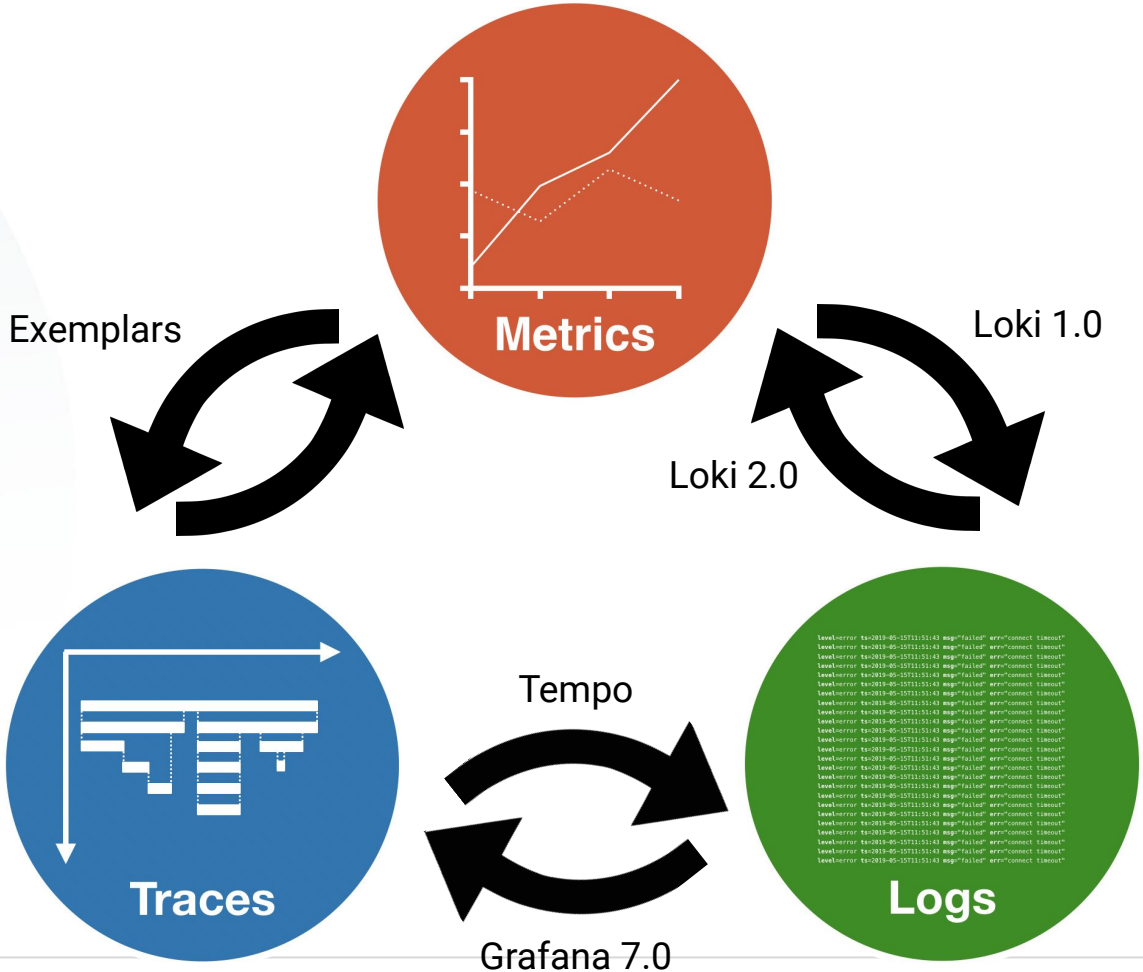


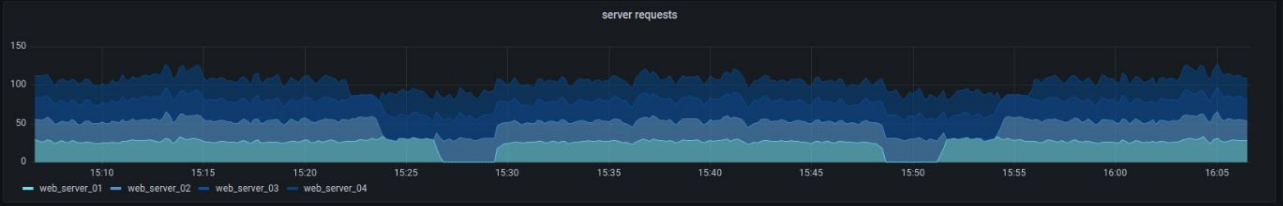
From metrics to traces

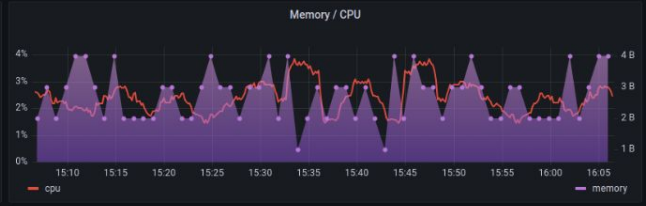


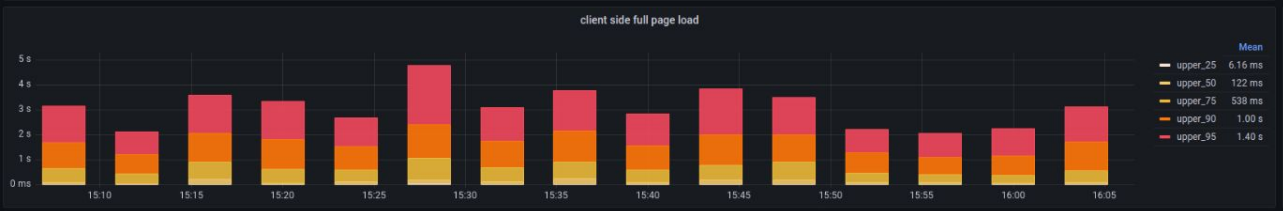
...and from traces to logs

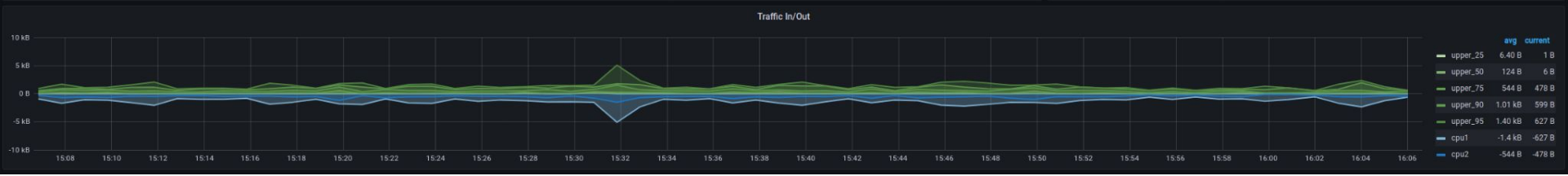



Metrics
Exemplars
Loki 1.0
Loki 2.0
Traces
Tempo
Logs
Grafana 7.0

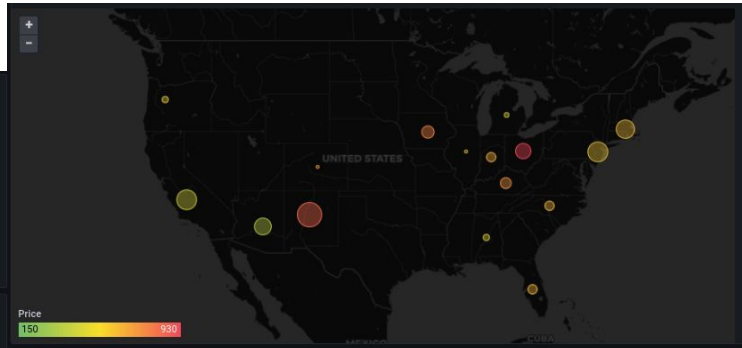


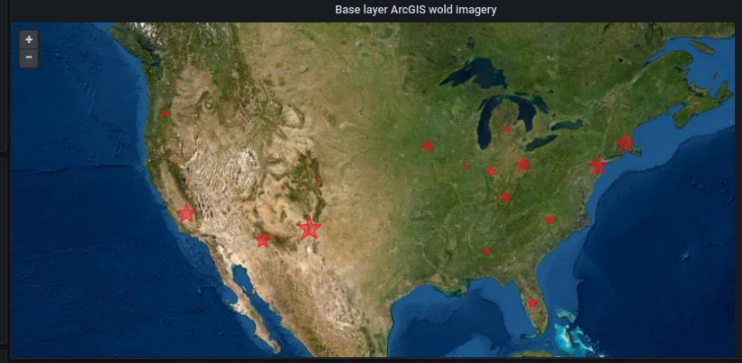


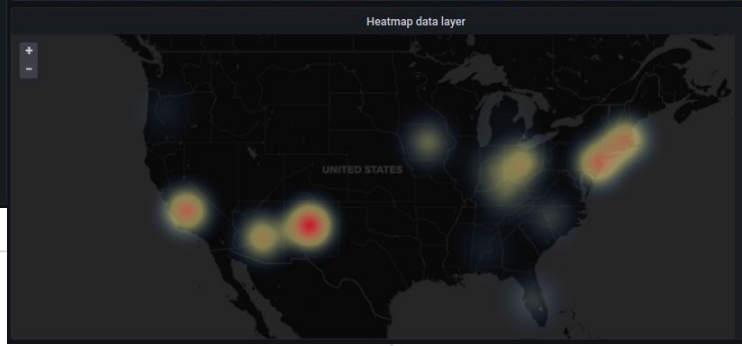




Grafana Labs







Grafana Labs

All of this is Open Source and you can run it yourself

(But we will also sell it to you happily)

For a deeper dive into
Open Source Observability, go to:

grafana.com/about/events/observabilitycon/2022/

Thank you!

richih@richih.org
chaos.social/@RichiH
github.com/RichiH/talks

Grafana Labs

Holiday Readiness - Let's get to LGTM!

