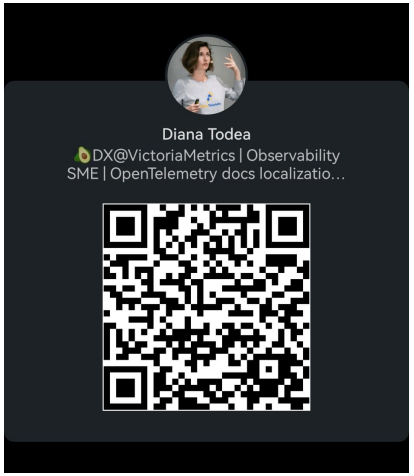
Open Source Cloud-Native Observability with VictoriaMetrics and OpenTelemetry

Cloud Native Valencia | 27 November 2025

Diana Todea - DX Engineer



OpenTelemetry member and contributor

Cloud Native Days Romania organizer

Co-lead CNCF Merge-Forward Neurodiversity

https://github.com/didiViking/Conferences_Talks

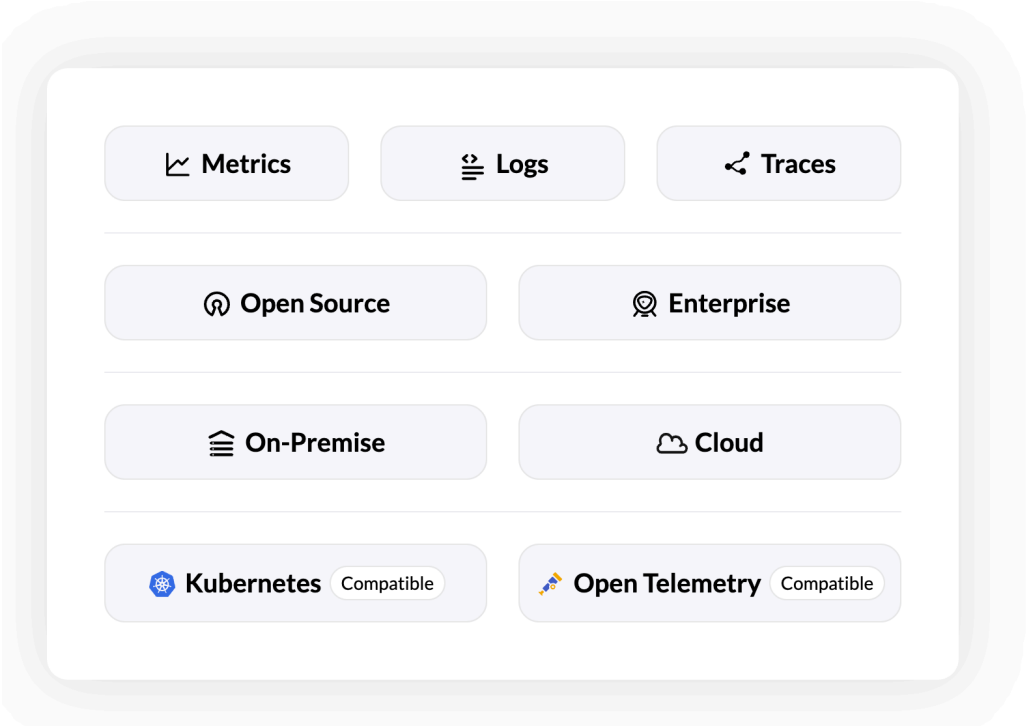
What VictoriaMetrics isn't

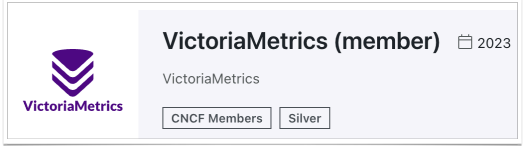
A fork of Prom[illegible]us CNCF project

Named after Queen Victoria [illegible]toria British Columbia

Just a frontend visuali[illegible]ion tool running Clickhou[illegible] or Thanos in the backend

VictoriaMetrics Observability Stack





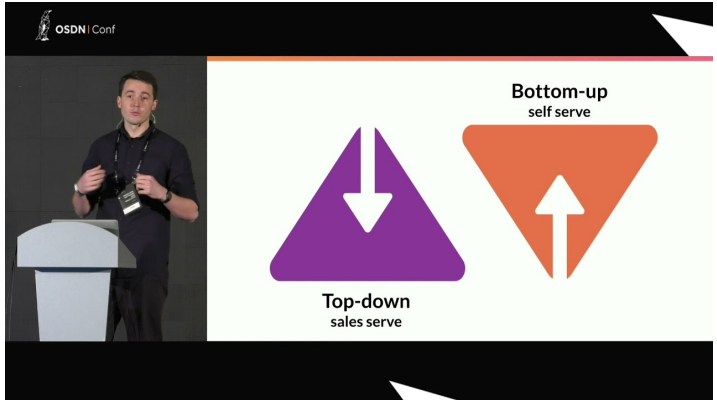
VictoriaMetrics and its Open-Source strategy

How the co-founders built and tried to sell VictoriaMetrics in the beginning

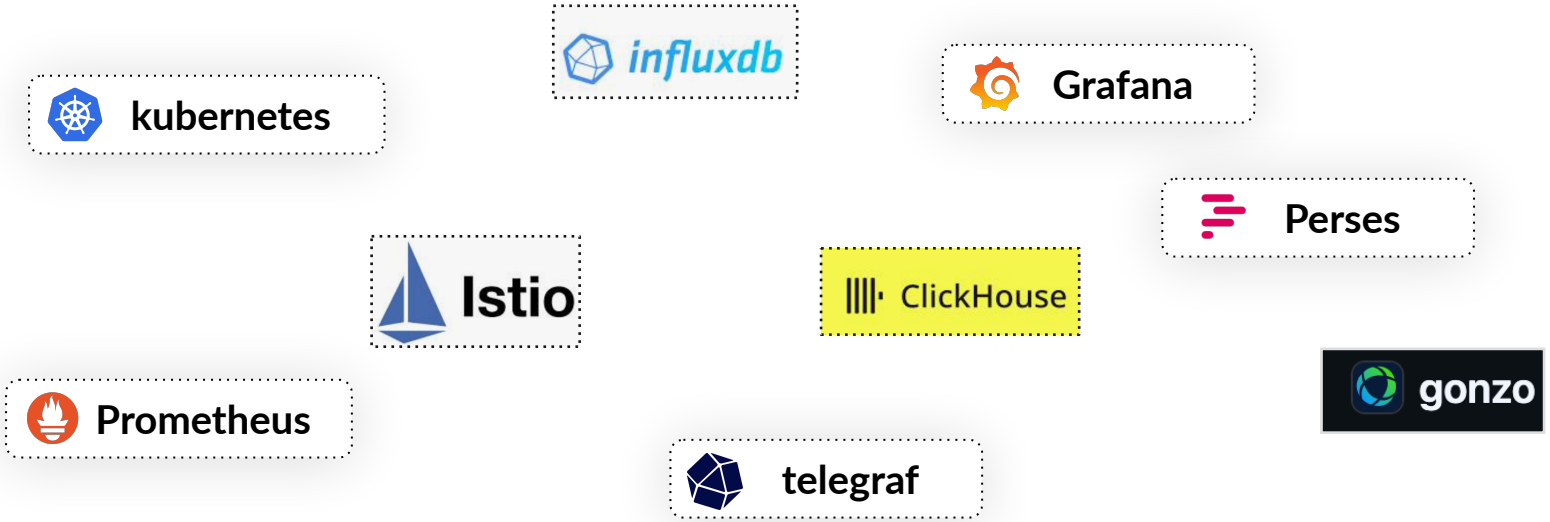
They decided to create an Open Source community and build excellent products

No investors led strategy, no obligation to close the OSS project

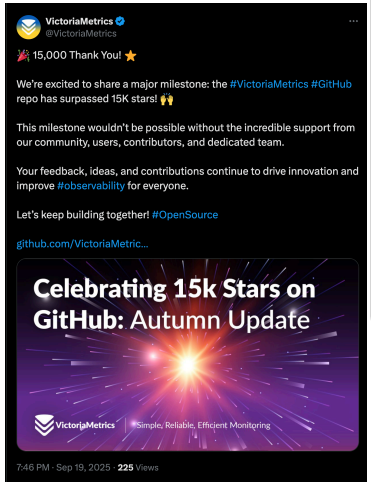
https://youtu.be/-DbbIZzFHIY?si=BKJ9njiTNhdY_xyv

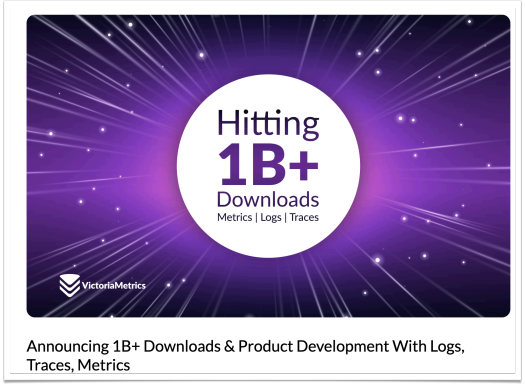


VictoriaMetrics contributes to OSS and Cloud Native projects

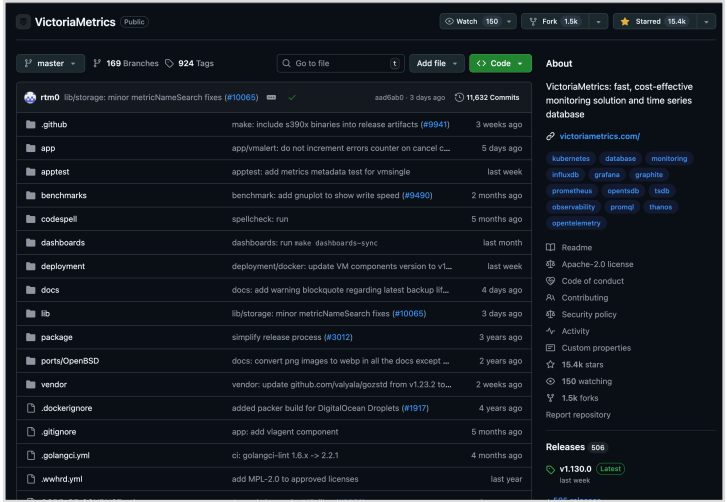


VictoriaMetrics hits 1 billion downloads and 15k+ stars on GitHub

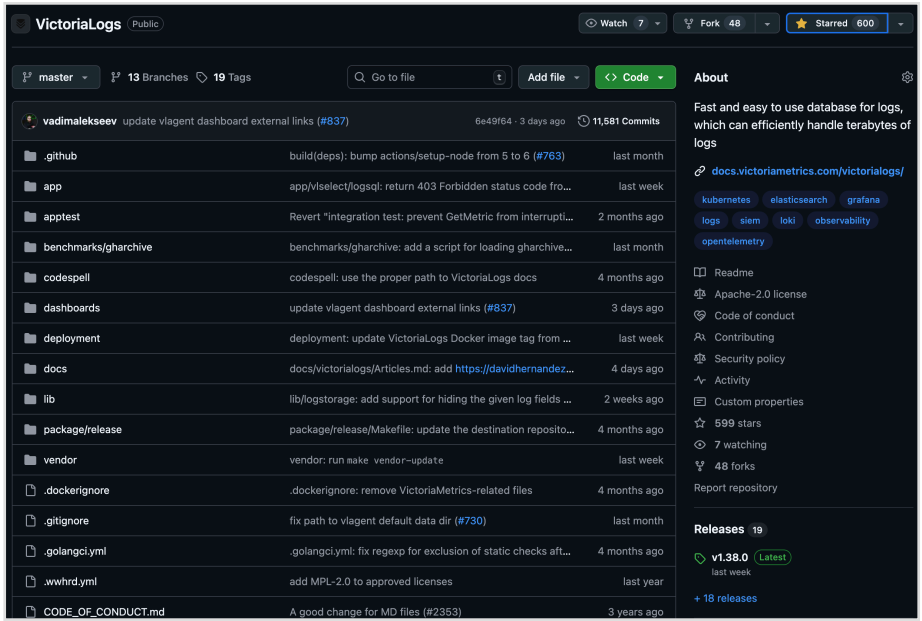


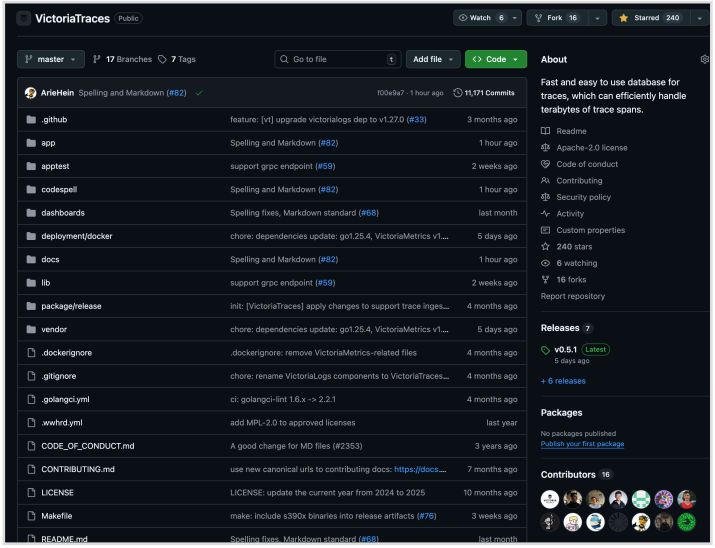