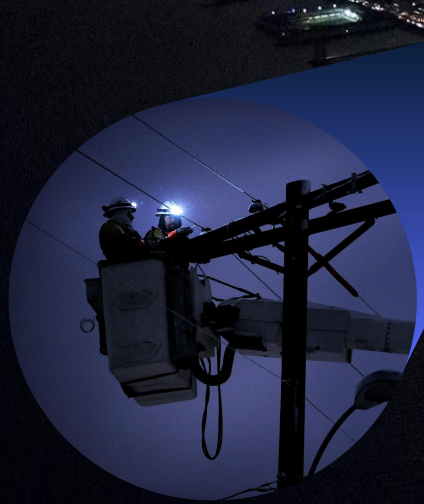
Harish Madhavan

Always in front of Customers @ Kong 🦍 Inc

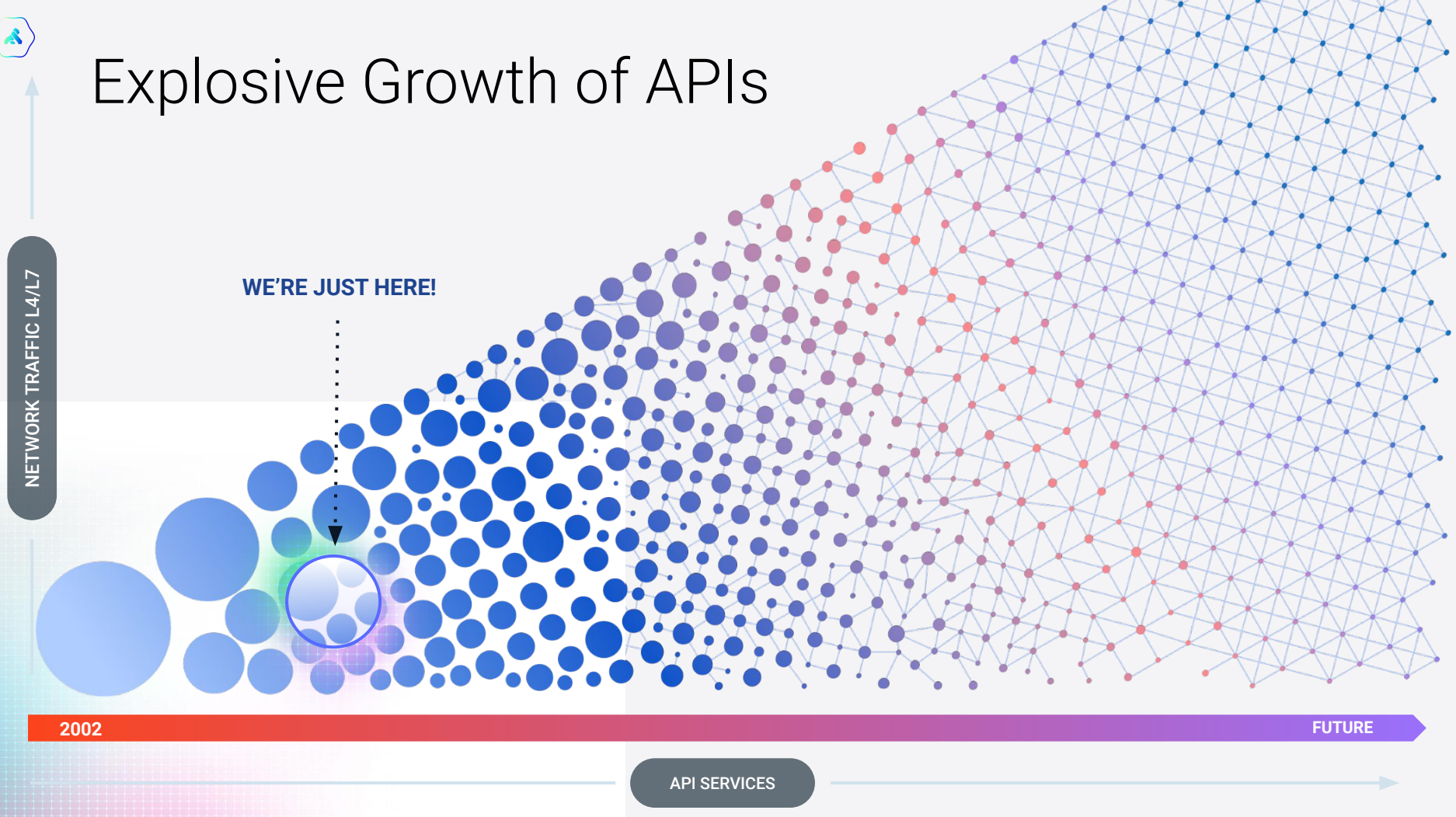
CNCF Singapore 🇸🇬 Chair

CLOUD NATIVE COMPUTING FOUNDATION

Blackout



Explosive Growth of APIs



NETWORK TRAFFIC L4/L7

Service Connectivity
Challenges

2002

FUTURE

(API) SERVICES

The Challenges of Hyper-Connectivity

1 **Reliability**

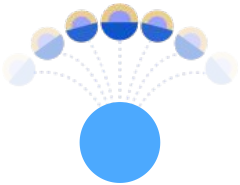
2 **Security**

3 **Observability**

Service Connectivity Patterns

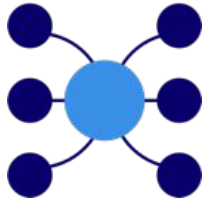
API Gateway
North - South Traffic | Consumer Client to Service
Centralised Deployment