Announcing 1B+ Downloads & Product Development With Logs, Traces, Metrics



New kids on the block: VictoriaLogs and VictoriaTraces





VictoriaMetrics Sustainable Features

1.7x
less memory

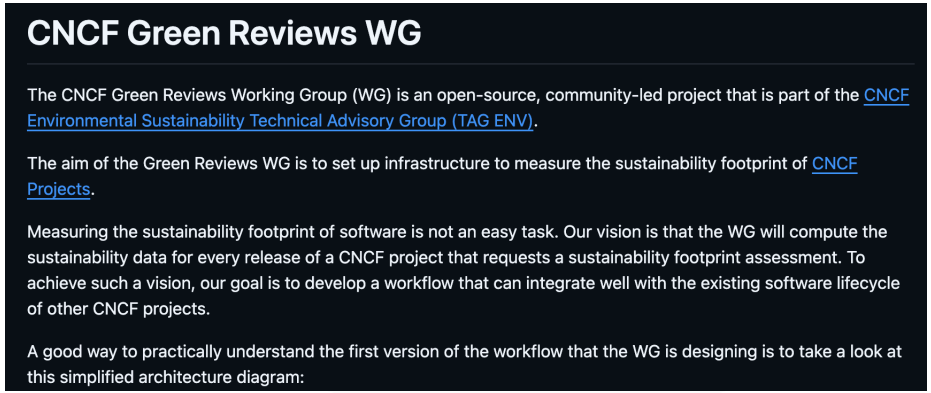
90%
energy cost reduction

2.5x
less disk space

10x
infrastructure cost Savings

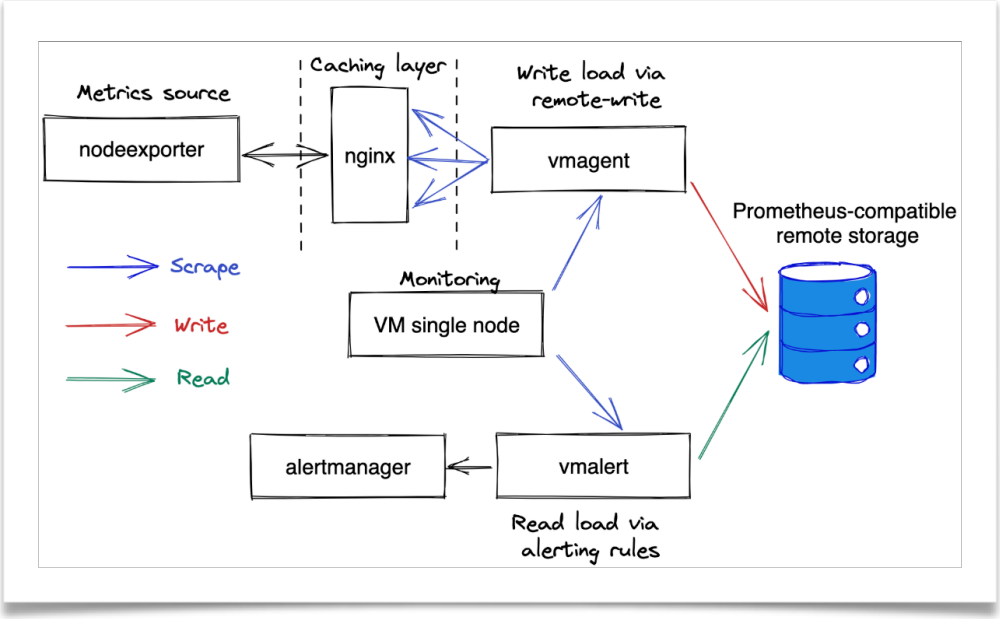
16x
faster query latency

4x
network cost Savings



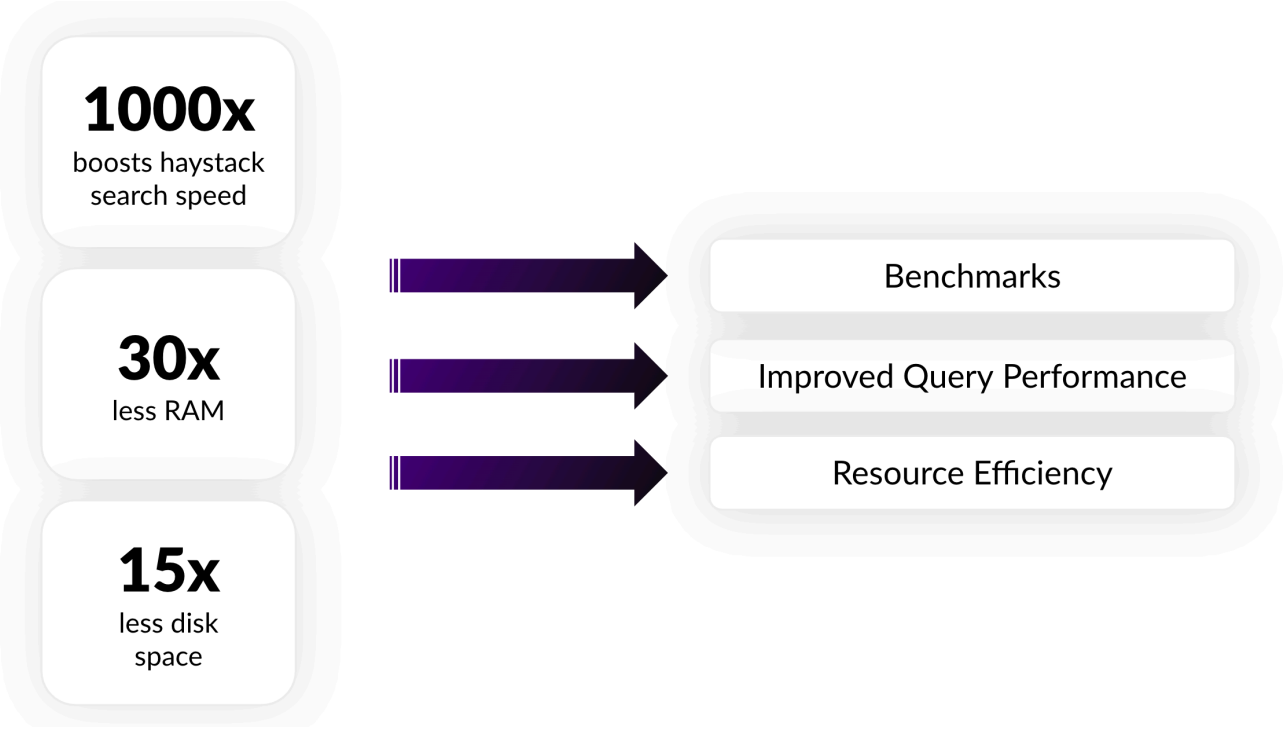
☆https://github.com/cncf-tags/green-reviews-tooling

Run your own benchmarks!

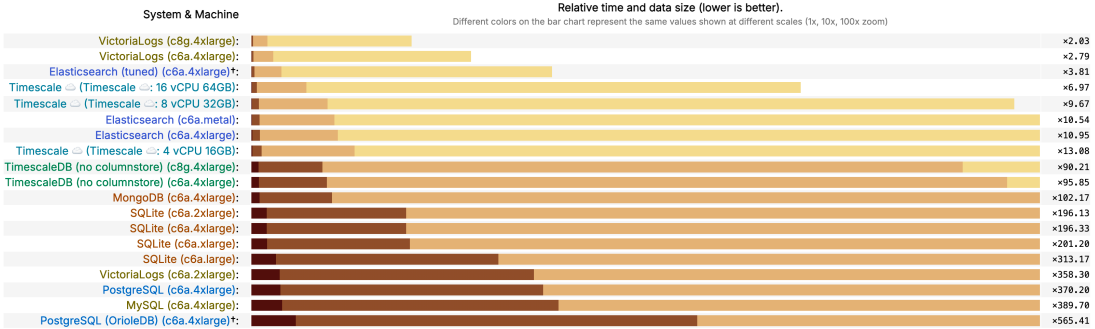


- https://github.com/VictoriaMetrics/prometheus-benchmark
- https://victoriametrics.com/blog/benchmark-100m/

VictoriaLogs Sustainable Features



VictoriaLogs vs. Elasticsearch vs. MongoDB etc.

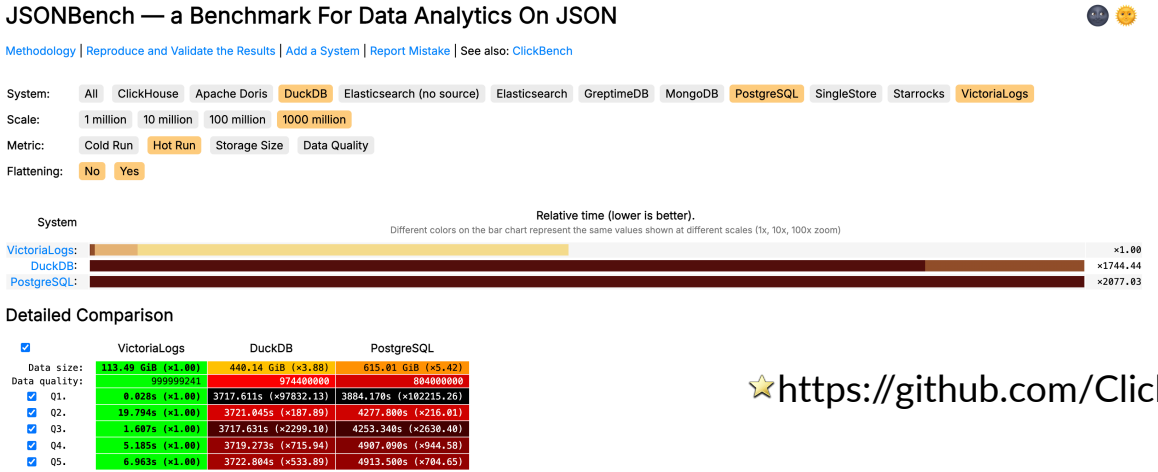


Detailed Comparison

	VictoriaLogs (c8g.4xlarge)	VictoriaLogs (c6a.4xlarge)	Elasticsearch (tuned) (c6a.4xlarge)	Timescale ☁ (Timescale ☁: 16 vCPU 64GB)	Timescale ☁ (Timescale ☁: 8 vCPU 32GB)	Elasticsearch (c6a.metal)	Elasticsearch (c6a.4xlarge)	Timescale ☁ (Timescale ☁: 4 vCPU 16GB)	TimescaleDB (c8g
Load time:	1055s (×1.13)	1050s (×1.12)	7140s (×7.62)	3582s (×3.82)	4100s (×4.38)	9565s (×10.21)	9531s (×10.17)	4536s (×4.84)	
Data size:	15.80 GiB (×1.00)	15.81 GiB (×1.00)	106.47 GiB (×6.75)	17.98 GiB (×1.14)	17.98 GiB (×1.14)	78.85 GiB (×5.00)	79.96 GiB (×5.07)	17.98 GiB (×1.14)	
Q0.	0.044s (×2.39)	0.060s (×3.10)	0.017s (×1.20)	0.064s (×3.28)	0.063s (×3.27)	**0.012s (×1.00)**	0.024s (×1.49)	0.083s (×4.11)	
Q1.	0.026s (×2.20)	0.037s (×2.83)	0.007s (×1.05)	0.252s (×15.86)	0.373s (×23.21)	**0.006s (×1.00)**	0.044s (×3.24)	3.431s (×208.55)	257
Q2.	0.923s (×5.07)	0.574s (×3.17)	0.896s (×4.93)	0.322s (×1.80)	0.504s (×2.79)	0.575s (×3.18)	0.634s (×3.50)	3.901s (×21.26)	25
Q3.	1.018s (×5.91)	0.790s (×4.60)	1.558s (×9.01)	0.261s (×1.56)	0.411s (×2.42)	0.240s (×1.44)	0.680s (×3.97)	4.422s (×25.47)	25
Q4.	1.113s (×3.77)	2.121s (×7.16)	0.503s (×1.72)	14.705s (×49.42)	22.495s (×75.59)	0.290s (×1.01)	**0.297s (×1.03)**	22.636s (×76.06)	2
Q5.	0.929s (×1.74)	1.637s (×3.06)	1.458s (×2.72)	21.977s (×40.81)	43.578s (×80.91)	**0.529s (×1.00)**	0.745s (×1.40)	58.296s (×108.22)	2
Q6.	0.040s (×3.72)	0.057s (×5.00)	0.439s (×33.28)	0.220s (×17.00)	0.318s (×24.33)	0.252s (×19.43)	0.306s (×23.39)	2.393s (×178.04)	255
Q7.	**0.025s (×1.00)**	0.035s (×1.29)	0.265s (×7.90)	0.238s (×7.14)	0.325s (×9.65)	0.903s (×26.28)	0.887s (×25.80)	0.468s (×13.76)	25
Q8.	**1.364s (×1.00)**	2.168s (×1.59)	2.391s (×1.75)	25.253s (×18.39)	80.955s (×58.94)	9.155s (×6.67)	8.898s (×6.48)	91.126s (×66.34)	2
Q9.	**1.808s (×1.00)**	2.901s (×1.60)	3.333s (×1.84)	29.337s (×16.14)	88.163s (×48.50)	13.552s (×7.46)	12.829s (×7.06)	97.452s (×53.61)	2
Q10.	0.279s (×1.07)	0.434s (×1.64)	0.305s (×1.16)	2.502s (×9.28)	9.625s (×35.59)	1.154s (×4.30)	1.258s (×4.69)	13.087s (×48.37)	2
Q11.	0.336s (×1.06)	0.501s (×1.56)	0.359s (×1.13)	3.282s (×10.05)	13.538s (×41.37)	1.410s (×4.34)	1.592s (×4.89)	16.479s (×50.35)	2
Q12.	0.769s (×2.39)	1.290s (×3.99)	**0.316s (×1.00)**	6.921s (×21.28)	7.551s (×23.21)	4.271s (×13.14)	4.254s (×13.09)	15.055s (×46.25)	2