Service Mesh
East - West Traffic | Service to Service
Distributed Deployment



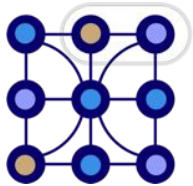
Edge Connectivity

Enable external clients like mobile applications, IoT devices, developers and partners to consume our APIs



Cross-App Connectivity

Enable internal cross-team and cross-application connectivity to consume each others APIs and services

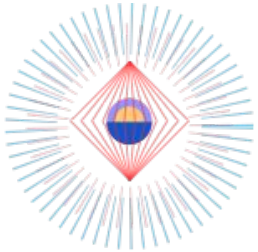


In-App Connectivity

As individual applications become decoupled and distributed, it enables to connect all of their microservices

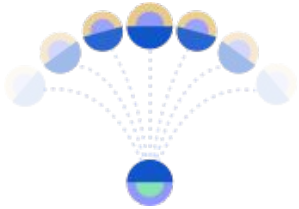
Kuma

Kuma Service Mesh - Key Outcomes



Achieve Zero-Trust Security

Restrict access and encrypt all traffic by default to only complete transactions when identity is verified



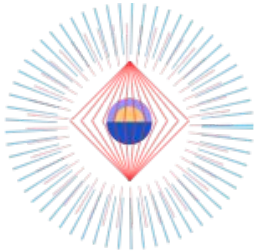
Ensure service connectivity, discovery and traffic reliability

Intelligently route traffic across any platform and any cloud to meet expectations and SLAs

Gain Global Traffic Observability

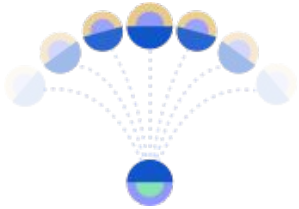
Gain a detailed understanding of service behavior to increase application reliability and the efficiency of teams

Holiday Readiness



Observable Infrastructure

Understanding of cloud native observability techniques - logs, traces and metrics.



Thorough understanding of latencies

Ability to predict the slowness, raise alerts and scale the infrastructure

Drive SRE Principles

Error Budget | Time Series | Forecasting | Provisioning with autoscalers | (and many)

Let’s keep in touch!

Harish Madhavan
CNCF Singapore chair
Kong In, Customer Success
harishmadhavan-k8s

Be a CNCF Singapore member!

https://bit.ly/CNCFSGmembership

Grafana Labs

Let’s hear it for our SREs....

Grafana Labs

CLOUD NATIVE COMPUTING FOUNDATION

Loki

Grafana

Tempo

Mimir

Thank You!

Grafana Labs

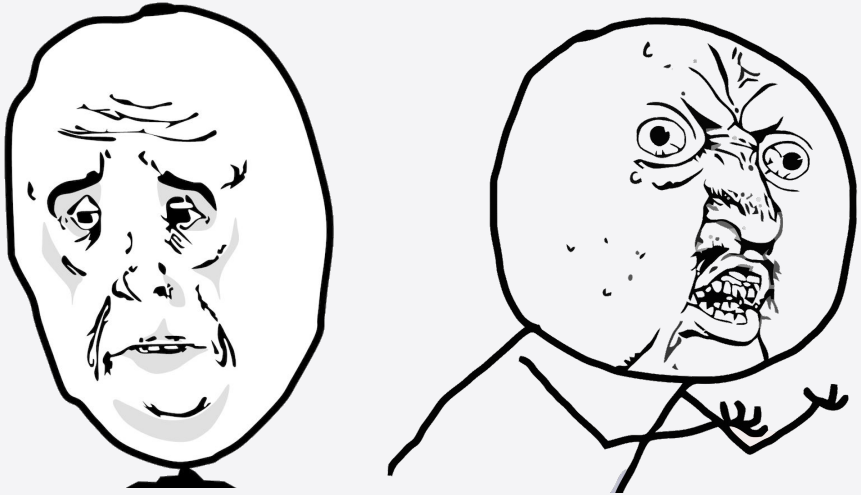
Autoscaling Grafana Mimir with Prometheus, Kubernetes HPA, and KEDA

Ganesh Vernekar
Senior Software Engineer @ Grafana Labs

Problem 1

Manual scaling

- Reactive vs proactive. Paged in the midnight?
- Slow, tedious, API errors that could be avoided.



* they just got paged in the middle of the night for this

Problem 2

Forgot to scale down after the holiday was over?