- https://benchmark.clickhouse.com/#system=+ltrc|ehed|noB|yL|gS(|gQ|Lt|m%E2%98%81|%20nu|coog&type=-&machine=-&cluster_size=-&opensource=-&tuned=-&metric=combined&queries=-
- https://github.com/ClickHouse/ClickBench/

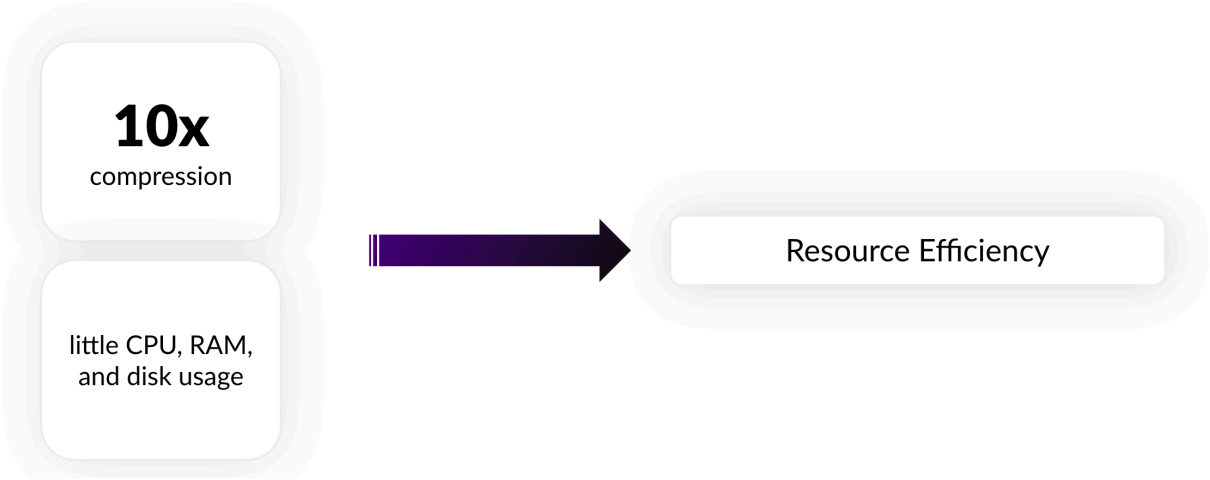
VictoriaLogs vs. DuckDB vs. PostgreSQL



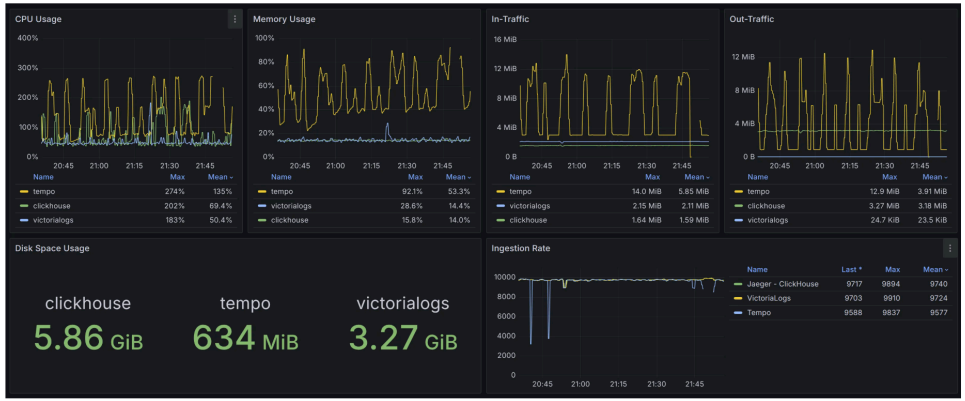
☆https://github.com/ClickHouse/JSONBench

☆https://docs.victoriametrics.com/victorialogs/#benchmarks

VictoriaTraces Sustainable Features



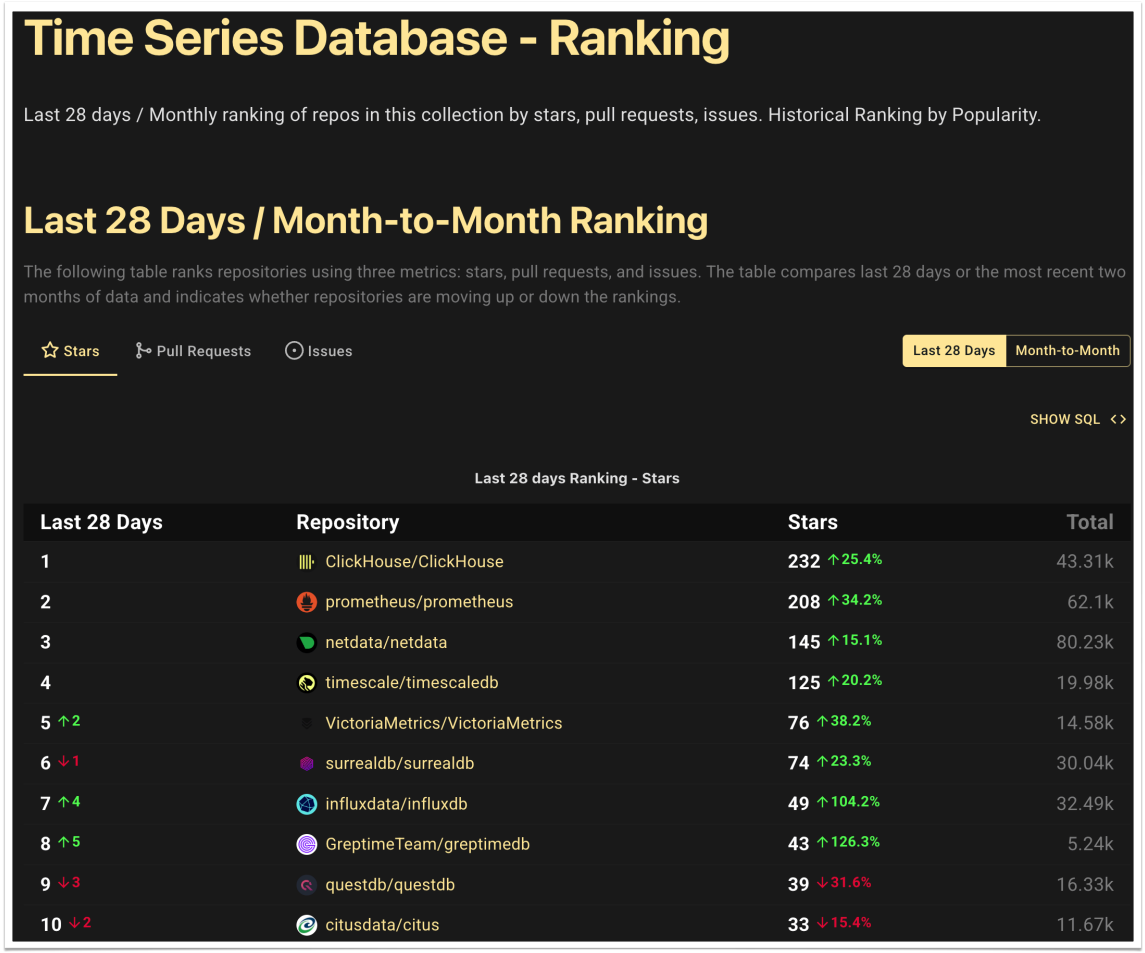
VictoriaLogs vs. Clickhouse vs. Grafana Tempo



Benchmark result overview

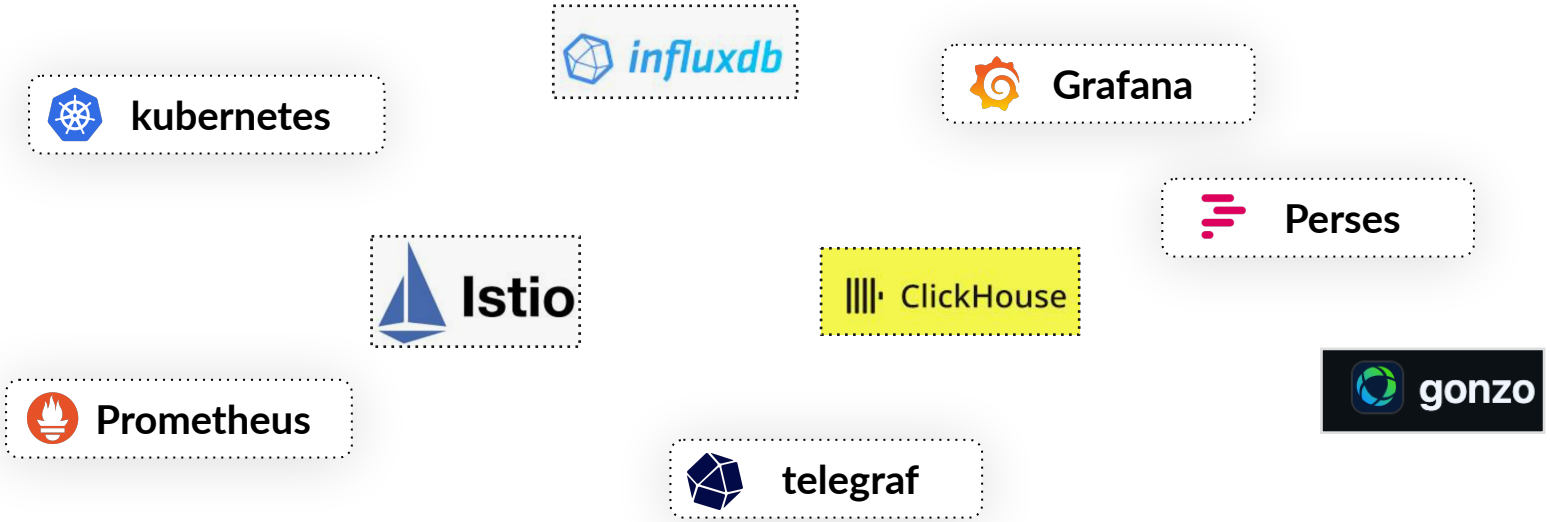
> Why is there no benchmark result for Jaeger + Elasticsearch? And why use 10,000 spans/s?
> The Elasticsearch-based solution began to crash at an ingestion rate as around 5,000 spans/s, and it couldn't remain stable under higher loads with default settings. That's why it's excluded from the final comparison.
> We conducted multiple benchmarks under different loads to ensure all competitors could run smoothly. If you're interested in these more extreme benchmark cases, feel free to comment below or join our Slack channel for discussion.