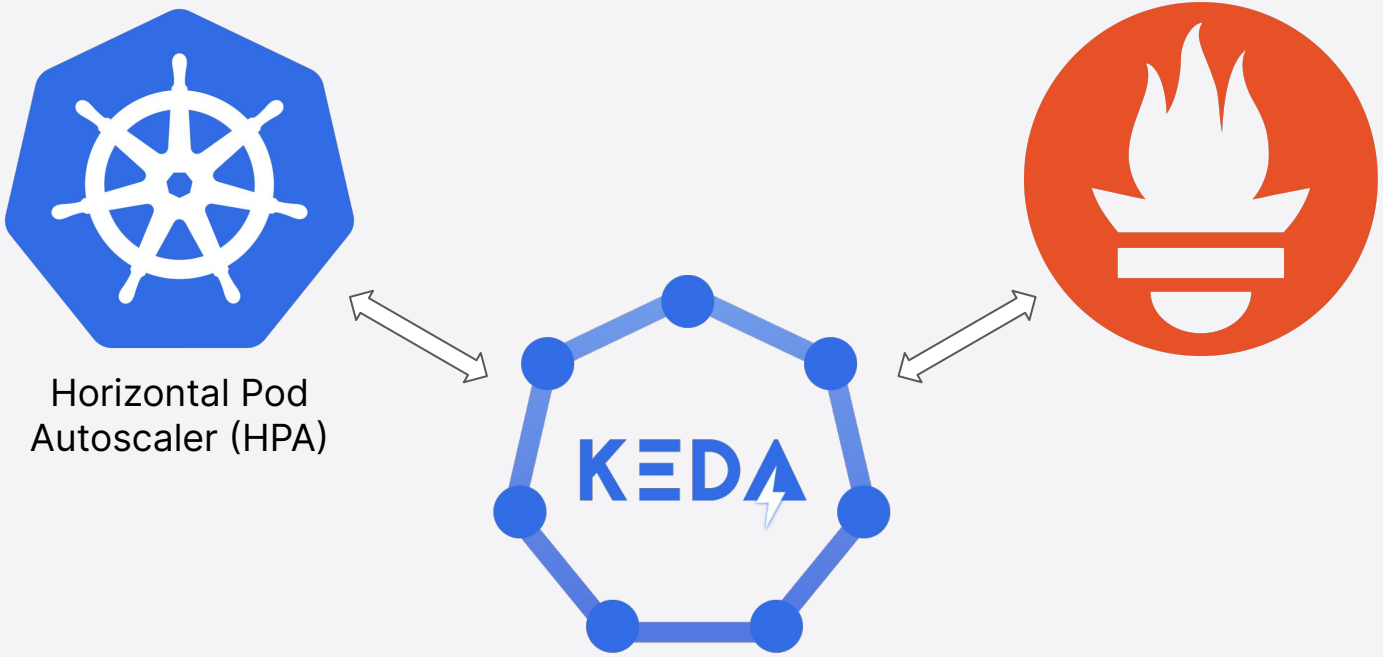
Proposal 1: Autoscale queriers with HPA + KEDA

Proposal 2: Do nothing

Keep waking up in the middle of the night to scale up queriers.

Horizontal Pod
Autoscaler (HPA)

Horizontal Pod Autoscaler (HPA)


Horizontal Pod
Autoscaler (HPA)
KEDA

Example of Grafana Mimir Queriers

Max 75%ile above threshold in last 5 mins? Scale out

- Scale out quickly
- Scale down slowly

Example of Grafana Mimir Queriers

MetricName: cortex_querier_hpa_<namespace>

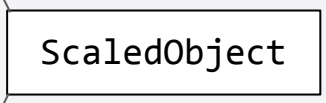
Query:

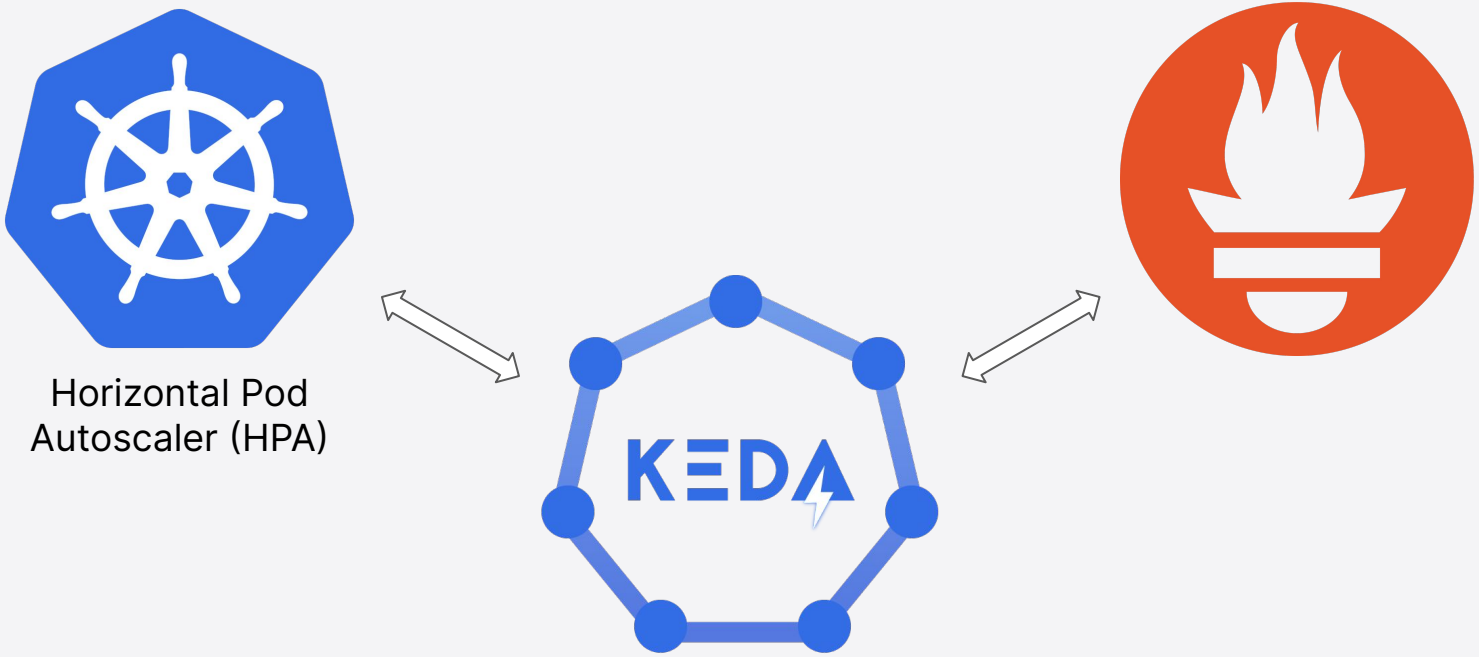
```
sum(
  max_over_time(
    cortex_query_scheduler_inflight_requests{container="X",namespace="Y",quantile="0.75"}[5m]
  )
)
```