- ☆3.7x less RAM and up to 2.6x less CPU
- ☆https://victoriametrics.com/blog/dev-note-distributed-tracing-with-victorialogs/



☆https://ossinsight.io/collections/time-series-database

VictoriaMetrics contributes to OSS and Cloud Native projects



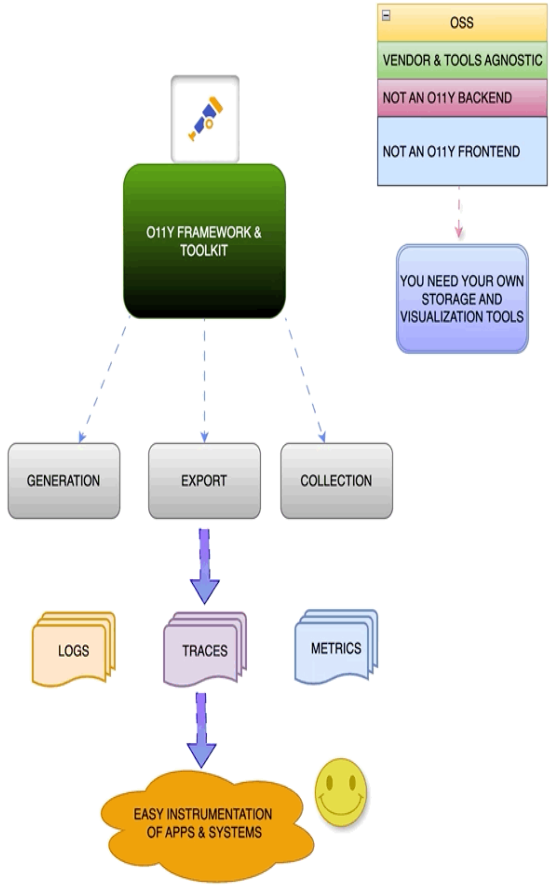
What is OpenTelemetry

OpenTelemetry

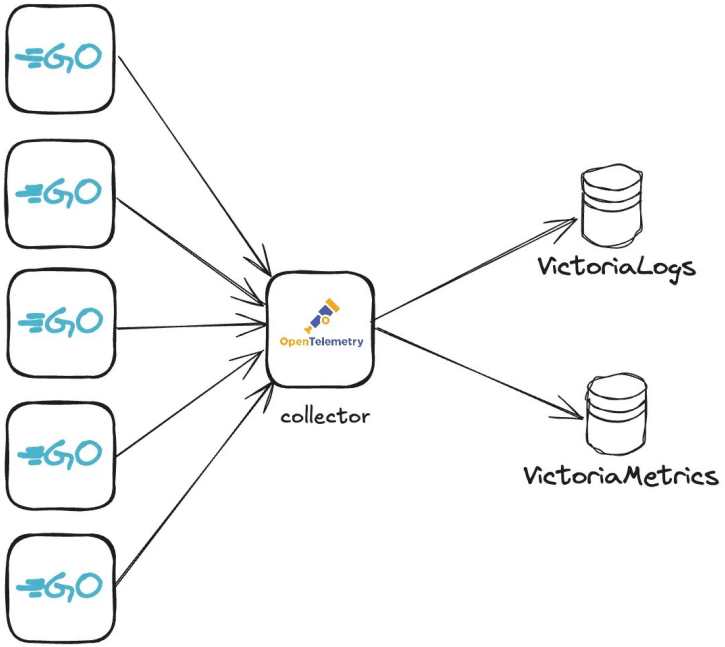
OpenTelemetry community content

OpenTelemetry was accepted to CNCF on May 7, 2019 and moved to the **Incubating** maturity level on August 26, 2021.

VISIT PROJECT WEBSITE



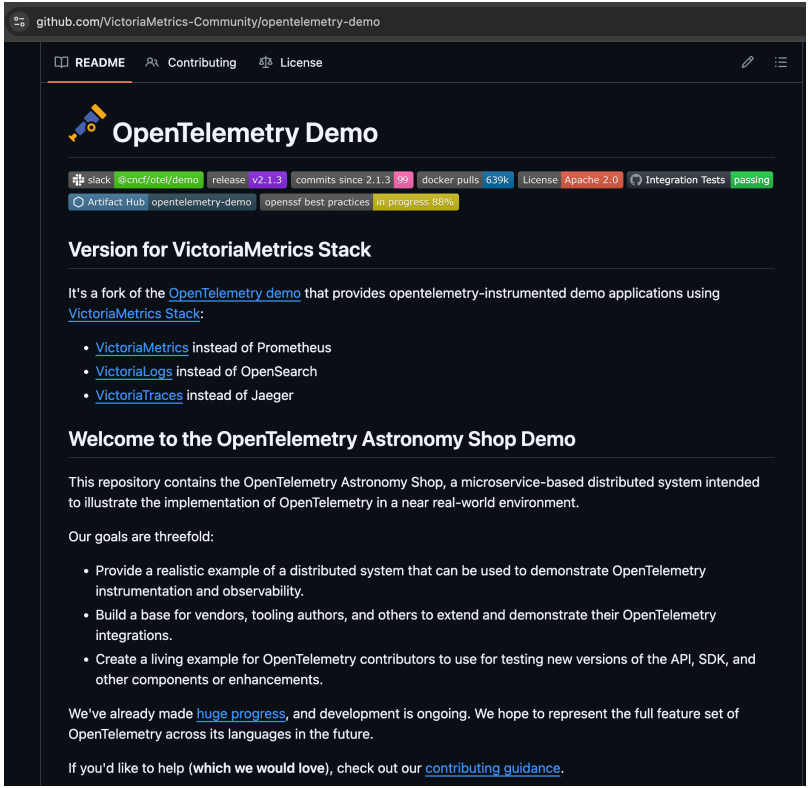
VictoriaMetrics + OpenTelemetry



VictoriaMetrics and VictoriaLogs support ingestion of metrics and logs in the OTel format.
Pre-requirements:
k8s cluster, kubectl, helm

VictoriaMetrics OTel Demo fork

☆https://github.com/VictoriaMetrics-Community/opentelemetry-demo



KubeCon Atlanta 2025: meeting the community

Keep your 👀 on these OpenTelemetry initiatives

README | Code of conduct | Contributing | Apache-2.0 license | Security

OpenTelemetry Weaver

Observability by Design

Treat your telemetry like a public API

CI passing | codecov 78% | Security Audit passing | License Apache 2.0 | Slack #otel-weaver

OpenTelemetry Weaver helps teams build observability by design, enabling consistent, type-safe, and automated telemetry through semantic conventions. With Weaver, you can define, validate, and evolve your telemetry schemas, ensuring reliability and clarity across your systems.

What is Observability by Design?

Have you ever experienced:

- Broken alerts after a deployment because metric names changed?
- Complex, hard-to-understand queries due to inconsistent naming?
- Teams struggling to interpret unclear or undocumented signals?
- Missing critical instrumentation discovered only in production?

Observability by Design solves these problems by treating your observability signals (metrics, traces, logs) as a first-class public API that requires the same quality standards as your code.