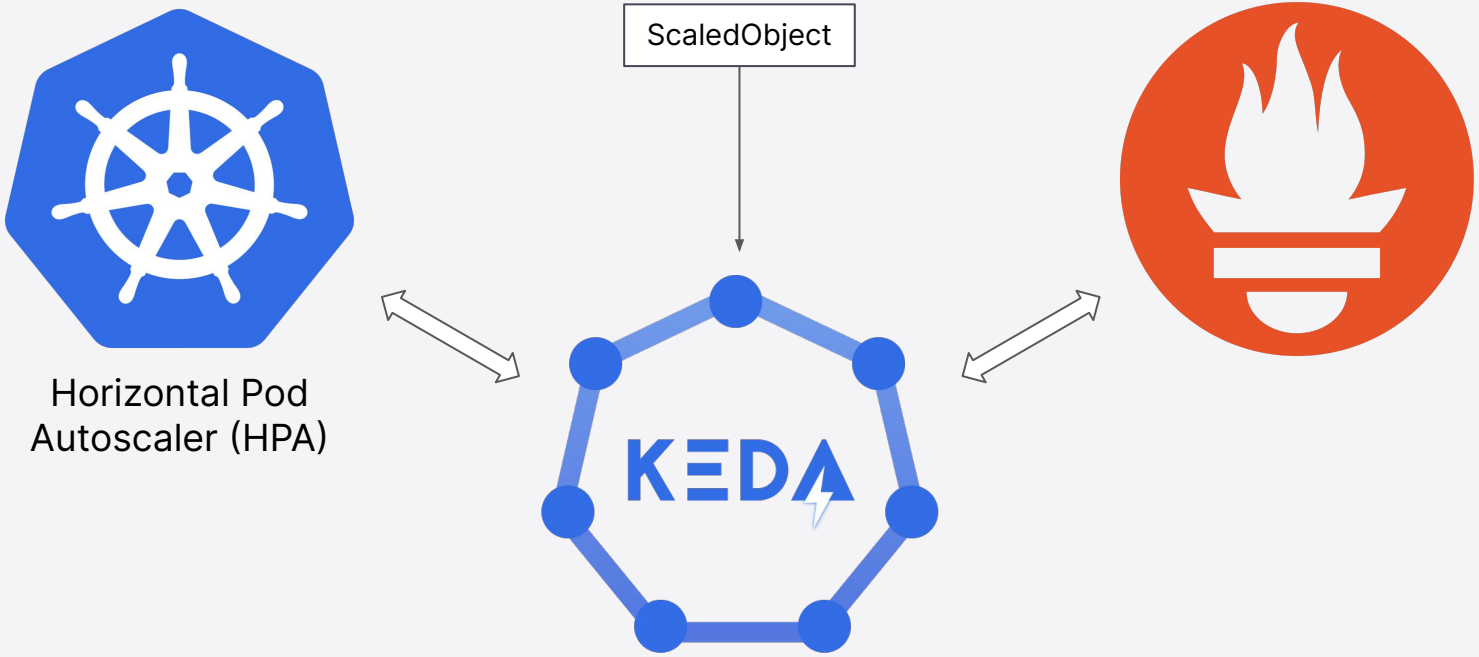
Threshold: 15

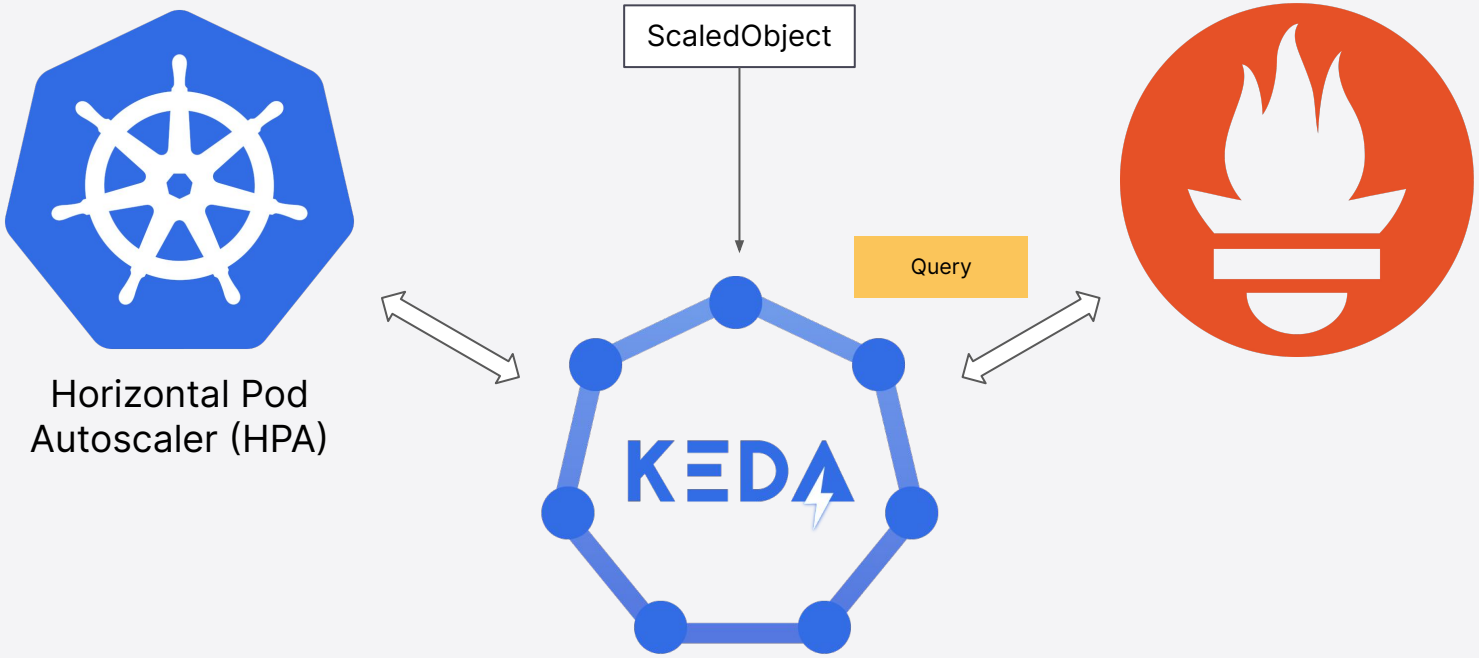
MinReplica: 5

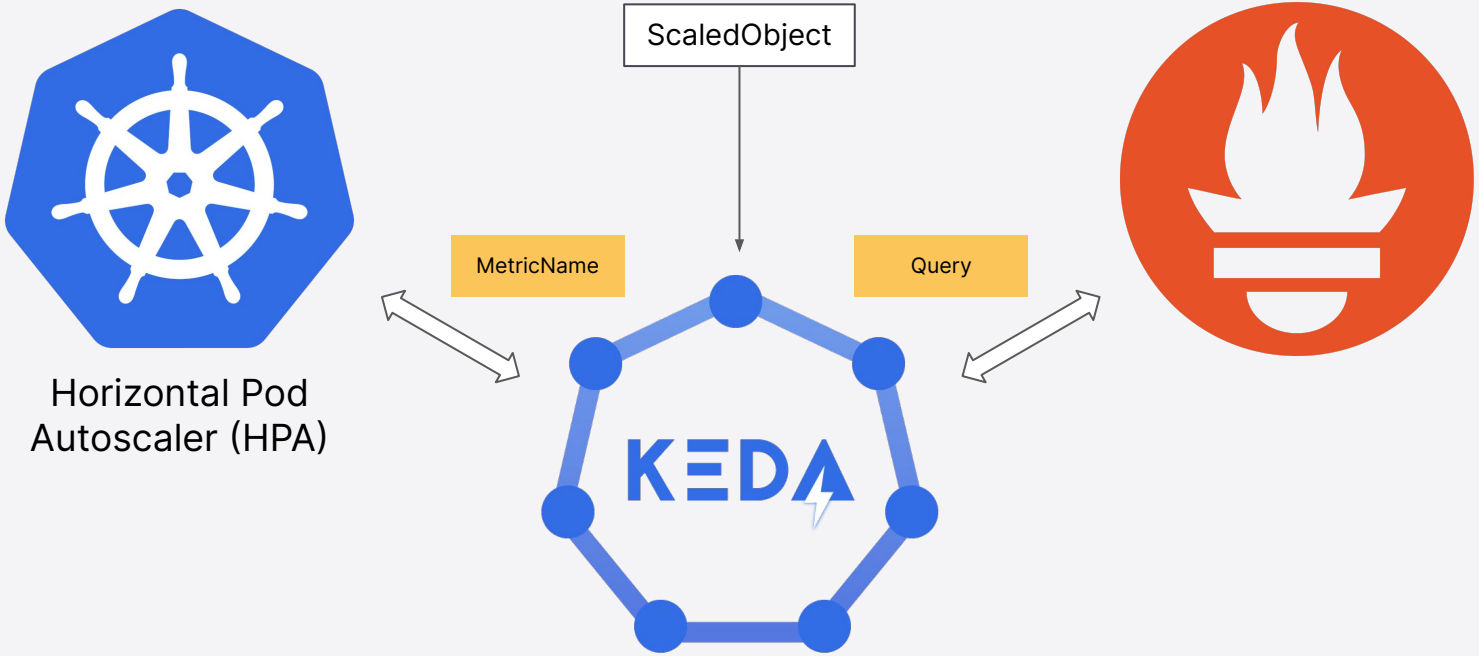
MaxReplica: 20




Horizontal Pod
Autoscaler (HPA)
KEDA


ScaledObject
Horizontal Pod Autoscaler (HPA)
KEDA


ScaledObject
Query
KEDA
Horizontal Pod
Autoscaler (HPA)


ScaledObject
MetricName
Query
KEDA
Horizontal Pod
Autoscaler (HPA)

Example with CPU and Memory

```
// To scale out relatively quickly, but scale in slower, we look at the average CPU utilization per ruler over 5m (rolling window)
// and then we pick the highest value over the last 15m.
// Multiply by 1000 to get the result in millicores. This is due to KEDA only working with ints.
query: 'max_over_time(sum(rate(container_cpu_usage_seconds_total{container="%s",namespace="%s"}[5m]))[15m:]) * 1000' % [
  name,
  $._config.namespace,
],
```

```
// To scale out relatively quickly, but scale in slower, we look at the max memory utilization across all rulers over 15m.
query: 'max_over_time(sum(container_memory_working_set_bytes{container="%s",namespace="%s"})[15m:])' % [
  name,
  $._config.namespace,
],
```

https://github.com/grafana/mimir/blob/main/operations/mimir/autoscaling.libsonnet

How to use KEDA to auto scale Kubernetes deployment using Prometheus?

To use KEDA (Kubernetes-based Event Driven Autoscaling) to automatically scale a Kubernetes deployment using Prometheus, you can follow these steps:

1. Install KEDA on your Kubernetes cluster by following the official installation guide: https://keda.sh/docs/2.4/deploy/
2. Configure your Prometheus server to expose metrics for the deployment you want to autoscale. You can use the Prometheus Operator to do this, which simplifies the process of deploying and managing Prometheus on Kubernetes. The Prometheus Operator documentation can be found here: https://github.com/prometheus-operator/prometheus-operator/tree/main/Documentation
3. Create a ScaledObject in Kubernetes that tells KEDA which deployment to autoscale, and how to calculate the desired number of replicas based on the Prometheus metrics. Here is an example ScaledObject definition:

yaml Copy code

```yaml
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: myapp-scaler
spec:
  scaleTargetRef:
    name: myapp-deployment
  pollingInterval: 5
  triggers:
  - type: prometheus
    metadata:
      serverAddress: http://prometheus-server:9090
      metricName: myapp_request_count
      query: sum(rate(myapp_request_count{status="200"}[1m]))
      threshold: '10'
      minReplicaCount: 1
      maxReplicaCount: 10
```

This ScaledObject tells KEDA to monitor the Prometheus metric `myapp_request_count`, and to calculate the desired number of replicas based on the rate of requests with a 200 status code. If the rate exceeds 10 requests per second, KEDA will scale up the deployment up to a maximum of 10 replicas. If the rate falls below 10 requests per second, KEDA will scale down the deployment to a minimum of 1 replica.

4. Apply the ScaledObject to your Kubernetes cluster using `kubectl apply -f <filename>.yaml`