An introduction to Weaver and Observability by Design is presented in the official blog post: Observability by Design: Unlocking Consistency with OpenTelemetry Weaver

README | Code of conduct | Contributing | Apache-2.0 license | Security

OpenTelemetry Injector

The OpenTelemetry injector is a shared library (written in Zig) that is intended to be used via the environment variable `LD_PRELOAD`, the `/etc/ld.so.preload` file, or similar mechanisms to inject environment variables into processes at startup.

It serves two main purposes:

- Inject an OpenTelemetry Auto Instrumentation agent into the process to capture and report distributed traces and metrics to the OpenTelemetry Collector for supported runtimes.
- Set resource attributes automatically, (for example Kubernetes related resource attributes and service related resource attributes in environments where this is applicable).

The injector can be used to enable automatic zero-touch instrumentation of processes. For this to work, the injector binary needs to be bundled together with the OpenTelemetry auto-instrumentation agents for the target runtimes.

Official RPM and DEB packages that contain the injector as well as the auto-instrumentation agents are available, and can be downloaded from the releases page. The OpenTelemetry injector Debian/RPM packages install the OpenTelemetry auto-instrumentation agents, the `libotelinject.so` shared object library, and a default configuration file to automatically instrument applications and services to capture and report distributed traces and metrics to the OpenTelemetry Collector

The `opentelemetry-injector` deb/rpm package installs and supports configuration of the following Auto Instrumentation agents:

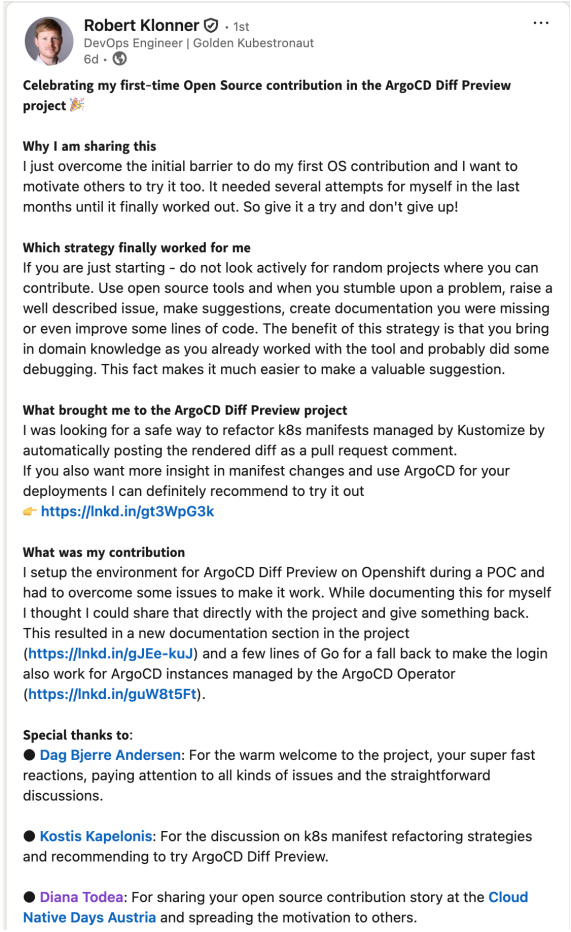
- Java
- Node.js
- .NET

Support local communities

#otel-docs-localization
#otel-localization-es
#cloud-native-valencia
#cnd-romania
#otel-localization-ro

Inspire other Cloud Native contributors

MOTIVATION = 🔑



Robert Klonner · 1st
DevOps Engineer | Golden Kubestronaut
6d ·

Celebrating my first-time Open Source contribution in the ArgoCD Diff Preview project 🎉

Why I am sharing this
I just overcome the initial barrier to do my first OS contribution and I want to motivate others to try it too. It needed several attempts for myself in the last months until it finally worked out. So give it a try and don't give up!

Which strategy finally worked for me
If you are just starting - do not look actively for random projects where you can contribute. Use open source tools and when you stumble upon a problem, raise a well described issue, make suggestions, create documentation you were missing or even improve some lines of code. The benefit of this strategy is that you bring in domain knowledge as you already worked with the tool and probably did some debugging. This fact makes it much easier to make a valuable suggestion.

What brought me to the ArgoCD Diff Preview project
I was looking for a safe way to refactor k8s manifests managed by Kustomize by automatically posting the rendered diff as a pull request comment.
If you also want more insight in manifest changes and use ArgoCD for your deployments I can definitely recommend to try it out
👉 https://lnkd.in/gt3WpG3k

What was my contribution
I setup the environment for ArgoCD Diff Preview on Openshift during a POC and had to overcome some issues to make it work. While documenting this for myself I thought I could share that directly with the project and give something back. This resulted in a new documentation section in the project (https://lnkd.in/gJEe-kuJ) and a few lines of Go for a fall back to make the login also work for ArgoCD instances managed by the ArgoCD Operator (https://lnkd.in/guW8t5Ft).

Special thanks to:
● Dag Bjerre Andersen: For the warm welcome to the project, your super fast reactions, paying attention to all kinds of issues and the straightforward discussions.

● Kostis Kapelonis: For the discussion on k8s manifest refactoring strategies and recommending to try ArgoCD Diff Preview.

● Diana Todea: For sharing your open source contribution story at the Cloud Native Days Austria and spreading the motivation to others.

Make your own journey!

- Your motivation is your super power!
- Diversify your contributions
-  code-no-code contributions
- Pair up and mentor
- Share your end user stories!

Resources

Full-Stack Observability with VictoriaMetrics in the OTel Demo

https://opentelemetry.io/community/
https://training.linuxfoundation.org/certification/opentelemetry-certified-associate-otca/
https://github.com/open-telemetry/opentelemetry-injector
https://github.com/open-telemetry/weaver
https://play.victoriametrics.com/
https://play-vmlogs.victoriametrics.com/
https://play-vtraces.victoriametrics.com/
https://play-grafana.victoriametrics.com/dashboards
https://docs.victoriametrics.com/guides/getting-started-with-opentelemetry/
https://github.com/VictoriaMetrics-Community/opentelemetry-demo

Explore VictoriaMetrics Solutions

VictoriaMetrics

Open Source Available on Github

VictoriaTraces

Open Source Available on Github

VictoriaLogs

Open Source Available on Github

Muchas gracias!

Bsky: @didiviking.bsky.social
X: @dianavtodea
Github: @didiViking/Conferences_Talks
LinkedIn: @diana-todea-b2a79968

VictoriaMetrics Community