With these steps, KEDA will automatically adjust the number of replicas for your deployment based on the metrics exposed by Prometheus. This can help ensure that your application has enough resources to handle incoming traffic, while minimizing wasted resources when traffic is low.

Let's keep in touch!

https://bit.ly/fossasiasummit2023

ObservabilityCON
ON THE ROAD • 2023

SINGAPORE
Tuesday, May 16

SYDNEY
Tuesday, May 18

https://bit.ly/ocr2023-singapore
https://bit.ly/syd-obscon-2023

GrafanaCON 2023
June 12-14
Livestreaming around the world

https://bit.ly/grafanacon2023

Richard Hartmann
Director of Community
richih

Ganesh Vernekar
Sr. Software Engineer, Prometheus squad
ganeshvernekar

Celebrating a decade of Grafana Dashboards

Grafana Golden Grot Awards

bit.ly/goldengrotawards2023-apac

Personal Category

(a.k.a. Dashboard confessionals)

Did you create a Grafana dashboard to monitor your sourdough starter, visualize your own fitness metrics, or keep tabs on another aspect of your personal life? We'd love to see it. If your dashboard was created outside of work, this is the category for you.

Professional Category

(a.k.a. Dashboard professionals)

From monitoring maritime dredging vessels to unlocking the power of recruitment data, we've seen teams build dashboards to help meet KPIs, speed development, and smash performance goals. If your dashboard was created in a professional setting, this is the category for you.

Enter your dashboards by April 21

Grafana Labs

Thank you

Grafana Labs

ASK-US-ANYTHING

Get your very own Grafana “Democratize Metrics” t-shirt by completing this survey